

**PROTOTIPO DE UNA APLICACIÓN WEB PARA REALIZAR STREAMING DE VIDEO A TRAVÉS DE UNA RED, UTILIZANDO EL PROTOCOLO RTMP (REAL TIME MESSAGING PROTOCOL).**

**ROQUE ALEXANDER LOZANO LOZANO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES  
BUCARAMANGA  
2011**

**PROTOTIPO DE UNA APLICACIÓN WEB PARA REALIZAR STREAMING DE VIDEO A TRAVÉS DE UNA RED, UTILIZANDO EL PROTOCOLO RTMP (REAL TIME MESSAGING PROTOCOL).**

**ROQUE ALEXANDER LOZANO LOZANO**

**Trabajo de Grado presentado como requisito  
para optar al título de Especialista en Telecomunicaciones**

**Director:  
Mag. SANDRA CRISTINA SANGUINO GALVIS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES  
BUCARAMANGA  
2011**

## CONTENIDO

	Pág.
INTRODUCCIÓN	10
OBJETIVO GENERAL	11
OBJETIVOS ESPECIFICOS	11
METODOLOGÍA	12
1. CAPITULO 1	17
1.1 CONCEPTOS BÁSICOS	17
1.1.1 Arquitectura de un Sistema de Streaming de Video	18
1.1.2 Codecs de Compresión de video	19
1.1.3 Protocolos de Streaming	20
1.2 DESCRIPCIÓN DE PLATAFORMAS PARA EL DESARROLLO DE APLICACIONES DE STREAMING DE VIDEO.	25
1.2.1 Plataforma Adobe Flash Media Server	25
1.2.2 RED 5	26
1.2.3 Wowza Media Server Pro	27
1.2.4 Unreal Media Server	27
1.2.5 Helix Universal Server	27
1.2.6 Windows Media Server	28
1.2.7 Darwin Streaming Server	28
1.2.8 QuickTime Streaming Server	28
CAPITULO 2	29
2.1 ANÁLISIS Y DISEÑO	29
2.1.1 Entorno de Trabajo	29
2.1.3 Tecnologías.	29
2.1.4 Especificación	31
3. CAPITULO 3	33
3.1 DESARROLLO Y PRUEBAS	33
3.1.1 Desarrollo de la aplicación	33

3.1.2 Pruebas y resultados	35
4. CONCLUSIONES	42
BIBLIOGRAFÍA	43

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1. Estructura del proceso – Dos dimensiones [1].	13
Figura 2. Encabezado RTP.	21
Figura 3. Arquitectura utilizada en la aplicación [Autor].	30
Figura 4. Caso de uso ver video por demanda	31
Figura 5. Caso de uso emitir video en vivo	31
Figura 6. Caso de uso ver video en vivo	32
Figura 7. Módulo video por demanda	33
Figura 9. Módulo ver video en vivo	35
Figura 10. Comportamiento inicial del consumo de ancho de banda emitiendo video en vivo	36
Figura 11. Consumo de ancho de banda emitiendo video en vivo con alta calidad luego de cinco minutos	36
Figura 12. Consumo de ancho de banda emitiendo video en vivo con baja calidad	37
Figura 13. Consumo de ancho de banda emitiendo video en vivo con calidad media	38
Figura 14. Consumo de ancho de banda emitiendo video por demanda y video en vivo con cinco usuarios simultáneos	39
Figura 15. Consumo de ancho de banda emitiendo video por demanda y video en vivo con dos usuarios simultáneos	39
Figura 16. Consumo de ancho de banda emitiendo video por demanda con seis usuarios simultáneos	40
Figura 17. Consumo de ancho de banda al finalizar las pruebas	41

## RESUMEN

**TITULO:** PROTOTIPO DE UNA APLICACIÓN WEB PARA REALIZAR STREAMING DE VIDEO A TRAVÉS DE UNA RED, UTILIZANDO EL PROTOCOLO RTMP (REAL TIME MESSAGING PROTOCOL).

**AUTOR:** Roque Alexander Lozano Lozano \*\*.

**PALABRAS CLAVE:** Streaming, Protocolo RTMP, Aplicación Web, Server Media, Video por Demanda, Video en vivo, red IP.

**CONTENIDO:** La presente monografía presenta la arquitectura de un sistema de Streaming de video mostrando los diferentes protocolos, codecs y plataformas existentes, con el fin de determinar las ventajas de cada uno de ellos estableciendo cual es el más adecuado para enviar un flujo de video a través de una red IP.

Así mismo se dan a conocer las principales características y capacidades del protocolo RTMP (REAL TIME MESSAGING PROTOCOL) para la emisión de video en vivo y video por demanda. Para ello se propuso diseñar un prototipo de una aplicación Web, mediante la plataforma Flash Media Server. Con esta aplicación se realizaron pruebas, con el fin de comprobar la eficiencia y calidad del protocolo a la hora de emitir contenido de video. Durante las pruebas se pudo observar que el protocolo RTMP permite una transmisión rápida y eficiente de un flujo de video a través de una red cableada o inalámbrica ya sea de forma local o en Internet.

Adicionalmente se describe el proceso de diseño de la aplicación Web utilizando la metodología RUP (Rational Unified Process), describiendo cada una de las diferentes etapas que se deben realizar para desarrollar dicha aplicación, con el fin de dar los lineamientos necesarios para su implementación.

---

\* Proyecto de grado

\*\* Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Sandra Cristina Sanguino Galvis.

## ABSTRACT

**TITLE:** PROTOTYPE OF A WEB APPLICATION FOR PERFORMING VIDEO STREAMING OVER A NETWORK USING THE PROTOCOL RTMP (REAL TIME MESSAGING PROTOCOL).

**AUTHOR:** Roque Alexander Lozano Lozano\*\*.

**KEY WORDS:** Streaming, Protocol RTMP, Web Application, Server Media, Video On Demand, Live Video, IP Network.

**CONTENT:** This monograph presents the architecture of a streaming video system showing the different protocols, codecs and platforms, in order to determine the benefits of each setting that is most appropriate to send a video stream through an IP network.

Also disclosed are the main features and capabilities of the RTMP (Real Time Messaging Protocol) to broadcast live video and video on demand. This set out to design a prototype of a Web application using Flash Media Server platform. This application was tested, in order to prove the efficiency and quality of the protocol when making video content. During testing it was observed that the RTMP protocol allows a rapid and efficient transmission of streaming video through a wired or wireless network either locally or online.

Additionally is described the design process of the Web application using the RUP (Rational Unified Process), describing each of the different stages that must be done to develop this application in order to provide the necessary guidelines for implementation.

---

\* Project of grade

\*\* Faculty of Physical Mechanical Engineering. Electric, Electronic and Telecommunications School.  
Director: Sandra Cristina Sanguino Galvis.

## INTRODUCCIÓN

La emisión de contenido de video a través de Internet ofrece muchas ventajas tanto a empresas como a usuarios en general, permitiendo ofrecer servicios tales como videollamadas, videoconferencias, telemedicina, aplicaciones de seguridad entre otros. La transmisión del contenido de video se realiza ya sea por redes cableadas o inalámbricas, utilizando protocolos desarrollados para tal fin, además de una arquitectura cliente-servidor, en la que el servidor emite o envía el flujo de video a los usuarios (clientes) quienes a través de una aplicación podrán observarlo.

Es importante analizar las características más importantes de las diferentes herramientas y protocolos para realizar streaming con el fin de determinar sus ventajas de cada uno de ellos a fin de establecer cual de ellos es el más adecuado para desarrollar una aplicación a través de la cual lograr enviar un flujo de video a través de una red.

Además contando con una aplicación que permita la emisión de video se pueden realizar pruebas de rendimiento sobre un protocolo específico en este caso especial para el proyecto es el protocolo RTMP (Real Time Messaging Protocol) de una manera mucho más sencilla, ordenada y eficiente.

## **OBJETIVO GENERAL**

Diseñar un prototipo de aplicación Web para realizar streaming de video a través de una red, utilizando el protocolo RTMP (Real Time Messaging Protocol).

## **OBJETIVOS ESPECIFICOS**

- Identificar las diferencias entre las plataformas existentes para el desarrollo de aplicaciones de streaming de video, con el objetivo de tener un marco de referencia sobre esta tecnología.
- Construir un prototipo de aplicación Web de streaming de video, utilizando la plataforma Flash.
- Ejecutar pruebas de comportamiento del protocolo RTMP, a través del prototipo desarrollado.

## METODOLOGÍA

La metodología usada para el desarrollo de la aplicación del presente proyecto está basada en Rational Unified Process (RUP) [1], el cual es un proceso de ingeniería de software que proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de un grupo de desarrollo. Su objetivo es garantizar la producción de software de alta calidad que satisfaga las necesidades de los usuarios finales dentro de un periodo de tiempo calculado. RUP reúne muchas de las mejores prácticas para el desarrollo de software en una forma que es conveniente para una amplia gama de proyectos y organizaciones.

Este proceso posee tres características específicas [1]:

- **Casos de Uso:** El proceso utiliza Casos de Uso para manejar el proceso de desarrollo desde el inicio (inception) hasta el Despliegue.
- **Centrado en Arquitectura:** El proceso busca entender los aspectos estáticos y dinámicos más significativos en términos de arquitectura de software. La arquitectura se define en función de las necesidades de los usuarios y se determina a partir de los Casos de Uso.
- **Iterativo e Incremental:** El proceso muestra que es práctico dividir grandes proyectos en proyectos más pequeños o mini-proyectos. Cada mini-proyecto comprenderá una iteración que resulta en un incremento. Una iteración puede abarcar la totalidad de los flujos del proceso. Las iteraciones son planificadas en base a los Casos de Uso.

### **Estructura del proceso.**

El proceso consta de dos estructuras o dos dimensiones como se muestran en la imagen, en la cual el eje horizontal representa tiempo y muestra los aspectos del ciclo de vida del proceso, mientras que el eje vertical representa los flujos de trabajo fundamentales del proceso.

La primera dimensión representa el aspecto dinámico del proceso y se expresa en términos de ciclos, fases, iteraciones e hitos.

La segunda dimensión representa el aspecto estático del proceso: su descripción en términos de componentes de procesos, actividades, flujos de trabajo, artefactos y trabajadores.

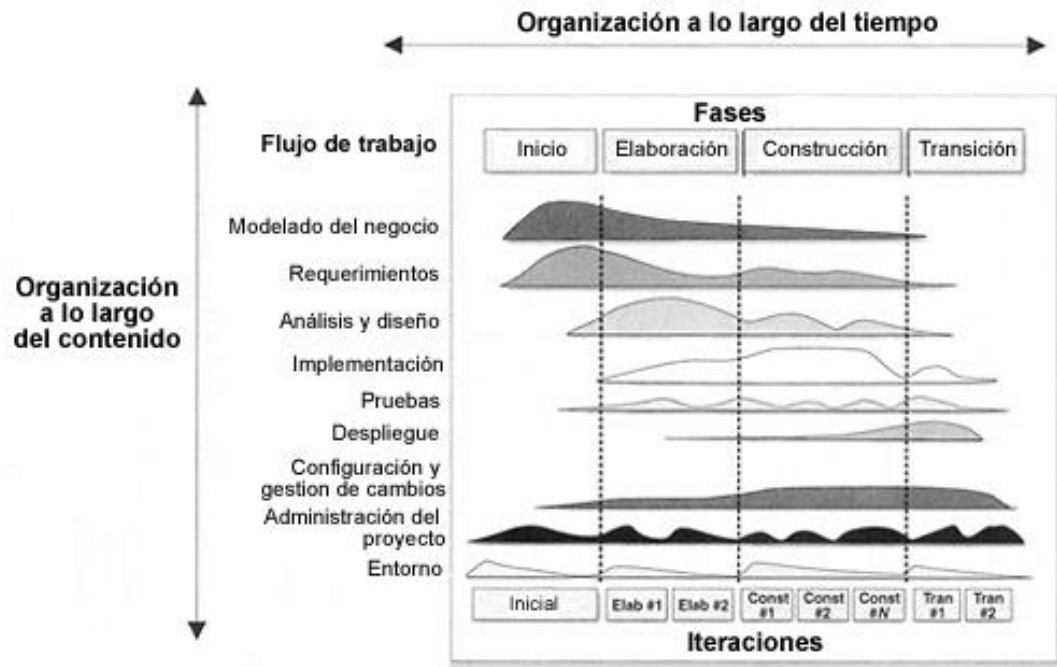


Figura 1. Estructura del proceso – Dos dimensiones [1].

Como se puede observar en la gráfica este proceso se basa en ciclos que pueden repetirse a lo largo del ciclo de vida del sistema. Cada ciclo consta de cuatro fases que son, Inicio, Elaboración, Construcción y Transición [1].

