DESARROLLO DE UN SISITEMA STREAMING DE AUDIO ORIENTADO A DISPOSITIVOS MÓVILES

LUZ CAROLINA JAIMES CONTRERAS
YESSIKA LILIANA PLATA JAIMES

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTA DE INGENIERÍAS FÍSICO MÉCANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2010

DESARROLLO DE UN SISITEMA STREAMING DE AUDIO ORIENTADO A DISPOSITIVOS MÓVILES

LUZ CAROLINA JAIMES CONTRERAS
YESSIKA LILIANA PLATA JAIMES

Trabajo de Grado para optar por el título de
Ingeniero de Sistemas

Director:
MPE. HENRY ARGUELLO FUENTES
Profesor
Escuela de Ingeniería de Sistemas e Informática - UIS
Codirector:
ING. IRENE LIZETH MANOTAS GUTIÉRREZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTA DE INGENIERÍAS FÍSICO MÉCANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2010

Dedicatorias

Dedico este trabajo a los seres que más amo: A Dios, A Henry Jaimes y Teresa Conteras mis padres, quienes con su amor y gran esfuerzo han logrado darme la oportunidad de estudiar y ser una mujer profesional.

A Marisol y Henry mis hermanos, quienes son parte importante de mi vida. A Sergio Alonso por creer en mí y por sus palabras de aliento en los momentos de decaimiento.

A mis compañeros de carrera y compañera de proyecto porque siempre me han ofrecido su amistad y ayuda durante todo el proceso de aprendizaje.

Luz Carolina.

"Dedico este trabajo a Dios por ser mi fortaleza y mi guía.

A mis padres Nelson Plata y Marlene Jaimes que con su amor, dedicación y apoyo han hecho de mi lo que soy ahora.

A mi hermana Silvia Plata, por ser mi motivación cada día.

A Diana Aguilar por su compañía en mi afán por alcanzar mi sueño.

A Luz Jaimes por subir conmigo el último escalón de mi meta.

A mis amigos y compañeros de carrera por su apoyo incondicional y su voz de aliento en todo el proceso para la culminación de este sueño."

Yessika.

AGRADECIMIENTOS

Los autores del presente proyecto expresan sus agradecimientos a:

A nuestras familiares y seres queridos, por todo el apoyo brindado durante el transcurso de nuestra carrera, por darnos su voz de aliento en los momentos difíciles y ser un apoyo incondicional en la culminación del proyecto.

Máster Henry Arguello, director de trabajo de grado, por su apoyo durante la inicialización del proyecto.

Ing. Irene Lizeth Manotas Gutiérrez codirectora de trabajo de grado, Ingeniera de Sistemas de la Universidad Industrial de Santander, por su interés y apoyo permanente durante la elaboración del proyecto.

Ing. Sergio Antonio Pino Gallardo, Ingeniero de Sistemas de la Universidad Industrial de Santander, por su tiempo dedicado en enseñar y sus palabras de apoyo para llevar a cabo la elaboración de este proyecto.

A nuestros compañeros y amigos, que nos acompañaron durante el recorrido hacia nuestra meta.

A todos los demás profesores que contribuyeron a nuestra formación académica a lo largo de la carrera.

TABLA DE CONTENIDO

Pág.

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

Pág.

TABLA DE ANEXOS

Pág.

RESUMEN

Titulo: DESARROLLO DE UN SISTEMA STREAMING DE AUDIO ORIENTADO A DISPOSITIVOS MOVILES.*

Autores: Luz Carolina Jaimes Contreras, Yessika Liliana Plata Jaimes**.

Palabras Claves: Streaming, J2ME, RTSP, RTP, códec de audio AMR.

Descripción: Este trabajo de grado presenta el desarrollo de una aplicación móvil que permite transmitir audio desde un servidor a un cliente en el que se ha instalado una aplicación, a través de la técnica de streaming.

Para el presente trabajo de grado se utilizó el lenguaje de programación Java Micro Edition (J2ME) y la librería de interfaz gráfica LWUIT para el desarrollo de la aplicación movil. Debido a la falta de soporte del lenguaje J2ME para el manejo de streaming a través de protocolos RTSP y RTP, se requirió su implementación en la aplicación para la comunicación y transmisión del material de audio desde el Darwin Streaming Server, servidor streaming escogido para el sistema hacia el dispositivo móvil.

La aplicación desarrollada ofrece una solución para la transmisión de audio multimedia en dispositivos móviles utilizando protocolos propios de streaming, aun cuando los dispositivos moviles no tengan soporte para dichos protocolos. Sin embargo, los dispositivos móviles deben cumplir ciertos requisitos de hardware y de software para el correcto funcionamiento de la aplicación que se describen en el capitulo quinto. Las pruebas realizadas se exponen en el capitulo siete, en las cuales se observa una comparación de la eficiencia al aplicar protocolos propios de streaming frente a otros protocolos en el proceso de transmisión de contenido de audio.

* Trabajo de investigación

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Henry Arguello Fuentes. Codirector: Irene Lizeth Manotas Gutiérrez.

ABSTRACT

Title: DEVELOPMENT OF STREAMING AUDIO SYSTEM ORIENTED MOBILE DEVICES.*

Authors: Luz Carolina Jaimes Contreras, Yessika Liliana Plata Jaimes**.

Keywords: Streaming, J2ME, RTSP, RTP, códec de audio AMR.

Description: This paperwork shows the development of a mobile application that allows to transmit audio from a server to a client in which an application is been installed through the streaming technique.

In order to do this paperwork the program language Java Micro Edition (J2ME) and also the library of graphical interface LWUIT was used for its development on mobile devices.Because of the lack support of the language J2ME for the managing of streaming through protocols RTSP and RTP, it required its implementation in the application for communication and transition of the audio material from the Darwin Streaming Server, chosen for the system to the mobile device.

The developed application offers an answer for the transmission of audio multimedia in mobile devices using streaming protocols, even though if mobile devices do not have support for these mentioned protocols. However, the mobile devices must fulfill certain requirements of hardware for the right functioning of the application that is described in chapter five. The done test shown in chapter seven, in which can be observed a comparison of the efficiency when streaming protocols are applied in comparison to other protocols in the transmission process of audio content.

* Research Work

** Faculty of Mechanical Engineering Physics. School of Engineering and Computer Science. Director: Henry Arguello Fuentes. Codirector: Irene Lizeth Manotas Gutiérrez.

INTRODUCCIÓN

La difusión de contenido multimedia por internet a través de streaming es uno de los servicios más atractivos actualmente. Sin embargo en un principio esta técnica sólo se orientó a equipos de escritorio limitando los alcances de la misma. El auge de los teléfonos móviles con capacidades multimedia que además permiten acceso inalámbrico a internet, dieron un vuelco al concepto de streaming orientándolo a dispositivos que permiten la movilidad del usuario como son los teléfonos celulares.

El presente trabajo de grado expone el desarrollo de un sistema prototipo de streaming de audio orientado a dispositivos móviles (celulares). El sistema está compuesto por un servidor web, un servidor de streaming y la aplicación móvil desarrollada en el lenguaje J2ME. Los clientes móviles acceden a los contenidos del servidor a través de la aplicación, realizando la transferencia de archivos por la técnica de streaming utilizando las redes de telefonía móvil o redes inalámbricas dependiendo de las capacidades del dispositivo celular a utilizar.

El trabajo de grado presentado en este libro fue desarrollado en el grupo de investigación en ingeniería biomédica — GIIB de la Universidad Industrial de Santander UIS — y es el primero relacionado con el tema de streaming móvil hecho en la universidad. Actualmente en nuestro país el estudio de este tema es aún muy escaso.

El capitulo 1 aborda la descripción del problema, presentando también la justificación del proyecto en la cual se describe las ventajas de utilizar streaming multimedia para la trasmisión de archivos por la red. Finalmente se exponen los objetivos que se quieren alcanzar con la realización de este trabajo de grado, dando una solución a la problemática planteada.

El capítulo 2 presenta el estado del arte en el cual se muestra los estudios previos sobre streaming, además se citan algunas de las principales compañías que actualmente ofrecen este servicio. También se expone el marco teórico que fundamenta este trabajo de investigación, conjuntamente se describe los principales componentes que conforman un sistema streaming multimedia. Por último, este capítulo presenta un estudio sobre la tecnología J2ME; lenguaje utilizado para el desarrollo de la aplicación, al igual que el paquete opcional MMAPI (Mobile Media API) como herramienta que permite procesar y reproducir contenido multimedia en aplicaciones J2ME.

El capítulo 3 presenta las actividades efectuadas para el cumplimiento de uno de los objetivos específicos del proyecto; en cual consistió en determinar las características de implementación de un servidor streaming de audio orientado a clientes en dispositivos móviles. Para esto se realizó el estudio de algunos servidores streaming que permiten la transferencia de datos hacia clientes móviles. Por último se presentan las pruebas realizadas y se exponen los motivos para la selección del servidor streaming.

El capítulo 4 se hace una breve descripción de las redes de telefonía celular ideales para prestar un servicio streaming, indicando sus características principales.

El capítulo 5 plantea la metodología seguida para el desarrollo del proyecto describiendo las actividades realizadas e igualmente se listan los requerimientos de hardware y software para los componentes del sistema.

El capítulo 6 explica el desarrollo de la aplicación móvil STREAMJASSMOBILE, aquí se detalla cómo fue implementado los protocolos RTSP y RTP en J2ME para realizar el streaming. Por último, el capítulo 7 muestra los resultados obtenidos y las pruebas realizadas para verificar el funcionamiento de la aplicación.

CAPITULO I

1. PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción del Problema

Durante los últimos años las tecnologías de la información y comunicación han evolucionado en varios aspectos, uno de ellos es la adquisición de los recursos, siendo cada vez más las personas que tienen acceso a estas tecnologías. De igual manera, la evolutiva y amplia capacidad de los sistemas informáticos de usuario y de las redes que los intercomunican ha impulsado la aparición y el estudio de técnicas de entrega y distribución de contenidos multimedia.

Sumando a estos aspectos, el auge de los teléfonos móviles con capacidades multimedia y el acceso inalámbrico a Internet crea un entorno tecnológico que posibilita el desarrollo de diversas aplicaciones. Esta evolución presente en la actualidad ha modificado profundamente las relaciones entre las personas encaminándolas hacia una sociedad altamente interconectada donde el eje fundamental es la información.

Sin embargo, no basta solo con comunicarse de la manera tradicional. Las personas manejan un volumen considerable de información, no solo a manera de texto, sino también en contenido multimedia, y se hace indispensable tener en todo momento esta información disponible, es aquí donde la telefonía móvil es una opción para una comunicación más eficaz.

Aun así existen inconvenientes, la movilidad tiene implícita retos que aun se están afrontando, uno de ellos y además de importante, es la limitada capacidad de almacenamiento de los dispositivos móviles, entre otras, que continuarán surgiendo a medida que se avance en la prestación de los diferentes tipos de servicios.

Los avances en los servicios multimedia han producido, después de muchos años de investigación, una técnica optimizada de comunicación denominada streaming, esta técnica es utilizada para transmitir información por la red de tal forma que su emisión desde el servidor, su comunicación por la red y la descarga y procesamiento en el cliente se produce simultáneamente sin necesidad de guardarlo en la memoria del cliente.

Se plantea entonces como solución a la adquisición de información a tiempo y en todas partes, la realización de un sistema de streaming orientado a dispositivos móviles. El sistema streaming tendrá como objetivo entregar material digital de audio a clientes móviles que hagan parte de su red de distribución, mediante la implementación de un servidor en donde se encontrará el material digital de audio y que proveerá este servicio a través de una interfaz gráfica amigable alojada en un dispositivo móvil.

1.2 Justificación

Algunos años atrás existía la imposibilidad de trabajar con temas de multimedia en equipos móviles debido a las amplias restricciones que estos dispositivos presentan en cuanto a memoria, potencia en el procesamiento, tamaño de pantalla entre otros, además del exiguo ancho de banda de las redes. Sin embargo, el auge de la telefonía móvil, ha impulsado significativamente el desarrollo de dispositivos que incorporan diversas tecnologías de hardware y software a costos cada vez menores.

En la actualidad los dispositivos móviles disponen de mayor capacidad de almacenamiento y procesamiento; poseen además diferentes dispositivos periféricos incorporados (micrófono, cámara), pantallas de alta resolución, puertos de comunicación con otros dispositivos (infrarrojo, Bluetooth) y cuentan con servicios de redes de telefonía celular como 3G y GPRS que permiten enlaces de comunicación. Todo esto convierte a los dispositivos móviles en plataformas excelentes para el impulso de nuevas aplicaciones, donde la movilidad y la portabilidad son requisitos relevantes.

Gracias a estos avances actualmente es posible desarrollar aplicaciones que permiten la adquisición de contenido multimedia a través de internet para móviles como lo es el servicio streaming. En general esta tecnología permite aligerar la descarga y ejecución de contenido multimedia con la facilidad de reproducir sin tener que grabar la información en disco. Sin embargo, gran parte de esta solución fue orientada en un principio a equipos de escritorio, siendo aun precarias las soluciones que se ofrecen en este aspecto para dispositivos móviles. En la actualidad aplicaciones como Didiom[1] y Spotify[2] ofrecen un servicio de streaming de audio para clientes en equipos móviles.
Precisamente por dicha carencia, este proyecto se torna importante, surge de la necesidad de orientar los esquemas actuales del servicio streaming hacia aplicaciones en dispositivos móviles. Para el desarrollo del proyecto inicialmente se realizará la evaluación y selección del servidor de streaming observando los

[1] Didiom. (Online). 12 de Abril de 2009. http://www.didiom.com.
[2] Spotify. (Online). 22 de Mayo de 2009. http://www.spotify.com

criterios que cobran mayor importancia en cuanto a la distribución de contenido de audio, para luego realizar el desarrollo del software que preste el servicio streaming como interfaz de conexión entre el dispositivo y el servidor.

1.3 Objetivo General

Diseñar e implementar una herramienta software para dispositivos móviles que permita la adquisición de material digital de audio utilizando la técnica de streaming.

1.4 Objetivos Específicos

- Determinar las características de implementación de un servidor de streaming de audio orientado a clientes en dispositivos móviles.
- Determinar las características de los dispositivos móviles y de la red de acceso a internet para la implementación del servicio de streaming.
- Diseñar y desarrollar una aplicación software para acceder al servicio de streaming de audio en dispositivos móviles.
- Verificar el desempeño de la aplicación desarrollada en cuanto a la transferencia del material digital de audio y compararla con el servicio streaming de audio en un PC de escritorio.

CAPITULO II

2. MARCO TEORICO

2. 1 Antecedentes Históricos e investigaciones Previas

2.1.1 Aplicaciones Streaming en PC de escritorio

La trasmisión de información hace algunos años a través de Internet suponía siempre, que ésta no podía visualizarse hasta que se encontrara por completo en el ordenador del cliente. Esta noción cambió en el momento en que la información manejada en Internet pasó a ser algo más que solo texto.

Con la inclusión de archivos gráficos JPG a mediados de 1990 se mejoró el tiempo de recibir un archivo por Internet, las personas imaginaban la apariencia final que tendría la información que llegaba hasta su ordenador en forma de grandes pixeles de colores. No obstante, debido a la evolución de la tecnología, la trasmisión de video, música, animaciones, juegos y películas no se hicieron esperar.

Sin embargo, el terreno de las comunicaciones parecía sufrir una gran demora evolutiva en cuanto a las velocidades de trasmisión de la información, esto provocaba que la difusión de contenido multimedia encontrara obstáculos debido a las grandes esperas para visualizar, por ejemplo, un archivo de 5 MB que se reproduce en 2 minutos en donde se supone que se debía esperar 15 minutos para su descarga, contando con que no se cortara la conexión.

Es aquí donde como solución a este problema aparece la técnica de streaming, que consiste en la reproducción del archivo de video o música a medida que va llegando al ordenador del cliente y sin necesidad de tener el archivo en su totalidad para comenzar su escucha o visionado. Para ese entonces, Apple[3] abrió el campo multimedia en 1991.

El streaming, como tecnología surge en 1995, cuando la empresa RealNetworks[4] con el reproductor RealAudio, encontró la solución para emitir contenidos en directo por la World Wide Web para audio y posteriormente para video. En 1999

[3] Apple Inc. QuickTime (Online). 26 de Mayo de 2009. http://www.apple.com/es/quicktime/.
[4] RealNetworks, Inc. (Online), 26 de Mayo de 2009. http://www.realnetworks.com/.

Microsoft lanzó al mercado NetShow que más tarde pasaría a llamarse Windows Media[5]. Estas dos compañías terminarían por aliarse para dar origen al RealVideo, logrando que este reproductor permitiera recibir video y audio de mayor calidad con el plugin que crearon. En diciembre de 2001, Microsoft lanza Corona, después llamado Windows Media 9 que incluía una tecnología llamada FastStream que optimiza automáticamente el envío de video y audio para aprovechar al máximo el ancho de banda disponible del usuario.

Actualmente existen compañías que ofrecen servicios de streaming de contenido multimedia a través de Internet, una de las más conocidas y ya nombrada es RealNetworks[4] que se dedica a ofrecer servicios tanto gratuitos como por pago, de radio y televisión así como películas. Esta compañía desarrolló el servidor de streaming de audio y video denominando Hélix que cuenta con protocolos propios para el streaming de audio y video como el RTSP. Otra empresa que ofrece servicio streaming de audio es Shoutcast[6] ofreciendo a sus usuarios aplicaciones fáciles de usar para poder tener su propia radio en casa. La red de estaciones de radio de Shoutcast es tal vez la más popular entre los usuarios de internet y la más extensa en cuanto a contenido ofrecido. RTP (Real Time Transport Protocol) es el protocolo más usado en estas redes. Microsoft con Windows Media[5] y Apple con QuickTime[3] son otras de las compañías que ofrecen servicio streaming para PCs.

2.1.2 Aplicaciones streaming en Dispositivos Móviles

Grandes compañías ofrecen servicios streaming de audio y video como por ejemplo, Bell Canadá[7] que se ha encargado de ofrecer servicios de streaming para películas a lo subscriptores que posean dispositivos móviles que cumplen con una serie de características para soportar dicho servicio.

Dentro de las compañías que se dedican a desarrollar aplicaciones para dispositivos móviles con relación a streaming de audio y video se encuentra Packet Video[8] esta empresa se dedica a desarrollar aplicaciones para dispositivos móviles en varios entornos de desarrollo como: Symbian, BREW, Linux,Windows, MOAP.

Desde hace muy poco grandes proveedores de servicios para dispositivos móviles están interesados en dar soluciones de streaming mutimedia. Permitiendo explotar este campo y logrando que en estos dos últimos años se creen productos novedosos y revolucionarios como son los siguientes:

---

[5] Windows Media Inc. (Online). http://www.microsoft.com/windows/windowsmedia/es.

[6] Shoutcast (Online), 26 de Mayo de 2009. http://www.shoutcast.com/.

[7] Bell Canada. Bell. (Online), 26 de Mayo de 2009. http://www.bell.ca/home/.

[8] Packetvideo Corporation. Packet Video. (Online), 26 de Mayo de 2009. http://www.packetvideo.com/

- Didiom[1] lanzada a finales del año 2008, es una aplicación multimedia con dos facetas diferenciadas. Una es la tienda de música que tienen disponible, accesible tanto desde el ordenador como desde un teléfono móvil. Pero la más interesante es la función de streaming a un móvil de la música que se tenga almacenada en un ordenador. Instalando Didiom en una máquina se puede reproducir desde el móvil las canciones o las listas de reproducción que se tenga en iTunes.
- Spotify[2] lanzada en abril de 2009, es una aplicación que permite escuchar la música que se quiera sin necesidad de tenerla en el disco duro, directamente desde sus servidores de streaming.

Actualmente estas aplicaciones no están disponibles para todo el mercado de manera que no es posible acceder a ellas en nuestro país.
En nuestro pais Elibom[9] es una compañía líder en servicios de valor agregado a través de telefonía móvil con mensajería de texto SMS, mensajería de texto Premium PSMS, mensajería multimedia MMS, aplicaciones móviles J2ME y de internet móvil WAP. Sin embargo, a la fecha no se conoce de la existencia de alguna aplicación móvil que soporte streaming multimedia.

Finalmente en la Universidad industrial de Santander se ha tenido antecedentes de trabajos de grado, relacionado con las aplicaciones móviles, entre ellos están:

1. Software Aplicado A Dispositivos Móviles Para La Asistencia En Cirugía Vascular.

2. Movilinventory: Herramienta Software Para La Gestión Y Control De Inventarios de Activos Fijos Mediante Dispositivos Móviles PDAs.

