

# Universidad Nacional Autónoma de Nicaragua Unan León



**Facultad de Ciencias y Tecnología.**

**Componente: Software como un servicio**

**Tema: Practica 8**

Grupo:1

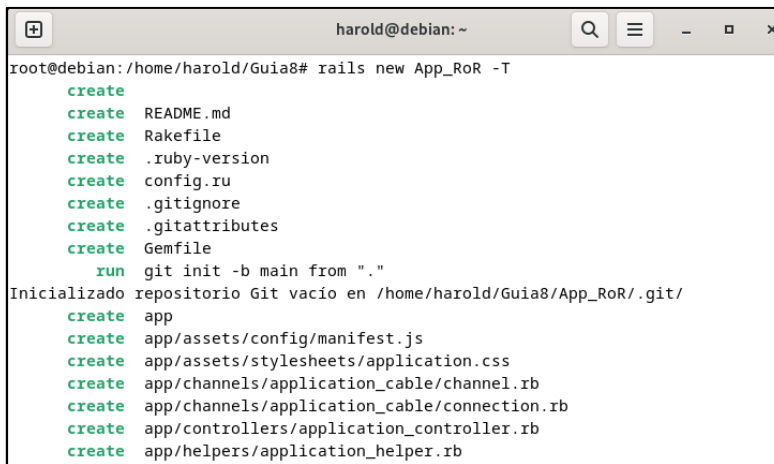
**Docente: Erving Montes**

**Integrante:**

-Harold Steven Garcia Ramos.

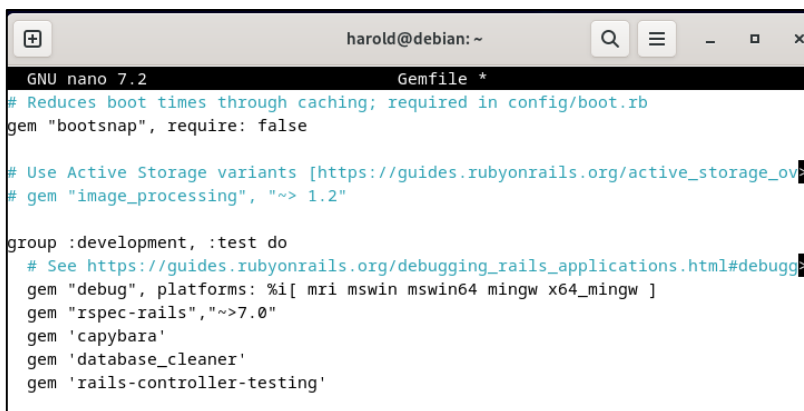
**Fecha:** 2 de octubre de 2024

## 1. Generar un nuevo proyecto.



```
harold@debian: ~  
root@debian:/home/harold/Guia8# rails new App_RoR -T  
create  
create README.md  
create Rakefile  
create .ruby-version  
create config.ru  
create .gitignore  
create .gitattributes  
create Gemfile  
run git init -b main from "."  
Inicializado repositorio Git vacio en /home/harold/Guia8/App_RoR/.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
```

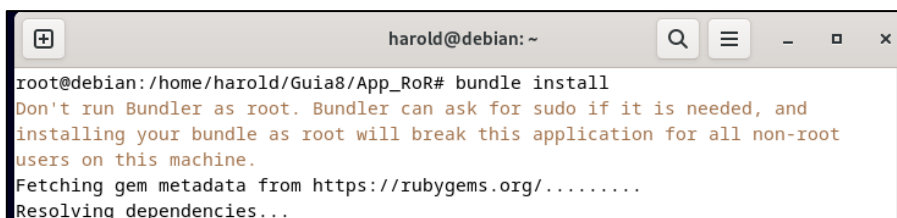
### 1.1. Agregar las gemas que se utilizarán en el proyecto, dentro grupo development, test.



```
harold@debian: ~  
GNU nano 7.2 Gemfile *  
# Reduces boot times through caching; required in config/boot.rb  
gem "bootsnap", require: false  
  
# Use Active Storage variants [https://guides.rubyonrails.org/active_storage_ov>  
# gem "image_processing", "~> 1.2"  
  
group :development, :test do  
# See https://guides.rubyonrails.org/debugging_rails_applications.html#debugg>  
gem "debug", platforms: %i[ mri mswin mswin64 mingw x64_mingw ]  
gem "rspec-rails", "~>7.0"  
gem 'capybara'  
gem 'database_cleaner'  
gem 'rails-controller-testing'
```

