

# Plan de Trabajo Final

## RestfulRuby: Implementación de un framework para el desarrollo de aplicaciones que cumplan con el standard Restful Objects

Facultad de Ciencias Exactas – UNICEN – Tandil - 20 de Agosto del 2013

Alumno  
**Pablo Daniel Vizcay**

Director  
**Dr. Alejandro Zunino**

### 1. Motivación

En los últimos diez años se produjo un gran avance en el desarrollo de aplicaciones distribuidas que se comuniquen mediante web services lo que ha permitido que las mismas interactúen y se complementen en Internet de forma nunca antes pensada. El protocolo principal elegido como medio transporte, por ser omnipresente en Internet es HTTP, que cuenta con implementaciones robustas, está ampliamente extendido y generalmente no presenta problemas de conexión con la mayoría de los firewalls.

Sin embargo, en vez de utilizar dicho protocolo de mensajes de forma nativa para implementar dichos web services, se lo ha utilizado simplemente como un medio de transporte para otros dos protocolos: SOAP (Simple Object Access Protocol) y WSDL (Web Services Description Language). SOAP es en si un protocolo de comunicación basado en mensajes bastante sencillo que utiliza XML. WSDL es un standard, también basado en XML, de descripción de interfaces que permite especificar la funcionalidad ofrecida por un web service. La simplicidad de SOAP, contrasta con la complejidad que puede implicar las especificaciones generadas WSDL. Excepto en los casos más sencillos, solamente es práctico producir y consumir interfaces WSDL por herramientas para el desarrollo de aplicaciones.

La comunidad tecnológica empezó a desarrollar toda clase de web services con dichas tecnologías, pero la complejidad agregada tenía un costo importante: problemas de compatibilidad por la interoperabilidad de distintos stacks tecnológicos que supuestamente cumplían el mismo standard, problemas para el diseño de las interfaces, gran costo asociado a la depuración de errores de los web services, y grandes problemas de escalabilidad junto a un alto requerimiento en la transferencia de datos (Pautasso, 2008) (Zunino, 2012).

Debido a estos motivos, recientemente se produjo un cambio de eje buscando nuevamente la simplicidad, y la comunidad tecnológica ha incrementado su interés en el desarrollo de aplicaciones consideradas “RESTful” (Fielding, 2000). Una arquitectura RESTful es simplemente un estilo arquitectónico una que adhiere al uso del protocolo HTTP como fue realmente diseñado: basado en recursos, que utilice URIs (Uniform Resource Identifier) para identificarlos, que use los verbos HTTP (Get, Post, Put, Patch y Delete) cumpliendo con la semántica de los mismos y que sea dirigido por el hipertexto generado en las mismas representaciones que se transfieren. Cuando un servicio cumple el último requerimiento, se dice que está basado en HATEOS (Hypermedia as the Engine of Application State) (Richardson, 2007). RESTful no obliga a el uso de una determinada representación para el mensaje en sí, siendo generalmente JSON la elección más habitual (seguramente debido a que es un formato que se consume de forma nativa en aplicaciones JavaScript, de alto auge hoy en día debido a los RIA o Rich Internet Applications que se ejecutan en los browsers).

Utilizar principios RESTful para el desarrollo de web services conlleva además de la simplicidad de las interfaces y la navegación de las mismas por hyperlinks, grandes ventajas en la escalabilidad al poder utilizar el caching nativo de HTTP de forma mucho más agresiva. Pero al ser un estilo arquitectónico y no un protocolo como SOAP+WSDL, deja mucha libertad al desarrollador de como exponer un modelo de objetos que implementen los servicios, que a fin de cuentas es el interés principal de los desarrolladores.

Este vacío es el que intenta llenar Restful Objects (Haywood, 2012), una especificación pública y abierta que adhiere a los principios RESTful, que define un conjunto de rutas genéricas y el formato del JSON para exponer de forma genérica cualquier dominio de modelos de objetos. Sin embargo, al tratarse de un standard medianamente nuevo, ya que la versión 1.0 fue publicada en el 2012, no existe una amplia gama de herramientas para facilitarle al desarrollador la implementación del mismo. En cuestión existen dos frameworks para servidores: “Apache Isis” para Java [6] y “Restful Objects for .NET” [7].

El objetivo de mi trabajo de tesis es ampliar dicho juego de herramientas, desarrollando RestfulRuby, un framework que facilite la implementación del standard en Ruby, un lenguaje muy utilizado en el ambiente web y extremadamente dinámico que viene ganando adeptos de forma sostenida en el tiempo desde su creación (Longworth, 2009).

