

Codificación de contenidos multimedia educativos para videostreaming

Coding multimedia educational content for videostreaming

Gabriel Elías Chanchí Golondrino* Wilmar Yesid Campo Muñoz**
Franco Arturo Urbano***

*Facultad de Ingeniería, Institución Universitaria Colegio Mayor del Cauca, Popayán, Cauca.

**Facultad de Ingeniería Electrónica, Universidad del Quindío, Armenia, Quindío.

***Facultad de Ingeniería de Sistemas, Fundación Universitaria de Popayán, Popayán, Cauca.

Resumen

En la actualidad existen diversas herramientas para la codificación de contenidos de vídeo, las cuales son usadas para cambiar el formato y modificar el nivel de compresión de los contenidos. En lo que respecta a las herramientas libres de codificación, por lo general estas carecen de interfaz gráfica, obligando al usuario a realizar este proceso mediante línea de comandos. En este artículo se presenta una propuesta de un proceso de codificación de contenidos multimedia educativos, considerando herramientas libres. A nivel metodológico, este trabajo se desarrolló en 4 fases: análisis de tecnologías y herramientas, diseño e implementación de la herramienta de codificación, construcción del escenario de videostreaming y evaluación del escenario de videostreaming. Como principales resultados de este trabajo se destacan la construcción de una herramienta de codificación para contenidos multimedia educativos y la conformación de un escenario de transmisión de streaming a partir de la herramienta desarrollada. Este trabajo permite concluir que la herramienta desarrollada posibilita la automatización del proceso de codificación de contenidos multimedia, haciendo uso de herramientas libres. Así mismo, el escenario conformado puede ser considerado como referencia en diversos entornos de aplicaciones que requieran el montaje e implementación de servicios basados en videostreaming, como es el caso de los procesos de educación virtuales.

Palabras clave: Codificación, contenido multimedia, herramienta automática, videostreaming.

Abstract

There are diverse video content coding tools, used to change the format, and modify content comprehension level. Regarding free coding tools, in general, these tools lack of a graphic interface, forcing the user to perform this process through a line of commands. This article presents a proposal for a process of coding educational multimedia contents, considering free tools. In methodological terms, this project was prepared in four phases: a) technology and tool analysis; b) design and implementation of coding tool; c) construction of videostreaming scenario, and, d) evaluation of the videostreaming scenario. The main results of this work include construction of a coding tool for educational multimedia contents, and organization of a streaming transmission space, parting from the developed tool. This work leads to conclude that the developed tool enables automation

Recibido: 20/03/2016
Revisado: 08/04/2016
Aceptado: 01/12/2016

Correspondencia de autor:
gchanchi@unimayor.edu.co
wycampo@uniquindio.edu.co
frurbano5@gmail.com

© 2016 Universidad La Gran Colombia. Este es un artículo de acceso abierto, distribuido bajo los términos de la licencia Creative Commons Attribution License, que permite el uso ilimitado, distribución y reproducción en cualquier medio, siempre que el autor original y la fuente se acrediten.

Cómo citar:

Chanchí, G.G., Campo, M.W., Urbano.F (2016) Codificación de contenidos multimedia educativos para videostreaming. *UGCiencia* 22, 71-84.



of multimedia content coding process, by using free tools. Likewise, the organized scenario may be considered as a reference in various application environments which require mounting and implementation of videostreaming-based services, as the case of virtual educational processes.

Key words: Coding, multimedia content, automatic tool, videostreaming.

Introducción

El streaming es una tecnología que permite ver u oír vídeos y audios desde internet, ya sea en directo o en diferido, sin necesidad de tener que descargar el contenido a diferentes equipos, permitiendo la reproducción de los archivos a medida que se van descargando y mejorando considerablemente los tiempos de espera. Los sistemas de videostreaming operan sobre el modelo cliente/servidor, en donde el cliente solicita datos a un servidor a través de una red y este responde a las peticiones entregando contenido multimedia el cual es interpretado por el cliente. El audio y vídeo se codifican en formatos especiales que comprimen los datos a tamaños fáciles de manejar, para que el servidor los entregue a través de la red y el cliente los interprete y despliegue para presentarlos al usuario final (Vazquez & Berrio, 2009).

Los primeros protocolos de streaming fueron desarrollados por multinacionales como Microsoft, Real o Apple, quienes vieron importantes posibilidades de ofrecer vídeo en tiempo real. A nivel técnico, existen protocolos diseñados y estandarizados para la comunicación entre los clientes y los servidores de streaming, como es el caso de RTP y RTSP (Durresti & Jain, 2005), los cuales permiten el envío extremo a extremo de contenidos multimedia a través de la red.

El videostreaming es una de las tecnologías más importantes en campos como la educación, ya que permite que los procesos de formación puedan realizarse a distancia, brindando la

oportunidad de acceder a infinidad de contenidos multimedia educativos de videostreaming no solo de audio sino también de vídeo (Nole, 2011). A pesar de las anteriores ventajas, una de las tareas más complejas de realizar en el despliegue de un escenario de videostreaming, es la de codificación y adecuación de los contenidos multimedia. Lo anterior, teniendo en cuenta que aunque existen herramientas de libre acceso para escenarios académicos, su manejo y personalización requiere conocimientos avanzados sobre las características de los contenidos (tamaño, formato, duración, calidad, resolución).

Una de las herramientas libres multiplataforma que ha sido comúnmente utilizada en entornos educativos, para la generación de contenidos multimedia es FFmpeg. A pesar de las ventajas de esta, se tiene como dificultad, la necesidad del uso de comandos especializados desde la consola del sistema operativo (tamaño, calidad, formato, resolución), los cuales cambian dependiendo del tipo de formato a codificar. Lo anterior hace que el proceso de codificación no sea inmediato y requiera cierto nivel de experticia por parte de los usuarios de la herramienta, lo que representa un mayor problema si esta herramienta es usada con propósitos educativos.