### 1.2. El grupo development, test quedará como se muestra en la siguiente figura.

### 1.3. Ejecutar el siguiente comando en el terminal para instalar las gemas.



```
harold@debian: ~  
root@debian:/home/harold/Guia8/App_RoR# bundle install  
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and  
installing your bundle as root will break this application for all non-root  
users on this machine.  
Fetching gem metadata from https://rubygems.org/.....  
Resolving dependencies...
```

```

Using sqlite3 1.7.3 (x86_64-linux)
Using stimulus-rails 1.3.4
Using turbo-rails 2.0.10
Using web-console 4.2.1
Bundle complete! 17 Gemfile dependencies, 88 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
root@debian: /home/harold/Guia8/App_RoR#

```

1.4. Ejecutar el siguiente comando para crear los directorios que rspec necesita para trabajar.



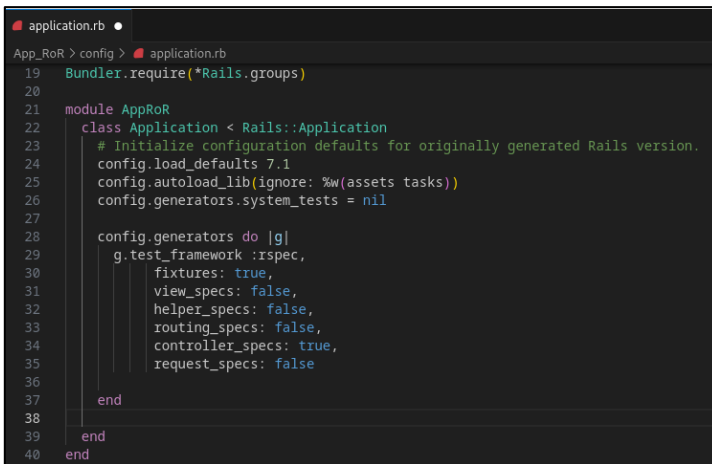
```

harold@debian: ~
root@debian: /home/harold/Guia8/App_RoR# rails g rspec:install
create .rspec
create spec
create spec/spec_helper.rb
create spec/rails_helper.rb
root@debian: /home/harold/Guia8/App_RoR#

```

2. Configuración de Rspec.

2.1. Abrir el archivo config/application.rb y agregar el siguiente código dentro de la clase, en el mismo archivo.

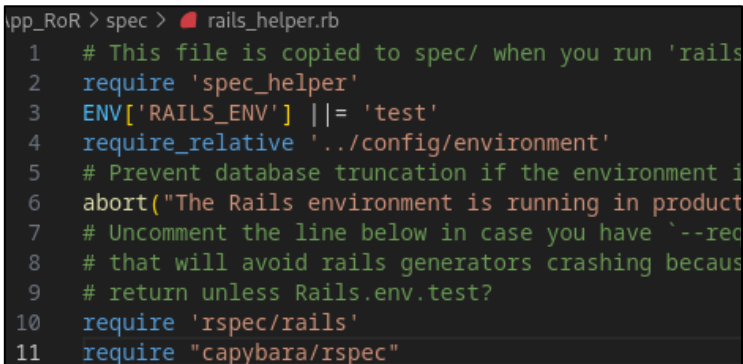


```

application.rb
App_RoR > config > application.rb
19 Bundler.require(*Rails.groups)
20
21 module AppRoR
22   class Application < Rails::Application
23     # Initialize configuration defaults for originally generated Rails version.
24     config.load_defaults 7.1
25     config.autoload_lib(ignore: %w(assets tasks))
26     config.generators.system_tests = nil
27
28     config.generators do |g|
29       g.test_framework :rspec,
30         fixtures: true,
31         view_specs: false,
32         helper_specs: false,
33         routing_specs: false,
34         controller_specs: true,
35         request_specs: false
36     end
37   end
38 end
39
40 end

