

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

CONTROL DE DESCONEXIÓN EN REDES WIFI PARA
DISPOSITIVOS MÓVILES BASADOS EN SISTEMA
OPERATIVO ANDROID PARA SERVICIOS DE
VIDEOSTREAMING

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR:

CRISTINA ALEXANDRA NAVARRO RUBIO
DARÍO JAVIER LARENAS ARRESE

SANGOLQUÍ, octubre de 2010

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por la Srta. Cristina Alexandra Navarro Rubio y el Sr. Darío Javier Larenas Arrese, como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Octubre de 2010

ING. DIEGO MARCILLO

DIRECTOR

ING. DANILO MARTÍNEZ

COORDIRECTOR

DEDICATORIA

Le dedico a Dios todo mi trabajo y esfuerzo, quien me ha dado las fuerzas para continuar.

A mis padres, Dolores Rubio y José Navarro, pilar fundamental de mi vida, a quienes les admiro y les amo con todo mi corazón. Gracias a su apoyo, a su amor, a su esfuerzo, a sus ánimos, hoy he podido dar un gran paso en mi vida.

CRISTINA ALEXANDRA NAVARRO RUBIO

Este proyecto va dedicado a todos mis familiares que desde el cielo me inspiran, me guían y siempre tendrán un espacio en mi corazón, donde se mantendrán vivos en mis recuerdos.

DARÍO JAVIER LARENAS ARRESE

AGRADECIMIENTO

Agradezo a Dios por iluminar mi camino, darme fuerzas para vencer todos los obstáculos y culminar con éxito mi carrera.

A mi padres Dolores Rubio y José Navarro, dignos de ejemplo, de trabajo y de constancia quienes me han brindado su amor incondicional, han estado en todos los momentos de mi vida apoyandome, alentándome a seguir adelante. Y quienes han dado lo mejor de ellos para que yo pueda cumplir mis metas y sueños.

A mis hermanos Walter y Javier quienes con su ejemplo, cariño, consejos y ayuda han hecho que mi largo camino sea más fácil.

A mis sobrinas que con sus sonrisas y bromas inocentes me han dado ánimos día a día.

A mi compañero de tesis, por la paciencia, el cariño y el apoyo que me ha brindado durante todo este tiempo que nos conocemos.

A cada uno de los Ingenieros que han compartido su conocimiento y a cada una de las personas que han puesto un granito de arena.

CRISTINA ALEXANDRA NAVARRO RUBIO

Agradezco a Dios y a esos seres maravillosos que son mis padres, Josefina Arrese y César Larenas, porque siempre han estado a mi lado, me han mostrado el camino del éxito y me han apoyado en toda mi vida dándome cada día su amor y confianza.

Además doy gracias a mi mejor amigo y hermano, Diego Larenas, cuyos consejos y cariño nunca me ha faltado.

A mi compañera, por su paciencia, colaboración y sobre todo por su amistad que me ha brindado.

También un profundo agradecimiento a mis profesores por sus sabias enseñanzas, que han hecho de mí, un profesional de éxito.

DARÍO JAVIER LARENAS ARRESE

Tabla de Contenidos

RESUMEN	1
CAPITULO I.....	2
1. INTRODUCCION	2
1.1 PLANTEAMIENTO DEL PROBLEMA	3
1.1.1 Conceptualización del problema	3
1.1.2 Formulación del problema.....	4
1.1.3 Delimitación espacial	6
1.1.4 Delimitación temporal	6
1.2 Objetivos	7
1.2.1 Objetivo General.....	7
1.2.2 Objetivos Específicos	7
1.3 Justificación	8
1.4 Alcance	8
1.5 Hipótesis de Trabajo	9
1.6 Metodología	9
CAPITULO II.....	10
2 MARCO TEORICO	10
2.1 REVISIÓN DEL ESTADO DE ARTE	10
2.2 Android	10
2.2.1 Historia	10
2.2.2 Versiones	11
2.2.3 Arquitectura	14
2.2.4 Máquina Virtual Dalvik VM	15
2.2.5 Componentes de las Aplicaciones Android.....	15
2.2.6 Ciclo de Vida de las Actividades.....	17

2.3	Agentes inteligentes	19
2.3.1	JADE	19
2.3.2	Agente JADE.....	22
2.3.3	Ciclo de vida de un agente.....	23
2.3.4	Estados de un agente.....	23
2.3.5	Comunicación entre Agentes.....	28
2.3.6	LEAP	30
2.4	Protocolos de transporte.....	31
2.5	UDP	31
2.5.1	CARACTERÍSTICAS.....	31
2.5.2	Cabecera UDP	32
2.5.3	Aplicaciones sobre UDP.....	34
2.6	TCP.....	34
2.6.1	Propiedades.....	35
2.6.2	Fases de TCP	35
2.6.3	Encapsulado de TCP.....	36
2.6.4	Formato de Segmentos TCP.....	37
2.7	Diferencia entre TCP y UDP	38
2.8	Protocolos de comunicación	38
2.9	PROTOCOLO RTSP	39
2.9.1	Características.....	39
2.9.2	Similitudes con HTTP	40
2.9.3	Diferencias con HTTP.....	40
2.9.4	Componentes	40
2.9.5	Modos de Operación.....	40
2.9.6	Métodos RTSP.....	42
2.9.7	Formato URL.....	42

2.9.8	Formato del Mensaje de Petición	43
2.9.9	Formato del Mensaje de Respuesta	43
2.9.10	Negociación RTSP	44
2.10	PROTOCOLO RTP	50
2.10.1	Funcionamiento	51
2.10.2	Cabecera RTP	51
2.11	PROTOCOLO RTCP	53
2.11.1	Funciones.....	53
2.11.2	Paquetes RTCP	53
2.12	Streaming	54
2.12.1	Proceso Streaming	55
2.13	Contenedor de video 3gp	56
2.14	Códec	57
2.15	H.263.....	58
2.16	H.264.....	58
2.16.1	Estructura de las capas.....	59
2.17	AAC.....	59
2.18	AMR	60
2.19	Metodología	60
2.19.1	Etapas	61
2.20	Herramientas de Programación Específica	61
2.20.1	Java Development Kit	61
2.20.2	ANDROID SDK.....	62
2.20.3	JADE-LEAP	62
2.20.4	JADE-ANDROID.....	62
2.21	Rendimiento de los servicios para video streaming.....	65
2.21.1	RETARDO	65

2.21.2	JITTER	66
2.22	Factores que provocan la desconexión WIFI.....	67
2.23	Métodos estadísticos	67
2.23.1	Muestreo	67
CAPITULO III		70
3	DESARROLLO DE LA APLICACION	70
3.1	Introducción.....	70
3.2	Planificación	71
3.2.1	Historias de Usuario	71
3.2.2	Plan Realese.....	73
3.3	Diseño	75
3.3.1	Paso de imágenes con Agente JADE.....	75
3.3.2	Paso de video streaming utilizando el protocolo RTSP y agentes JADE	76
3.3.3	Aplicación Jade -Streaming.....	76
3.3.4	Diagramas	82
3.3.5	Diagrama de casos de Uso.....	82
3.3.6	Diagrama de clases APS.....	83
3.3.7	Diagrama de Secuencia APS	87
3.3.8	Diagrama de clases APC	88
3.3.9	Diagramas de Secuencia APC	93
3.3.10	Diagrama de Actividades.....	95
3.4	Codificación.....	99
3.5	Pruebas.....	99
3.5.1	Reproducir video sin Proxy	99
3.5.2	Reconexión automática.....	100
3.5.3	Creación de Agentes Jade de manera dinámica.....	105
3.5.4	Configurar Proxy	105

CAPITULO IV	107
4 IMPLANTACION Y PRUEBAS	107
4.1 Introducción	107
4.2 Indicadores de rendimiento del servicio	110
4.3 Benchmarking	178
4.4 Jitter	178
4.4.1 Paquetes de Audio	178
4.4.2 Paquetes de Video	179
4.5 Retardo	179
4.5.1 Paquetes de Audio	179
4.5.2 Paquetes de Video	180
4.6 Pérdida de Paquetes	181
4.6.1 Paquetes de Audio	181
4.6.2 Paquetes de video	182
4.7 Ajuste y reconfiguración.....	182
4.7.1 Obtención real del tamaño de los paquetes recibidos del servidor VLC.....	182
4.7.2 Sincronizar Timestamp para mejoras a futuro.....	184
4.8 Obtención de resultados analíticos y discusión.	186
4.9 Determinar condiciones, factores y ajustes a fin de conseguir que la interrupción sea mínima.	187
CAPITULO V	188
5 CONCLUSIONES Y RECOMENDACIONES.....	188
5.1 Conclusiones Generales	188
5.2 Recomendaciones	189
ABREVIATURAS	191
GLOSARIO	194
ANEXOS	199
1 Anexo 1- Manual de Instalación	199

1.1	Instalación de Android SDK.....	199
1.2	Instalación de plugins y herramientas de programación.....	201
1.2.1	Instalación de JAVA.....	201
1.2.2	Instalación de Eclipse	204
1.2.3	Instalación Plugin de Android	205
1.2.4	Configuración del Emulador de Android	208
1.2.5	Configuración de JADE	210
2	Anexo 2.-Manual del Usuario	214
2.1.1	Requerimientos Hardware	214
2.1.2	Instalación del Software	214
2.1.3	Funcionamiento del Software.....	214
3	Anexo 3.-Código de la Aplicación.....	216
3.1.1	Cliente Android - APC	216
3.1.2	Servidor Proxy - APS	266
	BIBLIOGRAFÍA	291
	TRABAJOS CITADOS.....	294

LISTADO DE TABLAS

Tabla 2.1 - Versiones SDK Android	11
Tabla 2.2 - Códigos de Respuesta RTP	44
Tabla 2.3 - Versiones de Eclipse	64
Tabla 3.1 - Herramientas y Versiones	70
Tabla 3.2 - Historia de Usuario 1	71
Tabla 3.3 - Historia de Usuario 2	72
Tabla 3.4 - Historia de Usuario 3	72
Tabla 3.5 - Plan Release	73
Tabla 3.6 - Comportamientos APC	78
Tabla 3.7 - Comportamientos APS.....	80
Tabla 3.8 - Caso de Prueba 1	99
Tabla 3.9 - Caso de Prueba APS - APC 2	100
Tabla 3.10 - Caso de Prueba APS -APC 3	102
Tabla 3.11 - Caso de Prueba APS -APC 4	103
Tabla 3.12- Caso de Prueba APS –APC 5.....	104
Tabla 3.13 - Caso de Prueba 6 APS.....	105
Tabla 3.14- Caso de Prueba 7 APS –APC.....	105
Tabla 4.1 - Muestra Escenario 1	109
Tabla 4.2 - Muestra Escenario 2	109
Tabla 4.3 - Cálculo del Jitter y Retarco con los Paquetes de Video.....	110
Tabla 4.4 - Cálculo de Jitter y Retardo paquetes audio	144
Tabla 2.1- Opciones de Configuración.....	215

LISTADO DE FIGURAS

Ilustración 2.1 - Arquitectura Android	14
Ilustración 2.2 - Proceso de Generación del formato dex.....	15
Ilustración 2.3 - Ciclo de Vida.....	18
Ilustración 2.4- Arquitectura de Jade	21
Ilustración 2.5 –Plataformas y Contenedores	22
Ilustración 2.6 - Máquina de Estados Finita	26
Ilustración 2.7 - Comportamiento Secuencial	27
Ilustración 2.8 - Comportamiento Paralelo	27
Ilustración 2.9 - Encapsulamiento IP.....	32
Ilustración 2.10 - Cabecera UDP	32
Ilustración 2.11 - Pseudocabecera	34
Ilustración 2.12- Encapsulado de un segmento TCP en un datagrama	36
Ilustración 2.13 - Formato de un segmento TCP	37
Ilustración 2.14 – Unicast.....	41
Ilustración 2.15 – Multicast	41
Ilustración 2.16 - Negociación RTSP	50
Ilustración 2.17 - Cabecera RTP.....	51
Ilustración 2.18- Ciclo XP	61
Ilustración 3.1- Arquitectura Imágenes	75
Ilustración 3.2 - Arquitectura JADE - RTSP.....	77
Ilustración 3.3 - Interfaz y Reproducción Video	100
Ilustración 3.4 - Paso de Imágenes	101
Ilustración 3.5 – Reconexión.....	101
Ilustración 3.6 - Paso de Video con Jade.....	102
Ilustración 3.7- Jade y Rtsp	103
Ilustración 3.8 - Interfaz y Video Emitido.....	104
Ilustración 3.9 - Interfaz y Menú configuración del Cliente	106
Ilustración 4.1 - Paso de mensajes ACL.....	107
Ilustración 4.2 - Jitter Paquetes de Video	178
Ilustración 4.3 - Jitter Paquetes de Video	179
Ilustración 4.4 - Retardo Paquetes de Audio	179
Ilustración 4.5 - Retardo Paquetes de Video	180
Ilustración 4.6 - Pérdida de Paquetes Audio.....	181

Ilustración 4.7 - Pérdida de Paquetes Video.....	182
Ilustración 4.8 - Error de Código.....	183
Ilustración 4.9 - Corrección.....	184
Ilustración 1.1 - Android SDK.....	199
Ilustración 1.2 - Paquetes Disponibles.....	200
Ilustración 1.3 - Descarga SDK.....	201
Ilustración 1.4 - Descarga JAVA.....	201
Ilustración 1.5 - Directorio JDK.....	202
Ilustración 1.6 - Fin de instalación.....	202
Ilustración 1.7 - Variables de Entorno.....	203
Ilustración 1.8 - Variable PATH.....	203
Ilustración 1.9 - Variable CLASSPATH.....	203
Ilustración 1.10 – Descarga Eclipse.....	204
Ilustración 1.11 - Menú Eclipse.....	205
Ilustración 1.12 - URL.....	205
Ilustración 1.13 - Android Tools.....	206
Ilustración 1.14 - Instalación.....	206
Ilustración 1.15 - Configuración Android SDK.....	207
Ilustración 1.16 - Localización SDK.....	208
Ilustración 1.17 - Paso 3 y 4.....	209
Ilustración 1.18 - Configuración emulador.....	209
Ilustración 1.19 - CMD Ant.....	210
Ilustración 1.20 - Página Oficial JADE.....	211
Ilustración 1.21 - Compilación JADE.....	211
Ilustración 1.22 - add-ons.....	212
Ilustración 2.1 - Ícono.....	214
Ilustración 2.2 - interfaz de la aplicación.....	215
Ilustración 2.3 - Menú.....	216

LISTA DE ANEXOS

Anexo1. – Manual de Instalación	199
Anexo2. – Manual de Usuario	214
Anexo3. – Código de la Aplicación	217

RESUMEN

Con la penetrante adopción de las redes inalámbricas sobre dispositivos móviles en ámbitos empresariales, residenciales y públicos, con base en los estándares Wireless Fidelity – WiFi, Worldwide Interoperability for Microwave Access - WiMAX, Bluetooth y 3rd Generation Partnership Project - 3GPP - 3GPP2, ha permitido visualizar contenidos multimedia en tiempo real mediante la utilización de técnicas de compresión streaming.

El problema para poder implementar visualización de contenidos multimediales en dispositivos móviles en tiempo real radica en dos aspectos: limitaciones de memoria y un comportamiento irregular de los sistemas de comunicación inalámbricos, que dispone el dispositivo, debido a rupturas de los enlaces por varias causas (deterioro de la señal, fuera de cobertura).

Una técnica que permite administrar los inconvenientes anteriormente descritos es streaming, mediante la segmentación del video (pequeños streams), que son enviados desde un servidor hacia el móvil para su visualización inmediata sin requerir la descarga total del contenido multimedial, en el móvil para su visualización.

Este proyecto describe un mecanismo para el control de desconexión sobre interfaces de redes inalámbricas WiFi, para servicios de video streaming en dispositivos móviles con sistema operativo android, mediante la implementación de un proxy basado en agentes de software sobre plataforma JADE-LEAP, indicado aquí la utilización de herramientas específicas, metodologías y protocolos para el desarrollo de una aplicación android, complementada con la potencia de los agentes Jade y sus diversas plataformas, donde el objetivo de la misma es controlar la desconexión que puede estar condicionado, por diversos factores así como limitaciones de las herramientas y de los protocolos a utilizar.

CAPITULO I

1. INTRODUCCION

Los teléfonos móviles ahora ya no sólo sirven para hablar con otra persona o enviar mensajes de textos, hoy son dispositivos poderosos capaces de enviar así como reproducir audio y video, tanto así que los mismos se convertirán en los PC's del futuro.

La tendencia de hoy del desarrollo de aplicaciones, lenguajes de programación, desarrollo de frameworks y sistemas operativos se están inclinando hacia los dispositivos móviles, convirtiéndose en la nueva tecnología codiciada por todos.

A la vanguardia de estos avances se encuentran diversas plataformas como: Symbian OS, Windows Mobile de Microsoft, iPhone OS de Apple y Android de Google. Siendo la mayoría de estas propietarias limitando a los desarrolladores a generar nuevas aplicaciones, es por esto que Android toma ventaja ya que es una plataforma de código abierto.

Todas estas tecnologías más los avances en las redes WIFI¹ , HSDPA² , han hecho posible desarrollar nuevas aplicaciones para los dispositivos móviles, en donde la multimedia juega un papel muy importante y depende del servicio de streaming.

Compañías líderes en video streaming como Youtube están empezando apostar a la tecnología móvil, pero se enfrentan a problemas de desconexión siendo estos muy molestos para el usuario final. Para estos inconvenientes existe una interesante solución basada en código JAVA, la plataforma JADE-LEAP, que posee agentes inteligentes a los cuales se les puede encargar cualquier tarea o comportamiento, como por ejemplo ser los protagonistas en el control de la conexión en el servicio streaming. Su ventaja es que no

¹ Wireless Fidelity

² High Speed Downlink Packet Access.

utilizan gran cantidad de memoria en el terminal móvil, además de enviar varios tipos de mensajes sirven para controlar estados de envío y recepción de información.

Por lo tanto en la presente investigación se busca evaluar la tecnología de agentes basada en JADE, como mecanismo de control de conexiones sobre redes WiFi y la técnica de video streaming en los dispositivos móviles Android.

1.1 PLANTEAMIENTO DEL PROBLEMA

1.1.1 Conceptualización del problema

El Vicerrectorado de Investigación y Vinculación con la Colectividad de la ESCUELA POLITECNICA DEL EJERCITO, con el objetivo de incentivar y potenciar nuevos talentos entre los estudiantes de pregrado, tecnologías y postgrado en el desarrollo de la investigación, convoca a la comunidad politécnica a presentar Proyectos de Investigación Científica y Tecnológica que se desarrollen como tesis de graduación, para formar parte del portafolio de proyectos del programa de INICIACION CIENTIFICA de la ESPE - 2010.

En este contexto, se presenta el plan de tesis, con título: **“CONTROL DE DESCONEJIÓN EN REDES WIFI PARA DISPOSITIVOS MÓVILES BASADOS EN SISTEMA OPERATIVO ANDROID PARA SERVICIOS DE VIDEOSTREAMING”**, como requisito previo a la obtención del título de Ingenieros de Sistemas e Informática.

El desarrollo de la investigación y la implementación con la tecnología android permite el desarrollo de aplicaciones para dispositivos móviles. Los teléfonos móviles son dispositivos poderosos con varias interfaces de red inalámbricas heterogéneas que permiten recibir la información multimedia a través del método streaming a bajo costo desde un servidor de vídeo. Streaming es una técnica que consiste en objetos multimedia que se reproducen y descargan simultáneamente desde el Internet sin la necesidad de almacenar toda la información en la memoria del cliente.

En los últimos años algunos protocolos y mecanismos han sido elaborados para soportar transacciones móviles en canales inalámbricos no seguros. El comportamiento caótico de los canales inalámbricos en especial de Wifi causa roturas de comunicación impredecibles que impide al móvil comunicarse por instantes de tiempo, ocasionando que el servidor finalice la sesión streaming, convirtiéndose en un problema importante porque el cliente debe negociar una nueva sesión y la técnica streaming pierde eficiencia. Este problema puede aparecer mientras el móvil está ejecutando un proceso roaming, mientras sale de cobertura. Por otro lado el servidor continúa enviando información mientras el cliente está fuera de cobertura, ocasionando gastos de recursos del servidor.

Es importante controlar la disponibilidad del móvil y establecer un mecanismo transparente que permita al cliente recuperar las tramas multimedia perdidos en caso de darse quiebres de comunicación. También es importante permitir al usuario continuar visualizando las tramas multimedia sin perder la semántica.

En base a un estudio anterior en el que se ha logrado controlar la desconexión en redes WiFi para celular Nokia N95 usando servicios de video streaming, en el presente proyecto se busca definir el mecanismo de control de conexiones intermitentes en redes WiFi al usarse video streaming en los dispositivos móviles android, mediante la tecnología de agentes basada en JADE se propone un mecanismo de reanudación automático a través de aplicaciones streaming sobre enlaces inalámbricos no seguros para controlar la información multimedia perdida durante los quiebres de cobertura.

1.1.2 Formulación del problema

Los avances en microelectrónica, la comunicación de banda ancha con nuevas tecnologías como: WiFi³, HSDPA (High Speed Downlink Packet Access)⁴, han permitido

³ The Working Group for Wireless LANs", IEEE P802.11. [Online]. Available: <http://grouper.ieee.org/groups/802/11/>.

el desarrollo de diversos servicios para dispositivos portables ; uno de los servicios para acceder a contenidos multimedia se denomina “streaming” es una técnica optimizada de alto rendimiento usado en dispositivos móviles que permite visualizar parte de un objeto multimedia mientras otra parte está siendo bajada y otra está en la red y en el servidor, no almacena datos en la memoria del cliente.

Streaming es muy apropiada para aplicaciones cliente que corren en dispositivos portables (celulares, PDA, mp3, GPS) debido a su limitada memoria y ancho de banda; esta técnica trata de ocultar la latencia de la red en una sesión entre el Cliente y el Servidor de vídeo, consiguiendo que el tiempo de espera por tramas de video consecutivas sea mínimo.

Con streaming es posible acceder a contenidos de Televisión Digital que proporcionan cadenas de televisión (como por ejemplo⁵) esto permite el aumento de sus cuotas de audiencia a un precio razonable. La ventaja de streaming es que no requiere que el video se almacene en el teléfono móvil dejando espacio para albergar programas eficientes y poderosos en los terminales móviles incluso aquellos que permitan tratar el vídeo de forma eficiente.

Los dispositivos portables incluyen comunicación inalámbrica cuyo comportamiento es impredecible y no seguro es bien conocida la conducta caótica espacio – temporal de los canales inalámbricos⁶ (todas las tecnologías sufren de este inconveniente, en especial Wifi que provoca impredecibles pérdidas de cobertura).

Streaming en redes no seguras es un problema que aún no ha sido solucionado eficientemente, existe dificultad de controlar las desconexiones intermitentes ocasionados por pérdida de cobertura u otras causas (la pérdida de conexión puede ser por naturalezas muy variadas y producirse a diferentes niveles de la arquitectura de Red); surgen entonces

⁴ Harri Holma, Antti Toskala, HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications, Wiley, 2006, ISBN: 978-0-470-01884-2.

⁵ “TVE a la Carta”, TVE. [Online]. Available: <http://www.rtve.es/alacarta/index.html>.

⁶ IEEE P802.11, “The Working Group for Wireless LANs”. Disponible: <http://grouper.ieee.org/groups/802/11/>.

dificultades de recepción de datos multimedia, cierre de la sesión actual y obligatoriedad de abrir una nueva sesión, gastos de los recursos del servidor al enviar tramas multimedia desconociendo el estado real del cliente; todo esto confluye a la pérdida de efectividad de streaming y a la ineficiencia de protocolos como RTSP, ocasionando también pérdida de tiempo del usuario que abandona la sesión de forma definitiva.

1.1.3 Delimitación espacial

La investigación será realizada con equipos proporcionados por el Departamento de Ciencias de la Computación, utilizando software de código abierto y de libre distribución.

1.1.4 Delimitación temporal

El presente proyecto será analizado dentro de los lineamientos de la investigación científica y en base al tema desconexiones intermitentes en redes WiFi usando servicios de video streaming para android.

Las tecnologías de dispositivos móviles están siendo utilizadas en la educación, investigación y a nivel empresarial. Sin embargo, a la fecha existe un trabajo de arquitectura de proxies que soluciona las pérdidas intermitentes de desconexión cuando se realiza streaming con el teléfono móvil Nokia N95⁷, en este se evita la pérdida de información al recibir un flujo de vídeo completo (recibiendo en diferido los datos que se hubieran perdido por efecto de la desconexión) para ello se ha utilizado la plataforma de agentes JADE-LEAP^{8,9}, la razón más importante del uso de JADE-LEAP es que los propios agentes proporcionan mecanismos de buffering necesario para no provocar pérdidas de fotogramas y de la sesión de vídeo streaming, además la solución se programa en el lenguaje Java y no hay necesidad de reprogramar ni en el Cliente ni en el Servidor.

⁷"Nokia N95", Nokia Inc. [Online]. Available: http://www.nokia.es/link?cid=PLAIN_TEXT_815211.

⁸ (Bellifemine, Caire, Poggi, & Rimassa, septiembre 2003)

⁹ (Lixin, Zhi-Li, & Towsley, December 2003)

El presente proyecto se apoya en el trabajo citado anteriormente para definir el mecanismo de control de desconexiones intermitentes en redes WiFi al usarse video streaming en los dispositivos móviles. Los dispositivos móviles pueden albergar software especializado como son los intermediarios (proxies) que soportan un servicio de calidad de reproducción de video streaming. El mecanismo se basa en la plataforma JADE-LEAP generando agentes software¹⁰(proxies)¹¹ que no utilizan gran cantidad de memoria del terminal móvil y resuelven las desconexiones utilizando un esquema de memorización intermedia (buffering) proactivo¹², con ello se evita la pérdida de información al recibir un flujo de vídeo completo pues se recibe en diferido los datos que se hubieran perdido por efecto de la desconexión.

1.2 Objetivos

1.2.1 Objetivo General

- Generar un mecanismo que controle las desconexiones del teléfono celular android al ejecutar video streaming en un enlace WiFi y garantice la recepción completa de la información multimedia.

1.2.2 Objetivos Específicos

- Recuperar la sesión streaming automáticamente en un android mediante el soporte eficiente de quiebres de comunicación inalámbrica.
- Desarrollar un software que trabaje en el android y garantice la recepción completa del flujo de información multimedia.
- Lograr que la comunicación streaming sea de calidad al transmitir información multimedia en diferentes formatos o estándares.

¹⁰ (Lixin, Zhi-Li, & Towsley, December 2003)

¹¹ (Bellavista, Corradi, & Giannelli, June 2005)

¹² (Vandalore, Feng, Jain, & Fahm, 1999)

1.3 Justificación

El conocer con exactitud los resultados producidos por este proyecto, beneficiará principalmente a Empresas públicas y privadas a nivel internacional con mayor porcentaje de empresas y usuarios que utilicen video streaming, además existirá un incremento en el posicionamiento de televisión digital en teléfonos móviles.

Disminución de pérdidas de recursos provocados por la inseguridad de canales inalámbricos de transmisión, obteniendo una mayor demanda para los servicios de video streaming

1.4 Alcance

El desarrollo del presente tema de tesis se enmarca dentro de la convocatoria para presentar Proyectos de Iniciación científica que realiza el Vicerrectorado de Investigación para el año 2010 y tiene como alcance realizar pruebas de video streaming en una red Wifi.

Esta investigación se desarrollará con plataforma android de código abierto y de libre distribución. Los resultados de esta investigación permitirán:

- Proveer al mercado de un mecanismo que controle las desconexiones generadas por el enlace de comunicación inalámbrico y no seguro al ejecutar sesiones de streaming entre el Cliente (teléfono celular android) y el Servidor (servidor de video).
- Determinar los factores que incrementan la penalización del rendimiento computacional en la realización de experimentos de dimensionado de redes.
- Incrementar el uso de teléfonos móviles para servicios de Televisión Digital, Multimedia entre otros

1.5 Hipótesis de Trabajo

La plataforma android y JADE mejoran significativamente el servicio de video streaming.

1.6 Metodología

A continuación se describen los pasos a seguir:

- Investigar la tecnología android y las distintas plataformas que esta posee.
- Investigar los factores que afectan el Rendimiento del servicio de video streaming.
- Investigar las herramientas de programación específica.
- Implementar el software en plataforma android.
- Verificar los resultados.
- Evaluar y discutir los resultados experimentales.

CAPITULO II

2 MARCO TEORICO

2.1 REVISIÓN DEL ESTADO DE ARTE

2.2 Android

En el año de 2005 Google entró en el negocio de los dispositivos móviles comprando la compañía Android Inc, con la finalidad de desarrollar una Plataforma que satisface las necesidades de los operadores móviles, con la característica adicional de código abierto, es así como en el año 2008 se lanza la Plataforma Android.

2.2.1 Historia

El lanzamiento inicial del Android Software Development Kit fue en noviembre de 2007 y a mediados de agosto de 2008 apareció el Android 0.9 SDK en su versión beta.

En septiembre 2008 google lanzó Android 1.0 SDK. Seis meses después presentó la versión 1.1 de Android para el “dev phone”, esta actualización incluía algunos cambios estéticos.

En mayo del 2009 se crea Android 1.5, la versión que más personas usaron para iniciarse en Android (con el T-Mobile G1 y HTC Dream en USA) y sigue siendo actualmente una versión que se encuentra disponible en muchos móviles

En septiembre del 2009 aparece Android 1.6 “Donut” con mejoras en las búsquedas, indicador de uso de batería.

La versión SDK 2.1 de Android aparece en Enero del 2010, la cual incluye cambios en sus APIs y corrige errores de versiones anteriores.

Android 2.2 es la última versión, estrenada el 20 Mayo 2010, la cual trae nuevas características y es la que se encuentra en crecimiento.

2.2.2 Versiones

En la siguiente tabla se muestra las diferentes versiones de Android SDK y con sus respectivas características¹³:

Tabla 2.1 - Versiones SDK Android

VERSIÓN	CARACTERÍSTICAS
1.1	9 de Febrero de 2009
1.5 (Cupcake) Basado en Linux Kernel 2.6.27	<ul style="list-style-type: none"> • Rediseño completo de todos los elementos de la interfaz. • Transiciones animadas entre ventanas. • Mejoras en la velocidad de la cámara. • Menor tiempo de búsqueda de los satélites GPS¹⁴. • Mejoras en la velocidad del navegador web. • Intérprete JavaScript.¹⁵ • Añade la posibilidad de copiar y pegar texto y buscar texto dentro de una página web. • Posibilidad de personalizar los widgets¹⁶ mostrados en la pantalla de inicio. • Añade la posibilidad de grabar y reproducir vídeos.
1.6 (Donut) Basado en Linux Kernel 2.6.29	<ul style="list-style-type: none"> • Adiciona el Quick Search Box, una caja de búsqueda en la pantalla de inicio con autocompletado y capacidad de aprendizaje. • Mejora la velocidad de la cámara. • Posibilidad de conectarse a redes VPN¹⁷, 802.1x¹⁸.

¹³ <http://es.wikipedia.org/wiki/Android>.

¹⁴ Global Positioning System – Sistema de Posicionamiento Global.

¹⁵ Lenguaje de scripting basado en objetos no tipeado y liviano.

¹⁶ Pequeña aplicación o programa.

	<ul style="list-style-type: none"> • Nueva pantalla para controlar la batería, que permite comprobar qué aplicaciones y servicios son los que más consumen. Desde esta pantalla se puede también parar o desinstalar estas aplicaciones • Las aplicaciones de Android Market aparecen ahora ordenadas por categorías. • Nuevo motor de texto a voz.
<p>2.0 (Eclair) Basado en Linux Kernel 2.6.29</p>	<ul style="list-style-type: none"> • Rediseñó la interfaz del navegador, contando ahora con soporte para distintas características de HTML5. • Zoom digital, modo scene, balance de blanco, efectos de color y modo macro. • Mejoras en el teclado virtual. • Soporte para nuevos tamaños y resoluciones de pantalla. • Contactos rápidos. • Bluetooth 2.1 • Soporte nativo de Facebook. • Mejoras en Google Maps. • Soporte de Microsoft Exchange. • Mejoras en el calendario.
<p>2.1 (Eclair) Basado en Linux Kernel 2.6.29</p>	<ul style="list-style-type: none"> • Reconocimiento de voz, (se puede dictar en lugar de escribir en cualquier campo de texto). • Mejoras en el teclado virtual. • Galería 3D, al estilo Cover Flow. • Nuevas aplicaciones de reloj/tiempo y noticias. • Mejoras en Google Maps: sincronización de nuestros sitios favoritos, modo noche y auto completado de búsquedas. • Google Goggles¹⁹. • Mejoras en la duración de la batería.

¹⁷ Virtual Private Network- Redes virtuales

¹⁸ Norma del IEEE para el control de acceso a red basada en puertos

¹⁹ Aplicación para descripción de imágenes.

<p>2.2 (Froyo) Basado en Linux Kernel 2.6.32</p>	<ul style="list-style-type: none"> • Actualizaciones automáticas para aplicaciones: Las aplicaciones recibirán actualizaciones automáticas, consiguiendo tener siempre la última versión del software. • Soporte WiFi IEEE 802.11n²⁰. • Soporte Flash 10.1 y Adobe AIR 2.5 • Soporte de la API gráfica OpenGL 2.0 • Creación de un compilador JIT que mejora entre 2 y 5 veces en Rendimiento frente a Eclair. • Incorporación del mismo motor de Javascript V8 de Chrome. • Posibilidad de enviar posiciones desde Google Maps al teléfono, usando un plugin del navegador Google Chrome. • Posibilidad de mover una aplicación instalada desde el teléfono a la tarjeta de memoria, y viceversa. • Opciones avanzadas de gestión energética
<p>3.0 (Gingerbread) Based on Linux Kernel 2.6.33 o 34</p>	<ul style="list-style-type: none"> • Lanzamiento 15-16 de octubre de 2010. • Soporte de resoluciones de hasta 1.280x760. • Interfaz de usuario renovada.

²⁰ Estándar IEEE 802.11-2007 para mejorar significativamente el desempeño de la red.

2.2.3 Arquitectura

Los componentes principales del sistema operativo de Android son²¹:

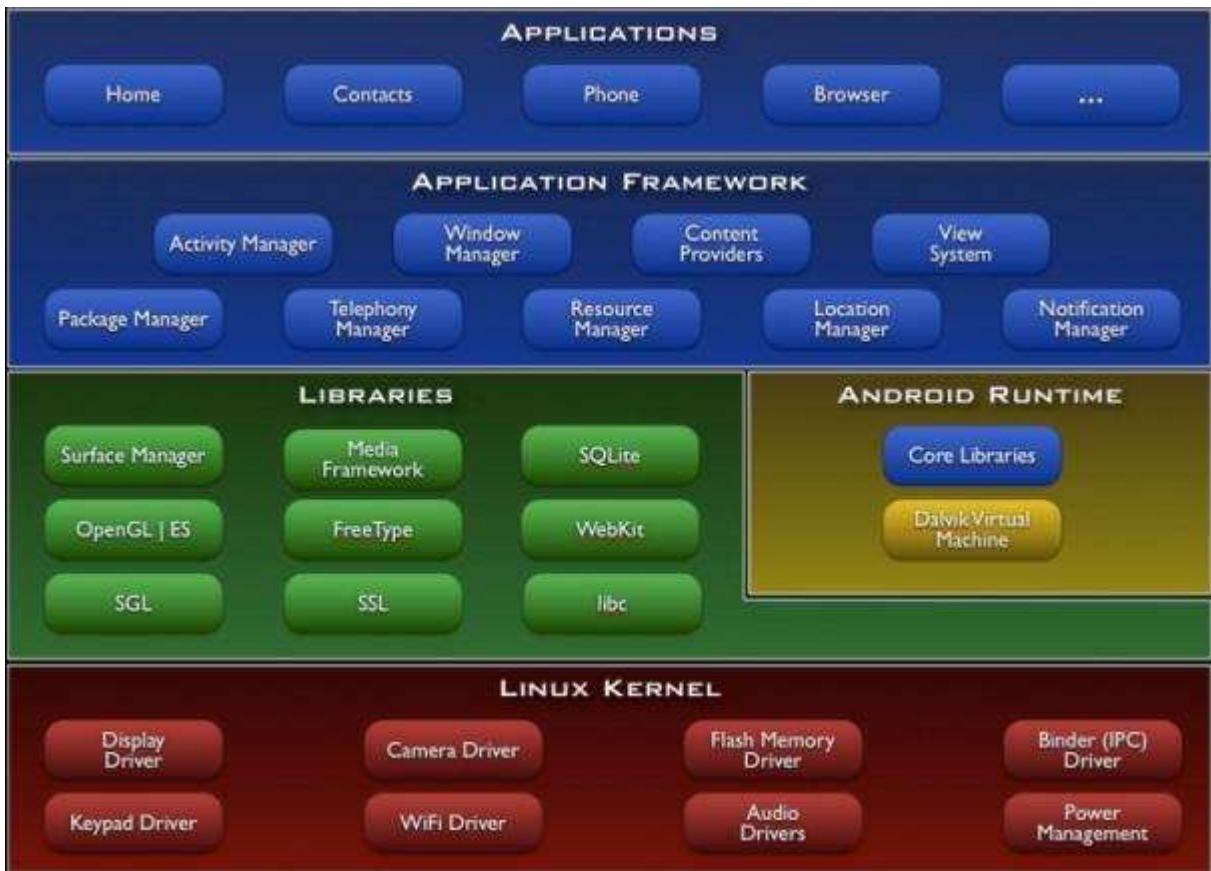


Ilustración 2.1 - Arquitectura Android²²

- **Aplicaciones:** Este nivel contiene, tanto las aplicaciones incluidas por defecto de Android como aquellas que el usuario vaya añadiendo o desarrollando posteriormente.
- **Framework de aplicaciones:** Representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación, las cuales utilizan el mismo conjunto de API y el mismo "framework".

²¹ (Universidad Carlos III de Madrid)

²² <http://developer.android.com/guide/basics/what-is-android.html>.

- **Bibliotecas:** Están escritas en C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Algunas son: System C library (implementación biblioteca C standard), bibliotecas de medios, bibliotecas de gráficos, 3d, SQLite, entre otras.
- **Tiempo de ejecución de Android:** Lo constituyen las Core Libraries, que son librerías con multitud de clases Java y la máquina virtual Dalvik. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik²³.
- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, stack de red, y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto del stack de software.

2.2.4 Máquina Virtual Dalvik VM

Es el nombre de la máquina virtual que utiliza Android (DalvikVM) fue creada diseñada "Dan Bornstein" y algunos ingenieros de Google.

Dalvik VM es un intérprete que sólo ejecuta los archivos con formato .dex²⁴. Este formato está optimizado para el almacenamiento eficiente de la memoria, lo cual consigue delegando en el kernel la gestión de hilos, de memoria y de procesos.



Ilustración 2.2 - Proceso de Generación del formato dex

2.2.5 Componentes de las Aplicaciones Android

Para el desarrollo de las aplicaciones, Android provee de una amplia gama de componentes, los cuales se deben explotar al máximo, para esto se debe conocer los diferentes conceptos de los mismos.

²³ Máquina virtual de Android

²⁴ Dalvik Executable

2.2.5.1 Vista

El concepto de una vista en J2EE²⁵ y Swing²⁶ se traslada a Android. Las vistas son elementos de interfaz de usuario que forman los bloques básicos de construcción de una interfaz de usuario. Las vistas son jerárquicas.

2.2.5.2 Actividad

Una actividad es un concepto de interfaz de usuario. Una actividad por lo general representa una sola pantalla en la aplicación. Contiene uno o más vistas²⁷.

2.2.5.3 Intent

Un Intent consiste básicamente en la voluntad de realizar alguna acción, generalmente asociada a unos datos. Lanzando un Intent, una aplicación puede delegar el trabajo en otra, de forma que el sistema se encarga de buscar qué aplicación de entre las instaladas, es la que puede llevar a cabo la acción solicitada. Se puede usar intents para llevar a cabo las siguientes tareas, por ejemplo:

- Emitir un mensaje
- Inicia un servicio
- Iniciar una actividad
- Mostrar una página web o una lista de contactos
- Marque un número telefónico o contestar una llamada telefónica

²⁵ Java Enterprise Edition

²⁶ Biblioteca gráfica para Java

²⁷ Komatineni, S. Y. (2009). Pro Android. New York: Apress.

2.2.5.4 Content Provider

Con el componente Content Provider, cualquier aplicación en Android puede almacenar datos en un fichero, en una base de datos SQLite o en cualquier otro formato que considere. Además, estos datos pueden ser compartidos entre distintas aplicaciones. Una clase que implemente el componente Content Provider contendrá una serie de métodos que permiten almacenar, recuperar, actualizar y compartir los datos de una aplicación.

2.2.5.5 Services

Representa una aplicación ejecutada sin interfaz de usuario, y que generalmente tiene lugar en segundo plano mientras otras aplicaciones son las que están activas en la pantalla del dispositivo. Un ejemplo típico de este componente es un reproductor de música.

AndroidManifest.xml

Es similar al archivo web.xml en el mundo J2EE, define el contenido y el comportamiento de su aplicación. Por ejemplo, se enumeran las actividades de su aplicación y los servicios, junto con los permisos de la aplicación que necesita para funcionar.

2.2.6 Ciclo de Vida de las Actividades

Android lanza tantos procesos como permitan los recursos del dispositivo. Cada proceso, correspondiente a una aplicación, estará formado por una o varias actividades independientes de esa aplicación. Cuando el usuario navega de una actividad a otra, o abre una nueva aplicación, el sistema duerme dicho proceso y realiza una copia de su estado para poder recuperarlo más tarde. El proceso y la actividad siguen existiendo en el sistema, pero están dormidos y su estado ha sido guardado. Es entonces cuando crea, o despierta si ya existe, el proceso para la aplicación que debe ser lanzada, asumiendo que existan recursos para ello.

Cada uno de los componentes básicos de Android tiene un ciclo de vida bien definido; esto implica que el desarrollador puede controlar en cada momento en qué estado se encuentra

dicho componente, pudiendo así programar las acciones que mejor convengan. El componente Activity, probablemente el más importante, tiene un ciclo de vida como el mostrado en la siguiente figura:

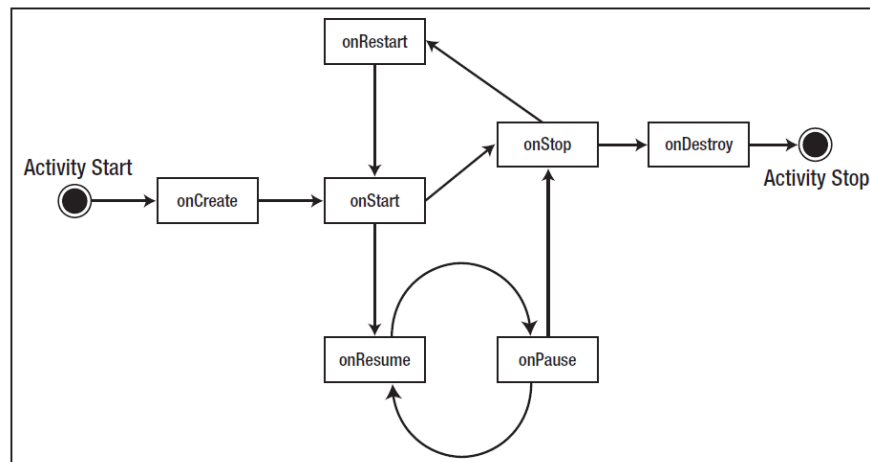


Ilustración 2.3 - Ciclo de Vida²⁸

En la figura anterior se puede observar los diferentes estados²⁹:

- **onCreate(), onDestroy():** Abarcan todo el ciclo de vida. Cada uno de estos métodos representan el principio y el fin de la actividad.
- **onStart(), onStop():** Representan la parte visible del ciclo de vida. Desde onStart() hasta onStop(), la actividad será visible para el usuario, aunque es posible que no tenga el foco de acción por existir otras actividades superpuestas con las que el usuario está interactuando. Pueden ser llamados múltiples veces.
- **onResume(), onPause():** Delimitan la parte útil del ciclo de vida. Desde onResume() hasta onPause(), la actividad no sólo es visible, sino que además tiene el foco de la acción y el usuario puede interactuar con ella.

²⁸ (Komatineni, 2009, pág. 40)

²⁹ (Universidad Carlos III de Madrid)

2.3 Agentes inteligentes

Un agente inteligente, es una entidad capaz de percibir su entorno, procesar tales percepciones y responder o actuar en su entorno de manera racional, es decir, de manera correcta y tendiendo a maximizar un resultado esperado. Todo agente tiene una función u objetivo.

Existe una gran cantidad de software denominado como software de agentes para Linux, Windows y otros sistemas operativos, sin embargo la herramienta para desarrollar agentes más extendida y utilizada es JADE gracias a sus buenas herramientas gráficas, documentación, soporte, licencia LGPL³⁰.

2.3.1 JADE

JADE (Java Agent Development Framework) es un middleware que proporciona tanto un entorno de desarrollo como un entorno de ejecución para la realización y mantenimiento de sistemas multiagente.

El *entorno de desarrollo* está formado por una serie de librerías en Java que permiten la implementación de agentes de manera limpia e independiente de la plataforma sobre la que se va a ejecutar.

El *entorno de ejecución* permite a los agentes vivir y comunicarse entre ellos. Está realizado enteramente en Java y proporciona una serie de herramientas que permiten al desarrollador controlar y depurar a los agentes en tiempo real³¹.

2.3.1.1 Características de JADE:

Es una plataforma para ejecutar agentes que permiten:

³⁰Licencia pública general reducida de GNU

³¹Agente Inteligentes: JADE, Juan Fco. Garamendi Bragado, Abril 2004, Programa Doctorado: Informática y Modelización Matemática, Universidad Rey Juan Carlos, Abril 2004

- Ejecución de agentes completamente asíncrona
- Comunicación entre agentes en la misma o diferentes plataformas JADE/LEAP/JADEX
- Programación de agentes mediante un conjunto de paquetes Java.
- Validación de la ejecución mediante seguimiento mensajes y estado interno del agente.
- Es la plataforma más extendida porque implementa el estándar FIPA³².

2.3.1.2 *Funciones y ventajas:*

- Ejecución distribuida.
- Interfaz gráfica para monitorización y depuración de agentes, incluyendo los que se encuentran en hosts remotos.
- Creación de agentes móviles.
- Ejecución de actividades en paralelo
- Intercambio de mensajes ACL entre agentes.
- Registro automático de agentes (via AMS).
- Servicio de nombres.

2.3.1.3 *Arquitectura*

Los agentes JADE necesitan del entorno de ejecución donde poder "vivir". Cada instancia del entorno de ejecución se denomina contenedor (container). Al conjunto de los contenedores se le denomina plataforma (platform) y proporciona una capa que oculta a los agentes (y al desarrollador) el entorno donde se ha decidido ejecutar la aplicación.

³²Foundation for Intelligent Physical Agents.- Indica como interaccionan y son gestionados los agentes.

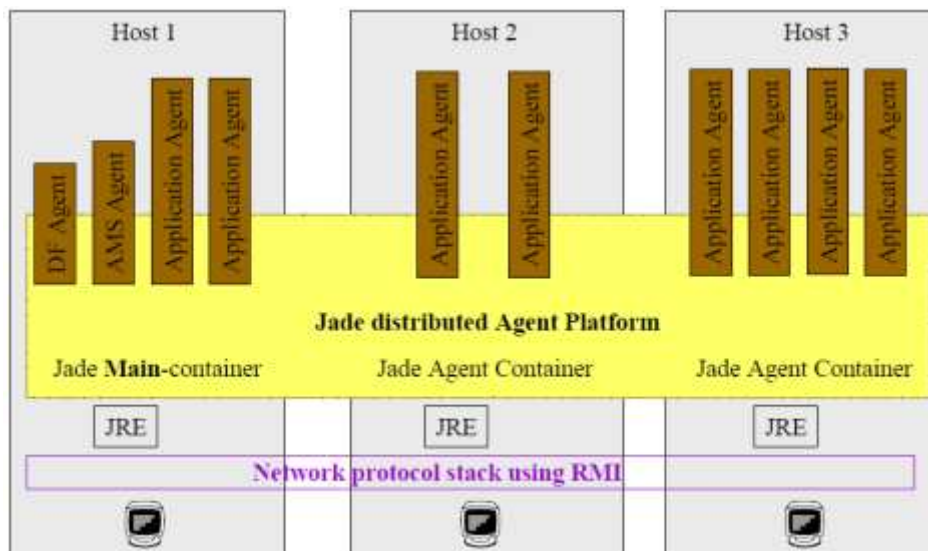


Ilustración 2.4– Arquitectura de Jade³³

En cada plataforma debe existir un contenedor especial denominado contenedor principal (main container). La principal diferencia del contenedor principal respecto al resto de contenedores es que alberga dos agentes especiales:

- AMS (Agent Management System): Este agente proporciona el servicio de nombres asegurando que cada agente en la plataforma disponga de un nombre único. También representa la autoridad, es posible crear y matar agentes en contenedores remotos requiriéndoselo al agente AMS.
- DF (Directory Facilitator): Proporciona el servicio de Páginas Amarillas. Gracias al agente DF, un agente puede encontrar otros agentes que provean los servicios necesarios para lograr sus objetivos.

En la figura 2.5 aparecen dos plataformas diferentes (platform1 y platform2), cada una con sus contenedores principales y contenedores normales.

³³ Curso “Tecnología de Agentes en Inteligencia Ambiental”, Javier Carbó, Grupo de Inteligencia Artificial Aplicada, Campuse de Colmenarejo, Univ. Carlos III de Madrid, 6 de Julio de 2007

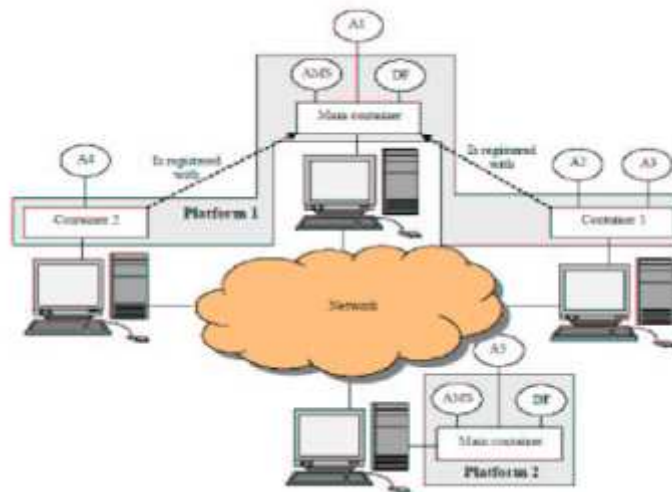


Ilustración 2.5 –Plataformas y Contenedores³⁴

2.3.2 Agente JADE

Los agentes JADE tienen nombres únicos y se permite a cada agente descubrir a otros agentes y comunicarse con ellos mediante comunicaciones **punto a punto**. Los agentes proporcionan servicios, cada agente puede buscar a otros dependiendo de los servicios que proporcionen otros agentes.

La comunicación entre agentes se lleva a cabo a través de mensajes asíncronos, es decir, el agente que envía el mensaje y el destinatario del mensaje no tienen por qué estar disponibles al mismo tiempo. Es más, el destinatario no tiene por qué existir en ese instante. Los mensajes se pueden enviar a un agente en concreto o se pueden enviar a agentes que se desconocen pero se sabe que poseen unas ciertas características.

La estructura de los mensajes se basa en el lenguaje ACL³⁵ que ha sido definido por la FIPA.

³⁴ (Carbó, 2007)

³⁵ Agent Communication Language.

2.3.2.1 Características

- Un agente es un programa JADE de un solo thread.
- Incluye un modelo para programar su funcionamiento basado en tareas:
 - Una tarea= un comportamiento(*behaviour*)
 - Programadas por el agente
 - Tareas operan en paralelo.
 - Permite coordinación explícita entre tareas
- Envía y recibe mensajes a través de una API³⁶: Que facilita la codificación/decodificación de mensajes

2.3.3 Ciclo de vida de un agente

Un agente está sujeto a un ciclo de vida en el que se definen los estados en los cuales se puede encontrar el agente, así como los cambios que se pueden realizar entre los diferentes estados.

El ciclo de vida de un agente JADE sigue el ciclo propuesto por FIPA, es decir, cumple con la propuesta del estándar de interoperabilidad entre agentes más aceptado.

2.3.4 Estados de un agente

Un agente puede estar en los siguientes estados³⁷:

- **Iniciado:** El objeto Agente está creado pero todavía no se ha registrado en el AMS, no tiene nombre ni dirección y tampoco se puede comunicar con otros agentes.
- **Activo:** El Agente está registrado en el AMS, tiene un nombre, una dirección y puede acceder a todas las opciones de JADE.
- **Suspendido:** El Agente está parado. Su hilo de ejecución está detenido y no ejecuta ningún Comportamiento.

³⁶ Application Programming Interface-Interfaz de Programación de Aplicaciones.

³⁷ (WikiSpaces, 2009)

- **En espera:** El Agente está bloqueado esperando por algo. Su hilo de ejecución está dormido en un monitor de java y se despertará cuando se cumpla una cierta condición (cuando reciba un mensaje).
- **Desconocido:** El Agente ha sido eliminado. El hilo de ejecución ha terminado y se ha eliminado del registro del AMS.
- **Tránsito:** Un Agente móvil entra en este estado mientras está migrando a una nueva localización. El sistema sigue guardando los mensajes en el buffer hasta que el agente vuelve a estar activo.

2.3.4.1 *Comportamientos JADE*

Un comportamiento o behaviour hace referencia a una funcionalidad que incorpora el agente.

Los comportamientos especifican tareas o servicios que realiza un agente para lograr sus objetivos. Los agentes JADE programan sus comportamientos con un solo hilo

Cada comportamiento puede realizar una tarea simple como: “envíaEsteMensaje”, “ComprarX”, etc.

2.3.4.2 *Métodos de un comportamiento*

Toda clase que herede de Behaviour deberá implementar³⁸:

- Método action().
 - Define la acción a ser ejecutada cuando inicie el comportamiento.
 - Debe incluir el código de las acciones a realizar.
 - Es invocado cuando se produce el evento asociado al comportamiento.
 - Es recomendable que los métodos action() no tengan un tiempo de ejecución alto ya que mientras que se ejecutan no pueden ser interrumpidos por otro comportamiento.

³⁸ (WikiSpaces, 2009)

- Método done().
 - Es invocado cuando finaliza la ejecución del método action().
 - Este método determina si el comportamiento ha sido completado o no, devuelve un booleano (true si ha terminado o false en caso contrario).
 - Si el comportamiento ha finalizado, éste se elimina de la cola de comportamientos activos.

2.3.4.3 Tipos de comportamientos

Los agentes pueden realizar funcionalidades complejas que pueden llegar a implicar tareas simultáneas forzando a implementar agentes multi-hilo causando problemas.

JADE proporciona un sistema de comportamientos que ayudan al usuario a construir sistemas multiagente y reutilizar código.

El paquete jade.core.behaviours contiene las clases que se usan para implementar comportamientos básicos de agentes.

Se eliminan los problemas de sincronización entre comportamientos concurrentes que acceden al mismo recurso, haciendo que cada agente sea equivalente a un único hilo.

Los tipos de comportamientos de jade se dividen en cuatro grupos³⁹:

2.3.4.3.1 Comportamientos Genéricos:

- Hereda de la clase abstracta Behaviour.
- Mantiene un estado del agente y en base a él se ejecutan diferentes operaciones.
- Finalizan cuando cierta condición se cumple.

2.3.4.3.2 Comportamientos Simples:

Hereda de la clase SimpleBehaviour que representa a comportamientos atómicos, que suelen realizar tareas simples.

³⁹ (WikiSpaces, 2009)

- OneShotBehaviour.- No se puede definir el método done ya que es como si estuviera permanentemente devolviendo true.
- CyclicBehaviour.- representa un comportamiento que debe ejecutarse una serie de veces.
 - El método done() devuelve false.
 - Se mantiene activo tanto tiempo como esté activo el agente.
 - Hay riesgo de que se pueda quedar con toda la CPU.

2.3.4.3.3 Comportamientos Compuestos:

Esta clase abstracta modela comportamientos a partir de la composición de otros comportamientos.

Está compuesta por diferentes subcomportamientos que se pueden ejecutar siguiendo diferentes políticas de planificación.

Las diferentes políticas vienen determinadas por la subclase elegida, que puede ser del tipo:

- FSMBehavior.- Esta clase permite definir una Máquina de Estados finita mediante subcomportamientos. Cada subcomportamiento representa un estado de la máquina y las transiciones se van produciendo según la salida de dichos estados.

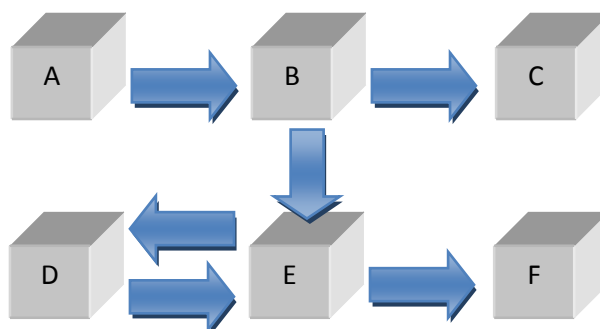


Ilustración 2.6 - Máquina de Estados Finita

- SequentialBehaviour.- Esta subclase de CompositeBehaviour ejecuta los subcomportamientos de manera secuencial y termina cuando todos ellos han terminado. Se utiliza cuando una tarea compleja se puede descomponer en una secuencia de pasos atómicos.

Para añadir los subcomportamientos se utiliza el método `addSubBehaviour()` y se ejecutarán en el orden en que sean agregados.

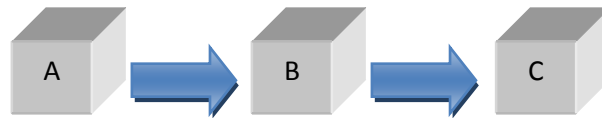


Ilustración 2.7 - Comportamiento Secuencial

- ParallelBehaviour.- Esta subclase de CompositeBehaviour ejecuta los subcomportamientos de manera concurrente y termina cuando se cumple una determinada condición sobre la terminación de los subcomportamientos.

Para añadir los subcomportamientos se utiliza el método `addSubBehaviour()` como en el comportamiento secuencial.

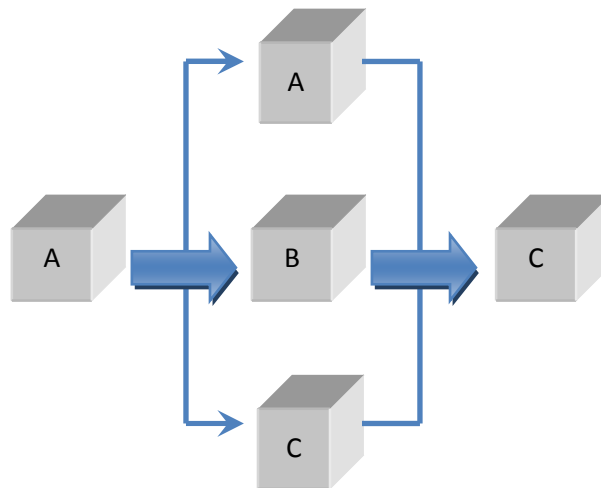


Ilustración 2.8 - Comportamiento Paralelo

2.3.4.3.4 Comportamientos temporales

Son comportamientos sencillos para ejecutar operaciones en determinados instantes de tiempo.

- Comportamiento TickerBehaviour.- Permite definir un comportamiento cíclico que ejecutará periódicamente una tarea.
- Comportamiento WakerBehaviour.- Este comportamiento implementa un comportamiento one-shot que se ejecuta una vez haya transcurrido un tiempo específico.

