MARCO DE REFERENCIA PARA LA INTEGRACIÓN ARQUITECTÓNICA DE SERVICIOS DE STREAMING A SISTEMAS DE GESTIÓN DE APRENDIZAJE DE CÓDIGO ABIERTO

Anexos

Rubén Darío Benavides Cabrera
José Leonardo Díaz Ordoñez

Director
Mag. Mario Fernando Solarte Sarasty

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
DEPARTAMENTOS DE TELEMATICA Y SISTEMAS
Línea de Investigación en Aplicaciones y Servicios sobre Internet

Popayán, Mayo de 2010

Tabla de contenido

Anexo 1. Protocolos, códecs y estándares

1.1 Pila de protocolos

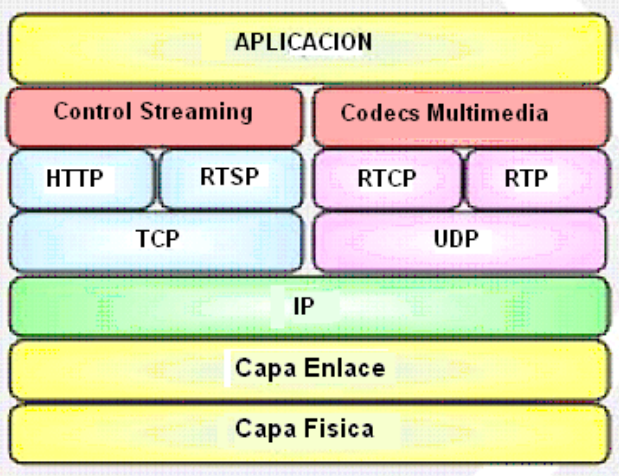

Figura 1. Pila de protocolos para aplicaciones *streaming.*

Desde el desarrollo de la tecnología de *streaming* se han establecido y elaborado un conjunto de protocolos enfocados en la correcta aplicación de este tipo de tecnología.

Como se mencionó con anterioridad, debido a la interpretación y uso del *streaming* como servicio tradicional de audio/video, se desarrollaron un conjunto de protocolos especializados en este tipo de servicios, por otro lado, existen protocolos más generales que aunque no se enfocaron en los servicios de *streaming,* si fueron diseñados para servicios que requirieran de tiempos de retardo mínimo. Uno de ellos tienen como objetivo inicializar y controlar las sesiones, entre los que se encuentran RTCP, RTSP y SDP, otro protocolo se creó exclusivamente para transferir los datos de la carga útil, este es conocido como RTP, enfocado a aplicaciones de tiempo real.

1.1.1 UDP, User Datagram Protocol

Es un protocolo de nivel de transporte no orientado a la conexión, es decir, no tiene cuidado en el orden de transmisión de los paquetes y tampoco garantiza la llegada de los mismos. UDP tiene el objetivo de mejorar la velocidad en la entrega de los datos sobre la confiabilidad, por estas razones UDP es usado comúnmente para transmisiones multimedia en tiempo real.

El uso exclusivo de UDP para servicios de *streaming* implica un riesgo en la disponibilidad y accesibilidad de los servicios debido a las prácticas comunes de seguridad que se aplican en los entornos empresariales, donde algunos puertos y protocolos son bloqueados, entre los que se destacan el puerto del protocolo UDP [1].

1.1.2 TCP, Transmission Control Protocol

Al igual que UDP, este protocolo es usado para el transporte de la carga útil. TCP es orientado a la conexión, verificando la recepción de cada paquete que es enviado [2].

Tradicionalmente el *streaming* se relaciona, a manera de obligación, al uso del protocolo UDP como medio de transporte frente a TCP. La razón principal radica en que los medios que garantizan el acuse de recibo de los paquetes y los mecanismos de retransmisión, en TCP, pueden provocar demoras del lado del cliente que violan los requerimientos de latencia en las transmisiones de *streaming*. Sin embargo, TCP es usado ampliamente en sistemas de *streaming* comerciales, como los basados en soluciones con Windows Media, Real Media o Flash Media Server, y en la mayoría de sistemas que ofrecen servicios de *streaming* en Internet [3]. Esta tendencia está corroborada según [4], en donde se afirma y concluye la factibilidad de usar TCP como medio de transmisión para servicio de *streaming*.

Aunque es notable la existencia de una deficiencia del *streaming* basado en TCP, se han desarrollado algunas estrategias que mejoran el rendimiento de las transmisiones como el uso de buffers en la capa de aplicación, lo que minimiza las pequeñas fluctuaciones de carga TCP (*TCP throughput*), la adaptación de la tasa de cambio de bits de los recursos audio-visuales que mejoran las fluctuaciones prolongadas en la carga TCP o la eliminación de paquetes o *frames* del lado del servidor cuando se detecta algún tipo de latencia en el cliente, usada generalmente en transmisiones audio-visuales en vivo [5-8].

1.1.3 HTTP, HyperText Transfer Protocol

HTTP es un protocolo de la capa de aplicación que permite la recepción remota de datos, este se basa en transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor [9].

Aunque HTTP no fue diseñado para la transmisión de recursos multimedia, mucho menos basado en *streaming*, ha sido ampliamente utilizado como medio para la transmisión de contenido *streaming* y como tal es interesante resaltar [10, 11]:

- HTTP *streaming* es quizás el protocolo de *streaming* más simple, ya que no emplea ningún tipo de adaptación de tasa de bits en la capa de aplicación, a diferencia de otras aproximaciones de *streaming* basado en TCP [5-8].

- No se necesita ningún otro mecanismo para asegurar la interoperabilidad con TCP, ya que se basa en este último.
- No necesita de mecanismos adicionales para recuperarse de la pérdida de conexiones de *streaming*, a diferencia de las basadas en UDP.
- HTTP *streaming*, tiene un amplio nivel de acceso y disponibilidad, ya que la mayoría de redes acepta las transmisiones de este tipo.

1.1.4 RTP, Real-time Transport Protocol

Es un protocolo definido por la IETF en el RFC 3550 [12], proporciona solución extremo a extremo para el transporte de datos multimedia en tiempo real, con RTP se puede soportar diferentes tipos de transmisiones como *unicast* y *multicast*, se debe aclarar que RTP es exclusivamente un protocolo de transporte, esto implica que no está encargado de la calidad del servicio en ninguna transmisión de *streaming*, RTP es independiente de los protocolos de transporte subyacentes y de las redes en general. RTP permite:

- Identificar el tipo de información transmitida.
- Agregar marcadores temporales y números de secuencia a la información transmitida.
- Controlar la llegada de los paquetes a su destino.
- Define paquetes de difusión múltiple, los cuales pueden utilizar RTP para establecer comunicaciones con múltiples destinos.

1.1.5 RTCP, RTP Control Protocol

La principal función de RTCP [12] es proporcionar información sobre la calidad del servicio de RTP durante la transmisión de datos, enviando información estadística de forma periódica a los participantes en una sesión de *streaming* multimedia. También se transmiten información de control de un flujo RTP.

La información que reúne y transmite RTCP se utiliza para que las aplicaciones controlen los parámetros asociados a la calidad del servicio, QoS, por ejemplo limitando el tamaño del flujo de información o usando un códec diferente para la codificación/decodificación de la información. Algunos datos que se envían mediante RTCP son:

- Bytes transmitidos.
- Conteo de paquetes transmitidos.
- Conteo de paquetes perdidos.
- Jitter.
- Tiempo de retardo de paquetes (RTT, *Round-Trip delay Time*).

RTCP no ofrece ningún mecanismo de seguridad, encriptación ni de autenticación, estos se implementan mediante otro tipo de protocolos y estrategias.

1.1.6 RTSP, Real-Time Streaming Protocol

Es un protocolo de nivel de aplicación que trabaja sobre el puerto 554, usado para establecer y controlar uno o varios tiempos de sincronización para el envío continuo de *streaming* multimedia, como el audio o el video, es decir, se utiliza para el control de la entrega de datos en tiempo real [13]. RTSP no es usado para entregar la carga útil, aunque es posible hacerlo, normalmente se usa RTP para este propósito. RTSP puede pensarse como el control de la red para los servidores multimedia. El funcionamiento óptimo de RTSP ocurre cuando trabaja de la mano con RTP y RTCP.

RTSP es un protocolo basado en texto, similar a HTTP, pero difiere de éste, ya que RTSP es un protocolo basado en la sesión de usuario. Al igual que RTP, RTSP es independiente de los protocolos de transporte, en consecuencia la sesión de usuario es administrada en el mismo nivel del protocolo.

RTSP cuenta con un mecanismo de control aleatorio que le permite un acceso fortuito a los datos que se transmiten, permitiendo realizar funciones como pausar, reproducir, parar, adelantar y retroceder sobre la transmisión del recurso, generalmente de *streaming* audio-visual.

1.1.7 AMF, Action Message Format

AMF, es un protocolo introducido en el año 2001, en el reproductor Flash Player 6, en su momento por la empresa Macromedia, adquirida posteriormente por Adobe Systems Inc. Aunque es un protocolo propietario también es abierto bajo licencia de software libre.

AMF permite serializar y codificar objetos del lenguaje de programación ActionScript bajo un formato binario compacto, que permite la comunicación remota entre dos puntos usando tipos de datos de alto nivel.

El protocolo AMF está inspirado en el protocolo SOAP, *Simple Object Access Protocol*, enfocado a la ejecución o llamada de procedimientos remotos, RPC [14].

1.1.8 RTMP, Real Time Messaging Protocol

RMTP [15], es un protocolo usado por las tecnologías de la plataforma Flash de Adobe, principalmente por Adobe Flash Player y Adobe AIR. Se trata de un protocolo de alto rendimiento para la transmisión en tiempo real de audio, video y otro tipo de datos sobre una conexión binaria del protocolo TCP. Por defecto este protocolo usa el puerto 1935, sin

embargo bajo ciertas condiciones puede ser consumido en otros puertos, incluso el puerto 80.

RMTP define el uso de diferentes canales para la transmisión de flujos de información, independientes entre sí, pero gestionados por mensajes de control común a todos los canales.

Cada canal transmite fragmentos de mensajes. Esto permite, por ejemplo, prevenir que mensajes de gran tamaño y de baja prioridad bloqueen pequeños mensajes de alta prioridad. Así mismo facilita la multiplexación de los diferentes canales sobre una sola conexión persistente TCP y asegura que cada canal alcance los requerimientos de ancho de banda, latencia y otras propiedades relacionadas con QoS.

Algunos de los canales que define RTMP tienen aplicaciones específicas. Por ejemplo existe un canal dedicado para manejar las peticiones y respuestas RPC, un canal para datos de video *streaming*, un canal para audio *streaming*, un canal dedicado para la transmisión de mensajes de control, que permite entre otras la negociación del tamaño de los fragmentos, entre otros [16].

Algunos de los beneficios que ofrece RMTP:

- Bajo una sesión RMTP, todos los canales pueden estar activos de forma simultánea. Dependiendo del tipo de aplicación y de las necesidades del servicio se puede negociar las prioridades de cada canal desde el servidor.
- El tamaño de las cabeceras de los paquetes y mensajes en RMTP varían de acuerdo al tipo de mensajes o servicios que se consuman, maximizando la carga útil que se pueda transmitir.
- Gracias a la multiplexación que se hace a nivel de los fragmentos de mensajes se logra la reutilización y aprovechamiento de los canales, inclusive permitiendo el flujo de datos simultaneo de dos o más recursos audio-visuales sobre la misma conexión TCP, útil para servicios como IPTV donde varias señales de video son transmitidas simultáneamente al cliente.
- Las aplicaciones de alto nivel desarrolladas sobre el protocolo RTMP tienen la capacidad de gestionar dinámicamente el ancho de banda del flujo de *streaming*, optimizar un canal para un servicio específico (definiendo un tamaño para el fragmento de mensaje) y controlar, en cierta medida, los mensajes que se pueden eliminar bajo condiciones especiales. Por ejemplo, para una transmisión de un canal de video y otro de audio sobre una conexión de baja velocidad, es deseable mantener el canal de audio con alta prioridad, el servidor puede decidir la eliminación algunos mensajes que transporten datos del video para asegurar una transmisión de audio limpia y continua.

1.1.9 RTMPT, RTMP Tunneled

RTMPT es básicamente un enmascaramiento sobre HTTP del protocolo RTMP, el cual es enviado mediante una petición tipo POST desde el cliente hacia el servidor. Debido a la naturaleza no persistente de las conexiones HTTP, RTMPT requiere que el cliente invoque periódicamente al servidor para ser notificado de cualquier evento que se genere por el mismo o por otro cliente [17]. El uso de este protocolo solventa los inconvenientes que presentan algunas redes, cuando implementan reglas de enrutamiento avanzadas que solo permiten transmisión de paquetes HTTP sobre el puerto 80. El hecho de enmascarar los paquetes de RTP sobre los de HTTP genera mayores latencias y retardos en el servicio, comparado con el uso nativo de RTP.

RTMPT es un protocolo propietario, cuya especificación oficial sigue siendo cerrada y no publica. Sin embargo existen especificaciones del protocolo no oficiales [18], realizadas mediante ingeniería inversa que buscan la interoperabilidad entre diferentes sistemas de código abierto.

1.1.10 RTMPS, RTMP over SSL

Este protocolo tiene una funcionalidad similar a RTMPT, pero en este caso el enmascaramiento se realiza sobre una conexión segura mediante SSL, similar a HTTPS [17].

Al igual que RTMPT, el protocolo RTMPS es cerrado y no existe una especificación oficial abierta al público.

1.1.11 RTMPE, Encrypted Real-Time Messaging Protocol

Este protocolo tiene un propósito similar a RTMPS, sin la sobrecarga de las cabeceras del protocolo de enmascaramiento. RTMPE [19] cifra el contenido antes de transmitirlo, a diferencia de SSL no usa intercambio de claves, pero cifra el contenido con claves generadas en el servidor. Este protocolo requiere menos tiempo de procesamiento en el servidor.

Al igual que RTMPT, el protocolo RTMPE es cerrado y no existe una especificación oficial abierta al público. Aunque existen especificaciones no oficiales, debido a la naturaleza y aplicación de este protocolo Adobe Inc. ha presentado quejas y como resultado es ilegal hacer uso de este protocolo o cualquier implementación relacionada [20].

1.1.12 MMS, Microsoft Media Server

Se trata de un protocolo desarrollado por Microsoft para la transmisión de *streaming* de audio/video en los modos *unicast/multicast* [21]. El protocolo original hacia uso de los

protocolos de transporte UDP y TCP en el puerto 1775. En el año 2008 el soporte, desarrollo y mantenimiento para este protocolo terminó, sin embargo todavía se mantiene activo para efectos de compatibilidad con versiones de servicios y aplicaciones anteriores [22]. Actualmente cuando se hace referencia a un recurso mediante este protocolo, primero se hace una serie de pruebas en busca de la estabilización de la conexión con el servidor de *streaming* así:

a. Se trata de establecer la conexión mediante RTSP sobre UDP, si falla, se intenta la conexión sobre TCP.
b. Si la anterior falla, se intenta la antigua versión de MMS sobre UDP y posteriormente sobre TCP.
c. Si no se establece ninguna de las anteriores se intenta la conexión mediante el protocolo MS-WMSP [23], *Windows Media HTTP Streaming Protocol*, el cual es básicamente un enmascaramiento sobre el protocolo HTTP del protocolo MMS.

1.2 Codecs y estándares

Dentro de los codecs mas importantes se encuentran los desarrollados por el grupo de trabajo de un subcomité de ISO/IEC (*International Organization for Standarization / International Electrotechnical Commission*), conocido como el Grupo de Expertos en Imagenes en Movimiento o MPEG, *Moving Picture Experts Group,* encargado del desarrollo internacional de estándares para la compresión, decompresión, procesado y representación codificada de vídeo, audio y su combinación, así mismo, se destaca el grupo de expertos en codificación de video, VCEG**,** de la ITU-T, Sector de estandarización de las telecomunicaciones, antiguamente conocido como el CCITT, Comité Consultivo Internacional Telegráfico y Telefónico, encargado del desarrollo de la línea de recomendaciones H.26X**.**

1.2.1 MPEG-1

Estándar popular, ISO/IEC 11172-3, para el audio y vídeo, publicado en el año 1993, uno de los primeros estándares que se usó para la publicación de contenido multimedia en Internet. Proporciona codificación de un canal (mono) o dos canales (estéreo o mono dual) con tasas de muestreo de 32, 44.1 y 48 KHz.

Estandariza tres esquemas distintos para la codificación de ondas de sonido denominados capas (*layers*) I, II y III. No estandariza el codificador sino el tipo de información que éste debe producir y cómo un decodificador debe dividir, descomprimir y

sintetizar esta información para obtener el sonido codificado [24]. Las tasas de bits predefinidas son:

- **Layer I**: De 32 a 448 Kbps
- **Layer II**: De 32 a 384 Kbps
- **Layer III**: De 32 a 320 Kbps

En la actualidad la capa de audio 3 del estándar, *MPEG-1 audio layer 3,* más conocido como MP3, se usa para la distribución de música bajo demanda o *streaming* en modelos económicos exitosos en países desarrollados [25].

1.2.2 MPEG-2

Definido bajo el estándar ISO/IEC 13818-1, publicado en el año 1995 con actualización en 1997 bajo el nombre MPEG-2 AAC, ISO/IEC 13818-7, MPEG-2 es una extensión multicanal compatible con MPEG-1. Permite hasta cinco canales principales y además un canal de mejora de bajas frecuencias, identificado comúnmente como sonido 5.1. Adicionalmente ofrece una extensión para tasas de muestreo menores. Ofrece tasas de muestreo a 16, 22.05 y 24 KHz para tasas de bits, *bitrate*, de 32 a 256 Kbps para la capa I y de 8 a 160 Kbps para las capas II y III.

MPEG-2 vídeo no está optimizado para bajas tasas de bits (menores que 1 Mbps), pero supera en desempeño a MPEG-1 a 3 Mbps y superiores, por esto es usado tradicionalmente para transmisiones de televisión en formato estándar y actualmente se usa para transmisiones de televisión de alta definición, HDTV. También es usado en los formatos de video en DVD y SVCD [26].

El estándar MPEG-2 AAC mejora ostensiblemente la calidad de audio, en comparación con las versiones anteriores, definiendo hasta 48 canales con tasas de muestreo desde 8 hasta 96 KHz con capacidades multicanal, multilenguaje y multiprograma. De igual forma define tasas de bits desde 8 Kbps para señales monofónicas de voz hasta más de 160 Kbps por canal para codificación de alta definición.

1.2.3 MPEG-4

Definido bajo el estándar ISO/IEC 14496 finalizó en 1998 y fue publicado en 1999, ese mismo año fue actualizado y publicada una nueva versión en el año 2000. A diferencia de los estándares previos, enfocados primordialmente en la codificación y reproducción de contenido multimedia, MPEG-4 ofrece un gran número de elementos tecnológicos

estandarizados que permiten la integración de los paradigmas de producción, distribución y acceso al contenido en 3 campos principales:

a. Televisión digital
b. Aplicaciones de gráficos interactivos (contenido sintetizado)
c. Multimedia interactiva (distribución y acceso al contenido en internet)

El estándar MPEG-4 puede ser descargado en su totalidad y usado de forma libre para realizar implementaciones del mismo, esto no implica que el software desarrollado este exento de respetar las patentes incluidas en el estándar, de hecho, para poder realizar negocios comerciales que tengan relación con este, se debe pagar por los derechos de uso de las tecnologías que se usen en la implementación del estándar. El grupo MPEG intenta ser en la actualidad el responsable de recaudar los pagos relacionados con este estándar, sin embargo, no existe un censo general entre los entes con derechos sobre el estándar por lo que el licenciamiento de esta tecnología se ha convertido en un proceso engorroso y extenso.

Debido a la cantidad de elementos en el estándar, se han definido los conceptos de perfil, *profile*, y nivel, *level*, que agrupan funcionalidad relacionada. Para limitar la complejidad computacional de los decodificadores los perfiles definen un grupo limitado de niveles.

Dentro de las mejoras de MPEG-4 se destaca la posibilidad de codificar canales de voz mediante tasas binarias extra bajas, usando codificadores paramétricos. Se pueden obtener tasas entre 2 a 8 Kbps. La estrategia usada se basa en la definición de un modelo de señal que se ajusta de acuerdo con la señal a codificar, de manera que la información transmitida son los parámetros del modelo que mejor lo ajustan a la señal [27].

1.2.4 MPEG-7

MPEG-7 es un estándar que permite describir la información contenida en un recurso multimedia. Para que la descripción sea posible, se parte del hecho que la información contenida en el recurso tiene un significado, en su propio contexto, y dicho significado tiene un grado de interpretación (quién, qué, cómo, cuándo, dónde, formas, colores, texturas, movimientos, sonidos). La información descriptiva que se puede reunir con MPEG-7 no está enfocada a un tipo de aplicación en particular, por el contrario, los elementos de información que estandariza MPEG-7 pueden ser usados por un amplio rango de aplicaciones en diferentes contextos.

El estándar se divide en cuatro componentes principales:

a. Lenguaje de definición de descripciones, *Description Definition Language*, DDL, el cual define las bases generales para el lenguaje de metadatos de MPEG-7.

b. Elementos descriptivos para recursos con componente audible.
c. Elementos descriptivos para recursos con componente visual.
d. Esquemas de descripción multimedia, MDS. Los descriptores para “capturar” los aspectos semánticos del contenido multimedia, por ejemplo, lugares, actores, objetos, eventos, etc.

MPEG-7 se implementa mediante esquemas XML. El conjunto de esquemas XML de MPEG-7 definen 1182 elementos, 417 atributos y 377 tipos de datos complejos. Dada la gran cantidad de elementos en el estándar, resulta difícil la administración de estos, más aun, cuando algunos elementos pueden ser definidos en diferentes niveles. Otra consecuencia de la cantidad de elementos y debido al uso de XML es que una gran parte de la semántica permanece implícita, en consecuencia, cada vez que se desarrolla una aplicación, la semántica debe ser extraída del estándar, reimplementada y procesada directamente por la aplicación [28, 29].

1.2.5 H.261

H.261 [30] (recomendación ITU-T H.261), que forma parte del grupo de estándares H.320 para comunicaciones audiovisuales. Fue diseñado para una tasa de datos múltiplo de 64 Kbit/s. Lo cual coincide con las tasas de datos ofrecidas por los servicios ISDN, *Integrated Services Digital Network*. Se pueden usar entre 1 y 30 canales ISDN (64 Kbit/s a 1920 Kbit/s). Aplicaciones que motivaron el diseño de este tipo de estándar son: videoconferencia, vigilancia y monitoreo, telemedicina, y otros servicios audiovisuales. A continuación se muestra un diagrama que resume los bloques que conforman el códec.

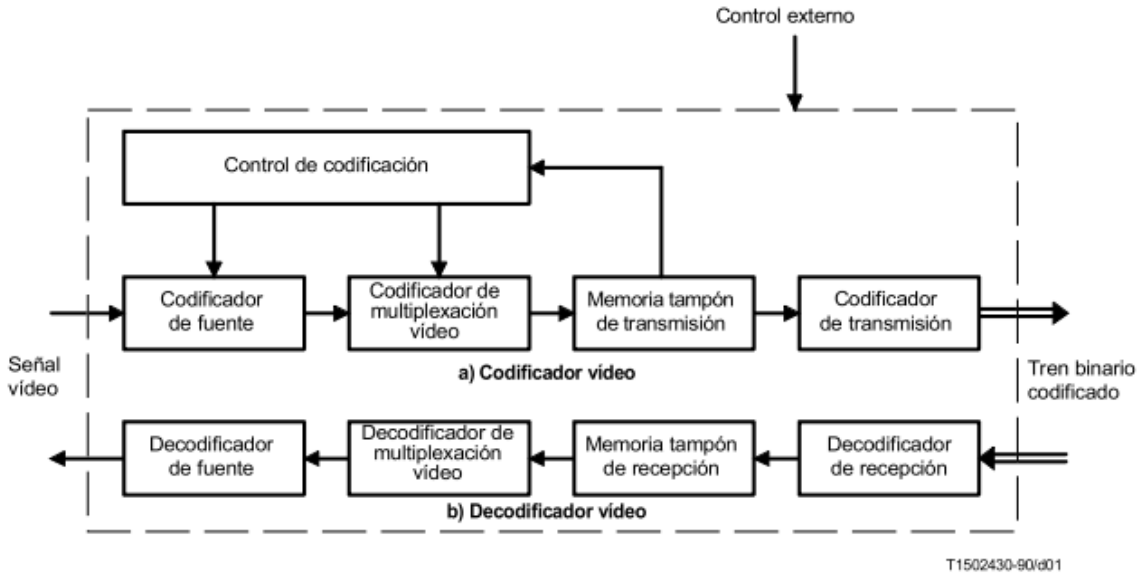

Figura 2. Diagrama de bloques resumido códec H.261.

