

Artículo de Investigación

REVISTA  
Ingeniería

# Construcción y evaluación de servicios interactivos en entornos de TVDi

## Construction and evaluation of interactive services in TVDi environments.

Gabriel Elías Chanchí Golondrino<sup>1\*</sup>, José Luis Arciniegas Herrera<sup>1</sup>,  
Wilmar Yesid Campo Muñoz<sup>2</sup>

<sup>1</sup>Universidad del Cauca, <sup>2</sup>Universidad del Quindío

\* gabrielc@unicauca.edu.co

Received: 09/06/2015. Modified: 09/11/2015. Accepted: 31/12/2015

### Resumen

**Contexto:** Con el fin de ampliar el abanico de oportunidades de la educación virtual, es importante considerar dos aspectos tecnológicos relevantes: el gran potencial de la penetración de la televisión y el auge de servicios de la Web 2.0 en redes sociales y comunidades en Internet, espacios en los cuales los usuarios comparten y generan conocimiento alrededor de una temática. Así, se hace necesario definir la forma adecuada de implementar y desplegar dichos servicios interactivos en entornos de televisión, dadas las características particulares de este escenario.

**Método:** Con el propósito de guiar el proceso de construcción de servicios interactivos de televisión, en este artículo se propone un esquema para el consumo de servicios para escenarios de televisión digital interactiva (TVDi), el cual fue adaptado a partir del estilo arquitectónico REST-JSON (*Representational State Transfer – Javascript Object Notation*).

**Resultados:** Como resultados del uso del esquema propuesto, se construyeron los servicios de chat, tablón de mensajes y acceso a correo electrónico, en los escenarios de televisión digital terrestre (TDT) y TV Móvil del proyecto ST-CAV (Servicios de T-Learning para el soporte de Comunidades Académicas Virtuales). Asimismo, otro de los resultados del presente artículo fue la evaluación de los servicios implementados, mediante pruebas de tráfico de red y consumo de memoria.

**Conclusiones:** De acuerdo a los tiempos de procesamiento y respuesta obtenidos en la evaluación de los servicios interactivos implementados, es posible concluir que el esquema planteado en este artículo puede considerarse como una alternativa adecuada para el diseño y construcción de servicios en escenarios de TVDi, permitiendo la convergencia con aplicaciones de Internet (Web 2.0).

**Palabras clave:** comunidades académicas virtuales, REST-JSON, servicios de la Web 2.0, Televisión Digital Interactiva (TVDi).

Open access



Citación: Chanchí, G. E., Arciniegas, J. L. y Campo, W. Y. (2016). Construcción y evaluación de servicios interactivos en entornos de TVDi. En: Ingeniería, Vol. 21, No. 1, pp. 63 –82  
© los autores; titular de derechos de reproducción Universidad Distrital Francisco José de Caldas.  
En línea DOI: <http://dx.doi.org/10.14483/udistrital.jour.reveng.2016.1.a05>

## Abstract

**Context:** In order to expand the range of virtual education opportunities, it's important to consider two technological aspects: the great potential of TV penetration and the Web 2.0 services boom in social networks and Internet communities, spaces in which users share and generate knowledge around a topic. So, it's necessary to define how to implement and deploy these interactive services in television environments, given the particular characteristics of this scenario.

**Method:** For the purpose of guiding the building process of interactive services, in this paper we propose a scheme of service consumption in scenarios of interactive digital television (iDTV), which has been adapted from the REST-JSON (*Representational State Transfer – Javascript Object Notation*) architectural style.

**Results:** As a result of the use of the proposed scheme, we implemented the message board, chat and e-mail services, in the television scenarios of digital terrestrial television (DTT) and Mobile TV of the ST-CAV project (T-Learning services for the support of virtual academic communities). Likewise, another result obtained in the present paper, was the evaluation of the implemented services through network traffic and memory consumption tests.

**Conclusions:** According to processing and response times obtained in the evaluation of implemented interactive services, the presented scheme can be considered as a viable alternative for the design and building of services in iDTV scenarios, allowing convergence with Internet applications (Web 2.0).

**Key words:** interactive digital television, REST-JSON, virtual academic communities, Web 2.0 services.

## 1. Introducción

Dado el proceso de migración de los sistemas analógicos a los digitales [1], la Televisión Digital Abierta (TDA), ha sido concebida como un servicio de cubrimiento cercano al 100 % en cada uno de los países. Además, gracias al soporte de interactividad que ofrece la TDA, esta se convierte en una de las tecnologías llamadas a reducir la brecha digital, mediante posibles entornos interactivos que promuevan el aprendizaje de forma personalizada, como es el caso del T-Learning [2].

En [3] los autores muestran cómo diferentes investigaciones predicen el rápido desarrollo de la TVDi en Europa, exponen cómo los usuarios demandan un mayor valor agregado sobre la televisión convencional. Así, el artículo describe una arquitectura cuyo objetivo es proveer un *Framework* digital interactivo que permita la convergencia de los servicios de la televisión por *Broadcast* con la televisión sobre IP. En [4] se presenta una arquitectura para proveer servicios interactivos para un sistema de TVDI, asegurando una comunicación estandarizada entre las aplicaciones del cliente y sus servicios interactivos, a través de un canal externo. Los autores usan como caso de estudio el contexto de la intención de la inclusión social de Brasil.

Por otra parte, la marcada evolución de Internet en los últimos años, ha permitido el intercambio y generación de una gran cantidad de información de forma ágil y flexible, gracias a la tendencia de los servicios actuales de la red de redes por fomentar la creación de contenidos de forma colaborativa y comunitaria. Entre estos servicios se destacan: foros, *wikis*, blogs, chat, entre otros, los cuales son conocidos también como herramientas de la Web 2.0. Dichas herramientas por lo general se encuentran asociadas a entornos virtuales de aprendizaje (E-Learning), redes sociales y comunidades académicas virtuales (CAV) [5][6]. Dadas las características de integración, reutilización y flexibilidad necesarias para el diseño e implementación de estos servicios, los esquemas de consumo más difundidos para este propósito, se basan en el uso de servicios web [7].

De acuerdo a lo anterior, resulta importante potenciar las ventajas de la interactividad en TD, aprovechando el auge y la aceptación de los servicios y tecnologías propias de la web 2.0. Dentro de las tecnologías de implementación de esquemas de consumo de servicios más

usados para las herramientas de la Web 2.0, se destacan dos aproximaciones tecnológicas: el protocolo Simple Object Access Protocol (SOAP, por sus siglas en inglés) y el estilo arquitectónico Representational State Transfer (REST, por sus siglas en inglés). La diferencia básica entre estos dos métodos de consumo de servicios, se da en cuanto al tipo de mensajes que se intercambian. Para el caso de SOAP, los mensajes son en formato Web Service Description Language (WSDL, por sus siglas en inglés), mientras que en el caso de REST, el tipo de mensaje más difundido es JavaScript Object Notation (JSON, por sus siglas en inglés), el cual basa su sintaxis en el lenguaje JavaScript. Los mensajes WSDL al estar basados en el lenguaje Extensive Markup Language (XML, por sus siglas en inglés) son complejos comparados con los de tipo JSON, razón por la que muchos de los servicios de las CAV y redes sociales (Facebook, Twitter) han sido implementados y tienen interfaces abiertas para desarrollo con el estilo arquitectónico REST [8].

