



Universidad de las Ciencias
Informáticas

Universidad de las Ciencias Informáticas

Facultad 9

Desarrollo de la Arquitectura de Software del proyecto Plataforma VideoWeb.

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

Autor:

Frank Torres Rodríguez

Tutor:

Ing. Yunier Albrecht Delgado

Ciudad de La Habana, mayo de 2009

“Año del 50 aniversario del triunfo de la Revolución”

DATOS GENERALES

Generales de la investigación

Título: Desarrollo de la Arquitectura de Software del proyecto Plataforma VideoWeb

- **Clasificación de la tesis:** Informe de investigación práctico del desarrollo de un rol de producción determinado.

Generales del diplomante

- **Nombre y apellidos:** Frank Torres Rodriguez
- **Sexo:** M
- **Grupo:**9504
- **Correo electrónico:** ftrodriguez@estudiantes.uci.cu

Generales del tutor

- **Nombre y apellidos:** Ing. Yunier Albrecht Delgado.
- **Categoría docente:** Adiestrado
- **Grado científico:**
- **Centro de trabajo:** Universidad de las Ciencias Informáticas
- **Cargo:** Jefe del Polo de Video y Sonido Digital.
- **Título de la especialidad de graduado:** Ingeniero en Ciencias Informáticas
- **Año de graduación:** 2007
- **Institución en la que se graduó:** Universidad de las Ciencias Informáticas
- **Correo electrónico:** yalbrecht@uci.cu
- **Teléfono del centro de trabajo:** 835 8284
- **Teléfono particular:** 835 8902

Generales del asesor



Plataforma VideoWeb

- **Nombre y apellidos:** Anisleidis Infante Rodriguez
- **Categoría docente:** Adiestrado
- **Grado científico:**
- **Centro de trabajo:**
- **Cargo:** Profesor
- **Título de la especialidad de graduado:** Lic. en Lenguas Extranjeras con Segunda Lengua
- **Año de graduación:** 2007
- **Institución en la que se graduó:** Universidad de Oriente
- **Correo electrónico:** anisleidisi@uci.cu
- **Teléfono del centro de trabajo:** 837 25 60
- **Teléfono particular:** 835 88 70

Generales del oponente

- **Nombre y apellidos:** Julio Alberto Leyva Duran
- **Categoría docente:** Adiestrado
- **Grado científico:**
- **Centro de trabajo:**
- **Cargo:** Profesor
- **Título de la especialidad de graduado:** Ingeniero en Ciencias Informáticas
- **Año de graduación:** 2008
- **Institución en la que se graduó:** Universidad de las Ciencias Informáticas
- **Correo electrónico:** jaleyva@uci.cu
- **Teléfono del centro de trabajo:** 835 2556
- **Teléfono particular:**



AGRADECIMIENTOS

Primeramente a mis padres por luchar conmigo todos estos años y hacer posible este sueño.

A mi novia Annies por comprenderme y apoyarme en estos días de tensión.

A los miembros del Proyecto VideoWeb que han contribuido a la realización de este trabajo.

Al compañero tutor Albrecht y mis amigos de lucha en la tesis Angel, Yordany y Maggie

A todos mis amigos y compañeros de grupo, ya son cinco años...

A la medio vieja tropa de UCITeVe.

A todos mis amigos de Holguín



Plataforma VideoWeb

DEDICATORIA

Dedico el presente trabajo a mis padres Isabel y Frank por su apoyo y cariño, sepan que estaré eternamente agradecido por todo lo que han hecho por mí. De forma especial quisiera dedicar a dos personas que ya no están presente pero su amistad y cariño serán recordadas eternamente, mi tío Papito y mi primo Manuel Alejandro.



Plataforma VideoWeb

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> 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 _____.

Frank Torres Rodríguez

Yunier Albrecht Delgado



OPINIONES Y AVALES



OPINIÓN DEL TUTOR



RESUMEN

Debido al auge de los servicios de video y audio por la Web el Polo de Video y Sonido Digital de la Facultad 9 de la Universidad de las Ciencias Informáticas, Cuba, decidió implementar la Plataforma VideoWeb con el objetivo de insertarse en el mercado.

El presente trabajo pretende conformar la propuesta de una arquitectura de software robusta y confiable que permita el desarrollo de un sistema modificable, interoperable, seguro, funcional y disponible, que asegure el mantenimiento, flexibilidad y reusabilidad de la Plataforma VideoWeb, apoyándose en el uso de herramientas libres.

Se realizó el análisis de plataformas que brindan servicios de audio y video at través de la web y los elementos relacionados con la Arquitectura de Software, los principales conceptos, estilos y patrones arquitectónicos. Se abordaron temas sobre las tecnologías de desarrollo como CMS y Framework, lenguajes de programación y Sistema Gestor de Base de Datos. Se analizaron las principales metodologías de Desarrollo de Software y las tecnologías de Media Streaming para la transmisión de video y audio a través de la web.

Se realizo la propuesta arquitectónica para la Plataforma VideoWeb seleccionando el CMS Drupal para el desarrollo de la aplicación, usando PHP como lenguaje de programación, Darwin Streaming Server como servidor de Streaming, PostgreSQL como Sistema Gestor de Base de Datos y servidor web Apache. Fue seleccionada como metodología de Desarrollo de Software el Proceso Unificado (RUP) usando como herramienta CASE Visual Paradigm. Se propusieron dos estrategias de evaluación de arquitectura de software con el objetivo de identificar los riesgos y fortalezas de la propuesta y poder tomar medidas correctivas a tiempo.

PALABRAS CLAVES

Arquitectura, video, VideoWeb, patrones, prueba, streaming.



ABSTRACT

Due to the rise of video and audio services on the Web, The Pole of Video and Digital Sound, belonging to the Faculty #9 of The University of Informatics Sciences, Cuba, decided to implement the VideoWeb Platform in order to enter the market.

This work is focus to present a proposal of robust and reliable software architecture to allow the development of a modifiable, interoperable, secure, functional and available system, to ensure the maintenance, reusability and flexibility of the VideoWeb Platform, relying on the use of free tools.

The analysis of platforms that provide audio and video at web and elements related to software architecture key concepts, patterns and architectural styles was accomplished. Topics about development technology such as CMS and Framework, programming languages and Database Management System were analyzed. The main software development methodologies and the Media Streaming technologies for video and audio transmission throughout the web were also analyzed.

The architectural proposal for the VideoWeb Platform was accomplished, selecting the Drupal CMS for developing the application using PHP as a programming language, Darwin Streaming Server as Streaming Server, PostgreSQL as a Database Management System and Apache Web server. Rational Unified Process (RUP) was selected as Software Development Methodology using Visual Paradigm as CASE tool. Two strategies for evaluating software architecture were proposed with the aim of identifying the risks and strengths of the proposal and in order to take corrective measures in time.

KEYWORDS

Architecture, video, VideoWeb, patterns, styles, streaming.

ÍNDICE DE TABLAS Y FIGURAS

Figura 1: Comparación entre plataformas	7
Figura 2: Un pequeño catálogo de modelos arquitectónicos organizado por las relaciones es-un.....	11
Figura 3: Arquitectura en Tres Capa.....	13
Figura 4: Modelo Vista Controlador.	14
Figura 5: Arquitectura Orientada a Servicios.	15
Figura 6: Arquitectura multicapas orientadas a objetos.....	15
Figura 7: Comparación entre Drupal, Joomla! y Plone.	19
Figura 8: Arquitectura del Streaming	25
Figura 9: Diferencias entre metodologías ágiles y no ágiles (18)	34
Figura 10: Interrelación entre las prácticas.....	37
Figura 11: Características del Servidor de archivos multimedia.....	49
Figura 12: Cantidad de Casos de Uso del sistema por módulos.....	51
Figura 13: Vista Lógica	52
Figura 14: Módulo de gestión de archivos multimedia.....	53
Figura 15: Módulo Administración	54
Figura 16: Capa Modelo	55
Figura 17: Diagrama de Despliegue.	57

Figura 18: Paquetes principales de la aplicación.....	59
Figura 19: Componentes de la capa Vista.....	60
Figura 20: Componentes de la capa Controlador.....	61
Figura 21: Componentes de la capa Modelo.....	61
Figura 22: Componentes de la capa Núcleo Drupal.....	62
Figura 23: Descripción de atributos de calidad observables vía ejecución.	67
Figura 24: Descripción de atributos de calidad no observables vía ejecución.	68
Figura 25: Clasificación de las Técnicas de Evaluación.....	69
Figura 26: Fases del ATAM.....	76
Figura 27: TV.com.....	86
Figura 28: es.youtube.com.....	86
Figura 29: Video Web Perú.....	87
Figura 30: Inter-Nos.....	87
Figura 31: Administración del Darwin Streaming Server.....	88
Figura 32 RUP en dos dimensiones.....	89
Figura 33: Vista de Caso de Uso del Módulo de Gestión de Contenido.....	90
Figura 34: Vista de Caso de Uso del Módulo de Administración.....	90
Figura 35: Breve descripción de los casos de usos arquitectónicamente significativos.....	92
Figura 36: Distribución física alternativa.....	93



Plataforma VideoWeb

Figura 37: Prueba usando CMS Joomla 1.5.....	95
Figura 38: Resultado por Usuario	95
Figura 39: Resultados Generales	95
Figura 40: Prueba usando CMS Drupal 6.10.....	96
Figura 41: Resultado por Usuario	96
Figura 42: Resultados Generales	97
Figura 43: Rendimiento del servidor de streaming.	98



ÍNDICE

Introducción	1
Capítulo 1: Fundamentación Teórica.....	4
1.1. Sistemas similares a la Plataforma VideoWeb.....	4
1.2. Arquitectura de Software	7
1.3. Definición de Arquitectura de Software	7
1.4. Estilos y Patrones Arquitectónicos	9
1.4.1. Estilo Arquitectónico.....	10
1.4.2. Patrones Arquitectónicos	11
1.4.2.1. Arquitectura en tres capas.....	12
1.4.2.2. Modelo Vista Controlador (MVC):	13
1.4.2.3. Arquitectura Orientada a Servicios (SOA):	14
1.4.2.4. Arquitecturas multicapas orientadas a objetos	15
1.5. Tecnologías para el Desarrollo de Aplicaciones	16
1.5.1. CMS	16
1.5.1.1. Principales CMSs.....	17
1.5.2. Framework	20
1.6. Lenguajes de Programación del lado del servidor.	21
1.6.1. PHP	21



Plataforma VideoWeb

1.6.2.	Java Server Pages (JSP)	22
1.6.3.	Python	22
1.7.	Sistema Gestor de Base de Datos	22
1.7.1.	PostgreSQL.....	23
1.7.2.	MySQL	23
1.8.	Media Streaming (Streaming de audio y video).	24
1.8.1.	La arquitectura del streaming.....	25
1.8.2.	Streaming en vivo y bajo demanda.....	26
1.8.3.	Principales Servidores Streaming	26
1.8.4.	FFmpeg y FFServer	27
1.8.5.	Darwin Streaming Server	28
1.8.6.	VideoLan (VLC).....	28
1.8.7.	Feng	29
1.8.8.	MPEG4IP	30
1.8.8.1.	Características del reproductor.....	30
1.8.8.2.	Características de Mp4live	30
1.8.8.3.	Herramientas de codificación.....	31
1.8.9.	MEC4	31
1.8.9.1.	Principales características:.....	31
1.9.	Metodología de Desarrollo de Software	32



Plataforma VideoWeb

1.9.1.	Proceso Unificado (RUP)	34
1.9.2.	Extreme Programming (XP)	35
1.9.3.	Microsoft Solution Framework (MSF)	37
1.10.	Conclusiones	38
Capítulo 2: Arquitectura Propuesta		40
2.1.	Metodología de Desarrollo de Software	40
2.2.	Sistema Gestor de Base de Datos	40
2.3.	Herramienta de desarrollo	41
2.3.1.	Selección de la Herramienta de Desarrollo	42
2.4.	Modelo Vista Controlador (MVC)	42
2.5.	Servidor Web	43
2.6.	Lenguaje de Programación	43
2.7.	Entorno de desarrollo integrado (IDE)	44
2.8.	Herramienta para el modelado del sistema	44
2.9.	Servidor Streaming	45
2.10.	Servidor FTP	46
2.11.	Conclusiones	46
Capítulo 3: Descripción de la arquitectura de software.		48
3.1.	Requerimientos no funcionales del sistema	48



Plataforma VideoWeb

3.1.1.	Requerimientos de Software	48
3.1.2.	Requerimientos de Hardware.....	48
3.1.3.	Restricciones en el diseño y la implementación.....	50
3.1.4.	Requerimientos de apariencia o interfaz externa	50
3.1.5.	Requerimientos de confidencialidad	50
3.1.6.	Requerimientos de disponibilidad	50
3.1.7.	Requerimientos de Usabilidad	50
3.1.8.	Requerimientos de Soporte.....	51
3.1.9.	Rendimiento	51
3.2.	Vista de casos de uso	51
3.3.	Vista Lógica	52
3.3.1.	Capa Vista:.....	52
3.3.2.	Capa Controlador:	53
3.3.2.1.	Gestión de Contenido.....	53
3.3.3.	Capa Modelo	55
3.3.4.	Núcleo Drupal.....	55
3.4.	Vista de procesos	55
3.5.	Vista de despliegue	56
3.5.1.	Diagrama de despliegue.	56



Plataforma VideoWeb

3.5.2.	Estación de captura.....	58
3.5.3.	Antena	58
3.5.4.	Cámara.....	58
3.5.5.	Reproductor de Sonido	58
3.5.6.	Descripción de elementos e interfaces de comunicación	58
3.6.	Vista de Implementación	59
3.6.1.	Vista	59
3.6.2.	Controlador.....	60
3.6.3.	Modelo.....	61
3.6.4.	Núcleo Drupal.....	62
3.7.	Vista de Datos	62
3.8.	Estructura del Equipo de Desarrollo	62
3.8.1.	Configuración de los puestos de trabajos por roles	63
3.9.	Conclusiones	63
Capítulo 4: Evaluando la Arquitectura de Software.		65
4.1.	¿Cuándo una Arquitectura puede ser evaluada?	65
4.1.1.	La evaluación temprana	65
4.1.2.	La evaluación tardía	65
4.2.	¿Qué resultados produce la evaluación de la Arquitectura de Software?	66



Plataforma VideoWeb

4.3.	¿Por cuales cualidades puede ser evaluada la Arquitectura de Software?	66
4.4.	Técnicas de evaluación	68
4.5.	Principales métodos de prueba de arquitectura de software	69
4.5.1.	SAAM	69
4.5.2.	ADR	70
4.5.3.	ATAM	70
4.5.4.	ARID	71
4.6.	Propuesta de Estrategia de evaluación de la Arquitectura.....	71
4.6.1.	Estrategia de evaluación temprana (ARID).....	71
4.6.2.	Estrategia de evaluación Tardía (ATAM).	75
4.7.	Conclusiones	76
	Conclusiones Generales	78
	Recomendaciones	80
	Bibliografía Citada.....	81
	Bibliografía Consultada.....	84
	Anexos	86

INTRODUCCIÓN

El video y el audio tienen gran poder para transmitir información a otras personas, ya sea educativa, comercial o simplemente para el entretenimiento. En los primeros años de internet era muy difícil lograr difundir información en estos formatos o de lograrse se hacía con muy mala calidad. Esto estaba dado por el poco ancho de banda con que contaban las redes en sus inicios y el gran tamaño que tienen los archivos de audio y video.

El desarrollo de internet conjuntamente con el aumento del ancho de banda de las redes de telecomunicación ha dado paso al auge de los servicios de video y audio por la Web. Existen dos tipos de servicios de videos asociados a esta tendencia: video en vivo (live-video) y video bajo demanda. Se pueden brindar servicios similar a la televisión, el usuario solo puede ver la información que se está transmitiendo en el instante que accede, esta forma es la llamada video en vivo. La otra alternativa, video bajo demanda, es similar a cuando usamos una videocasetera, permite avanzar, pausa, aumentar la velocidad, es decir, interactuar con el flujo de datos recibido. (1).

Los servicios de videos y audio por internet se pueden implementar de diversas formas, entre ellas se encuentran la Descarga (Download), que consiste en publicar los ficheros para ser descargados por el usuario y una vez completo el archivo poder reproducirlo. Otra vía es la Descarga progresiva (Progressive download), a medida que se está descargando la media se va reproduciendo (1). Pero frecuentemente se usa la tecnología de *Media Streaming* (MS) para lograrlo (2). Bajo el término MS se engloban una serie de productos y técnicas cuyo objetivo es la difusión de contenidos multimedia tales como audio y video. Este sistema de distribución se caracteriza por la visualización de los contenidos en el cliente sin la necesidad de esperar la descarga completa de un fichero (3). Al usar sistemas basados en la tecnología de MS se logra aprovechar de forma más eficiente el ancho de banda, se brinda un servicio más rápido y con mejor calidad con respecto al tamaño de video y velocidad de transmisión en comparación con las alternativas de descarga y descarga progresiva.

Como parte del auge de los servicios de videos y audio en internet proliferan los sitios web que se dedican parcial o completamente a los mismos (*Plataforma VideoWeb*). Entre los servicios más comunes que brindan se encuentra la difusión de video, ya sea seriales televisivos, películas, videos humorísticos, promocionales o simplemente la transmisión de canales de televisión en vivo entre otros. La difusión de audio es habitual en estos espacios también, se transmiten programas de radios, música, etc.

Debido a la gran demanda de estos servicios y la necesidad de una plataforma que logre brindarlos, se decidió implementar la Plataforma VideoWeb por parte del Polo de Video y Sonido Digital de la Facultad 9 de la Universidad de las Ciencias Informáticas, Cuba. Por lo cual, las principales tareas de presente trabajo se encaminan a dar solución al **problema** de la necesidad de una arquitectura de software robusta y confiable que permita el desarrollo de un sistema modificable, interoperable, seguro, funcional y disponible, que asegure el mantenimiento, flexibilidad y reusabilidad de la Plataforma VideoWeb, apoyándose en el uso de herramientas libres.

El **objeto de estudio** lo constituye el proceso de desarrollo de software de la Plataforma VideoWeb, siendo el **campo de acción** la arquitectura de software para la misma.

El principal **objetivo** es diseñar una arquitectura de software robusta y confiable para la Plataforma VideoWeb que permita que el sistema sea modificable, interoperable, seguro, funcional y disponible, asegurando el mantenimiento, flexibilidad y reusabilidad de la misma apoyándose en el uso de herramientas libres.

La **idea a defender** de la presente investigación es: El diseño de una arquitectura de software robusta y confiable, que cumpla con los requerimientos no funcionales y permita que la Plataforma VideoWeb sea modificable, interoperable, segura, funcional y disponible garantizará el mantenimiento, flexibilidad y reusabilidad del producto.

Las **tareas** trazadas para dar cumplimiento al objetivo general son:

1. Analizar diferentes sistemas similares a la Plataforma VideoWeb.
2. Valorar el estado del arte de varios estilos y patrones de la arquitectura de software y tomar posición en cuanto al más idóneo para este tipo de aplicación.
3. Valorar las diferentes propuestas arquitectónicas usadas en sistemas de similar propósito a partir de consultar la bibliografía especializada.
4. Analizar las herramientas y plataformas a utilizar para el desarrollo.
5. Plantear la propuesta arquitectónica para la Plataforma VideoWeb.
6. Especificar mediante el estudio de varias metodologías de prueba de arquitectura cuál es aplicable a la arquitectura definida.
7. Diseñar al menos 2 estrategias de prueba para la arquitectura de software seleccionada.

Para la realización de estas tareas se utilizaron diferentes métodos científicos de investigación.

Se emplearon Métodos Teóricos como el *Analítico-Sintético* para lograr un entendimiento más profundo de



Plataforma VideoWeb

los procesos relacionados con las plataformas que brindan servicios de video y audio por la web así como las tecnologías de video streaming y las tendencias de Arquitectura de software, patrones y estilos arquitectónicos.

Esto permitirá conocer mejor cuales son las funcionalidades principales que brindan estos servicios y a partir de ello proponer la arquitectura de software más conveniente usando los estilos y patrones adecuados para aplicaciones web de este tipo.

Otro método teórico empleado es el *Análisis-Histórico-Lógico* se utiliza para determinar el desarrollo y evolución de las tecnologías de media streaming y las plataformas que brindan servicios de video y audio por la web así como analizar las principales tendencias de la arquitectura de software, patrones y estilos arquitectónicos que se apliquen en el campo del desarrollo de aplicaciones Web. Entre los métodos empíricos utilizados se encuentra la *Entrevista*. Se realizaron dos entrevistas, la primera con el objetivo de conocer las principales tendencias de la arquitectura de software, patrones y estilos arquitectónicos; la segunda entrevista se hizo para comprender el funcionamiento de *Inter-nos*, que es una plataforma que brinda servicios de audio y video por la web dentro de la UCI.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

En este capítulo se realizará un análisis de los sistemas similares a la Plataforma VideoWeb, analizando sus características y funcionalidades. Se abordan temas relacionados con la arquitectura de software, estilos y patrones arquitectónicos, principales características y cual se ajusta a las necesidades de la Plataforma VideoWeb. Serán tratados los principales conceptos asociados al desarrollo del presente trabajo.

1.1. **Sistemas similares a la Plataforma VideoWeb**

Una plataforma Web es un sistema formado por un conjunto de componentes hardware y software que proporcionan capacidades (servicios) sobre las que se deberá apoyar cualquier aplicación software y cuyo funcionamiento es a través de internet o la Web (4).

El resultado final debe ser un conjunto (plataforma) formado por componentes horizontales o herramientas de colaboración (gestor de usuarios, gestores de contenido y gestores documentales) que puedan ser reutilizado en las distintas aplicaciones específicas que se introduzcan (5).

Podemos considerar entonces la Plataforma VideoWeb como una plataforma Web, ya que se conforma con el objetivo funcional principal de brindar servicios de audio y video por la web.

Los servicios de audio y video a través web son cada vez más frecuentes, diversas instituciones, empresas o personas lo utilizan con fines educativos, económicos, de publicidad entre otras. Existen gran cantidad de sitios dedicados a brindar estos servicios, entre los principales y más conocidos se encuentran TV.com, es.youtube.com, VideoWebPeru, de carácter internacional y en la Universidad de las Ciencias Informáticas el sitio Inter-nos.

TV.com dedica varios espacios a la promoción, divulgación e información de seriales televisivos, películas y actores. Brinda la posibilidad de reproducir videos desde la propia página mediante un reproductor flash¹. Contiene sesiones para fotos relativas al mundo del cine y la televisión, bibliografías de actores, datos de los seriales que promocionan, dando a conocer los capítulos, sinopsis, actores entre otros datos.

¹ Reproductor cuya interfaz está implementada usando tecnología flash.



En su foro se tratan diversos temas de televisión. Cuenta con un diseño cargado de imágenes y muy vistoso, bien estructurado en sesiones y buscador incluido para la búsqueda de información.

El sitio **es.youtube.com** contiene gran cantidad de videos publicados de diversas índoles, estructurados por categorías, brindando la posibilidad de verlos mediante un reproductor flash. Implementado sobre herramientas libre ha logrado gran eficiencia en sus servicios, para ello utiliza herramientas como el servidor web Apache, lenguaje de programación Python, sistema operativo GNU/Linux (Suse), gestor de Base de Datos MySQL, para transmitir los vídeos no utiliza Apache sino un servidor web mucho más liviano como es Lighttpd y los videos para celulares lo transmite usando Darwin Streaming Server. Permite subir videos y compartirlos para que otros puedan acceder al mismo, esta funcionalidad es clave en su popularidad. Cada video tiene un ranking de calidad, donde los usuarios pueden votar y valorar el video según su punto de vista, esto permite tener una visión de cuan popular o no es. Contiene un buscador de videos, lo que facilita encontrar el deseado, pues a pesar de estar categorizados, son muchos y cada día se añaden nuevos, siendo difícil de otra forma encontrar el deseado. El diseño del sitio es sencillo pero agradable, no cuenta con gran cantidad de imágenes o animaciones extras y es de fácil navegación.

Video Web Perú integra servicios de videos y audios por la web. Con un diseño agradable y funcionalidades diversas, es un sitio web bastante completo, en lo respectivo al tema de los servicios de audio y video que más se solicitan en internet. Es posible reproducir tanto videos como audio desde un reproductor flash en el navegador embebido. Los videos están separados en categorías bien organizadas, cuentan con una breve información de cada material y el ranking de popularidad, el cual, los usuarios registrados pueden votar. En la sesión de radio es posible conformar una lista de reproducción con los temas que brinda el sitio, luego se puede reproducir la lista seleccionada. Posee un buscador solo para videos, la búsqueda puede ser ordenada por varias categorías (relevancia, fecha, vistas, Ranking). Brinda la posibilidad de subir videos a los usuarios registrados, aumentando las funcionalidades del sitio. Como complemento contiene una sesión para noticias de diversos géneros, desde políticas hasta informativas de eventos del sitio.