1.2.6 H.263

H.263 [30] es un codec recomendado por la ITU-T de vídeo. Tiene una estructura de codificación basada en la recomendación H.261 pero ofrece mejor calidad de imagen a bajas tasas de bits, además ofrece cuatro modos de codificación avanzada que pueden ser negociados por los nodos durante un proceso de transmisión. Los modos opcionales permiten a los desarrolladores decidir entre un proceso de codificación complejo u otro que garantice rendimiento. Una explicación detallada de los modos de codificación opcionales puede encontrarse en [31, 32] y . El gráfico siguiente muestra un resumen de los bloques que conforman este códec.

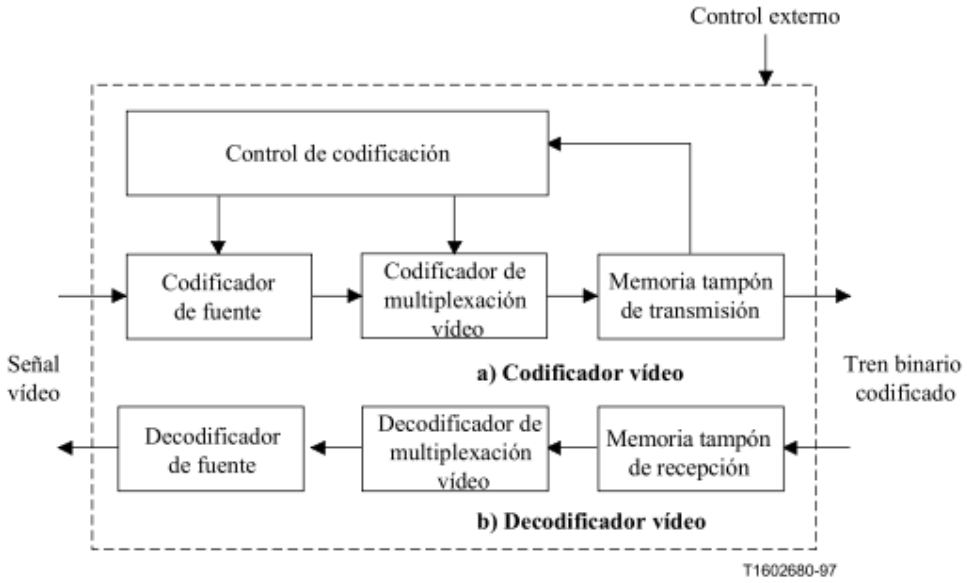

Figura 3. Diagrama de bloques resumido códec H.263.

1.2.7 H.264

Esta recomendación se desarrolla en conjunto con el grupo MPEG, para el desarrollo del un códec basado en la compensación de movimiento orientado en bloques, publicado en el 2003 bajo el nombre de H.264/MPEG-4 AVC, la cual actualizó el componente visual del estándar MPEG-4 (*MPEG-4 Visual* o *MPEG-4 Part 2*) [33].

1.2.8 WMV, Windows Media Video

Constituye un formato de video para un conjunto de codecs propietarios desarrollados por Microsoft [34]. Originalmente se trataba de un único códec orientado al *streaming* de video en internet, posteriormente se desarrollaron nuevos codecs especializados basados en

implementaciones propietarias de estándares como MPEG-4 por lo cual el acrónimo WMV se usa para distinguir al grupo de códecs en su totalidad.

Tradicionalmente los recursos multimedia que usaban tecnología WMV, tenían la extensión ASF, *Advanced Systems Format*, el cual se define como un formato contenedor que puede manejar diferentes canales, por ejemplo para transmitir múltiples flujos de información de video de diferentes tasas de transmisión. Sin embargo para facilitar la identificación y organización de los recursos Microsoft decidió establecer dos nuevas extensiones WMA y WMV [35], las cuales representan recursos audio y de video respectivamente.

Existen cuatro líneas destacadas bajo las cuales se organizan las tecnologías WMV

a. Pre – DMO, *DirectX Media Object*, basados en códecs de video como MPEG-4 y otros, que no son compatibles con dichos estándares.
b. Windows Media Video 7 (WMV1). Códecs basados en DMO, los cuales definen interfaces estandarizadas que permiten interoperabilidad entre sistemas y reutilización de componentes. Cumple con el perfil simple del estándar MPEG-4.
c. Windows Media Video 8 (WMV2). Códecs basados en DMO.
d. Windows Media Video 9 (WMV3 - VC-1). Al igual que los anteriores está basado en DMO, en el año 2003 se presenta como estándar ante la sociedad de ingenieros de cine y televisión [36, 37], *SMPTE*, en el año 2006 fue establecido como estándar internacional, manteniendo la línea de licenciamiento por los derechos de autor a Microsoft.

En la actualidad Microsoft promueve el uso de una versión de alta definición, denominada WMV HD, que se basa en VC-1.

1.2.9 RealMedia

Se trata de un formato contenedor que agrupa los codecs de audio y video, *RealAudio* y *RealVideo* respectivamente, desarrollados por *RealNetworks* en 1997. Vigente hasta el momento con su versión 10, publicada en el 2008. El formato RealMedia, caracterizado por los archivos con extensión .rm y .rmvb se enfocan a la reproducción usando tecnologías de *streaming*.

Los codecs definidos por esta tecnología se identifican bajo los códigos rv10, rv13, incluidos en la versión 1.0, rv20, desde la versión 6.0, rv30, desde la versión 8.0, rv40, incluido en la versión 9.0 y actualizado en la versión 10.0. Hasta la versión 8, los códecs de audio y video se basan en el estándar H.263, usando tasas de bits constantes, posteriormente, *RealNetwork* decide implementar sus propios códecs, los cuales se caracterizan por usar tasas de bits variables (RMVB, *RealMedia Variable Bitrate*).

Los codecs desde la versión 9.0 ofrecen los siguientes beneficios:

- La misma calidad que MPEG-4 a doble tasa de bits y que MPEG-2 a 4 veces la tasa de bits.
- 30% mejor rendimiento que la versión 8.0 y 50% mejor que la versión *RealVideo G2*.
- Casi la calidad de DVD a 500 Kbps.
- Implementaciones para multiples plataformas (Windows, Linux, MacOS, Symbian), bajo diferentes arquitecturas (ARM, Intel, AMD, etc.).
- No limita la resolución con valores mínimos ni máximos, se puede codificar usando resolución 32x32 como 1920x1080.

1.2.10 QuickTime

Desarrollado por Apple Inc., se trata de un formato contenedor, bajo la extensión .MOV, que dio origen al formato de archivo de MPEG-4 y que puede agrupar flujos de información multimedia codificados mediante diferentes códecs como MKV, 3GPP, DV, Ogg, ASF, WAV, MP3, DivX, entre muchos otros.

1.2.11 VP6

Es un códec propietario desarrollado por la empresa On2 Technologies. El códec VP6 se hizo popular con su adición al reproductor FlashPlayer 8, caracterizado por la producción de videos con una alta calidad a bajas tasas de bits, videos con mejor nitidez y mejor calidad de color, así mismo destaca un nivel de procesamiento mucho menor, para la decodificación, que estándares como MPEG-4, que necesitan de niveles de procesamiento elevado. Otra característica destacada en el códec, es la definición de un canal de información "*Alpha*", asociada a la transparencia, en tiempo real e interactivo, usado por ejemplo, en videos donde se necesita reemplazar una imagen o zona de fondo, en el contexto fílmico conocida como "fondo verde", con otro tipo de representación visual (imagen estática, video, efecto especial, etc.) [38].

En la actualidad, empresas como SUN Microsystems y eBay Company, entre otras, han optado por licenciar este tipo de códec para productos como JavaFX y Skype respectivamente, dados los buenos resultados que se evidencian de plataformas como FlashPlayer y además por la facilidad en el licenciamiento de la tecnología [39].

Al igual que el códec RealMedia, VP6, no impone limitantes de resolución para los videos, observándose excelentes resultados incluso en videos de alta definición.

La eficiencia del códec, radica en estrategias como una codificación dividida en dos fases, la primera fase se considera de análisis, donde el codificador puede identificar las

“escenas” o secciones similares del video, y a su vez, determinar qué tanto, que tan fácil o difícil dichas escenas pueden ser codificadas. De esta manera se obtiene un video con codificación inteligente que puede mantener tasas de bits elevadas, en secuencias visualmente complejas, sin la necesidad de modificar parámetros como el número de frames por segundo ni la resolución [38, 40].

ANEXO 2. Análisis del caso de estudio y recomendaciones

2.1 Introducción

Como se describió en el capítulo 4 sobre el caso de estudio, se lleva a cabo la integración de un servicio de video *streaming* dentro del LMS del Entorno Virtual de Aprendizaje, EVA, el cual incluye la distribución de video, a diferentes velocidades de transmisión, sincronización con las diapositivas asociadas a la presentación y herramientas que permiten al estudiante interactuar con el contenido, entre estas la posibilidad de agregar comentarios sobre la línea de tiempo del recurso reproducido. En esta sección se presenta información relevante relacionada con la implementación y puesta en marcha del prototipo funcional, resultados asociados a la ejecución del mismo en un ambiente real y finalmente recomendaciones asociadas al servicio de video *streaming*.

2.2 Entorno de ejecución

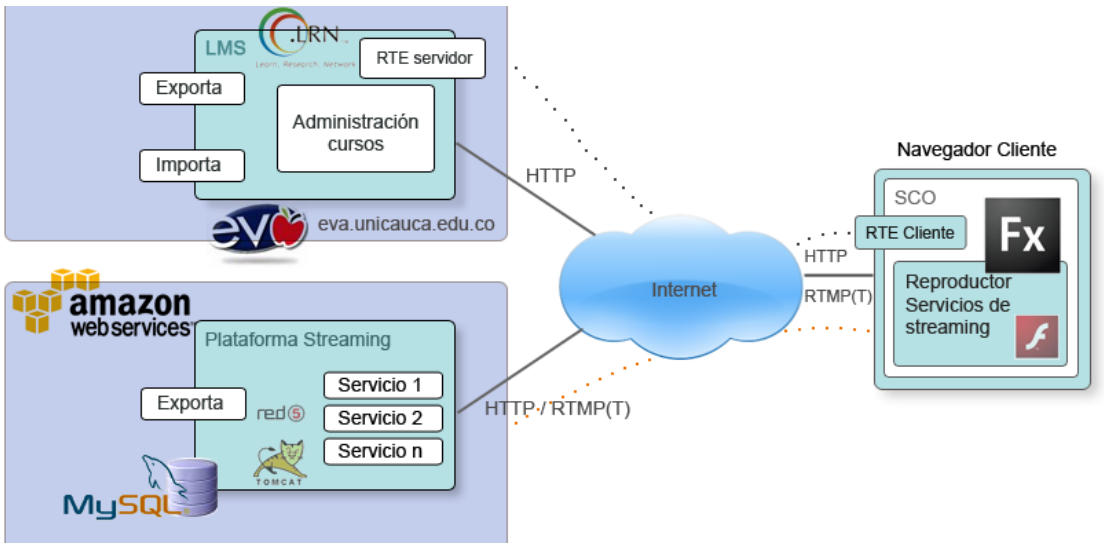

FIGURA 4. ESQUEMA DE DESPLIEGUE DE PLATAFORMA DE STREAMING

La figura 1 muestra la forma en que se desplegó la plataforma de *streaming* para la puesta en marcha del servicio de video *streaming*.

El LMS se basa en el sistema EVA que funciona actualmente dentro de la red local en la Universidad del Cauca y se accede desde el exterior mediante el nombre de dominio http://eva.unicauca.edu.co. Para esta parte de la arquitectura no fue necesario realizar ninguna modificación. Es importante destacar que para la navegación web dentro de la infraestructura de red de la Universidad del Cauca, utilizando el servicio de proxy (proxy.unicauca.edu.co:3128), se habilitan únicamente los puertos 80 (HTTP) y 443 (HTTPS) hacia el exterior, lo que limita la forma en que se consumen otro tipo de servicios, en nuestro caso, el de interés sobre el puerto 1935 (RTMP).

Para la instalación de los servidores de aplicación (Tomcat 6), de streaming (Red5) y el motor de bases de datos (MySQL) se utiliza la infraestructura ofrecida por AWS, *Amazon Web Services,* bajo el concepto de *Cloud Computing*, o computación en la nube, que permite ofrecer los servicios desde internet.

A continuación se describe en detalle las versiones de los componentes utilizados para la plataforma de *streaming*:

Aplicación	Uso	Versión
		Entornos de ejecución
Linux	Sistema operativo	Ubuntu Server 9.04
Tomcat	Servidor de aplicaciones, usado para desplegar las aplicaciones web y las *gateways* asociadas a los protocolos RTMP y RTMPT	6.0
Red5	Servidor de streaming de recursos multimedia y se aplicaciones tipo RPC, SharedObjects y Push.	0.8 final.
MySQL	Servidor de bases de datos	5.1.37
Flash Player	Entorno de ejecución, del lado del cliente, para el consumo de servicios de streaming.	9.0.115.0 mínima
		Lenguajes de programación
Java	Lenguaje de programación de aplicaciones web y servicios de streaming.	OpenJDK 1.6
Flex	Lenguaje de programación para el desarrollo de la aplicación de consumo de servicios de streaming.	3.0

		Librerías o *framework*
Spring	Herramientas para la inyección de dependencias, usada en lado del servidor para las aplicaciones basadas en Java y en el cliente para la aplicación basada en Flex.	2.5
SCORM	Especificación para la el uso del cliente basado en Flex como objeto de aprendizaje	1.2
Hibernate	Como framework de persistencia de datos	3.2.6
JTA	Java Transaction API, especificación para el mapeo objeto-relacional de la capa de datos y gestión de transacciones, usando el estándar EJB 3.0.	Atomikos 3.5.5

TABLA 1. VERSIONES DE COMPONENTES USADOS EN LA IMPLEMENTACIÓN DEL PROTOTIPO DE VIDEO STREAMING

2.2.1 Instancia m1.small y c1.medium de AWS

La infraestructura de *cloud computing* de AWS especifica varios tipos de "instancias" [41] que definen un conjunto de características computacionales particulares, enfocadas a diferentes tipos de aplicaciones. Las instancias de la familia "*Standard Instances*", donde se incluye la m1.small, son enfocadas a todo tipo de aplicaciones con requerimientos de procesamiento, RAM y procesos de I/O, *input/output* generales, las instancias de la familia "*High-CPU Instances*" son útiles para aplicaciones que requieren un nivel de procesamiento mayor en relación con la RAM y los procesos de I/O, en esta se incluye la versión c1.medium. La siguiente tabla describe en términos generales las características de las instancias utilizadas en las pruebas del servicio.

	Instancia m1.small
RAM	1.7 Gbytes
Arquitectura	32 Bits
Almacenamiento	160 Gbytes por defecto
Rendimiento I/O	Moderado
Unidad de computo [42]	1 EC2, corresponde a un núcleo de un procesador virtualizado a 2.6 Ghz. [43]
Interfaz de red	100Mbps

TABLA 2. CARACTERÍSTICAS INSTANCIA M1.SMALL

	Instancia c1.medium
RAM	1.7 Gbytes
Arquitectura	32 Bits
Almacenamiento	350 Gbytes por defecto
Rendimiento I/O	Moderado
Unidad de computo	5 EC2, corresponde a dos núcleos reales a 2.4 Ghz [43]
Interfaz de red	100 Mbps

TABLA 3. CARACTERÍSTICAS INSTANCIA C1.MEDIUM

Con las características de las instancias descritas se realizan las siguientes pruebas de estrés:

2.2.2 Limite de conexiones simultáneas

Para las 2 instancias se realizaron pruebas que determinan en promedio el límite de conexiones simultáneas hacia el servidor de streaming (Red5). La herramienta [44] que permite realizar peticiones se ejecuta sobre un navegador con soporte Flash. Para minimizar el error se realizan un total de 30 muestras, variando el navegador, sistema operativo del cliente, la velocidad a la que se realizan las peticiones y la cantidad de clientes que realizan peticiones.

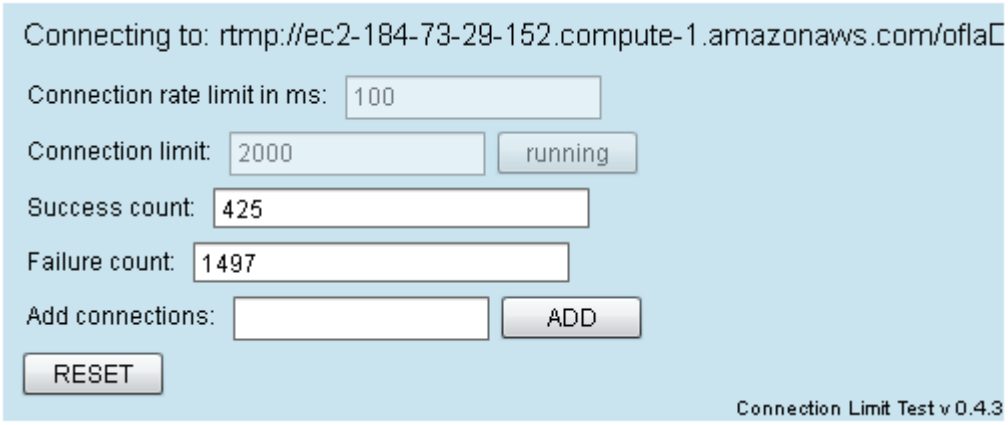

FIGURA 5. RESULTADO LIMITE DE CONEXIONES CLIENTE 1.

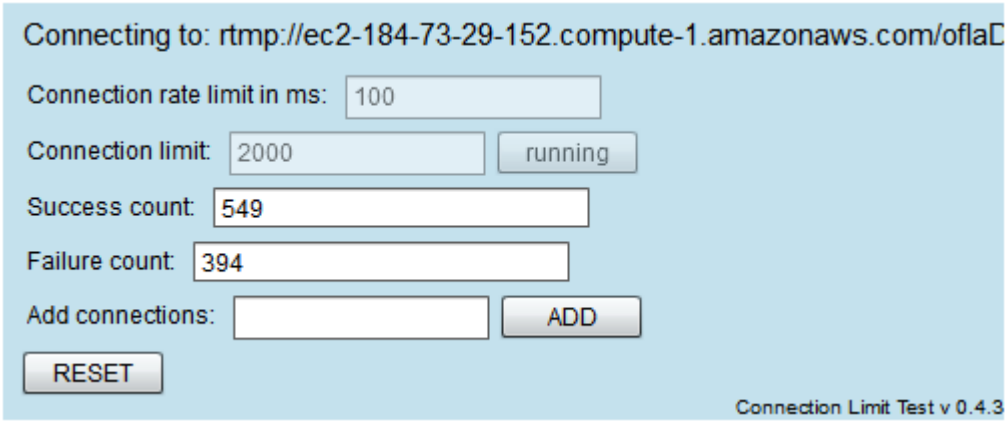

FIGURA 6. RESULTADO LIMITE DE CONEXIONES CLIENTE 2.

Las figuras 2 y 3 muestran dos clientes de prueba que suman un total de 984 conexiones activas sobre el servidor. Dicho valor es la media de las muestras con una desviación estándar de 10.11.

2.2.3 Conexiones concurrentes con consumo de servicio de video *streaming*.

Para cada una de las instancias se evaluó la cantidad de conexiones simultáneas con transmisión de video. Este video se caracteriza por estar codificado bajo el modo de velocidad de transmisión CBR, *Constant Bit Rate*, que trata mantener una velocidad de transmisión constante. Para la prueba se utilizo un video codificado con el códec VP6 a una tasa de bits de 275Kbps, el ancho de banda requerido para la reproducción del video de forma continua oscila entre los 30.5 KB/s y 40.8KB/s.

Dado que no se cuenta con la infraestructura necesaria para consumir el servicio de video *streaming* hasta que el servidor colapse, se requiere hacer uso de una herramienta que soporte el protocolo RMPT para realizar peticiones de este tipo. Con la ayuda de Andrei Sochirca, representante de la empresa Smaxe Ltd, se pudo hacer uso de la herramienta comercial JUV RTMP loadTester [45], para realizar la prueba de concurrencia, limitada únicamente al uso del protocolo RTMP. Para efectuar la prueba de la forma más real posible se inicio una nueva instancia m1.small sobre AWS desde donde saldrían todas las peticiones hacia el servidor de *streaming*, para garantizar que las peticiones se hicieran por las interfaces de red que se comunican con internet y no mediante la red interna de AWS se alquiló una dirección IP Real pública y se asignó al servidor de *streaming.*

Para determinar la calidad de la reproducción se iniciaron dos clientes que consumen el servicio de forma manual, denominados clientes de control, luego se procedió a ejecutar

la herramienta que simula conexiones hacia el servidor y solicita el recurso multimedia. La figura 4 muestra la forma en que se organizan los clientes y la herramienta de simulación.

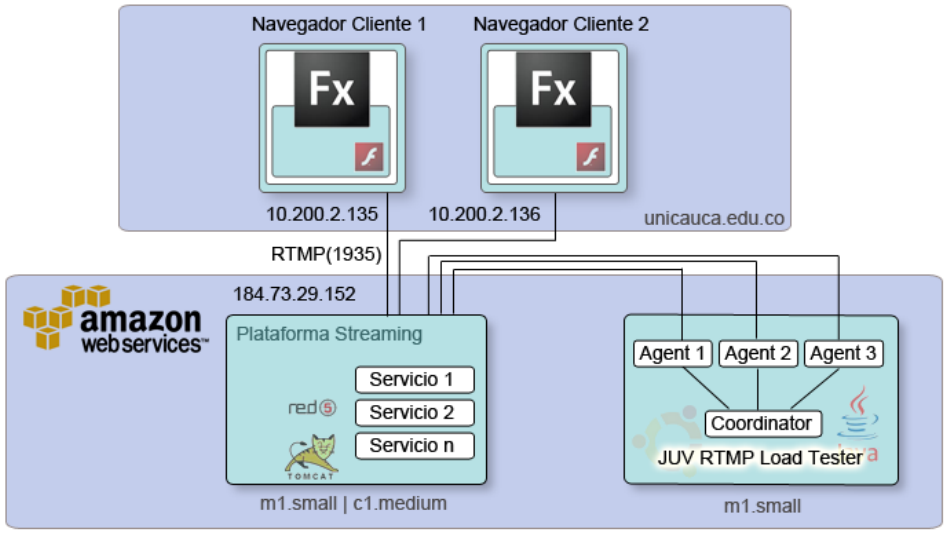

FIGURA 7. ORGANIZACIÓN DE COMPONENTES PRUEBA DE CONCURRENCIA Y CONSUMO DE SERVICIO.

