Table of Contents

Introducción 1.1

Acerca del autor 1.2

Acerca de los editores 1.3

Acerca de este libro 1.4

Primeros pasos 1.5

- Requerimientos del sistema 1.5.1
- Configurando el proyecto 1.5.2
- Estructura de los directorios del proyecto 1.5.3

Introducción

Ruby on Rails es un *framework* que ofrece un ecosistema completo para la construcción de aplicaciones Web, brindando tecnologías de *Backend* en el lenguaje *Ruby*, tareas para: el procesamiento de los *assets*, manejo de conexión a base de datos, *scripts* y demas; Tecnologías de *Frontend* como *HTML*, *CSS* y *JavaScript*, pero a su vez brindado la posibilidad de trabajar con *frameworks* de *FrontEnd* como *Bootstrap*, *frameworks* de *JavaScript* como *Angular JS* o *Ampersand JS* y librerias de *JavaScript* como *jQuery* o *Backbone*, por solo mostrar unos ejemplos; Permitiendo la incorporación de gemas que amplian su funcionalidad dejando así al programador un *Stack* completo para desarrollar aplicaciones completas o sencillamente un API.

Posee una característica muy importante a resaltar, es la facilidad que ofrece a los programadores de construir aplicaciones de forma rápida con unos sencillos comandos y a su vez incorporando las mejores prácticas de programación como *DRY* y amoldándose a metodologías de desarrollo ágil como *SCRUM*. Pero tambien es importante resaltar que fácil no significa simple, ya que por cada comando que *Rails* ejecuta este realiza una gran cantidad funciones y llamados internamente.

Se basa en el patrón de diseño: Modelo Vista Controlador (MVC), garantizando así una independencia de los diferentes dominios. Viene preparado para trabajar con sistemas de control de versiones como *git* facilitando el versionamiento del código fuente y permitiendo una manera sencilla de trabajar en equipo.

El objeto *Active Record* de *Rails* es una representación de *ORM* para el manejo de la persistencia de datos e independiza el origen de los datos pudiendo así utilizar base de datos relacionales como *PostgreSQL* o *MySQL*, y con la inclusión de gemas base de datos no relacionales como *MongoDB*.

Su popularidad se ha incrementado durante estos años y ha generado la creación de plataformas y servicios alrededor de él como: la plataforma de publicación de aplicaciones *Heroku* o servicios de integración continua como *Travis CI*.

Abrumador el panorama hasta ahora, cierto?, pero no te preocupes las tecnologías mencionadas anteriormente son solo un ejemplo de la gran variedad del ecosistema de *Ruby on Rails*, pero muchas de ellas se tocarán superficialmente en este libro, porque recuerda que es una introducción a *Ruby on Rails*, bienvenido y relájate espero que disfrutes este libro tanto como yo lo hice escribiéndolo.

Acerca del autor

Soy un apasionado de la programación, creo profundamente que la programación es un arte que permite crear desde pequeñas obras hasta grandes y complejas representaciones de soluciones, ademas con la posibilidad de compartirlas con todo el mundo y permitir a su vez entenderlas, modificarlas e inclusive construir a partir de ellas. Creo que la mejor forma de aprender y enseñar es compartiendo, por ello mi segunda pasión es la enseñanza.

Soy entusiasta de la Web desde el momento que comencé a crear páginas desde el año 1999 con editores de texto como Notepad y ver su resultado inmediato en el navegador Web como Internet "Exploiter" 5.0 (leasé bien), pasando por editores gráficos como Macromedia Dreamweaver 3 (en esa época pertenecía a la empresa Macromedia), navegadores como Netscape a editores de texto avanzados como Sublime Text o Atom y navegadores modernos como Google Chrome, teniendo una gran cantidad de herramientas apoyadas por la terminal y el poder de *nix.

Soy Ingeniero de Sistemas y Magister en Gobierno de TI de la Universidad del Norte, allí tambien profesor de pregrado con más 10 años de experiencia en temas relacionados con el desarrollo Web y en posgrado en los temas relacionados con la Ingeniería de Software. Actualmente estoy vinculado a una agencia internacional de desarrollo Web que me permite trabajar remotamente desde la comodidad de mi hogar.

Soy esposo y padre de familia, a ellos les agradezco el apoyo y la paciencia de las horas invertidas frente al computador para la creación de este libro, vivo en la ciudad de Barranquilla en Colombia, disfruto cada día de la calidez de su gente y su maravilloso clima.

