



# **Propuesta de Proyecto Final**

## **Para optar al grado de**

## **Ingeniero en Informática**

**Profesor Guía:** Dahianna Vega L.

**Alumnos:** Rogelio Elías  
Rodrigo Riquelme  
Manuel Canales

**Fecha:** Noviembre 2010

## 1. ANTECEDENTES.

<b>1.Título del Proyecto</b>	<b>“Marco de Trabajo para un Acceso Multimedia Universal Mediante Patrones Modelo-Vista-Controlador”</b>
<b>2.Nombre de los Alumnos</b>	Rogelio Elias Rodrigo Riquelme Manuel Canales
<b>3. RUT</b>	10.096.280-2 13.272.164-5 14.124.827-8
<b>4. Dirección</b>	Vicuña Mackenna 1962, Nuñoa
<b>5. E-Mail</b>	<a href="mailto:Rogelio.elias@sonda.com">Rogelio.elias@sonda.com</a> <a href="mailto:rodrigo.riquelme@latercera.com">rodrigo.riquelme@latercera.com</a> <a href="mailto:mcanalesaraneda@yahoo.es">mcanalesaraneda@yahoo.es</a>
<b>6.Profesor Guía Propuesto</b>	Dahianna Vega L

---

**FIRMA**

## Índice General

---

<b>Capítulo 1. Introducción .....</b>	<b>12</b>
Resumen.....	12
1.1. Formulación General del Proyecto.....	15
1.2. Objetivos .....	18
1.2.1.Objetivo General .....	18
1.2.1.Objetivos Específicos.....	18
1.3.Metodología a Emplear para Desarrollar el Proyecto.....	19
1.4.Planificación Inicial.....	20
<b>Capítulo 2. Marco Teórico.....</b>	<b>22</b>
2.1. Acceso Multimedia Universal .....	22
2.2. Protocolo XML orientado a objetos .....	26
2.2.1. SOAP.....	26
2.2.2. REST .....	28
2.2.3. RSS .....	29

2.2.4. XML Orientado a MVC .....	30
2.3.1. Servidor Web .....	31
2.3.2. Stream .....	32
2.3.2.1. HTTP Delivery.....	32
2.3.2.2. Streaming .....	33
2.3.2.3. Media Streaming.....	34
2.4. Codecs de Video.....	37
2.4.1. H263 Sorenson.....	38
2.4.2. H264 Mpeg-4 Parte 10.....	38
2.4.4. OGG Theora .....	39
2.4.5. MPEG-4 .....	39
2.4.6. WMV .....	40
2.5. Tecnologías Clientes.....	41
2.5.1. Real Media Player.....	42
2.5.2. Windows Media Player .....	43
2.5.3.Quicktime Player.....	44
2.5.4. Adobe Flash.....	45

2.5.5. Video HTML5 .....	48
2.6. Conversión de Videos .....	49
2.6.1. FFmpeg .....	49
2.7. IPTV .....	51
2.8. Metodología de Desarrollo .....	53
2.8.1. Extreme Programming .....	54
2.8.2. Scrum .....	57
2.8.3. Software Libre .....	59
2.8.3.1. Licencia GNU GPL v2 .....	63
2.9. Frameworks .....	64
2.9.1. Zend Framework .....	65
2.9.2. Google Web Toolkit .....	66
<b>Capítulo 3: Estado del Arte.....</b>	<b>67</b>
3.1. Gestores de Contenidos multimedia existentes .....	67
3.1.1. PHPMotion .....	67
3.1.2. OsTube .....	69

3.2. Sitios de contenidos multimedia de referencia .....	70
3.2.1. Youtube .....	70
3.2.2. Google Video .....	71
3.2.3. Vimeo.....	74
3.2.4.TerraTV.....	75
3.2.6. 3TV .....	77
3.3. Google TV .....	78
<b>4. Desarrollo.....</b>	<b>80</b>
4.1. Toma de requerimientos .....	80
4.1.1. Requerimientos Funcionales .....	80
4.1.2. Requerimientos No Funcionales .....	81
4.2. Tecnología a Utilizar .....	82
4.2.1. Lado Servidor .....	82
4.2.1.1. PHP 5.3 .....	82
4.2.1.2. MySQL 5.....	84
4.2.1.3. FFmpeg .....	85

4.2.2. Lado Cliente.....	86
4.2.2.1 Javascript.....	86
4.2.2.2 JW Player .....	88
4.3. Entorno de Desarrollo .....	89
4.3.1. Entorno Integrado de Desarrollo (IDE).....	89
4.3.2. Control de versiones .....	90
4.3. Diagrama de Datos .....	91
4.4. Diagrama de Clases.....	92
4.4.1. Namespace Models .....	93
4.4.2. Namespace Views .....	97
4.4.3. Namespace Controllers .....	99
4.5. Especificaciones de desarrollo Back Office .....	106
4.5.1. Configuración de Sitio.....	106
4.5.2. Componentes XML .....	107
4.6. Especificaciones Front Office.....	108
4.7. Maquetas Funcionales Back Office.....	112

<b>5. Bibliografía.....</b>	<b>120</b>
<b>Glosario.....</b>	<b>122</b>
<b>Acrónimos.....</b>	<b>123</b>

## Tabla de Ilustraciones

Ilustración 1 - Componentes que intervienen en acceso multimedia web .....	14
Ilustración 2 - Adaptación de contenidos para un acceso universal .....	25
Ilustración 3 - Esquema SOAP según la W3C .....	27
Ilustración 4 - Esquema del funcionamiento de RSS .....	29
Ilustración 5 - Esquema de XML Orientado a MVC .....	31
Ilustración 6 - Modelo típico de un servicio streaming.....	36
Ilustración 7 - Logotipos de reproductores comerciales .....	41
Ilustración 8 - Real Player 11 .....	42
Ilustración 9 - Presentación de Windows Media Center en Windows 7.....	43
Ilustración 10 - Reproductor Quicktime 7 .....	44
Ilustración 11- JW Player .....	46
Ilustración 12 - Esquema de componentes de FFmpeg .....	50
Ilustración 13 - Infraestructura de redes IPTV .....	52
Ilustración 14 - Visión general Zend Framework .....	65
Ilustración 15 - Esquema de Widgets GWT.....	66
Ilustración 16 - Web PHPMotion .....	68
Ilustración 17 - OSTube .....	69
Ilustración 18 - Youtube .....	71
Ilustración 19 - Google Video .....	72
Ilustración 20 - Vimeo.....	74
Ilustración 21 - Terra TV .....	75

Ilustración 22 - Emol TV .....	76
Ilustración 23 - 3TV .....	77
Ilustración 24 – Google TV en un televisor IPTV conectado a internet .....	79
Ilustración 25 - Estructura Clases PHP del Core del CMS .....	83
Ilustración 26 - Estructura de carpetas javascript.....	87
Ilustración 27 - Zend Studio en Ubuntu Linux.....	89
Ilustración 28 - Estructura de repositorio Subversion vista en Zend Studio .....	90
Ilustración 29 - Diagrama de datos primera fase de desarrollo .....	91
Ilustración 31 - Namespace Models - Parte 2.....	96
Ilustración 32 - Namespace Views .....	98
Ilustración 33 - Namespace Controllers .....	99
Ilustración 34 - Namespace Admin.....	101
Ilustración 35 - Namespace Lib - Parte 1 .....	103
Ilustración 36 - Namespace Lib - Parte 2 .....	104
Ilustración 37 - Namespace Lib - Parte 3 .....	105
Ilustración 38 - Ingreso al Back Office.....	112
Ilustración 39 - Menú Principal .....	113
Ilustración 40 - Configuración del Servidor .....	114
Ilustración 41 - Configuración del Sitio .....	<b>¡Error! Marcador no definido.</b>
Ilustración 42 - Contenido Menú.....	115
Ilustración 43 - Contenido Páginas.....	116
Ilustración 44 - Videos .....	117
Ilustración 45 - Tipos de Videos .....	118

Ilustración 46 - Miniaturas ..... 119

## Capítulo 1. Introducción

---

### Resumen

Durante los últimos años ha existido un gran incremento de la oferta y demanda de material audiovisual disponible en Internet, en gran parte debido a las mejores condiciones de acceso e incremento en los anchos de banda. Junto con esto ha aumentado la variedad de dispositivos que son usados para acceder a este material, pero muchas veces este contenido no es compatible con el dispositivo con que se accede, ya que la forma para mostrarlo no ha sido diseñada para éste.

Como una forma de enfrentar esta problemática ha surgido en forma incipiente en el sector de las telecomunicaciones el concepto de **UMA** o **Acceso Multimedia Universal**.

El Acceso Multimedia Universal se enfoca a una línea de investigación en el sector del área de la multimedia, que busca una solución universal a la problemática nacida por el vertiginoso acrecentamiento del contenido audiovisual disponible para la gran mayoría de la población nacional y mundial. Además de la disparidad de redes de acceso y terminales del mercado.

Hoy existen además temas de restricciones y patentes de tecnologías de reproducción de tecnología multimedia, esto debido en gran medida al lanzamiento al mercado de diferentes tipos de dispositivos portátiles con capacidades de navegar por la Internet y acceder a contenidos audiovisuales.

Algunas empresas involucradas en estas nuevas tecnologías tienen como parte de su política de negocios hacer soluciones privativas por lo que no les interesa fomentar la compatibilidad.

Un conjunto de circunstancias y factores plantean la necesidad de tecnología UMA:

- Gran cantidad de contenido audiovisual.
- Difícil acceso a la información sin etiquetar.
- Condiciones de acceso a la red diferentes y variables.
- Heterogeneidad de dispositivos cliente.
- Exigencias del usuario, calidad insatisfactoria para tecnología cliente.
- Altos costos de mantenimiento.

En el siguiente diagrama se encuentra un esquema de un Sistema UMA.

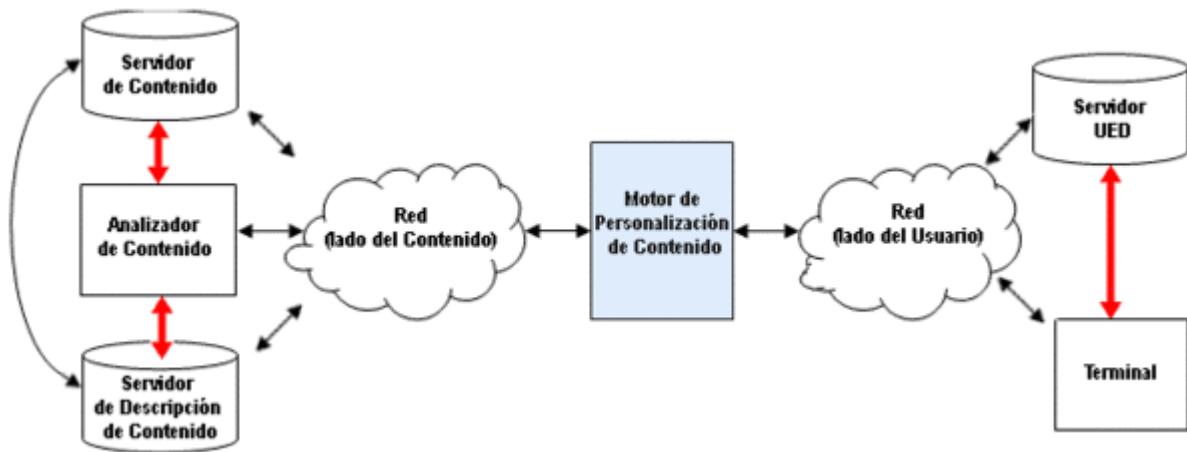


Ilustración 1 - Componentes que intervienen en acceso multimedia web

[http://es.wikipedia.org/wiki/Archivo:Sistema\\_UMA.gif](http://es.wikipedia.org/wiki/Archivo:Sistema_UMA.gif)

---

Profesor:

Dahianna Vega L.

Página 14 de 124

Alumnos:

Rogelio Elías, Rodrigo Riquelme, Manuel Canales

Tema:

Marco de Trabajo para un Acceso Multimedia Universal Mediante Patrones Modelo-Vista-Controlador

## 1.1. Formulación General del Proyecto

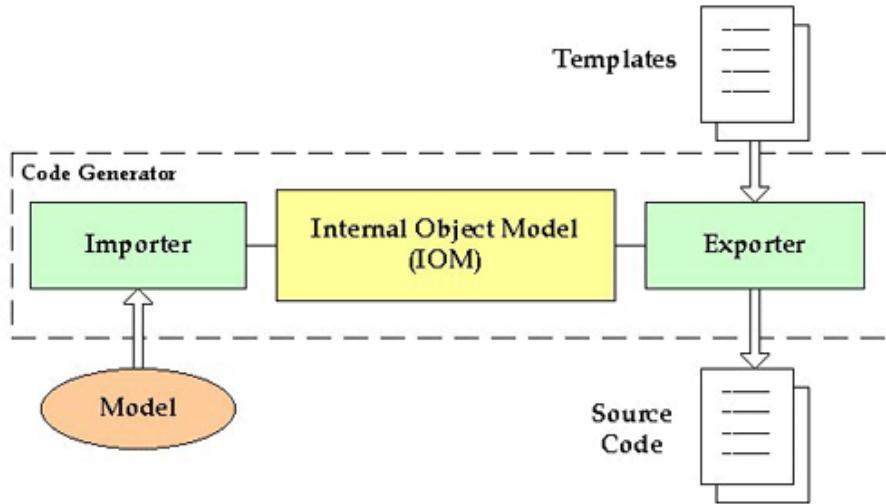
Para dar una solución a la problemática se plantea implementar un **marco de trabajo** que sirva de plataforma de desarrollo de sistemas con capacidades **UMA** basado en patrones Modelo-Vista-Controlador.

Como marco de trabajo nos referimos a un enfoque conceptual y técnico con implementación de software que sirva de guía al desarrollo de aplicaciones compatibles con UMA.

Un marco de trabajo para un Acceso Multimedia Universal debiera lograr la independencia entre contenido y dispositivo cliente en el cual se despliega tal contenido.

Por ello este Marco de Trabajo usará los patrones Modelo-Vista-Controlador para armar componentes MVC.

En la siguiente ilustración se muestra un diagrama MVC con uso de plantillas.



Esquema de MVC con uso de templates  
<http://onjava.com/onjava/2004/06/02/cg-vel-2.html>

El siguiente es un esquema propuesto de un componente XML con elementos modelo y vista que reflejaría en un código universal una situación similar al diagrama anterior.

```
<? xml version="1.0"?>
<component name="Model">
  <model name="InternalObjectModel">
    <id>%v</id>
  </model>
  <view name="Exporter">
    <template>cajita.html</template>
    <public_attribute>true</public_attribute>
  </view>
</component>
```

Como se puede apreciar en este ejemplo existe un objeto modelo y un objeto vista, dichos objetos modelados en XML no dependen de ningún lenguaje de programación en particular, por lo tanto se puede mantener el **diseño** de los componentes XML sobre diferentes plataformas logrando una buena portabilidad. Ocupando estos mismos componentes es posible comunicarse con diferentes plataformas a la vez, las cuales no necesariamente deben ser compatibles entre sí de forma nativa, de esta forma se maximiza la interoperabilidad.

Otro factor importante es que se separa el trabajo de programación en duro del diseño lógico de los componentes.

## 1.2. Objetivos

### 1.2.1. Objetivo General

Desarrollar un Marco de Trabajo para el desarrollo de aplicaciones con características de Acceso Multimedia Universal.

### 1.2.1. Objetivos Específicos

- Generar un Framework MVC sobre una plataforma específica como Java o PHP que interactúe con componentes XML que a su vez puedan ser interpretados en cualquier plataforma genérica con la programación necesaria.
- Generar un CMS con este framework tomando en cuenta los principios UMA, esto quiere decir que el contenido audiovisual subido una sola vez debiera ser compatible con distintas plataformas (PCs, móviles, blackberrys, iphones, ipads, etc).
- Resolver la codificación (FLV, MP4, MPG, etc.) del material audiovisual para que este pueda ser reproducido en las principales plataformas.
- Mostrar el despliegue de la información adecuado para cada dispositivo para que los contenidos sean accesibles en forma transparente a través de ellos, mediante los componentes XML generados.

- Usar los reproductores adecuados para la reproducción en cada plataforma (Flash, Quicktime, HTML 5, etc.).
- El CMS debiera ser escalable para aceptar nuevos dispositivos aunque aún no existan.

### **1.3.Metodología a Emplear para Desarrollar el Proyecto**

Se utilizará la metodología **XP** para el desarrollo del proyecto para mantener una retroalimentación fluida entre los desarrolladores, lo cual implica lanzamientos cortos y frecuentes.

Se utilizará un sistema de control de versiones para el software y documentación del desarrollo basado en Subversion para ello se usara el servicio de repositorio Subversión de Google y liberara el software bajo la licencia GNU GPL.

Se fijarán reuniones semanales para controlar y coordinar el avance del proyecto.

Se utilizará un IDE basado en Eclipse para desarrollar en un lenguaje con orientación a objetos y sincronizar el proyecto con el cliente SVN de éste IDE.

Se usarán las convenciones Java para generación de código y se documentará en el mismo proceso de codificación mediante un sistema compatible con Java docs.

Se sincronizará el proyecto en 2 ambientes:

**Desarrollo:** instalado en cada PC local con un IDE basado en Eclipse y sincronización mediante Subversion.

**Producción:** instalado en un servidor web accesible públicamente montado en una máquina Linux.

## 1.4. Planificación Inicial

Según lo planteado el proyecto se dividirá en 7 etapas, las cuales se presentan planificadas a continuación:

Etapas	Tareas posibles	Tiempo Tentativo
Recopilación de antecedentes (Creación del marco conceptual y estado del arte)	Investigación de sistemas con capacidades UMA (Universal Media Access). Investigación de frameworks MVC multiplataforma.	2 Semanas
Toma de requerimientos	Se definirán requerimientos específicos del sistema mediante la investigación previa y recopilación de antecedentes.	2 Semanas
Modelado del Sistema.	Modelamiento del framework Lanzamiento de pequeños prototipos para	2 Semanas

	hacer pruebas del framework MVC.	
Maqueta del sistema	Creación de maqueta funcional  Lanzamientos de pequeños prototipos para verificar comportamiento del framework.	1 Semana
Desarrollo e Implementación	Desarrollo de la aplicación  Implementación de los ambientes de desarrollo y producción en Beta.	3 semanas
Pruebas	- Se generará un set de pruebas unitarias y de integración. Dejando documentación y observaciones pertinentes muy breves.	2 semanas
Liberación	- Al término de las pruebas se entregará un primer prototipo de producción junto con la documentación generada en la etapa de desarrollo.  - Despues de un período de uso se definirán nuevos requerimientos como parte de un programa de mejoras continuas.	1 semana

## Capítulo 2. Marco Teórico

### 2.1. Acceso Multimedia Universal

El concepto de **Acceso Multimedia Universal** hace referencia a una línea de investigación en el sector de las comunicaciones multimedia, que busca una solución universal a la problemática surgida por el vertiginoso aumento del contenido audiovisual disponible para la gran mayoría de la población y la disparidad de redes de acceso y terminales del mercado.

