Análisis, Diseño E Implementación De Un Marco De Trabajo Para La Generación Rápida De Aplicaciones Web Multimedia Interactivas

David Chang Villacreses
Carlos Villavicencio Moreira
Ing. Xavier Ochoa Chehab
Escuela Superior Politécnica del Litoral
Campus Gustavo Galindo Km. 30 y medio, Vía Perimetral, Guayaquil, Ecuador
dchang@fiec.espol.edu.ec, cvillavi@fiec.espol.edu.ec, xavier@cti.espol.edu.ec

Resumen

Este trabajo tiene como objetivo crear un marco de trabajo que pueda ser usado para la generación de aplicaciones Web interactivas utilizando e integrando servicios Web existentes de información de distintas clases de medios. Herramientas similares a la que se propone diseñar no han sido explotadas en el ámbito del desarrollo web. Este marco de trabajo está orientado a ser utilizado por desarrolladores web y la intención es sembrar una semilla para la evolución de la comunidad informática.
Inicialmente se realiza una exploración de las herramientas a utilizar para la implementación de soluciones web, se explora el concepto de Ajax y Mashups.
Se analiza la resolución del problema y todas las entidades que se relacionan con el marco de trabajo, y se propone el diseño de la solución, su modelo de funcionamiento, y el comportamiento dinámico del mismo. Se implementa el diseño antes mencionado utilizando los lenguajes de programación seleccionados para el desarrollo óptimo del marco de trabajo, y se detalla la lógica y funcionamiento de cada una de las clases que componen el sistema. Finalmente, se implementa un ejemplo para probar el funcionamiento del marco de trabajo.

Palabras Claves: *Ajax, Mashup, Framework, Búsqueda, Servicio Web, Json, API.*

Abstract

This work has as main objective the creation of a Framework that can be used in order to generate Interactive Web Applications using and integrating existing Information Web Services from different kinds of media. Tools of this nature have not been exploited in the web development community. This framework is oriented for use of web developers, as a seed to further evolution for the information technology community.
Initially, common tools are explored in order to developing web resources. The terms Ajax and Mashup are explained.
The solution of the problem and all entities related to the framework are analyzed. A design for this solution is given, considering its interaction model and its dynamic behavior. This design is implemented using programming languages selected for optimal framework performance, the logic used by the solution, the classes that make up the system, and the main functions are detailed.
Finally, an example is developed in order to test the Framework's performance.

1. Introducción

La variedad de medios y de servicios en la red es impresionante. Teniendo sitios que combinen muchos tipos de medios y además servicios cada vez más interesantes y creativos, la variedad de lo que se puede encontrar en la red crece a pasos agigantados. Existen nuevas tecnologías que se están generalizando y ciertos formatos en los que es posible encontrar esta información.

2. Herramientas y Aplicaciones Multimedia Web

Un medio es generalmente conocido como una forma de presentar un contenido. Existen muchos tipos de medios en línea, todos estos de libre acceso público, y su acceso es tan sencillo como el presionar un botón de "Buscar".

Es necesario tener estándares en los cuales la información debe de ser presentada, de lo contrario el desarrollo de cualquier aplicación sería un caos, pero tampoco puede existir un formato único, ya que cada uno es creado con el fin de satisfacer ciertos requerimientos y necesidades, tanto para el desarrollador, como para el computador.

Los formatos parcialmente estructurados, o semiestructurados están orientados a ser leídos tanto por navegadores y lenguajes de programación, como por programadores, y llevan este nombre porque sus estructuras están separadas por encabezados o tags legibles, es decir, la mayoría de estos tags están en lenguaje natural, de manera organizada y jerárquica.

Los formatos estructurados son formatos descriptivos que nos permiten dar forma o estructura a un objeto cualquiera de manera estandarizada. Se los denomina estructurados, porque están orientados a ser interpretados por los lenguajes de programación, ahorrando caracteres al ser usados, es decir, minimalizan el nivel de procesamiento en lo más posible. Dos de los formatos más utilizados para la web son el recientemente popularizado JSON y el XML.