2.3.5 Comunicación entre Agentes

La comunicación entre agentes es fundamental para poder conseguir la potencia propia de los sistemas multiagente. Para que los agentes se puedan comunicar deben usar el mismo lenguaje de comunicación⁴⁰. El lenguaje de comunicación de agentes (ACL) permitirá transmitir una serie de conocimiento que vendrá expresado en un lenguaje de contenido⁴¹. Toda la comunicación está basada en el intercambio de mensajes. La Plataforma JADE que alberga al agente se encarga de hacerle llegar los mensajes a la plataforma del agente destinatario. La codificación decodificación de mensajes la hacen automáticamente los agentes.

Los mensajes que maneja JADE son FIPA-ACL y pueden contener los siguientes campos⁴²:

- **performative:** tipo de acto comunicativo (acción que realiza el mensaje). Es el único campo obligatorio y puede tomar uno de los siguientes valores:
 - accept-proposal: aceptar una propuesta recibida previamente
 - agree: estar de acuerdo en realizar alguna acción

⁴⁰ Curso "Tecnología de Agentes en Inteligencia Ambiental", Javier Carbó, Grupo de Inteligencia Artificial Aplicada, Campusde Colmenarejo, Univ. Carlos III de Madrid, 6 de Julio de 2007

⁴¹ <http://programacionjade.wikispaces.com/Comunicaci%C3%B3n#Fundamentos>

⁴² <http://programacionjade.wikispaces.com/Comunicaci%C3%B3n#Fundamentos>

- cancel: cancelar alguna acción pedida previamente
- cfp: solicitar propuestas para realizar una acción dada
- confirm: informar a un receptor que una proposición es cierta
- disconfirm: informar a un receptor que una proposición es falsa
- failure: informar a otro agente que se intentó una acción pero falló
- inform: informar a un receptor que una proposición es cierta
- inform-if: si el agente que recibe la acción cree que la sentencia es verdadera informará de manera afirmativa, sino indicará que es falsa.
- inform-ref: permite que el emisor informe al receptor de un objeto que cree que corresponde a un descriptor, como puede ser un nombre u otra descripción que lo identifique.
- not-understood: informar a un receptor que el emisor no entendió el mensaje
- propagate: el receptor trata el mensaje como si fuese dirigido directamente a él, y debe identificar los agentes en este descriptor y enviarles el mensaje a ellos
- propose: enviar una propuesta para realizar una cierta acción
- proxy: el receptor debe seleccionar agentes objetivo denotados por una descripción dada, y enviarles un mensaje embebido
- query-if: preguntarle a otro agente si una determinada proposición es cierta
- query-ref: preguntar a otro agente por el objeto referenciado en una expresión
- refuse: rechazar realizar una acción
- reject-proposal: rechazar una propuesta durante una negociación
- request: solicitar a un receptor que realice alguna acción
- request-when: solicitar al receptor que realice alguna acción cuando una proposición dada sea cierta
- request-whenever: solicitar al receptor que realice alguna acción cada vez que una proposición dada sea cierta
- subscribe: una intención persistente de notificar al emisor de un determinado valor, y volver a notificarle cada vez que dicho valor cambie

- **sender:** AID⁴³ del emisor
- **receiver:** lista de AID's de los receptores
- **reply-to:** receptor de un mensaje reply

2.3.6 LEAP

Los agentes escritos en JADE pueden ejecutarse en entornos móviles como teléfonos o PDAs integrando las redes inalámbricas junto con las redes convencionales. El problema es que JADE no puede correr en pequeños dispositivos debido a diversas razones:

- **Espacio:** La memoria necesaria para el entorno de ejecución es de varios megas.
- **Version del JDK:** JADE necesita el JDK 1.4 o posterior, mientras que la mayoría de los dispositivos solo soportan PJava (Personal Java).
- **Características propias de las redes inalámbricas:** Como son IP dinámicas, conectividad intermitente o bajo ancho de banda, hace que JADE no sea lo apropiado⁴⁴.

Para ello surge LEAP (Lightweight Extensible Agent Platform) que permite ejecutar agentes JADE en dispositivos móviles y/o conectados a través de redes inalámbricas.

Los container del entorno de ejecución de JADE-LEAP se dividen en dos partes, un FrontEnd, que se ejecuta en el dispositivo móvil, y un backend, que se ejecuta en un servidor de la red fija

2.3.6.1 Limitaciones de LEAP

- No se puede utilizar el SNIFFER
- No se pueden utilizar clases de Java que no estén incluidas en la versión micro (J2ME).

⁴³ AID.-Agent Identifier

⁴⁴ (Carbó, 2007)

- Hay que contar con los fuentes de cualquier otra clase java a incluir, ya que hay que recompilar directamente todo el código fuente a la vez para generar el .jad

2.4 Protocolos de transporte

En ciertos aspectos, los protocolos de transporte se parecen a los protocolos de red. Ambos se encargan del control de errores, la secuenciación y el control del flujo. Pero también existen diferencias importantes entre ambas, como los entornos en que operan, la capa transporte necesita el direccionamiento explícito de los destinos, mientras que la capa de red no, otra diferencia es la cantidad de datos, mucho mayor en la capa de transporte que en la de enlace de datos.

Básicamente se utilizarán dos protocolos de transporte TCP y UDP.

2.5 UDP

Sus siglas significan user datagram protocol, es un protocolo no orientado a la conexión, no proporciona ningún tipo de control de errores ni de flujo, aunque utiliza detección de errores, esto quiere decir que en caso de detectar un error, el UDP no entrega el datagrama a la aplicación si no la descarta. UDP es un protocolo simple haciendo que sea ideal para aplicaciones en tiempo real.

2.5.1 CARACTERÍSTICAS

- *No garantiza la fiabilidad.*- no hay seguridad que cada datagrama enviado llegue a su destino.
- *Protocolo best effort.*- hace todo lo posible por transmitir los datagramas pero no garantiza su entrega.
- *No preserva la secuencia de la información.*- la información que recibe el receptor puede llegarle de forma desordena, es decir existe la posibilidad de que

no lleguen todos los datagramas o lleguen con retardo y de manera desordenada.

UDP utiliza el protocolo IP⁴⁵, los datos enviados por la aplicación se les añaden la cabecera UDP y se encapsulan en un datagrama IP. Cada operación de salida de un datagrama UDP provoca la generación de un datagrama IP.

En la siguiente figura se muestra el protocolo UDP y el encapsulamiento en un datagrama IP:

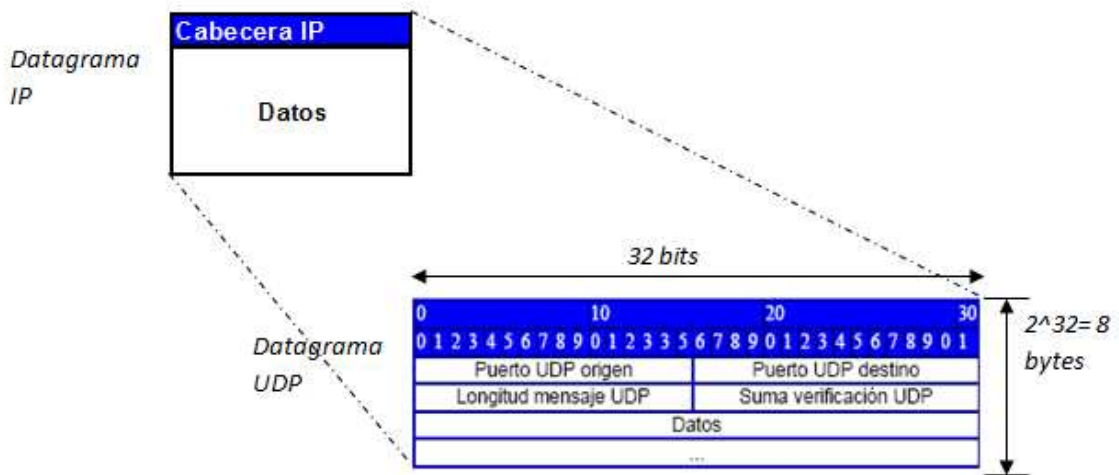


Ilustración 2.9 - Encapsulamiento IP⁴⁶

2.5.2 Cabecera UDP

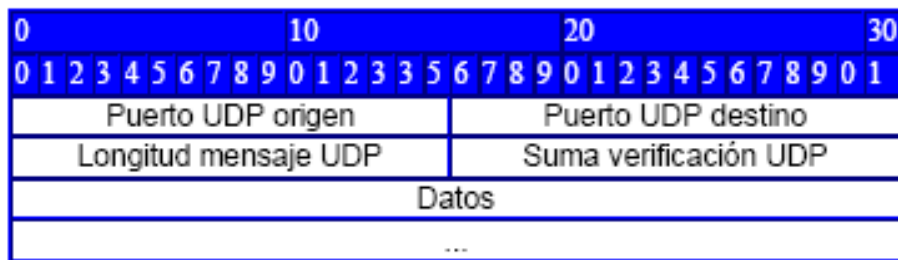


Ilustración 2.10 - Cabecera UDP

⁴⁵ Protocolo de Internet no orientado a la conexión

⁴⁶ (Barceló, y otros, 2008, pág. Capítulo 1)

El datagrama UDP consta de una cabecera y un cuerpo para encapsular los datos. La cabecera consta de lo siguiente:

- **Puerto de origen.-** 16 bits, número de puerto de la máquina de origen
- **Puerto destino.-** 16 bits, número de puerto de la máquina destino.

El campo puerto por lo general está en un rango de valores válidos va de 0 a 65.535.

- El puerto 0 está reservado, pero es un valor permitido como puerto origen si el proceso emisor no espera recibir mensajes como respuesta.
 - Los puertos 1 a 1023 se llaman puertos "bien conocidos".
 - Los puertos 1024 a 49.151 son puertos registrados.
 - Los puertos 49.152 a 65.535 son puertos momentáneos y son utilizados como puertos temporales, sobre todo por los clientes al comunicarse con los servidores.
- **Longitud del mensaje UDP.-** 16 bits. Especifica la longitud de medida en bytes del mensaje UDP incluyendo la cabecera. Longitud mínima de 8 bytes.
 - **Suma de verificación UDP o checksum.-** 16 bits - opcional. Suma de comprobación de errores del mensaje, si no se usa se poner el valor 0.

El checksum se obtiene a partir del datagrama completo (datos + cabecera) sumando palabras de 16 bits. Si el tamaño del datagrama no es múltiplo de 16, se calcula añadiendo un byte 0. Se incluye en el checksum la pseudocabecera (IP) con el fin de asegurarse que la entrega se hace en el sitio correcto.

Dirección IP origen	
Dirección IP destino	
0	Protocolo
Tamaño de Datagrama	
Puerto origen	Puerto destino
Tamaño	Checksum
Datos	

Ilustración 2.11 - Pseudocabecera

En el cuerpo encontramos los **Datos** aquí viajan lo que envían las aplicaciones.

2.5.3 Aplicaciones sobre UDP

- **DNS:** Domain Name System.
- **Enrutamiento dinámico:** RIP⁴⁷, OSPF⁴⁸, IGMP⁴⁹.
- **SNMP:** Simple Network Management Protocol.
- **RTP:** Real Time Protocol.
- **DHCP:** Dynamic Host Configuration Protocol.

2.6 TCP

Sus siglas significan Transmission Control Protocol, este protocolo proporciona fiabilidad a la aplicación, es decir garantiza la entrega de la información en el mismo orden que ha sido transmitida por la aplicación origen, teniendo un control de flujo y de errores y basándose en los siguientes principios:

⁴⁷ Protocolo de encaminamiento de información

⁴⁸ Open Shortest Path First

⁴⁹ Protocolo de administración de grupos de Internet

- Transmisión libre de errores.- debe entregar a la aplicación de destino la misma información que entregó la aplicación de origen.
- Garantía de entrega de la información.- garantiza que toda la información llegue completa al destino.
- Garantía de mantenimiento de la secuencia de transmisión.- garantiza la entrega del flujo de información en el mismo orden en que le fue entregado por la aplicación origen.
- Eliminación de duplicados.- garantiza que sólo entregará una copia de la información transmitida a la aplicación de destino, en caso de que exista copias por problemas en la red TCP las eliminará.

2.6.1 Propiedades

- Define flujos de transmisión (stream oriented): la aplicación organiza los datos de información en flujos de bits estructurados en bytes.
- Orientado a la conexión: posee una fase de establecimiento de conexión, una de transmisión de datos y una de desconexión.

El flujo de datos es tratado como una secuencia de bytes (stream), siendo responsabilidad del protocolo la decisión de cómo dividir (o agrupar) las unidades de datos de la aplicación a la hora de transferirlos.

2.6.2 Fases de TCP

Para garantizar que no se pierde ningún bloque de datos, una sesión TCP consta de tres fases:

- *Establecimiento de conexión:* Crea un marco para el intercambio fiable de información. Durante esta fase los hosts origen y destino determinan parámetros necesarios como son el número inicial de secuencia y los tamaños de buffer necesarios.

- *Transferencia de datos:* En esta fase entran en juego los mecanismos de detección y corrección de errores, control de flujo y control de la congestión.
- *Cierre de conexión:* A propuesta de uno de los extremos, se intenta una desconexión negociada, donde no queden datos sin entregar por ninguna de las dos partes.

2.6.3 Encapsulado de TCP

Al igual que en el caso de UDP e ICMP⁵⁰, los segmentos TCP viajan dentro de tramas IP, como se indica en la siguiente figura:

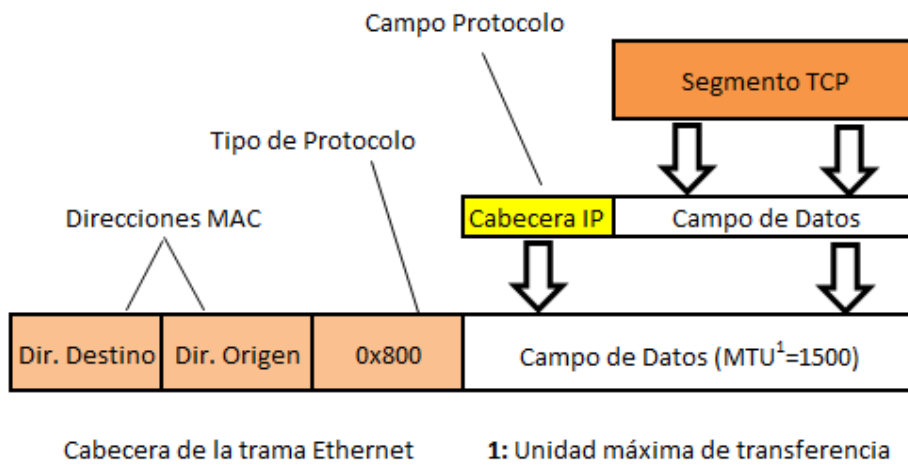


Ilustración 2.12- Encapsulado de un segmento TCP en un datagrama⁵¹

⁵⁰ Protocolo de Mensajes de Control de Internet

⁵¹ (Redes de computadores, 2010)

2.6.4 Formato de Segmentos TCP

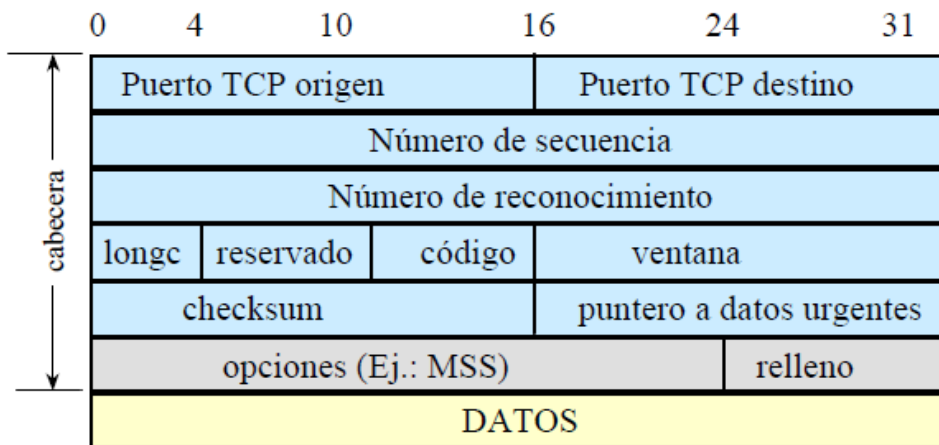


Ilustración 2.13 - Formato de un segmento TCP⁵²

- **Puerto TCP origen - Puerto TCP destino:** identifican, junto a las direcciones origen y destino del paquete IP, la conexión a la que pertenece el segmento.
- **Número de secuencia:** indica el orden, dentro de la transmisión, del primer byte de datos contenido en el segmento. Junto al tamaño total del paquete IP, que permite conocer cuántos bytes de datos se reciben.
- **Número de reconocimiento:** indica cuál es el byte que el emisor del segmento espera recibir como número de secuencia.
- **Longc o longitud de la cabecera:** indica cuántas palabras de 32 bits (4 bytes) componen la cabecera TCP.
- **Código:** consta de seis bits, teniendo cada uno de ellos un significado independiente del resto.
- **Ventana:** indica cuál es el tamaño de la ventana de recepción del emisor del segmento, es decir, el espacio en memoria disponible en el receptor.
- **Checksum:** contiene un código detector de errores similar al UDP.
- Pueden existir una o varias opciones, que permiten gestionar aspectos como el tamaño máximo de segmento, los reconocimientos selectivos.

⁵² (Redes de computadores, 2010)

2.7 Diferencia entre TCP y UDP

En UDP cada vez que se envía un datagrama, se debe enviar el descriptor de la aplicación local y la dirección de la aplicación que va a recibir el datagrama, siendo así los mensajes más grandes que los de TCP.

TCP es un protocolo orientado a la conexión es por esto que antes de enviar la información se debe establecer la conexión entre las dos aplicaciones, lo que da como resultado que la conexión en TCP sea más demorada que en UDP.

En UDP existe un límite de tamaño de los datagramas, mientras que en TCP no tiene límite.

UDP es un protocolo desordenado, no garantiza que los datagramas que se hayan enviado sean recibidos en el mismo orden en el que fueron enviados, al contrario TCP es un protocolo ordenado, garantiza que todos los paquetes que se envían serán recibidos en el mismo orden.

Se puede ver que el protocolo TCP es mucho más seguro que UDP, pero este último protocolo es más utilizado para streaming ya que en esta técnica lo más importante es enviar datos de manera rápida más que la verificación de los datos.

2.8 Protocolos de comunicación

Se presentan los protocolos de establecimiento y gestión de sesión por parte del cliente (RTSP), los relativos al transporte y control (RTP/RTCP) con las principales funciones y mensajes disponibles.

2.9 PROTOCOLO RTSP

El protocolo de flujo de datos en tiempo real (Real Time Streaming Protocol) establece y controla uno o muchos flujos sincronizados de datos, ya sean de audio o de video.

RTSP es un protocolo no orientado a conexión, en lugar de esto el servidor mantiene una sesión asociada a un identificador, en la mayoría de los casos RTSP usa TCP para datos de control del reproductor y UDP para los datos de audio y vídeo.

La conexión debe permanecer activa durante todo el proceso de comunicación, es decir, desde el momento en el que el cliente lanza la primera petición, hasta que termina de recibir los datos correspondientes al vídeo o audio y utiliza el propio protocolo RTSP para cerrar la sesión.

RTSP guarda ciertas similitudes con protocolos como HTTP al ser ambos protocolos destinados a solicitar información de un elemento de la comunicación a otro; a diferencia de HTTP, RTSP es un protocolo que necesita mantener el estado de las conexiones, además la información se transporta en un protocolo diferente. El protocolo de transporte utilizado para transferir RTP es siempre TCP, además tiene el puerto 554 por defecto, aunque puede ser modificado tanto en el cliente como en el servidor.

En RTSP existe sesiones mantenidas por el servidor, cada una de estas posee su identificador, durante estas sesiones se pueden abrir y cerrar tantas sesiones como sea necesario, también se puede utilizar UDP como protocolo de transporte.

2.9.1 Características

- Dispone de un identificador de protocolo propio `rtsp://`.
- Tanto el cliente como el servidor pueden realizar peticiones.

- Los datos son transportados por el protocolo RTP.

2.9.2 Similitudes con HTTP⁵³

- Formato de las peticiones/respuestas.
- Formato de las URL.
- Sintaxis.
- Negociación de los contenidos.

2.9.3 Diferencias con HTTP⁵⁴

- RTSP es un protocolo con estado a diferencia de HTTP.
- Tanto servidores como clientes RTSP pueden realizar peticiones.
- Los datos son transportados mediante un protocolo diferente.
- Cada flujo transportado puede residir en un servidor distinto.

2.9.4 Componentes

- RTP.- Real Transport Protocol.
- RTCP.-Real Time Control Protocol

2.9.5 Modos de Operación

RTSP utiliza los siguientes modos de operación para enviar audio y video:

- Unicast.- se envía un flujo de información a cada usuario.

⁵³ (Alonso, 2009-2010)

⁵⁴ (Alonso, 2009-2010)

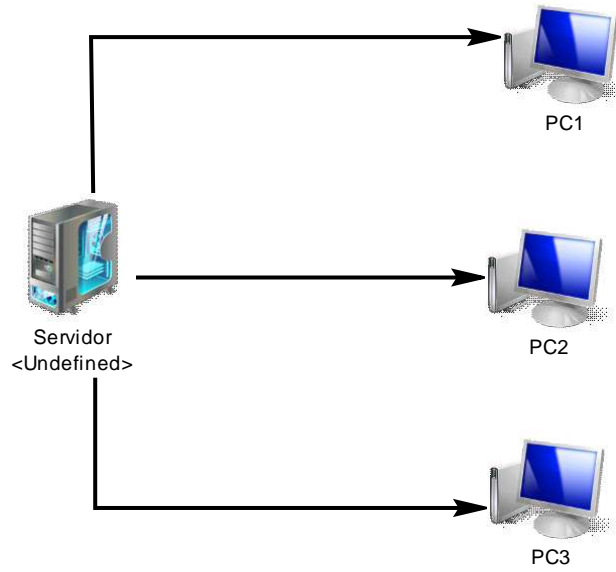


Ilustración 2.14 – Unicast

- Multicast.- se envía un flujo único de información.

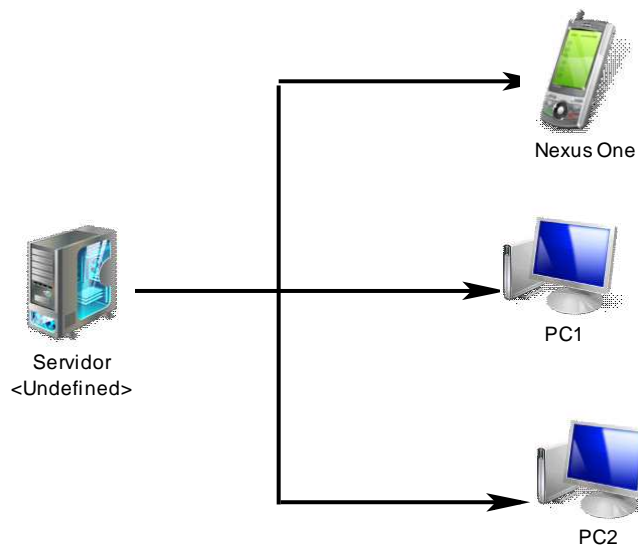


Ilustración 2.15 – Multicast

2.9.6 Métodos RTSP

Están basados en peticiones HTTP y generalmente son enviados del cliente al servidor.

A continuación se va a describir las más usadas:

- **OPTIONS:** el cliente solicita la información sobre las opciones de comunicación con el servidor.
- **DESCRIBE:** solicita la configuración del contenido multimedia.
- **ANNOUNCE:** cuando el servidor lo envía se actualiza la información de la sesión en tiempo real.
- **SETUP:** reserva un recurso en el servidor, indicando la url del flujo multimedia e inicia una sesión RTSP.
- **PLAY:** posterior al SETUP, provoca que el servidor comience a enviar el flujo de datos.
- **PAUSE:** detiene de forma temporal el flujo, permitiendo que sea reiniciado desde el mismo punto mediante otra petición PLAY.
- **TEARDOWN:** detiene definitivamente la entrega de datos de la url indicada, liberando los recursos asociados.
- **GET_PARAMETER:** solicita un valor de la descripción de la sesión.
- **SET_PARAMETER:** envía un valor de la descripción de la sesión.
- **REDIRECT:** indica que el cliente debe conectarse a otro servidor.
- **RECORD:** grabación de datos.

2.9.7 Formato URL

El URL del protocolo RTSP tiene la forma siguiente:

rtsp://host:puerto/camino/al/recurso

Ejemplo:

rtsp://servidor.com:1234/video.3gp

2.9.8 Formato del Mensaje de Petición

El mensaje de petición consta de tres partes:

- Línea de Petición.
- Cabecera de Petición.
- Cuerpo del Mensaje

2.9.8.1 Línea de Petición

Identifica al recurso al que queremos acceder y la versión de RTSP que estamos utilizando, quedando el formato de la URI de petición de la siguiente manera:

| **OrdenRTSP** | |*URI-petición*/ |Versión-RTSP| [Salto de Línea]

2.9.8.2 Cabecera de Petición

Consta de tres partes la cabecera general representa la información general del mensaje, la cabecera de petición que contiene información solo cuando el mensaje es de petición y la cabecera de entidad que contiene información sobre el cuerpo del mensaje.

El formato de la cabecera de petición es el siguiente:

Nombre-Campo: valor-campo [Salto de Línea]

2.9.8.3 Cuerpo del Mensaje

Lleva información asociada a la petición.

2.9.9 Formato del Mensaje de Respuesta

El mensaje de respuesta consta de tres partes:

- Línea de Estado de la respuesta.
- Cabecera de Respuesta.
- Cuerpo del Mensaje

2.9.9.1 Línea de Estado

En la línea de estado se diferencia la versión de RTSP, el código de estado el cual nos indica si hubo éxito o fracaso en la petición y la descripción del estado.

Teniendo la siguiente forma la línea de estado:

Versión-RTSP Código-Estado Descripción-Estado [Salto de Línea]

Siempre que enviemos una petición, el código del estado estará representado con lenguaje HTML, en la siguiente tabla podemos observar los diferentes códigos:

Tabla 2.2 - Códigos de Respuesta RTP

Código	Características
1XX	Informativo: Petición recibida, procesándola.
2XX	Éxito: Petición recibida, procesada y aceptada.
3XX	Redirección: La siguiente acción debe ser ejecutada para completar la petición.
4XX	Error en el cliente: Sintáxis errónea o imposible de procesar.
5XX	Error en el servidor: Error al ejecutar una petición aparentemente correcta

2.9.9.2 Cuerpo del Mensaje

Lleva información asociada a la respuesta.

2.9.10 Negociación RTSP

Al empezar la negociación con el comando OPTIONS, el cliente enviará la siguiente petición:

```
*****Petición Cliente OPTIONS*****
```

```
OPTIONS rtsp://192.168.3.13:1234/stream.sdp RTSP/1.0
CSeq: 0
User-Agent: PVPlayer CORE/6.506.4.1 OpenCORE/2.02 (Linux;Android 2.2)
ClientChallenge: 9e26d33f2984236010ef6253fb1887f7
PlayerStarttime: [28/03/2003:22:50:23 00:00]
CompanyID: KnKV4M4I/B2FjJ1TToLycw==
GUID: 00000000-0000-0000-0000-000000000000
*****Respuesta Servidor OPTIONS*****
RTSP/1.0 200 OK
Server: vlc 1.0.5
Content-Length: 0
Cseq: 0
Public: DESCRIBE,SETUP,TEARDOWN,PLAY,PAUSE,GET_PARAMETER
```

El ejemplo anterior es la respuesta del servidor, podemos observar que los mensajes llevan cabeceras que indican el número de secuencia del mensaje identificados con el campo **Cseq** así como también se puede observar que en este mensaje nos retorna los comandos que soporta el reproductor VLC.

Seguimos con el comando DESCRIBE, teniendo la siguiente petición del cliente:

```
*****Petición Cliente DESCRIBE*****
DESCRIBE rtsp://192.168.3.13:1234/stream.sdp RTSP/1.0
CSeq: 1
User-Agent: PVPlayer CORE/6.506.4.1 OpenCORE/2.02 (Linux;Android 2.2)
Accept: application/sdp
Accept-Encoding:
*****Respuesta Servidor DESCRIBE*****
RTSP/1.0 200 OK
Server: vlc 1.0.5
```

Date: Mon, 06 Sep 2010 17:34:17 GMT

Content-Type: application/sdp

Content-Base: rtsp://192.168.3.13:1234/stream.sdp

Content-Length: 653

Cache-Control: no-cache

Cseq: 1

v=0

o=- 15001389861663027654 15001389861663027654 IN IP4 DARIO-PC

s=Unnamed

i=N/A

c=IN IP4 0.0.0.0

t=0 0

a=tool:vlc 1.0.5

a=recvonly

a=type:broadcast

a=charset:UTF-8

a=control:rtsp://192.168.3.13:1234/stream.sdp

m=audio 0 RTP/AVP 96

b=AS:64

a=rtpmap:96 MP4A-LATM/44100/2

a=fmtp:96 profile-level-id=15; object=2; cpresent=0; config=400024203fc0

a=control:rtsp://192.168.3.13:1234/stream.sdp/trackID=0

m=video 0 RTP/AVP 97

b=AS:400

a=rtpmap:97 H264/90000

a=fmtp:97 packetization-mode=1;profile-level-id=4d400c;sprop-parameter-sets=Z01ADJJWDwU3/gE2AMAgAAADACAAAABZB4oVc,aM8yyA==;

a=control:rtsp://192.168.3.13:1234/stream.sdp/trackID=1

A diferencia del anterior este nos devuelve el tipo de aplicación que soporta, el tipo de codificación la configuración del reproductor y las diferentes pistas (track).

Luego de esto el cliente envía el comando SETUP, donde le dice al servidor los mecanismos de transporte que quiere utilizar y el orden de preferencia, además indica que va utilizar el protocolo UDP y que los puertos 5067 y 5068 van a ser para la transferencia de los paquetes RTP y los mensajes de control (RTCP) respectivamente.

```
*****Petición Cliente*****>>>>Setup - Track 0>>>>>
```

```
SETUP rtsp://192.168.3.13:1234/stream.sdp/trackID=0 RTSP/1.0
```

```
CSeq: 2
```

```
User-Agent: PVPlayer CORE/6.506.4.1 OpenCORE/2.02 (Linux;Android 2.2)
```

```
Transport: RTP/AVP/UDP;unicast;client_port=57701-57702
```

```
Blocksize: 1400
```

El servidor entonces nos devuelve la siguiente respuesta:

```
*****Respuesta Servidor*****>>>>Setup - Track 0>>>>>
```

```
RTSP/1.0 200 OK
```

```
Server: vlc 1.0.5
```

```
Date: Mon, 06 Sep 2010 17:34:17 GMT
```

```
Transport: RTP/AVP/UDP;unicast;client_port=57701-57702;server_port=57705-57706;ssrc=FD50A3C;mode=play
```

```
Session: 4b1e2590bf10f7e2
```

```
Content-Length: 0
```

```
Cache-Control: no-cache
```

```
Cseq: 2
```

En donde se puede observar que confirma los puertos del cliente añadiendo los puertos de transferencia y el código de la sesión el cual va ser el mismo durante toda la sesión.

Como tenemos otra pista se enviará otro SETUP, el mismo tendrá los mismos parámetros del anterior.

```
*****Petición Cliente*****>>>>Setup - Track 1>>>>>
```

```
SETUP rtsp://192.168.3.13:1234/stream.sdp/trackID=1 RTSP/1.0
```

```
CSeq: 3
```

```
Session: 4b1e2590bf10f7e2
```

```
User-Agent: PVPlayer CORE/6.506.4.1 OpenCORE/2.02 (Linux;Android 2.2)
```

```
Transport: RTP/AVP/UDP;unicast;client_port=57703-57704
```

```
Blocksize: 1400
```

```
*****Respuesta Servidor*****>>>>Setup - Track 1>>>>>
```

```
RTSP/1.0 200 OK
```

```
Server: vlc 1.0.5
```

```
Date: Mon, 06 Sep 2010 17:34:18 GMT
```

```
Transport: RTP/AVP/UDP;unicast;client_port=57703-57704;server_port=57706-
```

```
57707;ssrc=CE2B11AE;mode=play
```

```
Session: 4b1e2590bf10f7e2
```

```
Content-Length: 0
```

```
Cache-Control: no-cache
```

```
Cseq: 3
```

Después de establecer la conexión, el cliente está listo y envía el comando PLAY, el mismo contiene el URL del contenido y el identificador de la sesión.

```
*****Petición Cliente PLAY*****
```

```
PLAY rtsp://192.168.3.13:1234/stream.sdp RTSP/1.0
```

```
CSeq: 4
```

```
Session: 4b1e2590bf10f7e2
```

```
User-Agent: PVPlayer CORE/6.506.4.1 OpenCORE/2.02 (Linux;Android 2.2)
```

Lo que nos devuelve es el rango de la transmisión.

*****Respuesta Servidor PLAY*****

RTSP/1.0 200 OK

Server: vlc 1.0.5

Date: Mon, 06 Sep 2010 17:34:19 GMT

RTP-Info: url=rtsp://192.168.3.13:1234/stream.sdp/trackID=0;seq=29753;rtptime=22

6825855, url=rtsp://192.168.3.13:1234/stream.sdp/trackID=1;seq=2049;rtptime=4628

66607

Session: 4b1e2590bf10f7e2

Content-Length: 0

Cache-Control: no-cache

Cseq: 4

Si el cliente quiere terminar con la transmisión, lo puede hacer con el comando TEARDOWN, el servidor envía que ha recibido con éxito la petición y para la transmisión.

Lo que se ha explicado anteriormente se puede observar en la siguiente figura:

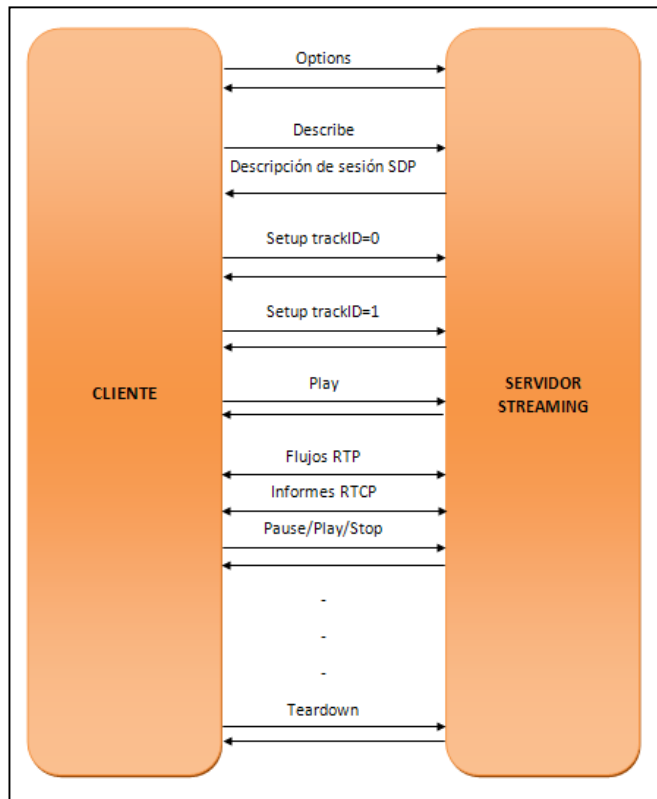


Ilustración 2.16 - Negociación RTSP

2.10 PROTOCOLO RTP

Real Time Transport Protocol es un protocolo que se utiliza para enviar cualquier tipo de formato de audio y video, funciona sobre el protocolo UDP, haciendo que el protocolo RTP no garantice que los paquetes lleguen a su destino y con la calidad de servicio ni en el orden de envío.

RTP además de los datos en sí, transporta cierta información útil para realizar un cierto control de la comunicación, pero no informa sobre el estado de la misma, es por esto que trabaja junto RTCP⁵⁵, el cual genera los mensajes de control de las transferencias que incluye información sobre los paquetes perdidos y será posible conocer también el estado del buffer el cliente.

⁵⁵ Real Time Control Protocol

2.10.1 Funcionamiento

El funcionamiento de RTP se basa en UDP, el emisor encapsula los datos en un paquete RTP, y este a su vez es encapsulado en un datagrama UDP y pasan los datos a un reproductor, el mismo los decodifica y los reproduce.

Cada flujo RTP tiene una fuente para el cliente, llamado SSRC o Synchronization Source, la cual es única y viene identificada en el paquete. Esta fuente suele ser la que provee el flujo final al cliente, es decir, la última que realiza alguna modificación sobre los contenidos RTP, generalmente se trata del servidor. Además de SSRC se pueden encontrar CSRC o Contributing Sources y son cada uno de los elementos que inicialmente generaron un flujo RTP y fue sometido a un proceso a algún proceso intermedio antes de llegar al cliente. También son indicadores únicos que, al no estar presentes en todos los casos, forman parte opcionalmente en los mensajes RTP.

2.10.2 Cabecera RTP

En la siguiente figura se muestra los campos posibles en una cabecera RTP⁵⁶:

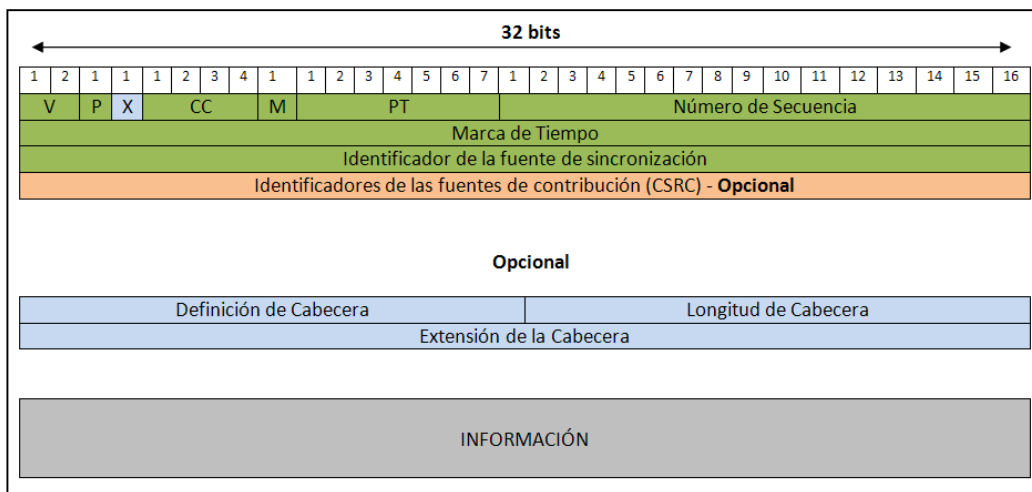


Ilustración 2.17 - Cabecera RTP⁵⁷

⁵⁶ (Catalán, 2009)

⁵⁷ (Catalán, 2009)

- **Versión (V):** 2bits. Identifica la versión de RTP utilizada, la versión actual es la dos.
- **Relleno (P):** 1 bit. Si este bit está activado, indica que el paquete contiene uno o más octetos de relleno adicional de los datos, es decir el paquete RTP debería tener un múltiplo de 32 bits, en caso de no llegar a este número con los datos transportados se añadirán bits de relleno.
- **Extensión (X):** 1 bit. Indica la existencia de una cabecera de extensión, es decir, al menos un bloque de 32 bits. Estos 32 bits se componen de 16 bits de definición de la cabecera y 16 bits indicando el número de palabras de 32 bits que siguen a continuación.
- **Número de CSRC⁵⁸ (CC):** Contador de CSRC que indica el número de identificadores que existe en la cabecera.
- **Marcador (M):** 1bit. Bit de marcado utilizado por aplicaciones específicas.
- **Tipo de Contenido (PT):** 7 bits. Tipo de datos contenidos en el paquete. Estos 7 bits presentan 128 opciones para indicar la codificación utilizada en los datos.
- **Número de secuencia:** 16 bits. Iniciado en un valor aleatorio y se incrementa de uno en uno cada vez que se envía un paquete RTP, permite al receptor conocer los paquetes fuera de secuencia y realizar análisis de pérdidas.
- **Marca de tiempo:** 32 bits. Indica el tiempo de muestreo del primer octeto contenido en los datos, puede ser utilizado por el cliente para evaluar el jitter y mantener la sincronización de flujos. Incrementa de manera lineal.
- **SSRC⁵⁹:** 32 bits. Indica la fuente de sincronización para el cliente. Suele ser única y fija durante la transmisión. El valor utilizado también responde, inicialmente, a un número aleatorio.

⁵⁸ Contributing Source.- son cada uno de los elementos que inicialmente generaron un flujo RTP.

⁵⁹ Synchronization Source

- **Lista de CSRC:** de 0 a 15 elementos, 32 bits por elemento. Identifica las fuentes que contribuyen a los datos contenidos en el paquete RTP.

2.11 PROTOCOLO RTCP

El RTCP real time protocol control se basa en la transmisión periódica de paquetes de control durante una sesión, utilizando el mismo mecanismo de distribución de los paquetes RTP.

La función principal de RTCP es informar de la calidad de servicio proporcionada por RTP. Este protocolo recoge estadísticas de la conexión e información adicional como: bytes enviados, paquetes enviados, paquetes perdidos o jitter entre otros.

Una aplicación puede usar esta información para incrementar la calidad de servicio, ya sea limitando el flujo o usando un códec de compresión más baja. De modo general RTCP se usa para informar de la QoS⁶⁰.

2.11.1 Funciones

1. Da información sobre los datos distribuidos.
2. Mantiene un identificador persistente en el transporte de una fuente RTP, que se denomina *nombre canónico o CNAME*. El identificador SSRC puede cambiar por alguna interrupción entonces con el CNAME se puede recuperar los participantes de la sesión.
3. Ayuda a calcular la tasa a la cual se envía los paquetes.
4. Comunica información de control de sesión, esta es opcional

2.11.2 Paquetes RTCP

Los paquetes RTCP empiezan con una parte fija, seguida de una serie de elementos estructurados que pueden variar de acuerdo al tipo de paquete pero siempre deben tener

⁶⁰ Quality of Service, calidad de servicio.

una longitud de múltiplos de 32 bits, además se incluyen campos para indicar la alineación y un campo de longitud para que puedan ser apilables.

Pueden transportar diferentes tipos de información de control:

- SR informe del emisor para transmitir y recibir estadísticas de los participantes que son emisores activos.
- RR informe del receptor para recibir estadísticas de los participantes que no son emisores activos.

Estos informes el SR Y RR son muy similares entre sí, Sus principales diferencias respecto a la cabecera de un paquete RTP son: La aparición del campo RC que supone la eliminación del bit de extensión, bit de marcado y CC. Se utiliza para indicar el número de bloques de informe que contiene el paquete. La aparición de un campo de longitud en lugar del número de secuencia y la desaparición de la marca temporal.

- SDES elementos de descripción de la fuente, incluye el CNAME.
- BYE: indica que se ha dejado de participar.
- APP: funcionalidades específicas de la aplicación.

2.12 Streaming

Streaming es una técnica que consiste en objetos multimedia que se reproducen y descargan simultáneamente desde el Internet sin la necesidad de almacenar toda la información en la memoria del cliente.

La ventaja de esta técnica es que no requiere que el vídeo se almacene en el teléfono móvil dejando espacio para albergar memoria auxiliar y programas para tratar el vídeo de manera

eficiente. Por otro lado, gracias a que no se consume esa cantidad de memoria, se pueden desplegar programas eficientes y poderosos en los terminales móviles.

2.12.1 Proceso Streaming

El proceso se inicia desde el cliente, conectándose con el servidor que comienza a mandar el fichero. Para el envío se utiliza un códec que comprime el archivo que viaja sobre un protocolo ligero como puede ser RTP. En el cliente se constituye un buffer que almacena la información que va llegando. Una vez que el buffer contiene una parte del archivo comienza la reproducción del mismo, al mismo tiempo que continúa con la descarga. Cabe destacar la importancia de la sincronización en este tipo de sistemas, ya que los programas no reproducirán el archivo si este no está, y, si la conexión varía su retardo, se podrá reproducir el contenido del buffer, según el protocolo. Si se sufren demasiados retardos, el buffer se vacía y la ejecución del archivo también se cortaría hasta que se normalice.

En la telefonía móvil, streaming está sometido al comportamiento de los canales, existiendo problemas externos como interferencias provocando desconexiones en la transmisión que muchas veces son molestas para el usuario, y hacen que la técnica streaming pierda efectividad. Es por esto que para recibir los datos de streaming se debe tener una aplicación dedicada llamado reproductor o media player, que debe ofrecer las siguientes características:

- Descompresión.- el reproductor debe ser capaz de descomprimir los datos de audio y video a medida que vayan llegando.
- Eliminación de jitter.- el jitter es la variabilidad del retraso que existe entre el origen y el destino dentro de un mismo stream, el reproductor debe ser capaz de eliminar el jitter antes de enviar los paquetes de stream.
- Corrección de errores para poder evitar la pérdida de paquetes.

2.13 Contenedor de video 3gp

Actualmente este formato es ampliamente utilizado para disminuir el almacenamiento y los requisitos de ancho de banda para que pueda ser utilizado con teléfonos móviles.

Formato contenedor de multimedia definido por Third Generation Partnership Project(3GPP)⁶¹ para ser usado en teléfonos celulares de tercera generación. Los archivos en este formato suelen tener la extensión ".3gp" o ".3g2".

3GP es una versión simplificada del formato MPEG-4⁶², MP4⁶³, diseñado para disminuir tanto los requerimientos de espacio como de ancho de banda para estos archivos por la menor capacidad de los celulares.

Su estructura es muy similar a la de MP4, de hecho, se puede considerar 3GP como una versión simplificada de MP4. La diferencia fundamental entre ellos son las codificaciones de audio soportadas y las resoluciones a las que pueden trabajar. En 3GP es habitual encontrar codificaciones como MPEG-4 y H.263 en cuanto a vídeo y AMR-NB o AAC-LC en cuanto a audio.

Se distingue dos normas diferentes: el 3GPP, que se utiliza en los teléfonos GSM⁶⁴ y que tienen la extensión de archivo .3gp. Y el 3GPP2, que es para los teléfonos basados en CDMA⁶⁵, y tienen la extensión de archivo .3gp2.

Este formato se puede reproducir desde los siguientes reproductores:

- VLC Media player

⁶¹ Fundado con la intención de definir especificaciones técnicas comunes a nivel global durante el desarrollo de la tercera generación de telefonía móvil. www.3gpp.org

⁶² Nombre de un grupo de estándares de codificación de audio y video así como su tecnología relacionada normalizada por el grupo MPEG (Moving Picture Experts Group).

⁶³ Una abreviatura de la especificación MPEG-4 Part 14, formato de almacenamiento multimedia, puede ser grabar diversos tipos de datos, entre ellos audio digital, video digital, imágenes, subtítulos.

⁶⁴ Sistema Global para las Comunicaciones Móviles

⁶⁵ Acceso múltiple por división de código.

- Tótem
- Media Player Classic
- The KMPlayer
- QuickTime
- RealPlayer
- GOM Player

2.14 Códec

El término *códec*, es una abreviatura de Compresor-Descompresor. Un códec puede servir en distintos estados de la información y con diferentes funcionalidades. Puede utilizarse en la transmisión de un archivo o en el almacenaje, para comprimir propiamente, para cifrar o para modificar su formato.

Existen códecs que provocan pérdidas de información para conseguir un tamaño lo más pequeño posible del archivo destino, así como también existen codecs sin pérdidas (lossless). Pero los más utilizados son los primeros, ya que no justifica un aumento de calidad casi imperceptible para el ser humano, por un notable aumento del tamaño de los datos. Los códecs que no pierden información en el proceso de compresión, suelen emplearse cuando la calidad del archivo multimedia enviado por la web, se requiere para un tratamiento posterior de dicho archivo.

A principios de los 90's, el grupo de expertos en codificación de video (VCEG) de la Unión Internacional de Telecomunicaciones – Sector Telecomunicaciones (ITU-T) y el grupo de expertos para imágenes en movimiento (MPEG) de la ISO/IEC, enfocaron sus investigaciones en las diferentes técnicas de codificación de video para diversas aplicaciones, de las cuales se va a profundizar en el H263 y H264 que son las más recientes.

2.15 H.263

H.263 es un estándar de la Unión Internacional de Telecomunicaciones (ITU) para la codificación de vídeos con compresión, describe un Codec, que su objetivo principal fue enfocado para videoconferencias. Está optimizado para una tasa de bits baja y un movimiento relativo reducido.

Se basa en técnicas comunes de compresión de video como son: la compresión espacial, compresión temporal, predicción y compensación de movimiento.

2.16 H.264

El H.264 o MPEG-4 parte 10 fue creado por el equipo conjunto de video (JVT)⁶⁶, el cual fue conformado por el grupo de expertos de la ITU-T y el grupo de expertos de la ISO/IEC, su objetivo fue el desarrollo de un nuevo estándar, con un mejor desempeño, tanto en la calidad de video como en la eficiencia de codificación.

El H.264 posee especificaciones simples de su sintaxis, lo cual proporciona una mejor integración con todos los protocolos actuales y arquitecturas múltiples, además permite incluir otras aplicaciones, tales como la transmisión de video y video conferencia en redes fijas e inalámbricas y en diferentes protocolos de transporte.

El estándar H.264, es una norma que define un códec de vídeo de alta compresión, capaz de proporcionar una buena calidad de imagen, es utilizado en la técnica streaming por el ahorro de ancho de banda sin disminución de la calidad. No sólo es eficiente para el almacenamiento de video, sino que también proporciona un alto rendimiento en compresión y es más robusto a errores de transmisión que sus antecesores MPEG-2, H.263 y MPEG-4 parte 2.

⁶⁶ Sus siglas significan Joint Video Team.

2.16.1 Estructura de las capas

Se compone de dos capas⁶⁷:

- Capa de la red de abstracción (NAL)⁶⁸
- Capa de codificación del video (VCL)⁶⁹

NAL.- Abstrae los datos para hacer compatible al tren de bits de salida del codificador con casi todos los canales de comunicación o medios de almacenamiento. Esta unidad de red, especifica los datos en un formato de bytes (byte-stream) o de paquetes.

VCL.- constituye el núcleo de los datos codificados. Ésta consiste en la secuencia de video a codificar.

2.17 AAC

Códec de audio, basado en el algoritmo de compresión con pérdida, un proceso por el que se eliminan algunos de los datos de audio para poder obtener el mayor grado de compresión posible, resultando en un archivo de salida que suena lo más parecido posible al original.

Es un estándar creado por MPEG. Debido a su excepcional rendimiento y la calidad, el AAC se encuentra en el núcleo del MPEG-4, 3GPP y 3GPP2, y es el códec de audio de elección para Internet, conexiones inalámbricas y de radio difusión digital.

AAC es un algoritmo de codificación de banda ancha de audio que tiene un rendimiento superior al del MP3, que produce una mejor calidad en archivos pequeños y requiere menos recursos del sistema para codificar y descodificar⁷⁰.

⁶⁷ (Ochoa-Domínguez & García, 2007)

⁶⁸ Network Abstraction Layer

⁶⁹ Video Coding Layer

⁷⁰ (Wikipedia, Advanced Audio Coding, 2010)

2.18 AMR

Adaptive Multi-Rate es un formato de compresión de audio optimizado para la codificación de voz. AMR ha sido adoptado como el estándar de codificación de audio por 3gpp en octubre de 1998 y actualmente se utiliza ampliamente en GSM. Gestiona dinámicamente el ancho de banda seleccionando entre ocho diferentes tasas de bits.

Existen programas para transformar este formato a formatos de audio como son el MP3 o el WAV. El uso de éstos es relativamente necesario a causa de que no todos los reproductores de sonido tienen por defecto los archivos necesarios como para reproducir un archivo codificado como AMR⁷¹.

2.19 Metodología

La metodología escogida para este proyecto es programación extrema, ya que es necesario realizar varias pruebas e ir mejorando hasta encontrar la versión ideal.

La programación extrema es un enfoque de la ingeniería de software formulado por Kent Beck. Es una de las metodologías más destacadas de los procesos ágiles de desarrollo de software, se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la planificación.

La principal característica de la programación extrema es que se adapta a los cambios de requisitos en cualquier punto de la vida del proyecto, teniendo así un software con una aproximación mejor y más realista.

⁷¹ (Wikipedia, Compresión multi-tasa adaptativa, 2010)

2.19.1 Etapas

Las etapas de la programación extrema son:

- Planificación
- Diseño
- Codificación
- Pruebas

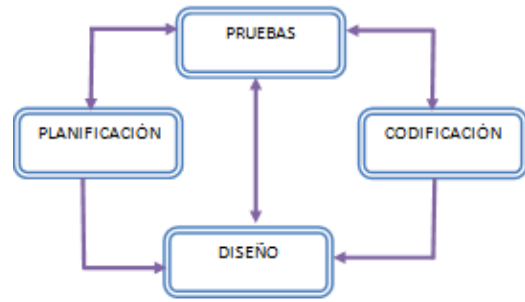


Ilustración 2.18- Ciclo XP

2.20 Herramientas de Programación Específica

2.20.1 Java Development Kit

Más conocido como JDK, es un software que provee herramientas de desarrollo para la creación de programas en java. Puede instalarse en una computadora local o en una unidad de red. En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación⁷².

Los programas que incluyen el JDK son:

- **Appletviewer**: es un visor de applet para generar sus vistas previas.
- **Javac**: es el compilador de JAVA.
- **java**: es el intérprete de JAVA.
- **javadoc**: genera la documentación de las clases java de un programa.

2.20.1.1 JDK en Windows

Para que el JDK funcione en Windows además de instalarlo se debe configurar las diferentes variables de entorno:

- **JAVAPATH**: path del directorio donde está instalado el JDK.
- **CLASSPATH**: son las librerías o clases de usuario.
- **PATH**: variable donde se agrega la ubicación de JDK

⁷² (Wikipedia, JDK, 2010)

2.20.2 ANDROID SDK

El desarrollo de aplicaciones Android se realiza mediante SDK, y el lenguaje de programación Java.

El SDK (Software development kit) de Android proporciona las herramientas y las API necesarias para empezar a desarrollar aplicaciones que se puedan ejecutar en dispositivos con la tecnología de Android.

El SDK de Android incluye un conjunto completo de herramientas de desarrollo:

- Un depurador
- Un conjunto de Bibliotecas.
- Un emulador de terminal (basado en QEMU⁷³),
- Documentación.
- Código de ejemplo
- Tutoriales

Android SDK en sí es programar en java usando una serie de librerías concretas, y donde tenemos la opción de usar un IDE como Eclipse o NetBeans.

2.20.3 JADE-LEAP

La unión entre la plataforma JADE y las bibliotecas LEAP permite obtener una plataforma de agentes FIPA compatible con espacio reducido y la compatibilidad con entornos Java móviles.

Proporciona los mecanismos adecuados para la construcción de sistemas multiagente, la comunicación y construcción de agentes.

2.20.4 JADE-ANDROID

Jade-Android es un complemento de Jade que proporciona soporte para el uso de JADE-LEAP en la plataforma de Android.

⁷³ Emulador de procesadores basado en la traducción dinámica de binarios

Jade-Android es un paquete de software que permite el desarrollo de aplicaciones orientadas a agente basado en JADE para la plataforma Android.

Jade-Android es la posibilidad de combinar la expresividad de los agentes JADE con el poder de la plataforma Android, convirtiéndose en una herramienta muy importante en el desarrollo de aplicaciones innovadoras basadas en modelos sociales y paradigmas peer-to-peer.

Se distribuye como un add-on de JADE y está disponible para su descarga en los términos de la licencia LGPL. El complemento también incluye a un agente de aplicación de ejemplo9.

2.20.4.1 ECLIPSE

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma, tiene soporte para lenguajes de programación java, php, c++, perl.

El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código.

Eclipse tiene ayuda incorporada para autocompletar así como también nos brinda la posibilidad de conocer todos los atributos y funciones de la clase que se está utilizando y que hacen cada uno de ellos.

Una de las principales características de Eclipse es que nos permite instalar plugins, que se los puede descargar del internet, haciéndolo más funcional, gracias a esta opción y junto con el android SDK se puede realizar las aplicaciones para Android.

Eclipse cuenta con una interfaz gráfica para poder manejar las aplicaciones Android, haciendo más sencillo estructurar una aplicación y siendo esta una gran ventaja que posee Android frente a otros IDEs's.

2.20.4.1.1 Historia

Eclipse comenzó como un proyecto de IBM Canadá. Fue desarrollado por OTI⁷⁴ como reemplazo de VisualAge también desarrollado por OTI.

En noviembre del 2001, se formó un consorcio para el desarrollo futuro de Eclipse como código abierto.

Las versiones que ha ido teniendo Eclipse en han sido las siguientes:

Tabla 2.3 - Versiones de Eclipse

Nombre	Fecha	Número
<i>Callisto</i>	2006	La fundación Eclipse coordinó sus 10 proyectos de código abierto y lanza la primera versión de Eclipse
<i>Europa</i>	29 de Junio del 2007	Versión 3.3 de Eclipse
<i>Ganymede</i>	25 de Junio del 2008	Versión 3.4 de Eclipse
<i>Galileo</i>	24 de junio del 2009.	Versión 3.5 de Eclipse
<i>Helios</i>	23 de junio de 2010	Versión 3.6 de Eclipse

⁷⁴Object Technology International

2.20.4.1.2 Plugin de Android

Para integrar Android con Eclipse, necesitamos un Plugin, que dependiendo de la versión del IDE que tengamos, se instalará de una forma concreta.

2.20.4.2 VLC

VLC media player es un reproductor y framework multimedia del proyecto VideoLAN; es software libre distribuido bajo la licencia GPL. Soporta muchos códecs de audio y video, así como diferentes formatos de archivos, además de DVD, VCD y varios protocolos de streaming; también tiene la capacidad de transmitir datos streaming a través de redes y convertir archivos multimedia en formatos distintos al original.

Es uno de los reproductores más independientes en cuanto a plataforma se refiere, con versiones para GNU/Linux, Microsoft Windows, Mac OS X, BeOS y BSD, entre otros.

VLC incluye de forma nativa un gran número bibliotecas de códecs, reduciendo la necesidad de instalar o calibrar códecs propietarios. Muchos de los códecs incluidos en VLC son proporcionados por la biblioteca libavcodec.

VLC media player puede leer multitud de archivos dependiendo del sistema operativo.

2.21 Rendimiento de los servicios para video streaming

2.21.1 RETARDO

Es el intervalo de tiempo que existe entre la transmisión de dos señales sincronizadas, desde la misma o distinta estación.

2.21.1.1 Fórmula del Retardo

$$\text{Retardo} = T_{n-1} - T_n \text{ (ms)}$$

Donde:

- T_{n-1} = tiempo de recepción del anterior paquete.

- T_n = tiempo de llegada del paquete n.
- Unidades de tiempo Ms = milisegundos

2.21.2 JITTER

Es la diferencia entre el tiempo que llega un paquete y el tiempo en que se creará que llegará el paquete, es decir la variación en el retardo.

El jitter se considera como una señal de ruido no deseada. En general se denomina jitter a un cambio indeseado de la propiedad de una señal. Esto puede afectar tanto a la amplitud como a la frecuencia y la situación de fase, es la primera consecuencia de un retraso de la señal

Es especialmente molesto en aplicaciones multimedia en Internet o telefonía IP y en técnicas streaming, ya que provoca que algunos paquetes lleguen demasiado pronto o tarde para poder entregarlos a tiempo, el efecto puede reducirse con un búfer de jitter, un búfer de datos, pero a costa de un tiempo de ejecución mayor.