En [9] se muestra cómo, mediante los servicios web basados en Single Sign-on (SSO, por sus siglas en inglés), se provee a los usuarios un fácil acceso a las aplicaciones y a los recursos de red, donde la parte de seguridad es tratada bajo JSON. Además, en este trabajo sustituyen el XML por JSON debido a su fácil compresión para los seres humanos y su mayor velocidad de procesamiento para los equipos de cómputo. En [10] se demuestra cómo usando XML se requiere un proceso de interpretación o análisis, mientras que usando JSON, se presenta un nuevo enfoque, pasando por alto la necesidad de procesamiento del lenguaje, que lo convierte en una solución más ágil. En [11] se describe la construcción de aplicaciones web de alto desempeño computacional, High Performance Computing (HPC, por sus siglas en inglés) usando servicios web que aprovechan las ventajas del estilo arquitectónico REST-JSON. Donde, para lograr una completa funcionalidad de las aplicaciones web, se tienen en cuenta los estándares de la Web 2.0.

A pesar de que los esquemas de consumo de servicios usados han sido ampliamente difundidos y aplicados en Internet, en el contexto de la televisión se deben considerar algunas restricciones adicionales, relacionadas con las capacidades de procesamiento, los tiempos de respuesta y memoria de los dispositivos de acceso, con el fin de permitir la eficiencia en el consumo de los servicios. Como aporte principal, este artículo propone la adaptación de un esquema de consumo de servicios para escenarios de TVDi, tomando como referencia el estilo arquitectónico REST. El esquema propuesto parte del diseño preliminar presentado por los autores en [12]. Así, el presente artículo realimenta las ideas propuestas en [12], permitiendo extender los escenarios de implementación y validar los servicios desarrollados a través del esquema propuesto. De igual forma es importante mencionar que el esquema propuesto y los servicios implementados a partir de este, han sido realimentados por las directrices de diseño de aplicaciones usables en TVDi propuestas en [13], desarrolladas también en el laboratorio de televisión de la Universidad del Cauca a partir de la adaptación y aplicación de heurísticas de usabilidad a escenarios de televisión. El propósito de este esquema es guiar el proceso de diseño e implementación de servicios interactivos de televisión, de manera independiente al campo de aplicación.

El esquema presentado en esta investigación fue usado en el proceso de construcción de los servicios asociados a las comunidades académicas virtuales (CAV) de los diferentes escenarios (televisión digital terrestre y Tv móvil) del proyecto ST-CAV (Servicios de T-Learning para el soporte de comunidades académicas virtuales) [14], desarrollado en el laboratorio experimental de TVDi de la Universidad del Cauca [15]. La importancia de las comunidades en el proceso de aprendizaje es abordada en [16], donde se investiga las necesidades de los usuarios para el desarrollo de una comunidad virtual (CV) para el curso de SAP UCC (University Com-

petence Center) del Grupo de Usuarios. El Grupo de Usuarios está compuesto por profesores y estudiantes para propósitos educativos. El objetivo de la CV es mejorar la comunicación y la cooperación entre los profesores y apoyar el desarrollo sistemático de las innovaciones, en el ámbito de la enseñanza de esta planificación de recursos empresariales de software. Asimismo, en [17] se estudia el proceso de aprendizaje y se presenta la metodología de diseño de un sistema interactivo de E-Learning basado en IPTV, se introduce un marco de servicios para IPTV basados en E-Learning. Además, se discuten las estrategias tradicionales de conversión de recursos de aprendizaje basados en la web a la plataforma de IPTV.

El proyecto ST-CAV, en el cual se enmarca esta investigación, tuvo por objetivo dar soporte a una CAV desde diversos escenarios televisivos (TDT, TV Móvil e IPTV), con el propósito de apoyar los procesos de aprendizaje y construcción de conocimiento en televisión (T-Learning), a través de servicios de la Web 2.0. Dentro de los servicios implementados en el proyecto y presentados en este artículo, se destacan: chat, tablón de mensajes o micro-blog y acceso al correo electrónico, los cuales fueron desplegados en escenarios de televisión digital terrestre y tv móvil. Finalmente, este artículo presenta como aporte adicional, la evaluación de los servicios implementados según el esquema propuesto, a través de pruebas de tráfico y pruebas de consumo de memoria, las cuales tienen por función verificar la eficiencia del esquema para escenarios de TVDi. Estas pruebas están de acuerdo a lo propuesto en [18], en donde se presenta un modelo de tráfico de los servicios soportados por una CAV, los cuales generan un comportamiento propio y diferente que depende del comportamiento de los usuarios y los horarios de acceso a los servicios.

La estructura de este artículo es la siguiente: en la sección 2 se muestran un conjunto de conceptos utilizados en el presente artículo. En la sección 3 se presenta el esquema de servicios para TVDi basados en servicios web REST-JSON. En la sección 4 se presentan los resultados más relevantes del escenario de experimentación realizado en la Universidad del Cauca. En la sección 5 se muestran las pruebas de tráfico y consumo de memoria sobre los servicios desarrollados a partir del esquema. Por último, en la sección 6 se muestran las conclusiones y trabajos futuros derivados del presente trabajo.

## 2. Marco teórico

A continuación, se presentan algunos conceptos relevantes, que se tuvieron en cuenta para el desarrollo de la presente investigación. Dentro de estos se encuentran: CAV, Servicios Web, REST, JSON.

### 2.2. CAV

Una CAV es definida como “uno o varios grupos de individuos que están vinculados por intereses en común, que tienen la capacidad de poseer una fuerza de voluntad autónoma y están comprometidos en un proceso de aprendizaje continuo, cuyo principal objetivo es el de construir conocimientos de forma compartida utilizando las TIC como un medio de expresión, como herramienta de comunicación, como recurso didáctico e incluso como instrumento de gestión” [5]. Para el caso de una CAV en TVDi, se plantea que el proceso de construcción de conocimiento sea impulsado por los contenidos multimedia aportados por los miembros de

la comunidad, así como por el conjunto de servicios que buscan promover la participación en torno a esos contenidos [19].

## 2.2. Servicios web

Según la W3C, un servicio web es un sistema software diseñado para soportar una interacción interoperable entre diferentes equipos en red [20]. Estos suelen ser librerías (API's) que son accedidas desde Internet y se ejecutan en el equipo que los aloja, cumpliendo una función determinada y permitiendo la integración con otros componentes o funcionalidades. Dentro de las implementaciones comunes de servicios web se encuentran SOAP y REST. El primero hace referencia al protocolo usado para la comunicación entre cliente y servidor intercambiando mensajes basados en XML (WSDL), mientras que en el segundo caso los mensajes son por lo general en formato JSON. En ambos casos tanto el cliente como el servidor deben conocer el formato de los mensajes para poder encapsular y des-encapsular peticiones y respuestas [21].

## 2.3. REST

El estilo arquitectónico Representational State Transfer (REST, por sus siglas en inglés), plantea una arquitectura cliente-servidor, en la cual un servicio es visto como un recurso y es identificado a través de una dirección Uniform Resource Locator (URL, por sus siglas en inglés), mediante la cual puede ser consumido. Para acceder a estos servicios web, se hace uso de mensajes en formato simple, los cuales se intercambian entre cliente y servidor [22]. REST define a partir de HTTP, cuatro métodos: GET, PUT, DELETE y POST, de los cuales los más utilizados son: GET y PUT. El primero de los métodos es usado para enviar la representación de un recurso o servicio al cliente, mientras que el otro es usado para transferir el estado de un cliente al recurso [23]. Para el intercambio de información entre cliente y servidor a través de REST, se puede hacer uso de diversos formatos y lenguajes: XML, HTML, JSON.