La herramienta de simulación tiene dos componentes principales, el *coordinator* y el *agent*. El componente *agent* se encarga de crear procesos o hilos que se conectan hacia el servidor de streaming a pedido del *coordinator*, este último se encarga de administrar los *agent*, consultar su estado y solicitar la creación de nuevas conexiones. Para la prueba se crearon tres *agent* y su correspondiente *coordinator.* Para determinar la carga en el servidor se hace uso de las herramientas "TOP" e "IPTRAF", las cuales en nuestra evaluación resultan ser las más "livianas" en cuanto al uso de procesamiento y RAM, para minimizar el impacto en las medidas, los resultados se presentan en la tabla 4 y 5.

m1.small	
Conexiones concurrentes	82
Nivel de procesamiento del proceso	72% us[1]
Incremento en consumo de RAM	60.3 MBytes
Ancho de banda consumido	75,34 Mbps aprox

TABLA 4. RESULTADOS DE PROCESAMIENTO PARA INSTANCIA M1.SMALL

c1.medium	
Conexiones concurrentes	204
Nivel de procesamiento del proceso	52.4% us

[1] us, *User CPU Time*: en Linux se diferencian dos medidas de procesamiento, asociadas al usuario y al sistema. "us" representa el tiempo de procesamiento asignado a las tareas que realiza una aplicación propia del usuario y que no son del sistema o llamados del kernel.

Incremento en consumo de RAM	39,99 MBytes
Ancho de banda consumido	77,64 Mbps

TABLA 5. RESULTADOS DE PROCESAMIENTO PARA INSTANCIA C1.MEDIUM

Antes de presentar la explicar la información de las tablas 4 y 5 se debe mencionar que la herramienta JUV RTMP LoadTester, no realiza control de ancho de banda sobre la conexión RTMP, esto quiere decir que cuando solicita el recurso multimedia al servidor lo hace tratando de consumir la mayor cantidad de ancho de banda posible, a diferencia de un cliente sobre la plataforma Flash que solo usa lo necesario del canal.

Durante las pruebas sobre la instancia m1.small se obtuvieron más conexiones, sin embargo cuando las conexiones concurrentes superaban las 81, los videos reproducidos en los clientes de control se interrumpían. Las 80 conexiones generadas por la herramienta de simulación consumían en promedio 120KBytes/segundo del ancho de banda, 3.5 veces más que el promedio de velocidad del recurso multimedia.

Desde el punto de vista del ancho de banda cada conexión simulada podría representar en la realidad 3 conexiones con clientes reales. Sin embargo el hecho que los videos de control se interrumpan cuando el valor de conexiones sobrepasa 80 puede significar que dadas las características de I/O y nivel de procesamiento no sean suficientes para manejar los procesos asociados al control de nuevas conexiones, pero esto solamente son presunciones ya que los valores de I/O y procesamiento de las instancias son sugeridos dentro y por la comunidad de usuarios de AWS.

Sobre la instancia c1.medium se establecieron en promedio 202 conexiones desde la herramienta de simulación, cada una consumiendo en promedio 393.58 Kbps o 49.20 KBytes/segundo, lo que significa en términos de ancho de banda que por cada 7 conexiones simuladas pueden establecerse 10 conexiones reales mediante un cliente basado en Flash. A diferencia de la prueba sobre la instancia m1.small las 202 conexiones fueren el límite debido a la herramienta de simulación, que por algún motivo no incrementaba la cantidad de conexiones. Durante esta prueba los clientes de control nunca se vieron afectados por la cantidad de conexiones concurrentes, además se pudo comprobar manualmente que el servidor seguía soportando más clientes basados en Flash.

2.2.4 Evaluación RTMP vs RTMPT

Las figuras 5 y 6 muestran el tráfico durante un mismo periodo de tiempo (3 minutos y 36 segundos), de un video codificado en modo CBR a un tasa de bits promedio de 220 Kbps y duración de 2 minutos 30 segundos, mediante los protocolos RTMP y RTMPT, .

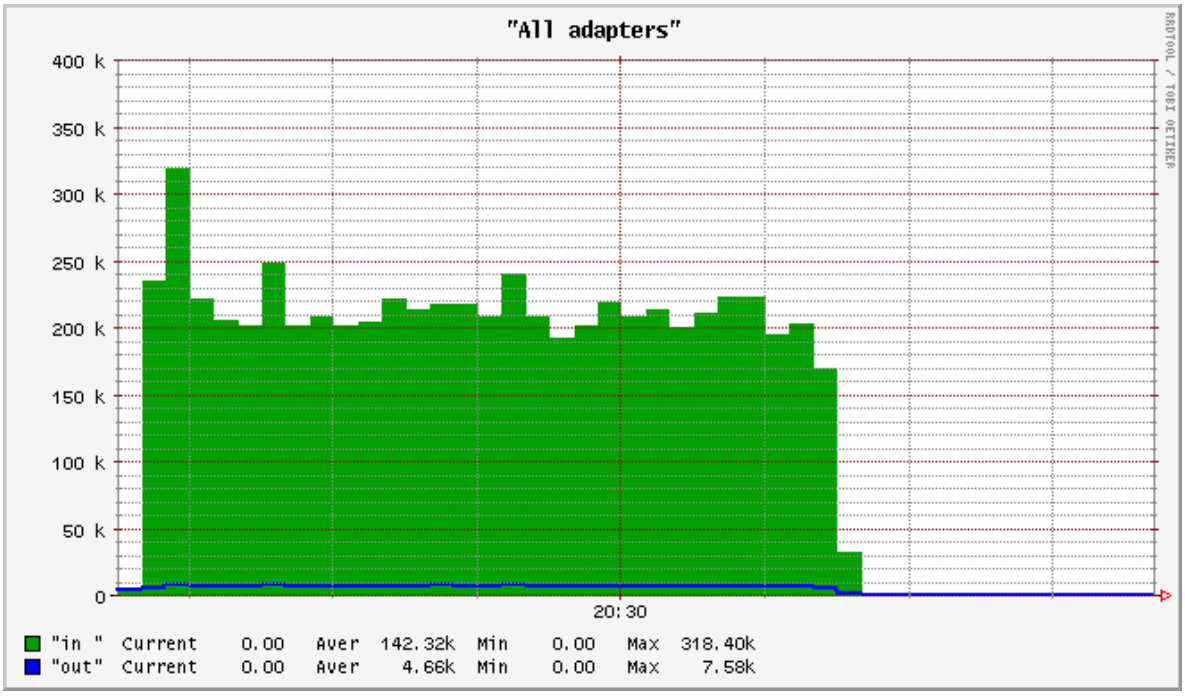

FIGURA 8. PROTOCOLO RTMP EN UN PERIODO DE TRES MINUTOS Y 36 SEGUNDOS, BW(KBPS) VS TIEMPO(S)

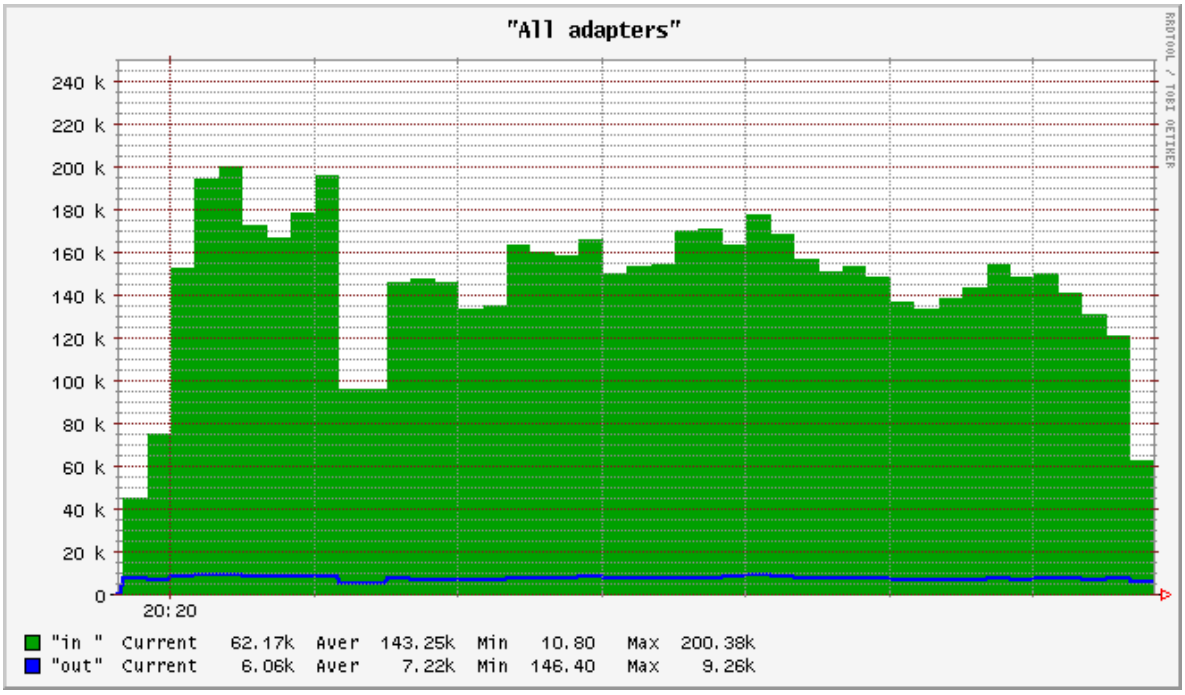

FIGURA 9. PROTOCOLO RTMPT EN UN PERIODO DE TRES MINUTOS Y 36 SEGUNDOS, BW(KBPS) VS TIEMPO(S)

En la figura 5 el protocolo RTMP sigue fielmente la tasa de bits promedio del recurso audio visual y las características de tiempo real del protocolo se verifican desde el momento en que se inicia la recepción de los datos y cuando finaliza, los cuales corresponden con la duración total del video.

En la figura 6 donde se presenta el tráfico mediante el protocolo RTMPT se evidencia una mayor duración de la sesión de reproducción del video, lo cual se refleja en pausas debido a la necesidad de recargar el buffer. De igual forma se observa que la duración de la sesión de transmisión y en consecuencia de reproducción se extiende por 1 minuto y 6 segundos. Aunque no existe especificación oficial pública del protocolo RTMPT, la dinámica se basa en el uso de peticiones tipo POST del estándar HTTP 1.1, de tal forma que el cliente realiza peticiones continuas al servidor con el objetivo de verificar si existe nueva información en la cola donde se agrupan los paquetes de datos asociados al video. Teóricamente el ciclo de peticiones tipo POST se realiza bajo un parámetro de tiempo secuencial, esto significa que al inicio de la sesión de transmisión se realizan varias peticiones y a medida que transcurre el tiempo se reduce el número de transmisiones hasta un máximo de 2 por segundo, o lo que es lo mismo peticiones cada 500ms. La figura 7 muestra el tiempo entre peticiones tipo POST promedio de 300ms, esto implica que los paquetes contienen menos información útil asociada al video en cada segundo de transmisión.

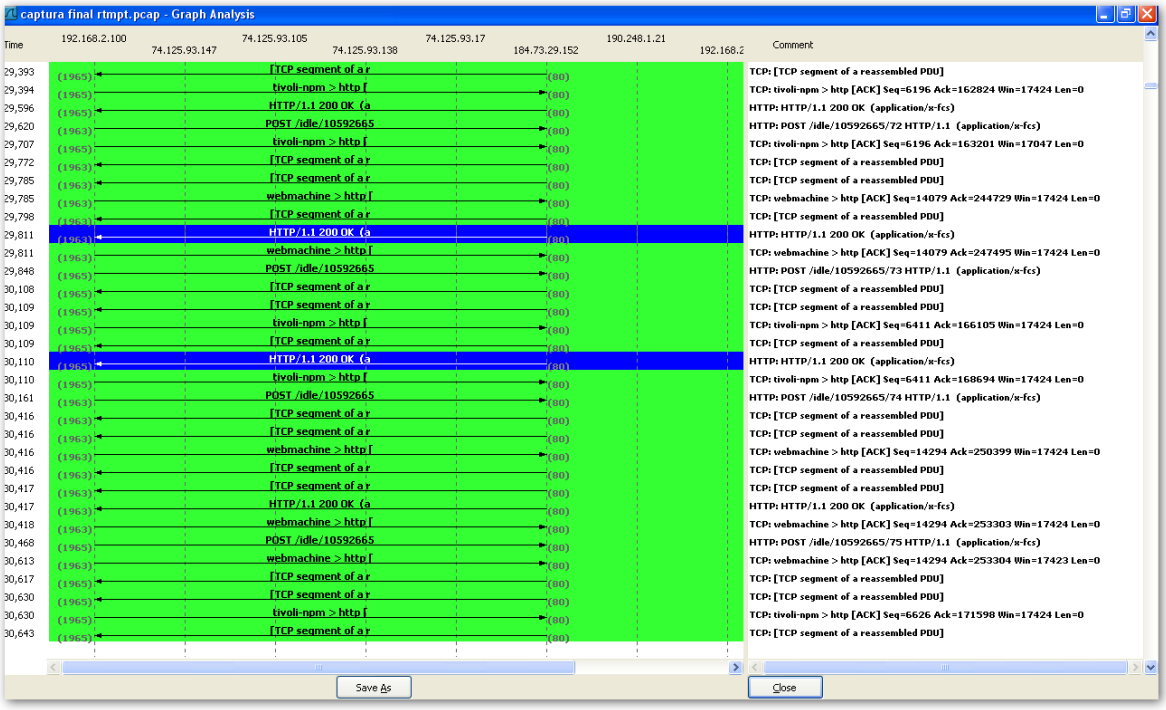

FIGURA 10. DIFERENCIA DE TIEMPO ENTRE PETICIONES TIPO POST PARA RTMPT.

Se debe aclarar que el efecto de las peticiones POST sobre la reproducción no continúa del recurso multimedia, fue evidente en conexiones que presentan latencias de alrededor de 200ms con el servidor de streaming, lo que muestra una relación proporcional inversa con este parámetro.

2.3 Herramientas recomendadas para la creación de contenido para el servicio de video streaming

Las siguientes recomendaciones se enfocan en la creación de los recursos multimedia utilizados por la plataforma de *streaming*, específicamente en la creación de las presentaciones y los videos en los formatos requeridos usando en lo posible herramientas de software libre o gratis.

2.3.1 Diapositivas en formato PDF

Para sistemas operativos Linux se cuenta con Open Office [46], una suite de herramientas ofimáticas basadas en software libre que desde su versión 3 exporta todo tipo de documentos a PDF de forma nativa.

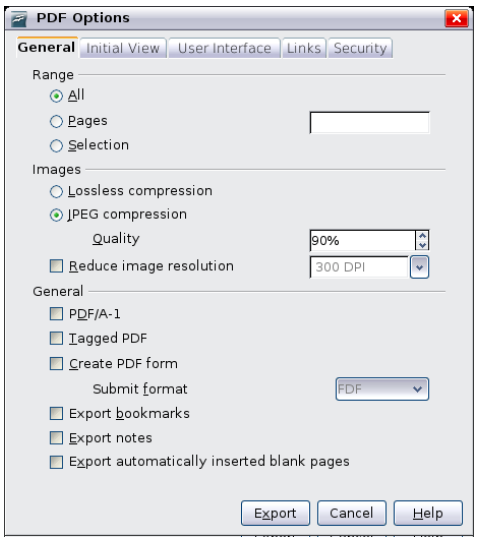

FIGURA 11. GUI PARA EXPORTAR A PDF EN OPENOFFICE 3.

Para sistemas operativos Windows se recomienda el uso de la herramienta doPDF [47], la cual instala una impresora virtual que permite que cualquier documento que se envié a imprimir, desde cualquier aplicación, se almacene como un documento PDF, la figura muestra parte del interfaz de usuario.

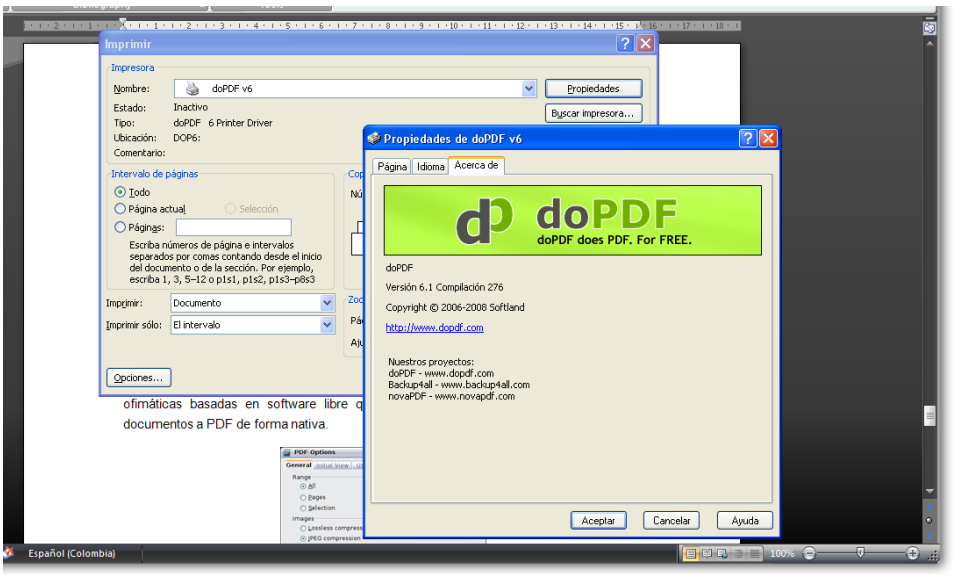

FIGURA 12. GUI DOPDF PARA CREACIÓN DE PDF EN WINDOWS.

2.3.2 Videos en formato FLV

El formato FLV es un tipo de archivo contenedor que agrupa flujos de audio y de video, para el video básicamente existen dos códecs principales Sorenson [48] y los basados en tecnologías On2 (VP3, VP5 y VP6) [39], en general la calidad de los videos basados en VP6 ofrece mejor calidad para videos a bajas tasas de transmisión. Para la creación de los videos se verificaron las herramientas Avanti [49] y Super@ [50], de las cuales se recomienda el uso de Super@ debido a su simplicidad a la hora de crear videos, además de permitir modificar la mayor cantidad de parámetros asociados a la codificación de los videos.

2.3.2.1 Super@

Se trata de una aplicación para sistemas operativos Windows que facilita el uso de las herramientas FFmpeg y MEncoder, las cuales son muy populares dentro de la comunidad de software libre para la edición de audio y video. La figura 10 muestra la interfaz de usuario principal de la herramienta.

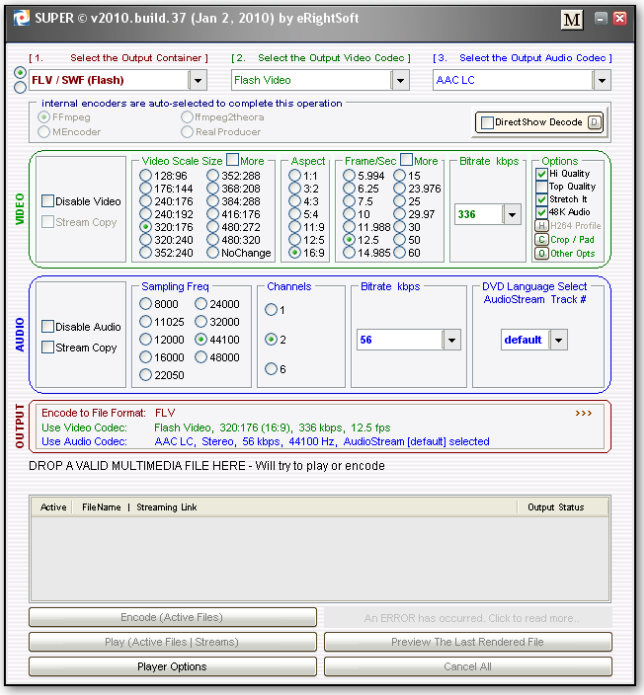

FIGURA 13. GUI PRINCIPAL SUPER@

Basados en esta herramienta se recomienda para cada servicio de video streaming generar como mínimo tres recursos que permitan a usuarios con diferentes velocidades de acceso reproducir los recursos de forma continua. La tabla siguiente describe las los parámetros para cada tipo de recurso optimizado para su reproducción a diferentes velocidades, los parámetros de altura (*high*) y relación de aspecto (*aspect ratio*) dependen de cada recurso.

Video optimizado para velocidades de 512 kbps	
	VIDEO
Formato contenedor,*Output Container*	FLV/SWF (Flash)
Códec de video, *output video codec*	Flash Video
Ancho de video	460
Alto de video	Depende del aspecto del video (1:1, 4:3, 16:9, los más comunes)
Fotogramas por segundo, *Frame/Sec*	25 fps
Tasa de transferencia, *Bitrate*	432 kbps
	AUDIO
Codec de audio, *output audio codec*	MP3
Frecuencia muestreo, *sampling frec*	44100
Canales, channels	2

Tasa de transferencia, *Bitrate*	64 kbps

TABLA 6. PARAMETROS CODIFICACIÓN VIDEO OPTIMIZADO PARA 512 KBPS

Video optimizado para velocidades de 384 kbps	
	VIDEO
Formato contenedor,*Output Container*	FLV/SWF (Flash)
Códec de video, *output video codec*	Flash Video
Ancho de video	Desde 320 a 360
Alto de video	Depende del aspecto del video (1:1, 4:3, 16:9, los más comunes)
Fotogramas por segundo, *Frame/Sec*	Desde 15 a 17 fps
Tasa de transferencia, *Bitrate*	288 kbps
	AUDIO
Codec de audio, *output audio codec*	MP3
Frecuencia muestreo, *sampling frec*	22050
Canales, channels	1
Tasa de transferencia, *Bitrate*	32 kbps

TABLA 7. PARAMETROS CODIFICACIÓN VIDEO OPTIMIZADO PARA 384 KBPS

Solo audio optimizado para velocidades de 64 kbps	
	VIDEO
Formato contenedor,*Output Container*	FLV/SWF (Flash)
Códec de video, *output video codec*	Video deshabilitado
	AUDIO
Codec de audio, *output audio codec*	MP3
Frecuencia muestreo, *sampling frec*	44100
Canales, channels	2
Tasa de transferencia, *Bitrate*	56 kbps

TABLA 8. PARAMETROS CODIFICACIÓN SOLAMENTE AUDIO OPTIMIZADO PARA 64 KBPS

ANEXO 3. Referencia para la evaluación de costos en la creación de recursos multimedia.

En el capítulo 2 se describieron los modelos conceptuales que permiten caracterizar los servicios de streaming, específicamente el modelo de selección de tecnologías, describe una etapa asociada a la evaluación de costos, en los cuales se tienen en cuenta los costos relacionados con la producción del material usado en la actividad de aprendizaje. Este apartado tiene como propósito presentar al lector una referencia que puede servir de ayuda en la determinación de estos valores.

Dado que en nuestro ambiente no se cuenta con la experiencia suficiente en esta tarea, se toma como referencia el trabajo presentado por Matt Lobel quien demuestra una vasta experiencia en esta área. Los costos sugeridos por el autor fueron modificados en algunos casos para reflejar su valor real en los sectores laborales relacionados, con la ayuda de personas especializadas (Diseñador grafico y gerente de empresa de diseño).

3.1 Datos importantes del autor

Matt Lobel, presidente de la compañía Sparrow InterActive [51], 16 años de experiencia en el desarrollo y provisión de recursos multimedia y ofertas de recursos relacionados. Cuenta con una base de 500 clientes y se encuentra muy involucrado con la industria tanto como vendedor outsourcing de servicios relacionados con el *e-Learning* como consultor.

3.2 Análisis de tipos de recursos multimedia

Se presenta algunos tipos de recursos, realizando una breve descripción de cada uno de ellos, en las graficas se muestra en general los costos de implementación y uso de cada uno de éstos.

3.2.1 Producción de fotografías propias

Los diseñadores de las actividades de aprendizaje crean sus propios recursos fotográficos

Beneficios

- Rápido cambio de fotografías.
- Control completo sobre imágenes finales.

Consideraciones

- Costoso, debido a equipos y herramientas de post-producción o retoque.
- Complejo de planificar y producir.

Costos

- Costos variables, dependiendo de la ubicación, herramientas y duración de las sesiones fotográficas.
- Desde $100.000 hasta $3.000.000 por día en tomas.

Usos recomendados

Idóneo para el desarrollo de *e-Learning* dónde una imagen o estilo específico es requerido y los presupuestos son flexibles. El *e-Learning* basado en escenarios [52, 53] puede aprovechar efectivamente este tipo de recurso.

