



aprenderaprogramar.com

¿Qué es Ruby on Rails? Un framework de desarrollos web software libre (DV00201A)

Sección: Divulgación

Categoría: Lenguajes y entornos

Fecha revisión: 2029

Autor: Mario Rodríguez Rancel

Resumen: Este artículo explica qué es Ruby on Rails: un framework para desarrollo web que está siendo utilizado por muchos programadores. Resume y comenta una conferencia impartida por Gumersindo Coronel.

¿QUÉ ES RUBY ON RAILS?

Ruby on Rails es una plataforma de trabajo para realizar desarrollos web. Podemos definirlo como un framework de software libre. Ruby hace referencia al lenguaje de programación que se utiliza: todo se escribe en Ruby. Por otro lado, on Rails (sobre raíles) indica que el framework "nos va guiando" para hacer fáciles nuestros desarrollos.



Por framework entendemos un entorno o conjunto de programas o herramientas agrupadas. Por software libre entendemos software que cualquier usuario puede ejecutar, copiar, modificar, distribuir, cambiar o mejorar.

Ruby es un lenguaje de alto nivel, es decir, de sintaxis más aproximada a la forma de expresarnos de los humanos que otros lenguajes. Por ejemplo el método "pluralize" nos permite obtener el plural de una palabra y el método "singularize" obtener el singular. Lo gracioso es que el lenguaje es casi humano: pluralize "cat" nos devuelve "cats" pero pluralize "person" nos devuelve "people", que es el plural en inglés de person. El número de usuarios de Ruby está yendo en aumento en los últimos años; sus usuarios lo defienden dando varios argumentos entre los que podríamos destacar que "funciona de forma eficiente" y por otra lado valoran mucho la sencillez de su sintaxis, que se aproxima a lo que podría ser el inglés escrito, un lenguaje "casi natural". Hemos dicho que todo lo que se escribe en Ruby on Rails se escribe en Ruby, luego el framework se encarga de transformarlo en otros lenguajes en función de las necesidades: por ejemplo, en javascript para mostrar un formulario en una página web o en SQL para realizar comunicaciones con una base de datos. El hecho de manejar un solo lenguaje hace que Ruby on Rails resulte sencillo de utilizar. Como contrapartida, la transformación desde Ruby on Rails a otros lenguajes da lugar a que el código que subyace (digamos el código que existe detrás de nuestro desarrollo web) no va a ser tan limpio y depurado como si hubiéramos utilizado directamente los lenguajes correspondientes.

Aunque Ruby es multiplataforma, es decir, se puede ejecutar bajo distintos sistemas operativos, es bajo Unix como se consigue un mejor rendimiento. Ruby on Rails debemos correrlo en Unix, en otras plataformas, por ejemplo Windows, su uso resulta problemático.

Como es propio en el software libre, hay una comunidad de programadores trabajando en Ruby on Rails. En España existe una comunidad bastante activa.

Ruby on Rails evoluciona al ir apareciendo sucesivamente nuevas versiones. Las migraciones (cambio de versión) con las últimas versiones pueden calificarse de sencillas y también reversibles, es decir, si después de hacer una migración se comprueba que existen problemas, se puede revertir el proceso con pocas complicaciones. Esto no significa que no puedan aparecer problemas, sino que con versiones antiguas estos procesos eran mucho más complicados.

NOCIONES SOBRE ASPECTOS TÉCNICOS DE RUBY ON RAILS

Ruby on Rails se basa en el MVC (Modelo – Vista – Controlador). De forma simplificada, diremos que existe:

- a) Un elemento Action View, encargado de controlar las vistas: aquello que se muestra al usuario o se envía. Action View se encarga de “pintar” los datos adecuadamente: html para navegadores, xml para servicios web o wml para teléfonos móviles.
- b) Otro elemento, Active Record, encargado de gestionar el modelo de datos.
- c) Otro elemento, Action Controller, que se encarga de “dirigir” o comunicar modelo y vista.

Un proyecto de Ruby on Rails se organiza en carpetas. Orientativamente vamos a tener al menos las siguientes carpetas:

- a) app: carpeta donde va a ir prácticamente todo el código del proyecto.
- b) config: carpeta donde especificamos aspectos como la configuración de la base de datos entre otros aspectos de configuración.
- c) public: carpeta destinada a contener ficheros javascript y hojas de estilo (CSS) entre otros.
- d) scripts: carpeta con varias herramientas para creación de contenidos.

Ruby on Rails utiliza como elementos fundamentales plugins: pequeños fragmentos de código que permiten realizar tareas muy concretas. Por ejemplo hay un plugin que permite que un usuario se registre y que cuando se registre se le envíe un e-mail. Los desarrolladores de Ruby on Rails tienen su propia colección de plugins y sus desarrollos se basan principalmente en ellos. La comunidad de programadores publica los plugins en el sitio web “GitHub Social Coding” (github.com) donde quedan a disposición del resto de usuarios. ¿Para qué estar generando código que ya está generado? Por tanto, para aquellas personas que empiecen con Ruby on Rails, es muy recomendable que dediquen sus primeros días a explorar algunos de los plugins disponibles, su implementación y funcionamiento: encontrarán más plugins de los que van a tener tiempo a revisar o probar. Hay que tener en cuenta que los desarrolladores profesionales basan sus proyectos en aproximadamente un 90 % de código en plugins y un 10 % de código propio. Por algo será.

Los plugins están desarrollados para versiones concretas de Ruby on Rails, por tanto no tenemos garantías de que vayan a funcionar cuando realizamos una migración. Esto es uno de los aspectos “problemáticos” de Ruby on Rails. Como siempre, procede ser cautelosos, y mantener copias de seguridad que eviten pérdidas de información si aparecen problemas.

Ruby on Rails puede generar aplicaciones que usen servidores Apache. Se suelen usar bases de datos MySQL, pero también se pueden usar otras como Oracle. Podemos reseñar que trabajando bajo este framework podemos, en principio, cambiar de base de datos sin tener que hacer nada: por ejemplo, pasarnos de MySQL a Oracle sin mayores problemas ya que el framework se encarga automáticamente de las adaptaciones necesarias. Cuando creamos una aplicación con Ruby on Rails es muy sencillo hacer tests para comprobar que la aplicación responde tal y como esperamos antes de poner la aplicación en uso. El framework ejecuta un test y nos devuelve un informe. Es muy recomendable ir haciendo pruebas conforme se va realizando el desarrollo con Ruby on Rails, ya que no existe compilación como en otros lenguajes. Por tanto, testear aunque parezca una pérdida de tiempo en realidad va a evitarnos tener que hacer tediosas correcciones por fallos en el comienzo del desarrollo de un proyecto.

REFERENCIAS Y MÁS INFORMACIÓN

Este artículo resume y comenta la conferencia pública impartida por Gumersindo Coronel, Ingeniero Informático de la empresa Emergya, en el marco de las "Jornadas Imaginática: La informática del futuro", que tuvieron lugar en la Escuela Técnica Superior de Informática de la Universidad de Sevilla (España) y a las que tuvimos la oportunidad de asistir. Puedes ampliar información sobre la empresa en emergya.es.

Referencias en la web:

Para Ruby on Rails: <http://rubyonrails.org>

Para Ruby: www.ruby-lang.org

Para desarrolladores: www.github.com; página con código, proyectos y recursos colaborativos.

Libros:

Programming Ruby. Dave Thomas, Chad Fowler y Andy Hunt.

Agile web development with rails. Dave Thomas y David Heinemeier.