Diseño e Implementación de un Web API para el Sistema Interactivo de Desarrollo para el Web (SIDWeb).

Giancarlo Vera [(1)]; Xavier Ochoa, Ph.D. [(2)]
Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km. 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
giancarlo.vera.rivera@gmail.com [(1)], xochoa@espol.edu.ec[(2)]

Resumen

El presente artículo presenta el diseño y la implementación de una interfaz de comunicación para el software de aprendizaje electrónico SIDWeb 4, con la cuál se espera proveer un mecanismo de interacción con los contenidos y funcionalidades del software en cuestión con aplicaciones externas, para esto se realizó un análisis del problema del despliegue de aplicaciones Web enfocadas a navegadores de escritorio en dispositivos móviles, la solución propuesta para el caso del software SIDWeb 4, detalles técnicos de la implementación realizada bajo el lenguaje de programación PHP haciendo uso del Framework Symfony, y finalmente como parte de las pruebas de la implementación se presenta el funcionamiento y resultados de pruebas con usuarios de un aplicativo desarrollado con jQuery Mobile y JavaScript que hace uso de la interfaz de comunicación desarrollada.

Palabras Claves: *API, LMS, Servicio Web, PHP, Symfony, Aplicaciones Móviles.*

Abstract

This article presents the design and implementation of a communication interface for e-learning software SIDWeb 4, this interface is expected to provide an interaction mechanism with the content and functionality of the software SIDWeb 4 with external applications, this work presents an analysis of presenting Web applications focused on desktop browsers on mobile devices problem, the solution proposed for SIDWeb 4 software, technical details of the implementation under the PHP programming language using the Symfony Framework, and finally as part of the functionality tests this work presents the implementation and results of users testing a Mobile application developed with jQuery Mobile and JavaScript that uses the communication interface developed.

Keywords*: API, LMS, Webservice, PHP, Symfony, Mobile Applications.*

1. Introducción

SIDWeb es un software de aprendizaje electrónico, de código abierto desarrollado por el Centro de Tecnologías de Información, este software es usado ampliamente por la comunidad de profesores y estudiantes de la ESPOL para el manejo de contenidos y recursos adicionales de clases dictadas por la universidad.

La idea de implementación de este trabajo se originó luego de una consulta realizada al Dr. Xavier Ochoa sobre la posibilidad de implementar una versión móvil del Sistema Interactivo de Desarrollo para el Web (SIDWeb), sobre lo cual el Dr. Ochoa propuso la implementación de un API que expusiera las operaciones del sistema para posteriormente utilizarlo en implementaciones de aplicativos móviles o interacciones con otras aplicaciones que hicieran uso de las funcionalidades de SIDWeb.

El objetivo principal de este trabajo es implementar un API de comunicaciones que permita la interacción de aplicaciones externas con el software SIDWeb 4.

2. Análisis del Problema

Debido al creciente uso de dispositivos móviles por parte de estudiantes, profesores y otros usuarios del SIDWeb es necesario considerar los aspectos alrededor del uso del sistema en esta clase de dispositivos, actualmente el acceso al sitio Web de SIDWeb desde dispositivos móviles posee importantes limitaciones inherentes a las capacidades del dispositivo móvil, otro aspecto importante a considerar dada la interacción entre aplicaciones es la implementación de mecanismos que permitan a estas diferentes aplicaciones comunicarse entre sí.

2.1. Limitaciones de acceso en dispositivos móviles

Entre las limitaciones de acceso a SIDWeb en dispositivos móviles se pueden mencionar:

- Aumento de tiempo en cargas de contenidos debido a bajas velocidades de conexión y a limitada capacidad de procesamiento de dispositivos móviles, lo que provoca velocidades menores en el renderizado de los contenidos [1].
- Las interfaces móviles comúnmente son más restrictivas que las interfaces de escritorio en tamaños de pantalla, lo que causa que sitios diseñados para interfaces de escritorio requieran más desplazamientos y acercamientos para ser navegados en interfaces móviles [2].
- Muchas de las aplicaciones para navegación en dispositivos móviles carecen de soporte para ciertas características como JavaScript, objetos Flash, etc.

2.2. Comunicación con SIDWeb desde otras aplicaciones

Actualmente las operaciones de SIDWeb no están expuestas para plataformas diferentes a navegadores, debido a que, los contenidos generados por el software solo son legibles por navegadores Web, esto provoca que no sea posible que otros sistemas se comuniquen con el SIDWeb para propósitos de integración o intercambio de información, esto representa una limitante si se considera la opción del desarrollo de aplicaciones externas que interactúen con SIDWeb.

