

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**Diseño e implementación de una interfaz multimedia
interactiva basada en un servidor de video para un
laboratorio remoto virtual**

Tesis para optar el Título de Ingeniero de las Telecomunicaciones, que
presenta el bachiller:

Andrew Roy Ramos Castellanos

ASESOR: Ing. Arturo Díaz Rosemberg

Lima, marzo del 2012

Resumen

El presente proyecto de tesis consiste en el desarrollo de una interfaz multimedia interactiva basada en un servidor de video para la implementación de un laboratorio remoto virtual. Esta interfaz permitirá la visualización de los equipos que se encuentren remotamente y la interacción entre usuarios conectados a una sesión de laboratorio.

El primer capítulo está centrado en el análisis de la situación actual del *e-learning* y las ventajas que traería consigo el laboratorio virtual.

El segundo capítulo busca presentar el análisis desarrollado de todas las tecnologías disponibles para el desarrollo de la interfaz.

El tercer capítulo presenta los conceptos importantes que se deben considerar para que la aplicación pueda cumplir con los objetivos trazados, así como la elección y justificación de las herramientas a utilizar para llevar a cabo la solución planteada.

El cuarto capítulo esta enfocado al diseño de la solución. Así pues se presenta arquitectura del sistema, las consideraciones que se deben tener con el servidor de video y el diseño de las aplicaciones que formarán parte de la interfaz.

El quinto capítulo describe la implementación de las aplicaciones y la integración de éstas para obtener la interfaz que se presenta como solución de este trabajo de tesis. Asimismo, luego se realiza una evaluación de la interfaz desarrollada y del servidor a través de pruebas en Internet.



Dedicado a Blanca, José Luis, Daniela y Vanessa.

Agradecimientos

En primer lugar quiero agradecer a mis padres Blanca y José Luis por que su ejemplo de esfuerzo, dedicación y trabajo me alentaron a siempre seguir adelante y cumplir mis objetivos. Asimismo sin su apoyo incondicional no hubiera sido posible llevar a cabo mis estudios y finalizar la carrera.

Del mismo modo agradezco a mis abuelos Félix y Rafael, así como mis abuelas Mery y Esther, que en paz descansen por inculcar a mis padres los valores de que desde niño me enseñaron, además de la importancia de tener estudios universitarios.

El desarrollo de esta tesis no hubiera sido posible sin el apoyo de mi asesor Arturo Díaz Rosenberg ya que gracias a su soporte y consejos durante todo este tiempo pude desarrollar el documento que hoy presento. También agradezco a los ingenieros Ángelo Velarde y Enrique Larios por hacer que descubra la importancia de mi carrera para el desarrollo de la sociedad y mi país a lo largo de mis años de estudio.

Por último agradezco a mi compañero el ingeniero Gustavo Mendoza por su apoyo en la elaboración de esta tesis.

Finalmente agradezco a Dios y Jesucristo por acompañarme en todo este camino.

Índice

Índice.....	vii
Lista de Figuras	ix
Lista de Tablas	x
Introducción	11
Capítulo 1 Análisis de la situación actual.....	12
1.1 Definición del Problema	12
1.2 Justificación.....	13
1.3 Educación en la era tecnológica	13
1.3.1 <i>e-learning</i>	14
1.3.2 <i>m-learning</i>	15
Capítulo 2 Estudio de las Tecnologías a Usar.....	17
2.1 Laboratorios sobre Internet	18
2.1.1 Laboratorio virtual	18
2.1.2 Laboratorio remoto	18
2.1.3 Laboratorio virtual remoto.....	19
2.2 Tecnologías de <i>Streaming</i>	19
2.2.1 Arquitectura de la tecnología <i>Streaming</i>	21
2.2.2 Flash.....	23
2.2.3 Windows Media	24
2.2.4 Quicktime.....	24
2.2.5 Silverlight	25
2.3 Protocolos de <i>Streaming</i>	25
2.3.1 RTSP	26
2.3.2 RTMP	26
2.4 Aplicaciones de Internet Enriquecidas	27
2.4.1 Adobe Flash	29
2.4.2 Adobe Flex	30
2.4.3 JavaFX	32
2.4.4 Microsoft Silverlight.....	33
2.4.5 HTML5.....	34
Capítulo 3 Análisis de la solución	35
3.1 Aspectos a considerar para la elección de las mejores herramientas que se adecue a la implementación.....	36

3.1.1	Servidor de video.....	36
3.1.2	Ancho de banda requerido	37
3.1.3	Códecs usados	39
3.1.4	Compatibilidad	40
3.1.5	Herramientas de desarrollo	40
3.1.6	Seguridad	40
3.1.7	Precio	41
3.2	Elección de las herramientas a utilizar	41
3.2.1	Análisis de las necesidades del laboratorio	42
3.2.2	Selección de las herramientas que mejor integren las necesidades del laboratorio.....	43
3.3	Descripción de la solución.....	47
Capítulo 4 Diseño de la solución		49
4.1	Arquitectura de la solución	50
4.2	Configuración del Servidor de Video	53
4.3	Diseño de las aplicaciones	55
Capítulo 5 Prototipo de la solución		59
5.1	Implementación del prototipo	60
5.1.1	Desarrollo de aplicaciones.....	60
5.1.2	Integración de componentes	71
5.2	Evaluación del prototipo	75
Conclusiones y Avances Futuros.....		86
	Conclusiones	86
	Avances Futuros.....	88
Bibliografía		89

Lista de Figuras

Figura 2 - 1 Arquitectura de laboratorio remoto.....	19
Figura 2 - 2 Arquitectura de un Sistema de Streaming	22
Figura 2 - 3 Estructura de una RIA.....	28
Figura 2 - 4 Arquitectura de RIA en plataforma Adobe	31
Figura 2 - 5 Componentes de Tecnología JavaFX.....	32
Figura 3 - 1 Método de <i>streaming</i> con video en distintas velocidades	38
Figura 3 - 2 Método de <i>streaming</i> con escalabilidad de video	38
Figura 3 - 3 QuickCam® Orbit AF de Logitech.....	46
Figura 4 - 1 Arquitectura de la solución.....	52
Figura 4 - 2 Consola de Administración	54
Figura 4 - 3 Bosquejo aplicación de video	56
Figura 4 - 4 Bosquejo de la aplicación Chat.....	57
Figura 5 - 1 Interfaz de Flash Media Live Encoder.....	63
Figura 5 - 2 Primera aplicación desarrollada.....	64
Figura 5 - 3 Diagrama de flujo de aplicación Chat	67
Figura 5 - 4 Aplicación Chat.....	68
Figura 5 - 5 Interfaz de Video con múltiples cámaras 1	70
Figura 5 - 6 Interfaz de Video con múltiples cámaras 2	71
Figura 5 - 7 Interfaz integrada de video y Chat	73
Figura 5 - 8 Guía de laboratorio en formato swf.....	74
Figura 5 - 9 Interfaz y guía dentro de una página web	75
Figura 5 - 10 Interfaz evaluada utilizando Internet Explorer.....	76
Figura 5 - 11 Interfaz evaluada utilizando Mozilla Firefox	77
Figura 5 - 12 Interfaz evaluada utilizando Google Chrome	78
Figura 5 - 13 Interfaz evaluada en Safari	78
Figura 5 - 14 Consola de administración del FMS - Conexiones Existentes....	80
Figura 5 - 15 Consola de administración del FMS - Objetos Compartidos.....	81
Figura 5 - 16 Consola de administración del FMS - <i>Streams</i> Activos	82
Figura 5 - 17 Consola de administración del FMS - Performance del Servidor	83
Figura 5 - 18 Trama de TCP capturada	85
Figura 5 - 19 Trama de RTMP capturada	85

Lista de Tablas

Tabla 1 - 1: Características de formación en red	15
Tabla 3 - 1: Principales códecs en la actualidad	39
Tabla 3 - 2: Precios de las herramientas en evaluación.....	41
Tabla 5 - 1: Análisis de velocidad de transmisión de video.....	84



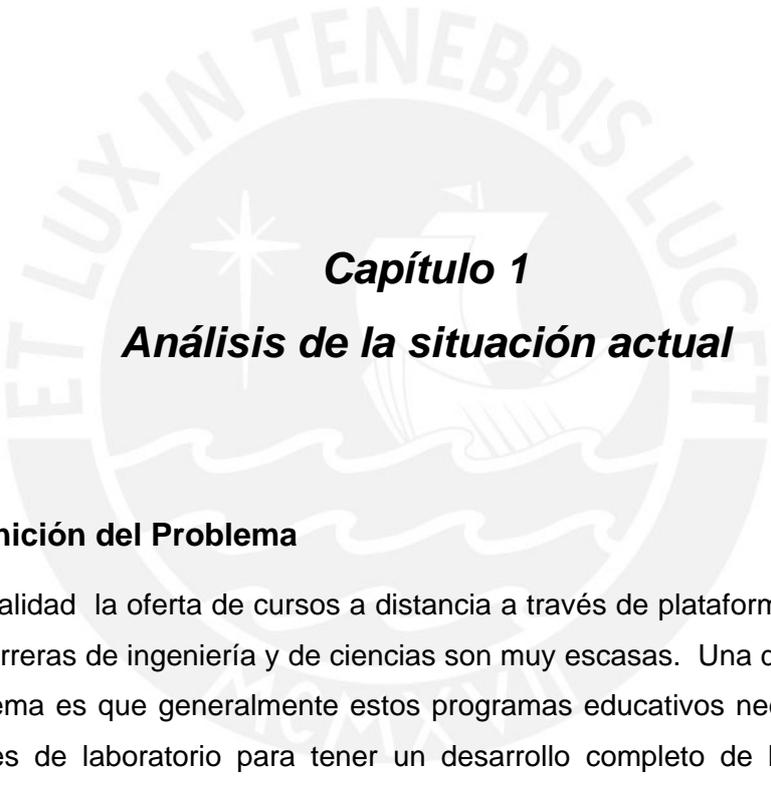
Introducción

El avance de la tecnología ha permitido en los últimos tiempos superar barreras de comunicación y acceso a la información que hace algunos años parecían inimaginables. Es por medio del Internet que se han creado vías de acceso a fuentes de información provenientes de todas partes del mundo y al mismo tiempo se han desarrollado nuevos canales de comunicación. Las universidades y otras entidades dedicadas a la educación, no quedándose atrás con el desarrollo tecnológico y haciendo uso de los recursos existentes del Internet, han desarrollado servicios que permiten el acceso a los programas educativos de manera remota, es así que se conciben las plataformas educativas virtuales o de *e-learning*.

Estas nuevas modalidades de enseñanza llevan la educación a nuevos horizontes. El contar con cursos a distancias permiten que tanto los alumnos como los profesores puedan encontrarse en cualquier parte del mundo y llevar o dictar programas académicos universitarios sin la necesidad de abandonar la comodidad de su hogar. La tendencia actual indica que cada año surgen más programas de esta línea que son impartidos a través de estas nuevas plataformas, sin embargo, la mayoría de estas tienden a hacer en carreras de humanidades y afines, dejando las de ingeniería y ciencias en una menor proporción [PUC2009].

Una de las principales razones de no brindar cursos a distancia en estos programas universitarios es la necesidad del uso de laboratorios. En las carreras de ingeniería es fundamental contar con laboratorios equipados para que así los cursos que se enseñen sean mejor comprendidos por los alumnos. Es a través de las sesiones realizadas en los laboratorios que los conceptos vistos en clase son más entendidos ya que permite al estudiante realizar una experiencia con equipos, instrumentación y mediciones reales.

No obstante, no todas las universidades del Perú cuentan con ambientes necesarios y aptos para el desarrollo de los laboratorios. Es común encontrar que estos ambientes no se encuentran debidamente equipados con la instrumentación necesaria y además, que solo se cuenten con pocos equipos para desarrollar una experiencia y se tenga una gran cantidad de alumnos que tengan la necesidad de acceder estas herramientas.



Capítulo 1

Análisis de la situación actual

1.1 Definición del Problema

En la actualidad la oferta de cursos a distancia a través de plataformas de *e-learning* para las carreras de ingeniería y de ciencias son muy escasas. Una de las razones de este problema es que generalmente estos programas educativos necesitan apoyarse de sesiones de laboratorio para tener un desarrollo completo de los contenidos a tratar.

Esta limitación no solo se encuentra en los cursos a distancia, en las modalidades presenciales el empleo de laboratorios es condicionado a la disponibilidad de estos. Las aulas de laboratorios son comúnmente compartidas por varios cursos de la misma línea, asimismo, debido a los precios en el mercado, la cantidad de equipos con los que se cuenta para estos ambientes son reducidos.

Si bien, pueden existir plataformas en las que es posible simular experiencias de laboratorios por medio de software especializado, como por ejemplo los emuladores, estas no dan una sensación real de lo que es el manejo de equipos de instrumentación

y medición. Es más, generalmente estas plataformas están diseñadas para el empleo de un único usuario, esto limita la utilización de las herramientas y no brinda una interfaz de colaboración e interacción multiusuario.

1.2 Justificación

Ante esta situación la idea de implementar un laboratorio virtual que permita el acceso, a través de Internet, a un laboratorio real y debidamente equipado se ve como una posible solución para subsanar los problemas y limitaciones de la educación actual. Esta solución a su vez permite eliminar el problema del aislamiento geográfico [CAB2006] dado que cualquier universidad del Perú que cuente con un acceso a Internet apropiado sería capaz de acceder a este servicio y contar con equipos que sus instalaciones no cuentan.

Este laboratorio virtual remoto traería consigo varios beneficios para la institución donde se implementa. En primer lugar, a partir de esta plataforma se podrían desarrollar cursos a distancia en programas educativos orientados a la ingeniería y ciencias. Asimismo, brindaría un soporte a los cursos que actualmente se dictan ya que habría una mayor disponibilidad de los ambientes de laboratorio. Este acceso no se tendría que limitar únicamente a los alumnos de la institución ya que se podría dar entrada a terceros, especialmente a investigadores, y de esta manera dar un aporte a la sociedad.

1.3 Educación en la era tecnológica

A partir de que el uso de Internet se volvió masivo se han desarrollado plataformas que empleen las nuevas tecnologías de información y comunicación en distintos ámbitos de la vida cotidiana. Un claro ejemplo de esto lo podemos encontrar en la educación, en ella tanto alumnos, profesores y personas en general tienen acceso a recursos educativos para adquirir conocimientos a través de la Web.

Términos como *e-learning*, tele-educación, aprendizaje virtual, entre otros hacen referencia a estas nuevas modalidades de aprendizaje que utilizan la red como técnica de distribución de la información. Asimismo, nuevas metodologías y políticas educativas surgen y llegan a reestructurar las prácticas tradicionales de formación y

adquisición de conocimientos. A continuación se indicarán algunas de estas que hacen uso de las nuevas herramientas de educación [BAT2005].

- **Aprendizaje Abierto:** Es una política educacional cuya principal meta es remover las barreras de aprendizaje. Esto se refiere a que no exista la necesidad de pasar por un proceso de calificación para estudiar, asimismo esto incluye a personas con discapacidad las cuales, con determinadas herramientas, pueden llevar programas educativos a pesar de sus impedimentos. Tiene como ideal que a nadie se le debe negar el acceder un programa de aprendizaje abierto y es a través de la Internet que es posible el acceso desde cualquier parte del mundo.
- **Educación a Distancia:** Es un método educativo en el que cada estudiante proporciona su tiempo y lugar de estudio para adquirir conocimientos. En ello la presencia de un profesor no es importante dado que este puede encontrarse en otro espacio geográfico. Es de esta forma que el papel de la tecnología se hace crítico.
- **Aprendizaje Flexible:** Este método no se relaciona a una institución educativa en particular, en cambio se asocia a un grupo de individuos avocados a adquirir un conocimiento en particular. La educación a distancia se hace presente pero también es necesario las sesiones presenciales en un lugar determinado. En este caso el uso de la tecnología facilita la comunicación y el acceso a la información.

1.3.1 *e-learning*

El *e-learning* es una plataforma de enseñanza que hace uso de tecnologías basadas en Internet para proporcionar un amplio abanico de soluciones que unifican la adquisición de conocimiento, habilidades y capacidades. Esta se basa en tres criterios fundamentales los cuales determinan su correcta aplicación [ROS2001]:

- o Debe realizarse en red de manera que permita una actualización inmediata, almacenaje, recuperación, distribución y compartición de los contenidos y de la información.

- Debe llegar al usuario final a través de una computadora utilizando estándares tecnológicos de Internet
- Debe centrarse en la más amplia visión de soluciones al aprendizaje que vayan más allá de los paradigmas tradicionales de la formación.

Además de los criterios mencionados previamente existen ciertas características distintivas de esta plataforma educativa que a continuación se describirán [CAB2006].

Tabla 1 - 1: Características de formación en red

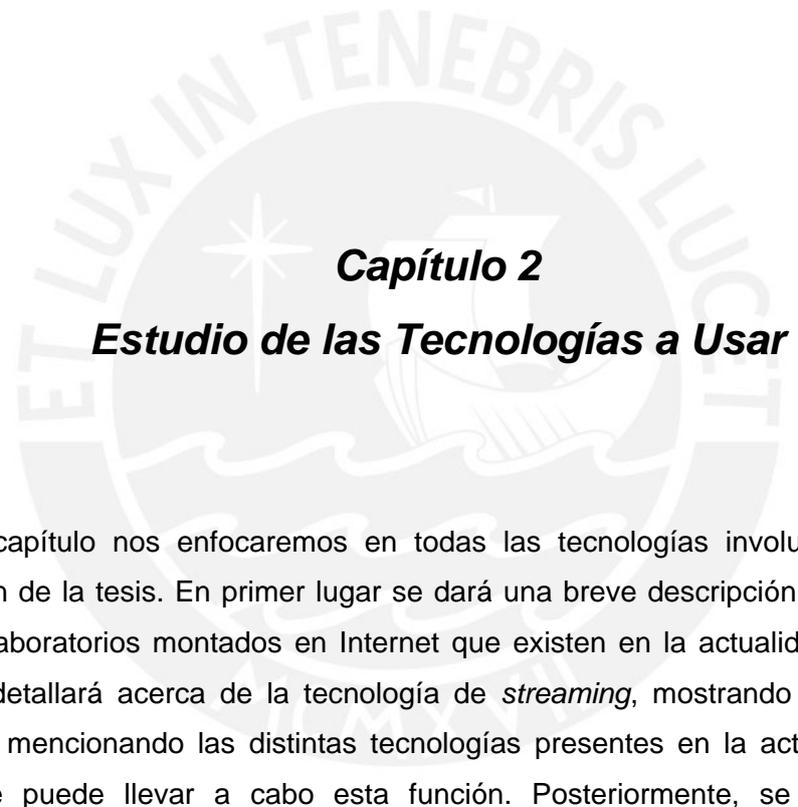
Características distintivas de la formación en red
Aprendizaje mediado por ordenador
Uso de navegadores Web para acceder a la información
Conexión profesor-alumno separados por el espacio y el tiempo
Utilización de diferentes herramientas de comunicación tanto sincrónica como asincrónica
Multimedia
Almacenaje, mantenimiento y administración de los materiales sobre un servidor web
Aprendizaje flexible
Aprendizaje muy apoyado en tutorías
Materiales digitales
Interactiva
Uso de protocolos TCP y HTTP para facilitar la comunicación entre los estudiantes y los materiales de aprendizaje, o los recursos

1.3.2 m-learning

El *m-learning* es una plataforma de educación en la cual el aprendizaje es independiente del tiempo y del espacio ya que es facilitado por medio de dispositivos móviles. Estos equipos que brindan el acceso a información son conformados por PDAs, *smartphones*, teléfonos celulares y dispositivos reproductores de media. En general, las características con la que deben cumplir un aparato que brinde esta plataforma son que deben ser pequeños, autónomos y discretos de manera que puedan acompañar a cualquier persona en cada

momento de su vida diaria y puedan ser utilizados en alguna forma de aprendizaje [MIC2009].

Del mismo modo, existen ciertas características que estos sistemas deben tener y se deben considerar al momento de diseñar. En primer lugar, debe ser ubicuo ya que el contenido debe ser accedido desde cualquier parte sin importar su ubicación. Asimismo los componentes educativos deben ser cortos y de poca duración debido a que existe la posibilidad de ocurran potenciales interrupciones que afecten la concentración. Por otra parte toda la información brindada debe ser bajo demanda ya que el usuario es el que decide cuando acceder a los contenidos que le son útiles. Cabe destacar que generalmente esta plataforma sirve de apoyo o extensión a otros programas educativos, así pues se definen mayormente como parte de un enfoque combinado, es más, es poco común encontrar que sean la principal fuente de distribución para un tema en específico. Finalmente, estos sistemas tienden a ser colaborativos, dado que los dispositivos están diseñados para habilitar la comunicación, *m-learning* puede sacar ventaja de esto y permitir el intercambio de información entre estudiantes e instructores [KIN2009].



Capítulo 2

Estudio de las Tecnologías a Usar

En este capítulo nos enfocaremos en todas las tecnologías involucradas para la realización de la tesis. En primer lugar se dará una breve descripción de los distintos tipos de laboratorios montados en Internet que existen en la actualidad. Seguido de esto, se detallará acerca de la tecnología de *streaming*, mostrando su arquitectura general y mencionando las distintas tecnologías presentes en la actualidad con las cuales se puede llevar a cabo esta función. Posteriormente, se expondrán los protocolos de Internet implicados en la realización de un *streaming* incluyendo libres y de propietario. Finalmente, se dará una explicación de las aplicaciones de Internet enriquecidas (RIA), las cuales permitirán la elaboración de la interfaz multimedia y mencionaremos las principales tecnologías, presentes en la actualidad, que permiten el desarrollo de estas aplicaciones Web.

2.1 Laboratorios sobre Internet

Los laboratorios en línea permiten llevar las sesiones y practicas fuera de los ambientes habituales donde se realizan. Estos se basan en programas informáticos y específicamente en la Internet para llevar a cabo las experiencia. Estas permiten el desarrollo de actividades técnicas dentro de los cursos de manera que se pueda ampliar la experiencia de los programas educativos. En la actualidad existen diversos laboratorios implementados en universidades de todo el mundo como el MIT en Estados Unidos [MIT2009], la UNED con su AutomatL@abs [UNE2009] y la Universidad de Deusto en España, entre otras. Todas estas han sido desarrolladas con fines distintos sin embargo, el objetivo principal de estas plataformas es llevar los laboratorios a cualquier lugar, de modo que el estudiante tiene la opción de contar con más tiempo para llevar a cabo sus experiencias [GAR2002].

2.1.1 Laboratorio virtual

Son escenarios donde los alumnos realizan una experiencia de laboratorio a través de simuladores, emuladores, generadores de señales, etc. Toda sesión se lleva a cabo por medio de software, por ende los usuarios no tienen contacto alguno con equipos de laboratorio y hardware en general. Programas como MATLAB ® y Mathematica ® son comúnmente usadas en estas plataformas y han permitido la difusión de estos especialmente sobre Internet lo que permite el acceso y disponibilidad a una gran cantidad de usuarios [GAR2002].

2.1.2 Laboratorio remoto

En las sesiones en laboratorios remotos los usuarios de estos sistemas tienen la facilidad de entrar en contacto con hardware y equipos reales a pesar de encontrarse en un distinto espacio geográfico. Esto es posible a través del protocolo TCP/IP de manera que los estudiantes envían señales de control a las herramientas y pueden observar los resultados por medio de la reproducción de video de cámaras que se encuentran instaladas dentro del laboratorio. A continuación en la figura 2-1 se podrá apreciar un esquema de la arquitectura de un laboratorio remoto.