## 2.3. JSON

JSON es un formato ligero basado en de texto, cuya sintaxis es tomada de JavaScript. Debido a su sencillez es fácil generar y procesar un documento con este formato [22]. JSON usa convenciones para el manejo de datos, que son comunes a la familia de lenguajes: C, C++, Java, Perl, Python, etc. Lo anterior hace que JSON sea ideal para el intercambio de datos entre aplicaciones cliente servidor [24]. Un mensaje JSON está constituido por dos estructuras básicas [25]: la primera es una colección de estructuras nombre-valor, las cuales son conocidas en varios lenguajes como: diccionarios, tablas, hash, listas de claves o arreglos asociativos; y la segunda es una lista ordenada de valores (arreglos, vectores, listas, etc.). Estas estructuras son usadas para conformar los mensajes de intercambio entre cliente y servidor, en los cuales se define un protocolo interno con la representación: palabra clave - valor.

El presente trabajo hace uso de las definiciones anteriores, al proveer un conjunto de servicios basados en los conceptos de la Web 2.0 y vincularlos al contexto de la televisión digital interactiva, con el ánimo de promover la participación y, por ende, la generación de conocimiento alrededor de una CAV. Estos servicios fueron implementados usando el estilo arquitectónico REST y el lenguaje JSON como formato para el intercambio de mensajes.

### 3. Esquema de servicios para televisión digital interactiva

Dentro del proyecto ST-CAV se escogió un conjunto de servicios (tablón de mensajes o micro-blog, chat, acceso a correo electrónico), tomados de la Web 2.0 para dar soporte a las CAV en televisión, los cuales fueron diseñados e implementados de acuerdo al estilo arquitectónico REST-JSON. A continuación, se presentan las condiciones de diseño de estos servicios y el escenario de experimentación usado para los mismos.

Considerando que los mensajes intercambiados a través de REST deben contener la información necesaria para el funcionamiento de cada servicio en el escenario de televisión, es importante definir dentro del formato de los mensajes JSON, el conjunto de parejas nombre-valor relacionadas con la funcionalidad de los servicios. En la figura 1 se presenta un ejemplo de mensaje JSON perteneciente al servicio de tablón de mensajes del proyecto ST-CAV. En este mensaje “nombre” puede hacer referencia a los atributos: *login*, mensaje, hora; mientras que “valor” se refiere a una cadena de texto asociada a alguno de los atributos mencionados.

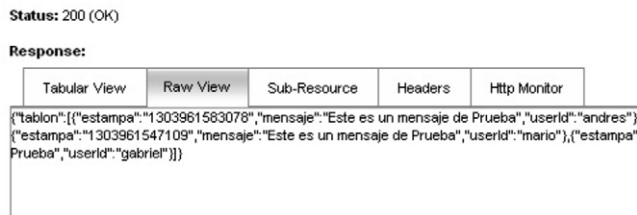


Figura 1. Mensaje en formato JSON.

En TVDi los servicios son consumidos mediante el canal de retorno usando el protocolo IP, e integrados por el STB (*Set-top box*) o dispositivo móvil en la interfaz del televisor, o en la pantalla del dispositivo móvil. Los mensajes a intercambiar entre los clientes de televisión y el servidor de aplicaciones, contienen características funcionales más no de presentación, razón por la cual cada cliente (dispositivo móvil o STB), recupera la información a partir de los mensajes e implementa de manera independiente la lógica de presentación.

En la figura 2 se presenta el esquema de consumo de servicios propuesto para el contexto de las CAV en TVDi, considerando el estilo arquitectónico REST-JSON. Cada servicio es representado como una instancia o recurso: R1, R2, ..., RN, dentro del repositorio de servicios o servidor de aplicaciones, de tal forma que estos se encuentran en la capacidad de interactuar entre sí para procesos de composición.

A cada recurso se le asigna una dirección URL desde la cual, el cliente: 1, 2, ..., N, puede acceder vía Internet y recibir el mensaje correspondiente a cada recurso. El mensaje recibido por cada cliente contiene una estructura con un conjunto de parejas nombre-valor, conocidas por los clientes y el servidor de aplicaciones (figura 1). Tales estructuras son desencapsuladas por cada cliente, y de acuerdo al nombre del atributo, son clasificadas funcionalmente y presentadas en la interfaz correspondiente. Cada tipo de cliente (STB, móvil) debe contar con la librería apropiada según las características de su hardware, para así permitir la interpretación de los mensajes JSON.

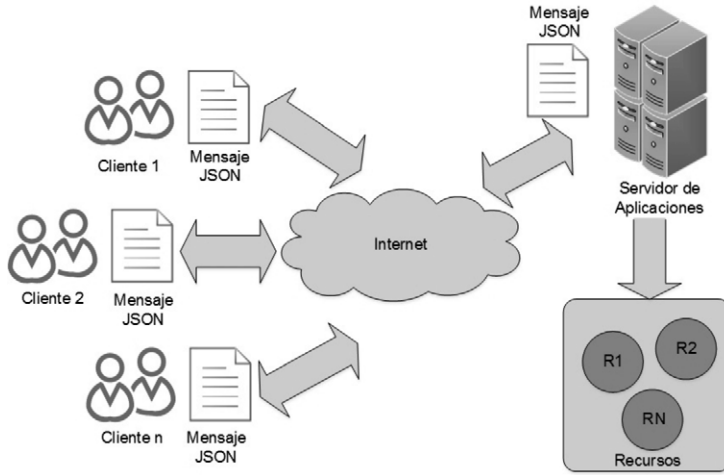


Figura 2. Esquema de consumo de servicios REST-JSON.

### 3.1. Escenario de experimentación

El escenario de experimentación utilizado para la implementación y despliegue de los servicios (figura 3), está formado por los siguientes componentes: servidor de difusión, servidor de aplicaciones, STB TDT – MHP (*Multimedia Home Platform*), dispositivo móvil DVB-H (*Digital Video Broadcasting Handheld*), *Access Point* y *Switch*. A continuación, se describe el funcionamiento de estos componentes.

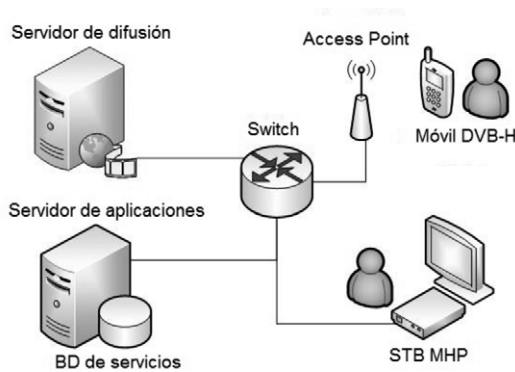


Figura 3. Escenario de experimentación.