- Inicio: En esta fase, se definen y analizan los objetivos del proyecto, así como la viabilidad y el alcance del mismo.
- Elaboración: En esta fase se definen y se especifican los casos de uso, además se diseña la arquitectura básica del sistema.
- Construcción: En esta fase se realizan las tareas de análisis, diseño e implementación permitiendo realizar la documentación del sistema además de proporcionar un producto desarrollado.
- Transición: En esta fase se verifica que los objetivos se hayan cumplido de forma satisfactoria. Se entrega el producto al usuario final para uso. Además la documentación es revisada para completar el manual de usuario.

### Flujos de Trabajo [1].

- Modelado del Negocio: Describe cómo obtener una visión de los objetivos de la organización, y con base en esta visión, lograr definir los procesos, roles y

responsabilidades de esta en un modelo de negocio. Este modelado utiliza los casos de uso con el objeto de describir los procesos de la organización.

- **Requerimientos:** Describe cómo definir detalladamente los requisitos del sistema y convertirlos en un modelo de casos de uso. Además, permite manejar el alcance y la evolución de las necesidades del sistema, así como estimar el costo y el tiempo de desarrollo del mismo.
- **Análisis y Diseño:** Su objetivo es convertir los requisitos en una especificación que permita describir cómo se implementará el sistema. Para realizar esta conversión se deben entender claramente los requerimientos y transformarlos en un diseño de sistema a través de la elección de la mejor estrategia de implementación. También se debe tener en cuenta que al inicio del proyecto se debe establecer una arquitectura robusta, con el fin de poder diseñar un sistema que sea fácil de entender, construir y evolucionar. Este diseño debe ajustarse al entorno donde se implementará el sistema para que tenga un buen desempeño, robustez y escalabilidad.
- **Implementación:** Se limita a implantar y probar cada uno de los componentes de la aplicación de forma individual.
- **Pruebas:** El propósito es evaluar la calidad del producto. Esto no solo implica el producto final, sino también la arquitectura del sistema. Además el producto también es evaluado por los usuarios finales.
- **Despliegue:** El propósito es entregar el producto terminado al usuario final. Esto involucra una serie de actividades como son probar el software en su ambiente de operación real (pruebas beta), empaquetar el software para su entrega, instalarlo, entrenar a los usuarios y/o migrar software existente o convertir bases de datos.
- **Configuración y Gestión de Cambios:** Su propósito es realizar un seguimiento y mantener la integridad de la evolución de los activos del proyecto. Durante el transcurso del ciclo de desarrollo, muchos artefactos software que son valiosos se crean. El desarrollo de estos artefactos requiere mucho esfuerzo y representan una inversión significativa, por esta razón se convierten en elementos importantes que deberán ser protegidos y estar disponibles para su reutilización. Estos artefactos evolucionan, sobre todo en un desarrollo iterativo, por lo cual se actualizan una y otra vez. Por eso se deben identificar y localizar los artefactos software, para seleccionar la versión adecuada de cada uno de ellos, y comprender su estado actual y las razones por las que ha cambiado.
- **Administración del proyecto.** Consiste en equilibrar los objetivos competitivos, gestionar los riesgos, y eliminar los obstáculos con el fin de poder ofrecer un

producto que satisfaga las necesidades de los clientes (los que pagan por el producto) y los usuarios finales.

- Entorno. Consiste en brindar soporte al desarrollo de la organización a través de herramientas y procesos. Este soporte incluye selección y adquisición de herramientas, instalación y configuración de las mismas con el fin de adaptarlas a la organización, además de ofrecer servicio técnico en aspectos tales como tecnologías de información (TICs), administración de cuentas de usuario y copias de seguridad.

### **Lenguaje Unificado de Modelado (UML).**

El Lenguaje Unificado de Modelado (UML) [2] es el lenguaje de modelado estándar para el desarrollo de software y sistemas. Es un lenguaje de modelado visual que permite a los desarrolladores de sistemas poder crear modelos que capturan sus visiones de un proyecto en un estándar fácil de entender, y que proporciona un mecanismo eficaz para compartir y comunicar estas visiones con otros.

Antes de existir UML, los analistas de sistemas intentaban evaluar las necesidades de sus clientes, generando un análisis de requisitos en alguna notación que el analista lograra entender (pero no siempre el cliente), enviando este análisis a un programador o equipo de programadores, con la esperanza de que el producto final fuera el sistema que el cliente deseaba.

La clave es organizar el proceso de diseño de manera que los analistas, clientes, programadores, y otros involucrados en el desarrollo del sistema puedan estar de acuerdo. UML provee esta organización.

UML fue creado por Grady Booch, James Rumbaugh, e Ivar Jacobson. Estas personas trabajaron en organizaciones diferentes a través de los años 80 y principios de los 90, cada uno de ellos elaborando su propia metodología para el análisis y diseño orientado a objetos. Sus metodologías lograron imponerse sobre las de sus competidores. A mediados de los años 90 comenzaron a utilizar las ideas del otro, así que decidieron desarrollar su trabajo juntos.

Los borradores de UML comenzaron a circular en toda la industria del software, y la retroalimentación resultante ha traído consigo cambios sustanciales. Debido a que muchas empresas descubrieron que UML podría servir a sus fines estratégicos, se creó el consorcio UML, el cual cuenta entre sus miembros con organizaciones tales como DEC, Hewlett-Packard, Intellicorp, Microsoft, Oracle, Texas Instruments, Rational, entre otras. En 1997 el consorcio produjo la versión 1.0 de UML y lo presentó a la OMG (Object Management Group), en respuesta la OMG propuso el crear un lenguaje de modelado estándar.

El consorcio amplió UML y generó la versión 1.1 presentándola a la OMG quien la

adoptó a finales de 1997. La OMG se hizo cargo del mantenimiento de UML y produjo dos revisiones más en 1998. UML se ha convertido en un estándar de facto en la industria del software, y este sigue evolucionando. Se han desarrollado las versiones 1.3, 1.4 y 1.5. Actualmente se encuentra en la Version 2 para la cual la OMG también ha dado su aprobación.

### **Componentes de UML.**

UML consiste en una serie de elementos gráficos que se combinan para formar diagramas. El propósito de los diagramas es presentar múltiples vistas de un sistema. Este conjunto de múltiples vistas es llamado modelo. Es importante señalar que un modelo UML describe lo que se supone que debe hacer un sistema, más no cómo implementar el sistema [2].

- Diagrama de Clases: Muestra las clases (una clase es una categoría o grupo de cosas que tienen los mismos atributos y los mismos comportamientos.) que hacen parte del sistema y cómo se relacionan entre sí.
- Diagrama de Objetos: Muestra los objetos y sus relaciones. Un objeto es una instancia de una clase.
- Diagrama de casos de uso: Un caso de uso es una descripción del comportamiento de un sistema desde el punto de vista de un usuario.
- Diagrama de Estados: Muestra el comportamiento del sistema con base a eventos.
- Diagrama de Secuencia: Muestra la interacción entre los objetos del sistema.
- Diagrama de Actividades: Muestra el comportamiento del sistema a través de un flujo de actividades.
- Diagrama de Comunicación: Muestra como los elementos de un sistema trabajan en conjunto para lograr los objetivos del sistema.
- Diagrama de Componentes: Muestra como están organizados los componentes y sus dependencias.
- Diagrama de Despliegue: Muestra la arquitectura física del sistema. En este diagrama se muestran los equipos y sus conexiones, así como el software que se encuentra en cada uno de ellos.

## 1. CAPITULO 1

### 1.1 CONCEPTOS BÁSICOS

Antes de hablar acerca del diseño de una aplicación para realizar difusión de video ya sea por demanda o en vivo se debe tener una concepción clara acerca de lo que es streaming, así como los protocolos y demás elementos que forman parte importante para su implementación.

Streaming de multimedia en esencia es la reproducción en tiempo real o por demanda de contenido multimedia vía Internet o Intranet<sup>1</sup>.

También se puede definir Streaming como la tecnología multimedia que permite la difusión de grandes archivos de audio y video a través de Internet, pudiendo escuchar o visualizar contenido en tiempo real o previamente editado. El usuario podrá ver y/o escuchar contenido en tiempo real o previamente almacenado sin la necesidad de descargarlo a su computador<sup>2</sup>.

El streaming tiene diversos usos, que van desde el entretenimiento hasta aplicaciones de negocios. Puede ser usado para emitir un evento en vivo desde un estadio (béisbol, fútbol, etc.), un teatro, un concierto, una sala de juntas, o reuniones de ventas; proporcionar seguridad y vigilancia a una casa o un negocio.

Entre las variadas aplicaciones del streaming se tiene<sup>3</sup>:

- **Entretenimiento.** Para transmitir eventos deportivos, conciertos, entre otros.
- **Publicidad.** Para el posicionamiento de una marca o dar a conocer un producto, el cual puede ser visto por todo el mundo a través de la Web.
- **Videoconferencia.** Para realizar videollamadas en la cual los usuarios pueden hablar unos con otros, lo que la convierte en una poderosa herramienta para los negocios y la educación.
- **Telemedicina.** Permitiendo a los médicos realizar su trabajo en un lugar remoto. Entre los ejemplos de telemedicina, se tiene la teleradiología que permite enviar imágenes de rayos X, a un radiólogo en una ubicación distante para su interpretación; también se tiene la telepatología en la cual un patólogo examina muestras de tejidos de un paciente que se encuentra ubicado en otro lugar, y la telepsicología en donde a través de una videollamada se puede

---

<sup>1</sup> Prieto Serpa, Felipe Andrés; Rodríguez Rodríguez, Luisa Fernanda. **Streaming de Audio a Través de Dispositivos Móviles Utilizando J2ME**. Tesis de Pregrado Ingeniería de Sistemas. Bogotá D.C : Universidad EL BOSQUE, Noviembre 2008.

<sup>2</sup> Grupo Mnenmix. Soluciones de Streaming. [www.mnenmixip.com/html/descargas/mirame.pdf](http://www.mnenmixip.com/html/descargas/mirame.pdf). Septiembre de 2008. [Consulta: 1 de Septiembre 2010].

<sup>3</sup> Simpson, Wes. **Video Over IP: A Practical Guide to Technology and Applications**. Burlington: Focal Press, 2008.

hacer la evaluación de pacientes, terapias, entre otros.

- **Educación a Distancia.** Para brindar instrucción a estudiantes que están físicamente separados de sus profesores. Esta instrucción puede ser en vivo o pregrabada.
- **Telejusticia.** Proporciona servicios a los tribunales y abogados que se encuentran lejos de los clientes. Se utiliza normalmente para asuntos de rutina en la corte, tales como audiencias de fianza y lectura de cargos. La tele justicia también puede usarse en juicios reales, en los que los acusados o los testigos requieren precauciones especiales de seguridad.
- **Presentaciones de Ventas.** Usado para brindar información acerca de un amplio rango de productos y servicios a clientes potenciales.
- **Seguridad.** Proporciona vigilancia a un largo número de espacios públicos, incluyendo aeropuertos, calles y centros comerciales. También es usado para monitorear propiedad privada, tales como bancos, casinos y plantas industriales.

Los sistemas de streaming pueden difundir contenido de video de dos maneras diferentes. La primera es el streaming en vivo, en la cual el contenido de vídeo aparece en la pantalla de cada persona al mismo tiempo. La segunda es el streaming por demanda, el cual sucede cuando los usuarios pueden seleccionar el contenido de video que ellos desean en cualquier momento. En este caso, el contenido es enviado a cada usuario, pero cada uno de ellos puede elegir cuando empieza y cuando termina el video (y posiblemente adelantarlo o retrocederlo). El video por demanda es uno de los más usados y preferido por los usuarios debido a que ellos pueden tener el control de cuando desean ver el contenido<sup>4</sup>.

**1.1.1 Arquitectura de un Sistema de Streaming de Video.** Los elementos básicos de un sistema de streaming incluyen elementos de hardware y software cada uno con una función específica con el fin de permitir que el flujo de video pueda ser emitido a través de una red.

**1.1.1.1 Elementos Hardware.** La mayoría del hardware necesario para realizar streaming de video a través de una red pueden encontrarse dentro de una misma máquina, aunque en algunas ocasiones se requiere que esté localizado en diversas ubicaciones geográficas para mejorar el rendimiento del sistema y eliminar algunos cuellos de botella que pueden producirse en la red<sup>5</sup>.

Entre los elementos de hardware se tienen<sup>5</sup>:

- **Sistema de Preparación de Contenido.** Encargado de procesar el contenido

---

<sup>4</sup> Simpson, Wes. **Video Over IP: A Practical Guide to Technology and Applications.** Burlington: Focal Press, 2008.

<sup>5</sup> Simpson, Wes; Greenfield, Howard. **IPTV and Internet Video: New Markets in Televisión Broadcasting.** Burlington: Focal Press, 2007.

de video para que esté listo para la transmisión. El proceso puede incluir conversión del formato, compresión de vídeo, etiquetado e indexación y publicación para la emisión.