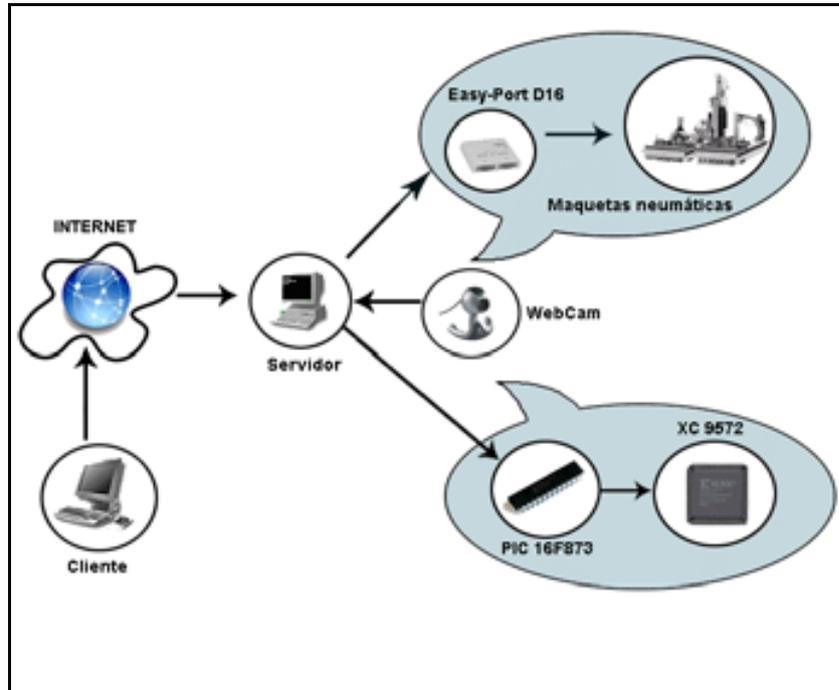


Figura 2 - 1 Arquitectura de laboratorio remoto
Fuente: [GAR2002]

2.1.3 Laboratorio remoto virtual

Es la solución que se propone en esta tesis, este sistema adopta las funcionalidades y ventajas de tanto los laboratorios virtuales como remotos. Al igual que en el laboratorio remoto por medio de esta plataforma el alumno entra en contacto directo con los equipos y hardware presente en el ambiente donde se realizan las experiencias enviando señales de control y observando los resultados en su PC. Asimismo permite que se desarrollen sesiones multiusuarios donde un grupo de estudiantes pueden realizar la misma experiencia simultáneamente, comunicándose e interactuando con las funcionalidades del sistema.

2.2 Tecnologías de Streaming

El pasado 20 de Enero del 2009 el portal de Internet de la cadena internacional de noticias CNN transmitió en vivo y en directo la toma de mando de Barack Obama a más de veintisiete millones de conexiones alrededor del mundo [ECB2009]. Del mismo

modo, el 20 de Octubre del 2009 más de diez millones de personas siguieron en directo, a través de Youtube, la transmisión en vivo del concierto realizado por la banda irlandesa U2 en el Rose Bowl de la ciudad de Los Ángeles [ECU2009]. Eventos como estos con tal cantidad masiva de audiencia dispersa en todo el mundo se llevan a cabo gracias a las tecnologías de *Streaming*.

El Video *Streaming* es el proceso de enviar contenidos de audio y video a través de una red o Internet, a un dispositivo o aplicación de reproducción para su inmediata visualización. Estos contenidos digitales pueden encontrarse en vivo o bajo demanda y hacen uso de la tecnología TCP/IP para el transporte y distribución de la información. Todo material que se requiera reproducir parte de un servidor de video el cual se encarga de la transmisión de la información a la red y luego una aplicación cliente se hace cargo de la recepción y reproducción. Cabe destacar que para una correcta emisión de los contenidos es necesario que se de a cabo una reproducción continua con poca espera, bajo retardo y sin interrupciones [AUS2005].

El término *Streaming* en si es muy genérico ya que cubre un grupo de distintas tecnologías, las más comunes que encontramos en la actualidad se describirán brevemente a continuación [SIM2008].

- ***True Streaming***: La señal de video llega en tiempo real y es reproducida automáticamente, con esto no se requiere descarga. Por ejemplo, si un video tiene una duración de dos minutos entonces el tiempo de entrega del contenido debe ser igualmente de dos minutos. Es indispensable, no obstante, que la red en la que se transmita el contenido tenga un ancho de banda suficiente para que cada paquete que transporta el video llegue en su momento y, de esta forma, la visualización se haga fluida. Asimismo, otro requisito del sistema es la necesidad de utilizar protocolos apropiados como RTP y RTSP, los cuales más adelante se tocarán.
- ***Download and Play***: Como su nombre lo indica en esta tecnología el video requerido primero es descargado completamente en el dispositivo de reproducción el cual se encarga de decodificarlo y mostrarlo. Al igual que las páginas Web, utiliza los mismos protocolos tanto HTTP, FTP encapsulados sobre TCP, es este caso el tiempo de espera depende del

tamaño del video deseado y del ancho de banda del usuario. Sin embargo dado que implica la descarga total de los contenido previamente a su visualización esta tecnología no es apta para contenido en tiempo real.

- ***Progressive Download and Play***: Es un variación de la tecnología mencionada previamente, es esta ni bien es descargado un segmento del video se inicia la reproducción y simultáneamente se continúa descargado el contenido. Es común encontrar en reproductores de este sistema que se encuentra en estado de “*Buffering*” lo que significa que se encuentra en plena descarga antes de su reproducción. Esto hace posible una visualización más continua y menor tiempo de espera.

En el escenario en el que se propone implementar la tesis la tecnología a emplear será la de *True Streaming* dado que lo que se desea mostrar son equipos en tiempo real donde se puedan apreciar los resultados al instante. Esto implica que se deberá tomar el cuidado necesario con el manejo del ancho de banda de modo que la reproducción en las PCs de los usuarios sea continua y fluida.

2.2.1 Arquitectura de la tecnología *Streaming*

La arquitectura de un sistema de *streaming* está compuesto por cuatro componentes,

- La captura, codificación y compresión
- El servidor
- La distribución y entrega
- El reproductor

En la figura 2-2 se puede apreciar la estructura del sistema.

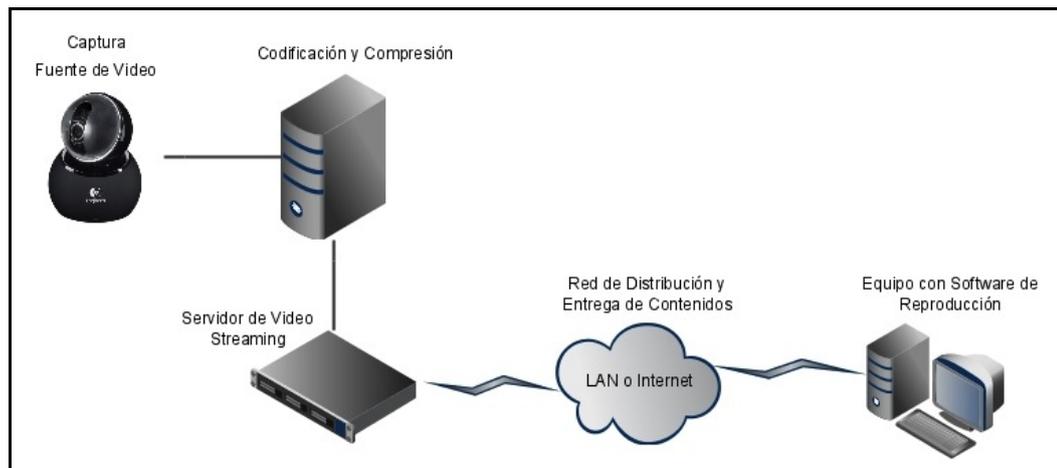


Figura 2 - 2 Arquitectura de un Sistema de *Streaming*
Fuente: Elaboración Propia

La captura, codificación y compresión implica el proceso del contenido por una serie de etapas las cuales permiten que sean aptas para su distribución por un sistema de *streaming*. En primer lugar se realiza la obtención de la señal de video por medio de filmadoras, cámaras de foto o cámaras Web, estas convierten las señales analógicas a un formato digital. Seguido de esto entra la parte de la compresión, la cual mediante un algoritmo especializado, denominado *códec*, se procede a la codificación y la compresión del contenido de modo que se pueda reducir su tamaño, su resolución de imagen y su tasa de *stream* para que de esta manera sea apto para transmitirlo por el sistema. Una vez completado estas etapas el video puede ser transferido al servidor de *streaming* [SIM2008].

El servidor de *streaming* es el responsable de la distribución de los contenidos hacia los usuarios. Ante un pedido de visualización este adapta el video obtenido y crea los flujos contenedores de la información que son enviados en forma de *unicast* o *multicast*, que en otras palabras significa que la data es enviada a un único usuario o a un grupo de ellos. Una de las funciones del servidor que le cuesta mayor trabajo de procesamiento es la creación de los paquetes para cada flujo de salida en tiempo real, este debe hacer las cabeceras de los paquetes a enviar con la correcta dirección de destino. Asimismo, el servidor tiene la responsabilidad de crear flujos que permitan que el movimiento de los paquetes sea regular y estable para que no existan demoras ni cortes en la visualización, esto implica que la tasa de transmisión iguale a la tasa requerida

por el reproductor. Finalmente otra característica de este componente es la opción de poder controlar la tasa en la que se transfiere los contenidos, así pues el servidor es capaz de disminuir la velocidad y la calidad del contenido si este detecta problemas en la red de modo que la visualización no se detenga y se mantenga el flujo de información [SIM2008].

La distribución y entrega de los videos se realiza a través de una red basada en IP, ya sea por medio de una red local o hacia Internet, sin embargo dada los requerimientos del envío de un video en forma de *streaming* es necesario mantener un control de ciertos parámetros que pueden afectar la visualización de los contenidos. En general los parámetros que se deben tener en cuenta son la tasa de pérdida de paquetes, el retraso y la variación del retraso (*jitter*). Estos son más críticos cuando se envían los paquetes hacia Internet ya que no cuenta con calidad de servicio y no se da prioridad a los diferentes tipos de servicios. En una red local no se presentaría mucho estos problemas sin embargo es necesario la configuración de la red para habilitar y facilitar el envío de *streams* [SIM2008].

El reproductor del contenido generalmente reside dentro de una PC aunque también puede encontrarse en un dispositivo especializado. Este tiene la función de aceptar la información entrante y convertirla en una señal de video apta para su visualización. Asimismo también debe realizar una serie de actividades para que la reproducción de los contenidos sea correcta. En primer lugar, tiene que reagrupar la información encapsulada en los paquetes recibidos y resincronizar el video del flujo de información entrante. Por otro lado, debe descomprimir la señal recibida y crear la imagen para su visualización, generalmente esta es la tarea que le implica mayor trabajo de procesamiento aunque depende del método de comprensión y del tamaño de la imagen [SIM2008].

2.2.2 Flash

Flash Player es un software multiplataforma que permite la visualización de contenidos multimedia mediante programas navegadores de Web elaborado por Macromedia y desarrollado actualmente por Adobe luego de su adquisición. Este reproduce archivos con la extensión *swf* los cuales son elaborados con las

herramienta de desarrollo Adobe Flash, Adobe Flex o programas de terceros. Asimismo se basa en gráficos vectoriales escalables (SVG), un lenguaje nativo de script llamado ActionScript y una transmisión de *streaming* bidireccional de audio y video que se conecta a al servidor Flash Media Server. En si se puede denominar a Flash Player como la máquina virtual en la que se monta el entorno desarrollado en Flash, así pues este reproductor no solo se avoca en la reproducción de videos ya que también ejecuta toda aplicación desarrollada en Flash. Cabe destacar que actualmente se encuentra instalado en el 98% de los usuarios de Internet en todo el mundo y debido a que es multiplataforma se puede encontrar en todo sistema operativo existente, incluyendo dispositivos móviles [FLA2009].

2.2.3 Windows Media

Microsoft desarrollo Windows Media para permitir que archivos de audio y video puedan ser reproducidos en computadoras que tengan como sistema operativo a Windows. No obstante esta aplicación también está habilitada para la visualización a través de navegadores Web por medio de *plug-in's*. Para la reproducción de contenidos por medio de *streaming* se llevaron a cabo ciertos formatos especiales capaces de soportar esta tecnología. Uno de ellos es **Windows Advance System Format** (.asf) el cual es un formato de archivo diseñado especialmente para el transporte y almacenamiento de contenido destinado a aplicaciones de *streaming*. Asimismo también esta **Windows Media Video** (.wmv) que es un archivo que contiene codificado señales de audio y video comprimidas y el formato ASF. Por último encontramos **Windows Active Stream Redirector** (.asx) el cual conecta una página Web a un archivo de reproducción, este meta-archivo contiene la dirección de uno o más video localizados en el servidor [SIM2008].

2.2.4 Quicktime

Quicktime es el sistema de medios de *streaming* multiplataforma elaborado por Apple Computers, en un principio esta aplicación fue desarrollada para la reproducción de media proveniente de CDs. Sin embargo, en la actualidad se ha convertido en un formato muy utilizado y descargado tanto en computadoras con

Windows como con Mac. A diferencia de las tecnologías presentadas anteriormente Quicktime no utiliza formatos propias para el *streaming* en cambio emplea *códecs* de estándar tales como MPEG-4 y H.264 para la compresión y reproducción de los contenidos [SIM2008].

2.2.5 Silverlight

El desarrollo de Silverlight llevado a cabo por Microsoft fue con el objetivo de competir directamente con Adobe Flash Player. Este se encuentra desarrollado bajo la plataforma .NET y al igual que Flash también se basa en gráficos vectoriales escalable (SVG) sin embargo añade funciones más avanzadas como conectividad a servidores y una excelente reproducción de video [COL2007]. Asimismo trabaja con los formatos de video existentes de Microsoft mencionados previamente, no necesita ser codificado con un formato propietario y utiliza la última versión de Windows Server para el *streaming*. Es una aplicación multiplataforma que permite su utilización fuera de Windows, en navegadores de Internet como Firefox y Safari, y soporta la reproducción de aplicaciones creadas con lenguajes de programación dinámicos como Python y Ruby [TAF2007].

2.3 Protocolos de *Streaming*

Como ya se menciona, en toda transmisión de *streaming* es necesario que los contenidos viajen por medio de una red TCP/IP para que un usuario pueda visualizarlos. Para ello se requiere que la información a enviar tiene que estar encapsulada en paquetes de modo que sea entendida y transmitida. Existen muchos protocolos destinados para la realización de *streaming* los cuales pertenecen a distintas capas del modelo OSI. En primer lugar, en la capa 2 de red encontramos el protocolo IP, luego para la capa 3 de transporte se utiliza tanto TCP como UDP, siendo la primera más segura que la segunda, pero en UDP al generarse menos retardos la velocidad de conexión es más rápida. Del mismo modo en la capa 3 se utiliza el protocolo RTP (Real Time Protocol). Finalmente en la capa de aplicación existen dos protocolos que son los más utilizados para esta tecnología: RTSP y RTMP, a continuación se describirán.

2.3.1 RTSP

RTSP (*Real Time Streaming Protocol*) es un protocolo de nivel de aplicación que controla la entrega de datos en tiempo real. Este proporciona una plataforma de control para la entrega de datos bajo demanda en tiempo real tales como audio y video, las cuales pueden provenir de fuentes en vivo o almacenadas en un servidor. Tiene como objetivo controlar múltiples datos de distintas sesiones de *streaming*, proveer los medios para decidir los canales de entrega ya sea UDP o TCP y decidir qué mecanismos tomar basados en RTP.

En su funcionamiento el protocolo únicamente define sesiones las cuales son mantenidas por un servidor a través de un identificador. Este no se encuentra ligado a una única conexión de nivel de transporte como TCP, en cambio durante una sesión RTSP un cliente puede abrir y cerrar múltiples conexiones de transporte, por medio de TCP o UDP, para que el servidor envíe solicitudes RTSP.

Los *streams* controlados por RTSP pueden usar RTP pero la operación de este protocolo no depende del mecanismo de transporte usado para llevar los medios de comunicación. Asimismo, este protocolo es intencionalmente similar en sintaxis y operación al protocolo HTTP/1.1. Finalmente, RTSP soporta las siguientes operaciones [JAV2007] :

- Recuperación de contenidos desde un servidor
- Invitación a un servidor de medios a una conferencia
- Adición de contenidos a una existente presentación

2.3.2 RTMP

RTMP (*Real-Time Messaging Protocol*) es un protocolo de aplicación propietario de alta performance desarrollado por Macromedia, actualmente propiedad de Adobe Systems y diseñado para la transmisión de audio, video y datos entre las plataformas tecnológicas de Adobe Flash, en especial entre un reproductor Flash Player y el servidor Flash Media Server. Este permite el envío de una amplia gama de aplicaciones montadas en Adobe Flash como la transmisión de contenidos, video conferencias, *streaming* y envío de mensajes multimedia sobre

Internet. El puerto predeterminado de conexión que utiliza es el 1935 y únicamente se monta sobre el protocolo de transporte TCP, así pues no utiliza UDP [CAS2009].

A través de TCP el protocolo mantiene una conexión permanente y segura en una comunicación en tiempo real. Por otro lado, para garantizar la entrega continua de video y audio, aun cuando se envían grandes cantidades de información, el protocolo divide los videos y datos en fragmentos. Los tamaños de los fragmentos pueden ser variables dado que dependen de la negociación entre el cliente y el servidor, sin embargo los tamaños predeterminados son de 128 bytes para el video y 64 bytes para el audio [CAS2009].

Para el envío y recepción de paquetes RTMP puede emplear varios canales que operan de manera independiente uno de otro. Por ejemplo en una conexión puede existir un canal exclusivo para el flujo de video, otro canal para el flujo de audio, un canal para el envío de mensajes de control, entre otros. Así pues, durante una sesión típica de RTMP, varios canales pueden encontrarse activos simultáneamente en un momento determinado [RTM2009].

Finalmente, un paquete de RTMP esta compuesto por una cabecera fija y una carga útil variable. La cabecera puede ser de cuatro tamaños 12, 8, 4 o 1 bytes y como se menciono previamente la carga útil tiene un tamaño predeterminado de 128 bytes [RTM2009].

2.4 Aplicaciones de Internet Enriquecidas

Una aplicación de Internet enriquecida (RIA) es una nueva clase de aplicación Web que tiene como objetivo incrementar y mejorar las opciones y capacidades de las tradicionales páginas Web ya que tienen muchas de las características de las aplicaciones de escritorio [NOD2005]. Estas permiten a los usuarios:

- Nuevas experiencia en audio, video y comunicación
- Ausencia de la necesidad de refrescar la pagina Web
- Soporte *online* y *offline*
- Aplicaciones más complejas
- Mayor interactividad y velocidad de respuesta

Son desarrolladas por plataformas más modernas pero su complejidad de diseño no difiere mucho con las actuales aplicaciones Web. Asimismo, son aplicaciones multiplataforma compatibles con todos los navegadores existentes ya que para su ejecución solo es necesaria la instalación de un *plug-in*.

El funcionamiento de una RIA se basa en los siguientes principios [FAR2005]:

- Utilización de modelo contenedor en el lado del cliente que almacene la parte gráfica de la aplicación de manera que se reduzca la comunicación con el servidor y el refresco de la página.
- Desarrollo de aplicaciones a través de navegadores Web.
- Uso de lenguaje basado en XML para el diseño de la interfaz de usuario.

Por otro lado, la estructura general de una RIA no difiere mucho a la de una aplicación Web tradicional, a continuación en la figura 2-3 se mostrará una estructura general.



Figura 2 - 3 Estructura de una RIA
Fuente: [FAR2005]

En primer lugar se encuentra el navegador Web el cual contiene el *plug-in* que se necesita para montar la aplicación e interpretar los datos que llegan a esta. Como se menciono previamente parte de los gráficos se almacenan dentro de la PC para aumentar la velocidad de repuesta y evitar las recargas.

En segundo lugar se tiene el controlador de la aplicación y el *Gateway*. El primero se encarga de la interacción con el cliente para brindarle los servicios solicitados. Por otro lado, el *Gateway* tiene la función de transformar los datos que maneja la aplicación de modo que puedan ser entendidos por el navegador del cliente.

El elemento que sigue es el servidor donde se almacena la aplicación RIA, este tiene como objetivo compilar la aplicación Web y permitir la comunicación de esta con otras

aplicaciones. Asimismo, es la que se encarga de obtener los datos que la aplicación necesita para ejecutarse.

Finalmente, los datos necesarios por la RIA son almacenados generalmente en una base de datos, esta contiene toda la información relacionada con los servicios ofrecidos. No obstante, de acuerdo a la aplicación desarrollada, no siempre se requiere el uso de este elemento.

2.4.1 Adobe Flash

Adobe Flash es una plataforma multimedia gráfica elaborada para el diseño de animaciones y componentes interactivas dentro de las páginas Web. Es comúnmente utilizada para la creación de animaciones, anuncios publicitarios, componentes de integración de audio y video y actualmente para el desarrollo de aplicaciones de Internet enriquecidas (RIA).

Esta plataforma manipula imágenes en mapas de bits y gráficos vectoriales de modo que puedan escalarse a cualquier tamaño sin perder claridad y calidad. Asimismo se desarrolla bajo el lenguaje de script *ActionScript* y su ejecución y visualización se lleva a cabo por medio de *Flash Media Player*. Por otra parte, los archivos contenedores de Flash tienen el formato de extensión *.swf* los cuales pueden ser un objeto dentro de una página Web o un elemento a reproducir en un *Flash Player*.

Para la elaboración de RIAs, Flash ofrece al desarrollador un nuevo juego de herramientas de programación que permiten mayor interactividad mejorando así el desempeño en las páginas Web al eliminar los problemas presentes en el diseño con HTML/JavaScript. A su vez tiene una mejor capacidad para interactuar de forma rápida y sin problemas con tecnologías de transferencia de datos como los archivos XML y tiene compatibilidad de trabajo con entornos como Java y .NET.

Junto con la creación de RIAs se desarrolló ActionScript 2.0, una actualización de ActionScript 1.0 que estaba destinada principalmente a la elaboración de animaciones. Este nuevo lenguaje de programación que se encuentra

parcialmente basado en objetos era necesario para el diseño de las aplicaciones enriquecidas. Sin embargo, el uso de este nuevo lenguaje implicaba no solo conocer las reglas de sintaxis de la programación en objetos sino también tener conocimientos del entorno de Flash y además las herramientas para la depuración eran muy ineficientes. Para abordar estos problemas se elaboro Flex [ADO2009].

2.4.2 Adobe Flex

Adobe Flex fue desarrollado en un principio por Macromedia en el 2004, antes que fuera adquirido por Adobe Systems, con el objetivo de subsanar los problemas presentados con Flash y el lenguaje ActionScript 2.0. Esta es una plataforma exclusiva para el diseño de RIAs con un entorno de programación más tradicional comparado al de Flash. Este también introdujo una nueva actualización, ActionScript 3.0 completamente basado en objetos y muy similar a C++ y Java. Así pues, con este lenguaje es posible desarrollar aplicación únicamente a partir de programación en código sin la necesidad de utilizar el entorno de Flash [ADO2009].

Fundamentalmente, Flex es un servidor de presentación, este se monta sobre cualquier servidor de aplicaciones, como por ejemplo JSP, PHP ASPX, y toma el lugar de HTML/JavaScript para la presentación de datos. De esta manera la visualización de la información se realiza utilizando un archivo SWF obteniendo una mayor capacidad de dinamismo.

Por otra parte, para el diseño Web esta plataforma utiliza una basada en Eclipse, el cual es un IDE (*integrated development enviroment* o entorno de desarrollo integrado) gratis de libre disponibilidad y ampliamente usado que permite al programador trabajar con múltiples entornos de desarrollo simultáneamente [BRO2008]. Así pues el SDK de esta plataforma es libre, pero el entorno de desarrollo elaborado por Adobe si tiene costo.