Ciclo de producción

- Lista de producción.
- Reunir al equipo de trabajo.
- Casting.
- Locación segura.
- Selección de fotos.
- Retoque de fotos.
- Producción.

Resultado

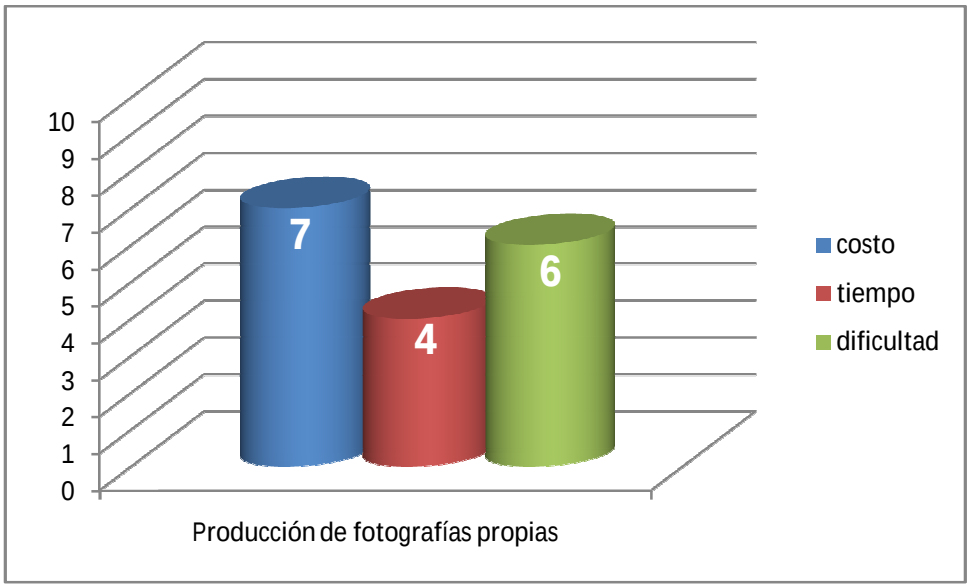

FIGURA 14. RESULTADO DE PRODUCCIÓN DE FOTOGRAFÍAS PROPIAS.

3.2.2 Fotografías de *stock*

Hace referencia a la adquisición de este tipo de recursos gráficos a través de terceros dedicados a este trabajo, por ejemplo desde ShutterStock.com o PhotoStock.com, etc.

Beneficios

- Fácil de encontrar.
- Costo medio: $4000 a $20000 pesos colombianos por imagen.
- Generalmente se paga por la licencia una sola vez.
- Gran variedad temática de imágenes.

Consideraciones

Lo que se observa es lo que se compra, sin embargo puede tomar mucho tiempo la búsqueda de imágenes apropiadas.

Costos

Entre $2000 y $20000 por imagen, las libertades sobre los derechos de autor varían.

Usos recomendados

Apropiado para cualquier tipo de desarrollo *e-Learning*, pero es altamente efectivo para el desarrollo de *e-Learning* rápido.

Ciclo de producción

- Especificar los requerimientos sobre la imagen.
- Refinar los resultados.
- Selección de imágenes, recortar, modificar o editar y colocar las imágenes.

Resultado

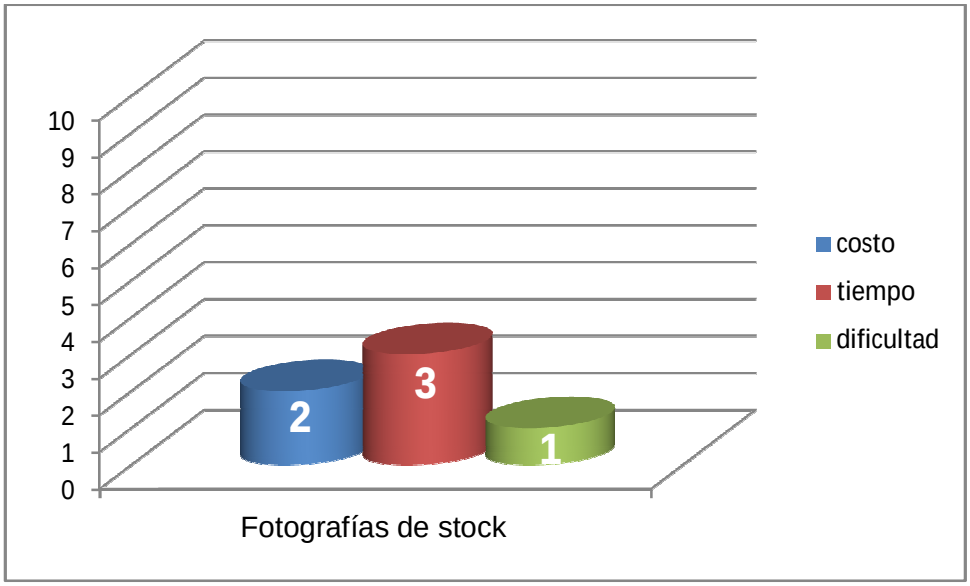

FIGURA 15. RESULTADO FOTOGRAFÍAS DE STOCK.

3.2.3 Gráficos vectorizados

Imágenes en formatos WMF, SVG, SWF, etc.

Beneficios

- Excelentes capacidades ilustrativas.
- Facilidad de producción y manipulación.
- Escalable, mantiene la calidad independiente de las dimensiones.
- El tamaño de los ficheros es liviano.

Consideraciones

- Algunas gráficas complejas requieren de un artista o diseñador.
- Uso limitado de efectos.

Costos

Costos variables, dependiendo de la complejidad, $15.000 - $50.000 por hora.

Usos recomendados

Apropiado para todo tipo de *e-Learning*. Efectivo para *e-Learning* basado en escenarios.

Ciclo de producción

- Diseño de conceptos.
- Realización de bosquejos.
- Bosquejo preliminar desarrollado y versión final.

Resultado

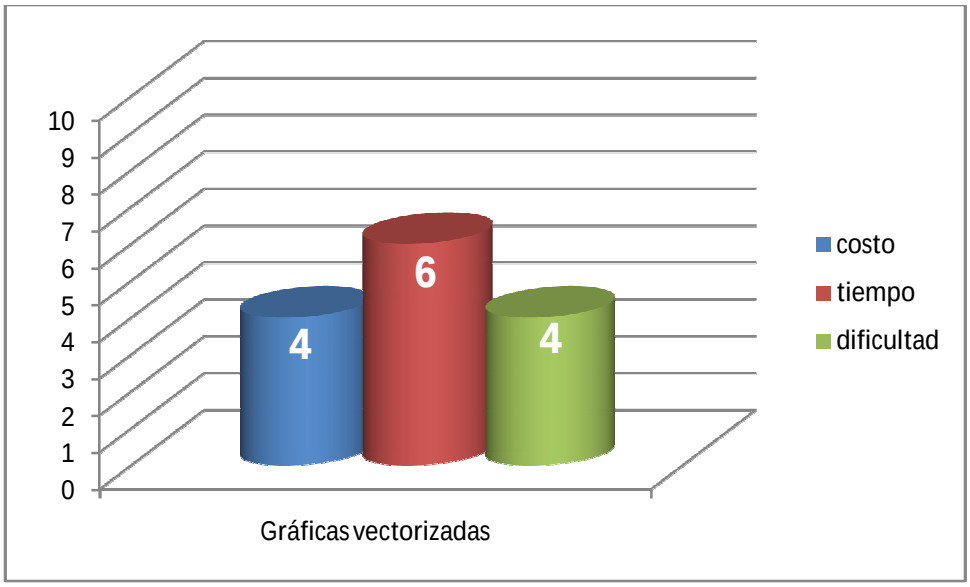

FIGURA 16. EVALUACIÓN GRAFICAS VECTORIZADAS.

3.2.4 Gráficos basados en mapas de bits

Imágenes en formatos JPG, PNG, GIF, etc.

Beneficios

- Puede ser producido por principiantes.
- Los efectos son fáciles de conseguir.
- Gran variedad de formatos y herramientas de producción.

Consideraciones

La calidad relativa de la imagen se mantiene solo al reducirla de tamaño. El incremento en la calidad es directamente proporcional al tamaño del fichero y en consecuencia es más difícil su manipulación.

Costos

Costos variables, dependiendo de la complejidad. $15.000 - $50.000 por hora.

Usos recomendados

Apropiado para todo tipo de *e-Learning*.

Ciclo de producción

- Diseño de conceptos.
- Realización de bosquejos.
- Bosquejo preliminar desarrollado y versión final.

Resultado

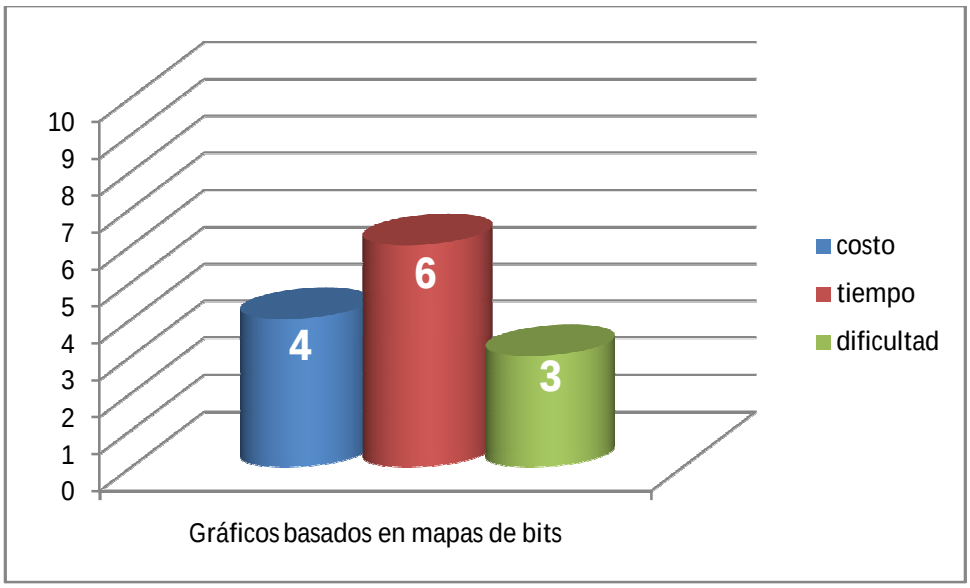

FIGURA 17. EVALUACIÓN DE GRAFICAS BASADOS EN MAPAS DE BITS

3.2.5 Video

Beneficios

- Potencial elevado.
- Agrega un toque de naturalidad a la experiencia de aprendizaje.
- Percepción de última moda o tecnología.
- Alternativa para personas discapacitadas.

Consideraciones

- Ancho de banda, ofrecer el recurso a varias velocidades de transmisión.
- Es un recurso permanente, si se requiere modificar algo asociado al contenido el proceso puede ser engorroso.
- Toma mucho tiempo.
- Costo, depende de las herramientas y la forma en que se realice la grabación.
- Continuamente aparecen nuevas herramientas hardware y software que facilitan los procesos de creación, producción, postproducción y distribución.

Costos

Alto, se debe contratar talento o preparar efectivamente al recurso humano que se involucra en el video y su producción, equipo, objetos de utilería y selección de la ubicación.

Usos recomendados

Idóneo donde el presupuesto no es un obstáculo.

Ciclo de producción

- Escena de producción.
- Casting.
- Asegurar la locación y equipo de producción.
- Edición de videos.
- Incorporación de videos en el ambiente *e-Learning*.

Resultado

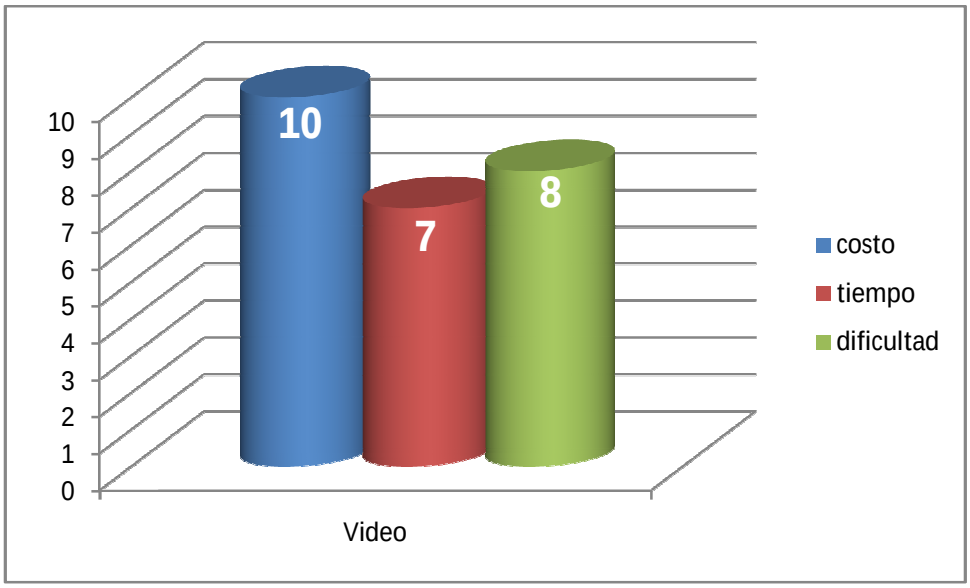

FIGURA 18. EVALUACIÓN DE VIDEO

3.2.6 Audio narrativo

Beneficios

- Trabaja bien con algunos estilos de *e-Learning*.
- Proporciona un toque de naturalidad.
- Alternativa para personas discapacitadas.

Consideraciones

- Ancho de banda.
- Es un recurso permanente.
- Toma tiempo para la producción (diseñador) y para la reproducción (estudiantes)
- Requieres de parlantes o audífonos.
- Talento en la voz.

Costos

El costo variable, depende de los métodos utilizados para la grabación del recurso.

Usos recomendados

Idóneo para todo tipo de *e-Learning* donde el contenido no es muy fluido. Herramientas para variar pausar, avanzar y retroceder sobre la reproducción del audio es recomendable, al igual que herramientas para controlar el volumen.

Ciclo de producción

- Preparativos de grabación.
- Casting.
- Grabación de audio.
- Edición.
- Inserción del audio en la plataforma de *e-Learning*.

Resultado

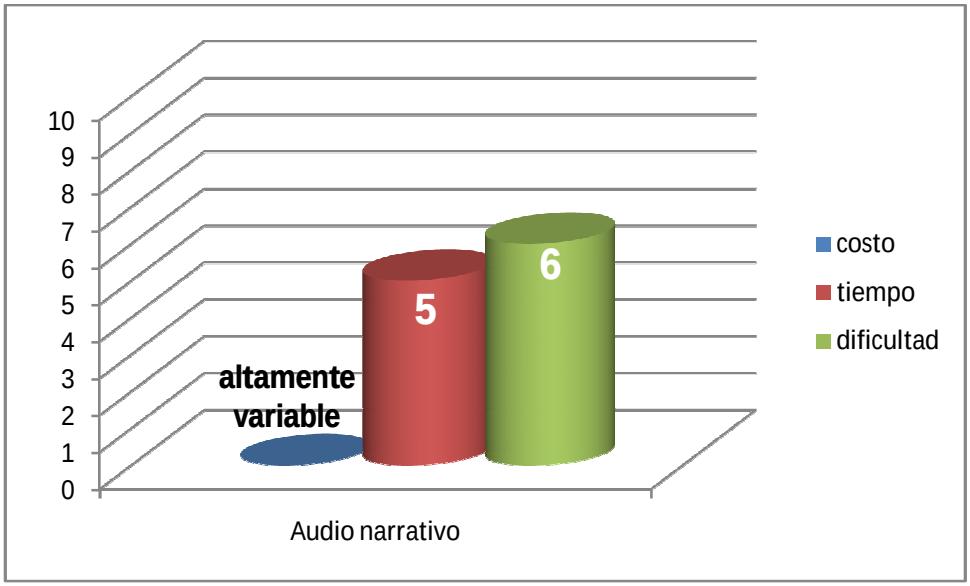

FIGURA 19- EVALUACIÓN DE AUDIO NARRATIVO

3.2.7 Audio de fondo

Beneficios

Puede amenizar o levantar el ánimo en una sesión de formación que tienda a aburrir. Este tipo de recursos son de fácil obtención o creación.

Consideraciones

El gusto en la música es muy subjetivo, puede ser impertinente para la experiencia de aprendizaje, requiere altavoces.

Costos

Bajo costo, un simple audio puede costar $2500 completa y los programas para crear música desde $100.000, un solo pago.

Usos recomendados

Solo el e-learning y las sesiones de formación que inmersa al aprendiz en una actividad basada en la experiencia (juego, laboratorio virtual, etc). Se recomienda ofrecer opciones para detener o controlar el volumen del audio.

Ciclo de producción

- Selección o creación de música.
- incorporación de música.

Resultado

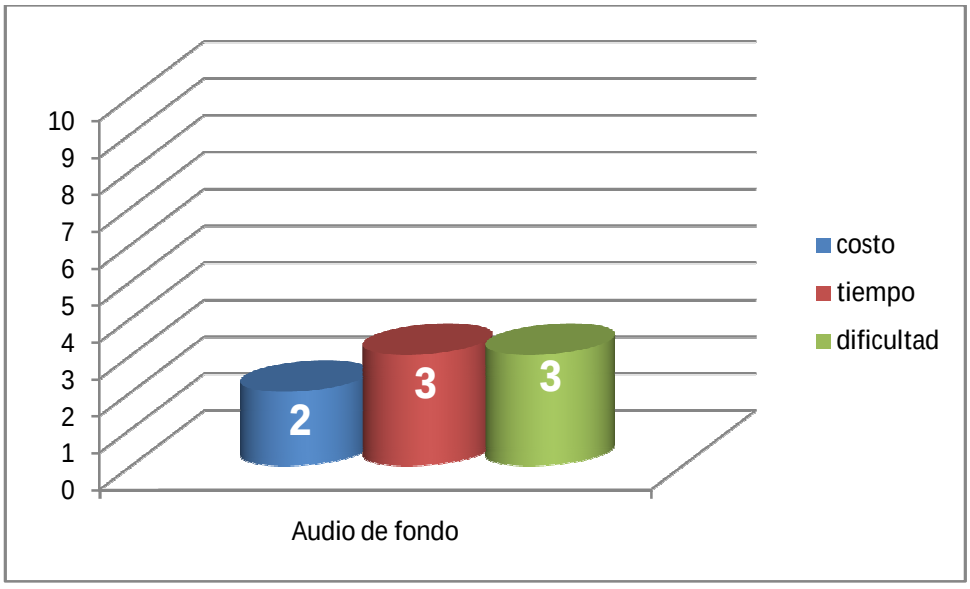

FIGURA 20. EVALUACIÓN DE AUDIO DE FONDO

3.2.8 Animación 2D

Animaciones simples, por ejemplo en formato GIF, SWF, Presentaciones PowerPoint animadas, etc.

Beneficios

- Archivos de tamaño ligero.
- Tiempos de carga rápidas.
- Manera excelente de ilustrar los conceptos.
- Gran variedad de herramientas para la creación y edición.

Consideraciones

- Puede requerir un tiempo cuantioso para crear la animación.
- El costo se incrementa con la complejidad de la animación.
- Ofrece una imagen estilo “Cartoon”.
- Puede requerir de plugins para mostrar el contenido.

Costos

Puede ser económico para animaciones de baja calidad. Las animaciones complicadas requerirán a un artista especializado.

Usos recomendados

Apropiado para cualquier tipo de *e-Learning* donde la línea de tiempo es flexible.

Ciclo de producción

- Desarrollo del guion grafico.
- Bosquejos preliminares.
- Desarrollo de objetos animados.
- Animación final.
- Integración.

Resultado

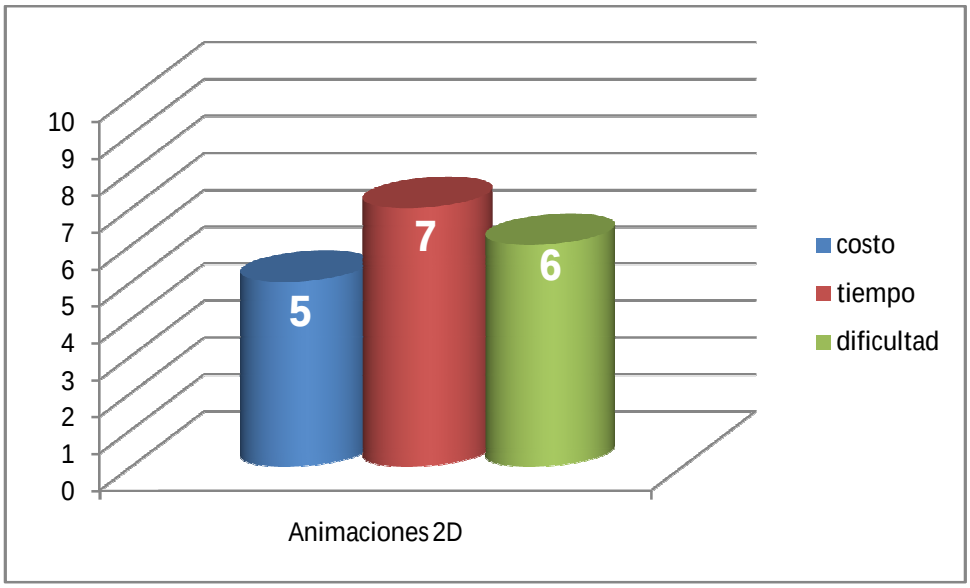

FIGURA 21. EVALUACIÓN DE ANIMACIONES 2D

3.2.9 Animación 3D

Animaciones avanzadas.

Beneficios

- Causa una percepción valiosa.
- Ilustrar efectivamente conceptos difíciles.
- Altamente realista.

Consideraciones

- Se debe considerar el ancho de banda en algunas situaciones.
- Elevado costo de producción.
- Requiere la experiencia específica en el tema.

Costos

Es una alternativa muy costosa tanto en la inversión de tiempo como en honorarios de los subcontratados.

Usos recomendados

Útil en ambientes donde el presupuesto no es un obstáculo. Idóneo para exponer simulaciones complejas y donde conceptos físicos complicados argumentan una explicación adicional.

Ciclo de producción

- Producción de la escena.
- Preparación del software.
- Bosquejos preliminares
- Modelado en 3D.
- Animación 3D.
- Generación e interpretación de los modelos 3D
- Incorporación en la plataforma de *e-Learning*.

Resultado

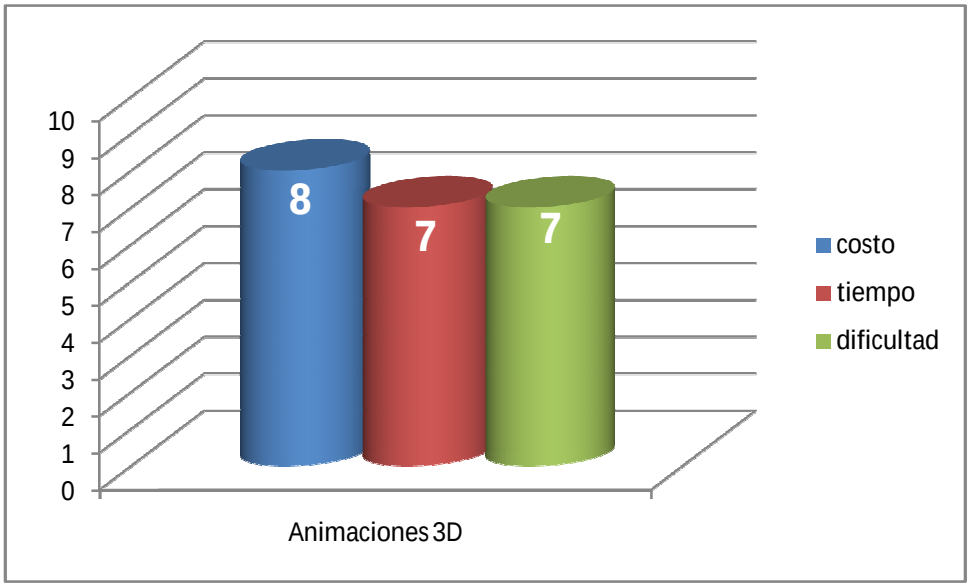

FIGURA 22. EVALUACIÓN DE ANIMACIONES 3D

ANEXO 4. Extensión de la implementación SCORM en .LRN para el soporte de nuevos modelos de datos.

