

**IMPLEMENTACIÓN DEL PROTOCOLO DE CONTROL DE TRANSMISIÓN
(RTCP) EN UNA APLICACIÓN ORIENTADA A DISPOSITIVOS MÓVILES
CELULARES, QUE UTILIZA LA TÉCNICA STREAMING PARA
REPRODUCIR AUDIO**

**ANGÉLICA ZULAY MENDOZA CALDERÓN
ALEX DAVID MORALES ACOSTA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2012

**IMPLEMENTACIÓN DEL PROTOCOLO DE CONTROL DE TRANSMISIÓN
(RTCP) EN UNA APLICACIÓN ORIENTADA A DISPOSITIVOS MÓVILES
CELULARES, QUE UTILIZA LA TÉCNICA STREAMING PARA
REPRODUCIR AUDIO**

**ANGÉLICA ZULAY MENDOZA CALDERÓN
ALEX DAVID MORALES ACOSTA**

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director(a)

PhD(c) Lola Xiomara Bautista Rozo

Codirector(a)

Irene Lizeth Manotas Gutiérrez

Estudiante de Maestría en Ingeniería de Sistemas e Informática

Universidad Industrial De Santander

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2012

Agradezco a Dios por permitirme haber alcanzado este logro y por darme la sabiduría y la fortaleza en cada momento. A mis padres por su apoyo y compañía, a mi familia por ser el motor que me impulsa a realizar cada cosa que hago.

Agradezco a mi compañero Alex Morales por su compañía, su paciencia y por su trabajo para llevar a cabo este proyecto.

Agradezco a la profesora Lola Bautista por su apoyo y tiempo dedicado, a Irene Manotas por toda su colaboración y por estar ahí orientándonos aún en la distancia.

Agradezco a mis compañeros de carrera por estar pendientes de mi evolución durante la realización del proyecto y por todas sus palabras.

Angélica Zulay Mendoza Calderón

Antes que nada agradezco a mis padres Robinson Morales y Jineth Acosta por todo su apoyo y sus buenos consejos durante toda mi vida, gracias a ellos hoy en día soy un profesional y una persona de bien.

Por otra parte manifiesto mi más sincero agradecimiento a Irene Manotas estudiante de Maestría codirectora de esta investigación por la orientación, el seguimiento y la supervisión continua de la misma, pero sobre todo por la motivación y el apoyo recibido a lo largo de esta experiencia. Así mismo agradezco a la PhD Lola Bautista directora de este proyecto por su colaboración y apoyo.

A mis hermanas, María Morales y Kelly Morales por siempre estar ahí para mí para y darme todo su apoyo. A mi compañera y amiga de proyecto, Angélica Zulay Mendoza por brindarme todos sus conocimientos y respetarme, comprenderme y tolerarme durante la realización de este proyecto. A mis compañeros y amigos de la universidad por compartir sus conocimientos y ayudarme en difíciles situaciones.

A todos ellos, muchas gracias.

Alex David Morales Acosta

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN	17
1. PRESENTACIÓN DEL PROYECTO	19
1.1. PLANTEAMIENTO DEL PROBLEMA	19
1.2. ESTADO DEL ARTE	20
EXPERIMENTS IN STREAMING CONTENT IN JAVA ME	23
RTPLIB: UNA LIBRERÍA EN JAVA DE LOS PROTOCOLOS RTP Y RTCP	23
STACK RTP PARA DISPOSITIVOS MÓVILES SYMBIAN	23
1.3. JUSTIFICACIÓN	23
1.4. OBJETIVOS	24
1.4.1. Objetivo General	24
1.4.2. Objetivos Específicos	24
2. MARCO TEÓRICO	26
2.1. TÉCNICA STREAMING	26
2.1.1. Distribución Multimedia	26
2.1.2. Esquema de Distribución	27
2.1.2.1. Broadcast	27
2.1.2.2. Unicast	28
2.1.2.3. Multicast	28
2.2. PROTOCOLOS DE STREAMING	29
2.2.1. Protocolo RTSP (Real Time Streaming Protocol)	30
2.2.1.1. Estados de RTSP	30
2.2.1.2. Métodos de RTSP	30
2.2.2. Protocolo RTP (Real-time Transpor Protocol)	32
2.2.2.1. Encabezado del Paquete RTP	32
2.2.2.2. Funciones de RTP	33
2.3. PROTOCOLO RTCP	34
2.3.1. Funciones Básicas	35
2.3.2. Tipos de Paquetes RTCP	35
2.3.3. Estructura de Paquetes RTCP	36
2.3.3.1. Formato Paquete SR	37
2.3.3.2. Formato Paquete RR	39

2.3.3.3. Formato Paquete SDES	39
2.3.3.4. Formato Paquete BYE	40
2.3.3.5. Formato Paquete APP	41
2.3.4. Paquetes Compuestos	42
2.3.5. Restricciones	43
2.4. SERVIDORES DE STREAMING	44
2.5. REDES MÓVILES	44
2.5.1. Redes de Telefonía Móvil	44
2.5.2. Arquitectura de Red para el Sistema Streaming	47
2.5.2.1. Conexión a internet por red móvil celular	47
2.5.2.2. Conexión a internet por Wi-Fi	47
2.6. CÓDECS DE AUDIO	48
2.6.1. Familia de Códecs de audio AMR	49
2.6.2. Comparación de Códecs de Audio	50
2.7. DESARROLLO DE APLICACIONES MÓVILES CELULARES	51
2.7.1. JAVA	52
2.7.2. J2ME (Java 2 Micro Edition)	52
2.7.3. MMAPAPI (Mobile Media API)	53
3. METODOLOGÍA Y REQUERIMIENTOS	55
3.1. REQUERIMIENTOS	55
3.1.1. Requerimientos Funcionales	55
3.1.2. Requerimientos No Funcionales	56
3.2. HERRAMIENTAS UTILIZADAS	56
3.3. CASOS DE USO	57
4. DESARROLLO DEL PROYECTO	58
4.1. REVISIÓN E IMPLEMENTACIÓN DE LOS PROTOCOLOS RTSP Y RTP	58
4.1.1. Implementación de RTSP	58
4.1.2. Implementación de RTP	59
4.2. IMPLEMENTACIÓN DE RTCP	60
4.2.1. Lectura de paquetes RTCP recibidos	61
4.2.1.1. Paquete Compuesto RTCP Recibido	61
4.2.1.2. Paquete SR Recibido	62
4.2.1.3. Paquete SDES Recibido	62
4.2.1.4. Paquete BYE Recibido	63
4.2.2. Armado y Envío de Paquetes RTCP	64

4.2.2.1. Paquete RR Enviado	64
4.2.2.2. Paquete SDES Enviado	67
4.2.2.3. Paquete BYE Enviado	68
4.2.3. Tiempos de Envío de los Paquetes RTCP Compuestos	69
4.3. SELECCIÓN DE SERVIDOR DE STREAMING	70
4.4. CÓDECS	71
4.4.1. Codificación de Archivo de Audio AMR-WB	72
4.4.2. Estructura de un Paquete RTP	74
4.4.3. Estructura de un Archivo AMR-WB	74
4.5. MÓDULOS DE LA INTERFAZ GRÁFICA DE LA APLICACIÓN MÓVIL	75
4.5.1. Base de Datos de la Aplicación	75
4.5.2. Interfaz Gráfica del Reproductor de Audio	76
4.5.3. Buscador	77
4.5.3.1. Interfaz Móvil del Buscador	77
4.5.3.2. Algoritmo de Búsqueda	77
4.5.4. Usuarios	79
4.5.4.1. Formularios de la Interfaz Móvil de Usuarios	79
4.5.4.2. Servlet de Usuarios	80
5. RESULTADOS Y PRUEBAS	81
5.1. PRUEBAS UNITARIAS	81
5.2. PRUEBAS DE LA APLICACIÓN EN EMULADORES Y DISPOSITIVOS REALES	82
5.3. PRUEBA DE RTCP	84
5.3.1. Ancho de Banda Usado por RTCP	85
5.3.2. Comportamiento de RTCP	86
5.3.3. Comportamiento de la Fluctuación	88
5.3.4. Pérdida de Paquetes	88
5.4. RENDIMIENTO DE LA RED EN LA TRANSMISIÓN DE CÓDECS	90
5.5. CONSUMO DE RAM DE LA APLICACIÓN	90
6. CONCLUSIONES	92
7. RECOMENDACIONES	94
BIBLIOGRAFÍA	95
ANEXOS	99

LISTA DE FIGURAS

	Pág.
Figura 1. Envío de un archivo desde el servidor al cliente	27
Figura 2. Broadcast	28
Figura 3. Unicast	28
Figura 4. Multicast	29
Figura 5. Diagrama de estado RTSP	31
Figura 6. Estructura del paquete RTP	32
Figura 7. Formato del Informe del Emisor (SR)	37
Figura 8. Formato del Informe del Receptor (RR)	39
Figura 9. Formato de la Descripción de la Fuente (SDES)	40
Figura 10. Formato del Fin de la Transmisión (BYE)	41
Figura 11. Formato del Paquete Definido por la Aplicación (APP)	41
Figura 12. Estructura de un Paquete Compuesto RTCP	43
Figura 13. Generaciones de la Telefonía Móvil	45
Figura 14. Diagrama de Despliegue: Arquitectura de Red del Sistema Streaming	48
Figura 15. Arquitectura del API Multimedia	53
Figura 16. Ciclo de vida de un objeto Player	54
Figura 17. Casos de Uso de la Aplicación	57
Figura 18. Métodos RTSP en la aplicación WIRESHARK	59
Figura 19. Carga Útil Paquete RTP	60
Figura 20. Envío de paquetes RTCP	60
Figura 21. Paquete Compuesto RTCP	61
Figura 22. Campos del Paquete SR	62
Figura 23. Campos del Paquete SDDES	63
Figura 24. Campos del Paquete BYE	63
Figura 25. Campos del Paquete RR	64
Figura 26. Representación de la Diferencia Entre Llegadas	65
Figura 27. Representación Gráfica del Cálculo de LSR	66

Figura 28. Representación Gráfica del DLSR	67
Figura 29. Campos del Paquete SDES	68
Figura 30. Campos del Paquete BYE	68
Figura 31. Aplicaciones y Tasas de bits de Códecs	71
Figura 32. Proceso Hint	73
Figura 33. Tamaños códec	73
Figura 34. Formato de Archivo AMR-WB	74
Figura 35. Diagrama UML de Base de Datos de la Aplicación	75
Figura 36. Reproductor de Audio Sobre Emulador.	76
Figura 37. Pantalla de Interfaz del Buscador.	77
Figura 38. Formularios del Módulo de Usuarios.	79
Figura 39. Pantallas JMunit	82
Figura 40. Ancho de Banda de la Sesión	85
Figura 41. Gráfica de Paquetes RTCP	87
Figura 42. Fluctuación de los paquetes RTP	88
Figura 43. Pérdida de Paquetes	89
Figura 44. Rendimiento de Red	90
Figura 45. Consumo de RAM	91

LISTA DE TABLAS

	Pág.
Tabla 1. Inicios del Streaming	21
Tabla 2. Aplicaciones de Streaming	21
Tabla 3. Investigaciones sobre Streaming	22
Tabla 4. Distribución Multimedia	27
Tabla 5. Tabla de estados del protocolo RTSP	30
Tabla 6. Métodos RTSP	30
Tabla 7. Encabezado RTP	32
Tabla 8. Funciones RTP	33
Tabla 9. Tipos de Paquetes RTCP	36
Tabla 10. Cabecera de los paquetes RTCP	36
Tabla 11. Tipos de SDES	40
Tabla 12. Generaciones, tecnologías y redes de la telefonía móvil	46
Tabla 13. Características de la Familia de Códecs AMR	50
Tabla 14. Comparación de la familia de Códecs AMR	51
Tabla 15. Plataformas JAVA	52
Tabla 16. Estados el Ciclo del Vida del Objeto Player	54
Tabla 17. Herramientas Utilizadas	56
Tabla 18. Servidores de Streaming	70
Tabla 19. Tabla "Audio" de la BD	78
Tabla 20. Matriz creada a partir de la tabla "Audio"	78
Tabla 21. Resultado interno de una búsqueda	78
Tabla 22. Prueba de Emuladores y Dispositivos Físicos	83
Tabla 23. Valores estadísticos del porcentaje de ancho de banda ocupado por RTCP	86

LISTA DE ANEXOS

	Pág.
Anexo A. Caso de Uso de la Aplicación	99
Anexo B. Características de Referencias de Dispositivos Móviles Celulares	103
Anexo C. Diagrama de Secuencia (Buscar)	105
Anexo D. UML Aplicación	106
Anexo E. Pruebas JMunit	111

RESUMEN

Título: IMPLEMENTACIÓN DEL PROTOCOLO DE CONTROL DE TRANSMISIÓN (RTCP) EN UNA APLICACIÓN ORIENTADA A DISPOSITIVOS MÓVILES CELULARES, QUE UTILIZA LA TÉCNICA STREAMING PARA REPRODUCIR AUDIO.*

Autores: Angélica Zulay Mendoza Calderón y Alex David Morales Acosta**

Palabras Clave: Streaming, RTP, RTSP, RTCP, AMR, AMR-WB, AMR-WB+, JME.

Descripción:

Esta investigación se presenta como un complemento del proyecto realizado en la Universidad Industrial de Santander titulado “*Desarrollo de un sistema Streaming de audio orientado a dispositivos móviles*”, ésta investigación trabajó la implementación de los protocolos de comunicación que dan soporte a la técnica Streaming, utilizando *Real Time Transport Protocol* (RTP) y *Real Time Streaming Protocol* (RTSP), para el transporte de datos y establecer la comunicación entre los miembros de una sesión Unicast.

En el presente trabajo de investigación se logró implementar el protocolo de control RTCP, en el lenguaje Java Micro Edition (JME), el cual brinda al protocolo RTP, servicios de control, sincronización e identificación de sesión, entre otros, mediante el envío y recepción de paquetes tipo RTCP. La tarea principal del protocolo RTCP es dar soporte a la Calidad del Servicio (QoS) en un sistema de Streaming.

Además, se realizó un estudio sobre los códecs de audio pertenecientes a la familia AMR con el fin de determinar cuál era el más adecuado para el uso de Streaming en dispositivos móviles celulares. Adicionalmente, el prototipo de la aplicación móvil desarrollado muestra funciones adicionales respecto a la aplicación desarrollada en el proyecto anterior, presentando mejoras en la interacción entre los usuarios y la aplicación.

La aplicación móvil incluye un reproductor que cuenta con las funciones principales como lo son play, stop, pause, next y back, adicionalmente permite el manejo para el control de volumen y suministra información acerca del audio y de la reproducción. Por otra parte, la aplicación tiene un módulo para el manejo de usuarios que permite guardar listas.

* Proyecto de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas.
Director(a): Lola Xiomara Bautista Rozo. Codirector(a): Irene Lizeth Manotas Gutiérrez

SUMMARY

Title: CONTROL PROTOCOL (RTCP) IMPLEMENTATION IN A MOBILE DEVICE-ORIENTED APPLICATION USING THE STREAMING TECHNIQUE TO PLAY AUDIO.*

Authors: Angélica Zulay Mendoza Calderón and Alex David Morales Acosta**

Keywords: Streaming, RTP, RTSP, RTCP, AMR, AMR-WB, AMR-WB+, JME.

Description:

This research project is presented as a complement to the project made in the Industrial University of Santander titled "Development of a Mobile device-oriented Audio Streaming System". That research worked the implementation of communication protocols that give support Streaming technique, using Real Time Transport Protocol (RTP) and Real Time Streaming Protocol (RTSP), to transport data and establish communication between members of a unicast session.

In the present research project was implemented the Real Time Control Protocol (RTCP) for mobile phones based on Java Micro Edition (JME). This protocol provides control services to RTP, synchronization and identification session, among others, sending and receiving RTCP packet type. The main task of the RTCP protocol is to support the Quality of Service (QoS) in a streaming system.

Additionally, we performed a study of the audio codecs belonging to the AMR family, in order to determine which was best suited for Streaming in mobile phones. Additionally, the prototype of the mobile developed application shows additional features compared to the previous project, introducing improvements in the interaction between users and the application.

The mobile application includes a player that has the main functions such as play, stop, pause, forward and backward, additionally allows management to control volume and provides information about audio and playback. Moreover, the application has a module for handling users that allows store lists as well.

* Minor Degree Project

** Faculty of Physic and Mechanic Engineering. School of System Engineering.
Director: Lola Xiomara Bautista Rozo. Codirector: Irene Lizeth Manotas Gutiérrez

INTRODUCCIÓN

La reproducción mediante la técnica Streaming ha tomado gran importancia dentro del desarrollo de aplicaciones multimedia, y sus avances en cuanto a su utilización en dispositivos móviles se encuentran en acenso. La técnica de Streaming permite iniciar la reproducción de archivos de audio o video alojados en un servidor sin tener que descargar un archivo en su totalidad a través de la implementación de protocolos de transmisión de datos en la parte del cliente.

Por ser una comunicación sobre una red, en una transmisión de Streaming es apropiado tener un control sobre la transferencia de los datos. Dentro del desarrollo de este proyecto se pretende llevar a cabo la implementación de un protocolo de control que le de soporte a la técnica de Streaming sobre una aplicación móvil creada en JME.

Este documento muestra el resultado de la investigación que permitió llevar a cabo la implementación del protocolo de control Real Time Control Protocol (RTCP) y la utilización de los códecs de audio de la familia AMR para la técnica Streaming. El documento se distribuye de la siguiente forma: El Capítulo 1 presenta la descripción del problema a resolver y la justificación del proyecto, donde se enfatiza en la importancia del desarrollo de aplicaciones móviles que utilicen la técnica Streaming para el acceso a archivos multimedia. Así mismo se exponen los objetivos que se quieren alcanzar dentro del proyecto. El Capítulo 2 muestra los conceptos necesarios para el desarrollo de este proyecto. En el Capítulo 3 se hace una descripción del uso de la metodología utilizada para el desarrollo del proyecto y se muestran los requerimientos y el diseño del sistema, así como las herramientas usadas para llevar a cabo el seguimiento de los avances proyecto. El Capítulo 4 detalla paso a paso cómo se logró desarrollar la aplicación móvil que incluye: la reproducción de audio con el soporte de los protocolos propios de Streaming, el manejo de usuarios y un motor de búsqueda que permite elegir qué se desea reproducir. El Capítulo 5 expone las pruebas realizadas a la aplicación y los resultados obtenidos en el

desarrollo del proyecto. En el Capítulo 6 se describe las conclusiones obtenidas de la investigación e implementación realizadas. Finalmente, el Capítulo 7 plantea algunas recomendaciones para posteriores investigaciones en el desarrollo de aplicaciones que utilicen la técnica de Streaming

1. PRESENTACIÓN DEL PROYECTO

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad el avance tecnológico ha forzado el aumento en las necesidades de comunicación, lo que nos lleva a querer incorporar las herramientas que la tecnología ofrece cada vez más a nuestras vidas, debido a que éstas además de facilitar la comunicación permiten prestar otros servicios, entre ellos la utilización de información multimedia. En este contexto, la tecnología móvil aparece para garantizar que estos servicios también puedan ser utilizados desde diferentes lugares y en constante movimiento.

Sin embargo, y a pesar de los avances en cuanto a compresión de información y aumento de las velocidades de transmisión se refiere, aún la utilización de archivos (particularmente audio y video) desde Internet, mediante previa descarga, requiere un tiempo de espera considerable. Como respuesta a esta situación surge la técnica denominada *Streaming*, cuya labor es permitir la reproducción de archivos sin la necesidad de guardar los datos en el disco del cliente de manera permanente.

A pesar de que la técnica de Streaming ha tenido un gran auge en Internet, mostrando grandes avances en cuanto a la utilización desde computadores, su implementación en dispositivos móviles celulares es escasa. Hoy en día existen pocas aplicaciones, principalmente privadas, que prestan este servicio. Sin embargo, dentro de universidades nacionales e internacionales se han venido realizando investigaciones, en las que se ha trabajado la utilización de la técnica Streaming para éstos dispositivos, en base a los protocolos TCP y HTTP, o los protocolos RTP y RTSP. En este proyecto se pretende crear una aplicación móvil celular que agrupe las principales características que brinda la técnica Streaming, con ayuda de los protocolos *Real-time Transport Protocol* (RTP), *Real-time Streaming Protocol* (RTSP), e incorporando el protocolo de control *Real-time Control Protocol* (RTCP).

El protocolo RTP basado en UDP, se usa frecuentemente en sistemas de Streaming junto con el protocolo RTSP. Dentro de este proyecto se propone la implementación del Protocolo de Control de Transmisión (RTCP), como complemento al trabajo de investigación “*Desarrollo de un Sistema de Streaming de Audio Orientado a Dispositivos Móviles*” realizado en la Universidad Industrial de Santander; donde se implementaron los protocolos RTSP y RTP para dispositivos móviles, pero no se llevó a cabo la implementación del protocolo RTCP. El protocolo RTCP proporciona al protocolo RTP servicios de control, sincronización e identificación de sesión, entre otros. Con la implementación del protocolo RTCP se busca que la técnica Streaming tenga soporte para la calidad del servicio (QoS) en el sistema realizado anteriormente.

1.2. ESTADO DEL ARTE

La utilización de archivos multimedia en Internet está en auge, es por esto que los esfuerzos por realizar mejoramientos en cuanto a compresión de información, progreso de la tecnología y búsqueda de nuevas formas para transferir y compartir dichos archivos, son sobresalientes. Es así como la técnica de Streaming se convierte en una solución eficaz y sirve como apoyo para solucionar problemas relacionados con compartir archivos multimedia.

Aunque la técnica de Streaming inicialmente se desarrolló para computadores de escritorio, hoy en día ya se ven aplicaciones orientadas a dispositivos móviles que utilizan su implementación. En Colombia estos avances aún no son muy notorios, como si sucede en países desarrollados como Estados Unidos, países europeos y orientales.

De los inicios de la implementación de la técnica de Streaming se puede destacar las siguientes compañías (ver Tabla 1), cada una con su respectiva aplicación con que iniciaron en el campo de desarrollo de aplicaciones para Streaming multimedia.

Tabla 1. Inicios de la técnica de Streaming

Compañía	Año	Aplicación
Apple	1991	Quicktime
RealNetwork	1995	Reallplayer (Real Audio y Real Video)
Microsoft	1996	Windows Media Service (Netshow)

Autores del Proyecto

Existen varios sitios Web que prestan servicios de reproducción de video y audio por medio de la técnica de Streaming. Al mismo tiempo ya existen aplicaciones móviles que usan la técnica con diferentes fines. La Tabla 2 muestra un resumen de dichas aplicaciones y sus principales características, separando las aplicaciones Web o de escritorio de las aplicaciones desarrolladas para dispositivos móviles.

En Colombia existen actualmente pocas empresas dedicadas a la creación de aplicaciones móviles, entre ellas *Elibom*¹ la cual solo presta servicios de *envío de mensajes cortos SMS*. Otra empresa es *Hi Media Digital*² que hasta el momento ha realizado algunos desarrollos de aplicaciones móviles para Iphone. Sin embargo, aún no se conoce alguna aplicación creada en Colombia que brinde servicios de Streaming para video o audio.

Tabla 2. Aplicaciones de Streaming

Tipo de Aplicación	Aplicación	Características
Web o de Escritorio	last.fm ³	Radio en Internet.
	Pandora ⁴	Radio en Internet. No disponible para Colombia.
	rokola.fm ⁵	Radio en Internet.
	Sonora ⁶	Permite crear y guardar listas de reproducción. Versión gratuita limitada.
	Grooveshark ⁷	Permite crear y guardar listas de reproducción. Versión gratuita limitada.

¹ Página oficial de la empresa. [online]. Disponible en Internet: <http://www.elibom.com/>

² Página oficial de la empresa. [online]. Disponible en Internet: <http://himediaigital.com/>

³ Página oficial de la aplicación. [online]. Disponible en Internet: <http://www.lastfm.es/home>

⁴ Página oficial de la aplicación. [online]. Disponible en Internet: <http://www.pandora.com/restricted>

⁵ Página oficial de la aplicación. [online]. Disponible en Internet: <http://www.rockola.fm/>

⁶ Página oficial de la aplicación. [online]. Disponible en Internet: <http://sonora.terra.com.co/#/home>

⁷ Página oficial de la aplicación. [online]. Disponible en Internet: <http://grooveshark.com/>

	Spotify ⁸	No disponible para Internet pero si en los sistemas operativos Microsoft Windows, Mac OS X y Linux. Permite crear y guardar listas de reproducción. Versión gratuita limitada.
	Youtube	Servicios generalmente gratuitos.
	Megavideo	Gratis a reproducciones que no superen los 72 minutos
	Netflix	Permite ver películas y series de televisión en internet (Ya se encuentra disponible para Colombia)
Móviles	Groovesharkmobile ⁹	Android, iPhone, BlackBerry y Nokia
	Spotify	Disponible para Symbian, iPhone, Android y BlackBerry
	Last.fm	Disponible sólo para Iphone
	Didiom ¹⁰	Es una aplicación móvil orientada a la reproducción de audio streaming que se encuentre en el computador personal
	Skifta ¹¹	Orientada a la utilización de archivos almacenados en dispositivos electrónicos conectados en red (televisores, equipos de sonido, computadores).

Autores del Proyecto

En lo referente a proyectos de investigación en universidades tanto nacionales como internacionales se han realizado trabajos a nivel de maestría donde se ha llevado a cabo la implementación de la técnica de Streaming y el uso de códecs de audio adecuados para multimedia en dispositivos móviles. Por lo general dichas investigaciones se enfocan en la implementación del protocolo RTP principalmente. Dentro de las investigaciones consultadas se destacan aquellas mostradas en la Tabla 3.

