

Esquema de servicios para Televisión Digital Interactiva, basados en el protocolo REST-JSON

Gabriel E. Chanchí G.

Est. Doctorado Dpto de
Telemática

Universidad del Cauca
Calle 5 No. 4-70, Popayán,
Colombia
57- 3008672608

gabrielc@unicauca.edu.co

Wilmar Y. Campo

Est. Doctorado Dpto de
Telemática

Universidad del Cauca
Calle 5 No. 4-70, Popayán,
Colombia
57- 3157854590

wilcampo@unicauca.edu.co

Juan P. Amaya

Investigador Dpto de
Telemática

Universidad del Cauca
Calle 5 No. 4-70, Popayán,
Colombia
57- 3177729839

jpamaya@unicauca.edu.co

José L. Arciniegas

Prof. Titular Dpto de
Telemática

Universidad del Cauca
Calle 5 No. 4-70, Popayán,
Colombia
57- 3128345864

jlarci@unicauca.edu.co

RESUMEN

Con el propósito de ampliar el abanico de oportunidades para la educación virtual y teniendo en cuenta: las características de interactividad y amplio potencial de penetración de la televisión, así como el auge de servicios de la Web 2.0 en redes sociales y comunidades en Internet, mediante las cuales los usuarios comparten y generan conocimiento alrededor de una temática. Este artículo plantea un esquema para el despliegue de servicios creados bajo el concepto de la Web 2.0, tales como: Foros, Chat, Blogs; en entornos de Televisión digital interactiva – TDi, y presenta como resultado la implementación de 2 servicios: el servicios de tablón y acceso a correo; el esquema y la implementación están basados en el uso protocolo de servicios web REST-JSON para el consumo y diseño de servicios. El presente trabajo está enmarcado dentro del proyecto ST-CAV (Servicios de T-Learning para el soporte de Comunidades Académicas Virtuales - CAV), el cual pretende dar soporte a comunidades académicas virtuales en entornos de Televisión Digital Interactiva TDi.

Palabras Clave

Servicios, Televisión Digital Interactiva (TDi), Comunidades Académicas Virtuales, REST-JSON, Servicios de la Web 2.0

1. INTRODUCCIÓN

Internet ha tenido una marcada evolución en los últimos años, facilitando el intercambio de información de manera flexible y ágil gracias a los múltiples servicios disponibles en red. Estos a su vez hacen parte de la tendencia actual de la red de redes, por fomentar la colaboración y el trabajo en comunidad; ejemplos de estos servicios son: foros, wikis, blogs, chat, entre otros, más conocidos como herramientas de la Web 2.0. Actualmente estos están integrados en plataformas tales como: redes sociales, comunidades académicas virtuales (CAV) [1] o en el ámbito del E-Learning. Debido a las características de integración y flexibilidad y reutilización necesarias para la implementación de estos servicios o herramientas, la tecnología más difundida en este ámbito es la de los Servicios Web [2].

De otra parte la Televisión Digital (TD) está llamada a garantizar un grado de cubrimiento mayor sobre la población, que Internet, lo anterior teniendo en cuenta el apagón analógico que se dará en Latinoamérica en los próximos años y la amplia penetración de la televisión convencional en Colombia (cerca del 99,3%),[3]. Además de lo anterior, gracias al soporte de interactividad que ofrece la Televisión Digital, esta se convierte en una de las tecnologías llamadas a reducir la brecha digital, mediante posibles

entornos interactivos que promuevan el aprendizaje por televisión de forma personalizada, como es el caso del T-Learning [4]. A partir de lo anterior, resulta importante potenciar la interactividad en TD, aprovechando el auge y aceptación de los servicios y tecnologías propias de la Web 2.0. Actualmente dentro del laboratorio experimental de Televisión Digital interactiva de la Universidad del Cauca [5], se viene desarrollando el proyecto ST-CAV (Servicios de T-Learning para el soporte de comunidades académicas virtuales) [6], cuyo objetivo es dar soporte a una CAV en televisión desde diversos escenarios, tales como: Televisión Digital Terrestre basada en el estándar DVB (Digital Video Broadcasting), Televisión Móvil basada en DVB-H e IPTV; de esta forma, uno de los principales objetivos del proyecto ST-CAV es apoyar procesos de aprendizaje en televisión (T-Learning) a través de servicios de la Web 2.0 (Chat, Foros, Blogs, Wikis, entre otros), buscando apoyar y facilitar el aprendizaje y la construcción de conocimiento entorno a diversas temáticas dentro de una CAV.