Inter-nos integra servicios de audio y video por la web usando como reproductor embebido al Reproductor de Windows Media. Su principal función es el contribuir con el proceso docente educativo en la Universidad de Ciencias Informáticas, publicando teleclases, materiales educativos, políticos y de apoyo a la docencia. De forma complementaria publica películas, seriales televisivos, documentales, programas de radio, festivales de artistas aficionados, la transmisión de canales de televisión y radio en vivo. Usa



Plataforma VideoWeb

tecnología de streaming para la transmisión de los archivos multimedia. Para este fin se emplea Windows Media Server, tecnología de Microsoft para proporcionar el servidor de streaming, el formato de video que utiliza es Windows Media Video (WMV). Está implementado usando el lenguaje de programación ASP, como servidor web Microsoft Internet Information Services (IIS), Windows Media Server como servidor de Streaming y el sistema gestor de base de datos Microsoft SQL Server 2000. El proceso de publicación de media se realiza de forma manual, desde el sitio se habilita el vínculo, pero se debe copiar el fichero a la carpeta indicada para que el servidor logre conformar el streaming. Las modificaciones al sitio se realizan directamente al código, por lo que suele ser un proceso complicado, pues no cuenta con un sistema de gestión de informaciones. El diseño de la página es sencillo, sin muchas imágenes o animaciones por lo que es de rápida navegabilidad. Cuenta con gran audiencia, y en ocasiones se ralentiza la transmisión del streaming por esta causa.

A continuación se presenta una tabla resumen de las características principales de las plataformas presentadas anteriormente.

Plataformas	Servicios de Videos	Servicios de Audio	Otros
TV.com	<ul style="list-style-type: none"> • Buscador • Reproductor embebido(Flash) • Informaciones adicionales de los videos 		<ul style="list-style-type: none"> • Noticias • Fotos • Fóruns • Encuestas
es.youtube.com	<ul style="list-style-type: none"> • Buscador • Reproductor embebido 		<ul style="list-style-type: none"> • Subir Videos • Implementado con herramientas y tecnologías libres

Video Web Perú	<ul style="list-style-type: none"> • Buscador • Reproductor embebido • Transmisión de Canales 	<ul style="list-style-type: none"> • Buscador • Reproductor embebido 	<ul style="list-style-type: none"> • Noticias • Subir Videos • Implementado con herramientas y tecnologías de software propietario.
Inter-nos	<ul style="list-style-type: none"> • Reproductor embebido • Transmisión de Canales 	<ul style="list-style-type: none"> • Reproductor embebido • Transmisión en vivo de radio 	<ul style="list-style-type: none"> • Soporte Técnico • Implementado con herramientas y tecnologías de software propietario.

Figura 1: Comparación entre plataformas

1.2. Arquitectura de Software

Las distintas organizaciones e instituciones requieren cada vez más de productos de mediana y gran envergadura, los cuales necesitan diferentes tecnologías y plataformas de hardware y software para alcanzar el funcionamiento deseado. La calidad de estos productos es indispensable, por lo que los sistemas deben ser modificables, interoperables, seguros, funcionales y disponibles, asegurando el mantenimiento, flexibilidad y reusabilidad. Para lograr estos requisitos es necesario el diseño de una arquitectura de software robusta y confiable que se adapte a las necesidades de cada software.

1.3. Definición de Arquitectura de Software

La definición de Arquitectura de Software (AS) no es única, existen varias y muchas veces polémicas.

Ninguna de ellas es totalmente respaldada por los arquitectos; pero existen algunas que han sido más reconocidas. Para comprender mejor este tema es necesario conocer el criterio de algunos de los principales concededores del mismo.

Según Paul Clements (6): La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.

El libro *Software Architecture in Practice, Second Edition* (7) expresa: la arquitectura de software de un programa o sistema informático es la estructura o las estructuras del sistema, que incluyen elementos de software, las propiedades externamente visibles de esos elementos y las relaciones entre ellos.

Abowd expresa (8): La arquitectura de software es un nivel importante de descripción para los sistemas. En este nivel de abstracción, las decisiones claves incluyen componentes de alto nivel, protocolos o interacciones entre componentes, propiedades globales de sistemas y de ciclo de vida.

Como se puede apreciar estos conceptos tienen algunos puntos en común pero no unifican una única idea. Cada autor expresa su criterio y brinda su propia definición. Simplemente, la AS es parte fundamental en el desarrollo de las aplicaciones y define cuales son las principales características del software, con cierto nivel de abstracción.

Ante la variedad de definiciones existentes de Arquitectura de Software, Mary Shaw y David Garlan (9) presentan una serie de modelos:

1. *Modelos estructurales*: Sostienen que la AS está compuesta por componentes, conexiones entre ellos y otros aspectos tales como configuración, estilo, restricciones, semántica, análisis, propiedades, racionalizaciones, requerimientos, necesidades de los participantes. El trabajo en esta área está caracterizado por el desarrollo de lenguajes de descripción arquitectónica (ADLs).
2. *Modelos de framework*: Son similares a la vista estructural, pero su énfasis primario radica en la (usualmente una sola) estructura coherente del sistema completo, en vez de concentrarse en su composición. Los modelos de framework a menudo se refieren a dominios o clases de problemas



específicos. El trabajo que ejemplifica esta variante incluye arquitecturas de software específicas de dominios, como CORBA², o modelos basados en CORBA, o repositorios de componentes específicos, como PRISM.

3. *Modelos dinámicos*: Enfatizan la cualidad conductual de los sistemas. “Dinámico” puede referirse a los cambios en la configuración del sistema, o a la dinámica involucrada en el progreso de la computación, tales como valores cambiantes de datos.
4. *Modelos de proceso*: Se concentran en la construcción de la arquitectura, y en los pasos o procesos involucrados en esa construcción. En esta perspectiva, la arquitectura es el resultado de seguir un argumento (script) de proceso. Esta vista se ejemplifica con el actual trabajo sobre programación de procesos para derivar arquitecturas.
5. *Modelos funcionales*: Una minoría considera la arquitectura como un conjunto de componentes funcionales, organizados en capas que proporcionan servicios hacia arriba. Es tal vez útil pensar en esta visión como un framework particular.

Estos modelos lejos de excluir las otras definiciones de AS representan distintas formas en que pueden aplicarse y agruparse.

Los elementos anteriormente expuestos permiten concluir que la arquitectura de software representa los principales componentes que posibilita el funcionamiento de las aplicaciones. Incluye las descripciones de estos componentes, lo que permite comprender el objetivo y la funcionalidad de cada uno de ellos. La AS es clave en el desarrollo de los sistemas informáticos, constituye un basamento y guía para alcanzar el éxito.

1.4. Estilos y Patrones Arquitectónicos

Dentro del marco de la arquitectura de software es muy frecuente escuchar los términos de estilos y patrones arquitectónicos. A continuación se brindan definiciones, puntos de vistas y característica relacionados a los mismos.

² Common Object Request Broker Architecture (CORBA) es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

1.4.1. Estilo Arquitectónico

Se definen estilo arquitectónico como una familia de sistemas de software en términos de su organización estructural. Expresa componentes y las relaciones entre estos, con las restricciones de su aplicación y la composición asociada, así como también las reglas para su construcción (10).

Según Garlan y Shaw (11) un estilo arquitectónico determina el vocabulario de los componentes y conectores que pueden ser utilizados en instancias de los mismos, junto con una serie de limitaciones sobre la forma en que pueden ser combinados. Estos pueden incluir limitaciones topológica y descripciones de la arquitectura (por ejemplo, no hay ciclos). Otras restricciones, por ejemplo, las relacionadas con la ejecución semántica, también puede ser parte de la definición de estilo.

Para definir cuáles son los estilos arquitectónicos existentes existen varios puntos de vistas de diferentes autores, incluso, existen autores que han variado sus definiciones anteriores. A continuación se presenta una de las más conocidas.

En Software Architecture in Practice (7), se proporciona cinco grupos de estilo:

1. Flujo de datos (El sistema es visto como una serie de transformaciones sobre piezas sucesivas de datos de entrada. El dato ingresa en el sistema, y fluye entre los componentes, de uno en uno, hasta que se le asigne un destino final).
 - Proceso secuencial por lotes.
 - Red de flujo de datos.
 - Tubería-filtros.
2. Llamado y retorno (estilo dominado por orden de computación, usualmente con un solo hilo de control, un programa principal controla varios subprogramas que se comunican mediante el uso de llamadas).
 - Programa principal / Subrutinas
 - Tipos de dato abstracto
 - Objetos
 - Cliente-servidor basado en llamadas
 - Sistemas en capas
 - Modelo Vista Controlador
3. Componentes independientes (dominado por patrones de comunicación entre procesos independientes, casi siempre concurrentes).

- Sistemas orientados por eventos
 - Procesos de comunicación
4. Centrados en datos (Se caracteriza por un almacenamiento central complejo, manipulado por sistemas independientes).
- Repositorio
 - Pizarra
5. Máquina virtual (caracterizado por la traducción de una instrucción en otra, Simulando funcionalidades que no son nativa al hardware o software sobre el que está implementado).
- Intérprete
 - Sistema basado en regla.

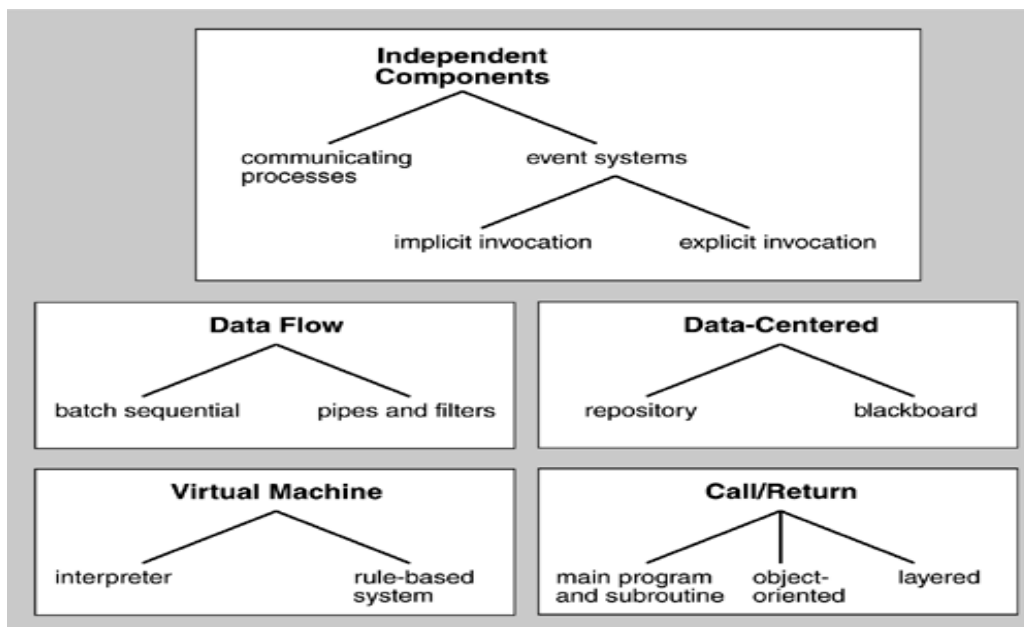


Figura 2: Un pequeño catálogo de modelos arquitectónicos organizado por las relaciones es-un.

1.4.2. Patrones Arquitectónicos

Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Un sistema bien estructurado está lleno de patrones (12).

Cada patrón describe cierto problema que ocurre frecuentemente, describe el núcleo de la solución al problema, de tal manera que se puede usar en varias ocasiones sin necesidad de hacer lo mismo (12).

Elementos que conforman un patrón (12).

- *Nombre del patrón:* nombre estándar del patrón por el cual será reconocido en la comunidad.
- *Clasificación del patrón:* creacional, estructural o de comportamiento.
- *Intención:* ¿Qué problema resuelve el patrón?
- *También conocido como:* Otros nombres de uso común para el patrón.
- *Motivación:* Escenario de ejemplo para la aplicación del patrón.
- *Aplicabilidad:* Criterios de aplicabilidad del patrón.
- *Estructura:* Diagramas de clases oportunos para describir las clases que intervienen en el patrón.
- *Participantes:* Enumeración y descripción de las entidades abstractas (y sus roles) que participan en el patrón.
- *Colaboraciones:* Explicación de las interrelaciones que se dan entre los participantes.
- *Consecuencias:* Consecuencias positivas y negativas en el diseño derivadas de la aplicación del patrón.
- *Implementación:* Técnicas o comentarios oportunos de cara a la implementación del patrón.
- *Código de ejemplo:* Código fuente ejemplo de implementación del patrón.
- *Patrones relacionados:* Referencias cruzadas con otros patrones.

Los patrones se pueden clasificar según la escala o nivel de abstracción de las siguientes formas:

- *Patrones arquitecturales:* Aquellos que expresan un esquema organizativo estructural fundamental para sistemas software.
- *Patrones de diseño:* Aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software.
- *Idiomas:* Patrones de bajo nivel específicos para un lenguaje de programación o entorno concreto.

Entre los patrones arquitectónicos más comunes se encuentran:

1.4.2.1. **Arquitectura en tres capas**

La arquitectura en tres capas consiste en aislar la lógica de la aplicación y en convertirla en una capa intermedia bien definida. En la capa de presentación se realiza relativamente poco procesamiento de la aplicación; las ventanas envían a la capa intermedia peticiones de trabajo. Y este se comunica con la capa

de almacenamiento del extremo posterior (13).

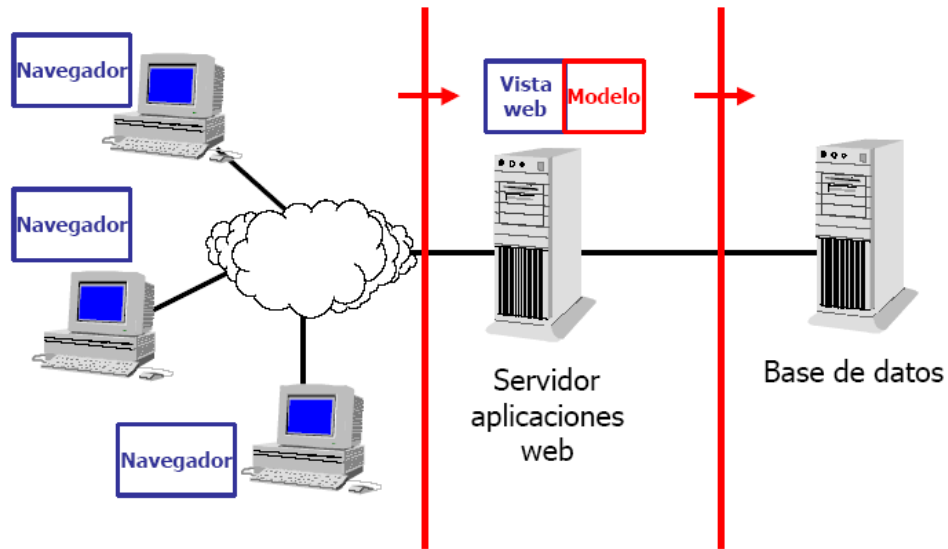


Figura 3: Arquitectura en Tres Capa

1.4.2.2. Modelo Vista Controlador (MVC):

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario que puede invocar cambios en el modelo y probablemente en la vista (12).

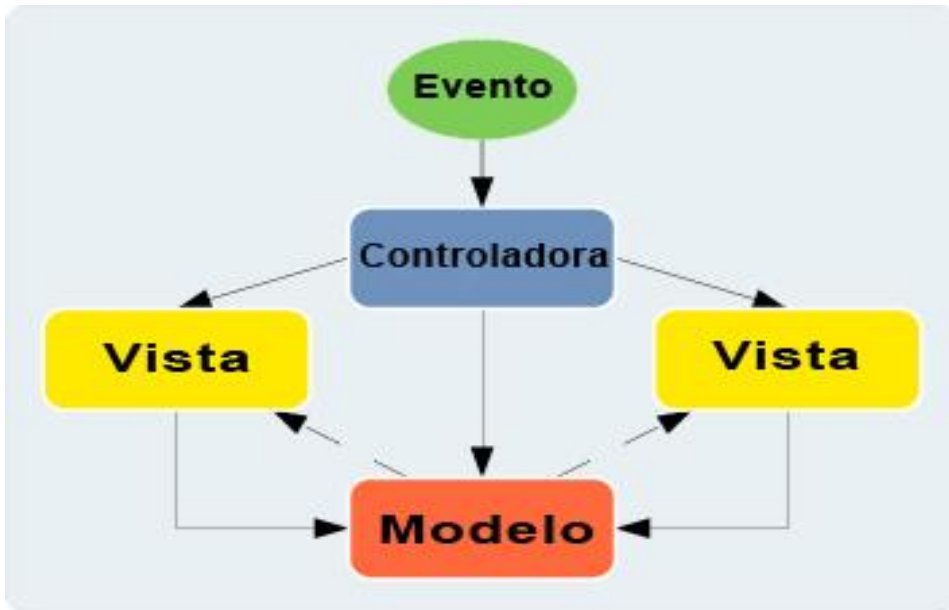


Figura 4: Modelo Vista Controlador.

1.4.2.3. Arquitectura Orientada a Servicios (SOA):

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular (14).

La arquitectura SOA permite simplificar las relaciones entre distintos sistemas, optimizar su funcionamiento, facilitar la incorporación de nuevos elementos en la arquitectura e incluso cambiar los existentes de una manera sencilla (15).

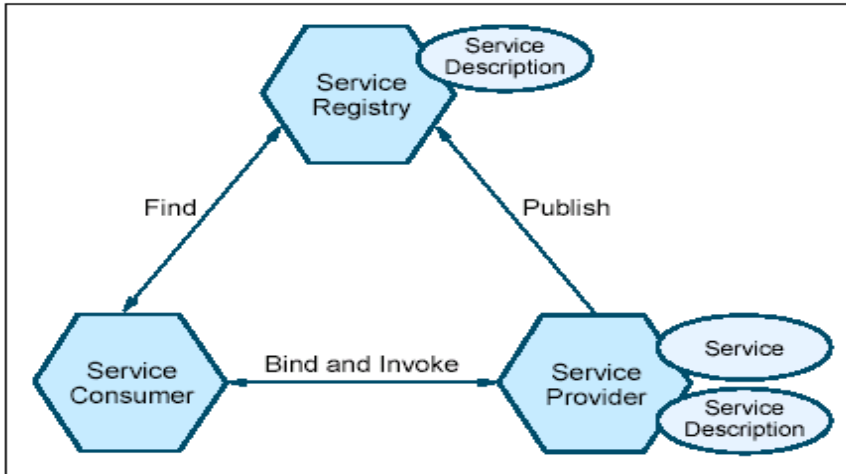


Figura 5: Arquitectura Orientada a Servicios.

1.4.2.4. Arquitecturas multicapas orientadas a objetos

Una arquitectura multicapas que se adecue a los sistemas de información orientados a objetos, incluye la división de las responsabilidades que encontramos en la arquitectura clásica de tres capas. Las responsabilidades se asignan a los objetos de software (13).

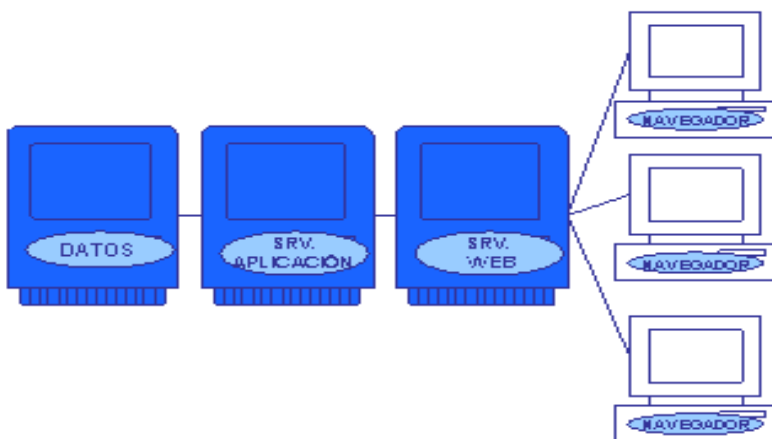


Figura 6: Arquitectura multicapas orientadas a objetos

Como se puede apreciar existe gran relación entre los términos de estilo y patrón arquitectónico pero la principal diferencia está en el nivel de abstracción. Los estilos presentan un nivel de abstracción muy alto, sin especificar detalles, mientras que los patrones se adaptan a problemas específicos, dando una

solución obtenida por la experiencia en el tema.

1.5. Tecnologías para el Desarrollo de Aplicaciones

Existen disímiles formas para desarrollar una aplicación web. Se puede realizar desarrollando todos los componentes necesarios sin utilizar código existente, siguiendo estándares o simplemente de forma libre. Esto trae como consecuencia una gran demora, particularmente, en sistemas de gran envergadura, aunque posibilita adaptar las funcionalidades según las preferencias de los desarrolladores. Otra forma es el empleo de herramientas como Frameworks o CMSs, los que posibilitan una base de partida, pues tienen por lo general una estructura sólida, que agiliza el proceso de construcción y permiten desarrollar aplicaciones complejas en un tiempo relativamente más rápido que cuando se realiza desde cero. En numerosas aplicaciones es común encontrarse servicios y funcionalidades como el registro de usuarios, sesiones para brindar informaciones, crear blogs o la manipulación de roles para controlar el acceso, las cuales ya están implementadas o es fácil su implementación en Frameworks o CMSs.

1.5.1. CMS

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.

Con frecuencia se emplean los CMS en la web debido a las ventajas que estos brindan. Por lo general son de fácil y rápida configuración, de forma nativa permiten crear sitios web, blog etc. El tiempo que tomaría realizar una aplicación con características similares de forma manual sería muy grande. Para su instalación deben estar copiados en el servidor web, luego solicitan los datos necesarios para su correcto funcionamiento como nombre del sitio, los datos para configurar la conexión con la base de datos entre otros dependiendo del CMS. Una vez instalado permiten la gestión de información mediante una interfaz que brinda abstracción de cómo está almacenada, permitiendo interactuar con ella sin tener un alto nivel de conocimiento de programación. La gestión de los datos incluye la forma en que estos se mostraran a los usuarios, posibilitando cambiar la interfaz sin alterar la información y viceversa.

Los CMS dividen la aplicación en dos partes fundamentales, la administrativa y la de usuario. En la parte

administrativa se realiza la configuración del sitio web, administración de usuarios y sus niveles y la forma en que se mostrara la información conjuntamente con la mayor parte de la gestión del contenido. En la sección correspondiente al usuario se puede realizar algunas modificaciones a los contenidos, pero su función fundamental es la presentación de la información.

1.5.1.1. Principales CMSs

EL empleo de CMS en portales web es cada vez más frecuente, existe una gran cantidad de ellos, cada cual con sus ventajas y desventajas. Para el estudio del presente trabajo se seleccionaron por su calidad el Joomla! 1.5.9, Drupal 6.9 y Plone 3.2.

A continuación una tabla comparativa ente estos CMSs, abordando aspectos importantes de software, hardware y principales funcionalidades para la gestión de contenido.

Características	Drupal	Joomla!	Plone
Servidor de Aplicación	PHP 4.3.5+	Cualquiera que soporte PHP (Recomendado Apache)	Zope
Base de datos	MySQL, PostgreSQL	MySQL	Zope
Licencia	GNU/GPL	GNU/GPL v2	GNU/GPL
Lenguaje de programación	PHP	PHP	Python
Servidor Web	Apache, Microsoft Internet Information Services (IIS),	Apache	Apache, Nginx, Microsoft IIS
Privilegio	Si	No	Si



Plataforma VideoWeb

granular³			
Varios mecanismos de autenticación	Si	Si	Si
Autenticación con SSL	No	Si	Complemento libre
Comunidad de desarrollo	Si	Si	Si
Framework de prueba	Complemento libre	Si	Si
URL amistosas	Si	Si	Si
Editor WYSIWYG⁴	Complemento libre	Si	Si
Soporte de varios lenguajes	Si	Si	Si
Temas/Plantillas	Si	Si	Si

³ El sistema permite gestión de privilegios página por página o elemento por elemento y separar los privilegios para otras funcionalidades

⁴ Lo que ves es lo que obtienes por sus siglas en ingles. Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final.

Estadísticas web	Si	Si	Complemento libre
Soporte FTP	Limitado	Si	Si

Figura 7: Comparación entre Drupal, Joomla! y Plone.

Como se puede apreciar existe gran paridad entre estos CMSs. Plone es una solución bastante completa y muy potente. Existen varios inconvenientes para el empleo del CMS Plone, entre ellos podemos significar la complejidad en su configuración y administración. La curva de aprendizaje para la administración y configuración es muy elevada, siendo complejo adaptar las funcionalidades a las necesidades de la Plataforma VideoWeb. Lo cual provocaría invertir gran cantidad de tiempo, atentando contra la rapidez del proceso de desarrollo.

Drupal y Joomla! están desarrollados 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. Las funcionalidades que brindan estos CMSs son similares; aunque Drupal además de soportar MySQL como Sistema Gestor de Base de Datos, también permite el uso de PostgreSQL. Unido a esto cuenta con gestión de privilegios granulares, lo que posibilita una mayor especificación del acceso a los recursos. Existen comunidades internacionales que brindan soporte 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. Su administración posibilita configurar y adaptar el sistema de forma detallada. Destacar que en la Universidad de las Ciencias Informáticas, existen diferentes proyectos que desarrollan portales usando el CMS Drupal, lo que posibilita el apoyo e intercambio de experiencias, facilitando la preparación del equipo de desarrollo.