3. Desarrollo De Juego Multijugador Bluetooth para Dispositivos Móviles Con J2ME.

Como se puede ver la transferencia de contenido multimedia a través de la técnica de streaming hasta ahora no ha sido tema de investigación para proyecto de grado en la Universidad Industrial de Santander.

[9] Elibom. (Online), 26 de Mayo de 2009. http://www.elibom.com.

2.2 Marco Teórico

En el siguiente mapa conceptual se muestran de manera resumida los conceptos teóricos con los cuales se fundamenta el proyecto.

2.2.1 Tecnología Streaming

2.2.1.1 Definición

El streaming es una técnica utilizada para la distribución de contenido multimedia en Internet [1], con la característica de poder visualizar y/o escuchar estos contenidos en el cliente cuando la información aun está siendo trasmitida por la red, de tal manera que no es necesaria la descarga completa del contenido multimedia en el cliente para que este pueda empezar su visualización.

Un sistema streaming funciona de la siguiente manera: En primer lugar el cliente se conecta con el servidor solicitando algún contenido que se encuentra almacenado allí y éste inicia la transmisión del archivo. El cliente comienza a recibir el archivo y construye un buffer en memoria donde es almacenado el contenido a medida que se recibe [1]. Cuando este buffer contiene la mínima parte del archivo el cliente empieza a mostrarlo mientras se continúa con la descarga. De esta manera el sistema está sincronizado para que pueda ser visto el contenido del archivo mientras se descarga, de modo que una vez éste termina de descargarse posiblemente también ha acabado de visualizarse. Las figura 1 ilustra el funcionamiento de un servicio de streaming

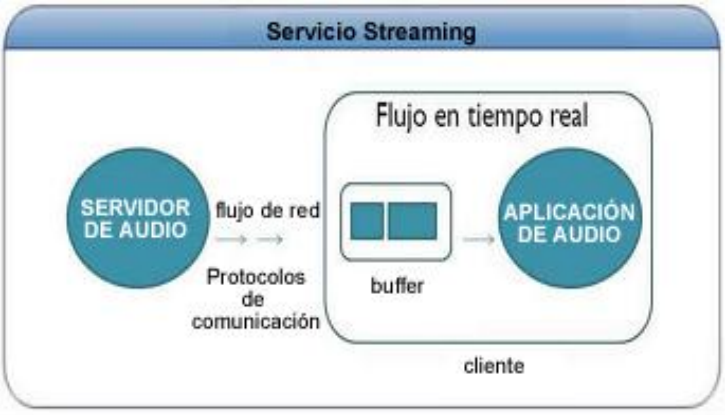


Figura 1. *Envío de un archivo desde el servidor al cliente*

Fuente: Imagen modificada por los autores obtenida de: Evaluación de Servidores Streaming [2]

2.2.1.2 Real Time en Streaming

En la tecnología streaming el término “tiempo real” no se emplea con exactitud en la trasmisión de contenido multimedia en la red, ya que en realidad este contenido se trasmite de manera confluente debido a que existe una cierta cantidad de contenido que se retrasa entre la transmisión y recepción por factores diversos factores como: el ancho de banda, congestión en la red o por las capacidades de recepción del cliente.

2.2.1.3 Principales componentes de un sistema Streaming

En la figura 2 se muestra los componentes principales para la realización de un sistema streaming, que consta de tres partes importantes: generación del contenido multimedia, servidor streaming y el reproductor del cliente.

- Generación del contenido multimedia: La primera parte es la generación del contenido a trasmitir, para esto es necesario digitalizar el contenido generado por la fuente de información si es que ésta genera contenido análogo, además es preciso editar y/o codificar los datos multimedia según los formatos designados para el sistema teniendo en cuenta aspectos como el ancho de banda de la red y los formatos soportados por el servidor y el reproductor del cliente.

- Servidor Streaming: Una segunda parte consta del almacenamiento y transmisión de contenido por parte del servidor. Se debe tener en cuenta que para la difusión de contenido multimedia por la red se hace imprescindible de servidores de streaming que gestione las conexiones y anchos de banda de los usuarios conectados al sistema.

- Reproductor del cliente: El reproductor del cliente es la última etapa de un sistema streaming. Lo más importante a tener en cuenta respecto al reproductor son los tipos de formatos de archivo que este puede reproducir. El reproductor es parte fundamental de un sistema streaming pues entre sus responsabilidades está la interacción con el usuario a través de una interfaz, el manejo de errores en la trasmisión y la decodificación de los datos recibidos.

Figura 2. *Servicio Streaming de audio*

Fuente: Imagen modificada por los autores obtenida de: Evaluación de Servidores Streaming [2].

2.2.1.4 Errores en la transferencia de datos vía streaming.

Cuando se envía contenido multimedia por la red a través de streaming es normal que se presenten errores durante la transferencia como perdida de información o daño en la misma. Esto se debe a diversos factores, los más frecuentes son:

• Latencia en la red: Ocurre cuando hay demasiado tráfico en la red por la cual viajan los paquetes, esto hace que se retrase su llegada al cliente.

• Bloqueo en la reproducción: Cuando existe latencia en la red o arribo de paquetes corruptos al cliente, el reproductor deja de funcionar debido a que no hay datos en el buffer para continuar con la reproducción.
• Tiempos muertos o silencios en la transmisión: Para la trasmisión de contenido a través de streaming multimedia se utilizan protocolos no orientados a la conexión como UDP, por lo mismo no se garantiza la entrega de paquetes al cliente. En el caso de contenido de audio esto puede generar silencios durante la reproducción.

2.3 Distribución y Esquemas de Distribución de Audio

2.3.1 Distribución de audio

Con la tecnología de streaming es posible tener acceso a servicios como videoconferencia, video o audio bajo demanda ó transmisión de eventos en vivo.

2.3.1.1 Bajo demanda

En una distribución de audio bajo demanda los archivos son almacenados en un servidor de streaming, por lo que es posible acceder al contenido y escucharlo en el momento que se desee. Una vez que el audio esté disponible para la transmisión y un cliente solicita un archivo, el servidor debe estar en la capacidad de hacer conexión con éste y con otras máquinas clientes que se encuentren también realizando alguna petición. En este momento el servidor empieza la entrega del flujo de bits para ser escuchado en el reproductor del cliente al otro lado de la conexión; llegado a este punto el usuario tiene la posibilidad de controlar y manejar el flujo según lo prefiera, pudiendo detener su ejecución, realizar una pausa o un retroceso. La figura 3 que se muestra a continuación, ilustra el funcionamiento de la distribución bajo demanda.

Figura 3. *Streaming Bajo Demanda*

Fuente: DINAMIC. Servicio Streaming. Disponible en: http://www.dinamic.com

2.3.1.2 Directo o En Vivo

Streaming en vivo o directo se refiere al flujo de contenido multimedia en tiempo real. En este caso es necesario el uso de un software de producción que permita codificar y editar el contenido y que tenga la capacidad de transmitirlo a un servidor desde el cual generar el flujo hacia los clientes. (Ver Figura 4)

En una distribución bajo demanda el cliente debe escuchar el canal por el cual fluye el contenido; en este caso es común el uso de grupos Multicast (ver este concepto más adelante) como destino, siendo el cliente el que demuestra la intención de recibir el flujo.

Figura 4. *Streaming En Vivo*

Fuente: DINAMIC. Servicio Streaming. Disponible en: http://www.dinamic.com/

2.3.2 Esquemas de Distribución

2.3.2.1 Broadcast

En redes de computadoras el término Broadcasting hace referencia a la difusión de paquetes de información por red que serán entregados simultáneamente a todos los clientes que pertenezcan a esta red. La difusión a través de Broadcasting está limitada a un dominio Broadcast. Generalmente las redes que soportan Broadcasting están asociadas a redes de área local [1].

2.3.1.2 Unicast

En redes de computadoras el término Unicast hace referencia a la transmisión de paquetes a través de una conexión uno-a-uno entre el servidor y el cliente, lo que significa que cada cliente recibe un flujo distinto y lo recibe sólo aquellos clientes que también lo soliciten.

2.3.1.3 Multicast

En redes de computadoras el término Multicast hace referencia a la transmisión de un flujo de datos proveniente de un servidor simultáneamente hacia un grupo de usuarios en la red. El flujo de datos es enviado a una dirección que identifica al grupo; de acuerdo al RFC 3171 las direcciones destinadas para ser Multicast están comprendidas desde la 224.0.0.0 a la 239.255.255.255. Este rango de direcciones se llama formalmente "Clase D". Para que los usuarios puedan recibir el flujo enviado por parte del servidor deben estar configurados con la dirección de red asignada al grupo. Para realizar una distribución Multicast, es necesario que los routers en el trayecto entre el servidor y el grupo estén habilitados para distribución Multicast.

2.4 Protocolos Streaming

El funcionamiento de Internet está basado en un conjunto de protocolos que permiten la difusión de datos entre redes de computadoras, muchas veces este conjunto de protocolos es llamado pila TCP/IP, ya que estos dos son los protocolos más importantes y los más utilizados, sin embargo existen más de 100 protocolos que componen la pila entre los más conocidos están: HTTP, ARP, FTP, SMPT, UDP, POP. entre otros. Cada uno de los protocolos mencionados tiene características que los hacen adecuados o no para ciertas funcionalidades como la transmisión de contenido multimedia.

Los protocolos User Datagram Protocol (UDP) [3] y RTSP [4] (Ver figura 5.) son ideales para la difusión de contenido multimedia por la red. Debido a que son protocolos no orientados a la conexión dan mayor importancia a la transmisión continua de flujo que a la integridad de la información. De esta manera si un paquete UDP se pierde en la red, el servidor continúa enviando datos, esto producirá un pequeño salto en la secuencia de reproducción del cliente, pero evitará que se detenga su ejecución. En protocolos orientados a la conexión como TCP es más importante reenviar el paquete extraviado antes de seguir con el flujo de datos, esto ocasiona retardos en la llegada de la información y por lo tanto rupturas en la reproducción del contenido, características no gratas en la difusión de audio y/o video.

El hecho que posibilitó la implementación de la tecnología streaming, fue la adopción del protocolo de Internet User Datagram Protocol (UDP). El protocolo UDP hizo factible el streaming multimedia permitiendo la transmisión de datos de una manera más eficiente que los demás protocolos comúnmente utilizados en Internet. Algunos otros protocolos más recientes como el Real Time Streaming Protocol (RTSP) hacen aun más eficiente este tipo de transmisiones.

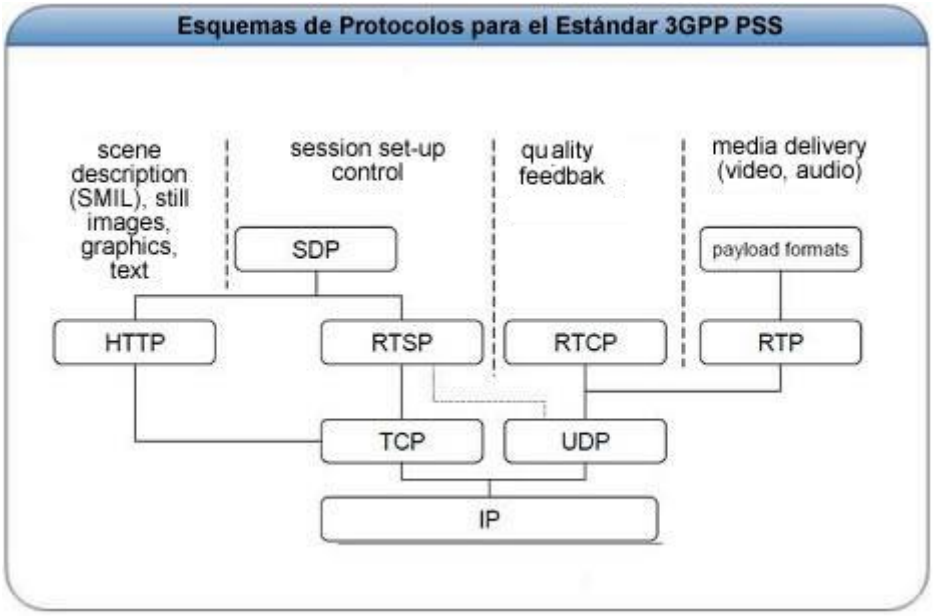


Figura 5. *Pila de protocolos para el estándar 3GPP PSS*

Fuente: Disponible en: http://www.medialab.sonera.fi/workspace/PacketSwitchedStreamingWP.pdf

2.4.1. Protocolo RTP

Las comunicaciones basadas en RTP (Real Time Protocol) [5] están orientadas a la transmisión de datos con características de ejecución en tiempo real, como video y audio interactivo, en estos casos es muy común el uso en conjunto de RTP y RTSP.

El Tráfico generado por RTP se transporta en el protocolo UDP, esto es debido a que las aplicaciones que usan RTP por lo general son menos sensibles a la pérdida de paquetes, pero muy sensibles a los retardos; por eso se prefiere UDP sobre TCP. La figura 6 muestra el encabezado del protocolo RTP.

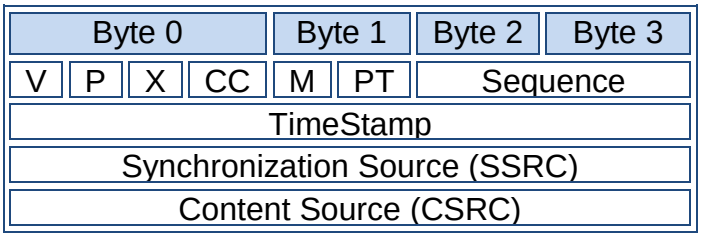


Figura 6. *Encabezado del protocolo RTP*

Fuente: Wikipedia: RTP Protocol.

Campos del encabezado del protocolo RTP [6].

- **Versión**: Versión del protocolo.
- **Padding**: si el bit P es puesto, se indica que el paquete se ha rellenado a un múltiplo de 4 bytes. El último byte indica el número de bytes de relleno.
- **Extensión**: Este bit X indica que hay un encabezado de extensión.
- **Contributing Source Count**: Indica cuantos orígenes de contribución (CSRC) están presentes, si este bit es cero, entonces la fuente de sincronización es seguida por la carga útil.
- **Marker**: Es un bit de marcador específico de la aplicación.
- **PayLoad Type**: Indica cual algoritmo de codificación se ha utilizado para el formato de carga útil. Los tipos de carga útil se especifican en el RFC 3551.
- **Séquence Number**: Contador que se incrementa en cada paquete enviado. Se utiliza para detectar paquetes perdidos.
- **TimeStamp**: El origen del flujo produce la marca del tiempo para indicar cuándo se creó la primera muestra en el paquete.
- **Synchronisation Source**: Indica a cual flujo pertenece el paquete. Es el método utilizado para multiplexar y desmultiplexar varios flujos de datos en un solo flujo de paquetes UDP.
- **Content Source**: En caso de que hayan, se utilizan cuando los mezcladores están presentes en el estudio.

La implementación del protocolo RTP tiene como protocolo de transporte subyacente a UDP que a su vez trabaja sobre IP. La Figura 7 muestra los diferentes niveles usados para la implementación del protocolo RTP.

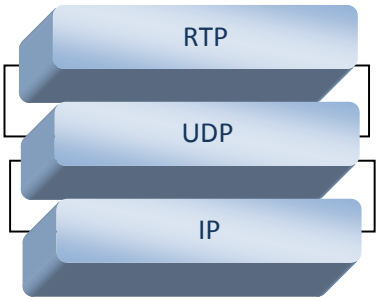


Figura 7. *Lista protocolos para RTP*

Fuente: Autores. Basada en el modelo TCP/IP.

2.4.2 Protocolo RTSP

El protocolo RTSP (Real Time Streaming Protocol) [4] es usado para transmitir contenido multimedia en tiempo real y permite hacer un control directo en el cliente de la reproducción del flujo de datos enviado desde el servidor por medio de métodos propios del protocolo. La mayoría de los servicios basados en RTSP hacen uso del protocolo RTP para la transmisión del contenido.

El protocolo RTSP hereda el diseño de HTTP/1.1, pero difiere de este en algunos aspectos como[10]:

- Un servidor RTSP debe mantener el estado en la mayoría de los casos, al contrario de lo que hace HTTP.
- Tanto el cliente como el servidor RTSP pueden realizar peticiones.
- Los datos son transportados por un protocolo diferente.
- La URI de petición RTSP siempre contiene el URI absoluto.

2.4.2.1 Métodos RTPS

El protocolo RTPS define una serie de métodos que permiten el establecimiento y control de una sesión en la transmisión de contenido multimedia. Los métodos más representativos del protocolo se muestran en la siguiente figura 8 con una

[10] BARCELO ORDINAS, José M. Protocolos y Aplicaciones Internet. Barcelona: Editorial UOC, 2008. Pág. 196.

breve descripción, en el capítulo 3 se describe de una manera más detallada estos métodos.

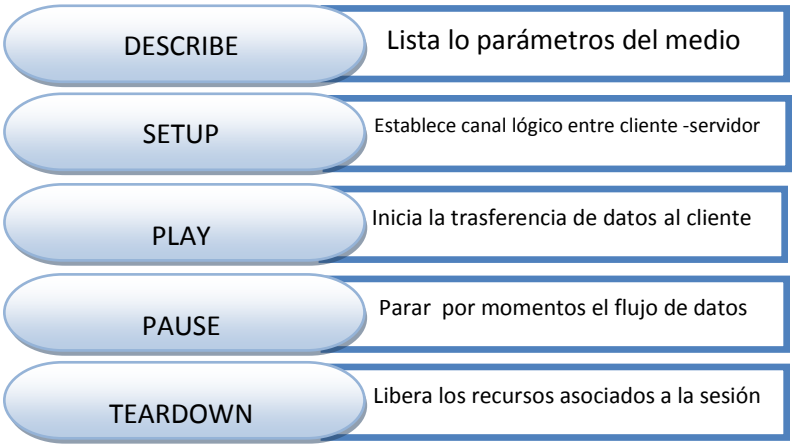


Figura 8. *Métodos Principales del Protocolo RTSP*

2.4.2.2 Mensajes RTSP

Los principales mensajes que componen el protocolo RTSP se pueden dividir en peticiones y respuestas. Las peticiones son de tipo textual y son siempre contestadas por un mensaje de tipo respuesta basado en un código (ver Anexo A).

Un mensaje de petición RTSP está compuesto por tres partes: la línea de petición, la cabecera de petición y el cuerpo del mensaje. Un mensaje de respuesta RTSP está conformado también por tres partes: línea de estado de la respuesta, cabecera de la respuesta y el cuerpo del mensaje. La figura 9 muestra el formato de un mensaje de RTSP.

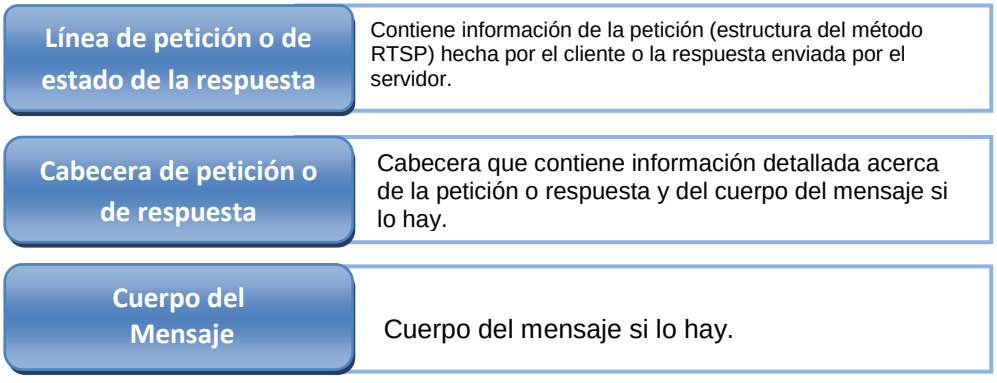


Figura 9. *Formato de un mensaje RTSP.*

Fuente: Autores. Basada en RFC 2326. *Real Time Streaming Protocol* [4].

2.5 Servidores Streaming

Existen dos maneras de implementar un servicio de streaming, los cuales se caracterizan por utilizar una de las siguientes opciones:

- Servidores Web : Implementando los protocolos HTTP y TCP
- Servidores de Streaming: Basado en protocolos TCP, UDP y protocolos específicos de streaming.

