

MASHUP DE CALENDARIOS Y LOCALIZACIÓN GEOGRÁFICA DE REDES SOCIALES*

Por: Jorge Hernández Mercado**

Fecha de recibido: 6 de julio de 2010 • Fecha de aceptación: 30 de septiembre de 2010

RESUMEN:

Las redes sociales son unos pilares de la web 2.0, que avanzan a pasos agigantados. Aunque estas redes generalmente interactúan virtualmente, eventualmente se reúnen en un lugar geográfico para llevar a cabo alguna actividad de importancia presencial. Estas actividades son programadas en un calendario donde se asignan fechas y direcciones físicas. Este artículo presenta el desarrollo de GCel, una aplicación web híbrida (Mashup) creada para la generación automática de calendarios y localizaciones geográficas de redes sociales usando Google Calendar y Google Map respectivamente. La arquitectura utilizada por GCel es una combinación de aplicaciones jMaki y Jersey. La aplicación Jersey usa servicios web RESTful para acceder a la base de datos de las redes sociales, creada en MySQL. La aplicación jMaki utiliza los widgets: Google Maps, Dojo Table y Yahoo Button. Este último widget toma la información dada por el servicio web RESTful y gracias al mecanismo publish/suscribe de jMaki lo publica a los dos widgets suscritos. Se muestra el uso de GCel para la creación de calendarios y localización geográfica de cada una de las redes sociales de la Fundación Consentir.

PALABRAS CLAVE:

Mashup, jMaki, Jersey, Widgets, RESTful, Red social.



* Artículo de avance de la investigación Construcción Mashup SOA para la Gestión de Redes Sociales como Apoyo a la Formación por Competencias: SPORS. El ing. Jorge Hernández es el investigador principal y miembro del grupo de investigación Ingeniería de Software y Redes de la Corporación Universitaria de la Costa, CUC registrado en Colciencias en la categoría B.

** Ingeniero de Sistemas de la Universidad del Norte. Especialista en Ingeniería del Software de la Universidad del Norte. Especialista en Alta Gerencia de la Universidad Autónoma del Caribe. Magíster en Informática industrial y Automática de la Universidad de Girona, España. Profesor tiempo completo en el programa de Ingeniería de Sistemas de la Corporación Universitaria de la Costa. jhernandez@cuc.edu.co



MASHUP FOR GEOGRAPHICAL LOCATION OF SOCIAL NETWORKS

By: Jorge Hernández Mercado

ABSTRACT:

The social networks(nets) are advancing in Internet by leaps and bounds. Though these networks(nets) generally interact virtually, eventually they meet in a geographical place to carry out some activity of importance presence. This article presents the development of GCel, Mashup for the geographical location of social networks(nets) by means of Google Maps. The architecture used by GCel is a combination of applications jMaki and Jersey. The application Jersey uses services web RESTful to accede to a database

in MySQL. The application jMaki uses the widgets: Google Maps, Dojo Table and Yahoo Button. The latter widget takes the information given by the service web RESTful and thanks to the mechanism publish/subscribe of jMaki it publishes to the two widgets signed. We show GCel's use for the geographical location of her(it) sedate principally of each of the social networks(nets), called cells, of the Foundation To consent.

KEY WORDS:

Mashup, jMaki, Jersey, Widgets, RESTful, Social Networks.



I. INTRODUCCIÓN

Una red social¹ es una estructura social que se puede representar en forma de uno o varios grafos en el cual los nodos representan individuos y las aristas relaciones entre ellos.

Mashup² es una aplicación web que usa contenido de otras aplicaciones web para crear un nuevo contenido completo, consumiendo servicios directamente, siempre a través de protocolo http. Normalmente realizan este proceso por medio de una interfaz pública o usando un API. Mashup³ es uno de los pilares de la famosa web 2.0, la web social y colaborativa donde los usuarios tienen algo que aportar.

Existen herramientas que facilitan la construcción de este tipo de aplicaciones, entre ellas está jMaki⁴, un framework que provee un modelo liviano para crear aplicaciones web habilitadas para AJAX. Provee un wrapper para widgets de múltiples toolkits tales como Yahoo, Dojo, Google y muchos otros. Los widgets asociados a jMaki pueden ser fácilmente usados en una aplicación JSP, PHP y Phobos.

