



Curso: **Bacharelado em Computação**
Disciplina: **Tópicos Especiais em Tecnologia**
Professor(a): **João Antonio Silva**
2014/02



Configuração do Ambiente

Neste tutorial vamos deixar o ambiente de desenvolvimento pronto para o desenvolvimento com Ruby on Rails. Primeiro, vamos instalar o RVM (*Ruby Version Manager*) o Gerenciador de Versões do Ruby, uma ferramenta que permite trabalhar com ambientes diferentes do Ruby e do Rails.

Para proceder com a instalação, vamos precisar de um editor de texto e do terminal do Ubuntu.

Antes da instalação, recomendo atualizar a lista de repositórios.

Passo 1 – instalar RVM:

Primeiro vamos verificar algumas dependências de instalação. Abra o terminal e verifique se o curl está instalado:

```
curl -help
```

No caso de ele não estar, instale-o com o comando.

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

Agora podemos dar início a instalação do RVM. O primeiro passo é baixar o RVM. Para isso digite o comando abaixo:

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

Sua saída deve ser similar a imagem abaixo:

```
joao@joao-R430-R480-R440: ~
  Dload  Upload  Total  Spent  Left  Speed
100  184  100  184    0    0   138      0  0:00:01  0:00:01 --:--:--  138
100 20776  100 20776    0    0  5118      0  0:00:04  0:00:04 --:--:--  9362
Downloading https://github.com/wayneeseguin/rvm/archive/stable.tar.gz

Upgrading the RVM installation in /home/joao/.rvm/
  RVM PATH line found in /home/joao/.profile /home/joao/.bashrc /home/joao/.zshrc.
  RVM sourcing line found in /home/joao/.bashrc /home/joao/.bash_profile /home/joao/.zlogin.
Upgrade of RVM in /home/joao/.rvm/ is complete.

# João Silva,
#
# Thank you for using RVM!
# We sincerely hope that RVM helps to make your life easier and more enjoyable!!!
#
# ~Wayne, Michal & team.

In case of problems: http://rvm.io/help and https://twitter.com/rvm_io

Upgrade Notes:

* No new notes to display.

joao@joao-R430-R480-R440:~$
```

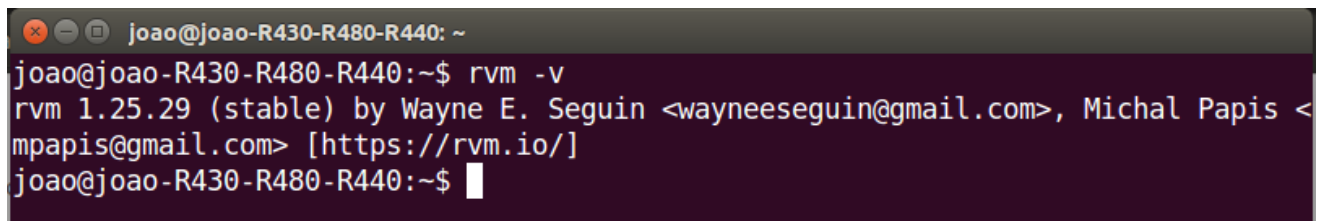
O próximo passo é configurar o rvm no Sistema Operacional. Digite o seguinte comando no terminal:

```
source ~/.rvm/scripts/rvm
```

Para verificar se o procedimento deu certo, digite no terminal o comando abaixo:

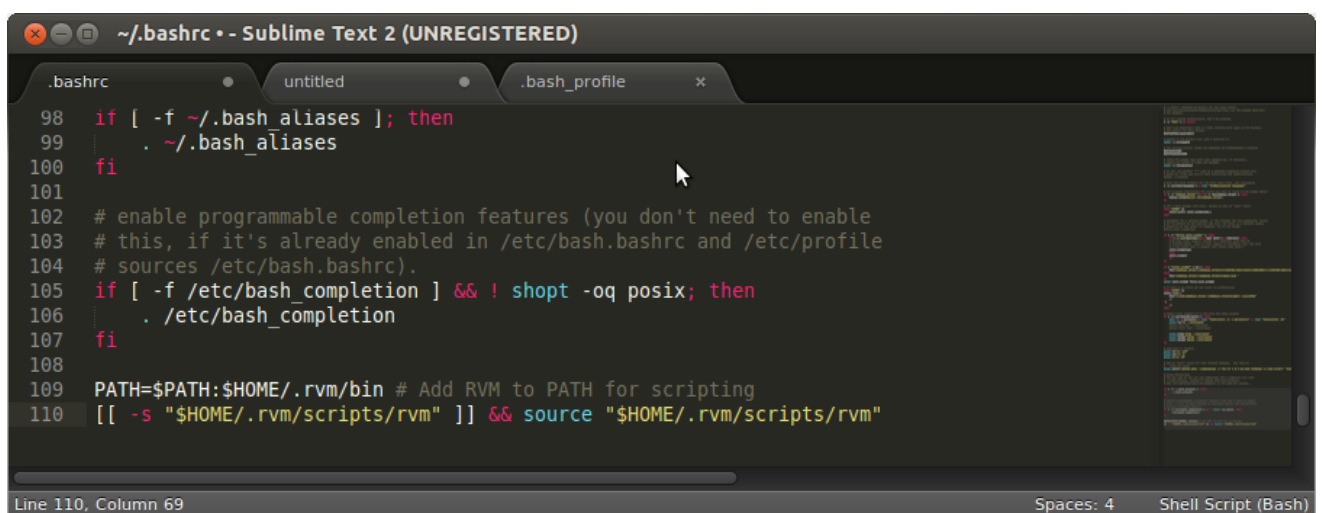
```
rvm -v
```

Se sua foi similar a da imagem, o RVM está instalado na sua máquina.

A terminal window with a dark background. The prompt is 'joao@joao-R430-R480-R440: ~'. The user enters 'rvm -v'. The output is 'rvm 1.25.29 (stable) by Wayne E. Seguin <wayneesequin@gmail.com>, Michal Papis <mpapis@gmail.com> [https://rvm.io/]'. The prompt returns to 'joao@joao-R430-R480-R440:~\$'.

Agora precisamos fazer o Ubuntu reconhecer os comandos do RVM toda vez que utilizarmos o terminal. Para isso, abra o arquivo **bashrc** e adicione o seguinte trecho de código ao final dele e, depois, salve e feche esse arquivo.

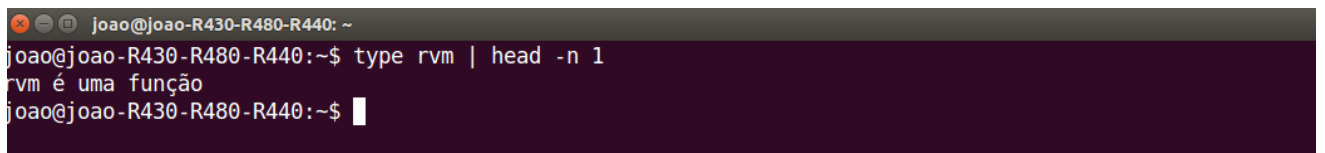
```
[[ -s "$HOME/.rvm/scripts/rvm" ]] && source "$HOME/.rvm/scripts/rvm"
```

A screenshot of the Sublime Text 2 editor. The title bar says '~/.bashrc - Sublime Text 2 (UNREGISTERED)'. There are three tabs: '.bashrc', 'untitled', and '.bash_profile'. The '.bashrc' tab is active, showing a shell script. Lines 109 and 110 are highlighted. Line 109 is 'PATH=\$PATH:\$HOME/.rvm/bin # Add RVM to PATH for scripting'. Line 110 is '[[-s "\$HOME/.rvm/scripts/rvm"]] && source "\$HOME/.rvm/scripts/rvm"'. The status bar at the bottom shows 'Line 110, Column 69', 'Spaces: 4', and 'Shell Script (Bash)'.