Existen varias formas de obtener información de los sitios web cuyo contenido es predominantemente textual. Es posible obtener documentos estructurados, o simplemente clasificarlos por su contenido. Entre estas tecnologías, dos de las más usadas son web crawling y scraping, provistas por distintos servicios web.

A groso modo, un servicio web, como su nombre lo indica, provee utilidades para que dos equipos se comuniquen a través de la red.
Todos los medios antes vistos son alojados en servidores distintos, y como se pudo apreciar, algunos de estos sitios proveen servicios web para organizarlos e incluso para compartirlos, independientemente de la forma de desarrollo, un servicio web debe proveer facilidad para que cualquier cliente pueda obtener la información que necesita. Los servicios más utilizados por los internautas mayormente son de clasificación de información y de búsqueda.

Los Mashup son sitios web que han combinado contenido y funcionalidad de otros sitios mayores, los cuales proveen públicamente sus APIs. Con estas herramientas es posible mostrar información de manera inteligente de sitios que pueden ser rivales como Google y Yahoo, pero de manera transparente para el usuario final, quien después de todo es el mayor beneficiado, aunque muchas veces no es capaz de confiar en la capacidad de un Mashup.
Al inicio se fueron integrando pocas APIs, como ejemplo se puede citar una búsqueda de imágenes de un servicio de repositorio por medio de un mapa mundial de Google Maps, pero cada vez se unen más APIs o cualquier otro medio como RSS feeds, con el fin de obtener la mayor cantidad de información posible y siempre innovando para no caer en monotonía. Existen sitios que ofrecen alojamiento e incluso bases de datos enteras para que los desarrolladores ingresen sus proyectos que mezclan diferentes APIs.

La tecnología usada por los Mashups lleva el nombre de Ajax. Éstas son las siglas en inglés que describen "*Asyncronous JavaScript and XML*". Ajax utiliza JavaScript, un código web en el lenguaje Java, para la ejecución remota a nivel del servidor, o fuera de nuestros equipos, para modificar la página que se está viendo de manera dinámica, es decir, que solo se carga lo que se necesita ver, en la misma página, o en otra si así se lo desea.

3. Analisis del problema

En internet se puede observar pocos mashups que realicen búsquedas simultaneas, y la mayoría lo único que ofrece es clasificar los resultados por motor de búsqueda, añadirlos a una gran lista o mezclarlos sin ninguna restricción. Pocos son los buscadores que realizan alguna innovación para el usuario y que ofrecen de forma transparente una utilidad inigualable al combinar servicios distintos.

En otras palabras, todos estos distintos tipos de medios son similares en resultados, pero tienen muy distintas formas de acceso. Como se puede apreciar, es cada vez más necesario el poder trabajar con estos medios de diferentes fuentes, como si fueran una sola. El propósito de este trabajo es el de unificar estas fuentes, y de proveer una capa para permitir a los desarrolladores futuros el uso generalizado y sencillo de los resultados de estas. Este Framework será una solución para este nuevo problema.

El framework utilizará diversas fuentes de servicios web para realizar búsquedas tomando en cuenta el tipo de medio. Los resultados serán almacenados en objetos de tipo de dato abstracto que representarán los diversos tipos de medio que se manejarán, estos son textual, imágenes, video, audio y archivos independientes. Estos objetos serán denominados repositorios y los datos de resultados guardarán un formato adecuado y estándar para su fácil utilización y organización.

El sistema también será orientado para el uso de motores de búsqueda integrados, o para el desarrollo de homepages, que utilizarán las capacidades del framework combinándolo con una interfaz amigable y usable para el usuario, quien busca encontrar las mejores opciones de resultados a una búsqueda.

