



SAAS (guía 4)

practica 4 Ruby

JOSE ABRAHAM SCHIFFMANN VANEGA 19-00501-0

Desarrollo

1. Verificar las versiones de software instaladas

- Abre una terminal y ejecuta los siguientes comandos para verificar las versiones instaladas:

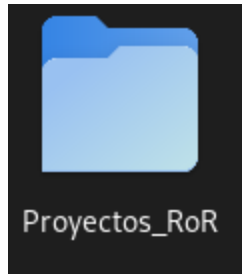
```
$ ruby -v      # Verifica la versión de Ruby
$ rails -v     # Verifica la versión de Rails
$ sqlite3 --version  # Verifica la versión de SQLite
```

```
riot@H3NT41: ~
riot@H3NT41:~$ ruby -v
ruby 3.1.2p20 (2022-04-12 revision 4491bb740a) [x86_64-linux]
riot@H3NT41:~$ rails -v
Rails 7.1.3.4
riot@H3NT41:~$ sqlite3 --version
3.40.1 2022-12-28 14:03:47 df5c253c0b3dd24916e4ec7cf77d3db5294cc9fd45ae7b9c5e82ad8197f3alt1
riot@H3NT41:~$
```

2. Crear un nuevo proyecto en Rails

1. Navega a la carpeta donde quieras almacenar el proyecto:

```
$ cd /home/debian/Proyectos_RoR
```



2. Genera un nuevo proyecto en Rails:

```
$ rails new first_app
```

Este comando creará la estructura de directorios y archivos necesarios para la aplicación.

```
riot@H3NT41:~/Proyectos_RoR$ rails new first_app
create
create  README.md
create  Rakefile
create  .ruby-version
create  config.ru
create  .gitignore
create  .gitattributes
create  Gemfile
run     git init -b main from "."
Iniciado repositorio Git vacío en /home/riot/Proyectos_RoR/first_app/.git/
create  app
create  app/assets/config/manifest.js
create  app/assets/stylesheets/application.css
create  app/channels/application_cable/channel.rb
create  app/channels/application_cable/connection.rb
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/jobs/application_job.rb
create  app/mailers/application_mailer.rb
create  app/models/application_record.rb
create  app/views/layouts/application.html.erb
create  app/views/layouts/mailer.html.erb
create  app/views/layouts/mailer.text.erb
create  app/assets/images
create  app/assets/images/.keep
create  app/controllers/concerns/.keep
create  app/models/concerns/.keep
create  bin
create  bin/rails
create  bin/rake
create  bin/setup
create  Dockerfile
create  .dockerignore
create  bin/docker-entrypoint
create  config
create  config/routes.rb
create  config/application.rb
```

3. Entra en el directorio del proyecto creado:

```
$ cd first_app/
```

```
riot@H3NT41: ~
riot@H3NT41: ~/Proyectos_RoR/first_app
riot@H3NT41:~/Proyectos_RoR$ cd first_app/
riot@H3NT41:~/Proyectos_RoR/first_app$
```

3. Iniciar el servidor Rails

- Para comprobar que todo está funcionando correctamente, inicia el servidor con:

```
$ rails server
```



```
riot@H3NT41: ~/Proyectos_RoR/first_app
riot@H3NT41: ~/Proyectos_RoR/first_app$ rails server
> Booting Puma
> Rails 7.1.3.4 application starting in development
> Run `bin/rails server --help` for more startup options
uma starting in single mode...
Puma version: 6.4.2 (ruby 3.1.2-p20) ("The Eagle of Durango")
Min threads: 5
Max threads: 5
Environment: development
PID: 4467
Listening on http://127.0.0.1:3000
Listening on http://[::1]:3000
se Ctrl-C to stop
```

- Abre un navegador web y accede a la dirección `http://127.0.0.1:3000/`. Deberías ver la página de bienvenida de Rails.
- Detén el servidor con `Ctrl+C` en la terminal.