Teniendo en cuenta lo anterior, es importante considerar las tecnologías empleadas en la implementación de los servicios de Internet. Tal como se menciona antes, la más difundida es la de los servicios Web; los cuales usan comúnmente 2 tipos de protocolos: SOAP (Simple Object Access Protocol) y REST (Representational State Transfer). La diferencia básica entre estos dos protocolos es en cuanto al tipo de mensajes que se intercambian; para el caso de SOAP, los mensajes son en formato WSDL (basado en XML); mientras que en el caso de REST, el tipo de mensaje más difundido es JSON, el cual basa su sintaxis en el lenguaje JavaScript. Los mensajes WSDL al estar basados en el lenguaje XML son complejos comparados con los de tipo JSON, razón por la que muchos de los servicios de las CAV y redes sociales más difundidas (Facebook, Twitter) han sido implementados y tienen interfaces abiertas para desarrollo con el protocolo REST-JSON[7].

El presente artículo plantea un acercamiento entre el mundo de las telecomunicaciones y los servicios educativos de Internet, particularmente la televisión y los servicios educativos de Internet, basados en el protocolo REST-JSON, teniendo en cuenta la difusión de este en redes sociales y CAV; por lo anterior, este trabajo presenta el diseño y esquema de consumo de servicios propios de las CAV, además de un prototipo de servicios integrados al entorno de televisión del proyecto ST-CAV.

Finalmente, la estructura de este artículo es la siguiente: En la sección 2 se presentan los trabajos relacionados, en la sección 3 se muestra el marco teórico y conceptos relacionados, en la sección 4

se presenta el esquema de servicios para TDi basados en servicios web REST-JSON, en la sección 5 se presentan los resultados de la presente investigación y por último en la sección 6 se muestran las conclusiones y trabajos futuros.

2. TRABAJOS RELACIONADOS

A continuación se presentan los trabajos de investigación que han sido tenidos en cuenta como referencia para el presente artículo, los cuales están divididos en las siguientes temáticas: los relacionados con los servicios de televisión, Servicios Web implementados mediante el protocolo REST - JSON, CAV y mashups en televisión. En estos trabajos se constata las diferentes tendencias y el interés de la comunidad científica en estas temáticas.

En [8] los autores muestran como diferentes investigaciones predicen el rápido desarrollo de la TDi en Europa, exponen como en el Reino Unido los usuarios demandan un mayor valor agregado sobre la televisión convencional, puesto que existen expectativas a cerca de las mejoras reales que ofrece la TDi gracias a sus ventajas técnicas. 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 [9] se presenta una arquitectura para proveer servicios interactivos para un sistema de TDi, basada en una arquitectura orientada a servicios, 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.

Respecto a los protocolos JSON y REST, en [10] se muestra como mediante los Servicios Web basados en SSO (Single Sign On – Inicio de Sesión Único) 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 (JavaScript Object Notation) definido por los autores como un formato ligero para el intercambio de datos de fácil lectura y escritura para los seres humanos. Además en este trabajo sustituyen el XML por JSON debido a su fácil comprensión para los seres humanos y su mayor velocidad de procesamiento para los equipos de cómputo. En [11] se demuestra como usando XML se requiere un proceso de interpretación o análisis, como es característico en el procesamiento del lenguaje, 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 [12] describen la construcción de aplicaciones web de alto desempeño computacional (High Performance Computing - HPC) usando REST Web Service. Donde, para lograr una completa funcionalidad de las aplicaciones web se siguen los estándares de la Web 2.0 y los protocolos JSON y REST.

Respecto a la tendencias de las CAV existen diversas investigaciones que las abordan desde el punto de vista pedagógico, social y técnico, así por ejemplo en [13] se investiga las necesidades de los usuarios para el desarrollo de una comunidad virtual (CV) para el curso de SAP UCC (University Competence Center) del Grupo de Usuarios. El Grupo de Usuarios está compuesto por profesores y estudiantes para propósitos educativos. Se trabajó con 1.300 profesores para esta comunidad. 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.

En [14] 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. Los autores han propuesto las metodologías de diseño para el aprendizaje interactivo, que es la característica clave de la demanda basada en IPTV. Para evaluar el marco y la metodología propuestos, se desarrollo un prototipo de un servicio de IPTV basado en el aprendizaje.

