

**BANCO DE PRUEBAS PARA EL SOPORTE DEL SERVICIO DE VIDEO BAJO DEMANDA DE CONTENIDOS MULTIMEDIA ADAPTATIVOS, BASADOS EN EL ESTANDAR DASH (ISO/IEC 23009-1).**



**DUVERNEI ORTIZ TRUJILLO**

**Trabajo de grado en Ingeniería Electrónica y Telecomunicaciones**

**Director:**

**Mag. Gabriel Elías Chanchí Golondrino**

**Co-director:**

**PhD. José Luis Arciniegas Herrera**

**Universidad del Cauca  
Facultad de ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Línea de Investigación en Servicios Avanzados de Telecomunicaciones  
Popayán, diciembre de 2015**

**DUVERNEI ORTIZ TRUJILLO**

**BANCO DE PRUEBAS PARA EL SOPORTE DEL  
SERVICIO DE VIDEO BAJO DEMANDA DE  
CONTENIDOS MULTIMEDIA ADAPTATIVOS,  
BASADOS EN EL ESTANDAR DASH  
(ISO/IEC 23009-1).**

**Tesis presentada en la Facultad de Ingeniería  
Electrónica y Telecomunicaciones de la  
Universidad del Cauca para la obtención del  
Título de:**

**Ingeniero en  
Electrónica y Telecomunicaciones**

**Director:  
Mag. Gabriel Elías Chanchí Golondrino**

**Co-director:  
PhD. José Luis Arciniegas Herrera**

**Popayán  
2015**

# **Agradecimientos**

A Dios por sus bendiciones recibidas,  
A mi familia, por su apoyo incondicional.

## Tabla de contenido

Lista de Figuras .....	iii
Lista de Tablas .....	iv
Capítulo 1 .....	1
Introducción .....	1
1.1    Introducción .....	1
1.2    Marco teórico .....	2
Streaming .....	2
IPTV .....	2
Video bajo demanda .....	4
DASH .....	5
1.3    Planteamiento del problema .....	6
1.4    Justificación .....	11
1.5    Estado actual del conocimiento .....	12
1.6    Aportes .....	19
1.7    Objetivos .....	21
1.7.1    Objetivo general .....	21
1.7.2    Objetivos específicos .....	21
1.8    Actividades .....	21
1.9    Estructura del trabajo de grado .....	24
Capítulo 2 .....	25
Banco de pruebas .....	25
2.1    Introducción .....	25
2.2    Requisitos .....	26
Requisitos funcionales .....	26
Requisitos no funcionales .....	28
2.3    Marco teórico .....	28
Video .....	29
Codec .....	29
WebM .....	29
MP4Box .....	29
Osmo4 .....	30
VLC Player .....	30
Red5 .....	30
Darwin .....	31
Live555 .....	31
Apache .....	31
Apache Bench (ab) .....	31
FFMPEG .....	32
Libwebm .....	32
webm-tools .....	32

webm-dash-javascript .....	33
JavaScript .....	33
Python .....	33
Selección de herramientas .....	34
<b>2.4 Metodología de trabajo .....</b>	<b>36</b>
<b>2.5 Planteamiento del banco de pruebas .....</b>	<b>38</b>
2.5.1 Vista de negocio .....	39
2.5.2 Vista de servicios .....	39
2.5.3 Vista de flujo de trabajo .....	41
2.5.4 Vista funcional general .....	43
<b>Capítulo 3 .....</b>	<b>47</b>
<b>Herramienta de codificación automática .....</b>	<b>47</b>
3.1 Introducción .....	47
3.2 Herramientas seleccionadas para la generación de contenido multimedia DASH .....	48
FFMPEG .....	48
Sample_muxer .....	48
Webm_dash_manifest .....	49
3.3 Proceso de codificación de streaming adaptativo .....	50
3.4 Diagrama modular de la herramienta DASH WebM Converter .....	53
3.5 Archivo descriptor MPD .....	55
<b>Capítulo 4 .....</b>	<b>59</b>
<b>Servicio de video bajo demanda .....</b>	<b>59</b>
4.1 Introducción .....	59
4.2 Funcionamiento de los componentes interactivo y administrativo .....	60
4.3 Funcionamiento del servicio de VoD .....	68
4.4 Herramienta de codificación automática online .....	72
4.5 Disposición de los servicios .....	74
<b>Capítulo 5 .....</b>	<b>77</b>
<b>Pruebas .....</b>	<b>77</b>
5.1 Introducción .....	77
5.2 Prueba de consumo de memoria .....	78
5.3 Prueba de ancho de banda .....	82
5.4 Prueba de estrés .....	86
<b>Capítulo 6 .....</b>	<b>91</b>
<b>Conclusiones .....</b>	<b>91</b>
6.1 Introducción .....	91
6.2 Conclusiones .....	91
6.3 Trabajos futuros .....	94
<b>Referencias .....</b>	<b>97</b>

## Lista de Figuras

<b>Figura 1.1.</b> Cadena de IPTV.....	3
<b>Figura 1.2.</b> Ejemplo de servicio de video bajo demanda, Youtube.....	4
<b>Figura 1.3.</b> Escenario de streaming adaptativo DASH.....	5
<b>Figura 1.4.</b> Estructura de archivo descriptor MPD.....	6
<b>Figura 2.1.</b> Metodología de trabajo.....	37
<b>Figura 2.2.</b> Vistas para el desarrollo del banco de pruebas.....	38
<b>Figura 2.3.</b> Modelo de negocio.....	39
<b>Figura 2.4.</b> Esquema de consumo de servicios interactivos.....	41
<b>Figura 2.5.</b> Flujo de trabajo.....	42
<b>Figura 2.6.</b> Diagrama modular de funcionamiento general del banco de pruebas.....	43
<b>Figura 2.7.</b> Esquema general de banco de pruebas.....	44
<b>Figura 3.1.</b> Scripts utilizados en la herramienta de codificación automática.....	49
<b>Figura 3.2.</b> Proceso de codificación de streaming adaptativo.....	50
<b>Figura 3.3.</b> Interfaz Gráfica de la herramienta Dash WebM Converter.....	51
<b>Figura 3.4.</b> Fase de codificación de los contenidos WebM.....	51
<b>Figura 3.5.</b> Recursos de la herramienta de codificación automática.....	53
<b>Figura 3.6.</b> Diagrama de bloques herramienta de codificación automática.....	54
<b>Figura 3.7.</b> Documento de descripción MPD (cabecera).....	56
<b>Figura 3.8.</b> Documento de descripción MPD (pie).....	57
<b>Figura 4.1.</b> Casos de uso para el rol de usuario.....	61
<b>Figura 4.2.</b> Diagrama de secuencia del servicio de VoD.....	63
<b>Figura 4.3.</b> Diagrama de secuencia del sistema de valoración de contenidos.....	64
<b>Figura 4.4.</b> Casos de uso para el rol de administrador.....	65
<b>Figura 4.5.</b> Diagrama de secuencia del sistema de codificación automática online.....	67
<b>Figura 4.6.</b> Interfaz gráfica del servicio de VoD.....	68
<b>Figura 4.7.</b> Catálogo de contenidos multimedia para el servicio de VoD.....	69
<b>Figura 4.8.</b> Componente de reproducción del servicio de VoD.....	69
<b>Figura 4.9.</b> Sistema de valoración de contenidos multimedia.....	70
<b>Figura 4.10.</b> Servicio interactivo de chat.....	71
<b>Figura 4.11.</b> Autenticación para acceder al servicio de chat.....	71
<b>Figura 4.12.</b> Ventana inicial de la herramienta de codificación automática online.....	72
<b>Figura 4.13.</b> Formulario de la herramienta de codificación automática online.....	73
<b>Figura 4.14.</b> Multitarea de la herramienta de codificación automática online.....	73
<b>Figura 4.15.</b> Contenido multimedia publicado en el servicio de VoD.....	74
<b>Figura 4.16.</b> Organización de los servicios en el banco de pruebas.....	75
<b>Figura 5.1.</b> Entorno de prueba para evaluar Dash WebM Converter.....	78
<b>Figura 5.2.</b> Porcentaje de CPU vs tiempo (FFMPEG).....	79
<b>Figura 5.3.</b> Memoria RAM vs tiempo (FFMPEG).....	80
<b>Figura 5.4.</b> Porcentaje de CPU vs tiempo.....	81
<b>Figura 5.5.</b> RAM vs tiempo.....	81
<b>Figura 5.6.</b> Script para emulación de ancho de banda.....	83
<b>Figura 5.7.</b> Entorno de prueba para emulación de ancho de banda.....	84
<b>Figura 5.8.</b> Servicio de VoD en la nube.....	84
<b>Figura 5.9.</b> Emulación de ancho de banda.....	85
<b>Figura 5.10.</b> Peticiones en modo secuencial (30).....	87
<b>Figura 5.11.</b> Peticiones en modo secuencial (100).....	87
<b>Figura 5.12.</b> Peticiones en modo concurrente (50).....	88
<b>Figura 5.13.</b> Peticiones en modo concurrente (500).....	89
<b>Figura 5.14.</b> Peticiones en modo secuencial (total).....	90
<b>Figura 5.15.</b> Peticiones en modo concurrente (total).....	90

## Lista de Tablas

<b>Tabla 1.1.</b> Actividades para el desarrollo del trabajo de grado.....	23
<b>Tabla 2.1.</b> Herramientas seleccionadas para conformar el banco de pruebas. ....	35
<b>Tabla 4.1.</b> Descripción del caso de uso seleccionar película. ....	61
<b>Tabla 4.2.</b> Descripción del caso de uso reproducir película. ....	62
<b>Tabla 4.3.</b> Descripción del caso de uso valorar contenido multimedia. ....	62
<b>Tabla 4.4.</b> Descripción del caso de uso acceder al servicio de chat.....	62
<b>Tabla 4.5.</b> Descripción del caso de uso subir contenido WebM. ....	65
<b>Tabla 4.6.</b> Descripción del caso de uso ingresar datos de codificación.....	66
<b>Tabla 4.7.</b> Descripción del caso de uso codificar contenido.....	66
<b>Tabla 4.8.</b> Descripción del caso de uso publicar contenido.....	66

# Capítulo 1

## Introducción

### 1.1 Introducción

En los últimos años, Internet se ha convertido en un importante canal para la transmisión de contenidos multimedia que utilizan HTTP como protocolo principal. Según [1], en Norte América el *streaming* de contenidos de entretenimiento representa más del 68% del tráfico en redes de acceso fijo, en donde solo Netflix constituye el 31.6%. De igual manera en Europa el *streaming* de entretenimiento en tiempo real para redes de acceso fijo supera el 47.4%, como resultado del incremento en la disponibilidad de servicios de video Over The Top (OTT<sup>1</sup>).

Por su parte, el *streaming* de contenidos de entretenimiento en América latina representa un 50% en cuanto al tráfico de acceso fijo y el 29% del tráfico de acceso móvil. Considerando lo anterior, y que el 35% del tráfico total de la red de internet en América Latina utiliza HTTP en la capa de aplicación para el sistema de *streaming* multimedia, de los cuales el 8% del tráfico es para upstream, el 20% del tráfico para downstream y el 18,1% corresponde a navegación en general [2], puede establecerse HTTP como el principal protocolo en las redes modernas [3].

Por otro lado, es estimado que los servicios de televisión paga y video bajo demanda (VoD) aumentarán su presencia para los próximos años. Según [4] el mercado OTT VoD en América latina para el 2013 ha representado 3,63 millones de afiliaciones;

---

<sup>1</sup> Over-the-top (OTT) es el término general para los servicios que se despliegan y se consumen sobre Internet, en la cual no está implicado un operador de red en el control o distribución del contenido. Estos servicios se montan en la capa superior de un servicio que el usuario ya tiene, por lo tanto no requiere ninguna afiliación de negocio o de tecnología con el operador de red. [31]



cifra equivalente al 6,9% de los hogares con televisión paga y considera que para el 2018 incrementará a 25,63 millones de cuentas, es decir, a un 28,4%.

Considerando las anteriores cifras y la gran acogida del servicio de video bajo demanda en entornos de IPTV, es necesario conocer un poco las problemáticas que de una u otra manera intervienen en la calidad de presentación del servicio, debido a las diferentes características inherentes de las redes de datos y la capacidad de los nuevos dispositivos de acceso a este tipo de servicios.

En este capítulo son descritos los conceptos más relevantes y las problemáticas tenidas en cuenta para la presente investigación, así como también es planteada la propuesta para el desarrollo del presente trabajo. La estructura de este primer capítulo está organizado de la siguiente manera: primero es presentado un marco teórico para definir los conceptos principales en este trabajo, a continuación es presentado el planteamiento del problema que motiva esta investigación y su justificación, después el estado actual del conocimiento junto con los aportes que van de acuerdo al estado presentado, finalmente son planteados los objetivos del proyecto, las actividades pertinentes a este trabajo y la estructura de cada uno de los capítulos de la monografía.

## 1.2 Marco teórico

Esta sección presenta los conceptos más relevantes para el desarrollo del presente trabajo:

### ***Streaming***

Es una técnica para la distribución de contenido multimedia en internet [7], en la cual no es necesario que el cliente descargue completamente ésta información para consumirla, ya que es almacenada en un buffer y ejecutada al mismo tiempo que es transmitida por la red.

### **IPTV**

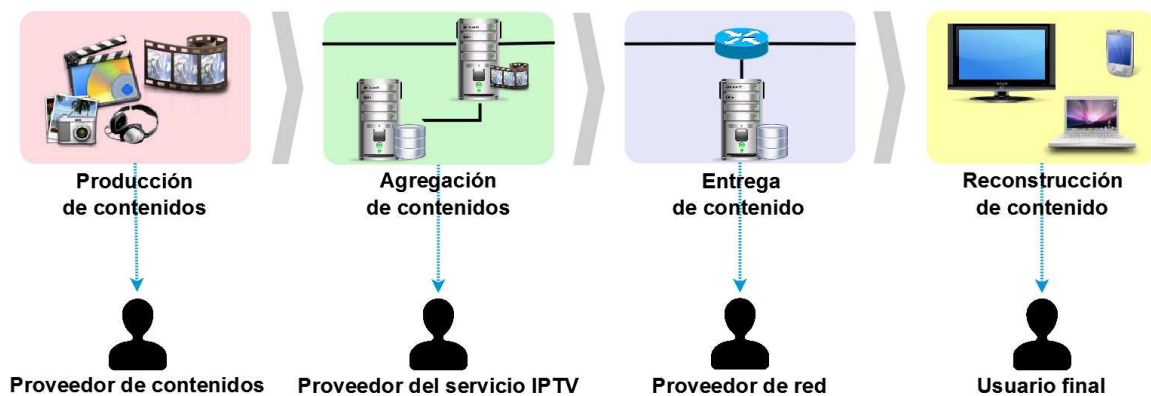
Según la ITU-T (2008) IPTV consiste en el envío de servicios multimedia (televisión, video, audio, texto, gráficos y datos) de alta calidad a través de redes IP de banda ancha, mediante el uso de protocolos y tecnologías de *streaming*. Los servicios

provistos por IPTV, desde la perspectiva del usuario final, son de comunicación (servicios de mensajería, telefonía y video llamadas), interactividad (servicios de entretenimiento, noticias, búsqueda y recomendación) y distribución de contenidos (servicios de pago por ver, cine, publicidad y video bajo demanda).

IPTV es un sistema cerrado con una infraestructura completamente dentro del entorno del proveedor de servicios multimedia, por lo tanto, el proveedor posee control absoluto sobre dichos contenidos. Gracias a que IPTV está basado en el video *streaming*, es posible entre otras cosas:

- Ofrecer servicio de video bajo demanda.
- Propiciar el desarrollo de nuevos estándares de codificación, compresión y producción de contenidos multimedia.
- Tenencia de productos, servicios, audiencia y publicidad personalizada.
- Adquisición de control y mejoramiento de la calidad de los servicios.

La figura 1.1, muestra la cadena de valor de IPTV, estructurada de acuerdo a los organismos de estandarización UIT, TISPAN y OIPF:



**Figura 1.1.** Cadena de IPTV.

La cadena de valor de IPTV parte desde el proveedor de contenidos, quien es el encargado de la producción y edición de contenidos multimedia con algún tipo de licencia para vender y/o proporcionar dichos contenidos al proveedor de servicios de IPTV. A partir de aquí es agregado y adaptado el contenido de acuerdo a los servicios que serán enviados al usuario final a través del proveedor de red, el cual permite la comunicación constante entre el proveedor del servicio IPTV y el usuario final.

## Video bajo demanda

Está definido como la entrega de contenido de vídeo a través de IP (protocolo internet) de banda ancha para el espectador individual iniciada por él y en el momento de su elección [5], es decir, el usuario tiene el control de la sesión de video. En este propósito, una buena capacidad de almacenamiento, gran ancho de banda, servicio en tiempo real (reproducir, parar, avanzar) y calidad de servicio, son requisitos esenciales para su implementación, por lo cual la técnica de *streaming* es apropiada, por su sencillez y flexibilidad [6].

La figura 1.2 muestra el servicio Youtube, un ejemplo de video bajo demanda muy conocido en internet.



**Figura 1.2.** Ejemplo de servicio de video bajo demanda, Youtube.

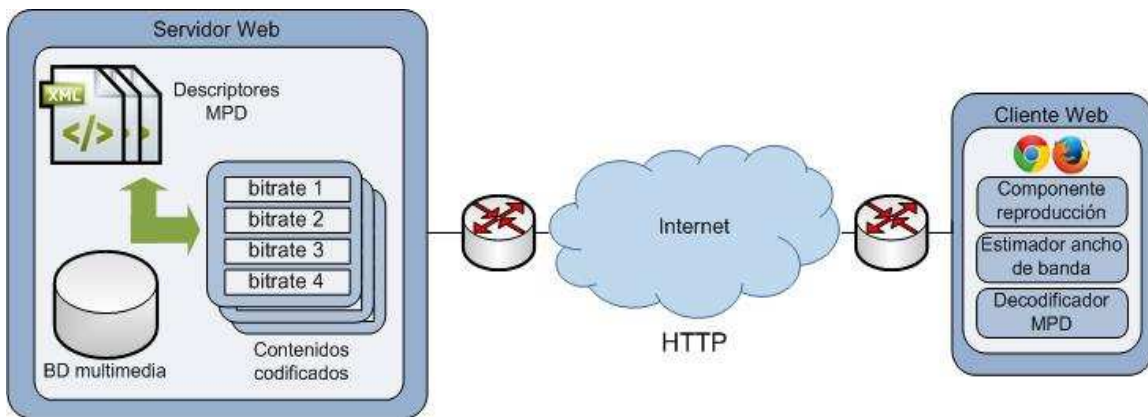
Los elementos más representativos de este servicio son nombrados a continuación:

1. Componente de reproducción.
2. Contador de reproducciones.
3. Valoración del contenido.
4. Descripción del contenido.
5. Recomendaciones y/o relacionados.
6. Buscador de contenidos.

## DASH

DASH (Dynamic Adaptive *Streaming* over HTTP) es un estándar especificado por el grupo MPEG (Moving Picture Expert Group), para el *streaming* de video en internet a través de cualquier servidor web HTTP [16]. Su objetivo principal es mejorar la experiencia audiovisual del usuario final, a través de la adaptación del contenido multimedia respecto a las condiciones de red cambiantes, como el ancho de banda, inherente a las redes de datos.

La figura 1.3 muestra la estructura básica para el funcionamiento del estándar DASH:



**Figura 1.3.** Escenario de *streaming* adaptativo DASH.

El escenario de *streaming* adaptativo DASH consiste en la disposición de varios flujos de video codificados a diferentes resoluciones y tasas de bit, en un servidor web para ser enviados al cliente vía HTTP. Cada agrupación de flujos de video, que conforman un contenido multimedia diferente, están descritos en un archivo de manifiesto en formato MPD, propio del estándar DASH.

La figura 1.4 muestra la estructura del archivo descriptor MPD especificado por el estándar, está escrito en lenguaje XML.

```

 9  <profiles="urn:webm:dash:profile:webm-on-demand:2012">
10  <Period id="0" start="PT0S" duration="PT264.16S" >
11    <AdaptationSet id="0" mimeType="video/webm" codecs="vp8" width="320" height="240"
12      subsegmentAlignment="true" subsegmentStartsWithSAP="1" bitstreamSwitching="true">
13      <Representation id="0" bandwidth="74285">
14        <BaseURL>sports_50k_sm.webm</BaseURL>
15        <SegmentBase indexRange="1925436-1926323">
16          <Initialization range="0-248" />
17        </SegmentBase>
18      </Representation>
19      <Representation id="1" bandwidth="119023">
20        <BaseURL>sports_100k_sm.webm</BaseURL>
21        <SegmentBase indexRange="3435010-3435898">
22          <Initialization range="0-248" />
23        </SegmentBase>
24      </Representation>

```

**Figura 1.4.** Estructura de archivo descriptor MPD.

El cliente web es el encargado del despliegue y control del contenido multimedia DASH, según las condiciones actuales de red y de acuerdo a los contenidos descritos en el archivo MPD, recibido al inicio de la transmisión.

### 1.3 Planteamiento del problema

Tradicionalmente el *Streaming* en internet es llevado a cabo por medio de protocolos como *Real-time Transport Protocol* (RTP) y *Real Time Streaming Protocol* (RTSP). El protocolo RTP trabaja sobre UDP, por lo cual no garantiza que todos los paquetes lleguen a su destino, es por esto que el servidor debe gestionar una sesión diferente para cada cliente, además de coordinar la entrega de paquetes por medio de otros protocolos como son: RTSP y RTCP (*Real Time Control Protocol*) [8], por lo tanto incrementa la información que debe transmitirse por la red. Por su parte, RTSP puede utilizar tanto UDP como TCP. En caso de usarse TCP, el servidor necesita mantener el estado de la conexión, cuestión que incrementa la información de control entre cliente y servidor.

Una solución a los anteriores inconvenientes ha sido la descarga progresiva. A diferencia de RTP, ésta hace uso del protocolo de transferencia de hipertexto (HTTP). En consecuencia, la sesión no tiene estado y el cliente extrae los datos desde el servidor. Esto tiene una gran ventaja y es la explotación plena de la infraestructura existente de Internet. Sin embargo, este enfoque tiene también

algunas desventajas. El protocolo HTTP añade una sobrecarga significativa en la transmisión, que es aproximadamente el doble de la tasa de bits de los medios de comunicación [9]. Por lo tanto, la descarga básica progresiva sobre HTTP no es adecuada para los entornos móviles actualmente en expansión y sus considerables fluctuaciones en el ancho de banda. Así entonces, surge una nueva necesidad; que el flujo de vídeo sea adaptado a las capacidades de ancho de banda con el fin de ofrecer al usuario una secuencia de vídeo continua, de la mejor calidad posible.

Una de las principales soluciones para superar el anterior inconveniente fue especificada en 3GPP como *Adaptive HTTP Streaming* (AHS) [10]. Esta técnica consiste en cortar el archivo multimedia en segmentos de igual duración que pueden ser codificados en diferentes resoluciones y tasas de bits, es decir es codificada a una variedad de velocidades de bits, los cuales son suministrados a un servidor web para ser descargados a través de peticiones HTTP GET estándar. Con el fin de establecer la relación entre tasas de bits, segmentos y el orden de los mismos, AHS hace uso del archivo *Media Presentation Description* (MPD), éste contiene una descripción formal sobre una colección de datos (resolución, rango de bits, dirección HTTP-URL, tiempo de duración de cada segmento, entre otros) el cual constituye una presentación limitada o ilimitada del contenido multimedia [11].

Cada cliente primero solicita el MPD y con base en esa información indica a los segmentos individuales mejor ajustados a sus necesidades. Así entonces, el control está del lado del cliente, de tal manera que cada segmento puede cambiar a una velocidad de bits en función del ancho de banda que disponga. Lo anterior trae varias ventajas debido a que mejora la calidad de experiencia del usuario.

La industria ha desplegado varias soluciones propietarias basadas en *Streaming* adaptativo, entre las más representativas están: *Microsoft Smooth Streaming* [12], *Apple HTTP Live Streaming* [13] y *Adobe Dynamic Streaming* HTTP [14]. Estas plataformas de *streaming* propietarios suelen utilizar el estándar de video digital MPEG-4 H.264, junto con MPEG *Advanced Audio Coding* (AAC). H.264 permite lograr la mayor calidad de imagen de video para una frecuencia de bits determinada, mientras que AAC está encargado de la distribución eficiente de sonido a través de conexiones de ancho de banda moderada. Sin embargo, el inconveniente radica en que cada uno emplea sus propias técnicas de segmentación, secuencia de tiempo y formatos de MPD. Por lo tanto, si los dispositivos de consumo son usados para

recibir y mostrar el contenido de un servidor de aplicaciones en cualquiera de estas plataformas, deben apoyar el protocolo de cliente correspondiente [15].

Con el fin de lograr la entrega eficiente de contenidos multimedia utilizando HTTP en sus diferentes formas: adaptativa, progresiva, *streaming*/descarga y además de garantizar la *interoperabilidad* entre las soluciones propietarias, MPEG (Moving Picture Expert Group) desarrolla Dynamic Adaptive *Streaming* over HTTP (DASH). Es un estándar internacional a partir de noviembre de 2011, siendo publicado como ISO/IEC 23009-1:2012 [11] en abril de 2012. De acuerdo con [16], DASH puede definirse como un sistema por el cual son proporcionados formatos que habilitan la entrega eficiente y de alta calidad de servicios de *streaming*, destacado por presentar las siguientes ventajas [17]:

- Reutiliza la tecnología existente de contenedores, códec, DRM<sup>2</sup>, etc.
- Puede ser desplegado sobre la infraestructura actual de CDN-HTTPS<sup>3</sup>.
- El usuario percibe un mejor servicio: menor tiempo de arranque debido a que no es necesario que el archivo multimedia sea almacenado temporalmente a la espera de su reproducción.
- Permite seleccionar la calidad en función de la red y capacidad del dispositivo.
- Los cambios entre calidades son automáticos y transparentes al usuario.
- Permite contenidos en directo con menús y características tipo DVD.
- Puede coexistir con tecnologías propietarias existentes.

El establecimiento de la *interoperabilidad* entre toda la gama de servidores y clientes simplifica la preparación y entrega de contenido multimedia a una gran variedad de dispositivos compatibles con el estándar. Así entonces, los proveedores de

---

<sup>2</sup> Digital Rights Management (DRM), gestión digital de derechos, hace referencia a una colección de técnicas usadas por editoriales y titulares de derechos de autor para limitar el uso de medios o dispositivos digitales. [32]

<sup>3</sup> Content Delivery Network (CDN), una red de distribución de contenidos representa un grupo de servidores dispersos geográficamente, desplegados para facilitar la distribución oportuna y eficiente de información generada por los editores web.

contenido pueden producir un único conjunto de archivos para el consumo desde cualquier dispositivo compatible. Por consiguiente, el estándar DASH acelera en gran medida el crecimiento de nuevos servicios. Sin embargo, el estándar no está definido para la implementación de servicios de video bajo demanda, en consecuencia, deberían abordarse aspectos relevantes en el desarrollo de los mismos como: los procesos de segmentación y codificación del contenido multimedia, implementación de un esquema liviano para el consumo de aplicaciones basadas en el estándar, la adecuación del cliente web correspondiente y el despliegue de servicios interactivos vinculados al servicio de VoD de contenidos adaptativos.

En este sentido, surgen múltiples necesidades a la hora de incursionar de una manera eficiente en el desarrollo de nuevos servicios y aplicaciones interactivas bajo el escenario en cuestión, entre ellos son destacados los siguientes:

- Los servicios basados en DASH requieren de procesos previos que tienden a ser repetitivos y tediosos como son: la codificación, la segmentación y la generación del archivo descriptor MPD. De tal manera que es necesario automatizar dichos procesos.
- La implementación de aplicaciones interactivas vinculadas al servicio de VoD de contenidos multimedia adaptativos, teniendo en cuenta que el descriptor MPD solo define las características del contenido multimedia.
- Es necesario un esquema liviano que permita el intercambio de datos entre las aplicaciones interactivas vinculadas al servicio de video bajo demanda, facilitando el acceso a las aplicaciones y a los recursos de red sin descuidar la velocidad de procesamiento en los dispositivos que trabajan sobre el estándar.
- Adecuación de un cliente web, el cual facilite el despliegue del servicio de video bajo demanda de contenidos multimedia DASH-WebM y las aplicaciones interactivas vinculadas al mismo (Ej. foros, chats, sistemas de votos u otros servicios de participación en línea).