Para terminar a instalação do RVM, aplique o seguinte comando no terminal:

```
source ~/.bashrc
```

Agora o Ubuntu vai ler o RVM sempre que utilizarmos a linha de comando. Para verificar isso, digite o comando abaixo e veja que o RVM é agora uma função do do shell.

A terminal window with a dark background. The prompt is 'joao@joao-R430-R480-R440: ~'. The user enters 'type rvm | head -n 1'. The output is 'rvm é uma função'. The prompt returns to 'joao@joao-R430-R480-R440:~\$'.

O RVM tem algumas dependências, necessárias para trabalhar perfeitamente no Ubuntu. Para vermos quais são, digite o comando e observe a saída na próxima figura. Quando for solicitado, informe sua senha de root. Após este passo, todas as dependências necessárias foram instaladas.

```
rvm requirements
```

```
Atingido http://br.archive.ubuntu.com precise-updates/multiverse Translation-en
Obter:7 http://br.archive.ubuntu.com precise-updates/restricted Translation-pt_BR [2.722 B]
Atingido http://br.archive.ubuntu.com precise-updates/restricted Translation-pt
Atingido http://br.archive.ubuntu.com precise-updates/restricted Translation-en
Obter:8 http://br.archive.ubuntu.com precise-updates/universe Translation-pt_BR [752 kB]
Atingido http://br.archive.ubuntu.com precise-updates/universe Translation-pt
Atingido http://br.archive.ubuntu.com precise-updates/universe Translation-en
Atingido http://br.archive.ubuntu.com precise-backports/main Translation-en
Atingido http://br.archive.ubuntu.com precise-backports/multiverse Translation-en
Atingido http://br.archive.ubuntu.com precise-backports/restricted Translation-en
Atingido http://br.archive.ubuntu.com precise-backports/universe Translation-en
Ign http://dl.google.com stable/main Translation-pt_BR
Ign http://dl.google.com stable/main Translation-pt
Ign http://extras.ubuntu.com precise/main Translation-pt_BR
Ign http://dl.google.com stable/main Translation-en
Ign http://extras.ubuntu.com precise/main Translation-pt
Ign http://extras.ubuntu.com precise/main Translation-en
Ign http://ppa.launchpad.net precise/main Translation-pt_BR
Ign http://ppa.launchpad.net precise/main Translation-pt
Ign http://ppa.launchpad.net precise/main Translation-en
Ign http://ppa.launchpad.net precise/main Translation-pt_BR
Ign http://ppa.launchpad.net precise/main Translation-pt
Ign http://ppa.launchpad.net precise/main Translation-en
Baixados 2.172 kB em 5s (387 kB/s)
Lendo listas de pacotes...
Installing required packages: gawk, g++, libreadline6-dev, zlib1g-dev, libssl-dev, libyaml-de
qlite3-dev, sqlite3, libxml2-dev, libxslt1-dev, autoconf, libgdbm-dev, libncurses5-dev, autom
btool, bison, libffi-dev.....
```

Já que temos o RVM instalado, vamos verificar quais versões do Ruby estão disponíveis no gerenciador de versões. Para isto, digite:

```
rvm list known
```

```
joao@joao-R430-R480-R440: ~
joao@joao-R430-R480-R440:~$ rvm list known
# MRI Rubies
[ruby-]1.8.6[-p420]
[ruby-]1.8.7[-head] # security released on head
[ruby-]1.9.1[-p431]
[ruby-]1.9.2[-p330]
[ruby-]1.9.3[-p547]
[ruby-]2.0.0-p451
[ruby-]2.0.0[-p481]
[ruby-]2.1.1
[ruby-]2.1[.2]
[ruby-]2.1-head
ruby-head

# GoRuby
goruby
```

Se sua saída foi semelhante a imagem acima então o RVM está funcional corretamente em sua máquina e estamos aptos a criar o ambiente de desenvolvimento.

Passo 2 -Instalação do Ruby

O RVM permite que instalemos várias versões do Ruby. E é possível instalar várias versões do Rails em cada uma dessas instalações. Neste tutorial, criaremos um ambiente com o Ruby 2 e o Rails 4.