```

2.2. Abrir el archivo rails\_helper.rb y agregar debajo de la línea require 'rspec/rails'.



```

App_RoR > spec > rails_helper.rb
1 # This file is copied to spec/ when you run 'rails
2 require 'spec_helper'
3 ENV['RAILS_ENV'] ||= 'test'
4 require_relative '../config/environment'
5 # Prevent database truncation if the environment is
6 abort("The Rails environment is running in product
7 # Uncomment the line below in case you have '--rec
8 # that will avoid rails generators crashing becaus
9 # return unless Rails.env.test?
10 require 'rspec/rails'
11 require "capybara/rspec"

```

2.3. Ejecutar el comando y verificar que no muestra ningún mensaje en rojo.

```
root@debian:/home/harold/Guia8/App_RoR# rspec
No examples found.

Finished in 0.00048 seconds (files took 0.64734 seconds to load)
0 examples, 0 failures

root@debian:/home/harold/Guia8/App_RoR#
```

### 3. Static pages

3.1. Para crear páginas estáticas escribir lo siguiente en el terminal.

```
root@debian:/home/harold/Guia8/App_RoR# rails generate controller StaticPages home help
  create  app/controllers/static_pages_controller.rb
  route   get 'static_pages/home'
         get 'static_pages/help'
  invoke  erb
  create  app/views/static_pages
  create  app/views/static_pages/home.html.erb
  create  app/views/static_pages/help.html.erb
  invoke  rspec
  create  spec/controllers/static_pages_controller_spec.rb
  invoke  helper
  create  app/helpers/static_pages_helper.rb
  invoke  rspec
root@debian:/home/harold/Guia8/App_RoR#
```

3.2. Verificar la configuración del archivo routes.rb.

```
RoR > config > routes.rb

Rails.application.routes.draw do
  get 'static_pages/home'
  get 'static_pages/help'
  # Define your application routes per the DSL in https://guides.rubyonrails.org/

  # Reveal health status on /up that returns 200 if the app boots with no exceptions
  # 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"
end
```

3.3. Verificar que se han creado las vistas en app/views/static\_pages/.

```
static_pages
  help.html.erb
  home.html.erb
```

#### 4. Modificar las vistas.

##### 4.1. Agregar al archivo [home.html.erb](#)

```
App_RoR > app > views > static_pages > <> home.html.erb
1 <h1>StaticPages#home</h1>
2 <h1>Sample App</h1>
3 <p>
4   This is the home page for the
5   <a href="http://www.railstutorial.org/">Ruby on Rails Tutorial</a>
6   sample application.
7 </p>
8
```

##### 4.2. Modificar el archivo help.html.erb.

```
App_RoR > app > views > static_pages > <> help.html.erb
1 <h1>StaticPages#help</h1>
2 <h1>Help</h1>
3 <p>
4   Get help on the Ruby on Rails Tutorial at the
5   <a href="http://www.railstutorial.org/help">Rails Tutorial help page</a>
6   To get help on this sample app, see the
7   <a href="http://www.railstutorial.org/book"><em>Ruby on Rails Tutorial</em>
8   book</a>
9 </p>
```

#### 5. Hacer testing con rspec.

5.1. El primer paso será ejecutar el comando db:migrate en el terminal para crear el esquema de pruebas con el que se ejecutarán los tests.

```
root@debian:/home/harold/Guia8/App_RoR# rails db:migrate
```

##### 5.2. Probar hacer testing escribiendo en el terminal el comando.

```
root@debian:/home/harold/Guia8/App_RoR# bundle exec rspec spec/controllers/static_pages_controller_spec.rb
..

Finished in 0.19572 seconds (files took 5.3 seconds to load)
2 examples, 0 failures

root@debian:/home/harold/Guia8/App_RoR#
```

5.3. Agregar siguiente código dentro del archivo `static_pages_controller_spec.rb` del directorio `spec/controllers/`.

```
App_RoR > spec > controllers > static_pages_controller_spec.rb
3  RSpec.describe StaticPagesController, type: :controller do
4
5      describe "GET #home" do
6          it "returns http success" do
7              get :home
8              expect(response).to have_http_status(:success)
9          end
10     end
11
12     describe "GET #help" do
13         it "returns http success" do
14             get :help
15             expect(response).to have_http_status(:success)
16         end
17     end
18     describe "GET #about" do
19         it "returns http success" do
20             get :about
21             expect(response).to have_http_status(:success)
22         end
23     end
24
25 end
```