> Gustavo Jose Morales Carpio

Porque escribí este libro

Una de las competencias fundamentales para las personas que están en el mundo de la informática es el autoaprendizaje, ya que día a día surgen nuevas tecnologías, paradigmas y formas de solucionar los problemas, por ello es indispensable nunca dejar de aprender. Aprender es una cualidad de todos los seres humanos, pero a su vez todos aprendemos de maneras diferentes, por lo cual enseñar se convierte en un reto y como lo mencione anteriormente la mejor manera de aprender y enseñar es compartir, debido a esto quiero plasmar en este libro toda mi experiencia como profesor y usuario que aprendió *Ruby* y *Rails* por sus propios medios y viniendo de otros lenguajes como *PHP*.

Sugerencias y comentarios

La mejor forma de construir es en comunidad, anexo mis datos de contacto:

- Web: http://gmoralesc.me
- Correo electrónico: gustavo.morales@gmail.com
- Twitter: @gmoralesc

Acerca de los editores

Acerca de este libro

A quien esta dirigido y que espero de este libro

Este libro esta dirigido a personas que desean introducirse en el mundo de la programación Web, con conocimientos básicos de programación e inclusive con experiencia en otros lenguajes de programación y/o *frameworks* que quieran conocer que ofrece *Ruby on Rails*.

El nivel de este libro es para principiantes pero ofrece un recorrido por los aspectos fundamentales de *Ruby on Rails*.

Como leer este libro

La metodología empleada es la introducción de conceptos a medida que se desarrolla el proyecto de ejemplo, por lo cual es recomendado leer en el orden que se presenta.

Que debo saber para leer este libro

Se asume que el lector tiene conocimientos básicos de programación en: lenguaje *HTML*, lenguaje *CSS* y *JavaScript* básico. No es necesario conocer el lenguaje *Ruby* pero se recomienda conocer al menos la sintaxis básica del lenguaje. Si desea reforzar estos temas puede dirigirse a los siguientes enlaces:

- HTML y CSS
- JavaScript
- Ruby

Que necesito para desarrollar los ejemplos de este libro

Para una mejor experiencia se recomienda utilizar un sistema basado en Mac OS o en su defecto Linux, es posible realizarlo en un sistema Windows pero no es recomendable debido a temas relacionados con los permisos e instalación de algunas librerias especificas. Tambien se pueden utilizar entornos de programación integrados en la nube como Cloud9 por citar un ejemplo.

Que NO incluye este libro

No incluye temas intermedios como el *Behaviour Driven Development* (BDD) (siguiente libro de esta serie) o temas avanzados como incorporación de *frameworks* de *JavaScript* o creación de un *API* (otro libro de esta serie), pues recuerda que solo es una introducción. Tampoco se incluirá la utilización de *git* en el desarrollo de la aplicación que es casí mandatorio en el desarrollo de aplicaiones como esta y muy comúnmente usado, ya que el objetivo de este libro es focalizarse en la introducción de los aspectos básicos y fundamentales de *Rails*.

Convenciones

Los terminos con *enfasis* no son traducidos de su idioma nativo pero su descripción se encuentra en el glosario:

frameworks

Las notas se utilizan para dar información adicional:

> Es posible que este proceso tarde un poco dependiendo de la velocidad de la maquina y la conexión a Internet.

Los comandos que se ejecutan en la consola están identificados por que se preceden del signo $ (el cual no esta incluido en el comando):

```
$ rails --version
```

El código fuente de los archivos se especifica de la siguiente forma:

```
<div>
  <%= yield %>
</div>
```

Requerimientos del sistema

Ruby Version Manager (RVM)

RVM es un administrador de versiones del lenguaje *Ruby*, el cual permite instalar diferentes versiones del lenguaje en la misma maquina sin que entren en conflicto, brindado la capacidad de pasar de una versión a otra dependiendo el proyecto en el cual se este trabajando. Adicionalmente permite crear conjuntos independientes de gemas (*gemsets*) para no combinar diferentes de versiones de gemas cuando se trabaje en diferentes proyectos.