Un servicio de streaming usando servidores web tiene la ventaja de poder operar en una infraestructura web, sin embargo, servidores especializados en streaming son mucho más eficientes y ofrecen importantes prestaciones como las que se listan a continuación:

- Enviar un archivo de mayor o menor calidad dependiendo de la velocidad de la línea.
- Debido a que trabaja sobre UDP la velocidad de transmisión es más eficiente. De hecho, la naturaleza de streaming no requiere la recuperación de paquetes perdidos como en TCP. Esta recuperación es innecesaria y degrada el rendimiento.
- Ofrece servicios avanzados en el control de flujo (reproducir, pausar, etc.).
- Se acepta la multidifusión para soportar muchos clientes en un único flujo: de esta manera la capacidad de una máquina es mucho mayor que un servidor web tradicional.

Actualmente están disponibles en el mercado varios servidores para la implementación de servicios de streaming, pero para tomar una decisión y encontrar el más adecuado para una necesidad específica es necesario considerar lo siguiente:

- Soporte de códecs de audio y vídeo, incluyendo en este caso, utilizados para el streaming en el dispositivo móvil.
- Soporte para protocolos de streaming.
- El rendimiento y la capacidad de los servidores.
- La posibilidad de decodificar dinámicamente el contenido multimedia.
- Un soporte multiplataforma.

2.6 Herramientas para el Desarrollo de Aplicaciones: J2ME

La plataforma escogida para el desarrollo de la aplicación es J2ME, que es un lenguaje portable ya que la mayor parte de teléfonos móviles soportan aplicaciones JAVA.

2.6.1 Java

Java [7] es un moderno y potente lenguaje de programación desarrollado por la compañía Sun Microsystems[11]. Tiene características como:

- Lenguaje simple: Java posee una curva de aprendizaje muy rápida.
- Orientado a objetos.
- Distribuido: Java proporciona una colección de clases para su uso en aplicaciones de red, permitiendo establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.
- Interpretado y compilado a la vez.
- Robusto, diseñado para crear software altamente fiable.
- Indiferente a la arquitectura.
- Portable.
- Alto rendimiento.

Todas estas características han hecho de Java uno de los lenguajes con mayor crecimiento de uso en distintos ámbitos de la industria de la informática como lo son en dispositivos móviles y sistemas empotrados[12].

2.6.2 J2ME

La plataforma Java 2 Micro Edition (J2ME) [8], está orientada a proveer una colección certificada de APIs de desarrollo de software para dispositivos electrónicos con capacidades computacionales limitadas como teléfonos móviles, PDAs o electrodomésticos inteligentes.

"Las tecnologías J2ME contienen un JRE altamente optimizado, especialmente desarrollado para el mercado de gran consumo, abarcan una amplia gama de aparatos de tamaño muy reducido y permiten ejecutar programas de seguridad, conectividad y utilidades en tarjetas inteligentes, buscapersonas, sintonizadores de TV y otros pequeños electrodomésticos. Las tecnologías J2ME representan únicamente una parte de la gama de productos de software de Java. Las plataformas Java relacionadas son la Plataforma Java 2, Edición estándar (plataforma J2SE) y la Plataforma Java 2, Edición empresa (plataforma J2EE). La tecnología Java ofrece, asimismo, métodos de creación de servicios Web, transferencia de información XML, numerosos protocolos de red, kits de herramientas y la aplicación Java Web Start"[13].

[11] Sun Microsystems. http://www.oracle.com/us/sun/index.html. 22 de Junio de 2009.
[12] Wikipedia. Sistemas Empotrados: http://es.wikipedia.org/
[13] http://www.java.com/es/download/faq/whatis_j2me.xml. 17 de Noviembre de 2009.

Dentro de los paquetes opcionales para J2ME se encuentra el Mobile Media API (MMAP). MMAPI es una librería que proporciona funcionalidades multimedia en cualquier dispositivo que soporte java. La figura 10 presenta la arquitectura de la plataforma J2ME incluyendo el paquete opcional MMAPI.

2.6.2.1 MMAPI

MMAPI [9] es uno de los paquetes opcionales para J2ME que permite a los desarrolladores de aplicaciones móviles, el acceso a las capacidades multimedia del dispositivo siempre que este tenga soporte JAVA. El uso de este paquete permite incluir en aplicaciones móviles la reproducción y captura de diferentes formatos de archivos de audio y video, toma de fotografías a través de la cámara del dispositivo, entre otras utilidades.

MMAPI no es el único paquete opcional disponible para el desarrollo de aplicaciones para la plataforma J2ME. Otros paquetes incluyen los Servicios Web de la API (JSR 172), Mobile 3D Graphics API (JSR 184), y la API de ubicación (JSR 179).

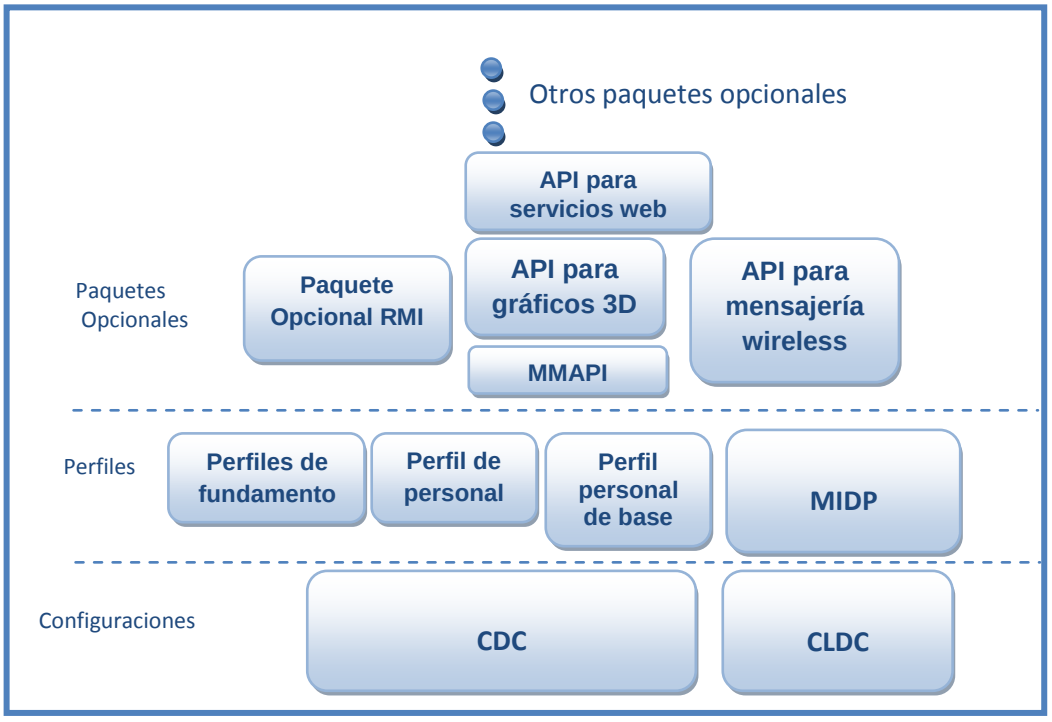


Figura 10. *Arquitectura J2ME*

Fuente: *Pro Java ME MMAPI: Mobile Media API for Java Micro Edition.*

2.6.2.2 Arquitectura MMAPI

El procesamiento de datos multimedia se divide en dos partes:

- Obtener el contenido multimedia: Lectura de los datos multimedia desde una fuente como un archivo o un servidor.
- Mostrar o reproducir este contenido: Análisis o decodificación de los datos multimedia para la visualización en el dispositivo.

Para realizar las operaciones antes mencionadas el MMAPI proporciona tres tipos de objetos de alto nivel: Un objeto DataSource, un objeto Player y un objeto Manager. A continuación se muestra la responsabilidad de casa una de estas clases.

Clase Datasource: Encapsula el protocolo de manipulación al ocultar los detalles de cómo se leen los datos desde su fuente. Puede que esta clase no se utilice directamente, a menos que se desee crear un DataSource para un protocolo específico. Cada DataSource se compone de uno o varios stream que son objetos de la clase SourceStream. Un SourceStream se utiliza para abstraer un solo flujo de datos multimedia.

Interface Player: La Interface Player es la encargada de manipular los datos multimedia que fueron encapsulados por la clase DataSource. Esta Interface provee métodos para el control y manipulación del contenido extraído del objeto DataSource durante su reproducción. La interfaz Player consiste en un ciclo de vida para los reproductores, de esta manera el programador será capaz de tener cierto control sobre una serie de operaciones que son susceptibles de consumir gran cantidad de recursos. El ciclo de vida de un objeto Player(ver Figura 11) consta de cinco estados: UNREALIZED, REALIZED, PREFETCHED, STARTED, CLOSED.

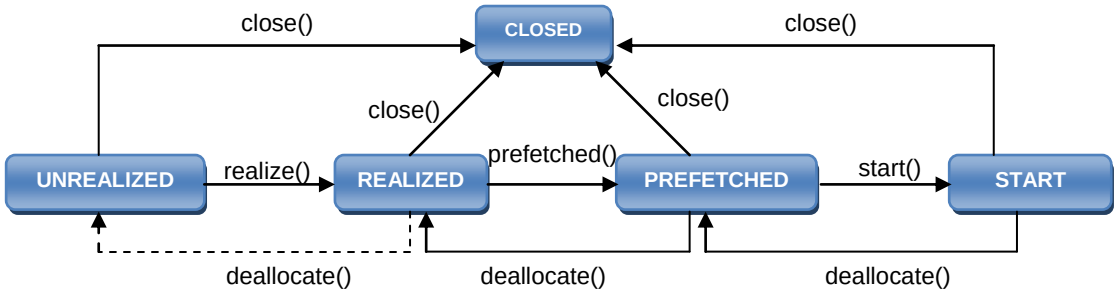


Figura 11. *Ciclo de vida de un objeto Player.*

Fuente: *Pro Java ME MMAPI: Mobile Media API for Java Micro Edition.*

Clase Manager: La clase Manager sirve como puente entre la interface Player y la clase DataSource, esta crea los accesos a la instancia Player y DataSource, como lo muestra la figura 12.

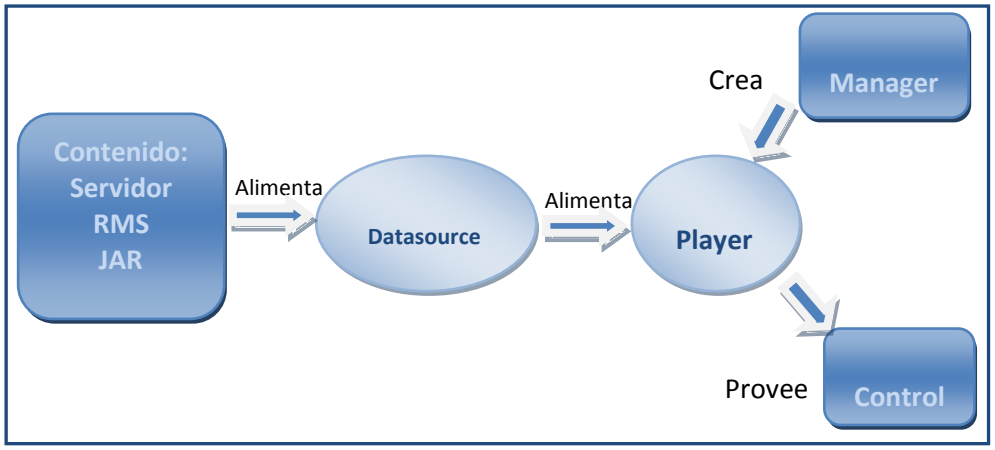


Figura 12. *Arquitectura MMAPI*

Fuente: The J2ME Mobile Media API.
Disponible en: http://developers.sun.com/mobility/midp/articles/mmapioverview/

CAPITULO III

3. EL SERVIDOR DE STREAMING

En este capítulo se presentan las dos alternativas de servidores que se tuvieron en cuenta para la elaboración del proyecto con base a criterios definidos; explicando sus principales características. Posteriormente se describe el proceso de funcionamiento de los servidores cuando se hace la transmisión de contenido al cliente y finalmente se exponen las razones para la escogencia del servidor.

3.1 Criterios de Selección

Los criterios técnicos para la escogencia de una herramienta se pueden dar desde diferentes perspectivas. Una de ellas es la de realizar un estudio detallado de las principales características y valorarlas de acuerdo a una necesidad especifica. Para este caso particular, la elección del servidor streaming se realizó teniendo en cuenta varios criterios que se acoplaban a los objetivos planteados en el proyecto y así satisfacer el tipo de usuarios específicos, en este caso los usuarios de dispositivos celulares móviles.

En la selección de servidores de audio streaming a evaluar se tuvieron en cuenta los siguientes aspectos:

- Usabilidad: software que sea estable y utilizado con frecuencia para la realización de este tipo de proyectos que involucra clientes móviles.

- Soporte Unicast: Todos los servidores de streaming tienen esta característica, la única diferencia está en las conexiones (número de usuarios) concurrentes que la licencia permita (número de usuarios).

- Soporte de formatos de audio: Cada servidor de streaming maneja unos formatos de audio y video para su trasmisión en el cliente.

- Documentación: Servidores que cuenten con una amplia y completa información sobre su funcionamiento interno, además de la instalación, configuración y puesta en marcha con el fin de facilitar y agilizar la curva de aprendizaje.

- Open Source: Servidores de código abierto, de fácil acceso y sin ningún costo.

Con base en los aspectos antes mencionados se hicieron las respectivas investigaciones para encontrar que servidores streaming cumplían con los

requisitos planteados. Los servidores encontrados fueron: Darwin Streaming Server y Hélix DNA Server.

3.2 Darwin Streaming Server (DSS)

El DSS [10] es un software desarrollado por la compañía Apple[14], es la versión de código abierto de la tecnología Apple QuickTime Streaming Server que permite enviar flujos (streams) de datos a los clientes a través de Internet utilizando los protocolos RTP Y RTSP. Esta versión proporciona un alto nivel de personalización y se ejecuta en diversas plataformas, permitiendo manipular el código para satisfacer sus necesidades. Fue el servidor escogido para el proyecto, la versión que se utilizó es la 5.5.5 instalada en un sistema operativo Linux (DEBIAN Lanny).

Las principales características de este servidor son las siguientes[15]:

- Es capaz de trabajar como servidor de archivos MP3, MP4, MOV y 3GP *hinted.*
- Permite creación de listas de reproducción.
- Permite utilizar un sistema de relays *(*retrasos) para obtener o servir flujos de/a otra fuente o servidor.
- No soporta adaptación de tasa en tiempo real aunque se anuncie de manera experimental mediante el intercambio de cabeceras del mismo tipo.
- Robusto: la versión número 5ª se comporta de manera muy estable en las pruebas realizadas en diferentes plataformas.
- Sencillo: No necesita ningún tipo de configuración adicional por parte del cliente. Se instala, ejecuta y se almacenan los archivos en las carpetas correspondientes.
- Soportado: Cuenta con un sistema de actualizaciones periódicas. Soporta los principales terminales, protocolos y tipos de archivos relacionados con PSS.
- Módulos programables: Presenta una estructura modular para implementar módulos externos que no afecten al funcionamiento del núcleo, la idea es permitir que estos módulos puedan ser compatibles con versiones posteriores.
- Soporta RTSP sobre TCP.
  RTP sobre UDP.
  RTP sobre Apple's Reliable UDP.
  RTSP/RTP en HTTP (tunneled).
  RTP sobre RTSP (RTP sobre TCP).

[14] APPLE (Online), 9 de Noviembre de 2009. http://www.apple.com/
[15] Darwin Streaming Server. Administrator's Guide (Online)
http://manuals.info.apple.com/en/QuickTimeStreamingSrvrAdminGuide.pdf

- Posee una interfaz gráfica vía web para configuración y monitorización de las principales características aunque pueden ser gestionadas de manera manual.

La Figura 13 muestra la interfaz de administración Web del DSS.

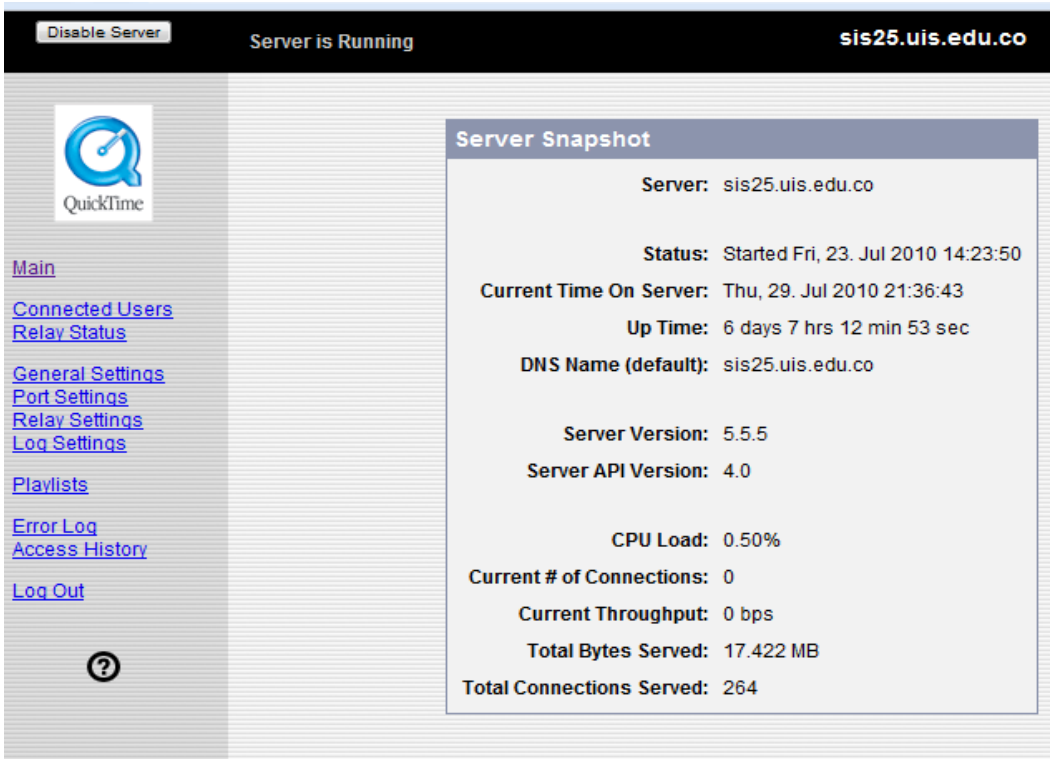


Figura 13. *Interfaz Web del Servidor DSS*

Main: Contiene la información básica del servidor, como el nombre, el estado, la versión instalada, numero reciente de conexiones entre otros.
Connected Users: Muestra los usuarios que en un momento determinado estén conectados al servidor.
Configuraciones: Directorio archivos, puerto RTSP, número máximo de conexiones simultáneas soportadas, throughput máximo soportado, método de autenticación, control de usuarios, preferencias en la generación de logs, creación de listas de reproducción.
Logs: Monitorización de actividad reciente, errores, acceso a archivos.

El DSS Contiene varios archivos de configuración que pueden ser modificados por el usuario, permitiendo adaptar valores de buffers, tasas máximas de transmisión, listas de acceso etc.

3.3 Hélix DNA Server

El servidor Hélix [11])es un software desarrollado por la compañía RealNetwork[16] que da solución a la distribución de audio y video a través de streaming tanto en PC's como en dispositivos móviles. Hélix funciona en diversas plataformas incluyendo Linux RHEL 4.0, Windows 2003 y Sistema operativo Solaris. Helix DNA Server es una versión gratuita de Hélix Server disponible en su página de proyecto. Se distribuye bajo General Public License. La versión utilizada para las pruebas fue la de Linux 11.1. Sus principales características son:

- Servidor open source.
- Soporta RTP, RTSP, SDP.
- No soporta archivos 3gp. Únicamente .rm .ra. rv. mp3 de manera nativa.
- Para utilizarlo con otros formatos sería necesario implementar un paquetizador para cada cada uno de ellos.
- Tiene una interfaz grafica de usuario para monitorizar y configurar el servidor.