En la figura 2-4 se muestra el papel que tiene Flex dentro de una estructura de una aplicación enriquecida de Internet bajo la plataforma de Adobe. Como se aprecia este se encuentra en el cliente y permite la comunicación entre el servidor de aplicación y el entorno del usuario. Cabe destacar también que para

la ejecución es necesario tener *Flash Player* o Adobe AIR que es utilizada para entornos de escritorio sin la necesidad de un navegador Web. Por otra parte cabe mencionar que esta solución es multiplataforma por lo que funciona sobre cualquier sistema operativo existente en la actualidad. Finalmente, como se puede observar Flex se monta sobre los servidores que Adobe tiene, por ejemplo *Flash Media Server*, y sobre demás servidores Web existentes.

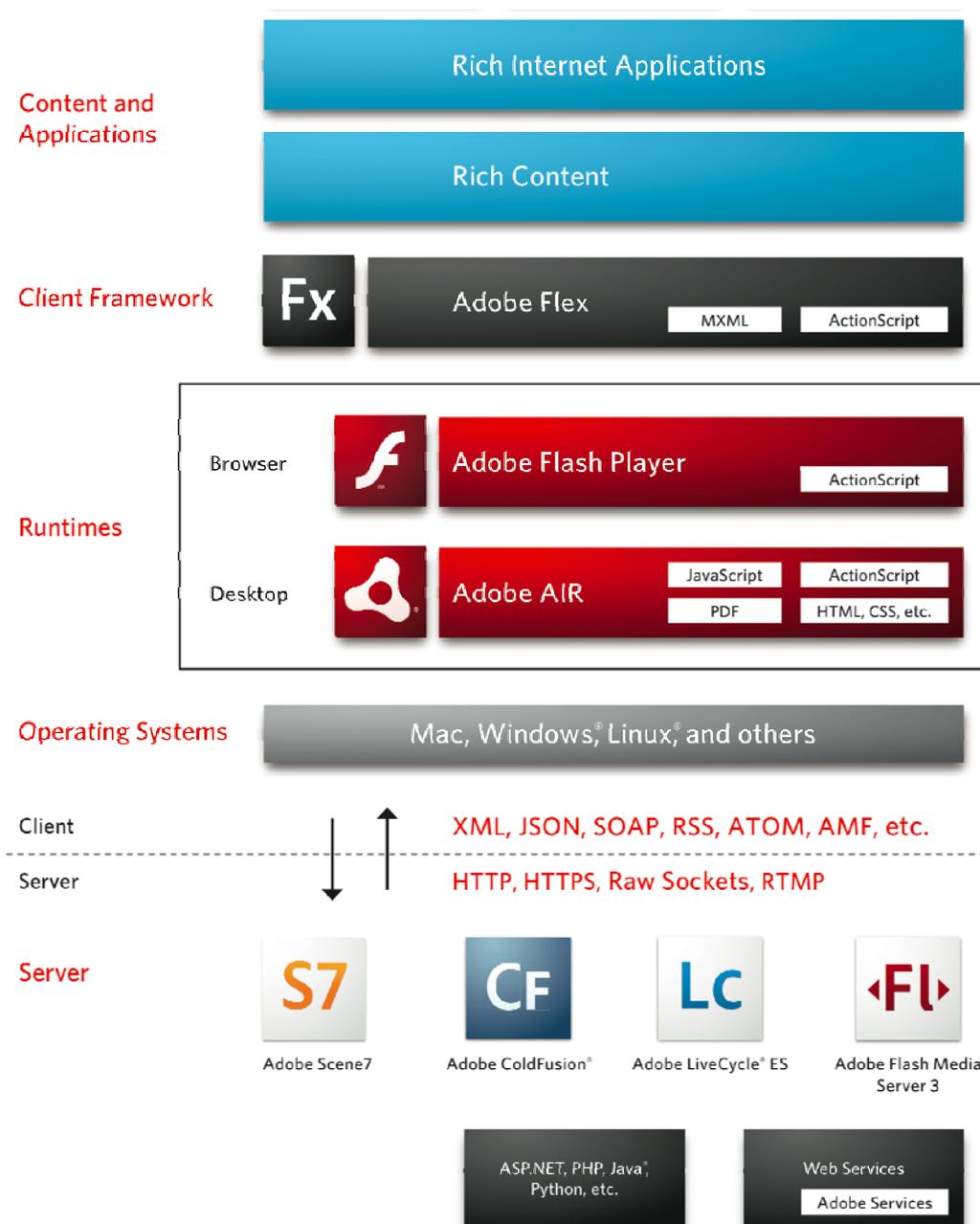


Figura 2 - 4 Arquitectura de RIA en plataforma Adobe
Fuente: [ADO2009]

2.4.3 JavaFX

JavaFX es la plataforma de desarrollo de aplicaciones de Internet enriquecidas que ofrece Sun Microsystems para la ejecución en múltiples plataformas y dispositivos. Esta se encuentra completamente integrado con *Java Runtime* y para su ejecución tanto en navegadores Web como en el escritorio solo es necesario contar con *Java Runtime Environment* [CLA2009].

Las aplicaciones de JavaFX se desarrollan a través de JavaFX Script un lenguaje orientado a objetos que permite una programación con herramientas visuales. Asimismo es capaz de soportar múltiples lenguajes de script tales como ECMA y XML, junto con las librerías nativas de Java, y proporciona una sintaxis declarativa sencilla para la integración de componentes interactivos como gráficos, animaciones, audio, video y textos enriquecidos [JAV2009].

Para el desarrollo de estas aplicaciones se utiliza como IDE a NetBeans 6.5.1, este programa provee un entorno integrado para crear, previsualizar y depurar aplicaciones de JavaFX. Asimismo cuenta con editores de arrastre simple y copiado y pegado que facilitan el diseño y el entorno de presentación de estas RIAs.

La figura 2-5 presenta un gráfico con todas las componentes que forman parte de la tecnología JavaFX.

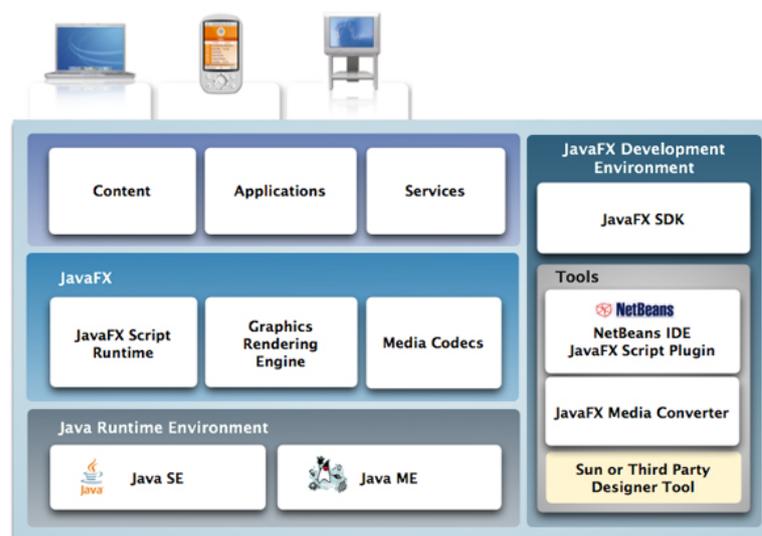


Figura 2 - 5 Componentes de Tecnología JavaFX
Fuente: [TAR2009]

2.4.4 Microsoft Silverlight

Previamente se ha mencionado las características generales de un *streaming* por medio de Silverlight, esta sección se enfoca en el desarrollo de RIAs por medio de esta tecnología. Silverlight es una plataforma Web diseñada para aplicaciones de Internet enriquecidas (RIA) y medios multimedia elaborada por Microsoft. Esta combina múltiples tecnologías en una plataforma de desarrollo que permite elegir entre distintas herramientas y lenguajes de programación para la elaboración de aplicaciones enriquecidas.

Esta tecnología se basa principalmente en el *framework* .NET, y es a través de .NET CLR (*Common Language Runtime*) que permite al programador desarrollar aplicaciones tanto en C# como en Visual Basic, ambos lenguajes de esta plataforma. Del mismo modo, cuenta con los siguientes softwares de desarrollo que permiten elaborar RIAs de última generación. Visual Studio para la programación y depuración, *Expression Blend* y *Expression Design* para la creación de gráficos vectoriales y animaciones y *Expression Media Encoder* para la codificación y elaboración de reproductores de medios [SIL2009].

Cabe destacar que Silverlight utiliza una sintaxis declarativa denominada XAML (*Extensible Application Markup Language*) para la creación de elementos visuales, esta es una extensión del formato XML elaborado por Microsoft especialmente para ser usado en Silverlight. Este lenguaje trabaja en conjunto con los mencionados previamente, sin embargo se enfoca en la presentación de las aplicaciones mas no en la lógica de éstas, así pues cada lenguaje utilizado en una RIA es independiente del otro [MCH2009].

Las aplicaciones Web desarrolladas en Silverlight son capaces de ejecutarse sobre cualquier navegador Web e incluso en otros sistemas operativos distintos a Microsoft tales como Mac OS e inclusive Linux, bajo el proyecto Moonlight. Para ello solo es necesario contar con el *plug-in* que contiene el entorno de ejecución, el cual también realiza la reproducción de *streaming* con esta tecnología.

2.4.5 HTML5

HTML 5 o *Web Applications 1.0* es la nueva generación del estándar HTML (*Hypertext Markup Language*) desarrollado por WHATWG (*Web Hypertext Application Technology Working Group*) y por el W3C (*World Wide Web Consortium*) con el objetivo brindar una estandarización para el desarrollo de aplicaciones Web. Ante la existencia de varias tecnologías propietarias de elaboración de RIAs y el requerimiento de la instalación de *plug-in's* para su ejecución, diferentes empresas del rubro de Internet, tales como Apple, Mozilla y Opera, se juntaron para elaborar los principios básicos de lo que será un nuevo estándar que sirva como plataforma de desarrollo para esta clase de aplicaciones Web.

El primer *draft* de este estándar fue lanzado a inicios del 2008 donde el cambio más resaltante son la creación de nuevas interfaces de programación, denominadas APIs (*Application Programming Interfaces*), que incluyen la edición de gráficos, edición de documentos, *drag and drop*, almacenamiento de archivos y reproducción de video [CDE2009].

En sí HTML5 marca una evolución tanto para HTML4 como para XHTML1, con respecto a la semántica de este estándar se han realizado cambios de las versiones anteriores ya que se desea uniformizar la programación con HTML dejando de lado su sintaxis de SGML (*Standard Generalized Markup Language*) para estar basado en XML. Del mismo modo elementos comunes en lenguaje de HTML se han modificado, también se han introducido nuevos elementos tales como `<nav>`, `<footer>`, `<audio>` y `<video>` que brindan nuevas funcionalidades a las páginas Web [IBM2009].

A la fecha aun no se ha lanzado la versión final de este estándar, según los grupos encargados de su elaboración tienen estimado que se encuentre finalizado para el 2012. Actualmente se cuenta con revisiones de los *drafts* que se emiten periódicamente, el más reciente, al momento de la elaboración de esta Tesis, fue publicado en Diciembre del 2009.



Capítulo 3

Análisis de la solución

Una vez realizado el estudio de las tecnologías presentes en la actualidad que van a facilitar el desarrollo de la tesis lo que a continuación se tratará es la elección de las herramientas que brindarán una solución al problema planteado en este trabajo. En primer lugar, se realizará una evaluación de los requerimientos que van a tener que poseer las herramientas de desarrollo que se utilizarán para la implementación. Luego de esto se analizarán las necesidades del laboratorio y con ello ya nos enfocaremos a la elección de los componentes. Finalmente se dará una descripción general de las características que tendrán el servidor de video y la interfaz del laboratorio.

3.1 Aspectos a considerar para la elección de las mejores herramientas que se adecue a la implementación

Luego de haber realizado un análisis de las tecnologías presentes y las herramientas disponibles se ha encontrado que existen ventajas en cada una de las estudiadas que permiten tanto al desarrollador como al usuario final hacer uso de éstas para llevar a cabo sin ninguna dificultad sus funciones. Sin embargo, es necesario realizar una evaluación de algunos aspectos y requerimientos que se deben tomar en cuenta para la elección de las herramientas que mejor se adecuen a las necesidades del sistema; de modo que se pueda llevar a cabo una implementación exitosa y los usuarios cuenten con un interfaz que permita la realización del laboratorio remotamente.

Los aspectos que se van a tener que considerar son de suma importancia ya que van a posibilitar la elección de las herramientas de manera más eficaz y rápida. Asimismo es importante considerar la relación que existe entre estas ya que de esta manera se brindará una herramienta completa lo cual es lo que se busca en este capítulo.

3.1.1 Servidor de video

Este es uno de los aspectos más importantes en tomar en cuenta para la implementación, a través del servidor de video se va a ser posible distribuir los contenidos visuales del laboratorio a los usuarios que se encuentren utilizando la aplicación. Es a través de esta herramienta que la información capturada por medio de una cámara Web o video cámara va a ser procesada, comprimida y codificada de modo que se pueda entregar a los clientes.

Un servidor de video, a diferencia de un servidor Web, mantiene abierta una conexión con un puerto específico de un cliente por el cual se envían todos los contenidos de multimedia durante todo el tiempo que el usuario lo requiera y la aplicación que se ejecuta se mantenga abierta. Así pues, este tipo de servidores son denominados *Open-Socket Server* o servidor de puerto-abierto. Por otro lado, en el caso de un servidor Web en el momento en que la información requerida llega al usuario la conexión encargada de transmitirla es cerrada, y en caso que se realice otra petición, una nueva conexión con un puerto del cliente se debe llevar a cabo [SAN2008].

Es necesario que el servidor de video a utilizar permita la opción de transmitir video en tiempo real ya que en la sesión de laboratorio los alumnos usuarios del sistema van a necesitar visualizar lo que sucede en la experiencia de manera inmediata. Es más, no solamente debe manejar el flujo de video de una única cámara sino va a tener que controlar por lo menos tres dispositivos de manera simultánea.

Existen en la actualidad diversas soluciones para la elaboración de un servidor de video. La más utilizada es Flash Media Server [ADO2009], asimismo existen otros software que brindan en su mayoría un resultado similar a ésta, Red 5 [RED2009] y Wowza Media System [WOW2009] son dos ejemplos de servidores que tienen características parecidas.

3.1.2 Ancho de banda requerido

El ancho de banda requerido es un aspecto a considerar de modo que los usuarios del sistema puedan acceder a los contenidos y visualizarlos de manera fluida sin interrupciones o cortes del video. Este además de ser un requerimiento del sistema es a su vez un requisito que los usuarios deben de tener para la reproducción de los contenidos. Al referirnos de requisito nos enfocamos en el hecho que los clientes de la aplicación deberán tener un mínimo ancho de banda para realizar una sesión en el laboratorio.

Además de ser un requerimiento del usuario es una obligación del desarrollador brindar una velocidad de conexión considerable tomando en cuenta las velocidades existentes en los hogares y centros de estudio del país. Así pues es necesario que el sistema cuente con más de una sola velocidad de modo que ante usuarios con bajo ancho de banda estos puedan observar los contenidos con una calidad de video baja y a clientes con mayor velocidad se les pueda entregar un video con mejor resolución y calidad de imagen.

En la actualidad existen alternativas para la entrega de contenidos de video *streaming* a distintas velocidades de acuerdo a las capacidades de cada usuario, estas son denominadas métodos de *streaming*. La primera alternativa es contar distintos *bit-stream* o velocidades de bits de un mismo video la cual dependiendo

de la velocidad de cada usuario el sistema elige la que mejor se adecue y se pueda enviar de manera eficiente. Esta es la solución más simple ya que no requiere de mucho procesamiento sin embargo si es necesario que el sistema tenga mayor capacidad de memoria debido a que tiene que almacenar un mismo contenido más de una vez dependiendo del número de velocidades que ofrezca [SUN2007].

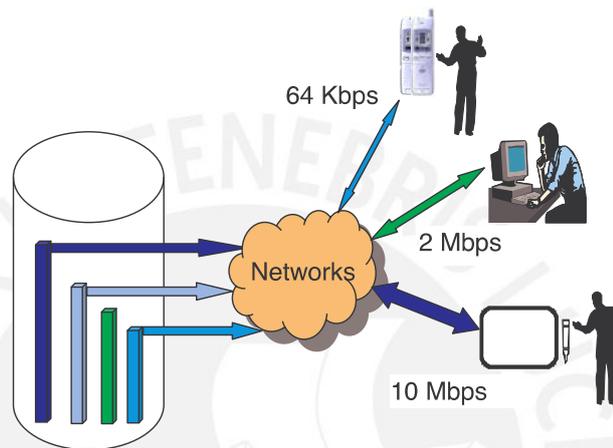


Figura 3 - 1 Método de *streaming* con video en distintas velocidades
Fuente: [SUN2007]

Por otro lado, en el segundo método de entrega solo se almacena un ejemplar del video a transmitir en alta calidad y por medio de un transductor se disminuye el *bit-stream* de modo que pueda ser enviado a usuarios con menor ancho de banda. Esta escalabilidad del video requiere mayor trabajo computacional en el servidor pero ofrece una mayor flexibilidad ya que depende de la velocidad del usuario la calidad de video que obtendrá [SUN2007].

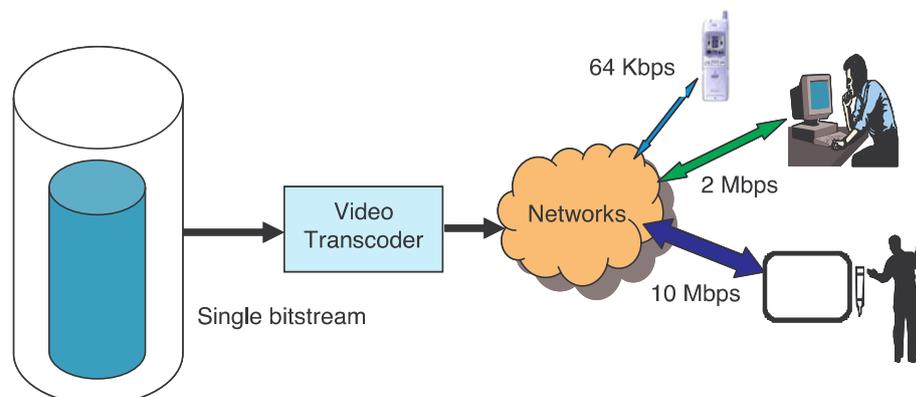


Figura 3 - 2 Método de *streaming* con escalabilidad de video
Fuente: [SUN2007]

3.1.3 Códecs usados

Los *códecs* permiten una compresión del video a reproducir por medio de algoritmos predefinidos, los cuales con el tiempo se han llegado a convertir en estándares. Estos algoritmos detectan las semejanzas y redundancias temporales en cuadros de video y las agrupan de modo que se pueda reducir la información irrelevante. Las técnicas de compresión son diversas ya que cada una hace uso de un algoritmo distinto; los diferentes formatos de video, propietarios y libres, utilizan estas en múltiples aplicaciones, el video *streaming* es una de ellas. Es por medio de los *códecs* que se garantiza la calidad de la imagen a reproducir y la cantidad de datos a transmitir ante la escasez de ancho de banda que soporta las redes que trasportan el video *streaming*. Para que una reproducción de video por medio de *streaming* se lleve a cabo es necesario que el contenido se comprima de manera eficiente de modo que pueda ser transportado por Internet y pueda ser reproducido en el tiempo en que es requerido [RIJ2004].

En la siguiente tabla se mencionarán los *códecs* más usados en la actualidad junto con su aplicación y su *Bit Rate* promedio [RIJ2004].

Tabla 3 - 1: Principales códecs en la actualidad

<i>Códec</i>	Aplicación	<i>Bit Rate</i>
H.261	Video telefonía y tele conferencias sobre ISDN	p x 64 kb/s
MPEG-1	Video en medios digitales de almacenamiento (CD-ROM)	1.5 Mb/s
MPEG-2	Televisión digital	2-20 Mb/s
H.263	Video telefonía sobre PSTN	33.6 kb/s y superiores
MPEG-4	Codificación basada en objetos, contenido sintético, interacción, <i>streaming</i> de video	Variable
H.264/MPEG-4 Parte 10 (AVC)	Compresión de video mejorada	Variable

Para nuestra implementación se va a requerir trabajar con un *códec* que permita el video *Streaming* y a la vez cuente con un *bit rate* pertinente que permita la reproducción fluida y sin interrupciones del contenido.

3.1.4 Compatibilidad

Es necesario que el cliente tenga la facilidad de utilizar el sistema desde cualquier navegador de Internet, ya sea Internet Explorer, Mozilla Firefox, Google Chrome, Opera, entre otros. Dado que para la ejecución del sistema es necesaria la instalación de un *plug-in* o extensión, se debe tomar en cuenta que este pueda ser ejecutable en la mayoría de exploradores presentes en la actualidad, tanto en las versiones más recientes como en las anteriores.

Del mismo modo es importante que exista compatibilidad entre el servidor de video y todo el sistema de video *streaming* y la interfaz de usuario que debe mostrar todos los elementos para la realización de la sesión de laboratorio. Estos componentes deben trabajar en conjunto ya que forman parte de la misma arquitectura.

3.1.5 Herramientas de desarrollo

Es importante que para el desarrollo del sistema se pueda contar con herramientas, documentación, tutoriales, guías útiles que permitan al desarrollador elaborar con la menor dificultad sus objetivos. Estas no solo deben facilitar al desarrollo de la aplicación sino de ser posible deben guiar y ayudar a la resolución de problemas que se encuentren durante el proceso.

3.1.6 Seguridad

Para garantizar la integridad de la información y de los usuarios del sistema es importante que las herramientas de desarrollo proporcionen técnicas y métodos de seguridad. Estas deben ser implementadas en el proceso de desarrollo y deberán estar presentes tanto en la parte del servidor, la parte del cliente y en la etapa de transmisión a través de la Web.

3.1.7 Precio

Se debe considerar que al ser un proyecto académico el presupuesto para la realización de este no es elevado. Es por ello que se debe considerar trabajar con herramientas de desarrollo que de ser posible sean gratuitas o de un precio accesible. Esto también se aplica al servidor de video y todos los componentes del sistema de *streaming*.

En la siguiente tabla se presentan algunos precios de las herramientas que posiblemente se utilicen para la implementación.

Tabla 3 - 2: Precios de las herramientas en evaluación

Software	Precio
Adobe Flash Media Interactive Server [ADO2009]	\$4500
Adobe Flash Media Development Server (Limitado) [ADO2009]	Gratuito
Red 5 [RED2009]	Gratuito
Wowza Server [WOW2009]	\$995
Wowza Development Server (Limitado) [WOW2009]	Gratuito
Flash Media Live Encoder [ADO2009]	Gratuito
Adobe Flex Builder [ADO2009]	\$249
Microsoft Web Platform (Silverlight) [MIC2009]	Gratuito
Netbeans y JavaFX SDK [JAV2009]	Gratuito

3.2 Elección de las herramientas a utilizar

Ahora que ya conocemos los requerimientos necesarios que deben tener las herramientas de desarrollo a utilizar lo que a continuación resta es seleccionar estas. De este modo ya se podría dar inicio al diseño en primera instancia y luego a la implementación de la solución que plantea esta tesis.

Lo que a continuación se tratará primero es el análisis de las necesidades que el laboratorio va tener con respecto a la interfaz de visualización de los equipos y la comunicación entre los usuarios de una misma sesión de laboratorio. Una vez con esto ya se llevará a cabo la selección de las herramientas con las que se trabajarán durante el proceso de implementación.