4. Análisis y Diseño del Framework

4.1. Descripción de los servicios del sistema

El sistema principalmente se comunicará con los APIs de desarrollo soportados, principalmente interfaces en JSON o XML, formatos estándares semiestructurados, los cuales serán traducidos a un formato estándar del framework, para ser unificados en un repositorio de información. Los servicios soportados incluyen resultados web de Yahoo!, SearchMash (una interfaz del motor de Google), y Microsoft Live. Resultados de repositorios de videos como Youtube, Yahoo y Google Video, otros medios soportados incluyen podcasts desde los servidores de iTunes y Odeo, imágenes de los servidores de Flickr, y resultados de buscadores de noticias y feeds, como Technorati. Algunas empresas como Microsoft, Yahoo y Google ofrecen múltiples servicios para distintos tipos de medios, los cuales también estarán soportados por el sistema. El mecanismo de la comunicación con los servicios web se ejemplificará más adelante en este artículo.
Otros servicios interesantes incluyen la creación de mapas interactivos, y servicios de scraping para directorios de archivos alojados en la red. Este último permite crear listas de reproducción, si un directorio posee archivos de formato mp3.

4.2. Ubicación del Framework

Desde el usuario objetivo que se encuentra al final de la cadena, es decir la persona que necesita información y realizará las búsquedas a través de una interfaz que utilice FindJira framework, hasta el internet que es la fuente de toda información en línea, los datos viajan a través de varias etapas a las que se denominan capas, estas capas se comunican entre sí y a su nivel, el mismo que cada vez es más bajo. FindJira framework actuará en la mitad del camino, como puente entre el desarrollador y hacia la comunicación con los APIs de los servicios web que indexan contenidos de internet.

Figura 1. Ubicación del framework entre el usuario y el internet.

Usuario - Usuario meta al que estará dirigido todo servicio que entrega información, está constantemente informado de los servicios que se encuentran en la red y es su decisión cual le conviene más para obtener la información que necesita.
Aplicación - Es una aplicación que utilizará el FindJira Framework y se comunicará con el usuario. La tecnología que mejor se ajusta lo que se necesita para crear un ejemplo que demuestre el potencial del FindJira Framework es Ajax. Esta capa es la que implementará el desarrollador que utilice el framework.
FindJira Framework - El marco de trabajo que obtendrá la información de algunos servicios y la tendrá disponible para que sea utilizada oportunamente a través de una interfaz, este capítulo se concentrará en explicar su diseño.
Internet - La fuente de todo tipo de información en línea.
Web API - Interfaces que ofrecen compañías para acceder a los resultados de búsqueda, el API de cada servicio es único y están orientadas a desarrolladores.
Servicios Web - Son las compañías que brindan servicios de búsqueda a través de toda la red, las cuales se explicaron en detalle en el capítulo anterior.

4.3. Diseño e Implementación del Sistema

Hasta este punto ya se ha revisado como se piensa implementar el sistema, y que servicios son los que el Framework soportará. A continuación, se revisará como el sistema trabajará internamente. La secuencia en que los módulos del sistema interactúan se puede apreciar en la figura siguiente.

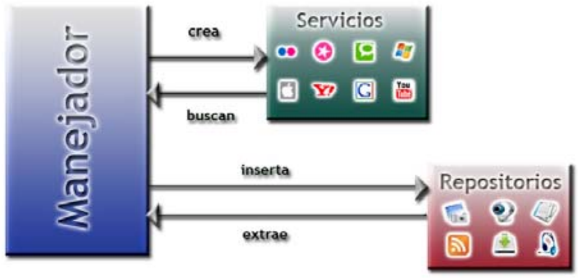

Figura 2. Diseño Dinámico del Framework.

Primeramente, para obtener resultados de los servicios web en formato puro, o raw, es necesario crear un objeto del servicio a minar en el código del handler. Éste objeto será el encargado de realizar los pedidos en el formato de dicho servicio. Por ejemplo, si un desarrollador desea realizar búsquedas en el servicio de Yahoo, este deberá crear un objeto del servicio respectivo, después de esto, deberá especificar el query de búsqueda, y opcionalmente, el número de resultados a obtener, o desde que resultado se desea obtenerlo, y si es requerido, especificar algún filtro de búsqueda provisto por Yahoo para la misma. Una vez hecho esto, el desarrollador deberá llamar a la función de búsqueda del medio correspondiente, es decir, deberá especificar si desea buscar imágenes, web, video, etcétera. Esta información es enviada al servicio, y una vez llamada la función de búsqueda, éste preparará con los datos provistos un pedido o URL de request en el objeto de Yahoo. La respuesta retornada por Yahoo está en formato JSON, para el caso de la implementación del Framework, para otros servicios el formato puede variar. Para obtener esta respuesta en formato nativo, es posible llamar a la función respectiva del servicio, y se obtendrá una cadena de caracteres con el contenido sin procesar de la respuesta.