La Figura 14 muestra la interfaz de administración Web del HELIX SERVER

Figura 14. *Interfaz Web del Servidor Hélix.*

[16] REALNETWORKS (Online), 9 de Noviembre de 2009. http://www.realnetworks.com/

3.4 Pruebas para los Servidores

Las pruebas fueron realizadas en el cuarto piso del edificio CENTIC de la Universidad industrial de Santander con el fin de de observar si cumplían con los criterios antes mencionados. Los recursos que se utilizaron fueron:

Recursos de HARDWARE:

- **Servidor**: Se utilizaron dos computadores de escritorio en los cuales se instaló cada uno de los servidores escogidos: Hélix DNA Server y Darwin Streaming Server para la distribución a los clientes del contenido multimedia almacenado en los equipos por medio del servicio de streaming (Ver tabla 1.)

CARACTERÍSTICAS DE LOS COMPUTADORES	
Sistema Operativo:	DEBIAN LENNY GNU/Linux 5.0.5
Disco Duro:	
RAM:	2 GB
Procesador:	PENTIUM CPU 4 3,20 GHz y 3,19 GHz
CARACTERÍSTICAS DE LA RED	
Tarjeta de Red:	FAST ETHERNET 100Mbps

Tabla 1. Características del computador utilizado para pruebas de servidores streaming.

- **Cliente**: Equipo portátil bajo prueba, sobre el cual se ejecuta el reproductor de contenido streaming multimedia, además de la herramienta de trafico de red (Ver tabla 2.)

CARACTERÍSTICAS DE LOS COMPUTADORES	
Sistema Operativo:	WINDOWS XP Service P ack 2
Disco Duro:	160 GB HDD
RAM:	2 GB
Procesador:	Intel® Pentium Core DUO T2080 1.73 GHZ 1MB L2 Cache
CARACTERÍSTICAS DE LA RED	
Tarjeta de Red:	Intel® PRO/Wireless 3945ABG network connection (dual-band tri-mode 802.11a/b/g) Wi-Fi CERTIFIED™

Tabla 2. Características del computador cliente utilizado para pruebas.

Recursos de SOFTWARE:

- Sistema Operativo Debian Lenny 5.0 instalado en los Computadores.
- Hélix DNA Server en su versión básica para Linux.
- Darwin Streaming Server 5.5.5 para Linux.
- Reproductores QuickTime[17] y RealPlayer[18] Para efectos de las pruebas.
- herramienta de tráfico de red Wireshark[19].

3.4.1 Pruebas

Después de la escogencia de los servidores con los cuales se trabajaría, se instaló cada uno de ellos en un equipo y se entró en la etapa de familiarización con el fin de conocer a profundidad cada una de sus características, su administración y configuración.

Luego se realizaron pruebas de funcionamiento con los dos servidores: Hélix y Darwin, utilizando un archivo para estas pruebas con formato .mov: RealQT.mov para el Darwin y un archivo con formato .rm: RealVideo8.rm. Fue necesario realizar las pruebas con archivos en formatos diferentes debido a los formatos que soportan los servidores. Cada uno de estos archivos fue almacenado en el servidor correspondiente para ser distribuido por ellos y escuchados en el cliente con el reproductor Quicktime y RealPlayer.

Para la transmisión de streaming con los servidores ofrecidos por la compañía Apple como el Darwin Streaming Server o su versión paga el Quicktime Streaming Server el archivo a distribuir se debe someter a un proceso denominado “hinting” antes de ser almacenado en el servidor. Esta técnica hace que los archivos multimedia contengan información sobre cómo se debe realizar el streaming por parte del servidor.

3.4.1.1 Pruebas con el Servidor Darwin

La tabla 3 muestra las propiedades del archivo que fue transmitido desde el servidor DSS hacia el cliente.

[17] QUICKTIME (Online), 5 de Noviembre de 2009. http://www.apple.com/es/quicktime/
[18] REALPLYAER (Online), 5 de Noviembre de 2009. http://www.real.com/player/realplayer_intl.html?lang=es
[19] WHIRESHARK (Online), 7 de Noviembre de 2009. http://www.wireshark.org/

Pruebas desde el servidor darwin	
Tamaño de la resolución	160x120
Tiempo de duración	51 seg
Tamaño del archivo	1209.09375 KB
Tasa promedio de transferencia	32,87 kB/s

Tabla 3. Prueba de Streaming en Darwin Streaming Server

En las figuras 15 y 16 se incluyen las descripciones de los puertos usados en esta sesión. Estos datos fueron lecturas directas de la herramienta de tráfico de red Wireshark.

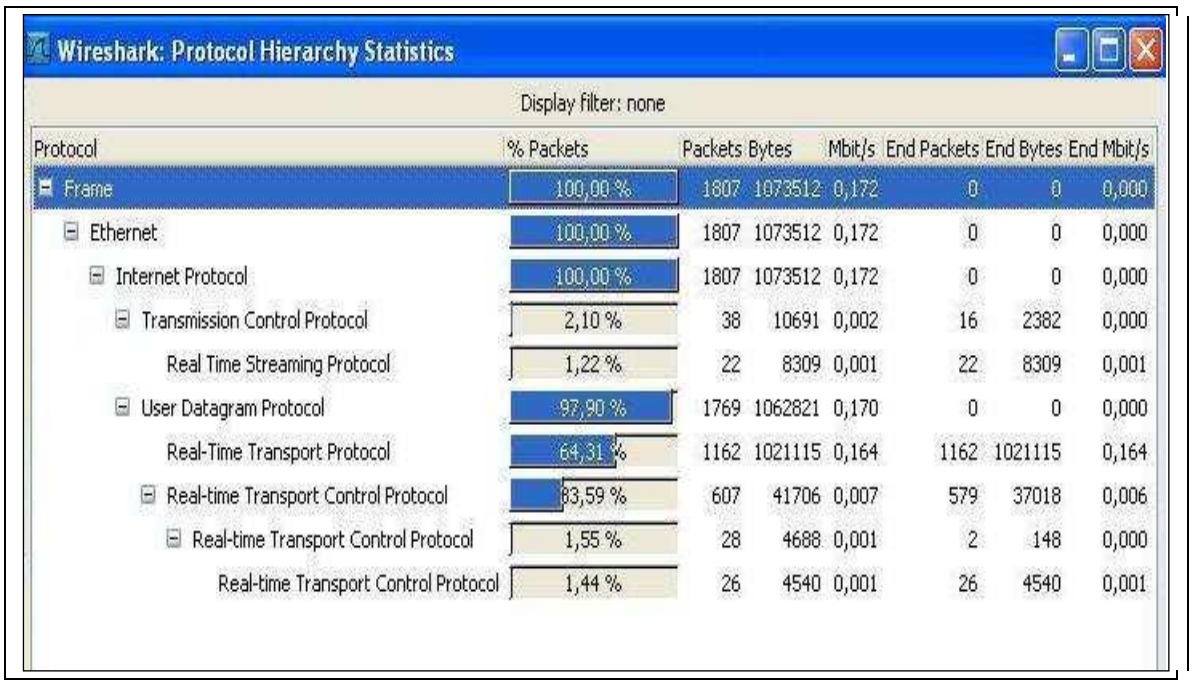


Figura 15. *Estadística del Trafico de Paquetes con Wireshark para el servidor DSS*

Fuente: Autores. Obtenida de Wireshark.

En la Figura 15 se puede observar el uso del protocolo RTSP dividiendo el tráfico de datos usando el protocolo UDP y el control de datos usando el protocolo TCP. Lo que interpreta que con esto se logra minimizar el desperdicio en el que incurriría si se usara TCP para la trasmisión de datos por streaming. Además, se comprueba que la técnica de streaming se implementa sobre el protocolo UDP, en el que no se hace retransmisiones debidas a las pérdidas de paquetes, por lo que la disminución en el rendimiento debido a retransmisiones no se presenta, situación que suele ocurrir cuando se usa el protocolo TCP donde la latencia se hace notable.

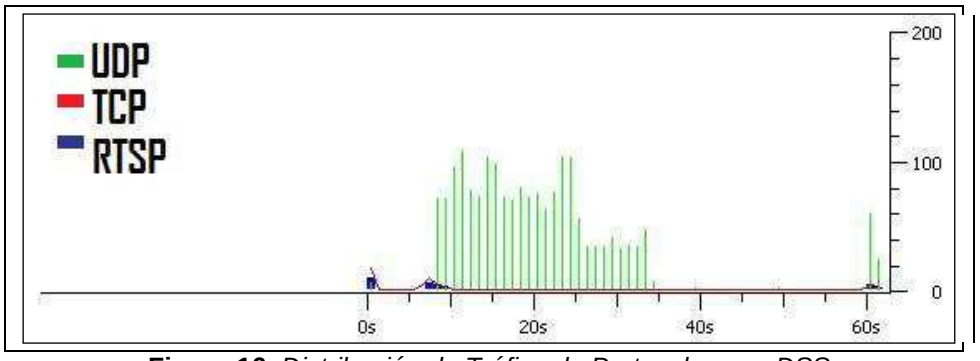


Figura 16. *Distribución de Tráfico de Protocolos con DSS*

Fuente: Autores. Obtenida de Wireshark.

En la figura 16 se muestra la distribución de tráficos de los protocolos UDP, TCP Y RTSP en un intervalo de tiempo de 60 segundos, tiempo que demoro en reproducir el archivo en el cliente. En esta gráfica el eje horizontal se entiende como el tiempo trascurrido en la sección y el eje vertical el número de paquetes recibidos.

3.4.1.2 Pruebas con el Servidor Hélix DNA Server

La tabla 4 muestra las propiedades de los archivos que se trasmitieron desde el servidor hasta el cliente.

Pruebas desde el servidor Hélix	
Tamaño de la resolución	160x120
Tiempo de duración	51 seg
Tamaño del archivo	753.37012 KB
Tasa promedio de transferencia	27,56 KB/s

Tabla 4. Prueba de streaming en el servidor Hélix Server

Para las pruebas con el servidor Hélix fue necesario cambiar el tipo de formato debido a que la versión gratuita, la cual posee una licencia básica maneja solo cierto tipos de formatos además que solo permite la conexión simultánea de 5 clientes a través de archivos codificados mediante Real Media. Otras licencias de pago permiten trabajar con transmisiones multicast y otros formatos de vídeo. En las figuras 17 y 18 se incluyen las descripciones de los puertos usados en esta sesión. Estos datos fueron lecturas directas de la herramienta de tráfico de red Wireshark.

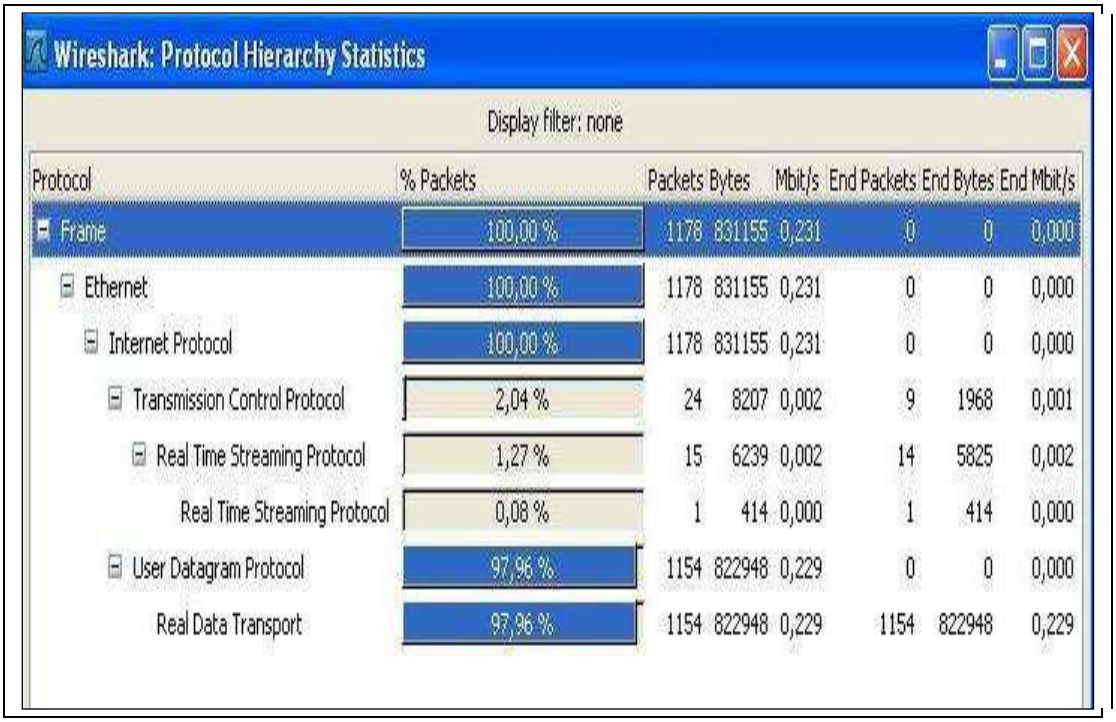


Figura 17. *Estadística del Trafico de Paquetes con Wireshark para el servidor Hélix Server*

Fuente: Autores. Obtenida de Wireshark.

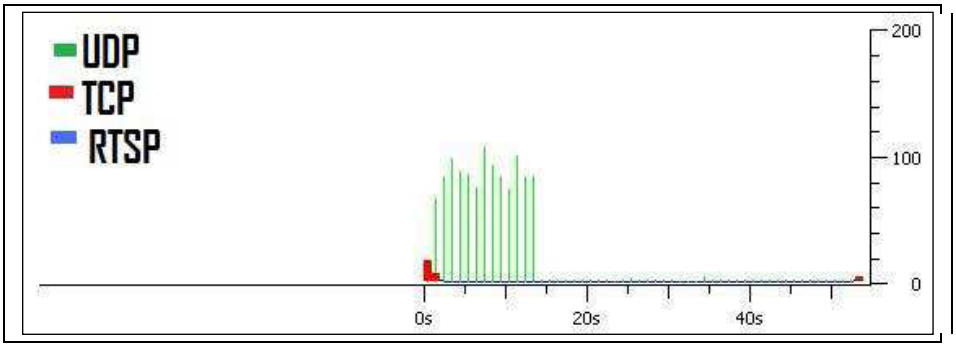


Figura 18. *Distribución de Tráfico de Protocolos con Hélix Server*

Con las pruebas realizadas anteriormente y discutiendo las principales características que debe tener el servidor streaming, se comparó los formatos que soportaba cada servidor en la trasmisión de audio hacia los clientes, la curva de aprendizaje para su administración, manejo de cada uno de los módulos que lo integraban y en las características de arquitectura de cada servidor. Debido a que el servidor Hélix en su versión open source no da soporte a formatos compatibles con clientes móviles se tomó la decisión de utilizar como único servidor para continuar con el proyecto el Darwin Streaming Server DSS.

3.5 Funcionamiento del Servidor Streaming

El DSS utiliza el protocolo RTSP para la comunicación con el cliente. Para que éste pueda establecer una sesión con el servidor se hace necesario el uso de los métodos que provee el protocolo RTSP. Estos métodos se describen a continuación:

La figura 19. Muestra un resumen de una sesión streaming usando RTSP.

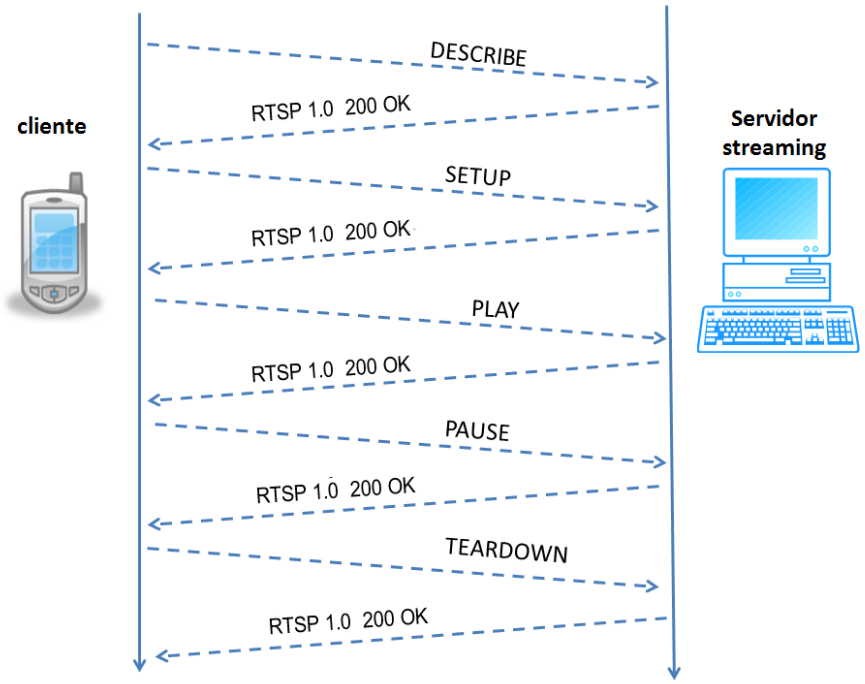


Figura 19.*Uso de los Métodos RTSP en una sesión streaming*

DESCRIBE: Fase inicial en la comunicación que obtiene la descripción del archivo solicitado. Es un método obligatorio debido a que los métodos subsiguientes dependen de él. A continuación se muestra una petición DESCRIBE y la respuesta del servidor a esta petición obtenidas de durante las pruebas.

```
====== PETICION CLIENTE ======
       DESCRIBE rtsp://sis25.uis.edu.co:554/21_questions.3gp rtsp/1.0
       CSeq: 1
============================

====== RESPUESTA SERVIDOR ======
       RTSP/1.0 200 OK
       Server:    DSS/5.5.5    (Build/489.16;    Platform/Linux;    Release/Darwin;
       state/beta; )
       Cseq: 1
       Last-Modified: Mon, 05 Jul 2010 15:48:42 GMT
```

```
       Cache-Control: must-revalidate
       Content-length: 334
       Date: Fri, 23 Jul 2010 14:28:33 GMT
       Expires: Fri, 23 Jul 2010 14:28:33 GMT
       Content-Type: application/sdp
       x-Accept-Retransmit: our-retransmit
       x-Accept-Dynamic-Rate: 1
       Content-Base: rtsp://sis25.uis.edu.co:554/21_questions.3gp/
       v=0
       o=StreamingServer 3488884112 1278344922000 IN IP4 192.168.24.109
       s=/21_questions.3gp
       u=http:///
       e=admin@
       c=IN IP4 0.0.0.0
       b=AS:14
       t=0 0
       a=control:*
       a=maxprate:2.000000
       a=range:npt=0- 224.70000
       m=audio 0 RTP/AVP 96
       b=AS:14
       b=TIAS:13
       a=maxprate:2
       a=rtpmap:96 AMR/8000/1
       a=control:trackID=2
       a=fmtp:96 octet-align=1
============================
```

- SETUP: Este método cumple la función de hacer que el servidor reserve los recursos necesarios para comenzar la transmisión del flujo, además indica el protocolo a utilizar para esta trasferencia (UDP, TCP, HTTP).

```
====== PETICION CLIENTE ======
SETUP rtsp://sis25.uis.edu.co:554/21_questions.3gp/trackID=2 rtsp/1.0
CSeq: 2
TRANSPORT: UDP;unicast;client_port=9090-9091
============================

====== RESPUESTA SERVIDOR ======
       RTSP/1.0 200 OK
       Server:    DSS/5.5.5    (Build/489.16;    Platform/Linux;    Release/Darwin;
       state/beta; )
       Cseq: 2
       Last-Modified: Mon, 05 Jul 2010 15:48:42 GMT
       Cache-Control: must-revalidate
       Session: 7623836820559374007

       Date: Fri, 23 Jul 2010 14:28:33 GMT
       Expires: Fri, 23 Jul 2010 14:28:33 GMT
       Transport:UDP;unicast;source=192.168.24.109;client_port=9090-
       9091;server_port=6970-6971;ssrc=133118EC
============================
```

- PLAY: Este método informa el servidor que puede iniciar el envío de flujos de datos establecidos. Para realizar una solicitud de PLAY el cliente debe hacer primero una solicitud SETUP.

```
====== PETICION CLIENTE ======
       PLAY rtsp://sis25.uis.edu.co:554/21_questions.3gp rtsp/1.0
       CSeq: 3
       Session: 7623836820559374007
============================

====== SERVER RESPONSE ======
       RTSP/1.0 200 OK
       Server: DSS/5.5.5 (Build/489.16; Platform/Linux; Release/Darwin;
       state/beta; )
       Cseq: 3
       Session: 7623836820559374007
       Range: npt=0.00000-224.70000
       RTP-Info:
       url=rtsp://sis25.uis.edu.co:554/21_questions.3gp/trackID=2;seq=27333;rtptim
       e=161719988
============================
```

- PAUSE: Este método interrumpe temporalmente el envío de datos por parte del servidor, sin embargo no libera los recursos reservados para dicha sesión.

```
====== PETICION DEL CLIENTE ======
       PAUSE rtsp://sis25.uis.edu.co:554/21_questions.3gp rtsp/1.0
       CSeq: 4
       Session: 7623836820559374007
============================
====== RESPUESTA DEL SERVIDOR ======
       RTSP/1.0 200 OK
       Server:    DSS/5.5.5    (Build/489.16;    Platform/Linux;    Release/Darwin;
       state/beta; )
       Cseq: 4
       Session: 7623836820559374007
============================
```

- TEARDOWN: Método utilizado para detener la transmisión de un recurso determinado, se libera todos los recursos relativos a dicha comunicación, invalidando cualquier petición realizada sobre dicha sesión.

```
====== CLIENT REQUEST ======
TEARDOWN rtsp://sis25.uis.edu.co:554/21_questions.3gp rtsp/1.0
CSeq: 5
Session: 7623836820559374007
============================
====== SERVER RESPONSE ======
RTSP/1.0 200 OK
Server: DSS/5.5.5 (Build/489.16; Platform/Linux; Release/Darwin;
state/beta; )
Cseq: 5
Session: 7623836820559374007
Connection: Close
============================
```

CAPITULO IV

4. REDES CELULARES

Cada una de las redes móviles existentes tiene un ancho de banda limitado en comparación con las conexiones de banda ancha de una conexión cableada. Esta limitación en el ancho de banda disponible hace que la tasa de bits del flujo de streaming esté también limitada.