3. Análisis de la Solución

La solución propuesta es una interfaz de programación de aplicaciones (API) que exponga las operaciones más críticas del software SIDWeb en un lenguaje estándar que permita la comunicación con otras aplicaciones, el nombre asignado a esta interfaz es SIDWeb API.

El API implementado hace uso de servicios Web proporcionados por la universidad, tales como Directorio ESPOL y CAS ESPOL para realizar la autenticación de usuarios que usen las funcionalidades expuestas por el API.

Las funcionalidades que se implementan en el API constan el acceso a:

- Información y planificación de cursos.
- Calendario de actividades.
- Visualización de Anuncios.
- Visualización de Tareas
- Interacción con foros.

4. Diseño de la Solución

SIDWeb API ha sido desarrollado en el mismo entorno que el software SIDWeb, codificado con PHP y bajo el Framework Symfony este API consta como uno de los módulos del software, esto facilita la instalación / desinstalación del módulo dentro del software.

4.1 Componentes de la Solución

En la figura 1 se muestran los componentes y los enlaces de interacción entre los mismos, el Cliente SIDWeb API es cualquier aplicación que haya sido registrada para el uso del API y que realice requisiciones al sistema.

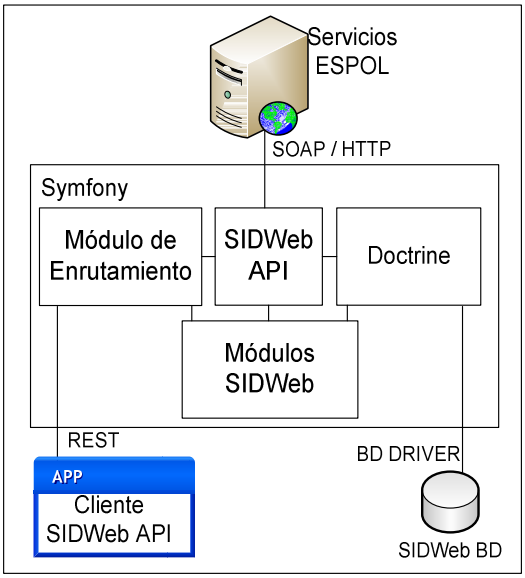

Figura 1. Componentes de la solución

El módulo de enrutamiento es un componente del Framework Symfony que se encarga de analizar patrones de URLs en las requisiciones hacer comparaciones con un archivo de patrones definido y de acuerdo a la comparación re-direccionar las requisiciones a un método específico para que sean procesadas, esta operación facilita la organización de los archivos en el proyecto dado que centraliza el proceso de direccionamiento de requisiciones.

El módulo SIDWeb API se encarga de procesar requisiciones enviadas por el módulo de enrutamiento, en sí éste pertenece al conjunto de Módulos de SIDWeb pero para efectos del diagrama se lo ha separado para hacer explícitas las relaciones entre SIDWeb API con los demás módulos de SIDWeb, a su vez SIDWeb API puede tener interacciones con otros módulos para obtener información o con la capa de abstracción de base de datos Doctrine para obtener datos directamente de la base de datos.

Doctrine es un ORM [3] integrado a Symfony que provee una capa de abstracción para la comunicación con la base de datos (SIDWeb BD), esto facilita que la aplicación sea escrita sin dependencias a un motor de base de datos específico, además de proporcionar un entorno orientado a objetos para el manejo de los datos en la base de datos.

El componente Servicios ESPOL se refiere específicamente a los servicios de Directorio ESPOL y CAS ESPOL los cuales tienen interacción con el módulo SIDWeb API para realizar las operaciones de autenticación de usuarios, el primero es utilizado en operaciones de autenticación con credenciales manejadas por la aplicación cliente, mientras que el segundo se utiliza cuando se requiere una autenticación donde la aplicación cliente no maneje credenciales del usuario para lo cual se redirecciona al usuario a una página manejada por ESPOL donde se entregan las credenciales y se devuelve un ticket de acceso que luego puede ser verificado con el servicio CAS.