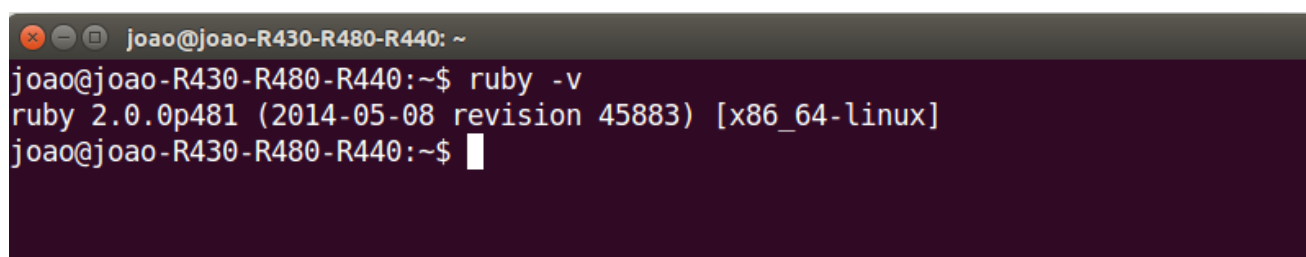
O comando abaixo instala esta versão do Ruby:

```
rvm install 2.0.0 --autolibs=packages
```

Para os que desconhecem o conceito de autolibs, este é um recurso embutido no RVM para permitir a instalação de dependências automaticamente em seu sistema. Em outras palavras, tudo que a versão do ruby necessitar para funcionar é configurado e instalado automaticamente.

Para verificar se a instalação foi bem sucedida digite o seguinte comando para ver a versão do Ruby ativa:

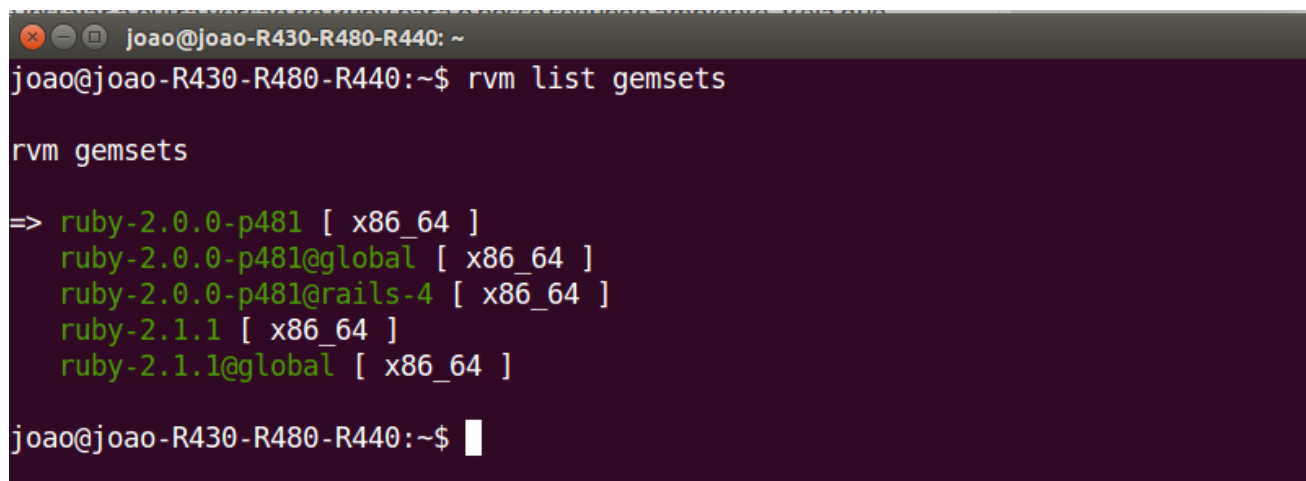
```
ruby -v
```



```
joao@joao-R430-R480-R440: ~  
joao@joao-R430-R480-R440:~$ ruby -v  
ruby 2.0.0p481 (2014-05-08 revision 45883) [x86_64-linux]  
joao@joao-R430-R480-R440:~$
```

