

Instalation Notes

Sistema operativo requerido: Ubuntu Linux

Preparando su sistema

Nota: No debe instalar Ruby desde un paquete predeterminado, si bien Ubuntu proporciona un sistema de administrador de paquetes para instalar el software del sistema, este solo se usa para preparar su sistema antes de instalar Ruby, no debe instalar Ruby con este administrador de paquetes ya que se encuentra desactualizado.

Tendrá que preparar su computadora con el software del sistema requerido antes de instalar Ruby on Rails.

Necesitará acceso de superusuario (root) para actualizar el software del sistema.

Actualice su administrador de paquetes primero:

```
$ sudo apt-get update
```

Esto debe finalizar sin error o el siguiente paso fallará.

Instalar Curl :

```
$ sudo apt-get install curl
```

Utilizarás Curl para instalar RVM .

Instalar Ruby usando RVM

Use RVM , para instalar Ruby y administrar sus versiones de Rails.

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB
```

```
$ \curl -L https://get.rvm.io | bash -s estable --ruby
```

Tenga en cuenta la barra invertida antes de "curl" (esto evita posibles conflictos de versión).

Para activar rvm y comenzar a utilizar sus comandos debe correrlo primero

```
Source /usr/share/rvm/scripts/rvm
```

Si ya tiene RVM instalado, actualícelo a la última versión e instale Ruby:

```
$ rvm get stable --autolibs = enable
```

```
$ rvm install "ruby-2.4.2"
```

```
$ rvm --default use ruby-2.4.2
```

Instalar Node.js

El tiempo de ejecución de JavaScript es necesario para compilar el código para la canalización de activos de Rails. Para el desarrollo en Ubuntu Linux, es mejor instalar el entorno de JavaScript del lado del servidor Node.js.

```
$ sudo apt-get install nodejs
```

Compruebe el Gem Manager

Verifique la versión del administrador de gemas instalado.

```
$ gem -v
```

```
2.4.2
```

En el momento en que esto fue escrito, una versión más nueva de RubyGems estaba disponible. Úselo `gem update --system` para actualizar el administrador Ruby gem:

```
$ gem update --system
```

RVM Gemsets

Muestra una lista de gemas:

```
$ rvm gemset list
```

```
gemsets for ruby-2.4.2
```

```
=> (default)
```

```
global
```

Solo las gemas "predeterminadas" y "globales" están preinstaladas.

Si obtiene un error "rvm is not a function", cierre la consola y ábrala nuevamente.

Global Gemset de RVM

Vea qué gemas están instaladas en el gemset "global":

```
$ rvm gemset use global
```

```
$ gem list
```

Varias gemas se instalan con Ruby o el gemset predeterminado de RVM.

Para obtener una lista de gemas que están desactualizadas:

```
$ gem outdated
```

```
### list not shown for brevity
```

Para actualizar todas las gemas antiguas:

```
$ gem update
```

```
### list not shown for brevity
```

En particular, rake debería actualizarse a la versión 10.2.1 o posterior.

Instalación de gemas más rápida

A continuación, le mostramos cómo acelerar la instalación de gemas al deshabilitar el paso de documentación:

```
$ echo "gem: --no-document" >> ~/.gemrc
```

Instalar Bundler

La gema Bundler es una herramienta esencial para administrar gemas al desarrollar y ejecutar aplicaciones Rails. RVM solía instalar Bundler automáticamente. A partir de la versión 1.26.11 (publicada el 31 de marzo de 2015), RVM ya no instala Bundler automáticamente. Debe instalar Bundler:

```
$ gem install bundler
```

Nokogiri

Cada vez que instales la gema Nokogiri, esperarás (hasta cinco minutos).

Para ahorrar tiempo, instale la gema Nokogiri en el gemset global RVM :

```
$ gem install nokogiri
```

Durante la instalación, Nokogiri mostrará dos mensajes largos en la consola. También se detendrá sin mostrar ningún progreso durante cinco minutos. No suponga que la instalación ha fallado a menos que vea un mensaje de error o que haya esperado más de diez minutos.

Opciones de instalación de Rails

Verifique la versión actual de Rails . Rails 5.1 era actual y Rails 5.2 estaba disponible en versión beta cuando se escribió esto.

Recomiendo instalar la versión beta:

```
$ gem install rails --pre
```

Si instala Rails en este punto, lo instalará en el gemset global.

Recomiendo hacer un gemset para la versión beta:

```
$ rvm use ruby-2.4.2@rails5.2 --create
```

Crear una carpeta de espacio de trabajo

Necesitará una carpeta conveniente para almacenar sus proyectos de Rails. La llamaremos **workspace** / .

Crea una carpeta de proyectos y muévete a la carpeta:

```
$ mkdir workspace
```

```
$ cd workspace
```

Aquí es donde crearás tus aplicaciones de Rails.

Nueva aplicación Rails – Migración de PmbokProjects

Para migrar nuestra aplicación hecha en Ruby on Rails se debe hacer una clonación desde el repositorio de Github. Para ello lo primero que debemos hacer es crear una llave SSH publica con los siguientes comandos:

```
$ ssh-keygen -t rsa
```

Solicitará generar una contraseña, no olvidarla.

Luego para poder acceder a la clonación del proyecto se debe copiar el contenido de la llave publica y enviarla a r.villablancafernand@uandresbello.edu para que pueda ser agregada como una llave confiable en el repositorio de GitHub.

Para acceder al contenido de esta llave se puede hacer con el siguiente comando:

```
$ more .ssh/id_rsa.pub
```

Una vez obtenido el permiso se debe utilizar el siguiente comando para clonar el proyecto en el workspace (Se debe estar posicionado en el workspace)

```
$ git clone git@github.com:rvillablancafernand/pmbokprojects.git
```

Solicitará la clave generada en los puntos anteriores.

Agregar el Gemset

Creamos el Gemset para el proyecto, para esto debemos estar ubicados en él.