3.2.1 Análisis de las necesidades del laboratorio

Para brindar una interfaz que permita la visualización de los equipos y herramientas de trabajo durante una sesión de laboratorio en línea, es de suma importancia contar con un servidor de video *streaming* que reproduzca en el instante en el que ocurre la elaboración de una experiencia ordenada por los usuarios a través de la Web. Es así pues que surge la necesidad de un servidor capaz de administrar la entrega de los contenidos de video, realizando al mismo tiempo las tareas que implican todo el proceso de emisión que se pudo apreciar en la arquitectura de un sistema de video *streaming*. Este debe obtener el video capturado a través de las cámaras Web, codificarlo y comprimirlos a un formato de modo que pueda ser capaz de ser transmitido a través de la Internet, escalar el video a una calidad y resolución que pueda ser reproducido por el usuario y entregar los contenidos a todos los alumnos que se encuentren haciendo uso del laboratorio en línea. Cabe destacar que todo este proceso lo debe realizar al momento ya que los contenidos del laboratorio se deben entregar en vivo o con un mínimo retardo de tiempo.

Del mismo modo es imprescindible que exista una comunicación entre los alumnos que se encuentren haciendo uso de los equipos de laboratorio remotamente; ya que al ser un servicio línea está implícita la posibilidad que cada usuario se ubique en un espacio geográfico distinto. Para esto la elaboración de un Chat resulta como la mejor opción para que los clientes del sistema se mantengan comunicados. Esta función de Chat es necesaria que se encuentre habilitada para todos los usuarios del laboratorio y que ni bien se conecten al servicio Web esta se active de modo que permita un trabajo grupal y en colaboración, como normalmente ocurre en una experiencia de laboratorio presencial.

Por otro lado, más que una necesidad para el laboratorio se puede tomar como una funcionalidad extra el poder apreciar las guías de laboratorio directamente en la interfaz Web. Esto facilitaría la realización de la experiencia del laboratorio ya que el usuario no tendría que tener físicamente ésta sino que se encontraría en la misma página Web del servicio.

Finalmente, es importante contar con una interfaz RIA que permita la integración de todas las herramientas multimedia y comunicación que se van a implementar en esta tesis. Esto implica que sea compatible con todas las herramientas a utilizar y a su vez con los demás módulos del proyecto del laboratorio que no se encuentran en plasmados en este trabajo pero que se encuentra realizando simultáneamente.

3.2.2 Selección de las herramientas que mejor integren las necesidades del laboratorio

Ahora que ya se definió con precisión lo que se va a necesitar implementar para brindar una solución al problema planteado al inicio de esta tesis, lo que corresponde es hacer la selección de las herramientas con las que se van a trabajar de ahora en adelante.

La herramienta principal de este trabajo es el servidor de video, para ello se selecciona la solución brindada por Adobe, Flash Media Server. Esta fue elegida entre las demás alternativas propuestas, Red 5 y Wowza Server, debido a que es la solución cuyas características y funciones se encuentra a la vanguardia y cumplen con los requerimientos y aspectos mencionados en la primera parte de este apartado.

En primer lugar, permite la reproducción de contenidos de videos en vivo por medio de la herramienta Flash Media Live Encoder. Asimismo, permite la entrega de videos a diferentes *bit-streams* dependiendo del ancho de banda de los usuarios ya que soporta escalabilidad. Por otro lado, trabaja con un gran número de *códecs* y estándares de compresión, incluyendo algunos de su

propiedad, permitiendo de esta manera elegir el más apropiado para nuestra implementación. Del mismo modo, el único requerimiento para su ejecución en los navegadores de Internet es que estos tengan instalado el *plug-in* de Flash Media Player, lo cual indica que tiene alta compatibilidad para reproducir los contenidos en cualquier plataforma Web actual. También, cabe destacar que, dado que es la herramienta para contenidos de multimedia más utilizada en el mundo, no carece de documentación, guías y tutoriales capaces de proporcionar un soporte durante todo el proceso de desarrollo. Igualmente, este cuenta con múltiples mecanismos de seguridad que abarcan todas las capas del modelo OSI, garantizando de esta manera la integridad de la información con la que trabajará el sistema. Finalmente, con respecto al precio para hacer uso de esta utilidad existe una versión para desarrolladores la cual permite como máximo diez conexiones en simultáneo [ADO2009].

Asimismo, a través de esta herramienta que es posible el desarrollo de un servicio de chat. Dado que por medio del lenguaje de programación ActionScript se puede personalizar ciertas características de la interfaz, para activar esta función solo se deben colocar algunas instrucciones en el código fuente.

A continuación se indicará todos los beneficios que tiene el utilizar Flash Media Server como servidor de video [ADO2009].

Experiencia del usuario:

- Reproducción de video más rápido.
- Soporte de video en vivo.

Calidad:

- *Streaming* en *códec* de video H264/AAC y On2 vp6.
- Detección automática del ancho de banda y memoria dinámica.

Ubicuidad de entrega:

- La plataforma al cliente se puede ejecutar en Windows, Mac o Linux.
- Adobe Flash Player cuenta con más de 8 millones de descargas diarias y penetración del 98% de las computadoras personales alrededor del mundo.
- Los servidores pueden correr tanto en Windows y Linux.

Interactividad

- Manejo integrado de vídeo, audio, streaming de datos y un completo conjunto de herramientas de comunicación social.
- Servidor personalizable mediante lenguaje ActionScript.

Seguridad

- Mayor seguridad en la entrega de contenido con codificación y control de acceso a las características del cliente.
- SWF de verificación para asegurar que las aplicaciones estén autorizadas.

Con respecto a las cámaras Web a utilizar, como parte del equipamiento del laboratorio, se han adquirido cuatro de la marca Logitech siendo el modelo seleccionado QuickCam® Orbit AF. Estas cuentan con las siguientes características [LOG2010].

- Componentes ópticos Carl Zeiss®
- Sistema de enfoque automático
- Sensor de 2 megapíxeles de resolución superior con tecnología RightLight™ 2
- Profundidad de color: Color verdadero de 24 bits
- Captura de vídeo: hasta 1600 x 1200 píxeles (calidad de alta definición)
- Frecuencia de cuadro: hasta 30 cps
- Captura de fotografías: 8 megapíxeles (mejora por software)
- Micrófono integrado con tecnología RightSound™



Figura 3 - 3 QuickCam® Orbit AF de Logitech
Fuente: [LOG2010]

Por otro lado, para ofrecer la facilidad de visualizar las guías de laboratorio directamente de la interfaz se va a hacer uso de la solución de código libre FlexPaper [DEV2010]. Esta herramienta permite visualizar archivos en pdf en un formato swf, es decir en un archivo Flash. A través del aplicativo PDF2SWF [SWF2010] las guías son convertidas en este formato lo cual permite su lectura en cualquier navegador Web que tenga instalado Flash Media Player. Como se indica se ofrece la opción de leer los archivos mas no modificarlos.

Para finalizar, con respecto a la selección de la herramienta capaz de integrar todas las funcionalidades descritas previamente para mostrarlas en una interfaz basada en la tecnología de las aplicaciones enriquecidas de Internet, se elige a Flash como la mejor alternativa frente a las mencionadas en el capítulo anterior. Se llegó a seleccionar a esta herramienta de desarrollo de RIA's ya que se va a desarrollar un servidor de video basado en esta tecnología. Así pues, para garantizar la compatibilidad entre las herramientas a trabajar se opto por esta solución.

3.3 Descripción de la solución

Con la selección de herramientas a trabajar ya realizada, lo que a continuación se desarrollará será una breve descripción de lo que se propone implementar. Asimismo, se describirá la interfaz que permitirá al usuario poder visualizar los equipos de laboratorio para la realización de las experiencias y mantener una comunicación entre todos los clientes que se encuentren haciendo uso del servicio Web.

En primer lugar, las cámaras Web estarán desplegadas en el laboratorio que albergará los equipos e instrumentación que permitirían la realización de las sesiones en línea. Estas estarán conectadas a una PC destinada únicamente al soporte del servidor de video y las herramientas informáticas con las que trabaja. A su vez este servidor se encontrará conectado con el servidor Web que dará acceso a los recursos de multimedia cuando se de inicio a una sesión de laboratorio.

Cuando un alumno ingrese a la página Web del laboratorio virtual remoto este automáticamente cargará la interfaz que permitirá la reproducción de las cámaras Web. De acuerdo al equipo de laboratorio al que se encuentre haciendo uso en la sesión en línea se tendrá la opción de elegir una vista distinta por medio de otra de las cámaras Web desplegada en el ambiente trabajo. Cabe destacar que cada alumno tiene a su disposición cualquier ángulo de visualización así que en cualquier momento puede observar la cámara que desee. Asimismo esta interfaz trabajará en conjunto y de manera simultánea con las aplicaciones encargadas de controlar los equipos remotamente.

Con respecto a las guías que se encontrarán disponibles estas serán cargadas desde el servidor Web utilizando la herramienta FlexPaper y de esta manera encontrarán disponibles dentro de la interfaz.

Por otro lado, el servicio del Chat estará disponible desde que el alumno ingresa a la página del laboratorio. De modo que se pueda verificar si el grupo esta completo y de esta manera se programe el inicio de la sesión.

Como se ha podido apreciar, todos los servicios que se van a ofrecer en la interfaz de comunicaciones van a ser posibles por medio de la plataforma de Flash Media Server y Flash. Es por ello que en lo que respecta al desarrollo de tesis se enfocará en la

definición del diseño y la implementación de esta utilizando estas herramientas. En los próximos capítulos se podrá apreciar esto, definiendo primero la arquitectura de la solución, procediendo luego a la implementación y finalmente a la ejecución de pruebas y conclusiones de esta.





Capítulo 4

Diseño de la solución

En capítulos anteriores se eligieron las herramientas que resolverán el problema planteado, lo que a continuación se desarrollará es el diseño de la solución propuesta. En primer lugar, se describirá como estará desplegada la arquitectura del sistema que permitirá la visualización remota del laboratorio y sus herramientas. Luego de esto, se mencionarán las consideraciones que se deberán tomar con el servidor de medios de modo que pueda atender todos los requerimientos que la aplicación a desarrollar necesitará. Finalmente procederemos a describir las aplicaciones que harán uso del servidor y formarán parte de la interfaz que distribuirán los contenidos de video *streaming* y establecerán una comunicación entre los usuarios del laboratorio.

4.1 Arquitectura de la solución

El equipo principal de todo el sistema es el servidor de video, este es el que tendrá conectado la PC encargada de las cámaras desplegadas en el laboratorio y al que el servidor web enlazará, por medio de una aplicación flash, para poder visualizar los contenidos de video en vivo y hacer uso del sistema interactivo. Así pues, este equipo tendrá como herramienta principal el servidor de medios Flash Media Server, el cual será el encargado de gestionar las aplicaciones que hacen uso del servidor, manejar las conexiones de todos los usuarios que se encuentren en el laboratorio remoto y llevar un registro de todos los eventos que ocurren en el servidor.

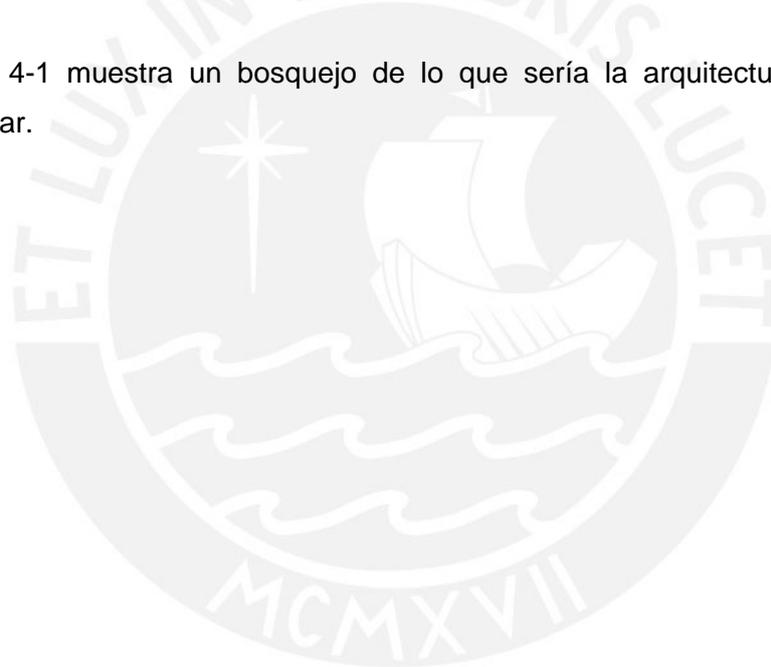
Por otra parte, el servidor estará conectado a un equipo que soportará varias instancias de Flash Media Live Encoder el cual le permitirá obtener la señal de *streaming* que se transmitirá hacia los clientes. Este software, desarrollado también por Adobe, capturarán el video proveniente de las cámaras web, codificará los contenidos con un *códec* determinado, comprimirá la información en distintos *bitrates* de modo que pueda ser transportada por la red y generará las conexiones necesarias con el servidor de video para que se encuentre disponible de modo que se pueda realizar el *streaming* en vivo. Al referirnos de varias instancias de esta aplicación es que cada una manejará una cámara conectada al equipo. Cabe destacar que es posible que este equipo sea el mismo utilizado en el servidor, sin embargo para la solución planteada se decidió separarlas para tener más escalabilidad.

Por otro lado, además del servidor de video el sistema contempla un servidor web, este es el que cargará la aplicación flash cliente el cual permitirá a los usuarios visualizar los equipos del laboratorio durante sus experiencias y habilitará el chat que les permitirá comunicarse. La aplicación flash cliente se almacenará dentro de los archivos del servidor y desde este se generará la conexión al servidor de medios. Es a partir del servidor web que los alumnos del laboratorio remoto tendrán acceso a las funcionalidades de este, ya sea acceder e identificarse, manejar remotamente la instrumentación y obtener los resultados de las experiencias realizadas. Cabe destacar que el diseño de la aplicación web no está contemplada en este trabajo de tesis, únicamente se tratará que el sistema se conecte a partir de un servidor web de pruebas.

Asimismo, cuatro cámaras web se encontrarán desplegadas en el laboratorio y enfocarán distintos ángulos de las herramientas del laboratorio de manera que los usuarios puedan apreciar claramente las experiencias y actividades que se realicen remotamente. Estas estarán conectadas al equipo encargado de ejecutar la aplicación Flash Media Live Encoder donde se podrán ajustar los parámetros de video de manera que los contenidos puedan transportarse en una forma más eficiente a través de la red.

Finalmente, es necesario que los usuarios del laboratorio remoto tengan instalados en las computadoras donde realizarán las experiencias el *plugin* para los navegadores de Internet del Flash Media Player. Así pues, este utilitario permitirá cargar la aplicación flash una vez que el usuario ingrese al portal web del laboratorio.

La figura 4-1 muestra un bosquejo de lo que sería la arquitectura del sistema a implementar.



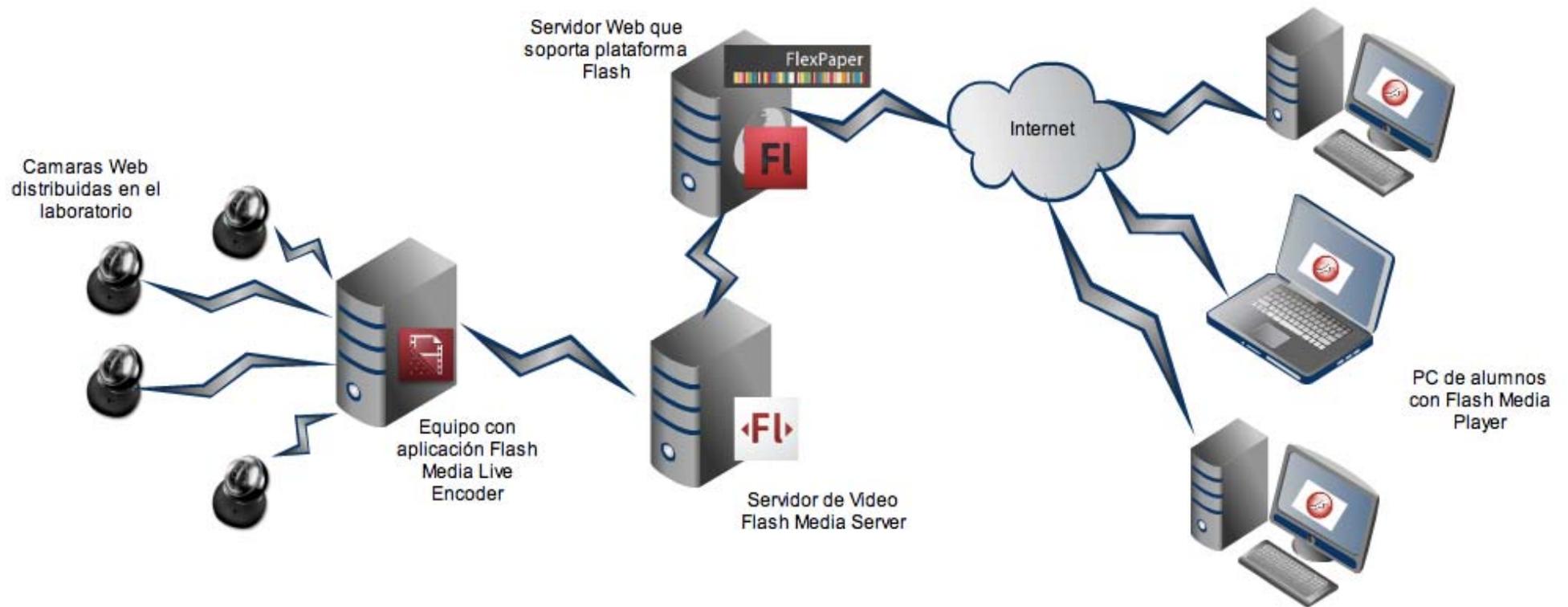


Figura 4 - 1 Arquitectura de la solución
Fuente: Elaboración propia

4.2 Configuración del Servidor de Video

Adobe ofrece Flash Media Server en tres diferentes configuraciones de servidor de acuerdo a las necesidades que se tengan. En primer lugar tenemos Flash Media Interactive Server (FMIS) el cual tiene como destacada ventaja el número ilimitado de conexiones concurrentes a su servidor, este brinda un gran número de características que permitirán al administrador contar con aplicaciones interactivas de *streaming*. La segunda versión ofrecida es Flash Medias Streaming Server (FMSS) la cual esta limitada únicamente a ofrecer *streaming* de audio y video. Finalmente, en tercer lugar encontramos a Flash Media Development Server, el cual es una versión gratuita de la primera solución pero que se limita a tener como máximo diez conexiones simultáneas. Durante el desarrollo de este trabajo de tesis se utilizará la tercera opción.

Para hacer uso del servidor en nuestra solución hay que tener ciertas consideraciones de modo que se pueda emplear de manera más eficiente. Con respecto a la instalación del servidor esta se encuentra detallada en el **Anexo 1**. Una vez instalado, la primera consideración a tener es que en el servidor se debe ubicar la ruta en la que se encuentran los archivos de este. Esto es importante porque cada aplicación que haga uso de los recursos de este servidor deberá tener una carpeta dentro del file de aplicaciones que se crea una vez instalado en el computador. De esta manera estas aplicaciones podrán generar conexiones necesarias que los usuarios utilizaran para tener la experiencia del laboratorio.

Otra consideración a tomar es con respecto al usuario del modulo de administración, el cual se crea en el proceso de instalación. En este caso se debe siempre tener a la mano este acceso ya que a través del modulo de administración se podrá controlar el servidor y llevar un registro de todos los eventos que ocurren cuando el servidor esta activo y las aplicaciones se encuentran haciendo uso de este. En este modulo que se instala simultáneamente con el servidor se puede llevar un control de todas las aplicaciones que se han conectado a este, todos los clientes que han accedido y todos los recursos que se utilizan en el transcurso de una conexión. De esta manera se puede llevar un seguimiento más detallado de cómo se comporta el servidor una vez que el laboratorio ya se encuentre implementado. En la figura 4-2 se puede apreciar la interfaz de la consola de administración.

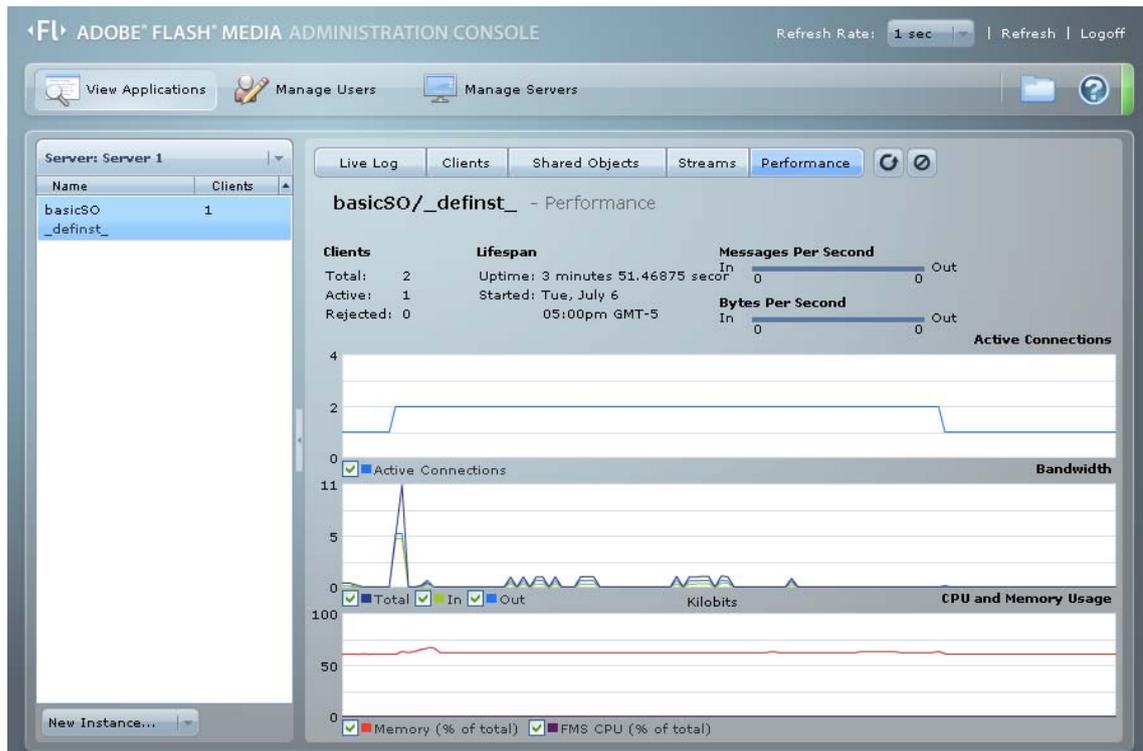


Figura 4 - 2 Consola de Administración
Fuente: Herramienta Flash Media Server

Por otro lado, es importante también hacer un buen manejo del ancho de banda, dado que son aplicaciones con video streaming, estos tienden a consumir una gran parte del ancho de banda que se tiene para brindar los servicios del laboratorio a través del Internet. Es por ello que al distribuir los contenidos se debe tener un control apropiado de este recurso enviando a una tasa de bits que permita a los usuarios una visualización fluida de lo que ocurre remotamente sin interrupciones y a la vez sin perjudicar los demás servicios. Para ello es necesario que el servidor ofrezca a los usuarios la calidad de video que mejor se adecue al ancho de banda de los usuarios de manera que a la vez no se consuma el innecesariamente el ancho de banda que este tiene. Así pues, el servidor de video deberá tener por lo menos dos versiones del mismo contenido a diferentes *bit-streams* y al momento de generarse la conexión, una vez obtenido el ancho de banda del cliente, este deberá ofrecer el más adecuado.

Finalmente, es de suma importancia determinar que cuando el servidor este activo y tenga conectado la aplicación de los clientes no se exceda en el consumo de recursos del computador. Esto es que el procesamiento no se eleve excesivamente y que la memoria RAM no llegue a consumir toda su capacidad. Si esto ocurre podría ser perjudicial para el servicio ya que el servidor podría caerse e interrumpir el servicio.

