

# **Escola Universitària Politécnica de Mataró**

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA  
DE CATALUNYA**

**Ingeniería Técnica de Telecomunicación, especialidad Telemática**

**DIFUSIÓN DE VIDEO STREAMING CON FRAMEWORK SYMFONY Y HTML5**

**EDUARDO JUSTICIA DIAZ  
PONENTE: JOSEP MARIA GABRIEL SOLANILLA**

**PRIMAVERA 2011**



**TecnoCampus  
Mataró-Maresme**



## **Dedicatoria**

Este proyecto lo dedico a mi familia por haberme traspasado los valores que me han servido para llegar hasta aquí, a mis amigos por haber sido un punto de apoyo fundamental durante estos años, pero sobretodo a mi pareja por haber lidiado conmigo en estos años tan complicados, parte de este éxito también es suyo.



## **Agradecimientos**

Quiero agradecer a mis amigos el apoyo que me han dado en este proyecto, ya sea por sus ideas o inquietudes, a Alberto por ayudarme a entender Symfony y estar siempre que he tenido una consulta. A Roberto por contarme sus experiencias dentro del mundo del streaming, también a mi ponente Josep M<sup>a</sup> Gabriel por ser un buen guía durante este proyecto y saberme dar consejo.



## **Resum**

Aquest projecte presenta una proposta de servei de lloguer de series de televisió a través d'Internet mitjançant les últimes tecnologies com HTML5 i el framework Symfony. Amb el suport de l'arquitectura streaming Unicast basat en el projecte lliure Flumotion, s'ha dissenyat una solució que podria ser hàbil dins del marc legal actual de difusió de continguts. Aquest projecte també realitza un estudi de mercat i analitza les principals eines de difusió de continguts més utilitzades en Internet.

## **Resumen**

Este proyecto presenta una propuesta de servicio de alquiler de series de televisión, a través de Internet, mediante las últimas tecnologías como HTML5 y el framework Symfony. Con el aporte de una arquitectura de streaming Unicast, basada en el proyecto libre Flumotion, se ha diseñado una solución que podría ser utilizada dentro del marco legal actual de difusión de contenidos. Este proyecto además realiza un estudio del mercado y analiza las principales herramientas de difusión de contenidos más utilizadas en Internet.

## **Abstract**

This project seeks to provide a service solution to rent TV shows over the Internet using the latest technologies such as HTML5 and the symfony framework. With input from Unicast streaming architecture based on free Flumotion project, has designed a solution that could be working within the current legal framework for the diffusion of content. This project also carries out market research and analyzes the main tools used for distributing content over the Internet.



# Índice.

Índice de figuras.....	III
Índice de tablas.....	V
Glosario de términos.....	VII
Introducción: .....	1
-Que es PoNNLo? , Que pretende?.....	1
-Objetivos.....	1
1. Antecedentes .....	3
1.1. Contexto Actual .....	3
1.1.1. Visión del mercado audiovisual en Internet .....	3
1.1.2. Introducción de la ley Sinde.....	5
1.1.3. Contenidos digitales en Internet .....	6
1.1.4. LSSI.....	7
1.1.5. Propiedad Intelectual.....	8
1.2. Problemática .....	9
1.2.1. Disolución de páginas ilegales .....	9
1.3. Estudio del Mercado .....	10
1.3.1. Perfil del nuevo Usuario Audiovisual .....	10
1.3.2. Demanda actual .....	10
1.3.3. Oferta Actual .....	11
1.3.4. Casos de éxito en Internet .....	12
1.3.5. Mercado de servidores de streaming actual .....	14
1.3.6. Oportunidad de negocio .....	17
2. Tecnologías disponibles .....	19
2.1. Propuesta de solución .....	19
2.2. Arquitectura del sistema .....	20
2.3. Herramientas a utilizar .....	23
2.3.1. Elección del lenguaje .....	23
2.3.2. Elección de herramientas de trabajo.....	26
2.3.3. FrameWork Symfony .....	29
2.3.4. Información referente a Firefogg .....	34
2.4. Elección del formato de video .....	34

2.4.1 Características principales de los formatos de video más utilizados.....	35
2.5. Uso de HTML5 en video .....	38
3. Solución aportada.....	43
3.1. Entorno de Desarrollo.....	43
3.1.1. Preparación del entorno de desarrollo .....	43
3.2. Implementación .....	48
3.2.1. Definición de roles de Usuario.....	48
3.2.2. Configuración y creación del Modelo .....	50
3.2.3 Creación y configuración de módulos.....	56
3.2.4. Integrar seguridad en los módulos.....	72
3.2.5. Routing asociado .....	73
3.2.6. Aspectos personalizados CSS.....	74
4. Instalación .....	77
4.1. Configuración del servicio Flumotion.....	78
4.2. Uso del servicio .....	81
4.3. Estadísticas Flumotion.....	86
5. Valoración económica.....	91
5.1. Análisis, búsqueda de información y herramientas .....	91
5.1.1. Análisis y búsqueda de información.....	91
5.1.2. Herramientas y utilidades .....	92
5.2. Codificación.....	93
5.3. Documentación y pruebas .....	94
5.4. Hardware inicial .....	95
5.5. Valoración final .....	96
6. Conclusiones y posibles ampliaciones .....	97
6.1. Propuestas de Mejora y posibles ampliaciones .....	97
6.2. Conclusiones.....	100
7. Referencias. ....	103

## Índice de figuras.

Figura 1.1.1: Grafico de progresión audiencia televisión y Internet.....	4
Figura 1.3.5.1: Esquema básico difusión por streaming.....	14
Figura 1.3.5.2: Esquema lógico VLC.....	16
Figura 2.2.1: Diagrama conceptual de comportamiento.....	21
Figura 2.2.2: Arquitectura de video streaming .....	22
Figura 2.3.1.1: Esquema de funcionamiento PHP .....	25
Figura 2.3.2.1: Cronología versiones MySQL.....	27
Figura 2.3.2.2: Cronología versiones Netbeans .....	28
Figura 2.3.3.1: Concepto de modelo-vista-controlador .....	31
Figura 2.3.3.2: Concepto general ORM.....	32
Figura 2.4: Estadísticas Navegadores 2010 .....	35
Figura 2.5.1: Marco predefinido etiqueta <video> .....	41
Figura 3.1.1.1: Pantalla principal Symfony .....	45
Figura 3.1.1.2: Acceso a Opciones personalizadas Netbeans .....	46
Figura 3.1.1.3: Adaptación de framework Symfony a Netbeans.....	46
Figura 3.1.1.4: Configuración debugger PHP en Netbeans.....	47
Figura 3.2.2.1: Mapa relacional tratamiento de video .....	51
Figura 3.2.2.2: Acceso Plugin Doctrine desde MySQL Workbench.....	52
Figura 3.2.2.3: Relación de tablas usuario:.....	53
Figura 3.2.3.1: Esquema secciones de la página.....	60
Figura 3.2.3.2: Información personal reducida .....	60
Figura 3.2.3.3: Pantalla principal Home .....	62
Figura 3.2.3.4: Información Personal Usuario.....	64
Figura 3.2.3.5: Acceso a Series .....	65
Figura 3.2.3.6: Pantalla de listado de temporadas .....	67
Figura 3.2.3.7: Visualización de listado capítulos .....	67
Figura 3.2.3.8: Acceso a visualización del contenido.....	70
Figura 3.2.3.9: Acceso Entorno colaborador .....	72

Figura 4.1: Instalación Flumotion Synaptics .....	77
Figura 4.1.1: Arranque servicio Flumotion.....	78
Figura 4.1.2: Configuración como Manager Flumotion .....	78
Figura 4.1.3: Archivos de configuración.....	79
Figura 4.1.4: Archivos de configuración.....	79
Figura 4.1.5: Archivos de configuración.....	80
Figura 4.1.6: Configuración satisfactoria de Flumotion .....	80
Figura 4.1.7: Conexión Worker & Manager .....	81
Figura 4.2.1: Menú de Opciones Superior .....	82
Figura 4.2.2: Menú Conexión .....	82
Figura 4.2.3: Menú Administrar.....	83
Figura 4.2.4: Botones de administración básica.....	84
Figura 4.2.5: Menú conexiones recientes.....	85
Figura 4.2.6: Menú conexiones activas.....	85
Figura 4.3.1: Estadísticas de consumo por cliente del servicio streaming.....	86
Figura 4.3.2: Consumo de recursos hardware del servicio Flumotion .....	86

## Índice de tablas.

Tabla 4.2.1: Consumo de recursos hardware del servicio Flumoti.....	87
Tabla 4.2.2: Requisitos de almacenamiento para el servicio .....	88
Tabla 4.2.3: Tráfico estimado en el servicio según escenario .....	88
Tabla 4.2.4: Videos concurrentes por Ancho de banda estimado.....	89
Tabla 5.1.1: Coste del Análisis del Proyecto .....	92
Tabla 5.1.2: Coste de las herramientas utilizadas .....	92
Tabla 5.2.1: Coste de montaje del escenario.....	93
Tabla 5.2.2: Coste de codificación.....	94
Tabla 5.3: Coste de pruebas .....	95
Tabla 5.4: Coste Hardware Inicial .....	95
Tabla 5.5: Relación de costes.....	96



## Glosario de términos.

MVC	Modelo vista controlador
AIMC	Asociación para la investigación de medios de comunicación.
LSSI	Ley de Servicios de la sociedad de Información
CD	Compact Disc
VHS	Video Home System
<i>DVD</i>	Digital Versatile Disc
SD	Standard Definition
HD	High Definition
QTSS	QuickTime Streaming Server
RTP	Real Time Transport Protocol
RTSP	Real Time Streaming Protocol
JSP	JavaServer Pages
ID	Identificador
ORM	Object Relational Mapping
YAML	YAML Ain't Markup Language
PHP	HyperText PreProcessor
HTTP	HyperText Transfer Protocol
CSS	Cascading Style Sheet
ISP	Internet Service Provider



## **Introducción:**

### **-Que es PoNNLo? , Que pretende?**

PoNNLo es una plataforma Web que ofrecerá al usuario mediante una interfaz amigable y sencilla un servicio de acceso a contenido audiovisual, concretamente series de televisión. La idea de PoNNLo surgió durante una conversación informal entre colegas de profesión, donde se expresaba un cierto malestar por la dificultad de acceso a este tipo de contenido debido al escenario actual que se presenta dentro de este entorno y donde se llegó a la conclusión que esta industria necesitaba una ventana nueva al cliente a través del canal más usado en estos últimos años. Mediante tecnologías como Streaming y Unicast es posible pensar en una solución robusta que ofrezca garantías de difusión seguros de este tipo de contenido.

Lo que pretende PoNNLo es ofrecer una alternativa al usuario actual de Internet que quiera acceder a contenido audiovisual desde la red, cumpliendo con la normativa vigente de difusión de contenidos de autor. Ofreciendo por un precio razonable al acceso a cualquier contenido audiovisual que haya sido publicado por cualquier entidad propietaria del contenido. PoNNLo, a diferencia de otros servicios de acceso a contenidos ya caducos e ilegales, preservara la propiedad intelectual del contenido difundido ofreciendo además un acceso seguro e íntegro al mismo.

### **-Objetivos**

#### a) Seguridad

Debido a que el contenido que se retransmite a través de este servicio contiene copyright y derechos de autor, es obligatorio y fundamental que el acceso total al contenido esté protegido mediante contraseñas y sesiones cifradas, así como que no se pueda almacenar en los equipos en que se visualiza, evitando así la difusión del mismo de forma ilegal.

### b) Velocidad

Es evidente que una de las cosas importantes en Internet y sobre todo en los servicios de acceso a contenidos audiovisuales es la velocidad. Por ello la plataforma PoNNLo debe tener tiempos de carga de contenidos razonables y una navegación veloz.

### c) Simplicidad

Al igual que en el caso de la velocidad, la simplicidad es un aspecto también a tener en cuenta en este tipo de servicio. Está comprobado estadísticamente que aquellas páginas que tienen un contenido dispar y su acceso es muy engorroso, es decir, navegar por ventanas y ventanas hasta llegar al contenido, obtienen resultados negativos en cuanto a número de visitas y usuarios.

PoNNLo tratara de suprimir páginas intermedias gracias a los accesos directos mediante el buscador y a la navegación entre pocas pantallas, ofreciendo siempre la información clasificada y ordenada, preservando su integridad y su consistencia y permitiendo al usuario final poder acotar la búsqueda por diferentes campos.

# **1. Antecedentes**

## **1.1. Contexto Actual**

Internet modifica mercados, conductas y costumbres, se ha transformado en la herramienta de esta nueva era, como cada cosa nueva que se implementa en el día a día de nuestras vidas, modifica con el tiempo nuestro estilo de vida. El entretenimiento diario está siendo uno de los afectados y en los últimos años gracias a Internet se está viendo obligado a cambiar. Se dice que los cambios siempre son buenos y probablemente en Internet este la salvaguarda que necesite esta industrial.

### **1.1.1. Visión del mercado audiovisual en Internet**

El mercado en Internet está en continuo cambio y eso se ha demostrado en los últimos 10 años, donde su evolución ha tenido más celeridad. Con la introducción de las páginas web 2.0, los usuarios han cogido un papel más que relevante frente a la prestación de servicios ofertados, convirtiendo a Internet en un mercado más global y más social. Sin duda alguna, la creciente y satisfactoria respuesta que han tenido las distintas redes sociales que han nacido en los últimos 5 años (Facebook, Twitter, Tuenti, LinkedIn, etc.), ha dado una voz al internauta que antes tal vez no tenía. Este hecho, junto con la creciente demanda de contenidos audiovisuales a través de Internet, ha provocado que surjan multitud de portales y páginas web dedicadas a la publicación de contenidos o bien a establecer vínculos hacia las paginas donde se alojan.

El usuario actual pasa más horas conectado a un dispositivo con acceso a Internet, ya sea teléfono u ordenador, que mirando el propio televisor. Esto hace que la industria audiovisual se recicle y prepare contenidos para su difusión a través de internet sin descuidar el formato “antiguo”. El paso de Internet hacia un mercado muchísimo más globalizado, donde el usuario final decide u orienta el tipo de contenido de las paginas, hace que surjan algunas lagunas, agujeros legales que la ley actual no contempla.

No hace falta decir que el usuario final ha crecido informáticamente hablando, es más experimentado, más ágil a la hora de tratar la información que recibe de Internet y mucho más crítico con ella. Esto conlleva que las páginas de difusión de contenidos se esfuercen

por captar usuarios en línea en sus páginas, haciéndolas más amigables, con un contenido mayor y mejor ordenado.

Sin duda alguna es un mercado en auge y económicamente muy interesante según muestran las últimas encuestas publicadas en **PricewaterhouseCoopers**. Si alguien ha tenido algo que ver en toda esta transformación y este nuevo enfoque, ése es Youtube, la primera plataforma de contenidos audiovisuales de Internet, que tiene hoy en día una gran aceptación, como ponen de manifiesto las estadísticas de este portal ofrecidas por la misma agencia:

**300 millones de visitas al mes**

**5000 millones de videos streams al mes 40% de los videos totales**

**15 horas de video subidas a la web por minuto**

En el grafico que se refleja en la imagen 1.1.1, cedida por la Asociación para la investigación de Medios de Comunicación (AIMC), se puede observar la creciente demanda de contenidos audiovisuales en Internet respecto a la televisión.

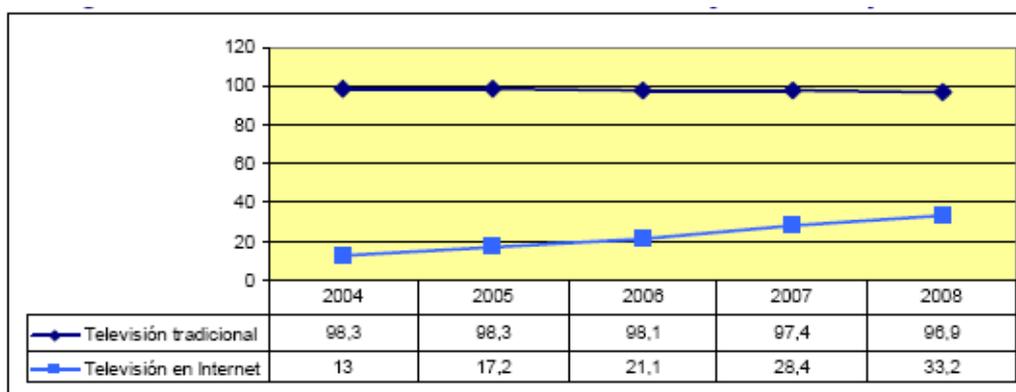


Figura 1.1.1: Grafico de progresión audiencia televisión y Internet

La industria audiovisual también ha salido muy beneficiada con la creciente demanda de tecnología móvil y redes de datos 3G. Estudios realizados en el año 2009, indican que la penetración de la tecnología móvil en el mercado español fue del **111.11%**, habiendo más teléfonos móviles que habitantes. Un **35%** de la población afirma haber consumido Internet a través del uso de los canales de datos.

Los datos reflejan la excelente vía de negocio que está teniendo la industria audiovisual a su alcance con la entrada en Internet. Este fenómeno es más acentuado en la población joven que según encuestas de la AIMC, en 2008 se obtuvo que el 82% de los jóvenes se conecten entre 5 y 7 días a la semana mientras tan solo un 77% vio la televisión. Son datos que revelan la creciente demanda de contenidos por Internet.

### **1.1.2. Introducción de la ley Sinde**

En los últimos años se ha estado elaborando un documento, que actualmente ese ha convertido en un apartado de la vigente y actual *Ley de Economía Sostenible*, la *Disposición Final Segunda*.

El objetivo fundamental de este apartado promovido por la ministra de cultura Ángeles González Sinde, es el de modificar el siguiente conjunto de leyes:

*-Ley de Servicios de la Sociedad de la Información*

*-Ley de Propiedad Intelectual*

*-Ley Reguladora de la Jurisdicción Contencioso-Administrativa*

Fundamentalmente, el objetivo de esta ley es el de otorgar a un organismo (comisión cultural) dependiente del ministerio de cultura el rol de regulador difusión de contenidos. Este organismo tendrá la potestad de ordenar el cierre de aquellas páginas web, que según su criterio, estén vulnerando los derechos de la propiedad intelectual. No obstante, aun ejerciendo el rol regulador, las decisiones de este organismo estarán sujetas a la previa autorización de los juzgados centrales de lo Contencioso Administrativo.

A grandes rasgos el texto de este apartado intenta ser el árbitro de una partida desigual entre creadores, autores y distribuidores o difusores.

Funcionamiento de la Ley Sinde:

El responsable de los contenidos que supuestamente vulneran la propiedad intelectual o aquellos que hospeden el contenido o aquel proveedor de acceso al que está conectado un servidor que hospeda los contenidos, inclusive quien los publicó tiene 48 horas de margen para que de forma voluntaria, retire el contenido en cuestión. Este mecanismo se hace por

medio de una petición en la que no intercede ningún organismo con poder jurídico. En caso de no retirar el contenido en el plazo indicado se procede del siguiente modo:

Plazo de 2 días para que se presenten las pruebas del lado de quien demanda y del lado de quien es demandado, cuya postura no es retirar el contenido

Plazo de 5 días para presentar las conclusiones de las pruebas introducidas por las partes.

La comisión de Propiedad Intelectual tiene un plazo de 3 días para dictar resolución.

Dicha resolución es presentada nuevamente a un juez quien es el que la autoriza o no. Este juez no tiene la obligación de participar en el proceso de determinación de culpabilidad o no, simplemente ratifica o niega la conclusión final.

En la primera versión del apartado **Disposición Final Segunda**, a diferencia de la segunda no contemplaba la intervención de un juez para dictar sentencia, permitiendo plenos poderes de ejecución a la comisión cultural sobre el contenido.

### **1.1.3. Contenidos digitales en Internet**

Sin lugar a dudas, la expresión “contenidos digitales en Internet” abarca un campo muy extenso dentro del sector de la industria audiovisual. Desde la compra de un libro en formato PDF, e-book, para nuestros e-readers y la publicación de un artículo de información en un blog hasta la difusión audiovisual a través de Internet. Con la introducción de los procesadores de texto, este sector ha experimentado un crecimiento totalmente exponencial que presenta una serie de problemas para el usuario final que accede a ellos, como son:

- Desbordamiento cognitivo

La información disponible en la red es tanta que resulta difícil escoger aquella que realmente interesa. A, día de hoy el usuario tiene la sensación de sentirse desbordado en la red.

- Accesibilidad

El modo de acceso a toda la información que hay en la red es desigual. Ciertos canales parecen tener prioridad sobre otros, haciendo que el usuario siempre consulte los canales más accesibles o bien confíe más en ellos por su influencia

- Desorganización Web

Los contenidos en las páginas web públicas se encuentran desorganizados, impidiendo una correcta indexación de los mismos por los motores de búsqueda, lo que provoca que muchas veces no encaje una búsqueda con el resultado de la misma. Para solucionar este problema actualmente se está trabajando en sistemas de metadatos, que buscan catalogar de forma eficiente, los contenidos.

La presencia del video en Internet es una realidad incontestable. Las mejoras obtenidas en la compresión del video y la salida de nuevos formatos como por ejemplo MPEG-4, OGV, etc. Así como el creciente ancho de banda en nuestras líneas domésticas, permiten que la inserción del video en internet sea posible. Las mejoras que han surgido en nuestros navegadores Web en los dos últimos años, hacen que las páginas web hayan conseguido obtener una estética que no tiene nada que envidiar al que el mundo de la televisión ofrece.

#### 1.1.4. LSSI

LSSI son las siglas de una ley llamada “**Ley de los Servicios de la Sociedad de la Información**”, a grandes rasgos el objetivo de esta ley es de fenómeno regulador. Exactamente regula cualquier servicio prestado, ya sea a distancia o por medio electrónico y a petición individual del destinatario. En esta ley también se contemplan todos aquellos servicios no remunerados, es decir sin ánimo de lucro, dependiendo de que estos puedan constituirse finalmente como una actividad económica.

Actualmente existe una clasificación de tipos de servicios que son afectados por dicha ley:

- La contratación de bienes o servicios por vía electrónica

- La organización y gestión de subastas por medios electrónicos o de mercados y centros comerciales virtuales

- La gestión de comprar en la red por grupos de personas u organizaciones.
- El envío de comunicaciones comerciales, como por ejemplo folletos o banners vía correo.
- El suministro de información por vía telemática, por ejemplo newsletters.
- El video bajo demanda, como servicio en que el usuario puede seleccionar a través de la red, tanto el programa deseado como el momento de su suministro y recepción y en general, la distribución de contenidos previa petición individual.

Las sanciones interpuestas por cometer alguna infracción respecto al no cumplimiento de la ley LSSI son elevadas y se distribuyen en 3 tipos según su gravedad:

- Sanciones leves que pueden alcanzar hasta los 30000 euros.
- Sanciones graves oscilan entre los 30001 euros y los 150000 euros.
- Sanciones muy graves tienen un coste entre 150001 y 600000 euros.

### **1.1.5. Propiedad Intelectual**

El concepto de Propiedad Intelectual es algo amplio e incluye varios tipos de derechos reconocidos legalmente y que surgen en cierto modo de la creatividad intelectual. Los derechos de la propiedad intelectual son derechos sobre bienes intangibles, desde una idea a una implementación práctica de la misma. Actualmente la propiedad intelectual incluye como mínimo los derechos de autor, marcas registradas, patentes y secreto industrial.

En este proyecto y dentro de este apartado haremos especial énfasis en los derechos de autor. Estos derechos son otorgados a autores de trabajos originales, como libros, películas, programas de ordenador y dan un permiso exclusivo para reproducir el trabajo o contenido, realizar trabajos derivados del original o presentarlos o representarlos de forma pública. Hay que dejar claro que los derechos de autor solo protegen la forma o la expresión de las ideas, no las propias ideas en sí mismas.

El derecho de autor sobre un material existe aunque no esté recogido en el registro legal. Normalmente se registra un derecho de autor con el fin de obtener beneficios sobre el

contenido o material creado. En el caso de que el derecho pertenezca a una persona física, éste durara toda la vida del autor más 70 años, no obstante si el derecho ha sido otorgado a una empresa, éste puede alargarse 95 años. Por lo tanto el derecho de autor convive con el autor y la obra para proteger la integridad del origen de la obra.

## **1.2. Problemática**

En el escenario actual, con una oferta de difusión que no cumple los requisitos legales y con una gran demanda dentro de internet, es prácticamente obligatorio el crear plataformas de difusión, con tecnología streaming, para abastecer a los usuarios de Internet. Con la controversia que se ha generado dentro de la comunidad de actores y productores, es necesario establecer las bases de un modelo de negocio que identifique los problemas actuales de distribución de series y así tener obtener un éxito como conjunto. Está visto que la oferta actual es caduca y en un futuro dejara de ser legal, por lo tanto hay que mantener a los usuarios actuales con un producto sencillo y completo.