Joomla! cuenta con gran aceptación a nivel mundial y funcionalmente es tan competitivo como los anteriores. Está enfocado en facilitar las funcionalidades de administración, contando con una estructura solida y bien definida. La sesión administrativa permite adaptarse de forma rápida al estilo de presentación y organización de la información. El soporte FTP aumenta la rapidez y seguridad de la gestión de archivos. Nuevas funcionalidades pueden ser agregadas a través de las extensiones o complementos, los

cuales se adicionan instalándolos (desde una carpeta o fichero compactado en Zip). Si se requiere instalar una gran cantidad de complementos pudiera resultar molesto, pero no requiere que el encargado de esta tarea tenga permisos físicamente en el directorio donde se encuentra la aplicación, pues Joomla! se encarga de copiar los ficheros a el lugar indicado.

Los complementos se implementan usando el patrón arquitectónico Modelo-Vista-Controlador, lo cual permite separar la lógica de la presentación y el acceso a dato, aunque se pueden realizar complementos sin usar este patrón. La comunidad de Joomla! es muy fuerte y se encuentra desarrollando y brindando ayuda constantemente, donde una parte importante de ella es de habla hispana. La seguridad, que es un aspecto fundamental, es muy buena, cada vez que se detectan errores a las nuevas versiones, la comunidad se encarga de corregirlos y de lanzar las actualizaciones. Los desarrolladores de la Plataforma VideoWeb están familiarizados con la estructura y funcionamiento del Joomla!, además de tener los conocimientos para poder desarrollar los complementos necesarias para el funcionamiento del la plataforma.

Los CMSs son factibles emplearlos en sistemas que sus funcionalidades principales se centren en la gestión de información para los cuales está diseñado el CMS. Por el contrario, no es factible cuando la información que manipula el sistema no se ajusta a las características del mismo, aunque la mayoría de estos permiten adicionarles funcionalidades para adaptarlos a diversas necesidades. Cada CMS tiene formas específicas para lograrlo, como es el caso de Joomla! que se le pueden instalar módulos, componentes o plug-ins, ó Drupal que lo logra mediante módulos. En cada caso es posible ampliar las posibilidades de funcionamiento, pero si el sistema que se desea lograr es de gran magnitud, requiere de un procesamiento complejo y con gran cantidad de interacción con bases de datos no es recomendable emplear CMS.

1.5.2. Framework

El término framework es muy utilizado en el campo de la informática, normalmente tratado como framework de desarrollo, que no es más que un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación, proporciona una estructura al código fuente, forzando al desarrollador a crear código legible y más fácil de mantener (16).

Para el desarrollo de aplicaciones web existe una serie de frameworks con una estructura de clases bien definida incluyendo las de interacción con los sistemas gestores de bases de datos y las clases

específicas para el desarrollo de la lógica de la aplicación. Para ayudar a los desarrolladores incluyen soportes a distintos lenguajes script, librerías, paquetes de clases que automatizan la interacción con base de datos y simplifica la creación de funciones complejas. Esto permite dedicar más tiempo a desarrollar la lógica de la aplicación sin necesidad de comenzar por los pasos básicos nativos a cada programa para asegurar las funcionalidades.

Entre los inconvenientes para el uso de los frameworks se encuentra el tiempo que requiere la preparación para lograr entender el funcionamiento y la estructura del mismo, restando tiempo para el diseño y desarrollo del producto. Añadido a esto, en ocasiones incluye código innecesario, principalmente para aplicaciones pequeñas. Por otro lado es factible el empleo de framework como base para el desarrollo de sistemas complejos en cuanto a funcionalidades e interacción con bases de datos.

1.6. 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 que serán enviadas de respuesta al cliente, pueden acceder a bases de datos, recursos en la red y ficheros en el servidor.

Estos lenguajes son imperceptibles por el cliente, cuando se realiza la petición al servidor este convierte todo el código a que el navegador lo interprete. Entre los principales lenguajes del lado del servidor se encuentran PHP, Java Server Pages (JSP) y Python.

1.6.1.PHP

PHP acrónimo recursivo⁵ para "PHP Hypertext Pre-processor" diseñado originalmente para la creación de páginas dinámicas. Es un lenguaje script del lado del servidor.

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 conectarse a varios sistemas de bases de datos brindando múltiples funcionalidades, destacando en este aspecto la conectividad con PostgreSQL y

⁵ Si la primera letra del mismo hace referencia al propio acrónimo.

MySQL que son gestores muy utilizados en el desarrollo de aplicaciones web, destacando que los desarrolladores están preparados en este lenguaje. 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.

1.6.2. Java Server Pages (JSP)

JSP es un acrónimo de Java Server Pages, tecnología para crear aplicaciones web. Fue desarrollada por la compañía Sun Microsystems, y su funcionamiento se basa en scripts, que utilizan una variante del lenguaje Java. Permite a los programadores generar contenido dinámico para web, en forma de documentos HTML y e incrustar en el contenido estático de las páginas Web código Java y comando predefinidos. Con JSP se pueden crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas.

1.6.3. Python

Python se utiliza como lenguaje de programación interpretado, permite dividir el programa en módulos reutilizables desde otros programas escritos en Python. Permite varios estilos de programación como programación orientada a objetos, programación estructurada y programación funcional. Usa tipo de dato dinámico, resolución dinámica de nombres, lo que enlaza un método y un nombre de variable durante la ejecución del programa. Python fue diseñado para ser leído con facilidad, en vez de delimitar los bloques de código mediante el uso de llaves utiliza la indentación⁶. En Python, todo es un objeto y además soporta herencia múltiple y polimorfismo.

1.7. Sistema Gestor de Base de Datos

Los Sistema Gestor de Base de Datos son un tipo de software dedicado a servir de interfaz entre la base

⁶ Significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores para separarlo del texto adyacente.

de datos, el usuario y las aplicaciones que la utilizan. Entre los principales gestores bajo licencias libres se encuentran PostgreSQL y MySQL.

1.7.1. PostgreSQL

PostgreSQL un sistema gestor de base de datos objeto-relacional (ORDBMS) de código abierto. 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 podremos 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.

Aunque su velocidad de respuestas en base de datos pequeñas es relativamente lenta mantiene la misma en las de gran tamaño, siendo estable y robusta ante grandes peticiones.

1.7.2. MySQL

MySQL es un sistema de gestión de base de datos muy potente y popular. Usado tanto por empresas pequeñas como de gran envergadura, presenta un esquema de licenciamiento dual. Por una parte se ofrece bajo la licencia GNU/GPL para cualquier uso compatible con esta licencia y además permite adquirir la versión privativa, por lo cual la compañía ofrece soporte y servicios.

MySQL es un sistema de gestión de base de datos relacional, multihilo, multiusuario y multiplataforma. Se caracteriza por:

- Aprovechar la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soportar gran cantidad de tipos de datos para las columnas.
- Disponer de API's en gran cantidad de lenguajes (C, Java, ODBC, .NET, PHP, Perl, Python, etc.).
- Disponibles en múltiples plataformas como Linux, Windows, y Mac, ambos de 32 y 64 bit.
- Soporta hasta 32 índices por tabla.

El SGBD MySQL es robusto y seguro, cuenta con modelo de protección de datos, que incluye autenticación granular, permisos de objeto, cifrado y descifrado de datos incorporado, conexión SSL y servidor de copias de seguridad flexible con punto de recuperación en el tiempo. También cuenta con alta disponibilidad que incluyen opciones de replicación, soporte para clúster de bases de datos en medios compartidos y apoyo de terceros para obtener otras funcionalidades con clúster. MySQL es hoy un gestor potente y probado a nivel mundial, constituye una opción de calidad para el manejo de los datos necesarios para la Plataforma VideoWeb. Señalar que Cuba tiene denegada la descarga directa de MySQL por problemas del bloqueo de Estados Unidos, lo cual constituye una barrera para su uso y explotación.

1.8. Media Streaming (Streaming de audio y video).

Media Streaming (MS) es una tecnología para transmitir video o audio por la web, de forma que optimiza la reproducción y el aprovechamiento del ancho de banda (2). Permite la reproducción de archivos multimedia sin la necesidad de copiarlo completamente, debido al gran tamaño que caracteriza a los mismos, la tecnología Streaming se ajusta perfectamente a las necesidades de la web.

La forma de funcionamiento de la MS es la siguiente. Cuando se realiza una petición de algún fichero al Servidor de Streaming, este comienza a enviar el fichero al cliente que lo solicitó. El flujo es entregado directamente de la fuente al reproductor en tiempo real mediante un proceso continuo, solo almacenando una pequeña porción del streaming en el buffer del reproductor (el tamaño del buffer depende del reproductor), del cual se reproduce la media, a medida que se reproduce del buffer se elimina esa información y se copia la siguiente. De esta manera si varía la velocidad de copia de la red, o se pierde la conexión con el servidor, se utiliza la información del buffer para continuar reproduciendo (Mientras mayor sea el tamaño del buffer, mayor será el tiempo que se pueda estar sin conexión o con poca velocidad de

Plataforma VideoWeb

conexión con el servidor) (1). Esta tecnología posibilita que no sea necesario descargar completamente los archivos multimedia antes de comenzar la reproducción, lo cual tomaría una cantidad de tiempo relativamente grande, pues los archivos multimedia son de gran tamaño, por ejemplo, un fichero codificado en MPG2 de 1 horas de duración, requiere aproximadamente 1.8 gigabytes (GB).

1.8.1. La arquitectura del streaming.

La arquitectura del Streaming está compuesta por cuatro partes fundamentales:

- Captura y codificación
- Almacenamiento y Servicio
- Distribución y entrega
- Reproducción



Figura 8: Arquitectura del Streaming

En la primera etapa de la conformación del streaming se realiza la captura y codificación mediante algún dispositivo especializado para esto (dispositivo de captura). El audio y el video se digitalizan mediante micrófonos, cámaras u otros medios dedicados a estos fines. En el momento de captura se codifican y almacenan los ficheros en un formato soportado por el servidor para realizar el streaming (1).

Los canales de distribución conectan a los clientes con el servidor. La red puede ser desde local hasta

internet. La distribución del streaming generalmente puede usar administración de derechos digitales (Digital rights management (DRM)) para proteger la propiedad intelectual (1).

El reproductor recibe el streaming y lo descomprime en formato de audio y video tradicional, donde puede ser mostrado en la PC.

1.8.2. Streaming en vivo y bajo demanda.

El streaming puede ser usado en dos formas principales de publicar el video, *video en vivo* (live-video) y *video bajo demanda* (video-on-demand). El servicio de video en vivo envía la información del video mientras el usuario esté conectado, puede estar grabada previamente o ser capturada y difundirla en vivo, en cualquiera de los casos, solo puede ser vista una vez, durante el momento de transmisión. Solo es posible un tipo de interacción, la pausa, su comportamiento es similar al de la televisión tradicional. Por otra parte, los principios del video bajo demanda son totalmente diferentes. El usuario realiza la petición del material deseado en cualquier momento y el servidor lo proporciona exclusivamente. Este sistema permite al usuario interactuar con la información de forma más activa: pausa, retroceder, avanzar, saltos, velocidad de reproducción, etc. Es un comportamiento similar a la de un videotape (17).

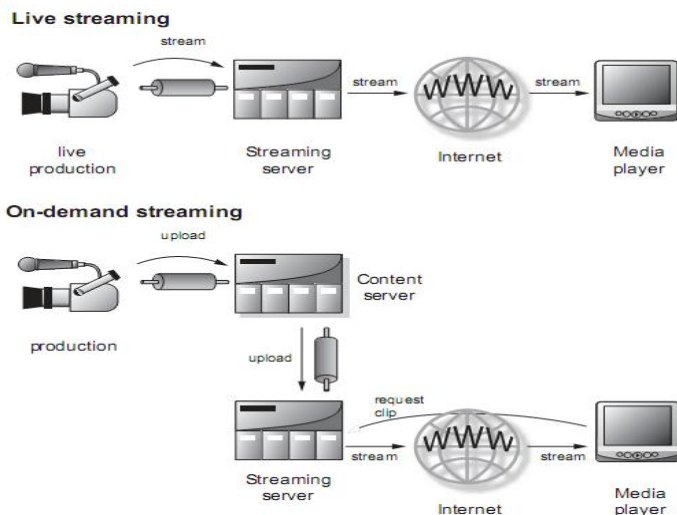


Figura 11: Streaming en vivo y bajo demanda

1.8.3. Principales Servidores Streaming

Existen varios servidores streaming bajo licencias libres que pueden usarse en la plataforma, pero

presentan actualmente algunas deficiencias que hacen que no ajusten a todas las necesidades de la Plataforma VideoWeb. Entre los principales servidores streaming libres están: FFmpeg y FFServer, Darwin Streaming Server (DSS), VideoLan (VLC), Feng, MPEG4IP y MEC4.

1.8.4. FFmpeg y FFServer

El **FFmpeg** es un servidor streaming de audio/video para Linux apropiado para transmisión en vivo. Soporta compresión de audio/video en tiempo real a diferentes formatos (MPEG, mjpeg, real, flash, asf) y diferentes bitrates.

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

El proyecto está compuesto por varios componentes:

- **Ffmpeg**: una herramienta de línea de comandos para convertir archivos multimedia entre distintos formatos.
- **Ffserver**: un servidor streaming de multimedia para transmisiones en vivo.
- **Ffplay**: un reproductor de multimedia simple basado en SDL (Simple Directmedia Layer, librería de bajo nivel para el acceso al audio, teclado, mouse, tarjetas gráficas, etc.) y las librerías FFmpeg.
- **Libavcodec**: es una librería contenedora de codificadores y decodificadores de audio y video.
- **Libavformat**: es una librería contenedora de multiplexadores y demultiplexadores para formatos contenedores de multimedia.

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

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

El FFmpeg es una excelente herramienta para la conversión de formatos de video desde la línea de comandos o consola, pero a la hora de transmitir contenidos por la red se complica el proceso ya que la configuración del servidor así como el proceso de agregarle nuevos ficheros de entrada se realiza mediante la edición de un fichero de configuración llamado ffserver.conf. Los archivos o flujos de entrada son llamados semillas (feeds) y cada uno debe ser especificado en una sección <Feed> en el archivo de

configuración. Lo cual hace este trabajo engorroso y complejo de automatizar.

1.8.5. Darwin Streaming Server

Es la versión de código abierto de la tecnología del Apple's QuickTime Streaming Server que permite enviar un streaming de multimedia a clientes a través de internet utilizando los protocolos RTP y RTSP. Basado en el código base del QuickTime Streaming Server, el Darwin Streaming Server proporciona un alto nivel de personalización y funciona en una gran variedad de plataformas permitiendo manipular el código fuente para satisfacer las necesidades de los desarrolladores. Darwin Streaming Server es un proyecto de código abierto pensado para desarrolladores que necesiten hacer streaming de archivos QuickTime o MPEG-4 en plataformas alternativas como Windows, Linux y Solaris; o para aquellos desarrolladores que necesiten extender y/o modificar el código del servidor streaming existente para satisfacer sus necesidades. Darwin Streaming Server solo es soportado por la comunidad del software libre y no es elegible para soporte técnico por Apple. Apple mantiene un número de listas de discusión para usuarios y desarrolladores de Darwin Streaming Server para compartir ideas y discutir escenarios de despliegue. Darwin Streaming Server está disponible como para descarga gratuita bajo la Apple Public Source License v2.

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. Permite realizar streaming a partir de videos codificados en el formato H264, un formato que brinda archivos multimedia de alta calidad visual con poco tamaño de los mismos en relación con otros formatos. El servidor cuenta con una interfaz web que permite visualizar su estado y además posibilita su configuración y administración.

1.8.6. VideoLan (VLC)

La solución para streaming de VideoLan incluye dos programas:

- VLC media player: el cual puede ser utilizado como un servidor y como un cliente para hacer streaming y recibir streaming por la red. VLC es capaz de hacer streaming con todos los archivos que puede leer.
- VLS (VideoLAN Server): el cual puede hacer streaming de archivos MPEG-1, MPEG-2 y MPEG-4, DVDs, canales digitales de satélite, canales de televisión digital terrestre y videos en vivo por la red en

multidifusión o unidifusión. La mayoría de las funcionalidades del VLS pueden ser ahora encontradas en el VLC.

El VLC es capaz de transmitir todos los archivos que sea capaz de reproducir por lo que puede ser utilizado como servidor Streaming para transmitir desde un dispositivo de captura o desde un fichero previamente almacenado con numerosas opciones de codificación en tiempo real. El proceso de creación de un nuevo flujo a transmitir lleva una serie de pasos que deben ser realizados desde la interfaz (ya sea gráfica, web o telnet) que incluyen la selección del tipo de flujo (en demanda o en vivo), la selección del origen (dispositivo de captura, DVD, archivo previamente almacenado, etc.), selección del formato de salida (transcodificación en tiempo real a numerosos formatos). En caso que se cierre la instancia del VLC que se encuentra configurada y transmitiendo los contenidos hay que realizar el proceso nuevamente para cada contenido.

Para lograr el uso óptimo de VLC como servidor de Streaming para la Plataforma VideoWeb habría que lograr que fuera capaz de leer, configurar y transmitir automáticamente los archivos que se encuentren almacenados en una carpeta específica y que esto lo haga cuando se inicie el servidor y cada cierto tiempo para que pueda detectar cuando se añada un nuevo video a la carpeta. Añadido a esto se debe encontrar la forma de mantener la instancia del VLC en ejecución de forma constante, brindando servicios. Esto es esencial para una aplicación que funcione como servidor.

1.8.7. Feng

Feng es un servidor streaming de multimedia que cumple con los estándares de la IETF (Internet Engineering Task Force) para el streaming en tiempo real de contenidos multimedia en internet. Feng implementa RTSP y RTP/RTCP (Real-Time Protocol/ Real-Time Control Protocol).

Feng soporta los siguientes estándares de codificación:

- Audio: Audio MPEG (MPEG-1/2 Layer I/II/III), Ogg Vorbis, AAC (MPEG-4 Part 3)
- Video: Video MPEG (MPEG-1/2), MPEG 4 (MPEG-4 Part 2), H.264 (MPEG-4 Part 10), H.263 / H.263+, Ogg Theora

Las principales características de Feng son el soporte de contenedores, la habilidad de manipular saltos en la reproducción y la estructura modular enfocada a facilitar la extensión del códec y el soporte de protocolo.

Dependencias:



- glib versión 2
- ffmpeg
- netembryo
- bufferpool

El principal problema de Feng consiste en que no ha podido ser instalado, luego de varios intentos, en las distribuciones de GNU/Linux en las que se ha intentado como Ubuntu 8.10, Nova 1.0.0 y Nova 1.1.2. Lo cual ha imposibilitado que se pruebe y demostrar que es factible su empleo.

1.8.8. MPEG4IP

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

1.8.8.1. Características del reproductor

El reproductor **mpeg4ip** soporta:

- **Formatos:** avi, mp4, mov, mpg(TS y PS), wav, raw aac, raw mp3, raw mp4v, raw .h264
- **Códec de Video:** mpeg-4 (xvid, xvid-1.0), mpeg1/2, H.261, YUV (i420 raw). Mediante ffmpeg: H.263, Sorenson, algunos MJPEG, H.264
- **Codecs de Audio:** aac (faad, faad2), mp3, celp, ac3, raw PCM, G.711 alaw y ulaw. Mediante ffmpeg: AMR NB, AMR WB, G.711 alaw y ulaw.
- **Capacidades de streaming:** Protocolos RTSP, SDP y RPT; streaming de ficheros mpeg2 TS, streaming por unidifusión y multidifusión.

1.8.8.2. Características de Mp4live

Mp4live solo está disponible para Linux, soporta lo siguiente:

- **Codecs de video:** mpeg4 (xvid, xvid-0.9.2 o xvid-1.0), H.261, mpeg2, h263, h264
- **Codecs de audio:** AAC, MP3, Mpeg2, AMR NB and WB, G.711 alaw y ulaw, Mpeg1
- **Filtros de video:** desentrelazado (Y solamente), decímate.
- **Interfaces de entrada de video:** V4L (Video for Linux), V4L2
- **Interfaces de salida de audio:** OSS
- **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

1.8.8.3. Herramientas de codificación

Debido a que otros proyectos (lame, faac, ffmpeg) proveen buenas herramientas de línea de comando, **Mp4live** solo provee y soporta el xvidenc el cual codifica videos mpeg4.

El uso del MPEGIP como servidor está pensado para la transmisión de contenidos en tiempo real o la creación de un archivo Mp4 desde un dispositivo de captura mediante el **Mp4live**. En la actualidad se utiliza en conjunto con el Darwin Streaming Server para la transmisión de contenidos en vivo, los cuales se capturan y codifican a Mp4 en tiempo real con el **Mp4live** y estos archivos Mp4 se transmiten con el Darwin como si fuera un archivo bajo demanda.

1.8.9.MEC4

MEC4 es un servidor streaming de video bajo demanda basado en una arquitectura multihilos para sistemas operativos Linux. El servidor puede manipular múltiples conexiones al mismo tiempo, las cuales entrega mediante un protocolo propietario simple SSP (SysInfo Service Protocol)/MSP (Multicast Synchronization Protocol). “Proprietario” se refiere a que fue creado por los desarrolladores del proyecto y funciona bien para sus propósitos.

1.8.9.1. Principales características:



Plataforma VideoWeb

- Número ilimitado de conexiones concurrentes (el único límite lo constituye el hardware)
- Volumen de información de hasta 2 Gbits/s por servidor.
- Capacidades de clústeres extendidas con número ilimitado de servidores en el servidor.
- Protocolo de Streaming muy simple pero efectivo: SSP/MSP
- Extremadamente configurable.
- Validación de usuario basado en dirección IP y permisos.
- Control de acceso a ficheros por usuario.
- Licencia GPL.

El MEC4 requiere un kernel de linux versión 2,2 con soporte para real time clock (rtc) habilitado, no se instala en Nova, Debian ni ninguna distribución de Ubuntu.

1.9. Metodología de Desarrollo de Software

El proceso de desarrollo de software no es simplemente tener un grupo de personas produciendo para terminar en tiempo el producto, pues, sin una guía basada en buenas prácticas, que organice el trabajo es casi imposible lograrlo. Las metodologías de desarrollo de software posibilitan lograr productos con mayor calidad y eficiencia, asegurando su desarrollo y mantenimiento.

La metodología de desarrollo se define como un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevos software (17).

Existen diversas metodologías, cada una con sus peculiaridades y puntos en común. Las metodologías tradicionales o robustas confieren gran peso a la planificación, conceptualización y descripción del sistema se pretende realizar. De esta forma garantizan que al comenzar la implementación este bien definido cada elemento a desarrollar. Este tipo de metodologías se ajusta para proyectos a largo plazo de gran envergadura, con equipos de desarrollo numerosos donde la organización sea fundamental. También es recomendable cuando se requiera una documentación amplia que detalle cada elemento para lo lograr un entendimiento posterior del software u otras razones.

Las Metodologías ágiles o livianas eliminan el burocratismo de las robustas, que en ocasiones resulta contraproducente emplearlas. Estas metodologías se centran en la capacidad de las personas involucradas en el proceso, evitando ir al detalle en cada paso, pero obteniendo el mayor fruto del trabajo de cada integrante. Perfecta para equipos de desarrollo con gran experiencia, la experiencia es vital en este tipo de metodologías para obtener buenos resultados. Estos equipos preferiblemente pequeños y en



Plataforma VideoWeb

proyectos que lo fundamental sea el producto final y la rapidez con que se concluya, sin que la documentación sea primordial.

Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Más Artefactos. El modelado es esencial, mantenimiento de modelos
Pocos Roles, más genéricos y flexibles	Más Roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de

	desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Figura 9: Diferencias entre metodologías ágiles y no ágiles (18)

Entre las metodologías de desarrollo de software más importantes por su calidad y aplicación en el mundo se encuentran la Metodología Rational Unified Process (RUP) o Proceso Unificado que se enmarca dentro de las metodologías clásicas, aunque existen variaciones de la misma para adaptarse a procesos ágiles. La Metodología Extreme Programming (XP), exponente de las metodologías ágiles y la Microsoft Solution Framework (MSF) que se adapta como metodología robusta o ágil.

1.9.1. Proceso Unificado (RUP)

El Proceso Unificado se caracteriza por ser dirigido por *casos de uso*⁷, centrado en la arquitectura e iterativo 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 productos en varias iteraciones, en cada una de ellas se obtiene una series de artefactos que servirá 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.

El ciclo de vida del Proceso Unificado cuenta de cuatro fases: inicio, elaboración, construcción y transición las que se subdividen en iteraciones y nueve flujos de trabajos, seis flujos de ingeniería y tres de apoyo.

⁷ Fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante.



Flujos de Trabajo

Flujos de ingeniería

- Modelamiento del negocio
- Requerimientos
- Análisis y diseño
- Implementación
- Prueba (Testeo)
- Instalación

Flujos de apoyo

- Administración del proyecto
- Administración de configuración y cambios
- Ambiente

1.9.2. Extreme Programming (XP)

XP proporciona una metodología para desarrollo de proyectos a corto plazo centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo (19).

Como elemento de gran importancia en la metodología XP se encuentran las historias de usuario. Las historias de usuarios son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (20).

De acuerdo con la propuesta original de Beck XP define los siguientes roles de usuarios: Programador, Cliente, Encargado de pruebas (Tester), Encargado de seguimiento (Tracker), Entrenador (Coach), Consultor y Gestor (Big boss).

El ciclo de vida ideal de XP consiste de seis fases (19): Exploración, Planificación de la Entrega (Release),

Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Entre las prácticas que propone XP se encuentran (18):

El juego de la planificación: El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

Entregas pequeñas: Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad pretendida para la aplicación.

Metáfora: El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema.

Diseño simple: Propone el diseño de la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto

Pruebas: La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse.