El jitter es un efecto de las redes no orientadas a la conexión y basadas en conmutación de paquetes. Como la información se divide en paquetes cada uno de los paquetes puede seguir una ruta distinta para llegar al destino.

2.21.2.1 Fórmula del Jitter

La siguiente fórmula para calcular el jitter se obtuvo de la documentación del Wireshak.

$$\mathbf{Jitter(i)} = J(i-1) + (|D(i-1),i| - J(i-1)) / 16$$

Donde:

- $D(i,j) = (R_j - S_j) - (R_i - S_i)$
 - S_i = es el timestamp para el paquete i.
 - R_i = es el tiempo de llegada en timestamp unitario para cada paquete i.

2.22 Factores que provocan la desconexión WIFI

Las redes Wifi son cómodas, nos permiten una buena movilidad y nos ahorran una buena cantidad de cableado, pero también tienen una serie de inconvenientes provocados por algunos factores que hacen que la red pierda la estabilidad, y sufra varias desconexiones.

Los factores más importantes que provocan desconexiones son:

- Banda que utiliza.- las conexiones Wifi trabajan en la banda de 2.4GHz, esta banda es utilizada por una gran cantidad de dispositivos de comunicación, incluso recibe interferencias de una gran cantidad de electrodomésticos, esto hace que la señal muchas veces se pierda, y sea casi imposible volver a reconectarse.
- Interferencias.- sensible a emisiones de radio y de televisión, proximidad de otras redes inalámbricas, cámara inalámbrica.
- Factores atmosféricos.- como la nieve, la lluvia o el granizo, pueden interferir en la señal.
- Limitación de ancho de banda.
- Tráfico de red.
- Pérdida de cobertura.
- Potencia de emisión.
- Protocolo de red inestable

2.23 Métodos estadísticos

2.23.1 Muestreo

Una parte fundamental para realizar un estudio estadístico de cualquier tipo es obtener unos resultados confiables y que puedan ser aplicables. Resulta casi imposible o impráctico llevar todos los paquetes de un video transmitido para realizar cálculos o gráficas, por lo

que la solución es llevar a cabo el estudio basándose en un subconjunto de paquetes denominada muestra.

Sin embargo, para que los estudios tengan la validez y confiabilidad buscada es necesario que la muestra, posea algunas características específicas que permitan, al final, generalizar los resultados. Esas características tienen que ver principalmente con el tamaño de la muestra y con la manera de obtenerla.

Para calcular el tamaño de una muestra hay que tomar en cuenta tres factores:

1. *La confianza o el porcentaje de confianza.*- es el porcentaje de seguridad que existe para generalizar los resultados obtenidos. Generalmente es del 95%, pero siempre debe ser menor del 100%.
2. *El error o porcentaje de error.*- equivale a elegir una probabilidad de aceptar una hipótesis que sea falsa como si fuera verdadera, o la inversa: rechazar a hipótesis verdadera por considerarla falsa. Comúnmente se aceptan entre el 4% y el 6% como error.
3. *La variabilidad es la probabilidad.*- con el que se aceptó y se rechazó la hipótesis que se quiere investigar en alguna investigación anterior o en un ensayo previo a la investigación actual. Cuando se habla de la máxima variabilidad, en el caso de no existir antecedentes sobre la investigación, entonces los valores de variabilidad sería 0.5.

Una vez que se han determinado estos tres factores, entonces se puede calcular el tamaño de la muestra, existen dos fórmulas cuando la muestra total es conocida y cuando no lo es, a continuación se indicará la fórmula que va ser utilizada, conociendo la muestra total:

$$n = \frac{Z^2 pqN}{NE^2 + Z^2 pq}$$

Donde:

n = es el tamaño de la muestra.

Z = es el nivel de confianza (0.98).

p = es la variabilidad positiva (0.05).

q = es la variabilidad negativa (0.05).

N = es el tamaño de la población (depende de la muestra).

E = es la precisión o el error (0.02).

Para poder obtener las muestras de los paquetes enviados se utilizará una herramienta denominada sniffer, tanto en el lado del servidor como en el lado del teléfono móvil, esta herramienta nos ayudará a conocer el tráfico de la red, los bytes enviados en un tiempo determinado.

El sniffer que se ha decidido utilizar es el WireShark es de software libre, trabaja bajo licencia GPL y es utilizado para realizar análisis y solucionar problemas en redes de comunicaciones para desarrollo de software y protocolos. Además maneja una interfaz gráfica que permite filtrar la información y obtener solo los datos necesarios. Así como herramientas complementarias para el análisis del tráfico de red.

CAPITULO III

3 DESARROLLO DE LA APLICACION

3.1 Introducción.

La transmisión de video utiliza el mecanismo Streaming, el mismo permite la descarga parcial del video en pequeños fragmentos de menor tamaño en el Cliente, mejorando la usabilidad al enviar una secuencia de cuadros de video que presentan bajas demoras al iniciar la visualización y menores requerimientos de almacenamiento.

Durante el proceso de transmisión del video, podrían existir rupturas o cortes especialmente en comunicación con redes inalámbricas; sin embargo, si se privilegia la usabilidad del sistema, es decir, que la descarga sea paulatina a medida en que divide en streams el video a transmitir, sería deseable que al sufrir una desconexión el propio aplicativo identifique el restablecimiento de la comunicación y continúe el proceso de descarga sin que el usuario vuelva a reiniciar la sesión. Es en este escenario donde los sistemas basados en Agentes de Software como JADE, figuran como alternativa para controlar y mejorar el proceso de transmisión de datos, incrementando de esta forma, la usabilidad del sistema.

Para poder realizar la aplicación que cumpla con todos estos requisitos, se necesita de las siguientes herramientas de programación con sus versiones respectivas:

Tabla 3.1 - Herramientas y Versiones

Herramientas	Versión
<i>Jdk</i>	1.6
<i>Android SDK</i>	2.2
<i>Eclipse</i>	3.5

<i>Jade</i>	4.0.1
<i>Jade-Leap</i>	4.0
<i>Jade-Android</i>	1.2
<i>Vlc</i>	1.4.1

Todas estas herramientas deben ser instaladas y configuradas previamente para poder iniciar con la aplicación, para esto consulte en la sección Anexos, anexo 1.-Manual de instalación.

Además de esto se necesita de un celular android, ya que en el emulador la aplicación no funciona completamente, se realiza hasta la negociación RTSP, pero el momento de reproducir el audio y video no lo hace, por limitaciones de memoria.

3.2 Planificación

3.2.1 Historias de Usuario

Tabla 3.2 - Historia de Usuario 1

Historia de Usuario	
Número: 1	Usuario: Ing. Diego Marcillo
Nombre historia: Reproducir video sin proxy.	
Prioridad en negocio: Alta (Alta/media/baja)	Riesgo en desarrollo: Medio (Alto/medio/bajo)
Puntos estimados:	Iteración asignada: 1
Programador responsable: Cristina Navarro – Darío Larenas	
Descripción: Visualizar un video del internet en el Nexus One.	
Observaciones:	

Tabla 3.3 - Historia de Usuario 2

Historia de Usuario	
Número: 2	Usuario: Ing. Diego Marcillo
Nombre historia: Reconexión automática	
Prioridad en negocio: Alta (Alta/media/baja)	Riesgo en desarrollo: Alto (Alto/medio/bajo)
Puntos estimados:	Iteración asignada: 5
Programador responsable: Cristina Navarro – Darío Larenas	
Descripción: El momento de estar utilizando el servicio de video streaming por factores externos se pierde la conexión, lo que se solicita es que, cuando se vuelva al área de cobertura se reconecte de manera automática y siga la transmisión, desde donde se quedó sin perder ninguna información y sin iniciar sesión nuevamente.	
Observaciones:	

Tabla 3.4 - Historia de Usuario 3

Historia de Usuario	
Número: 3	Usuario: Ing. Diego Marcillo
Nombre historia: Configurar Proxy	
Prioridad en negocio: Alta (Alta/media/baja)	Riesgo en desarrollo: Bajo (Alto/medio/bajo)
Puntos estimados:	Iteración asignada: 1
Programador responsable: Cristina Navarro – Darío Larenas	
Descripción: Agregar un menú de configuración de proxy a la aplicación ya terminada.	
Observaciones:	

3.2.2 Plan Realese

Tabla 3.5 - Plan Release

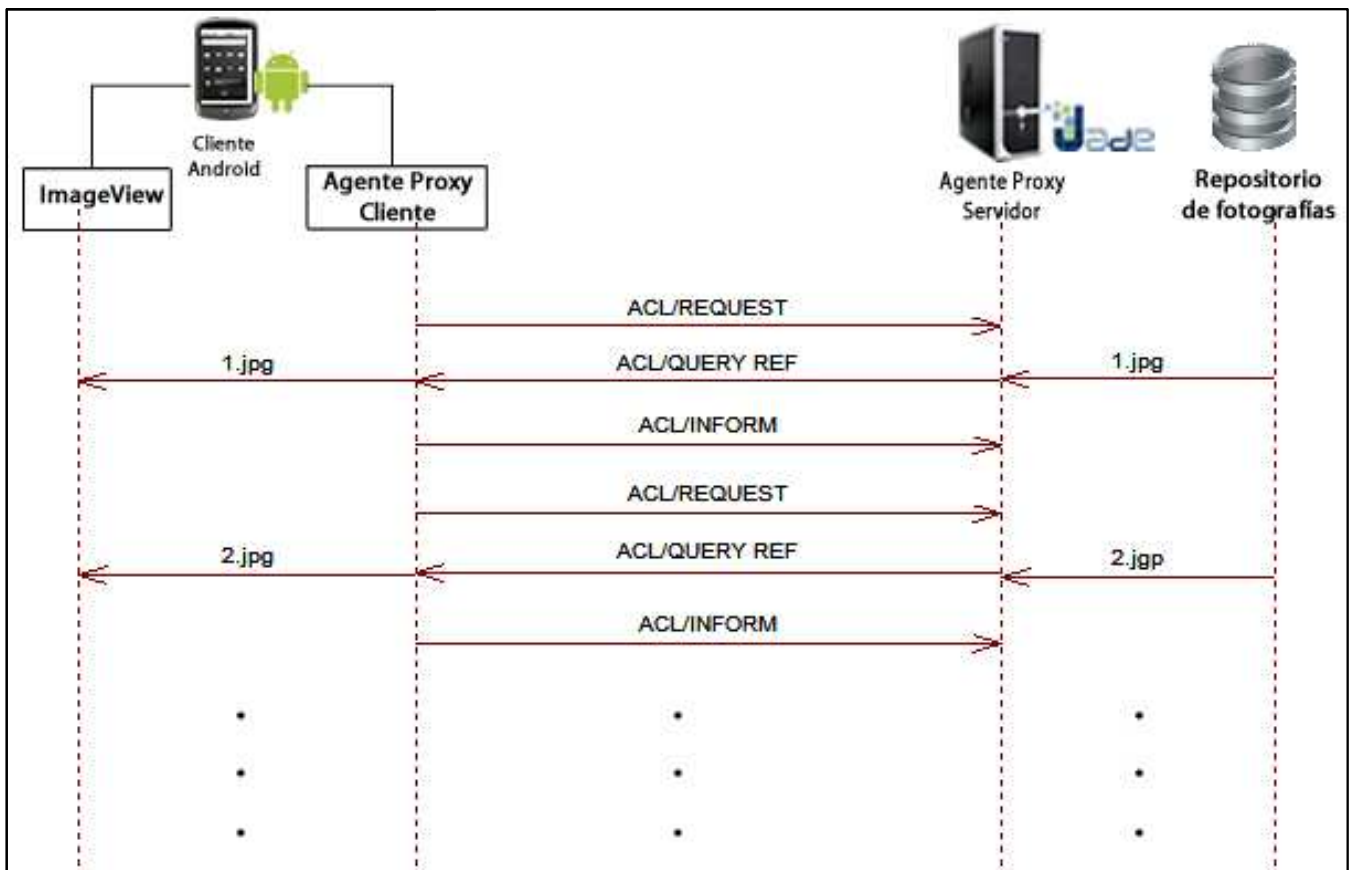
# Historia de Usuario	Tareas	Actividades	Descripción	Encargado	Tipo	Fecha Inicio	Fecha Fin
1	Reproducir Video sin proxy	Reproducir un video utilizando el protocolo RTSP, desarrollo solo de la aplicación cliente.	Activar la librería propia de android, que permite la visualización en el Nexus One de los video streaming con RTSP	Cristina Navarro Darío Larenas	Desarrollo	02-08-2010	13-08-2010
2	Reconexión Automática	Pasar imágenes con agentes Jade, desarrollo de los agentes: <ul style="list-style-type: none"> • Proxy servidor (APS). • Proxy Cliente (APC) 	Configurar la plataforma y los agentes Jade para que trabajen conjuntamente con Android, para probar el rendimiento de los mismos.	Cristina Navarro Darío Larenas	Desarrollo	16-08-2010	03-09-2010
2		Pasar video con agentes Jade, desarrollo de los agentes: <ul style="list-style-type: none"> • Proxy servidor (APS). • Proxy Cliente (APC) 	Se utilizará la misma estructura anterior, pero se pasará un video	Cristina Navarro Darío Larenas	Desarrollo	06-09-2010	17-09-2010

2	Reconexión Automática	<p>Pasar video streaming utilizando el protocolo RTSP y agentes JADE, desarrollo de los agentes:</p> <ul style="list-style-type: none"> • Proxy servidor (APS). • Proxy Cliente (APC) 	<p>Luego de todas las pruebas realizadas, se pondrá en marcha la aplicación que unirá Jade, android y RTSP</p>	<p>Cristina Navarro Darío Larenas</p>	<p>Desarrollo</p>	<p>20-09-2010</p>	<p>01-10-2010</p>
		<p>Desarrollar agentes de manera dinámica en el Proxy Servidor (APS)</p>	<p>APS sea capaz de crear agentes según las necesidades.</p>	<p>Cristina Navarro Darío Larenas</p>	<p>Mejora</p>	<p>04-10-2010</p>	<p>08-10-2010</p>
3	Configurar Proxy	<p>Agregar menú de configuración de proxy a la aplicación.</p>	<p>Menú donde el usuario pueda activar / desactivar el proxy además ayudará a la configuración del puerto y de la dirección IP del contenedor JADE, y el puerto local que se utilizará para la comunicación RTSP</p>	<p>Cristina Navarro Darío Larenas</p>	<p>Mejora</p>	<p>11-10-2010</p>	<p>15-10-2010</p>

3.3 Diseño

3.3.1 Paso de imágenes con Agente JADE

La arquitectura a establecer es la siguiente: un Agente Jade que realiza la función de servidor, al cual se le designó el comportamiento de obtener la imagen del repositorio y enviarla al cliente. En el cliente, que es el dispositivo móvil Android, se instaló otro Agente cuyo comportamiento fue recibir la fotografía y desplegarla en la pantalla.



3.3.2 Paso de video streaming utilizando el protocolo RTSP y agentes JADE .

3.3.3 Aplicación Jade -Streaming

Se procedió a adaptar la arquitectura ya explicada de los agentes Jade al protocolo RTSP/RTP que ha logrado que la técnica de video streaming sea muy utilizada, y esté vigente hasta la actualidad. El éxito del protocolo RTSP, se debe a su velocidad y eficiencia a la hora de transmitir videos de gran tamaño y calidad. A esta eficiencia en los dispositivos móviles se le añade los codec de video H263 y H264, que comprimen mediante algoritmos el tamaño de los fotogramas, para evitar ocupar un gran espacio de memoria en el dispositivo móvil. RTSP envía al video por dos canales, uno donde emitirá los fotogramas y otra el audio, los cuales obedecen a un instante en el tiempo para poderlos sincronizar.

La característica principal de RTSP es enviar paquetes RTP a través del protocolo de transporte UDP, lo cual hace que los paquetes sean enviados y recibidos de manera rápida, porque no es orientado a la conexión, siendo esta su gran fortaleza, pero también se convierte en su mayor debilidad, ya que nadie garantiza que el cliente reciba los paquetes o que exista la pérdida de estos, que es evidente en una desconexión del cliente.

Para superar este gran inconveniente, el mecanismo que realizará la transferencia de video en tiempo real sobre teléfonos móviles, determina las siguientes entidades: Un Servidor de video RTSP, un Agente Proxy Servidor - APS, un dispositivo móvil que actuará como cliente, albergando un Agente Proxy Cliente - APC y a la aplicación RTSP Cliente la cual se llama PV Player y viene integrada al dispositivo Android, esta generará las peticiones y visualizará el video.

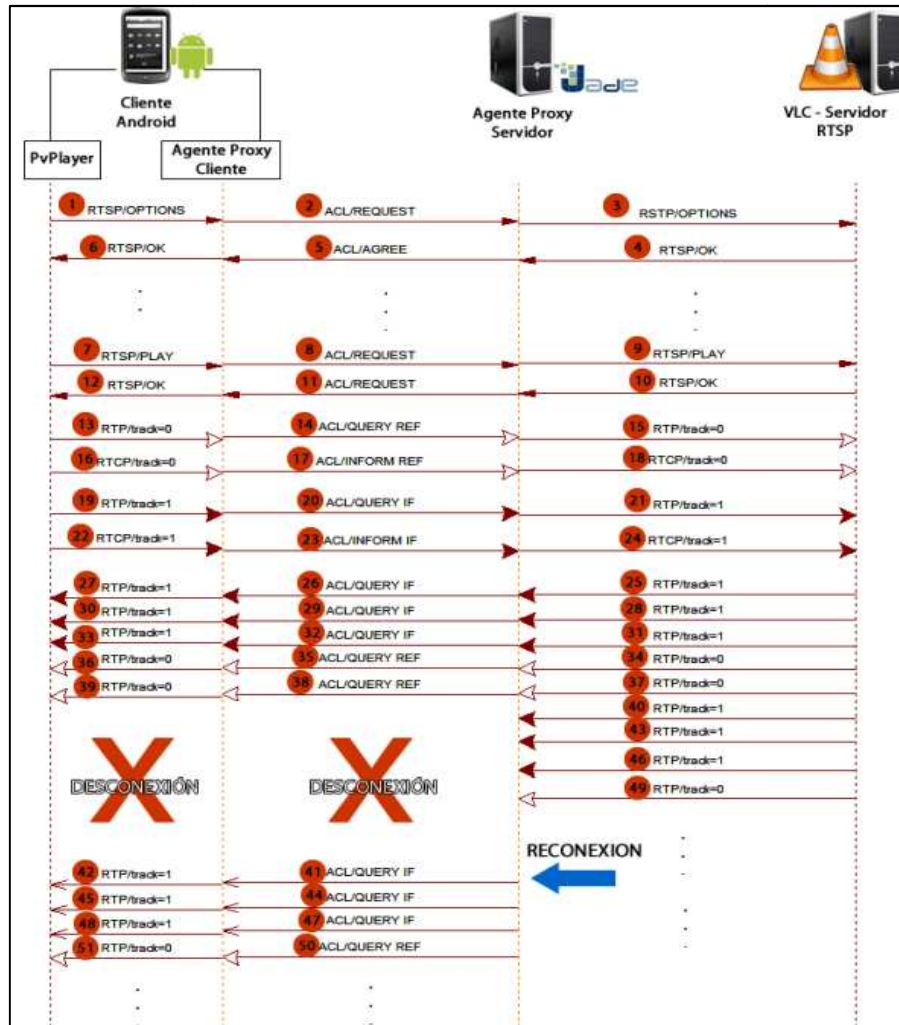


Ilustración 3.2 - Arquitectura JADE - RTSP

El APS cumple dos funciones, la primera es la de recibir los mensajes del APC y los envía al servidor según los puertos asignados en la negociación y la función restante es recibir los mensajes provenientes del servidor y enviarlos al APC

Este agente va a tener diez comportamientos:

Tabla 3.6 - Comportamientos APC

Comportamiento	Mensaje ACL	Origen	Destino	Descripción
ListenerMensajeRTSP	REQUEST	APC	Servidor VLC Puerto 1234	Negociación RTSP
ListenerMensajeRTPTrack0	QUERY_REF	APC	Servidor VLC Puerto aleatorio	Pista de Audio
ListenerMensajeRTCPTrack0	INFORM_REF	APC	Servidor VLC Puerto aleatorio	Control de Pistas de Audio
ListenerMensajeRTPTrack1	QUERY_IF	APC	Servidor VLC Puerto aleatorio	Pista de Video
ListenerMensajeRTCPTrack1	INFORM_IF	APC	Servidor VLC Puerto aleatorio	Control de Pistas de Video
ListenerPuertoRTSP	AGREE	Servidor VLC Puerto 1234	APC	Respuesta a negociación RTSP

ListenerPuertoRTPTrack0	QUERY_REF	Servidor VLC Puerto aleatorio	APC	Pista de Audio
ListenerPuertoRTCPTrack0	INFORM_REF	Servidor VLC Puerto aleatorio	APC	Control de Pistas de Audio
ListenerPuertoRTPTrack1	QUERY_IF	Servidor VLC Puerto aleatorio	APC	Pista de Video
ListenerPuertoRTCPTrack1	INFORM_IF	Servidor VLC Puerto aleatorio	APC	Control de Pistas de Video

El APC permite que el dispositivo móvil pueda recibir y enviar los mensajes RTSP, los paquetes RTP y RTCP de manera segura. Este APC implementa un mecanismo de almacenamiento y envío, así como el de filtrar la negociación RTSP para que sea lo más transparente posible. Este reside en el dispositivo móvil lo cual asegura que el agente nunca se va a desconectar del cliente.

Este agente va a tener diez comportamientos:

Tabla 3.7 - Comportamientos APS

Comportamiento	Mensaje ACL	Origen	Destino	Descripción
ListenerMensajeRTSP	AGREE	APS	PV Player puerto 1234	Respuesta a negociación RTSP
ListenerMensajeRTPTrack0	QUERY_REF	APS	PV Player puerto aleatorio	Pista de Audio
ListenerMensajeRTCPTrack0	INFORM_REF	APS	PV Player puerto aleatorio	Control de Pistas de Audio
ListenerMensajeRTPTrack1	QUERY_IF	APS	PV Player puerto aleatorio	Pista de Video
ListenerMensajeRTCPTrack1	INFORM_IF	APS	PV Player puerto aleatorio	Control de Pistas de Video
ListenerPuertoRTSP	REQUEST	PV Player puerto 1234	APS	Petición RTSP
ListenerPuertoRTPTrack0	QUERY_REF	PV Player puerto aleatorio	APS	Pista de Audio
ListenerPuertoRTCPTrack0	INFORM_REF	PV Player puerto	APS	Control de Pistas de Audio

		aleatorio		
ListenerPuertoRTPTrack1	QUERY_IF	PV Player puerto aleatorio	APS	Pista de Video
ListenerPuertoRTCPTrack1	INFORM_IF	PV Player puerto aleatorio	APS	Control de Pistas de Video

Al colocar agentes en el *APS* y *APC*, éstos cooperan en definir cuándo el móvil está fuera de cobertura al usar señalización MTP, encargándose de resolver las desconexiones intermitentes y reanudar automáticamente la sesión de video streaming. Los mensajes transmitidos FIPA-ACL al teléfono móvil son guardados en el buffer del *APS* ya que no pueden ser entregados al destinatario, reenviándose una vez que se reanude la conexión.

Uno de los beneficios de JADE ANDROID es que intentará reconectar el *APS* y el *APC* durante algún tiempo. El protocolo MTP permite definir el tiempo de espera para que el cliente se reconecte, valor que por defecto es un minuto; éste tiempo influye en el dimensionamiento del buffer. Al restablecerse la conexión entre los agentes, el *APC* leerá los streams ordenados en el buffer y los enviará al video *reproductor del Cliente*. De esta forma se permite que el cliente recupere el video desde el punto de quiebre.

3.3.4 Diagramas

3.3.5 Diagrama de casos de Uso

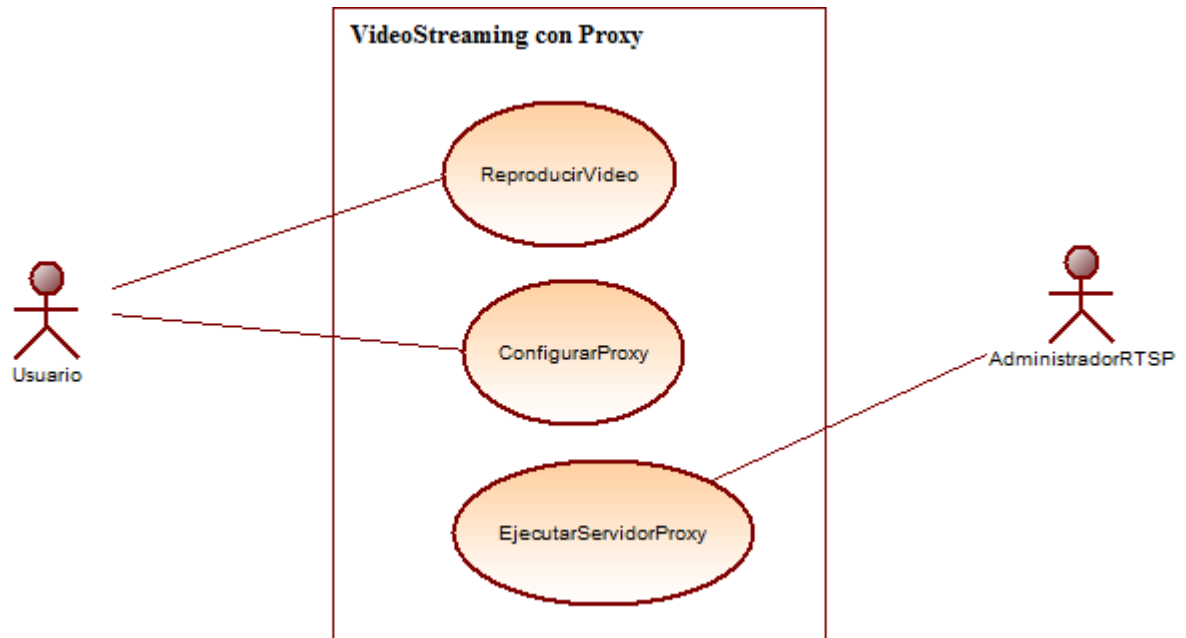


Diagrama 3.1 - Casos de Uso

3.3.6 Diagrama de clases APS

3.3.6.1.1 Contenedor

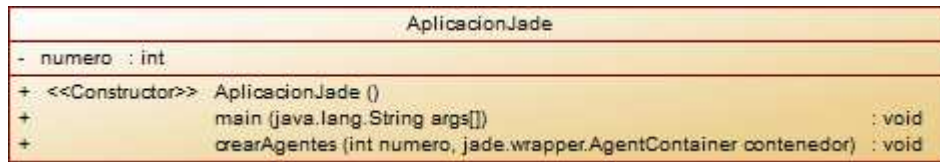


Diagrama 3.2 - Diagrama de Clases APS Paquete Contenedor

3.3.6.1.2 Proxy

Diagrama 3.3- Diagrama de Clases APS Paquete Proxy

3.3.6.1.3 Comportamientos

Diagrama 3.4 - Diagrama de Clases APS Paquete Comportamientos

3.3.6.1.4 Utilitarios

UriRTSP	
-	host : java.lang.String
-	puerto : int
-	ip : java.lang.String
-	tienePuerto : boolean
-	recurso : java.lang.String
+	<<Constructor>> UriRTSP (java.lang.String URL)
+	getRecurso () : java.lang.String
+	setRecurso (java.lang.String recurso) : void
+	getURL () : java.lang.String
+	getHost () : java.lang.String
+	setHost (java.lang.String host) : void
+	getPuerto () : int
+	setPuerto (int puerto) : void
+	getIp () : java.lang.String
+	setIp (java.lang.String ip) : void
+	getTienePuerto () : boolean
MensajeRTSP	
-	CRLF : java.lang.String
+	<<Constructor>> MensajeRTSP ()
+	getURL (java.lang.String mensaje) : java.lang.String
+	cambiarPuertoCliente (java.lang.String mensaje, int rtp, int rtcp) : java.lang.String
+	getPuertosCliente (java.lang.String mensaje) : int[]
+	cambiarPuertoServidor (java.lang.String mensaje, int rtp, int rtcp) : java.lang.String
+	getPuertosServidor (java.lang.String mensaje) : int[]
+	getSource (java.lang.String mensaje) : java.lang.String
+	cambiarSource (java.lang.String mensaje, java.lang.String source) : java.lang.String
+	cambiarLongitudContenidoMensaje (java.lang.String mensaje) : java.lang.String
+	getContenidoMensaje (java.lang.String mensaje) : java.lang.String
+	getLongitudContenidoMensaje (java.lang.String mensaje) : int
+	cambiarUriRTSP (java.lang.String mensaje, UriRTSP url, UriRTSP urlSustituto) : java.lang.String
+	cambiarHost (java.lang.String mensaje, UriRTSP url, UriRTSP urlSustituto) : java.lang.String
-	getComando (java.lang.String mensaje) : java.lang.String
+	isDescribe (java.lang.String mensaje) : boolean
+	isOptions (java.lang.String mensaje) : boolean
+	isPause (java.lang.String mensaje) : boolean
+	isplay (java.lang.String mensaje) : boolean
+	isSetup (java.lang.String mensaje) : boolean
+	isSetupTrack0 (java.lang.String mensaje) : boolean
+	isSetupTrack1 (java.lang.String mensaje) : boolean
+	isTearDown (java.lang.String mensaje) : boolean
+	isSetupRespuesta (java.lang.String mensaje) : boolean

Diagrama 3.5 - Diagrama de Clases APS Paquete Util

3.3.7 Diagrama de Secuencia APS

3.3.7.1.1 Ejecutar Servidor Proxy

Diagrama 3.6 - Diagrama de Secuencia APS Ejecutar Servidor Proxy

3.3.8 Diagrama de clases APC

3.3.8.1.1 Aplicación

Diagrama 3.7 - Diagrama de Clases APC Paquete Aplicación

3.3.8.1.2 Configuración

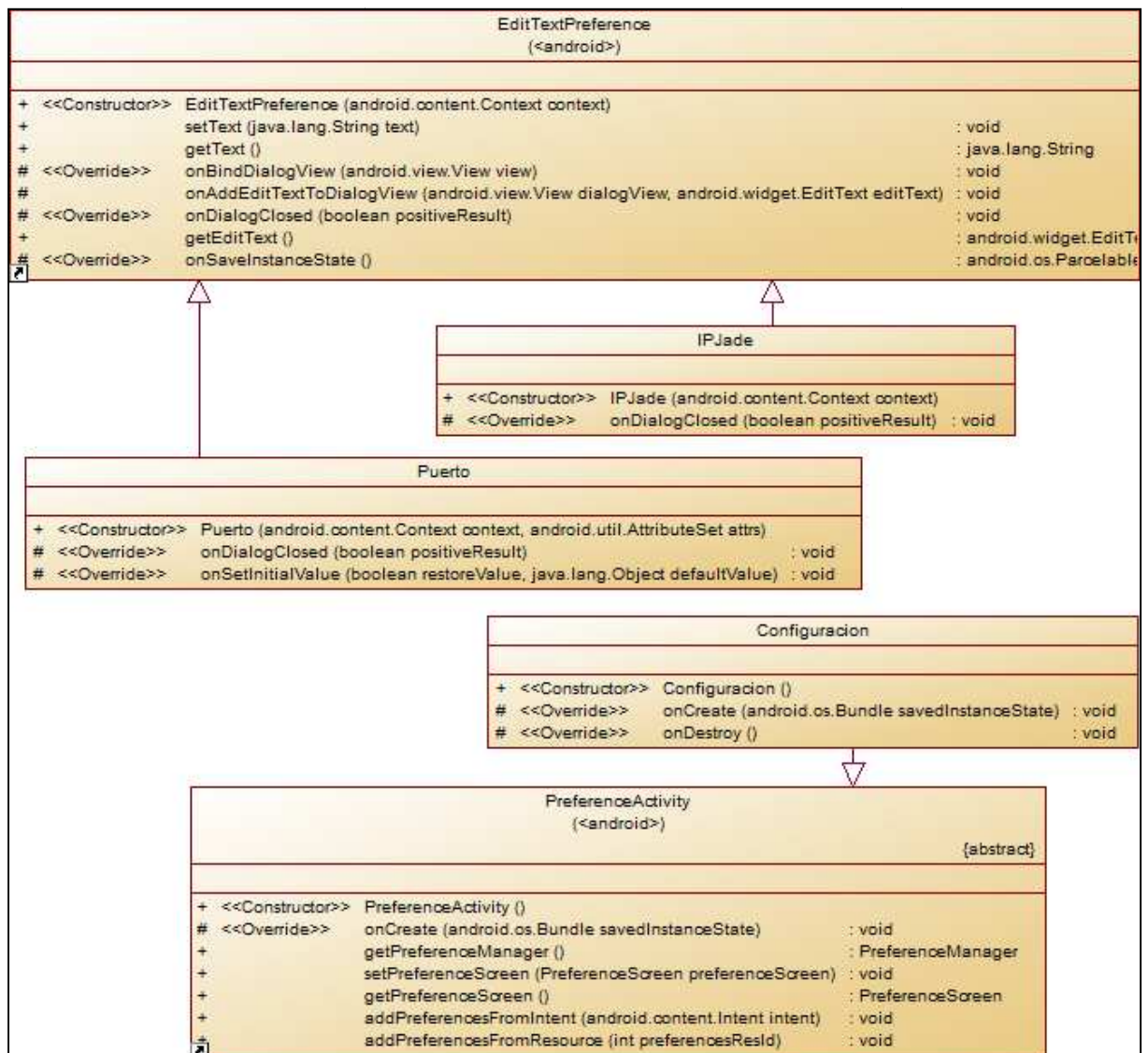


Diagrama 3.8 - Diagrama de Clases APC Paquete Configuración

3.3.8.1.3 Proxy

Diagrama 3.9 - Diagrama de Clases APC Paquete Proxy

3.3.8.1.4 Comportamiento

Diagrama 3.10 - Diagrama de Clases APC Paquete Comportamiento

3.3.8.1.5 Utilitarios

UriRTSP	
-	host : java.lang.String
-	puerto : int
-	ip : java.lang.String
-	recurso : java.lang.String
+	<<Constructor>> UriRTSP (java.lang.String URL)
+	getRecurso () : java.lang.String
+	setRecurso (java.lang.String recurso) : void
+	getURL () : java.lang.String
+	getHost () : java.lang.String
+	setHost (java.lang.String host) : void
+	getPuerto () : int
+	setPuerto (int puerto) : void
+	getIp () : java.lang.String
+	setIp (java.lang.String ip) : void
+	getTienePuerto () : boolean
MensajeRTSP	
+	CRLF : java.lang.String = "..."
+	<<Constructor>> MensajeRTSP ()
+	getURL (java.lang.String mensaje) : java.lang.String
+	cambiarPuertoCliente (java.lang.String mensaje, int rtp, int rtp) : java.lang.String
+	getPuertosCliente (java.lang.String mensaje) : int[]
+	cambiarPuertoServidor (java.lang.String mensaje, int rtp, int rtp) : java.lang.String
+	getPuertosServidor (java.lang.String mensaje) : int[]
+	getSource (java.lang.String mensaje) : java.lang.String
+	cambiarSource (java.lang.String mensaje, java.lang.String source) : java.lang.String
+	cambiarLongitudContenidoMensaje (java.lang.String mensaje) : java.lang.String
+	getContenidoMensaje (java.lang.String mensaje) : java.lang.String
+	getLongitudContenidoMensaje (java.lang.String mensaje) : int
+	cambiarUriRTSP (java.lang.String mensaje, UriRTSP url, UriRTSP urlSustituto) : java.lang.String
+	cambiarHost (java.lang.String mensaje, UriRTSP url, UriRTSP urlSustituto) : java.lang.String
-	getComando (java.lang.String mensaje) : java.lang.String
+	isDescribe (java.lang.String mensaje) : boolean
+	isOptions (java.lang.String mensaje) : boolean
+	isPause (java.lang.String mensaje) : boolean
+	isPlay (java.lang.String mensaje) : boolean
+	isSetup (java.lang.String mensaje) : boolean
+	isSetupTrack0 (java.lang.String mensaje) : boolean
+	isSetupTrack1 (java.lang.String mensaje) : boolean
+	isTearDown (java.lang.String mensaje) : boolean
+	isSetupRespuesta (java.lang.String mensaje) : boolean

Diagrama 3.11 - Diagrama de Clases APC Paquete Utilitarios

3.3.9 Diagramas de Secuencia APC

3.3.9.1.1 Reproducir Video

Diagrama 3.12 - Diagrama de Secuencia APC Reproducir Video

3.3.9.1.2 Reproducir Video (continuación)

Diagrama 3.13 -Continuación diagrama de Secuencia APC Reproducir Video

3.3.10 Diagrama de Actividades

3.3.10.1.1 Listener Mensaje RTSP



Diagrama 3.14 – Diagrama de Actividades Listener Mensaje RTSP

3.3.10.1.2 Listener Mensaje UDP



Diagrama 3.15- Diagrama de Actividades Listener Mensaje UDP

3.3.10.1.3 Listener Puerto RTSP

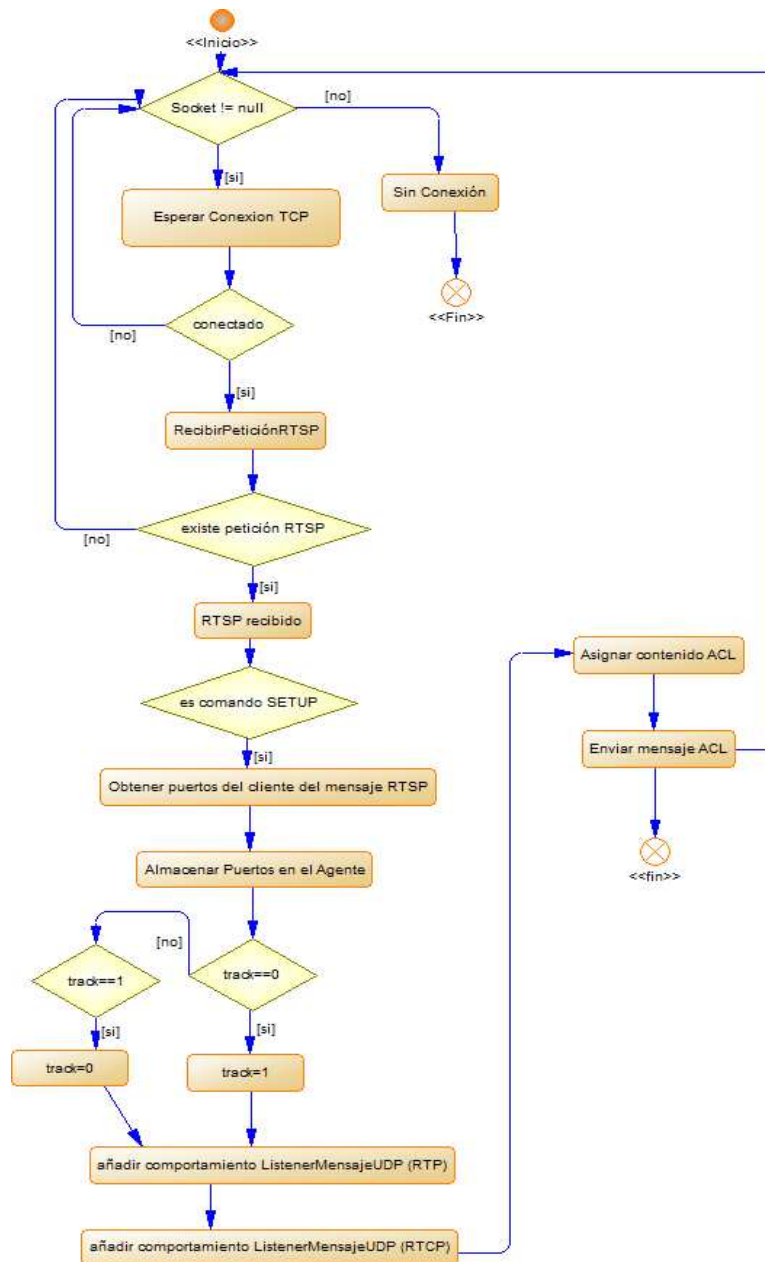


Diagrama 3.16 - Diagrama de Actividades Listener Puerto RTSP

3.3.10.1.4 Listener Puerto UDP



Diagrama 3.17 - Diagrama de Actividades Listener Puerto UDP

3.4 Codificación

Ver en anexo 3 Código de la Aplicación.

3.5 Pruebas

Para realizar las pruebas se ha utilizado los siguientes requerimientos:

- Una red inalámbrica con tecnología WiFi con un punto de acceso de 54Mbps.
- Un equipo portátil, procesador AMD de 2.20GHz, 4 GB de RAM y tarjeta inalámbrica Atheros AR5009 802.11 a/g/n Wifi Adapter; en el cual se ejecuta el Agente Proxy Servidor APS.
- Un equipo portátil, procesador Centrino Duo de 1.66GHz, 1 GB de RAM y tarjeta inalámbrica Intel Pro/Wireless 3945ABG.
- Un dispositivo móvil Android, modelo Nexus One, Procesador Qualcomm QSD 8250 1 GHz, 512MB RAM, Tarjeta microSD de 4GB expandible hasta 32 GB, Wi-Fi (802.11b/g/n); en el cual se ejecuta el Agente Proxy Cliente APC.

3.5.1 Reproducir video sin Proxy

Tabla 3.8 - Caso de Prueba 1

Caso de Prueba	
Número Caso de Prueba: 1	Número Historia de Usuario: 1
Nombre Caso de Prueba: Reproducir un video en el Nexus One	
Descripción: Reproducir un video del internet.	
Condiciones de ejecución: La dirección URL debe estar correcta y que el VLC se encuentre emitiendo un video.	
Entradas: URL del video.	

Resultado esperado: Se visualizará en el Nexus One una pantalla con una caja de texto donde el usuario pueda ingresar la dirección del video, al momento de poner play el video se reproducirá.

Evaluación: Satisfactoria.
El video se visualiza con una calidad del 100%



Ilustración 3.3 - Interfaz y Reproducción Video

3.5.2 Reconexión automática

3.5.2.1 Paso de imágenes con agente JADE

Tabla 3.9 - Caso de Prueba APS - APC 2

Caso de Prueba APS - APC	
Número Caso de Prueba: 2	Número Historia de Usuario: 2
Nombre Caso de Prueba: Paso de imágenes con agente JADE	
Descripción: Enviar desde el servidor al cliente una serie de imágenes, asegurando que lleguen en el mismo orden y sin ninguna pérdida después de una desconexión.	
Condiciones de ejecución: Debe existir un contenedor principal de agentes JADE y ejecutarse el agente APS, además que el VLC se encuentre emitiendo un video.	
Entradas: Imágenes.	
Resultado esperado: Se visualizará en el Nexus One un conjunto de imágenes cuyo contenido será un número el mismo que se irá incrementando, además se espera que las imágenes se pasen claramente y de manera rápida. Luego de la desconexión, se espera que la transmisión se realice de manera inmediata, automática	

y sin perder ninguna imagen en la transmisión.

Evaluación: Satisfactoria

Para realizar esta prueba se obtuvo un conjunto de imágenes cuyo contenido será un número el mismo que se incrementa, para poder observar claramente los resultados.

Ahora para emular una desconexión y reconexión durante la transmisión, se procedió a apagar el Access Point en la imagen del número 5 y se lo volvió a encender aproximadamente después de 5 minutos.

Los resultados fueron los esperados, debido a que luego de la reconexión, de manera inmediata y automática el Agente Proxy Servidor procedió a enviar las fotografías faltantes, es decir de la 6 a la 10, sin perder ninguna imagen en la transmisión.



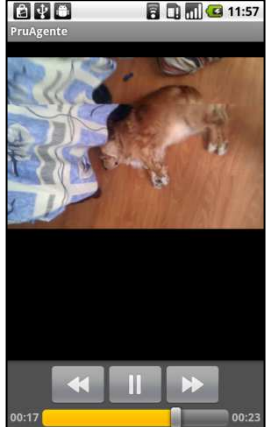
Ilustración 3.4 - Paso de Imágenes



Ilustración 3.5 – Reconexión

3.5.2.2 Paso de video con agente JADE


Tabla 3.10 - Caso de Prueba APS -APC 3

Caso de Prueba APS -APC	
Número Caso de Prueba: 3	Número Historia de Usuario: 2
Nombre Caso de Prueba: Paso de video con agente JADE	
Descripción: Enviar desde el servidor al cliente móvil el video, asegurando que no exista ninguna pérdida después de una desconexión y se reconecte de manera automática.	
Condiciones de ejecución: Debe existir un contenedor principal de agentes JADE y ejecutarse el agente APS, además que el VLC se encuentre emitiendo un video.	
Entradas: Video.	
Resultado esperado: Se visualizará en el Nexus One el video emitido, se espera que se vea con una calidad del 100%. Luego de la desconexión, se desea que la transmisión se realice de manera inmediata, automática y sin perder ninguna parte del video transmitido.	
<p>Evaluación: Satisfactoria.</p> <p>Para realizar esta prueba se emitió un video de tipo 3gp. Y el video se emitió con una calidad del 100%.</p> <p>Para emular una desconexión y reconexión durante la transmisión, se procedió a mover el móvil fuera del área de cobertura aproximadamente por 1 minuto.</p> <p>Los resultados fueron los esperados, debido a que luego de la reconexión, de manera inmediata y automática el Agente Proxy Servidor procedió a seguir descargando el video, sin perder ninguna parte.</p>	 <p>Ilustración 3.6 - Paso de Video con Jade</p>

3.5.2.3 Streaming Jade 0.9

3.5.2.3.1 Iteración 1


Tabla 3.11 - Caso de Prueba APS -APC 4

Caso de Prueba APS -APC	
Número Caso de Prueba: 4	Número Historia de Usuario: 2
Nombre Caso de Prueba: Streaming Jade Versión Beta.	
Descripción: Realizar una aplicación que emita video streaming y conste de: un servidor de video RTSP, un Agente Proxy Servidor – APS. Un cliente móvil Nexus One, que albergará un Agente Proxy Cliente - APC y a la aplicación RTSP Cliente PV Player que ya viene integrada al dispositivo Android.	
Condiciones de ejecución: Debe existir un contenedor principal de agentes JADE y ejecutarse el agente APS, además que el VLC se encuentre emitiendo un video.	
Entradas: URL del video.	
Resultado esperado: Visualizar en el Nexus One el video emitido, se espera que se vea con una calidad del 100%. Luego de la desconexión, se desea que la transmisión se realice de manera inmediata, automática y sin perder ninguna parte del video transmitido.	
Evaluación: Insatisfactoria. Para realizar esta prueba se emitió un video .avi. Al momento de visualizar el audio y el video en el Nexus One, estos están distorsionados completamente. Cuando se realiza la desconexión, los paquetes de video no regresan y los de audio regresan pero de una manera muy rápida sin entenderse nada. Con estos resultados no se obtiene lo esperado.	 <p>Ilustración 3.7- Jade y Rtsp</p>

3.5.2.4 Streaming Jade Versión 1

3.5.2.4.1 Iteración 2

Tabla 3.12- Caso de Prueba APS –APC 5

Caso de Prueba APS -APC	
Número Caso de Prueba: 5	Número Historia de Usuario: 2
Nombre Caso de Prueba: Streaming Jade Versión 1.	
<p>Descripción: Realizar una aplicación que emita video streaming y conste de: un servidor de video RTSP, un Agente Proxy Servidor – APS. Un cliente móvil Nexus One, que albergará un Agente Proxy Cliente - APC y a la aplicación RTSP Cliente PV Player que ya viene integrada al dispositivo Android. Mejorando la calidad tanto del video y del audio emitido de la versión anterior.</p>	
<p>Condiciones de ejecución: Debe existir un contenedor principal de agentes JADE y ejecutarse el agente APS, además que el VLC se encuentre emitiendo un video.</p>	
<p>Entradas: URL del video.</p>	
<p>Resultado esperado: Visualizar en el Nexus One el video emitido, se espera que se vea con una calidad del 100%. Luego de la desconexión, se desea que la transmisión se realice de manera inmediata, automática y sin perder ninguna parte del video transmitido.</p>	
<p>Evaluación: Satisfactoria.</p> <p>Luego de realizar los respectivos arreglos en el código, se realiza la prueba emitiendo un video .avi. Al momento de visualizar el audio y el video en el Nexus One han pasado correctamente con una calidad del 100%.</p> <p>Pero el momento de emular la desconexión por un tiempo mayor de 30 segundos por condiciones del reproductor y de los protocolos utilizados no se visualiza el video.</p>	 <p style="text-align: center;">Ilustración 3.8 - Interfaz y Video Emitido</p>

3.5.3 Creación de Agentes Jade de manera dinámica

Tabla 3.13 - Caso de Prueba 6 APS

Caso de Prueba APS	
Número Caso de Prueba: 6	Número Historia de Usuario: 2
Nombre Caso de Prueba: Creación de agentes de manera dinámica.	
Descripción: APS sea capaz de crear agentes según las necesidades, además que el usuario elija el número de agentes a crear.	
Condiciones de ejecución: Debe existir un contenedor principal de agentes JADE y que se encuentre en ejecución el APS.	
Entradas: Número de agentes a crear.	
Resultado esperado: Creación dinámica de los agentes en el número especificado.	
Evaluación: Satisfactoria. Se crean los agentes pedidos.	

3.5.4 Configurar Proxy

3.5.4.1 Streaming Jade Versión 1.1

Tabla 3.14- Caso de Prueba 7 APS –APC

Caso de Prueba APC	
Número Caso de Prueba: 7	Número Historia de Usuario: 2
Nombre Caso de Prueba: Streaming Jade Versión 1.1	
Descripción: Integrar un menú donde el usuario pueda activar / desactivar el proxy además ayudará a la configuración del puerto y de la dirección IP del contenedor JADE, también el puerto local que se utilizará para la comunicación RTSP.	
Condiciones de ejecución: Debe existir un contenedor principal de agentes JADE y ejecutarse	

el agente APS, además que el VLC se encuentre emitiendo un video.

Entradas: URL del video.

Resultado esperado: Visualizar el menú de configuraciones y reproducir el video emitido, según las configuraciones ingresadas.

Evaluación: Satisfactoria.

Luego de realizar los respectivos arreglos en el código, se realiza la prueba emitiendo un video .avi.

Se tiene ya el menú de opciones, el cliente ingresa los parámetros de configuración para el funcionamiento del proxy, y de este modo el video es visualizado con éxito.



Ilustración 3.9 - Interfaz y Menú configuración del Cliente

CAPITULO IV

4 IMPLANTACION Y PRUEBAS

4.1 Introducción

Antes de realizar las pruebas, verificamos el paso mensajes ACL entre los agentes Proxy, con el sniffer propio de la plataforma JADE, teniendo un resultado exitoso, ya que los mensajes se pasan de manera correcta, sin errores, como se puede ver en la siguiente figura:

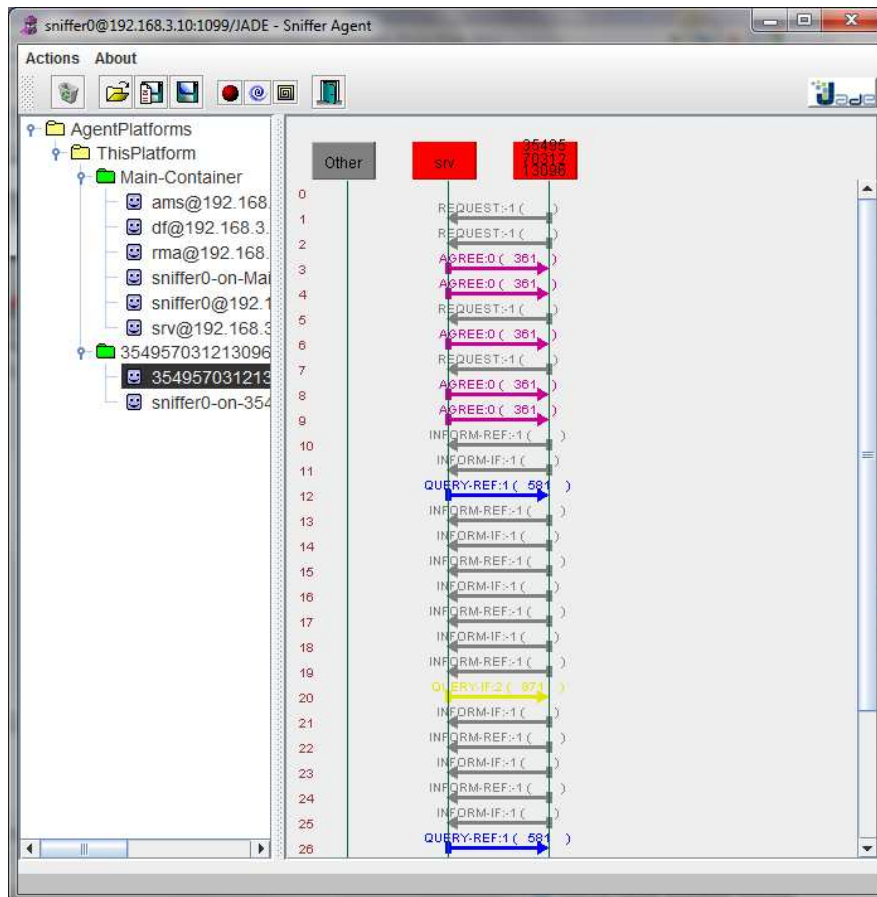


Ilustración 4.1 - Paso de mensajes ACL

Para todas las pruebas que se han realizado, se ha utilizado los equipos explicados en el anterior apartado, en el tema Pruebas. Además se ha instalado una herramienta sniffer

WireShark, que es la que facilitará obtener los paquetes de audio y video con toda su información como tiempo, bytes, timestamp, entre otros.

El sniffer estará ubicado en el Cliente Móvil Nexus One. Para que esta aplicación funcione deberá tener permisos de root.

El servidor RTSP que se ha usado es el VLC y se ha decidido utilizar la siguiente configuración para emitir el video:

Video

<i>Códec</i>	<i>Tasa de Bits</i>	<i>Fps</i> ⁷⁵	<i>Ancho</i>	<i>Alto</i>
H263	600 kb/s	25	176	144

Audio

<i>Códec</i>	<i>Tasa de Bits</i>	<i>Canales</i>	<i>Tasa de muestreo</i>
AAC ⁷⁶	128 kb/s	1	44100

Para poder observar las ventajas o desventajas del software desarrollado se ha tomado dos escenarios:

- **Escenario 1:** Jade-Streaming Versión 1.1.
- **Escenario 2:** Sin Proxy.

En ambos escenarios se ha emitido un clip de video (.avi), con una duración de 03:19 minutos, así como también se calculará una muestra de los paquetes de audio y otra de video, para cada software.

⁷⁵ Frames por segundo.

⁷⁶ Advanced Audio Coding.- es un formato informático de señal digital audio basado en el Algoritmo de compresión con pérdida.

Cálculo de la Muestra

Para calcular la muestra en los dos escenarios el único valor que cambia es N.