En este apartado se presenta una guía general para la extensión del estándar SCORM, versión 1.2 [54], implementado dentro del LMS .LRN [55] en su versión 2.4, enfocada hacia el soporte de nuevos modelos de datos, basados en el modelo de datos CMI, Computer Managed Instruction. La guía se enfoca en describir puntualmente los cambios necesarios dentro del código fuente de la plataforma LMS, *Learning Management System*, para lograr la correcta lectura y almacenamiento de los mensajes definidos en modelos basados en el estándar CMI, transmitidos durante una sesión de aprendizaje entre el objeto de aprendizaje SCORM (SCO) y el LMS.

Antes de iniciar formalmente la guía, es necesario contextualizarnos sobre la forma en que SCORM define los procesos de ejecución y comunicación, entre el SCO y el LMS, y como es implementado este proceso dentro de .LRN.

4.1 Run-Time Environtment, RTE: El entorno de ejecución de SCORM

El objetivo de SCORM es que los recursos de aprendizaje sean reusables e interoperables entre múltiples LMS. Para que esto pueda ser posible, debe existir:

- Una forma común para iniciar recursos de aprendizaje (**Ejecución**). Este mecanismo define los procedimientos y las responsabilidades para el establecimiento de la comunicación entre el objeto de aprendizaje y el LMS. El protocolo de comunicación está estandarizado a través del uso de un API común.

- Un mecanismo común para comunicarse con el LMS (**API**). La comunicación gira en torno al estado del recurso de aprendizaje (inicializado, terminado, condición de error) y además permite obtener y fijar datos (puntajes, límites, etc.) entre el LMS y el SCO.

- Un lenguaje predefinido (vocabulario) que forme las bases de la comunicación (**Modelo de datos**). En su forma más simple, el "modelo de datos" define elementos que tanto el LMS como el SCO están esperando conocer. El LMS debe

mantener el estado de los elementos requeridos a través de sesiones, y el contenido de aprendizaje debe utilizar sólo estos elementos predefinidos para garantizar el rehúso en diversos sistemas, que cumplan con la especificación de SCORM.

4.1.1 Ejecución

Dentro del estándar SCORM se definen los recursos de aprendizaje de dos formas, los *Assets* y los SCO. Los *Assets* son recursos simples, como páginas web estáticas, un documento PDF, un grupo de imágenes, etc. Los SCO son recursos avanzados que se comunican con el LMS.

Asset: SCORM exige que un LMS inicie un Asset usando el protocolo HTTP. Dado que al asset no se le hace seguimiento, no necesita comunicarse y por lo tanto no hace uso del API ni del modelo de datos.

SCO**:** SCORM exige que:

- El LMS ejecute uno y sólo un SCO al tiempo.
- El LMS es el único que puede "inicializar" un SCO.

El LMS debe "inicializar" un SCO en una ventana o *frame* hijo de la ventana del navegador web donde se despliega la interfaz grafica del LMS, ofreciendo adicionalmente un ***adaptador de API*** de comunicación como un objeto dentro del DOM. El *adaptador de API* debe ser proporcionado por el LMS. Es responsabilidad del SCO buscar recursivamente, en niveles superiores, hasta que el *adaptador del API* sea encontrado para poder iniciar la comunicación con el LMS.

4.1.2 API

El uso de un API común satisface muchos de los requerimientos de más alto nivel de SCORM para la reusabilidad y la interoperabilidad. Proporciona un forma estandarizada a los SCO para comunicarse con los LMS, encapsulando la implementación de la capa de comunicación usada por el desarrollador. El ***adaptador de API*** es una aplicación software funcional, implementado por el LMS, que define las funciones del API y mediante interfaces las hace disponibles a los clientes SCO. Una vez la comunicación entre el SCO y el LMS se establece, el SCO puede leer y escribir información en el LMS. Toda la comunicación entre el adaptador del API y el SCO es iniciada por el SCO.

Las funciones del **adaptador de API** pueden ser agrupados en tres grupos básicos:

- **Estado de ejecución de un SCO: P**ermite iniciar y finalizar la ejecución de un SCO.
- **Gestión del estado de un SCO: U**sadas principalmente para manejar errores.
- **Transferencia de datos: U**sadas para transferir datos, obteniendo y fijando información en el LMS.

Las responsabilidades del *adaptador de API* son:

- El LMS debe lanzar el SCO en una ventana del navegador que debe ser una ventana o un *frame* hijo de la ventana principal del LMS, el cual contiene el adaptador.
- El adaptador debe ser suministrado por el LMS, implementado en cualquier lenguaje de programación.
- El único mecanismo permitido para la interacción entre el API y el SCO es a través de llamadas ECMAScript (*Javascript*).
- El adaptador de api debe ser accesible por medio del DOM [56] como un objeto llamado "API".

Lo mínimo que debe hacer un SCO es invocar los métodos de inicialización y finalización del API. Para poder hacerlo, el recurso debe poder ubicar el adaptador. Es pues deber del contenido encontrar y establecer comunicación con el adaptador.

4.1.3 Modelo de datos

El propósito de establecer un modelo de datos común se hace para asegurarse que un conjunto definido de información, relacionada con el SCO, puede ser gestionada por diferentes entornos LMS.

En la versión 1.2 de SCORM se utiliza por defecto el modelo de datos CMI, definido por la AICC [57]. Por con convención los modelos designan a sus elementos usando el patrón “Nombre_Modelo.nombre_elemento”, por ejemplo, “cmi.core.student_name”, “adl.core”.

Los elementos del modelo datos CMI se clasifican en dos grandes grupos, los obligatorios y los opcionales. Todos los elementos "obligados" deben ser implementados por todos los LMS que cumplan con el estándar SCORM.

Para cumplir el estándar SCORM 1.2, todo LMS está forzado a soportar:

- Todos los métodos y elementos del CORE (id, nombre, Ubicación de la lección, estado de la lección, etc.)
- Todos los métodos y elementos que indiquen el desempeño del estudiante (Valor que representa el rendimiento, tiempo acumulado total, indicadores sobre cómo o por que se abandona un SCO, duración de la ultima sesión, etc.)
- Los datos de suspensión y lanzamiento de un SCO.

La figura 1 muestra en resumen la forma en que se integran los conceptos anteriores para la implementación del estándar SCORM

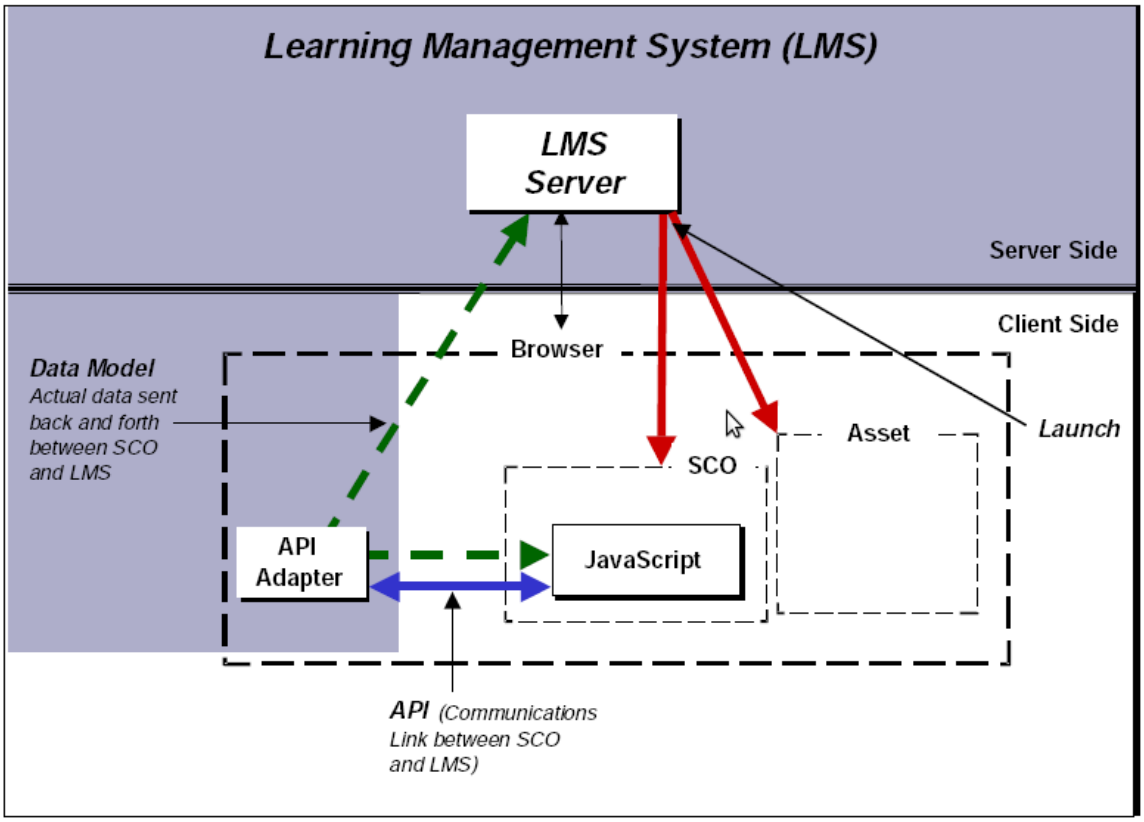

Figura 23. Vista de componentes en el estándar SCORM

4.2 RTE en .LRN

El LMS .LRN se apoya en las herramientas que ofrece el proyecto OpenACS, dentro de las cuales se encuentra el paquete *Learning Object Repository Service*, LORS [58] el cual contiene los servicios necesarios para administrar recursos de aprendizaje de diferentes estándares, entre estos SCORM, IMS y el formato propietario utilizado por Blackboard [59]. Adicionalmente .LRN requiere del paquete LORS Management, LORSM [60] para administrar los servicios del paquete LORS y efectivamente realizar las tareas de carga de recursos, creación, administración, etc.

La figura 2 muestra la forma como se estructura la implementación de SCORM, específicamente del RTE en .LRN

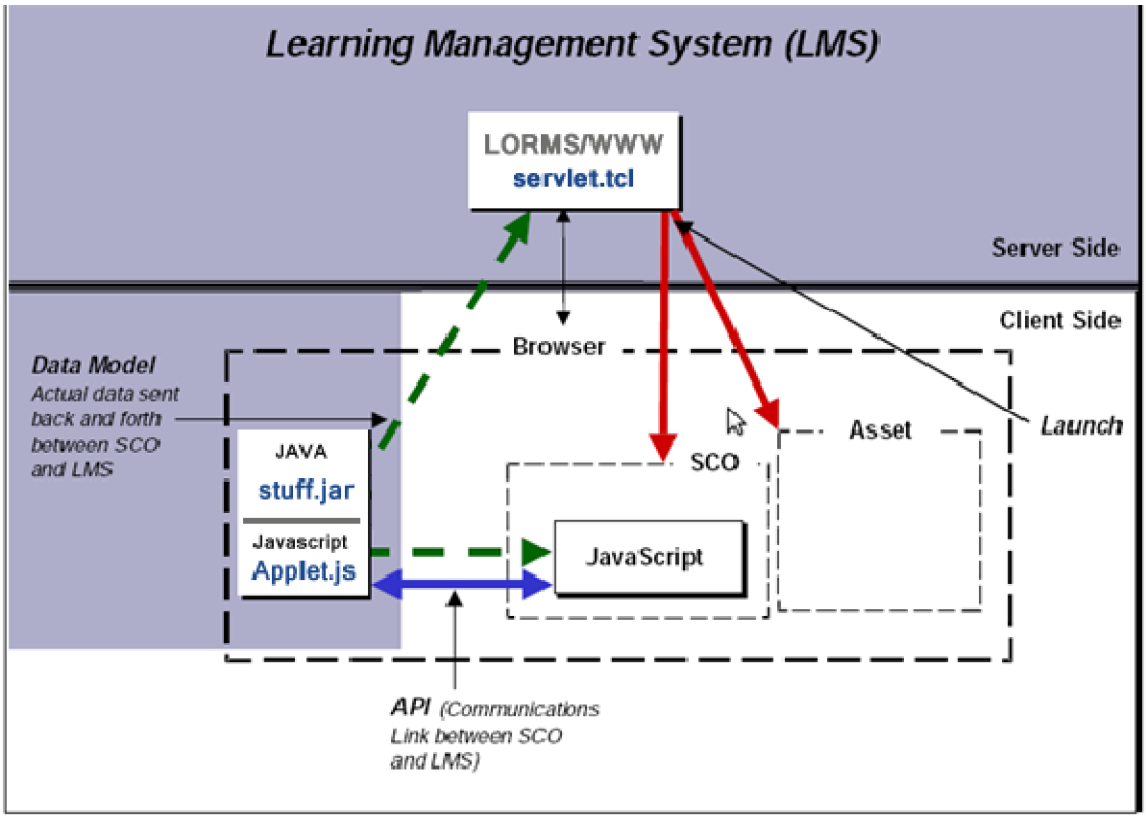

Figura 24. Implementación SCORM en .LRN

En la figura 2 se muestran los archivos principales que interactúan durante una sesión de comunicación entre el SCO y el LMS.

Applet.js: Archivo ubicado en el directorio *lorms/www/delivery* del paquete LORMS, cuando se descarga y corre en el navegador del cliente se encarga de ubicar el applet APIAdapterApplet, figura 3, y crear la instancia del adaptador de API dentro del DOM de la pagina desplegada, figura 4.

```
208     var findAPITries = 0;
209     while (APIFinder == null) {
210         debug("whithin APIFinder while loop");
211         findAPITries++;
212         APIFinder = this.document.applets[0]; // this returns object for opera and seems to be workin
213         // APIFinder = this.document.applets[1]; // this returns object for opera
214         // APIFinder = this.document.applets['APIAdapter']; // this returns function for opera
215         if (findAPITries > 70) {
216             debug("FRAMESET Couldn't find the APIAdapter ");
217                 messaging("Java JRE not found (error 223). Please check requirements.");
218                 this.releasemenu=-1;
219             break;
220         }
221     }
```

Figura 25. Applet.js busqueda del API de SCORM

```
222     if (findAPITries <=70) {
223         debug("Frameset FOUND API within applet (menu) - using APIfinder - loading learning object");
224         this.releasemenu=1;
225         messaging("System check passed");
226         //temporarly added for testing ADL conformance tests - which rely on typeof - miserably failing
227         //alert(typeof(APIFinder.LMSInitialize));
228         API=new APIHolder(); //Crea el adaptador de API, que usará el SCO
229         //exporting API to parent
230         parent.API=this.API;   //Agrega al DOM de la pagina el API
231         //alert(typeof(API));
232         debug("the type of API.LMSInitialize is "+typeof(API.LMSInitialize));
233         } else {
234         debug("API not found, don't know what to do");
235         }
```

Figura 26. Applet.js creación de la referencia 'API' en el DOM

stuff.jar: Archivo ubicado en el directorio *lorms/www/delivery* del paquete LORMS, es descargado por el navegador del cliente, mediante una petición automática al LMS, procesada por el recurso *lorms/www/delivery/applet.tcl* y *lorms/www/delivery/applet.adp,* grafico z3. En este paquete java se incluyen las clases necesarias para la correcta ejecución del *faceless applet* (Applet sin interfaz de usuario), llamado ***APIAdapterApplet***, encargado de la comunicación con el entorno de ejecución en el LMS y del procesamiento de los mensajes CMI.

```
30 <applet code="org.adl.samplerte.client.APIAdapterApplet.class" archive="stuff.jar" width=@app_width@ height=@app_height@></xmp>
31 <param name = "code" value = "org.adl.samplerte.client.APIAdapterApplet.class" >
32 <param name = "type" value="application/x-java-applet">
33 <param name = "JS" value="false">
34 <param name = "cookie" value="@cookie@">
35 <if @debuglevel@ gt 0>
36  <param name = "debug" value="true">
37 </if>
```

Figura 27. Código adp de carga del applet

servlet.tcl: Archivo ubicado en el directorio *lorms/www/delivery. S*e encarga de procesar y responder a las peticiones realizadas por el applet APIAdapterApplet, relacionadas con la inicialización y finalización del SCO, y además de procesar los llamados que buscan persistir en la base de datos del LMS algunos elementos definidos en el modelo CMI.

Cuando el adaptador del API, APIAdapterApplet solicita la inicialización del SCO, el archivo servlet.tlc, ejecuta el procedimiento llamado “cmigetcat”:

```
 96 switch -regexp $functionCalled {
 97         null -
 98         keepalive
 99         {
100                 ns_return 200 text/plain "OK"
101                 ns_log warning "SCORM Keepalive"
102         }
103         cmigetcat
104     {
105         ns_log $tracelevel "----------------------------------------"
106         ns_log $basiclevel "            LMSInitialise "
107         ns_log $tracelevel "----------------------------------------"
```

Figura 28. Bucle de control *switch*, filtro de llamado a procedimientos.

Las sentencias de código posteriores a la línea 107 se encargan de buscar en la base de datos, si existe un registro previo de un seguimiento para el SCO (en la tabla “lorsm_cmi_core”), asociado a el estudiante que accede al recurso en cuestión; en caso de no existir se crea un nuevo registro con información por defecto, de acuerdo a lo establecido en el modelo CMI definido dentro del estándar SCORM. La función finaliza con el retorno de la variable “returndata”, que almacena la información, como una simple cadena de caracteres, de los elementos del modelo de datos CMI soportados por dicha implementación, ver figura 7.

```
293         set returndata "cmi.core.student_id=$student_id,cmi.core.student_name=$name,"
294         append returndata "cmi.core.credit=$credit,cmi.core.lesson_status=$lesson_status,cmi.core.entry=$entry,"
295         append returndata "cmi.core.lesson_mode=normal,cmi.core.lesson_location=$lesson_location,"
296         append returndata "cmi.student_preference.language=italian,cmi.comments=$comments,cmi.comments_from_lms=$comments_from_lms"
297         append returndata ",cmi.suspend_data=$suspend_data,cmi.launch_data=$launch_data"
298         append returndata ",cmi.student_data.max_time_allowed=$max_time_allowed,cmi.student_data.time_limit_action=$time_limit_action"
299         append returndata ",cmi.student_data.mastery_score=$mastery_score"
300     }

        ...

325     ns_log $tracelevel "$returndata"
326
327     ns_return 200 text/plain "$returndata"
328 }
```

Figura 29. Sentencias de retorno para el bloque cmigetcat

La información retornada desde el LMS es utilizada por el applet APIAdapterApplet en el navegador del cliente para responder a los llamados que pueda realizar el SCO a través del API respectiva. Como se puede deducir de lo anterior, esta implementación del RTE de SCORM en particular envía toda la información relacionada con el SCO durante la

inicialización del mismo, esta información es almacenada por el applet mientras dura la sesión de practica con el SCO.

Cuando el SCO consulta al LMS por información mediante los elementos del modelo de datos CMI, en realidad no se hace ningún llamado al servidor, ya que el applet APIAdapterApplet cuenta con toda la información para responder por si solo con la información requerida. Caso contrario ocurre cuando el SCO modifica el valor de algún elemento del modelo CMI, ejecutando además, el llamado LMSCommit() para persistir los datos dentro del LMS, el applet APIAdapterApplet realiza una petición HTTP, tipo POST, enviando toda la información almacenada al recurso **servlet.tcl** para su procesamiento.

Cuando el SCO ejecuta el método LMSCommit() o LMSFinish() sobre el adaptador de API en el navegador del cliente, el recurso servlet.tcl, ejecuta el procedimiento “cmiputcat*”, figura 8, el cual se encarga de interpretar la información enviada desde el applet (líneas 346 a 362) y almacenar los elementos CMI actualizados en la base de datos.

```
330     cmiputcat*
331     {
332         ns_log $tracelevel "---------------------------------------"
333         switch $functionCalled {
334                         cmiputcat {
335         ns_log $basiclevel "            LMSCommit" }
336                         cmiputcatONFINISH {
337         ns_log $basiclevel "            LMSFinish" }
338         }
339         ns_log $tracelevel "---------------------------------------"
340         ns_log $tracelevel "received data $data from applet: processing. "
341         ns_log $tracelevel "Reference cmi track is $currenttrackid, while lorsmstudenttrack is: $lorsmstudenttrack"
342         set preparselist [lrange [ split $data "," ] 1 end]
343         set lista [list]
344         set value ""
345         #Aqui se reconstruye los elementos CMI de forma "cmi.elemento"=valor usando una busqueda por patrones
346         foreach couple $preparselist {
347             if { [ regexp ^cmi\.* $couple ] } {
348                 if { ! [empty_string_p $value]  } {
349                     set value [concat [lindex $lista end],$value]
350                                     ns_log debug "SCORM PARSER ending recomposing $value "
351                     set lista [lreplace $lista end end $value]
352                     set value ""
353                 } else {
354                                     ns_log debug "SCORM PARSER full couple $couple "
355                     lappend lista $couple
356                 }
357             } else {
358                                 ns_log debug "SCORM PARSER partial couple $couple "
359                 set value [concat $value,$couple]
360                                 ns_log debug "SCORM PARSER partial couple $couple "
361             }
362         }
```

Figura 30. Bloque cmiputcat para invocaciones de persistencia de datos

4.3 Extensión de SCORM para nuevos modelos de datos en .LRN

Teniendo en cuenta la forma en que ha sido implementado el entorno de ejecución de SCORM en .LRN, para dar soporte a nuevos modelos de datos solo es necesario editar los recursos **servlet.tcl** y el applet **APIAdapterApplet**. El recurso **Applet.js** se puede dejar intacto ya que este solo redirige los mensajes hacia el adaptador de API.

De manera general los cambios que se requieren realizar son:

1. Agregar una tabla en la base de datos de .LRN, llamada **dotlrn**, para almacenar la información de los elementos del nuevo modelo de datos.

2. Modificar el archivo **servlet.tcl**, el cual debe cargar y almacenar la información del nuevo modelo de datos y enviar la información al applet en el navegador del cliente.

3. Modificar el applet **APIAdapterApplet**, en el cliente para que pueda interpretar correctamente los elementos del nuevo modelo de datos.

Creación de nueva tabla en la base de datos

Para gestionar, de forma efectiva, los elementos del nuevo modelo de datos, se requiere la adición de una o varias tablas que permitan organizar la información recolectada desde el navegador del cliente. La figura 9 muestra la forma en que .LRN maneja la información asociada al modelo de datos CMI.

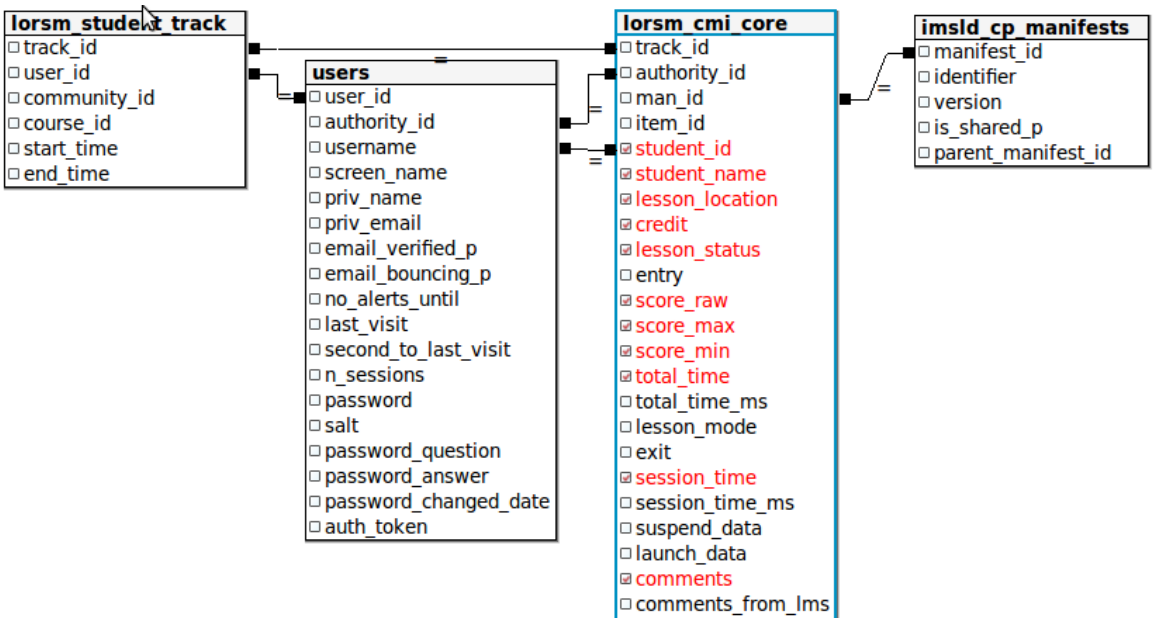

Figura 31. Estructura de datos .LRN para SCORM

Con dicha implementación se busca mantener la menor cantidad de registros dentro de la tabla lorsm_cmi_core, independiente del número de intentos que haga el usuario al consultar el recurso SCORM; cada intento queda registrado dentro de la tabla lorsm_student_track.

