



UNIVERSIDAD DE CHILE  
DEPARTAMENTO DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DISEÑO Y VALIDACIÓN DE LAS COMPONENTES DE UN PROVEEDOR DE  
SERVICIOS DE ACADEMIA VIRTUAL ECONÓMICO Y CON FINES  
DOCENTES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL  
ELECTRICISTA

CRISTIAN ANDRÉS HUEPE REBOLLEDO

PROFESOR GUÍA:  
ALFONSO OCTAVIO EHIJO BENBOW

MIEMBROS DE LA COMISIÓN:  
NÉSTOR BECERRA YOMA  
NICOLÁS BELTRÁN MATURANA

SANTIAGO DE CHILE  
ENERO 2010

RESUMEN DE LA MEMORIA  
PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL  
ELECTRICISTA  
POR: CRISTIAN HUEPE R.  
FECHA: ENERO 2010  
PROF. GUÍA: Sr. ALFONSO EHIJO B.

“DISEÑO Y VALIDACIÓN DE LAS COMPONENTES DE UN PROVEEDOR DE  
SERVICIOS DE ACADEMIA VIRTUAL ECONÓMICO Y CON FINES  
DOCENTES”

Los actuales sistemas de gestión de contenidos y de aprendizaje no son suficientes para asegurar una correcta comprensión de los cursos a tratar en estos. Es por esto, que la integración de nuevos componentes que permitan que un tutor entregue los contenidos de forma remota se hacen necesarios. Estos componentes son audio-video y sincronización de presentaciones, los que permiten potenciar los sistemas de aprendizaje basados en web.

El objetivo general del presente trabajo de título es mostrar alternativas open source o de código libre que permitan llevar a cabo el diseño e implementación de servicios basados en web, con la finalidad de ser usados en un proveedor de servicios de academia virtual, y con esto proyectar nuevas aplicaciones en las cuales estos servicios permitan una mejoría sustancial en futuros desarrollos a un bajo costo.

Se utiliza una metodología general que consiste en realizar una recopilación de los antecedentes de las tecnologías, como protocolos de comunicación, códecs y las plataformas de software de carácter open source o código libre de preferencia que permitan cumplir los objetivos planteados en este trabajo de título. Se definen los servicios que ha de tener la academia virtual, tomando en cuenta el ámbito económico y los fines docentes de este trabajo. Se procede al diseño en base a los servicios, subdividiendo estos de manera modular según sus funcionalidades. Se construyen ejemplos de los servicios y se ponen a prueba, analizando el posible potencial y adaptabilidad a los objetivos propuestos en la memoria.

Como resultado final se entrega una revisión de la tecnología Flash, del servidor de streaming RED5, de la plataforma de software Openlaszlo, y de herramientas de conversión de documentos por medio de ejemplos funcionales aplicados a una academia virtual. Además se pone en funcionamiento el nuevo canal desarrollado para Asterisk, el cual permite la comunicación con Skype.

Finalmente se concluye que se ha implementado de manera exitosa los componentes de la plataforma de academia virtual, en los cuales está la transmisión de audio-video, sincronización de eventos y la conversión de documentos, cumpliéndose de esta forma los objetivos propuestos por utilizarse en su totalidad software open source o código libre. Con el adicional de nombrar futuros trabajos de título que puedan dar nuevas aplicaciones a los software usados dentro de esta memoria.

## **AGRADECIMIENTOS**

Al finalizar este importante proceso en mi vida, que significa el comienzo de nuevos cambios y desafíos, sueños y anhelos, es fundamental agradecer a quienes me han acompañado en el trayecto recorrido hasta ahora.

Es por ello, que en primer lugar agradezco a mis padres y hermanas, por su esfuerzo, preocupación y confianza incondicional, pues a pesar de la distancia, siempre sentí su respaldo y amor.

A mi Pauli por acompañarme, por darme su amor y animarme en cada momento durante este trabajo de título.

A mis amigos eléctricos, con quienes he compartido incontables momentos, los cuales han ido desde el rigor del estudio hasta los proyectos y aspiraciones que cada uno tiene para el futuro.

También quiero mencionar a mis amigos de Colegio, con los cuales comenzaron muchos sueños que hoy se cumplen, así como los amigos de primer año de Universidad, aunque muchos de ellos hayan tomado diversos rumbos a nivel de especialidad, han dejado con su amistad una huella imborrable en mi.

Finalmente, agradezco a mi profesor guía Alfonso Ehijo, por darme la oportunidad de realizar mi Trabajo de título en un tema de actualidad, quien supo darme las directrices que permitieron llevar a un término satisfactorio tantas horas de trabajo.

Muchas gracias a todos ustedes, ya que cada uno constituye parte de los cimientos que me han ayudado enormemente en el logro de esta importante meta, y seguirán conmigo en cada nuevo desafío que emprenda en el futuro.

## ÍNDICE GENERAL

<b>1. CAPÍTULO I: INTRODUCCIÓN</b> .....	<b>5</b>
1.1. Motivación.....	5
1.2. Objetivos.....	5
1.2.1. Objetivo General.....	5
1.2.2. Objetivos Específicos.....	5
1.3. Hipótesis del Trabajo.....	5
1.4. Alcances.....	6
1.5. Descripción del Documento.....	6
<b>2. CAPÍTULO II: REVISIÓN BIBLIOGRÁFICA</b> .....	<b>7</b>
2.1. Reseña histórica.....	7
2.2. Plataformas de Academia Virtual.....	8
2.2.1. Desarrollo de Plataformas Comerciales.....	8
2.2.2. Desarrollo de Plataformas Open Source.....	8
2.3. Opensource y Software Libre.....	8
2.4. Protocolos Utilizados.....	9
2.4.1. Protocolo RTMP.....	9
2.5. Tecnologías Involucradas.....	14
2.5.1. RED5 Servidor de streaming en Flash.....	14
2.5.2. OpenLaszlo.....	15
2.5.3. JODConverter.....	20
2.5.4. SWFTools.....	21
2.5.5. ImageMagick.....	21
2.5.6. Skype.....	21
2.5.7. Asterisk.....	22
2.5.8. Asterisk y Skype.....	24
2.6. Códecs y formatos contenedores.....	24
2.6.1. Video en Flash.....	24
2.6.2. Audio en Flash.....	30
2.6.3. Audio en telefonía IP.....	32
2.7. Funcionalidades adicionales de Flash.....	36
2.7.1. Objetos compartidos.....	36
2.8. Tecnologías de Medición en redes.....	37
<b>3. CAPITULO III: METODOLOGÍA</b> .....	<b>39</b>
3.1. Diseño solución.....	39
3.1.1. Audio y video.....	40
3.1.2. Presentaciones sincronizadas.....	41
3.1.3. Comunicación vía texto.....	42
3.1.4. Interfaz web integrada.....	42
3.2. Implementación solución.....	42
3.2.1. Comunicación audio y video.....	42
3.2.2. Presentaciones sincronizadas.....	43
3.2.3. Conversión de documentos.....	43
3.2.4. Telefonía-audio.....	44
3.3. Requerimientos de software.....	46
3.4. Requerimientos de hardware.....	46
<b>4. CAPITULO IV: RESULTADOS</b> .....	<b>48</b>

4.1.	Interfaces.....	48
4.1.1.	Interfaz web audio y video.....	48
4.1.2.	Interfaz web del conversor de documentos.....	49
4.1.3.	Interfaz web presentación.....	52
4.1.4.	Interfaz integrada.....	53
4.1.5.	Módulo de telefonía.....	55
<b>5.</b>	<b>CAPÍTULO V: DISCUSIÓN.....</b>	<b>56</b>
5.1.	Acerca del funcionamiento.....	56
5.1.1.	Comunicación audio video.....	56
5.1.2.	Comunicación vía texto.....	56
5.1.3.	Presentaciones y conversor de documentos.....	57
5.1.4.	Módulo de telefonía.....	58
5.1.5.	Mejoras en la Interfaz.....	58
5.2.	Propuesta de nuevos servicios a futuro.....	58
5.3.	Acerca de los objetivos.....	61
<b>6.</b>	<b>CAPITULO VI: CONCLUSIONES.....</b>	<b>62</b>
<b>7.</b>	<b>CAPITULO VI: ANEXOS.....</b>	<b>64</b>
7.1.	Referencias bibliográficas.....	64
7.2.	Configuraciones.....	65
7.2.1.	Pasos de configuración de SFA (Skype for Asterisk).....	65
7.2.2.	Pasos de instalación del conversor.....	67
7.2.3.	Código Interfaz Audio Video.....	67
7.2.4.	Código Interfaz Presentaciones.....	69
7.2.5.	Código Interfaz Consultas.....	73
7.2.6.	Código Interfaz Conversor.....	74

## ÍNDICE DE FIGURAS

Fig 2.1. Formas de desplegar la aplicación creada en código LZX. ....	17
Fig 2.2. Flujo de datos entre el servidor Openlaszlo y la aplicación.....	18
Fig 2.3. Flujo de datos entre los servicios, el servidor y la aplicación OpenLaszlo.....	19
Fig 2.4. Posibilidades de conexión vía el programa Skype.....	22
Fig 2.5. Posibles interconexiones con Asterisk.....	23
Fig. 3.1. Diseño funcional de la solución.....	39
Fig. 3.2. Arquitectura que sustentará algunos servicios de la academia virtual.....	41
Fig. 3.3. Diagrama de la configuración de pruebas audio-video y sincronización de eventos.....	44
Fig. 3.4. Diagrama de la configuración de prueba módulo telefonía-audio.....	45
Fig. 3.5. Diagrama de cómo se inserta Asterisk+Skype para una audio-conferencia.....	45
Fig. 4.1. Ventana de consulta de acceso a cámara y micrófono.....	48
Fig. 4.2. Interfaz ejemplo de audio y video en web.....	49
Fig. 4.3. Esquema del código php del conversor de documentos.....	50
Fig. 4.4. Página del conversor de documentos.....	51
Fig. 4.5. Interfaz ejemplo de presentaciones en web.....	52
Fig. 4.6. Interfaz integrada para el tutor.....	53
Fig. 4.7. Interfaz integrada para el alumno.....	54
Fig. 4.8. Comunicación de la interfaces con el servidor RED5.....	55

## ÍNDICE DE TABLAS

Tabla 2.1. Tipos de datos que puede enviar un paquete RTMP.....	12
Tabla 2.2. Tipos de datos utilizados para objetos compartidos.....	12
Tabla 2.3. Posibles largos del encabezado del paquete RTMP.....	13
Tabla 2.4. Códecs de video que están soportados.....	25
Tabla 2.5. Extensión de archivo Flash video y su descripción.....	26
Tabla 2.6. Descripción de los campos de la cabecera FLV.....	26
Tabla 2.7. Descripción del campo de FLV Stream.....	27
Tabla 2.8. Descripción de los campos de la etiqueta FLV.....	27
Tabla 2.9. Descripción del campo de la etiqueta anterior para verificar el tamaño.....	27
Tabla 2.10. Descripción de los campos de los tipos de etiqueta.....	28
Tabla 2.11. Información contenida en un paquete sobre el códec de audio empleado.....	28
Tabla 2.13. Códecs de audio que están soportados.....	30
Tabla 2.14. Datos importantes contenidos en un paquete de Metadata.....	31
Tabla 2.15. Comparativas entre Anexos del códec G.729.....	34
Tabla 3.1. Características del servidor.....	47

# **1. CAPÍTULO I: INTRODUCCIÓN**

## **1.1. Motivación.**

A lo largo de los años, los costos en traslado a los centros de estudio han crecido sostenidamente, esto ha ido impidiendo la transferencia de conocimientos de forma eficiente, sobre todo en un país que apunta a la globalización. El desarrollo de plataformas de aprendizaje basadas en web, que puedan aprovecharse de los avances en las aplicaciones de código libre y open source, y de esta forma reducir drásticamente los costos incurridos en la compra de licencias, es la principal motivación de este trabajo de título.

## **1.2. Objetivos.**

### **1.2.1. Objetivo General.**

El objetivo general del trabajo de título es generar una plataforma que sea capaz de transmitir audio-video y sincronización de eventos desde el computador del profesor en base a herramientas de bajo costo, con la finalidad de ser usados en un proveedor de servicios de academia virtual, y con esto, proyectar nuevas aplicaciones en las cuales estos servicios permitan una mejoría sustancial en futuros desarrollos a un bajo costo.

### **1.2.2. Objetivos Específicos.**

- Dar a conocer alternativas open source o de código libre para desarrollar aplicaciones en Internet enriquecidas basadas en Web.
- Dar a conocer alternativas open source o de código libre para la transmisión de video y audio en streaming, como también el uso de objetos compartidos remotos para la sincronización de presentaciones y habilitación de módulos de chat.
- Desarrollar ejemplos en los cuales, por medio de un browser, se puedan mostrar los diferentes servicios aptos para ser usados en una academia virtual.
- Proyectar nuevas aplicaciones de estos servicios que están basados en plataformas open source o de código libre.

## **1.3. Hipótesis del Trabajo.**

Se plantea como hipótesis, que las componentes de la academia virtual, audio-video y sincronización de eventos propuestas en este trabajo de título sean posibles de desarrollar con herramientas open-source o de código libre. En donde el tutor requiera de una computadora con audifono, micrófono y una cámara web, para conectarse a un sistema basado en web que permita realizar una clase virtual en donde el alumno solo requiera una computadora básica conectada a Internet.

## **1.4. Alcances.**

El trabajo de título tiene relevancia en todo el campo de la docencia y en la capacitación profesional a distancia, como también, en el uso de alternativas de bajo costo para implementar nuevos sistemas web que requieran el empleo de audio, video y sincronización de eventos.

## **1.5. Descripción del Documento**

**Introducción:** En este capítulo se da un marco general del documento, planteándose las motivaciones del trabajo de título y los alcances de este. Se define además, el objetivo general, los específicos y la hipótesis de trabajo. Se incluye, también, una descripción de los contenidos de cada uno de los restantes capítulos del documento de memoria.

**Antecedentes:** En este capítulo se tratan temas que son necesarios para contextualizar la memoria, el aporte de ésta e información que permita al lector entender los temas como protocolos, códecs, herramientas de software y hardware, entre otros.

**Metodología:** Se realiza un análisis de las alternativas para lograr los objetivos propuestos, y en base a esto, un diseño e implementación de los servicios para una academia virtual.

**Resultados:** Se presentan y analizan los resultados obtenidos en base al diseño descrito en el capítulo anterior.

**Discusión:** Se discuten los servicios en base a los módulos de ejemplo que fueron desarrollados y a una proyección de estos en futuros servicios multimedia.

**Conclusiones:** Se presenta un análisis de cumplimiento de los objetivos propuestos al inicio del trabajo, además, de los alcances y proyecciones en este u otros donde sea posible la utilización de la plataforma probada.

## **2. CAPÍTULO II: REVISIÓN BIBLIOGRÁFICA.**

### **2.1. Reseña histórica**

#### **Fases de evolución de las tecnologías que dan el soporte en sistemas de aprendizaje.**

La primera fase de evolución del aprendizaje vía computadora comienza con las primeras librerías de contenidos (principalmente enciclopedias, programas interactivos de aprendizaje, etc), las cuales fueron principalmente distribuidas por medios físicos, como disquete y CDROMs. Luego, con la aparición de Internet, el acceso a estas primeras librerías aumentó radicalmente, permitiéndose compartir esta información sin restricciones.

La segunda fase está catalogada por el surgimiento de los sistemas de administración del aprendizaje (en inglés, LMS, Learning Management System), en donde un software basado en un servidor web provee módulos para los procesos administrativos y de seguimiento del alumno.

La tercera fase se destaca por la incursión de empresas que se especializan en albergar y mantener un sistema LMS en funcionamiento, dado que no todas las organizaciones disponían de los recursos necesarios.

La fase cuatro corresponde a un intento de integración entre las virtudes de los Sistemas de gestión de contenidos (en inglés, CMS, Content Management System) y de aprendizaje (en inglés, LMS, Learning Management System). Cabe destacar que estos dos sistemas han ido evolucionando de forma independiente y en esta fase se logra que el proceso de enseñanza/aprendizaje incluya herramientas que permitan tanto la distribución, generación y gestión de contenidos, como también, de herramientas de gestión del proceso de aprendizaje.

La fase cinco se destaca por dar continuidad a la fase cuatro, pero adicionando nuevos servicios que permitan la interacción, online en una clase, entre el profesor y el alumno.

## **2.2. Plataformas de Academia Virtual.**

### **2.2.1. Desarrollo de Plataformas Comerciales**

La mayoría de los desarrollos basados en web están hechos en Flash y son soportados por Adobe Flash Media Server, el cual se ha convertido en la solución líder del sector para el video de flujo continuo y la comunicación en tiempo real. Los principales browser usados en el mundo soportan el reproductor Adobe Flash Player, siendo este utilizado en el 98% de los ordenadores del mundo. Flash Communication Server permite el diseño e implementación de todas aquellas herramientas colaborativas como son conferencias de audio y video multiusuario, pizarra compartida, presentaciones remotas, entre otras. Este sistema se ha empleado en la mayoría de las universidades en torno al desarrollo de cursos por medio de una academia virtual.

### **2.2.2. Desarrollo de Plataformas Open Source.**

De acuerdo al párrafo anterior, la mayoría de los desarrollos están hechos en Flash. En la línea del open source o del código libre se encuentra el servidor Red5, el cual, es un símil de Adobe Flash Media Server. El servidor Red5 permite realizar la mayoría de las funciones sin pagar por licencias propietarias.

Se pueden encontrar desarrollos en aplicaciones de gestión de cursos o más conocidos como plataformas LMS. Además, de sistemas que integran video conferencia.

Dentro de la variada cantidad de aplicaciones se puede destacar Moodle, un LMS que posee una de las interfaces gráficas más amigables dentro de las plataformas open source y tiene todas las herramientas necesarias para usarla como una plataforma base. Además, es posible integrar un módulo de video conferencia por medio de la iniciativa OpenMeeting.

## **2.3. Opensource y Software Libre.**

Tanto Open Source como Free Software son movimientos sociales, preocupados por los aspectos legales ( vía derechos ó licencias) de los programas (software). Tienen diferencias filosóficas, pero pocas diferencias prácticas. Free Software tiene una visión moral “el software debería ser libre”. Open Source tiene una visión práctica “el software es mejor si su código es abierto“, permitiendo que sea mejorado por una mayor cantidad de personas y de esta forma evolucionar de forma más rápida.

El software open source puede ser propiedad de una empresa determinada. Esto, no asegura que se mantenga como open source, ó que una mejora que un externo introduzca pase a formar parte de la propiedad de la empresa, frente al código libre, en el que las mejoras deben seguir siendo abiertas y libres (se pueden ampliar, o mejorar), tal como lo dice, una de las cuatro libertades del software libre. El proyecto más importante del software libre es el sistema GNU concebido con el objetivo de crear un sistema operativo completo tipo Unix.

Existen variados modelos de negocio que sustentan estas iniciativas, el principal, es de asesoría y la adaptación de estos para fines específicos.[2]

## **2.4. Protocolos Utilizados**

### **2.4.1. Protocolo RTMP**

El Protocolo de Mensajería en Tiempo Real (RTMP, Real Time Messsaging Protocol) es un protocolo propietario desarrollado por Adobe Systems (anteriormente desarrollado por Macromedia) que se utiliza principalmente con Adobe Flash Media Server para transmitir audio y video a través de Internet hacia el cliente Adobe Flash Player.[3]

La información contenida aquí sobre este no es oficial y se atiene a un proceso de ingeniería inversa usando software de captura de paquetes, en ejemplo Wireshark y tcpdump, con el objetivo de comprender el protocolo.

