



resumen

El presente artículo tiene como finalidad presentar un modelo de comunicaciones para la transmisión de audio en tiempo real y almacenado sobre Internet mediante el desarrollo de un componente de *software* bajo arquitectura cliente - servidor. Para su desarrollo fue necesario tener en cuenta algunos parámetros establecidos sobre APIs de Java para transmisión de audio, elementos relacionados con transmisión y estándares de comunicación para el soporte de tráfico multimedial.

# Modelo de comunicación multimedial para la transmisión de audio sobre internet

Luis Carlos Cañaverall S. • Paulo Alonso Gaona G.

introducción

**Dentro** de las transmisiones que se realizan en Internet hay una que ha tomado mucha fuerza en los últimos tiempos y es la transmisión multimedial en tiempo real, por una parte gracias a la difusión que ha tenido la banda ancha a lo largo y ancho de toda la sociedad y la gran acogida que ha tenido en la mayoría de empresas para disminuir costos y tiempos en desplazamiento. Pero esa disminución de costos lo contrarresta la tecnología necesaria para poder realizar este tipo de transmisiones, ya que para poderlas utilizar se necesitan realizar implementaciones que van desde el punto de vista arquitectónico mediante el *hardware* con equipos sofisticados para realizar encaminamiento de este tipo de transmisión, como desde el punto de vista lógico mediante el desarrollo e implementación de una arquitectura a nivel de software que me permita establecer una comunicación en tiempo real.

El propósito del presente artículo es presentar el diseño realizado del desarrollo de una plataforma (compendio de implementación conocida como framework) que dará soporte a uno de los productos transmisión de audio conocida como “Emisora Hermes”.





**cuatro**  
**ORISUJ**

## Marco teórico

Existen pocos proyectos en el campo investigativo que trabajen la parte de transmisión de medios multimedia por Internet, es por ello que impulsado por la evolución y los resultados obtenidos a nivel transaccional de este tipo de transmisiones en el grupo de investigación phEANix<sup>1</sup>, se dieron las bases para empezar a determinar elementos y requerimientos desde el punto de vista lógico y de transmisiones a nivel de Internet.

A nivel de investigación, la Universidad Nacional de Colombia viene desarrollando un proyecto para tratar de implementar espacios virtuales colaborativos de aprendizaje bajo la modalidad de e-learning<sup>2</sup>. Inicialmente ellos empezaron a trabajar sobre arquitectura J2EE y sobre el API JMF, integrados en una arquitectura multi-capas. Pero finalmente dentro de muchos estudios que realizaron de interconectividad, desistieron en seguir trabajando sobre esta arquitectura y decidieron trabajar sobre una arquitectura propietaria de Macromedia Flash, llamada Macromedia Breeze.

En Europa existen proyectos de nombre, cabe resaltar el que esta llevando a cabo la Universidad de Granada con el proyecto etutor<sup>3</sup>, el cuál maneja el mismo concepto de comunicación mediante videoconferencia y su arquitectura se basa en J2EE y el API JMF.

Uno de los medios más difundidos a nivel comercial es Progressive Networks, propietaria del software de emisión de audio y vídeo RealPlayer (antes RealAudio), Netscape y la Universidad de Columbia, la cuál trabajaron juntos para encontrar una solución al envío de datos multimedia a través de redes basada en IP. El resultado final fue un nuevo protocolo. Se trata de un protocolo de control para dirigir la emisión de audio/vídeo desde un servidor. Pero a nivel transaccional este tipo de soluciones, hacen que los buffers de datos se llenen y por lo tanto se tiene que refrescar la conexión, dependiendo de la ubicación del cliente.