Para el modelo de datos propuesto en el capítulo 2, enfocado hacia el seguimiento de diferentes eventos dentro de la actividad formativa, las relaciones necesarias son simples y puede gestionarse con una sola tabla que haga uso de claves foráneas para identificar al estudiante y al recurso SCORM, ver figura 10. Sin embargo, el diseño en la estructura de las tablas, debe realizarse teniendo en cuenta los objetivos y tipos de consultas que se vayan a ejecutar sobre la base de datos, asegurando el rendimiento del motor de bases de datos.

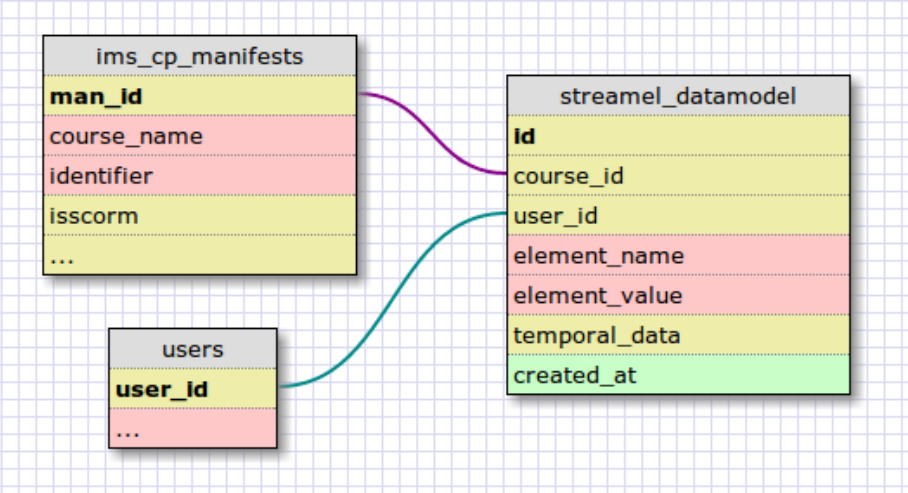

Figura 32. Estructura relacional, tablas modelo streamel

La figura 11 muestra un modelo relacional general que permite integrar varios modelos de datos, nuevos campos pueden ser agregados de acuerdo a las necesidades. Las tablas *users* y *ims_cp_mamifests* son proporcionadas por .LRN, utilizadas para gestionar los usuarios y los cursos SCORM importados a la plataforma, respectivamente.

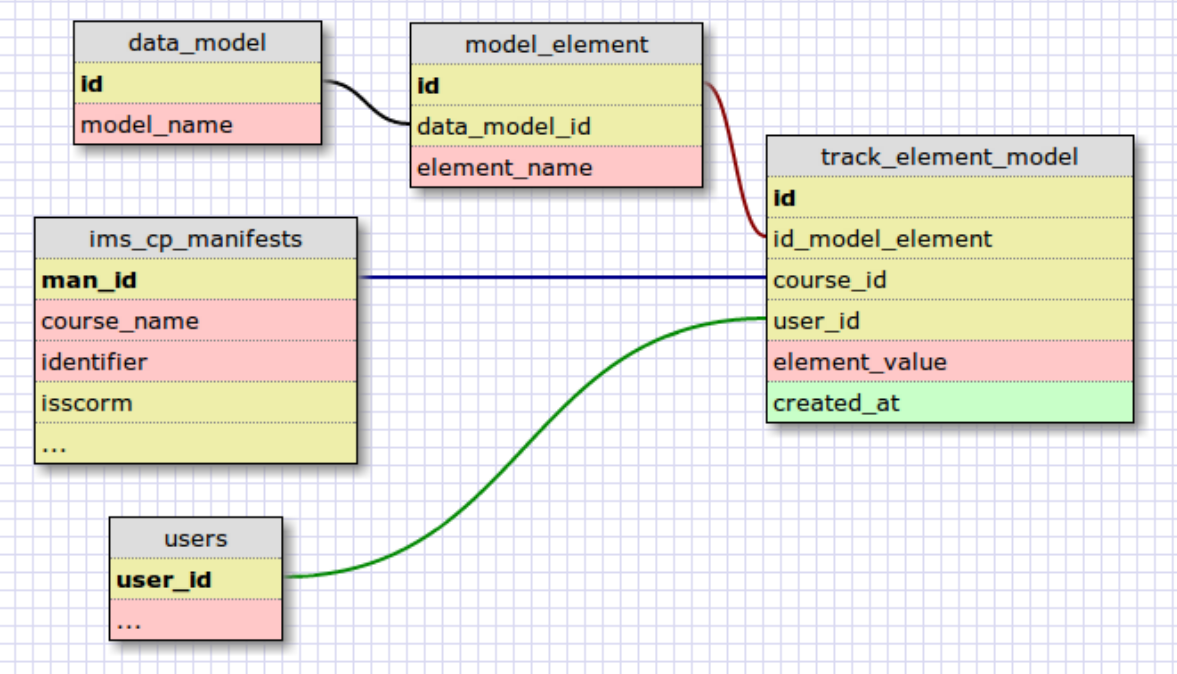

Figura 33. Estructura relacional general para diferentes modelos de datos

Cambios sobre el archivo servlet.tcl

Los cambios requeridos para dar soporte a nuevos modelos de datos, incluyen las consultas necesarias para inicializar el SCO, enviando la información de los elementos del modelo CMI y otros modelos, y la persistencia de los elementos del modelo de datos.

Como se mencionó anteriormente, durante la inicialización se envía información del modelo de datos CMI relevante para el SCO. Aunque se recomienda que el SCO no dependa de información extra a la definida en el modelo CMI, en caso de necesitar enviar información de otros modelos es necesario leer dichos datos y enviarlos junto con la datos del modelo CMI, para esto se debe adicionar el código que permita cargar y adaptar la información después de la línea 300, ver figura 7, por ejemplo como en las líneas 301 a 308 en la figura 12, las cuales retornan el valor del último mensaje asociado a al estado de un video, útil si se quiere que el estudiante retorne al instante en que termino de ver el video, en una sesión previa.

```
293             set returndata "cmi.core.student_id=$student_id,cmi.core.student_name=$name,"
294             append returndata "cmi.core.credit=$credit,cmi.core.lesson_status=$lesson_status,cmi.core.entry=$entry,"
295             append returndata "cmi.core.lesson_mode=normal,cmi.core.lesson_location=$lesson_location,"
296             append returndata "cmi.student_preference.language=italian,cmi.comments=$comments,cmi.comments_from_lms=$comments_from_lms"
297             append returndata ",cmi.suspend_data=$suspend_data,cmi.launch_data=$launch_data"
298             append returndata ",cmi.student_data.max_time_allowed=$max_time_allowed,cmi.student_data.time_limit_action=$time_limit_action"
299             append returndata ",cmi.student_data.mastery_score=$mastery_score"
300         }
301         if { [ db_0or1row streamel_last_position "SELECT temporal_data AS last_position
302                                                   FROM streamel_datamodel
303                                                   WHERE course_id = $currentcourse AND user_id= $user_id
304                                                   AND element_name='streamel.video.state'
305                                                   AND ( element_value='paused' OR element_value='completed')
306                                                   ORDER BY created_at DESC"]} {
307                 append returndata ",streamel.video.last_position=$last_position"
308         }
309         #treating time from table back to system (lorsm.cmi.time fields are showing just seconds)
310         set prefix "0"
311         set hours [expr int ($session_time/3600) ]
```

Figura 34. Ejemplo carga de datos en el modelo streamel

Para el almacenamiento de la información en los mensajes del nuevo modelo de datos que se envía desde el SCO se puede crear otro bloque dentro de la sentencia de control de flujo *switch* (figura 6, línea 96), similar a los bloques cmigetcat y cmiputcat, por ejemplo en la figura 13, líneas 337 a 354, el bloque "*commit_for_new_models*" procesaría la cadena enviada desde el cliente y la almacenaría en la base de datos.

```
334         ns_return 200 text/plain "$returndata"
335     }
336
337     commit_for_new_models
338     {
339     #Codigo para cambiar la cadena codificada "...,streamel.video.state=completed[240],..." a:
340     #variables con la informacion decodificada $video_state=completed y $temporal_data=240
341         set tm_now [timestamp -format "%Y-%m-%d %X"]
342         set todo "INSERT INTO streamel_datamodel (id, course_id, user_id, element_name, element_value,temporal_data,created_at)
343                         values  (0,$currentcourse,$user_id,'streamel.video.state',$video_state,$temporal_data, $tm_now)
344         db_dml todo $todo
345         set http_response "OK"
346         if { [db_resultrows] == 1 } {
347                 ns_log debug "StreamelModel element processing INSERT: '$todo' successfull"
348         } else {
349                 set http_response "Error=103,ErrorDescription=\"No functionCalled meaningful value provided\""
350                 ns_log Warning "StreamelModel element processing INSERT: '$todo' not successfull -> please check"
351         }
352
353         ns_return 200 text/plain http_response
354     }
355     cmiputcat*
356     {
357         ns_log $tracelevel "----------------------------------------"
358         switch $functionCalled {
359                         cmiputcat {
360         ns_log $basiclevel "            LMSCommit" }
```

Figura 35. Ejemplo persistencia de datos de modelo de datos streamel

Cambios sobre el applet APIAdapterApplet

El applet se encuentra dentro del archivo `stuff.jar`, que se descarga desde el LMS hacia el cliente. El archivo `stuff.jar` está conformado diferentes clases, la figura 14 muestra las más relevantes.

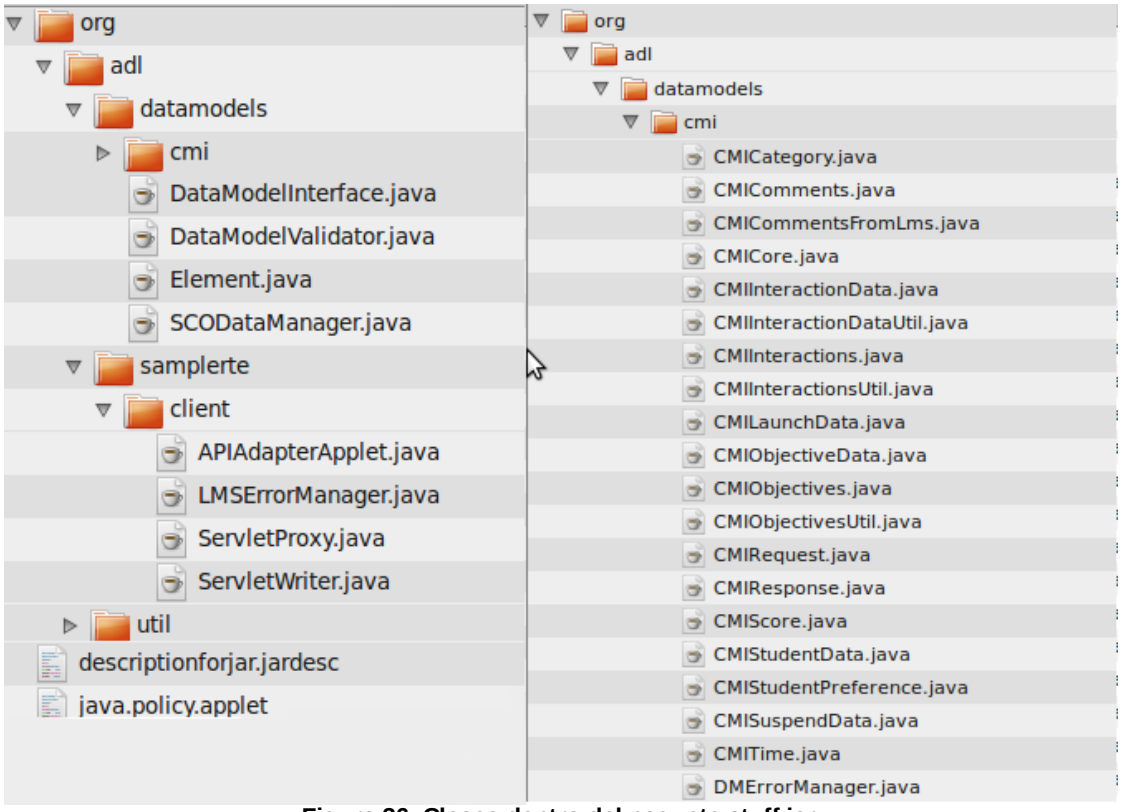

Figura 36. Clases dentro del paquete stuff.jar

En el paquete `org.adl.samplerte.client` se encuentran las clases principales en el funcionamiento del adaptador de API de SCORM. El paquete `org.adl.datamodels.cmi` incluye las clases que dan soporte al estándar CMI.

Durante la inicialización del SCO el applet hace un llamado al método "`LMSInitialize`" el cual hace uso del objeto `ServletProxy` para abstraer el medio de comunicación con el LMS. La clase `ServletProxy` no interpreta mensajes CMI, solo se encarga de enviar y solicitar información como simples cadenas de texto. La figura 15 muestra el llamado sobre el objeto proxy de la clase `ServletProxy`.

```
388         if ( _Debug ) { System.out.println("Trying to get SCO Data from servlet...");   }
389         // Build the local (client-side) LMS data model cache by getting the SCODataManager class instances from the LMS Servlet
390         ServletProxy servletProxy = new ServletProxy(this.servletURL, this.cookie);
391         String DataFromServer =  servletProxy.GetSCOData();
392         if(DataFromServer.startsWith("ERROR")) {
393                 APIAdapterApplet.isLMSInitialized=false;
394                 APIAdapterApplet.lmsErrorManager.SetCurrentErrorCode("201");
395                 System.out.println( "Error on server on during communication" );
```

Figura 37. APIAdapterApplet.java: Ejecución de método GetSCOData para la inicialización del SCO.

Durante la invocación del metodo GetSCOData(), suponiendo hechas las modificaciones en el archivo servlet.tcl mencionadas anteriormente, la cadena retornada sigue el patrón:

`cmi.core.student_name=Leonardo,cmi.core.student_id=ldiaz@unicauca.edu.co,cmi.core.location=1,streamel.video.status=paused,streamel.video.forward_to=10`

Lo primero que se debe hacer es extraer de la cadena la información de otros modelos, lo cual se debe realizar antes de la línea 414, ver figura 16,

```
410             System.out.println("In LMSInitialize : now parsing server data" );
411             //String[] couples = DataFromServer.split(",");
412             Pattern p=Pattern.compile("^([^=]+)=(.*)$");
413             DataFromServer=","+DataFromServer;
414             String[] couples = DataFromServer.split(",cmi\\.");
415             int howmany = couples.length;
416             System.out.println("In LMSInitialize, we have : " + howmany + " couples ");
417             //note we start from first couple since zeroest is null
418             for ( int z = 1 ; z< howmany; z++) {
419                 String Element="";
420                 String Value="";
421                 Matcher m=p.matcher(couples[z]);
```

Figura 38. ServletWriter.java: extracción de elementos CMI de la cadena de retorno

La eliminación puede realizarse usando patrones, por ejemplo una implementación para el modelo streamel se muestra en la figura 17, líneas 411-418:

```
410             System.out.println("In LMSInitialize : now parsing server data" );
411             //to match elements from other data models, in this case streamel Data model
412             Pattern streamelp = Pattern.compile(",streamel[^,\\r\\n]*");//finds streamel messages
413             Matcher strMessages = streamelp.matcher(DataFromServer);
414             String streamelElements = new String();
415             while (strMessages.find()) {//here we have all streamel messages
416                 streamelElements.concat(strMessages.group());
417             }
418             DataFromServer = strMessages.replaceAll("");//removes all streamel messages
419             Pattern p=Pattern.compile("^([^=]+)=(.*)$");
420             DataFromServer=","+DataFromServer;
421             String[] couples = DataFromServer.split(",cmi\\.");
```

Figura 39. ServletWriter.java: Adición para extracción de elementos del modelo de datos streamel

De la misma forma en que se procesa el contenido de la cadena DataFromServer, en las líneas 413 a 458 del archivo original `APIAdapterApplet.java`, se debe procesar la cadena `streamelElements`, que contiene los mensajes del modelo de datos streamel, para almacenarlos posteriormente en un objeto de la clase `SCODataManager`, la cual administra toda la información de los modelos de datos en el cliente. Solo hay que tener en cuenta a la hora de procesar los mensajes de otros modelos, que se debe cambiar la sentencia 431 de esto:

```
431             if(z>0) { Element="cmi."+Element; }
```

A esto, dependiendo del nombre del modelo que se procese, por ejemplo:

```
431          if(z>0) { Element="streamel."+Element; }
```

El objeto de la clase `SCODataManager` es populado por otro objeto de la clase `DataModelInterface`, figura 18, encargado de validar el tipo de mensaje, de crear la petición correspondiente y luego consultar efectivamente al `SCODataManager`.

```
750       String theRequest = element + "," + setValue;
751       if(_Debug)
752       {
753          System.out.println("Request being processed: LMSSetValue(" + theRequest + ")");
754          System.out.println( "Looking for the element " + element );
755       }
756
757       DataModelInterface dmInterface = new DataModelInterface();
758       dmInterface.processSet(theRequest,theSCOData,dmErrorManager);
759
760       // Set the LMS Error Manager from the DataModel Manager
```

Figura 40. APIAdapterApplet.java: Metodo LMSSetValue para popular el objeto SCODataManager

Dentro de la clase `DataModelInterface`, lo primero que se debe hacer es modificar el método `isValidRequest()`, para permitir el procesamiento de elementos de otros modelos, ver figura 19 línea 233.

```
218    private boolean isValidRequest(String theRequest)
219    {
220       boolean rtrnFlag = true;
221       StringTokenizer stk = new StringTokenizer(theRequest, ".", false);
222       int totalNumOfTok = stk.countTokens();
223
224       // The request must have at least 2 tokens.  If a request has less
225       // than two tokens then there was an invalid request received
226       if ( totalNumOfTok < 2 )
227       {
228          rtrnFlag = false;
229       }
230
231       String token = stk.nextToken();
232       // Check to make sure SCO is using the cmi data model
233       if ( token.equals("cmi") || token.equals("streamel") )
234       {
235          rtrnFlag = true;
236       }
237       else
238       {
239          rtrnFlag = false;
240       }
241
242       return rtrnFlag;
243
244    } // end of isValidRequest()
```

Figura 41. DataModelInterface.java: Modificación método isValidRequest()

Con ese simple cambio los metodos `processSet`, cuando se modifican elementos del modelo de datos, y `processGet`, cuando se consulta por elementos del modelo de datos, funcionan de acuerdo a lo esperado. Esto métodos hacen uso de objetos de la clase

CMIRequest para comunicarse con el “almacén” de datos en la clase SCODataManager. La clase CMIRequest, aunque su nombre de a entender lo contrario, puede manejar peticiones para cualquier tipo de modelo de datos, por tal razón puede y debería ser utilizado sin ningún cambio.

Para que la clase SCODataManager pueda almacenar los elementos de diferentes modelos de datos, es necesario que dicho modelo sea implementado de la misma forma que ha sido implementado el modelo de datos CMI. El paquete org.adl.datamodels.cmi.CMI contiene las clases que conforman este modelo y es una herramienta útil como referencia para la implementación de otros modelos de datos.

El LMS .LRN soporta únicamente el elemento cmi.core de SCORM, el cual se encuentra implementado por la clase CMICore.java, basado en esta clase se puede implementar la clase STREAMELVideo.java que defina las reglas de cada uno de sus elementos. Para aprovechar las facilidades que ofrece esta implementación del RTE, el diseño de la clase STREAMELVideo.java o cualquier elemento de otro modelo de datos deben seguir un conjunto de convenciones:

1. El nombre del elemento de jerarquía superior será implementado con un formato especial, cada palabra de su nombre estará compuesta por su primera letra en tamaño capital seguida del resto de la palabra en minúscula, por ejemplo:

    - Para el mensaje streamel.video.status, la clase que la implemente se debe llamar STREAMELVideo.java.

    - Si el mensaje es streamel.video_comment.total_comments, si implementación se debe hacer sobre la clase STREAMELVideoComment.java.

2. Cada sub-categoría definida en el mensaje debe implementarse como una propiedad con su correspondiente método get y set teniendo en cuenta que, en los métodos, cada palabra que la conforme será nombrada por la primera letra en formato capital, seguida del resto en minúscula. Por ejemplo:

    - En el mensaje streamel.video.status su implementación en la clase STREAMELVideo cuenta con el método getStatus() y setStatus().

    - En el mensaje streamel.video_comment.total_comments su implementación en la clase STREAMELVideoComment.java cuenta con el metodo getTotalComments() y setTotalComments().

3. La clase que implemente un elemento del modelo de datos debe contar con la implementación de los métodos:

    - `performSet(CMIRequest theRequest,DMErrorManager dmErrorMgr)`
    - `performGet(CMIRequest theRequest,DMErrorManager dmErrorMgr)`

    Que conozcan la forma de interpretar la petición tipo `CMIRequest` y almacenar su información de forma correcta. Dicha implementación puede ser copiada de la clase `CMICore` sin mayores cambios.

Finalmente retomando las modificaciones necesarias sobre la clase `SCODataManager` la cual usa introspección sobre los mensajes para determinar que objeto dentro de su colección puede interpretar dicho mensaje. Es necesario, en primera instancia, agregar dentro de esta clase el nuevo elemento del modelo de datos, figura 20, inicializar el elemento y agregar sus correspondientes métodos `set` y `get`:

```
84 import java.lang.reflect.*;
85
86 //adl imports
87 import org.adl.util.debug.*;
88 import org.adl.datamodels.cmi.*;
89 import org.adl.datamodels.streamel.*;
90
91 public class SCODataManager implements Serializable
92 {
93    // Information required to be furnished by all LMS Systems.  What
94    // all SCOs may depend on upon start up
95    public CMICore core;
96    public STREAMELVideo video;
97    // Unique information generated by the SCO during
98    // previous uses, that is needed for the current use
```

Figura 42. SCODataManager.java: adición de nueva variable para representar un elemento de un nuevo modelo de datos

Cuando se hace consultas de lectura o escritura sobre los elementos del modelo de datos en el objeto de la clase `SCODataManager`, internamente se hacen llamados al método `processRequest()`, el cual debe ser modificado para cargar, de acuerdo a las convenciones mencionadas con anterioridad, el objeto responsable de interpretar el tipo de mensaje especifico. El cambio necesario se muestra en la figura 21, donde se determina de acuerdo al tipo de mensaje el nombre de la clase a cargar en tiempo de ejecución.

```
329   private String processRequest(CMIRequest theRequest,
330                                 DMErrorManager dmErrorMgr)
331   {
332
333      // Result to be returned
334      String result = new String("");
335
336      // takes the base category and returns the class name
337      // (i.e. student_data --> CMIStudentData
338      String tmpClassName = convertString(workingBaseCategory);
339      String className = DM_CLASSNAME + tmpClassName;
340
341     StringTokenizer stk = new StringTokenizer(CMIRequest, ".", false);
342     String token = stk.nextToken();
343       // Check to make sure SCO is using the cmi data model or streamel
344     if ( token.equals("streamel") ){
345        className = "org.adl.datamodels.streamel.STREAMEL" + tmpClassName;
346     }
347
348      if ( DebugIndicator.ON )
349      {
350         System.out.println("Class Name: " + className);
351         System.out.println("Working Base Cat: " + workingBaseCategory);
352      }
353
354      try
355      {
356         // Find out the Class of the request
357         Class c = Class.forName(className);
358         try
359         {
360            // Find the field in the SCODataManager that maps
361            // to the baseCategory
362            Field tmpField = this.getClass().getField(workingBaseCategory);
```

Figura 43. SCODataManager.java: configuración de clases a cargar para el procesamiento de nuevos mensajes

Con estos cambios el objeto de la clase `SCODataManager` ya está en capacidad de cargar y almacenar la información enviada desde el LMS durante la inicialización del SCO. Además con los cambios realizados hasta el momento cada vez que mediante *javascript* se realice una consulta de lectura o escritura sobre el adaptador de API, el applet puede responder con la información solicitada ya sea CMI u otro modelo, siempre y cuando exista una implementación de ese modelo de datos.