Este es un protocolo simple, optimizado para conexiones de bajo ancho de banda. Puede soportar hasta 64 streams concurrentes sobre la misma red de conexión. Parte de cada cabecera del paquete AMF contiene el número de índice del stream. Un solo mensaje RTMP puede contener múltiple paquetes AMF. Cada paquete AMF tiene 128 bytes, excepto para el streaming de audio, el cual, es de 64 bytes.

Como en muchas redes corporativas y en el hogar se bloquean los puertos y protocolos que no se utilizan a través de conexiones con servidores de seguridad, existen variantes del protocolo que se pueden utilizar con éxito en entornos en los que las medidas de seguridad bloquean RTMP. Las variantes se enumeran a continuación:

- RTMP (por defecto)
- RTMPT (RTMP tunelizados)
- RTMPS (RTMP Seguro)
- RTMPE (RTMP codificado)
- RTMPTE (RTMP tunelizados cifrados)
- RTMFP (RTM Flow Protocol)

### **RTMP (por defecto)**

De forma predeterminada, Flash Player utiliza el protocolo RTMP sobre el puerto 1935. Si esto falla, se intenta de nuevo sobre los puertos 443 (RTMPS) y 80 (RTMPT) en un intento de solución ante la configuración del firewall, que impide las conexiones TCP / IP en los puertos no estándar. El puerto de administración es por defecto el 1111.

### **RTMPT (RTMP tunelizados)**

Este protocolo incluye los datos RTMP como datos válidos en formato HTTP, y se comunica utilizando solicitudes POST desde el cliente al servidor en forma predeterminada en el puerto 80. Debido a la falta de persistencia de las conexiones HTTP, RTMPT requiere que los clientes se actualicen periódicamente a fin de notificarse sobre los eventos que son generados por el servidor u otros clientes. Si bien RTMPT requiere de anchos de banda ligeramente superiores debido a la adición de cabeceras HTTP, el protocolo puede ser utilizado con éxito en entornos en donde las medidas de seguridad bloquean RTMP.

### **RTMPS (RTMP Seguro)**

Este es RTMP tunelizado vía HTTPS, con el puerto 443 por defecto.

### **RTMPE (RTMP codificado)**

Este protocolo es una versión mejorada y cifrada de RTMP. RTMPE es más rápido que SSL y no requiere certificado de gestión como SSL. Se especifica explícitamente RTMPE sin especificar un puerto, Adobe Flash Player escanea los puertos igual que con la norma RTMP, en el siguiente orden: 1935, 443, 80, y 80 (RTMPE) (compatible con Flash Player 9.0.115.0 mínimo, Adobe AIR, Adobe Media Player).

### **RTMPTE (RTMP tunelizados cifrados)**

Adobe Flash Media Server 3 introduce una versión mejorada del protocolo RTMP, que encripta el canal de comunicación, realizando un túnel cifrado a través de HTTP. El puerto predeterminado es 80 (compatible con Adobe Flash Player 9.0.115.0 como mínimo; Adobe AIR, Adobe Media Player). Los principales beneficios por encima de SSL (RTMPS) son el rendimiento, la facilidad de aplicación y un impacto limitado sobre la capacidad del servidor.

### **RTMFP (RTM Flow Protocol)**

RTMFP se basa en UDP (User Datagram Protocol). RTMFP soporta el envío de datos directamente desde Adobe Flash Player a otro, sin pasar por un servidor. Un servidor del lado de la conexión será siempre necesario para establecer la conexión inicial entre los usuarios finales y se pueden utilizar para proporcionar los datos del lado del servidor de ejecución o puertas de

enlace en otros sistemas. El usuario de un Flash Media Server también estará obligado a autorizar la búsqueda de direcciones de red y servicios de NAT para los clientes de Adobe Flash Player, impidiendo que se utilicen en una forma no administrativa.

## **Handshake**

**Cliente -> Servidor:** Envía Solicitud de Handshake. Esto no es un protocolo de paquetes, sino un sólo byte (0x03) seguido por 1536 bytes.

**Servidor <- Cliente:** Envía una respuesta al Handshake. No se trata de un paquete de RTMP, pero un sólo byte (0x03) seguido de dos trozos de 1536 bytes (para un total de 3.072 en bruto de bytes). La segunda parte del fragmento es la solicitud original del cliente enviada en la solicitud de handshake. El primer fragmento puede ser cualquier cosa. Utilizar bytes null no parece importar.

**Cliente -> Servidor:** Se envían 1536 bytes sin tratar, que corresponden a la segunda parte del fragmento de 1536 bytes del handshake generado por el servidor.

En este momento, el handshake está hecho y los próximos paquetes son RTMP.

**Cliente -> Servidor:** Envía el paquete de Conexión RTMP.

**Servidor <- Cliente:** El servidor responde y así sucesivamente se realiza la comunicación en RTMP.

## Tipos de Datos

Tipos de datos que pueden ser enviados por paquetes RTMP.

0x01	Chunk Size	Cambia el tamaño de los fragmentos de los paquetes
0x02	Unknown	Desconocido hasta ahora
0x03	Bytes Read	Enviar cada x bytes leídos por ambas partes
0x04	Ping	Ping es un mensaje de control de flujo, a subtipos
0x05	Server BW	Ancho de Banda de bajada del servidor
0x06	Client BW	Ancho de Banda de subida del cliente
0x07	Unknown	Desconocido
0x08	Audio Data	Paquete que contiene datos de audio
0x09	Video Data	Paquete que contiene datos de video
0x10 0x11	Unknown	Desconocido
0x12	Notify	Notificar una invocación que no espera una respuesta
0x13	Shared Object	Tiene Subtipos
0x14	Invoke	Llamada remota, utilizada también para transmitir las acciones.

**Tabla 2.1. Tipos de datos que puede enviar un paquete RTMP.**

## Tipos de datos de objetos compartidos

0x01	Connect
0x02	Disconnect
0x03	Set Attribute
0x04	Update Data
0x05	Update Attribute
0x06	Send Message
0x07	Status
0x08	Clear Data
0x09	Delete Data
0x0A	Delete Attribute
0x0B	Initial Data

**Tabla 2.2. Tipos de datos utilizados para objetos compartidos.**

## Estructura de Paquetes

Los paquetes RTMP consisten en una cabecera de longitud fija y un cuerpo de longitud variable de hasta 128 bytes. El encabezado puede venir en uno de los cuatro tamaños: 12, 8, 4, ó 1 byte (s).

Los dos bits más significativos del primer byte del paquete (que también cuenta como el primer byte de la cabecera) determinar la longitud de la cabecera. Ellos pueden ser extraídos por el byte ANDING con la máscara 0xC0. Los posibles largos de la cabecera se especifican en el cuadro siguiente.

Bits	Largo del encabezado
00	12 bytes
01	8 bytes
10	4 bytes
11	1 byte

**Tabla 2.3. Posibles largos del encabezado del paquete RTMP.**

## Streaming

Ciclo básico de lo que ocurre al momento de realizar un streaming:

**Cliente -> Servidor** : envía una petición CreateStream .

**Servidor -> Cliente**: envía una respuesta con un número streamIndex.

**Cliente -> Servidor**: envía que detecto que es una solicitud de publicación de contenido.

**Cliente -> Servidor**: enviar los paquetes de audio y video (los paquetes se envían desde la fuente como se indica en el streamIndex a través del mismo canal de la petición de publicación).

## **2.5. Tecnologías Involucradas.**

### **2.5.1. RED5 Servidor de streaming en Flash**

Red5 es un servidor Open Source para entregar contenido en streaming en Flash. Para ello utiliza el protocolo RTMP (Real Time Messaging Protocol) con lo cual se puede transmitir contenido en tiempo Real.[3]

Este servidor tiene todas las cualidades del Flash Media Server de Adobe. Utiliza la sintaxis de ActionScript Communication con la cual se pueden desarrollar aplicaciones de comunicación en tiempo real.

#### **Características**

Este servidor está escrito enteramente en Java y tiene las siguientes características:

- Streaming de Audio y Video (FLV y MP3).
- Facilidad para guardar Streaming del cliente.
- Intercambio de Objetos.
- Publicación Live Streaming.
- Soporte para AMF o Flash Remoting.

Gracias a lo anterior es posible crear aplicaciones como:

- Chats multiusuario en tiempo real.
- Streaming de video y MP3, logrando reproducir parte del video sin necesidad de descargar el video completo.
- Transmisión de señal de televisión en tiempo real.
- Chats con soporte de audio y video, con la cual los usuarios pueden transmitir lo que captura su webcam.
- Facilidad para implementar pantallas compartidas, en la cual múltiples usuarios pueden interactuar en una misma pantalla.
- Grabación de contenido en formato FLV.
- Protección de derechos de Autor (DRM) pues no se transmiten los archivos físicos, sino que se envía contenido en streaming.
- Análisis y reportes de datos en tiempo real.

Red5 nace como una alternativa al Flash Media Server, aunque siendo este último más robusto, y con un excelente soporte y versiones para la mayoría de los Sistemas Operativos, el costo muy elevado hace difícil la viabilidad de desarrollar aplicaciones pequeñas.

## 2.5.2. OpenLaszlo

OpenLaszlo es una plataforma para aplicaciones ricas de Internet que son fáciles de desarrollar y utilizar. La arquitectura del sistema OpenLaszlo combina la potencia y facilidad de uso de un diseño cliente/servidor con las ventajas administrativas y de costo-rendimiento de las aplicaciones web. [4]

Las aplicaciones OpenLaszlo están escritas en el lenguaje XML LZX, que compila, incluyendo, a partir de OpenLaszlo 4.0 en los formatos SWF7, SWF8, SWF9 y nativas del navegador JavaScript, también a veces llamado DHTML o Ajax. Las aplicaciones compiladas para SWF8 o SWF9 ejecutan en el reproductor de Flash 9.

### 1. Modos de implementación

Las aplicaciones OpenLaszlo pueden estar disponibles en la web, o que se instalen en una de dos maneras:

Como Proxy. El servidor de OpenLaszlo se ejecuta en una máquina y

- Compila los programas de código según sea necesario y envía el archivo resultante a ser ejecutado en el cliente.
- Interactúa como un Proxy entre el cliente y los otros servidores en la Internet, realizando la manipulación de datos cuando sea necesario.

En aplicaciones "SOLO" se utiliza el compilador OpenLaszlo para "precompilar" programas y hacer que el archivo resultante (en formato SWF o JavaScript) esté disponible en el servidor. Cuando se ejecuta en el cliente, la aplicación contacta a otros servidores directamente, sin mediación por el servidor de OpenLaszlo. Esta aplicación "sin servidor", o independiente del despliegue de OpenLaszlo se llama SOLO.

En general:

- Las aplicaciones "Proxied" pueden hacer algunas cosas que las aplicaciones SOLO no puede hacer, pero son más problemáticas para desplegar y, a veces se llevan a cabo más lentamente.

- Las aplicaciones SOLO son más fáciles de implementar y, a veces con mejor rendimiento.

El servidor de OpenLaszlo es una aplicación Java que se ejecuta en un contenedor de servlets J2EE. El servidor OpenLaszlo puede comunicarse con los servidores de back-end (que realizan las aplicaciones adicionales que se requiere para la ejecución) y fuentes de datos usando una variedad de protocolos. Las aplicaciones OpenLaszlo son escritas en LZX y son compilados por el servidor OpenLaszlo y servido, ya sea como código de bytes de un plug-in que se ejecuta en el navegador del cliente (como Flash o J2ME) o JavaScript (DHTML) que se ejecuta directamente por el propio navegador. Esto constituye la interfaz (en inglés, front-end). OpenLaszlo ejecuta aplicaciones de manera consistente y fiable en una gran variedad de sistemas operativos y tipos de dispositivos, incluyendo Windows, Pocket PC, Mac OS, Linux y Solaris, y una variedad de móviles y plataformas set-top box.

Cuando se compila para su ejecución en Flash Player, el servidor OpenLaszlo entrega la aplicación en el formato de archivo SWF (formato "shockwave") reconocido por Macromedia Flash Player (versión 7 y superior).

Cuando se compila para la ejecución de forma nativa en el navegador, el servidor OpenLaszlo entrega la aplicación en código en JavaScript. Este código es legible, pero optimizado en el tamaño y la velocidad de ejecución, con los comentarios eliminados, nombres de variables reducido, etc.

En el futuro, se espera que OpenLaszlo pueda entregar la aplicación en otros formatos o códigos a medida que estén ampliamente disponibles.

En el contexto OpenLaszlo, el cliente solicita una aplicación LZX que ejecute en el navegador web del usuario y el servidor de medios OpenLaszlo Server, que a su vez, se comunica con otros servidores. El cliente LZX y el servidor OpenLaszlo se comunican a través de HTTP, el servidor OpenLaszlo envía bytecodes y la aplicación LZX envía XML.

Todas las características de la plataforma OpenLaszlo, incluyendo el streaming de medios de comunicación y notificación, se entregan a través de HTTP o HTTPS. Las aplicaciones basadas en OpenLaszlo mantienen la compatibilidad con los cortafuegos corporativos estándar. Esta es una capacidad crítica para aplicaciones de Internet públicas.

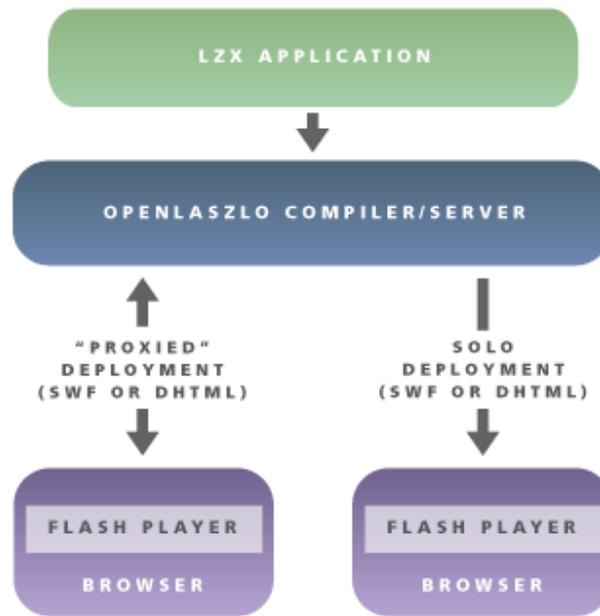


Fig 2.1. Formas de desplegar la aplicación creada en código LZX.

En la figura 3.1, se aprecian las dos formas de desplegar la aplicación creada en código LZX, en el caso de la aplicación “Proxied”, se mantiene una comunicaciones bi-direccional entre la aplicación LZX cliente y el Servidor Openlaszlo. En el caso de la aplicación SOLO, luego de ser compilada no se mantiene ninguna comunicación con el servidor Openlaszlo.

### Un flujo de datos de un servidor OpenLaszlo a una Aplicación OpenLaszlo

En este ejemplo se utiliza una aplicación OpenLaszlo almacenada en el servidor como weather.lzx. El siguiente diagrama ilustra cómo esta aplicación sería ejecutada por el servidor OpenLaszlo.

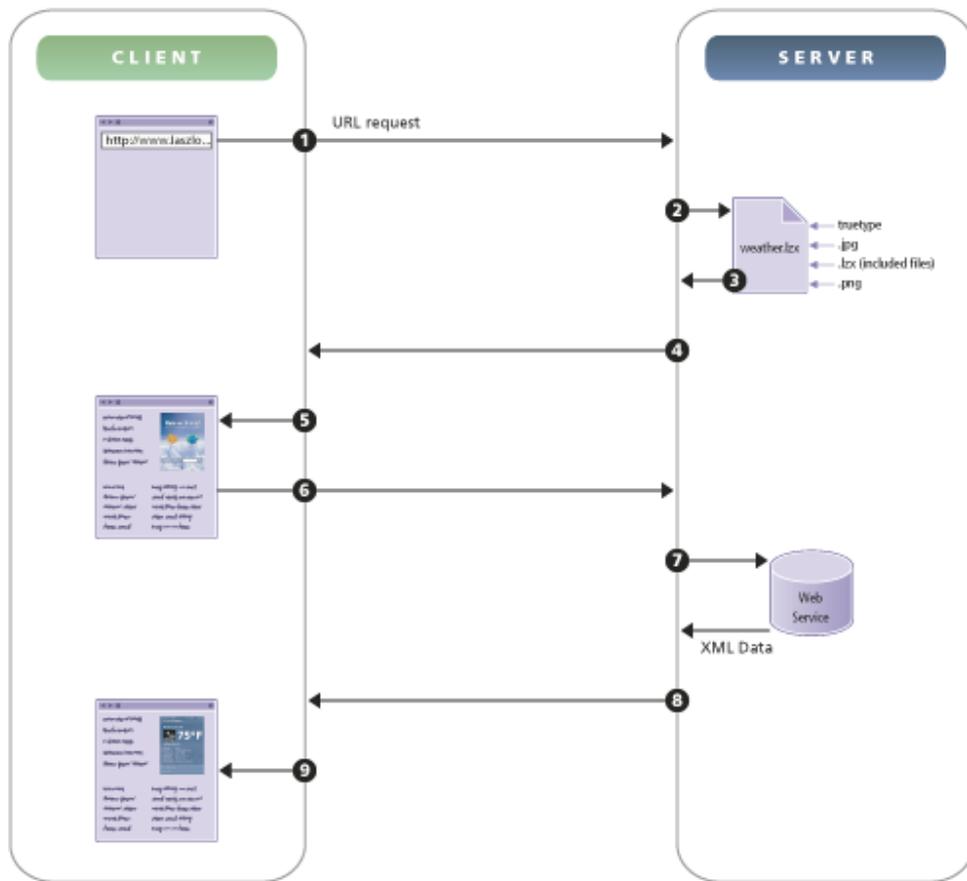


Fig 2.2. Flujo de datos entre el servidor Openlaszlo y la aplicación.

1. Petición de la URL <http://www.laszlo.com/weather.lzx>
2. El servidor OpenLaszlo localiza weather.lzx
3. El servidor OpenLaszlo compila todos los recursos (gráficos, fuentes, sonidos, componentes) dentro de un binario .swf
4. El Servidor OpenLaszlo entrega un archivo ejecutable en formato SWF o DHTML que incluye las Clases Bases de Laszlo(LFC).
5. Flash Player (SWF) o el browser (DHTML) despliega la aplicación.
6. Usuario ingresa un código ZIP.
7. El servidor OpenLaszlo invoca el apropiado XML con la información.
8. El servidor OpenLaszlo envía la información al cliente.

- Las Clases Bases de Laszlo en el cliente amarran la información al apropiado objeto de interfase de usuario. El apropiado elemento de pantalla es actualizado con la información transmitida.

A partir de un usuario que introduce una URL que solicita la aplicación “Weather”, este diagrama ilustra la secuencia de flujo de datos desde el cliente al servidor, integrando datos de un servicio Web XML, y enviando el resultado de vuelta al cliente.

El servidor OpenLaszlo envía al cliente datos binarios comprimidos en lugar de texto, reduciendo la cantidad de datos transportados en comparación con los basados en HTML y otras aplicaciones Web. El almacenamiento en caché en el servidor y cliente elimina la ejecución de código innecesario y de transmisión de datos.