Refactorización (Refactoring): Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.

Programación en parejas: Toda la producción de código debe realizarse con trabajo en parejas de programadores.

Propiedad colectiva del código: Cualquier programador puede cambiar cualquier parte del código en cualquier momento.

Integración continua: Cada pieza de código es integrada en el sistema una vez que esté lista.

40 horas por semana: Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas.

Cliente in-situ: El cliente tiene que estar presente y disponible todo el tiempo para el equipo.

Estándares de programación: XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación.

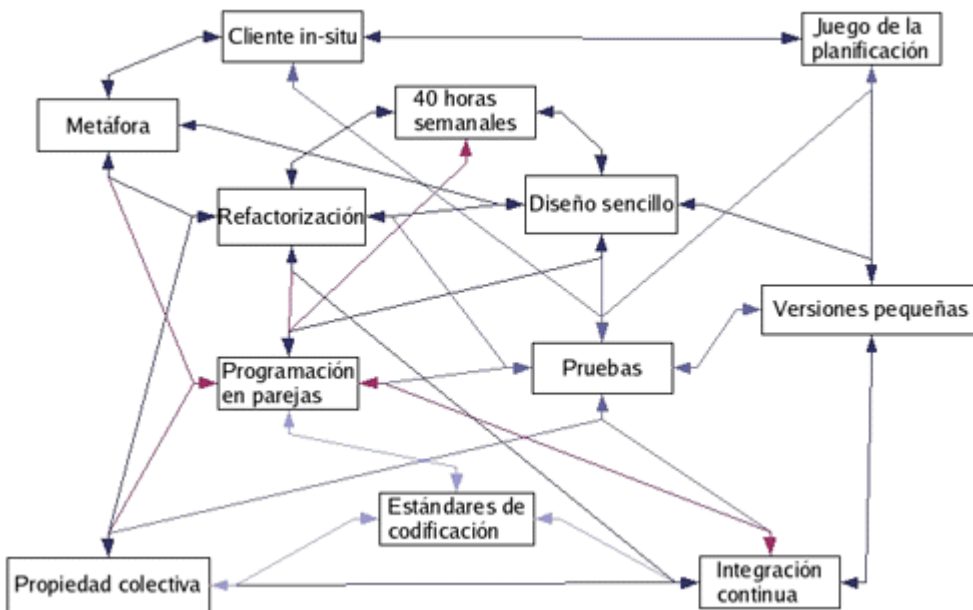


Figura 10: Interrelación entre las prácticas.

1.9.3. Microsoft Solution Framework (MSF)

MSF es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF se basa en Principios, Modelos y Disciplinas.

Principios:

- Promover comunicaciones abiertas.
- Trabajar para una visión compartida.
- Fortalecer los miembros del equipo.
- Establecer responsabilidades claras y compartidas.
- Focalizarse en agregar valor al negocio.
- Permanecer ágil y esperar los cambios.
- Invertir en calidad.
- Aprender de todas las experiencias.

Disciplinas

Administración de proyecto.

Esta disciplina se basa en planificar sobre entregas cortas, incorporar nuevas características sucesivamente e identificar cambios ajustando el cronograma.

Administración de riesgos

Esta disciplina está centrada en ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan salir. Proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos.

Control de cambio

Diseñada para que el equipo sea proactivo y no reactivo. Los cambios deben considerarse riesgos inherentes por lo que deben registrarse y hacerse evidente.

Modelos

Equipo de Trabajo

Alienta la agilidad para enfrentar los cambios involucrando a todo el equipo en las decisiones fundamentales asegurando una perspectiva crítica.

Proceso

Estrategia iterativa en la construcción de productos del proyecto, suministra una imagen más clara del estado de los mismos en cada etapa. Mejora el control del proyecto, minimiza los riesgos y aumenta la calidad acortando el tiempo de entrega.

1.10. **Conclusiones**

En este capítulo se han expuesto conceptos relacionados a los procesos asociados a la Plataforma VideoWeb, fundamentales para la comprensión de la presente investigación. Se analizaron las funcionalidades de plataformas que brindan servicios de audio y video a través de la web, entre ellas [VideoWebPeru](#), [es.youtube.com](#), [TV.com](#) de carácter internacional e [Inter-nos](#) en nuestro país, tomando de ellas las mejores experiencias para la Plataforma VideoWeb. Se abordaron temas relacionados a la Arquitectura de Software, analizando características y ventajas de los principales estilos arquitectónicos como flujo de datos, llamado y retorno, componentes independientes, centrados en datos y máquina virtual; y patrones arquitectónicos como Modelo Vista Controlador, Arquitectura en Tres Capas y Arquitectura Orientada a Objetos. Se analizaron las características de un conjunto de tecnologías,



Plataforma VideoWeb

herramientas y tendencias asociadas a la posible propuesta para solucionar el problema científico que concierne la presente investigación. Entre ellas se encuentran los CMS Plone, Joomla y Drupal, este último destacándose por su flexibilidad, robustez y funcionalidad. Se brindaron una serie de características de diversos lenguajes de programación como: PHP, Python y Java Server Pages (JSP), lo que son empleados en el desarrollo de sistemas informáticos Web. Se analizaron Sistemas Gestores de Bases de Datos libres como PostgreSQL, que se destaca por su robustez, facilidades de uso y soporte, y MySQL que a pesar de su potencia existen restricciones que dificultan su uso. Se proporcionan características de la tecnología de Media Streaming y de los principales servidores libres. Se destaca el servidor de streaming Darwin Streaming Server por la facilidad que brinda para la publicación de los archivos multimedia y la capacidad de realizar el streaming de videos en el formato H264, el cual mantiene alta calidad visual con poco tamaño. Se analizaron diferentes Metodologías de desarrollo de software, destacando el Proceso Unificado (RUP) debido a la amplia documentación que brinda, lo cual es esencial por la concepción como producto de la Plataforma VideoWeb. De esta manera se proporcionan los conocimientos teóricos necesarios para el entendimiento del presente trabajo.

CAPÍTULO 2: ARQUITECTURA PROPUESTA.

2. En el presente capítulo se abordarán aspectos relacionados con la Arquitectura propuesta para el desarrollo de la Plataforma VideoWeb. Se aborda sobre las principales ventajas de las herramientas y tecnologías seleccionadas desde un punto de vista que favorezca la rapidez y eficiencia del equipo de desarrollo en la elaboración de la Plataforma VideoWeb con el objetivo de obtener un producto de buena calidad.

2.1. Metodología de Desarrollo de Software

Como metodología para el desarrollo de software se propone el Proceso Unificado (RUP). La metodología RUP es muy popular y utilizada para la documentación y realización de software orientados a objetos.

El equipo de desarrollo cuenta con experiencia en el uso de la metodología RUP, aumentando la rapidez del desarrollo del producto. RUP propone varias iteraciones en el proceso de desarrollo de software posibilitando la corrección de errores y mejoras del software a medida que avanza el proyecto, lo que permite a los desarrolladores poder realizar un producto con gran calidad.

RUP brinda una fuerte y abundante documentación, permitiendo un futuro entendimiento de cada componente del producto. Esto es fundamental para la Plataforma VideoWeb debido a que la mayor parte del equipo de desarrollo lo conforman estudiantes de 5to año, los cuales pudieran no continuar en el equipo dentro de poco tiempo, requiriendo que sea documentado todo el proceso de desarrollo. Unido a esto, la concepción de la Plataforma VideoWeb como producto hace necesaria una buena documentación, favoreciendo el soporte, mantenimiento y producción de nuevas versiones. La experiencia del equipo de desarrollo en esta metodología constituye un factor fundamental en la selección de la misma.

2.2. Sistema Gestor de Base de Datos

Para el almacenamiento de la información referente a la Plataforma VideoWeb se propone utilizar PostgreSQL como Sistema Gestor de Base de Datos (SGBD). La Universidad de las Ciencias Informáticas creó una comunidad de PostgreSQL con el objetivo de garantizar soporte y mantenimiento a las aplicaciones que usen este SGBD. Esto contribuye con el prestigio y la calidad de los productos ya que ante cualquier problema o necesidad de información para optimizar las funcionalidades asociadas a la

base de datos, la comunidad garantiza la ayuda y apoya a los desarrolladores. PostgreSQL permite que los atributos de las tuplas tengan tipos complejos. Además, la base de datos (BD) es relacional, por lo que conserva su rapidez, eficiencia y permite hacer uso de nuevos elementos como las relaciones de herencia, que modelan los objetos que serán guardados en dicha BD, por lo que el analista y el diseñador perciben un modelo orientado a objetos. PostgreSQL garantiza la validez de los datos de la BD mediante Integridad Referencial y mediante MVCC (Multi-Version Concurrency Control) asegura una alta concurrencia de acceso a los datos, evita bloqueos innecesarios, es decir, permite la lectura de los datos sin que sea bloqueada por los que están actualizando registros. Además, PostgreSQL cumple con las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). Teniendo en cuenta lo descrito anteriormente, se escoge PostgreSQL como Sistema Gestor de Base de Datos para la Plataforma VideoWeb.

2.3. Herramienta de desarrollo

Para seleccionar la herramienta sobre la cual desarrollar la Plataforma VideoWeb se debe tener en cuenta que es más factible, usar un CMS o un Framework. Para esto debemos evaluar que facilidades brindan que posibilite un producto de mayor calidad y determinar cual se ajusta mejor a las necesidades del proyecto.

La Plataforma VideoWeb brindará servicios de audio y video, para lo cual es necesario realizar una buena gestión de los contenidos, unido a esto se pudieran agregar noticias, imágenes entre otros contenido que amplíen las funcionalidades del producto. Realizar esto implica tener los mecanismos de administración de usuarios necesarios y las funcionalidades para la gestión de audio y video. Los CMS tienen incorporados la gestión de usuarios que de utilizar un Framework se tendría que realizar. Los CMS están pensados para la gestión de contenidos por lo cual adicionarles las funcionalidades necesarias para la gestión de media no rompería con la concepción de los mismos.

Los Framework son potentes con el manejo de bases de datos, tienen clases de abstracción (varios motores de base de datos generalmente) y funcionalidades que ayudan a manipular los datos de forma más simple. En la Plataforma VideoWeb la interacción con base de datos no es de gran complejidad y perfectamente se puede utilizar las funcionalidades que brindan los CMS para esto.

Los CMS permiten modificar la interfaz sin alterar los datos ni las funcionalidades, lo que permite modificar de forma rápida y sencilla el producto y adaptarlo a cualquier cliente. La Plataforma VideoWeb como

producto requiere que sea flexible y fácil de mantener, los CMS están concebidos de esta forma.

Se propone el empleo de CMS como base para el desarrollo de la Plataforma VideoWeb ya que se ajusta más a las necesidades existentes. Permite enfocarse en el desarrollo de las funcionalidades para la gestión y publicación de archivos multimedia debido a que la mayor parte de las restantes funciones están implementadas en los CMS o requieren pocas modificaciones, contribuyendo con un desarrollo rápido y eficiente.

2.3.1. Selección de la Herramienta de Desarrollo

Después de un profundo análisis, debido a la paridad de los CMSs analizados en la presente investigación, se propone Drupal para el desarrollo de la Plataforma VideoWeb. El cual soporta como SGBD a PostgreSQL, que es el seleccionado para emplearse en el producto. Existen comunidades internacionales que brindan soporte y ponen a la disposición de todos numerosos módulos libres que sirven para ampliar las funcionalidades del CMS. Sobresale la flexibilidad y potencia del Drupal, módulos como Views, CCK y Panels permiten configurar y adaptar las funcionalidades y el diseño según las necesidades de cada cliente. Su rendimiento es superior al de Joomla, teniendo tiempos de respuestas más rápidos (ver Anexos). El lenguaje de programación que emplea para el desarrollo de sus módulos es PHP, en el cual el equipo de desarrollo está capacitado, facilitando la creación de los componentes necesarios para adaptar las funcionalidades a los requerimientos de la Plataforma VideoWeb.

2.4. Modelo Vista Controlador (MVC).

Durante el estudio realizado se escogió para el desarrollo de los módulos el patrón arquitectónico Modelo-Vista-Controlador, que forma parte del estilo arquitectónico de llamada y retorno.

El patrón MVC 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.

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.

Los equipos de trabajos pueden ser divididos por roles de acuerdo a los tres componentes de esta arquitectura. Los encargados de desarrollar las funcionalidades relacionadas con la manipulación de los

datos pueden especializarse en el *Modelo*. Los diseñadores de la interfaz se vinculan directamente con la *Vista*, siendo este el punto de interacción con el usuario, y donde se muestran las informaciones. Si la información no conlleva un procesamiento de datos profundo, se puede acceder directamente a las funciones de acceso a datos, lo que no implica que los encargados de la *Vista* conozcan las interioridades del *Modelo*, sino, como interactuar con él. Las funcionalidades complejas, que implique un procesamiento de datos relativamente grande se deben realizar en la *Controladora*. Para especializarse en este componente es necesario conocer cuáles serán las entradas desde la *Vista* y la interfaz de comunicación con el *Modelo*. Requiere de un mejor dominio de todo el entorno, pero no es necesario profundizar tanto en las particularidades de cada componente. El procesamiento es independiente de la forma en que se muestra o almacena la información.

Al emplear esta arquitectura, en los equipos de desarrollo se elimina el concepto de superhéroe, que realiza la carga principal del producto y se convierte en indispensable para que funcione completamente y se fomenta el concepto de trabajo en equipo, que favorece la calidad y rapidez de desarrollo.

2.5. Servidor Web

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

Tiene una arquitectura modular que permite adicionar o quitar de forma sencilla funcionalidades, permitiendo incrementar las prestaciones del servidor para numerosos usos. Aunque no cuenta con una interfaz visual para la administración del servidor, esta se realiza modificando un único fichero de configuración, de forma no muy compleja. Se caracteriza por la escalabilidad, seguridad y rendimiento. Apache trabaja con varios lenguajes como Perl, Python y PHP lo que permite desarrollar aplicaciones web de gran calidad.

2.6. Lenguaje de Programación

El lenguaje de programación a utilizar es PHP ya que es el empleado por el CMS seleccionado. La experiencia del equipo de desarrollo en el mismo posibilitará agilizar la preparación en la construcción de los módulos necesarios para el funcionamiento de la Plataforma VideoWeb.

2.7. Entorno de desarrollo integrado (IDE)

Como entorno de desarrollo se escoge el Eclipse con el módulo de PHP Development Tools (PDT) 2.0 para el desarrollo usando el lenguaje PHP.

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma que permite el desarrollo en varios lenguajes por sus características modulares. Cuenta con una gran comunidad que desarrolla constantemente nuevas extensiones y mejoras las existentes.

Eclipse cuenta con las siguientes características:

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

El eclipse permite incorporar el módulo de PHP Development Tools (PDT) 2.0 el cual es una popular herramienta de código abierto para el desarrollo en PHP. Ofrece todas las capacidades básicas de edición de código que los desarrolladores necesitan. La versión 2.0 se enfoca en el apoyo a la programación orientada a objeto 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.

2.8. Herramienta para el modelado del sistema

Se propone para el modelado del sistema la herramienta CASE Visual Paradigm. La cual constituye una herramienta profesional para el modelado UML que 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.

Algunas ventajas que ofrece:

- Multiplataforma (Incluyendo GNU/Linux)
- Diseño centrado en casos de uso y enfocado al negocio, permitiendo generar un diseño de software gran calidad.
- Permite realizar de ingeniería directa e inversa
- Uso de un lenguaje estándar común a todo el equipo de desarrollo facilitando la comunicación y agilidad de desarrollo.
- Modelo y código permanecen sincronizados en todo el ciclo de desarrollo.
- Permite integrarse con diferentes IDE (Integrated Development Environment).

Para el desarrollo de la Plataforma VideoWeb es fundamental que pueda ejecutarse en GNU/Linux ya que se desea un software implementado completamente con herramientas libres. El equipo de desarrollo no está familiarizado con la herramienta pero existe buena documentación de la misma, lo que hace que se pueda adaptar a la misma en un plazo de tiempo pequeño.

2.9. Servidor Streaming

Con el objetivo de realizar los servicios de audio y video es necesario incorporar en la Plataforma VideoWeb un servidor de streaming para optimizar este proceso. Para el desarrollo se requiere el uso de tecnologías libres, lo cual dificulta la selección del servidor streaming. Existen varias propuestas de software propietarios de estos que cumplen con las necesidades que el producto requiere, ejemplo de estos son: *QuickTime Streaming Server*, *Microsoft Windows Media Services* y el *Helix Server*, todos con múltiples funcionalidades y prestaciones que facilitan la realización de streaming, pero son propietarios, por lo que no se tomaran en cuenta para esta investigación.

Definir el servidor apropiado para la Plataforma VideoWeb no es una decisión sencilla, si bien no influye de manera significativa en la arquitectura del resto del producto, es vital para un funcionamiento adecuado de la misma. Sobre la base de lo que realmente se ha logrado utilizar, las funcionalidades y facilidades que ofrece se utilizará Darwin Streaming Server para realizar la transmisión de video bajo demanda. Como complemento para realizar la transmisión de video en vivo se seleccionó MP4LIVE, este se emplea para capturar el video y conformar un fichero que el Darwin Streaming Server sea capaz de realizar el streaming.

2.10. Servidor FTP

Para realizar la manipulación de los archivos multimedia (copiarlas al servidor de media y al servidor de streaming) se propone utilizar un servidor FTP, logrando que se agilice el proceso y no se carga con responsabilidades adicionales a los demás servidores. Para esta función se escoge 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.

VSFTPD se caracteriza por:

- Configuraciones de IP virtual
- Usuarios virtuales
- Potente configuración por usuarios.
- Límite por ancho de banda.
- Soporte para IPv6
- Apoyo a través de la encriptación SSL.

La configuración y administración del mismo se realiza de forma sencilla. Permite crear accesos a diferentes directorios por cada usuario, con distintos niveles de permisos sobre los mismos. Resulta ser un servidor muy estable y confiable.

2.11. Conclusiones

En el presente capítulo se describieron aspectos relacionados con la Arquitectura de Software propuesta para la Plataforma VideoWeb. Se seleccionaron las principales herramientas y tecnologías para el desarrollo del producto, teniendo en cuenta las ventajas que estas brindan y las facilidades para la implementación. Se seleccionó como metodología de desarrollo de software el Proceso Unificado (RUP), el cual brinda una abundante documentación, siendo esta característica fundamental para la Plataforma VideoWeb debido a su concepción como producto, significando la experiencia del equipo de desarrollo en esta metodología. PostgreSQL es el sistema gestor de base de datos (SGBD) seleccionado debido a que existe una comunidad que brinda soporte en la Universidad de las Ciencias Informáticas (UCI), es un SGBD robusto, ya que garantiza la integridad de los datos y posibilita una alta concurrencia de acceso a los mismos, además, conserva su rapidez y eficiencia ante grandes volúmenes de datos. El CMS Drupal fue seleccionado para desarrollar la Plataforma VideoWeb, destacando su flexibilidad y potencia, contando



Plataforma VideoWeb

con módulos libres para aumentar sus funcionalidades y la experiencia de varios proyectos en la UCI que desarrollan plataformas web empleando Drupal. Para el desarrollo de los módulos necesarios para la Plataforma VideoWeb se propuso el patrón arquitectónico Modelo Vista Controlador (MVC), facilitando el trabajo en equipo y logrando flexibilidad en los componentes desarrollados. Otras herramientas propuestas son:

- Eclipse con el módulo de PHP Development Tools (PDT) 2.0 como IDE de desarrollo
- Herramienta CASE Visual Paradigm con el objetivo de realizar el modelado de la aplicación.
- Darwin Streaming Server y MP4LIVE como servidores de streaming.
- VSFTPD como servidor ftp.
- Apache 2.2 como servidor web.

Esta propuesta fue confeccionada teniendo en cuenta que el producto debe ser desarrollado desde y para software libre.

CAPÍTULO 3: DESCRIPCIÓN DE LA ARQUITECTURA DE SOFTWARE.

En el presente capítulo se tomarán decisiones importantes para la arquitectura del sistema, proporcionando una visión general de la arquitectura del sistema, las metas y restricciones, mostrando las vistas arquitectónicas para representar diferentes aspectos del sistema. Serán descritas las metas y restricciones que influyen de manera significativa en la arquitectura.

3.1. Requerimientos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los requerimientos no funcionales influyen de manera significativa en la arquitectura. La Plataforma VideoWeb debe cumplir con los siguientes requerimientos:

3.1.1. Requerimientos de Software

- Se debe utilizar como servidor web el “Apache”.
- Se debe utilizar como servidor Streaming el “Darwin Streaming Server”.
- Se debe utilizar para captura y transmisión en vivo el “Mp4live”.
- Se debe utilizar como sistema operativo en los servidores el “Nova-Linux”.
- Para la reproducción de los archivos multimedia se debe utilizar el reproductor VLC, como reproductor externo y como plugin embebido en el navegador, por ser multiplataforma y soportar tanto el protocolo de transmisión como el formato de fichero empleado.

3.1.2. Requerimientos de Hardware

Servidor de archivos multimedia:

Cantidad de ficheros (1 hora promedio)	CPU	Memoria RAM	Cantidad de Almacenamiento	Tarjeta de red
0-1000	Intel Dual-Core E8400 (3.0GHz 1333MHz, 6MB	512 Mb	300 Gb - SCSI	Gigabit Ethernet

	L2)			
1000-4000	Intel Core 2 Duo Processor E8400 (6M Cache, 3.00 GHz, 1333 MHz FSB).	1 Gb	1 Tb - SCSI	Gigabit Ethernet
4000-10000	Intel Core 2 Duo Processor E8400 (6M Cache, 3.00 GHz, 1333 MHz FSB).	2 Gb	3 Tb – SCSI	Gigabit Ethernet

Figura 11: Características del Servidor de archivos multimedia

Servidor de aplicaciones

N# conexiones	CPU	RAM	Disco duro	Tarjeta de red
0-500	Intel Dual-Core Xeon 3085 (L2 - 4 MB, 3.00 GHz, 1333 MHz)	2 GB – DDR2	500 Gb SCSI (Recomendado implementar RAID 0)	Gigabit Ethernet
+ 500	Intel Quad-Core Xeon E5450 (3 GHz, 12 MB, 1333 MHz)	4 GB - DDR3	1 TB SCSI (Recomendado implementar RAID 0)	Gigabit Ethernet

Estación de captura (Al menos una estación por cada canal de Transmisión en Vivo)

CPU	RAM	Disco Duro	Tarjeta de red	Tarjetas de captura
Intel Dual-Core Xeon 3085 (L2 - 4 MB, 3.00 GHz, 1333 MHz)	2 Gb - DDR2	250 Gb - SCSI	Gigabit Ethernet	HAUPPAGE WINTV-PVR 350 PCI WATCH RECORD TV

Pc Cliente



Plataforma VideoWeb

CPU	RAM
Intel Pentium III – 2.0 GHz (o superior)	128 MB – DDR1

Al menos una Cámara de video Sony HDR-FX1000E con sus accesorios.

Para la confección de la propuesta de hardware de realizaron un conjunto de pruebas y valoraciones ([Ver Anexos](#)).

3.1.3. Restricciones en el diseño y la implementación.

- Diseño e implementación de una arquitectura flexible, que permita la fácil integración o desintegración de componentes.
- La arquitectura debe soportar migrar la interfaz de usuario de forma rápida.
- El patrón arquitectónico que se debe emplear en el desarrollo es el modelo-vista-controlador.
- El lenguaje de programación que se debe utilizar es PHP.
- Los protocolos para la comunicación que se deben usar son HTTP entre el cliente y el servidor web y RTSP entre el cliente y el servidor streaming.

3.1.4. Requerimientos de apariencia o interfaz externa

- Interfaz amigable, interactiva, intuitiva y de fácil comprensión para el usuario.

3.1.5. Requerimientos de confidencialidad

- Permitir autenticación segura.
- La seguridad se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema.

3.1.6. Requerimientos de disponibilidad

- El sistema debe de estar disponible las 24 horas del día.

3.1.7. Requerimientos de Usabilidad

- El módulo de administración debe caracterizarse por un alto grado de flexibilidad.
- Disponibilidad de visualizar los archivos multimedia en un reproductor.

- Mostar la información de forma lógica y correctamente estructurada.
- El servidor streaming debe mantener buenas prestaciones (demanda de archivos)

3.1.8. Requerimientos de Soporte

- El soporte y/o mantenimiento del sitio no debe detener el servicio.
- La capa de acceso a datos, debe soportar una migración del Gestor de Bases de Datos en proyecciones futuras.

3.1.9. Rendimiento

- El sistema debe responder en un tiempo relativamente rápido a las peticiones del usuario (menos de 5 segundos)

3.2. Vista de casos de uso

Se presentan a continuación los casos de usos que influyen de manera significativa en la arquitectura de la Plataforma VideoWeb. Para el mejor entendimiento de los mismos se separaron de acuerdo a los subsistemas por los que está compuesto la aplicación.

A partir de la vista de Casos de Uso, se puede definir los escenarios o los casos de uso que serán de interés para cada iteración del ciclo de desarrollo. Esta describe los escenarios o casos de uso que tienen significación y que encapsulan la funcionalidad central del sistema.

Los Casos de Uso arquitectónicamente significativos, son aquellos que describen funcionalidades imprescindibles para el sistema, y que a través de estos se valida la arquitectura propuesta para el mismo.