- **Servidor Web.** Utilizados en los sistemas de streaming de vídeo para ayudar a los reproductores o visores en la selección de los contenidos que serán visualizados a través de una página Web, dentro de la cual se encontrarán los enlaces al contenido de vídeo que iniciarán su reproducción, tan pronto como un usuario lo visite.
- **Servidor de Vídeo.** El servidor de vídeo es la parte fundamental de un sistema de streaming de vídeo. Está encargado de almacenar de forma segura los archivos de vídeo que pueden ser vistos y además debe crear los flujos de paquetes que se entregan a cada usuario.
- **Cliente PC.** Los sistemas de streaming entregan el contenido de video a diversos dispositivos tales como computadores o móviles para que sean visualizados. Cada dispositivo debe estar equipado con software capaz de recibir los paquetes de video entrantes y convertirlos en una imagen de video para que el usuario pueda observarlo.

**1.1.1.2 Elementos Software.** Al igual que el hardware el software es un elemento clave en un sistema de streaming de video, puesto que permite encontrar y seleccionar el contenido, así como entregarlo al dispositivo del usuario.

Entre los elementos software usados en un sistema de streaming se tienen<sup>5</sup>:

- **Portal Web.** Los portales generalmente consisten en una serie de páginas Web que sirven al usuario de interfaz para acceder al contenido de video.
- **Streaming Engine.** Encargado de crear una serie de paquetes de vídeo IP para cada flujo de salida.
- **Navegador Web.** Permite a los usuarios ver y navegar a través de las páginas las cuales contienen texto, gráficos, entre otros. Los navegadores usan plug-ins multimedia para soportar otras funciones tales como la reproducción de audio y video.
- **Reproductor Multimedia.** Reside en el dispositivo (computador, teléfono móvil, etc.) del usuario. Su función es muy similar a la de un plug-in multimedia de un navegador Web, en donde toma un flujo de video/audio y lo convierte en imágenes y sonidos que pueden ser vistos y escuchados por los usuarios.

**1.1.2 Codecs de Compresión de video.** La compresión reduce el número de bits usados para representar cada píxel de una imagen y así lograr transmitirla satisfactoriamente a través de una red. Algunos de estos codecs pueden ser de carácter abierto y otros propietarios.

- **MPEG-1.** Fue desarrollado en la década de los noventa y fue el primer estándar creado para compresión de video. Se diseñó para ser utilizado en la

creación de discos compactos de vídeo. Este codec aún se usa ocasionalmente en algunas aplicaciones como son las cámaras de vigilancia de bajo costo y en vídeo Web, aunque estas son cada vez más escasas<sup>6</sup>.

- **MPEG-2.** Es el estándar más usado en la actualidad. Fue aprobado en 1996 y es usado por un amplio rango de aplicaciones, incluyendo la producción y difusión de televisión digital, televisión de alta definición, video en formato DVD, televisión por satélite y por cable<sup>6</sup>.
- **MPEG-4.** Es un estándar reciente el cual fue aprobado en el año 2003. Ha tenido un impacto significativo en lo que respecta a la compresión de video en general, y en particular el transporte de video IP, ya que ha permitido una reducción del 50 por ciento en el ancho de banda en comparación con MPEG-2 y ha permitido nuevas aplicaciones como la entrega de vídeo de alta definición a través de Internet y por medio de discos Blu-ray y redes IPTV<sup>6</sup>.
- **H.264 Advanced Video Codec.** Publicado en 2003, ofrece una mayor eficiencia y compresión, lo que se traduce en una mayor calidad en la compresión de video, y mayor flexibilidad en el transporte y almacenamiento del mismo<sup>7</sup>.
- **Apple QuickTime.** Presentado en 1991, es usado para crear contenido de video digital y JPEG en movimiento<sup>8</sup>.
- **RealNetworks.** Publicado en 1997, posee una compresión, eficiencia y soporte mejorados para video de alta definición<sup>8</sup>.
- **Windows Media Video.** Desarrollado por Microsoft, es similar en principio al estándar MPEG-4, pero ha sido optimizado para brindar una mejor eficiencia en la compresión. Es más adecuado para aplicaciones regulares de streaming con anchos de banda inferiores 1 Mbit/s<sup>8</sup>.

**1.1.3 Protocolos de Streaming.** Los protocolos de Streaming son utilizados para la transmisión de flujos video en tiempo real a través de una red.

**1.1.1.3 HyperText Transfer Protocol (HTTP).** La mayoría del contenido que se encuentra presente en el interior de un sitio Web, se sirve a través de HTTP. Cualquier servidor Web, como por ejemplo Apache o Microsoft Internet Information Services (IIS), pueden entregar archivos de video. Las principales razones para usar un servidor Web con HTTP para alojar contenido de video son la simplicidad y el costo.

Sin embargo, el contenido de vídeo emitido a través de HTTP carece de protección sobre todo para archivos con propiedad intelectual, ya que la caché del navegador Web guarda los archivos de vídeo en su almacenamiento temporal, el cual puede ser visitado por el usuario después de navegar en el sitio.

---

<sup>6</sup> Simpson, Wes. **Video Over IP: A Practical Guide to Technology and Applications.** Burlington: Focal Press, 2008.

<sup>7</sup> Richardson, Iain E. **The H.264 Advanced Video Compression Standard.** United Kingdom: Wiley, 2003.

<sup>8</sup> Austerberry, David. **The Technology of Video and Audio Streaming.** Burlington: Focal Press, 2005.

Otro inconveniente de la entrega de vídeo a través de HTTP es la falta de acceso a todas las partes del archivo de vídeo, puesto que este es descargado progresivamente a la caché del navegador Web y el usuario puede acceder únicamente a la porción de video que ha sido descargada. Por ejemplo, si se tiene un video de 30 minutos y el usuario quiere ver sólo los últimos 5 minutos del video, debe esperar hasta que los primeros 25 minutos se hayan descargado en la caché del navegador<sup>9</sup>.

**1.1.4.1 The Real Time Transport Protocol (RTP).** El protocolo RTP<sup>10</sup> define un modelo para la transmisión de medios en tiempo real. Fue desarrollado por el Audio-Video Transport Working Group de la IETF (Internet Engineering Task Force) y publicado en 1996 como RFC 1889. El modelo describe métodos para empaquetar datos con características de tiempo real, tales como video y audio interactivo. Usualmente corre en UDP.

El protocolo RTP incluye campos de datos adicionales que no están presentes en TCP. Posee un campo llamado timestamp y sequence number (número de secuencia) para facilitar la sincronización en el transporte de los datos, y brindan una forma de control al servidor de medios (media server) para que la secuencia de video sea mostrada en tiempo real. El reproductor de video, utiliza estos campos para ensamblar los paquetes recibidos en el orden y velocidad de reproducción correcta<sup>11</sup>.

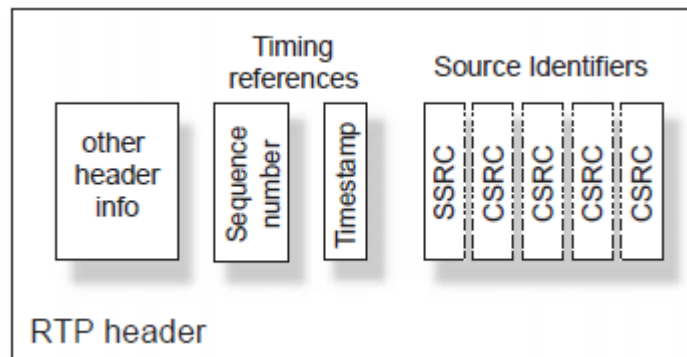


Figura 2. Encabezado RTP<sup>12</sup>.

- **Sequence Number.** El valor de este número de 16 bits se incrementa en uno por cada paquete. Es usado por el reproductor para detectar paquetes

<sup>9</sup> Reinhardt, Robert. **Video with Adobe® Flash® CS4 Professional Studio Techniques**. Berkeley: Adobe Press, 2009.

<sup>10</sup> <http://www.faqs.org/rfcs/rfc1889.html>. [Consulta: 10 de Septiembre 2010].

<sup>11</sup> Austerberry, David. **The Technology of Video and Audio Streaming**. Burlington: Focal Press, 2005.

<sup>12</sup> Simpson, Wes; Greenfield, Howard. **IPTV and Internet Video: New Markets in Televisión Broadcasting**. Burlington: Focal Press, 2007.

perdidos y luego poner la secuencia de los paquetes en el orden correcto. El número inicial para una sesión de streaming es seleccionado al azar.

- **Timestamp.** Es un campo de 32 bits asignado en el momento del envío tomando como referencia un reloj del sistema. Esto permite la sincronización y cálculos de “jitter<sup>13</sup>”. El valor inicial es seleccionado de forma aleatoria para evitar confusión con otras secuencias RTP presentes.
- **Source Identifiers.** CSRC es un identificador único para la sincronización de un flujo de datos RTP. Uno o más CSRCs existen cuando el flujo RTP está llevando información de múltiples medios y fuentes. Este podría ser el caso de una mezcla entre dos fuentes de vídeo o de contenido embebido.

El protocolo RTP posee algunas ventajas y desventajas las cuales deben tenerse en cuenta en el momento de utilizarlo para la transmisión de contenido de video a través de una red de datos.

Entre las ventajas que tiene este protocolo se tiene<sup>14</sup>:

- *Múltiples formatos.* Soporta diferentes tipos de flujos de audio y video, permitiendo a las aplicaciones fácilmente compartir video en formato MPEG entre otros.
- *Números de secuencia de paquetes.* Estos permiten a las aplicaciones que reciben los datos determinar si los paquetes están perdidos o han llegado fuera de secuencia, así podrá reordenarlos y corregir los errores que se hayan producido.
- *Soporta Multicasting (Multidifusión).* El streaming a través de RTP puede ser distribuido mediante Multicasting (Multidifusión), de esta manera puede ser visto por varios (o cientos) de usuarios al mismo tiempo.
- *Datos de Sincronización Incluidos.* La sincronización entre múltiples tipos de stream permite a las señales multimedia (video, audio, etc) ser transmitidas separadamente y realineadas a un tiempo base común en el receptor.

Entre las desventajas que posee están<sup>14</sup>:

- *Bloqueo de Firewall.* Algunos firewall pueden bloquear el tráfico RTP, por que los paquetes UDP son rutinariamente bloqueados en muchas instalaciones

---

<sup>13</sup> [http://www.voipforo.com/QoS/QoS\\_Jitter.php](http://www.voipforo.com/QoS/QoS_Jitter.php). [Consulta: 13 de Septiembre 2010].

<sup>14</sup> Simpson, Wes. **Video Over IP: A Practical Guide to Technology and Applications**. Burlington: Focal Press, 2008.

- *No hay Prioridad.* No existe un mecanismo dentro de RTP que asegure el transporte oportuno de los paquetes (como una cola de prioridad).

**1.1.4.2 Real Time Streaming Protocol (RTSP).** El protocolo RTSP<sup>15</sup> fue desarrollado por el IETF (Internet Engineering Task Force) y publicado en 1998 como RFC 2326. RTSP es un protocolo el cual permite a un cliente controlar remotamente un servidor de streaming. El cliente controla el servidor mediante la emisión de órdenes a distancia con comandos, tales como "play" y "pause".

La función principal de RTSP es establecer y controlar el tiempo y la sincronización de uno o más flujos de streaming. RTSP es un protocolo que se encuentra en la capa de aplicación del modelo OSI. No transmite el flujo de stream por si mismo, pero ayuda a los protocolos de las capas inferiores como RTP a establecer conexiones entre las partes involucradas en la sesión<sup>16</sup>.

RTSP fue desarrollado intencionalmente para ser similar en sintaxis y operación a HTTP versión 1.1; aunque difieren en varios aspectos importantes. Con RTSP tanto el cliente como el servidor pueden emitir solicitudes durante la interacción, mientras que en HTTP el cliente siempre realiza las solicitudes (de documentos). RTSP tiene que mantener el estado de una sesión, mientras que HTTP no. RTSP soporta el uso de RTP como el protocolo de entrega de datos fundamental. El protocolo tiene como objetivo brindar una forma de elegir el canal de distribución óptimo a un cliente. Algunos firewalls no dejarán pasar los paquetes UDP. El servidor de streaming debe ofrecer una opción de protocolos de entrega, como UDP, multicast UDP, y TCP para satisfacer a los clientes<sup>28</sup>.

**1.1.4.3 Real Time Messaging Protocol (RTMP).** RTMP es un protocolo desarrollado por Adobe Systems para realizar streaming de audio, video y datos a través de Internet, entre un reproductor Flash y un servidor. Adobe ha publicado la especificación del protocolo para uso público<sup>17</sup>.