La motivación inicial de **UMA** era posibilitar el acceso a contenido multimedia enriquecido a dispositivos con limitaciones de comunicación por disponer de capacidades "pobres" de procesamiento, almacenamiento y visualización.

Los esfuerzos se han centrado en dos líneas de trabajo:

- **Técnicas de transcodificación del contenido:** un sistema UMA requiere incorporar métodos de adaptación del contenido original a los recursos de la sesión y preferencias del usuario, tales como cambios de formato, reducción de tasa de bits, velocidad de reproducción o cambio de modalidad (*transmoding*), como por ejemplo pasar de texto a voz, o viceversa, etc.

- **Herramientas de acceso al contenido:** tareas como la descripción, indexado, análisis, búsqueda y filtrado. Son necesarias para conseguir entregar el contenido deseado y se basan en el uso de meta datos y aplicaciones dirigidas a realizar estas funciones. La mayoría se centra en los estándares MPEG-7 y MPEG-21 <sup>1</sup>Un conjunto de circunstancias y factores plantean la necesidad de tecnología UMA:
  - **Gran cantidad de contenido audiovisual:** el aumento de los aparatos capaces de generar contenido multimedia ha disparado la cantidad de material disponible.
  - **Difícil acceso a la información:** la búsqueda de un determinado contenido es una tarea ardua, ya que la mayoría de la información no es ordenada ni catalogada, y por lo tanto no se tiene ningún control sobre el contenido ni sobre su localización.
  - **Condiciones de acceso a la red diferentes y variables.** no todos los terminales manejan los mismos protocolos de comunicación ni tienen el mismo ancho de banda.

---

<sup>1</sup>UMA, Wikipedia [http://es.wikipedia.org/wiki/Acceso\\_Multimedia\\_Universal](http://es.wikipedia.org/wiki/Acceso_Multimedia_Universal)

- **Heterogeneidad de dispositivos cliente**, con capacidades y necesidades diferentes (cada fabricante intenta diferenciarse de sus competidores y no facilitan la compatibilidad entre aparatos). El auge de las comunicaciones móviles ha fomentado la diversidad de terminales.
- **Exigencias del usuario:** No siempre la información o contenido deseado consigue llegar al usuario final con la mejor calidad posible. Cuando un dispositivo intenta acceder a contenido para el cuál no ha sido diseñado, el resultado es decepcionante.<sup>2</sup>
- **Altos costos de mantenimiento y almacenamiento** de información. Es imposible tener una versión original del contenido para cada tipo de terminal.

---

<sup>2</sup>UMA, Wikipedia [http://es.wikipedia.org/wiki/Acceso\\_Multimedia\\_Universal](http://es.wikipedia.org/wiki/Acceso_Multimedia_Universal)

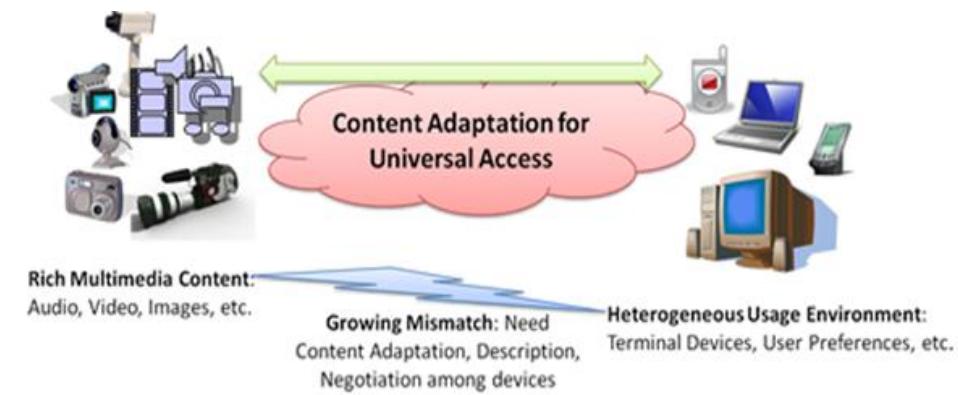


Ilustración 2 - Adaptación de contenidos para un acceso universal

<http://multimediacommunication.blogspot.com/2007/02/multimedia-communication-for-universal.html>

Los sistemas UMA están orientados a establecer una especie de "puente" entre el contenido original y el terminal de uso final, siendo un nexo de unión entre los diferentes componentes de la cadena de comunicación para optimizar su gestión.

Se trabaja en la adaptación de contenido creado una sola vez y almacenado en un solo formato, implementando sistemas capaces de transformar la información original en el formato que aproveche al máximo las características y condiciones<sup>3</sup>

<sup>3</sup>UMA, Wikipedia [http://es.wikipedia.org/wiki/Acceso\\_Multimedia\\_Universal](http://es.wikipedia.org/wiki/Acceso_Multimedia_Universal)

## 2.2. Protocolo XML orientado a objetos

Se pretende generar un marco de trabajo basado en objetos XML el cual funcionara como una capa de objetos a ser interpretados por un motor de scripting principalmente en el servidor (JSP, PHP) y eventualmente en el cliente (javascript).

Por esta razón se hizo para este proyecto un resumen de la investigación en implementaciones lo más parecidas posible en este paradigma de desarrollo en cuanto a construcción de objetos con XML.

### 2.2.1. SOAP

SOAP (Simple Object Access Protocol) es un protocolo de mensajes entre computadores basado en XML para invocar procedimientos en forma remota y utiliza protocolos para trasportar mensajes de texto. Siendo cualquier aplicación que cumpla las especificaciones puede invocar y proveer de servicios.

Soap contiene información adicional en el documento XML, como arreglos, el modelo descentralizado, es decir, puede ser procesado por varios intermediarios. Además posee características específicas para operaciones RPC con parámetros de entrada / salida.

La siguiente figura muestra un esquema de un objeto SOAP como envoltura para un mensaje de correo electrónico.

#### Attachment feature properties

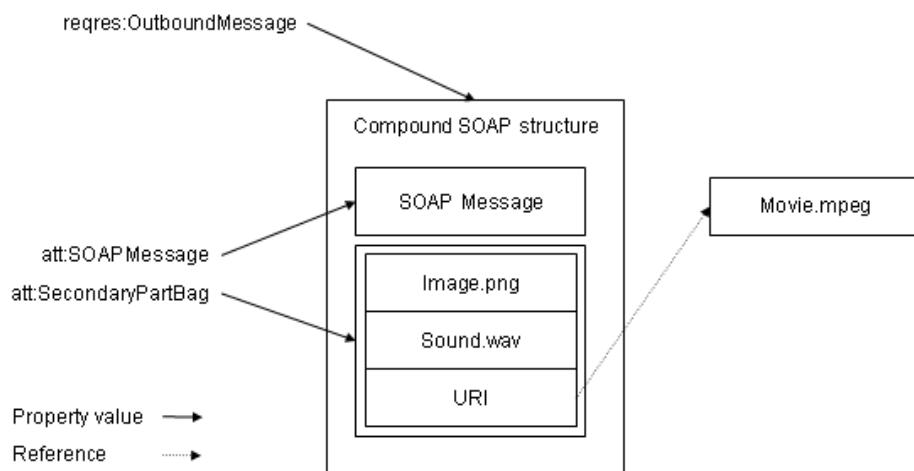


Ilustración 3 - Esquema SOAP según la W3C

<http://www.w3.org/TR/soap12-af/#W3C.WD-soap-part2>

SOAP fue desarrollado por Microsoft junto con IBM y actualmente es parte de los estándares de la W3C.

## 2.2.2. REST

REST (Representational State Transfer) es un conjunto de principios de una arquitectura de software para sistemas hipermedia

REST posee una serie de diseños fundamentales y que son claves:

- Un protocolo cliente/servidor sin estado.
- Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información: POST, GET, PUT y DELETE.
- Una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su url.
- El uso de hipermedios, HTML o XML.

## 2.2.3. RSS

RSS (SiteSummary o Rich Site Summary) es un formato XML para compartir contenido en la web. El cual es usado para difundir información a los usuarios que se han adherido a una fuente de contenidos que es actualizado de forma permanente. El formato RSS permite gestionar y distribuir contenido sin la necesidad de tener un navegador, ocupando un software desarrollado para leer el contenido RSS.

El RSS facilita la gestión y publicación de información y noticias de la web de forma estandarizada entre las páginas y los usuarios lectores de las páginas. Información que se distribuye a través de las fuentes RSS o Canales RSS conservando toda la información en un solo lugar que se actualiza de forma permanente y automática.

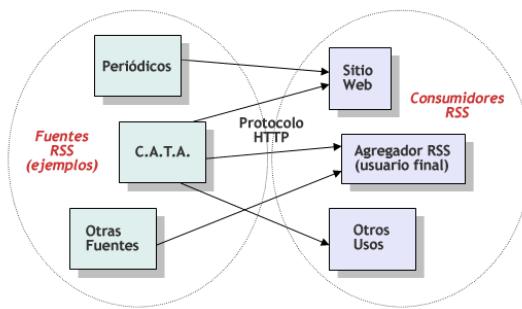


Ilustración 4 - Esquema del funcionamiento de RSS

<http://www.monografias.com/trabajos29/protocolo-acceso/protocolo-acceso.shtml>

## 2.2.4. XML Orientado a MVC

El protocolo XML orientado a MVC (Modelo Vista Controlador) o XML a 3 capas, es la integración de diferentes aplicaciones en tiempo real compartiendo bases de datos diferentes, siendo cada capa la que se preocupa de gestionar cada recurso en su nivel correspondiente.

La capa de presentación es usada para gestionar las interfaces graficas de las aplicaciones.

La capa de datos es la encargada de obtener los datos de la aplicación desde la capa anterior o capa de presentación.

La capa xml (controlador) nos permitirá obtener y manejar la distribución de los datos de las distintas aplicaciones desde una interfaz.

De esta forma se obtienen los siguientes beneficios:

- Acceso a la información en tiempo real.
- Indexación y organización de la información desde una misma interfaz
- Obtención y distribución de datos desde varias aplicaciones desde un mismo.
- Portabilidad de la interfaces de presentación a otras plataformas.
- Desarrollos abiertos y escalables, que permitirán ampliar funcionalidades e integrarlas.

En la siguiente figura se puede resumir el esquema de una solución XML bajo un modelo de 3 capas.



Ilustración 5 - Esquema de XML Orientado a MVC

<http://www.titansol.com/?sec=bloque4&lang=es>

### 2.3.1. Servidor Web

Dado que el ámbito del proyecto es la web y los contenidos que transporten los objetos están alojados en servidores web se hará una pequeña definición de los que es un servidor web

Un servidor web es una máquina, real o virtual, en la cual se encuentra instalado un software especial que implementa el protocolo http, con él, la máquina servidor atiende las peticiones de conexión de las máquinas clientes. Las máquinas clientes se comunican con el servidor entregándole la URL a un navegador Web u otro programa de conexión, la primera parte de la URL indica el nombre de dominio del servidor y el resto indica el archivo que se desea obtener dentro de la máquina servidor.

### 2.3.2. Stream

Para que los contenidos almacenados en un servidor web sean visibles por usuarios del servicio debe existir una descarga total o parcial de contenido.

A este flujo de descarga de contenidos audiovisuales en tiempo real se le llama **stream**.

Stream es una tecnología que permite la visualización de archivos multimedia mientras éstos se encuentran en plena transferencia.

Existen dos modos de realizar Streaming de video: HTTP Delivery y Streaming.

#### 2.3.2.1. HTTP Delivery

Es cuando se descarga un archivo multimedia almacenándolo en el dispositivo, generalmente de manera temporal, para luego ser abierto y leído. Este método posee la ventaja de poder acceder a las diferentes partes del archivo en forma más expedita pero a su vez posee la gran desventaja de que es necesario descargar todo el archivo para poder acceder a cualquier parte de este. La forma más sencilla de difundir videos de esta manera es realizando un vínculo al archivo.

Esta forma de entrega de archivos también es conocida como HTTP Streaming porque utiliza el protocolo HTTP para su difusión. Dicho protocolo es el encargado de enviar la información desde sitios Web, es por esta razón que es tan simple de implementar ya que prácticamente cualquier navegador será capaz de descargar el archivo.

Técnicamente este método no es Streaming de video pero es considerado como uno de los métodos.

### 2.3.2.2. Streaming

Permite la descarga de un archivo el cual no posee ni principio ni final, donde el caso más claro es la transmisión en directo. El tema es que un Stream debe ser transmitido de modo que cualquiera pueda conectar con él en cualquier momento, y no solo al principio de la transmisión. El Streaming de video funciona de manera diferente al HTTP Delivery, el usuario final es capaz de ver el archivo desde que éste se comienza a descargar. El archivo es enviado en forma de tramas relativamente constantes para que el usuario final pueda verlas mientras descarga el archivo.

Este método posee la desventaja de una velocidad de acceso a distintas partes del archivo de forma más lenta, la ventaja de este método es que es posible ver el

archivo en forma inmediata. A su vez existe la posibilidad de transmitir eventos en vivo gracias a que las tramas son enviadas mientras se generan.

Para realizar Streaming de video es necesario un servidor especializado en Streaming.

### 2.3.2.3. Media Streaming

Bajo este término 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.<sup>4</sup>

---

<sup>4</sup>[web15] What is Streaming?, Matt Voss. Texas A&M University.<http://helpdesk.doit.wisc.edu/helpdesk/page.php?id=5325>

Con la recepción de una pequeña parte el cliente es capaz de entregar su contenido al usuario, mientras continua recibiendo la corriente de datos (Streaming) que irá mostrando posteriormente. La parte almacenada actúa como “colchón” entre el ancho de banda irregular, que caracteriza a las redes TCP/IP y la continuidad que requieren las transmisiones de audio y video.

Los productos de media Streaming contemplan la distribución de contenidos tanto en la Internet. Los contenidos pueden estar almacenados previamente en un servidor (video ondemand, media Streaming), o crearse en el mismo momento de su difusión (live media Streaming). En ambos casos el audio y video se distribuyen con un formato de codificación.<sup>5</sup>

---

<sup>5</sup> What is Streaming?, Matt Voss. Texas A&M University.<http://helpdesk.doit.wisc.edu/helpdesk/page.php?id=5325>

### 2.3.2.4. Modelo de un servicio de streaming

El esquema convencional para la instalación de un servicio de video streaming tiene dos actividades fundamentales y bien diferenciadas, la elaboración de contenidos en un formato digital, utilizando procedimientos de compresión y la distribución de los contenidos por la red de los clientes o usuarios finales.

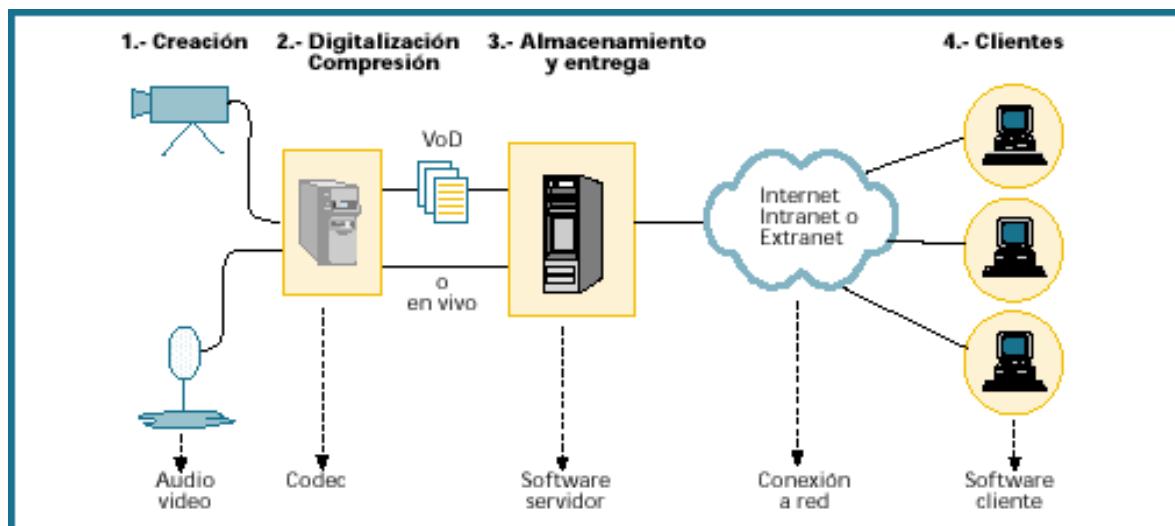


Ilustración 6 - Modelo típico de un servicio streaming

<http://www.rediris.es/difusion/publicaciones/boletin/58-59/ponencia10.html>

## 2.4. Codecs de Video

Los contenidos multimedia son interpretados por codecs los cuales permiten capturar y reproducir el contenido con distintas tasas de compresión según el algoritmo que posean. La eficiencia de estos algoritmos ha ido mejorando progresivamente lo que nos permite una mejor calidad con menos peso, esto también implica la conveniencia de reconvertir contenidos con codecs antiguos a otros más avanzados en la medida de lo posible.

Un códec de video es un programa computacional que permite comprimir y descomprimir un archivo de video digital. La utilización de un códec de video genera una pérdida de información en el archivo por la utilización de algoritmos de compresión que permite archivo de video de tamaño más reducido. En la compresión de datos de video, se ahorra espacio analizando cada cuadro (“frame”) y almacenando o muestreando solo la diferencia con el cuadro precedente. Este tipo de comprensión es conocido como “comprensión temporal”. El otro método de compresión de video elimina los datos de los pixeles que no cambian y es conocido como “comprensión espacial”. Regularmente un códec es asociado a un formato de archivo en particular, pero un formato de archivo puede trabajar con más de un tipo de códec. Al momento de la visualización del archivo de video son descomprimidas parte de éste.

## 2.4.1. H263 Sorenson

También llamado Sorenson Video Códec, Sorenson video Quantizer(SVQ), es un códec de video digital desarrollado por la empresa Sorenson media. Este códec es utilizado en formatos de video como Apple's QuickTime y Macromedia flash.

La primera vez que se utilizó este códec fue con el lanzamiento de QuickTime 3 en 1998, este método de comprensión se hizo popular luego que la incorporación en QuickTime 4 y la codificación del tráiler de la película “las guerras de las galaxias episodio 1:amenaza fantasma” en 1999. La versión de QuickTime 5.0.2, que incorporaba la 3 versión del códec de sorenson, fue la única en incluir exclusivamente este códec ya que Apple decidió migrar a sistemas de codificación propietarios de la empresa. La nueva versión de Sorenson códec llamada SorensonSpark fue incluida en Macromedia flash MX convirtiéndose en el códec a utilizar por los productos Macromedia (Ahora Adobe) para la codificación de video en su formato FLV.

## 2.4.2. H264 Mpeg-4 Parte 10

Algoritmo de codificación de video también llamado MPG-4 parte 10 AVC. Creado específicamente para uso de video conferencias o Internet. Fue adaptado para ser usado con video de alta calidad manteniendo las propiedades de transportabilidad por internet.