Dentro de las principales redes celulares se encuentra: GPRS[20] casi ideal y 3G[20] ideal para prestar un servicio de streaming en clientes móviles. En este capítulo se menciona las características principales de estas redes

4.1 GPRS

GPRS (*General Packet Radio Service*) [12] se desarrolló para permitir la transmisión de los datos en una red con paquetes basados en IP utilizando la infraestructura GSM.

El flujo a través de esta red permite la transferencia de datos del paquete con una tasa de datos teórica de alrededor de 171,2 Kbits/s (hasta 114 Kbits/s en la práctica). Gracias a su modo de transferencia en paquetes, las transmisiones de datos sólo usan la red cuando es necesario. Por lo tanto, el estándar GPRS permite que el usuario reciba facturas por volumen de datos en lugar de la duración de la conexión, lo que significa especialmente que el usuario puede permanecer conectado sin costo adicional.

Para el transporte de voz, el estándar GPRS emplea la arquitectura de red GSM y provee acceso a la red de datos (especialmente Internet) por medio del protocolo IP o del protocolo X.25.

GPRS admite características nuevas que no están disponibles en el estándar GSM y que se pueden clasificar en los siguientes tipos de servicios:

- Servicio de punto a punto (PTP): es la capacidad de conectarse en modo cliente-servidor a un equipo en una red IP.
- Servicio de punto a multipunto (PTMP): constituye la capacidad de enviar paquetes a un grupo de destinatarios (Multidifusión).
- Servicio de mensajes cortos (SMS).

[20] International Telecommunication Union. http://www.itu.int/en/pages/default.aspx

4.2 3G

La tecnología 3G (tecnología inalámbrica de tercera generación) [12] es el estándar actualmente más avanzado es telecomunicaciones inalámbricas y está definido por la unión internacional de telecomunicaciones.

Las ventajas de la tecnología 3G comparadas con la 1G Y 2G es el incremento en capacidad para el número de usuarios, es decir, en el mismo espectro con la misma red se le puede dar servicio a un mayor número de usuarios y a la vez dentro del buzón de servicios de transmisión de datos a alta velocidades. Esto quiere decir que la misma red puede continuar dando los servicios tradicionales de voz además de servicios novedosos como la transmisión de video, la transmisión de audio, descarga de música, envió de fotografías, aumentar calidades de fotografía, etc.

También pude prestar acceso de servicios de banda ancha comparables o superior con los que se tiene en la casa o oficina pero con velocidades de datos de hasta 384 Kbps, es casi siete veces más rápida que una conexión telefónica estándar desde el dispositivo inalámbrico en cualquier lugar donde exista cobertura 3g.

4.3 Arquitectura de Red del Sistema

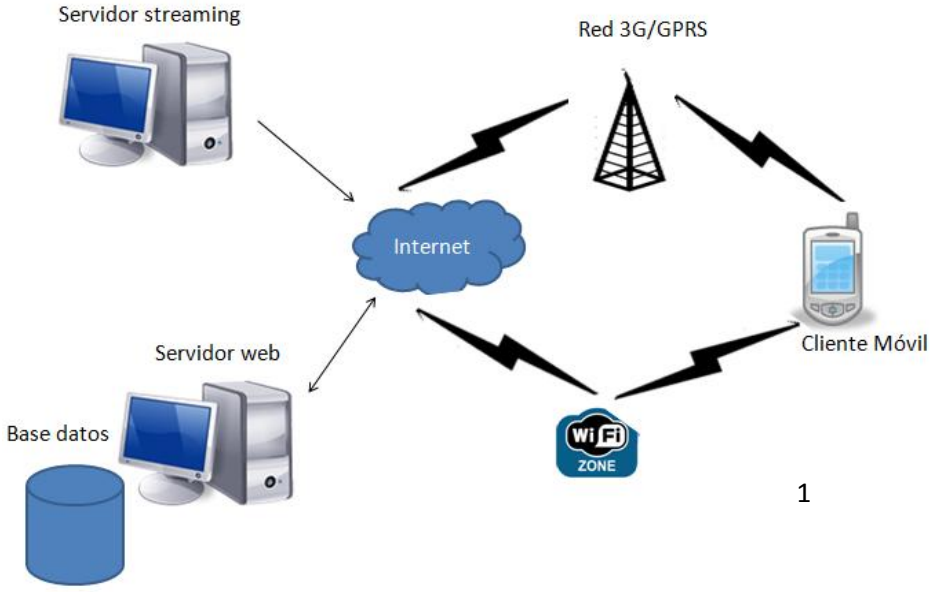


Figura 20. *Arquitectura de Red del Sistema Streaming*

Cuando un dispositivo celular (cliente en el sistema) desea descargar un contenido multimedia, en este caso un archivo de audio desde el servidor streaming por medio de la aplicación J2ME, este debe conectarse a internet a través de una red celular (3G o GPRS) o a través de una conexión inalámbrica (WLAN). La figura 20 ilustra la arquitectura de red del sistema.

El procedimiento de solicitud de un contenido multimedia se lista a continuación:

1. La aplicación se conecta primero al servidor web que accede a la base de datos, obteniendo información del archivo de audio solicitado y la ubicación de éste en el servidor de streaming.

2. Obtenida la ubicación del archivo se establece la comunicación con el servidor streaming, esta comunicación se hace a través de los métodos que provee el protocolo de comunicación RTSP.

3. Si la comunicación es exitosa, el servidor empezará a enviar el archivo de audio en paquetes por la red hacia el cliente móvil, estos paquetes se envían haciendo uso del protocolo de transporte RTP.

4. Al recibir los paquetes la aplicación se encarga de desempaquetarlos y decodificarlos para iniciar su reproducción.

CAPITULO V

5. METODOLOGIA Y REQUERIMIENTOS

5.1. Introducción

El uso de una metodología es importante en el desarrollo de un proyecto investigativo, esto, debido a que las ideas iniciales normalmente son vagas e imprecisas. Una metodología permite transformar estos primeros planteamientos para darle una forma más estructurada.

Extreme Programming [13] es un enfoque ágil para el desarrollo de software que hace hincapié en la satisfacción del cliente y el trabajo en equipo. La Programación Extrema es una metodología ligera de desarrollo de software que se basa en los siguientes aspectos importantes: comunicación, coraje, simplicidad y feedback.

Siguiendo esta metodología las actividades realizadas durante el desarrollo del proyecto se describen a continuación.

5.2 Actividades

En primer lugar se identificó la problemática a la cual el proyecto daría solución, para dar lugar al planteamiento de los objetivos y la justificación del mismo.

Como segunda instancia se realizó una investigación de las bases teóricas sobre conceptos que enmarcan un servicio streaming multimedia permitiendo dar una visión más precisa de lo que se deseaba desarrollar, adquiriendo conocimientos que fueron fundamentales durante todo el desarrollo del proyecto. Una vez claros los conceptos importantes, se consultó sobre la arquitectura general de un sistema streaming multimedia identificando sus componentes principales y enfocando esta arquitectura a un sistema streaming para clientes móviles, definiendo así, las características mínimas de los componentes principales: el servidor streaming y el dispositivo cliente. Paralelamente se consultó a cerca de las herramientas y librerías utilizadas en la programación de aplicaciones móviles.

Con un conocimiento más formado acerca del tema, se optó por empezar con la instalación, implementación y pruebas del servidor streaming, planteando los criterios de selección para la escogencia de un servidor que permitiera la distribución de audio en dispositivos móviles; para ésto se realizó un estudio de dos de los servidores más utilizados para la difusión de contenido multimedia que actualmente existen en el mercado, verificando que cumplan con los criterios ya

planteados. Se procede a la instalación, configuración y puesta en marcha de los servidores seleccionados. Se realizaron pruebas con equipos portátiles monitoreando los servidores a través de un software para monitoreo de red. Se observó el ancho de banda consumido, velocidad de los datos por cada conexión a cada servidor y se visualizó los protocolos involucrados durante la transmisión y su comportamiento frente a las peticiones. Junto a la instalación del servidor streaming se instaló también el servidor web.

Una vez instalados los servidores del sistema, se procedio al diseño y desarrollo de la aplicación móvil. Inicialmente se empezó a desarrollar la comunicación con el servidor web para la obtención de la información alojada en la base datos con respecto a las consultas hechas por el usuario sobre las canciones en el dispositivo móvil, durante el desarrollo se tuvo en cuenta la interfaz gráfica de la aplicación para que esta fuera agradable y fácil de usar.

Terminada esta primera fase se siguió con la programación del objetivo principal de este proyecto que era lograr realizar la trasferencia de datos de audio desde el servidor de streaming Darwin con los protocolos propios del streaming: RTSP/RTP hacia el dispositivo móvil. Para ésto se hizo la codificación de los archivos de audio en un formato escogido para el sistema y el estudio del envío de ese tipo de archivos dentro de paquetes RTP que son los paquetes enviados por el servidor streaming. Una vez hecho ésto se realizó la decodificación de los paquetes en la aplicación móvil para su posterior reproducción.

5.3 Requerimientos del Sistema

Los requerimientos necesarios del sistema considerados son los siguientes:

- RS 1: Toda la información del material digital de audio debe mantenerse en una base de datos.
- RS 2: El sistema debe funcionar sobre redes 3G y GPRS.
- RS 3: El sistema debe ser escalable.
- RS 4: El sistema debe utilizar herramientas open source.

5.4 Requerimientos de la Aplicación

Los requerimientos necesarios de la aplicación considerados son los siguientes:

- RA 1: La aplicación debe ser desarrollada en J2ME.
- RA 2: Soporta streaming de audio sobre los protocolos RTSP, SDP Y RTP.
- RA 3: La aplicación debe ser capaz de mostrar la información e imagen del álbum que se encuentra en la base de datos.

- RA 4: Un usuario que utilice la aplicación debe ser capaz de buscar archivos de audio alojados en el servidor por género, artista o todo el listado de canciones.
- RA 5: Un usuario que utilice la aplicación puede pausar y reanudar una canción.
- RA 6: L a aplicación debe contener un módulo de ayuda que oriente al usuario sobre las funcionalidades que este otorga.

5.5 Herramientas de Uso

Las siguientes herramientas fueron utilizadas durante el desarrollo del proyecto:

- Plataformas: Windows XP y Linux.
- Lenguajes de Programacion: Java (j2me), MySQL .
- Editor: Netbeans 6.7 y 6.8.
- Servidor streaming: Darwin streaming 5.5.
- Servidor web: Tomcat 6.
- Base de datos: MYSQL.
- Emuladores: Sun Java Wireless Toolkit 2.5.2, Java MicroEdition SDK 3.0, LG SDK 1.5, Samsung SDK 1.1.2 , Sony Ericsson SDK 2.5.0.6
- Teléfono celular: Nokia N95, Sony Ericsson W580i.
- Archivos de audio y video: MP3, 3GP, MOV, MP4.
- Programa para monitorear la red: Wireshark 1.3.

5.6 Casos de Uso

En el desarrollo de la aplicación para realizar streaming de audio se identificaron los siguientes requerimientos funcionales.

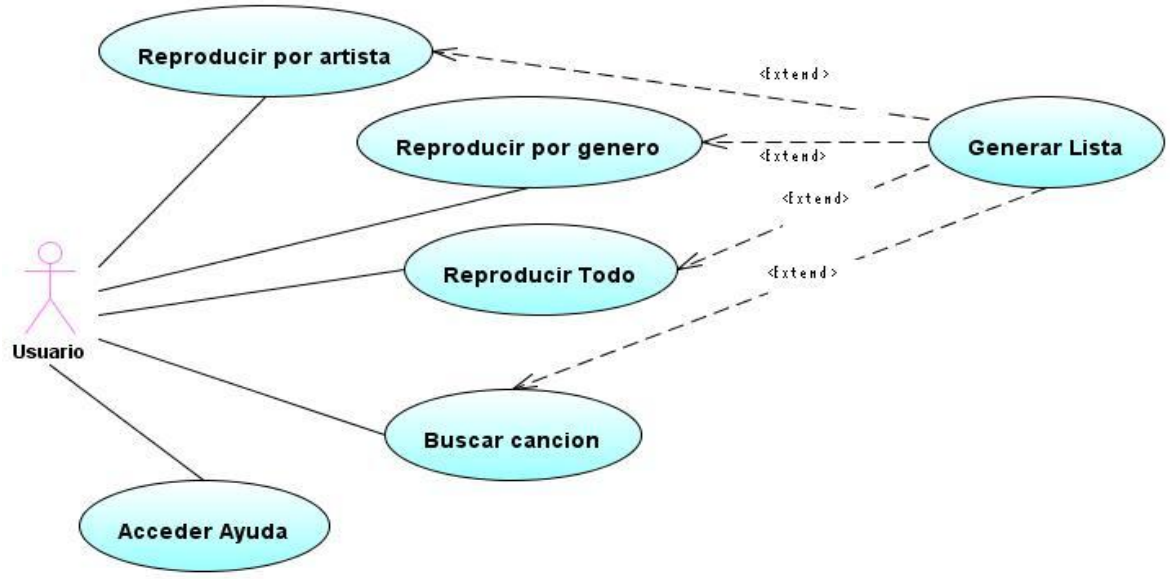


Figura 21. *Diagrama de Casos de Uso*

A continuación se describen los casos de usos presentes en el desarrollo de la aplicación:

Nombre:	Reproducir por artista
Descripción:	Reproduce una canción escogida por el usuario dependiendo del artista que este seleccionó previamente.
Eventos:	

ACTOR	SISTEMA
1.El usuario selecciona el artista y la canción del artista que desea escuchar.	
	2. El sistema realiza la petición al servidor.
	3. El sistema recibe respuesta directa del servidor.
	4. El sistema reproduce el archivo de audio.

Tabla 5. Caso de Uso 1

<table>
<tr><td>Nombre:</td><td>Reproducir por género</td></tr>
<tr><td>Descripción:</td><td>Reproduce una canción escogida por el usuario dependiendo del género que este seleccionó previamente.</td></tr>
<tr><td>Eventos:</td><td>
<table>
<tr><th>ACTOR</th><th>SISTEMA</th></tr>
<tr><td>1.El usuario selecciona el género y la canción que pertenece a este género que desea escuchar.</td><td></td></tr>
<tr><td></td><td>2. El sistema realiza la petición al servidor.</td></tr>
<tr><td></td><td>3. El sistema recibe respuesta directa del servidor.</td></tr>
<tr><td></td><td>4. El sistema reproduce el archivo de audio.</td></tr>
</table>
</td></tr>
</table>

Tabla 6. Caso de Uso 2

<table>
<tr><td>Nombre:</td><td>Reproducir todo</td></tr>
<tr><td>Descripción:</td><td>Reproduce una canción escogida por el usuario de la lista completa de canciones disponibles para la aplicación.</td></tr>
<tr><td>Eventos:</td><td>
<table>
<tr><th>ACTOR</th><th>SISTEMA</th></tr>
<tr><td>5 El usuario selecciona la canción que desea escuchar de la lista.</td><td></td></tr>
<tr><td></td><td>6El sistema realiza la petición al servidor.</td></tr>
<tr><td></td><td>7 El sistema recibe respuesta directa del servidor.</td></tr>
<tr><td></td><td>8 El sistema reproduce el archivo de audio.</td></tr>
</table>
</td></tr>
</table>

Tabla 7. Caso de Uso 3

Nombre:	Buscar canción
Descripción:	El usuario busca la canción que desea escuchar a través de un buscador rápido ofrecido por la aplicación.
Eventos:	

ACTOR	SISTEMA
1.El usuario busca la canción que desea escuchar por nombre de la misma.	
	2. El sistema realiza la petición al servidor.
	3. El sistema recibe respuesta directa del servidor.
	4. El sistema reproduce el archivo de audio.

Tabla 8. Caso Uso 4

Nombre:	Ayuda
Descripción:	Ofrece una pequeña documentación del manejo de la aplicación.
Eventos:	

ACTOR	SISTEMA
1.El usuario accede a la ayuda	
	2. El sistema muestra la ayuda al usuario

Tabla 9. Caso de Uso 5

.

5.7 Requerimientos de los Clientes Móviles

Para que un cliente móvil pueda utilizar el servicio de trasmisión de audio por la técnica de streaming de este proyecto de trabajo debe cumplir con los siguientes requisitos:

- Memoria de disco: 700 KB
- Versión java: JAVA-MIDP 2.0 y soporte para el paquete opcional MMAPI.
- Comunicación: HTTP, DATRAGRAMAS, SOCKET.
- Formato de audio: AMR.

5.8 Requerimientos hardware y software para servidores

Servidor web – Tomcat 6.x[21]

hardware	Software
• Procesador mínimo de 200 MHz. • 512 GB RAM • Mínimo 1 GB disco duro.	• JDK 1.5 -1.6 • MysqlConnector; si se trabaja con la base de datos Mysql. • Sistema operativo: Linux. Windows NT 2000, XP.

Tabla 10. Requisitos HW/SW para el Servidor Tomcat

Servidor Darwin Streaming Server[22]

hardware	Software
• Un mínimo de 512 MB de RAM. • Procesador mínimo de 500 MHz. • Mínimo 1GB de disco duro.	• Mac OS X. • Linux. • Windows NT, XP.

Tabla 11. Requisitos HW/SW para el Servidor DSS

[21] Apache Software Foundation. http://tomcat.apache.org/
[22] Darwin Streaming Server. Administrator's Guide (Online)
http://manuals.info.apple.com/en/QuickTimeStreamingSrvrAdminGuide.pdf

CAPITULO VI

6. DESCRIPCIÓN DE LA APLICACIÓN MOVIL STREAMJASSMOBILE

En este capítulo se muestra el proceso para el desarrollo del trabajo de grado objeto de este libro. Inicialmente se describe el códec de audio que se utilizó, luego se explica cómo fueron implementados los protocolos RTSP y RTP para J2ME y finalmente se muestra como resultado la aplicación móvil desarrollada y sus principales módulos.

6.1 Archivos de audio AMR.

Para la transferencia de audio digital por la red se hace indispensable que esté codificado y de esta forma evitar consumos excesivos de ancho de banda. Dentro del sistema streaming desarrollado el códec utilizado para la compresión es el Adaptive Multi-Rate (AMR), este códec permite una mejor tasa de compresión frente a otros formatos como MP3 y WAV con una pérdida de calidad baja. A demás el códec AMR fue escogido por la Third Generation Partnership Project (3GPP) como el códec obligatorio para los sistemas celulares de tercera generación [14].

En la figura 22. Se muestra una comparación de tamaños de un archivo de audio entre tres diferentes formatos.