El sistema cuenta con 30 Casos de Usos, de los cuales 24 son arquitectónicamente significativos. Están representados por subsistemas de la siguiente forma:

Subsistema	Cantidad de Casos de Usos
Administración.	6
Gestión de Contenido.	18

Figura 12: Cantidad de Casos de Uso del sistema por módulos.

Los casos de usos arquitectónicamente significativos y una breve descripción de cada uno de ellos se podrán encontrar en los anexos:

Vista de Caso de Uso del Módulo de Gestión de Contenido.

Vista de Caso de Uso del Módulo de Administración.

Breve descripción de los casos de usos arquitectónicamente significativos.

3.3. Vista Lógica

En la descripción de la arquitectura, la vista lógica describe las clases más importantes, su organización en paquetes y subsistemas, y la organización de estos paquetes y subsistemas en capas.

La siguiente imagen muestra la distribución de los paquetes más significativos dentro de cada una de las partes definidas para la aplicación y la dependencia entre ellos.

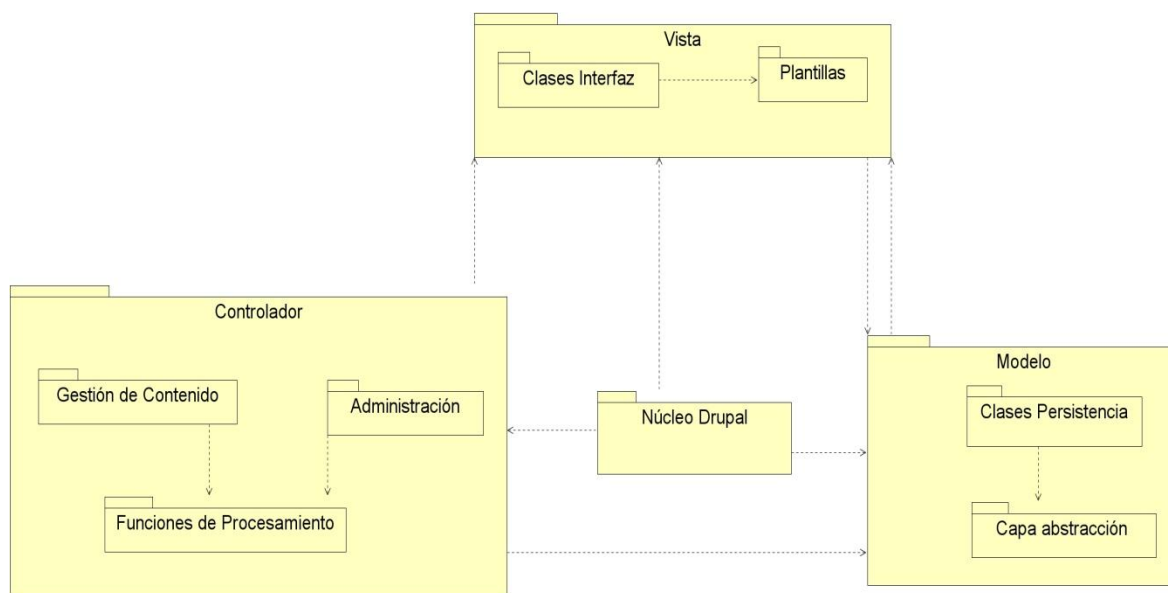


Figura 13: Vista Lógica

La vista está compuesta por cuatro paquetes principales:

3.3.1. Capa Vista:

En este paquete están las clases de interfaz las cuales se encargan de la presentación del contenido de

la aplicación e interactuar con el usuario. Se apoya con una serie de funciones que brinda Drupal para manipular y configurar la presentación del contenido a través de plantillas. Esto brinda facilidad de cambio y personalización de la aplicación ajustando la interfaz de forma eficiente.

La estructura de la capa Vista es la siguiente:

Clases Interfaz: Contiene todas las clases interfaz de los subsistema existentes de Administración y Gestión de Contenido.

Plantillas: Contiene funcionalidades del Drupal que facilita el trabajo con las platillas definidas para mostrar el contenido.

3.3.2. Capa Controlador:

Contiene las clases controladoras que encapsulan la lógica del negocio separado en los paquetes Administración y Gestión de Contenido que están relacionados con las funcionalidades del Drupal que facilitan la manipulación y validación de datos representados en el paquete Funciones de Procesamiento.

La estructura de la capa Controlador es la siguiente:

3.3.2.1. Gestión de Contenido

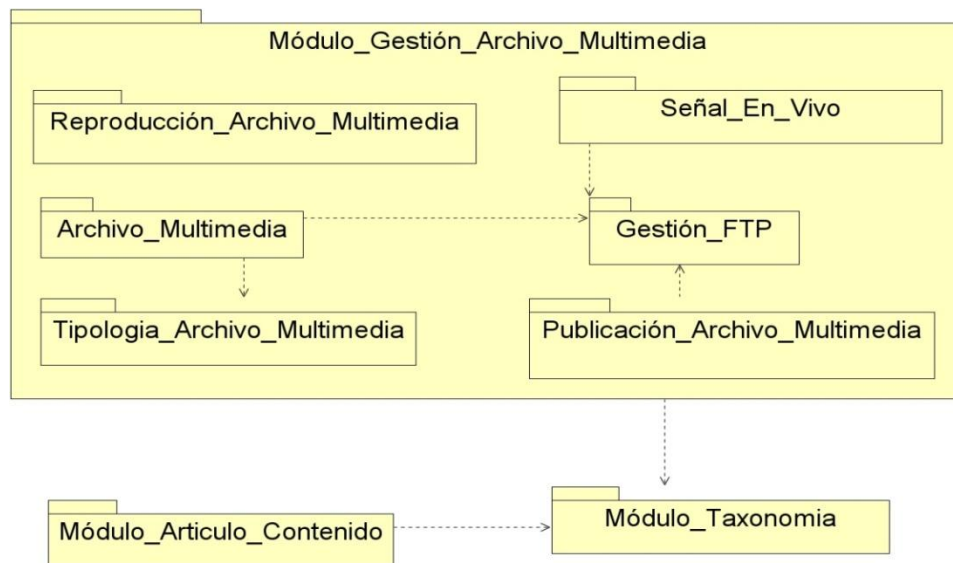


Figura 14: Módulo de gestión de archivos multimedia

Paquete **Módulo_Gestión_Archivo_Multimedia:** Contiene las clases asociadas a la gestión y

publicación de Archivo Multimedia agrupadas en paquetes según sus funcionalidades.

Los paquetes que contiene el Módulo de Gestión de Archivo Multimedia son:

Paquete **Archivo_Multimedia**: Contiene las clases asociadas a la manipulación de Archivos Multimedia.

Paquete **Reproducción_Archivo_Multimedia**: Contiene las clases asociadas a la reproducción de Archivo Multimedia

Paquete **Señal_En_Vivo**: Contiene las clases asociadas a la manipulación de Señal de audio y video en vivo.

Paquete **Gestión_FTP**: Contiene las clases asociadas a la gestión de servidores ftp para la manipulación de archivos multimedia entre los servidores de almacenamiento y streaming.

Paquete **Tipología_Archivo_Multimedia**: Contiene las clases asociadas a la gestión de las tipologías de Archivo Multimedia.

Paquete **Publicación_Archivo_Multimedia**: Contiene las clases asociadas a la gestión de publicaciones de Archivos Multimedia

Paquete **Módulo_Articulo_Contenido**: Contiene las clases asociadas a la gestión de Artículo de Contenido.

Paquete **Módulo_Taxonomia**: Contiene las clases asociadas a la gestión de las taxonomías que permitirá organizar las publicaciones de artículos de contenido.

Paquete **Funciones de Procesamiento**: Contiene funciones del Drupal que facilitan la manipulación y validación de datos.

Administración

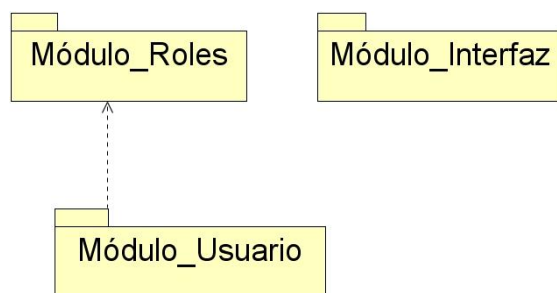


Figura 15: Módulo Administración

Paquete **Módulo_Roles**: Contiene las clases asociadas a la gestión de Roles, por los cuales los usuarios obtendrán los permisos.

Paquete **Módulo_Usuario**: Contiene las clases asociadas a la gestión de Usuarios así como la

asignación del Rol que desempeña en la aplicación.

Paquete **Módulo_Interfaz**: Contiene las clases asociadas a la gestión de Interfaz de usuario por la cual se administrara la forma en que se mostrara y organizara la información de la aplicación.

3.3.3. Capa Modelo

Contiene las clases de acceso a datos que encapsulan la lógica de interacción con la Base de Datos relacionadas con la capa de abstracción de datos del Drupal que facilita la manipulación de los datos y posibilita el uso de MySQL o PostgreSQL como SGBD.

La estructura de la capa Modelo es la siguiente:

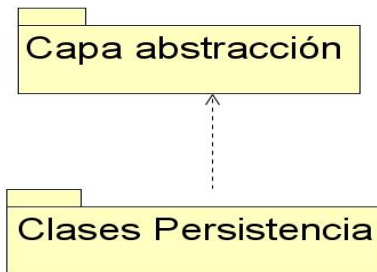


Figura 16: Capa Modelo

Paquete **Capa Abstracción**: Contiene las funcionalidades del Drupal que facilita la interacción y permite la abstracción con la Base de Datos.

Paquete **Capa Persistencia**: Contiene las clases de acceso a datos que encapsulan la lógica de interacción con la Base de Datos.

3.3.4. Núcleo Drupal

Contiene los componentes y funcionalidades de Drupal que aseguran el funcionamiento del sistema, permitiendo la integración de los módulos e interacción entre ellos.

3.4. Vista de procesos

La vista de procesos suministra una base para el entendimiento de la organización de los procesos del sistema, ilustrados en el mapeo de las clases y subsistemas en procesos e hilos. Solo suele usarse cuando el sistema presenta procesos o hilos concurrentes.

El acceso de los clientes a la Plataforma VideoWeb es independiente de la ejecución de otro cliente. La

conurrencia de uso del servidor Web, el servidor de Streaming y la Base de Datos es manejada por los propios servidores. No existen procesos del negocio concurrentes que deba ser manejado por la aplicación. Por estas razones no se requiere una Vista de Procesos.

3.5. Vista de despliegue

La vista de despliegue propone la distribución física del sistema y la distribución de los componentes de la aplicación en ellos. Existe una traza directa del modelo de implementación, puesto que cada componente físico debe estar almacenado en un nodo, esto incluye también la asignación de tareas provenientes de la vista de procesos en los nodos.

3.5.1. Diagrama de despliegue.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema. Es un conjunto de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos y procesos.

La distribución física de los principales componentes de la Plataforma VideoWeb se puede adaptar a múltiples configuraciones dependiendo del presupuesto disponible para la misma. Puede estar separado en servidores independientes el servidor de streaming, el servidor de almacenamiento, la estación de captura y el servidor web junto al paquete de la aplicación (*Ver anexos*). También es posible combinar estos componentes y lograr abaratar los costos a expensas de un descenso en el rendimiento. En la Figura 17 se presenta una de las posibles configuraciones físicas que puede tener la aplicación.

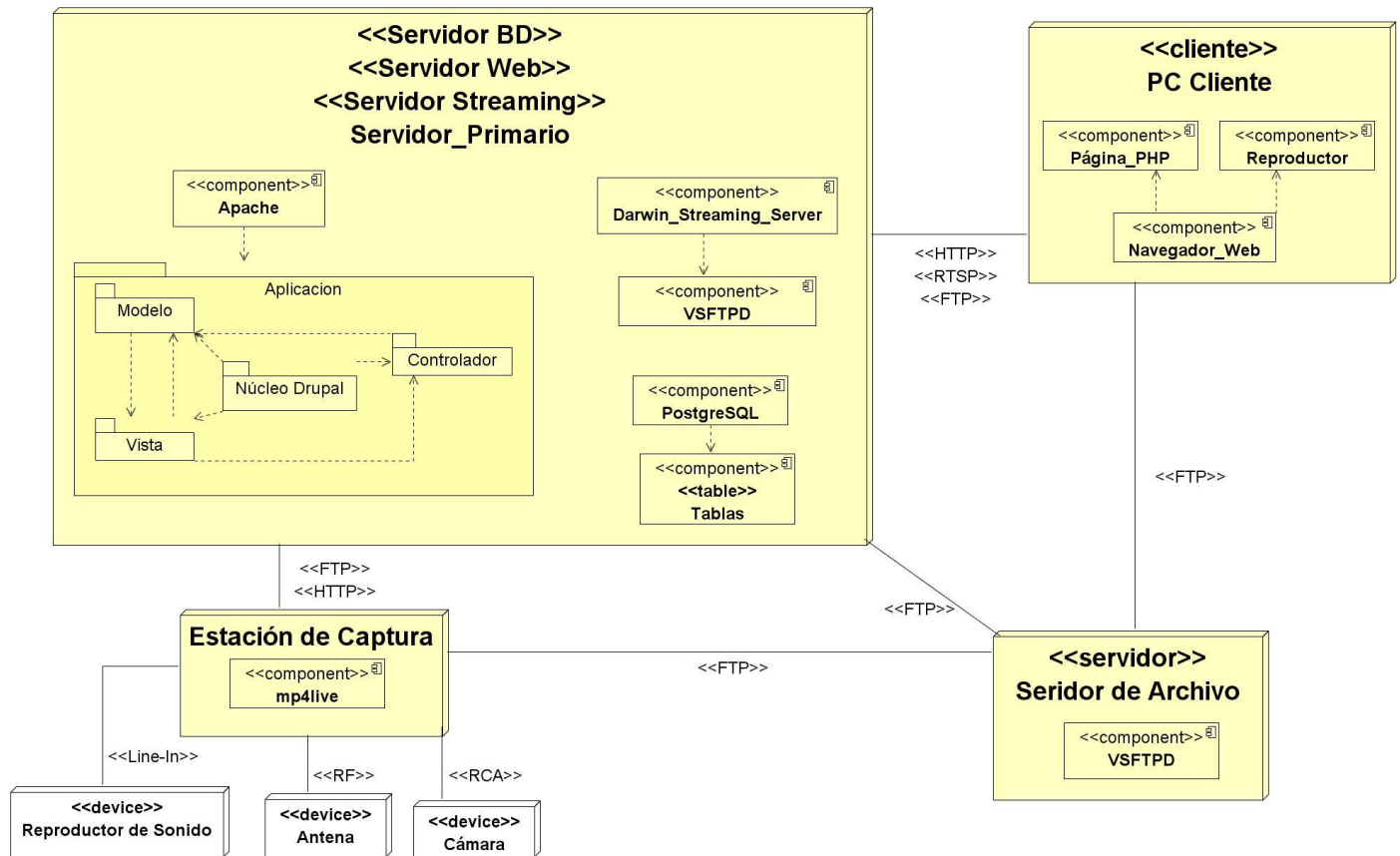


Figura 17: Diagrama de Despliegue.

A continuación se describirán los nodos físicos representados en el diagrama de despliegue anterior:

Servidor Primario

Representa el nodo que contiene los servicios fundamentales de la Plataforma VideoWeb. Estos servicios son: Servicio de Streaming mediante el Darwin Streaming Server para la transmisión de archivos multimedia, contiene el servidor web para la publicación de las páginas web y mostrar la interfaz de la aplicación. Cuenta con el servidor FTP VSFTPD para la manipulación de los ficheros del servidor de streaming.

Servidor de Archivo

Representa el nodo que contiene el servidor de media donde de almacenara los archivos de contenido. La manipulación de los archivos de hará mediante el servidor FTP VSFTPD.

PC Cliente

Representa el nodo que permite la interacción de los usuarios con la Plataforma VideoWeb. Permite la visualización de contenidos así como la configuración y administración del sistema.

3.5.2. Estación de captura

Representa el nodo que permite la captura de señales para la transmisión en vivo, para lo cual emplea mp4live, transmite la señal capturada al servidor streaming principal y este a su vez transmitirá la señal hacia los clientes.

3.5.3. Antena

Representa el dispositivo Antena para la captura de señal de televisión a través de la tarjeta de video de la estación de captura.

3.5.4. Cámara

Representa el dispositivo Cámara captura de señal de video a través de la tarjeta de video de la estación de captura.

3.5.5. Reproductor de Sonido

Representa el dispositivo Reproductor de Sonido para la captura de señal de audio a través de la entrada de línea de la estación de captura.

3.5.6. Descripción de elementos e interfaces de comunicación

Los protocolos de comunicación entre los componentes serán descritos a continuación:

RTSP: El protocolo de flujo de datos en tiempo real usado para la transmisión de media streaming desde la estación de captura y el servidor de streaming.

HTTP: Protocolo de transferencia de hipertexto usado en las transacciones hacia el servidor Web proveniente de los usuarios de la aplicación.

FTP: Protocolo empleado para la transferencia de archivos para el servidor de streaming y el servidor de almacenamiento.

Line-In: Línea de entrada para recibir el audio que va desde el reproductor de sonido hasta la estación de captura.

RF: Entrada de antena para recibir la señal de videos de canales hasta la tarjeta de video de la estación de captura.

RCA: Entrada de video para recibir la señal de cámaras de video hasta la tarjeta de video de la estación de captura.

3.6. Vista de Implementación

La vista de implementación proporciona una descripción de las principales capas y subsistemas de componentes de la aplicación. Los paquetes principales de la aplicación por cada uno de los módulos son:

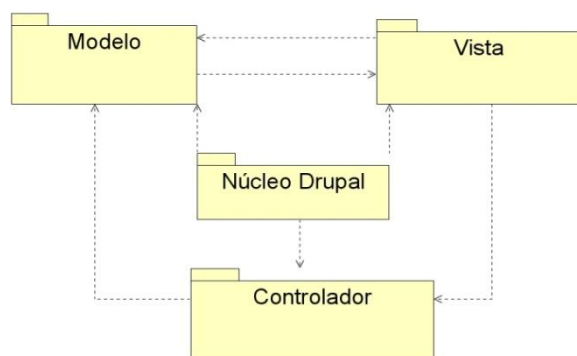


Figura 18: Paquetes principales de la aplicación

Los paquetes Modelo, Vista y Controlador encapsulan uno o más componentes que se interrelacionan entre ellos para darle solución a la aplicación los cuales son usados por el Núcleo de Drupal para conformar la aplicación.

La distribución de los componentes mediante capas y paquetes de la aplicación se realiza de la siguiente forma:

3.6.1. Vista

La capa Vista contiene los componentes relacionado a la presentación del contenido relacionados con el componente del Drupal theme.inc para la manipulación de las plantillas y la visualización.

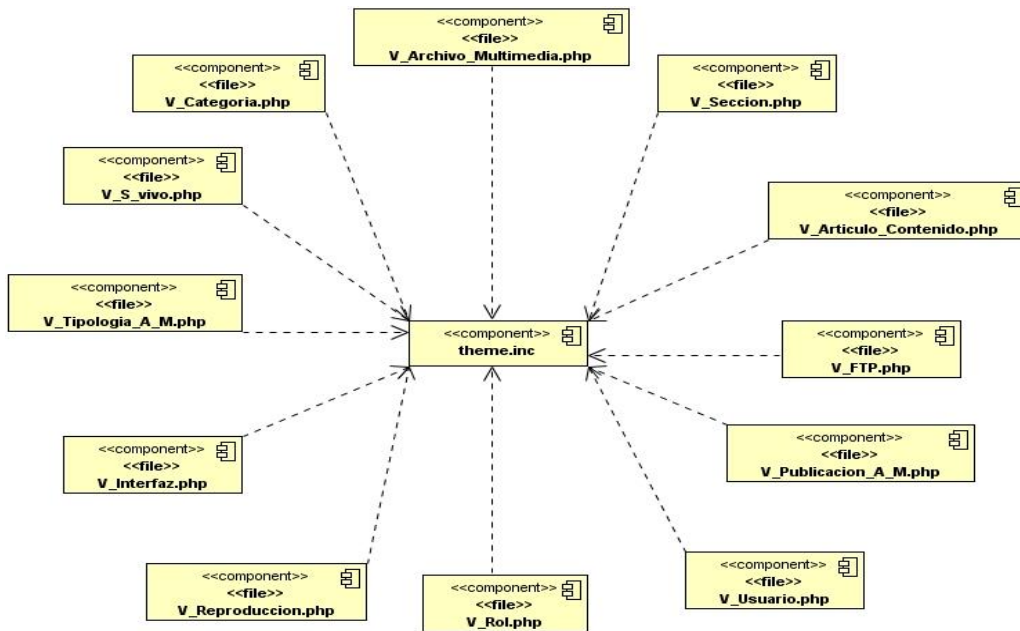


Figura 19: Componentes de la capa Vista

3.6.2. Controlador

La capa Controlador contiene los componentes relacionados al procesamiento, en cada Módulo existe un componente .module que interactúa con la controladora correspondiente, la cual encapsulan la lógica de la aplicación. Los .modules se encargan de manipular la información y gestionar la relación con el modelo y la vista.

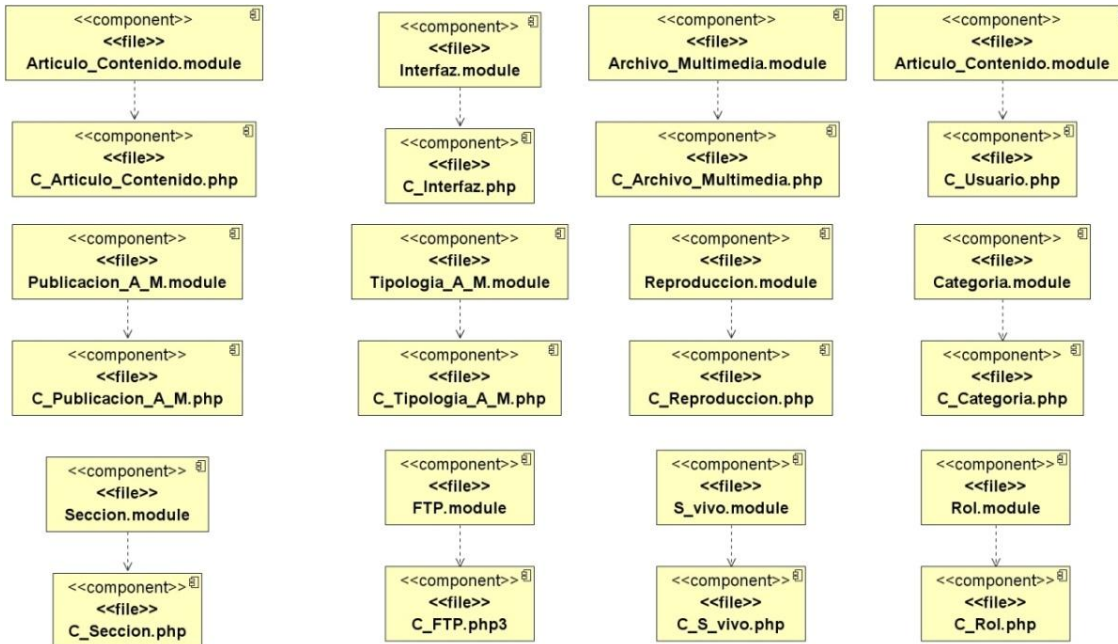


Figura 20: Componentes de la capa Controlador

3.6.3. Modelo

La capa Modelo contiene los componentes correspondientes al acceso a datos, estos están relacionados con el componente del Drupal database.inc que contiene las funcionalidades que posibilita la abstracción con la Base de Datos.

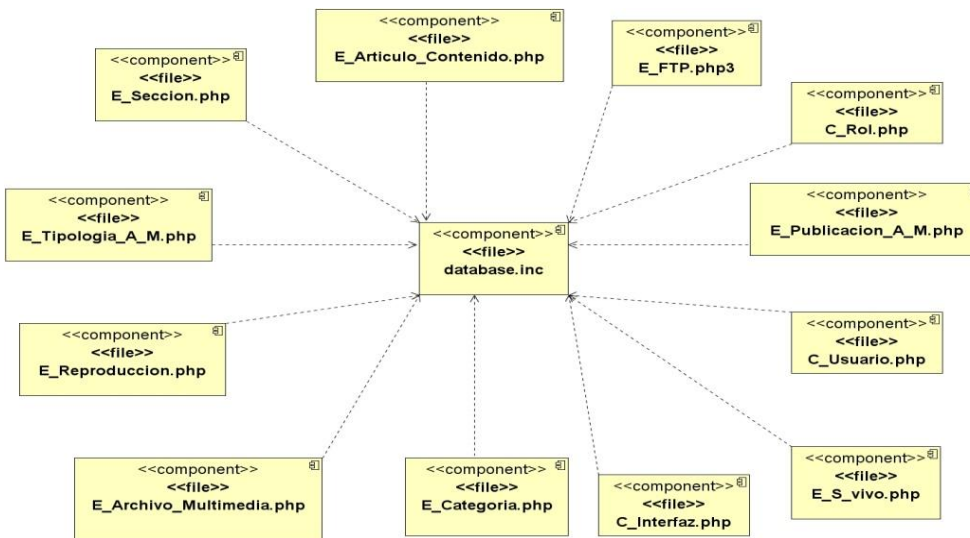


Figura 21: Componentes de la capa Modelo

3.6.4. Núcleo Drupal

La capa Núcleo Drupal contiene los componentes fundamentales de Drupal que posibilitan el funcionamiento de la aplicación, se encargan de ejecutar los módulos y manipular el acceso a los mismos así como las funcionalidades complementarias del Drupal.