El servidor de difusión es el encargado de almacenar, adecuar y transmitir el contenido multimedia vía *Broadcast*, para lo cual cuenta con la herramienta libre OpenCaster [26] y con la tarjeta moduladora de televisión: Dektec DTA 110T. El servidor de aplicaciones corresponde al equipo de cómputo encargado de almacenar los servicios, recibir y procesar las peticiones REST por parte de los clientes. En el ámbito del proyecto ST-CAV, se utilizaron dos

servidores para el despliegue de los servicios, un servidor basado en lenguaje Java (Glashfish) y un servidor basado en lenguaje Python (web.py). Ambos servidores cuentan con el soporte para la creación de aplicaciones usando el estilo arquitectónico REST-JSON. El *Switch* y El *Access Point* tienen la función de distribuir Internet de forma cableada e inalámbrica al STB y al dispositivo móvil respectivamente, permitiendo la conexión por canal de retorno.

El STB MHP de TDT o cliente de televisión, es el encargado de recibir la señal de televisión vía *Broadcast* y adecuarla para ser presentada en la pantalla del televisor. De igual forma, el STB puede acceder a los recursos del servidor de aplicaciones mediante la URL designada por cada servicio, para lo cual usa las librerías de conexión que provee el middleware MHP y la librería RestClient[27], desarrollada para escenarios JavaME y usada para las peticiones y conexiones REST. Cada vez que accede a un recurso, el STB recibe un mensaje en formato JSON, el cual es interpretado usando la librería compatible con MHP: json-simple.

A partir de la información interpretada del mensaje, el STB presenta en la pantalla del televisor. El dispositivo móvil es el encargado de recibir la señal de televisión DVB-H, la cual es adaptada en el servidor de difusión. Asimismo, este componente se encarga de acceder a los servicios de la CAV mediante la URL de cada recurso. Cada vez que el dispositivo móvil accede a un recurso, recibe un mensaje en formato JSON, lo procesa y lo presenta en la pantalla del dispositivo. Este mensaje contiene la información del servicio, sin incluir la lógica de presentación. El dispositivo móvil debe soportar el estándar DVB-H, así como el estándar 802.11 para conexiones inalámbricas y las librerías necesarias para hacer peticiones HTTP de tipo GET y PUT y para decodificar los mensajes en formato JSON. Los dispositivos móviles usados para las pruebas de consumo de servicios fueron: Nokia N96, N95 (compatibles con el estándar DVB-H).

## 4. Resultados

En esta sección se presentan los resultados de implementación y evaluación de los servicios interactivos del escenario de experimentación. Dentro de estos servicios se destacan principalmente tres: micro-blog o tablón de mensajes, chat y acceso a correo electrónico.

En las figuras 4, 5 y 6, se presentan diferentes versiones de interfaz del proyecto ST-CAV. Dentro de estas se pueden visualizar a la izquierda el contenido multimedia transmitido desde el servidor de difusión y a la derecha una ventana desplegable con un conjunto de pestañas, cada una correspondiente a un servicio de soporte desplegado y consumido a partir del Servidor de Aplicaciones. Esta configuración obedece a las recomendaciones para despliegue de contenidos para televisión digital [28]. La ventana desplegable o ventana de servicios puede ocultarse a la derecha de la pantalla, de tal manera que el tamaño del contenido multimedia se adapta al tamaño de la pantalla. Esta ventana permanece siempre activa en la interfaz de televisión y es independiente del contenido multimedia que se esté transmitiendo. Para navegar a través de la pantalla y de los servicios, se hace uso de las flechas y del botón OK del control remoto.

La ventana desplegable presenta información relacionada con la lógica de los servicios de apoyo al contenido (foro, chat, acceso a correo electrónico). Estos servicios buscan facilitar la interacción de los usuarios en torno a la comunidad académica, propiciando la generación de conocimiento alrededor de las temáticas de la CAV, sin embargo, es necesario tener en cuenta que el tamaño de la fuente debe obedecer a las recomendaciones para presentación de





**Figura 4.** Interfaz principal del proyecto ST-CAV.

contenidos de televisión [28]. Así, los servicios fueron diseñados para manejar una cantidad reducida de texto y adaptarla al tamaño de la ventana de servicios, de tal forma que este pueda ser visualizado fácilmente a 3m de distancia. Dado que los servicios presentes en la ventana desplegable son independientes al contenido multimedia, reciben el nombre de servicios no asociados al contenido. De igual forma existen un conjunto de aplicaciones que hacen parte del contenido multimedia, tales como las encuestas y la información asociada al contenido multimedia. Estas aplicaciones no son consumidas a través del canal de retorno, sino que viajan con el contenido multimedia en el carrusel de objetos propio del estándar Digital Video Broadcasting (DVB, por sus siglas en inglés). A continuación se describen los servicios de micro-blog, chat y acceso a correo electrónico.

#### 4.1. Servicios de tablón y chat

El servicio de tablón de mensajes o micro-blog (figura 5), consiste en un mini foro similar al “microblogging” de Twitter, en el cual los miembros de una CAV pueden publicar mensajes con un restringido número de caracteres (considerando el uso de control remoto), al mismo tiempo que están visualizando un programa de televisión. La interfaz del servicio de tablón de mensajes permite listar los últimos cinco mensajes de los miembros de la comunidad y navegar a través de ellos mediante las flechas del control. Para ingresar mensajes al tablón o micro-blog, se hace uso de las teclas numéricas y del botón OK del control.

Los mensajes JSON que se intercambian en este servicio constan de un arreglo de parejas nombre-valor: `userid`, `mensaje`, `estampa` (figura 6), las cuales contienen el identificador de usuario, el mensaje a publicar y un dato de tipo long que representa una estampa de tiempo para obtener la hora a la que fueron enviados los mensajes. El tamaño del arreglo de mensajes es de cinco teniendo en cuenta que se listan en pantalla los últimos cinco mensajes de la comunidad. El STB se encarga de procesar el mensaje JSON, obteniendo cada uno de los valores de las parejas y presentando la información del servicio en la pestaña correspondiente.

El servicio de chat por su parte, tiene un estilo y funcionamiento semejante al servicio de tablón de mensajes, con la diferencia que no incluye comentarios sobre las publicaciones hechas por los miembros de la comunidad.



**Figura 5.** Servicio de Micro-Blog o Tablón.

Status: 200 (OK)

Response:

Tabular View	Raw View	Sub-Resource	Headers	Http Monitor
<pre>{   "tablón": [     {       "estampa": "1303964555531",       "mensaje": "Esta es una prueba corta para el artículo de 2011",       "userid": "mateo"     },     {       "estampa": "1303964532812",       "mensaje": "Esta es una prueba corta par 2011",       "userid": "lucas"     },     {       "estampa": "1303964498953",       "mensaje": "Esta es una prueba corta par 2011",       "userid": "eduardo"     },     {       "estampa": "1303964269687",       "mensaje": "Esta es una prueba corta",       "userid": "marcos"     }   ] }</pre>				

**Figura 6.** Mensaje JSON–Tablón.

## 4.2 Servicio de acceso a correo electrónico

El servicio de acceso a correo electrónico (figura 7), permite acceder a los encabezados de los mensajes del correo de Gmail, asociados a un miembro de la comunidad. La interfaz del servicio permite listar los últimos cinco mensajes de correo del miembro de la comunidad registrado y navegar a través de ellos mediante las flechas del control remoto. Para visualizar uno de los mensajes en detalle, se hace uso del botón OK del control.

