



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Herramienta para la medición de la calidad de experiencia en servicios streaming de audio y video sobre redes cableadas de datos entre extremos.

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Autores

Vladimir Alvarez Pérez

Dainier Aguila Medina

Tutor

MSc. Salvador González Gómez

Co-tutor

Dr. Juan Francisco López Fernández

La Habana, junio de 2013

“Año 55 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaración de autoría

Declaramos ser los autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de junio del año 2013.

Vladimir Alvarez Pérez
Firma del Autor

Dainier Aguila Medina
Firma del Autor

MSc. Salvador González Gómez
Firma del Tutor

Dr. Juan Francisco López Fernández
Firma del Co-tutor

Datos de contacto

MSc. Salvador González Gómez, Ingeniero en Ciencias Informáticas, profesor asistente, máster en Telemática y Redes de Telecomunicación, labora en el Centro de Telemática de la Universidad de las Ciencias Informáticas, teléfono 837 3137. Actualmente se desempeña como profesor de las asignaturas de Investigación de Operaciones y Probabilidad y Estadística, trabaja en el doctorado en el grupo PEFCI, en la parte productiva se desempeña como Asesor de Tecnología.

Correo electrónico: sgonzalezg@uci.cu

Dr. Juan Francisco López Fernández, Doctor en Ciencias, profesor titular, trabaja en el centro CICE (Centro de Innovación y Calidad de la Educación) de la Universidad de las Ciencias Informáticas, teléfono 835 8126.

Correo electrónico: jflopezf@uci.cu

Resumen

La investigación aborda acerca de la medición de la calidad de experiencia en servicios streaming de audio y video sobre redes cableadas de datos entre los extremos. Importante elemento a tener en cuenta por los proveedores de streaming, pues informará en gran medida el éxito o fracaso de los servicios.

Los problemas entre los extremos que afectan la calidad, provoca la insatisfacción por partes de los usuarios e implica pérdidas económicas. Las herramientas estudiadas se encuentran orientadas a otros tipos de servicios o no tienen en cuenta cómo afecta los indicadores de calidad de servicio a la calidad experimentada por los usuarios. De ahí la necesidad de desarrollar una que cumpla con dichas características. Para lograr la implementación fue necesario realizar un estudio que permitiera identificar los indicadores a usar y un modelo para la medición. La herramienta aplica un modelo elaborado teniendo en cuenta el modelo E, usado para la estimación de la calidad percibida en redes operativas, pero ajustándolo a los indicadores definidos. Además es capaz de consumir los servicios (reproducción de audio y video), capturar los paquetes que viajan por la red correspondientes a los servicios streaming de audio y video, graficar los indicadores de calidad según su comportamiento y variación en el tiempo y proporcionar una medida cualitativa de la calidad de experiencia a las pruebas realizadas por la herramienta.

Palabras clave: calidad, medición, modelo, servicio, streaming.



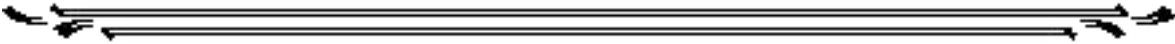
Índice de contenidos

Declaración de autoría	I
Datos de contacto.....	II
Resumen.....	III
Índice de contenidos	IV
Glosario de términos.....	VII
Introducción	1
Capítulo 1: Fundamentación Teórica.....	5
Introducción.....	5
1.1 Servicios streaming. Características principales.....	5
1.1.1 Principales etapas conceptuales dentro del proceso de streaming.....	7
1.1.2 Arquitectura del servicio para realizar la transmisión.....	7
1.1.3 Formas de streaming	10
1.1.4 Protocolos de comunicación	11
1.1.4.1 Protocolos de transporte	11
1.1.4.3 Protocolo de Transporte en Tiempo Real y Protocolo de Control en Tiempo Real	12
1.2 Calidad en servicios streaming de audio y video	14
1.2.1 QoS y QoE.....	14
1.2.2 Indicadores para medir la QoE desde la QoS en servicios streaming.....	15
1.2.3 Cuantificaciones para las métricas definidas de QoS	17
1.2.4 Modelos para medir la QoE en servicios telemáticos a partir de QoS.....	18
1.2.4.1 El modelo MOS.....	18
1.2.4.2 Otros modelos.....	18

ÍNDICE DE CONTENIDOS

1.2.4.3	Modelo a usar en la herramienta.....	19
1.2.5	Herramientas de software relacionadas	19
1.2.6	Tecnologías utilizadas	23
1.2.6.1	Lenguaje de Programación	23
1.2.6.2	Framework de desarrollo.....	24
1.2.6.3	Entorno de Desarrollo Integrado	24
1.2.6.4	Sistema Gestor de Base de Datos.....	25
1.2.6.5	Metodología de desarrollo	25
	Conclusiones del capítulo	26
	Capítulo 2: Características del Sistema.....	27
	Introducción.....	27
2.1	Indicadores de QoS	27
2.2	Modelo a emplear	29
2.3	Explicación del modelo para la medición de la QoE	29
2.4	Modelo MOS	31
2.5	Historias de usuarios	31
2.6	Requisitos	34
2.6.1	Requisitos funcionales	35
2.6.2	Requisitos no funcionales	35
2.7	Tarjetas CRC	36
2.8	Plan de iteraciones	41
2.9	Arquitectura	42
2.9.1	Patrones de diseño.....	43
2.9.2	Diagrama de despliegue	44
2.9.3	Diagrama de componentes.....	45

ÍNDICE DE CONTENIDOS



2.9.4	Diseño de base de datos.....	46
2.9.4.1	Modelo de datos	46
2.9.5	Estructura de la herramienta.....	47
	Conclusiones del capítulo	49
Capítulo 3:	Validación de la Solución.....	50
	Introducción.....	50
3.1	Pruebas unitarias	50
3.2	Pruebas de aceptación	53
	Conclusiones del capítulo	57
	Conclusiones generales	58
	Recomendaciones	60
	Referencias bibliográficas	61
	Bibliografía	65

Glosario de términos

Bit: es un dígito del sistema de numeración binaria.

Buffer: es una porción de memoria dedicada a almacenar datos, para posteriormente enviarlos hacia un dispositivo externo o recibirlos desde un dispositivo externo.

Byte: formado por ocho bits. Se trata de la unidad básica con la que funcionan los ordenadores. 1024 bytes son un Kbyte (KB), y 1024 Kbyte hacen un Mbyte (MB).

Codec: es la abreviatura de codificador–decodificador.

CPU: unidad central de procesamiento o también llamado microprocesador (del inglés, Central Processing Unit).

Datagrama: forma de encaminar los paquetes en la red. Cada paquete se trata de forma independiente, conteniendo cada uno la dirección de destino.

Encoder: dispositivo que convierte una señal RGB (rojo, verde y azul por sus siglas en inglés) en una señal NTSC. Es utilizado para grabar clips de video digital a una videocasete.

Enrutador: es un dispositivo hardware para la interconexión de redes. Encamina los paquetes de información electrónica tomando decisiones de tráfico en base a las condiciones de la red.

Ethernet: es un estándar de redes de área local para computadores.

Frame: es un fotograma o cuadro, una imagen particular dentro de una sucesión de imágenes que componen una animación.

Framework: es un marco de trabajo que define 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.

FTP: protocolo de transferencia de archivos (del inglés, File Transfer Protocol).

Host: ordenador conectado a una red, que provee y utiliza servicios de ella.

HTTP: protocolo de transferencia de hipertexto (del inglés, Hypertext Transfer Protocol).

GLOSARIO DE TÉRMINOS

ICMP: protocolo de mensajes de control de Internet (del inglés, Internet Control Message Protocol).

IMAP: protocolo de acceso a mensajes en Internet (del inglés, Internet Message Access Protocol).

IP: protocolo de Internet (del inglés, Internet Protocol).

IPMI: interfaz de administración de plataforma inteligente (del inglés, Intelligent Platform Management Interface).

LAN: red de área local (del inglés, Local Area Network).

MOS: puntuación media de opinión (del inglés, Mean Opinion Score).

Multiplexación: es la combinación de dos o más canales de información en un solo medio de transmisión usando un dispositivo llamado multiplexor.

NNTP: protocolo de transferencia de noticias en red (del inglés, Network News Transport Protocol).

PDN: paquete de datos para el nivel de red (del inglés, Packet Data Network).

POP3: protocolo de oficina de correo (del inglés, Post Office Protocol).

QoE: calidad de experiencia (del inglés, Quality of Experience).

QoS: calidad de servicio (del inglés, Quality of Service).

RTCP: protocolo de control en tiempo real (del inglés, Real Time Control Protocol).

RTP: protocolo de transporte en tiempo real (del inglés, Real Time Transport Protocol).

RTSP: protocolo de streaming en tiempo real (del inglés, Real Time Streaming Protocol).

SMS: servicio de mensaje corto (del inglés, Short Message Service).

SMTP: protocolo simple para la transferencia de mensajes (del inglés, Simple Mail Transfer Protocol).

SNMP: protocolo simple de administración de red (del inglés, Simple Network Management Protocol).

SSH: intérprete de órdenes segura (del inglés, Secure SHell).

GLOSARIO DE TÉRMINOS



SSL: capa de conexión segura (del inglés, Secure Sockets Layer).

TCP: protocolo de control de transmisión (del inglés, Transmission Control Protocol).

UDP: protocolo de datagrama de usuario (del inglés, User Datagram Protocol).

XML: lenguaje de marcas extensible (del inglés, eXtensible Markup Language).

Introducción

Los dispositivos que se conectan a las redes cableadas son usados a gran escala en la actualidad. Permiten la comunicación entre las personas sin importar las fronteras. Diferentes acontecimientos han marcado hitos en la historia: el surgimiento de la radio, el teléfono, la televisión, y más importante aún, Internet; una red lógica global de ordenadores interconectados entre sí.

Los diferentes aparatos capaces de conectarse a Internet, cada vez soportan más servicios, como son: envío de mensajes de texto, de voz; compartición de videos, fotos; entre muchos otros. Ello es posible gracias al continuo avance de las técnicas de las telecomunicaciones, que se encuentran ampliamente desarrolladas; pero aún no cumplen con todas las expectativas que se van generando. La sociedad evoluciona, se transforma, crece a un ritmo imparable. Las necesidades de ampliación junto a la optimización de los servicios prestados por los sistemas de redes son sorprendentes, poniendo cada vez más alto el reto para los ingenieros.

Uno de los servicios difundidos sobre las redes y sus continuas evoluciones, es el streaming de audio y video. Una tecnología que se utiliza para aligerar la descarga y reproducción de contenidos en la web, debido a que permite escuchar y visualizar los archivos mientras se están descargando, (1). El montaje del servicio sobre dichas redes impone dos conceptos de importancia. Uno de ellos es la calidad de servicio (QoS, por sus siglas en inglés) que se presta a los consumidores. Las investigaciones y pruebas realizadas sobre el tema, generalmente están centradas en aspectos objetivos, como las métricas para indicadores de rendimiento de la red (pérdida de paquetes, latencia, ancho de banda, entre otros). Vale destacar para mayor comprensión, que se entiende como QoS la capacidad de la red para proporcionar una prestación a un nivel garantizado. Además la QoS abarca todas las funciones, mecanismos y procedimientos en la red y el terminal, para garantizar la obtención de la calidad negociada entre el equipo del usuario y el servidor. (2)

El otro concepto es la calidad de experiencia (QoE, por siglas en inglés), que se enfoca más bien a cómo repercute en los usuarios la QoS y cómo se sienten con los servicios prestados. Dicho concepto surge de la necesidad de determinar el grado de satisfacción

de los usuarios con el servicio. Por otro lado, la QoE es una extensión de la QoS en el sentido de que la primera proporciona información acerca de la entrega de servicios desde el punto de vista del usuario final, (4). La QoE está orientada a la calidad percibida, cuando se consume uno o varios de los servicios prestados en relación con la accesibilidad, rendimiento, demora, pérdida de paquetes, capacidad de retención e integridad del servicio, seguridad de acceso, tiempo de respuesta del servicio, tiempo en que tarda en configurarse un servicio determinado, entre otros.

Para los proveedores de los servicios streaming la QoE es un elemento importante a tener en cuenta. Pues informará en gran medida el éxito o fracaso del servicio. Si se mira en términos comerciales, empresariales y/o económicos, tener los usuarios satisfechos, implica garantía de continuidad de los contratos firmados entre las partes (cliente-proveedor), portando estabilidad financiera y credibilidad en el mercado. Los problemas entre los extremos que afectan la calidad, provoca la insatisfacción por partes de los usuarios e implica pérdidas económicas.

Tomando en consideración los elementos antes expuestos y en función de dar respuesta a la necesidad mencionada, para que pueda ser aplicada en un futuro a las nuevas tecnologías y vayan evolucionando en el camino hacia la perfección de las comunicaciones, se plantea el siguiente **problema a resolver**: ¿cómo medir la calidad de experiencia en servicios streaming de audio y video sobre redes cableadas de datos entre extremos?

Sustentando la investigación bajo el **objeto de estudio**: la QoE en servicios telemáticos sobre redes cableadas de datos entre extremos, enmarcada en el **campo de acción**: la QoE en servicios streaming de audio y video sobre redes cableadas de datos entre extremos. Se concibe como **objetivo general**: desarrollar una herramienta para la medición de calidad de experiencia en servicios streaming de audio y video sobre redes cableadas de datos entre extremos.

Para dar cumplimiento al objetivo se trazan las siguientes **preguntas científicas**:

1. ¿Cuáles son los fundamentos teóricos que sustentan el desarrollo de una herramienta para medir la QoE en servicios streaming de audio y video sobre redes cableadas de datos entre extremos?



2. ¿Cuál es el estado actual de las herramientas que permiten medir la QoE en servicios streaming de audio y video sobre redes cableadas de datos entre extremos?
3. ¿Qué características esenciales debe poseer la herramienta que permita medir la QoE en servicios streaming de audio y video sobre redes cableadas de datos entre extremos?
4. ¿Cómo validar la efectividad de la herramienta que se propone para medir la QoE en servicios streaming de audio y video sobre redes cableadas de datos entre extremos?

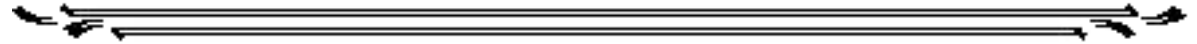
Para dar respuesta a las preguntas científicas se plantean las siguientes **tareas de la investigación:**

1. Sistematización del estudio de los fundamentos teóricos y metodológicos que sustentan el desarrollo de una herramienta para la medición de la QoE en servicios streaming de audio y video sobre redes cableadas de datos entre extremos.
2. Determinación del estado actual de las herramientas que permiten medir la QoE en servicios streaming de audio y video sobre redes cableadas de datos entre extremos.
3. Definición de las características esenciales que debe poseer la herramienta para que permita medir la QoE en servicios streaming de audio y video sobre redes cableadas de datos entre extremos.
4. Determinación de la forma de validación de la herramienta para medir la QoE en servicios streaming de audio y video sobre redes cableadas de datos entre extremos.

Con el desarrollo de la investigación se espera obtener una herramienta cuyo impacto social evidencie la medición de la QoE en el servicio streaming de audio y video sobre redes cableadas de datos entre extremos. Lo cual permitirá obtener una visión extremo a extremo del comportamiento del servicio, evitar la deserción de clientes y la detección de posibles problemas en la red.

El documento está estructurado de la siguiente forma:

CAPÍTULO 1. Fundamentación Teórica.



El capítulo está concebido para describir todos los elementos teóricos y metodológicos necesarios para el desarrollo de la investigación. Se encuentra dividido por temas y cada tema en subtemas. Han sido organizados de forma coherente para lograr una mejor comprensión por parte del lector. Comienza describiendo los servicios streaming, sus características, principales etapas conceptuales, formas de distribución y los protocolos de comunicación. Luego de comprendido los conceptos básicos mencionados se pasa al entendimiento de otros elementos conceptuales de mayor complejidad e importancia, para el estudio como son la QoS, la QoE y los indicadores que informarán el estado con que se presta el servicio, los procedimientos y modelos para la monitorización y evaluación continua. Finalmente se refleja el estado del arte relacionado al tema investigativo, las herramientas a utilizar, así como las metodologías de desarrollo.

CAPÍTULO 2. Características del sistema.

Describe los elementos implicados en el proceso de desarrollo, los indicadores a monitorizar, el modelo para la medición de la QoE que se propone, así como los requisitos que se deben tener en cuenta. También se define un plan de iteraciones que contempla el tiempo de desarrollo de cada etapa hasta lograr la completa implementación del software. La arquitectura de la herramienta es descrita mencionando los patrones de diseño utilizados, los diagramas de componentes y despliegue así como el modelo de datos y la estructura de la herramienta.

CAPÍTULO 3. Validación de la solución.

El capítulo describe las pruebas realizadas a la aplicación una vez implementada. Se realizan pruebas unitarias y de aceptación para validar el correcto funcionamiento y el cumplimiento de los requisitos definidos.

También contiene las **conclusiones** donde se realiza un breve resumen acerca del cumplimiento de los objetivos de la investigación; y se proponen **recomendaciones** para continuar desarrollando el objeto de la investigación.