Tabla 3. Investigaciones sobre Streaming

Titulo de la Investigación	Lugar de Desarrollo	Características de la Investigación
		Nacionales
STREAMING DE AUDIO A TRAVES DE DISPOSITIVOS MOVILES UTILIZANDO J2ME	Universidad el Bosque (Bogotá)	Consiste en la implementación de Streaming de audio a través de dispositivos móviles utilizando J2ME. Este desarrollo se centró en el manejo de los protocolos HTTP y TCP como soporte a la utilización de la técnica de Streaming. [1]

⁸ Página oficial de la aplicación. [online]. Disponible en Internet: <http://www.spotify.com/int/>

⁹ Página oficial de la aplicación. [online]. Disponible en Internet: <http://grooveshark.com/>

¹⁰ Página oficial de la aplicación. [online]. Disponible en Internet: <http://www.didiom.com/>

¹¹ Página oficial de la aplicación. [online]. Disponible en Internet: <http://www.skifta.com/>

EVALUACIÓN DE SERVIDORES DE STREAMING DE VIDEO ORIENTADO A DISPOSITIVOS MÓVILES	Universidad de Antioquia (Medellín)	En este proyecto se realizaron pruebas con diferentes servidores Streaming, El objetivo fue seleccionar el servidor más apropiado para brindar un buen servicio orientado a clientes en dispositivos móviles celulares. [2]
DESARROLLO DE UN SISTEMA STREAMING DE AUDIO ORIENTADO A DISPOSITIVOS MÓVILES	Universidad Industrial de Santander (Bucaramanga)	Se logró la implementación de los protocolos RTP y RTSP dentro del desarrollo de un sistema Streaming de audio para dispositivos móviles celulares. [3]
Internacionales		
EXPERIMENTS IN STREAMING CONTENT IN JAVA ME	Java.net	Da sugerencias de cómo llevar a cabo la implementación de los protocolos de streaming. [4]
RTPLIB: UNA LIBRERÍA EN JAVA DE LOS PROTOCOLOS RTP Y RTCP	Universidad Rey Juan Carlos (Madrid)	Implementación de una librería RTP en Java, ofreciendo compatibilidad con determinados dispositivos Nokia e implementación del Protocolo de Control RTCP. [5]
STACK RTP PARA DISPOSITIVOS MÓVILES SYMBIAN	Universidad Politécnica de Cataluña (Barcelona)	Este proyecto es una modificación a una librería ya existente JRTPLIB (escrita en C++), lo que busca este proyecto es crear una PILA de RTP específica para Symbian OS (que tiene como lenguaje nativo a C++). [6]
ANALYSIS AND EVALUATION OF AUDIO CODECS FOR MOBILE DEVICES	Lund University (Suecia)	El propósito de esta tesis es evaluar el códec de audio adecuado para dispositivos móviles. [7]

Autores del Proyecto

1.3. JUSTIFICACIÓN

Es evidente que el avance tecnológico en el campo de aplicaciones para dispositivos móviles celulares esta en ascenso, es ahí donde este proyecto se convierte en una clara oportunidad para que la Universidad Industrial de Santander esté presente en esta evolución.

El impacto esperado será beneficioso particularmente para el Grupo de Investigación en Ingeniería Biomédica (GIIB) y para la comunidad de la Escuela de Ingeniería de Sistemas e Informática, contribuyendo con una base de investigación en lo que respecta a la técnica Streaming y su correspondiente aplicación en dispositivos móviles celulares.

Además, debido a que este proyecto se plantea como continuación de uno previo, realizado dentro de la Universidad Industrial de Santander, se espera lograr unificar ambas investigaciones con el fin de entregar resultados que permitan a partir de la aplicación móvil propuesta, mostrar un avance significativo en cuanto a la investigación de la técnica Streaming y sus posibles aplicaciones tanto dentro como fuera del ambiente académico.

Dentro de los resultados esperados se encuentra el lograr una contribución al mejoramiento de la calidad en la transmisión que maneja el sistema de Streaming previamente propuesto para dispositivos móviles celulares, a través de la implementación del protocolo RTCP.

Además, uno de los objetivos presentados es la actualización de la aplicación software y la inclusión de otras opciones como lo son el manejo de listas y el control de volumen de los archivos de audio recibidos a través de Streaming.

1.4. OBJETIVOS

1.4.1. Objetivo General

Implementar el protocolo de control de transmisión Real Time Control Protocol (RTCP) como complemento al trabajo de investigación “DESARROLLO DE UN SISTEMA DE STREAMING DE AUDIO ORIENTADO A DISPOSITIVOS MÓVILES”, dentro de la realización de un prototipo de aplicación software orientada a dispositivos móviles celulares que permite compartir y reproducir archivos de audio sobre redes de telefonía móvil o redes inalámbricas.

1.4.2. Objetivos Específicos

- Implementar el protocolo de control de transmisión Real Time Control Protocol (RTCP) para dispositivos móviles celulares.

- Aplicar los protocolos RTSP, RTP y RTCP para la transmisión de audio en un sistema de streaming.
- Determinar qué códec de audio de la familia AMR es el más apropiado para la transmisión de streaming en dispositivos móviles.
- Realizar el análisis, diseño e implementación del prototipo software para el dispositivo móvil celular.

2. MARCO TEÓRICO

2.1. TÉCNICA STREAMING

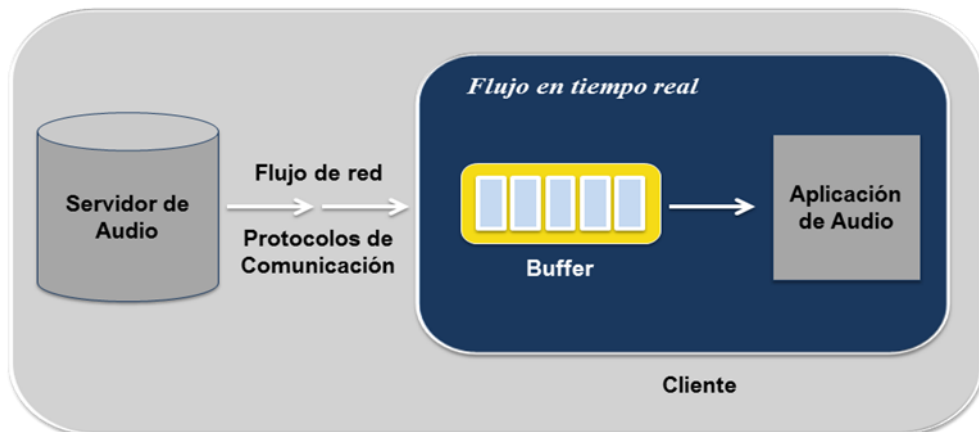
“Se entiende por Streaming, la capacidad de distribución de contenido multimedia, con la característica de poder escuchar y/o visualizar estos contenidos en el cliente mientras la información está siendo transmitida por la red, esto quiere decir, que no es necesario descargar completamente el contenido multimedia para comenzar su reproducción, esta se va almacenando en un buffer y se va ejecutando al mismo tiempo que se desarrolla la transmisión” [2].

Los sistemas Streaming funcionan de la siguiente manera: primero, el cliente se conecta con el servidor solicitando algún contenido que se encuentra almacenado allí y con esta solicitud el servidor inicia la transmisión hacia el cliente. Seguidamente, el cliente va recibiendo el contenido y construyendo un buffer en memoria donde se va almacenando la información a medida en que se recibe. Cuando el buffer contiene la mínima parte del contenido, el cliente empieza a reproducirlo mientras se continúa con la descarga, de modo que una vez que el contenido termina de descargarse posiblemente también haya acabado de reproducirse [3]. Este procedimiento depende de algunos factores como los son el ancho de banda, la congestión de la red o las capacidades de recepción del cliente. La Figura 1 muestra el funcionamiento de un servicio de Streaming.

2.1.1. Distribución Multimedia

La distribución de contenido multimedia se puede realizar bajo dos modalidades en general: bajo demanda y en vivo o en directo, en la Tabla 4 se muestran las diferencias entre estas.

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



Fuente: Modificado de *Desarrollo de un Sistema Streaming de Audio Orientado a Dispositivos Móviles*. [3]

Tabla 4. Distribución Multimedia

Distribución Multimedia	
Bajo demanda	En vivo
<ul style="list-style-type: none"> • Permite la reproducción del contenido en el momento que se desee. • El archivo es almacenado en un servidor de Streaming. • El servidor maneja conexiones individuales con las maquinas cliente. • El usuario tiene la posibilidad de controlar el flujo. • El archivo multimedia puede permanecer en el servidor indefinidamente. • El número posible de clientes conectados en un mismo instante depende directamente del ancho de banda disponible en la red. 	<ul style="list-style-type: none"> • La reproducción no se puede acceder en cualquier momento, solo cuando se está transmitiendo. • Transmisión de contenido multimedia en tiempo real. • El cliente debe “escuchar” el canal por el cual fluye el contenido. • Es común el uso de grupos Multicast como destino, siendo el cliente el que demuestra la intención de recibir el flujo.

Autores del Proyecto

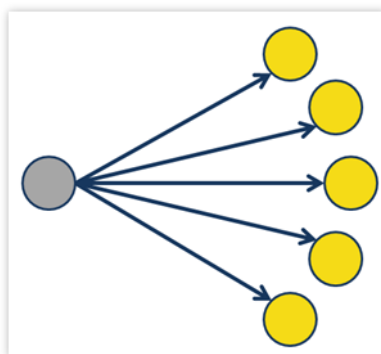
2.1.2. Esquema de Distribución

Existen tres tipos de esquemas en los que se hacen las distribuciones de los archivos en una red como lo son: Broadcast, Unicast y Multicast.

2.1.2.1. 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 la red. La

difusión a través de Broadcasting está limitada a un dominio Broadcast. Generalmente las redes que soportan Broadcast están asociadas a redes de área local. La figura 2 muestra la distribución Broadcast.

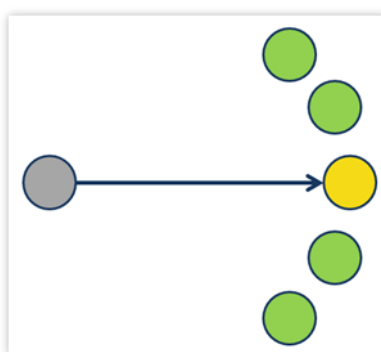
Figura 2. Broadcast



Autores del Proyecto

2.1.2.2. Unicast: En la terminología de redes de computadoras, se dice que una transmisión es Unicast cuando el destino de los paquetes de información es uno solo. Esta transmisión es todo lo contrario a una transmisión Broadcast, cada transmisión Unicast desde el servidor hacia el cliente requiere un flujo independiente. En la Figura 3 se ilustra una distribución Unicast.

Figura 3. Unicast

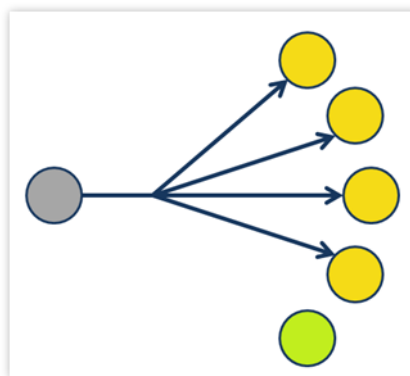


Autores del Proyecto

2.1.2.3. Multicast: 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, este grupo puede estar conformado por uno o más clientes. El flujo de datos es enviado a una dirección que identifica al grupo. En la Figura 4 se puede observar la arquitectura de una distribución Multicast.

Figura 4. Multicast



Autores del Proyecto

2.2. PROTOCOLOS DE STREAMING

La distribución de datos sobre redes está dada por una serie de protocolos a nivel de transporte, entre estos protocolos se encuentran el protocolo TCP y el protocolo UDP. El protocolo TCP fue diseñado para brindar un flujo de bytes confiables de extremo a extremo, esto es posible gracias a la ayuda de temporizadores y a la retransmisión de datagramas mientras sea necesario¹, TCP también desempeña la tarea de reensamblar los paquetes en un orden correcto sin importar el orden de llegada. Por su parte, el protocolo UDP transporta datagramas sin establecer una conexión, debido a esto no proporciona un control de flujo, ni un control de errores o retransmisión, al tener un error en la entrega de algún datagrama.

El protocolo UDP es el más recomendado, junto a RTP y RTSP para ser utilizado en la transmisión mediante la técnica de Streaming. Por otra parte se encuentra el protocolo RTCP el cual gracias a sus mecanismos de control da soporte a RTP que es encargado de transportar los datos.

¹ este mecanismo es mejor conocido como ventana deslizante

2.2.1. Protocolo RTSP (Real Time Streaming Protocol) [8]

RTSP es un protocolo a nivel de aplicación usado para el control remoto a través de la red de los servicios de datos multimedia, es decir, no se encarga del envío de datos sino que establece y mantiene una sesión RTSP con una serie de peticiones que se asocian a un flujo de datos RTP. Para su correcto funcionamiento el protocolo RTSP requiere de una serie de estados y métodos.

2.2.1.1. Estados de RTSP

El protocolo RTSP maneja 4 estados principales (ver Tabla 5).

Tabla 5. Tabla de estados del protocolo RTSP

Estados	Descripción
Init	Es el estado inicial donde aun no se ha recibido ningún mensaje SETUP.
Ready	Es el estado donde el último mensaje SETUP o PAUSE recibido se ha llevado a cabo satisfactoriamente, es decir la respuesta ha sido enviada.
Playing	Es el estado en el cual el dato se está enviando.
Recording	Es el estado donde el servidor está grabando los datos.

Autores del Proyecto

2.2.1.2. Métodos de RTSP

Los métodos del protocolo RTSP se listan a continuación en la Tabla 6.

Tabla 6. Métodos RTSP

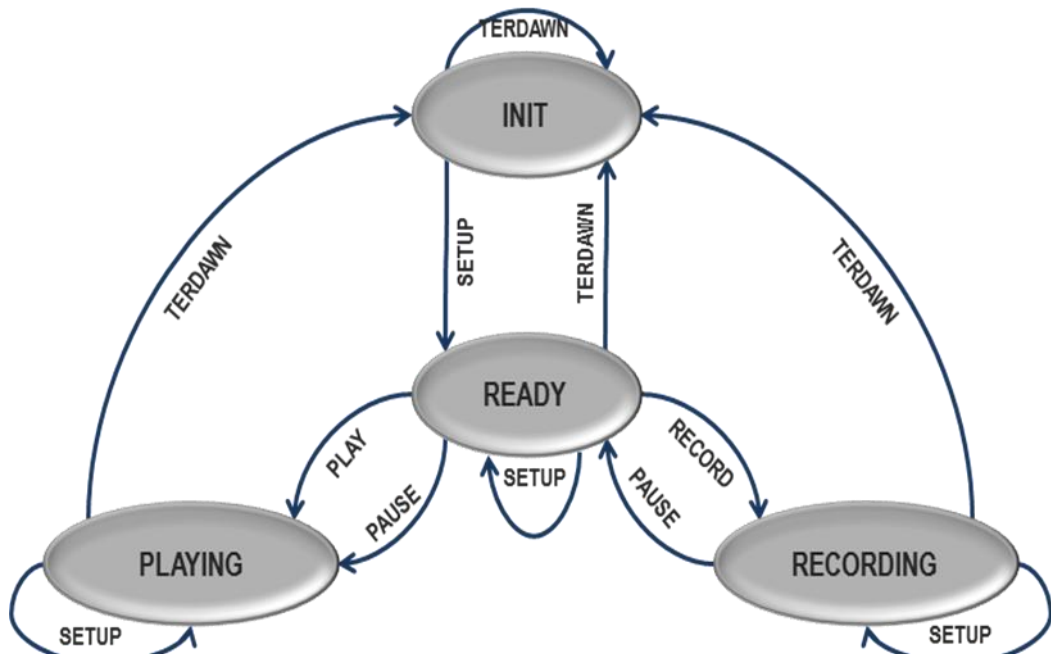
Método	Dirección	Requerimiento	Descripción
DESCRIBE	C→S (*)	Recomendado	Este método obtiene una descripción de una presentación o del objeto multimedia apuntado por una URL RTSP situada en un servidor.
ANNOUNCE	C→S S→C	Opcional	De C→S descripción de una presentación y de S→C anuncia actualizaciones de una sesión.

GET_PARAMETER	C→S S→C	Opcional	Recupera el valor de un parámetro de una presentación.
OPTIONS	C→S S→C	Requerido (S→C: opcional)	Su comportamiento es equivalente a DESCRIBE y podrá ser emitido en cualquier momento.
PAUSE	C→S	Recomen-dado	Detiene temporalmente el flujo de información sin liberar recursos del servidor.
PLAY	C→S	Requerido	Comienza la transmisión de datos mediante un flujo asignado vía SETUP.
RECORD	S→C	Opcional	Comienza la transmisión de datos mediante un flujo asignado vía SETUP.
REDIRECT	C→S	Opcional	Una solicitud de redireccionamiento informa al cliente que debe conectarse a otra ubicación de servidor.
SETUP	C→S	Requerido	Hace que el servidor asigne los recursos para un flujo de datos y para comenzar una sesión de RTSP.
SET_PARAMETER	C→S S→C	Opcional	Solicita ajustar el valor de un parámetro para una presentación o secuencia especificada.
TEARDOWN	C→S	Requerido	Libera recursos asociados con el flujo de información (video y/o audio). La sesión de RTSP deja de existir en el servidor.

(*) C = Cliente y S = Servidor

Fuente: RFC 2326: Real Time Streaming Protocol (RTSP). [8]

Figura 5. Diagrama de estado RTSP



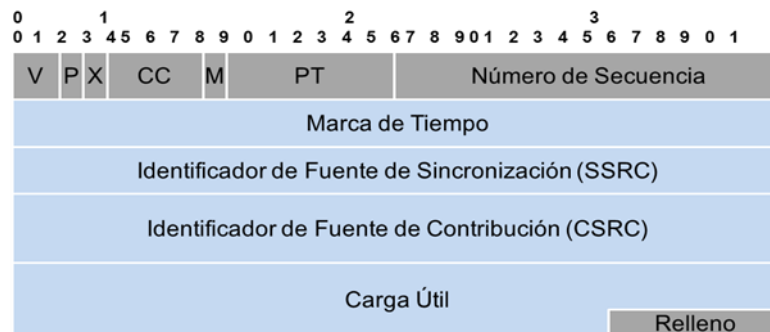
Fuente: Modificado de *Mobile Streaming - Qualidade de Experiência* [9].

El diagrama de estados de RTSP muestra la iteración que estos tienen con los métodos del mismo protocolo (ver Figura 5), donde se puede ver que métodos se requieren para ir de un estado a otro.

2.2.2. Protocolo RTP (Real-time Transpor Protocol)

El protocolo RTP es utilizado para la transmisión de información de archivos multimedia en “tiempo real”. Es decir es el protocolo encargado del transporte de los datos. La estructura de los paquetes RTP se muestra a continuación en la Figura 56.

Figura 6. Estructura del paquete RTP



Fuente: Modificada de *RFC 3550: RTP: A Transport Protocol for Real-Time Applications*. [10]

2.2.2.1. Encabezado del Paquete RTP

En la Tabla 7 se muestra en detalle las funciones de cada campo del encabezado de los paquetes RTP:

Tabla 7. Encabezado RTP

Campo	(bits)	Función [11]
Versión	2	(V) Identifica la versión RTP. La versión actual es 2.
Relleno	1	Indica si los octetos de relleno aparecen al final de la carga útil.
Extensión	1	Si se establece, la cabecera fija viene seguida por exactamente una cabecera de extensión, la cual se emplea para ampliaciones experimentales de RTP.
Cuenta CSRC	4	El número de identificadores CSRC que siguen la cabecera fija.
Marcador	1	La interpretación depende del tipo de carga útil, para audio,

		marca el comienzo de una secuencia de habla.
Tipo de Carga Útil	7	Identifica el formato de carga útil que va a continuación de la cabecera.
Número de Secuencia	16	Cada fuente comienza con un número de secuencia aleatorio que aumenta con cada paquete RTP.
Marca de Tiempo	32	Corresponde al instante de generación del primer octeto de datos de la carga útil.
Identificador de la Fuente de Sincronización SSRC		Un valor generado aleatoriamente que identifica de manera exclusiva cuál es la fuente dentro de una sesión.
Identificador de la fuente de contribución CSRC		Identifica una fuente que contribuye o añade datos a la carga útil.

Autores del Proyecto

2.2.2.2. Funciones de RTP

Para cumplir con los requerimientos del protocolo RTP se cuenta con las funciones mostradas en Tabla 8.

Tabla 8. Funciones RTP

Función	Características
Fragmentación	Cada paquete RTP contiene un número de secuencia empleado para la detección de pérdidas durante el re ensamblado del mensaje en recepción.
Sincronización intramedia	Los paquetes del mismo flujo pueden sufrir retardos diferentes, dando lugar a la aparición del <i>jitter</i> . Para compensarlo las aplicaciones emplean <i>buffers</i> que utilizan las marcas temporales proporcionadas por RTP para medir el <i>jitter</i> .
Identificación del tipo de carga	RTP contiene un identificador del tipo de carga en cada paquete, el cual describe el tipo de codificación que se ha empleado en su generación. Esto se debe a que en algunas ocasiones es deseable cambiar dinámicamente la codificación de la información en la misma medida que varían las condiciones de la red.
Identificación de trama	Las señales de audio y video se envían en unidades lógicas llamadas tramas. Es necesario indicar al receptor el principio y final de cada trama, para tener una sincronización con los niveles superiores, por esto se emplea un bit de marca.
Identificación de fuente	Como RTP fue diseñado para soportar <i>multicast</i> , y en una sección <i>multicast</i> existen varios participantes, se debe poder identificar de algún modo cuál de los participantes generó un determinado paquete. El encargado de esta tarea es el campo SSRC (SynchronizationSource).

Fuente: *Integración de voz y datos*. [12]

RTP trabaja con un formato de paquetes que consta de dos partes. La primera de estas partes es llamada la *cabecera* donde se soportan todas las funciones anteriormente mencionadas y que ayudan al receptor a reconstruir el flujo de Bytes. La segunda parte es llamada la *carga útil*, donde se almacena el flujo de Bytes.

Por lo general RTP se ejecuta sobre el protocolo UDP para hacer uso de sus funciones de multiplexación. La razón de que se elija UDP como protocolo de transporte para RTP en vez de TCP, es debido a la poca flexibilidad de este protocolo, porque para un envío de datos en tiempo real, es más relevante la llegada a tiempo de dichos datos que la fiabilidad de los mismos. En la práctica el protocolo RTP se implementa en el lado del cliente dentro de la aplicación. Cuando se prepara una sesión de RTP, la aplicación define dos direcciones de transporte formadas por una dirección de red y un par de puertos, uno para RTP y otro para el protocolo *Real Time Control Protocol* (RTCP), que se describe más adelante. Debido a que el protocolo RTP no tiene algún mecanismo para garantizar la calidad del servicio, no realiza control de flujo, solo se limita a proporcionar *marcas temporales* que se pueden utilizar en la aplicación receptora para implementar un control de flujo local. Por esta razón RTP necesita la ayuda de otro protocolo que se encargue de proporcionar servicios de control como lo es el protocolo RTCP.

2.3. PROTOCOLO RTCP

RTCP (RTP Control Protocol) se basa en la transmisión periódica de paquetes de control a todos los participantes en una sesión, utilizando el mismo mecanismo de distribución (multidifusión) que los paquetes de datos enviados con RTP. El protocolo que se encuentre por debajo del protocolo RTCP ha de ofrecer la posibilidad de multiplexar paquetes de datos y de control, por ejemplo, por medio de números de puerto UDP diferente.

2.3.1. Funciones Básicas

RTCP posee cuatro funciones básicas según la RFC 3550 [10]:

1. Proveer realimentación sobre la calidad de la distribución de los datos, esta es la parte que se integra al protocolo RTP para que como otros protocolos de transporte se encargue del control de congestión. Los informes de emisión permiten a los receptores calcular las velocidades de transferencia y la calidad de la transmisión y los informes del receptor señalan los problemas encontrados por los receptores, incluyendo las pérdidas de paquetes y el exceso de fluctuación.
2. Llevar un identificador persistente de nivel de transporte de una fuente RTP, llamado *nombre canónico* o *CNAME*, esto es importante debido a que en el transcurso de la comunicación el identificador SSRC puede cambiar si se detecta un conflicto o se reinicia un programa. Con el CNAME, si el identificador SSRC (identificador de la fuente) cambia, se pueden recuperar los participantes de la sesión.
3. Las dos funciones anteriores necesitan que cada participante envíe paquetes RTCP a todos los demás, así, cada participante puede observar independientemente el número total de participantes. Este número sirve para calcular la tasa a la cual se envían los paquetes.
4. Proporcionar de manera opcional una información mínima de control de la sesión.

2.3.2. Tipos de Paquetes RTCP

Existen cinco tipos de paquetes RTCP, los cuales son enviados en una sesión para darle funcionamiento al protocolo (ver Tabla 9).

Tabla 9. Tipos de Paquetes RTCP

Tipo de paquete RTCP		Características
SR (SenderReport)	Informe del Emisor	Trasmitir y recibir estadísticas de los participantes que son emisores activos.
RR (Receiver Report)	Informe del Receptor	Recibir estadísticas de los participantes que no son emisores activos.
SDES (SourceDescription)	Descripción de la Fuente	Elementos de descripción de la fuente, incluyendo el CNAME.
BYE (Goodbye)	Fin de la transmisión	Indica que un participante ha dejado de participar.
APP	Paquete definido por la aplicación	Funcionalidades específicas de la aplicación

Autores del Proyecto

2.3.3. Estructura de Paquetes RTCP

Cada tipo de paquete RTCP comienza con una palabra de 64 bits que contiene los campos indicados en la Tabla 10.

