



**Diseño e Implementación de una Plataforma OTT basada en Protocolos de Streaming
Adaptativo y Software Libre**

Mateus Llerena, Edison Rodrigo

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica y
Telecomunicaciones

Dr. Olmedo Cifuentes, Gonzalo Fernando, PhD.

Sangolquí

17 de julio del 2020

Document Information

Analyzed document	TESIS_EDISON_MATEUS.pdf (D77435222)
Submitted	8/4/2020 6:54:00 AM
Submitted by	Olmedo Cifuentes Gonzalo Fernando
Submitter email	gfolmedo@espe.edu.ec
Similarity	2%
Analysis address	gfolmedo.espe@analysis.arkund.com

Sources included in the report

W	URL: https://www.idet.org.mx/noticias/la-evolucion-de-la-television-en-un-siglo-de-las- ... Fetched: 8/4/2020 6:56:00 AM	 1
W	URL: http://repositorio.puce.edu.ec/bitstream/handle/22000/11305/INFORME%20CASO%20DE%20 ... Fetched: 8/4/2020 6:56:00 AM	 5
W	URL: https://mundocontact.com/plataformas-de-servicios-ott-la-cuarta-ola-tecnologica/ Fetched: 8/4/2020 6:56:00 AM	 2
W	URL: http://www.dit.upm.es/~posgrado/doc/TFM/TFMs2015-2016/TFM_Felipe_David_Valdez_Mont ... Fetched: 8/4/2020 6:56:00 AM	 1
W	URL: https://pdfs.semanticscholar.org/f110/1845b48c607f9533428ad37dcd83ed7584a9.pdf Fetched: 8/4/2020 6:56:00 AM	 1
SA	Tesis - AÑAZCO LAZO.docx Document Tesis - AÑAZCO LAZO.docx (D41623794)	 2
W	URL: https://norfipc.com/youtube/calidad-resolucion-tamano-videos-youtube.php Fetched: 8/4/2020 6:56:00 AM	 2
SA	PTG Jorge Andree Sardinas Zuna.docx Document PTG Jorge Andree Sardinas Zuna.docx (D61776142)	 1
W	URL: https://www.xataka.com/streaming/la-compleja-infraestructura-detras-de-netflix-que ... Fetched: 8/4/2020 6:56:00 AM	 1



Firmado electrónicamente por:
GONZALO FERNANDO
OLMEDO CIFUENTES



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**Diseño e Implementación de una Plataforma OTT basada en Protocolos de Streaming Adaptativo y Software Libre**” fue realizado por el señor **Mateus Llerena, Edison Rodrigo** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 17 de julio del 2020



Firmado electrónicamente por:
**GONZALO FERNANDO
OLMEDO CIFUENTES**

.....
Dr. Olmedo Cifuentes, Gonzalo Fernando

C.C.: 1711696342

DIRECTOR



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

RESPONSABILIDAD DE AUTORÍA

Yo, **Mateus Llerena, Edison Rodrigo**, con cédula de ciudadanía n°1727500660, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Diseño e Implementación de una Plataforma OTT basada en Protocolos de Streaming Adaptativo y Software Libre”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 17 de julio del 2020

Mateus Llerena, Edison Rodrigo

C.C.: 1727500660



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Mateus Llerena, Edison Rodrigo**, con cédula de ciudadanía n°1727500660, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Diseño e Implementación de una Plataforma OTT basada en Protocolos de Streaming Adaptativo y Software Libre”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 17 de julio del 2020

Mateus Llerena, Edison Rodrigo

C.C.: 1727500660

Dedicatoria

Dedico este trabajo de titulación a Dios, por darme la sabiduría y fuerza para no decaer, quien me ha acompañado y guiado durante toda mi vida, me ha bendecido con unos padres maravillosos, con unos hermanos y familiares que de una u otra manera han podido ayudarme y alentarme cuando más lo necesitaba.

Y este logro en especial lo dedico a mis padres Edgar Mateus y Nancy Llerena, quienes han tenido que sacrificar sueños y anhelos para que mis hermanos y yo podamos cumplir los nuestros, siendo el eje fundamental de mi vida y de este logro, por el cual yo me estoy llevando el mérito gracias a ellos.

Edison Rodrigo Mateus Llerena

Agradecimiento

Agradezco a Dios por permitirme compartir mis logros con las personas que más quiero, a mi madre Nancy Llerena, a quien amo con todo mi corazón y mi vida, ella es la persona quien se encargó de tallar sus más virtuosas cualidades en mí, su inteligencia, habilidades, valores y enseñanzas me han formado como persona de bien y me han preparado para afrontar cualquier reto que se presente en mi vida. A mi padre Edgar Mateus, por el sacrificio que día a día realizó para tener el sustento económico de la familia, por impartirme sus conocimientos y gustos técnicos, además de contribuir en mi formación como persona y como hombre del hogar. A mis hermanos Viviana y Martín, quienes me han podido ayudar en todo momento, brindándome su apoyo cuando más los necesito, y a mis familiares más cercanos, quienes han contribuido en mi aprendizaje durante todo este período, aportando en mi formación como profesional.

También quiero agradecer a mis amigos y compañeros de la universidad, con quienes a lo largo de la carrera he compartido gratos momentos y han sido piezas importantes que contribuyeron para cumplir los retos que exigió este duro camino.

Finalmente, agradezco a cada uno de mis docentes de la ESPE y en especial a mi director de tesis, quienes supieron formarme como profesional, aportando con su experiencia y conocimientos que me ayudarán para desenvolverme en el campo laboral.

Edison Rodrigo Mateus Llerena

Índice de Contenidos

<i>Urkund Analysis Result</i> _____	2
<i>Certificación</i> _____	3
<i>Responsabilidad de Autoría</i> _____	4
<i>Autorización de Publicación</i> _____	5
<i>Dedicatoria</i> _____	6
<i>Agradecimiento</i> _____	7
<i>Índice de Contenidos</i> _____	8
<i>Índice de Tablas</i> _____	11
<i>Índice de Figuras</i> _____	12
<i>Resumen</i> _____	15
<i>Abstract</i> _____	16
<i>Capítulo I</i> _____	17
Planteamiento del Proyecto de Investigación _____	17
Antecedentes del Proyecto _____	17
Justificación e Importancia del Proyecto _____	19
Alcance del Proyecto _____	21
Objetivos del Proyecto _____	23
Objetivo General _____	23
Objetivos Específicos _____	23
Organización del Trabajo de Titulación _____	24
<i>CAPÍTULO II</i> _____	25
MARCO TEÓRICO _____	25
Descripción de un Servicio OTT _____	25
Difusión de Contenidos Audiovisuales _____	27
Televisión en Vivo (TV Live) _____	27

	9
Video Bajo Demanda (VoD) _____	29
Características de un Servicio de Streaming OTT _____	30
Protocolos de Streaming _____	31
Codificación de Contenidos _____	34
<i>Compresión de Audio y Video</i> _____	35
<i>Códecs de Video</i> _____	35
<i>Grupo de Imágenes (GOP)</i> _____	38
Streaming Adaptativo _____	40
<i>Sincronización del Streaming Adaptativo</i> _____	41
Componentes de una Plataforma de Streaming OTT _____	43
Ingesta de Contenidos CMS (Content Management System) __	43
Codificación de los Contenidos _____	45
Paquetización y Distribución o CDN (Content Delivery Network)	48
Encriptación del Contenido o DRM (Digital Right Management)	50
Reproducción del Contenido (Dispositivos de Usuario Final) __	51
<i>Capítulo III</i> _____	54
Implementación de la Plataforma OTT _____	54
Descripción General del Proyecto de Investigación _____	54
Hardware Utilizado _____	55
Ingesta y Codificación de Contenidos _____	58
Instalación del Software FFMPEG _____	58
Selección de los Perfiles de Codificación (Encoding Ladder) __	63
Codificación y Compresión de los Contenidos _____	67
Paquetización y Distribución del Contenido _____	71
Instalación del Software Wowza _____	71
Configuración del Motor de Streaming _____	75
Reproducción del Contenido _____	84
Acceso a los contenidos mediante VLC Media Player _____	84
Diseño Web de la Plataforma OTT _____	87
<i>Configuración del Servidor WEB</i> _____	87

	10
<i>Implementación de JW Player</i> _____	89
<i>Diseño Final de la Plataforma OTT</i> _____	91
<i>Capítulo IV</i> _____	95
Pruebas y Análisis de Resultados _____	95
Validación del Contenido Transmitido _____	95
Acceso mediante VLC Media Player _____	95
Acceso mediante la Plataforma Web (JW Player) _____	99
Desempeño del Servidor _____	103
Escenario 1: Atendiendo a un solo Usuario _____	104
<i>Conexiones en el Servidor</i> _____	106
<i>Tráfico en la Red</i> _____	107
<i>Estado del CPU</i> _____	108
Escenario 2: Atendiendo a varios Usuarios Simultáneos _____	109
<i>Conexiones en el Servidor</i> _____	111
<i>Tráfico en la Red</i> _____	113
<i>Estado del CPU</i> _____	114
Análisis y Comparación de los Escenarios _____	115
Evaluación MOS sobre el Desempeño de la Plataforma _____	116
Video de las Pruebas Realizadas _____	118
<i>Capítulo V</i> _____	119
Conclusiones y Recomendaciones _____	119
Conclusiones _____	119
Recomendaciones _____	121
Líneas de Trabajos Futuros _____	122
<i>Bibliografía</i> _____	123

Índice de Tablas

Tabla 1 <i>Protocolos de Streaming de Video</i> _____	32
Tabla 2 <i>Especificaciones Técnicas de la laptop servidor</i> _____	57
Tabla 3 <i>Especificaciones Técnicas del router</i> _____	57
Tabla 4 <i>Posiciones de ISP's según el rendimiento de Netflix en Febrero 2020</i> _____	63
Tabla 5 <i>Escalera de codificación de Netflix</i> _____	64
Tabla 6 <i>Escalera de codificación de YouTube</i> _____	65
Tabla 7 <i>Escalera de codificación propuesta para la implementación</i> _____	66
Tabla 8 <i>Descripción de los comandos ingresados para la codificación del video</i> ____	69
Tabla 9 <i>Comparación de resultados de los escenarios de pruebas</i> _____	115
Tabla 10 <i>Parámetros utilizados para la evaluación bajo el criterio MOS.</i> _____	116

Índice de Figuras

Figura 1	<i>Distribución de los servicios OTT de acuerdo con el campo en que se aplica</i>	25
Figura 2	<i>Transmisión en vivo de Diario El Comercio con 10,5 mil espectadores</i>	28
Figura 3	<i>Plataformas populares de Video Bajo Demanda VoD</i>	29
Figura 4	<i>Servicio OTT multipantalla compatible para distintos dispositivos</i>	31
Figura 5	<i>Transporte del contenido adaptativo hacia el buffer del usuario</i>	33
Figura 6	<i>Esquemas de codificación de imagen H.264/AVC vs H.265/HEVC</i>	34
Figura 7	<i>Esquemas de codificación diferencial AVC y HEVC</i>	36
Figura 8	<i>Diferencias entre codificación única y codificación diferencial</i>	37
Figura 9	<i>Secuencia típica de fotogramas o GOP</i>	39
Figura 10	<i>Cambios de perfil de calidad en un streaming adaptativo</i>	40
Figura 11	<i>Cambios de perfil de calidad y almacenamiento en el buffer</i>	42
Figura 12	<i>Empresas que brindan servicios en la nube</i>	43
Figura 13	<i>Servidores de Netflix repartidos en carriers de todo el mundo</i>	44
Figura 14	<i>Miles de archivos en los servidores de Netflix</i>	46
Figura 15	<i>Software FFMPEG de código abierto</i>	47
Figura 16	<i>Motor de Streaming Wowza, de Wowza Media Systems</i>	48
Figura 17	<i>Esquema de distribución CDN a los usuarios</i>	49
Figura 18	<i>Reproductor de código abierto VLC Media Player</i>	51
Figura 19	<i>Reproductor profesional HTML5 JW Player</i>	52
Figura 20	<i>VLC accede al contenido mediante una dirección IP</i>	53
Figura 21	<i>Diagrama de bloques de la Plataforma OTT implementada</i>	54
Figura 22	<i>Diagrama físico de red utilizado para la implementación</i>	56
Figura 23	<i>Selección del paquete de descarga de FFMPEG</i>	58
Figura 24	<i>Directorio de la carpeta creada para alojar FFMPEG</i>	59
Figura 25	<i>Archivos extraídos en la carpeta creada para alojar FFMPEG</i>	59
Figura 26	<i>Configuración del software FFMPEG mediante el Panel de Control</i>	60
Figura 27	<i>Configuración de la variable PATH con la ubicación de FFMPEG</i>	61
Figura 28	<i>Ejecución del cmd</i>	61
Figura 29	<i>Resultado del comando ingresado para verificar la versión de FFMPEG</i>	62

Figura 30	<i>Captura de pantalla de FFMPEG al finalizar la codificación del video</i>	68
Figura 31	<i>Captura de la página oficial de descarga de Wowza</i>	72
Figura 32	<i>Captura de la página para solicitar la licencia de desarrollador de Wowza</i>	72
Figura 33	<i>Ventana de instalación guiada de Wowza</i>	73
Figura 34	<i>Ventana de “Servicios” de Windows para iniciar Wowza</i>	74
Figura 35	<i>Verificación del funcionamiento de Wowza en el ordenador</i>	75
Figura 36	<i>Interfaz principal del motor de streaming de Wowza</i>	76
Figura 37	<i>Selección del tipo de servicio de streaming que se va a ofrecer</i>	77
Figura 38	<i>Asignación de un nombre a la aplicación de VoD creada</i>	77
Figura 39	<i>Configuración de la tecnología de streaming soportada por la plataforma</i>	78
Figura 40	<i>Configuración del directorio para alojar los videos del servidor</i>	79
Figura 41	<i>Autenticación y restricciones del cliente para acceder a los contenidos</i>	79
Figura 42	<i>Creación del archivo SMIL para el primer video codificado</i>	80
Figura 43	<i>Configuración del primer nivel de la escalera de codificación</i>	81
Figura 44	<i>Grupo de niveles correspondientes al primer video codificado</i>	81
Figura 45	<i>Archivos SMIL que referencian a los videos disponibles en la plataforma</i>	82
Figura 46	<i>Contenido de la carpeta “content” dentro del disco local</i>	83
Figura 47	<i>Captura de la página oficial de descarga de VLC Media Player</i>	84
Figura 48	<i>URL del video mediante el reproductor de prueba de Wowza</i>	85
Figura 49	<i>Acceso al contenido de la plataforma mediante VLC Media Player</i>	86
Figura 50	<i>Reproducción del video mediante VLC Media Player</i>	86
Figura 51	<i>Página oficial de descarga del software NetBeans para el desarrollo web</i>	87
Figura 52	<i>Página oficial de descarga del paquete de software libre XAMPP</i>	88
Figura 53	<i>Captura del panel de control de XAMPP donde se activa Apache</i>	89
Figura 54	<i>Captura de la página oficial de registro de JW Player</i>	89
Figura 55	<i>Captura de la página web resultante del código para reproducir un video</i>	91
Figura 56	<i>Captura de la página web resultante del algoritmo implementado</i>	92
Figura 57	<i>Diseño final de la plataforma OTT implementada vista en un PC</i>	93
Figura 58	<i>Capturas de la plataforma OTT desde varios dispositivos móviles</i>	94
Figura 59	<i>Información en tiempo real de contenido reproducido en VL</i>	95
Figura 60	<i>Captura de VLC e información en tiempo real después de unos segundos</i>	96

Figura 61	<i>Información en tiempo real del contenido mediante la consola de VLC</i>	97
Figura 62	<i>Información estadística de la transmisión mediante la consola de VLC</i>	98
Figura 63	<i>Captura de la plataforma y ajuste de calidad en el reproductor</i>	99
Figura 64	<i>Posible flujo reproducido en los distintos niveles de calidad del video</i>	100
Figura 65	<i>Variación de la calidad del video reproducido en un dispositivo móvil</i>	101
Figura 66	<i>Variedad de contenidos disponibles en la plataforma implementada</i>	102
Figura 67	<i>Gráficos en tiempo real que monitorean el servidor de Wowza</i>	103
Figura 68	<i>Cambio automático de calidad evidenciado</i>	104
Figura 69	<i>Uso actual del servidor mientras atiende a un solo usuario</i>	105
Figura 70	<i>Número de conexiones reales en el servidor</i>	106
Figura 71	<i>Cantidad de bits salientes en el servidor de la plataforma</i>	107
Figura 72	<i>Porcentaje de CPU utilizado por el servidor</i>	108
Figura 73	<i>Calidad establecida a 360p en todos los dispositivos</i>	110
Figura 74	<i>Uso actual del servidor mientras atiende a varios dispositivos</i>	111
Figura 75	<i>Número de conexiones promedio y reales en el servidor</i>	112
Figura 76	<i>Cantidad de bits salientes promedio y reales en el servidor</i>	113
Figura 77	<i>Porcentaje promedio y real de CPU utilizado por el servidor</i>	114
Figura 78	<i>Mínima puntuación obtenidos mediante la evaluación MOS</i>	117
Figura 79	<i>Código QR del enlace para visualizar el video realizado</i>	118

Resumen

La manera de ver televisión ha evolucionado durante los últimos años, convirtiendo al Internet el medio preferido para la difusión de contenidos, pero la inestabilidad de las redes exige que dichos contenidos puedan ser adaptados a estas condiciones, lo que estimula a grandes plataformas y empresas de streaming ofrecer servicios OTT para mejorar la experiencia del usuario. En el presente trabajo de titulación se explica el diseño e implementación de una plataforma OTT de video bajo demanda VoD que utiliza técnicas de streaming de video adaptativo y herramientas de software libre, lo que servirá como guía de desarrollo para quienes desean emprender en el mercado de la difusión de contenidos. En primera instancia, se describen conceptos básicos para comprender el proceso adaptativo que se realiza, y se configura paso a paso todos los módulos que conforman esta plataforma, además se realiza un estudio de la velocidad de conexión en el Ecuador para definir la escalera de codificación de la plataforma y así asegurar que los usuarios gocen de un servicio de calidad, adicionalmente se desarrolla una interfaz web amigable para facilitar el acceso a los contenidos que se encuentran disponibles. Finalmente se somete a varias pruebas de funcionamiento para corroborar el correcto desempeño del servicio, analizando parámetros que caracterizan a una transmisión y evaluando el rendimiento que presenta la plataforma bajo el criterio MOS.

- Palabras Clave:

- **SERVICIO OTT**
- **STREAMING ADAPTATIVO**
- **ESCALERA DE CODIFICACIÓN**
- **GRUPO DE IMÁGENES GOP**
- **CRITERIO MOS**

Abstract

The way of watching television has evolved in recent years, making the Internet the preferred medium for the dissemination of content, but the instability of networks requires that such content can be adapted to these conditions, which stimulates large platforms and companies of streaming offer OTT services to enhance the user experience. In this degree work, the design and implementation of an OTT platform for video on demand VoD that uses adaptive video streaming techniques and free software tools is explained, which will serve as a development guide for those who want to start in the content broadcasting market. In the first instance, basic concepts are described to understand the adaptive process that is carried out, and all the modules that make up this platform are configured step by step. In addition, a study of the connection speed in Ecuador is carried out to define the coding ladder. of the platform and thus ensure that users enjoy a quality service, in addition a friendly web interface is developed to facilitate access to the content that is available. Finally, it undergoes several performance tests to corroborate the correct performance of the service, analyzing parameters that characterize a transmission and evaluating the performance presented by the platform under the MOS criteria.

- Keywords:

- **OTT SERVICE**
- **ADAPTIVE STREAMING**
- **CODING LADDER**
- **GROUP OF PICTURES GOP**
- **MOS CRITERIA**

Capítulo I

1. Planteamiento del Proyecto de Investigación

1.1 Antecedentes del Proyecto

En los últimos años, la evolución de las telecomunicaciones ha dado un giro importante en la manera de ver televisión y escuchar radio, ahora se pueden acceder a estas emisiones a través de Internet. La mayor ventaja de esta forma de transmisión es no hacer uso del espectro radioeléctrico, el cual es un recurso limitado controlado por el Estado. Además, gracias al desarrollo de plataformas con contenidos bajo demanda, los usuarios ya no se rigen a un horario de programación dispuesto por alguna cadena televisiva, más bien, el usuario es quien controla qué es lo que quiere ver, cuándo lo quiere ver y el momento en que lo visualizará. Estas plataformas han logrado una gran popularidad en todo el mundo, tales como Netflix, HBO, YouTube, etc., que actualmente son el sitio preferido por las personas para ver series, novelas o películas con el requisito principal de tener acceso a Internet. El desarrollo de estas plataformas está basado en un modelo de negocio que usan las empresas de telecomunicaciones para brindar el servicio OTT (*Over the Top TV*), que utiliza técnicas de codificación adaptativa con el fin de adaptarse a las condiciones de red del usuario y acoplarse al tamaño de la pantalla donde se visualice el contenido (Xataca, 2019). Un servicio OTT permite un streaming continuo de los contenidos, recibiendo contenido a través de Internet abierto y no gestionado, mejorando enormemente la experiencia del usuario (QoE, de sus siglas en inglés *Quality of Experience*) y la calidad del servicio (QoS, de sus siglas en inglés *Quality of Service*), lo que IPTV no puede lograr ya que exige una red propia y gestionada para todos los usuarios, lo cual no es posible de lograr (Telefónica, 2015).

En este trabajo de titulación, se realiza un análisis de la manera en que se difunden los contenidos bajo demanda o televisión en vivo. Además, se trata de explicar de una manera clara y precisa las características y bondades que el servicio OTT ofrece, diferenciándola de IPTV y así aclarar que son modelos de negocio que se diferencian el uno del otro al momento de prestar los servicios de streaming.

La implementación de una plataforma OTT basándose en software libre es la clave que permitirá avanzar con el estudio y desarrollo de este servicio, ya que, teniendo una herramienta de simulación, se la puede someter a pruebas de laboratorio con el fin de detectar fallas y mejorar esos aspectos con el tiempo. Para esto, es importante destacar que en el presente trabajo se realiza un análisis del funcionamiento de la plataforma OTT y de los módulos que la componen, para así determinar y elegir el software que simule las funciones de cada módulo, teniendo en cuenta aspectos que caracterizan al servicio OTT, como lo es, el proceso de codificación adaptativa y distribución de contenidos adaptables a múltiples pantallas.

Finalmente, se evalúa el desempeño de la plataforma mediante métricas como el tráfico cursado, número de conexiones simultáneas, calidad de la imagen transmitida y la cantidad de recursos utilizados por el servidor y así determinar la de calidad del servicio (QoS), mientras que para evaluar la calidad de experiencia de usuario (QoE) se utiliza un método de valoración denominado MOS (del inglés, *Mean Opinion Score*) que permite valorar la experiencia del usuario, realizando pruebas con distintos dispositivos conectados a la red, accediendo a los contenidos de la plataforma desde múltiples pantallas. Como paso final, se detallan caminos de estudio que permitan continuar con la investigación, que día a día utilizamos este tipo de servicios sin conocer de ellos, ni darnos cuenta del proceso adaptativo que este realiza.

1.2 Justificación e Importancia del Proyecto

Uno de los grandes inventos de la historia de la humanidad es, indudablemente, la televisión. La 'caja tonta' fue un hito nacido en el siglo XX y marcó el inicio de una nueva etapa en la historia de la comunicación y del desarrollo del hombre. Su evolución hasta los televisores inteligentes de hoy en día ha durado más de un siglo (IDET, 2018).

Hoy en día la industria de la televisión está sufriendo una evolución importante en la manera de ver televisión, ya no es necesario tener un televisor para poder ver la programación diaria, simplemente es necesario tener acceso a Internet y poder disfrutar de una gran cantidad de canales en vivo tanto locales como internacionales. Los medios de comunicación han migrado hacia la web, la radio y televisión son los medios de comunicación más antiguos de la historia y finalmente tienen un avance importante después de tantos años de su aparición.

El surgimiento de IPTV fue a partir del año 2005, algunos operadores de telecomunicaciones empezaron a dar el servicio de televisión sobre redes de Internet, en lugar de medios tradicionales como TV terrestre (analógica o digital), satélite o cable. La Televisión por Protocolo de Internet (IPTV, en inglés *Internet Protocol Television*) se ha convertido en la denominación más común para los sistemas de distribución por suscripción (Huidobro, 2017). Mientras que la tendencia a nivel mundial apunta al crecimiento del uso de servicios de video por internet a través de múltiples dispositivos. El control remoto del televisor ha sido reemplazado por los clics en la red y esto ha cambiado el comportamiento del usuario, de acuerdo con un estudio realizado por la Asociación de Consumo Electrónico de Estados Unidos, el suscriptor es quien escoge lo que quiere ver, el horario y lo repite cuantas veces lo desea (El Herald, 2015).

Pese a que IPTV se ha convertido en un tipo de servicio de streaming muy popular e innovadora en los últimos años, la mayoría de los usuarios que accedían a este servicio disponían de anchos de banda variables y distintas condiciones en su red para salir a internet, por lo cual, la calidad de servicio (QoS) y calidad de experiencia del usuario (QoE) no estaban garantizadas por el proveedor del servicio. El desarrollo de nuevas tecnologías da como resultado otro modelo de negocio, el cual usan las empresas de telecomunicaciones para brindar servicios de streaming, como lo es OTT (*Over the Top*), que ha incrementado enormemente gracias a sus características que se adaptan a distintos dispositivos y diferentes anchos de banda que el usuario pueda tener, teniendo una mejoría en la calidad de servicio y mejorando la experiencia del usuario, convirtiéndose en el servicio de streaming que mayor acogida tiene.

Los productores de contenido (cadenas de televisión, estudios de cine, productoras independientes, etc.) así como los comercializadores de contenido (Netflix, iTunes, Google Play, Amazon Video, HBO GO, etc.) están sacando provecho y capitalizando esta tendencia, utilizando la infraestructura de acceso y ancho de banda de Internet de los operadores de telecomunicaciones como medio para llegar al usuario final. Sin embargo, por lo general no existen acuerdos comerciales entre los operadores y los comercializadores de contenido que permitan compartir los créditos generados por este negocio (Naranjo, 2016).

Con este tipo de plataformas, se tiene una visión amplia de negocio en la manera de difundir los contenidos, sin usar espectro radioeléctrico y logrando un mayor alcance de usuarios, a diferencia de un medio de comunicación convencional (como la televisión analógica o la TDT), donde su señal tiene cobertura solo en la zona que fue concesionada por el gobierno.

Mediante el internet, el alcance puede ser mundial, cumpliendo con el concepto de “*TV Everywhere*”, donde el usuario puede acceder al contenido desde distintas ubicaciones geográficas, en el momento que quiera y las veces que el usuario lo desee. Convirtiéndose en un servicio tan utilizado cada día debido a múltiples aspectos técnicos, de disponibilidad, múltiples pantallas, calidad de servicio y experiencia de usuario que mejoran rápidamente con el tiempo gracias a los estudios realizados por expertos del procesamiento del video (Beltrán & Fernández, 2017).

Actualmente, el mercado de la difusión de contenidos en Internet está siendo dominado por empresas regionales y por plataformas que ofrecen las mismas operadoras de telecomunicaciones, por estos motivos, es muy importante tener expertos nacionales que conozcan de este servicio denominado OTT, el cual usamos día a día sin darnos cuenta, las bondades que ofrece, y el impacto comercial que posee, teniendo una visión más clara para implementar plataformas similares pero a manera local, difundiendo contenidos nacionales e internacionales, noticieros y problemáticas propias del país donde se ponga en marcha la plataforma, además dando paso a la difusión de eventos culturales y musicales que se generen en cada ciudad, permitiendo administrar el ancho de banda de forma adecuada por las empresas de telecomunicaciones locales, ahorrando costos en capacidad de tráfico que sale del país y además, mejorar la calidad del contenido hacia los usuarios.

1.3 Alcance del Proyecto

El alcance de este proyecto empieza por el análisis del estado del arte, para entender la manera en que los contenidos se difunden en Internet, las plataformas existentes y el método de acceso a dichos contenidos.

Además, se determina las características principales que una plataforma OTT debe ofrecer, las cualidades que posee este tipo de servicio, sus ventajas y desventajas y diferenciando de otros servicios como IPTV, logrando una idea más clara y concreta del servicio OTT, adicionalmente se realiza una breve investigación sobre el ancho de banda promedio que posee la mayoría de hogares en el Ecuador, esto es con el fin de elegir la escalera de codificación de los contenidos, la cual hace referencia a las distintas resoluciones de video disponibles y puedan ser visualizadas por la mayoría de usuarios en su mejor calidad posible.

Mediante un análisis técnico sobre las plataformas OTT, se detalla las funciones realizadas por los módulos que las componen y así, identificar herramientas de software que permitan realizar dichas tareas en menor escala.

Finalmente, se implementa un simulador de la plataforma OTT, adaptando los módulos que ejecutan las funciones de ingesta de contenidos, codificación de estos, y distribución hacia los usuarios. Esta plataforma está basada en software libre, la cual permite desarrollar y realizar mejoras pertinentes en la plataforma, con el fin de convertirla en algo más robusto, a medida que se realicen experimentos y trabajos futuros, evaluando el desempeño de la plataforma, teniendo en cuenta las métricas que se generan al momento de realizar una transmisión de datos, realizando pruebas de desempeño en distintos dispositivos conectados en la red y distintos tamaños de pantalla que son servidos por la plataforma implementada.

Se encaminan futuros campos de estudio que puedan mejorar el desempeño de la plataforma OTT implementada, para así darle continuidad al trabajo realizado; dejando constancia de esto, la redacción de los resultados obtenidos, conclusiones,

recomendaciones e inconvenientes que se presenten durante el desarrollo, además de un video cargado a YouTube sobre las pruebas de funcionamiento realizadas.

1.4 Objetivos del Proyecto

1.4.1 Objetivo General

Diseñar, implementar y evaluar el desempeño de una plataforma OTT basada en protocolos de streaming adaptativo, utilizando software libre.

1.4.2 Objetivos Específicos

- Analizar la manera que se difunden contenidos audiovisuales en plataformas de televisión por internet.
- Determinar las principales características que un servicio OTT puede ofrecer a los usuarios y diferenciarlo con otro tipo de servicios de streaming de video como IPTV.
- Analizar las distintas etapas o módulos que posee una plataforma de streaming de video que se basa en un modelo de negocio del tipo OTT.
- Implementar un simulador de la plataforma OTT, basándose en software libre con sus distintos módulos y técnicas de codificación adaptativa del contenido que caracterizan a este servicio.
- Evaluar el desempeño, usando las métricas generadas en una transmisión de datos, con distintos dispositivos servidos por la plataforma OTT que se encuentra implementada.
- Proponer caminos de continuidad para este proyecto, estableciendo líneas futuras de estudio sobre el servicio OTT.

1.5 Organización del Trabajo de Titulación

El proyecto de investigación realizado está detallado en el presente documentado de la siguiente manera:

El Capítulo I, expone el planteamiento del proyecto de investigación, describiendo los antecedentes sobre el tema, la justificación e importancia, el alcance y los objetivos que se pretende alcanzar con el desarrollo este proyecto de investigación.

El Capítulo II, describe los fundamentos teóricos del proyecto, indagando en la manera en que se distribuyen los contenidos hoy en día, y detallando las cualidades que caracterizan al servicio de streaming OTT, términos y procesos que intervienen para implementar la plataforma mediante herramientas útiles y de código abierto.

El Capítulo III, detalla la implementación de la Plataforma OTT paso a paso y las distintas configuraciones de las herramientas de software utilizadas en el proyecto, además del desarrollo de una página web de la plataforma, con el fin de presentar los contenidos bajo demanda de una forma mucho más amigable para los usuarios.

El Capítulo IV, presenta los resultados y pruebas realizadas con la ayuda de 6 usuarios reales, esto fue a causa de la crisis sanitaria del Covid-19 que atraviesa el mundo, lo que impidió realizar pruebas de laboratorio con una cantidad significativa de usuarios; además se evidencia el funcionamiento mediante un video subido a YouTube.