Se incluye un **glosario de términos** que facilita el entendimiento del documento, en el que se explican aquellos términos que podrían resultar nuevos o de diferente significado en el contexto de la investigación. Se exponen todos los materiales consultados y referenciados, quedando organizados en **bibliografía consultada** y **referencias bibliográficas**, según corresponda en cada caso.

Capítulo 1: Fundamentación Teórica

Introducción

En el presente capítulo se describen detalladamente los conceptos que sustentan la herramienta a desarrollar, como fundamentación teórica base para obtener una amplia comprensión del problema a resolver. Se comienza abordando todo lo relacionado con los servicios streaming: características, protocolos de transmisión, arquitectura, entre otros. Luego se profundiza en los conceptos relacionados con la calidad del servicio y la calidad de experiencia, así como en los modelos e indicadores seleccionados para medir la QoE a partir de la QoS. Seguido se presentan modelos y procedimientos relacionados con el tema de la investigación, finalizando con las tecnologías y metodología a usar en el desarrollo de la herramienta.

1.1 Servicios streaming. Características principales

Cuando se habla de streaming, se refiere a una tecnología que posibilita la transmisión de audio y video por Internet de forma directa o diferida. Se utiliza para aligerar la descarga y ejecución de audio y video en la web, debido a que permite escuchar y visualizar los archivos mientras se descargan, (1). Si no se utiliza streaming para mostrar un contenido multimedia en la red, se tendría que descargar primero el archivo completo en un dispositivo para luego reproducirlo; por lo tanto las emisiones en directo no se podrían realizar. Al consumir los servicios streaming, la información se reproduce a medida que se recibe, o sea, el cliente no tiene que esperar a que toda la información esté disponible. En el momento que existan los datos necesarios para comenzar la reproducción, se iniciará. De esta manera, un cliente de streaming puede no descargar el audio/video, simplemente reproduce la información de cada uno de los paquetes y los descarta.

La información a reproducir se trasmite desde un servidor, el cual es el responsable de establecer un canal para el control de los flujos y otro para el envío de los paquetes. Los clientes tienen la misión de reproducir los datos y proporcionar la interfaz necesaria para la interacción con el servicio. Primero el dispositivo móvil se conecta con el servidor comenzando a recibir la información que se transmite. El cliente, mientras la recibe, construye un buffer donde la guarda. Cuando se ha almacenado una pequeña parte, inicia la reproducción y continúa con la descarga simultáneamente; de modo que cuando

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

termine la descarga, también haya acabado la reproducción. Si en algún momento la conexión sufre descensos de velocidad, se utiliza la información almacenada para evitar interrupciones. Si la comunicación se corta por demasiado tiempo, el buffer se vacía y la reproducción se cortaría hasta que se restaurase la señal. (5)

Mientras el cliente reproduce la secuencia de video (del inglés, stream), las peticiones realizadas son controladas usando el Protocolo de Streaming en Tiempo Real (RTSP), explicado en próximos epígrafes, que permite realizar acciones como cambiar el elemento en reproducción por otro siguiente o correr el contenido a un punto determinado. En cuanto a los datos sincronizados contenidos en el stream, viajan a través del Protocolo de Transporte en Tiempo Real (RTP), que permite el traspaso de información sobre cualquier tipo de red, de igual manera explicado en próximos epígrafes.

Se pueden reproducir en una amplia variedad de aplicaciones, ya sean reproductores multimedia o páginas Web que incluyan módulos o complementos de estos reproductores. Los clientes no están sujetos a un sistema operativo para poder reproducir los datos, pues la mayoría de los sistemas operativos de uso común tienen aplicaciones para poder ejecutar contenido multimedia como streaming. (6)

Los servicios streaming cubren las necesidades de varios sectores: educacionales, empresariales, institucionales entre otros. Pueden ser usados con diferentes fines, entre ellos:

- Radio online.
- Retransmisiones deportivas.
- Conferencias.
- Congresos.
- Cursos online.
- Entrevistas.
- Seminarios.
- Presentaciones de productos.
- Consultas médicas a distancia.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Consultas jurídicas.

Para lograr la puesta en marcha de los servicios streaming se hace necesario transitar por varios momentos desde la grabación del elemento audio/video a transmitir hasta su puesta en línea. Estos momentos se conocen como etapas conceptuales dentro del proceso de streaming.

1.1.1 Principales etapas conceptuales dentro del proceso de streaming

Durante el proceso de streaming se pasa por seis etapas fundamentales. Primeramente es necesario realizar la **captura**¹. En este momento el stream es obtenido, aún sin codificación, ni compresión. Puede obtenerse a través de dispositivos como cámaras de video o grabadores de audio. También pudiera haber sido almacenado previamente en un disco u otro medio para guardar información digital. El próximo paso es **codificar**, para transmitir el stream por una red cualquiera debe de estar codificado y comprimido en un formato correcto. El **servidor** es el encargado de esperar las peticiones del cliente. Cuando estas llegan, las procesa buscando rápidamente el elemento de video o audio solicitado en su localización real y comienza a prestar el servicio de forma sincronizada. La fase siguiente es el recibo de los datos por el **cliente**. Los datos son tomados desde la red y se envían como un flujo a la etapa de **decodificación**. Aquí los datos son decodificados y descomprimidos de acuerdo al formato contenedor y a los codec usados en el proceso de codificación. Finalmente se pasa a la **reproducción**. Es el momento en que el contenido regenerado a partir de los datos es mostrado en pantalla.

El flujo de información entre los clientes y el servidor se transmite usando una de las siguientes arquitecturas en dependencia del tipo o forma de servicio a prestar. Puede ser Multidifusión, Unidifusión o Difusión. Dichas arquitecturas son explicadas en el próximo epígrafe.

1.1.2 Arquitectura del servicio para realizar la transmisión

- **Multidifusión:**

¹ Captura: proceso mediante el cual se obtiene el stream.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La arquitectura Multidifusión es análoga a la forma en que funciona la televisión por antenna. Cada cliente sintoniza el canal deseado y recibe los datos en la manera en que se encuentren en el periodo de duración. En el caso del streaming por Internet usando Multidifusión, el servidor está siempre enviando una señal de la transmisión a los clientes que decidan escuchar. (6)

La información es enviada por un único flujo, todos los usuarios comparten los flujos que se envían. La figura 1 muestra la arquitectura multidifusión.

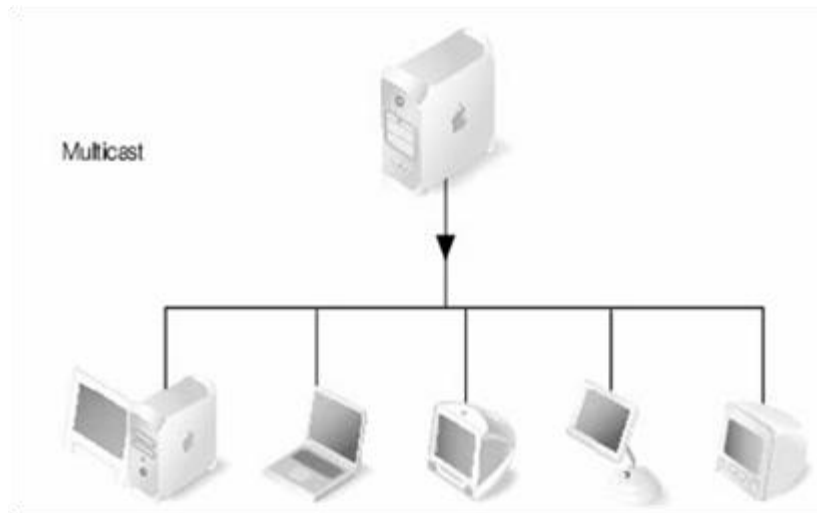


Figura 1 Arquitectura Multidifusión. (7)

Multidifusión está basado en las siguientes premisas dadas por S. Deering en (8):

Los emisores multidifusión podrán enviar datagramas a grupos multidifusión sin necesidad de unirse a los grupos, mientras que los receptores del mismo tipo sí deberán unirse para poder recibir el tráfico multidifusión. Los enrutadores multidifusión o “mrouter²” se encargarán de hacer que los flujos emitidos lleguen desde los emisores hasta los receptores, (9). Si bien se añade la complejidad de necesitar enrutadores con capacidad para gestionar flujos multidifusión, también es cierto que permite una reducción muy significativa del ancho de banda consumido. De modo que se puede conseguir que un simple dispositivo móvil pueda emplearse para videoconferencias con miles de participantes, (9). Multidifusión reduce el tráfico en la red y evita posibles congestiones en

² Mrouter: enrutador para la interconexión de redes con servicios multidifusión.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

los enrutadores; pero es obligatorio tener acceso a una troncal³ con soporte para multidifusión; o que los clientes y el servidor estén conectados a una red o redes IP bajo un mismo dominio de administración, en las que la multidifusión esté habilitado y existan enrutadores dispuestos a encaminar la información multidifusión.

Con la arquitectura multidifusión la complejidad recae en los enrutadores, que serán los encargados de replicar los datagramas por sus diferentes interfaces sólo cuando sea necesario. De forma general, se podría decir que los enrutadores se encargan de determinar las interfaces por las que se tendrá que reenviar un determinado datagrama en función de la interfaz de entrada, así como de algunos datos que obtienen mediante la ejecución de algún algoritmo de enrutamiento multidifusión. (9)

- **Unidifusión:**

La arquitectura Unidifusión se caracteriza porque cada usuario que consume el servicio, lo hace a través de una conexión cliente–servidor, en la cual cada cliente inicia su propio stream. Se envía un flujo de información a cada consumidor conectado. Si se desea enviar la misma información y hay “n” destinatarios, habrá “n” comunicaciones independientes punto a punto, o “n” copias de la misma información enviadas desde la máquina origen. La figura 2 muestra la arquitectura unidifusión.

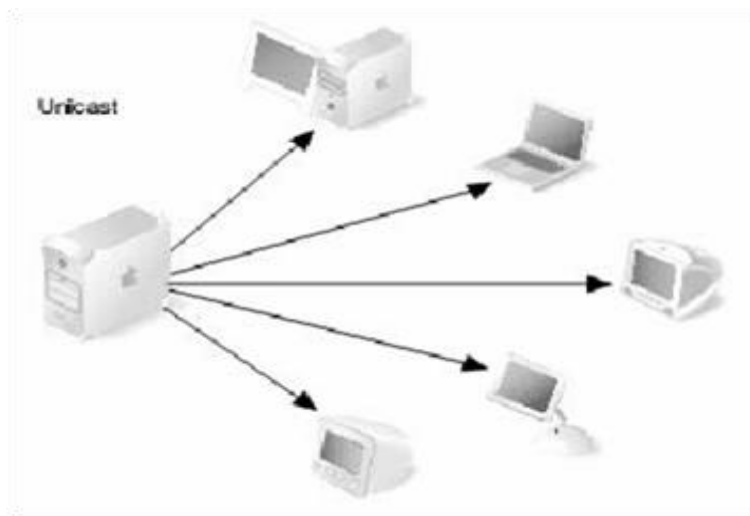


Figura 2 Arquitectura Unidifusión. (7)

³ Red troncal multimedia: permite actualmente la realización de audio y videoconferencias entre centenares de usuarios remotos mediante dos canales de video y cuatro de audio.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Difusión (Broadcast):**

Se basa en un único proceso de envío, independientemente del número de potenciales máquinas receptoras, de una misma información en una o más unidades de datos (datagramas) desde un origen a todas las máquinas de una red de área local, (4). Todo ello, sin necesidad de transmitir desde el origen una copia de la misma información, por separada, a cada una de dichas máquinas. Se resalta el hecho de que desde la máquina origen sólo se envía una vez la pertinente información y no se transmiten “n” copias de la misma aunque haya “n” destinatarios.

Con una de estas arquitecturas, cualquiera que se use, el usuario puede consumir el servicio de dos maneras, la primera: en el instante en que está ocurriendo el suceso, como es el caso de los eventos en vivo y la segunda: de forma diferida como películas, series, documentales o programas radiales que hayan sido grabados.

1.1.3 Formas de streaming

El streaming puede ser clasificado dentro de dos grupos según la forma en la que se va a obtener la información: en directo o bajo demanda, ampliados a continuación.

Streaming en directo: ocurre cuando se transmiten eventos que están ocurriendo en tiempo real como juegos deportivos, conferencia, eventos competitivos, docentes o de otra clase. La transmisión de radio y televisión por Internet también tiene estas características, aunque en ocasiones parte de la información que se difunde no proviene de un evento en directo; por ejemplo, un programa que ha sido grabado previamente, pero que se va a difundir en un momento determinado.

Streaming bajo demanda: la transmisión del medio comienza al inicio del evento. El medio a transmitir puede estar preparado desde el comienzo del proceso en un fichero comprimido. En este caso no representa una ventaja adicional el disponer de la posibilidad del realizar streaming multidifusión en la red, puesto que cada usuario recibe una parte distinta del stream y por lo tanto un paquete de datos diferente. (10)

La comunicación entre el servidor y el dispositivo del usuario, al igual que en otros servicios, se realiza usando protocolos estándares que permiten el entendimiento. No importa si es servicio bajo demanda o en directo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1.4 Protocolos de comunicación

Un protocolo es un método estándar que permite la comunicación entre procesos (que potencialmente se ejecutan en diferentes equipos). Es un conjunto de reglas y procedimientos que están obligados a cumplir todas las máquinas y programas que intervienen en una comunicación. El mismo debe respetarse para el envío y recepción de datos a través de una red, pues sin éste la comunicación resultaría caótica y por lo tanto imposible. (11) (12)

La comunicación entre el cliente y el servidor en los servicios streaming se realiza a través de protocolos de comunicación. Dichos protocolos se encuentran en la capa de transporte, sesión y aplicación.

1.1.4.1 Protocolos de transporte

El Protocolo de Datagrama de Usuario (UDP) y el Protocolo de Control de Transmisión (TCP) proveen las funciones básicas de transporte mientras que RTP y RTCP son portadores de la información a transmitir así como su control. (10)

UDP y TCP realizan las funciones de multiplexado⁴, control de error o control de flujo, multiplexado de los flujos de datos de las diferentes aplicaciones corriendo en la misma máquina con la misma dirección IP. Luego, con el objetivo de control de errores, las implementaciones de los mismos realizan un control de paridad para detectar errores de bit. Si se detecta un error, TCP o UDP descarta el paquete para que la capa superior (por ejemplo, RTP) no lo reciba. A diferencia de UDP, TCP utiliza retransmisiones para recuperar el paquete descartado, lo que hace que sea un protocolo de transmisión confiable. TCP también utiliza control de flujo para adaptar la tasa de transmisión según el nivel de congestamiento de la red. (10)

Dado que las retransmisiones de TCP provocan demoras, UDP es el protocolo más usado para streaming de video. Debido a que UDP no asegura la entrega, el receptor deberá confiar en la capa superior (RTP), para detectar las pérdidas de paquetes, (10). En la presente investigación se usan UDP y RTP como protocolos de transporte, así como el

⁴ Multiplexación: es la combinación de dos o más canales de información en un solo medio de transmisión usando un dispositivo llamado multiplexor.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

protocolo RTCP para el control de la QoS, así como el protocolo RTSP para el control de la sesión.

1.1.4.2 Protocolo de Streaming en Tiempo Real

El protocolo RTSP es usado para transmitir contenido multimedia en tiempo real. Permite hacer un control directo, en el cliente, de la reproducción del contenido enviado desde el servidor por medio de comandos como comenzar y detener la reproducción. Según (10), RTSP permite la elección de los canales de envío (por ejemplo UDP, o multidifusión UDP o TCP). Los mecanismos de envío basados en RTP y RTSP funcionan tanto en difusión punto a punto como en multidifusión. RTSP también provee:

- Recuperación de medios a solicitud del cliente.
- Invitación de un servidor de medios a una conferencia.
- Adición de medios a una sesión existente.

Los servicios basados en RTSP para la transmisión del contenido, hacen uso del protocolo RTP, el cual a su vez usa el protocolo UDP. (13)

1.1.4.3 Protocolo de Transporte en Tiempo Real y Protocolo de Control en Tiempo Real

RTP es un protocolo que se emplea para la transmisión de información en tiempo real. Proporciona soporte para el transporte de flujos de audio y de video. Comúnmente se usa junto con el Protocolo de Control en Tiempo Real (RTCP), (14). Dicho protocolo proporciona información de control. Está asociado con un flujo de datos para una aplicación multimedia (flujo RTP). Trabaja junto con RTP en el transporte y empaquetado de datos multimedia, pero no transporta ningún dato por sí solo. Se usa habitualmente para transmitir paquetes de control a los participantes de una sesión multimedia de streaming. La función principal de RTCP es informar de la calidad de servicio proporcionada por RTP.

Este protocolo recoge estadísticas de la conexión y también información como por ejemplo bytes, paquetes enviados, paquetes perdidos, entre otros. Una aplicación puede usar esta información para incrementar la calidad del servicio, ya sea limitando el flujo o

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

usando un codec de compresión más bajo. En resumen RTCP se usa para informar sobre la calidad del servicio.