Tabla 10. Cabecera de los paquetes RTCP

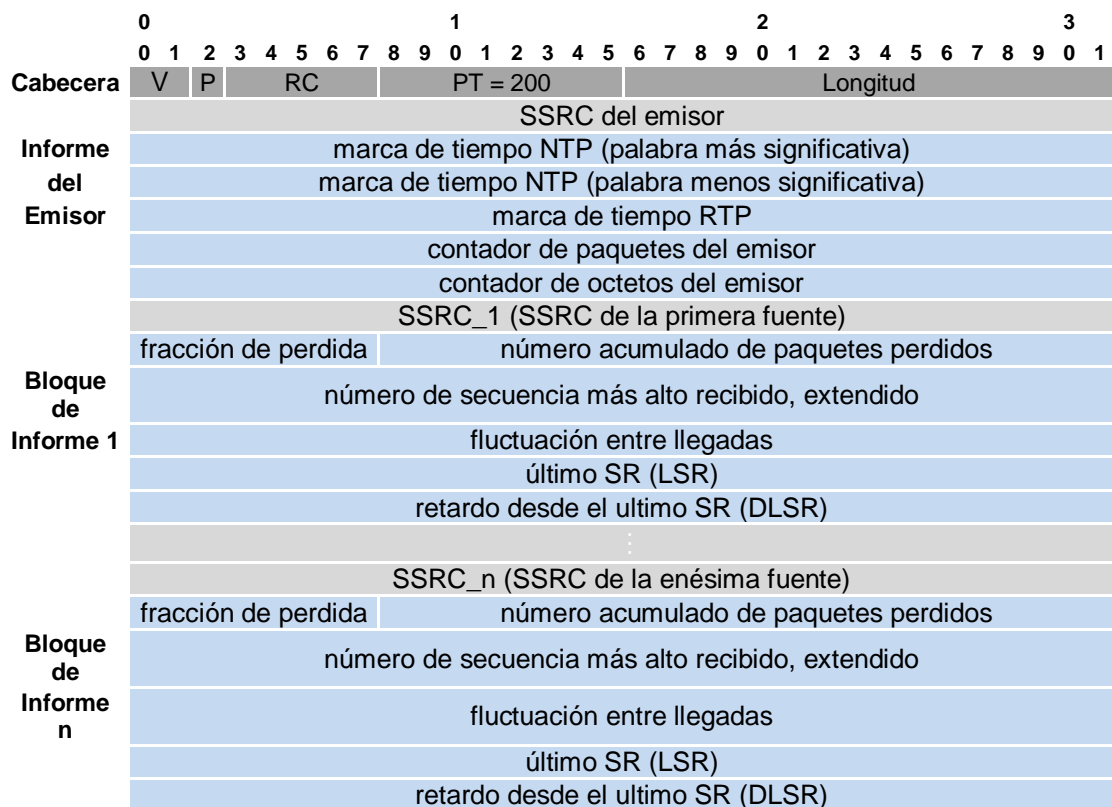
Campo	(bits)	Función [11]
Versión	2	(V) Identifica la versión RTP. La versión actual es 2.
Relleno	1	(P) Si el bit de relleno está activo, este paquete RTCP tiene octetos de relleno que no forman parte de la información de control pero que están incluidos en la longitud del paquete. El último octeto de relleno indica cuántos octetos se deben ignorar, incluido él mismo. En un paquete compuesto, sólo se ha de añadir relleno a los paquetes individuales.
Cuenta	5	(RC) Número de bloques del informe de recepción contenidos en un paquete SR o RR. (SC) Número de elementos de la fuente contenidos en un paquete SDES o BYE. (Sut-tipo) Puede utilizarse para definir un sub-tipo de aplicaciones.
Tipo de Paquete	8	(PT) 200 = SR 201 = RR 202 = SDES 203 = BYE 204 = APP
Longitud	16	(longitud) Longitud del paquete en palabras de 32 bits menos uno, incluyendo cabecera y relleno.
SSRC del emisor	32	Identificador de la fuente de sincronización para el autor de ese paquete. (Para SR y RR)

Autores del Proyecto

2.3.3.1. Formato Paquete SR

Además de la cabecera, los paquetes SR cuentan con un bloque que provee información sobre el emisor y cero o más bloques con el informe de recepción como se muestra en la Figura 7.

Figura 7. Formato del Informe del Emisor (SR)



Fuente: RFC 3550: RTP: A Transport Protocol for Real-Time Applications. [10]

Bloque de información sobre el emisor:

- **Marca de tiempo NTP (64 bits):** El tiempo absoluto en que fue enviado este informe; éste es un número de coma fija sin signo con la parte entera en los primeros 32 bits y la parte fraccionaria en los últimos 32 bits. Éste puede ser utilizado por el emisor junto con las marcas de tiempos devueltas por otros informes del receptor para medir el tiempo de ida y vuelta a estos receptores.

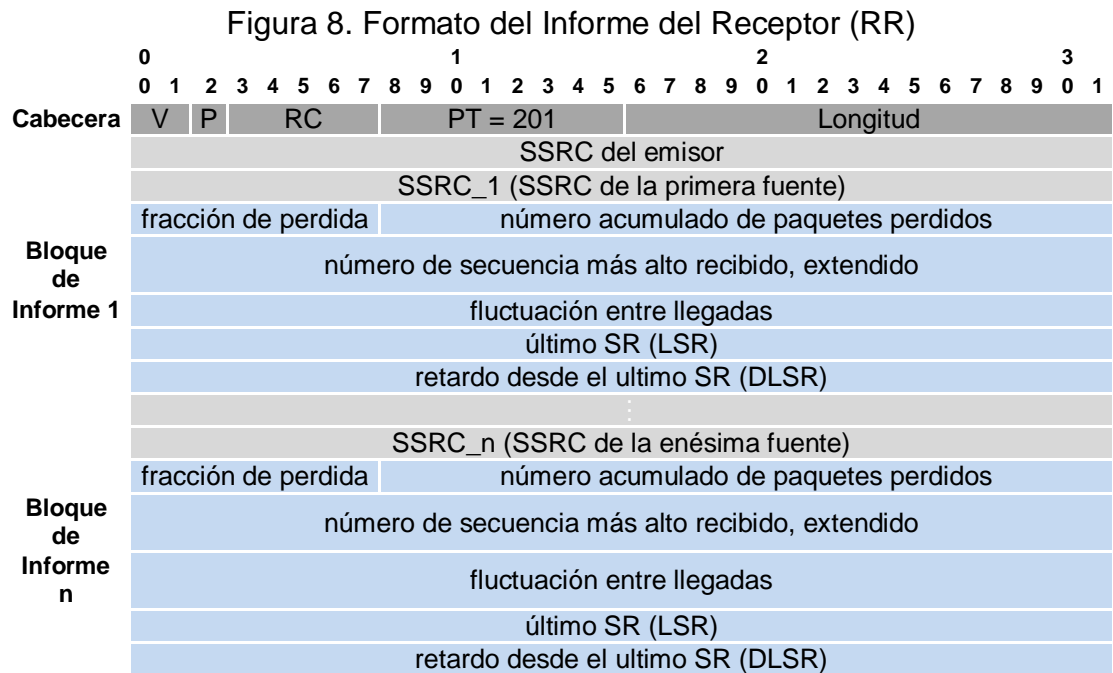
- **Marca de tiempo RTP (32 bits):** Éste es un tiempo relativo que se utiliza para crear marcas de tiempo en los paquetes de datos RTP. Permite a los receptores colocar este informe en la secuencia de tiempo apropiada con los paquetes de datos RTP de su fuente.
- **Contador de paquetes de emisor (32 bits):** Número total de paquetes de datos RTP transmitidos por este emisor hasta este momento en esta sesión.
- **Contador de octetos de emisor (32 bits):** Número total de octetos de carga útil RTP transmitidos por ese emisor hasta este momento en esta sesión.

Bloque de informes de recepción:

- **SSRC_n (32 bits):** Identifica la fuente a la que hace referencia este bloque de informes.
- **Fracción perdida (8 bits):** Fracción de paquetes de datos RTP de SSRC_n perdidos desde que el anterior paquete SR o RR fue enviado.
- **Número acumulado de paquetes perdidos (24 bits):** Número total de paquetes de datos RTP de SSRC_n perdidos durante esta sesión.
- **Número secuencial extendido más alto recibido (32 bits):** Los 16 bits menos significativos registran el número secuencial más alto de datos RTP recibido de SSRC_n. Los 16 bits más significativos registran el número de veces que ha sido puesto a cero el número secuencial.
- **Fluctuación entre llegadas (32 bits):** Cálculo de la fluctuación experimentada por los paquetes de datos RTP provenientes de SSRC_n.
- **Último SR (32 bits):** La mitad de los 64 bits recibidos en la marca de tiempo NTP del último paquete de tipo SR recibido desde la fuente SSRC_n.
- **Retardo desde el último SR (32 bits):** Retraso entre el último SR recibido desde SSRC_n y el envío de este paquete de información.

2.3.3.2. Formato Paquete RR

El formato del informe del receptor (ver Figura 8) es muy parecido al informe del emisor excepto que en el campo *tipo de paquete* tiene un valor diferente (PT = 201) y no hay un bloque con información del emisor.

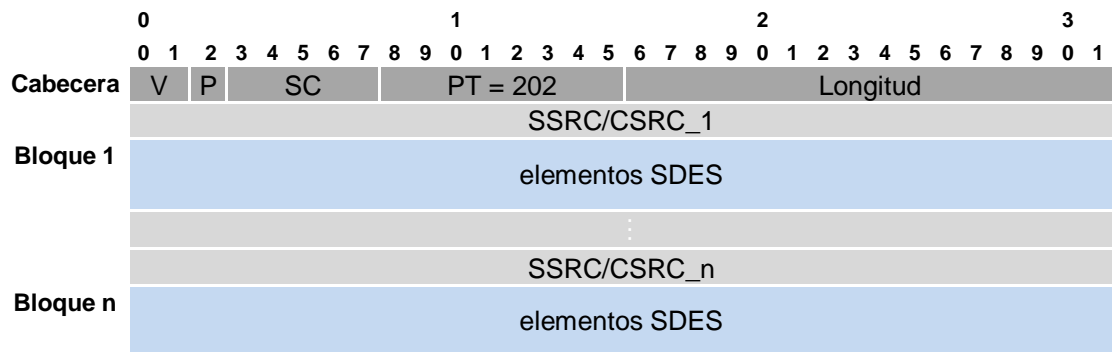


Fuente: *RFC 3550: RTP: A Transport Protocol for Real-Time Applications.*[10]

2.3.3.3. Formato Paquete SDES

El paquete de descripción de la fuente (ver Figura 9) lo usa una fuente para proporcionar más información sobre sí misma. El paquete se compone de una cabecera de 32 bits (ver Tabla 10) seguida por cero o más secciones, cada una de las cuales contiene información sobre la fuente. Cada sección comienza con un identificador de la fuente o de una de las fuentes que contribuyen a la transmisión.

Figura 9. Formato de la Descripción de la Fuente (SDES)



Fuente: RFC 3550: RTP: A Transport Protocol for Real-Time Applications. [10]

Generalmente la descripción se realiza mediante una *lista de elementos descriptivos* (ver Tabla 11), para los cuales, su uso es dependiente de la aplicación.

Tabla 11. Tipos de SDES

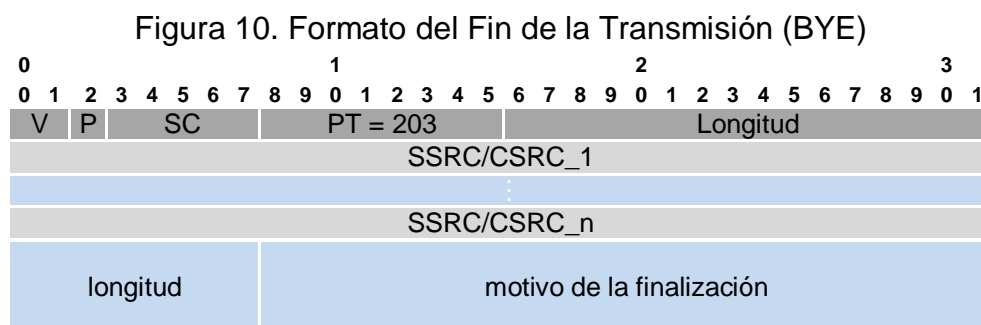
Valor	Nombre	Descripción
0	END	Fin de lista SDES.
1	CNAME	Nombre canónico: único para todos los participantes dentro de una sesión RTP.
2	NAME	Nombre de usuario real de la fuente.
3	EMAIL	Dirección de correo electrónico.
4	PHONE	Número de teléfono.
5	LOC	Ubicación geográfica.
6	TOOL	Nombre de la aplicación que genera el flujo.
7	NOTE	Mensaje transitorio que describe el estado actual de la fuente.
8	PRIV	Extensiones experimentales privadas o específicas de la aplicación.

Fuente: *Redes en Internet de Alta Velocidad. Rendimiento y Calidad del Servicio.* [11]

2.3.3.4. Formato Paquete BYE

El paquete BYE indica que uno o más participantes ya no están activos. Esto confirma a los demás participantes que un silencio prolongado se debe a una salida de un participante más que a un fallo de la red. Si un mezclador recibe

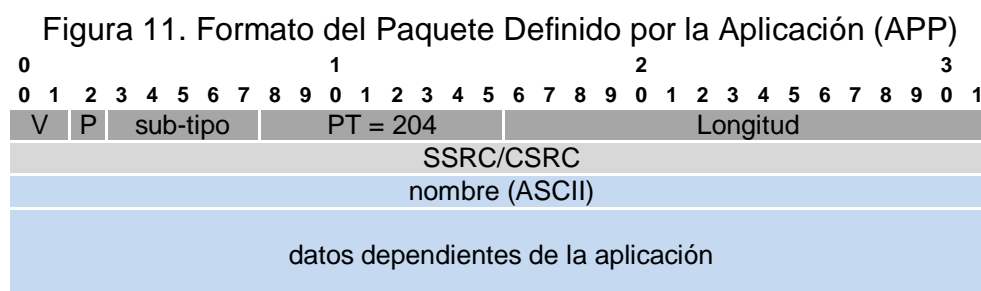
un paquete BYE, éste es enviado junto con la lista de fuentes sin modificarlo. El formato del paquete BYE se compone de una cabecera de 32 bits seguida de uno o más identificadores de fuente (ver Figura 10). Opcionalmente, el paquete puede incluir una descripción textual de la razón por la que se abandona la transmisión.



Fuente: RFC 3550: RTP: A Transport Protocol for Real-Time Applications. [10]

2.3.3.5. Formato Paquete APP

Este paquete tiene un uso experimental para funciones y características que son específicas de la aplicación. Por último, a un tipo de paquete experimental que demuestre ser una utilidad general, se le puede asignar un número de *tipo de paquete* y llegar a formar parte del RTCP estándar (ver Figura 11) [13].



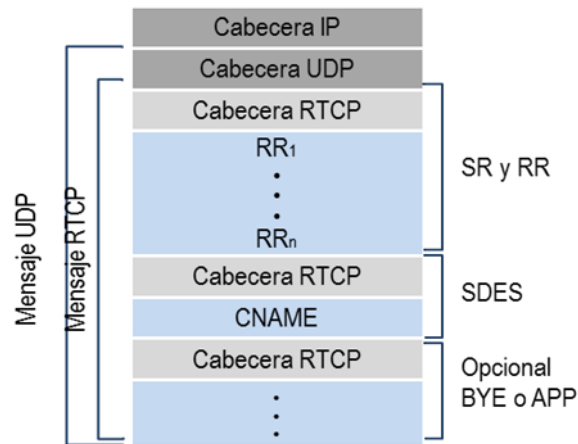
Fuente: RFC 3550: RTP: A Transport Protocol for Real-Time Applications. [10]

2.3.4. Paquetes Compuestos

Todos los paquetes RTCP se deben enviar obligatoriamente en paquetes compuestos de al menos dos paquetes individuales, con el formato siguiente [13]:

- *Prefijo de cifrado*: Si y sólo si el paquete compuesto debe ser encriptado, entonces se debe poner delante un prefijo aleatorio de 32 bits, que se ha de recalcular para cada paquete compuesto. Si es necesario relleno, se debe poner en el último paquete del paquete compuesto.
- *SR o RR*: El primer paquete del paquete compuesto debe ser de tipo informe, para facilitar la validación de la cabecera. Esto se tiene que hacer siempre, aunque no se hayan enviado o no se hayan recibido datos, en este caso se debe enviar un paquete RR vacío. E incluso si el otro paquete en el paquete compuesto RTCP es un BYE.
- *RR adicionales*: Si el número de fuentes de las cuales se debe informar es mayor que 31, entonces se han de enviar tantos paquetes RR adicionales hasta completar los datos del primer paquete SR o RR enviado.
- *SDES*: Un paquete de tipo SDES que contenga un CNAME se ha de incluir siempre en el paquete compuesto. Otros elementos de descripción de la fuente, opcionalmente se pueden incluir si son requeridos por la aplicación en particular.
- *BYE o APP*: El resto de paquetes puede aparecer en cualquier orden e incluso repetidos, excepto el tipo BYE, que debe ser el último paquete enviado con un determinado SSRC/CSRC.

Figura 12. Estructura de un Paquete Compuesto RTCP



Fuente: Modificado de *IP Communications and Services for NGN*. [14]

2.3.5. Restricciones

Para llevar a cabo las funciones del protocolo RTCP tienen en cuenta las siguientes restricciones [13]:

- La recepción de estadísticas (en SR o RR) se debe enviar tan frecuentemente como lo permitan las restricciones de ancho de banda para maximizar la resolución de las estadísticas. Por ello cada paquete RTCP compuesto debe llevar obligatoriamente un paquete de tipo informe.
- Los nuevos receptores deben recibir el CNAME tan pronto como sea posible para identificar la fuente y poder asociar el contenido multimedia.
- El número de tipos de paquetes que pueden aparecer por primera vez en el paquete compuesto se debe limitar con el fin de aumentar el número de bits constantes en la primera palabra (cabecera) del paquete y la probabilidad de validar con éxito los paquetes RTCP contra paquetes de datos RTP que tengan la dirección mal o que no estén relacionados.

2.4. SERVIDORES DE STREAMING

Existen varios tipos de servidores, sin embargo para la transmisión de servicios de Streaming, se usan los siguientes:

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

No obstante, los servidores de Streaming son los apropiados para la implementación de la técnica. Debido a que en la transmisión de archivos de audio y video el volumen de información que debe transmitirse aumenta linealmente con el tiempo, es importante que el servidor que soporta esta transmisión esté diseñado para afrontar estos requisitos. En un servidor Streaming, la transmisión de datos multimedia se puede lograr de una manera más rápida. Además, con estos servidores se pueden ofrecer servicios de control de flujo (reproducir, pausar,...) y soportar multidifusión. Las diferencias entre los servidores de Streaming existentes en el mercado, se fundamentan principalmente en el soporte de los protocolos de la técnica y de los códecs de audio y video.

2.5. REDES MÓVILES

Las redes móviles son un derivado del proceso evolutivo de las telecomunicaciones², en donde la búsqueda de la integración de las redes de voz y las redes de datos ha sido el principal objetivo.

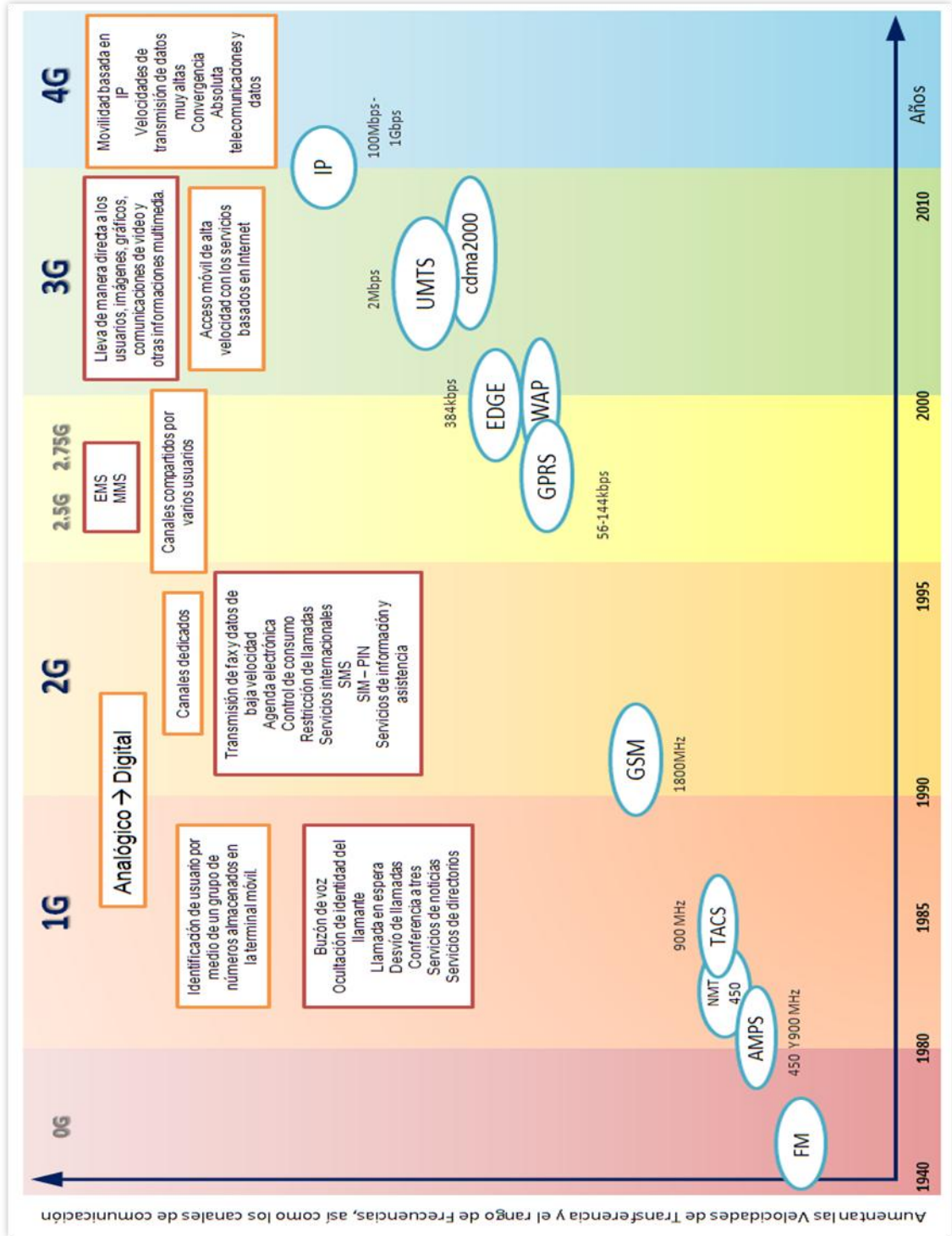
2.5.1. Redes de Telefonía Móvil

Los sistemas de telefonía móvil han venido avanzando a pasos agigantados y muestra de esto es la evolución generacional que se ha dado desde sus inicios

² Telecomunicación. Si la comunicación se realiza a distancia, empleando medios eléctricos, ópticos o de cualquier otro tipo, utilizando para ello hilos metálicos, radio, fibra óptica, microondas, satélites, etc.

en la década de los 40, pasando por apariciones de tecnologías y sistemas muy importantes, poniendo a la telefonía móvil en la mira de la evolución tecnológica en todo el planeta (ver Figura 13).

Figura 13. Generaciones de la Telefonía Móvil



Autores del Proyecto

Hoy en día, se aceptan cuatro generaciones principales en la historia de la telefonía móvil celular (1G, 2G, 3G y 4G). Las principales características de cada una de las generaciones de la telefonía móvil se presentan en la Tabla 12, allí se destacan las tecnologías y las redes móviles que sobresalieron en cada generación.