Respecto a los mashups, en [15] se muestra como IBM ha detectado la proliferación y el rápido avance que ha tenido el uso de las mashups en la construcción de aplicaciones Web, así mismo detectan la falta de estandarización y compatibilidad. Para hacer frente a esto, la plataforma de IBM usa un lenguaje de dominio específico innovador que proporciona una estructura común y el diseño de aplicaciones web híbridadas, así como instalaciones para ayudar a compartir y reutilizar componentes mashup.

En [16] los autores consideran una clase especial de servicios mashups, denominados por los autores como “manejador de eventos de servicios mashups”. Dicha clase especial es utilizada para la construcción de servicios, tanto para el mundo de las telecomunicaciones, como para el mundo de Internet, todo bajo un framework denominado orquestador de mashups. En [17] se exploran cómo una arquitectura orientada a servicios web (WSOA) puede definirse como un flujo de datos en un enfoque mashup, donde las técnicas de ingeniería de manejo de modelos permiten una composición inteligente de flujos de datos y la generación de flujos de control basado en la arquitectura. Observamos como los mashups pueden ser aprovechados en diferentes contextos.

Este artículo recoge las diferentes iniciativas de investigación y reúne todos estos conceptos con el objeto de generar un Esquema de servicios para Televisión Digital Interactiva.

3. MARCO TEÓRICO

A continuación se presentan conceptos relevantes, que se han tenido en cuenta para el desarrollo de la presente investigación, estos son: CAV, Servicios Web, REST, JSON.

3.1 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” [1]; para el caso de una CAV en entornos de TDi, se plantea que el proceso de construcción de conocimiento sea impulsado por los contenidos multimedia académicos, aportados por la miembros de la comunidad y por el conjunto de aplicaciones o servicios que buscan promover la participación en torno a esos contenidos. En [18] se plantean un conjunto de directrices para el proyecto ST-CAV, en las cuales se propone el modelo de estas comunidades en televisión, desde el punto de vista de funcionalidad, despliegue e implementación.

3.2 Servicios Web

La W3C define un Servicio Web como un sistema software diseñado para soportar una interacción interoperable entre

diferentes equipos en red [19]. Estos suelen ser API's Web 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. [Referencia Servicios]. En ambos casos tanto el cliente como el servidor deben conocer el formato y lenguaje de los mensajes para poder encapsular y des-encapsular peticiones y respuestas.

3.3 REST

El protocolo REST (Representational State Transfer), plantea un estilo de arquitectura cliente-servidor en la cual un servicio es visto como un recurso es identificado a través de una dirección URL, mediante la cual puede ser accedido o consumido. Para acceder a estos servicios web, se hace uso de mensajes en formato simple, los cuales se intercambian entre cliente y servidor [20]. REST define a partir de HTTP, cuatro métodos: GET, PUT, DELETE y POST, de los cuales los más usados 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 [21]. Para la comunicación e intercambio de información entre cliente y servidor a través de REST, se puede hacer uso de diversos formatos y lenguajes: XML, HTML, JSON; debido a la sencillez de los mensajes JSON, este es el tipo de mensajes mas difundido en diferentes servicios propios de redes sociales (Facebook y Twitter) y comunidades en Internet. Cada mensaje intercambiado contiene la información necesaria para cada el funcionamiento de los servicio, de tal forma que para cada servicio el cliente y servidor conocen el formato o protocolo interno de los mensajes.

3.4 JSON

JSON es un formato ligero basado en de texto, con definición de lenguaje completamente independiente, cuya sintaxis está basada en un subconjunto de funcionalidades y palabras reservadas del lenguaje de script Web: JavaScript. Debido a su sencillez es fácil generar y procesar un documento con este formato [22]; JSON usa las convenciones para el manejo de datos, que son comunes a la familia de lenguajes: C, C++, C#, Java, Perl, Python, etc; esta característica hace de JSON un lenguaje ideal para el intercambio de datos entre aplicaciones cliente servidor o aplicaciones que se ejecuten en diferentes escenarios. [8]. Un mensaje JSON está constituido por dos estructuras básicas [23]: la primera es una colección de parejas o estructuras nombre-valor, las cuales en varios lenguajes son conocidas como: registros, diccionarios, tablas, hash, listas de claves o arreglos asociativos; y la segunda es una lista ordenada de valores; en la mayoría de los lenguajes, esta se suele representar en forma de: arreglos, vectores, listas, etc. Las anteriores estructuras son usadas para conformar los mensajes, en los cuales se define un protocolo interno con la representación: palabra clave - valor o un conjunto de valores ordenado con significado; estos mensajes son interpretados por el cliente y servidor y decodificados de acuerdo al protocolo interno definido, para lo cual se requiere el uso de librerías que permitan interpretar la sintaxis propia del lenguaje JSON. Para el caso del presente artículo se hizo uso del protocolo para consumir servicios desde un dispositivo móvil y un STB (Set Top Box), en cada caso se usó diferentes librerías.