Las funciones que provee RTP expuestas en (14) son:

- **Marcado-temporal:** las marcas temporales permiten sincronizar diferentes flujos de medios.
- **Numeración de secuencias:** dado que UDP no envía los paquetes en secuencia, RTP los numera para que puedan ser ordenados a su llegada.
- **Identificación del tipo de carga:** se identifica el tipo de carga útil en el paquete con un campo de cabezal RTP.
- **Identificación de fuente:** cada paquete RTP se identifica con un cabezal llamado SSRC que actúa como identificador de la fuente.

RTCP es un protocolo de control diseñado para trabajar junto con RTP y provee los siguientes servicios:

- **Realimentación de QoS:** es la función principal del RTCP. La información se envía a través de reportes de remitente y reportes de receptor. Estos reportes contienen la información de: fracción de paquetes RTP perdidos desde el último reporte, número de paquetes perdidos acumulado desde el comienzo de la recepción, jitter de paquetes y demora desde la recepción del último reporte de remitente.
- **Identificación del participante:** la fuente puede ser identificada por el campo SSRC en el cabezal RTP, fuente de sincronización, identificación de manera única dentro de una sesión RTP que la fuente emite. Los SSRC se eligen al azar, cuidando que no se repitan.
- **Escalado de control de paquetes:** para escalar según el número de participantes, se mantiene el número total de paquetes a un 5 % de ancho de banda total de la sesión. A su vez, dentro de los paquetes de control, un 25 % se utiliza para reportes de envío y un 75 % para reportes de recepción.
- **Información de control de sesión mínima:** transporta información de la sesión.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Hasta aquí los conceptos básicos relacionados a los servicios streaming, ahora, ¿cómo se comportan?, para responder la pregunta anterior es necesario pasar a otros conceptos relacionados a la calidad de prestación.

1.2 Calidad en servicios streaming de audio y video

En streaming de audio y video, como en otros servicios telemáticos, es importante saber con qué calidad perciben los consumidores el servicio prestado. Resulta difícil conocer cuándo el servicio está con los elementos técnicos en uso a un nivel que garantice la conformidad de los usuarios. Streaming suele coexistir con otros servicios que se brindan de manera conjunta en forma de un paquete, lo cual pone aún más difícil el proceso de determinar el grado de satisfacción de los usuarios. Para garantizar la QoE en servicios streaming deben tenerse en cuenta varios aspectos. No basta con tener un buen estado técnico de las conexiones o las tecnologías necesarias para prestar el servicio. Además de los indicadores que puedan ser identificados como elementos de QoS que afecten la QoE, hay que tener en cuenta otros elementos, que van desde la forma en la que se presenta el servicio hasta la disponibilidad del mismo. La investigación se centra en cómo afecta el comportamiento de algunos indicadores de QoS a la QoE. Para ello primero se debe comprender que es cada uno de estos conceptos.

1.2.1 QoS y QoE

La QoS se define como la capacidad de la red para proporcionar un servicio a un nivel garantizado. Abarca todas las funciones, mecanismos y procedimientos en la red que garanticen la obtención de la calidad de servicio negociada entre los extremos. Según (15), es el conjunto de características relacionadas con el rendimiento de los elementos que intervienen en la provisión del servicio, que tienen un efecto en la percepción final del usuario.

QoE es cómo un usuario percibe el estado de un servicio cuando es usado. El término se refiere a la percepción del usuario sobre la calidad de un determinado servicio o red. Se expresa en sentimientos humanos como: “bueno”, “excelente” o “pobre”. Por otra parte, QoS es intrínsecamente un concepto técnico que se mide expresado y entendido en términos de redes y elementos de red, que por lo general tienen poco significado para los usuarios.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Los conceptos de QoE y QoS, aunque tienden a ser confundidos, son conceptos diferentes. Según (16), cuando se analiza un servicio hay tres capas diferentes:

Capa de servicio, es la que está expuesta al usuario, es donde se definen y miden las variables que contribuyen a una calidad de experiencia satisfactoria.

Capa de aplicación, donde se configuran los diferentes parámetros de la aplicación (por ejemplo tipo de codificación, tasa de bits, etc.) que son requeridos para una calidad de experiencia satisfactoria.

Capa de transporte, donde se deben de aplicar mecanismos de calidad de servicio y corrección de errores para ajustar los parámetros requeridos por el servicio de tal forma que se obtenga una calidad de experiencia máxima.

Una red con mejor QoS, en muchos casos dará como resultado una mejor QoE. El objetivo de QoS es obtener una alta QoE. (2)

La entrega de alta calidad de la experiencia depende de profundizar en el conocimiento de los factores que contribuyen a la percepción de los usuarios de los servicios, y aplicar ese conocimiento para definir los requisitos. Este enfoque, desde arriba hacia abajo, reduce los costes de desarrollo, los riesgos de rechazo y deserción de los usuarios. Además asegura que el dispositivo o sistema cumpla con los requerimientos de los mismos. (2)

Para medir la QoE de los servicios streaming de audio y video es necesario partir de qué indicadores informan cómo se comporta el servicio en la capa de transporte. (16)

1.2.2 Indicadores para medir la QoE desde la QoS en servicios streaming

Según la norma vigente descrita en (17) y el informe sobre QoE en servicios multimedia descrito en (16), los siguientes indicadores afectan a los servicios streaming de la manera que se expone a continuación:

- **Retardo:** las aplicaciones de tiempo real como las de streaming requieren demoras inferiores a los 150 ms. Si la latencia (tiempo que le toma a un paquete ir de un lado al otro, o de extremo a extremo) sobrepasa estos valores, la transmisión o comunicación se verá afectada con paradas en los cuadros (frame) de video o silencio en el audio. Por la saturación de la red, los enrutadores pueden

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

demorarse en procesar los paquetes, causando demoras en el arribo al no haber suficientes datos en el buffer. Esto provoca indisponibilidad en el servicio, bloqueo o pausa de la aplicación, lo que implica que la reproducción del archivo se vea afectada.

- **Jitter (variación del retardo):** es la diferencia entre las demoras introducidas por la red. Provoca defectos en la imagen debido a que los componentes de red no actúan siempre bajo las mismas circunstancias. No introduce demoras iguales, lo que puede provocar que los paquetes lleguen desordenados o a destiempo.
- **Ancho de banda** (de ahora en adelante AB): es fundamental. De él depende hasta qué punto se podrá brindar QoS a un nivel garantizado. Si el medio de transmisión está limitado en ancho de banda y se necesita transmitir video, telefonía IP, datos y web, se debe tener en cuenta hasta qué punto se podrá reservar ancho de banda para las aplicaciones de tiempo real sin que se vean afectadas las demás.

Caso A

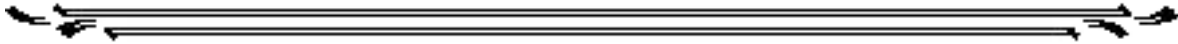
Se implementa QoS para que los servicios de tiempo real tengan reservado el AB suficiente para un buen desempeño. Las demás aplicaciones pueden verse afectadas dependiendo del tráfico de la red.

Caso B

Se implementa QoS pero no se asigna todo el AB necesario para el mejor desempeño de los servicios de tiempo real. Los usuarios quedan insatisfechos con las conversaciones y los videos.

- **Velocidad de transmisión** (de ahora en adelante Tx): está estrechamente ligada al AB que al mismo tiempo depende del medio de Tx. El tema de la movilidad es fundamental en este punto. Se necesita un método de acceso al medio eficiente en estos casos para poder aprovechar al máximo el AB. Las interferencias son superiores en los enlaces de radio. La velocidad de desplazamiento de la entidad móvil también define en el desempeño del sistema.
- **Demora en el comienzo de la reproducción:** hasta que no haya suficiente información en el buffer como para comenzar a reproducir, la reproducción no iniciará.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA



- **Pérdida de paquetes:** una cantidad de paquetes perdidos que sobrepase los valores permisibles puede provocar que la reproducción se interrumpa y genere efecto de bloque en el video y discontinuidades en el audio, entre otros efectos.

Una vez conocidos los indicadores de QoS para poder medir la QoE es necesario saber, ¿en qué rango de valores deben de oscilar sin que afecten la QoE? En el próximo epígrafe se exponen dichos valores.

1.2.3 Cuantificaciones para las métricas definidas de QoS

En la norma vigente descrita en (17) se definen de la siguiente forma los umbrales de valores para los indicadores como se muestra en la tabla 1:

Tabla 1 Objetivos de rendimiento para aplicaciones de audio y video. (17)

Medio	Aplicación	Grado de Simetría	Tasa de datos típica	Parámetros de rendimiento			
				Retardo en un sentido	Variación del retardo	Pérdida de paquetes	Otras
Audio	Voz convencional	Bidireccional	4 – 64 KB/s	Límite < 400 ms Preferible <150 ms	< 1 ms	< 3%	
Audio	Mensaje de voz	Básicamente unidimensional	4 – 32 KB/s	< 1 s para reproducción < 2 s para grabación	< 1 ms	< 3%	
Audio	Distribución de audio de alta calidad	Básicamente unidimensional	16 – 128 KB/s	< 10 s	<< 1 ms	< 1%	
Video	Video conferencia	Bidireccional	16 – 384 KB/s	Límite < 400 ms Preferible <150 ms		< 1%	Retardo audio video < 80 ms
Video	Distribución de video	unidimensional	16 – 384 KB/s	< 10 s		< 1%	

Determinados los indicadores y los valores umbrales para cada uno, es necesario definir los modelos para medir la QoE. Para ello es necesario realizar un estudio que permita la identificación de los mismos.

1.2.4 Modelos para medir la QoE en servicios telemáticos a partir de QoS

Algunos de los modelos para medir la QoE en servicios telemáticos a partir de indicadores de QoS son los de estimación. Básicamente tratan de representar de forma cuantitativa, la relación existente entre los indicadores de QoS y la QoE que perciben los usuarios cuando consumen un servicio. Los modelos se establecen a partir de numerosas pruebas experimentales en diferentes escenarios. Permiten la evaluación continua del servicio. Además se caracterizan por estar centrados en un servicio específico.

El primer modelo a tener en cuenta es Puntuación Media de Opinión (MOS), por ser usado en los modelos que se refieren aquí.

1.2.4.1 El modelo MOS

El modelo MOS está documentado en (18). En (19) se establece la terminología relacionada con la escala de evaluación subjetiva, inicialmente aplicada para la evaluación subjetiva de las comunicaciones de voz tradicionales. El método es extendido a aplicaciones multimedia en (20), para comunicaciones de audio unidimensionales no interactivas; en (21) para audio y video unidimensionales no interactivas como la difusión de televisión, aplicable también al video bajo demanda; y en (22) para la aplicación en comunicaciones de audio y video bidireccionales como la videoconferencia y videotelefonía.

1.2.4.2 Otros modelos

A continuación se listan otros modelos aplicados a diferentes tipos de servicios que han sido usados como base para la decisión del modelo a aplicar.

- **Modelo E:** se creó como modelo de planificación de redes de telefonía y posteriormente usado para la estimación de la calidad percibida en redes operativas. El modelo combina diferentes factores de degradación de la señal de voz, de modo que a partir de valores de determinados parámetros, se calcula la estimación de la señal ruido (factor R) y luego la estimación de la calidad percibida con el método (MOS). (23)
- **PEAQ (Evaluación Perceptual de la Calidad de Audio):** es un algoritmo estandarizado para medir objetivamente la calidad de audio percibida para codificadores de baja tasa de codificación, utilizando un modelo resultante de la

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

combinación de 6 modelos indicados. Es el resultado del análisis del rendimiento de esos modelos y la integración de las mejores funciones de cada uno en un único modelo. El modelo es aplicable a la mayoría de los dispositivos de procesamiento de audio tanto analógico como digital. Este modelo no debe ser visto como un sustituto de las pruebas subjetivas. (24)

- **VQM (Modelo de Calidad de Video):** se basa en la comparación de la señal recibida con una señal fuente, para la estimación de la calidad percibida para la televisión. El software para la aplicación del modelo está disponible de forma gratuita junto con manuales técnicos y la descripción de los algoritmos utilizados. (25)

Para el caso de streaming de audio y video, no se encontró un modelo general que incluya indicadores críticos que afecten a la QoE. Por tanto, se propone desarrollar un modelo que incluya medidas objetivas sobre el rendimiento de indicadores técnicos.

1.2.4.3 Modelo a usar en la herramienta

Para la determinación del modelo a aplicar en la herramienta es necesario tener en cuenta los modelos mencionados en los epígrafes 1.2.4.1 y 1.2.4.2 debido a que están orientados a servicios similares que tienen básicamente los mismos indicadores de QoS que streaming. La forma en la que se pueden adaptar los indicadores y la categorización de los mismos en próximos capítulos se explica detalladamente.

Finalmente es útil disponer de una herramienta para obtener una medida de la QoE de los usuarios, que tenga en cuenta los indicadores críticos de rendimiento y haga seguimiento del comportamiento de los mismos. Algunas herramientas que dan una medida de cómo se comportan algunos servicios telemáticos son mencionadas a continuación. Es válido destacar que ninguna analiza el punto de vista de los usuarios finales.

1.2.5 Herramientas de software relacionadas

La persona responsable de administrar un servicio de telecomunicación se basa en programas de monitorización de red y/o servidores que pueden detectar comportamientos defectuosos. Por ejemplo, para conocer el estado de un servidor web, el software de monitoreo periódicamente puede enviar una petición HTTP a buscar una página. Para los servidores de correo electrónico, un mensaje de prueba puede ser enviado a través de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

SMTP y traído por IMAP o POP3. En el caso de solicitar el estado de las fallas, el software de monitoreo puede enviar un mensaje de alarma al administrador de sistemas para que tome acciones correctivas. (26)

Alguna de las herramientas de monitoreo que pueden manejar pequeñas y grandes infraestructuras son las listadas a continuación:

- **Nagios:** es considerado como uno de los más populares sistemas de monitorización de red de código abierto disponible. Fue diseñado originalmente para ejecutarse en Linux, pero otras variantes de Unix son soportadas también. Nagios proporciona supervisión de los servicios de red (SMTP, POP3, HTTP, NNTP, ICMP, SNMP, FTP, SSH) y recursos de host (carga del procesador, uso de disco, los registros del sistema) entre otros. El control remoto es manejado a través de túneles SSH o SSL cifrados. Nagios tiene un diseño simple que ofrece a los usuarios la libertad para desarrollar sus chequeos de servicio sin esfuerzo propio basado en las necesidades, mediante el uso de cualquiera de las herramientas de apoyo que guste. Para detectar y diferenciar entre los hosts que están abajo y los que son inalcanzables, Nagios permite definir la jerarquía de la red de acogida con los hosts "padre". Cuando los servicios o los problemas de acogida se plantean, la notificación es enviada a la persona que está a cargo de la red a través del correo electrónico o SMS. (26)
- **Zabbix:** es una clase de mecanismo de vigilancia tipo empresarial que está completamente equipado y tiene soporte comercial. Es capaz de monitorear y dar seguimiento de la situación de los diferentes tipos de servicios de red. Zabbix tiene grandes funcionalidades de visualización, incluidas las vistas definidas por el usuario, zoom y la cartografía. Tiene un método de comunicación versátil que permite una configuración rápida y sencilla de los diferentes tipos de notificaciones de eventos predefinidos. Además cuenta con tres módulos principales: el servidor, los agentes y el usuario. Para almacenar los datos de seguimiento, puede utilizar MySQL, PostgreSQL, Oracle o SQLite como base de datos, sin necesidad de instalar ningún software en el host de seguimiento. Zabbix permite a los usuarios comprobar la disponibilidad y capacidad de respuesta de los servicios estándares, como SMTP o HTTP. Para supervisar las estadísticas tales como carga de la CPU, utilización de la red y espacio en disco, un agente de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Zabbix debe estar instalado en la máquina host. Zabbix incluye soporte para el monitoreo a través de SNMP, TCP y controles ICMP, IPMI y parámetros personalizados como una opción para instalar un agente en los hosts. (26)

- **Cacti:** es una herramienta web de gráficas que está diseñada como una interfaz completa para almacenamiento de datos de RRDtool⁵ y la utilidad gráfica que permite a los usuarios monitorear y graficar la carga de la CPU, la utilización de ancho de banda de red, el tráfico de red y mucho más. Puede ser utilizado para configurar la recopilación de datos en sí, lo que permite configuraciones particulares, a controlar sin ningún tipo de configuración manual de RRDtool. Cacti permite sondear los servicios en el período preestablecido y el gráfico de los datos resultantes. Se utiliza principalmente para representar de manera gráfica los datos de series temporales de parámetros tales como la carga de la CPU y la utilización de ancho de banda de red. Cacti se puede ampliar para controlar cualquier fuente a través de scripts de Shell⁶ y ejecutables. También es compatible con arquitectura de complementos. Tiene una comunidad grande y activa que se ha reunido en torno a los foros de Cacti para proporcionar scripts, plantillas y consejos sobre creación de complementos. (26)
- **Zenoss (Zenoss Core):** basado en el servidor de aplicaciones Zope y escrito en Python⁷. Es un servidor y la plataforma de gestión de red que combina la programación original y varios proyectos de código abierto para integrar el almacenamiento de datos y los procesos de recopilación de datos a través de la interfaz de usuario basada en Web. Permite a los usuarios supervisar la disponibilidad, inventario, configuración, desempeño y acontecimientos. Zenoss Core es capaz de supervisar la disponibilidad de dispositivos de red mediante SNMP, SSH, servicios de red (HTTP, POP3, NNTP, SNMP, FTP) y los recursos del host (procesador, uso de disco) en la mayoría de sistemas operativos de red. (26)

⁵ Round Robin Database Tool, Se trata de una herramienta que trabaja con una base de datos que maneja planificación según Round-Robin.