Tabla 12. Generaciones, tecnologías y redes de la telefonía móvil

Generación	Principales Tecnologías y Redes	País(es) de Origen y/o Entidad(es) Gestora(s) [Fecha]	Características
Primera Generación 1G [15]	NMT (<i>Nordiska Mobile Telphongruppen</i>)	Telecomunicaciones Escandinavas [1981]	<ul style="list-style-type: none"> - Redes móviles analógicas o semianalógicas (trayecto de radio analógico pero conmutación digital). - Interfaz de Radio. - Se centran en los servicios de voz - Operan en la banda de 450 y 900 MHz. - Desarrollados a escala nacional. - Por su naturaleza regional no eran compatibles entre sí. - Identificación del usuario realizada mediante un grupo de números almacenados en la memoria del terminal móvil. - Un solo canal de comunicación.
	AMPS (<i>Advanced Mobile Phone System</i>)	Estados Unidos [1983]	
	TACS (<i>Total Access Communications System</i>)	Reino Unido [1985]	
Segunda Generación 2G	GSM (<i>Global System for Mobile Communications</i>)	CEPT (<i>Conference Européenne Postes et Telecommunications</i>) [1992]	<ul style="list-style-type: none"> - Se basan en tecnologías de acceso digital. - Cuenta con toda una arquitectura de red y no solo con una interfaz de radio. - La relación entre el usuario y el operador se realiza a través de una tarjeta personal (SIM), activada mediante un número de identificación personal (PIN). - Servicios de datos y de fax. - Se empieza a permitir que un mismo usuario utilice simultáneamente varios canales. No siendo necesaria la ocupación continua de los canales de forma permanente y exclusiva. - Acceso a Internet.
	GPRS (<i>General Packet Radio System</i>)	ETSI (<i>European Telecommunications Standards Institute</i>)	
	EDGE (<i>Enhanced Data Rates for GSM Evolution</i>) o EGPRS (<i>Enhanced GPRS</i>)	3GPP (3rd Generation Partnership Project) [2000]	
	WAP (<i>Wireless Application Protocol</i>)	Forum WAP: Ericsson, Motorola, Nokia y Phone.com [1997]	
Tercera Generación 3G	UMTS (<i>Universal Mobile Telecommunications System</i>) o IMT-2000	Europa y Japón UIT (International Telecommunication Union) [2004]	<ul style="list-style-type: none"> - Servicios Multimedia. - Comunicaciones de vídeo. - Se introduce el tema de movilidad de los servicios, además de la movilidad del terminal personal. - Integran todos los servicios ofrecidos por las distintas tecnologías y redes actuales. - Soporta personalización por el usuario. - Combina el acceso móvil de alta velocidad con los servicios basados en Internet.
	CDMA2000 (<i>Code Division Multiple Access 2000</i>)	Estados Unidos 3GPP: ETSI(Europa), ARIB y TCC(Japón), TTA(Corea) y ANSI T1(USA)	

Cuarta Generación 4G [16] [17]	WiMax(World wide Interoperability for Microwave Access) y LTE (Long Term Evolution), son candidatas firmes para formar parte de esta generación	ITU	<ul style="list-style-type: none"> - Se encuentra en su proceso de especificación y versiones prueba. - Basada totalmente sobre IP. - Alta usabilidad (en cualquier momento, lugar y tecnología) - Velocidades de transmisión alta y de bajo costo. - Prestación de servicios integrados. - Convergencia absoluta de telecomunicaciones y datos.
--------------------------------	---	-----	--

Autores del Proyecto

2.5.2. Arquitectura de Red para el Sistema Streaming

Para que se realice la conexión entre un cliente del sistema y el servidor de Streaming, el cliente debe conectarse a internet mediante una red móvil celular o a través de una conexión inalámbrica con el fin de poder reproducir mediante Streaming un archivo que se aloje en el servidor (ver Figura 14). A continuación se muestran los tipos de conexión que se pueden establecer entre el dispositivo cliente y el servidor de Streaming:

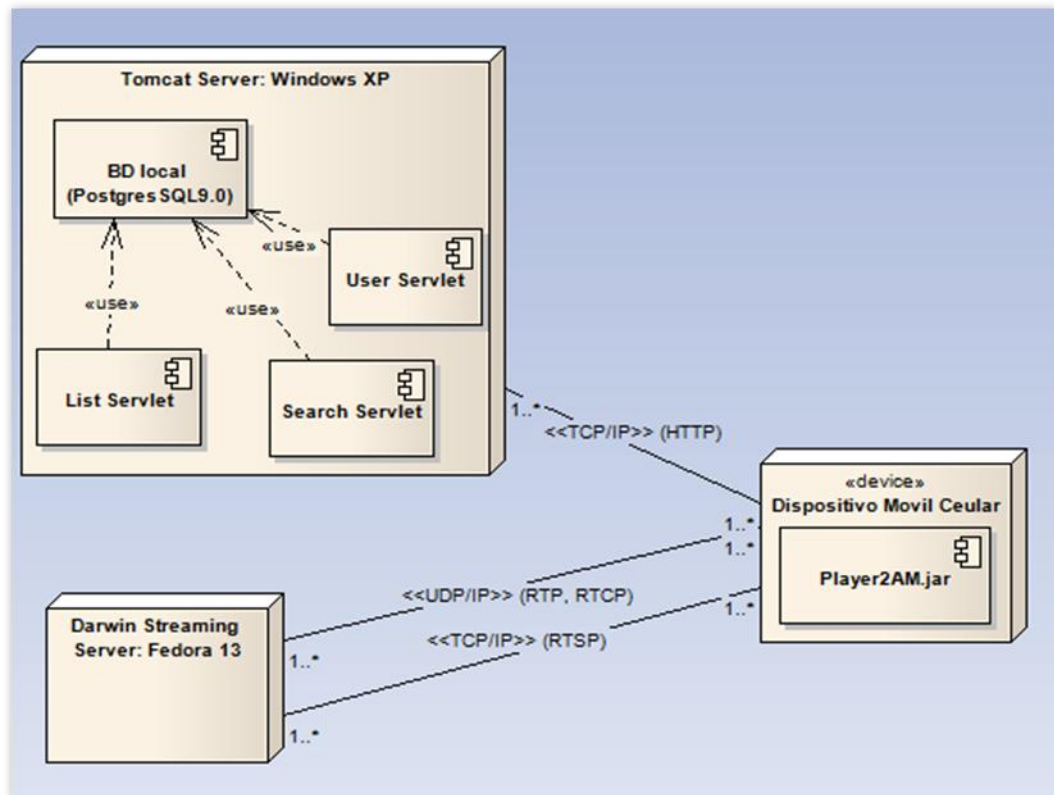
2.5.2.1. Conexión a internet por red móvil celular: Esta conexión realizada inicialmente con el protocolo *Wireless Application Protocol* (WAP) y más recientemente con tecnologías de 3G, permiten que esta red celular sea utilizada para el uso de aplicaciones móviles que necesiten Internet. Esta conexión implica tener asociado a la línea celular un plan o paquete de datos, servicio prestado por un operador de telefonía celular.

2.5.2.2. Conexión a internet por Wi-Fi: Wi-Fi (*Wireless Fidelity*) es el nombre coloquial de la familia de estándares IEEE 802.11 para redes locales inalámbricas (WLAN) que soportan el estándar IEEE 802.11x. La tecnología Wi-Fi fue diseñada como una alternativa a la conexión estándar LAN usando un cable de red.

Hoy en día, varios dispositivos móviles celulares tienen incluida la tecnología Wi-Fi, actuando como dispositivo terminal, lo que les permite acceder a Internet sin necesidad de hacerlo por tecnologías como WAP o 3G. De esta forma, el

dispositivo solo necesita conectarse con el dispositivo de distribución de red Wi-Fi.

Figura 14. Diagrama de Despliegue: Arquitectura de Red del Sistema Streaming



Autores del Proyecto

2.6. CÓDECS DE AUDIO

Un *códec de audio* es el hardware y/o software capaz de codificar y decodificar los datos auditivos, mientras estos son almacenados en un tipo de *formato de audio* específico.

Existen tres tipos de formatos de audio: *Formatos de audio sin comprimir*, *formatos con pérdida de datos* y *formatos sin pérdidas de datos*. Los formatos que permiten pérdidas de datos en general aprovechan las limitaciones del sistema auditivo eliminando datos redundantes de la fuente de datos.

Por otra parte, existen algunos formatos que se catalogan como *formatos contenedores*, estos albergan los archivos de audio, video, texto, que están comprimidos con un códec específico.

Los códecs de audio también puede ser clasificados en abiertos o cerrados. Los códecs abiertos son aquellos a los cuales cualquier usuario puede tener acceso a sus algoritmos de codificación, mientras que los códecs cerrados poseen restricciones y son mejores conocidos como códecs propietarios.

2.6.1. Familia de Códecs de audio AMR

La familia de códecs AMR está basada en la tecnología ACELP (*Algebraic Code Exited Linear Prediction*), siendo el códec AMR-WB+ la última versión.

A continuación se indica el proceso que un archivo de audio tiene al ser codificado por un códec de la familia AMR. Primero la señal analógica es convertida en digital para ser transmitida y luego en el reproductor cuando vuelve a ser analógica suena mejor con ACELP que sin su utilización. ACELP³ es un sistema de compresión de voz que se utiliza para proporcionar un nivel de calidad de voz apropiado para cuando la red está operando a bajas tasas de datos (ancho de banda estrecha). [18]

Por otra parte en servicios de transmisión de Streaming en dispositivos móviles se requiere que los códecs tengan ciertas características:

- Baja tasa de bits para que las velocidades de transmisión aumenten, pero teniendo calidad de audio razonable.
- Frecuencia de muestreo pequeñas que tomen pocos detalles pero sin afectar notablemente la calidad para el oído humano.

³ Algebraic Code Exited Linear Prediction

La Tabla 13 muestra una comparación de las principales características de codificación y el propósito de los códecs miembros de la familia AMR.

Tabla 13. Características de la Familia de Códecs AMR

Características	AMR (AdaptativeMuti-Rate)	AMR-WB (AMR Wideband)	AMR-WB+ (Extended AMR-WB)
Origen	- Fue introducido por la 3GPP en 1998.	- AMR-WB fue estandarizado por la ETSI/3GPP en diciembre de 2000 y seleccionado y aprobado por la ITU-T en julio de 2001 y enero de 2002 respectivamente.	- Fue adoptado como un estándar de codificación de audio por la 3GPP en septiembre de 2004.
Técnicas usadas y detalles de codificación	- Técnica de codificación de velocidad de bits variable, para ajustar la velocidad de envío a condiciones determinadas.	- ACELP	- Es una extensión de AMR-WB, que utiliza un híbrido de 2 tecnologías, ACELP y TCX (Transform Codec Exitacion), ofreciendo una calidad muy alta para voz, música, voz entre la música y voz sobre la música. - Tiene un amplio rango amplio de tasas de bits, tasas de mono y estéreo escalables.
Uso	- Uso general para codificación voz durante una llamada. - Fue adoptado por la 3GPP como el códec obligatorio para los sistemas de 3 ^o generación inalámbricos basados en el núcleo de red GSM evolucionada (WCDMA, EDGE, GPRS).	- Este códec proporciona excelente calidad de voz, gracias a su ancho de banda de voz más amplio, mejorando significativamente la naturalidad del habla. - Con la incursión de los teléfonos de 3G se utiliza para la codificación de las grabaciones de voz, tonos de llamada y streaming. [7]	- Está diseñado para manejar voz y música a muy bajas tasas de bits. - Su reproducción de ancho de banda acerca a la calidad de CD. - Proporciona compatibilidad con AMR-WB. [18]

Autores del Proyecto

2.6.2. Comparación de Códecs de Audio

Dentro de las características más importante a tener en cuenta para comparar los códecs de audio en un sistema de Streaming para dispositivos móviles celulares se encuentran las siguientes:

- Las redes de comunicaciones móviles tienen una capacidad limitada, esto hace que sea importante que los códecs tengan tasas de bits muy bajas para generar audio que posean una calidad aceptable.
- Los códecs deben ser capaces de ser soportados por un servidor de streaming.
- Los códecs deben ser capaces de ser integrados en un *Sistema Embebido*⁴ como un teléfono móvil celular de manera fácil. [7]

La Tabla 14 presenta los parámetros más importantes de los códecs de la familia AMR que se tuvieron en cuenta en la comparación.

Tabla 14. Comparación de la familia de Códecs AMR [7] [18]

Variables	AMR	AMR-WB	AMR-WB+
Tipo de Tasa de Bits	VBR	VBR	VBR
Tasa de Bits	12.20kbps – 1.80kbps	23.85kbps – 6.6kbps	48 kbps – 5.2 kbps
Frecuencia de Muestreo	8 kHz	12.8kHz - 16kHz	12.8kHz – 48kHz
Extensión o formato (.*)	.amr	.awb	.wb+
Tipo MIME	audio/amr audio/3gpp audio/3gpp2	audio/amr-wb audio/3gpp	audio/amr-wb+ audio/3gpp

Autores del Proyecto

2.7. DESARROLLO DE APLICACIONES MÓVILES CELULARES

El desarrollo de aplicaciones móviles celulares es bastante amplio, éste depende del tipo de dispositivo y/o del sistema operativo (SO) que posea el dispositivo en particular. Algunas marcas de dispositivos móviles celulares tienen su propio SO como los dispositivos Iphone y BlackBerry cuyos sistemas operativos son iOS y Blackberry OS respectivamente. Otros, por su parte

⁴ dispositivo de propósito específico

utilizan un SO que sirve para diferentes marcas, entre esos se encuentran los SO Android y Symbian.

Muchas de las aplicaciones para dispositivos móviles pueden ser desarrolladas en el lenguaje JAVA, utilizando la plataforma Java Micro Edition (JME). El único requisito es que el dispositivo posea una Máquina Virtual de JAVA (JVM).

2.7.1. JAVA

La tecnología Java comprende el lenguaje de programación y una plataforma (Máquina Virtual de Java). Como lenguaje, Java es simple, orientado a objetos, distribuido, interpretado, robusto, seguro, independiente de las arquitecturas de hardware, portable, potente, multitarea y dinámico.

La plataforma Java está compuesta por tres ramas (ver Tabla 15) en función del sector al que va orientada la edición correspondiente.

Tabla 15. Plataformas JAVA

PLATAFORMA	SECTOR
JSE (Java Standard Edition)	Orientado a ordenadores de escritorio.
JEE (Java Enterprise Edition)	Es JSE añadiéndole clases para el desarrollo en entornos corporativos.
JME (Java Micro Edition)	Es un subconjunto de JSE orientado al desarrollo de aplicaciones Java destinadas a dispositivos de pocos recursos, como los dispositivos móviles celulares

Autores del proyecto

2.7.2. Java Micro Edition (JME)

J2ME es un subconjunto de J2SE orientado al desarrollo de aplicaciones Java destinadas a dispositivos con pocos recursos, con capacidades restringidas,

tanto con respecto a la capacidad de memoria disponible, limitaciones de la pantalla gráfica como con respecto a la capacidad de procesamiento [19].

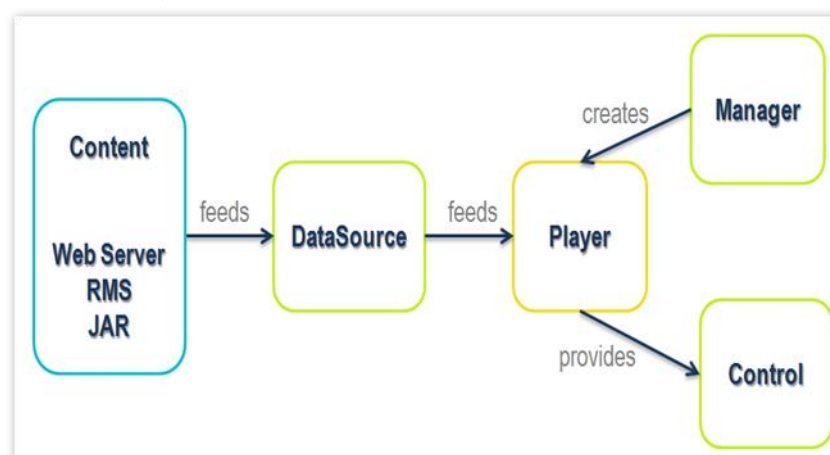
2.7.3. Mobile Media API (MMAPI)

El MMAPI es uno de los paquetes opcionales para JME, permitiendo el manejo de archivos multimedia como es reproducir y grabar datos de audio y/o video entre otras utilidades. Para definir la arquitectura del API multimedia es necesario conocer que su funcionamiento está basado en los siguientes conceptos:

- **Reproductor** (*Player*): Reproduce archivos multimedia concretos. Por ejemplo se necesita uno para reproducir audio y otros para tonos simples.
- **Controlador** (*Control*): Utilizado para modificar el comportamiento de un reproductor.
- **Gestor** (*Manager*): Es el objeto que cohesiona todos los anteriores y sirve como punto de entrada al sistema ya que dispone de métodos estáticos que permiten crear reproductores y/o fuentes de datos.

En la Figura 15 se ilustra la arquitectura del API multimedia de JME.

Figura 15. Arquitectura del API Multimedia



Fuente: Modificado de *The J2ME Mobile Media API*. [20]

Es importante mencionar que el objeto Player consta de un ciclo de vida, el cual permite tener un control sobre este, con el fin de no consumir recursos innecesarios. Este ciclo consta de 5 estados (ver Tabla 16) [21].

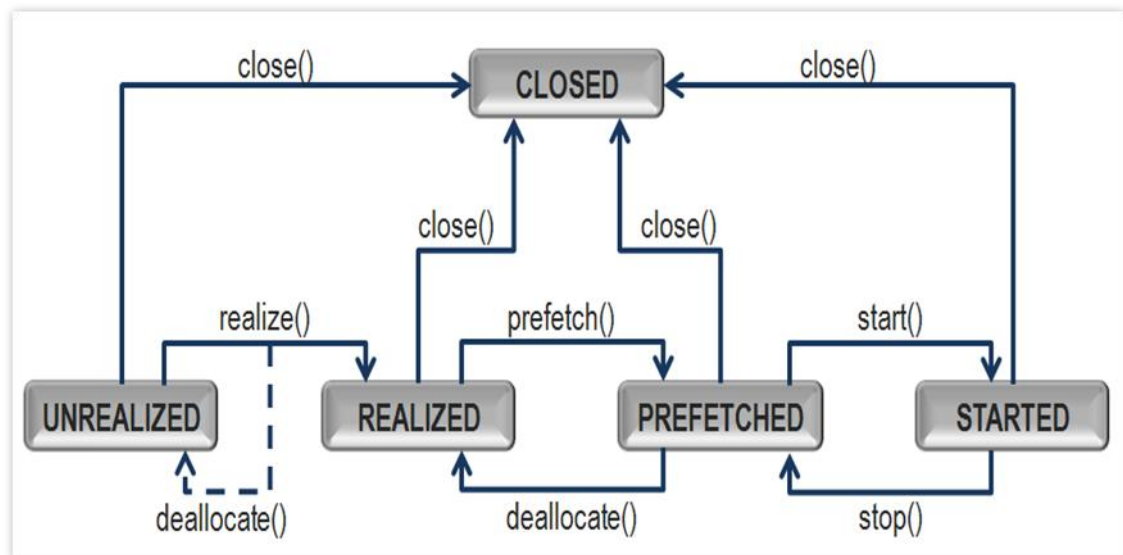
Tabla 16. Estados el Ciclo del Vida del Objeto Player

Estado	Descripción
UNREALIZED	Estado inicial del reproductor, no dispone información necesaria para la iniciar la reproducción.
REALIZED	Estado donde empieza obtener recursos.
PREFETCHED	Contiene los recursos necesarios para empezar la reproducción.
STARTED	Estado donde se está reproduciendo.
CLOSED	Libera los recursos de la reproducción.

Autores del Proyecto

El ciclo de vida ocurrido en el objeto de la clase Player se puede observar en la Figura 16.

Figura 16. Ciclo de vida de un objeto Player



Fuente: Modificado de *Multimedia en MIDP 2.0*. [21]

3. METODOLOGÍA Y REQUERIMIENTOS

Para el desarrollo de este trabajo de grado se escogió la metodología ágil denominada *eXtreme Programming* (XP). La metodología XP es una metodología de desarrollo de software de gran aceptación dado que se enfoca en la satisfacción del cliente y hace énfasis en el trabajo en equipo. Pero su mayor fortaleza radica en la adaptación al cambio de requerimientos. XP se fundamenta en 5 reglas simples que enmarcan las actividades dentro del desarrollo: *Planeación, Administración, Diseño, Codificación y Pruebas*.

3.1. REQUERIMIENTOS

Se plantean una serie de requerimientos funcionales y no funcionales de la aplicación, para cumplirlos se establecieron Historias de Usuarios (HU) según como lo sugiere la metodología, los requerimientos se listan a continuación:

3.1.1. Requerimientos Funcionales

- Soporte de los protocolos RTSP, RTP y RTCP en cliente y servidor.
- Reproducción de archivos de audio en el cliente.
- La aplicación debe permitir realizar búsquedas de los archivos de audio que se quieran oír.
- La aplicación debe permitir tener un control de la reproducción (pausar, parar, cambiar de audio)
- La aplicación debe permitir el manejo del volumen del audio que se encuentre en reproducción.
- La aplicación debe permitir guardar listas con información de los archivos de audio para posteriores reproducciones.
- La aplicación debe permitir el registro e inicio de sesión de usuarios.
- La aplicación debe permitir a los usuarios registrados modificar su información de registro y recordar su contraseña.
- El cliente y servidor deben soportar el códec AMR-WB.

- La aplicación debe suministrar en pantalla información del archivo en reproducción.

3.1.2. Requerimientos No Funcionales

- La aplicación debe ser desarrollada en JME.
- El cliente debe tener soporte de Java.
- El sistema debe tener archivos de audio alojados en un servidor.
- El cliente debe tener acceso a internet.
- El sistema debe tener información de los archivos de audio en una base de datos.

3.2. HERRAMIENTAS UTILIZADAS

La Tabla 17 expone todas las herramientas utilizadas para llevar a cabo el presente proyecto de investigación.

Tabla 17. Herramientas Utilizadas

Tipo	Herramienta
Servidor de Streaming	Darwin Streaming Server 5.5 ⁵
Servidor Web	Tomcat5.5 ⁶
Sistemas Operativos	Windows XP, Fedora 13
Lenguajes de Programación	Java (JME) y SQL
Ambiente de Desarrollo Integrado	Netbeans 7.0.1 ⁷
Motor de Base de Datos	PostgreSQL
Emuladores de Dispositivos Móviles Celulares	Java Micro Edition SDK 3.0, Sony Ericsson SDK 25.5.0.4, Samsun SDK 1.2.1, Motoved SDK 3.x, Nokia SDK 1.0 Sun Java™ WirelessToolking 2.5.2
Dispositivo Móvil Celular	Nokia X3-02
Formato Contenedor	MOV
Códecs	AMR, AMR-WB

⁵ Página Oficial. The Apache Software Foundation. [online] Disponible en: <http://tomcat.apache.org/>

⁶ Página de Descarga. [online] Disponible en: <http://dss.macosforge.org/>

⁷ Página Oficial. [online] Disponible en: <http://netbeans.org/>

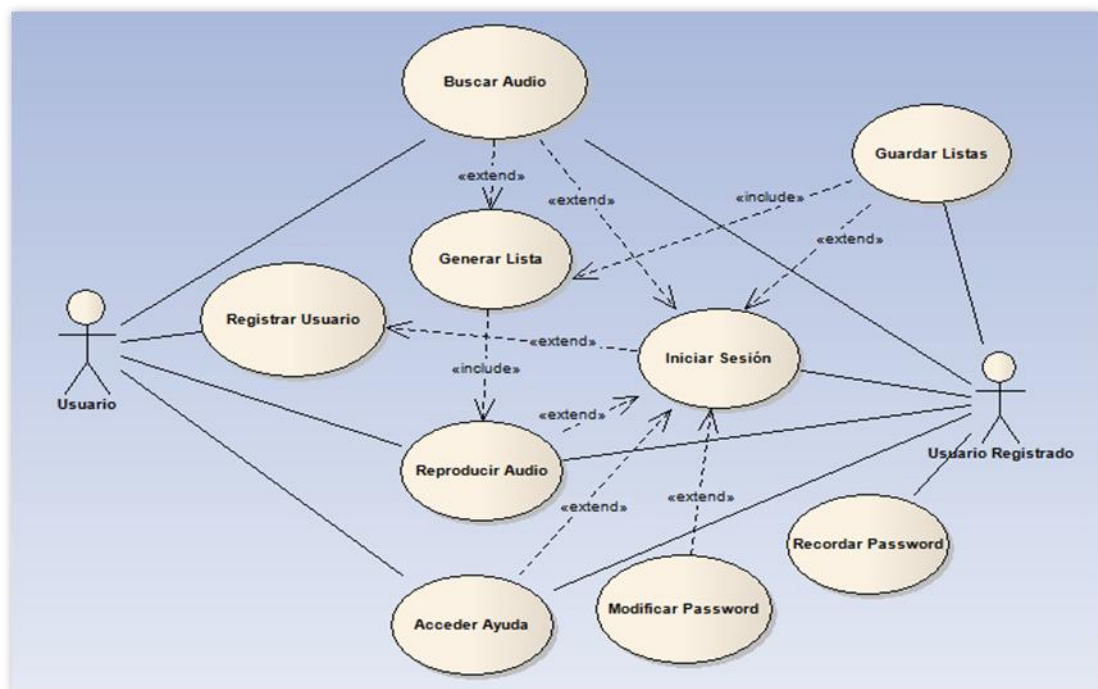
Monitor de Red	WIRESHARK ⁸
Administración de Proyecto	ClockingIT ⁹
Sistema de Control de Versiones	GitHub ¹⁰
Librerías	LWUIT, MPEG4IP, POSTGRES, JMUNIT, SERVLET
Diseño UML	Enterprise Architect ¹¹

Autores del Proyecto

3.3. CASOS DE USO

Utilizando la herramienta *Enterprise Architect* (ver Tabla 17) se llevó a cabo el diseño de los casos de uso (ver Figura 17), los cuales describen el funcionamiento de la aplicación móvil para transmisión de Streaming de audio.

Figura 17. Casos de Uso de la Aplicación



Autores del Proyecto

La descripción de los casos de uso se encuentra en el Anexo A

⁸ Página Oficial. [online] Disponible en: <http://www.wireshark.org/>

⁹ Página Oficial: [online] Disponible en: <http://www.clockingit.com/>

¹⁰ Página Oficial: [online] Disponible en: <https://github.com/>

¹¹ Página de Descarga. [online] Disponible en: <http://www.sparxsystems.com.au/>

4. DESARROLLO DEL PROYECTO

4.1. REVISIÓN E IMPLEMENTACIÓN DE LOS PROTOCOLOS RTSP Y RTP

Como parte importante de esta investigación se propone la implementación de los protocolos de aplicación RTP, RTSP y RTCP como soporte necesario para la realización de una aplicación que reproduce audio mediante la técnica de Streaming. A continuación se da una descripción de los detalles de la implementación de cada protocolo, teniendo en cuenta la implementación de RTP y RTSP llevada a cabo en el proyecto de grado realizado anteriormente y haciendo un mayor énfasis en la implementación del protocolo de control RTCP, siendo este la base fundamental de la realización de esta nueva investigación.

4.1.1. Implementación de RTSP

La implementación del protocolo RTSP se fundamenta principalmente en el uso de sus métodos descritos en la sección 2.2.1.2, los cuales permiten tener un control de la reproducción de los archivos de audio desde el cliente.

La comunicación RTSP es iniciada por la aplicación en el cliente, éste emite un mensaje de petición al servidor DSS, esta petición es de tipo *Describe* con la cual se busca obtener la información necesaria del archivo a reproducir; el servidor responderá con un código de aprobación y la descripción del archivo en un formato SDP. Al recibir la respuesta del servidor se emite una petición de tipo *Setup*, la cual 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 la transmisión (UDP, TCP, HTTP), el modo de transmisión (Unicast o Multicast) y los números de puertos del cliente a los cuales debe enviar el flujo de datos, como respuesta el servidor indicara el número de sesión y los puertos por los cuales el servidor enviara este flujo.

Para empezar la transmisión el cliente envía una petición de tipo *Play* que debe llevar el número de sesión indicado en la respuesta *Setup*. Al recibir esta petición el servidor DSS empieza la transmisión del archivo de audio. Para finalizar, el cliente puede enviar una petición de tipo *Teardown* para liberar los recursos del servidor y ponerlo en disposición para que pueda tomar los recursos de un nuevo archivo. En la Figura 18 se muestra la iteración que ocurre entre el servidor y el cliente cuando se comunican con los mensajes correspondientes a los métodos RTSP; los mensajes fueron registrados con la aplicación WIRESHARK¹².