### **1.2.1. Disolución de páginas ilegales**

En la actualidad hay páginas web que ofrecen servicios de difusión audiovisual. Concretamente, en España, una de las páginas más conocidas es “seriesyonkis”. Este tipo de páginas publicitan un listado con las series de éxito y proporcionan un lugar donde poder encontrar la serie buscada, sin tener alojados en sus servidores el contenido a difundir, por lo que se refuerzan con servidores de alojamiento gratuito como Megaupload, Rapidshare, donde si que están alojados los videos relacionados con las series, que, normalmente, son colgados por usuarios que se mantienen en el anonimato mediante seudónimos aunque tengan muchos seguidores en Internet.

Con la introducción del texto de la ley Sinde, estas páginas no cumplen con los derechos fundamentales de difusión de contenido explícitamente redactados en la ley, lo que provocara el cierre de este tipo de páginas así como el bloqueo de los servidores de hospedaje anteriormente mencionados.

## **Servicios actuales que no cumplen con la nueva normativa**

Actualmente en Internet disponemos de varios servicios que con la más que probable entrada en vigor de la nueva ley, dejarán de existir. Los servicios más conocidos actualmente son:

- Cinetube.es
- Seriesyonkis.com

Ninguno de estas dos páginas cumple con los derechos de autor ni de difusión de contenidos audiovisuales, **cinetube** es posterior a **seriesyonkis** y siguiendo un patrón parecido intentó abarcar el mundo del cine y de las series dentro de un mismo portal a diferencia de seriesyonkis que era exclusivamente de series. No obstante ambas hacen servir el mismo mecanismo para acceder al contenido, que es mediante el acceso a proveedores de alojamiento como los anteriores mencionados. Estas páginas actualmente tienen millones de visitas en todo el país y explican la repercusión de esta ley en este último año, así como la negativa de los internautas a que ésta se apruebe.

## **1.3. Estudio del Mercado**

### **1.3.1. Perfil del nuevo Usuario Audiovisual**

El perfil del usuario que accede a los medios audiovisuales, música, video, etc., está cambiando, de hecho este perfil está siempre en continuo cambio. Queda muy atrás aquel usuario que alquilaba películas, a través de su tarjeta de socio, en una máquina expendedora de videos VHS o DVD. El usuario actual está cada vez más capacitado y preparado para acceder a la información desde Internet o a través de aplicaciones del ordenador.

### **1.3.2. Demanda actual**

En internet tal y como se ha visto en los últimos años, la demanda de contenidos de video ha continuado creciendo, sobretodo en estos últimos 4 años, donde han surgido empresas diseñadas exclusivamente para este mercado e incluso las grandes corporaciones de internet han preparado productos dirigidos sólo a este mercado. Actualmente el usuario no

busca almacenar películas en su casa, ya son pocos los que coleccionan este tipo de contenidos, sino que quiere ver una serie o una película en el momento que lo solicita y una vez finalizada la reproducción poder acceder a ella tantas veces como lo solicite previo pago. Actualmente se puede obtener todo tipo de contenido audiovisual previo pago y posterior descarga o bien mediante los servicios de alquiler online que ofrecen el mismo contenido sin poder ser descargado.

Según un estudio realizado por **Universal McCann**, el consumo de videos definitivamente está en evolución. El **82%** de los usuarios de internet ven videos en online y al menos un **23%** ha subido alguna vez un video a un portal de descargas. También es importante destacar otro estudio realizado por **Horowitz Associates** en que se da a conocer que aún sigue siendo la televisión el medio preferido y que según una encuesta de la misma compañía, el **70%** de usuarios recurren a internet para ver una serie que no pudieron ver por televisión. Según datos del año 2006, **6** de cada **10** usuarios ven o descargan contenido audiovisual desde la red, al menos una vez por semana. Cabe mencionar que a lo largo del año 2007 el consumo de capítulos enteros en Internet se duplico respecto al año 2006.

### **1.3.3. Oferta Actual**

Actualmente existen muchos servicios de música en línea disponibles que están teniendo un gran éxito, y en los últimos 3 años con la aparición sobretodo de Microsoft Zune como servicio de alquiler de películas, se ha cubierto gran parte del mercado de difusión de películas y música de una forma legal, o bien con portales como CineClick.com o la esperada Vodder ya se ofrece un servicio de alquiler de películas mediante streaming competitivo. No obstante, actualmente no existe en Internet una plataforma dedicada exclusivamente a la difusión de series de éxito, tal y como se comentaba en apartados anteriores, solo existen páginas que no cumplen con la normativa nueva y por ello están condenadas a desaparecer si finalmente la ley Sinde se aprueba finalmente.

Con el panorama actual se abre un abanico de posibilidades de obtener un rendimiento económico sobre este campo y este proyecto intentará ofrecer una solución ingeniosa para resolver este problema.

### 1.3.4. Casos de éxito en Internet

#### Modelos de éxito de retransmisión de audio

El mundo de la música ha sido uno de los precursores de la nueva forma de entretenimiento, con aplicaciones verdaderamente excepcionales que han conseguido suplir las debilidades de este mercado que no hace mucho mas de 7 años parecía condenado con la copia ilegal de CDs y los puestos de venta ilegal. Se han redefinido consiguiendo mantenerse gracias a su buen hacer e ir cogido de la mano de Internet. A continuación se analizaran las dos aplicaciones más relevantes de este mercado.

##### -iTunes

Sin duda alguna la que más impacto ha tenido a nivel musical ha sido la aplicación iTunes. Esta primera aplicación ofrece un gran abanico de pistas de audio a un precio razonable para su descarga, se pueden descargar álbumes completos así como pistas individuales, en la medida de lo posible están muy actualizados e incluso suelen tener estrenos antes que otros medios. Claramente iTunes está destinado a ser un portal de descarga de contenidos previo pago. También gracias a la colaboración entre discográficas y Apple, este portal ofrece contenido extra y una calidad de audio simplemente excepcional. Se cuenta por millones de usuarios los que tienen acceso a contenido a través de iTunes, que, prácticamente, está presente en cada ordenador.

##### -Spotify

Spotify, de origen sueco, ha sido toda una revelación y su impacto y crecimiento en Europa está siendo implacable. Es la nueva estación de radio de Internet y difusión de contenidos de audio y se basa en todo lo contrario que iTunes. Por un precio relativamente económico, se puede escuchar cualquier canción sin interrupciones ni anuncios, no obstante no se pueden descargar las pistas publicadas, pero su adaptación a los sistemas móviles está haciendo que muchos usuarios se decanten por este tipo de servicio. Su servicio en versión gratuita es más que aceptable y tanto la calidad del contenido como la cantidad son suficientes. Con su versión de pago además de ofrecer música sin interrupción, ofrece movilidad, un factor clave hoy por hoy. Con su versión para Smartphone Android e Iphone permite llevar la música a cualquier lado, siendo la contraposición de los dispositivos mp3,

en cierto modo condenado a su desaparición si este sistema finalmente se implanta como norma en cada hogar. No obstante está disponible un servicio de descarga para los usuarios Premium que lo deseen.

Los servicios de Spotify solo están disponibles a nivel europeo, pero contemplan para los próximos años expandirse al resto del mundo. Actualmente dispone de más de 10 millones de usuarios en países como España, Reino Unido, Noruega, Francia,... y aproximadamente el **4%** dispone de una cuenta de acceso Premium, siendo más de 1 millón de personas las que acceden de este modo.

### **Modelos de éxito de retransmisión de video**

Así como en la industria musical se han rediseñado y han acabado sacando servicios que interesa al usuario desde hace ya unos años, en el ámbito audiovisual es más reciente su adaptación a Internet. Empresas como Microsoft, Sony y Apple ya han elaborado estrategias para adquirir este ámbito.

#### **- Microsoft Zune MarketPlace**

Microsoft a mediados del 2007 ya publicó un servicio para sus videoconsolas y reproductores. Se trata del servicio Microsoft Zune MarketPlace, un servicio de alquiler de películas basado en streaming, donde el usuario puede elegir entre 2 tipos de calidad de video, SD o HD y puede decidir además si quiere alquilar o bien descargar la película a distinto coste. Este servicio está disponible en España y parte de Europa desde finales del año 2009, aunque parece que empezó un poco falto de contenido y con cierto desorden, poco a poco ha ido consolidando una interfaz amigable y una colección amplia de títulos. Su elevado coste y su falta de novedades hacen que aun en España no haya sido un caso totalmente de éxito, si bien en Estados Unidos está siendo un verdadero fenómeno. En este último año, han estado trabajando para adaptar su servicio al sistema operativo de Microsoft, así como para la versión móvil Windows Phone 7, esperando ganar más cuota de mercado.

#### **- Sony PlayStation Network**

El caso de Sony viene muy arraigado a su producto estrella de entretenimiento, con la gran aceptación de la videoconsola PlayStation 3. Con, aproximadamente, 54 millones de

usuarios en todo el mundo, decidieron sacar un servicio de alquiler parecido al de Microsoft también de alquiler y compra de películas. Con un precio más asequible que el anterior y con una cuota de usuarios tan grande, el éxito estaba asegurado. Ofrecen exactamente lo mismo que Microsoft Zune, no obstante la estrategia es ampliarse a los televisores Sony Bravia y portátiles Vaio. Según estadísticas realizadas por el propio director sénior de relaciones públicas de Sony Computer Entertainment América, el señor Dave Karraker, en el año 2007 el servicio de Sony registro más de 1,3 millones de usuarios registrados en todo el mundo, de los cuales 600000 son americanos y se han registrado más de 3,7 millones de descargas a través de sus servicio.

### 1.3.5. Mercado de servidores de streaming actual

La tecnología Streaming ha sido determinante para la implementación del audio y video en la web de hoy. Fundamentalmente, esta tecnología se basa en emitir partes de una pista y poder así ser reproducida en el mismo momento sin verse obligado a descargarse todo el archivo para poderlo reproducir. En definitiva, un ordenador (el cliente), se conecta a un servidor y éste le empieza a mandar el fichero. El cliente comienza a recibir el fichero y reserva un *buffer* donde empieza a guardar la información. Cuando se ha llenado el buffer con una pequeña parte del archivo, el cliente lo empieza a reproducir y a la vez continúa con la descarga. El sistema esta sincronizado para que el archivo se pueda ver mientras que el archivo se descarga.



Figura 1.3.5.1: Esquema básico difusión por streaming

En la actualidad detrás del mercado audiovisual existe otro mercado dedicado a la confección y preparación de plataformas de Streaming y video bajo demanda. Han surgido

muchos proyectos Open Source de gran valor, que aportan arquitecturas y tecnologías muy bien diseñadas para implementar de una forma sencilla y eficaz una plataforma robusta de difusión de contenidos audiovisuales. Muchos de estos proyectos han sido adquiridos por empresas que, posteriormente, tras mejorarlo, lo han ofrecido después de pago para las principales cadenas. También son muchas las empresas de alojamiento web (*Hosting*) que ofrecen paquetes con el servicio de streaming ya integrado. A continuación se analizarán los principales productos *Open Source* del mercado de streaming.

## Software actual disponible

Actualmente existen 3 paquetes de software Open Source que son la referencia en este mercado, cada uno de ellos tiene sus ventajas y desventajas. VideoLan, Darwin y Flumotion son los productos a tener en cuenta cuando se quiere crear una arquitectura robusta basada en streaming.

### - VideoLan Streaming Server

VideoLan es un proyecto de software abierto que produce software gratuito de video bajo licencia GNU. Este sistema incluye dos programas:

-VLC Media Player: Este software puede ser usado tanto como cliente como de servidor, permite enviar y recibir contenidos multimedia por la red, simplemente VLC es capaz de transmitir todo lo que sea legible, incluyendo entradas a través de tarjetas capturadoras, cámaras, etc.

-VLS: Este software es exclusivamente un software servidor que permite transmitir archivos codificados en MPEG-1, MPEG-2, MPEG-4, DVDs, canales del televisor y videos en directo tanto en *unicast* como en *multicast*. No obstante este proyecto carece de mantenimiento desde 2005 y se recomienda hacer uso del primer paquete de software que ya incorpora todas estas funcionalidades.

A continuación se muestra un ejemplo de la arquitectura que hace servir VideoLan para la retransmisión de ficheros multimedia:

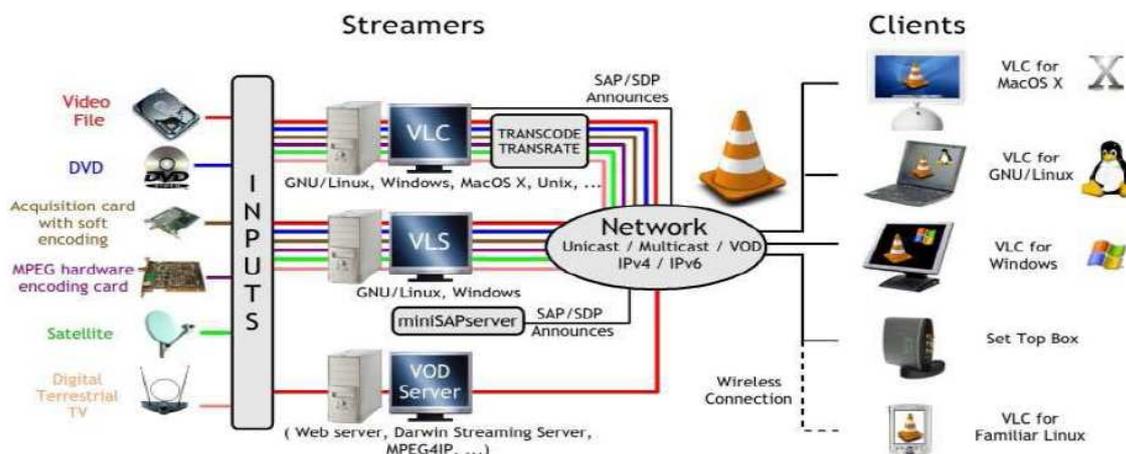


Figura 1.3.5.2: Esquema lógico VLC

#### - Darwin Streaming Server

Darwin es un proyecto también de código abierto pensado para desarrolladores que realizan streaming de contenidos bajo QuickTime y MPEG-4, este proyecto es la versión Open Source del servicio propietario de Apple llamado QTTS, aun siendo el desarrollador principal de esta tecnología, no ofrece ningún tipo de soporte a aquellos que la implementen en su infraestructura, siendo la propia comunidad Open Source quien realiza estas tareas. Darwin al igual que VLC permite la distribución de contenidos multimedia bajo el protocolo RTP y RTSP. Proporciona un alto nivel de personalización y configuración permitiendo al desarrollador modificar el código a su gusto. Tanto Darwin Streaming Server (DSS) como su versión de pago (QTSS), son capaces de proporcionar alto rendimiento a nivel de QoS (calidad de servicio), y proporcionan características como protección ante interrupciones (Skip protección) y arranque instantáneo de la transmisión Instant-On, proporcionando además soporte a los formatos más actuales como son el 3GPP y el MPEG-4.

#### - Flumotion Streaming Server

Flumotion aun siendo la menos conocida también cabe decir que es la más novedosa y actual. Su paquete de software Open Source Flumotion Streaming Server, deriva de uno más antiguo que quedo en desuso llamado Gstreamer. Adquirido por la empresa barcelonesa Flumotion, ha conseguido en los 2 últimos años un papel más que relevante. Con una tecnología muy bien diseñada a nivel de arquitectura de red, con diferentes roles,

como es el Manager y el Worker y una sencilla configuración, se pueden obtener resultados notorios y una experiencia en general satisfactoria. Es un proyecto codificado en su base en *python*, y con un compromiso claro con los formatos de código abierto como son ogg y ogv de Theora Vorbis. Actualmente páginas web como LaSexta.tv y CUATRO.tv se benefician de este software para satisfacer a sus usuarios. Permite emitir bajo cualquier protocolo multimedia incluyendo el protocolo web HTTP. El diseño de red basado en Manager vs Worker permite tener una red tan grande y escalable como se desee, permitiendo que varios servidores bajo el rol de managers arbitren y vigilen el estado de las peticiones que reciben los servidores Workers que administran, pudiendo balancear la carga entre varios Workers según las necesidades y peticiones del momento.

## **Compañías de servicio de difusión**

Existen muchas compañías en España que ofrecen servicios de este tipo, muchas de ellas eran proveedoras de alojamiento anteriormente. Empresas como MultiStream y Flumotion, mencionada anteriormente, destacan sobre el resto, ofreciendo soluciones completas de difusión de contenidos multimedia. La empresa MultiStream actualmente ofrece servicios de streaming para entidades financieras, la liga profesional del fútbol, Vodafone, etc. Esta compañía que ofrece compatibilidades con todos los sistemas móviles así como calidad en alta definición Full HD, ofrece distintos paquetes de servicio, desde retransmisión de eventos en vivo hasta codificación avanzada en H.264.

Flumotion también es una compañía de éxito, con casos de éxito como los anteriormente comentados. Lo más destacable es su compromiso con las tecnologías Open Source y el desarrollo libre de su propio software y también su alto grado de compatibilidad con infinidad de formatos.

### **1.3.6. Oportunidad de negocio**

Visto que no hay una oferta marcada dentro del sector de la retransmisión de series a través de internet, podemos afirmar que este sector necesita propuestas innovadoras y por ello nace este proyecto. Un dato a tener en cuenta es el volumen de usuarios y el número de visitas de las páginas web que actualmente no cumplen con la nueva Ley Sinde. Por ejemplo seriesyonkis.com tiene alrededor de 3 millones de visitas al mes en España. Según el site **paginas911.com** encargada de obtener estadísticas de las páginas más relevantes,

diariamente tiene 6959880 visitas en todo el mundo, siendo los países latinoamericanos junto con España, los países de origen de estos usuarios. Esta página diariamente genera 12527.28 dólares de ganancias. Por lo tanto y visto los números, el proyecto PoNNLo sería viable y se podría generar beneficios de una forma temprana si su coste es flexible y adecuado para sus usuarios potenciales.

## 2. Tecnologías disponibles

### 2.1. Propuesta de solución

La propuesta de solución escogida debe cumplir una serie de prerequisites para que sea funcional y útil. La solución se basa en 4 términos fundamentales:

#### -Flexibilidad

Esta propiedad vendrá condicionada por la arquitectura de streaming que se plantee. El software Open Source Flumotion y su concepto Manager vs Worker permitirá adaptarse a todo tipo de situaciones y ofrecer control en los momentos de más actividad del entorno.

#### -Robustez

La robustez viene muchas veces acompañada de la propiedad anterior, con el concepto Manager vs Worker. La segmentación que ofrece el poder difundir el contenido audiovisual a través de distintos servidores con el rol “Worker” y con la supervisión de otros servidores con el rol “Manager”, permite ofrecer una calidad óptima de retransmisión y a la vez una protección sobre los sistemas.

#### -Independencia

Probablemente la propiedad más importante en Internet si se quiere tener un producto competitivo. La independencia prácticamente en su totalidad viene dada por ofrecer un servicio basado en un aplicativo WEB multiplataforma y multiformato. Esta solución será compatible con todos los navegadores excepto con Internet Explorer 8, será compatible con cualquier sistema operativo y dispositivo, ya sea PC, Tablet o móvil.

#### -Escalabilidad.

Disponer de esta propiedad es sin duda alguna una garantía de futuro para un producto. La escalabilidad es fundamental si los servicios prestados aumentan para poder mantener la eficiencia de la tecnología y del servicio prestado a los usuarios a través del canal.

## 2.2. Arquitectura del sistema

La arquitectura la podemos dividir en 2 secciones, la primera trataría la solución a nivel de red y la segunda trataría la solución a nivel web. A nivel de red y hardware se basará en el concepto que propone el software de streaming Flumotion para realizar la difusión del contenido.

Mediante los 2 roles que ya se han comentado en apartados anteriores (Managers y Workers), se podrá diseñar una arquitectura sencilla y a la vez completa del sistema. Las claves del éxito del servicio propuesto se basan en una correcta identificación de los equipos (Workers) más saturados a nivel de ancho de banda y una buena compensación de la carga. El diseño del escenario a nivel de red dependerá mucho del criterio del diseñador y es necesario que éste sea integro. Otras tecnologías de streaming no ofrecen la segmentación necesaria para crear un servicio profesional y por ello han sido descartadas.

La instalación de agentes SNMP en el sistema será fundamental para determinar dónde están los puntos de criticidad más altos cuando se esté prestando el servicio, también hay que indicar que los enlaces entre los servidores donde se aloje el video a difundir y los frontales web serán unicast, estableciendo varios puntos de conexión al repositorio de video. No obstante la información referente a títulos y contenido disponible será repartida a nivel multicast a todos los portales web para que todos estén actualizados sin necesidad de hacer un uso excesivo del ancho de banda.

La falta de adaptación de Internet a las técnicas multicast impide poder hacer un aprovechamiento más óptimo de la difusión de video bajo demanda y por ello la solución propuesta usa un método un tanto menos óptimo como es unicast. El tamaño de la arquitectura mostrada dependerá del volumen de usuarios que hacen uso del servicio web, a continuación se muestra un esquema lógico general de cómo sería la solución pensada para ofrecer un servicio de calidad basándose en la arquitectura más simple para dejar claro el mecanismo de interacción entre los distintos componentes.

En las figuras adjuntas se muestra El esquema conceptual de comportamiento de PoNNLo (figura 2.2.1) así como su topología (figura 2.2.2).