### 2.4.3. TrueMotion

Códec de video desarrollado por la empresa On2 technologies principalmente para juego de consolas y PC pero esta tecnología fue migrada por la empresa para la aplicación de Streaming de video de otras empresas como Adobe, Apple, Skype y AoL.

### 2.4.4. OGG Theora

Es un formato de comprensión de archivos multimedia, desarrollado por la fundación xiph.org, este formato que es libre de patentes comerciales. Está diseñado principalmente para soporte a la transmisión de videos on line y la comprensión de archivos multimedia.

Ese formato encapsula el audio y el video comprimido, el mismo formato puede ser utilizado para contener tanto audio como video codificado.

### 2.4.5. MPEG-4

Formato realizado en la década de los 90 como estándar para la industria pero lamentablemente no obtuvo los resultados esperados y las empresas que lo apoyaban dejaron de utilizarlo aunque existen algunas que aun dan soporte a este formato. La principal ventaja es la alta calidad de video y el reducido tamaño de sus archivos.

## 2.4.6. WMV

Formato de video generado por Microsoft, por lo que unas de sus ventajas es que es soportado por todos los dispositivos que posean algunas de las versiones de sistemas operativos de la compañía. Este formato se caracteriza por tener buena calidad de imagen y tamaño de los archivos relativamente pequeños. Las extensiones de este formato son Windows media video (.WMV) y AdvancedStreamingFormat(.ASF).

## 2.4.7. VP8

VP8 es el último códec de video de On2 Technologies diseñado para reemplazar a su antecesor, VP7. Fue anunciado el 13 de septiembre de 2008.

El 19 de mayo de 2010, Google, la cual adquirió On2 Technologies en 2009, liberó el códec VP8 como código abierto (bajo una licencia permisiva similar a la licencia BSD), en el marco de la conferencia Google I/O en mayo del 2010. Esto causó que VP8 fuese el segundo producto de On2 cuyo código fuente esté disponible a la comunidad del software libre, siendo el primero el códec VP3, liberado bajo la licencia GNU Lesser General Public License en 2002 a la Fundación Xiph.org, del cual posteriormente derivó el códec de video Theora.

Junto con el lanzamiento del código fuente de VP8 también se presentó el denominado proyecto WebM, que incorpora contribuciones y apoyo oficial de empresas como «Mozilla, Opera, Google, además de otros 40 editores y fabricantes de software y hardware» en un esfuerzo combinado para utilizar VP8 como el formato multimedia estándar en el lenguaje web HTML5. Microsoft también anunció que su más reciente versión de navegador Internet Explorer 9 también será capaz de soportar el códec VP8 si está instalado en el equipo.

## 2.5. Tecnologías Clientes

Existe una gran variedad de formatos de video con los cuales es posible realizar streaming, a continuación serán nombrados los más usados como Quicktime, Real Player, Windows Media y el más difundido a la fecha: Flash Player, sin embargo Flash está fuertemente amenazado con el objeto player de HTML 5 el cual ya viene integrado en los navegadores más avanzados y además de su no inclusión en gadgets Apple a partir de la creación del iPod.



Ilustración 7 - Logotipos de reproductores comerciales

## 2.5.1. Real Media Player

Real Media Player es uno de los primeros formatos en generar una solución de Streaming de video real, por esta razón es que su uso fue bastante difundido a mediados de la década de los 90, pero a pesar de su gran popularidad ha sido relevado por otros formatos de Streaming por el hecho de que su reproductor no estaba disponible en forma gratuita. La extensión de este formato es “.RM” (de real media). La última versión de su reproductor es Real Player version11.

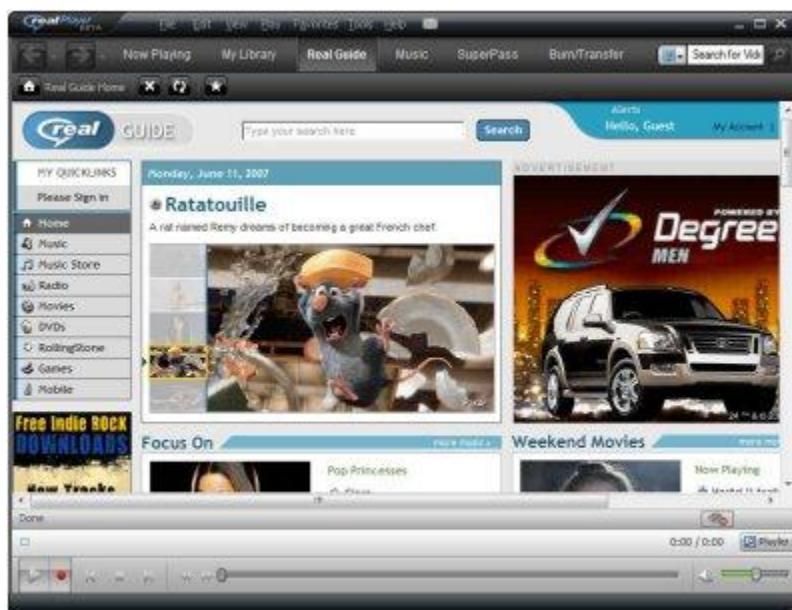


Ilustración 8 - Real Player 11

<http://www.real.com/>

## 2.5.2. Windows Media Player

Microsoft se encuentra en un proyecto bastante ambicioso el cual consiste en agrupar y estabilizar todos los tipos de contenidos multimedia dentro de un único concepto de estación de entretenimiento. aunque no se han podido ver avances reales dentro de esta tecnología, solo versiones de la aplicación que reúnen los contenidos dentro de un manejador multimedia.



Ilustración 9 - Presentación de Windows Media Center en Windows 7

### 2.5.3.Quicktime Player

Es uno de los formatos más utilizados para la transmisión de Streaming de video, una de sus ventajas es la posibilidad de realizar videos interactivos, panoramas, realidad virtual, entre otras opciones multimedia. La gran desventaja es lo complejo de utilizar para la realización de videos por parte de usuarios con poca experiencia. La extensión de este formato es .mov, las versiones más recientes soportan el códec H.264. La última versión de su reproductor es quicktime 7 es una alternativa propietaria al HTML 5 en iPods y iPhones.



Ilustración 10 - Reproductor Quicktime 7

## 2.5.4. Adobe Flash

Es uno de los players más usados para el uso de Streaming de video. La gran ventaja de este formato es el número de opciones de presentación que entrega Adobe Flash entre las cuales está la generación de video interactivo. Usa el formato FLV para la decodificación de los archivos, a partir de la versión 9 soporta el códec H.264.

Al ser Flash una plataforma de desarrollo permite ofrecer una amplia gama de posibilidades para personalizar el reproductor. Dado esto, existen players genéricos comerciales y no comerciales basados en esta plataforma, entre los cuales podemos mencionar:

### Flowplayer

Es un reproductor de vídeo para la Red en el ámbito mundial. Permite incrustar streams de videos. Es Software Libre bajo la licencia GPL. También tiene una versión comercial.

Algunas de las características incluyen alto nivel de posibilidades de personalización, una API javascript, arquitectura de plugins y el apoyo a diversos servidores de streaming.

## JWPlayer

De Long Tail Video, es uno de los reproductores más populares de Internet, es flexible y de código abierto. Admite la reproducción de cualquier formato que Adobe Flash Player pueda manejar (FLV, MP4, MP3, AAC, JPG, PNG y GIF).

Además, JW Player admite RTMP, HTTP, transmisión en vivo, formatos de listas diferentes, una amplia gama de ajustes y una extensa API de JavaScript.

Tiene una arquitectura de plugins que lo hace bastante versátil.



Ilustración 11- JW Player

- <http://www.longtailvideo.com>

## Open Video Player

Open Video Player (OVP) es una iniciativa que abarca el uso de estándares abiertos, mejores prácticas y metodologías establecidas de desarrollo de aplicaciones para reproductores multimedia.

En la actualidad este proyecto ofrece una base de código que puede ser utilizado para resolver problemas comunes, como parte de la construcción de prácticas avanzadas, escalables y sólidas basadas en la Web aplicadas a reproductores multimedia basados en Flash y Silverlight.

No es un reproductor específico es un marco de trabajo creado por una comunidad de desarrolladores en tecnologías propietarias y apoyado por grandes empresas como Adobe y Microsoft.

## 2.5.5. Video HTML5

Algunas de las nuevas características de HTML 5 son funciones para incluir, audio, vídeo, gráficos, almacenamiento de datos del lado del cliente y documentos interactivos.

El grupo de trabajo HTML 5 incluye AOL, Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera, y muchos otros proveedores.<sup>6</sup>

Aún no es un estándar de la W3C pero ya existen distintas implementaciones en los navegadores más avanzados.

---

<sup>6</sup> HTML5, W3C <http://dev.w3.org/html5/spec/>

## 2.6. Conversión de Videos

La conversión de videos consiste en recodificarlo para hacerlo compatible con otras plataformas además de ajustar la relación de peso y calidad para un streaming fluido tomando en cuenta factores como tamaño de pantalla y ancho de banda.

Si bien existe una gran variedad de software de conversión de escritorio con interfaz gráfica dedicado a productores audiovisuales y aficionados, para hacer un conversión a través de un servidor web requiere algo de más bajo nivel y que pueda ser ejecutado y configurado a través un script o líneas de comandos que se ejecuten en una consola.

Una tecnología que cumple estas condiciones es FFmpeg.

### 2.6.1. FFmpeg

FFmpeg es una colección de software libre que sirve para grabar, convertir y realizar streaming de video y audio, la cual está desarrollada en lenguaje de programación C. Incluye libavcodec, la cual es una biblioteca que contiene la gran mayoría de codecs FFmpeg en Linux, también puede ser compilado en plataformas Windows.

Cabe mencionar que muchos de los codecs se realizaron de desde cero y la visión de que este código sea altamente reutilizable.

Libavcodec es la biblioteca principal del proyecto FFmpeg. La cual es capaz de codificar y decodificar en diferentes formatos de audio y video.

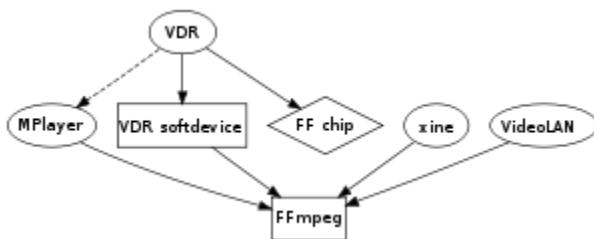


Ilustración 12 - Esquema de componentes de FFmpeg

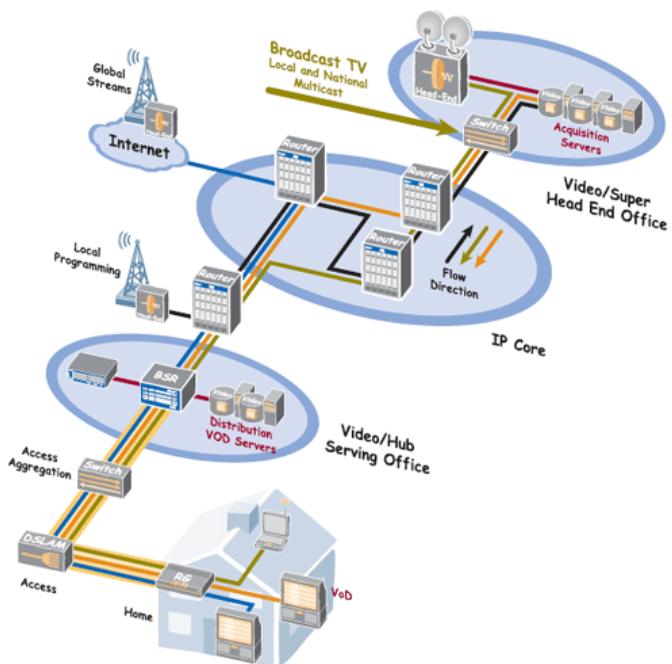
<http://es.wikipedia.org/wiki/Archivo:FFmpeg.svg>

## 2.7. IPTV

Internet Protocol Televisión (IPTV) de servicios es cada vez más y más popular entre las compañías de telecomunicaciones, ya que puede ofrecer programas de televisión en cualquier momento en cualquier lugar. Basado en el protocolo IP, IPTV características ventajas como la eficiencia de ancho de banda y la facilidad de gestión. IPTV es compatible con los servicios de radiodifusión como LiveTV y VideoOnDemand. Sistema de WiMAX móvil, capaz de garantizar altos anchos de banda y baja latencia, es adecuado para la prestación de servicios multimedia. Además, también proporciona una cobertura de área amplia, apoyo a la movilidad, y no la línea de operación de la vista. Por lo tanto, WiMAX es una solución prometedora para la entrega de servicios de IPTV en cualquier momento en cualquier lugar, especialmente a las zonas rurales o lugares remotos.

Formatos más ocupados son: H261, MPEG1, H263, MPEG2, MPEG3, MPEG4, WMV.

La siguiente ilustración muestra el funcionamiento e infraestructura de los servicios de IPTV



**Ilustración 13 - Infraestructura de redes IPTV**

<http://edna.dml.ce.sharif.edu/dmbsite/content/iptv>

## 2.8. Metodología de Desarrollo

Dentro del campo de acción de este proyecto hay un grado considerable de incertidumbre en cuanto a cuales tecnologías dominarán el negocio de la difusión de contenidos multimedia en la Internet. Faltan cosas por estandarizar y existen muchas tecnologías candidatas a ser un estándar dentro del mediano plazo que en este momento están en un estado incipiente.

Por es necesario un modelo de desarrollo altamente iterativo para que el software pueda adaptarse lo más rápidamente posible a los cambios del entorno.

Para cumplir con esto se trabajará en el campo de las metodologías ágiles Dentro del paradigma de métodos ágiles se propone la metodología Extreme Programming o XP bajo el modelo de Software Libre.

## 2.8.1. Extreme Programming

Extreme Programming (XP) es un enfoque de la ingeniería de software y el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.<sup>7</sup>

### Algunas Características de XP

- Enfatiza la satisfacción del cliente y el trabajo en equipo.
- Consiste en reglas y prácticas simples, ingenuas o extrañas al principio, fácilmente adoptadas luego, apoyadas unas en otras, con reducción de actividades improductivas.

---

<sup>7</sup> Extreme Programming, Dos Ideas [http://www.dosideas.com/wiki/Extreme\\_Programming](http://www.dosideas.com/wiki/Extreme_Programming)

- Acepta cambios de requerimientos aún tardíos en el ciclo de desarrollo.
- Integra gerentes, clientes y desarrolladores en la búsqueda de calidad en el software.
- Mejora el proyecto en comunicación, simplicidad, realimentación y emprendimiento.
- Mantiene el diseño simple y claro.
- Ensaya el software desde el primer día.
- Entrega temprano, implementa los cambios al ir siendo sugeridos.
- No ahorra en hardware, construye programas entendibles y extendibles.
- Automatiza las pruebas: escribe código de prueba antes, durante y después de la programación: antes, al definir la funcionalidad, al descubrir errores y después, en la integración.
- Actitud hacia el cambio: los clientes ven el software tempranamente.
- XP es una revalorización del software con mayor énfasis en la calidad.<sup>8</sup>

---

<sup>8</sup>Extreme Programming, Universidad La República,  
Uruguay <http://iie.fing.edu.uy/~nacho/blandos/seminario/XProg1.html>



## 2.8.2.Scrum

XP se complementa bien con Scrum ya que para el éxito de un proyecto basado en metodologías ágiles como XP, es necesario contar con una masa crítica de recursos humanos comprometidos con el proyecto.

Scrum se basa en la actitud y los principios de las personas para llevar adelante el proyecto, estos principios son esenciales para el desarrollo ágil .

**Compromiso:** Estar dispuesto para comprometerse a una meta. La metodología la da a las personas la autoridad que necesitan para cumplir con sus compromisos.

**Enfoque:** Haz tu trabajo. Enfoca todos tus esfuerzos y habilidades para trabajar en lo que te comprometiste a hacer. No te preocupes por nada más. Alguien lo hará por ti.

**Transparencia / Honestidad:** Scrum mantiene todo acerca del proyecto visible a todos.

**Respeto:** Los individuos estamos formados por nuestros orígenes y nuestras experiencias. Es importante respetar las diferentes a las personas del equipo y sus formas de pensar.

**Coraje:** Tener el coraje para comprometerse, actuar, ser honesto y esperar respeto.<sup>9</sup>

## Roles en Scrum

### Roles "Cerdo"

Los Cerdos son los que están comprometidos con el proyecto y el proceso Scrum; ellos son los que "ponen el jamón en el plato".

ProductOwner

El ProductOwner representa la voz del cliente. Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio. El ProductOwner escribe historias de usuario, las prioriza, y las coloca en el ProductBacklog.

ScrumMaster (o Facilitador)

El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga.

---

<sup>9</sup>Scrum, Dos Ideas <http://www.dosideas.com/wiki/Scrum>

El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.

### Equipo

El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 5 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc).

### Roles "Gallina"

Los roles gallina en realidad no son parte del proceso Scrum, pero deben tenerse en cuenta.

La gallina alimenta al proyecto "poniendo huevos", no se ve comprometida como el cerdo que va al matadero.

### 2.8.3. Software Libre

Es importante mencionar que en el presente proyecto se utilizará el modelo de software libre.

Exitosos proyectos como Linux, Apache, Open Office y Firefox (cuyos orígenes se remontan al moribundo proyecto propietario Netscape) entre otros, han demostrado la eficacia de liberar el Software para mantener los proyectos de software vivos y en desarrollo.

Al respecto, se citarán algunos puntos del libro “The Cathedral & The Bazaar” escrito por Eric S. Raymond en 1997 como principios del modelo de desarrollo de software libre los cuales encajan perfectamente en el campo de métodos ágiles propuesto<sup>10</sup>.

1. Todo buen trabajo de software comienza a partir de las necesidades personales del programador (todo buen trabajo empieza cuando uno tiene que rascarse su propia comezón).
2. Los buenos programadores saben qué escribir. Los mejores, qué reescribir (y reutilizar).
3. Considere desecharlo; de todos modos tendrá que hacerlo
4. Si tienes la actitud adecuada, encontrarás problemas interesantes.
5. Cuando se pierde el interés en un programa, el último deber es darlo en herencia a un sucesor competente.
6. Tratar a los usuarios como colaboradores es la forma más apropiada de mejorar el código, y la más efectiva de depurarlo.
7. Libere rápido y a menudo, y escuche a sus clientes.