El presente trabajo hace uso de los 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 en torno a la generación de conocimiento en una CAV. Estos servicios han sido implementados usando el protocolo REST y el lenguaje JSON como formato para el intercambio de mensajes.

4. ESQUEMA DE SERVICIOS PARA TELEVISIÓN DIGITAL INTERACTIVA

Dentro del proyecto ST-CAV, se escogieron un conjunto de servicios, tomados de la WEB 2.0 para dar soporte a las CAV en televisión, estos son: foros, tablón de mensajes o micro-blog, chat, acceso a correo electrónico, notificaciones. Estos servicios han sido implementados haciendo uso del protocolo de servicios Web REST – JSON. A continuación se presenta las condiciones de diseño de estos servicios y el escenario de experimentación usado para los mismos.

Teniendo en cuenta que los mensajes intercambiados a través del protocolo REST, deben contener la información necesaria para el funcionamiento de cada servicio en el escenario de televisión; es necesario definir dentro del formato de los mensajes JSON, el conjunto de parejas: nombre-valor relacionadas con la funcionalidad de los servicios, Ej: En el ámbito de las CAV, "nombre" puede hacer referencia a los atributos: login, mensaje, hora, etc, mientras que "valor" se refiere a una cadena de texto con la información de ese atributo, Ver Figura 1.

En TDi, los servicios son consumidos mediante un canal de retorno o canal bi-direccional, a través del protocolo IP e integrados por el STB o dispositivo móvil, en la interfaz del televisor o en la pantalla del dispositivo respectivamente. Los mensajes a intercambiar entre los clientes de televisión y el servidor de aplicaciones contienen características funcionales y no de presentación, razón por la cual cada cliente: dispositivo móvil o STB, recupera la información a partir de los mensajes e implementa la lógica de presentación de manera independiente.

En la Figura 2, se presenta el esquema de despliegue de servicios para el protocolo 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, están en la capacidad de interactuar entre si para procesos de composición. A cada recurso o servicio se le asigna una URL desde la cual, el Cliente: 1,2,...,N, puede acceder via Internet y recibir el mensaje correspondiente a cada recurso; para el caso del escenario de televisión el acceso a internet es a través del canal de retorno. 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, Ver Figura 1, las cuales son des-encapsuladas por cada cliente, y de acuerdo al nombre del atributo, son clasificadas funcionalmente y presentadas en la interfaz correspondiente; cada tipo de cliente debe contar con la librería apropiada según las características de su Hardware, para permitir la interpretación de los mensajes JSON; de igual forma el cliente deben contar con las capacidades necesarias para hacer peticiones vía HTTP de tipo GET o PUT propias del protocolo REST . El presente artículo plantea un escenario con 2 tipos de clientes: Un cliente de televisión digital terrestre: STB y un cliente de televisión móvil: dispositivo móvil, cada uno de ellos cuenta con la librería apropiada para sus características Hardware.

Status: 200 (OK)

Response:

| Tabular View | Raw View | Sub-Resource | Headers | Http Monitor |
|--|----------|--------------|---------|--------------|
| <pre>{ "tablon": [{ "estampa": "1303961583078", "mensaje": "Este es un mensaje de Prueba", "userid": "andres" }, { "estampa": "1303961547109", "mensaje": "Este es un mensaje de Prueba", "userid": "mario" }, { "estampa": "Prueba", "userid": "gabriel" }] }</pre> | | | | |