4.3 Diseño de las aplicaciones

Como ya se menciona las aplicaciones serán las que permitirán al los clientes acceder al servidor de medios y de esta manera poder apreciar lo que se realiza en el laboratorio por medio del *streaming* de video en tiempo real y a la vez poder establecer la comunicación entre todos los usuarios del laboratorio por medio de un Chat. A continuación se dará una descripción de estas aplicaciones que se desarrollarán el siguiente capítulo.

En primer lugar, tenemos a la aplicación que permitirá visualizar el video streaming de lo que ocurre en el instante en el que se esta realizando la experiencia de laboratorio. Este tomará el contenido del servidor de video que es capturado por medio de una instancia de Flash Media Live Encoder a unos determinados parámetros configurados previamente en la aplicación. Asimismo, esta aplicación a la vez capturara el mismo contenido de video en dos diferentes velocidades de bits de manera que se pueda distribuir a cada usuario la que mejor se adecuara a su ancho de banda. Una vez capturado el video este también generara una conexión al servidor de manera que el contenido pueda ser subido a este y estar disponible para nuestra aplicación. Es así que la aplicación a diseñar tendrá que generar una conexión al servidor y luego obtener el video disponible gracias al flash media Live Encoder. Asimismo, como se menciona previamente a la visualización de los contenidos, esta detectara el ancho de banda de cada usuario que se este conectando de modo que se pueda entregar contenido de video con el bitstream mas adecuado.

Por otro lado, esta aplicación deberá mostrar el contenido obtenido de todas las cámaras desplegadas en el laboratorio. Así pues la aplicación deberá darle al usuario la opción de poder elegir la cámara que desea visualizar independientemente de los demás usuarios conectados. Es por esta razón que se utilizaran más de una instancia del aplicativo Flash Media Live Encoder lo que permitirá que cada una de estas controle una de las cámaras web disponibles y suba al servidor el video capturado. Es a través de esta facilidad que cada alumno podrá elegir en cualquier momento durante la experiencia del laboratorio el ángulo de visualización que desee de manera que se pueda enfocar en una determinada herramienta o prueba que se realice.

En la figura 4-3 se mostrará un bosquejo de como será la interfaz de los clientes una vez implementada.



Figura 4 - 3 Bosquejo aplicación de video
Fuente: Elaboración propia

En segundo lugar, como ya se ha mencionado previamente se elaborará una aplicación de Chat que permitirá a los usuarios de laboratorio comunicarse entre ellos de manera que puedan realizar una experiencia colaborativa. Esta se integrará a la aplicación visualizadora de video de manera que el usuario pueda hacer uso de los dos al mismo tiempo. Del mismo modo, esta aplicación también estará soportada a través de Flash Media Server así que de igual manera que la primera tendrá que generar una conexión con este servidor para funcionar. En si la aplicación será básica pero cumplirá su función de establecer la comunicación entre usuarios, ya que solo soportará texto. El elemento clave que permitirá implementar esta aplicación es la propiedad del servidor de medio de contar con objetos compartidos. En este caso el objeto que los usuarios van a compartir es el texto de la conversación, el cual será el mismo en todos los clientes y se actualizará cada vez que se envíe un nuevo mensaje.

A continuación en la figura 4-4 se muestra un bosquejo de esta aplicación.



Aplicación de Chat

Área del texto de conversación

Input de mensaje a enviar

Enviar

Nombre de usuario

Figura 4 - 4 Bosquejo de la aplicación Chat
Fuente: Elaboración propia

Como se aprecia en el bosquejo de la aplicación esta contiene todas las casillas que normalmente se encontrarían en una aplicación de Chat, un área donde se muestra la conversación, una casilla donde se escribe el mensaje y el botón de enviar. Sin embargo la aplicación a desarrollar tiene un área de texto donde el usuario colocará su nombre. Para la integración final de todos los módulos del laboratorio, que no esta contemplada en este trabajo, la aplicación obtendrá el nombre del usuario de la base de datos del sistema.

Finalmente, previamente ya mencionamos como una funcionalidad que tendrá la interfaz es poder apreciar los documentos de las guías de laboratorio que se va a desarrollar. Para ello se trabajará con FlexPaper, el cual es una aplicación libre elaborada por Devaldi Group [DEV2010], la cual se añadirá en el portal Web del laboratorio. Esta aplicación no se integrará a la aplicación que esta diseñando ya que contiene sus propios archivos SWF los cuales se agregaran al servidor Web pero si formará parte de la interfaz de la solución. En el siguiente capítulo se desarrollará como se integrará a nuestra solución y como poder subir los documentos para que puedan ser visualizados.

Como se ha podido observar en el capítulo se ha desarrollado el diseño de las aplicaciones que permitirán una experiencia real de laboratorio a todos los usuarios que hagan uso del portal. En lo que respecta al siguiente capítulo se llevara a cabo la implementación del prototipo de la solución y la integración de todas las aplicaciones que acabos de mencionar. Finalmente, una vez que ya se tenga el prototipo se precederá a realizar las pruebas necesarias para evaluar el desempeño de nuestra interfaz.





Capítulo 5

Prototipo de la solución

En este capítulo se desarrolla el proceso de implementación de las aplicaciones que brindarán la solución al problema planteado en esta tesis. Es de esta manera que se detallará cómo fueron elaboradas cada una de las aplicaciones diseñadas en el capítulo anterior indicando las herramientas que se utilizaron y la lógica para la programación. Del mismo modo a través de las figuras se podrán apreciar las interfaces obtenidas. Seguido de esto se describirá como se llevo a cabo la etapa de integración para obtener una interfaz unificada y completa lo cual es el objetivo que se estaba buscando en esta tesis. Finalmente, se describirán las pruebas que se realizaron para analizar el desempeño de la solución propuesta.

5.1 Implementación del prototipo

La implementación del prototipo consta de dos etapas. En un primer lugar se desarrollará como se elaboraron las diferentes aplicaciones que han sido descritos en el capítulo anterior. Así pues se indicará como se desarrollaron las aplicaciones y las funciones principales de estas, la configuración del Flash Media Live Encoder y la interfaz resultante de la programación. Luego como corresponde en la segunda parte se indicará como se llegó a integrar estas aplicaciones de modo que como resultado se tenga el prototipo de la solución presentada.

5.1.1 Desarrollo de aplicaciones

Antes de proceder a describir el desarrollo de las aplicaciones, se dará una breve explicación de la herramienta principal de trabajo. Para el desarrollo de las aplicaciones en Flash se trabajó con la herramienta Adobe Flash CS4, la cual permite tanto el diseño de la interfaz de usuario como la programación lógica de la aplicación.

En la herramienta de desarrollo siempre se trabaja con dos archivos principalmente que deben tener el mismo nombre pero con distintas extensiones. El primero de estos es el archivo flash, con extensión **.FLA**, en el cual se diseñara la interfaz de usuario y se colocarán los componentes necesarios para su funcionamiento. Por ejemplo en el caso de la primera aplicación es necesario añadir el componente de video en la librería de trabajo. Esto permitirá que el código de programación que se ingresará haga uso de este componente al invocarlo. Por otro lado, el segundo archivo de trabajo es el que contiene el código de programación que tendrá toda la parte lógica de la aplicación, este es un archivo ActionScript 3.0 con la extensión **.AS**. Este archivo, como ya se menciono, esta basado en el paradigma de programación orientada a objetos por lo que al desarrollar se deberán colocar las librerías, clases y funciones necesarias para que la aplicación se ejecute. De acuerdo a la necesidad de la aplicación es posible que se lleguen a utilizar más de un archivo con extensión **.AS**, sin embargo solo el principal es el que tiene que tener el mismo nombre que el archivo con extensión **.FLA**. Por último, se obtiene un archivo con extensión **.SWF** el cual es el que ejecuta la aplicación y es reproducido con Flash Player.

La primera aplicación que se desarrollará es la que permite mostrar el video capturado de Flash Media Live Encoder. En este caso esta solo mostrará el contenido obtenido de una cámara Web ya que más adelante se desarrolla la aplicación con las múltiples cámaras. Cabe destacar que todos los códigos elaborados con la herramienta Flash CS4 para las aplicaciones de esta solución se encontrarán en el **Anexo 2**.

Al inicio de toda aplicación Flash elaborada con ActionScript 3.0 se debe indicar las librería que se utilizarán. En este caso, las principales librerías son NetConnection, NetStream, y Video. La primera hace referencia a la conexión que se debe generar con el servidor, la segunda al manejo de *streams* de video que se obtendrá del servidor y la ultima permitirá la reproducción del video obtenido. Seguido de esto se encuentra la clase principal que hace referencia al nombre de la aplicación Flash, dentro de ella se crearan todas las variables necesarias para la aplicación y todas las funciones que se ejecutarán para que esta trabaje. La función principal también es llamada como la aplicación y es la que invoca a la función que configura el video y la que genera la conexión con el servidor.

Una conexión con el servidor de medios se lleva a cabo primero creando una variable String que contenga la ruta del servidor y de la aplicación que se invocará. Dependiendo del caso esta ruta puede ser un dominio de Internet, una dirección IP o una ruta interna de un computador. En este caso, dado que es una aplicación de prueba con el servidor en la misma computadora de trabajo, se colocó una ruta interna. Luego se crea una variable de tipo NetConnection que permitirá invocar al método **connect** el cual deberá tener como parámetro al String que contiene la ruta previamente mencionada. Al mismo tiempo por medio de un evento, que en este caso es que se haya realizado la conexión, se hace un llamado a las otras funciones de la aplicación.

La siguiente función se llama **CheckConnect** la cual, una vez evaluada que la conexión se haya llevado a cabo exitosamente, se procede a crear un objeto *stream* que permitirá obtener el video que se encuentra en el servidor. Este *stream* es invocado por medio del nombre que se le asigna en el Flash Media Live Encoder y es añadido a un objeto de video de la interfaz, de modo que pueda ser reproducido y visualizado por los usuarios. Del mismo modo otra de las funciones importantes en la aplicación es la de **SetVideo** la cual establece

los parámetros que tendrá el video que se reproducirá en la interfaz una vez que se obtenga el *stream* del servidor. Esta principalmente ubicará el contenido en unas coordenadas determinadas dentro de la aplicación flash e indicará el tamaño de resolución con que se mostrará.

Con esto se tiene las principales funciones que se programan en el archivo .AS que conforman la parte lógica de la aplicación. A continuación se desarrollará la configuración que necesariamente se debe tener en el aplicativo Flash Media Live Encoder que hará posible capturar el video por medio de una cámara web y cargarlo al servidor de video para que el *stream* pueda estar disponible para reproducirse en la aplicación.

En la figura 5-1 se puede apreciar la interfaz de esta aplicación. En ella lo primero que se debe llevar a cabo es la elección de la cámara web que capturará el video. Seguido de esto se determina la resolución con la que se capturará el contenido, que deberá ser del mismo tamaño que se estableció en el código, y la velocidad de bits que tendrá el *stream* que se subirá al servidor. Cabe destacar que este *bitstream* es el que determinara el ancho de banda necesario que el usuario deberá tener para poder visualizar el contenido. Así pues, más adelante se utilizará dos velocidades distintas de modo que puedan mejor adecuarse a cada cliente del sistema. Esta aplicación también permite codificar audio sin embargo para nuestro sistema no es necesario. Una vez que ya se ha determinado los parámetros más adecuados, se coloca la ruta de la aplicación del servidor en donde el stream estará disponible. Este deberá tener la misma ruta que se estableció en la aplicación para generar la conexión. Seguido de esto, se ingresa el nombre que tendrá el *stream* de modo que pueda ser identificado tanto en el servidor como en la aplicación para su invocación.

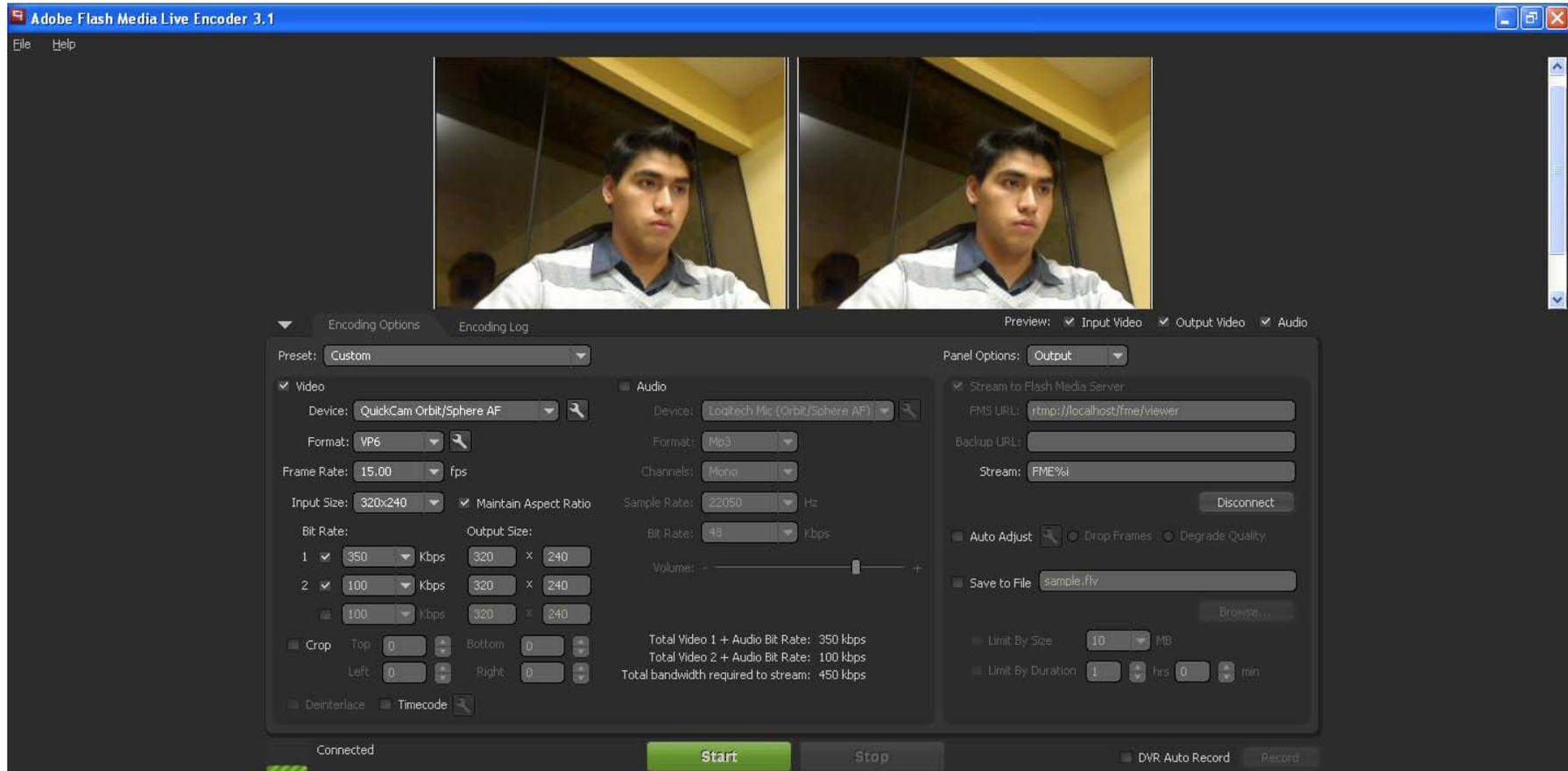


Figura 5 - 1 Interfaz de Flash Media Live Encoder

Con esta configuración ya se puede levantar la conexión al servidor para que el *stream* de video pueda ser cargado a este, para ello primero se presiona el botón connect y luego Start. Una vez realizado esto, la aplicación notifica si se ha establecido la conexión exitosamente y si se ha llegado a cargar el *stream* de video. Finalmente lo ultimo que corresponde hacer es levantar la aplicación Flash para de terminar si el proceso de elaboración se ha realizado con éxito.

En la gráfica 5-2 se puede apreciar el resultado del desarrollo. Como se observa el *streaming* de video que se reproduce es el mismo que se obtiene de la aplicación Flash Media Live Encoder.



Figura 5 - 2 Primera aplicación desarrollada

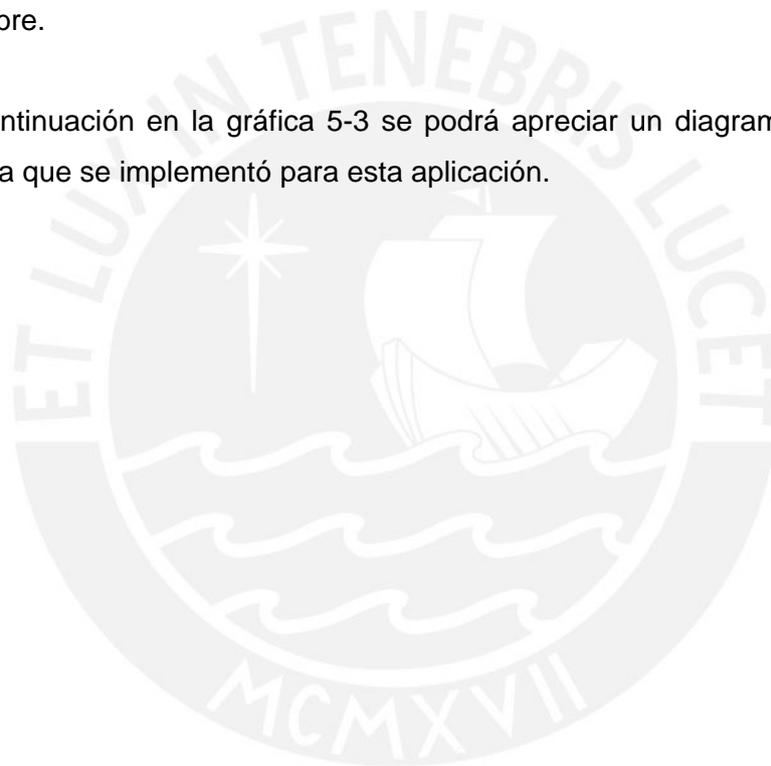
Con este desarrollo se ha explicado detalladamente como elaborar una aplicación en Flash de modo que pueda obtener una interfaz capaz de reproducir un video *streaming* por medio de un servidor de video. A continuación se explicará como se elaboraron la aplicación de Chat y la interfaz que permite interactuar con diversos *streams* de video pertenecientes a las distintas cámaras web que se desplegarán en el laboratorio.

En la aplicación de Chat el método para conectarse al servidor es el mismo que se utilizó para la primera aplicación. Así pues el código tiene la misma estructura que el que se desarrolló previamente sin embargo en este caso no se hace uso de un objeto de video. El elemento principal para el funcionamiento de esta aplicación es el uso de objetos compartidos, estos tienen como característica principal el que pueden ser visualizados y manipulados por todos los usuarios que se conecten al servidor. Para definirlo se le puede describir como una suscripción al cual todos los usuarios de un servicio son capaces de utilizar. Cabe destacar que estos objetos no son persistentes ya que no se almacenan en el servidor y una vez que la conexión se finaliza estos se borran del sistema. El manejo de estos elementos es similar al de cualquier variable disponible en la aplicación. Sin embargo, para instanciarlos es necesario asignarle un nombre que permita identificarlo en el servidor y asociarlo a la conexión establecida.

El objeto que se compartirá entre los usuarios será el texto que enviarán los clientes en forma de mensajes. Así pues, como en una aplicación de Chat tradicional, cada vez que un usuario envíe un mensaje a través de la aplicación esta será visualizada en las interfaces de los demás clientes. Ahora para que esto ocurra es necesario que se sincronice en todas las interfaces el objeto compartido. Es por ello que en el código existe una función que ante la ocurrencia de eventos como el envío de nuevos mensajes el sistema debe actualizar este objeto en todos los demás usuarios. Es de esta manera como es posible el establecimiento de la comunicación. Cabe mencionar que, este manejo de eventos debe estar presente tanto cuando un usuario presiona el botón de enviar el mensaje y cuando estos presionan la tecla de **enter**, es por ello que para cada uno de estos eventos se elabora una función.

Del mismo, como se menciona en el capítulo cuatro, dado que en este trabajo no se van a obtener los nombres de los usuarios del sistema de laboratorio, se tendrá un input en donde los usuarios de la interfaz colocarán su nombre de modo que se pueda identificar de quien provino el mensaje enviado. Para esto la aplicación condicionara de que si no ingresa el nombre no se podrá enviar ningún mensaje al Chat. Esto se realiza tanto en la función de presionar el botón y en la de la tecla **enter**, de esta manera antes de enviar el mensaje a través del objeto compartido se evalúa que el input de nombre no se encuentre vacío, en caso que no se cumpla la aplicación notifica al usuario que deberá ingresar su nombre.

A continuación en la gráfica 5-3 se podrá apreciar un diagrama de flujo con la lógica que se implementó para esta aplicación.



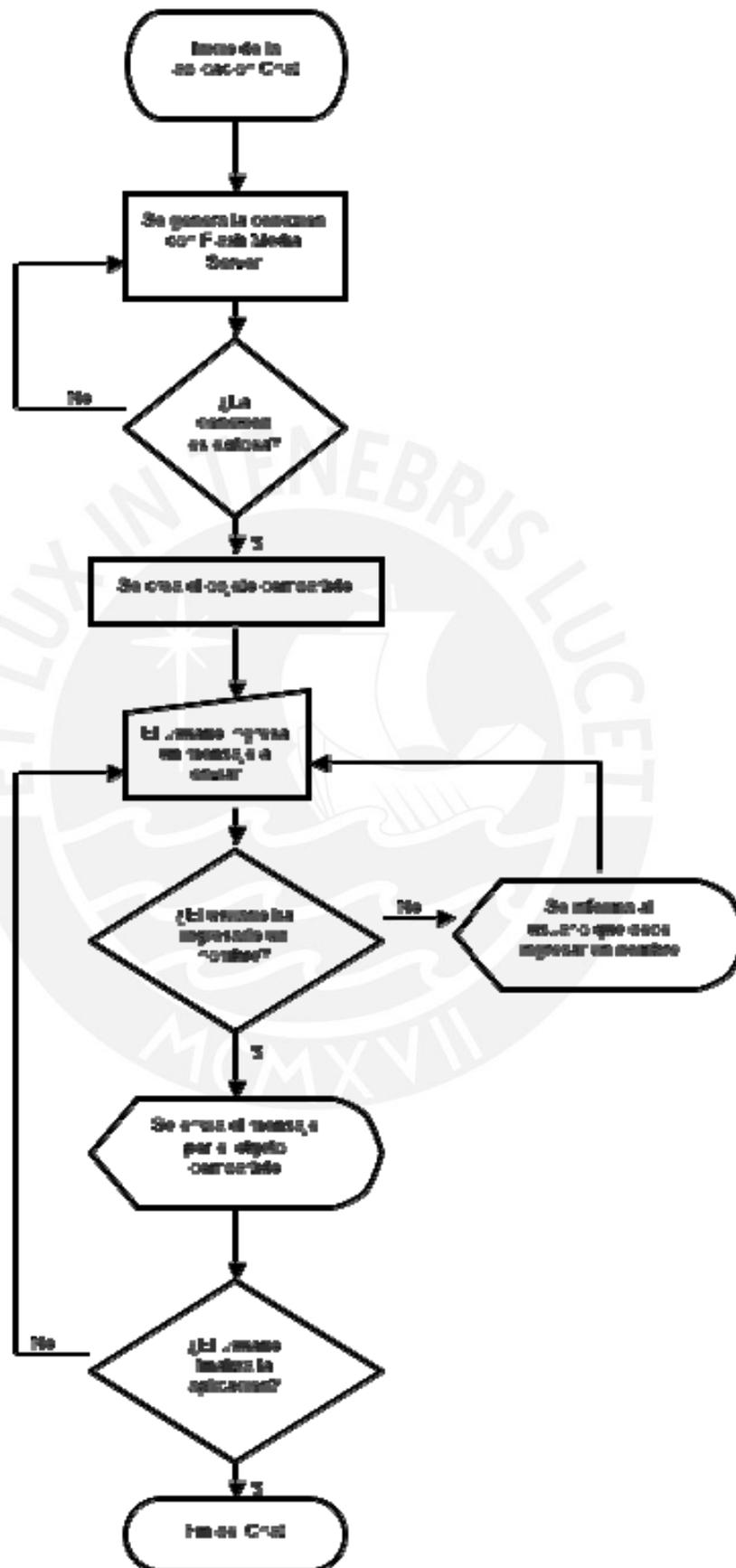


Figura 5 - 3 Diagrama de flujo de aplicación Chat

En la gráfica 5-4 se muestra la aplicación de Chat desarrollada que luego se integrará con la interfaz de Video, en ella se observa una comunicación entre dos usuarios conectados.