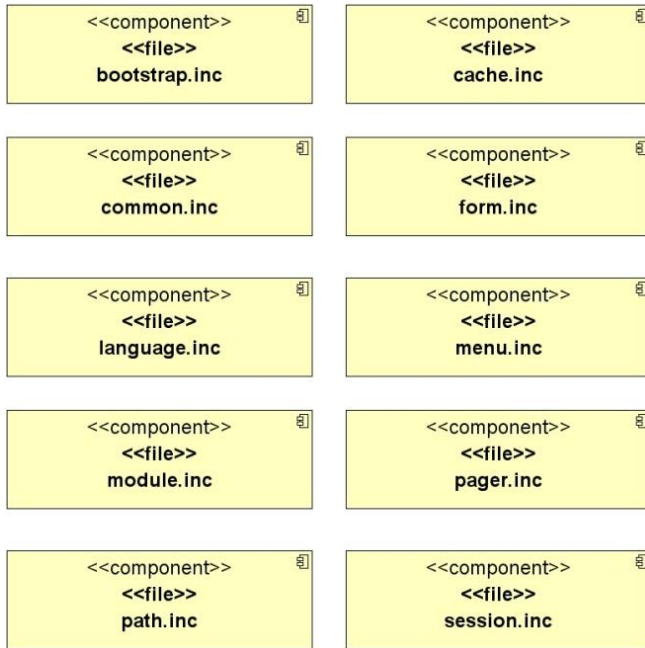


Figura 22: Componentes de la capa Núcleo Drupal

3.7. Vista de Datos

El sistema cuenta con 34 tablas que permiten la persistencia de datos para los módulos de forma común. Todas las tablas son arquitectónicamente significativas para el sistema.

Como Gestor de Bases de Datos se utiliza PostgreSQL. Es un sistema gestor de base de datos objeto-relacional (ORDBMS) de código abierto que tiene una alta concurrencia, usa integridad referencial para garantizar la validez de los datos y es muy estable y robusta ante grandes peticiones.

Drupal brinda funcionalidades que posibilitan la abstracción a la Base de Datos, por lo cual toda la interacción del sistema con la Base de Datos se hará mediante estas funcionalidades.

3.8. Estructura del Equipo de Desarrollo

El equipo de trabajo está distribuido de la siguiente forma:

- Líder de proyecto
- Analista-Diseñador
- Arquitecto de software
- Diseñador de base de datos
- Desarrollador
- Calidad

3.8.1. Configuración de los puestos de trabajos por roles

Analista-Diseñador (2 miembros)

Una PC, con mouse y teclado.

Instalación de Visual Paradigm.

Instalación del paquete Oficce/ Open Oficce.

Desarrollador (2 miembros)

PC, con mouse y teclado.

Instalación Del IDE Eclipse + PHP Development Tools (PDT) 2.0.

Servidor Apache y Drupal.

Arquitecto de software y Líder de proyecto

Una PC, con mouse y teclado.

Instalación de Visual Paradigm.

Instalación del paquete Oficce/ Open Oficce.

Diseñador de base de datos y Calidad.

Una PC, con mouse y teclado.

Instalación de Visual Paradigm.

Instalación del paquete Oficce/ Open Oficce.

Instalación del PostgreSQL 8.2 y pgAdmin III

3.9. Conclusiones

En el presente capítulo se plantearon los requerimientos no funcionales, que influyen de manera significativa en la Arquitectura de Software. Se definieron las vistas arquitectónicas (Vista de casos de



Plataforma VideoWeb

uso, Vista Lógica, Vista de procesos, Vista de despliegue y Vista de Implementación) vitales para el desarrollo de la plataforma y la estructura del equipo de desarrollo.

CAPÍTULO 4: EVALUANDO LA ARQUITECTURA DE SOFTWARE.

La evaluación de la arquitectura de software es fundamental para conocer el impacto que tiene sobre los atributos de calidad definidos. De esta forma permite conocer los problemas, debilidades y puntos de interés para posibilitar una corrección a tiempo de los mismos. Cuanto más temprano se encuentren los problemas en un proyecto de software, menor serán los daños que puedan ocasionar.

4.1. ¿Cuándo una Arquitectura puede ser evaluada?

Un aspecto fundamental para realizar la evaluación de la arquitectura de software es el momento que se realiza. Existen dos variantes útiles para realizar la misma, la evaluación temprana y la evaluación tardía.

4.1.1. La evaluación temprana

Para realizar este tipo de evaluación no es necesario que la arquitectura se encuentre completamente realizada. Permite efectuar decisiones sobre la arquitectura en cualquier nivel, puesto que se pueden atribuir cambios arquitectónicos producto de una evaluación en función de los atributos de calidad esperados.

4.1.2. La evaluación tardía

Para realizar este tipo de evaluación es necesario que la arquitectura del sistema se encuentre establecida y su implementación esté concluida, es decir, en el momento que el sistema ya esté terminado. Se considera que la evaluación en este punto es muy importante y útil, puesto que puede observarse el cumplimiento de los atributos de calidad asociados al sistema y su comportamiento general. La realización de la evolución de la AS debe realizarse cuando estén listos los elementos necesarios para justificarla. Puede ser determinada en el momento que el equipo de desarrollo necesite tomar decisiones que dependen de la arquitectura propuesta y que el costo de no tenerlas en cuenta daría al trasto con un costo superior al de llevar a cabo una evaluación.

Generalmente se responsabiliza de la evaluación de la AS a los miembros del equipo de desarrollo (Arquitecto, Analista, Diseñador, etc.) pero no son los únicos que pueden realizar esta tarea. Se pueden contratar especialistas en el tema para que realicen las pruebas o los clientes involucrados en el negocio los que en dependencia de los resultados pueden decidir si es factible para ellos continuar el proyecto o no.

4.2. ¿Qué resultados produce la evaluación de la Arquitectura de Software?

La evaluación de la arquitectura de software no especifica si la arquitectura es buena o mala, simplemente expresa donde están los riesgos y fortalezas de la arquitectura de software. La evaluación de la arquitectura produce un informe, el cual puede ser diferente en dependencia del método utilizado. Este informe produce respuesta a dos tipos de preguntas fundamentalmente:

- ¿Es esta arquitectura adecuada para el sistema para el cual fue diseñada?
- ¿Cuál de las arquitecturas propuestas es la más adecuada para el sistema?

Una arquitectura es adecuada cuando cumple dos criterios:

- El sistema resultante cumple con los atributos de calidad definidos.
- El sistema puede ser construido con los recursos disponibles para el mismo.

Lo fundamental en la evaluación es conocer como los atributos de calidad son afectados por las decisiones del diseño arquitectónico.

4.3. ¿Por cuales cualidades puede ser evaluada la Arquitectura de Software?

Las cualidades o atributos de calidad son requerimientos del sistema que hacen referencia a características que éste debe satisfacer. Algunos de los atributos por los cuales puede ser evaluada una arquitectura son:

Atributo Calidad	de	Descripción
Disponibilidad (<i>Availability</i>)		Es la medida de disponibilidad del sistema para el uso (21).
Confidencialidad (<i>Confidentiality</i>)		Es la ausencia de acceso no autorizado a la información (21)
Funcionalidad (<i>Functionality</i>)		Habilidad del sistema para realizar el trabajo para el cual fue concebido (22).

Desempeño (Performance)	Es el grado en el cual un sistema o componente cumple con sus funciones designadas, dentro de ciertas restricciones dadas, como velocidad, exactitud o uso de memoria. (23).
Confiabilidad (Reliability)	Es la medida de la habilidad de un sistema a mantenerse operativo a lo largo del tiempo (21).
Seguridad externa (Safety)	Ausencia de consecuencias catastróficas en el ambiente. Es la medida de ausencia de errores que generan pérdidas de información (21).
Seguridad interna (Security)	Es la medida de la habilidad del sistema para resistir a intentos de uso no autorizados y negación del servicio, mientras se sirve a usuarios legítimos (21).

Figura 23: Descripción de atributos de calidad observables vía ejecución.

Atributo de Calidad	Descripción
Configurabilidad (Configurability)	Posibilidad que se otorga a un usuario experto a realizar ciertos cambios al sistema (24).
Integrabilidad (Integrability)	Es la medida en que trabajan correctamente componentes del sistema que fueron desarrollados separadamente al ser integrados.
Integridad (Integrity)	Es la ausencia de alteraciones inapropiadas de la información (21).
Interoperabilidad (Interoperability)	Es la medida de la habilidad de que un grupo de partes del sistema trabajen con otro sistema. Es un tipo especial de <i>integrabilidad</i> (22).
Modificabilidad (Modifiability)	Es la habilidad de realizar cambios futuros al sistema (24).



Mantenibilidad (<i>Maintainability</i>)	Es la capacidad de someter a un sistema a reparaciones y evolución (21). Capacidad de modificar el sistema de manera rápida y a bajo costo (24).
Portabilidad (<i>Portability</i>)	Es la habilidad del sistema para ser ejecutado en diferentes ambientes de computación. Estos ambientes pueden ser hardware, software o una combinación de los dos (22)
Reusabilidad (<i>Reusability</i>)	Es la capacidad de diseñar un sistema de forma tal que su estructura o parte de sus componentes puedan ser reutilizados en futuras aplicaciones
Escalabilidad (<i>Scalability</i>)	Es el grado con el que se pueden ampliar el diseño arquitectónico, de datos o procedimental (25)
Capacidad de Prueba (<i>Testability</i>)	Es la medida de la facilidad con la que el software, al ser sometido a una serie de pruebas, puede demostrar sus fallas. Es la probabilidad de que, asumiendo que tiene al menos una falla, el software fallará en su próxima ejecución de prueba (21).
Flexibilidad (<i>flexibility</i>)	Grado en que el sistema se puede adaptar a los cambios y la puesta en práctica de innovaciones.

Figura 24: Descripción de atributos de calidad no observables vía ejecución.

4.4. Técnicas de evaluación

La evaluación de la arquitectura de software se puede realizar usando varias técnicas, estas se clasifican en cualitativas y cuantitativas. En las técnicas de evaluación cualitativas se utilizan escenarios, cuestionarios o listas de verificación. Por su parte en las técnicas de evaluación cuantitativas se emplean métricas, simulaciones, prototipos, experimentos o modelos matemáticos.

Las técnicas de evaluación cualitativas son usadas cuando la arquitectura se encuentra en construcción y arrojan como resultados respuestas de sí o no. Posibilitan evaluar una arquitectura o hacer comparaciones entre arquitecturas candidatas y determinar cual satisface más un atributo de calidad específico. Mientras que las técnicas de evaluación cuantitativas, se usan cuando la arquitectura ya ha sido implantada.

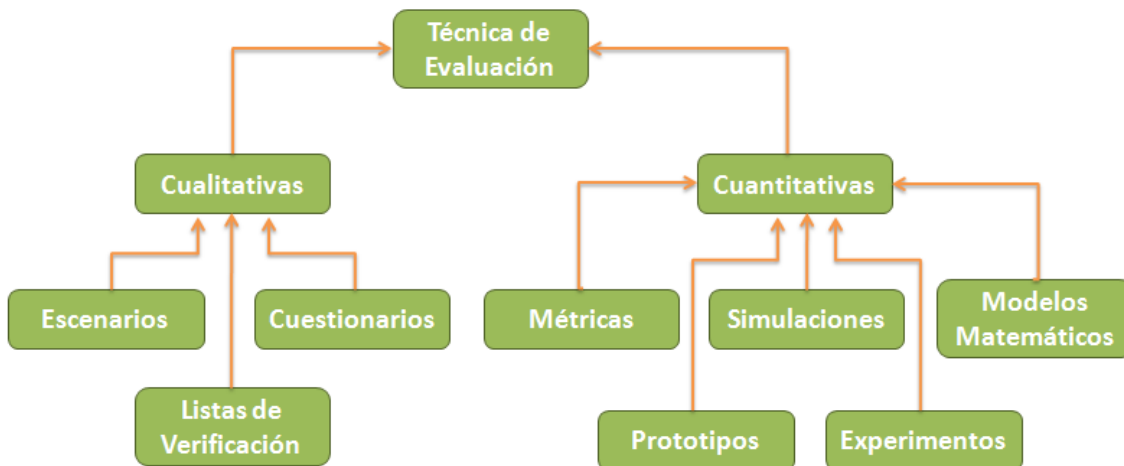


Figura 25: Clasificación de las Técnicas de Evaluación

4.5. Principales métodos de prueba de arquitectura de software

Existen múltiples métodos para realizar pruebas a la arquitectura de software, cada uno con sus características específicas. Es necesario tener en cuenta las características fundamentales del software para escoger la ideal, teniendo en cuenta las fortalezas y debilidades de la misma. Entre los principales métodos se encuentran: SAAM (Software Architecture Analysis Method), ARD (Active Design Review), ATAM (Architecture Tradeoff Analysis Method) y ARID (Active Review Intermediate Designs).

4.5.1. SAAM

El Método de Análisis de Arquitecturas de Software (Software Architecture Analysis Method, SAAM) es un método basado en escenarios que permite la evaluación de atributos de calidad, especialmente en la modificabilidad. También evalúa la relación del diseño arquitectónico con los requerimientos del sistema. Es posible evaluar una arquitectura o múltiples arquitecturas utilizando SAAM aunque el resultado no emite un valor absoluto referente a la calidad del sistema.

Para la realización de las pruebas SAAM define tres roles:

- **Interesados externos:** Organización, cliente, usuarios finales, administradores del sistema, etc.
- **Interesados internos:** Arquitectos de Software, analistas, especialista en requerimientos.
- **Equipo SAAM:** Líder, expertos en el dominio de la aplicación, expertos externos en arquitectura y secretario.

El método SAAM define una serie de pasos para realizar la evaluación:

- Desarrollo de escenarios.
- Descripción de la arquitectura.
- Clasificación de escenarios.
- Clasificación y asignación de prioridad de los escenarios.
- Evaluación individual de los escenarios indirectos.
- Evaluación de la interacción entre escenarios.

Entre las potencialidades de esta metodología se encuentra que los interesados comprenden con facilidad las arquitecturas evaluadas y la documentación es mejorada. El esfuerzo y el costo de los cambios pueden ser estimados con anticipación.

Como deficiencia de este método se encuentra la generación de escenarios, pues está basada en la visión de los interesados. No provee una métrica clara sobre la calidad de la arquitectura evaluada. El equipo de evaluación depende de la experiencia de los arquitectos para proponer arquitecturas candidatas.

4.5.2. **ADR**

El método de Revisión de Diseño Activo o ARD (Active Design Review) se emplea en la evaluación de diseños bien detallados de unidades de software como los componentes. Se centra en la calidad y el nivel de completamiento de la documentación, y en el nivel de conveniencia y suficiencia de los servicios que contiene el diseño propuesto (26).

4.5.3. **ATAM**

La ATAM pretende ser un método de identificación de riesgo, un medio de detectar áreas de riesgo potencial dentro de la arquitectura de un sistema intensivo de software complejo. Revela la forma en que una arquitectura satisface ciertos atributos de calidad, provee una visión de la forma que interactúan estos atributos con otros. ATAM se inspira en las áreas de estilos arquitectónicos, análisis de atributos de calidad y el método de evaluación SAAM.

La ATAM no presenta ninguna restricción con respecto a la característica de calidad a evaluar. Suele ser relativamente barata y rápida (porque se trata de evaluar el diseño de la arquitectura de los artefactos). El análisis de ATAM producirá resultados en consonancia con el nivel de detalle de la especificación de la

arquitectura. Además, no es necesario producir análisis detallados de cualquier atributo de calidad del sistema (como el tiempo medio de retardo o el tiempo de fallo) para tener éxito (27). ATAM consta de nueve pasos, divididos en cuatro fases.

4.5.4. ARID

El método ARID es un híbrido del método ARD y ATAM. ARID constituye un método conveniente para la evaluación de diseños parciales en las etapas tempranas del desarrollo usando técnicas de evaluación basada en escenario. Utiliza para la evaluación del diseño unidades del software como los componentes o módulos. Las preguntas giran en torno a la calidad y completitud de la documentación y la suficiencia, el ajuste y la conveniencia de los servicios que provee el diseño propuesto. Define los roles de Arquitecto, Equipo de verificación y Stakeholders y comprende nueve pasos agrupados en dos fases.

El método ARID evalúa el grado en que los atributos de calidad satisfacen en cada uno de los escenarios definidos. Como resultado de la aplicación de dicho procedimiento se obtiene un diseño de alta fidelidad acompañado de una alta familiarización con el diseño de los stakeholders (28).

4.6. Propuesta de Estrategia de evaluación de la Arquitectura

Para la evaluación de la Arquitectura de Software de la Plataforma VideoWeb se proponen dos estrategias de evaluación. Se realizará una primera evaluación en etapas tempranas del desarrollo con el fin de medir los principales atributos de calidad propuestos y tomar medidas de acuerdo a los resultados arrojados por la misma. La segunda evaluación se realizará cuando la arquitectura del sistema se encuentre establecida y su implementación esté concluida lo cual podrá arrojar resultados benéficos para el conocimiento del grado de cumplimiento de los atributos de calidad propuestos, cuando el sistema este en funcionamiento.

4.6.1. Estrategia de evaluación temprana (ARID).

En la estrategia de evaluación temprana de la AS se analizarán los atributos de calidad de Modificabilidad, Interoperabilidad, Seguridad, Funcionalidad y Disponibilidad. Se empleará el método de evaluación ARID que posibilita la evaluación en etapas tempranas del desarrollo de software evaluando el grado en que los atributos de calidad satisfacen en cada uno de los escenarios definidos. Se debe seleccionar al menos tres miembros de equipo de desarrollo (preferiblemente los de mayor conocimientos sobre las funcionalidades de la aplicación) para conformar el equipo de evaluación.

El método ARID comprende nueve pasos agrupados en dos fases:



Plataforma VideoWeb

- Fase 1 (Actividades Previas)
 - Identificación de los encargados de la revisión.
 - Preparar el informe de diseño.
 - Preparar los escenarios base.
 - Preparar los materiales
- Fase 2 (Evaluación)
 - Presentación del ARID.
 - Presentación del diseño.
 - Lluvia de ideas y establecimiento de prioridad de escenarios.
 - Aplicación de los escenarios.
 - Conclusiones

Se propone para la selección de los escenarios que los miembros del equipo de desarrollo implicado en el proceso de evaluación establezcan un escenario por cada caso de uso del sistema. En cada uno de estos escenarios se realizarán las pruebas en correspondencia con los atributos de calidad definidos.

A continuación se definirán puntos clave a tener en cuenta durante la realización de la estrategia de evaluación según los atributos de calidad:

Evaluación del atributo Modificabilidad:

Debido a que la Modificabilidad es la habilidad de realizar cambios futuros al sistema, se propone:

Valorar las consecuencias y el costo de:

- Cambiar la interfaz de la aplicación.
- Adaptar la aplicación a otros CMS.
- La incorporación de nuevas tipologías de archivos multimedia.
- Cambiar el servidor de streaming.
- Cambiar el servidor ftp.
- Cambiar el Sistema Gestor de Base de Datos.
- Cambiar el Servidor Web.

Analizar los requerimientos que puedan variar para futuras versiones de la plataforma y valorar la factibilidad de la arquitectura para adaptarse a los mismos.

Evaluación del atributo Interoperabilidad:

Teniendo en cuenta que la Interoperabilidad es la medida de la habilidad de que un grupo de partes del sistema trabajen con otro sistema, se propone:

El equipo de evaluación escogerá las funcionalidades que puedan ser de interés para otras aplicaciones. Valorará el costo de realizar las funcionalidades necesarias para posibilitar la interrelación con otras aplicaciones (preferiblemente usando servicios web).

Se deben tener en cuenta las siguientes funcionalidades:

- Mostrar artículos de contenidos publicados.
- Mostrar las nuevas publicaciones de artículos de contenido.
- Mostrar los archivos multimedia publicados.
- Mostrar las nuevas publicaciones de archivos multimedia.

El equipo de desarrollo deberá incluir otras funcionalidades.

Evaluación del atributo Seguridad:

Teniendo en cuenta que la Seguridad es la medida de ausencia de errores que generan pérdidas de información y la habilidad del sistema para resistir a intentos de usos no autorizados y negación del servicio, mientras se sirve a usuarios legítimos, se propone:

El equipo de evaluación seleccionará diferentes escenarios donde el acceso a los recursos esté limitado a un rol de usuario, al menos dos escenarios por cada rol del sistema. Se probará en cada escenario definido el acceso de usuario con diferentes roles.

Realizar intentos de acceso a la interfaz web de administración del Darwin Streaming Server y el mp4life por usuarios con privilegios y usuarios sin privilegios.

Durante la publicación de contenidos, introducir cadenas que contengan códigos que puedan dañar el funcionamiento de la aplicación.

Introducir cadenas que contengan códigos que puedan dañar el funcionamiento de la aplicación en el formulario de registro.

Evaluación del atributo Funcionalidad:

Teniendo en cuenta que la Funcionalidad es la habilidad del sistema para realizar el trabajo para el cual fue concebido, se propone:

El equipo de evaluación medirá el grado de cumplimiento de cada uno de los requerimientos funcionales definidos en el sistema.

Evaluación del atributo Disponibilidad:

Teniendo en cuenta que la Disponibilidad es la medida de disponibilidad del sistema para el uso, se propone:

Valorar las consecuencias sobre los servicios cuando se efectúan las acciones siguientes:

- Adicionar, Modificar y Eliminar artículos de contenido.
- Publicar y dejar de publicar archivos multimedia.
- Realizar modificaciones en el Sistema Gestor de Bases de Datos sin afectar la conexión con el resto de la aplicación.
- Realizar modificaciones en el Servidor de Streaming sin afectar la conexión con el resto de la aplicación.
- Realizar modificaciones en el Servidor de almacenamiento de Medias sin afectar la conexión con el resto de la aplicación.
- Realizar modificaciones en el Servidor Web sin afectar la conexión con el resto de la aplicación.
- Detener los servicios del Sistema Gestor de Base de Datos.
- Detener los Servicios del Servidor de Streaming.
- Detener los Servicios del Servidor de almacenamiento de Medias.
- Detener los Servicios Del Servidor Web.
- Solicitar el mismo recurso (artículo de contenido o publicación de archivo multimedia) por múltiples usuarios.
- Solicitar recursos independientes por múltiples usuarios.

Incorporar a estas pruebas otras acciones de mantenimiento que el equipo de evaluación estime conveniente.

Los resultados obtenidos de la evaluación se expondrán ante todos los involucrados en el desarrollo de la Plataforma VideoWeb. Si se detectan dificultades se procederá a valorar los cambios que se imponen en la arquitectura, aplicar los mismos de forma inmediata y comenzar un nuevo proceso evaluativo empleando nuevamente el método ARID, enfatizando en los errores detectados en la anterior evaluación.

4.6.2. Estrategia de evaluación Tardía (ATAM).

En la estrategia de evaluación tardía de la AS se realizará una vez que el sistema este implementado completamente y esté puesto en funcionamiento. Se analizarán los atributos de calidad definidos en la evaluación temprana Modificabilidad, Interoperabilidad, Seguridad, Funcionalidad y Disponibilidad conjuntamente con los atributos de Mantenibilidad, Reusabilidad y Flexibilidad. Se empleara el método de evaluación ATAM que revela la forma en que la arquitectura satisface los atributos de calidad definidos y provee una visión de la forma que estos atributos interactúan entre sí.

ATAM define tres tipos de escenarios: Escenarios de casos de Usos (Use Case Scenarios) son los que describen las interacciones de los usuarios con el sistema, Escenarios de crecimiento (Growth Scenarios) son los que representa la anticipación de los cambios y los Escenarios exploratorios (Exploratory Scenarios) son los que tienen como objetivo fundamental exponer al sistema a condiciones extremas, llevar el diseño a casos extremos de condiciones. ATAM consta de nueve pasos, divididos en cuatro fases.

Presentación	
Presentación del ATAM	El líder de evaluación describe el método al resto del equipo y se establece el alcance.
Presentación de las metas de negocio	Se realiza la descripción de las metas del negocio que motivan el esfuerzo y aclara que se persiguen objetivos de tipo arquitectónico.
Presentación de la Arquitectura	El arquitecto describe la arquitectura, enfocándose en cómo esta cumple con los objetivos del negocio.
Investigación y análisis	
Identificación de los enfoques arquitectónicos.	Se detectan sin entrar en detalles.
Generación del Utility Tree.	Se priorizan los atributos de calidad que engloban las funcionalidades del sistema, especificados en forma de escenarios. Se tienen en cuenta los estímulos y respuestas, así como se establece las prioridades entre ellos.

Análisis de los enfoques arquitectónicos.	Con base en los resultados del establecimiento de prioridades del paso anterior, se analizan los elementos identificados. Se identifican riesgos arquitectónicos, puntos de sensibilidad y puntos de balance.
Pruebas	
Lluvia de ideas y establecimiento de prioridad de escenarios.	Con la colaboración de todos los involucrados, se complementa el conjunto de escenarios.
Análisis de los enfoques arquitectónicos.	Este paso repite las actividades del análisis de los enfoques arquitectónicos, haciendo uso de los resultados de la lluvia del paso anterior. Los escenarios son considerados como caso de prueba para confirmar el análisis realizado hasta el momento.
Reporte	
Presentación de los resultados	Basado en la información recolectada a lo largo de la evaluación del ATAM, se presentan los hallazgos a los participantes.