Los mensajes JSON que se intercambian en este servicio constan de un arreglo de parejas nombre-valor con la información de los atributos del correo: fecha, asunto, encabezado del mensaje, remitente, etc., las cuales contienen la información de cada uno de los mensajes de la bandeja de entrada que están sin leer (figura 8). El tamaño del arreglo de mensajes de correo es de cinco, teniendo en cuenta que se listan en pantalla los últimos cinco mensajes de la comunidad. El STB se encarga de procesar el mensaje JSON, obteniendo cada uno de los valores de las parejas y presentando la información del servicio en la pestaña correspondiente. Para la implementación de este servicio en el servidor de aplicaciones, se hizo uso de la API gmail4j [29], la cual es compatible con el lenguaje JSON y permite obtener por defecto un conjunto de mensajes en este formato, a partir de los cuales se envían al cliente solo cinco.

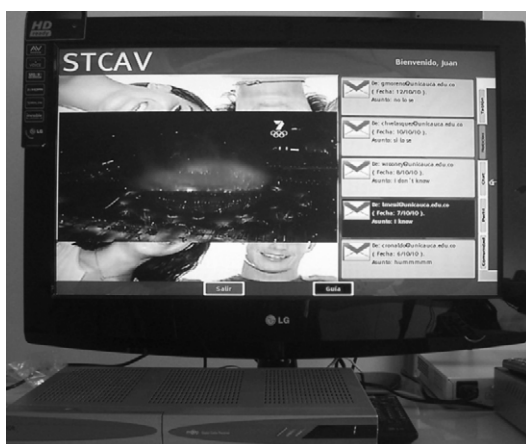


Figura 7. Servicio de acceso a correo.

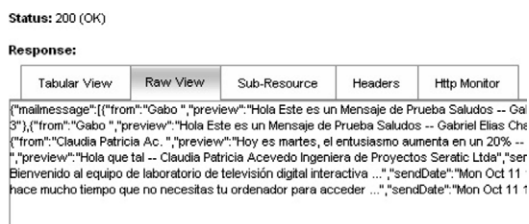


Figura 8. Interfaz de prueba del servidor–Correo.

### 4.3. Servicios desde el dispositivo móvil

Teniendo en cuenta el sistema operativo de los dispositivos DVB-H utilizados (N95 y N96), el cliente móvil se desarrolló usando el lenguaje Python a través de la librería PYS60, la cual es una implementación reducida del intérprete del lenguaje Python. Mediante el cliente desarrollado es posible invocar a los servicios de la CAV vía red inalámbrica usando la librería de conexión URLIB de Python. Una vez hecha la invocación a la URL del recurso o su servicio, se procede a la decodificación del mensaje JSON a través de la librería s60-json-library de Python [30]. Finalmente, la información obtenida del mensaje JSON es presentada en la pantalla del dispositivo (figura 9).

Dadas las ventajas del estilo arquitectónico REST-JSON, el consumo de servicios interactivos también puede realizarse en sistemas operativos como Android en el escenario de IPTV Móvil o WebTV. Lo anterior es posible gracias a la facilidad de procesamiento de los mensajes JSON y su compatibilidad con el lenguaje Javascript.

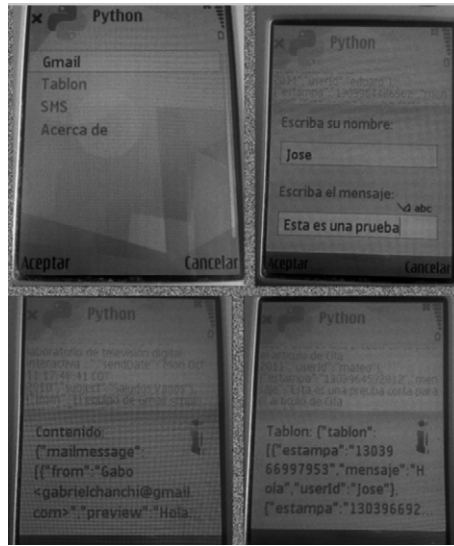


Figura 9. Servicio de Micro-Blog o Tablón.

## 5. Pruebas

En esta sección se presenta la evaluación de los servicios desarrollados usando el estilo arquitectónico REST-JSON. Esta evaluación fue realizada a través de pruebas de tráfico en la red y pruebas de consumo de memoria sobre el servidor de aplicaciones y sus servicios.

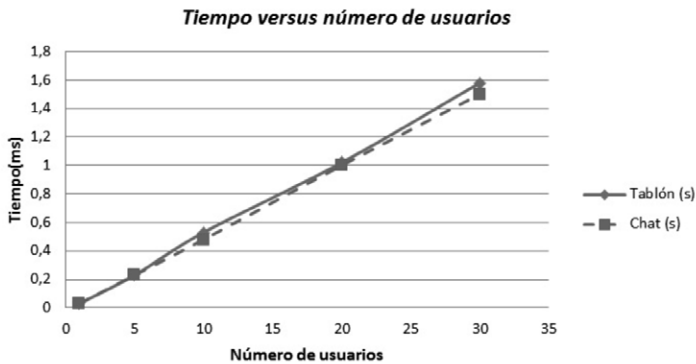
### 5.1. Análisis del tráfico de los servicios

El tráfico generado por los servicios de tablón de mensajes (microblog) y de chat corresponde a peticiones HTTP. La captura de este tráfico se realizó usando el analizador de protocolos Wireshark [31]. Se tomaron muestras con 1, 5 y 10, 20 y 30 usuarios pertenecientes a dos comunidades del proyecto ST-CAV. En la tabla 1 se presentan los tiempos de respuesta obtenidos en el lado del cliente para peticiones simultáneas a los servicios de tablón de mensajes y chat. En ambos casos, el tiempo de respuesta para un solo usuario es de 0,03 s, mientras que para 30 usuarios el tiempo de respuesta es de 1,58 s para el servicio de tablón y 1,50 s para el servicio de chat.

En la tabla 1 se observa que los incrementos de los tiempos de respuesta al aumentar el número de usuarios, son lineales para los dos servicios (figura 10). En el caso del servicio de tablón de mensajes, la pendiente estimada es de 0,053 segundos/usuario, mientras que, en el caso del servicio de chat, la pendiente de la curva estimada es de 0,051 segundos/usuario. El anterior factor permite determinar en promedio, el incremento en tiempo por cada usuario conectado a los servicios. Los anteriores datos están de acuerdo con los tiempos de respuesta en el procesamiento de un mensaje para el formato de intercambio JSON, los cuales oscilan alrededor de los 30 ms [33].