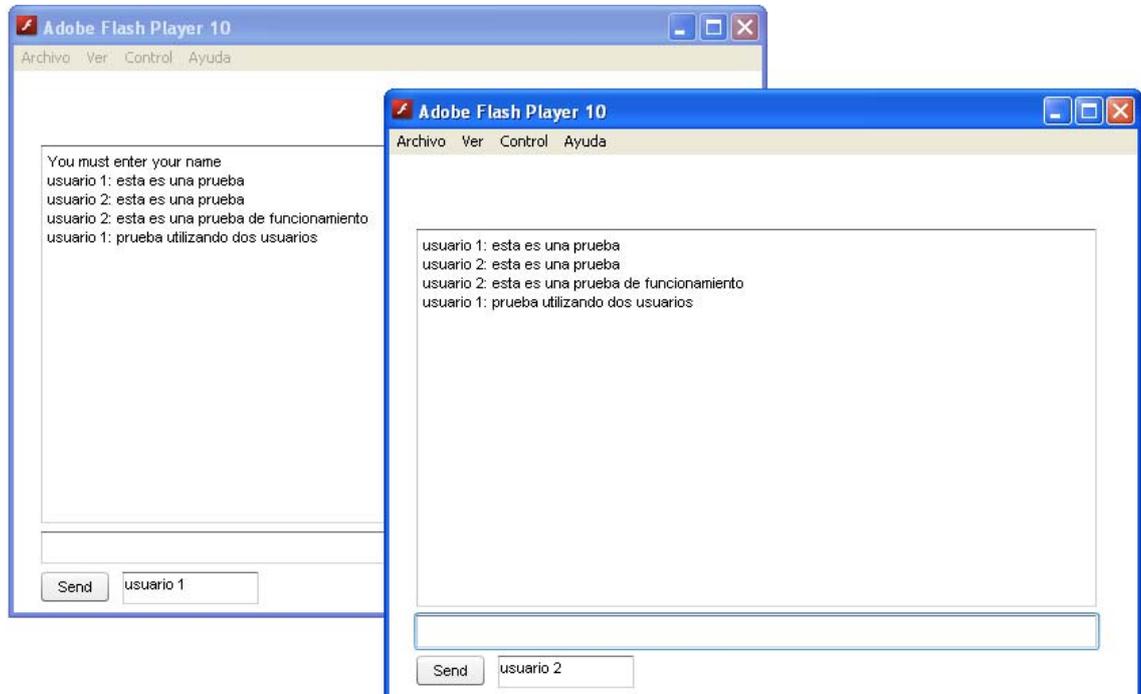


Figura 5 - 4 Aplicación Chat

Finalmente, la última aplicación que se desarrollará es la interfaz de video completa. Esta será similar a la primera aplicación que se describió pero tendrá nuevas funcionalidades. En primer lugar, el usuario tendrá la opción de elegir el ángulo de visualización del laboratorio por medio de unos botones que tendrá la interfaz y además al realizarse la conexión con el servidor, esta analizará el ancho de banda con el que se encuentra enlazado cada usuario de modo que se pueda brindar a cada cliente el *bitstream* de video que mejor se adecue a su conexión.

Como ya mencionamos esta aplicación tiene la misma lógica que la primera con respecto a la reproducción de los videos obtenidos del *stream*. Así pues estos utilizaran las mismas funciones para la conexión y la configuración del objeto de video, así como el uso del Flash Media Live Encoder para la captura de los contenidos y la carga al servidor de video. Lo que diferencia a la primera es que

para cada una de las cámaras que se utilizaran en la interfaz se hará uso de una instancia del FMLE, así pues cada instancia será independiente de la otra y cada una subirá al servidor un contenido distinto. Cabe mencionar además que en las instancias se codificará el video en dos *bitstreams* distintos para los diferentes enlaces de los usuarios.

En un primer momento, al iniciarse la aplicación esta llamara a reproducir a una cámara al igual como se llevo a cabo en la primera aplicación. Luego, en la interfaz de video se agregarán botones siendo asignados cada uno de estos a un ángulo de visualización del laboratorio. Estos botones invocarán a una función que habilitará la reproducción en la interfaz. Así pues una vez presionado, la función llamará al *stream* seleccionado por medio del nombre que se le asigno en el FMLE, a través del método **play** que proviene del objeto de video creado al inicio. Es de esta manera que todas las reproducciones de los *streams* de video se visualizarán por medio del mismo objeto al cual en cada selección de cámara se le asigna un nuevo *stream*. Debido a esto cuando se lleve a cabo una transición entre dos *stream* el cambio es fluido y en ningún momento se detiene o congela la reproducción.

Por otro lado, para determinar el ancho de banda que tiene cada usuario conectado a la interfaz se desarrollo otro archivo **.as**. Este tiene la función de devolver a la aplicación principal la velocidad en *kilobits* que tiene el usuario que hace uso del sistema. Con esta información la aplicación es capaz de decidir el video con un determinado *bitstream* que es el más adecuado para visualizar.

Así pues, esta información del usuario se utiliza al inicio de la aplicación una vez que se haya realizado la conexión con el servidor y antes de iniciar la reproducción del video *streaming*. Del mismo modo también es necesario en cada función que es ingresada por medio de los funciones de selección de cámara. El motivo de esto es para que la aplicación decida cual es el *bitstream* de video que se le asignará al usuario.

Finalmente, para obtener las dos alternativas de *bitstream* que se podrán distribuir en la interfaz, es necesario configurar en Flash Media Live Encoder la opción de codificación doble. De esta manera se obtendrá el mismo contenido de

video en una velocidad de 100 kbps para anchos de banda limitados, y una velocidad de 350 kbps para usuarios con una velocidad de conexión más alta. Luego para poder identificar los diferentes *bitstream* en el servidor al que se cargarán, en el nombre del *stream* se agrega los símbolos *%i* que identificaran a cada uno por medio de un número. Por ejemplo, ante un *stream* llamado **envivo%i**, en el servidor se obtendrá un *stream* de 100 kbps llamado **envivo1** y otro de 350 kbps llamado **envivo2**. Como ya mencionamos esta configuración deberá estar presente en cada instancia del FMLE que controla cada cámara del laboratorio.

En las graficas 5-5 y 5-6 se puede apreciar la interfaz obtenida con estas dos nuevas funcionalidades y en dos ángulos de video distintos.

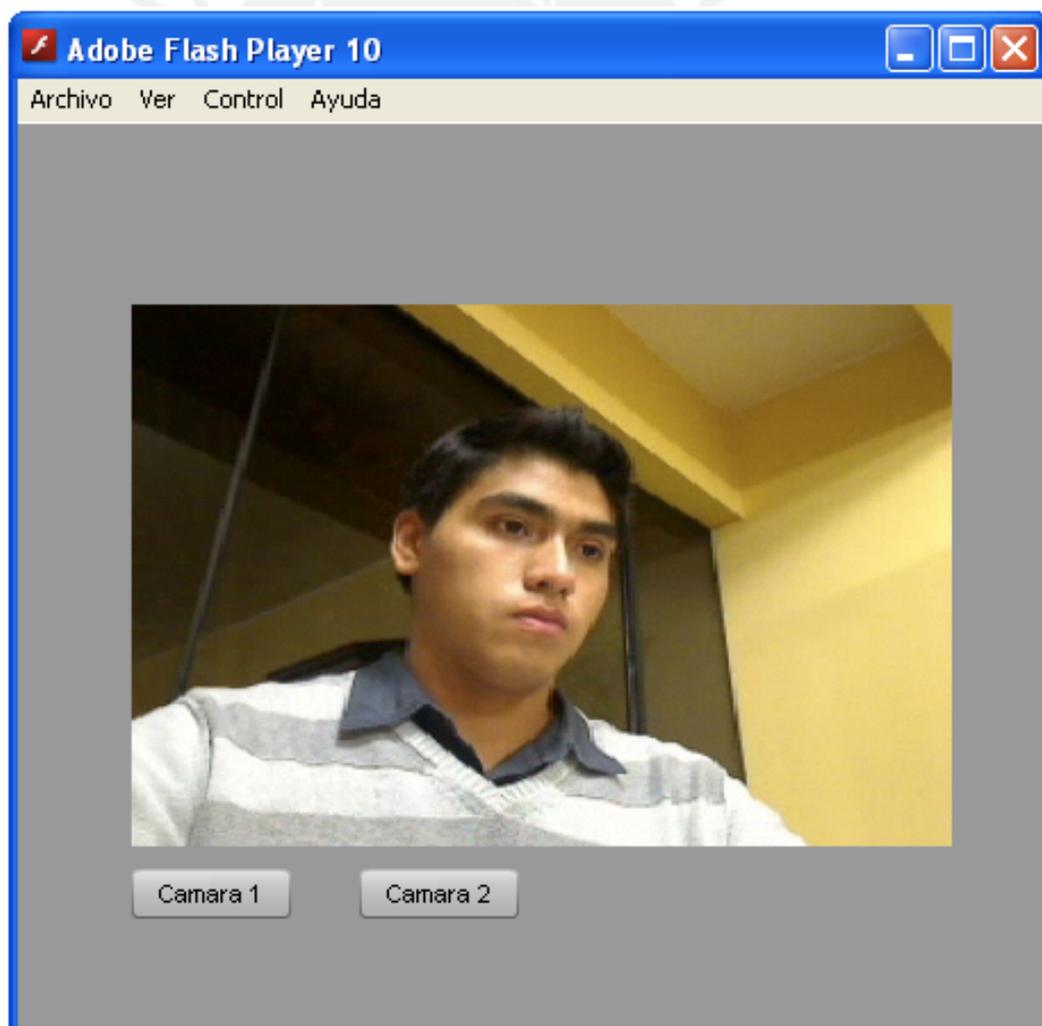


Figura 5 - 5 Interfaz de Video con múltiples cámaras 1



Figura 5 - 6 Interfaz de Video con múltiples cámaras 2

Estas han sido las aplicaciones que se han desarrollado para brindar la solución al problema planteado en este trabajo de tesis. A continuación se presentará como se ha llegado a integrar estas aplicaciones de modo que se pueda contar con una interfaz completa de comunicación para el laboratorio remoto.

5.1.2 Integración de componentes

La integración de las aplicaciones que previamente se han desarrollado corresponde a la unión de la interfaz visualizadora de video *streaming* con la herramienta de Chat. Estas en un principio han sido elaboradas independientemente una de la otra sin embargo para este trabajo de tesis es

necesario que el usuario cuente con estas en una interfaz unificada. A continuación se describirá el proceso que se llevo a cabo para poder fusionar estas aplicaciones, así como la adición del visualizador de las guías de laboratorio y la integración de estas en un página web.

Para llevar a cabo la integración de las dos aplicaciones desarrolladas se debe considerar cuales son las funciones, métodos y variables que son comunes para ambas para llevar a cabo el funcionamiento de esta. Es de esta manera que se identifican los puntos claves del código que se podrán compartir y así no realicen acciones redundantes.

Dado que la mayoría funciones de estas aplicaciones son completamente distintas lo que se llevo a cabo es la adición del código de ambas en un nuevo archivo **.as**. Así pues, en este nuevo archivo se colocan todas las funciones tales como se encontraban en las originales. Por otro lado, los elementos comunes que se deben considerar para llevar a cabo son los métodos que permiten realizar la conexión con el servidor de medios además de las variables que se emplean en estos. Es por ello que esta nueva aplicación solo existe una función en donde se alojarán todas las instrucciones principales que llevaran a cabo el enlace con el servidor y el levantamiento de los componentes principales tales como el *stream* de video y el objeto compartido. Para esto se tuvo una consideración especial con las variables que iban a servir a la interfaz integrada. Asimismo todas las variables y objetos ya sean comunes como independientes son instanciadas en la clase principal de la aplicación.

Por otro lado, otro elemento que se debe modificar es el archivo **.fla** el cual deberá tener un tamaño en el que se pueda alojar tanto los componentes de la interfaz visualizadora de video así como los de la aplicación de Chat, los cuales deben también estar presentes en la librería de la aplicación. Una vez que todas las funciones ya se encuentren en el nuevo archivo y todos los componentes en la librería se lleva a cabo la prueba de funcionamiento, obteniendo como resultado la interfaz que se presenta en la grafica 5-7. Como se observa en ella se encuentran todas las funcionalidades que se tenían en las aplicaciones previamente desarrolladas en el capítulo anterior. Por un lado tenemos la interfaz que reproduce el video *stream* con los botones de selección de las cámaras

Web, y más abajo se tiene la aplicación Chat. Cabe destacar también que esta interfaz detecta el ancho de banda de cada uno de los usuarios del laboratorio y muestra el *stream* de video más adecuado.

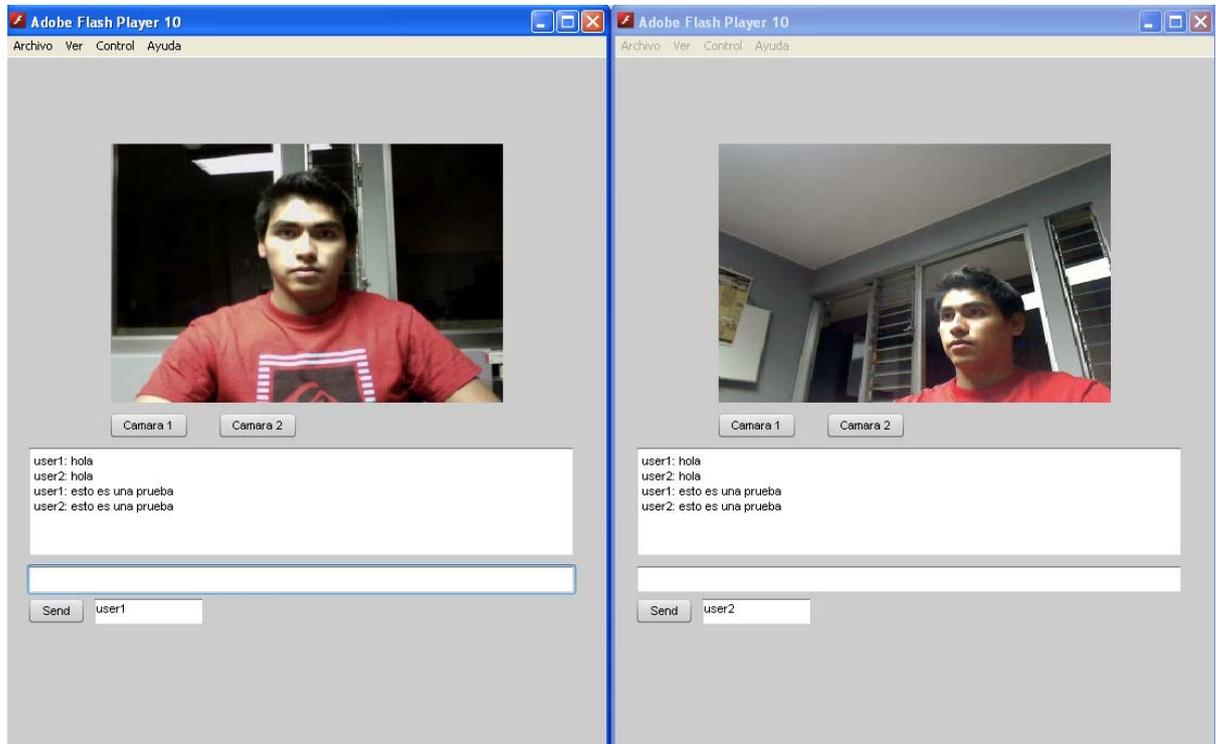
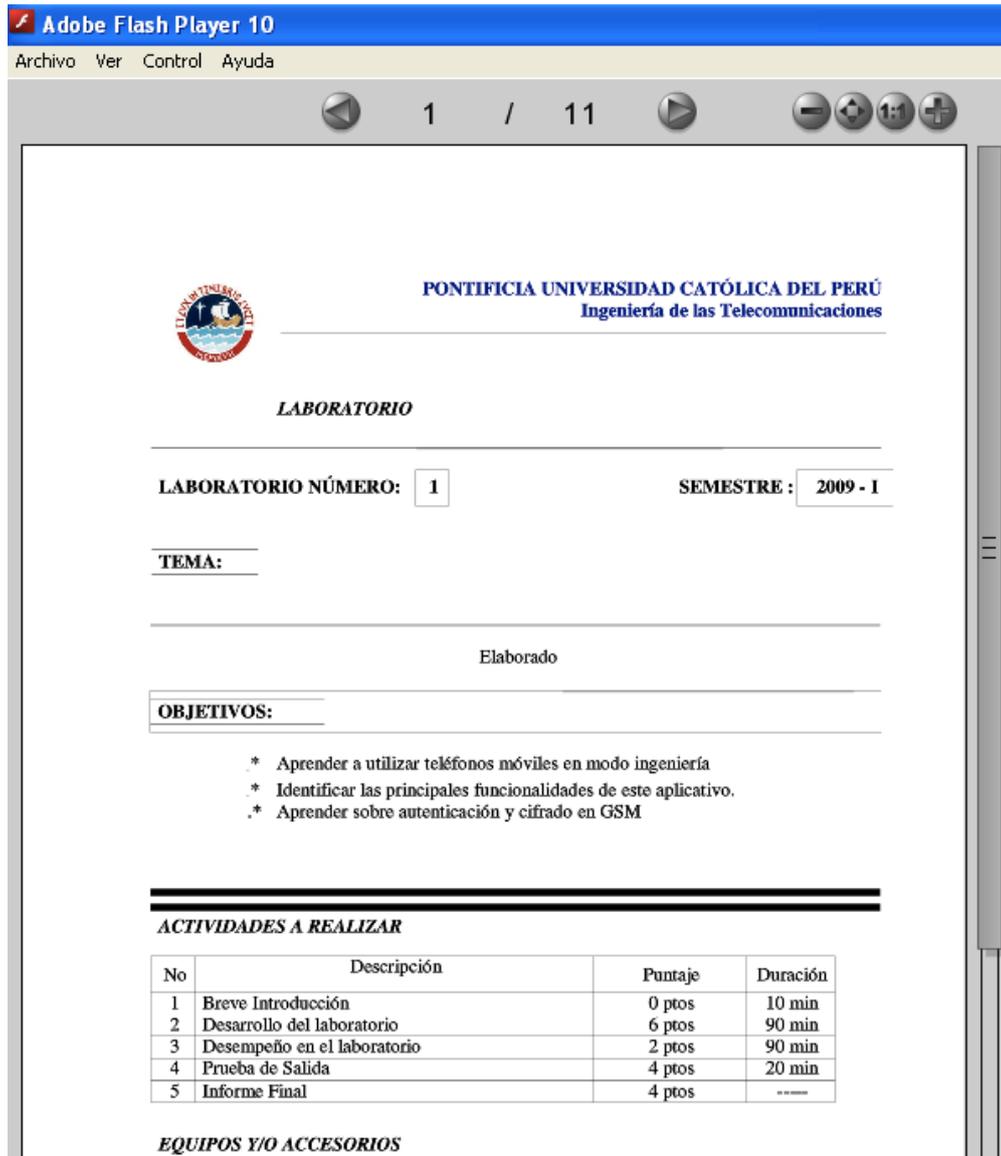


Figura 5 - 7 Interfaz integrada de video y Chat

Con respecto a la interfaz que permite visualizar las guías de laboratorio se llegó a utilizar una solución de libre disponibilidad llamada Flex Paper. Es a través de esta solución junto con la herramienta pdf2swf que se puede obtener un archivo swf a partir de un pdf para añadirlo a la página web del laboratorio. Permitiendo de esta manera que los alumnos puedan tener a la mano y a su disposición la guía de laboratorio para realizar la experiencia en línea. Cabe destacar que el proceso de integración de los módulos al sistema de laboratorio remoto no se encuentra dentro del alcance de esta tesis, para las pruebas el archivo swf que se obtiene se colocará en la misma página de prueba de la interfaz de video únicamente. En la gráfica 5-8 se puede apreciar un ejemplo de cómo se vería esta herramienta en el portal Web así como las herramientas de control que tiene.



Adobe Flash Player 10
Archivo Ver Control Ayuda

1 / 11

 **PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**
Ingeniería de las Telecomunicaciones

LABORATORIO

LABORATORIO NÚMERO: SEMESTRE :

TEMA:

Elaborado

OBJETIVOS:

- * Aprender a utilizar teléfonos móviles en modo ingeniería
- * Identificar las principales funcionalidades de este aplicativo.
- * Aprender sobre autenticación y cifrado en GSM

ACTIVIDADES A REALIZAR

No	Descripción	Puntaje	Duración
1	Breve Introducción	0 ptos	10 min
2	Desarrollo del laboratorio	6 ptos	90 min
3	Desempeño en el laboratorio	2 ptos	90 min
4	Prueba de Salida	4 ptos	20 min
5	Informe Final	4 ptos	...

EQUIPOS Y/O ACCESORIOS

Figura 5 - 8 Guía de laboratorio en formato swf

Por último es necesario para las pruebas que se van a realizar agregar los dos archivos .swf elaborados, por un lado el obtenido de la aplicación integrada de video con Chat y por el otro lado la guía de laboratorio de prueba convertida en este formato, a un archivo HTML de modo que pueda ser cargado en el servidor Web. Con esto se puede iniciar a realizar las pruebas en línea para analizar el desempeño del prototipo elaborado. Para llevar a cabo esto se diseñó un archivo en HTML en donde con ciertas instrucciones se introdujo en una página Web estos dos archivos. Lo principal para esta introducción es indicar las dimensiones de cada aplicación y la invocación determinando la ubicación de estas, para esto es preferible que estos .swf se encuentren dentro de la carpeta

donde también se almacena el HTML. Finalmente es necesario que los navegadores con los que se trabajará y evaluará la performance tengan instalados el plugin del Flash Media Player.

En la figura 5-9 se puede apreciar la interfaz de video y la guía de laboratorio dentro de una página web de pruebas.