Finalmente, el Capítulo V muestra las conclusiones que se obtuvo en el desarrollo de este proyecto, se detallan recomendaciones basadas a la implementación realizada y se encaminan trabajos futuros con el uso de esta plataforma OTT.

CAPÍTULO II

2. Marco Teórico

2.1 Descripción de un Servicio OTT

Para entender lo que es un Servicio OTT, primero se debe conocer el significado de sus siglas OTT (*Over The Top*), siendo un término asociado a la distribución de contenidos mediante la red de Internet desde un servidor hasta los usuarios finales, sin intervención alguna del proveedor de Internet (ISP, de sus siglas en inglés *Internet Service Provider*) en su venta, promoción y distribución del servicio (Valdez, 2016).

Los servicios OTT, son aquellos que su distribución se la realiza por Internet, pero no necesitan grandes inversiones de dinero, ni de gran infraestructura de telecomunicaciones ya que su difusión utiliza las redes de los ISP's y demás Carrier's a través del mundo, tampoco utilizan el espectro radioeléctrico para su distribución.

Figura 1

Distribución de los servicios OTT de acuerdo con el campo en que se aplica



Nota. Tomado de *Regulación de los Servicios OTT*, AFN Noticias México, 2018.

Los servicios OTT que más popularidad tienen hoy en día, son las aplicaciones que en un inicio fueron desarrolladas por startups como Skype, Netflix, WhatsApp, YouTube, Spotify, y que ahora son grandes empresas, que inundan de un gran tráfico a las redes de nuestros proveedores. Estos servicios mencionados no requieren un ancho de banda específico o una tasa de bit fija por parte de los usuarios, siendo esta la razón para que los contenidos se adapten a la tasa de bit que posee el usuario en su red, mediante protocolos que monitorean periódicamente las condiciones del acceso a Internet que posee el usuario, la capacidad del buffer que dispone el dispositivo final y entre otros parámetros, para que la experiencia del usuario QoE (*Quality of Experience*) no se vea afectada, pero no todo es perfecto en estos servicios, al mejorar enormemente la QoE, se debe sacrificar la calidad de los contenidos, afectando la calidad del servicio QoS (*Quality of Service*), la cual no es garantizada por dichos servicios, sin embargo el usuario a preferido mejorar la experiencia del servicio, que mantener calidades que no soportan todas las redes (Naranjo, 2016).

El surgimiento de los Servicios OTT se han convertido en una problemática para las grandes operadoras de telecomunicaciones en cada país, ya que son competidores directos con los servicios convencionales ofrecidos por dichas operadoras como lo es la voz, mensajería y la TV pagada. Por ejemplo, Skype (creada en el 2003 y actualmente perteneciente a *Microsoft*) cuenta con 250 millones de usuarios activos cada mes, los cuales hablan 100 minutos promedio, lo que se convierte en una alternativa que evita el uso de la telefonía tradicional (Sarmiento, 2014). A este mercado se le suma WhatsApp (ahora perteneciente al gigante *Facebook*) que en un inicio solo se podía enviar y recibir mensajes con otros contactos en tiempo real, pero ahora es posible realizar llamadas y videollamadas con otros usuarios en cualquier parte del mundo.

Hoy en día las operadoras de servicios de telecomunicaciones, se están convirtiendo en simples proveedores de “Internet Móvil”, basándose en las últimas tecnologías de banda ancha móvil 2G, 3G, 4G y el nuevo 5G, que se caracterizan por su velocidad a la hora de descargar contenidos por parte de los usuarios, ya que los competidores en mensajería, llamadas de voz y videoconferencias como Facebook Messenger, WhatsApp, Skype y otras más, están impactando en el mercado abarcando gran cantidad de usuarios día a día (Sarmiento, 2014).

2.2 Difusión de Contenidos Audiovisuales

La difusión de contenidos audiovisuales en la red puede ser presentada como contenidos en vivo y contenidos grabados, que han ido transformado la manera de ver televisión hoy en día.

2.2.1 Televisión en Vivo (TV Live)

La televisión en vivo, sigue siendo la manera tradicional de ver televisión, pero la diferencia es que, para acceder a los canales no necesariamente debemos tener un televisor (ya sea analógico o digital), las empresas televisivas han migrado al Internet, y al mismo tiempo que su señal se distribuye dando uso del espectro radioeléctrico para los usuarios convencionales de TV (regidas por el ente regulador de cada país), también realizan al mismo tiempo transmisiones en vivo de sus contenidos, es decir, se ayudan de herramientas como Facebook Live, YouTube o su propia página web, donde no existe un ente que regule el uso de estas plataformas, y así llegar a más espectadores en el mercado mundial, inundando el Internet con su señal.

Al ser una transmisión en vivo, significa que el contenido se reproduce a medida que se lo va capturando, es decir, el usuario final no tiene posibilidad de controlar la línea del tiempo de dicha reproducción (Beltrán & Fernández, 2017). En la *Figura 2* se puede observar una transmisión en vivo de Diario El Comercio vía Facebook Live con alrededor de 10.500 espectadores.

Figura 2

Transmisión en vivo de Diario El Comercio con 10,5 mil espectadores



En el Ecuador varias televisoras han migrado a plataformas digitales como es el caso de Telemazonas, Ecuavisa, Ecuador TV, etc. y además, esta forma de ver televisión a dado paso para que empresas que en un inicio no se dedicaban a brindar este servicio, transformen su modelo de negocio y empiecen a difundir contenidos audiovisuales en la red, como es el caso de Diario El Comercio, siendo un diario ecuatoriano que inició en los años de 1906, teniendo más de un siglo de existencia en el país, evidenciaron que con la inundación tecnológica, poco a poco menos lectores adquieren sus ejemplares matutinos, esta razón fue motivo para migrar a lo digital con

su noticiero El Comercio TV, así como este ejemplo de migración digital, entidades gubernamentales y municipales también utilizan estas plataformas para difundir noticias, comunicados, etc. y entablar un contacto más directo con la comunidad.

2.2.2 Video Bajo Demanda (VoD)

Esta forma de ver televisión muestra contenidos que son previamente grabados y almacenados en un servidor en el Internet, donde los usuarios pueden acceder a dichos contenidos en el momento y a la hora que lo desee, desde cualquier parte del mundo (Piqueres, 2019).

El usuario aquí tiene el control total del contenido y de su línea del tiempo, teniendo la posibilidad de pausar, regresar, adelantar y ver los contenidos en cualquier momento, ya sea desde un smartphone, tablet, PC o Smart TV (Beltrán & Fernández, 2017).

Figura 3

Plataformas populares de Video Bajo Demanda VoD



En el país se utilizan varias plataformas populares de VoD como Netflix, YouTube y HBO, pero últimamente las operadoras de telefonía móvil como las empresas de televisión satelital pagada también lanzaron sus plataformas, como es el caso de Movistar Play, Claro Video, Directv Play, entre otros. Hoy en día nadie quiere quedarse en el pasado, la evolución al momento de ver televisión ha generado nuevos mercados con mayores ingresos a los habituales.

2.3 Características de un Servicio de Streaming OTT

Existen varias características clave que diferencian un servicio de streaming de video OTT con un servicio IPTV, en las que interviene mucho la QoS (Calidad del Servicio) y la QoE (Calidad de Experiencia), donde el servicio OTT sacrifica algo de QoS para mejorar enormemente la QoE, basándose en estudios y análisis realizados con el método de valoración MOS (del inglés, *Mean Opinion Score*), el cual permite valorar la experiencia del usuario, que sin duda, se ha llegado a la conclusión de que el usuario prefiere una reproducción sin cortes, dándole mayor prioridad a ser solventado este inconveniente, sin importar que se deba sacrificar la calidad de la imagen que se recibe al momento de visualizar una transmisión; a diferencia de IPTV que mantiene una sola calidad en la imagen, donde la reproducción se ve afectada debido a que los usuarios no poseemos un ancho de banda controlado, ni poseemos una velocidad constante de conexión hacia Internet, evidenciando el inconveniente mediante pausas o cortes al ver la transmisión (Xataca, 2019).

A continuación, se describen parámetros importantes, los cuales deben ser explicados para comprender el proceso y funcionamiento de un servicio de streaming de video OTT.

Figura 4

Servicio OTT multipantalla compatible para distintos dispositivos



Nota. Tomado de *Análisis y diseño de una plataforma para brindar servicios de video multipantalla OTT en una empresa de telecomunicaciones*, Naranjo J. G., PUCE, 2016.

2.3.1 Protocolos de Streaming

Existen varios protocolos necesarios para el transporte de video por Internet, que se concentran en dos grandes grupos: TCP (*Transmission Control Protocol*) y UDP (*User Datagram Protocol*), el primero garantiza la entrega del contenido hacia el usuario final, pero el segundo no garantiza la entrega de dicha información al usuario, sin embargo, posee baja latencia en su transmisión, ideal para aplicaciones de videoconferencia o video en tiempo real. Protocolos que intervienen en páginas web, como HTTP (*Hypertext Transfer Protocol*) utiliza mecanismos de transporte basados en TCP, mientras que el protocolo RSTP (*Rapid Spanning Tree Protocol*) utiliza los mecanismos de UDP para el transporte (Valdez, 2016). Resumiendo, los protocolos que realizan streaming de video, se detallan en la *Tabla 1* mostrada a continuación:

Tabla 1*Protocolos de Streaming de Video*

Protocolos UDP	Protocolos TCP
Real-Time Protocol (RTP)	HTTP Live Streaming (HLS)
Real-Time Streaming Protocol (RTSP)	Microsoft Smooth Streaming (MSS)
Real-Time Messaging Protocol (RTMP)	HTTP Dynamic Streaming (HDS)
File Delivery Over Unidirectional Transport Protocol (FLUTE)	Dynamic Adaptive Streaming over HTTP (DASH)

Nota. Tomado de *Video Streaming de Alta Calidad*, Valdez F., 2016.

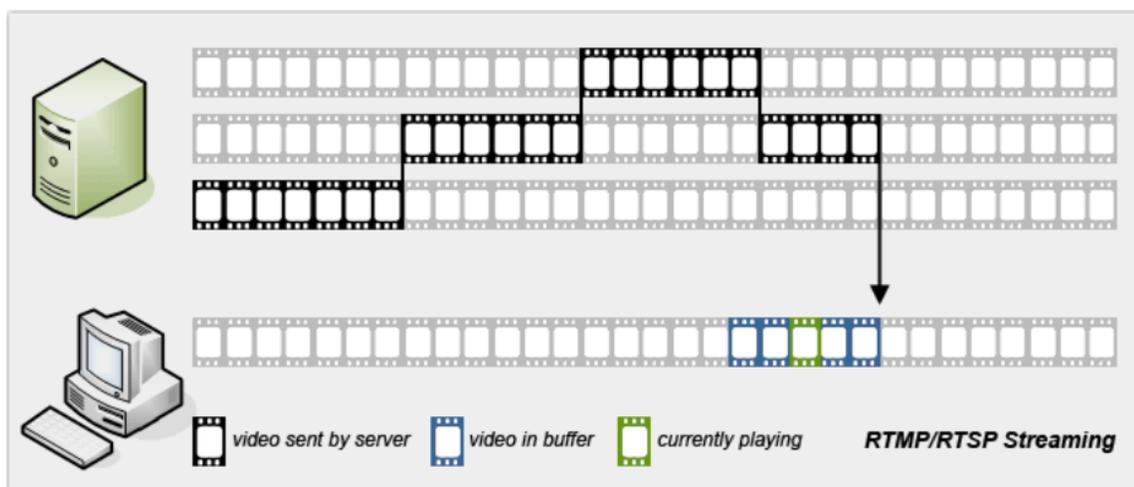
Todos estos protocolos se han desarrollado, teniendo en cuenta las aplicaciones en las que se les va a utilizar, por ejemplo, RTP es un protocolo de la capa de aplicación que proporciona funciones para la transmisión de extremo a extremo, ideal para aplicaciones en tiempo real; RTSP es una versión modificada donde se agrega cierta seguridad que RTP no tenía. RTMP permite adaptar la calidad del contenido de acuerdo con la tasa de bit que posee el usuario, mientras que FLUTE fue diseñado para dar soporte a los demás protocolos de UDP (Valdez, 2016).

Los protocolos TCP son más confiables y aseguran la entrega del contenido al usuario, en el caso de HLS que fue desarrollado por Apple Inc. en el año 2009, es utilizado en aplicaciones específicas de la marca como por ejemplo Safari, QuickTime y Apple TV, que se basa en la entrega de contenido adaptativo, lo mismo sucede con MSS que fue desarrollado por Microsoft en el año 2009 y es compatible entre dispositivos del mismo fabricante, mientras que HDS fue desarrollado por Adobe y Akamai en el año 2010, enfocándose en el streaming adaptativo con los protocolos

HTTP y RTMP. Finalmente se desarrolló DASH en el año 2012, siendo un protocolo estandarizado para resolver la incompatibilidad en la reproducción de contenidos de distintos dispositivos que tenían su propio protocolo de streaming, gracias al protocolo *Dynamic Adaptive Streaming over HTTP* fue solventado estos inconvenientes en la actualidad (Valdez, 2016). En la *Figura 5* se puede observar un gráfico que interpreta el flujo del contenido adaptado al canal del usuario.

Figura 5

Transporte del contenido adaptativo hacia el buffer del usuario



Nota. Tomado de *OTT/ABR Demystified and Monitored*, Vermulm S., 2015.

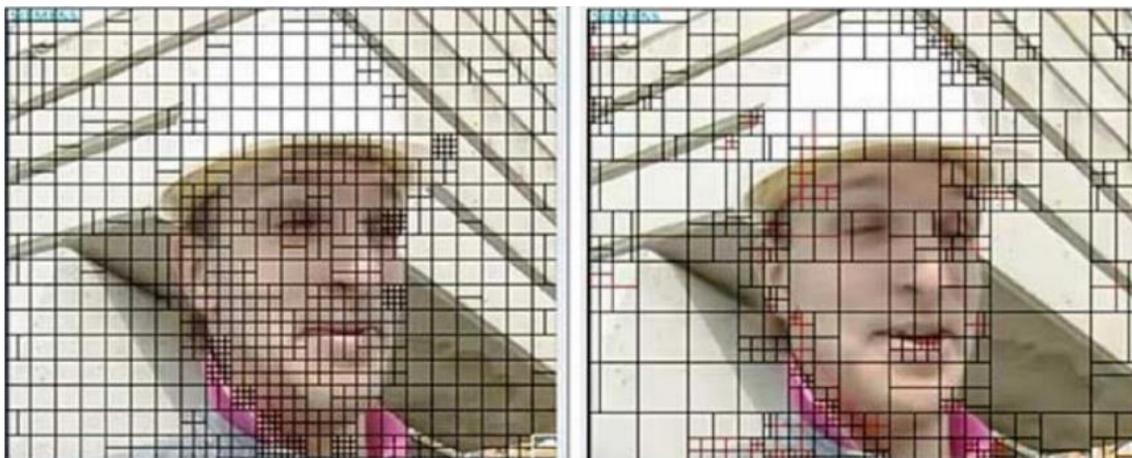
La tecnología conocida como ABR (*Adaptive Bit Rate*), contempla protocolos de streaming que permiten la transmisión de contenidos multimedia a través de Internet, ajustando la calidad de las imágenes en función de la detección en tiempo real, monitoreando constantemente el ancho de banda del canal, dependiendo también del tamaño del buffer del dispositivo terminal, siendo estos parámetros los recursos disponibles para el correcto funcionamiento (Verizon Digital Media, 2015).

2.3.2 Codificación de Contenidos

El término “contenido adaptativo” mencionado en el ítem anterior, tiene relación directa con la codificación de los contenidos, codificar hace referencia al proceso en el que un archivo (ya sea de video, audio o texto) es alterado mediante algoritmos de compresión, para reducir su tamaño sin perder información en el proceso, el software que realiza el proceso es conocido como *códec*, el cual está presente en nuestras tarjetas gráficas, tarjetas de audio y procesadores de distintos dispositivos; la decodificación es el proceso contrario, donde se extrae el archivo comprimido y se lo muestra al usuario sin alteración alguna del contenido (Marquines & Añazco, 2014).

Figura 6

Esquemas de codificación de imagen H.264/AVC vs H.265/HEVC



Nota. Tomado de *Diseño e Implementación de una Plataforma de Video Streaming OTT Multiformato*, Beltrán M. y Fernández P., Universidad Politécnica de Madrid, 2017.

La codificación y decodificación es una labor constante e intensa para los procesadores, teniendo en cuenta el tiempo que toma el proceso en función del tamaño

de los archivos a transmitir, por lo cual, se ha desarrollado una técnica denominada MBR (*Multiple Bit Rate*), que mejora considerablemente la transmisión de video en un canal, donde el ancho de banda no es fijo, entonces el contenido es codificado a velocidades distintas y el dispositivo final se encarga de negociar con el servidor de streaming la mejor tasa de bits y luego combinarlas en una sola secuencia o flujo de video para que el usuario pueda visualizarlo (Marquines & Añazco, 2014).

2.3.2.1 Compresión de Audio y Video

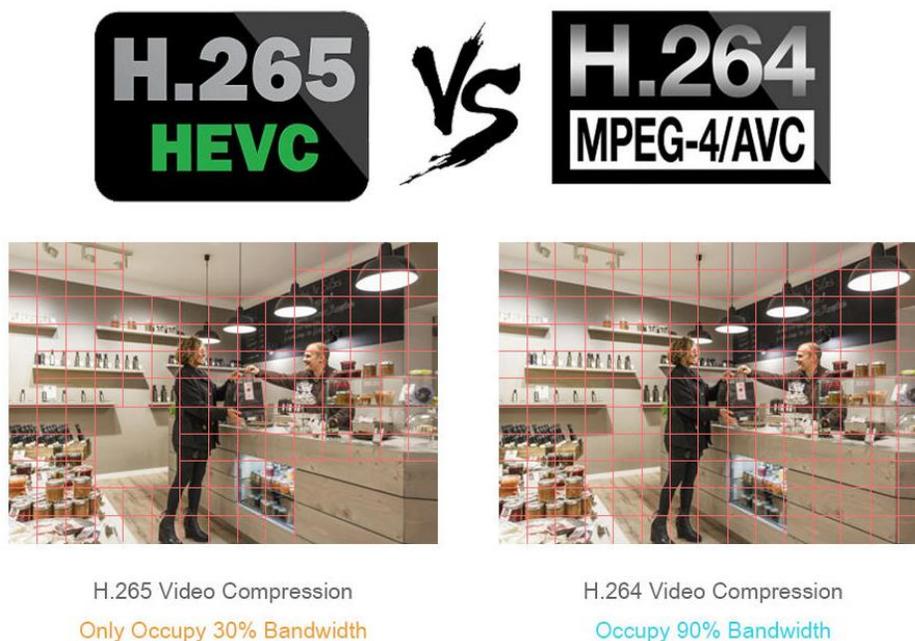
La digitalización de contenidos necesita un consumo considerable de recursos del procesador, por ejemplo, para poder transmitir un video con tamaño 720x485 a una velocidad de 30Hz por cuadro y obteniendo una resolución de 24 bits por pixel, es necesario 250Mbps aproximadamente de ancho de banda de la red de datos, y si se desea grabar una película de 2 horas, se necesitaría 225Gb (Savatier, 2013), por lo tanto es primordial el desarrollo de técnicas de compresión para minimizar tiempos y utilizar menor espacio de almacenamiento.

2.3.2.2 Códecs de Video

Con la necesidad de transformar señales de audio y video, para ser adaptadas a los distintos canales de distribución, nacen estándares que, mediante algoritmos de codificación de la señal han optimizado el ancho de banda que se requiere para que las señales lleguen a los clientes sin presentar cambios o pérdidas en dichas señales, así el cliente puede disfrutar de más contenidos a la vez, utilizando el mismo ancho de banda que dispone (Beltrán & Fernández, 2017). Los primeros estándares que aparecieron son los conocidos como MPEG-2, actualmente existen otros estándares como el H.264/MPEG-4 AVC y el H.265/HEVC.

Figura 7

Esquemas de codificación diferencial AVC y HEVC



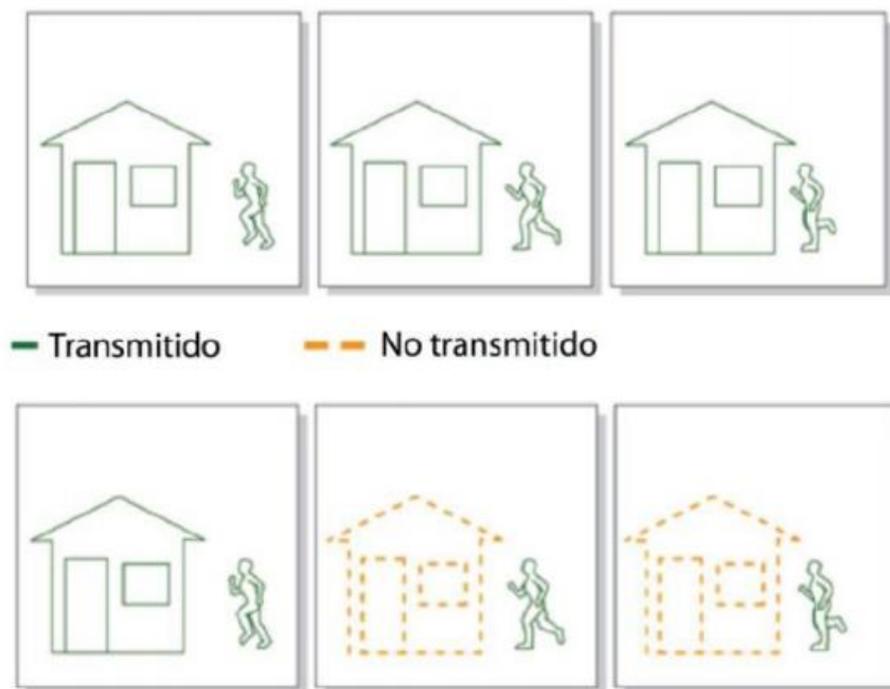
Nota. Tomado de *High Efficiency Video Coding*, ITU-T-H.265, 2015.

El estándar H.264, conocido también como MPEG-4 Parte 10/AVC para Codificación de Video Avanzado (*Advanced Video Coding*), es de código abierto y compatible con técnicas eficientes de compresión de video, que no compromete la calidad de la imagen y reduce el tamaño de un archivo de video digital en aproximadamente más de un 80% (ITU-T-H.264, 2010). Este estándar fue desarrollado por un proyecto conjunto entre expertos en imágenes en movimiento de ISO/IEC (MPEG) y expertos de codificación de video de la ITU-T. La Unión Internacional de Telecomunicaciones coordina los estándares en ese sector, ISO son las siglas de la Organización Internacional de Normalización e IEC son las siglas de la Comisión Electrónica Internacional que supervisa los estándares de las tecnologías eléctricas y electrónicas (Beltrán & Fernández, 2017).

El estándar H.265, conocido también como HEVC (*High Efficiency Video Coding*) es el sucesor del códec H.264 desarrollado conjuntamente por la ISO/IEC MPEG (*Moving Picture Group*) y la ITU-T VCEG (*Video Coding Expert Group*). Este códec fue desarrollado principalmente para soportar resoluciones que pasan del Full HD (1080 líneas verticales) a resoluciones Ultra HD 4K (2160 líneas verticales) y hasta el 8K (8192x4320) siendo un 50% más eficiente que el códec anterior (ITU-T-H.265, 2015). La diferencia en la codificación de los primeros estándares (como por ejemplo el *Motion JPEG*) con los dos estándares mencionados, son los procesos de optimización y compresión que se utilizan, observe la *Figura 8* para poder entender el ejemplo.

Figura 8

Diferencias entre codificación única y codificación diferencial



Nota. Tomado de *Diseño e Implementación de una Plataforma de Video Streaming OTT Multiformato*, Beltrán M. y Fernández P., Universidad Politécnica de Madrid, 2017.

Con el formato Motion JPEG, se codifican y se envían las tres imágenes de la secuencia de arriba de la *Figura 8* como imágenes separadas y únicas (conocidos como fotogramas I, que no existe dependencia entre ellas). Mientras que en los formatos actuales como el H.264 y el H.265, se realiza una codificación diferencial, donde sólo se codifica completamente la primera imagen (fotograma I), y en las siguientes dos imágenes (conocidas como fotogramas P), existe cierta referencia a la primera imagen en lo que se refiere a elementos estáticos (como lo es la casa en las tres imágenes), entonces la codificación solo se realiza a los elementos móviles (como es el hombre que corre hacia la casa), usando vectores de movimiento y reduciendo así la cantidad de información que se almacena y se envía por el canal (Beltrán & Fernández, 2017).

2.3.2.3 Grupo de Imágenes (GOP)

Para entender el proceso de adaptación de los contenidos de video, hay que diferenciar la funcionalidad que tiene los tres tipos de imágenes en un video conocidos como fotogramas:

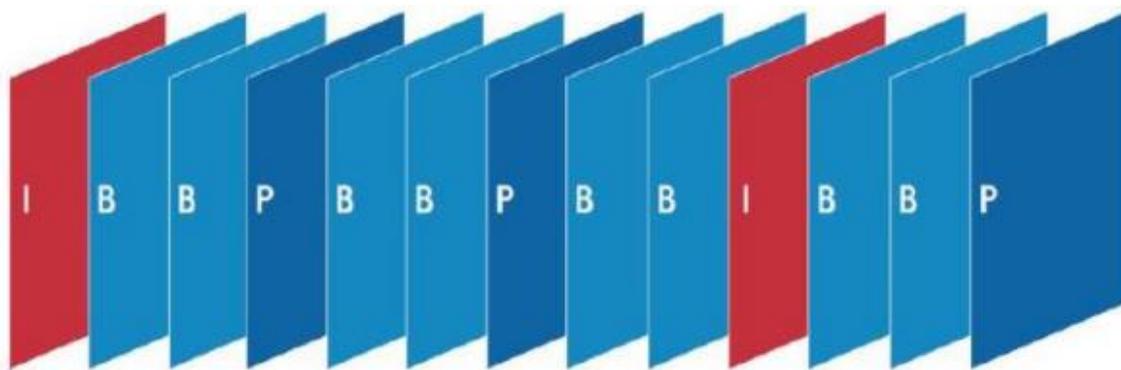
Fotogramas I (*Intraframes*) o Intra Cuadros, son imágenes independientes que pueden ser codificadas completamente, sin tener referencia alguna con otras imágenes. Este fotograma siempre es el primero en una secuencia de video y sirve como puntos iniciales a nuevas visualizaciones de video, posibilitando una resincronización en la transmisión en el caso de presentar errores de transmisión o por interferencias. Estos fotogramas también sirven para brindar características de control como avance y retroceso en el video, además, no generan demasiados errores o defectos en la visualización del video, pero es necesario una mayor cantidad de bits para poder codificarlos (Beltrán & Fernández, 2017).

Fotogramas P (*Predictive*) o Cuadros Predictivos, son imágenes que hacen referencia a características de fotogramas I o P anteriores para poder ser codificados, por esta razón, estos fotogramas requieren menor cantidad de bits a diferencia de los I, pero presentan mayor desventaja al visualizar los contenidos, es decir son sensibles a errores por ser dependientes de los I o P anteriores (Beltrán & Fernández, 2017).

Fotogramas B (*Bi-Directional*) o Cuadros Bidireccionales, son imágenes que hacen referencia a fotogramas anteriores como posteriores, por esta razón tiene el nombre de bidireccional, y en estos fotogramas es donde mayor cantidad de errores puede presentarse (Beltrán & Fernández, 2017). En la *Figura 9* se puede observar una secuencia típica de los fotogramas al momento de codificar un video:

Figura 9

Secuencia típica de fotogramas o GOP



Nota. Tomado de *Diseño e Implementación de una Plataforma de Video Streaming OTT Multiformato*, Beltrán M. y Fernández P., Universidad Politécnica de Madrid, 2017.

Al conjunto de fotogramas I, B y P se le conoce como GOP (*Group of Pictures*) o también es llamado como *Chunk* o Segmento, donde un fotograma P sólo puede hacer referencia a fotogramas I o P anteriores, y un fotograma B puede hacer referencia a

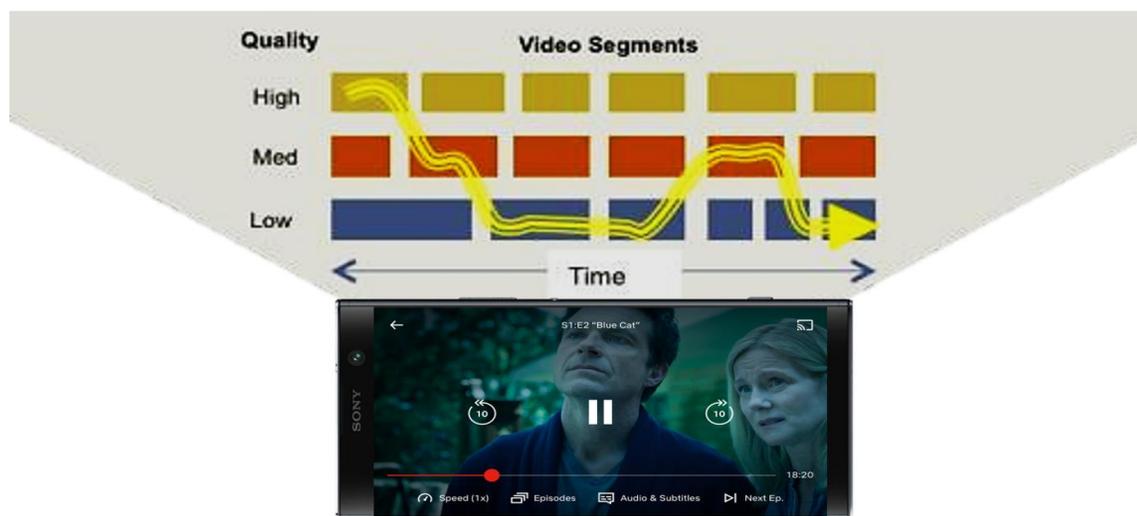
fotogramas I o P anteriores o posteriores, codificadas por sí mismas (Valdez, 2016). El proceso de codificación diferencial que realizan los códecs de compresión H.264 y H.265 actualmente, es tomar un fotograma y compararlo con otro fotograma de referencia (como por ejemplo un fotograma I o P anterior), y solo se codificará a los píxeles que han cambiado respecto al de referencia, reduciendo la cantidad de píxeles a codificar y transmitir por un canal (Beltrán & Fernández, 2017).

2.3.3 Streaming Adaptativo

El servicio de streaming adaptativo OTT está basado en técnicas de codificación de video, y está regido a una escalera de codificación, conocida como “*Encoding Ladder*”, que consiste en una cantidad determinada de perfiles de codificación, regidos a una resolución distinta y un bit rate adecuado (Beltrán & Fernández, 2017).

Figura 10

Cambios de perfil de calidad en un streaming adaptativo



Nota. Tomado de *Diseño e Implementación de una Plataforma de Video Streaming OTT Multiformato*, Beltrán M. y Fernández P., Universidad Politécnica de Madrid, 2017.

En la *Figura 10* se puede observar cómo va cambiando automáticamente la resolución del video en función del tiempo, esto es posible ya que, al disponer varios perfiles de video codificados con distintas resoluciones, el reproductor de video en el dispositivo de los usuarios se encarga de elegir y negociar con el servidor la calidad de video que debe descargar, de acuerdo con las condiciones del ancho de banda disponible en cada instante, en resumen, el reproductor elegirá el perfil más alto e irá subiendo o bajando la calidad del video en función de las condiciones del canal a medida que se va reproduciendo el contenido en cada instante de tiempo.

2.3.3.1 Sincronización del Streaming Adaptativo

La secuencia unificada de los distintos segmentos que se va descargando progresivamente de cada perfil de calidad, da origen a un archivo del tipo SMIL (*Synchronized Multimedia Integration Language*) o Lenguaje Sincronizado de Integración Multimedia, que es un estándar abierto el cual define un lenguaje coordinado y de sincronismo directamente con el reproductor. Entonces el estándar es quien especifica una secuencia para la reproducción del contenido y va creando un archivo del tipo *.smil* a medida que se va reproduciendo el video, siempre y cuando el reproductor sea compatible con SMIL (Marquines & Añazco, 2014).