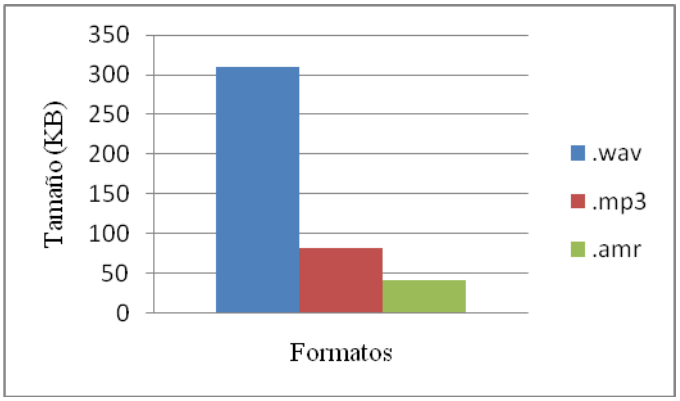


Figura 22. *Diferencia de tamaño entre archivos de audio WAV, MP3 y AMR*

El formato de audio AMR soporta 8 diferentes tasas de bits con velocidades entre 4.75 y 12.2 Kbps, la frecuencia de muestreo utilizada es de 8000 Hz y la codificación se realiza en 20 ms de speech por frame.

La estructura de un archivo de audio AMR está compuesta por una cabecera principal de 6 bytes, seguido de un número determinado de frames cada uno con su respectiva cabecera. La cabecera de cada frame es de 1 byte y el tamaño total del frame depende de la velocidad a la que fue codificado. La siguiente figura muestra la estructura de un archivo AMR.

Cabecera del Archivo
Cabecera del Frame 1
Frame 1
Cabecera del Frame 2
Frame 2
Cabecera del Frame 3
Frame 3
…
Cabecera del Frame N
Frame N

Figura 23. *Estructura de una Archivo AMR*

6.6.1 Hint Track

Los archivos multimedia que están destinados para el streaming enviados por el servidor deben pasar por un proceso llamado “hinting”, este proceso hace que el archivo se subdivida en tracks, estos tracks son enviados por el servidor en paquetes RTP. En síntesis el proceso de hinting da información al servidor de cómo enviar trozos del archivo hacia el cliente, indicando la longitud máxima en bytes y los milisegundos de audio por cada paquete a ser transmitido.

Los archivos de audio utilizados para las pruebas fueron archivos codificados bajo el formato de audio AMR a los cuales se le aplicó el proceso anteriormente mencionado, a través de una de las opciones que ofrece el reproductor Quicktime Pro 7[23]. Los archivos de audio fueron empaquetados a 10 frames cada uno con 20 ms de audio a una velocidad de transferencia de 12 Kbps.

[23] Apple QuickTime. http://www.apple.com/es/quicktime/download/

6.2 RTSP en J2ME

El lenguaje J2ME no tiene soporte para el manejo de este protocolo por lo que se hace necesario la implementación de los métodos descritos en el capítulo 3 en el lado del cliente y así realizar el establecimiento y manejo de una sesión RTSP con el DSS.

En primer lugar, la aplicación inicia la comunicación enviando una petición DESCRIBE al servidor. El propósito de este método es obtener información acerca de un recurso específico y de cómo puede ser adquirido. El servidor responde con un código: “200 OK” y ofrece una descripción en formato SDP del recurso de audio.

Hecho el método DESCRIBE el siguiente mensaje a enviar al servidor es el correspondiente al método SETUP que contiene la URL del archivo; el identificador asociado a éste, extraído de la respuesta al método DESCRIBE; el número de secuencia e indicaciones de que protocolo utilizar para enviar el flujo de datos (RTP, TCP, HTTP), el modo de transmisión (UNICAST o MULTICAST) y los números de los puertos del cliente a los cuales debe enviarse este flujo. Como respuesta a este método el servidor devuelve el identificador de la sesión establecida y los puertos por los cuales el DSS enviará el flujo de datos.

Una vez reservados los recursos mediante el método SETUP, el cliente envía un último mensaje para iniciar la transmisión, este mensaje corresponde al método PLAY cuyo contenido es la URL del archivo, el número de secuencia y el id de sesión extraído de la respuesta al método SETUP. Hecho esto el DSS empezará la entrega del contenido de audio. El procedimiento final es el de enviar el método TEARDOWN para liberar los recursos asociados y realizar posteriores peticiones.

6.3 RTP en J2ME

Debido a que J2ME no tiene soporte para el protocolo RTP también se hace necesario la implementación del mismo en la aplicación ya que el servidor streaming DSS utiliza este protocolo para el envío de la información a través de paquetes RTP.

6.3.1 Paquetes RTP/AMR a Archivos de audio AMR

Para el desarrollo del sistema se utilizó el formato de audio AMR. Como se explicó anteriormente el DSS utiliza como protocolo de transporte el protocolo RTP. Cuando el audio AMR es enviado dentro de la carga útil de un paquete RTP su estructura difiere de la de un archivo AMR original [15] (Ver figura 23), ya que la lista de cabeceras es seguida por los marcos (frames) de audio como se muestra en la figura 24. La estructura de la carga útil en un paquete RTP con audio AMR

se denomina paquete RTP-AMR y está formada además de de las cabeceras y los frames de audio, de una cabecera de 1 byte con información acerca del audio amr.

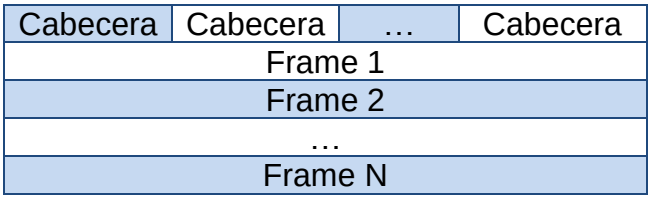


Figura 24. *Audio AMR como carta útil en un paquete RTP*

Para el manejo de los protocolos RTP enviados por el servidor la aplicación sigue un proceso:

1. *Abrir conexión para recibir los paquetes:*

   Cuando se solicita un archivo de audio al servidor, luego de la interacción con el mismo a través del protocolo RSTP la aplicación abre una conexión por Datagramas por el puerto especificado en el método SETUP del RTSP, que será el canal por el cual se recibirán los paquetes RTP. Esta conexión se abre una única vez por cada sesión RTSP establecida.

2. *Extracción de la carga útil*:

Al recibir el paquete RTP la aplicación lo desempaqueta, extrayendo la cabecera y obteniendo la carga útil del mismo que es en donde finalmente se encuentra el audio AMR.

3. *Construcción del archivo AMR*:

   Cuando el audio AMR es transmitido en paquetes RTP la estructura enviada en la carga útil del paquete no es un archivo AMR, por lo que la aplicación debe construir este archivo para ser reproducido posteriormente. Una vez extraído el paquete RTP-AMR (carga útil) se extrae la cabecera principal y se organizan las demás cabeceras y los frames de audio para construir el archivo AMR, además se agrega la cabecera propia de un archivo AMR obteniendo la estructura mostrada en la figura 25.

   Una vez construido el archivo AMR, estará listo para ser reproducido. La siguiente figura resume las iteraciones entre los protocolos RTSP y RTP.

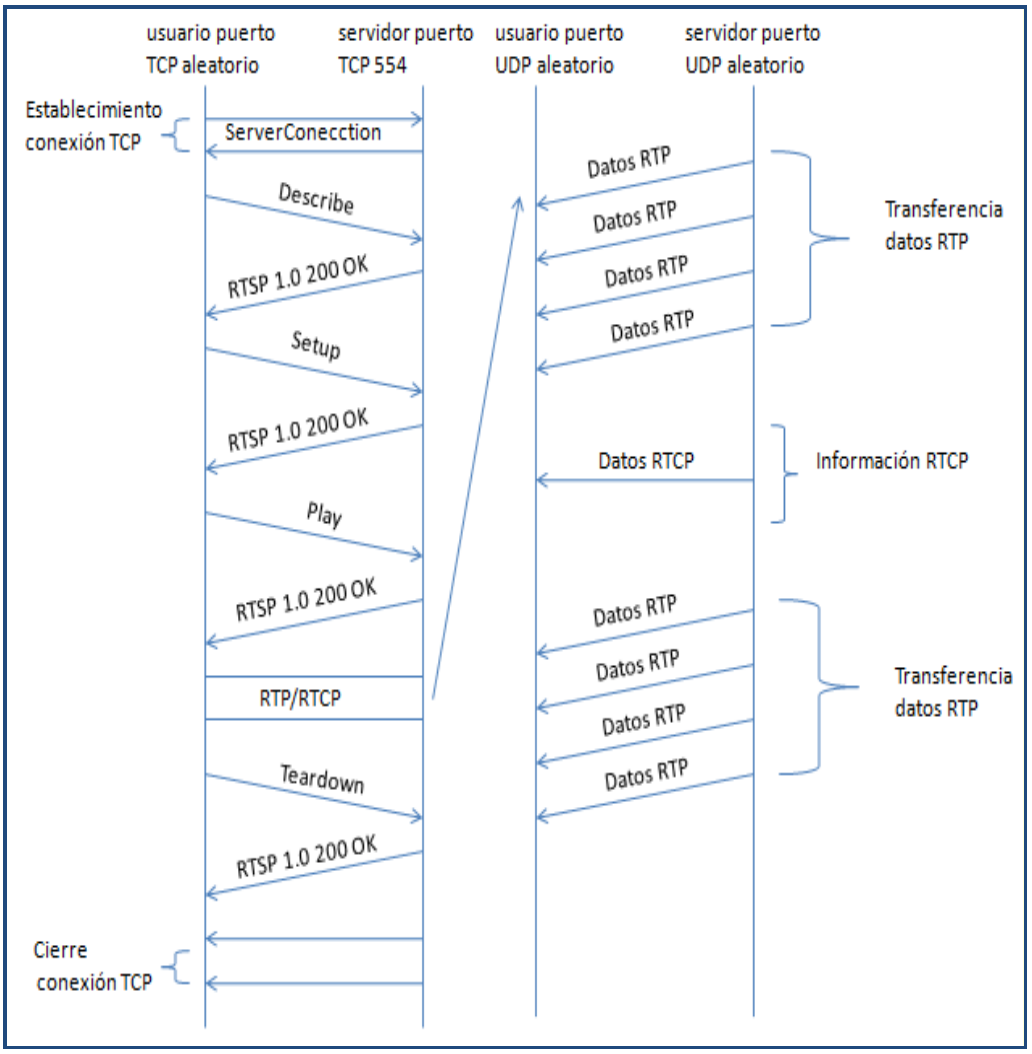


Figura 25. *Iteración de Protocolos RTSP y RTP*

Fuente: BARCELO ORDINAS, José M. Protocolos y Aplicaciones Internet. Barcelona: Editorial UOC, 2008. Pág. 204.

6.4 Reproducción de streaming de audio en J2ME

El lenguaje J2ME no tiene soporte playback para streaming, es decir, el PLAYER (objeto que se encarga de la reproducción del audio en J2ME) requiere el archivo completo AMR para empezar su reproducción por lo que no se puede ir llenando el buffer del PLAYER y reproducirlo al mismo tiempo. Para solucionar este problema se opto por construir pequeños archivos AMR con trozos de audio del archivo original y utilizar dos objetos PLAYER para la reproducción de todo el archivo. Como el DSS envía frames de audio dentro de cada paquete, el cliente espera hasta tener una cantidad específica de frames, construye el archivo AMR del paquete recibido y lo carga en el buffer del PLAYER. Mientras este trozo de

audio es reproducido, el servidor sigue enviando paquetes RTP, es aquí donde se hace necesario el uso de dos objetos PLAYER.

Cuando un pequeño archivo AMR es construido se carga al buffer del primer PLAYER, mientras éste reproduce el audio, se construye un nuevo archivo a partir de los paquetes RTP entrantes y se carga al buffer del segundo PLAYER, a continuación; cuando el primer PLAYER ha terminado de reproducirse, el segundo PLAYER comienza su reproducción mientras que un nuevo archivo es construido y cargado al buffer del primer PLAYER. Este proceso se repite durante toda la trasmisión del audio solicitado al servidor. El proceso descrito anteriormente se ilustra en la siguiente figura.

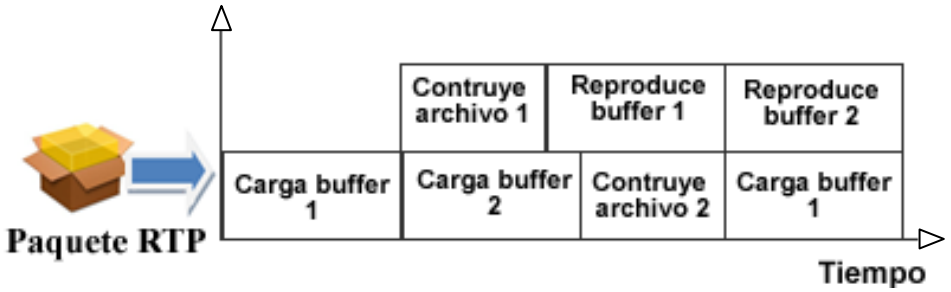


Figura 26. *Reproducción de Paquetes RTP en STREAMJASSMOBILE*

La transmisión y reproducción del archivo en el cliente utilizando los emuladores mencionados fue exitosa, sin embargo existe un pequeña pausa en el audio que se produce durante el cambio entre un PLAYER y el otro. Las figuras que se muestran a continuación son el resultado del proceso que se ha venido mencionando.

6.5 Interfaz Gráfica Aplicación Móvil

Para Desarrollar aplicaciones en dispositivos móviles existen varias maneras para realizar la interfaz gráfica, usando el API de bajo o alto nivel de J2ME, pero estos proveen muy pocos elementos para hacer una UI amigable al usuario.

Se decidió entonces experimentar con una nueva librería que se ofrece para el desarrollo de una interfaz J2ME: *Lightweight UI Toolkit for Java ME* LWUIT [16], esta librería permite crear aplicaciones modernas, agradables, con mayor usabilidad, al mismo tiempo que las hace fáciles de adaptarse a las particularidades de cada dispositivo móvil.

Para comenzar se trabajo el tema utilizando la aplicación ResourceEditor que incluye la librería; Es una aplicación Java que está ubicada en el directorio útil de la distribución oficial. Figura 27.

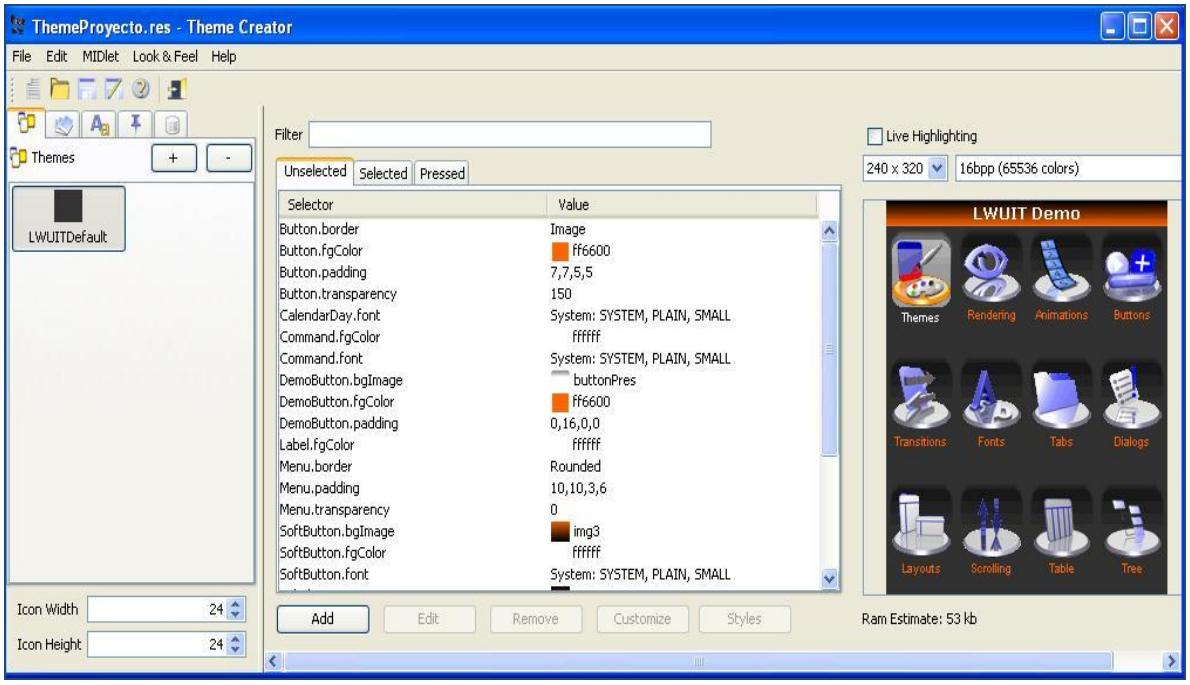


Figura 27. *Resource Editor LWUIT*

El tema se guarda como un archivo .res que va hacer llamado desde el Midlet para que se muestre en el Display.

A continuación se muestra los pantallazos que componen toda la aplicación con una breve explicación.

6.5.1 Pantalla Inicial-Menú inicial

Primero se muestra una pantalla con el logo de la aplicación, para luego mostrar el menú inicial de la aplicación. Éste menú consta de tres módulos: Lista, Buscar y Ayuda, como se muestra en la Figura 28.

Figura 28. *Pantalla y Menú Incial de la aplicación*

6.5.2 Menú Lista

En el formulario de Lista el usuario puede visualizar otro menú que le muestra 3 botones: Artista, Género, Mylist. Para el acceso a la información según el módulo que haya seleccionado. Las figuras 29 y 30 se muestran estas pantallas.

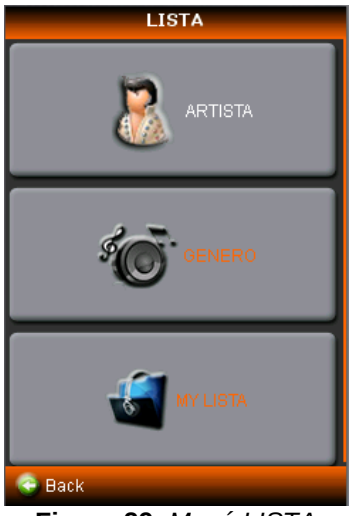


Figura 29. *Menú LISTA*

Figura 30. *Módulos del Lista*

Si se elige el botón Artista se muestra una lista con todos los artistas que contiene la base datos, al elegir alguno de estos artistas, se muestra otra lista con todas sus canciones.

Si elige el botón Género se muestra una lista de los géneros que contiene la base datos, al elegir algún género se muestra una lista de los artistas que tiene canciones de ese género, y al elegir el artista se muestra la lista de sus canciones de ese tipo de género.

Si eligió el botón Mylist se muestra una lista de todas las canciones que se encuentran almacenadas en la base de datos.

Cuando ya se ha elegido una canción de alguna de las lista de canciones anteriores, comienza la reproducción del audio y muestra la interfaz del reproductor con la imagen y el nombre del álbum al que se asocia esa canción, el nombre de la canción y el artista. La figura 31 muestra la interfaz del reproductor.

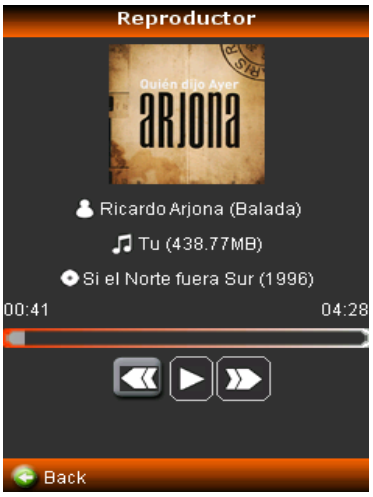


Figura 31. *Interfaz del Reproductor*

6.5.3 Menú Buscar

Este menú es un acceso rápido a alguna canción; en una caja de texto se escribe el nombre o parte del nombre de alguna canción que desea buscar, esta búsqueda genera una lista de canciones cuyo nombre contiene la palabra digitada en la caja de texto. Si digita alguna palabra mal, que no exista en la base de datos o deja la caja de texto vacía se arroja un alerta informando al usuario.

Figura 32. *Módulo Buscar*

Cuando del menú Buscar se elige alguna canción de la lista que genero igualmente empieza la reproducción se muestra el menú de reproducción.

6.5.4 Menú Ayuda

En el menú ayuda se muestra un texto de soporte para dar información al usuario de cómo manejar la aplicación y que puede encontrar en cada menú. Además le informa de las funcionalidades de algunas teclas.