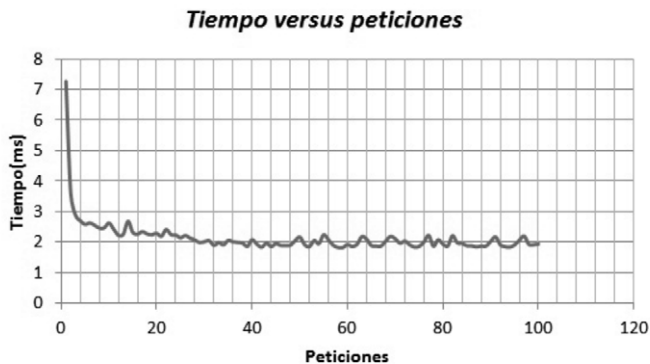
**Tabla I.** Tiempos de respuesta (seg)

Usuarios	Tablón	Chat
1	0,03	0,03
5	0,23	0,23
10	0,53	0,48
20	1,02	1,00
30	1,58	1,50
Pendiente (m)	0,053	0,051



**Figura 10.** Tiempos de respuesta servicios chat y tablón.

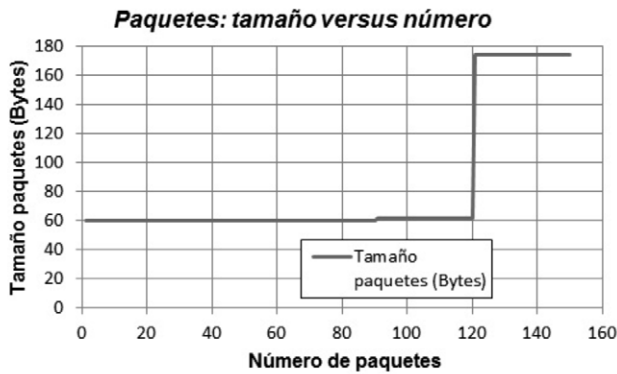
De igual manera, en la figura 11 se presentan los tiempos de respuesta en el lado del servidor o tiempos en servir una petición. Estos valores son obtenidos al aplicar 100 conexiones secuenciales al servicio de tablón, desde la herramienta Apache Benchmark [32]. Según la figura 11, el tiempo en servir una petición es cercano a los 2 milisegundos, salvo las primeras peticiones en las cuales el servidor de aplicaciones se está estabilizando. Lo anterior está de acuerdo con lo presentado en [33], en donde se evalúan los tiempos de respuesta en el procesamiento de un



**Figura 11.** Tiempo en servir peticiones.

mensaje de un servicio REST-JSON JSON estándar con respecto a SOAP, obteniendo valores cercanos a los 30 milisegundos. El hecho de que los tiempos sean inferiores a los obtenidos con un servicio REST-JSON estándar, permite verificar la pertinencia de aplicación del esquema en escenarios limitados como el de TVDi.

Respecto al comportamiento en tamaño de los paquetes de tráfico, se encontró el mismo patrón para los dos servicios y para cualquier número de usuarios (figura 12). De acuerdo a esta gráfica, el 60% de los paquetes tiene un tamaño de 60 Bytes, el 20% de 62 Bytes y el 20% restante de 174 Bytes. De la figura 12 también se observa que el número total de paquetes enviados desde el servidor alcanza un máximo de 180 bytes con 30 usuarios, valor que es pequeño comparado con el tráfico generado por el formato de intercambio XML, en el que se triplica el número de bytes transmitidos para la misma cantidad de información y el mismo número de usuarios [34]. Lo anterior permite corroborar que JSON es un formato adecuado para el intercambio de datos en escenarios limitados como la televisión.



**Figura 12.** Tráfico para el servicio de chat con 30 usuarios simultáneos.

El número de paquetes para los dos servicios se encontró que es exactamente igual (tabla 2). Asimismo, se observa que el número de paquetes se incrementa de una manera lineal con el aumento de los usuarios (figura 13). De acuerdo a la pendiente de la curva estimada para los datos de la tabla 2, el incremento en número de paquetes por cada usuario conectado es de 5. Este valor es menor comparado con el tráfico generado por el formato de intercambio XML, en el que la cantidad de información sería mayor a quince paquetes por usuario [34].

**Tabla II.** Número de paquetes

Usuarios	Número de paquetes Tablón y chat
1	5
5	25
10	50
20	100
30	150
Pendiente (m)	5

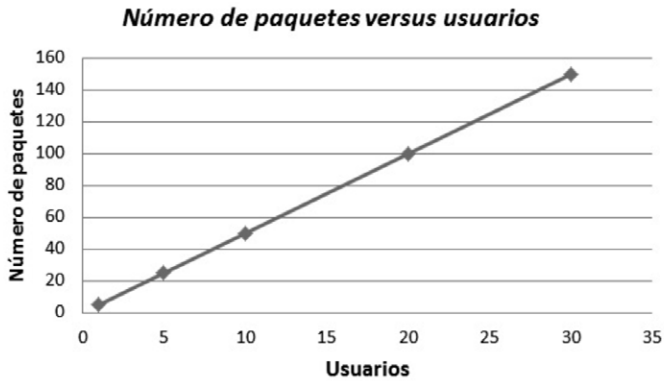


Figura 13. Número de paquetes vs usuarios.

## 5.2. Pruebas de consumo de memoria

Para realizar las pruebas de consumo de memoria en el lado del servidor de aplicaciones, se usó la herramienta de medición de estrés Apache Benchmark (ab) [32], la cual permite simular peticiones HTTP simultáneas y secuenciales. Asimismo, se desarrolló en Python una herramienta para monitorear el consumo de memoria del servidor (figura 14). Esta herramienta genera reportes del consumo de memoria RAM (Random Access Memory) y el porcentaje de consumo de CPU (Central Processing Unit), antes, durante y después de recibir las peticiones simuladas de la herramienta apache benchmark, usando para ello los comandos de consumo de memoria del sistema operativo Linux. En la figura 15 se presenta el porcentaje de uso de %CPU a lo largo del tiempo, al realizar 500 conexiones simultáneas al servicio de tablón de mensajes, mediante la herramienta ab. El servidor de aplicaciones web.py se ejecutó sobre un procesador doble núcleo con 4 Gb de memoria RAM, en el sistema operativo Linux Xubuntu 14.10.

```

archivo1 = open( 'servicios.csv', 'w' )
archivo1.write("SERVICIOS;CPU;MEM\n")
cont1=0
while(True):
    comando = "ps -aux | grep servicios.py | grep -v grep | awk {'print $3,$4'} "
    p = os.popen(comando)
    out = p.read()
    lista = out.split()
    i = 0
    t = len(lista)

    while(t >= 1):
        cad1=lista[i]
        cad2=lista[i+1]
        cad1=cad1.replace(".",",")
        cad2=cad2.replace(".",",")
        fila = str(cont1)+";"+cad1+";"+cad2+"\n"
        print "la lectura es",fila
        archivo1.write(fila)
        cont1 = cont1+1
        i = i+2
        t = t-2

    time.sleep(1)
archivo1.close()

```

Figura 14. Herramienta de medición de memoria.

De acuerdo a la figura 15, cuando el servidor de aplicaciones se encuentra sin recibir peticiones, su porcentaje de consumo de memoria es inferior a 0,5 %. De igual manera, cuando se lanzan las 300 peticiones simultaneas sobre el segundo 5, el porcentaje del consumo de CPU se eleva hasta un 4%. Después del segundo 5, el porcentaje de uso de CPU disminuye, debido a que el servidor de comienza a estabilizarse paulatinamente.

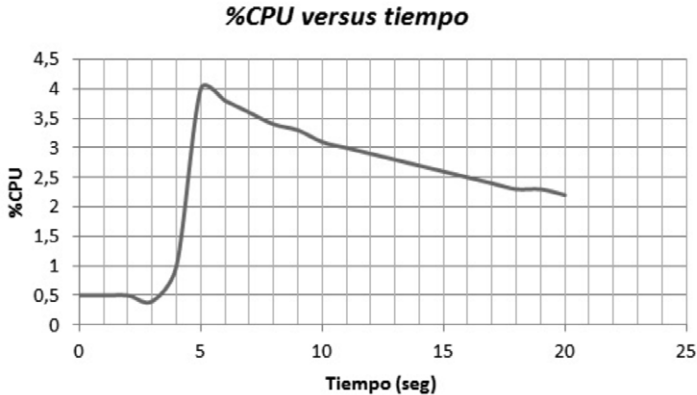


Figura 15. Porcentaje de consumo de CPU.