## Audio digital

Gracias a los avances tecnológicos que se vienen desarrollando en aras a mejorar la comunicación, se plantea una serie de elementos dinamizadores que permean todas las características para el desarrollo de una arquitectura acorde a un modelo de comunicación en tiempo real. Dentro de los elementos que hacen parte de las necesidades de la transmisión se encuentran los medios acústicos, tal es el caso del sonido a través del audio. El audio se conoce como la representación de ondas sonoras y para poderlo





digitalizar se requiere del almacenamiento de su amplitud y de su frecuencia cada cierto periodo de tiempo [1], a este periodo de tiempo se le conoce como *sample rate*.

La señal de audio se conoce como la representación de una señal eléctrica en una señal sonora [2] y la representación de este tipo de señales se realiza de manera binaria que es finalmente lo que entiende todos los sistemas de información para establecer una comunicación, a esta representación se le conoce como el nombre de cuantificación [3]. Finalmente para poder transmitir esta secuencia binaria, se necesitan adaptarlo al medio de transmisión, es por ello que se utiliza una codificación adecuada para no tener pérdidas de datos binarios [4].

**El paradigma de la comunicación conlleva a retos y dificultades. Al iniciarse la exploración de canales de comunicaciones apareció Arpanet<sup>4</sup>, desarrollada con fondos del ministerio de defensa americano, es hoy en día una arteria extendida de lo que conocemos hoy como Internet.**



Con Internet actualmente podemos disipar distancias, ya sea con servicios básicos como lo es el e-mail, los FTP (File Transfer Protocol), SVN (Sub Version Network), RSTP (Real Time Secure Transfer Protocol) o con implementaciones híbridas de los anteriores como los conocidos, MSN Messenger, Skype, Google Talk entre otros. Percibimos a Internet como una fuente de conocimiento, gracias al pensamiento comunitario de algunos autores los cuales alojan grandes cantidades de información teórica e empírica, esperando solo que se difunda el conocimiento. Aprender a utilizar Internet es embarcarse en un aventura, nunca se podrá pronosticar que encontraremos en nuestra siguiente búsqueda.



## **Transmisión de audio sobre IP**

Recientemente se ha propuesto un sinnúmero de técnicas para automatizar el análisis de la información del audio [5], gracias a estos avances tecnológicos, se plantean una serie de elementos dinamizadores que permean todas las características para el desarrollo de una arquitectura acorde a un modelo de comunicación para transmisión de audio en tiempo real.

Para transmitir audio sobre Internet, se necesita que el componente que se vaya a implementar cumpla con las características y soporte de formatos adecuados para su trans-





misión en tiempo real, para ello es necesario identificar los parámetros para describir la secuencia de muestras que representa el sonido [6], son:

1. *El número de canales:* 1 para mono, 2 para estéreo, 4 para el sonido cuadrafónico, etc.
2. *Tasa de muestreo:* El número de muestras tomadas por segundo en cada canal.
3. *Número de bits por muestra:* Habitualmente 8 o 16 bits.

El costo de la transmisión de audio en vivo se calcula basándose en dos factores:

1. Calidad de la transmisión deseada. 24 Kbps., 32 Kbps., 56Kbps
2. Cantidad de usuarios conectados al mismo tiempo (simultáneamente).

Dentro de las características de la transmisión de audio sobre redes IP se encuentran algunos parámetros que se utilizan para medir la capacidad de los formatos de audio, a continuación se presentan algunos de ellos.

## Codificación de audio

Los esquema de codificación escalable presentes en este capítulo son representaciones tomadas de estudios realizados hasta el momento y estándares trabajados hasta el momento desde hace diez años [7], [8]. Existe un sin número de esquemas de compresión de audio representados en los modelos [9]. Las técnicas de compresión son la herramienta fundamental de la que se dispone para alcanzar el compromiso adecuado entre capacidad de almacenamiento y de procesamiento requeridas.

Según Levine [10] las técnicas de compresión más elaboradas proporcionan una reducción muy importante de la capacidad de almacenamiento, pero requieren también de un importante procesamiento tanto para compresión como para la descompresión, pero según Hamdy [11] las técnicas más simples ofrecen reducciones moderadas con poco procesamiento.