Fórmula a utilizar:

$$n = \frac{Z^2 p q N}{NE^2 + Z^2 p q}$$

Donde:

$$Z = 0,98$$

$$p = 0,50$$

$$q = 0,5$$

$$E = 0,02$$

Escenario 1 – Streaming-Jade Versión 1.1

Tabla 4.1 - Muestra Escenario 1

	N (# total de Paquetes Recibidos)	n(muestra)
Audio	8542	560
Video	12072	571

Escenario 2 –Sin Proxy

Tabla 4.2 - Muestra Escenario 2

	N (# total de Paquetes Recibidos)	n(muestra)
Audio	8368	560
Video	11823	571

4.2 Indicadores de rendimiento del servicio

4.2.1.1 Cálculo del Jitter y Retardo con los Paquetes de Video en el Escenario 1 y 2

Tabla 4.3 - Cálculo del Jitter y Retardo con los Paquetes de Video

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
1	1,2877E+12	36996	1777967621	0	0	0	53250	612637749	0	0
2	1,2877E+12	36997	1777971221	0,046	0,0025	0,037	53251	612641349	0,037	0,0001875
3	1,2877E+12	36998	1777974821	0,043	0,00271875	0,078	53252	612644949	0,041	0,00023828
4	1,2877E+12	36999	1777974821	0,011	0,00523633	0,117	53253	612648549	0,039	0,00028589
5	1,2877E+12	37000	1777974821	0,021	0,00559656	0,158	53254	612652149	0,041	0,00033052
6	1,2877E+12	37001	1777978421	0,003	0,00643427	0,171	53255	612652149	0,013	0,00112236
7	1,2877E+12	37002	1777982021	0,037	0,00834463	0,197	53256	612652149	0,026	0,00267722
8	1,2877E+12	37003	1777985621	0,044	0,00801059	0,197	53257	612655749	0	0,00500989
9	1,2877E+12	37004	1777989221	0,036	0,00775993	0,237	53258	612659349	0,04	0,00469677
10	1,2877E+12	37005	1777992821	0,042	0,00752493	0,277	53259	612662949	0,04	0,00440322
11	1,2877E+12	37006	1777996421	0,043	0,00717963	0,317	53260	612666549	0,04	0,00412802
12	1,2877E+12	37007	1778000021	0,039	0,0069184	0,357	53261	612670149	0,04	0,00387002
13	1,2877E+12	37008	1778000021	0,019	0,0089235	0,397	53262	612673749	0,04	0,00362814

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
14	1,2877E+12	37009	1778000021	0,007	0,00955328	0,438	53263	612677349	0,041	0,00346389
15	1,2877E+12	37010	1778003621	0,015	0,0110187	0,451	53264	612677349	0,013	0,00405989
16	1,2877E+12	37011	1778007221	0,042	0,01189253	0,464	53265	612677349	0,013	0,00461865
17	1,2877E+12	37012	1778010821	0,065	0,01127425	0,477	53266	612680949	0,013	0,00601748
18	1,2877E+12	37013	1778010821	0,023	0,01463211	0,517	53267	612684549	0,04	0,00564139
19	1,2877E+12	37014	1778014421	0,003	0,0147801	0,558	53268	612688149	0,041	0,0053513
20	1,2877E+12	37015	1778018021	0,018	0,01616884	0,578	53269	612688149	0,02	0,00626685
21	1,2877E+12	37016	1778021621	0,045	0,01653329	0,6	53270	612691749	0,022	0,00700017
22	1,2877E+12	37017	1778025221	0,038	0,01581246	0,637	53271	612695349	0,037	0,00675016
23	1,2877E+12	37018	1778028821	0,039	0,01494918	0,677	53272	612698949	0,04	0,00632827
24	1,2877E+12	37019	1778032421	0,043	0,01407736	0,717	53273	612702549	0,04	0,00593276
25	1,2877E+12	37020	1778036021	0,046	0,01338502	0,758	53274	612706149	0,041	0,00562446
26	1,2877E+12	37021	1778039621	0,033	0,01292346	0,798	53275	612709749	0,04	0,00527293
27	1,2877E+12	37022	1778043221	0,042	0,01255324	0,837	53276	612713349	0,039	0,00500587
28	1,2877E+12	37023	1778043221	0,022	0,01439367	0,877	53277	612716949	0,04	0,00469301
29	1,2877E+12	37024	1778043221	0,012	0,01486906	0,918	53278	612720549	0,041	0,00446219
30	1,2877E+12	37025	1778046821	0,024	0,01568975	0,931	53279	612720549	0,013	0,00499581

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
31	1,2877E+12	37026	1778050421	0,016	0,01570914	0,944	53280	612720549	0,013	0,00549607
32	1,2877E+12	37027	1778054021	0,042	0,01622732	0,957	53281	612724149	0,013	0,00684006
33	1,2877E+12	37028	1778057621	0,041	0,01533811	0,998	53282	612727749	0,041	0,00647506
34	1,2877E+12	37029	1778057621	0,107	0,01694198	1,037	53283	612731349	0,039	0,00613287
35	1,2877E+12	37030	1778061221	0,013	0,0200706	1,078	53284	612734949	0,041	0,00581206
36	1,2877E+12	37031	1778064821	0,012	0,02050369	1,098	53285	612734949	0,02	0,00669881
37	1,2877E+12	37032	1778068421	0,018	0,02097221	1,118	53286	612738549	0,02	0,00753013
38	1,2877E+12	37033	1778072021	0,016	0,02103645	1,157	53287	612742149	0,039	0,007122
39	1,2877E+12	37034	1778072021	0,009	0,02072167	1,198	53288	612745749	0,041	0,00673938
40	1,2877E+12	37035	1778072021	0,011	0,01998906	1,238	53289	612749349	0,04	0,00631816
41	1,2877E+12	37036	1778072021	0,009	0,01942725	1,248	53290	612749349	0,01	0,00654828
42	1,2877E+12	37037	1778075621	0,012	0,02015054	1,258	53291	612749349	0,01	0,00676401
43	1,2877E+12	37038	1778079221	0,035	0,02064114	1,27	53292	612749349	0,012	0,00709126
44	1,2877E+12	37039	1778079221	0,016	0,02153856	1,278	53293	612752949	0,008	0,00864806
45	1,2877E+12	37040	1778082821	0,025	0,0216924	1,318	53294	612756549	0,04	0,00810755
46	1,2877E+12	37041	1778082821	0,018	0,02189913	1,338	53295	612756549	0,02	0,00885083
47	1,2877E+12	37042	1778086421	0,021	0,02190543	1,359	53296	612760149	0,021	0,00948515

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
48	1,2877E+12	37043	1778086421	0,006	0,02184884	1,378	53297	612760149	0,019	0,01007983
49	1,2877E+12	37044	1778086421	0,008	0,02085829	1,398	53298	612763749	0,02	0,01069984
50	1,2877E+12	37045	1778086421	0,01	0,02005465	1,406	53299	612763749	0,008	0,0105311
51	1,2877E+12	37046	1778086421	0,006	0,01942623	1,414	53300	612763749	0,008	0,01037291
52	1,2877E+12	37047	1778090021	0,007	0,02033709	1,422	53301	612763749	0,008	0,0102246
53	1,2877E+12	37048	1778093621	0,039	0,02112852	1,43	53302	612763749	0,008	0,01008556
54	1,2877E+12	37049	1778093621	0,025	0,02224549	1,438	53303	612767349	0,008	0,01145522
55	1,2877E+12	37050	1778097221	0,03	0,02179265	1,478	53304	612770949	0,04	0,01073927
56	1,2877E+12	37051	1778097221	0,021	0,02230561	1,497	53305	612770949	0,019	0,01125556
57	1,2877E+12	37052	1778100821	0,043	0,02209901	1,518	53306	612774549	0,021	0,01173959
58	1,2877E+12	37053	1778100821	0,028	0,02340532	1,538	53307	612774549	0,02	0,01225586
59	1,2877E+12	37054	1778104421	0,019	0,02269249	1,558	53308	612778149	0,02	0,01273987
60	1,2877E+12	37055	1778104421	0,022	0,02246171	1,58	53309	612778149	0,022	0,01331863
61	1,2877E+12	37056	1778108021	0,029	0,02218285	1,6	53310	612781749	0,02	0,01373622
62	1,2877E+12	37057	1778108021	0,071	0,02260892	1,617	53311	612781749	0,017	0,0139402
63	1,2877E+12	37058	1778111621	0,028	0,02313336	1,638	53312	612785349	0,021	0,01425644
64	1,2877E+12	37059	1778111621	0,022	0,02343753	1,658	53313	612785349	0,02	0,01461541

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
65	1,2877E+12	37060	1778115221	0,01	0,02309768	1,68	53314	612788949	0,022	0,01482695
66	1,2877E+12	37061	1778115221	0,021	0,02227908	1,697	53315	612788949	0,017	0,01496277
67	1,2877E+12	37062	1778118821	0,023	0,02207414	1,718	53316	612792549	0,021	0,01521509
68	1,2877E+12	37063	1778118821	0,018	0,022132	1,74	53317	612792549	0,022	0,01563915
69	1,2877E+12	37064	1778122421	0,042	0,02212375	1,758	53318	612796149	0,018	0,0160367
70	1,2877E+12	37065	1778122421	0,005	0,02336602	1,777	53319	612796149	0,019	0,01622191
71	1,2877E+12	37066	1778126021	0,005	0,02409314	1,799	53320	612799749	0,022	0,01633304
72	1,2877E+12	37067	1778126021	0,002	0,02289982	1,818	53321	612799749	0,019	0,01649972
73	1,2877E+12	37068	1778129621	0,013	0,02384358	1,838	53322	612803349	0,02	0,01671849
74	1,2877E+12	37069	1778129621	0,021	0,02316586	1,858	53323	612803349	0,02	0,01692359
75	1,2877E+12	37070	1778129621	0,007	0,02303049	1,878	53324	612806949	0,02	0,01711586
76	1,2877E+12	37071	1778129621	0,014	0,02202859	1,886	53325	612806949	0,008	0,01654612
77	1,2877E+12	37072	1778129621	0,031	0,0215268	1,894	53326	612806949	0,008	0,01601199
78	1,2877E+12	37073	1778133221	0,007	0,02074387	1,902	53327	612806949	0,008	0,01551124
79	1,2877E+12	37074	1778136821	0,094	0,02150988	1,91	53328	612806949	0,008	0,01504179
80	1,2877E+12	37075	1778136821	0,008	0,02604051	1,918	53329	612810549	0,008	0,01610167
81	1,2877E+12	37076	1778140421	0,012	0,02641298	1,958	53330	612814149	0,04	0,01509532

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
82	1,2877E+12	37077	1778140421	0,006	0,02551217	1,977	53331	612814149	0,019	0,01533936
83	1,2877E+12	37078	1778144021	0,009	0,02604266	1,999	53332	612817749	0,022	0,01550565
84	1,2877E+12	37079	1778144021	0,007	0,02497749	2,019	53333	612817749	0,02	0,01578655
85	1,2877E+12	37080	1778147621	0,009	0,0254789	2,038	53334	612821349	0,019	0,01611239
86	1,2877E+12	37081	1778147621	0,01	0,02444897	2,058	53335	612821349	0,02	0,01635537
87	1,2877E+12	37082	1778151221	0,01	0,02479591	2,078	53336	612824949	0,02	0,01658315
88	1,2877E+12	37083	1778151221	0,009	0,02387116	2,098	53337	612824949	0,02	0,01679671
89	1,2877E+12	37084	1778154821	0,014	0,02431672	2,118	53338	612828549	0,02	0,01699691
90	1,2877E+12	37085	1778154821	0,029	0,02367192	2,137	53339	612828549	0,019	0,01712211
91	1,2877E+12	37086	1778158421	0,014	0,02287993	2,158	53340	612832149	0,021	0,01723947
92	1,2877E+12	37087	1778158421	0,011	0,02232493	2,178	53341	612832149	0,02	0,01741201
93	1,2877E+12	37088	1778162021	0,01	0,02274212	2,201	53342	612835749	0,023	0,01738626
94	1,2877E+12	37089	1778162021	0,017	0,02194574	2,218	53343	612835749	0,017	0,01736212
95	1,2877E+12	37090	1778165621	0,011	0,02201163	2,238	53344	612839349	0,02	0,01752698
96	1,2877E+12	37091	1778165621	0,007	0,0213234	2,258	53345	612839349	0,02	0,01768155
97	1,2877E+12	37092	1778169221	0,011	0,02205319	2,278	53346	612842949	0,02	0,01782645
98	1,2877E+12	37093	1778169221	0,011	0,02136237	2,298	53347	612842949	0,02	0,0179623

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
99	1,2877E+12	37094	1778172821	0,015	0,02183972	2,318	53348	612846549	0,02	0,01808965
100	1,2877E+12	37095	1778172821	0,009	0,02141224	2,338	53349	612846549	0,02	0,01820905
101	1,2877E+12	37096	1778172821	0,011	0,02063647	2,358	53350	612850149	0,02	0,01832098
102	1,2877E+12	37097	1778172821	0,006	0,02003419	2,367	53351	612850149	0,009	0,01773842
103	1,2877E+12	37098	1778172821	0,007	0,01915706	2,374	53352	612850149	0,007	0,01706727
104	1,2877E+12	37099	1778176421	0,014	0,02002224	2,382	53353	612850149	0,008	0,01650057
105	1,2877E+12	37100	1778176421	0,023	0,01964585	2,39	53354	612850149	0,008	0,01596928
106	1,2877E+12	37101	1778180021	0,011	0,01948048	2,398	53355	612853749	0,008	0,0169712
107	1,2877E+12	37102	1778180021	0,02	0,01895045	2,418	53356	612853749	0,02	0,0171605
108	1,2877E+12	37103	1778183621	0,021	0,01901605	2,447	53357	612857349	0,029	0,01677547
109	1,2877E+12	37104	1778183621	0,045	0,01914005	2,464	53358	612857349	0,017	0,0167895
110	1,2877E+12	37105	1778187221	0,015	0,01825629	2,478	53359	612860949	0,014	0,01736516
111	1,2877E+12	37106	1778187221	0,014	0,01805278	2,499	53360	612860949	0,021	0,01759234
112	1,2877E+12	37107	1778190821	0,015	0,01854948	2,518	53361	612864549	0,019	0,01780532
113	1,2877E+12	37108	1778190821	0,023	0,01832764	2,538	53362	612864549	0,02	0,01794248
114	1,2877E+12	37109	1778194421	0,015	0,01824466	2,558	53363	612868149	0,02	0,01807108
115	1,2877E+12	37110	1778194421	0,013	0,01804187	2,584	53364	612868149	0,026	0,01856664

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
116	1,2877E+12	37111	1778198021	0,03	0,01860175	2,601	53365	612871749	0,017	0,01884372
117	1,2877E+12	37112	1778198021	0,013	0,01931414	2,618	53366	612871749	0,017	0,01872849
118	1,2877E+12	37113	1778201621	0,026	0,01979451	2,647	53367	612875349	0,029	0,01824546
119	1,2877E+12	37114	1778201621	0,019	0,02018235	2,658	53368	612875349	0,011	0,01779262
120	1,2877E+12	37115	1778205221	0,02	0,02023345	2,678	53369	612878949	0,02	0,01793058
121	1,2877E+12	37116	1778205221	0,01	0,02021886	2,698	53370	612878949	0,02	0,01805992
122	1,2877E+12	37117	1778208821	0,036	0,02083018	2,718	53371	612882549	0,02	0,01818117
123	1,2877E+12	37118	1778208821	0,018	0,0217783	2,738	53372	612882549	0,02	0,01829485
124	1,2877E+12	37119	1778212421	0,063	0,02179215	2,758	53373	612886149	0,02	0,01840142
125	1,2877E+12	37120	1778212421	0,007	0,02436764	2,778	53374	612886149	0,02	0,01850133
126	1,2877E+12	37121	1778216021	0,016	0,02490717	2,798	53375	612889749	0,02	0,018595
127	1,2877E+12	37122	1778216021	0,005	0,02435047	2,823	53376	612889749	0,025	0,01899531
128	1,2877E+12	37123	1778216021	0,007	0,02314106	2,838	53377	612893349	0,015	0,0193706
129	1,2877E+12	37124	1778216021	0,009	0,02213225	2,846	53378	612893349	0,008	0,01865994
130	1,2877E+12	37125	1778216021	0,007	0,02131148	2,854	53379	612893349	0,008	0,0179937
131	1,2877E+12	37126	1778219621	0,022	0,02204201	2,862	53380	612893349	0,008	0,01736909
132	1,2877E+12	37127	1778223221	0,033	0,02178939	2,87	53381	612893349	0,008	0,01678352

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
133	1,2877E+12	37128	1778223221	0,018	0,02249005	2,878	53382	612896949	0,008	0,01773455
134	1,2877E+12	37129	1778226821	0,012	0,02245942	2,918	53383	612900549	0,04	0,01662614
135	1,2877E+12	37130	1778226821	0,011	0,02180571	2,937	53384	612900549	0,019	0,01677451
136	1,2877E+12	37131	1778230421	0,011	0,02225535	2,958	53385	612904149	0,021	0,0169136
137	1,2877E+12	37132	1778230421	0,009	0,02155189	2,977	53386	612904149	0,019	0,017044
138	1,2877E+12	37133	1778234021	0,031	0,0221424	3	53387	612907749	0,023	0,01704125
139	1,2877E+12	37134	1778234021	0,01	0,022696	3,018	53388	612907749	0,018	0,01710117
140	1,2877E+12	37135	1778237621	0,049	0,0231525	3,038	53389	612911349	0,02	0,01728235
141	1,2877E+12	37136	1778237621	0,054	0,02476797	3,058	53390	612911349	0,02	0,0174522
142	1,2877E+12	37137	1778241221	0,015	0,02409497	3,081	53391	612914949	0,023	0,01742394
143	1,2877E+12	37138	1778241221	0,033	0,02352653	3,098	53392	612914949	0,017	0,01739744
144	1,2877E+12	37139	1778244821	0,029	0,02249363	3,119	53393	612918549	0,021	0,0174976
145	1,2877E+12	37140	1778244821	0,002	0,02290027	3,138	53394	612918549	0,019	0,0175915
146	1,2877E+12	37141	1778248421	0,001	0,02384401	3,158	53395	612922149	0,02	0,01774203
147	1,2877E+12	37142	1778248421	0,002	0,02241626	3,178	53396	612922149	0,02	0,01788316
148	1,2877E+12	37143	1778252021	0,004	0,02339024	3,199	53397	612925749	0,021	0,01795296
149	1,2877E+12	37144	1778252021	0,006	0,02217835	3,218	53398	612925749	0,019	0,0180184

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
150	1,2877E+12	37145	1778255621	0,025	0,0229172	3,238	53399	612929349	0,02	0,01814225
151	1,2877E+12	37146	1778255621	0,017	0,02304738	3,258	53400	612929349	0,02	0,01825836
152	1,2877E+12	37147	1778259221	0,019	0,02304442	3,278	53401	612932949	0,02	0,01836721
153	1,2877E+12	37148	1778259221	0,029	0,02279164	3,298	53402	612932949	0,02	0,01846926
154	1,2877E+12	37149	1778259221	0,007	0,02317966	3,318	53403	612936549	0,02	0,01856493
155	1,2877E+12	37150	1778259221	0,007	0,02216843	3,326	53404	612936549	0,008	0,01790462
156	1,2877E+12	37151	1778259221	0,018	0,02122041	3,334	53405	612936549	0,008	0,01728559
157	1,2877E+12	37152	1778262821	0,033	0,02126913	3,342	53406	612936549	0,008	0,01670524
158	1,2877E+12	37153	1778262821	0,037	0,02200231	3,35	53407	612936549	0,008	0,01616116
159	1,2877E+12	37154	1778266421	0,014	0,02081467	3,358	53408	612940149	0,008	0,01715109
160	1,2877E+12	37155	1778266421	0,012	0,02038875	3,378	53409	612940149	0,02	0,01732914
161	1,2877E+12	37156	1778270021	0,015	0,02086445	3,398	53410	612943749	0,02	0,01749607
162	1,2877E+12	37157	1778270021	0,007	0,02049793	3,421	53411	612943749	0,023	0,01784007
163	1,2877E+12	37158	1778273621	0,019	0,0212793	3,438	53412	612947349	0,017	0,01816256
164	1,2877E+12	37159	1778273621	0,012	0,02113685	3,458	53413	612947349	0,02	0,0182774
165	1,2877E+12	37160	1778277221	0,008	0,0215658	3,478	53414	612950949	0,02	0,01838507
166	1,2877E+12	37161	1778277221	0,016	0,02071793	3,498	53415	612950949	0,02	0,018486

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
167	1,2877E+12	37162	1778280821	0,009	0,02092306	3,518	53416	612954549	0,02	0,01858062
168	1,2877E+12	37163	1778280821	0,072	0,02017787	3,54	53417	612954549	0,022	0,01879433
169	1,2877E+12	37164	1778284421	0,009	0,02091675	3,559	53418	612958149	0,019	0,01893219
170	1,2877E+12	37165	1778284421	0,01	0,02017196	3,578	53419	612958149	0,019	0,01893643
171	1,2877E+12	37166	1778288021	0,008	0,02078621	3,598	53420	612961749	0,02	0,0190029
172	1,2877E+12	37167	1778288021	0,007	0,01998707	3,618	53421	612961749	0,02	0,01906522
173	1,2877E+12	37168	1778291621	0,011	0,02080038	3,638	53422	612965349	0,02	0,01912364
174	1,2877E+12	37169	1778291621	0,019	0,02018786	3,658	53423	612965349	0,02	0,01917842
175	1,2877E+12	37170	1778295221	0,029	0,02023861	3,678	53424	612968949	0,02	0,01922976
176	1,2877E+12	37171	1778295221	0,01	0,0207862	3,698	53425	612968949	0,02	0,0192779
177	1,2877E+12	37172	1778298821	0,02	0,02136206	3,718	53426	612972549	0,02	0,01932304
178	1,2877E+12	37173	1778298821	0,025	0,02127693	3,738	53427	612972549	0,02	0,01936535
179	1,2877E+12	37174	1778302421	0,017	0,02088463	3,758	53428	612976149	0,02	0,01940501
180	1,2877E+12	37175	1778302421	0,008	0,02064184	3,778	53429	612976149	0,02	0,0194422
181	1,2877E+12	37176	1778302421	0,022	0,01985172	3,798	53430	612979749	0,02	0,01947706
182	1,2877E+12	37177	1778302421	0,008	0,01998599	3,806	53431	612979749	0,008	0,01875974
183	1,2877E+12	37178	1778302421	0,006	0,01923687	3,814	53432	612979749	0,008	0,01808726

184	1,2877E+12	37179	1778306021	0,015	0,02015956	3,822	53433	612979749	0,008	0,01745681
185	1,2877E+12	37180	1778306021	0,011	0,01983709	3,83	53434	612979749	0,008	0,01686576
186	1,2877E+12	37181	1778309621	0,016	0,02040977	3,838	53435	612983349	0,008	0,01781165
187	1,2877E+12	37182	1778313221	0,034	0,02063416	3,858	53436	612983349	0,02	0,01794842
188	1,2877E+12	37183	1778313221	0,022	0,02146952	3,878	53437	612986949	0,02	0,01807664
189	1,2877E+12	37184	1778316821	0,018	0,02125268	3,919	53438	612990549	0,041	0,01700935
190	1,2877E+12	37185	1778316821	0,025	0,02104939	3,938	53439	612990549	0,019	0,01713377
191	1,2877E+12	37186	1778320421	0,019	0,0206713	3,958	53440	612994149	0,02	0,01731291
192	1,2877E+12	37187	1778320421	0,024	0,02056684	3,978	53441	612994149	0,02	0,01748085
193	1,2877E+12	37188	1778324021	0,033	0,02028142	3,998	53442	612997749	0,02	0,0176383
194	1,2877E+12	37189	1778324021	0,031	0,02107633	4,018	53443	612997749	0,02	0,0177859
195	1,2877E+12	37190	1778327621	0,054	0,02032156	4,038	53444	613001349	0,02	0,01792428
196	1,2877E+12	37191	1778327621	0,006	0,02242646	4,058	53445	613001349	0,02	0,01805402
197	1,2877E+12	37192	1778331221	0,01	0,02314981	4,078	53446	613004949	0,02	0,01817564
198	1,2877E+12	37193	1778331221	0,008	0,02232794	4,098	53447	613004949	0,02	0,01828966
199	1,2877E+12	37194	1778334821	0,004	0,02293245	4,118	53448	613008549	0,02	0,01839656
200	1,2877E+12	37195	1778334821	0,003	0,02174917	4,138	53449	613008549	0,02	0,01849677
201	1,2877E+12	37196	1778338421	0,041	0,02270235	4,158	53450	613012149	0,02	0,01859073
202	1,2877E+12	37197	1778338421	0,002	0,02384595	4,179	53451	613012149	0,021	0,01874131

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
203	1,2877E+12	37198	1778342021	0,018	0,02473058	4,199	53452	613015749	0,02	0,01881997
204	1,2877E+12	37199	1778342021	0,009	0,02430992	4,218	53453	613015749	0,019	0,01883123
205	1,2877E+12	37200	1778342021	0,008	0,02335305	4,24	53454	613019349	0,022	0,01877927
206	1,2877E+12	37201	1778342021	0,007	0,02239348	4,246	53455	613019349	0,006	0,01798057
207	1,2877E+12	37202	1778342021	0,007	0,02143139	4,254	53456	613019349	0,008	0,01735678
208	1,2877E+12	37203	1778345621	0,009	0,02215443	4,262	53457	613019349	0,008	0,01677198
209	1,2877E+12	37204	1778345621	0,008	0,02133228	4,27	53458	613019349	0,008	0,01622374
210	1,2877E+12	37205	1778345621	0,008	0,02049901	4,278	53459	613022949	0,008	0,01720975
211	1,2877E+12	37206	1778345621	0,009	0,01971782	4,286	53460	613022949	0,008	0,01663414
212	1,2877E+12	37207	1778345621	0,007	0,01904796	4,294	53461	613022949	0,008	0,01609451
213	1,2877E+12	37208	1778349221	0,01	0,01991996	4,302	53462	613022949	0,008	0,0155886
214	1,2877E+12	37209	1778349221	0,047	0,01929996	4,31	53463	613022949	0,008	0,01511431
215	1,2877E+12	37210	1778352821	0,123	0,01853121	4,318	53464	613026549	0,008	0,01616967
216	1,2877E+12	37211	1778352821	0,079	0,02506051	4,338	53465	613026549	0,02	0,01640907
217	1,2877E+12	37212	1778356421	0,02	0,02593173	4,358	53466	613030149	0,02	0,0166335
218	1,2877E+12	37213	1778356421	0,015	0,025561	4,378	53467	613030149	0,02	0,01684391
219	1,2877E+12	37214	1778360021	0,017	0,02552594	4,4	53468	613033749	0,022	0,01691616

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
220	1,2877E+12	37215	1778360021	0,002	0,02499306	4,418	53469	613033749	0,018	0,0169839
221	1,2877E+12	37216	1778363621	0,002	0,025806	4,438	53470	613037349	0,02	0,01717241
222	1,2877E+12	37217	1778363621	0,003	0,02431812	4,458	53471	613037349	0,02	0,01734913
223	1,2877E+12	37218	1778367221	0,002	0,02511074	4,478	53472	613040949	0,02	0,01751481
224	1,2877E+12	37219	1778367221	0,002	0,02366632	4,498	53473	613040949	0,02	0,01767014
225	1,2877E+12	37220	1778370821	0,003	0,02456217	4,518	53474	613044549	0,02	0,01781575
226	1,2877E+12	37221	1778370821	0,006	0,02321454	4,538	53475	613044549	0,02	0,01795227
227	1,2877E+12	37222	1778374421	0,007	0,02388863	4,558	53476	613048149	0,02	0,01808025
228	1,2877E+12	37223	1778374421	0,003	0,02283309	4,578	53477	613048149	0,02	0,01820024
229	1,2877E+12	37224	1778378021	0,001	0,02371852	4,598	53478	613051749	0,02	0,01831272
230	1,2877E+12	37225	1778378021	0,005	0,02229861	4,621	53479	613051749	0,023	0,01860568
231	1,2877E+12	37226	1778381621	0,022	0,02309245	4,638	53480	613055349	0,017	0,01888032
232	1,2877E+12	37227	1778381621	0,021	0,02302417	4,658	53481	613055349	0,02	0,0189503
233	1,2877E+12	37228	1778385221	0,025	0,02277266	4,678	53482	613058949	0,02	0,01901591
234	1,2877E+12	37229	1778385221	0,014	0,02291187	4,698	53483	613058949	0,02	0,01907741
235	1,2877E+12	37230	1778388821	0,03	0,02310488	4,718	53484	613062549	0,02	0,01913507
236	1,2877E+12	37231	1778388821	0,035	0,02353582	4,738	53485	613062549	0,02	0,01918913

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
237	1,2877E+12	37232	1778388821	0,009	0,02425233	4,758	53486	613066149	0,02	0,01923981
238	1,2877E+12	37233	1778388821	0,001	0,02329906	4,766	53487	613066149	0,008	0,01853732
239	1,2877E+12	37234	1778388821	0,01	0,02190537	4,775	53488	613066149	0,009	0,01794124
240	1,2877E+12	37235	1778392421	0,005	0,02241129	4,782	53489	613066149	0,007	0,01725741
241	1,2877E+12	37236	1778392421	0,015	0,02132308	4,79	53490	613066149	0,008	0,01667882
242	1,2877E+12	37237	1778396021	0,009	0,02155289	4,799	53491	613069749	0,009	0,0175739
243	1,2877E+12	37238	1778399621	0,032	0,02214333	4,818	53492	613069749	0,019	0,01766303
244	1,2877E+12	37239	1778399621	0,015	0,02275937	4,842	53493	613073349	0,024	0,01755909
245	1,2877E+12	37240	1778403221	0,027	0,02289941	4,878	53494	613076949	0,036	0,01671165
246	1,2877E+12	37241	1778403221	0,011	0,0231557	4,9	53495	613076949	0,022	0,01704217
247	1,2877E+12	37242	1778406821	0,026	0,02352097	4,918	53496	613080549	0,018	0,01735203
248	1,2877E+12	37243	1778406821	0,014	0,02367591	4,938	53497	613080549	0,02	0,01751753
249	1,2877E+12	37244	1778410421	0,03	0,02382116	4,958	53498	613084149	0,02	0,01767269
250	1,2877E+12	37245	1778410421	0,016	0,02420734	4,978	53499	613084149	0,02	0,01781814
251	1,2877E+12	37246	1778414021	0,045	0,02419438	4,999	53500	613087749	0,021	0,01789201
252	1,2877E+12	37247	1778414021	0,038	0,02549473	5,017	53501	613087749	0,018	0,01789876
253	1,2877E+12	37248	1778417621	0,011	0,02402631	5,038	53502	613091349	0,021	0,01796759

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
254	1,2877E+12	37249	1778417621	0,009	0,02321217	5,059	53503	613091349	0,021	0,01815711
255	1,2877E+12	37250	1778421221	0,01	0,02369891	5,078	53504	613094949	0,019	0,01833479
256	1,2877E+12	37251	1778421221	0,007	0,02284273	5,098	53505	613094949	0,02	0,01843887
257	1,2877E+12	37252	1778424821	0,02	0,02347756	5,118	53506	613098549	0,02	0,01853644
258	1,2877E+12	37253	1778424821	0,013	0,02326021	5,138	53507	613098549	0,02	0,01862791
259	1,2877E+12	37254	1778428421	0,097	0,02349395	5,158	53508	613102149	0,02	0,01871367
260	1,2877E+12	37255	1778428421	0,012	0,02808807	5,178	53509	613102149	0,02	0,01879406
261	1,2877E+12	37256	1778432021	0,009	0,02808257	5,2	53510	613105749	0,022	0,01874443
262	1,2877E+12	37257	1778432021	0,003	0,02688991	5,218	53511	613105749	0,018	0,01869791
263	1,2877E+12	37258	1778432021	0,008	0,02539679	5,238	53512	613109349	0,02	0,01877929
264	1,2877E+12	37259	1778432021	0,005	0,02430949	5,246	53513	613109349	0,008	0,01810558
265	1,2877E+12	37260	1778432021	0,005	0,02310265	5,254	53514	613109349	0,008	0,01747398
266	1,2877E+12	37261	1778435621	0,021	0,02384623	5,264	53515	613109349	0,01	0,01700686
267	1,2877E+12	37262	1778435621	0,004	0,02366834	5,27	53516	613109349	0,006	0,01631893
268	1,2877E+12	37263	1778439221	0,003	0,02443907	5,278	53517	613112949	0,008	0,017299
269	1,2877E+12	37264	1778439221	0,008	0,02309913	5,299	53518	613112949	0,021	0,01753031
270	1,2877E+12	37265	1778442821	0,012	0,02365543	5,318	53519	613116549	0,019	0,01774717

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
271	1,2877E+12	37266	1778442821	0,013	0,02292697	5,338	53520	613116549	0,02	0,01788797
272	1,2877E+12	37267	1778446421	0,038	0,02318153	5,358	53521	613120149	0,02	0,01801997
273	1,2877E+12	37268	1778446421	0,01	0,02410769	5,378	53522	613120149	0,02	0,01814372
274	1,2877E+12	37269	1778450021	0,015	0,02447596	5,401	53523	613123749	0,023	0,01807224
275	1,2877E+12	37270	1778450021	0,022	0,02388371	5,418	53524	613123749	0,017	0,01800522
276	1,2877E+12	37271	1778453621	0,026	0,02351598	5,438	53525	613127349	0,02	0,0181299
277	1,2877E+12	37272	1778453621	0,012	0,02367123	5,458	53526	613127349	0,02	0,01824678
278	1,2877E+12	37273	1778457221	0,025	0,02394178	5,48	53527	613130949	0,022	0,01823136
279	1,2877E+12	37274	1778457221	0,014	0,02400792	5,498	53528	613130949	0,018	0,0182169
280	1,2877E+12	37275	1778460821	0,041	0,02413242	5,518	53529	613134549	0,02	0,01832834
281	1,2877E+12	37276	1778460821	0,006	0,02518665	5,538	53530	613134549	0,02	0,01843282
282	1,2877E+12	37277	1778464421	0,038	0,02573748	5,558	53531	613138149	0,02	0,01853077
283	1,2877E+12	37278	1778464421	0,005	0,02650389	5,578	53532	613138149	0,02	0,01862259
284	1,2877E+12	37279	1778468021	0,015	0,02703489	5,599	53533	613141749	0,021	0,01864618
285	1,2877E+12	37280	1778471621	0,043	0,02690771	5,618	53534	613141749	0,019	0,0186683
286	1,2877E+12	37281	1778475221	0,043	0,02541348	5,638	53535	613145349	0,02	0,01875153
287	1,2877E+12	37282	1778475221	0,009	0,02651264	5,678	53536	613148949	0,04	0,01757956

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
288	1,2877E+12	37283	1778475221	0,006	0,0254181	5,718	53537	613152549	0,04	0,01648083
289	1,2877E+12	37284	1778475221	0,012	0,02420447	5,726	53538	613152549	0,008	0,01595078
290	1,2877E+12	37285	1778475221	0,007	0,02344169	5,734	53539	613152549	0,008	0,01545386
291	1,2877E+12	37286	1778478821	0,006	0,02403908	5,742	53540	613152549	0,008	0,01498799
292	1,2877E+12	37287	1778482421	0,03	0,02466164	5,75	53541	613152549	0,008	0,01455124
293	1,2877E+12	37288	1778482421	0,017	0,02499529	5,758	53542	613156149	0,008	0,01564179
294	1,2877E+12	37289	1778486021	0,023	0,02487058	5,798	53543	613159749	0,04	0,01466418
295	1,2877E+12	37290	1778489621	0,069	0,02437867	5,818	53544	613159749	0,02	0,01499767
296	1,2877E+12	37291	1778493221	0,021	0,0246675	5,838	53545	613163349	0,02	0,01531031
297	1,2877E+12	37292	1778496821	0,029	0,02431328	5,879	53546	613166949	0,041	0,01441592
298	1,2877E+12	37293	1778500421	0,045	0,0234812	5,919	53547	613170549	0,04	0,01351492
299	1,2877E+12	37294	1778504021	0,043	0,02232613	5,958	53548	613174149	0,039	0,01273274
300	1,2877E+12	37295	1778507621	0,041	0,02111825	5,999	53549	613177749	0,041	0,01199944
301	1,2877E+12	37296	1778511221	0,096	0,01986086	6,038	53550	613181349	0,039	0,01131198
302	1,2877E+12	37297	1778514821	0,016	0,02211955	6,078	53551	613184949	0,04	0,01060498
303	1,2877E+12	37298	1778518421	0,009	0,02223708	6,119	53552	613188549	0,041	0,01000467
304	1,2877E+12	37299	1778518421	0,009	0,02140976	6,158	53553	613192149	0,039	0,00944188

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
305	1,2877E+12	37300	1778518421	0,004	0,02063415	6,197	53554	613195749	0,039	0,00891426
306	1,2877E+12	37301	1778518421	0,005	0,01959452	6,207	53555	613195749	0,01	0,00898212
307	1,2877E+12	37302	1778518421	0,008	0,01868236	6,213	53556	613195749	0,006	0,00879574
308	1,2877E+12	37303	1778522021	0,008	0,01951471	6,221	53557	613195749	0,008	0,008746
309	1,2877E+12	37304	1778525621	0,042	0,02029504	6,229	53558	613195749	0,008	0,00869938
310	1,2877E+12	37305	1778529221	0,043	0,0191516	6,237	53559	613199349	0,008	0,01015567
311	1,2877E+12	37306	1778532821	0,038	0,01814213	6,277	53560	613202949	0,04	0,00952094
312	1,2877E+12	37307	1778536421	0,036	0,01713325	6,327	53561	613206549	0,05	0,00955088
313	1,2877E+12	37308	1778540021	0,04	0,01631242	6,357	53562	613210149	0,03	0,00957895
314	1,2877E+12	37309	1778543621	0,041	0,01529289	6,398	53563	613213749	0,041	0,00904276
315	1,2877E+12	37310	1778543621	0,016	0,01689959	6,437	53564	613217349	0,039	0,00854009
316	1,2877E+12	37311	1778547221	0,028	0,01734336	6,477	53565	613220949	0,04	0,00800634
317	1,2877E+12	37312	1778550821	0,038	0,0170094	6,497	53566	613220949	0,02	0,00875594
318	1,2877E+12	37313	1778554421	0,046	0,01607131	6,517	53567	613224549	0,02	0,00945869
319	1,2877E+12	37314	1778558021	0,037	0,01544186	6,558	53568	613228149	0,041	0,00893003
320	1,2877E+12	37315	1778558021	0,016	0,01678924	6,597	53569	613231749	0,039	0,0084344
321	1,2877E+12	37316	1778558021	0,008	0,01673991	6,637	53570	613235349	0,04	0,00790725

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
322	1,2877E+12	37317	1778558021	0,02	0,01619367	6,647	53571	613235349	0,01	0,00803805
323	1,2877E+12	37318	1778561621	0,008	0,01643156	6,657	53572	613235349	0,01	0,00816067
324	1,2877E+12	37319	1778561621	0,008	0,01590459	6,667	53573	613235349	0,01	0,00827563
325	1,2877E+12	37320	1778561621	0,011	0,01541055	6,677	53574	613238949	0,01	0,0096334
326	1,2877E+12	37321	1778561621	0,014	0,01513489	6,685	53575	613238949	0,008	0,00953131
327	1,2877E+12	37322	1778561621	0,003	0,01506396	6,693	53576	613238949	0,008	0,00943561
328	1,2877E+12	37323	1778565221	0,007	0,01643497	6,701	53577	613238949	0,008	0,00934588
329	1,2877E+12	37324	1778568821	0,029	0,01747028	6,709	53578	613238949	0,008	0,00926176
330	1,2877E+12	37325	1778568821	0,005	0,01819089	6,717	53579	613242549	0,008	0,0106829
331	1,2877E+12	37326	1778572421	0,022	0,01924146	6,757	53580	613246149	0,04	0,01001522
332	1,2877E+12	37327	1778576021	0,046	0,01916387	6,777	53581	613246149	0,02	0,01063927
333	1,2877E+12	37328	1778579621	0,067	0,01834112	6,8	53582	613249749	0,023	0,01103682
334	1,2877E+12	37329	1778583221	0,014	0,0188823	6,837	53583	613253349	0,037	0,01053451
335	1,2877E+12	37330	1778586821	0,037	0,01932716	6,877	53584	613256949	0,04	0,00987611
336	1,2877E+12	37331	1778590421	0,058	0,01830671	6,917	53585	613260549	0,04	0,00925885
337	1,2877E+12	37332	1778590421	0,008	0,02078754	6,957	53586	613264149	0,04	0,00868017
338	1,2877E+12	37333	1778594021	0,014	0,02148832	6,998	53587	613267749	0,041	0,00820016

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
339	1,2877E+12	37334	1778594021	0,055	0,0210203	7,017	53588	613267749	0,019	0,00887515
340	1,2877E+12	37335	1778597621	0,01	0,02064403	7,037	53589	613271349	0,02	0,00957045
341	1,2877E+12	37336	1778597621	0,012	0,01997878	7,057	53590	613271349	0,02	0,0102223
342	1,2877E+12	37337	1778601221	0,01	0,02048011	7,077	53591	613274949	0,02	0,01083341
343	1,2877E+12	37338	1778601221	0,01	0,0198251	7,097	53592	613274949	0,02	0,01140632
344	1,2877E+12	37339	1778604821	0,024	0,02046103	7,117	53593	613278549	0,02	0,01194342
345	1,2877E+12	37340	1778604821	0,004	0,02068222	7,137	53594	613278549	0,02	0,01244696
346	1,2877E+12	37341	1778604821	0,007	0,01963958	7,157	53595	613282149	0,02	0,01291903
347	1,2877E+12	37342	1778604821	0,009	0,0188496	7,165	53596	613282149	0,008	0,01261159
348	1,2877E+12	37343	1778604821	0,006	0,018234	7,174	53597	613282149	0,009	0,01238586
349	1,2877E+12	37344	1778608421	0,019	0,01921938	7,181	53598	613282149	0,007	0,01204925
350	1,2877E+12	37345	1778608421	0,009	0,01920567	7,189	53599	613282149	0,008	0,01179617
351	1,2877E+12	37346	1778612021	0,03	0,01994281	7,199	53600	613285749	0,01	0,01293391
352	1,2877E+12	37347	1778612021	0,011	0,02057139	7,219	53601	613285749	0,02	0,01337554
353	1,2877E+12	37348	1778615621	0,039	0,02109818	7,237	53602	613289349	0,018	0,01391457
354	1,2877E+12	37349	1778615621	0,009	0,02221704	7,258	53603	613289349	0,021	0,01435741
355	1,2877E+12	37350	1778619221	0,019	0,02276598	7,277	53604	613292949	0,019	0,01477257

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
356	1,2877E+12	37351	1778619221	0,018	0,0225306	7,298	53605	613292949	0,021	0,01516178
357	1,2877E+12	37352	1778622821	0,094	0,02249744	7,317	53606	613296549	0,019	0,01552667
358	1,2877E+12	37353	1778622821	0,017	0,02696635	7,337	53607	613296549	0,02	0,01580625
359	1,2877E+12	37354	1778626421	0,01	0,02671845	7,357	53608	613300149	0,02	0,01606836
360	1,2877E+12	37355	1778626421	0,003	0,02567355	7,377	53609	613300149	0,02	0,01631409
361	1,2877E+12	37356	1778630021	0,005	0,02638145	7,397	53610	613303749	0,02	0,01654446
362	1,2877E+12	37357	1778630021	0,006	0,02504511	7,417	53611	613303749	0,02	0,01676043
363	1,2877E+12	37358	1778633621	0,007	0,02560479	7,437	53612	613307349	0,02	0,0169629
364	1,2877E+12	37359	1778633621	0,017	0,02444199	7,457	53613	613307349	0,02	0,01715272
365	1,2877E+12	37360	1778637221	0,019	0,02435187	7,477	53614	613310949	0,02	0,01733068
366	1,2877E+12	37361	1778637221	0,015	0,02401738	7,499	53615	613310949	0,022	0,01762251
367	1,2877E+12	37362	1778640821	0,038	0,02407879	7,517	53616	613314549	0,018	0,0178961
368	1,2877E+12	37363	1778640821	0,008	0,02494887	7,537	53617	613314549	0,02	0,0180276
369	1,2877E+12	37364	1778644421	0,031	0,02538956	7,557	53618	613318149	0,02	0,01815087
370	1,2877E+12	37365	1778648021	0,031	0,02436521	7,577	53619	613318149	0,02	0,01826644
371	1,2877E+12	37366	1778648021	0,01	0,02477989	7,598	53620	613321749	0,021	0,01831229
372	1,2877E+12	37367	1778648021	0,006	0,02385615	7,644	53621	613325349	0,046	0,01754277

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
373	1,2877E+12	37368	1778648021	0,005	0,02274014	7,647	53622	613325349	0,003	0,01663385
374	1,2877E+12	37369	1778648021	0,01	0,02163138	7,654	53623	613325349	0,007	0,01603173
375	1,2877E+12	37370	1778651621	0,011	0,02215442	7,669	53624	613325349	0,015	0,01596725
376	1,2877E+12	37371	1778651621	0,018	0,02145727	7,67	53625	613325349	0,001	0,0150318
377	1,2877E+12	37372	1778655221	0,02	0,02149119	7,679	53626	613328949	0,009	0,01602981
378	1,2877E+12	37373	1778655221	0,013	0,02139799	7,701	53627	613328949	0,022	0,01640295
379	1,2877E+12	37374	1778658821	0,029	0,02174811	7,721	53628	613332549	0,02	0,01662776
380	1,2877E+12	37375	1778658821	0,011	0,02220136	7,743	53629	613332549	0,022	0,01696353
381	1,2877E+12	37376	1778662421	0,026	0,02262627	7,765	53630	613336149	0,022	0,01702831
382	1,2877E+12	37377	1778662421	0,014	0,02283713	7,777	53631	613336149	0,012	0,01671404
383	1,2877E+12	37378	1778666021	0,04	0,02303481	7,797	53632	613339749	0,02	0,01691941
384	1,2877E+12	37379	1778669621	0,081	0,02159513	7,817	53633	613339749	0,02	0,01711195
385	1,2877E+12	37380	1778669621	0,004	0,02530794	7,837	53634	613343349	0,02	0,01729245
386	1,2877E+12	37381	1778673221	0,012	0,02597619	7,877	53635	613346949	0,04	0,01621167
387	1,2877E+12	37382	1778673221	0,011	0,02510268	7,898	53636	613346949	0,021	0,01651094
388	1,2877E+12	37383	1778676821	0,005	0,02534626	7,917	53637	613350549	0,019	0,01679151
389	1,2877E+12	37384	1778676821	0,008	0,02407462	7,937	53638	613350549	0,02	0,01699204

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
390	1,2877E+12	37385	1778680421	0,024	0,02456996	7,957	53639	613354149	0,02	0,01718004
391	1,2877E+12	37386	1778680421	0,005	0,02453433	7,977	53640	613354149	0,02	0,01735628
392	1,2877E+12	37387	1778680421	0,01	0,02331344	7,997	53641	613357749	0,02	0,01752152
393	1,2877E+12	37388	1778680421	0,01	0,02248135	8,007	53642	613357749	0,01	0,01705142
394	1,2877E+12	37389	1778684021	0,016	0,02295126	8,017	53643	613357749	0,01	0,01661071
395	1,2877E+12	37390	1778684021	0,014	0,02251681	8,029	53644	613357749	0,012	0,01632254
396	1,2877E+12	37391	1778687621	0,028	0,02273451	8,037	53645	613361349	0,008	0,01730238
397	1,2877E+12	37392	1778687621	0,016	0,0230636	8,057	53646	613361349	0,02	0,01747098
398	1,2877E+12	37393	1778691221	0,037	0,02312213	8,077	53647	613364949	0,02	0,01762905
399	1,2877E+12	37394	1778691221	0,012	0,02398949	8,097	53648	613364949	0,02	0,01777723
400	1,2877E+12	37395	1778691221	0,002	0,02324015	8,117	53649	613368549	0,02	0,01791615
401	1,2877E+12	37396	1778691221	0,003	0,02191264	8,125	53650	613368549	0,008	0,01729639
402	1,2877E+12	37397	1778691221	0,009	0,0207306	8,133	53651	613368549	0,008	0,01671537
403	1,2877E+12	37398	1778694821	0,005	0,02137244	8,141	53652	613368549	0,008	0,01617066
404	1,2877E+12	37399	1778694821	0,008	0,02034916	8,149	53653	613368549	0,008	0,01565999
405	1,2877E+12	37400	1778698421	0,026	0,02107734	8,157	53654	613372149	0,008	0,01668124
406	1,2877E+12	37401	1778698421	0,014	0,02138501	8,177	53655	613372149	0,02	0,01688866

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
407	1,2877E+12	37402	1778702021	0,034	0,02167344	8,198	53656	613375749	0,021	0,01702062
408	1,2877E+12	37403	1778702021	0,009	0,02244385	8,217	53657	613375749	0,019	0,01714433
409	1,2877E+12	37404	1778705621	0,021	0,02297861	8,237	53658	613379349	0,02	0,01732281
410	1,2877E+12	37405	1778705621	0,018	0,02285495	8,257	53659	613379349	0,02	0,01749014
411	1,2877E+12	37406	1778709221	0,027	0,02280151	8,277	53660	613382949	0,02	0,017647
412	1,2877E+12	37407	1778712821	0,034	0,02218892	8,297	53661	613382949	0,02	0,01779407
413	1,2877E+12	37408	1778716421	0,053	0,02117711	8,319	53662	613386549	0,022	0,01780694
414	1,2877E+12	37409	1778720021	0,032	0,02066604	8,359	53663	613390149	0,04	0,016694
415	1,2877E+12	37410	1778723621	0,048	0,01987441	8,397	53664	613393749	0,038	0,01577563
416	1,2877E+12	37411	1778727221	0,032	0,01913226	8,439	53665	613397349	0,042	0,01491465
417	1,2877E+12	37412	1778730821	0,042	0,0184365	8,477	53666	613400949	0,038	0,01410749
418	1,2877E+12	37413	1778734421	0,041	0,01740922	8,517	53667	613404549	0,04	0,01322577
419	1,2877E+12	37414	1778734421	0,017	0,01888364	8,557	53668	613408149	0,04	0,01239916
420	1,2877E+12	37415	1778734421	0,021	0,01876591	8,597	53669	613411749	0,04	0,01162421
421	1,2877E+12	37416	1778734421	0,011	0,01890554	8,607	53670	613411749	0,01	0,0115227
422	1,2877E+12	37417	1778738021	0,006	0,01953645	8,617	53671	613411749	0,01	0,01142753
423	1,2877E+12	37418	1778741621	0,042	0,02044042	8,627	53672	613411749	0,01	0,01133831

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
424	1,2877E+12	37419	1778741621	0,007	0,02178789	8,637	53673	613415349	0,01	0,01250466
425	1,2877E+12	37420	1778745221	0,011	0,02248865	8,677	53674	613418949	0,04	0,01172312
426	1,2877E+12	37421	1778745221	0,049	0,02177061	8,697	53675	613418949	0,02	0,01224043
427	1,2877E+12	37422	1778748821	0,009	0,02097245	8,717	53676	613422549	0,02	0,0127254
428	1,2877E+12	37423	1778752421	0,018	0,02159917	8,737	53677	613422549	0,02	0,01318006
429	1,2877E+12	37424	1778756021	0,039	0,02162422	8,757	53678	613426149	0,02	0,01360631
430	1,2877E+12	37425	1778759621	0,045	0,02033521	8,798	53679	613429749	0,041	0,01281841
431	1,2877E+12	37426	1778763221	0,043	0,01937676	8,837	53680	613433349	0,039	0,01207976
432	1,2877E+12	37427	1778766821	0,054	0,01835321	8,877	53681	613436949	0,04	0,01132478
433	1,2877E+12	37428	1778770421	0,026	0,01808113	8,917	53682	613440549	0,04	0,01061698
434	1,2877E+12	37429	1778774021	0,032	0,01782606	8,957	53683	613444149	0,04	0,00995342
435	1,2877E+12	37430	1778777621	0,05	0,01721193	8,998	53684	613447749	0,041	0,00939383
436	1,2877E+12	37431	1778777621	0,018	0,01926119	9,037	53685	613451349	0,039	0,00886922
437	1,2877E+12	37432	1778777621	0,018	0,01918236	9,077	53686	613454949	0,04	0,00831489
438	1,2877E+12	37433	1778777621	0,001	0,01910847	9,087	53687	613454949	0,01	0,00842021
439	1,2877E+12	37434	1778781221	0,003	0,02035169	9,097	53688	613454949	0,01	0,00851895
440	1,2877E+12	37435	1778784821	0,034	0,02139221	9,107	53689	613454949	0,01	0,00861151

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
441	1,2877E+12	37436	1778788421	0,042	0,02043019	9,117	53690	613458549	0,01	0,00994829
442	1,2877E+12	37437	1778792021	0,043	0,01927831	9,157	53691	613462149	0,04	0,00932652
443	1,2877E+12	37438	1778792021	0,011	0,02076091	9,198	53692	613465749	0,041	0,00880612
444	1,2877E+12	37439	1778795621	0,027	0,02127585	9,241	53693	613469349	0,043	0,00844323
445	1,2877E+12	37440	1778799221	0,04	0,02075861	9,258	53694	613469349	0,017	0,00897803
446	1,2877E+12	37441	1778802821	0,049	0,0194612	9,277	53695	613472949	0,019	0,0097294
447	1,2877E+12	37442	1778806421	0,034	0,01880738	9,317	53696	613476549	0,04	0,00912132
448	1,2877E+12	37443	1778810021	0,058	0,01800691	9,36	53697	613480149	0,043	0,00873873
449	1,2877E+12	37444	1778813621	0,031	0,01800648	9,398	53698	613483749	0,038	0,00831756
450	1,2877E+12	37445	1778817221	0,035	0,01744358	9,437	53699	613487349	0,039	0,00786022
451	1,2877E+12	37446	1778820821	0,03	0,01666585	9,478	53700	613490949	0,041	0,00743145
452	1,2877E+12	37447	1778820821	0,005	0,01749924	9,517	53701	613494549	0,039	0,00702949
453	1,2877E+12	37448	1778820821	0,014	0,01671804	9,557	53702	613498149	0,04	0,00659014
454	1,2877E+12	37449	1778820821	0,007	0,01654816	9,567	53703	613498149	0,01	0,00680326
455	1,2877E+12	37450	1778824421	0,016	0,0175764	9,577	53704	613498149	0,01	0,00700306
456	1,2877E+12	37451	1778828021	0,055	0,01797787	9,587	53705	613498149	0,01	0,00719037
457	1,2877E+12	37452	1778828021	0,009	0,02029176	9,597	53706	613501749	0,01	0,00861597

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
458	1,2877E+12	37453	1778831621	0,022	0,02096102	9,637	53707	613505349	0,04	0,00807747
459	1,2877E+12	37454	1778831621	0,015	0,02102596	9,657	53708	613505349	0,02	0,00882263
460	1,2877E+12	37455	1778835221	0,021	0,02127434	9,677	53709	613508949	0,02	0,00952121
461	1,2877E+12	37456	1778838821	0,083	0,02113219	9,699	53710	613508949	0,022	0,01030114
462	1,2877E+12	37457	1778842421	0,049	0,02249893	9,717	53711	613512549	0,018	0,01103232
463	1,2877E+12	37458	1778846021	0,016	0,02165524	9,757	53712	613516149	0,04	0,0103428
464	1,2877E+12	37459	1778849621	0,016	0,02180179	9,801	53713	613519749	0,044	0,00994637
465	1,2877E+12	37460	1778853221	0,033	0,02193918	9,837	53714	613523349	0,036	0,00957472
466	1,2877E+12	37461	1778853221	0,014	0,02263048	9,885	53715	613526949	0,048	0,0094763
467	1,2877E+12	37462	1778856821	0,022	0,02284108	9,917	53716	613530549	0,032	0,00938403
468	1,2877E+12	37463	1778856821	0,024	0,02278851	9,937	53717	613530549	0,02	0,01004753
469	1,2877E+12	37464	1778860421	0,02	0,02236423	9,957	53718	613534149	0,02	0,01066956
470	1,2877E+12	37465	1778860421	0,015	0,02221646	9,977	53719	613534149	0,02	0,01125271
471	1,2877E+12	37466	1778864021	0,03	0,02239043	9,999	53720	613537749	0,022	0,01167442
472	1,2877E+12	37467	1778864021	0,007	0,02286603	10,017	53721	613537749	0,018	0,01206977
473	1,2877E+12	37468	1778864021	0,007	0,0218744	10,037	53722	613541349	0,02	0,01256541
474	1,2877E+12	37469	1778864021	0,029	0,02094475	10,045	53723	613541349	0,008	0,01228007

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
475	1,2877E+12	37470	1778864021	0,005	0,02144821	10,053	53724	613541349	0,008	0,01201257
476	1,2877E+12	37471	1778867621	0,006	0,02229519	10,061	53725	613541349	0,008	0,01176178
477	1,2877E+12	37472	1778871221	0,022	0,02302674	10,069	53726	613541349	0,008	0,01152667
478	1,2877E+12	37473	1778871221	0,012	0,02296257	10,077	53727	613544949	0,008	0,01280625
479	1,2877E+12	37474	1778874821	0,027	0,02327741	10,117	53728	613548549	0,04	0,01200586
480	1,2877E+12	37475	1778874821	0,014	0,02351007	10,137	53729	613548549	0,02	0,01250549
481	1,2877E+12	37476	1778878421	0,027	0,02366569	10,157	53730	613552149	0,02	0,0129739
482	1,2877E+12	37477	1778878421	0,007	0,02387409	10,179	53731	613552149	0,022	0,01353803
483	1,2877E+12	37478	1778878421	0,005	0,02281946	10,201	53732	613555749	0,022	0,01381691
484	1,2877E+12	37479	1778878421	0,005	0,02170574	10,203	53733	613555749	0,002	0,01307835
485	1,2877E+12	37480	1778878421	0,012	0,02066163	10,21	53734	613555749	0,007	0,01269845
486	1,2877E+12	37481	1778878421	0,01	0,02012028	10,217	53735	613555749	0,007	0,0123423
487	1,2877E+12	37482	1778882021	0,005	0,02073776	10,223	53736	613555749	0,006	0,01194591
488	1,2877E+12	37483	1778882021	0,009	0,01975415	10,23	53737	613555749	0,007	0,01163679
489	1,2877E+12	37484	1778885621	0,022	0,02045702	10,237	53738	613559349	0,007	0,01297199
490	1,2877E+12	37485	1778885621	0,017	0,02055345	10,257	53739	613559349	0,02	0,01341124
491	1,2877E+12	37486	1778889221	0,028	0,02070636	10,277	53740	613562949	0,02	0,01382304

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
492	1,2877E+12	37487	1778889221	0,013	0,02116222	10,299	53741	613562949	0,022	0,0143341
493	1,2877E+12	37488	1778892821	0,03	0,02152708	10,319	53742	613566549	0,02	0,01468821
494	1,2877E+12	37489	1778892821	0,01	0,02205664	10,338	53743	613566549	0,019	0,0149577
495	1,2877E+12	37490	1778896421	0,026	0,0225531	10,357	53744	613570149	0,019	0,01533534
496	1,2877E+12	37491	1778896421	0,014	0,02276853	10,377	53745	613570149	0,02	0,01562689
497	1,2877E+12	37492	1778900021	0,027	0,02297049	10,398	53746	613573749	0,021	0,01583771
498	1,2877E+12	37493	1778900021	0,016	0,02322234	10,417	53747	613573749	0,019	0,01603535
499	1,2877E+12	37494	1778903621	0,024	0,02327094	10,437	53748	613577349	0,02	0,01628314
500	1,2877E+12	37495	1778903621	0,013	0,02331651	10,457	53749	613577349	0,02	0,01651544
501	1,2877E+12	37496	1778907221	0,028	0,02354673	10,477	53750	613580949	0,02	0,01673323
502	1,2877E+12	37497	1778907221	0,019	0,02382506	10,499	53751	613580949	0,022	0,0170624
503	1,2877E+12	37498	1778907221	0,008	0,02352349	10,517	53752	613584549	0,018	0,017371
504	1,2877E+12	37499	1778907221	0,014	0,02255327	10,522	53753	613584549	0,005	0,01659781
505	1,2877E+12	37500	1778907221	0,003	0,02201869	10,529	53754	613584549	0,007	0,01599795
506	1,2877E+12	37501	1778907221	0,008	0,02083002	10,534	53755	613584549	0,005	0,01531058
507	1,2877E+12	37502	1778907221	0,004	0,02002815	10,539	53756	613584549	0,005	0,01466617
508	1,2877E+12	37503	1778910821	0,004	0,02102639	10,545	53757	613584549	0,006	0,01412453

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
509	1,2877E+12	37504	1778910821	0,049	0,01996224	10,551	53758	613584549	0,006	0,01361675
510	1,2877E+12	37505	1778914421	0,011	0,0192771	10,557	53759	613588149	0,006	0,0148907
511	1,2877E+12	37506	1778914421	0,007	0,01875978	10,577	53760	613588149	0,02	0,01521003
512	1,2877E+12	37507	1778918021	0,008	0,01964979	10,597	53761	613591749	0,02	0,01550941
513	1,2877E+12	37508	1778918021	0,004	0,01892168	10,617	53762	613591749	0,02	0,01579007
514	1,2877E+12	37509	1778921621	0,02	0,01998908	10,637	53763	613595349	0,02	0,01605319
515	1,2877E+12	37510	1778921621	0,014	0,01998976	10,657	53764	613595349	0,02	0,01629986
516	1,2877E+12	37511	1778925221	0,021	0,0203654	10,677	53765	613598949	0,02	0,01653112
517	1,2877E+12	37512	1778925221	0,019	0,02040506	10,699	53766	613598949	0,022	0,01687293
518	1,2877E+12	37513	1778928821	0,026	0,02044225	10,717	53767	613602549	0,018	0,01719337
519	1,2877E+12	37514	1778928821	0,018	0,02078961	10,737	53768	613602549	0,02	0,01736878
520	1,2877E+12	37515	1778932421	0,039	0,02086526	10,757	53769	613606149	0,02	0,01753324
521	1,2877E+12	37516	1778932421	0,004	0,02199868	10,777	53770	613606149	0,02	0,01768741
522	1,2877E+12	37517	1778936021	0,021	0,02287376	10,797	53771	613609749	0,02	0,01783194
523	1,2877E+12	37518	1778936021	0,011	0,02275665	10,817	53772	613609749	0,02	0,01796745
524	1,2877E+12	37519	1778939621	0,027	0,02314686	10,837	53773	613613349	0,02	0,01809448
525	1,2877E+12	37520	1778939621	0,014	0,02338768	10,857	53774	613613349	0,02	0,01821358

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
526	1,2877E+12	37521	1778943221	0,028	0,02355095	10,877	53775	613616949	0,02	0,01832523
527	1,2877E+12	37522	1778943221	0,012	0,02382902	10,899	53776	613616949	0,022	0,0185549
528	1,2877E+12	37523	1778946821	0,024	0,0240897	10,917	53777	613620549	0,018	0,01877022
529	1,2877E+12	37524	1778946821	0,015	0,0240841	10,937	53778	613620549	0,02	0,01884708
530	1,2877E+12	37525	1778950421	0,027	0,02414134	10,961	53779	613624149	0,024	0,01866914
531	1,2877E+12	37526	1778950421	0,016	0,02432001	10,978	53780	613624149	0,017	0,01856482
532	1,2877E+12	37527	1778950421	0,012	0,02380001	10,998	53781	613627749	0,02	0,01865452
533	1,2877E+12	37528	1778950421	0,015	0,02306251	11,002	53782	613627749	0,004	0,01773861
534	1,2877E+12	37529	1778950421	0,001	0,0225586	11,008	53783	613627749	0,006	0,01700495
535	1,2877E+12	37530	1778950421	0,004	0,02121119	11,014	53784	613627749	0,006	0,01631714
536	1,2877E+12	37531	1778950421	0,014	0,02013549	11,019	53785	613627749	0,005	0,01560982
537	1,2877E+12	37532	1778954021	0,005	0,02050202	11,026	53786	613627749	0,007	0,0150717
538	1,2877E+12	37533	1778954021	0,012	0,01953314	11,031	53787	613627749	0,005	0,01444222
539	1,2877E+12	37534	1778957621	0,011	0,02006232	11,037	53788	613631349	0,006	0,01566458
540	1,2877E+12	37535	1778957621	0,009	0,01949593	11,057	53789	613631349	0,02	0,01593555
541	1,2877E+12	37536	1778961221	0,015	0,02021493	11,077	53790	613634949	0,02	0,01618957
542	1,2877E+12	37537	1778961221	0,018	0,019889	11,101	53791	613634949	0,024	0,01667773