Figura 18. Métodos RTSP en la aplicación WIRESHARK

Source	Destination	Protocol	Info
192.168.65.227	192.168.65.218	RTSP	DESCRIBE rtsp://192.168.65.218/audio/prue
192.168.65.218	192.168.65.227	RTSP/SDP	Reply: RTSP/1.0 200 OK, with session desc
192.168.65.227	192.168.65.218	RTSP	SETUP rtsp://192.168.65.218/audio/pruebar
192.168.65.218	192.168.65.227	RTSP	Reply: RTSP/1.0 200 OK
192.168.65.227	192.168.65.218	RTSP	PLAY rtsp://192.168.65.218/audio/pruebame
192.168.65.218	192.168.65.227	RTSP	Reply: RTSP/1.0 200 OK
192.168.65.227	192.168.65.218	RTSP	TEARDOWN rtsp://192.168.65.218/audio/prue
192.168.65.218	192.168.65.227	RTSP	Reply: RTSP/1.0 200 OK

Autores del Proyecto

4.1.2. Implementación de RTP

El protocolo RTP es el encargado de transportar la información generada por el servidor DSS sobre el protocolo de transporte UDP, y para su implementación se debe tener en cuenta la estructura del paquete RTP ilustrada en la Figura 6. La sesión RTP se inicia en el momento en el que el servidor comienza a emitir mensajes RTP. Al recibir el datagrama del servidor DSS, se le extrae la cabecera RTP del paquete y se deja la carga útil que será el contenido a reproducir por la aplicación de Streaming de audio, esta carga útil viene con una estructura específica del protocolo RTP que se puede observar en la Figura 19. El paso a seguir es reestructurarla para su reproducción, basándose en la estructura del códec correspondiente ver sección 4.4.3.

¹² Página Oficial. Disponible en: <http://www.wireshark.org/>

Figura 19. Carga Útil Paquete RTP

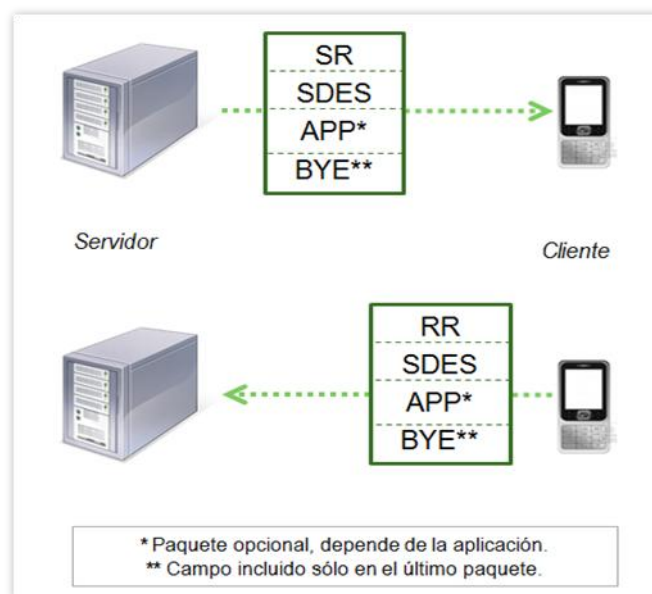


Fuente: RFC 3640: RTP Payload Format for Transport of MPEG-4 Elementary Streams. [22]

4.2. IMPLEMENTACIÓN DE RTCP

La implementación del protocolo RTCP se realizó en varios procesos. En primera instancia y luego de investigar acerca de su funcionamiento se realizó una revisión sobre la lectura de los paquetes RTCP enviados desde el servidor a la aplicación. A continuación, se procedió a armar la estructura de los paquetes de respuesta del cliente al servidor y finalmente se estableció el tiempo para el envío de los paquetes, según la sugerencia de la RFC 3550 [10] y algunas publicaciones. Los paquetes RTCP son enviados desde el servidor hacia el cliente en paquetes compuestos como se muestra en la Figura 20.

Figura 20. Envío de paquetes RTCP



Autores del Proyecto

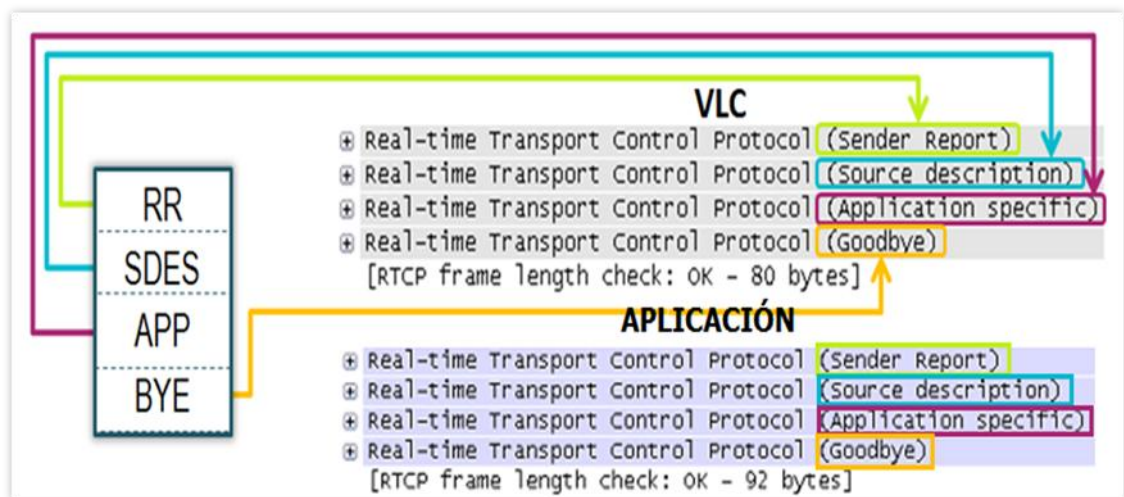
4.2.1. Lectura de Paquetes RTCP Recibidos

Para lograr una interpretación correcta de los campos de cada paquete RTCP recibido, se realizó un análisis que se apoyó en el valor de los campos de cada paquete compuesto RTCP que mostraba la herramienta WIRESHARK tanto cuando la reproducción se hacía con VLC como cuando se realizaba con la aplicación móvil.

4.2.1.1. Paquete Compuesto RTCP Recibido

La Figura 21 refleja que un paquete compuesto generado dentro de una transmisión con la aplicación móvil no contiene el paquete APP, debido a que este paquete es opcional y para el desarrollo de esta aplicación no es necesario, pero si cuenta con los demás tipos de paquetes RR, SDES y BYE. El paquete BYE sólo es enviado por el servidor cuando él presente alguna razón para terminar la transmisión.

Figura 21. Paquete Compuesto RTCP



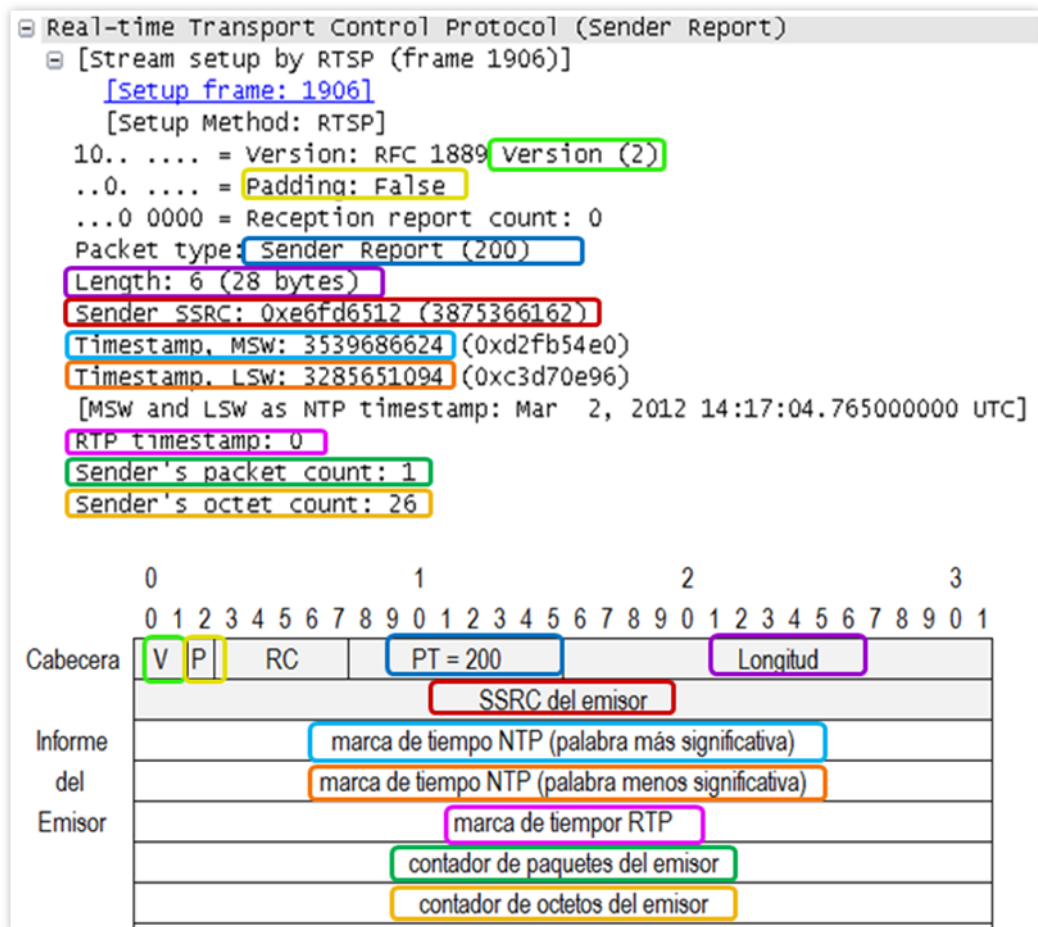
Autores del Proyecto

Por otra parte, en la anterior imagen se logró observar que el paquete compuesto enviado desde el servidor (fuente) hacia el reproductor (receptor) contenía varios tipos de paquetes RTCP, los cuales se detallan a continuación.

4.2.1.2. Paquete SR Recibido

El primero es el paquete *Sender Report (SR)* es mostrado en la Figura 22, allí se puede observar que los campos de la estructura del paquete SR son plenamente identificados por la aplicación y sus valores corresponden a los mostrados por WIRESHARK.

Figura 22. Campos del Paquete SR

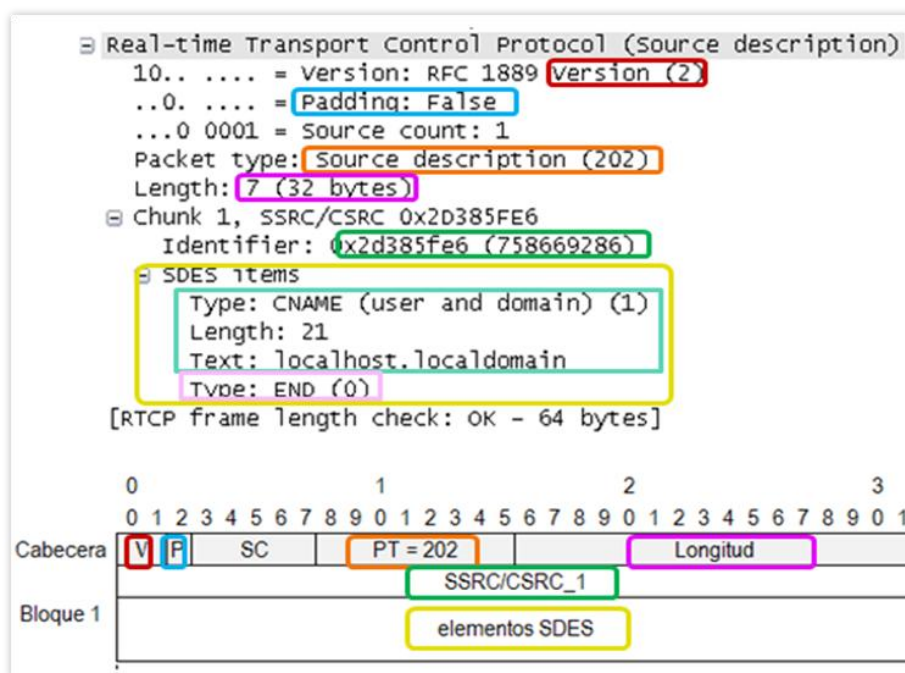


Autores del Proyecto

4.2.1.3. Paquete SDES Recibido

El siguiente paquete es el denominado *Source Description (SDES)*, la Figura 23 muestra los campos que se obtienen del paquete.

Figura 23. Campos del Paquete SDES

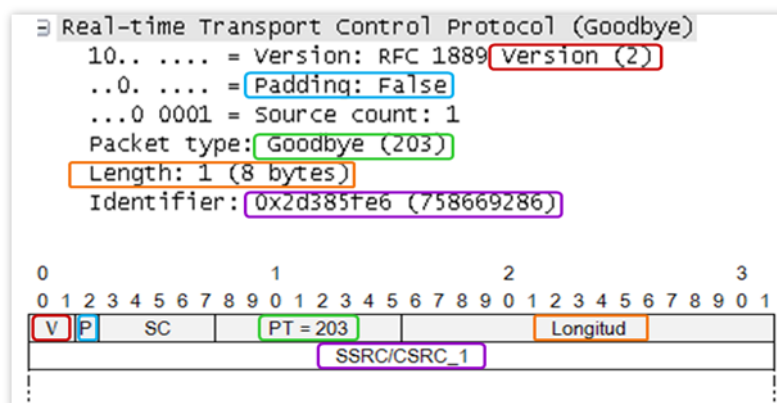


Autores del Proyecto

4.2.1.4. Paquete BYE Recibido

Para finalizar se muestra en la Figura 24 el paquete Goodbye recibido por la aplicación, para esta investigación este paquete es enviado sin ningún campo adicional ya que la aplicación no lo requiere, es suficiente con identificar el paquete para determinar que la transmisión es detenida y/o terminada.

Figura 24. Campos del Paquete BYE



Autores del Proyecto

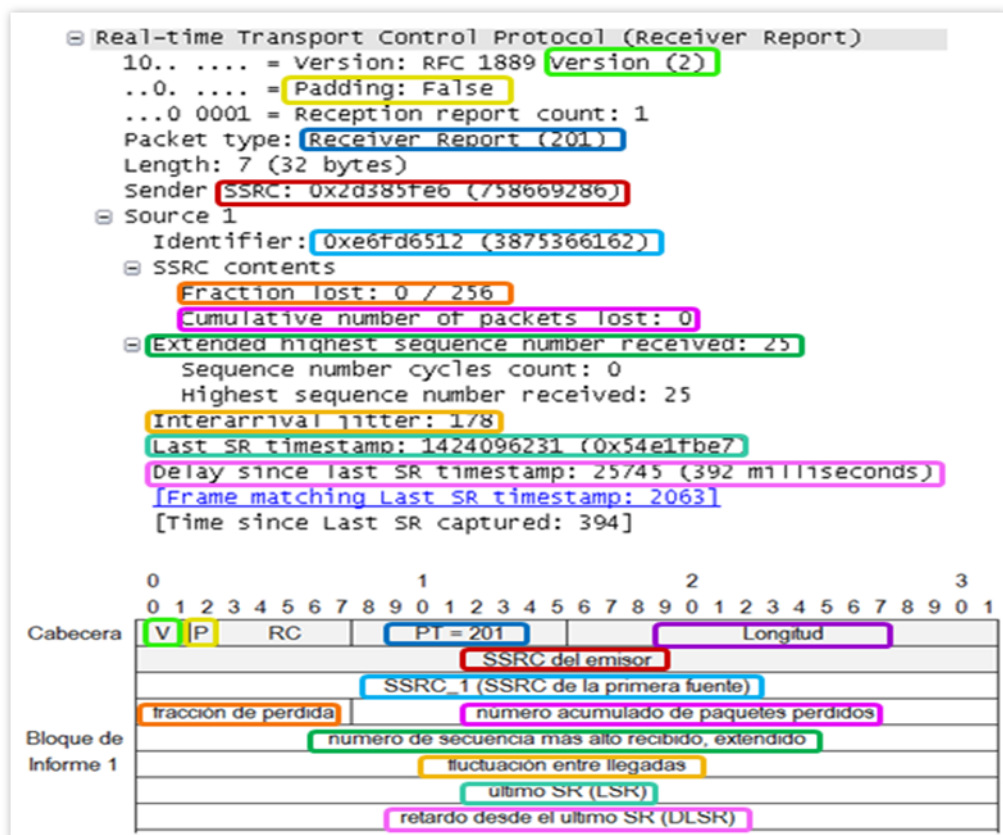
4.2.2. Armado y Envío de Paquetes RTCP

Al momento de armar el *paquete compuesto RTCP* a enviar es importante resaltar el cálculo de varios campos de algunos de los paquetes y el formato de un paquete compuesto (ver Capítulo 2.3.4.). A continuación se describen los tipos de paquetes RTCP enviados.

4.2.2.1. Paquete RR Enviado

Con el fin de verificar que el análisis a los campos del paquete RR es el adecuado y que éste sea enviado correctamente, se realizó la comparación de los datos calculados más adelante (ver Figura 25) los cuales fueron extraídos de WIRESHARK cuando se realizaba una transmisión de Streaming con ayuda del reproductor VLC.

Figura 25. Campos del Paquete RR



Autores del Proyecto

A continuación se detalla la forma de calcular algunos de los campos más relevantes de los paquetes RR [23].

a) Fracción de Pérdida: Fracción de paquetes perdidos (ver Ecuación 1) [14].

Esta variable sirve para tener un control de los paquetes RTP perdidos desde el último informe emitido.

Ecuación 1. Fracción de Pérdida

$$F = \frac{P}{E}$$

Donde:

F = Fracción de pérdida

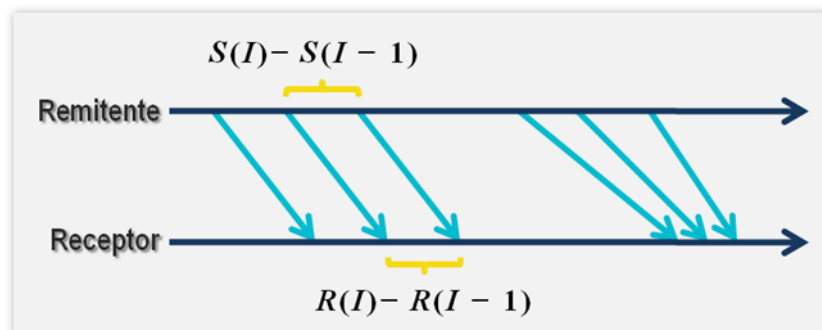
P = No. de paquetes perdidos

E = No. de paquetes esperados

b) Fluctuación (*jitter*): Este campo informa de la variación del retardo de transmisión de los paquetes RTP.

En la Figura 26 se puede observar las diferencias de tiempo entre las salidas y llegadas de los paquetes RTP, que se calculan mediante la Ecuación 2.

Figura 26. Representación de la Diferencia Entre Llegadas



Fuente: *The Multimedia Control Protocol RTCP* [23]

Ecuación 2. Diferencia Salidas y Llegadas

$$D(I) = [R(I) - R(I - 1)] - [S(I) - S(I - 1)]$$

Donde:

$D(I)$ = Diferencia entre el tiempo entre llegadas en el receptor y el espacio entre los *paquetes de datos RTP* adyacentes que salen de la fuente (remitente).

$R(I)$ = Tiempo de llegada al receptor.

$S(I)$ = Tiempo de salida del emisor.

El valor de $D(I)$ es utilizado en la Ecuación 3 para calcular el valor de la fluctuación.

Ecuación 3. Cálculo del Jitter

$$J(I) = J(I - 1) + \frac{D(I) - J(I - 1)}{16}$$

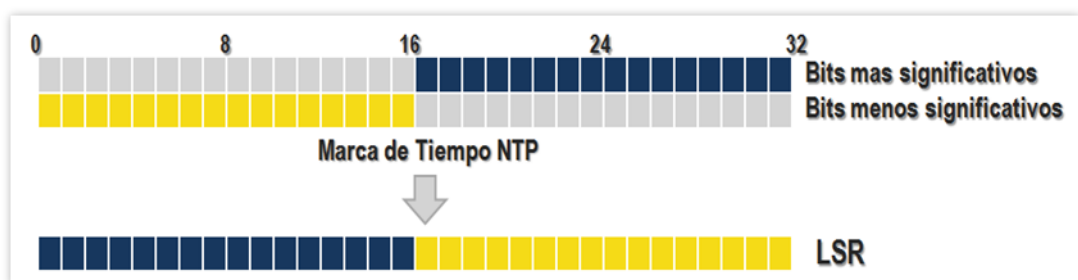
Donde:

$J(I)$ = Jitter o fluctuación media entre llegadas hasta la recepción del paquete I de datos RTP.

A falta de fluctuación los espacios serán iguales y $D(I)$ tendrá un valor de cero.

c) **LSR (Último SR)**: El cálculo de este campo es mostrado en la Figura 27.

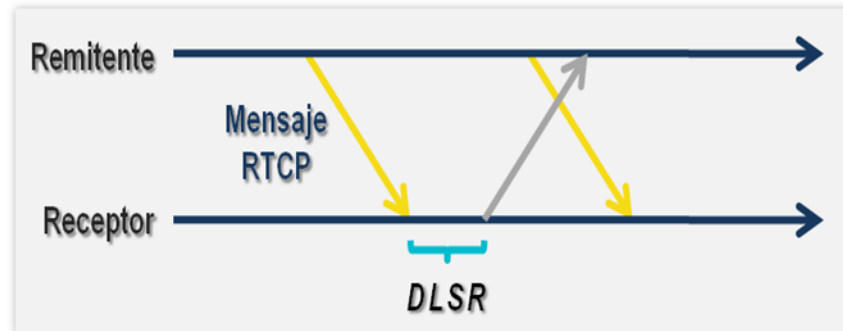
Figura 27. Representación Gráfica del Cálculo de LSR



Autores del Proyecto

- d) **DLSR (Retardo desde el último SR):** La descripción de este campo se representa en la Figura 28, en donde el DLSR es el tiempo entre la llegada de un SR al cliente (receptor) y la salida del siguiente RR.

Figura 28. Representación Gráfica del DLSR



Autores del Proyecto

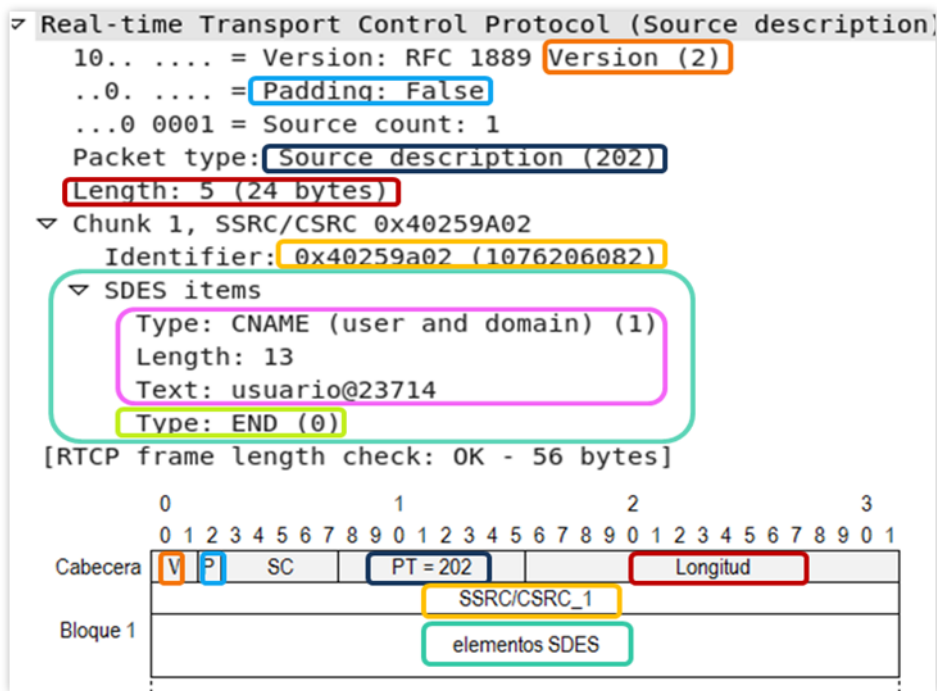
4.2.2.2. Paquete SDES Enviado

El paquete SDES que se envía a la fuente tiene como elementos el CNAME (nombre canónico) y al elemento END que la da fin a lista de elementos.

- a) **CNAME:** Éste el único campo obligatorio del paquete SDES y es utilizado para mantener una identificación del participante o cliente de la sesión en caso tal de que haya problemas en la conexión. Para este caso se establece la estructura *usuario@numeroAleatorio*.

Seguidamente el cálculo de los campos del paquete SDES es comparado con los campos de la transmisión usando VLC que se muestra en la Figura 29.

Figura 29. Campos del Paquete SDES

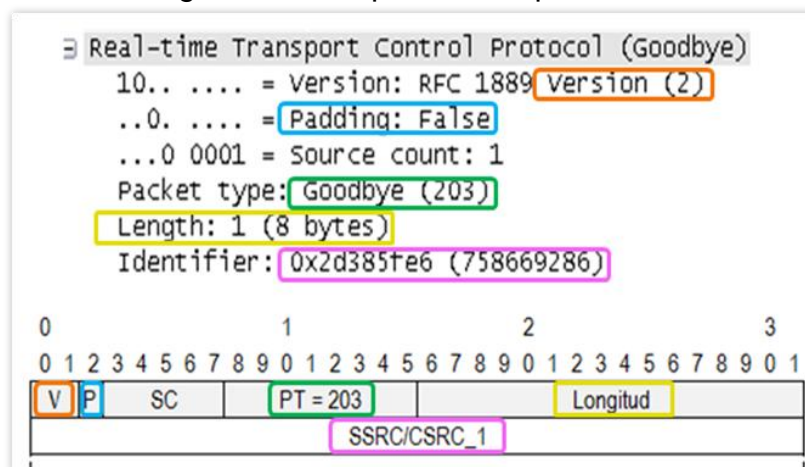


Autores del Proyecto

4.2.2.3. Paquete BYE Enviado

El paquete BYE es enviado para informar que se le da fin a la sesión, por lo tanto se envía únicamente en el último paquete compuesto RTCP. Este paquete también es comparado con el arrojado en la reproducción con VLC (ver Figura 30).