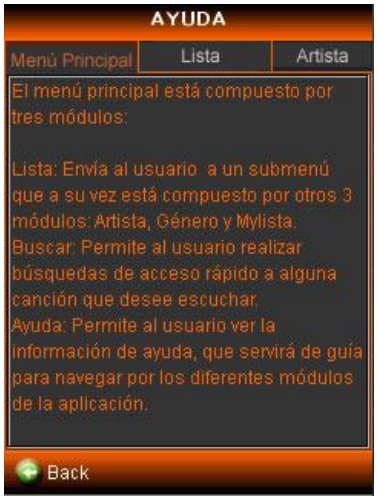


Figura 33. *Módulo Ayuda*

CAPITULO VII

7. PRUEBAS

Las pruebas realizadas a la aplicación móvil se efectuaron desde el inicio del desarrollo de la misma. A continuación se listan.

Pruebas Unitarias

Con las pruebas unitarias a medida que se programaba el código de los módulos que la conforman se verificaba su funcionamiento con el fin de encontrar errores y corregirlos a tiempo. Estas pruebas permiten comprobar el correcto funcionamiento de las líneas de código desarrolladas para la aplicación móvil, estas pruebas se aplican a clases o procesos principales de la aplicación de forma aislada con el objetivo de anticipar posibles fallas y se realizan continuamente durante el desarrollo. Se utilizó la librería JMUnit [17] para CLCD 1.1 para poder realizar estas pruebas. En la Figura 34 se puede observar el uso de estas librerías.

Al iniciar el test se descubren si existen o errores en las clases, en caso de que haya errores aparecerá una barra de progreso en color rojo y luego aparecerá un informe más detallado. De lo contrario aparece una barra de color verde indicando que todos los test realizados fueron exitosos.

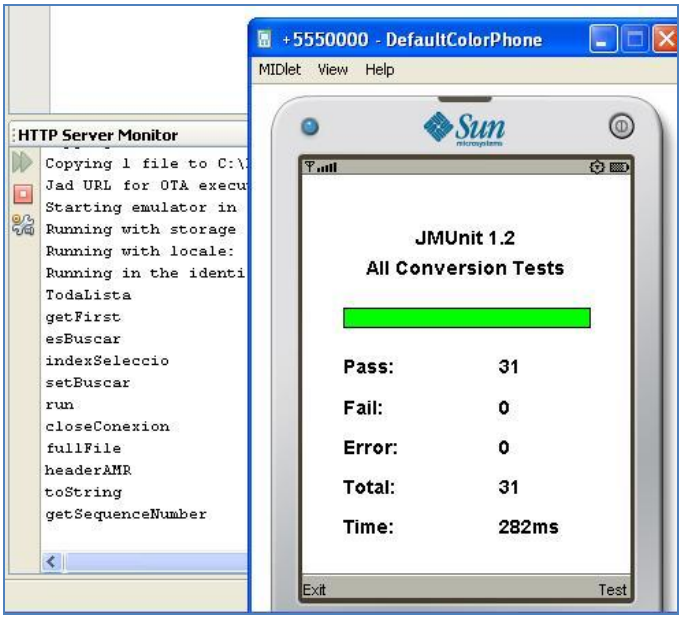


Figura 34. *Pruebas Unitarias JMUnit*

Pruebas Reales

El ejecutable de la aplicación móvil (streamjassmobile.jar) tiene un tamaño de 661 KB, tamaño aceptable para dispositivos móviles y fue probada en varios emuladores ofrecidos por diferentes entornos de desarrollo como el SDK 3.0, Samsumg SDK, LG SDK y NOKIA SDK. La transmisión y reproducción del archivo en el cliente utilizando los emuladores mencionados fue exitosa, sin embargo existe un pequeña pausa en el audio que se produce durante el cambio entre un PLAYER y el otro. Sin embargo, los emuladores de teléfonos móviles están disponibles para probar la aplicación en un entorno PC, esto no representa los dispositivos reales, con sus restricciones en la capacidad de procesamiento y memoria. En el desarrollo de aplicaciones móviles, es primordial realizar pruebas en los dispositivos reales para verificar su funcionamiento.

La aplicación móvil fue instalada y ejecutada en tres dispositivos móviles, la tabla 12 muestra el resultado de las pruebas hechas en dispositivos reales y la Figura 35 muestra la aplicación en un emulador y en un celular N95.

Emulador	Conexión Servidor Web	Servidor Streaming	Reproducción
Sun Java Wireless Toolkit 2.5.2	Exitosa	Exitosa	Fallida*
Java MicroEdition SDK 3.0	Exitosa	Exitosa	Exitosa
LG SDK 1.5	Exitosa	Exitosa	Exitosa
Samsung SDK 1.1.2	Exitosa	Exitosa	Exitosa
Sony Ericsson SDK 2.5.0.6	Fallida**	Fallida**	Fallida**

Dispositivo	Servidor Web	Servidor Streaming	Reproducción
Nokia n95	Exitosa	Exitosa	Exitosa***
Nokia E72	Exitosa	Exitosa	Exitosa***
Sony Ericcson W580i	Exitosa	Fallida****	Fallida****

Tabla 12. Pruebas en Emuladores y Dispositivos Reales

* Se comunica con el servidor y logra reproducir el audio de manera defectuosa. Cada paquete suena muchas veces, no permite la sincronización de los objetos players.
** La aplicación nunca se ejecutó en este emulador.
*** Reproducción del audio con pequeños cortes debido al cambio entre los objetos players.
**** El celular no soporta conexión por sockets.

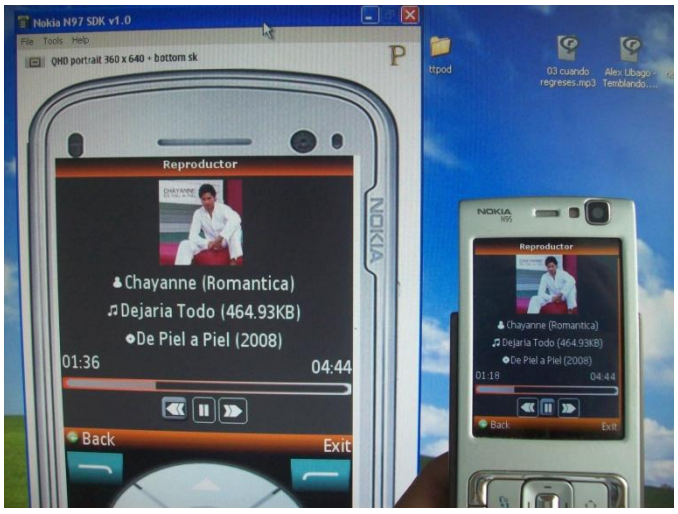


Figura 35. *Pruebas de funcionamiento de la aplicación en emuladores y dispositivos reales*

Pruebas de latencia de la red

Para una buena calidad de servicio streaming se hace necesario de las redes inalámbricas de última generación (2.5G, 3G) que son ideales para la transmisión de contenido multimedia. A continuación se muestra un diagrama que refleja los tiempos de latencia para establecer una sesión de streaming desde un emulador a través de una red cableada y desde el dispositivo móvil Nokia N95 en una red wifi y 3G.

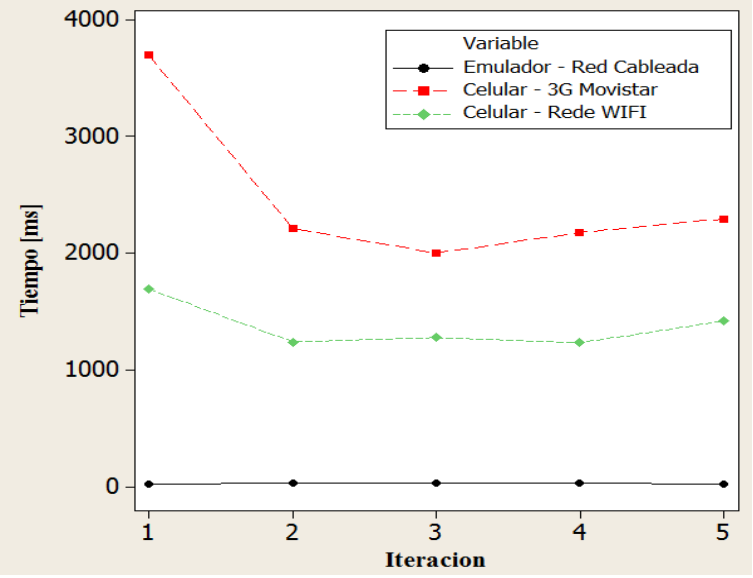


Figura 36. *Comparación de latencias en diferentes redes.*

En la grafica anterior se puede observar la diferencia que existe entre probar una aplicación móvil con un emulador y un dispositivo real, debido a que el emulador se encuentra en un entorno de PC con velocidades de tranmision mas altas que las que ofrecen las redes celulares. De la grafica tambien se puede corroborar que la calidad de un servicio de streaming siempre depende de las condiciones de la red, ademas apreciando los tiempos de latencia obtenidos con el dispositivo movil se observa que es mucho mejor el servicio en una red wifi debido a que en las redes de telefonia móvil no ofrecen velocidades para tranferencia multimedia a menos que se contrate este servicio.

Comparaciones

Generalmente cuando se accede a contenidos multimedia desde un dispositivo celular se hace uso del protocolo HTTP[24], este protocolo genera un retardo en los tiempos de reproducción debido a que el contenido debe ser totalmente transferido para ser escuchado. Por esta razón se planteó el uso de protocolos propios de streaming (RTPS y RTP) para aligera los tiempos de descarga. La siguiente tabla muestra una comparación entre el tiempo que demora en empezar la reproducción la aplicación desarrollada (STREAMIJASSMOBILE) la cual utiliza los protocolos RTSP y RTP frente a una pequeña aplicación desarrollada también en J2ME que utiliza el protocolo HTTP. Para la realización de las muestras se utilizó una red WLAN con tecnología *WiFi* con un punto de acceso de 54 Mbps sobre un emulador.

TAMAÑO ARCHIVO	HTTP Tiempo [ms]	RTSP Tiempo [ms]
824 KB	5470	2953
737 KB	5258	2215

Tabla 13. Comparaciones en transferencia de un archivo de audio entre HTTP y RTSP-RTP

La aplicación se comparó con otra aplicación que ofrece reproducción de contenido multimedia a través de streaming como RealPlayer, sin embargo cabe anotar que esta comparación no se hace bajo las mismas condiciones ya que RealPlayer está desarrollado en un lenguaje diferente al utilizado para el desarrollo de STREAMJASSMOBILE, además RealPlayer hace uso directo del protocolo RTSP y RTP para esto el dispositivo móvil debe tener soporte de estos protocolos, a diferencia de la aplicación STREAMJASSMOBILE en la que se

[24] IETF, RFC 2616. *Hypertext Transfer Protocol (HTTP).* 18 de Noviembre de 2009.

implementaron los protocolos permitiendo que la aplicación se ejecute en el dispositivo móvil sin que este los soporte.

Las comparaciones se hicieron de manera subjetiva debido a que la aplicación RealPlayer al ser comercial no permite el acceso al código fuente para realizar modificaciones con el fin de que el programa tome los tiempos de descarga que es lo que se quiere comparar. Por esta razón se debió ejecutar las dos aplicaciones al mismo tiempo en el emulador del NOKIA N97 para observar cual iniciaba primero la reproducción de un archivo de audio desde el servidor DSS, y se comprobó que la aplicación STREAMJASSMOBILE y RealPlayer tiene tiempos de latencia casi iguales con diferencias de 1 a 2 seg.

Consumo de memoria RAM

Estas pruebas se realizaron utilizando las herramientas de monitoreo preinstaladas en el NETBEANS 6.8[25]. En la imagen 37 se muestra un gráfica del consumo de memoria cuando se ejecuta la aplicación.

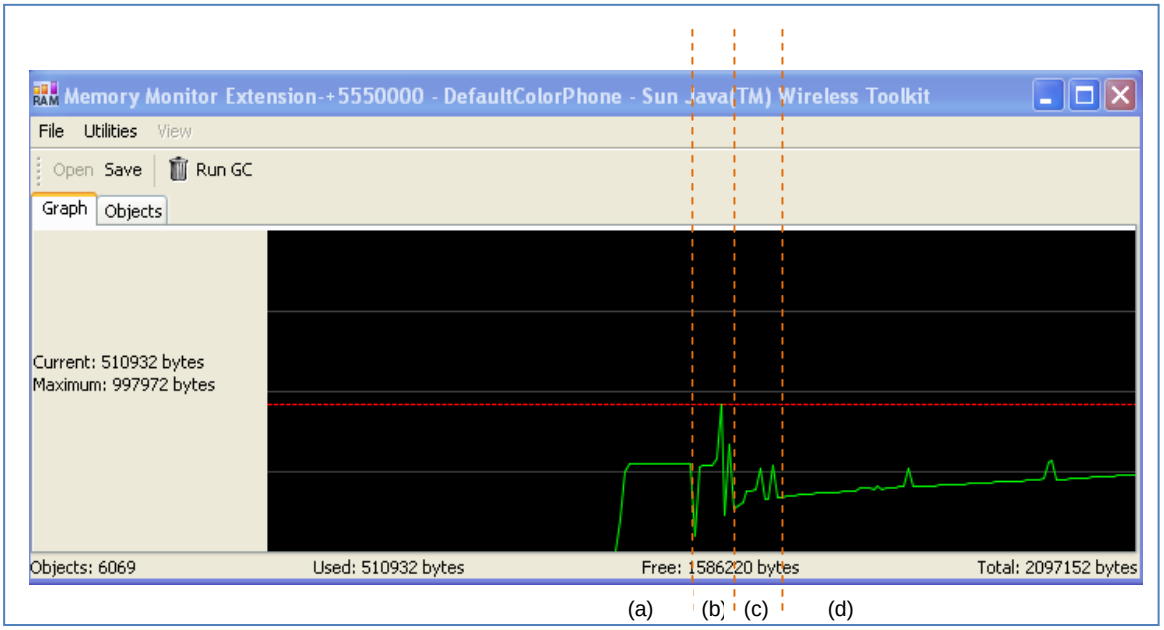


Figura 37. *Monitoreo del consumo de memoria RAM de la aplicación móvil en un emulador*

La anterior imagen muestra una línea constante de consumo de memoria (a) que sucede cuando el emulador carga la aplicación pero ésta aun no se ha ejecutado,

[25] Midlet Profile, http://developers.sun.com/mobility/midp/articles/wtk104/. 23 de Julio 2010.

inmediatamente iniciada la aplicación se establece comunicación con el servidor web para hacer consultas a la base de datos, el consumo de memoria de estas conexiones se refleja en los picos consecutivos de la imagen (b). Cuando se inicia una comunicación con el servidor de streaming también se hace consumo de memoria para estas conexiones (c), por último cuando se inicia una transmisión streaming (d) la grafica refleja una línea de consumo de memoria casi constante, esto sucede porque la aplicación constantemente limpia objetos y hace uso del garbage collection para optimizar el consumo de memoria en el dispositivo. La aplicación consume aproximadamente 420 KB durante su ejecución que es el pico más alto en la grafica.

CONCLUSIONES

- La transmisión de datos multimedia es posible con el uso de protocolos como HTTP y RTSP, pero ya que HTTP es basado sobre TCP su prioridad es que los paquetes lleguen completos y no a tiempo. Todo lo contrario ocurre con el protocolo RTSP basado en UDP que fue creado para hacer entregas a tiempo ideal para la prestación de servicios de streaming.

- Entre los posibles servidores de streaming se escogió el Darwin Streaming Server debido a que es un software open source y que cumple con los requisitos que se adaptan al sistema. Cuando se realizaron las pruebas iniciales de monitoreo del tráfico en el flujo de los datos desde el servidor a un pc se observó dos tipos de trafico principales, tráfico RTP y TCP. La ausencia del tráfico UDP es debido a que el DSS encapsula este tipo de tráfico sobre RTP, protocolo ideal en el envió de los paquetes para streaming.

- Durante el establecimiento de una sesión y captura de los paquetes del servidor DSS se pudo comprobar su excelente desarrollo ya que cumple con cada una de las observaciones consignadas en el RFC (Request For Comments 1889-2367-2326) para la utilización del protocolo RTSP y RTP/AMR. Esto tanto para el inicio de una sesión como la finalización de esta; Sin embargo para que la aplicación móvil pudiera mantener la sesión abierta con el DSS se debió tener permanente comunicación con el servidor por medio de los métodos que este reconoce de lo contrario la conexión se cierra impidiendo recibir paquetes.

- En el desarrollo de la aplicación móvil en cuanto a la programación de la interfaz gráfica y comunicación con el servidor web no se presentaron inconveniente en las pruebas con los emuladores y dispositivos reales gracias a que la librería LWUIT permitió que las imágenes de la interfaz se adaptara al tamaño de las diferentes pantallas y a que actualmente los dispositivos móviles soportan el protocolo HTTP.

- La sección que requirió mayor tiempo en el desarrollo del proyecto fue la implementación de los protocolos RTSP y RTP para J2ME para lograr establecer la comunicación y trasferencia de paquetes desde el servidor streaming. El desarrollo de esta parte del proyecto llevó más tiempo debido a que no se encontraba suficiente información de cómo empezar a realizar la implementación. Se requirió de una investigación minuciosa de las especificaciones de los protocolos para luego esa teoría pasarla a la práctica mediante la programación.

- Otro aspecto importante es que J2ME no tiene soporte para streaming, ya que para reproducir un archivo de audio este debe estar totalmente cargado en el PLAYER. Este problema se pudo solventar con el uso de dos PLAYERS y dos buffers y la construcción de pequeños archivos AMR, sin embargo existe una pausa mínima durante el cambio de los PLAYERS.

RECOMENDACIONES

Para futuras versiones se recomienda que la aplicación también cumpla con la funcionalidad de seguir la secuencia de una lista de canciones automáticamente con el fin de ser más cómoda al usuario y, además que la aplicación permita tener control sobre el volumen del audio.

Para el sistema desarrollado sería de gran aporte que se realice la gestión del contenido del servidor por medio de los usuarios que utilizan la aplicación móvil, desarrollando un portal web donde los usuarios informen sobre el material de audio que desean encontrar en el servidor streaming.

Dentro de la investigación se conoció del protocolo RTCP donde sus mensajes especiales son utilizados para notificar el retardo de la reproducción y el grado de utilización del *buffer*; estas herramientas resultan claves para poder realizar una adaptación de tasa efectiva y sería bueno implementarlo también en j2me para la aplicación móvil.

REFERENCIAS

[1] MACK, Steve. *Streaming media bible*. Hungry Minds, 2002. 869p. ISBN 9780764536502.

[2] QUINTERO O., Juan P. & CASTRO S., Cristian A. *Evaluación de Servidores Streaming de Video Orientado a Dispositivos Móviles*. Trabajo de grado (Ingeniero Electrónico). Universidad de Antioquia. Facultad de Ingeniería. Departamento de Ingeniería Electrónica. Disponible en: http://microe.udea.edu.co/proyectos/DMA. 15 de Noviembre de 2009.

[3] IETF, RFC 1980. *User Datagram Protocol.* Disponible en: http://www.rfc-es.org/rfc/rfc0768-es.txt. 15 de Marzo de 2010.

[4] IETF, RFC 2326. *Real Time Streaming Protocol*. Disponible en: http://www.ietf.org/rfc/rfc2326.txt. 15 de Marzo de 2010.

[5] IETF, RFC 1889. *Transport Protocol for Real-Time Applications*. Disponible en: http://www.ietf.org/rfc/rfc1889.txt. 15 de Marzo de 2010.

[6] Tanenbaum, Andrew S. *Redes de computadoras* (4ª edición), México: Prentice Hall, 2003. 891p. ISBN: 9702601622.

[7] Sun. Java. http://java.sun.com/. 14 de Agosto de 2009.

[8] Sun. J2me. http://java.sun.com/javame/index.jsp. 14 de Agosto de 2009.

[9] MMAPI. JRS 135. http://java.sun.com/products/mmapi/. 17 de Enero de 2010.

[10] Darwin Streaming Server. http://developer.apple.com/darwin/projects/streaming. 02 de Diciembre de 2009.

[11] Helix DNA Server. https://helix-server.helixcommunity.org. 02 de Diciembre de 2009.

[12] SANCHÉZ G, Patricia M. ANÁLISIS DEL DESEMPEÑO DE LAS REDES CELULARES GSM-GPRS. Tesis de grado (Ingeniero Telecomunicaciones). Facultad de Ingeniería. Universidad Autónoma de México.

[13] JOSKOWICZ, José. *Reglas y Prácticas en eXtreme Programming*. Disponible en: http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf. 03 de Agosto de 2009.