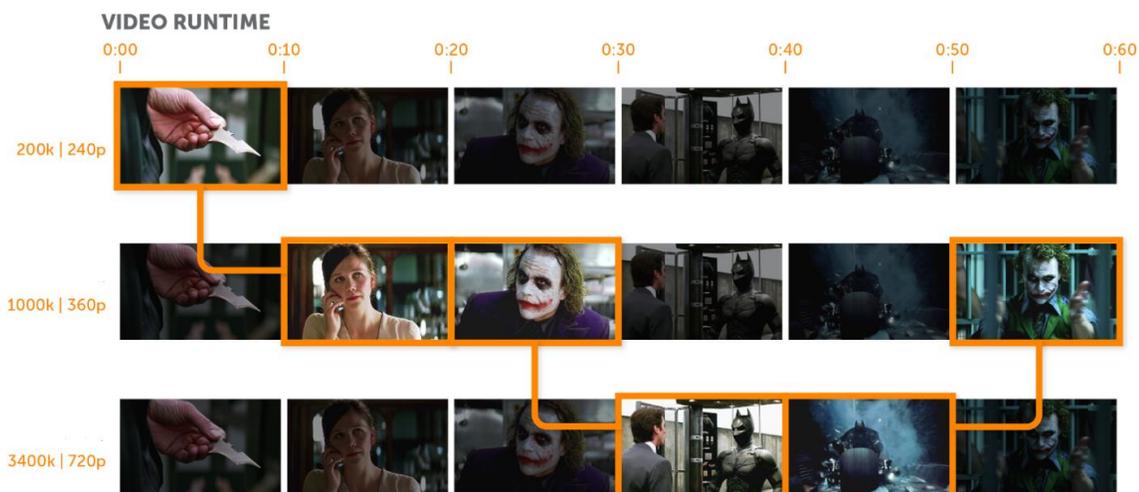
El archivo SMIL es considerado como un meta-archivo, y es quien define características a nivel de sesión para establecer la conexión hacia el usuario, además está ligado hacia los botones de control del reproductor, dándole al usuario una interacción directa con el contenido visualizado, así podrá retroceder o adelantar y tener un correcto sincronismo del audio, video y los subtítulos de este. Sin embargo, existe también un fichero índice llamado *Manifest*, quien se encargada de dirigir al reproductor

y apuntar hacia el perfil más adecuado para el canal, subiendo y bajando la calidad si es necesario o mantenerlo en la misma calidad (Beltrán & Fernández, 2017).

En la *Figura 11* se puede visualizar la dinámica entre los distintos perfiles del video codificado, por defecto, los segmentos son de tamaño variable, pero es posible configurar de manera avanzada el códec para establecer un tamaño fijo de GOP.

Figura 11

Cambios de perfil de calidad y almacenamiento en el buffer



Nota. Tomado de *Análisis y diseño de una plataforma para brindar servicios de video multipantalla OTT en una empresa de telecomunicaciones*, Naranjo J. G., PUCE, 2016.

Los segmentos del video se van descargando uno a uno, hasta que el buffer del dispositivo se llene a su máxima capacidad, y a medida que el contenido se reproduce, se va liberando espacio en el buffer e inmediatamente se descarga el siguiente segmento, así funciona la lógica de los protocolos de ABR, permitiéndole al reproductor adaptarse a las condiciones de la red de cada usuario, eligiendo el perfil más adecuado y mejorando enormemente la experiencia de los usuarios.

2.4 Componentes de una Plataforma de Streaming OTT

Conociendo algunos términos y características importantes que definen a un servicio OTT como tal, es necesario determinar qué componentes intervienen en el proceso, desde el servidor de streaming hasta el dispositivo del usuario final.

2.4.1 Ingesta de Contenidos CMS (Content Management System)

Siendo esta la etapa inicial, donde se alojan todos los videos que el usuario puede visualizar desde sus dispositivos, apegándose a la definición de “streaming”, donde el contenido está alojado en un servidor en Internet, el cual envía el contenido mediante una tecnología conocida como “descarga progresiva” que, a medida que el contenido va llegando al dispositivo, se lo reproduce, pero no necesariamente el video descargado crea una copia del mismo en el dispositivo del usuario, más bien se va eliminando con forme se reproduce, para seguir recibiendo más segmentos de video.

Figura 12

Empresas que brindan servicios en la nube



Nota. Tomado de *IPTV Ventajas y Desventajas*, Xataca Basics Webedia, 2019.

Tener un centro de datos propios que almacene todos los contenidos de video puede ser sumamente costoso, por lo cual, se han ido migrando los servidores a la nube, como es el caso de Netflix, que aloja sus contenidos de video en AWS (*Amazon Web Services*), gracias a la flexibilidad de la nube, ahorra costos de mantenimiento y Netflix tiene la facilidad para poder agregar más servidores cuando sea necesario o devolverlos cuando no se los use, dependiendo de la demanda de usuarios que posea; algo que no sería tan fácil al tener un centro de datos propio, ya que los servidores extra que fueron adquiridos para atender picos de demanda de usuarios, quedarían parados y sin funcionar cuando esa demanda disminuya (Xataca Streaming, 2018).

Figura 13

Servidores de Netflix repartidos en carriers de todo el mundo



Nota. Tomado de *La compleja infraestructura detrás de Netflix*, Xataca Streaming, 2018.

Debido a la gran demanda de usuarios que posee Netflix, al momento de atender cada solicitud de los usuarios de todo el mundo, teniendo una sola ubicación de los contenidos de la plataforma, se empezaría a generar retardos que molestarían a los usuarios; es más eficaz y eficiente tener varias réplicas de los contenidos en servidores repartidos por todo el mundo, tal como se observa en la *Figura 13*, ya que al tener el contenido más cerca del usuario, es mucho más rápido acceder a estos, y así evitar inundación innecesaria de tráfico en la red mundial de Internet.

Para el desarrollo de este proyecto de investigación, se pretende limitar el alcance del servicio a una red LAN, esto significaría que la demanda de usuarios no pretende ser tan numerosa, alojando todos los contenidos de una manera local en un PC que se convertirá en el servidor de la plataforma.

2.4.2 Codificación de los Contenidos

Para que la plataforma sea considerada como un servicio OTT, es estrictamente necesario disponer de los contenidos en todos los formatos y calidades de video existentes, Netflix por ejemplo, tiene el propósito de permitir el acceso a sus contenidos desde cualquier dispositivo mediante el Internet, teniendo aproximadamente 2200 formatos de cada archivo para distintos dispositivos, además deben crearse los archivos comprimidos a distintos perfiles de calidad y ponerlos a disposición de los usuarios, ofreciendo la posibilidad de adaptar la calidad de los videos en función de las condiciones del canal que cada usuario posea, teniendo en cuenta la gran cantidad de usuarios que acceden a su plataforma de manera simultánea, por lo cual debe transmitir los contenidos de manera ininterrumpida y en su mejor calidad posible, en varios idiomas y adicional pequeños tráileres o escenas del video (Xataca Streaming, 2018).

Figura 14

Miles de archivos en los servidores de Netflix



Nota. Tomado de *La compleja infraestructura detrás de Netflix*, Xataca Streaming, 2018.

Existe una gran variedad de herramientas de software y hardware para codificar y decodificar contenidos, herramientas de libre acceso que se han vuelto muy populares al día de hoy, con el fin de abaratar costos de inversión, teniendo fines educativos o para el desarrollo de proyectos de investigación, como es el caso de FFMPEG, que es una herramienta de conversión y codificación muy potente, un software de libre distribución muy completo, con librerías que nos facilitan la tarea de codificar un contenido de video (FFMPEG, 2017).

Para este proyecto de investigación, se utilizará FFMPEG como software libre, con el propósito de impulsar el desarrollo y aportar a una gran comunidad de expertos del procesamiento de video, obteniendo contenidos procesados y alojados en el servidor local, disponiendo de los distintos perfiles de codificación al momento de observar un video en la plataforma implementada.

Figura 15

Software FFMPEG de código abierto



Nota. Tomado de *FFMPEG World Wide Web Documentation*, FFMPEG, 2017.

FFMPEG fue un proyecto desarrollado en lenguaje C y *Assembler* a manos de *Fabrice Bellar*, quien le dio un enfoque *Open Source*, posibilitando a que cualquier desarrollador pueda utilizar su herramienta de codificación, modificarlo o añadirle más librerías si fuera necesario, en fin, mejorar su proyecto a través de los aportes de una gran comunidad de expertos del procesamiento de video alrededor del mundo (FFMPEG, 2017).

Para plataformas de streaming OTT de uso masivo, teniendo alojados sus contenidos en la nube, utilizan herramientas de codificación mucho más sofisticados y potentes, capaces de trabajar en paralelo para disminuir tiempos y recursos, pero sus costos de procesamiento son muy elevados; existen grandes empresas dedicadas a la codificación de contenidos como lo es *Expression Encoder* (perteneciente a Microsoft), *Media Coder* (perteneciente a Adobe), *Harmonic* (perteneciente a Ericsson), entre otras (Beltrán & Fernández, 2017).

2.4.3 Paquetización y Distribución o CDN (Content Delivery Network)

Después de codificar los contenidos, comprimirlos y establecer los perfiles de codificación en las distintas calidades posibles, estos archivos deben ser paquetizados dando con forme a los distintos protocolos de streaming vigentes. En el presente proyecto, llevando a cabo esta función por la herramienta *Wowza*, siendo *Open Source* ha sido muy utilizado hoy en día tanto a nivel particular y educativo, como a nivel profesional y empresarial (Beltrán & Fernández, 2017).

Figura 16

Motor de Streaming Wowza, de Wowza Media Systems



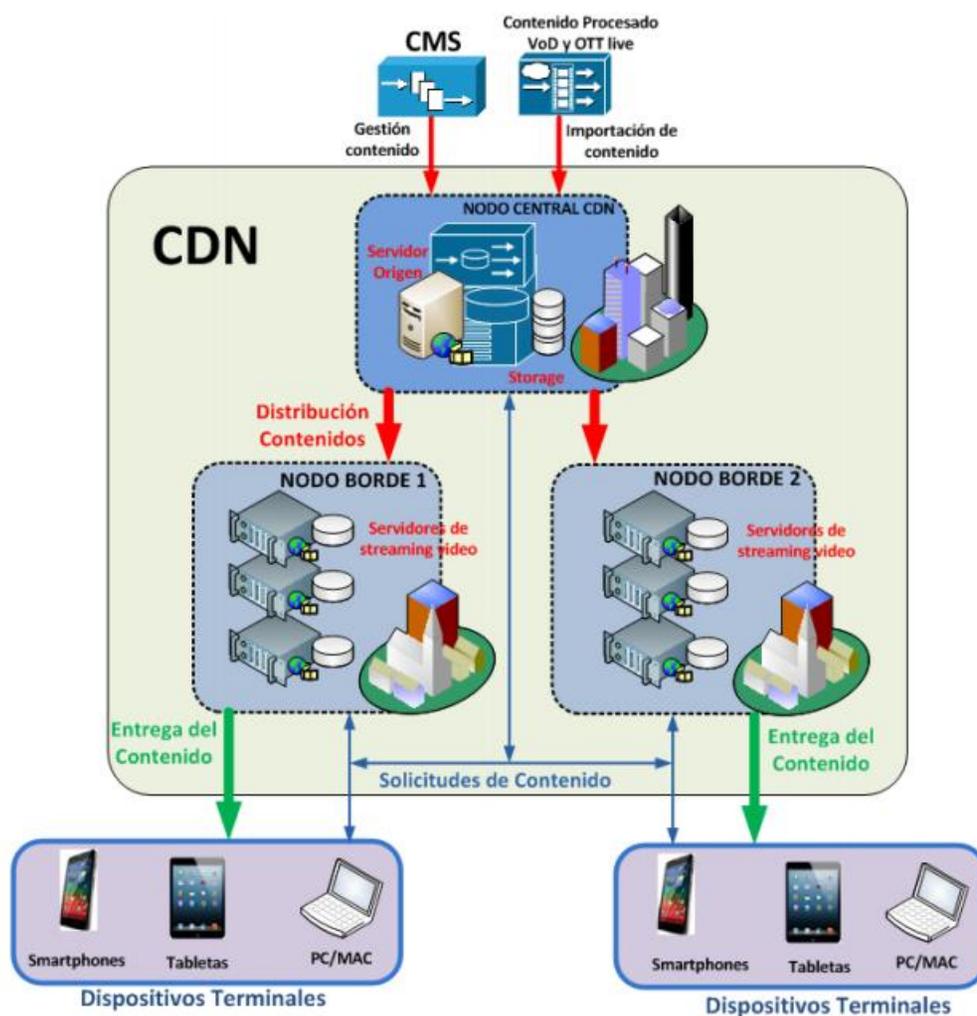
Nota. Tomado de *Wowza Streaming Engine Documentation*, Wowza, 2020.

Wowza es un software desarrollado por *Wowza Media Systems*, siendo una herramienta de gran utilidad que brinda servicios de streaming de Video Bajo Demanda (VoD) y Video Live. Esta herramienta utiliza los distintos protocolos de streaming adaptativo vigentes, como lo es HDS, MSS, HLS y DASH, brindándonos una herramienta configurable y centralizada para realizar streaming, además de proporcionar parámetros y métricas del uso y distribución (Wowza, 2020).

Mediante la herramienta encargada de la paquetización, también realizará la difusión, gracias a los beneficios que Wowza proporciona, siendo una herramienta completa que unifica diversos protocolos de streaming ABR y realiza la paquetización del contenido en tiempo real.

Figura 17

Esquema de distribución CDN a los usuarios



Nota. Tomado de *Análisis y diseño de una plataforma para brindar servicios de video multipantalla OTT en una empresa de telecomunicaciones*, Naranjo J. G., PUCE, 2016.

El presente proyecto, al ser una plataforma implementada en red LAN, no necesitamos de una red de distribución conocida como CDN (*Content Delivery Network*), estas redes se encargan de la difusión de los contenidos, además de que mejoran enormemente la QoE de los usuarios, ya que mediante mecanismos de redundancia entre servidores en todo el mundo, entrega el contenido con baja latencia y sin mayor impacto de tráfico en las redes de Internet (Naranjo, 2016).

2.4.4 Encriptación del Contenido o DRM (*Digital Right Management*)

Para proteger derechos de autor y evitar la piratería, el DRM restringe el libre acceso a los contenidos que se difundan en esta plataforma, controlando el acceso solo a los subscriptores.

Esta etapa tiene una integración directa desde los servidores de licencias hacia los dispositivos de los usuarios, utilizando mecanismos de encriptación, se suministran claves asociadas a los contenidos, las cuales tienen un tiempo de caducidad, para evitar inicios de sesión no autorizados y se pueda proteger el contenido que se está reproduciendo (Naranjo, 2016).

En este proyecto, no se realiza la etapa de encriptación, ya que al montarse en una red LAN, la meta principal es levantar el servicio correctamente y realizar pruebas correspondientes al funcionamiento sin ningún tipo de restricciones, entonces esta etapa queda como un trabajo a futuro sobre la misma plataforma implementada; sin embargo, el sistema de distribución Wowza tiene compatibilidad con distintas herramientas de encriptación como BuyDRM, VeriMatrix y EzDRM.

2.4.5 Reproducción del Contenido (Dispositivos de Usuario Final)

La forma de presentar los contenidos y reproducirlos desde cualquier dispositivo del usuario, tiene un alto impacto comercial e influye en la experiencia del usuario QoE, ya que aquí es donde se genera el interés de los usuarios, para mantenerlos suscritos al servicio y adicional, atraer más posibles suscriptores del servicio OTT ofrecido, siempre y cuando el servicio a ofrecer sea bueno, no presente problemas al establecer conexión y tenga la capacidad de realizar streaming adaptativo sin interrupción alguna. En este proyecto se utilizan dos reproductores:

El reproductor *VLC Media Player*, es una herramienta multiplataforma gratuita y de código abierto, desarrollado en 1996 por el proyecto académico “*VideoLAN*” de *École Centrale Paris* en Francia, hoy en día es compatible con la mayoría de sistemas operativos y puede reproducir casi cualquier formato de archivos multimedia, sin la necesidad de instalar códecs externos y tiene la capacidad de realizar streaming (VideoLAN Organization, 2020). Siendo la herramienta más adecuada y sencilla de utilizar para realizar las pruebas del servicio, desde una PC o dispositivo móvil.

Figura 18

Reproductor de código abierto VLC Media Player



Nota. Tomado de *VLC Media Player*, Video LAN Organization, 2020.

El reproductor *JW Player*, es una herramienta *responsive* gratuita y de pago, es decir, el reproductor puede ser integrado a una página web y tiene la característica de adaptarse al tamaño de la pantalla de los dispositivos móviles. La versión no comercial gratuita, permite utilizar el reproductor HTML5 sin tener acceso a la plataforma de soporte, brindando la posibilidad de implementarlo en cualquier proyecto donde se lo requiera y se adapte a cualquier pantalla, que soporte protocolos como DASH o HLS, y brinde la posibilidad de ajustar de manera automática o manual la calidad de los contenidos de acuerdo al ancho de banda del usuario final (Ortiz, 2017).

Figura 19

Reproductor profesional HTML5 JW Player

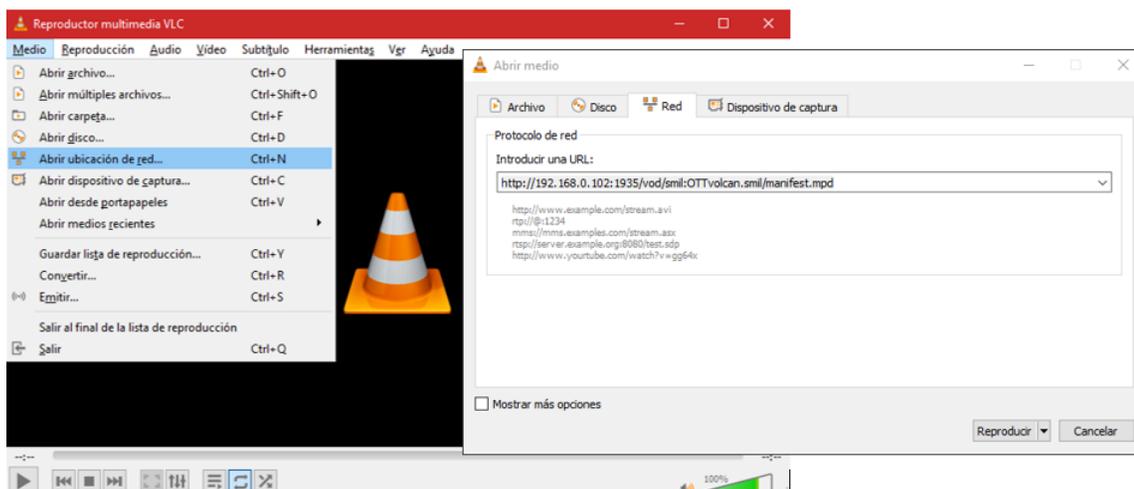


Nota. Tomado de *JW Player para proyectos profesionales*, Ortiz J., 2017.

En los objetivos de este proyecto de investigación, no se contempla el desarrollo de una página web para el acceso a la plataforma implementada, pero al ser muy importante la manera en que los contenidos se muestran al usuario, se plantea el desarrollo de una plataforma web con contenido bajo demanda, la cual nos muestra un catálogo completo de todos los videos cargados al servidor de streaming, algo que no se puede realizar con el reproductor VLC, siendo un software de escritorio, solo se reproduciría el contenido al que hace referencia la URL que ingresemos, tal como se observa la *Figura 20*, por esta razón se utiliza dos tipos de reproductores en el proyecto.

Figura 20

VLC accede al contenido mediante una dirección IP



Nota. Tomado de *VLC Media Player*, Video LAN Organization, 2020.

El lenguaje de programación HTML5 (*HyperText Markup Language*, versión 5), siendo la quinta versión en desarrollo del lenguaje, define ciertos estándares para desarrollar y diseñar el código, resolviendo problemas y adaptándolo a nuevas necesidades que el contenido multimedia (video, audio, entre otros) de gran calidad posee, pudiendo codificar y decodificar los contenidos sin presentar errores en su presentación web y en cualquier pantalla (Garro, 2014).

Capítulo III

3. Implementación de la Plataforma OTT

3.1 Descripción General del Proyecto de Investigación

En esta etapa del proyecto de investigación, se implementa una plataforma de video bajo demanda (VoD) con todas las características que un servicio de streaming OTT debe poseer, y así prestar el servicio a distintos usuarios y sus dispositivos en una red LAN, a la cual se redujo el escenario para implementar este proyecto. Se utiliza herramientas de software libre para conformar cada etapa de la plataforma y en la *Figura 21* se muestra el diagrama de bloques para comprender el proceso que realiza esta plataforma para prestar sus servicios de streaming VoD.

Figura 21

Diagrama de bloques de la Plataforma OTT implementada



Al ser implementada una plataforma OTT a menor escala, en la etapa de ingesta y codificación de contenidos, se reemplaza los codificadores físicos y robustos, por el software FFMPEG, el cual se encargará de codificar y comprimir los videos en distintas

calidades, los cuales serán alojados de forma local y cumplirá la función que realiza el CMS (*Content Management System*). En la paquetización y distribución, el software Wowza cumplirá esta función y reemplazará a la red de distribución CDN (*Content Delivery Network*), siendo este el motor principal de la plataforma.

Sin embargo, para la reproducción del contenido, se llega de dos modos a los usuarios: un modo de reproducir el contenido (no tan adecuada) es mediante el software VLC instalado en los PC's de cada usuario o en sus dispositivos móviles, donde el usuario debe conocer con exactitud la dirección URL del contenido que desea visualizar y así empezar con el streaming. Para el segundo modo (siendo éste el más apropiado), se implementa una pequeña página web, en lenguaje de programación HTML5 con una interfaz amigable, que se adapte al tamaño de pantalla que disponga el usuario, en donde se integra JW Player, siendo un reproductor de video que entiende protocolos de streaming adaptativo, el cual se convierte en el más apto para este proyecto.

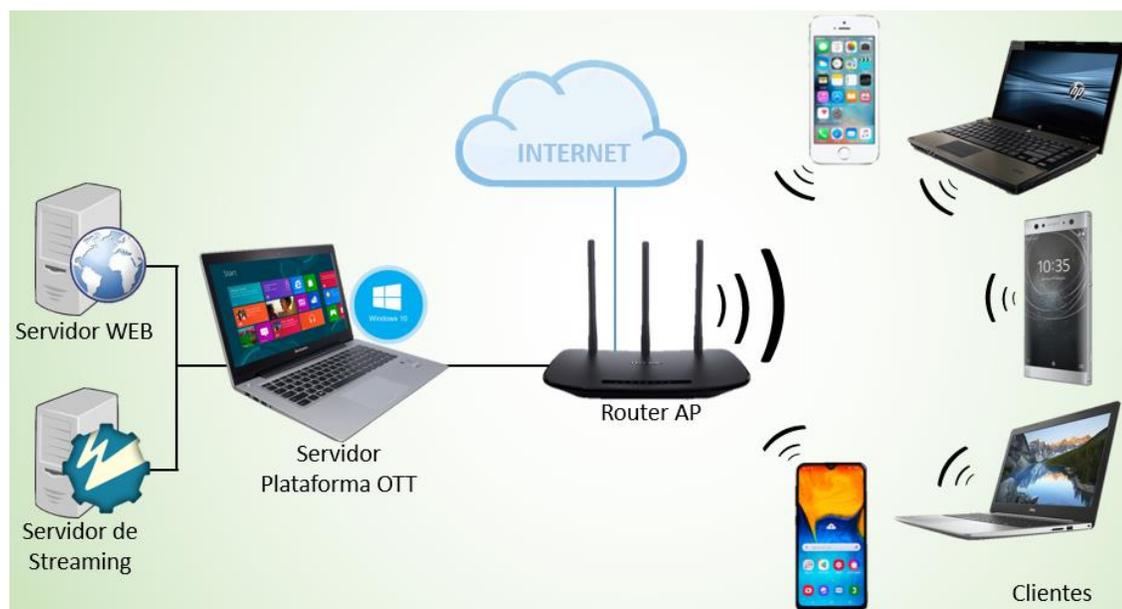
Para compilar la página web diseñada para esta plataforma, se utiliza XAMPP, que es un paquete gratuito de múltiples herramientas *open source*, el cual dispone de un servidor web multiplataforma conocido como Apache, quien atenderá las peticiones de cada usuario.

3.2 Hardware Utilizado

El escenario de implementación es una red LAN, cumpliendo con una arquitectura cliente-servidor tal como se muestra en la *Figura 22*, donde el único dispositivo de hardware adicional es el router que interconecta a los usuarios en la red.

Figura 22

Diagrama físico de red utilizado para la implementación



Para el servidor de streaming se utiliza una laptop Lenovo IdeaPad U430 Touch, en donde se alojan los videos codificados mediante el software FFMPEG en distintas calidades, además este dispositivo corre el servicio de *Wowza Streaming Engine*, el cual se encarga de la difusión y adaptación de los contenidos.

Adicional, debido a que se implementa una página web para visualizar la plataforma OTT desarrollada en este proyecto, se utiliza la misma laptop para correr el servidor web Apache, para que los usuarios puedan acceder de una forma más amigable a la plataforma dentro de la red.

En la *Tabla 2* se detallan las características técnicas que posee la laptop utilizada como servidor, que serán consideradas como características mínimas para el funcionamiento de esta plataforma.

Tabla 2*Especificaciones Técnicas de la laptop servidor*

Lenovo IdeaPad U430 Touch	
Procesador	Intel® Core i5-4200U CPU @ 1.60 GHz 2.30 GHz
Sistema Operativo	Windows 10 Pro x64
Gráficos	Intel® HD Graphics 4000
Pantalla	Full HD (1960x1080) Multitáctil de 14" y de 10 puntos
Memoria RAM	4.0 GB
Disco Duro	Hibrido SSHD 500GB 5400 RPM
Conectividad WiFi	IEEE 802.11 b/g/n
Conectividad LAN	LAN 10/10/1000 Mbps

Nota. Tomado de *IdeaPad U430 Touch Ultrabook*, Lenovo, 2020.

Para interconectar los usuarios hacia el servidor, se utilizó un router TP-Link de tres antenas, cuyas características técnicas se detallan en la *Tabla 3* a continuación.

Tabla 3*Especificaciones Técnicas del router*

TP - Link TL-WR940N	
Interface	4 puertos LAN 10/100Mbps y 1 puerto WAN10/100Mbps
Estándares Inalámbricos	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
Antena	3 Antenas desmontables omnidireccionales de 5 dBi
Frecuencia	2.4 - 2.4835 GHz
Tasa de Señal	11n: hasta 450Mbps (dinámica), 11g: hasta 54Mbps (dinámica), 11b: hasta 11Mbps (dinámica)

Nota. Tomado de *TP-link TL.WR940N*, TP-Link Technologies Co. Ltda., 2020.

Finalmente, los dispositivos de los usuarios para realizar las pruebas de funcionamiento de la plataforma son aquellos dispositivos convencionales y de uso diario, como laptops y smartphones, los cuales disponen conexión WiFi hacia el router, garantizando el acceso al servidor de streaming y a la página web de la plataforma.

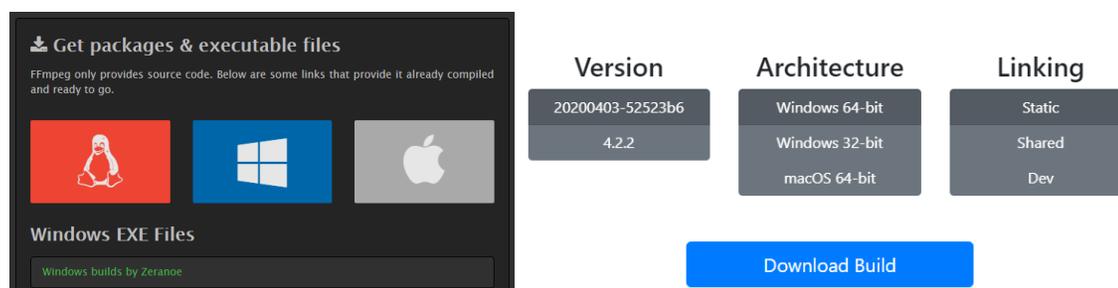
3.3 Ingesta y Codificación de Contenidos

3.3.1 Instalación del Software FFMPEG

Para instalar este software, es necesario seguir un cuidadoso proceso, debido a que no dispone de una interfaz amigable con el usuario, sino más bien se lo utiliza mediante línea de comandos. Primero hay que dirigirse a la página oficial de descarga <http://ffmpeg.org/download.html> y seleccionar el paquete acorde al sistema operativo que se dispone, tal como se observa en la *Figura 23* mostrada a continuación, en este caso se utiliza Windows 10 de 64 bits.

Figura 23

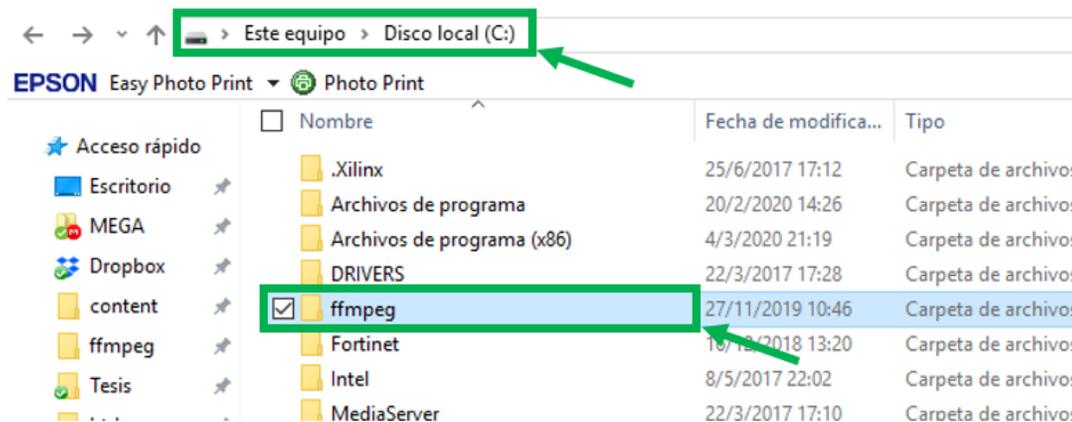
Selección del paquete de descarga de FFMPEG



Se debe crear una nueva carpeta llamada "ffmpeg" en la raíz del disco local C, tal como se muestra en la *Figura 24*, la cual servirá para alojar los archivos que contiene el paquete descargado.

Figura 24

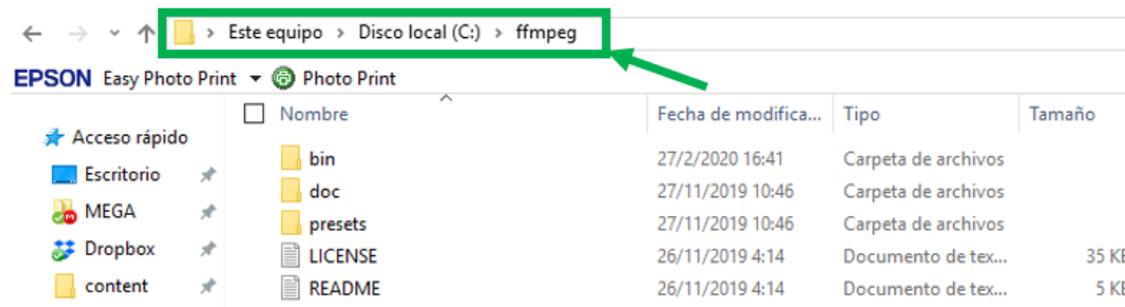
Directorio de la carpeta creada para alojar FFMPEG



Se descomprime el archivo .zip descargado y los archivos se los debe colocar en la carpeta “ffmpeg” que fue creada, tal como muestra en la *Figura 25*.