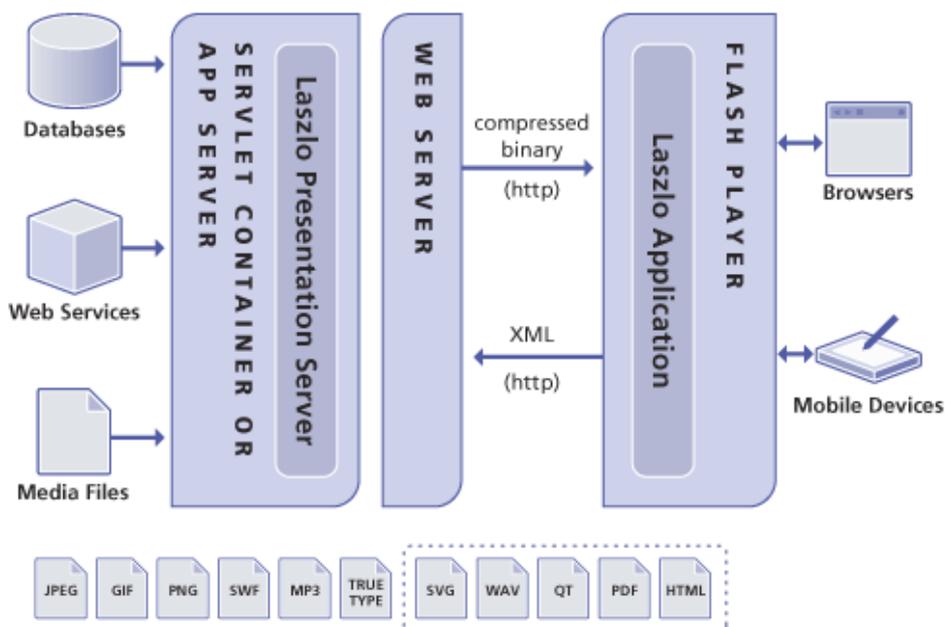


Fig 2.3. Flujo de datos entre los servicios, el servidor y la aplicación OpenLaszlo.

### 2.5.3 JODConverter

JODConverter, o Java OpenDocument Converter, convierte documentos entre diferentes formatos de oficina.[5]

Se aprovecha de la iniciativa OpenOffice.org, que ofrece indiscutiblemente las mejores herramientas de importación y exportación de documentos OpenDocument y de formatos de Microsoft Office disponibles hoy en día.

JODConverter automatiza todas las conversiones con el apoyo de OpenOffice.org, incluyendo:

Microsoft Office a OpenDocument, y viceversa

- Word a texto de OpenDocument (odt), texto de OpenDocument (odt) a Word.
- Excel a OpenDocument hoja de cálculo (ods); OpenDocument hoja de cálculo (ods) a Excel.
- PowerPoint o de OpenDocument Presentation (PAO); OpenDocument Presentation (PAO) a PowerPoint.

Cualquier formato a PDF

- OpenDocument (texto, hoja de cálculo, presentación) a PDF.
- Word a PDF o, Excel a PDF, PowerPoint a PDF.
- RTF a PDF, WordPerfect a formato PDF.

Y más

- Presentación OpenDocument (PAO) a Flash, PowerPoint a Flash.
- RTF para OpenDocument, WordPerfect a OpenDocument.
- Cualquier formato HTML (con limitaciones).

JODConverter se puede utilizar de muchas maneras diferentes

- Como una biblioteca de Java, incrustado en su propia aplicación Java.
- Como una herramienta de línea de comandos, posiblemente invocado desde un scripts.
- Como una aplicación web sencilla: cargar un documento de entrada, seleccionar el formato deseado y descargar la versión convertida.

- Como un servicio web, invocado desde una aplicación escrita desde un lenguaje como .NET, PHP, Python, Ruby.

Se destaca que JODConverter es software de código abierto liberado bajo los términos de la LGPL y se puede descargar desde SourceForge.net.

#### **2.5.4. SWFTools**

SWFTools es una colección de utilidades para trabajar con archivos de Adobe Flash (SWF). La colección de herramientas incluye los programas para leer los archivos SWF, la combinación de estos, y la creación desde otros contenidos (como imágenes, archivos de sonido, videos o código fuente) a archivos SWF. SWFTools es liberado bajo GPL.[6]

La aplicación que se usará en esta memoria es PDF2SWF, el cual convierte un PDF a SWF. Genera un cuadro por página. Permite manejar plenamente formato de texto, incluyendo tablas, fórmulas, gráficos, etc, dentro de la película Flash. Está basado en el analizador PDF xpdf de Derek B. Noonburg.

#### **2.5.5. ImageMagick**

ImageMagick es una suite de software para crear, editar y componer imágenes de mapa de bits. Puede leer, convertir y escribir imágenes en una variedad de formatos (más de 100) incluyendo DPX, EXR, GIF, JPEG, JPEG-2000, PDF, PhotoCD, PNG, Postscript, SVG y TIFF.[7]

La funcionalidad de ImageMagick es normalmente utilizado desde la línea de comandos o se puede usar las características desde programas escritos en el lenguaje de programación favorito.

ImageMagick es un software gratuito entregado de forma ejecutable o como código fuente que se puede utilizar libremente, copiar, modificar y distribuir. Su licencia es compatible con la GPL. Funciona en todos los principales sistemas operativos.

#### **2.5.6. Skype**

Skype básicamente es un software para realizar llamadas por Internet, fue fundada el año 2003 por los creadores de Kazaa. Este software puede ser descargado gratuitamente desde el sitio oficial, pero el código y protocolo son propietarios y se mantienen cerrados. Sus ventajas radican en ser un software empleado a nivel global, con más de 400 millones de usuario registrados.[8]

Dentro de los servicios que los diferencian de otros programas es la capacidad de recibir llamadas desde un número real de la red pública telefónica (mejor conocida por sus siglas en inglés PSTN), del país que se desee, este servicio tiene el nombre de SkypeIn. Como también realizar llamadas por medio de SkypeOut con tarifas muy económicas ya que la llamada entre ciudades o países viaja gratuitamente por la red de internet, y solo baja a la PSTN en la ciudad en donde se registra el número telefónico.

Este software también permite realizar videollamadas y mensajería instantánea.

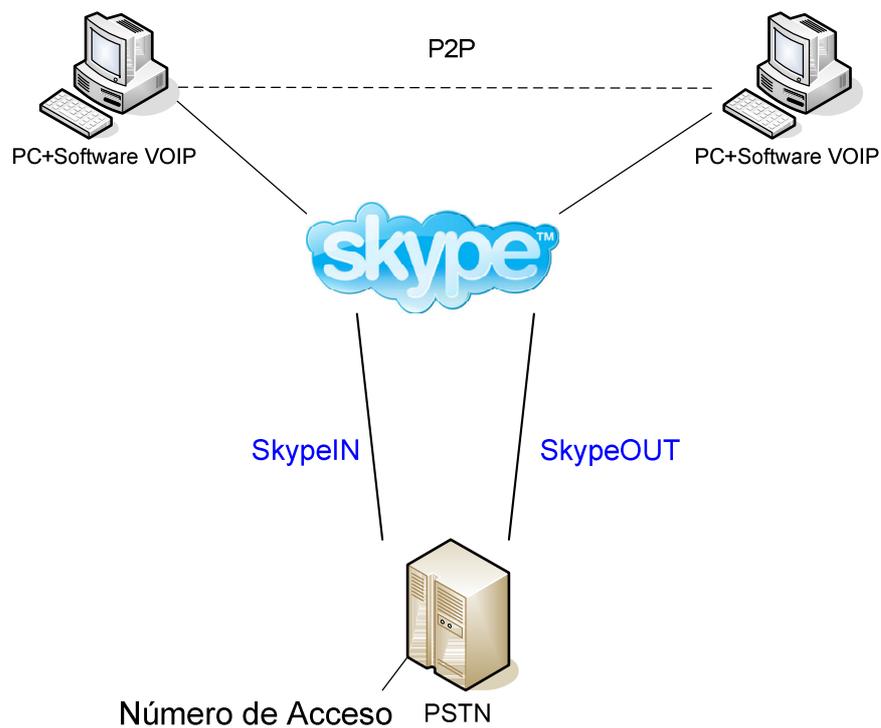


Fig 2.4. Posibilidades de conexión vía el programa Skype.

### 2.5.7. Asterisk

Asterisk es el líder mundial en su uso como sistema de telefonía de código abierto y con un amplio kit de herramientas que le dan cada vez más y mejores funcionalidades. Ofrecer flexibilidad sin precedentes en el mundo de las comunicaciones que se rigen de sistemas propietarios, Asterisk permite a los desarrolladores e integradores crear soluciones avanzadas de comunicación sin pagar ninguna licencia.[8]

Asterisk está desarrollado en GNU/Linux para x/86, pero actualmente también se distribuye en versiones para los sistemas operativos BSD, MacOSX, Solaris y Microsoft Windows y está disponible para su descarga gratuita en el sitio oficial.

Asterisk incluye muchas características, anteriormente sólo disponibles en costosos sistemas propietarios PBX como buzón de voz, conferencias, IVR, distribución automática de llamadas, y otros. Los usuarios pueden crear nuevas funcionalidades escribiendo un dialplan en el lenguaje de script de Asterisk o añadiendo módulos escritos en lenguaje C o en cualquier otro lenguaje de programación soportado por Linux.

Para conectar los teléfonos analógicos estándar se utilizan preferentemente adaptadores ATA VOIP( adaptador de teléfono analógico para voz sobre IP), y para conectar directamente a la PSTN se hace necesario tarjetas electrónicas telefónicas FXO fabricadas por Digium u otros proveedores.

Asterisk soporta una amplia gama de protocolos para el manejo y la transmisión de voz sobre interfaces de telefonía tradicional incluyendo H.323, SIP, MGCP, SCCP.

Cabe mencionar que la Comunidad Asterisk es bastante influyente en los temas relacionados con VoIP.

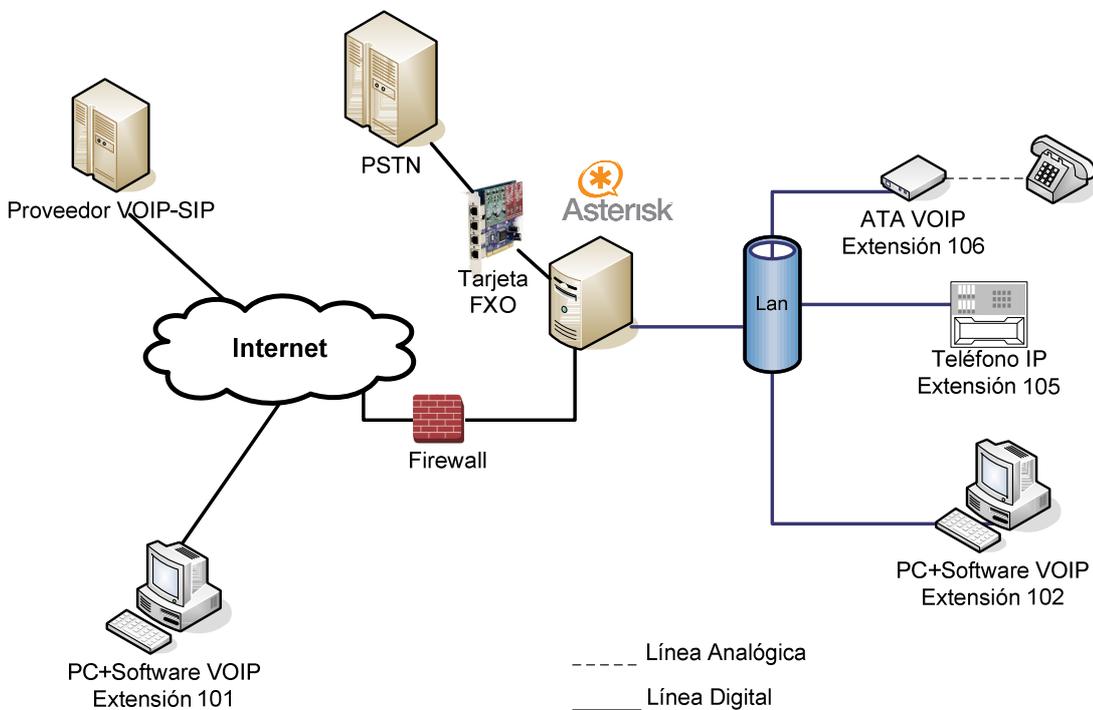


Fig 2.5. Posibles interconexiones con Asterisk.

La figura 3.5. muestra la versatilidad de Asterisk para lograr conectarse a la PSTN por medio de tarjetas FXO, además de un proveedor de VOIP, registrar teléfonos IP, adaptadores y software VOIP.

### **2.5.8. Asterisk y Skype**

La historia de Asterisk y Skype tiene ya unos años, debido a la popularidad de Skype, que desde sus inicios ha gozado de una gran publicidad por el propio público, ha logrado penetrar fuerte en el empresa, ya que éstas se ven beneficiadas por los múltiples servicios que ofrece, los cuales ya fueron mencionados anteriormente. Dado que muchas de estas empresas se han cambiado desde las antiguas centrales telefónicas a las nuevas basadas en VOIP, y donde Asterisk lleva la delantera en términos de instalación y popularidad, existe la necesidad de conectar ésta a Skype, para aprovecharse de los mejor de los dos mundos. Actualmente existen varios hacks que involucran a máquinas virtuales y emuladores de hardware de sonido, con el objetivo de juntar Asterisk y Skype, pero estas soluciones solo permiten las funciones básicas y no son escalables.

Se señala en la descripción de cada uno, que en el caso de los sistemas que emplean SIP sólo podrán comunicarse con dispositivos SIP, y lo mismo se repite en el caso de Skype, además de recalcar que usa un protocolo cerrado. En este caso se desarrolla un channel driver genérico para Asterisk, el cual permite la mayoría de las cosas que los channel drivers de Asterisk ya soportan. No se requiere ninguna programación especial, ni nada fuera de lo común a los otros channels, simplemente instalar, configurar y usar. Además admite nombres de usuario, encriptación, terminales y admite tanto salida y entrada de llamadas a usuarios normales de Skype, como también el uso de los servicios de SkypeOut y SkypeIn.

Para crear usuarios en Asterisk se deben crear primero en el “Busines Control Panel” de Skype, y por ahora no es posible usar las cuentas existentes.

## **2.6. Códecs y formatos contenedores.**

### **2.6.1. Video en Flash.**

#### **Descripción general**

Es posible cargar y reproducir archivos .Mp4, .M4v, .M4a, .Mov y .3gp mediante el mismo NetStream API que se utiliza para cargar archivos FLV. Pero hay algunas cosas que se deben conocer:

- El video debe estar sólo en formato H.264. MPEG-4 Parte 2 (Xvid, DivX etc) no está soportado, video H.263 no está soportado, Sorenson Video no es compatible. Dado que

una gran cantidad de podcast siguen en MPEG-4 Part 2, estos no podrán ser reproducidos.

- Flash Player está muy cerca de un 100% compatibilidad con el estándar H.264.

## Códecs

Códec	Introducido en la versión de Flash Player	Introducido en la versión Flash Lite	Formatos contenedor	Especificación de la norma ISO	Códec ID
Sorenson Spark	6	3	FLV		2
Macromedia Screen Video	6	-	FLV		3
Macromedia ScreenVideo	8	-	FLV		6
On2 TrueMotion VP6-E	8	3	MOV		4
On2 TrueMotion VP6-S	9.0.115.0	-	MP4V, M4V		5
H.264 (MPEG-4 Part 10)	9.0.115.0	-	MP4, F4V, 3GP, 3G2.	ISO 14496-10	

**Tabla 2.4. Códecs de video que están soportados.**

### Flash Video (FLV)

Flash Video es el nombre de un formato de archivo que permite entregar video sobre Internet usando Adobe Flash Player versión 6 o posterior. El contenido de video Flash también puede ser embebido dentro de archivos SWF. Antes de la 3ra actualización de la versión 9 de Adobe Flash Player, Flash Video se refiere a un formato de archivo propietario que tenía la extensión FLV, pero desde ésta, Adobe introdujo nuevas extensiones de archivo y tipos MIME y sugiere el uso de esos en lugar de la antigua extensión FLV[10]:

Extensión de archivo	FTYP	Tipo MIME	Descripción
.f4v	'F4V '	video/mp4	Video de Adobe Flash Player
.f4p	'F4P '	video/mp4	Medios de comunicación protegidos por Adobe Flash Player
.f4a	'F4A '	video/mp4	Audio de Adobe Flash Player
.f4b	'F4B '	video/mp4	Audio Book para Adobe Flash Player
.flv		video/x-flv	Flash Video

**Tabla 2.5. Extensión de archivo Flash video y su descripción.**

Es posible colocar H.264 y AAC streams dentro del tradicional archivo FLV, pero Adobe alienta encarecidamente a todos a adoptar el nuevo formato de archivo estándar. Hay límites funcionales con la estructura FLV cuando se realiza streaming H.264, los cuales no pueden ser superados sin un rediseño del formato de archivo, esta es una de las razones por las que Adobe se está alejando de la tradicional estructura de archivos FLV. Específicamente tratándose de secuencia de cabeceras y Enders es difícil con streams FLV.

### Formato FLV

Un archivo de Flash Video (extensión de archivo FLV) se compone de un encabezado corto y, a continuación, intercalados audio, video y los paquetes de metadatos. Los paquetes de audio y video se almacenan de manera muy similar a las de SWF, y los paquetes de metadatos consisten en datos AMF.

### FLV Cabecera

Campo	Tipo de datos	Ejemplo	Descripción
Signature	byte[3]	"FLV"	Siempre "FLV"
Versión	uint8	"\x01" (1)	Actualmente 1 para conocer los archivos FLV
Flags	uint8 bitmask	"\x05" (5, audio+video) "	Máscara de bits(Bitmask): 4 es de audio, 1 es video
Offset	uint32_be	"\x00\x00\x00\x09" (9)	Tamaño total de la cabecera

**Tabla 2.6. Descripción de los campos de la cabecera FLV.**

## FLV Stream

Campo	Tipo de datos	Ejemplo	Descripción
PreviousTagSize	uint32_be	“\x00\x00\x00\x00” (0)	Siempre 0

**Tabla 2.7. Descripción del campo de FLV Stream.**

## FLV Etiqueta

Campo	Tipo de datos	Ejemplo	Descripción
Type	uint8	“\x12” (0×12, META)	Determina la disposición del cuerpo, véase más abajo los tipos de etiqueta
BodyLength	uint24_be	“\x00\x00\xe0” (224)	Tamaño del cuerpo (el tamaño total de la etiqueta - 11)
Timestamp	uint24_be	“\x00\x00\x00” (0)	Etiqueta de fecha y hora (en milisegundos)
TimestampExtended	uint8	“\x00” (0)	Extensión de tiempo para formar un uint32_be. Este campo tiene la parte superior de 8 bits.
StreamId	uint24_be	“\x00\x00\x00” (0)	Siempre 0
Body	byte[BodyLength]	... ..	Dependen del valor de Type

**Tabla 2.8. Descripción de los campos de la etiqueta FLV**

## Previous tag size (tamaño etiqueta anterior)

Campo	Tipo de datos	Ejemplo	Descripción
PreviousTagSize	uint32_be	“\x00\x00\x00\x00”	Tamaño total de la etiqueta anterior, o “0” para la primera etiqueta

**Tabla 2.9. Descripción del campo de la etiqueta anterior para verificar el tamaño.**

### Tipos de etiquetas FLV ( FLV Tag Types)

Código de Etiqueta	Nombre	Descripción
0x08	AUDIO	Contiene un paquete de audio similar a un SWF SoundStreamBlock.
0x09	VIDEO	Contiene un paquete de video similar a un SWF VideoFrame.
0x12	META	Contiene dos paquetes AMF, el nombre del evento y los datos que van con él .

**Tabla 2.10. Descripción de los campos de los tipos de etiqueta.**

### FLV Etiqueta 0x09: Audio

El primer byte de un paquete de audio contiene bitflags que describen el códec utilizado, con el siguiente esquema:

Name	Expression	Descripción
soundType	(byte & 0x01) » 0	0: mono, 1: stereo
soundSize	(byte & 0x02) » 1	0: 8-bit, 1: 16-bit
soundRate	(byte & 0x0C) » 2	0: 5.5 kHz, 1: 11 kHz, 2: 22 kHz, 3: 44 kHz
soundFormat	(byte & 0xf0) » 4	0: No comprimido, 1: ADPCM, 2: MP3 , 5: Nellymoser 8kHz mono, 6: Nellymoser, 11: Speex

**Tabla 2.11. Información contenida en un paquete sobre el códec de audio empleado.**

El resto de los paquetes de audio son simplemente datos relevantes para el formato, como para un SWF SoundStreamBlock.