Figura 1. Mensajes en Formato JSON

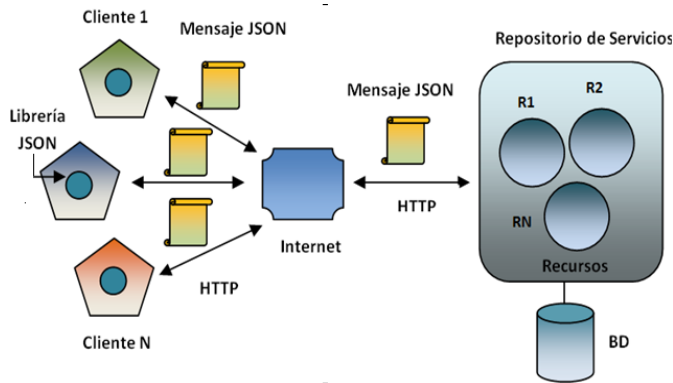


Figura 2. Esquema de Despliegue Servicios REST-JSON

4.1 Escenario de Experimentación

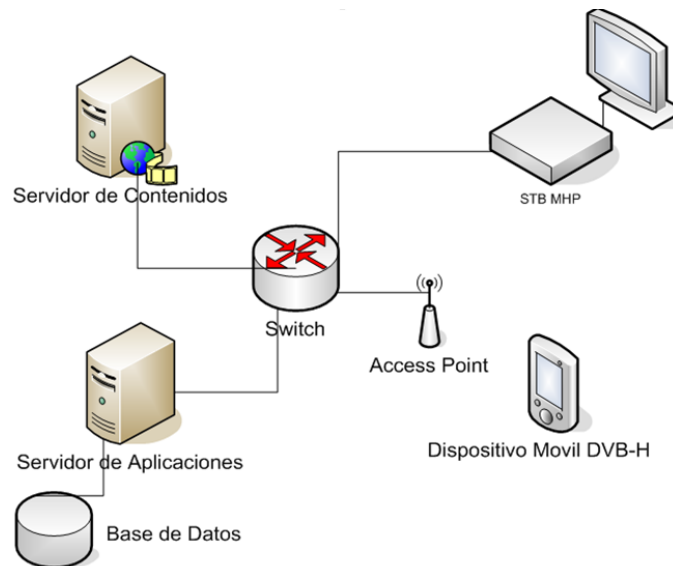


Figura 3. Escenario de Experimentación.

El Escenario de Experimentación está formado por los siguientes componentes: Servidor de Contenidos, Servidor de Aplicaciones, STB TDT – MHP, Dispositivo Móvil DVB-H WLAN, Access Point y Switch, los cuales serán descritos a continuación.

El Servidor de Difusión o Servidor de Broadcast es el encargado de albergar, adecuar y transmitir el contenido multimedia vía Broadcast; para ello cuenta con la Herramienta libre Open Caster y la Tarjeta Moduladora de Televisión: Dektec DTA 110T. El servidor de Aplicaciones, corresponde al equipo de computo encargado de almacenar los servicios, recibir y procesar las peticiones REST por parte de los clientes; en este escenario está formado por el servidor de aplicaciones GlashFish versión 3, el cual cuenta con el soporte para la creación y despliegue de Servicios Web basados en REST – JSON, esta herramienta hace parte del entorno de desarrollo para J2SE, J2EE y J2ME Netbeans 6.9.1 usada para el desarrollo de los servicios. El Switch y El Access Point son los encargados de distribuir Internet de forma cableada e inalámbrica al STB y al dispositivo móvil respectivamente, de esta forma se realiza la conexión por canal de retorno para los dos clientes. El STB MHP TDT, 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 o servicios del Servidor de Aplicaciones a través de la URL designada por cada recurso, para ello hace uso de las librerías de conexión que provee el middleware MHP; cada vez que accede a un recurso, recibe un mensaje en formato JSON que interpreta mediante las librerías compatibles con MHP que permitan la decodificación del mensaje; a partir del mensaje recibido obtiene la información y la presenta en pantalla según corresponda. El dispositivo móvil es el encargado de recibir la señal de televisión DVB-H, la cual es adaptada en el Servidor de Broadcast, así como acceder a los servicios de la CAV mediante la URL del recurso; cada vez que el móvil accede a un recurso, recibe un mensaje en formato JSON, el cual obtiene la información del servicio, la procesa y presenta en la interfaz del dispositivo. 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 y E61i.

5. RESULTADOS

En esta sección se presenta los resultados de la implementación de 2 de los servicios para el soporte de CAV del proyecto ST-CAV, los cuales son: micro-blog o tabón de mensajes y acceso a cuenta de correo Gmail.

En la Figura 4, se muestra la interfaz principal de televisión del proyecto ST-CAV, dentro de esta, se pueden visualizar a la izquierda el contenido multimedia transmitido desde el Servidor de Broadcast 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 [24]. 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. 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 contenidos de televisión [24], según las cuales se sugiere el uso de la fuente Tiresias 16; 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: encuestas e información asociada al contenido; 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 DVB. A continuación se describen los servicios de micro-blog y acceso al correo.