Figura 25

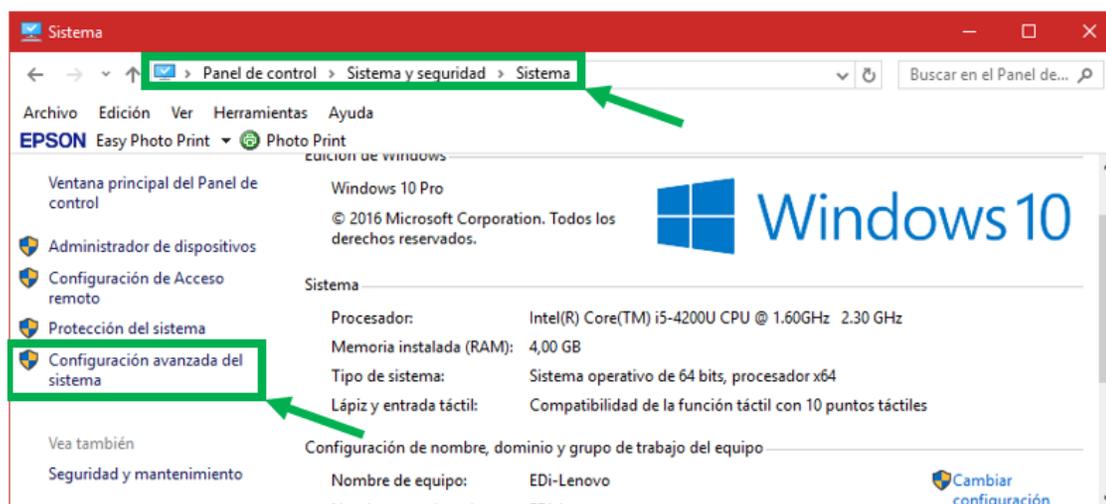
Archivos extraídos en la carpeta creada para alojar FFMPEG



Para habilitar el software, hay que dirigirse al “Panel de Control”, elegir “Sistema y Seguridad” y seleccionar “Sistema”, para finalmente dirigirse a “Configuración Avanzada del Sistema” donde se desplegará una ventana denominada “Propiedades del sistema”, tal como se observa en la *Figura 26* a continuación.

Figura 26

Configuración del software FFMPEG mediante el Panel de Control



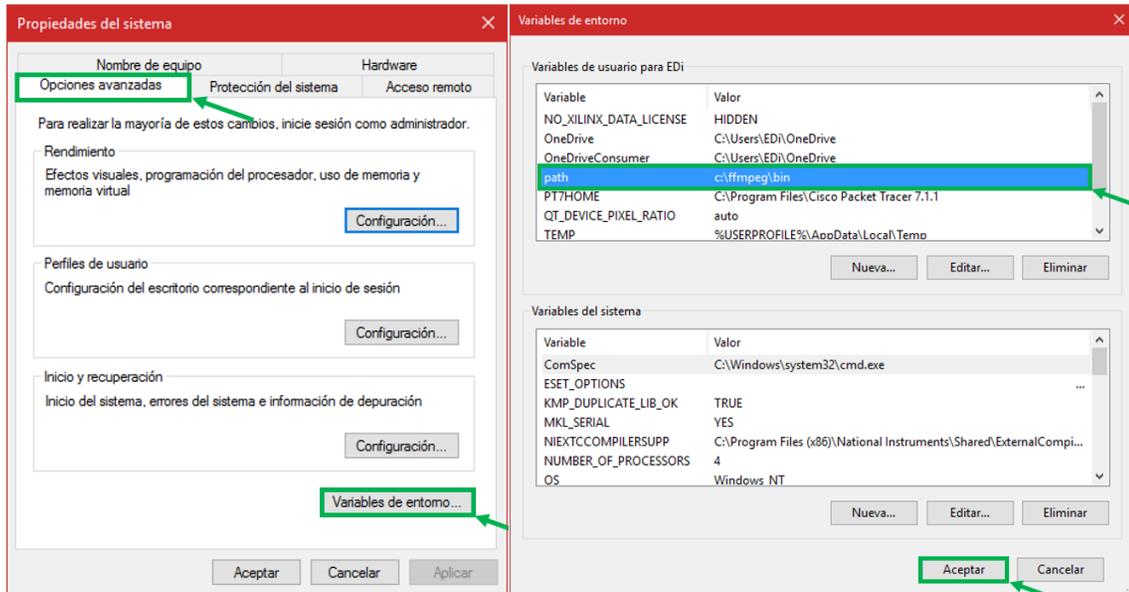
En la *Figura 27* se puede observar la ventana desplegada, en donde se debe dirigirse a la pestaña “Opciones Avanzadas” y ahí seleccionar la opción de “Variables de Entorno”, la cual desplegará una nueva ventana en donde se debe editar la variable de usuario “path” en la primera sección de la ventana, aquí es donde se debe escribir la ubicación de los archivos de ffmpeg donde fueron alojados en el disco local, en este caso la dirección es la siguiente: “C:\ffmpeg\bin”.

Si existen más usuarios creados en el ordenador, es preferible editar la variable “path” en la segunda sección de la ventana desplegada, con el fin de evitar problemas en el funcionamiento del software, en el instante en que otro usuario del ordenador desee utilizar esta herramienta.

También es posible acceder a la consola del sistema de Windows como administrador, en el caso en que el software presente inconveniente alguno en su funcionamiento pese a la configuración mencionada en la *Figura 26*.

Figura 27

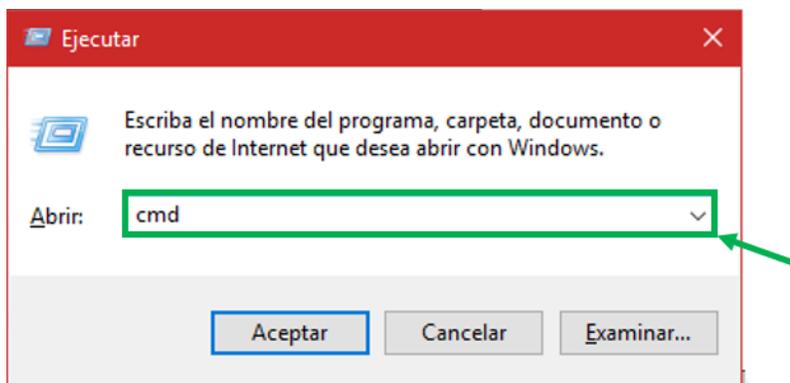
Configuración de la variable PATH con la ubicación de FFMPEG



Puede pasar que la variable “path” no exista, por lo cual se debe seleccionar “Nueva” y crearla manualmente. Finalmente, este proceso habilitará el software de ffmpeg en el ordenador.

Figura 28

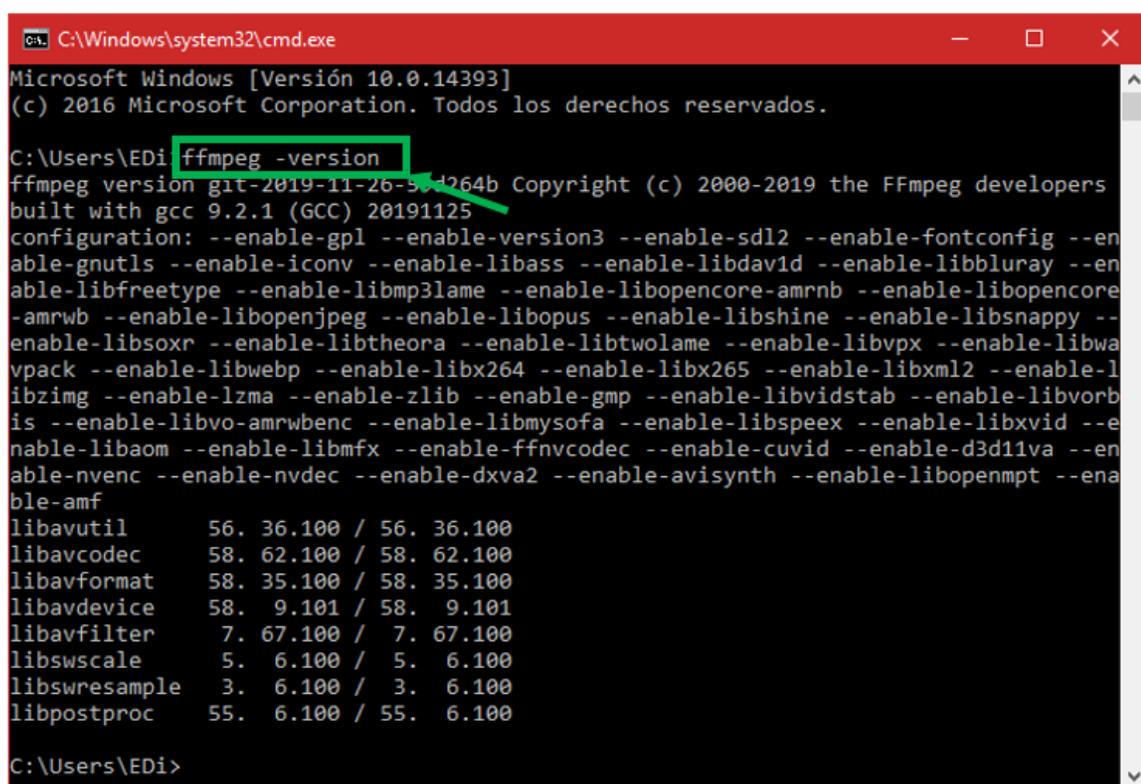
Ejecución del cmd



Para comprobar que el programa está funcionando correctamente, se debe dar clic derecho en el botón “Inicio” y seleccionar la opción “Ejecutar”, donde al escribir “cmd” se desplegará la consola de Windows para ingresar el comando “ffmpeg -version” el cual mostrará la versión de la compilación del programa FFMPEG que se dispone, similar a la ventana mostrada en la *Figura 29*.

Figura 29

Resultado del comando ingresado para verificar la versión de FFMPEG



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.14393]
(c) 2016 Microsoft Corporation. Todos los derechos reservados.

C:\Users\EDi> ffmpeg -version
ffmpeg version git-2019-11-26-504264b Copyright (c) 2000-2019 the FFmpeg developers
built with gcc 9.2.1 (GCC) 20191125
configuration: --enable-gpl --enable-version3 --enable-sdl2 --enable-fontconfig --en
able-gnutls --enable-iconv --enable-libass --enable-libdav1d --enable-libbluray --en
able-libfreetype --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore
-amrwb --enable-libopenjpeg --enable-libopus --enable-libshine --enable-libsnelly --
enable-libsoxr --enable-libtheora --enable-libtwolame --enable-libvpx --enable-libwa
vpack --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxml2 --enable-l
ibzing --enable-lzma --enable-zlib --enable-gmp --enable-libvidstab --enable-libvorbi
s --enable-libvo-amrwbenc --enable-libmysofa --enable-libspeex --enable-libxvid --e
nable-libaom --enable-libbfx --enable-ffnvcodec --enable-cuvid --enable-d3d11va --en
able-nvenc --enable-nvdec --enable-dxva2 --enable-avisynth --enable-libopenmpt --ena
ble-amf
libavutil      56. 36.100 / 56. 36.100
libavcodec     58. 62.100 / 58. 62.100
libavformat    58. 35.100 / 58. 35.100
libavdevice    58.  9.101 / 58.  9.101
libavfilter     7. 67.100 /  7. 67.100
libswscale      5.  6.100 /  5.  6.100
libswresample  3.  6.100 /  3.  6.100
libpostproc   55.  6.100 / 55.  6.100

C:\Users\EDi>

```

Si al ingresar el comando de la versión del programa, muestra algún error como resultado, se debe realizar el proceso de instalación nuevamente, caso contrario, la instalación y habilitación del software se ha completado correctamente.

3.3.2 Selección de los Perfiles de Codificación (Encoding Ladder)

Antes de codificar y comprimir el contenido, es necesario establecer los perfiles de codificación en base al ancho de banda que posee la mayor cantidad de potenciales usuarios que utilizarán el servicio. En una investigación inglesa denominada “*Worldwide Broadband Speed League 2019*”, sitúa al Ecuador en el puesto 132 a nivel mundial, con una velocidad media de descarga de 3,46 Mbps (Cable Company, 2019). Netflix, gracias a su gran infraestructura de red desplegada a nivel mundial, realiza estudios y mediciones del rendimiento de su plataforma y de la velocidad de determinados ISP’s durante el horario de mayor audiencia de la plataforma en todo el mundo, Ecuador no está exento de estos estudios, presentando los siguientes resultados.

Tabla 4

Posiciones de ISP’s según el rendimiento de Netflix en Febrero 2020

Puesto	Proveedor de Servicios de Internet (ISP)	Velocidad (Mbps)	Tipo
1	CNT Fibra	4,00	Fibra Óptica
2	Netlife	3,90	Fibra Óptica
3	PuntoNet Celerity	3,71	Fibra Óptica
4	TVCable	3,54	Fibra Óptica/Cable
5	ETAPA Fibra	3,46	Fibra Óptica
6	Claro	3,30	Cable/Wireless
7	PuntoNet	2,52	DSL/Wireless/Satélite
8	CNT	2,38	DSL
9	ETAPA	1,95	DSL/WiMAX

Nota. Tomado de *Tabla de Posiciones de ISP en Ecuador*, Netflix, febrero 2020.

Además, se investigó el “*Encoding Ladder*” de dos plataformas muy populares utilizadas a diario, empezando por Netflix quien ofrece los siguientes niveles:

Tabla 5

Escalera de codificación de Netflix

Bitrate (Mbps)	Resolución	Escala
0,24	320x240p	4:3
0,38	426x240p	16:9
0,56	512x384p	4:3
0,74	640x360p	16:9
1,05	640x480p	4:3
1,75	854x480p	16:9
2,35	1280x720p	16:9
3,00	1280x720p	16:9
4,30	1920x1080p	16:9
5,80	1920x1080p	16:9

Nota. Tomado de *La compleja infraestructura detrás de Netflix*, Xataka Streaming, 2018.

Analizando la *Tabla 5 mostrada anteriormente*, se determina que Netflix tiene un enfoque en usuarios con velocidades muy bajas de conexión, contemplando un rango desde los 0,24 Mbps hasta los 5,80 Mbps, similares a las velocidades que se disponen en el Ecuador, teniendo en cuenta los datos presentados en la

Tabla 4, donde las velocidades promedio de conexión a Internet no sobrepasan los 5 Mbps.

La siguiente plataforma muy popular es YouTube, que ofrece a los usuarios un rango de resoluciones de mayor calidad, pero exige mayor velocidad de conexión para soportarlas, tal como se observa en la *Tabla 6* mostrada a continuación:

Tabla 6

Escalera de codificación de YouTube

Bitrate (Mbps)	Resolución	Escala
0,25	256x144p	16:9
0,65	426x240p	16:9
1,00	640x360p	16:9
2,50	854x480p	16:9
5,00	1280x720p	16:9
8,00	1920x1080p	16:9
16,00	2560x1440p	16:9
26,00	3840x2160p	16:9

Nota. Tomado de *Calidad, resolución y tamaño de videos en YouTube*, Norfipc YouTube, 2020.

Al relacionar la *Tabla 5* y *Tabla 6* de los perfiles de codificación pertenecientes a las plataformas Netflix y YouTube respectivamente, y teniendo en cuenta que las velocidades de conexión en la mayoría de los hogares en el Ecuador no sobrepasan los 5 Mbps, se propone los siguientes perfiles de codificación detallados en la *Tabla 7* con los que se implementará este proyecto de investigación.:

Tabla 7

Escalera de codificación propuesta para la implementación

Bitrate de Video (Mbps)	Resolución	Escala	Bitrate de Audio (kbps)
0,235	426x240p	16:9	64
0,560	640x360p	16:9	64
1,05	854x480p	16:9	96
2,35	1280x720p	16:9	128
4,30	1920x1080p	16:9	128

Al establecer estos perfiles de codificación, se podría asegurar que, la gran mayoría de los usuarios en el Ecuador disfrutarán de un servicio de calidad, teniendo la posibilidad de adaptarse a la mejor resolución posible, con *Bitrates* que no sobrepasan los 5 Mbps y visualizar los videos en pantallas 16:9, compatible con varios dispositivos.

Como consecuencia de establecer esta escalera de codificación, se busca optimizar el tiempo que toma iniciar una reproducción, mostrando imágenes con la calidad más baja posible en los primeros *frames*, y gracias al proceso adaptativo, la calidad mejorará en unos segundos de acuerdo con el ancho de banda que disponga el usuario. Esta cualidad de mostrar fragmentos del video a penas se lo reproduzca, se la conoce como “*Time to Play*”, y es un parámetro que interviene directamente con la experiencia del usuario, ya que la mayoría de personas prefiere visualizar el inicio del video en baja calidad (al instante en que se lo reproduce), que regularmente son escenas sin relevancia como títulos o escenas de la productora.

3.3.3 Codificación y Compresión de los Contenidos

Para adquirir los videos que se difundirán en esta implementación (por ser un proyecto de investigación sin fines comerciales), se puede utilizar cualquier herramienta de descarga gratuita de videos, como por ejemplo *aTube Catcher*. Con FFMPEG, se codificarán los videos descargados en varios archivos de distintas calidades (alineados a la escalera de codificación establecida), y estos serán encapsulados con el contenedor mp4.

Respecto a los códecs de video disponibles, se utilizará el códec H.264 para la codificación, porque las resoluciones de video a emplear son bajas y no sobrepasan al Full HD(1920x1080p), mientras que el códec H.265 fue enfocando para resoluciones Ultra HD o 4K(3840x2160p), que no se utilizarán en esta implementación, a pesar de ello, es posible utilizar este códec, pero necesariamente tendría un mayor costo computacional para el procesador. Para codificar el audio, se utilizará el códec AAC(*Advanced Audio Coding*), siendo el más popular y estandarizado hoy en día, teniendo en cuenta que la calidad del audio no es muy perceptible por el oído humano.

Para codificar un video se debe ingresar al cmd, colocando líneas de comandos que definen ciertos parámetros de codificación, por ejemplo, para codificar un video identificado con el nombre "video", el cual muestra la erupción del volcán Cotopaxi del año 2015, primero se debe colocar dicho video dentro de la carpeta "bin", que se encuentra en la carpeta "ffmpeg" del disco local donde se instaló el programa, después desde la consola del cmd, acceder al directorio donde se colocó el video, ingresando el comando "cd C:\ffmpeg\bin", y al ubicarse en ese directorio se debe ingresar el siguiente listado de comandos, línea por línea para cada resolución establecida:

- a) `ffmpeg -i video.mp4 -vcodec libx264 -x264opts keyint=58:min-keyint=58:scenecut=1 -acodec aac -b:v 235k -b:a 64k -s 426x240 video240.mp4`
- b) `ffmpeg -i video.mp4 -vcodec libx264 -x264opts keyint=58:min-keyint=58:scenecut=1 -acodec aac -b:v 560k -b:a 64k -s 640x360 video360.mp4`
- c) `ffmpeg -i video.mp4 -vcodec libx264 -x264opts keyint=58:min-keyint=58:scenecut=1 -acodec aac -b:v 1050k -b:a 96k -s 854x480 video480.mp4`
- d) `ffmpeg -i video.mp4 -vcodec libx264 -x264opts keyint=58:min-keyint=58:scenecut=1 -acodec aac -b:v 2350k -b:a 128k -s 1280x720 video720.mp4`
- e) `ffmpeg -i video.mp4 -vcodec libx264 -x264opts keyint=58:min-keyint=58:scenecut=1 -acodec aac -b:v 4300k -b:a 128k -s 1920x1080 video1080.mp4`

Figura 30

Captura de pantalla de FFMPEG al finalizar la codificación del video