Para verificar se tudo está correto, informe o seguinte comando no terminal:

```
rvm list gemsets
```



```
joao@joao-R430-R480-R440: ~  
joao@joao-R430-R480-R440:~$ rvm list gemsets  
  
rvm gemsets  
  
=> ruby-2.0.0-p481 [ x86_64 ]  
    ruby-2.0.0-p481@global [ x86_64 ]  
    ruby-2.0.0-p481@rails-4 [ x86_64 ]  
    ruby-2.1.1 [ x86_64 ]  
    ruby-2.1.1@global [ x86_64 ]  
  
joao@joao-R430-R480-R440:~$
```

A imagem acima indica que há duas versões do Ruby instaladas, e a versão padrão está indicada pela seta. E para mudar de uma versão para outra basta utilizar o comando:

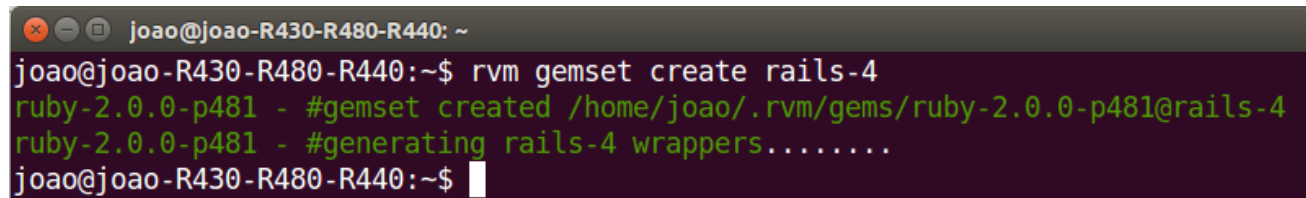
```
rvm use <versão> [--default]
```

Passo 3 – Configurando o ambientes

O ambiente vai ser composto pelo Ruby 2 e o Rails 4. Neste ponto, falta instalar o framework Rails. Antes da instalação, certifique-se de que a versão 2 do Ruby está setada para trabalhar com o Rails. É

bom lembrar que com o RVM, pode-se ter várias versões tanto do Ruby quanto do Rails. Chamaremos o ambiente criado neste tutorial de rails-4. Agora digite o comando:

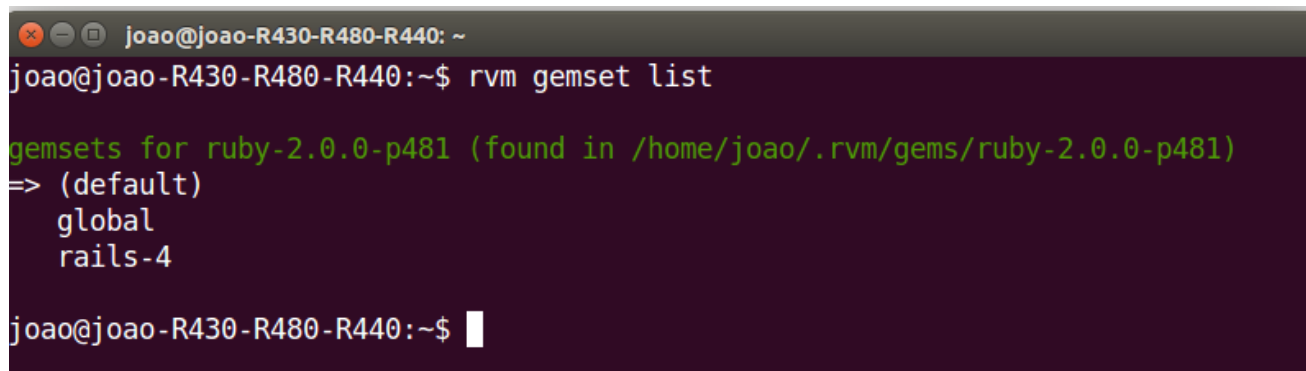
```
rvm gemset create rails-4
```



```
joao@joao-R430-R480-R440: ~  
joao@joao-R430-R480-R440:~$ rvm gemset create rails-4  
ruby-2.0.0-p481 - #gemset created /home/joao/.rvm/gems/ruby-2.0.0-p481@rails-4  
ruby-2.0.0-p481 - #generating rails-4 wrappers.....  
joao@joao-R430-R480-R440:~$
```