Figura 30. Campos del Paquete BYE



Autores del Proyecto

4.2.3. Tiempos de Envío de los Paquetes RTCP Compuestos

Según la RFC 3550 [10] el valor mínimo para el envío de los paquetes RTCP se fija en 5 segundos, y que para el caso de la secciones Unicast este valor se usa para los participantes que no son emisores activos. Teniendo en cuenta que la aplicación desarrollada se encuentra sobre una distribución bajo demanda con esquema Unicast este es el tiempo utilizado para el intervalo entre envío de paquetes RTCP.

Por otra parte, las condiciones de los tiempos de envíos en sesiones Multicast varían drásticamente, estos ya no son constantes si no que dependen de algoritmo para calcularlos, este algoritmo lo que buscan es calcular el intervalo para enviar los RTCP ya sea cuando envía la aplicación (ver Ecuación 4) o cuando lo hace el servidor (ver Ecuación 5).

Ecuación 4. Tiempo para Envío de RR

$$PR_{rr} = \frac{0.75 \times BW}{PS_{rr} \times nr}$$

Donde:

BW = Ancho de Banda

PS_{rr} = Tamaño medio del paquete RR

nr = Número de participantes o número de receptores

Ecuación 5. Tiempo para Envío de SR

$$PR_{sr} = \frac{0.25 \times BW}{PS_{sr} \times ns}$$

Donde:

BW = Ancho de Banda

PS_{sr} = Tamaño medio del paquete SR

nr = Número de emisores o número de fuentes

4.3. SELECCIÓN DE SERVIDOR DE STREAMING

Dentro de la investigación se encontraron varios servidores de Streaming disponibles en el mercado y se escogieron dos como los más adecuados para el desarrollo del proyecto. Sus características más importantes se detallan en la Tabla 18.

Tabla 18. Servidores de Streaming

Variable	Servidor	
	Darwin Streaming Server (DSS) [24]	Helix Mobile Server [25]
Soporte de Protocolos de Streaming	RTSP RTP RTCP	RTSP RTP RTCP
Interfaz Web	SI. Administrador web que usa generalmente el puerto 1220.	SI. Es una interfaz administrador donde los puertos usados generalmente son 20214/16739
Licencia	Apple PublicSourceLicense	RealNetworksPublicSourceLicense
Multiplataforma	SI	SI
Soporte de Códec AMR-WB	Soportado. Hinting requerido.	Soportado. Hinting no es requerido (.3gp, .mp4, .3g2, .mov)
Conexiones	Se aceptan hasta 1000 usuarios (valor por defecto).	Depende de la licencia, la licencia gratuita acepta hasta 10 streams.
Tipo o Esquema de Distribución	Unicast Multicast	Unicast Multicast

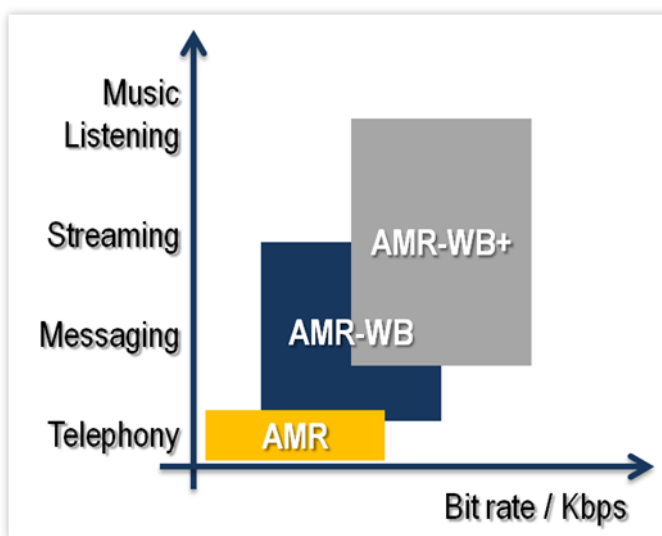
Autores del Proyecto

Para el desarrollo del proyecto se seleccionó el servidor *Darwin Streaming Server*, en base a la investigación realizada en el proyecto anterior, llevado a cabo en la Universidad Industrial de Santander.

4.4. CÓDECS

Los códecs de audio de la familia AMR inicialmente fueron diseñados para compresión de voz en llamadas telefónicas, en la medida en la que aparecían nuevas versiones y tecnologías fue ampliándose su rango de uso, en la Figura 31 se muestran las aplicaciones a las cuales son dirigidas los códecs en comparación de las tasas de bits que estos requieren.

Figura 31. Aplicaciones y Tasas de bits de Códecs



Fuente: *Mobile Audio Formats*. [26]

A partir de la gráfica anterior y de investigaciones realizadas por la 3GPP [27] [28] [29] [30] se puede afirmar que el códec AMR-WB+ es el más apropiado para la transmisión mediante Streaming, debido a su calidad de audio a baja tasas de bits como se puede ver en 2.6.2, lo que garantiza una velocidad de transmisión adecuada manteniendo la calidad del audio. Además se encontró en las diferentes paginas oficiales, que la mayoría de los dispositivos móviles celulares de marcas disponibles en Colombia (Motolora, Nokia, Sony Ericsson, Samsung) no incluyen en sus características el soporte para el códec AMR-WB+ (ver Anexo B), es por esto que el códec AMR-WB+ no pudo ser usado como códec base de esta investigación.

En consecuencia y teniendo en cuenta que el códec AMR-WB tiene ventajas frente al códec AMR en cuanto a la calidad de audio a bajas tasas de bits (ver sección 2.6), lo que implica que tenga un rango más amplio de uso como se puede ver en la Figura 31. También se encontró que el códec AMR-WB es soportado en la mayoría de las marcas de celulares disponibles en Colombia. Por esto el códec AMR-WB fue el seleccionado para usarse dentro del desarrollo de la aplicación.

4.4.1. Codificación de Archivo de Audio AMR-WB

Para el proceso de Codificación inicialmente se seleccionaron archivos de audio con formato Windows Media Audio (WMA), seguidamente se les realizó el proceso de codificación al formato AMR-WB con la herramienta *Total Video Converter*¹³, teniendo en cuenta las siguientes variables:

Tasa de bits: 23.85 kbps

Frecuencia de muestreo: 16 KHz

Formato contenedor: *.mov.

Los valores tomados para la tasa de bits y la frecuencia de muestreo fueron pensados con el fin de garantizar la mayor calidad de audio que se puede obtener con el códec AMR-WB como se muestra en la Tabla 13 y la Tabla 14. En cuanto al formato contenedor se escogió el *.mov, ya que es uno de los pocos formatos que el servidor DSS soporta.

Por otra parte, para que los archivos de audio puedan ser interpretados por los servidores de Streaming generalmente se requiere que estos pasen por un proceso llamado *hint*, el cual permite que los archivos de audio contengan información de cómo realizar el Streaming por parte del servidor [3].

¹³ No es marca registrada

El Hint de los archivos se realizó mediante la herramienta MPEG4IP (ver Figura 32), la cual requería que los archivos se encontraran en formato contenedor MOV.

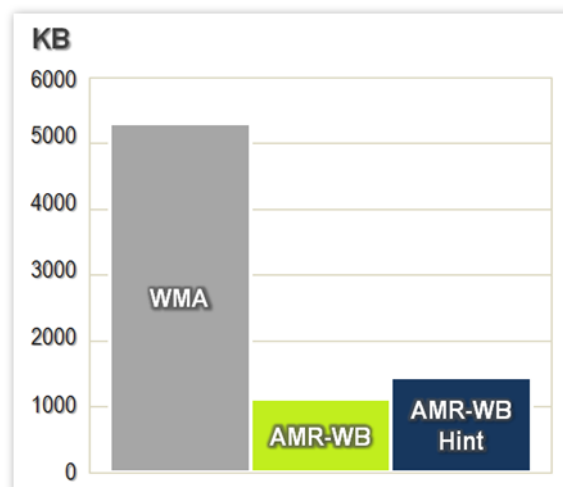
Figura 32. Proceso Hint

```
D:\>cd MPEG4IP
D:\MPEG4IP>mp4creator.exe -hint=1 Audio.mov
ReadAtom: invalid atom size, extends outside parent atom - skipping to end of "
awb" "" 1045446 vs 1045142
MP4ERROR: MP4File::FindIntegerProperty: no such property - moov.trak[0].mdia.mi
f.stbl.stsd.*.esds.decConfigDescr.objectTypeId
MP4ERROR: MP4File::FindIntegerProperty: no such property - moov.trak[1].mdia.mi
f.stbl.stsd.*.esds.decConfigDescr.objectTypeId
D:\MPEG4IP>mp4info.exe Audio.mov
mp4info.exe version 1.5.0.1
Audio.mov:
Track  Type      Info
1      audio    AMR-WB, 342.500 secs, 24 kbps, 16000 Hz
2      hint     Payload AMR-WB for track 1
Metadata 1001: mp4creator 1.5.0.1
D:\MPEG4IP>
```

Autores del Proyecto

Durante el proceso que tienen los archivos de audio, su tamaño en KBytes también cambia, la Figura 33 muestra un ejemplo del tamaño de los archivos en cada momento.

Figura 33. Tamaños códec



Autores del Proyecto

4.4.2. Estructura de un Paquete RTP

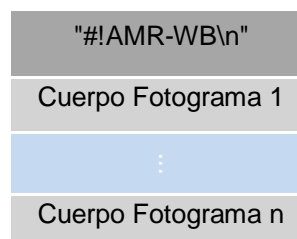
Para obtener los datos correspondientes al audio incluido en el paquete RTP hay que tener en cuenta su estructura ver Figura 19, la cual está compuesta por los siguientes elementos:

- a) **Una cabecera RTP:** Proporciona información necesaria para su uso en dentro de una sesión RTP.
- b) **Sesión de cabeceras de la Unidad de Acceso (AU):** Contiene las cabeceras de cada AU.
- c) **Sesión auxiliar:** Brinda información opcional.
- d) **Sesión de datos de la AU:** Lleva los datos del audio que se están transmitiendo.

4.4.3. Estructura de un Archivo AMR-WB

Debido a que los objetos Player no permiten la reproducción de un flujo de bytes a medida que este se va llenando, si no por el contrario sólo reproducen flujos de bytes completos, se requiere la construcción de pequeños archivos AMR-WB de cada paquete RTP. Para lograr esto se debe tener en cuenta la estructura correspondiente a los archivos AMR-WB la cual se muestra en laFigura 34.

Figura 34. Formato de Archivo AMR-WB



Fuente: *RFC 3267: 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.* [31]

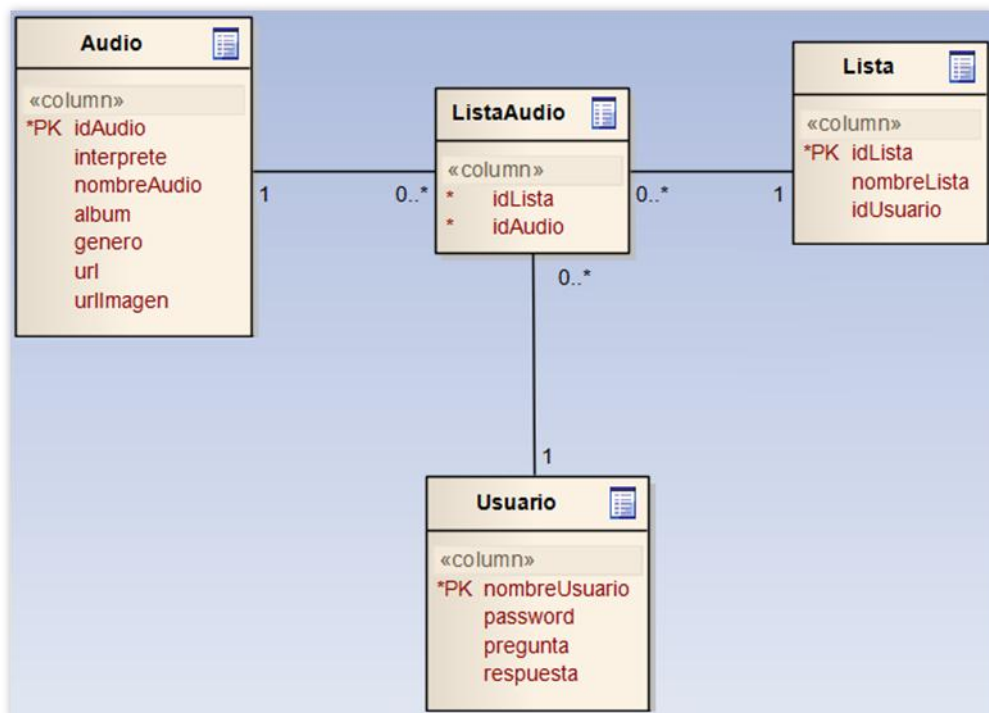
4.5. MÓDULOS DE LA INTERFAZ GRÁFICA DE LA APLICACIÓN MÓVIL

Para la interfaz gráfica del reproductor de audio y de toda la aplicación móvil se utilizó la Librería LWUIT¹⁴, con el fin de proporcionar una interfaz más agradable al usuario y que permitiera mayor usabilidad y portabilidad de la aplicación.

4.5.1. Diseño UML de la Base de Datos de la Aplicación

Para el diseño se establecieron 4 tablas: Lista, Audio, ListaAudio y Usuario. El diagrama UML de la Base de Datos se muestra en la Figura 35.

Figura 35. Diagrama UML de Base de Datos de la Aplicación



Autores del Proyecto

A continuación se detallan los módulos que hacen parte de la aplicación móvil:

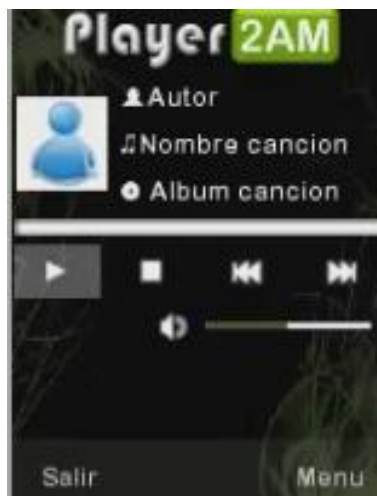
¹⁴ permite controlar las propiedades de los componentes gráficos que se muestran en la aplicación móvil.

4.5.2. Interfaz Gráfica del Reproductor de Audio

La interfaz gráfica del reproductor está compuesta por los siguientes campos (ver Figura 36):

- *Imagen e información del audio:* En esta sección se muestra una imagen referente al audio en reproducción además de otra información pertinente al audio, como intérprete, nombre del audio y nombre del álbum al cual pertenece.
- *Línea de tiempo de reproducción:* Esta línea permite determinar el progreso de la reproducción del archivo de audio.
- *Botones del reproductor:* En este campo se muestran los botones principales para el control de la reproducción de los archivos de audio (play, stop, pause, next, back).
- *Control de volumen:* Este campo cuenta con una línea que permite ver el estado del volumen y un botón que permite silenciar el audio en cualquier momento.
- *Información de la reproducción:* Acá se muestra el tiempo transcurrido de la reproducción respecto al tiempo total del audio y la posición que ocupa el audio dentro de una lista.

Figura 36. Reproductor de Audio Sobre Emulador.



Autores del Proyecto

4.5.3. Buscador

4.5.3.1. Interfaz Móvil del Buscador

El módulo de *Buscar* (ver Figura 37) funciona de la siguiente manera:

- Posee una caja de texto para ingresar lo que se desea buscar, bien sea un artista, un nombre de un audio, ambos campos o parte de ellos.
- Luego de digitar las palabras a buscar y presionando el comando “*Buscar*” se listan en pantalla los resultados.

Figura 37. Pantalla de Interfaz del Buscador.



Autores del Proyecto

4.5.3.2. Algoritmo de Búsqueda

El algoritmo utilizado para llevar a cabo la búsqueda funciona de la siguiente manera (ver Anexo C):

- a) La aplicación móvil toma lo digitado en la caja de texto y lo envía al *servlet* como una cadena de caracteres.
- b) El *servlet* separa la búsqueda por palabras y las almacena en un vector llamado “*vectorBusqueda*”, cuyos elementos son cada una de las palabras ingresadas en el texto de búsqueda.

- c) En el *servlet* se crea una matriz llamada “*matrizAudio*” en donde los registros de la tabla “Audio” (ver Tabla 19) son ingresados separando las palabras por cada columna de la matriz. Un ejemplo de la *matrizAudio* se muestra en la Tabla 20.

Tabla 19. Tabla “Audio” de la BD

idAudio	interprete	nombreAudio	album	genero
10	Ricardo arjona	mujeres	mujeres	romántica
11	Fanny lu	Celos	regreso	pop
12	Vicente fernandez	Celos	el rey	ranchera

Autores del Proyecto

Tabla 20. Matriz creada a partir de la tabla “Audio”

10	ricardo	arjona	mujeres	mujeres	romántica	
11	fanny	lu	celos	regreso	pop	
12	vicnete	fernandez	celos	el	rey	ranchera

Autores del Proyecto

- d) Teniendo el *vectorBusqueda* y la *matrizAudio* creados se comparan uno a uno los campos y se establece un resultado a partir del número de coincidencias en cada registro. Finalmente, las coincidencias se ordenan de mayor a menor como se muestra en la Tabla 21.

Tabla 21. Resultado interno de una búsqueda

Mostrar en pantalla	No. coincidencias
Celos, Fanny Lu	3
Celos, Vicente Fernandez	1

Autores del Proyecto

4.5.4. Usuarios

El módulo de *Usuarios* se creó para darle un agregado visual e interactivo a la aplicación, al ser un usuario registrado se tienen ciertas ventajas frente a los usuarios no registrados.

4.5.4.1. Formularios de la Interfaz Móvil de Usuarios

Figura 38. Formularios del Módulo de Usuarios.



Autores del Proyecto

a) Inicio de sesión: Este formulario es el formulario principal de la GUI y permite que un usuario registrado ingrese a su cuenta. Además tiene botones de acceso a otros dos formularios: Registrarse y Recordar Password (ver Figura 38a).

- b) Registrarse:** Permite que un usuario nuevo haga su respectivo registro controlando que el nombre de usuario no se haya ingresado anteriormente (ver Figura 38b).

- c) Recordar Password:** Este formulario le permite a los usuarios registrados recordar su contraseña dando respuesta a su pregunta de seguridad ingresada al momento de registrarse (ver Figura 38c).

- d) Modificar Password:** A este formulario sólo se puede acceder ya habiendo ingresado a la cuenta y sirve para modificar la password que le permite ingresar a la sesión (ver Figura 38d).

4.5.4.2. Servlet de Usuarios

Es el *servlet* que hace la conexión a la base de datos de la aplicación y que a su vez permite consultar la tabla de usuarios, modificar y/o agregar cualquier registro, este *servlet* se aloja en la carpeta *Tomcat* en el servidor Web, para que esté listo a cualquier solicitud por parte de la aplicación móvil.

5. RESULTADOS Y PRUEBAS

Todo proceso de investigación requiere demostrar sus resultados y esto lo hace a través de la realización de pruebas que permitan determinar lo ocurrido en los procesos del proyecto. En este caso se realizaron pruebas unitarias del código, pruebas de la aplicación en emuladores y dispositivos móviles reales, pruebas del funcionamiento del protocolo RTCP, pruebas del rendimiento de la red y uso de la memoria RAM, como se describen a continuación.

5.1. PRUEBAS UNITARIAS

De acuerdo con la metodología seleccionada, se realizaron las *Pruebas Unitarias* a algunos métodos desarrollados durante la elaboración del proyecto con el propósito de descubrir posibles errores que pudieran producirse en el código.

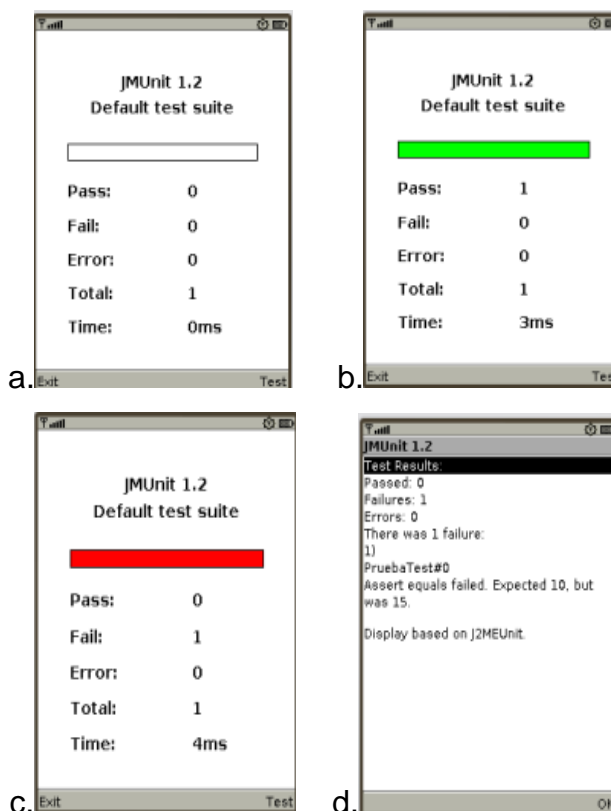
Para la realización de las pruebas unitarias se utilizó la librería JMunit¹ basada en JUnit, está facilita el proceso de elaboración de cada una de las pruebas y sus principales características se listan a continuación [32]:

- Funciona tanto en el emulador de SUN como en dispositivos reales.
- Son pequeñas, se ejecutan incluso en dispositivos antiguos.
- Tiene una amplia colección de métodos *Assert*, que permiten el control de fallos.
- Las clases TestCases y TestSuites son soportadas.

En la Figura 39 se puede apreciar lo que ocurre al realizar una prueba unitaria con JMunit:

¹ Página de Descarga. [online] Disponible en: <http://sourceforge.net/projects/jmunit/>

Figura 39. Pantallas JUnit



- a) Esta pantalla muestra los campos a evaluar *antes de realizar la prueba*.
- b) La barra de progreso se muestra de color *verde* cuando el resultado de la prueba es exitoso.
- c) La barra de progreso es de color *rojo* si se dieron fallos en el código.
- d) Informe que genera la prueba al encontrar *fallos* en el código.

Autores del Proyecto

En el Anexo E se muestran los métodos de cada una de las clases de la aplicación puestos a prueba con JUnit en el desarrollo del proyecto.

5.2. PRUEBAS DE LA APLICACIÓN EN EMULADORES Y DISPOSITIVOS REALES

La aplicación móvil fue probada y usada en varios emuladores de diferentes marcas y en algunos dispositivos celulares, esto con el fin de mostrar el funcionamiento y el soporte de la aplicación. Estas pruebas evalúan la conexión al servidor Web, la conexión al servidor Streaming y la reproducción

de audio en los dispositivos y/o emuladores dependiendo del códec de audio usado: AMR o AMR-WB.

La Tabla 22 muestra los aspectos evaluados tanto para los emuladores como para los dispositivos reales, por lo general la conexión al servidor Web o al servidor de Streaming era exitosa, los inconvenientes radicaban en la reproducción debido a que en el caso de los emuladores, Netbeans depende del reproductor QuickTime para la utilización de MMAPI y este reproductor no soporta el códec AMR-WB por lo tanto no se lleva a cabo la reproducción.

Para el caso de los dispositivos reales probados en el desarrollo de esta prueba se tuvo diferentes inconvenientes que se describen a continuación:

- Nokia X3-02: No logra interpretar la estructura codificada del archivo de audio AMR-WB, por lo tanto no se tiene una reproducción exitosa.
- Samsung Galaxy Mini s5570: lleva con éxito la reproducción pero en un tiempo muy corto debido a que este dispositivo no soporta Java, por lo tanto se requiere la instalación de un emulador, el cual posee algunas restricciones.
- BlackBerry 9300: No soporta conexión con Sockets, es por esto que no se logra tener una conexión exitosa con el servidor Streaming.

Tabla 22. Prueba de Emuladores y Dispositivos Físicos

Emulador	Conexión a servidor Web	Conexión a servidor Streaming	Reproducción	Códec	MIME
Java™ Platform Micro Edition SDK 3.0	Exitosa	Exitosa	Exitosa	AMR	audio/amr
			Fallida	AMR-WB	-
Samsung SDK 1.2.1	Exitosa	Exitosa	Exitosa	AMR	audio/amr
			Fallida	AMR-WB	-
MOTODEV SDK for	Exitosa	Exitosa	Fallida	AMR	audio/x-amr

Java™ ME 3.x			Fallida	AMR-WB	audio/amr-wb
Nokia SDK 1.0	Fallida	Exitosa	Fallida	AMR	audio/amr
			Fallida	AMR-WB	audio/amr-wb
Dispositivo Real	Conexión a servidor Web	Conexión a servidor Streaming	Reproducción	Códec	MIME
Nokia X3-02	Exitosa	Exitosa	Fallida	AMR	audio/amr
			Fallida	AMR-WB	audio/amr-wb
Samsung Galaxy Mini s5570	Exitosa	Exitosa	Fallida	AMR	audio/amr
			Exitosa	AMR-WB	audio/amr-wb
BlackBerry 9300	Exitosa	Fallida	-	AMR	audio/amr

Autores del Proyecto

5.3. PRUEBA DE RTCP

Las siguientes pruebas fueron realizadas para verificar que la implementación del protocolo RTCP llevada a cabo en este proyecto se realizó de forma correcta, para esto se tuvo en cuenta a VLC como herramienta guía, ya que este reproductor cuenta con la implementación de este protocolo en su funcionamiento. En las pruebas se tuvo en cuenta las características del protocolo RTCP, como el control de pérdida y fluctuación de los paquetes RTP, además del manejo de envío de los mensajes RTCP con el fin de no congestionar la red.