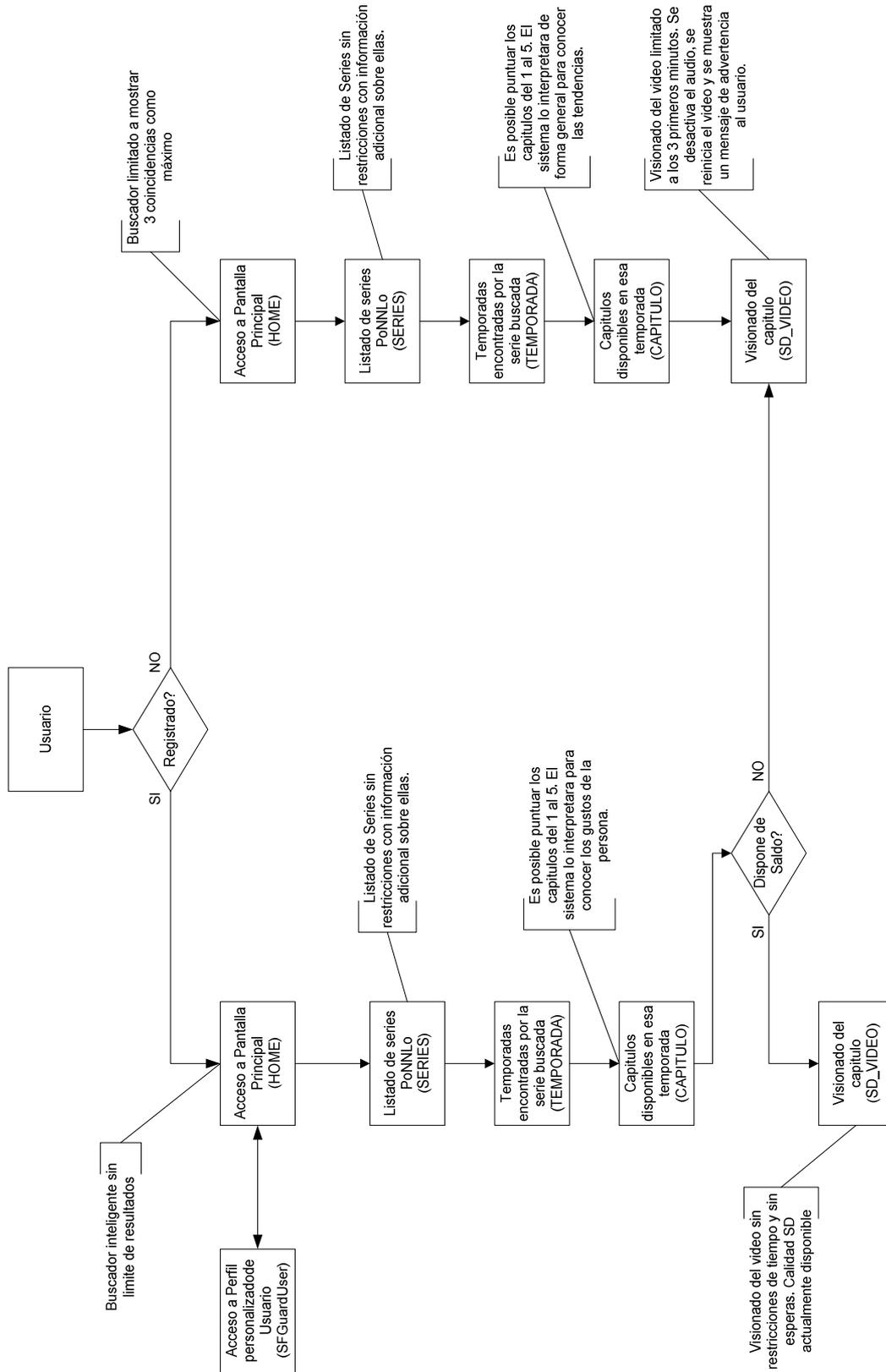


Figura 2.2.1: Diagrama conceptual de comportamiento

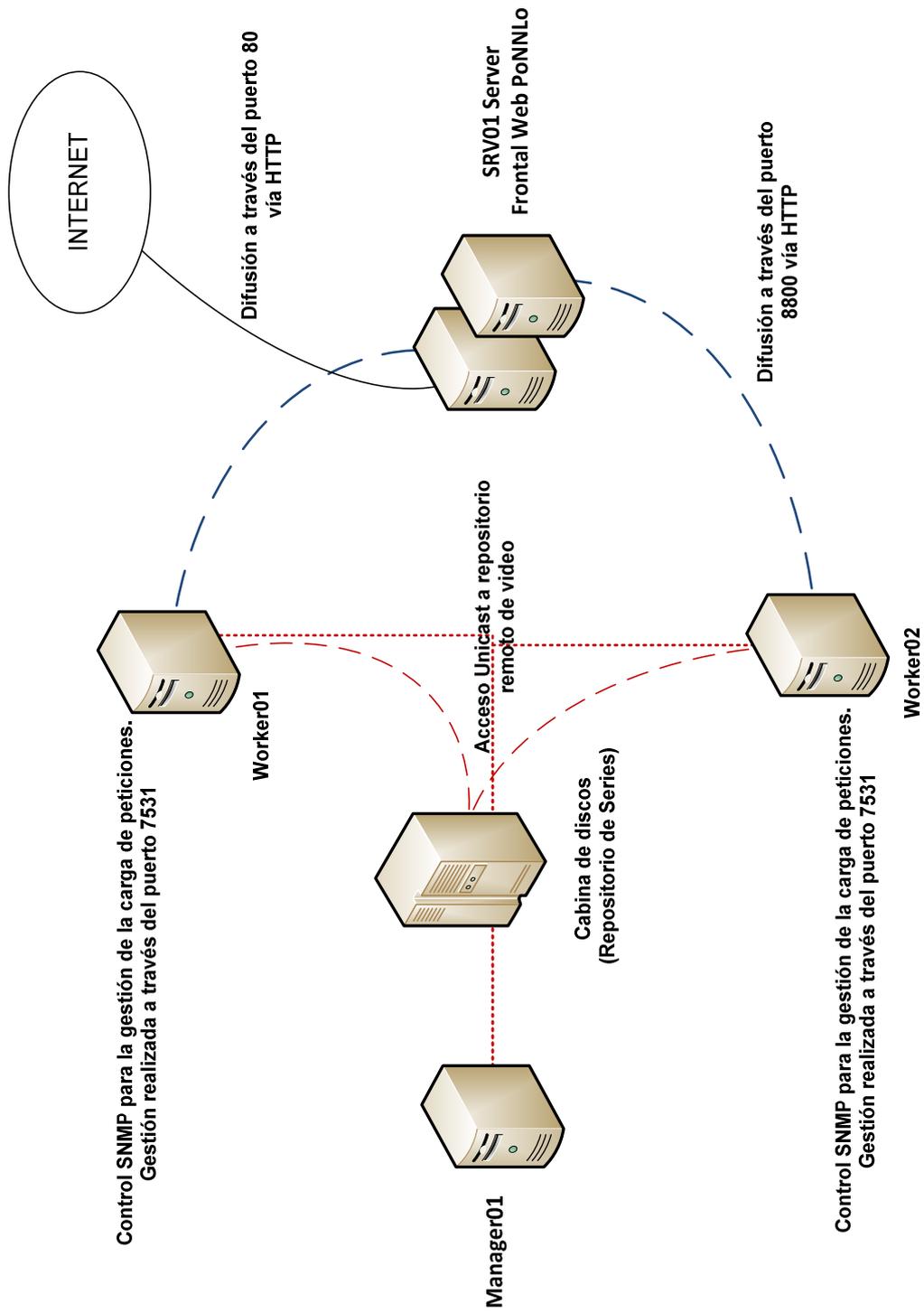


Figura 2.2.2: Arquitectura de video streaming

## 2.3. Herramientas a utilizar

En este proyecto han sido necesarios una serie de herramientas y aplicaciones para poder recrear un escenario de desarrollo y pruebas del estudio. Se ha procurado que todas las herramientas fueran gratuitas para aportar valor añadido al proyecto sin perder de vista la calidad del servicio final pensado. En los siguientes apartados se explicara con detalle todas las herramientas que han jugado un papel clave durante el desarrollo del estudio, también se comentaran detalles históricos de la evolución de cada una de ellas.

### 2.3.1. Elección del lenguaje

Una de las elecciones de este proyecto ha sido el lenguaje de programación con el que se iba a desarrollar. Debido a la cantidad de opciones a escoger como lenguaje de programación, se decidió buscar un lenguaje que cumpliera con los requisitos que la arquitectura del sistema requería. Teniendo en cuenta que se trataba de una aplicación o servicio web, se decidió optar por un lenguaje cliente – servidor como es PHP en su base, junto con JavaScript para especificaciones y funcionalidades concretas que tendrá el proyecto. Como la finalidad era realizar una aplicación web y no una simple página web se decidió aportar valor haciendo servir un framework que en los últimos años ha tenido mucho éxito, llamado Symfony, que posteriormente, se explicara con mucho detalle. A continuación se comentan las características más relevantes de este lenguaje web.

- Es un lenguaje de script interpretado utilizado en el lado del servidor para la generación de páginas Web dinámicas, muy similar al lenguaje ASP promovido por la empresa Microsoft o JSP de Sun MicroSystem. Es un lenguaje embebido sobre el lenguaje web HTML (HyperText Markup Language), pensado para interactuar con el usuario y ofrecer una experiencia avanzada a los usuarios.

Sus principales ventajas son:

Compatibilidad SQL:

Es un lenguaje compatible con gran variedad de bases de datos como MySQL, PostgreSQL, Oracle, Informix y muchas más, haciéndola muy flexible en los proyectos.

## Integración

Integración con varias bibliotecas externas, permite tanto generar documentos en PDF (documentos de Acrobat Reader) cómo analizar código XML basado en etiquetas.

## Soporte

Dispone de una gran comunidad de desarrolladores los cuales ofrecen soporte, gracias a ser un proyecto de código abierto dispone del apoyo de un gran grupo de programadores, permitiendo que los errores puedan resolverse de forma fácil y rápida, evitando así problemas.

## Evolución

Tras el paso de los años, este lenguaje a tenidas numerosas mejoras y extensiones añadidas, ampliando las capacidades del lenguaje, prácticamente se puede hacer cualquier cosa bajo este lenguaje.

## Seguridad

Ya que existen diferentes modos de utilizar PHP, existe también una multitud de opciones de configuración que permiten controlar su funcionamiento. PHP ha sido diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI, Perl o C y con la correcta selección de las opciones de configuración se consigue la combinación exacta de seguridad y rendimiento que se necesita.

Las principales características de PHP son: su rapidez; su facilidad de aprendizaje; su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta.

A continuación se muestra un esquema básico para facilitar el entendimiento del funcionamiento de este lenguaje a nivel cliente servidor, en el dibujo se observa la interacción que se obtiene con este lenguaje cuando estamos accediendo al servicio web o página.



Figura 2.3.1.1: Esquema de funcionamiento PHP

#### -Lenguaje JavaScript

Es un lenguaje que está basado en objetos y en eventos de usuario. Son scripts comúnmente utilizados en sitios web. Su principal característica, es que la interacción con el usuario no requiere contactar con el servidor, todo es interpretado desde el lado del ordenador del usuario, por lo tanto no requiere de una transmisión para interactuar con el usuario. Se parece mucho al lenguaje Java pero no es lo mismo, Java es un lenguaje mucho más complejo y completo.

Este lenguaje fue creado en el año 1995 por uno de los fundadores del navegador NETSCAPE y actualmente JAVASCRIPT es una marca de la empresa Oracle. Sin duda alguna sus principales características son:

#### -Interacción:

No requiere de conexión cliente-servidor para interpretar e interactuar con el usuario, por eso es un lenguaje muy usado para comprobar campos de un formulario por ejemplo.

-Fácil:

Su codificación es muy intuitiva y fácil de aprender para los desarrolladores y es uno de los motivos por el cual se ha extendido tanto en todas las páginas web que hoy día conocemos.

### **2.3.2. Elección de herramientas de trabajo**

En el momento de configurar el entorno de desarrollo, previamente se hizo a nivel local para poder trabajar mejor con el aplicativo de desarrollo. Las herramientas para poder desarrollar la solución son por una parte un servidor XAMPP, la aplicación de desarrollo Netbeans y el entorno de virtualización VmWare Workstation y el sistema operativo Ubuntu 10.10.

El servidor XAMPP es una distribución compacta de una serie de paquetes. No todos los paquetes han sido necesarios para crear el entorno de desarrollo, tan solo se explican aquellos que han intervenido, a continuación hay una explicación detallada de cada uno de ellos así como gráficos que muestran la evolución creciente de los sistemas empleados en este desarrollo.

-APACHE

Se trata del servicio web nativo UNIX de código abierto bajo licencia GNU para poder hacer peticiones por los protocolos HTTP y HTTPS a nuestra dirección IP local. Es uno de los servidores con más aceptación del mundo, concretamente el 70% de los servidores web de Internet hacen servir este sistema según Wikipedia en el año 2005. Ha ido evolucionando desde el 1995 admitiendo mayor compatibilidad de lenguajes como PHP, Python y el reciente Ruby.

Su principales ventajas son que es un sistema modular, de código abierto, multi-plataforma, extensible y fácil de conseguir. Su configuración será detallada en el apartado 3.1.1.

## -MySQL

Es un sistema de gestión de base de datos relacional muy conocido implementado generalmente en proyectos web por su sencillez y su compatibilidad con variedad lenguajes de programación a nivel de conexión. Ha sufrido muchas mejoras desde sus inicios, volviéndose el software de bases de datos multiplataforma más aceptado por la comunidad de desarrolladores. Este software a diferencia de APACHE fue impulsado por la empresa MYSQL AB desde un inicio quien también ofrece soporte tanto como a empresas como a usuarios individuales. A continuación se detalla en la siguiente figura las distintas versiones que ha tenido MYSQL desde el 2000.

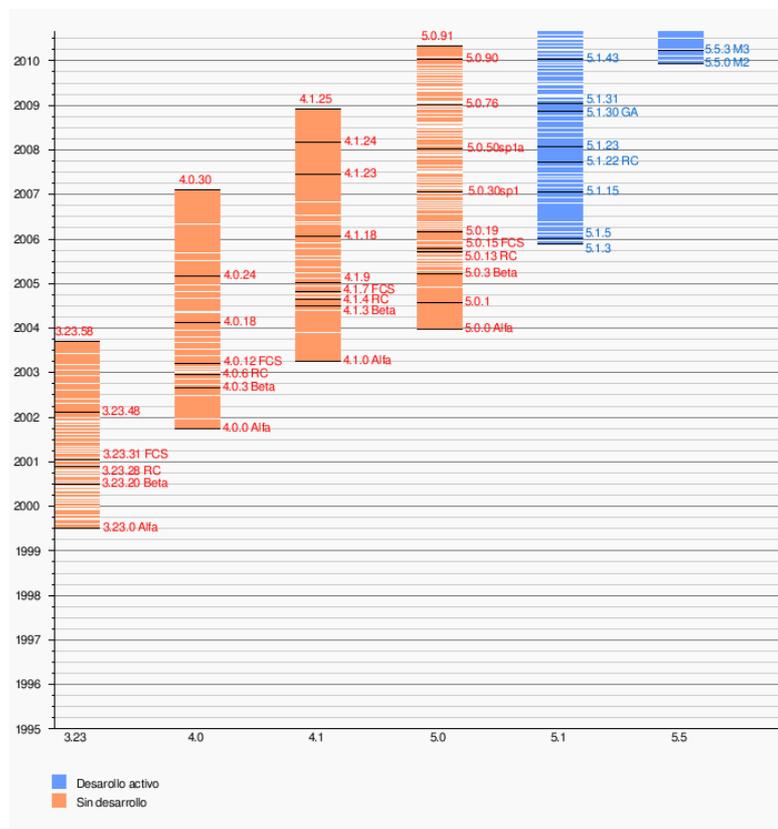


Figura 2.3.2.1: Cronología versiones MySQL

## -Netbeans

Netbeans es la aplicación de desarrollo escogida para realizar este proyecto por muchos motivos. El equipo de desarrolladores de Sun Microsystem creó la aplicación en junio del año 2000 y a día de hoy sigue siendo acogido por la misma empresa. Se trata de una aplicación libre, gratuita y sin ningún tipo de restricción en su uso. Esta aplicación

principalmente fue desarrollada para abordar proyectos basados en el lenguaje Java pero con la adición de módulos y componentes es posible realizar prácticamente cualquier proyecto web. Permite trabajar con distintos frameworks de una forma cómoda.

Concretamente, para este proyecto, se ha hecho servir un pack denominado Visual Web Pack, que contiene soporte para Ruby, PHP y Python. Actualmente se encuentra liberada su última versión, la 7.0.0, aunque este proyecto está hecho íntegramente en su versión anterior, la 6.9.1. A continuación se muestra la cronología de la evolución de versiones que ha tenido esta aplicación publicada en el portal Wikipedia.

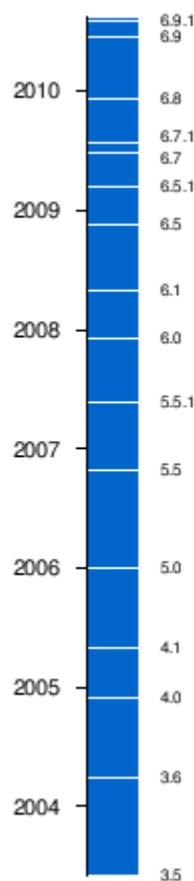


Figura 2.3.2.2: Cronología versiones Netbeans

En el apartado 5.1.3 se explicara con todo detalle la configuración del entorno Netbeans para poder crear la aplicación web.

### 2.3.3. Framework Symfony

Es un entorno de trabajo para el desarrollo ágil de aplicaciones. Automatiza varios de los patrones de diseño más utilizados, añadiendo estructura al código con intención de mejorar la escritura, legibilidad y el mantenimiento del código generado. En definitiva, hace de la programación una tarea más fácil.

A partir de la guía que proporciona Symfony, para aquellos que empiezan a usar este framework, se puede decir que este conjunto de librerías está creado íntegramente en PHP 5 y es totalmente compatible con la mayoría de gestores de bases de datos como MySQL, PostgreSQL, SQL Server de Microsoft, etc. El objetivo de Symfony no es más que el de facilitar al desarrollador más herramientas para poder agilizar el proyecto web y dar con una solución profesional. Las características más destacadas son las que a continuación se detallan:

- Orden:

Los proyectos bajo Symfony se separa en 3 grandes bloques, la lógica de negocio, la lógica de servidor y la presentación de la propia aplicación web, facilitando el entendimiento al desarrollador a la hora de hacer su análisis.

- Automatización

La creación de los métodos más sencillos, set() y get(), de los distintos objetos que relacionan los campos de las tablas a tratar es automática, sin la necesidad de intervención del desarrollador a través de la consola del sistema.

- Independencia

No depende del gestor de base de datos que esté por debajo, tal y como era de prever después de ver la lista de compatibilidades anteriormente descritas.

- Profesional:

Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser tan estable como para crear aplicaciones a largo plazo.

- Extensible:

Su integración con otras librerías le permite al desarrollador extender su aplicación cuando así lo necesite por requisitos específicos.

Este entorno ha sido usado por empresas de primera línea como Yahoo en su página web “yahoo respuestas”, eso muestra el alcance de este framework. El proyecto Symfony fue basado en uno anterior denominado Mojavi que seguía el mismo funcionamiento basado en MVC ( Modelo Vista Controlador), con el uso también desde su creación del ORM (“object-relational mapping”) propel y los helpers ya conocidos por Ruby on Rails para sus plantillas.

## **Modelo Vista Controlador**

El concepto MVC facilita la comprensión y separa las acciones sobre los objetos, la constitución de ellos y la presentación de los mismos en diferentes apartados. Al principio es algo difícil de entender pero una vez que el desarrollador se adapta, se acelera el aprendizaje así como la creación de la aplicación.

Este concepto define la arquitectura de nuestra aplicación, si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original intactos. El controlador tan solo se encarga de aislar al modelo y a la vista del protocolo utilizado contra la aplicación, el modelo por su parte es quien se encarga de hacer la abstracción que realiza la herramienta ORM que más adelante se explica.

A continuación se muestra un esquema que representa el concepto de Modelo-Vista-Controlador de una forma sencilla y fácil de entender.

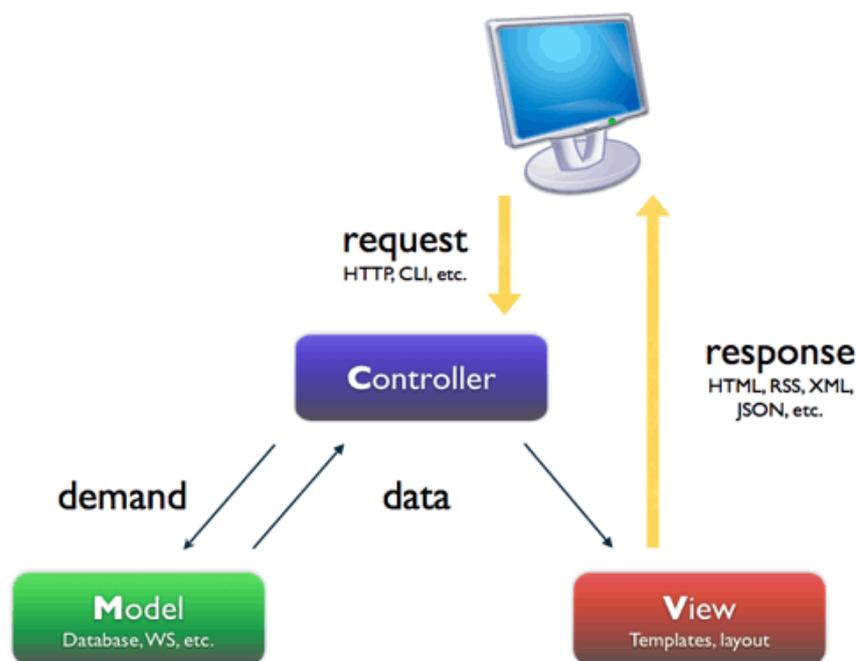


Figura 2.3.3.1: Concepto de modelo-vista-controlador

## ORM

El uso de mapas relacionales de objetos (ORM) es la base del entendimiento de este framework ya que trata los campos de una tabla como objetos independientes con sus métodos asociados. El hecho de usar objetos facilita la creación y acceso a los campos que se hacen servir en la aplicación WEB acercando al desarrollador a una experiencia parecida a la de programar bajo el lenguaje JAVA. En este proyecto se ha hecho servir un ORM distinto a Propel, el ORM utilizado es Doctrine ya que integra más seguridad y parece que será el ORM que en los próximos años tendrá continuidad por el volumen de plugins y componentes que van apareciendo para el mismo. El uso de ORM hace posible la independencia antes comentada sobre el uso de gestores de base de datos, ya que permite una abstracción de los objetos/relacional evitando tener que hacer el uso de una sintaxis específica para poder hacer consultas a los campos de una tabla, así, en el caso de cambiar de gestor, no será necesario modificar gran cosa del proyecto. Para entender mejor el mecanismo de ORM Doctrine, se muestra el siguiente esquema de la figura 2.3.3.2.

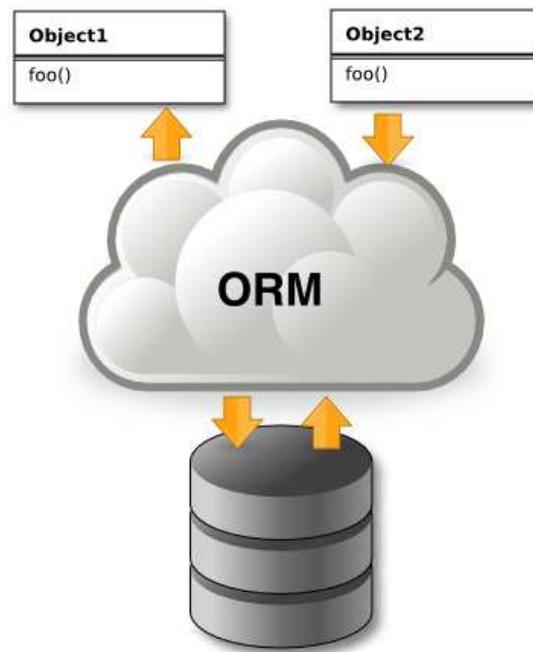


Figura 2.3.3.2: Concepto general ORM

## YAML

Otro de los aspectos más importantes del uso de Symfony es conocer la codificación YAML, sus siglas vienen dadas por (“*YAML Ain’t Markup Language*”). Su estructura es parecida al lenguaje XML basado en etiquetas, pero en este caso la diferenciación de campos se hace a través de tabulaciones y saltos de línea. Aunque al principio parece un poco rudimentario, el hecho de usar este sistema facilita la comprensión al desarrollador cuando lo lee. Fue creado en el año 2001, por lo tanto podemos decir que es bastante reciente este lenguaje, además se pueden crear proyectos no comerciales sobre él. Symfony escogió este lenguaje porque es muy fácil de serializar los datos, fácil de procesar por los ordenadores y fácil de interactuar con lenguajes del tipo script. YAML usa tabulaciones para marcar su estructura, para indicar los elementos que forman parte de una sección se usa el guion y la relación de claves y sus valores se hace mediante los dos puntos (:), los arrays simples con corchetes duros ([]) y los asociativos con corchetes ({}).

En un proyecto YAML sirve para crear las tablas, campos y relaciones entre ellos para poder construir el modelo relacional de tablas. La extensión de los ficheros es “\*.yaml” y normalmente se aloja dentro de la carpeta config/doctrine del proyecto. Este mismo

lenguaje también permite hacer las cargas de datos a las tablas, siguiendo un patrón parecido al del constructor YML de las tablas.

A continuación se muestra un ejemplo sencillo del uso de yaml tanto en la construcción de una tabla como en la carga de datos de la misma.

```
sdVideo:
  tableName: sd_video
  columns:
    id_video:
      type: integer(4)
      primary: true
      notnull: true
      default: '0'
    format:
      type: string(50)
      default: null
    link:
      type: string(200)
      default: null
  options:
    charset: utf8
collate: utf8_spanish_ci
```

Como se puede observar, el código es muy fácil de leer y entender, como se puede ver en las líneas anteriores, a cada campo se le define el tipo, la longitud en caso de ser string como se ve indicado y que valor por defecto o estado inicial comprenderá el campo. También se puede ver el nombre de la tabla a la que hará referencia, en este caso “sd\_video”. Es fácil de ver que esta tabla consta de 3 campos, id\_video, format y link y el tipo de codificación que tienen, en este caso es utf8\_spanish\_c.

### **2.3.4. Información referente a Firefogg**

En este apartado se explica una de las herramientas más innovadoras de este proyecto, la pregunta es, ¿qué es Firefogg?. Firefogg es una extensión o complemento del navegador Mozilla Firefox que permite la codificación de video online desde la web. Esta herramienta hace servir en su base la aplicación ffmpeg ya conocida para codificar video. Para poder hacer uso de esta herramienta, es necesario tener instalado el complemento en nuestro navegador ya que, de lo contrario, no será posible realizar la codificación en la aplicación web creada. Para tener un alcance de la innovación que aporta el uso de firefogg sobre el proyecto, se puede indicar que a través de la página principal del complemento se pueden ver las páginas que actualmente lo han integrado para ofrecer el servicio de transcodificación a través de ellas. Actualmente tan solo hay 3 sites, entre ellas wikipedia.org.

Permite en la actualidad codificar video en formato Theora bajo la extensión .ogg desde una interfaz web adaptando de forma fácil mediante JavaScript su servicio. En el proyecto se ha integrado dentro de un módulo llamado upload donde un rol llamado Agregattor o colaborador podrá tratar sus videos locales antes de subirlos y mediante una herramienta que le proporciona el propio servicio web con el fin de poder estandarizar los videos contenidos en los repositorios de la aplicación.

Se ha integrado en este proyecto este servicio porque aporta valor añadido y cierra el círculo del servicio web pudiendo dar en cierto modo independencia en su mantenimiento y continuidad. En el apartado de explicación del módulo Upload se puede encontrar todo el proceso de adaptación de esta extensión mediante JavaScript para la codificación en el formato .OGV.

## **2.4. Elección del formato de video**

Para escoger el formato de video que se iba a difundir a través del canal se ha mirado las compatibilidades que estos tienen sobre los diferentes navegadores. Atendiendo que este proyecto se basa también en buscar un modelo que no contemple el formato flash para sus contenidos, se ha tenido que realizar previamente un estudio de los formatos actuales disponibles y ver qué grado de alcance tiene para los usuarios.