## FLV Etiqueta 0x09: VIDEO

El primer byte de un paquete de video describe bitflags contenidos que describen el códec usado, y el tipo de frame.

Nombre	Expresión	Descripción
codecID	(byte & 0x0f) » 0	2: Sorensen H.263, 3: Screen video, 4: On2 VP6, 5: On2 VP6 Alpha, 6: ScreenVideo 2
frameType	(byte & 0xf0) » 4	1: keyframe, 2: inter frame, 3: disposable inter frame

Tabla 2.12. Información contenida en un paquete sobre el códec de video empleado.

En algunos casos también es útil descifrar algunos de los paquetes del video, tales como la adquisición de su resolución (si la etiqueta META onMetaData inicial se pierde, por ejemplo).

### Códec de video Sorensen

El códec Sorensen es un códec de video digital, propiedad de la compañía Sorensen Media. Este es usado por Apple Quicktime y en Adobe Flash, existen dos diferentes códec de video conocidos como “Sorensen códec”: Sorensen Video y Sorensen Spark. Sorensen Spark es utilizado en las versiones posteriores a Macromedia Flash 5 (conocido actualmente por Adobe Flash, luego de ser Macromedia adquirida por Adobe)[11].

Sorensen Media fue fundada en Abril de 2000 para proveer soluciones multimedia de streaming que permitan en cualquier lugar recibir fácilmente un stream de video de calidad en web. Sorensen media es reconocido como un líder en la compresión de video y servicios de streaming.

Sorensen Spark es un códec de video en movimiento incluido en Flash, que permite añadir contenido de video incorporado a Flash. Spark es un codificador/decodificador de video de alta calidad que disminuye enormemente el ancho de banda necesario para publicar imágenes en Flash y, al mismo tiempo, aumenta la calidad de video. Con la inclusión de Spark, Flash dio un importante paso hacia delante en materia de capacidad de video. En Flash 5 o anteriores sólo se podía simular video mediante imágenes secuenciales de mapa de bits.

Sorensen Spark es un códec antiguo pero es también ampliamente utilizado. Se considera como una implementación incompleta de H.263. Actualmente en las últimas versiones de Flash se ha agregado como un futuro reemplazo los códec On2 VP6 y el códec H.264 que requieren equipos con más capacidad de cómputo.

## 2.6.2. Audio en Flash

### Descripción general

- Puede ser audio AAC Main, AAC LC y SBR, que corresponden a tipos de objetos de audio 0, 1 y 2.
- Los tipos de muestreos .mp3 para las pistas de audio con mp3 también están soportados.
- MP3 en MP4 que pretende hacer de múltiples canales de reproducción en mp3 dentro de archivos mp4 no está soportado.
- Las antiguas especificaciones de QuickTime para la integración de datos AAC y MP3 no es compatible.
- Archivos de libros de audio sin encriptar que contienen el capítulo de información. Esta información está expuesta en la llamada onMetaData como un array de objetos con el nombre 'chapters' (capítulos).
- Flash Player puede reproducir canales múltiples de archivos AAC, aunque el sonido se mezcla a dos canales y se remuestrea a 44.1KHz. Se destaca que la reproducción multicanal está marcada para una de las próximas revisiones importantes de Flash Player. Esto requiere el rediseño completo del motor del sonido de Flash Player que data de alrededor de 1996 y no ha mejorado desde entonces.
- Todos los tipos de muestreo desde 8kHz a 96kHz están soportados.

### Códecs

Códec	Introducido en la versión de Flash Player	Formato contenedor	Códec ID
MP3	6	MP3	2
Nellymoser ASAO Codec (speech compression) audio content	6	FLV	5, 6
Raw PCM sampled audio content	6	WAV	0
ADPCM (Adaptive Delta Pulse Code Modulation) audio content	6		1
AAC ( HE-AAC /AAC SBR , AAC Main Profile, and AAC-LC)	9.0.115.0	M4A, MP4	
Speex	10	FLV	11

**Tabla 2.13. Códecs de audio que están soportados.**

## Metadatos

Los metadatos ayudan a ubicar datos importantes de los paquetes enviados.

Propiedad	Valor	Notas
duration	Evidente	A diferencia de los archivos FLV este ámbito siempre estará presente.
videocodecid	Para H.264 se informa como 'avc1'.	
audiocodecid	Para AAC se informa como "mp4a", para MP3 se informa como "Mp3".	
avcprofile	66, 77, 88, 100, 110, 122 o 144	Corresponde a los perfiles de H.264
avclevel	Un número entre 10 y 51.	
Aotype	Cualquiera 0, 1 or 2	Esto corresponde a AAC Main, AAC LC y SBR tipos de audio.
moovposition	Entero	
trackinfo	Arreglo	Una serie de objetos que contiene diversas informaciones acerca de todas las pistas en un archivo.
chapters	Arreglo	Información sobre los capítulos de los audio libros.
seekpoints	Arreglo	Veces que se puede introducir directamente <code>NetStream.seek ()</code> ;
videoframerate	Entero	La velocidad de fotogramas de video, si una tasa de fotogramas monótona es usada. La mayoría de los videos tendrán una velocidad de cuadro monótono.
audiosamplerate		La frecuencia original de muestreo de la pista de audio.
audiochannels		El número original de los canales de la pista de audio.
tags		Información de etiqueta ID3

**Tabla 2.14. Datos importantes contenidos en un paquete de Metadata.**

Se detalla a continuación el códec Asao Nellymoser, el cual es empleado para codificar la voz que es capturada por el micrófono del usuario.

## **Códec Asao Nellymoser**

Nellymoser Asao es un códec de voz propietario con un formato de canal simple (mono) optimizado para transmisiones de bajo bitrate, desarrollado por Nellymoser Inc. Los datos de sonido se agrupan en frames de 256 muestras. Cada frame se convierte en el dominio de la frecuencia y la más significativas (de mayor amplitud) frecuencias son identificadas. Una serie de bandas de frecuencias son seleccionadas para la codificación, y el resto se descartan. El flujo de bits para cada frame a continuación se codifica en las bandas de frecuencia que están en uso y con su respectiva amplitud. Este códec no tiene en cuenta la tasa de muestreo real, y ha fijado relación entre la cantidad de muestras de entrada y salida del tamaño del paquete (2 bits por cada entrada de la muestra). [12]

Un completo códec codificador/descodificador ha sido liberado, por el proyecto FFmpeg.

### **Uso en la tecnología Flash**

El códec está licenciado para su uso en la tecnología Flash de Macromedia/Adobe. El códec Asao Nellymoser es una parte integral del plugin de Flash desde la versión 6. El códec está optimizado para su uso en tiempo real y con baja latencia en la codificación de audio. En Adobe Flash Player, cuando se graba audio desde el micrófono del usuario, se utiliza el códec Asao Nellymoser y no se permite que los programadores de Flash puedan seleccionar cualquier otro códec. La tasa de muestreo de captura de audio puede ser controlada por el programador de Flash para aumentar y disminuir la tasa de bits de codificación y de calidad. La codificación se realiza en la máquina cliente, y los datos comprimidos son enviados a través del protocolo de Adobe RTMP a un servidor RTMP (Flash Media Server, Red5, Wowza).

### **2.6.3. Audio en telefonía IP**

La comunicación de voz es analógica, mientras que la red de datos es digital. El proceso de convertir ondas analógicas a información digital se hace con un codificador-decodificador (el CODEC). Hay muchas maneras de transformar una señal de voz analógica, todas ellas gobernadas por varios estándares. El proceso de la conversión es complejo. Es suficiente decir que la mayoría de las conversiones se basan en la modulación codificada mediante pulsos (PCM) o variaciones. [13]

Además de la ejecución de la conversión de analógico a digital, el CODEC comprime la secuencia de datos, y proporciona la cancelación del eco. La compresión de la forma de onda representada puede permitir el ahorro del ancho de banda. Esto es especialmente interesante en los enlaces de poca capacidad y permite tener un mayor número de conexiones de VoIP simultáneamente. Otra manera de ahorrar ancho de banda es el uso de la supresión del silencio, que es el proceso de no enviar los paquetes de la voz entre silencios en conversaciones humanas.

A continuación se analizan dos códecs, los cuales son permitidos en la comunicación con skype por medio del “chan\_skype” desarrollado para Asterisk. Posteriormente se presenta una tabla comparativa entre estos.

### **Códec de voz G.729**

G.729 es un estándar de la ITU-T para la compresión de audio. Este algoritmo de compresión de datos de audio para voz, comprime audio de voz en trozos de 10 milisegundos. Los tonos DTMF, transmisiones de fax, y audio de alta calidad no pueden ser transportados confiablemente con este códec. DTMF requiere el uso de RTP Payload para dígitos DTMF.

G.729 se usa mayoritariamente en aplicaciones de Voz sobre IP (VoIP) por sus bajos requerimientos en ancho de banda. El estándar G.729 opera a una tasa de bits de 8 kbit/s, pero existen extensiones, las cuales suministran también tasas de 6.4 kbit/s y de 11.8 kbit/s para peor o mejor calidad en la conversación respectivamente. También es muy común G.729a el cual es compatible con G.729, pero requiere menos cómputo. Esta menor complejidad afecta en que la calidad de la conversación es empeorada marginalmente.

G.729 se ha ampliado en el Anexo B (G.729B), que proporciona un método de compresión del silencio que permite por medio de un módulo la detección de actividad de voz (VAD). Se utiliza para detectar la actividad de voz en la señal. También incluye un módulo de transmisión discontinua (en sus siglas en inglés, DTX) que decide sobre la actualización de los parámetros de ruido de fondo para no voz (entornos ruidosos). Se utilizan 2-byte para las tramas del Descriptor de Inserción del silencio (en sus siglas en inglés, SID) que son transmitidas para iniciar la generación de ruido confortable (en sus siglas en inglés, CNG). Si la transmisión se detiene, y el vínculo queda en silencio porque no hay voz, el lado receptor puede asumir que el vínculo ha sido cortado. Pero mediante la inserción de ruido de confort, un silbido analógico se simula digitalmente en caso de silencio para asegurar que el receptor crea que el enlace está activo y en funcionamiento.

A continuación se muestra una tabla resumen de las variantes del códec :

- El Bit Rate indica la cantidad de información que se manda por segundo.
- El Sampling Rate indica la frecuencia de muestreo de la señal vocal(cada cuanto se toma una muestra de la señal analógica).
- El Frame size indica cada cuantos milisegundos se envía un paquete con la información sonora.
- El MOS indica la calidad general del códec (valor de 1 a 5).

## Comparativas entre anexos

	<b>G.729</b>	<b>G.729A</b>	<b>G.729D</b>	<b>G.729E</b>
<b>Bitrate</b>	8 kb/s	8 kb/s	6.4 kb/s	11.8 kb/s
<b>Tipo</b>	CS-ACELP	CS-ACELP	CS-ACELP	CS-ACELP
<b>Delay</b>				
Frame size	10ms	10ms	10ms	10ms
Lookahead	5ms	5ms	5ms	5ms
Quality	toll	toll	toll	toll
<b>Complejidad</b>				
MIPS	20-25	10	Menos que G.729	~30
RAM	<4K	<4K	Menos que G.729	~4K

**Tabla 2.15. Comparativas entre Anexos del códec G.729.**

### Atributos

- La tasa de bits más baja (8 kb/s) estándar ITU-T.
- Uno de los estándares más testeados para todas las aplicaciones, incluyendo comunicaciones inalámbricas.
- Exactitud que asegura que, independientemente de la aplicación, el mismo nivel de calidad se mantendrá.
- La calidad se ve menos afectada por errores en la transmisión en relación con otros códec de voz de baja tasa de bits.
- Un códec principal de la ITU-T, en continuo desarrollo, como lo demuestran sus anexos:

Anexo A: DSVD (Digital Simultaneous Voice and Data).

Anexo B: VAD / CNG / DTX (Voice Activity Detection / Comfort NoiseGenerator / Discontinuous Transmission).

Anexo C: Versión de punto flotante de G.729 y sus anexos.

Anexo D: Extensión a 6,4 kb/s.

Anexo E: Extensión a 11,8 kb/s.

Anexo A ofrece la mejor la complejidad/calidad de la industria.

- Es uno de los códec de voz , y el único con baja tasa de bits( bitrate) recomendado por el Foro “Frame Relay Forum FRF.11”. Además de ser un potencial códec de bajo bitrate candidato para Packet Cable. También se esta considerando en otros foros.

## Códec de voz G.711

G.711 es un estándar de la ITU-T para la compresión de audio. Este estándar es usado principalmente en telefonía, y fue liberado para su uso en el año 1972.

G.711 es un estándar para representar señales de audio con frecuencias de la voz humana, mediante muestras comprimidas de una señal de audio.

El códec de voz G.711, también conocido como modulación por código de pulsos (en inglés, PCM, Pulse Code Modulation), es un códec de forma de onda muy utilizado. Utiliza una frecuencia de muestreo de 8.000 muestras por segundo, una cuantización no uniforme de 8 bits, se utilizan para representar cada muestra, dando lugar a un bitrate de 64 kbit/s. G.711 utiliza un sistema de mapeo logarítmico que enfatiza las partes de la señal a las que el oído humano es más sensible.

Existen dos versiones ligeramente diferentes;  $\mu$ -law, que se utiliza principalmente en América del Norte, y A-law, que está en uso en muchos otros países fuera de Norteamérica. G.711  $\mu$ -law tiende a dar más resolución a las señales de gama más alta, mientras que G.711 A-law establece los niveles más bajos de cuantificación de los niveles de señal. A-Law es utilizado con conexiones E1 (Europa y Australia), y  $\mu$ -law es utilizado con conexiones T1 y J1.

El estándar también define un código para secuencia de repetición de valores, el cual define el nivel de potencia de 0 dB.

### Características

- Frecuencia de muestreo de 8 kHz.
- 64 kbit/s bitrate (8 kHz de frecuencia de muestreo x 8 bits por muestra).
- G.711 es un codificador de voz de forma de onda.
- G.711 Apéndice I se define un algoritmo de ocultamiento de pérdida de paquetes (en inglés, Packet Loss Concealment) para ayudar a ocultar las pérdidas de transmisión en una red paquetizada.
- G.711 Apéndice II define un algoritmo de transmisión discontinua (DTX) que utiliza la detección de actividad de voz (en inglés, Voice Activity Detection AD) y generación de ruido de confort (en inglés, Comfort Noise Generation CNG) para reducir el uso de ancho de banda durante los períodos de silencio.
- Pruebas de PSQM bajo condiciones ideales el MOS es de 4,45 para G.711  $\mu$ -law, 4,45 para G.711 A-law.

- Pruebas de PSQM bajo un red semi-congestionada el MOS es de 4,13 para G.711  $\mu$ -law, 4.11 para G.711 A-law.

A continuación se presenta una tabla resumen de los dos códec anteriormente analizados.

Nombre	Estándar	Descripción	Bit rate (kb/s)	Sampling rate (kHz)	Frame size (ms)	Observaciones	MOS (Mean Opinion Score)
G.711	ITU-T	Pulse code modulation (PCM)	64	8	Muestreada	Tiene dos versiones $\mu$ -law (US, Japan) y A-law (Europa) para muestrear la señal	4.1
G.729	ITU-T	Codificación de voz a 8kbit/s usando CS-ACELP	8	8	10	Bajo retardo (15 ms)	3.92

Tabla 2.16. Comparativas códecs G.711 y G.729

En resumen, el estándar G.729 ITU-T es una óptima opción, en vista de las necesidades de comunicación de voz de calidad, de forma comprimida, con un bajo requisito de ancho de banda y de retraso razonable. El éxito de este códec puede ser atribuido, en parte, al tremendo esfuerzo presentado por los dueños de derechos de propiedad intelectual y Sipro Lab Telecom a fin de facilitar el acceso a la concesión de licencias a un precio razonable.

## 2.7. Funcionalidades adicionales de Flash.

### 2.7.1. Objetos compartidos.

Los objetos compartidos (Shared Object) son una característica de Adobe Flash, y se han comparado con "super cookies", porque permiten almacenar datos de forma local o remota y, en este último caso, compartir información con otros usuarios.

Los objetos compartidos guardan la información en slots, cada uno de los cuales puede ser de un tipo de datos determinado y conteniendo información diferente en cada uno de ellos. Cuando son usados de forma remota, el cambio de información en un slot causa que el servidor FMS o su símil Red5, sincronice toda la información del slot. El servidor no sincronizara los otros slots, a menos que se modifiquen.

Cuando uno se conecta por primera vez a un objeto compartido remoto se recibe un evento "onSync" con una lista de toda la información en el objeto compartido. El evento "onSync" también informa de los cambios exitosos que se realicen en el objeto compartido, o cuando un error o conflicto ocurre.

Hay dos tipos básicos de objetos compartidos: locales o remotos. En este caso sólo se empleará el uso de objetos compartidos remotos. Sin embargo, se puede emplear el uso de objetos compartidos locales con la finalidad de almacenar información acerca de preferencias de los usuarios (nick, password, etc.). Cabe destacar que para emplear objetos compartidos remotos se debe tener instalado FMS(Flash Media Server) o un símil como Red5, y en el caso que sean locales no se requiere.

También se puede configurar un objeto compartido como persistente, lo cual significa que el objeto compartido puede permanecer incluso cuando la solicitud no lo está utilizando, esto es útil para guardar el estado de los objetos compartidos. En el caso de una pizarra compartida, si se cortara la conexión de forma inesperada en medio de la creación de un diagrama de flujo y si la solicitud no persistió el objeto compartido, entonces se perdería toda la información que se había almacenado.

Debido a que la pizarra compartida persiste, el objeto compartido almacena toda la información de dibujo en ella y uno o cualquier otra persona puede salir y volver a una instancia de la aplicación de pizarra, a su regreso, la aplicación automáticamente almacenará y recuperará toda la información de dibujo. Se pueden fijar los objetos compartidos a persistir a nivel local, de manera que, incluso cuando se desconecta, una versión local accesible de la información, esté disponible para uno. Luego se puede actualizar y sincronizar la información la próxima vez que se conecte al servidor.

## **2.8. Tecnologías de Medición en redes.**

Dentro de las tecnologías de medición en redes, destaca Wireshark, dado que es el analizador de protocolo de red más usado en el mundo, y es un estándar de hecho por su gran utilización en muchas industrias e instituciones educativas a nivel mundial. Permite inspeccionar cientos de protocolos, además de filtrar datos y correr en diversos sistemas operativos.[14]

Wireshark es un software OpenSource, el cual posee versiones ejecutables para la mayoría de los sistemas operativos Unix y compatibles, Linux, Solaris, FreeBSD, NetBSD, OpenBSD, y Mac OS X, así como en Microsoft Windows.

Permite examinar datos de una red viva o de un archivo de captura guardado en disco, permitiendo analizar la información capturada, a través de los detalles y sumarios por cada paquete. Además Wireshark incluye un completo lenguaje para filtrar lo que se desea ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

Es utilizado para realizar análisis y solucionar problemas en redes de comunicaciones para desarrollo de software y protocolos, siendo usado además, como una herramienta didáctica para la educación.

### 3. CAPITULO III: METODOLOGÍA.

En este capítulo se describe la metodología general utilizada a lo largo de este trabajo, y los pasos que se llevaron a cabo para cumplir con los objetivos planteados. Se comienza con una descripción a nivel general para luego profundizar cada punto con mayor detalle.