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
543	1,2877E+12	37538	1778964821	0,028	0,02002094	11,117	53792	613638549	0,016	0,01713537
544	1,2877E+12	37539	1778964821	0,026	0,02051963	11,137	53793	613638549	0,02	0,01731441
545	1,2877E+12	37540	1778968421	0,016	0,02011215	11,157	53794	613642149	0,02	0,01748226
546	1,2877E+12	37541	1778968421	0,014	0,01985514	11,177	53795	613642149	0,02	0,01763962
547	1,2877E+12	37542	1778972021	0,024	0,02023919	11,197	53796	613645749	0,02	0,01778714
548	1,2877E+12	37543	1778972021	0,013	0,02047425	11,217	53797	613645749	0,02	0,01792544
549	1,2877E+12	37544	1778975621	0,028	0,0208821	11,237	53798	613649349	0,02	0,0180551
550	1,2877E+12	37545	1778975621	0,013	0,02132697	11,257	53799	613649349	0,02	0,01817666
551	1,2877E+12	37546	1778979221	0,028	0,02168154	11,277	53800	613652949	0,02	0,01829062
552	1,2877E+12	37547	1778979221	0,015	0,02207644	11,297	53801	613652949	0,02	0,01839745
553	1,2877E+12	37548	1778982821	0,109	0,02225916	11,825	53802	613656549	0,528	0,04774761
554	1,2877E+12	37549	1778982821	0,326	0,02768047	11,826	53803	613656549	0,001	0,04482589
555	1,2877E+12	37550	1778986421	0,043	0,04382544	11,826	53804	613660149	0	0,04452427
556	1,2877E+12	37551	1778986421	0,021	0,04377385	11,826	53805	613660149	0	0,0417415
557	1,2877E+12	37552	1778990021	0,03	0,04222548	11,827	53806	613663749	0,001	0,04157016
558	1,2877E+12	37553	1778990021	0,014	0,04146139	11,827	53807	613663749	0	0,03897202
559	1,2877E+12	37554	1778993621	0,012	0,04049505	11,828	53808	613667349	0,001	0,03897377

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Tiempo	Secuencia	TimeStamp	Retardo	Jitter
560	1,2877E+12	37555	1778993621	0,014	0,03871411	11,829	53809	613667349	0,001	0,03660041
561	1,2877E+12	37556	1778993621	0,005	0,03716948	11,829	53810	613670949	0	0,03681289
562	1,2877E+12	37557	1778993621	0,009	0,03515889	11,829	53811	613670949	0	0,03451208
563	1,2877E+12	37558	1778993621	0,007	0,03352396	11,83	53812	613670949	0,001	0,03241758
564	1,2877E+12	37559	1778993621	0,004	0,03186621	11,83	53813	613670949	0	0,03039148
565	1,2877E+12	37560	1778993621	0,004	0,03012457	11,83	53814	613670949	0	0,02849201
566	1,2877E+12	37561	1778997221	0,003	0,03049179	11,831	53815	613670949	0,001	0,02677376
567	1,2877E+12	37562	1778997221	0,002	0,02877355	11,831	53816	613670949	0	0,0251004
568	1,2877E+12	37563	1779000821	0,006	0,0293502	11,832	53817	613674549	0,001	0,02596912
569	1,2877E+12	37564	1779000821	0,004	0,02789081	11,832	53818	613674549	0	0,02434605
570	1,2877E+12	37565	1779004421	0,009	0,02839764	11,832	53819	613678149	0	0,02532443
571	1,2877E+12	37566	1779004421	0,004	0,02718529	11,833	53820	613678149	0,001	0,02380415
PROMEDIO				0,02081961	0,02149951	PROMEDIO			0,02075789	0,01511602

4.2.1.2 Cálculo del Jitter y Retardo con los Paquetes de Audio en el Escenario 1 y 2

Tabla 4.4 - Cálculo de Jitter y Retardo paquetes audio

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
1	1,2877E+12	61717	871202817	0	0	0	7141	213518512	0	0
2	1,2877E+12	61718	871203841	0,024	0,0014512	0	7142	213519536	0	0,00145125
3	1,2877E+12	61719	871204865	0,024	0,0014093	0,632	7143	214294338	0,632	1,05993594
4	1,2877E+12	61720	871205889	0,029	0,00137	0,657	7144	214295362	0,025	0,9938012
5	1,2877E+12	61721	871206913	0,021	0,0016456	0,679	7145	214296386	0,022	0,93176487
6	1,2877E+12	61722	871207937	0,027	0,0016815	0,702	7146	214297410	0,023	0,87354331
7	1,2877E+12	61723	871208961	0,013	0,0018127	0,725	7147	214298434	0,023	0,8189606
8	1,2877E+12	61724	871209985	0,024	0,0023381	0,75	7148	214299458	0,025	0,76788682
9	1,2877E+12	61725	871211009	0,024	0,0022407	0,772	7149	214300482	0,022	0,71997014
10	1,2877E+12	61726	871212033	0,021	0,0021494	0,806	7150	214301506	0,034	0,67564576
11	1,2877E+12	61727	871213057	0,03	0,0021538	0,818	7151	214302530	0,012	0,63411915
12	1,2877E+12	61728	871214081	0,02	0,002443	0,841	7152	214303554	0,023	0,59450045
13	1,2877E+12	61729	871215105	0,022	0,0024915	0,865	7153	214304578	0,024	0,55739292
14	1,2877E+12	61730	871216129	0,024	0,0024121	0,888	7154	214305602	0,023	0,52256961
15	1,2877E+12	61731	871217153	0,03	0,0023101	0,914	7155	214306626	0,026	0,49008276

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
16	1,2877E+12	61732	871218177	0,024	0,0025894	0,934	7156	214307650	0,02	0,45965384
17	1,2877E+12	61733	871219201	0,016	0,0024764	0,959	7157	214308674	0,025	0,43103673
18	1,2877E+12	61734	871220225	0,025	0,0027728	0,983	7158	214309698	0,024	0,40414568
19	1,2877E+12	61735	871221249	0,03	0,0027108	1,005	7159	214310722	0,022	0,37896282
20	1,2877E+12	61736	871222273	0,039	0,0029651	1,028	7160	214311746	0,023	0,3552914
21	1,2877E+12	61737	871223297	0,024	0,003766	1,051	7161	214312770	0,023	0,33309943
22	1,2877E+12	61738	871224321	0,004	0,0035794	1,074	7162	214313794	0,023	0,31229446
23	1,2877E+12	61739	871225345	0,025	0,004557	1,097	7163	214314818	0,023	0,29278981
24	1,2877E+12	61740	871226369	0,036	0,0043834	1,121	7164	214315842	0,024	0,2745392
25	1,2877E+12	61741	871227393	0,017	0,0049082	1,144	7165	214316866	0,023	0,25739424
26	1,2877E+12	61742	871228417	0,004	0,0049902	1,167	7166	214317890	0,023	0,24132085
27	1,2877E+12	61743	871229441	0,024	0,0058795	1,191	7167	214318914	0,024	0,22628705
28	1,2877E+12	61744	871230465	0,023	0,0055608	1,213	7168	214319938	0,022	0,21222036
29	1,2877E+12	61745	871231489	0,025	0,005227	1,237	7169	214320962	0,024	0,19900534
30	1,2877E+12	61746	871232513	0,03	0,0050116	1,26	7170	214321986	0,023	0,18658125
31	1,2877E+12	61747	871233537	0,016	0,0051221	1,284	7171	214323010	0,024	0,17496868
32	1,2877E+12	61748	871234561	0,024	0,0052532	1,309	7172	214324034	0,025	0,16414439

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
33	1,2877E+12	61749	871235585	0,022	0,0049736	1,33	7173	214325058	0,021	0,15402411
34	1,2877E+12	61750	871236609	0,029	0,004739	1,356	7174	214326082	0,026	0,14457136
35	1,2877E+12	61751	871237633	0,032	0,0048041	1,377	7175	214327106	0,021	0,13567439
36	1,2877E+12	61752	871238657	0,014	0,0050526	1,402	7176	214328130	0,025	0,127306
37	1,2877E+12	61753	871239681	0,022	0,0053131	1,423	7177	214329154	0,021	0,11948812
38	1,2877E+12	61754	871240705	0,022	0,0050572	1,446	7178	214330178	0,023	0,11203386
39	1,2877E+12	61755	871241729	0,021	0,0048174	1,469	7179	214331202	0,023	0,10504549
40	1,2877E+12	61756	871242753	0,044	0,0046551	1,493	7180	214332226	0,024	0,0985289
41	1,2877E+12	61757	871243777	0,013	0,0056629	1,516	7181	214333250	0,023	0,09238459
42	1,2877E+12	61758	871244801	0,013	0,0059477	1,539	7182	214334274	0,023	0,0866243
43	1,2877E+12	61759	871245825	0,023	0,0062147	1,562	7183	214335298	0,023	0,08122403
44	1,2877E+12	61760	871246849	0,022	0,00584	1,585	7184	214336322	0,023	0,07616127
45	1,2877E+12	61761	871247873	0,025	0,0055513	1,609	7185	214337346	0,024	0,07144995
46	1,2877E+12	61762	871248897	0,11	0,0053156	1,634	7186	214338370	0,025	0,06709558
47	1,2877E+12	61763	871249921	0,011	0,0104071	1,654	7187	214339394	0,02	0,06310335
48	1,2877E+12	61764	871250945	0,012	0,0105204	1,678	7188	214340418	0,024	0,05920815
49	1,2877E+12	61765	871251969	0,018	0,0105641	1,702	7189	214341442	0,024	0,05555639

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
50	1,2877E+12	61766	871252993	0,003	0,0102301	1,725	7190	214342466	0,023	0,05209786
51	1,2877E+12	61767	871254017	0,011	0,0108545	1,748	7191	214343490	0,023	0,04885549
52	1,2877E+12	61768	871255041	0,005	0,0109398	1,771	7192	214344514	0,023	0,04581577
53	1,2877E+12	61769	871256065	0,018	0,0113948	1,794	7193	214345538	0,023	0,04296603
54	1,2877E+12	61770	871257089	0,02	0,0110089	1,818	7194	214346562	0,024	0,04032941
55	1,2877E+12	61771	871258113	0,025	0,0105221	1,841	7195	214347586	0,023	0,03782257
56	1,2877E+12	61772	871259137	0,023	0,0099757	1,864	7196	214348610	0,023	0,0354724
57	1,2877E+12	61773	871260161	0,025	0,009366	1,887	7197	214349634	0,023	0,03326913
58	1,2877E+12	61774	871261185	0,018	0,0088919	1,911	7198	214350658	0,024	0,03123856
59	1,2877E+12	61775	871262209	0,025	0,0086624	1,934	7199	214351682	0,023	0,0292999
60	1,2877E+12	61776	871263233	0,024	0,0082322	1,957	7200	214352706	0,023	0,0274824
61	1,2877E+12	61777	871264257	0,025	0,0077665	1,984	7201	214353730	0,027	0,026001
62	1,2877E+12	61778	871265281	0,02	0,0073923	2,005	7202	214354754	0,021	0,02451469
63	1,2877E+12	61779	871266305	0,022	0,0071315	2,027	7203	214355778	0,022	0,02305877
64	1,2877E+12	61780	871267329	0,027	0,0067621	2,05	7204	214356802	0,023	0,02163134
65	1,2877E+12	61781	871268353	0,031	0,0065757	2,073	7205	214357826	0,023	0,02029313
66	1,2877E+12	61782	871269377	0,029	0,006651	2,096	7206	214358850	0,023	0,01903856

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
67	1,2877E+12	61783	871270401	0,062	0,0065965	2,121	7207	214359874	0,025	0,0179599
68	1,2877E+12	61784	871271425	0,021	0,008608	2,143	7208	214360898	0,022	0,01691365
69	1,2877E+12	61785	871272449	0,026	0,0082087	2,166	7209	214361922	0,023	0,0158703
70	1,2877E+12	61786	871273473	0,077	0,0078695	2,189	7210	214362946	0,023	0,01489215
71	1,2877E+12	61787	871274497	0,026	0,0107389	2,212	7211	214363970	0,023	0,01397514
72	1,2877E+12	61788	871275521	0,016	0,0102414	2,236	7212	214364994	0,024	0,01315044
73	1,2877E+12	61789	871276545	0,034	0,0100526	2,259	7213	214366018	0,023	0,01234229
74	1,2877E+12	61790	871277569	0,023	0,0100981	2,282	7214	214367042	0,023	0,01158464
75	1,2877E+12	61791	871278593	0,018	0,0094807	2,305	7215	214368066	0,023	0,01087435
76	1,2877E+12	61792	871279617	0,051	0,0092144	2,327	7216	214369090	0,022	0,01027095
77	1,2877E+12	61793	871280641	0,006	0,0103747	2,352	7217	214370114	0,025	0,00974027
78	1,2877E+12	61794	871281665	0,004	0,0108026	2,375	7218	214371138	0,023	0,00914525
79	1,2877E+12	61795	871282689	0,003	0,0113287	2,398	7219	214372162	0,023	0,00858742
80	1,2877E+12	61796	871283713	0,03	0,0118844	2,421	7220	214373186	0,023	0,00806445
81	1,2877E+12	61797	871284737	0,049	0,0115653	2,445	7221	214374210	0,024	0,00760918
82	1,2877E+12	61798	871285761	0,093	0,0124538	2,467	7222	214375234	0,022	0,00720985
83	1,2877E+12	61799	871286785	0,009	0,0160367	2,493	7223	214376258	0,026	0,00693299

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
84	1,2877E+12	61800	871287809	0,01	0,0159231	2,514	7224	214377282	0,021	0,00663842
85	1,2877E+12	61801	871288833	0,01	0,0157542	2,537	7225	214378306	0,023	0,00623727
86	1,2877E+12	61802	871289857	0,017	0,0155958	2,561	7226	214379330	0,024	0,00589619
87	1,2877E+12	61803	871290881	0,008	0,0150098	2,584	7227	214380354	0,023	0,00554143
88	1,2877E+12	61804	871291905	0,008	0,0150229	2,609	7228	214381378	0,025	0,00530634
89	1,2877E+12	61805	871292929	0,011	0,0150352	2,63	7229	214382402	0,021	0,00511344
90	1,2877E+12	61806	871293953	0,008	0,0148593	2,654	7230	214383426	0,024	0,0048426
91	1,2877E+12	61807	871294977	0,011	0,0148818	2,677	7231	214384450	0,023	0,00455369
92	1,2877E+12	61808	871296001	0,04	0,0147155	2,7	7232	214385474	0,023	0,00428283
93	1,2877E+12	61809	871297025	0,011	0,0148445	2,723	7233	214386498	0,023	0,0040289
94	1,2877E+12	61810	871298049	0,016	0,0146805	2,746	7234	214387522	0,023	0,00379084
95	1,2877E+12	61811	871299073	0,013	0,0142142	2,77	7235	214388546	0,024	0,00360267
96	1,2877E+12	61812	871300097	0,009	0,0139645	2,793	7236	214389570	0,023	0,00339125
97	1,2877E+12	61813	871301121	0,014	0,0139805	2,816	7237	214390594	0,023	0,00319304
98	1,2877E+12	61814	871302145	0,02	0,013683	2,839	7238	214391618	0,023	0,00300722
99	1,2877E+12	61815	871303169	0,011	0,013029	2,863	7239	214392642	0,024	0,00286802
100	1,2877E+12	61816	871304193	0,012	0,0129785	2,886	7240	214393666	0,023	0,00270252

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
101	1,2877E+12	61817	871305217	0,023	0,0128686	2,909	7241	214394690	0,023	0,00254736
102	1,2877E+12	61818	871306241	0,024	0,012078	2,933	7242	214395714	0,024	0,0024369
103	1,2877E+12	61819	871307265	0,018	0,0113719	2,955	7243	214396738	0,022	0,00236084
104	1,2877E+12	61820	871308289	0,031	0,0109874	2,979	7244	214397762	0,024	0,00226204
105	1,2877E+12	61821	871309313	0,019	0,0107869	3,002	7245	214398786	0,023	0,00213441
106	1,2877E+12	61822	871310337	0,053	0,0103765	3,025	7246	214399810	0,023	0,00201476
107	1,2877E+12	61823	871311361	0,014	0,0115892	3,052	7247	214400834	0,027	0,00212509
108	1,2877E+12	61824	871312385	0,015	0,0114411	3,075	7248	214401858	0,023	0,00200602
109	1,2877E+12	61825	871313409	0,014	0,0112398	3,095	7249	214402882	0,02	0,00208189
110	1,2877E+12	61826	871314433	0,039	0,0111136	3,118	7250	214403906	0,023	0,00196552
111	1,2877E+12	61827	871315457	0,014	0,0114052	3,141	7251	214404930	0,023	0,00185642
112	1,2877E+12	61828	871316481	0,029	0,0112687	3,164	7252	214405954	0,023	0,00175414
113	1,2877E+12	61829	871317505	0,014	0,0109256	3,195	7253	214406978	0,031	0,00213076
114	1,2877E+12	61830	871318529	0,019	0,010819	3,227	7254	214408002	0,032	0,00254634
115	1,2877E+12	61831	871319553	0,022	0,0104066	3,235	7255	214409026	0,008	0,00333844
116	1,2877E+12	61832	871320577	0,021	0,0098324	3,257	7256	214410050	0,022	0,00320604
117	1,2877E+12	61833	871321601	0,036	0,0093566	3,281	7257	214411074	0,024	0,00305441

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
118	1,2877E+12	61834	871322625	0,03	0,0095706	3,304	7258	214412098	0,023	0,00287726
119	1,2877E+12	61835	871323649	0,066	0,0093962	3,327	7259	214413122	0,023	0,00271118
120	1,2877E+12	61836	871324673	0,007	0,0114827	3,35	7260	214414146	0,023	0,00255548
121	1,2877E+12	61837	871325697	0,015	0,0117788	3,373	7261	214415170	0,023	0,00240951
122	1,2877E+12	61838	871326721	0,021	0,0115563	3,397	7262	214416194	0,024	0,00230766
123	1,2877E+12	61839	871327745	0,055	0,0109728	3,431	7263	214417218	0,034	0,00283719
124	1,2877E+12	61840	871328769	0,015	0,0122733	3,443	7264	214418242	0,012	0,00336111
125	1,2877E+12	61841	871329793	0,01	0,0120199	3,466	7265	214419266	0,023	0,00316479
126	1,2877E+12	61842	871330817	0,012	0,0120949	3,489	7266	214420290	0,023	0,00298074
127	1,2877E+12	61843	871331841	0,01	0,0120402	3,513	7267	214421314	0,024	0,00284319
128	1,2877E+12	61844	871332865	0,012	0,012114	3,536	7268	214422338	0,023	0,00267924
129	1,2877E+12	61845	871333889	0,038	0,0120581	3,559	7269	214423362	0,023	0,00252554
130	1,2877E+12	61846	871334913	0,011	0,0122282	3,582	7270	214424386	0,023	0,00238144
131	1,2877E+12	61847	871335937	0,01	0,0122277	3,606	7271	214425410	0,024	0,00228135
132	1,2877E+12	61848	871336961	0,053	0,0122897	3,629	7272	214426434	0,023	0,00215251
133	1,2877E+12	61849	871337985	0,049	0,0133829	3,652	7273	214427458	0,023	0,00203173
134	1,2877E+12	61850	871339009	0,022	0,0141577	3,675	7274	214428482	0,023	0,00191849

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
135	1,2877E+12	61851	871340033	0,027	0,0133491	3,701	7275	214429506	0,026	0,00197234
136	1,2877E+12	61852	871341057	0,026	0,012751	3,722	7276	214430530	0,021	0,00198782
137	1,2877E+12	61853	871342081	0,002	0,0121278	3,745	7277	214431554	0,023	0,00187732
138	1,2877E+12	61854	871343105	0,001	0,0126961	3,768	7278	214432578	0,023	0,00177374
139	1,2877E+12	61855	871344129	0,004	0,0132913	3,79	7279	214433602	0,022	0,00173913
140	1,2877E+12	61856	871345153	0,027	0,0136619	3,815	7280	214434626	0,025	0,00174168
141	1,2877E+12	61857	871346177	0,029	0,0130443	3,839	7281	214435650	0,024	0,00168158
142	1,2877E+12	61858	871347201	0,035	0,0125902	3,861	7282	214436674	0,022	0,00165273
143	1,2877E+12	61859	871348225	0,028	0,0125396	3,884	7283	214437698	0,023	0,00156318
144	1,2877E+12	61860	871349249	0,074	0,0120546	3,907	7284	214438722	0,023	0,00147923
145	1,2877E+12	61861	871350273	0,014	0,014475	3,931	7285	214439746	0,024	0,00143553
146	1,2877E+12	61862	871351297	0,013	0,0141465	3,954	7286	214440770	0,023	0,00135956
147	1,2877E+12	61863	871352321	0,012	0,0139011	3,977	7287	214441794	0,023	0,00128833
148	1,2877E+12	61864	871353345	0,022	0,0137335	4	7288	214442818	0,023	0,00122156
149	1,2877E+12	61865	871354369	0,018	0,0129514	4,024	7289	214443842	0,024	0,00119396
150	1,2877E+12	61866	871355393	0,01	0,0124682	4,047	7290	214444866	0,023	0,00113309
151	1,2877E+12	61867	871356417	0,008	0,0125152	4,07	7291	214445890	0,023	0,00107602

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
152	1,2877E+12	61868	871357441	0,016	0,0126843	4,093	7292	214446914	0,023	0,00102251
153	1,2877E+12	61869	871358465	0,019	0,0123427	4,116	7293	214447938	0,023	0,00097235
154	1,2877E+12	61870	871359489	0,072	0,0118351	4,14	7294	214448962	0,024	0,00096033
155	1,2877E+12	61871	871360513	0,008	0,0141441	4,164	7295	214449986	0,024	0,00094907
156	1,2877E+12	61872	871361537	0,009	0,0142114	4,187	7296	214451010	0,023	0,0009035
157	1,2877E+12	61873	871362561	0,007	0,0142119	4,209	7297	214452034	0,022	0,00092328
158	1,2877E+12	61874	871363585	0,02	0,0143374	4,235	7298	214453058	0,026	0,00103932
159	1,2877E+12	61875	871364609	0,026	0,0136426	4,256	7299	214454082	0,021	0,00111311
160	1,2877E+12	61876	871365633	0,024	0,0129637	4,279	7300	214455106	0,023	0,00105729
161	1,2877E+12	61877	871366657	0,022	0,0122022	4,302	7301	214456130	0,023	0,00100496
162	1,2877E+12	61878	871367681	0,026	0,0115158	4,325	7302	214457154	0,023	0,00095589
163	1,2877E+12	61879	871368705	0,032	0,0109698	4,349	7303	214458178	0,024	0,0009449
164	1,2877E+12	61880	871369729	0,029	0,0108329	4,372	7304	214459202	0,023	0,00089959
165	1,2877E+12	61881	871370753	0,012	0,0105171	4,395	7305	214460226	0,023	0,00085712
166	1,2877E+12	61882	871371777	0,027	0,0105611	4,418	7306	214461250	0,023	0,00081729
167	1,2877E+12	61883	871372801	0,011	0,0101373	4,442	7307	214462274	0,024	0,00081497
168	1,2877E+12	61884	871373825	0,024	0,0102674	4,465	7308	214463298	0,023	0,00077778

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
169	1,2877E+12	61885	871374849	0,025	0,0096745	4,487	7309	214464322	0,022	0,00080541
170	1,2877E+12	61886	871375873	0,027	0,0091811	4,511	7310	214465346	0,024	0,00080383
171	1,2877E+12	61887	871376897	0,019	0,0088435	4,534	7311	214466370	0,023	0,00076734
172	1,2877E+12	61888	871377921	0,026	0,0085545	4,558	7312	214467394	0,024	0,00076813
173	1,2877E+12	61889	871378945	0,029	0,0081936	4,581	7313	214468418	0,023	0,00073387
174	1,2877E+12	61890	871379969	0,029	0,0080428	4,604	7314	214469442	0,023	0,00070175
175	1,2877E+12	61891	871380993	0,062	0,0079014	4,627	7315	214470466	0,023	0,00067164
176	1,2877E+12	61892	871382017	0,006	0,0098313	4,65	7316	214471490	0,023	0,00064341
177	1,2877E+12	61893	871383041	0,008	0,0102931	4,674	7317	214472514	0,024	0,00065195
178	1,2877E+12	61894	871384065	0,011	0,010601	4,697	7318	214473538	0,023	0,00062495
179	1,2877E+12	61895	871385089	0,017	0,0107022	4,72	7319	214474562	0,023	0,00059964
180	1,2877E+12	61896	871386113	0,034	0,010422	4,743	7320	214475586	0,023	0,00057591
181	1,2877E+12	61897	871387137	0,007	0,0104444	4,767	7321	214476610	0,024	0,00058866
182	1,2877E+12	61898	871388161	0,025	0,0108054	4,793	7322	214477634	0,026	0,00072563
183	1,2877E+12	61899	871389185	0,022	0,0102413	4,813	7323	214478658	0,02	0,00088152
184	1,2877E+12	61900	871390209	0,025	0,0096775	4,836	7324	214479682	0,023	0,00084017
185	1,2877E+12	61901	871391233	0,056	0,0091839	4,859	7325	214480706	0,023	0,00080141

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
186	1,2877E+12	61902	871392257	0,105	0,0106586	4,883	7326	214481730	0,024	0,00080007
187	1,2877E+12	61903	871393281	0,059	0,0151037	4,906	7327	214482754	0,023	0,00076382
188	1,2877E+12	61904	871394305	0,058	0,016396	4,929	7328	214483778	0,023	0,00072983
189	1,2877E+12	61905	871395329	0,025	0,017545	4,952	7329	214484802	0,023	0,00069796
190	1,2877E+12	61906	871396353	0,009	0,0165597	4,976	7330	214485826	0,024	0,00070309
191	1,2877E+12	61907	871397377	0,012	0,0164135	4,999	7331	214486850	0,023	0,00067289
192	1,2877E+12	61908	871398401	0,002	0,0160889	5,022	7332	214487874	0,023	0,00064458
193	1,2877E+12	61909	871399425	0,002	0,0164096	5,045	7333	214488898	0,023	0,00061805
194	1,2877E+12	61910	871400449	0,003	0,0167102	5,067	7334	214489922	0,022	0,00065566
195	1,2877E+12	61911	871401473	0,002	0,0169296	5,092	7335	214490946	0,025	0,00072594
196	1,2877E+12	61912	871402497	0,002	0,0171977	5,115	7336	214491970	0,023	0,00069431
197	1,2877E+12	61913	871403521	0,007	0,0174491	5,138	7337	214492994	0,023	0,00066467
198	1,2877E+12	61914	871404545	0,005	0,0173723	5,161	7338	214494018	0,023	0,00063687
199	1,2877E+12	61915	871405569	0,004	0,0174253	5,185	7339	214495042	0,024	0,00064582
200	1,2877E+12	61916	871406593	0,018	0,0175374	5,208	7340	214496066	0,023	0,0006192
201	1,2877E+12	61917	871407617	0,025	0,0167676	5,231	7341	214497090	0,023	0,00059425
202	1,2877E+12	61918	871408641	0,023	0,0158309	5,254	7342	214498114	0,023	0,00057086

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
203	1,2877E+12	61919	871409665	0,031	0,0148552	5,277	7343	214499138	0,023	0,00054893
204	1,2877E+12	61920	871410689	0,046	0,014413	5,301	7344	214500162	0,024	0,00056337
205	1,2877E+12	61921	871411713	0,028	0,0149359	5,324	7345	214501186	0,023	0,00054191
206	1,2877E+12	61922	871412737	0,016	0,0143012	5,347	7346	214502210	0,023	0,00052179
207	1,2877E+12	61923	871413761	0,011	0,0138586	5,37	7347	214503234	0,023	0,00050292
208	1,2877E+12	61924	871414785	0,009	0,0137562	5,394	7348	214504258	0,024	0,00052024
209	1,2877E+12	61925	871415809	0,02	0,0137852	5,417	7349	214505282	0,023	0,00050147
210	1,2877E+12	61926	871416833	0,027	0,0131249	5,441	7350	214506306	0,024	0,00051888
211	1,2877E+12	61927	871417857	0,03	0,0125408	5,463	7351	214507330	0,022	0,0005627
212	1,2877E+12	61928	871418881	0,014	0,0121808	5,486	7352	214508354	0,023	0,00054128
213	1,2877E+12	61929	871419905	0,024	0,0119957	5,51	7353	214509378	0,024	0,0005562
214	1,2877E+12	61930	871420929	0,024	0,0112947	5,533	7354	214510402	0,023	0,00053519
215	1,2877E+12	61931	871421953	0,024	0,0106376	5,556	7355	214511426	0,023	0,00051548
216	1,2877E+12	61932	871422977	0,072	0,0100215	5,579	7356	214512450	0,023	0,00049701
217	1,2877E+12	61933	871424001	0,021	0,0124439	5,603	7357	214513474	0,024	0,0005147
218	1,2877E+12	61934	871425025	0,012	0,0118049	5,626	7358	214514498	0,023	0,00049628
219	1,2877E+12	61935	871426049	0,008	0,0117683	5,649	7359	214515522	0,023	0,00047901

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
220	1,2877E+12	61936	871427073	0,009	0,011984	5,672	7360	214516546	0,023	0,00046282
221	1,2877E+12	61937	871428097	0,014	0,0121238	5,695	7361	214517570	0,023	0,00044764
222	1,2877E+12	61938	871429121	0,026	0,0119423	5,719	7362	214518594	0,024	0,00046842
223	1,2877E+12	61939	871430145	0,104	0,0113697	5,742	7363	214519618	0,023	0,00045289
224	1,2877E+12	61940	871431169	0,009	0,0157078	5,765	7364	214520642	0,023	0,00043833
225	1,2877E+12	61941	871432193	0,008	0,0156148	5,788	7365	214521666	0,023	0,00042468
226	1,2877E+12	61942	871433217	0,034	0,0155901	5,811	7366	214522690	0,023	0,00041189
227	1,2877E+12	61943	871434241	0,004	0,0152895	5,835	7367	214523714	0,024	0,0004349
228	1,2877E+12	61944	871435265	0,008	0,0155352	5,858	7368	214524738	0,023	0,00042146
229	1,2877E+12	61945	871436289	0,008	0,0155155	5,881	7369	214525762	0,023	0,00040887
230	1,2877E+12	61946	871437313	0,011	0,015497	5,904	7370	214526786	0,023	0,00039706
231	1,2877E+12	61947	871438337	0,044	0,0152922	5,928	7371	214527810	0,024	0,000421
232	1,2877E+12	61948	871439361	0,011	0,0156352	5,951	7372	214528834	0,023	0,00040843
233	1,2877E+12	61949	871440385	0,013	0,0154217	5,975	7373	214529858	0,024	0,00043166
234	1,2877E+12	61950	871441409	0,022	0,0150966	5,997	7374	214530882	0,022	0,00048093
235	1,2877E+12	61951	871442433	0,029	0,0142293	6,02	7375	214531906	0,023	0,00046462
236	1,2877E+12	61952	871443457	0,018	0,0137012	6,044	7376	214532930	0,024	0,00048433

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
237	1,2877E+12	61953	871444481	0,028	0,0131712	6,067	7377	214533954	0,023	0,00046781
238	1,2877E+12	61954	871445505	0,019	0,0126467	6,093	7378	214534978	0,026	0,00061232
239	1,2877E+12	61955	871446529	0,033	0,01212	6,114	7379	214536002	0,021	0,0007128
240	1,2877E+12	61956	871447553	0,021	0,0119738	6,137	7380	214537026	0,023	0,000682
241	1,2877E+12	61957	871448577	0,029	0,0113642	6,16	7381	214538050	0,023	0,00065312
242	1,2877E+12	61958	871449601	0,019	0,0110152	6,183	7382	214539074	0,023	0,00062605
243	1,2877E+12	61959	871450625	0,014	0,0105905	6,206	7383	214540098	0,023	0,00060066
244	1,2877E+12	61960	871451649	0,024	0,0105048	6,229	7384	214541122	0,023	0,00057687
245	1,2877E+12	61961	871452673	0,024	0,009897	6,253	7385	214542146	0,024	0,00058957
246	1,2877E+12	61962	871453697	0,037	0,0093272	6,276	7386	214543170	0,023	0,00056647
247	1,2877E+12	61963	871454721	0,021	0,0096055	6,299	7387	214544194	0,023	0,00054481
248	1,2877E+12	61964	871455745	0,012	0,0091439	6,322	7388	214545218	0,023	0,00052451
249	1,2877E+12	61965	871456769	0,023	0,0092737	6,346	7389	214546242	0,024	0,00054048
250	1,2877E+12	61966	871457793	0,02	0,0087078	6,369	7390	214547266	0,023	0,00052045
251	1,2877E+12	61967	871458817	0,026	0,0083648	6,392	7391	214548290	0,023	0,00050167
252	1,2877E+12	61968	871459841	0,061	0,0080158	6,415	7392	214549314	0,023	0,00048406
253	1,2877E+12	61969	871460865	0,023	0,009876	6,438	7393	214550338	0,023	0,00046755

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
254	1,2877E+12	61970	871461889	0,003	0,0092725	6,462	7394	214551362	0,024	0,00048708
255	1,2877E+12	61971	871462913	0,006	0,0099567	6,487	7395	214552386	0,025	0,00056789
256	1,2877E+12	61972	871463937	0,021	0,0104107	6,509	7396	214553410	0,022	0,00060865
257	1,2877E+12	61973	871464961	0,024	0,0098988	6,533	7397	214554434	0,024	0,00061936
258	1,2877E+12	61974	871465985	0,022	0,0093289	6,555	7398	214555458	0,022	0,0006569
259	1,2877E+12	61975	871467009	0,037	0,008822	6,578	7399	214556482	0,023	0,00062959
260	1,2877E+12	61976	871468033	0,012	0,0091319	6,601	7400	214557506	0,023	0,00060399
261	1,2877E+12	61977	871469057	0,026	0,0092624	6,624	7401	214558530	0,023	0,00057998
262	1,2877E+12	61978	871470081	0,089	0,0088573	6,647	7402	214559554	0,023	0,00055748
263	1,2877E+12	61979	871471105	0,009	0,0124149	6,671	7403	214560578	0,024	0,00057139
264	1,2877E+12	61980	871472129	0,006	0,0125278	6,694	7404	214561602	0,023	0,00054943
265	1,2877E+12	61981	871473153	0,012	0,012821	6,717	7405	214562626	0,023	0,00052883
266	1,2877E+12	61982	871474177	0,008	0,012721	6,74	7406	214563650	0,023	0,00050953
267	1,2877E+12	61983	871475201	0,018	0,0128771	6,764	7407	214564674	0,024	0,00052644
268	1,2877E+12	61984	871476225	0,02	0,0123986	6,787	7408	214565698	0,023	0,00050728
269	1,2877E+12	61985	871477249	0,024	0,0118249	6,809	7409	214566722	0,022	0,00055182
270	1,2877E+12	61986	871478273	0,026	0,0111346	6,832	7410	214567746	0,023	0,00053108

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
271	1,2877E+12	61987	871479297	0,029	0,0106124	6,855	7411	214568770	0,023	0,00051164
272	1,2877E+12	61988	871480321	0,011	0,0103104	6,879	7412	214569794	0,024	0,00052841
273	1,2877E+12	61989	871481345	0,05	0,0104298	6,919	7413	214570818	0,04	0,00154414
274	1,2877E+12	61990	871482369	0,03	0,0114517	6,936	7414	214571842	0,017	0,00183638
275	1,2877E+12	61991	871483393	0,013	0,0111597	6,956	7415	214572866	0,02	0,00192285
276	1,2877E+12	61992	871484417	0,001	0,0111009	6,971	7416	214573890	0,015	0,00231642
277	1,2877E+12	61993	871485441	0,021	0,0117959	6,995	7417	214574914	0,024	0,0022204
278	1,2877E+12	61994	871486465	0,031	0,0111974	7,018	7418	214575938	0,023	0,00209537
279	1,2877E+12	61995	871487489	0,017	0,0109838	7,041	7419	214576962	0,023	0,00197816
280	1,2877E+12	61996	871488513	0,025	0,0106861	7,064	7420	214577986	0,023	0,00186827
281	1,2877E+12	61997	871489537	0,021	0,0101294	7,088	7421	214579010	0,024	0,00180025
282	1,2877E+12	61998	871490561	0,032	0,0096351	7,111	7422	214580034	0,023	0,00170149
283	1,2877E+12	61999	871491585	0,019	0,0095817	7,134	7423	214581058	0,023	0,00160889
284	1,2877E+12	62000	871492609	0,019	0,0092465	7,157	7424	214582082	0,023	0,00152208
285	1,2877E+12	62001	871493633	0,036	0,0089324	7,183	7425	214583106	0,026	0,0016007
286	1,2877E+12	62002	871494657	0,025	0,0091729	7,204	7426	214584130	0,021	0,00163941
287	1,2877E+12	62003	871495681	0,032	0,0087108	7,227	7427	214585154	0,023	0,00155069

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
288	1,2877E+12	62004	871496705	0,026	0,0087151	7,25	7428	214586178	0,023	0,00146752
289	1,2877E+12	62005	871497729	0,008	0,0083442	7,273	7429	214587202	0,023	0,00138955
290	1,2877E+12	62006	871498753	0,019	0,0087739	7,297	7430	214588226	0,024	0,00135145
291	1,2877E+12	62007	871499777	0,016	0,0084893	7,32	7431	214589250	0,023	0,00128074
292	1,2877E+12	62008	871500801	0,025	0,00841	7,343	7432	214590274	0,023	0,00121444
293	1,2877E+12	62009	871501825	0,021	0,0079956	7,367	7433	214591298	0,024	0,00118729
294	1,2877E+12	62010	871502849	0,025	0,0076346	7,389	7434	214592322	0,022	0,00118933
295	1,2877E+12	62011	871503873	0,023	0,0072687	7,413	7435	214593346	0,024	0,00116375
296	1,2877E+12	62012	871504897	0,048	0,0068282	7,436	7436	214594370	0,023	0,00110476
297	1,2877E+12	62013	871505921	0,003	0,0079502	7,459	7437	214595394	0,023	0,00104946
298	1,2877E+12	62014	871506945	0,016	0,008717	7,483	7438	214596418	0,024	0,00103262
299	1,2877E+12	62015	871507969	0,024	0,0086235	7,506	7439	214597442	0,023	0,00098183
300	1,2877E+12	62016	871508993	0,025	0,0081332	7,529	7440	214598466	0,023	0,00093421
301	1,2877E+12	62017	871510017	0,033	0,0077362	7,552	7441	214599490	0,023	0,00088957
302	1,2877E+12	62018	871511041	0,011	0,0078639	7,575	7442	214600514	0,023	0,00084772
303	1,2877E+12	62019	871512065	0,067	0,0081362	7,598	7443	214601538	0,023	0,00080849
304	1,2877E+12	62020	871513089	0,011	0,0103639	7,622	7444	214602562	0,024	0,00080671

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
305	1,2877E+12	62021	871514113	0,012	0,0104799	7,645	7445	214603586	0,023	0,00077004
306	1,2877E+12	62022	871515137	0,008	0,0105262	7,668	7446	214604610	0,023	0,00073566
307	1,2877E+12	62023	871516161	0,019	0,0108195	7,691	7447	214605634	0,023	0,00070342
308	1,2877E+12	62024	871517185	0,026	0,010407	7,716	7448	214606658	0,025	0,00077071
309	1,2877E+12	62025	871518209	0,027	0,0099304	7,738	7449	214607682	0,022	0,00079879
310	1,2877E+12	62026	871519233	0,019	0,009546	7,761	7450	214608706	0,023	0,00076261
311	1,2877E+12	62027	871520257	0,024	0,0092131	7,785	7451	214609730	0,024	0,0007637
312	1,2877E+12	62028	871521281	0,02	0,008686	7,807	7452	214610754	0,022	0,00079222
313	1,2877E+12	62029	871522305	0,033	0,0083444	7,833	7453	214611778	0,026	0,00091646
314	1,2877E+12	62030	871523329	0,014	0,0084341	7,854	7454	214612802	0,021	0,00099793
315	1,2877E+12	62031	871524353	0,033	0,0084832	7,877	7455	214613826	0,023	0,0009493
316	1,2877E+12	62032	871525377	0,094	0,0085643	7,9	7456	214614850	0,023	0,00090372
317	1,2877E+12	62033	871526401	0,016	0,0124528	7,924	7457	214615874	0,024	0,00089599
318	1,2877E+12	62034	871527425	0,007	0,0121257	7,946	7458	214616898	0,022	0,00091624
319	1,2877E+12	62035	871528449	0,003	0,0123816	7,97	7459	214617922	0,024	0,00090773
320	1,2877E+12	62036	871529473	0,005	0,0128715	7,993	7460	214618946	0,023	0,00086474
321	1,2877E+12	62037	871530497	0,011	0,0132058	8,016	7461	214619970	0,023	0,00082444

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
322	1,2877E+12	62038	871531521	0,022	0,0131442	8,04	7462	214620994	0,024	0,00082167
323	1,2877E+12	62039	871532545	0,019	0,0123989	8,063	7463	214622018	0,023	0,00078406
324	1,2877E+12	62040	871533569	0,023	0,0118877	8,086	7464	214623042	0,023	0,0007488
325	1,2877E+12	62041	871534593	0,03	0,0111585	8,109	7465	214624066	0,023	0,00071575
326	1,2877E+12	62042	871535617	0,015	0,0108848	8,132	7466	214625090	0,023	0,00068476
327	1,2877E+12	62043	871536641	0,024	0,0107183	8,156	7467	214626114	0,024	0,00069072
328	1,2877E+12	62044	871537665	0,031	0,0100971	8,179	7468	214627138	0,023	0,00066129
329	1,2877E+12	62045	871538689	0,022	0,0099523	8,202	7469	214628162	0,023	0,00063371
330	1,2877E+12	62046	871539713	0,023	0,0094065	8,228	7470	214629186	0,026	0,00076786
331	1,2877E+12	62047	871540737	0,017	0,0088324	8,252	7471	214630210	0,024	0,00076862
332	1,2877E+12	62048	871541761	0,023	0,0086691	8,272	7472	214631234	0,02	0,00092183
333	1,2877E+12	62049	871542785	0,025	0,008141	8,296	7473	214632258	0,024	0,00091297
334	1,2877E+12	62050	871543809	0,028	0,0077435	8,318	7474	214633282	0,022	0,00093215
335	1,2877E+12	62051	871544833	0,026	0,0075583	8,352	7475	214634306	0,034	0,00154765
336	1,2877E+12	62052	871545857	0,014	0,0072596	8,365	7476	214635330	0,013	0,00208967
337	1,2877E+12	62053	871546881	0,103	0,0073821	8,39	7477	214636354	0,025	0,00207031
338	1,2877E+12	62054	871547905	0,004	0,011907	8,411	7478	214637378	0,021	0,00207967

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
339	1,2877E+12	62055	871548929	0,013	0,0123641	8,435	7479	214638402	0,024	0,00199844
340	1,2877E+12	62056	871549953	0,003	0,0122301	8,458	7480	214639426	0,023	0,00188728
341	1,2877E+12	62057	871550977	0,015	0,0127294	8,481	7481	214640450	0,023	0,00178308
342	1,2877E+12	62058	871552001	0,009	0,0124476	8,504	7482	214641474	0,023	0,00168538
343	1,2877E+12	62059	871553025	0,014	0,0125584	8,527	7483	214642498	0,023	0,00159379
344	1,2877E+12	62060	871554049	0,025	0,0123497	8,55	7484	214643522	0,023	0,00150793
345	1,2877E+12	62061	871555073	0,023	0,0116891	8,574	7485	214644546	0,024	0,00146243
346	1,2877E+12	62062	871556097	0,026	0,0109723	8,597	7486	214645570	0,023	0,00138478
347	1,2877E+12	62063	871557121	0,029	0,0104603	8,62	7487	214646594	0,023	0,00131198
348	1,2877E+12	62064	871558145	0,015	0,0101678	8,643	7488	214647618	0,023	0,00124373
349	1,2877E+12	62065	871559169	0,038	0,010046	8,667	7489	214648642	0,024	0,00121475
350	1,2877E+12	62066	871560193	0,02	0,0103419	8,69	7490	214649666	0,023	0,00115257
351	1,2877E+12	62067	871561217	0,011	0,0098968	8,713	7491	214650690	0,023	0,00109428
352	1,2877E+12	62068	871562241	0,033	0,010042	8,736	7492	214651714	0,023	0,00103964
353	1,2877E+12	62069	871563265	0,013	0,0100256	8,759	7493	214652738	0,023	0,00098841
354	1,2877E+12	62070	871564289	0,036	0,0100377	8,783	7494	214653762	0,024	0,00097539
355	1,2877E+12	62071	871565313	0,011	0,0102091	8,807	7495	214654786	0,024	0,00096318

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
356	1,2877E+12	62072	871566337	0,023	0,0103348	8,829	7496	214655810	0,022	0,00097923
357	1,2877E+12	62073	871567361	0,024	0,0097026	8,852	7497	214656834	0,023	0,00093177
358	1,2877E+12	62074	871568385	0,024	0,009145	8,876	7498	214657858	0,024	0,00092229
359	1,2877E+12	62075	871569409	0,032	0,0086222	8,899	7499	214658882	0,023	0,00087839
360	1,2877E+12	62076	871570433	0,013	0,008632	8,922	7500	214659906	0,023	0,00083724
361	1,2877E+12	62077	871571457	0,036	0,0087313	8,949	7501	214660930	0,027	0,00102117
362	1,2877E+12	62078	871572481	0,011	0,0089843	8,97	7502	214661954	0,021	0,00109609
363	1,2877E+12	62079	871573505	0,024	0,0091866	8,992	7503	214662978	0,022	0,00110383
364	1,2877E+12	62080	871574529	0,034	0,0086611	9,015	7504	214664002	0,023	0,00104859
365	1,2877E+12	62081	871575553	0,013	0,0087936	9,038	7505	214665026	0,023	0,0009968
366	1,2877E+12	62082	871576577	0,032	0,0088827	9,061	7506	214666050	0,023	0,00094825
367	1,2877E+12	62083	871577601	0,016	0,0088763	9,085	7507	214667074	0,024	0,00093773
368	1,2877E+12	62084	871578625	0,025	0,0087728	9,108	7508	214668098	0,023	0,00089287
369	1,2877E+12	62085	871579649	0,044	0,0083357	9,133	7509	214669122	0,025	0,00094832
370	1,2877E+12	62086	871580673	0,015	0,0091135	9,154	7510	214670146	0,021	0,0010278
371	1,2877E+12	62087	871581697	0,058	0,0090577	9,177	7511	214671170	0,023	0,00097731
372	1,2877E+12	62088	871582721	0,007	0,0106653	9,201	7512	214672194	0,024	0,00096498

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
373	1,2877E+12	62089	871583745	0,011	0,0110125	9,224	7513	214673218	0,023	0,00091841
374	1,2877E+12	62090	871584769	0,007	0,0110879	9,247	7514	214674242	0,023	0,00087476
375	1,2877E+12	62091	871585793	0,015	0,0114087	9,27	7515	214675266	0,023	0,00083384
376	1,2877E+12	62092	871586817	0,046	0,0112094	9,293	7516	214676290	0,023	0,00079547
377	1,2877E+12	62093	871587841	0,008	0,0119326	9,317	7517	214677314	0,024	0,0007945
378	1,2877E+12	62094	871588865	0,019	0,012138	9,34	7518	214678338	0,023	0,00075859
379	1,2877E+12	62095	871589889	0,019	0,0116431	9,363	7519	214679362	0,023	0,00072493
380	1,2877E+12	62096	871590913	0,032	0,0111792	9,386	7520	214680386	0,023	0,00069337
381	1,2877E+12	62097	871591937	0,017	0,0110293	9,413	7521	214681410	0,027	0,00088629
382	1,2877E+12	62098	871592961	0,023	0,0107287	9,433	7522	214682434	0,02	0,00103214
383	1,2877E+12	62099	871593985	0,041	0,0100719	9,457	7523	214683458	0,024	0,00101638
384	1,2877E+12	62100	871595009	0,007	0,0105536	9,479	7524	214684482	0,022	0,00102911
385	1,2877E+12	62101	871596033	0,044	0,0109078	9,502	7525	214685506	0,023	0,00097854
386	1,2877E+12	62102	871597057	0,006	0,0115248	9,526	7526	214686530	0,024	0,00096613
387	1,2877E+12	62103	871598081	0,017	0,0118807	9,549	7527	214687554	0,023	0,00091949
388	1,2877E+12	62104	871599105	0,022	0,0115269	9,572	7528	214688578	0,023	0,00087577
389	1,2877E+12	62105	871600129	0,027	0,0108828	9,595	7529	214689602	0,023	0,00083478

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
390	1,2877E+12	62106	871601153	0,039	0,0104388	9,619	7530	214690626	0,024	0,00083136
391	1,2877E+12	62107	871602177	0,017	0,0107727	9,642	7531	214691650	0,023	0,00079315
392	1,2877E+12	62108	871603201	0,018	0,0104881	9,665	7532	214692674	0,023	0,00075733
393	1,2877E+12	62109	871604225	0,016	0,0101589	9,688	7533	214693698	0,023	0,00072374
394	1,2877E+12	62110	871605249	0,025	0,0099752	9,711	7534	214694722	0,023	0,00069225
395	1,2877E+12	62111	871606273	0,021	0,009463	9,735	7535	214695746	0,024	0,00069774
396	1,2877E+12	62112	871607297	0,024	0,0090103	9,758	7536	214696770	0,023	0,00066788
397	1,2877E+12	62113	871608321	0,034	0,0084959	9,781	7537	214697794	0,023	0,00063988
398	1,2877E+12	62114	871609345	0,012	0,0086387	9,804	7538	214698818	0,023	0,00061364
399	1,2877E+12	62115	871610369	0,026	0,0088	9,828	7539	214699842	0,024	0,00062404
400	1,2877E+12	62116	871611393	0,02	0,0084237	9,852	7540	214700866	0,024	0,00063379
401	1,2877E+12	62117	871612417	0,027	0,0080985	9,874	7541	214701890	0,022	0,00067042
402	1,2877E+12	62118	871613441	0,038	0,0078286	9,897	7542	214702914	0,023	0,00064227
403	1,2877E+12	62119	871614465	0,009	0,0082631	9,92	7543	214703938	0,023	0,00061587
404	1,2877E+12	62120	871615489	0,033	0,0086354	9,944	7544	214704962	0,024	0,00062614
405	1,2877E+12	62121	871616513	0,026	0,0087069	9,969	7545	214705986	0,025	0,00069825
406	1,2877E+12	62122	871617537	0,043	0,0083365	9,99	7546	214707010	0,021	0,00079336

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
407	1,2877E+12	62123	871618561	0,012	0,0090517	10,013	7547	214708034	0,023	0,00075752
408	1,2877E+12	62124	871619585	0,037	0,0091872	10,037	7548	214709058	0,024	0,00075893
409	1,2877E+12	62125	871620609	0,001	0,0094743	10,06	7549	214710082	0,023	0,00072524
410	1,2877E+12	62126	871621633	0,006	0,0102709	10,083	7550	214711106	0,023	0,00069366
411	1,2877E+12	62127	871622657	0,03	0,0107052	10,107	7551	214712130	0,024	0,00069906
412	1,2877E+12	62128	871623681	0,018	0,0104599	10,129	7552	214713154	0,022	0,00073162
413	1,2877E+12	62129	871624705	0,033	0,0101324	10,153	7553	214714178	0,024	0,00073465
414	1,2877E+12	62130	871625729	0,028	0,0101104	10,177	7554	214715202	0,024	0,00073748
415	1,2877E+12	62131	871626753	0,023	0,0097772	10,199	7555	214716226	0,022	0,00076764
416	1,2877E+12	62132	871627777	0,024	0,0091799	10,222	7556	214717250	0,023	0,00073341
417	1,2877E+12	62133	871628801	0,011	0,0086549	10,245	7557	214718274	0,023	0,00070132
418	1,2877E+12	62134	871629825	0,025	0,0088777	10,269	7558	214719298	0,024	0,00070624
419	1,2877E+12	62135	871630849	0,035	0,0084341	10,292	7559	214720322	0,023	0,00067584
420	1,2877E+12	62136	871631873	0,086	0,0086432	10,315	7560	214721346	0,023	0,00064735
421	1,2877E+12	62137	871632897	0,005	0,0120268	10,338	7561	214722370	0,023	0,00062064
422	1,2877E+12	62138	871633921	0,006	0,0124139	10,362	7562	214723394	0,024	0,0006306
423	1,2877E+12	62139	871634945	0,016	0,0127142	10,385	7563	214724418	0,023	0,00060494

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
424	1,2877E+12	62140	871635969	0,003	0,0123708	10,41	7564	214725442	0,025	0,00067838
425	1,2877E+12	62141	871636993	0,007	0,0128614	10,431	7565	214726466	0,021	0,00077473
426	1,2877E+12	62142	871638017	0,021	0,0130713	10,454	7566	214727490	0,023	0,00074006
427	1,2877E+12	62143	871639041	0,027	0,0123931	10,481	7567	214728514	0,027	0,00093006
428	1,2877E+12	62144	871640065	0,028	0,0118548	10,503	7568	214729538	0,022	0,00094817
429	1,2877E+12	62145	871641089	0,018	0,0114126	10,524	7569	214730562	0,021	0,00102766
430	1,2877E+12	62146	871642113	0,022	0,0110256	10,547	7570	214731586	0,023	0,00097718
431	1,2877E+12	62147	871643137	0,024	0,0104127	10,571	7571	214732610	0,024	0,00096486
432	1,2877E+12	62148	871644161	0,053	0,0098107	10,594	7572	214733634	0,023	0,0009183
433	1,2877E+12	62149	871645185	0,021	0,0110588	10,617	7573	214734658	0,023	0,00087465
434	1,2877E+12	62150	871646209	0,004	0,0105063	10,64	7574	214735682	0,023	0,00083374
435	1,2877E+12	62151	871647233	0,016	0,0110509	10,663	7575	214736706	0,023	0,00079537
436	1,2877E+12	62152	871648257	0,024	0,0108115	10,687	7576	214737730	0,024	0,00079442
437	1,2877E+12	62153	871649281	0,02	0,0101845	10,71	7577	214738754	0,023	0,00075851
438	1,2877E+12	62154	871650305	0,023	0,0097493	10,734	7578	214739778	0,024	0,00075986
439	1,2877E+12	62155	871651329	0,033	0,0091537	10,756	7579	214740802	0,022	0,00078861
440	1,2877E+12	62156	871652353	0,024	0,0091928	10,78	7580	214741826	0,024	0,00078808

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
441	1,2877E+12	62157	871653377	0,011	0,008667	10,803	7581	214742850	0,023	0,00075257
442	1,2877E+12	62158	871654401	0,028	0,0088891	10,827	7582	214743874	0,024	0,00075429
443	1,2877E+12	62159	871655425	0,019	0,0086323	10,849	7583	214744898	0,022	0,00078339
444	1,2877E+12	62160	871656449	0,025	0,0083565	10,872	7584	214745922	0,023	0,00074818
445	1,2877E+12	62161	871657473	0,022	0,0079455	10,896	7585	214746946	0,024	0,00075017
446	1,2877E+12	62162	871658497	0,026	0,0075251	10,923	7586	214747970	0,027	0,00093954
447	1,2877E+12	62163	871659521	0,025	0,0072286	10,943	7587	214748994	0,02	0,00108206
448	1,2877E+12	62164	871660545	0,019	0,006888	10,965	7588	214750018	0,022	0,00109068
449	1,2877E+12	62165	871661569	0,025	0,0067213	10,989	7589	214751042	0,024	0,00107127
450	1,2877E+12	62166	871662593	0,022	0,0064124	11,012	7590	214752066	0,023	0,00101806
451	1,2877E+12	62167	871663617	0,03	0,0060879	11,035	7591	214753090	0,023	0,00096818
452	1,2877E+12	62168	871664641	0,032	0,0061312	11,058	7592	214754114	0,023	0,00092141
453	1,2877E+12	62169	871665665	0,038	0,0062967	11,082	7593	214755138	0,024	0,00091258
454	1,2877E+12	62170	871666689	0,057	0,0068269	11,105	7594	214756162	0,023	0,00086929
455	1,2877E+12	62171	871667713	0,014	0,0085115	11,128	7595	214757186	0,023	0,00082871
456	1,2877E+12	62172	871668737	0,005	0,0085558	11,151	7596	214758210	0,023	0,00079066
457	1,2877E+12	62173	871669761	0,007	0,0091598	11,174	7597	214759234	0,023	0,00075499

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
458	1,2877E+12	62174	871670785	0,011	0,0096011	11,199	7598	214760258	0,025	0,00081906
459	1,2877E+12	62175	871671809	0,018	0,0097647	11,221	7599	214761282	0,022	0,00084411
460	1,2877E+12	62176	871672833	0,023	0,0094807	11,244	7600	214762306	0,023	0,0008051
461	1,2877E+12	62177	871673857	0,022	0,0089019	11,267	7601	214763330	0,023	0,00076853
462	1,2877E+12	62178	871674881	0,021	0,0084218	11,29	7602	214764354	0,023	0,00073424
463	1,2877E+12	62179	871675905	0,023	0,0080342	11,314	7603	214765378	0,024	0,00073711
464	1,2877E+12	62180	871676929	0,042	0,0075458	11,338	7604	214766402	0,024	0,00073979
465	1,2877E+12	62181	871677953	0,019	0,0082479	11,36	7605	214767426	0,022	0,0007698
466	1,2877E+12	62182	871678977	0,018	0,0079962	11,383	7606	214768450	0,023	0,00073544
467	1,2877E+12	62183	871680001	0,015	0,0078226	11,406	7607	214769474	0,023	0,00070322
468	1,2877E+12	62184	871681025	0,024	0,0078475	11,43	7608	214770498	0,024	0,00070802
469	1,2877E+12	62185	871682049	0,033	0,0074058	11,453	7609	214771522	0,023	0,00067752
470	1,2877E+12	62186	871683073	0,018	0,0075542	11,476	7610	214772546	0,023	0,00064892
471	1,2877E+12	62187	871684097	0,03	0,0074083	11,499	7611	214773570	0,023	0,00062211
472	1,2877E+12	62188	871685121	0,01	0,007369	11,523	7612	214774594	0,024	0,00063198
473	1,2877E+12	62189	871686145	0,044	0,0077347	11,547	7613	214775618	0,024	0,00064123
474	1,2877E+12	62190	871687169	0,029	0,00855	11,569	7614	214776642	0,022	0,0006774