De acuerdo a los anteriores aspectos es importante resaltar que el principal reto del *streaming* adaptativo DASH, es facilitar una amplia *interoperabilidad* entre servidores y dispositivos desplegados sobre la actual infraestructura HTTP, de tal manera que los proveedores de contenido puedan adaptar sus servicios a las crecientes expectativas de los usuarios. El estándar DASH no define aspectos importantes para la implementación de VoD y posibles servicios asociados, por lo cual es necesaria la búsqueda de alternativas que permitan cumplir el reto anteriormente citado.

A partir de los anteriores retos y situaciones problemáticas, surge la pregunta de investigación del presente trabajo de grado: ¿Cómo permitir el soporte del servicio de video bajo demanda, de contenidos multimedia adaptativos, y sus aplicaciones asociadas, considerando el estándar DASH (ISO/IEC 23009-1)? El presente trabajo pretende responder a la anterior pregunta por medio del planteamiento de un banco de pruebas para el soporte del servicio de video bajo demanda de contenidos multimedia adaptativos, basados en el estándar DASH (ISO/IEC 23009-1).

De acuerdo a [18], [19] y [20], un banco de pruebas es un conjunto de herramientas software, que proporcionan un entorno uniforme para explorar, probar, comparar y reproducir diversas tecnologías, permitiendo así, una amplia gama de actividades de verificación y desarrollo. El banco de pruebas a plantear incluirá: la implementación de aplicaciones interactivas vinculadas al servicio de VoD de contenidos multimedia adaptativos, la automatización de procesos para la generación de esos contenidos, un esquema liviano para el intercambio de datos entre las aplicaciones anteriormente nombrados, así como la adecuación de un cliente web para el despliegue del contenido multimedia y sus aplicaciones asociadas. Este banco de pruebas pretende servir de base para el desarrollo de futuros trabajos dentro del laboratorio de TDi de la Universidad del Cauca, relacionados con la implementación de servicios soportados en *streaming* adaptativo y aplicaciones avanzadas (ej. sistemas de recomendaciones, guías inteligentes de contenido) asociadas al servicio de VoD.

## 1.4 Justificación

El presente trabajo de grado permitirá ampliar el campo de estudio de los proyectos desarrollados dentro del “Laboratorio Experimental de Televisión Digital Interactiva de la Universidad del Cauca”. El aporte fundamental, dado que el estándar DASH solamente está definido para el soporte básico de *streaming* de contenido multimedia adaptativo sobre HTTP, radica en la generación de un banco de pruebas, para el soporte del servicio de VoD, de contenidos multimedia adaptativos basados en el estándar DASH, en conjunto con el despliegue de aplicaciones interactivas vinculadas a dicho servicio.

Teniendo en cuenta lo anterior, resulta pertinente citar los proyectos desarrollados a nivel institucional, los cuales abarcan temáticas de T-Learning (Aprendizaje interactivo a través de la televisión) y generación de contenidos interactivos para televisión digital (TDi). Así entonces, en [21] es presentado el proyecto denominado EDiTV (Educación Virtual Basada en Televisión Interactiva para Apoyar Procesos Educativos a Distancia), cuyo objetivo principal fue apoyar procesos educativos a distancia ofrecidos por la Universidad del Cauca, haciendo uso de la red RENATA. Posteriormente el proyecto denominado ST-CAV (Servicios de T-Learning para el soporte de Comunidades Académicas Virtuales) brindaba soporte a comunidades académicas virtuales (CAV) desde varios escenarios de manera simultánea: IPTV, Televisión Digital Terrestre y TV Móvil [22].

De acuerdo a [23], entre los servicios avanzados ofrecidos para televisión interactiva está el VoD, el cual permite que el usuario sea transformado en el verdadero actor de la comunicación, dotándolo con la capacidad de consumo atemporal y la libre elección de contenidos. Sin embargo, para lograr estas características es necesario satisfacer ciertos aspectos, como por ejemplo; consumo adecuado de ancho de banda, formatos livianos para el contenido multimedia, clientes y servidores apropiados, entre otros, sobre todo en el entorno de las CAV en donde es generada información de manera activa. Así entonces, soportar el servicio de VoD, de contenidos multimedia adaptativos basados en el estándar DASH representa una solución a los anteriores requerimientos.

Así mismo, este trabajo busca contribuir al mejoramiento de la experiencia a nivel de usuario, gracias a la ventaja que ofrecen los contenidos multimedia adaptativos, relacionada con la descarga progresiva de estos, minimizando problemas de

reproducción ocasionados por características limitantes de la red como la variación del ancho de banda. Finalmente este trabajo constituye un punto de partida importante para la implementación futura de servicios avanzados, ampliando el campo de acción de los mismos en las nuevas tecnologías que están siendo desarrolladas, como son los servicios interactivos de televisión en escenarios de TV Móvil e IPTV Móvil.

## 1.5 Estado actual del conocimiento

En ésta sección son presentados los trabajos de investigación tenidos en cuenta para el planteamiento del presente proyecto y en la sección de aportes están indicadas las contribuciones que proporcionará este trabajo de grado.

***Light Weight Content Fingerprinting for Video Playback Verification in MPEG DASH*** [24]. Este artículo describe el principal modelo de negocio de los actuales proveedores de contenidos multimedia DASH, es decir los anuncios de video, los cuales son, ahora, su principal fuente de ingresos, gracias a la gran demanda de contenidos multimedia en internet. Estos anuncios son de alrededor de 15 segundos de duración y suelen aparecer al inicio, al final o en el intermedio, durante el consumo de videos en línea (similar a los clásicos anuncios en televisión). El proveedor de servicios ofrece al cliente una variedad de los mismos para elegir y combinar, mejorando de esta manera la experiencia de entrega, dichos servicios están descritos en el archivo MPD, el cual es un documento escrito en formato XML, que es proporcionado a los clientes DASH al iniciar la transmisión. Este trabajo propone un nuevo enfoque para la verificación de la reproducción de anuncios en contenidos DASH, basado en la creación de dependencias entre contenidos, y muestra los resultados de simulación para analizar la exactitud del uso de huellas digitales de contenido, como una herramienta para exigir y comprobar la reproducción de los anuncios en línea, con características de representación muy liviana dentro del archivo MPD.

De acuerdo a lo anterior es importante destacar, que este artículo proporciona bases esenciales para conocer las diversas capacidades y la flexibilidad que ofrece el estándar DASH, para vincular servicios adicionales dentro del archivo MPD, sin embargo este artículo solo describe la forma de entrelazar y generar dependencia

entre contenidos multimedia DASH, para su despliegue y presentación a un usuario final. Un aporte de este trabajo de grado es proporcionar un servicio de video bajo demanda, de contenidos multimedia DASH-WebM; descrito dentro del archivo MPD especificado por el estándar, teniendo en cuenta características de representación liviana y un esquema adecuado para el consumo de aplicaciones interactivas, vinculadas al servicio.

***Open Source Column: Dynamic Adaptive Streaming over HTTP Toolset*** [25].

Este artículo menciona los actuales y principales proveedores de contenidos multimedia adaptativo, Netflix y YouTube, que no despliegan su propia arquitectura de *streaming*, pero ofrecen sus servicios OTT (*Over-The-Top*), basándose en el uso de la infraestructura ya desplegada para HTTP. Además este escrito recoge un conjunto de herramientas *open source*, que han sido desarrolladas para propósitos de evaluación, con el fin de aportar al despliegue e implementación del estándar DASH.

Dentro de estas herramientas están en el lado del cliente Libdash, que es una biblioteca para el control de *streaming* DASH y el acceso a los segmentos multimedia descargables; otra herramienta en el lado del cliente es el *plugin dash* de VLC *Media Player*, necesario para la recepción de contenido multimedia en formato DASH (disponible también en *Android*); por último un cliente basado en *JavaScript* (DASH-JS), que integra perfectamente DASH en la web, haciendo uso de las extensiones para manejo de medios en HTML5 propias del navegador *Google Chrome*, DASH-JS también utiliza segmentos de medios basados en WebM y segmentos basados en el formato de archivo ISO *Base Media*. Con respecto a las herramientas del lado del servidor este artículo presenta una base de datos DASH, que contiene diferentes secuencias de películas, disponibles en diferentes longitudes de segmentos, diferentes tasas de bit y resoluciones; además de un conjunto de contenidos codificados usando *DASHEncoder*, la cual es una herramienta de generación de contenido DASH, recomendada especialmente para el servicio de video bajo demanda (el usuario no necesita codificar y multiplexar por separado cada nivel de calidad del contenido DASH final) y usada para codificación por x264 y Mp4box de GPAC.

De acuerdo a lo anterior, este artículo describe un conjunto de herramientas tanto del lado del servidor como del lado del cliente, para la codificación y recepción de

contenidos multimedia adaptativos, soportados en el estándar DASH; sin embargo las herramientas presentadas, solo están consideradas para dar soporte al servicio básico de video *streaming*, más no para el servicio de video bajo demanda. Por lo anterior, este trabajo de grado propone integrar dichas herramientas en un banco de pruebas, haciendo uso de características particulares de las mismas, para soportar el servicio de VoD de contenido multimedia DASH-WebM y además la adecuación de un cliente web, para permitir el despliegue de aplicaciones interactivas y contenidos multimedia *DASH-WebM*, vinculados al servicio de video bajo demanda.

### ***A VLC Media Player Plugin enabling Dynamic Adaptive Streaming over HTTP***

[9]. Este trabajo describe la implementación de un *plugin* basado en el estándar DASH para el reproductor de contenidos multimedia VLC. El *plugin* proporciona una estructura flexible que puede extenderse fácilmente con respecto a diferentes lógicas de adaptación o perfiles del estándar DASH. En general, define una arquitectura basada en cuatro componentes esenciales; el primer módulo es el XML, éste es responsable del inicio de una sesión DASH, y del análisis XML del archivo MPD para brindar apoyo a los demás componentes, en esta etapa no es interpretada la información ni son tomadas decisiones en cuanto a la adaptación lógica. El segundo componente es el HTTP, encargado de gestionar todas las conexiones HTTP que deben ser abiertas o cerradas durante una sesión de DASH. En el *plugin* actual es abierta una conexión HTTP para cada segmento individual, cuestión que plantea como trabajo futuro utilizar conexiones HTTP persistentes para reducir la sobrecarga de HTTP. El tercer componente es el XML-Logic, el cual proporciona la lógica para la presentación de datos proporcionados por el primer módulo. Por último está el módulo Adaptation-Logic, responsable de la adaptación a las preferencias del usuario o a las capacidades del dispositivo, como son; el ancho de banda y la resolución. Éste componente permite al *plugin* integrar una variedad de lógicas de adaptación, basándose en el patrón de estrategia *well-known*, lo cual favorece el cambio de la lógica sin afectar a los otros componentes.

Este trabajo hace uso del archivo MPD para describir el contenido multimedia a transmitir, de tal manera que no es explorada la posibilidad de adaptarlo para ofrecer aplicaciones interactivas vinculadas al servicio de VoD de contenidos multimedia adaptativos, sin embargo, el artículo aporta bases importantes para el diseño e implementación del banco de pruebas planteado, ya que expone un modelo de

arquitectura en donde es desplegada una aplicación que dota al reproductor VLC con las funcionalidades del estándar DASH. Además, esta arquitectura está caracterizada por lograr una gran flexibilidad en el componente encargado de la lógica de adaptación, cuestión que puede adaptarse al desarrollo de los objetivos propuestos en el presente trabajo de grado. El artículo también aporta información relevante sobre diversos métodos para la indexación de segmentos dentro del archivo MPD, tanto para contenidos multimedia bajo demanda, como para sesiones en vivo.

***Dynamic Adaptive Streaming over HTTP: From Content Creation to Consumption*** [26]. Este artículo presenta una guía sobre *streaming* dinámico adaptativo a través de HTTP, partiendo desde la creación de contenidos hasta su consumo. En particular, proporciona una visión general del estándar DASH; el cómo crear contenido para ser entregado usando DASH, su consumo, así como la evaluación de dicho contenido con respecto a las soluciones de la industria de la competencia. En cuanto a la creación de contenidos están incluidos temas relacionados con el formato adecuado de codificación, tasas de bits, resolución, velocidad de fotogramas, la longitud del segmento y la calidad actual del contenido multimedia. Además, tiene en cuenta si las diferentes modalidades de contenido como audio, vídeo, texto son multiplexados o proveídos como tramas individuales. Respecto a la entrega de contenido usando DASH, el artículo expone una de las ventajas más importantes del estándar; la utilización de la infraestructura actual de distribución de contenidos (proxys, caches, etc.). Así mismo, menciona que los patrones de tráfico estudiados en *streaming* DASH difieren de los tráficos conocidos, de tal manera que los efectos positivos y negativos aún no han sido evaluados.

En cuanto al consumo de contenidos usando DASH, este trabajo muestra que a diferencia de los métodos antiguos de *streaming*, los clientes en lugar de los servidores toman las decisiones oportunas y son encargados de la adaptación. En éste punto son involucradas las aplicaciones de cliente, la lógica de adaptación, la estimación del ancho de banda, y la programación de solicitud de segmentos; en consecuencia, las decisiones están influenciadas por las capacidades del dispositivo y tienen en cuenta la satisfacción del usuario. Dado que una buena adaptación tiene por objeto ofrecer la mejor experiencia posible al usuario, el trabajo también expone diferencias entre las políticas de adaptación más conocidas. Así

mismo, en cuanto a las variaciones en el rendimiento de la red, está indicado cómo abordar diferentes situaciones en donde aparecen cuellos de botella en la red.

Este artículo hace referencia a varios trabajos relacionados con el estándar [19], [20], [21], y deja claro que el *streaming* adaptativo dinámico es poco explorado en comparación con los enfoques clásicos de *streaming*, por lo cual requiere investigación para identificar cómo los componentes interactúan unos con otros, cómo utilizar técnicas alternativas de compresión, cómo aplicar el estándar bajo el contexto de aplicaciones y cómo optimizar su rendimiento. Aunque este artículo presenta una guía que parte desde la creación de contenidos DASH hasta el consumo de los mismos, no vincula servicios adicionales ligados al estándar, por lo tanto el presente trabajo de grado pretende aportar un servicio de video bajo demanda, de contenidos multimedia basados en el estándar DASH y la adecuación de un cliente web para permitir el despliegue de aplicaciones interactivas vinculadas al servicio de video bajo demanda, además proporcionar un conjunto de herramientas para facilitar la generación de contenidos multimedia DASH-WebM.

**Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON [27].** Este artículo presenta un esquema para el consumo de servicios interactivos de la web 2.0 (Foros, Micro-Blog, Chat), sobre diversos entornos de TDi (Televisión Digital Terrestre, Televisión Móvil, IPTV), propuestos por el proyecto ST-CAV de la Universidad del Cauca, cuyo objetivo fue el despliegue de comunidades académicas virtuales (CAV) en entornos de televisión. El esquema de consumo de servicios está basado en el estilo arquitectónico REST-JSON, el cual permite el consumo liviano de 2 servicios: el servicio de tablón y acceso a correo. En éste artículo son presentadas importantes ventajas asociadas al uso del protocolo REST-JSON, una de ellas es permitir un diseño sencillo y flexible para servicios consumidos a través de Internet, lo cual facilita la integración de estos en diversos escenarios, además que el diseño e implementación de servicios REST-JSON son independientes de la lógica de presentación, en la interfaz del televisor o del dispositivo móvil, lo cual permite extender el escenario de aplicación de los servicios a otros entornos como el IPTV.

Sin embargo este artículo solo define un esquema liviano para el consumo de servicios interactivos, mas no para contenidos multimedia adaptativos, por lo tanto el presente trabajo de grado pretende aportar un servicio de video bajo demanda, de contenidos multimedia DASH-WebM, utilizando un esquema liviano que permita

el intercambio de datos entre las aplicaciones interactivas vinculadas al servicio, para facilitar el acceso a dichas aplicaciones y a los recursos de red.

**DASH: Un estándar MPEG para *Streaming* sobre HTTP** [16]. Este trabajo presenta la implementación de una aplicación de escritorio en Java, que permite validarse en la plataforma MIPAMS (*Multimedia Information Protection and Management System*) con el fin de reproducir el contenido multimedia de un MPD. La aplicación permite seleccionar entre una determinada calidad durante toda la reproducción o un cambio de calidad automático, de tal forma que puedan observarse las transiciones de calidad del contenido multimedia, en función del ancho de banda disponible y de la capacidad de la CPU del usuario, como también, simular los cambios en el ancho de banda, de manera que sea comprobado el cambio de calidad en función de los cambios de red. Esta aplicación permite interpretar los archivos descriptores disponibles en una plataforma privada para la posterior reproducción del contenido multimedia, en función de las variaciones de ancho de banda.

Este trabajo no aporta la creación de los descriptores mencionados, es decir, no establece su propio modelo jerárquico de datos, en donde sean definidas la segmentación, periodos de adaptación, resoluciones, tasa de bits, fotogramas, idioma y demás características que permiten personalizar las pruebas y abordar la implementación de servicios y aplicaciones interactivas vinculadas al mismo.

Este artículo hace evidente la necesidad de un banco de pruebas para el desarrollo de prácticas basadas en el estándar DASH. Teniendo en cuenta lo anterior, el trabajo aporta referencias teóricas significativas sobre el funcionamiento de los archivos descriptores, además que prueba la verdadera función del estándar al hacer un contenido multimedia adaptativo y dinámico, por lo tanto sienta bases importantes en el diseño y uso de herramientas para el consumo del estándar DASH.

***A Test-Bed for the Dynamic Adaptive Streaming over HTTP featuring Session Mobility*** [28]. Este trabajo presenta un banco de pruebas de contenidos multimedia que permite la movilidad de sesiones, a través del estándar DASH. La movilidad de una sesión es entendida como la transferencia de una sesión de *streaming* desde un dispositivo a otro, realizándose de forma adaptativa dependiendo de las



capacidades del dispositivo para el cual la sesión es transferida. El sistema propuesto pretende, además, la realización de la transferencia real de forma eficaz e interoperable, con ayuda de un prototipo de aplicación integrada en el reproductor VLC. Lo anterior consiste en lo siguiente: un usuario desde un dispositivo, puede haber iniciado la descarga progresiva de contenidos multimedia adaptativo, soportados en el estándar DASH, a través del reproductor VLC, y en determinado instante decide cambiar de dispositivo, entonces el usuario mediante un menú de opciones, puede elegir el nuevo dispositivo de recepción y además decidir la configuración de su sesión que desea transferir, dicha configuración es enviada al otro dispositivo, e inclusive la reproducción del contenido multimedia, puede continuar en el mismo instante en el que fue detenido, mientras era desplegado en el anterior dispositivo. Los resultados muestran que la *interoperabilidad* es conseguida gracias a la adopción de los estándares existentes, pero el rendimiento de este sistema no depende de dichos estándares, sin embargo, los módulos responsables para el rendimiento del sistema, están diseñados de una manera extensible y capaz de adaptarse a tales circunstancias.

Este trabajo incluye aportes importantes tales como, un menú interactivo que es visualizado al hacer clic sobre el video en reproducción y la adaptación de un cliente DASH, para permitir el despliegue del menú y la movilidad de una sesión, sin embargo la información de configuración de la sesión es manejada por un archivo independiente en formato XML, y no por el archivo MPD especificado por el estándar DASH, asimismo la recepción es realizada a través de un cliente DASH y no por medio de un cliente Web.

De esta manera, el presente trabajo de grado pretende integrar la descripción del servicio de video bajo demanda y la descripción de contenidos multimedia adaptativos, soportados en el estándar DASH, en un único archivo MPD, a través de la implementación de un esquema liviano que permita el intercambio de datos entre las aplicaciones interactivas vinculadas al servicio de video bajo demanda, asimismo este trabajo de grado propone la adecuación de un cliente web, para permitir el despliegue de aplicaciones interactivas vinculadas al servicio de video bajo demanda, descrito en el archivo MPD especificado por el estándar.

***A Seamless Web Integration of Adaptive HTTP streaming*** [29]. Este artículo presenta la integración del estándar DASH en un entorno web, mediante el uso de los elementos de video que provee HTML5 y la API de medios del navegador web

de Google, en la herramienta DASH-JS, la cual haciendo uso de Javascript permite el consumo de contenidos multimedia soportados en el estándar DASH. Este trabajo presenta además, la manera de describir contenidos multimedia WebM en el archivo MPD, especificado por el estándar DASH. Al final del artículo, es realizada una evaluación para comprobar las capacidades de adaptación al cambio del ancho de banda, del sistema implementado. De esta manera, este trabajo aporta bases importantes sobre la forma de integrar el estándar DASH en un entorno web, a través de la implementación de un cliente DASH basado en JavaScript y la forma de integrar el formato WebM, como contenido multimedia adaptativo DASH, lo cual es un aporte importante, para la implementación del servicio de VoD, de contenido multimedia adaptativo, propuesto en este trabajo de grado.

***In-Segment Content Server Adaptation for Dual Adaptation Mechanism in DASH*** [30]. Este artículo presenta la implementación y las pruebas de una nueva solución de adaptación de medios, la cual aprovecha la capacidad de adaptación dual de un servidor de contenidos multimedia, realizada durante la transmisión de segmentos de contenidos DASH, como alternativa y complemento para las soluciones de adaptación de medios actuales especificadas por este estándar, logrando así contrarrestar ciertos efectos inherentes a la transmisión de contenidos y alcanzando mayor calidad de experiencia (QoE), en caso de que exista una situación de sobrecarga en el servidor o de la red central. Las diversas pruebas realizadas en este trabajo muestran la capacidad y el límite de sobrecarga admitida en un determinado servidor de contenidos, aportando las bases para realizar la evaluación del funcionamiento de un servicio de video bajo demanda de contenidos DASH-WebM, mediante pruebas de estrés y variación del ancho de banda, lo cual es uno de los objetivos específicos de este trabajo de grado.

## 1.6 Aportes

A continuación son presentados los aportes académicos y de valor agregado de este trabajo:

- Generación de un banco de pruebas que incluye las herramientas apropiadas para la codificación, difusión y consumo de servicios de video bajo demanda, de contenidos multimedia adaptativos, basados en el estándar DASH. Dicho

banco pretende ser un punto de partida para la implementación futura de servicios avanzados bajo el escenario descrito, además de ampliar el campo de acción de los mismos en las nuevas tecnologías que han venido desarrollándose.

- Implementación de un servicio de video bajo demanda para contenidos multimedia adaptativos, basados en el estándar DASH. Considerando que este estándar está definido para el soporte básico de *streaming* de contenido multimedia adaptativo sobre HTTP y no para la implementación del servicio de video bajo demanda, este aporte permite dotar a DASH de la funcionalidad de dicho servicio incluyendo el consumo de aplicaciones interactivas.
- Diseño e implementación de una herramienta para la automatización del proceso de codificación de contenidos multimedia adaptativos, basados en el estándar DASH, la cual permite realizar procesos de codificación, segmentación de contenidos y generación del manifiesto de configuración especificado por el estándar, logrando así, mejorar la eficiencia en la generación de los formatos soportados.
- Implementación de un esquema liviano para el consumo del servicio de video bajo demanda de contenidos multimedia DASH-WebM, para permitir el intercambio de datos de las aplicaciones interactivas vinculadas al servicio, lo anterior teniendo en cuenta que el estándar DASH solamente define el consumo de contenidos multimedia.
- Adecuación de un cliente web, que permite el despliegue de aplicaciones interactivas y contenidos multimedia DASH-WebM, vinculados al servicio de video bajo demanda. Lo anterior considerando que el archivo manifiesto especificado por el estándar DASH solo contiene información del contenido multimedia y no la información de las aplicaciones interactivas vinculadas al servicio de video bajo demanda.

## 1.7 Objetivos

### 1.7.1 Objetivo general

Proporcionar<sup>4</sup> un banco de pruebas para el soporte del servicio de video bajo demanda de contenidos multimedia adaptativos, basados en el estándar DASH (ISO/IEC 23009-1).

### 1.7.2 Objetivos específicos

- Evaluar diferentes herramientas para la codificación, difusión y recepción de contenidos multimedia adaptativos, soportados en el estándar DASH (ISO/IEC 23009-1).
- Diseñar un banco de pruebas para el soporte del servicio de video bajo demanda de contenidos multimedia adaptativos, basados en el estándar DASH (ISO/IEC 23009-1).
- Implementar un banco de pruebas para el soporte del servicio de video bajo demanda de contenidos multimedia adaptativos, basados en el estándar DASH (ISO/IEC 23009-1).
- Evaluar mediante pruebas de estrés y variación del ancho de banda, el funcionamiento del servicio de video bajo demanda, de contenidos multimedia adaptativos, basados en el estándar DASH (ISO/IEC 23009-1).

## 1.8 Actividades

En esta sección son planteadas las actividades pertinentes, para el desarrollo del presente trabajo, las actividades a saber están descritas a continuación en la tabla 1.1:

---

<sup>4</sup> A través del objetivo general se pretende la adecuación de un conjunto de herramientas de *streaming* adaptativo, para la generación de un banco de pruebas para el soporte del servicio de VoD. Un banco de pruebas es un conjunto de herramientas software, que proporcionan un entorno uniforme para explorar, probar, comparar y reproducir diversas tecnologías, permitiendo así, una amplia gama de actividades de verificación y desarrollo.

**CODIGO ACTIVIDAD**

<b>A0</b>	Formulación del trabajo de grado
<b>A01</b>	Elaboración de la propuesta de trabajo de grado.
<b>A02</b>	Elaboración del anteproyecto del trabajo de grado.
<b>A1</b>	Evaluación de herramientas para el diseño del banco de pruebas
<b>A11</b>	Recopilación de herramientas para la codificación, difusión y recepción de contenido multimedia adaptativo.
<b>A12</b>	Evaluación y comparación de las herramientas de codificación, difusión y recepción de contenido multimedia adaptativo.
<b>A2</b>	Diseño del banco de pruebas
<b>A21</b>	Definición de herramientas apropiadas para codificación, difusión y recepción de contenido multimedia adaptativo.
<b>A22</b>	Definición de la arquitectura del banco de pruebas para el soporte del servicio de VoD de contenido multimedia adaptativo.
<b>A3</b>	Implementación del banco de pruebas
<b>A31</b>	Montaje del banco de pruebas base, con las herramientas de codificación, difusión y recepción seleccionadas.
<b>A32</b>	Implementación de la herramienta de automatización del proceso de codificación de contenido multimedia adaptativo.

<b>A33</b>	Desarrollo del servicio de VoD base.
<b>A34</b>	Adecuación de un esquema para el consumo de aplicaciones interactivas, asociadas al servicio de VoD.
<b>A35</b>	Desarrollo de una aplicación interactiva base, vinculada al servicio de VoD.
<b>A36</b>	Adecuación de un cliente web para el consumo del servicio de VoD.
<b>A4</b>	Evaluación del banco de pruebas
<b>A41</b>	Evaluación del funcionamiento de cada uno de los módulos funcionales del banco de pruebas implementado.
<b>A42</b>	Evaluación del comportamiento del servicio de VoD mediante pruebas de estrés y de variación del ancho de banda.
<b>A5</b>	Generación de entregables
<b>A51</b>	Elaboración del manual de instalación y configuración del banco de pruebas.
<b>A52</b>	Elaboración de por lo menos un artículo de investigación para congreso nacional o revista indexada.
<b>A53</b>	Elaboración de la Monografía.

**Tabla 1.1.** Actividades para el desarrollo del trabajo de grado.

## 1.9 Estructura del trabajo de grado

El contenido de la monografía del presente trabajo de grado está organizado en 6 capítulos, descritos de la siguiente manera:

*Capítulo 2* – Banco de pruebas: presenta el escenario extremo a extremo de *streaming* adaptativo Dash, el cual fue configurado con ayuda de la herramienta de codificación automática Dash WebM Converter y un cliente web para el despliegue del servicio de video bajo demanda y de las aplicaciones interactivas vinculadas al mismo. El escenario está formado por 3 módulos principales: módulo de codificación, módulo de difusión y módulo de recepción.