4.2 Flujo de requisiciones

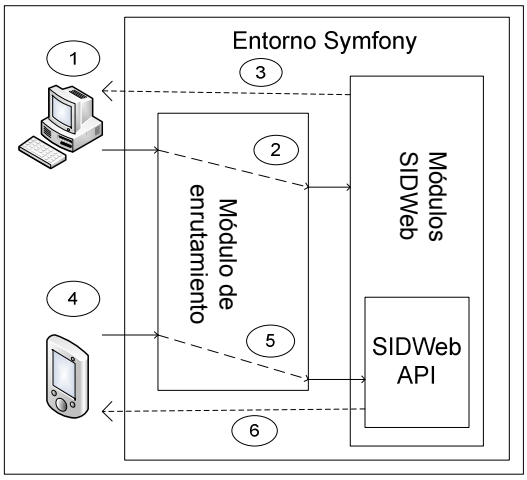

Figura 2. Diagrama de Flujo de Requisiciones

En la figura 2 se muestra el flujo de requisiciones, la parte superior muestra las requisiciones dirigidas a módulos diferentes a SIDWeb API, en ese caso el flujo es el siguiente:

1. Un cliente Web, tal como un navegador origina una requisición a SIDWeb.
2. Una vez en el entorno de Symfony, la requisición es analizada por el módulo de enrutamiento en el cual se redirecciona de acuerdo al mapeo que se haya especificado, con esto por ejemplo una requisición a la URL: "http://beta.sidweb.espol.edu.ec/AuthCAS/logout" es redirigida al módulo "AuthCas" a la ejecución del método "logout", la información de patrones y a que módulos/métodos se debe redirigir es contenida en un archivo de mapeo.
3. Luego de procesar la requisición el módulo devuelve una respuesta al cliente Web.

Las requisiciones dirigidas a SIDWeb API son mostradas en la parte inferior de la figura 2, estas requisiciones contienen URLs que siguen el patrón: [URL_BASE]/sidwebapi/*, para estas requisiciones el flujo es el siguiente:

4. Un cliente del API, tal como una aplicación en un dispositivo móvil, genera una requisición con el patrón mencionado.
5. La requisición en redirigida por el módulo de enrutamiento al módulo de SIDWeb API mediante el mapeo especificado.
6. SIDWeb API procesa la requisición y devuelve una respuesta en formato JSON.

4.3. Códigos de respuesta

Las respuestas de requisiciones que entrega SIDWeb API incluyen códigos HTTP de respuesta, estos códigos representan el status de la operación realizada, así una respuesta con código 200 ("OK") significa que la operación fue completada exitosamente, en cambio un código entre los rangos 3xx, 4xx, 5xx significa que la requisición no fue procesada con éxito, estos códigos son útiles para comprobar rápidamente del lado del cliente si hubo algún error al procesar la requisición, la tabla 1 muestra el detalle de los códigos de respuestas utilizados en la implementación de SIDWeb API.

Tabla 1. Códigos de respuesta entregados por SIDWeb API

Código	Estatus	Descripción
200	OK	Éxitosa.
400	Bad Request	Petición inválida, un mensaje de error explicará la razón.
401	Unauthorized	No autorizado, un mensaje de descripcion explicará el error
404	Not Found	El URI solicitado es inválido, o el recurso solicitado no existe.
500	Internal Server Error	Error interno del API.

5. Implementación

La implementación de SIDWeb API fue realizada usando varias tecnologías como PHP, el Framework Symfony, el formato de intercambio de datos JSON y SSL para el uso de canales seguros.

5.1. PHP – Symfony

SIDWeb API fue desarrollado enteramente en PHP bajo el Framework Symfony dado que fue el lenguaje y el Framework facilitaban la integración del API como un módulo del software SIDWeb 4.

Symfony posee características que fueron de mucha importancia como el enrutamiento de requisiciones [4] y la posibilidad de agregar el API en su totalidad como módulo del software SIDWeb.

Las invocaciones al servicio de CAS-ESPOL el cual provee autenticación sin manejo directo de credenciales fueron realizadas mediante la librería CURL de PHP, esta permite conectarse a servidores con diferentes tipos de protocolo tales como: http, https, ftp, Telnet, entre otros. Para las invocaciones al servicio de Directorio ESPOL se utilizó la extensión de SOAP de PHP la cual encapsula la complejidad de la conexión y permite manejarla como un objeto facilitando la codificación.

5.2. JSON

La notación de objetos JavaScript "JSON" por sus siglas en inglés es un formato liviano de rápido intercambio de datos [5], independiente de plataforma, leíble por humanos dado que es un formato de texto para serialización de estructuras de datos. JSON puede representar cuatro tipos primitivos de datos: cadenas de texto, números, nulos y datos lógicos (boolean), también puede representar dos tipos estructurados: Objetos y Arreglos.