Se comienza con una recopilación de los antecedentes de las diversas tecnologías que cumplieran con los objetivos planteados, con una investigación referente a los protocolos de comunicación a utilizar, junto con el estudio de las herramientas de desarrollo, observando el posible potencial y adaptabilidad a los objetivos propuestos en la memoria. Luego de escoger las tecnologías, se analizan los elementos necesarios para el funcionamiento de la plataforma y con esto se procede al diseño de la plataforma en base a los servicios que ha de tener según lo propuesto para posteriormente implementar y configurar un laboratorio de pruebas, con el objetivo de poner en funcionamiento los módulos de ejemplo en condiciones reales.

#### 3.1. Diseño solución.

El diseño a realizar consiste en una plataforma que permita a un tutor poder transmitir sus conocimientos a los oyentes, integrando audio y video, presentaciones sincronizadas y un módulo de consulta. La idea de diseño debe basarse en una plataforma que aproveche los avances en el software de código libre, con el objeto de ser utilizado con fines promocionales de las potencialidades de éstos, en el desarrollo de sistemas e-learning de bajo costo y de rápida implementación. Por lo tanto, en la implementación se desarrollarán ejemplos de acuerdo a los servicios que debe poseer. El tutor se conectará a una plataforma web por medio de un browser estándar con Adobe Flash Player, el cual se encuentra instalado en casi el 98% de los computadores conectados a internet.

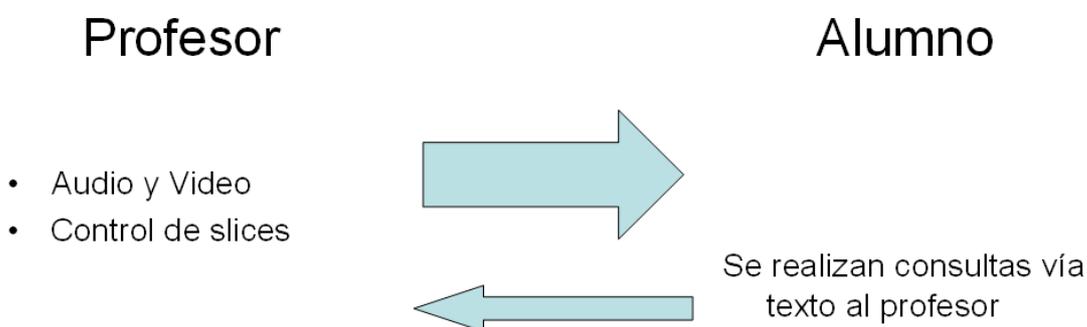


Fig. 3.1. Diseño funcional de la solución.

A continuación se exponen de manera específica y con detalle cada uno de los servicios e interfaces que requiere contener la plataforma.

### **3.1.1. Audio y video.**

Dentro de la variedad de sistemas para la transmisión y recepción de audio-video en tiempo real basado en web, donde todos requieren la instalación de un plug-in, en los navegadores web, se puede destacar a Windows Media Player ( WMP), Adobe Flash Player, Real Player, VideoLan, entre otros. Y dentro de éstos el que lleva sin duda la delantera en términos de instalación es Adobe Flash Player con una penetración del 98% en computadoras conectadas a Internet, pero el principal impedimento que se tiene a la hora de usar esta tecnología es el alto costo del servidor de streaming Flash Media Server de Adobe, por esta razón se hace necesario el uso de un símil de este servidor por el lado del open source. Este servidor llamado RED5, permitirá transmitir de modo broadcast audio-video y sincronizar eventos. El protocolo utilizado por Flash es RTMP (Real Time Media Protocol), el cual es propiedad de Adobe.

Para implementar la plataforma que soporte el audio y el video, se requiere establecer la plataforma base de desarrollo y los protocolos involucrados, como también los códecs a usar. La plataforma descrita en el capítulo anterior, Openlaszlo, se escoge por sobre Flex de Adobe, ya que requiere una licencia de pago para poder ser utilizado y la primera es open source. Openlaszlo permite crear, como resultado de la compilación de código XML, una película Flash o código DHTML. Para utilizar el audio y el video se usa la compilación en Flash, debido a que no es posible en DHTML. Debido a esto, se debe usar el protocolo RTMP (Real Time Media Protocol), el cual es propietario de Adobe, siendo este protocolo explicado a cabalidad en el capítulo anterior.

Luego de lo anteriormente expuesto es necesario determinar la arquitectura a utilizar ya que Openlaszlo y Red5 se encuentran enteramente programados en Java, y para que trabajen necesitan estar en un contenedor Java Servlet. En este caso, se analizan dos alternativas Glassfish de Sun y Tomcat, los dos tienen la característica de código libre o software libre. En el primero resultó fácil la instalación del archivo .war de OpenLaszlo, dado que la información de las dependencias que se requieren desde Apache se encuentran en la página oficial de Glassfish, pero de Red5 no se encuentra información clara de cómo portarlo a Glassfish, es por esto que se decide trabajar con Tomcat, ya que es ampliamente referido en los manuales de instalación que se encuentran en la página oficial de Red5.

Para facilitar la implementación se utilizan los archivos .tar que contienen a Tomcat ya instalado. Luego de esto es necesario descomprimir y copiar en una carpeta para posteriormente poner en marcha el servidor.

A continuación se detalla a través de un esquema la arquitectura que sustentará algunos servicios.

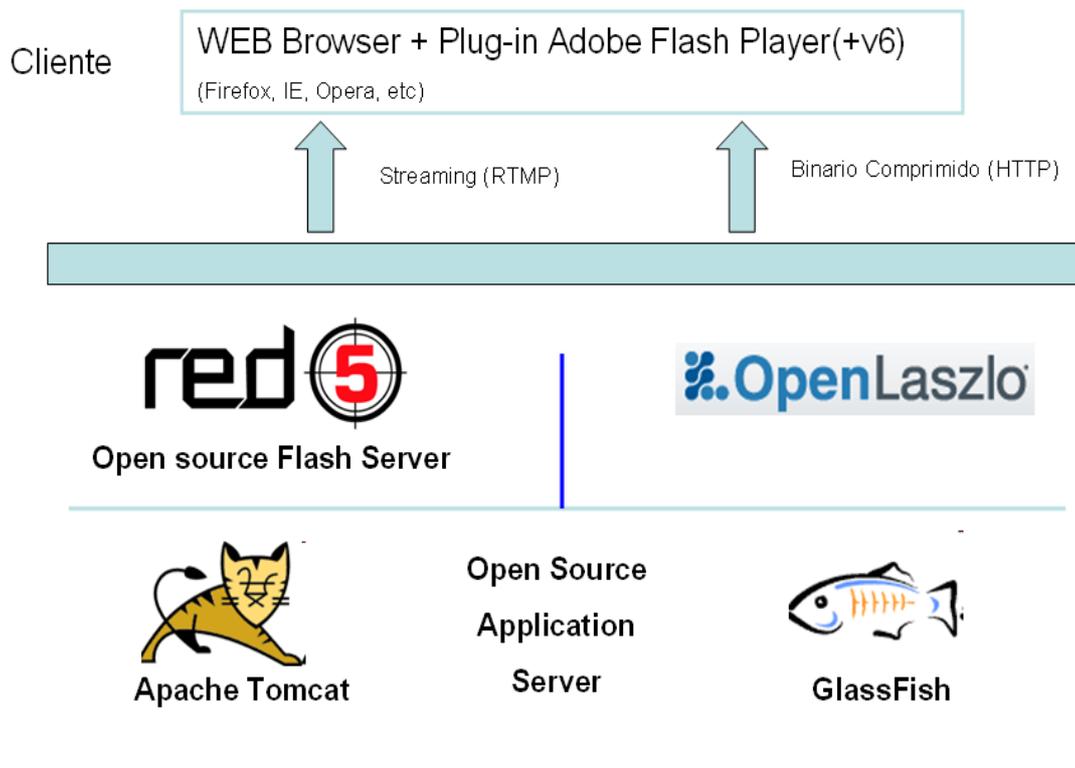


Fig. 3.2. Arquitectura que sustentará algunos servicios de la academia virtual.

Para continuar el proceso, es necesario implementar una interfaz para cada uno de los agentes, usuarios y administradores. Como ya se comentó, la interfaz debe estar basada en web, en la cual, un tutor pueda transmitir el video y audio de él, además de manejar la sesión sincronizando las diapositivas de la presentación. Por otro lado, el usuario debe poder realizar las consultas propias por medio de un chat básico.

### 3.1.2. Presentaciones sincronizadas.

Las presentaciones en línea tienen como fin lograr que el profesor tenga un medio donde respaldar la información que está explicando, por lo tanto, se hace obvio que aquel debe poseer la capacidad de administrar las presentaciones.

Para poder desplegar las presentaciones, se requiere que los documentos sean traspasados a un documento manejable por Flash, ya que nativamente no es posible desplegar documentos MSOffice. Debido a esto es necesario realizar una conversión de documentos, lo cual se detalla en el apartado siguiente.

### **3.1.3. Comunicación vía texto.**

La comunicación de texto se realizará con el objetivo de que la audiencia concretice las preguntas de forma on-line al profesor, pero sin interrumpir.

### **3.1.4. Interfaz web integrada.**

La interfaz web contendrá la presentación en línea, el sistema de consulta vía texto, el video y audio embebido en la página. El objetivo es presentar un ejemplo, donde en una sola página se integren todos los elementos necesarios para lograr una óptima clase por parte del tutor.

## **3.2. Implementación solución.**

### **3.2.1. Comunicación audio y video.**

Para la transmisión de audio y video se utiliza el servidor de streaming, el cual se encargará de enviar en modo broadcast a los clientes que se suscriban a la dirección rtmp que se esté usando para este caso.

Para lograr dicho objetivo fue necesario revisar la documentación de Openlaszlo, además de los ejemplos que se encuentran disponibles en el trunk de éste. En el caso de Red5, basta con crear una sala y conectarse a ella para que éste transmita el audio y video.

En el caso de la interfaz del tutor, ésta transmitirá el audio y video desde su cámara web. Para llevar esto a cabo, en la aplicación LZX, se utilizará el tag <camera> con el objeto de conectar la cámara al servidor RTMP, aunque para proteger la privacidad, el funcionamiento de la cámara debe ser explícitamente permitido por el usuario por medio de una ventana de consulta.

En el caso de la interfaz del alumno, éste se suscribirá a la transmisión, y para realizar esto en la aplicación LZX, se utilizará el tag <videoview>, el que permitirá reproducir el video desde servidores RTMP o HTTP.

### **3.2.2. Presentaciones sincronizadas.**

Para la sincronización de eventos se estudian dos opciones, una de ellas es el uso de los objetos compartidos (Shared Object, en inglés) de Flash, el cual está implementado en el servidor de streaming Red5, y obviamente en FMS (Flash Media Server). La otra posibilidad es el uso de Ajax, la cual es una tecnología asíncrona, debido a que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir en la visualización ni en el comportamiento de la página. Las funciones de llamadas son realizadas por medio del lenguaje interpretado Javascript, y el acceso a los datos se realiza mediante XMLHttpRequest, un objeto que se encuentra disponible en todos los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Con el fin de promover otras opciones a las ampliamente conocidas por los desarrolladores se escoge en este diseño el uso de los objetos compartidos. Su elección se debe a que permitirá dar a conocer las bondades del servidor Red5 y las adicionales que se deben agregar en el código de Openlaszlo para llevar a cabo la conexión con este servidor. De esta forma, se crea un slot donde se almacena el número de la página, siendo la interfaz del tutor la que modificará este valor. Este slot será sincronizado por el servidor hacia la misma interfaz del tutor y también hacia la interfaz de los alumnos, permitiendo tener un control total de la sesión por parte del tutor.

Debido a que Openlaszlo sólo soporta los objetos compartidos locales, y no implementa los objetos compartidos remotos, se hace necesario adicionar una librería que lo implemente.

### **3.2.3. Conversión de documentos.**

Para realizar la conversión de múltiples formatos se emplean tres conversores. El primero es Jodconverter, que emplea como motor de conversión a OpenOffice; de esta forma se logra convertir una variedad de formatos al tipo pdf. Después de esto, es posible convertir el documento en formato pdf a una película swf, utilizando SWFtools, el que crea un frame por cada hoja del archivo que permite de forma nativa reproducirla en Flash Player. Además, utilizando el conversor ImageMagick, es posible crear una imagen (en este caso se escogió el formato png) por cada página del documento pdf. Esto puede ser útil si es que en el navegador no es posible reproducir Flash.

Ya que es requisito tener instalado el plug-in Flash Player, se escoge para el despliegue de la presentación en el navegador web el uso de la película swf, porque preserva una excelente calidad y permite una rápida implementación, pues en Openlaszlo se tienen comandos específicos para controlarla.

Finalmente la plataforma de pruebas de los módulos anteriormente descritos se detalla en el siguiente diagrama.

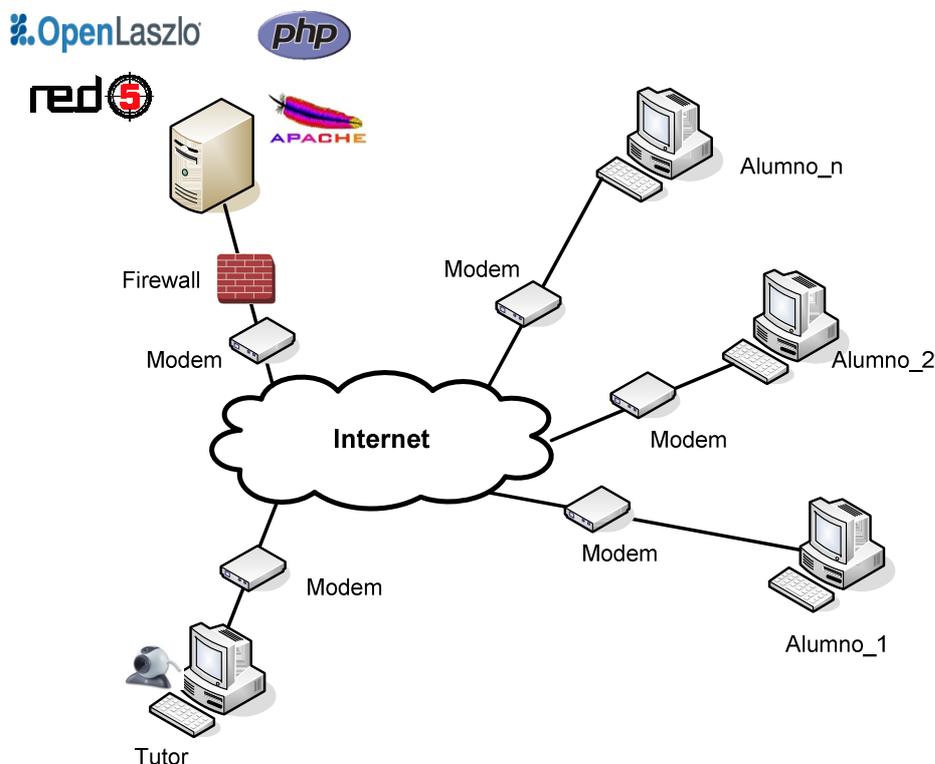


Fig. 3.3. Diagrama de la configuración de pruebas audio-video y sincronización de eventos.

### 3.2.4. Telefonía-audio.

El profesor guía propone la puesta en marcha de la herramienta de audio-video conferencia skype, debido a que recientemente fue liberado "chan\_skype" en su versión beta, el cual corresponde a un canal de comunicación entre Asterisk y Skype. De esta forma es posible conectar un equipo o software de voz por IP (VOIP) a la central telefónica Asterisk, y rutear esta comunicación a un usuario Skype y viceversa, según la configuración en la central. Esto permite integrar dos tecnologías, y acceder a un potencial de más de 400 millones de usuarios que usan Skype.

Para llevar a cabo esta propuesta, se requirió de manuales de configuración de Asterisk los cuales están disponibles en su gran mayoría en Internet. Además el módulo de Skype para Asterisk trae un Readme explicativo de las configuraciones que se pueden realizar en "chan\_skype.conf", como también acceso a los foros habilitados para el testing de este módulo.

La plataforma a configurar se detalla en el siguiente diagrama.

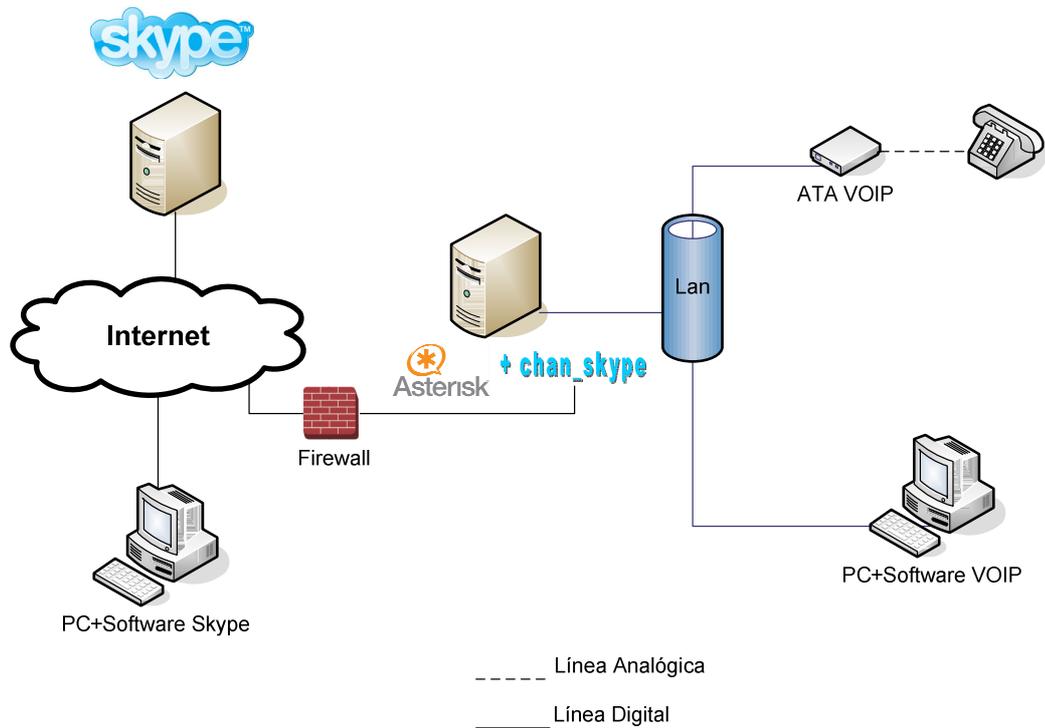


Fig. 3.4. Diagrama de la configuración de prueba módulo telefonía-audio.

En este caso en una conferencia de audio los alumnos y el tutor podrán estar en skype y en Asterisk.

### Audio-Conferencia

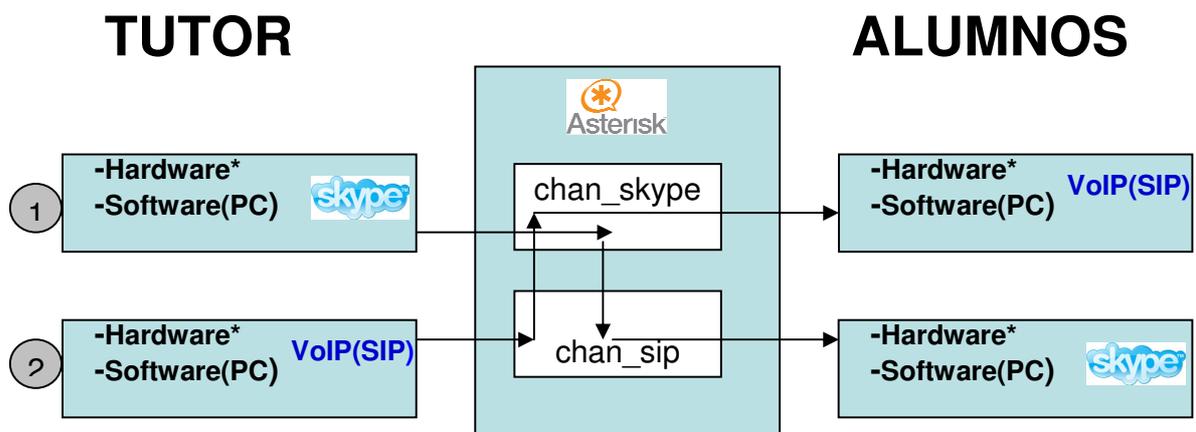


Fig. 3.5. Diagrama de cómo se inserta Asterisk+Skype para una audio-conferencia.