El objetivo del Framework, es el obtener una respuesta estandarizada. Es por esto que una vez realizados los pasos anteriores, es también posible obtener la respuesta en Formato FindJira, es decir, el formato estandarizado descrito en los apartados anteriores. Para crear la respuesta en este formato es necesario que cada módulo de cada servicio tenga implementado un traductor o parser de cada uno de los formatos propios al formato FindJira. Dado que estos resultados "raw" son simples cadenas, es necesario convertirlas de su formato de origen a un formato comprensible para el lenguaje, en este caso, un arreglo asociativo, o de niveles jerárquicos de PHP. Esta transformación ya fue descrita cuando se mencionó la clase de servicio, y depende, de librerías internas a PHP, o en su defecto, librerías externas para el manejo de formatos no soportados por este lenguaje.

Este resultado en formato FindJira es devuelto al handler para su manipulación. En este momento, la lógica del handler implementado en el Framework insertará este resultado en un objeto repositorio persistente creado previamente para cada tipo de medio. La inserción de cada una de estas clases de repositorio se encargará de mantener datos de las distintas fuentes organizados. Los datos en el repositorio, como se conoce, estarán listos para su inserción en la web.

El handler podrá, en este punto realizar más búsquedas, limpiar los repositorios o extraer información de los mismos. Para éste último caso, se utilizarán los métodos de la clase repositorio que podrán extraer todos los elementos o un rango de los mismos. Éstos resultados pueden entonces ser trabajados con una interfaz dinámica realizada en Ajax o alguna otra tecnología web para su inserción o uso apropiado en una interfaz.

Para el caso de los mapas interactivos, el handler irá generando el código web para cada servicio. El handler se comunicará con el servicio de mapas, el cual a su vez hará uso del API de su servicio respectivo, actualizando sus puntos geográficos válidos, y una vez llenos estos datos previos, se mostrará el mismo llamando a la función respectiva. El handler podrá llamar otros mapas utilizando nuevos objetos de mapas, e incluso refrescar un mapa actualizado volviendo a llamar a las funciones que los muestran.

5. Ejemplo y Pruebas

En este trabajo se han descrito muchas formas de servicios web, y los muchos medios que estos servicios exploran e indexan. Se explicará el proceso de realización de un Mashup con el Framework, un buscador integrado. En este buscador se darán a conocer las funciones más interesantes del Framework, y se mostrará cómo el mismo interactúa con la aplicación desarrollada.

Al ser una aplicación web dinámica, la página ejemplo, llamada Iguana search, deberá tener funcionalidades implementadas en Ajax. Las funcionalidades principales del sitio son, como es lógico, la búsqueda y la presentación dinámica de los resultados en la misma página, y la comunicación con archivos PHP externos pertenecientes al Framework.
Iguana Search como producto final no utiliza todo lo que FindJira framework ofrece, como el soporte de sindicación en feeds, scraping de archivos en páginas web de directorios, entre otros. Pero ofrece una búsqueda dinámica de los tipos de medios más comunes en la red, como contenidos textuales, de audio y de video son minados y mostrados.

Este ejemplo fue colocado en la red y sometido a pruebas, para probar la robustez del mismo, y del framework, se obtuvieron resultados positivos de parte de los usuarios. El ejemplo puede ser visualizado en www.findjira.com. Se concluyó que el framework funciona de manera deseada y se decidió implementar y pulir ciertas opciones del mismo, como el soporte para otros servicios web, y otros idiomas, para ofrecer un mejor servicio en la red.

6. Referencias

[1] Michael Mahemoff (Junio 2006). *Ajax Design Patterns*. Primera Edición. 655 páginas.

[2] David Sklar, Adam Trachtenberg (Agosto 2006). *PHP Cookbook*. Segunda Edición. 810 páginas.

[3] Christopher Schmitt, Dan Cederholm (Agosto 2004). *CSS Cookbook.* Primera Edición. 252 páginas.

