

Geo-localización con Ruby on Rails

Edmundo Figueroa Herbas

17 de octubre de 2012

Capítulo 1

Ruby on Rails y patrones Web 2.0

1.1. Porque usar Ruby on Rails para desarrollar una aplicación web ?

La gran propaganda de Ruby on Rails (RoR) o más sencillamente Rails se basa en el rápido desarrollo de aplicaciones web conocido como agile development.

Ruby on Rails maneja las filosofías DRY¹ y convención sobre configuración.

No te repitas(DRY) según el creador de RoR, David Heinemeier Hansson, significa que cada pieza de conocimiento en un sistema debe ser declarado en un solo lugar.[1] Esto se logra gracias al patrón MVC² y el lenguaje multiparadigma Ruby sobre los cuales está construido Ruby on Rails.

Convención sobre configuración significa que Rails tiene parámetros por defecto para casi todos los aspectos que mantiene unida una aplicación, siguiendo las convenciones de Rails se llega a simplificar el código escrito en una aplicación.

¹Don't Repeat Yourself

²Model, View, Controller

David Heinemeier cita en su libro [1], que “*Rails es Ágil porque simplemente la agilidad es parte de su construcción*”.

Los principios del manifiesto por el desarrollo ágil de software:

- **Individuos e interacciones** sobre procesos y herramientas
- **Software funcionando** sobre documentación extensiva
- **Colaboración con el cliente** sobre negociación contractual
- **Respuesta ante el cambio** sobre seguir un plan

Rails se enfoca bastante en conseguir un prototipo funcional en muy poco tiempo y sobre ese prototipo seguir incrementalmente hasta conseguir una aplicación de calidad en poco tiempo.

Los más grandes obstáculos que se enfrenta una aplicación en el tiempo es el mantenimiento y escalabilidad, actualmente se estima que existen 230,000 websites[2] desarrolladas sobre RoR³, entre ellas se puede nombrar a GitHub, Hulu, Yellow Pages. Son sitios con miles de visitas con una alta carga del servidor y son un claro ejemplo de que Rails puede manejar sitios de alto perfil.

Los detractores de Rails sostienen que escalar una aplicación construida sobre RoR es muy difícil pero los defensores argumentan que lo que se tiene que escalar es el código de la aplicación no el framework.

Twitter nació sobre Ruby on Rails y no fue hasta que era un servicio usado a nivel mundial y manejaba millones de request por día que empezaron a surgir problemas debido a que Ruby no estaba optimizado para un trabajo muy pesado, según palabras de Alex Payne, Twitter developer, “Ruby es lento”[3]. Actualmente Twitter migró su backend a Scala, framework basado en Java(que esta mas optimizado que Ruby), pero para su front-end no cambian a Rails.[4]

Se puede agregar que Rails es una muy buena opción a la hora de empezar cualquier proyecto web, ya que implementa las herramientas necesarias para un desarrollo ágil, sólido y de calidad respaldado por un modelo de desarrollo basado en pruebas(TDD), las filosofías DRY y convención sobre

³Ruby on Rails, tambien se lo puede nombrar *Rails*, *RoR*

configuración. y cuando la aplicación haya crecido y empiecen a aparecer muchos problemas es decisión de los programadores el ver si mantener el código actual y parchearlo o cambiar de tecnología para mejorar el rendimiento y la experiencia del usuario

1.2. Patrones de diseño de la Web2.0

Que es la Web2.0 ?, primeramente se debe explicar que este término fue acuñado por 1999 para describir paginas web que usaban tecnologías mas alla de las simples estaticas paginas web.

No fue hasta que en el 2004 en la conferencian sobre la Web2.0 que popularizo este termino, y asi mismo como la Web que evoluciona, la definicion se actualiza con el tiempo, pero Tim O'Reilly trato de definirla en su articulo "What is Web 2.0"[5], del que se puede extraer la siguiente definición :

"Web 2.0 is the business revolution in the computer industry caused by the move to the Internet as a platform, and an attempt to understand the rules for success on that new platform. Chief among those rules is this: build applications that harness network effects to get better the more people use them."

—Tim O'Reilly [9]

En resumen se puede definir que una aplicación web2.0 es aquella que mejora y crece con la participación activa de sus usuarios.

"Software que mejora mientras más gente la usa" [5]

Un patrón de diseño es una solución general, reusable y flexible que describe cómo resolver algún problema general en el desarrollo de software, un patrón puede ser usado y modificado segun el problema al cual se esta aplicando.

Se pueden observar los siguientes patrones de diseño en la aplicación:

- REST
- MVC
- Mashup

Bibliografía

- [1] Sam Ruby, Dave Thomas, David Heinemeier Hansson, *Agile Web Development with Rails, Fourth Edition*
- [2] <http://trends.builtwith.com/topsites/Ruby-on-Rails>
- [3] tumblr.yasulab.jp/post/10271634919/5-question-interview-with-twitter-developer-alex-payne
- [4] http://www.artima.com/scalazine/articles/twitter_on_scala.html
- [5] <http://oreilly.com/web2/archive/what-is-web-20.html>
- [6] <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [7] <http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>
- [8] <http://msdn.microsoft.com/en-us/architecture/bb906060.aspx>
- [9] <http://radar.oreilly.com/2006/12/web-20-compact-definition-tryi.html>