Uno de los beneficios de la entrega de video a través de RTMP para el usuario es la reproducción casi instantánea del vídeo, cuando el archivo es codificado con una tasa de bits adecuada a la velocidad de conexión del usuario. Además también se puede buscar cualquier punto en el contenido que se emite. Esta característica es particularmente ventajosa para contenido de larga duración porque el usuario no tiene que esperar a que el archivo de vídeo se cargue antes de saltar hacia adelante, como es el caso del vídeo emitido en HTTP.

---

<sup>15</sup> <http://www.faqs.org/rfcs/rfc2326.html>. [Consulta: 13 de Septiembre 2010].

<sup>16</sup> Psaiar, Harald. **A Java-Based Streaming Media Server**. Tesis. Viena : Institut für Interaktive Medien Systeme der Technischen Universität Wien. 2005.

<sup>17</sup> [http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp\\_specification\\_1.0.pdf](http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf). [Consulta: 14 de Septiembre 2010].

Además para realizar streaming de video mediante el protocolo RTMP se puede usar cualquier formato de video, no solo el basado en Flash (FLV), sino otros tales como Apple QuickTime o Windows Media de Microsoft. Cuando el flujo de streaming de video se realiza mediante el protocolo RTMP y se envía a un reproductor basado en Flash, el audio y el video son almacenados únicamente en un buffer de memoria de dicho reproductor. El archivo de vídeo completo nunca se copia o se almacena en la caché del navegador Web<sup>18</sup>.

**1.1.4.4 Real Time Media Flow Protocol (RTMFP).** El Real Time Media Flow Protocol<sup>19</sup> es un nuevo protocolo de comunicaciones desarrollado por Adobe que permite la comunicación directa entre múltiples reproductores de Flash y aplicaciones usando el framework Adobe AIR.

RTMFP ayuda a reducir los costes de ancho de banda en transmisiones de soluciones de comunicación en tiempo real, tales como audio y video Chat, debido a que RTMFP puede enviar datos directamente entre usuarios sin tener que pasar por el servidor, así el ancho de banda no se está utilizando en el servidor de modo que las soluciones son menos costosas.

RTMFP también aumenta la calidad de la entrega del contenido a través del uso del protocolo UDP. A través de UDP se puede de una manera más eficiente enviar vídeo y datos de audio a través de Internet, lo que asegura que las conexiones no se verán interrumpidas, si se producen variaciones al interior de la red.

RTMFP tiene dos características que mejoran la fiabilidad de las conexiones sobre otros protocolos, como TCP.

- **Rápida Restauración de la Conexión.** Las conexiones son reestablecidas rápidamente después de interrupciones breves. Por ejemplo, cuando una conexión de red inalámbrica experimenta una caída. Después de la reconexión la conexión se encuentra al cien por ciento de sus capacidades de forma instantánea.
- **Movilidad Ip.** Las sesiones de red activas se mantienen incluso si un cliente cambia a una nueva dirección IP. Por ejemplo cuando un computador portátil se encuentra conectado a una red inalámbrica, la conexión no será interrumpida. Este es un elemento crítico para transmisiones en vivo.

---

<sup>18</sup> Reinhardt, Robert. **Video with Adobe® Flash® CS4 Professional Studio Techniques**. Berkeley: Adobe Press, 2009.

<sup>19</sup> [http://download.macromedia.com/pub/labs/flashplayer10/flashplayer10\\_rtmfp\\_faq\\_081010.pdf](http://download.macromedia.com/pub/labs/flashplayer10/flashplayer10_rtmfp_faq_081010.pdf). [Consulta: 18 de Septiembre 2010].

## 1.2 DESCRIPCIÓN DE PLATAFORMAS PARA EL DESARROLLO DE APLICACIONES DE STREAMING DE VIDEO.

Existen diversas plataformas para realizar streaming de video a través de una red de datos algunas son de carácter propietario y otras de código abierto, las cuales ofrecen una gran cantidad de características y ventajas que pueden ser utilizadas en la mayoría de sistemas operativos existentes.

**1.2.1 Plataforma Adobe Flash Media Server.** El servidor Flash Media Server, es un servidor multimedia en tiempo real que ofrece una calidad alta (hasta el nivel de HD, High Definition), para video por demanda, video y audio en vivo, con gran eficiencia y calidad de servicio sin importar el sistema operativo. Se puede emitir vídeo pregrabado, vídeo en directo, listas de reproducción de música, video blogging, vídeo mensajería, entornos para Chat multimedia, difusión de datos en tiempo real, juegos multiusuario, etc. Flash Media Server se comunica y emite flujos de audio, video datos hacia Flash Player, Adobe AIR, Flash Lite 3, y el nuevo Adobe Media Player a través de plataformas y navegadores<sup>20</sup>.

Adobe Flash Media Server posee tres versiones

- **Adobe Flash Media Development Server.** Es una edición gratuita que se encuentra disponible en el sitio Web de Adobe, el cual puede ser utilizado en producción por cualquier usuario para implementar servicios básicos de streaming o soluciones de comunicación. Puede también ser usado para desarrollar aplicaciones de streaming avanzadas, puesto que no existe límite en la funcionalidad. También se puede utilizar la característica de publicación multipunto, la cual permite crear un punto de publicación en vivo en la red e insertar mensajes de datos en el flujo de video. Este servidor gratuito tiene una capacidad límite de 10 conexiones de entrada. El número de conexiones de salida está limitado únicamente por el ancho de banda y la capacidad del procesador<sup>14</sup>. Esta versión fue utilizada para desarrollar la aplicación de prototipo para el presente proyecto.
- **Adobe Flash Media Streaming Server.** Es un servidor que permite realizar streaming de video de alta calidad (HD) y proporciona más seguridad encriptando el flujo emitido.
- **Adobe Flash Media Interactive Server.** Es un servidor que posee características mucho más avanzadas como son la grabación de video del lado del servidor, funcionalidades de DVR (Digital Video Recorder, Grabador de Video Digital), comunicación multipunto y núcleo de procesamiento distribuido, publicación multipunto e inserción de metadatos en streaming en

---

<sup>20</sup> [www.adobe.com/.../flashmediaserver/pdfs/fms3\\_5\\_wp\\_ue.pdf](http://www.adobe.com/.../flashmediaserver/pdfs/fms3_5_wp_ue.pdf). [Consulta: 25 de Septiembre 2010].

vivo. Además soporta el desarrollo de aplicaciones multipunto, incluyendo aplicaciones de video Chat a través de WebCam, grabación, voz sobre IP (VoIP)<sup>21</sup>.

La plataforma Adobe Flash Media Server es una excelente opción para la entrega de contenido, y es muy utilizado debido a su seguridad y eficiencia.

Esta plataforma ofrece muchos beneficios, entre los que se tienen:

- Reproducción de video más rápida y soporte de video en vivo.
- Calidad de video mejorada, debido a que soporta los codecs On2, VP6, y H.264/AAC para realizar streaming.
- Detección automática del ancho de banda y búfer dinámico.
- Soporta clientes en diversas plataformas (Windows, Linux y Mac).
- La distribución del servidor se encuentra disponible para sistemas Windows y Linux.

Seguridad para el contenido debido a las características de cifrado y de control de acceso al no trabajar con la cache del cliente.

**1.2.2 RED 5.** Red5<sup>22</sup> es un servidor de código abierto escrito en Java diseñado específicamente para trabajar con Adobe Flash Player. Es compatible con TCP basado en RTMP y el protocolo AMF, así como XML-RPC. A través de el es posible realizar streaming de audio/video por demanda o en vivo.

Mediante esta plataforma se pueden desarrollar una variedad de aplicaciones, las cuales incluyen aplicaciones de vídeo, de Chat, juegos multiusuario, y otros, todos estos basados en Flash.

Red5 funciona en cualquier sistema que soporte Java 1.5. Por lo tanto, se puede ejecutar Red5 en cualquier sistema operativo, incluyendo Windows, Linux, Macintosh y Sun<sup>23</sup>, permitiendo emitir streaming de video (FLV), streaming de audio (MP3) y publicar video en vivo.

---

<sup>21</sup> [www.adobe.com/.../flashmediaserver/pdfs/fms3\\_5\\_wp\\_ue.pdf](http://www.adobe.com/.../flashmediaserver/pdfs/fms3_5_wp_ue.pdf). [Consulta: 25 de Septiembre 2010].

<sup>22</sup> Larson, Lisa; Costantini, Renée. **Flash Video for Professionals: Expert Techniques for Integrating Video on the Web**. Indianapolis, Indiana, 2007.

<sup>23</sup> Allen, Chris; Arnold Wade; Balkan Aral, Cannasse, Nicolas; Grden, John; Gunesch, Moses; Hughes, Marc; Macdonald, R.Jon; Zupko, Andy. **The Essential Guide to Open Source Flash Development**. United States of America: friendsof ED, 2008.

**1.2.3 Wowza Media Server Pro.** Wowza<sup>24</sup> es otro servidor escrito en Java, el cual fue creado por ex desarrolladores de Adobe que lo construyeron desde cero. Permite realizar streaming de audio/video y soporta objetos compartidos, y permite scripting del lado del servidor (Java, ActionScript no se permite del lado del servidor).

Entre las características que posee esta plataforma se tienen:

- Multiplataforma (Windows, Linux, Mac OS X, Solaris, y otros).
- Arquitectura de 64 bits.
- Es compatible con Adobe Flash, Apple iOS: iPhone, iPad, iPod touch, Microsoft Silverlight, Apple QuickTime, Android, Blackberry y otras plataformas 3GPP.

**1.2.4 Unreal Media Server.** Es un servidor de streaming para sistemas Windows que ofrece una gran funcionalidad para emitir video en vivo y por demanda. El servidor soporta fuentes de video en vivo y archivos pregrabados.

Unreal Media Server<sup>25</sup> soporta una variedad de protocolos de streaming como por ejemplo RTMP o RTP, permitiendo emitir hacia Flash Player, Silverlight, Windows Media Player y Unreal Media Player en Windows, Mac, Linux y dispositivos móviles.

Unreal Media Server puede correr en computadores con sistema operativo Windows y en dispositivos que ejecuten el sistema Windows Mobile 5 o superior. Los reproductores pueden encontrarse embebidos en una página Web como un control ActiveX para Internet Explorer, o como un plugin para Firefox, Safari, Opera y Chrome.

El contenido no es almacenado en el computador del usuario y este no puede guardarlo localmente, así los derechos de autor están completamente protegidos. También posee restricciones de autenticación y control de acceso a los recursos multimedia.

**1.2.5 Helix Universal Server.** Helix Universal Server<sup>26</sup> es un servidor de streaming multiformato y multiplataforma para brindar contenido de alta calidad para PC y dispositivos móviles.

---

<sup>24</sup> Larson, Lisa; Costantini, Renée. **Flash Video for Professionals: Expert Techniques for Integrating Video on the Web.** Indianapolis, Indiana, 2007.

<sup>25</sup> <http://www.umediaserver.net/umediaserver/overview.html>. [Consulta: 21 de Septiembre 2010].

<sup>26</sup> <http://www.realnetworks.com/products-services/helix-server-proxy.aspx>. [Consulta: 21 de Septiembre 2010].

Soporta diferentes tecnologías y dispositivos:

- Flash, Silverlight, QuickTime.
- Dispositivos Apple, tales como iPhone, iPod touch y iPad.
- Smartphones incluyendo Android and BlackBerry.
- Soporta streaming de contenido en vivo o por demanda y los protocolos RTP, RTSP y RTP.

Helix Server está disponible para Linux, Windows y Solaris.

**1.2.6 Windows Media Server.** Windows Media Server<sup>27</sup> es un software desarrollado por Microsoft que permite hacer streaming digital de audio y video a clientes sobre Internet o una intranet. Estos clientes pueden ser computadores o dispositivos que puedan utilizar un reproductor, como Windows Media Player o también pueden ser computadores que estén corriendo Windows Media Services (llamados Windows media Servers).

El contenido que se emite a los clientes a través de Windows Media Server puede ser en vivo o un archivo digital pregrabado. Además únicamente soporta los formatos Windows Media como son Advanced Systems Format (ASF), Windows Media Audio (WMA), Windows Media Video (WMV), así como JPEG y MP3.

**1.2.7 Darwin Streaming Server.** Darwin Streaming Server<sup>28</sup> es una versión de código abierto del Quicktime Streaming Server de Apple que permite enviar un flujo de video a los clientes a través de Internet usando los protocolos RTP y RTSP. Proporciona un alto nivel de personalización y corre en una variedad de plataformas permitiendo la manipulación del código según las necesidades del usuario. Además soporta el estándar MPEG-4 y 3GPP.