## Arquitectura planteada para el desarrollo del componente

Existen actualmente en el mercado, aplicativos de funcionalidad parcial, tales como Adobe Breeze<sup>5</sup>, Plone<sup>6</sup>, Microsoft NetMeeting entre otros. Este conjunto de aplicativos no satisfacen las necesidades del mercado puesto que son productos de software propietario<sup>7</sup>, no extensibles<sup>8</sup>, no escalables<sup>9</sup> y limitados por su funcionalidad.





En el proyecto se utilizaron tecnologías de software libre complementado con tecnologías propietarias. Para así ir orquestando un híbrido tecnológico capaz de cumplir los patrones de escalabilidad, extensibilidad y multiplataforma esperadas para un proyecto diseñado a la medida.

## Requerimientos para el diseño del Core de la aplicación

Para poder realizar cualquier tipo de comunicación es necesario contar primero con un medio que transporte para el transmisor a los receptores y/o viceversa o paralelamente (en dos direcciones). En los sistemas de transmisión de flujos de datos se requiere que este medio proporcione una conexión digital bidireccional y de alta velocidad entre los múltiples dominios a enlazar.

**Se hizo clara la necesidad de un proceso intermedio que encauzara peticiones y devolviera información en procesos separados y remotos, para así devolver estos múltiples procesos como un solo resultado. Con este objetivo se pensó en lo que hoy en día conocemos como aplicativo Web Dinámico (DHTML). Como HTML es un lenguaje interpretado, era necesario involucrar un tercero que generara flujos de datos por peticiones.**

Inicialmente se introdujo el CCI (Common Client Interface) lenguajes basados en DOM (Document Object Model), algunos ejemplos de la implementación de este estándar fueron Javascript, Microsoft Visual Basic entre otros. Su intención era buena, pero la lógica de negocio se quedaba en cliente y era un riesgo innecesario implementar software que podía ser visto por el usuario final sin mayor problema. Luego se pensó en un estándar para comunicar diferentes lenguajes de programación con el servidor Web, CGI (Common Gateway Interface), que leía la entrada estándar, procesaba y luego escribía el resultado en la salida estándar del servidor Web en forma de HTML estático. Entre otros CGI se podía comunicar con lenguajes como C, PERL, ASP entre otros.

Esta tecnología comprobó fácilmente que la Web podría ser más que una interfaz para presentar contenidos. Adoptado CGI como un patrón de diseño, se empezó a desarrollar CMS, LMS, Webmails y otro tipo de aplicaciones que satisficieran las necesidades del público creciente.





La arquitectura cliente - servidor proporcionó innumerables soluciones de software aun vigentes en este momento, aunque tiene algunas desventajas. La mas visible es, cuando una nueva petición se generaba desde el servidor Web el procesamiento para cada requerimiento era ejecutado de manera independiente y aislada. Consecuencia de lo anterior, un desperdicio de recursos de procesamiento que generaba las famosas sobrecargas (overLoad). Además dos capas no son suficientes para generar un esquema de seguridad que proteja de manera eficaz y eficiente el repositorio de datos y genere una arquitectura limpia.

## Arquitectura de la aplicación

**Con la acogida de lenguajes orientados a objetos se empezó a hablar de arquitecturas distribuidas y servidores de aplicación, que pudieran desarrollar esta lógica, los cuales separan la aplicación en abstracciones de accesos a medios.** Aunque el propósito de esta referencia conceptual no es enseñar al lector Aplicaciones Distribuidas y J2EE, explicaremos de forma corta para que el lector pueda darse una idea de las ventajas que representan un desarrollo distribuido.