5.1 Servicio de Tablón o Micro-Blog



Figure 4. 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 para 2011", "userid": "lucas" }, { "estampa": "1303964498953", "mensaje": "Esta es una prueba corta para: 2011", "userid": "eduardo" }, { "estampa": "1303964269687", "mensaje": "Esta es una prueba corta", "estampa": "1303964248640", "mensaje": "Esta es una prueba corta", "userid": "marcos" }] }</pre> | | | | |

Figura 5. Interfaz de Prueba del Servidor - Tablón

El servicio de tablón de mensajes o micro-blog, ver Figura 4, 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 mientras están viendo un programa de televisión, considerando que el ingreso de los mensajes en el tablón se hace a través de un control remoto. La interfaz del servicio de tablón de mensajes permite listar los últimos 5 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 ver Figura 5, 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 5 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.

5.2 Servicio de Acceso a Correo

El servicio de acceso a correo electrónico de Gmail, ver Figura 6, 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 5 mensajes de correo del miembro de la comunidad registrado y navegar a través de ellos mediante las flechas del control. 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, ver Figura 7; el tamaño del arreglo de mensajes de correo es de 5, 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 del API gmail4j [25], el 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 5.



Figura 6. Servicio de Acceso a Correo

Status: 200 (OK)

Response:

| Tabular View | Raw View | Sub-Resource | Headers | Http Monitor |
|--|----------|--------------|---------|--------------|
| <pre>{ "mailmessage": [{ "from": "Gabo", "preview": "Hola Este es un Mensaje de Prueba Saludos -- Gal 3" }, { "from": "Gabo", "preview": "Hola Este es un Mensaje de Prueba Saludos -- Gabriel Elias Ch" }, { "from": "Claudia Patricia Ac.", "preview": "Hoy es martes, el entusiasmo aumenta en un 20% --" }, { "from": "Claudia Patricia Acevedo Ingeniera de Proyectos Seratic Ltda", "preview": "Hola que tal -- Claudia Patricia Acevedo Ingeniera de Proyectos Seratic Ltda", "subject": "Bienvenido al equipo de laboratorio de televisión digital interactiva ...", "sendDate": "Mon Oct 11 17:48:41 COT 2010", "body": "hace mucho tiempo que no necesitas tu ordenador para acceder ..." }] }</pre> | | | | |

Figura 7. Interfaz de Prueba del Servidor - Correo

5.3 Servicios desde el Móvil o vía Web

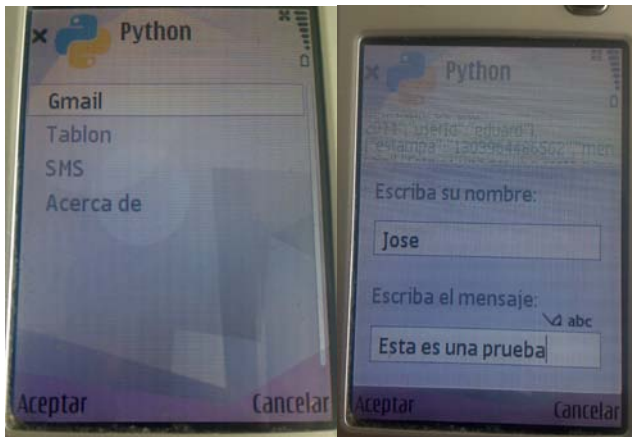


Figure 8. Servicio de Micro-Blog o Tablón

Desde el dispositivo móvil se desarrollaron un conjunto de scripts Python con interfaz gráfica simple, sobre el sistema operativo Symbian V9.2, ver Figura 8; los cuales permiten el acceso a los servicios de la CAV vía red inalámbrica desde dispositivo con soporte WLAN (N95, N96); estos scripts hacen la invocación a la URL de recursos o servicios y decodifican el mensaje JSON haciendo uso de la librería s60-json-library de Python [26], gracias a la cual es posible instalar el interprete de Python en el sistema operativo Symbian. Cada vez que se hace una invocación desde el dispositivo móvil, o se accede a un recurso o servicio remoto, se recibe un mensaje en formato JSON el cual es

interpretado usando una implementación de json para PYS60 y presentado en la pantalla del dispositivo, ver Figura8. Los servicios de tablón de mensajes y correo electrónico también pueden ser consultados a través de una implementación web, ver Figura 9, la cual hace uso de la tecnología AJAX (Asynchronous Javascript and XML) para solicitar los mensajes JSON de los servicios; los mensajes son procesados usando Javascript y presentados mediante HTML en la interfaz Web. Esta implementación implica un avance para futuros trabajos en el campo de IPTV, escenario en el cual es necesario hacer uso de tecnologías afines para el consumo de los servicios REST-JSON.