**1.2.8 QuickTime Streaming Server.** Es un servidor de streaming de video comercial que hace parte del MAC OS X Server. Quick Time Streaming Server proporciona a los usuarios una administración mejorada y herramientas de gestión de multimedia como resultado de la estrecha integración con Mac OS X Server; estas herramientas no están disponibles en Darwin Streaming Server<sup>28</sup>.

Al igual que la version OpenSource soporta los estándares MPEG-4 y 3GPP.

---

<sup>27</sup> <http://www.realnetworks.com/helix/products/server/>. [Consulta: 21 de Septiembre 2010].

<sup>28</sup> <http://dss.macosforge.org/>. [Consulta: 21 de Septiembre 2010].

## CAPITULO 2

### 2.1 ANÁLISIS Y DISEÑO

**2.1.1 Entorno de Trabajo.** En esta sección se describirán las herramientas de desarrollo, servidores y otras aplicaciones que fueron utilizadas para desarrollar la aplicación de prototipo para el presente proyecto.

**2.1.1.1 Adobe Flash Media Development Server.** Como se mencionó en el capítulo 1 es un servidor de edición gratuita para desarrolladores que permite implementar servicios básicos de streaming. Este servidor posee una capacidad límite de 10 conexiones de entrada, mientras que el número de conexiones de salida está limitado únicamente por el ancho de banda y la capacidad del procesador<sup>14</sup>.de la maquina en donde se encuentre instalado.

**2.1.1.2 Adobe Flash CS 4.** Es una aplicación que permite la creación y manipulación vectorial, además del manejo de código a través del cual se puede producir y desarrollar contenido interactivo para diversas plataformas. A través de Flash es posible generar animaciones, gráficos o flujos de video; esto ultimo integrando Flash Media Server. Los archivos generados por Flash poseen la extensión SWF, los cuales serán ejecutados a través de una página Web.

### 2.1.3 Tecnologías.

**2.1.3.1 Adobe ActionScript 3.** ActionScript es el lenguaje de la plataforma Adobe Flash. Es un lenguaje orientado a objetos basado en ECMAScript, que es el mismo estándar base del lenguaje JavaScript, el cual originalmente fue desarrollado como una forma para que los desarrolladores tuvieran la capacidad de añadir interactividad a los programas; además permite la programación de varios tipos de aplicaciones en Flash, desde animaciones simples hasta complejas, interfaces interactivas, etc<sup>29</sup>. Fue el lenguaje usado para desarrollar la aplicación de prototipo para lograr emitir video en vivo y por demanda del presente proyecto.

**2.1.3.2 Arquitectura.** La arquitectura que se empleó para la construcción de la aplicación fue cliente-servidor. En este caso se utilizaron dos servidores, un servidor Web para mostrar las páginas en formato HTML las cuales cargan los archivos SWF de las aplicaciones prototipo que fueron desarrolladas y un servidor de video encargado de enviar el flujo de contenido video a las aplicaciones a través del protocolo RTMP.

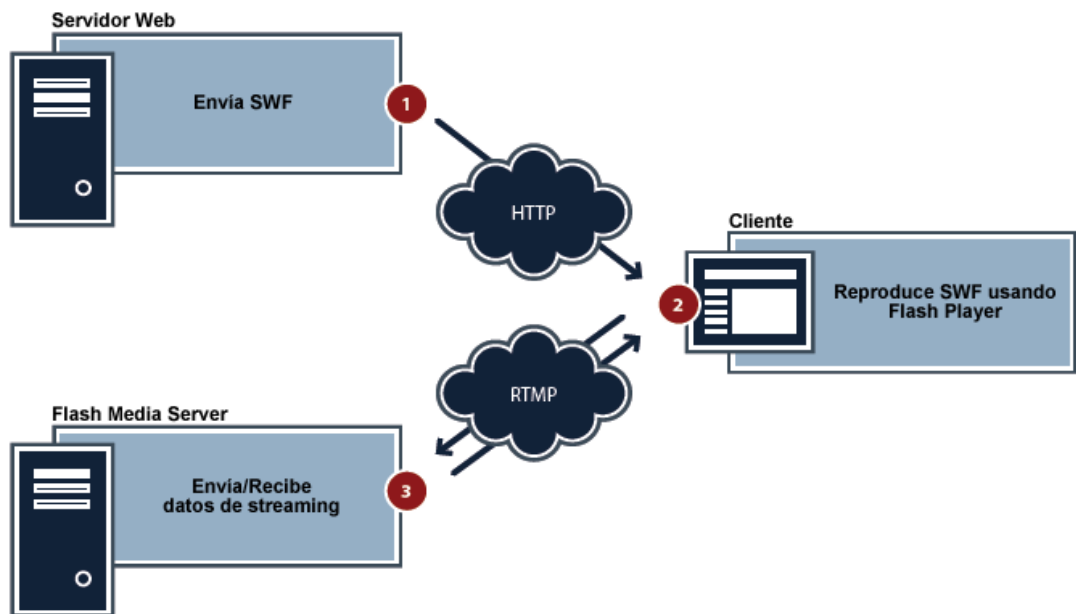
El lado del cliente es desplegado por un archivo SWF, creado en Flash y

---

<sup>29</sup> <http://www.adobe.com/devnet/actionscript.html>. [Consulta: 19 de Marzo 2011].

programado en ActionScript 3.0. El cliente corre en un navegador Web (plugin Flash Player), dispositivos móviles (Flash Lite 3), o como una aplicación de escritorio (Adobe AIR). El servidor gestiona las conexiones de cliente y la seguridad, además lee y escribe en el sistema de archivos, y realiza otras tareas.

El cliente es quien inicia la conexión con el servidor. Una vez conectado, el cliente puede comunicarse con el servidor y con otros clientes conectados. Los clientes se conectan a instancias de las aplicaciones; por ejemplo una aplicación de Chat posee muchas salas. Cada sala es una instancia de la aplicación Chat. Múltiples instancias de una aplicación pueden correr simultáneamente. Cada instancia de una aplicación tiene un nombre único y proporciona recursos a los clientes que se encuentran conectados.



**Figura 3. Arquitectura utilizada en la aplicación [Autor].**

La aplicación consta de 3 módulos cada uno con una función específica. El primero se encarga del video por demanda en donde el usuario elige entre una lista cual desea observar. El segundo maneja todo lo relacionado con la emisión de video en vivo, selección de dispositivos de audio y/o video y calidad de la emisión. El tercero se encarga de visualizar la emisión en vivo generada por el módulo anterior.

## 2.1.4 Especificación

### 2.1.4.1 Modelo de Casos de uso.

- Caso de Uso Ver video por demanda.



Figura 4. Caso de uso ver video por demanda  
Fuente: [Autor].

Cuando se ingresa a la sección que permite observar los videos que se encuentran almacenados en el servidor (video por demanda) el usuario puede seleccionar que video desea reproducir, puede colocarlo en pausa, además de controlar el audio y la reproducción del mismo pudiéndolo adelantar o retroceder en cualquier momento.

### 2.1.4.2 Emitir video en vivo

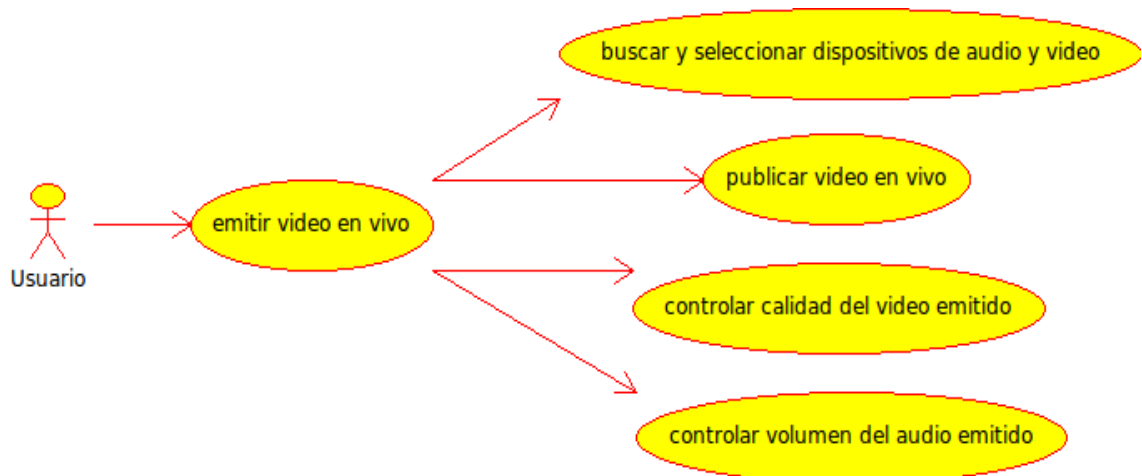
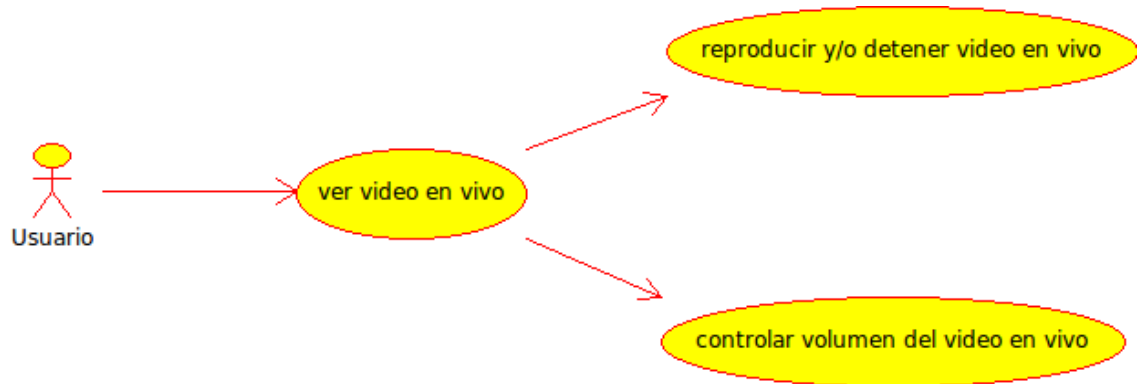


Figura 5. Caso de uso emitir video en vivo  
Fuente: [Autor].

Cuando se ingresa en la sección que permite emitir un flujo de video en vivo el usuario tiene la posibilidad de seleccionar con que dispositivo de audio y video desea realizar la transmisión, además puede controlar la calidad de la imagen como el volumen del audio que se emite.

### 2.1.4.3 Ver video en vivo



**Figura 6. Caso de uso ver video en vivo**  
Fuente: [Autor].

Cuando se ingresa en la sección que permite observar un flujo de video en vivo el usuario tiene la posibilidad de ver y detener lo que está visualizando además de controlar el audio de la emisión que está recibiendo.

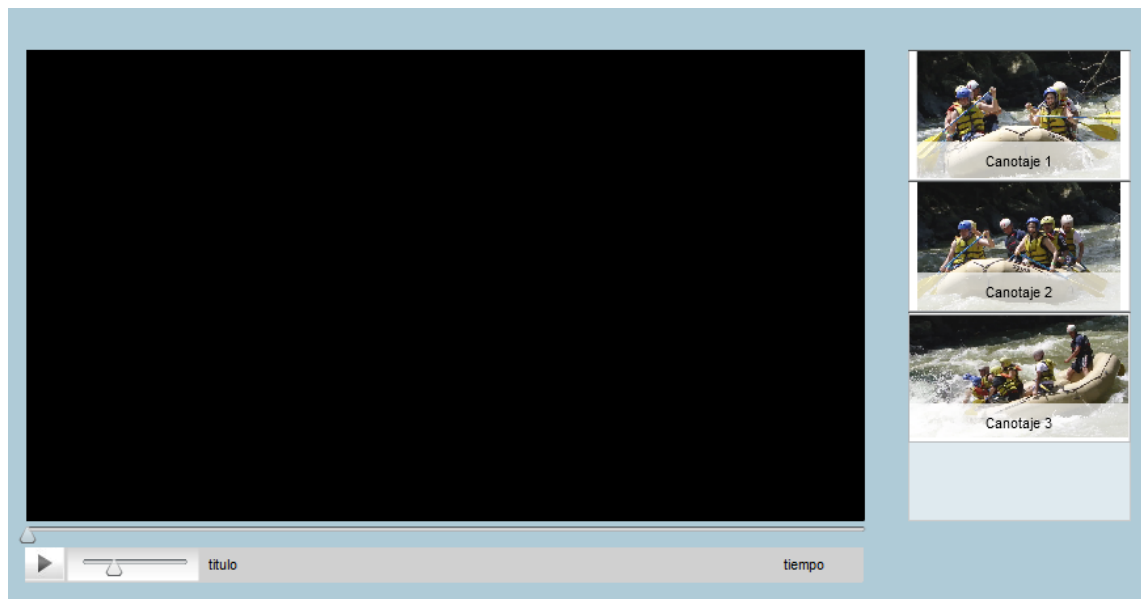
### 3. CAPITULO 3

#### 3.1 DESARROLLO Y PRUEBAS

**3.1.1 Desarrollo de la aplicación.** La aplicación de prototipo fue desarrollada en ActionScript 3, lenguaje de script que hace parte de Adobe Flash CS4. Los módulos desarrollados como se mencionó anteriormente fueron para las tareas de transmitir video por demanda, así como las de emitir y visualizar video en vivo.