```

C:\Windows\system32\cmd.exe
frame= 1351 fps=172 q=-1.0 Lsize= 1755kB time=00:00:45.09 bitrate= 318.7kbits/s speed=5.76x
video:1355kB audio:353kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 2.71
0095%
[libx264 @ 0000019945f62240] frame I:24 Avg QP:19.82 size: 14629
[libx264 @ 0000019945f62240] frame P:791 Avg QP:22.31 size: 1261
[libx264 @ 0000019945f62240] frame B:536 Avg QP:23.53 size: 72
[libx264 @ 0000019945f62240] consecutive B-frames: 38.7% 20.1% 15.1% 26.1%
[libx264 @ 0000019945f62240] mb I I16..4: 24.6% 40.1% 35.3%
[libx264 @ 0000019945f62240] mb P I16..4: 1.0% 0.3% 0.1% P16..4: 27.4% 9.8% 6.1% 0.0% 0
.0% skip:55.3%
[libx264 @ 0000019945f62240] mb B I16..4: 0.0% 0.0% 0.0% B16..8: 14.9% 0.1% 0.0% direct:
0.2% skip:84.7% L0:36.1% L1:61.8% BI: 2.1%
[libx264 @ 0000019945f62240] final ratefactor: 22.87
[libx264 @ 0000019945f62240] 8x8 transform intra:34.4% inter:59.2%
[libx264 @ 0000019945f62240] coded y,uvDC,uvAC intra: 51.6% 65.4% 40.8% inter: 9.5% 7.3% 0.6%
[libx264 @ 0000019945f62240] i16 v,h,dc,p: 1% 55% 1% 43%
[libx264 @ 0000019945f62240] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 3% 49% 16% 3% 2% 1% 6% 2% 16%
[libx264 @ 0000019945f62240] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 5% 42% 11% 5% 5% 3% 10% 3% 17%
[libx264 @ 0000019945f62240] i8c dc,h,v,p: 38% 51% 2% 8%
[libx264 @ 0000019945f62240] Weighted P-Frames: Y:14.9% UV:3.4%
[libx264 @ 0000019945f62240] ref P L0: 72.4% 13.8% 9.0% 3.9% 0.9%
[libx264 @ 0000019945f62240] ref B L0: 90.3% 7.1% 2.6%
[libx264 @ 0000019945f62240] ref B L1: 97.2% 2.8%
[libx264 @ 0000019945f62240] kb/s:246.21
[aac @ 0000019945f63f40] Qavg: 220.208
C:\ffmpeg\bin>

```

Al ingresar la primera línea de comandos (tal como está escrito en el ítem a)), se procesará el video y después de unos segundos se mostrará una pantalla similar a la expuesta por la *Figura 30*. Sin aparecer algún error en el proceso, se debe ingresar la siguiente línea de comandos (ítem b)) en el cmd, para codificar el video con su nivel subsiguiente, y así sucesivamente los ítems faltantes, en el caso de presentar un error en este procedimiento, es probable que cierto comando esté mal escrito o hubo un problema en la instalación del programa y es necesario repetir la instalación. Para comprender de mejor manera las líneas de comandos ingresadas, en la *Tabla 8* se exponen los comandos con la descripción de cada uno de ellos.

Tabla 8

Descripción de los comandos ingresados para la codificación del video

Comandos expuestos en la primera línea ingresada "a)"	
ffmpeg -i video.mp4	Indica el nombre del archivo que se va a codificar, en este caso "video".
-vcodec libx264	Menciona la librería de codificación que se va a utilizar, en este caso el códec utilizado es H.264.
-x264opts keyint=58:min- keyint=58:scenecut=-1	Ingresa a opciones avanzadas del códec, donde impone al codificador una longitud mínima del GOP=58, siendo ésta la distancia entre <i>frames</i> "I", y también desactiva la detección de cambio de escena automática con el valor "-1", dejándole fijo al GOP en 58. Si no se desactiva este valor, se generarían <i>frames</i> "I" automáticamente cada vez que se detecte un cambio.
-acodec aac	Menciona el códec de audio que se va a utilizar, en este caso AAC.
-b:v 235k	Indica el Bitrate con el que se va a codificar el video en [bps], en este caso 235 [kbps].
-b:a 64k	Indica el Bitrate con el que se va a codificar el audio en [bps], en este caso 64 [kbps].
-s 426x240	Indica la resolución con la que se debe codificar el video, en este caso la primera línea de comandos es para 426x240 pixeles.
video240.mp4	Establece el nombre del archivo resultante del proceso, en este caso será el mismo nombre "video" junto con su resolución para identificarlo.

Las otras cuatro líneas de código representan las mismas características de la primera línea mencionada en la tabla, al codificar el mismo video de entrada, solo varía los cuatro comandos del final, con sus *Bitrates* de audio y video respectivos, resolución establecida por la escalera de codificación, y el nombre del archivo saliente, caso contrario se sobrescribirá dicho archivo.

Se debe tener en cuenta el valor ingresado en el GOP, se decidió establecer un GOP fijo para este caso, pero no necesariamente debe ser así, ya que la misma tecnología del códec se encarga de detectar cambios en la escena y genera automáticamente los *frames* "I", si desea hacerlo así, se debe omitir el tercer comando de la tabla ocasionando un GOP variable, pero si se va a establecerlo como un valor fijo, debe fijarse en las características del contenido de origen, por ejemplo, el video de entrada que fue codificado con la primera línea de comandos, tiene una tasa de cuadro o *frame rate* de 29 fotogramas por segundo, y se pretende generar segmentos de duración 2 segundos, por esta razón se impone generar fotogramas tipo "I" cada 58 cuadros en el comando ingresado.

En la plataforma a implementar, se expondrán varios contenidos codificados con GOP fijo y GOP variable, porque ambos son compatibles con la tecnología de adaptación sin presentar inconvenientes al reproducir el video, por ejemplo, con el siguiente grupo de comandos (en el que se omite el tercer comando de la *Tabla 8*) se codifica un video musical con GOP variable, el archivo tiene el nombre de "guayaquil" (que hace referencia al nombre de la canción), los *Bitrates* de audio, video y las resoluciones son los mismos valores que se utilizaron en la codificación del primer video y se establece el nombre del archivo de salida para cada resolución respectivamente.

- a) `ffmpeg -i guayaquil.mp4 -vcodec libx264 -acodec aac -b:v 235k -b:a 64k -s 426x240 guayaquil 240.mp4`
- b) `ffmpeg -i guayaquil.mp4 -vcodec libx264 -acodec aac -b:v 560k -b:a 64k -s 640x360 guayaquil 360.mp4`
- c) `ffmpeg -i guayaquil.mp4 -vcodec libx264 -acodec aac -b:v 1050k -b:a 96k -s 854x480 guayaquil 480.mp4`
- d) `ffmpeg -i guayaquil.mp4 -vcodec libx264 -acodec aac -b:v 2350k -b:a 128k -s 1280x720 guayaquil 720.mp4`
- e) `ffmpeg -i guayaquil.mp4 -vcodec libx264 -acodec aac -b:v 4300k -b:a 128k -s 1920x1080 guayaquil 1080.mp4`

Sin embargo, después de realizar varias pruebas con videos codificados con un valor de GOP variable, las transmisiones presentaban inconvenientes cuando el ancho de banda disponible no era el suficiente, por lo que se llegaba a desincronizar las secuencias de video y se perdía en su totalidad la transmisión del contenido, por este motivo práctico, se recomienda establecer un valor de GOP fijo, con el fin de generar fotogramas I a un mismo intervalo, lo que evitará los inconvenientes en la transmisión.

3.4 Paquetización y Distribución del Contenido

3.4.1 Instalación del Software Wowza

Para instalar el motor de streaming de Wowza, siendo una herramienta de tipo Java, su ejecutable proporciona una instalación guiada en cualquier sistema operativo, convirtiéndose en un proceso muy sencillo de seguir, con unos simples pasos, aceptación de términos y validación de su licencia, podemos empezar a utilizar esta poderosa herramienta de streaming.

Figura 31

Captura de la página oficial de descarga de Wowza



1. Obtenga una licencia de Wowza Streaming Engine

Para usar el software Wowza Streaming Engine™, necesita una licencia válida. Usted puede:

- Regístrese para obtener una [prueba estándar gratuita](#) o una [prueba de desarrollador](#) para recibir una clave de licencia de prueba por correo electrónico.
- Compre una licencia [mensual](#), [anual](#) o [perpetua](#).

2. Descargue el instalador de Wowza Streaming Engine *

La versión actual es Wowza Streaming Engine 4.8.0, compilación 20200213142111, lanzada el 18 de febrero de 2020.

* Para actualizar una instalación existente de Wowza Streaming Engine con las últimas funciones y mejoras, inicie sesión en su cuenta de Wowza y vaya a [Mis descargas](#). Después de descargar la actualización, consulte [Cómo actualizar su instalación de Wowza Streaming Engine](#).



Se debe ingresar en la página oficial <https://www.wowza.com/pricing/installer>, tal como se muestra en la *Figura 31*, y solicitar una licencia para activar la herramienta, en el sitio puede adquirir licencias para uso comercial (con un costo mensual) y en este caso, adquirir licencias de desarrollador con fines no comerciales totalmente gratuita.

Figura 32

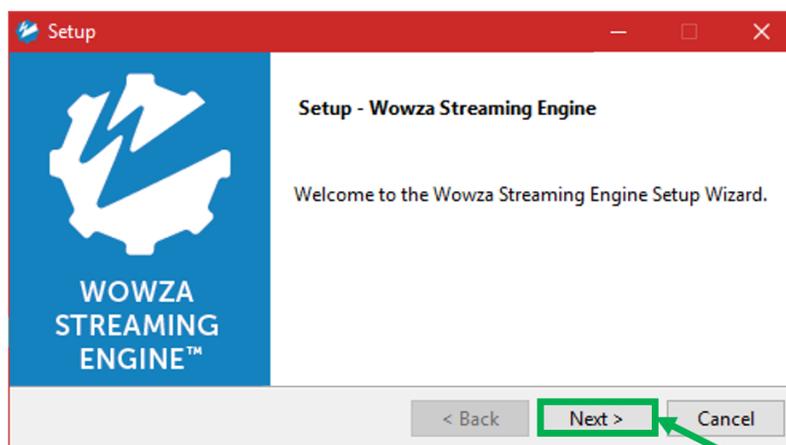
Captura de la página para solicitar la licencia de desarrollador de Wowza



Se solicita la licencia con el formulario de la *Figura 32*, resultado de esto, en el correo electrónico registrado recibirá la clave de la licencia solicitada, la cual tiene una duración de 180 días para utilizar la herramienta, sin embargo, transcurrido ese tiempo es posible solicitar una nueva clave, esto extenderá la duración de la licencia las veces que sea, en este caso, tuve que realizar el proceso mencionado, confirmando la posibilidad de utilizar el software el tiempo que se requiera, siempre y cuando el uso sea estricto para proyectos de investigación o para ámbitos educativos.

Figura 33

Ventana de instalación guiada de Wowza

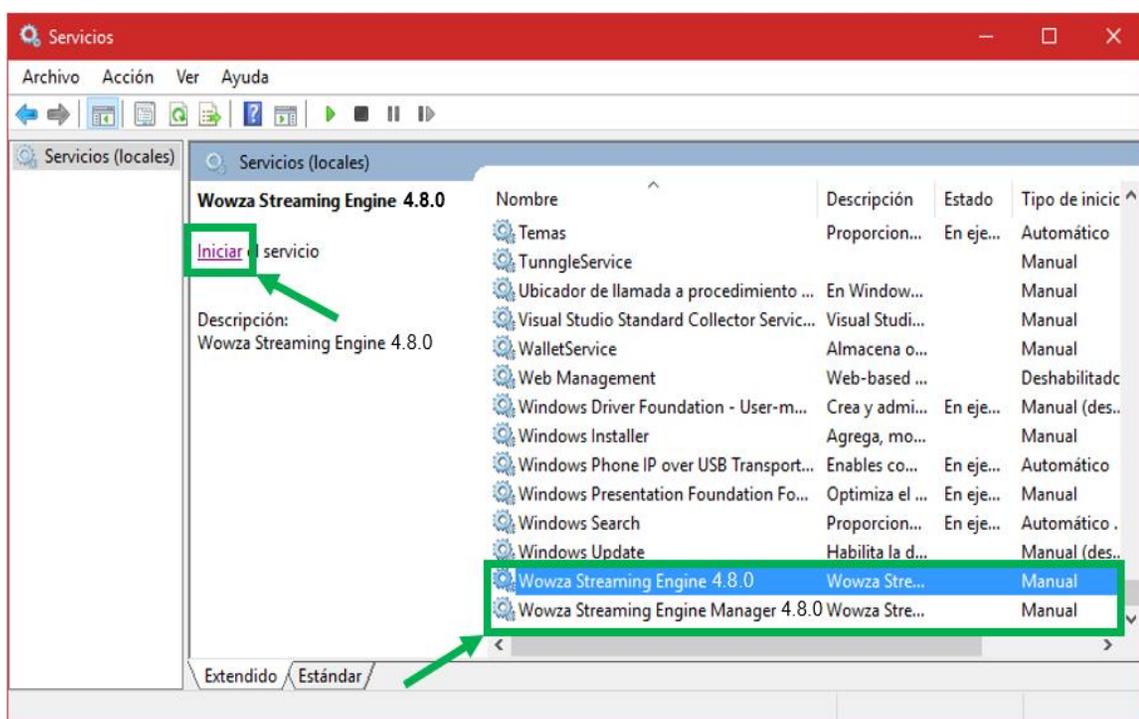


Con la clave de la licencia de desarrollador en su poder, debe descargar el instalador para el sistema operativo que se utilice, en este caso se utiliza Windows 10 Pro, y finalmente debe seguir los pasos de instalación que se muestren e ingresar la clave de la licencia que debió recibir en el correo registrado. Con el proceso de instalación terminado, se configura la opción para que inicie automáticamente (al encender el computador), o en este caso, se prefirió la opción de inicio manual del servidor (recomendable para evitar el consumo innecesario de recursos).

Para arrancar el servidor instalado (efecto de configurar el inicio manual), en la barra de “Búsqueda en Windows” debe escribir “Servicios” para abrir dicha ventana y dirigirse al servicio de “Wowza Streaming Engine”, aquí debe seleccionar la opción “Iniciar”, y lo mismo realizar con el servicio de “WSE Manager” el cual gestionará al servidor, tal como se observa en la *Figura 34*.

Figura 34

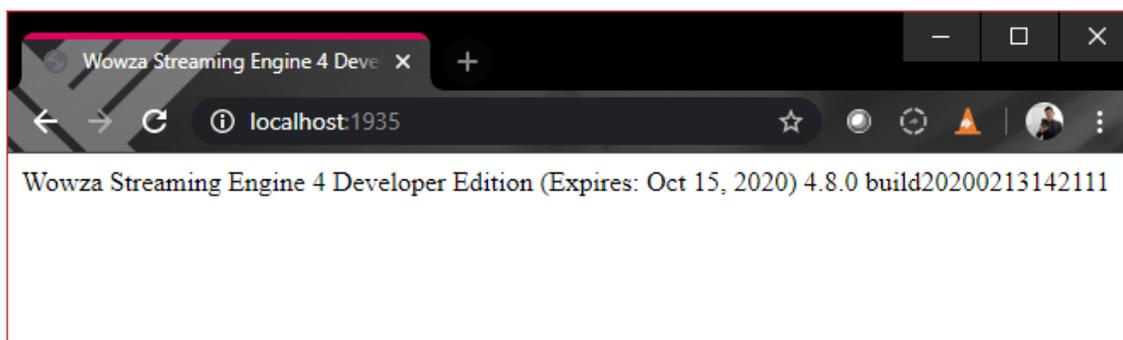
Ventana de “Servicios” de Windows para iniciar Wowza



Para verificar el correcto funcionamiento del servidor de streaming en el ordenador, se debe dirigir a la siguiente dirección <http://localhost:1935/> en el navegador, dicha dirección debería mostrar la versión del servidor que fue instalada, el número de compilación del software y la fecha de expiración de la licencia registrada, tal como se puede evidenciar en la *Figura 35*.

Figura 35

Verificación del funcionamiento de Wowza en el ordenador



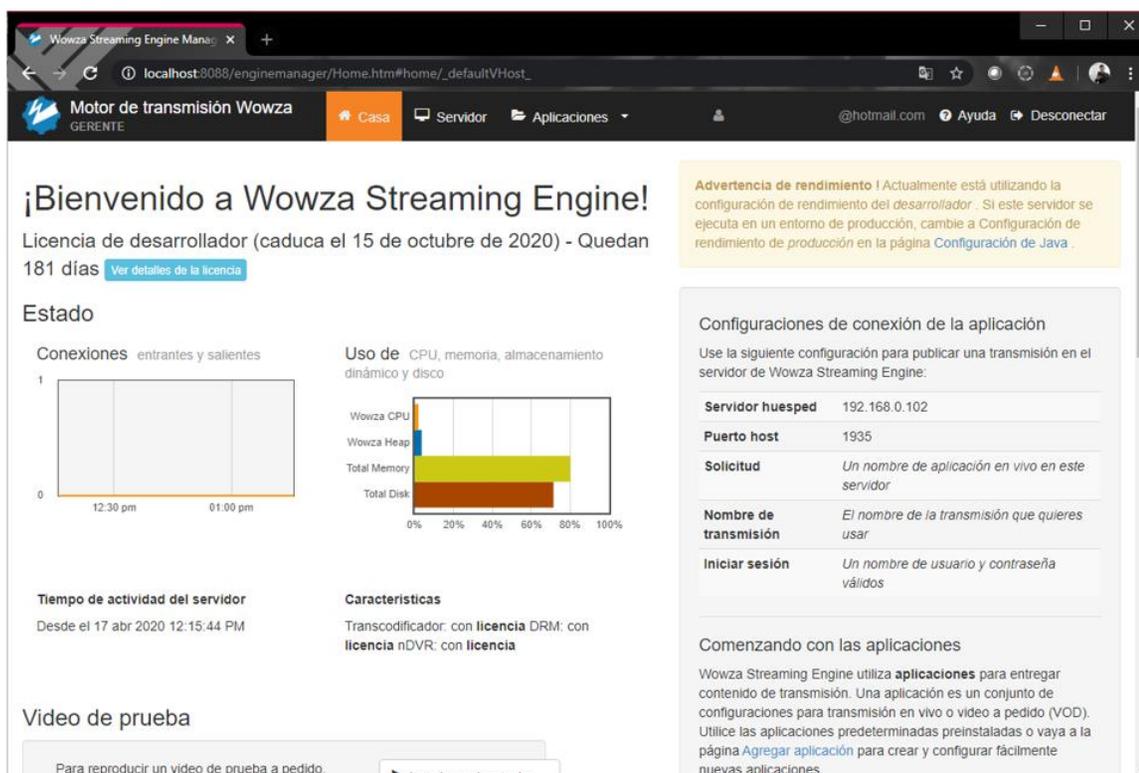
3.4.2 Configuración del Motor de Streaming

El servidor Wowza posee una interfaz basada en web, para ingresar a dicha interfaz posterior a la instalación e inicio del servicio, mediante un navegador web debe ingresar al siguiente enlace <http://localhost:8088/enginemanager> para configurar una contraseña de ingreso al servidor, realizado esto, se direccionará al siguiente link <http://localhost:8088/enginemanager/login.htm> para validar sus credenciales y así poder ingresar a configurar el servidor de streaming de Wowza.

Al acceder en la interfaz web del servidor, se observa la pantalla principal similar a la captura mostrada en la *Figura 36*, y es aquí donde se gestiona, configura y administra el servicio de streaming. El servidor pone a disposición gráficos en tiempo real, los cuales muestran el rendimiento del CPU, la memoria utilizada, las conexiones entrantes y salientes al servidor, y otras métricas útiles para monitorear el estado de la plataforma; también el servidor dispone de un video de prueba y reproductores de las distintas tecnologías de streaming, para comprobar que el servidor de streaming de Wowza está en la capacidad de realizar transmisiones.

Figura 36

Interfaz principal del motor de streaming de Wowza

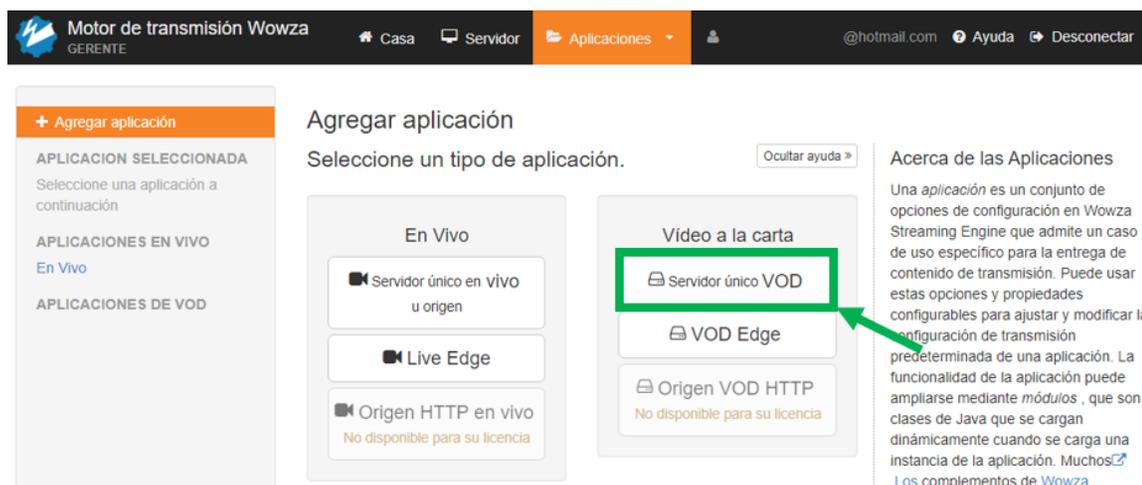


Las configuraciones realizadas en esta interfaz web serán recopiladas en distintos archivos XML alojados en el disco local (donde fue instalado el software) y para lograr la transmisión de los videos codificados, fue necesario apoyarse en el manual de configuración oficial de Wowza, aquí referenciado (Wowza Media Systems, 2020).

Primero, hay que dirigirse a la pestaña “Aplicaciones”, aquí es donde se elige el tipo de servicio que se va a ofrecer (*Live TV* o *Video on Demand*), en este caso, se pretende ofrecer a los usuarios un servicio de streaming de video bajo demanda VoD, entonces se debe seleccionar “Servidor Único VOD”, tal como se puede observar en la *Figura 37* mostrada a continuación.

Figura 37

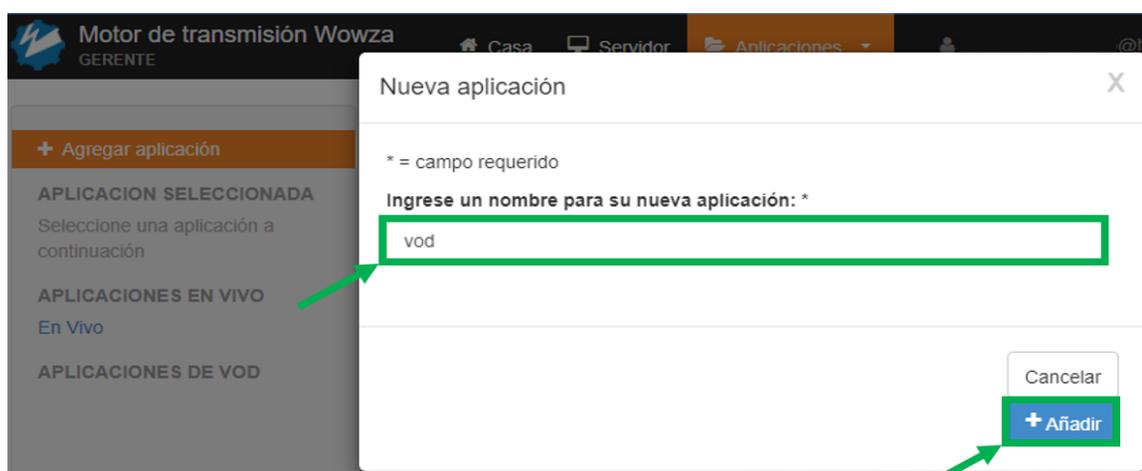
Selección del tipo de servicio de streaming que se va a ofrecer



Después, hay que ingresar un nombre a la aplicación, en este caso se la nombró “vod”, y finalmente se pulsa “Añadir”, tal como se observa en la *Figura 38*.

Figura 38

Asignación de un nombre a la aplicación de VoD creada



En la aplicación creada se puede habilitar varias tecnologías de streaming de video, tal como en la *Figura 39*, donde es posible crear más aplicaciones del tipo VoD y configurarlas de distintas formas, en este caso, se usa una misma aplicación con la capacidad de soportar todas las tecnologías de streaming disponibles,.

Figura 39

Configuración de la tecnología de streaming soportada por la plataforma

The screenshot shows the 'Motor de transmisión Wowza' interface. The top navigation bar includes 'Casa', 'Servidor', and 'Aplicaciones'. The left sidebar shows a list of applications, with 'vod' selected under 'APLICACION SELECCIONADA'. The main content area is titled 'vod' and describes it as a 'Servidor único de Video on Demand u Origin'. It has tabs for 'Preparar', 'Propiedades', and 'Módulos'. Below the tabs, there is a note: '* = Campo obligatorio' and a 'Guardar' button. The 'Descripción de la aplicación' section contains a text box with the following text: 'Aplicación predeterminada para transmisión de video a pedido (VOD) creada cuando se instala Wowza Streaming Engine. Use esta aplicación con su configuración predeterminada o modifique la configuración según sea necesario. También puede copiarlo para crear otra aplicación VOD.' The 'Tipos de reproducción *' section lists several options with checked checkboxes: MPEG-DASH, Apple HLS, Adobe RTMP, Adobe HDS, and Microsoft Smooth Streaming.

En la *Figura 40* se habilita “Intercambio de recursos de origen cruzado CORS”, para poder compartir contenidos con distintos dominios para secuencias basadas en HTTP, se debe dejar el directorio en “Uso por defecto”, y se selecciona “3GPP/MPEG4”, recomendada para aplicaciones de VoD, sin embargo existen opciones en el caso de utilizar subtítulos en archivos externos y puedan ser sincronizados con la transmisión, pero en este caso, no serán implementados videos subtitulados.

Figura 40

Configuración del directorio para alojar los videos del servidor

The screenshot shows the configuration page for the 'vod' application. On the left, a sidebar lists application categories: 'APLICACION SELECCIONADA' (with 'vod' selected), 'APLICACIONES EN VIVO', and 'APLICACIONES DE VOD'. The main content area is divided into sections: 'Opciones' (with 'Intercambio de recursos de origen cruzado (CORS)' checked), 'Directorio de contenido' (with 'Uso por defecto' selected and a path field containing '\$ {com.wowza.wms.context.VHostConfigHome} / content / vod'), and 'Fuentes de subtítulos' (with 'Pistas de texto sincronizado 3GPP / MPEG-4 integradas' checked). At the bottom, there are 'Guardar' and 'Cancelar' buttons.

Después de guardar los cambios, en la misma aplicación “vod” se selecciona “Seguridad de Reproducción” tal como se observa en la *Figura 41*, y así deshabilitar el “SecureToken” y las “Restricciones del Cliente”, lo que permitiría a los usuarios acceder al contenido sin autenticación o restricción alguna.

Figura 41

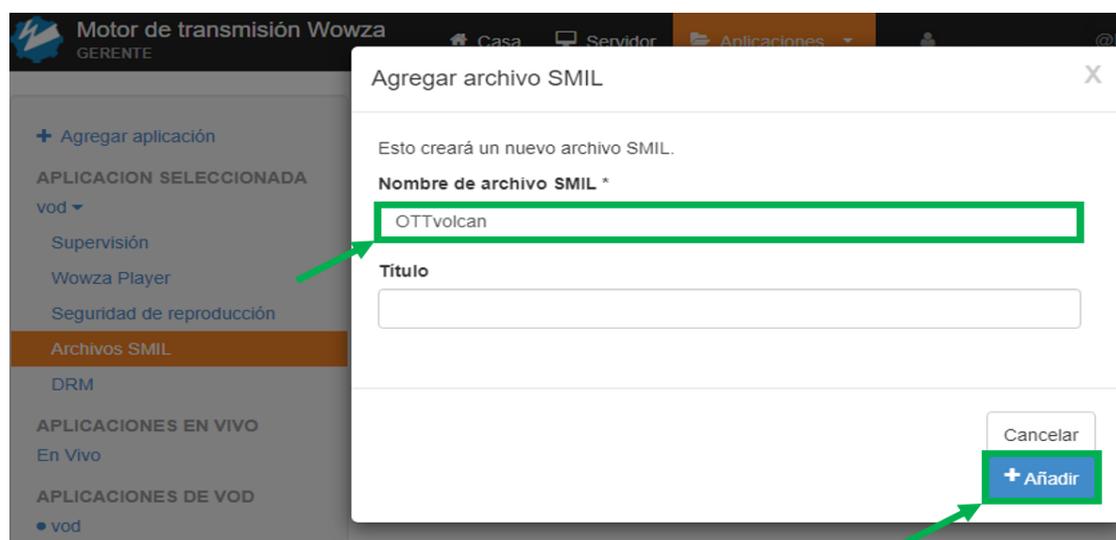
Autenticación y restricciones del cliente para acceder a los contenidos

The screenshot shows the configuration page for the 'vod' application with 'Seguridad de reproducción' selected in the sidebar. The main content area has a heading 'Configure las opciones para ayudar a asegurar las conexiones de los jugadores a esta aplicación.' followed by an 'Editar' button. Below this, two sections are shown: 'SecureToken' (No disponible) and 'Restricciones del cliente' (No disponible).

A continuación, se selecciona la opción “Archivos SMIL” para crear en el servidor un archivo del tipo SMIL (*Synchronized Multimedia Integration Language*), el cual estará alineado a la escalera de codificación establecida. Cada archivo .smil añadido hace referencia al conjunto de archivos en distintas calidades de un solo video, por ejemplo, para el primer video codificado (erupción del volcán Cotopaxi), se crea el archivo SMIL con un nombre que referencie al video, en este caso “OTTvolcan”.

Figura 42

Creación del archivo SMIL para el primer video codificado



En el archivo “OTTvolcan.smil” creado, se selecciona “Agregar corriente SMIL” y aquí es donde se ingresa toda la información correspondiente al primer archivo resultante de la codificación, en este caso el archivo es “video240.mp4”, siendo el primer nivel de la escalera de codificación del video, tal como se puede observar la *Figura 43*. Este proceso se repite para cada archivo resultante de la codificación, finalizando con el archivo “video1080.mp4”, logrando un listado de cinco archivos en total dentro del archivo SMIL, similar a lo mostrado en la *Figura 44*.

Figura 43

Configuración del primer nivel de la escalera de codificación

vod> OTTvolcan.smil> video240.mp4
 Servidor único de Video on Demand u Origin
[← Regresar al archivo SMIL](#)

Básico **Propiedades**

* = Campo obligatorio Ocultar ayuda >

Guardar **Cancelar**

Tipo de etiqueta
 vídeo

Fuente (src) * **Ancho de video**
 video240.mp4 426

Lenguaje del sistema **Altura de video**
 eng 240

Bitrate de video * **Bitrate de audio**
 235000 64000
 bps bps

Figura 44

Grupo de niveles correspondientes al primer video codificado

vod> OTTvolcan.sm Jugadores de prueba ...

Servidor único de Video on Demand u Origin
[← Regresar a archivos SMIL](#)

Editar Ocultar ayuda >

Título
 -No conjunto-

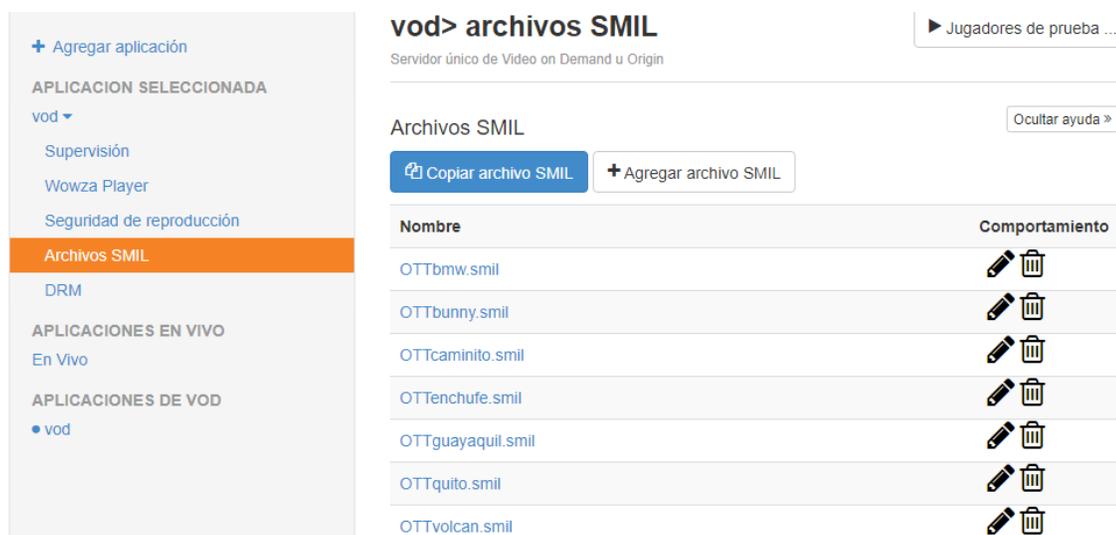
Secuencias en este archivo SMIL
 + Agregar corriente SMIL

Fuente (src)	Idioma	Tipo	Comportamiento
video240.mp4	eng	video	 
video360.mp4	eng	video	 
video480.mp4	eng	video	 
video720.mp4	eng	video	 
video1080.mp4	eng	video	 

Como ya fue mencionado, cada archivo .smil añadido representa al grupo de archivos (de distintas calidades) del video a difundir en la plataforma, esto quiere decir que, si se desea agregar nuevos contenidos a la plataforma, primero se debe obtener el video en su máxima calidad posible, codificarlo (de acuerdo a la escalera establecida) para disponer de sus distintas resoluciones, y finalmente, crear el archivo SMIL en el servidor de streaming con los niveles de codificación correspondientes, algo similar a la *Figura 45*, donde se puede observar un grupo de archivos SMIL que hacen referencia a videos que ahora se encuentran disponibles en la plataforma.

Figura 45

Archivos SMIL que referencian a los videos disponibles en la plataforma



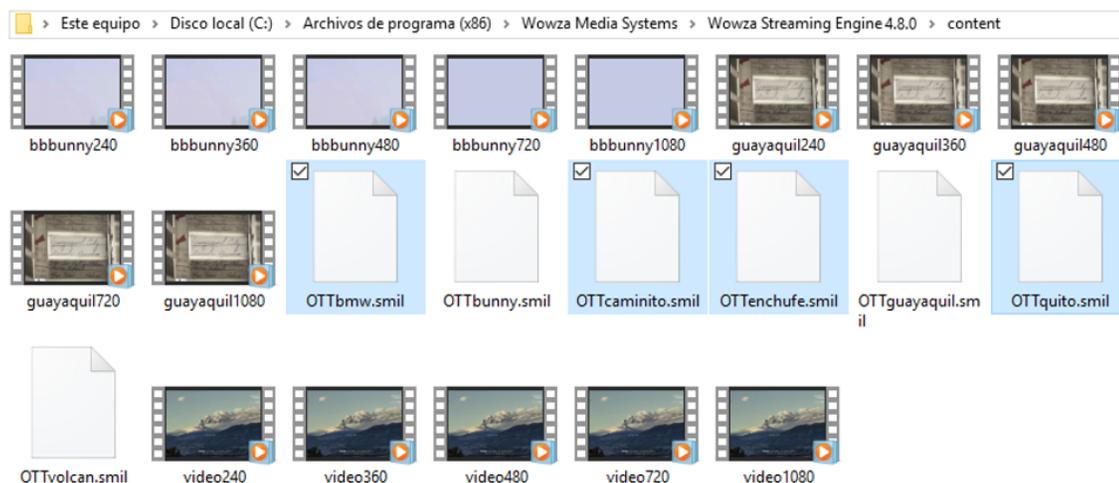
The screenshot shows a web interface for managing SMIL files. On the left is a sidebar with navigation options: '+ Agregar aplicación', 'APLICACION SELECCIONADA' (with 'vod' selected), 'Supervisión', 'Wowza Player', 'Seguridad de reproducción', 'Archivos SMIL' (highlighted), 'DRM', 'APLICACIONES EN VIVO' (with 'En Vivo'), and 'APLICACIONES DE VOD' (with 'vod' selected). The main content area is titled 'vod> archivos SMIL' and includes a sub-header 'Servidor único de Video on Demand u Origin'. Below this, there are buttons for 'Copiar archivo SMIL' and '+ Agregar archivo SMIL', and a link for 'Ocultar ayuda >>'. A table lists the SMIL files with their names and actions (edit and delete icons):

Nombre	Comportamiento
OTTbmw.smil	
OTTbunny.smil	
OTTcaminito.smil	
OTTenchufe.smil	
OTTguayaquil.smil	
OTTquito.smil	
OTTvolcan.smil	

El grupo de archivos de distintas resoluciones correspondientes a cada video, se colocan en "C:\Program Files (x86)\Wowza Media Systems\Wowza Streaming Engine4.8.0\content", ya que ese fue el directorio predeterminado configurado al momento de crear la aplicación "vod", por ejemplo, en la *Figura 46* se puede observar el contenido de la carpeta "content", después de crear los archivos SMIL en la interfaz web del servidor.

Figura 46

Contenido de la carpeta “content” dentro del disco local



Se puede observar que todos los archivos SMIL creados en la interfaz web del servidor se encuentran aquí alojados, sin embargo, solo se encuentran tres grupos de archivos (de las cinco resoluciones) correspondientes a los tres archivos .smil sin seleccionar en la figura y faltan los cuatro grupos de archivos (de las cinco resoluciones) que corresponden a los cuatro archivos .smil seleccionados en la figura, como consecuencia, al momento de crear los archivos SMIL en el servidor, estos son publicados en la red automáticamente para poder acceder a ellos, y si el usuario pretende reproducir un video y precisamente el grupo de archivos (de las cinco resoluciones) correspondientes a ese video no se encuentran alojados en el servidor (en la carpeta “content”), se provocaría un error de reproducción al no ubicar el contenido para ser transmitido, para evitar este problema, siempre debe verificarse que los videos estén situados dentro de la carpeta predeterminada del servidor y el archivo SMIL sea añadido correctamente.

3.5 Reproducción del Contenido

3.5.1 Acceso a los contenidos mediante VLC Media Player

Ahora es el momento de acceder al contenido y comprobar que la plataforma funciona correctamente, siendo fundamental el uso de un reproductor de video que sea compatible con distintos protocolos de streaming actuales, uno muy utilizado es VLC Media Player, siendo un software multiplataforma gratuito y de código abierto, tiene la capacidad de reproducir una gran variedad de archivos multimedia y es compatible con diversos protocolos de transmisión. Se puede descargar el software visitando la página oficial <https://www.videolan.org/vlc/index.es.html> mostrada en la *Figura 47*, para cualquier sistema operativo y también implementado para smartphones iOS y Android.

Figura 47

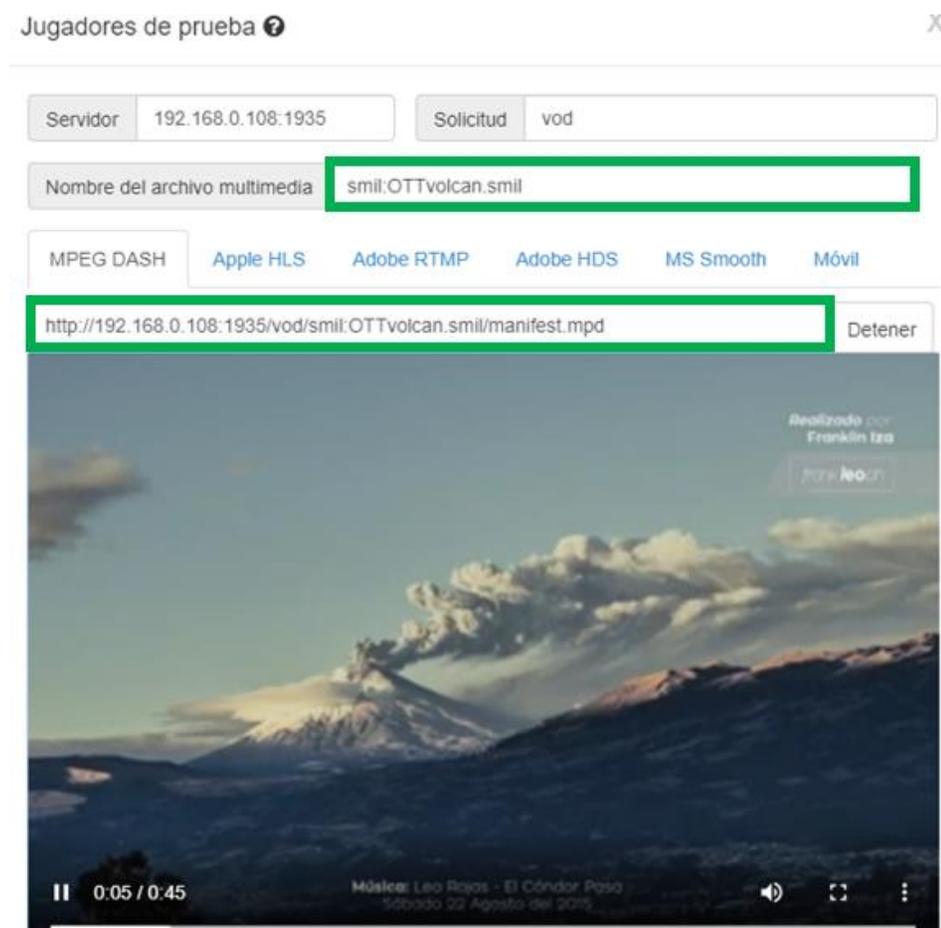
Captura de la página oficial de descarga de VLC Media Player



Para acceder a los contenidos, es necesario conocer la URL de cada video e ingresarla en el reproductor de VLC, dicha dirección se obtiene con la ayuda de los reproductores de prueba incorporados en el servidor de Wowza, como en la *Figura 48*.

Figura 48

URL del video mediante el reproductor de prueba de Wowza



Por ejemplo, para visualizar el video del volcán Cotopaxi, la dirección URL que lo referencia es <http://192.168.0.108:1935/vod/smil:OTTvolcan.smil/manifest.mpd>, ahora en el reproductor de VLC mostrado en la *Figura 49*, en la pestaña "Medio" se selecciona la opción "Abrir ubicación de red" y se ingresa la dirección URL del video, teniendo en cuenta que, el dispositivo donde se quiere acceder al servidor se encuentra dentro de la misma red, como resultado mostrado en la *Figura 50*, el video se reproduce inmediatamente, iniciando con una baja calidad de la imagen, y en unos segundos se visualiza la mejor calidad posible, adaptándose al estado de red que se disponga.

Figura 49

Acceso al contenido de la plataforma mediante VLC Media Player

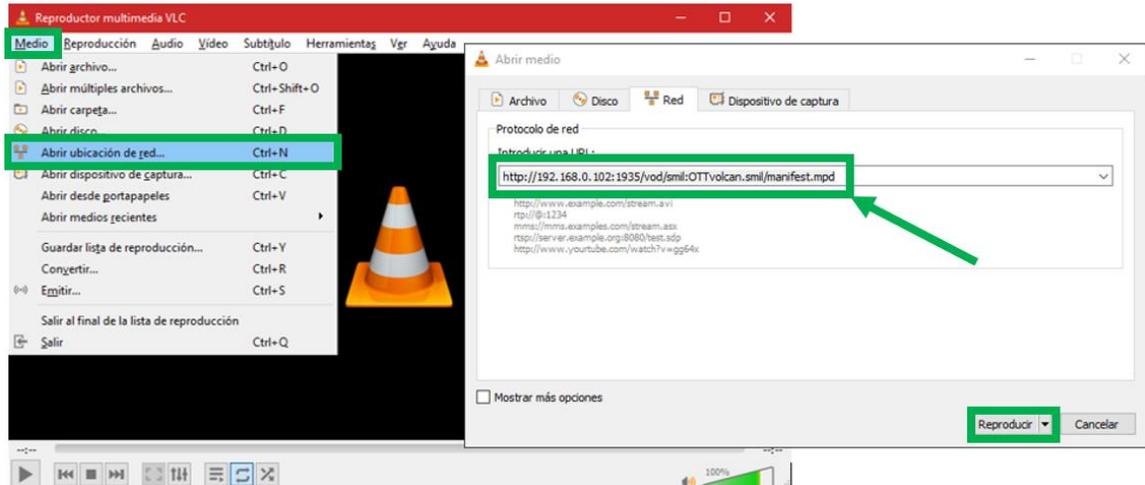
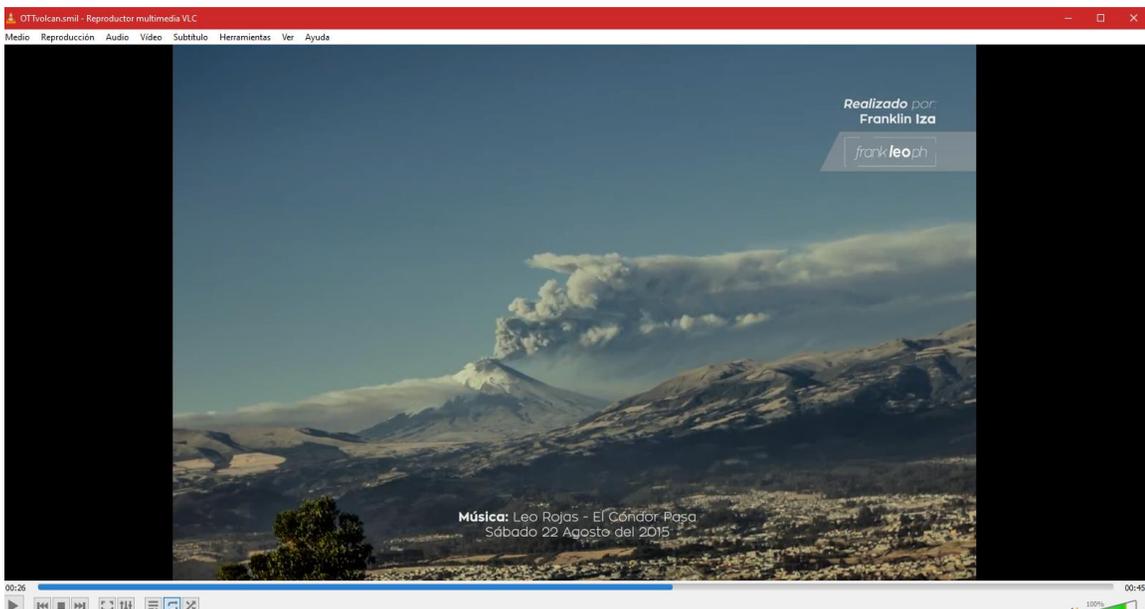


Figura 50

Reproducción del video mediante VLC Media Player



Con esto se demuestra que la plataforma funciona perfectamente con streaming adaptativo, iniciando la reproducción en la calidad más baja disponible y después

mejora su calidad de acuerdo a las condiciones del canal, esto caracteriza a un servicio OTT como tal, sin embargo, para acceder a los demás videos difundidos por la plataforma, se debería mostrar un listado a los usuarios de los videos disponibles junto con su dirección URL, para que puedan ingresar la URL del video que se desee reproducir, pero esta manera es algo tediosa y no muy amigable para los usuarios.

Por esta razón, se ha decidido implementar una página web que solucione este inconveniente, brindando una experiencia amigable a los usuarios, con un diseño que facilite el acceso a los contenidos de la plataforma y que se adapte al tamaño de pantalla de cualquier dispositivo.

3.5.2 *Diseño Web de la Plataforma OTT*

3.5.2.1 *Configuración del Servidor WEB*

Para implementar la página web de la plataforma, se utilizó *NetBeans IDE*, siendo un software libre y compatible con Java Script y HTML5, se lo considera una herramienta útil para proyectos que involucra el desarrollo web, este software se lo puede descargar en su sitio oficial <https://platform.netbeans.org/platform-get.html>.

Figura 51

Página oficial de descarga del software NetBeans para el desarrollo web

The screenshot shows the NetBeans website interface. At the top, there is a navigation menu with links for 'NetBeans IDE', 'Plataforma NetBeans', 'Complementos', 'Documentos y soporte', 'Comunidad', and 'Socios'. A search bar is located on the right side of the menu. Below the menu, the breadcrumb trail reads 'netbeans.org > proyectos > plataforma > Sitio web'. The main content area features a heading 'El | Inicio | Descripción | Escaparate' and a paragraph of introductory text. A prominent button labeled 'Download NetBeans IDE' is highlighted with a red border. To the right, there is a sidebar with a 'Navegación' section containing links for 'Casa', 'Mi página', 'Proyectos', and 'Personas', and a 'Características del proyecto' section.

Ahora, se requiere un servidor HTTP *open source*, que interprete el código programado y responda a las peticiones de los usuarios que visiten la página web desarrollada, para esto se utilizó el paquete de herramientas *open source* XAMPP, que dispone de un servidor web llamado Apache, compatible con varios sistemas operativos y se encuentra disponible en el sitio <https://www.apachefriends.org/es/download.html>.

Figura 52

Página oficial de descarga del paquete de software libre XAMPP

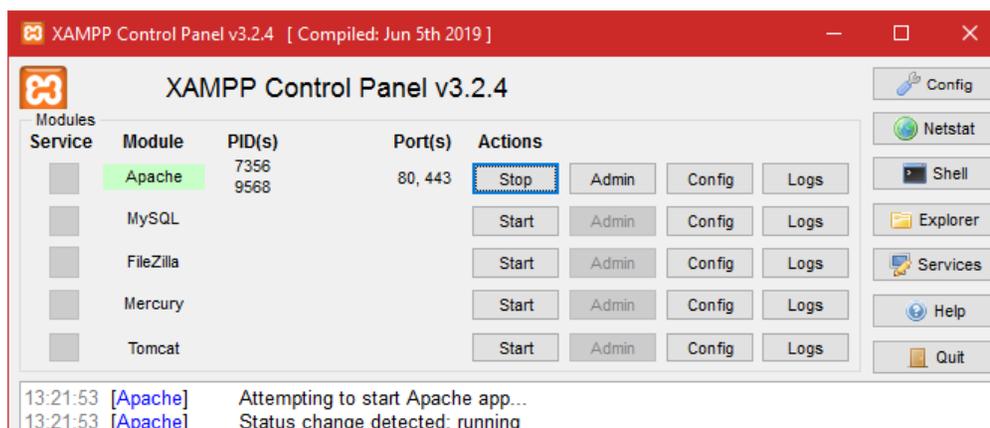
The screenshot shows the Apache Friends website's download page for XAMPP. The navigation bar includes links for 'Descargar', 'Complementos', 'Alojamiento', 'Comunidad', and 'Acerca de', along with a search bar and a language selector set to 'ES'. Below the navigation, a brief description states: 'XAMPP es una distribución de Apache fácil de instalar que contiene MariaDB, PHP y Perl. Simplemente descarga y ejecuta el instalador. ¡Es así de fácil!'. The main content area features three download options, each with an icon and version numbers: Linux (with a penguin icon), OS X (with an Apple icon), and Windows (with a Windows logo icon). To the right, a 'Documentación/FAQs' section explains that there is no manual and provides links to frequent questions for Linux, Windows, OS X, and OS X XAMPP-VM.

El servidor web Apache se lo utiliza para atender a los usuarios, los cuales podrán acceder a la página web de la plataforma sin inconvenientes de conexión, este servidor web se encuentra instalado en el mismo ordenador donde se implementó el servidor de streaming de Wowza, es decir, dos servidores están corriendo al mismo tiempo, entonces las características de software y hardware que posee el ordenador Lenovo IdeaPad U430 Touch intervienen en su desempeño.

Para iniciar el servidor web, se debe acceder al panel de control que posee el programa XAMPP similar al mostrado en la *Figura 53* a continuación, y aquí es posible activar o desactivar las herramientas que contiene el paquete, en este caso solo se activa el módulo de Apache.

Figura 53

Captura del panel de control de XAMPP donde se activa Apache

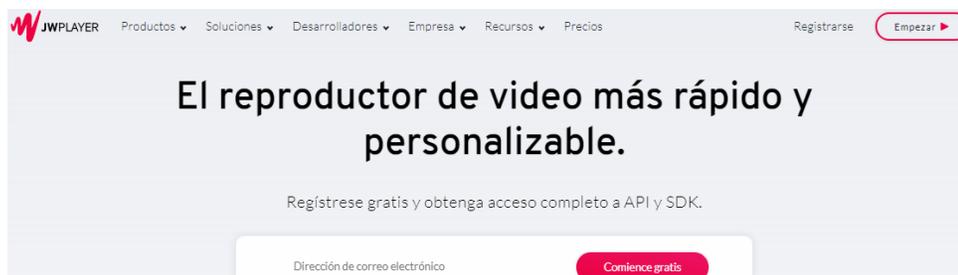


3.5.2.2 Implementación de JW Player

En la página web a diseñar, es necesario incrustar un reproductor compatible con los protocolos de transmisión utilizados y adicional, que posea un control manual sobre las resoluciones del video, lo que evidenciaría de mejor forma el dominio que posee el servidor de streaming sobre los niveles de codificación, siendo el reproductor JW Player el indicado y también ofrece un licenciamiento para desarrolladores, primero se debe descargar el archivo java script del reproductor y después registrarse en el sitio oficial de JW Player <https://www.jwplayer.com/sign-up/>, para obtener la llave de licencia.

Figura 54

Captura de la página oficial de registro de JW Player



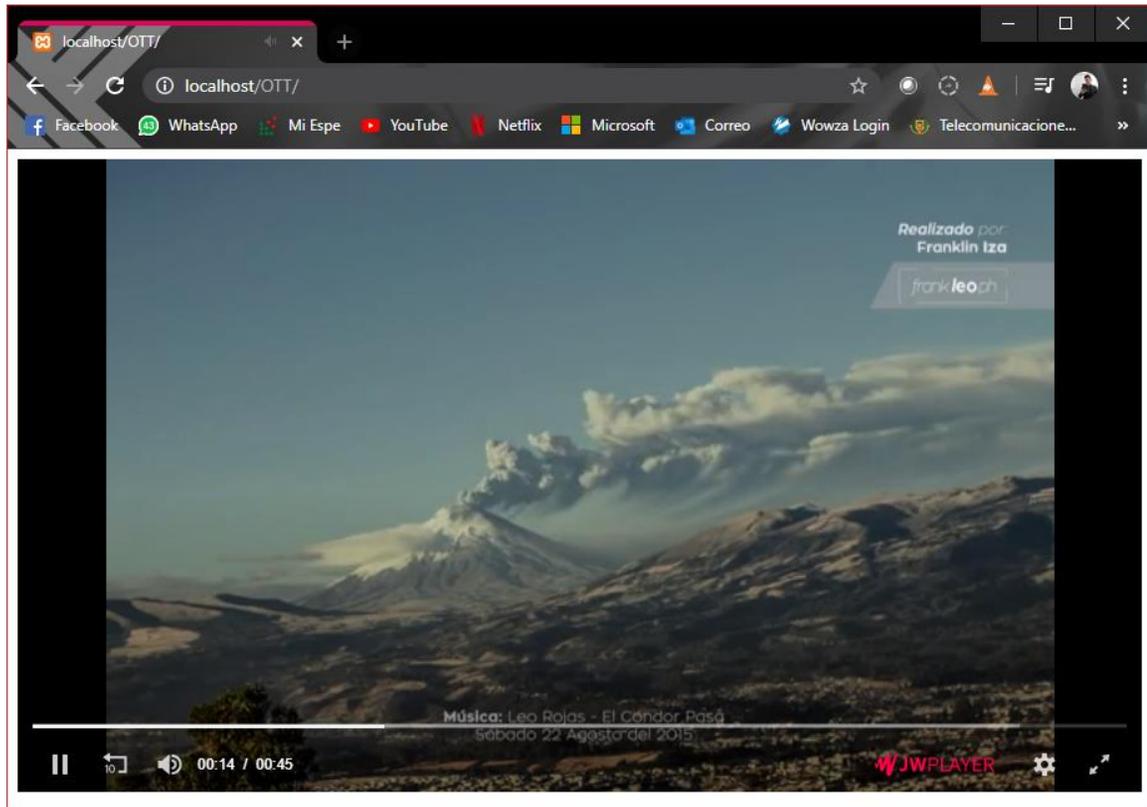
Para comprobar que el reproductor funciona y es compatible con el servidor, se implementó una página web sencilla para reproducir un video, ingresando como origen o fuente del video a la dirección URL que hace referencia al video, para esto se utilizó el siguiente código.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Plataforma OTT</title>
    <script
src="https://cdn.jwplayer.com/libraries/9IaQ9Hw7.js"
type="text/javascript"></script>
    <script type="text/javascript">
jwplayer.key="XXXXXXXX"</script>
    <script src="js/index.js" type="text/javascript"></script>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div id="volcan"></div>
        <script type="text/javascript">
          jwplayer('volcan').setup({
            file:
'http://192.168.0.108:1935/vod/smil:OTTvolcan.smil/manifest.mpd',
            image: 'portadas/portada-volcan.jpg'
          });
        </script>
      </div>
    </div>
  </body>
</html>
```