Para instalar la versión estable de *RVM*:

```
$ \curl -sSL https://get.rvm.io | bash -s stable
```

> Este comando puede ejecutarse desde cualquier directorio donde se encuentre ubicado el usuario en la terminal, por defecto *rvm* se instala en `~/.rvm`

Para comprobar la versión:

```
$ rvm --version
```

> Nota: Si el comando anterior no es reconocido se debe reiniciar la terminal

Para conocer todo lo que se encuentra disponible para instalar con *RVM*:

```
$ rvm list known
```

Si se presenta algún problema en la instalación en la Web oficial de *RVM* encontrará mas información:

- RVM

Instalar Ruby

Una vez instalado *RVM* se puede instalar la versión especifica que se desee de *Ruby*, trabajaremos la versión 2.2.4:

```
$ rvm install 2.2.4
```

> Este proceso puede demorar dependiendo de la velocidad de la maquina y conexión a Internet

Una vez el proceso ha finalizado, comprobamos la versión de *Ruby*:

```
$ ruby --version
```

Para listar todas las versiones del lenguaje *Ruby* instaladas en el sistema:

```
$ rvm list
```

Se puede establecer una versión del lenguaje *Ruby* por defecto cada vez que se abre una nueva sesión en la terminal, en este caso para la versión 2.2.4:

```
$ rvm use 2.2.4 --default
```

Para conocer mas acerca del lenguaje *Ruby*, versiones y demas información:

- Ruby

RubyGems

Las gemas son el mayor atractivo de *Ruby*, podemos decir que son librerias que agregan funcionalidad al entorno del lenguaje *Ruby*, inclusive *Rails* es una gema de *Ruby* de allí su nombre *Ruby on Rails*.

RubyGems (el comando gem en la terminal) es la utilidad de *Ruby* que permite instalar dichas gemas en el sistema, esta utilidad se instala junto con el lenguaje *Ruby* en el paso anterior.

Para comprobar la versión de *RubyGems*:

```
$ gem --version
```

Se recomienda que cada vez que se inicie un nuevo proyecto se cree un *gemset* correspondiente para que las gemas queden instaladas dentro de ese *gemset* y no interfieran con las otras instaladas por otros proyectos o del sistema. En consecuencia cada

vez que vamos a trabajar en nuevo proyecto o uno existente debemos indicarle a *rvm* que seleccione el *gemset* asociado a este, afortunadamente esta tarea se puede automatizar evitando posibles errores al no seleccionar el *gemset* correspondiente.

Por defecto nn la terminal no se muestra cual es la versión del lenguaje y el *gemset* seleccionado para trabajar, pero podemos ver el listado con:

```
$ rvm gemset list
```

```
gemsets for ruby-2.2.4 (found in /Users/[username]/.rvm/gems/ruby-2.2.4)
=> (default)
   global
```

En la salida del comando anterior podemos observar que la versión del lenguaje que se esta trabajando es la `2.2.4` y el *gemset* seleccionado por defecto es `default`.

> El `[username]` corresponde al nombre de usuario del sistema y la ruta puede variar dependiendo del sistema operativo.

Para comprobar el listado de gemas que existen en un *gemset* utilizamos:

```
gem list
```

Para buscar gemas y versiones especificas disponibles puede consultar la página oficial:

- RubyGems

Instalar Rails

Antes de instalar la gema de *Rails* creamos un *gemset* para tener agrupadas todo el conjunto de gemas relacionadas con *Rails*.

Para crear un conjunto de gemas vacío con un nombre descriptivo para su contenido:

```
$ rvm gemset create rails-4.2.6
```

Para establecer cual es la versión del lenguaje y el conjunto de gemas a utilizar:

```
$ rvm use 2.2.4@rails-4.2.6
```

> Es posible utilizar la bandera `--default` para establecer el conjunto de gemas por defecto que se utilizará cada vez que se inicie una nueva sesión en la terminal.

Para instalar la versión especifica de *Rails*:

```
$ gem install rails --version=4.2.6 --no-ri --no-rdoc
```

> Las banderas `--no-ri --no-rdoc` no son obligatorias, pero al utilizarlas omiten la instalación de la documentación local de Rails ya que normalmente es mas utilizada la versión en linea y ahorra una gran cantidad de tiempo en la descarga e instalación de la misma.

Para comprobar la versión de *Rails* instalada:

```
$ rails --version
```

Para conocer mas acerca del *framework* de *Rails*, que versiones están disponibles:

- Ruby on Rails

Instalar Node Version Manager (NVM)

Rails necesita un interpretador de *JavaScript* y uno de los mas recomendados es *Node JS*.

Asi como *RVM* administra las versiones de *Ruby*, *NVM* administra las versiones instaladas de *Node JS* en el sistema.