Para verificar os ambientes criados, é só digitar:

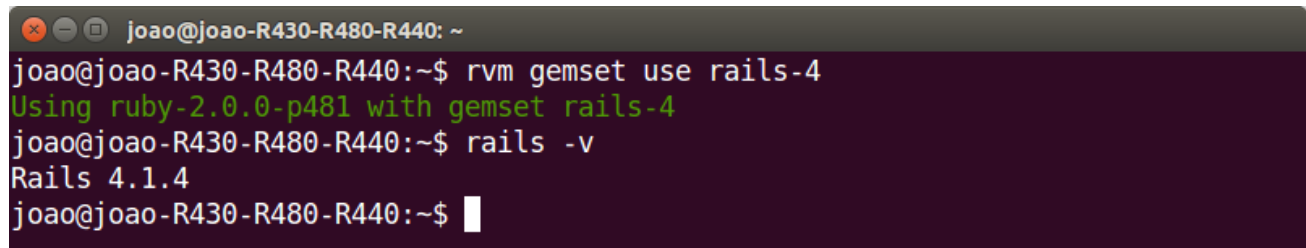
```
rvm gemset list
```



```
joao@joao-R430-R480-R440: ~  
joao@joao-R430-R480-R440:~$ rvm gemset list  
  
gemsets for ruby-2.0.0-p481 (found in /home/joao/.rvm/gems/ruby-2.0.0-p481)  
=> (default)  
    global  
    rails-4  
  
joao@joao-R430-R480-R440:~$
```

O próximo passo é selecionar o ambiente que acabou de ser criado:

```
rvm gemset use rails-4
```



```
joao@joao-R430-R480-R440: ~  
joao@joao-R430-R480-R440:~$ rvm gemset use rails-4  
Using ruby-2.0.0-p481 with gemset rails-4  
joao@joao-R430-R480-R440:~$ rails -v  
Rails 4.1.4  
joao@joao-R430-R480-R440:~$
```

E instalar o Rails nele:

```
gem install rails
```

Vamos fazer alguns testes. Antes feche o terminal e depois de abri um outro, informe os comandos:

```
rvm gemset use rails-4
```

```
rails -v
```

```
joao@joao-R430-R480-R440: ~
joao@joao-R430-R480-R440:~$ rvm gemset use rails-4
Using ruby-2.0.0-p481 with gemset rails-4
joao@joao-R430-R480-R440:~$ rails -v
Rails 4.1.4
joao@joao-R430-R480-R440:~$
```

Se após o comando acima sua saída foi algo semelhante a versão “**Rails 4.X.X**” então está tudo ok para você utilizar essa versão do Rails.

Para finalizar, vamos criar uma aplicação em Rails 4. Digite o comando:

```
rails new myFirstApp
```

Se não ocorrer nenhum erro, a seguinte frase deve ser exibida:

Your bundle is complete!
Use bundle show [gemname] to see where a bundled gem is installed.