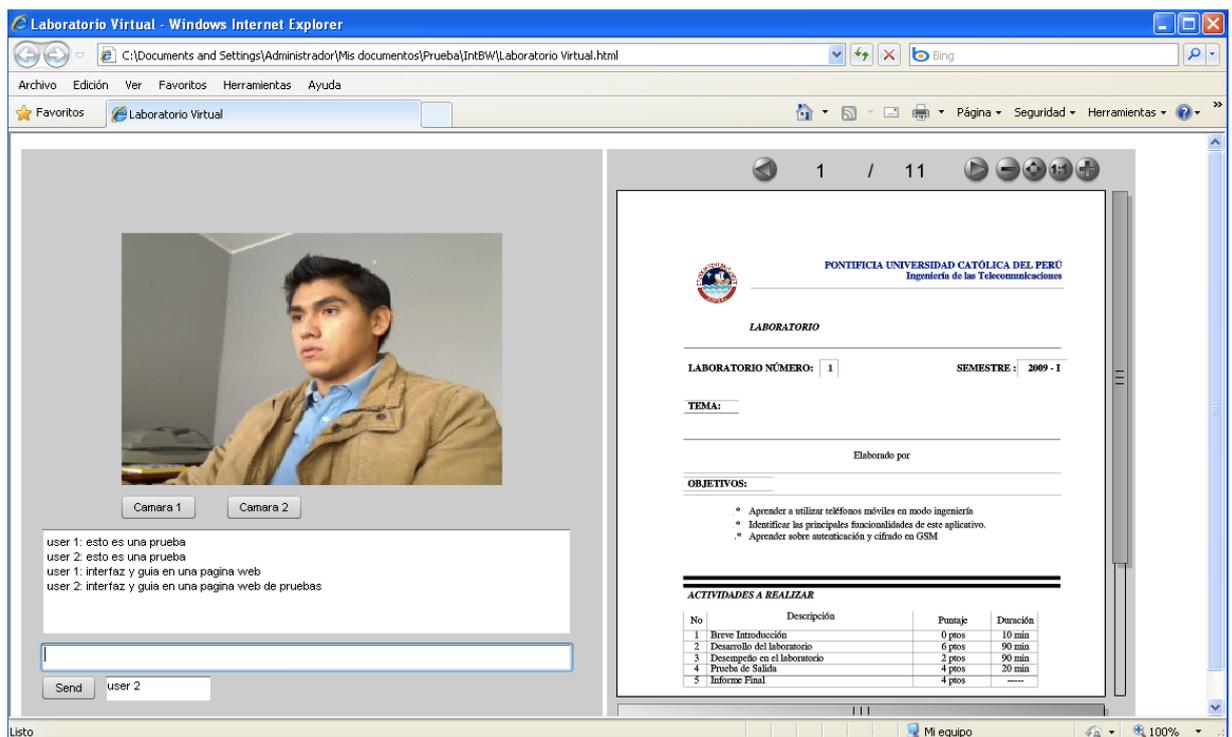
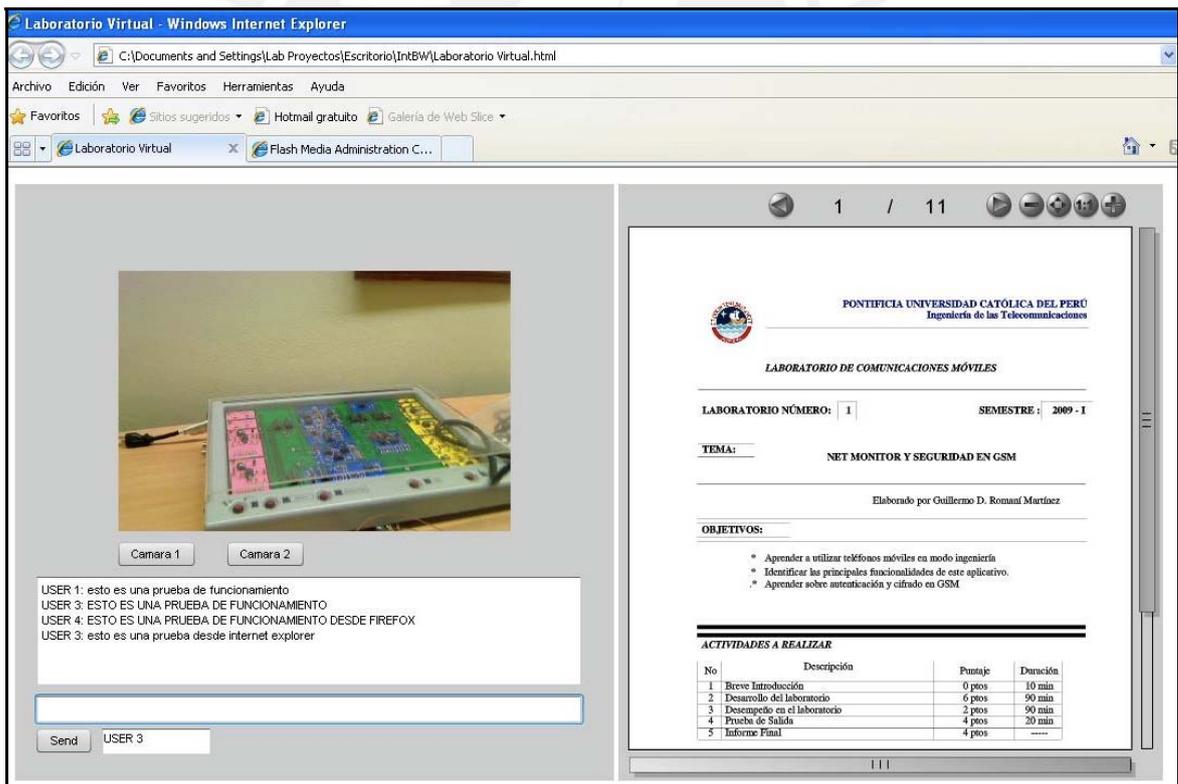


Figura 5 - 9 Interfaz y guía dentro de una página web

5.2 Evaluación del prototipo

Finalizada la integración de los componentes y ya obtenido el prototipo de la solución de esta tesis, lo que corresponde es poner a prueba ésta de modo que se pueda evaluar el funcionamiento de las aplicaciones y del servidor. Cabe destacar que las pruebas se realizaron con un servidor de medios ubicado en la red de la Universidad Católica, asignando una dirección IP externa la cual permita a cada cliente realizar una conexión a la interfaz y hacer uso del prototipo desde cualquier computador con Internet.

En este proceso se llevaron a cabo varias conexiones al servidor a través de la Internet de manera simultánea en diferentes computadoras y navegadores web. Dos de las conexiones principales al servidor correspondieron a las instancias de Flash Media Live Encoder que se utilizaron para cargar los contenidos de video al servidor de modo que puedan estar disponibles para su reproducción automática por medio de *streaming* en cualquier cliente que llegue a conectarse. Estas conexiones se ubicaron dentro de la red de la universidad con el propósito de que la transmisión de los contenidos de video sea ágil y fluida. Como se mencionó dado que el servidor con el que se trabajó era una versión gratuita de desarrollo solo obtuvieron como máximo diez conexiones simultáneas, es por ello que al realizar las pruebas solo se pudieron conectar ocho usuarios al mismo tiempo porque dos de ellos ya estaban siendo utilizados. A continuación se mostrará la captura de las pruebas realizadas en distintos navegadores web.



The screenshot shows a web browser window with the following content:

LABORATORIO DE COMUNICACIONES MÓVILES

LABORATORIO NÚMERO: SEMESTRE: 2009 - I

TEMA: NET MONITOR Y SEGURIDAD EN GSM

Elaborado por Guillermo D. Román Martaez

OBJETIVOS:

- * Aprender a utilizar teléfonos móviles en modo ingeniería
- * Identificar las principales funcionalidades de este aplicativo.
- * Aprender sobre autenticación y cifrado en GSM.

ACTIVIDADES A REALIZAR

No.	Descripción	Puntaje	Duración
1	Breve Introducción	0 pto	10 min
2	Desarrollo del laboratorio	6 pto	90 min
3	Desempeño en el laboratorio	2 pto	90 min
4	Prueba de Salida	4 pto	20 min
5	Informe Final	4 pto	-----

Figura 5 - 10 Interfaz evaluada utilizando Internet Explorer 6.0

Como se puede apreciar en esta figura y en las que a continuación se presentarán se comprueba el funcionamiento y compatibilidad de la interfaz de laboratorio virtual en cualquier plataforma de navegación web. Esto se debe a que en cada una de estas

solo es necesario que tengan instalados el aplicativo Flash Player de modo que se reproduzcan los archivos con formato .swf, los cuales son los que se desarrollan en este prototipo. Es importante mencionar que las pruebas realizadas se llevaron a cabo utilizando Internet Explorer, Mozilla Firefox, Google Chrome y Safari de Apple obteniendo en cada una de ellas el mismo resultado exitoso de funcionamiento.

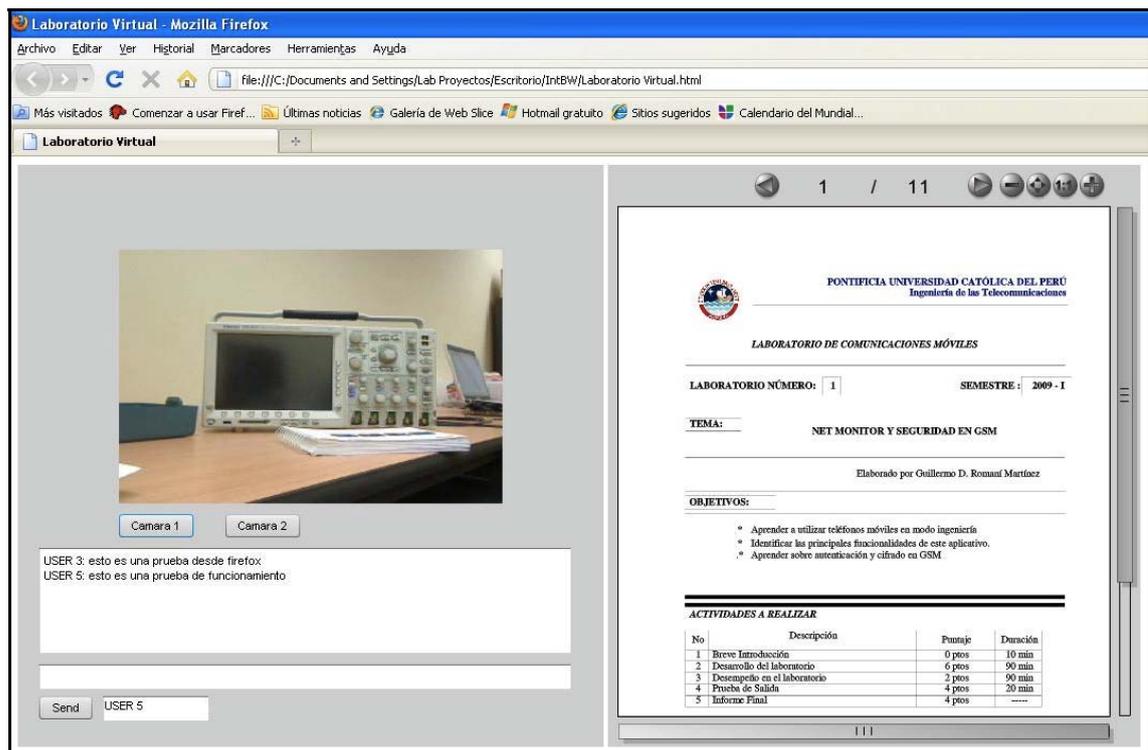


Figura 5 - 11 Interfaz evaluada utilizando Mozilla Firefox 3.5

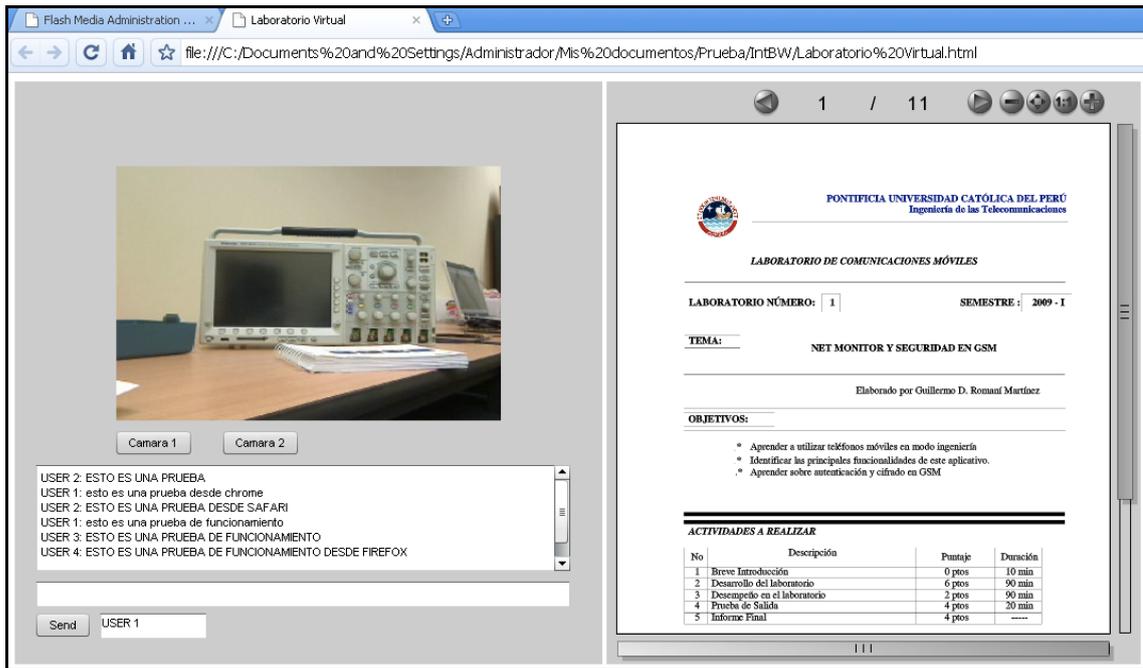


Figura 5 - 12 Interfaz evaluada utilizando Google Chrome 3.1

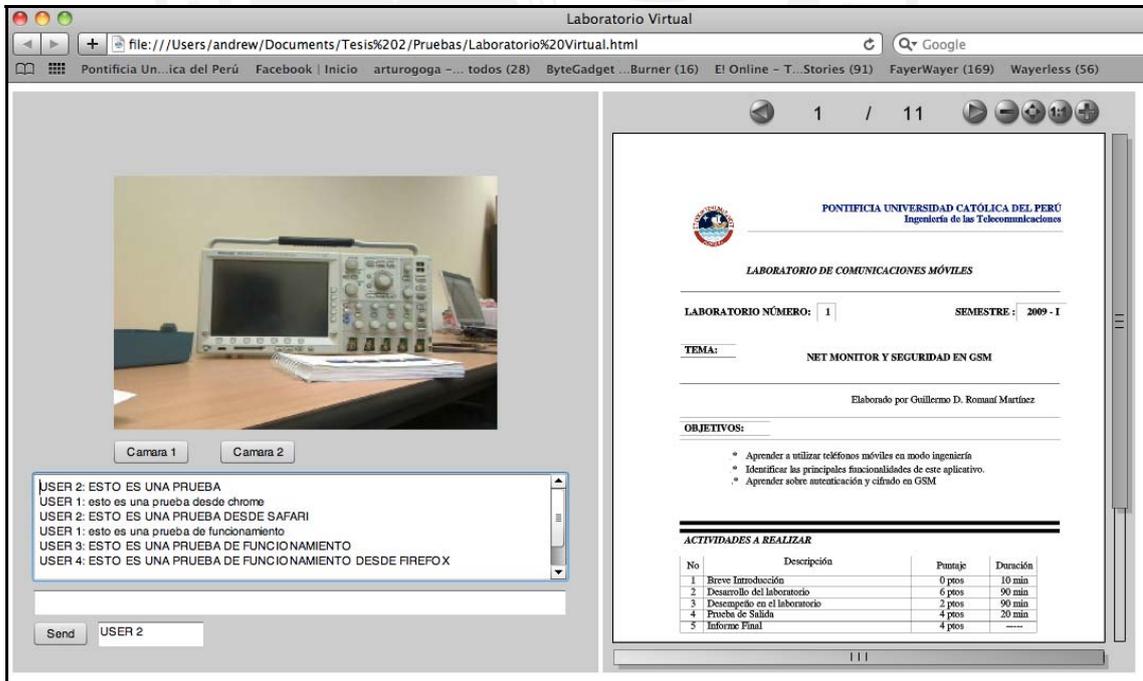
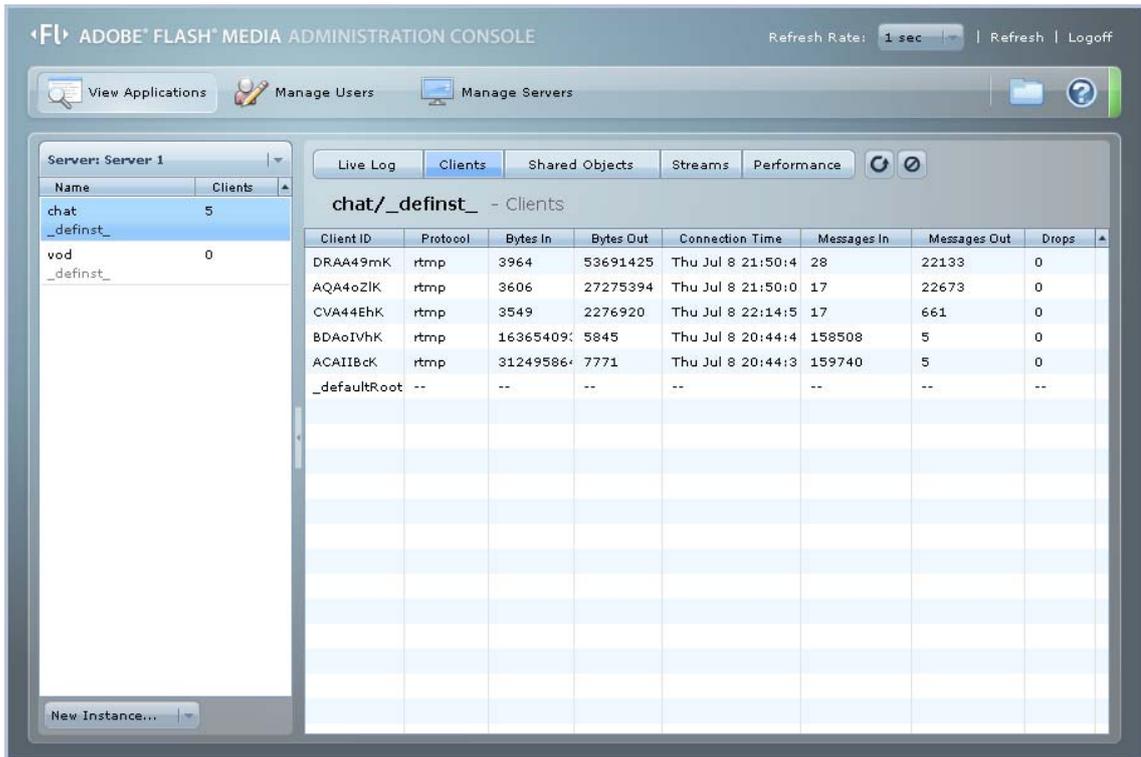


Figura 5 - 13 Interfaz evaluada en Safari 5

En cada uno de los navegadores que se utilizó, la interfaz evaluada funcionó sin ningún inconveniente. Así pues, cada una de las interfaces pudo reproducir el video *streaming* proveniente del servidor y entre los usuarios que se encontraron activos se estableció una comunicación por medio del Chat integrado. Cabe destacar, además, que cada usuario tuvo la independencia de seleccionar la cámara con el ángulo de visión que requirió visualizar.

Por otro lado, con respecto al servidor de video, se pudo acceder a la consola de administración remotamente para obtener los resultados de desempeño. En esta herramienta se brindó información acerca de todas las conexiones activas en el momento, los *streams* utilizados y videos cargados, los objetos compartidos creados, las aplicaciones a las que les está brindando servicio y el ancho de banda consumido tanto enviado como recibido. Esto permitió realizar un seguimiento detallado de la performance del servidor puesto en prueba. Más adelante se mostrarán algunas capturas realizadas de esta consola administradora durante la realización de las pruebas de desempeño.

Una vez que se lleva a cabo la conexión con el módulo administrativo la primera información obtenida es acerca de los clientes que se encuentran conectados al servidor. A través de esto se extrae el identificador que se le asigna a cada cliente conectado, el protocolo que llevó a cabo la conexión, la fecha y hora que realizó el enlace, la cantidad de bytes ingresados al servidor y también enviados por éste, el número de mensajes enviados y recibidos y todas las veces en las que ocurrió un fallo en el envío. En la figura 5-14 se puede apreciar una captura con esta información.



ADOBE® FLASH™ MEDIA ADMINISTRATION CONSOLE

Refresh Rate: 1 sec | Refresh | Logout

View Applications | Manage Users | Manage Servers

Server: Server 1

Name	Clients
chat _definst_	5
vod _definst_	0

chat/_definst_ - Clients

Client ID	Protocol	Bytes In	Bytes Out	Connection Time	Messages In	Messages Out	Drops
DRAA49mK	rtmp	3964	53691425	Thu Jul 8 21:50:4	28	22133	0
AQA4oZIK	rtmp	3606	27275394	Thu Jul 8 21:50:0	17	22673	0
CVA44EhK	rtmp	3549	2276920	Thu Jul 8 22:14:5	17	661	0
BDAoIVhK	rtmp	16365409	5845	Thu Jul 8 20:44:4	158508	5	0
ACAIIBcK	rtmp	31249586	7771	Thu Jul 8 20:44:3	159740	5	0
_defaultRoot	--	--	--	--	--	--	--

New Instance...

Figura 5 - 14 Consola de administración del FMS - Conexiones Existentes

Como se puede observar cada cliente del servidor tiene un identificador asignado por el sistema, asimismo se aprecia que se realizó la conexión por medio del protocolo RTMP, propietario de la solución. Del mismo modo, existe una diferencia evidente entre clientes con respecto al envío y recepción de bytes y es con esto con que se puede determinar cuales son los que han enviado los contenidos de video y cuales son los que han recibido. Así pues, los clientes con mayor cantidad de bytes ingresados son los que han sido originados a través del Flash Media Live Encoder y respectivamente los que tiene una mayor cantidad de bytes recibidos son los usuarios de la interfaz.

En la siguiente pestaña de la consola administradora se lleva un registro de los objetos compartidos por las aplicaciones existentes. Como se mencionó este elemento es el que permite establecer la comunicación a través del Chat y se mantiene disponible durante todo el tiempo en que la aplicación se encuentra levantada y activa en el servidor. En la figura 5-15 se presentará la captura de lo descrito.

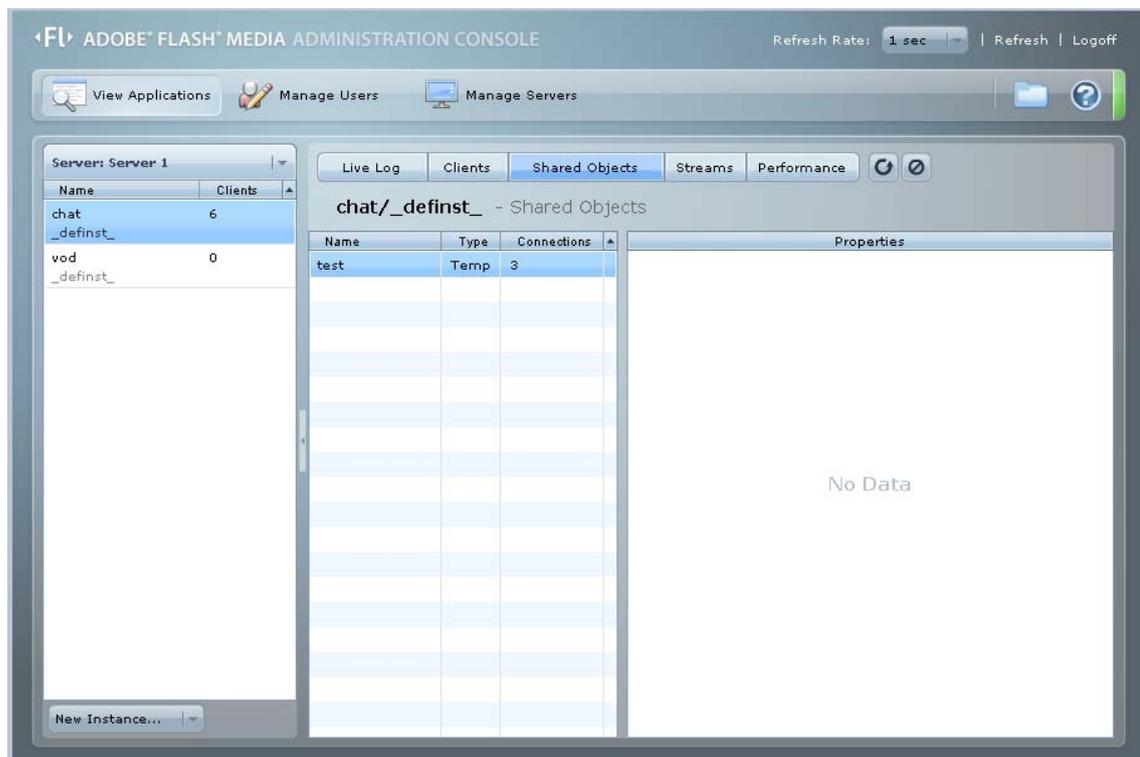


Figura 5 - 15 Consola de administración del FMS - Objetos Compartidos

En esta imagen se puede observar la información que se brinda acerca del nombre del objeto compartido, determinado en la aplicación desarrollada, el tipo de elemento creado, que en este caso es temporal ya que solo persiste mientras la aplicación esta levantada, y el numero de clientes que se encuentran haciendo uso de éste objeto.