Mensajes Tablon ST-CAV!

| Mensaje | Usuario | TimeStamp |
|------------------------------|---------|---------------|
| Este es un mensaje de Prueba | andres | 1303961583078 |
| Este es un mensaje de Prueba | jose | 1303961561125 |
| Este es un mensaje de Prueba | mario | 1303961547109 |
| Este es un mensaje de Prueba | luis | 1303961475953 |
| Mensaje de Prueba | gabriel | 1303961454812 |

Lector de Correos ST-CAV!

| From | Fecha | Asunto | Preview |
|----------------------|------------------------------|------------------------------------|--|
| Gabo | Thu Apr 28 11:14:38 COT 2011 | MENSAJE DE PRUEBA 3 | Hola Este es un Mensaje de Prueba Chanchi G Ing. Electronica y Tel ... |
| Gabo | Thu Apr 28 11:14:11 COT 2011 | Fwd: MENSAJE DE PRUEBA2 | Hola Este es un Mensaje de Prueba Chanchi G Ing. Electronica y Tel ... |
| Claudia Patricia Ac. | Tue Oct 12 16:04:12 COT 2010 | Martes !!! | Hoy es martes, el entusiasmo aume Patricia Acevedo |
| Claudia Patricia Ac. | Mon Oct 11 17:49:31 COT 2010 | PRUEBA | Hola que tal -- Claudia Patricia Ac Proyectos Seratic Ltda |
| Claudia Patricia Ac. | Mon Oct 11 17:48:41 COT 2010 | Saludos Varios | Hola equipo ST-CAV Bienvenido al laboratorio de televisión digital interactiva ... |
| El equipo de Gmail | Mon Oct 11 16:41:11 COT 2010 | Obtener Gmail en tu teléfono móvil | Access Gmail on your mobile phone que no necesitas tu ordenador para |

Figure 9. Interfaz Web Servicios –Tablón - Email

5.4 Condiciones Técnicas del Servicio

Los servicios de de micro-blog o tablón de mensajes y el servicio de acceso al correo del proyecto ST-CAV fueron implementado usando el protocolo para servicios web REST, mediante el intercambio de mensajes JSON; a través de este protocolo, y mediante mensajes de tipo JSON, el servidor de aplicaciones intercambia datos propios de los servicios con el STB (Set Top Box) y el dispositivo móvil. Para el desarrollo de estos servicios en el lado del servidor, se hizo uso del entorno de desarrollo Netbeans 6.9.1 [27] el cual cuenta con el soporte para la creación y despliegue de servicios web REST-JSON en Java y con el servidor de aplicaciones GlashFish V3; en el caso del cliente de televisión se hizo uso de: el simulador de televisión: Mi MundoTV; las librerías de desarrollo de MHP vinculadas al entorno de desarrollo Eclipse; la librería RestClient [28], desarrollada para escenarios J2ME y usada para las peticiones y conexiones REST desde el STB MHP; y la librería json-simple [29], usada para la codificación y decodificación de los mensajes en formato JSON.

Desde el dispositivo móvil, teniendo en cuenta que el sistema operativo Symbian, cuenta con librerías para el manejo de DVB-H, se ha optó por el uso de este como middleware para acceder a los servicios; dentro de Symbian se hizo uso de la librería PYS60, la cual es una implementación para del intérprete del lenguaje Python para Symbian; así se definieron un conjunto de scripts Python para acceder a los servicios, mediante la librería de conexión URLLIB y la librería de interpretación de mensajes

JSON para python s60-json-library. De los dispositivos móviles usados para las pruebas de consumo de servicios, el N96 soporta el servicio de televisión, las características de conexión necesarias y el sistema operativo Symbian para consumir los servicios usando las librerías mencionadas; por su parte los Nokia N95 y E61i solo cuentan con el sistema operativo Symbian y el soporte para conexión inalámbrica.

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.

El protocolo 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.

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 TDi, tales como: T-Gobierno y T-Comercio.

Como trabajo futuro, dentro del Laboratorio Experimental de televisión digital interactiva de la Universidad del Cauca, se pretende vincular servicios de personalización y recomendación de contenidos multimedia al escenario de las CAV en TDi, los cuales puedan ser accedidos o consumidos a través del protocolo REST-JSON.

7. AGRADECIMIENTOS

Este trabajo ha sido realizado en la Universidad del Cauca y es financiado parcialmente por el 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.