Estadísticas uso de navegadores año 2010:

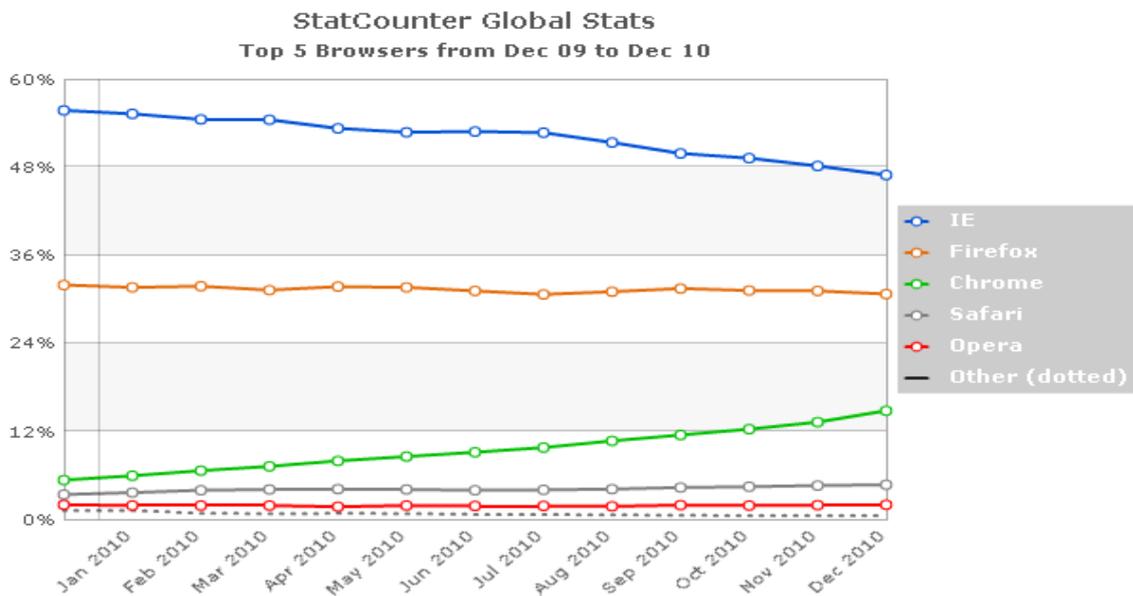


Figura 2.4: Estadísticas Navegadores 2010

A continuación se detallaran los formatos de video que hay disponibles así como sus principales características que puedan afectar a la difusión VoD (video bajo demanda) que ofrecerá el Servicio.

### 2.4.1 Características principales de los formatos de video más utilizados

#### -MPEG-1

Se trata de un conjunto de estándares y formatos de compresión de video y audio diseñados por MPEG (Moving Picture Expert Group). MPEG-1 es el formato común usado en el video CD. Su calidad es muy parecida a la VHS. MPEG-1 viene acompañado por las tres capas de audio que son MP1,MP2 y MP3

Características:

Bitrate: 1.5Mbit/s

Resolución: 352x240 (29.97 imágenes/seg), 352x288 (25 imágenes/seg) .

## -MPEG-2

Es una versión mejorada del formato anterior, es de mayor calidad y una de sus características es que se definen 2 tipos de formato contenedor. El primero es el flujo de transporte y está orientado a aplicaciones que transmiten video y audio por canales poco confiables, es decir, es recomendado para su uso en aplicaciones VoD y similares. El Segundo, llamado flujo de programa está orientado para su reproducción en discos. El códec de video es similar a la versión 1, pero se introduce la posibilidad de codificar con interlineado. Por la parte del códec de audio, se introduce la posibilidad de reproducir en 2 canales ofreciendo la sensación del denominado sonido 5.1.

Es un formato usado sobre todo en DVD y también en las retransmisiones de video digital como la televisión digital terrestre.

Características:

Bitrate: 0.3 Mbits/seg (PDA), 4 Mbits/seg (STB), 15 Mbits/seg (DVD) y 60Mbits/seg (HDV)

Resoluciones: 176x144 (PDA), 352x288 (STB), 720x576 (DVD-PAL) y finalmente 1440x1080 (HDV).

Imágenes/seg: 15-60

## -MPEG-4

Es igual que las versiones anteriores aunque se le añaden un par de extensiones adicionales bastante importantes. Entre las cuales está la gestión de derechos de digitales (DRM), también soporta VRML para permitir el renderizado 3d, este formato soporta un abanico más amplio de códec de video y audio que sus anteriores.

## -H.264

Es técnicamente idéntico al códec MPEG-4, fue creado de forma conjunta por la organización Video Coding Experts Group (VCEG) y la creadora de MPEG. El objetivo de este nuevo formato era el de poder ofrecer mayor calidad a menor bitrate que los anteriores formatos. Este formato es usado en la actualidad en los DVD de alta definición, tanto HD-

DVD como en BluRay y también en los canales HD de televisión. Este formato de video ha tenido muy buena aceptación en este último año.

#### -Theora

Códec de video, desarrollado por el grupo y fundación Xiph.org como parte del proyecto Ogg. La característica que le define y además le hace destacar sobre el resto de formatos, es la posibilidad de poderlo usar de forma libre y totalmente gratuita, sin tener que pagar ningún tipo de cuota de uso a nadie. El códec está basado en el códec VP3 que fue cedido por la propia fundación. Este códec puede ser utilizado en cualquier contenedor, por ejemplo el contenedor MPEG-4. Theora suele usar a la vez el códec de audio Vorbis dentro de un contenedor Ogg. En este proyecto se mencionara muchas veces este formato igual que el anterior ya que serán los puestos en práctica por el servicio.

#### -WebM

Es un formato de video muy nuevo, ha nacido de la mano de la poderosa empresa Google y ha sido creado por la necesidad de adaptar su portal de vídeos más conocido YouTube al nuevo formato web HTML5. Es un proyecto libre, también como el formato anterior, y sujeto al códec VP8 tiene mucha aceptación entre los navegadores como Mozilla, Opera y Chrome, además la Empresa Microsoft ha anunciado que la versión 9 de su navegador Internet Explorer también dispondrá de soporte para este formato. El audio sigue basándose en Vorbis al igual que el anterior contenedor y actualmente este formato de video ya está disponible en el 33% de contenidos publicados por el canal Youtube.

#### -3GPP

Finalmente hacemos hincapié en uno de los formatos diseñados exclusivamente para dispositivos móviles. Se trata de una versión reducida del contenedor MPG-4, está diseñado para reducir al máximo los requisitos del sistema, tanto de almacenamiento como de ancho de banda. Este contenedor fue diseñado por el proyecto 3<sup>a</sup> Generation (Third Generation PartnerShip Project (3GPP)). Este contenedor está diseñado específicamente para aquellos dispositivos que hagan uso de redes 3G como pueden ser smartphones como las nuevas tabletas que han aparecido en el mercado.

Después de analizar uno a uno los diferentes formatos que existen en la actualidad, por criterios de calidad del formato, su buen aprovechamiento del ancho de banda requerido, su origen Open Source así como su compatibilidad con distintos formatos se ha escogido que el repositorio de video del proyecto contemple 3 formatos.

## **Elección de formatos para la plataforma**

Los formatos son WebM por su compatibilidad y buena adaptación a HTML5, el formato Theora por su origen libre y finalmente H.264 por su calidad y su adaptación a sistemas móviles y distribuciones Apple. Escogiendo estos 3 contenedores se puede ofrecer una calidad de servicio y un alcance necesarios para tener una aplicación con presencia en Internet.

Las series difundidas por el servicio dispondrán de sus capítulos en estos 3 formatos, de este modo será reproducible el contenido tanto en los Navegadores Firefox, Chrome, Opera, Safari y Internet Explorer 9. Los 3 formatos son compatibles con HTML5 y por lo tanto no existirá conflicto.

## **2.5. Uso de HTML5 en video**

Antes de empezar explicando con detalle la etiqueta `<video>` en HTML5 y su repercusión en las páginas web, es importante explicar que es HTML5 y que pretende. Como es evidente, se trata de una revisión del lenguaje HTML versión 4, aunque no se trata tan solo de una simple revisión. Esta nueva versión aparte de incorporar muchas más funcionalidades, etiquetas y marcas, intenta ir un poco más allá, con la introducción de nuevas APIs sobre este lenguaje, se intentara crear aplicaciones tan versátiles y potentes como las actuales de escritorio. Las principales características de este lenguaje se pueden resumir en un total de 9.

**Estructura del cuerpo:** La mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada uno de las partes típicas de una página. Por ejemplo para indicar la cabecera o el pie de página, tan solo necesitamos usar las etiquetas `<header></header>` y para el pie de página seria `<footer></footer>`.

**Etiquetas para contenido específico:** Hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones Flash o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.

**Canvas:** es un nuevo componente que permitirá dibujar, por medio de las funciones de un API, en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario. Es algo así como las posibilidades que nos ofrece Flash, pero dentro de la especificación del HTML y sin la necesidad de tener instalado ningún plugin.

**Bases de datos locales:** el navegador permitirá el uso de una base de datos local, con la que se podrá trabajar en una página web por medio del cliente y a través de un API. Es algo así como las Cookies, pero pensadas para almacenar grandes cantidades de información, lo que permitirá la creación de aplicaciones web que funcionen sin necesidad de estar conectados a Internet.

**Web Workers:** son procesos que requieren bastante tiempo de procesamiento por parte del navegador, pero que se podrán realizar en un segundo plano, para que el usuario no tenga que esperar que se terminen para empezar a usar la página. Para ello se dispondrá también de un API para el trabajo con los Web Workers.

**Aplicaciones web Offline:** Existirá otro API para el trabajo con aplicaciones web, que se podrán desarrollar de modo que funcionen también en local y sin estar conectados a Internet.

**Geolocalización:** Las páginas web se podrán localizar geográficamente por medio de un API que permita la Geolocalización.

**Nuevas APIs para interfaz de usuario:** temas tan utilizados como el "drag & drop" (arrastrar y soltar) en las interfaces de usuario de los programas convencionales, serán incorporadas al HTML 5 por medio de un API.

**Fin de las etiquetas de presentación:** todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de CSS.

Una vez comentado las características más importantes ya podemos explicar que es la etiqueta `<video>`. Esta etiqueta está destinada a poner fin a la dependencia al plugin Flash para reproducción de video que en la actualidad existe en este tipo de páginas, aun es una etiqueta que esta por acabar de definirse pero ya está disponible en su mayor parte.

Esta etiqueta proporciona un control prácticamente completo al desarrollador, ofreciéndoles desde controles predefinidos para el video hasta distintos comportamientos sobre el mismo. Aunque HTML5 esta implementado en los navegadores más recientes, algunos de ellos no disponen de todas las funcionalidades que aportan las etiquetas creadas para HTML5.

Su uso es muy sencillo, tan solo hay que especificar el origen de la fuente, el tipo MIME del contenido, es decir, que códec utiliza el formato del video y especificaciones adicionales que proporciona la etiqueta. A continuación se muestra un ejemplo de implementación básica de la etiqueta.

```
<video width="560" height="340" controls>
  <source src="path/to/myvideo.mp4" type='video/mp4; codecs="avc1.42E01E,
mp4a.40.2"'>
  <source src="path/to/myvideo.ogv" type='video/ogg; codecs="theora, vorbis"'>
</video>
```

Como se puede observar en las líneas anteriores la etiqueta video es muy sencilla de implementar y permite definir muchos parámetros, desde el tamaño del marco predefinido donde se visualizara el contenido hasta distintos orígenes del video, permitiendo tener varios formatos dentro de un mismo video, adaptándose de este modo a muchos formatos.

Aquí se listan los atributos fundamentales que aporta esta etiqueta, entre los atributos cabe destacar el llamado preload que será importante durante la creación del proyecto, también se explica brevemente de cada una de ellas a continuación:

**Src:** Enlaza el archivo de video que se quiere reproducir.

**Width:** Define el ancho del video en pixeles.

**Height:** Define la altura del video en pixeles.

**Controls:** Permite implementar los controles del reproductor por defecto del navegador como son los botones play-pause, seek y volumen.

**Autoplay:** Permite reproducir el archivo de video desde que se carga la página.

**Preload:** Carga en el buffer una fracción del archivo de video antes de iniciar la reproducción para que no se pare por reproducir más de lo que carga

**End:** (valor numérico). Define en qué momento el player dejará de reproducir el audio y video. Por defecto se reproduce hasta el final

**Poster:**(dirección URL). La URL de una imagen que se mostrará mientras el video no esté listo.

**Loopstart:** (valor numérico). Define donde debería empezar el loop. Por defecto es el inicio.

**Playcount:** (valor numérico). Define cuantas veces el archivo debería de reproducirse. Por defecto es 1.

En la siguiente figura se muestra el aspecto del esqueleto que conforma esta etiqueta de forma nativa así como los controles de serie que ya incorpora. La etiqueta <video> aporta de inicio una interfaz sencilla pero completa para ya poder disponer de una experiencia de video sin tener que escribir muchas líneas de código para embeber el video dentro de nuestra página web.

Obsérvese la siguiente figura 2.5.1:



Figura 2.5.1: Marco predefinido etiqueta <video>



## 3. Solución aportada

### 3.1. Entorno de Desarrollo

A continuación se explica el proceso de montaje y adaptación del entorno de desarrollo utilizado para reproducir el estudio. Al usar el framework Symfony, éste requiere de ciertas características específicas. En los apartados posteriores se explica la configuración del software y servicios empleados.

#### 3.1.1. Preparación del entorno de desarrollo

Para poder reproducir el estudio es necesario adaptar el servicio web así como la herramienta de desarrollo. En los siguientes apartados se explica con detalle los aspectos más relevantes de la configuración del servicio web Lampp instalado en el servidor así como las especificaciones de Netbeans para poder usarlo como herramienta codificadora de php.

### Instalación de XAMPP

Para instalar el conjunto de servicios que proporciona tan solo hay que ir a [sourceforce.net](http://sourceforce.net) y descargar del repositorio del grupo APACHE2. Desde la página web [www.apachefriends.org](http://www.apachefriends.org) podemos descargar la distribución correcta para Ubuntu Linux.

Una vez descargado podemos pasar a la instalación, elevando los privilegios sobre el sistema con el comando “su” y a continuación extraer el contenido del paquete sobre la carpeta “/opt”.

Comando:

```
tar xvfz xampp-linux-1.7.4.tar.gz -C /opt
```

Ahora ya se podría ejecutar nuestro servidor mediante el comando:

```
/opt/lampp/lampp start
```

## Configuración servidor web

Al usar el framework Symfony es necesario hacer unas modificaciones para que funcione correctamente. Se indica los pasos a realizar en orden:

-Modificación del archivo hosts

Se debe modificar el archivo hosts alojado en la ruta “/etc/hosts” para asociar el nombre de nuestro dominio de pruebas a la IP de nuestra máquina.

Parte del contenido del archivo hosts:

```
192.168.0.196 SRV01-TFC # Added by NetworkManager
127.0.0.1     localhost.localdomainlocalhost
::1         SRV01-TFC localhost6.localdomain6    localhost6
127.0.1.1    CL01-TFC
127.0.0.1    ponnlo.es
```

- Creación del VirtualHost

Para poder personalizar nuestro proyecto es necesario crear un VirtualHost donde alojar y referenciar el contenido de ponnlo.es. También al usar el framework Symfony, este, requiere ciertas características adicionales.

Se debe referenciar creando Alias a las carpetas sf y video. Se crea un Alias a la carpeta “sf” para poder vincular las hojas de estilos nativas del framework que afectan a páginas de error predefinidas, también se procede de igual forma con la carpeta video para poder hacer referencia a él sin tener que hacer servir rutas tan largas.

Contenido del archivo httpd.conf alojado en la ruta /opt/lampp/etc/ :

```
<VirtualHost *>
ServerName ponnlo
DocumentRoot "/opt/lampp/htdocs/PoNNLo/web"
DirectoryIndex index.php
<Directory "/opt/lampp/htdocs/PoNNLo/web">
    AllowOverride All
    Allow from All
</Directory>

Alias /sf "/opt/lampp/htdocs/PoNNLo/web/sf"
<Directory "/opt/lampp/htdocs/PoNNLo/web/sf">
```

```

AllowOverride All
Allow from All
</Directory>
Alias /video "/opt/lampp/htdocs/PoNNLo/web/video"
<Directory "/opt/lampp/htdocs/PoNNLo/web/video">
    AllowOverride All
    Allow from All
</Directory>
</VirtualHost>
    
```

Se indica en negrita los alias que anteriormente se comentaban así como la ruta origen “DocumentRoot” del proyecto. También se indica con el atributo “ServerName el nombre con el que responderá cuando se escriba en el navegador la dirección ponnlo.es”. También es conveniente indicar los formatos y contenedores que se harán servir, en este caso se debe ir al fichero /opt/lampp/etc/httpd.conf y añadir las siguientes líneas:

```

AddType video/ogg .ogg
AddType video/mp4 .mp4
AddType video/webm .webm
    
```

Una vez hechos estos dos pasos y mediante el comando → “/opt/lampp/lampp restart” , se debe ver la página predefinida de Symfony conforme la configuración del proyecto ha sido realizada correctamente. A continuación en la figura 3.1.1.1 se muestra un ejemplo:

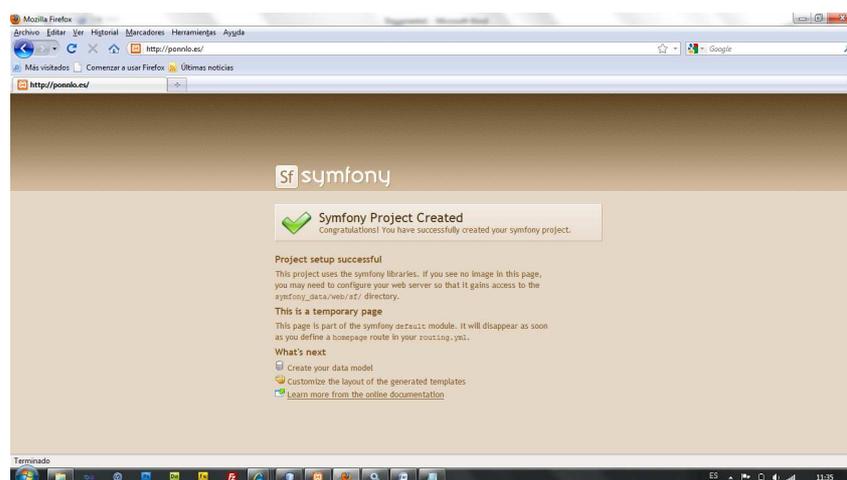


Figura 3.1.1.1: Pantalla principal Symfony

Una vez llegados a esta pantalla ya está listo el servidor web para empezar a desarrollar la aplicación.

## Configuración previa Netbeans 6.9.1

Primero se obtiene el framework que hay que hacer servir. Está disponible a través de la página oficial del proyecto en la dirección <http://www.symfony.es/descargar-symfony/> . En este proyecto se usa la versión estable 1.4, debido a que la versión 2.0 aún no está completamente definida y podrían surgir errores insalvables durante la creación del proyecto. A continuación abrimos netbeans y nos dirigimos a las opciones de la aplicación, dentro del apartado PHP indicamos la ruta donde se aloja el proyecto symfony para poder utilizarlo.

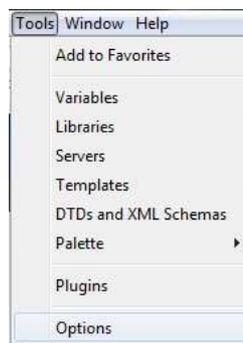


Figura 3.1.1.2: Acceso a Opciones personalizadas Netbeans

En la siguiente captura se muestra la pantalla donde se debe configurar symfony para Netbeans.

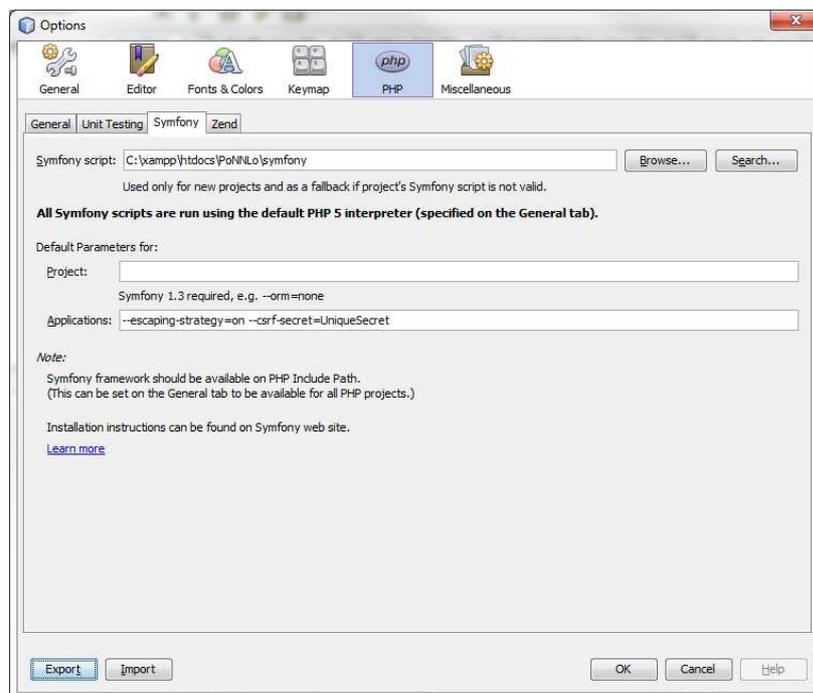


Figura 3.1.1.3: Adaptación de framework Symfony a Nebeans

Ahora tan solo resta configurar el debugger php para poder analizar el código antes de cargarlo a la aplicación web. Para poder hacer uso del debugger tan solo se debe indicar dónde está alojado el ejecutable php.exe del servicio LAMPP.

En la siguiente captura se muestra cómo hacerlo:

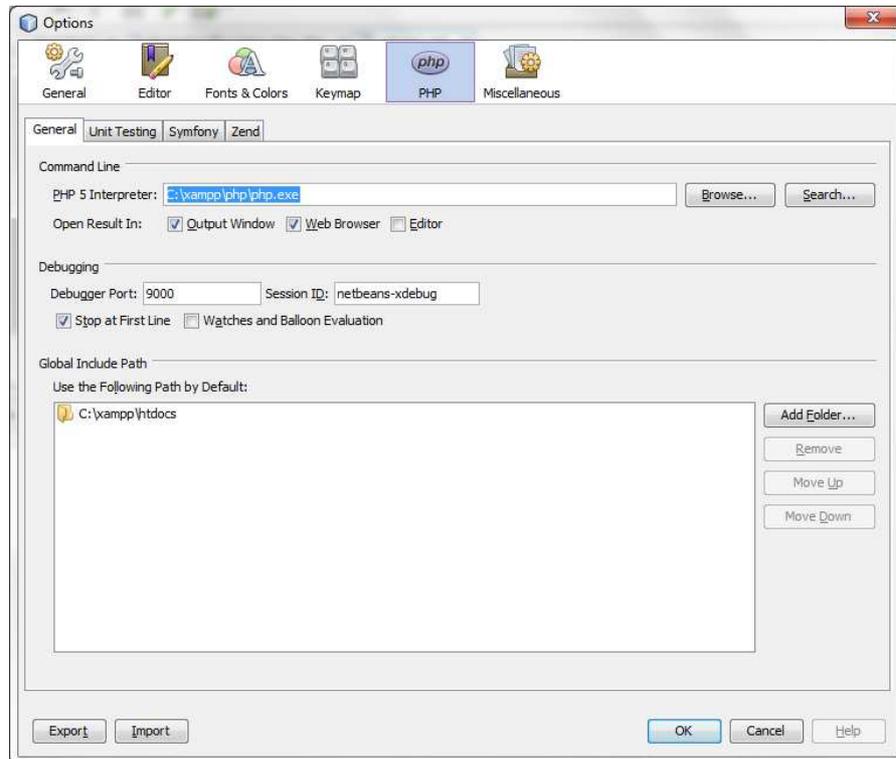


Figura 3.1.1.4: Configuración debugger PHP en Netbeans

## **3.2. Implementación**

### **3.2.1. Definición de roles de Usuario**

Antes de empezar con el diseño y estructura de la aplicación se debe conocer que tipos de usuarios intervendrán en la aplicación y definir los niveles de acceso y usabilidad que tendrán dentro de él. Para este estudio se han definido 3 tipos de usuarios. El administrador, el aggregator o colaborador y el usuario o también dicho cliente que accede.

#### **Usuario**

El usuario o cliente debe tener un acceso limitado, en este proyecto todo el que accede por si solo ya es un usuario independientemente de que este registrado o no, haciendo no obstante una diferenciación entre ellos. El usuario debe poder acceder al contenido de una forma sencilla y poder manipular sus datos personales, acceso a sus estadísticas así como a la reproducción definitiva del contenido.