La ventaja de JSON sobre XML al momento de intercambio de datos, es su simplicidad y la poca sobrecarga en las representaciones en comparado con XML [6], esto lo hace ideal para intercambio de datos entre cliente – servidor.

Los métodos de SIDWeb API entregan las respuestas en formato JSON, para la construcción de las estructuras JSON se hace uso de la función de PHP json_encode, esta función toma como argumento un arreglo de datos y devuelve la representación JSON del arreglo de datos en formato de texto, esto facilita la construcción de las estructuras JSON dado que el proceso se reduce en construir un arreglo de datos con la información necesaria y luego usar la función "json_encode" para obtener la representación JSON.

```
[
 {
  "rol":"Estudiante",
  "materia_nombre":"Aplicaciones Web",
  "sitioweb_token":"88e198056asd445265huic"
 }
]
```

Figura 3. Ejemplo de representación de objeto mediante JSON

La figura 3 muestra un ejemplo de la representación de un objeto registro de estudiante con el formato JSON.

5.3. Canales Seguros

Las invocaciones a SIDWeb API deben ser realizadas utilizando canales seguros, dado que involucran información sensible tal como credenciales de usuario, datos personales de usuario, información de registros, entre otros. El uso de canales seguros previene principalmente ataques del "hombre en la mitad". Para la implementación de canales seguros se hace uso del protocolo SSL/TLS el cual provee comunicaciones encriptadas e identificación segura de un Web Server.

Para el uso del canal seguro en el entorno de desarrollo se generó un certificado para el servidor mediante la herramienta OpenSSL y se modificó la configuración del servidor Apache para el uso de https.

6. Pruebas

Para realizar las pruebas de la aplicación se construyeron dos aplicaciones clientes, una consola de pruebas y una aplicación Web dirigida a dispositivos móviles llamada SIDWeb Móvil, finalmente se probaron las funcionalidades del API mediante una prueba con 5 usuarios.

6.1. Consola de Pruebas

La consola de pruebas de SIDWeb API es una pequeña aplicación basada en PHP y con algunas funcionalidades en JavaScript, mediante la cual fue posible realizar las siguientes acciones:

- Realizar invocaciones a los métodos expuestos por SIDWeb API.
- Especificar parámetros para las invocaciones a los métodos.
- Obtener el formato de texto JSON de la respuesta.
- Obtener el código de respuesta de la invocación.

Los resultados de las invocaciones en la consola de pruebas muestran información acerca de la requisición enviada, el código de respuesta recibido, el texto JSON y la decodificación de JSON adquirida mediante el método "var_dump" de PHP.

6.2. Aplicativo SIDWeb Móvil

Con el propósito de probar las funcionalidades de SIDWeb API en un ambiente multiplataforma que sea compatible con despliegue en dispositivos móviles se desarrolló un aplicativo llamado SIDWeb Móvil basado en JavaScript que hace uso del framework jQuery Mobile.

JQuery Mobile es un conjunto de plugins y widgets enfocados a proveer un API multi-plataforma para crear aplicaciones Web [7]. Mediante jQuery Mobile se accede al uso de transiciones, estilos, menús, diálogos, eventos de toques de pantalla (touch events) dirigidos a dispositivos móviles.

El aplicativo fue probado bajo el emulador Ripple, el cual es un emulador multi-plataforma para realizar pruebas de aplicaciones basadas en HTML5, este emulador se ejecuta como una extensión del navegador Web Google Chrome, también fue probado con un emulador de la plataforma Android que se ejecuta como un plugin del IDE Eclipse, también se realizaron pruebas bajo dispositivos con el sistema operativo iOS 5 (iPhone 4, iPad 2).

Figura 4. Pantalla de Login de SIDWeb Móvil

La figura 4 muestra la pantalla de Login del aplicativo SIDWeb Móvil ejecutándose sobre el emulador Ripple, al realizarse la operación de login el aplicativo realiza una requisición de autenticación a SIDWeb API con las credenciales del usuario y la llave de la aplicación para obtener un ticket de acceso para las consecuentes invocaciones.