Cada módulo se trata de un archivo SWF el cual es insertado mediante lenguaje HTML en una página Web para que los usuarios puedan cargar cada uno de ellos a través de una red de datos.

**3.1.1.1 Módulo Video Por Demanda.** El módulo de video por demanda permite elegir un video y reproducirlo o pausarlo cuantas veces el usuario lo desee. Además es posible controlar el volumen y adelantar o retroceder la reproducción del video.



**Figura 7. Módulo video por demanda**  
Fuente: [Autor].

Este módulo cuenta con un espacio en la parte izquierda en el cual se muestra el video, en la parte derecha se encuentran los videos disponibles.

En la parte baja se encuentra la barra de reproducción del video que permite

adelantar o retroceder el video que está siendo visualizado, además están los controles play y pause, así mismo se muestra el título del video y el tiempo de reproducción.

**3.1.1.2 Módulo Emitir Video en Vivo.** El módulo para emitir video en vivo permite seleccionar un dispositivo de audio y video instalado en el computador del usuario y enviar las imágenes y el sonido capturado por estos hacia el servidor para que puedan ser escuchados y visualizados por el módulo ver video en vivo.

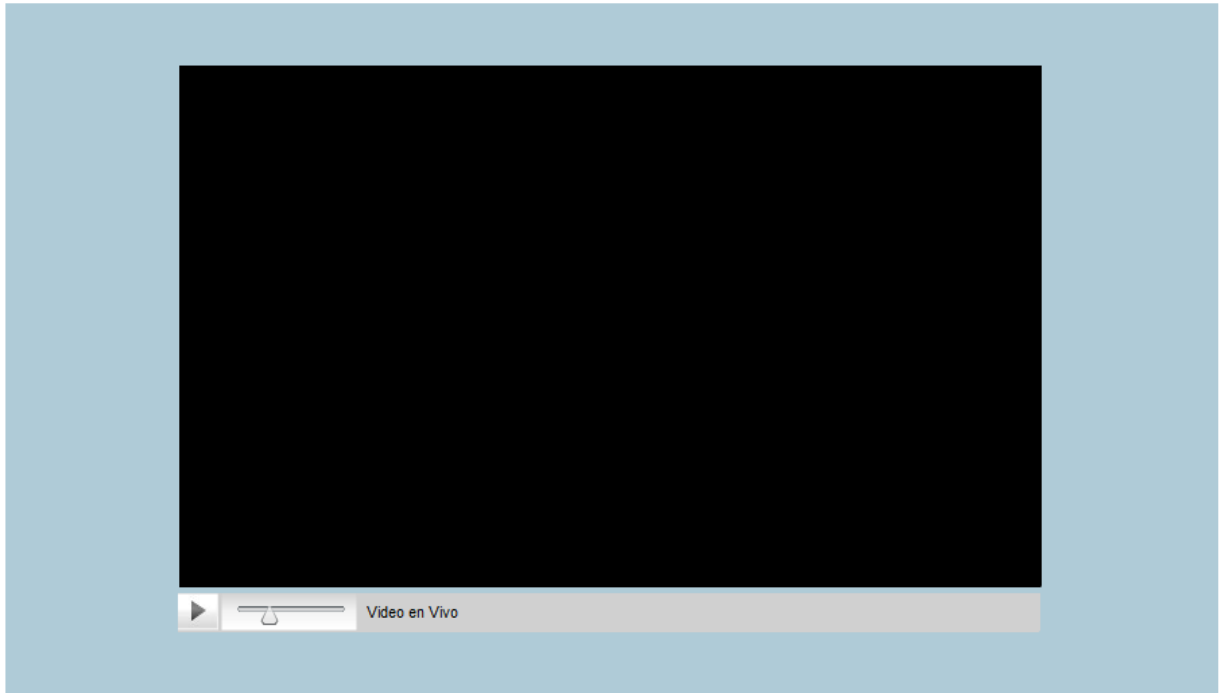
Para emitir video en vivo el usuario pulsa el botón detectar Dispositivos que se encuentra en la parte derecha, haciendo esto el módulo buscará todos los dispositivos de audio y video que se encuentren instalados en la máquina del usuario, estos dispositivos son cargados en dos Combo Box, uno con la lista de cámaras o tarjetas exportadoras de video disponibles y otro con los diferentes micrófonos o salidas de audio.

Luego de haber seleccionado el dispositivo de audio y video el usuario pulsa el botón Emitir el cual procede a enviar el flujo de audio y video hacia el servidor. El espacio que se encuentra en la derecha permitir previsualizar lo que se emite. También el usuario tiene la posibilidad de controlar el volumen de salida del audio emitido así como la calidad del video, además de detener la emisión en cualquier momento pulsando el botón Detener.



Figura 8. Módulo emitir video en vivo  
Fuente: [Autor].

**3.1.1.3 Módulo Ver Video en Vivo.** El módulo para ver video en vivo permite a los demás usuarios visualizar lo que se emite a través del módulo que emite video en vivo. Consta de un espacio en el cual se puede observa la imagen que se está emitiendo, y en la parte de abajo posee controles para detener o regresar a la emisión y un control de volumen.



**Figura 9. Módulo ver video en vivo**  
Fuente: [Autor].

**3.1.2 Pruebas y resultados.** Se realizaron diferentes tipos de pruebas sobre la aplicación desarrollada con el fin de observar el comportamiento del flujo de video y de la red, en aspectos como la ralentización de la transmisión y el ancho de banda del servidor. Estas pruebas se realizaron a través de la Internet y no en una red local, esto con el fin de que las pruebas fueran lo más parecido a un ambiente real. Además los computadores que participaron en estas pruebas se encontraban conectados a través de distintos medios de transmisión, uno de ellos se conectó mediante cable de red y se encontraba en la ciudad de Barrancabermeja y los demás de forma inalámbrica los cuales se encontraban en la ciudad de Bucaramanga.

**3.1.2.1 Pruebas transmisión video en vivo.** La primera prueba se realizó con una transmisión de video en vivo a la máxima calidad permitida por la aplicación con 3 usuarios conectados en forma simultánea, en donde se observó una buena nitidez en la imagen pero a cambio de algunos pequeños saltos, que hacían que la imagen algunas veces se observaran cuadro a cuadro. Además se pudo observar

una variación significativa en el consumo de ancho de banda el cual en algunas ocasiones tendió a incrementarse como se puede observar en las imágenes.

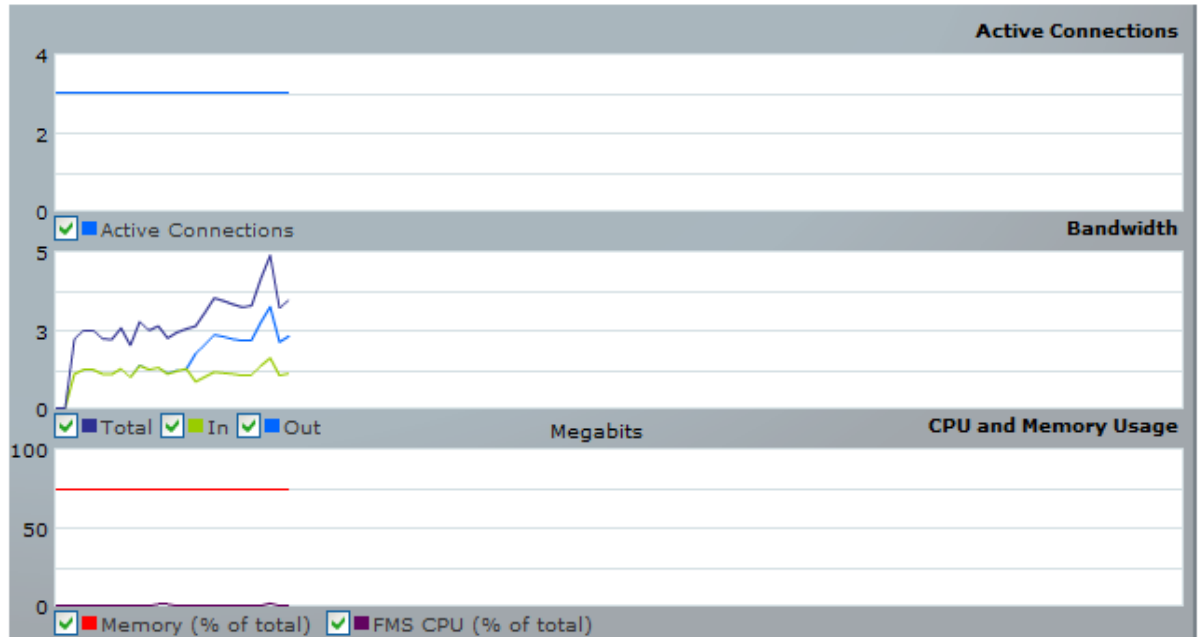


Figura 10. Comportamiento inicial del consumo de ancho de banda emitiendo video en vivo  
Fuente: [Autor].

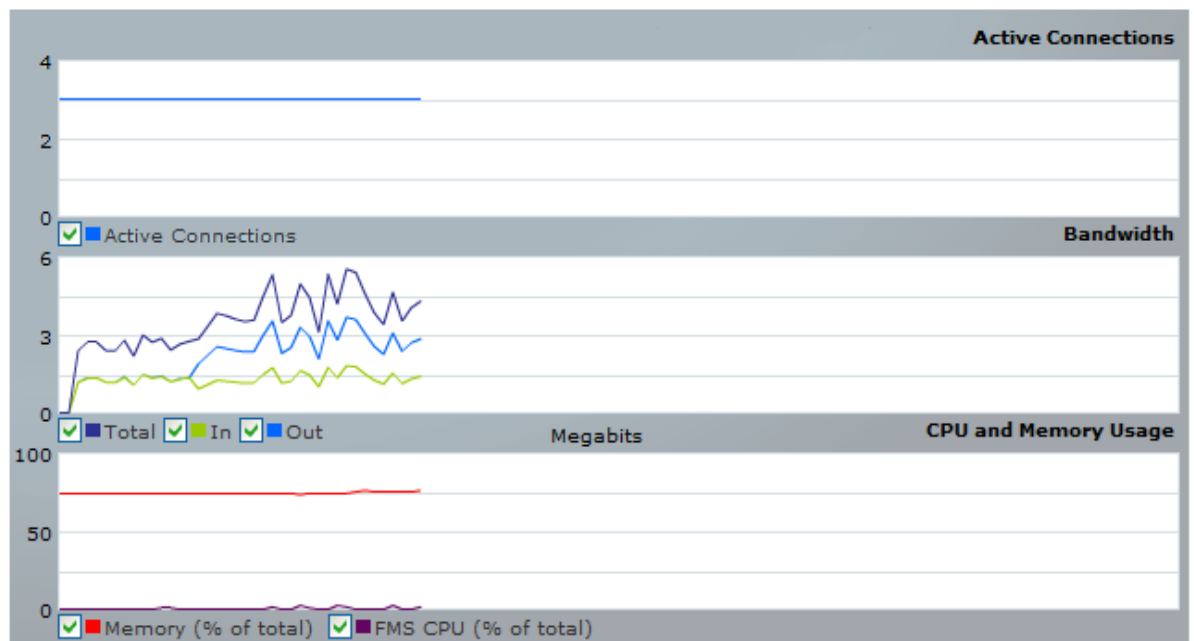
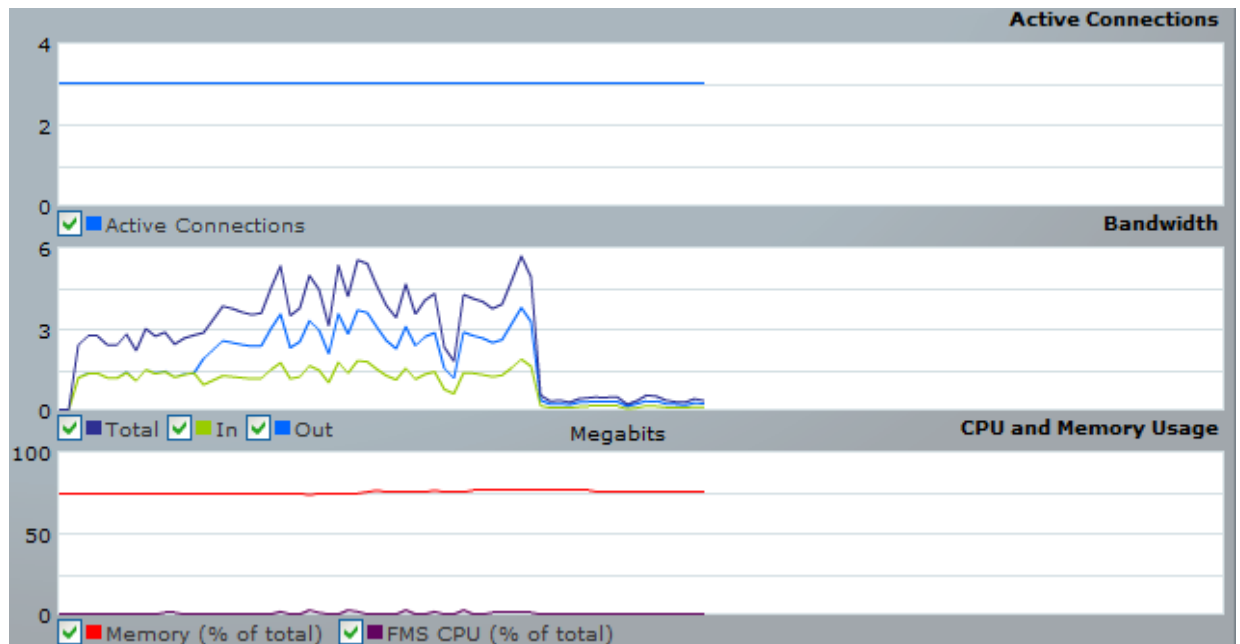


Figura 11. Consumo de ancho de banda emitiendo video en vivo con alta calidad luego de cinco minutos  
Fuente: [Autor].