[4] Christian Gross (Diciembre 2006). *Ajax and REST Recipes: A Problem-Solution Approach*. Primera Edición. 360 páginas.

[5] Christian Heilmann (Julio 2006). *Beginning JavaScript with DOM Scripting and Ajax: From Novice to Professional.* Primera Edición. 512 páginas.

[6] Steve Ph.D. Holzner (Marzo 2006). *AJAX for dummies*. Primera Edición. 384 páginas.

[7] Richard Mansfield (Marzo 2005). *CSS Web Design for dummies*. 380 páginas.

[8] Nicholas Chase (Agosto 2006). The Ultimate Mashup - Web services and the semantic Web

[9] Unicode Consortium. 1996. The Unicode Standard. Version 2.0. Reading: Addison-Wesley.

[10] Programmable Web. *API List by category* [en línea]. <http://www.programmableweb.com/apilist/bycat> [Consulta: 14 octubre 2006]

[11] Web Mashup [en linea] <http://www.webmashup.com/> [Consulta: 15 octubre 2006]

[12] Google GData API. [en línea]. <http://code.google.com/apis/gdata/> [Consulta: 18 octubre 2006]

[13] Yahoo! Developer Network For Yahoo! Search Web Services [en línea] <http://developer.yahoo.com/search/> [Consulta: 13 diciembre 2006]

[14] Google SOAP Search Documentation [en línea] <http://code.google.com/apis/soapsearch/reference.html> [Consulta: 13 diciembre 2006]

[15] Flickr Services API Documentantion [en línea] <http://www.flickr.com/services/api/ > [Consulta: 17 diciembre 2006]

[16] YouTube API Documentantion [en línea] < http://www.youtube.com/dev_docs> [Consulta: 18 diciembre 2006]

[17] Wikipedia: Special Export [en linea] <http://www.mediawiki.org/wiki/Parameters_to_Special:Export> [Consulta: 19 diciembre 2006]

[18] PHP Online Manual [en linea] <http://www.php.net/docs.php> [Consulta: 28 diciembre 2006]

[19] Ajax Patterns Frameworks List [en línea] <http://ajaxpatterns.org/Frameworks> [Consulta: 26 diciembre 2006]

[20] MSDN *Live Search API* [en linea] <http://msdn2.microsoft.com/en-us/library/bb251794.aspx> [Consulta: 12 enero 2007]

[21] W3C: Cascading Stylesheets Homepage [en linea] <http://www.w3.org/Style/CSS/> [Consulta: 8 enero 2007]

[22] Wikipedia: Codec Article [en linea] <http://en.wikipedia.org/wiki/Codec> [Consulta: 10 enero 2007]

[23] Wkipedia: Syndication [en linea] <http://en.wikipedia.org/wiki/Web_syndication> [Consulta: 10 enero 2007]

[24] The Atom Project: RSS and Atom format Comparison [en linea] <http://www.intertwingly.net/wiki/pie/Rss20AndAtom10Compared> [Consulta: 10 enero 2007]

[25] Harvard Law: RSS Specification [en linea] <http://cyber.law.harvard.edu/rss/rss.html> > [Consulta: 10 enero 2007]

[26] Macromedia Flash SWF Format Specification <http://www.adobe.com/licensing/developer/> [Consulta: 10 enero 2007]

[27] Snook.ca - *iTunes search API Around the Corner* [en linea] <http://snook.ca/archives/javascript/itunes_search_a/> [Consulta: 15 enero 2007]

[28] Joshua Bloch, *How to Design a Good API and Why it Matters* [en linea] <http://lcsd05.cs.tamu.edu/slides/keynote.pdf> [Consulta: 15 enero 2007]

[29] Wikipedia: Application Programming Interface Article [en linea] <http://en.wikipedia.org/wiki/Application_programming_interface> [Consulta: 18 enero 2007]

Firma de los Autores:

DAVID CHANG VILLACRESES
(Autor)

CARLOS VILLAVICENCIO MOREIRA
(Autor)

XAVIER OCHOA CHEHAB
(Director de Tesis)