Por otro lado, como se mencionó la consola también brinda información acerca de los *streams* activos. Estos pueden ser los que envían los contenidos a reproducir a los clientes conectados y los videos cargados por la herramienta Flash Media Live Encoder. Por medio de esto, de acuerdo al tipo de *stream*, podemos saber si el contenido es en vivo o grabado previamente, asimismo también es posible poder reproducirlo en la consola de manera que es posible llevar un mejor control sobre los contenidos permitidos dentro del servidor. La figura 5-16 muestra la captura de esta parte durante la evaluación del prototipo.

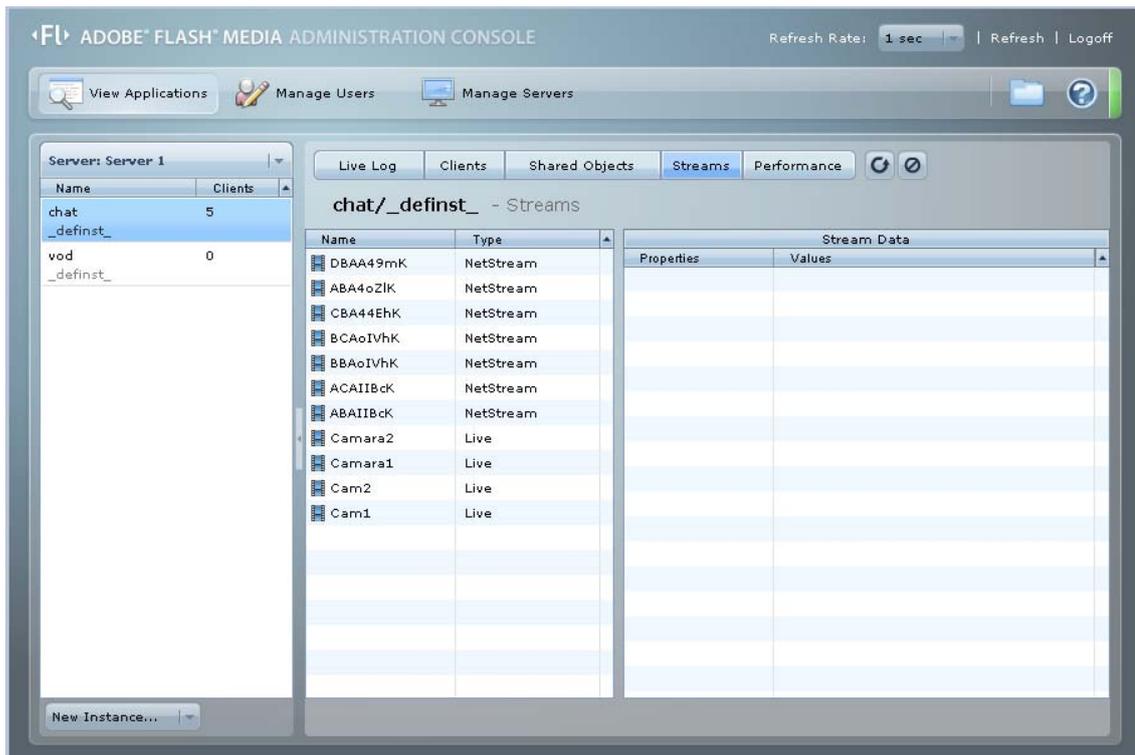


Figura 5 - 16 Consola de administración del FMS - *Streams* Activos

Como se puede apreciar en la captura, existen cuatro objetos de video en vivo, que provienen de la aplicación encargada de la codificación y subida al servidor, cuyos nombres fueron asignados también en esta. Luego se encuentran los *streams* activos los cuales tres de ellos han sido creados por los clientes conectados que se encuentran utilizando la interfaz y los cuatro restantes originados en el Flash Media Live Encoder para habilitar la carga de los contenidos de video. Así pues se demuestra que los *streams* no solo son para la reproducción sino también para la carga de videos.

Finalmente la última interfaz que muestra la consola de administración brinda información sobre la performance del servidor, así pues en esta se obtiene una gráfica del ancho de banda consumido por el servidor, las conexiones realizadas en un tiempo determinado y el uso de memoria y de procesamiento. La principal de estas es la de ancho de banda ya que permite observar la variación y oscilaciones del consumo, ya sea de entrada como de salida, en el tiempo en el que el servidor se encuentra activo. Con esto podemos determinar cuanto sería necesario tener de velocidad de conexión de modo que se pueda brindar a todos los usuarios del laboratorio un servicio óptimo.

A continuación en la figura 5-17 se presentará una captura tomada durante las pruebas de evaluación del servidor.

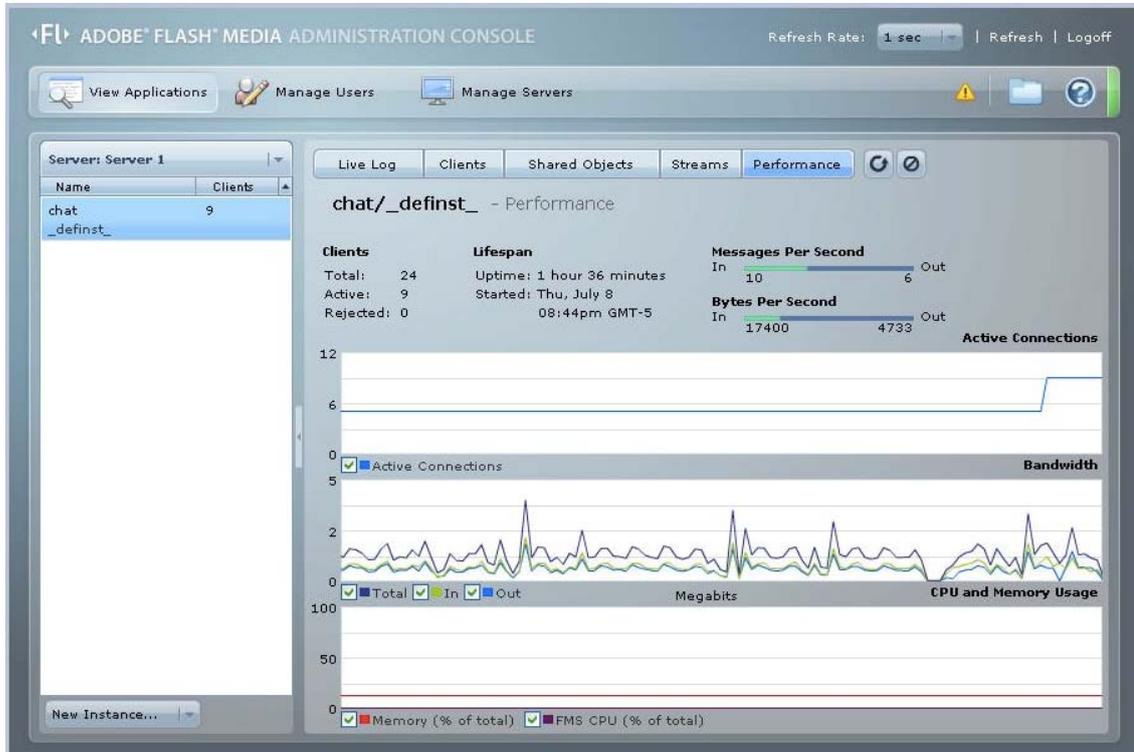


Figura 5 - 17 Consola de administración del FMS - Performance del Servidor

En la captura se muestra que el ancho de banda es el elemento del servidor que tiene mayor oscilación. Esto depende del número de conexiones que se estén llevando a cabo y a la vez de la estabilidad que tenga la red en la que se transporta la información. En este caso como se puede observar la tasa no excede de los 4 Mbps, sin embargo esto puede variar dependiendo de la calidad de los streams que se configuren en la aplicación Flash Media Live Encoder. Por otro lado, con respecto al consumo de memoria y procesamiento se aprecia que es estable por lo que no se tendría ningún inconveniente al respecto.

Con el propósito de realizar una estimación del ancho de banda utilizado por el prototipo para la transmisión de los contenidos de video se llevo a cabo un análisis de las velocidades de bitstream que se configuraron en la aplicación Flash Media Live Encoder. Así pues se determino las velocidades de transmisión que se envían hacia el servidor y el rango de velocidades máximas que se envía a los clientes conectados en un determinado momento. La tabla 5-1 muestra los resultados obtenidos.

Tabla 5 - 1: Análisis de velocidad de transmisión de video

Detalle	Velocidad (kbps)
Bitrate de primer video (alta calidad)	350
Bitrate de primer video (baja calidad)	100
Bitrate Total de primer video	450
Bitrate de segundo video (alta calidad)	350
Bitrate de segundo video (baja calidad)	100
Bitrate Total de segundo video	450
Bitrate Total de contenidos de video enviados al servidor	900
Bitrate máximo entregado a un usuario	350
Bitrate mínimo entregado a un usuario	100
Bitrate máximo entregado a 8 usuarios conectados	2800
Bitrate mínimo entregado a 8 usuarios conectados	800
Rango de velocidad enviada a usuarios desde servidor	2800 - 800

Como se aprecia en la tabla la velocidad total de transmisión que se envía al servidor de medios es de 900 kbps que proviene de la suma de velocidades de los dos videos con los que se trabaja en el prototipo que representa un diferente ángulo de visión dentro del laboratorio. Por otro lado, los contenidos que se envían a los clientes conectados varía en un rango entre 2800 a 800 kbps ya que esto depende del ancho de banda con que el usuario se conecte. Si todos los conectados tienen un ancho de banda rápido entonces a todos se les entregaría el video con un bitstream de 350 kbps, caso contrario de tener una baja velocidad el video a entregar sería el de 100 kbps. Cabe destacar que el rango de velocidades obtenida esta delimitada por el número de conexiones máximas que se pueden tener con el servidor de video, dado que es una versión de desarrollo solo se consideran ocho conexiones concurrentes ya que dos de ellas son utilizadas por las dos instancias de Flash Media Live Encoder.

La ultima parte de la evaluación se llevo a cabo utilizando la herramienta Wireshark [WIR2010], con esta se pudo capturar las tramas que se enviaban y recibían y determinar los protocolos presentes en el establecimiento de la conexión y durante todo el tiempo en que la aplicación se mantuvo activa. Previamente, en el segundo

capítulo se mencionaron algunos protocolos con los que probablemente se trabajaría. Luego de analizar las tramas capturadas se comprobó que los protocolos presentes para esta solución son TCP y RTMP. Estos se encargaron de enviar y recibir los contenidos de video y los mensajes del Chat a través de un único puerto de enlace, lo cual resolvió el problema de los NATs al facilitar la identificación de los usuarios conectados a la interfaz.

Por un lado el protocolo TCP es el que establece mantiene la conexión con el servidor, manteniendo un puerto de conexión abierto durante el tiempo que la interfaz que se encuentre activa. Por otra parte, el protocolo RTMP es el encargado de enviar los paquetes que contiene el video *streaming*, así pues estos transportan todos los datos esenciales para el funcionamiento de la aplicación. Cabe destacar que este último para transportarse, primero debe encapsularse dentro del TCP. A continuación dos figuras con las tramas capturadas con Wireshark.

63334	993.463092	192.168.35.248	192.168.35.240	TCP	exbit-escp > macromedia-fcs [ACK] Seq=4614 Ack=42994575 win=65535 Len=0
63336	993.463219	192.168.35.248	192.168.35.240	TCP	exbit-escp > macromedia-fcs [ACK] Seq=4614 Ack=42994575 win=65535 Len=0
<ul style="list-style-type: none"> ▣ Frame 63334 (54 bytes on wire (54 bytes captured)) ▣ Ethernet II, Src: usi_48:ef:f1 (00:1a:6b:48:ef:f1), Dst: Vmware_04:8b:92 (00:0c:29:04:8b:92) ▣ Internet Protocol, Src: 192.168.35.248 (192.168.35.248), Dst: 192.168.35.240 (192.168.35.240) ▣ Transmission Control Protocol, Src Port: exbit-escp (1316), Dst Port: macromedia-fcs (1935), Seq: 4614, Ack: 42994575, Len: 0 <ul style="list-style-type: none"> ▣ Source port: exbit-escp (1316) ▣ Destination port: macromedia-fcs (1935) ▣ [Stream index: 1] ▣ Sequence number: 4614 (relative sequence number) ▣ Acknowledgement number: 42994575 (relative ack number) ▣ Header length: 20 bytes ▣ Flags: 0x10 (ACK) ▣ Window size: 65535 ▣ Checksum: 0xc933 [validation disabled] ▣ [Seq/Ack analysis] 					

Figura 5 - 18 Trama de TCP capturada

55323	843.807357	192.168.35.240	192.168.35.248	RTMP	Ping Video data
55330	843.807828	192.168.35.240	192.168.35.248	RTMP	Ping
55332	843.926215	192.168.35.240	192.168.35.248	RTMP	Ping Video data
<ul style="list-style-type: none"> ▣ Frame 55323 (1514 bytes on wire, 1514 bytes captured) ▣ Ethernet II, Src: Vmware_04:8b:92 (00:0c:29:04:8b:92), Dst: usi_48:ef:f1 (00:1a:6b:48:ef:f1) ▣ Internet Protocol, Src: 192.168.35.240 (192.168.35.240), Dst: 192.168.35.248 (192.168.35.248) ▣ Transmission Control Protocol, Src Port: macromedia-fcs (1935), Dst Port: exbit-escp (1316), Seq: 36593546, Ack: 4589, Len: 1460 ▣ Real Time Messaging Protocol (Ping) <ul style="list-style-type: none"> ▣ RTMP Header (Ping) ▣ RTMP Body ▣ Real Time Messaging Protocol (Video Data) <ul style="list-style-type: none"> ▣ RTMP Header (Video Data) ▣ RTMP Body 					

Figura 5 - 19 Trama de RTMP capturada

Conclusiones y Avances Futuros

Conclusiones

A continuación se muestran las conclusiones a las que se han llegado luego del trabajo realizado:

- Se logró desarrollar una interfaz multimedia interactiva basándose en un servidor de video *streaming* para la implementación de un laboratorio virtual.
- Se realizó un adecuado estudio de cada una de las posibles tecnologías a utilizar.
- Se analizó cuáles son las características y requerimientos de la aplicación para posteriormente elegir las herramientas con las que se desarrollaría la interfaz.
- Se diseñó una arquitectura de un sistema de video *streaming* basado en el servidor de video Flash Media Server.

- Se configuró la herramienta de Flash Media Live Encoder para que controle un grupo de cámaras web, capture el video y lo codifique para cargarlo al servidor de medios.
- Se diseñó e implementó una aplicación en flash para la reproducción en vivo de videos que muestren los equipos y herramientas existentes en el laboratorio remoto.
- Se logró que la interfaz de video pueda trabajar simultáneamente con varias cámaras web posibilitando al usuario la visualización de diferentes ángulos y equipos del laboratorio remoto.
- Se consiguió que la interfaz de video obtenga el ancho de banda de cada usuario conectado para de esta manera entregarle la resolución de video más adecuada para su reproducción.
- Se diseñó e implementó una aplicación de Chat que permita la comunicación entre los alumnos presentes en una experiencia del laboratorio virtual.
- Se logró convertir las guías de laboratorio, por medio de una herramienta, en un formato que permita visualizarlas en la interfaz del laboratorio virtual.
- Se realizó una evaluación de la interfaz implementada, el servidor utilizado y los protocolos utilizados por medio de pruebas en Internet.

Avances Futuros

La interfaz desarrollada es el elemento que permite al usuario visualizar los equipos y herramientas que se encuentran en el laboratorio que se accede remotamente, y al mismo tiempo establecer una comunicación entre todos los usuarios conectados a éste. Así pues, este compone solo una parte de la totalidad del proyecto. Los avances futuros guardan relación con el enlace entre los elementos del proyecto con la aplicación elaborada en este trabajo de tesis.

- Implementación de la aplicación desarrollada en un servidor Web que brinde el acceso a esta interfaz desde el portal del laboratorio remoto virtual.
- Integración de la interfaz con el sistema de administración del laboratorio y con los equipos de laboratorio, permitiendo al usuario ingresar instrucciones para trabajar directamente con los equipos y realizar una experiencia de laboratorio.
- Integración de la aplicación de Chat con la base de datos del sistema para poder obtener el nombre de los usuarios del laboratorio.
- Acciones adicionales dentro de la interfaz como aspectos de diseño para la uniformizar el portal del laboratorio.

Bibliografía

- [AUS2005] AUSTENBERRY, DAVID. The Technology of Video and Audio Streaming. Elsevier. 2005
- [BRO2008] BROWN, CHARLES E. Essential Guide to Flex 3. Friends of Ed. 2008
- [CAB2006] CABERO, Julio (2006). «Bases pedagógicas del *e-learning*». Revista de Universidad y Sociedad del Conocimiento (RUSC) [artículo en línea]. Vol. 3, n.º 1. UOC. [Fecha de consulta: 05/10/09].
<http://www.uoc.edu/rusc/3/1/dt/esp/cabero.pdf>
- [CAS2009] VIEWCAST BROADENS H.264 SUPPORT WITH ADOBE-BASED SOLUTIONS. (cover story)." Computer Protocols 23.5 (2009): 1-2. Computers & Applied Sciences Complete. EBSCO. Web. Recuperado 15 de Noviembre del 2009.
- [CDE2009] HTML 5. *Computer Desktop Encyclopedia*, 1. Retrieved from Computers & Applied Sciences Complete database. Consulta de la base de datos EBSCO el 15/12/09.
- [CLA2009] CLARKE, JIM. JavaFX Developing Rich Internet Applications. Addison-Wesley 2009
- [COL2007] COLINSKY, CHRIS (2007) Silverlight Is No Flash Killer Streaming Media Magazine; Aug/Sep 2007; ABI/INFORM Global pg. 103 Base de Datos ProQuest. Recuperado el 19 de Octubre del 2009
- [FAR2005] FARRÉ LÓPEZ, XAVIER. Tesis de Grado : "Rich Internet Applications". Universitat Politècnica de Catalunya, España 2005.
<http://hdl.handle.net/2072/17546>

- [GAR2002] GARCÍA ZUBÍA, JAVIER (2002). Diseño de laboratorios remotos virtuales: WebLab. [Versión electrónica]. Recuperada el 8 de noviembre de 2009. Disponible en:
<http://bioinfo.uib.es/~joemiro/aenui/procJenui/Jen2005/gardise.pdf>
- [HAN2010] HANLEY, MICHAEL. E-learning Curve Blog. Publicación en línea extraído de: <http://michaelhanley.ie/elearningcurve/m-learning-via-the-iphone-3-some-approaches-and-technologies/2009/02/20/> Recuperado el 22 de Octubre del 2009
- [JAV2007] "Protocols Guide: Voice over IP and VOIP Protocols: Signalling: RTSP: Real-Time Streaming Protocol." 113-114. Javvin Technologies, Inc., 2007. Computers & Applied Sciences Complete. EBSCO. Web. 15 Nov. 2009.
- [KIN2009] Kineo and UFI/Learndirect (2009) Mobile Learning Reviewed. Disponible en: http://www.kineo.com/documents/Mobile_learning_reviewed_final.pdf Recuperado el 22 de Octubre del 2009
- [MCH2009] MICHAEL, ASHRAF. Essential Silverlight 3. Addison-Wesley 2009
- [NOD2005] NODA, TOM. Rich Internet Applications. University of Wisconsin-Madison. Recuperado el 15 de Noviembre del 2009 de Google Scholar <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.119.4482&rep=rep1&type=pdf>
- [RIJ2004] RIJO SCIARA, DANIEL. Fundamentos de Video Streaming. Monografía de curso "Codificación de Imágenes y Video" Universidad de la Republica Montevideo. Recuperado 6 de Enero del 2010.
<https://iie.fing.edu.uy/ense/asign/codif/material/monografias/2004-01.pdf>
- [ROS2001] ROSENBERG, M.J. (2001). *E-learning. Strategies for Delivering Knowledge in the Digital Age*. New York: McGraw-Hill.
- [SAN2008] SANDER, W. *Learning Flash® Media Server 3*. O'Reilly Media. 2008

- [SIM2008] SIMPSON, WES. Video Over IP. Elsevier 2008
- [SUN2007] Sun, H., Vetro, A., & Xin, J. An overview of scalable video streaming. *Wireless Communications & Mobile Computing*, 7(2), 159-172. Recuperado 8 de Enero del 2010.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.590&rep=rep1&type=pdf>
- [TAF2007] TAFT, DARRYL K. (2007) Silverlight gets versatile. eWeek; 5/7/2007, Vol. 24 Issue 16, p11-12, EBSCO. Web Recuperado 19 de Noviembre del 2009
- [ADO2009] Adobe Systems
<http://www.adobe.com> Visitado 14/01/12
- [DEV2010] Herramienta Flexpaper
<http://flexpaper.devaldi.com/> Visitado 09/07/12
- [ECB2009] La "nueva televisión" es de todos. Vida y Futuro. El Comercio Perú.
<http://blogs.elcomercio.pe/vidayfuturo/2009/10/la-nueva-television-es-de-todo.html>
Visitado 14/01/12
- [ECU2009] Diez millones vieron en vivo a U2 en YouTube. El Comercio Perú.
<http://elcomercio.pe/noticia/361892/diez-millones-vieron-vivo-u2-youtube>
Visitado 14/01/12
- [FLA2009] Adobe Flash Player
<http://get.adobe.com/es/flashplayer/?promoid=BUIGP> Visitado 14/01/12
- [IBM2009] New Elements in HTML 5
<http://www.ibm.com/developerworks/library/x-html5/?ca=dgr'-Inxw01NewHTML> Visitado 16/01/12

- [JAV2009] JavaFX
<http://www.javafx.com> Visitado 14/01/12
- [LOG2010] Logitech
<http://www.logitech.com> Visitado 14/01/12
- [MIC2009] Microsoft
<http://www.microsoft.com> Visitado 14/01/12
- [MIT2009] Massachusetts Institute of Technology
<http://web.mit.edu/> Visitado 14/01/12
- [PUC2009] PUCP Virtual
<http://www.pucp.edu.pe/content/pagina40.php?pid=3186&pidSeccionWeb=4&pidReferencial=> Visitado 13/12/11
- [RED2009] RED5
<http://osflash.org/red5> Visitado 14/01/12
- [RTM2009] RTMP. Adobe Systems
<http://www.adobe.com/devnet/rtmp/> Visitado 14/01/12
- [SIL2009] Silverlight
<http://silverlight.net/> Visitado 14/01/12
- [SWF2010] SwfTools
<http://www.swftools.org/> Visitado 09/01/12
- [TAR2009] A Comprehensive Review on JavaFX
<http://blog.taragana.com/index.php/archive/a-comprehensive-review-on-javafx-the-latest-java-innovation/> Visitado 11/01/12
- [UNE2009] Automat Labs
<http://lab.dia.uned.es/automatlab/> Visitado 14/01/12

[WIR2010] Wireshark
<http://www.wireshark.org> Visitado 09/01/12

[WOW2009] WOWZA MEDIA SYSTEMS.
<http://www.wowzamedia.com/> Visitado 14/01/12