En este artículo se presenta la herramienta automática de codificación de contenidos multimedia educativos Fenoda Converter, la cual permite facilitar y hacer eficientes las

tareas de codificación de contenidos multimedia educativos, aprovechando las ventajas de la herramienta FFmpeg. A modo de validación, la herramienta Fenoda Converter fue usada para la conformación de un escenario educativo de transmisión de streaming, para soportar los procesos de enseñanza a distancia en la Fundación Universitaria de Popayán.

El presente artículo está organizado de la siguiente forma: en la sección 2 se describen los conceptos y tecnologías que se tuvieron en cuenta para el desarrollo de este artículo. En la sección 3 se presenta la estructura funcional de la herramienta automática para la codificación de contenidos multimedia elaborados, incluyendo el diagrama de casos de uso y el diagrama de clases. Así mismo se describe el escenario educativo de transmisión de streaming, en el que fue validada la herramienta Fenoda Converter. En la sección 4 se muestran las pruebas de estrés realizadas sobre el escenario de streaming en el que se enmarca la herramienta. Finalmente en la sección 5 se presentan las conclusiones y trabajos futuros derivados de este trabajo.

Materiales y métodos

Para el desarrollo del presente trabajo de investigación, se propusieron 4 fases a saber: análisis de tecnologías y herramientas, diseño e implementación de la herramienta de codificación, construcción del escenario de videostreaming y evaluación del escenario de videostreaming (ver figura 1).

En la primera fase se hizo la exploración y selección de las herramientas y tecnologías más adecuadas para la conformación de un escenario de videostreaming. En la segunda fase se procedió con el diseño e implementación de la herramienta de codificación Fenoda Converter, la cual permitió la codificación de los contenidos multimedia educativos a transmitir. En la fase

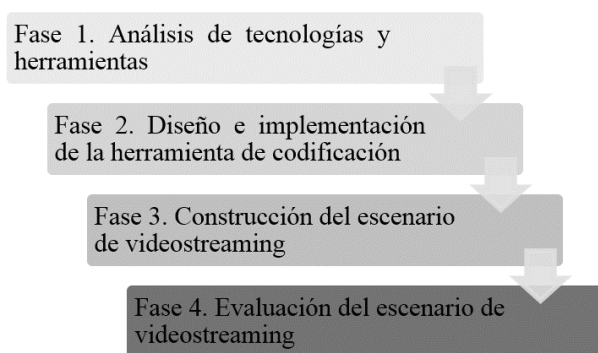


Figura 1. Metodología del trabajo de investigación

Fuente: Elaboración propia

3 se realizó la construcción de un escenario de videostreaming, para la codificación, difusión y consumo de contenidos multimedia educativos. Finalmente en la fase 4 se efectuaron las pruebas para verificar la pertinencia del escenario construido. La fase uno de la metodología es abordada en la sección: “marco conceptual”, la fase dos fue considerada en la sección: “herramienta de codificación”, la fase tres fue contemplada en la sección “escenario de videostreaming” y finalmente la fase cuatro fue incluida en la sección: “pruebas de estrés”.

Marco conceptual

En este apartado se presentan las tecnologías y conceptos más representativos, que se tuvieron en cuenta para el desarrollo del presente trabajo. Dentro de estos se encuentran: videostreaming, códec, formato de contenedor de video digital, contenidos multimedia educativos, FFmpeg.

Videostreaming

La tecnología de videostreaming es aquella que permite la reproducción de los contenidos multimedia a medida que éstos se van descargando, en lugar de forzar al usuario a obtener el video completo para poder reproducirlo. Esta forma de reproducción

es casi en tiempo real, acogiéndose a las necesidades de los usuarios y sus dispositivos de acceso. El videostreaming puede ser distribuido punto a punto (Unicast) o mediante transmisión múltiple (Multicast). Los canales de comunicación pueden ser estáticos o dinámicos, con conmutación de paquetes y/o circuitos. Estas comunicaciones pueden soportar bitrate variable y alguna forma de calidad de servicio (QoS) o simplemente trabajar en el modo de mejor esfuerzo (Rodríguez, Rosa, Costa, Abrahao, & Bressan, 2014). Las características específicas de los contenidos multimedia de video, están determinadas fuertemente por los códec y el formato contenedor, conceptos que se describen a continuación.

Códec

Un códec es un algoritmo de compresión y codificación, utilizado para reducir el tamaño de un flujo de bytes, que ocupa un archivo de video. La compresión de video se consigue mediante la explotación de las semejanzas o redundancias que existen en una señal de video típica, ejemplos de estos son: MPEG-1, MPEG-2, MPEG-4, Vorbis, DivX. Normalmente los algoritmos de compresión empleados conllevan una pérdida de calidad, por lo que siempre interesará emplear los códec con los que más compresión se logre, y menos calidad se pierda (Ghanbari, 2011).

Formato de contenedor de video digital

Un formato contenedor es un tipo de formato de archivo que almacena información de video, audio, subtítulos, capítulos, metadatos e información de sincronización (entre los datos de video, audio, etc...) Algunos ejemplos de formatos contenedores son: AVI (Audio Video Interleave), MPEG (Moving Picture Experts Group, en sus variantes MPEG 2, MPEG4, etc...), MOV (QuickTime File Format),

MXF (Material eXchange Format), WMV (Windows Media Video), OGG-Vorbis, FLV (Flash video), MKV (Matroska), entre otros. Los formatos de video más utilizados en el entorno profesional son: AVI, MOV, MXF y FLV, siendo este último ampliamente usado en plataformas como Youtube o Vimeo, y en aplicaciones de distribución de video en tiempo real. En los formatos contenedores de video aunque el envoltorio sea uno solo, las pistas de video van separadas de las pistas de audio o de los metadatos. Es importante aclarar que los formatos del tipo contenedor, no hacen referencia a los formatos de compresión o códec, puesto que la señal de video presente dentro de un archivo contenedor puede ir comprimida (o no) con un códec determinado (Covadonga, Delgado, & Sastre, 2012).