Por otro lado se tienen herramientas para la creación y consumo de servicios web, por ejemplo: Jersey⁵, un framework que permite crear servicio web RESTful con Java.

Google Calendar y Google Maps son aplicaciones que permiten gestionar calendarios de actividades y localización geográfica respectivamente. Las aplicaciones de Google presentan la posibilidad de enviar o conseguir datos por medio de una API llamada Google API.

Entonces, ¿cómo se puede utilizar la información de redes sociales almacenada en una base de datos MySQL e integrarlas con Google Ca-

lendar y Google Maps usando los frameworks jMaki & Jersey?

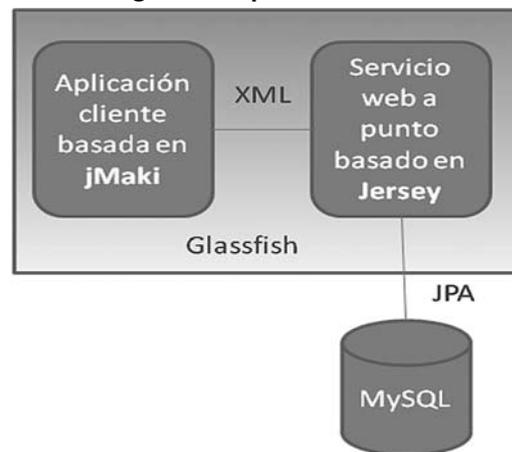
En este artículo se presenta a GCell, Mashup de calendarios y localización geográfica de redes sociales por medio de la API de Google utilizando los frameworks jMaki y Jersey.

El artículo está estructurado de la siguiente manera, la sección 2 describe la arquitectura utilizada por GCell. La sección 3 empieza con una descripción del framework Jersey y luego ilustra los pasos para crear un servicio web RESTful. La sección 4 presenta el framework jMaki, su modelo de comunicación y los pasos para crear una aplicación web que consume el servicio web creado con Jersey. La sección 5 presenta el caso para las redes sociales de la Fundación Consentir. Finalmente se presentan las conclusiones y trabajos futuros.

2. ARQUITECTURA UTILIZADA POR GCEL

GCell es un sistema Cliente/Servidor que tiene aplicaciones, del lado del servidor y del lado del cliente. La figura 1 muestra esta arquitectura⁶.

Figura 1. Arquitectura GCell



1. Wikipedia. Red social. http://es.wikipedia.org/wiki/Red_social. 2009
 2. Wikipedia. Mashup (aplicación web híbrida). [http://es.wikipedia.org/wiki/Mashup_\(aplicaci%C3%B3n_web_h%C3%ADbrida\)](http://es.wikipedia.org/wiki/Mashup_(aplicaci%C3%B3n_web_h%C3%ADbrida)). 2009
 3. Techtear.com. Los Mashups, uno de los pilares de la web 2.0. <http://www.techtear.com/2007/03/26/los-mashups-uno-de-los-pilares-de-la-web-20/>. Consultado junio 21 de 2009.
 4. Project jMaki. <https://ajax.dev.java.net/>. 2007
 5. Glassfish – Jersey. <https://jersey.dev.java.net/>. Consultado junio 21 de 2009.
 6. Gupta Arun. Travel Map-Another Real-life app using jMaki & Jersey. http://blogs.sun.com/arungupta/entry/screencast_web11_travel_map_another



Los datos relacionados a las redes sociales residen en una base de datos en MySQL. Se maneja información tales como: nombre de la red, localización geográfica, fecha de creación, líder y los miembros de la red. Para cada persona de la red se guarda: el nombre, fecha nacimiento, localización geográfica, correo electrónico, entre otras. Esta información es tomada por un servicio web RESTful por medio de JPA-Java Persistence API⁷. Se utilizó el framework Jersey para crear un servicio web RESTful construyendo una aplicación web que sube al servidor web, el servicio.