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
475	1,2877E+12	62191	871688193	0,018	0,0083769	11,593	7615	214777666	0,024	0,00068382
476	1,2877E+12	62192	871689217	0,016	0,0081796	11,615	7616	214778690	0,022	0,00071733
477	1,2877E+12	62193	871690241	0,009	0,0081196	11,639	7617	214779714	0,024	0,00072125
478	1,2877E+12	62194	871691265	0,026	0,0085009	11,662	7618	214780738	0,023	0,00068992
479	1,2877E+12	62195	871692289	0,021	0,0081433	11,685	7619	214781762	0,023	0,00066054
480	1,2877E+12	62196	871693313	0,025	0,0077731	11,71	7620	214782786	0,025	0,00073051
481	1,2877E+12	62197	871694337	0,023	0,0073986	11,732	7621	214783810	0,022	0,0007611
482	1,2877E+12	62198	871695361	0,021	0,0069499	11,755	7622	214784834	0,023	0,00072728
483	1,2877E+12	62199	871696385	0,037	0,0066543	11,778	7623	214785858	0,023	0,00069557
484	1,2877E+12	62200	871697409	0,012	0,0070996	11,801	7624	214786882	0,023	0,00066585
485	1,2877E+12	62201	871698433	0,026	0,0073572	11,824	7625	214787906	0,023	0,00063798
486	1,2877E+12	62202	871699457	0,018	0,0070711	11,848	7626	214788930	0,024	0,00064686
487	1,2877E+12	62203	871700481	0,027	0,0069554	11,872	7627	214789954	0,024	0,00065518
488	1,2877E+12	62204	871701505	0,113	0,0067569	11,894	7628	214790978	0,022	0,00069048
489	1,2877E+12	62205	871702529	0,32	0,0119459	12,434	7629	214792002	0,54	0,03294608
490	1,2877E+12	62206	871703553	0,049	0,029748	12,434	7630	214793026	0	0,0323382
491	1,2877E+12	62207	871704577	0,017	0,0295	12,435	7631	214794050	0,001	0,03170581

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
492	1,2877E+12	62208	871705601	0,031	0,028045	12,436	7632	214795074	0,001	0,03111294
493	1,2877E+12	62209	871706625	0,018	0,0267784	12,436	7633	214796098	0	0,03061963
494	1,2877E+12	62210	871707649	0,008	0,025431	12,437	7634	214797122	0,001	0,03009465
495	1,2877E+12	62211	871708673	0,018	0,0247928	12,437	7635	214798146	0	0,02966498
496	1,2877E+12	62212	871709697	0,005	0,0235695	12,438	7636	214799170	0,001	0,02919967
497	1,2877E+12	62213	871710721	0,008	0,0232352	12,439	7637	214800194	0,001	0,02876343
498	1,2877E+12	62214	871711745	0,005	0,0227342	12,441	7638	214801218	0,002	0,02829197
499	1,2877E+12	62215	871712769	0,005	0,0224521	12,441	7639	214802242	0	0,02797497
500	1,2877E+12	62216	871713793	0,004	0,0221876	12,442	7640	214803266	0,001	0,02761528
501	1,2877E+12	62217	871714817	0,004	0,0220021	12,46	7646	214809410	0,018	0,03347181
502	1,2877E+12	62218	871715841	0,003	0,0218282	12,46	7647	214810434	0	0,03283107
503	1,2877E+12	62219	871716865	0,004	0,0217277	12,461	7648	214811458	0,001	0,03216787
504	1,2877E+12	62220	871717889	0,004	0,021571	12,461	7649	214812482	0	0,03160863
505	1,2877E+12	62221	871718913	0,004	0,021424	12,462	7650	214813506	0,001	0,03102183
506	1,2877E+12	62222	871719937	0,009	0,0212863	12,463	7651	214814530	0,001	0,03047172
507	1,2877E+12	62223	871720961	0,008	0,0208446	12,463	7652	214815554	0	0,03001848
508	1,2877E+12	62224	871721985	0,013	0,0204931	12,476	7653	214816578	0,013	0,02878107

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
509	1,2877E+12	62225	871723009	0,008	0,019851	12,498	7654	214817602	0,022	0,0270585
510	1,2877E+12	62226	871724033	0,01	0,0195616	12,521	7655	214818626	0,023	0,02538109
511	1,2877E+12	62227	871725057	0,007	0,0191652	12,544	7656	214819650	0,023	0,02380852
512	1,2877E+12	62228	871726081	0,013	0,0189811	12,567	7657	214820674	0,023	0,02233424
513	1,2877E+12	62229	871727105	0,01	0,0184336	12,591	7658	214821698	0,024	0,0209871
514	1,2877E+12	62230	871728129	0,014	0,0181077	12,614	7659	214822722	0,023	0,01968915
515	1,2877E+12	62231	871729153	0,021	0,0175522	12,637	7660	214823746	0,023	0,01847233
516	1,2877E+12	62232	871730177	0,019	0,016594	12,66	7661	214824770	0,023	0,01733156
517	1,2877E+12	62233	871731201	0,007	0,0158206	12,684	7662	214825794	0,024	0,01629709
518	1,2877E+12	62234	871732225	0,009	0,0158456	12,708	7663	214826818	0,024	0,01532727
519	1,2877E+12	62235	871733249	0,008	0,015744	12,73	7664	214827842	0,022	0,01444556
520	1,2877E+12	62236	871734273	0,071	0,0157112	12,753	7665	214828866	0,023	0,01355646
521	1,2877E+12	62237	871735297	0,006	0,0177155	12,776	7666	214829890	0,023	0,01272293
522	1,2877E+12	62238	871736321	0,008	0,0176845	12,8	7667	214830914	0,024	0,0119765
523	1,2877E+12	62239	871737345	0,015	0,0175305	12,823	7668	214831938	0,023	0,01124172
524	1,2877E+12	62240	871738369	0,011	0,0169486	12,855	7669	214832962	0,032	0,01108786
525	1,2877E+12	62241	871739393	0,011	0,016653	12,873	7670	214833986	0,018	0,01072112

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
526	1,2877E+12	62242	871740417	0,01	0,016376	12,893	7671	214835010	0,02	0,0102523
527	1,2877E+12	62243	871741441	0,018	0,0161787	12,916	7672	214836034	0,023	0,00962527
528	1,2877E+12	62244	871742465	0,024	0,0154938	12,939	7673	214837058	0,023	0,00903744
529	1,2877E+12	62245	871743489	0,027	0,0145742	12,962	7674	214838082	0,023	0,00848635
530	1,2877E+12	62246	871744513	0,02	0,0138996	12,985	7675	214839106	0,023	0,0079697
531	1,2877E+12	62247	871745537	0,02	0,0132321	13,009	7676	214840130	0,024	0,00752035
532	1,2877E+12	62248	871746561	0,024	0,0126063	13,033	7677	214841154	0,024	0,00709908
533	1,2877E+12	62249	871747585	0,036	0,0118672	13,055	7678	214842178	0,022	0,00673163
534	1,2877E+12	62250	871748609	0,013	0,0119242	13,078	7679	214843202	0,023	0,00632465
535	1,2877E+12	62251	871749633	0,033	0,0118177	13,102	7680	214844226	0,024	0,00597811
536	1,2877E+12	62252	871750657	0,034	0,0116904	13,125	7681	214845250	0,023	0,00561823
537	1,2877E+12	62253	871751681	0,013	0,0116335	13,148	7682	214846274	0,023	0,00528084
538	1,2877E+12	62254	871752705	0,01	0,0115451	13,171	7683	214847298	0,023	0,00496453
539	1,2877E+12	62255	871753729	0,028	0,0116498	13,194	7684	214848322	0,023	0,004668
540	1,2877E+12	62256	871754753	0,022	0,0112204	13,218	7685	214849346	0,024	0,004425
541	1,2877E+12	62257	871755777	0,023	0,0105954	13,241	7686	214850370	0,023	0,00416218
542	1,2877E+12	62258	871756801	0,023	0,0099469	13,264	7687	214851394	0,023	0,00391579

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
543	1,2877E+12	62259	871757825	0,021	0,009339	13,287	7688	214852418	0,023	0,0036848
544	1,2877E+12	62260	871758849	0,025	0,0088941	13,311	7689	214853442	0,024	0,00350326
545	1,2877E+12	62261	871759873	0,036	0,0084494	13,334	7690	214854466	0,023	0,00329805
546	1,2877E+12	62262	871760897	0,01	0,0087201	13,359	7691	214855490	0,025	0,00320318
547	1,2877E+12	62263	871761921	0,025	0,0090013	13,38	7692	214856514	0,021	0,00314172
548	1,2877E+12	62264	871762945	0,045	0,00855	13,406	7693	214857538	0,026	0,00311912
549	1,2877E+12	62265	871763969	0,011	0,0093769	13,427	7694	214858562	0,021	0,00306292
550	1,2877E+12	62266	871764993	0,011	0,0095546	13,459	7695	214859586	0,032	0,00342024
551	1,2877E+12	62267	871766017	0,028	0,0097212	13,473	7696	214860610	0,014	0,00378272
552	1,2877E+12	62268	871767041	0,025	0,0094123	13,496	7697	214861634	0,023	0,00356005
553	1,2877E+12	62269	871768065	0,083	0,0089353	13,52	7698	214862658	0,024	0,0033863
554	1,2877E+12	62270	871769089	0,008	0,0121131	13,543	7699	214863682	0,023	0,0031884
555	1,2877E+12	62271	871770113	0,015	0,0123073	13,567	7700	214864706	0,024	0,00303788
556	1,2877E+12	62272	871771137	0,021	0,0120518	13,589	7701	214865730	0,022	0,00292426
557	1,2877E+12	62273	871772161	0,015	0,0114373	13,612	7702	214866754	0,023	0,00275524
558	1,2877E+12	62274	871773185	0,003	0,0112363	13,636	7703	214867778	0,024	0,00263179
559	1,2877E+12	62275	871774209	0,011	0,0117977	13,659	7704	214868802	0,023	0,00248105

ESCENARIO 1						ESCENARIO 2				
No.	Tiempo	Secuencia	TimeStamp	Retardo	Jitter	Time	Secuencia	TimeStamp	Retardo	Jitter
560	1,2877E+12	62276	871775233	0,022	0,0118241	13,682	7705	214869826	0,023	0,00233973
PROMEDIO				0,023171	0,0108934	PROMEDIO			0,024432143	0,03301771

Importante:

En las tablas 20 y 21:

- Los tiempos están dados en segundos.
- Por consecuencia el jitter y el retardo también está en segundos.

4.3 Benchmarking

Los siguientes gráficos se han realizado en los dos escenarios, los que demostraran las diferencias que existen entre los mismos.

4.4 Jitter

4.4.1 Paquetes de Audio

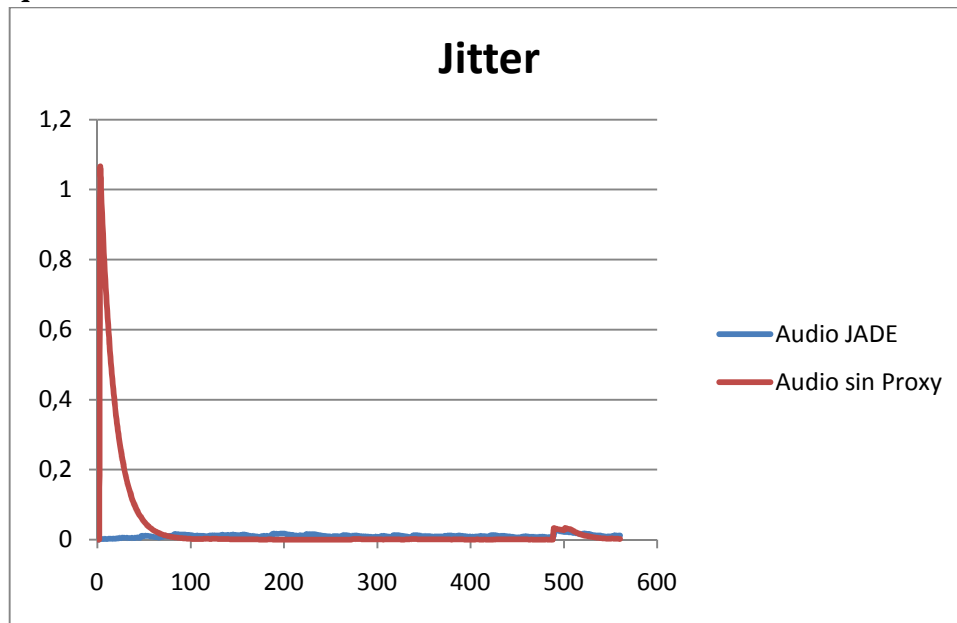


Ilustración 4.2 - Jitter Paquetes de Video

4.4.2 Paquetes de Video

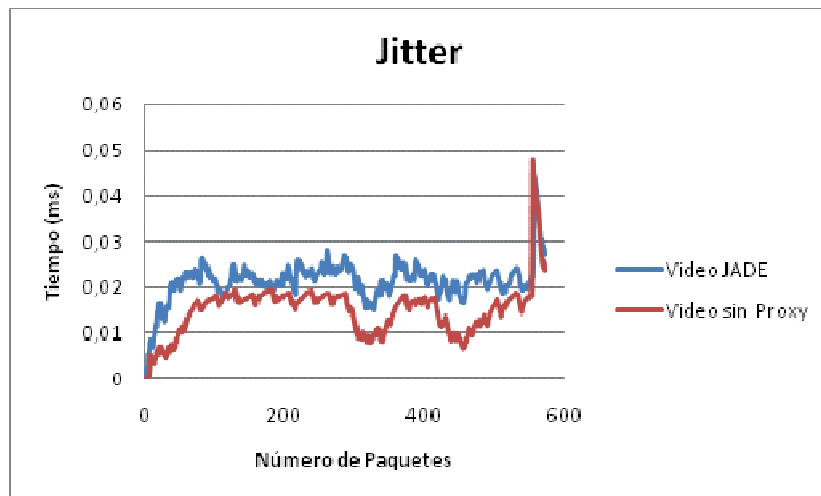


Ilustración 4.3 - Jitter Paquetes de Video

4.5 Retardo

4.5.1 Paquetes de Audio

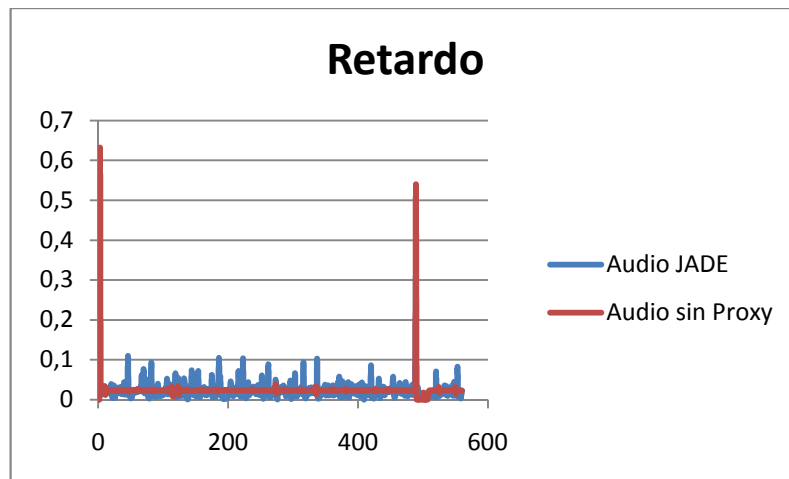


Ilustración 4.4 - Retardo Paquetes de Audio

4.5.2 Paquetes de Video

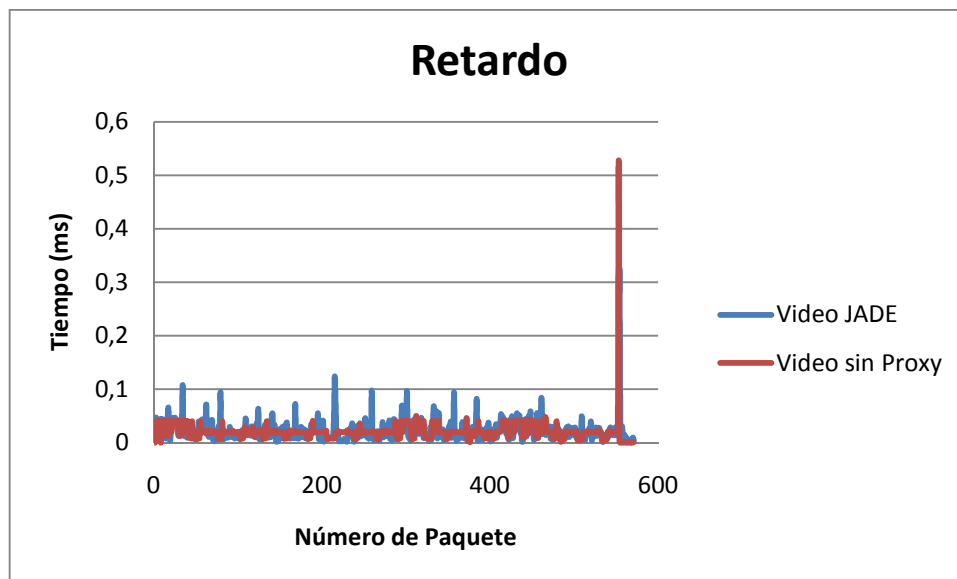


Ilustración 4.5 - Retardo Paquetes de Video

Tanto los gráficos del Jitter y del Retardo nos demuestran que Jade lleva paquetes más grandes, pues poseen mayor información como es la de control, pero si se puede observar mayor estabilidad en la comunicación, ya que no existe desconexiones, o gran pérdida de paquetes como sucede con RTSP sin jade.

4.6 Pérdida de Paquetes

4.6.1 Paquetes de Audio

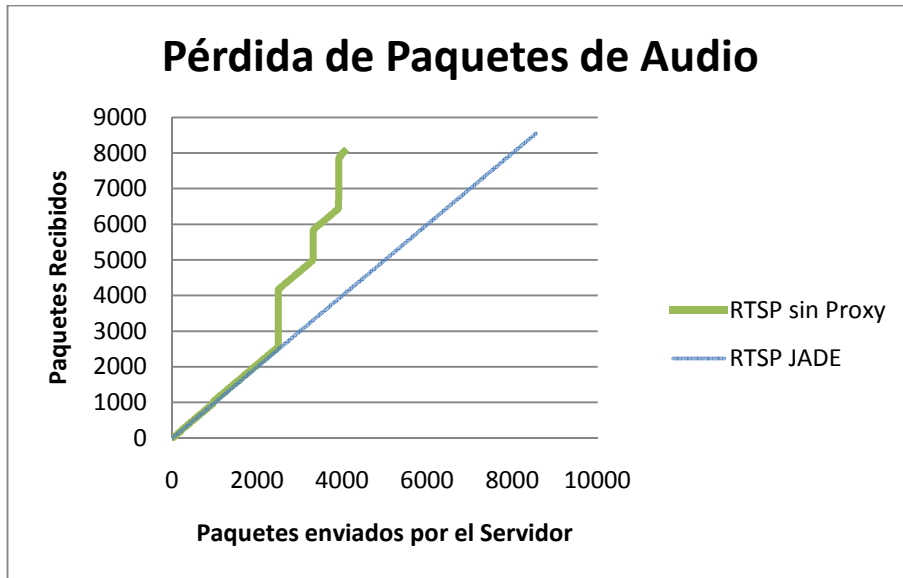


Ilustración 4.6 - Pérdida de Paquetes Audio

4.6.2 Paquetes de video

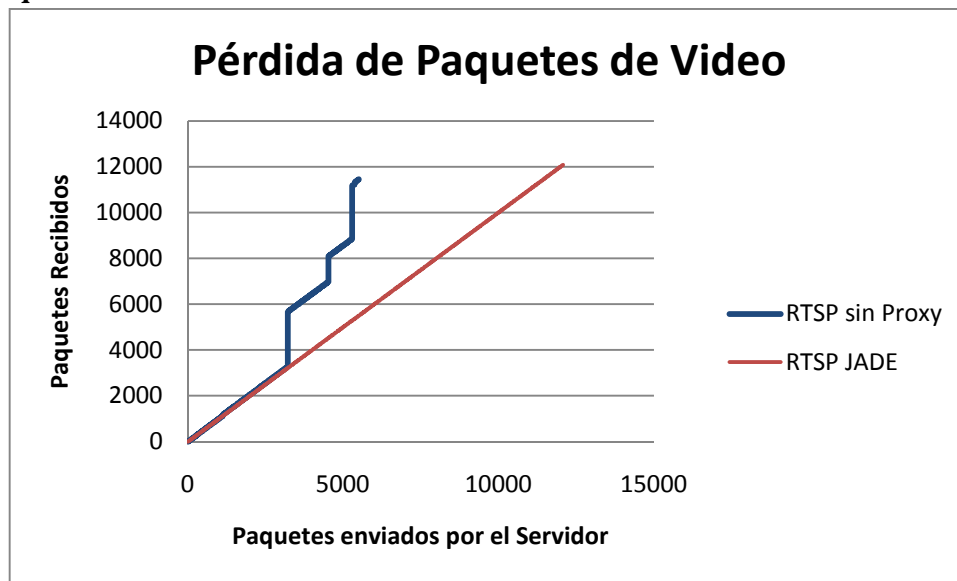


Ilustración 4.7 - Pérdida de Paquetes Video

Estos gráficos se realizaron, simulando una desconexión en la red, para verificar y comprobar la efectividad de JADE al momento recuperar los paquetes después de una desconexión.

Representan la transmisión de los paquetes de audio y video en situaciones de desconexiones, demostrando que las pérdidas de paquetes en el software sin proxy son representativas, al contrario de lo que sucede con JADE, ya que en el video las pérdidas son imperceptibles y en el audio no existe pérdida alguna. Demostrando que la aplicación desarrollada cumple con sus objetivos, y que los problemas ya depende de factores externos a la misma, siendo uno de sus principales limitantes la arquitectura del PvPlayer.

4.7 Ajuste y reconfiguración.

4.7.1 Obtención real del tamaño de los paquetes recibidos del servidor VLC.

Al momento de realizar las pruebas con el software Jade –Streaming versión 0.9, se utilizó la herramienta wireshark en donde pudimos observar que todos los paquetes se estaban transmitiendo con un tamaño de 1500 bytes, lo cual hizo que se revise el archivo de captura

de tráfico de red, y pudimos constatar que la mayoría de los paquetes se completaban con ceros, muchas veces casi vacíos pero su tamaño era constante, es decir 1500 bytes

Luego de obtener estos resultados, se revisó el código del APS, y se logró localizar lo siguiente:

```
@Override
public void action() {
    byte[] contenido=new byte[1500];
    this.paquete=new DatagramPacket(contenido, contenido.length);
    try {
        socket.receive(paquete);
        mensaje.setPerformative(tipo);
        mensaje.setByteSequenceContent(paquete.getData());
        myAgent.send(mensaje);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Ilustración 4.8 - Error de Código

En donde se pensaba que se obtenía el valor real del paquete, pero no era así, se estaba siempre enviando con 1500 bytes, valor con el que se inicializó.

EL cambio que se realizó es el siguiente:

```

@Override
public void action() {
    byte[] contenido=new byte[1500];
    this.paquete=new DatagramPacket(contenido, contenido.length);
    try {
        socket.receive(paquete);
        mensaje.setPerformative(tipo);
        baos=new ByteArrayOutputStream();
        baos.write(paquete.getData(), 0, paquete.getLength());
        mensaje.setByteSequenceContent(baos.toByteArray());
        myAgent.send(mensaje);
        baos.reset();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Ilustración 4.9 - Corrección

En donde se utilizó un manejador de Bytes (ByteArrayOutputStream), para poder obtener el contenido útil del array y de esta manera enviar en el mensaje JADE.

Luego de esta implementación se obtuvo lo esperado, ahora si los paquetes se están enviando con el tamaño real. Este fue un error de programación que generalmente se comete, pero que fue solucionado con éxito, ya que después de esto el resultado fue el Software Streaming – Jade versión 1, en donde el video y el audio se pasa sin distorsión alguna.

Con este ajuste, se pudo llegar a arreglar el error de la distorsión de la imagen y el video, ya que estaba utilizando ancho de banda innecesario.

4.7.2 Sincronizar Timestamp para mejoras a futuro

Luego de haber realizado varias pruebas, el software desarrollado soporta 30 segundos de desconexión sin ningún problema, es por esto que luego de haber investigado se ha decidido

variar el timestamp, para descubrir el motivo por el cual al salir del área de cobertura por más de 45 segundos no regresa el audio ni el video, pese a que los paquetes si llegan a su destino inmediatamente después de la desconexión.

Para esto se analizó varios archivos de captura de tráfico de red, observando que existía un retardo promedio en los paquetes de audio de 23 ms y en los de video de 20 ms, llegando a la conclusión que en esos intervalos de tiempo se envía un paquete. Luego de esto se investigó el intervalo de cada timestamp del audio y del video según el códec, teniendo 3600 y 1020 respectivamente. Y se decidió hacer una suma incremental del valor por defecto hasta que se reanude la conexión, para finalmente aumentarles a los paquetes que lleguen posteriormente. Es decir si la desconexión duró un segundo se sumará 50 veces 3600, teniendo el acumulado de 180.000, en el caso del video, y en el caso del audio 50 veces 1024, teniendo un acumulado de 51.200.

Luego de estos cambios, se volvió a probar el software y funciona sin ningún problema por más de un minuto pero con la transmisión de audio y video por separado, lo cual supera a los resultados anteriores, y mejora la funcionalidad de la aplicación. Llegando a la conclusión que el PvPlayer desecha los paquetes cuando estos tienen un retardo excesivo, situación por la cual la aplicación no responde.

Esta modificación no ha sido tomada para la versión actual del software, debido a que esto sería un inicio de una mejora a futuro, porque solo funciona en el escenario planteado, más no en una transmisión de audio y video simultáneamente, ya que debería existir un algoritmo que se ajuste a las todas las condiciones posibles. Este resultado abre las puertas a JADE en el video streaming, ampliando el alcance de esta herramienta y observando que aquí el

problema no solo depende de JADE sino también de los protocolos y del reproductor utilizado.

4.8 Obtención de resultados analíticos y discusión.

Los resultados obtenidos es la reproducción de audio y video, con calidad al 100%. Al momento de salir del área de cobertura por aproximadamente 30 segundos, los paquetes tanto de audio y video se recuperan de manera exitosa, sin pérdida de ningún paquete, pero si con una demora al momento de reconectarse, debido a que los paquetes que se han quedado guardados en el buffer deben ser despachados y recalculados su retardo.

En un intervalo de tiempo de 30-45 segundos fuera del área de conexión, los paquetes de audio y video llegan a su destino, el audio se logra escuchar, pero el video se atrasa y muchas veces no se reproduce, ya que el buffer es muy pequeño y no logra recalcular totalmente el frame.

A un tiempo mayor de 45 segundos, los paquetes de audio y video llegan al APC, pero el PvPlayer no los reproduce. Motivo por el cual, se decidió realizar más pruebas para comprobar la efectividad del sistema, y saber si el software desarrollado estaba fallando o dependía de otros factores ajenos al mismo. El resultado de estas últimas pruebas explicadas en el tema Ajuste y Reconfiguración, afirmaron que el motivo del mal funcionamiento de los 30 segundos, se debía al timestamp (marca de tiempo), ya que el PvPlayer decide desplazar los paquetes que tienen un gran retardo y no tomarlos en cuenta, produciendo que la aplicación ya no responda.

Así también cuando existe una desconexión, muchas veces se puede observar que el video se demora un poco más en regresar que el audio, existiendo una desincronización, esto se debe

porque un paquete de audio es igual a un frame, en cambio en el video, un frame son varios paquetes, teniendo que esperar que lleguen todos los paquetes para que se complete el frame y poder reproducir. Además de esto los paquetes de audio oscilan en un tamaño de 97 a 450 bytes, en cambio los de video de 300 a 1400 bytes, haciendo que el audio utilice menos ancho de banda que el video.

Luego de estas pruebas se ha realizado con el PvPlayer el streaming sin proxy y a los 30 segundos se recupera de manera inmediata la transmisión, pero se pierden todos los paquetes que se han emitido en el lapso de tiempo de la desconexión. Pasado de los 30 segundos la conexión se corta y ya no regresa ni el audio ni el video.

4.9 Determinar condiciones, factores y ajustes a fin de conseguir que la interrupción sea mínima.

De acuerdo a los resultados obtenidos se concluye que para obtener una mayor calidad de video streaming, es necesario que:

- El usuario final debe tener una conexión de banda ancha con tecnología de acceso ADSL con una velocidad mínima de 600 kbps.
- El proveedor de servicios de streaming debe en lo posible realizar sus emisiones de video en tiempo real.
- Configurar de manera correcta todos los parámetros del servidor RTSP.
- Utilizar los códec adecuados para cada dispositivo móvil.
- El usuario final se encuentre en un área de cobertura donde la intensidad de la señal sea buena.

CAPITULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones Generales

El protocolo RTSP trabaja sobre TCP, que es el encargado de iniciar y mantener la sesión de usuario en una transmisión de video streaming. Así también se presenta el protocolo RTP en el cual se envía los paquetes, este protocolo trabaja sobre UDP, el mismo tiene como desventaja la pérdida en la transmisión y el desorden en la llegada de los paquetes. Debido a estos inconvenientes existe un protocolo de control llamado RTCP, siendo la función de este el de enviar estadísticas del retardo de los paquetes entre el emisor y el receptor, así como la cantidad de paquetes perdidos y otros datos que servirán para ser correctivos durante la transmisión, pero aun así este protocolo no controla ni evita la desconexión.

Para esto se presenta un mecanismo basado en agentes de software capaz de recuperar automáticamente sesiones de video streaming, ocurridas por una desconexión física de los dispositivos móviles Android, mediante el uso de las plataformas JADE-LEAP y JADE-ANDROID.

El sistema resultante permite que el cliente recupere los cuadros de video perdidos al experimentar quiebres de comunicación, y que continúe visualizando el video desde el punto en que se generó una desconexión en un tiempo limitado.

Con este aplicativo se disminuye los efectos de las desconexiones intermitentes que sufren los dispositivos móviles. Producto de esto, no se pierde la información ya que la sesión no es cancelada y todos los cuadros video o imagen que componen el video, son guardados hasta

que el cliente recupere su conexión para reenviar los cuadros faltantes. Cabe aclarar que se debe imponer un tiempo máximo y razonable para la reconexión, para evitar desperdiciar recursos en el servidor.

Para verificar el funcionamiento se han efectuado varias pruebas tanto en posiciones dentro de cobertura hasta llegar a aéreas de no cobertura en una red Wifi. En todos los escenarios el aplicativo basado en plataforma JADE recupera automáticamente la conexión, continuando el proceso de video streaming, mejorando así la usabilidad del sistema.

Se ha encontrado limitaciones en JADE-ANDROID, ya que para que la arquitectura propuesta funcione al 100% se debería implementar tres agentes JADE, el primero para que se encargue de los mensajes RTSP y RTCP, ya que podrían compartir el mismo canal porque no importaría su envío inmediato; el segundo agente se necesitaría para que pueda asegurar el paso del audio, por un canal dedicado sólo a esta causa, para evitar retardos. En cambio el tercer agente se necesitaría para que se encargue totalmente del video, lo cual JADE-ANDROID no permite, ya que sólo limita a que se pueda usar un agente JADE por aplicación android, por lo que la transmisión se ve penalizada porque todos los flujos se los envía por un solo canal.

Otra limitación es el reproductor PvPlayer, ya que como se indicó en el capítulo anterior, cuando existe un retardo significativo, el PvPlayer desecha los paquetes y no los reproduce.

5.2 Recomendaciones

- Se recomienda el desarrollo de agentes similares a JADE para otro tipo de dispositivos móviles como Iphone, BlackBerry, etc. que sean dinámicamente adaptables a las

condiciones reales del canal inalámbrico y que permitan aplicar la técnica de video streaming en videos de alta calidad.

- Para que la emisión del video se presente con calidad, es muy importante configurar de manera correcta los parámetros de VLC.
- En el caso de la arquitectura propuesta, se recomienda adaptar un mecanismo que permita crear dinámicamente agentes proxy en el lado del servidor, ya que cada agente JADE trabaja punto a punto, lo que quiere decir que si otro cliente se conecta afectará terriblemente en el desempeño de la transmisión. Por lo tanto debería existir un agente proxy servidor por cada cliente, para que exista un servicio de video streaming totalmente de calidad.
- Otra recomendación que mejoraría notablemente el desempeño de la arquitectura, sería la de implantar directamente en el reproductor de video, al agente JADE, utilizando el framework JNI, que permite que un programa escrito en Java pueda interactuar con programas escritos en otros lenguajes como C y C++, en los cuales la mayoría de los reproductores de video son desarrollados, como es el caso de PvPlayer, que viene por defecto en Android.
- Con los resultados obtenidos se recomienda elaborar un algoritmo que maneje la variación del timestamp, durante una desconexión tanto en el audio como en el video.

ABREVIATURAS

AAC	Advanced Audio Coding.
ACL	Lista de control de acceso.
ADSL	Asymmetric Digital Subscriber Line.
AID	Agent Identifier
AMS	Agent managment system.
APS	Agente proxy servidor
APC	Agente proxy cliente.
AVI	Audio Video Interleave.
CDMA	Acceso múltiple por división de código.
CMD	Command Prompt.
Códec	Codificador/Decodificador.
DF	Directoty Facilitator
DHCP	Dynamic Host Configuration Protocol.
DNS	Domain Name System
FIPA	Foundation for intelligent Physical Agent.
FPS	Frame por segundo
GPL	Licencia Pública General.

GPS	Sistema de Posicionamiento Global.
HSDPA	High Speed Downlink Packet Access.
HTTP	Protocolo de Transferencia de Hipertexto.
IDE	Entorno de Desarrollo Integrado.
IGMP	Protocolo de administración de grupos de Internet.
IP	Protocolo de Internet
JADE	Java Agent Development Framework.
JAR	Java Archive
JDK	Java Development Kit o kit de desarrollo de Java.
Kb/s	Kilo bytes por segundo
LEAP	Leap Lightweight Extensible Agent Platform.
LGPL	Licencia Pública General Reducida
Mp3	Conocido como MPEG-1 Moving Picture Experts Group.
ms	Milisegundos
MTP	Media Transfer Protocol.
OSPF	Open Shortest Path First.
PDA	Asistente Digital Personal.

QoS	Calidad de servicio.
RIP	Protocolo de encaminamiento de información.
RTCP	Protocolo de control RTP.
RTP	Protocolo de transporte de tiempo real.
RTSP	Real time streaming protocol.
SDK	Software development kit o Kit de desarrollo de software.
SNMP	Simple Network Management Protocol.
SSRC	Synchronization Source
TCP	Protocolo de control de transmisión.
UDP	Protocolo de datagrama de usuario.
URL	Localizador uniforme de recursos.
VLC	Video Lan Client.
WIFI	Wireless Fidelity
3GPP	3rd Generation Partnership Project.

GLOSARIO

Agente Jade	Un agente JADE es una instancia de una clase de JAVA definida por el usuario que extiende la clase AGENTE básica.
Ancho de Banda	Es la cantidad de información o de datos que se puede enviar a través de una conexión de red en un período de tiempo dado.
Ant	Es una herramienta usada en programación para la realización de tareas mecánicas durante la fase de complicación y construcción.
Área de Cobertura	Es el espacio donde el teléfono móvil tiene una buena intensidad de señal para poder establecer comunicación con las aplicaciones.
Array	Es una zona de almacenamiento contiguo, que contiene una serie de elementos del mismo tipo
Backend	Se relaciona con el principio y el final de un proceso.
Bibliotecas de Software	Es un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos. Esto permite que el código y los datos se compartan y puedan modificarse
Buffer	Memoria de almacenamiento temporal de información que permite transferir los datos entre unidades funcionales con características de transferencia diferentes.

Calidad	Es un conjunto de propiedades inherentes a un objeto o servicio, que le confieren capacidad para satisfacer necesidades implícitas o explícitas.
Cliente	Es el que realiza las peticiones.
Contenedor	Es una instancia de ejecución.
Contenedor principal	Es donde se encuentran las herramientas que hacen funcional a JADE.
Contenedor de video	Es un tipo de formato de archivo que almacena información de vídeo, audio, subtítulos, capítulos, meta-datos e información de sincronización siguiendo un formato preestablecido en su especificación.
Conmutación de Paquetes	Es el envío de datos en una red de computadoras. Un paquete es un grupo de información que consta de dos partes: los datos propiamente dichos y la información de control, que especifica la ruta a seguir a lo largo de la red hasta el destino del paquete
Frame	Es un fotograma o cuadro, una imagen particular dentro de una sucesión de imágenes que componen una animación o video.
Framework	Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.
Frontend	Se relaciona con el principio y el final de un proceso. Se

encuentra en el dispositivo móvil.

Java	Es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++.
Máquina Virtual de Proceso	Se ejecuta como un proceso normal dentro de un sistema operativo y soporta un solo proceso. La máquina se inicia automáticamente cuando se lanza el proceso que se desea ejecutar y se detiene para cuando éste finaliza. Su objetivo es el de proporcionar un entorno de ejecución independiente de la plataforma de hardware y del sistema operativo, que oculte los detalles de la plataforma subyacente y permita que un programa se ejecute siempre de la misma forma sobre cualquier plataforma.
Metodología	Se encarga de elaborar estrategias de desarrollo de software que promuevan prácticas adaptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente
Multimedia	Se refiere a cualquier objeto o sistema que utiliza múltiples medios de expresión (físicos o digitales) para presentar o comunicar información.
Payload	Área de datos
Plugin	Es un programa que incrementa o aumenta las funcionalidades de un programa principal.

Protocolo HTTP	Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición.
Proxy	Programa o dispositivo que realiza una acción en representación de otro
Puerto	Interfaz para comunicar programa a través de una red.
Reproductor Multimedia	Es un programa informático o un dispositivo capaz de mostrar un abanico de contenidos audiovisuales.
Servidor	Es una computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes.
Sincronización	Es hacer concordar la imagen de video y el audio de una película
Sniffer	Aplicación de monitorización y de análisis para el tráfico de una red para detectar problemas.
Socket	Es la combinación de una dirección IP y un número de puerto.
Tasa de Muestreo	Es el número de muestras por unidad de tiempo que se toman de una señal continua para producir una señal discreta.
Timestamp	Marca de tiempo. Es una secuencia de caracteres, que denotan la hora y fecha en la cual ocurrió un determinado evento
Tecnología ADSL	Esta nueva tecnología digital ha permitido a las empresas que ofrecen servicios telefónicos competir en el mercado de

servicios de Internet de alta velocidad, utilizando la misma línea telefónica mediante la separación de las señales de voz y datos.

Televisión digital

Se refiere al conjunto de tecnologías de transmisión y recepción de imagen y sonido, a través de señales digitales.

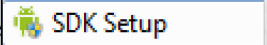
Variables de entorno

Son un conjunto de valores dinámicos que normalmente afectan el comportamiento de los procesos en una computadora.

ANEXOS

1 Anexo 1- Manual de Instalación

1.1 Instalación de Android SDK.

- 1.- Descargar el sdk de Android de la url <http://developer.android.com/sdk/index.html>, según el sistema operativo en este caso windows.
- 2.- Descomprimir el archivo descargado “android-sdk_r06-windows”, se recomienda hacerlo en una carpeta (ejemplo “android”) en la raíz de la unidad C.
- 3.- Ejecutamos c:\android\sdk setup.exe 
- 4.- Seleccionar en el menú lateral “Settings”.
- 5.- Activar la casilla “Force https://...”.
- 6.- Clic en el botón Save & Apply.

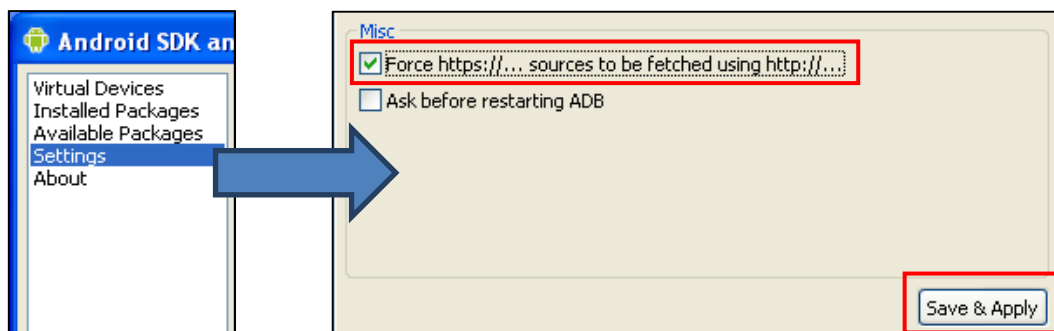


Ilustración 1.1 - Android SDK

- 7.- Seleccionamos Android SDK Tools (herramientas necesarias), SDK Plataforma Android 2.2 y USB Driver package (servirá para poder cargar las aplicaciones directamente al celular Android).

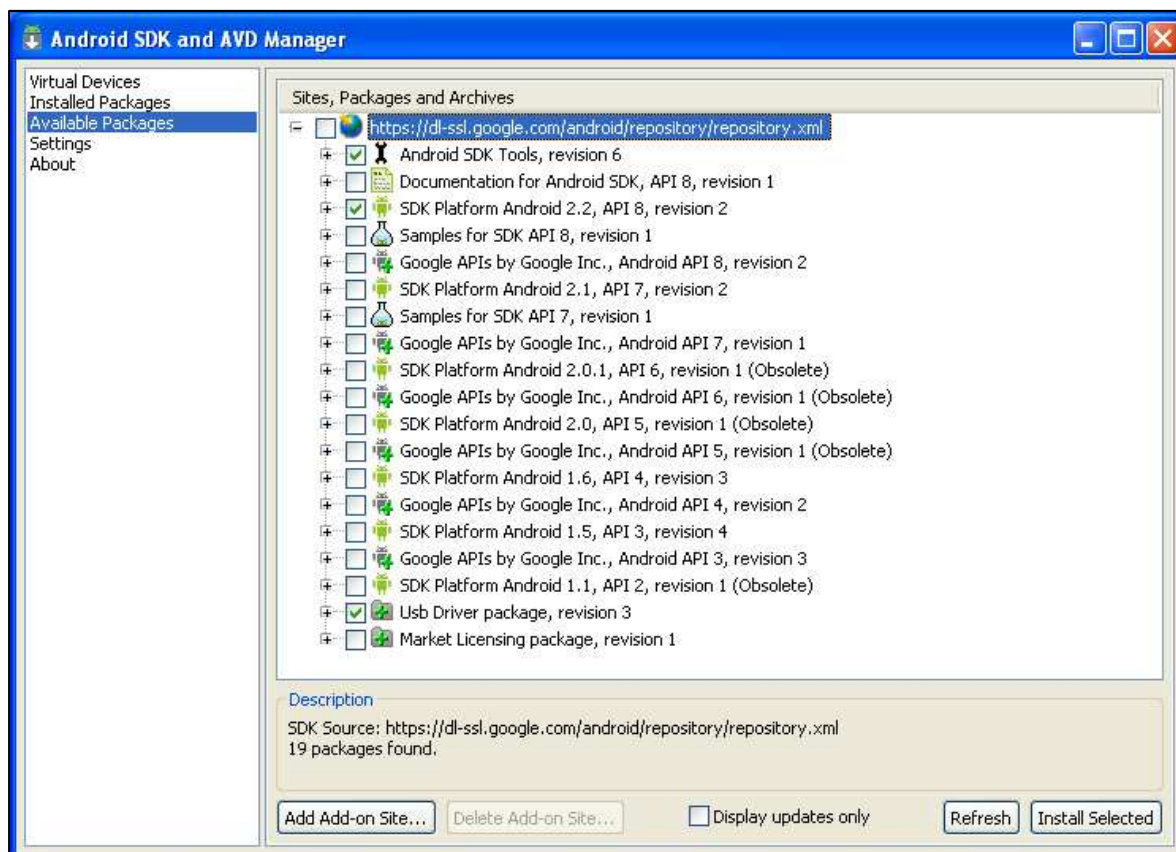


Ilustración 1.2 - Paquetes Disponibles

8.- Clic en el botón Install Selected

9.- Clic en Install

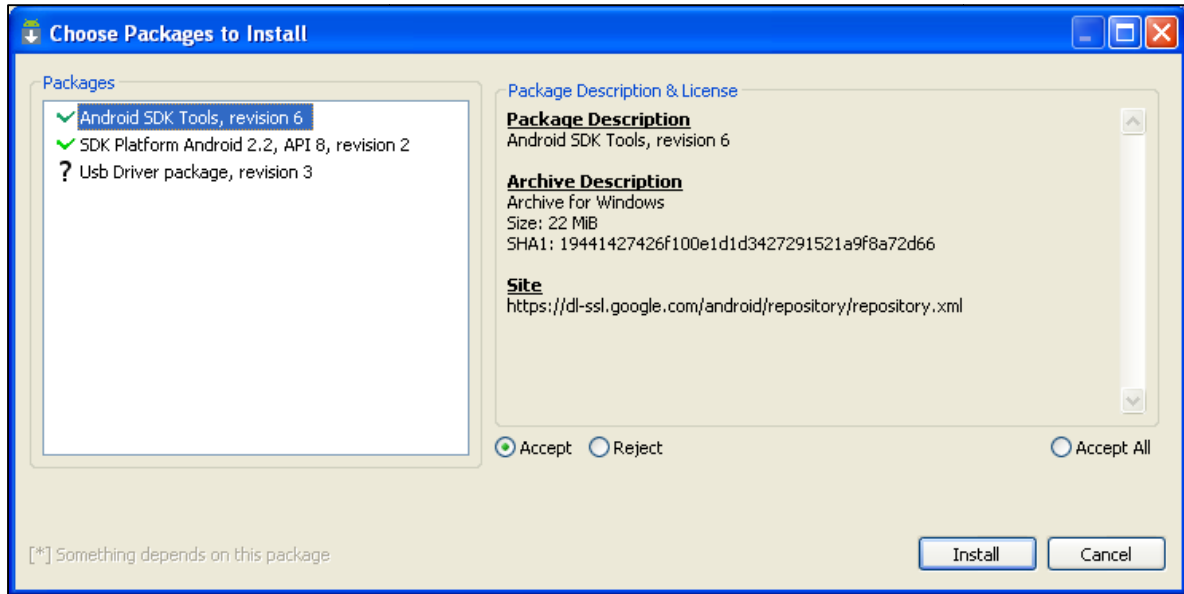


Ilustración 1.3 - Descarga SDK

1.2 Instalación de plugins y herramientas de programación.

1.2.1 Instalación de JAVA

1. Descargar Java Development Kit de la página oficial <http://www.oracle.com> según las características de la máquina en donde se desea instalar (32 o 64 bits) y sistema operativo, en este caso se usará windows.



Ilustración 1.4 - Descarga JAVA

2. Ejecutar el archivo descargado, para proceder con la instalación:
 - jdk-version-windows-i586.exe (32 bits).
 - jdk-version-windows-x64.exe (64 bits).
3. Señalar el directorio donde se instalará el jdk, por defecto es C:\Program Files\Java

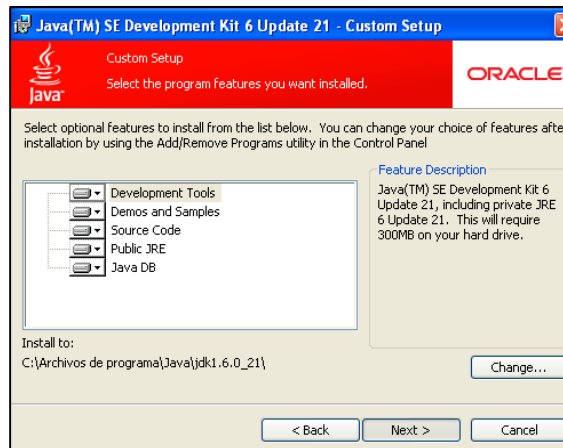


Ilustración 1.5 - Directorio JDK

4. Instalar el jdk.
5. Señalar el directorio donde se instalará el jre, por defecto es C:\Program Files\Java
6. Instalar el jre.



Ilustración 1.6 - Fin de instalación

1.2.1.1 Configuración de las Variables de Entorno

1. Para abrir las variables de entorno del sistema, click derecho en MiPc y seleccionar Propiedades, Opciones Avanzadas y click en el botón variables de entorno.

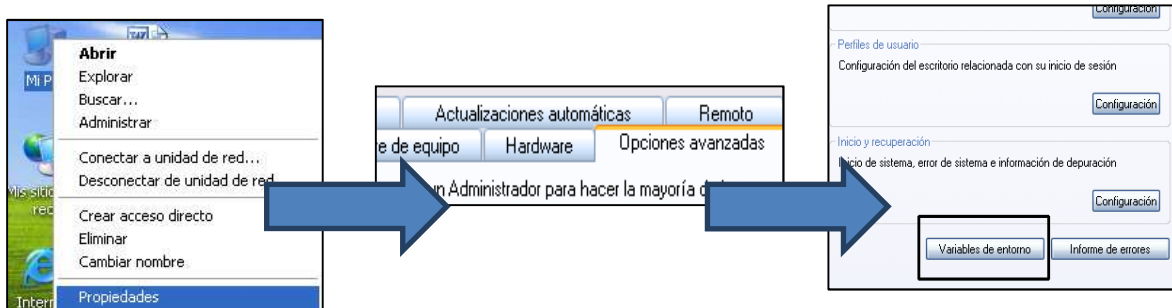


Ilustración 1.7 - Variables de Entorno

2. Modificar la variable Path, añadiendole al valor de la variable la dirección donde se encuentra instalado la carpeta bin del jdk, ante puesto el símbolo “;”.
En este caso debería ser “;C:\Program Files\Java\jdk1.6.0_18\bin”.



Ilustración 1.8 - Variable PATH

3. Aceptar
4. Añadir la variable CLASSPATH, con el valor de la variable, la dirección donde se encuentra src.zip del jdk. Lo cuál quedaría de la siguiente manera:
“C:\Program Files\Java\jdk1.6.0_18\src.zip”

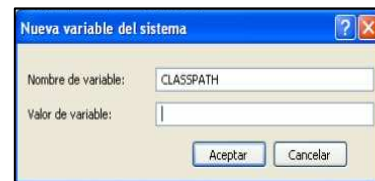


Ilustración 1.9 - Variable CLASSPATH

5. Aceptar

6. Añadir la variable JAVA_HOME cuyo valor será la dirección donde está instalado el jdk y el jre. “C:\Program Files\Java\jdk1.6.0_18”

1.2.2 Instalación de Eclipse

1.- Descargar Eclipse IDE for Java EE Developers de la siguiente url

<http://www.eclipse.org/downloads/>, según el tipo de arquitectura que posea el computador (32 o 64 bits).

2.- Descomprimir el archivo descargado.

3.- Copiar la carpeta eclipse en el directorio de preferencia.

4.- (No Obligatorio) Crear un acceso directo en el escritorio del archivo eclipse.exe.

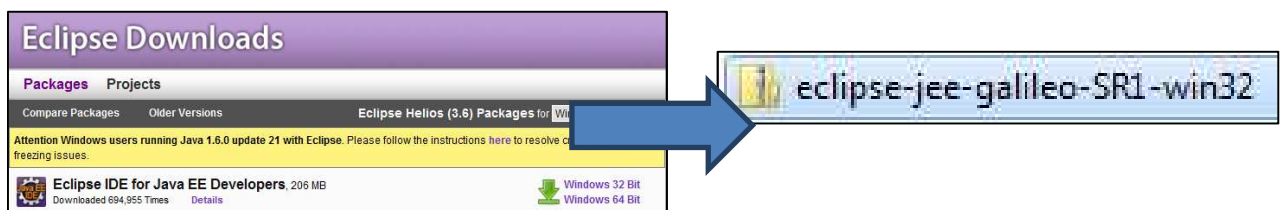


Ilustración 1.10 – Descarga Eclipse

1.2.3 Instalación Plugin de Android

1. Ejecutamos Eclipse.
2. En el menú principal escogemos la opción Help-> Install new software.

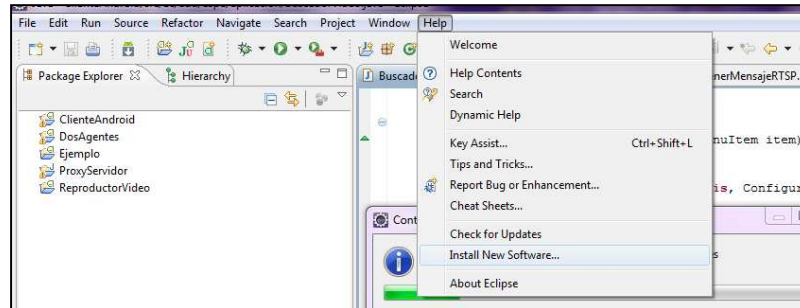


Ilustración 1.11 - Menú Eclipse

3. Escogemos la opción add y ponemos la siguiente URL: <https://dl-ssl.google.com/android/eclipse/>, y en el campo Name puede ir un nombre que identifique a este plugin, en este caso se llamará Android. Seleccionamos la opción OK.

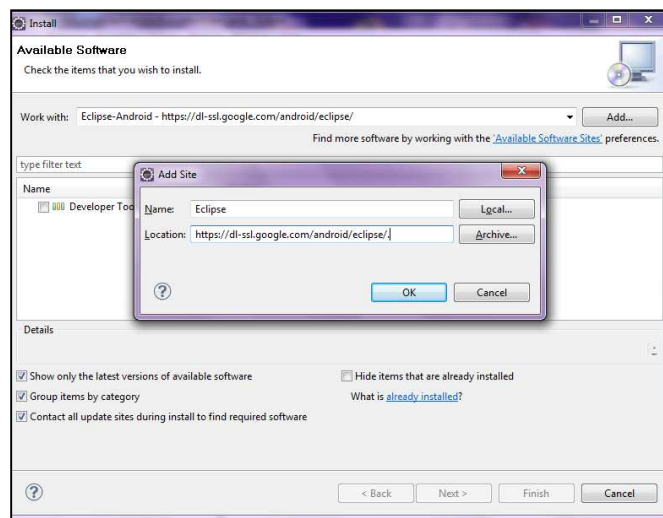


Ilustración 1.12 - URL

4. Seleccionamos en el campo Work with la dirección ingresada en el paso anterior, automáticamente cargará los paquetes disponibles a instalar, los seleccionamos a todos dando clic en la opción Developer Tools. Por último damos clic en Next.

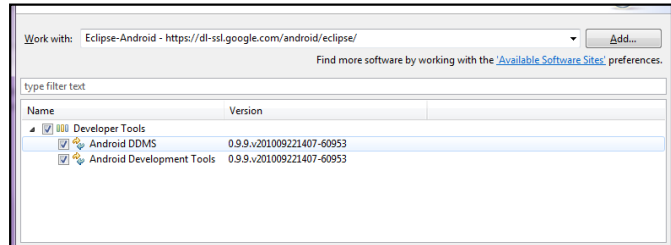


Ilustración 1.13 - Android Tools

5. Damos clic en finish para proceder a instalar, se espera a que se instale, operación que demorará según la velocidad de internet.

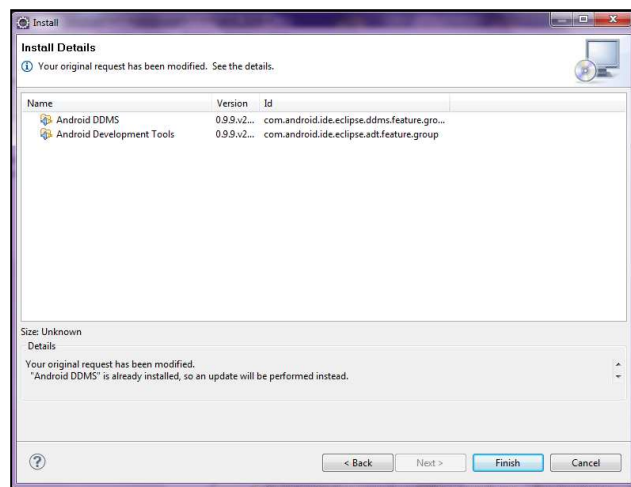


Ilustración 1.14 - Instalación

6. Luego de acabar con el proceso de instalación se reiniciará Eclipse y se configurará los parámetros necesarios para el funcionamiento de la plataforma Android. En el menú principal de Eclipse escogemos la opción Window -> Preferences.

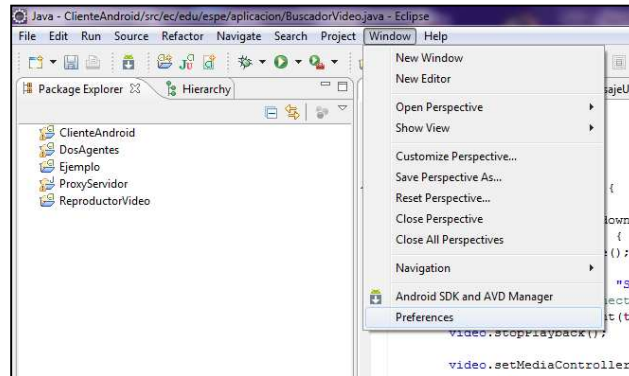


Ilustración 1.15 - Configuración Android SDK

7. En la opción Android señalamos donde se encuentra localizado el SDK. En este caso está ubicado en la siguiente dirección: c:\Android.

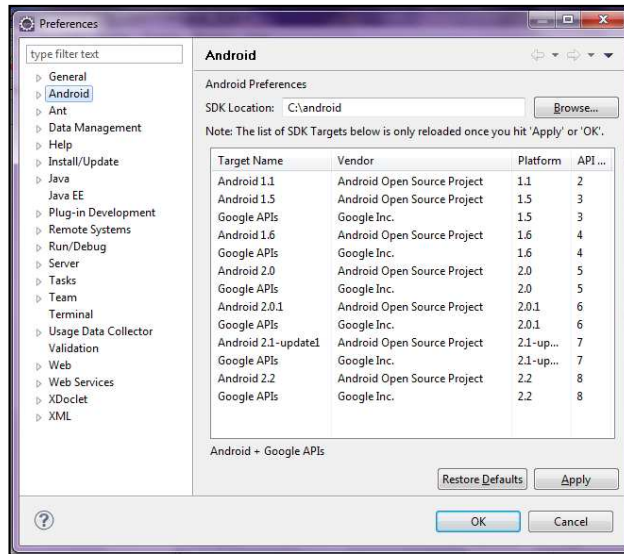



Ilustración 1.16 - Localización SDK

1.2.4 Configuración del Emulador de Android

1. Dentro de Eclipse en la barra de herramientas, encontramos una opción llamada Android SDK and AVD Manager, representada por el siguiente icono:  Esta opción ayuda a configurar un dispositivo en el emulador.
2. Nos ubicamos en la barra izquierda y escogemos Virtual Devices.
3. Ahora en la parte derecha de la ventana, presionamos el botón New.

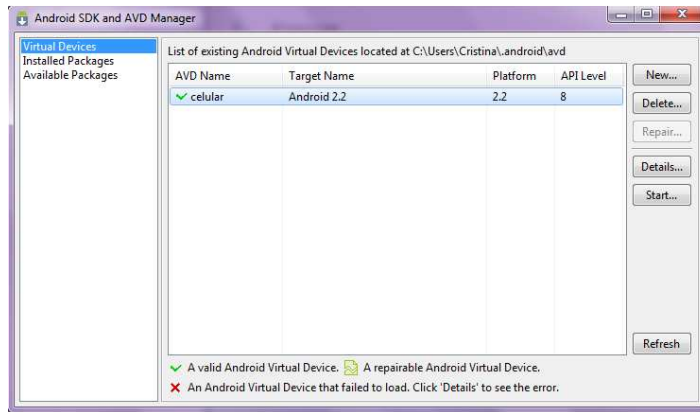


Ilustración 1.17 - Paso 3 y 4

- Configuramos el emulador, en el campo Name ponemos algo que lo identifique, en target escogemos la versión de SDK a utilizar en este caso Android 2.2, por último configuramos el tamaño de la memoria del emulador, recomendamos 300. Y damos clic en Create AVD.