- Capa pública. La conforman los GUI (Graphical User Interface) que interactúan de manera directa con el usuario, sean WBA (Windows Based Application) o un sistema de Webforms en un navegador Web. Esta capa se comunica con la capa lógica de negocio por medio de una Capa intermedia.
- Capa remota. Esta capa intermedia concede o niega accesos dependiendo del role del usuario.
- Lógica de negocio. Responde a las peticiones del usuario, y esta únicamente tiene acceso al repositorio de datos, que no es otra cosa que un envoltente (Wrapper) de nuestra Base de datos.

## Creación de archivos de intercambio y WebServices

Los lenguajes de programación, compilados o interpretados, propietarios o libres, son plataformas que pueden interactuar solo con objetos de su misma familia. Aunque muchas interfaces han sido desarrolladas con el fin de implementar híbridos capaces de comunicarse y interactuar, tales como Corba, RMI, entre otros, se han quedado cortos puesto que delimitan un campo de acción el cual solo puede ser soportado si la tecnología se quedara estática.



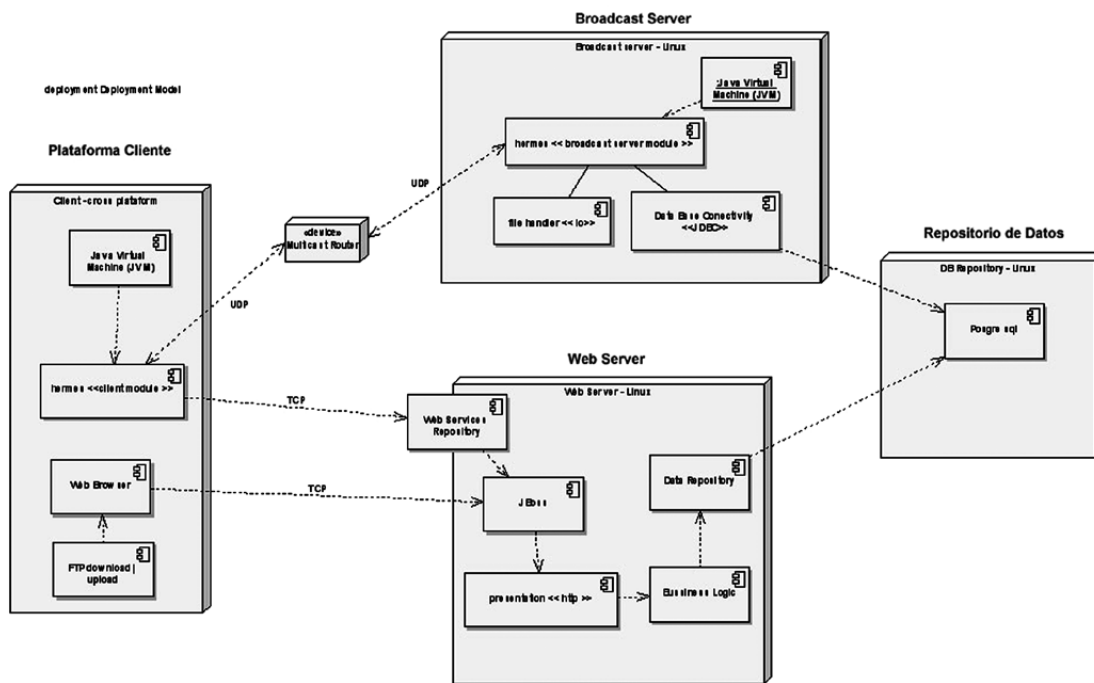


Figura No. 1 Modelo Funcional del Componente

Pero fue en la exploración de otra necesidad que se encontró una respuesta. Las aplicaciones ofimáticas codifican y decodifican la información contenida en un archivo para poder presentar e interpretar de manera adecuada la forma del documento (dedúzcase por forma todo aquello que mejora la presentación del contenido, estilos, sangrías, viñetas entre otros). Esto generaba muchos problemas cuando se quería cambiar de editor de ofimática o migrar a otro sistema operativo. Para compensar esta falencia fue creado XML (Extensible Marked Language) el cual podía ser decodificado e interpretado sin que su contenido fuera alterado. La premisa es separar el contenido de su presentación sin alterar la fidelidad en la presentación, para esto es necesario generar un intérprete el cual entienda el contenido del archivo XML y pueda presentarlo de forma apropiada.