Contenidos multimedia educativos

La importancia de la incorporación de las nuevas tecnologías en las aulas ha supuesto una nueva concepción del modelo educativo. No obstante, debido a la diversidad de herramientas didácticas es necesario un trabajo previo de exploración y selección, que permita enfocar la utilidad de cada una de ellas. De acuerdo con (Vazquez & Berrio, 2009) y (Serrano & Jodar, 2010), si además de contar con el acceso a los recursos educativos que proveen los entornos virtuales de aprendizaje, tales como: presentaciones, foros, blogs, etc; se hace uso de recursos multimedia de video soportados en el servicio de videostreaming en vivo o en diferido, se tiene mayor probabilidad de apropiación del conocimiento.

Lo anterior considerando que el uso de contenidos multimedia, incentiva el desarrollo de las capacidades multi-sensoriales en los estudiantes, al incluir imagen, audio y video en un solo contenedor.

Dentro del proceso de distribución de contenido multimedia a través de videostreaming, es importante destacar la fase de codificación, debido a que esta influye en la reproducción adecuada del contenido. A continuación se analizan un conjunto de formatos contenedores de video, que se exploraron de cara a seleccionar los más adecuados para el contexto educativo, considerando la calidad y tamaño de los mismos.

1) Windows Media Video (WMV): WMV es un formato propietario de Microsoft Windows Media Video:

El flujo de vídeo es a menudo combinado con un flujo de audio de Windows Media Audio para imprimir grandes archivos de vídeo manteniendo considerablemente alta calidad. Las extensiones de este tipo de contenidos son las .asf y .wmv para el vídeo y .wma para el audio. Este tipo de vídeo se empaqueta normalmente en algún contenedor multimedia, como pueden ser AVI o ASF (Chafón, 2012).

2) Moving Picture Experts Group (MPEG): MPEG es un estándar de compresión de audio, vídeo y datos establecido por la Unión Internacional de Telecomunicaciones. Originariamente había 4 tipos diferentes MPEG-1, 2, 3 y 4, que se diferencian en la calidad y ancho de banda usado. Ofrece tres ventajas: compatibilidad mundial, gran compresión y poca degradación de la imagen (Jorissen, y otros, 2014).

3) MOV: Este contenedor fue desarrollado inicialmente por Apple y tiene una gran presencia en los entornos de video profesional, sobre todo en aquellas plataformas que utilizan el sistema operativo Mac. Presenta una gran variedad de códecs, tanto para video como para audio. Generalmente para trabajar con archivos MOV en cualquier editor de video, es necesario tener instalado el reproductor de Apple Quicktime (Ghanbari, 2011).

4) Flash video (FLV): FLV es un formato de archivo propietario usado para transmitir vídeo sobre internet (streaming) empleando Adobe Flash Player. Los contenidos FLV pueden ser incrustados dentro de archivos SWF. Están compuestos de una secuencia de bits de vídeo y un flujo de bits de audio. Los archivos FLV pueden ser distribuidos mediante la descarga progresiva vía HTTP (soportada en Flash Player 7 y versiones superiores) o mediante streaming empleando el protocolo RTSP (Chafón, 2012).

FFmpeg

La herramienta FFmpeg es una colección de software libre que brinda soluciones completas para grabar, convertir y transmitir (streaming) audio y vídeo. Está desarrollada bajo la plataforma de Linux, pero puede ser compilado en la mayoría de los sistemas operativos incluyendo Windows. Aunque FFmpeg es una herramienta que opera en modo línea de comandos, es una de las opciones con mayor número de prestaciones y mayor velocidad de procesamiento. Dentro de las librerías que conforman FFmpeg se encuentran: libavcodec, libavformat, libavutil, libpostproc, libswscale (Xiaohua, Xiuhua, & Caihong, 2013).

La presente investigación permitió elegir entre los diferentes formatos descritos, los más adecuados de acuerdo con sus beneficios en el contexto educativo. Las principales diferencias entre un formato u otro vienen dadas por el tamaño del vídeo, su calidad, el soporte para streaming o descarga progresiva y qué tan soportado y difundido es el mismo por diferentes equipos. Basándose en estas características, para la creación de la herramienta de codificación Fenoda Converter se ha elegido a FLV, MP4 (MPEG-4), como formato idóneo para distribuir vídeo en internet

y además WMV, 3GP, WMA, MP3 (MPEG-1 o MPEG-2) como formatos adicionales, todos los mencionados para la codificación de contenidos educativos multimedia usando a FFmpeg en segundo plano.

Resultados

Como principales resultados de este trabajo se generó una herramienta para automatizar el proceso de codificación de contenidos multimedia educativos. Así mismo a partir de la herramienta construida, se conformó un escenario para la transmisión de streaming educativo. A continuación, se describe cada uno de estos resultados.

A) Diagrama de bloques de la herramienta de codificación

En este apartado se describe la estructura modular y funcional de la herramienta de codificación (ver figura 2).

El módulo de “Interfaz Gráfica de Usuario”, hace uso de las librerías IO (manejo de ficheros) y Swing de java (componentes de interfaz gráfica), a través de las cuales se crean un

conjunto de componentes de interfaz gráfica, mediante los que se solicita la dirección del archivo a codificar, el tipo de formato, la calidad de la codificación y por último el destino del archivo a codificar.

Cada vez que la herramienta Fenoda Converter realiza el proceso de codificación, lanza un hilo independiente al hilo de ejecución principal. El módulo “Ejecutor de Comandos” es llamado al lanzar el hilo de codificación, y tiene por función ejecutar un comando del sistema operativo en el cual se esté ejecutando la herramienta, es decir, en el caso de Linux el comando ejecutado recibe el nombre de “Comando Linux”, mientras que el caso de Windows, este comando es llamado “Comando Windows” (ver figura 1).