También se creó una aplicación web⁸, construida con el framework jMaki, para consumir los datos entregados por el servicio web utilizando los widgets: Yahoo Button, Dojo Table y Google Maps.

Las dos aplicaciones web fueron creadas como proyectos en Netbeans 6.5.1 desplegadas en el servidor web Glassfish V3, ejecutando primero la aplicación Jersey para subir el servicio web y luego la aplicación jMaki para consumir y visualizar el servicio.

3. APLICACIÓN JERSEY: SERVICIO WEB RESTFUL CONECTADO A UNA BASE DE DATOS

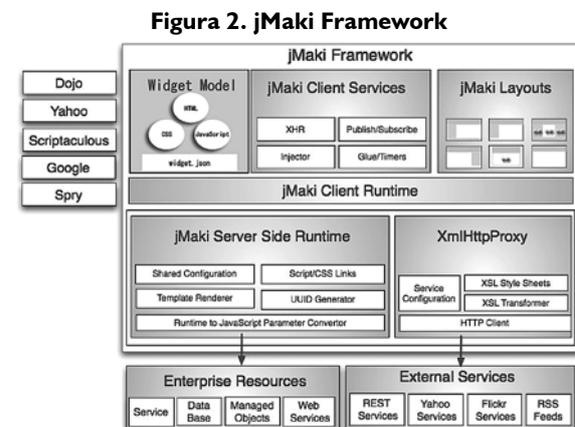
Los servicios RESTful⁹ son servicios construidos para trabajar mejor en la web. REST (Representational State Transfer) es un estilo arquitectónico que especifica restricciones tales como interface uniforme para trabajar con propiedades deseables en los servicios web tales como rendimiento, escalabilidad y modificabilidad que permiten a los servicios web trabajar mejor en la web.

De una manera muy sencilla, los pasos¹⁰ utiliza-

dos para crear la aplicación Jersey en Netbeans son:

1. Crear un proyecto de aplicación web sin framework.
2. Configurar la base de datos. Conectar a la base de datos, crear las tablas e insertar registros en las tablas.
3. Generar las clases Entity desde la base de datos.
4. Generar los servicios web RESTful desde las clases Entity.
5. Probar el servicio web RESTful.

4. APLICACIÓN JMAKI: CONSUMIENDO SERVICIOS WEB



La figura 2 muestra el framework jMaki¹¹, un framework liviano Cliente/Servidor para crear aplicaciones javascript para la web 2.0 usando CSS, modelo widget widget, servicios clientes tales como los eventos publish/suscribe para widgets, manejadores de acciones Javascript y un proxy genérico para interactuar con servicios web externos RESTful.

El framework jMaki utiliza un modelo para reu-

7. Sun Microsystems. Article The Java Persistence API – A Simpler Programming Model for Entity Persistence. <http://java.sun.com/developer/technicalArticles/J2EE/jpa/> Mayo de 2006.

8. Netbeans.org. A mashup using REST jMaki Table Component. <http://wiki.netbeans.org/RestjMakiUsecase>. 2007

9. Sun Microsystems. RESTful Web Services Developer's Guide. <http://docs.sun.com/app/docs/doc/820-4867/ggnyk?l=en&a=view>

10. Netbeans.org. Getting Started with RESTful Web Services. <http://www.netbeans.org/kb/docs/websvc/rest.html>. Consultado el 21 de junio de 2009.

11. Java.Net. About jMaki. <https://ajax.dev.java.net/about.html> . 2007

sar componentes Javascript llamado “jMaki Widget Model”. Su estructura es basada en HTML, Javascript y CSS. Este modelo permite crear nuestros propios widgets.

jMaki utiliza el mecanismo Publish/Subscribe¹² para la comunicación entre widgets^{13,14}, donde los publishers son los productores de eventos y los subscribers son los consumidores de eventos. Los widgets pueden ser ambos: productores y consumidores. Cuando un evento ocurre para un widget, el widget publica una notificación de este evento a un tópico. Un tópico es un string o un nombre que asocia el productor de un evento a los consumidores de este evento, permitiendo a productores y consumidores comunicarse asíncronamente y en forma desacoplada.