El usuario que no está registrado a diferencia del registrado, tendrá una visualización limitada del video y el resultado de las búsquedas deberá ser limitado, de este modo el usuario puede decidir si es interesante registrarse más tarde después de probar el potencial del servicio. En cambio el usuario registrado podrá tener acceso total al contenido así como a la totalidad de la reproducción del video. A aspectos generales la usabilidad para ambos debe ser prácticamente la misma, tan solo el registrado tendrá características adicionales. A través del formulario de registro el usuario podrá rápidamente estar registrado por la aplicación.

#### **Agregador o Colaborador**

Este rol está pensado para aquellas compañías o empresas con derechos de autor que quieran mostrar sus contenidos a través de la página. Este perfil dispondrá de herramientas online de codificación de video y creación de registros de capítulos. Se ha diseñado un módulo específico para este rol para facilitar la interoperabilidad de la aplicación.

## Administrador

El administrador como ya se sobreentiende, tendrá acceso total a todas las áreas de la aplicación así como acceso al entorno personalizado de administración. Puede obtener la misma experiencia que un usuario registrado, podrá acceder a la zona para colaboradores así como al entorno de administración interno de la aplicación para poder gestionar los mismos usuarios y el contenido audiovisual del entorno.

Para poder definir estos 3 roles en symfony es especialmente fácil, tan solo se deben especificar durante la creación de los usuarios. Por defecto todos serán usuarios y durante un formulario poder especificar si se es agregador.

Si se hace de un modo manual la creación de los usuarios con más privilegios tan solo hay que crear un archivo con extensión “.yaml” y especificar los tipos de grupos privilegiados disponibles. En este proyecto se hace uso de un plugin llamado **SFGuardPluginDoctrine**, que se encargara de gestionar la creación, eliminación y modificación de los usuarios.

A continuación se muestra como de una forma manual se puede definir un usuario con un privilegio en concreto:

Contenido archivo sfGuard.yaml:

```
sfGuardGroup:

  Group_admin:

    name:      admin

    description:  Administrator group PoNNLo

    Permissions:  [Permission_admin]
```

Se indica en negrita el apartado de personalización del grupo, en este caso se ha creado un usuario con el rol de administrador. Tan solo se muestra la parte que hace referencia a la creación del rol administrador, el resto sería idéntico.

### 3.2.2. Configuración y creación del Modelo

Para poder construir el modelo primero de todo hay que diseñar y crear la base de datos. Para ello symfony tal y como ya se comentó en el capítulo 2, hace uso de una herramienta llamada YAML. Consiste en crear un archivo tabulado en el que existirán las especificaciones de cada tabla, sus campos y sus relaciones entre sí.

#### Adaptación de Modelo MySQL WorkBench a Symfony

Para crear la base de datos se utiliza la herramienta grafica MySQL WorkBench, que dispone de un plugin de Symfony para crear el fichero anteriormente comentado. El plugin se llama Doctrine Export 0.4.1 y está disponible gratuitamente en la página oficial de la aplicación MySQL WorkBench.

No obstante antes de hacer uso de esta herramienta es necesario definir la base de datos. A continuación se muestra el esquema general separado en 2 partes, la primera la gestión de los usuarios que hace el plugin SFGuardPlugin y una segunda donde se muestra la relación entre las tablas que se encargara de gestionar el contenido de video.

Para poder tratar y crear las relaciones de las tablas que actuaran durante la gestión de los videos se han creado de una forma jerárquica. La gestión del video está separada por Serie, Temporada, Capitulo y Video, haciendo una diferenciación entre videos de alta calidad como los de calidad reducida para futuras ampliaciones del proyecto. Para poder establecer los vínculos entre las tablas se ha hecho la siguiente interpretación.

Para una serie puede haber varias temporadas, por cada temporada pueden haber varios capítulos y por cada capítulo puede haber hasta 2 tipos de calidad. Con esta premisa se puede diseñar el esquema relacional de estas tablas y así definir como se compone cada clave primaria. Tal y como se interpreta por lo que se ha explicado, la clave primaria de la tabla Episode (Serie) , estará compuesta de las clave primaria de la propia tablas así como una clave foránea de la tabla Season (Temporada), la tabla Season tendrá como clave primaria el ID de Temporada así como una referencia a la serie que pertenece. Por último la tabla Chapter (Serie) dispondrá de su propia clave primaria y una relación directa con la tabla Season.

A continuación se muestra el esquema relacional de las tablas anteriormente comentadas:

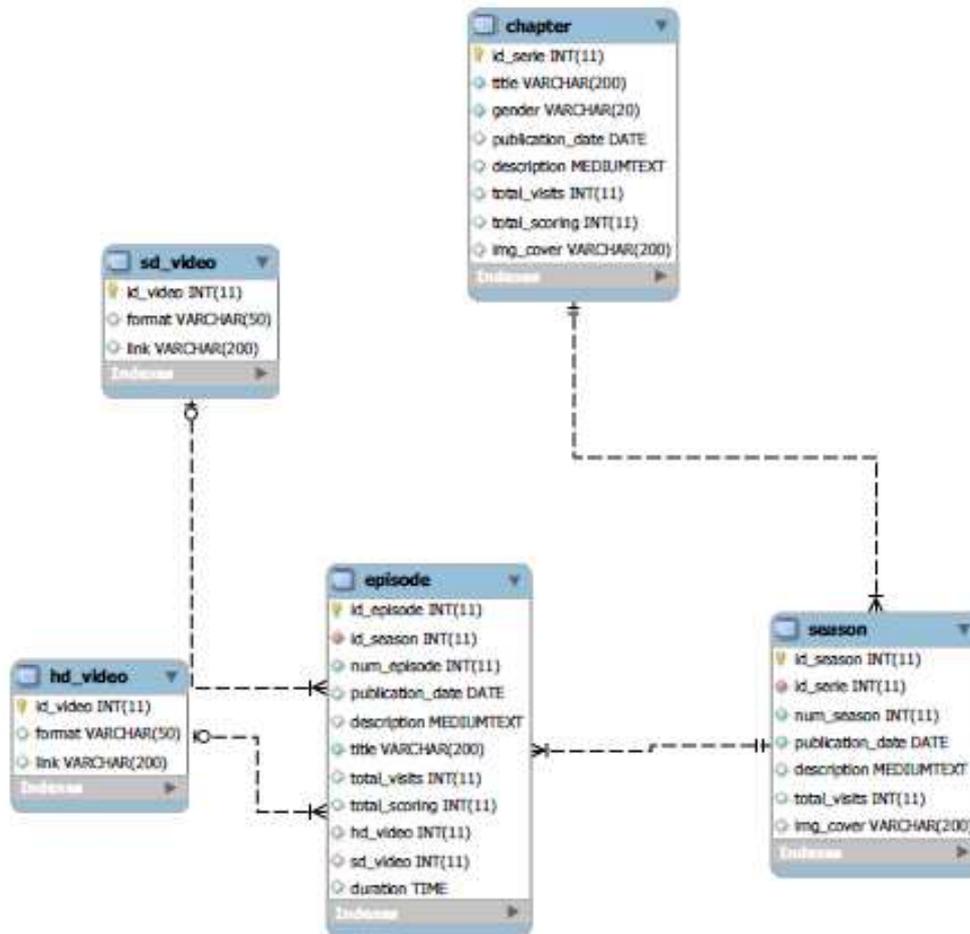


Figura 3.2.2.1: Mapa relacional tratamiento de video

### Creación de schema.yml

Una vez diseñadas las tablas con el plugin de workbench se creará el archivo YAML donde se alojaran las especificaciones de las tablas tal y como lo requiere symfony.

Para ello tan solo hay que seleccionar la pestaña Plugin/Catalog y exportar las tablas. A continuación se muestra como llegar a exportar el diseño con la herramienta:

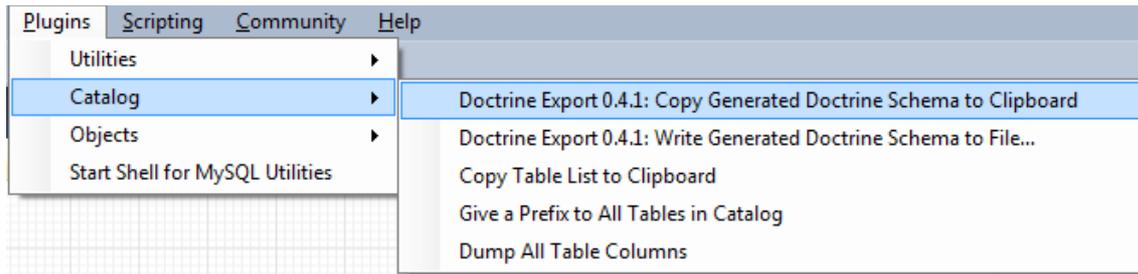


Figura 3.2.2.2: Acceso Plugin Doctrine desde MySQL Workbench

La estructura devuelta permite a symfony recrear el modelo tantas veces como sea necesario. Se muestra una tabla en el archivo generado para que se pueda ver la sintaxis del fichero con extensión .yaml.

```
hdVideo:
  tableName: hd_video
  columns:
    id_video:
      type: integer(4)
      primary: true
      notnull: true
      default: '0'
    format:
      type: string(50)
      default: null
    link:
      type: string(200)
      default: null
  options:
    charset: utf8
    collate: utf8_spanish_ci
```



Tal y como comenta en el capítulo 2 , una de las peculiaridades que le da velocidad a la hora de construir el proyecto, es la herramienta ORM que nos permite crear las clases y métodos sencillos que acceden a las tablas que ya están construidas.

A través del terminal se pueden generar estas clases mediante el comando:

```
php symfony doctrine:build --all --no-confirmation >>ponnlo_classes_schema.log
```

Este comando generara por cada tabla un conjunto de clases que nos permite editar, eliminar y crear registros sobre las tablas de una forma indirecta.

A continuación se muestra una breve salida de este comando para que se pueda ver como genera de forma automática las clases.

Ejemplo construcción del plugin SFGuardPluginDoctrine:

```
>> doctrine generating model classes
>> file+   C:\Users\Eduard\AppData\Local\Temp\doctrine_schema_33455.yml
>> tokens  C:/xampp/htdocs/PoNNLo/lib/mode...sfGuardForgotPassword.class.php
>> tokens  C:/xampp/htdocs/PoNNLo/lib/mode...rdForgotPasswordTable.class.php
>> tokens  C:/xampp/htdocs/PoNNLo/lib/mode...rdPlugin/sfGuardGroup.class.php
```

## Sistema de carga masiva de datos Symfony

Con Symfony y el ORM Doctrine es posible hacer una carga masiva de datos sobre la base de datos para poder empezar el proyecto con información o bien si es necesario volver a construir el proyecto de 0 con unos datos concretos precargados.

Para hacer esto tan solo debemos crear ficheros con extensión .yml alojados en la ruta data/fixtures. El contenido de estos ficheros debe ser concordante con la estructura de nuestra base de datos, en caso contrario el terminal devolverá un error inesperado. En este caso se ha realizado una carga sencilla de series, estableciendo únicamente 4 capítulos que serán los que se visualizaran durante la demostración.

Se crea entonces, un archivo yml para la carga de series, en este caso 4 series, a continuación se muestra la creación de una única serie.

```
chapter:

ESPARTACUS_series:

  id_serie: 3

  title: Espartacus

  gender: Historia

  publication_date: '2007-04-21'

  description: |

    Serie basada en la épica historia de un hombre que sometió a todo

    un imperio siendo un gladiador implacable

  total_visits: 0

  total_scoring: 0

  img_cover: SPART1.png
```

Para crear correctamente el archivo, YAML establece una serie de requisitos. Primero se debe indicar en la primera línea, en que tabla será cargado el contenido, segundo, no se puede repetir el identificador de la serie en otras series sólo por propia consistencia de MySQL, y tercero el contenido de cada campo debe ir después del nombre del campo seguido de ":". En los casos que el campo tenga un contenido largo, como es el caso del campo "description", mediante el carácter pipe "|" se pueden hacer tantas líneas como sea necesario.

No se muestra el contenido de todos los ficheros porque solo se trata de explicar el patrón de construcción que sigue YAML.

Una vez creados todos los archivos, para proceder a la carga se hace desde el terminal escribiendo el comando anterior → **php symfony doctrine:build –all**.

### 3.2.3 Creación y configuración de módulos

Una vez creado el modelo y realizada una carga de datos básica ya se puede empezar a crear las diferentes partes del proyecto. Symfony establece las distintas partes de un proyecto a través de lo que se llama App. No es más que el concepto de separar el proyecto en las distintas aplicaciones que tendrá dentro.

#### Concepto de Modulo App en Symfony

Symfony crea los módulos App a partir del modelo previamente generado, por lo tanto es importante tener un buen concepto del diseño de los datos a trabajar para después facilitar la creación de cada una de las aplicaciones disponibles. Para crear una App acorde a una de nuestras tablas tan solo se debe escribir a través del terminal el siguiente comando:

Ejemplo de creación modulo Series:

```
php symfony doctrine:generate-module --with-show --non-verbose-templates  
frontend series chapter
```

Como se puede observar en el comando, se indica que se quiere crear un módulo con la función generate-module y que se quiere en el frontend, es decir en el frontal visible de la página. El modulo se llamara series y estará directamente relacionado con la tabla Chapter de nuestro modelo.

Una vez invocado el anterior comando, se verá que el terminal muestra la creación de las clases predefinidas para poder acceder a los campos de la tabla Chapter, creará los formularios por defecto de la tabla a la que hace referencia así como también creará validadores específicos según el tipo de campo que sea dentro de cada tabla. Por ejemplo si se quiere introducir un número en un campo del tipo booleano, symfony por defecto detectaría que es un error y que podría provocar un problema, saltando un mensaje de error predefinido por el framework. El hecho de que el propio framework genere validadores por defecto facilita mucho al desarrollador su trabajo.

Ejemplo de salida del comando: (solo se muestran algunas líneas)

```
>> dir+      PoNNLo\apps\frontend\modules\series\actions
>> file+     PoNNLo\apps\fro...eries\actions/actions.class.php
>> dir+      PoNNLo\apps\frontend\modules\series\templates
>> file+     PoNNLo\apps\fro...eries\templates/editSuccess.php
>> file+     PoNNLo\apps\fro...eries\templates/indexSuccess.php
>> file+     PoNNLo\apps\fro...series\templates/newSuccess.php
>> file+     PoNNLo\apps\fro...eries\templates/showSuccess.php
```

Como puede verse, el comando ha creado las páginas de edición, nuevo y mostrar de la tabla así como los métodos relacionados a cada página alojados en el fichero actions del módulo.

Un módulo App está segmentado en 3 partes: las acciones, las plantillas y por último su configuración, que puede ser opcional. Existen pues 3 carpetas dentro de cada módulo. En la carpeta actions se aloja siempre fichero con nombre actions.class.php que contendrá los métodos que harán servir cada una de las plantillas del módulo.

En la carpeta plantillas se alojan las páginas como por ejemplo editSuccess.php que serán la parte visible del módulo.

Para que se tenga un concepto básico de cómo funciona la relación entre la plantilla y la acción, si se quiere ejecutar el contenido de la página indexSuccess.php se debería crear un método en actions.class.php llamado executeIndex para poder llamar a la hoja dentro del módulo, así de sencillo. En el caso de que se quisiera una personalización más en la plantilla tan solo se debería editar entonces las acciones que recaen sobre ella.

En el apartado config se pueden definir aspectos de accesibilidad al módulo o bien si se desea vincular alguna hoja de estilos css de forma predeterminada. Así, dentro de la carpeta config, pueden existir varios ficheros.

A continuación se explican los ficheros que han sido utilizados durante la creación de cada módulo en este proyecto.

## View.yml

En este fichero se contendrá tal y como ya indica el nombre, aspectos de la vista del módulo, como por ejemplo que hojas de estilos intervendrán, el tipo de codificación del propio modulo o bien los metatags o palabras claves por los cuales será indexado por los buscadores para cada apartado.

Ejemplo:

```
default:

  http_metas:

    content-type: text/html

  metas:

    #title:    symfony project

    #description: symfony project

    #keywords:  symfony, project

    #language:  en

    #robots:    index, follow

  indexSuccess:

    stylesheets: [series.css]

  showSuccess:

    stylesheets: [serie.css]
```

## Security.yml

En este fichero se podrá limitar el acceso a la sección de forma específica si no se indica en el apartado security del propio frontend. No obstante si está definido a nivel de frontend y también a nivel de módulo, prevalecerá la configuración impuesta a nivel de módulo.

Ejemplo:

default:

```
is_secure: false
```

En este ejemplo básico se puede ver que el acceso es abierto debido a que la propiedad `is_secure` está en falso.

## Módulos

En esta aplicación se dispone de dos entornos, el frontend y el backend, frontend es la parte pública del proyecto, es decir, donde accederán los usuarios, en cambio el backend es el entorno propio de administración para perfiles avanzados que no son usuarios comunes.

En este apartado se explica las aplicaciones disponibles en cada uno de estos dos entornos.

### - Frontend:

En el Frontend, como es obvio, habrá todas aquellas aplicaciones que el usuario normal podrá tener al alcance, es decir, desde la pantalla principal de búsqueda hasta la información personal de su cuenta como usuario. La estructura que sigue el framework utilizado se divide en Layout y template, el layout es la estructura principal de la página y que perdura durante todas las ventanas de la aplicación. El template es la parte de la página que variara según en qué pantalla estemos. En la parte del template es donde entra en acción los módulos o apps de symfony. Para poder hacerse una idea de cómo distribuye las diferentes secciones este framework se muestra un dibujo a modo de esquema de cómo esta estructura.

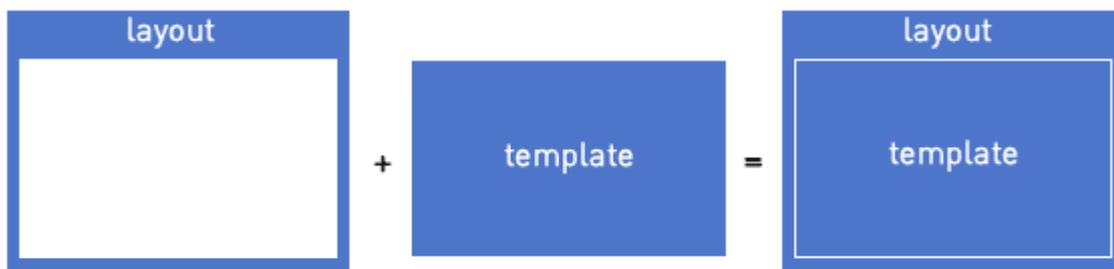


Figura 3.2.3.1: Esquema secciones de la página

A nivel de codificación del frontend son importantes 2 cosas, la primera es el modo en que el framework symfony es capaz de identificar si el usuario ya está validado o no. Para comprobar si está validado tan solo hay que invocar al método **isAuthenticated()**, tan solo se debe comprobar si el objeto `sf_user` está autenticado o no.

Ejemplo de condición de autenticación de usuario:

```
<?php if (!$sf_user->isAuthenticated()): ?>
```

```
<?php endif ?>
```

De este modo se puede manipular el contenido que se verá en el caso de que se esté o no autenticado en la pantalla de inicio. En el caso de estar autenticado en la pantalla principal se mostrará información personal del usuario que ha realizado login tal y como se muestra en la siguiente figura:

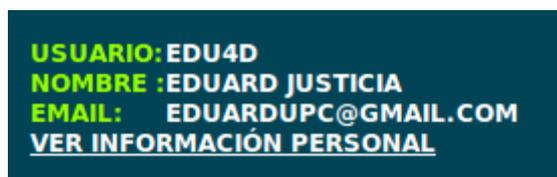


Figura 3.2.3.2: Información personal reducida

El segundo aspecto interesante del layout es la codificación del sistema de búsqueda. Para el sistema de búsqueda, siguiendo la guía de Jobeet de Symfony, se ha utilizado otro framework también muy conocido llamado ZendFramework. Se han creado los métodos de indexación de datos, y gestión de los datos nuevos. Como se trata de bastante código , en este apartado tan solo se explica que los campos a indexar por el buscador, actualmente el

buscador realiza las búsquedas sobre 4 campos de la tabla Chapter, en concreto busca por título, genero, descripción y fecha de publicación.

De este modo se implementa sobre la página principal el buscador. La creación del buscador se explicara más adelante dentro del módulo Series ya que está directamente codificado sobre él.

Código relevante:

```
<form action="<?php echo url_for('@series_search') ?>" method="get">
    <input type="text" name="query" autofocus
    value="<?php echo $sf_request->getParameter('query') ?>"
id="search_keywords" />
    <input type="submit" value="search" />
    
    <div class="help">
        Puedes buscar por (genero, titulo , fecha, ...)
    </div>
</form>
```

Como se puede observar en el código anterior, el formulario de búsqueda invoca al método `series_search` y envía la query a la función registrada en el módulo series para poder ser comparada. Un detalle importante en el código es ver el uso de HTML5 con el atributo `autofocus` que permite que el usuario no tenga que situarse sobre la barra de búsqueda ya que lo hace de forma automática con este atributo.

A continuación se explica el objetivo de cada módulo habilitado así como sus características más relevantes

- Home

El modulo Home será la página por defecto de inicio, será la página con que se iniciara la aplicación, mostrando las novedades y series más vistas según las puntuaciones que realicen los usuarios dentro del entorno, el acceso a usuarios registrados así como una

barra de búsqueda grande por donde se podrá buscar el título de la serie que se busca de una forma intuitiva.

Este módulo es totalmente público y no es necesario estar registrado previamente en él para poder acceder a los recursos del mismo, ya que es la vista principal de la aplicación y sería absurdo si tuviese alguna restricción. Su apariencia y la disposición de los recursos son muy simples facilitando el uso y haciendo sentir al usuario una experiencia amigable, que es el objetivo fundamental de la aplicación.

A continuación se puede ver una imagen del diseño de la sección así como la disposición de los distintos recursos:



Figura 3.2.3.3: Pantalla principal Home

En este módulo es sencillo de codificar, para darle un aspecto más animado y divertido a las caratulas que irán cambiando en el contenido de este apartado, se ha hecho servir jquery, concretamente se ha adaptado el código zoomer.js proporcionado por el repositorio oficial de javascript, de este modo se puede ampliar las imágenes en el momento de pasar el ratón por encima, tan solo se ha tenido que identificar a la tabla que contiene las caratulas con la clase css con nombre “thumb”. La personalización de la animación que proporciona este javascript está disponible en el mismo modulo.

Código para manipular las características de zoomer.js

```
<script type="text/javascript">
$(document).ready(function(){
    $('ul.thumb
li').Zoomer({speedView:200,speedRemove:400,altAnim:true,speedTitle:400,debug:false})
;});</script>
```

- InfoUser

Este módulo sólo es accesible para usuarios registrados, simplemente es una breve vista del formulario ya rellenado por el usuario durante el registro, donde se muestra la información personal del usuario así como el avatar identificativo del usuario. Haciendo uso de los campos incluidos en la tabla `sf_guard_user_profile`, se recogen los datos del usuario y se muestran para que el usuario pueda ver su información personal.

A nivel de codificación es importante conocer el modo en que se recogen estos campos para poder ser mostrados y de este modo ver la sencillez con que symfony trabaja con las propiedades del usuario.

```
<?php
$id_user = $sf_user->getGuardUser()->getID();
$user_personal_info = Doctrine_Core::getTable('userPersonalInfo')-
>find($id_user);
$tarjetaCredito= $sf_user->getProfile()->getCreditCard();
$paypal= $sf_user->getProfile()->getPaypalAccount();
$nombre = $sf_user->getGuardUser()->getFirstName();
$apellido1 = $sf_user->getGuardUser()->getLastName();
$apellido2 = $sf_user->getProfile()->getSurname2();
$email = $sf_user->getProfile()->getEmail();
$zip= $sf_user->getProfile()->getPostalCode();
$provincia = $sf_user->getProfile()->getState();
$pais =$sf_user->getProfile()->getCountry();
$direccion= $sf_user->getProfile()->getAddress();
$avatar = $sf_user->getProfile()->getImgUser();
?>
```

Como se puede observar invocando al método `getProfile()` se pueden recoger todas las propiedades que contiene la tabla `sf_guard_user_profile` y mostrarlas de una forma sencilla. El código mostrado está directamente codificado sobre la plantilla visible del módulo. Para acceder a la información personal tan solo hay que clicar sobre el vínculo “Ver información Personal” disponible en el layout.