Figura 26: Fases del ATAM

De forma similar a la anterior estrategia los resultados obtenidos de la evaluación se expondrán ante todos los involucrados en la realización de la Plataforma VideoWeb. Si se detectan dificultades se procederá a valorar la factibilidad de realizar cambios en la arquitectura u otras variantes que contribuyan con la mitigación de estos problemas. De ser necesario se realizarán los cambios y al concluir el mismo se proponen realizar nuevamente la evaluación de la arquitectura empleando el método ATAM.

4.7. Conclusiones

En el presente capítulo se han analizado los elementos fundamentales relacionados con la evaluación de la Arquitectura de Software (AS), destacando las ventajas de realizar la evaluación, ya que permite identificar el impacto que tiene la arquitectura sobre los atributos de calidad definidos, posibilita detectar los problemas, debilidades y puntos de interés para corregir a tiempo los mismos. Se planteó la existencia de dos etapas para la evaluación de la arquitectura: la evaluación temprana y la evaluación tardía. Se



Plataforma VideoWeb

analizaron cuales son los resultados que puede arrojar la evaluación de la AS, entre los que se encuentra: determinar si la arquitectura es adecuada para el sistema para el cual fue diseñada o cuál de las arquitecturas propuestas es la más adecuada para el sistema. Se determinaron una serie de atributos de calidad por los que puede ser evaluada la arquitectura, entre ellos se seleccionaron para evaluar la Plataforma VideoWeb los atributos: Modificabilidad, Interoperabilidad, Seguridad, Funcionalidad, Disponibilidad, Mantenibilidad, Reusabilidad y Flexibilidad. Se abordaron características y aplicaciones de los principales métodos de evaluación de la Arquitectura de Software, como: ARD, SAAM, ARID y ATAM. Se elaboró una propuesta de evaluación de la arquitectura en etapas tempranas del desarrollo usando el método ARID, definiendo los atributos de calidad a medir para mitigar los errores que puedan aparecer, con el objetivo de mitigar el impacto de estos en el futuro. De igual forma se define una estrategia de prueba para la etapa tardía empleando el método ATAM, para medir los atributos de calidad definidos y comprobar el grado de cumplimiento de los mismos en el producto final.

CONCLUSIONES GENERALES

En el presente trabajo se analizaron los procesos asociados a la Plataforma VideoWeb. Se analizaron las funcionalidades de plataformas que brindan servicios de audio y video a través de la web, entre ellas [VideoWebPeru](#), [es.youtube.com](#), [TV.com](#) de carácter internacional e [Inter-nos](#) en nuestro país, de los cuales se pudo apreciar un conjunto de funcionalidades claves para un producto de este tipo.

Se abordaron los aspectos fundamentales relacionados con la Arquitectura de Software, analizando características de los principales estilos arquitectónicos, como flujo de datos, llamado y retorno, componentes independientes, centrados en datos y máquina virtual.

Se realizó un estudio de los principales patrones arquitectónicos y se seleccionó para la solución el patrón Modelo Vista Controlador, definiéndose sus principales componentes, configuraciones y restricciones.

Se analizaron las características de un conjunto de tecnologías y herramientas para el desarrollo de la Plataforma VideoWeb. Se analizaron herramientas de desarrollo como los CMSs Drupal, Plone y Joomla; y lenguajes de programación empleados en el desarrollo de sistemas informáticos Web como: PHP, Python y Java Server Pages (JSP). Se brindaron características de Sistemas Gestores de Bases de Datos libres como PostgreSQL y MySQL. Se proporcionaron características de la tecnología de Media Streaming y los principales servidores streaming libres como: Darwin Streaming Server, VideoLan (VLC), mpeg4ip, Feng, FFServer y MEC4.

Se seleccionaron las herramientas para el funcionamiento adecuado de la aplicación. Se escogió como metodología de desarrollo de software el Proceso Unificado (RUP), el cual brinda la documentación necesaria para la Plataforma VideoWeb, garantizando el desarrollo, soporte, mantenimiento y producción de nuevas versiones. Como sistema gestor de base de datos (SGBD) se seleccionó PostgreSQL, entre sus características se encuentra la robustez, la seguridad y la alta concurrencia de acceso a los datos; destacando la existencia de una comunidad que brinda soporte al mismo en la Universidad de las Ciencias Informáticas (UCI). Se escogió para el desarrollo de la aplicación el CMS Drupal, el cual se destaca por su flexibilidad y potencia, contando con módulos libres para aumentar sus funcionalidades y la experiencia en la Universidad de las Ciencias Informáticas en el desarrollo de aplicaciones informáticas empleando el CMS Drupal. El patrón arquitectónico Modelo Vista Controlador (MVC) se seleccionó para el desarrollo de los módulos necesarios para la Plataforma VideoWeb, ya que facilita el trabajo en equipo y



permite la flexibilidad de los componentes desarrollados. La selección se hizo completamente con herramientas de software libre, lo cual posibilitará su uso y comercialización sin restricciones por concepto de licencias de software, logrando soberanía tecnológica.

Se plantearon los requerimientos no funcionales que influyen de manera significativa en la Arquitectura de Software. Se definieron las vistas arquitectónicas (Vista de casos de uso, Vista Lógica, Vista de procesos, Vista de despliegue y Vista de Implementación) vitales para el desarrollo de la plataforma y la estructura del equipo de desarrollo.

Se analizaron elementos fundamentales relacionados con la evaluación de la Arquitectura de Software (AS). Se analizaron las ventajas de evaluar la AS, la cual permite identificar el impacto que tiene la misma sobre los atributos de calidad definidos, posibilitando detectar los problemas y debilidades. Se planteó la existencia de dos etapas para la evaluación de la arquitectura: la evaluación temprana y la evaluación tardía. Se determinaron una serie de atributos de calidad por los que puede ser evaluada la arquitectura, entre ellos se seleccionaron para evaluar la Plataforma VideoWeb los atributos: Modificabilidad, Interoperabilidad, Seguridad, Funcionalidad, Disponibilidad, Mantenibilidad, Reusabilidad y Flexibilidad. Se presentaron características y aplicaciones de los principales métodos de evaluación de la Arquitectura de Software, como: ARD, SAAM, ARID y ATAM. Se elaboró una propuesta de evaluación de la arquitectura en etapas tempranas del desarrollo usando el método ARID y una estrategia de prueba para la evaluación tardía empleando el método ATAM.

Se obtuvieron todos los artefactos necesarios para dar cumplimiento al objetivo general trazado.



RECOMENDACIONES

Realizar la evaluación de la arquitectura mediante las estrategias de evaluación definidas.

El refinamiento permanente de la arquitectura propuesta, durante el ciclo de desarrollo.

Valorar el uso de otros CMSs para el desarrollo de la aplicación con aras de ampliar el posible mercado.

BIBLIOGRAFÍA CITADA

1. **Austerberry, David.** *The Technology of Video and Audio Streaming Second Edition.* s.l. : Focal Press, 2005. ISBN 0-240-80580-1.
2. *Streaming of Continuous Media for Distance Education Systems.* **Dashti, Ali and Safar, Maytham.** 3, s.l. : International Journal of Distance Education Technologies, 2007, Vol. 5, pp. 42-66. ITJ3739.
3. **Gómez, Abel Valdés.** *SERVICIO DE MEDIA STREAMING PARA LA WEB. PORTAL INTER-NOS. MÓDULOS: TELECLASES Y TV.* Ciudad de la Habana : s.n., Abril de 2006.
4. **Aportela Rodríguez, I. M.** *Intranets: las tecnologías de información y comunicación en función de la organización.* 2007.
5. **Sarduy Domínguez, Yanetsys and Urra González, Pedro.** *Sistemas de gestión de contenidos: En busca de una plataforma ideal.* [Online] 2006. [Cited: Enero 23, 2009.] http://bvs.sld.cu/revistas/aci/vol14_4_06/aci11406.htm.
6. **Clements, Paul.** *A Survey of Architecture Description Languages.* Alemania : Proceedings of the International Workshop on , 1996.
7. **Bass, Len, Clements, Paul and Kazman, Rick.** *Software Architecture in Practice, Second Edition.* s.l. : Addison Wesley, April 11, 2003. 0-321-15495-9.
8. **Abowd, Gregory D, Garlan, David and Allen, Robert.** *Formalizing style to understand descriptions of software architecture.* USA : Pittsburgh, Computer Science Department, Carnegie Mellon University, 21 de junio, 1995.
9. **Shaw, Mary and Garlan, David.** *Formulations and Formalisms in Software Architecture.* USA : Carnegie Mellon University, Pittsburgh, 1995.
10. **Buschmann, Frank, et al.** *Pattern-Oriented Software Architecture Volume 1: A System of Patterns.* Inglaterra : s.n., 1995.

11. **Garlan, David and Shaw, Mary.** *An Introduction to Software Architecture*. USA : Pittsburgh, School of Computer Science, Carnegie Mellon University, Enero, 1994.
12. *Arquitectura y Patrones de diseño, Semana: 13, Actividad No. 25, Conferencia #8, Tema No. 3., Ingeniería de Software II*. Habana, Cuba : teleformacion.uci.cu, Curso: 2007-2008.
13. **Larman, Craig.** *UML y Patrones*. 1999.
14. **Microsoft Corporation.** *La Arquitectura Orientada a Servicios (SOA) de Microsoft*. s.l. : Microsoft Corporation, Diciembre de 2006.
15. **Alba, Julio.** *SOA: ARQUITECTURA ORIENTADA AL SERVICIO*. febrer-marzo, 2008. 0210-3923.
16. **Zaninotto, F and Potencier, F.** *Synfony la guía definitiva*. 2008.
17. **Fernández, Carlos Rafael Ballestero.** *Arquitectura de Software del Sistema de Gestión de Información Intranet 2* . Ciudad de la Habana : Universidad de las Ciencias Informáticas, Julio 2007 .
18. **Letelier Torres, Patricio Orlando and Penadés, M^a Carmen.** *MÉTODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SOFTWARE: EXTREME PROGRAMMING (XP)*. Universidad Politécnica de Valencia (UPV) : Departamento de Sistemas Informáticos y Computación , 2006. Vol. 5. ISSN 1666-1680.
19. **Beck, Kent.** *Extreme Programming Explained. Embrace Change*. s.l. : Addison-Wesley Professional, 2000. 0201616416.
20. **Jeffries, Ron, Anderson, Ann and Hendrickson, Chet.** *Extreme Programming Installed*. 2001.
21. **Barbacci, M., et al.** *Quality Attributes*. s.l. : Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, diciembre 1995.
22. **Kazman, R. and Olson, D.** *From requirements negotiation to software architectural decisions. In: Second International Workshop from Software Requirements to Architectures (STRAW '01)*, . Toronto, Canada. : s.n., 2001.



23. **610.12, IEEE Std.** *IEEE Standard Glossary of Software Engineering Terminology -Description.* 1990 .
24. **Bosch, J., et al.** *Building Application Frameworks, chapter Object-oriented frameworks - Problems & Experiences.* .
25. **Pressman, Roger.** *Ingeniería de Software, un enfoque práctico.* New York : MacGraw-Hill, 2003.
26. **Collins-Sussman, Ben, W. Fitzpatrick, Brian and Pilato, C. Michael.** *Version Control with Subversion For Subversion 1.1.* Copyright © 2002, 2003, 2004, 2005 .
27. **Kazman, Rick, Klein, Mark and Clements, Paul.** *ATAM:Method for Architecture Evaluation.* s.l. : Pittsburgh, August 2000.
28. **Gómez, Omar Salvador.** *Evaluando Arquitecturas de Software. Parte 2. Métodos de Evaluación.* México : Brainworx S.A, 2007.

BIBLIOGRAFÍA CONSULTADA

Meléndrez, Edelsys Hernández. *Cómo escribir una tesis* . Cuba : Escuela Nacional de Salud Pública , 2006.

Paneda, Xabiel G., et al. *A Methodology for Performance, Content Analysis, and Configuration of Audio/Video-on-Demand Services*. España : International Journal of Business Data Communications and Networkin, 2007.

Coplien, James O. *Idioms and Patterns as Architectural Literature*. s.l. : IEEE Software, Enero/febrero 1997. Vol. 14(1).

Reynoso, Carlos and Kicillof, Nicolás. *De Lenguajes de descripción arquitectónica de Software (ADL), version 1*. Argentina : Universidad de Buenos Aires, Marzo, 2004.

Medvidovic, Nenad and Taylor, Richard N. *A classification and comparison framework for software Architecture Description Languages*. s.l. : Technical Report UCI-ICS-97-02, 1996.

Zimmermann, Roger, Fu, Kun and Desai, Dwipal A. HYDRA: High-performance Data Recording Architecture for Streaming Media. *Video Data Management and Information Retrieval*. 2005.

Alvarez, Miguel Angel. Descubre la tecnología que nos acerca hacia una Internet de radio y televisión. *DesarrolloWeb*. [Online] Julio 9, 2001. [Cited: Diciembre 3, 2008.] <http://www.desarrolloweb.com/articulos/482.php>.

Álvarez de Zayas, C. *Metodología de la Investigación Científica*. Stgo. de Cuba : Universidad de Oriente, 1995.

Bayarre, H and Hersford, R. *Metodología de la Investigación*. Ciudad de La Habana : Ciencias Médicas, 2004.

Putting Video on Your Web Site: The Basics. **Bremser, Wayne.** 2, 1996, Web Developer, Vol. II.

Bremser, Wayne. Putting Video on Your Web Site: The Basics. *Webdeveloper*. [Online] 1996. [Cited: Enero 10, 2009.] http://www.webdeveloper.com/multimedia/multimedia_putting_video_on_website.html.

Caraballo Pérez, Yeter. La gestión de contenidos en portales Web. [Online] 2007. [Cited: Enero 9, 2009.] http://bvs.sld.cu/revistas/aci/vol15_3_07/aci07307.htm.

Voces Merayo, Ramón. El contenido audiovisual: otro reto para la accesibilidad web. [Online] 2008. [Cited: Enero 23, 2009.] <http://www.ub.edu/bid/21/voces2.htm>. ISSN 1575-5886.

González, Aleksander, et al. *Método de Evaluación de Arquitecturas de Software Basadas en Componentes (MECABIC)*. s.l. : Mexico : s.n., 2005, Vol. vol 1.

Clements, Paul, Kazman, Rick and Klein, Mark. *Evaluating software architectures: methods and case studies*. s.l. : Addison-Wesley, 2002.

Plataforma Video Web

ANEXOS

Anexo 1



Figura 27: TV.com

Anexo 2

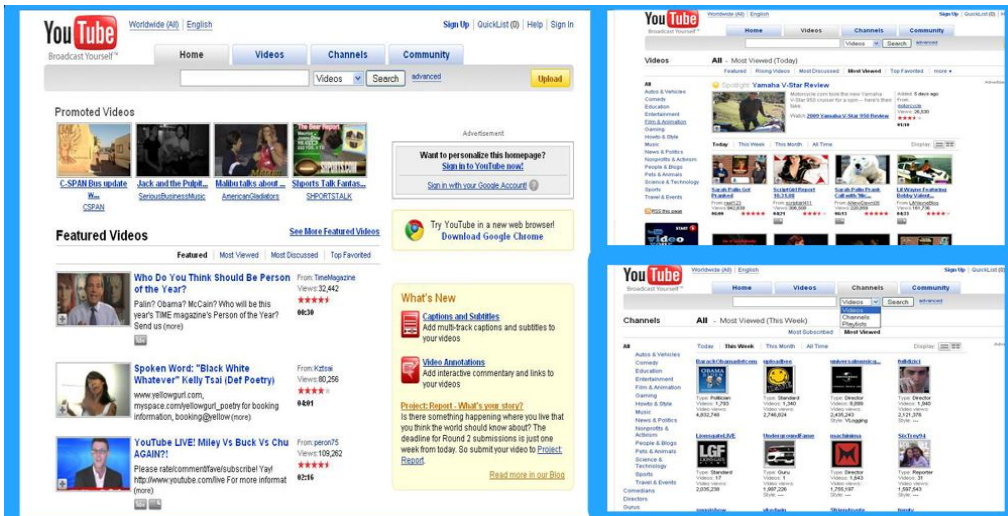


Figura 28: es.youtube.com

Plataforma Video Web

Anexo 3



Figura 29: Video Web Perú

Anexo 4



Figura 30: Inter-Nos.

Anexo 5

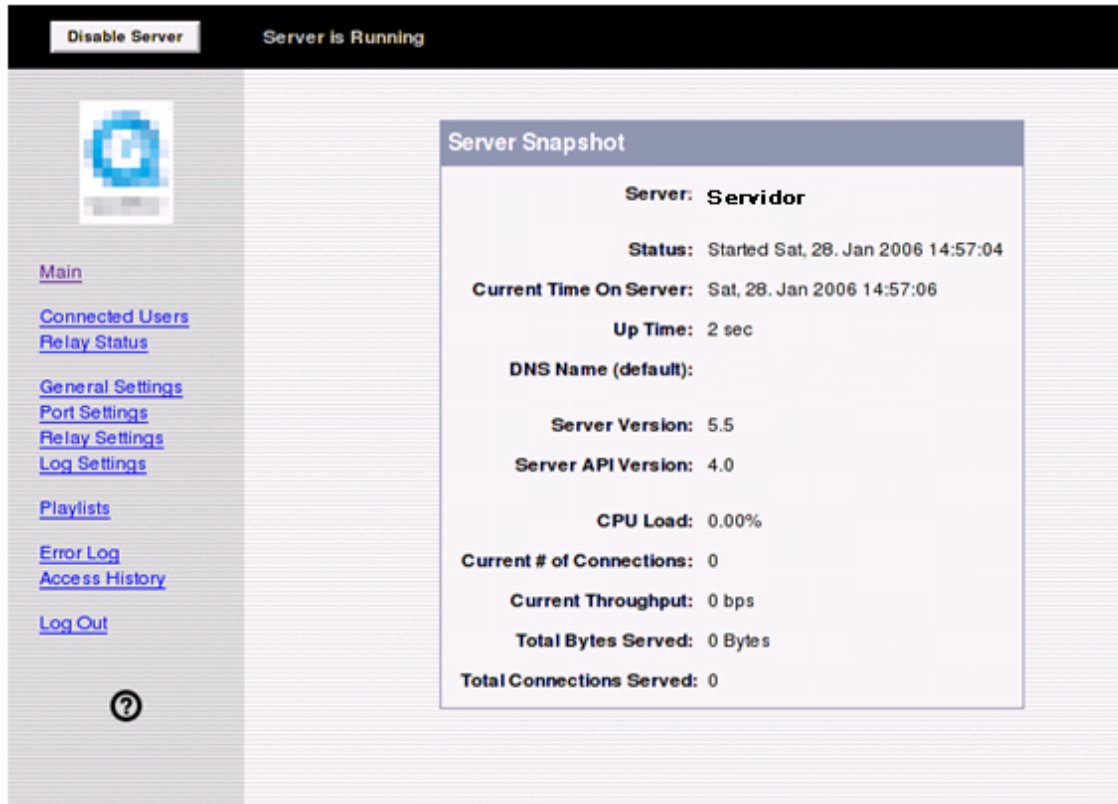


Figura 31: Administración del Darwin Streaming Server

Anexo 6

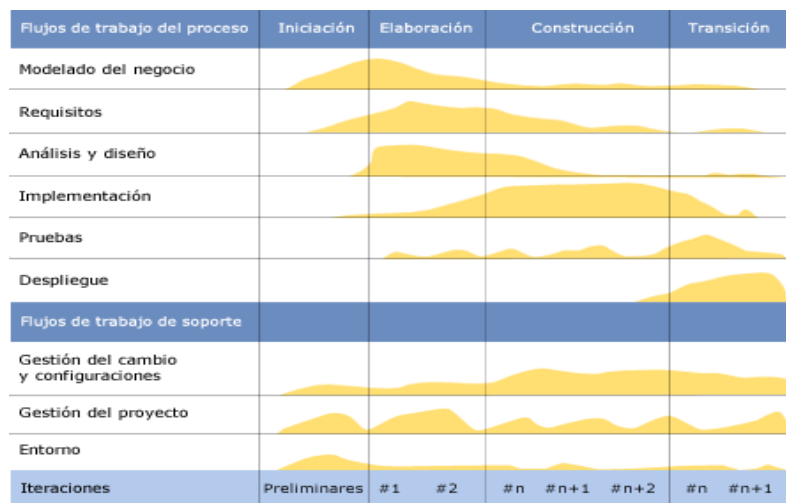


Figura 32 RUP en dos dimensiones.

Anexo 7

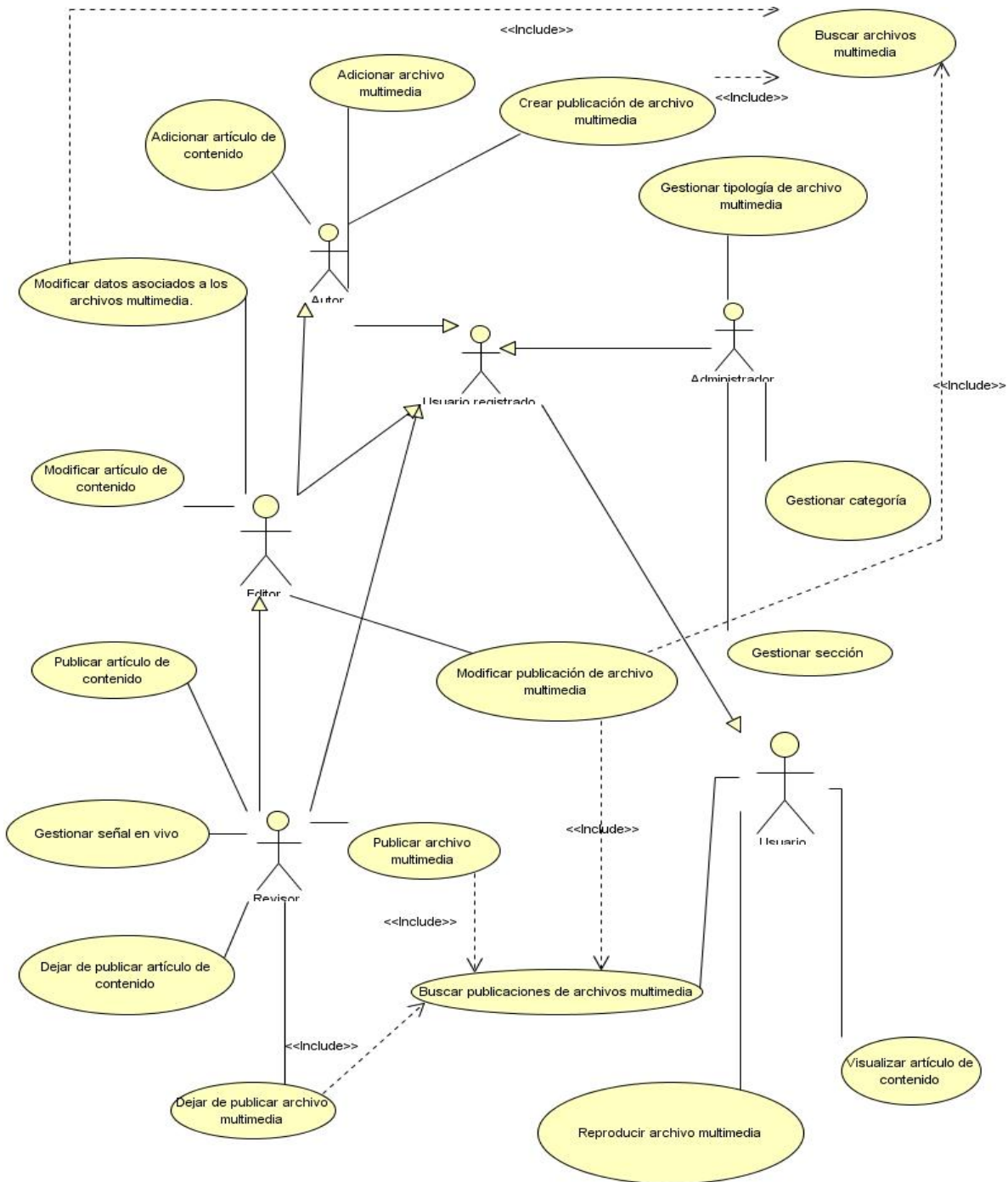


Figura 33: Vista de Caso de Uso del Módulo de Gestión de Contenido.