4. Agregar una acción al controlador

1. Rails genera un controlador llamado `ApplicationController` por defecto. Para verificar su existencia, usa:

```
$ ls app/controllers/*_controller.rb
```

```
riot@H3NT41:~/Proyectos_RoR/first_app$ cd app
riot@H3NT41:~/Proyectos_RoR/first_app/app$ ls
assets  channels  controllers  helpers  javascript  jobs  mailers  models  views
riot@H3NT41:~/Proyectos_RoR/first_app/app$ cd controllers/
riot@H3NT41:~/Proyectos_RoR/first_app/app/controllers$ ls
application_controller.rb  concerns
riot@H3NT41:~/Proyectos_RoR/first_app/app/controllers$
```

Abre el archivo

`app/controllers/application_controller.rb` en tu editor de texto (Sublime Text) y define una nueva acción en el controlador:

```
class ApplicationController < ActionController::Base
  def hola
    render html: "¡Hola, mundo!"
  end
end
```

```
riot@H3NT41: ~/Proyectos_RoR/first_app/app/controllers
GNU nano 7.2 application_controller.rb *
class ApplicationController < ActionController::Base
  def hola
    render html: "¡Hola, mundo!"
  end
end
```

5. Configurar el archivo routes.rb

1. Abre el archivo `config/routes.rb` y agrega la siguiente línea para definir la ruta raíz que ejecutará la acción `hola`:

```
riot@H3NT41:~/Proyectos_RoR/first_app/config$ ls
application.rb  credentials.yml.enc  environments  locales  routes.rb
boot.rb         database.yml         importmap.rb  master.key  storage.yml
cable.yml       environment.rb       initializers  puma.rb
```

```
root 'application#hola'
```

```
riot@H3NT41: ~/Proyectos_RoR/first_app/config
GNU nano 7.2 routes.rb *
# Define your application routes per the DSL in https://guides.rubyonrails.org/routing.html

# Reveal health status on /up that returns 200 if the app boots with no exceptions, otherwise 500.
# Can be used by load balancers and uptime monitors to verify that the app is live.
get "up" => "rails/health#show", as: :rails_health_check

# Defines the root path route ("/")
# root "posts#index"
root 'application#hola'
```

- 2.Guarda los cambios.

6. Verificar el funcionamiento

1. Inicia nuevamente el servidor Rails con:

```
$ rails server
```

```
riot@H3NT41:~/Proyectos_RoR/first_app$ rails server
=> Booting Puma
=> Rails 7.1.3.4 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 6.4.2 (ruby 3.1.2-p20) ("The Eagle of Durango")
*   Min threads: 5
*   Max threads: 5
*   Environment: development
*         PID: 4691
* Listening on http://127.0.0.1:3000
* Listening on http://[::1]:3000
Use Ctrl-C to stop
■
```

- Abre el navegador y accede a `http://127.0.0.1:3000/`. Ahora deberías ver el texto "¡Hola, mundo!".



- Detén el servidor con `Ctrl+C`.

Ejercicio Propuesto

1. Crear una nueva acción que muestre un texto diferente:

- En el archivo `application_controller.rb`, añade una nueva acción llamada `saludo`:

```
class ApplicationController < ActionController::Base
  def hola
    render html: "¡Hola, mundo!"
  end

  def saludo
    render html: "¡Bienvenido a mi aplicación Rails!"
  end
end
```

Editar el archivo `routes.rb` para que la aplicación inicie con la nueva acción:

- Cambia la ruta raíz para que apunte a la nueva acción `saludo`:

```
root 'application#saludo'
```

```
riot@H3NT41: ~/Proyectos_RoR/first_app/config × riot@H3NT41: ~/Proyectos_RoR/first_app ×
GNU nano 7.2 routes.rb *
rails.application.routes.draw do

  # Define your application routes per the DSL in https://guides.rubyonrails.org/routing.html

  # Reveal health status on /up that returns 200 if the app boots with no exceptions, otherwise 500.
  # Can be used by load balancers and uptime monitors to verify that the app is live.
  get "up" => "rails/health#show", as: :rails_health_check

  # Defines the root path route ("/")
  # root "posts#index"
root 'application#hola'
root 'application#saludo'

end
```

Verificar el funcionamiento:

- Guarda los cambios y reinicia el servidor. Ahora, al acceder a `http://127.0.0.1:3000/`, deberías ver el texto "¡Bienvenido a mi aplicación Rails!".