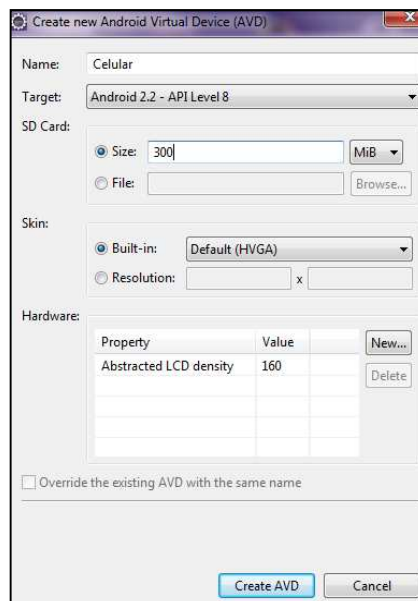


Ilustración 1.18 - Configuración emulador

1.2.5 Configuración de JADE

1.2.5.1 Herramienta Apache Ant

1. Para compilar y construir la plataforma Jade vamos a utilizar la herramienta ant, la misma que debe ser descargada de la siguiente dirección: <http://ant.apache.org>.
2. Teniendo la herramienta ant lo descomprimos y lo ubicamos en la unidad c. Se recomienda la siguiente dirección: c:\ant.
3. El siguiente paso es añadir las variables de entorno necesarias para la configuración:
 - ANT_HOME: debe apuntar donde se encuentra el directorio ant (c:\ant).
 - PATH: se le debe añadir la dirección donde se encuentra el directorio bin de ant (c:\ant\bin).
 - Para probar la correcta configuración abrimos el símbolo del sistema, y escribimos el comando ant, y debemos obtener el siguiente mensaje:

```
C:\>ant
Buildfile: build.xml does not exist!
Build failed
C:\>_
```

Ilustración 1.19 - CMD Ant

1.2.5.2 Jade

1. Descargamos la plataforma JADE de la siguiente página web: <http://jade.tilab.com/>, para esto debemos estar registrarnos.

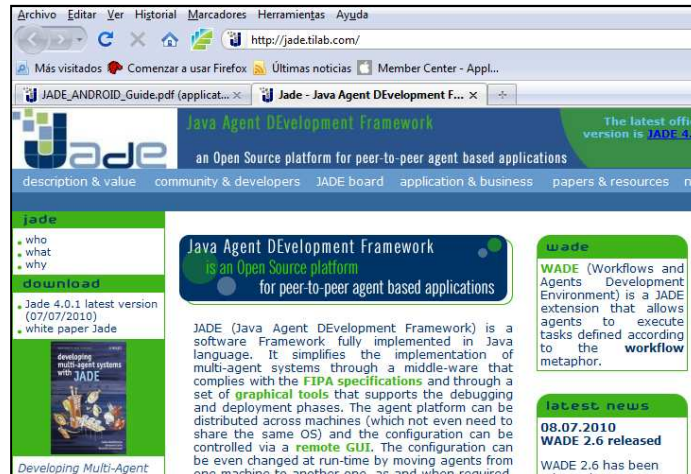


Ilustración 1.20 - Página Oficial JADE

2. Procedemos a descargar el código fuente de la versión de Jade 4.0.1, que en este momento es la más actual a demás la configuración que va a ser explicada a continuación es realizada exclusivamente para esta versión.
3. Luego de habernos descargado jade, descomprimos el archivo y copiamos la carpeta JADE en la unidad C. Se recomienda la siguiente dirección: c:\jade.
4. Para compilar y construir la plataforma ejecutamos los siguientes comandos en el símbolo del sistema y dentro del directorio jade:
 - ant lib.- teniendo como resultado exitoso BUILD SUCCESSFUL.
 - ant jade.- teniendo como resultado exitoso BUILD SUCCESSFUL.

```

makeExtendedSLParser:
parser:
check_execIdlj:
[echo] Default behaviour: IDLJ task is being skipped.
idlj:
check_execSubWCRev:
[echo] WARNING: SubWCRev is being skipped. The copyright notice of JADE Run
time will not be updated
SubWCRev:
doTag:
jade:
[java] C:\jade\build.xml:143: warning: 'includeantruntime' was not set, def
aulting to build.sysclasspath=last; set to false for repeatable builds
BUILD SUCCESSFUL
Total time: 3 seconds
C:\jade>

```

Ilustración 1.21 - Compilación JADE

1.2.5.3 Complemento Leap

1. Descargamos el complemento de la siguiente página web: <http://jade.tilab.com/>, dentro de la sección “community & developers” -> “[add-ons & 3rd party sw](#)”.
2. Y descargamos el complemento [Leap](#).
3. Descomprimos el archivo y la carpeta leap la copiamos en la ruta donde se encuentre localizado jade. (c:\jade).

1.2.5.4 Complemento JadeLeap-Android

1. Descargamos el complemento de la siguiente página web: <http://jade.tilab.com/>, dentro de la sección “community & developers” -> “[add-ons & 3rd party sw](#)”.
2. Y descargamos el complemento [JADE-Android](#).
3. Descomprimos el archivo y la carpeta add-ons la copiamos en la ruta donde se encuentre localizado jade. (c:\jade).

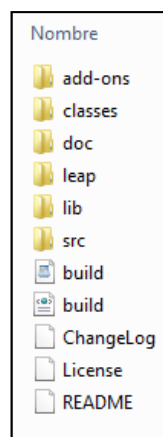


Ilustración 1.22 - add-ons

4. Nos ubicamos en la ruta C:\jade\add-ons\jade4android, y procedemos a modificar el archivo build.properties:
 - **android-sdk-folder**: propiedad que apunta al directorio donde se encuentra instalado el android SDK. (c:/android)

- **android-target:** propiedad que especifica que versión de Android SDK se va a utilizar: android-3 para Plataforma Android 1.5, android-4 para Plataforma Android 1.6, android-7 para Plataforma Android 2.1 y android-8 para Plataforma Android 2.2
- **jade-home-dir:** propiedad que apunta al directorio donde JADE está instalado. (c:/jade).

Nota: las direcciones que se deben poner en las propiedades deben estar separadas por slash (/) en vez de backslash(\).

5. En el símbolo del sistema, nos ubicamos en la ruta C:\jade\add-ons\jade4android y ejecutamos el siguiente comando:

- ant jar: teniendo como resultado exitoso BUILD SUCCESSFUL, y la configuración completa para JADE.

2 Anexo 2.-Manual del Usuario

2.1.1 Requerimientos Hardware

Para que el software Streaming-Jade V1.1, funcione de manera correcta se necesita de un teléfono móvil, Android 2.2 Nexus One, con las siguientes características:

- Pantalla táctil de 3.7 pulgadas.
- Procesador Qualcomm QSD 8250 1 GHz.
- Android 2.2.
- 512MB Flash
- 512MB RAM
- Tarjeta microSD.
- Wi-Fi (802.11b/g/n)

Más información de las Características del Nexus One:

<http://www.celularis.com/htc/caracteristicas-nexus-one.php#ixzz1328zRHtl>.

2.1.2 Instalación del Software

1. En el Nexus One, vamos al menú principal y abrimos la aplicación Streaming –Jade, identificada por el siguiente icono:



Ilustración 2.1 - Ícono

2.1.3 Funcionamiento del Software

2. En el Nexus One, vamos al menú principal y abrimos la aplicación Streaming –Jade, identificada por el siguiente icono:
3. La aplicación se inicia de la siguiente manera:

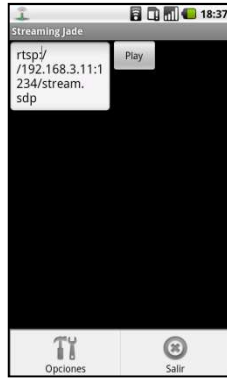


Ilustración 2.2 - interfaz de la aplicación

4. Pulsamos el botón de menú -> opciones, y configuramos lo siguiente:

Tabla 2.1- Opciones de Configuración

Opción	Descripción
Agente Proxy	
Proxy	Activar o Desactivar, si se desactiva ya no se configura las demás opciones.
Puerto RTSP	Puerto local que se va a utilizar para la negociación RTSP. Por defecto 1234.
Plataforma Jade	
Dirección IP	Contenedor principal JADE.
Puerto	Puerto que escucha el contenedor principal. Por defecto 1099.

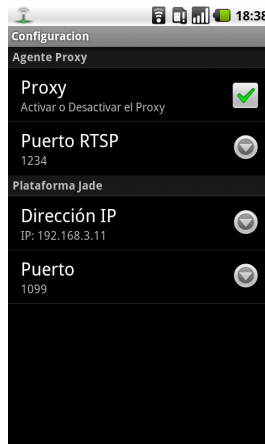


Ilustración 2.3 - Menú

5. Pulsamos el botón regresar, e ingresamos en la caja de texto la dirección IP en donde se encuentra el recurso de video RTSP (VLC),
6. Por último pulsamos en Play, y el video se reproduce.

3 Anexo 3.-Código de la Aplicación

3.1.1 Cliente Android - APC

3.1.1.1 Interfaz de la Aplicación

3.1.1.1.1 Buscador de Video

BuscadorVideo.java

```

package ec.edu.espe.aplicacion;
import android.app.Activity;
/**
 *
 * @author ESPE
 * Actividad en la cuál se ingresará las direcciones RTSP a reproducir
 *
 */
public class BuscadorVideo extends Activity {
private EditText txtDireccion;
private Button btnStart;
private SharedPreferences sharedPreferences; //Manejador de configuraciones
almacenada
/*
 * (non-Javadoc)
 * @see
 android.app.Activity#onConfigurationChanged(android.content.res.Configurati
 on)
 * Evento que evita que la actividad se reinicie cada que el dispositivo
 móvil
 * cambia de lado
 */
@Override
public void onConfigurationChanged(Configuration newConfig) {
super.onConfigurationChanged(newConfig);
}
/*
 * (non-Javadoc)
 * @see android.app.Activity#create(android.os.Bundle)
 * Evento que actúa cuando la actividad inicia, en donde se carga los
 * componentes gráficos como las cajas de texto y los botones
 */
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.buscar);
sharedPreferences =
PreferenceManager.getDefaultSharedPreferences(this); //instancia el manejador
que almacena
los datos
btnStart = (Button) this.findViewById(R.id.Button01);
txtDireccion = (EditText) findViewById(R.id.EditText01);
txtDireccion.setText("rtsp://192.168.3.11:1234/stream.sdp");
/*
 * Evento que se ejecuta cuando el botón es presionado
 */
btnStart.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
String path1 = txtDireccion.getText().toString();
//obtención y creación de las dos direcciones
//url (real y local)

```

```

UrlRTSP direccion=new UrlRTSP(path1);
UrlRTSP direccionLocal=new UrlRTSP(direccion.getURL());
direccionLocal.setHost("localhost");
direccionLocal.setPuerto(Integer.parseInt(sharedPreferences.getString("pu
erto_rtsp_configuracion", "1234")));
//crea la actividad que visualiza el video
iniciarVideoPlayer(direccion.getURL(),direccionLocal.getURL());
}

});
}
/**
 * función encargada de iniciar la actividad que visualiza el video
streaming
 * @param urlReal.- dirección que el usuario desea observar
 * @param urlLocal.- modificación de la dirección real para que apunte a
localhost
 */
public void iniciarVideoPlayer(String urlReal,String urlLocal){
Intent videoPlayer=new Intent(this, VideoPlayer.class);
videoPlayer.putExtra(VideoPlayer.DIRECCION, urlReal);//envío de parámetros
urlReal
videoPlayer.putExtra(VideoPlayer.DIRECCION_LOCAL, urlLocal);//envío de
parámetro
urlLocal
startActivity(videoPlayer);//inicio de la actividad
}
/*
 * (non-Javadoc)
 * @see android.app.Activity#onCreateOptionsMenu(android.view.Menu)
 * Llama al menú de los recursos de la aplicación y la instancia
 */
@Override
public boolean onCreateOptionsMenu(Menu menu) {
MenuInflater inflater = getMenuInflater();
inflater.inflate(R.menu.menu, menu);
return true;
}
/*
 * (non-Javadoc)
 * @see android.app.Activity#onOptionsItemSelected(android.view.MenuItem)
 * Se define las acciones a realizar por cada elemento en el menú
 */
@Override
public boolean onOptionsItemSelected(MenuItem item) {
switch (item.getItemId()) {
case R.id.opcion_menu:
startActivity(new Intent(this, Configuracion.class));//inicia a la
actividad de la Configuración
return(true);
case R.id.salir_menu:
finish();//termina la aplicacion

```



```

return(true);
}
return super.onOptionsItemSelected(item);
}
@Override
protected void onStop() {
super.onStop();
}
@Override
protected void onPause() {
super.onPause();
}
@Override
protected void onDestroy() {
super.onDestroy();
}
}

```

3.1.1.1.2 JadeAplicación

```

package ec.edu.espe.aplicacion;

```

```

import jade.android.ConnectionListener;
import jade.android.JadeGateway;
import jade.core.Profile;
import jade.imtp.leap.JICP.JICPProtocol;
import jade.util.leap.Properties;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.preference.PreferenceManager;
import android.telephony.TelephonyManager;
import android.util.Log;
import android.widget.Toast;
import ec.edu.espe.proxy.AgenteProxy;
import ec.edu.espe.util.UrlRTPSP;

/**
 *
 * @author ESPE Esta clase está encargada de iniciar el servicio
MicroRuntime de
 * JADE y activar al agente proxy. Además gestiona la conexión y
 * desconexión del servicio
 */
public class JadeAplicacion implements ConnectionListener {

    private Context contexto; // contexto donde se ejecutará (Actividad
// VideoPlayer)
    private JadeGateway jadeGateway; // Puerta de enlace con el agente
proxy

```

```

    private Object[] utilitarios; // arreglo donde se almacena las urls y
    se los
                                                // pasará

    // como parámetros al agente proxy

    public JadeAplicacion(Context contexto, String direccionStr,
        String direccionLocalStr) {
        super();

        this.contexto = contexto;
        UrlRTSP direccion = new UrlRTSP(direccionStr);
        UrlRTSP direccionLocal = new UrlRTSP(direccionLocalStr);
        utilitarios = new Object[2];
        utilitarios[0] = direccion;
        utilitarios[1] = direccionLocal;
    }

    // Activa el servicio MicroRuntime de JADE para android
    public void conectar() {
        try {
            // connect(Agente Jade a ejecutar, Propiedades de Inicio,
Contexto,
            // ConnectionListener)
            JadeGateway.connect(AgenteProxy.class.getName(),
                getPropiedadesJade(), contexto, this);
        } catch (Exception e) {
        }
    }

    // Termina al agente y desvincula la actividad del servicio
    public void desconectar() {
        Log.e("onDestroy", "ShutdownJade");
        try {
            if (jadeGateway != null) {
                jadeGateway.shutdownJADE();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        if (jadeGateway != null) {
            jadeGateway.disconnect(contexto);
        }
        Log.e("Desconectado", "ShutdownJade");
        // Termina el servicio MicroRuntime
        contexto.stopService(new Intent(contexto,
            jade.android.MicroRuntimeService.class));
    }
}

```

```

        Toast.makeText(contexto, "Agente Proxy JADE Finalizado",
5000).show();
    }

    /*
     * (non-Javadoc)
     *
     * @see
     *
jade.android.ConnectionListener#onConnected(jade.android.JadeGateway)
     * Evento que se activa cuando el servicio MicroRuntime ha  sido
 activado y
     *  vinculado al contexto
     */
    @Override
    public void onConnected(JadeGateway jadeGateway) {
        this.jadeGateway = jadeGateway;
        try {
            //  activación del agente proxy y envío de parámetro
            (UrlReal y
                // urlLocal)
            this.jadeGateway.execute(utilitarios);
            Log.e("JADEGateway - OnConnected", "Agente Proxy
Iniciado");
            Toast.makeText(contexto, "Agente Proxy JADE Iniciado",
5000).show();
        } catch (Exception e) {
            Log.e("JADEGateway - OnConnected", e.toString());
            Toast.makeText(contexto, "Agente Proxy JADE - Error",
5000).show();
        }
    }

    /*
     * (non-Javadoc)
     *
     * @see jade.android.ConnectionListener#onDisconnected() Evento que
se
     *  ejecuta cuando el servicio fue desvinculado de la actividad por
 algún
     *  inconveniente
     */
    @Override
    public void onDisconnected() {
        Log.e("Desconectado", "Jade se desconecto");
        Toast.makeText(contexto, "JADE desconectado", 5000).show();
    }
}

```

```

    /*
    * Obtiene las propiedades necesarias para iniciar el agente y el
servicio
    * JADE
    */
    public Properties getPropiedadesJade() {
        SharedPreferences sharedPreferences = PreferenceManager
            .getDefaultSharedPreferences(contexto);
        TelephonyManager adminTelefono = (TelephonyManager) contexto
            .getSystemService(Context.TELEPHONY_SERVICE); //
IMEI del

                // celular
        Properties propiedad = new Properties();
        propiedad.setProperty(Profile.MAIN_HOST,
sharedPreferences.getString(
            "ip_servidor_configuracion", "192.168.3.15"));
        propiedad.setProperty(Profile.MAIN_PORT,
sharedPreferences.getString(
            "puerto_servidor_configuracion", "1099"));
        propiedad.setProperty(JICPPProtocol.MSISDN_KEY, adminTelefono
            .getDeviceId());
        return propiedad;
    }
}

```

3.1.1.1.3 *VideoPlayer*

```

package ec.edu.espe.aplicacion;

import android.app.Activity;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.view.Window;
import android.view.WindowManager;
import android.widget.MediaController;
import android.widget.VideoView;
/**
 *
 * @author ESPE
 * VideoPlayer es la actividad que se encarga de reproducir el video
 * así como activar a JADE por medio del objeto jadeAplicacion
 */
public class VideoPlayer extends Activity {

```

```

    public final static String DIRECCION_LOCAL="direccionLocal"; //nombre
del parámetro direccionLocal
    public final static String DIRECCION="direccion"; //nombre del
parámetro direccionLocal
    private VideoView video; //herramienta que permite visualizar el
video
    private MediaController controlVideo; //barra de estado del video
    private Uri uri;
    private JadeAplicacion jadeAplicacion;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        configuracionPantalla();//permite configurar y visualizar el
pantalla completa el video
        setContentView(R.layout.main);

        controlVideo=new MediaController(this);//inicializa al
mediacontroller
        if(estaActivadoProxyJade()){ //verifica si el proxy jade está
activado en la configuración
            jadeAplicacion=new JadeAplicacion(this,
getIntent().getStringExtra(VideoPlayer.DIRECCION),
getIntent().getStringExtra(VideoPlayer.DIRECCION_LOCAL));
            jadeAplicacion.conectar(); //activa el proxy
            uri =
Uri.parse(getIntent().getStringExtra(VideoPlayer.DIRECCION_LOCAL));
        }else{
            //reproduce al video de manera normal sin proxy
            jadeAplicacion=null;
            uri =
Uri.parse(getIntent().getStringExtra(VideoPlayer.DIRECCION));
        }
        video = (VideoView) findViewById(R.id.VideoView01);
        video.setKeepScreenOn(true);
        controlVideo.setMediaPlayer(video); //vincula la barra de
estado al video
        video.setMediaController(controlVideo);
        video.requestFocus();
        video.setVideoURI(uri);
        video.start();//inicia la visualización del video
    }

    /**
     * Verifica si en la configuración está activado el proxy jade
     * @return
     */

```

```

    public boolean estaActivadoProxyJade(){
        SharedPreferences sharedPreferences =
PreferenceManager.getDefaultSharedPreferences(this);
        return sharedPreferences.getBoolean("chk_activar_agente",
true);
    }

/**
 * configura la pantalla para la visualización
 */
    public void configuracionPantalla(){
        //quita la barra de título de la aplicación
requestWindowFeature(Window.FEATURE_NO_TITLE);
        //hace que la aplicación ocupe toda la pantalla

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        //evita que el teléfono se ponga en modo de standby

getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
    }
    @Override
    protected void onPause() {
        super.onPause();
        finish();
    }

    @Override
    protected void onStop() {
        super.onStop();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        video.stopPlayback(); //detiene la visualización del video
        video.setMediaController(null);
        if(jadeAplicacion!=null){
            jadeAplicacion.desconectar(); //desconecta JADE
        }
    }
}

```

3.1.1.2 Configuración

3.1.1.2.1.1 Configuración

```
package ec.edu.espe.configuracion;

import android.os.Bundle;
import android.preference.PreferenceActivity;
import ec.edu.espe.aplicacion.R;

/**
 *
 * @author ESPE Configuración es una actividad donde se puede cambiar los
 * parámetros que servirán para el desempeño de la visualización
como la
 * activación y desactivación del proxy
 */
public class Configuracion extends PreferenceActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.configuracion); // carga el
recurso
    }

    @Override
    protected void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
    }

}
```

3.1.1.2.1.2 IPJade

```
package ec.edu.espe.configuracion;

import java.net.InetAddress;

import android.content.Context;
import android.preference.EditTextPreference;
import android.util.AttributeSet;

/**
 *
 * @author ESPE Esta clase extiende de la herramienta existente
```

```

*      EditTextPreference con el fin de validar algunos datos y añadir
más
*      funcionalidades
*
*/
public class IPJade extends EditTextPreference {

    public IPJade(Context context) {
        super(context);
    }

    public IPJade(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    /*
    * (non-Javadoc)
    * @see
    android.preference.EditTextPreference#onSetInitialValue(boolean,
    * java.lang.Object) Inicializa los valores por defecto en caso no
existan
    * valores almacenados
    */
    @Override
    protected void onSetInitialValue(boolean restoreValue, Object
    defaultValue) {
        setSummary("IP: " + getPersistedString((String) defaultValue));
        super.onSetInitialValue(restoreValue, defaultValue);
    }

    // Actualiza el resumen de la herramienta cuando la alerta es cerrada
    @Override
    protected void onDialogClosed(boolean positiveResult) {
        if (positiveResult) {
            try {
                InetAddress direccion =
                InetAddress.getByName(getEditText()
                .getText().toString());
                setSummary("IP: " +
                getEditText().getText().toString());
                // getSharedPreferences().edit().commit();
            } catch (Exception e) {
                setSummary("No establecido");
            }
        }
        super.onDialogClosed(positiveResult);
    }
}

```



```
    }  
}
```

3.1.1.2.1.3 Puerto

```
package ec.edu.espe.configuracion;  
  
import android.content.Context;  
import android.preference.EditTextPreference;  
import android.util.AttributeSet;  
  
/**  
 *  
 * @author ESPE Esta clase extiende de la herramienta existente  
 * EditTextPreference con el fin de validar algunos datos y añadir  
más  
 * funcionalidades  
 */  
public class Puerto extends EditTextPreference {  
  
    public Puerto(Context context, AttributeSet attrs, int defStyle) {  
        super(context, attrs, defStyle);  
    }  
  
    public Puerto(Context context) {  
        super(context);  
    }  
  
    public Puerto(Context context, AttributeSet attrs) {  
        super(context, attrs);  
    }  
  
    // Verifica que los datos ingresados sean puertos válidos  
    @Override  
    protected void onDialogClosed(boolean positiveResult) {  
        if (positiveResult) {  
            try {  
                int puerto =  
Integer.parseInt(getEditText().getText()  
                    .toString());  
                if (puerto > 0  
                    &&  
getEditText().getText().toString().length() < 5) {  
  
                    setSummary(getEditText().getText().toString());  
                }  
            }  
        }  
    }  
}
```

```

        } else {
            setSummary("No establecido");
        }
    } catch (Exception e) {
        setSummary("No establecido");
    }
}
super.onDialogClosed(positiveResult);
}

/*
 * (non-Javadoc)
 *
 * @see
android.preference.EditTextPreference#onSetInitialValue(boolean,
 * java.lang.Object) Inicia los valores por defecto si no existen
valores
 * almacenados
 */
@Override
protected void onSetInitialValue(boolean restoreValue, Object
defaultValue) {
    if (restoreValue) {
        setSummary(getPersistedString((String) defaultValue));
    }
    super.onSetInitialValue(restoreValue, defaultValue);
}
}
}

```

3.1.1.3 Agente Proxy

```

package ec.edu.espe.proxy;

import jade.core.AID;
import jade.core.behaviours.Behaviour;
import jade.core.behaviours.ThreadedBehaviourFactory;
import jade.domain.DFService;
import jade.domain.FIPAEException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.SearchConstraints;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.lang.acl.ACLMessage;
import jade.wrapper.gateway.GatewayAgent;

import java.net.DatagramSocket;

```

```

import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.SocketException;

import android.util.Log;
import ec.edu.espe.proxy.comportamiento.ListenerPuertoRTSP;
import ec.edu.espe.proxy.comportamiento.ListenerPuertoUDP;
import ec.edu.espe.util.UrlRTSP;

/**
 *
 * @author ESPE La clase AgenteProxy es la puerta de enlace entre el
 *          VideoReproductor y Servidor. Sus comportamientos serán los
encargados
 *          de asegurar que los flujos de información lleguen a su
destinatario.
 *
 */
public class AgenteProxy extends GatewayAgent {

    private static final long serialVersionUID = 1L;

    private Socket socketRTSP;

    /**
     * *****Variables para Track 0*****
     */
    // -----RTP-----
    private DatagramSocket socketRTPTrack0;
    private InetSocketAddress socketClienteRTPTrack0;
    // -----RTCP-----
    private DatagramSocket socketRTCPTrack0;
    private InetSocketAddress socketClienteRTCPTrack0;

    /**
     * *****Variables para Track 1*****
     */
    // -----RTP-----
    private DatagramSocket socketRTPTrack1;
    private InetSocketAddress socketClienteRTPTrack1;
    // -----RTCP-----
    private DatagramSocket socketRTCPTrack1;
    private InetSocketAddress socketClienteRTCPTrack1;

    private UrlRTSP urlDestino = null; // rtsp://host:puerto/recurso
    private UrlRTSP urlLocal; // rtsp://localhost:puerto/recurso
    (urlDestino
                                     // modificado)

```

```

private AID agenteDestino;

private ThreadedBehaviourFactory tbf; // convierte en hilos a los
//
comportamientos jade

/*
 * Comportamientos que escucharán los flujos provenientes del
 * videoreproductor
 */
private Behaviour listenerPuertoRTSP;

private Behaviour listenerPuertoRTPTrack0;
private Behaviour listenerPuertoRTPTrack1;

private Behaviour listenerPuertoRTCPTrack0;
private Behaviour listenerPuertoRTCPTrack1;

/*
 * (non-Javadoc)
 *
 * @see
jade.wrapper.gateway.GatewayAgent#processCommand(java.lang.Object)
 * Recibe los parámetros: urlDestino y urlLocal
 */
@Override
protected void processCommand(final Object command) {
    if (command instanceof Object[]) {
        Log.i("AgenteProxy - Command", "Instancia Object");
        Object[] aux = (Object[]) command;
        urlDestino = (UrlRTSP) aux[0];
        Log.i("Comando - urlDestino", urlDestino.getURL());
        urlLocal = (UrlRTSP) aux[1];
        Log.i("Comando - urlLocal", urlLocal.getURL());
    }
    tbf = new ThreadedBehaviourFactory();
    // Adición de comportamientos al agente
    listenerPuertoRTSP = new ListenerPuertoRTSP(this, urlDestino,
urlLocal);
    addBehaviour(tbf.wrap(listenerPuertoRTSP));
    listenerPuertoRTPTrack0 = new ListenerPuertoUDP(this,
socketRTPTrack0,
        ACLMessage.QUERY_REF, getAgenteDestino());
    addBehaviour(tbf.wrap(listenerPuertoRTPTrack0));
    listenerPuertoRTCPTrack0 = new ListenerPuertoUDP(this,
socketRTCPTrack0, ACLMessage.INFORM_REF,
getAgenteDestino());

```

```

        addBehaviour(tbf.wrap(listenerPuertoRTCPTrack0));
        listenerPuertoRTPTrack1 = new ListenerPuertoUDP(this,
socketRTPTrack1,
                ACLMessage.QUERY_IF, getAgenteDestino());
        addBehaviour(tbf.wrap(listenerPuertoRTPTrack1));
        listenerPuertoRTCPTrack1 = new ListenerPuertoUDP(this,
                socketRTCPTrack1, ACLMessage.INFORM_IF,
getAgenteDestino());
        addBehaviour(tbf.wrap(listenerPuertoRTCPTrack1));
        releaseCommand(command); // retorno y confirmación, que el
agente
// recibió los
// parámetros correctamente
    }

    @Override
    protected void setup() {
        super.setup();
        Log.e("Setup", "AgenteProxy");
        try {
            /*
            * Inicialización de los sockets para escuchar los flujos
del
            * videoreproductor
            */
            socketRTPTrack0 = new DatagramSocket();
            Log.i("RTP track0", socketRTPTrack0.getLocalPort() + "");
            socketRTCPTrack0 = new DatagramSocket();
            Log.i("RTCP track0", socketRTCPTrack0.getLocalPort() +
"");
            socketRTPTrack1 = new DatagramSocket();
            Log.i("RTP track1", socketRTPTrack1.getLocalPort() + "");
            socketRTCPTrack1 = new DatagramSocket();
            Log.i("RTCP track1", socketRTCPTrack1.getLocalPort() +
"");
        } catch (SocketException e) {
            e.printStackTrace();
        }
        // Busca un agente disponible en el servidor para la
comunicación
        agenteDestino = buscarAgenteDestino();
        Log.e("Servidor", agenteDestino.getName());
        Log.e("Servidor", agenteDestino.getLocalName());
    }

    public InetSocketAddress getSocketClienteRTPTrack0() {
        return socketClienteRTPTrack0;
    }

```

```

}

public void setSocketClienteRTPTrack0(
    InetAddress socketClienteRTPTrack0) {
    this.socketClienteRTPTrack0 = socketClienteRTPTrack0;
}

public InetAddress getSocketClienteRTPTrack1() {
    return socketClienteRTPTrack1;
}

public void setSocketClienteRTPTrack1(
    InetAddress socketClienteRTPTrack1) {
    this.socketClienteRTPTrack1 = socketClienteRTPTrack1;
}

public InetAddress getSocketClienteRTCPTrack0() {
    return socketClienteRTCPTrack0;
}

public void setSocketClienteRTCPTrack0(
    InetAddress socketClienteRTCPTrack0) {
    this.socketClienteRTCPTrack0 = socketClienteRTCPTrack0;
}

public InetAddress getSocketClienteRTCPTrack1() {
    return socketClienteRTCPTrack1;
}

public void setSocketClienteRTCPTrack1(
    InetAddress socketClienteRTCPTrack1) {
    this.socketClienteRTCPTrack1 = socketClienteRTCPTrack1;
}

/*
 * busca un agente en el DF (páginas amarillas) cuya característica
sea la
 * de transmisión RTSP
 */
private AID buscarAgenteDestino() {
    DFAgentDescription dfd = new DFAgentDescription();
    ServiceDescription sd = new ServiceDescription();
    sd.setType("RTSP");
    dfd.addServices(sd);

    SearchConstraints ALL = new SearchConstraints();
    ALL.setMaxResults(new Long(1)); // solicitamos sólo un
resultado

```

```

        try {
            DFAgentDescription[] result = DFService.search(this, dfd,
ALL);
            AID agente = new AID();
            agente = result[0].getName();
            return agente;
        } catch (FIPAException fe) {
            Log.e("Busqueda", "AgenteProxy " + fe.toString());
            return null;
        }
    }

    public AID getAgenteDestino() {
        return agenteDestino;
    }

    public void setAgenteDestino(AID agenteDestino) {
        this.agenteDestino = agenteDestino;
    }

    public UrlRTSP getUrlDestino() {
        return urlDestino;
    }

    public void setUrlDestino(UrlRTSP urlDestino) {
        this.urlDestino = urlDestino;
    }

    public UrlRTSP getUrlLocal() {
        return urlLocal;
    }

    public void setUrlLocal(UrlRTSP urlLocal) {
        this.urlLocal = urlLocal;
    }

    /*
     * (non-Javadoc)
     *
     * @see jade.wrapper.gateway.GatewayAgent#takeDown() Evento que es
llamado
     * cuando el agente va a terminar su ciclo de vida
     */
    @Override
    protected void takeDown() {
        try {
            // terminación de los comportamientos del agente

```

```

        tbf.threads.get(listenerPuertoRTPTrack0).interrupt();
        tbf.threads.get(listenerPuertoRTCPTrack0).interrupt();
        tbf.threads.get(listenerPuertoRTPTrack1).interrupt();
        tbf.threads.get(listenerPuertoRTCPTrack1).interrupt();
    } catch (Exception e) {
        Log.e("Agente Proxy", "Error Take down: " +
e.toString());
        e.printStackTrace();
    }
    super.takeDown();
    Log.i("TakeDown", "Agente Desconectado");
}

public void setSocketRTSP(Socket socketRTSP) {
    this.socketRTSP = socketRTSP;
}

public Socket getSocketRTSP() {
    return socketRTSP;
}

public DatagramSocket getSocketRTPTrack0() {
    return socketRTPTrack0;
}

public void setSocketRTPTrack0(DatagramSocket socketRTPTrack0) {
    this.socketRTPTrack0 = socketRTPTrack0;
}

public DatagramSocket getSocketRTCPTrack0() {
    return socketRTCPTrack0;
}

public void setSocketRTCPTrack0(DatagramSocket socketRTCPTrack0) {
    this.socketRTCPTrack0 = socketRTCPTrack0;
}

public DatagramSocket getSocketRTPTrack1() {
    return socketRTPTrack1;
}

public void setSocketRTPTrack1(DatagramSocket socketRTPTrack1) {
    this.socketRTPTrack1 = socketRTPTrack1;
}

public DatagramSocket getSocketRTCPTrack1() {
    return socketRTCPTrack1;
}

```



```

        public void setSocketRTCPTrack1(DatagramSocket socketRTCPTrack1) {
            this.socketRTCPTrack1 = socketRTCPTrack1;
        }
    }
}

```

3.1.1.4 Comportamientos

3.1.1.4.1.1 ListenerMensajeRTSP

```

package ec.edu.espe.proxy.comportamiento;

import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

import java.io.IOException;
import java.io.OutputStream;
import java.net.Socket;

import ec.edu.espe.proxy.AgenteProxy;
import ec.edu.espe.util.MensajeRTSP;

/**
 *
 * @author ESPE El comportamiento ListenerMensajeRTSP tiene la función de
 * recibir los mensajes ACL provenientes del AgenteProxy del
servidor,
 * con la negociación RTSP, y dicha información será filtrada, para
que
 * sea lo más transparente posible para el video reproductor
 */
public class ListenerMensajeRTSP extends CyclicBehaviour {

    private static final long serialVersionUID = 1L;
    private Socket socket; // socket TCP
    private ACLMessage mensaje; // mensaje ACL
    private MessageTemplate tipo; // plantilla para recepción de mensajes
ACL
    private AgenteProxy agente;
    private int track = 0; // bandera para configurar las pistas del
video

    public ListenerMensajeRTSP(AgenteProxy agente, Socket socket) {

```

```

        super(agente);

        this.agente = agente;
        this.socket = socket;
        this.tipo =
MessageTemplate.MatchPerformative(ACLMessage.AGREE); // tipo
                // de
                // mensaje
                // ACL
        // que recibirá este comportamiento
    }

    @Override
    public void action() {
        mensaje = myAgent.receive(tipo); // recibe los mensajes según
la
                                                //
plantilla señalada
        if (mensaje != null) {
            try {
                OutputStream out = socket.getOutputStream();
                // Filtrado del mensaje recibido UrlDestino ->
UrlLocal
                String mensajeFiltrado =
MensajeRTSP.cambiarUrlRTSP(mensaje
                .getContent(), agente.getUrlDestino(),
agente
                .getUrlLocal());
                // Intenta cambiar en el mensaje, el host del cuál
procede
                // host -> "localhost"
                mensajeFiltrado =
MensajeRTSP.cambiarHost(mensajeFiltrado,
                agente.getUrlDestino(),
agente.getUrlLocal());

                // verifica si el mensaje RTSP es de tipo Setup
                if (MensajeRTSP.isSetupRespuesta(mensajeFiltrado))
{
                    if (track == 0) { // configura al setup del
track 0,
                                                // generalmente
pista de audio
                                                // cambia los puertos RTP y RTCP del
cliente para la

```

```

// pista 0 a los locales
mensajeFiltrado =
MensajerTSP.cambiarPuertoCliente(
    mensajeFiltrado, agente
    .getSocketClienteRTPTrack0().getPort(),
    agente.getSocketClienteRTCPTrack0().getPort());
// cambia los puertos RTP y RTCP del
servidor para la
// pista 0 a los locales
mensajeFiltrado =
MensajerTSP.cambiarPuertoServidor(
    mensajeFiltrado,
    agente.getSocketRTPTrack0()
    .getLocalPort(),
    agente
    .getSocketRTCPTrack0().getLocalPort());
    track = 1; // pista a configurar
} else if (track == 1) { // configura al setup
del track 1,
//
generalmente pista de video
// cambia los puertos RTP y RTCP del
cliente para la
// pista 1 a los locales
mensajeFiltrado =
MensajerTSP.cambiarPuertoCliente(
    mensajeFiltrado, agente
    .getSocketClienteRTPTrack1().getPort(),
    agente.getSocketClienteRTCPTrack1().getPort());
// cambia los puertos RTP y RTCP del
servidor para la
// pista 0 a los locales
mensajeFiltrado =
MensajerTSP.cambiarPuertoServidor(
    mensajeFiltrado,
    agente.getSocketRTPTrack1()
    .getLocalPort(),
    agente
    .getSocketRTCPTrack1().getLocalPort());
    track = 0; // pista a configurar
}

```

```

    }
    /*
    * Muy importante, al cambiar los parámetros del
mensaje
de longitud
    * original también se debe modificar el parámetro
    * del mensaje para evitar errores en la negociación
    */
    mensajeFiltrado = MensajeRTSP

    .cambiarLongitudContenidoMensaje(mensajeFiltrado);
    out.write(mensajeFiltrado.getBytes()); // envía el
mensaje

    // filtrado al socket

    // del video reproductor
    out.flush();

    } catch (IOException e) {
        System.out.print("Mensaje no enviado: " +
e.toString());
    }

    } else {
        block(); // espera hasta que llegue un mensaje
    }

}
}

```

3.1.1.4.1.2 *ListenerMensajeUDP*

```

package ec.edu.espe.proxy.comportamiento;

import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetSocketAddress;

/**
 *

```

```

* @author ESPE El comportamiento ListenerMensajeUDP es el encargado de
recibir
* los mensajes provenientes del AgenteProxy servidor,
correspondientes
* a los protocolos RTP o RTCP Se creará un comportamiento por cada
* protocolo.
*
* RTP pista 0-> tipo de mensaje ACL Query_ref RTCP pista 0-> tipo
de
* mensaje ACL Inform_ref
*
* RTP pista 1-> tipo de mensaje ACL Query_if RTCP pista 1-> tipo
de
* mensaje ACL Inform_if
*/
public class ListenerMensajeUDP extends CyclicBehaviour {
    private static final long serialVersionUID = -2878475751457781308L;

    private DatagramSocket socket; // socket UDP
    private DatagramPacket paquete;
    private ACLMessage mensaje;
    private MessageTemplate tipo;
    private InetAddress destino; // socket destinatario

    public ListenerMensajeUDP(Agent agente, DatagramSocket socket,
        InetAddress destino, int tipoMensaje) {
        super(agente);
        this.socket = socket;
        this.destino = destino;
        this.tipo = MessageTemplate.MatchPerformative(tipoMensaje);
    }

    @Override
    public void action() {
        mensaje = myAgent.receive(tipo); // recibe un mensaje ACL según
        // de
        // protocolo
        if (mensaje != null) {
            byte[] contenido = mensaje.getByteSequenceContent(); //
            // obtiene el
            // contenido
            // del mensaje ACL
            try {
                paquete = new DatagramPacket(contenido,
                    contenido.length,
                    destino);
            }
        }
    }
}

```

```

        socket.send(paquete); // envía el paquete RTP o
RTCP al video
                                                                    // reproductor
    } catch (Exception e1) {
        e1.printStackTrace();
    }
} else {
    block(); // se bloquea hasta que llegue un mensaje
}
}
}

```

3.1.1.4.1.3 ListenerPuertoRTSP

```
package ec.edu.espe.proxy.comportamiento;
```

```
import jade.core.behaviours.Behaviour;
import jade.lang.acl.ACLMessage;
```

```
import java.io.ByteArrayOutputStream;
import java.io.DataInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetSocketAddress;
import java.net.ServerSocket;
import java.net.Socket;
```

```
import android.util.Log;
import ec.edu.espe.proxy.AgenteProxy;
import ec.edu.espe.util.MensajeRTSP;
import ec.edu.espe.util.UrlRTSP;
```

```
/**
 *
 * @author ESPE El comportamiento ListenerPuertoRTSP tiene la función de
recibir
 * la negociación RTSP proveniente del video reproductor, los
mismos que
 * deberán ser filtrados (ej. host, url, puerto), para intentar que
sean
 * los más transparentes posibles en la negociación RTSP
 *
 */
```

```
public class ListenerPuertoRTSP extends Behaviour {
    private static final long serialVersionUID = 1L;

    private Socket socketRTSP; // puerto TCP a escuchar

```

```

    private AgenteProxy agente;
    private boolean conectado; // bandera para determinar si la conexión
termina
    private int puerto; // puerto por el que escucha
    private UrlRTSP urlDestino; // dirección URL real
    private UrlRTSP urlLocal; // dirección cambiada a localhost
    private int track; // pista a configurar
    private int options = 0; // parámetro que ayudará a mantener y
prolongar la
// conexión RTSP

    public ListenerPuertoRTSP(AgenteProxy agente, UrlRTSP urlDestino,
        UrlRTSP urlLocal) {
        super(agente);
        this.track = 0;
        this.puerto = urlLocal.getPuerto();
        this.agente = agente;
        this.urlDestino = urlDestino;
        this.urlLocal = urlLocal;
        this.conectado = true;
    }

    @Override
    public void action() {
        if (socketRTSP == null || !socketRTSP.isConnected()) { //
verifica si el

            // socket está

            // conectado
            ServerSocket servidor;
            try {
                servidor = new ServerSocket(puerto);
                socketRTSP = servidor.accept(); // espera las
peticiones del
video reproductor
                conectado = true; // señala que la conexión está
activa
                myAgent
                    .addBehaviour(new
ListenerMensajeroRTSP(agente,
                    socketRTSP));
                // añade comportamiento para recibir la respuesta
del servidor
            } catch (Exception e) {
                Log.e("ListenerPuertoRTSP", "ServerSocket:" +
e.toString());

```

```

        conectado = false; // indica que no hay conexiones
activas    }
    }
    if (conectado) { // verifica si hay un conexión activa
        try {
inicia un flujo    InputStream in = socketRTSP.getInputStream(); //

                    // de entrada
                    DataInputStream din = new DataInputStream(in);
                    byte ch = din.readByte();

                    ByteArrayOutputStream baos = new
ByteArrayOutputStream();
                    // maneja el flujo de entrada en bytes
                    baos.write(ch);
                    // Añade un flujo
                    // lee hasta encontrar el fin de línea (\r\n\r\n)
                    while (!eomReached(baos.toByteArray())) {
                        baos.write(din.readByte());
                    }
                    // obtiene el tamaño del contenido del mensaje
                    int length = getContentLength(new
String(baos.toByteArray()));
                    // lee el contenido del mensaje, generalmente el
mensaje RTSP
                    // posee un contenido, y los demás comandos no.
                    for (int i = 0; i < length; i++) {
                        baos.write(din.readByte());
                    }

                    // inicializa el tipo de mensaje ACL -> request
                    ACLMessage mensajeACL = new
ACLMessage(ACLMessage.REQUEST);

                    mensajeACL.addReceiver(agente.getAgenteDestino()); // asigna el

                                // agente

                                // receptor
                                mensajeACL.setSender(agente.getAID()); // asigna el
agente

                    // emisor
                    String cadenaRTSP = baos.toString(); // almacena en
un string el

```



```

        // mensaje RTSP
        // Filtra la urlLocal por la real y así como
también el host
        cadenaRTSP = MensajeRTSP.cambiarUrlRTSP(cadenaRTSP,
urlLocal,
        urlDestino);
agente
        cadenaRTSP = MensajeRTSP.cambiarHost(cadenaRTSP,
        .getUrlLocal(), agente.getUrlDestino());

        // verifica si el mensaje es un comando SETUP
        if (MensajeRTSP.isSetup(cadenaRTSP)) {
            int puertos[] =
MensajeRTSP.getPuertosCliente(cadenaRTSP);
            if (track == 0) { // configura la pista 0
                // añade los comportamientos necesarios
                // mensajes ACL, cuyo contenido
                // son los paquetes RTP y RTCP
                agente.setSocketClienteRTPTrack0(new
InetSocketAddress(
                    "localhost", puertos[0]));
                agente
                    .setSocketClienteRTCPTrack0(new InetSocketAddress(
                        "localhost",
puertos[1]));
                myAgent.addBehaviour(new
ListenerMensajeUDP(myAgent,
                    agente.getSocketRTPTrack0(),
agente
                        .getSocketClienteRTPTrack0(),
                            ACLMessage.QUERY_REF));
                myAgent.addBehaviour(new
ListenerMensajeUDP(myAgent,
                    agente.getSocketRTCPTrack0(), agente
                        .getSocketClienteRTCPTrack0(),
                            ACLMessage.INFORM_REF));
                track = 1;
            } else if (track == 1) { // configura la pista
1
                // añade los comportamientos necesarios
                // mensajes ACL, cuyo contenido

```

```

// son los paquetes RTP y RTCP
agente.setSocketClienteRTPTrack1(new
InetSocketAddress(
    "localhost", puertos[0]));
agente

    .setSocketClienteRTCPTrack1(new InetSocketAddress(
        "localhost",
puertos[1]));
myAgent.addBehaviour(new
ListenerMensajeUDP(myAgent,
    agente.getSocketRTPTrack1(),
agente

    .getSocketClienteRTPTrack1(),
        ACLMessage.QUERY_IF));
myAgent.addBehaviour(new
ListenerMensajeUDP(myAgent,
    agente.getSocketRTCPTrack1(), agente

    .getSocketClienteRTCPTrack1(),
        ACLMessage.INFORM_IF));
    track = 0;
}
}

mensajeACL.setContent(cadenaRTSP); // asigna el
mensaje RTSP al
// mensaje ACL

// Se encarga de contestar los mensajes OPTIONS
// para evitar la desconexión
if (!MensajeRTSP.isOptions(cadenaRTSP)) {
    agente.send(mensajeACL);
} else {
    if (options < 1) {
        // se encarga de asegurar que sólo
        // a los comandos OPTIONS que lleguen
        // del comando PLAY
        agente.send(mensajeACL);
        options++;
    } else {
        // responde al comando OPTIONS
        enviarRespuesta(baos.toString());
    }
}

```

```

        options++;
    }
}
} catch (Exception e) {
    Log.i("ListenerPuertoRTSP", "RTSP Conexión
terminada");
    conectado = false;
}
}
}

/**
 * Esta función es la encargada de mantener la sesión durante una
 * desconexión ya que el video reproductor envía mensajes OPTIONS
cada
 * cierto tiempo para verificar que el servidor sigue activo.
 *
 * @param option
 * Comando OPTIONS enviado por el cliente, el cuál será
 * respondido con un mensaje plantilla, donde lo que más
se debe
 * prestar atención es a la secuencia, para evitar
alterarla CSeq
 * -> secuencia que poseen los mensajes RTSP
 */
private void enviarRespuesta(String option) {
    // obtiene la secuencia del mensaje RTSP proveniente del video
    // reproductor
    Log.i("Mensaje", option);
    int inicio = option.indexOf("CSeq: ") + "CSeq: ".length();
    Log.i("MensajeInicio", inicio + "");
    int fin = option.indexOf("\r\n", inicio);
    String sesion = option.substring(inicio, fin);

    // le asigna la secuencia a la respuesta del options
    // NOTA: cambiar el nombre del Server, según el servidor RTSP
se use
    String men = "RTSP/1.0 200 OK" + "\r\n" + "Server: VLC/1.1.4" +
"\r\n"
        + "Content-Length: 0" + "\r\n" + "CSeq: " + sesion
+ "\r\n"
        + "Public:
DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, GET_PARAMETER"
        + "\r\n" + "\r\n";
    try {
        OutputStream out = socketRTSP.getOutputStream();
        out.write(men.getBytes()); // envía la respuesta al video

```

```

// reproductor
        out.flush();
        Log.e("Mensaje", men);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

/**
 *
 * @param buffer
 * @return true si el mensaje llegó al final ("/r/n/r/n"), caso
contrario
 *         retorna false
 */
private boolean eomReached(byte buffer[]) {
    boolean endReached = false;

    int size = buffer.length;

    if (size >= 4) {
        if (buffer[size - 4] == '\r' && buffer[size - 3] == '\n'
1] == '\n') {
            && buffer[size - 2] == '\r' && buffer[size -
                endReached = true;
            }
        }

    return endReached;
}

/**
 *
 * @param msg_header
 *         Cabecera del mensaje, del cuál se obtiene el parámetro
 *         Content-length para obtener la longitud de su contenido
 * @return Número entero, que representa la longitud del contenido
 */
private int getContentLength(String msg_header) {
    int length;

    int start = msg_header.indexOf("Content-length");

    if (start == -1) {
        start = msg_header.indexOf("Content-Length");
    }

    if (start == -1) {

```

```

        length = 0;
    } else {
        start = msg_header.indexOf(':', start) + 2;

        int end = msg_header.indexOf('\r', start);

        String length_str = msg_header.substring(start, end);

        length = new Integer(length_str).intValue();
    }

    return length;
}

/*
 * (non-Javadoc)
 *
 * @see jade.core.behaviours.Behaviour#done() Se activa cuando el
 * comportamiento llega a su fin y retornará true al agente cuando la
 * conexión RTSP se haya perdido por algún problema
 */
@Override
public boolean done() {
    return !conectado;
}

/*
 * (non-Javadoc)
 *
 * @see jade.core.behaviours.Behaviour#onEnd() Se activa un poco
antes de
 * que el comportamiento termine
 */
@Override
public int onEnd() {
    try {
        socketRTSP.close();
        socketRTSP = null;
        Log.i("ListenerPuertoRTSP",
            "Comportamiento finalizado correctamente");
    } catch (Exception e) {
        Log.e("ListenerPuertoRTSP", "Comportamiento no
finalizado: "
            + e.toString());
    }
    return super.onEnd();
}
}

```

3.1.1.4.1.4 ListenerPuertoUDP

```
package ec.edu.espe.proxy.comportamiento;

import jade.core.AID;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;

import ec.edu.espe.proxy.AgenteProxy;

/**
 *
 * @author ESPE ListenerPuertoUDP es el comportamiento encargado de recibir
 * todos los paquetes RTP o RTCP provenientes del video reproductor
 * y
 * los envía al Agente proxy servidor
 *
 * RTP pista 0-> tipo de mensaje ACL Query_ref RTCP pista 0-> tipo
de
 * mensaje ACL Inform_ref
 *
 * RTP pista 1-> tipo de mensaje ACL Query_if RTCP pista 1-> tipo
de
 * mensaje ACL Inform_if
 *
 */
public class ListenerPuertoUDP extends CyclicBehaviour {
    private static final long serialVersionUID = 1L;

    private DatagramSocket socket;
    private DatagramPacket paquete;
    private AID agenteDestino;
    private int tipo;
    private ByteArrayOutputStream baos;

    public ListenerPuertoUDP(AgenteProxy agente, DatagramSocket socket,
        int tipoMensaje, AID agenteDestino) {
        super(agente);
        this.socket = socket; // socket UDP
        this.tipo = tipoMensaje; // tipo de mensaje ACL
        this.agenteDestino = agenteDestino;
    }
}
```

```

    }

    @Override
    public void action() {
        byte[] contenido = new byte[1500];
        this.paquete = new DatagramPacket(contenido, contenido.length);
        try {
            socket.receive(paquete); // recibe el datagrama del video
                                     // reproductor
            ACLMessage mensajeACL = new ACLMessage(tipo);
            mensajeACL.setSender(myAgent.getAID()); // asigna al
receptor del
            // mensaje ACL
            mensajeACL.addReceiver(agenteDestino); // asigna el
emisor del
            // mensaje ACL
            baos = new ByteArrayOutputStream(); // manejador para
arreglos de
                                                    //
bytes
            baos.write(paquete.getData(), 0, paquete.getLength()); //
escribe

            // sólo la
            // información
            // válida

            mensajeACL.setByteSequenceContent(baos.toByteArray()); //
lo asigna

            // al
            // contenido
            // del
            // mensaje

            // ACL
            myAgent.send(mensajeACL); // envía el mensaje ACL
            baos.reset();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

```

```
}  
}
```

3.1.1.5 Utilitarios

3.1.1.5.1.1 MensajeRTSP

```
package ec.edu.espe.util;  
  
import java.util.StringTokenizer;  
  
/**  
 *  
 * @author ESPE MensajeRTSP, es una clase utilitaria, que ayuda al filtrado  
de  
 * los comandos RTSP  
 */  
public class MensajeRTSP {  
  
    public final static String CRLF = "\r\n"; // retorno de carro y salto  
de //  
línea  
  
    /**  
     * obtiene la url que poseen los comandos RTSP  
     *  
     * @param mensaje  
     * .- Comando RTSP  
     * @return url del mensaje RTSP  
     */  
    // ej. OPTIONS rtps://....  
    public static String getURL(String mensaje) {  
        StringTokenizer tokens = new StringTokenizer(mensaje);  
        tokens.nextToken();  
        String url_string = tokens.nextToken();  
        return url_string;  
    }  
  
    /**  
     * localiza los puertos del cliente en los mensajes setup y los  
cambia  
     *  
     * @param mensaje  
     * .-Comando RTSP  
     * @param rtp  
     * .- puerto rtp
```



```

    * @param rtcp
    *          .- puerto rtcp
    * @return
    */
    public static String cambiarPuertoCliente(String mensaje, int rtp,
int rtcp) {
        int[] puertos = getPuertosCliente(mensaje);
        String aux = mensaje.replace("client_port=" + puertos[0] + "-"
+ puertos[1], "client_port=" + rtp + "-" + rtcp);
        return aux;
    }

/**
 *
 * @param mensaje
 *          .- comando RTSP
 * @return arreglo de los puertos RTP y RTCP el cliente obtenidos del
 *          comando RTSP
 */
    public static int[] getPuertosCliente(String mensaje) {
        int inicio = mensaje.indexOf("client_port=") +
"client_port=" .length();
        int fin = mensaje.indexOf(CRLF, inicio);
        String puertos = mensaje.substring(inicio, fin);
        if (puertos.contains(";")) {
            fin = puertos.indexOf(";");
            puertos = puertos.substring(0, fin);
        }

        int[] puertosInt = new int[] {
Integer.parseInt(puertos.split("-")[0]),
Integer.parseInt(puertos.split("-")[1]) };

        return puertosInt;
    }

/**
 * localiza los puertos del servidor en las respuestas a los mensajes
 * setup
 * y los cambia
 *
 * @param mensaje
 *          .- comando RTSP
 * @param rtp
 *          .- puerto rtp del servidor
 * @param rtcp
 *          .- puerto rtcp del servidor

```

```

    * @return mensaje RTSP filtrado
    */
    public static String cambiarPuertoServidor(String mensaje, int rtp,
int rtcp) {
        int[] puertos = getPuertosServidor(mensaje);
        String aux = mensaje.replace("server_port=" + puertos[0] + "-"
            + puertos[1], "server_port=" + rtp + "-" + rtcp);
        return aux;
    }

/**
 * Obtiene un arreglo de los puertos rtp y rtcp del servidor
 *
 * @param mensaje
 *      .- comando RTSP
 * @return.- arreglo de puertos
 */
    public static int[] getPuertosServidor(String mensaje) {
        int inicio = mensaje.indexOf("server_port=") +
"server_port=".length();
        int fin = mensaje.indexOf(";", inicio);
        String puertos = mensaje.substring(inicio, fin);
        int[] puertosInt = new int[] {
Integer.parseInt(puertos.split("-")[0]),
            Integer.parseInt(puertos.split("-")[1]) };

        return puertosInt;
    }

/**
 * Obtiene la fuente de los mensaje RTSP
 *
 * @param mensaje
 *      .- comando RTSP
 * @return.- fuente RTSP
 */
    public static String getSource(String mensaje) {
        String source = new String("");
        if (mensaje.contains("source=")) {
            int inicio = mensaje.indexOf("source=") +
"source=".length();
            int fin = mensaje.indexOf(";", inicio);
            source = mensaje.substring(inicio, fin);
        }
        return source;
    }

/**

```

```

* Cambia la fuente del comando RTSP
*
* @param mensaje
*     .- Comando RTSP
* @param source
*     .- parámetro por el cuál será cambiado
* @return Mensaje filtrado
*/
public static String cambiarSource(String mensaje, String source) {
    String aux = getSource(mensaje);
    String resultado = mensaje.replace("source=" + aux, "source=" +
source);
    return resultado;
}

/**
* Recalcula la longitud del mensaje RTSP y lo cambia en el parámetro
* Content-length
*
* @param mensaje
*     .- comando RTSP
* @return mensaje RTSP filtrado
*/
public static String cambiarLongitudContenidoMensaje(String mensaje)
{
    String mensajeCambiado;
    if (mensaje.contains("Content-Length: ")
        || mensaje.contains("Content-length: ")) {
        int longitud = getLongitudContenidoMensaje(mensaje);
        if (longitud > 0) {
            int nuevaLongitud =
getContenidoMensaje(mensaje).length();
            mensajeCambiado = mensaje.replace(
                "Content-Length: " + longitud, "Content-
Length: "
                + nuevaLongitud);
            mensajeCambiado = mensajeCambiado.replace("Content-
length: "
                + longitud, "Content-length: " +
nuevaLongitud);
            return mensajeCambiado;
        }
    }
    return mensaje;
}

/**
* Obtiene el contenido de un mensaje RTSP

```

```

*
* @param mensaje
*      .- comando RTSP
* @return contenido del comando RTSP
*/
public static String getContenidoMensaje(String mensaje) {
    int inicio = mensaje.indexOf(CRLF + CRLF);
    String cadena = mensaje.substring(inicio + (CRLF +
CRLF).length());
    return cadena;
}

/**
* Obtiene el parámetro longitud de los comandos RTSP
*
* @param mensaje
*      .- comando RTSP
* @return longitud del mensaje
*/
public static int getLongitudContenidoMensaje(String mensaje) {
    int longitud = -1;
    if (mensaje.contains("Content-Length: ")
        || mensaje.contains("Content-length: ")) {
        int inicio = mensaje.indexOf("Content-Length: ")
            + "Content-Length: ".length();
        int fin = mensaje.indexOf(CRLF, inicio);
        longitud = Integer.parseInt(mensaje.substring(inicio,
fin));
    }
    return longitud;
}

/**
* Cambia las urls para el filtrado de los comandos RTSP
*
* @param mensaje
*      .- comando rtsp
* @param url
*      .- url a ser cambiado
* @param urlSustituto
*      .- url por el cuál va a ser cambiado
* @return
*/
public static String cambiarUrlRTSP(String mensaje, UrlRTSP url,
    UrlRTSP urlSustituto) {
    String resultado = "";
    String url1 = "rtsp://" + url.getHost();
    String url2 = "rtsp://" + urlSustituto.getHost();

```

```

        if (url.getTienePuerto()) {
            url1 += ":" + url.getPuerto();
        }

        if (urlSustituto.getTienePuerto()) {
            url2 += ":" + urlSustituto.getPuerto();
        }
        resultado = mensaje.replaceAll(url1, url2);

        return resultado;
    }

    /**
     * Cambia la cadena host que se encuentren el el mensaje
     *
     * @param mensaje
     *         .- comando RTSP
     * @param url
     * @param urlSustituto
     * @return comando RTSP filtrado
     */
    public static String cambiarHost(String mensaje, UrlRTSP url,
        UrlRTSP urlSustituto) {
        String resultado = "";
        resultado = mensaje.replaceAll(url.getHost(),
urlSustituto.getHost());
        resultado = mensaje.replaceAll(url.getIp(),
urlSustituto.getIp());
        return resultado;
    }

    /**
     * Obtiene el comando RTSP
     *
     * @param mensaje
     * @return retorna el Comando RTSP
     */
    private static String getComando(String mensaje) {
        StringTokenizer tokens = new StringTokenizer(mensaje);
        String comando_string = tokens.nextToken();
        return comando_string;
    }

    public static boolean isDescribe(String mensaje) {
        if (getComando(mensaje).equalsIgnoreCase("describe")) {
            return true;
        }
    }