---

<sup>10</sup>The Cathedral & the Bazaar - Eric S. Raymond - O'Reilly Media 2001

8. Dada una base suficiente de desarrolladores asistentes y beta-testers, casi cualquier problema puede ser caracterizado rápidamente, y su solución ser obvia al menos para alguien. O, dicho de manera menos formal, "con muchas miradas, todos los errores saltarán a la vista".
9. Las estructuras de datos inteligentes y el código burdo funcionan mucho mejor que en el caso inverso.
10. Si usted trata a sus analistas (beta-testers) como si fueran su recurso más valioso, ellos le responderán convirtiéndose en su recurso más valioso.
11. Lo mejor después de tener buenas ideas es reconocer las buenas ideas de sus usuarios. Esto último es a veces lo mejor.
12. Frecuentemente, las soluciones más innovadoras y espectaculares provienen de comprender que la concepción del problema era errónea.
13. La perfección (en diseño) se alcanza no cuando ya no hay nada que agregar, sino cuando ya no hay nada que quitar.
14. Toda herramienta es útil empleándose de la forma prevista, pero una \*gran\* herramienta es la que se presta a ser utilizada de la manera menos esperada.
15. Cuándo se escribe software para una puerta de enlace de cualquier tipo, hay que tomar la precaución de alterar el flujo de datos lo menos posible, y nunca eliminar información a menos que los receptores obliguen a hacerlo.

16. Un sistema de seguridad es tan seguro como secreto. Cuídese de los secretos a medias.
17. Para resolver un problema interesante, comience por encontrar un problema que le resulte interesante.
18. Si el coordinador de desarrollo tiene un medio al menos tan bueno como lo es Internet, y sabe dirigir sin coerción, muchas cabezas serán, inevitablemente, mejor que una.

### 2.8.3.1. Licencia GNU GPL v2

El proyecto a desarrollar estará liberado bajo la licencia GNU GPL v2.

La Licencia Pública General de GNU pretende garantizar la libertad de compartir y modificar el software y asegurar que el software es libre para todos sus usuarios.

Cuando se habla de software libre, se refiere a libertad, no a precio. Es posible distribuir copias de software libre y cobrar por ese servicio si se quiere, de recibir el código fuente o que se pueda conseguir si se quiere, de que se pueda modificar el software o usar fragmentos de él en nuevos programas libres, y de que los involucrados sepan que pueden hacer todas estas cosas.

Para proteger esos derechos se necesitan algunas restricciones que prohíban a cualquiera negar estos derechos a terceros o pedir que se renuncie a ellos. Estas restricciones se traducen en ciertas obligaciones que afectan a la distribución de copias del software y a su modificación.<sup>11</sup>

---

<sup>11</sup> Licencia GPL GNU v2 <http://www.gnu.org/licenses/gpl-2.0.html>

## 2.9. Frameworks

Parte medular de este proyecto es la creación de un marco de trabajo o Framework, por esta razón es pertinente de hacer una definición técnica de lo que es un Framework una pequeña descripción de algunos de los más usados en la web.

Un **Framework** es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Típicamente, puede incluir soporte de Programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Los Framework hoy en día simplifican el desarrollo de aplicaciones web y muchos implementan el patrón arquitectónico MVC (Modelo, Vista, Controlador).

## 2.9.1. Zend Framework

Zend es la principal compañía que está detrás del desarrollo de PHP. Este framework se centra en la construcción de desarrollo más seguro, fiable y moderno en aplicaciones y servicios Web 2.0. Es de código abierto simple, cien por ciento dirigido a la orientación a objeto. Zend Framework ofrece un gran rendimiento y robusta implementación Modelo Vista Controlador (MVC).

En el siguiente diagrama podemos ver un resumen de sus componentes.

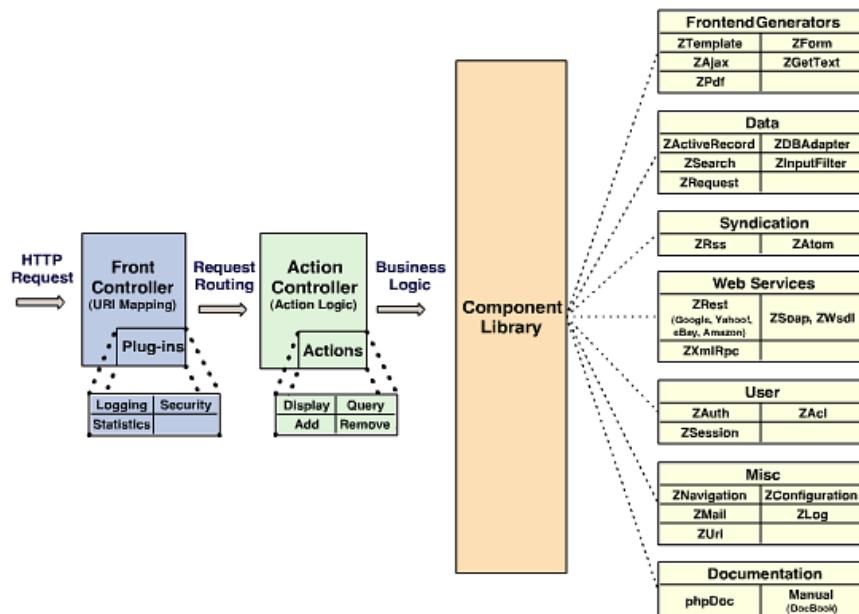


Ilustración 14 - Visión general Zend Framework

<http://www.programania.net/otros/zend-framework-una-vision-general/>

## 2.9.2. Google Web Toolkit

Google Web Toolkit es un conjunto de herramientas de desarrollo para crear y optimizar la complejidad basada en las aplicaciones del navegador que se encuentran bajo la plataforma Java. Su objetivo es permitir el desarrollo de aplicaciones Web de alto nivel sin que el desarrollador tenga que ser un experto en navegadores, XMLHttpRequest y JavaScript. GWT. Es utilizado por muchos productos de Google, incluyendo Google Wave y la nueva versión de AdWords. Finalmente es importante mencionar que Google Web Toolkit es de código abierto.

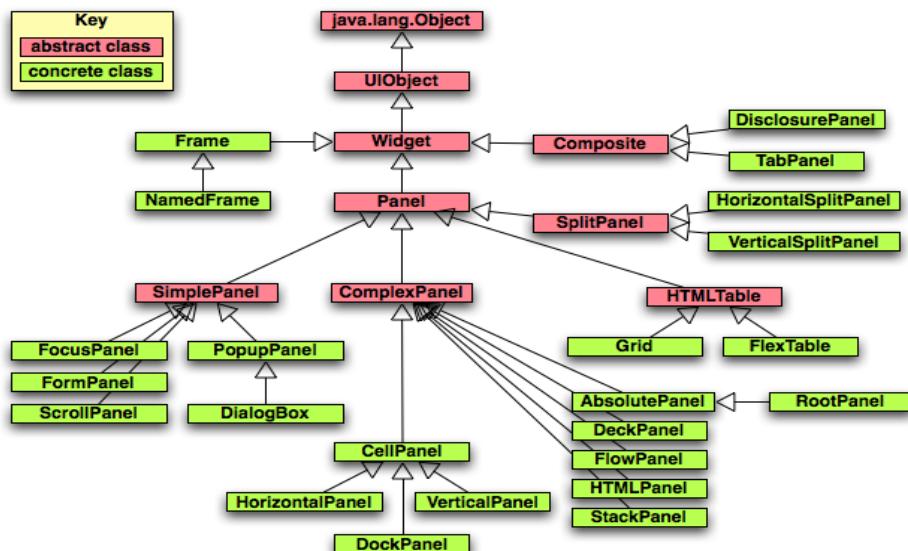


Ilustración 15 - Esquema de Widgets GWT

<http://java.ociweb.com/mark/programming/GWT.html>

## Capítulo 3: Estado del Arte

### 3.1. Gestores de Contenidos multimedia existentes

El marco de trabajo o Framework a desarrollar deberá servir en última instancia para un Gestor de Contenidos o CMS (Content Management System). A continuación se presentará un resumen de los CMS que se encuentran en el mercado que tengan características multimedia.

#### 3.1.1. PHPMotion

PHPMotiones un script creado en PHP que permite crear un sitio tipo Youtube. Entre sus principales características están:

- Facilidad para subir videos directamente desde el computador del usuario.
- Soporta varios formatos de vídeo (mpg, avi, divx y mas)
- Facilidad para editar los datos del vídeo (título, descripción y tags)
- Soporte para videos públicos y privados.
- Soporte para comentarios en los videos.
- Soporte para embeber videos en otros dominios.
- Soporte para grupos.

- Facilidad para crear favoritos
- Cuenta con un administrador de publicidad.

Usa PHP 4.3 MySQL, LAME MP3 Encoder, Libogg + Libvorbis, Mencoder y además Mplayer, FFmpeg-PHP, GD 2.

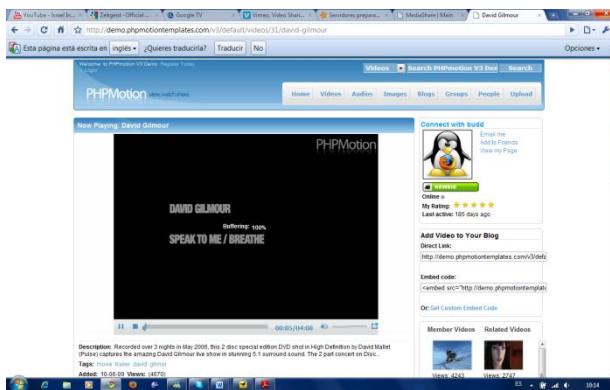


Ilustración 16 - Web PHPMotion

<http://www.phpmotion.com/>

### 3.1.2.OsTube

OSTube es un CMS de videos basado en PHP4 con MySQL, con soporte de línea de comandos.

Usa MPlayer para conversión de videos, flvtool2 para generar los metadatos de los archivos generados.

Está en Inglés y Alemán, tiene versiones Free, Profesional y empresarial.

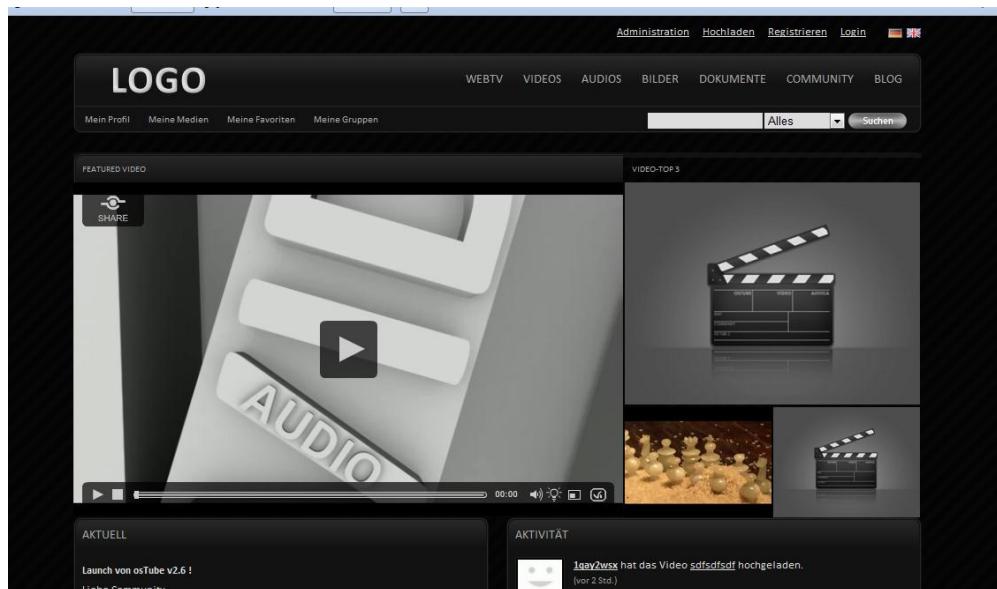


Ilustración 17 - OSTube

<http://www.ostube.de/en/ostube>

### **3.2. Sitios de contenidos multimedia de referencia**

Los gestores de contenidos multimedia como los que se presentaron tienen como finalidad ser el soporte de software para desarrollar y mantener un sitio web con cuyo fuerte sean los contenidos audiovisuales.

A continuación se mostrará un listado de los principales sitios web de esas características a nivel mundial y nacional.

#### **3.2.1.Youtube**

Fue creado por tres antiguos empleados de PayPal en febrero de 2005. En noviembre de 2006 lo adquirió Google y ahora opera como una de sus filiales. YouTube usa un reproductor en línea basado en Adobe Flash para servir su contenido. Es muy popular gracias a la posibilidad de alojar videos personales de manera sencilla. Aloja una variedad de clips de películas, programas de televisión, videos musicales, así como contenidos amateur como videoblogs. Los enlaces a videos de YouTube pueden ser también puestos en blogs y sitios electrónicos personales usando API o incrustando cierto código HTML.



Ilustración 18 - Youtube

<http://www.youtube.com>

### 3.2.2. Google Video

**Google Video** es un servicio de Google que hasta enero de 2009 permitía subir clips de vídeo a sus servidores para que cualquier persona los pudiera buscar y ver directamente desde su navegador. Inicialmente nació como competencia de YouTube, a la que terminó comprando el 2006. Finalmente, Google Vídeo pasó a funcionar como un mero buscador de videos en la red, pasando a ser YouTube el único servicio que permite la subida de vídeos.

En su sitio web describen así el servicio:

*Nuestra misión es organizar la información del mundo, y eso incluye los miles de programas de televisión de cada día. Google Video permite buscar en un creciente*

*archivo de contenido televisivo – cualquier cosa desde deportes a documentales de televisión o programas de noticias.<sup>12</sup>*

Su objetivo, además de indexar la mayor cantidad posible de video, es llegar a tener todos los programas televisivos. En la actualidad ofrece dos tipos de servicios, uno orientado a los usuarios finales, que permite a cualquiera subir archivos de video, y otro que permite a los creadores de contenido multimedia distribuir sus creaciones pagando una pequeña tarifa.

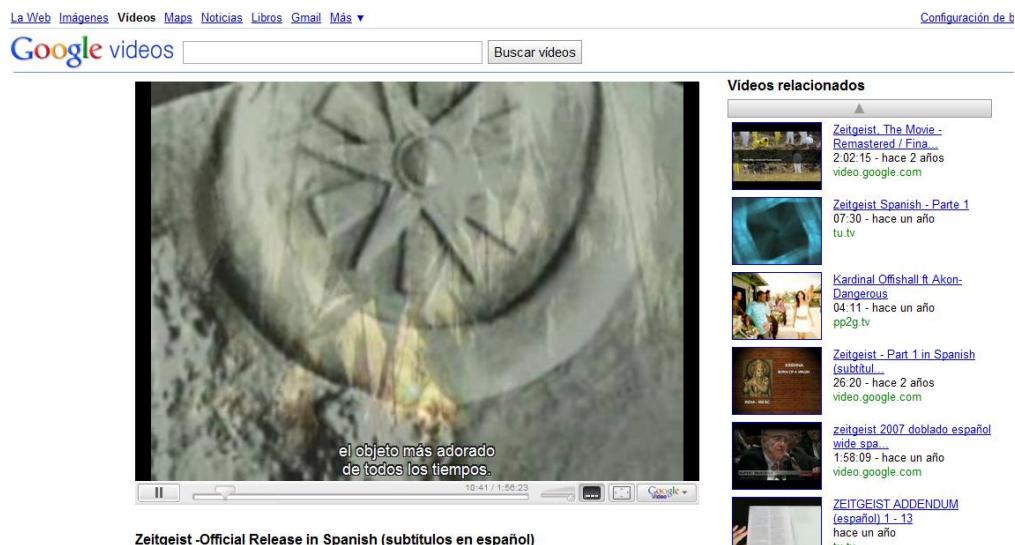


Ilustración 19 - Google Video

<http://video.google.es/>

<sup>12</sup>Google Video, Wikipedia [http://es.wikipedia.org/wiki/Google\\_Video](http://es.wikipedia.org/wiki/Google_Video)

El 9 de enero de 2006, Google puso a disposición de sus usuarios su propio reproductor de video y añadió la posibilidad de descargar los videos de su página en formato gvi, y en formato .mp4 preparado para iPod y PSP.

Se prevee que Google paulatinamente irá migrando este servicio hacia Google TV, un servicio que funciona por IPTV, tecnología que ya se encuentra presente en los televisores más avanzados.

### 3.2.3. Vimeo

**Vimeo** es una red social de Internet basada en videos, lanzada en noviembre de 2004 por la compañía InterActiveCorp (IAC). El sitio permite compartir y almacenar videos digitales para que los usuarios comenten en la página de cada uno de ellos. Los usuarios deben estar registrados para subir videos, crear su perfil, cargar avatares, comentar y armar listas de favoritos.

Vimeo no admite comerciales de televisión, demostraciones de videojuegos, pornografía o cualquier contenido que no haya sido creado por el usuario. El sitio, además, ha ganado reputación como «proveedor de imágenes» para diversos artistas, debido a la alta tasa de bits y resolución de sus videos. Desde octubre de 2007, Vimeo ofrece una opción para videos de alta definición



---

Ilustración 20 - Vimeo

<http://vimeo.com/>

### 3.2.4.TerraTV

Está basado en Flash Player tiene muy bien resuelto el tema de las cortinas publicitarias, logra emular el concepto de televisión a través de un flujo continuo logrado con el uso de listas de reproducción que hace que los videos corran uno detrás de otro sin necesidad de actividad de parte del usuario, al igual que en la TV el usuario no tiene control de las listas.

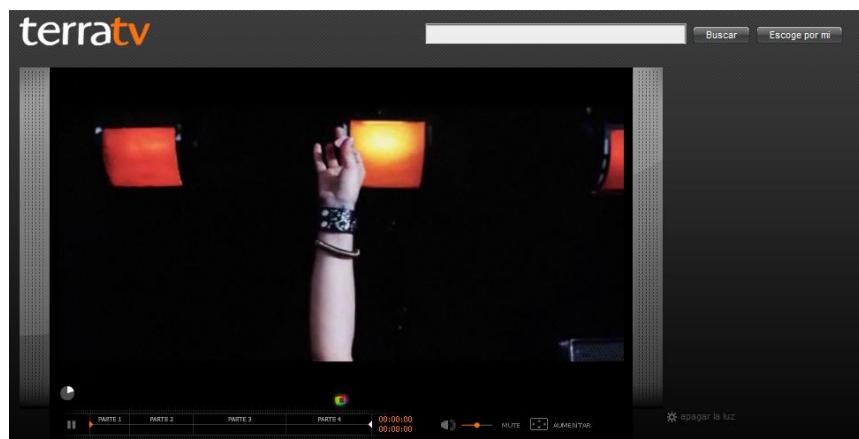


Ilustración 21 - Terra TV

<http://www.terratv.terra.cl>

### 3.2.5. EmolTV

Es el portal de videos de El Mercurio, tiene una interfaz bastante ágil en la navegación aprovechando las ventajas de Ajax para cargar los videos. Esto se ha logrado sacrificando un poco el feedback del usuario el cual no tiene la posibilidad de poner sus comentarios en el mismo sitio, lo cual permite que la interfaz sea más simple.

No maneja listas de reproducción lo que exige la interacción del usuario para mantenerse funcionando.

Aparentemente no tiene una solución inteligente para las cortinas publicitarias, éstas aparecen fusionadas en el proceso de edición manual con el video mismo.



Ilustración 22 - Emol TV

<http://www.emoltv.emol.com>

### 3.2.6. 3TV

Es el sitio de videos de La Tercera, tiene un despliegue bastante predecible, parece un clon de una versión antigua de youtube, el único elemento visual que sale de este esquema son los carruseles de imágenes flash en los cuales se destacan algunos videos por canal.