Los anteriores comandos son usados desde el sistema operativo respectivo, por el módulo “Lanzador FFmpeg”, con el fin de ejecutar desde segundo plano la herramienta FFmpeg, enviándole como parámetros los datos de conversión obtenidos desde la interfaz gráfica de la aplicación Java. Para que los comandos anteriores puedan ser ejecutados, se hace uso de la “Librería OS” de JAVA, la cual permite

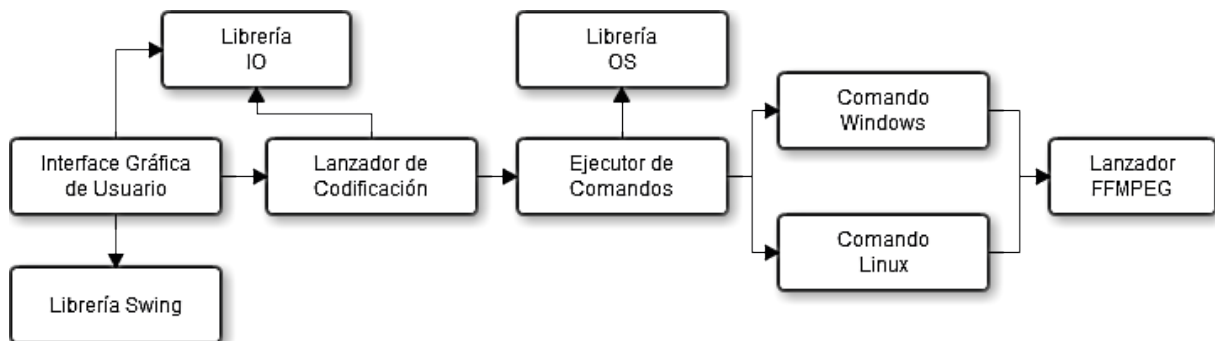


Figura 2. Diagrama de bloques herramienta de codificación.

Fuente: elaboración propia

interactuar de manera directa con la consola del sistema, ejecutando las sentencias enviadas desde el lenguaje de programación.

B) Casos de uso de la herramienta

A continuación se describen los 3 casos de uso más representativos de la herramienta, los cuales están relacionados directamente con el proceso de codificación de archivos de audio y video. Estos son: Convertir archivo de audio, convertir archivo de video y convertir calidad del archivo.

1) Convertir archivos de audio: Realiza la conversión del archivo de audio (que el usuario adjunta desde la interfaz gráfica de la herramienta), a un archivo de audio codificado en dos posibles formatos: wma o mp3.

2) Convertir archivos de video: Tiene como

propósito realizar la conversión del archivo de video (que el usuario adjunta desde la interfaz gráfica de la herramienta), a un archivo de video codificado en cuatro posibles formatos: flv, mp4, 3gp, wmv.

3) Convertir calidad del archivo: Permite modificar la calidad del archivo de audio o video a tres posibles calidades: alta media o baja. Lo anterior permite que en redes de baja velocidad, se pueda tener una mejor experiencia en el consumo de streaming.

C) Diagrama de clases de la herramienta

En esta sección se presenta y describe el modelo de clases de la herramienta Java de codificación: Fenoda Converter (ver figura 3).

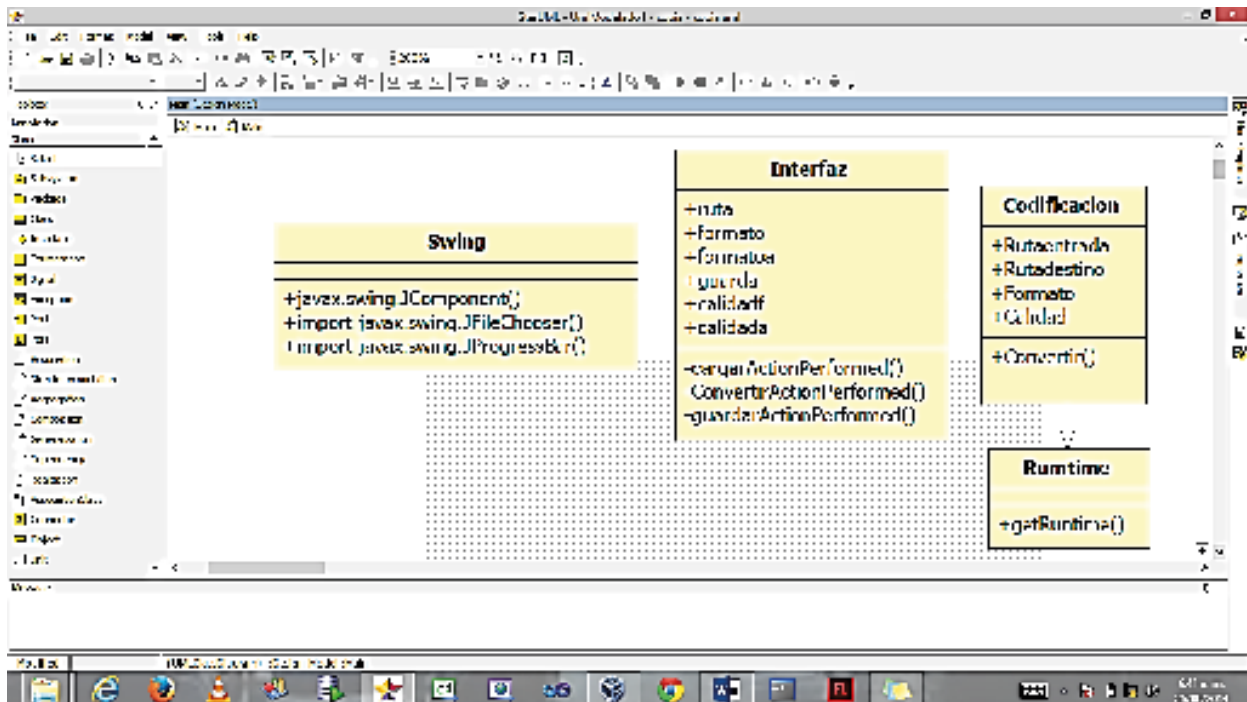


Figura 3. Diagrama de clases.