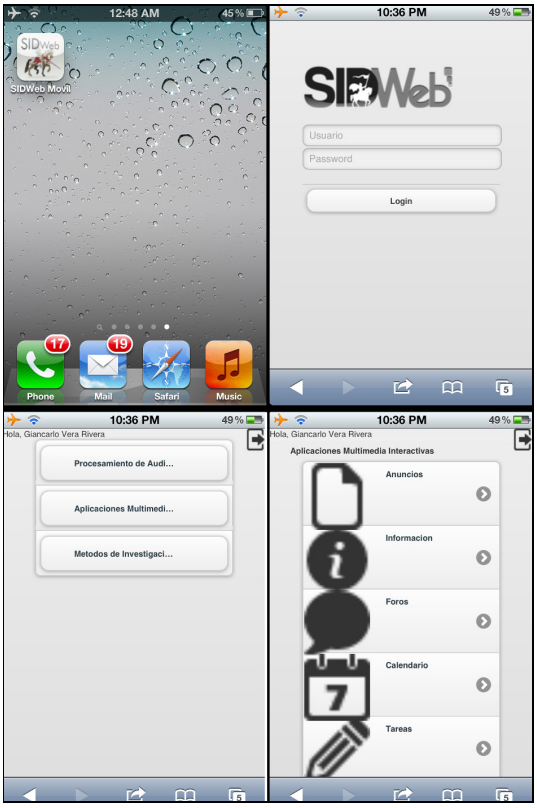

Figura 5. SIDWeb Móvil en iPhone 4

La figura 5 muestra el despliegue del aplicativo SIDWeb Móvil bajo un dispositivo iPhone 4.

6.3. Pruebas con Usuarios

El objetivo de las pruebas con usuarios era constatar que se recibieran las respuestas adecuadas a las requisiciones realizadas a SIDWeb API mediante el uso de la aplicación SIDWeb Móvil. Se utilizaron las funcionalidades que ofrece el aplicativo móvil desarrollado sobre un dispositivo iPhone 4 con sistema operativo iOS 5, a cada usuario se le nombró las tareas que debía realizar, se constato que haya realizado la tarea y se midió el tiempo que le tomo realizarla. La tabla X muestra los resultados de la prueba.

Las pruebas se realizaron con cinco individuos los que fueron identificados con el código "Usr(x)" (donde (x) es un número entre 1 y 5), los perfiles de los usuarios eran variados, todos los usuarios estaban habituados al uso de aplicaciones Web, el usuario 1 y 5 eran estudiantes que usan el software SIDWeb, el usuario 2 y 3 tenían familiaridad con la administración del software SIDWeb, mientras que el usuario 4 no era estudiante ni tenía familiaridad con la administración del software. Para los ensayos se creó un curso de prueba que contenía datos en las secciones de anuncios, foros, calendario y tareas.

Tabla 2. Resultados de las pruebas con usuarios

Tareas	Usr 1		Usr 2		Usr 3		Usr 4		Usr 5	
	Resultado	Tiempo (s)	Resultado	Tiempo (s)	Resultado	Tiempo (s)	Resultado	Tiempo (s)	Resultado	Tiempo (s)
Hacer Login	OK	15	OK	17	OK	16	OK	15	OK	16
Ingresar a Curso	OK	5	OK	5	OK	4	OK	4	OK	5
Ver Anuncios	OK	6	OK	8	OK	4	OK	7	OK	6
Ver Foros	OK	7	OK	6	OK	7	OK	6	OK	6
Responder Foro	OK	13	OK	14	OK	20	OK	15	OK	15
Ver Tareas	OK	5	OK	6	OK	7	OK	5	OK	6
Ver Calendario	OK	7	OK	8	OK	6	OK	6	OK	7
Hacer Logout	OK	3	OK	3	OK	4	OK	5	OK	4

Como se observa en la tabla IV todos los usuarios realizaron satisfactoriamente las pruebas, los tiempos más extensos se observan en las tareas de "login" y "responder foro" dado que estas tareas implicaban ingreso de datos, mientras las tareas que solo implicaban navegación en la aplicación poseen tiempos mucho menores.

Se realizó una prueba sobre un dispositivo iPhone 3 bajo el sistema operativo iOS 3.1, esta prueba falló dado que el aplicativo SIDWeb Mobile esta construido sobre jQuery Mobile y este tiene soporte desde la versión 3.2 de iOS.

6.4. Discusión de los resultados

Los resultados esperados por este trabajo era construir una interfaz que habilite la comunicación del software SIDWeb 4 con otras aplicaciones las cuales se basen o integren sus servicios con SIDWeb 4, la interfaz desarrollada: SIDWeb API permite la comunicación de SIDWeb 4 con otras aplicaciones tal como se aplicó con la comunicación del aplicativo SIDWeb Móvil y la consola de pruebas, al haber sido desarrollado utilizando un formato de respuesta sencillo y liviano como JSON los tiempos de carga y actualización de información mejoran.