En la fig. 3.5. se puede ver que el tutor tiene dos opciones en la interacción Asterisk+Skype como un sistema de audio-conferencia asociada a una clase. Una de ellas es conectarse por medio de una cuenta de Skype, para luego agregar a un usuario administrado por Asterisk. Con esto se logra la comunicación con Asterisk en donde se habilitan cuentas SIP para cada uno de los alumnos, por lo tanto estos usarán un software o hardware VoIP que soporte el protocolo SIP. La otra opción es que el tutor se conecte por medio de VoIP a Asterisk y luego por medio del “chan\_skype” salir como un usuario de skype, con esto los alumnos usando sus cuentas de skype agregan al usuario administrado por Asterisk.

### **3.3. Requerimientos de software.**

#### **Ciente:**

Debe tener un browser, (por ejemplo Firefox, Opera, Internet Explorer) que tenga instalado el último Adobe Flash Player soportado por Openlaszlo.

#### **Servidor:**

Sistema operativo basado en Linux (Debian, Ubuntu, etc).  
Apache Tomcat.  
Servidor de streaming RED5.

### **3.4. Requerimientos de hardware.**

En el caso del cliente los requerimientos son bastante básicos y, hoy en día cualquier computador cumpliría con las exigencias, sin olvidarse que el computador que ocupará el profesor debe poseer cámara web.

En el caso del servidor, los requerimientos del hardware necesario para el correcto funcionamiento corresponden a la capacidad de procesamiento y memoria, debido a la carga que se produce al realizar variados streaming simultáneos. Considerando que la orientación del trabajo de título es de carácter docente y económico, los equipos comerciales estándar disponibles en el mercado son adecuados.

Con lo expuesto anteriormente, se concluye que el uso de uno de los computadores del Laboratorio de Telecomunicaciones, ubicado en el Edificio de Electrotecnologías de la Universidad de Chile, se adecúa a los requerimientos previamente comentados.

Las características del computador DELL se indican en la tabla 3.1.

Procesador	Pentium 4. 1.5 [GHz]
Memoria	512 [MB] PC133
Disco Duro	20 [GB]
Obs	2 Tarj. PCI Ethernet 10/100 (1 IP Red del Lab y 1 IP Pública)

**Tabla 3.1. Características del servidor.**

## 4. CAPITULO IV: RESULTADOS.

En este capítulo se describen los resultados obtenidos durante el trabajo de memoria de título. En los resultados se analizan las interfaces Web desarrolladas y la estructura funcional e interna de la plataforma ejemplo y también los resultados obtenidos de la puesta en marcha del módulo de Skype para Asterisk.

### 4.1. Interfaces.

Las distintas interfaces encontradas en esta sección corresponden a interfaces Web que han sido el fruto de la fusión de código propio desarrollado para este trabajo de título y, en su menor medida, de ejemplos obtenidos en internet.

Se hace hincapié en que el objetivo de la plataforma implementada es sólo de difusión de los distintos servicios enfocados a una academia virtual, que pueden ser desarrollados con tecnologías de código abierto y software libre.

#### 4.1.1. Interfaz web audio y video

Esta interfaz de conexión con el servidor Red5, fue llevada a cabo con base en los ejemplos que están disponibles en el trunk de Openlaszlo.

Con ciertas modificaciones se agrega una barra de texto, en la cual se puede ingresar la dirección IP del servidor Red5. Cabe señalar que para acceder a los recursos del usuario audio y video se solicita por medio de una ventana de consulta, como se puede apreciar en la fig. 4.1. Si la persona permite el acceso, el atributo del tag <camera> y <microphone>, se establecerá en “true”, teniendo en cuenta que no es posible dar permiso en un dispositivo y no en otro. Es decir, se otorga acceso a todas las cámaras y micrófonos, o a ninguno de ellos. Es un todo o nada.



Fig. 4.1. Ventana de consulta de acceso a cámara y micrófono.

Es posible cambiar los permisos de una aplicación que se está ejecutando mediante el botón derecho del mouse, luego, en configuración, es posible modificar los permisos de acceso como también los dispositivos a emplear, teniendo en cuenta que el menú puede mostrar el nombre de los controladores de dispositivo y no los dispositivos reales.

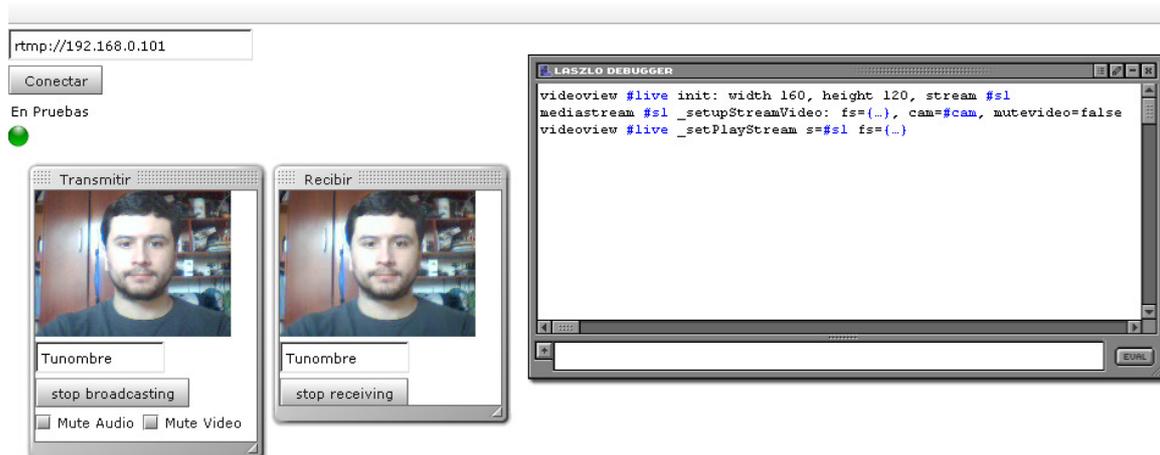


Fig. 4.2. Interfaz ejemplo de audio y video en web.

En la página ejemplo se tiene dos ventanas, la del lado izquierdo permite transmitir y la de la derecha, recibir el streaming. Para esto, se debe tener un mismo nombre, como se ve en la figura anterior como “Tunombre”. También se muestra una ventana del debugger de Openlaszlo.

Cabe destacar que para realizar el streaming se está usando el formato contenedor FLV y el códec Sorenson Spark.

Los códigos completos se encuentran en la sección de anexos, los cuales están debidamente comentados.

#### 4.1.2. Interfaz web del convertor de documentos.

Esta interfaz fue desarrollada en php modificando un ejemplo de código de upload típico, el cual, por medio del formulario permite subir un archivo al servidor.

Para integrar la conversión de documentos, luego de haber instalado correctamente los tres convertidores en el servidor, es posible invocarlos por medio de la funcionalidad de php para llamar comandos en el servidor por medio de la función `exec()`.

A continuación, en la fig. 4.3. se muestra un esquema del código php desarrollado y en la sección de anexos se encuentra el código completo.

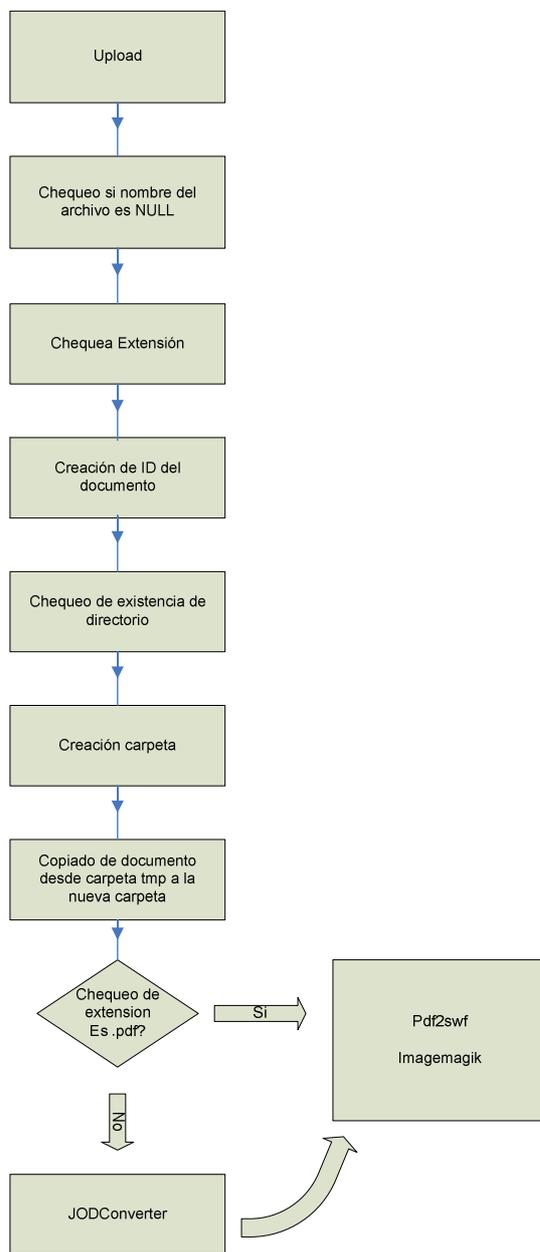


Fig. 4.3. Esquema del código php del conversor de documentos.

Del esquema se puede extraer que al momento de subir un archivo se realiza un chequeo de extensión o tipo de documento, esto permite advertir al usuario que el documento a convertir no está soportado. Si es que el documento está soportado, antes de ser convertido, se crea una

carpeta utilizando como nombre un identificador más el nombre del documento con el objeto de evitar sobrescribir carpetas, ya que puede darse el caso que se intente convertir un documento que posea igual nombre que uno anteriormente convertido. Este identificador es obtenido por medio del cálculo de un hash md5 sobre una función de php que retorna un valor según la hora actual en microsegundos. Luego de esto, se ejecuta un chequeo de extensión con el objeto de saber si el documento está o no en formato pdf; si no lo está, se usa el conversor Jodconverter, para luego tomar el documento en pdf y convertirlo en un arreglo de imágenes y una película swf. Todos estos documentos son almacenados en la carpeta anteriormente creada.

En la siguiente figura se presenta una captura de pantalla de la interfaz del conversor.



Fig. 4.4. Página del conversor de documentos.

Como se puede ver en la fig. 4.4., el conversor posee una interfaz muy simple, en la cual se presenta un resumen de los archivos que han sido convertidos.

Los formatos que son posibles convertir a formato pdf son: .doc, .docx, .xls, .xlsx, .ppt, .pptx, .odt, .ods, .odp, .txt, .rtf.

En resumen, el conversor permite el manejo de una gran variedad de formatos, dando gran flexibilidad en el despliegue de la presentación.

### 4.1.3. Interfaz web presentación

Para desarrollar esta interfaz, se revisa la documentación de Openlaszlo en lo que concierne al manejo de archivos .swf. De esta forma, se logra crear botones que permiten el control de la presentación.



Fig. 4.5. Interfaz ejemplo de presentaciones en web.

Es posible apreciar en la fig. 4.5. que por el lado del tutor se tiene un control total de la presentación con avance y retroceso de ésta, además de ir directamente a la página que se desee.

En relación a la sincronización del avance de las páginas, se utilizan los objetos compartidos, creándose un slot que permitirá sincronizar los cambios de las páginas. Para tal efecto, la interfaz se conecta a una aplicación de ejemplo que trae el servidor Red5. Esta es SOSample (Shared Object Sample), la cual es una implementación en el servidor de los objetos compartidos. Los botones de control envían la instrucción al servidor y ésta es recibida por la interfaz, la cual cambia la página de la presentación, por lo tanto, el tutor y los alumnos estarán viendo lo mismo.

#### 4.1.4. Interfaz integrada.

Con el objetivo de mostrar el resultado de los ejemplos integrados en una plataforma, se juntaron los módulos de audio-video y presentación.

A continuación una captura de pantalla de la interfaz del tutor.

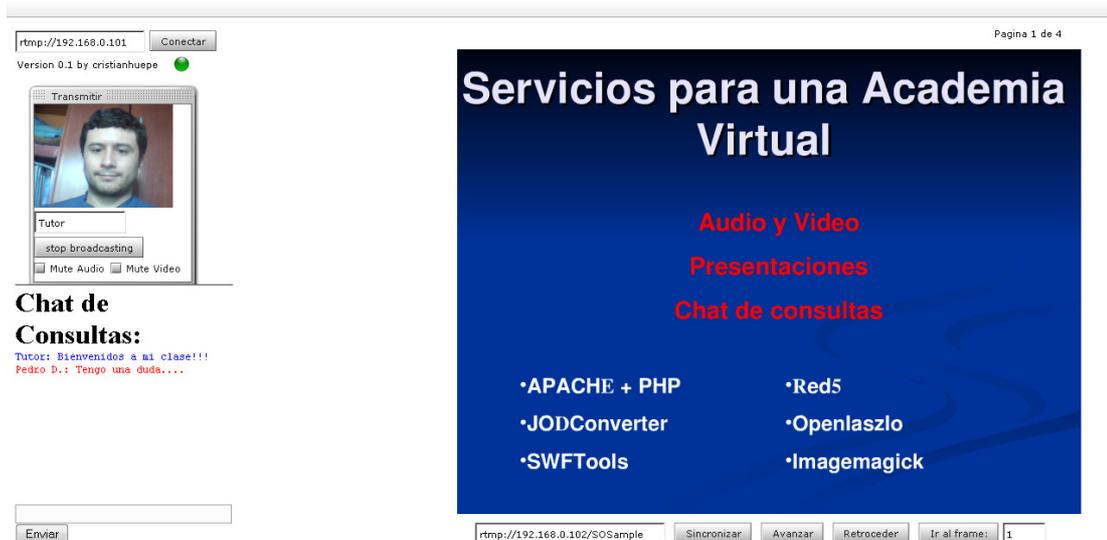


Fig. 4.6. Interfaz integrada para el tutor.

Como se puede apreciar en la fig. 4.6., el tutor tiene el dominio en la reproducción de la presentación y transmite el audio y video de su cámara web; además recibirá las consultas en el chat. De preferencia el tutor deberá responder las consultas de forma oral, y solo si es necesario podrá usar el chat de consultas.

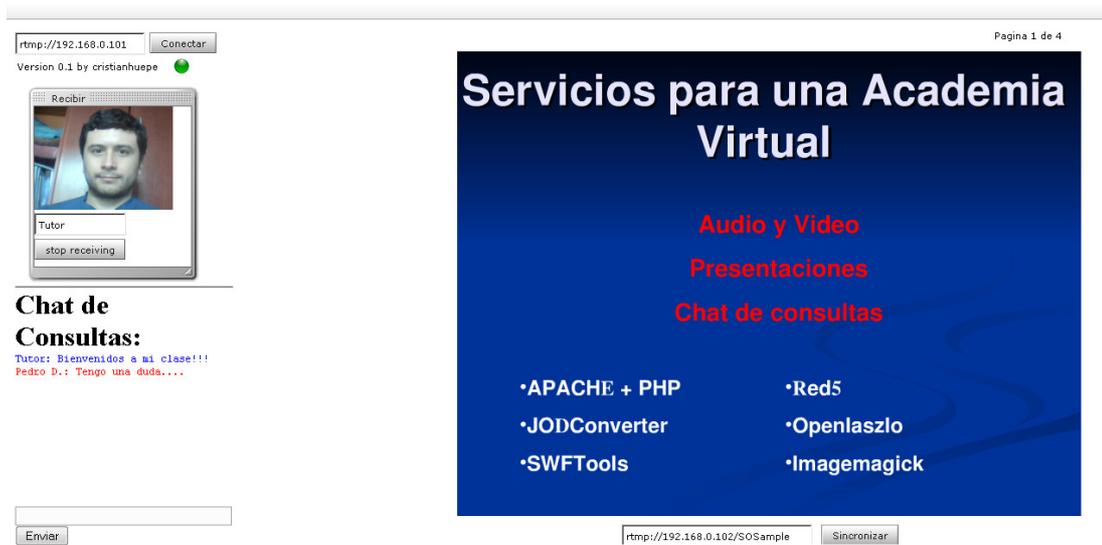


Fig. 4.7. Interfaz integrada para el alumno.

En el caso del alumno, él verá la presentación según el control del tutor y podrá recibir el audio y video de parte de éste, además el podrá realizar consultas según estime conveniente en el módulo de chat.

Para chequear el correcto intercambio de paquetes se realizará una captura de éstos con el software Wireshark, con una configuración donde un pc tiene las dos interfaces en funcionamiento( máquina con IP 192.168.0.102), y otro pc( con dirección ip 192.168.0.101) tiene instalado el servidor de streaming RED5. Luego de realizada la captura se ordenan los protocolos visualizando solo los RTMP.

Dado que esta captura fue realizada antes de conectar las interfaces al servidor Red5 se aprecia la secuencia de conexión, donde se destaca el Handshake, el Ping y el Invoke.

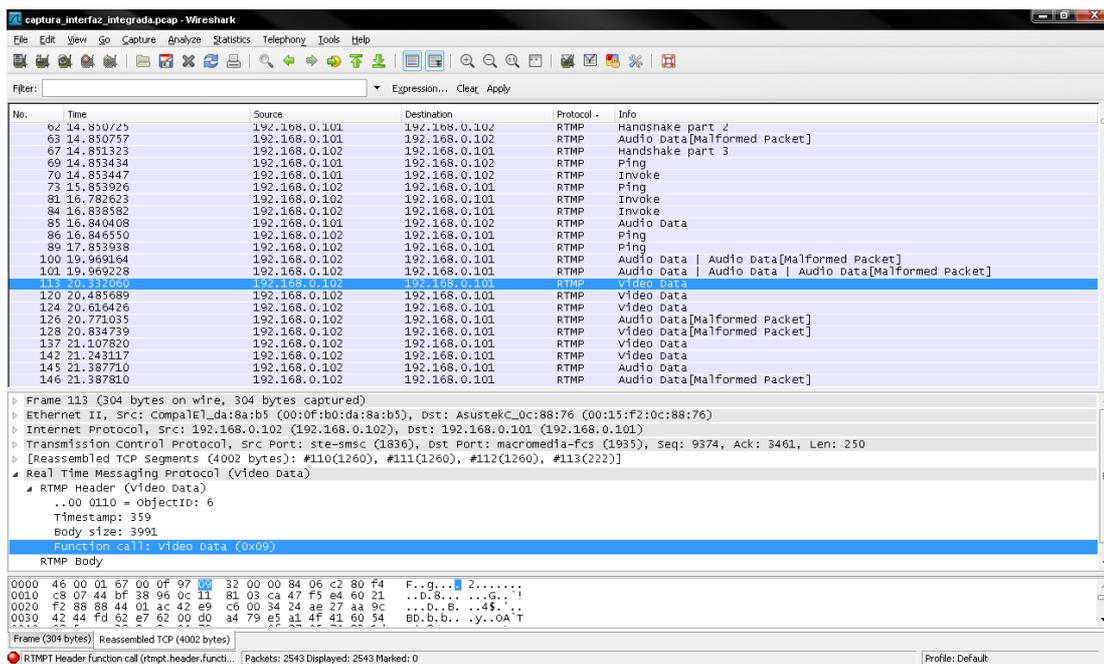


Fig. 4.8. Comunicación de la interfaces con el servidor RED5.