Fuente: elaboración propia

1) Clase interfaz: Esta clase permite implementar la interfaz principal de la herramienta de codificación, haciendo uso de los componentes gráficos de la librería Swing de Java. Esta última pertenece al paquete “javax.swing” y se incluyó en la versión 1.2 de la máquina virtual, como una evolución de la librería “java.awt”. Los métodos que contiene la clase interfaz son: cargarActionPerformed(), guardar ActionPerformed() y convertir ActionPerformed().

El primero se encarga de obtener la ruta de acceso del archivo adjunto y mostrarla en una caja de texto. El siguiente cumple la función de direccionar la ruta de destino del archivo final. Finalmente, el último método permite convertir el archivo adjunto, al formato y calidad escogidos desde la interfaz gráfica de usuario.

2) Clase codificación: Esta es la encargada de realizar la conversión del contenido multimedia de un formato a otro, usando en segundo plano la herramienta FFmpeg, lo anterior gracias a la clase Runtime de Java, la cual permite la ejecución de comandos del sistema operativo. Es decir, gracias a esta clase es posible la generación automática de comandos tipo FFmpeg, a partir de

los parámetros recibidos desde el objeto gráfico de la Clase Interfaz. El método que contiene esta clase es convertir(), cuya principal funcionalidad es recibir la información en cuatro variables las cuales son: Rutaentrada, Rutadestino, Formato y Calidad. De esta manera, la clase Interfaz se encarga de crear la interfaz gráfica de la herramienta, a partir de la cual el usuario puede adjuntar desde una carpeta local el contenido multimedia a codificar. Cada vez que el usuario llame al evento codificar audio o video, se crea un objeto de la clase Codificación, el cual realiza de manera independiente el proceso de codificación específico, invocando en segundo plano a la herramienta FFmpeg. Cada objeto de la clase Codificación realiza el proceso anterior en un hilo diferente, usando las capacidades multihilo de Java.

D) Uso de la herramienta

En este apartado se presenta la herramienta de codificación desarrollada en el lenguaje Java, la cual fue desplegada sobre los sistemas operativos Linux y Windows, usando FFmpeg como herramienta de background (ver figura 4).

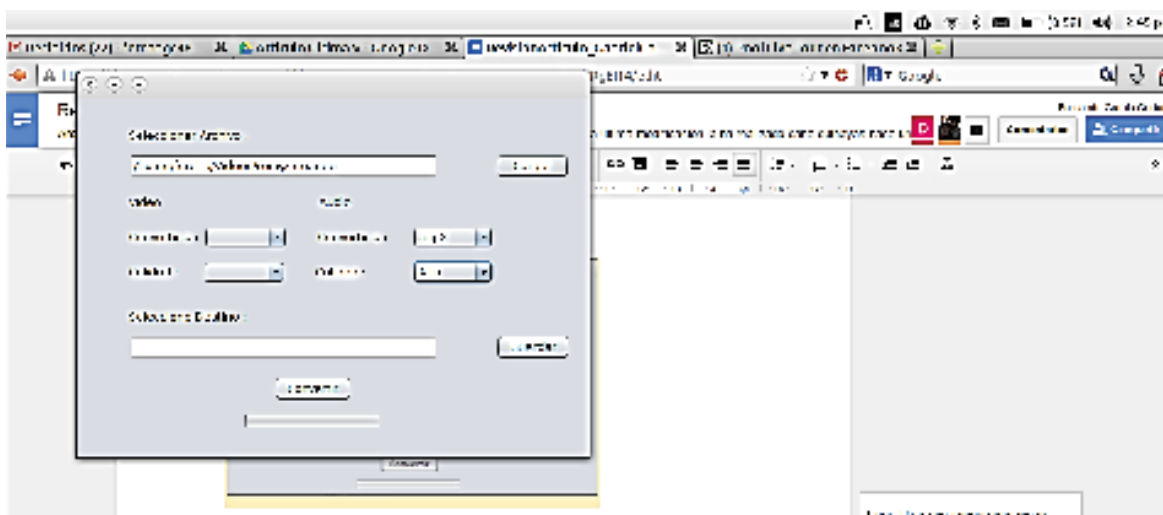


Figura 4. GUI Fenoda Converter.
Fuente: elaboración propia

Para realizar la codificación del contenido multimedia, se ingresa la ruta de destino a guardar, se eligen las opciones de formato y calidad para proceder a convertir el archivo (ver figura 5).