*Capítulo 3* – Herramienta de codificación automática: presenta el diseño, la implementación y el despliegue de la herramienta de codificación automática Dash WebM Converter, configurada para el escenario de *streaming* adaptativo propuesto en el capítulo anterior.

*Capítulo 4* - Servicio de video bajo demanda: presenta el diseño, la implementación y el despliegue del servicio de video bajo demanda y de las aplicaciones interactivas vinculadas al mismo, sobre el escenario de *streaming* adaptativo planteado en el capítulo 2.

*Capítulo 5* - Resultados y Pruebas: contiene los resultados obtenidos mediante el presente trabajo, así como también las pruebas hechas sobre la herramienta Dash WebM Converter, el cliente web y el servicio de video bajo demanda y tiene como propósito validar el funcionamiento de las herramientas y los servicios desplegados dentro del escenario Dash extremo a extremo.

*Capítulo 6* – Conclusiones: son planteadas las conclusiones, derivadas del desarrollo del presente trabajo de grado, asimismo las sugerencias y posibles trabajos futuros.

## Capítulo 2

### Banco de pruebas

#### 2.1 Introducción

El video *streaming* es una técnica para la distribución de contenido multimedia, en la cual no es necesario esperar a que lleguen todos los flujos de video del archivo solicitado para iniciar la reproducción. Tradicionalmente esta técnica ha sido soportada por los protocolos RTP y RTSP, de tal modo que el servidor debe gestionar una sesión diferente para cada cliente y coordinar la entrega de paquetes. Actualmente el estándar de *streaming* adaptativo DASH ofrece un enfoque diferente mediante el uso del protocolo HTTP, de tal forma que el cliente extrae los datos del servidor, sin que sea necesario mantener el estado de la sesión. De esta forma, tiene como ventajas el pleno uso de la infraestructura existente en Internet, y la adaptación del contenido multimedia a las condiciones de ancho de banda de la red.

A pesar de que el estándar DASH solo especifica la entrega de contenidos multimedia adaptativos y de acuerdo al estado del arte presentado en el anterior capítulo, es necesario considerar las brechas y necesidades a la hora de brindar servicios como el VoD y demás servicios interactivos que pueden estar asociados al mismo. Además, no existe una herramienta automática para el proceso de generación de contenidos multimedia DASH en el conocido formato WebM, tampoco una herramienta que trabaje en este proceso mediante la asignación de tareas independientes o multitarea. Una forma de enfrentar estos requerimientos es mediante la generación de un banco de pruebas que integre los elementos necesarios para resolver los anteriores inconvenientes.



En este capítulo es planteada una arquitectura base para el despliegue de un escenario extremo a extremo para la transmisión y consumo de *streaming* adaptativo DASH, el cual fue configurado con ayuda de la herramienta DASH WebM Converter, desarrollada en este trabajo de grado, la cual permite automatizar el proceso de generación de contenidos multimedia WebM de acuerdo al formato del estándar DASH.

La estructura de este capítulo está organizada de la siguiente manera: primero son definidos los requisitos para el desarrollo del presente proyecto, seguidamente son descritos los conceptos, herramientas y tecnologías más relevantes, utilizadas en este capítulo y el resto de este trabajo, después es descrita la metodología utilizada para cumplir los requisitos planteados al inicio de este capítulo. Finalmente son descritos cada uno de los módulos y vistas que conforman el escenario propuesto.

## 2.2 Requisitos

Esta sección presenta el listado inicial de requisitos planteados para el desarrollo de este capítulo. Estos requisitos están de acuerdo a los objetivos y aportes presentados en el capítulo 1 de esta monografía:

### Requisitos funcionales

- Es requerido la implementación de una herramienta para automatizar el proceso de generación de contenidos multimedia DASH en formato WebM.
- Es necesaria la exploración y selección de un conjunto de herramientas libres para realizar los procesos de codificación, adaptación, difusión y consumo de contenidos multimedia DASH en formato WebM.
- Deben utilizarse contenidos WebM para los procesos de codificación y adaptación al estándar DASH, por medio de la herramienta de codificación automática desarrollada.

- La herramienta de codificación automática desarrollada debe permitir la generación de los flujos de video y audio a diferentes tasas de bit de manera simultánea.
- La herramienta de codificación automática debe tener una versión en línea, que permita ejecutar tareas de codificación a través de un cliente web.
- La codificación en línea debe permitir de manera simultánea, la publicación de los contenidos multimedia DASH en el servicio de VoD.
- Para la herramienta de codificación automática debe existir una versión offline, para realizar tareas de codificación desde una aplicación de escritorio.
- Es requerido que los contenidos multimedia DASH transmitidos a través de la red sean adaptados a la variación de ancho de banda de la misma.
- Debe visualizarse gráficamente la variación del ancho de banda de la red, a medida que los contenidos multimedia DASH sean consumidos.
- Es requerido la implementación de un servicio de VoD básico de contenidos multimedia DASH, el cual tenga asociado un catálogo de contenidos.
- Es necesario contar con un sistema de votación tipo *Star Rating*, para la valoración de los contenidos multimedia DASH, consumidos por los usuarios.
- Es requerido un sistema de conteo de reproducciones o visitas realizadas a un determinado contenido multimedia DASH.
- Los contenidos multimedia DASH deben presentarse de forma ordenada, de acuerdo al número de visitas realizadas por los usuarios del servicio de VoD.
- Es necesario contar con una herramienta de medición, que permita obtener la cantidad de memoria RAM y el porcentaje de uso de CPU consumidos por la herramienta de codificación automática desarrollada.

- Es necesario especificar las condiciones y requisitos técnicos que deben cumplirse del lado del cliente, para el adecuado funcionamiento del servicio de VoD.
- Es conveniente contar con un servidor dedicado para el proceso de codificación automática.
- Además debe implementarse un servidor únicamente para acceso a los servicios interactivos.

### **Requisitos no funcionales**

- El banco de pruebas debe diseñarse e implementarse de forma modular, para facilitar el desarrollo de servicios y la organización del mismo.
- Es requerido la implementación de servicios interactivos livianos y flexibles adaptados a este entorno de pruebas, que puedan ser integrados al servicio VoD.
- El banco de pruebas debe tener características de *escalabilidad*, de manera que permita la fácil integración y desarrollo de nuevos servicios interactivos.
- Las aplicaciones interactivas asociadas al servicio de VoD deben ser interoperables con plataformas de televisión relacionadas (IPTV, TV Móvil, TDT).

## **2.3 Marco teórico**

A continuación son descritos los conceptos, herramientas y tecnologías claves, exploradas para el desarrollo del presente trabajo. A partir de las herramientas exploradas, fueron seleccionadas un conjunto de estas para la conformación del banco de pruebas propuesto. Estas permiten realizar tareas de codificación, difusión y consumo de contenidos multimedia DASH.

## Video

Básicamente es una secuencia de imágenes, que formando un encadenamiento contiene escenas en movimiento [34]. La generación de video requiere dispositivos electrónicos que permitan la captura, procesamiento, almacenamiento y reconstrucción de dichas imágenes. Para el almacenamiento de video digital es requerido el uso de algún tipo de software llamado códec. Generalmente es comprimido en algún tipo de formato (MPEG-4, VP8, H.264, WMV, etc.) para facilitar su almacenamiento y transmisión, debido al ancho de banda limitado de algunas redes de datos. Una característica propia del video digital es la resolución, la cual es medida por la cantidad de píxeles verticales y horizontales que componen la imagen.

## Codec

Es un software que contiene un conjunto de algoritmos para codificar y decodificar video y audio digital [33]. Una de las principales funciones de un *codec* es reducir el tamaño de video para facilitar su procesamiento y almacenamiento, encapsulando este mismo en algún tipo de formato. La visualización de un determinado formato de video requiere del *codec* adecuado.

## WebM

Es un formato contenedor multimedia libre, desarrollado por Google Inc. para ser utilizado en la web con HTML5 [38]. WebM utiliza los formatos Vorbis para audio y VPx para video. Vorbis es un *codec* libre desarrollado por Xiph.org, de igual manera la serie de versiones del *codec* VPx actualmente adquirido por Google son *open source*. Un archivo WebM contiene agrupaciones separadas por fotogramas clave ubicados al inicio de estas, los cuales permiten una búsqueda rápida para el cliente.

## MP4Box

Es un paquete de herramientas desarrolladas por GPAC [35], destinadas para trabajar sobre diversos contenidos multimedia, particularmente los de la especificación ISO multimedia (MP4, 3GP). MP4Box es ejecutado mediante línea de comandos y es utilizado para cifrar, codificar y *multiplexar* flujos de video y audio. Además permite adaptar contenidos MP4 al estándar DASH, este proceso consiste

en la generación de múltiples segmentos de contenido en formato .m4s a partir de un archivo multimedia MP4 fuente, dichos segmentos son descritos en el archivo MPD especificado por el estándar. MP4Box trabaja sobre plataformas Windows, Linux, Mac Os X, Android, iOS, entre otras.

### **Osmo4**

Es un reproductor multimedia de GPAC el cual permite el consumo de contenidos multimedia DASH en formato MP4, está apoyado en la mayoría de protocolos de entrega existentes haciendo posible la reproducción de cualquier tipo de contenido audiovisual. Está enfocado a los gráficos, animaciones y tecnologías de interactividad, además reproduce contenidos 2D y 3D [41]. Esta herramienta puede ejecutarse como una aplicación con interfaz gráfica (GUI) o como una consola de comandos, funciona además como un *plugin* embebido en navegadores web como Firefox, Opera o Internet Explorer.

### **VLC Player**

Es un reproductor multimedia libre y multiplataforma (Windows, Linux, Solaris, Mac OS X, etc.), soporta múltiples formatos de video y audio (MPEG-1, MPEG-2, MPEG-4, DVD, VCD, CD-ROM, archivos DivX y 3GP, etc.), además soporta transmisiones *multicast* a través de IPV4 o IPV6 y *streaming* adaptativo del estándar DASH [9]. VLC permite la recepción de contenido bajo demanda usando los protocolos RTP, RSTP y RTSP.

### **Red5**

Es una herramienta libre utilizada para realizar *streaming* de video bajo demanda, en vivo, video chat y conferencia. Este servidor está desarrollado en el lenguaje Java y puede detectar la velocidad de conexión de los clientes además de proporcionar la mejor secuencia de video a los mismos. Red5 soporta los formatos de video FLV, MP4, 3GP y F4V, también permite la grabación de *streaming* proveniente de sus clientes en formato FLV y AVC-ACC [47].

## **Darwin**

Es un servidor libre de *streaming* multimedia, que utiliza los protocolos RTP/RSTP, el cual fue desarrollado por Apple en 1999. Es una herramienta que proporciona un alto nivel de personalización, utilizada para realizar *streaming* de video bajo demanda, en vivo y video chat [46]. Darwin funciona sobre plataformas Mac OS, Linux, Windows, FreeBSD, Tru64 Unix y Solaris. Además soporta múltiples formatos multimedia incluyendo H.264/MPEG-4 AVC y 3GP.

## **Live555**

Es un servidor de *streaming* multimedia, el cual utiliza los protocolos RTP, RTSP y SDP (*Session Description Protocol*) [39], sin embargo no soporta *streaming* adaptativo. Live555 es libre y de código abierto, lo cual significa que puede modificarse y configurarse de acuerdo a requerimientos propios. Algunos de los formatos multimedia soportados por este servidor están: MPEG-2 (.mpeg), MPEG-4 (.m4e), MPEG Transport Stream (.ts), WebM (.webm), H264 (.264), MP3 (.mp3), AAC (.aac), etc.

## **Apache**

Es un servidor web HTTP de código abierto, destacado por su seguridad (SSL y TLS), rendimiento (capaz de manejar más de un millón de visitas por día) y por ser muy flexible y altamente configurable con módulos como PHP, Python, Perl, Cband, Ruby, entre otros [42]. Apache es el servidor web por defecto en muchas distribuciones Linux. Su versión local es muy útil para los desarrolladores de aplicaciones web. Su archivo `apache2.conf` es el más importante, puesto que es el que define el comportamiento general del servidor y es el encargado de acceso a los distintos módulos del mismo.

## **Apache Bench (ab)**

Es una herramienta libre, la cual es ejecutada mediante línea de comandos. Es utilizada para evaluar el rendimiento de servidores web. Esta herramienta viene con la versión estándar del código fuente de Apache y permite realizar pruebas de estrés de modo secuencial o concurrente para servicios que hacen uso de

peticiones HTTP, independientemente del lenguaje o framework que fue utilizado para desarrollarlos [40].

## **FFMPEG**

Es una herramienta de línea de comandos perteneciente al proyecto libre FFMPEG, cuyas funciones principales son: convertir archivos multimedia entre diferentes formatos, cambiar el tamaño de la imagen de video, generar archivos multimedia a diferentes frecuencias de muestreo y velocidad de fotogramas, entre otras. FFMPEG contiene múltiples bibliotecas útiles para codificar, decodificar, mux y demux, prácticamente cualquier tipo de archivo multimedia [36]. Dentro del presente trabajo esta herramienta fue utilizada para cambiar el formato original de diferentes contenidos multimedia al formato WebM. Así mismo FFMPEG es utilizada en segundo plano por la herramienta de codificación automática, con el fin de obtener múltiples flujos de audio y video a diferentes tasas de bits, a partir de un archivo multimedia WebM base.

## **Libwebm**

Es una biblioteca de código abierto asociada al proyecto WebM de Google Inc., la cual contiene el *script* `sample_muxer`, escrito en lenguaje C++. Este es utilizado para agregar índices de búsqueda para cada fotograma clave y para alinear puntos de sincronización en archivos multimedia WebM. Esta biblioteca provee además software para el control de *streaming* DASH y el acceso a los segmentos multimedia descargables. El *script* `sample_muxer` es utilizado en segundo plano por la herramienta propuesta en el presente trabajo, con el fin de agregar los índices de búsqueda a los contenidos multimedia previamente codificados en el formato WebM [45].

## **webm-tools**

Es otra biblioteca de código abierto asociada al proyecto WebM de Google Inc., la cual contiene *scripts* escritos en el lenguaje C++, útiles para cifrar flujos de video y generar el archivo de manifiesto XML DASH, usado para describir el contenido multimedia del *streaming* adaptativo. El *script* `webm_dash_manifest` de la biblioteca `webm-tools` es utilizado en segundo plano por la herramienta propuesta en este

trabajo, para la generación del archivo de manifiesto XML DASH de los contenidos multimedia previamente codificados e indexados [45].

### **webm-dash-javascript**

Es una librería basada en JavaScript [37], la cual permite la integración del estándar de *streaming* adaptativo DASH en la Web, para lo cual hace uso de las extensiones para manejo de medios en HTML5 propias del navegador Google Chrome (Media Source API). Gracias a sus componentes `bandwidth_manager.js`, `dash_player.js`, `dash_parser.js`, `webm_parser.js`, entre otros, es posible realizar el consumo de segmentos de medios basados en WebM y segmentos basados en el formato de archivo ISO Base Media. El presente trabajo hizo uso de las anteriores librerías javascript y del componente HTML5 de reproducción, para el consumo del contenido multimedia en formato DASH desde una página web en el navegador Google Chrome.

### **JavaScript**

Es un lenguaje de programación que permite generar acciones sobre páginas web, dándoles la característica de ser dinámicas. Funciona del lado del cliente por lo tanto no requiere de compilación, es interpretado por un navegador web [44]. Aunque JavaScript es soportado en la mayoría de los navegadores, este puede ser deshabilitado por el usuario.

### **Python**

Es un lenguaje de programación interpretado y de alto nivel, está enfocado a la facilidad y rapidez de desarrollo. Gracias a su gran conjunto de librerías es multipropósito. Permite desde el desarrollo de aplicaciones sencillas hasta la implementación de servidores de red o incluso páginas web. Otras características importantes es que está orientado a objetos y es gratuito incluso para propósitos empresariales. Por ser un lenguaje de script o interpretado, está limitado en rapidez de ejecución [43].



### Selección de herramientas

La tabla 2.1 resume las herramientas que fueron seleccionadas para el desarrollo del banco de pruebas planteado en el presente trabajo:

Herramientas seleccionadas	Herramientas	Descripción
<b>Codificación:</b> FFmpeg Libwebm Webm-tools	MP4Box	Permite la generación de contenidos multimedia DASH en formato MP4, junto con su archivo descriptor MPD.
	FFmpeg	Permite la codificación y adaptación previa de contenidos multimedia WebM para ser adecuados al estándar DASH.
	Libwebm	Permite agregar índices de búsqueda para fotogramas clave a flujos de video y audio en formato WebM para ser soportados por el estándar DASH.
	Webm-tools	Permite generar un archivo descriptor MPD para flujos de video y audio en formato WebM, previamente adaptados desde la herramienta LibWebm.
<b>Difusión:</b> Apache	Live555	Servidor de <i>streaming</i> multimedia basado en RTP, no permite ofrecer <i>streaming</i> adaptativo ni desplegar servicios.
	Darwin	Servidor de <i>streaming</i> multimedia basado en RTP, no permite ofrecer <i>streaming</i> adaptativo ni desplegar servicios.

	Apache	Servidor web HTTP, útil para realizar <i>streaming</i> adaptativo DASH y desplegar servicios.
	Red5	Es una herramienta utilizada para realizar <i>streaming</i> de video bajo demanda, en vivo y video chat. No soporta el formato multimedia WebM.
<b>Recepción:</b> Webm-dash- javascript	VLC player	Reproductor multimedia que permite el consumo de DASH.
	Webm-dash- javascript	Ciente web que permite el consumo de contenidos multimedia DASH en formato WebM.
	Osmo4	Reproductor multimedia que permite el consumo de contenidos DASH en formato MP4.

**Tabla 2.1.** Herramientas seleccionadas para conformar el banco de pruebas.

Para el proceso de codificación automática fueron seleccionadas las herramientas: FFmpeg por su potencia y capacidad para la codificación de contenidos multimedia, LibWebM y Webm-tools por ser desarrolladas específicamente para la adaptación de contenidos multimedia DASH WebM. Además es utilizado Python por su potencia, flexibilidad y facilidad en programación, por su disposición en sistemas operativos Linux y la posibilidad de trabajar en multitarea y en modo servidor.

Para la disposición de los servicios y la difusión de contenidos multimedia DASH fue seleccionado el servidor Apache por su facilidad de manejo y capacidad adecuada para este tipo de entorno de pruebas.

Para el consumo de los servicios y contenidos multimedia DASH es necesario el uso del navegador Chrome de Google a partir de la versión 17 o superiores a esta, puesto que hasta el momento este soporta contenido multimedia DASH en formato

WebM, de manera fácil. Webm-dash-javascript fue adaptada e integrada al servicio de VoD para permitir el despliegue del contenido multimedia.

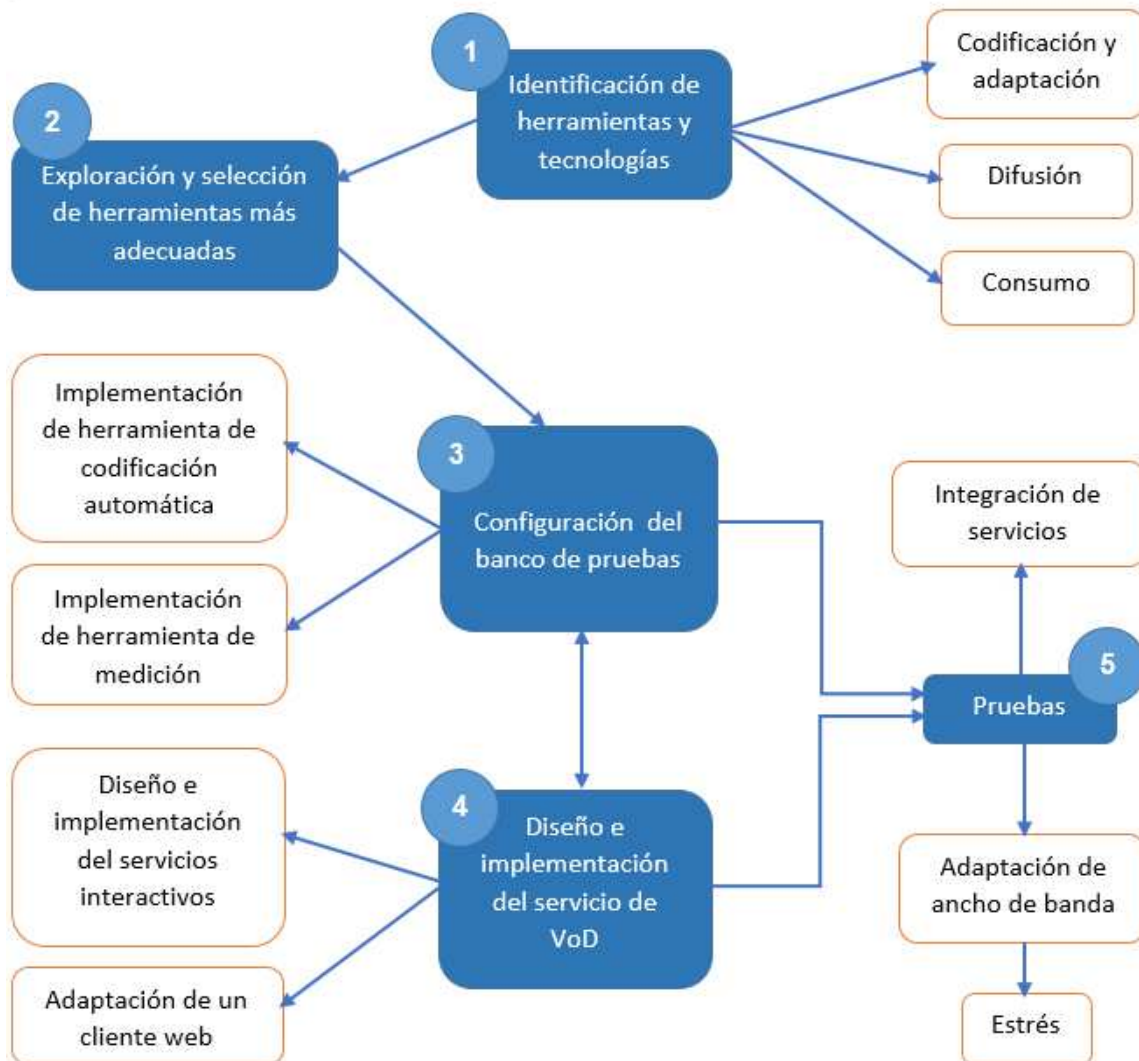
## 2.4 Metodología de trabajo

Para la consecución de los requisitos descritos al comienzo de este capítulo, fueron definidos cuatro fases principales, las cuales son mostradas en la figura 2.1. El desarrollo modular de cada fase permitió una metodología de trabajo secuencial, iterativa e incremental.

La primera fase comprende la identificación de herramientas libres, que permiten dentro del escenario de *streaming* adaptativo DASH, las funciones de codificación y adaptación de contenidos multimedia WebM de acuerdo al estándar. Así mismo, esta fase incluye las herramientas de difusión de contenidos multimedia a través de la red y su consumo desde el servicio de VoD.

En la segunda fase fue realizada la selección y exploración de las herramientas más adecuadas para la configuración del banco de pruebas propuesto. Estas herramientas fueron instaladas, configuradas y puestas en marcha para comprobar las principales características de funcionamiento.

En la fase de configuración del banco de pruebas, fue desarrollada la herramienta DASH WebM Converter, la cual permite la codificación automática de contenidos multimedia WebM al formato DASH, también fue implementada la herramienta de medición de consumo de memoria RAM y uso de CPU, planteada en los requisitos de la anterior sección de este capítulo.



**Figura 2.1.** Metodología de trabajo.

En la cuarta fase fue diseñado e implementado el servicio de VoD básico, el cual contiene un catálogo de contenidos multimedia y un reproductor de dichos contenidos. Después de esto, los servicios interactivos fueron implementados y asociados al servicio de VoD. Para concluir esta fase fue necesario adaptar un cliente web para permitir el consumo de estos servicios.

Finalmente en la quinta fase de pruebas, fue comprobado el adecuado funcionamiento del escenario de *streaming* adaptativo, mediante pruebas de integración de servicios, pruebas de adaptación de ancho de banda y pruebas de

estrés. Esta fase fue ejecutada de manera simultánea a las de configuración del banco de pruebas y diseño e implementación del servicio de VoD.

## 2.5 Planteamiento del banco de pruebas

Un banco de pruebas es un conjunto de herramientas software, que proporcionan un entorno uniforme para explorar, probar, comparar y reproducir diversas tecnologías, permitiendo así, una amplia gama de actividades de verificación y desarrollo [18], [19] y [20]. La comprobación a través de un banco de pruebas brinda transparencia y permite recrear procesos de experimentación. Su diseño y construcción inicial requiere de una preselección de elementos acordes al objetivo de desarrollo.

Para la configuración del banco de pruebas planteado en este trabajo fueron consideradas diferentes vistas, las cuales son descritas a continuación en la figura 2.2.



**Figura 2.2.** Vistas para el desarrollo del banco de pruebas.

La vista de negocio define la cadena lógica de funcionamiento del banco de pruebas planteado. La vista de servicios contiene la descripción de los servicios interactivos asociados al servicio de VoD que fueron desarrollados. Otra vista es la de flujo de trabajo en la cual es mostrado la secuencia de pasos necesarios para el correcto funcionamiento del escenario de *streaming* adaptativo DASH. La vista de infraestructura contiene el entorno general de desarrollo. A continuación son descritas cada una de las vistas que definen el banco de pruebas desarrollado.

### 2.5.1 Vista de negocio

Esta vista presenta la cadena lógica de funcionamiento del banco de pruebas desarrollado, la cual es mostrada en la figura 2.3. Esta secuencia inicia desde el proveedor de contenidos, el cual es el encargado de la producción y edición de contenidos multimedia bajo algún tipo de licencia. La siguiente interacción requiere la adaptación de dichos contenidos multimedia de acuerdo al estándar DASH, para después ser agregados al servicio de VoD, junto con otros servicios interactivos asociados al mismo. La siguiente etapa consiste en la entrega de los servicios y contenidos multimedia agregados en la anterior etapa. Finalmente la secuencia termina en el consumo de servicios y contenidos multimedia DASH. (Estimación del ancho de banda)

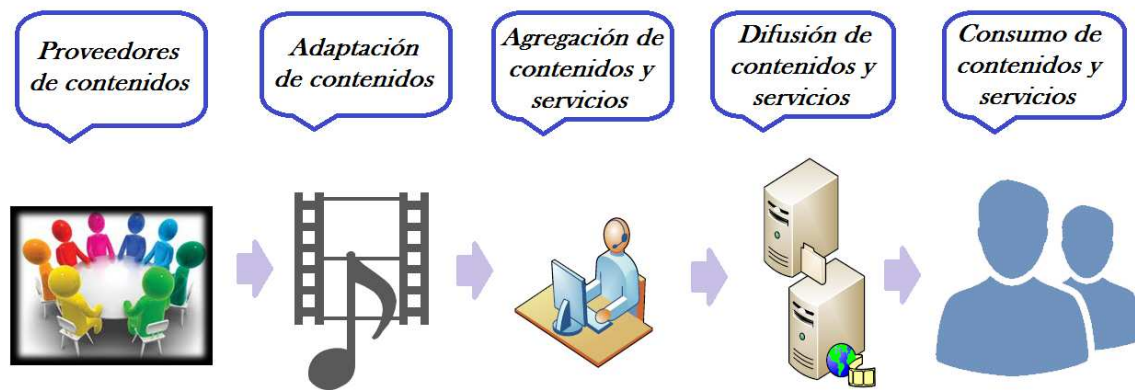


Figura 2.3. Modelo de negocio.