Tiene una implementación de publicidad que aprovecha las capacidades del reproductor JW Player los videos están en versión HD (H.264) y calidad normal (flv).

No hay listas de reproducción por lo que requiere una constante interacción con el usuario para mantenerse funcionando.



**Ilustración 23 - 3TV**

<http://www.3tv.cl>

### 3.3. Google TV

**Google TV** es una plataforma de hardware y software para televisores de alta definición basado en el sistema operativo Android, a través del uso de IPTV. En simples palabras este consiste en llevar todo el contenido de Internet más la TV digital a nuestras casas además de interactuar con estas.

Google TV funciona sobre Android 2.1 y puede ser actualizado de forma remota. El navegador es Google Chrome, corriendo Flash. El uso de Android permite que las aplicaciones de los móviles funcionen también en la TV.

Los desarrolladores ya pueden comenzar a crear aplicaciones para Google TV, y se espera que se lance un AndroidMarket para este sistema a principios de 2011. Para entonces también estarán disponibles APIs para Google TV.

Existe una alianza estratégica conformada con Google, Sony e Intel para el desarrollo de esta plataforma y otras similares basadas en IPTV.

Los protocolos de comunicación de estos dispositivos con servidores web están basados en XML también acepta formatos RSS ya existentes como playlists de iTunes.

Sony lanzó el primer modelo de televisores de alta definición fusionado con Google TV, a diferencia con otras TV con Internet, esta permite al usuario navegar a través de Google Chrome y ver televisión de manera simultánea utilizando la Opción Picture in Picture, estos TV están equipados con un hardware que contiene un procesador Intel Atom y una capacidad de 8GB de memoria, conectividad WiFi.



Ilustración 24 – Google TV en un televisor IPTV conectado a internet

<http://www.fayerwayer.com/2010/05/google-tv-ya-esta-al-aire/>

## 4. Desarrollo

---

### 4.1. Toma de requerimientos

Los requerimientos se definen de acuerdo a esta investigación, tomando en cuenta el estado del arte, en la primera iteración se tomaran requerimientos muy específicos y corresponden al core de la aplicación y del framework. A medida que se vayan alcanzando algunos objetivos se irán definiendo más objetivos que permitan perfeccionar el software.

En esta instancia definen los requerimientos de la primera etapa, los que están más sujetos a cambio son los requerimientos funcionales.

#### 4.1.1. Requerimientos Funcionales

- La plataforma debe ser web.
- El sistema debe administrar contenidos audiovisuales (videos) y etiquetarlos adecuadamente para la web (titulo, descripción, fecha de creación, tags).
- El sistema debe tener una interfaz de front-office y otra de back-office con por lo menos un perfil de administrador.
- Se deben poder convertir los videos en el back office para mostrar los formatos adecuados en diferentes dispositivos clientes.

#### 4.1.2. Requerimientos No Funcionales

- Debe estar basado en los patrones Modelo-Vista-Controlador, con componentes independientes y reutilizables
- Los componentes deben usar un lenguaje multiplataforma com XML o Json.
- Se deben tomar medidas para evitar inyecciones SQL malintencionadas.
- El front office debe funcionar con templates independientes de las vistas y modelos.
- Las vistas deben ser, en lo posible, independientes de los modelos y templates.

## 4.2. Tecnología a Utilizar

Ya que son bastantes las TI involucradas en el desarrollo de este proyecto, se dividirá este tema en 2 frentes: el lado servidor en el cual está el core de la aplicación sobre un servidor Linux, y el lado cliente donde se depende de las capacidades del agente de usuario y principalmente de el navegador web.

Para este lanzamiento específico se usaran PHP 5.3, MySQL 5, FFmpeg, JQuery, JW Player Flash.

### 4.2.1. Lado Servidor

#### 4.2.1.1. PHP 5.3

Para la elección de la tecnología es importante privilegiar las que nos ofrezcan la posibilidad de un desarrollo rápido y a la vez escalable.

PHP es uno de los lenguajes web más orientados al desarrollo rápido además permite usar orientación a objetos con las consiguientes ventajas en cuanto a escalabilidad y reusabilidad de componentes.

La última versión de PHP a la fecha es la 5.3, una de las principales ventajas de esta versión con respecto a las anteriores es la posibilidad de usar namespaces lo que nos permite encapsular clases, esto es equivalente al uso de packages en Java. Esta característica surgió como respuesta a algunos inconvenientes surgidos en proyectos relativamente grandes así como algunos frameworks en

que la falta de namespaces ha llevado a la creación de nombres de clases excesivamente largos.

Se correrá PHP con el servidor Apache 2.

Las clases principales estarán dentro de una carpeta de clases con namespaces y subcarpetas models, views, controllers, lib y admin.

La siguiente figura muestra la estructura de la carpeta classes (en la barra izquierda).

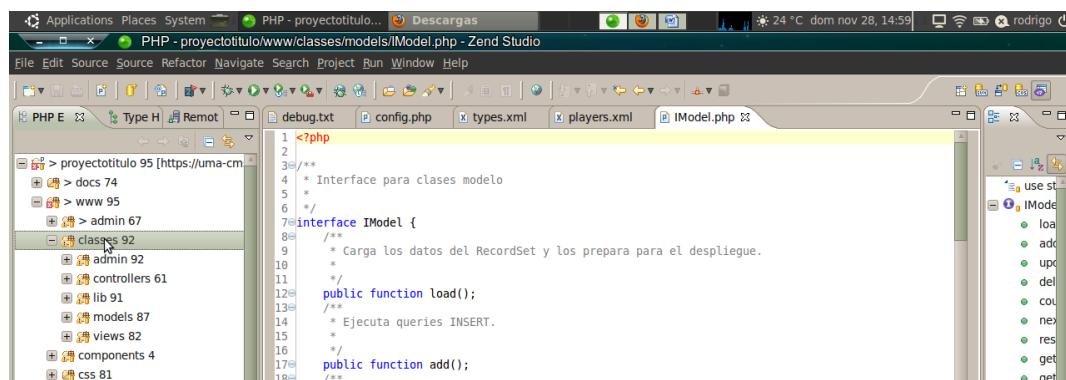


Ilustración 25 - Estructura Clases PHP del Core del CMS

#### 4.2.1.2. MySQL 5

MySQL es uno de los motores Open Source más usados a nivel mundial, el motor de MySQL MyISAM es muy rápido en consultas tipo SELECT, por otra parte el motor InnoDB nos permite usar características transaccionales si bien no es tan rápido en los SELECT como MyISAM como contraparte a esto tenemos la ventaja de que en las operaciones INSERT, UPDATE, DELETE los bloqueos de tablas son solo para el registro que se está tocando a diferencia de MyISAM en que se bloquea la tabla entera al hacer cualquier modificación.

Como resumen a lo expuesto anteriormente podemos decir que MyISAM es una buena elección para tablas que tienen muchas consultas y pocas modificaciones e InnoDB es la mejor elección para tablas que son modificadas recurrentemente y tengan consultas de listados.

Podemos usar estos dos engines en la misma base de datos de modo que las tablas de listado recurrente y actualizaciones menos recurrentes manejen el engine MyISAM y las tablas de actualizaciones más frecuentes tengan el engine InnoDB.

#### 4.2.1.3. FFmpeg

Se usará FFmpeg para realizar las conversiones de los videos, FFmpeg será invocados por scripts PHP mediante el comando exec(), el cual permite ejecutar instrucciones en la consola del sistema servidor.

El siguiente ejemplo muestra un esquema de la sintaxis de ffmpeg para realizar la conversión de un video donde \$infile es el archivo de entrada, \$outfile es el archivo de salida, \$acodec es el codec de audio, \$vcodec es el codec de video, el bitrate de audio es de 96 kb/s y 500 kb/s el de video.

```
ffmpeg -i {$infile} -acodec{$acodec} -ab 96k -vcodec{$vcodec} -b 500k {$outfile}
```

## 4.2.2. Lado Cliente

### 4.2.2.1 Javascript

Se estructuran las carpetas javascript dentro de una carpeta llamada js en la raíz del sitio con la siguiente estructura de carpetas relativas a la raíz:

**js/api:** funcionalidad desarrollada específicamente para este sistema particular aunque podría usarse para otros casos, la idea es generar acá un api propia del CMS.

**js/framework:** acá se servirán los frameworks javascript a utilizar, en la primera etapa se usará JQuery pero eventualmente podrían usarse otros frameworks.

**js/lib:** funcionalidades específicas las cuales dependen de los frameworks utilizados para trabajar, en esta etapa usaremos librerías basadas en JQuery.

Existe una técnica para reducir el peso de los archivos javascript llamada compresión, esta consiste en eliminar todos los espacios y saltos de líneas innecesarios lo que logra reducir el peso de los archivos entre un 30% y un 70%.

Sin embargo esta técnica también presenta dificultades ya que se pierde toda la indentación del código lo que va en desmedro de la legibilidad de este.

Como una forma de minimizar este inconveniente se guardarán todos los fuentes de los archivos javascript sin comprimir en carpetas llamadas source a lo largo de

todo el árbol de directorios javascript. Todas las modificaciones se harán sobre esos archivos para luego comprimir si se estima conveniente.

Siguiendo el principio de Extreme Programming “hágalo funcionar, después hágalo funcionar rápido” en esta etapa se trabajará sobre las carpetas source, es decir no se comprimirá ningún archivo que esté sujeto a posibles modificaciones aunque se deja la estructura de directorios para poder hacerlo más adelante.

La imagen a continuación muestra la estructura de carpetas javascript en el explorador del IDE PHP, Zend Studio.

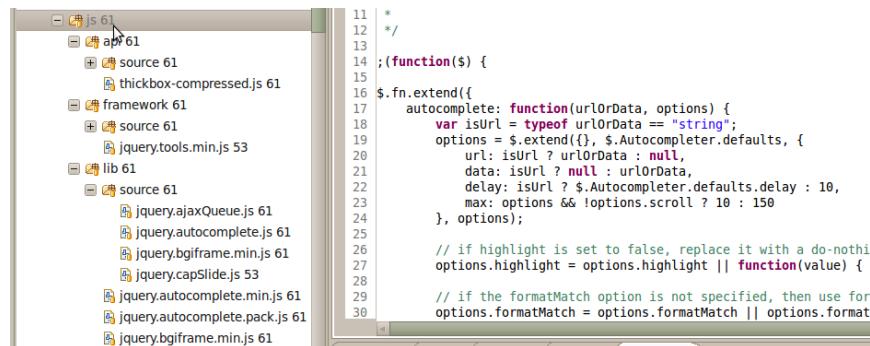


Ilustración 26 - Estructura de carpetas javascript

#### 4.2.2.2 JW Player

JW Player es reproductor Flash listo para usar, tiene una versión gratuita para uso no comercial, es totalmente personalizable mediante parámetros flashvars lo que nos permite extenderlo y hacer modificaciones en su look and feel sin modificar el código fuente del Flash.

También cuenta con una API javascript para trabajar con eventos del player sin necesidad de intervenirlo por dentro.

Es importante aclarar que el código embed HTML de este player y de cualquier otro que se pudiera usar es administrable en el CMS por lo que no es obligatorio usar este u otro player específico, sin embargo esto brinda una solución inmediata para la plataforma Flash la cual sigue siendo la dominante en el mercado.

## 4.3. Entorno de Desarrollo

### 4.3.1. Entorno Integrado de Desarrollo (IDE)

Se usará Zend Studio 7 que ofrece un entorno integrado para PHP basado en Eclipse, este entorno también incluye Zend Server y la posibilidad de integración nativa con Zend Framework, este se usara en ambientes Linux y Windows.

Zend Server se utilizará en los entornos de desarrollo por las facilidades que otorga para la configuración gráfica de Apache y PHP, no así en producción donde se utilizará el servidor Apache.

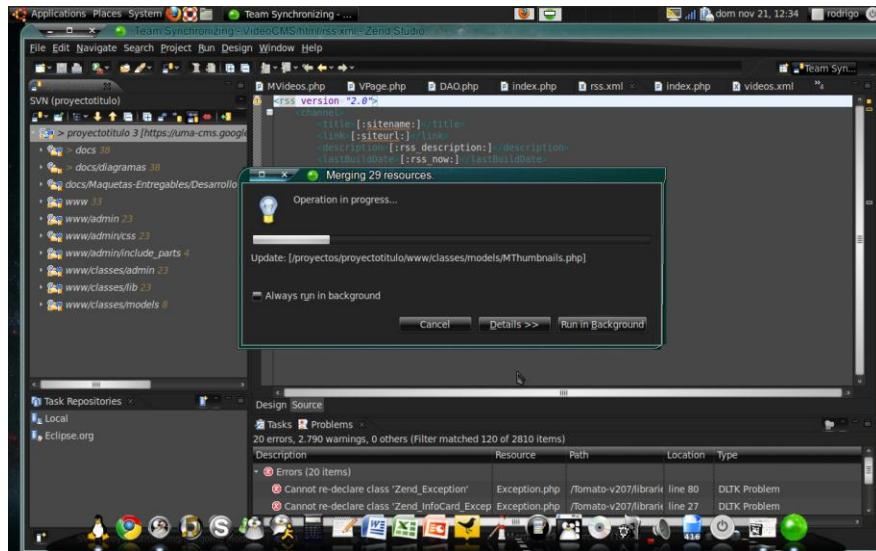


Ilustración 27 - Zend Studio en Ubuntu Linux

#### 4.3.2. Control de versiones

Se usará el repositorio SVN de Google Code para sincronizar con el cliente SVN de Zend Studio, se utilizará la rama trunk como el canal principal y se creará una rama de test para hacer pruebas que ayuden a los desarrolladores a comprender el sistema.

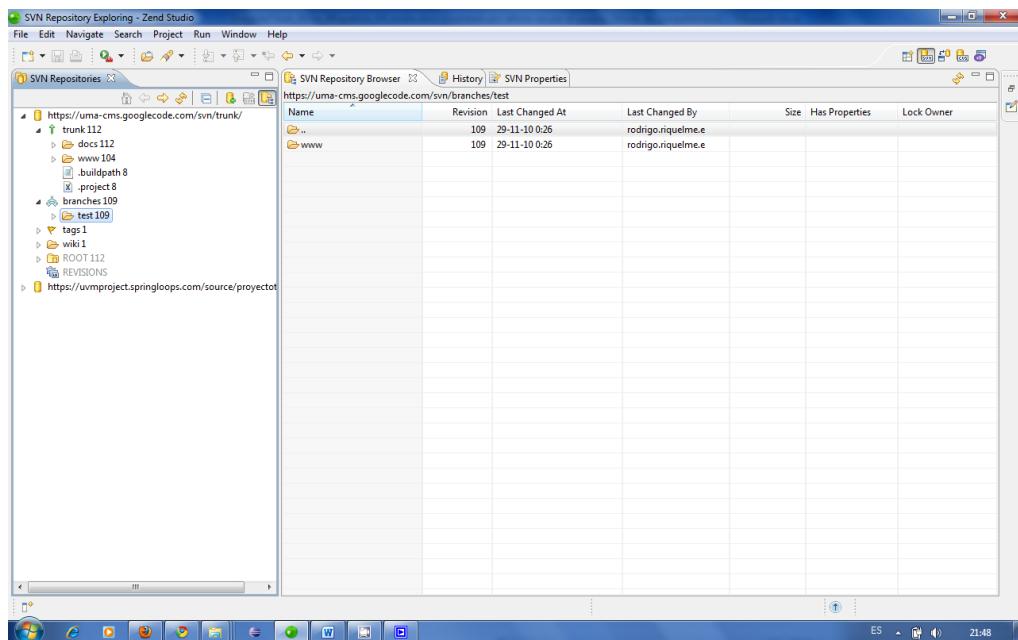


Ilustración 28 - Estructura de repositorio Subversion vista en Zend Studio

## 4.3. Diagrama de Datos

En esta fase se diseñara una base para el framework y CMS lo que incluye páginas base, videos, videos destacados, miniaturas, tags, categorías, players, tipos de video (conversiones) además menús y configuración del sitio.

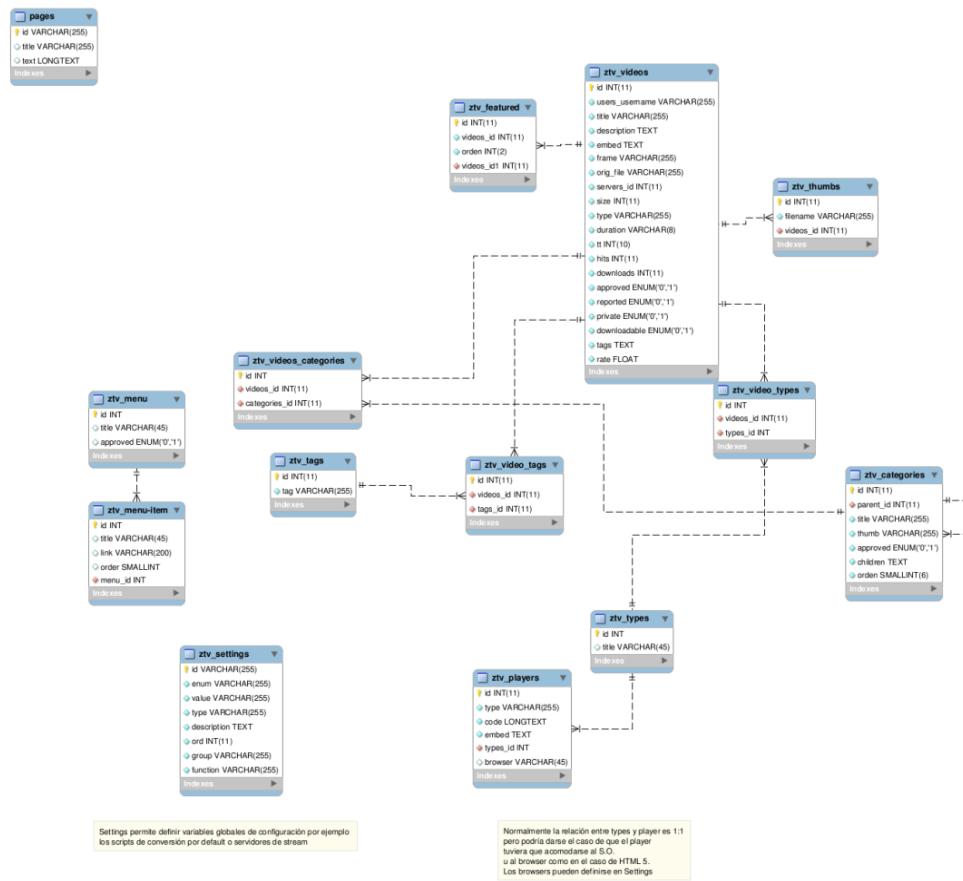


Ilustración 29 - Diagrama de datos primera fase de desarrollo

## 4.4. Diagrama de Clases

Como se mencionó anteriormente el Framework y el CMS está compuesto por clases agrupadas Namespaces (que sería lo equivalente a un Package en Java). Los Namespaces son soportados por la última versión de PHP a la fecha la cual es la 5.3.