Edite o arquivo Gemfile no diretório do projeto criado (no meu caso o projeto foi criado em minha pasta pessoal). Descomente a linha referente a **gem 'therubyracer'** e depois salve o arquivo.

```
~/Modelos/rails4ruby2/Gemfile - Sublime Text 2 (UNREGISTERED)
gemfile
4  gem 'rails', '4.0.0.beta1'
5
6  gem 'sqlite3'
7
8  # Gems used only for assets and not required
9  # in production environments by default.
10 group :assets do
11   gem 'sass-rails', '~> 4.0.0.beta1'
12   gem 'coffee-rails', '~> 4.0.0.beta1'
13 end
14 # See https://github.com/sstephenson/execjs#readme for more supported runtimes
15 gem 'therubyracer', :platforms => :ruby
16
17 gem 'uglifier', '>= 1.0.3'
18 end
19
20 gem 'jquery-rails'
21
22 # Turbolinks makes following links in your web application faster. Read more: https://github.
23 gem 'turbolinks'
24
25 # Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
26 gem 'jbuilder', '~> 1.0.1'
27
```

Dentro do diretório do projeto, execute o comando:

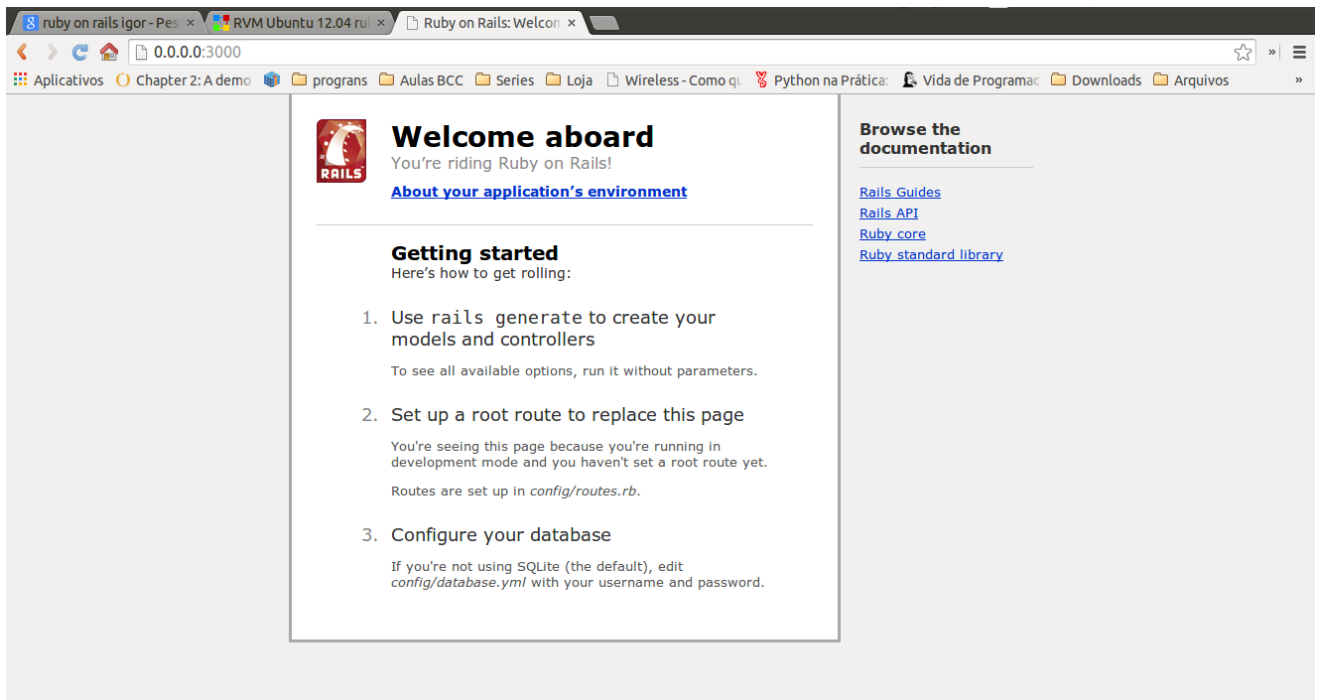
```
bundle install
```

Pronto, agora nossa aplicação em Rails 4 e Ruby 2 deve está funcionando. Para verificarmos nossa aplicação, suba o servidor Rails e depois verificar no browser. Digite o comando abaixo, dentro do diretório do nosso projeto, na linha de comando:

```
rails s
```

E em um browser:

<http://0.0.0.0:3000>



Voilà! Temos nossa primeira aplicação em Ruby on Rails.