Por otra parte se tuvo en cuenta sólo las siguientes funciones de RTCP, esto debido a que la implementación realizada fue dirigida a distribuciones Unicast y no a Multicast, en donde las otras funciones no mencionadas son de interés:

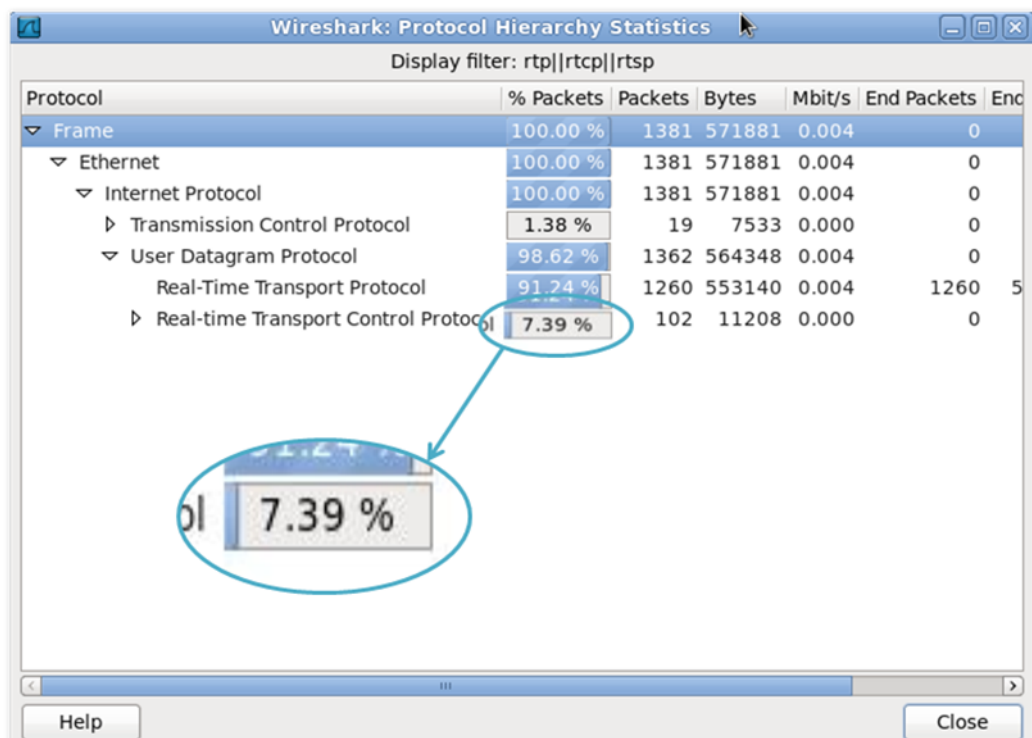
- Proveer realimentación sobre la calidad de la distribución de los datos como por ejemplo la pérdida de paquetes y el exceso de fluctuación.

- Controlar la tasa de envío de los paquetes RTCP con el fin de no congestionar la red.

5.3.1. Ancho de Banda Usado por RTCP

Según la RFC 3550 [10] la fracción del ancho de banda utilizado por RTCP debe ser alrededor de 5%, debido a esto se realizó una prueba con la cual se pudiera verificar el consumo de los paquetes del protocolo RTCP implementado. La Figura 40 ilustra el porcentaje del ancho de banda consumido en la transmisión de un archivo de audio transmitido desde el DSS a la aplicación funcionando desde el emulador Java™ Plataforma Micro Edition SDK 3.0. El consumo utilizado por RTCP es 7.39%, es decir, que el consumo del protocolo RTCP implementado en este proyecto es muy similar a la recomendaciones descritas en la RFC.

Figura 40. Ancho de Banda de la Sesión



Autores del Proyecto

Se llevó a cabo la reproducción de un mismo archivo de audio 20 veces, para cada reproducción se tomó el porcentaje de ancho de banda ocupado por RTCP, de estos datos se pudo obtener los valores estadísticos mostrados en la Tabla 23.

Tabla 23. Valores estadísticos del porcentaje de ancho de banda ocupado por RTCP

Variable Estadística	Valor Calculado
Mediana	6.75%
Media Aritmética o Promedio	6.47%
Valor Máximo	7.24%
Valor Mínimo	4.48%
Desviación Estándar	0.008066061

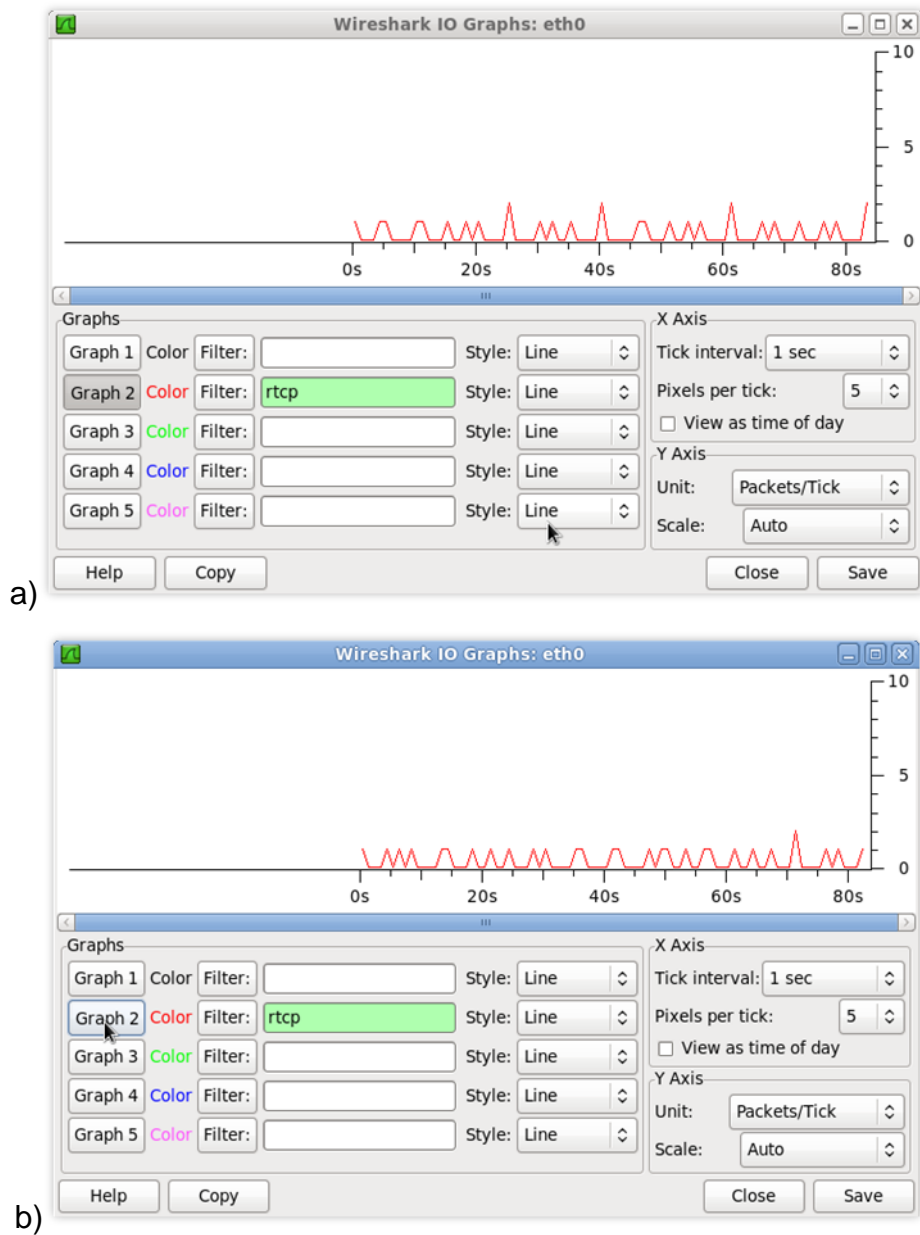
Autores del Proyecto

En la tabla anterior se puede ver claramente que la Mediana de los datos es 6,75%, además se observa que el valor máximo para el porcentaje ocupado por RTCP es 7,24%, mientras que el valor mínimo que ocupa es 4,48%. Es decir que los datos giran a valores muy cercanos del 5% que es el valor requerido.

5.3.2. Comportamiento de RTCP

Esta prueba es una comparación entre la herramienta VLC y la aplicación desarrollada en este proyecto, con el fin de mostrar el patrón de envío y recepción de los paquetes RTCP en un intervalo de tiempo, para esto se tomaron las gráficas en la Figura 41.

Figura 41. Gráfica de Paquetes RTCP



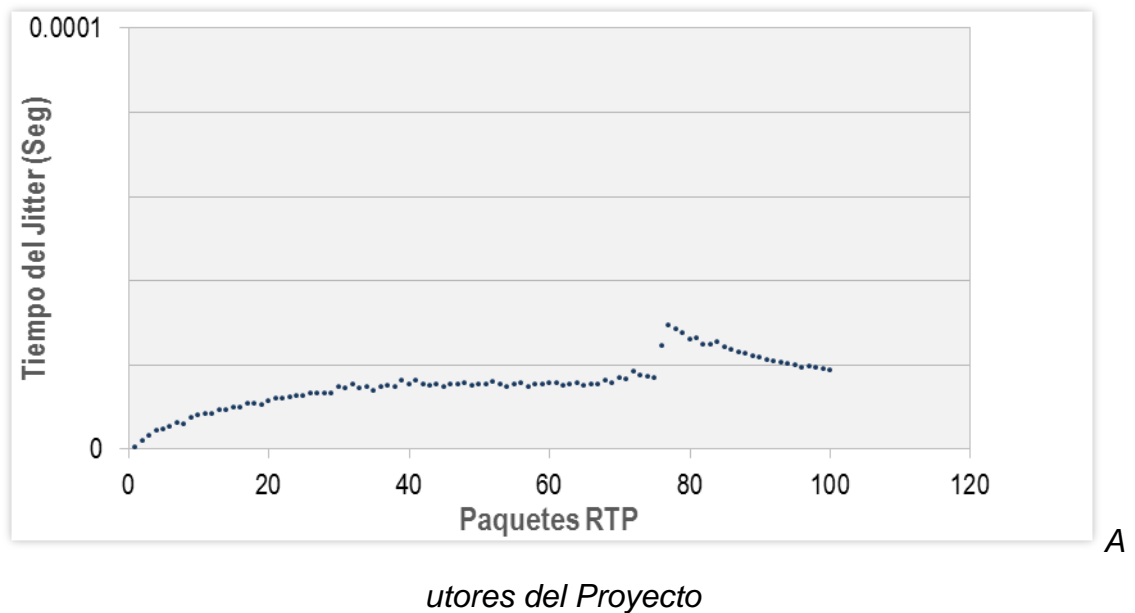
Autores del Proyecto

Para esta prueba se tomó como referencia el comportamiento de envío de los paquetes RTCP teniendo como objetivo verificar si el tiempo entre envíos de los paquetes RTCP de la aplicación desarrollada (ver Figura 41b) en este proyecto tiene un comportamiento similar al de los paquetes transmitidos usando VLC (ver Figura 41a). Al apreciar las dos figuras podemos concluir que el resultado de esta prueba fue exitoso.

5.3.3. Comportamiento de la Fluctuación

Teniendo en cuenta que la variable *Jitter* de la transmisión de los paquetes RTP es uno de los campos importantes en un paquete RR, debido a que representa la variabilidad del tiempo de ejecución (adelantos o retrasos en la llegada de los paquetes de datos), realizar esta prueba es importante para verificar la congestión de la red, es decir a menor valor en el jitter, es menor la congestión en la red. La Figura 42 muestra el comportamiento del *Jitter* en la reproducción de un archivo de audio con la aplicación móvil desarrollada. En esta figura el valor tomado por el *Jitter* varía entre 0 y 0.0001 segundos, es decir, toma valores muy pequeños lo cual implica que la red no está saturada y los paquetes RTP tendrán un tiempo de recepción aproximadamente constante.

Figura 42. Fluctuación de los paquetes RTP



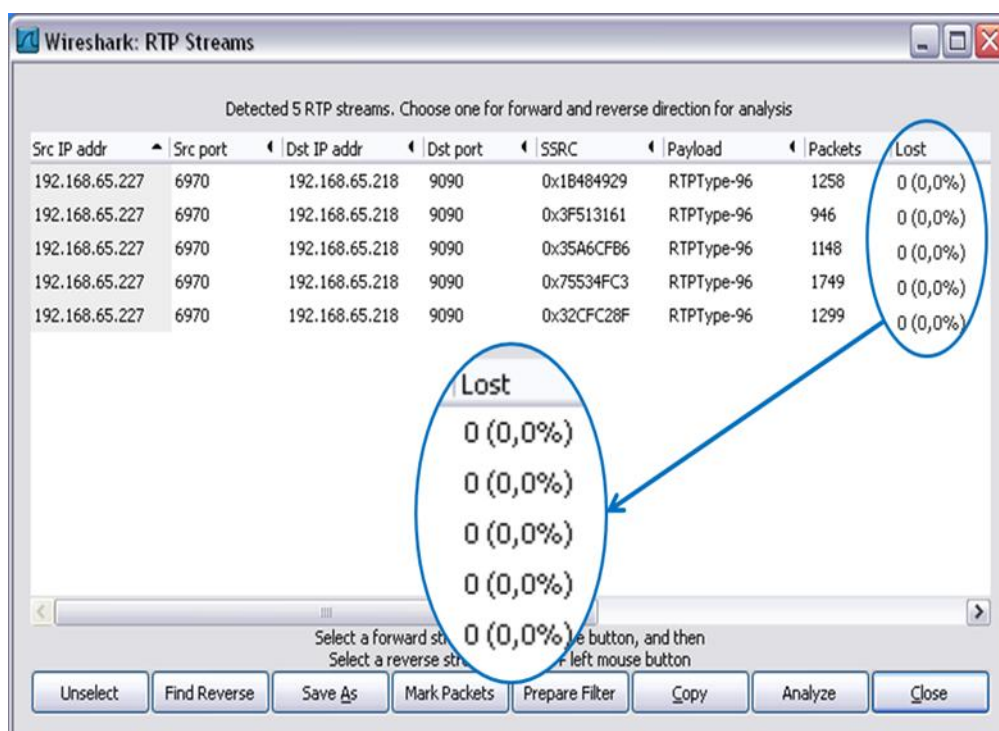
5.3.4. Pérdida de Paquetes

Con la ayuda del protocolo RTCP se puede llevar un control de los paquetes perdidos de una sesión RTP. En esta prueba se busca evidenciar este control a través del registro de los paquetes perdidos en uno de los campos de paquete

RTCP. En la Figura 43 se puede observar que la pérdida de paquetes durante la transmisión de diferentes archivos de audio a la aplicación equivale a cero en todos los casos, lo que significa que los paquetes RTP enviados por el servidor son recibidos en su totalidad en la aplicación. La variable de paquetes perdidos hace parte de los campos del paquete RR, en donde se obtiene tanto la fracción de paquetes perdidos desde el último SR, como el total de paquetes perdidos en toda la sesión.

Las condiciones de cada una de las transmisiones son dadas en una distribución Unicast y en condiciones ideales, es decir, con poca congestión de paquetes en la red. Esta prueba se realizó en dichas condiciones debido a que el DSS en su versión gratuita sólo permite un máximo de 10 conexiones simultáneas y al trabajar en una red LAN sin acceso a internet creada por los autores del proyecto no se pudo saturar la red con paquetes ajenos a los de los protocolos de Streaming (RTP, RTSP y RTCP).

Figura 43. Pérdida de Paquetes



The screenshot shows the 'Wireshark: RTP Streams' window. At the top, it says 'Detected 5 RTP streams. Choose one for forward and reverse direction for analysis'. Below this is a table with columns: Src IP addr, Src port, Dst IP addr, Dst port, SSRC, Payload, Packets, and Lost. The table contains five rows of data, all showing 0 (0,0%) in the 'Lost' column. A blue circle highlights the 'Lost' column header and the values in the first two rows. Another blue circle highlights the 'Lost' column values in the last two rows. A blue arrow points from the first circle to the second. At the bottom of the window, there are buttons: Unselect, Find Reverse, Save As, Mark Packets, Prepare Filter, Copy, Analyze, and Close. Below the buttons, there is a small text box with instructions: 'Select a forward stream and click the right mouse button, and then select a reverse stream and click the left mouse button'.

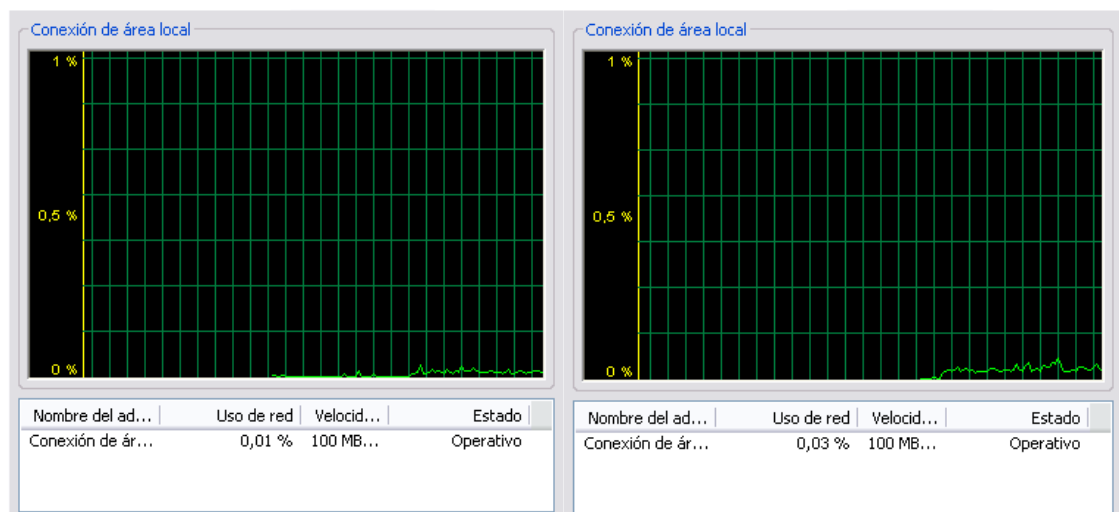
Src IP addr	Src port	Dst IP addr	Dst port	SSRC	Payload	Packets	Lost
192.168.65.227	6970	192.168.65.218	9090	0x1B484929	RTPTYPE-96	1258	0 (0,0%)
192.168.65.227	6970	192.168.65.218	9090	0x3F513161	RTPTYPE-96	946	0 (0,0%)
192.168.65.227	6970	192.168.65.218	9090	0x35A6CFB6	RTPTYPE-96	1148	0 (0,0%)
192.168.65.227	6970	192.168.65.218	9090	0x75534FC3	RTPTYPE-96	1749	0 (0,0%)
192.168.65.227	6970	192.168.65.218	9090	0x32CFC28F	RTPTYPE-96	1299	0 (0,0%)

Autores del Proyecto

5.4. RENDIMIENTO DE LA RED EN LA TRANSMISIÓN DE CÓDECS

El desarrollo de esta prueba permitió evaluar el rendimiento en la red al transmitir archivos de audio codificados en AMR y AMR-WB. En la Figura 44 se puede evidenciar un mayor consumo de red por parte del formato AMR-WB frente al formato AMR. Lo que quiere decir que al transferir un archivo de audio con formato AMR-WB se requiere un mayor consumo de la red.

Figura 44. Rendimiento de Red



a) Transmisión usando el códec AMR

b) Transmisión usando el códec AMR-WB

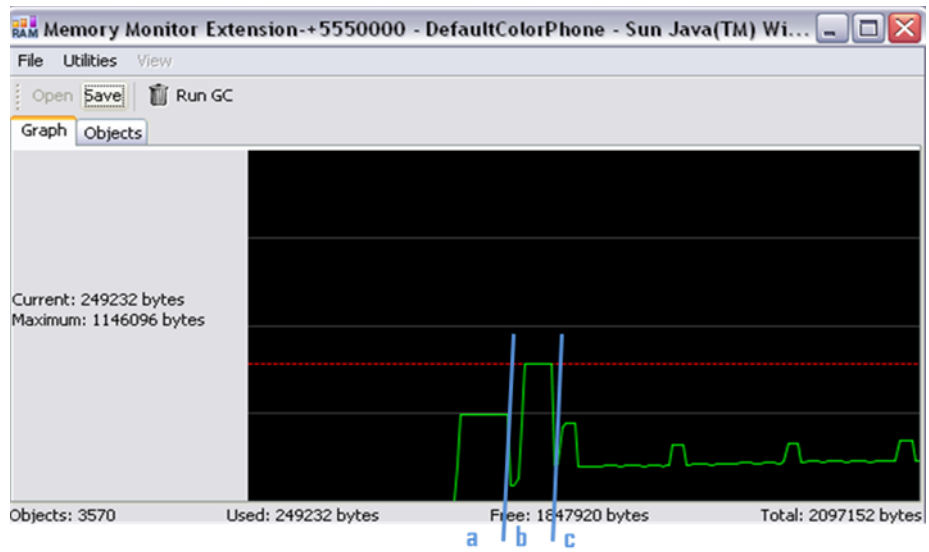
Autores del Proyecto

5.5. CONSUMO DE RAM DE LA APLICACIÓN

Esta prueba se llevó a cabo con el fin de determinar el consumo de RAM que tiene la aplicación desarrollada. La prueba se realizó con el uso de la herramienta Wireless Toolkit 2.5.2., la cual brinda la opción de monitoreo de RAM.

La siguiente figura muestra el momento del consumo de RAM al iniciar el emulador (a), luego el consumo de RAM al abrir a la aplicación (b) y finalmente el proceso de reproducción (c).

Figura 45. Consumo de RAM



Autores del Proyecto

6. CONCLUSIONES

Con base en la implementación realizada en la investigación anterior con el proyecto “*Desarrollo de un sistema Streaming de audio orientado a dispositivos móviles*” en el que se trabajaron los protocolos propios de Streaming RTP y RTSP y el uso que se le dio dentro de la creación de la aplicación de esa investigación, podemos concluir que la utilización de estos protocolos proporciona el soporte básico al uso de la técnica Streaming, donde el protocolo RTSP se encarga de proporcionar control de la reproducción y el protocolo RTP permite el transporte de los datos.

El protocolo de control RTCP fue diseñado para brindar *Calidad del Servicio* (QoS) y control a los paquetes dentro de una sesión RTP. En esta investigación se implementó RTCP para dispositivos móviles celulares sobre una distribución Unicast, con la cual se pudo determinar que el protocolo RTCP no proporciona un control significativo en una sesión RTP con distribución Unicast, debido a que los parámetros y servicios de control fueron diseñados especialmente para distribuciones Multicast. Sin embargo, la implementación del protocolo permitió conocer la especificación y opciones de control del protocolo que se podrían tener en cuenta para futuros proyectos que hicieran uso de transmisiones Multicast utilizando dispositivos móviles celulares.

Dentro de la investigación sobre los códecs pertenecientes a la familia AMR se encontró que el más adecuado para el uso de la técnica Streaming es el códec AMR-WB+. Sin embargo, este códec no es soportado por los dispositivos móviles celulares consultados en la literatura ni por los emuladores de marcas reconocidas como Nokia, Motorola, entre otros. Por esta razón, se eligió al códec AMR-WB como el códec de audio a utilizar dentro de este proyecto, teniendo en cuenta que brinda una mejor calidad con respecto al códec AMR y es adecuado para el servicio de Streaming como se puede ver en la Figura 31.

En cuanto a la interfaz móvil se lograron algunas funciones adicionales con respecto a la aplicación anterior. Estas funciones incluyen el manejo de usuarios para permitirles guardar información acerca de listas de audio seleccionadas, con el fin de usarlas en posteriores reproducciones. Así mismo la aplicación ahora cuenta con control del volumen del audio.

En esta implementación la aplicación software fue limitada a ser utilizada en celulares que posean conexión WI-FI. Esta restricción se debió a que el DSS se encontraba en una red LAN privada. Sin embargo, esta restricción se colocó debido a inconvenientes ajenos a los autores del proyecto que estaban relacionados con la adquisición de un servidor con IP pública que permitiera la distribución de los archivos de audio en Internet. En el caso de contar con el apoyo de un servidor con IP pública, la aplicación se podría probar en diversas marcas celulares de baja gama por medio de redes de Telefonía Celular.

Por último, la metodología XP utilizada en el desarrollo del proyecto fue de gran ayuda a la hora de realizar la implementación de los requisitos. La metodología escogida apoyó la programación por pares lo que permitió realizar la codificación de la aplicación con éxito, descubriendo y evitando errores a tiempo y aumentando la calidad del código. Además el manejo de iteraciones permitió una fácil adaptación a los cambios que fueron surgiendo durante el proceso de desarrollo.

7. RECOMENDACIONES

Teniendo en cuenta que el desarrollo en aplicaciones móviles sigue en ascenso y que un sistema software siempre podrá ser mejorado, se sugieren las siguientes recomendaciones para ser tenidas en cuenta en posteriores investigaciones:

- Teniendo en cuenta que la aplicación fue desarrollada para una distribución Unicast, se propone que el desarrollo en próximas versiones se extienda a distribuciones Multicast con el fin de darle uso completo al protocolo RTCP.
- Dentro de la investigación se observó que la cabecera de un paquete RTP cuenta con información necesaria para la realización de sus funciones como lo son la fragmentación, sincronización intramedia, identificación de tipo de carga, identificación de trama e identificación de fuente, estas funciones son necesarias si se quiere emigrar a distribuciones Multicast y por esta razón se propone tener en cuenta la implementación de la cabecera del protocolo RTP en futuros desarrollos.
- El proyecto desarrollado se propone como una base para la continuación de otros proyectos de pregrado y maestría en la Universidad. En pregrado, podría realizarse un estudio detallado de diferentes códecs de audio incluyendo la familia aacPlus y un modelo de distribución multicast. Además, la aplicación de este proyecto en una institución como el Instituto de Lenguas de la UIS, donde los archivos de audio suministrados por los cursos fueran accesibles a los estudiantes a través de sus dispositivos móviles, sería de gran interés para la comunidad. A nivel de postgrado, un análisis de algoritmos de transmisión sobre un modelo multicast, aprovechando la arquitectura y aplicación propuesta en este proyecto, sería una alternativa para continuar con el trabajo realizado.