Para instalar la versión estable de *NVM*:

```
$ curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.31.0/install.sh | bash
```

Para comprobar la instalación de *NVM*:

```
$ nvm --version
```

Si se presenta algún problema en la instalación en la Web oficial de *NVM* encontrará mas información:

- NVM

Instalar Node JS

Para instalar la versión estable de *Node JS*

```
$ nvm install stable
```

Para comprobar la instalación de *Node JS*:

```
$ node --version
```

Si desea conocer más acerca del proyecto de *NodeJS* puede visitar su página oficial:

- Node JS

Configurando el proyecto

Creando el *gemset*

Nuestro proyecto se llamará *photobook*, por ende creamos un *gemset* con el mismo nombre:

```
$ rvm gemset create photobook
```

> Para ver todas las acciones que se pueden ejecutar para los *gemset*: `rvm gemset -h`

Es recomendable verificar que versión del lenguaje y que *gemset* esta seleccionado antes de comenzar a trabajar:

```
$ rvm gemset list
```

Seleccionamos la versión del lenguaje y *gemset* correspondiente:

```
$ rvm use 2.2.4@photobook
```

En este caso `2.2.4` es la versión del lenguaje y `photobook` es el nombre del *gemset*.

Podemos aprovechar que tenemos instalado *Rails* y todas las demas gemas necesarias en otro *gemset* (rails-4.2.6) y copiarlas a este nuevo *gemset* (photobook):

```
$ rvm gemset copy 2.2.4@rails-4.2.6 2.2.4@photobook
```

Crando el proyecto de *Rails*

Cada proyecto generado con *Rails* no necesita estar dentro de ningún directorio particular, pues el internamente tiene una gema (WEBRick) que realiza la función de servidor Web, por lo tanto podemos crear nuestro proyecto en cualquier directorio, para efectos prácticos de este libro, seleccionaremos:

```
$ cd ~/Web
```

> La virgulilla ~ en los sistemas *nix* significa el directorio del usuario actualmente autenticando.

Creamos proyecto con el comando:

```
$ rails new photobook
```

La opción `new` permite crear un nuevo proyecto, en este caso de nombre `photobook`, pero adicionalmente se le pueden especificar muchas opciones como por ejemplo la base de datos: `--database=postgresql`

Despues de crear la estructura de directorios correspondientes el sistema ejecuta el comando `bundle install`, `bundle` es una gema que lee el archivo `Gemfile` ubicado en la raíz del directorio e instala todas las gemas y sus dependencias especificadas en él.

> Es posible no ejecutar este paso agregando la bandera `--no-bundle` al comando de creación de proyecto.

Adicionalmente es posible optimizar el inicio del proyecto con el siguiente comando:

```
$ gem pristine --all
`
```

Ingresamos al que llamaremos de ahora en adelante el directorio raiz del proyecto:

```
$ cd ~/Web/photobook
```

Antes habiamos mencionado que el proceso de seleccionar la versión de lenguaje y el *gemset* respectivo para el proyecto se puede automatizar, el siguiente comando crear un archivo en la raiz de nuestro proyecto indicandole a *rvm* que seleccionar cada vez que se ingrese a ese directorio:

```
$ echo "2.2.4@photobook" >> .ruby-version
```

Para probar el proyecto iniciamos el servidor Web que trae por defecto con el siguiente comando:

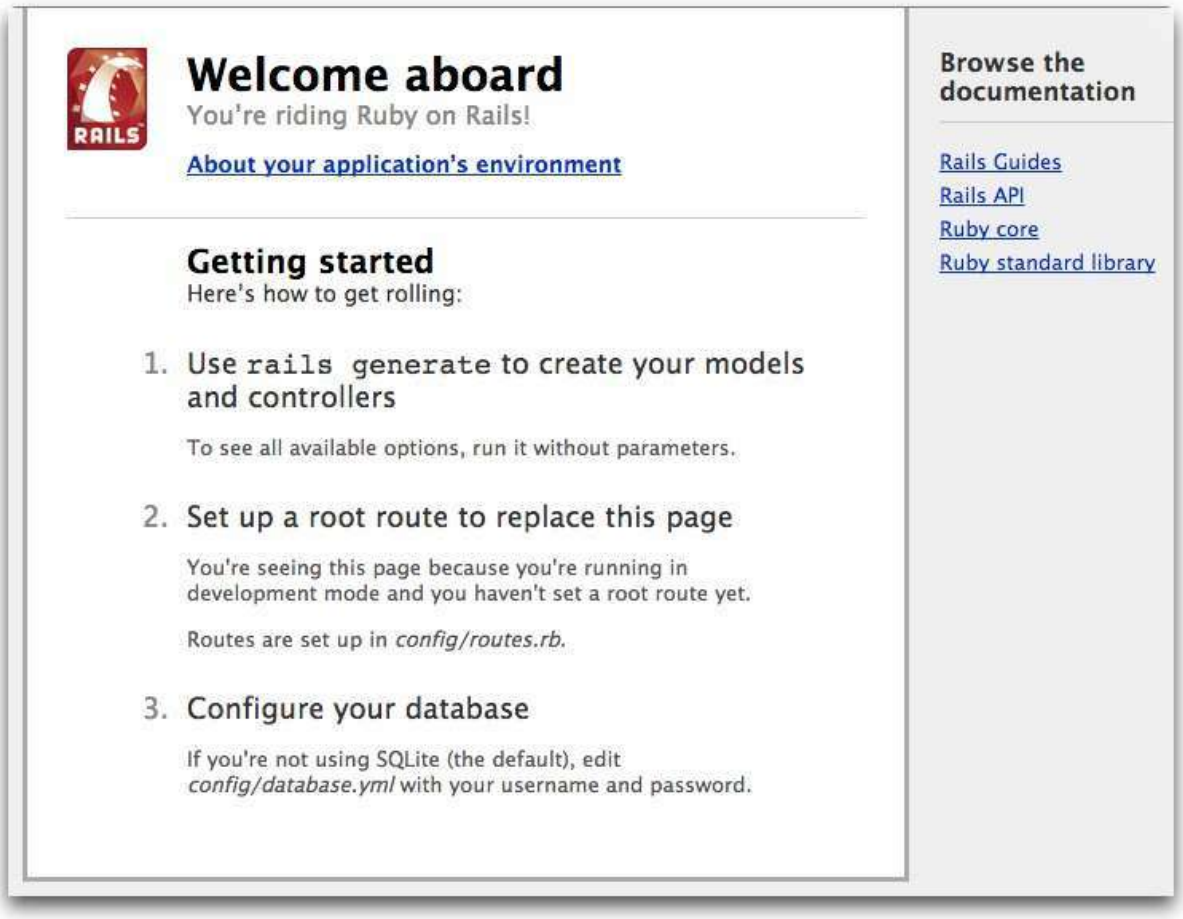
```
$ rails server
```

> Para detener el servidor se presiona la combinación de teclas CTRL+C en la terminal.

Iniciamos un navegador Web e ingresamos la siguiente dirección:

```
localhost:3000
```

Deberia salir la siguiente imagen:



Estructura de los directorios del proyecto

El directorio tiene un numero de ficheros y carpetas auto-generados que constituyen la estructura de una aplicación *Rails*. La mayor parte del trabajo en este libro se hará en la carpeta *app*, pero aqui esta la relación básica entre cada fichero/carpeta y la función que cumple:

Fichero/Carpeta	Propósito
app/	Contiene los controladores, modelos, vistas, funciones de ayuda, enviadores de correo, y ficheros assets fuente (sin compilar) para tu aplicación.
bin/	Contiene los scripts rails para comenzar tu aplicación, y pude contener otros scripts utilizados para configurar, desplegar y ejecutar tu aplicación.
config/	Configurar tu apliación, rutas, base de datos, y más.
db/	Contiene tu actual esquema de datos, como también las migraciones de la base de datos.
Gemfile Gemfile.lock	Estos ficheros te permiten especificar las gemas que necesitas y la dependencias de tu aplicación Rails. Estos ficheros son utilizados por la gema Bundler.
lib/	Modulos extendidos para tu aplicación.
log/	Ficheros de log de la aplicación.
public/	La única carpeta que es vista por el mundo exterior tal cual es. Contiene los ficheros estaticos y los ficheros assets compilados.
Rakefile	Este fichero contiene y graba las tareas que pueden ser ejecutadas a través de una linea de comando. Las definiciones de tareas son utilizadas a lo largo de todos de los componentes de Rails. Mejor que cambiar el Rakefile, tu deberías crear tu propias tareas añadiendo ficheros en la carpeta lib/tasks de tu aplicación.
README.rdoc	Este es un breve manual de instrucciones de tu apliación. Tu puedes editar este fichero para decirle a otros que hace tu aplicación, como configurala, etc.
test/	Test unitarios, ficheros con datos de prueba y otros tipos de test.
tmp/	Ficheros temporales (como los de cache e idenficadores de procesos pid).
vendor/	Un lugar por todo el código de terceros. En una aplicación rails típica esto incluye gemas.

> El texto anterior ha sido tomado y modificado de las guías oficiales de *Ruby on Rails*
> http://guiasrails.es/getting_started.html