[14] 3GPP TS 26.09 “Adaptive Multi-Rate (AMR) speech transcoding”, 3rd Generation Partnership Project (3GPP), 28 de Marzo de 2010.

[15] Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs. http://www.rfceditor.org/rfc/rfc3267.txt. 20 de Abril de 2010.

[16] Biswajit Sarkar. *LWUIT 1.1 for Java ME Developers.* Packt Publishing Ltd, 2009. 364p. ISBN 978-1-847197-40-5. 12 de Enero de 2010.

[17] SourceForge community. J2MEunit framework.
http://j2meunit.sourceforge.net. 21 de Febrero de 2010.

ARTICULOS:

Díaz, A. Merino, P. Panizo, L. Recio, A. *Un estudio práctico del rendimiento del servicio de Streaming de Video sobre redes móviles GPRS/UMTS,* Disponible en : www.lcc.uma.es/~pedro/publications/contelecom06.pdf. 13 de Abril del 2010.

Mao, G. Talevski, A. chang, E. (2007).*Voice over Internet Protocol on Mobile Devices. IEEE. Consultado el 18 de Abril del 2010.*

Lundan M and Igor D.D. (2005).*Mobile Streaming Services in WCDMA networks. IEEE.* Consultado el 11 de abril del 2010.

A. Suárez, E. Macías y F. J. Espino.(2009). *Automatic Resumption of RTSP Sessions in Mobile Phones using JADE-LEAP. IEEE.* Consultado el 18 de abril del 2010.

ANEXO A. TABLA DE CODIGOS DE ESTADO DEL PROTOCOLO RTSP

CODIGO	ESTADO
"100"	Continue
"200"	OK
"201"	Created
"250"	Low on Storage Space
"300"	Multiple Choices
"301"	Moved Permanently
"302"	Moved Temporarily
"303"	See Other
"304"	Not Modified
"305"	Use Proxy
"400"	Bad Request
"401"	Unauthorized
"402"	Payment Required
"403"	Forbidden
"404"	Not Found
"405"	Method Not Allowed
"406"	Not Acceptable
"407"	Proxy Authentication Required
"408"	Request Time-out
"410"	Gone
"411"	Length Required
"412"	Precondition Failed
"413"	Request Entity Too Large
"414"	Request-URI Too Large
"415"	Unsupported Media Type
"451"	Parameter Not Understood
"452"	Conference Not Found
"453"	Not Enough Bandwidth
"454"	Session Not Found
"455"	Method Not Valid in This State
"456"	Header Field Not Valid for Resource
"457"	Invalid Range
"458"	Parameter Is Read-Only
"459"	Aggregate operation not allowed
"460"	Only aggregate operation allowed
"461"	Unsupported transport
"462"	Destination unreachable
"500"	Internal Server Error

"501"	Not Implemented
"502"	Bad Gateway
"503"	Service Unavailable
"504"	Gateway Time-out
"505"	RTSP Version not supported
"551"	Option not supported

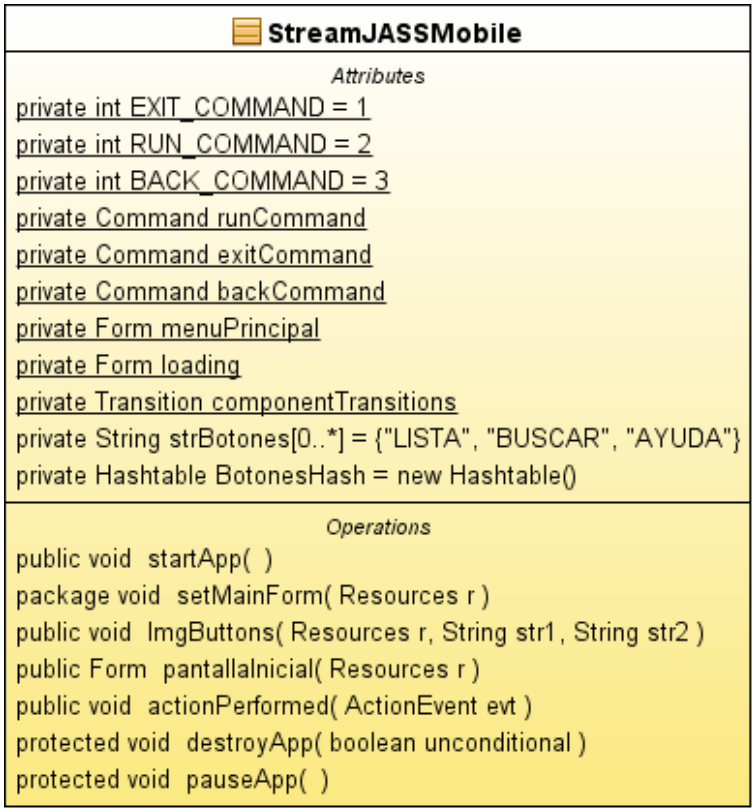
Tabla A.1: Códigos de Estado del Protocolo RTSP

ANEXO B. MANUAL DE USUARIO STREAMJASSMOBILE

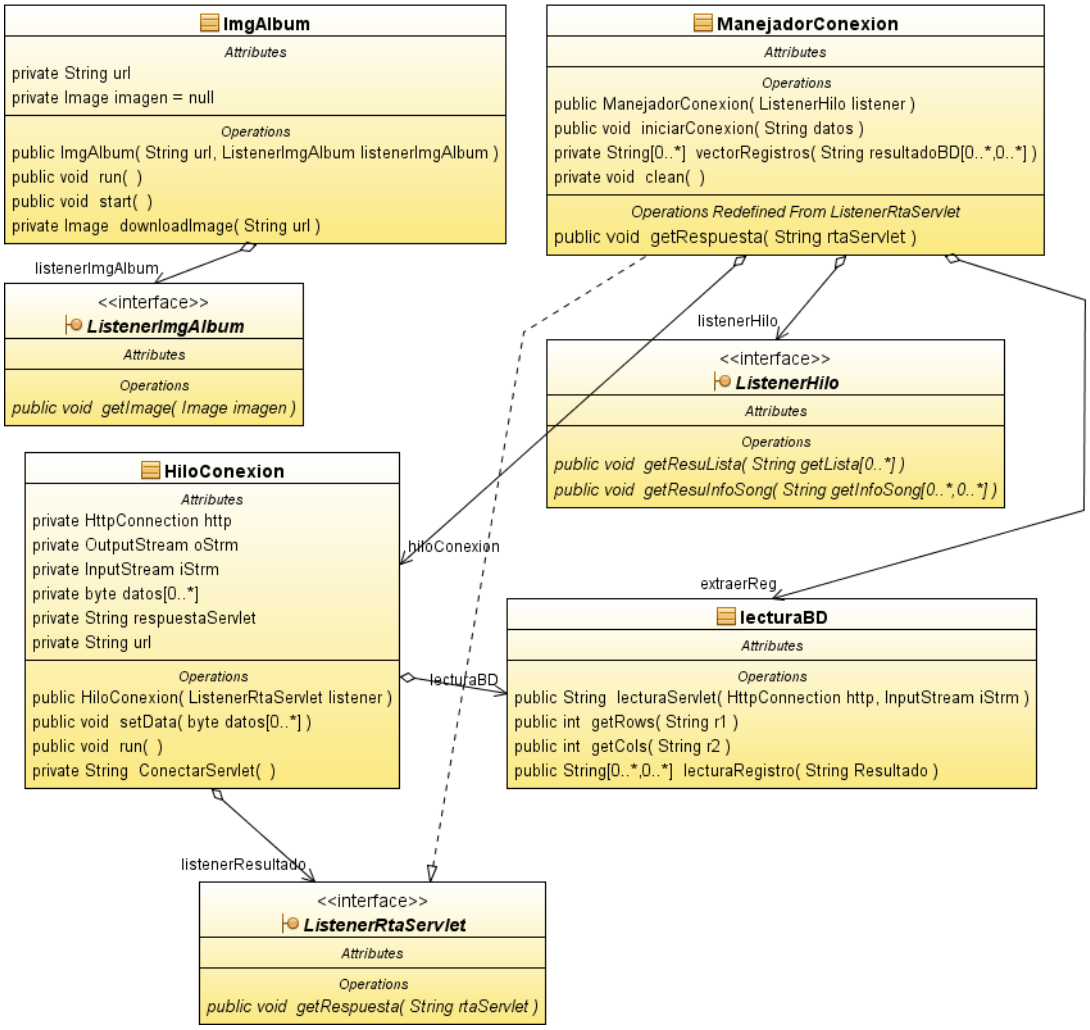
Ver CD.

ANEXO C. DIAGRAMA DE CLASES

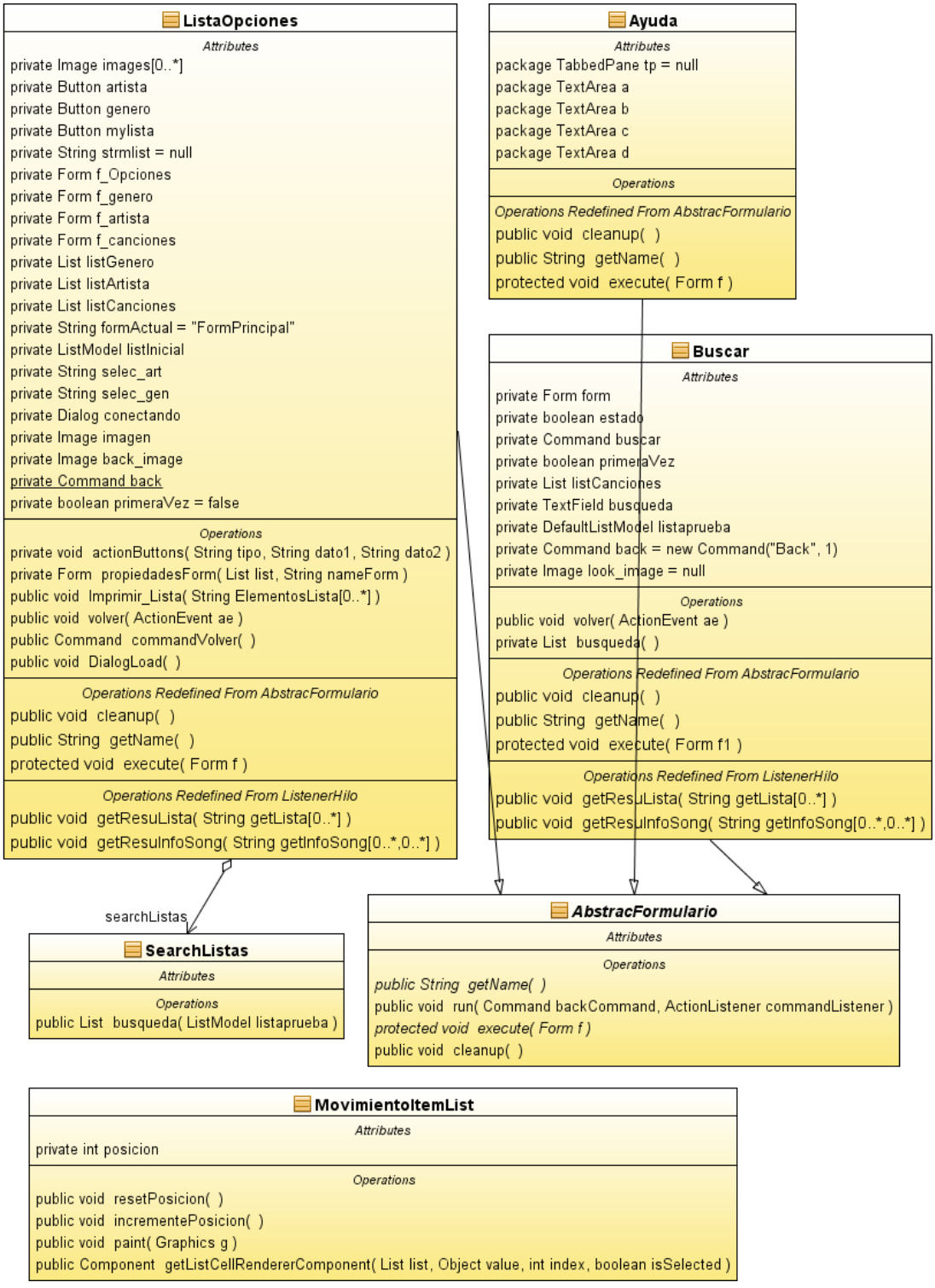
1. STREAMIDLET



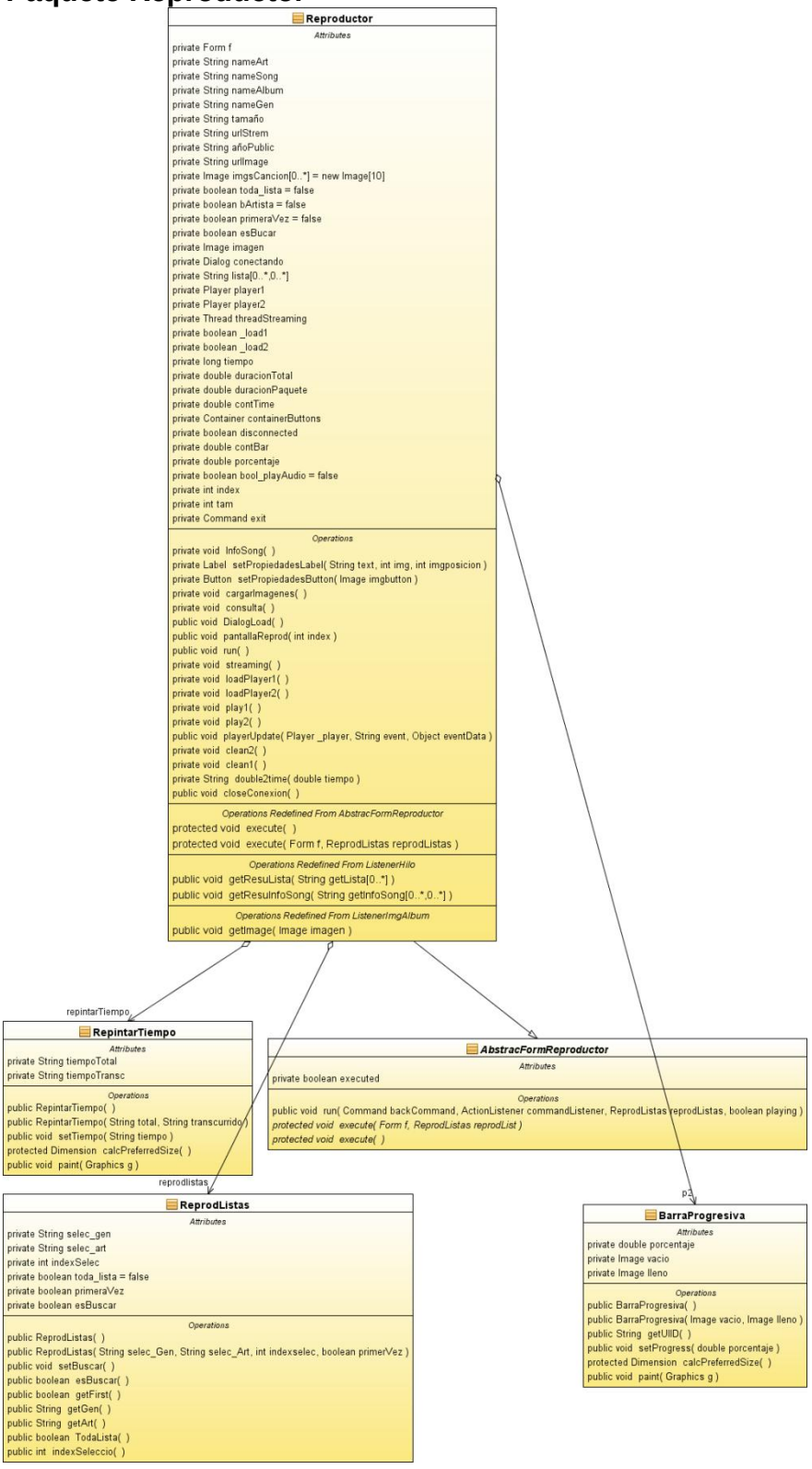
2. Paquete Conexión



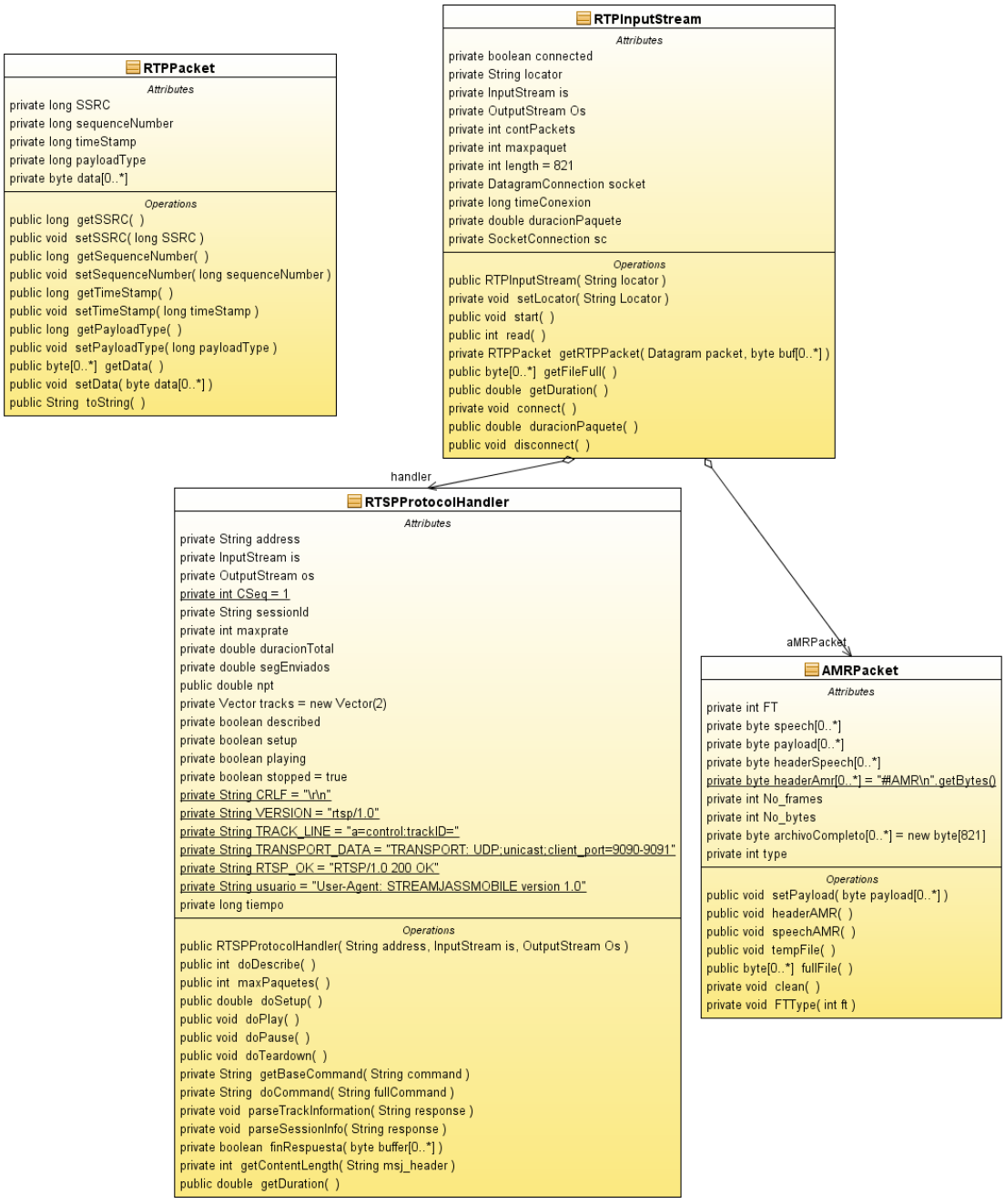
3. Paquete Módulos



4. Paquete Reproductor



5. Paquete STREAMING



ANEXO D. DISEÑO DE LA BASE DE DATOS

1. Entidades y Atributos

Canciones
nombreCancion
idArtista
idGenero
idAlbum
tamaño
url

Album
idAlbum
nombreAlbum
añoPublicacion
imagen

Genero
idGenero
Genero

Genero_has_Artista
idArtista
idGenero

Artista
idArtista
nombreArtista

Llave principal
Llave Foránea
Otros atributos

2. Modelo Entidad-Relación