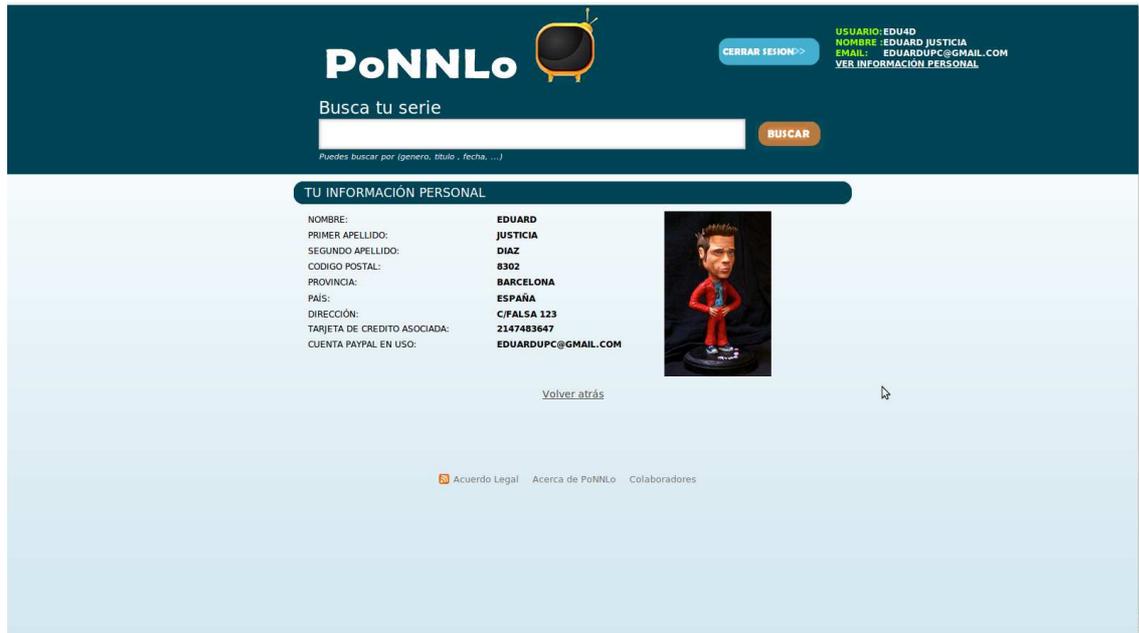


Figura 3.2.3.4: Información Personal Usuario

#### - Series

En este módulo está la representación de las series disponibles dentro de la aplicación. Lo que se visualice dependerá en gran medida de la búsqueda que se realice, no obstante se ha tratado de mostrar siempre un mensaje informativo en el caso de que no se dispusiera del contenido o si se hubiera producido un error durante la búsqueda. La codificación de la búsqueda es bastante extensa y no es necesario comentar todo el código creado.

Haciendo servir el framework Zend tal y como se comentaba anteriormente, se controla los cambios de datos contenidos en las tablas así mismo se indexa los datos para que las búsquedas no se realicen directamente sobre la base de datos, cosa que reduciría su velocidad y rendimiento en general.

Para indexar los datos a través de Zend, hay que abrir una consola de terminal y escribir el siguiente comando:

```
php symfony doctrine:data-load --env=dev
```

La lista que se muestra después de realizar la búsqueda, estará dividida por los 4 campos por los que el buscador actualmente es capaz de buscar, mostrando siempre resultados de series que presenten coincidencias sobre la búsqueda realizada.

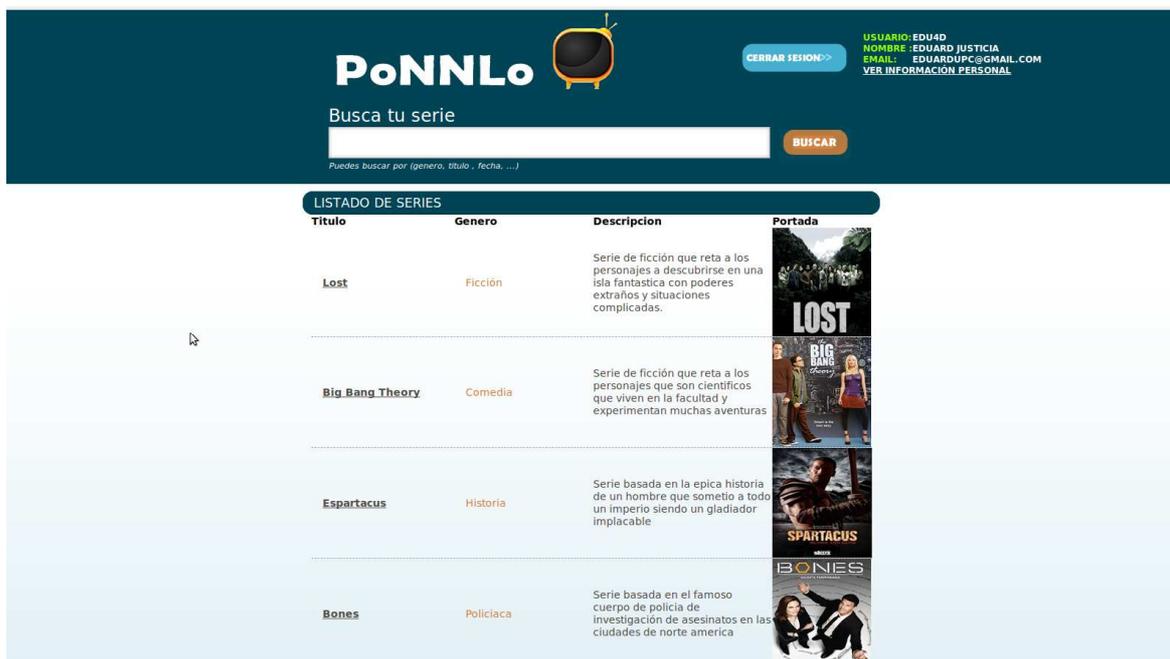


Figura 3.2.3.5: Acceso a Series

Como se puede observar muestra una sencilla vista donde el usuario con un solo clic podrá llegar al contenido de la serie que busca.

- Chapters

En este módulo tan solo se muestran las temporadas disponibles de cada serie, se realiza una query sobre la tabla Season según el ID de la serie en la cual se ha accedido y de este modo tan sencillo se obtiene las temporadas de la misma. A nivel de serie y a nivel de capítulo se añade una funcionalidad más. Se trata del sistema de puntuación, mediante un javascript llamado ratingsys.js que, adaptado al código, da la posibilidad de poder puntuar las series y capítulos para posicionarlas mejor en la pantalla principal.

Ejemplo de adaptación de ratingsys.js a la hoja Chapters:

```
<div id="puntuacion">
  <div name="rateMe" id="rateMe" >
    <a onclick="rateIt(this)" id="_1" onmouseover="rating(this)"
onmouseout="off(this)"></a>
    <a onclick="rateIt(this)" id="_2" onmouseover="rating(this)"
onmouseout="off(this)"></a>
```

```

        <a onclick="rateIt(this)" id="_3" onmouseover="rating(this)"
onmouseout="off(this)"></a>
        <a onclick="rateIt(this)" id="_4" onmouseover="rating(this)"
onmouseout="off(this)"></a>
        <a onclick="rateIt(this)" id="_5" onmouseover="rating(this)"
onmouseout="off(this)"></a>
    </div>
    <span id="rateStatus"></span>
    <span id="ratingSaved"></span>
    <!-- <img name="puntuacion" value="<?php $chapter-
>getTotalScoring()?" src="/images/0Estrellas.png" >-->
</div>

```

Se puede observar que con tan solo pasar el ratón por encima con el evento onmouseover, se aumenta el valor de la puntuación para la serie o capítulo.

Para poder mostrar las temporadas listadas por serie tan solo hay que hacer un bucle dentro de la tabla Season y mostrar las disponibles. Mediante la función count, se controla que en el caso de que no haya temporadas se muestre un mensaje de advertencia al usuario.

Ejemplo de uso de count para devolver siempre un mensaje al usuario:

```

<?php if (count($seasons)==0):?>
    <div id="contenido">
        Parece que aún no tenemos temporadas publicadas sobre esta serie...
    <?php endif ?>

```

De este modo se ha llegado a la segunda pantalla antes de acceder definitivamente al video, tal y como se comentaba en el capítulo 1, ya que el objetivo de esta aplicación web es que sea sencilla y muy accesible por que se cree que será la clave de su éxito.

También mediante CSS se ha querido dar dinamismo al pasar entre pantallas dando efectos divertidos a cada transición.

A continuación se muestra una captura donde se muestra el listado de temporadas:

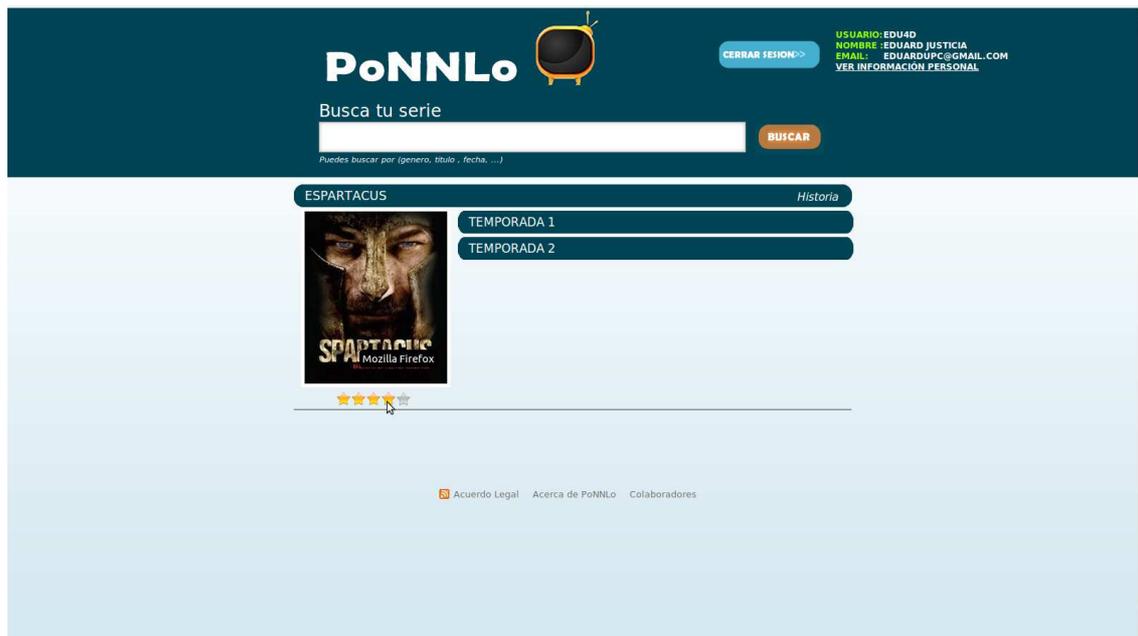


Figura 3.2.3.6: Pantalla de listado de temporadas

Al clicar en una temporada se muestra la lista de capítulos por temporada seleccionada, esto se consigue haciendo del mismo modo se hacía para listar las temporadas ahora se realiza lista los capítulos por cada temporada, a través del identificador de la temporada. En esta pantalla podremos ver los títulos de los capítulos de forma ordenada de menor a mayor. Este módulo es verdaderamente sencillo ya que tan solo consta de un bucle para poder listar los capítulos a visualizar.

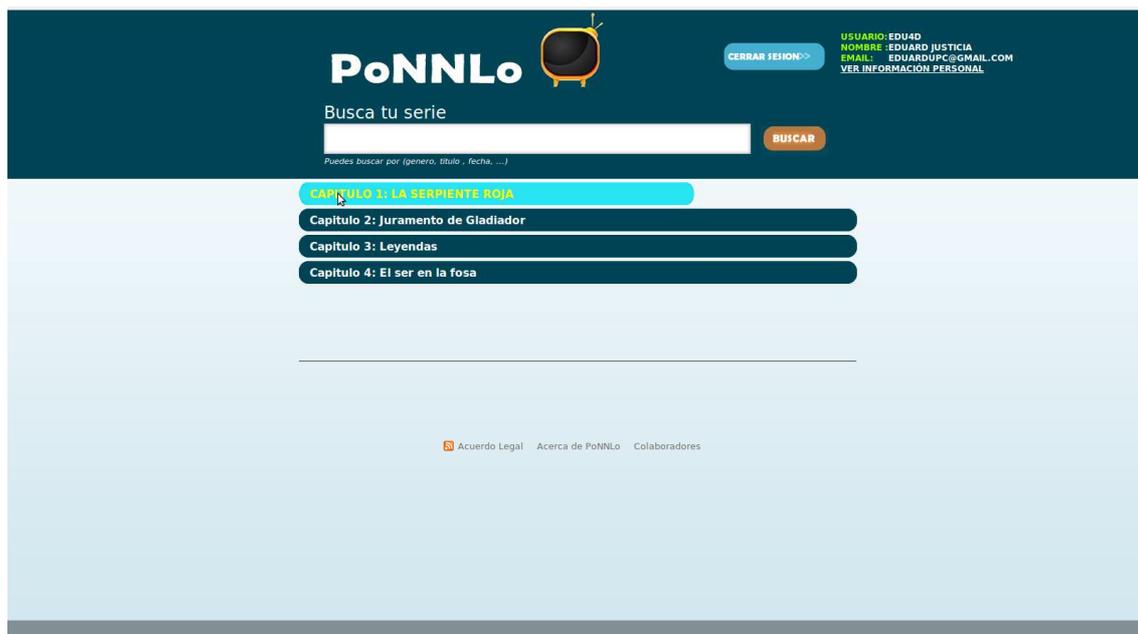


Figura 3.2.3.7: Visualización de listado capítulos

Para listar los capítulos tan solo hay que escribir el código que se muestra a continuación:

```
<?php foreach ($episodes as $episode){?>
<tr>
  <a href="<?php echo url_for('capitulo/view?id_episode='.$episode->getIdEpisode()
?>">
  <h3>Capitulo <?php echo $episode->getNumEpisode(); echo ": "; echo $episode-
>getTitle() ;} ?></h3></a>
</tr>
</div>
</div>
```

Como se puede observar a través del id del capítulo se encuentra el número de capítulo así como el título del mismo que finalmente se muestra.

#### - Video

Este módulo es el último y definitivo del proceso de búsqueda y visionado del video final a visualizar. Aquí es donde verdaderamente interviene HTML5 haciendo uso de la etiqueta <video> que en anteriores capítulos se comenta. Para ofrecer un aspecto más bonito y profesional al marco por defecto que incorpora la etiqueta, se ha hecho servir un javascript y unos esqueletos proporcionados por VideoJS. El origen de estos skins (esqueletos) como del código javascript son de origen OpenSource y no se vulnera ningún derecho de autor al utilizarlos. Lo importante de este apartado es la visualización en si misma del contenido. A continuación se muestra como gracias a la etiqueta <video> se puede hacer uso, de una forma sencilla, de los 3 formatos que serán disponibles en esta versión del aplicativo.

Ejemplo de uso de etiqueta video en el servicio y control de eventos:

```
<script type="text/javascript">
function comprobar(){
var mivideo = document.getElementById('example_video_1');
  if (mivideo.currentTime > '180')
  {
    alert("Has superado el tiempo de visualización, regístrate y obtén las
ventajas de pertenecer a PoNNLo.");
  }
}
```

```
        mivideo.muted=true;
        mivideo.currentTime='0';
    }
}
</script>
<?php if ($sf_user->isAuthenticated()): ?>
<div class="marco_video">
<div align="center">
<div class="video-js-box"
<video controls="controls" preload="auto" id="example_video_1"
ontimeupdate="comprobar()" class="video-js" width="640" height="264" >
    <source src="http://192.168.0.196:8800/<?php echo $sdVideo->getLink()
?>.webm" preload="auto" type='video/webm; codecs="vp8,vorbis"' />
    <source src="http://192.168.0.196:8800/<?php echo $sdVideo->getLink()
?>.mp4" preload="auto" type='video/mp4' />
    <source src="http://192.168.0.196:8800/<?php echo $sdVideo->getLink()
?>.ogv" preload="auto" type='video/ogg; codecs="theora, vorbis"' />
    </video>
</div>
</div>
</div>
<?php endif ?>
```

Es importante destacar en el código anterior, que se hace uso de los 3 contenedores h264, webm y ogv. Actualmente la codificación del video solo está disponible en este último. También hay que destacar el uso del protocolo http por el puerto 8800 porque hasta la fecha la etiqueta <video> no soporta difusión por protocolos especialmente diseñados para el uso, como son rtp y rtsp, no obstante hay que indicar que las principales cadenas hacen servir el mismo medio para difundirlo.

También es importante ver que es necesario especificar el códec hecho servir por cada contenedor. Como se puede observar todos los orígenes tienen marcada la opción **preload="auto"** para que antes de que el usuario se disponga a cargar el contenido este ya se esté cargando en segundo plano.

El hecho de no usar objetos embebidos hace que no se requiera de ningún plugin ni complemento del navegador para ser reproducido como era en el caso de reproducciones bajo flash, esto junto a la compatibilidad de los nuevos navegadores hace que el uso de la etiqueta <video> sea más funcional y no esté limitada al visor donde se está ejecutando.

También hay que recalcar el control de eventos que se realiza sobre la etiqueta <video>, comprobando mediante javascript, el tiempo en curso del video visto y en el caso de superar los 3 minutos y no estar autenticado el usuario, se finaliza el video y se apaga el sonido enseñando un mensaje de advertencia al usuario.

Vista del visor de video:



Figura 3.2.3.8: Acceso a visualización del contenido

Backend:

En el entorno de administración actualmente está disponible únicamente el módulo Upload. El backend tal y como indica la palabra está destinado a usuarios administrador y a los colaboradores, concretamente el modulo que a continuación se explicara esta exclusivamente dedicado a los colaboradores. En futuras ampliaciones se añadirán nuevos módulos para el administrador y la gestión interna de la aplicación web.

- Upload

Este módulo incorpora una de las innovaciones más destacadas del proyecto ya que utiliza el complemento para firefox llamado Firefog. Este es uno de los módulos más importantes porque cierra el círculo de uso de la aplicación dándole cierta autonomía.

Este apartado permitirá al colaborador codificar el video en formato .ogg y subir el video posteriormente, pudiendo tener una lista actual de los videos que actualmente hay en servicio dentro de la aplicación. El colaborador podrá ver si ha habido algún error durante la codificación del video o bien durante la subida del mismo.

Lo más importante es destacar la adaptación del complemento mediante javascript a nuestro servicio.

A continuación se muestra el código javascript hecho servir para codificar el video dentro de la sección upload:

```
<script type="text/javascript">
var ogg = new Firefogg();
function transcode ()
{
if(ogg.selectVideo())
{var options = JSON.stringify({'maxSize': 320, 'videoBitrate': 500});
ogg.encode(options);
document.getElementById('codificar').innerHTML="Estamos adaptando tu video a nuestro
formato, por favor espera...";
var transcodeStatus = function() {if(ogg.state == 'encoding') {setTimeout(transcodeStatus,
500);}
else if (ogg.state == 'encoding done')
{document.getElementById('codificar').innerHTML="Tu video ya es válido para ser
cargado!";
}
}
transcodeStatus();}
return false;}
</script>
```

Como se puede observar en el código, únicamente se transforma a ogg y se controla la codificación online en el caso de que diese un error. El video queda codificado bajo un bitrate igual a 500 y un marco cuadrado de 320, por lo tanto la calidad del video de momento será baja, en futuras ampliaciones se contemplara varios formatos y calidades

para su subida. A continuación se muestra una imagen con el entorno preparado para colaboradores de ponno:

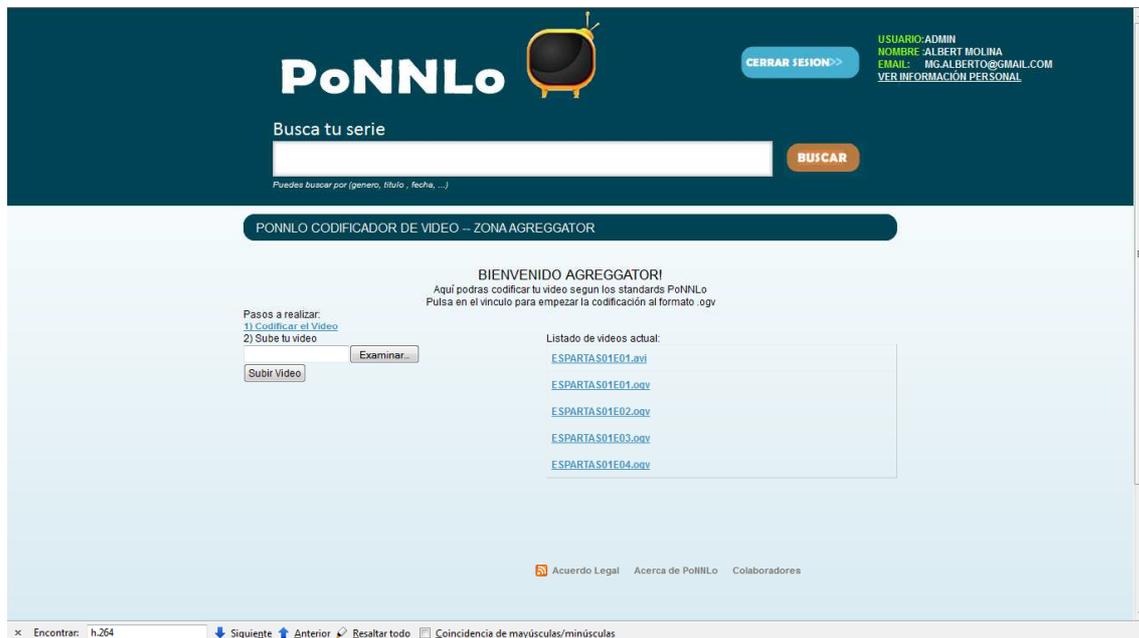


Figura 3.2.3.9: Acceso Entorno colaborador

### 3.2.4. Integrar seguridad en los módulos

Para definir los niveles de seguridad de cada apartado se puede hacer de 2 formas, o bien a nivel del frontend, o bien por cada módulo. Si la aplicación no conlleva muchos módulos que administrar es recomendable integrar la seguridad a nivel de frontend, en cambio sí se dispone de muchos módulos es bueno que cada uno lleve su propia seguridad. En este proyecto se integra a nivel de frontend debido a que no hay muchos módulos. Es muy sencillo de implementar, tan solo se ha de activar o no según el modulo que se desee. Para implementarlo, se debe crear un fichero con nombre security.yml y guardarlo en la ruta apps/frontend/config/ si se implementa la seguridad a nivel de frontend o bien en la ruta /apps/frontend/modules/nombre\_modulo en el caso de implementarse específicamente en cada módulo.

A continuación se muestra un ejemplo básico de cómo se codifica el fichero security.yml:

```
upload:
  is_secure: true
infoUser:
```

```
is_secure: true

default:

is_secure: false
```

Para establecer la seguridad se opta por dejar vía libre a todo el sistema excepto aquellas zonas prohibidas por el usuario como por ejemplo el entorno de colaboradores, de este modo se simplifica más aun la codificación del archivo security.

Es verdaderamente sencillo de implementar y para ver el efecto que tiene tan solo debemos vaciar la cache de symfony mediante el comando:

**php symfony cc**

### 3.2.5. Routing asociado

Antes de mostrar el routing que se ha realizado en este proyecto es importante explicar que es el routing para el framework symfony y para qué sirve. El routing , tal y como se puede sobreentender es la creación de rutas que hara symfony en el navegador cuando se está navegando a través de la aplicación web. Symfony para acceder a un módulo en concreto sigue el patrón de modulo:"nombre\_modulo" action:"accion". Es decir, si por ejemplo hay varias hojas dentro de un módulo se pueden tener rutas diferentes para acceder a ellas tan sólo modificando la parte de "action:". Es importante tener un buen routing porque symfony se vale de ello para generar la navegación entre las hojas y mediante la función `php url_for()`, como por ejemplo `url_for('upload/Write')`, este ejemplo indica que llamara a la hoja `writeSuccess.php` del módulo `upload` del frontend, así de sencillo es. Entonces para cada acción de cada uno de los módulos debe tener un routing asociado, esto hace que el archivo `routing.yml` alojado en la ruta `/apps/frontend/config/` sea grande.

A continuación se muestra un ejemplo reducido de cómo debería ser el routing para 2 módulos diferentes:

```
temporada:

url: /temporada

param: { module: temporada, action: view }
```

```
infoUser:  
  
url: /infoUser  
  
param: { module: infoUser, action: index }
```

En este ejemplo se observa que cuando en el navegador se introduzca la siguiente dirección → <http://ponnlo.es/temporada> , se estará invocando a la hoja viewSuccess.php alojada dentro del módulo temporada y cuando se escriba <http://ponnlo.es/infoUser> se estará invocando a la hoja indexSuccess.php del módulo infoUser. El nombre que se indique como etiqueta del routing no tiene por qué ser idéntico al nombre del módulo para que funcione.

De este modo tan sencillo se obtiene todo el enrutamiento de la aplicación manteniendo un orden y separándolo de las propias hojas php a las que invoca.

### 3.2.6. Aspectos personalizados CSS

En esta aplicación se ha hecho servir en gran medida el css obtenido del proyecto jobeet de symfony, no obstante tal y como se observa en las capturas ha variado en gran medida para darle un aspecto más sencillo y sobrio. Cabría comentar que symfony aloja las hojas de estilo en la ruta /web/css y que se ha generado un css por cada módulo para poderlo personalizar después mucho más rápido. Cuando se quiere añadir una hoja de estilos css a una plantilla php del proyecto, symfony dispone de un método para invocar a las plantillas desde php, se invoca con la siguiente línea de código:

```
<?php use_stylesheet('season.css') ?>
```

De este modo se puede cargar por ejemplo el estilo asociado a la sección temporada a cualquiera de las páginas que se tengan en el proyecto, ofreciendo versatilidad en el uso de las hojas de estilo.