```

```

    }
    return false;
}

public static boolean isOptions(String mensaje) {
    if (getComando(mensaje).equalsIgnoreCase("options")) {
        return true;
    }
    return false;
}

public static boolean isPause(String mensaje) {
    if (getComando(mensaje).equalsIgnoreCase("pause")) {
        return true;
    }
    return false;
}

public static boolean isPlay(String mensaje) {
    if (getComando(mensaje).equalsIgnoreCase("play")) {
        return true;
    }
    return false;
}

public static boolean isSetup(String mensaje) {
    if (getComando(mensaje).equalsIgnoreCase("setup")) {
        return true;
    }
    return false;
}

public static boolean isSetupTrack0(String mensaje) {
    if (getComando(mensaje).equalsIgnoreCase("setup")
        && mensaje.contains("trackID=0")) {
        return true;
    }
    return false;
}

public static boolean isSetupTrack1(String mensaje) {
    if (getComando(mensaje).equalsIgnoreCase("setup")
        && mensaje.contains("trackID=1")) {
        return true;
    }
    return false;
}
}

```

```

public static boolean isTearDown(String mensaje) {
    if (getComando(mensaje).equalsIgnoreCase("teardown")) {
        return true;
    }
    return false;
}

public static boolean isSetupRespuesta(String mensaje) {
    if (mensaje.contains("server_port=")
        && mensaje.contains("client_port=")) {
        return true;
    }
    return false;
}
}

```

3.1.1.5.1.2 *UrlRTSP*

```

package ec.edu.espe.util;

import java.net.InetAddress;
import java.security.InvalidParameterException;

/**
 *
 * @author ESPE La clase utilitaria UrlRTSP, ayuda en el tratamiento de las direcciones rtsp://
 *
 */

public class UrlRTSP {
    // Estructura dirección rtsp://host:puerto/recurso
    private String host;
    private int puerto;
    private String ip;
    private String recurso;

    public UrlRTSP(String URL) throws InvalidParameterException {
        if (URL.startsWith("rtsp://")) {
            String aux = URL.substring(7);
            int fin = aux.indexOf("/");
            String dominio = aux.substring(0, fin);
            String[] cadenas = dominio.split(":");
            if (cadenas.length > 1) {
                host = cadenas[0];
            }
        }
    }
}

```

```

        puerto = Integer.parseInt(cadenas[1]);
    } else {
        host = dominio;
        puerto = 554;
    }
    try {
        ip = InetAddress.getByName(host).getHostAddress();
    } catch (Exception e) {
        throw new InvalidParameterException("URL no es un
rstp valido");
    }

    recurso = aux.substring(fin + 1);
} else {
    throw new InvalidParameterException("URL no es un rstp
valido");
}

}

public String getRecurso() {
    return recurso;
}

public void setRecurso(String recurso) {
    this.recurso = recurso;
}

public String getURL() {
    String aux = "";
    if (getPuerto() == 554) {
        aux = "rtsp://" + getHost() + "/" + getRecurso();
    } else {
        aux = "rtsp://" + getHost() + ":" + getPuerto() + "/"
            + getRecurso();
    }
    return aux;
}

public String getHost() {
    return host;
}

public void setHost(String host) {
    this.host = host;
}

public int getPuerto() {

```



```

        return puerto;
    }

    public void setPuerto(int puerto) {
        this.puerto = puerto;
    }

    public String getIp() {
        return ip;
    }

    public void setIp(String ip) {
        this.ip = ip;
    }

    public boolean getTienePuerto() {
        if (getPuerto() == 554) {
            return false;
        }
        return true;
    }
}

```

3.1.1.6 Recursos de la Aplicación

3.1.1.6.1 Layout

3.1.1.6.2 Buscar

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <EditText android:id="@+id/EditText01" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:width="220px"></EditText>

    <Button android:id="@+id/Button01" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:width="100px"
    android:text="Play"></Button>
</LinearLayout>

```

3.1.1.6.2.1 Main

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

```

```

        android:layout_width="fill_parent"
        android:layout_height="fill_parent">

<VideoView android:id="@+id/VideoView01"
android:layout_height="fill_parent"
android:layout_width="fill_parent"></VideoView>

</LinearLayout>

```

3.1.1.6.2 Menu

```

<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:title="@string/titulo_opciones_menu"
        android:icon="@drawable/ic_menu_preferences"
        android:visible="true"
        android:id="@+id/opcion_menu"
        android:enabled="true">
        </item>
<item android:title="@string/titulo_salir_menu"
        android:visible="true"
        android:id="@+id/salir_menu"
        android:enabled="true"
        android:icon="@drawable/ic_menu_close_clear_cancel">
        </item>

</menu>

```

3.1.1.6.3 Value

3.1.1.6.3.1 String

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Buscar :</string>
    <string name="app_name">Streaming Jade</string>
    <string name="titulo_salir_menu">Salir</string>
    <string name="titulo_opciones_menu">Opciones</string>
    <string name="grupo_plataforma_jade_configuracion">Plataforma
Jade</string>
    <string name="grupo_agente_proxy_configuracion">Agente Proxy</string>

    <string name="direccionIP_jade_configuracion">Dirección IP</string>
    <string name="puerto_jade_configuracion">Puerto</string>
    <string name="puerto_rtsp_configuracion">Puerto RTSP</string>

```

```

<string name="default_chk_activar_agente">true</string>
<string name="default_puerto_rtsp_configuracion">1234</string>
<string name="default_ip_servidor_configuracion">192.168.3.11</string>
<string name="default_puerto_servidor_configuracion">1099</string>
</resources>

```

3.1.1.7 Manifest

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ec.edu.espe.aplicacion"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
android:label="@string/app_name">
        <service android:name="jade.android.MicroRuntimeService" />
        <activity android:name=".BuscadorVideo"
            android:label="@string/app_name"

android:configChanges="keyboard|keyboardHidden|orientation">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
        <activity android:name="ec.edu.espe.configuracion.Configuracion"
            android:label="@string/app_name">
        </activity>

        <activity android:name=".VideoPlayer"
            android:label="@string/app_name"
            android:screenOrientation="landscape"

android:configChanges="keyboard|keyboardHidden|orientation">
        </activity>

    </application>

```

```

    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
    <uses-sdk android:minSdkVersion="8" />
<uses-permission android:name="android.permission.INTERNET"></uses-
permission>
<uses-permission
android:name="android.permission.ACCESS_CHECKIN_PROPERTIES"></uses-
permission>
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-permission>
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
<uses-permission
android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"></uses-
permission>
<uses-permission
android:name="android.permission.ACCESS_MOCK_LOCATION"></uses-permission>
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
<uses-permission
android:name="android.permission.ACCESS_SURFACE_FLINGER"></uses-permission>
<uses-permission
android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.ACCOUNT_MANAGER"></uses-
permission>
<uses-permission
android:name="android.permission.AUTHENTICATE_ACCOUNTS"></uses-permission>
<uses-permission android:name="android.permission.BATTERY_STATS"></uses-
permission>
<uses-permission android:name="android.permission.BIND_APPWIDGET"></uses-
permission>
<uses-permission
android:name="android.permission.BIND_DEVICE_ADMIN"></uses-permission>
<uses-permission
android:name="android.permission.BIND_INPUT_METHOD"></uses-permission>
<uses-permission android:name="android.permission.BLUETOOTH"></uses-
permission>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"></uses-
permission>
<uses-permission android:name="android.permission.BRICK"></uses-permission>
<uses-permission
android:name="android.permission.BROADCAST_PACKAGE_REMOVED"></uses-
permission>
<uses-permission android:name="android.permission.BROADCAST_SMS"></uses-
permission>
<uses-permission android:name="android.permission.BROADCAST_STICKY"></uses-
permission>

```

```

<uses-permission
android:name="android.permission.BROADCAST_WAP_PUSH"></uses-permission>
<uses-permission android:name="android.permission.CALL_PHONE"></uses-
permission>
<uses-permission android:name="android.permission.CALL_PRIVILEGED"></uses-
permission>
<uses-permission
android:name="android.permission.CHANGE_COMPONENT_ENABLED_STATE"></uses-
permission>
<uses-permission
android:name="android.permission.CHANGE_CONFIGURATION"></uses-permission>
<uses-permission
android:name="android.permission.CHANGE_NETWORK_STATE"></uses-permission>
<uses-permission
android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE"></uses-
permission>
<uses-permission
android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.CLEAR_APP_CACHE"></uses-
permission>
<uses-permission
android:name="android.permission.CLEAR_APP_USER_DATA"></uses-permission>
<uses-permission
android:name="android.permission.CONTROL_LOCATION_UPDATES"></uses-
permission>
<uses-permission
android:name="android.permission.DELETE_CACHE_FILES"></uses-permission>
<uses-permission android:name="android.permission.DELETE_PACKAGES"></uses-
permission>
<uses-permission android:name="android.permission.DEVICE_POWER"></uses-
permission>
<uses-permission android:name="android.permission.DIAGNOSTIC"></uses-
permission>
<uses-permission android:name="android.permission.DISABLE_KEYGUARD"></uses-
permission>
<uses-permission android:name="android.permission.DUMP"></uses-permission>
<uses-permission
android:name="android.permission.EXPAND_STATUS_BAR"></uses-permission>
<uses-permission android:name="android.permission.FACTORY_TEST"></uses-
permission>
<uses-permission android:name="android.permission.FLASHLIGHT"></uses-
permission>
<uses-permission android:name="android.permission.FORCE_BACK"></uses-
permission>
<uses-permission android:name="android.permission.GET_ACCOUNTS"></uses-
permission>
<uses-permission android:name="android.permission.GET_PACKAGE_SIZE"></uses-
permission>

```

```
<uses-permission android:name="android.permission.GET_TASKS"></uses-  
permission>  
<uses-permission android:name="android.permission.GLOBAL_SEARCH"></uses-  
permission>  
<uses-permission android:name="android.permission.HARDWARE_TEST"></uses-  
permission>  
<uses-permission android:name="android.permission.INJECT_EVENTS"></uses-  
permission>  
<uses-permission  
android:name="android.permission.INSTALL_LOCATION_PROVIDER"></uses-  
permission>  
<uses-permission android:name="android.permission.INSTALL_PACKAGES"></uses-  
permission>  
<uses-permission  
android:name="android.permission.INTERNAL_SYSTEM_WINDOW"></uses-permission>  
<uses-permission  
android:name="android.permission.KILL_BACKGROUND_PROCESSES"></uses-  
permission>  
<uses-permission android:name="android.permission.MANAGE_ACCOUNTS"></uses-  
permission>  
<uses-permission  
android:name="android.permission.MANAGE_APP_TOKENS"></uses-permission>  
<uses-permission android:name="android.permission.MASTER_CLEAR"></uses-  
permission>  
<uses-permission  
android:name="android.permission.MODIFY_AUDIO_SETTINGS"></uses-permission>  
<uses-permission  
android:name="android.permission.MODIFY_PHONE_STATE"></uses-permission>  
<uses-permission  
android:name="android.permission.MOUNT_FORMAT_FILESYSTEMS"></uses-  
permission>  
<uses-permission  
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"></uses-  
permission>  
<uses-permission  
android:name="android.permission.PERSISTENT_ACTIVITY"></uses-permission>  
<uses-permission  
android:name="android.permission.PROCESS_OUTGOING_CALLS"></uses-permission>  
<uses-permission android:name="android.permission.READ_CALENDAR"></uses-  
permission>  
<uses-permission android:name="android.permission.READ_CONTACTS"></uses-  
permission>  
<uses-permission  
android:name="android.permission.READ_FRAME_BUFFER"></uses-permission>  
<uses-permission  
android:name="com.android.browser.permission.READ_HISTORY_BOOKMARKS"></uses-  
permission>
```

```

<uses-permission android:name="android.permission.READ_INPUT_STATE"></uses-
permission>
<uses-permission android:name="android.permission.READ_LOGS"></uses-
permission>
<uses-permission android:name="android.permission.READ_OWNER_DATA"></uses-
permission>
<uses-permission android:name="android.permission.READ_PHONE_STATE"></uses-
permission>
<uses-permission android:name="android.permission.READ_SMS"></uses-
permission>
<uses-permission
android:name="android.permission.READ_SYNC_SETTINGS"></uses-permission>
<uses-permission android:name="android.permission.READ_SYNC_STATS"></uses-
permission>
<uses-permission android:name="android.permission.REBOOT"></uses-
permission>
<uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED"></uses-permission>
<uses-permission android:name="android.permission.RECEIVE_MMS"></uses-
permission>
<uses-permission android:name="android.permission.RECEIVE_SMS"></uses-
permission>
<uses-permission android:name="android.permission.RECEIVE_WAP_PUSH"></uses-
permission>
<uses-permission android:name="android.permission.RECORD_AUDIO"></uses-
permission>
<uses-permission android:name="android.permission.REORDER_TASKS"></uses-
permission>
<uses-permission android:name="android.permission.RESTART_PACKAGES"></uses-
permission>
<uses-permission android:name="android.permission.SEND_SMS"></uses-
permission>
<uses-permission
android:name="android.permission.SET_ACTIVITY_WATCHER"></uses-permission>
<uses-permission
android:name="android.permission.SET_ALWAYS_FINISH"></uses-permission>
<uses-permission
android:name="android.permission.SET_ANIMATION_SCALE"></uses-permission>
<uses-permission android:name="android.permission.SET_DEBUG_APP"></uses-
permission>
<uses-permission android:name="android.permission.SET_ORIENTATION"></uses-
permission>
<uses-permission
android:name="android.permission.SET_PREFERRED_APPLICATIONS"></uses-
permission>
<uses-permission
android:name="android.permission.SET_PROCESS_LIMIT"></uses-permission>

```

```

<uses-permission android:name="android.permission.SET_TIME"></uses-
permission>
<uses-permission android:name="android.permission.SET_TIME_ZONE"></uses-
permission>
<uses-permission android:name="android.permission.SET_WALLPAPER"></uses-
permission>
<uses-permission
android:name="android.permission.SET_WALLPAPER_HINTS"></uses-permission>
<uses-permission
android:name="android.permission.SIGNAL_PERSISTENT_PROCESSES"></uses-
permission>
<uses-permission android:name="android.permission.STATUS_BAR"></uses-
permission>
<uses-permission
android:name="android.permission.SUBSCRIBED_FEEDS_READ"></uses-permission>
<uses-permission
android:name="android.permission.SUBSCRIBED_FEEDS_WRITE"></uses-permission>
<uses-permission
android:name="android.permission.SYSTEM_ALERT_WINDOW"></uses-permission>
<uses-permission
android:name="android.permission.UPDATE_DEVICE_STATS"></uses-permission>
<uses-permission android:name="android.permission.USE_CREDENTIALS"></uses-
permission>
</manifest>

```

3.1.2 Servidor Proxy - APS

3.1.2.1 Contenedor

```
package ec.edu.espe.contenedor;
```

```

import jade.core.Profile;
import jade.core.ProfileImpl;
import jade.core.Runtime;
import jade.wrapper.AgentContainer;
import jade.wrapper.AgentController;
import jade.wrapper.ControllerException;
import jade.wrapper.StaleProxyException;
/**
 *
 * @author ESPE
 * Esta clase es una alternativa de iniciar jade, y que nos servirá para
poder
 * crear n agentes para cubrir con las necesidades de los clientes
 */
public class AplicacionJade {

```



```

private static int numero=0;

public static void main(String[] args) {
    numero=0;
    Runtime rt=Runtime.instance(); //obtiene una instancia de JADE
    Profile propiedad = new ProfileImpl(); //asigna las propiedades
para su inicialización
    propiedad.setParameter(Profile.MAIN_HOST, "192.168.3.11");
    propiedad.setParameter(Profile.MAIN_PORT, "1099");
    AgentContainer contenedorPrincipal=
rt.createMainContainer(propiedad); //crea un contenedor principal

    Object reference = new Object();
    Object agentObject[] = new Object[1];
    agentObject[0]=reference;
    try {
        contenedorPrincipal.start(); //inicia el contenedor
        crearAgentes(5, contenedorPrincipal); //crea un número de
agentes
    } catch (ControllerException e) {
        e.printStackTrace();
    }
}
/**
 *
 * @param numero. cantidad de agentes que se desea crear
 * @param contenedor. a que contenedor van a pertenecer los agentes
 */
public static void crearAgentes(int numero,AgentContainer
contenedor){
    for(int i=0;i<numero;i++){
        try {
            AgentController agente =
contenedor.createNewAgent("srv"+(i+1), "ec.edu.espe.proxy.AgenteProxy",
null); //creación de agentes
            agente.start(); //inicialización de los agentes
        } catch (StaleProxyException e) {
            e.printStackTrace();
        }
    }
}
}

```

3.1.2.2 Agente Proxy

```

package ec.edu.espe.proxy;

```

```

import jade.core.Agent;
import jade.domain.DFService;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;

import java.net.DatagramSocket;
import java.net.InetSocketAddress;
import java.net.Socket;

import ec.edu.espe.proxy.comportamiento.ListenerMensajeRTSP;
import ec.edu.espe.util.UrlRTSP;

public class AgenteProxy extends Agent {
    /**
     *
     * @author ESPE
     * La clase AgenteProxy es la puerta de enlace entre el Servidor RTSP
y
     * el AgenteProxy del cliente. Sus comportamientos serán los
encargados de asegurar que los flujos
     * de información lleguen a su destinatario.
     */
    private static final long serialVersionUID = 1L;

    private Socket socketRTSP;
    /**
     * *****Variables para Track 0*****
     */
    //-----RTP-----
    private DatagramSocket socketRTPTrack0;
    private InetSocketAddress socketClienteRTPTrack0;
    //-----RTCP-----
    private DatagramSocket socketRTCPTrack0;
    private InetSocketAddress socketClienteRTCPTrack0;

    /**
     * *****Variables para Track 1*****
     */
    //-----RTP-----
    private DatagramSocket socketRTPTrack1;
    private InetSocketAddress socketClienteRTPTrack1;
    //-----RTCP-----
    private DatagramSocket socketRTCPTrack1;
    private InetSocketAddress socketClienteRTCPTrack1;

    private UrlRTSP urlDestino=null; //rtsp://host:puerto/recurso

```

```

    private UrlRTSP urlLocal; //rtsp://localhost:puerto/recurso
(urlDestino modificado)

@Override
protected void setup() {
    super.setup();
    //Registrar en las páginas amarillas el servicio que ofrece el
agente
    DFAgentDescription dfd = new DFAgentDescription();
    dfd.setName( getAID() );
    ServiceDescription sd = new ServiceDescription();
    sd.setType( "RTSP" );//Servicio que ofrece el agente
    sd.setName( "RTSP" );
    dfd.addServices(sd);
    try {
        DFService.register(this, dfd);
        //Obtención de puertos aleatoriamente y disponible para
RTP y RTCP
        socketRTPTrack0=new DatagramSocket();
        socketRTCPTrack0=new DatagramSocket();
        socketRTPTrack1=new DatagramSocket();
        socketRTCPTrack1=new DatagramSocket();
        addBehaviour(new ListenerMensajerTSP(this));
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public InetSocketAddress getSocketClienteRTPTrack0() {
    return socketClienteRTPTrack0;
}

public void setSocketClienteRTPTrack0(InetSocketAddress
socketClienteRTPTrack0) {
    this.socketClienteRTPTrack0 = socketClienteRTPTrack0;
}

public InetSocketAddress getSocketClienteRTPTrack1() {
    return socketClienteRTPTrack1;
}

public void setSocketClienteRTPTrack1(InetSocketAddress
socketClienteRTPTrack1) {
    this.socketClienteRTPTrack1 = socketClienteRTPTrack1;
}

public InetSocketAddress getSocketClienteRTCPTrack0() {

```

```

        return socketClienteRTCPTrack0;
    }

    public void setSocketClienteRTCPTrack0(InetSocketAddress
socketClienteRTCPTrack0) {
        this.socketClienteRTCPTrack0 = socketClienteRTCPTrack0;
    }

    public InetSocketAddress getSocketClienteRTCPTrack1() {
        return socketClienteRTCPTrack1;
    }

    public void setSocketClienteRTCPTrack1(InetSocketAddress
socketClienteRTCPTrack1) {
        this.socketClienteRTCPTrack1 = socketClienteRTCPTrack1;
    }

    public UrlRTSP getUrlDestino() {
        return urlDestino;
    }

    public void setUrlDestino(UrlRTSP urlDestino) {
        this.urlDestino = urlDestino;
    }

    public UrlRTSP getUrlLocal() {
        return urlLocal;
    }

    public void setUrlLocal(UrlRTSP urlLocal) {
        this.urlLocal = urlLocal;
    }

    @Override
    protected void takeDown() {
        try {
            try {
                //Quita del registro de las páginas amarillas el
servicio ofrecido
                DFService.deregister(this); }
            catch (Exception e) {
                System.out.println(e.toString());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        super.takeDown();
    }

```

```

}

public void setSocketRTSP(Socket socketRTSP) {
    this.socketRTSP = socketRTSP;
}

public Socket getSocketRTSP() {
    return socketRTSP;
}

public DatagramSocket getSocketRTPTrack0() {
    return socketRTPTrack0;
}

public void setSocketRTPTrack0(DatagramSocket socketRTPTrack0) {
    this.socketRTPTrack0 = socketRTPTrack0;
}

public DatagramSocket getSocketRTCPTrack0() {
    return socketRTCPTrack0;
}

public void setSocketRTCPTrack0(DatagramSocket socketRTCPTrack0) {
    this.socketRTCPTrack0 = socketRTCPTrack0;
}

public DatagramSocket getSocketRTPTrack1() {
    return socketRTPTrack1;
}

public void setSocketRTPTrack1(DatagramSocket socketRTPTrack1) {
    this.socketRTPTrack1 = socketRTPTrack1;
}

public DatagramSocket getSocketRTCPTrack1() {
    return socketRTCPTrack1;
}

public void setSocketRTCPTrack1(DatagramSocket socketRTCPTrack1) {
    this.socketRTCPTrack1 = socketRTCPTrack1;
}

```

```
}  
}
```

3.1.2.3 Comportamientos

3.1.2.3.1 ListenerMensajeRTSP

```
package ec.edu.espe.proxy.comportamiento;  
  
import jade.core.behaviours.CyclicBehaviour;  
import jade.core.behaviours.ThreadedBehaviourFactory;  
import jade.lang.acl.ACLMessage;  
import jade.lang.acl.MessageTemplate;  
  
import java.io.IOException;  
import java.io.OutputStream;  
import java.net.Socket;  
  
import ec.edu.espe.proxy.AgenteProxy;  
import ec.edu.espe.util.MensajeRTSP;  
import ec.edu.espe.util.UrlRTSP;  
/**  
 *  
 * @author ESPE  
 * El comportamiento ListenerMensajeRTSP tiene la función de recibir los  
mensajes  
 * ACL provenientes del AgenteProxy del cliente, con la negociación RTSP, y  
dicha  
 * información será filtrada, para que sea lo más transparente posible para  
el  
 * servidor RTSP  
 */  
public class ListenerMensajeRTSP extends CyclicBehaviour {  
  
    private static final long serialVersionUID = 1L;  
    private Socket socket;//socket TCP  
    private ACLMessage mensaje;//mensaje ACL  
    private MessageTemplate tipo;//plantilla para recepción de mensajes  
ACL  
    private AgenteProxy agente;  
    private ThreadedBehaviourFactory tbf; //convierte en hilo a los  
comportamientos  
    private UrlRTSP direccion; //UrlReal  
    private int track;//pista a configurar  
  
    public ListenerMensajeRTSP(AgenteProxy agente) {  
        super(agente);  
    }  
}
```

```

        this.track=0;
        this.agente=agente;

        this.tipo=MessageTemplate.MatchPerformative(ACLMessage.REQUEST);
        tbf=new ThreadedBehaviourFactory();
    }

    @Override
    public void action() {
        mensaje=myAgent.receive(tipo); //recepción del mensaje

        if(mensaje!=null){
            try {
                System.out.println("*****Cliente
envía*****");
                if(MensajeRTSP.isOptions(mensaje.getContent())){
//verifica si el mensaje RTSP
                    //es comando OPTIONS
                    String
url=MensajeRTSP.getURL(mensaje.getContent());
                    System.out.println("URL: " + url);
                    direccion=new UrlRTSP(url);
                    socket=new Socket(direccion.getIp(),
direccion.getPuerto());//Abre el puerto TCP a comunicar
                    myAgent.addBehaviour(tbf.wrap(new
ListenerPuertoRTSP(agente,direccion.getHost(),socket,mensaje.createReply()
));
                    //añade el comportamiento para escuchar al
puerto RTSP del servidorRTSP
                }

                String mensajeFiltrado=mensaje.getContent();

                if(MensajeRTSP.isSetup(mensaje.getContent())){
//verifica si el mensaje es de tipo SETUP
                    System.out.println(">>>>Setup - Track
"+track+">>>>");

                    if(track==0){ //pista 0 a configurar
                        //añade los comportamientos para
escuchar los puertos RTP y RTCP de la pista 0

                        mensajeFiltrado=MensajeRTSP.cambiarPuertoCliente(mensajeFiltrado,agen
te.getSocketRTPTrack0().getLocalPort(),agente.getSocketRTCPTrack0().getLocalPort());

                        myAgent.addBehaviour(tbf.wrap(new
ListenerPuertoUDP(myAgent,agente.getSocketRTPTrack0(),ACLMessage.QUERY_REF,
mensaje.createReply())));

```

```

        myAgent.addBehaviour(tbf.wrap(new
ListenerPuertoUDP(myAgent, agente.getSocketRTCPTrack0(), ACLMessage.INFORM_RE
F, mensaje.createReply())));
        track=1;
    }else if(track==1){ //pista 1 a configurar
        //añade los comportamientos para
escuchar los puertos RTP y RTCP de la pista 1

        mensajeFiltrado=MensajeRTSP.cambiarPuertoCliente(mensajeFiltrado, agen
te.getSocketRTPTrack1().getLocalPort(), agente.getSocketRTCPTrack1().getLoca
lPort());
        myAgent.addBehaviour(tbf.wrap(new
ListenerPuertoUDP(myAgent, agente.getSocketRTPTrack1(), ACLMessage.QUERY_IF, m
ensaje.createReply())));
        myAgent.addBehaviour(tbf.wrap(new
ListenerPuertoUDP(myAgent, agente.getSocketRTCPTrack1(), ACLMessage.INFORM_IF
, mensaje.createReply())));
        track=0;
    }
}

OutputStream out = socket.getOutputStream(); //crea un
flujo de salida

System.out.println(mensajeFiltrado);
System.out.println("");
out.write(mensajeFiltrado.getBytes()); //envía el mensaje
al servidor RTSP
out.flush();

} catch (IOException e) {
    System.out.print("Mensaje no enviado: "+e.toString());
}

}else{
    block(); //Espera a la llegada de un mensaje nuevo
}

}
}

```

3.1.2.3.2 ListenerMensajeUDP

package ec.edu.espe.proxy.comportamiento;


```

import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetSocketAddress;
/**
 *
 * @author ESPE
 * El comportamiento ListenerMensajeUDP es el encargado de recibir los
mensajes provenientes
 * del AgenteProxy cliente, correspondientes a los protocolos RTP o RTCP
 * Se creará un comportamiento por cada protocolo.
 *
 * RTP pista 0-> tipo de mensaje ACL Query_ref
 * RTCP pista 0-> tipo de mensaje ACL Inform_ref
 *
 * RTP pista 1-> tipo de mensaje ACL Query_if
 * RTCP pista 1-> tipo de mensaje ACL Inform_if
 */
public class ListenerMensajeUDP extends CyclicBehaviour {
    private static final long serialVersionUID = -2878475751457781308L;

    private DatagramSocket socket;//socket UDP
    private DatagramPacket paquete;
    private ACLMessage mensaje;
    private MessageTemplate tipo;
    private InetSocketAddress destino;//socket destinatario

    public ListenerMensajeUDP(Agent agente, DatagramSocket
socket,InetSocketAddress destino, int tipoMensaje) {
        super(agente);
        this.socket=socket;
        this.destino=destino;
        this.tipo=MessageTemplate.MatchPerformative(tipoMensaje);
    }

    @Override
    public void action() {
        mensaje=myAgent.receive(tipo);//recibe un mensaje ACL según el
tipo de protocolo
        if(mensaje!=null){
            byte[]
contenido=mensaje.getBytesSequenceContent();//obtiene el contenido
//del mensaje ACL
            try {

```

```

        paquete=new DatagramPacket(contenido,
contenido.length,destino);
        socket.send(paquete);//envía el paquete RTP o RTCP
al Servidor rtsp
    } catch (Exception e1) {
        e1.printStackTrace();
    }
} else {
    block();//espera a que exista un mensaje nuevo
}
}
}

```

3.1.2.3.3 ListenerPuertoRTSP

```

package ec.edu.espe.proxy.comportamiento;

import jade.core.behaviours.Behaviour;
import jade.lang.acl.ACLMessage;

import java.io.ByteArrayOutputStream;
import java.io.DataInputStream;
import java.io.InputStream;
import java.net.InetSocketAddress;
import java.net.Socket;

import ec.edu.espe.proxy.AgenteProxy;
import ec.edu.espe.util.MensajeRTSP;
/**
 *
 * @author ESPE
 * El comportamiento ListenerPuertoRTSP tiene la función de recibir la
negociación
 * RTSP proveniente del servidor RTSP, los mismos que deberán ser filtrados
 * (ej. host, url, puerto), para intentar que sean los más transparentes
posibles
 * en la negociación RTSP
 */
public class ListenerPuertoRTSP extends Behaviour{
    private static final long serialVersionUID = 1L;

    private Socket socketRTSP;//puerto TCP a escuchar

```

```

    private boolean conectado; //bandera para determinar si la conexión
termina
    private ACLMessage mensaje;
    private AgenteProxy agente;
    private String hostDestino;
    private int track=0; //pista a configurar

    public ListenerPuertoRTSP(AgenteProxy agente, String
hostDestino, Socket socket, ACLMessage mensaje) {
        super(agente);
        socketRTSP=socket;
        conectado=true;
        this.mensaje=mensaje;
        this.agente=agente;
        this.hostDestino=hostDestino;
    }

    @Override
    public void action() {
        if(socketRTSP==null || !socketRTSP.isConnected()){//verifica si
el socket está conectado
            conectado=false;
        }else{
            conectado=true;
        }
        if(conectado){//verifica si hay un conexión activa

            try {
                InputStream in = socketRTSP.getInputStream();//inicia un
flujo de entrada
                DataInputStream din = new DataInputStream(in);
                byte ch = din.readByte();

                ByteArrayOutputStream baos = new ByteArrayOutputStream();
                //maneja el flujo de entrada en bytes
                baos.write(ch);
                //Añade un flujo
                //lee hasta encontrar el fin de línea (\r\n\r\n)
                while (!eomReached(baos.toByteArray())) {
                    baos.write(din.readByte());
                }
                //obtiene el tamaño del contenido del mensaje
                int length = getContentLength(new
String(baos.toByteArray()));
                //lee el contenido del mensaje, generalmente el mensaje
RTSP

```

```

        //posee un contenido, y los demás comandos no.
        for (int i = 0; i < length; i++) {
            baos.write(din.readByte());
        }

        String mensajeFiltrado=baos.toString(); //obtención del
mensaje completo
        if(MensajeRTSP.isSetupRespuesta(mensajeFiltrado)){
//verificación si es una respuesta al Comando SETUP
            int[]
puertos=MensajeRTSP.getPuertosServidor(mensajeFiltrado);
            if(track==0){
                //configura la pista 0
                //añade los comportamientos necesarios para
recibir los mensajes ACL, cuyo contenido
                //son los paquetes RTP y RTCP
                myAgent.addBehaviour(new
ListenerMensajeUDP(myAgent, agente.getSocketRTPTrack0(), new
InetSocketAddress(hostDestino, puertos[0]),ACLMessage.QUERY_REF));
                myAgent.addBehaviour(new
ListenerMensajeUDP(myAgent, agente.getSocketRTCPTrack0(), new
InetSocketAddress(hostDestino, puertos[1]),ACLMessage.INFORM_REF));
                track=1;
            }else if(track==1){
                //configura la pista 1
                //añade los comportamientos necesarios para
recibir los mensajes ACL, cuyo contenido
                //son los paquetes RTP y RTCP
                myAgent.addBehaviour(new
ListenerMensajeUDP(myAgent, agente.getSocketRTPTrack1(), new
InetSocketAddress(hostDestino, puertos[0]),ACLMessage.QUERY_IF));
                myAgent.addBehaviour(new
ListenerMensajeUDP(myAgent, agente.getSocketRTCPTrack1(), new
InetSocketAddress(hostDestino, puertos[1]),ACLMessage.INFORM_IF));
                track=0;
            }
        }
        mensaje.setContent(baos.toString()); //asigna el conteido
al mensaje ACL
        mensaje.setPerformative(ACLMessage.AGREE); //tipo de
mensaje acl
        System.out.println("*****Servidor
envía*****");
        System.out.println(mensaje.getContent());
        System.out.println("");
        myAgent.send(mensaje); //envía el mensaje
    }
}

```

```

        catch (Exception e) {
            System.out.println( "RTSP Connection terminated");
            conectado= false;
        }
    }
}
/**
 *
 * @param buffer
 * @return true si el mensaje llegó al final ("/r/n/r/n"), caso
contrario retorna false
 */
private boolean eomReached(byte buffer[]) {
    boolean endReached = false;

    int size = buffer.length;

    if (size >= 4) {
        if (buffer[size - 4] == '\r' && buffer[size - 3] == '\n' &&
            buffer[size - 2] == '\r' && buffer[size - 1] == '\n') {
            endReached = true;
        }
    }

    return endReached;
}
/**
 *
 * @param msg_header
 * Cabecera del mensaje, del cuál se obtiene el parámetro Content-
length
 * para obtener la longitud de su contenido
 * @return Número entero, que representa la longitud del contenido
 */
private int getContentLength(String msg_header) {
    int length;

    int start = msg_header.indexOf("Content-length");

    if (start == -1) {
        start = msg_header.indexOf("Content-Length");
    }

    if (start == -1) {
        length = 0;
    } else {
        start = msg_header.indexOf(':', start) + 2;
    }
}

```

```

        int end = msg_header.indexOf('\r', start);

        String length_str = msg_header.substring(start, end);

        length = new Integer(length_str).intValue();
    }

    return length;
}

/*
 * (non-Javadoc)
 * @see jade.core.behaviours.Behaviour#done()
 * Se activa cuando el comportamiento llega a su fin
 * y retornará true al agente cuando la conexión RTSP se haya perdido
por algún problema
 */
@Override
public boolean done() {
    return !conectado;
}

/*
 * (non-Javadoc)
 * @see jade.core.behaviours.Behaviour#onEnd()
 * Se activa un poco antes de que el comportamiento termine
 */
@Override
public int onEnd() {
    try {
        socketRTSP.close();
        socketRTSP=null;
        System.out.println("ListenerPuertoRTSP Comportamiento
finalizado correctamente");
        myAgent.doDelete();
    } catch (Exception e) {
        System.err.println("ListenerPuertoRTSP Comportamiento no
finalizado: "+e.toString());
    }
    myAgent.doDelete(); //elimina al agente
    return super.onEnd();
}
}
}

```

3.1.2.3.4 ListenerPuertoUDP

```

package ec.edu.espe.proxy.comportamiento;

```

```

import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
/**
 *
 * @author ESPE
 * ListenerPuertoUDP es el comportamiento encargado de recibir todos los
paquetes
 * RTP o RTCP provenientes del servidor RTSP y los envía al Agente proxy
cliente
 *
 * RTP pista 0-> tipo de mensaje ACL Query_ref
 * RTCP pista 0-> tipo de mensaje ACL Inform_ref
 *
 * RTP pista 1-> tipo de mensaje ACL Query_if
 * RTCP pista 1-> tipo de mensaje ACL Inform_if
 *
 */
public class ListenerPuertoUDP extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    private DatagramSocket socket;//socket UDP
    private DatagramPacket paquete;
    private int tipo;
    private ACLMessage mensaje;//tipo de mensaje ACL
    private ByteArrayOutputStream baos;

    public ListenerPuertoUDP(Agent agente, DatagramSocket socket, int
tipoMensaje, ACLMessage destino) {
        super(agente);
        this.socket=socket;
        this.tipo=tipoMensaje;
        mensaje=destino;
    }

    @Override
    public void action() {
        byte[] contenido=new byte[1500];
        this.paquete=new DatagramPacket(contenido, contenido.length);
        try {
servidor RTSP
            socket.receive(paquete);//recibe el datagrama del
            mensaje.setPerformative(tipo);

```

```

        baos=new ByteArrayOutputStream();//manejador para
arreglos de bytes
        baos.write(paquete.getData(), 0,
paquete.getLength());//escribe sólo la información válida
        mensaje.setByteSequenceContent(baos.toByteArray());//lo
asigna al contenido del mensaje ACL
        myAgent.send(mensaje);//envía el mensaje ACL
        baos.reset();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```

3.1.2.4 Utilitarios

3.1.2.4.1 MensajeRTSP

```

package ec.edu.espe.util;

import java.util.StringTokenizer;

/**
 *
 * @author ESPE
 * MensajeRTSP, es una clase utilitaria, que ayuda al filtrado de los
comandos
 * RTSP
 */
public class MensajeRTSP {

    private static String CRLF = "\r\n";//retorno de carro y salto de
línea
    /**
     * obtiene la url que poseen los comandos RTSP
     * @param mensaje.- Comando RTSP
     * @return url del mensaje RTSP
     */
    //ej. OPTIONS rtps://....
    public static String getURL(String mensaje){
        StringTokenizer tokens = new StringTokenizer(mensaje);
        tokens.nextToken();
        String url_string = tokens.nextToken();
        return url_string;
    }
}

```



```

    }
    /**
     * localiza los puertos del cliente en los mensajes setup y los
cambia
     * @param mensaje.- Comando RTSP
     * @param rtp.- puerto rtp
     * @param rtcp.- puerto rtcp
     * @return
     */
    public static String cambiarPuertoCliente(String mensaje, int rtp, int
rtcp) {
        int[] puertos= getPuertosCliente(mensaje);
        String aux=mensaje.replace("client_port="+puertos[0]+"-
"+puertos[1], "client_port="+rtp+"-"+rtcp);
        return aux;
    }

    /**
     *
     * @param mensaje.- comando RTSP
     * @return arreglo de los puertos RTP y RTCP el cliente obtenidos del
comando RTSP
     */
    public static int[] getPuertosCliente(String mensaje) {
        int
inicio=mensaje.indexOf("client_port=")+"client_port=".length();
        int fin=mensaje.indexOf(CRLF, inicio);
        String puertos=mensaje.substring(inicio, fin);
        if(puertos.contains(";")){
            fin=puertos.indexOf(";");
            puertos=puertos.substring(0, fin);
        }

        int[] puertosInt=new int[]{
Integer.parseInt(puertos.split("-")[0]),
Integer.parseInt(puertos.split("-")[1])
};

        return puertosInt;
    }
    /**
     * localiza los puertos del servidor en las respuestas a los mensajes
setup y los cambia
     * @param mensaje.- comando RTSP
     * @param rtp .- puerto rtp del servidor

```

```

    * @param rtcp.- puerto rtcp del servidor
    * @return mensaje RTSP filtrado
    */
    public static String cambiarPuertoServidor(String mensaje,int rtp,int
rtcp) {
        int[] puertos= getPuertosServidor(mensaje);
        String aux=mensaje.replace("server_port="+puertos[0]+"-
"+puertos[1],"server_port="+rtp+"-"+rtcp);
        return aux;
    }
    /**
    * Obtiene un arreglo de los puertos rtp y rtcp del servidor
    * @param mensaje.- comando RTSP
    * @return.- arreglo de puertos
    */
    public static int[] getPuertosServidor(String mensaje) {
        int
inicio=mensaje.indexOf("server_port=")+"server_port=".length();
        int fin=mensaje.indexOf(";",inicio);
        String puertos=mensaje.substring(inicio, fin);
        int[] puertosInt=new int[]{

Integer.parseInt(puertos.split("-")[0]),

Integer.parseInt(puertos.split("-")[1])

        };

        return puertosInt;
    }
    /**
    * Obtiene la fuente de los mensaje RTSP
    * @param mensaje.- comando RTSP
    * @return.- fuente RTSP
    */
    public static String getSource(String mensaje) {
        String source=new String("");
        if(mensaje.contains("source=")){
            int inicio=mensaje.indexOf("source=")+"source=".length();
            int fin=mensaje.indexOf(";",inicio);
            source=mensaje.substring(inicio, fin);
        }
        return source;
    }
    /**
    * Cambia la fuente del comando RTSP
    * @param mensaje.- Comando RTSP
    * @param source.- parámetro por el cuál será cambiado
    * @return Mensaje filtrado

```

```

    */
    public static String cambiarSource(String mensaje,String source){
        String aux=getSource(mensaje);
        String resultado=mensaje.replace("source="+aux,
"source="+source);
        return resultado;
    }
    /**
     * Recalcula la longitud del mensaje RTSP y lo cambia en el parámetro
Content-length
     * @param mensaje.- comando RTSP
     * @return mensaje RTSP filtrado
     */
    public static String cambiarLongitudContenidoMensaje(String mensaje){
        String mensajeCambiado;
        if(mensaje.contains("Content-Length: ") ||
mensaje.contains("Content-length: ")){
            int longitud=getLongitudContenidoMensaje(mensaje);
            if(longitud>0){
                int
nuevaLongitud=getContenidoMensaje(mensaje).length();
                mensajeCambiado=mensaje.replace("Content-Length:
"+longitud, "Content-Length: "+nuevaLongitud);
                mensajeCambiado=mensajeCambiado.replace("Content-
length: "+longitud, "Content-length: "+nuevaLongitud);
                return mensajeCambiado;
            }
        }
        return mensaje;
    }
    /**
     * Obtiene el contenido de un mensaje RTSP
     * @param mensaje.- comando RTSP
     * @return contenido del comando RTSP
     */
    public static String getContenidoMensaje(String mensaje){
        int inicio=mensaje.indexOf(CRLF+CRLF);
        String cadena=mensaje.substring(inicio+(CRLF+CRLF).length());
        return cadena;
    }
    /**
     * Obtiene el parámetro longitud de los comandos RTSP
     * @param mensaje.- comando RTSP
     * @return longitud del mensaje
     */
    public static int getLongitudContenidoMensaje(String mensaje){
        int longitud=-1;

```

```

        if(mensaje.contains("Content-Length: ") ||
mensaje.contains("Content-length: ")) {
            int inicio=mensaje.indexOf("Content-Length: ")+"Content-
Length: ".length();
            int fin=mensaje.indexOf(CRLF,inicio);
            longitud=Integer.parseInt(mensaje.substring(inicio,
fin));
        }
        return longitud;
    }
}
/**
 * Cambia las urls para el filtrado de los comandos RTSP
 * @param mensaje.- comando rtsp
 * @param url.- url a ser cambiado
 * @param urlSustituto.- url por el cuál va a ser cambiado
 * @return
 */
public static String cambiarUrlRTSP(String mensaje,UrlRTSP url,
UrlRTSP urlSustituto) {
    String resultado="";
    String url1="rtsp://" +url.getHost();
    String url2="rtsp://" +urlSustituto.getHost();

    if(url.getTienePuerto()){
        url1+=":"+url.getPuerto();
    }

    if(urlSustituto.getTienePuerto()){
        url2+=":"+urlSustituto.getPuerto();
    }
    resultado=mensaje.replaceAll(url1, url2);

    return resultado;
}
}
/**
 * Cambia la cadena host que se encuentren el el mensaje
 * @param mensaje.- comando RTSP
 * @param url
 * @param urlSustituto
 * @return comando RTSP filtrado
 */
public static String cambiarHost(String mensaje,UrlRTSP url, UrlRTSP
urlSustituto){
    String resultado="";
    resultado= mensaje.replaceAll(url.getHost(),
urlSustituto.getHost());
}

```

```

        resultado= mensaje.replaceAll(url.getIp(),
urlSustituto.getIp());
        return resultado;
    }
    /**
     * Obtiene el comando RTSP
     * @param mensaje
     * @return retorna el Comando RTSP
     */
    private static String getComando(String mensaje){
        StringTokenizer tokens = new StringTokenizer(mensaje);
        String comando_string = tokens.nextToken();
        return comando_string;
    }

    public static boolean isDescribe(String mensaje) {
        if(getComando(mensaje).equalsIgnoreCase("describe"))
        {
            return true;
        }
        return false;
    }

    public static boolean isOptions(String mensaje) {
        if(getComando(mensaje).equalsIgnoreCase("options"))
        {
            return true;
        }
        return false;
    }

    public static boolean isPause(String mensaje) {
        if(getComando(mensaje).equalsIgnoreCase("pause"))
        {
            return true;
        }
        return false;
    }

    public static boolean isPlay(String mensaje) {
        if(getComando(mensaje).equalsIgnoreCase("play"))
        {
            return true;
        }
        return false;
    }
}

```

```

public static boolean isSetup(String mensaje){
    if(getComando(mensaje).equalsIgnoreCase("setup"))
    {
        return true;
    }
    return false;
}

public static boolean isSetupTrack0(String mensaje) {
    if(getComando(mensaje).equalsIgnoreCase("setup") &&
mensaje.contains("trackID=0"))
    {
        return true;
    }
    return false;
}

public static boolean isSetupTrack1(String mensaje) {
    if(getComando(mensaje).equalsIgnoreCase("setup") &&
mensaje.contains("trackID=1"))
    {
        return true;
    }
    return false;
}

public static boolean isTearDown(String mensaje) {
    if(getComando(mensaje).equalsIgnoreCase("teardown"))
    {
        return true;
    }
    return false;
}

public static boolean isSetupRespuesta(String mensaje){
    if(mensaje.contains("server_port=")&&mensaje.contains("client_port=")
){
        return true;
    }
    return false;
}
}

```

3.1.2.4.2 UriRTSP

```
package ec.edu.espe.util;
```

```

import java.net.InetAddress;
import java.security.InvalidParameterException;
/**
 *
 * @author ESPE
 * La clase utilitaria UrlRTSP, ayuda en el tratamiento de las direcciones
rtsp://
 *
 */

public class UrlRTSP {
    //Estructura dirección rtsp://host:puerto/recurso
    private String host;
    private int puerto;
    private String ip;
    private boolean tienePuerto;
    private String recurso;

    public UrlRTSP(String URL) throws InvalidParameterException {
        if(URL.startsWith("rtsp://")){
            String aux=URL.substring(7);
            int fin = aux.indexOf("/");
            String dominio=aux.substring(0, fin);
            String[] cadenas=dominio.split(":");
            if(cadenas.length>1){
                tienePuerto=true;
                host=cadenas[0];
                puerto=Integer.parseInt(cadenas[1]);
            }else{
                tienePuerto=false;
                host=dominio;
                puerto=554;
            }
            try{
                ip=InetAddress.getByName(host).getHostAddress();
            }catch (Exception e) {
                throw new InvalidParameterException("URL no es un
rtsp valido");
            }

            recurso=aux.substring(fin+1);
        }else{
            throw new InvalidParameterException("URL no es un
rtsp valido");
        }
    }
}

```

```

public String getRecurso() {
    return recurso;
}

public void setRecurso(String recurso) {
    this.recurso = recurso;
}

public String getURL() {
    String aux="";
    if(getPuerto()==554){
        aux="rtsp://" +getHost()+"/"+getRecurso();
    }else{
        aux="rtsp://" +getHost()+":"+getPuerto()+"/"+getRecurso();
    }
    return aux;
}

public String getHost() {
    return host;
}

public void setHost(String host) {
    this.host = host;
}

public int getPuerto() {
    return puerto;
}

public void setPuerto(int puerto) {
    this.puerto = puerto;
}

public String getIp() {
    return ip;
}

public void setIp(String ip) {
    this.ip = ip;
}

public boolean getTienePuerto(){
    return tienePuerto;
}

```


}

BIBLIOGRAFÍA

Alonso, F. J. (2009-2010). *Tecnologías de Streaming*. Oviedo.

Android Developers. (2010 de 08 de 23). *Android Developers*. Obtenido de developer.android.com:
<http://developer.android.com/guide/basics/what-is-android.html>

Anguiano Rey, E. (s.f.). *memnon.ii.uam.es*. Obtenido de User Datagram Protocol:
<http://memnon.ii.uam.es/~eloy/media/REDES/Tema7-udp.pdf>

Barceló, J., Giera, J., Llorente, S., Marqués, J., Martí, R., Peig, E., y otros. (2008). *Protocolos y aplicaciones internet*. Barcelona: UOC.

Bellavista, P., Corradi, A., & Giannelli, C. (June 2005). "Mobile Proxies for Proactive Buffering in Wireless Internet Multimedia Streaming". Proceedings of the IEEE International Conference on Distributed Computing Systems Workshop (ICDCSW'05).

Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (septiembre 2003). "JADE, A White Paper". Journal of Telecom Italia Lab,: Vol. 3, No. 3, pp. 6-19.

Bragado, J. F. (Abril 2004 de 2004).

http://zenon.etsii.urjc.es:8080/foros/AgentesInteligentes_2003_2004/1082414149/1082495069/Jade.pdf. Recuperado el 2010, de Agentes Inteligentes: JADE.

Carbó, J. (6 de Julio de 2007). *Grupo de Inteligencia Artificial Aplicada, Curso "Tecnología de Agentes en Inteligencia Ambiental", Campusde Colmenarejo, Univ. Carlos III de Madrid*. Obtenido de galahad.plg.inf.uc3m.es: <http://galahad.plg.inf.uc3m.es/~swagents/material/JADELEAPJADEX.pdf>

Catalán, E. (2009). *Implementación de un servidor de streaming de vídeo adaptativo*. Barcelona: Universidad Politecnica de Catalunya.

Garamendi Bragado, J. F. (Abril de 2004). *Agentes Inteligentes: JADE*. Obtenido de zenon.etsii.urjc.es:8080:

http://zenon.etsii.urjc.es:8080/foros/AgentesInteligentes_2003_2004/1082414149/1082495069/Jade.pdf

García Sandoval, O. Y. (30 de Junio de 2008). *seguimientoaprendices*. Obtenido de Proyecto Seguimiento Aprendices: <http://seguimientoaprendices.wordpress.com/>

Gotta, D., Trucco, T., Ughetti, M., Semeria, S., Cucè, C., & Porcino, A. (14 de Junio de 2010). *Jade Android*. Obtenido de Jade: http://jade.tilab.com/doc/tutorials/JADE_ANDROID_Guide.pdf

Komatineni, S. Y. (2009). *Pro Android*. New York: Apress.

Larios Osorio, V. (1997-1999). *Universidad Autónoma de Querétaro (México)*. Obtenido de www.uaq.mx: <http://www.uaq.mx/matematicas/estadisticas/xu5.html#t1>

Lixin, G., Zhi-Li, Z., & Towsley, D. (December 2003). *Proxy-Assisted Techniques for Delivering Continuous Multimedia Streams* (Vol. Vol. 11). *EEE/ACM Transactions on Networking I*.

Ochoa-Domínguez, H. J., & García, J. M. (2007). *scielo.org.mx*. Obtenido de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-77432007000300004&lng=pt&nrm=iso

(2010). *Redes de computadores*. <http://www.ace.ual.es/~leo/redes/Rel3-01-02.pdf>.

Universidad Carlos III de Madrid. (s.f.). *Software de Comunicaciones*. Obtenido de sites.google.com:
<http://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

UniversidadCarlosIIIdeMadrid. (s.f.). *Software de Comunicaciones*. Obtenido de
<http://sites.google.com/site/swcuc3m/home/android/generalidades/aplicacionespag3>

Vandalore, B., Feng, W.-C., Jain, R., & Fahm, S. (1999). "A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia" (Vols. Vol. 7, No. 3). Real-Time Imaging.

Wikipedia. (08 de Octubre de 2010). *Advanced Audio Coding*. Obtenido de Wikipedia:
http://es.wikipedia.org/wiki/Advanced_Audio_Coding

Wikipedia. (06 de 09 de 2010). *Android*. Obtenido de wikipedia.org:
<http://es.wikipedia.org/wiki/Android>

Wikipedia. (25 de Abril de 2010). *Comprensión multi-tasa adaptativa*. Obtenido de Wikipedia:
http://es.wikipedia.org/wiki/Compresi%C3%B3n_multi-tasa_adaptativa

Wikipedia. (28 de 08 de 2010). *es.wikipedia.org*. Obtenido de Eclipse(software):
[http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))

Wikipedia. (25 de 09 de 2010). *JDK*. Obtenido de es.wikipedia.org:
http://es.wikipedia.org/wiki/Java_Development_Kit

Wikipedia. (15 de Julio de 2010). *Jitter*. Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/Jitter>

Wikipedia. (27 de 08 de 2010). *RTSP*. Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/RTSP>

Wikipedia. (14 de 09 de 2010). *VLC Media Player*. Obtenido de Wikipedia:
http://es.wikipedia.org/wiki/VLC_media_player

Wikipedia. (20 de septiembre de 2010). *Wikipaida*. Obtenido de Programación Extrema:
http://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema

Wikipedia. (13 de 10 de 2010). *Wireshark*. Obtenido de Wikipedia:
<http://es.wikipedia.org/wiki/Wireshark>

WikiSpaces. (2009). *Programación Jade*. Obtenido de
<http://programacionjade.wikispaces.com/Comportamientos>

Zavala Yerovi, A. Y. (14 de 01 de 2009). *dspace.esepoch.edu.ec*. Obtenido de <http://dspace.esepoch.edu.ec/bitstream/123456789/328/1/18T00409.pdf>

TRABAJOS CITADOS

Alonso, F. J. (2009-2010). *Tecnologías de Streaming*. Oviedo.

Android Developers. (2010 de 08 de 23). *Android Developers*. Obtenido de [developer.android.com](http://developer.android.com/guide/basics/what-is-android.html): <http://developer.android.com/guide/basics/what-is-android.html>

Anguiano Rey, E. (s.f.). *memnon.ii.uam.es*. Obtenido de User Datagram Protocol: <http://memnon.ii.uam.es/~eloy/media/REDES/Tema7-udp.pdf>

Barceló, J., Griera, J., Llorente, S., Marqués, J., Martí, R., Peig, E., y otros. (2008). *Protocolos y aplicaciones internet*. Barcelona: UOC.

Bellavista, P., Corradi, A., & Giannelli, C. (June 2005). "Mobile Proxies for Proactive Buffering in Wireless Internet Multimedia Streaming". Proceedings of the IEEE International Conference on Distributed Computing Systems Workshop (ICDCSW'05).

Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (septiembre 2003). "JADE, A White Paper". Journal of Telecom Italia Lab,: Vol. 3, No. 3, pp. 6-19.

Bragado, J. F. (Abril 2004 de 2004).

http://zenon.etsii.urjc.es:8080/foros/AgentesInteligentes_2003_2004/1082414149/1082495069/Jade.pdf. Recuperado el 2010, de Agentes Inteligentes: JADE.

Carbó, J. (6 de Julio de 2007). *Grupo de Inteligencia Artificial Aplicada, Curso "Tecnología de Agentes en Inteligencia Ambiental", Campus de Colmenarejo, Univ. Carlos III de Madrid*. Obtenido de galahad.plg.inf.uc3m.es: <http://galahad.plg.inf.uc3m.es/~swagents/material/JADELEAPJADEX.pdf>

Catalán, E. (2009). *Implementación de un servidor de streaming de vídeo adaptativo*. Barcelona: Universidad Politecnica de Catalunya.

Garamendi Bragado, J. F. (Abril de 2004). *Agentes Inteligentes: JADE*. Obtenido de zenon.etsii.urjc.es:8080:

http://zenon.etsii.urjc.es:8080/foros/AgentesInteligentes_2003_2004/1082414149/1082495069/Jade.pdf

García Sandoval, O. Y. (30 de Junio de 2008). *seguimientoaprendices*. Obtenido de Proyecto Seguimiento Aprendices: <http://seguimientoaprendices.wordpress.com/>

Gotta, D., Trucco, T., Ughetti, M., Semeria, S., Cucè, C., & Porcino, A. (14 de Junio de 2010). *Jade Android*. Obtenido de Jade: http://jade.tilab.com/doc/tutorials/JADE_ANDROID_Guide.pdf

Komatineni, S. Y. (2009). *Pro Android*. New York: Apress.

Larios Osorio, V. (1997-1999). *Universidad Autónoma de Querétaro (México)*. Obtenido de www.uaq.mx: <http://www.uaq.mx/matematicas/estadisticas/xu5.html#t1>

Lixin, G., Zhi-Li, Z., & Towsley, D. (December 2003). *Proxy-Assisted Techniques for Delivering Continuous Multimedia Streams* (Vol. Vol. 11). *EEE/ACM Transactions on Networking I*.

Ochoa-Domínguez, H. J., & García, J. M. (2007). *scielo.org.mx*. Obtenido de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-77432007000300004&lng=pt&nrm=iso

(2010). *Redes de computadores*. <http://www.ace.ual.es/~leo/redes/Rel3-01-02.pdf>.

Universidad Carlos III de Madrid. (s.f.). *Software de Comunicaciones*. Obtenido de sites.google.com: <http://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

UniversidadCarlosIIIdeMadrid. (s.f.). *Software de Comunicaciones*. Obtenido de <http://sites.google.com/site/swcuc3m/home/android/generalidades/aplicacionespag3>

Vandalore, B., Feng, W.-C., Jain, R., & Fahm, S. (1999). "A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia" (Vols. Vol. 7, No. 3). Real-Time Imaging.

Wikipedia. (08 de Octubre de 2010). *Advanced Audio Coding*. Obtenido de Wikipedia: http://es.wikipedia.org/wiki/Advanced_Audio_Coding

Wikipedia. (06 de 09 de 2010). *Android*. Obtenido de wikipedia.org: <http://es.wikipedia.org/wiki/Android>

Wikipedia. (25 de Abril de 2010). *Comprensión multi-tasa adaptativa*. Obtenido de Wikipedia: http://es.wikipedia.org/wiki/Compresi%C3%B3n_multi-tasa_adaptativa

Wikipedia. (28 de 08 de 2010). *es.wikipedia.org*. Obtenido de Eclipse(software): [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))

Wikipedia. (25 de 09 de 2010). *JDK*. Obtenido de es.wikipedia.org: http://es.wikipedia.org/wiki/Java_Development_Kit

Wikipedia. (15 de Julio de 2010). *Jitter*. Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/Jitter>

Wikipedia. (27 de 08 de 2010). *RTSP*. Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/RTSP>

Wikipedia. (14 de 09 de 2010). *VLC Media Player*. Obtenido de Wikipedia: http://es.wikipedia.org/wiki/VLC_media_player

Wikipedia. (20 de septiembre de 2010). *Wikipaida*. Obtenido de Programación Extrema: http://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema

Wikipedia. (13 de 10 de 2010). *Wireshark*. Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/Wireshark>

WikiSpaces. (2009). *Programación Jade*. Obtenido de <http://programacionjade.wikispaces.com/Comportamientos>

Zavala Yerovi, A. Y. (14 de 01 de 2009). *dspace.esPOCH.edu.ec*. Obtenido de <http://dspace.esPOCH.edu.ec/bitstream/123456789/328/1/18T00409.pdf>

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

Srta. Cristina Navarro

Sr. Darío Larenas

COORDINADOR DE LA CARRERA

Ing. Danilo Martínez

Sangolquí, octubre de 2010