En lo que respecta a la memoria RAM utilizada por el servidor de aplicaciones, esta se mantiene en un valor invariante de 0.3 Gb a lo largo de todo el tiempo (figura 16).

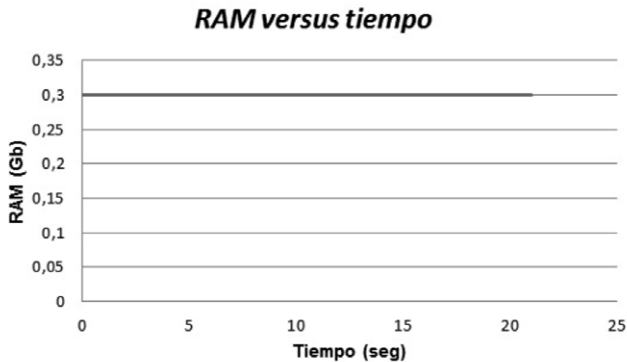


Figura 16. Cantidad de memoria consumida.

## 6. Conclusiones y trabajos futuros

El esquema de diseño y despliegue presentado permite la convergencia de servicios de Televisión (T-Learning) y servicios Internet (Web 2.0), mezclando las ventajas de interactividad en TD con la flexibilidad de los servicios de Internet. Lo anterior permitió que el proyecto



fuera validado en entornos educativos de la Universidad del Cauca, con el propósito de apoyar el desarrollo de prácticas de laboratorio de un curso del Departamento de Química. Como conclusión de lo anterior es importante destacar que, si bien los servicios se desplegaron de manera adecuada en el entorno de televisión, aún son necesarios esfuerzos investigativos para agilizar el acceso a los servicios, considerando las limitaciones de los mandos de entrada de los escenarios de televisión.

El estilo arquitectónico REST-JSON permite un diseño sencillo y flexible para servicios consumidos a través de Internet, lo cual facilita la integración de estos en escenarios de televisión. El diseño e implementación de servicios REST-JSON es independiente de la lógica de presentación en la interfaz del televisor o del dispositivo móvil, lo cual permite extender el escenario de aplicación de los servicios a otros entornos como IPTV.

El esquema de consumo de servicios presentado en este trabajo, así como las directrices de usabilidad presentadas en [13] y derivadas del proyecto ST-CAV, son un aporte importante para diseñadores, desarrolladores y demás actores de la cadena de negocios de TVDi, pues sirven como guía para el diseño, la implementación y el despliegue servicios flexibles y aplicaciones usables en TVDi.

Este trabajo representa un aporte significativo para el proyecto ST-CAV, al proveer un conjunto de servicios de Internet, los cuales buscan promover la participación y generación de conocimiento en comunidades académicas virtuales de televisión. Finalmente, este trabajo representa un punto de partida para proyectos en el que se deseen integrar servicios de Internet, con la flexibilidad que provee el protocolo REST-JSON, en otros ambientes de aplicación de la TVDi, tales como T-Gobierno y T-Comercio.

Asimismo, de acuerdo al análisis de tráfico realizado y presentado en la Tabla 2, se concluye que el consumo de servicios bajo el esquema REST-JSON, tiene un comportamiento lineal con el aumento del número de usuarios, específicamente en cuanto al tamaño de los paquetes, el número de paquetes y los tiempos de respuesta, siendo este el último parámetro el que más puede variar teniendo en cuenta otros servicios o usuarios que soporte la red de manera simultánea.

El análisis de tráfico realizado sobre los servicios construidos, permite evidenciar el bajo consumo de recursos, puesto que, para el caso de 30 usuarios simultáneos del servicio de chat, se genera un tráfico del orden de las decenas de bytes. De igual forma, los otros parámetros como el tamaño de los paquetes, el número de paquetes y el tiempo en servir peticiones, están en los órdenes de los bytes, decenas y *ms* respectivamente. Los anteriores valores no son críticos teniendo en cuenta las capacidades de las redes actuales, las cuales son capaces de manejar transmisiones que están en el orden de los megas en tiempo real. Asimismo, los valores obtenidos permiten corroborar que el formato de intercambio de datos JSON es adecuado en escenarios limitados, en comparación con el formato XML [34].

## 7. Agradecimientos

Este trabajo ha sido realizado en la Universidad del Cauca y ha contado con la financiación parcial del proyecto ST-CAV: Servicios de T-Learning para el soporte de Comunidades Académicas Virtuales, Cod 1103 489 25425, el cual es financiado por Colciencias y el SENA. De igual manera, cuenta con el apoyo de los proyectos de investigación: UsabiliTV (financiado por Colciencias y el MEN. ID 1103 521 28462) y RedAUTI (financiado por CYTED. ID

P511RT0184). Asimismo, este trabajo fue financiado por el programa de Doctorados Nacionales de Colciencias (Convocatoria 528 de 2011). Finalmente, se ha contado con el apoyo de un investigador del grupo de investigación GITUQ de la Universidad del Quindío.

## Referencias

- [1] G. Campos, D. Espinosa, P. Gutiérrez, y F. Martínez, “Televisión Digital en Colombia: Posibilidad para diseñar aplicativos interactivos”. *Revista Tecnología*, Volumen 10, Número 2, 2011, pp. 85-91.
- [2] E. O. Tulande, D. F. Rojas, *Recomendaciones para la generación y distribución de contenidos educativos orientados a Televisión Digital Interactiva*, Tesis de pregrado, Departamento de Telemática, Universidad del Cauca, Popayán, Cauca, Colombia, 2009.
- [3] T. Itagaki, J. Cosmas, y M. Haque, “An interactive digital television system designed for synchronised and scalable multi-media content over DVB and IP networks”, *Multimedia and Expo ICME IEEE International Conference on*, Volumen 3, 2004, pp. 2155-2158.
- [4] G. Prado y S. Zorzo, “Interactive Service Provider Architecture for Interactive Digital Television systems”, *International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, Cracovia, Polonia, 2010, pp. 541-546.
- [5] M. Blando, *Comunidades Académicas Virtuales: Compartir para mejorar*, México, 2003, disponible en: <http://www.te.ipn.mx/comunidadesvirtuales/>.
- [6] L. Ávila, A. Madrid y M. Echeverría, “Construcción de comunidades virtuales para la investigación”, *Revista de Universidad y Sociedad del Conocimiento*, Volumen 6, Número 1, 2009, pp.1-12.
- [7] T. O'Reilly, “What is Web 2.0: design patterns and business models for the next generation of software”, *Revista Communications & Strategies*, Volumen 1, Número 65, 2007, pp. 17-37.
- [8] A. Vela, H. Cerón, *Plataforma móvil para redes sociales*, Tesis de Pregrado, Universidad del Cauca, Departamento de Telemática, Universidad del Cauca, Popayán, Cauca, Colombia, 2009.
- [9] Ye Jun, Li Zhishu, y Ma Yanyan, “JSON Based Decentralized SSO Security Architecture in E-Commerce”, *International Symposium on Electronic Commerce and Security*, Guangzhou, República Popular de China, pp. 471-475.
- [10] S. Downes, L. Belliveau, S. Samet, A. Rahman, y R. Savoie, “Managing digital rights using JSON”, *Consumer Communications and Networking Conference (CCNC) 2010 7th IEEE*, 2010, pp. 1–10.
- [11] S. Cholia, D. Skinner, y J. Boverhof, “NEWT: A RESTful service for building High Performance Computing web applications”, *Gateway Computing Environments Workshop (GCE)*, 2010, pp. 1–11.
- [12] G. Chanchí, W. Campo, J.P. Amaya, J.L. Arciniegas, “Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON”, *Congreso Internacional de Telemática*, Gramado, Brasil, 2011.
- [13] A.F. Solano, G. E. Chanchí, C. Collazos, J.L. Arciniegas, C. Rusu, “Directrices para el diseño de aplicaciones usables en entornos de televisión digital interactiva”. *Revista Ingeniería y Universidad*, Volumen 18, Número 1, 2014, p.p. 103-119.
- [14] STCAV – *Servicios de T-Learning para el soporte de Comunidades Académicas Virtuales*, disponible en <http://www.unicauca.edu.co/stcav/>.
- [15] J. Amaya, F. Urbano, W. Campo, y J. Arciniegas, “Infraestructura Tecnológica para un laboratorio experimental de Televisión Digital Interactiva”, *Congreso Colombiano de Comunicaciones IEEE-Colcom*, 2008.
- [16] W. Ebner, U. Bretschneider, M. Leimeister, y H. Krcmar, “Virtual Communities for Innovations: Users’ Requirements for the Development of an Academic SAP User Group”, *Hawaii International Conference on System Sciences*, 2008.