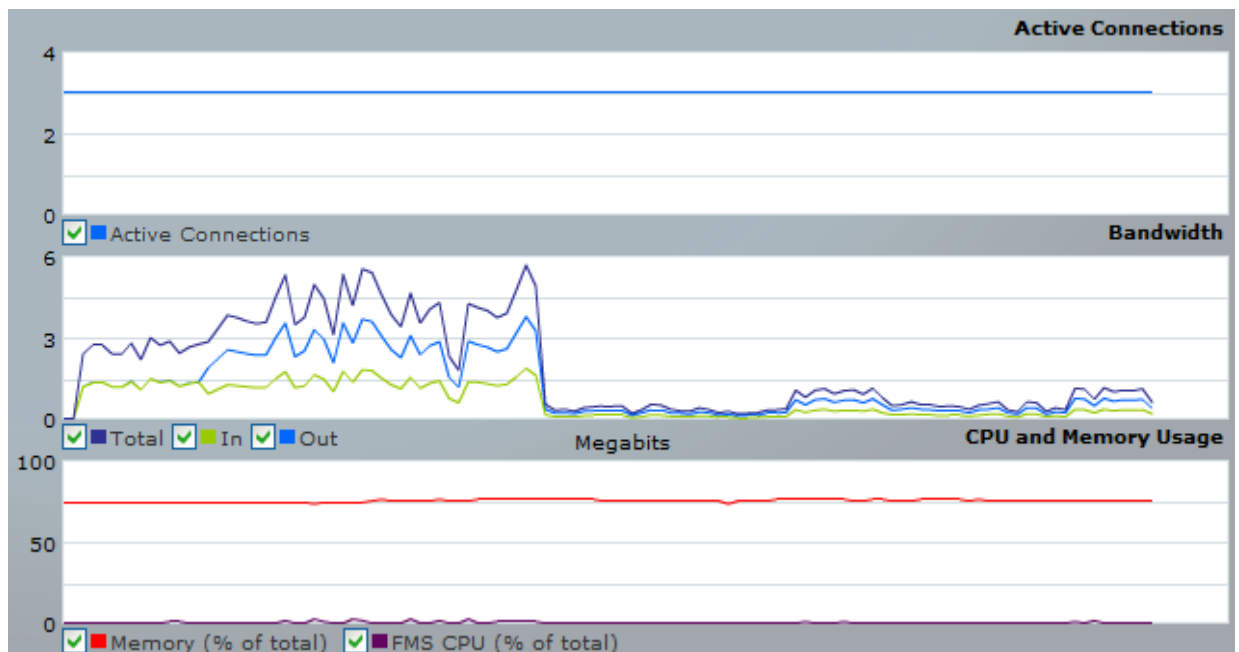
La segunda prueba se realizó con una transmisión de video en vivo pero con la mínima calidad permitida por la aplicación con 3 usuarios conectados en forma simultánea, en este caso se pudo observar como la imagen se tornaba pixelada y sin nitidez alguna, además el consumo de ancho de banda se redujo considerablemente llegando a unos niveles considerablemente bajos como se muestra en la siguiente imagen.



**Figura 12. Consumo de ancho de banda emitiendo video en vivo con baja calidad**

Fuente: [Autor].

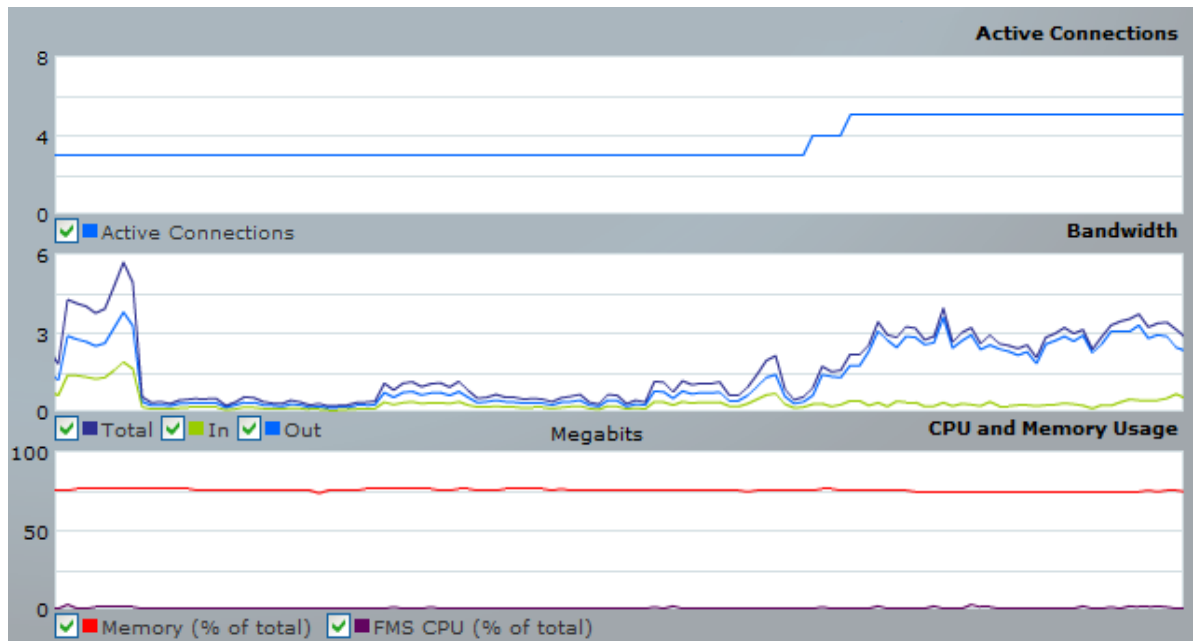
En la tercera prueba con video en vivo se realizó una transmisión con una calidad media y 3 usuarios conectados en forma simultánea, en este caso observó una calidad de imagen buena sin ralentizaciones ni saltos, además el consumo de ancho de banda tuvo algunos altibajos lo que quiere decir que no fue muy alto ni muy bajo como.



**Figura 13. Consumo de ancho de banda emitiendo video en vivo con calidad media**  
 Fuente: [Autor].

**3.1.2.2 Pruebas video por demanda y video en vivo.** En estas pruebas se transmitió simultáneamente video en vivo y video por demanda con un total de cinco usuarios conectados; cabe anotar que la emisión de video en vivo fue realizada con una calidad media.

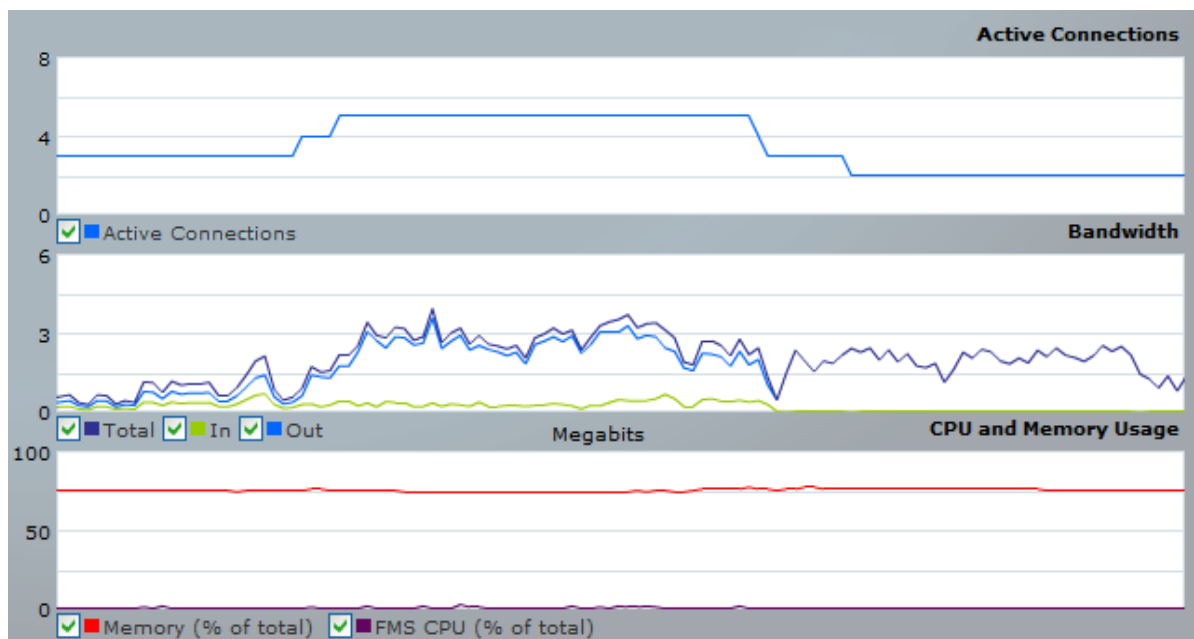
En esta prueba no se tuvo ninguna ralentización ni saltos que pudieran afectar la óptima visualización del video en vivo ni el video por demanda, además se observó un incremento en el consumo del ancho de banda de la red, esto debido a que se estaban emitiendo dos flujos de streaming de forma simultánea y al aumento del número de usuarios.



**Figura 14. Consumo de ancho de banda emitiendo video por demanda y video en vivo con cinco usuarios simultáneos**

Fuente: [Autor].

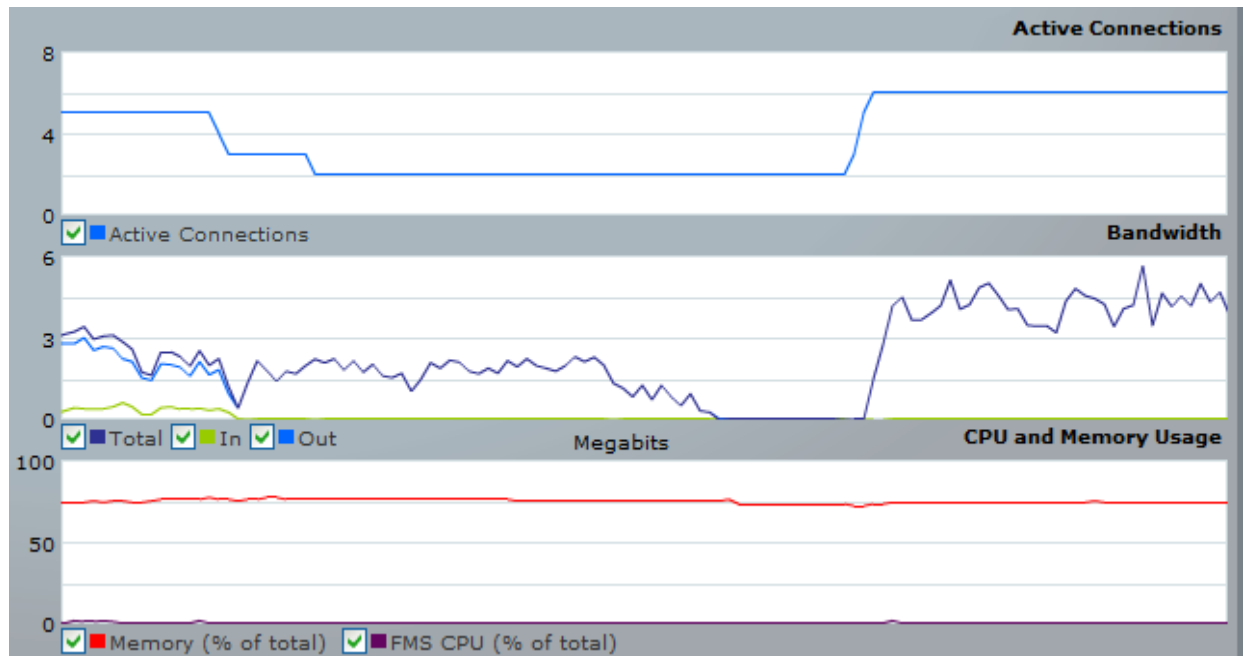
También se logró ver una ligera reducción del consumo de ancho de banda cuando el número de usuarios pasó de cinco a dos usuarios solamente.