BIBLIOGRAFÍA

- [1] PRIETO SERPA Felipe Andrés and RODRIGUEZ RODRIGUEZ Luisa Fernanda, Streaming de Audio a través de Dispositivos Móviles Utilizando J2ME, Bogotá: Universidad el Bosque, 2007.
- [2] QUINTERO ORTIZ Juan Pablo and CASTRO SERNA Cristian Andrey, Evaluación de Servidores de Streaming de Video Orientado a Dispositivos Móviles, Medellín: Universidad de Antioquia, 2006.
- [3] JAIMES CONTRERAS Luz Carolina and PLATA JAIMES Jessika Liliana, Desarrollo de un Sistema Streaming de Audio Orientado a Dispositivos Móviles, Bucaramanga: Universidad Industrial de Santander, 2010.
- [4] GOYAL Vikram, "Experiments in Streaming Content in Java ME," 22 Agosto 2006. [Online]. Available: <http://today.java.net/pub/a/today/2006/08/22/experiments-in-streaming-java-me.html?force=542#making-sense-of-rtsp-rtp-and-rtcp>. [Accessed 2 Diciembre 2011].
- [5] GARCÍA ANTÚNEZ Luis Fernando, RTPLIB: Una Librería en Java de los Protocolos RTP y RTCP, Madrid (España): Universidad Rey Juan Carlos, 2009.
- [6] RODRÍGUEZ ROLDÁN Daniel, Stack RTP para Dispositivos Móviles Symbian, Barcelona (España): Universidad Politécnica de Cataluña, 2006.
- [7] GORASSON Marcus and SONESSON Christian, Analysis and Evaluation of Audio Codecs for Mobile Devices, Suecia: Department of Computer Science of Lund Institute of Technology, 2006.
- [8] RFC 2326, "Real Time Streaming Protocol (RTSP)," Abril 1998. [Online]. Available: <http://www.ietf.org/rfc/rfc2326.txt>. [Accessed 12 Marzo 2011].
- [9] SANTOS Jose, "Mobile Streaming - Qualidade de Experiência," 18 Marzo 2010. [Online]. Available: http://www.websantos.net/tese/doku.php?id=estado_da_arte:protocolos_de_streaming:rtsp_rtp. [Accessed 27 Marzo 2011].

- [10] RFC 3550, "RTP: A Transport Protocol for Real-Time Applications," Julio 2003. [Online]. Available: <http://www.ietf.org/rfc/rfc3550.txt>. [Accessed 4 Febrero 2011].
- [11] STALLINGS William, Redes e Internet de Alta Velocidad. Rendimiento y Calidad del Servicio, Madrid (España): Pearson Education, 2004.
- [12] HUIDOBRO MOYA José Manuel and ROLDÁN MARTÍNEZ David, Integración de Voz y Datos, Madrid (España): McGraw Hill, 2003.
- [13] BARCELÓ José, Protocolos y Aplicaciones Internet, Barcelona: UOC, 2008.
- [14] AGBINYA Jhonson I, IP Communications and Services for NGN, Auerbach Publications, 2009.
- [15] KAARANEN Heikki, Redes UMTS: Arquitectura, Movilidad y Servicio, Ciudad de México (México): RA-MA, 2006.
- [16] HUI Suk Yu and YEUNG Kai Hau, "Callenges in the Migration to 4G Mobile Systems," *IEEE*, vol. 41, 2003.
- [17] ITU, "World Radiocommunication Seminar Highlight Future Communication Technologie," Diciembre 2010. [Online]. Available: http://www.itu.int/net/pressoffice/press_releases/2010/48.aspx. [Accessed 3 Junio 2011].
- [18] AUTTI Henri and BISTROM Johnny, Mobile Audio – from MP3 to AAC and Further, HUT, Telecommunications Software and Multimedia Laboratory, 2004.
- [19] FROUFE QUINTAS Agustin and JORGE CARDENAS Patricia, Java 2 Micro Edition: Manual de Usuario y Tutorial, Alfaomega, RA-MA, 2004.
- [20] MAHMOUND Qusay, "The J2ME Mobile Media API," Junio 2003. [Online]. Available: <http://developers.sun.com/mobility/midp/articles/mmapioverview/index.htm> I. [Accessed 5 Junio 2012].
- [21] UNIVERSIDAD CARLOS III, "Multimedia en MIDP 2.0," [Online]. Available:

- http://www.it.uc3m.es/celeste/docencia/j2me/tutoriales/midp2_0/PracticaMedia/. [Accessed 23 Septiembre 2011].
- [22] RFC 3640, "RTP Payload Format for Transport of MPEG-4 Elementary Streams," Noviembre 2003. [Online]. Available: <http://tools.ietf.org/html/rfc3640>. [Accessed 15 Febrero 2011].
- [23] JEFFAY Kevin, "The Multimedia Control Protocol RTCP," 5 Octubre 1999. [Online]. Available: <http://www.cs.odu.edu/~cs778/jeffay/Lecture6.pdf>. [Accessed 15 Agosto 2011].
- [24] Apple Computer Inc., "Administrator's Guide: Darwin Streaming Server," 20 Febrero 2011. [Online]. Available: <http://manuals.info.apple.com/en/quicktimestreamingsrvradminguide.pdf>. [Accessed 24 Octubre 2011].
- [25] RealNetworks Inc., "Helix Mobile Server Administration Guide," 28 Marzo 2008. [Online]. Available: http://docs.real.com/docs/server12/mobile/HelixMobileServerAdmin_v12.pdf. [Accessed 29 Septiembre 2011].
- [26] SEPPANEN Jarno, "Mobile Audio Formats," 26 Octubre 2004. [Online]. Available: <https://helix-server.helixcommunity.org/>. [Accessed 10 Enero 2012].
- [27] M. Jari, B. Bruno, B. Stefan, O. Pasi, S. Redwan and TALEB Anisse, "AMR-WB+: A New Audio Coding Standard for 3RD Generation Mobile Audio Services," *IEEE*, pp. 1109-1112, 2005.
- [28] VARGA Imre, "On New Audio Codec Specifications," *IEEE*, pp. 128-131, 2007.
- [29] 3GPP TS 26.234, "Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs," 30 Septiembre 2011. [Online]. Available: <http://www.3gpp.org/ftp/Specs/html-info/26234.htm>. [Accessed 5 Noviembre 2011].
- [30] 3GPP TR 26.936, "Performance characterization of 3GPP audio codecs," Marzo 2011. [Online]. Available: <http://www.3gpp.org/ftp/Specs/html-info/26936.htm>. [Accessed 5 Noviembre 2011].

- [31] RFC 3267, "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," Junio 2002. [Online]. Available: <http://www.ietf.org/rfc/rfc3267.txt>. [Accessed 15 Enero 2012].
- [32] "JMUnit," [Online]. Available: <http://jmunite.sourceforge.net/>. [Accessed 25 Marzo 2011].

ANEXOS

Anexo A. Caso de Uso de la Aplicación

Nombre	Buscar Audio	
Descripción	Permite al usuario realizar una búsqueda de un archivo de audio	
Actores	Usuario, Usuario Registrado	
Precondición	—	
Postcondición	—	
Eventos	Actor	Sistema
	1. El usuario digita respecto del audio que desea encontrar (titulo, autor, genero...).	
	2. Presiona el comando buscar.	
		3. Envía al servidor Web el contenido digitado por el usuario.
		4. En el servidor se ejecuta un algoritmo de búsqueda, que envía una respuesta a la aplicación con los registros que coinciden con la búsqueda.
		5. Se muestra en pantalla la respuesta del servidor.

Nombre	Reproducir Audio	
Descripción	Reproduce el audio almacenado en la lista en curso	
Actores	Usuario, Usuario Registrado	
Precondición	Necesita que haya una lista en curso para poder reproducir audio de esa lista	
Postcondición	—	
Eventos	Actor	Sistema
	1. El usuario selecciona el audio que desea escuchar y presiona el botón Play.	
		2. El cliente envía una petición RTSP al servidor de Streaming con la URL correspondiente al archivo de audio solicitado.
		3. El servidor Streaming envía una respuesta RTSP al cliente y emite los paquetes RTP
		4. El reproductor recibe los paquetes RTP a medida de que los reproduce.

Nombre	Registrar Usuario	
Descripción	Permite a un Usuario registrarse para posteriores ingresos	
Actores	Usuario	
Precondición	—	
Postcondición	Luego del registro el Usuario pasa a ser un Usuario Registrado y por tanto puede iniciar sesión	
Eventos	Actor	Sistema
	1. El usuario ingresa los datos correspondiente al registro y presiona el comando Registrar.	2. Recoge los datos registrados y los envía al servidor Web para que los almacene en la base de datos.

Nombre	Recordar Password	
Descripción	Permite al Usuario Registrado recordar la password mediante la respuesta a la pregunta secreta	
Actores	Usuario Registrado	
Precondición	Ser un Usuario Registrado	
Postcondición	—	
Eventos	Actor	Sistema
	1. Ingresa el nombre de usuario y presiona el comando Pregunta.	2. Envía al servidor Web el nombre de usuario y obtiene la pregunta correspondiente al Usuario Registrado.
		3. El servidor Web envía la pregunta para que se muestre en pantalla, la respuesta para verificarla posteriormente y el password correspondiente.
	4. El Usuario Registrado digita la respuesta y presiona el comando Verificar.	5. El sistema compara la respuesta, si es exitosa muestra en pantalla el password y si no muestra en pantalla que un mensaje de error.

Nombre	Iniciar Sesión	
Descripción	Permite al iniciar la sesión	
Actores	Usuario Registrado	
Precondición	Ser un Usuario Registrado	
Postcondición	Guardar listas de audio, modificar password.	
Eventos	Actor	Sistema
	1. Ingresa el nombre de usuario y la password y luego preciona el comando el botón Ingresar.	
		2. Envía los datos al servidor Web y los compara con los registros de la BD.
		3. Si los datos son correctos se abre la sesión, de lo contrario permite intentarlo nuevamente.

Nombre	Modificar Password	
Descripción	Permite modificar la password	
Actores	Usuario Registrado	
Precondición	Haber iniciado sesión	
Postcondición	—	
Eventos	Actor	Sistema
	1. Ingresa el password actual y el nuevo password y preciona el comando Modificar.	
		2. Envía los datos al servidor Web verificando la password actual y modifica el registro de la password en la BD.

Nombre	Guardar Listas	
Descripción	Permite guardar listas de audio.	
Actores	Usuarios Registrado	
Precondición	Iniciar Sesión	
Postcondición	—	
Eventos	Actor	Sistema
	1. Presiona el comando guardar e ingresa el nombre de la lista.	
		2. Toma los datos de la lista en curso enviándolos a el servidor Web para que los almacene en la BD asociándolo a el nombre de usuario.

Nombre	Acceder Ayuda	
Descripción	Da una sencilla explicación del funcionamiento de la aplicación móvil	
Actores	Usuario, Usuario Registrado	
Precondición	—	
Postcondición	—	
Eventos	Actor	Sistema
		1. El sistema muestra en pantalla un texto de ayuda.

Anexo B. Características de Referencias de Dispositivos Móviles Celulares

Marca	Referencia	Sistema Operativo	Soporte de JAVA	WIFI	Soporte de Códecs de Audio
ALCATEL ¹	OT-901A		SI	SI	MP3, AAC/AAC+,WMA
	OT-385A		SI	NO	MP3
BLACKBERRY ²	8520	BlackBerry OS	SI	SI	MP3, AMR-NB, AMR-WB, QCELP EVRC, AAC-LC, AAC+, eAAC+, WMA9
	9900	BlackBerry OS	SI	SI	MP3, AMR-NB, AAC-LC, AAC+, eAAC+, Flac, Vorbis .mp4, .m4a, .3gp, .m4v, .avi, .asf, .wma, .mp3, .flac, .ogg, .aac, .amr, .wav, .mid
	9300	BlackBerry OS	SI	SI	MP3, AMR-NB, AMR-WB, QCELP EVRC, AAC-LC, AAC+, eAAC+, WMA9
LG ³	T300 Cookie Music	Propietario	SI	NO	MP3, eAAC +, WAV
	LG T505 (Cookie WiFi)	Propietario	SI	SI	MP3/eAAC+/WAV
MOTOROLA ⁴	EX118 MOTOKEY XT	Android	SI	SI	H.264, AAC, H.263, MP3, MPEG-4, MIDI, AAC+
	PRO +	Android	NO	SI	H.264, AAC, H.263, MP3, MPEG-4, WAV, WMA, AMR WB, AMR NB, WMV v10, AAC+, WMV v9, MIDI
NOKIA ⁵	C3	SO Nokia	SI	SI	AAC, AMR-WB, Mobile XMF, MP3, MP4, WAV
	ASHA 300	S40 Asha	SI	NO	AAC, AAC+, AMR-NB, AMR-WB, HE AAC V2 MIDI poly 64, Mobile XMF MP3, MP4
	X3-02	SO Nokia (S40)	SI	SI	AAC, AAC+, AMR-WB, HE AAC V2, Mobile XMF, MP3, MP4, NRT, WAV, WMA
	E5	SO Symbian	SI	SI	AU, AAC, AAC+, AMR-WB, HE AAC V2, MP3 MP4, WAV, WMA
SAMSUNG ⁶	Samsung Star II	Android	SI	SI	MP3/WAV/WMA/AAC+

¹ Página Oficial de la Marca: <http://www.alcatelonetouch.com/co/>

² Página Oficial de la Marca: <http://co.blackberry.com/>

³ Página Oficial de la Marca: <http://www.lg.com/co/celulares/todos-los-telefonos/index.jsp>

⁴ Página Oficial de la Marca: <http://www.motorola.com/Consumers/MX-ES/Consumer-Products-and-Services/Mobile-Phones>

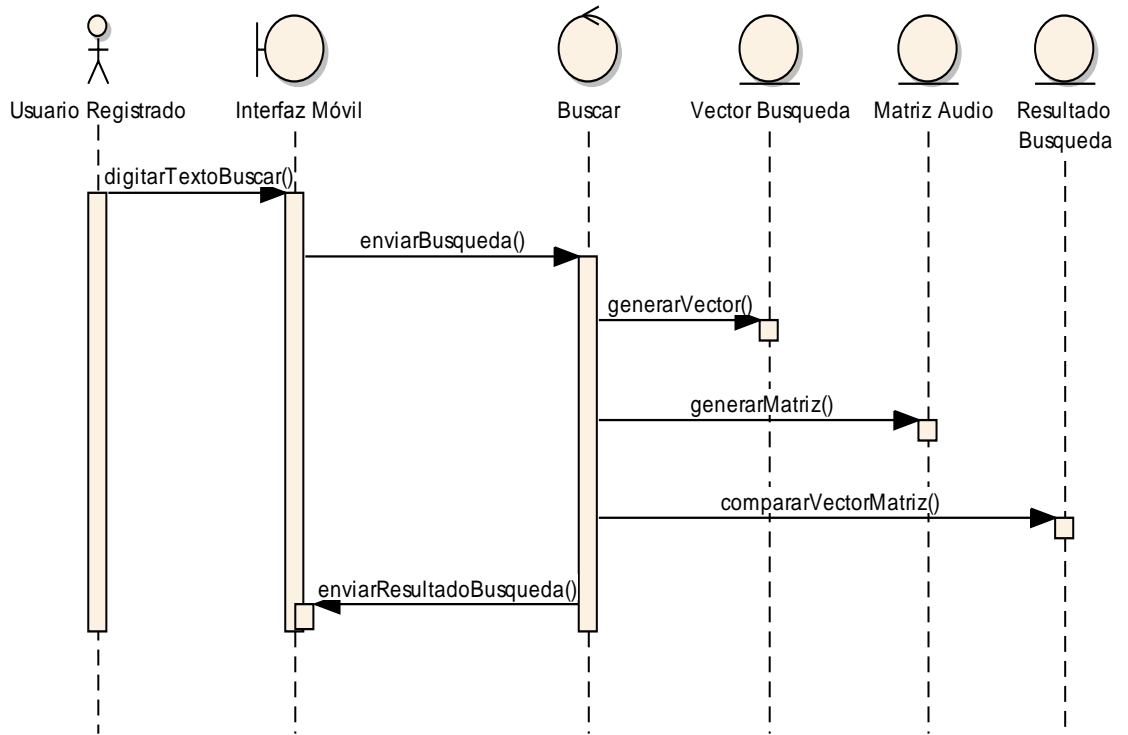
⁵ Página Oficial de la Marca: <http://www.nokia.com/co-es/>

	Samsung NARI	Android	SI	NO	MP3/AAC
	Samsung Galaxy Y	Android	NO	SI	MP3/WAV/eAAC+/OGG
SONY ERICSSON ⁷	Sony Ericsson Xperia mini pro	Android	NO	SI	MP3/WMA/WAV/eAAC+/AMR -WB
	Sony Ericsson TXT	Propietario	SI	SI	MP3,MIDI,AMR,WAV

⁶ Página Oficial de la Marca: <http://www.samsung.com/co/consumer/mobile-phones/mobile-phones>

⁷ Página Oficial de la Marca: <http://www.sonymobile.com/es/products/phones/>

Anexo C. Diagrama de Secuencia (Buscar)




Anexo D. UML Aplicación

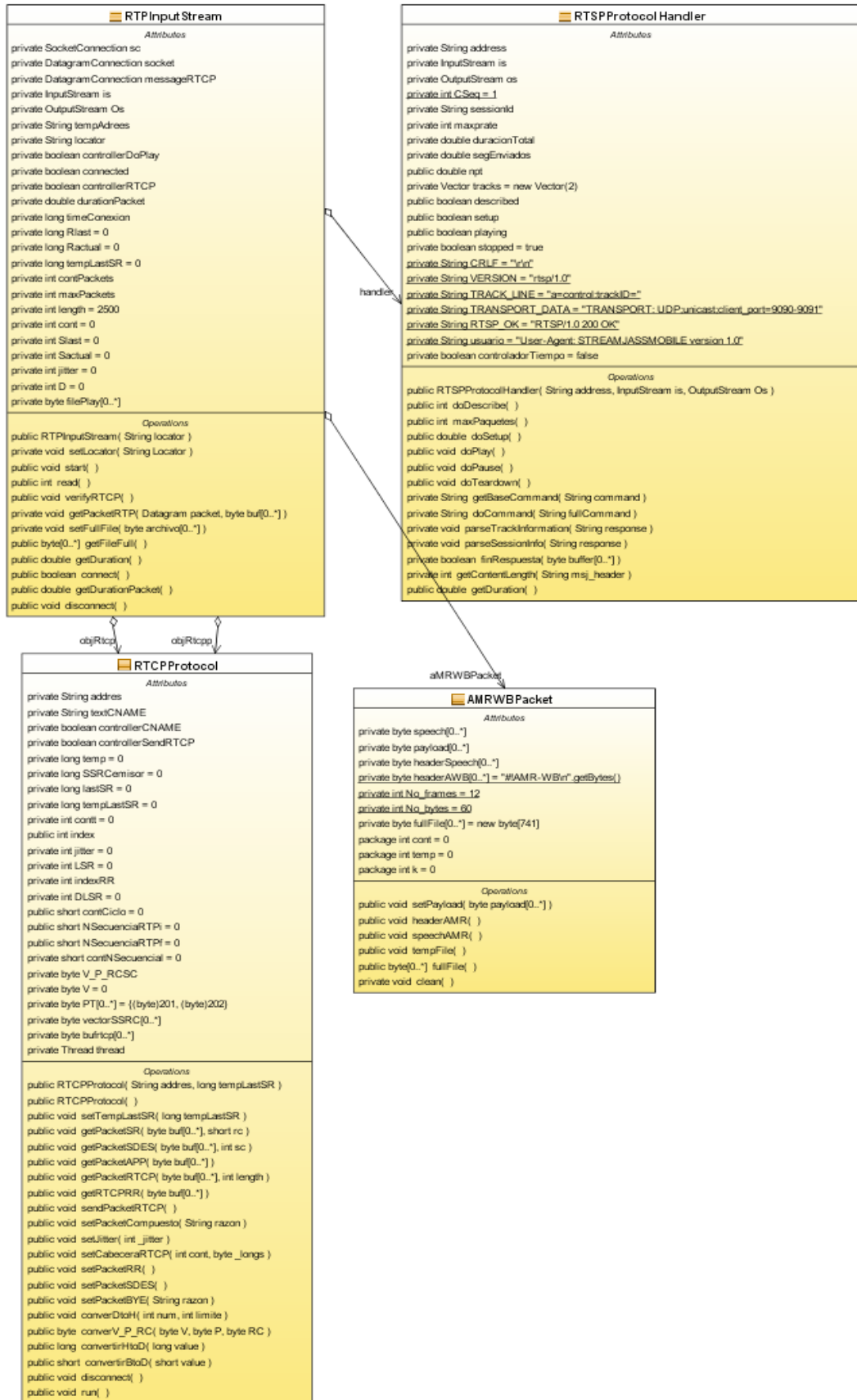
Paquete Listas



Paquete Buscador

 Search
<p style="text-align: center;"><i>Attributes</i></p> <pre>private TextField textSearch private List listSearch private List listCurso private Command search private Command back private String url private String temp private String Searching private String matrizInfoTemp[0..*,0..*]</pre>
<p style="text-align: center;"><i>Operations</i></p> <pre>public Search(FormAudioPlayer _formAudioPlayer, boolean _runnig) public void showBuscador() public void setMatrizInfoTemp(String temp) public void llenarLista(int indice) public void agregarListaCurso() public void invocarServlet()</pre>
<p style="text-align: center;"><i>Operations Redefined From FormAudioPlayer</i></p> <pre>public void actionPerformed(ActionEvent ae)</pre>

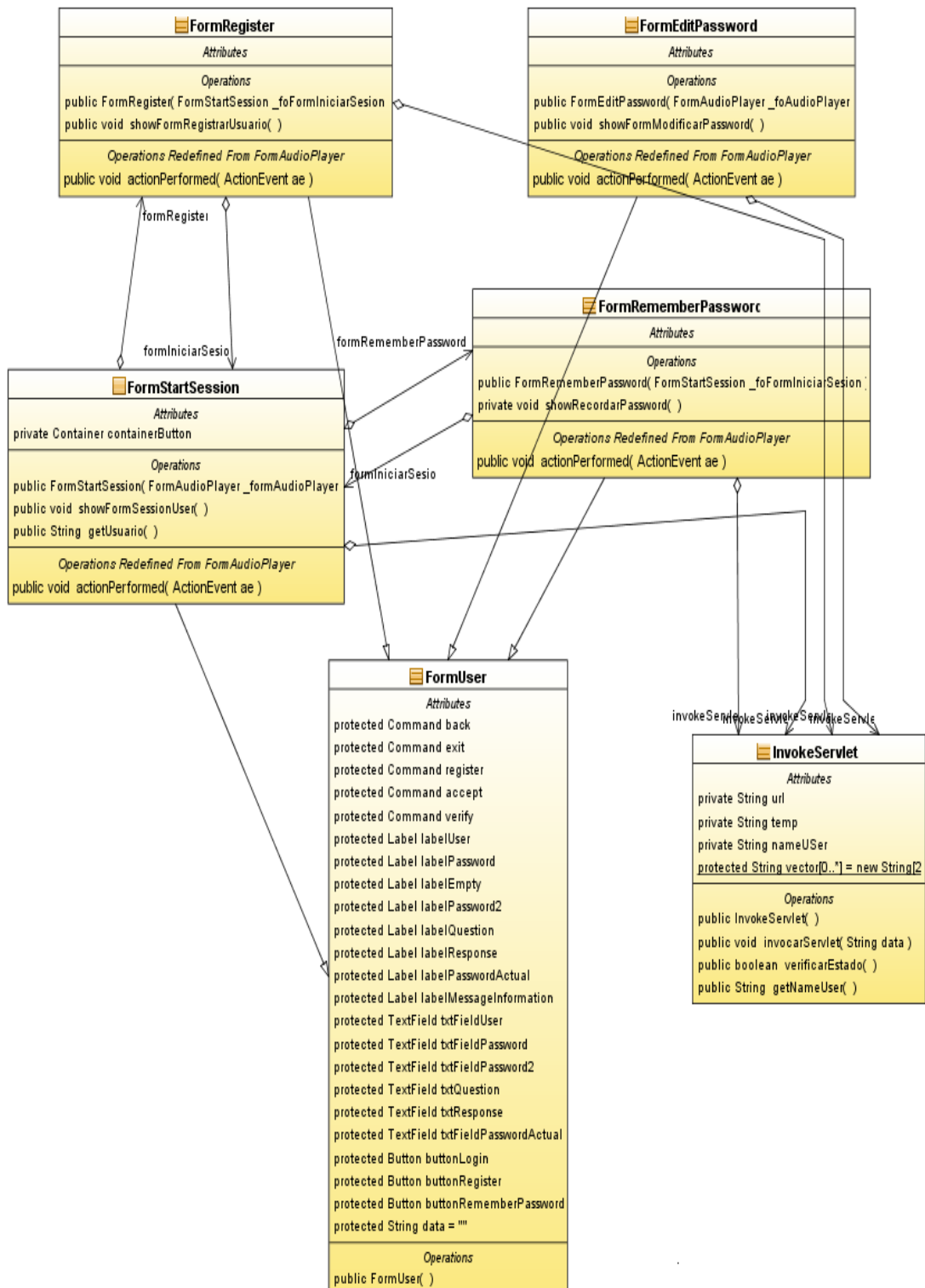
Paquete Protocolos



Paquete Reproductor



Paquete Usuario



Anexo E. Pruebas JMunit

Clase	Método	Intentos Fallidos*
RTCPProtocol	readPacketSR(byte[], short)	5
	readPacketSDES(byte[], short)	4
	readPacketAPP(byte[])	3
	convertV_P_RC(byte, byte, byte)	2
	convertTo_1Byte(short)	2
ControlVolume	control(Player)	2
	controlVolume(int, int)	3
	controlVolume(boolean)	1
InvokeServlet	invokeServlet	4
	verifyState	2

*Todos los intentos fallidos se daban cuando el método no funcionaba de forma correcta inicialmente, las pruebas finales eran exitosas.