Los pasos utilizados para crear la aplicación jMaki^{15,16} son:

1. Crear un proyecto jMaki de dos columnas y un pie de página.
2. Adicionar, en la columna derecha, un widget “Google Maps”.
3. Adicionar, en la columna izquierda, un widget “Dojo Table”.
4. Adicionar al pie un widget “Yahoo Button”.

Yahoo Button toma los datos desde el servicio web RESTful, y publica a Dojo Table y a Google Maps.

En la misma aplicación jMaki se adiciona la API de Google para, por medio de código de Java, enviar información a Google Calendar.

5. PROBANDO GCEL CON LAS REDES SOCIALES DE LA FUNDACIÓN CONSENTIR

La Fundación Consentir ubicada en Barranqui-

lla, Colombia está conformada por redes sociales llamadas Células. Estas redes sociales tienen varios miembros, de los cuales uno de ellos se asigna como líder del grupo. Cada célula tiene una dirección física.

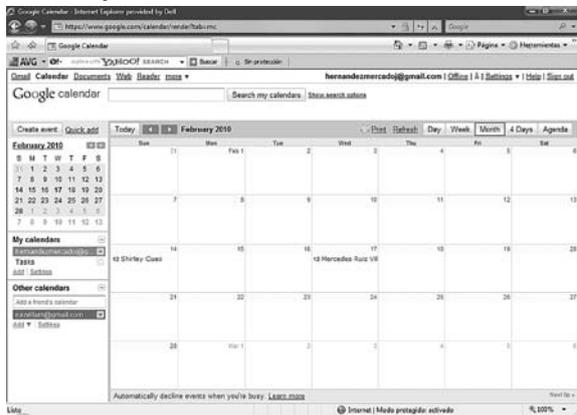
Figura 3. Localización geográfica de las Células de la Fundación Consentir



La figura 3 muestra la localización geográfica de estas células. Al lado izquierdo aparece el nombre, dirección y líder de la célula. Al lado derecho aparece la localización geográfica de cada una de las células de la Fundación. Al hacer click en el botón inferior, GCEL extrae la información de la base de datos por medio de un servicio web y lo publica a los widgets: Dojo Table y Google Maps. Al mismo tiempo envía a Google Calendar las fechas de cumpleaños de los líderes de estas células como se ilustra en la figura 4. Específicamente, el sistema agrega estas fechas al calendario del administrador del sistema. Para el caso del líder se agregará en su calendario las fechas de cumpleaños de sus miembros. El calendario ha sido configurado para manejar un recordatorio un día antes de la fecha de cumpleaños, tanto en pantalla como en e-mail.

12. Working with jMaki events. Carla Mott. http://blogs.sun.com/enterprisetechtips/entry/working_with_jmaki_events. Consultado en mayo de 2009.
 13. jMaki – widget talking to widgets. Carla Mott. http://weblogs.java.net/blog/carlavmott/archive/2007/07/jmaki_widgets_t.html. Consultado en mayo de 2009.
 14. jMaki publish/subscribe and debugging. Arun Gupta. http://blogs.sun.com/arungupta/entry/jmaki_publish_subscribe_and_debugging. Consultado en mayo de 2009.
 15. Gupta Arun. Travel Map-Another Real-life app using jMaki & Jersey. http://blogs.sun.com/arungupta/entry/screencast_web11_travel_map_another
 16. Gupta Arun. Delete/Update row from database using jMaki Data Table.2007. http://blogs.sun.com/arungupta/entry/totd_15_delete_update_row

Figura 4. Calendario que muestra las fechas de cumpleaños de los líderes de las redes sociales



Hay que tener en cuenta que para integrar Google Maps en nuestra aplicación web, es necesario tener una clave de API de Google Maps, la cual se obtiene registrando nuestra aplicación en la página de Google¹⁷.

Una consideración final: Colombia no tiene una distribución geográfica por área como la tienen otros países como Estados Unidos. Google Maps acepta, hasta el momento, la ciudad y una

carrera o una calle pero no la dirección completa. Para establecer las coordenadas exactas en latitud y longitud de una dirección usamos un programa de demostración de la galería de Google¹⁸ que genera la latitud y longitud al hacer un *click* en el mapa.

6. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se presentó la construcción de GCel, Mashup de calendarios y localización geográfica de redes sociales usando Google Calendar y Google Maps respectivamente. La aplicación fue construida utilizando los frameworks jMaki y Jersey.

Nuestro interés de investigación apunta a extender GCel hacia las siguientes direcciones: edición de coordenadas latitud/longitud¹⁹; ruta de desplazamiento de un lugar a otro²⁰; mapa en dispositivo móvil; jerarquización de las redes sociales. Para esto, deseamos aprovechar el potencial de la API de Google²¹: Groups, Calendar, Docs, entre otros.

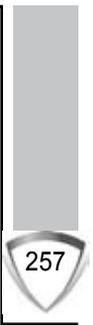
17. Registro en el API de Google Maps. <http://code.google.com/intl/es/apis/maps/signup.html>. 2009

18. Galería de demostraciones del API de Google Maps. LatLong Generator. <http://code.google.com/intl/es/apis/maps/documentation/demogallery.html>. 2009

19. Ejemplos del API de Google Maps. Ejemplos de marcadores. Marker-Drag. <http://code.google.com/intl/es-CO/apis/maps/documentation/examples/directions-simple.html>. 2009

20. Ejemplos del API de Google Maps. Ejemplos de servicios. Directions-simple. <http://code.google.com/intl/es-CO/apis/maps/documentation/examples/directions-simple.html>. 2009

21. Google Data APIs Client Libraries. <http://code.google.com/intl/es-CO/apis/gdata/clientlibs.html>. 2009



BIBLIOGRAFÍA

Ejemplos del API de Google Maps. Ejemplos de marcadores. Marker-Drag. <http://code.google.com/intl/es-CO/apis/maps/documentation/examples/directions-simple.html>. 2009

Ejemplos del API de Google Maps. Ejemplos de servicios. Directions-simple. <http://code.google.com/intl/es-CO/apis/maps/documentation/examples/directions-simple.html>. 2009

Galería de demostraciones del API de Google Maps. LatLong Generator. <http://code.google.com/intl/es/apis/maps/documentation/demogallery.html> 2009

Glassfish – Jersey. <https://jersey.dev.java.net/>. Consultado Junio 21 de 2009.

Google Data APIs Client Libraries. <http://code.google.com/intl/es-CO/apis/gdata/clientlibs.html>. 2009

Gupta Arun. Travel Map-Another Real-life app using jMaki & Jersey. http://blogs.sun.com/arungupta/entry/screencast_webII_travel_map_another

Netbeans.org. A mashup using REST jMaki Table Component. <http://wiki.netbeans.org/RestJMakiUsecase>. 2007

Netbeans.org. Getting Started with RESTful Web Services. <http://www.netbeans.org/kb/docs/websvc/rest.html>. Consultado el 21 de junio de 2009.

Project jMaki. <https://ajax.dev.java.net/> . 2007.

Registro en el API de Google Maps. <http://code.google.com/intl/es/apis/maps/signup.html>. 2009

Wikipedia. Red social. http://es.wikipedia.org/wiki/Red_social. 2009.

Sun Microsystems. Article The Java Persistence API – A Simpler Programming Model for Entity Persistence. <http://java.sun.com/developer/technicalArticles/J2EE/jpa/> mayo de 2006.

Sun Microsystems. RESTful Web Services Developer's Guide. <http://docs.sun.com/app/docs/doc/820-4867/ggnyk?l=en&a=view>

Techtear.com. Los Mashups, uno de los pilares de la web 2.0. <http://www.techtear.com/2007/03/26/los-mashups-uno-de-los-pilares-de-la-web-20/>. Consultado Junio 21 de 2009.

Wikipedia. Mashup (aplicación web híbrida). [http://es.wikipedia.org/wiki/Mashup_\(aplicaci%C3%B3n_web_h%C3%ADbrida\)](http://es.wikipedia.org/wiki/Mashup_(aplicaci%C3%B3n_web_h%C3%ADbrida)). 2009.