La llave suministrada se la puede hallar dentro del perfil registrado en el sitio oficial, dirigiéndose a la sección “Credenciales API” y se la inserta en “jwplayer.key” del código anterior, mediante esta licencia se activará la herramienta y se podrá visualizar el reproductor incrustado en la página web desarrollada, tal como se observa en la *Figura 55* mostrada a continuación.

Figura 55

Captura de la página web resultante del código para reproducir un video



El reproductor se despliega en la ventana del navegador y se adapta al tamaño que tenga dicha ventana, posee un control total del video para reproducirlo, pausarlo, retrocederlo o adelantarlo, adicional posee un botón de configuración para cambiar la calidad en la que se visualiza el video, sin embargo, se encuentra activado en modo automático, con el fin de evidenciar el streaming adaptativo que realiza la plataforma.

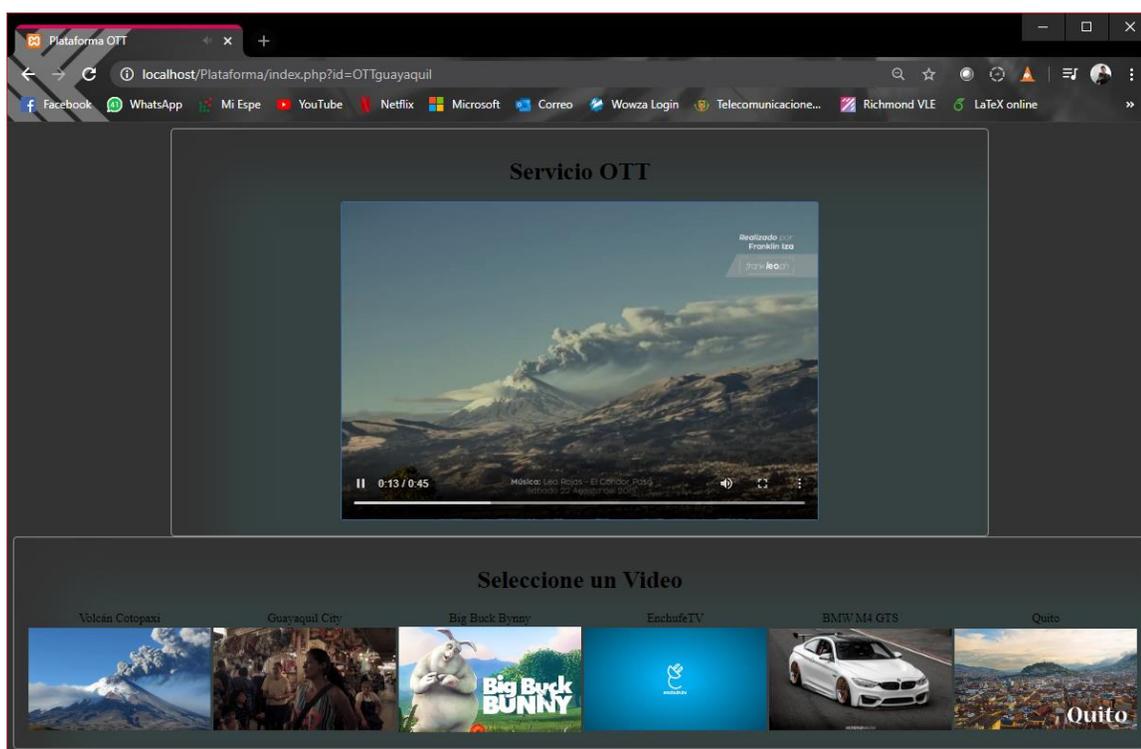
3.5.2.3 Diseño Final de la Plataforma OTT

Con un reproductor funcional incorporado en una página web y compatible con el servidor de streaming, es hora de programar un algoritmo que muestre portadas de los videos disponibles y se los pueda visualizar en el reproductor, sin la necesidad de

actualizar la página web y evitar el ingreso manual de las direcciones URL de cada video, algo similar a lo que se expone la *Figura 56* a continuación.

Figura 56

Captura de la página web resultante del algoritmo implementado



Consiguiendo un código que ejecute este procedimiento, finalmente se le puede dar un estilo más profesional a la página web de la plataforma y un diseño mucho más amigable para los usuarios, gracias a los expertos de la programación y el desarrollo web, en Internet existe un sin número de plantillas creadas para brindar un estilo sofisticado a las páginas web de una forma mucho más rápida, posibilitando a quien las utilice el poder de retocar el diseño a su gusto. En la *Figura 57* se muestra una captura del diseño final de la web implementada para la plataforma OTT.

Figura 57

Diseño final de la plataforma OTT implementada vista en un PC

Sábado, 25 de Abril de 2020 VoD Laboratorio de Televisión Digital, Departamento de Eléctrica, Electrónica y Telecomunicaciones.

PLATAFORMA OTT

Realizada por Franklin Iza

Música: Leo Rojas - El Condor Pasa, Sábado 22 Agosto del 2015

Más Videos

Guayaquil City Rocola Bacalao

Big Buck Bunny Open Movie Project

Enchufe TV MicroYapa: Monstruo

BMW M4 GTS Hurlin Foelling

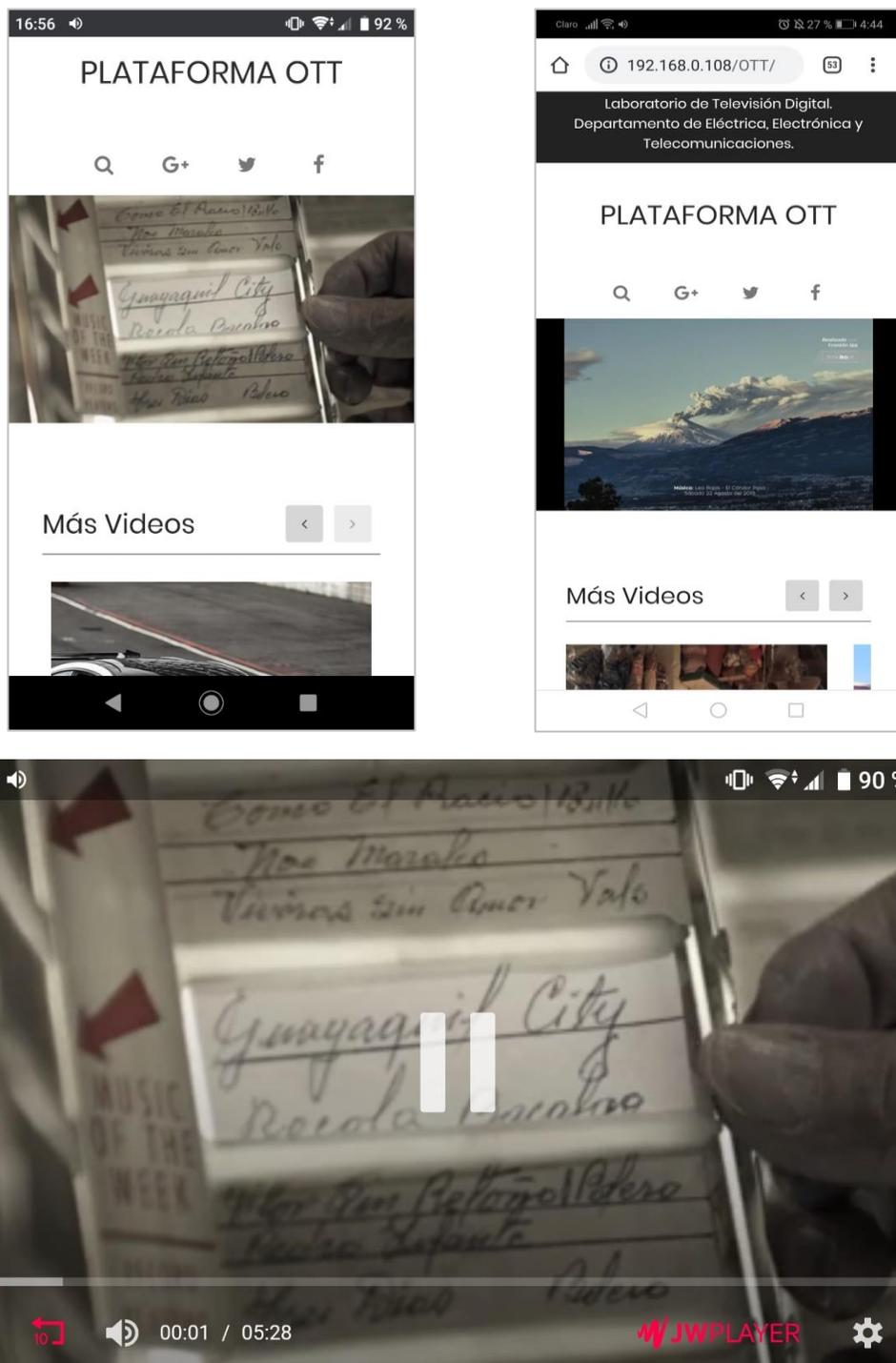
© Copyright 2020, Todos los derechos reservados. Designed by EDI Mateus. Sangolquí - Ecuador

Departamento de Eléctrica, Electrónica y Telecomunicaciones. Universidad de las Fuerzas Armadas - ESPE

Gracias al lenguaje HTML5 utilizado en este diseño, la página y el reproductor se adaptan a cualquier tamaño de pantalla, en la *Figura 57* se muestra una captura de la plataforma desde un PC y en la *Figura 58* capturas desde varios dispositivos móviles.

Figura 58

Capturas de la plataforma OTT desde varios dispositivos móviles



Capítulo IV

4. Pruebas y Análisis de Resultados

4.1 Validación del Contenido Transmitido

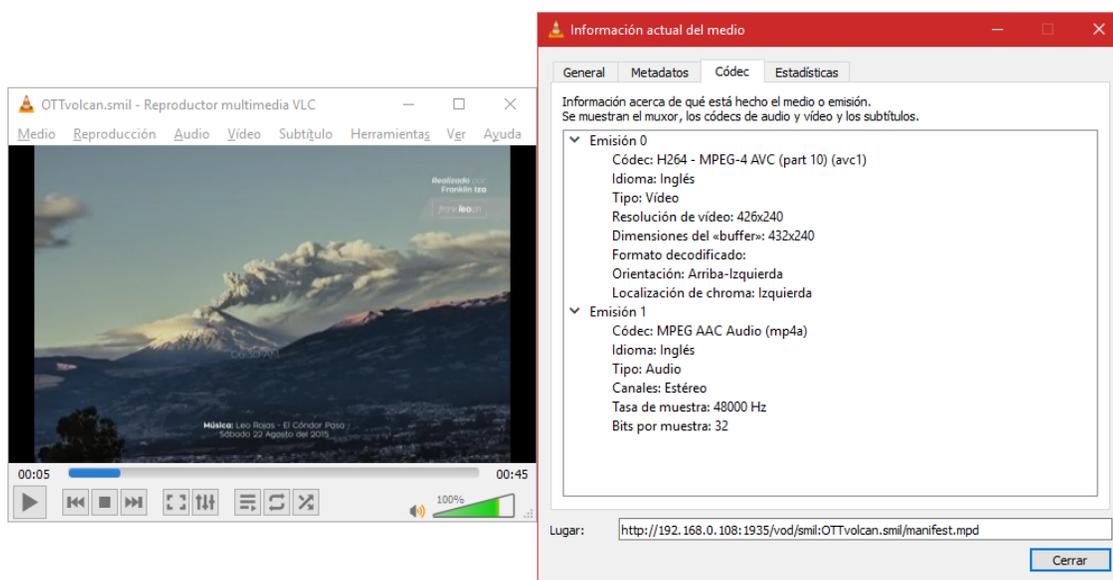
Para verificar que la transmisión se adapta en función de las condiciones del canal de los usuarios y emplea los archivos procesados y alineados a los niveles de codificación establecidos, a continuación, se muestran resultados de distintas pruebas de funcionamiento utilizando los métodos de acceso diseñados en la implementación.

4.1.1 Acceso mediante VLC Media Player

Mediante el primer método de acceso, siendo VLC Media Player un software de código abierto, se puede obtener parámetros correspondientes a la transmisión, para esto se debe reproducir un video disponible desde “Abrir Ubicación de Red”.

Figura 59

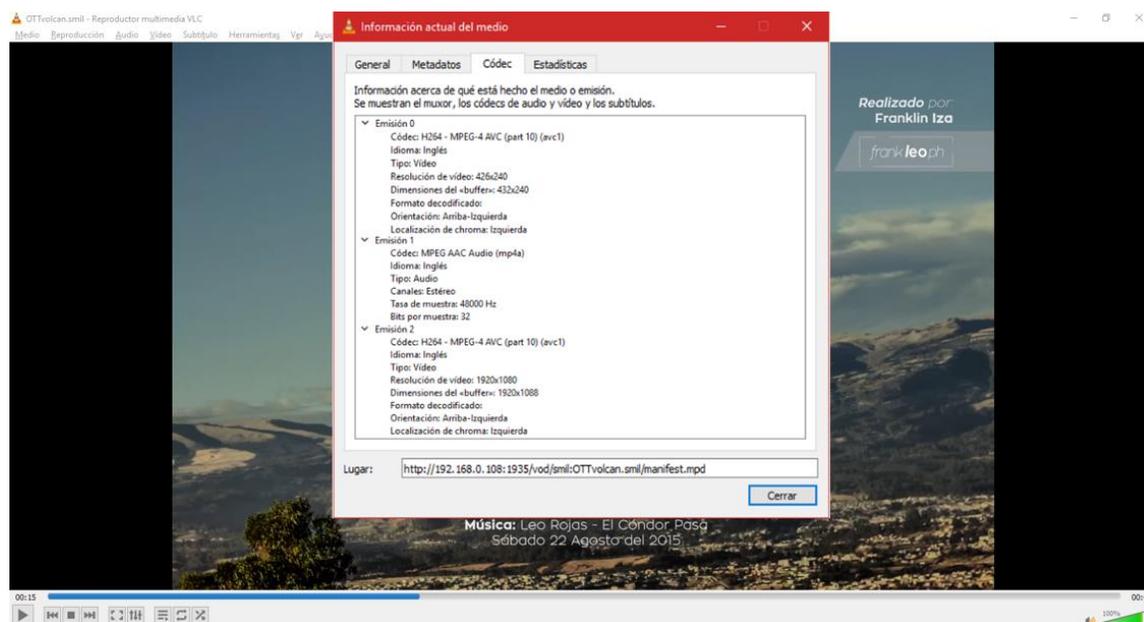
Información en tiempo real de contenido reproducido en VL



Para percibir mejor la adaptación automática de la resolución del video, se debe “Minimizar” el tamaño de la ventana de VLC, entonces en el instante en que el video sea reproducido, la ventana se ajustará de acuerdo al tamaño del contenido que se está visualizando, tal como se observa en la *Figura 59*, adicional, hay que dirigirse a la pestaña “Herramientas” y seleccionar “Información del Códec”, lo que desplegará una nueva ventana con información en tiempo real del contenido, y aquí es donde se puede identificar que el códec del video reproducido es H.264 MPEG-4/AVC y su resolución está en 426x240p, siendo esta la calidad más baja posible mostrada en los primeros segundos después de iniciar el video, también se muestra el códec AAC de audio.

Figura 60

Captura de VLC e información en tiempo real después de unos segundos



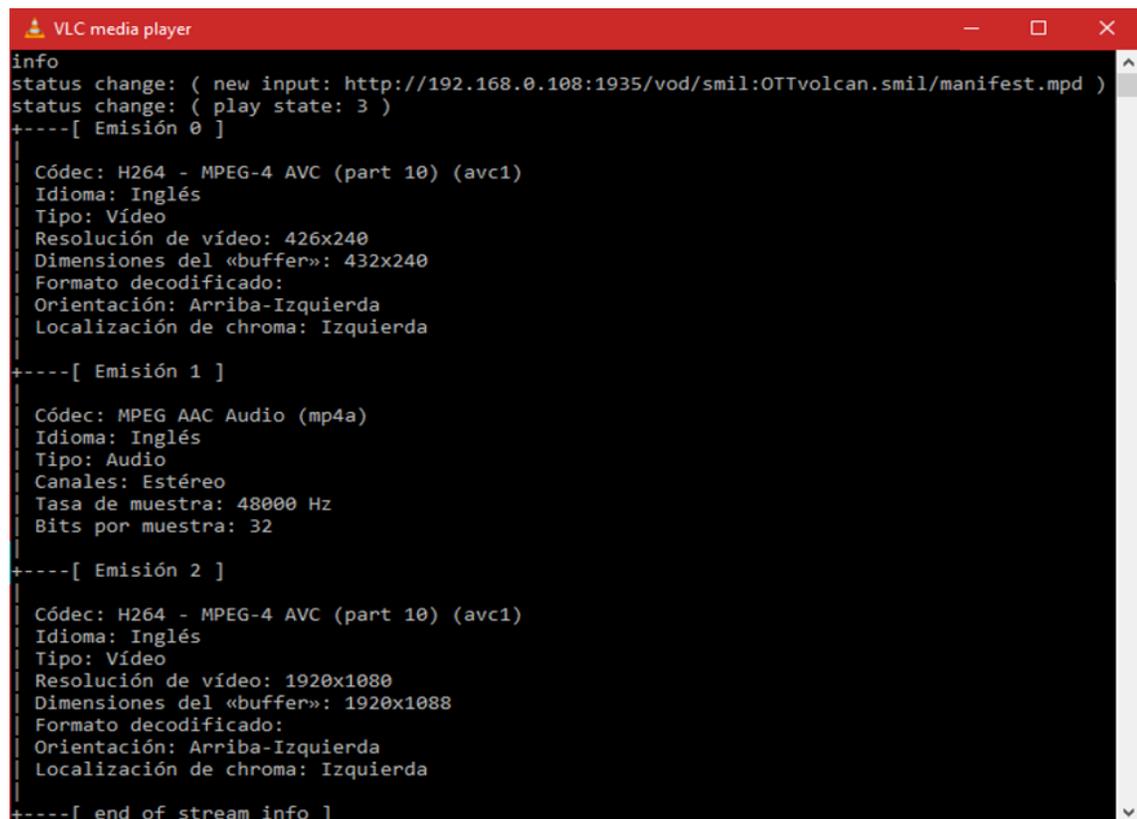
Después de unos segundos, la ventana de VLC cambia de tamaño de manera automática tal como se muestra en la *Figura 60* y se ajusta al tamaño del video recibido, en la que se aprecia una mejor calidad de la imagen y esto es verificado en la ventana

de información, donde el códec recibido se mantiene con H.264 y la resolución sube a 1920x1080p. Este proceso verifica el cumplimiento de las características de adaptación que un servicio OTT debe ofrecer, desplegando el video en su menor calidad posible al instante en que se lo reproduce y después de unos segundos, el contenido se adapta al canal del usuario, reproduciendo dicho video con una mejor calidad posible.

Otra manera de verificar el proceso adaptativo que se realiza es por medio de la consola de VLC, entonces mientras el video se esté reproduciendo, debe seleccionar la pestaña “Ver” y dirigirse a la opción “Añadir Interfaz”, aquí se elige “Consola” y en esta ventana se ingresa el comando “info”, mostrando información similar a la *Figura 61*.