Anexo 8

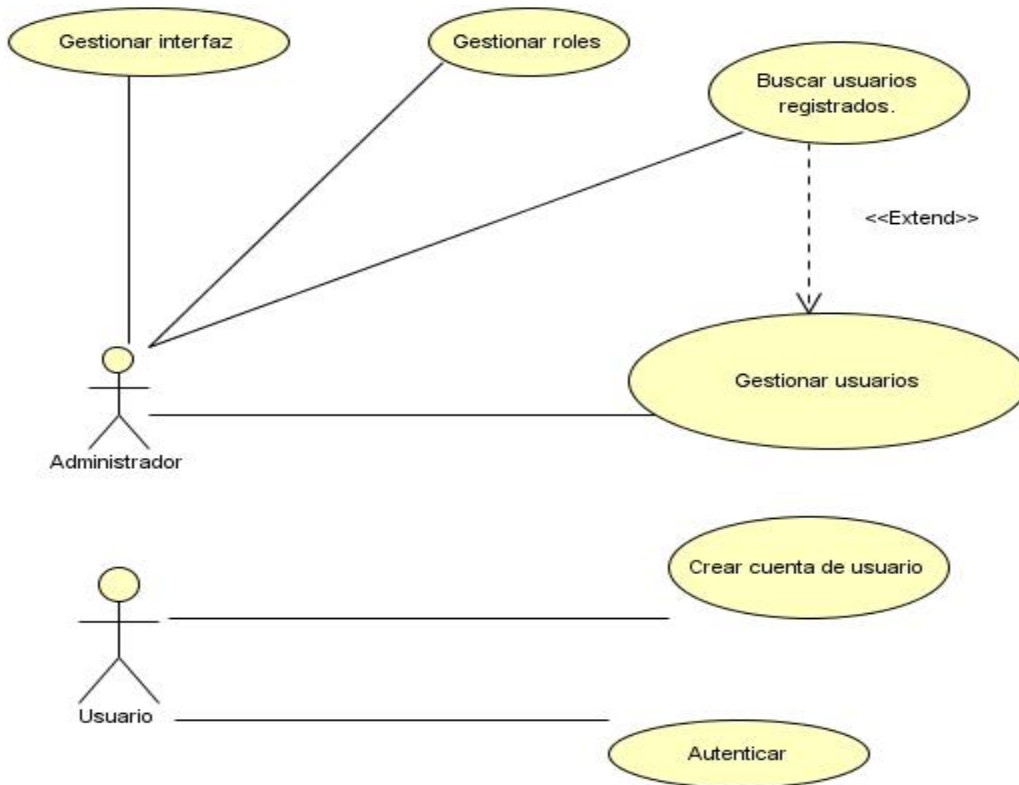


Figura 34: Vista de Caso de Uso del Módulo de Administración.

Anexo 9

Nombre de caso de uso	Breve descripción
Reproducir archivo multimedia	Este caso de uso tiene lugar cuando el usuario acciona un vínculo a un archivo multimedia para reproducirlo.
Visualizar artículo de contenido	Este caso de uso tiene lugar cuando el usuario acciona un vínculo a un artículo de contenido.
Buscar publicaciones de archivos multimedia	Este caso de uso tiene lugar cuando el usuario realiza la búsqueda de archivos multimedia publicado dado un

	criterio determinado.
Adicionar artículo de contenido	Este caso de uso tiene lugar cuando el autor crea un nuevo artículo de contenido y la adiciona.
Adicionar archivo multimedia	Este caso de uso tiene lugar cuando el autor adiciona archivos multimedia.
Buscar archivos multimedia	Este caso de uso tiene lugar cuando algún usuario con permiso realiza la búsqueda de archivos multimedia dado un criterio determinado en el momento de adicionar, modificar o publicar archivos en el servidor de media.
Buscar usuarios registrados	Este caso de uso tiene lugar cuando el administrador busca usuarios registrados dado algún criterio.
Crear cuenta de usuario	Este caso de uso tiene lugar cuando el usuario realiza la búsqueda de archivos multimedia dado un criterio determinado.
Crear publicación de archivo multimedia	Este caso de uso tiene lugar cuando el autor crea una publicación de archivo multimedia seleccionando un material del servidor de medias
Dejar de publicar artículo de contenido	Este caso de uso tiene lugar cuando el revisor termina una publicación de artículo de contenido.
Dejar de publicar archivo multimedia	Este caso de uso tiene lugar cuando el revisor termina una publicación de archivo multimedia.
Gestionar señal en vivo	Este caso de uso tiene lugar cuando el revisor gestiona archivos de configuración de transmisión de señal en vivo
Gestionar categoría	Este caso de uso tiene lugar cuando el administrador gestiona las categorías del sistema.
Gestionar sección	Este caso de uso tiene lugar cuando el administrador gestiona las secciones del sistema.
Gestionar interfaz	Este caso de uso tiene lugar cuando el administrador

	gestiona la interfaz del sistema.
Gestionar Roles	Este caso de uso tiene lugar cuando el administrador gestiona los roles del sistema.
Gestionar usuario	Este caso de uso tiene lugar cuando el administrador gestiona los usuarios del sistema.
Gestionar tipología de archivo multimedia	Este caso de uso tiene lugar cuando el administrador gestiona los archivos multimedia del sistema.
Modificar artículo de contenido	Este caso de uso tiene lugar cuando el editor selecciona un artículo de contenido para modificarla.
Modificar datos asociados a los archivos multimedia	Este caso de uso tiene lugar cuando el editor selecciona un archivo multimedia para modificarlo.
Modificar publicación de archivo multimedia	Este caso de uso tiene lugar cuando el editor selecciona una publicación archivo multimedia para modificarla
Publicar archivo multimedia	Este caso de uso tiene lugar cuando el revisor selecciona una publicación de archivo multimedia y aceptarla
Publicar artículo de contenido	Este caso de uso tiene lugar cuando el revisor selecciona un artículo de contenido para publicarla
Autenticar	Este caso de uso tiene lugar cuando el usuario desea registrarse para acceder con sus permisos al sistema

Figura 35: Breve descripción de los casos de usos arquitectónicamente significativos.

Anexo 10

Plataforma VideoWeb

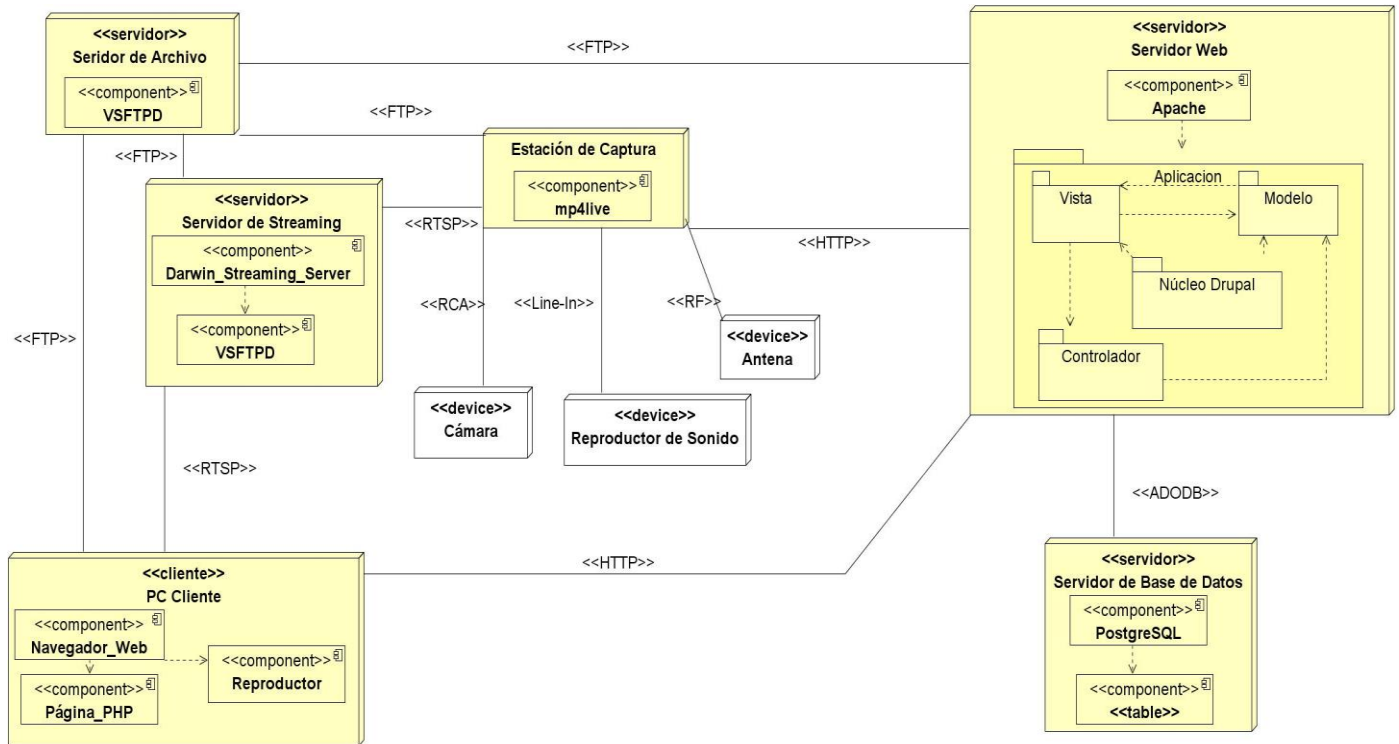


Figura 36: Distribución física alternativa

Anexo 11

Plataforma VideoWeb

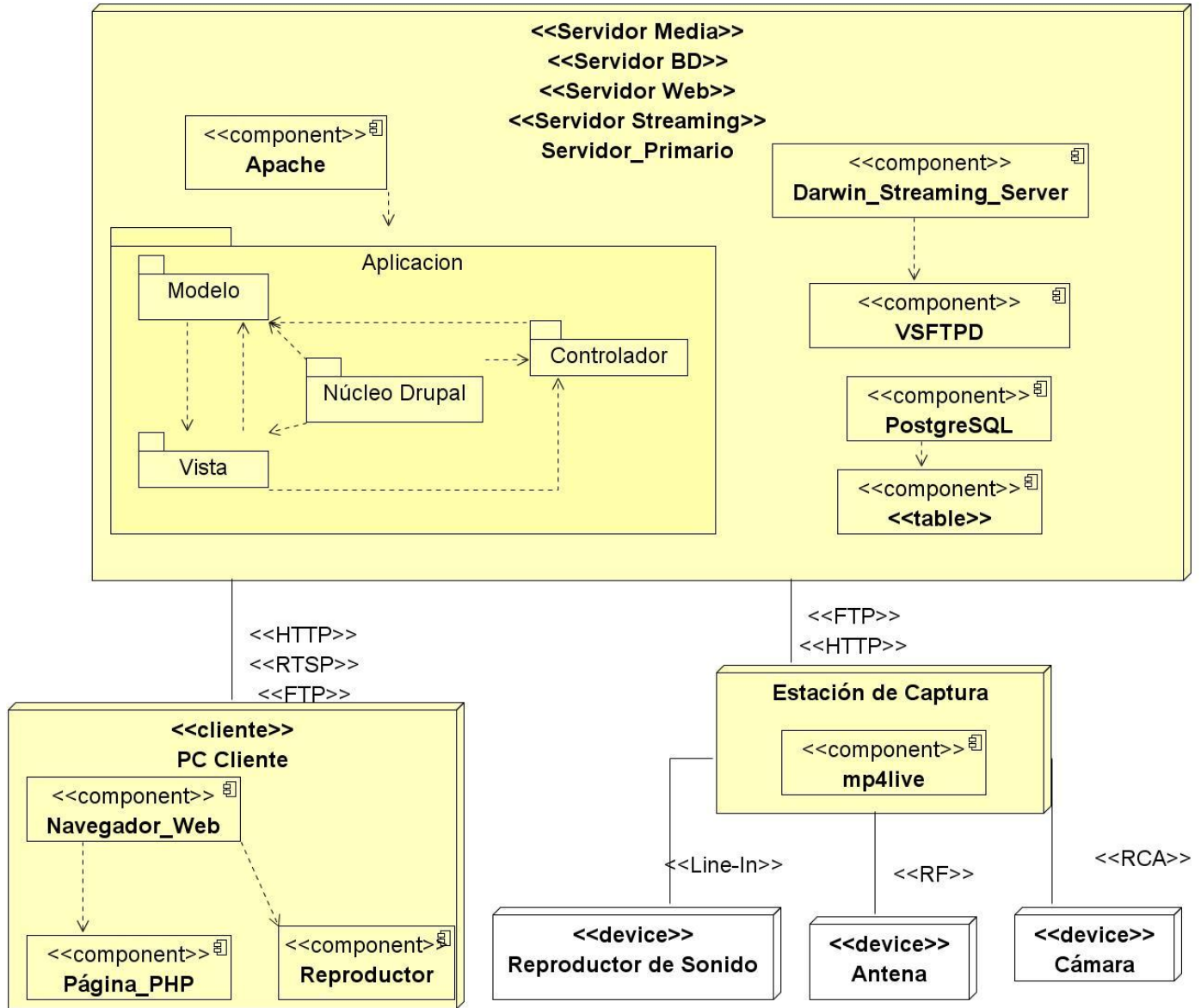


Figura 36: Distribución física alternativa II

Anexo 12

Comparación de Velocidades entre Drupal 6.10 y Joomla 1.5.6

Se realizaron pruebas de estrés a dos sitios usando el CMS Joomla 1.5.6 y Drupal 6.10 respectivamente, alojados en el mismo servidor, conectándose simultáneamente iguales cantidades de usuarios. Los resultados de las pruebas se exponen a continuación.

Joomla 1.5.6

URL#	Nombre	Tiempo entre Click [s]	URL	Datos enviados por POST	Usuario	Contraseña
1	Test	10	http://10.7.3.79:5801/Joomla_1_5/			

Figura 37: Prueba usando CMS Joomla 1.5.

Usuario No.	Clicks	Realizados	Errores	Tiempo promedio de Respuesta [ms]	Bytes	kbit/s
1	10	10	0	7.415	341.104	36,80
2	10	10	0	4.693	341.485	58,21
3	10	10	0	6.150	341.155	44,38
4	10	10	0	6.955	341.311	39,26
5	10	10	0	6.733	340.989	40,52
6	10	10	0	4.961	340.946	54,98
7	10	10	0	8.283	341.091	32,95
8	10	10	0	6.999	341.401	39,02
9	10	10	0	6.034	341.073	45,22
10	10	10	0	5.279	341.369	51,73

Figura 38: Resultado por Usuario

URL No.	Nombre	Clicks	Errores	Errores [%]	Tiempo Usado [ms]	Promedio de Clicks[ms]
1	Tes	99	0	0,00	633.376	6.398

Figura 39: Resultados Generales

Drupal 6.10

URL#	Nombre	Tiempo entre Click [s]	URL	Datos enviados por POST	Usuario	Contraseña
1		10	http://10.7.3.79:5801/drupal_6_10/			

Figura 40: Prueba usando CMS Drupal 6.10.

Usuario No.	Clicks	Realizados	Errores	Tiempo promedio de Respuesta [ms]	Bytes	kbit/s
1	10	10	0	4.944	131.820	21,33
2	10	10	0	4.864	131.820	21,68
3	10	10	0	4.952	131.820	21,30
4	10	10	0	4.528	131.820	23,29
5	10	10	0	6.054	131.820	17,42
6	10	10	0	6.115	131.820	17,24
7	10	10	0	4.901	131.820	21,52
8	10	10	0	4.732	131.820	22,29
9	10	10	0	4.232	131.820	24,92
10	10	10	0	4.799	131.820	21,98

Figura 41: Resultado por Usuario

URL No.	Nombre	Clicks	Errores	Errores [%]	Tiempo Usado [ms]	Promedio de Clicks[ms]
1		97	0	0,00	497.428	5.128

Figura 42: Resultados Generales

Conclusiones

Mediante los resultados anteriores se puede apreciar que el tiempo (en milisegundos) usado en responder las peticiones de los usuarios por el CMS Drupal 6.10 es significativamente menor al del empleado por el CMS Joomla 1.5.6 (497.428 y 633.376 respectivamente), demostrando que Drupal 6.10 es más rápido que Joomla 1.5.6.

Anexo 13

Validación de la propuesta de Hardware

Se realizó el análisis de las características de cada componente físico que interviene en la aplicación y se estimaron los requerimientos de hardware necesario para el funcionamiento de la plataforma.

Servidor de archivos multimedia

Este servidor tiene la responsabilidad de almacenar los archivos multimedia. La interacción con el resto del sistema es mediante el servidor FTP VSFTPD, el nivel de concurrencia de usuarios no es muy alto y se limitará el flujo de transferencias de datos a 1 MBit/s. El consumo de los recursos se centra fundamentalmente en la Capacidad de Almacenamiento. El formato de almacenamiento de videos es H.264, 500 kbit/s, esto implica que un fichero de 1 hora ocupa 225 Mb. En base a lo cual se realizó el estimado de recurso.

Servidor de Aplicaciones

Este servidor tiene la responsabilidad de brindar los principales servicios de la aplicación: Servidor de Streaming (Darwin Streaming Server), Servidor de Base de Datos (PostgreSQL) y servidor Web (Apache). Los recursos críticos son: Velocidad de Procesamiento del CPU, Memoria RAM, Almacenamiento y velocidad de acceso a datos del Disco Duro, y Acho de banda de la Tarjeta de Red. El servidor de

streaming es el que más recursos requieren, principalmente velocidad de acceso a datos del disco duro, CPU y memoria RAM. A continuación se exponen los resultados de las pruebas realizadas:

N# conexiones concurrentes	Uso de CPU (Intel Core 2 Duo E4500 2.2 GHz)	Uso de RAM (1 GB) – DDR2	SO	Disco Duro
20	1.5 %	9 Mb	Ubuntu 8.04	SATA2

Figura 43: Rendimiento del servidor de streaming.

Observaciones

El servidor se muestra estable, manteniendo el uso de CPU alrededor del 1.5% con 20 conexiones. El uso de memoria RAM aumenta proporcionalmente a la cantidad de conexiones concurrentes, a razón de 1 Mb cada 2 conexiones aproximadamente. Los contenidos audiovisuales se transmiten a 500 kbits/s, si la red no soporta transmisión multicast, cada conexión ocuparía 500 kbits de ancho de banda, permitiendo la conexión de 2000 usuarios concurrentes con una tarjeta Gigabit Ethernet.

Estación de captura

La estación de captura tiene la responsabilidad de realizar la transmisión en vivo de los contenidos capturados mediante los dispositivos de entrada de Audio y Video. También puede capturar y almacenar en estos contenidos para enviarlo al servidor de archivos multimedia. El consumo de recursos se centra en la Velocidad de Procesamiento del CPU para la codificación y trasmisión del contenido a través de media streaming. Solo enviará el streaming al servidor primario, el cual se encargara de difundirlo a los usuarios.

PC cliente

La PC cliente consumirá los servicios de la aplicación, para lo cual necesita un navegador web y un reproductor que soporte el protocolo RTSP, por lo cual los requerimientos de hardware no constituyen un factor crítico.

Anexo 14

GLOSARIO DE TÉRMINOS

--A--

AAC: Codificación de audio avanzada (del inglés Advanced Audio Coding), es un formato de audio digital.

Abstracción: Abstracción consiste en aislar un elemento de su contexto o del resto de los elementos que lo acompañan

Ancho de banda: Cantidad de datos que se pueden transmitir en una unidad de tiempo través de la red.

API: Interfaz de programación de aplicaciones o API (del inglés Application Programming).

Archivos multimedia: Archivos de audio, video o imagen.

--B--

Bitrates: El número de bits que son transmitidos o procesados por unidad de tiempo.

Blogs: Sitio web que recopila cronológicamente textos o artículos de uno o varios autores donde este conserva siempre la libertad de dejar publicado lo que crea pertinente.

Buffer: Área de la memoria que se utiliza para almacenar datos de forma temporal.

Buscador: Sistema informático que indexa archivos almacenados en servidores web.

--C--

Captura: Proceso por el cual se obtiene información de un dispositivo externo (Cámara, webcam, etc.) y se almacena o se observa en la computadora.

CASE: Por sus siglas en ingles Ingeniería de Software Asistida por Ordenador (Computer Aided Software Engineering).

CMS: Por sus siglas en ingles Sistema Gestor de Contenido (Content Management System).

Códec: Es una abreviatura de Compresor-Decompresor, permiten codificar el flujo o la señal y recuperarlo o descifrarlo del mismo modo para la reproducción o la manipulación en un formato más apropiado para estas operaciones.

Codificación: Convertir a un formato de video determinado.

--D--

Demultiplexador: Software que recibe una archivo de video y separa la pista de audio y la de video en archivos independientes.

Dispositivo de Captura: Dispositivo capaz de capturar una señal externa de audio y video para ser transmitida en vivo por la plataforma web para la publicación y gestión de contenidos audiovisuales o para ser almacenada como archivo multimedia en el servidor de archivos multimedia.

Distribuciones: Variante de un sistema operativo que incorpora determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones

hogareñas, empresariales y para servidores.

DVD: Disco de Video Digital del inglés Digital Video Disc.

--E--

Escenario: Es una descripción resumida de funcionalidades deseadas por el sistema, o cambios que podrían sufrir este en el futuro.

--F--

Framework: Marco de trabajo, generalmente brinda funcionalidades para agilizar el desarrollo de programas informáticos.

FTP: Protocolo de Transferencia de Archivos del inglés File Transfer Protocol.

--G--

GNU: Licencia Pública General de GNU o más conocida por su nombre en inglés GNU (General Public License).

GPL: La Licencia Pública General de GNU o más conocida por su nombre en inglés GPL (General Public License).

--H--

Hardware: parte física de un computador y más ampliamente de cualquier dispositivo electrónico

HTML: Siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto).

--I--

Internet Engineering Task Force (IETF): Institución formada básicamente por técnicos en Internet e informática cuya misión es velar porque la arquitectura de la red y los protocolos técnicos que unen a millones de usuarios de todo el mundo funcionen correctamente.

Interfaz: Zona de contacto o conexión entre dos componentes de "hardware"; entre dos aplicaciones, o entre un usuario y una aplicación. Apariencia externa de una aplicación informática.

Internet: Conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

Interoperables: Capacidad de los sistemas de información, y por ende de los procedimientos a los que éstos dan soporte, de compartir datos y posibilitar el intercambio de información y conocimiento entre ellos.

--M--

MAC: En redes de computadoras la dirección MAC (Media Access Control address o dirección de control de acceso al medio).

Modular: La modularidad es la capacidad que tiene un sistema de funcionar con la unión de varias partes que interactúan entre sí y que trabajan para alcanzar un objetivo común. Cada una de esas partes en que se encuentre dividido el sistema recibe el nombre de módulo. Idealmente un módulo debe poder cumplir las condiciones de caja negra, es decir, ser independiente del resto de los módulos y comunicarse con ellos (con todos o sólo con una parte) a través de unas entradas y salidas bien definidas.

Monolíticas: Son aquellas en las que el software se estructura en grupos funcionales muy acoplados, involucrando los aspectos referidos a la presentación, procesamiento y almacenamiento de la información.

Multicast o Multidifusión: Es el envío de la información en una red a múltiples destinos simultáneamente, usando la estrategia más eficiente para el envío de los mensajes sobre cada enlace de la red sólo una vez, creando copias cuando los enlaces en los destinos se dividen.

Multiplexador: Software que une archivos de audio y video en un único archivo de video

--N--

Navegabilidad: La navegabilidad o navegabilidad web es la facilidad con la que un usuario puede desplazarse por todas las páginas que componen un sitio web.

--O--

ODBC: Open Database Connectivity (ODBC) es un estándar de acceso a Bases de datos desarrollado por Microsoft Corporation.

--P--

PC: Por sus siglas en ingles Computador personal (Personal Computer)

--R--

Reproductor: Software empleado para la reproducción de audio y video.

Reproductor Embebido: Reproductor que se incluye dentro del contenido de una página web.

Reusabilidad: La capacidad de los programas, estructuras, clases, objetos y otros componentes del sistema de poder ser usados en distintos contextos.

RTSP: 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.



Plataforma VideoWeb

RTP: Es la abreviación de Real-time Transport Protocol, por su denominación en inglés. Es un estándar creado por la IETF para la transmisión confiable de voz y video a través de Internet.

RTCP: Abreviación de Real Time Control Protocol. Es utilizado para enviar datos de control entre el emisor y receptor de una secuencia RTP.

--S--

SDP: Protocolo de descripción de sesión o SDP (Session Description Protocol), es un formato de descripción de parámetros de inicialización de streaming de contenidos audiovisuales.

Seriales de televisión: Programa de televisión que se entrega de forma periódica. Presenta dos características principales: por un lado, sus episodios no poseen unidad argumental en sí mismos, sino que las tramas quedan abiertas para los capítulos siguientes; y, por otro, la producción arranca sin un calendario previsto de cierre.

Servicios de video y audio: Se refiere a transmitir video o audio de de dos formas fundamentalmente: transmisión en vivo de canales televisivos o videos; publicación de archivos multimedia para ser visto en el momento deseado.

Software: Conjunto de programas, documentos, procesamientos y rutinas asociadas con la operación de un sistema de computadoras, es decir, la parte intangible o lógica de una computadora.

SQL: El Lenguaje de consulta estructurado o SQL (Structured Query Language), es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

Stakeholders: Quienes pueden afectar o son afectados por las actividades de una empresa o negocio. Partes interesadas.

Streaming: Tecnología que permite la reproducción de sonido o vídeo sin que sea necesario descargar previamente todo el archivo de recurso.

--T--

Transcodificar: Conversión directa (de digital a digital) de un códec a otro.

Tupla: Una **ocurrencia de artículo** o **tupla** consiste en un grupo de ocurrencias de campos relacionados, representando una acción entre ellos.

--U--

Unicast o Unidifusión: Es el envío de información desde un único emisor a un único receptor.

--V--



Plataforma VideoWeb

Virtual: Elemento que tiene existencia aparente y no real. Realidad virtual es la simulación con computadora de todo el ambiente que rodea a una persona.

--W--

Web: El término se utiliza para definir el universo del World Wide Web, los sitios, la información y los servicios de la “teleraña”.

--X--

XML: siglas en inglés de Extensible Markup Language (Lenguaje de Etiquetado Extensible).