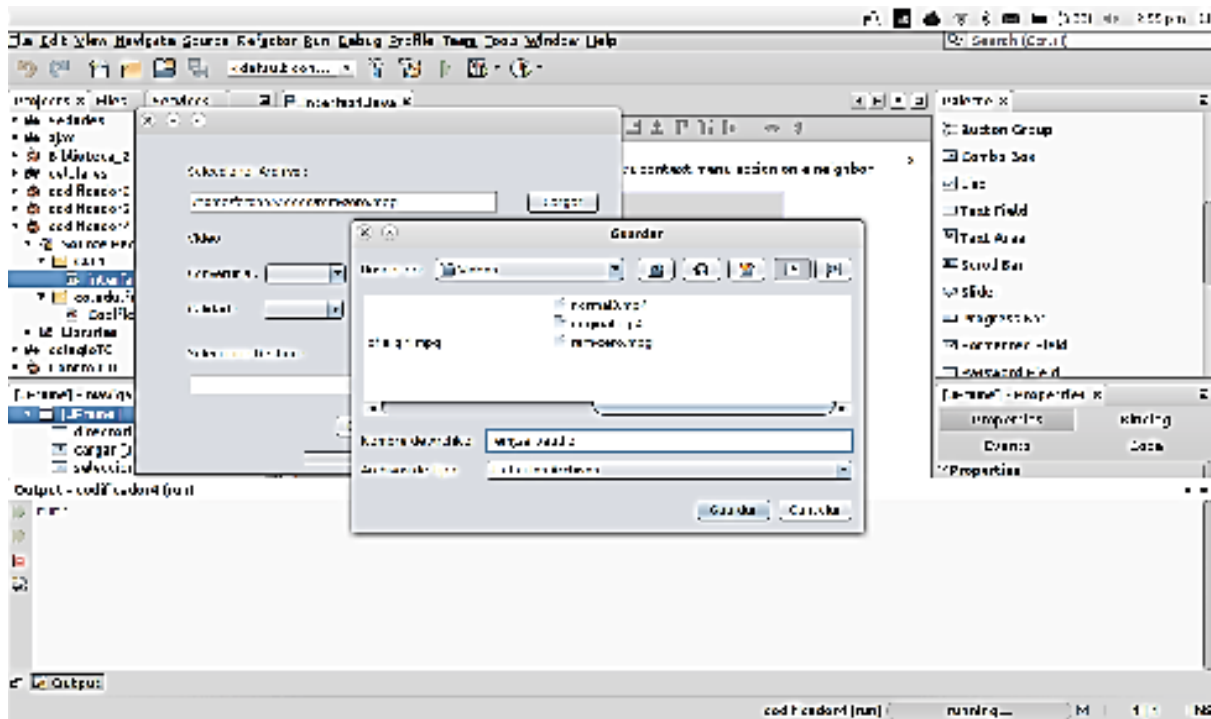


Figura 5. Manejo de la herramienta Fenoda Converter.
Fuente: elaboración propia

E) Escenario de transmisión de streaming educativo

En esta sección se presenta el diagrama con los diferentes componentes del escenario básico de videostreaming, en el cual se configuró con ayuda de la herramienta Fenoda Converter (ver figura 6). En este diagrama se destacan dos módulos principales: el servidor de streaming RTSP y los clientes de streaming, siendo estos últimos los que consumen los contenidos multimedia codificados a través de la herramienta Fenoda Converter. La comunicación entre los módulos cliente y el servidor se realiza a través de una red inalámbrica. La herramienta de codificación corre del lado del servidor y se encarga de disponer los contenidos multimedia

en un formato adecuado, dependiendo las características del servidor de *Streaming Live 555* (ver figuras 6 y 7).

1) Módulo servidor: Este está formado por el servidor de streaming live 555 y los contenidos multimedia codificados con la herramienta Fenoda Converter. El servidor de streaming escucha las peticiones RTSP por el puerto 8554 y soporta los siguientes contenedores multimedia: .264, .aac, .ac3, .amr, .dv, .m4e, .mkv, .mp3, .mpg, .ts, vob, .wav, .webm. Este módulo fue implementado sobre un computador AMD Quad Core con un sistema operativo Ubuntu Linux 11.10.

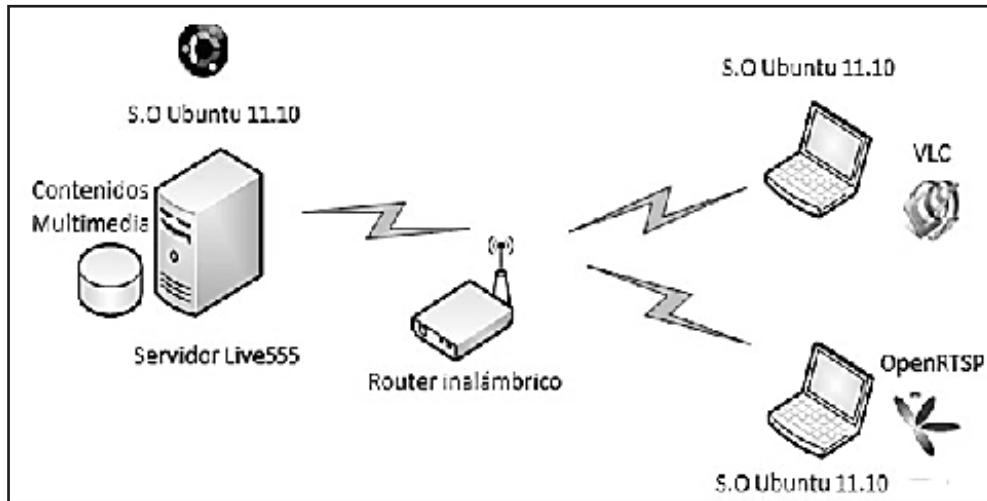


Figura 6. Diagrama del escenario de videostreaming.