### 2.5.2 Vista de servicios

En esta vista fue desarrollado el servicio de VoD de contenidos multimedia adaptativos DASH, el cual está conformado por un catálogo de contenidos, un

reproductor de contenidos adaptados al estándar y los contenidos mismos. También fueron desarrollados servicios interactivos tales como un sistema de votación “*Star Rating*” para la valoración de contenidos y un servicio de chat tomado y adaptado del proyecto ST-CAV de la Universidad del Cauca. El esquema para el consumo de servicios interactivos está basado en el estilo arquitectónico REST-JSON.

A continuación son descritos los conceptos básicos del estilo REST-JSON y el esquema de consumo de servicios, utilizado para el despliegue de los mismos.

## **REST**

Este estilo arquitectónico está basado en la arquitectura cliente-servidor, en donde el intercambio de información puede hacerse a través de diferentes formatos o lenguajes como JSON, HTML y XML. Un servicio REST es visto como un recurso, el cual puede ser consumido por medio de una dirección URL mediante cuatro métodos del protocolo HTTP, es decir, GET POST, PUT, DELETE.

## **JSON**

Es un formato liviano y flexible para la comunicación entre cliente y servidor en servicios REST. Este tipo de formato está ampliamente difundido en diversos servicios de redes sociales y comunidades en internet, debido a su sencillez. La estructura de los mensajes JSON consiste en un conjunto de parejas nombre-valor, conocidas por los clientes y el servidor, donde el servidor está encargado de formar el mensaje y el cliente descompone la información contenida en el mensaje para después desplegarla en pantalla.

## **Esquema de consumo de servicios interactivos**

La figura 2.4 presenta el esquema de despliegue de servicios, en donde es utilizado el estilo arquitectónico REST-JSON. Los servicios interactivos implementados en este trabajo corresponden al sistema de valoración de contenidos multimedia DASH y chat. Cada servicio es accedido a través de una URL mediante peticiones HTTP y estos a su vez interactúan con el cliente solicitante mediante mensajes en formato JSON. La respuesta desde los servicios es almacenada y después desplegada a los usuarios. El servicio de chat fue implementado a partir del proyecto ST-CAV el cual hace uso del servidor de aplicaciones webpy de Python [48].

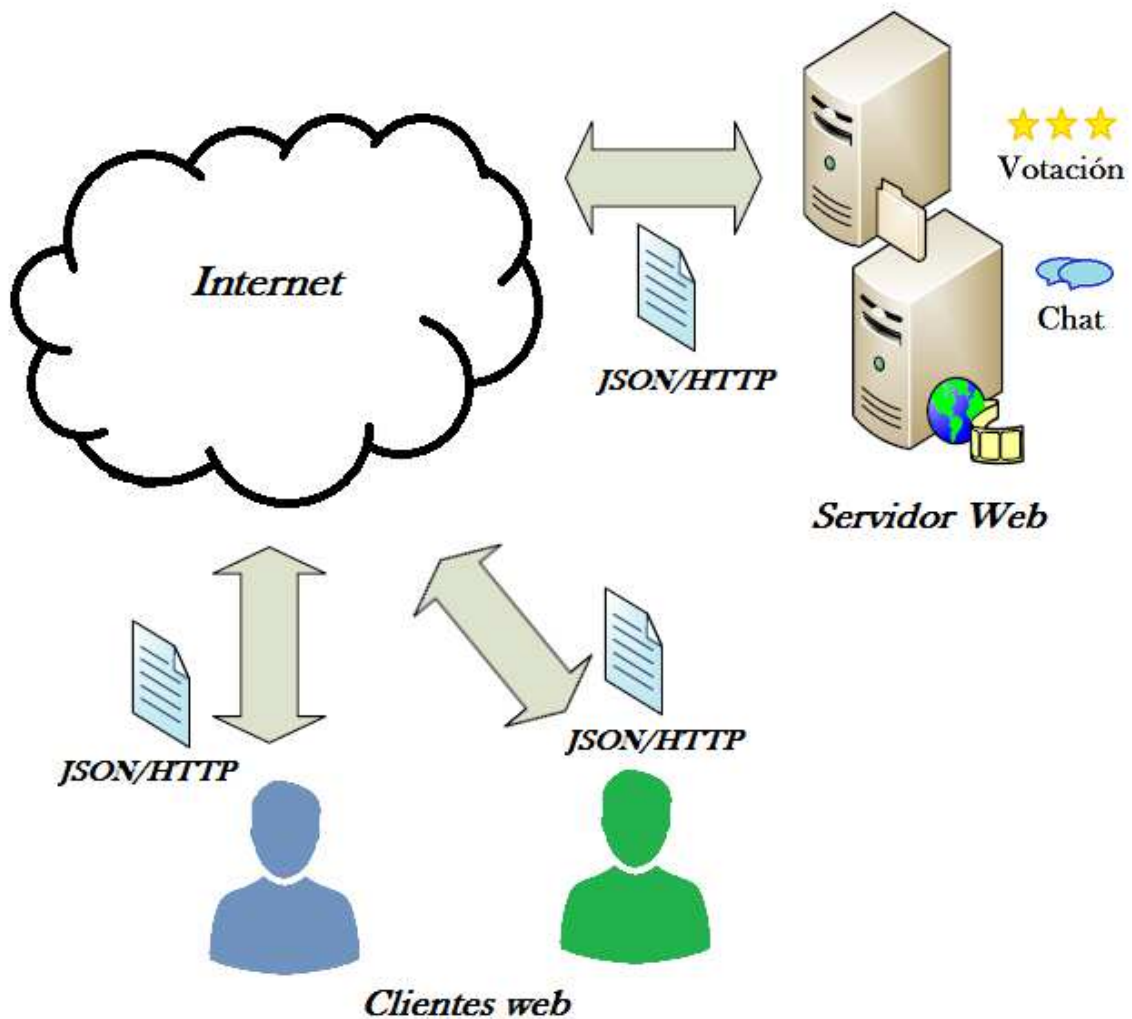


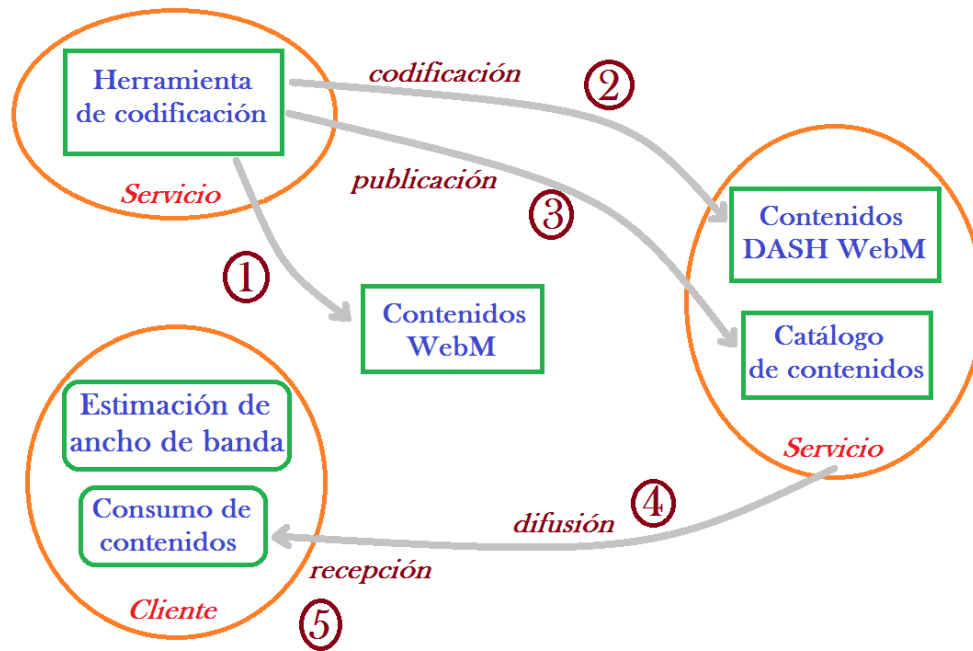
Figura 2.4. Esquema de consumo de servicios interactivos.

### 2.5.3 Vista de flujo de trabajo

Esta vista describe la lógica funcional del banco de pruebas planteado, para ello la figura 2.5 ilustra los pasos que conforman la secuencia del flujo de trabajo.

El proceso es iniciado a partir de la solicitud de un usuario tipo administrador, el cual accede a las funcionalidades de la herramienta de codificación automática desde un navegador web, la herramienta recibe archivos WebM desde el cliente o bien puede utilizar aquellos que ya están alojados en el servidor. La confirmación de continuar con el proceso de codificación hará que inmediatamente sean activados los pasos 2 y 3 de forma secuencial en el orden respectivo.





**Figura 2.5.** Flujo de trabajo.

Una vez iniciado el proceso de codificación, la herramienta adaptará el contenido WebM seleccionado al estándar DASH. Inmediatamente después de adaptado el contenido, es ubicado dentro de los recursos del servidor web, los cuales están disponibles para los clientes. Posteriormente el contenido multimedia DASH generado por la herramienta es añadido al catálogo de contenidos del servicio de VoD. Hasta este punto los procesos de codificación y publicación de contenidos multimedia WebM han sido desarrollados, para luego dar lugar al proceso de difusión de los mismos.

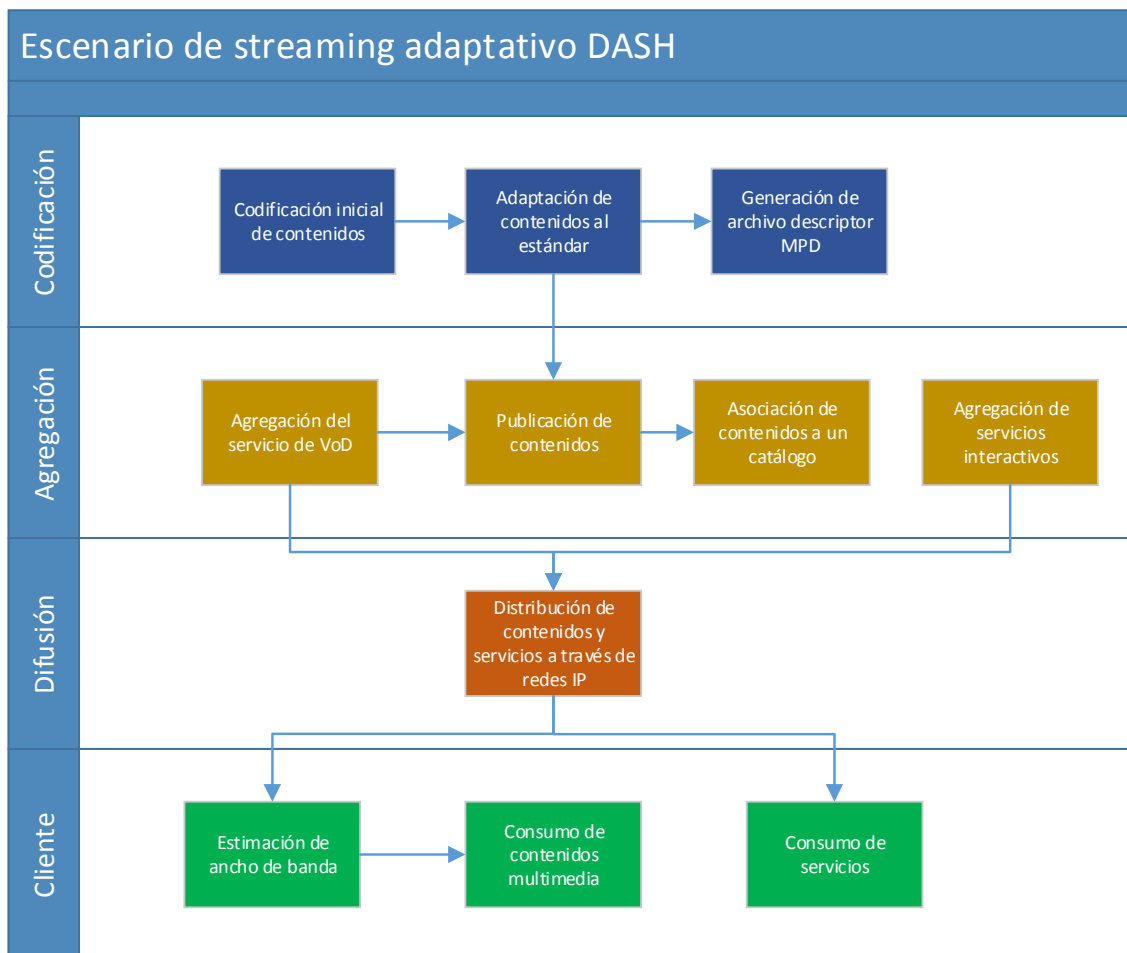
El cliente web es el encargado de la recepción y consumo de contenidos multimedia adaptados al estándar, mediante la función previa de estimación de ancho de banda de la red, desde donde el servicio de VoD es accedido.

### 2.5.4 Vista funcional general

Esta vista contiene información del esquema de etapas y funciones presentes en los módulos de codificación, agregación, difusión y recepción de contenidos multimedia DASH.

La figura 2.6, muestra el flujo que conforma el escenario de *streaming* adaptativo DASH, esta secuencia inicia desde las etapas de codificación inicial y adaptación de contenidos multimedia al estándar, generación del archivo descriptor MPD.

Dentro del módulo de agregación están, la conformación del servicio de VoD, la publicación de sus contenidos y asociación de los mismos al catálogo, también está la adición de servicios interactivos asociados al servicio de VoD.

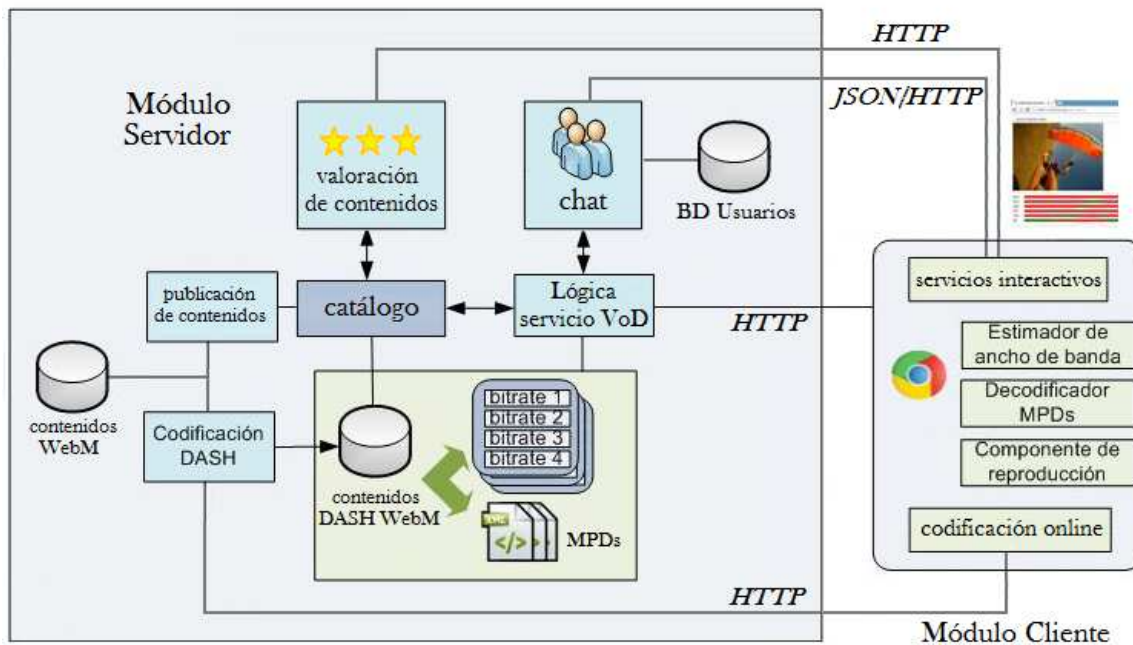


**Figura 2.6.** Diagrama modular de funcionamiento general del banco de pruebas.

La secuencia de etapas de los módulos difusión y cliente, contiene las funciones necesarias para el envío y consumo de los servicios implementados para el banco de pruebas.

A continuación en la figura 2.7 es presentada la arquitectura general de funcionamiento del banco de pruebas planteado en este trabajo, la cual está dividida en dos módulos básicos, módulo servidor y cliente.

El módulo cliente permite al usuario acceder a los servicios interactivos asociados al servicio de VoD, mediante el protocolo HTTP y el estilo arquitectónico REST-JSON. Este módulo contiene los elementos necesarios para el consumo de contenidos multimedia DASH: un estimador de ancho de banda, un decodificador de archivos MPD y un reproductor de contenidos multimedia. Adicionalmente está un componente administrativo que permite la codificación online por medio de la herramienta de codificación automática desarrollada en este trabajo de grado, la cual es descrita más adelante en el capítulo 3 de esta monografía.



**Figura 2.7.** Esquema general de banco de pruebas.

Por otra parte el módulo servidor está conformado por la lógica del servicio de VoD, la cual reúne los diferentes servicios interactivos asociados al servicio de VoD. En

la parte inferior de este módulo puede visualizarse el repositorio de contenidos multimedia DASH WebM, el cual tiene asociado un catálogo escrito en lenguaje XML. Este catálogo es actualizado constantemente desde el servicio de valoración de contenidos y desde la herramienta de codificación automática. Finalmente el servicio interactivo de chat complementa este conjunto de servicios.



## Capítulo 3

# Herramienta de codificación automática

### 3.1 Introducción

El proceso de generación de contenidos multimedia DASH requiere la ejecución secuencial de las tareas de codificación, segmentación y creación del archivo descriptor MPD. Para el caso del contenidos multimedia WebM, las anteriores tareas son realizadas por un conjunto de herramientas y/o librerías libres de manera separada, razón por la cual el proceso de generación del contenido multimedia adaptativo no es automático. Este capítulo describe la herramienta DASH WebM Converter, la cual permite automatizar el proceso de generación de contenidos multimedia WebM de acuerdo al formato del estándar DASH.

La herramienta desarrollada facilita la integración y ejecución secuencial de 3 tareas principales: codificación, segmentación y generación del archivo descriptor MPD. Estas tareas permiten la adecuación de archivos multimedia WebM al formato del estándar de *streaming* adaptativo DASH. Para realizar los procesos anteriores, la herramienta Dash WebM Converter ejecuta o invoca en segundo plano las librerías y/o herramientas: FFmpeg, libwebm y webm-tools, las cuales corren sobre el sistema operativo Linux.

Este capítulo contiene cuatro secciones principales, la primera sección describe las herramientas seleccionadas para la generación de contenido DASH en formato WebM, la segunda sección muestra paso a paso el proceso de codificación de contenidos multimedia adaptativos WebM seguido por la herramienta de codificación propuesta. La tercera sección presenta el diagrama funcional modular de la herramienta Dash WebM Converter; y en la sección final es descrita la

estructura del archivo de manifiesto MPD generado por la herramienta de codificación.

## 3.2 Herramientas seleccionadas para la generación de contenido multimedia DASH

Esta sección presenta los *scripts* y herramientas que han sido seleccionados para integrar la herramienta Dash WebM Converter, debido a su potencia para trabajar sobre contenido multimedia y sencillez en la forma de uso, además de pertenecer a proyectos de código libre.

### FFMPEG

Es una herramienta libre, la cual funciona mediante línea de comandos [36]. Su forma de instalación y manejo se indica en el anexo A y B de esta monografía. Este software posee múltiples funciones, sin embargo este trabajo la utiliza para la generación de flujos de video y audio, a diferentes tasas de bit a partir de un archivo WebM base.

Dentro del proceso de generación de contenido multimedia DASH, está herramienta hace la función de pre codificación, y los flujos multimedia generados son adaptados por el script `sample_muxer`, descrito a continuación. El correcto funcionamiento de esta herramienta requiere el uso de las librerías apropiadas para trabajar con archivos WebM, tales como, `libvpx` para video y `libvorbis` para audio.

### Sample\_muxer

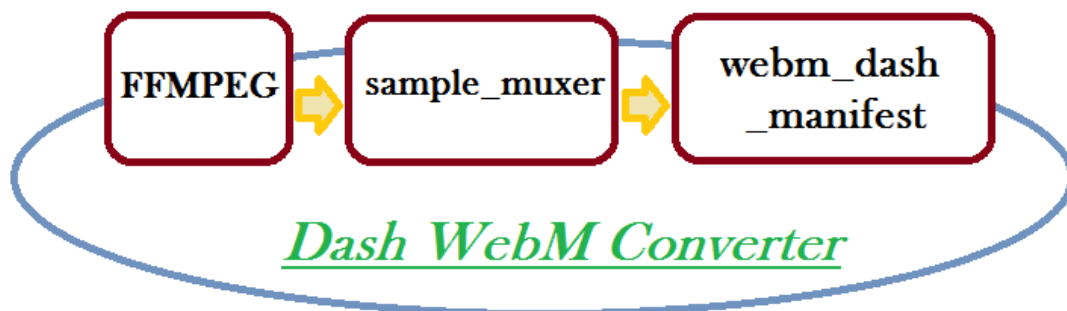
Es un script escrito en lenguaje C++, perteneciente a la biblioteca LibWebm. Funciona mediante línea de comandos, es utilizado para agregar índices de búsqueda a cada fotograma clave de un archivo WebM, con el fin de facilitar el canjeo entre cada *stream* de video al momento de reproducir el contenido multimedia adaptativo [45]. El modo de compilación de este script está descrito en el anexo A de esta monografía y su forma de ejecución en el anexo B de la misma. El tiempo de trabajo de este es de unos pocos segundos, dependiendo del tamaño

del stream de video o audio sobre el cual es aplicado. Su parámetro de entrada es el archivo WebM previamente adaptado por la herramienta FFMPEG y su salida es otro archivo WebM completamente adaptado al estándar DASH.

### **Webm\_dash\_manifest**

Es otro script escrito en lenguaje C++, incluido en la biblioteca Webm-tools del proyecto WebM de Google Inc. [45]. Su ejecución mediante línea de comandos permite la generación de un archivo descriptor MPD, los parámetros de entrada son cada uno de los flujos de video y audio que ya han sido adaptados al estándar DASH, cada flujo es identificado dentro del archivo MPD por un id diferente, además los *stream* de video y audio son descritos en dos agrupaciones diferentes. La compilación y forma de funcionamiento de este *script* es presentada en los anexos A y B, respectivamente. DASH.

Las herramientas seleccionadas funcionan en segundo plano y son ejecutadas desde la aplicación Dash WebM Converter, desarrollada en lenguaje Python. La secuencia de intervención de cada una de ellas es presentada en la figura 3.1, en donde la ejecución de la primera (FFMPEG) genera una secuencia en cadena de forma automática, que finaliza hasta que el contenido adaptativo es generado completamente.



**Figura 3.1.** Scripts utilizados en la herramienta de codificación automática.

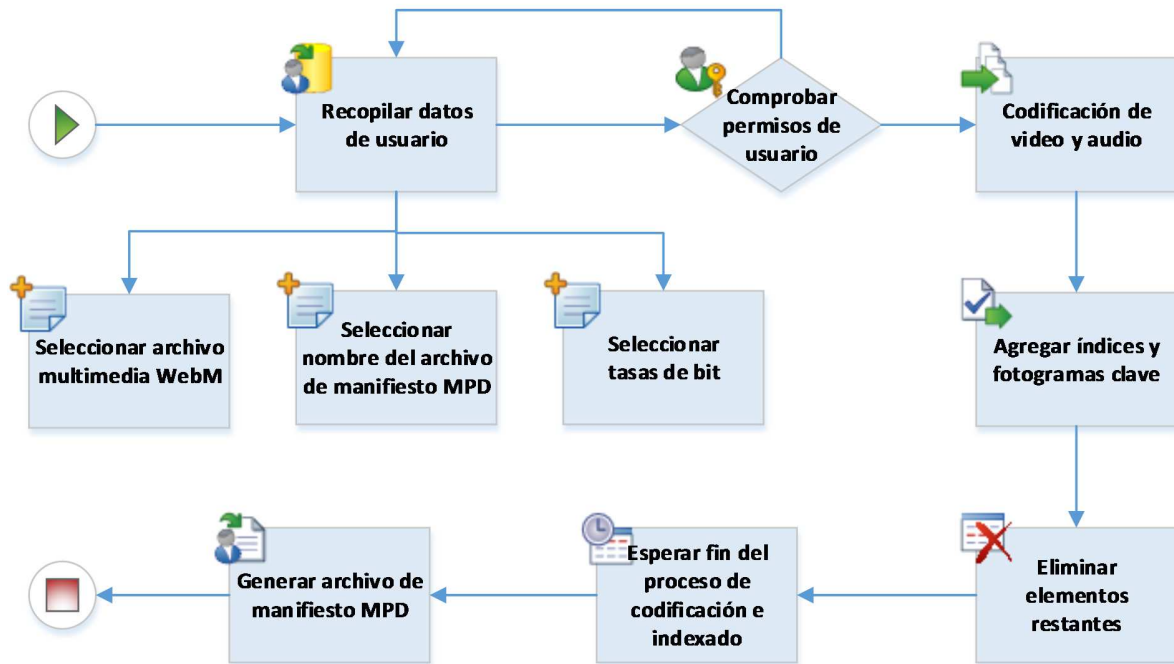
La versión de la herramienta de codificación automática en línea funciona como un servidor, el cual es accedido desde un cliente web por el administrador del contenido multimedia adaptativo.



### 3.3 Proceso de codificación de streaming adaptativo

La figura 3.2 presenta las diferentes fases del proceso de codificación de contenidos multimedia adaptativos WebM DASH, realizado por la herramienta propuesta.

Los archivos WebM para *streaming* adaptativo deben contar con flujos de audio y video multiplexados, con un índice de búsqueda asociado a cada fotograma clave del flujo de video, de tal forma que permita el cambio entre diversos flujos de *streaming* sin discontinuidad, a medida que varía el ancho de banda. A partir de lo anterior, una de las funciones de la herramienta de codificación es adaptar los contenidos WebM para ser transmitidos, ya que por defecto cada uno de estos archivos corresponde a un *stream*, sin fragmentación y sin puntos de sincronización.



**Figura 3.2.** Proceso de codificación de *streaming* adaptativo.

La primera fase del proceso de codificación consiste en recibir, desde la interfaz gráfica de la herramienta, tres datos básicos para este proceso, como son: la ruta del archivo multimedia WebM, las tasas de video a codificar, de acuerdo a la calidad del contenido multimedia y el nombre del documento de manifiesto MPD, ver la figura 3.3.



**Figura 3.3.** Interfaz Gráfica de la herramienta Dash WebM Converter.

Una vez brindados y comprobados los permisos del sistema operativo, inicia la fase de codificación, en la cual la herramienta Dash WebM Converter genera en paralelo tantos flujos de video WebM, como tasas de bits sean seleccionadas en la primera fase. De igual forma, un solo flujo de audio WebM por todos los flujos de videos codificados es generado. Para realizar lo anterior, la herramienta Dash WebM Converter hace invocaciones en segundo plano y en paralelo a la herramienta FFmpeg, la cual es ejecutada como un proceso del sistema operativo Linux.

Es importante mencionar que dentro de los parámetros usados para la invocación de FFmpeg, debe tenerse en cuenta el uso de la librería libvpx (códec VP8) para la codificación de video y de la librería libvorbis para la codificación de audio. Así mismo, debe indicarse a FFMPEG el intervalo de fotogramas máximo y mínimo entre cada fotograma clave. Dichos valores son utilizados para alinear todos los puntos de sincronización en las secuencias de video a canjear, ver la figura 3.4.



**Figura 3.4.** Fase de codificación de los contenidos WebM.

Después de que son generados los flujos de video WebM con diferentes tasas de bit y considerando que la herramienta FFmpeg no permite la alineación de los parámetros “Cluste” y “Cues” de los contenidos multimedia, la siguiente fase del proceso consiste en alinear los puntos de sincronización de las secuencias de vídeo y agregar los índices de búsqueda para cada fotograma clave dentro de los mismos flujos de video. Para realizar lo anterior, la herramienta Dash WebM Converter ejecuta en segundo plano el *script* `sample_muxer` de la librería `libwebm`. Los contenidos resultantes tras la ejecución del anterior proceso (nombre terminado en “\_sm”) son presentados en la figura 4, el resto de contenidos innecesarios de las fases anteriores son eliminados por la herramienta Dash WebM Converter.