Figura 61

Información en tiempo real del contenido mediante la consola de VLC

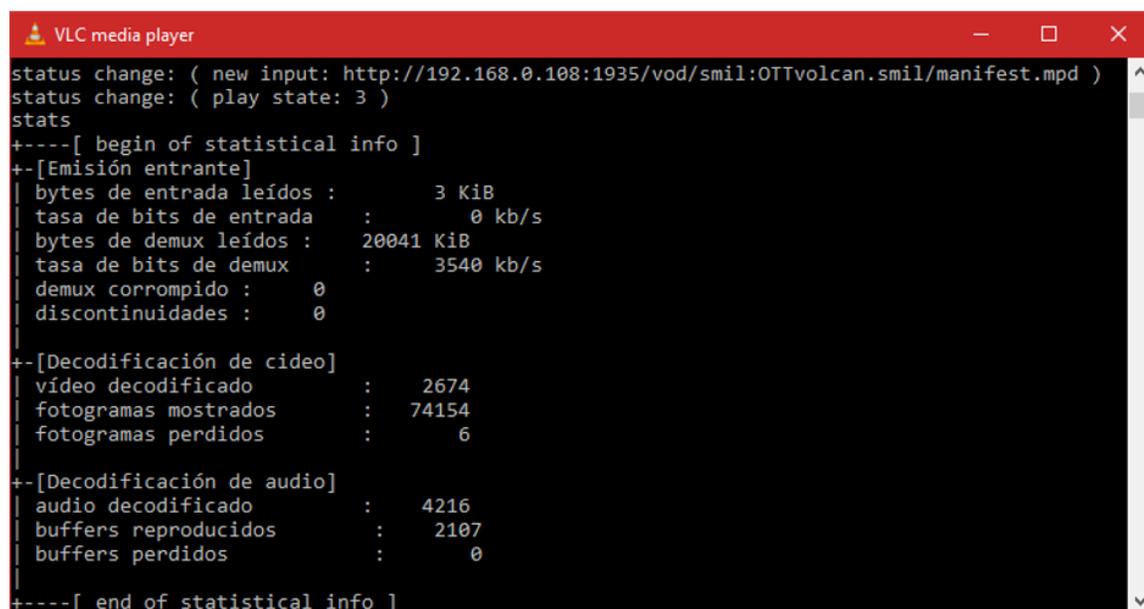


```
info
status change: ( new input: http://192.168.0.108:1935/vod/smil:OTTvolcan.smil/manifest.mpd )
status change: ( play state: 3 )
+----[ Emisión 0 ]
|
|  Códec: H264 - MPEG-4 AVC (part 10) (avc1)
|  Idioma: Inglés
|  Tipo: Vídeo
|  Resolución de vídeo: 426x240
|  Dimensiones del «buffer»: 432x240
|  Formato decodificado:
|  Orientación: Arriba-Izquierda
|  Localización de chroma: Izquierda
|
+----[ Emisión 1 ]
|
|  Códec: MPEG AAC Audio (mp4a)
|  Idioma: Inglés
|  Tipo: Audio
|  Canales: Estéreo
|  Tasa de muestra: 48000 Hz
|  Bits por muestra: 32
|
+----[ Emisión 2 ]
|
|  Códec: H264 - MPEG-4 AVC (part 10) (avc1)
|  Idioma: Inglés
|  Tipo: Vídeo
|  Resolución de vídeo: 1920x1080
|  Dimensiones del «buffer»: 1920x1088
|  Formato decodificado:
|  Orientación: Arriba-Izquierda
|  Localización de chroma: Izquierda
|
+----[ end of stream info ]
```

En la misma consola proporcionada por VLC, es posible ingresar el comando “stats” para desplegar información estadística en tiempo real de la transmisión, los bytes de entrada leídos constantemente, el número de fotogramas mostrados por el reproductor, los fotogramas perdidos durante la transmisión, la fuente de entrada del contenido y entre otros parámetros, los cuales pueden ser útiles para un monitoreo constante de la plataforma tal como muestra la *Figura 62*.

Figura 62

Información estadística de la transmisión mediante la consola de VLC



```
status change: ( new input: http://192.168.0.108:1935/vod/smil:OTTvolcan.smil/manifest.mpd )
status change: ( play state: 3 )
stats
+----[ begin of statistical info ]
+-[Emisión entrante]
| bytes de entrada leídos :      3 KiB
| tasa de bits de entrada  :      0 kb/s
| bytes de demux leídos   :    20041 KiB
| tasa de bits de demux   :     3540 kb/s
| demux corrompido       :      0
| discontinuidades       :      0
|
+-[Decodificación de video]
| video decodificado      :     2674
| fotogramas mostrados    :    74154
| fotogramas perdidos     :      6
|
+-[Decodificación de audio]
| audio decodificado      :     4216
| buffers reproducidos    :     2107
| buffers perdidos        :      0
|
+----[ end of statistical info ]
```

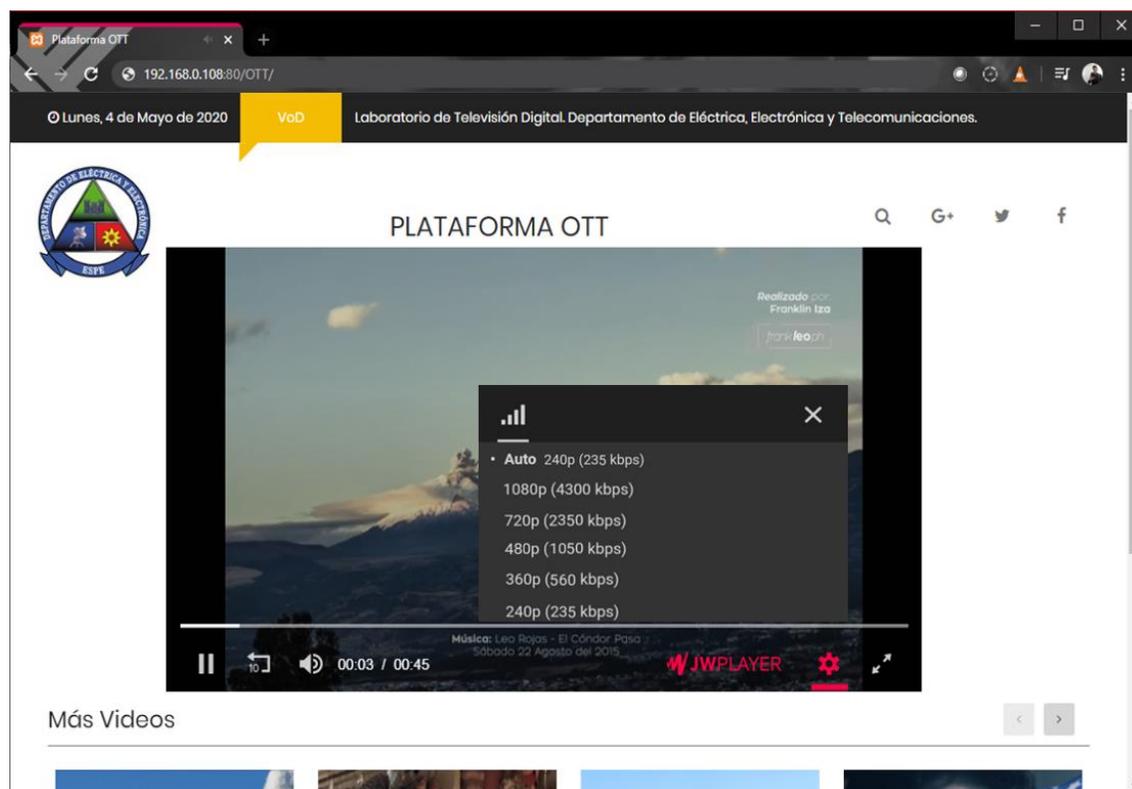
Gracias a VLC Media Player, es posible validar que los contenidos visualizados corresponden a los archivos alojados en el servidor, sin embargo, este software de reproducción no ofrece un control manual sobre las calidades disponibles de video, impidiendo verificar el funcionamiento de todos los niveles de codificación establecidos, y mediante herramientas de inyección de tráfico en la red, es posible manipular el estado del canal, lo que ocasionaría un cambio automático de calidad en el video.

4.1.2 Acceso mediante la Plataforma Web (JW Player)

Mediante el segundo método de acceso, el reproductor JW Player implementado en una página web, proporciona un mejor control de los niveles de calidad, ofreciendo a los usuarios la posibilidad de elegir la calidad en la que se desea visualizar el contenido de la plataforma. En la *Figura 63* se observa a la plataforma desde el navegador de una PC y al reproducir el contenido, este es visualizado en su menor calidad posible (porque la opción de “Calidad Auto” en el reproductor se encuentra activada por defecto), y en pocos segundos este se adapta al canal y cambia a la mejor resolución posible, siendo un proceso autónomo que interviene directamente en la experiencia del usuario.

Figura 63

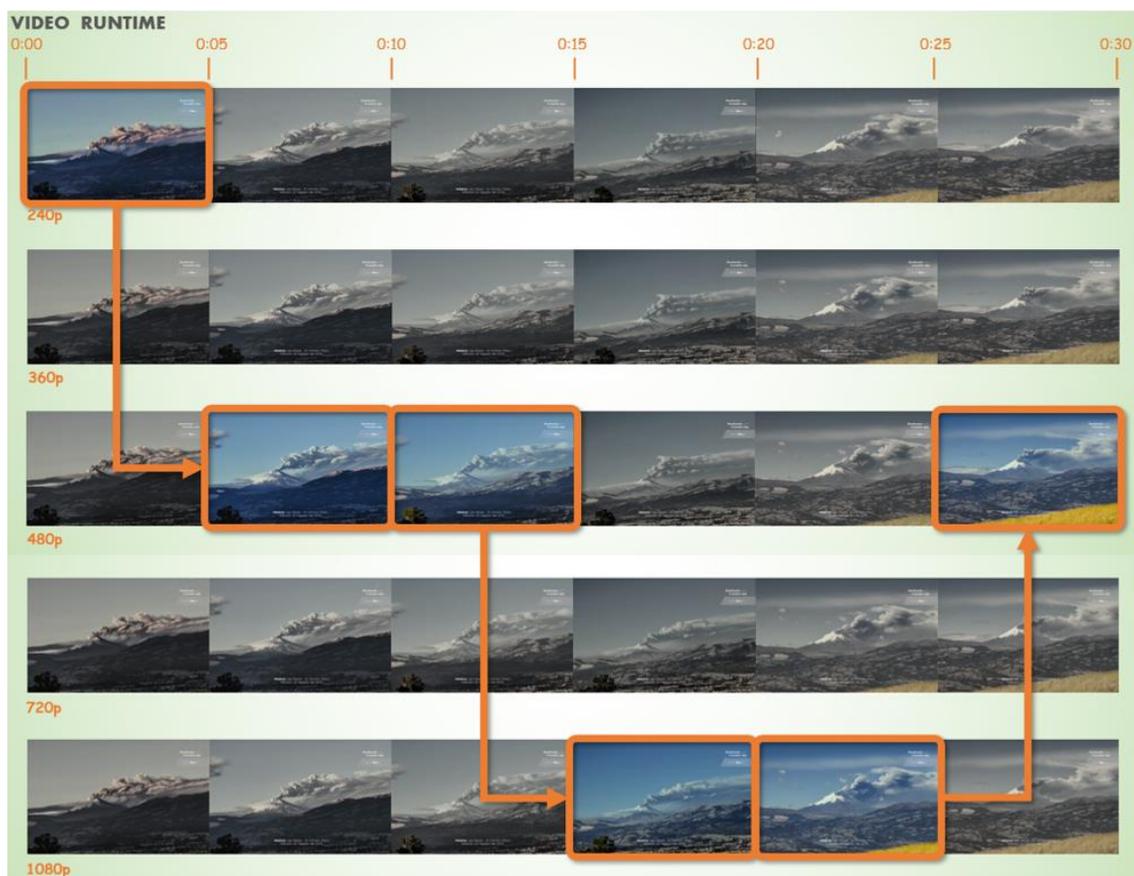
Captura de la plataforma y ajuste de calidad en el reproductor



En la *Figura 64* se observa los posibles cambios de calidad del video mientras es reproducido, los fotogramas I del video son los puntos de referencia del video y la distancia entre estos fotogramas puede ser fija o variable, dependiendo del GOP definido al momento de codificar el contenido, sin embargo, mediante pruebas de funcionamiento realizadas, al reproducir videos codificados con un GOP variable, en ocasiones se desincroniza los niveles de calidad de los videos, perdiendo la continuidad de la transmisión al no hallar oportunamente los puntos de referencia de los archivos.

Figura 64

Posible flujo reproducido en los distintos niveles de calidad del video



Los fotogramas I son el punto de partida de nuevas secuencias de video y el tiempo que demore en generarse un cambio automático de calidad dependerá estrictamente del estado en que se encuentre el canal, si dicho canal está apto para recibir mayor cantidad de información, el reproductor negocia con el servidor para que transmita el contenido en su mejor calidad posible.

Para verificar que todos los niveles de calidad funcionan, en la *Figura 65* se observa que con el botón de “Ajustes” en el reproductor, es posible cambiar la opción de calidad “Auto” y evidenciar uno a uno los distintos niveles de calidad disponibles mientras el contenido es reproducido.

Figura 65

Variación de la calidad del video reproducido en un dispositivo móvil

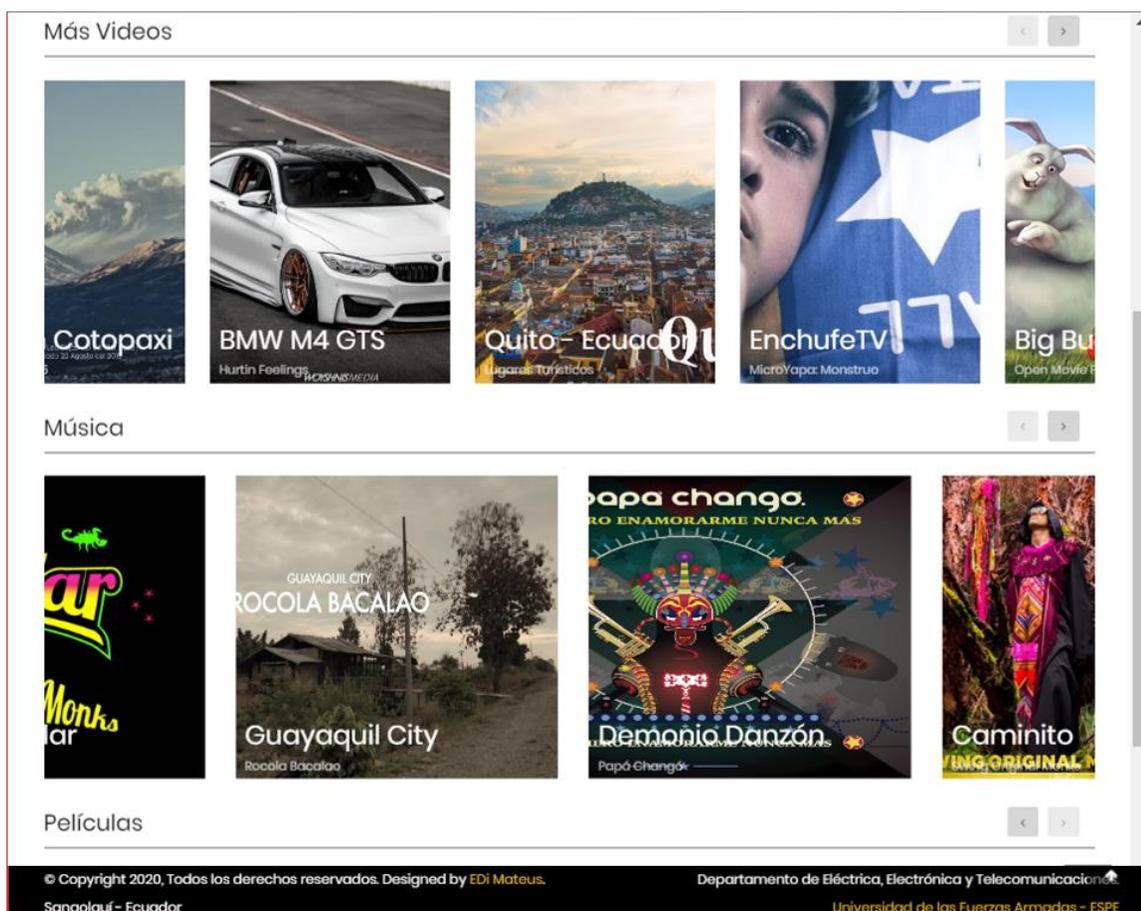


Para que la plataforma adquiriera un mayor interés por los usuarios, es necesario cargar nuevos contenidos continuamente, poniendo a disposición de los usuarios un

gran catálogo de videos a su elección, esto evitaría que el servicio ofrecido se vuelva monótono y aburrido. Dichos contenidos deben ser previamente procesados, codificados y configurados en el servidor de streaming, los cuales se reproducirán en su mejor calidad posible en función del ancho de banda que disponen los usuarios, mejorando enormemente su experiencia desde cualquier dispositivo conectado a la plataforma. La variedad de contenidos que se encuentran procesados y disponibles en la plataforma OTT implementada se puede observar en la *Figura 66* a continuación.

Figura 66

Variedad de contenidos disponibles en la plataforma implementada



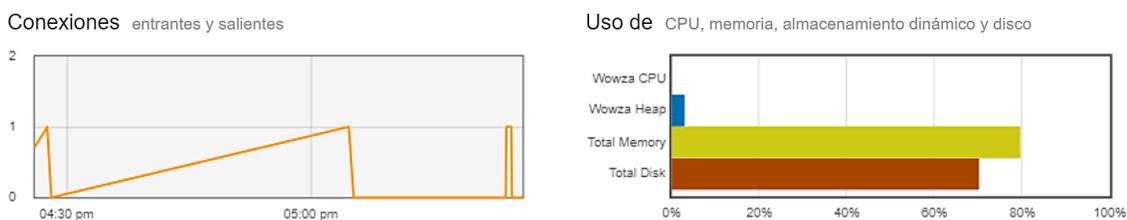
4.2 Desempeño del Servidor

Para evaluar el desempeño del servidor, se utiliza herramientas gráficas que proporciona el mismo software de Wowza, desde que inicia el servicio empieza un monitoreo constante del servidor que permite determinar las conexiones entrantes y salientes del mismo, útil para identificar la cantidad de usuarios que acceden a la plataforma y conocer el horario de mayor audiencia, también es posible monitorear la cantidad de recursos del CPU utilizados, el tiempo de funcionamiento del servicio, el espacio de almacenamiento que ocupa el servidor en el disco y la cantidad de bits entrantes y salientes en cada instante de tiempo, similar al gráfico de la *Figura 67*.

Figura 67

Gráficos en tiempo real que monitorean el servidor de Wowza

Estado



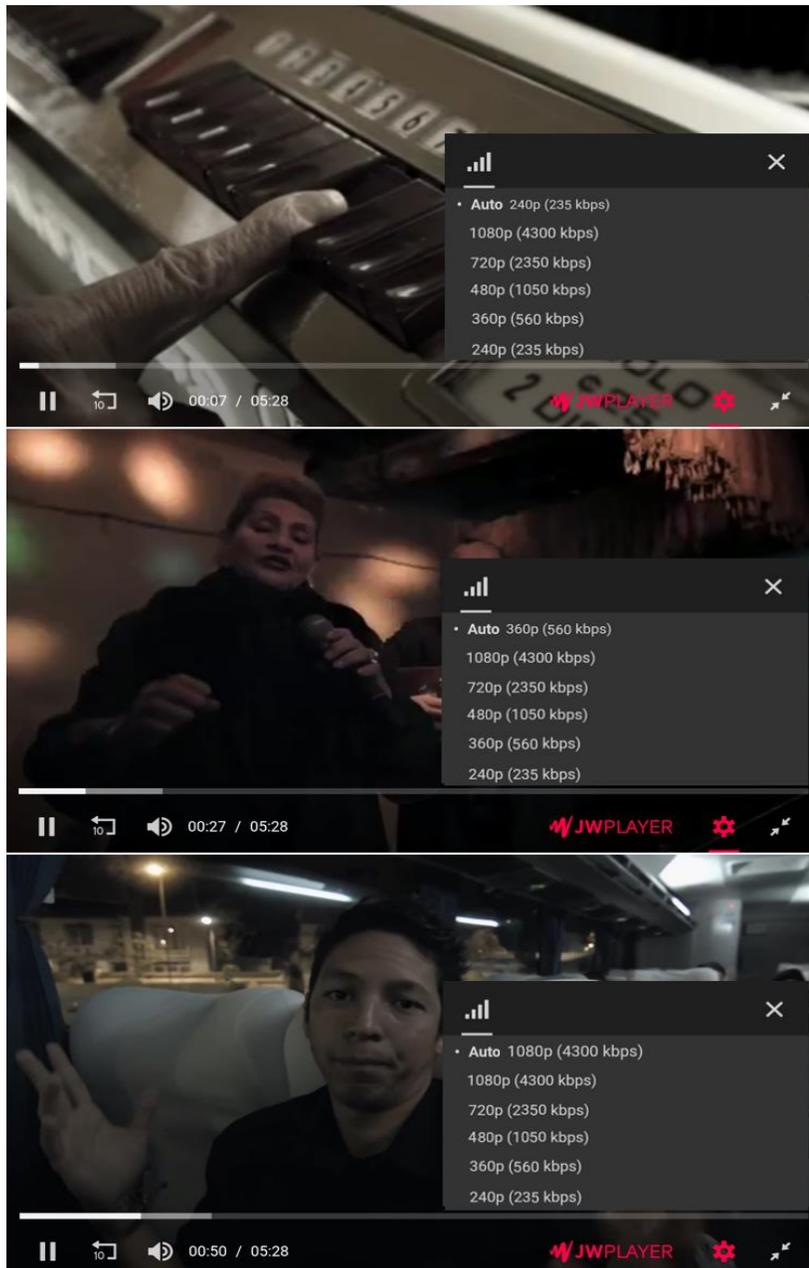
Para evidenciar la labor que cumple el servidor al momento de atender a los usuarios, se ha establecido dos escenarios de pruebas, en el primer escenario el servidor estará encargado de atender las peticiones de un solo usuario en la red durante un intervalo de 5 minutos, en el segundo escenario con un mismo intervalo de tiempo, el servidor deberá atender las peticiones de un mayor número de dispositivos conectados en la red de forma simultánea, mediante una comparativa de estos dos escenarios, es posible evaluar el desempeño del servidor implementado.

4.2.1 Escenario 1: Atendiendo a un solo Usuario

Con un dispositivo móvil se accedió al contenido, evidenciando lo siguiente:

Figura 68

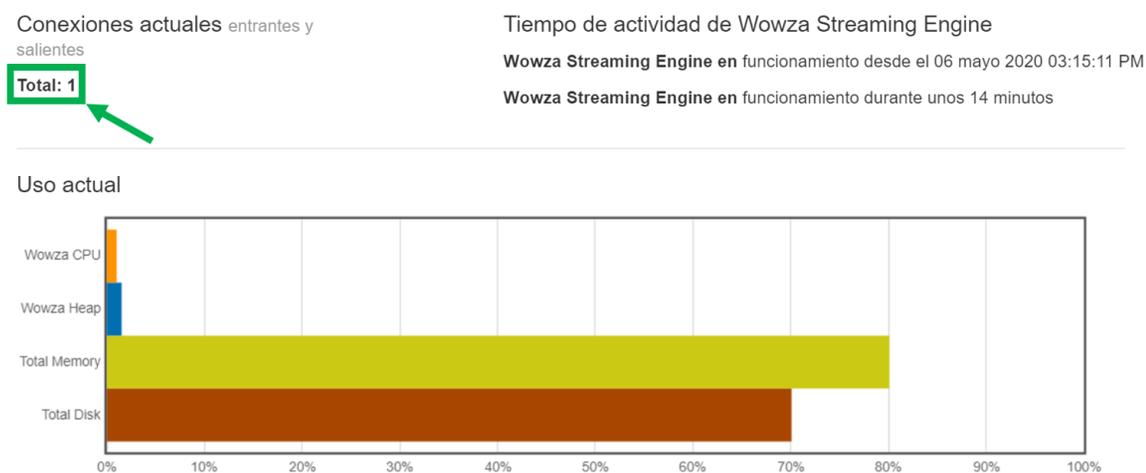
Cambio automático de calidad evidenciado



Para observar los cambios generados durante la transmisión, se desplegó la pestaña de calidades dentro de la opción “Ajustes” del reproductor, este listado se lo mantuvo abierto mientras el video seguía reproduciéndose, lo que permitió verificar la calidad de video que empezó de forma instantánea en 240p, pasado unos segundos cambió automáticamente su calidad a 360p y posteriormente, se evidenció un mayor salto en su calidad llegando a 1080p. Debido a que un solo dispositivo es atendido por el servidor y los demás dispositivos dentro de la red se encontraban en modo de suspensión, se deduce que el canal estaba disponible en su totalidad, por lo cual el dispositivo conectado a la plataforma pudo negociar con el servidor para que el contenido receptado tenga una mejor calidad posible.

Figura 69

Uso actual del servidor mientras atiende a un solo usuario



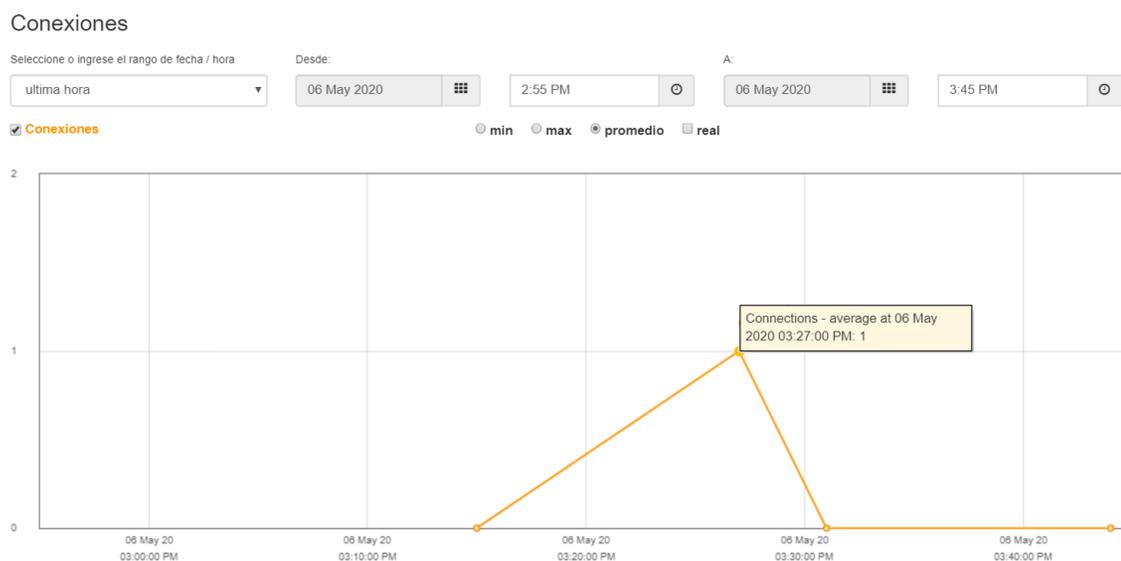
Mientras tanto, el servidor presentó un mínimo esfuerzo tal como se muestra en la *Figura 69*, el espacio en el disco se mantuvo constante y la memoria presentó mínimas variaciones en cada instante. A continuación, se describen los indicadores con sus distintos gráficos en tiempo real observados en este escenario de pruebas.

4.2.1.1 Conexiones en el Servidor

Mediante el gráfico de la *Figura 70*, es posible determinar el número de conexiones entrantes y salientes en cada momento, durante la última hora, semana o mes, lo que permite determinar con gran facilidad los horarios en los que mayor uso de la plataforma le dan los usuarios, además de conocer el mínimo y máximo número de conexiones, y la cantidad promedio de conexiones.

Figura 70

Número de conexiones reales en el servidor



A través de este gráfico se pudo evidenciar un pico generado por el dispositivo conectado al servidor y al ser una sola conexión, el valor real coincide con el valor promedio de conexiones en el servidor.

4.2.1.2 Tráfico en la Red

El tráfico es monitoreado constantemente en términos de bits entrantes y salientes, y al ser una plataforma que ofrece servicios de VoD no existirán bits entrantes al servidor, debido a que el contenido no es generado en vivo y deba ser procesado para su difusión, por el contrario, el contenido ya está alojado en el servidor, procesado y listo para su difusión hacia los distintos dispositivos de los usuarios que lo soliciten.

Figura 71

Cantidad de bits salientes en el servidor de la plataforma



En el instante en que el usuario accede al servidor se empieza a transferir datos en cada momento, siempre y cuando la conexión se la mantenga, en la *Figura 71* se puede evidenciar una cantidad mayor de bits apenas empieza la transmisión con un valor promedio de 760 kbps, mientras que, al desplegar el gráfico real, el pico inicial alcanzó los 2,3 Mbps aproximadamente y después de establecerse la conexión se observa una variación constante de bits que va en aumento pero no supera el 1 Mbps.

4.2.1.3 Estado del CPU

El rendimiento del servidor dependerá de la cantidad de recursos que este utiliza, los recursos son asignados automáticamente en función del número de usuarios que deben ser atendidos, por esta razón, grandes plataformas que ofrecen servicios OTT, utilizan granjas de servidores los cuales funcionan en paralelo, con el fin de atender la mayor cantidad de usuarios posible en todo el mundo.

Figura 72

Porcentaje de CPU utilizado por el servidor



Durante el período de prueba, en la *Figura 72* se pudo observar un pico que alcanzó el 6%, esto sucedió en el instante en que el dispositivo estableció la conexión con el servidor, posterior a esto, la conexión se la mantuvo y la cantidad de recursos que utilizó el servidor para transmitir contenidos bajó considerablemente y no superó el 3% de recursos del CPU.

4.2.2 Escenario 2: Atendiendo a varios Usuarios Simultáneos

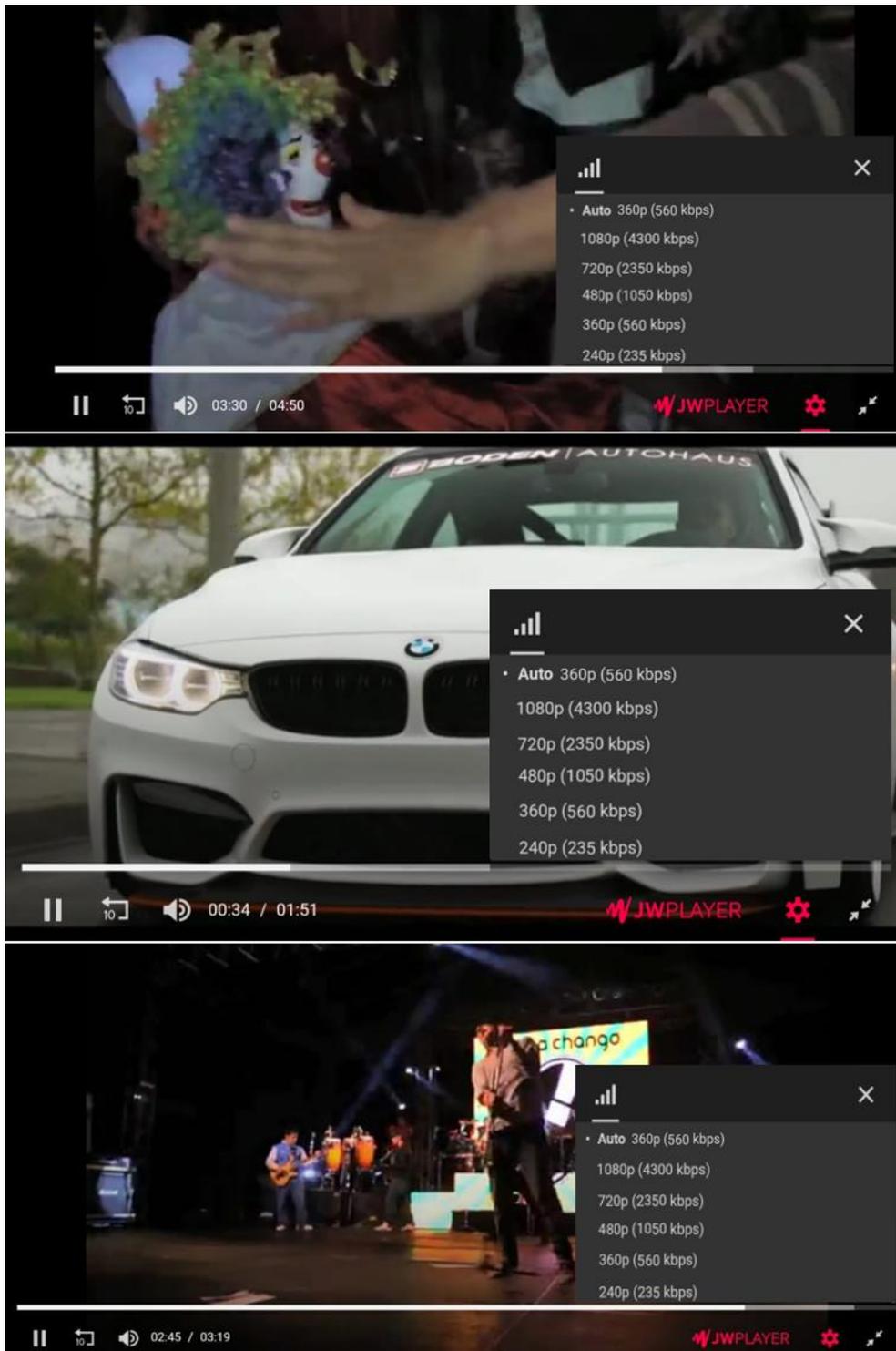
Se utilizó 6 dispositivos para acceder a la plataforma y durante el período de prueba se pudo evidenciar que la calidad del contenido que fue reproducido en todos los dispositivos se mantuvo en 360p, tal como se observa en la *Figura 73*.