En la codificación de las clases deben insertarse comentarios según la convenciones de PHPDocs los cuales están basados en el estándar de Java Docs, esto permite autogenerar la documentación en formato web.

El Core de la aplicación estará basado en los siguientes Namespaces

#### 4.4.1. Namespace Models

Este namespace o package es el componente encargado de manejar la capa de datos, y de la comunicación con la BD mediante scripts SQL.

La clase base o super clase es llamada **MModel** la cual a su vez es una implementación de interface **IModel** la cual implementa los siguientes métodos:

**IModel::add()**: Añade registros al objeto mediante la clausula SQL “INSERT”.

**IModel::addOrder()**: Reordena un recordset según un campo determinado, puede ser instanciado varias veces para hacer un orden por múltiples campos, esta asociado a la clausula SQL “ORDER BY”.

**IModel::countAll()**: Calcula el numero de filas totales de la RecordSet para paginación.

**IModel::delete()**: Ejecuta queries DELETE.

**IModel::getId()**: Retorna todas las claves primarias del objeto recordset

**IModel::getLimit()**: Retorna el valor numérico del puntero la consulta a ejecutar.

**IModel::getOrder()**: Retorna los campos de orden.

**IModel::getSize()**: Retorna el numero de filas de RecordSet.

**IModel::getStart()**: Retorna el puntero del primer registro del RecordSet.

**IModel::load():** Ejecuta la consulta SQL y crea el objeto RecordSet en memoria.

**IModel::next():** Mueve el puntero del objeto modelo al siguiente registro.

**IModel::reset():** Mueve el puntero al principio del RecordSet .

**IModel::setLimit():** Configura el límite de la consulta a ejecutar.

**IModel::setOrder():** Ordena el RecordSet según los campos y orden especificado.

**IModel::setStart():** Mueve el puntero del RecordSet sobre el cual se obtendrán los registros.

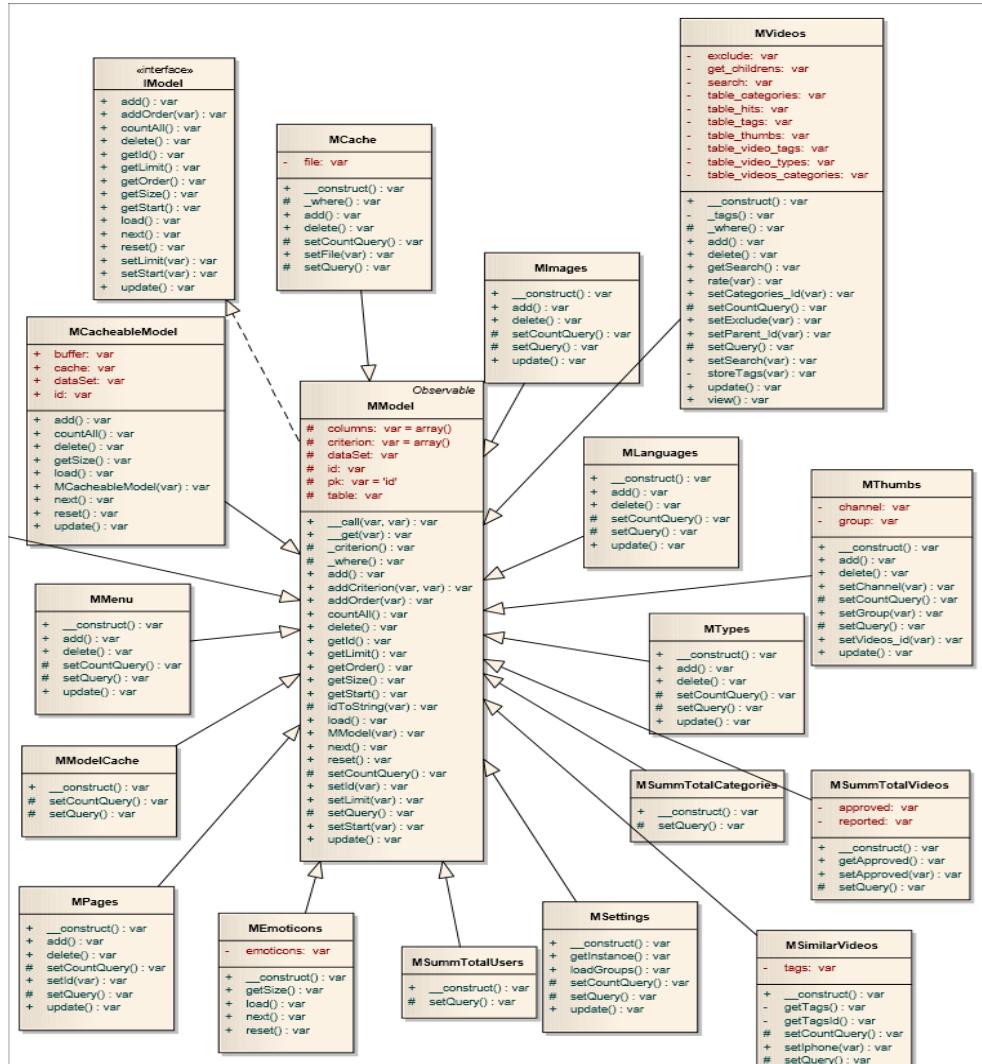


Ilustración 30 – Namespace Models - Parte 1

Otra super clase importante dentro de éste namespace es llamada **MBufferedModel** la cual hereda los atributos **MModel** pero además guarda el RecordSet en un buffer de memoria para ser usado en multiples consultas SQL.

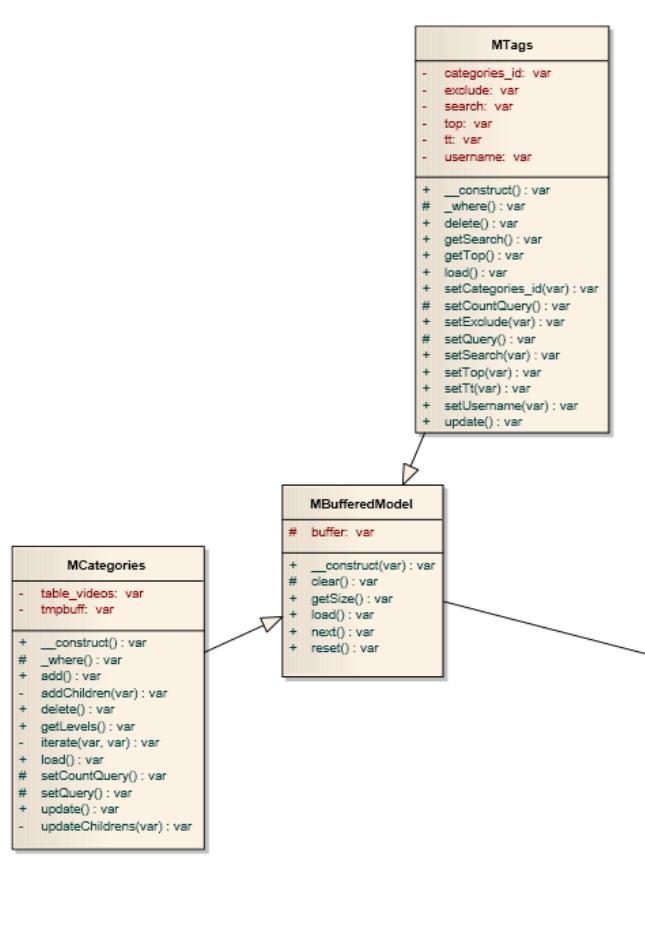
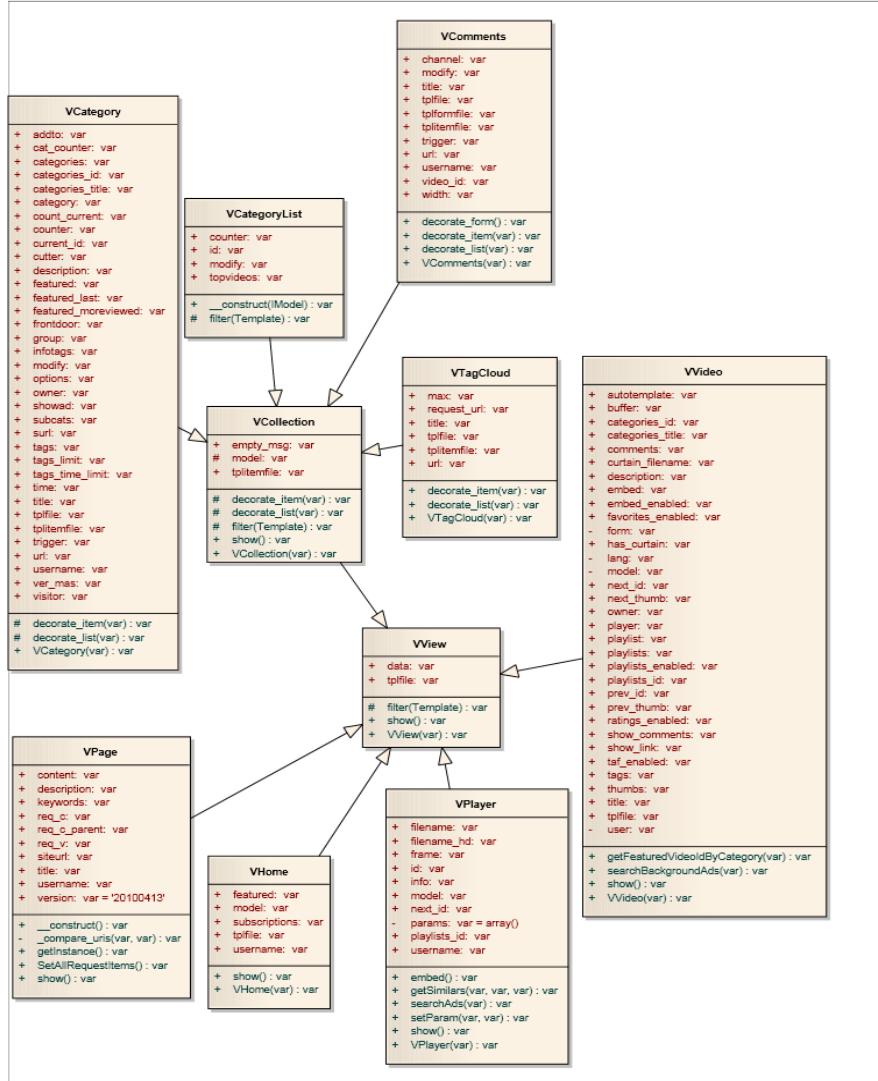


Ilustración 31 - Namespace Models - Parte 2

#### 4.4.2. Namespace Views

En este marco de trabajo este package tiene un papel puente entre la presentación y la capa de datos, para la instancia de un objeto View se usa generalmente un objeto Model como parámetro, el objeto View se encarga de asignar la plantilla o template HTML, rutear los atributos del objeto Model a variables de template y de asignar otros atributos de template en caso de ser necesario.

La clase base o super clase es llamada **VView**.



**Ilustración 32 - Namespace Views**

#### 4.4.3. Namespace Controllers

En el front office todo pasa por un objeto Controller para decidir cual componente Modelo-Vista deberá ejecutarse.

En estricto rigor un Controller se encarga de inicializar un modelo y una vista y asociarlos según parámetros GET o POST que sean ingresados vía web, sin embargo se deja abierta la posibilidad de ejecutar cualquier script web para así poder satisfacer cualquier requerimiento no contemplado en etapas anteriores del desarrollo y que pudiera surgir en el futuro.

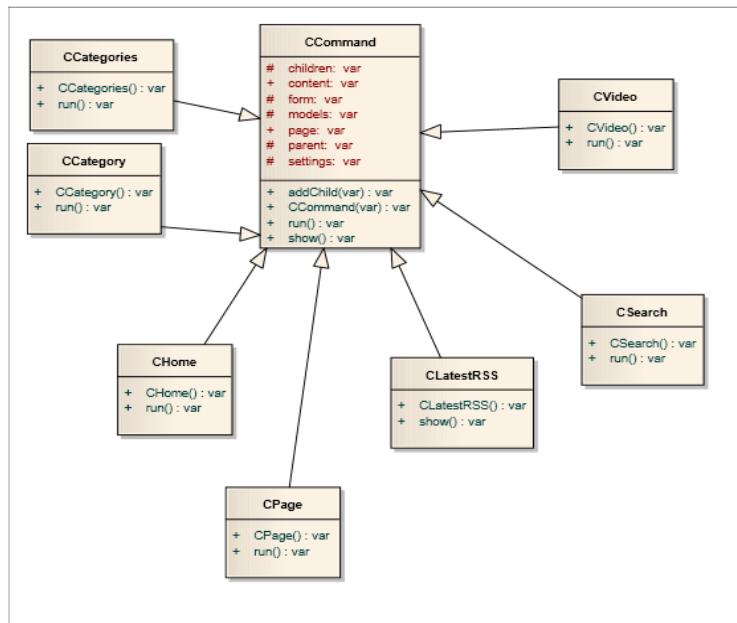


Ilustración 33 - Namespace Controllers

#### 4.4.4. Namespace Admin

Este paquete representa los datos del CMS en elementos de lista y formularios para ser llamados en el back office y simplificar y homogenizar las interfaces que usadas para hacer operaciones CRUD (Create, Read, Update, Delete).

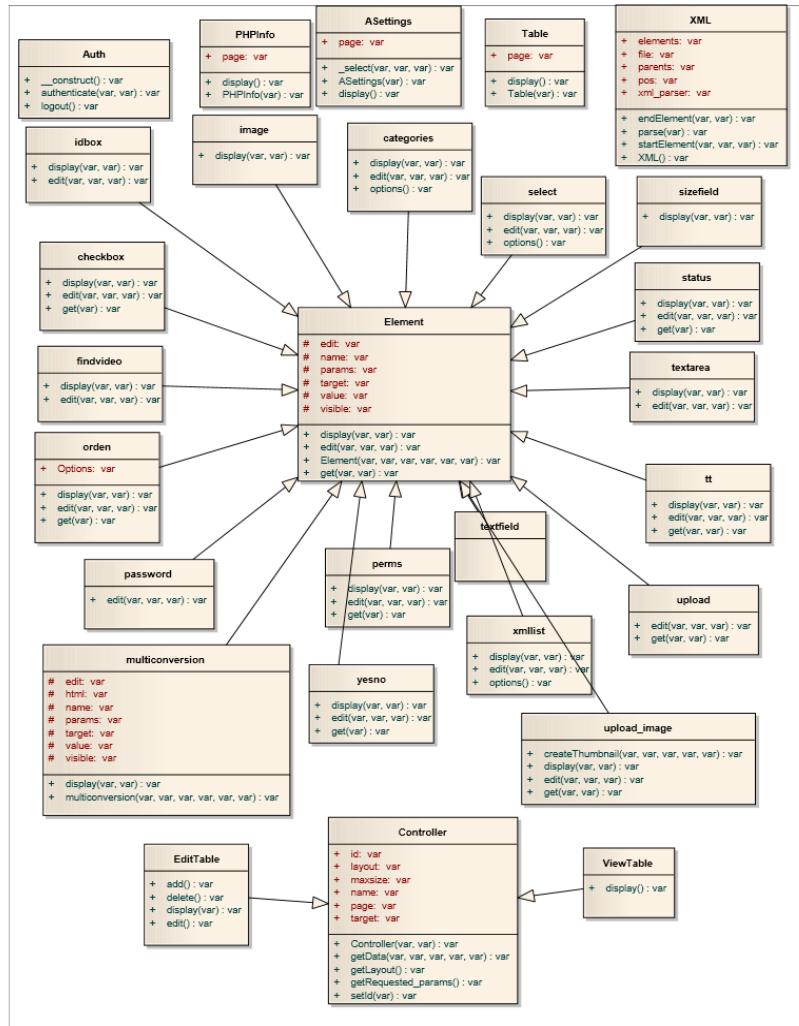


Ilustración 34 - Namespace Admin

**Profesor:**

Dahianna Vega L.

Página 101 de 124

**Alumnos:**

Rogelio Elías, Rodrigo Riquelme, Manuel Canales

**Tema:**

Marco de Trabajo para un Acceso Multimedia Universal Mediante Patrones Modelo-Vista-Controlador

#### 4.4.5. Namespace Lib

Este package contiene una colección de librerías de uso comun para ofrecer funcionalides diversas tales como: armar consultas SQL en base a parámetros, ofrecer conexión a Base de datos, funciones de fecha/hora, manejar a FFMpeg, generar HTML personalizado entre otras.