8. REFERENCES

[1] Blando, M., 2003. Comunidades Académicas Virtuales: Compartir para mejorar, (México, 2003).

[2] O'Reilly, T: "What is web 2.0: design patterns and business models for the next generation of software", DOI: <http://www.oreillynet.com/go/web2>, 2005.

[3] "Radiografía de la Televisión en Colombia", www.tvyvideo.com, 2007

[4] Tulande, E. O., Rojas, D. F.: "Recomendaciones para la generación y distribución de contenidos educativos orientados a

Televisión Digital Interactiva", Tesis Universidad del Cauca, 2009.

[5] Amaya, J., Urbano, F., Campo, W., Arciniegas, J.: "Infraestructura Tecnológica para un laboratorio experimental de Televisión Digital Interactiva", Congreso Colombiano de Comunicaciones IEEE, Colcom 2008.

[6] STCAV – Servicios de T-Learning para el soporte de Comunidades Académicas Virtuales, <http://www.unicauca.edu.co/stcav/>

[7] Vela, A., Cerón, H. "Plataforma móvil para redes sociales", Tesis Universidad del Cauca, 2009.

[8] T. Itagaki, J. Cosmas, y M. Haque, "An interactive digital television system designed for synchronised and scalable multimedia content over DVB and IP networks," Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on, 2004, págs. 2155-2158 Vol.3.

[9] G. Prado y S. Zorzo, "Interactive Service Provider Architecture for Interactive Digital Television systems," Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on, 2010, págs. 541-546.

[10] Ye Jun, Li Zhishu, y Ma Yanyan, "JSON Based Decentralized SSO Security Architecture in E-Commerce," Electronic Commerce and Security, 2008 International Symposium on, 2008, págs. 471-475.

[11] 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, págs. 1-10.

[12] S. Cholia, D. Skinner, y J. Boverhof, "NEWT: A RESTful service for building High Performance Computing web applications," Gateway Computing Environments Workshop (GCE), 2010, págs. 1-11.

[13] W. Ebner, U. Bretschneider, M. Leimeister, y H. Krcmar, "Virtual Communities for Innovations: Users? Requirements for the Development of an Academic SAP User Group,"

[14] 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, Ultimo, Australia: 2006, págs. 536-540.

[15] E.M. Maximilien, A. Ranabahu, y K. Gomadam, "An online platform for web apis and service mashups," Internet Computing, IEEE, vol. 12, 2008, págs. 32-43.

[16] M. Stecca, M. Maresca, y P. Baglietto, "Scalable Orchestration of Telco/IT Mashups," Intelligence in Next Generation Networks, 2009. ICIN 2009. 13th International Conference on, 2009, págs. 1-6.

[17] S. Mosser, F. Chauvel, M. Blay-Fornarino, y M. Riveill, "Web Services Composition: Mashups Driven Orchestration Definition," 2008 International Conference on Computational Intelligence for Modelling Control & Automation, Vienna, Austria: 2008, págs. 284-289.

[18] Chanchí, G., Campo, W., Arciniegas, J. 2010. Directrices para el soporte de Comunidades Académicas Virtuales en TDi, VI Congreso Internacional de Telecomunicaciones – CITTEL (La Habana, Cuba, Diciembre de 2010).

- [19] Guía breve de Servicios Web, <http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>
- [20] RestFul Web Services, <http://www.oracle.com/technetwork/articles/javase/index-137171.html>
- [21] Minimum Methods REST, <http://rest.blueoxen.net/cgi-bin/wiki.pl?MinimumMethods>
- [22] JSON, O'Reilly, T: "What is web 2.0: design patterns and business models for the next generation of software", Disponible: <http://www.oreillynet.com/go/web2>, 2005.
- [23] Introducción a JSON, <http://json.org/json-es.html>
- [24] Campo, W., Chanchí, G., Arciniegas, J. 2010. Recomendaciones para el despliegue de contenidos de T-Learning, XI Congreso Internacional - Interacción 2010 (Valencia, España, Septiembre de 2010). DOI=<http://dx.doi.org/10.4067/S0718-07642010000600005>
- [25] Gmail4j, <http://code.google.com/p/gmail4j/>
- [26] s60-json-library, <http://code.google.com/p/s60-json-library/>
- [27] Netbeans IDE 6.9.1, <http://netbeans.org/community/releases/69/>
- [28] RestClient library, <http://www.acidum.de/2008/12/29/j2me-rest-client>
- [29] json-simple library, <http://code.google.com/p/json-simple>