Ahora solo queda modificar el applet para que cuando se realice el llamado LMSCommit() definido en SCORM, se envíen los datos del modelo CMI y del nuevo modelo. Nuevamente el applet se basa en la funcionalidad definida en la clase `ServletProxy`, específicamente en el método `PutSCOData()`, por lo que se necesitaría duplicar dicho método con un pequeño cambio sobre el contenido de la línea 181, que define cual de los

procedimientos será invocado en el archivo servlet.tcl, la grafica 22 muestra como quedaría el cambio, teniendo en cuenta lo definido anteriormente.

```
171    public String PutSCOData2( SCODataManager scoData )
172    {
173       try
174       {
175          if(_Debug)
176          {
177             System.out.println("In ServletProxy::PutSCOData()");
178          }
179
180           //MICHELE adds function
181          String servletCommand = "commit_for_new_models";
182
183 /*       These are not needed for functionality for new models
184          if (APIAdapterApplet.arewefinishing==true) {
185               System.out.println("*********************************");
186               System.out.println("***LMSFinish                    ***");
187               System.out.println("*********************************");
188               servletCommand = "cmiputcatONFINISH";
189          }*/
190
191          Serializable[] data = {servletCommand, scoData};
192
```

Figura 44. . ServletProxy.java: modificación para ejecutar las sentencias almacenado en el LMS

Este nuevo método será invocado justo cuando termine el método original, es decir que primero se hace el envió de los mensajes CMI y luego los del nuevo modelo. Solo falta modificar el archivo `ServletWrite` para que envié efectivamente la información del nuevo modelo, esto se puede hacer modificando su método `postObjects()` para agregar los nuevos elementos a la cadena que se transmite los elementos CMI o duplicar con otro nombre este método para que solo envíe los mensajes del modelo de datos deseado. Dicho método solo requiere realizar los llamados de la misma forma que lo hacen para los elementos del objeto de la clase CMICore, por ejemplo:

```
STREAMELVideo video = inSCOData.getVideo();
context = ",streamel.video.";
result += context+"status="+video.getStatus().getValue().toString();
result += context+"forward_to="+video.getForwardTo.getValue().toString();
```

ANEXO 5. Evaluación del caso de estudio: Implementación del servicio de video *streaming* "STREAMEL"en EVA de la Universidad del Cauca.

5.1 Introducción

En este documento se presentan los resultados y análisis de la encuesta realizada al caso de estudio desarrollado en esta monografía, en particular el servicio de video streaming "STREAMEL" que se implementó como caso de estudio en el entorno virtual de aprendizaje EVA de la Universidad del Cauca. La encuesta se realiza sobre un grupo de personas que accedieron a este servicio e interactuaron con él, en los meses de diciembre del 2009 y enero de 2010.

El propósito de este documento es realizar un primer análisis y arrojar posibles resultados que nos permitan observar cuales de las características y prestaciones de este tipo de servicios "servicios de streaming y su integración con LMSs" se deben mantener o rediseñar, además cuales podrían ser sus deficiencias o herramientas más útiles para lograr maximizar o potenciar las actividades de aprendizaje; Teniendo como fin el que puedan servir de guía en el momento de crear los servicios a los profesores o diseñadores que realizan las actividades de aprendizaje.

Dentro de las preguntas claves que se desarrollan en la encuesta se pretende observar si las velocidades y calidad de reproducción de los servicios y algunos de los servicios de valor agregado como los comentarios o diapositivas asociadas al video son importantes para mejorar el impacto en los objetivos de aprendizaje.

5.2 Variables y resultados

- **Medio de Acceso a Internet**

Como primer dato importante a estudiar se encuentra el medio de acceso a internet que es más utilizado por los participantes del servicio de video streaming.

Con este tipo de información se pueden deducir otro tipo de datos técnicos como la velocidad de acceso y la disponibilidad del servicio que se presenta para los participantes.

En este caso se obtuvo que la gran mayoría de participantes realizan el acceso a internet desde lugares diferentes a la institución donde estudian, que es precisamente una de las características de la educación virtual (en cualquier momento y cualquier lugar) y como se menciona en la monografía la gran mayoría de participantes tiene como medio de acceso a internet "ADSL" con una velocidad promedio de 512 K.

Medio de Acceso a Internet

Módem telefónico	0	0 %
ADSL	14	61 %
Cable módem	7	30 %
Conexión inalámbrica	2	9 %

Tabla 1. Medio de Acceso a Internet

Velocidad de acceso a internet

56k-128k	2	9%
128k-256k	6	26%
256k-512k	9	39%
512k-1Mb	6	26%
Más de 1Mbps	0	0%

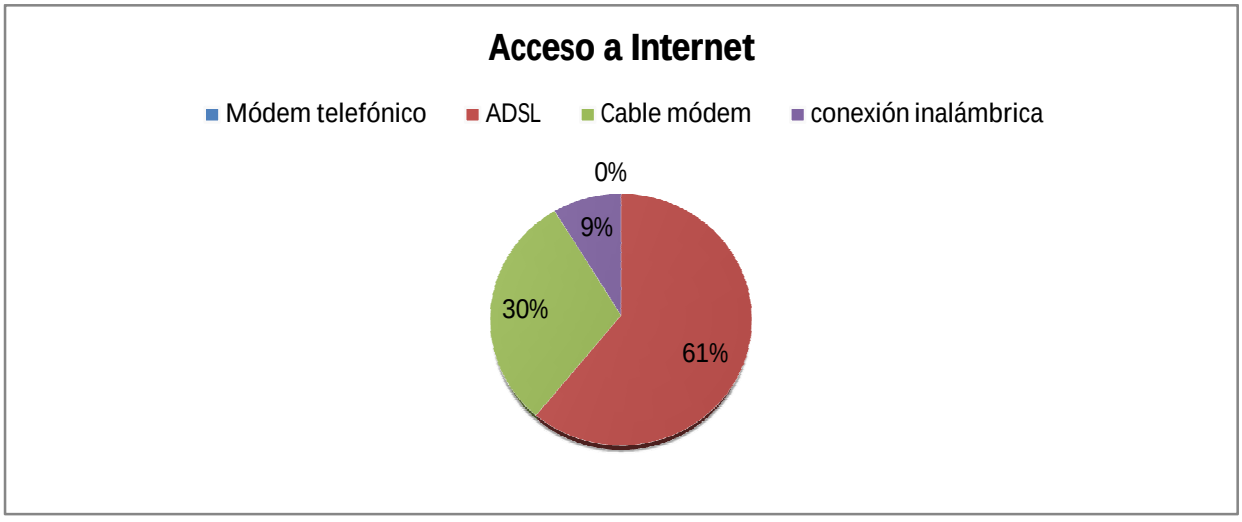

Figura 1. Medio de acceso a Internet

- **Forma de acceder al servicio de video streaming**

Esta variable indica si la forma en que se accede al servicio una vez se ingresa a la plataforma EVA es clara para los participantes del servicio. Se puede observar que para la mayoría de participantes gracias a que anteriormente han tenido acceso la plataforma EVA en los cursos virtuales que se ofrecen en la Universidad del Cauca, es intuitiva la manera de acceder al servicio.

También se debe tener en cuenta que esta variable está condicionada a la manera como despliega los objetos de aprendizaje cada plataforma LMS, por ejemplo Moodle lo hace de diferente forma que .LRN lo cual en algunas ocasiones confunde a los participantes la ubicación donde deben ejecutar el servicio. A pesar que dentro de los estándares que manejan los LMSs deberían hacerlo de igual manera.

Claro acceso	21	91%
No claro acceso	2	9%

Tabla 2. Forma de acceder al servicio Streamel.

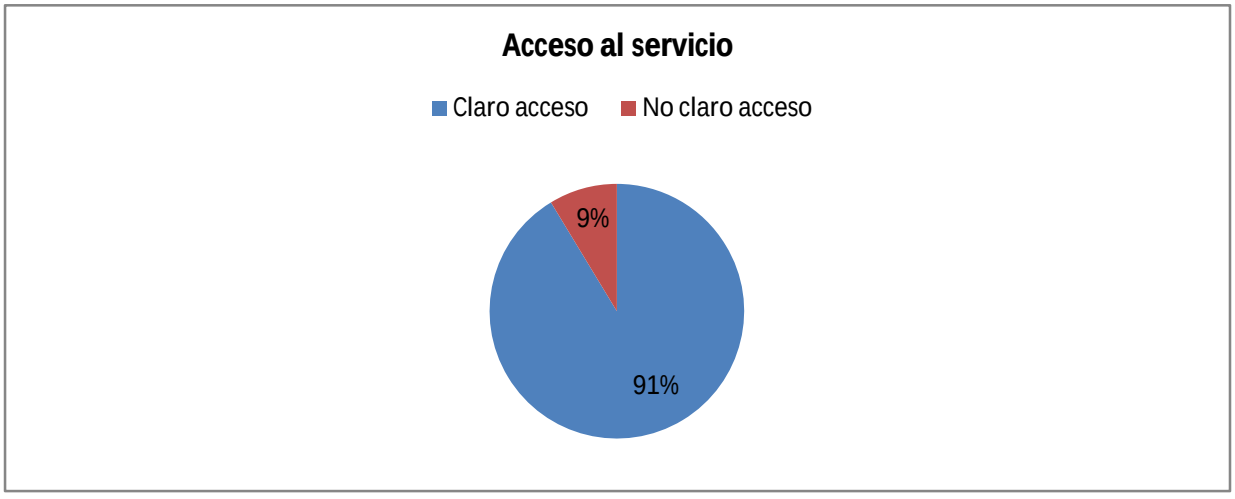

Figura 2. Acceso al servicio de streaming Streamel.

- **Velocidad de reproducción**

En esta variable se presenta la percepción subjetiva de los participantes de que tan rápido se accede al servicio y al mismo tiempo como perciben los participantes la velocidad de reproducción del servicio de streaming.

Como se puede observar para la gran mayoría de participantes la velocidad es buena aunque se debe tener en cuenta que esto se debe a que la mayoría tienen velocidad de conexión a internet alta.

Excelente	2	9%
Buena	16	69%
Regular	2	9%
Mala	3	13%

Tabla 3. Velocidad de reproducción del servicio

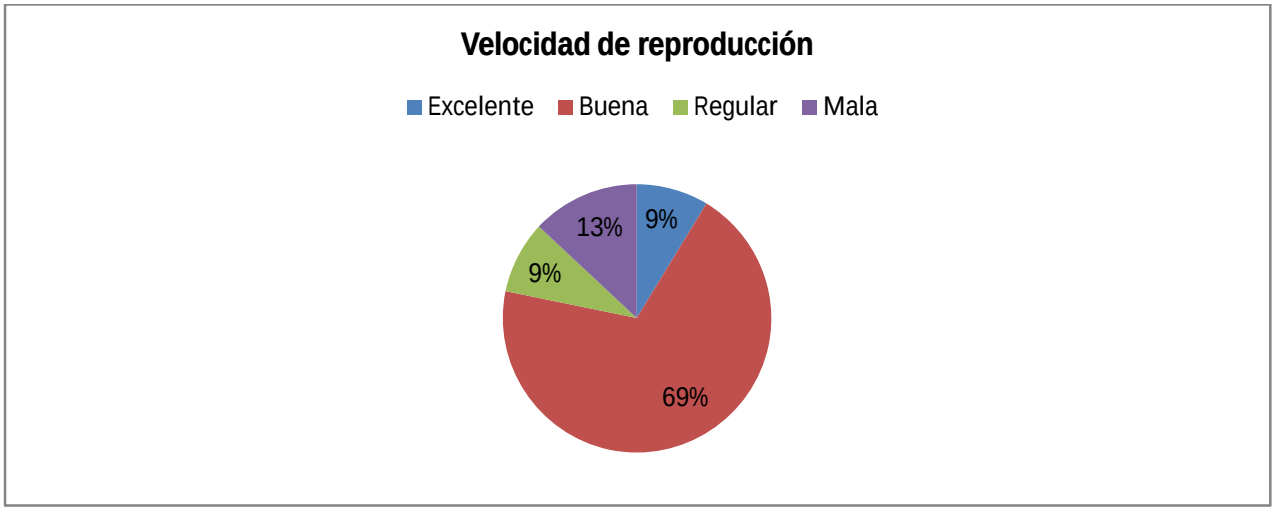

Figura 3. Velocidad de reproducción del servicio

- **Calidad de reproducción**

En esta variable al igual que en la anterior se muestra la percepción subjetiva del participante de cómo observa la calidad de los videos mediante el servicio de streaming. Se puede observar que en general la calidad de los videos no es la óptima, aunque esto era de esperarse ya que como se menciona en la monografía la calidad de un video depende de muchos factores.

Excelente	1	4%
Buena	10	44%
Regular	9	39%
Mala	3	13%

Tabla 4. Calidad de reproducción del servicio “videos”.

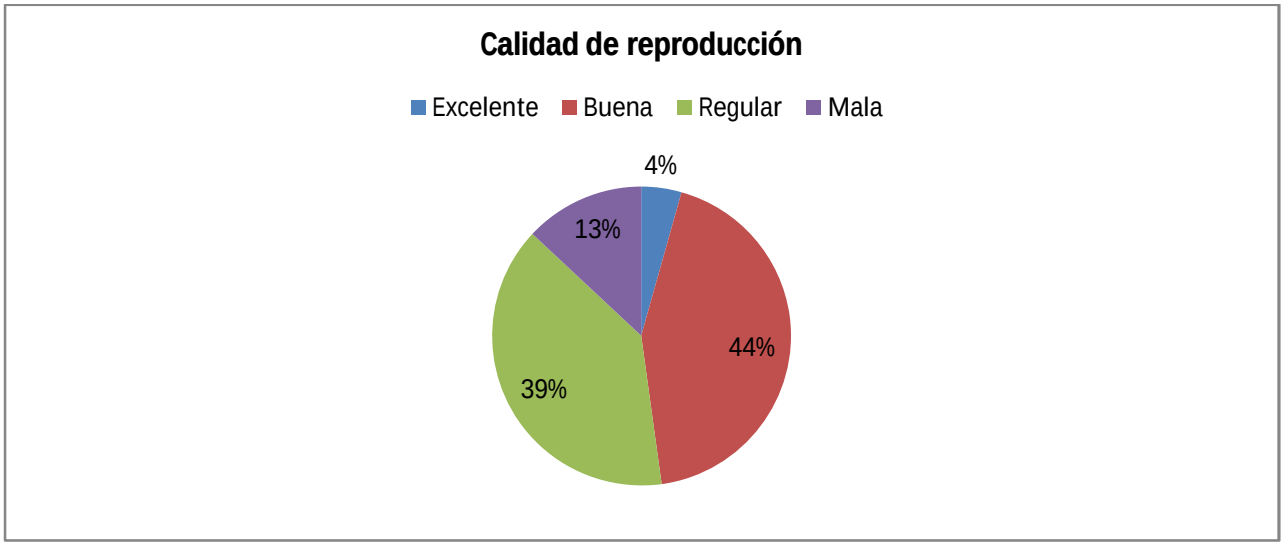

Figura 4. Calidad de reproducción del servicio “videos”.

- **Herramienta de adaptación de contenido**

Esta variable hace referencia al servicio de valor agregado al servicio de video streaming y cuál fue su impacto para los participantes, esta herramienta permite ofrecer varios formatos de los videos y audio del recurso que se está presentando, en general los participantes la utilizan como prueba pero no encuentran mucha diferencia de un formato a otro, exceptuando cuando el recurso se adapta a audio que permite una mayor velocidad de reproducción del recurso.

Es útil	21	91%
No es útil	2	9%

Tabla 5. Utilidad de la adaptación de contenido.

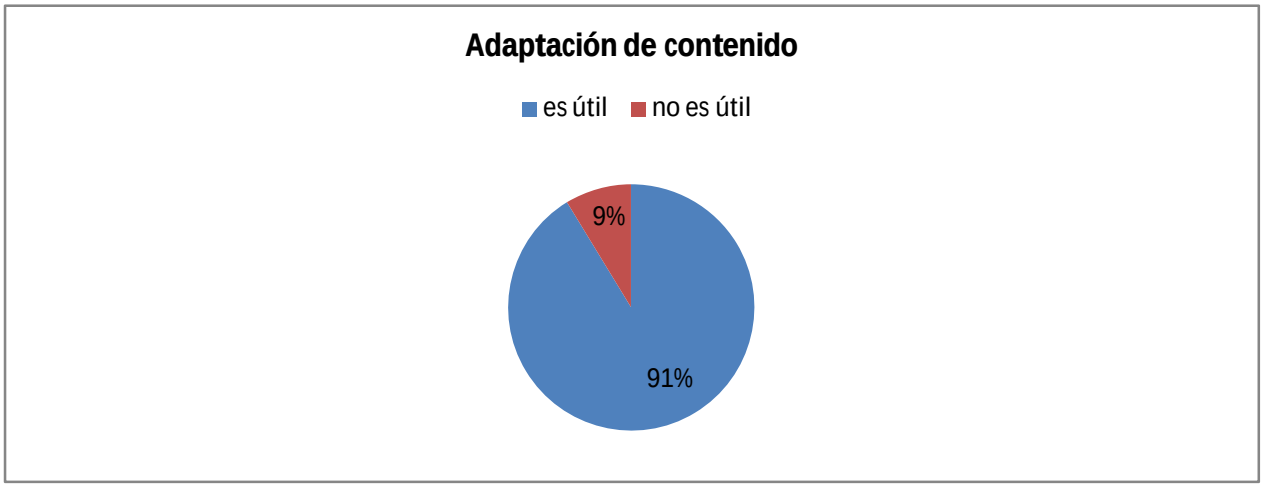

Figura 5. Utilidad de la adaptación de contenido.

- **Expandir pantalla**

En esta variable se refleja la utilización de la funcionalidad de expandir pantalla que permite la aplicación que se desarrolló para el servicio de streaming. A diferencia de la manera en que despliega comúnmente los objetos de aprendizaje los LMS, esta funcionalidad permite que se observe de manera expandida y en consecuencia ofrecer una interfaz común para todos los participantes de los contenidos educativos. Los participantes manifiestan con qué frecuencia utilizan esta funcionalidad.

Siempre	8	35%
A veces	15	65%
Nunca	0	0%

Tabla 6. Frecuencia de utilización de expandir pantalla

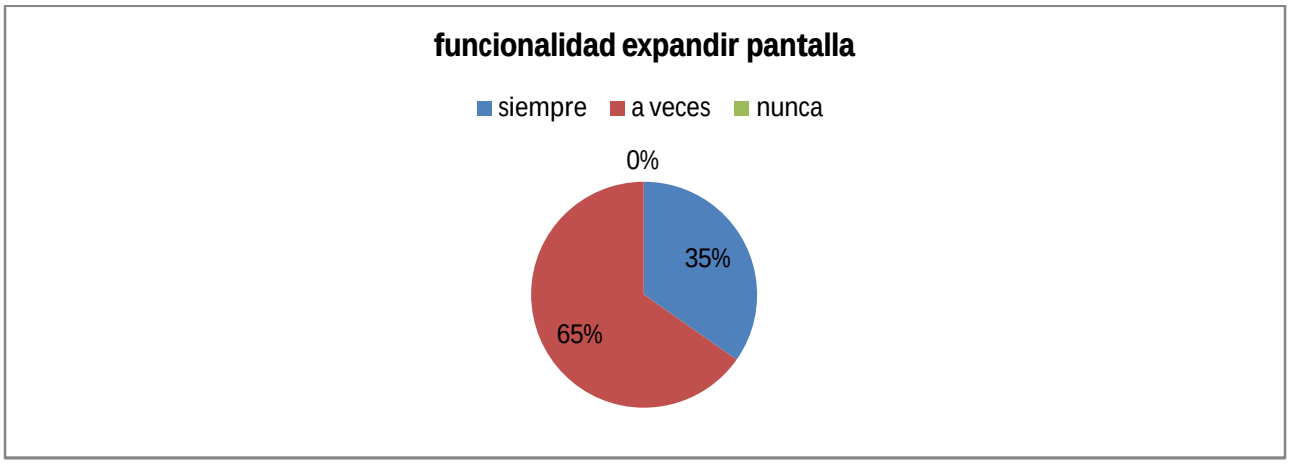

Figura 6. Frecuencia de utilización de expandir pantalla

- **Servicio de comentarios**

Esta variable indica la frecuencia con que fue utilizada esta característica dentro del servicio de video streaming y si los participantes consideran útil la utilización de la misma. A pesar que dentro de las observaciones descritas por los participantes se encuentra que el diseño y visualización de esta característica no es la más adecuada, muchos utilizaron este servicio.