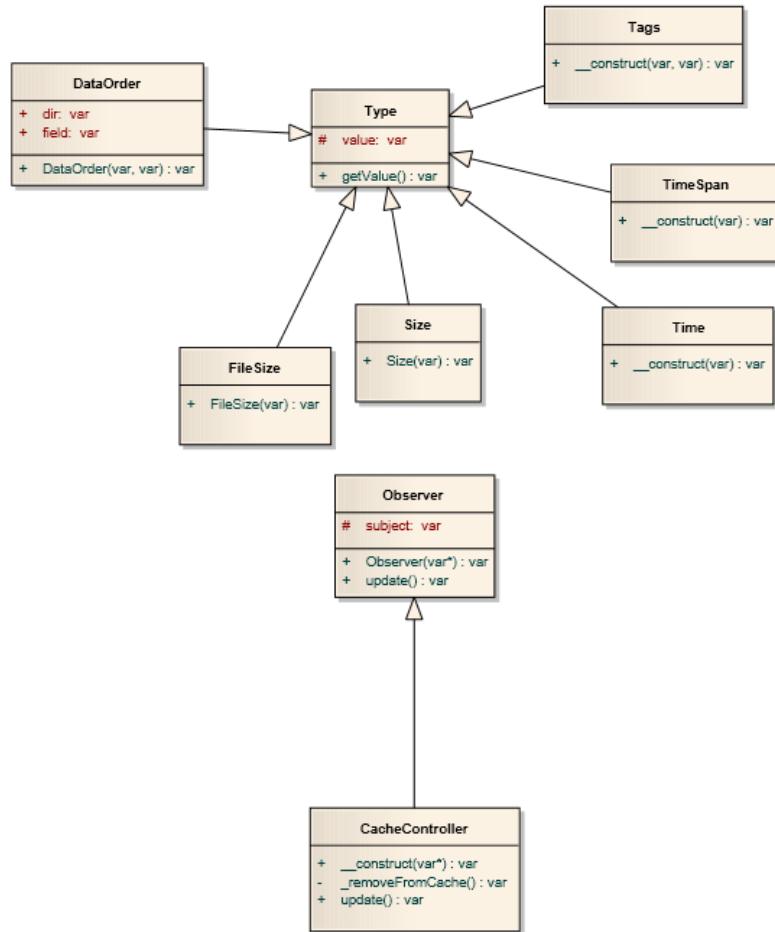


Ilustración 35 - Namespace Lib - Parte 1

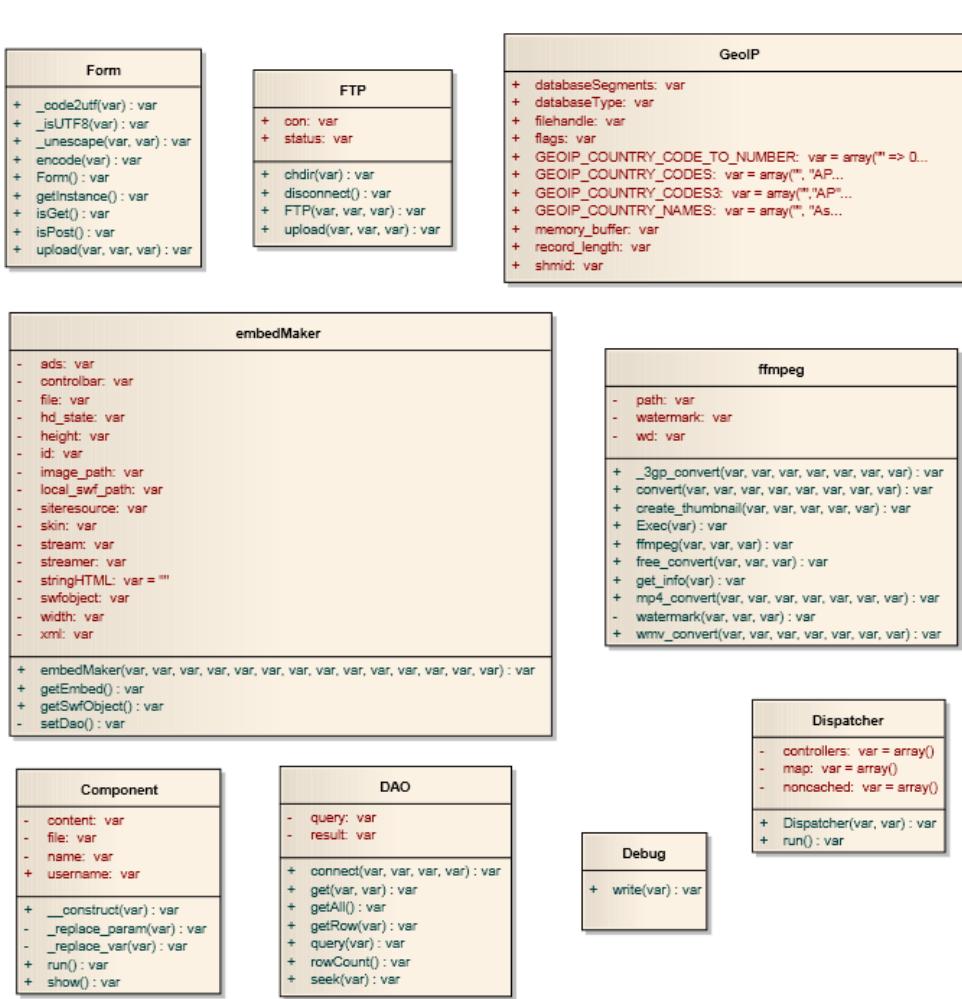


Ilustración 36 - Namespace Lib - Parte 2

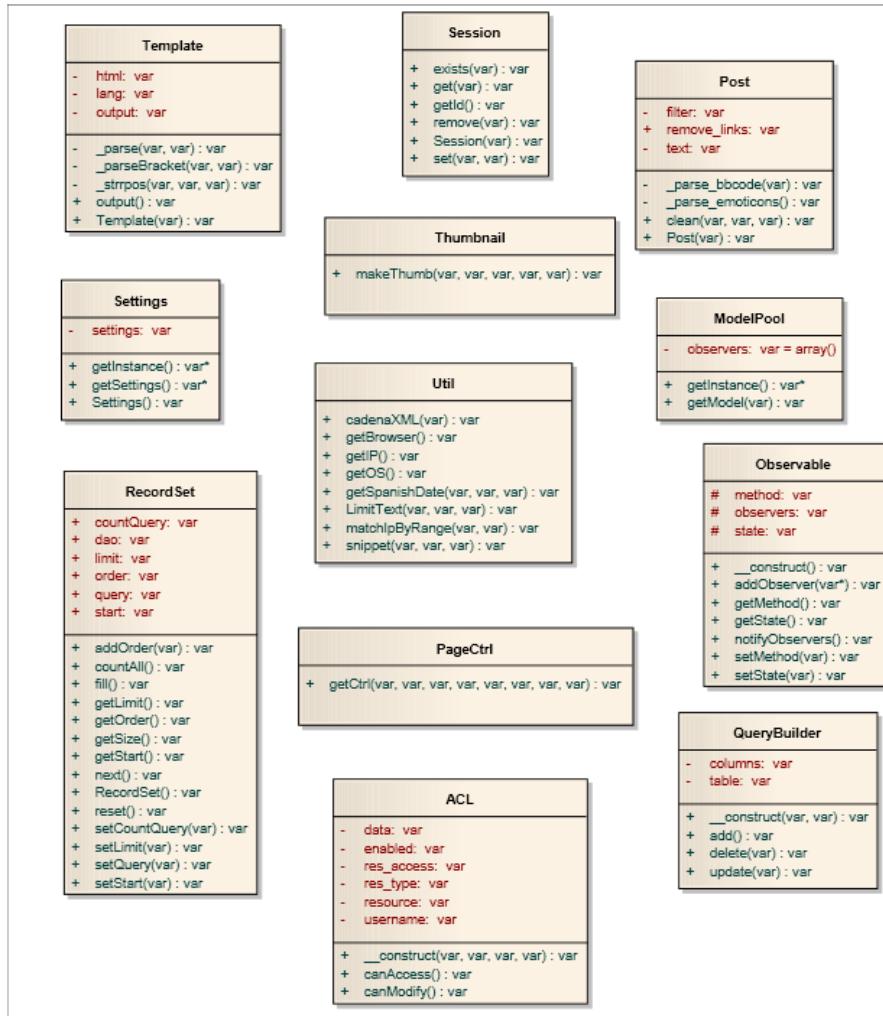


Ilustración 37 - Namespace Lib - Parte 3

**Profesor:**

Dahianna Vega L.

Página 105 de 124

**Alumnos:**

Rogelio Elías, Rodrigo Riquelme, Manuel Canales

**Tema:**

Marco de Trabajo para un Acceso Multimedia Universal Mediante Patrones Modelo-Vista-Controlador

## 4.5. Especificaciones de desarrollo Back Office

El backoffice, como en la mayoría de los CMS puede resumirse su funcionalidad al acrónimo CRUD (Create, Read, Update, Delete).

En un CRUD web los formularios son componentes recurrentes, y como actúan sobre un modelo de datos relacional, lo más frecuente es que tiendan a repetirse los elementos de formularios en diferentes instancias.

Estos elementos de formulario tienen atributos y métodos por lo tanto lo más lógico es encapsular los elementos de formularios en clases, para maximizar la escalabilidad y portabilidad de estos elementos se creará una capa XML que interprete estas clases y en los cuales se ingresen directamente los atributos y métodos de los elementos de formulario.

### 4.5.1. Configuración de Sitio

Tambien se creará una clase Settings que maneje toda la configuración global del sitio, todos estos parámetros de configuración se guardarán en una sola tabla, al ser alterada esta tabla, agregando o quitando campos debiera reflejarse este cambio en forma automática en el formulario de configuración del CMS.

## 4.5.2. Componentes XML

Los componentes XML están compuestos de secciones, lo más común es que una sección represente a un formulario y mantenga a una tabla específica para este caso se especifica el type="table" en este caso también se especifica si los elementos tienen permisos de edit, add y delete.

En este ejemplo tenemos un sección Tipo "Table" (clase namespace Admin) que puede ser crear, editar y borrar objetos de la clase MMenu.

```
<?xmlversion="1.0"?>
<sectionname="Menu" type="table" target="MMenu" edit="true" add="true" delete="true">
<fieldName="ID" target="id" type="idbox" visible="true" edit="true" add="false"/>
<fieldName="Title" target="title" type="textfield" link="index.php?p=menu&parent_id={id}" visible="true" edit="true" add="true"/>
<fieldName="URL" target="url" type="textfield" visible="true" edit="true" add="true"/>
<fieldName="Parent" target="parent_id" type="select" table="menu" field="title" visible="true" edit="true" add="true"/>
<fieldName="Order" target="menu_order" type="textfield" visible="true" edit="true" add="true"/>
<fieldName="Header" target="header" type="yesno" visible="true" edit="true" add="true"/>
<fieldName="Footer" target="footer" type="yesno" visible="true" edit="true" add="true"/>
<fieldName="Approved" target="approved" type="yesno" visible="true" edit="true" add="true"/>
</section>
```

En este otro ejemplo tenemos un menú con la configuración del sitio, type representa a la clase del package admin y Msettings a la clase del package Models.

```
<?xmlversion="1.0"?>
<sectionname="Settings" type="asettings" target="MSettings"/>
```

## 4.6. Especificaciones Front Office

En el Front Office se compone de templates HTML (en general) y componentes XML.

Todas las páginas del front office están basadas en un template común llamado index.html a no ser que por parámetro llamado **tpl** sea basado en otro template especificado en ese parámetro más la extensión .html. Esto es especificado en la clase Vista principal la cual es llamada **VPage**.

Para la funcionalidad de templates y la de componentes xml se usa clases **Template** y **Component** del namespace **lib**.

VPage tiene un atributo llamado **tpl** del tipo **Template**.

VPage retorna las variables de template:

**VPage::tpl::sitename:** Nombre del sitio según la configuración-

**VPage::tpl::menu:** Menú de navegación.

**VPage::tpl::content:** Contenido de la página, esta se configura según los parámetros de la URL los cuales inicializan un determinado controlador la cual a su vez llama a un nuevo template (por ejemplo home.html) con componentes MVC XML.

En el siguiente ejemplo **de index.html** (template principal) todas las variables encerradas entre corchetes [:] son variables de template de **VPage** (vista maestra)

```
index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
    <head>
        <title>[:sitename:] [:titleheader:]</title>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
    </head>
    <body>
        <div class="marco headerBg">
            <div id="header" class="span-24-2">
                <ul class="nav">
                    [:menu:]
                </ul>
            </div>
        </div>
        <div id="contenido">
            [:content:]
        </div>
        <div class="footer oscuro">
            <p class="pie">Powered by UMA CMS - Universal Media
Access Content Management System </div>
            </p>
        </div>
        <div id="popup_body"></div>
    </div>
</body>
</html>
```

En el siguiente ejemplo se supondrá que el controlador principal llama al home. Llamaremos a este controlador **CHome**, el cual llamará a un template llamado home y lo pondrá en la variable **VPage::tpl::content** la cual se ve en el template **index.html** como **[:content:]**.

```
home.html
[:main_video:]
<div class="span-24-2">
    <div class="span-24">
        <!-- the tabs -->
        <ul class="losmas">
            <li><a href="#">Lo + Reciente</a></li>
            <li><a href="#">Lo + Visto</a></li>
        </ul>
        <!-- tab "panes" -->

        <div class="contelosmas">
            <div>[:latest_videos:]</div>
            <div>[:mostviewed_videos:]</div>
        </div>
    </div>
<div class="span-24 boxTv box2">
    [:boxcanal:]
</div>
```

En este template de nivel dos ya incluye componentes XML encerrados entre corchetes [:] estos son.

**mainvideo.xml, latest\_videos.xml, mostviewed\_videos.xml, boxcanal.xml.**

Para exemplificar de que se tratan estos componentes XML mostraremos el componente **main\_video.xml** tentativo en el cual se asocia el Modelo **MFeaturedVideos** con la Vista **VVideo**.

Las variables que comienzan con % son parámetros de URL.

En este ejemplo se devolverá un registro de la Clase **MFeaturedVideos**, el parámetro **%start** indicará cual de los registros obtener según el orden por el campo **orden** en forma ascendente “ORDER BY orden ASC” luego el RecordSet es procesado por **VVideo** y envía la salida al template **video\_box.html**

```
<?xml version="1.0"?>
<component name="Main Video">
    <model name="MFeaturedVideos">
        <order dir="ASC">orden</order>
        <limit>1</limit>
        <start>%start</start>
    </model>
    <view name="VVideo">
        <tplfile>video_box.html</tplfile>
        <show_comments>false</show_comments>
    </view>
</component>
```

Esta forma de trabajar con templates y componentes proporciona una gran versatilidad para escoger diferentes modelos, vistas y templates según las necesidades del agente usuario lo que facilita el desarrollo para el Acceso Multimedia Universal y se evita la necesidad de volver a programar en un lenguaje servidor específico.