⁶ Scripts de Shell: es un archivo que contiene uno más comandos del Shell. Se puede crear un script a partir de un editor de textos, el Shell es un programa especial utilizado como una interfaz entre el usuario y el núcleo del Sistema Operativo (Kernel).

⁷ Python: es un lenguaje de programación interpretado, orientado a objetos y multiplataforma cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Munin:** al igual que los Cacti, utiliza RRDtool para presentar resultados en gráficos a través de una interfaz web. Cuenta con una arquitectura de maestro/nodo en el que el maestro enlaza a todos los nodos a intervalos regulares y que solicita los datos. Usando Munin, se puede supervisar rápida y fácilmente el rendimiento de los equipos, las redes y las aplicaciones. Esto hace que sea sencillo para detectar cuándo se produce un problema de rendimiento y ver claramente cómo lo está haciendo de la capacidad racional de todos los recursos restringidos. Para Munin, la prioridad principal es la arquitectura conectar y listo⁸. Tiene varios complementos de control disponibles que fácilmente funcionarán sin una gran cantidad de modificaciones. (26)

Ninguna de las mencionadas anteriormente, pese a su innegable relevancia, estaría aplicable a un servidor streaming. Están enfocadas más bien en otros tipos de servicios y servidores aunque la información que brindan sería útil. Por ejemplo para saber la carga la CPU y otros dispositivos de hardware usados en la red cuando existe un flujo de conexiones determinado. Para el caso de streaming MComms Monitors se ajusta bastante. MComms Monitors proporciona una manera fácil y confiable para monitorear múltiples flujos a nivel de contenido con una vista previa en la pantalla.

MComms Monitors según (27), se conecta al servidor de streaming de video de la misma manera que lo hacen sus clientes. A continuación, recibe las corrientes de audio y video, los decodifica y los chequea en busca de silencios / pantalla azul / pantallas negras (dos de los problemas de entrega más comunes que suelen aparecer cuando ocurre un problema en la transmisión) y plantea una alarma para los flujos fallidos.

Los operadores de servidores streaming de video pueden utilizar MComms Monitors para visualizar todas sus secuencias de video en la pared de un centro de gestión de red, o simplemente en la pantalla de un ordenador de sobremesa para pruebas rápidas de corrientes múltiples. Con un "semáforo" en la pantalla de interfaz gráfica de usuario para cada flujo, las operaciones y el personal de mantenimiento puede ver de un vistazo qué corrientes tienen problemas. El número de secuencias que se pueden visualizar simultáneamente está limitado solamente por el espacio de pantalla disponible, (27).

⁸Del inglés Plug and play: se refiere a la capacidad de un sistema informático de configurar automáticamente los dispositivos al conectarlos. Permite poder enchufar un dispositivo y utilizarlo inmediatamente, sin preocuparte de la configuración.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

MComms Monitors no tiene en cuenta los indicadores propuestos en 1.2.2 ni una medida del estado de satisfacción del usuario que está consumiendo el servicio. Es una muy buena herramienta por las características expuestas pero su uso por mes en dependencia de la categoría básica, avanzada o final está entre los \$35 y \$95 mensuales.

Por lo descrito anteriormente se decide el desarrollo de una con las características que respondan a las necesidades. Para su confección serán usadas las tecnologías que se explican en los próximos epígrafes.

1.2.6 Tecnologías utilizadas

Las siguientes tecnologías han sido usadas en el proceso de implementación de la herramienta.

1.2.6.1 Lenguaje de Programación

Se escogió como lenguaje de programación Microsoft C# en su versión 4.0 porque está diseñado para crear un amplio número de aplicaciones empresariales que se ejecutan en .NET Framework. Es sencillo, proporciona seguridad de tipos y está orientado a objetos. El código creado mediante C# se compila como código administrado, lo cual significa que se beneficia de los servicios de Common Language Runtime⁹. Estos servicios incluyen interoperabilidad entre lenguajes, recolección de elementos no utilizados, mejora de la seguridad y mayor compatibilidad entre versiones. Permite la creación de aplicaciones para Windows, servicios web, herramientas de base de datos, componentes, controles y mucho más. (28)

C# es un lenguaje de programación que utiliza clases y estructuras para implementar tipos como formularios Windows Forms¹⁰, controles de interfaz de usuario y estructuras de datos. Una aplicación de C# típica se compone de clases definidas por el programador, junto con clases de .NET Framework. Ofrece varias maneras eficaces de definir clases, proporciona niveles de acceso diferentes, hereda características de otras clases y permite que los programadores especifiquen qué sucede cuando se crean o destruyen instancias de tipos. Las clases también se pueden definir como genéricas mediante la utilización de

⁹ Common Language Runtime: es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET.

¹⁰ Windows Forms: es el nombre dado a una interfaz gráfica de aplicación Windows que se incluye como parte de .NET Framework.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

parámetros de tipo que permiten que el código del cliente personalice la clase de una manera eficiente y con seguridad de tipos. (29)

1.2.6.2 Framework de desarrollo

.NET Framework en sus versión 4.0 fue seleccionado por ser un componente integral de Windows que admite la compilación y ejecución de la siguiente generación de aplicaciones y servicios webs XML. Su diseño está encaminado a: proporcionar un entorno coherente de programación orientado a objetos y un entorno de ejecución de código que reduzca al máximo la implementación de software y los conflictos de versiones; promover la ejecución segura del mismo, que elimine los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos; y basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código. (30)

Facilita la implementación de una amplia gama de aplicaciones. Permite aislar la aplicación, eliminar conflictos de archivos DLL y posibilita la coexistencia de varias versiones de un componente o de una aplicación. Ofrece actualizaciones inmediatas, integración con Microsoft Windows Intaller¹¹ y descarga y almacenamiento en caché¹². Además los componentes y aplicaciones auto descriptivo e independiente pueden implementarse sin entradas del registro o dependencias. (31)

1.2.6.3 Entorno de Desarrollo Integrado

Como entorno de desarrollo integrado se seleccionó Visual Studio 2010 Ultimate por ser un conjunto completo de herramientas de desarrollo para la generación de aplicaciones móviles, de escritorio, web y servicios web XML. Soporta varios lenguajes de programación como Visual Basic, Visual C#, Visual C++ y ASP.NET. Habilita el uso compartido de herramientas y utiliza las funciones de .NET Framework. (32)

Simplifica todo el proceso de desarrollo de aplicaciones. Ofrece resultados empresariales usando procesos productivos, predecibles y personalizables, además de aumentar la transparencia y seguimiento durante el ciclo de vida con análisis detallados. Facilita la elaboración de prototipos, el uso de arquitecturas de desarrollo para la creación y mejora

¹¹ Windows Intaller:

¹² Caché: es un área especial de memoria que poseen los ordenadores.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

de soluciones. Aumenta la productividad del equipo usando características de colaboración avanzadas y herramientas de pruebas y depuración integradas para detectar y corregir errores de un modo rápido y sencillo, lo que permite elevar la calidad así como la reducción del costo de desarrollo. (33)

1.2.6.4 Sistema Gestor de Base de Datos

SQLite es una biblioteca de software que se enlaza con el programa pasando a ser parte integral del mismo. Implementa un gestor de bases de datos SQL embebido, sin configuración. Los usuarios más conocidos que la utilizan actualmente en sus aplicaciones son: Adobe, Apple, Mozilla, Google, McAfee, Microsoft, Philips, Sun y Toshiba, entre otros. Con todas las características habilitadas, el tamaño de la librería es inferior a 250Kb. Deshabilitando características opcionales, el tamaño puede quedarse por debajo de los 180Kb. Esto la hace muy apropiada para usarla en dispositivos con poca memoria, como teléfonos móviles, PDAs y reproductores MP3. Requiere muy poco soporte de librerías externas o del sistema operativo. Esto la hace adecuada para usarla en pequeños dispositivos que no son tan completos como los ordenadores de escritorio. (34)

Con SQLite, el proceso para acceder a la base de datos, lee y escribe directamente en disco. No hay servicio intermediario. De esta manera se puede hacer una aplicación totalmente autónoma y portable. Esto tiene ventajas y desventajas. La principal ventaja es que no debemos tener un servicio que instalar, configurar, inicializar, mantener, etc. (34)

Por otro lado, el uso de un servidor para la base de datos provee mayor protección frente a errores en el lado de cliente. Además, como el servidor es un único proceso, puede controlar mejor la concurrencia. (34)

1.2.6.5 Metodología de desarrollo

Programación Extrema (XP, por sus siglas en inglés) fue escogida como metodología de desarrollo de software debido a que es ágil y se centra en potenciar las relaciones interpersonales como clave para el éxito. Además promueve el trabajo en equipo, el aprendizaje de los desarrolladores y propicia un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

para enfrentar los cambios. Es adecuada para proyectos con requisitos imprecisos, muy cambiantes y donde existe un alto riesgo técnico. (32)

Conclusiones del capítulo

Mediante el estudio realizado en el capítulo que se concluye se adquieren los conocimientos teóricos necesarios para la elaboración de una herramienta para la monitorización de indicadores de calidad de experiencia en servicios streaming. Para ello se parte de la comprensión del servicio, su arquitectura, formas de distribución y protocolos de comunicación, para luego caer en el tema de la calidad y algunas sugerencias de cómo medirla, que sirvan de base para la próxima fase de la investigación.

Como resultado del análisis realizado sobre las herramientas y tecnologías relacionadas con la medición de calidad de experiencia se determina que hay muy pocas en el mercado y están orientadas a otros servicios. Las que monitorizan servicios streaming son potentes, pero no se adaptan a las necesidades de la presente investigación pues no tienen en cuenta la QoE. De ahí la necesidad de desarrollar una que sea capaz de aplicar los resultados del presente estudio y disminuir los costos por compras de software y licencias. También se han definidos las tecnologías para el desarrollo del sistema, preparando de este modo el escenario para la determinación de las características de la aplicación, la elaboración de una forma de evaluación de QoE y la selección de los indicadores más apropiados para ser usados en el proceso de monitorización y evaluación.

Capítulo 2: Características del Sistema

Introducción

El presente capítulo abarca los elementos relacionados con la solución desarrollada en torno a las fases de planeación y diseño de la metodología de desarrollo de software seleccionada, así como sus características. Se parte de un análisis de los indicadores que se tienen en cuenta para la medición de los servicios streaming de audio y video. Luego se correlacionan los indicadores en una propuesta de modelo que proporciona una clasificación de calidad para los servicios. Además se aborda acerca de la arquitectura en la cual fue implementada la herramienta, los patrones de diseño de programación utilizados entre otros aspectos importantes.

2.1 Indicadores de QoS

Jitter: en las redes de paquetes representa la diferencia entre las demoras introducidas. Debido a que es inevitable, los receptores deben recibir los paquetes a decodificar a intervalos constantes para poder regenerar de forma adecuada la señal original. Los mismos disponen de un buffer de entrada, con el objetivo de “suavizar” el efecto de la variación de las demoras. El buffer recibe los paquetes a intervalos variables, y los entrega a intervalos constantes. Para ello el buffer agrega una demora adicional al sistema, retiene los paquetes y realiza la entrega. Cuanta más variación de demoras exista, más grande deberá ser el buffer y por lo tanto, mayor será la demora introducida. Las demoras son indeseables y tienen impacto directo en la experiencia del usuario, sobre todo en contenidos de tiempo real (por ejemplo, distribución de eventos deportivos en línea) y en aplicaciones conversacionales (por ejemplo, video telefonía o video conferencias). Se hace necesario disponer de mecanismos que minimicen el tamaño de los jitter buffers, pero que a su vez, no comprometan la calidad debido a los paquetes perdidos o que no han llegado a tiempo.

Ancho de banda: el ancho de banda requerido en una red IP dependerá del tipo de codificación utilizada, de la resolución y del movimiento y textura de la imagen. La codificación digital de video utiliza algoritmos de compresión, los que generan codificación de largo variable y flujos de ancho de banda también variables. Al ancho de banda propio de la señal de video se le debe sumar la sobrecarga de los paquetes IP, UDP y RTP para

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

la LAN de las tramas Ethernet; todo lo que puede estimarse en aproximadamente un 25% adicional. A diferencia de la codificación de audio, donde el ancho de banda puede calcularse de forma exacta únicamente en base al codec utilizado, la codificación de video es estadística y depende de la imagen transmitida, por lo que los cálculos son también aproximados y estadísticos.

Para los servicios streaming un ancho de banda limitado hace imposible su implantación, debido a que estos tipos de servicios requieren del intercambio de gran cúmulo de información a través de las redes.

Pérdida de paquetes: cuando ocurren pérdidas de información afecta la calidad percibida y puede difundirse perjudicando no solo el contenido que viaja en el paquete perdido sino también en otras partes del video, posibilitando la propagación del error en el tiempo. Debido a que la codificación se realiza en forma diferencial, la pérdida de un paquete afectará a todos los bloques siguientes de la misma fila.

Existen técnicas de cancelación de paquetes perdidos que tratan de reconstruir la información perdida en base a la disponible. Por ejemplo, reemplazando los píxeles perdidos por los mismos valores de cuadros anteriores. Según (36) la pérdida de un paquete de video determinado puede o no afectar la calidad percibida. En (37) se propone un método que garantiza una calidad perceptual constante en la recepción del flujo de video. El mismo consiste en marcar sólo ciertos paquetes específicos con mayor prioridad. De esta forma se garantiza la llegada de los mismos a su destino sin inundar la red con todos los paquetes marcados como prioritarios.

Factor de compresión: el proceso de digitalización de video utiliza técnicas que transforman una secuencia de píxeles al dominio de la frecuencia espacial, cuantificando valores, descartando eventualmente componentes de alta frecuencia y haciendo uso de técnicas de predicción y compensación de movimientos. Ello genera ruido de cuantificación y puede degradar la imagen original a niveles perceptibles, que según (38) pueden verse en imágenes y videos con alta compresión; como es el conocido efecto de bloques, que hace ver a la imagen como un conjunto de bloques pequeños.

Demoras en el arribo de los cuadros: refleja el tiempo transcurrido entre la llegada de un cuadro y su inmediato sucesor, demoras elevadas es señal de mal funcionamiento del servicio y una evidencia de que el audio/video no está llegando al cliente con la calidad

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

requerida. Con altas demoras entre los cuadros la señal puede experimentar efectos de pausas momentáneas o el molesto efecto en bloques.

Demora en el inicio de la reproducción: una aplicación que tarde en responder demasiado tiempo es señal de que el servicio se encuentra colapsado o presentando problemas para responder las demandas. Las demoras en el inicio de la reproducción afectan a la experiencia de los usuarios que no están dispuestos a esperar demasiado tiempo y desertan su solicitud de consumo del servicio.

Una vez descritos los indicadores de QoS se prosigue con el modelo utilizado, el cual se aborda a continuación.

2.2 Modelo a emplear

Tomando como ejemplo el modelo E, debido a que es usado para determinar la QoE en servicios como telefonía con indicadores de QoS similares a los definidos para streaming de audio y video, (23) se propone para medir la calidad de experiencia un modelo que calcule una variable llamada factor R que expresa un valor cuantitativo de QoE. Luego el resultado de dicho factor, que debe ser un valor entre cero y cien, es llevado a una escala MOS descrita en 2.4. Las fórmulas matemáticas necesarias y procedimientos se describen en el próximo epígrafe.

2.3 Explicación del modelo para la medición de la QoE

El cálculo de la QoE estimada en un momento del tiempo está dado por el valor del factor R en una escala MOS. Para el cálculo del factor R ajustado a los servicios streaming de audio y video, y a los indicadores descritos anteriormente se propone la siguiente ecuación:

$$R = R_0 - D_r - D_{pp} - D_{ab} + A$$

Siendo R_0 un valor ideal igual a 100, el cual va a verse degradado en dependencia del comportamiento de los indicadores.

D_r , representa las degradaciones por retardos que se calcula de la siguiente forma:

$$D_r = R_{\text{jitter}} + R_{\text{frames}} + R_{\text{inicio de la reproduccion}}$$

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

$$R_{jitter} = f(j) = \begin{cases} 0, j \leq 1ms \\ 100, j \geq 400ms \\ 100j/400, 1 \leq j \leq 400ms \end{cases}$$

Siendo j el valor del Jitter en un instante de tiempo.

$$R_{frames} = f(d) = \begin{cases} 0, d \leq 1ms \\ 100, d \geq V_p ms \\ 100d/V_p, 1 \leq d \leq V_p \end{cases}$$

Siendo V_p el valor máximo permitido por el cliente streaming para recibir cuadros desde el servidor. En la investigación se usa un valor $V_p = 1000$ y d igual al valor de la demora entre el último cuadro recibido y el anterior.

$R_{inicio \text{ de la reproducción}}$, es igual al tiempo desde que se envía la petición de reproducir al servidor, más el proceso de llenar el buffer y comenzar la reproducción.