Durante las pruebas realizadas con los usuarios se recibieron recomendaciones acerca de la apariencia de los botones, la ubicación de los mismos y formatos de fechas, sin embargo se considero satisfactorios los resultados puesto que el objetivo era constatar que SIDWeb API estaba respondiendo adecuadamente a todas las requisiciones realizadas por el aplicativo móvil.

Mediante el aplicativo SIDWeb Móvil se reduce la cantidad de desplazamientos verticales para visualizar contenidos y se elimina la necesidad de realizar acercamientos para interactuar con los contenidos. El aplicativo brinda un acceso multiplataforma al haber sido probado en dos emuladores de dispositivos móviles y en dispositivos habilitados con el sistema operativo iOS v3.2-5 (iPhone, iPad).

7. Conclusiones

- Se constató la posibilidad de interactuar con aplicaciones externas, como es el caso del aplicativo SIDWeb Móvil que hace uso de los servicios que provee SIDWeb a través del API.
- El uso de canales seguros para comunicación de datos en SIDWeb API es imprescindible dado que la encriptación que estos ayudan a prevenir el robo o acceso no autorizado a información sensible.
- La utilización del Framework Symfony facilitó en gran medida el desarrollo del API, puesto que, aunque hay un tiempo involucrado para el aprendizaje del mismo, esto se ve recompensado al momento de la utilización de características del Framework que disminuyen el tiempo de desarrollo.
- El uso de un formato de transmisión con poca sobrecarga como JSON es importante en interfaces como SIDWeb API dado que el generar contenidos más livianos lo hace más ventajoso considerando factores críticos en aplicaciones móviles como el ancho de banda y la velocidad de procesamiento limitado.

8. Recomendaciones

- El uso de jQuery Mobile el cual fue empleado en el desarrollo de SIDWeb Móvil, dio la posibilidad de realizar pruebas en distintas plataformas realizando el desarrollo en una sola aplicación, por lo que es recomendable considerar el uso de tecnologías multi-plataforma al momento del desarrollo de aplicativos que tienen por objeto ser utilizados en más de una plataforma.
- SIDWeb API intercambia información sensible y protege estos datos mediante el uso de canales seguros, por lo que cualquier aplicación que gestione información sensible debe utilizar mecanismos para proteger tal información.
- La consulta de documentación y de foros relacionados a las tecnologías empleadas en el desarrollo del SIDWeb API y SIDWeb Móvil redujo el tiempo de solución de problemas encontrados, por lo que se recomienda hacer uso de estos recursos al momento de solucionar problemas que se presentan en el desarrollo de sistemas.

9. Agradecimientos

Agradezco a mi familia por siempre haberme brindado el apoyo necesario para conseguir mis metas, al Dr. Enrique Peláez Jarrín y a todo el equipo que conforma el Centro de Tecnologías de Información por el apoyo brindado en el desarrollo de este trabajo.

10. Referencias

[1] VIRPI ROTO – ANTTI OULASVIRTA, Need for Non-Visual Feedback with Long Response Times in Mobile HCI, http:// research.nokia.com/files/MobileFeedback.pdf, fecha de consulta 28 Diciembre 2012, p. 1.

[2] MATT JONES – GARY MARSDEN – NORLIZA MOHD – KEVIN BOONE, Improving Web Interaction on Small Displays, http://www.cs.waikato.ac.nz/oldcontent/mattj/web8 .pdf, fecha de consulta 8 Enero 2012, p. 8.

[3] FABIEN POTENCIER, Practical Symfony Create professional Web Applications with PHP and Symfony 1.3 & 1.4, Doctrine 1.2, Sensio S.A., 2011, p. 34.

[4] FABIEN POTENCIER, Practical Symfony Create Professional Web applications with PHP and Symfony 1.3 & 1.4, Doctrine 1.2, Sensio S.A., 2011, p. 87.

[5] NURZHAN NURSEITOV – MICHAEL PAULSON – RANDALL REYNOLDS – CLEMENTE IZURIETA, Comparison of JSON and XML Data Interchange Formats: A Case Study, http://www.cs.montana.edu/izurieta/pubs/caine2009.pdf, fecha de consulta 5 Febrero 2012.

[6] LEONARD RICHARDSON, RESTful Webservices, O'Reilly, 2007, p.44.

[7] JON REID, jQuery Mobile, O'Reilly, 2011, p. 1.