Figura 7. Escenario de videostreaming real

Fuente: elaboración propia

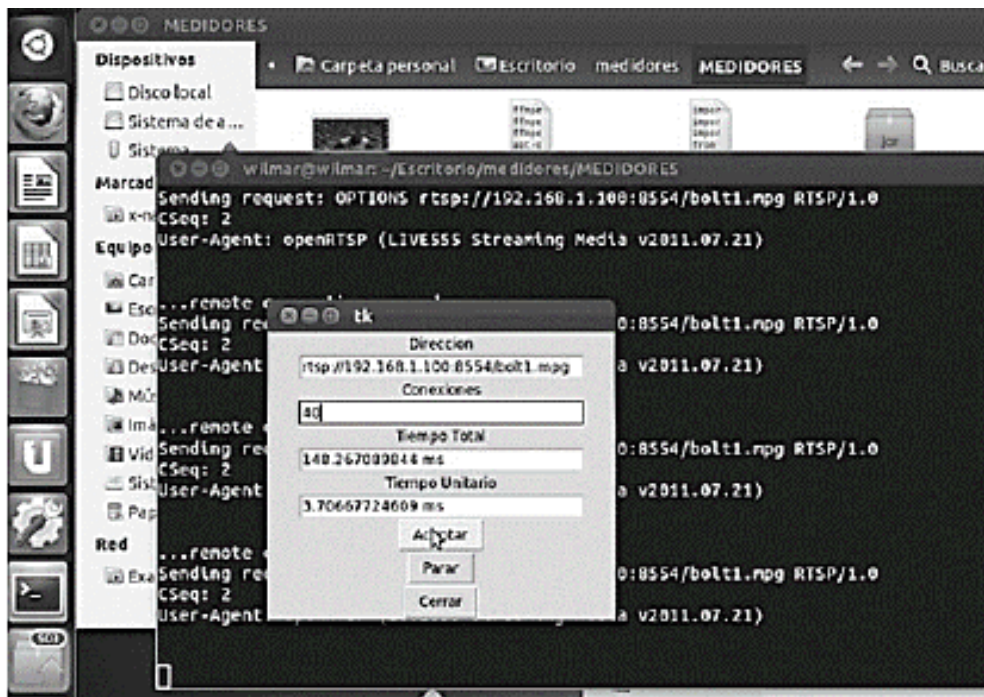
2)Modulo Cliente: Este módulo fue implementado sobre un computador AMD Quad Core con un sistema operativo Ubuntu Linux 11.10. Este módulo está formado por 2 clientes de streaming, desplegados sobre 2 computadores portátiles con un sistema operativo Ubuntu Linux 11.10. Cada computador ejecuta una aplicación Python que fue desarrollada exclusivamente para medir el estrés en el establecimiento de la conexión con el servidor de streaming.

La herramienta Python trabaja en segundo plano con los clientes libres VLC y OpenRTSP, los cuales permiten hacer peticiones RTSP al servidor de streaming Live 555. Cada cliente usa en segundo plano una herramienta diferente para la medición de estrés (VLC o OpenRTSP). Para efectos de la medición, los clientes de streaming se ejecutaron de manera independiente (ver figura 8).

Discusión de resultados

En esta sección se describen las pruebas de estrés realizadas sobre el módulo servidor del escenario de videostreaming. Estas pruebas tuvieron como objetivo la verificación de la pertinencia de la herramienta Fenoda Converter en cuanto a la codificación de los contenidos y la validez del escenario en cuanto a la transmisión de los mismos. En la figura 9 se muestran los resultados de las pruebas de tiempo de establecimiento de conexión realizadas sobre el servidor live555, y obtenidas al aplicar diversas conexiones simultáneas desde la herramienta Python de medición de estrés. Tal como se mencionó en la sección anterior, esta herramienta puede establecer dos tipos de conexiones RTSP: usando la herramienta VLC o mediante la herramienta OpenRTSP, ejecutándose estas en ambos casos en modo silencioso.

Figura 8. Herramienta de medición de estrés.



Fuente: elaboración propia

De acuerdo con la figura 9, cuando se establecen las conexiones RTSP con la herramienta de medición de estrés y con VLC, la relación entre el número de clientes y el tiempo de establecimiento de conexión es directamente proporcional, alcanzando un valor de 1500 milisegundos entre 0 y 60 clientes, y duplicando este tiempo (3000 milisegundos) entre 60 y 100 clientes.

Según la figura 9 cuando se establecen las conexiones RTSP con la herramienta de medición de estrés y con OpenRTSP, la relación entre el número de clientes y el tiempo de establecimiento de conexión, crece de manera más lenta con respecto a cuándo lo hace con VLC. De acuerdo con esta gráfica, el tiempo de conexión alcanza un valor de 250 milisegundos entre 0 y 60 clientes, y un valor de 500 milisegundos entre 60 y 100 clientes.

A partir de las anteriores pruebas, se puede concluir que los tiempos de establecimiento de

conexión provocados por peticiones simultáneas por debajo de 100 clientes, son manejables y permiten la recepción adecuada de contenido. La diferencia en el comportamiento entre VLC y OpenRTSP radica en que esta última es una herramienta para ejecutarse meramente desde consola (modo silencioso sin reproducción de contenido multimedia), por lo cual responde de mejor forma a las conexiones RTSP.

Conclusiones

La herramienta automática para la codificación de contenidos Educativos Fenoda Converter, permite la obtención de múltiples formatos (video: 3gp, flv, mp4 y audio: wma, mp3), ya sea de audio o vídeo. Esta herramienta representa una mejora en el proceso de codificación de contenidos multimedia educativos, puesto que facilita la ejecución de comandos complejos de codificación, desde una interfaz gráfica de usuario y usando en segundo plano FFmpeg.