Del mismo modo no hay que olvidar la importancia del componente a implementar, el cual cumple un papel importante a nivel de Aplicación dentro del modelo de comunicaciones TCP/IP; es por ello que la arquitectura que se propone para realizar la implementación cubre los aspectos desde el punto de vista arquitectónico a nivel de hardware





mediante dispositivos de internetworking y a nivel de software mediante la creación de un componente que funcione como un servicio orientado hacia la Web que permita realizar una transmisión de audio.

## Conclusiones

Los sistemas multimedia presentan características que van más allá de los alcances previstos para una comunicación, uno de los componentes primordiales a tener en cuenta es el tiempo real, que al mismo tiempo se convierte en una variable crítica dentro de un modelo de comunicación, esto, debido al alto valor prestacional que debe tener la red para poderlas tratar.

**Si bien en la actualidad se presentan variadas formas de comunicación en tiempo real mediante software propietario, cada una de ellas en el fondo presenta un modelo de desarrollo complejo que implica de cierta manera tiempo adicional por parte del usuario para descargar paquetes adicionales como plugins en Internet para poderlas compilar y de esta forma reproducirlas.**

Para el diseño del prototipo se tuvo en cuenta un enfoque independiente de la arquitectura, por lo tanto el prototipo funciona sobre cualquier plataforma en que se encuentre trabajando con el fin de poder determinar parámetros relacionados con la independencia total de las clases o funciones que se crean. El modelo planteado, el cuál se encuentra orientado al empaquetamiento total del aplicativo, permite realizar transmisiones de maneras seguras por Internet, al garantizar peticiones de conexión seguras mediante http.

1. Se logró implementar rutinas mediante sockets y comunicaciones mediante hilos para que el componente pudiera funcionar en un ambiente de trabajo de área local y fuera la base para la transmisión hacia Internet, esto con miras a mejorar el core del API.
2. Implementación de una rutina que permitiera al encapsulamiento RTP y determinara el cambio progresivo de puertos para establecer comunicación en tiempo real mediante utilización del protocolo RTCP para control de flujo de información, ya que es actualmente uno de los mayores problemas de este protocolo para su funcionamiento, lo que proporcionaría grandes huecos de seguridad en redes corporativas.







**Uno de los problemas encontrados en la implementación fue el encapsulamiento del audio, puesto que el flujo de datos con el protocolo RTP es algo complejo de manejar, ya que los puertos se escogen dinámicamente al establecerse una nueva conexión.**

Esto supone un serio problema para empresas que lleven a cabo políticas restrictivas de filtrado. Una forma de evitar estos problemas con el flujo RTP consiste en no filtrar los puertos por encima del 1024, ya que RTP selecciona puertos por encima de ese número. Esto en muchos casos puede resultar una solución inaceptable desde el punto de vista de seguridad de una red privada, por lo tanto en un punto a tratar para la parte de evolución del prototipo.