Como se aprecia en la fig. 4.8. el software Wireshark identifica los paquetes RTMP e informa de qué tipo es, en este caso se selecciona un paquete de video, en donde la cabecera del protocolo es de tipo “Video Data”, es decir, información de video, y aludiendo a la Tabla 2.1 se chequea que el id 0x09 corresponda a este tipo. Esto también se puede repetir con los tipos de datos Ping, Invoke, entre otros.

#### 4.1.5. Módulo de telefonía.

Se logra configurar exitosamente una llamada de skype a una cuenta sip, para lo cual se tiene que configurar una cuenta skype en el módulo “chan\_skype” de Asterisk, con el objetivo que este sea visto desde skype como una cuenta más. Además en “extensions.conf” se agrega una extensión a una cuenta sip creada para realizar las pruebas. De esta forma, se logra rutear directamente la llamada. Se realiza la llamada desde una cuenta sip hacia una cuenta skype agregándose para tal efecto la configuración correspondiente en “chan\_sip.conf” y la extensión hacia skype en “extensions.conf”. Se destaca que las pruebas fueron exitosas, presentándose una excelente calidad de voz. Para este efecto se emplea el códec de audio G.711(μ-law), ya que el códec G.729 requiere la compra de una licencia para su uso.

Los pasos que se llevaron a cabo para la configuración de Asterisk y Skype están detallados en el capítulo de Anexos.

## **5. CAPÍTULO V: DISCUSIÓN**

### **5.1. Acerca del funcionamiento.**

Dentro de las particularidades del sistema propuesto, es posible destacar que no se requiere la instalación y configuración de ningún programa, además de la modularidad del mismo dentro de la plataforma, el cual permite que sea ocupado para otros fines. El delay no presentó problemas para realizar una comunicación fluida, con una buena calidad de imagen y permitió el control de la transmisión del audio y video siendo capaz de cortar el envío de uno de éstos al servidor de streaming Red5. Es de esperar que a futuro el códec H.264 sea integrado en red5, con el fin de obtener una mejor calidad de imagen a una menor tasa de bits.

#### **5.1.1. Comunicación audio video.**

La comunicación vía audio y video de parte del tutor hacia los alumnos es esencial en la plataforma, permitiendo que el tutor transmita sus conocimientos en forma rápida y confiable.

Los resultados muestran que la calidad del audio y el video son óptimos para este tipo de servicios. Los requisitos de funcionamiento son mínimos para los usuarios, ya que sólo se requiere un navegador estándar con el plug-in de Flash instalado. Por otra parte, dado el diseño, se requiere un mínimo ancho de banda de subida ya que el alumno no transmite audio y video. Como no se requiere una gran resolución para ver al tutor, los requisitos de ancho de banda de bajada son alrededor de 56kbps.

#### **5.1.2. Comunicación vía texto.**

La comunicación vía texto también pasa a ser parte esencial de la plataforma, ya que permite al tutor tener feedback de la clase, y aclarar las dudas en el instante sin tener que generar desconcentración en la audiencia. En el caso de tener audio para los alumnos se generaría un caos que no permitiría realizar con fluidez una clase.

Como se comentó anteriormente, se propuso el uso de los objetos compartidos que han sido implementados en el servidor Red5, con el fin de dar a conocer su funcionalidad y ventajas sobre las otras alternativas disponibles para realizar esta tarea.

Gracias a un ejemplo que se encuentra junto a la librería JSSO, la cual hace uso de una película SWF que actúa como interfaz de comunicación entre un Javascript y el servidor de streaming, e integrando modificaciones, se logra poner en funcionamiento una chat básico, pero que cumple los objetivos. Con esto se pueden realizar otras aplicaciones, por ejemplo una pizarra compartida, en la cual todos los participantes sean capaces de interactuar.

### **5.1.3. Presentaciones y conversor de documentos.**

Las ventajas de las presentaciones sincronizadas se encuentran en la gran flexibilidad en el manejo de formatos de oficina ya que es capaz de convertirlos al popular formato PDF y en una película SWF.

La potencialidad radica en poder desplegar la presentación sobre múltiples navegadores que tengan instalado el plug-in de Flash, y la flexibilidad que se da al poder presentarla en múltiples formas. Esto último se ve reflejado en los visores utilizados por los servicios de intercambio de presentaciones y del papel digital que integran diversas funcionalidades.

Dentro de los productos que usan SWFTools para la creación de archivos SWF, se encuentran:

- Scrib.com, es un sitio que hospeda una gran cantidad de documentos, principalmente presentaciones tipo powerpoint que utiliza principalmente el conversor pdf2swf.
- Digipaper, es un producto que permite navegar en revistas y diarios online. Gracias al conversor pdf2swf y la calidad del visor desarrollado por ellos, se logra una excelente experiencia de usuario.
- Motion Paper, es una compañía que ofrece publicaciones online de documentos PDF, el cual soporta la búsqueda de texto a través del llamado de funciones en el servidor, ya que no es posible realizarlo directamente sobre la película SWF.

#### **5.1.4. Módulo de telefonía.**

Como anteriormente se mencionó, el canal de Asterisk para Skype, llamado “chan\_skype”, funcionó adecuadamente al establecerse la comunicación con una excelente calidad de voz y generar la llamada desde Skype y desde Asterisk. Aunque no fue posible recibir y generar las llamadas con la misma configuración, esto debería ser solucionado luego que el canal pase la fase beta. Se señala que sólo se evaluó el uso del códec de voz G.711, dado que el códec G.729 es propietario y requiere la compra de una licencia para su uso. Una de las ventajas de emplear el códec G.729 es que sólo se necesita un ancho de banda de 8kbps, mientras que con el códec G.711 se requiere de 64kbps. Estos códecs pueden ser estudiados en el capítulo 2.

Al proyectar un trabajo futuro, sería interesante conectar la interfaz de audio y video a Asterisk por medio de la iniciativa de Red5phone o del canal “chan\_rtmp”, los cuales deberían tomar un impulso luego que Adobe publique la documentación del protocolo RTMP.

#### **5.1.5. Mejoras en la Interfaz.**

Dentro de las mejoras que se pueden realizar a la interfaz del tutor, se recomienda insertar una ventana que permita ver los usuarios conectados, además de una ventana donde se vean los documentos que pueden ser desplegados en la presentación.

### **5.2. Propuesta de nuevos servicios a futuro.**

Gracias a varias iniciativas de código libre que actualmente están en desarrollo, es posible proponer otros servicios que permitan la integración de diversos sistemas que actualmente no son posibles de utilizar vía web sin que se incurra en la compra de licencias. Con el fin de abrir nuevos usos de estos servicios en el mundo de la empresa, principalmente en la video conferencia web, que permite además integrar hardware VoIP, se destacan a continuación una serie de iniciativas, las cuales tienen pocos años de vida e incluso meses, encontrándose en distintas etapas de desarrollo. Muchas de estas iniciativas se encuentran en búsqueda de desarrolladores independientes que aporten en el testing y mejoras del código.

En resumen, se recomienda revisar las siguientes iniciativas.

#### **Revisar avances en el chan\_rtmp de Asterisk.**

Este desarrollo permitiría a Asterisk conectarse como un cliente a un media server como Red5 o Flash Media Server, y a continuación enviar y recibir medias streams de dicho servidor, de esta forma el servidor red5 actuaría como un intermediador. Con esto el canal permitiría a Asterisk gestionar llamadas realizadas mediante este protocolo y unificar dos

“mundos” separados entre sí por una barrera inicialmente infranqueable, por lo que se podría utilizar un componente Flash en una página web que capture audio y video, pasarlo al servidor RED5 y conectarlo a Asterisk para poder hacer llamadas por los múltiples canales habilitados. Entre éstos se destaca SIP.[15]

Este desarrollo utiliza, entre otros, a las librerías libavcodec, incluidas en FFMPEG versión 0.5.

### **Revisar avances en el desarrollo de Red5phone.**

Red5phone, es un webphone SIP para Adobe Flash usando el servidor RED5, que no requiere plug-ins adicionales. Este desarrollo hace uso del servidor RED5 para capturar el audio y luego transcodificarlo (se pasa del códec asao( Flash )<->μ-law( sip ) ), además de utilizar MjSip, el cual es una implementación basada en Java del stack SIP. Implementar este webphone permitiría una rápida penetración de servicios VOIP, ya que no requeriría más que configurar la cuenta de acceso para lograr llamar desde una interfaz web.[16]

### **Revisar el estado del desarrollo en VLC para usar protocolo RTMP para envío de video y audio.**

VLC media player es un reproductor multimedia altamente portable para varios formatos de audio y video (MPEG, DivX/Xvid, Ogg, y muchos más). Sin embargo, en los últimos años se ha convertido además en un servidor extremadamente poderoso para el streaming en vivo y bajo demanda en varios formatos. Actualmente está disponible la recepción de streaming bajo el protocolo RTMP con ciertas limitaciones. Es por esto que se hace necesario estar al tanto de los últimos avances, ya que luego que se liberen las especificaciones de este protocolo por parte de Adobe, se debería dar total soporte a éste.[17]

### **Revisar el applet cortado para realizar streaming de audio y video.**

Cortado, es un applet de Java para streaming de video en formato Ogg. Este applet utiliza el servidor de streaming Flumotion y actualmente incluye decodificadores en Java de Ogg Theora, Ogg Vorbis, Mulaw audio, MJPEG y el códec propio Smoke.[18]

Cortado es un software libre bajo la licencia GPL. Al integrar este applet en un sitio web, se hace posible dar a los espectadores el acceso a los streams, ya sea del servidor de Streaming o reproducir un archivo local en el servidor sin necesidad de una instalación local de un media player que soporte los formatos correctos en el computador del visitante.

### **Revisar el servidor Openfire.**

El servidor Openfire es un servidor de colaboración en tiempo real ( en inglés, real time collaboration server, RTC) y es entregado bajo la licencia GPL. Este servidor utiliza sólo el protocolo abierto ampliamente adoptado para la mensajería instantánea XMPP, también llamado Jabber. Las particularidades que merecen la atención de este servidor, son los Plugins Asterisk-IM y el SIP Phone Plugin. El primero permite gestionar los mensajes a través de Asterisk en tanto que el segundo permite la comunicación de audio bajo el protocolo SIP. Por estas razones se recomienda revisar la documentación respectiva y poner a prueba estos módulos.[19]

### **Revisar el sistema de videoconferencia OpenMeetings.**

OpenMeeting es un sistema de videoconferencia que permite un número ilimitado de usuarios. Ocupa una gran variedad de software libre y actualmente está siendo portado a la plataforma Openlaszlo. Este sistema permite por medio de una interfaz web realizar reuniones en vivo, además de desplegar presentaciones y un sistema de chat. OpenMeeting fue testeado y se recomienda para su uso.[20]

### **Revisar la iniciativa de Adobe con Open Screen Project.**

El Open Screen Project es una iniciativa de la industria liderado por Adobe y respaldado por otros líderes del sector que comparten una visión clara: que los consumidores puedan interactuar con las experiencias enriquecidas de Internet sin problemas a través de cualquier dispositivo y en cualquier lugar. La primera y más importante noticia que trae el Adobe Open Screen Project es que elimina las restricciones de uso de las especificaciones de SWF, FLV y F4V.[21]

Como se comenta en la página del proyecto, los socios de esta iniciativa están trabajando juntos para proveer un runtime consistente para exploración Web y aplicaciones individuales abiertas, aprovechando Adobe Flash Player y, en el futuro, Adobe AIR. Este entorno de ejecución coherente eliminará las barreras para la publicación de contenido y aplicaciones a través de computadores, teléfonos móviles, televisores y otros electrónicos de consumo.

### **Revisar herramienta de desarrollo de software Eclipse.**

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).[22]

Actualmente se está trabajando en una implementación del protocolo SIP para el Eclipse Communication Framework bajo la licencia Eclipse Public Licence.

### **Revisar las nuevas funciones que trae HTML5.**

HTML 5 es la quinta revisión mayor del lenguaje básico de la World Wide Web. HTML 5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml). Se destaca que en esta versión será posible ver video y audio nativamente desde el navegador. Según la especificación que trabaja la W3C será posible ver video en streaming.

Uno de los grandes problemas que se presenta en integrar video de forma nativa en el navegador, es la elección del códec. Las listas de correo creadas por la W3C para la discusión de este, han sido infructuosas principalmente porque la gran mayoría de los códecs poseen derechos de propiedad intelectual en la codificación de audio y vídeo, siendo que se busca una implementación libre de pago por uso. Además de destacar los intereses comerciales de las grandes empresas que tienen navegadores web, como Microsoft y Apple.

### **5.3. Acerca de los objetivos.**

Los resultados obtenidos en los distintos módulos, permiten concluir que se han cumplido los objetivos generales y específicos.

El objetivo general del trabajo de título fue logrado al mostrar, utilizar y proyectar las alternativas open source o de código libre en distintos servicios basados en web para una academia virtual u otro proyecto que lo requiera.

Los objetivos específicos fueron abordados en las distintas fases del trabajo de memoria, justificándose el por qué del uso de una de las alternativas disponibles. Además, a partir de dichas alternativas se desarrollaron ejemplos funcionales de los distintos servicios posibles de desplegar. Por otra parte, estos servicios se han proyectado hacia otras aplicaciones como la videoconferencia y por medio de recientes iniciativas, se señaló la posibilidad de implementar un teléfono ip basado en Web ( en inglés Webphone), el cual no requiera más que tener instalado el plug-in de Adobe Flash.

## 6. CAPITULO VI: CONCLUSIONES

La metodología llevada a cabo con una revisión de las posibles tecnologías que cumplan con los objetivos propuestos y la elección de una de ellas vía criterios de rápida implementación y penetración, y de desarrollo open source o de código libre, que permita con bajo costo desarrollar e implementar los servicios propuestos para la academia virtual, además de la elección de un diseño que requiera un mínimo de ancho de banda, permitió proyectar de excelente forma el trabajo efectuado en esta memoria.

Las iniciativas open source o de código libre analizadas y puestas a pruebas, vía ejemplos desarrollados en este trabajo de título, permiten validar los componentes por sus potencialidades para ser usados en variados servicios basados en web, que en un futuro próximo puedan integrarse a otros sistemas como hardware VoIP. Cabe destacar, que la iniciativa Red5 actualmente tiene limitaciones en cuanto a los códecs de video, ya que sólo es posible utilizar el códec Sorenson Spark, pero se espera que en las próximas entregas se disponga del códec H.264.

Como se mencionó anteriormente, Red5 no soporta H.264, en vías que la industria impulsada por la propietaria de Flash, Adobe, apoyan el uso o cambio al códec de video H.264, teniendo en mente que es muy probable que este sea el códec que se empleará en sistemas colaborativos basados en sistemas vía web, software para pc, o hardware específico. Por lo tanto, se hace también inevitable estar al tanto de las capacidades de Asterisk para gestionar este códec.

Por medio de los resultados satisfactorios en la puesta en funcionamiento de los ejemplos desarrollados, con el fin de validar los servicios propuestos basados en alternativas open source o de código libre y luego proyectarlos en futuros trabajos y servicios, se da como cumplido el objetivo general.

Cabe señalar que uno de los grandes inconvenientes que se produjo y que se tiene en los desarrollos con software open source o de código libre es la escasa documentación que presentan, sea esta referida a la instalación o la arquitectura de software, lo que conlleva a la escasa participación de nuevos desarrolladores debido principalmente a la dificultad de comprender de manera rápida y eficiente la arquitectura del software, lo que entorpece la depuración de los bugs.

Se destaca la importancia que tiene la decisión de liberar los documentos referentes al protocolo RTMP por parte de Adobe, ya que esto permitirá un nuevo frente de desarrollo en sistemas multimedia basados en web, que sean capaces de integrarse a otras plataformas como la telefonía IP, sistemas de mensajería y dispositivos móviles, entre otros.

En el caso del diseño para los alumnos de la academia virtual, la interfaz otorga seguridad en lo que se está enseñando, al permitirse la transmisión del audio y el video del tutor, además de respaldar la información por medio de las presentaciones y resolver las dudas de manera rápida y sin interrumpir a la audiencia.

El módulo de telefonía-audio probado y analizado permite concluir que a futuro será ampliamente utilizado, en miras de aprovechar la gran penetración de Skype en las empresas transnacionales, con el fin de disminuir o eliminar los cobros que realizan las compañías telefónicas.

La utilización de un códec estándar en voz y video ahorrará en la transcodificación, lo que no sucede actualmente en los webphone comerciales basados en Flash, donde se requiere pasar a Asao (Flash) $\leftrightarrow$   $\mu$ -law(SIP) en el servidor.

Una línea de trabajo está relacionada con la integración de las iniciativas comentadas en el apartado de discusión referente a las posibilidades de implementar un webphone, que haga uso de Red5 para conectarse a Asterisk. Esto permitirá tener salida a la PSTN por medio de un operador de VoIP o con el hardware necesario conectado a Asterisk, o la conexión con Skype por medio del canal "chan\_skype" que fue instalado y probado en este trabajo de título.

Para futuros trabajos de título se sugiere enfocarse además de la academia virtual, en videoconferencia, sistemas de atención web por medio de un webphone que se conecte con un callcenter, sistemas webinars, reconocimiento de voz e imágenes vía web y otras herramientas colaborativas como edición de un documento por múltiples personas de forma simultánea en web, entre otros.

## 7. CAPITULO VI: ANEXOS

### 7.1. Referencias bibliográficas.

- [1] <http://www.opensource.org/> -Página oficial de la fundación Open Source.  
<http://www.fsf.org/> -Página de la Free Software Foundation.  
<http://www.gnu.org/> -Página del proyecto GNU.
- [2] <http://wiki.gnashdev.org/RTMP> -Descripción general del protocolo RTMP.  
<http://rtmpy.org/wiki/RTMP> -Descripción detallada del protocolo RTMP.
- [3] <http://osflash.org/red5> - Página del proyecto Red5  
<http://code.google.com/p/red5/> - Página donde se almacena el código de Red5.
- [4] <http://www.openlaszlo.org/lps4.2/docs/developers/video.html> -Documentación de OpenLaszlo sobre configuración del video.
- [5] <http://artofsolving.com/opensource/jodconverter> -Página oficial del proyecto.
- [6] <http://www.swftools.org/> -Página oficial del proyecto SWFTools
- [7] <http://www.imagemagick.org/> -Página oficial del proyecto.
- [8] <http://www.skype.com/> -Página oficial de Skype
- [9] <http://www.asterisk.org/> -Página oficial de Asterisk  
[http://files.pablasso.com/docs/manual\\_asterisk.pdf](http://files.pablasso.com/docs/manual_asterisk.pdf) -Manual básico de configuración de Asterisk.
- [10] <http://osflash.org/flv> -Descripción del formato Flash Video
- [11] [http://help.adobe.com/en\\_US/Flash/10.0\\_UsingFlash/](http://help.adobe.com/en_US/Flash/10.0_UsingFlash/) -Sobre Códec Sorenson Spark.
- [12] [http://en.wikipedia.org/wiki/Nellymoser\\_Asao\\_Codec](http://en.wikipedia.org/wiki/Nellymoser_Asao_Codec) -Información del Códec Asao  
<http://ffmpeg.org/> -Página del proyecto FFmpeg
- [13] <http://www.voipforo.com/codec/codecs.php> -Sobre Códecs G.711 y G.729.  
<http://www.itu.int/rec/T-REC-G/en> -Sobre Códecs G.711 y G.729.  
[http://www.sipro.com/g729\\_whychoose.php](http://www.sipro.com/g729_whychoose.php) -Web oficial de Sipro Lab Telecom proveedor de licencias del códec G.729.
- [14] <http://www.wireshark.org/> -Página del proyecto Wireshark
- [15] <https://issues.asterisk.org/view.php?id=15484> -Página de discusión del canal chan\_rtmp.
- [16] <http://code.google.com/p/red5phone/> -Página del proyecto Red5phone.
- [17] <http://www.videolan.org/videolan/> -Página del proyecto VideoLan.
- [18] <http://www.flumotion.net/cortado/> -Página del proyecto Cortado.
- [19] <http://www.igniterealtime.org/projects/openfire/index.jsp> -Página del proyecto OpenFire.
- [20] <http://code.google.com/p/openmeetings/> -Página del proyecto Openmeeting.
- [21] <http://www.openscreenproject.org/> -Página del proyecto Open Screen.
- [22] <http://www.eclipse.org/> -Página del proyecto Eclipse.