D_{pp} , representa las degradaciones producidas por las pérdidas de paquetes. Se calcula teniendo en cuenta el porcentaje de pérdidas permitido para aplicaciones de video, expuesto en el epígrafe 1.2.3, que es un 3 % máximo.

$$D_{pp} = C_{pp} * 100 / C_{pe}$$

C_{pp} , es la cantidad de paquetes perdidos.

C_{pe} , es la cantidad de paquetes esperados.

$$D_{ab} = f(ab) = \begin{cases} 100, ab < 16kb/s \\ 0, ab \geq 16kb/s \end{cases}$$

Siendo ab la velocidad del ancho de banda negociado entre la aplicación cliente y el servidor.

El parámetro A es sumado a la formula cuando el servicio está siendo consumido en situaciones extraordinarias como por ejemplo en un ómnibus o en un túnel; dichas condiciones provocan degradaciones de calidad que el usuario está dispuesto experimentar por tal de recibir el servicio. A tomará valores de cero para redes Ethernet y de tres para redes móviles en situaciones extraordinarias.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.4 Modelo MOS

El modelo se relaciona con el valor de “*R*”, con un gran nivel de aproximación, según la tabla 3 mostrada a continuación.

Tabla 2 Relación del factor *R* con en una escala MOS según (23).

Factor <i>R</i>	Categoría	Satisfacción del usuario
$90 \leq R < 100$	Excelente	Muy satisfechos.
$80 \leq R < 90$	Alta	Satisfechos.
$70 \leq R < 80$	Media	Algunos usuarios insatisfechos.
$60 \leq R < 70$	Baja	Muchos usuarios insatisfechos.
$50 \leq R < 60$	Mala	Prácticamente todos los usuarios insatisfechos.
NOTA: <i>R</i> por debajo de 50 no es recomendado.		

Una vez expuestos los algoritmos a utilizar es necesario definir las historias de usuario que regirán la implementación. Las mismas son mostradas a continuación.

2.5 Historias de usuarios

Tabla 3 Historia de usuario 1.

Historia de Usuario	
Número: 1	Usuario: Técnico de calidad.
Nombre: Conectarse con el servidor streaming.	
Prioridad del negocio: Alta.	Riesgo en desarrollo: Alta.
Puntos estimados: 5	Iteración asignada: 1
Programador responsable: Vladimir Alvarez Pérez.	
Descripción: El usuario debe ingresar la dirección del audio o video a reproducir, la dirección debe poseer una estructura (dirección válida) correcta e incluir la extensión del archivo, por ejemplo <code>rtsp://10.8.127.244/sample_100kbit.mp4</code> . Luego se establece la conexión con el servidor streaming mediante el protocolo RTSP.	
Observaciones:	

Tabla 4 Historia de usuario 2.

Historia de Usuario	
Número: 2	Usuario: Técnico de calidad.
Nombre: Consumir servicios streaming.	
Prioridad del negocio: Alta.	Riesgo en desarrollo: Alta.
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Vladimir Alvarez Pérez.	
Descripción:	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA



Luego de establecer la conexión con el servidor, una petición de reproducir es enviada al servidor para comenzar a consumir el servicio. También pueden ser enviadas peticiones de pausa, stop, correr hacia delante o hacia atrás.

Observaciones:

Ver anexo 1.

Tabla 5 Historia de usuario 3.

Historia de Usuario	
Número: 3	Usuario: Técnico de calidad.
Nombre: Mostrar información de la red	
Prioridad del negocio: Alta.	Riesgo en desarrollo: Bajo.
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Vladimir Alvarez Pérez.	
Descripción: Captura y muestra la información de la red relacionada con el tipo de red, la dirección IP, la velocidad de transmisión y el ancho de banda negociado entre la aplicación cliente y el servidor. La información es en tiempo real.	
Observaciones: Ver anexo 2.	

Tabla 6 Historia de usuario 4.

Historia de Usuario	
Número: 4	Usuario: Técnico de calidad.
Nombre: Capturar paquetes de la red.	
Prioridad del negocio: Alta.	Riesgo en desarrollo: Alto.
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Vladimir Alvarez Pérez.	
Descripción: La comunicación con el servidor streaming se realiza a través de paquetes de tipo RTSP, RTP y RTCP, los cuales viajan codificados, durante el proceso deben capturarse y decodificarse para poder ser analizados posteriormente. Cuando se completa un cuadro también debe ser capturado y decodificado.	
Observaciones:	

Tabla 7 Historia de usuario 5.

Historia de Usuario	
Número: 5	Usuario: Técnico de calidad.
Nombre: Mostrar paquetes de la red.	
Prioridad del negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Vladimir Alvarez Pérez.	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<p>Descripción: Una vez capturados y decodificados los paquetes de red estos son mostrados al usuario con sus correspondientes valores. Son visualizados por tipo de paquete y en orden de arribos.</p>
<p>Observaciones: Ver anexo 3.</p>

Tabla 8 Historia de usuario 6.

Historia de Usuario	
Número: 6	Usuario: Técnico de calidad.
Nombre: Calcular indicadores de QoS.	
Prioridad del negocio: Media.	Riesgo en desarrollo: Alto.
Puntos estimados: 4	Iteración asignada: 2
Programador responsable: Vladimir Alvarez Pérez.	
<p>Descripción: Luego de haber capturado los paquetes de red y los cuadros estos pueden ser usados para calcular los indicadores de la QoS.</p>	
Observaciones:	

Tabla 9 Historia de usuario 7.

Historia de Usuario	
Número: 7	Usuario: Técnico de calidad.
Nombre: Graficar indicadores de QoS.	
Prioridad del negocio: Media.	Riesgo en desarrollo: Medio.
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Vladimir Alvarez Pérez.	
<p>Descripción: Los valores de los diferentes indicadores calculados a medida que se van recibiendo los paquetes, son graficados en tiempo real.</p>	
Observaciones: Ver anexo 4.	

Tabla 10 Historia de usuario 8.

Historia de Usuario	
Número: 8	Usuario: Técnico de calidad.
Nombre: Correlacionar indicadores de QoS.	
Prioridad del negocio: Baja.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Vladimir Alvarez Pérez.	
<p>Descripción: Después de calcular los indicadores de QoS, se relacionan, a través del modelo</p>	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA



planteado, calculando cada valor de los parámetros del modelo y proporcionando como salida un valor numérico que representa el comportamiento de los servicios en una escala 0 – 100.

Observaciones:

Ver anexo 5.

Tabla 11 Historia de usuario 9.

Historia de Usuario	
Número: 9	Usuario: Técnico de calidad.
Nombre: Medir QoE global.	
Prioridad del negocio: Baja.	Riesgo en desarrollo: Alto.
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Vladimir Alvarez Pérez.	
Descripción: Proporciona una calificación del comportamiento de los servicios que se está monitorizando aplicando a la salida del modelo descrito en 2.3 el que se explica en 2.4. La calificación puede ser Excelente, Alta, Media, Baja, Mala o Impermissible.	
Observaciones: Ver anexo 5	

Tabla 12 Historia de usuario 10.

Historia de Usuario	
Número: 10	Usuario: Técnico de calidad.
Nombre: Guardar y cargar pruebas.	
Prioridad del negocio: Baja.	Riesgo en desarrollo: Alto.
Puntos estimados: 3	Iteración asignada: 5
Programador responsable: Vladimir Alvarez Pérez.	
Descripción: Permite guardar una prueba recién concluida y cargar una existente en la base de datos.	
Observaciones: Ver anexo 6	

Las historias de usuarios están basadas en los requerimientos a cumplir. Los mismos son listados a continuación.

2.6 Requisitos

Los requisitos de software son condiciones, propiedades y restricciones que se deben cumplir para garantizar el correcto funcionamiento del software y por ende, la satisfacción de las necesidades que le dan origen. Son fundamentales para el éxito del producto.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Estos pueden ser funcionales o no funcionales. A continuación se listan los requisitos de la herramienta desarrollada.

2.6.1 Requisitos funcionales

Los requisitos funcionales están enfocados a las funcionalidades y el comportamiento interno de la herramienta. Los requisitos funcionales de la herramienta son:

- RF 1. Conectarse con el servidor streaming.
- RF 2. Consumir servicios streaming.
- RF 3. Mostrar información de la red.
- RF 4. Capturar paquetes de la red.
- RF 5. Mostrar paquetes de la red.
- RF 6. Calcular indicadores de QoS.
- RF 7. Graficar indicadores de QoS.
- RF 8. Correlacionar indicadores de QoS.
- RF 9. Medir QoE global.
- RF 10. Guardar y cargar pruebas.

2.6.2 Requisitos no funcionales

Los requisitos no funcionales se basan en las propiedades y condiciones necesarias para el uso de la herramienta, así como en el entorno en que se ejecuta. Dichos requisitos son:

- Software
Es necesario el sistema operativo Windows XP Service Pack 3 o superior y el Framework .NET 4.0.
- Hardware
Se requiere tener instalado un mínimo de 1 GB de RAM y un procesador Pentium 4 con una velocidad de 2.0 GHz.
- Interfaz
La herramienta debe poseer una interfaz visual amigable y sencilla de forma tal que proporcione un fácil de entendimiento de las funcionalidades que realiza.
- Rendimiento
La herramienta debe trabajar de manera rápida ofreciendo un tiempo de respuesta mínimo para agilizar el procesamiento de la información.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Como guía para la implementación de la herramienta se diseñaron las tarjetas CRC (clases–responsabilidades–colaboradores). Las mismas se muestran en el próximo epígrafe.

2.7 Tarjetas CRC

Las tarjetas CRC constituyen una técnica para la representación de sistemas orientados a objetos. Identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software. Permiten al programador centrarse en el desarrollo evitando las malas prácticas de la programación. (39)

Tabla 13 Tarjeta CRC de la clase “RtspClient”.

Tarjeta CRC	
Clase RtspClient	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Es la clase encargada de administrar la conexión con el servidor streaming a través del envío de peticiones y el recibo de respuestas, también controla los eventos relacionados con la captura de los paquetes y el recibo de los cuadros. Almacena información sobre el estado de la conexión, el tipo de protocolo, dirección del servidor (incluye IP y puerto) y del archivo a reproducir (incluye extensión del archivo). La cantidad de bytes enviados y recibidos también es controlada por “RTSPClient”. <pre style="font-family: monospace; font-size: 0.9em;"> bool m_ForcedProtocol ClientProtocolType m_RtspProtocol RtspResponse m_LastRtspResponse AuthenticationSchemes m_AuthenticationScheme Uri m_Location byte[] m_Buffer IPAddress m_RemoteIP EndPoint m_RemoteRtsp Socket m_RtspSocket ProtocolType m_RtpProtocol RtspMethod m_LastMethod SessionDescription m_SessionDescription int m_SentBytes int m_RecievedBytes int m_RtspTimeoutSeconds int m_RtspPort int m_CSeq int m_ProtocolSwitchSeconds int m_RetryCount List<RtspMethod> m_SupportedMethods string m_UserAgent </pre>	<pre style="font-family: monospace; font-size: 0.9em;"> ClientProtocolType RtspMethod RtpClient MediaDescription RtspRequest RtspResponse SessionDescription </pre>

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<pre> RtpClient m_RtpClient Timer m_KeepAliveTimer Timer m_ProtocolSwitchTimer bool m_Live TimeSpan m_StartTime TimeSpan m_EndTime OnConnected() Requested(RtspRequest request) Received(RtspRequest request, RtspResponse response) OnDisconnected() Playing() Stopping(MediaDescription mediaDescription = null) RtspClient_OnDisconnect(RtspClient sender, object args) m_RtpClient_InterleavedData(RtpClient sender, ArraySegment<byte> slice) StartListening(TimeSpan? starttime = null) StopListening() Connect() Disconnect() SendRtspRequest(RtspRequest request) SendOptions() SendDescribe() SendTeardown(MediaDescription mediaDescription = null) SendSetup(MediaDescription mediaDescription) SendSetup(Uri location, MediaDescription mediaDescription, bool useMediaProtocol = true) SwitchProtocols(object state = null) SendPlay(Uri location = null, TimeSpan? startTime = null, TimeSpan? endTime = null, string rangeType = "ntp", string rangeFormat = null) SendKeepAlive(object state) SendGetParameter(string body = null) </pre>	
---	--

Tabla 14 Tarjeta CRC de la clase "RtpClient".

Tarjeta CRC	
Clase RtpClient	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Es usada por la clase "RtspClient" para comenzar a recibir los paquetes RTP, controla los eventos relacionados con el recibo de los paquetes. <pre> byte[] WakeUpBytes byte MAGIC uint RTP_SEQ_MOD int MAX_DROPOUT int MAX_MISORDER </pre>	<pre> RtpPacket RtpFrame ReceiversReport SendersReport SourceDescription MediaDescription </pre>

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

int MIN_SEQUENTIAL Interleaved(Socket existing) Sender(IP Address remoteAddress) Receiever(IP Address remoteAddress)	
---	--

Tabla 15 Tarjeta CRC de la clase "RtcpPacket".

Tarjeta CRC	
Clase RtcpPacket	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> Es la representación de los paquetes RTCP que viajan por la red, usada por la clase "RtspCliente" y por algunas interfaces, para el cálculo de los indicadores de calidad de servicio y mostrar la información al usuario respectivamente. int m_Version int m_Padding, m_Count short m_Length byte m_Type byte[] m_Data IsKnownPacketType(byte suspect) GetPackets(ArraySegment<byte> bufferReference) GetPackets(ArraySegment<byte> bufferReference, ref int offset) GetPackets(byte[] buffer, int offset = 0) ToBytes()	RtcpPacketType

Tabla 16 Tarjeta CRC de la clase "SessionDescription".

Tarjeta CRC	
Clase SessionDescription	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> Es usada por la clase "RtspClient" para almacenar la información correspondiente a la sesión activa. Incluye la descripción del medio a reproducir, la descripción del tiempo de la reproducción y demás elementos acordes con (40). string CR string LF string CRLF List<MediaDescription> m_MediaDescriptions List<TimeDescription> m_TimeDescriptions List<SessionDescriptionLine> m_Lines string CleanValue(string value) Add(MediaDescription mediaDescription, bool updateVersion = true) Add(TimeDescription timeDescription, bool updateVersion = true)	MediaType SessionVersionLine SessionOriginatorLine SessionNameLine MediaDescription SessionDescriptionLine

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Add(SessionDescriptionLine line, bool updateVersion = true) RemoveLine(int index, bool updateVersion = true) RemoveMediaDescription(int index, bool updateVersion = true) RemoveTimeDescription(int index, bool updateVersion = true)	
--	--

Tabla 17 Tarjeta CRC de la clase "QoErepote".

Tarjeta CRC	
Clase QoErepote	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Es la clase encargada de la aplicación del modelo que se propone para la medición de la QoE. Recibe los indicadores de calidad de servicio y la información de la red. <p> double R double R0 double Dr double Dpp double Dab double A InformacionRed net double Jitter double demora double demoraInicioReproduccion int cantidadPaquetesEsperados uint cantidadPaquetesPerdido double AnchoDevanda QoE MOS string QoELetras GetA() calcularQoE() CalcularR() CalcularDr() CalcularDpp() Calcularab() </p>	InformacionRed QoE

Tabla 18 Tarjeta CRC de la clase "InformacionRed".

Tarjeta CRC	
Clase InformacionRed	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Tiene la responsabilidad de obtener la información de la red. <p> NetworkInterface[] adaptadores IPAddress AnyCastAdress NetworkInterfaceType @Type </p>	NetworkInterface NetworkInterfaceType

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

long speed long Recieved long Send long SendSpeed long RecieveSpeed string urlAprovar	
--	--

Tabla 19 Tarjeta CRC de la clase "ManejadorSqlite".

Tarjeta CRC	
Clase ManejadorSqlite	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> Es la clase de la capa de acceso a datos encargada de manejar las conexiones a la información persistente. string direccionBD string StringConexion string password string user SQLiteCommand comand SQLiteConnection m_dbConnection CrearBD(string pathNombre) ConnetarBD() SimpleQuery(string query) CerrarConexion() DevolverEvaluacionesDeUnaPrueba(int id) DevolverPaquetesDeUnaPrueba(int id) DevolverDatosQuery(string query) GuardarPruebas(ref List<prueba> pruebas, List<paquete> paquetes, List<QoErepote> EvaluacionesQoE) CargarPruebas()	QoErepote Paquete prueba

Tabla 20 Tarjeta CRC de la clase "paquete".

Tarjeta CRC	
Clase paquete	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> Es la clase perteneciente al modelo de datos encargada de almacenar un paquete de forma codificada en bytes. List<byte> bytes int orden tipoPaquete tipopaquete int id_prueba int paquete_id long creado	tipoPaquete

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA



Tabla 21 Tarjeta CRC de la clase “prueba”.

Tarjeta CRC	
Clase prueba	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> • Es la clase encargada de representar una “prueba”, pertenece a la capa de acceso a datos. <p>int id_prueba string fecha</p>	

Para una mayor organización y optimización del tiempo se trazó un plan de iteraciones para la obtención de pequeñas versiones de la solución. El mismo se muestra a continuación.

2.8 Plan de iteraciones

Tabla 22 Plan de iteraciones.