Finalmente, la última fase consiste en la generación del archivo de manifiesto MPD, el cual contiene información de los flujos de video y audio disponibles para canjear al momento de ser solicitados por el cliente web de *streaming* DASH. Para generar el archivo MPD, la herramienta Dash WebM Converter ejecuta en segundo plano el *script* `webm_dash_manifest` de la librería `webm-tools`, enviándole como parámetro el listado de archivos asociados a los flujos de video y audio generados a partir de las tasas de bits asignadas en la fase 1.

El documento de manifiesto MPD generado, comprende características de los flujos de video y audio tales como: URL de los flujos, tipo de contenedor, *codecs* de audio y video, tasa de bits, resolución de pantalla del video, tasa de muestreo del audio, entre otros. La figura 3.5 muestra los archivos de *streaming* adaptativo generados, en conjunto con su archivo descriptor MPD.

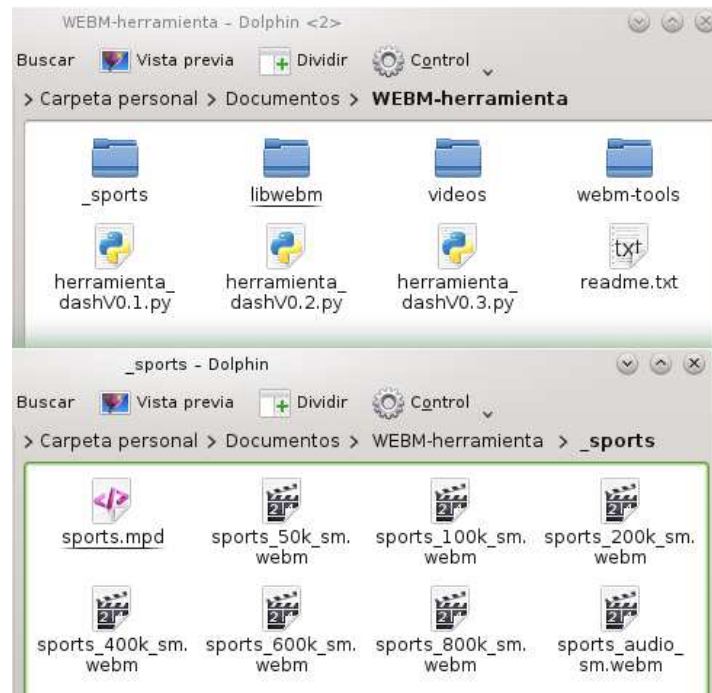
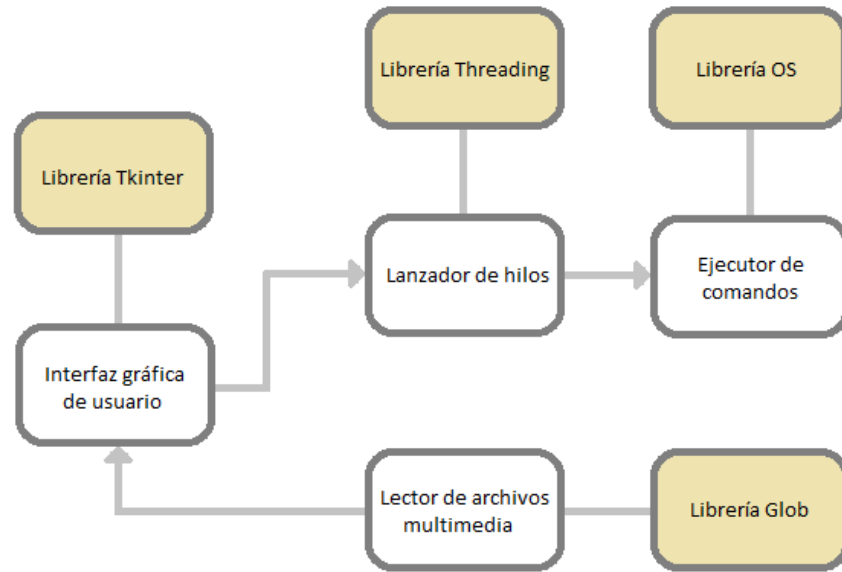


Figura 3.5. Recursos de la herramienta de codificación automática.

### 3.4 Diagrama modular de la herramienta DASH WebM Converter

Esta sección presenta el diagrama modular de la herramienta Dash WebM Converter, la cual fue desarrollada en el lenguaje Python. Tal como fue mencionado en la sección anterior, la función de la herramienta Dash WebM Converter es ejecutar de manera secuencial las tareas de codificación, segmentación y generación del archivo de manifiesto MPD para contenidos multimedia adaptativos DASH WebM.

El diagrama de la figura 3.6 distingue los siguientes los siguientes bloques funcionales principales: interfaz gráfico de usuario, lanzador de hilos, ejecutor de comandos, lector de archivos multimedia.



**Figura 3.6.** Diagrama de bloques herramienta de codificación automática.

El módulo de interfaz gráfica de usuario está conformado por la librería Tkinter de Python, la cual permite desplegar diferentes componentes gráficos como: campos de texto, etiquetas y botones, a través de los cuales es solicitado el archivo multimedia WebM base a codificar, el nombre del documento de manifiesto MPD y las tasas de bits a codificar. De manera simultánea en que es lanzada la herramienta Dash WebM Converter, puede visualizarse en la interfaz gráfica el contenido multimedia disponible para codificar, esta tarea es realizada por el modulo lector de archivos multimedia, a través de la librería Glob, ver la figura 3.6.

Por su parte el módulo Lanzador de Hilos está encargado de crear tantas instancias de hilos o procesos simultáneos, como tasas de bits seleccionadas desde la interfaz gráfica, más un hilo adicional para el contenido de audio. Para lo anterior, la herramienta Dash WebM Converter hace uso de la librería Threading de Python. Cada hilo de video hace una invocación en segundo plano de la herramienta FFmpeg, enviando como parámetro una tasa de codificación diferente, mientras que en el caso del hilo de audio, una tasa de bits estándar es elegida. Al terminar la tarea de codificación por parte de cada hilo lanzado, la herramienta Dash WebM Converter ejecuta el script `sample_muxer` de la librería `libwebm`, con el fin de agregar los índices de búsqueda para cada fotograma clave de cada flujo de video.

Finalmente, el módulo ejecutor de comandos, es el encargado de ejecutar comandos del sistema operativo Linux a través de la librería OS de python. Este módulo interactúa de manera directa con la consola del sistema operativo y permite lanzar en segundo plano y durante diferentes momentos del proceso de codificación, las herramientas y/o librerías: FFMPEG, y los scripts `sample_muxer` y `webm_dash_manifest`.

### 3.5 Archivo descriptor MPD

Esta sección presenta el esquema del documento de manifiesto MPD generado por la herramienta de codificación Dash WebM Converter en su fase final, ver las figuras 3.7 y 3.8. A modo de ilustración será considerado un documento MPD generado en una prueba de la herramienta planteada en este capítulo.

Tal como fue mencionado anteriormente, este documento descriptor es generado por la herramienta Dash WebM Converter a través de la ejecución en segundo plano del script `webm_dash_manifest` perteneciente a la librería `webm-tools`. El documento MPD está basado en el lenguaje de marcado extensible XML y contiene las siguientes configuraciones dentro de la etiqueta principal `<MPD>`:

- El parámetro “Perfil de presentación” está fijado por defecto en el perfil: “webm-on-demand”, indicando así el modo de transmisión del contenido multimedia, ver la figura 3.7.
- La etiqueta “Periodo” incluye el punto de inicio y la duración total del contenido multimedia, ver la figura 3.7.

```

9  <profiles="urn:webm:dash:profile:webm-on-demand:2012">
10 <Period id="0" start="PT0S" duration="PT264.16S" >
11   <AdaptationSet id="0" mimeType="video/webm" codecs="vp8" width="320" height="240"
12     subsegmentAlignment="true" subsegmentStartsWithSAP="1" bitstreamSwitching="true">
13     <Representation id="0" bandwidth="74285">
14       <BaseURL>sports_50k_sm.webm</BaseURL>
15       <SegmentBase indexRange="1925436-1926323">
16         <Initialization range="0-248" />
17       </SegmentBase>
18     </Representation>
19     <Representation id="1" bandwidth="119023">
20       <BaseURL>sports_100k_sm.webm</BaseURL>
21       <SegmentBase indexRange="3435010-3435898">
22         <Initialization range="0-248" />
23       </SegmentBase>
24     </Representation>

```

**Figura 3.7.** Documento de descripción MPD (cabecera).

- La etiqueta `<AdaptationSet>` con atributo `id=0`, describe la información de los flujos de video codificados a diferentes tasas de bits, especificando el formato de empaquetamiento, el códec de video y la resolución del contenido multimedia, ver la figura 3.7. Adicionalmente esta etiqueta incluye los atributos: `segmentAlignmentFlag = true` y `bitstreamSwitchingFlag = true`, los cuales al estar fijados en “true”, indican que es posible el intercambio de streams de video durante la reproducción del contenido multimedia, de acuerdo al ancho de banda disponible.
- La etiqueta `<AdaptationSet>` con atributo `id=1`, describe el único flujo de audio asociado los diferentes flujos de video codificados, indicando el formato de empaquetamiento, el códec de audio y la tasa de muestreo utilizados, ver la figura 3.8.
- La información más específica de cada uno de los flujos de video y audio codificados está contenida dentro de la etiqueta `<Representation>` con un `id` para cada *stream* individual, ver las figuras 3.7 y 3.8. En esta etiqueta hace parte del nodo de nivel superior `<AdaptationSet>`, y describe el ancho de banda asociado a cada flujo de audio o video y la URL de cada stream dentro del servidor de difusión.

```
50 <AdaptationSet id="1" mimeType="audio/webm" codecs="vorbis" audioSamplingRate="44100"  
51 subsegmentStartsWithSAP="1">  
52 <Representation id="6" bandwidth="119772">  
53 <BaseURL>sports_audio_sm.webm</BaseURL>  
54 <SegmentBase indexRange="3096600-3097489">  
55 <Initialization range="0-4520" />  
56 </SegmentBase>  
57 </Representation>  
58 </AdaptationSet>
```

**Figura 3.8.** Documento de descripción MPD (pie).





## Capítulo 4

# Servicio de video bajo demanda

### 4.1 Introducción

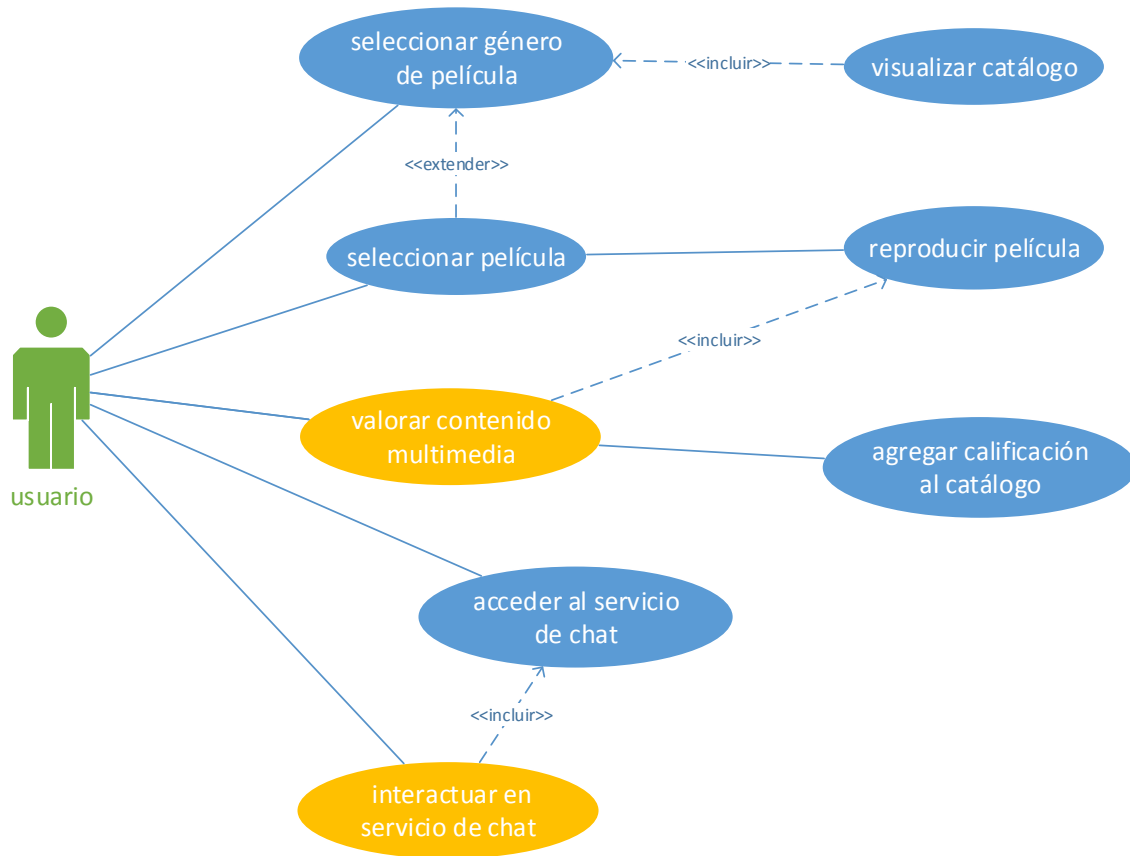
El servicio de VoD está definido como la entrega de contenido de vídeo a través de redes IP de banda ancha [5], en donde el usuario tiene el control de la sesión de video para reproducir, parar y avanzar los contenidos multimedia que actualmente esté accediendo. Entre los elementos más representativos de este servicio pueden nombrarse un componente de reproducción, un catálogo de contenidos, un sistema para la búsqueda, valoración y recomendación de contenidos, entre otros.

Los diferentes componentes interactivos y administrativos que han sido implementados en el banco de pruebas de este trabajo, son descritos a continuación en este capítulo. Primero el funcionamiento del servicio de VoD, de cada uno de sus componentes y los servicios interactivos asociados al mismo será explicado. Adicionalmente es presentado el funcionamiento del componente administrativo para la codificación automática online. Finalmente es presentada la disposición de los servicios desarrollados para los usuarios y el administrador en la nube, como también la enumeración de los scripts de control esenciales para el funcionamiento del banco de pruebas desarrollado.

## **4.2 Funcionamiento de los componentes interactivo y administrativo**

Esta sección describe los dos tipos de roles que intervienen dentro del banco de pruebas desarrollado, el rol de usuario y el rol de administrador. El rol de usuario es el encargado de consumir el servicio de VoD y los servicios interactivos asociados al mismo (valoración de contenidos y chat), los cuales fueron implementados dentro del banco de pruebas, mientras que el rol de administrador interviene en el proceso de codificación automática online de contenidos multimedia DASH.

La figura 4.1 presenta los casos de uso para el rol de usuario, quien puede intervenir directamente en los siguientes casos de uso: seleccionar género de película, seleccionar película, valorar contenido multimedia, acceder al servicio de chat e interactuar en servicio de chat. Los casos de uso realizados por el sistema son: visualizar catálogo, reproducir película y agregar calificación al catálogo. Cabe resaltar que los casos de uso en color amarillo corresponden a los servicios interactivos asociados al servicio de VoD.



**Figura 4.1.** Casos de uso para el rol de usuario.

A continuación son descritos los casos de uso más relevantes, para el rol de usuario.

<b>Caso de uso</b>	Seleccionar película
<b>Actores</b>	Usuario
<b>Propósito</b>	Permite reproducir un contenido multimedia DASH y activar el sistema de valoración.
<b>Resumen</b>	Una vez que el usuario haya seleccionado la película de su preferencia, el sistema iniciara la reproducción de la misma y además activara el sistema de valoración de contenidos multimedia.

**Tabla 4.1.** Descripción del caso de uso seleccionar película.

<b>Caso de uso</b>	Reproducir película
<b>Actores</b>	Sistema
<b>Propósito</b>	Permite activar el componente de reproducción de contenidos multimedia DASH.
<b>Resumen</b>	Una vez que es sistema activa el componente de reproducción, el usuario tiene la posibilidad de retroceder, adelantar, pausar y reanudar la reproducción de la película seleccionada desde el catálogo.

**Tabla 4.2.** Descripción del caso de uso reproducir película.

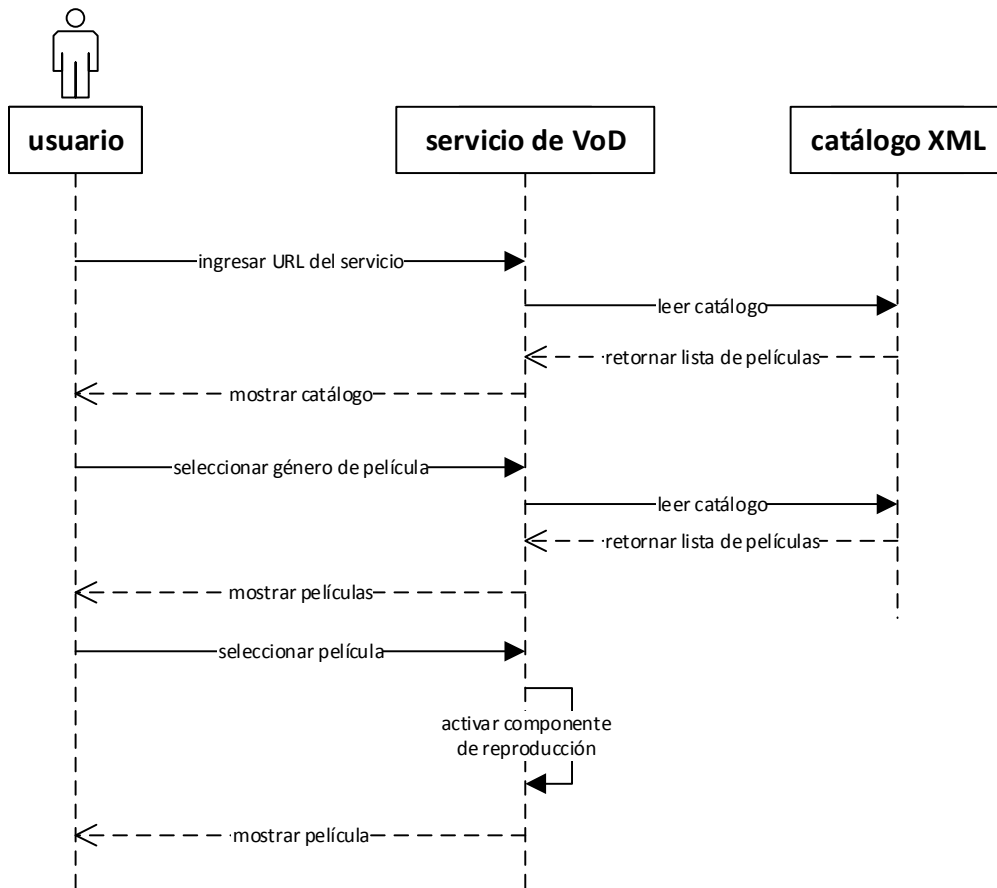
<b>Caso de uso</b>	Valorar contenido multimedia
<b>Actores</b>	Usuario
<b>Propósito</b>	Permite al usuario valorar la película que actualmente está observando.
<b>Resumen</b>	El sistema de valoración recibe del usuario la calificación del contenido, para luego agregarla al promedio general almacenado en el catálogo.

**Tabla 4.3.** Descripción del caso de uso valorar contenido multimedia.

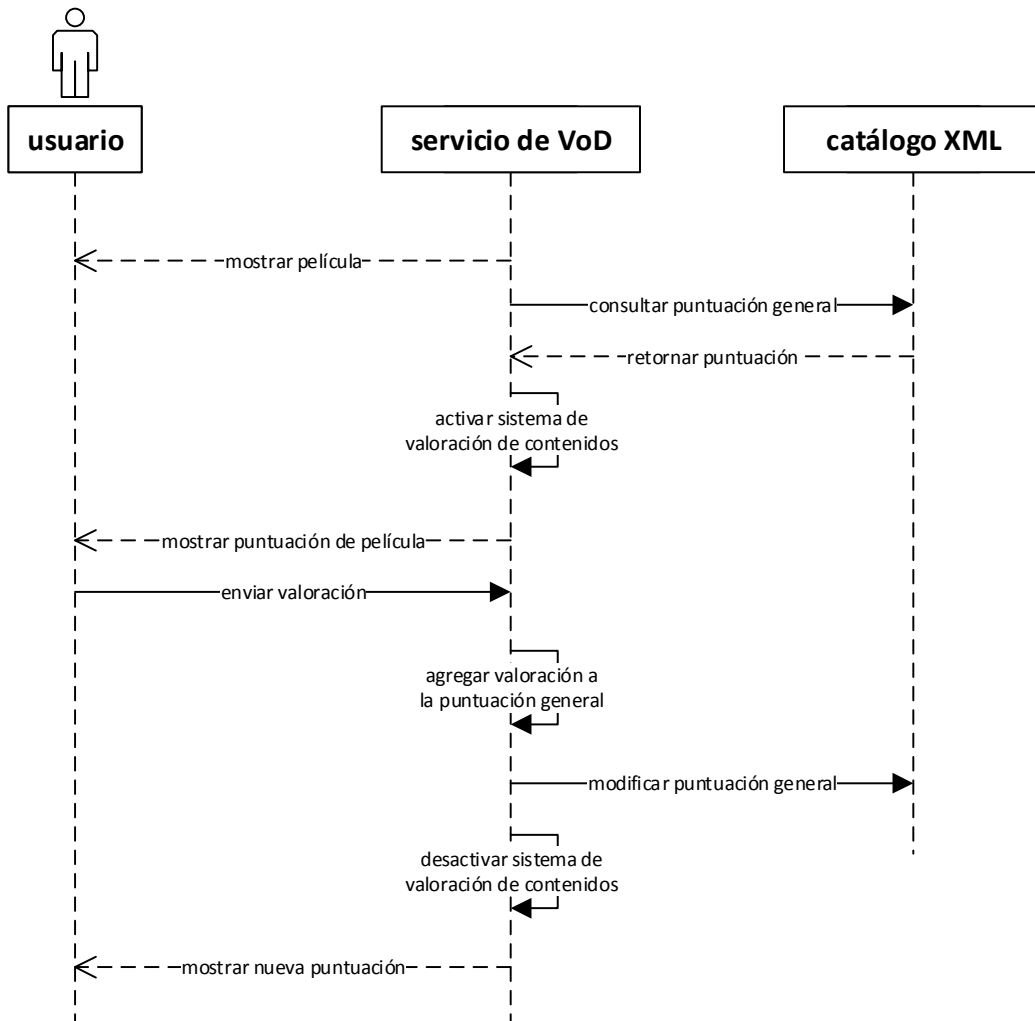
<b>Caso de uso</b>	Acceder al servicio de chat
<b>Actores</b>	Usuario
<b>Propósito</b>	Permite al usuario acceder a un determinado grupo de chat, en el cual este registrado.
<b>Resumen</b>	El usuario inicia el servicio interactivo de chat en un determinado grupo, una vez haya ingresado sus credenciales de autenticación, este puede interactuar con los demás usuarios que se encuentren activos en este grupo.

**Tabla 4.4.** Descripción del caso de uso acceder al servicio de chat.

La figura 4.2 presenta un diagrama de secuencia, para describir el funcionamiento del rol de usuario frente al servicio de VoD. Así mismo la figura 4.3 describe el funcionamiento del sistema de valoración de contenidos, por parte del rol de usuario.



**Figura 4.2.** Diagrama de secuencia del servicio de VoD.



**Figura 4.3.** Diagrama de secuencia del sistema de valoración de contenidos.

La figura 4.4 describe el funcionamiento del sistema de codificación automática online, mediante los casos de uso planteados. Inicialmente el sistema despliega al administrador un listado de los contenidos disponibles para ser codificados. Inmediatamente el actor administrador puede intervenir en el sistema mediante los casos de uso: subir contenido WebM, seleccionar contenido a codificar, ingresar datos de codificación y digitar información para el catálogo. Mientras que los casos de uso realizados por el sistema de color amarillo corresponden a: listar contenidos disponibles, codificar contenido y publicar contenido.

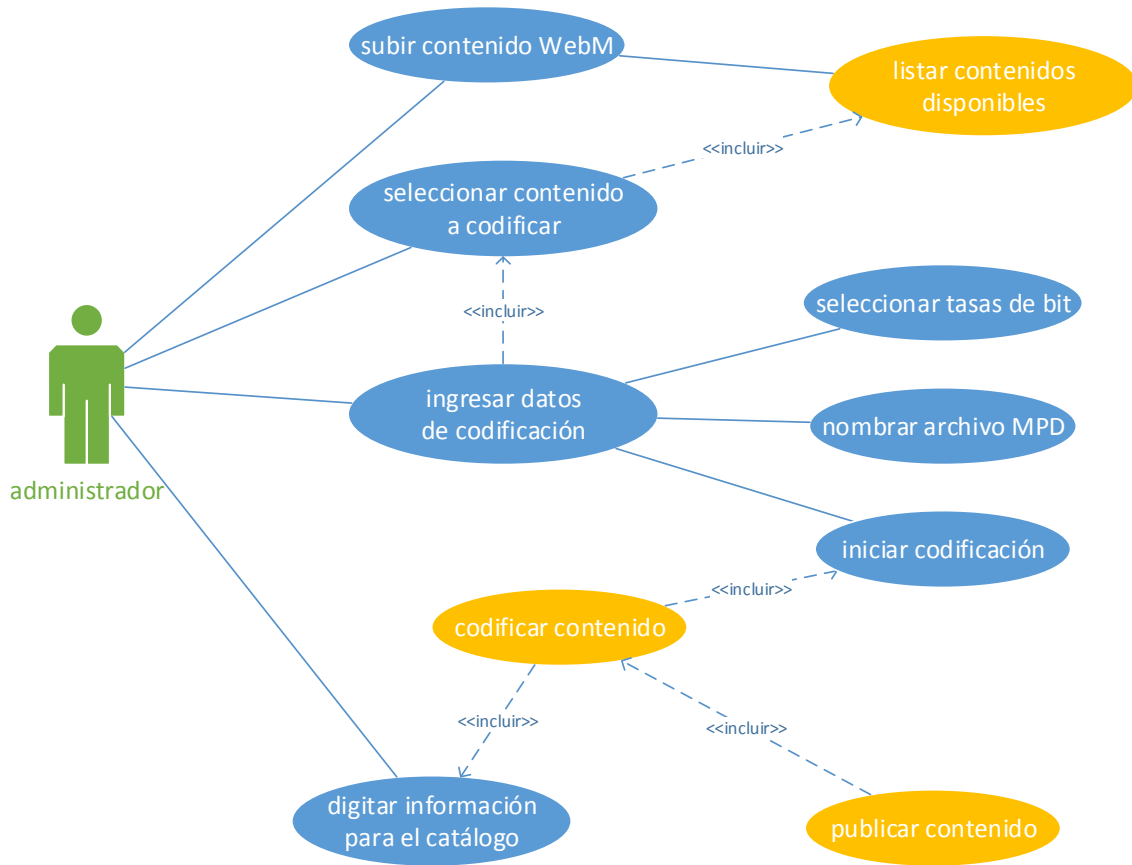


Figura 4.4. Casos de uso para el rol de administrador.

A continuación son descritos los casos de uso más relevantes, para el rol de administrador.

<b>Caso de uso</b>	Subir contenido WebM
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite subir un contenido WebM al servidor, para ser utilizado en el proceso de codificación automática.
<b>Resumen</b>	El administrador puede agregar contenidos WebM al servidor, los cuales aparecerán en el listado de contenidos multimedia disponibles que pueden ser codificados por la herramienta.

Tabla 4.5. Descripción del caso de uso subir contenido WebM.



<b>Caso de uso</b>	Ingresar datos de codificación
<b>Actores</b>	Administrador
<b>Propósito</b>	Permite insertar datos esenciales para realizar el proceso de codificación.
<b>Resumen</b>	Una vez que el administrador selecciona el contenido a codificar, este debe digitar datos relevantes para el proceso de codificación, tales como seleccionar tasas de bit y nombrar el archivo descriptor MPD, finalmente el administrador debe accionar el inicio del proceso de codificación.