```
$ cd pmbokprojects
```

```
$ rvm use ruby-2.4.2@pmbokprojects --ruby-version --create
```

Deberá instalar Rails y las gemas enumeradas en su Gemfile en el nuevo gemset ejecutando:

```
$ gem install rails
```

```
$ sudo apt-get install libpq-dev
```

```
$ gem install pg -v '0.21.00'
```

```
$ bundle install
```

```
$ echo "ruby-2.4.2"> .ruby-versión
```

```
$ echo "pmbokprojects"> .ruby-gemset
```

Migración de Base de Datos y Seeds

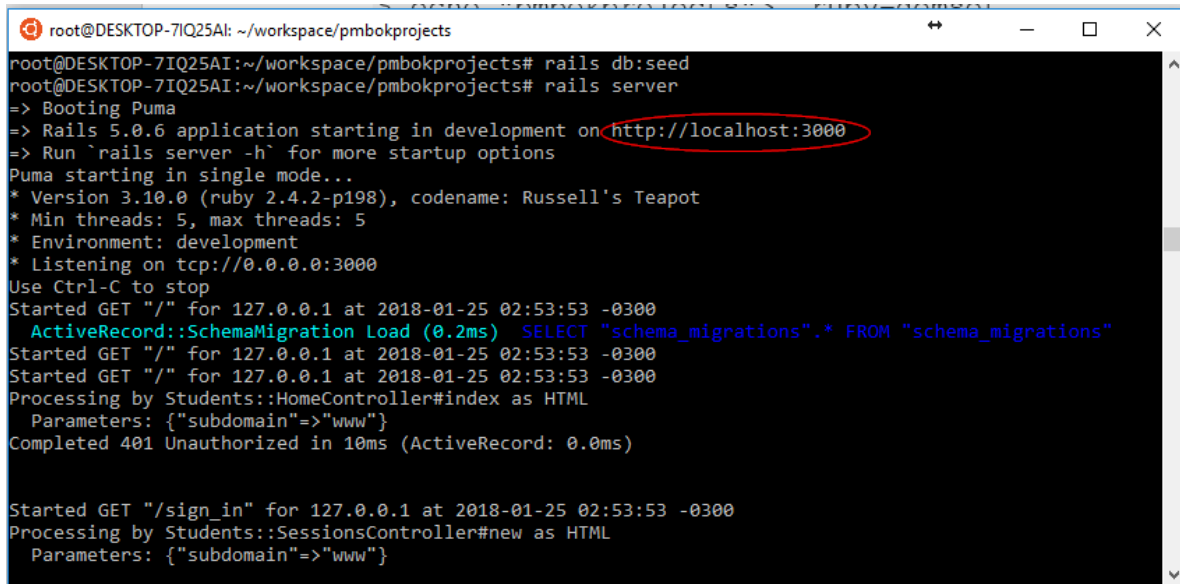
Una vez realizados los pasos anteriores corremos los siguientes comandos para poblar la base de datos y hacer correr el servidor.

```
$ rails db:migrate
```

```
$ rails db:seed
```

```
$ rails server
```

Una vez corriendo el servidor la aplicación puede ser revisada desde un navegador con la dirección que se encuentra encerrada con color rojo en la siguiente imagen:

A terminal window titled 'root@DESKTOP-7IQ25AI: ~/workspace/pmbokprojects' showing the execution of 'rails db:seed' and 'rails server'. The server starts on 'http://localhost:3000'. It shows several GET requests from 127.0.0.1, including a 401 Unauthorized response and a sign_in request. The URL 'http://localhost:3000' is circled in red in the original image.

```
root@DESKTOP-7IQ25AI: ~/workspace/pmbokprojects
root@DESKTOP-7IQ25AI:~/workspace/pmbokprojects# rails db:seed
root@DESKTOP-7IQ25AI:~/workspace/pmbokprojects# rails server
=> Booting Puma
=> Rails 5.0.6 application starting in development on http://localhost:3000
=> Run `rails server -h` for more startup options
Puma starting in single mode...
* Version 3.10.0 (ruby 2.4.2-p198), codename: Russell's Teapot
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://0.0.0.0:3000
Use Ctrl-C to stop
Started GET "/" for 127.0.0.1 at 2018-01-25 02:53:53 -0300
  ActiveRecord::SchemaMigration Load (0.2ms)  SELECT "schema_migrations".* FROM "schema_migrations"
Started GET "/" for 127.0.0.1 at 2018-01-25 02:53:53 -0300
Started GET "/" for 127.0.0.1 at 2018-01-25 02:53:53 -0300
Processing by Students::HomeController#index as HTML
  Parameters: {"subdomain"=>"www"}
Completed 401 Unauthorized in 10ms (ActiveRecord: 0.0ms)

Started GET "/sign_in" for 127.0.0.1 at 2018-01-25 02:53:53 -0300
Processing by Students::SessionsController#new as HTML
  Parameters: {"subdomain"=>"www"}
```