Iteración	Objetivo	Historia de usuario	Duración
1	Dar cumplimiento a las historia de usuarios enfocadas a la conexión y consumo del servicio para sentar las bases de la herramienta.	HU1, HU2, HU3, HU4, HU5	11 semanas
2	Dar cumplimiento a la historia de usuario relacionada con el cálculo de los indicadores de calidad de servicio.	HU6	4 semana
3	Dar cumplimiento a la historia de usuario relacionada con las gráficas de los indicadores de calidad de servicio.	HU7	2 semana
4	Dar cumplimiento a las historias de usuario relacionadas a la correlación de los indicadores de calidad y la medición de la calidad de experiencia.	HU8, HU9	4 semanas
5	Dar cumplimiento a las historias de usuarios Guardar y cargar pruebas	HU10	3 semanas
Total			24 semanas

Con el objetivo de organizar la estructura de la herramienta se utiliza una arquitectura de software. Sobre ello se aborda en el siguiente epígrafe.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.9 Arquitectura

La Arquitectura de Software es la organización fundamental de un sistema. Se basa en sus componentes, las relaciones entre ellos y el ambiente, así como los principios que rigen su diseño y evolución. Permite a los desarrolladores compartir la misma línea de trabajo y cubrir todos los objetivos y restricciones. (41)

Para el desarrollo de la herramienta se utilizó la arquitectura N-Capas, dividida en 3 capas una para la presentación que es la encargada de la creación de la interfaz visual con la que interactúa el usuario, una para el control del negocio que es la que se encarga de la realización de los algoritmos complejos como los relacionados con la captura codificación y decodificación de los paquetes de red y una última capa para el acceso a la información persistente. La figura 3 muestra la arquitectura descrita.

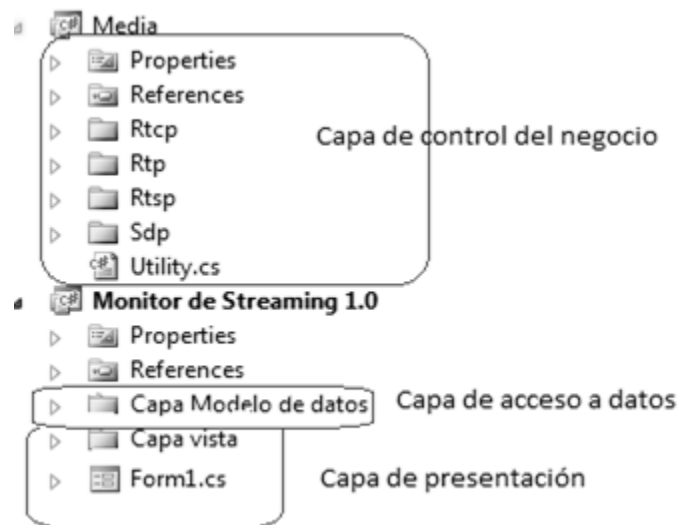


Figura 3 Arquitectura N-Capas dividida en 3 capas.

Mediante dicha arquitectura se garantiza un control centralizado de los accesos, recursos e integridad de los datos. Además, las funciones y responsabilidades son distribuidas, lo que facilita el mantenimiento. Teniendo en cuenta que a cada capa deben asignárseles las funcionalidades que le corresponden, se hace necesario el uso de patrones de diseño de software. A continuación se describen los usados en la solución.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.9.1 Patrones de diseño

Los patrones ofrecen soluciones a problemas de ingeniería de software. Proporcionan una descripción de los elementos y el tipo de relación que existe entre los mismos, además de un conjunto de restricciones sobre cómo pueden ser usados. Expresan esquemas de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. (42)

Para la elaboración de la herramienta serán usados los patrones GRASP debido a que describen los principios fundamentales de la asignación de responsabilidades a objetos y proporciona un diseño eficaz del software. (43)

A continuación se explica la selección de los patrones usados.

Experto: es el principio básico de asignación de responsabilidades. Indica qué clase es la responsable de la implementación de métodos según la información que maneja. Con ello se garantiza el encapsulamiento de la información y se obtiene un sistema robusto y fácil de mantener.

Uso del patrón: en el caso particular de la herramienta la clase “InformaciónRed” es la encargada de obtener todos los datos de la red, como la velocidad de transmisión, el tipo de red, la dirección IP entre otros.

Creador: en los sistemas programados bajo el paradigma orientado a objetos, la creación de instancias de clases es una de las actividades más frecuentes. En consecuencia conviene contar con un principio de asignación de responsabilidades para la creación de objetos. El patrón creador guía la asignación de responsabilidades, proporciona una mayor claridad, encapsulamiento y reutilización.

Uso del patrón: la clase “QoEresporte” se encarga crear una instancia de la clase “InformacionRed” que es la que tiene a su vez la información necesaria para obtener parámetros de red necesarios para la clase “QoEreporte”.

Bajo Acoplamiento: es el encargado de disminuir la dependencia de una clase con las demás. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño, estimulando asignar una responsabilidad de modo que su colocación no incremente el acoplamiento. Soporta el diseño de clases más independientes, que

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad.

Uso del patrón: en la capa del Modelo donde se encuentran las clases que implementan el acceso a datos, dichas clases tienen pocas asociaciones con otras de la vista o el negocio por lo que la dependencia en este caso es baja, poniéndose de manifiesto el patrón Bajo Acoplamiento.

Para garantizar un buen funcionamiento de la herramienta es necesario desplegarla de manera correcta. En el siguiente epígrafe se muestra cómo se realiza.

2.9.2 Diagrama de despliegue

Los diagramas de despliegue se utilizan para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Muestran la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software. Describen la arquitectura física del sistema durante la ejecución, (44). Poseen tres tipos de componentes:

- Nodos
Describen los elementos de procesamiento con al menos un procesador, memoria y cualquier otro dispositivo.
- Dispositivos
Son nodos que tienen capacidad de procesamiento en el nivel de abstracción que se modela.
- Conectores
Expresan el tipo de protocolo utilizado en el resto de los elementos del modelo.

La figura 4 muestra el diagrama de despliegue correspondiente a la herramienta desarrollada.

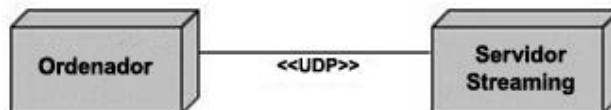


Figura 4 Diagrama de despliegue.

La herramienta posee diferentes componentes que interactúan entre sí, como se explica próximamente.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.9.3 Diagrama de componentes

Los diagramas de componentes se utilizan para representar cómo son divididos los sistemas de software en sus componentes y mostrar las dependencias entre ellos. Pueden ser usados para modelar vistas estáticas y dinámicas de un sistema. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables o paquetes. (44)

La herramienta está constituida por los componentes Controlador, encargado de la gestión y control de las funciones; Interfaz, para la interacción con el usuario; Modelo de Datos, para la representación y organización de los datos a persistir; y Librería, un conjunto de componentes encargados de la conexión con el servidor. Dentro de la Librería el componente RTSP es el encargado de manejar todas las peticiones y respuestas desde y hacia el servidor; el elemento RTCP es el encargado de controlar el tráfico de los paquetes RTCP; mientras que RTP y UDP son los encargados de controlar el flujo de datos desde el servidor. La figura 5 muestra el diagrama de componentes correspondiente a la herramienta.

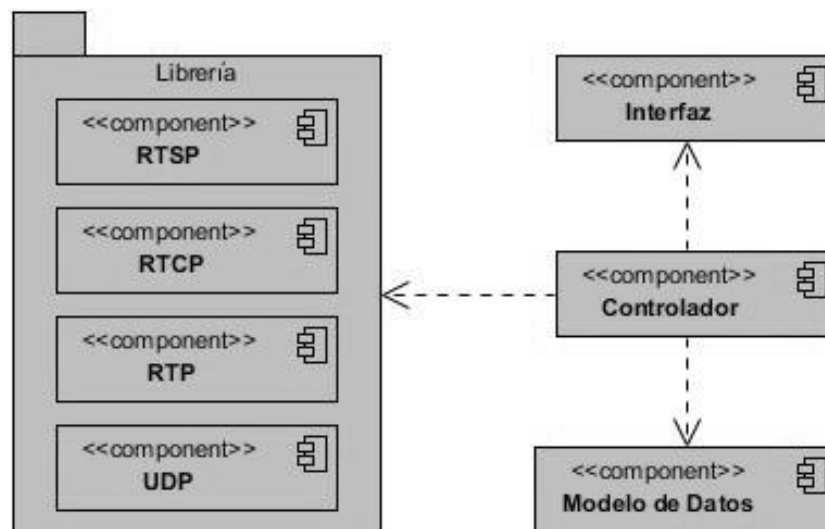


Figura 5 Diagrama de componentes.

Debido a que la herramienta almacena información, se requiere una base de datos de la cual se muestra su diseño en el próximo epígrafe.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.9.4 Diseño de base de datos

Una base de datos es una entidad en la que se puede almacenar de manera estructurada y persistente, un conjunto de datos pertenecientes a un mismo contexto para su posterior uso, (45). Entre sus objetivos se encuentran: el acceso eficiente a la información con redundancia mínima, el diseño de esquemas normalizados y la especificación de limitantes o dependencias funcionales, (46).

Para el almacenamiento de la información persistente a utilizar por la herramienta se seleccionó SQLite en su versión 3.0.

La base datos está compuesta por las clases “QoS”, “prueba” y “paquete”, donde cada prueba contiene un conjunto de paquetes e indicadores de calidad de servicio. La figura 6 muestra el diagrama de clases persistente.

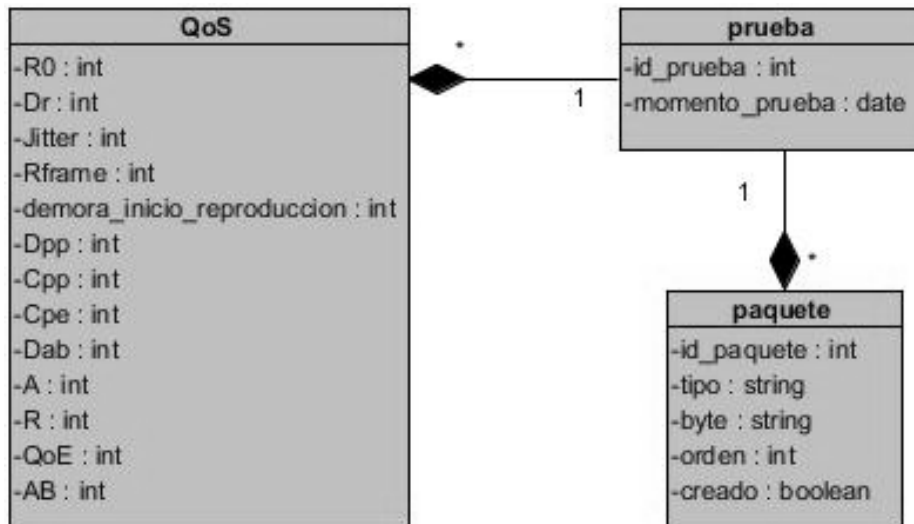


Figura 6 Diagrama de clases persistente.

Luego de definir las clases persistentes se procedió a elaborar el modelo de datos para el almacenamiento de la información que a continuación se presenta.

2.9.4.1 Modelo de datos

Las clases “QoS”, “prueba” y “paquete” fueron modeladas siguiendo los principios de la programación orientada a objetos para facilitar el trabajo con las mismas desde la aplicación. La siguiente figura muestra el modelo de datos realizado.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

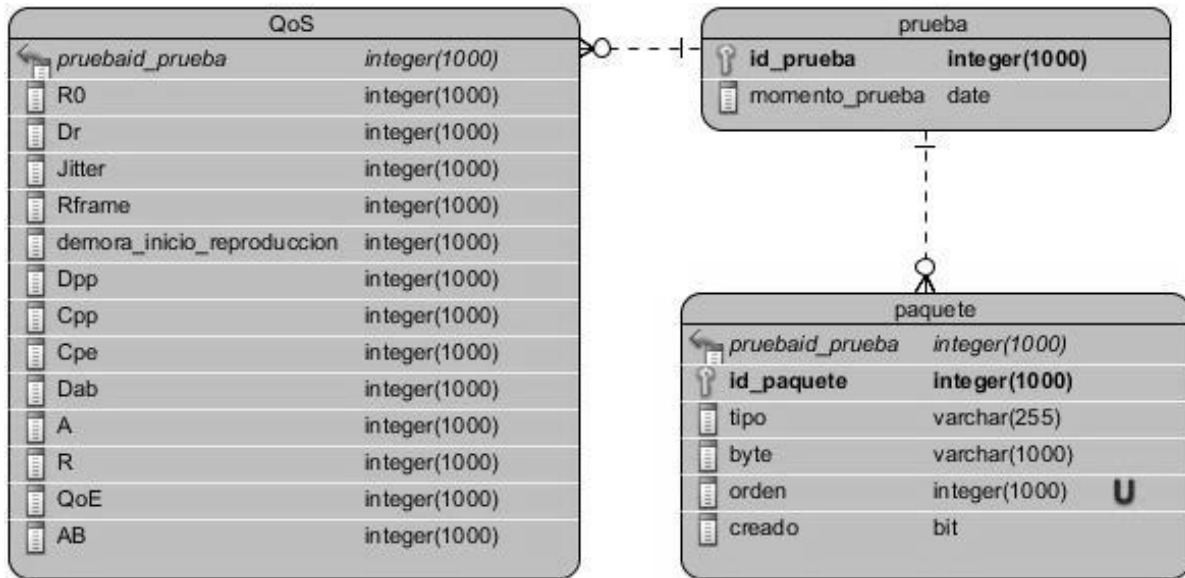


Figura 7 Modelo de datos.

Con la concluida elaboración del modelo de datos se puede dar paso a la estructuración de la aplicación descrita en el siguiente epígrafe.

2.9.5 Estructura de la herramienta

La herramienta está estructurada de forma sencilla y descriptiva para facilitar la interacción de los usuarios con la misma. El área de trabajo está compuesta por dos paneles y un menú que brinda las opciones a realizar como guardar prueba y seleccionar idioma. El panel de la izquierda presenta en su parte superior un reproductor de videos para la reproducción de los archivos consumidos, como se puede apreciar en la figura 8. Debajo muestra los datos de la conexión al servidor (ver figura 9), por ejemplo, tipo de red, ancho de banda, velocidad de transmisión, entre otros. El otro panel contiene un grupo de pestañas donde se muestran: las gráficas del comportamiento de los indicadores de calidad del servicio consumido (jitter, demora en el arribo de los cuadros, caudal instantáneo, paquetes perdidos), los paquetes de red recibidos por tipo de paquete (RTSP, RTP, RTCP), los paquetes de red en orden de arribo, los cuadros recibidos, las pruebas previas almacenadas en la base de datos y la calificación de la calidad obtenida (ver anexo 4, 5 y 6).

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

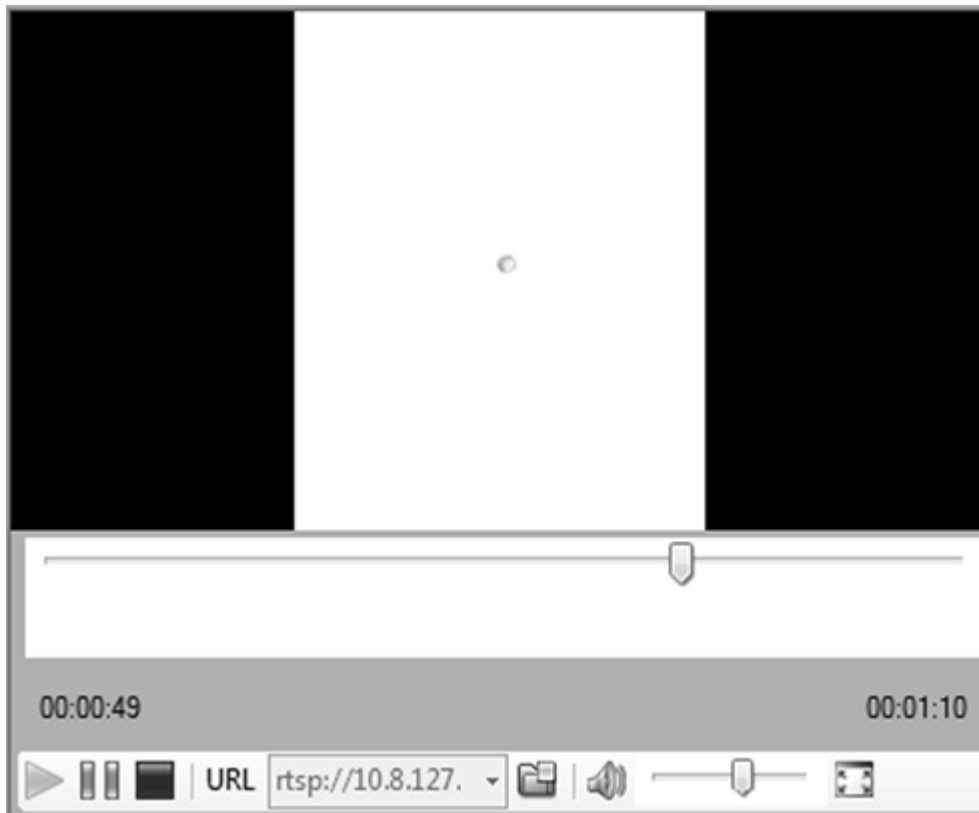


Figura 8 Consumo del servicio streaming.