- [17] Z. Gang, W.G. Lin, Y. Zongkai, L. QingTang, W. Ming, y Li Rong, “Research and Design of Interactive IPTV based E-Learning System”, *2006 7th International Conference on Information Technology Based Higher Education and Training*, 2006, pp. 536-540.
- [18] W. Campo, *Modelo de Tráfico para Servicios Interactivos de una Comunidad Académica Virtual, con contenidos de Audio y Video de Alta Calidad*, Tesis de Doctorado en Ingeniería Telemática, Universidad del Cauca, Popayán, Cauca, 2014.
- [19] G. Chanchí, W. Campo y J. Arciniegas, “Directrices para el soporte de Comunidades Académicas Virtuales en TDi”, *VI Congreso Internacional de Telecomunicaciones – CITTEL*, La Habana-Cuba, 2010.
- [20] R. Navarro, *REST vs Web Services*, Julio de 2006, disponible en: <http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>
- [21] Cessare Pautasso, “REST vs.SOAP:Making the Right Architectural Decision”, *SOA Symposium*, Amsterdam, 2008.
- [22] S. Tyagi, *RestFul Web Services*, Agosto de 2006, disponible en: <http://www.oracle.com/technetwork/articles/javase/index-137171.html>.
- [23] T. Fredrich, *RESTful Service Best Practices*, Mayo de 2012, disponible en: [http://www.restapitutorial.com/media/RESTful\\_Best\\_Practices-v1\\_1.pdf](http://www.restapitutorial.com/media/RESTful_Best_Practices-v1_1.pdf).
- [24] Ecma Intenational, *The JSON Data Interchange Format*, ECMA-404, Octubre de 2013, disponible en: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
- [25] M. Droettboom, *Understanding JSON Schema*, Octubre de 2015, disponible en: <http://spacetelescope.github.io/understanding-json-schema/UnderstandingJSONSchema.pdf>
- [26] Avalpa Digital Engineering, *OpenCaster 3.2.2: the free digital tv software*, disponible en: <http://www.avalpa.com/the-key-values/15-free-software/33-opencaster>
- [27] C. Hartmann, *RestClientlibrary*, Diciembre de 2009, disponible en: <http://www.acidum.de/2008/12/29/j2me-rest-client>
- [28] W. Campo, G. Chanchí y J. Arciniegas, “Recomendaciones para el despliegue de contenidos de T-Learning”, *XI Congreso Internacional–Interacción 2010*, Valencia-España, 2010.
- [29] T. Varaneckas, *Gmail4j*, disponible en: <https://github.com/spajus/gmail4j>
- [30] P. Wach, *s60-json-library*, disponible en: <http://code.google.com/p/s60-json-library/>
- [31] G. Combs, *Analizador de protocolos wireshark*, disponible en: <https://www.wireshark.org/>.
- [32] The Apache Software Foundation, *Apache benchmark*, disponible en: <http://httpd.apache.org/docs/2.2/programs/ab.html>.
- [33] R. Van der Broek, *Comparing the performance of SOAP and REST PHP clients*, Septiembre de 2015, disponible en: <http://referaat.cs.utwente.nl/conference/14/paper/7225/comparing-the-performance-of-soap-and-rest-php-clients.pdf>
- [34] G. Mulligan, D. Graanin, “A Comparison of SOAP and REST Implementations of a service based interaction independence middleware framework”, *Proceedings of the 2009 Winter Simulation Conference*, Austin-Texas, 2009.

---

## Gabriel Elías Chanchí Golondrino

Ingeniero en Electrónica y Telecomunicaciones de la Universidad del Cauca; magíster en Ingeniería Telemática de la Universidad del Cauca; candidato a Doctor en Ingeniería Telemática de la Universidad del Cauca; ha trabajado como docente universitario en cursos de Programación, Estructuras de Datos, Desarrollo de aplicaciones móviles y desarrollo de aplicaciones web. Entre sus campos de interés se encuentran los sistemas de recomendaciones, IPTV y los servicios interactivos en escenarios de televisión digital.

Contacto: [gabrielc@unicauca.edu.co](mailto:gabrielc@unicauca.edu.co)

---

### **José Luis Arciniegas Herrera**

Ingeniero en Electrónica y Telecomunicaciones; especialista en Redes y Servicios Telemáticos de la Universidad del Cauca; doctor en Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid; docente adscrito al Departamento de Telemática de la Universidad del Cauca; entre sus campos de interés están el Desarrollo del Software Evolutivo, Arquitectura del Software, Sistemas de Tiempo Real y la Televisión Digital Interactiva; forma parte del Grupo GIT de la Universidad del Cauca donde ha liderado varios proyectos de televisión Digital interactiva, EDiTV, MT2TDi y ST\_CAV. Contacto: jlarci@unicauca.edu.co

---

### **Wilmar Yesid Campo Muñoz**

Doctor en Ingeniería Telemática; magíster en Ingeniería, Área Telemática; ingeniero en Electrónica y Telecomunicaciones de la Universidad del Cauca; Ha Ejercido profesionalmente como investigador en diferentes proyectos en el ITEC Telecom, en la Universidad del Cauca y en la Universidad del Quindío donde se encuentra actualmente vinculado como docente de planta; forma parte del grupo GITUQ de la Universidad del Quindío; entre sus campos de interés se encuentran: IPTV, sistemas de teletráfico, redes de telecomunicaciones avanzadas y televisión digital Interactiva y la Telemedicina. Contacto: wycampo@uniquindio.edu.co