Uno de los aspectos más bonitos de ver es la modificación de las barras horizontales donde se muestran los títulos de los capítulos al pasar el mouse por encima. Para conseguir ese efecto tan solo se ha jugado con 2 imágenes con distinto color y tamaño para dar el efecto y también convirtiendo las letras en mayúsculas para resaltar el título. A continuación se muestra el código css escrito para conseguir el efecto.

```
#episode h3
{
margin-top: 5px;
margin-bottom: 1px;
padding: 0px 15px;
font-size: 110%;
background: url(../images/bg-listheader.gif) no-repeat;
line-height: 32px;
font-weight: bold;
color: #fff;
}
#episode h3:hover
{
width: 555px;
text-transform: uppercase;
background: url(../images/bg-listheaderh3_select.gif) no-repeat;
line-height: 32px;

color: #FFFF00;}
```

Como se puede ver tan solo se escoge la imagen y mediante la propiedad text-transform se convierten las letras a mayúsculas. El código que se muestra está diseñado específicamente para el modulo capitulo.



## 4. Instalación

Debido a no tener un sistema hardware robusto para realizar las pruebas, se ha constituido un servidor virtual con los dos servicios comentados, por lo tanto, dispondrá del servidor web XAMPP junto con el servidor de Streaming Flumotion. A continuación se explica los pasos de instalación del servidor de video bajo la distribución Ubuntu.

Para poder instalar el servidor tan solo hay que acceder al repositorio de synaptics y en el buscador introducir flumotion. Mostrará una pantalla como la que se muestra a continuación:

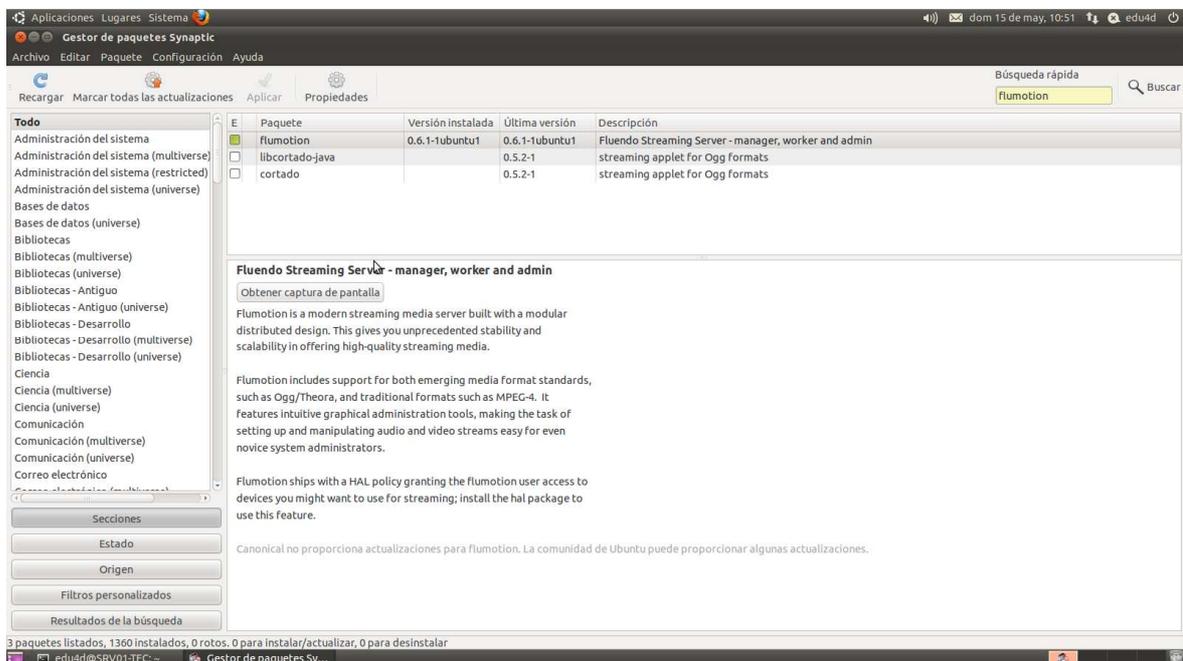


Figura 4.1: Instalación Flumotion Synaptics

Se selecciona instalar el producto y una vez instalado aparecerá la primera pantalla de configuración. La instalación de este producto es realmente sencilla y en unos pocos pasos se puede tener el servidor en funcionamiento. Ahora se va explicar paso a paso como montar un servidor bajo Flumotion, en este caso, se va a preparar una versión reducida del servicio activando únicamente el rol “Worker” comentado en anteriores apartados del proyecto

## 4.1. Configuración del servicio Flumotion

En la sección aplicaciones/Sonido y Video de la máquina Ubuntu se selecciona la aplicación Flumotion Streaming Server Administration.



Figura 4.1.1: Arranque servicio Flumotion

Aparecerá la primera pantalla de configuración del servicio de flumotion, donde se preguntara si se quiere conectar a un servidor con el rol de “Manager” para así unirse a otros servidores de streaming con el rol de “Worker” y ser supervisados por el Manager o bien si se quiere ser el mismo Manager. Se han seleccionado ambos roles en la misma máquina por falta de recursos. En este caso se escoge la última opción, es decir, “Arrancar un nuevo “manager y conectarse a él”.

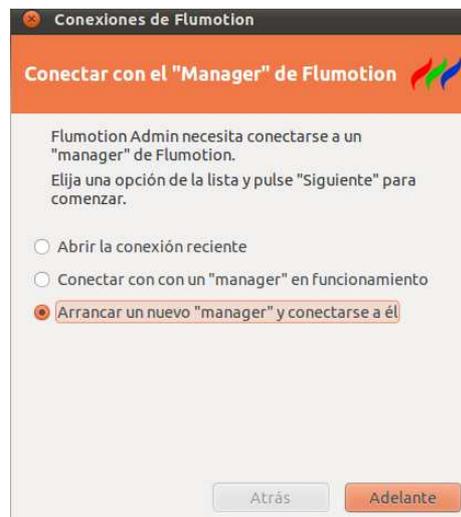


Figura 4.1.2: Configuración como Manager Flumotion

Al pulsar sobre el botón “Adelante”, el programa indicara que en estos momentos se están iniciando ambos roles bajo la cuenta del usuario que ejecuta el sistema, en este caso root. Una vez aceptado que ambos roles se instalaran bajo esa cuenta, aparecerá una nueva

pantalla donde flumotion explica dónde están los ficheros de configuración por si se desea hacer una configuración más avanzada o por si desea también revisar los logs que registra el servicio.

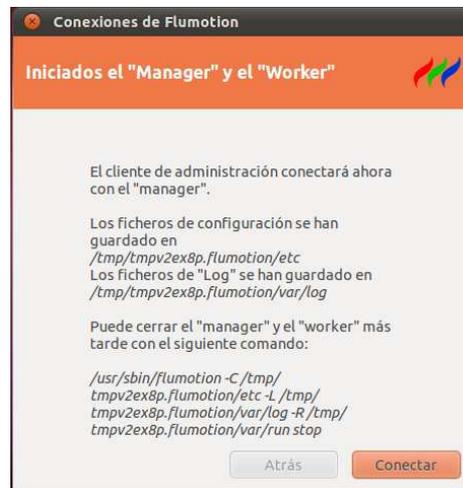


Figura 4.1.3: Archivos de configuración

Una vez realizados estos sencillos pasos ya aparecerá automáticamente el tipo de servidor de streaming que se quiere configurar, en este caso hay que escoger un servicio de video bajo de manda y no un servicio en directo (live).



Figura 4.1.4: Archivos de configuración

Éste probablemente es el paso más importante de la configuración del sistema Flumotion, en la siguiente imagen debemos configurar el repositorio de videos del servidor para que

los pueda difundir, en este caso será “/opt/lampp/htdocs/PoNNLo/web/video”, así como el puerto de escucha del servicio y también el punto de anclaje, que en este caso se ha dejado la raíz del sistema “/”.

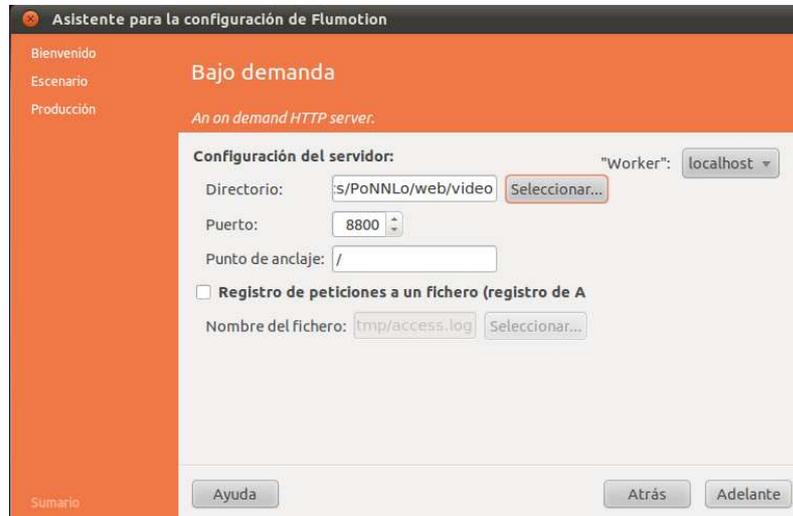


Figura 4.1.5: Archivos de configuración

Y finalmente se mostrara la pantalla final conforme el servicio ha sido bien configurado y ya se encuentra activo para aceptar peticiones.



Figura 4.1.6: Configuración satisfactoria de Flumotion

## Configuración avanzada Manager & Worker

En el caso de querer vincular un servidor Worker con un servidor Manager externo, desde la pantalla de configuración, hay que ir a la pestaña “Conexión” para poder realizar la conexión contra el Manager a través del puerto 7531, indicando el nombre del servidor donde se encuentra ubicado el rol de Manager.



Figura 4.1.7: Conexión Worker & Manager

Una vez realizados todos estos pasos ya se tiene el servidor listo para aceptar peticiones a través del puerto 8800 donde escucha el rol Worker, desde donde servirá todo el contenido multimedia ubicado en un repositorio local.

El repositorio de video no tiene por qué estar en el mismo servidor de streaming, es incluso conveniente por seguridad y rendimiento que esté ubicado en una cabina de discos externa ajena al servicio publicado.

## 4.2. Uso del servicio

Una vez ya está instalado el servicio de streaming que proporciona Flumotion, podemos empezar a emitir en 2 modos, bajo demanda o en directo. Para este proyecto se ha escogido la segunda opción.

En este apartado se explica de forma detallada el uso de la interface de Administrador que proporciona el servicio después de ser configurado.

## Menú Superior de Opciones

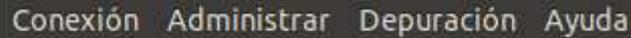
A horizontal menu bar with a dark background and light text. The items are 'Conexión', 'Administrar', 'Depuración', and 'Ayuda'.

Figura 4.2.1: Menú de Opciones Superior

En la parte superior izquierda de la interface se encuentra los apartados que se refleja en la figura 4.2.1, donde se puede desde establecer nuevas conexiones, administrar la actual conexión establecida durante la configuración, depurar la conexión en busca de posibles errores y una breve guía de ayuda.

### Conexión

En el apartado conexión se puede configurar nuevas conexiones, importar configuraciones de otros escenarios o exportarlos para poder montar nuevos servidores “Workers” con la misma configuración que el actual, es importante destacar que Flumotion guarda y recoge la configuración en formato XML, no acepta otros formatos.

También permite establecer varias conexiones a la vez, permitiendo que un mismo Worker este emitiendo a través de distintos puertos varios flujos de streaming. También permite conectar un mismo Worker a varios servidores Manager para que este sea administrado por varios árbitros, esto permitiría tener una estructura más compleja y balanceada. En la siguiente figura se puede observar las funcionalidades descritas.

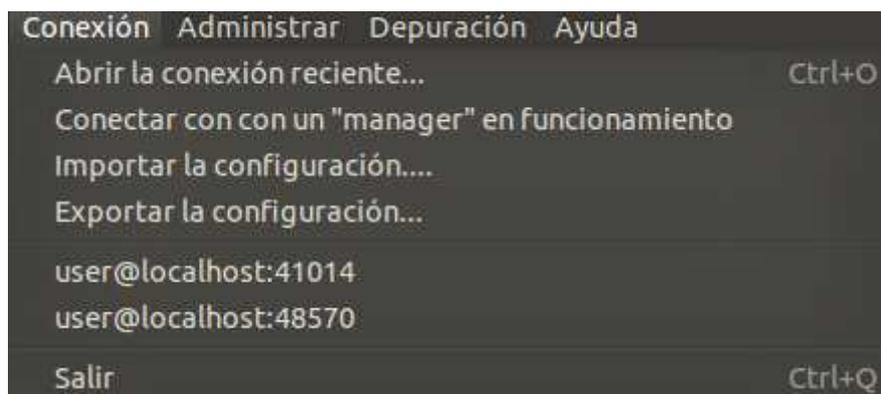


Figura 4.2.2: Menú Conexión

### Administrar

En esta opción se puede administrar la conexión actual, permitiendo poder parar los componentes que actualmente están en uso en la conexión o bien ejecutar un nuevo asistente por si es necesario configurar un nuevo flujo de streaming distinto, por ejemplo una emisión en vivo (live). En el caso de tener el servicio detenido, se puede añadir nuevos formatos de codificación de video.

Véase en la siguiente figura las distintas opciones indicadas:

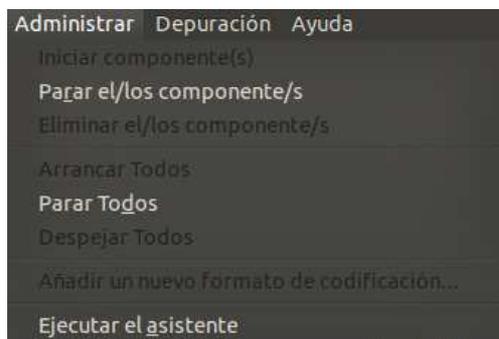


Figura 4.2.3: Menú Administrar

### Depuración

Este modo está dedicado exclusivamente a obtener un análisis del stream actual en funcionamiento. En el momento de activar el modo “Debugging”, aparecerá una nueva pestaña llamada Propiedades donde se puede ver con detalle el punto de montaje, el puerto y el path por defecto del flujo activo. Lo más destacado de este apartado es el punto es la parte de revisión de logs. En el momento de activar el modo debug, el servicio empieza a escribir varios logs dentro de la ruta “/var/log/flumotion/”. Únicamente el administrador tiene privilegios para acceder a la información que proporciona el modo debug. A continuación se muestra una breve parte de los ficheros de log escrito por el servicio de flumotion:

### Service.log

```

root@SRV01-TFC:/var/log# cat flumotion/service.log | more
INFO [ 3448]          servicer      Jun 06 18:57:21   Stopping man
ager default (flumotion/service/service.py:517)
INFO [ 3448]          servicer      Jun 06 18:57:21   Stopped mana
ger default with pid 1389 (flumotion/service/service.py:531)
    
```

En el fichero log anterior se puede obtener información relativa al funcionamiento del servicio, como por ejemplo paradas y puestas en marcha del mismo, PID asignado, ficheros python (.py) asignados al servicio, etc.

### Worker.default.log.x

```

root@SRV01-TFC:/var/log# cat flumotion/worker.default.log.1 |more
DEBUG [ 1435]                log                May 08 16:03:32    opened log
'/var/log/flumotion/worker.default.log' (flumotion/common/process.py:92
)
DEBUG [ 1435]                log                May 08 16:03:32    installing SIGHUP
handler (flumotion/common/process.py:92)
DEBUG [ 1439]                worker           May 08 16:03:33    Started daemon
(flumotion/common/process.py:176)
DEBUG [ 1439]                worker           May 08 16:03:33    written pid file
/var/run/flumotion/worker.default.pid (flumotion/common/process.p
y:180)

```

Este fichero log, donde la X final del título del fichero corresponde al stream en curso, se pueden ver aspectos más detallados del propio flujo que se está emitiendo. Se detallan más los ficheros python que están corriendo en ese momento, así como información de la versión del Worker, puertos abiertos y posibles incidencias durante el servicio.

### Ayuda

Dentro del menú “Ayuda” se puede obtener la versión de Flumotion instalada así como los desarrolladores que han intervenido, también se puede obtener una guía sobre el funcionamiento de los distintos componentes que proporciona el servicio.

### Acceso a opciones estándar rápido

Para poder hacer una administración básica, se puede realizar desde el conjunto de botones reducido que dispone el servicio en la parte superior izquierda.



Figura 4.2.4: Botones de administración básica

Desde los botones mostrados en la figura anterior, podemos prácticamente gestionar el servicio en un nivel básico. El primer botón  , permite conectar a un flujo ya

configurado con anterioridad, al pulsar sobre él, se muestra la ventana mostrada en la figura 4.2.5.

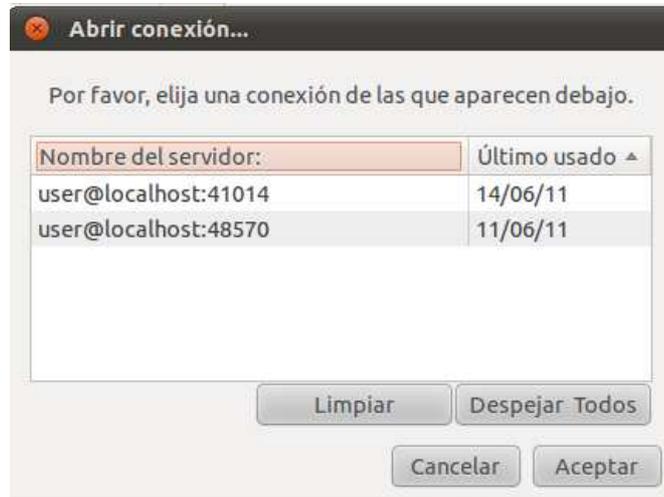


Figura 4.2.5: Menú conexiones recientes

Los botones  y  permiten poner en marcha o detener el stream en uso de forma inmediata. El tercer botón  permite eliminar el flujo configurado o bien el flujo que se seleccione de la lista. Para finalizar, el botón con forma de barita mágica, ejecuta el asistente para un nuevo flujo de streaming, iniciando la primera pantalla indicada en la figura 4.1.4.

En el visor del servicio se puede observar que está dividido en 2 partes, siendo una dedicada a mostrar los flujos en uso configurados en el Worker y otra de información sobre el Worker. En la zona donde se muestra los flujos activos, se muestra una lista tabulada donde aparece el componente iniciado, con que nombre se ejecuta y que PID asociado tiene.

	Componente	"Worker"	PID
	http-server-ondemand	localhost	2391

Figura 4.2.6: Menú conexiones activas

La información sobre cada flujo esta detallada a continuación dentro del apartado 4.3 que comienza a continuación.

### 4.3. Estadísticas Flumotion

Una vez instalado el servicio y configurado aparece ya la pantalla de administración e información del servicio, donde se muestra información relevante de conexiones activas, porcentajes de uso de CPU, el consumo que el cliente está haciendo sobre el servicio, así como estadísticas importantes para evaluar el estado del servicio arrancado. También en la parte superior izquierda se puede observar que se encuentra arrancado el rol “Worker” bajo el componente http-server-ondemand, en este caso particular el proceso está vinculado al PID 1819, aunque esto puede variar.

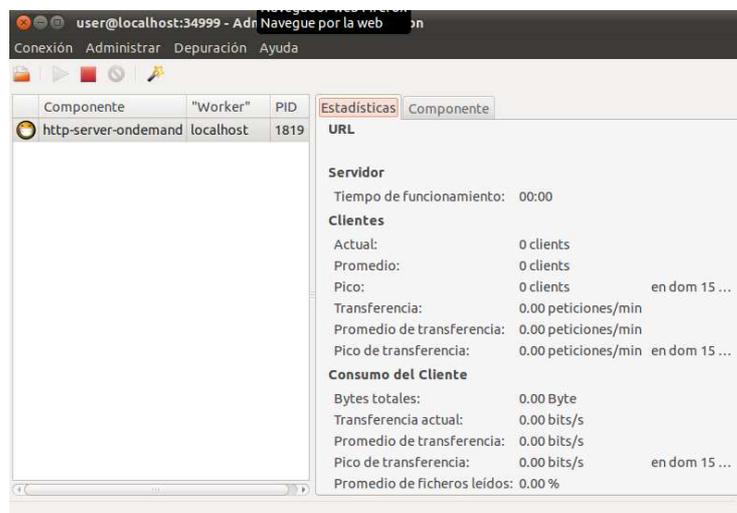


Figura 4.3.1: Estadísticas de consumo por cliente del servicio streaming

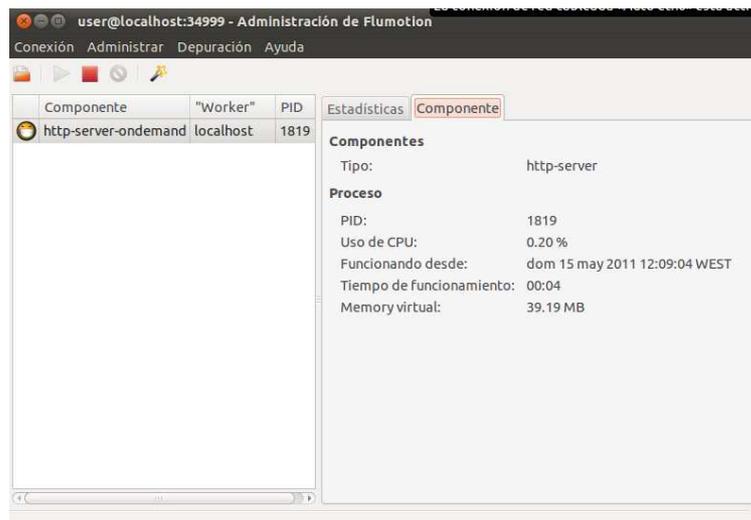


Figura 4.3.2: Consumo de recursos hardware del servicio Flumotion

A continuación se muestran las estadísticas de consumo y prerequisites de hardware necesarios a partir de un escenario de difusión de contenidos ya publicado en internet. Las mediciones que se han tomado están sujetas al consumo por cliente en el servicio flumotion, se pronostica que el site dispondrá de al menos 10000 usuarios al mes y que contendrá un repositorio con 1500 episodios, se estima que la duración del video por promedio es de aproximadamente de 30 minutos y que el tiempo máximo para empezar la reproducción de un video es de 1 minuto. También se estima que el bitrate necesario para difundir un video en calidad SD en los 3 formatos publicados será como máximo de 300kbps.

En las siguientes tablas de resultado se muestra datos relevantes a almacenamiento necesario, trafico concurrente y ancho de banda necesario para asegurar la calidad de servicio. El estudio evalúa para los distintos tipos de conexión disponibles en el hogar para ver los diferentes resultados según el cliente que se conecta.