## Referencias

- [1]. M. Nelson, *The Data Compression Book*, 2nd Ed., New York: M&T Books, 1995.
- [2]. Li, Ze-Nian and Mark S. Drew. *Fundamental of Multimedia*. School of Computing Science Simon Fraser University. Ed. Pearson Prentice Hall, 2004
- [3]. Zanuy, Marcos, *Tratamiento Digital de Voz e Imagen y aplicación a la Multimedia*. Ed. Alfaomega, ISBN 970-15-0651-0. 2001
- [4]. Kleijn B y Paliwal K. *Speech Coding and Synthesis*. Ed Elsevier. Cap 3. 1995
- [5]. J. Foote, "An overview of audio information retrieval," *ACM Multimedia Systems*, vol. 7, pp. 2-10, 1999.
- [6]. López Martín, Alberto. *Formatos de Audio Digital*, Universidad de Valladolid, 2001.
- [7]. G. Stoll, G. Theile, S. Nielsen, A. Silzle, M. Link, R. Sedlmeyer, and A. Brefort, "Extension of ISO/MPEG-audio layer II to multi-channel coding—The future standard for broadcasting, telecommunication, and multimedia applications," in *Proc. 94th Conv. Aud. Eng. Soc.*, Mar. 1993, preprint 3550.
- [8]. J. D. Johnston et al., "The AT&T perceptual audio coder (PAC)," presented at the AES Convention, New York, Oct. 1995.
- [9]. B. Edler, et al., "ASAS-analysis/synthesis codec for very low bit rates", in *AES 100th Convention*, May 1996.
- [10]. S. Levine, *Audio Representations for Data Compression and Compressed Domain Processing*, PhD thesis, Stanford University, 1998.
- [11]. K. Hamdy, et al., "Low bit rate high quality audio coding with combined harmonic and wavelet representations", in *Proc. ICASSP*, Mayo 1996.
- [12]. Kahn, *Transmission-Control Protocol/Internet Protocol - Communications Principles for Operating Systems*. BBN. 1972





- [13]. Java Media Framework <http://java.sun.com/products/java-media/jmf> [Consulta: Abril 2006]
- [14]. Pinedo, Cristian. Arquitectura de una solución de Voz sobre IP real compatible. 2001.

## Notas de pie de página

- <sup>1</sup> Línea del grupo de Investigación Ontare – Facultad de Ingeniería Universidad EAN
- <sup>2</sup> Proyecto Espacios colaborativos virtuales de aprendizaje site: <http://www.virtual.unal.edu.co/unvPortal/articles/ArticlesViewer.do?reqCode=viewDetails&idArticle=3> Última Visita [5 Agosto 2007]
- <sup>3</sup> Proyecoe tutor site: <http://ceres.ugr.es/it/etutor/etutor.php> Última Visita [8 Agosto 2007 ]
- <sup>4</sup> Primera Red de comunicaciones montada con fines académicos y de Investigación en Estados Unidos
- <sup>5</sup> Sistema de e-learning basado en Adobe Flash Player, [www.adobe.com/es/products/breeze/](http://www.adobe.com/es/products/breeze/)
- <sup>6</sup> Sistema de e-learning basado en patrones LMS y CMS, <http://plone.org/>
- <sup>7</sup> Tipo de software con un costo inicial por su utilización y regido por los derechos de propiedad intelectual
- <sup>8</sup> Parámetro indicador en la adaptación de nuevos módulos a un software ya existente
- <sup>9</sup> Parámetro Indicador que determina la explicación de un software

autores

**Luis Carlos Cañaveral S.** Nació en Bogotá, Colombia. Es estudiante de último semestre de Ingeniería de Sistemas de la Universidad EAN, actualmente se desempeña como monitor Académico en las áreas de Redes y Telecomunicaciones.

**Paulo Alonso Gaona G.** Nació en Arauca-Arauca, Colombia. Es Ingeniero de Sistemas de la Universidad Distrital Francisco José de Caldas, de Bogotá-Colombia. Magíster en Ciencias de la Información y las Comunicaciones con Énfasis en Teleinformática en la Universidad Distrital Francisco José de Caldas de Bogotá, Colombia. Actualmente se desempeña como docente Académico en la Universidad EAN vinculado como docente de Planta desde el año 2004 en Bogotá. Es Instructor Cisco CCNA de la Academia Local EAN-UTP Colombia y pertenece al grupo de investigación Ontare adscrito a Colciencias bajo modalidad Categoría B desde el año 2005, donde realiza estudios sobre transmisiones multimediales y afines.