**Tabla 4.6.** Descripción del caso de uso ingresar datos de codificación.

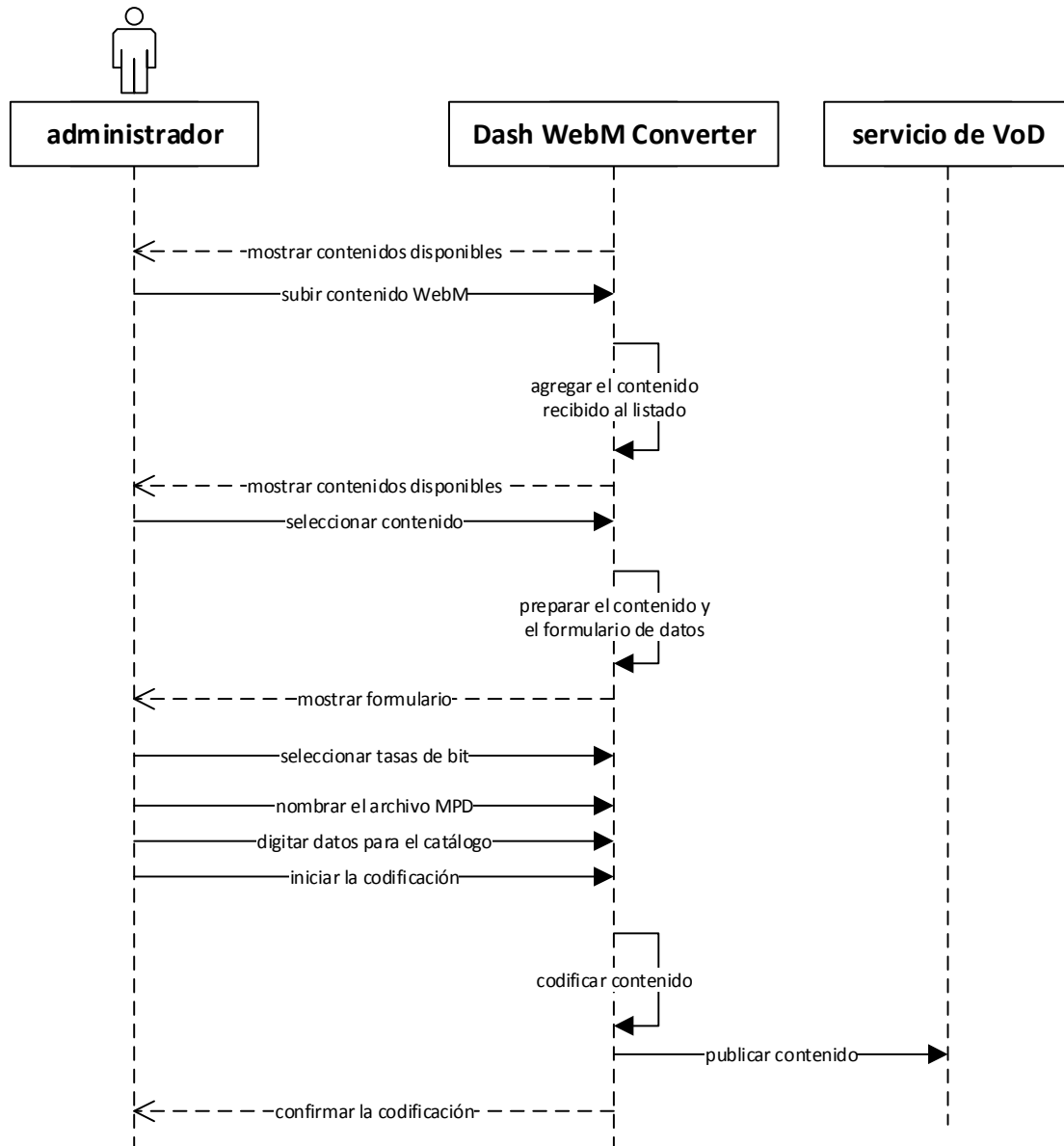
<b>Caso de uso</b>	Codificar contenido
<b>Actores</b>	Sistema
<b>Propósito</b>	Permite ejecutar el proceso de codificación automática.
<b>Resumen</b>	Después de que el administrador da inicio al proceso de codificación, el sistema responderá con un aviso indicando que el proceso ha iniciado y se ejecutará de acuerdo a los datos digitados previamente por el administrador.

**Tabla 4.7.** Descripción del caso de uso codificar contenido.

<b>Caso de uso</b>	Publicar contenido
<b>Actores</b>	Sistema
<b>Propósito</b>	Permite publicar el contenido codificado al servicio de VoD.
<b>Resumen</b>	Cuando el proceso de codificación automática haya finalizado, el sistema inmediatamente permitirá que el contenido esté disponible en el servicio de VoD.

**Tabla 4.8.** Descripción del caso de uso publicar contenido.

Para comprender mejor el funcionamiento del sistema de codificación online, la figura 4.5 muestra la secuencia de interacción del rol administrador sobre el sistema.



**Figura 4.5.** Diagrama de secuencia del sistema de codificación automática online.

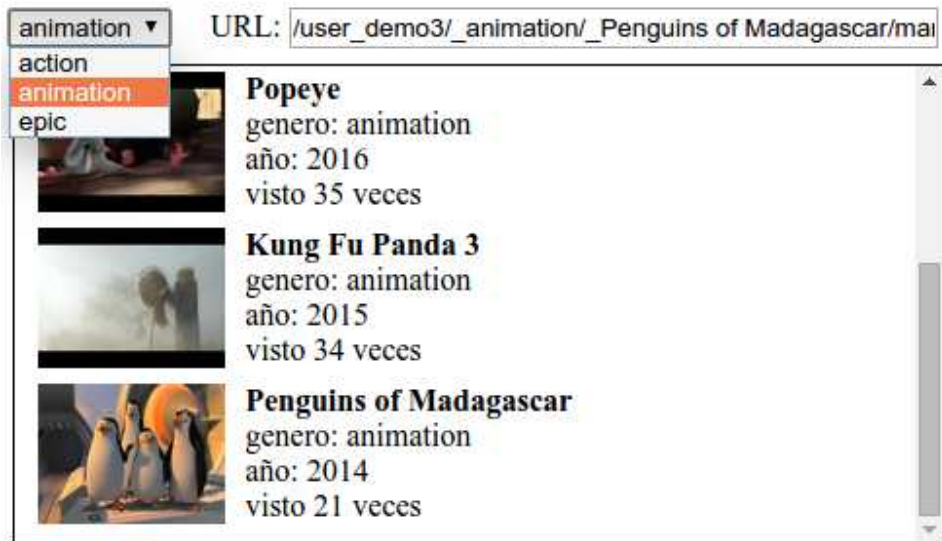
### 4.3 Funcionamiento del servicio de VoD

Para iniciar el servicio de VoD, primero un usuario debe ingresar a la URL del mismo, una vez iniciado este publicará en la parte derecha del navegador un catálogo de contenidos multimedia adaptativos, ver la figura 4.6. En la parte izquierda de la pantalla estará disponible un reproductor de contenidos multimedia DASH, la parte intermedia de la pantalla despliega un sistema de votación para la valoración del contenido multimedia que actualmente está consumiendo. Finalmente la parte inferior del navegador presenta un sistema para visualizar la variación del ancho de banda actual de la red.

The screenshot displays the VoD service interface. On the left, a video player shows a scene from 'Penguins of Madagascar' featuring a wolf character. Below the player is a voting system with five stars and the text 'puntuación: ★★★★★ votar!'. On the right, a media catalog is visible with tabs for 'Media', 'Chat', and 'Coding Tool'. The 'Media' tab is active, showing a list of animation titles: 'Popeye' (2016, 35 views), 'Kung Fu Panda 3' (2015, 34 views), and 'Penguins of Madagascar' (2014, 21 views). At the bottom, a 'Hide Chunk Graph' button is present.

**Figura 4.6.** Interfaz gráfica del servicio de VoD.

La figura 4.7 presenta la vista del catálogo de contenidos multimedia DASH, del servicio de VoD. En la parte superior izquierda el usuario puede seleccionar el género de películas de su preferencia, una vez que el usuario haya elegido el género deseado, automáticamente la parte inferior visualizará un listado de las películas disponibles. El listado de películas está ordenado en forma descendente, de acuerdo al número de visitas o reproducciones a dichos contenidos. La parte superior derecha de la figura, muestra la ubicación del archivo descriptor MPD de la película que actualmente se está reproduciendo.



**Figura 4.7.** Catálogo de contenidos multimedia para el servicio de VoD.

El componente de reproducción del servicio de VoD permite al usuario: retroceder, adelantar, pausar y reanudar el contenido que actualmente está consumiendo, además el usuario puede ajustar el nivel de volumen y ampliar el componente de reproducción a pantalla completa. Ver la figura 4.8.



**Figura 4.8.** Componente de reproducción del servicio de VoD.

El sistema de votación implementado permite al usuario valorar el contenido que actualmente está consumiendo. La figura 4.9 despliega la vista grafica del servicio de valoración tipo *Star Rating*. La marcación en color amarillo denota la calificación promedio del contenido, mientras que la de color rojo es la valoración seleccionada por el usuario, asimismo la parte inferior derecha despliega al usuario el valor numérico de la calificación que ha escogido. Una vez enviada la puntuación esta será adicionada a la calificación promedio del catálogo.

### Penguins of Madagascar



puntuación:



6.5

ENVIAR

**Figura 4.9.** Sistema de valoración de contenidos multimedia.

El servicio interactivo de chat es iniciado desde la pestaña *Chat*, ver la figura 4.10. Este servicio cuenta con las funcionalidades básicas de envío y recepción de mensajes en tiempo real, además cuenta con un listado de emoticones que el usuario podrá pintar para expresar algún estado o sensación.



**Figura 4.10.** Servicio interactivo de chat.

Inicialmente el servicio de chat mostrará al usuario un recuadro como el de la figura 4.11, para que este sea identificado dentro de la comunidad a la cual pertenece, y así pueda empezar a consumir este servicio. Ocasionalmente el usuario podrá interactuar con el catálogo de contenidos ingresando a la pestaña Media al mismo tiempo que recibe mensajes desde el servicio de chat.

**CHAT SERVICE**

Id Usuario:

Id comunidad:

**Figura 4.11.** Autenticación para acceder al servicio de chat.

## 4.4 Herramienta de codificación automática online

Durante el desarrollo del banco de pruebas de este trabajo, fue integrado un sistema de codificación automática online al servicio de VoD, este sistema es desplegado desde la pestaña *Coding Tool* como es mostrado en la figura 4.12. Primero el sistema presenta una ventana con un listado de contenidos WebM, dispuestos para ser codificados, opcionalmente el administrador puede elegir un archivo multimedia WebM y enviarlo al servidor para agregarlo a la lista de contenidos disponibles.



**Figura 4.12.** Ventana inicial de la herramienta de codificación automática online.

Una vez que el administrador haya elegido un archivo del listado desplegado, el sistema muestra una siguiente ventana, como la de la figura 4.13, en donde debe digitar toda la información de codificación (tasas de bit y nombre del archivo descriptor MPD), además de la información para agregar al catálogo del servicio de VoD (género de película, actores principales, año). Finalmente el administrador debe confirmar para iniciar el proceso de codificación.

The screenshot shows the 'Coding Tool' tab of the Dash WebM Converter. At the top, there are three tabs: 'Media', 'Chat', and 'Coding Tool'. Below the tabs, the title 'Dash WebM Converter' is centered. The interface includes a file selection area with buttons for 'Seleccionar archivo', 'No se eligió archivo', and 'Upload WebM video'. The form contains several input fields: 'Media content' (The Prince.webm), 'Bitrates' (100, 200), 'Manifest' (manifest.mpd), 'Genre' (epic), 'Actors' (prueba de codificación en línea), and 'Year' (2015). To the right of the form, there are five radio button options for bitrates: 100 kbps (checked), 200 kbps (checked), 400 kbps, 600 kbps, and 800 kbps. A 'Save and Codify' button is located at the bottom of the form.

**Figura 4.13.** Formulario de la herramienta de codificación automática online.

Después de confirmar el proceso de codificación, la herramienta permite al administrador realizar nuevas codificaciones con otros archivos, mientras que en la parte superior de la pantalla son visualizados los contenidos sobre los cuales está trabajando, ver la figura 4.14.

The screenshot shows the 'Dash WebM Converter' interface after processing. At the top, there are three tabs: 'Media', 'Chat', and 'Coding Tool'. Below the tabs, the title 'Dash WebM Converter' is centered. The interface includes a file selection area with buttons for 'Seleccionar archivo', 'No se eligió archivo', and 'Upload WebM video'. Below this, the text 'Coding Tool is working now:' is displayed, followed by 'The Prince.webm'. A horizontal line separates this from a list of files. The first file in the list is 'Penguins of Madagascar.webm', which is accompanied by a circular progress indicator.

**Figura 4.14.** Multitarea de la herramienta de codificación automática online.

Paralelamente puede observarse en el servicio de VoD, el contenido que actualmente está agregándose al mismo. La figura 4.15 muestra el catálogo y en la



parte final puede observarse el contenido publicado con un mensaje indicando que aún está siendo codificado.

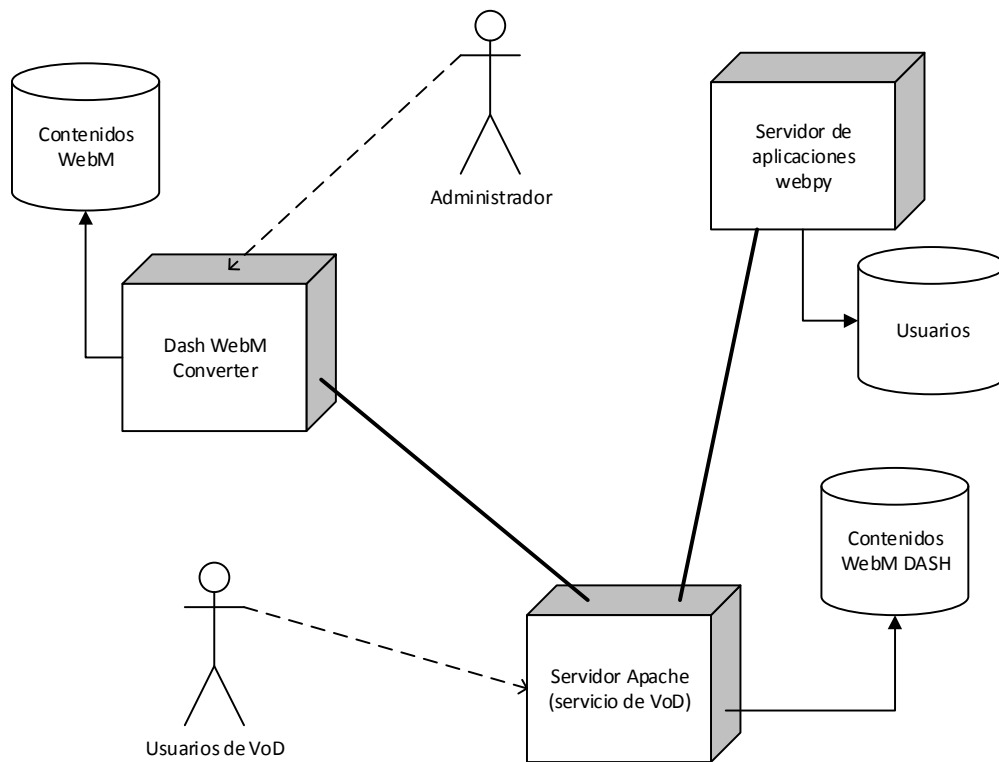


**Figura 4.15.** Contenido multimedia publicado en el servicio de VoD.

## 4.5 Disposición de los servicios

La gráfica de la figura 4.16 resume la forma en que están dispuestos los diferentes servicios y funcionalidades para los usuarios y el administrador. Los usuarios tienen acceso al servicio de VoD alojado en el servidor Apache, el cual tiene acceso al servidor de aplicaciones destinado para el servicio interactivo de chat, mientras que para el administrador es de su interés la herramienta de codificación automática.

El servidor de aplicaciones es una adaptación del servidor del proyecto ST-CAV de la Universidad del Cauca, el cual fue desarrollado utilizando el script webpy del lenguaje Python y posee diferentes servicios tales como: chat, tablón, votación, notificaciones, entre otros.



**Figura 4.16.** Organización de los servicios en el banco de pruebas.

La distribución de los módulos y servicios más sobresalientes para el funcionamiento del banco de pruebas son enumerados en la figura 4.17, donde son destacados un paquete principal de lógica del servicio de VoD, el componente de reproducción para contenidos multimedia DASH en formato WebM, los servicios de votación y chat, adicionalmente la herramienta de codificación automática. Cada paquete de la gráfica, contiene un conjunto de scripts en diferentes lenguajes de programación (Python, PHP, JavaScript), formando un componente funcional dentro del banco de pruebas.

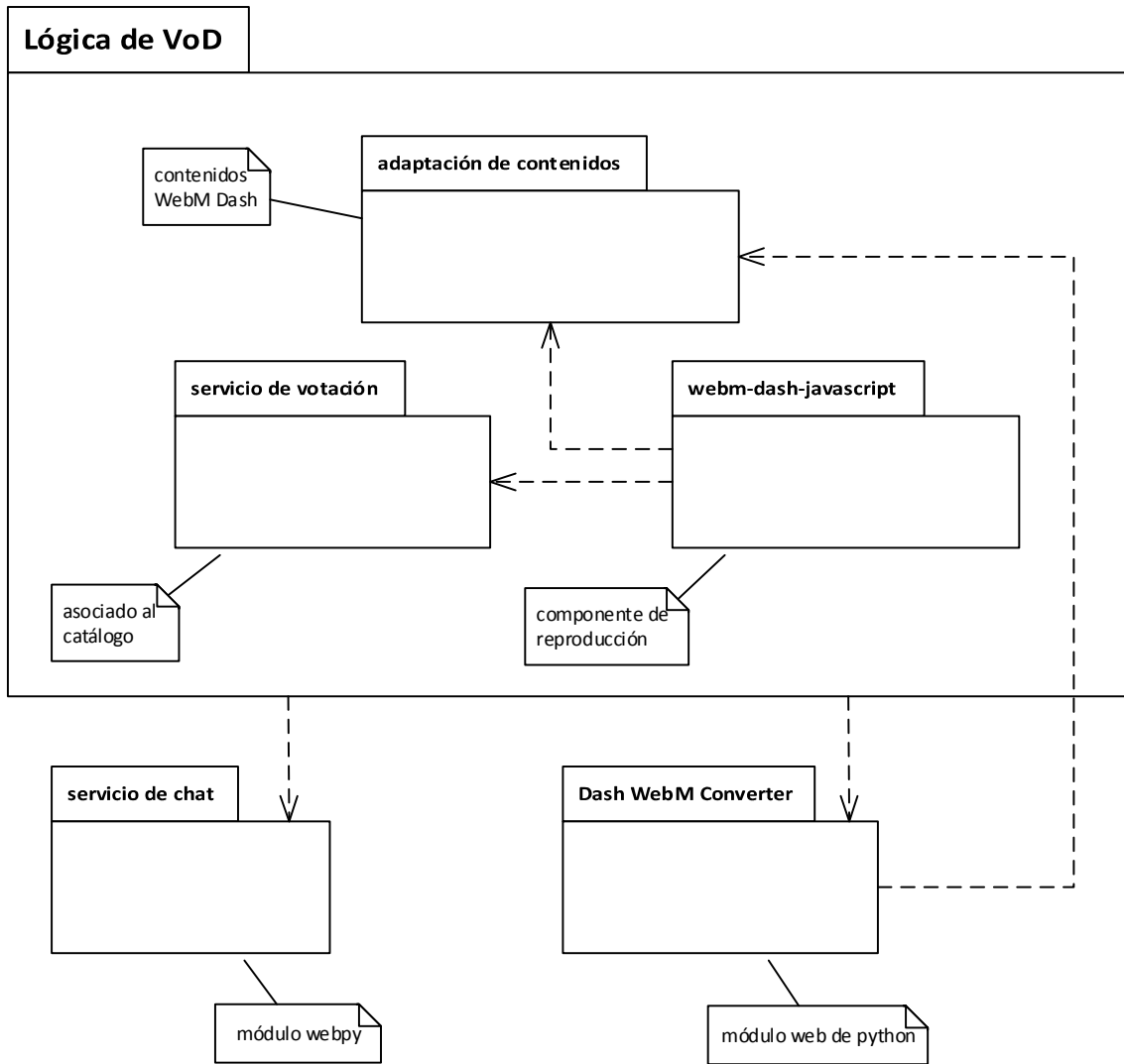


Figura 4.17. Diagrama de paquetes del banco de pruebas.

# Capítulo 5

## Pruebas

### 5.1 Introducción

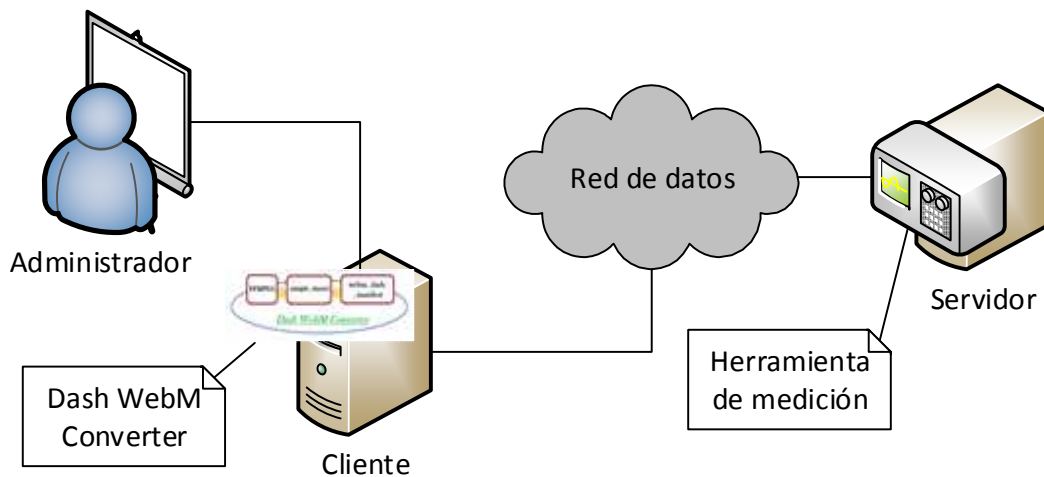
Durante el desarrollo de este trabajo de grado se consiguió implementar una herramienta para realizar el proceso de codificación automática y otra para la medición de consumo de memoria RAM y CPU de la primera, también fue desarrollado un servicio de VoD de contenidos multimedia adaptativos DASH y además fueron implementados los servicios interactivos asociados al anterior.

De acuerdo a lo anterior, en las siguientes secciones serán desarrolladas las siguientes pruebas: prueba de medición de memoria, la cual es utilizada para verificar el uso de memoria RAM y CPU utilizado por la herramienta de codificación automática y cada uno de sus hilos de trabajo que la conforman, después de esto será abordada la prueba de ancho de banda, que consiste en verificar el funcionamiento del servicio de VoD de contenidos multimedia DASH implementado en este trabajo, finalmente es presentada la sección de prueba de estrés, esta permite verificar la capacidad de estrés soportado por el servicio interactivo de chat desarrollado.

## 5.2 Prueba de consumo de memoria

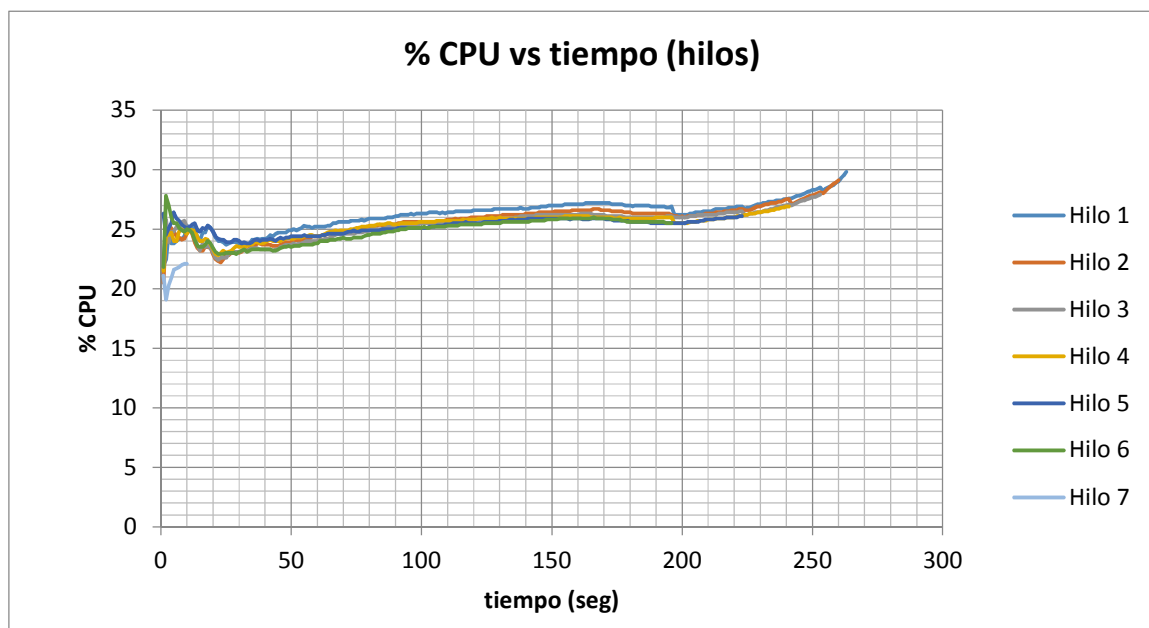
Esta sección presenta las pruebas de consumo de memoria RAM y porcentaje de uso de CPU, realizadas tanto a la herramienta de codificación propuesta, como a la herramienta FFMPEG usada en segundo plano. Para la captura de los datos presentados en esta sección, fue desarrollada una herramienta de monitorización en el lenguaje Python, la cual permite obtener cada 5 segundos, los valores de memoria y porcentaje de cpu de los procesos del sistema operativo asociados a la herramienta de codificación y la herramienta FFMPEG. La obtención de estos valores los realiza la herramienta, mediante el comando "ps aux" de Linux, el cual brinda un reporte de la cantidad de memoria RAM y el porcentaje de CPU utilizados por cada uno de los procesos activos del sistema operativo. Así mismo, usa el comando grep y el lenguaje de programación awk (pertenecientes al sistema operativo Linux), con el fin de filtrar la información de los procesos.

La figura 5.1 muestra el entorno de prueba utilizado para realizar la medición de consumo de memoria RAM y porcentaje de CPU, usados por la herramienta de codificación automática.