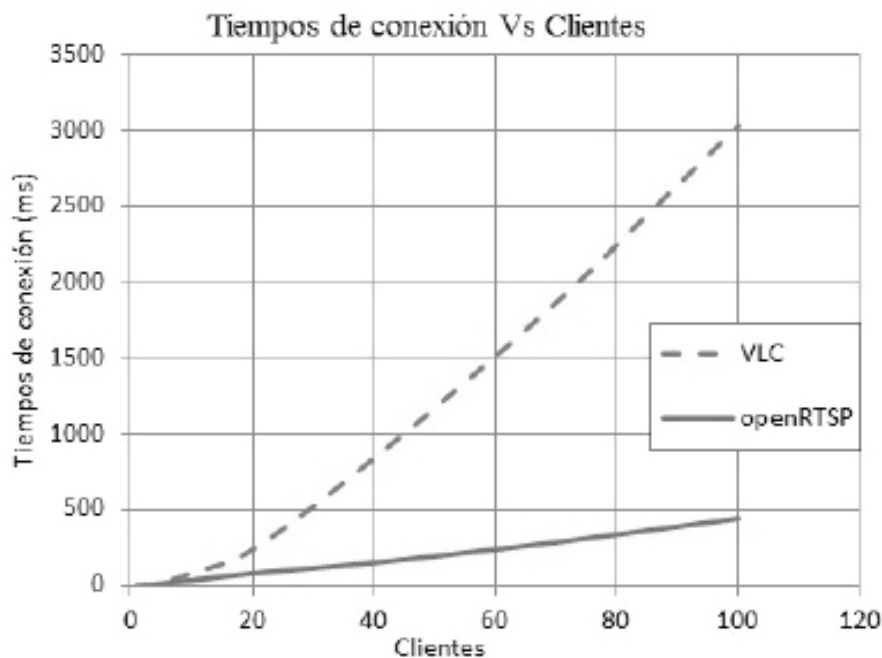


Figura 9. Evaluación de estrés

La función `exec()` de la Clase Runtime de Java, así como la capacidad de trabajo multihilo de este lenguaje, permiten la integración de múltiples aplicaciones del software libre, facilitando la automatización de tareas y el trabajo sobre diferentes sistemas operativos. En el presente trabajo, lo anterior se ve evidenciado en el desarrollo de una herramienta multiplataforma (Linux y Windows) que hace uso en segundo plano de FFmpeg, con el propósito de realizar procesos de codificación de contenidos multimedia educativos.

A través del software libre FFmpeg es posible editar un video para que este concuerde con las necesidades de transporte deseadas, como lo son formatos y códecs. Además, con FFmpeg se puede empaquetar los archivos de video en flujos de transporte, para que estos puedan ser reproducidos en diferentes dispositivos, como tabletas, móviles o computadores. Lo anterior hace que la herramienta Fenoda Converter herede estas ventajas y las potencialice en el proceso de codificación de contenidos multimedia educativos.

Actualmente los contenidos multimedia educativos pueden provenir de diferentes fuentes, lo cual se convierte en una dificultad a la hora de utilizar dichos contenidos para diferentes propósitos: reproducción local, reproducción en línea o difusión a través de un servidor de streaming. Por lo anterior, es necesario considerar los formatos, calidades, contenedores y ancho de banda más adecuados para este tipo de contenidos. La herramienta Fenoda Converter tiene en cuenta las anteriores características, con el fin de facilitar el proceso de generación de contenidos multimedia en el ámbito educativo, de tal manera que usuarios con o sin experiencia puedan usar dichos contenidos en conjunto con tecnologías como el videostreaming, para el apoyo actividades educativas.

El servidor de streaming de video Live555, mostró ser un software competitivo para la transmisión de información de video para diferentes dispositivos con distintos sistemas operativos. Su modo de operación está basado en el protocolo de transporte RTSP, por medio del cual vuelca el contenido del video en la red. Así, el escenario de streaming en el que se enmarca la herramienta “Fenoda Converter” busca servir de guía para proyectos que requieran la implementación de servicios basados en videostreaming sobre diferentes entornos de aplicación, como son el sector salud y educación.

Se demostró el correcto funcionamiento del escenario de pruebas en el que se enmarca la herramienta Fenoda Converter, con diferentes sistemas operativos y diferentes dispositivos, mediante un único servidor de videostreaming, lo que permite su potencial extensión hacia diferentes escenarios de aplicación.

Como trabajo futuro se pretende ampliar las funcionalidades de la herramienta automática Fenoda Converter, de tal modo que permita la codificación de contenidos multimedia pertenecientes a tecnologías y estándares de streaming adaptativo.

Referencias bibliográficas

- Chafón, G.** (2012). Selección de la base tecnología para la gestión de videos en la CUJAE. *Telemática. Revista Digital de las Tecnologías de la Información y las Comunicaciones*, 11(1), 71-78.
- Covadonga, R., Delgado, J., & Sastre, T.** (2012). Accesibilidad a los contenidos educativos audiovisuales: nuevas tecnologías con formatos contenedores. *Revista Iberoamericana de Educación a Distancia*, 13(2), 107-131.

- Durreesi, A, & Jain, R.** (2005). RTP, RTCP and RTSP for Real Time Applications. CRC Press.
- Emori, T, Tehrani, M. P., Takahashi, K., & Fujii, T.** (2015). Data format and view synthesis for free-viewpoint video streaming of super multiview video. Picture Coding Symposium (PCS). Antalya, Turquía.
- Fernandez, O.** (2005). *Introducción al lenguaje de programación Java - Una guía básica*. Madrid, España: Universidad Carlos III de Madrid .
- Ghanbari, M.** (2011). Standard Codecs: Image compression to advanced video coding. Colchester, UK: The Institution of Engineering and Technology.
- Jorissen, L, Goorts, P, Bex, B, Michiels, N, Rogmans, S, Bekaert, P, et-al** (2014). A qualitative comparison of MPEG view synthesis and light field rendering. 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), (pp. 1-4).
- Nole, S.** (2011). Blended Learning Nueva Alternativa al Futuro. *Revista Cognición*, 34, 1-14.
- Rodriguez, D, Rosa, R, Costa, E, Abrahao, J, & Bressan, G.** (2014). Video quality assessment in video streaming services considering user preference for video content. *Consumer Electronics, IEEE Transactions on*, 60(3), 436-444.
- Serrano, D, & Jodar, J.** (2010). El valor del videostreaming en la comunicación y formación no verbal. Un estudio en ciencias de la comunicación. *Revista Razón y Palabra* , 71, 1-12.
- Vazquez, J, & Berrio, S.** (2009). Las posibilidades didacticas y manejo de video streaming en las clases de lengua y literatura. *Revista Tejuelo*, 4, 84-101.
- Xiaohua, L, Xiuhua, J, & Caihong, W.** (2013). Design and Implementation of a Real-Time Video Stream Analysis System Based on FFMPEG. 2013 Fourth World Congress on Software Engineering (WCSE), (pp. 212-216). Hong Kong, China.