## 4.7. Maquetas Funcionales Back Office.

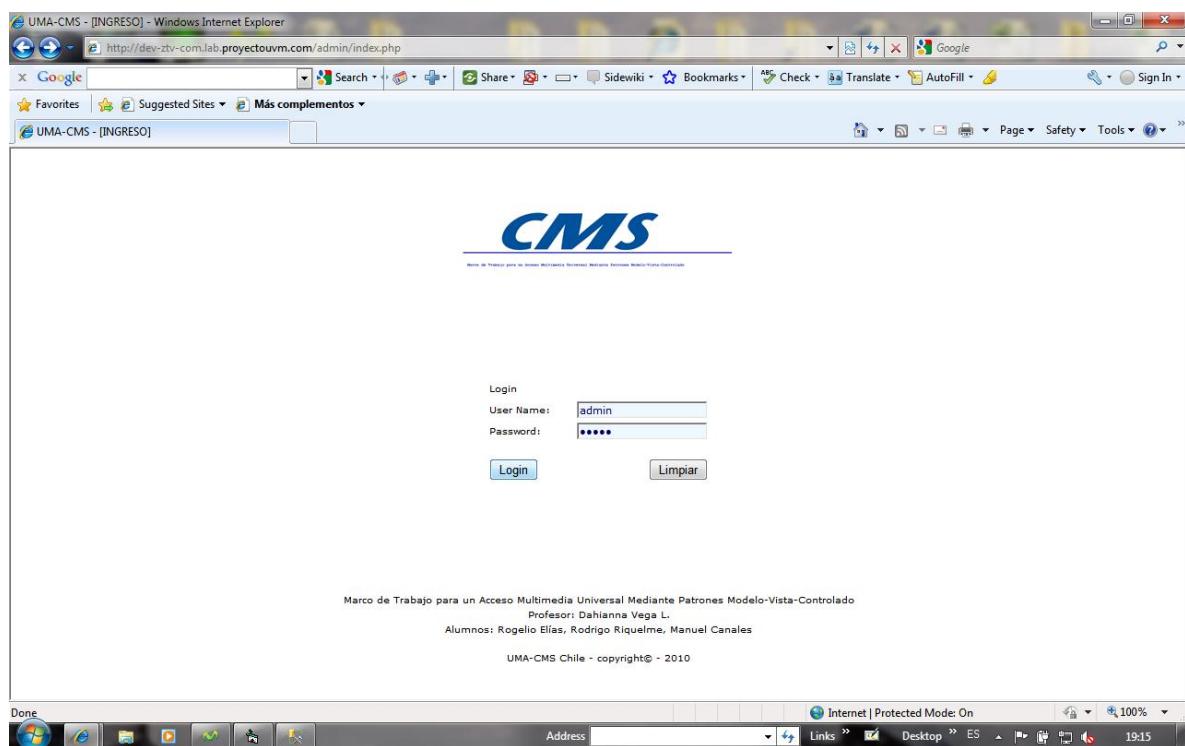


Ilustración 38 - Ingreso al Back Office

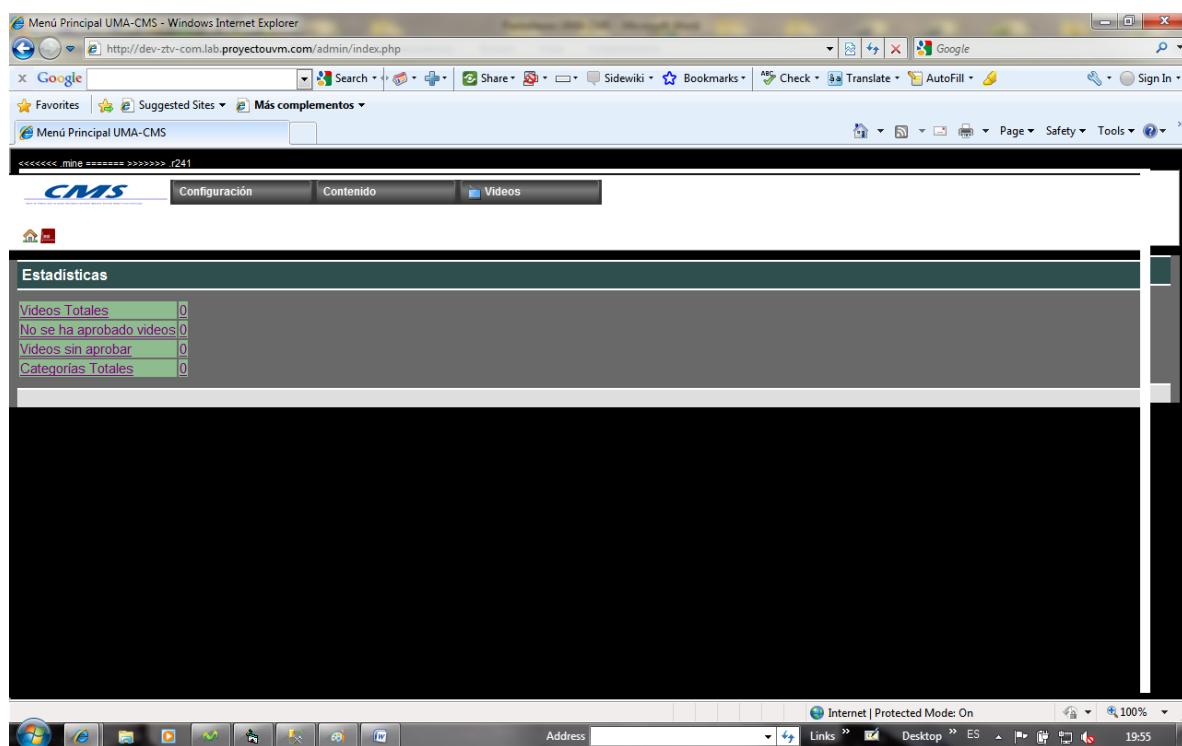


Ilustración 39 - Menú Principal

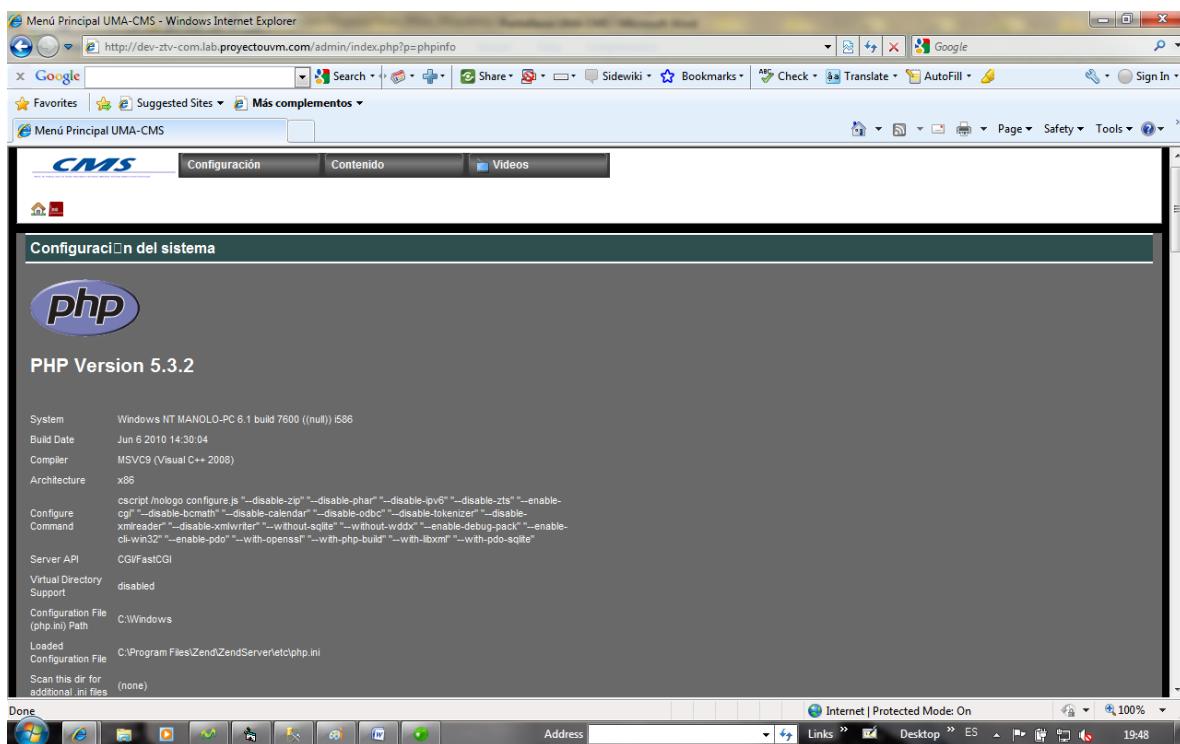


Ilustración 40 - Configuración del Servidor

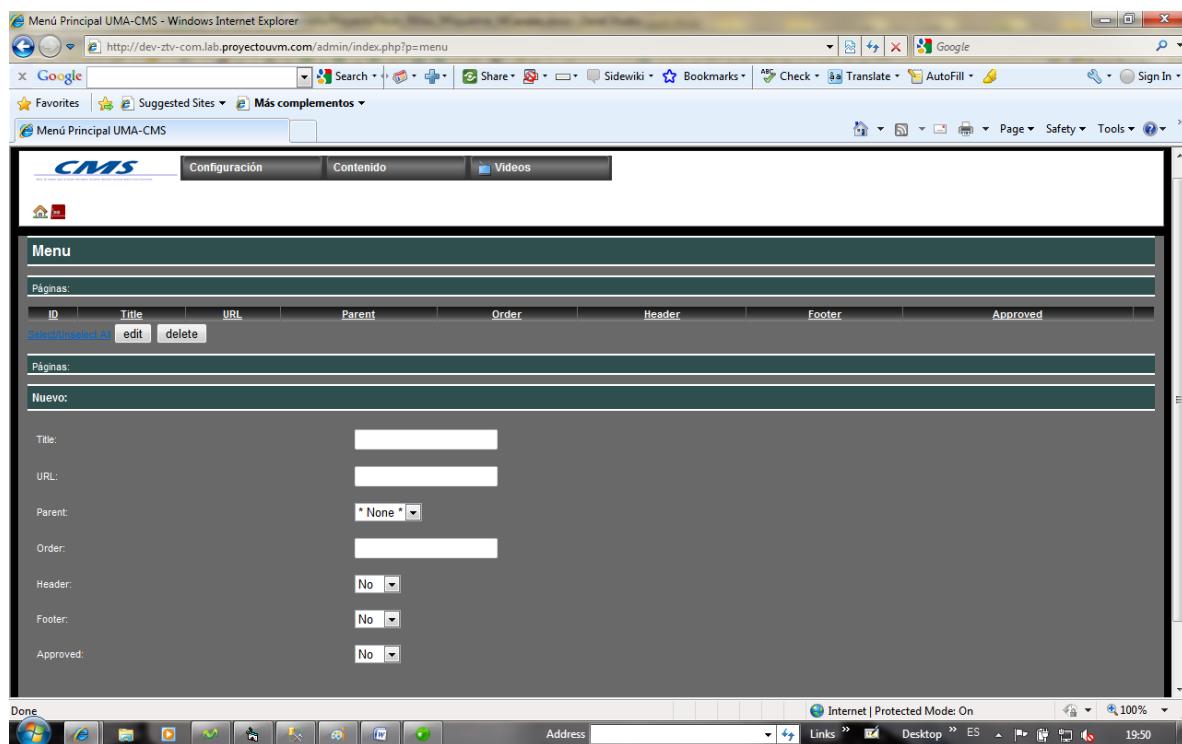


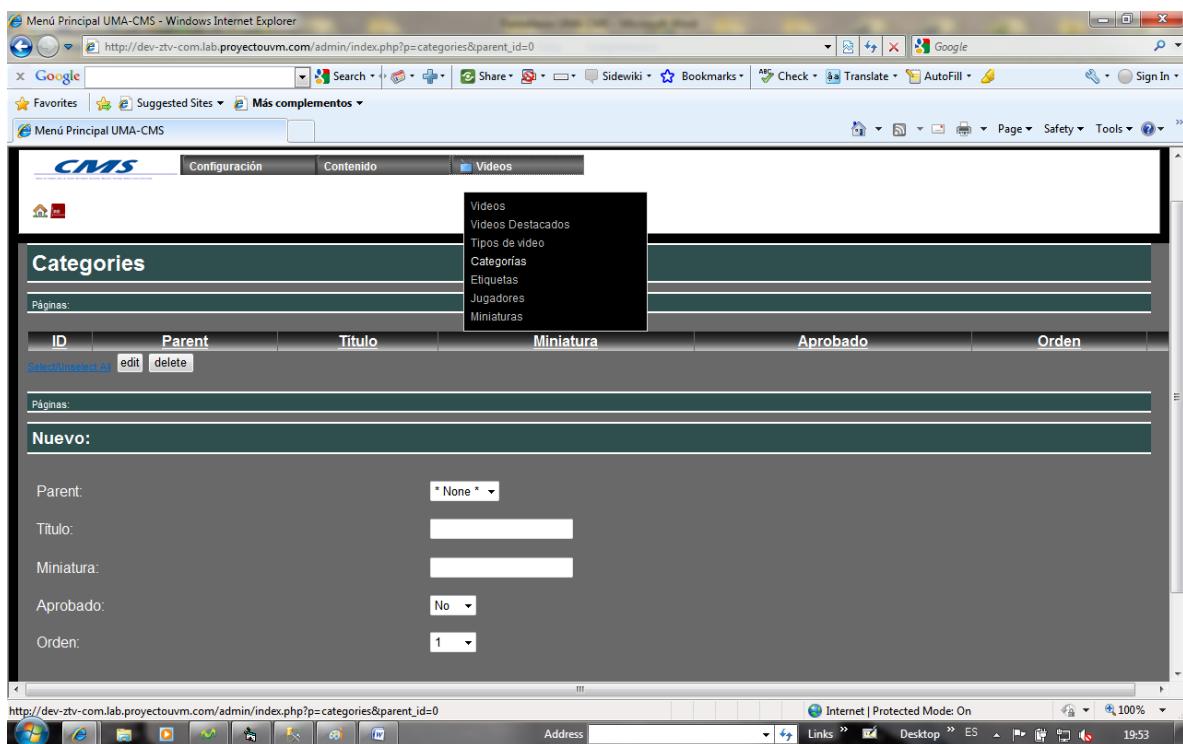
Ilustración 41 - Contenido Menú

The screenshot shows a Windows Internet Explorer window displaying a Content Management System (CMS) interface. The title bar reads "Menú Principal UMA-CMS - Windows Internet Explorer". The address bar shows the URL "http://dev-ztv-com.lab.proyectouv.com/admin/index.php?p=pages". The page content is a table titled "Pages" with columns: ID, Page, Title, URL, and Content. There are three rows of data:

ID	Page	Title	URL	Content
	category	category	index.php?m=page&p=category	
	home	home	index.php?m=page&p=home	
	video	video	index.php?m=page&p=video	

At the bottom of the table, there are buttons for "Select/Unselect All", "edit", and "delete". The status bar at the bottom of the browser window shows the URL "http://dev-ztv-com.lab.proyectouv.com/admin/index.php?p=pages", the system tray icons, and the address bar with "Internet | Protected Mode: On".

Ilustración 412 - Contenido Páginas



The screenshot shows a Windows Internet Explorer window displaying a CMS administration interface. The title bar reads "Menú Principal UMA-CMS - Windows Internet Explorer". The address bar shows the URL "http://dev-ztv-com.lab.proyectouvm.com/admin/index.php?p=categories&parent\_id=0". The main content area has a dark header with tabs: "CMS", "Configuración", "Contenido", and "Videos" (which is highlighted). A sidebar on the left lists categories like "Videos", "Videos Destacados", "Tipos de video", "Categorías", "Etiquetas", "Jugadores", and "Miniaturas". The main table below is titled "Categories" and has columns: "ID", "Parent", "Título", "Miniatura", "Aprobado", and "Orden". A sub-table for "Nuevo:" shows fields: "Parent" (set to "None"), "Título" (empty), "Miniatura" (empty), "Aprobado" (set to "No"), and "Orden" (set to "1"). At the bottom of the page, the URL "http://dev-ztv-com.lab.proyectouvm.com/admin/index.php?p=categories&parent\_id=0" is repeated in the address bar, along with browser status information like "Internet | Protected Mode: On" and "100%".

**Ilustración 42 - Videos**

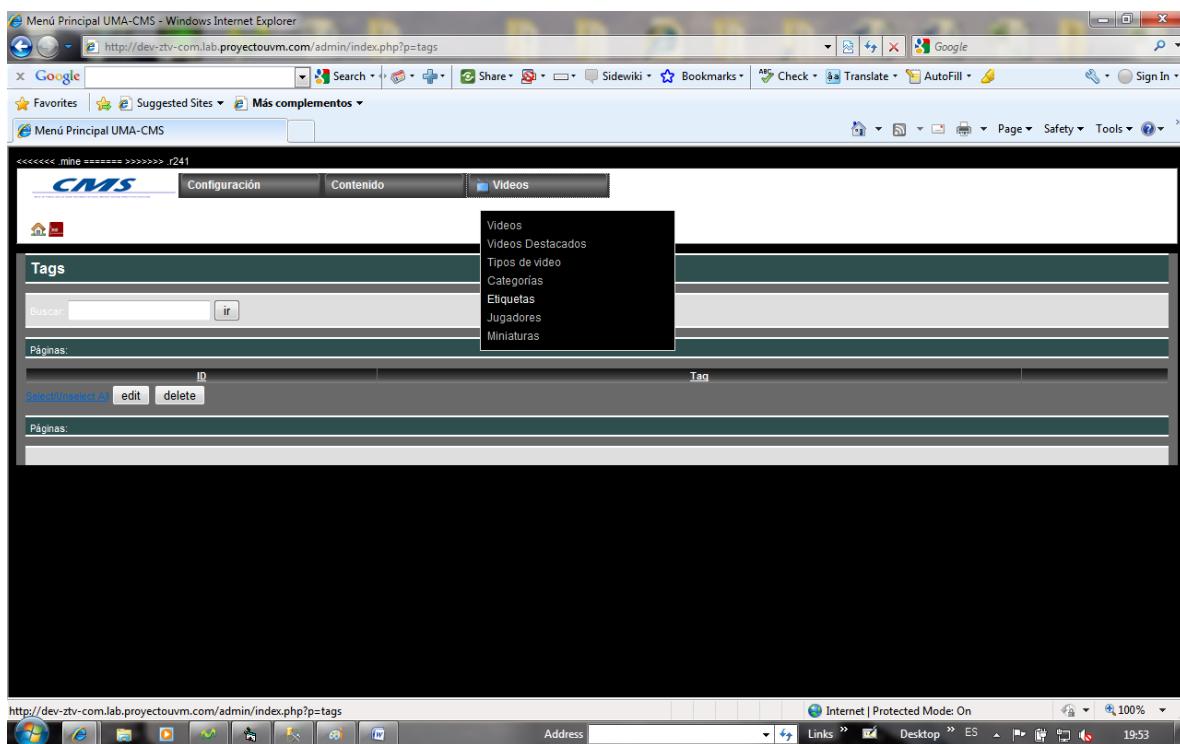


Ilustración 43 - Tipos de Videos

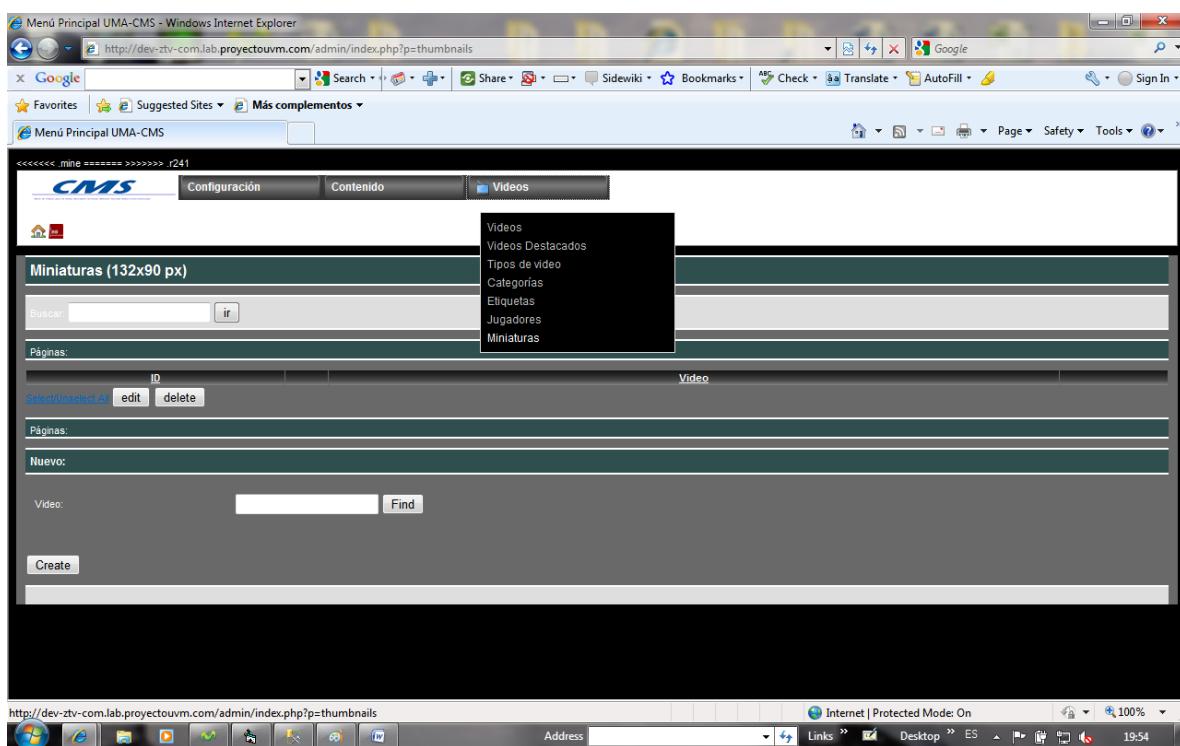


Ilustración 44 - Miniaturas

## 5. Bibliografía

---

### a) Libros

**“Feature Driven Development A Human-Powered Methodology for Small Teams”.** Autor: Alistair Cockburn- Octubre 2004.

**“Bibliografía, UML y Patrones, segunda Edición”, editorial Prentice Hall**

Autor: Craig Larman.

**“Desde el Desarrollo Evolutivo a las Metodologías Ágiles”, Introducción a la Gestión de Calidad de Software** Autor: Pablo Straub- 2008.

**“The Cathedral & The Bazaar”, editorial O'Reilly Media** Autor: Eric S. Raymond - 2001.

b) Sitios Web

**FFmpeg, FFmpeg Project** <http://www.ffmpeg.org/>(22 Marzo 2010)

**UMA, Wikipedia** [http://es.wikipedia.org/wiki/Acceso\\_Multimedia\\_Universal](http://es.wikipedia.org/wiki/Acceso_Multimedia_Universal)(02 de Mayo 2010)

**Manifiesto Ágil, dosideas.com** <http://www.dosideas.com/wiki/Agil> (20 de Abril de 2010)

**Google Web Toolkit (GWT)**<http://code.google.com/intl/es/webtoolkit/>

**Internet Protocol Televisión (IPTV)**<http://es.wikipedia.org/wiki/IPTV>

**Google TV**<http://www.google.com/tv/>

## Glosario

**Codecs:** Codificador/decodificador, especificación que utiliza un dispositivo o programa para realizar transformaciones bidireccionales sobre datos y señales.

**Feedback:** La realimentación o retroalimentación, comunicación de ida y devuelta.

**Frameworks:** Marco de Trabajo, conjunto de herramientas y reglas de desarrollo.

**Javascript:** Lenguaje de scripting ejecutado en el navegador.

**Protocolo:** Conjunto de reglas y normas usadas para la comunicación, conexión, trasmisión y transferencia de datos de forma estandarizada.

**Streaming:** Sistema de envío continuo de información, que permite, por ejemplo, ver un video a medida que se baja de la Red.

**TI:** Término utilizado para referirse a las tecnologías de la información.

**Transmoding:** Término utilizado en las técnicas de transcodificación de contenido y se refiere a la velocidad de reproducción o cambio de modalidad.

## Acrónimos

---

**AJAX:** Asynchronous JavaScript and XML.

**API:** Application Programming Interface.

**CMS:** Content Management System.

**GPL:** Licencia Pública General.

**HD:** Alta definición.

**HTML:** Lenguaje de Marcado de Hipertexto.

**HTTP:** Protocolo de transferencia de hipertexto.

**IBM:** International Business Machines.

**IP:** Protocolo de Internet.

**JSP:** Páginas de Servidor Java.

**MVC:** Modelo Vista Controlador.

**PHP:** Lenguaje de programación interpretado.

**PSP:** PlayStation Portable.

**REST:** Representational State Transfer.

**RPC:** Es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.

**RSS:** Site Sumary or Rich Site Sumary.

**SCRUM:** Metodología para la gestión y desarrollo de software basada en un proceso iterativo e incremental.

**SOAP:** Simple Object Access Protocol.

**SVQ:** Sorenson video Quantizer.

**TCP:** Protocolo de control de Transmisión.

**URL:** Localizadores uniformes de recursos.

**W3C:** World Wide Web Consortium.

**XML:** Extensible Markup Language.

**XP:** Programación extrema (metodología de desarrollo de software).