Frecuencia de utilización de servicio de comentarios

Siempre	7	30%
A veces	14	61%
Nunca	2	9%

Tabla 7. Frecuencia de utilización de servicio de comentarios

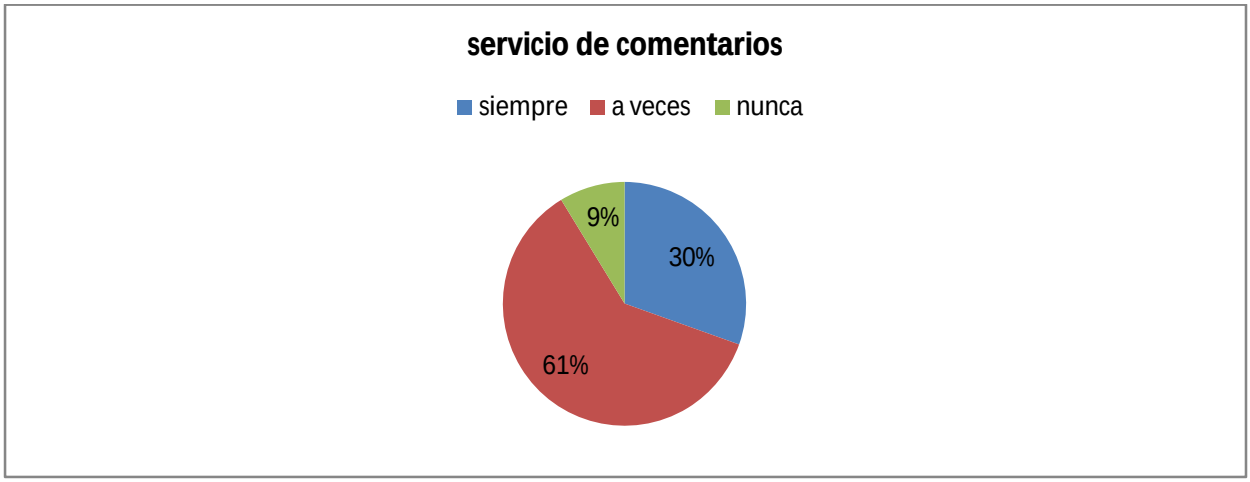

Figura 7. Frecuencia de utilización de servicio de comentarios

Importancia del servicio de comentarios para la interacción entre los participantes

Favorece la interacción	16	70%
No favorece la interacción	7	30%

Tabla 8. Importancia del servicio de comentarios

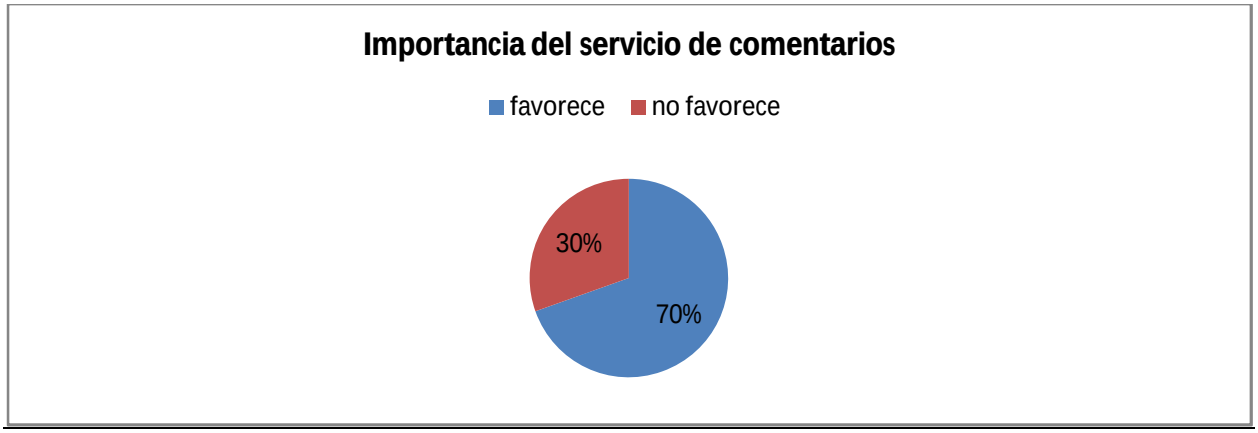

Figura 8. Importancia del servicio de comentarios

- **Consideración del servicio de video streaming como una herramienta útil en el aprendizaje**

Esta variable representa la consideración de los participantes a cerca de la utilidad de la integración de este tipo de servicios, en particular el servicio de video streaming y las herramientas de valor agregado, como herramienta útil en el momento de alcanzar los objetivos de aprendizaje, es decir, su consideración si creen que pueden aprender más o de mejor manera los contenidos que ofrece su tutor.

Útil	22	96%
No útil	1	4%

Tabla 9. Consideración del servicio como una herramienta útil en el aprendizaje

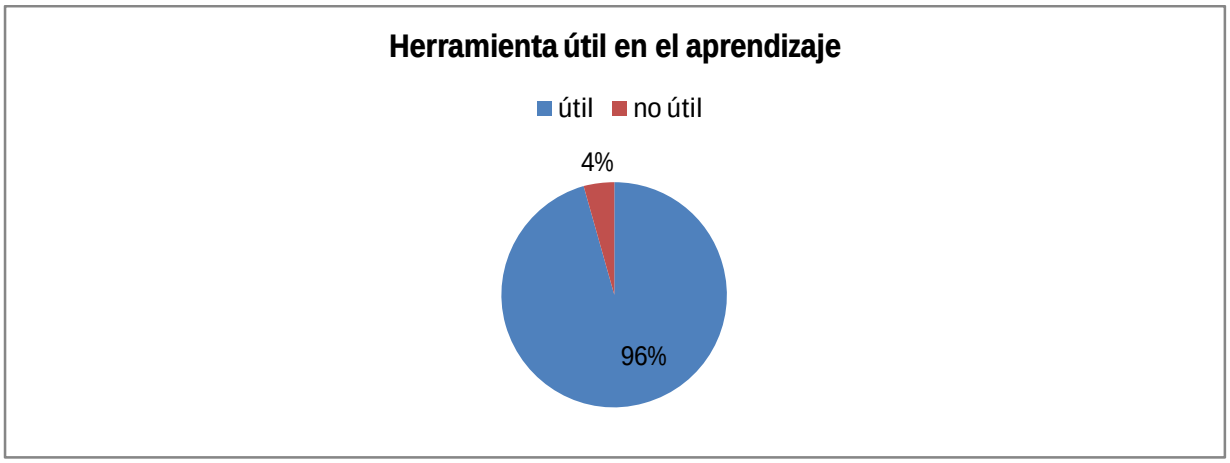

Figura 9. Consideración del servicio como una herramienta útil en el aprendizaje

Como se puede observar el servicio de video streaming Streamel se considera por la gran mayoría de participantes como una herramienta útil en el momento de pensar en una herramienta que permita a los estudiantes y profesores mejorar su calidad de educación en este tipo de ambientes virtuales.

- **Streamel como herramienta novedosa**

Esta variable indica si el servicio de video streaming es considerado por los participantes como una herramienta novedosa para los sistemas de gestión de aprendizaje o en otros términos, si es novedosa dentro las tecnologías que utilizan los estudiantes y profesores en general para su aprendizaje.

Novedosa	4	17%
No es novedosa	19	83%

Tabla 10. Consideración de Streamel como herramienta novedosa

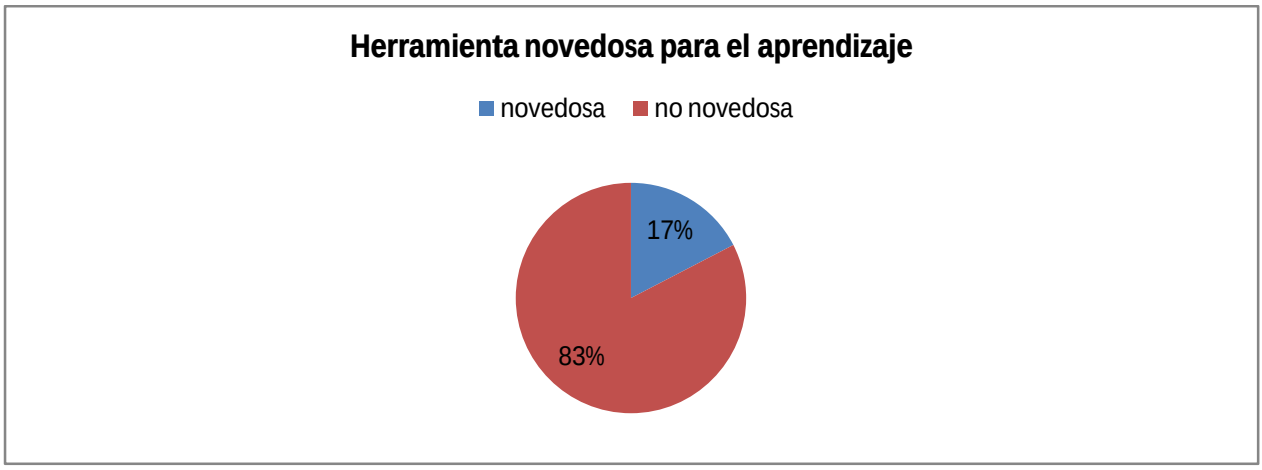

Figura 10. Consideración de Streamel como herramienta novedosa

Como se puede observar pocos participantes consideran la herramienta como novedosa en su aprendizaje actual, esto puede deberse a la gran penetración que tiene internet en la educación en la actualidad donde para aprender un tema no sólo se realiza a través de un sistema de gestión de aprendizaje LMS, si no que la red en general provee muchas herramientas para lograrlo y en donde se pueden encontrar servicios como el de video streaming y muchos otros.

Para los pocos que participantes que manifiestan que la herramienta es novedosa puede ser debido a que no tienen mucho contacto con las tecnologías de la información, como puede ser el caso de estudiantes de colegio y estudiantes de programas que no utilizan el computador como herramienta central de información.

5.3 Metodología

1. Determinar cuál es el objetivo de esta actividad.
2. Desarrollar la encuesta teniendo en cuenta los conceptos importantes que favorezcan en el momento de determinar si este tipo de servicios (servicios de streaming) y su integración a los sistemas de gestión de aprendizaje son herramientas útiles para mejorar las actividades de aprendizaje para los estudiantes. Algunas de las preguntas son abiertas para tener un entendimiento y respuestas más claras acerca de la interacción que tienen los participantes con el servicio de streaming. Gracias a la implementación y puesta en marcha de un servicio de video streaming en EVA en la Universidad del Cauca se logra obtener respuestas más acertadas y apropiadas para esta actividad.
3. Aplicar el cuestionario vía web y en documento impreso.
4. Revisión e identificación de las variables útiles para el desarrollo de la encuesta.

5. Elaboración de tablas y estadísticas que nos permiten observar los resultados y posteriormente hacer un análisis de los resultados.

5.4 Modelo de encuesta

Nombre:_____

Ocupación:_____

1. Al ingresar a la plataforma de educación virtual EVA de la Universidad del Cauca, considera que es claro la forma de acceder al servicio de video streaming Streamel?

   ☐ Si
   ☐ No

2. Evalúe la velocidad de reproducción que observó del servicio de streaming Streamel.

   ☐ Excelente
   ☐ Bueno
   ☐ Regular
   ☐ Malo

3. Informe el tipo de conexión desde donde accedió el recurso de video streaming.

   ☐ Línea telefónica
   ☐ ADSL
   ☐ Cable módem
   ☐ Conexión inalámbrica

   Escoja la velocidad de su conexión de internet, con la cual accedió al servicio de streaming.

   ☐ Entre 56k-128k
   ☐ 128k-256k
   ☐ 256k-512k
   ☐ 512k-1Mb
   ☐ Más de 1Mbps

4. Evalúe la calidad de reproducción que observó del servicio de streaming Streamel.

☐ Excelente
☐ Bueno
☐ Regular
☐ Malo

¿Por qué?

_____
_____
_____
_____
_____

5. Considero útil la herramienta de adaptación de contenido (varios formatos del mismo recurso) para enviar el recurso que mejor se desempeñe sobre las limitantes de red del participante?

   ☐ Si
   ☐ No

   ¿Por qué?

   _____
   _____
   _____
   _____
   _____

6. Utilizó la funcionalidad de “expandir pantalla”, fullscreen, en el objeto de aprendizaje?

   ☐ Siempre
   ☐ A veces
   ☐ Nunca

7. Utilizó el servicio de comentarios del objeto de aprendizaje Streamel?

   ☐ Siempre
   ☐ A veces
   ☐ Nunca

8. Cree que el servicio de comentarios favorece la interacción entre los participantes del servicio?
   ☐ Si
   ☐ No

¿Por qué?

_____
_____
_____
_____
_____

9. Cree que el servicio de video de *streaming* Streamel , es una herramienta útil para alcanzar objetivos de aprendizaje dentro de su plan de estudios.

☐ Si
☐ No

¿Por qué?

_____
_____
_____
_____
_____

10. Considera el servicio Streamel una herramienta novedosa dentro de las plataformas de educación virtual?

☐ Si
☐ No

¿Por qué?

_____
_____
_____
_____
_____

11. Qué cambios o adiciones considera necesarios realizar al servicio de streamel para mejorarlo.

_____
_____
_____

5.5 Análisis de las preguntas abiertas

Dentro de algunas de las preguntas que se formulan en el cuestionario se encuentran preguntas que son abiertas, es decir, permiten observar cual es la opinión más directa de los participantes.

En relación con la pregunta 4 los participantes manifestaron que en general la calidad de reproducción de los videos es regular pero como se había observado anteriormente esto se debe a que se necesitan personas más especializada en la creación de los mismos y presupuesto para lograrlo, además observan que en Internet hay muchos sitios que ofrecen mucha calidad en la reproducción de los videos y los participantes están acostumbrados a ingresar a este tipo de contenidos en internet.

En relación con la pregunta 5 los participantes manifestaron útil la funcionalidad de adaptación de contenido ya que no todos tenían la misma velocidad de conexión, y en muchos casos era más rápido cuando utilizaban el recurso de sólo audio, sobre todo cuando el ingreso lo realizaban desde la universidad del cauca, aunque esto no es por velocidad de conexión de la universidad sino por las restricciones de firewall y puertos que permiten los servidores , RTMP puerto 1935 (No permitido)-RTMPT puerto 80 (permitido).

Con respecto a la pregunta 8 los participantes manifestaron que aunque utilizan el servicio de comentarios, no lo hacen siempre, y esto es debido a que según los participantes el servicio no está bien diseñado ya que como se menciona en las variables, la presentación de los comentarios no es la más adecuada, por ejemplo cuando se realizaban los comentarios algunos quedaban encima o muy juntos y no era posible observarlos. Es de notar que como el servicio no era un curso formal de educación ni obligatorio pues los participantes no veían la necesidad de realizar los comentarios, en general hacían un solo comentario de prueba del servicio y no volvían a utilizarlo.

Con respecto a la pregunta 9 los participantes consideran en su gran mayoría que el servicio de video streaming “streamel” es una herramienta útil para alcanzar objetivos de aprendizaje y es una herramienta que permita acceder de una forma más ATRACTIVA a los contenidos ofrecidos por la plataforma EVA que los que ofrece tradicionalmente.

Con respecto a la pregunta 10 los participantes consideran en su gran mayoría que la herramienta no es novedosa, ya que en Internet se ofrece muchos sitios que permiten utilizar servicios de este tipo. Inclusive se menciona a *Youtube* que aunque no es un sistema dedicado a la educación, ofrece herramientas didácticas para utilizar en este campo. También se debe notar que en cuanto específicamente a los LMSs este tipo de servicio ya ha venido siendo utilizado.

En cuanto a la última pregunta los participantes opinan que al servicio le serian convenientes realizarle los siguientes cambios o modificaciones para mejorarlo.

- Mejor diseño en la interfaz gráfica, ya que colores y estructura no es de mucha calidad.
- Estructurar mejor el servicio de comentarios, por el análisis anteriormente mencionado.
- Mejorar acceso o la colocación de donde se ejecuta el servicio ya que algunas veces no se tiene claridad de donde hacerlo.
- Mejorar la calidad de los videos.
- El servicio se reduce a una solo contenido, una sola clase o charla por ejemplo, permitir que se pueda escoger o visualizar mayor contenido, diferentes charlas o clases. como ocurre en otras herramientas ofrecidas en internet.

La encuesta fue realizada a 23 personas que accedieron al servicio e interactuaron con él en los meses de diciembre de 2009 y enero del año 2010.

Bibliografía

[1] The Internet Engineering Task Force, "UDP - User Datagram Protocol, RFC 768."

[2] The Internet Engineering Task Force, "TCP - Transmission Control Protocol, RFC 793."

[3] J. van der Merwe, S. Sen, and C. Kalmanek, "Streaming video traffic: Characterization and network impact," in *Seventh International Web Content Caching and Distribution Workshop*, 2003.

[4] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia Streaming via TCP: An Analytic Performance Study," University of Massachusetts, 2004.

[5] C. Krasic and J. Walpole, "Priority-progress streaming for quality-adaptive multimedia," in *ACM Multimedia Doctoral Symposium*, Ottawa,Canada, 2001.

[6] N. Seelam, P. Sethi, and W. chi Feng, "A hysteresis based approach for quality, frame rate, and buffer management for video streaming using TCP," in *Management of Multimedia Networks and Services*, 2001.

[7] P. d. Cuetos, P. Guillotel, K. W. Ross, and D. Thoreau, "Implementation of adaptive streaming of stored MPEG-4 FGS video over TCP," in *International Conference on Multimedia and Expo (ICME02)*, 2002.

[8] P. de Cuetos and K. W. Ross, "Adaptive rate control for streaming stored fine-grained scalable video," in *NOSSDAV*, 2002.

[9] The Internet Engineering Task Force, "HTTP - HyperText Transfer Protocol, RFC 2616."

[10] M. Johanson, "A RTP to HTTP video gateway," in *Proceedings of the 10th international conference on World Wide Web* Hong Kong, Hong Kong: ACM, 2001.

[11] M. L. Husni Fahmi, S. Sedigh-Ali, A. Ghafoor, P. Liu, and L. H. Hsu, "Proxy Servers for Scalable Interactive Video Support " in *Computer*. vol. 34: IEEE Computer Society, 2001, pp. 54-60.

[12] The Internet Engineering Task Force, "RTP - A Transport Protocol for Real-Time Applications, RFC 3550."

[13] The Internet Engineering Task Force, "RTSP - Real Time Streaming Protocol, RFC 2326."

[14] Adobe Systems Inc, "AMF 3 Specification," Rev. 2008.

[15] Adobe Systems Inc, "RTMP," Último Acceso: Junio 2009.
[En línea] Disponible en:
http://www.adobe.com/devnet/rtmp/

[16] R. Reinhardt, "Adobe Flash CS3 Professional Video Studio Techniques," 1 ed: Adobe Press, 2007.

[17] Adobe Systems Inc, "HTTP Tunneling protocols," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://kb2.adobe.com/cps/166/tn_16631.html

[18] J. Bauch, "RTMPT protocol," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://www.joachim-bauch.de/tutorials/red5/SPEC-RTMPT.html

[19] Adobe Systems Inc, "Overview of streaming with Flash Media Server 3," Último Acceso: Junio 2009.
[En línea] Disponible en:
http://www.adobe.com/devnet/flashmediaserver/articles/overview_streaming_fms3_02.html

[20] K. Towes and T. Green, "Video content protection measures enabled by Adobe® Flash® Media Interactive Server 3," Adobe Systems Inc., 2008.
[21] Microsoft Corporation., "Microsoft Media Server (MMS) Protocol Specification," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/%5BMS-MMSP%5D.pdf
[22] Microsoft Corporation, "Windows Media Services," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://www.microsoft.com/windows/windowsmedia/forpros/server/version.aspx
[23] Microsoft Corporation, "[MS-WMSP]: Windows Media HTTP Streaming Protocol Specification," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/%5BMS-WMSP%5D.pdf
[24] International Organisation For Standardisation Organisation Internationale De Normalisation ISO/IEC JTC1/SC29/WG11 Coding Of Moving Pictures And Audio, "MPEG-1," 1996.
[25] TechCrunch, "iTunes Sells 6 Billion Songs, And Other Fun Stats From The Philnote," Último Acceso: Junio 2009.
[En línea] Disponible en:
http://www.techcrunch.com/2009/01/06/itunes-sells-6-billion-songs-and-other-fun-stats-from-the-philnote/
[26] International Organisation For Standardisation Organisation Internationale De Normalisation ISO/IEC JTC1/SC29/WG11 Coding Of Moving Pictures And Audio, "MPEG-2," 2000.
[27] International Organisation For Standardisation Organisation Internationale De Normalisation ISO/IEC JTC1/SC29/WG11 Coding Of Moving Pictures And Audio, "MPEG-4 Standard," 2002.
[28] International Organisation For Standardisation Organisation Internationale De Normalisation ISO/IEC JTC1/SC29/WG11 Coding Of Moving Pictures And Audio, "MPEG-7 Standard," 2004.
[29] R. García and O. Celma, "Semantic Integration and Retrieval of Multimedia Metadata," 2006.
[30] ITU - International Telecomunication Union, "H.263 : Video coding for low bit rate communication ", Rev. 2005.
[31] B. Girod, E. Steinbach, and N. Faerber, "Comparison of the H.263 and H.261 video compression standards," in *Standards and Common Interfaces for Video Information Systems*, Philadelphia, 1995, pp. 233-251.
[32] N. Faerber, B. Girod, and E. Steinbach, "Performance of the H.263 video compression standard," in *Special Issue on Recent Development in Video: Algorithms, Implementation and Applications*, 1997, pp. 101-111.
[33] ITU - International Telecomunication Union, "H.264 : Advanced video coding for generic audiovisual services," Rev. 2009.
[34] Microsoft Corporation, "Códecs de audio y vídeo de Windows Media 9 Series," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://www.microsoft.com/windows/windowsmedia/es/9series/codecs.aspx
[35] Microsoft Corporation, "Contenedor de archivos eficaz," Último Acceso: Octubre 2009.
[En línea] Disponible en:

http://www.microsoft.com/windows/windowsmedia/es/format/robust.aspx
[36] BetaNews, "Microsoft to Open Windows Media Video," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://www.betanews.com/article/Microsoft-to-Open-Windows-Media-Video/1063188729
[37] BetaNews, "Microsoft VC-1 Codec Now a Standard," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://www.betanews.com/article/Microsoft-VC1-Codec-Now-a-Standard/1144097224
[38] On2 Technologies Inc., "On2 VP6 for Flash 8 Video," 2005.
[39] On2 Technologies Inc., "On2 video," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://www.on2.com/index.php?564
[40] On2 Technologies Inc., "Flash Video Codec Comparison," 2008.
[41] A. W. S. LLC, "Amazon EC2 Instance Types," Último Acceso: Agosto 2009.
[En línea] Disponible en:
http://aws.amazon.com/ec2/instance-types/
[42] G. Perry, "What are Amazon EC2 Compute Units?," Último Acceso: Enero 2010.
[En línea] Disponible en:
http://gevaperry.typepad.com/main/2009/03/figuring-out-the-roi-of-infrastructureasaservice.html
[43] Jana Technology Services, "Amazon EC2 Instances and cpuinfo," Último Acceso: Enero 2010.
[En línea] Disponible en:
http://www.cloudiquity.com/2009/01/amazon-ec2-instances-and-cpuinfo/
[44] J. Hilton, "Flash Media Server, Wowza Media Server and Red5 connection limit test," Último Acceso: Enero 2010.
[En línea] Disponible en:
http://jakehilton.com/?q=node/64
[45] Smaxe Ltd, "JUV RTMP LoadTester (Lite)," Último Acceso: Diciembre 2009.
[En línea] Disponible en:
http://www.smaxe.com/product.jsf?id=juv-rtmp-loadtester-lite
[46] Sun Microsystems, "The OpenOffice.org," Último Acceso: Noviembre 2009.
[En línea] Disponible en:
http://wiki.services.openoffice.org/wiki/Main_Page
[47] Softland, "doPDF," Último Acceso: Diciembre 2009.
[En línea] Disponible en:
http://www.dopdf.com/
[48] Sorenson Media Inc., "Sorenson Spark," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://www.sorensonmedia.com/video-codec/
[49] byGui4Cli, "Avanti - FFmpeg / Avisynth GUI," Último Acceso: Noviembre 2009.
[En línea] Disponible en:
http://avanti.arrozcru.com/
[50] eRightSoft, "Super@: Simplified Universal Player Encoder & Renderer," Último Acceso: Noviembre 2009.
[En línea] Disponible en:
http://www.erightsoft.com/SUPER.html
[51] Sparrow InterActive, "Sparrow InterActive," Último Acceso: Enero 2010.
[En línea] Disponible en:
http://www.sparrowi.com/contact_us.html

[52] Ph.D, J. Lamos, and P. Parrish, "Pedagogical Approaches: Scenario-based Learning," in *CALMet '99*, Helsinki, Finland, 1999.
[53] University College London, "Scenario-based learning," Último Acceso: Enero 2010.
[En línea] Disponible en:
http://www.ucl.ac.uk/isd/staff/e-learning/tools/sbl
[54] Advance Distributed Learning, "SCORM 1.2 Specification," 2009.
[55] .LRN Consortium, ".LRN: Learn, Research, Network," Último Acceso: Octubre 2009.
[En línea] Disponible en:
http://www.dotlrn.com/
[56] W3C, "Document Object Model, DOM," 2005.
[57] AICC, "Aviation Industry CBT (Computer-Based Training) Committee," Último Acceso: Febrero 2010.
[En línea] Disponible en:
http://aicc.org/dev/index.php/faq.html
[58] OpenACS, "*Learning Object Repository Service, LORS*," Último Acceso: Abril 2010.
[En línea] Disponible en:
http://openacs.org/repository/apm/packages/LORS
[59] BlackBoard, "BlackBoard LMS," Último Acceso: Enenro 2009.
[En línea] Disponible en:
http://www.blackboard.com/us/index.bbb
[60] OpenACS, "LORS Management," Último Acceso: Abril 2010.
[En línea] Disponible en:
http://openacs.org/repository/apm/packages/lorm