## 2. Objetivo del trabajo

El objetivo del trabajo final es desarrollar un framework en Ruby que facilite la implementación de servidores que cumplan con el standard RestfulObjects. El mismo debe abstraer los detalles de implementación al programador, tales como el funcionamiento del protocolo sobre HTTP o la generación de JSON, y permitirle concentrarse en el desarrollo del modelo de objetos que implemente la lógica de negocios que es el verdadero interés del desarrollador.

El mismo se desarrollará siguiendo la práctica ágil TDD (Test Driven Development), es decir, primero se escribirán los tests de la funcionalidad a implementar y luego se procederá a escribir el código que cumpla con las especificaciones (Kent Beck, 2002). También el juego de tests se considerara un “asset” más del proyecto y estará incluido siempre con el código fuente de mismo.

La elección de Ruby no es casual, pues su naturaleza dinámica permite modificar las clases y objetos creados por el usuario de forma bastante transparente y declarativa; de esta forma no impone grandes restricciones en el diseño del modelo de objetos lo cual provee mucha libertad. Por ejemplo, un objeto del dominio no necesita ser derivado de una clase del framework sino que la funcionalidad se “incluye” a través de un módulo (técnica conocida como “mixins”). Se utilizarán varias técnicas de meta programación como “class macros”, “dynamics dispatchs”, etc. (Thomas, 2009) (Perrotta, 2010).

El framework se montará sobre Sinatra, una librería para para el desarrollo de aplicaciones web que utiliza Rack como base (Rack provee una interfaz para interactuar con web servers como Apache) [12] [13]. Se utilizará ampliamente la standard library de Ruby, por ej. para la generación de JSON. A la vez, el framewor se publicará como una “gema”, que es el método más utilizado para publicar librerías en Ruby. Los tests se implementaran utilizando RSpec, una de las librerías más utilizadas que provee un DSL (Domain Specific Language) especialmente diseñado para declarar las especificaciones del software [14].

También se comparará con métricas sobre un proyecto de ejemplo, las ventajas de utilizar RestfulRuby, en comparación con los otros implementados con Isis, con base en un lenguaje mucho más utilizado como Java.

### 3. Etapas del Plan de Trabajo

<i>Tarea</i>	<i>Tiempo Estimado</i>
Relevamiento bibliográfico acerca del desarrollo de webservices RESTful	2 semanas
Lectura y comprensión del standard Restful Objects (versión 1.0)	2 semanas
Relevamiento bibliográfico sobre el desarrollo en el lenguaje Ruby y sobre las técnicas de meta-programación dinámicas que permite el mismo	2 semanas
Desarrollo con metodología ágil del framework RestfulRuby	1 mes
Desarrollo de una aplicación demostrativa que utilice el framework	1 semana
Uso de métricas para comparar la efectividad de RestfulObjects contra el framework Isis	1 semana
Redacción del informe	1 mes

### 4. Bibliografía

- [1] Pautasso, Cesare and Zimmermann, Olaf and Leymann, Frank, "Restful web services vs. "big" web services: making the right architectural decision", in Proceedings of the 17th international conference on World Wide Web (New York, NY, USA: ACM, 2008)
- [2] Alejandro Zunino et. al. : "Application note: RESTful Web Services improve the efficiency of data transfer of a whole-farm simulator accessed by Android smartphones", Computers and Electronics in Agriculture, Septiembre 2012
- [3] Roy T. Fielding. Architectural styles and the design of network-based software architectures. PhD Thesis, University of California, Irvine, 2000.
- [4] Leonard Richardson & Sam Ruby: "Restful Web Services", O'Reilly, 1º Edición, Mayo 2007
- [5] Dan Haywood: "Restful Objects Specification", <http://restfulobjects.org/>, version 1.0, 2012
- [6] Apache Isis project homepage, <http://isis.apache.org/>
- [7] Restful Objects for .NET project homepage, <http://restfulobjects.codeplex.com/>
- [8] Longworth, David, "The Rise of Ruby", <http://www.information-age.com/technology/applications-and-development/1072762/the-rise-of-ruby> , 2009
- [9] Kent Beck, "Test Driven Development: By Example", Addison-Wesley, 2002
- [10] Dave Thomas et. al. : "Programming Ruby 1.9: The Pragmatic Programmers' Guide", The Pragmatic Programmers, 3º Edición, 2009
- [11] Paolo Perrotta: "Metaprogramming Ruby: Program Like the Ruby Pros", The Pragmatic Programmers, 1º Edición, 2010
- [12] Rack Webserver interface, <http://rack.github.io/>,
- [13] Apache webserver, <http://httpd.apache.org/>
- [14] Rspec testing suit project, <http://rspec.info/>