**Figura 5.1.** Entorno de prueba para evaluar Dash WebM Converter.

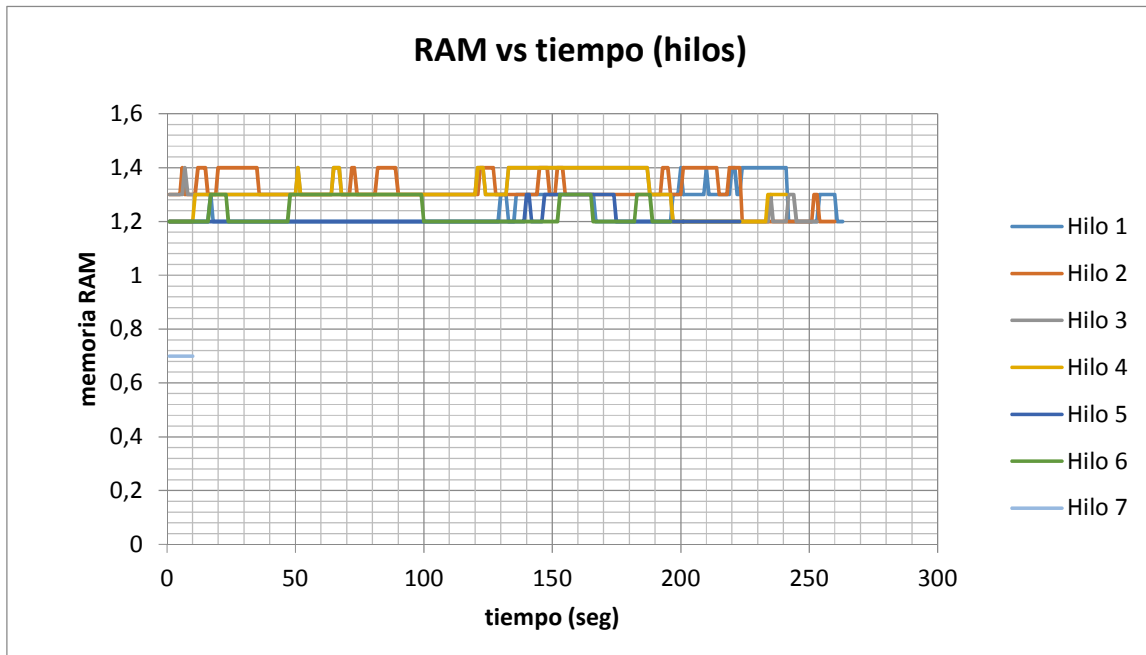
Las figuras 5.2 y 5.3 muestran los resultados de las pruebas de porcentaje de uso de CPU y consumo de memoria RAM, realizadas sobre la herramienta FFMPEG (usada en segundo plano por la herramienta propuesta en este artículo), durante la codificación de un contenido multimedia WebM a 6 tasas de bits diferentes (50,100, 200,400,600 y 800 kbps). Dado que la herramienta de codificación lanza tantos hilos de FFMPEG como tasas de bits a codificar, más un hilo adicional para el contenido de audio, las figuras 5.2 y 5.3 presentan el comportamiento de los 7 hilos (incluyendo el de audio) usados en una prueba de funcionamiento de la herramienta de codificación.



**Figura 5.2.** Porcentaje de CPU vs tiempo (FFMPEG).

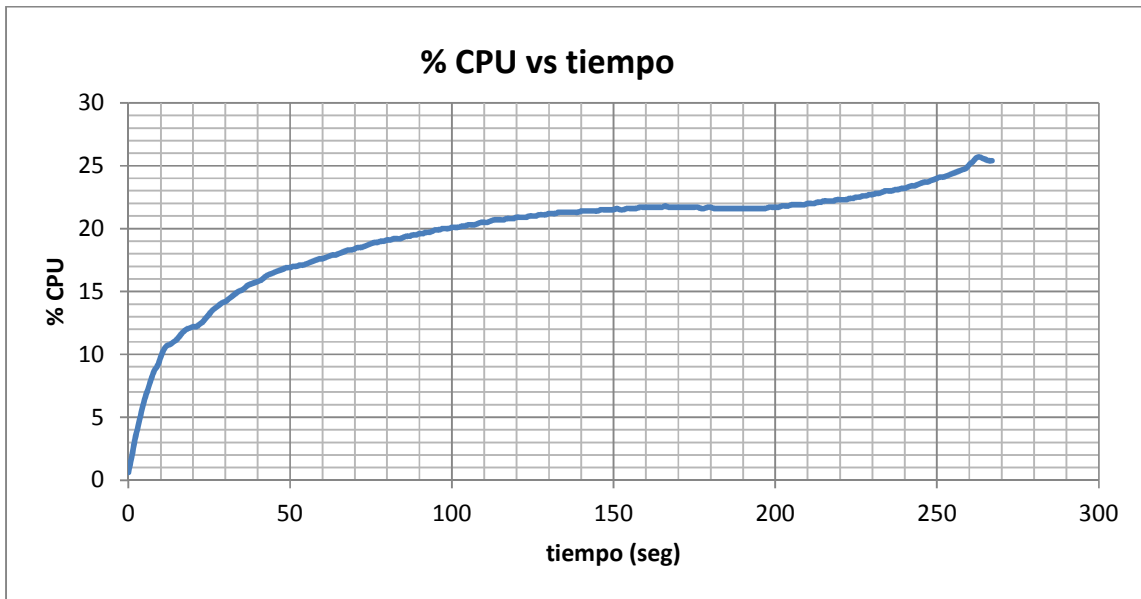
Para la realización de esta prueba fue utilizado un archivo en formato WebM de 51 MB de tamaño y 264 segundos de duración. De acuerdo a la figura 5.2, el hilo de audio (Hilo 7) es el primero en terminar su ejecución con un tiempo cercano a los 10 segundos y es el que menos porcentaje de CPU ocupa, con valores entre 19% y 22%. Así mismo, cabe resaltar que los hilos destinados a la codificación de video emplean un porcentaje de CPU similar durante todo el proceso, el cual oscila entre 22% y 30%.

De acuerdo a la Figura 5.3, el hilo de audio (Hilo 7) es el primero que termina su ejecución y es el que menos cantidad de memoria RAM consume, con un valor de 0.7 MB. Entre tanto los 6 hilos de video tienen un consumo similar de memoria que oscila entre 1.2 MB y 1.4 MB durante el proceso de codificación. Teniendo en cuenta lo anterior, en promedio el consumo de cada hilo de video en la herramienta FFMPEG es de 1,2 MB.

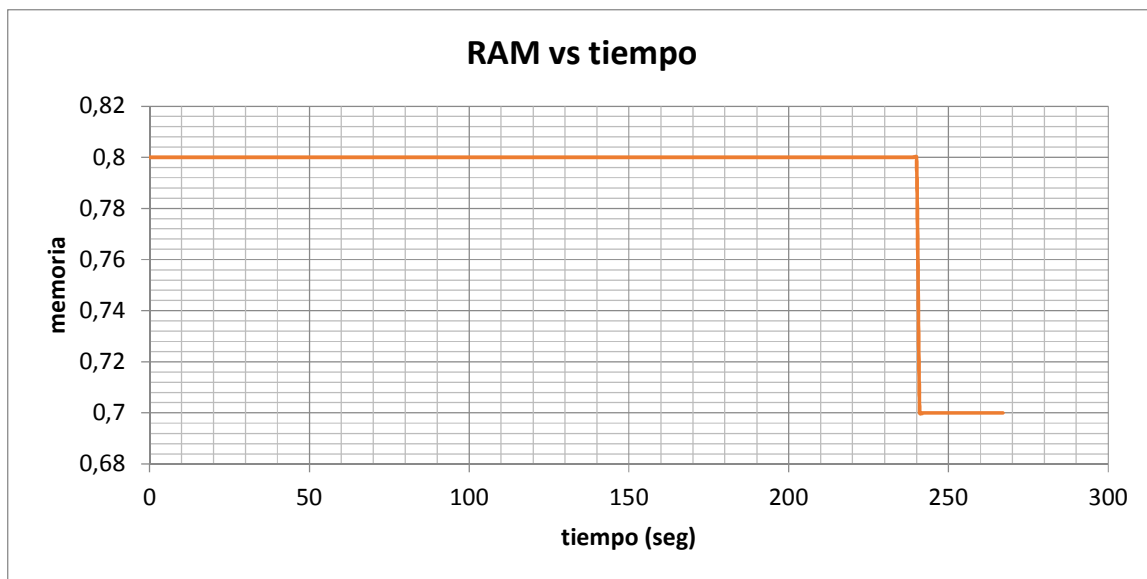


**Figura 5.3.** Memoria RAM vs tiempo (FFMPEG).

Las figuras 5.4 y 5.5 muestran los resultados de las pruebas de porcentaje de uso de CPU y consumo de memoria RAM, realizadas sobre la herramienta propuesta en este trabajo, durante la codificación de un contenido multimedia WebM a 6 tasas de bits diferentes. De acuerdo a la figura 5.4, el porcentaje de uso de CPU de la herramienta de codificación presenta un comportamiento creciente entre los 0 y 100 segundos, alcanzando un valor de 20 %. Por su parte entre los 100 y 200 segundos, el porcentaje de uso de CPU está alrededor del 21 %. Finalmente, después de los 200 segundos, el porcentaje de CPU tiene un comportamiento creciente alcanzando un valor máximo de 26%.



**Figura 5.4.** Porcentaje de CPU vs tiempo.



**Figura 5.5.** RAM vs tiempo.

De acuerdo a la figura 5.5, el consumo de memoria de la herramienta de codificación es casi constante hasta los 240 segundos, con un valor de 0.8 MB. Después de los 240 segundos, el valor de consumo de memoria disminuye a 0.7 MB hasta el final del proceso de codificación.



De acuerdo a las mediciones de consumo de memoria RAM realizadas sobre la herramienta FFmpeg, para un contenido multimedia con 6 tasas de bits a codificar, la cantidad de memoria RAM asignada a cada hilo de video es en promedio de 0.2 MB. Lo anterior permite inferir que el hecho de usar procesamiento multihilo no representa un incremento considerable en la cantidad de memoria RAM, con respecto a las ventajas obtenidas de optimización de tiempo.

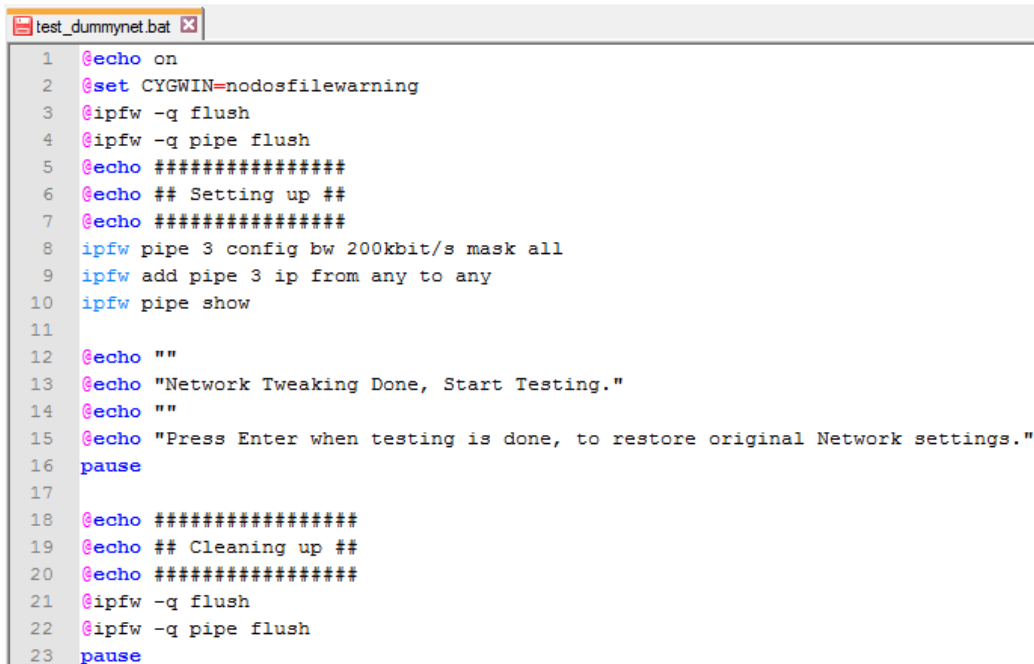
De acuerdo a los resultados obtenidos en esta prueba, sobre la cantidad de memoria RAM y porcentaje de CPU consumido por la herramienta de codificación automática desarrollada, puede considerarse que esta funciona de manera adecuada, dentro de este entorno de pruebas.

### 5.3 Prueba de ancho de banda

Para la realización de este tipo de prueba fue utilizada la herramienta Dummynet, la cual consiste de un emulador de red en vivo. Entre las funciones principales están la emulación de ancho de banda, pérdida de paquetes, retardo, entre otras funcionalidades. Esta herramienta funciona sobre sistemas operativos OSX, Linux, Windows y FreeBSD. La documentación para su manejo e instalación está disponible en la siguiente dirección web: <http://info.iet.unipi.it/~luigi/dummynet/>

En esta prueba es usado el sistema operativo Windows 7, para lo cual fue necesario utilizar la arquitectura de 32 bits puesto que es necesario el uso de un controlador sin firma digital para el adaptador de red. Es posible encontrar algunos inconvenientes para su instalación por permisos especiales en el sistema operativo, para ello es conveniente verificar que los scripts necesarios están en el directorio del sistema C:/Windows/System32/, ellos son: “**cygwin1.dll**” e “**ipfw**”.

Para facilitar la prueba de consumo de ancho de banda el script **test\_dummynet.bat** fue desarrollado (ver la figura 5.6), debe ejecutarse con permisos de administrador. El script primero limpia cualquier configuración de red hecha por la herramienta dummynet, después configura los parámetros para ajustar el ancho de banda de la red, a continuación el script es detenido mientras la prueba es realizada. Una vez terminada la prueba se presiona cualquier tecla y el script vuelve a deshacer la configuración de dummynet.



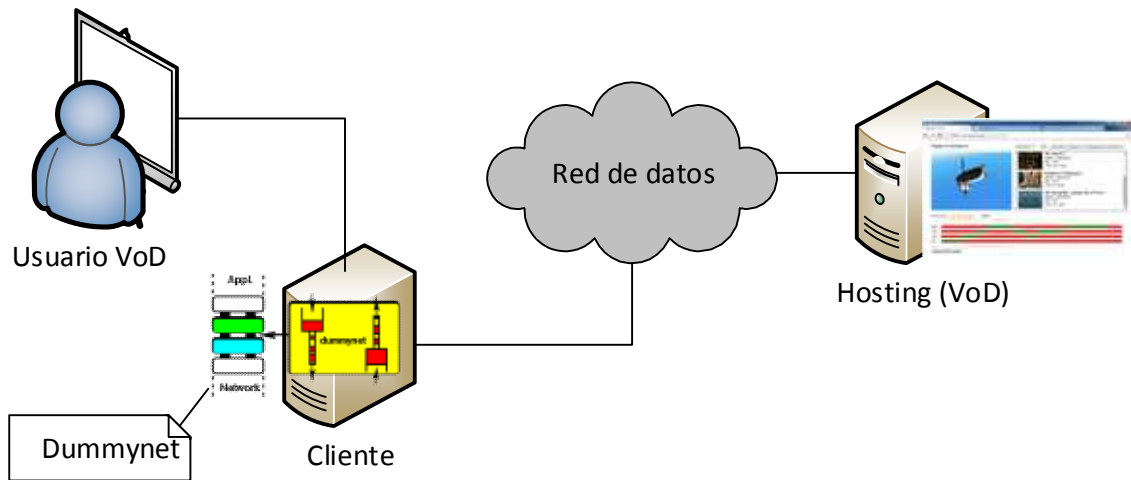
```
test_dummysnet.bat
1 @echo on
2 @set CYGWIN=nodosfilewarning
3 @ipfw -q flush
4 @ipfw -q pipe flush
5 @echo #####
6 @echo ## Setting up ##
7 @echo #####
8 ipfw pipe 3 config bw 200kbit/s mask all
9 ipfw add pipe 3 ip from any to any
10 ipfw pipe show
11
12 @echo ""
13 @echo "Network Tweaking Done, Start Testing."
14 @echo ""
15 @echo "Press Enter when testing is done, to restore original Network settings."
16 pause
17
18 @echo #####
19 @echo ## Cleaning up ##
20 @echo #####
21 @ipfw -q flush
22 @ipfw -q pipe flush
23 pause
```

**Figura 5.6.** Script para emulación de ancho de banda.

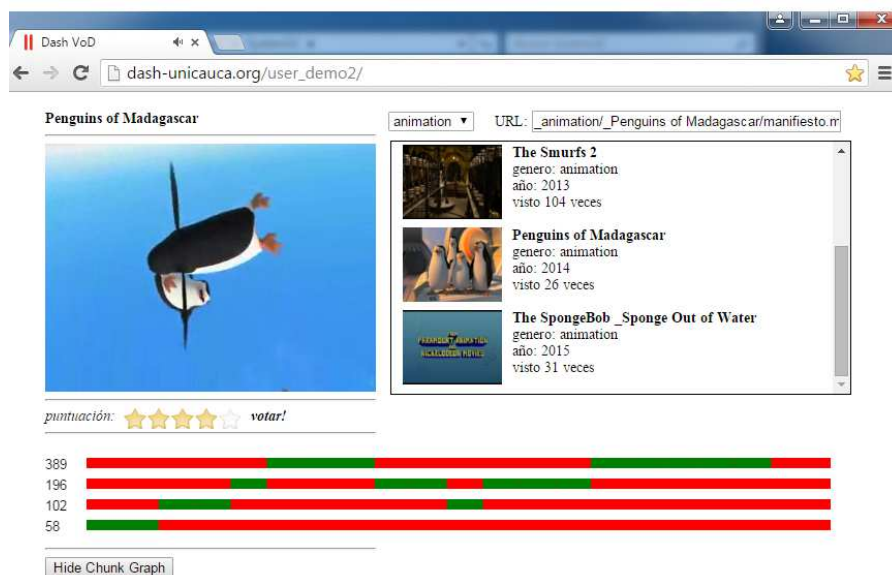
Es importante mencionar que, para ajustar el ancho de banda de la red, debe digitarse el valor del parámetro `bw` en la línea 8 del *script* `test_dummysnet`.

La emulación de ancho de banda fue realizada sobre el entorno de prueba mostrado en la figura 5.7, en donde fue utilizada una versión del servicio de VoD alojado en un *Hosting*, mientras que la herramienta de emulación corre sobre un equipo cliente con un ancho de banda de 1 Mbit/s, ofrecido por el operador de red.

La versión puesta a prueba del servicio de VoD DASH se encuentra disponible en: [http://dash-unicauca.org/user\\_demo2](http://dash-unicauca.org/user_demo2) (ver la figura 5.8).

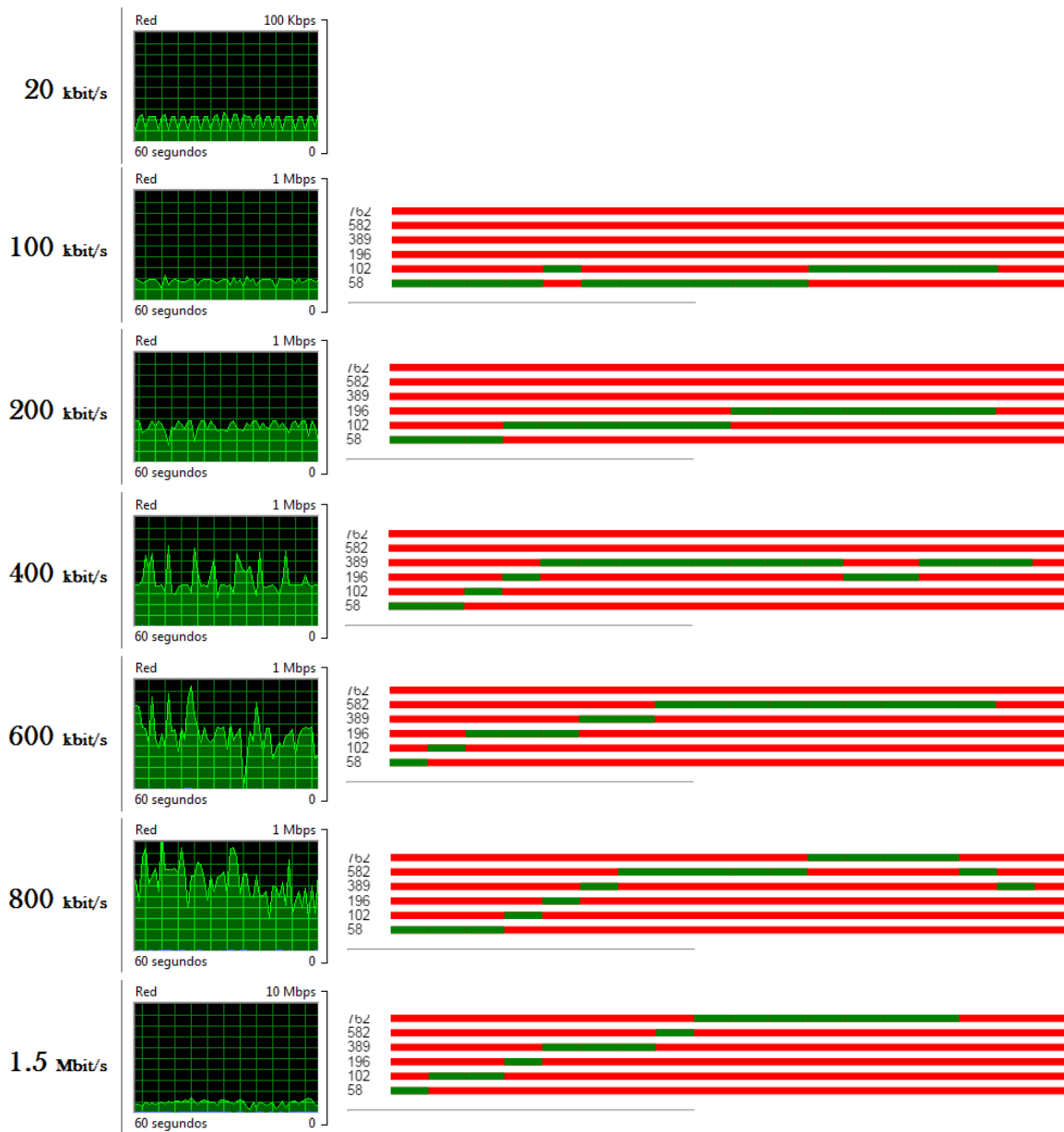


**Figura 5.7.** Entorno de prueba para emulación de ancho de banda.



**Figura 5.8.** Servicio de VoD en la nube.

Los valores de ancho de banda utilizados para la realización de la prueba fueron: 20, 100, 200, 400, 600, 800 y 1500 kbit/s. Los datos arrojados por la realización de la prueba pueden verse a continuación en la figura 5.9. En la parte izquierda de la gráfica es mostrado el monitor de recursos de Windows, donde es indicado el ancho de banda delimitado por dummynet, y en la parte derecha aparece el sistema de medición de ancho de banda, desarrollado sobre el cliente web.



**Figura 5.9.** Emulación de ancho de banda.

Como puede observarse en la figura 5.9, la herramienta dummynet cumple satisfactoriamente su función de emular el ancho de banda de la red, según el valor ingresado en el script. Según los resultados encontrados tras la realización de la prueba, puede concluirse que el servicio de VoD sigue los cambios emulados en el ancho de banda, de tal manera que cumple de manera satisfactoria los objetivos del estándar DASH.

## 5.4 Prueba de estrés

Como fue mencionado anteriormente, el servicio de chat fue implementado a partir del proyecto ST-CAV de la Universidad del Cauca, hace uso del servidor de aplicaciones webpy de Python [48]. También el esquema de consumo de este servicio es a través del estilo arquitectónico REST-JSON. Por lo tanto este será nuestro servicio objetivo en la prueba de estrés.

Para evaluar el servicio de chat integrado al servicio de VoD implementado en este trabajo, la herramienta Apache Benchmark fue utilizada, ella permite realizar múltiples peticiones a servicios web de modo secuencial y concurrente. El proceso de instalación e interacción de esta herramienta está descrito en el anexo C de este trabajo. A continuación cada una de las tareas y los datos obtenidos en esta prueba son descritos.

Para la prueba de estrés sobre el servicio de chat, 10, 30, 50, 100 y 500 peticiones en modo concurrente y secuencial son considerados, las figuras 5.10 y 5.11 muestran el tiempo de respuesta del servicio al realizar una serie de solicitudes de manera secuencial, obteniéndose un tiempo de respuesta cercano a 10 ms. Como puede observarse en ambas gráficas la primer solicitud realizada presenta mayor tiempo de respuesta, 0.08 segundos aproximadamente, a partir de ese momento el servidor responde en tiempos con variación muy suave.

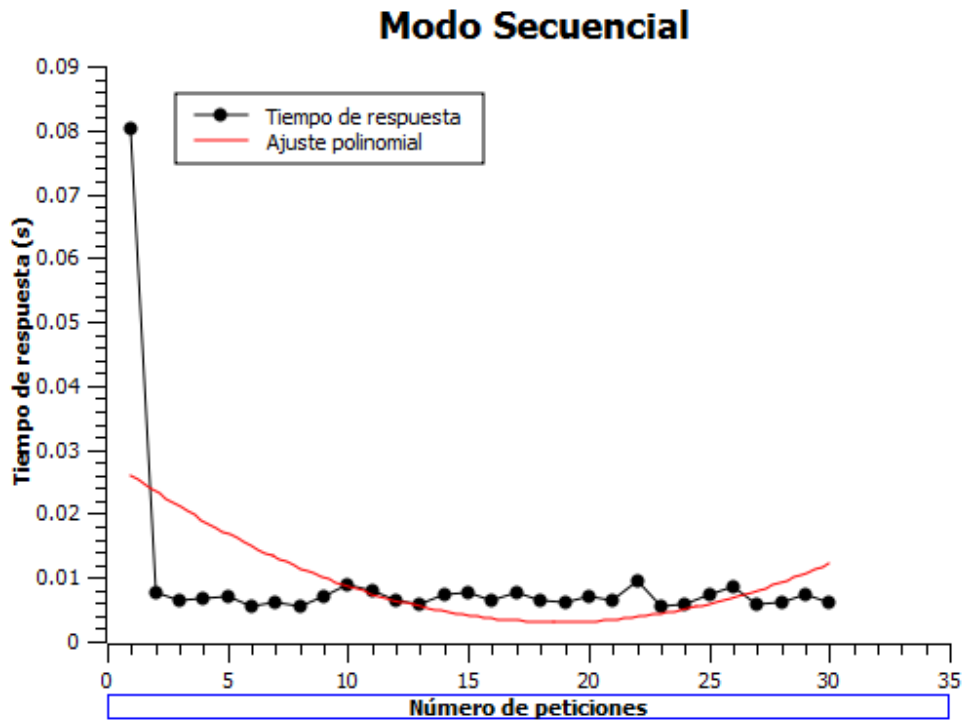


Figura 5.10. Peticiones en modo secuencial (30).

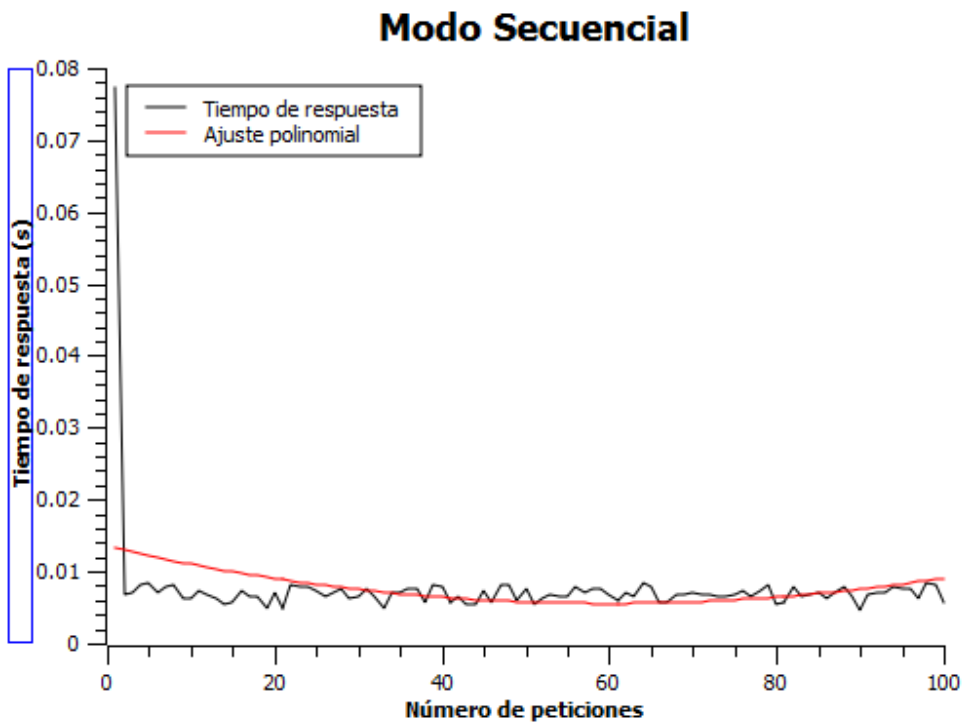
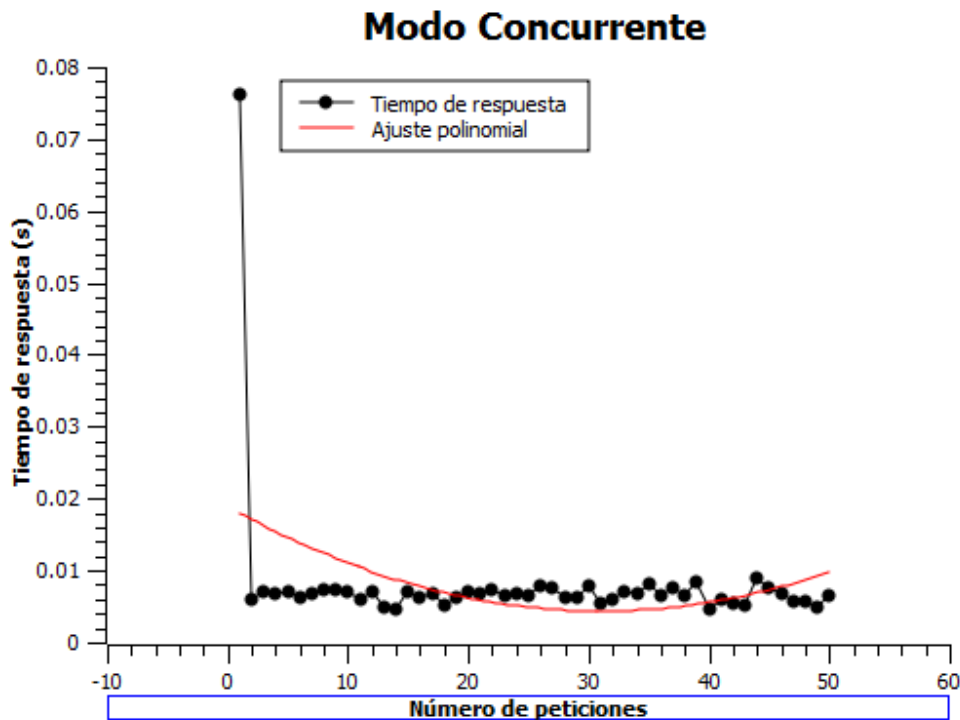


Figura 5.11. Peticiones en modo secuencial (100).