Los usuarios que se conectaron al servidor de manera simultánea, los cuales manifestaron que el contenido seleccionado por cada uno de ellos empezó instantáneamente, pero se mantuvo constante en la calidad de 360p durante el tiempo en que todos los dispositivos se encontraban conectados, esto se debe a que, al existir mayor cantidad de dispositivos en la misma red, el canal de cada uno de ellos es menor comparado al canal del escenario donde un solo dispositivo fue atendido que se observó en la *Figura 68*.

Sin embargo, dependiendo de la duración de los videos seleccionados por cada usuario, al terminar la reproducción de los videos, poco a poco iban desconectándose usuarios del servidor y los últimos dispositivos que quedan conectados empezaron a negociar una mayor recepción de datos con el servidor, lo que ocasionaría una mejora instantánea en el video reproducido por dichos usuarios.

Figura 73

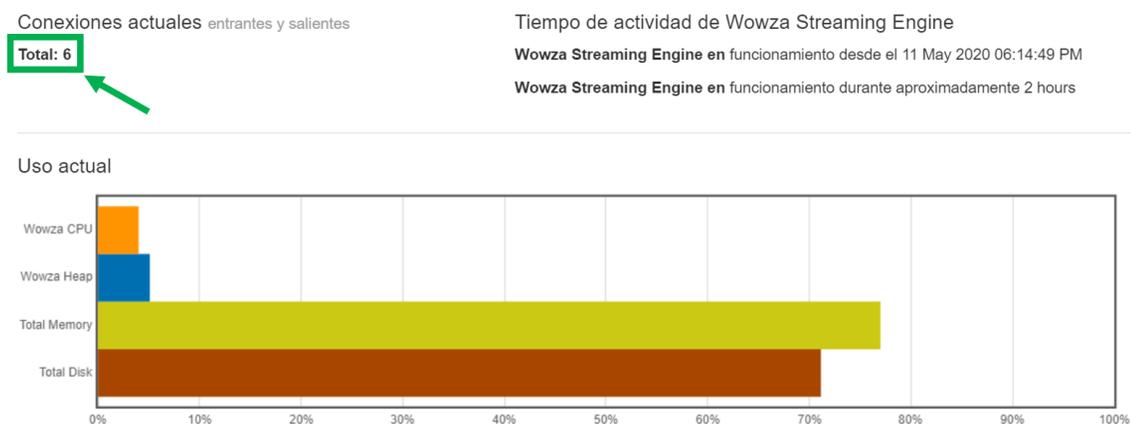
Calidad establecida a 360p en todos los dispositivos



En la *Figura 74* se observa que el servidor presentó pequeñas variaciones en el porcentaje de CPU utilizado durante el tiempo que duró la prueba, y a continuación, se describen los respectivos gráficos evidenciados en este escenario.

Figura 74

Uso actual del servidor mientras atiende a varios dispositivos



4.2.2.1 Conexiones en el Servidor

En la *Figura 75* se puede observar que las 6 conexiones simultaneas corresponden a los dispositivos de cada usuario que participaron durante esta prueba, los cuales accedieron a la plataforma de manera simultánea, lo que afecta directamente a los distintos gráficos en tiempo real que proporciona el servidor (el gráfico del tráfico cursado por la red y el gráfico del porcentaje de recursos del CPU utilizados).

Al existir una mayor cantidad de usuarios, los valores promedio que presenta cada gráfico tendrán una evidente variación respecto a los valores reales, por este motivo, se comparan ambos gráficos a continuación.

Figura 75

Número de conexiones promedio y reales en el servidor

Conexiones

Seleccione o ingrese el rango de fecha / hora

Desde:

11 May 2020

8:00 PM

A:

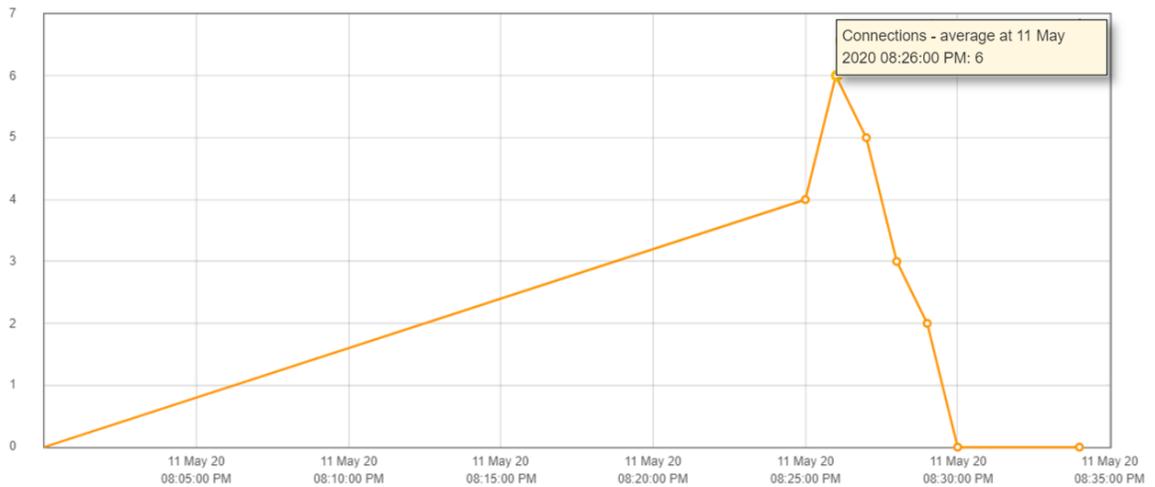
11 May 2020

8:35 PM

ultima hora

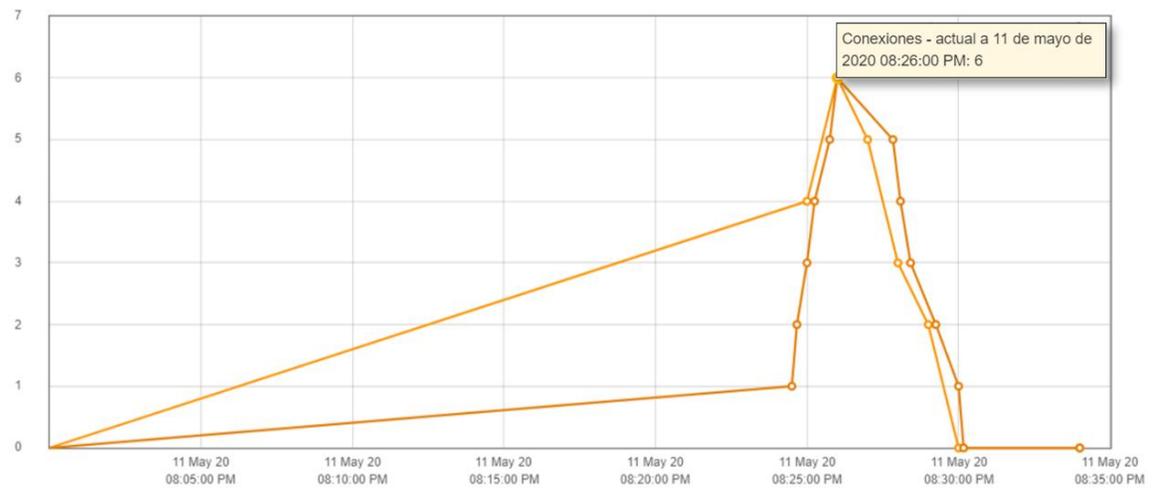
Conexiones

min max promedio real



Conexiones

min max promedio real



4.2.2.2 Tráfico en la Red

En este escenario de pruebas, la *Figura 76* mostrada a continuación presenta un valor promedio que alcanzó los 9,32 Mbps con los 6 dispositivos conectados.

Figura 76

Cantidad de bits salientes promedio y reales en el servidor



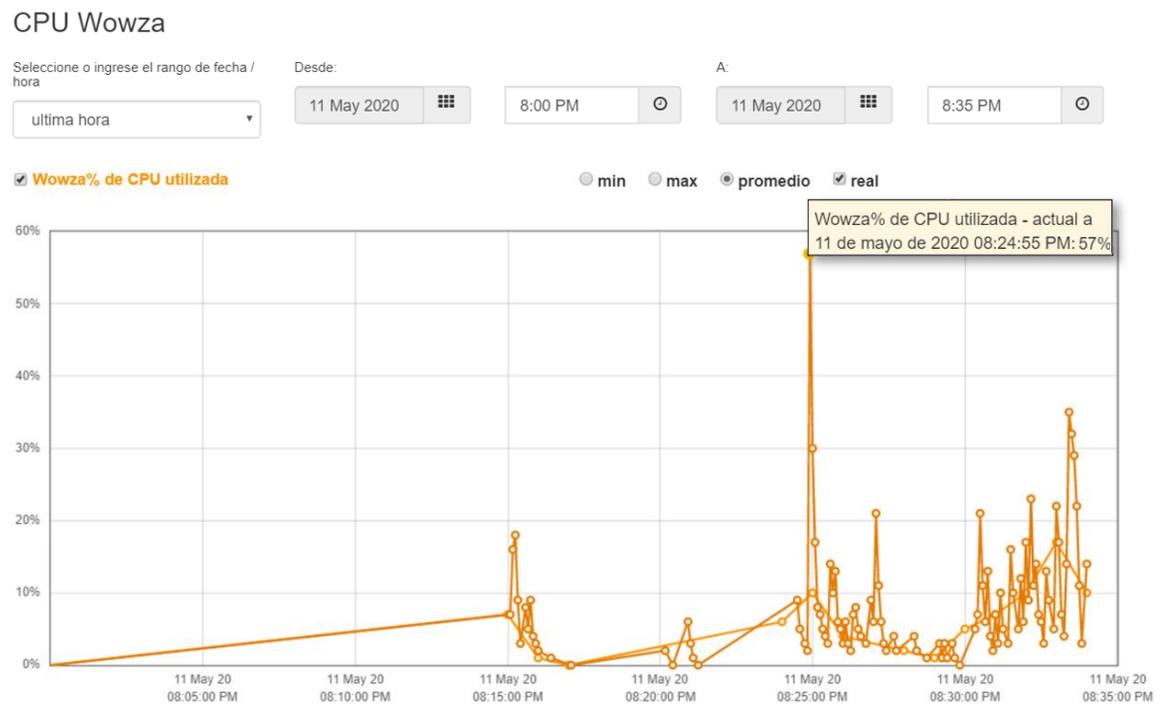
Mientras tanto, al activar el gráfico real se observa un pico en cada instante en que los dispositivos establecen la conexión, evidenciando un mayor tráfico soportado en el momento en que el último usuario establece la conexión alcanzando los 12,71 Mbps.

4.2.2.3 Estado del CPU

Al mantener la conexión simultánea con los 6 dispositivos, el servidor logra alcanzar un valor promedio del 10% de recursos del CPU utilizados.

Figura 77

Porcentaje promedio y real de CPU utilizado por el servidor



El gráfico real evidencia un pico que llega al 57% de recursos utilizados en el instante en que se establecen todas las conexiones, después de estabilizarlas se presentan constantes variaciones que no superan el 25% del CPU utilizado.

4.2.3 Análisis y Comparación de los Escenarios

En la *Tabla 9*, se resumen los resultados logrados mediante los escenarios de pruebas realizados con el fin de evidenciar el desempeño que presenta el servidor.

Tabla 9

Comparación de resultados de los escenarios de pruebas

Parámetros	Escenario 1	Escenario 2
Número de conexiones simultáneas	1 conexión	6 conexiones
Calidad entregada del contenido	240p, 360p y 1080p	360p
Tráfico promedio alcanzado	760 kbps	9,32 Mbps
Tráfico máximo logrado al establecer las conexiones	2,3 Mbps	12,71 Mbps
Porcentaje promedio de CPU utilizado	6%	10%

Por causa de la pandemia del Covid-19 que sufre el mundo entero, no fue posible realizar pruebas en el laboratorio con una cantidad significativa de dispositivos conectados a la plataforma, sin embargo, mediante las pruebas realizadas con un máximo de 6 dispositivos, se demuestra la capacidad que el servidor posee al momento de atender a varios usuarios simultáneos, transmitiendo todos los contenidos a los respectivos dispositivos de forma adaptativa sin afectación alguna, sin embargo, el escenario de pruebas es una red LAN doméstica y aquí participaron los integrantes de mi familia, quienes fueron sometidos a una evaluación según el criterio MOS (*Mean Opinion Score*), con el fin de evidenciar el desempeño y la experiencia que brinda la plataforma aquí implementada.

4.3 Evaluación MOS sobre el Desempeño de la Plataforma

Después de haber logrado una implementación exitosa de una plataforma que brinda los servicios OTT en una modalidad de VoD, es estrictamente necesario evaluar la calidad de la experiencia (QoE) de los usuarios que la utilizan, para esto, fue preciso realizar una encuesta basada en recomendaciones expuestas por la UIT-T P.910 (UIT-T, 1999), que hace referencia a métodos de evaluación subjetiva de la calidad de los videos en aplicaciones multimedios, donde se mide el desempeño de la plataforma utilizando una calificación de estrellas, enfocada a parámetros que pueden ser valorados por los usuarios. En la *Tabla 10* se muestran los parámetros utilizados para esta evaluación, junto con su mínima calificación obtenida por parte de los usuarios, debido a que la cantidad de usuarios encuestados no es significativa.

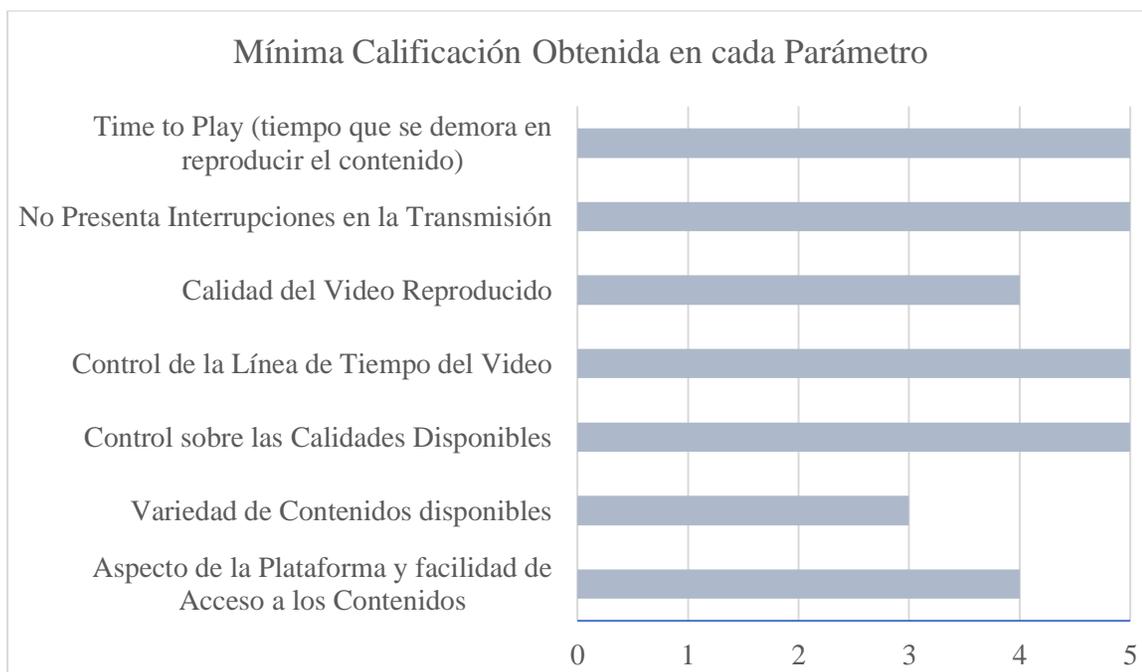
Tabla 10

Parámetros utilizados para la evaluación bajo el criterio MOS.

Parámetros	Mínima Calificación Obtenida	Valoración Numérica
Time to Play (tiempo que se demora en reproducir el contenido)	★ ★ ★ ★ ★	5
No presenta interrupciones en la transmisión	★ ★ ★ ★ ★	5
Calidad del video reproducido	★ ★ ★ ★ ☆	4
Control de la línea de tiempo del video	★ ★ ★ ★ ★	5
Control sobre las calidades disponibles	★ ★ ★ ★ ★	5
Variedad de contenidos disponibles en la plataforma	★ ★ ★ ☆ ☆	3
Aspecto de la plataforma web y facilidad para acceder a los contenidos	★ ★ ★ ★ ☆	4
	Valor Promedio	4,43

Figura 78

Mínima puntuación obtenidos mediante la evaluación MOS



Los resultados obtenidos mediante las 6 encuestas MOS realizadas, en su mayoría alcanzaron un valor promedio que se aproxima a 5, mientras que la mínima puntuación lograda en cada parámetro fue mostrada en la *Tabla 10* y *Figura 78* anteriormente mostradas, consiguiendo un valor promedio de 4,43 que pone en evidencia el desempeño de la plataforma desde el punto de vista de los usuarios y una QoE explícitamente satisfactoria, además se puede identificar los parámetros que deben ser analizados y mejorados previamente, como lo es la variedad de contenidos disponibles en la plataforma, el cual obtuvo una menor puntuación respecto a los demás parámetros, por consiguiente es fundamental la carga constante de nuevos contenidos en la plataforma, esto ocasionará un mayor interés de los usuarios para utilizarla y a la vez, atraería posibles usuarios que puedan gozar del servicio OTT ofrecido.

4.4 Video de las Pruebas Realizadas

Para evidenciar que el trabajo descrito en este documento fue realizado y los resultados que se alcanzaron fueron exitosos, debido a la situación sanitaria que atraviesa el país y el mundo entero a causa del Covid-19, se decidió grabar un video con las distintas pruebas de funcionamiento sometidas a la Plataforma OTT, el cual se encuentra alojado con libre acceso en la plataforma de YouTube. En la *Figura 79* se muestra el código QR que hace referencia a la URL del video, por lo que es necesario escanearlo con cualquier tipo de herramienta para la lectura de códigos QR.

Figura 79

Código QR del enlace para visualizar el video realizado



Capítulo V

5. Conclusiones y Recomendaciones

5.1 Conclusiones

- Con base al diseño de la plataforma OTT especificado en este documento, se demuestra que es viable la implementación de una plataforma con mayor escala, para ofrecer servicios de calidad a una mayor cantidad de usuarios, enfocándose en el mercado de la difusión de contenidos, además de impulsar el desarrollo tecnológico en el país, lo que reduciría el consumo excesivo de plataformas extranjeras y evitar grandes fugas de dinero.
- Por medio de una concreta descripción de los protocolos y principios utilizados en el streaming adaptativo, fue posible analizar algunos parámetros fundamentales para el proceso de adaptación, como lo es el valor del GOP y los fotogramas presentes en una secuencia de video, lo que permitió configurar de manera avanzada el códec y se pudo establecer un valor fijo del GOP durante la codificación de los contenidos, en este caso la duración de los segmentos fue fijado en 2 segundos, lo que solventó varios problemas de sincronismo en el flujo de video, asegurando un buen desempeño del servicio sin interrumpirse la transmisión.
- El desempeño del servidor durante las pruebas realizadas con múltiples usuarios fue exitoso, el cual soportó un tráfico máximo de 12,71 Mbps durante las 6 conexiones simultáneas y apenas el 10% de recursos del CPU fue utilizado, con esto se demuestra que la plataforma implementada puede soportar un mayor número de usuarios sin presentar inconvenientes en su rendimiento ni tampoco problemas de sincronismo en la secuencia de videos.

- Mediante la escalera de codificación establecida, es posible asegurar que la mayor parte de usuarios en el Ecuador puede gozar de un servicio de calidad y visualice contenidos HD pese a la inestabilidad de sus redes, gran mayoría de los hogares disponen anchos de banda que no sobrepasan los 5 Mbps lo que requiere contenidos adaptables a estas condiciones y que exijan menor tasa de recepción.
- La adaptación de los contenidos es evidenciada a simple vista y varía de forma automática de acuerdo con las condiciones del canal, además se verifica este proceso adaptativo mediante los ajustes del reproductor JW Player y la consola de VLC Player utilizados en la plataforma OTT, donde no presenta pausas ni discontinuidad en el instante en que cambia de calidad los videos.
- El desarrollo de la plataforma OTT descrita en este documento, puede contribuir en prácticas de laboratorio realizadas en asignaturas relacionadas con la Televisión Digital, donde los estudiantes podrán adquirir una visión más clara respecto a las arquitecturas implementadas por grandes plataformas que brindan servicios de streaming OTT y puedan experimentar de una manera más cercana con los protocolos y tecnologías del streaming adaptativo, que en el caso de esta implementación, se utilizó DASH para evitar deficiencias de interoperabilidad entre dispositivos de distintos fabricantes.
- La facilidad para acceder al contenido ofrecido por una plataforma de VoD y la variedad de los videos disponibles en esta, tienen un gran impacto en el interés que le ponen los usuarios a la plataforma, esto fue corroborado gracias a la valoración MOS lograda a través de los usuarios que participaron en las pruebas de la implementación, obteniendo una puntuación de 3 sobre 5 que evidencia la inconformidad con la variedad de contenidos disponibles y una puntuación de 4 sobre 5 en la facilidad para acceder a los mismos.

5.2 Recomendaciones

- Se recomienda establecer un valor fijo de GOP durante la codificación de las distintas calidades y así evitar problemas en el sincronismo de los flujos de video, a pesar de que la misma tecnología del códec genera automáticamente los fotogramas I, pero al establecer un GOP variable, en el instante en que se genera un cambio automático de nivel de calidad, el servidor no encuentra la secuencia correspondiente y la transmisión se interrumpe en su totalidad.
- Para mantener el interés por la plataforma y atraer nuevos usuarios a la misma, se recomienda una carga constante de contenidos, al igual que la mejora periódica el diseño web con el fin de simplificar los procesos de búsqueda de contenidos.
- La escalera de codificación establecida para esta implementación no es estricta para todas las plataformas, sino más bien es un ejemplo propuesto para utilizarlo basándose en un análisis que relaciona escaleras de codificación de otras plataformas populares y datos reales sobre la velocidad de conexión en los hogares ecuatorianos, si el proyecto es implementado en un país distinto al Ecuador se recomienda analizar datos correspondientes al país que lo desarrolle.
- Para mejorar el desempeño del servidor, se recomienda implementar el servicio de Wowza en un equipo diferente al del servidor Web, así los recursos disponibles del hardware estarán designados para procesos de streaming, sin embargo, el equipo i5 utilizado pudo soportar ambos servicios sin presentar inconvenientes.
- Cuando la licencia de desarrollador de Wowza o de JW Player esté próxima a caducar, es posible renovarla para prologar el tiempo de uso, según se lo desee.
- Es recomendable organizar los contenidos en el servidor con un nombre que haga referencia al mismo, seguido de la calidad de video en la que fue codificada, esto

evitará el desorden por la gran cantidad de contenidos que se vayan acumulando, además de facilitar el trabajo para solventar inconvenientes de videos.

5.3 Líneas de Trabajos Futuros

Con el fin de aprovechar la investigación realizada y emplear la plataforma implementada, se propone lo siguiente para futuros trabajos de investigación:

- El códec de video utilizado en este proyecto es H.264, sin embargo, se puede generar una interesante comparativa en el desempeño y tráfico cursado por el servidor si los contenidos fueran codificados con el códec H.265 el cual es enfocado para contenidos Ultra HD y 4K lo cual generaría una mayor cantidad de datos para su compresión y transmisión.
- Los contenidos disponibles en esta plataforma no contienen subtítulos y no se encuentran encriptados, lo que puede ser cautivador el uso de contenidos subtítulos y debidamente encriptados, para finalmente analizar en tiempo real las solicitudes del reproductor sobre las secuencias correspondientes que contienen los subtítulos y poder descryptar la información.
- El escenario en el que fue implementada esta plataforma está limitada a una red LAN, lo que provoca una gran incertidumbre por ampliar el alcance de la plataforma para una red WAN y analizar el desempeño de esta, basándose en el procedimiento descrito en este documento.
- Es posible aprovechar el desarrollo de esta plataforma OTT y así generar una serie de manuales prácticos para laboratorio, donde los estudiantes puedan efectuar pruebas, analizar tramas, inundar con tráfico la red y emplear los distintos protocolos adaptativos disponibles.

Bibliografía

- AFN Noticias. (26 de Junio 2018). *Agencia Fronteriza de Noticias de Mexico, Regulación de los Servicios OTT*.
http://www.afntijuana.info/tecnologia/84312_regulacion_de_los_servicios_ott
- Beltrán, M., & Fernández, P. (2017). *Diseño e implementación de una plataforma de video streaming OTT multiformato*. [Tesis de Pregado, Universidad Politécnica de Madrid]
http://oa.upm.es/48406/1/PFC_MANUEL_BELTRAN_SEVILLA_PABLO_FERNANDEZ_ROS_ALES.pdf
- Cable Company. (2019). *Worldwide Broadband Speed League 2019*.
<https://www.cable.co.uk/broadband/speed/worldwide-speed-league/>
- El Heraldo. (2015). *Hacia una televisión a la medida de los millennials*. Bogotá.
<https://www.elheraldo.co/tendencias/hacia-una-television-la-medida-de-los-millennials-180484>
- FFMPEG. (2017). *FFMPEG World Wide Web*. <http://ffmpeg.org/documentation.html>
- Garro, A. (2014). *¿Qué es HTML5?*. <https://www.arkaitzgarro.com/html5/capitulo-1.html>
- Huidobro, J. M. (2017). *IPTV, la televisión a través de Internet*. 39-44.
https://www.acta.es/medios/articulos/ciencias_y_tecnologia/043039.pdf
- IDET. (2018). *Intituto del Derecho de las Telecomunicaciones de México*.
<https://www.idet.org.mx/noticias/la-evolucion-de-la-television-en-un-siglo-de-las-televisiones-mecanicas-a-las-televisiones-enrollables/>
- ITU-T-H.264, I. T. (Octubre 2010). *Advanced Video Coding for generic audiovisual services H.264*. https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201610-S!!PDF-E&type=%C3%ADtems
- ITU-T-H.265, I. T. (Abril 2015). *High Efficiency Video Coding*.
<https://www.itu.int/rec/dologin.asp?lang=e&id=T-REC-H.265-201504-S!!PDF-E&type=items>
- Lenovo. (2020). *IdeaPad U430 Touch Ultrabook*.
<https://www.lenovo.com/us/en/laptops/ideapad/u-series/u430-touch/>
- Marquines, O., & Añazco, O. (2014). *Entretenimiento y Streaming en la nube* [Tesis de Pregado, Escuela Superior Politécnica del Litoral ESPOL, Ecuador].
<https://www.dspace.espol.edu.ec/retrieve/102092/D-84406.pdf>
- Naranjo, J. G. (2016). *Análisis y diseño de una plataforma tecnológica para brindar servicios de video multipantalla OTT (Over The Top) en una empresa operadora de telecomunicaciones* [Tesis de Maestría, Pontificia Universidad Católica del Ecuador].
<http://repositorio.puce.edu.ec/bitstream/handle/22000/11305/INFORME%20CASO%20DE%20ESTUDIO%20-%20GABRIEL%20NARANJO%20%20OROZCO-v3.0.pdf?sequence=1&isAllowed=y>
- Netflix. (Febrero 2020). *Tabla de Posiciones de ISP en Ecuador*.
<https://ispspeedindex.netflix.com/country/ecuador/>

- Norfipc YouTube. (2020). *Calidad, resolución y tamaño de los videos de YouTube*. <https://norfipc.com/youtube/calidad-resolucion-tamano-videos-youtube.php>
- Ortiz, J. (2017). *JW Player para proyectos profesionales*. Herramientas de Video México. <https://javierortiz.mx/herramientasvideo/jwplayer-player-para-videos-o-reproductor-de-videos/>
- Piqueres, A. (2019). *Implementación de una solución de streaming bajo demanda usando DASH* [Tesis de Pregado, Universidad Politécnica de Valencia, España]. <https://pdfs.semanticscholar.org/f110/1845b48c607f9533428ad37dcd83ed7584a9.pdf>
- Sarmiento, L. (4 de Agosto 2014). *Mundo Contact*. Plataformas de Servicios OTT: la "cuarta ola" tecnológica. <https://mundocontact.com/plataformas-de-servicios-ott-la-cuarta-ola-tecnologica/>
- Savatier, T. (Agosto 2013). *Reference website for MPEG*. <https://mpeg.chiariglione.org/>
- Telefónica. (2015). *Destino Negocio*. <https://destinonegocio.com/ec/emprendimiento-ec/conoce-las-oportunidades-de-negocio-en-ott/>
- TP-Link Technologies Co., Ltd. (2020). https://www.tp-link.com/ec/home-networking/wifi-router/tl-wr940n/?utm_medium=select-local#specifications
- UIT-T. (1999). *Métodos de evaluación subjetiva de la calidad de video para aplicaciones multimedios*. https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-P.910-199909-S!!PDF-S&type=items
- Valdez, F. (2016). *Video Streaming de Alta Calidad sobre Plataformas de Distribución de Contenidos* [Tesis de Maestría, Universidad Politécnica de Madrid, España]. http://www.dit.upm.es/~posgrado/doc/TFM/TFMs2015-2016/TFM_Felipe_David_Valdez_Montero_2016.pdf
- Verizon Digital Media. (Octubre 2015). *HDS, HLS, HSS Adaptive HTTP Streaming*. <https://www.verizondigitalmedia.com/blog/2013/07/hds-hls-hss-adaptive-http-streaming/>
- Vermulm, S. (2015). OTT/ABR Demystified and Monitored.
- VideoLAN Organization. (2020). *VLC Media Player*. <https://www.videolan.org/vlc/index.es.html>
- Wowza. (2020). *Wowza Streaming Engine*. Software para transmisión de medios. <https://www.wowza.com/es/products/streaming-engine>
- Wowza Media Systems. (2020). *Wowza Streaming Engine Configuration Reference*. https://www.wowza.com/resources/WowzaStreamingEngine_ConfigurationReference.pdf
- Xataka. (2019). *Xataka Basics Webedia*. <https://www.xataka.com/basics/iptv-que-ventajas-desventajas-que-listas-canales>
- Xataka Streaming. (10 de Febrero 2018). *La compleja infraestructura detrás de Netflix: ¿qué pasa cuando le das al "play"?*. <https://www.xataka.com/streaming/la-compleja-infraestructura-detras-de-netflix-que-pasa-cuando-le-das-al-play>