**Figura 15. Consumo de ancho de banda emitiendo video por demanda y video en vivo con dos usuarios simultáneos**

Fuente: [Autor].

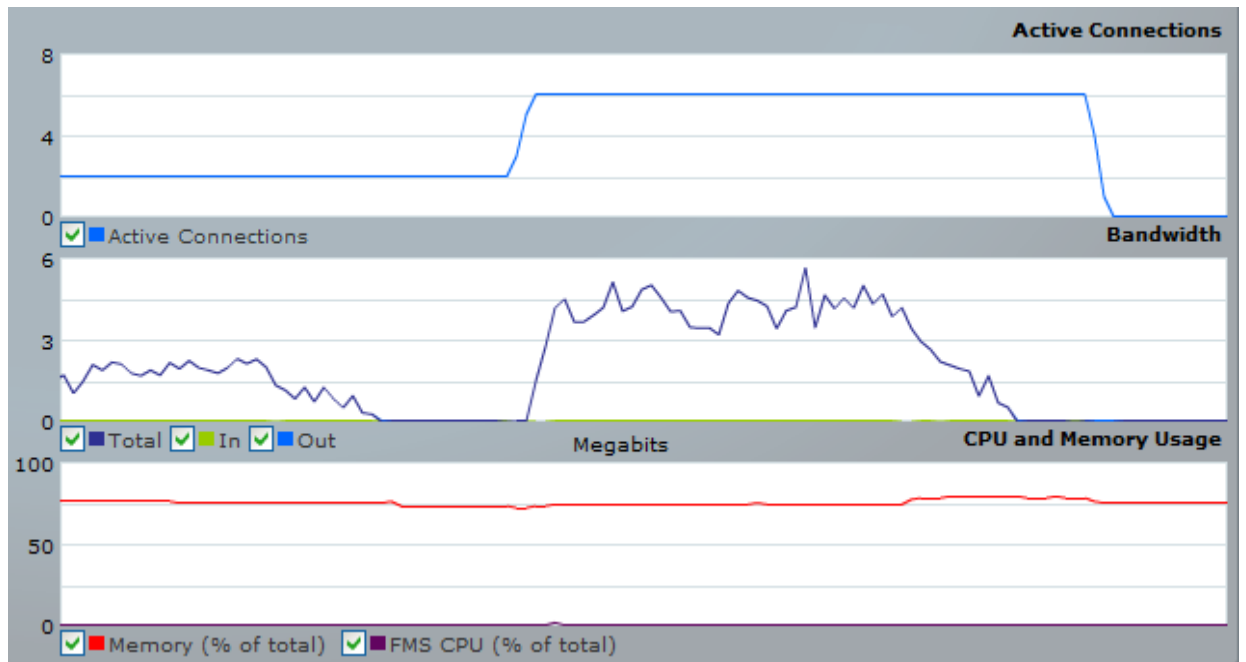
**3.1.2.3 Prueba video por demanda.** Esta prueba se realizó únicamente transmitiendo video por demanda con seis usuarios conectados de manera simultánea. En esta prueba se pudo observar como el consumo de ancho de banda se elevó considerablemente, pero no se evidenció ninguna ralentización y salto en los videos que estaban siendo visualizados.



**Figura 16. Consumo de ancho de banda emitiendo video por demanda con seis usuarios simultáneos**

Fuente: [Autor].

Cuando el número de usuarios conectados empezó a disminuir se logró ver como el consumo de ancho de banda comenzaba a bajar hasta el punto de llegar a cero.



**Figura 17. Consumo de ancho de banda al finalizar las pruebas**  
 Fuente: [Autor].

## 4. CONCLUSIONES

Se logró diseñar una aplicación de prototipo, la cual sirvió para mostrar como es el desarrollo de un sistema de streaming de video por demanda y video en vivo a través de una red de computadoras usando como base el protocolo RTMP (Real Time Messaging Protocol).

Durante las pruebas se pudo observar que el protocolo RTMP permite una transmisión rápida y eficiente de un flujo de video a través de una red cableada o inalámbrica ya sea de forma local o en Internet.

Se pudo determinar que a mayor número de usuarios conectados el consumo de ancho de banda de la transmisión de video por demanda se incrementa de forma significativa.

Se pudo observar que en el caso del video en vivo el consumo del ancho de banda depende del número de usuarios y de la calidad del video que se emite. Se conocieron las diferentes características de las plataformas disponibles para la realización de streaming de video a través de una red de computadoras. Se identificaron los protocolos más usados para la transmisión de un flujo de video sobre una red de datos.

Se definieron los conceptos básicos sobre el streaming de video.

Aunque se lograron cumplir los objetivos propuestos se encontraron algunas dificultades como fue la documentación para lograr desarrollar la aplicación ya que unos ejemplos se encontraban en lenguaje ActionScript 2 y la versión de Flash utilizada trabajaba con ActionScript 3 lo que conllevaba a un cambio en el código fuente de los scripts usados, puesto que las diferentes clases, librerías y sintaxis han sufrido modificaciones de una versión a otra.

Además es necesario un buen manejo de la programación orientada a objetos ya que sin unas buenas bases en este tema el desarrollo de cualquier aplicación en la plataforma Flash puede resultar un poco tedioso y complicado.

## BIBLIOGRAFÍA

- [1] Kruchten, Philippe. The Rational Unified Process An Introduction. Reading, Massachussets: Addison Wesley, 2000. 320p.
- [2] Schmuller, Joseph. Sams Teach Yourself UML in 24 Hours. Indianapolis, Indiana: Sams, 2004. 504p.
- [3] Sitio Web de YouTube. [En línea]. <http://www.youtube.com/>. [Consulta: 19 de Marzo 2011].
- [4] Sito Web de Ustream. [En línea]. <http://www.youtube.com/>. [Consulta: 19 de Marzo 2011].
- [5] Estudio de ip/tv multicasting para la universidad don bosco. [En línea]. [www.udb.edu.sv/Biblioteca/tesis/38872.pdf](http://www.udb.edu.sv/Biblioteca/tesis/38872.pdf). [Consulta: 10 de Noviembre 2010].
- [6] Arquitectura distribuida para una aplicación de videoconferencia Web. [En línea]. [irss.dit.upm.es/users/jcervino/documents/19](http://irss.dit.upm.es/users/jcervino/documents/19). [Consulta: 10 de Noviembre 2010].
- [7] Implementación de la tv por Internet en la universidad tecnológica de panamá. [En línea]. [http://www.virtualeduca.info/fveducasd/index.php?option=com\\_content&view=article&id=366%3Aimplementacion-de-la-tv-por-internet-en-la-universidad-tecnologica-de-panama&catid=36%3Aformacion-continua-profesional-y-corporativa&Itemid=56&lang=es](http://www.virtualeduca.info/fveducasd/index.php?option=com_content&view=article&id=366%3Aimplementacion-de-la-tv-por-internet-en-la-universidad-tecnologica-de-panama&catid=36%3Aformacion-continua-profesional-y-corporativa&Itemid=56&lang=es). [Consulta: 10 de Noviembre 2010].
- [8] Streaming de audio a través de dispositivos móviles utilizando j2me. [En línea]. [artemisa.unbosque.edu.co/facultades/sistemas/webinves/tesis0602/DocPrietoRodriguez.pdf](http://artemisa.unbosque.edu.co/facultades/sistemas/webinves/tesis0602/DocPrietoRodriguez.pdf). [Consulta: 20 de Noviembre 2010].
- [9] Diseño e implementación de una videoteca digital. [En línea]. [cita2003.fing.edu.uy/articulosvf/98.pdf](http://cita2003.fing.edu.uy/articulosvf/98.pdf). [Consulta: 20 Noviembre 2010].
- [10] Captura y despliegue de video remoto en dispositivos móviles. [En línea]. [www.revistas.unal.edu.co/index.php/avances/article/download/9926/10458](http://www.revistas.unal.edu.co/index.php/avances/article/download/9926/10458). [Consulta: 1 Diciembre 2010].
- [11] La educación virtual en ambientes iptv basada en servicios de tercera generación. [En línea]. [www.iiis.org/CDs2008/CD2009CSC/CISCi2009/PapersPdf/C870SJ.pdf](http://www.iiis.org/CDs2008/CD2009CSC/CISCi2009/PapersPdf/C870SJ.pdf). [Consulta: 1 Diciembre 2010].

Prieto Serpa, Felipe Andrés; Rodríguez Rodríguez, Luisa Fernanda. Streaming de Audio a Través de Dispositivos Móviles Utilizando J2ME. Tesis de Pregrado Ingeniería de Sistemas. Bogotá D.C : Universidad EL BOSQUE, Noviembre 2008. 107p.

Grupo Mnemmix. Soluciones de Streaming. [En línea] [www.mnemmixip.com/html/descargas/mirame.pdf](http://www.mnemmixip.com/html/descargas/mirame.pdf). [Consulta: 1 de Septiembre 2010].

Simpson, Wes. Video Over IP: A Practical Guide to Technology and Applications. Burlington: Focal Press, 2008. 504p.

Simpson, Wes; Greenfield, Howard. IPTV and Internet Video: New Markets in Televisión Broadcasting. Burlington: Focal Press, 2007. 240p.

Richardson, Iain E. The H.264 Advanced Video Compression Standard. United Kingdom: Wiley, 2003. 348p.

Austerberry, David. The Technology of Video and Audio Streaming. Burlington: Focal Press, 2005. 357p.

Reinhardt, Robert. Video with Adobe® Flash® CS4 Professional Studio Techniques. Berkeley: Adobe Press, 2009. 400p.

RFC 1889 - RTP: A Transport Protocol for Real-Time Applications. [En línea]. <http://www.faqs.org/rfcs/rfc1889.html>. [Consulta: 10 de Septiembre 2010].

QoS QualityOf sevice VoIP. [En línea]. [http://www.voipforo.com/QoS/QoS\\_Jitter.php](http://www.voipforo.com/QoS/QoS_Jitter.php). [Consulta: 13 de Septiembre 2010].

RFC 2326 - Real Time Streaming Protocol (RTSP). [En línea]. <http://www.faqs.org/rfcs/rfc2326.html>. [Consulta: 13 de Septiembre 2010].

Psaier, Harald. A Java-Based Streaming Media Server. Tesis. Viena : Institut für Interaktive Medien Systeme der Technischen Universität Wien. 2005. 76p.

RTMP Specification License. [En línea]. [http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp\\_specification\\_1.0.pdf](http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf). [Consulta: 14 de Septiembre 2010].

Real-Time Media Flow Protocol. [En línea]. [http://download.macromedia.com/pub/labs/flashplayer10/flashplayer10\\_rtmfp\\_faq\\_081010.pdf](http://download.macromedia.com/pub/labs/flashplayer10/flashplayer10_rtmfp_faq_081010.pdf). [Consulta: 18 de Septiembre 2010].

Adobe Flash Media Server 3.5. [En línea].

[www.adobe.com/.../flashmediaserver/pdfs/fms3\\_5\\_wp\\_ue.pdf](http://www.adobe.com/.../flashmediaserver/pdfs/fms3_5_wp_ue.pdf). [Consulta: 25 de Septiembre 2010].

Larson, Lisa; Costantini, Renée. Flash Video for Professionals: Expert Techniques for Integrating Video on the Web. Indianapolis, Indiana, 2007. 387p.

Unreal Media Server Overview. [En línea]. <http://www.umediaserver.net/umediaserver/overview.html>. [Consulta: 21 de Septiembre 2010].

Helix Universal Media Server. [En línea]. <http://www.realnetworks.com/helix/products/server/>. [Consulta: 21 de Septiembre 2010].

Windows Media Services 2008 Overview. [En línea]. <http://technet.microsoft.com/en-us/library/ee822833%28WS.10%29.aspx>. [Consulta: 21 de Septiembre 2010].

Darwin Streaming Server. [En línea]. <http://dss.macosforge.org/>. [Consulta: 21 de Septiembre 2010].

ActionScript Technology Center. [En línea]. <http://www.adobe.com/devnet/actionsript.html>. [Consulta: 19 de Marzo 2011].