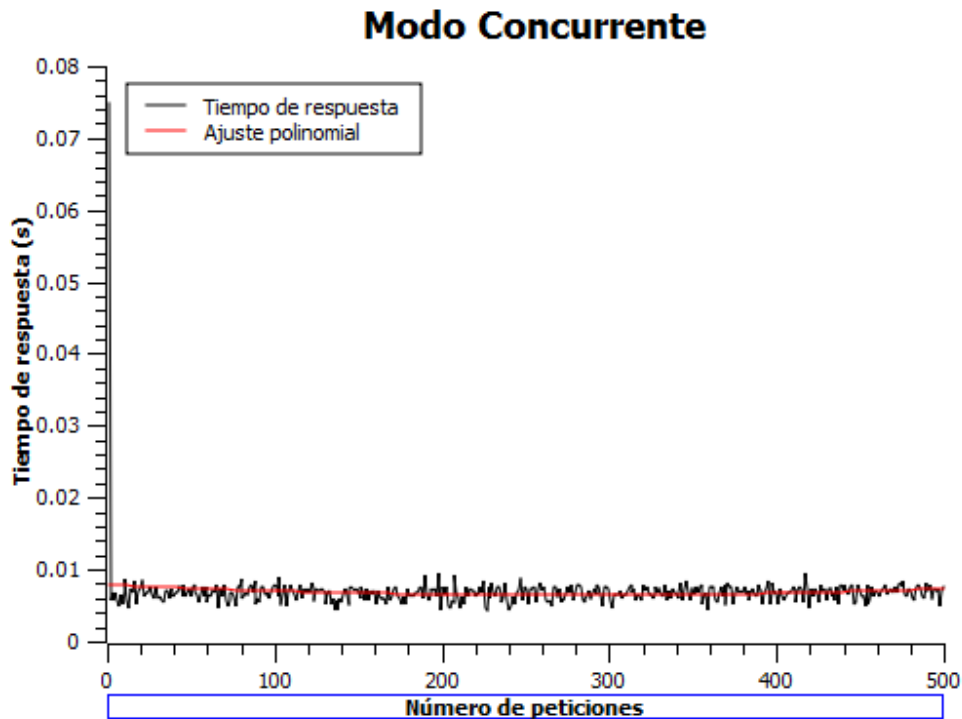
De la misma manera para la figura 5.12, es posible observar que al probarse 50 peticiones de modo concurrente sobre el servicio, existe un tiempo de respuesta no superior a los 10 ms. El ajuste polinomial denotado por la línea de color rojo, indica que el servicio posee un tiempo de respuesta poco variable para casi la totalidad de las conexiones simultáneas, realizadas sobre el mismo.



**Figura 5.12.** Peticiones en modo concurrente (50).

Fijándonos en las figuras 5.11 y 5.13 puede observarse que ambas presentan igual similitud, aun tratándose una en modo secuencial y la otra de manera concurrente, también puede deducirse que en ambos modos la respuesta es satisfactoria aun a 500 peticiones simultáneas, obteniéndose un promedio inferior a 10 ms por cada solicitud realizada al servidor.

Hasta este momento las gráficas de respuesta en modo secuencial y concurrente, presentan igual forma de respuesta, es decir, al inicio un tiempo de respuesta elevado en comparación al resto de las interacciones.



**Figura 5.13.** Peticiones en modo concurrente (500).

Finalmente las figuras 5.14 y 5.15 presentan las gráficas que muestran el tiempo total empleado por los grupos de peticiones realizadas al servicio. Según el ajuste lineal realizado (líneas de color rojo), en ellas puede observarse que el servidor responde prácticamente de igual manera en ambos modos de prueba (concurrente y secuencial), por lo tanto, puede deducirse que para entornos de prueba donde son manejados un número de conexiones simultáneas, inferiores a 500 peticiones, el servicio de chat funciona de manera eficiente.

Según los tiempos de respuesta de la prueba de estrés sobre el servicio de chat, puede concluirse que el servidor webpy presenta buena capacidad de respuesta al atender múltiples solicitudes de manera secuencial y concurrente, respecto a los tiempos de respuesta que pueden obtenerse utilizando mensajes SOAP (alrededor de las decenas de milisegundos), esto se debe al uso del estilo arquitectónico REST-JSON.



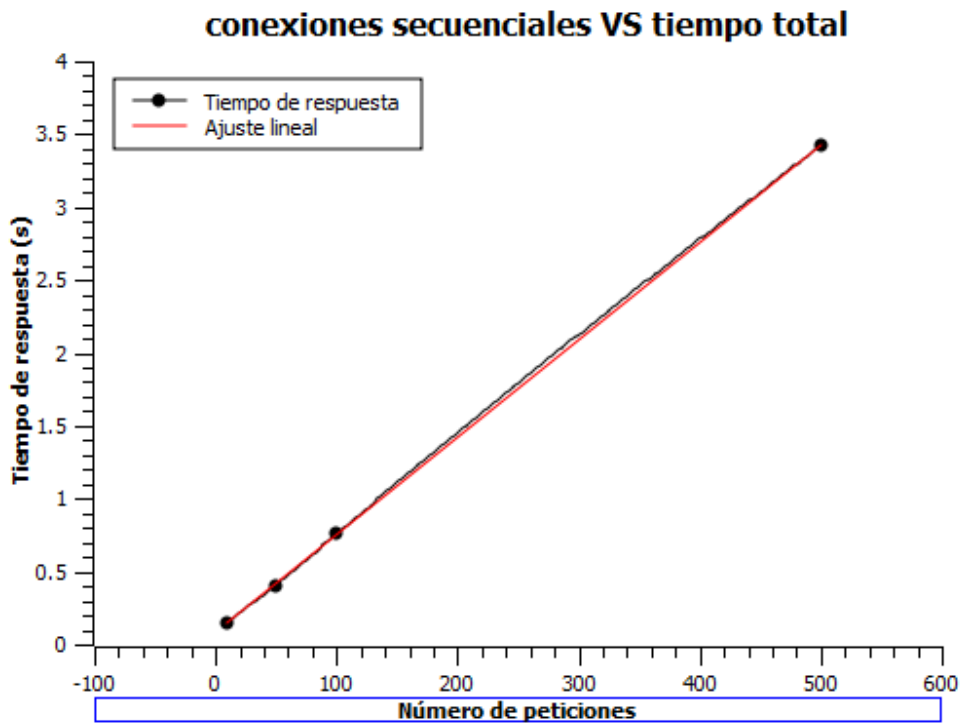


Figura 5.14. Peticiones en modo secuencial (total).

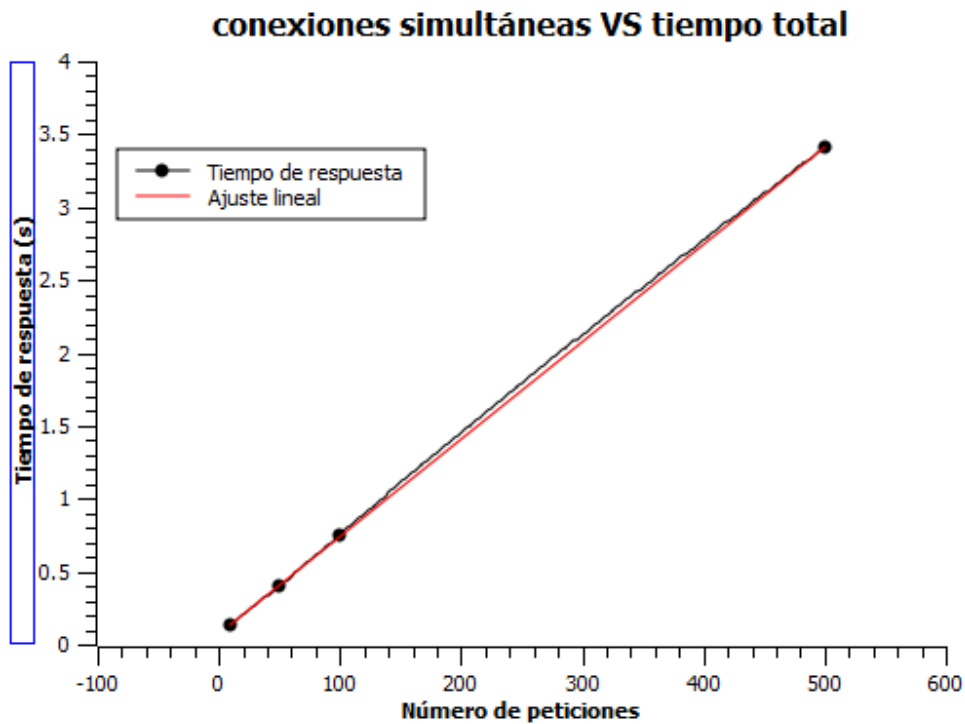


Figura 5.15. Peticiones en modo concurrente (total).

## Capítulo 6

### Conclusiones

#### 6.1 Introducción

Este capítulo presenta las conclusiones de este trabajo de grado, teniendo en cuenta los objetivos presentados en el primer capítulo. Además, trabajos adicionales y complementarios a este son propuestos para quienes deseen continuar en el proceso de investigación de la televisión digital interactiva.

#### 6.2 Conclusiones

- Para el soporte de servicios de transmisión de contenidos multimedia es necesaria una infraestructura tecnológica de un elevado costo. Este trabajo propone una infraestructura básica con componentes hardware de mediano costo, acompañado de herramientas software libre que permiten emular componentes reales de la cadena de televisión. Lo anterior facilita el despliegue de nuevos servicios.
- El banco de pruebas propuesto hace uso de herramientas libres para los módulos de codificación, difusión y recepción de contenidos multimedia DASH. También hace uso de protocolos abiertos en cuanto al esquema de acceso a los servicios interactivos. Por lo anterior este banco de pruebas es adecuado para ser implementado en entornos de prueba de centros educativos y de investigación en esta clase de tecnologías.

- Dentro de este trabajo fueron realizadas pruebas de transmisión de contenidos multimedia DASH a múltiples tasas de bit, como resultado fue verificado que el funcionamiento del estándar particularmente para contenidos en formato WebM.
- De acuerdo a la prueba de estrés realizada sobre el servicio interactivo de chat de este trabajo, como resultado fue obtenido un tiempo de respuesta inferior a 10 milisegundos para 500 conexiones simultaneas, por lo tanto puede considerarse que el uso de las herramientas implementadas, funcionan de manera eficiente garantizando un buen tiempo de acceso al contenido. Lo anterior sucede gracias a la utilización del estilo arquitectónico REST-JSON.
- La herramienta de codificación automática propuesta, permite optimizar el proceso de codificación, segmentación y generación del archivo de manifiesto MPD para contenidos multimedia adaptativos WebM DASH. Para lo anterior, esta herramienta integra en segundo plano las funcionalidades provistas de manera separada por las librerías libewbm, webm-tools y la herramienta de codificación FFmpeg. Así mismo, la herramienta Dash WebM Converter optimiza el tiempo de codificación al lanzar de manera simultánea tantos hilos como tasas de bits estén asociadas a un contenido multimedia.
- De acuerdo a las mediciones de consumo de memoria RAM realizadas sobre la herramienta FFmpeg, para un contenido multimedia con 6 tasas de bits a codificar, la cantidad de memoria RAM asignada a cada hilo de video es en promedio de 0.2 MB. Lo anterior permite inferir que el hecho de usar procesamiento multihilo no representa un incremento considerable en la cantidad de memoria RAM, con respecto a las ventajas obtenidas de optimización de tiempo.
- A través de la herramienta de codificación automática propuesta, es posible el despliegue de un escenario de transmisión de *streaming* adaptativo WebM basado en el estándar DASH. Este escenario tiene como ventaja la reducción de la infraestructura para el montaje de servicios de transmisión de contenido multimedia, al no requerir el uso de un servidor de *streaming*, puesto que los

contenidos y la lógica de navegación a través de ellos se encuentra centralizada en un servidor Web, evitando así problemas de sincronización.

- El escenario de *streaming* adaptativo en el que está la herramienta de codificación automática, recoge e integra las herramientas y librerías más adecuadas del mundo del software libre, para el despliegue de un servicio básico de transmisión y recepción de contenidos multimedia adaptativos WebM.
- La librería javascript webm-dash-javascript permite el despliegue adecuado del contenido multimedia DASH WebM desde un cliente web. Para lo anterior, esta librería estima el ancho de banda disponible en el lado del cliente, y de acuerdo a este valor obtiene el segmento de contenido multimedia apropiado para la reproducción, según la información disponible en el manifiesto MPD. De esta forma, webm-dash-javascript facilita la integración y reproducción de contenidos multimedia WebM en formato DASH a través del componente de video provisto por HTML5.
- La organización de la información del contenido publicado en el catálogo, puede ser modificada fácilmente debido a la sencillez del formato XML. Además la estructura de este formato (etiquetas), facilita la clasificación de contenidos multimedia de acuerdo a las preferencias del interesado.
- El uso del estándar de *streaming* adaptativo DASH, en un servicio de VoD, permite mejorar considerablemente la calidad del servicio en entornos de movilidad, respecto a los tradicionales protocolos de transmisión de contenidos multimedia RTP y RTSP, garantizando menor tiempo de arranque, sin buffer, permite seleccionar la calidad en función de la red y capacidad del dispositivo, además los cambios entre calidades son automáticos y transparentes al usuario.
- Un servicio de VoD de contenidos multimedia adaptativos DASH ofrece un enfoque diferente mediante el uso del protocolo HTTP, lo cual permite ser desplegado sobre la infraestructura existente en Internet y puede coexistir con tecnologías propietarias actuales.

- El uso del estilo arquitectónico REST-JSON facilita el acceso a las aplicaciones y a los recursos de red sin descuidar la velocidad de procesamiento en los dispositivos que trabajan sobre el estándar.
- El estilo arquitectónico REST-JSON es adecuado para el consumo de servicios interactivos asociados al servicio de VoD, debido a la sencillez y rápido consumo de recursos, proporcionado por este esquema y a la notable característica cliente-servidor que representa el servicio de VoD.
- La implementación de servidores dedicados para el servicio de VoD, los servicios interactivos asociados al mismo y la herramienta de codificación automática online, permite facilitar la escalabilidad y modularidad del banco de pruebas desarrollado en este trabajo.

### 6.3 Trabajos futuros

A continuación es presentado un listado de los posibles trabajos futuros complementarios a este.

- Implementación de un banco de pruebas y un servicio de video bajo demanda de contenidos multimedia adaptativo en tiempo real. La transmisión de flujos de video en vivo, adaptados al estándar DASH, será un gran reto de investigación para enriquecer los servicios de TDi. Adicionalmente será conveniente la realización de pruebas de retardo y jitter, los cuales son parámetros críticos en la calidad del servicio de VoD ofrecido.
- Es considerado pertinente la implementación de sistemas de recomendaciones basados en contexto (contexto de usuario), que permitan predecir la valoración de nuevos contenidos multimedia DASH a partir de valoraciones hechas por los usuarios sobre determinados ítems o contenidos. Para ello puede implementarse una arquitectura que sea desplegada sobre un entorno de movilidad.

- El banco de pruebas planteado en este trabajo de grado puede complementarse mediante la implementación de un sistema buscador de contenidos, esta solución es conveniente cuando la cantidad de contenidos multimedia DASH sea considerable. De igual manera es posible implementar un método adaptado para la inferencia y clasificación de los contenidos a partir de variables de contexto del servicio de VoD.
- Se considera adecuada la implementación de los demás servicios interactivos, desarrollados sobre el servidor webpy del proyecto ST-CAV de la Universidad del Cauca (servicio de Comentarios, Tablón, Programación, Notificaciones, Perfiles de usuarios, entre otros), para complementar el servicio de VoD desarrollado en este trabajo. Además puede considerarse la implementación de un servicio de guía interactiva para el usuario, esta guía será adecuada cuando sean añadidas más funcionalidades y servicios interactivos asociados al servicio de VoD de contenidos multimedia DASH.
- Se pretende ampliar el funcionamiento de la herramienta de codificación automática Dash WebM Converter, de tal manera que se pueda extender la funcionalidad de codificación a contenidos multimedia adaptativos en formato MPEG4. También es posible optimizar dicha herramienta mediante las configuraciones que ofrece FFMPEG, mediante la dedicación de núcleos del procesador para cada hilo de codificación.



## Referencias

- [1] Sandvine Intelligent Broadband Networks, «Global Internet Phenomena Report,» *Sandvine*, vol. 2H 2013, pp. 5,6,9,10,23, 2013.
- [2] Sandvine Intelligent Broadband Networks, «Global Internet Phenomena Report,» *Sandvine*, vol. 2H 2013, p. 24, 2013.
- [3] C. Mueller, S. Lederer, C. Timmerer y H. Hellwagner, «Dynamic adaptative streaming over HTTP/2.0,» *Alpen-Adria-Universität Klagenfurt*, 2013.
- [4] Dataxis - Latam Telecom Reports, «Mercado OTT en America Latina, 2013 - 2018,» 3 10 2013. [En línea]. Available: [http://dataxishop.dataxis.com/Brochure\\_OTT\\_Latam\\_2013.pdf](http://dataxishop.dataxis.com/Brochure_OTT_Latam_2013.pdf). [Último acceso: 08 06 2014].
- [5] J. Altgeld y D. Z. John, «The IPTV/VoD Challenge: Upcoming Business Models,» de *Achieving the Triple Play: Technologies and Business Models for Success*, Intl. Engineering Consortiu, 2006, p. 3.
- [6] O. D. Torres, «Evolución y tendencia de la tecnología de streaming en internet,» *FACE - UNNE*, p. 8, 2009.
- [7] S. Mack, «Streaming Media Bible,» John Wiley & Sons, 2002, p. 869.
- [8] D. Costilla Mateos y S. Reaño Montoro, «Streaming de Audio/Video. Protocolo RTSP,» *enginy@eps-Curs 2007/2008*, p. 15, 2008.
- [9] C. Müller y C. Timmerer, «A VLC Media Player Plugin enabling Dynamic Adaptive Streaming over HTTP,» *Alpen-Adria-Universität Klagenfurt - Multimedia Communication*, pp. 723-724, 2011.
- [10] 3GPP TS 26.234, «Transparent end-to-end Packet-switched Streaming Service (PSS),» 3GPP a Global Initiative, 2010. [En línea]. Available: <http://www.3gpp.org/DynaReport/26234.htm>. [Último acceso: 05 02 2014].
- [11] ISO/IEC 23009-1:2012, «Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats,» *ISO/IEC 2012*, p. 3, 4 1 2014.



- [12] Microsoft Corporation, «IIS Smooth Streaming Technical Overview,» 25 3 2009. [En línea]. Available: <http://www.microsoft.com/en-us/download/details.aspx?id=17678>. [Último acceso: 19 1 2014].
- [13] W. May y R. Pantos, «HTTP Live Streaming draft-pantos-http-live-streaming-07,» 30 9 2011. [En línea]. Available: <http://tools.ietf.org/html/draft-pantos-http-live-streaming-07>. [Último acceso: 20 1 2014].
- [14] Adobe, «HTTP Dynamic Streaming,» 2012. [En línea]. Available: <http://www.adobe.com/products/hds-dynamic-streaming.html>. [Último acceso: 20 1 2014].
- [15] CSI Analysing Converting technologies, «DASH it all,» *CSI Analysing Converting technologies*, p. 39, 24 05 2013.
- [16] B. C. Núñez, «DASH: Un estándar MPEG para streaming sobre HTTP,» *Facultat d'Informatica de Barcelona - Universitat Politècnica de Catalunya*, 2013.
- [17] J. D. Gambín Tomasi, «Desarrollo de un servicio de televisión interactiva HbbTV según el estándar ETSI TS 102 796 v1.1.1 (JUN 2010),» *UNIVERSIDAD POLITÉCNICA DE CARTAGENA - E.T.S. Ingeniería de Telecomunicación*, pp. 111-112, 2012.
- [18] W. Lu, C. Guoqiang, L. Weibin y X. Weiwei, Proceedings of the 2012 International Conference on Information Technology and Software Engineering, Berlin Heidelberg: Springer- Verlag, 2013, p. 881.
- [19] A. Lindley, A. N. Jackson y B. Aitken, «A Collaborative Research Environment for Digital Preservation - the Planets Testbed,» *1st International Workshop on Collaboration tools for Preservation of Environment and Cultural Heritage at*, p. 2, 2010.
- [20] U. d. Michigan, «Software Engineering in Higher Education,» *International Conference on Software Engineering in Higher Education*, vol. 1, p. 166, 2007.
- [21] U. d. Cauca, «EDiTV,» Universidad del Cauca, [En línea]. Available: <http://www.unicauca.edu.co/EDiTV/>. [Último acceso: 2014 05 24].
- [22] W. Campo, G. Chanchi y J. Arciniegas, «Servicios de T-learning para el soporte de una Comunidad Académica Virtual (ST-CAV),» *RENATA Colombia*, 2009.
- [23] J. Pindado, «T-Learning El Potencial Educativo De La Televisión Digital linteractiva,» *Universidad de Málaga*, p. 3, 2006.
- [24] Z. Li y I. Bouazizi, «Light Weight Content Fingerprinting for Video Playback Verification in MPEG DASH,» *Samsung Research America*, 2013.
- [25] C. Muller, S. Lederer, B. Rainer, M. Waltl, M. Grafl y C. Timmerer, «Open Source Column: Dynamic Adaptive Streaming over HTTP Toolset,» *ACM SIGMM Records*, 16 9 2013.

- 
- [26] C. Timmerer y C. Griwodz, «Dynamic Adaptive Streaming over HTTP: From Content Creation to Consumption,» *MM'12*, 2 Noviembre 2012.
- [27] G. E. Chanchí, W. Y. Campo, J. P. Amaya y J. L. Arciniegas, «Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON,» *Cadernos de Informática*, vol. 6, nº 1, 2011.
- [28] C. Müller y C. Timmerer, «A Test-Bed for the Dynamic Adaptive Streaming over HTTP featuring Session Mobility,» *ACM Multimedia Systems*, 25 Febrero 2011.
- [29] B. Rainer, S. Lederer, C. Müller y C. Timmerer, «A Seamless Web Integration of Adaptive HTTP Streaming,» de *20th European Signal Processing Conference (EUSIPCO 2012)*, Bucharest, 2012.
- [30] J. M. Batalla y S. Janikowski, «In-Segment Content Server Adaptation for Dual Adaptation Mechanism in DASH,» *National Institute of Telecommunications*, 2013.
- [31] J. Lemiegre, «2.3.6 Over the top TV,» de *Bye-Bye Advertising: How Generation Y will change the business model of commercial TV in Flanders*, London, 2012, p. 15.
- [32] J. J. Parsons y D. Oja, «DRM TECHNOLOGIES,» de *New Perspectives on Computer Concepts 2013: Introductory*, 2012, p. 466.
- [33] R. Tortosa, J. M. Jimenez, J. R. Diaz, J. Lloret, «Optimal codec selection algorithm for audio streaming,» *Univ. Politec. de Valencia, Valencia, Spain*, 2014.
- [34] I. -M. Pao, Ming-Ting Sun «Encoding stored video for streaming applications,» Dept. of Electr. Eng., Washington Univ., Seattle, WA, USA, 2001.
- [35] C. Muller, S. Lederer, B. Rainer, M. Walzl, M. Grafl y C. Timmerer, «Open Source Column: Dynamic Adaptive Streaming over HTTP Toolset,» *ACM SIGMM Records*, 16 9 2013.
- [36] J. Bailey, «Live Video Streaming from Android-Enabled Devices to Web Browsers,» University of South Florida, 2011.
- [37] B. Rainer, S. Lederer, C. Müller y C. Timmerer, «A Seamless Web Integration of Adaptive HTTP Streaming,» de *20th European Signal Processing Conference (EUSIPCO 2012)*, Bucharest, 2012.
- [38] B. Rainer, S. Lederer, C. Müller y C. Timmerer, «A Seamless Web Integration of Adaptive HTTP Streaming,» de *20th European Signal Processing Conference (EUSIPCO 2012)*, Bucharest, 2012.
- [39] N. Vun, M. Ansary, «Implementation of an embedded H.264 live video streaming system,» Sch. of Comput. Eng., Nanyang Technol. Univ., Singapore, Singapore, 2010.

- [40] M. Olson, K. Christensen, Lee SangHak, Yun JungMee, «Hybrid web server: Traffic analysis and prototype,» Dept. of Comput. Sci. & Eng., Univ. of South Florida, Tampa, FL, USA, 2011.
- [41] W. Hamidouche, G. Cocherel, J. Le Feuvre, M. Raulet, O. Deforges, «4k real time video streaming with SHVC decoder and GPAC player,» IETR/INSA, Rennes, France, 2014.
- [42] G. A. Hoffmann, K. S. Trivedi, M. Malek, «A best practice guide to resources forecasting for the apache webserver,» Dept. of Electr. & Comput. Eng., Duke Univ., Durham, NC, 2006.
- [43] Oliphant, E. Travis, «Python for scientific computing,» Brigham Young University, Prove, 2007.
- [44] S. Gude, M. Hafiz, A. Wirfs-Brock, «Javascript: The used parts,» Auburn University, Auburn, AL, USA, 07 2014.
- [45] WebM Project, «Instructions to playback Adaptive WebM using DASH,» [En línea]. Available: <http://wiki.webmproject.org/adaptive-streaming/instructions-to-playback-adaptive-webm-using-dash>. [Último acceso: 2 06 2015].
- [46] Zeng Hao, Cao Guang-Li, Che Hua-Xiang, «Performance improvement of DSS based on high-definition video on demand,» *Computer Science & Service System (CSSS), 2012 International Conference on*, Sch. of Commun. & Inf. Eng., Chongqing Univ. of Posts & Telecommun., Chong Qing, China, 08 2012.
- [47] Nan Zheng, X. Liu, «Load balance optimization of a Red5 cluster in the mobile classroom project,» *Natural Computation (ICNC), 2013 Ninth International Conference on*, Software Eng. Inst., East China Normal Univ., Shanghai, China, 07 2013.
- [48] «Servidor webpy,» [En línea]. Available: <http://webpy.org>. [Último acceso: 8 06 2015].