### 7.2. Memorias

Diseño e Implementación de una Ip-Contact Center Distribuida Económica y con Fines Docentes .

[http://www.cybertesis.cl/tesis/uchile/2007/tchernitchin\\_nl/html/index-frames.html](http://www.cybertesis.cl/tesis/uchile/2007/tchernitchin_nl/html/index-frames.html)

## 7.2. Configuraciones.

### 7.2.1. Pasos de configuración de SFA (Skype for Asterisk).

- 1) Comprar licencia beta de SFA, se envía una key al correo, la cual se solicita al momento de instalar SFA.
- 2) Instalar Asterisk 1.4.
- 3) Instalar módulo SFA.
- 4) Registrar módulo SFA con licencia beta.
- 5) Crear cuenta Skype BCP ( Business Control Panel), única forma de crear usuarios en SFA.
- 5) Habilitar en "chan\_skype.conf" el usuario creado en BCP.
- 6) Chequear que el usuario este logueado vía comando en CLI> de Asterisk.  
"skype show users" ( ejemplo de resultado esperado: empresashuepe "Logged In" ).

Se detalla a continuación algunos de los pasos que se requieren para la instalación.

- Pasos de la instalación de SFA.

```
$ tar xzvf skypeforasterisk-1.4_0.9.10.x86_32.tar.gz
$ cd skypeforasterisk-1.4_0.9.10-x86_32
$ make
$ make install
```

- Se modifica el archivo "chan\_skype.conf" (ver el archivo "Readme.txt" que viene en el módulo SFA para mayor detalle) en la parte de user section, se ve un modelo a seguir para configurar la cuenta de Skype, para esto se modifica el username, el password, y el context ( como default esta "demo", el cual es una IVR que informa que se tiene una instalación exitosa). Además se debe tomar en cuenta "direction", este atributo permite habilitar las llamadas entrantes o solo salientes. En este caso no fue posible habilitar la entrada y salida de llamadas. Se debe destacar que se ha seleccionado el códec  $\mu$ -law, debido a que el códec G.729 requiere licencia. Considerar que se debe agregar "auth\_policy=accept", para aceptar sin problemas los usuarios que agregan a su lista de contactos al cliente creado en BCP.

- Luego de agregar a una cuenta de Skype el usuario creado en BCP, se hace una llamada a éste para comprobar que todo está funcionando como se espera.

- En este paso se debe crear una cuenta SIP, con el objetivo de trasladar la llamada. Para esto se crea un usuario tipo con la siguiente configuración:

En sip.conf agregar:

```
[33]
type=friend
secret=welcome
qualify=yes ; Qualify peer is not more than 2000 mS away
nat=no ; This phone is not natted
host=dynamic ; This device registers with us
canreinvite=no ; Asterisk by default tries to redirect
context=demo3;
port=5065 ; descomentar si el cliente SIP esta en la misma maquina que Asterisk
```

Como se comenta en el código , si se usa la misma máquina, de debe cambiar el puerto de escucha de nuestro cliente SIP.

En extensions.conf agregar:

```
[demo3]
exten => 900,1,Dial(Skype/cristianhuepe)
exten => 900,2,Hangup()
```

- Como ya se señaló anteriormente, se modifica "direction=outgoing", en el archivo "chan\_skype.conf", y de esta forma permitir la salida de llamadas hacia Skype.

- Se realiza la llamada desde el cliente SIP, y si todo sale bien se recibirá una llamada en Skype.

- Ahora para direccionar la llamada entrante se debe cambiar en el archivo "chan\_skype.conf" los atributos "direction=Incoming" y "context=demo2" y en sip.conf se agrega,

```
[demo2]
exten => s,1,NoOp(Incoming Skype Call!)
exten => s,2,Dial(SIP/33)
exten => s,3,Hangup()
```

9) Se realiza la llamada desde skype al usuario creado en BCP, y si todo sale bien se recibirá una llamada en el cliente SIP.

## 7.2.2. Pasos de instalación del conversor.

En ubuntu se debe realizar lo siguiente:

Instalar SWFTools

```
sudo apt-get install swftools
```

Instalar ImageMagick

```
sudo apt-get install imagemagick
```

Instalar OpenOffice

```
sudo apt-get install openoffice.org-headless
```

```
sudo apt-get install openoffice.org-writer openoffice.org-calc openoffice.org-impress openoffice.org-draw openoffice.org-math
```

Instalar JodConverter

Para usar JODConverter desde la línea de comandos, se debe descargar el archivo jodconverter-2.2.0.zip, descomprimirlo y ejecutar jodconverter-cli-2.2.0.jar con java.

Más información en la página web del proyecto <http://www.artofsolving.com/>

## 7.2.3. Código Interfaz Audio Video.

```
<canvas debug="true">
```

```
  <rtmpconnection id="server"
    src="rtmp://192.168.0.102"
    autoconnect="false"
```

```
  /> // conexion a un servidor RTMP como RED5, se da una direccion predeterminada
```

```
<simplelayout axis="y" spacing="5" />
```

```
<view>
```

```
  <simplelayout axis="x" spacing="5" />
```

```
  <view>
```

```
    <edittext name="url_server" text="rtmp://192.168.0.102" width="150"> // direccion del servidor
```

```
    <handler name="onblur">
```

```
      server.setAttribute('src', parent.url_server.text);
```

```
    </handler>
```

```
  </edittext>
```

```
  </view>
```

```
  <view>
```

```
</button
```

```

        onclick="server.connect()"
    >Conectar</button> // boton para realizar la conexion con el servidor
    </view>
</view>

    <mediastream name="s1" // permite realizar un stream de video y audio
        type="rtmp"
    />

    <mediastream name="s2"
        type="rtmp"
    />

<view>
    <simplelayout axis="x" spacing="4"/>

    <text multiline="true" width="180">
    Version 0.1 by cristianhuepe<br/>
    </text>

    <rtmpstatus/> // muestra un indicador que esta en verde si estamos conectados al servidor RTMP

</view>

<view layout="axis:x; inset:10; spacing:10" >
    //<simplelayout axis="y" spacing="4"/>

    <view id="v1" layout="axis:x; spacing:4" >

<window x="5" y="10" width="200" height="250" title="Transmitir" resizable="true" closeable="false">
    <simplelayout axis="y" spacing="4"/>

        <videoview id="live" type="rtmp" width="160" height="120" stream="$once{canvas.s1}" debug="true">

            <camera id="cam" width="160" height="120"
                show="false"
            /> // tag para conectar la camara web a un servidor rtmp
            <microphone id="mic" capturing="false"/> // tag del microfono

        </videoview>

        <edittext name="username">Tunombre</edittext> // nombre asignado a la url de transmision

        <button>
            <attribute name="text"
                value="$${s1.broadcasting == false} ? 'broadcast' : 'stop broadcasting'"
            />
            <handler name="onclick"><![CDATA[
                if (cam.show == false) {
                    live.stream.setAttribute('url', parent.username.text);
                    live.stream.broadcast(); // transmitir en modo broadcast
                    cam.setAttribute('show', true); // transmision desde la camara
                } else {
                    live.stream.stop();
                    cam.setAttribute('show', false);
                }
            ]]>
        </button>
    </view>
</window>

```

```

    ]]>
  </handler>

</button>
<view >
  // Control de la transmision del audio y video
<simplelayout axis="x" spacing="5" />
<checkbox onvalue="s1.setAttribute('muteaudio', value)">Mute Audio</checkbox>
<checkbox onvalue="s1.setAttribute('mutevideo', value)">Mute Video</checkbox>
</view>

</window>
</view>

<view id="v2" layout="axis:x; spacing:4" >

<window x="5" y="10" width="200" height="220" title="Recibir" resizable="true" closeable="false">

  <simplelayout axis="y" spacing="4"/>

  <videoview name="vid"
    type="rtmp"
    stream="$once{canvas.s2}" width="160" height="120"
  /> // reproduccion del stream

  <edittext name="username">Tunombre</edittext> // ingreso de la direccion de la transmision

  <button
    text="$ {s2.playing ? 'stop receiving' : 'receive'}"
    onclick="s2.setAttribute('url', parent.username.text);
      if (s2.playing) s2.stop(); else s2.play();"
  /> // seteo de la direccion y reproduccion del stream

</window>
</view>

</view>

</canvas>

```

#### 7.2.4. Código Interfaz Presentaciones.

```

<canvas height="600" debug="true">

// pelicula swf que actua como interfaz
<view id="solib_view" width="100" height="200" resource="solibrary.swf" >

  <method name="getLib" >
    <![CDATA[
      return solib_view.getMRef().solib;
    ]]>
  </method>

```

```

<method name="getList" >
    <![CDATA[
        return solib_view.getLib().soList;
    ]]>
</method>
<method name="getRemote" args="name,uri,persistence">
    <![CDATA[
        if(persistence == "false"){
            solib_view.getLib().getRemote(name,uri,false);
        }else{
            solib_view.getLib().getRemote(name,uri,true);
        }
    ]]>
</method>
</view>

// pelicula swf que contiene la presentacion creada con el conversor
<resource name="myresource" src="resources/c5d6e1_Presentaciontest.swf"/>
<attribute name="current" value="1" type="number"/>
<attribute name="max" value="1" type="number"/>

// metodo que permie setear las pagina
<view id="set">
<method name="setear_pagina" args="numero">

        var sync_pages = new Object();
        var slot="pages"; // nombre del arreglo
        sync_pages.nppt=numero; // objeto a almacenarce en data[pages] contiene la variable
nppt

        solib_view.getList()['test'].data[slot]=sync_pages;

</method>

</view>
<view width="700" height="700">
    <simplelayout axis="y" spacing="10"/>
    <view align="right">
    <text text="\${ ' Pagina ' + me.frame + ' de ' + me.totalframes }"/>
    </view>

    <view id="me" resource="myresource" oninit="this.stop();" />
// coneccion con el servidor RTMP
<view align="center" >
    <simplelayout axis="x" spacing="10"/>
    <edittext id="url_server" text="rtmp://192.168.0.102/SOSample" width="220" />
    <button id="initbtn" onclick="this.initme(url_server.text)" text="Sincronizar" >
    <method name="initme" args="server">
        <![CDATA[
            nc = new NetConnection();

            nc.connect(server);
            solib_view.getLib().getRemote("test",nc.uri,false);
            solib_view.getList()['test'].onSync = function(objList){

```

```

        for (var n in objList) {
            var obj = this.data[objList[n].name];

            me.setAttribute('frame', Number(obj.nppt));
            ppts.tt.setAttribute(Number(obj.nppt))
        }
    }

    solib_view.getList()['test'].connect(nc);
]]>
</method>
</button>

```

```

// metodos de avance,retroceso e ir a
<view>
    <button onclick="Avanzar();"> Avanzar
        <method name="Avanzar" >

            this.max = me.totalframes;
            //Debug.write(this.max);
            this.current=me.frame;
            Debug.write(this.current);
            if( this.current >= this.max )
                this.current = 1;
            else
                ++this.current;
                Debug.write(this.current);

            // me.setAttribute('frame', Number(this.current));

set.setear_pagina(this.current);

        </method>
    </button>
</view>

<view>
<button onclick="Retroceder();">
    Retroceder
        <method name="Retroceder" >

            this.max = me.totalframes;
            //Debug.write(this.max);
            this.current=me.frame;
            Debug.write(this.current);
            if( Number("1") >= this.current )
                this.current = this.max;
            else
                --this.current;
                Debug.write(this.current);

            //me.setAttribute('frame', Number(this.current));

```

```

set.setear_pagina(this.current);

        </method>
</button>
</view>

<view id="ppts" layout="axis: x; spacing: 4">
<button isdefault="true" text="Ir al frame:"
onclick="this.chequear_rango(Number(tt.text))">
<method name="chequear_rango" args="num" >

    this.max = me.totalframes;
    //Debug.write(this.max);
    this.current=me.frame;
// Debug.write(this.current);
if( Number("0") >= num || num >=this.max )
    canvas.myalert.open()
    else {
        Debug.write(num);

//me.setAttribute('frame', Number(this.current));

set.setear_pagina(num);
    }

</method>
</button>

<edittext id="tt" width="50" > 1 </edittext>
</view>

</view>

</view>

// alerta cuando en "ir a" se setear una pagina fuera de rango
<alert name="myalert">
    Fuera de Rango!!!!
</alert>

</canvas>

```

### 7.2.5. Código Interfaz Consultas.

Este código corresponde al chat del tutor, el del alumno es idéntico solo varía en la variable from.

```
<html><head>
<style>
h1 {
    margin: 0;
}
#ChatLog {
    width: 350px;
    height: 180px;
    font-size: 12px;
    font-family: Courier New;
}
input[type=text] {
    width: 250px;
}
</style>

<script type="text/javascript" src="js/swfobject.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jsSO.js"></script>

<script type="text/javascript">

jsSO.connect('rtmp://192.168.0.101/SOSample', 'chat2');

// select color
var rand = Math.round(Math.random()*100000)%4;
var color = (['blue', 'red', 'green', 'black', 'yellow'])[rand];

// handle stage pixel updates
jsSO.onSync(function(event, updates, data) {
    if (updates.message) {
        var date = new Date();
        date.setTime(data.message.time);

        $('#ChatLog').append('<div style="color: '+data.message.color+'">'+data.message.from+' '+data.message.text+'</div>');
    }
});

$(function(){
    $('body').prepend('<h1>Chat de Consultas: </h1>');
    $('input[type=text]').focus();

    function send()
    {
        var text = $('input[type=text]').val();
        if (!text) {
            return false;
        }
    }
});
</script>
```

```

        jsSO.set('message', {
            from: "Tutor",
            color: color,
            time: (new Date()).getTime(),
            text: text
        });

        $('input[type=text]').val("").focus();

        return false;
    }

    $('input[type=button]').click(send);
    $('form').submit(send);
});
</script>

</head>
<body>
<form>
    <div id="ChatLog"></div>
    <input type="text" /><input type="button" value="Enviar" />
</form>
</body>
</html>

```

### 7.2.6. Código Interfaz Conversor.

```

<?php
$status = "";
$extPermitidas="";
//funcion que chequea la extension del archivo
function checaExtension($ext){
    // Lo pasamos a minusculas
    $extlow= strtolower($ext);

    //Extensiones permitidas
    $extPermitidas=array("pdf","doc","docx","xls","xlsx","ppt","pptx","odt","ods","odp","txt","rtf");
    for($i=0;$i<count($extPermitidas)-1;$i++){
        if($extlow==$extPermitidas[$i])
            return true;
    }
    return false;
}
// si es pdf no es necesario convertir , no uso de jodconverter
function checaExtension_pdf($ext){
    // Lo pasamos a minusculas
    $extlow= strtolower($ext);

    //Extensiones permitidas, agrega o elimina extensiones
    $extPermitidas=array("pdf");
    for($i=0;$i<count($extPermitidas)-1;$i++){
        if($extlow==$extPermitidas[$i])

```

```

    return true;
}
return false;
}

if ($_POST["action"] == "upload") {
    // obtenemos los datos del archivo
    $tamano = $_FILES["archivo"]["size"];
    $tipo = $_FILES["archivo"]["type"];
    $archivo = $_FILES["archivo"]["name"];
    $prefijo = substr(md5(uniqid(rand()),0,6)); // creamos el prefijo
    $path_parts = pathinfo($archivo);

    if($archivo != ""){ // verificamos que se ingreso un archivo

        if (checaExtension($path_parts['extension'])) {
            $path_parts = pathinfo($prefijo."_".$archivo);
            // crear carpeta
            $nombre_carpeta= "files/".$path_parts['filename']; // desde PHP 5.2.0
            // chequeamos que no existia
            if(!is_dir($nombre_carpeta)){
                @mkdir($nombre_carpeta, 0700);
                // fijamos la ruta donde se copiara el archivo
                $destino = $nombre_carpeta."/".$prefijo."_".$archivo;
                // guardamos el archivo en la nueva carpeta
                if (copy($_FILES['archivo']['tmp_name'],$destino)) {
                    $status = "Archivo subido: <b>".$archivo."</b>";

                    // si no es .pdf usamos jodconverter
                    if(!checaExtension_pdf($path_parts['extension'])){
                        // llamada a jodconverter java -jar lib/jodconverter-cli-2.2.0.jar document.doc
                        document.pdf
                        exec("java -jar lib/jodconverter-cli-2.2.2.jar ".$destino."
                        ".$nombre_carpeta."/".$path_parts['filename'].".pdf");

                    }
                    // Convertimos el archivo.pdf a una pelicula flash archivo.swf
                    exec("pdf2swf ".$nombre_carpeta."/".$path_parts['filename'].".pdf -o
                    ".$nombre_carpeta."/".$path_parts['filename'].".swf");
                    // Convertimos el archivo.pdf a un arreglo de imagenes archivo-i.png
                    exec("convert ".$nombre_carpeta."/".$path_parts['filename'].".pdf
                    ".$nombre_carpeta."/".$path_parts['filename'].".png");

                }
            } else {
                $status = "Error al subir el archivo";
            }

        } else {
            echo "Ya existe ese directorio\n";
        }
    } else {
        $status = "Extensi&oacute;n no permitida";
    }
}
else
echo "Ingrese archivo!!!";

```

```

}
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title> Conversor PHP - Servicio Academia Virtual</title>
<link href="estilo.css" rel="stylesheet" type="text/css" />
</head>
<body>
<center>
<table width="413" border="0" cellspacing="0" cellpadding="0">

<tr>
<td width="413" height="40" class="titulo"> Conversor PHP - Servicio Academia Virtual</td>
</tr>
<tr>
<td class="text">Por favor seleccione el archivo a subir:</td>
</tr>
<tr>
<form action="uploadv4_jod_swfv2.php" method="post" enctype="multipart/form-data">
<td class="text">
<input name="archivo" type="file" class="casilla" id="archivo" size="35" />
<input name="enviar" type="submit" class="boton" id="enviar" value="Upload File" />
<input name="action" type="hidden" value="upload" /> </td>
</form>
</tr>
<tr>
<td class="text" style="color:#990000"><?php echo $status; ?></td>
</tr>
<tr>
<td height="30" class="subtitulo">Listado de Archivos Subidos y Convertidos </td>
</tr>
<tr>
<td class="infsub">
<?php
// desplegamos las carpetas que se han creado con los documentos convertidos
if ($gestor = opendir('files')) {
    echo "<ul>";
    while (false !== ($sarch = readdir($gestor))) {
        if ($sarch != "." && $sarch != "..") {
            echo "<li><a href='\"files/\".\"$sarch.\"' class='\"linkli\"'>\".$sarch.\"</a></li>\n";
        }
    }
    closedir($gestor);
    echo "</ul>";
}
?> </td>
</tr>

</table>
</center>
</body>
</html>

```