Información de la red	
Tipo de red:	Ethernet
Dirección IP:	10.8.89.113
Velocidad de la red:	100 Mbps/s
Velocidad de envío:	4678 KB/s
Velocidad de recibo:	70039 KB/s
Ancho de banda consumido (Aplicación):	tipo
Ancho de banda consumido (Video):	tipo
Ancho de banda consumido (Audio):	tipo

Figura 9 Información de la red.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Conclusiones del capítulo

El capítulo que concluye, luego de la sistematización del estudio relacionado con los elementos teóricos que sustentan el desarrollo de una herramienta para la medición de la QoE en los servicios streaming, se logró definir los requisitos necesarios para la implementación. Se precisaron los indicadores a utilizar y la razón por la que se emplean. Se adaptaron los modelos a usar para el proceso de medición de la QoE. También se obtuvo las historias de usuarios como guías en el proceso de implementación y el plan de iteraciones que indica en qué momento se implementa qué requisito. Finalmente se describieron la arquitectura propuesta con los patrones de diseño de software a usar, los diagramas de despliegue, componentes, diseño, y el modelo de datos.

Lo mencionado anteriormente permite identificar como resultado del capítulo que se cuenta con los elementos para el desarrollo de la herramienta. Lo descrito evidencia el cumplimiento de las etapas de planificación y diseño de la metodología de desarrollo de software programación extrema. De ello se puede concluir que se cuenta con los conocimientos teóricos y metodológicos necesarios para comenzar las fases de codificación y pruebas.

Capítulo 3: Validación de la Solución

Introducción

En el presente capítulo se lleva a la práctica la solución desarrollada en escenarios reales. La herramienta fue sometida a pruebas para verificar su funcionamiento y así garantizar la calidad del software elaborado. Al final del capítulo se refleja el proceso de medición y monitorización de servicios streaming usando la herramienta implementada bajo un conjunto de condiciones particulares. Se comienza por las pruebas a las funcionalidades conocidas como pruebas unitarias.

3.1 Pruebas unitarias

Las pruebas unitarias se realizan a las funcionalidades para verificar que trabajen correctamente. Aseguran que cada una de las partes individuales que conforman la herramienta trabaje de forma adecuada. Las mismas deben ser automatizables, completas, reutilizables, independientes y profesionales para acotar más los errores y localizarlos más fácilmente. (35)

Para la realización de las pruebas unitarias se empleó como servidor una computadora con un procesador Intel a una velocidad de 3,0 GHz, 256 MB de RAM y un disco duro de 80 GB sobre el sistema operativo GNU/Linux Devian Lenny con núcleo 2.6.26-2-686. Como cliente se utilizó una computadora con un procesador AMD v120 a una velocidad de 2,2 GHz, 1 GB de RAM y un disco duro de 250 GB sobre el sistema operativo Windows 7.

A continuación se muestran las pruebas realizadas a las funcionalidades definidas.

Tabla 23 Prueba unitaria 1.

Prueba unitaria	
Código: 1	Nombre: Conectarse al servidor streaming.
Descripción: El usuario debe ingresar la dirección del audio o video a reproducir, la dirección debe poseer una estructura (dirección válida) correcta e incluir la extensión del archivo, por ejemplo rtsp://10.8.127.244/sample_100kbit.mp4. Luego se establece la conexión con el servidor streaming mediante el protocolo RTSP.	
Ejecutado por: Dainier Aguila Medina.	
Entrada: Dirección del audio/video ubicado en el servidor.	

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN



Resultados: parcialmente satisfactoria.
Observaciones: Fueron realizadas 5 pruebas de las cuales 1 resultó fallida (ver anexo 7).

Tabla 24 Prueba unitaria 2.

Prueba unitaria	
Código: 2	Nombre: Consumir servicios streaming.
Descripción: Luego de establecer la conexión con el servidor, una petición de reproducir es enviada al servidor para comenzar a consumir el servicio. También pueden ser enviadas peticiones de pausa, stop, correr hacia delante o hacia atrás.	
Ejecutado por: Dainier Aguila Medina.	
Entrada: Dirección del audio/video ubicado en el servidor.	
Resultados: satisfactoria.	

Tabla 25 Prueba unitaria 3.

Prueba unitaria	
Código: 3	Nombre: Mostrar información de la red.
Descripción: Captura y muestra la información de la red relacionada con el tipo de red, la dirección IP, la velocidad de trasmisión y el ancho de banda negociado entre la aplicación cliente y el servidor. La información es en tiempo real.	
Ejecutado por: Dainier Aguila Medina.	
Entrada: No hay entrada de datos.	
Resultados: satisfactoria.	

Tabla 26 Prueba unitaria 4.

Prueba unitaria	
Código: 4	Nombre: Capturar paquetes de red.
Descripción: La comunicación con el servidor streaming se realiza a través de paquetes de tipo RTSP, RTP y RTCP, los cuales viajan codificados, durante el proceso deben capturarse y decodificarse para poder ser analizados posteriormente. Cuando se completa un cuadro también debe ser capturado y decodificado.	
Ejecutado por: Dainier Aguila Medina.	
Entrada: Dirección del audio/video ubicado en el servidor.	

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

Resultados: satisfactoria.

Tabla 27 Prueba unitaria 5.

Prueba unitaria	
Código: 5	Nombre: Mostrar paquetes de red.
Descripción: Una vez capturados y decodificados los paquetes de red estos son mostrados al usuario con sus correspondientes valores. Son visualizados por tipo de paquete y en orden de arribos.	
Ejecutado por: Dainier Aguila Medina.	
Entrada: Dirección del audio/video ubicado en el servidor.	
Resultados: satisfactoria.	

Tabla 28 Prueba unitaria 6.

Prueba unitaria	
Código: 6	Nombre: Calcular indicadores de QoS.
Descripción: Luego de haber capturado los paquetes de red y los cuadros, estos son usados para calcular los indicadores de la QoS.	
Ejecutado por: Dainier Aguila Medina.	
Entrada: Dirección del audio/video ubicado en el servidor.	
Resultados: satisfactoria.	

Tabla 29 Prueba unitaria 7.

Prueba unitaria	
Código: 7	Nombre: Graficar indicadores de QoS.
Descripción: Con los valores de los diferentes indicadores calculados a medida que se van recibiendo los paquetes, estos son graficados en tiempo real.	
Ejecutado por: Dainier Aguila Medina.	
Entrada: Dirección del audio/video ubicado en el servidor.	
Resultados: satisfactoria.	

Tabla 30 Prueba unitaria 8.

Prueba unitaria	
Código: 8	Nombre: Correlacionar indicadores de QoS.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

Descripción: Después de calcular los indicadores de QoS, se relacionan, a través del modelo planteado, calculando cada valor de los parámetros del modelo y proporcionando como salida un valor numérico que representa el comportamiento de los servicios en una escala 0-100.
Ejecutado por: Dainier Aguila Medina.
Entrada: Dirección del audio/video ubicado en el servidor.
Resultados: satisfactoria.

Tabla 31 Prueba unitaria 9.

Prueba unitaria	
Código: 9	Nombre: Medir QoE global.
Descripción: Proporciona una calificación del comportamiento de los servicios que se está monitorizando aplicando a la salida del modelo descrito en 2.3 el que se explica en 2.4. La calificación puede ser Excelente, Alta, Media, Baja, Mala o Impermisible.	
Ejecutado por: Dainier Aguila Medina.	
Entrada: Dirección del audio/video ubicado en el servidor.	
Resultados: satisfactoria.	

Tabla 32 Prueba unitaria 10.

Prueba unitaria	
Código: 10	Nombre: Guardar y cargar pruebas.
Descripción: Permite guardar una prueba recién concluida y cargar una existente en la base de datos.	
Ejecutado por: Dainier Aguila Medina.	
Entrada: Dirección del audio/video ubicado en el servidor.	
Resultados: satisfactoria.	

3.2 Pruebas de aceptación

Las pruebas de aceptación son un conjunto de condiciones o variables bajo las cuales se determina si los requisitos de la herramienta se cumplen parcial o totalmente. Son realizadas a cada una de las historias de usuario para garantizar un correcto funcionamiento. (35)

A continuación se muestran los casos de pruebas de aceptación realizados.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN



Tabla 33 Caso de prueba de aceptación 1.

Caso de Prueba de Aceptación	
Código: 1	Historia de usuario: Conectarse al servidor streaming.
Nombre del probador: Dainier Aguila Medina.	
Descripción: El usuario debe ingresar la dirección del audio o video a reproducir, la dirección debe poseer una estructura (dirección válida) correcta e incluir la extensión del archivo, por ejemplo rtsp://10.8.127.244/sample_100kbit.mp4. Luego se establece la conexión con el servidor streaming mediante el protocolo RTSP.	
Condiciones de ejecución: <ul style="list-style-type: none">• Conexión de red establecida.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">• Se ejecuta la aplicación.	
Resultado esperado: correcta conexión con el servidor.	
Evaluación de la prueba: satisfactoria.	

Tabla 34 Caso de prueba de aceptación 2.

Caso de Prueba de Aceptación	
Código: 2	Historia de usuario: Consumir servicios streaming.
Nombre del probador: Dainier Aguila Medina.	
Descripción: Luego de establecer la conexión con el servidor, una petición de reproducir es enviada al servidor para comenzar a consumir el servicio. También pueden ser enviadas peticiones de pausa, stop, correr hacia delante o hacia atrás.	
Condiciones de ejecución: <ul style="list-style-type: none">• Conexión de la red establecida.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">• Se ejecuta la aplicación.• Se reproduce el audio/video.	
Resultado esperado: reproducción del audio/video del servidor.	
Evaluación de la prueba: satisfactoria.	

Tabla 35 Caso de prueba de aceptación 3.

Caso de Prueba de Aceptación	
Código: 3	Historia de usuario: Mostrar información de la red.
Nombre del probador: Dainier Aguila Medina.	
Descripción: Captura y muestra la información de la red relacionada con el tipo de red, la dirección IP, la velocidad de transmisión y el ancho de banda negociado entre la aplicación cliente y el servidor. La información es en tiempo real.	
Condiciones de ejecución: <ul style="list-style-type: none">• Conexión de red establecida.	

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

Entrada / Pasos de ejecución:
<ul style="list-style-type: none"> • Se ejecuta la aplicación.
Resultado esperado: visualización de la información de la red.
Evaluación de la prueba: satisfactoria.

Tabla 36 Caso de prueba de aceptación 4.

Caso de Prueba de Aceptación	
Código: 4	Historia de usuario: Capturar paquetes de red.
Nombre del probador: Dainier Aguila Medina.	
Descripción: La comunicación con el servidor streaming se realiza a través de paquetes de tipo RTSP, RTP y RTCP , los cuales viajan codificados, durante el proceso deben capturarse y decodificarse para poder ser analizados posteriormente. Cuando se completa un cuadro también debe ser capturado y decodificado.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Conexión de red establecida. 	
Entrada / Pasos de ejecución:	
<ul style="list-style-type: none"> • Se ejecuta la aplicación. • Se reproduce el audio/video. 	
Resultado esperado: recepción de los paquetes de red.	
Evaluación de la prueba: satisfactoria.	

Tabla 37 Caso de prueba de aceptación 5.

Caso de Prueba de Aceptación	
Código: 5	Historia de usuario: Mostrar paquetes de red.
Nombre del probador: Dainier Aguila Medina.	
Descripción: Una vez capturados y decodificados los paquetes de red estos son mostrados al usuario con sus correspondientes valores. Son visualizados por tipo de paquete y en orden de arribos.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Conexión de red establecida. 	
Entrada / Pasos de ejecución:	
<ul style="list-style-type: none"> • Se ejecuta la aplicación. • Se reproduce el audio/video. 	
Resultado esperado: visualización de los paquetes de red recibidos.	
Evaluación de la prueba: satisfactoria.	

Tabla 38 Caso de prueba de aceptación 6.

Caso de Prueba de Aceptación

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

Código: 6	Historia de usuario: Calcular indicadores de QoS.
Nombre del probador: Dainier Aguila Medina.	
Descripción: Luego de haber capturado los paquetes de red y los cuadros estos pueden ser usados para calcular los indicadores de la QoS.	
Condiciones de ejecución: <ul style="list-style-type: none"> • Conexión de red establecida. 	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Se ejecuta la aplicación. • Se reproduce el audio/video. 	
Resultado esperado: valor de los indicadores de calidad de servicio.	
Evaluación de la prueba: satisfactoria.	

Tabla 39 Caso de prueba de aceptación 7.

Caso de Prueba de Aceptación	
Código: 7	Historia de usuario: Graficar indicadores de QoS.
Nombre del probador: Dainier Aguila Medina.	
Descripción: Con los valores de los diferentes indicadores calculados a medida que se van recibiendo los paquetes estos son graficados en tiempo real.	
Condiciones de ejecución: <ul style="list-style-type: none"> • Conexión de red establecida. 	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Se ejecuta la aplicación. • Se reproduce el audio/video. 	
Resultado esperado: comportamiento graficado de los indicadores de QoS.	
Evaluación de la prueba: satisfactoria.	

Tabla 40 Caso de prueba de aceptación 8.

Caso de Prueba de Aceptación	
Código: 8	Historia de usuario: Correlacionar indicadores de QoS.
Nombre del probador: Dainier Aguila Medina.	
Descripción: Después de calcular los indicadores de QoS, se relacionan, a través del modelo planteado, calculando cada valor de los parámetros del modelo y proporcionando como salida una valor numérico que representa el comportamiento de los servicios en una escala 0–100.	
Condiciones de ejecución: <ul style="list-style-type: none"> • Conexión de red establecida. 	
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Se ejecuta la aplicación. • Se reproduce el audio/video. 	

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

Resultado esperado: cualificación de los indicadores de QoS.
Evaluación de la prueba: satisfactoria.

Tabla 41 Caso de prueba de aceptación 9.

Caso de Prueba de Aceptación	
Código: 9	Historia de usuario: Medir QoE global.
Nombre del probador: Dainier Aguila Medina.	
Descripción: Proporciona una calificación del comportamiento de los servicios que se está monitorizando aplicando a la salida del modelo descrito en 2.3 el que se explica en 2.4. La calificación puede ser Excelente, Alta, Media, Baja, Mala o Impermissible.	
Condiciones de ejecución: <ul style="list-style-type: none">• Conexión de red establecida.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">• Se ejecuta la aplicación.	
Resultado esperado: calificación de la QoE.	
Evaluación de la prueba: satisfactoria.	

Tabla 42 Prueba de aceptación 10.

Caso de Prueba de Aceptación	
Código: 10	Historia de usuario: Guardar y cargar pruebas.
Nombre del probador: Dainier Aguila Medina.	
Descripción: Permite guardar una prueba recién concluida y cargar una existente en la base de datos.	
Condiciones de ejecución: <ul style="list-style-type: none">• Conexión de red establecida.	
Entrada / Pasos de ejecución: <ul style="list-style-type: none">• Se ejecuta la aplicación.	
Resultado esperado: correcto funcionamiento de guardado y cargado de pruebas.	
Evaluación de la prueba: parcialmente satisfactoria.	
Observaciones Fueron realizados 10 intentos para guardar y cargar las pruebas llevadas a cabo por la herramienta de las cuales una resultó fallida (ver anexo 7).	

Conclusiones del capítulo

El capítulo que concluye fue diseñado para la comprobación del buen funcionamiento de la herramienta y el cumplimiento de los requerimientos planteados por el cliente. Para lograrlo fueron aplicadas las pruebas unitarias y de aceptación. Los resultados obtenidos demuestran que fueron alcanzados satisfactoriamente los objetivos trazados.

Conclusiones generales

Como resultado de la investigación se obtuvo una herramienta que permite la monitorización continua de los servicios streaming sobre redes móviles entre extremos. Se identificaron indicadores de calidad del servicio necesarios para la monitorización. Se logró elaborar un modelo de medición de los servicios streaming, tomando como ejemplo el modelo E pero ajustándolo a los indicadores definidos. La salida del modelo es llevada a la escala de otro modelo conocido como MOS, el cual da una medida de la calidad percibida por los usuarios. La herramienta es capaz de consumir los servicios (reproducción de audio y video), capturar los paquetes que viajan por la red correspondientes a los servicios streaming (paquetes RTSP, RTP, RTCP) y mostrarlos con todos sus elementos en forma de árbol jerárquico. Captura los cuadros correspondientes al flujo consumido, dicho flujo es mostrado en el orden en que se van recibiendo los paquetes. Cada indicador es graficado según su comportamiento y variación en el tiempo. La QoE es medida en varios momentos de la reproducción. Además, grafica y muestra la salida de los modelos para cada prueba. La herramienta permite la monitorización, la identificación de posibles problemas entre los extremos, y la percepción de los servicios prestados por parte de los usuarios. Imágenes de la herramienta en funcionamiento pueden verse en los anexos.



Recomendaciones

Se recomienda incluir en versiones posteriores de la herramienta otros indicadores de calidad para conocer de forma más exacta el comportamiento del servicio prestado.