### Información inicial del escenario a evaluar



Los datos para el streaming bajo demanda del proyecto:		
<b>Número de vídeos:</b>		2000
<b>Bitrate des/der Videos:</b>		300 kbps
<b>Duración de los videos:</b>	30 Minutos	0 Segundos
<b>Visitas esperadas por mes (total, no por el vídeo):</b>		10000
<b>Promedio de tiempo de visualización:</b>	30 Minutos	0 Segundos
<b>Almacenamiento:</b>		135000 MB
<b>Trafico:</b>		675 GB
<b>Ancho de banda concurrente necesario:</b>		7,5 Mbps

Tabla 4.2.1: Consumo de recursos hardware del servicio Flumoti

## Requisitos de almacenamiento para el servicio

### ALMACENAMIENTO NECESARIO

<b>Numero de videos:</b>		<b>2000</b>
<b>Duración del vídeo:</b>	<b>30</b> Minutos	<b>0</b> Segundos
<b>Tasa de codificación:</b>		<b>Necesidades de Almacenamiento:</b>
28.8k Modem	20 kbps	<b>4,5 MB</b>
56k Modem	40 kbps	<b>9 MB</b>
ISDN	80 kbps	<b>18 MB</b>
	100 kbps	<b>22,5 MB</b>
DSL	300 kbps	<b>67,5 MB</b>
	500 kbps	<b>112,5 MB</b>
	700 kbps	<b>157,5 MB</b>
<b>Definido por el usuario:</b>	<b>300</b> kbps	<b>67,5 MB</b>
<b>Requisitos de almacenamiento:</b>		<b>135000 MB</b>

Tabla 4.2.2: Requisitos de almacenamiento para el servicio

## Tráfico estimado en el servicio según escenario

<b>Trafico del mes</b>		
<b>Espera de visitas/mes</b>		<b>10000</b>
<b>Calidad de video</b>		<b>300</b> kbps
<b>Promedio de tiempo, para que un visitante vea un video:</b>	<b>30</b> Minutos	<b>0</b> Segundos
<b>Trafico esperado al mes:</b>		<b>675000 MB</b> <b>675 GB</b>

Tabla 4.2.3: Tráfico estimado en el servicio según escenario

## Videos concurrentes por Ancho de banda estimado

Necesidades de ancho de banda simultaneo	
<b>Visitas por mes:</b>	<b>10000</b>
<b>Puntuación por día</b>	333,3333
<b>Máximo de accesos por hora</b>	50
<b>Tiempo promedio de revisión:</b>	<b>30</b> Minutos <b>0</b> Segundos
<b>Visitantes simultáneos</b>	25
<b>Redondeo de visitantes</b>	25
<b>Bitrate de Videos:</b>	<b>300</b> kbps
<b>Requisitos de ancho de banda:</b>	<b>7500 kbps</b> <b>7,5 Mbps</b>

Tabla 4.2.4: Videos concurrentes por Ancho de banda estimado

Estas estadísticas muestran que en el caso de promover el proyecto y publicarlo en internet, sería necesario un buen equipamiento hardware para el repositorio de videos así como para los distintos frontal web y servidores de streaming. También hay que hacer hincapié, que la clave del éxito de este servicio está en su caudal de red, es decir, la inversión en una buena línea de comunicaciones es fundamental para obtener un buen servicio, tan solo hay que mirar que para 10000 conexiones al mes el tráfico generado es de 675GB lo cual indica que se está delante de un servicio exigente a nivel de consumo de red. También a nivel de disco se observa que al menos se necesitan 135GB para almacenar 2000 capítulos, teniendo en cuenta que el acceso al video debe ser rápido es muy probable que se necesite una cabina de disco redundada y en modalidad espejo para facilitar el acceso al contenido.

Sin duda se trata de una inversión inicial considerable pero también ante un sistema autónomo y eficaz según las pruebas.



## 5. Valoración económica

En este capítulo se estimará el esfuerzo económico que supondría realizar este proyecto en el caso de llevarlo a cabo. Este estudio económico parte de la base de las herramientas utilizadas durante su creación como del tiempo invertido en los 3 apartados que lo comprenden, partiendo del análisis y búsqueda de información y recursos, codificación del proyecto y ensayos de funcionamiento.

El precio por hora establecido para la persona que ejecutara el proyecto será de 16 Euros, siendo un valor razonable dentro del mercado actual de empleo.

### 5.1. Análisis, búsqueda de información y herramientas

Previa a la ejecución del proyecto, se ha tenido que dedicar un tiempo a 3 aspectos muy importantes. El primero, realizar un análisis de la plataforma que se quería crear, segundo, la búsqueda documental tanto en internet como en libros para tener los conceptos y la idea claras y tercero, las herramientas necesarias para poderlo llevar a cabo.

#### 5.1.1. Análisis y búsqueda de información

Para poder cuantificar el análisis y la búsqueda de documentación necesarias, se ha realizado un recuento de las horas invertidas en este proyecto. Se dividirá en tareas por hora y el precio estimado es el que se refleja más arriba por el coste de un ingeniero técnico. En ellas se contempla las sesiones con el tutor como parte del análisis.

A continuación se muestra una lista tabulada con la relación de Tarea y hora asociada a este apartado.

Tarea	Tiempo dedicado (h)	Precio Tarea €
Recopilación de información Multicast	4 h	64€
Información sobre Symfony	8 h	128€
Información sobre HTML5 y CSS3	4 h	64€
Etiqueta <video>, eventos asociados	6 h	96€
<b>Reunión 1 TFC</b>	1 h	16€

Especificaciones técnicas del proyecto	26 h	416€
Conectividad y aprendizaje	4 h	64€
Diseño de tablas y relaciones	2 h	32€
Empresas del sector del Streaming	5 h	80€
Sistema de Puntuación	2 h	32€
<b>Reunión 2 TFC</b>	2 h	32€
Asesoría VideoLAN	2 h	32€
Visión General del mercado Audiovisual en Internet	15,5 h	248€
<b>Reunión 3 TFC</b>	1,5 h	24€
Estudio de Firefogg	7 h	112€
<b>Total:</b>	90 h	1440€

Tabla 5.1.1: Coste del Análisis del Proyecto

### 5.1.2. Herramientas y utilidades

En este proyecto tal y como se ha comentado durante los capítulos anteriores se ha procurado realizar con el máximo número de herramientas libres sin coste alguno para ofrecer valor añadido al proyecto y a su vez reducir su coste de realización. A continuación se lista una tabla con la relación de precios de las herramientas hechas servir durante la creación del proyecto.

Herramienta o recurso	Precio de licencia
Netbeans Version 6.9.1	Gratuito
Visual Web Pack for Netbeans	Gratuito
Flumotion Streaming Server	Gratuito
Ubuntu Desktop 10.04	Gratuito
Microsoft Windows 7 Home Edition	199,99 €
Microsoft Office 2007 Profesional Edition	304,99 €
VmWare Player	Gratuito
Servidor Web LAMPP Linux	Gratuito
<b>Total:</b>	504,98 €

Tabla 5.1.2: Coste de las herramientas utilizadas

A la vista de los resultados se obtiene que las herramientas hechas servir para redactar como el sistema operativo del equipo cliente son los que verdaderamente tienen un coste sobre el proyecto, el resto de herramientas que hace posible este proyecto son todas de carácter gratuito.

## 5.2. Codificación

En este apartado se contabilizara las horas dedicadas a todas aquellas tareas dedicadas exclusivamente a la programación o codificación del proyecto. Del mismo modo que en el análisis, se mostrara el coste por tarea durante la creación del mismo, teniendo en cuenta en que únicamente hay una persona trabajando en el, en el caso de haber más personas, el número de horas se vería sensiblemente reducido.

Previamente se mostrara el tiempo invertido del montaje el escenario necesario para poder codificar el proyecto.

Tarea	Tiempo dedicado (h)	Precio Tarea €
Montaje de servidor remoto en hogar	2 h	32€
Instalación Servidor Web Linux	2,5 h	40€
Instalación de Cliente de Pruebas Linux	1 h	16€
Configuración Flumotion	4 h	64€
Adecuación de videos	10 h	160€
<b>Total:</b>	19,5 h	312€

Tabla 5.2.1: Coste de montaje del escenario

A continuación se mostrara una lista tabulada con la misma relación aplicada durante el análisis.

Tarea	Tiempo dedicado (h)	Precio Tarea €
Diseño de tablas y relaciones	2 h	32€
Creación de tablas en MySQL	2 h	32€
Creación de página de Inicio	1,5 h	24€
Creación del modulo de login	2 h	32€
Instalación de Entorno de Pruebas	3 h	48€
Creación de Perfil Usuario	3,5 h	56€

Creación del modulo Serie	3,5 h	56€
Redirección de login personalizada	2,5 h	40€
Creación de Modulo Temporada	5 h	80€
Personalización barra información User Layout	2 h	32€
Creación del buscador de series	20 h	320€
Acceso Serie –Temporada-Capitulo	4 h	64€
Creación de Modulo Capitulo	2 h	32€
Personalización vista temporada	3 h	48€
Sistema de Puntuación	2 h	32€
Aspectos Personalizados CSS	1,5 h	24€
Creación de Paginas Acuerdo Legal y PoNNLoAPI	2 h	32€
Limitaciones de visionado	32 h	512€
Firefog codificación Video	4 h	64€
Total:	97,5 h	1560€

Tabla 5.2.2: Coste de codificación

### 5.3. Documentación y pruebas

La estimación del coste referido a documentar este proyecto es orientativa ya que su redactado ha sido constante y su cuantificación lo hace más difícil. No obstante las horas invertidas aproximadas serian alrededor de unas 120 horas, teniendo en cuenta el tiempo calculado para el análisis y las pruebas. Por lo tanto, documentar este proyecto sabiendo que el salario por hora establecido es de 16 €/hora ascendería a **1920 €**.

Durante la creación de este proyecto se han realizado distintas pruebas tanto de forma local como con clientes remotos. En este apartado no solo se contarán las pruebas de visionado realizadas por los clientes, sino que también se ha contemplado las diferentes pruebas de carga de datos y operatividad del servicio. En la siguiente tabla y siguiendo con el mismo formato que las anteriores se muestra las tareas realizadas dentro de este aspecto y su respectivo precio orientativo.

Tarea	Tiempo dedicado (h)	Precio Tarea €
Prueba 1: Carga información general Series	5 h	80€
Prueba 2: Carga de información de Temporadas	4 h	64€
Prueba 3: Carga información de Capítulos	3 h	48€
Prueba 4: Carga de información de Usuarios	3 h	48€
Prueba 5: Pruebas de navegación de Usuario	2 h	32€
Prueba 6: Prueba de subida y carga de Videos	8 h	128€
Prueba 7: Codificación de Video desde Web	8 h	128€
Prueba 8: Visionado de Series local	4 h	64€
Prueba 9: Visionado de series en Remoto	3 h	48€
Prueba 10: Visionado de series con más de 5 clientes concurrentes	2 h	32€
<b>Total:</b>	42 h	672€

Tabla 5.3: Coste de pruebas

## 5.4. Hardware inicial

Para poder llevar a cabo este proyecto ha sido necesario únicamente un portátil HP Pavilion DV2000 cuyo precio de mercado actual ronda los 465 €, la amortización aplicable sobre el hardware hecho servir durante el proyecto correspondería a **114,39€**, este sería el único coste a nivel HW para poder llevar a cabo el desarrollo del proyecto.

Para poder llevar a cabo este proyecto a nivel profesional de una forma inicial, sería necesario contar con un servidor Web dónde hospedar el servicio y una cabina de discos donde alojar el contenido audiovisual, todo ello redundado para poder ofrecer un servicio continuado en caso de avería. Con esta premisa, se detalla a continuación un precio estimado del coste del hardware necesario.

Unidades	Hardware	Precio Unidad €	Precio total €
2	HP Proliant DL380 G7	1659,83€	3319,66€
2	HP StorageWorks 60 Modular Smart Array MSA60	3556,44€	7112,88€
	<b>Total:</b>		10432,54€

Tabla 5.4: Coste Hardware Inicial

## 5.5. Valoración final

Después de analizar los distintos apartados que han sido necesarios para crear este proyecto y cuantificarlos, se concluye que la realización de este proyecto ha supuesto un total de horas invertidas igual a 369h, lo que representa un coste estimado de **5.904 €**. Faltará añadir el precio de las herramientas y hardware utilizados durante la creación del mismo, que ascienden a un total de **965,98 €**. La amortización aplicable a las herramientas y hardware hechos servir durante la creación del proyecto dentro de las horas invertidas en el mismo sería de **237,63€**, estableciendo siempre la proporción anual de horas sobre las **369h** invertidas durante su creación.

A continuación se muestra una tabla con la relación de costes directos, indirectos y amortizaciones aplicables al proyecto.

Tipo de Coste	Descripción	Precio €
Directo	Horas invertidas en la realización del proyecto	5904€
Indirecto	Costes no imputables a la actividad, asociado un 10% sobre el total del coste del proyecto. (proveedores de servicio, alquiler de despacho)	590,4€
Amortización sobre materiales	Software y Hardware necesario para la realización del proyecto. Se aplica una amortización anual respecto a las horas invertidas en el proyecto (369h/1500h)	237,63€

Tabla 5.5: Relación de costes

El precio final del proyecto obtiene un precio de **6.732,03€**. En esta valoración económica no se contempla el precio del hardware necesario para poder llevar a cabo este servicio de forma profesional, únicamente se cuantifica el estudio y creación del proyecto que se presenta. En el caso de querer valorar el proyecto para iniciar un modelo de empresa, sería necesario contabilizar el hardware para poder ofrecer un servicio con garantía. Tal y como se detalla en la tabla 4.2.3, se debería hacer una inversión inicial de **10432,54€** sin IVA, para poder llevar a cabo el proyecto de una forma profesional. En los costes no se refleja los costes derivados de alojamiento físico ni conectividad con ISP.

A la vista de los resultados económicos que se plantean, el proyecto PoNNLo en un inicio no supondría un coste excesivo a nivel de montaje, siempre y cuando no se contemple el hardware. No obstante su rendimiento económico dentro del sector audiovisual en Internet podría tener un impacto mucho mayor a su coste inicial.

## 6. Conclusiones y posibles ampliaciones

### 6.1. Propuestas de Mejora y posibles ampliaciones

A continuación se proponen una serie de mejoras y ampliaciones que se podrían aplicar a este proyecto.

#### **Sistema de Pago:**

Sería importante implementar una pasarela de pago bancaria así como la posibilidad de hacer pagos mediante PayPal para facilitar los pagos de los usuarios así como para activar la funcionalidad de pago por cuenta bancaria contemplado en el proyecto tal y como se refleja en las propiedades del perfil de usuario.

#### **Internacionalización:**

Implementar el sistema de control de idioma para facilitar el acceso a clientes de habla no hispana al servicio. Actualmente Symfony dispone de herramientas para hacer de un modo fácil, la adaptación de la aplicación web a otros idiomas. Habilitar esta funcionalidad habría más cuota de mercado del servicio.

#### **Sistema Colaboradores:**

Actualmente el servicio solo dispone de codificación al formato .ogv, sería conveniente poder hacer esta codificación a los 3 formatos que dispone el servicio para facilitar la autonomía del servicio. Sería conveniente estudiar el modo de subida de video a los repositorios de video de una forma segura y cifrada.

#### **Protocolo streaming:**

Es necesario indicar que el protocolo HTTP no está diseñado para retransmitir video de forma eficiente, a diferencia de otros protocolos como RTP y RTSP que lo están. Debido a la reciente aparición de HTML5 y de la etiqueta <video>, no ha sido posible incorporarlo al proyecto, no obstante sería interesante en el futuro establecer estos dos últimos protocolos como solución definitiva.

**Red Social:**

Para fomentar la comunicación entre clientes del servicio sería conveniente estudiar las vías disponibles para adaptar el framework symfony a las diferentes redes sociales ya existentes para poder establecer un canal de comunicación ágil a nivel interno de la propia aplicación entre los usuario.

**Video en alta definición:**

En futuras versiones, sería conveniente poder ofrecer los capítulos en 2 calidades diferentes, tanto en SD como en HD. Aunque a nivel de base de datos si se contempla la incorporación del video en alta calidad, sería conveniente codificar una solución para poder ofrecer al usuario el contenido en alta definición.

**Personalización del perfil de usuario:**

En estos momentos aun disponiendo de un área de usuario donde poder ver la información relevante del cliente, no es posible modificar los campos que el usuario previamente ha rellenado, se debería crear una plantilla para que el usuario pudiese tener esa opción disponible.

**Presentación de novedades:**

En futuras versiones sería conveniente poder mostrar las series más vistas de forma dinámica a partir de los campos de puntuación y visualización, actualmente está de forma estática y preestablecida por el administrador.

**Boletín de novedades:**

Sería interesante incorporar un sistema de correo electrónico para alertar al usuario de las últimas novedades aparecidas en la página, mediante una sencilla suscripción el cliente podría obtener información de los géneros de serie que más visualiza y de este modo mantenerlo más activo dentro del site.

**Estadísticas de consumo del cliente:**

Dentro del apartado “Información Personal”, sería útil que el usuario pudiera ver las series que ha visualizado así como el tiempo de visualización, los puntos o dinero que le restan por consumir para otorgar al usuario cierto control sobre su estado.

**Limitaciones de usuario no registrado:**

Actualmente aun disponiendo de un sistema básico para limitar la visualización de contenidos si no se está registrado, sería conveniente profundizar más en la gestión de eventos de la etiqueta video para sugerir al usuario que renueve su suscripción al servicio o bien amplíe su saldo para continuar visualizando el contenido. Actualmente solo se muestra un mensaje y se bloquea la pantalla del usuario.

**Área de soporte y asistencia al usuario:**

Sería conveniente establecer un área de soporte al usuario, para poder ofrecer ayuda a incidencias provocadas durante el servicio o bien servir como guía. Se trataría de un servicio siempre visible en la pantalla para que el usuario pueda acceder de forma fácil al sistema de tickets de la propia aplicación. Esto ofrecería valor añadido al proyecto haciéndolo más estable.

**Sistema de búsqueda:**

Actualmente con el framework Zend se ha podido establecer un sistema de búsqueda básico con el que el usuario puede realizar sus peticiones, no obstante este sistema es aun algo básico ya que necesita tener coincidencias exactas de aquello que se busca. Sería interesante que el buscador pudiera predecir con tan solo 3 caracteres que contenido busca el usuario dentro de la aplicación.

**Módulos Backend:**

En la versión actual el entorno de administración es relativamente básico disponiendo de un único modulo activo, en este caso el modulo para colaboradores llamado “upload”, sería

conveniente crear nuevos módulos para facilitar la edición, modificación e incorporación de títulos nuevos a la aplicación y de este modo facilitar el trabajo al administrador.

## **6.2. Conclusiones**

Sin duda alguna este proyecto me ha hecho ver el potencial que el desarrollo web tiene, ha sido mi primera toma de contacto con multitud de herramientas como MySQL WorkBench así como el aprendizaje de un framework PHP nuevo con el que no me sentía para nada familiarizado. El hecho de querer introducir tecnologías muy innovadoras buscando como fin aportar valor añadido al servicio me ha supuesto muchos problemas y trabas durante el camino. La incorporación también de complementos de firefox tan nuevos como Firefogg me ha hecho buscar información y también tratar de buscar soluciones por mí mismo por la falta de ella, eso me ha enseñado que muchas veces innovar tiene un precio alto. Siento con este proyecto que se ha podido conseguir preparar un modelo de aplicación web moderna y actual que tiene cabida en estos tiempos donde el sector audiovisual se siente tan necesitado de ideas. Con este proyecto desde su inicio hasta su fin se ha tratado de ofrecer soluciones ingeniosas para poder acabar diseñando una página que cumple con los requisitos básicos para poder difundir contenidos audiovisuales, siguiendo el camino que otras compañías como Spotify, Microsoft Market Place, etc, se ha tratado de poder dar una solución a todos aquellos internautas que a día de hoy recorren a páginas web ilegales para ver las series que les gusta.

Con todo ello, también siento que esta aplicación esta aun en una fase inicial y le falta algunas mejoras, no obstante tengo la sensación de que verdaderamente es un proyecto con viabilidad y futuro, que es exactamente lo que me atrajo de él antes de diseñarlo. En todo momento se ha tratado de ofrecer una solución íntegramente desarrollada bajo Open Source y de forma gratuita, para reducir el coste de la solución y al mismo modo demostrar que existen tecnologías, aplicaciones y herramientas gratuitas muy profesionales con las que diseñar una aplicación comercial.

De este proyecto también puedo decir que he aprendido conceptos del sector audiovisual que antes no tenía, como por ejemplo las propiedades de cada formato y contenedor así como el motivo del porque en unos escenarios se aplica uno y en otros otro. La codificación del video así como las pruebas de retransmisión del mismo, me han acercado

un poco más a los requisitos reales de un servicio de difusión profesional y me ha hecho entender su dificultad. En todo momento tal y como se ha comentado en líneas anteriores se ha pretendido que este servicio simulase de la forma más real posible el comportamiento de una estructura de streaming compleja en Internet, el conjunto de las aplicaciones hechas servir como el diseño pensados hacen que sea posible en el futuro poder tener a nuestro alcance servicios como el que en este proyecto se estudia.

Lo más importante en este proyecto es poder acercar una solución ingeniosa a toda la comunidad de internet que se siente algo defraudada por la espera de las compañías audiovisuales a actuar y a tomarles en serio dentro del mercado de alquiler de videos. En este caso son series de televisión, pero podría trasladarse a muchos escenarios distintos sin necesidad de modificar mucho la base de la aplicación, por eso pienso que el proyecto PoNNLo ha conseguido definir, diseñar e implementar una solución viable a la difusión de video en Internet.



## 7. Referencias.

### HTML5:

- [1] <http://elpezojo.blogspot.com/2010/08/23-recursos-esenciales-para-entender.html> 23 Recursos esenciales para entender HTML5. [22/01/2011]
- [2] <http://www.w3.org/TR/html5/video.html#the-source-element> W3C Working Draft for HTML5. [28/01/2011]
- [3] <http://www.whatwg.org/specs/web-apps/current-work/#stream-api> Living Standard HTML5. [03/02/2011]
- [4] <http://www.chipwreck.de/blog/2010/03/01/html-5-video-dom-attributes-and-events/> HTML5 Video – Dom Attributes and Events. [26/02/2011]
- [5] <http://dev.opera.com/articles/view/everything-you-need-to-know-about-html5-video-and-audio/> Everything you need to know about HTML5 video and audio. [15/03/2011]
- [6] <http://blog.steveheffernan.com/2010/04/how-to-build-an-html5-video-player/> How to Build an HTML5 Video Player. [02/04/2011]
- [7] [http://www.w3schools.com/html5/html5\\_ref\\_eventattributes.asp](http://www.w3schools.com/html5/html5_ref_eventattributes.asp) HTML5 Event Attributes. [02/05/2011]

### Artículos y empresas Streaming:

- [8] [http://es.wikitel.info/wiki/Televisi%C3%B3n\\_IP\\_multicast](http://es.wikitel.info/wiki/Televisi%C3%B3n_IP_multicast) Televisión IP Multicast Universidad de Alicante. [07/03/2011]
- [9] <http://bulma.net/body.phtml?nIdNoticia=2290> Ofrecer live stream de video y audio por Joan Ginard. [10/03/2011]
- [10] <http://www.idg.es/iworld/articulo.asp?id=119675> El Streaming se gana a los grandes en la industria por IDG. [02/04/2011]

- [11] <http://www.interlake.net/> Proveedor de Streaming Europeo. [17/03/2011]
- [13] <http://www.flumotion.com/blog/open-source> Proyecto Open Source Streaming. [22/03/2011]
- [14] <http://www.crealogica.com/Streaming/Servidores/Streaming.html> Empresa de Servicios dedicada al servicio de Streaming. [01/04/2011]

### **JQuery:**

- [15] <http://jqueryui.com/home> Colección de librerías JQuery. [04/04/2011]
- [16] <http://www.recortex.com/pagina/1> Repositorio público de código. [02/04/2011]

### **Javascript:**

- [17] <http://www.anieto2k.com/2010/01/14/anade-subtitulos-a-tus-videos-con-javascript/> Añadir subtítulos a tus videos con javascript por ANieto2k. [06/04/2011]

### **Symfony PHP:**

- [18] [http://www.symfony-project.org/jobee/1\\_2/Propel/es/01](http://www.symfony-project.org/jobee/1_2/Propel/es/01) Practical Symfony is a part of the official symphony documentation. [01/03/2011]
- [19] <https://groups.google.com/group/symfony-es?hl=es&pli=1> Grupo Google Oficial Symfony de Soporte. [01/03/2011]
- [20] <http://symfony.davidvega.net/> Symfony por David Vega Tips para desarrolladores. [06/03/2011]
- [21] [http://www.librosweb.es/symfony\\_1\\_0/](http://www.librosweb.es/symfony_1_0/) Symfony 1.0, la guía definitiva. [25/02/2011]
- [22] [http://www.librosweb.es/symfony\\_formularios/](http://www.librosweb.es/symfony_formularios/) Los formularios de Symfony 1.2. [15/03/2011]

**Reproductores y servicios P2PTV:**

- [23] <http://code.google.com/p/sopcast-player/downloads/list> Reproductor gratuito SopCast. [22/04/2011]
- [24] <http://www.tlc-networks.polito.it/leonardi/papers/demo-p2p2010.pdf> Network Friendly P2P-TV: The Napa-Wine Approach. [24/04/2011]
- [25] <http://www.coolstreaming.us/forum/forum-generale-p2p-tv/7618-vari-iptv-p2p-tv-software.html> Listado de reproductores gratuitos P2P-TV. [24/04/2011]

**Firefogg:**

- [26] <http://firefogg.org/> Video encoding and uploading for Firefox. [28/04/2011]
- [27] <http://diveintohtml5.org/video.html> Video on the Web nº5. [17/05/2011]

**Artículos y estudios de mercado:**

- [28] La regulación de los contenidos audiovisuales en Internet, escrito por Angel I. García Castillejo.
- [29] Proyecto de Ley Economía Sostenible, por el Ministerio de Economía y Hacienda.
- [30] Estudio de los perfiles existentes y las necesidades formativas en el ámbito del sector de Imagen y Sonido, realizado por IRCUAL

**Libros y Manuales:**

- [31] Flumotion Manual for Murray Cumming, documentation Team Flumotion.
- [32] HTML cheat sheet Quick Reference Guide HTML5.
- [33] Manual de HTML5 español, por Alejandro Castillo Cantón.
- [34] Symfony Guide Getting Started, por SensioLabs.
- [35] 30 Symfony Best Practices, escrito por Nicolás Perriault.
- [36] El video en Internet, escrito por Juan Jiménez Martínez.