Referencias bibliográficas

1. **Venegas Morales, Mauricio Hernán, Yáñez Cañas, Aquiles y J. González, Agustín.** *Transmisión de video de alta calidad a través de redes IP.* Valparaíso : Casilla 110-V, 2005. Disponible en: <http://www.grc.upv.es/docencia/tdm/practic/P3.pdf>.
2. **Soldani, David.** *QoS and QoE Management in UMTS Cellular Systems.* s.l. : John Wiley & Sons, 2006. 0470016396.
3. *The Research of Quality of Experience Evaluation Method in Pervasive Computing Environment.* **Li-yuan, Liu.** Urumqi : s.n., 2006. págs. 178 - 182. 1-4244-0326-x.
4. **Pérez Agüera, José Ramón, Sánchez Jiménez, Rodrigo y Caldera Serrano, Jorge.** *ADAPTACIÓN DE TECNOLOGÍAS STREAM Y XML A CENTROS DE DOCUMENTACIÓN EN TELEVISIÓN.* 2004. Disponible en: http://redc.revistas.csic.es/index.php/redc/article/download/158/212&ei=2gbluly2nssarah8s4cicw&usg=afqjcngqrnbecjs_d-dpl-hploiz0c0rvq&bvm=bv.1355325884,d.cwe&cad=rja.
5. **Ortiz, Quintero, y otros.** *EVALUACION DE SERVIDORES DE STREAMING DE VIDEO ORIENTADO.* 2006.
6. **Prieto, Andres Felipe y Rodriguez, Luisa Fernanda.** *Streaming de audio a través de dispositivos móviles.* Bogotá : s.n., 2007. Disponible en: http://www.albertosantana.org/videosconvida/tareas/subidos_ver/Streaming%20de%20Audio%20en%20Dispositivos%20Moviles.pdf.
7. **APPLE COMPUTER. INC.** *Mac OS X QuickTime Streaming Server 5.0.* 2007.
8. **Deering, Stive.** *IP Multicast Extensions for 4.3BSD UNIX and related systems.* 2012. Disponible en: <http://andrew.triumf.ca/pub/mbone/ipmulticast.README>.
9. **Gómez, A. F., Mateo, A. L. y Ruíz, P.M.** *Arquitectura para el control de emisores en entornos multicast.* [En línea] 2007. [Citado el: 10 de diciembre de 2012.] <http://www.rediris.es/difusion/publicaciones/boletin/50-51/ponencia13.html>.
10. **Rijo Sciara, Daniel.** *Fundamentos de Video Streaming.* Montevideo : s.n., 2004. Vol. I, Disponible en: <http://iie.fing.edu.uy/ense/asign/codif/material/monografias/2004-01.pdf>. 0-13-017547-1.

REFERENCIAS BIBLIOGRÁFICAS

11. **María, Suyama.** Desarrollo Web. Protocolos de comunicación. [En línea] 2004. [Citado el: 15 de enero de 2013.] <http://www.desarrolloweb.com/articulos/1617.php>.
12. **Kioskea.** Kioskea. *Protocolos*. [En línea] 2008. [Citado el: 10 de enero de 2013.] <http://es.kioskea.net/contents/internet/protocol.php3>.
13. **Internet Engineering Task Force.** *RTSP RCF 3226*. 2001. Disponible en: www.ietf.org/rfc/rfc3226.txt.
14. **Schulzrinne, Henning, y otros.** IETF Tools. *RTP: A Transport Protocol for Real-Time Applications*. [En línea] julio de 2003. [Citado el: 20 de noviembre de 2012.] <http://tools.ietf.org/html/rfc3550>. rfc 3550.
15. **ITU-T.** *Recomendación QoS.02: Manual Calidad de servicio y rendimiento de la red*. 2004. Norma vigente.
16. **Rahrer, Tim, Fiandra, Riccardo y Wright, Steven.** *Triple-Play Services Quality of Experience*. 2006.
17. **ITU-T.** *Recomendación G1010: Categorías QoS para el usuario final en servicios multimedia*. 2001. Norma vigente.
18. —. *Recomendación P.800: Métodos de determinación subjetiva de la calidad de transmisión*. 1996. Norma vigente.
19. —. *Recomendación P.800.1: Terminología de las notas medias de opinión*. 2003. Norma vigente.
20. —. *Recomendación P.910: Subjective video quality assessment methods*. 2008. Norma vigente.
21. —. *Recomendación P.911: Métodos de evaluación subjetiva de la calidad*. 1999. Norma vigente.
22. —. *Recomendación P.920: Métodos de prueba interactivos para comunicaciones audiovisuales*. 2000. Norma vigente.
23. —. *Recomendación G.107: The E-model: a computational model for use in transmission planning*. 2011. Norma vigente.

REFERENCIAS BIBLIOGRÁFICAS

24. —. *Recomendación BS 1387: Evaluación perceptual de la calidad de audio*. Norma vigente.
25. —. *Recomendación J.144: Video Quality Model*. 2004. Norma vigente.
26. **Auza, Jun**. TeachSource. [En línea] 16 de diciembre de 2010. [Citado el: 26 de enero de 2013.] <http://www.junauza.com/2010/12/free-server-monitoring-software.html>.
27. **MComms TV**. MComms TV. [En línea] [Citado el: 18 de enero de 2013.] Compañía dedicada a la creación de software para el monitoreo de TV móvil, Level 1, 90 William Street email: sales@mcommstv.com . <http://www.mcommstv.com>.
28. **Microsoft Corporation**. MSDN. *Lenguaje Visual C#*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/aa287558%28v=VS.71%29.aspx>.
29. —. MSDN. *Objetos, clases y estructuras (c#)*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/ms173109%28v=VS.80%29.aspx>.
30. —. MSDN. *Información general y conceptual sobre .NET Framework*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/zw4w595w%28v=VS.80%29.aspx>.
31. —. MSDN. *Características de implementación de .NET Framework*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/410zh1ty%28v=VS.80%29.aspx>.
32. —. MSDN. *Introducción a Visual Studio*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/vstudio/fx6bk1f4%28v=vs.100%29.aspx>.
33. —. MSDN. *Microsoft Visual Studio 2010 Ultimate*. [En línea] [Citado el: 16 de enero de 2013.] <http://www.microsoft.com/es-es/download/details.aspx?id=12187>.
34. SQLite. [En línea] [Citado el: 5 de diciembre de 2012.] <http://sqlite.org/about.html>.
35. **Beck, Kent**. *Extreme Programming Explained: Embrace Change*. s.l.: Pearson Education, 1999. 978-0321278654.
36. *Visibility of individual packet losses in MPEG-2 video*. **Reibman, Amy R., y otros**. 2004. Vol. I, págs. 171-174. 0-7803-8554-3.

REFERENCIAS BIBLIOGRÁFICAS

37. *Delivery of MPEG video streams with constant perceptual quality of service*. **Quaglia, D. y De Martin, J.C.** 2002. Vol. II, págs. 85-88. 0-7803-7304-9.
38. *DCT Quantization Noise in Compressed Images*. **Robertson, M.A y Stevenson, R.L.** 1, s.l. : IEEE Transactions on Circuits and Systems for Video Technology, enero de 2005, Vol. XV, págs. 27-38. 1051-8215.
39. **Beck, Kent.** *Extreme Programming Explained: Embrace Change*. s.l. : Pearson Education, 1999. 978-0321278654.
40. **Internet Engineering Task Force.** *RTP RFC 3550*. 2003. Disponible en: www.ietf.org/rfc/rfc3550.txt.
41. **Gorton, Ian.** *Essential Software Architecture*. Segunda edición. New York : Springer, 2011. 978-3-642-19176-3.
42. **Buschmann, Frank, y otros.** *Pattern-Oriented Software Architecture: A System of Patterns*. s.l. : John Wiley & Sons, 1996. pág. 476. Vol. I. 978-0-471-95869-7.
43. **Grosso, Andrés.** *Prácticas de Software. Patrones GRASP*. [En línea] 21 de marzo de 2011. [Citado el: 25 de enero de 2013.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
44. **Ambler, Scott W.** *The Elements of UML 2.0 Style*. 2005. 0-521-61678-6.
45. **Kioskea.** *Kioskea. Introducción - Bases de datos*. [En línea] 2008. [Citado el: 7 de marzo de 2013.] <http://es.kioskea.net/contents/bdd/bddintro.php3>.
46. **Saldivar Vargas, Guillermo de Jesús.** *Monografías. Diseño de bases de datos*. [En línea] 12 de febrero de 2006. [Citado el: 7 de marzo de 2013.] <http://www.monografias.com/trabajos30/base-datos/base-datos.shtml>.

Bibliografía

Ambler, Scott W. *The Elements of UML 2.0 Style*. 2005. 0-521-61678-6.

APPLE COMPUTER. INC. *Mac OS X QuickTime Streaming Server 5.0*. 2007.

Auza, Jun. TeachSource. [En línea] 16 de diciembre de 2010. [Citado el: 26 de enero de 2013.] <http://www.junauza.com/2010/12/free-server-monitoring-software.html>.

Beck, Kent. *Extreme Programming Explained: Embrace Change*. s.l. : Pearson Education, 1999. 978-0321278654.

Beck, Kent. *Extreme Programming Explained: Embrace Change*. s.l. : Pearson Education, 1999. 978-0321278654.

Buschmann, Frank, y otros. *Pattern-Oriented Software Architecture: A System of Patterns*. s.l. : John Wiley & Sons, 1996. pág. 476. Vol. I. 978-0-471-95869-7.

Crispin, Mark. *Internet Message Access Protocol 4*. 2003. pág. 108.

DCT Quantization Noise in Compressed Images. **Robertson, M.A y Stevenson, R.L.** 1, s.l. : IEEE Transactions on Circuits and Systems for Video Technology, enero de 2005, Vol. XV, págs. 27-38. 1051-8215.

Deering, Stive. *IP Multicast Extensions for 4.3BSD UNIX and related systems*. 2012. Disponible en: <http://andrew.triumf.ca/pub/mbone/ipmulticast.README>.

Delivery of MPEG video streams with constant perceptual quality of service. **Quaglia, D. y De Martin, J.C.** 2002. Vol. II, págs. 85-88. 0-7803-7304-9.

Gómez, A. F., Mateo, A. L. y Ruíz, P.M. Arquitectura para el control de emisores en entornos multicast. [En línea] 2007. [Citado el: 10 de diciembre de 2012.] <http://www.rediris.es/difusion/publicaciones/boletin/50-51/ponencia13.html>.

Gorton, Ian. *Essential Software Architecture*. Segunda edición. New York : Springer, 2011. 978-3-642-19176-3.

Grosso, Andrés. Prácticas de Software. *Patrones GRASP*. [En línea] 21 de marzo de 2011. [Citado el: 25 de enero de 2013.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.

International Conference on Image Processing. Reibman, Amy R., y otros. 2004. Vol. I, págs. 171-174. 0-7803-8554-3.

Internet Engineering Task Force. *RTP RFC 3550.* 2003. Disponible en: www.ietf.org/rfc/rfc3550.txt.

Internet Engineering Task Force. *RTSP RCF 3226.* 2001. Disponible en: www.ietf.org/rfc/rfc3226.txt.

ITU-T. *Recomendación BS 1387: Evaluación perceptual de la calidad de audio.* Norma vigente.

ITU-T. *Recomendación G.107: The E-model: a computational model for use in transmission planning.* 2011. Norma vigente.

ITU-T. *Recomendación G1010: Categorías QoS para el usuario final en servicios multimedia.* 2001. Norma vigente.

ITU-T. *Recomendación P.800.1: Terminología de las notas medias de opinión.* 2003. Norma vigente.

ITU-T. *Recomendación P.800: Métodos de determinación subjetiva de la calidad de transmisión.* 1996. Norma vigente.

ITU-T. *Recomendación P.910: Subjective video quality assessment methods.* 2008. Norma vigente.

ITU-T. *Recomendación P.911: Métodos de evaluación subjetiva de la calidad.* 1999. Norma vigente.

ITU-T. *Recomendación P.920: Métodos de prueba interactivos para comunicaciones audiovisuales.* 2000. Norma vigente.

ITU-T. *Recomendación QoS.02: Manual Calidad de servicio y rendimiento de la red.* 2004. Norma vigente.

ITU-T. *Recomendación J.144: Video Quality Model.* 2004. Norma vigente.

Kioskea. Kioskea. *Introducción - Bases de datos.* [En línea] 2008. [Citado el: 7 de marzo de 2013.] <http://es.kioskea.net/contents/bdd/bddintro.php3>.

Kioskea. Kioskea. *Protocolos*. [En línea] 2008. [Citado el: 10 de enero de 2013.] <http://es.kioskea.net/contents/internet/protocol.php3>.

Klensin, J. *Simple Mail Transfer Protocol*. [ed.] AT&T Laboratories. 2001. ISTF; RFC 282.

María, Suyama. Desarrollo Web. Protocolos de comunicación. [En línea] 2004. [Citado el: 15 de enero de 2013.] <http://www.desarrolloweb.com/articulos/1617.php>.

MComms TV. MComms TV. [En línea] [Citado el: 18 de enero de 2013.] Compañía dedicada a la creación de software para el monitoreo de TV móvil, Level 1, 90 William Street email: sales@mcommstv.com . <http://www.mcommstv.com>.

Microsoft Corporation. MSDN. *Características de implementación de .NET Framework*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/410zh1ty%28v=VS.80%29.aspx>.

Microsoft Corporation. MSDN. *Información general y conceptual sobre .NET Framework*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/zw4w595w%28v=VS.80%29.aspx>.

Microsoft Corporation. MSDN. *Introducción a Visual Studio*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/vstudio/fx6bk1f4%28v=vs.100%29.aspx>.

Microsoft Corporation. MSDN. *Lenguaje Visual C#*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/aa287558%28v=VS.71%29.aspx>.

Microsoft Corporation. MSDN. *Microsoft Visual Studio 2010 Ultimate*. [En línea] [Citado el: 16 de enero de 2013.] <http://www.microsoft.com/es-es/download/details.aspx?id=12187>.

Microsoft Corporation. MSDN. *Objetos, clases y estructuras (c#)*. [En línea] [Citado el: 16 de enero de 2013.] <http://msdn.microsoft.com/es-es/library/ms173109%28v=VS.80%29.aspx>.

Ortiz, Quintero, y otros. *EVALUACION DE SERVIDORES DE STREAMING DE VIDEO ORIENTADO*. 2006.

Pérez Agüera, José Ramón, Sánchez Jiménez, Rodrigo y Caldera Serrano, Jorge. *ADAPTACIÓN DE TECNOLOGÍAS STREAM Y XML A CENTROS DE*

DOCUMENTACIÓN EN TELEVISIÓN. 2004. Disponible en: http://redc.revistas.csic.es/index.php/redc/article/download/158/212&ei=2gbluly2nssarah8s4cicw&usg=afqjcngqrnbecjs_d-dpl-hploiz0c0rv&bvm=bv.1355325884,d.cwe&cad=rja.

Prieto, Andres Felipe y Rodriguez, Luisa Fernanda. *Streaming de audio a través de dispositivos móviles*. Bogotá : s.n., 2007. Disponible en: http://www.albertosantana.org/videosconvida/tareas/subidos_ver/Streaming%20de%20Audio%20en%20Dispositivos%20Moviles.pdf.

Rahrer, Tim, Fiandra, Riccardo y Wright, Steven. *Triple-Play Services Quality of Experience*. 2006.

Rijo Sciara, Daniel. *Fundamentos de Video Streaming*. Montevideo : s.n., 2004. Vol. I, Disponible en: <http://ie.fing.edu.uy/ense/assign/codif/material/monografias/2004-01.pdf>. 0-13-017547-1.

Saldivar Vargas, Guillermo de Jesús. Monografias. *Diseño de bases de datos*. [En línea] 12 de febrero de 2006. [Citado el: 7 de marzo de 2013.] <http://www.monografias.com/trabajos30/base-datos/base-datos.shtml>.

Schulzrinne, Henning, y otros. IETF Tools. *RTP: A Transport Protocol for Real-Time Applications*. [En línea] julio de 2003. [Citado el: 20 de noviembre de 2012.] <http://tools.ietf.org/html/rfc3550>. rfc 3550.

Soldani, David. *QoS and QoE Management in UMTS Cellular Systems*. s.l. : John Wiley & Sons, 2006. 0470016396.

SQLite. [En línea] [Citado el: 5 de diciembre de 2012.] <http://sqlite.org/about.html>.

The Research of Quality of Experience Evaluation Method in Pervasive Computing Environment. **Li-yuan, Liu.** Urumqi : s.n., 2006. págs. 178 - 182. 1-4244-0326-x.

Venegas Morales, Mauricio Hernán, Yánez Cañas, Aquiles y J. González, Agustín. *Transmisión de video de alta calidad a través de redes IP*. Valparaíso : Casilla 110-V, 2005. Disponible en: <http://www.grc.upv.es/docencia/tadm/practicas/P3.pdf>.

Visibility of individual packet losses in MPEG-2 video. **Reibman, Amy R., y otros.** 2004. Vol. I, págs. 171-174. 0-7803-8554-3.

BIBLIOGRAFÍA



Wu, H. R. y Rao, K. R. *Digital Video Image Quality and Perceptual Coding.* s.l. : Boca Raton : CRC Taylor & Francis, 2006. 0-8247-2777-0.