5.4. Ejecutar el comando `rspec` y ver que muestra el test en rojo.

```
root@debian:/home/harold/Guia8/App_RoR# bundle exec rspec spec/controllers/static_pages_controller_spec.rb
..F

Failures:

  1) StaticPagesController GET #about returns http success
     Failure/Error: get :about

     ActionController::UrlGenerationError:
       No route matches {:action=>"about", :controller=>"static_pages"}
       # ./spec/controllers/static_pages_controller_spec.rb:20:in `block (3 levels) in <top (required)>'

Finished in 0.05729 seconds (files took 1.65 seconds to load)
3 examples, 1 failure

Failed examples:

rspec ./spec/controllers/static_pages_controller_spec.rb:19 # StaticPagesControl
```

5.5. Editar el archivo `routes.rb` y agregar lo siguiente.

```
pp_RoR > config > routes.rb
1  Rails.application.routes.draw do
2      get 'static_pages/home'
3      get 'static_pages/help'
4      get 'static_pages/about'
5      # Define your application routes per the DSL
6  end
```

5.6. Ejecutar el test de nuevo. El mensaje que muestra, es debido a que no existe la acción about que intenta alcanzar el método get.

```
root@debian:/home/harold/Guia8/App_RoR# bundle exec rspec spec/controllers/static_pages_controller_spec.rb
..F

Failures:

  1) StaticPagesController GET #about returns http success
     Failure/Error: get :about

     ActionController::ActionNotFound:
       The action 'about' could not be found for StaticPagesController
     # ./spec/controllers/static_pages_controller_spec.rb:20:in `block (3 levels) in <top (required)>'

Finished in 0.06207 seconds (files took 1.53 seconds to load)
3 examples, 1 failure

Failed examples:

rspec ./spec/controllers/static_pages_controller_spec.rb:19 # StaticPagesCont
```

5.7. Agregar el método en el archivo controllers/static\_pages\_controller.rb y luego ejecutar el test.

```
def about
end
```

5.8. Crear una vista para ese método. Crear el archivo about.html.erb en el directorio app/views/static\_pages/ y agregar el código html.

```
> app > views > static_pages > <> about.html.erb
<h1>About</h1>
<p>
  The <a href="http://www.railstutorial.org/"><em>Ruby on Rails Tutorial</em></a> is
  <a href="http://www.railstutorial.org/book">book</a> and
  <a href="http://screencasts.railstutorial.org/">screencasts series</a>
  to teach web development with
  <a href="http://rubyonrails.org/">Ruby on Rails</a>
  This is the sample application for the tutorial.
</p>
```

5.9. Para comprobar que todo funciona, ejecutar el test nuevamente.

```
root@debian:/home/harold/Guia8/App_RoR# bundle exec rspec spec/controllers/static_pages_controller_spec.rb
...

Finished in 0.11178 seconds (files took 2.46 seconds to load)
3 examples, 0 failures
```

## 6. Configurar el título de la página.

6.1. Para que el título de la página cambie por cada una de las vistas a la que se accede en la aplicación, copiar el código en el archivo `app/helpers/application_helper.rb` dentro del módulo.

```
App_RoR > app > helpers > application_helper.rb
1  module ApplicationHelper
2    def full_title(page_title="")
3      base_title= "Rails_App"
4      if page_title.empty?
5        base_title
6      else
7        page_title+" | "+base_title
8      end
9    end
10 end
11
```

6.2. En el archivo `app/views/layouts/application.html.erb` modificar la línea de la etiqueta `title` por la siguiente.

```
<!DOCTYPE html>
<html>
  <head>
    <title><%=full_title(yield(:title))%></title>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <%= csrf_meta_tags %>
    <%= csp_meta_tag %>
    <%= stylesheet_link_tag "application", "data-turbo-track": "reload" %>
    <%= javascript_importmap_tags %>
  </head>
  <body>
    <%= yield %>
  </body>
</html>
```

## 7. Modificar los archivos de las vistas.

7.1. En el archivo `help.html.erb`, agregar la siguiente línea en la parte superior del código.

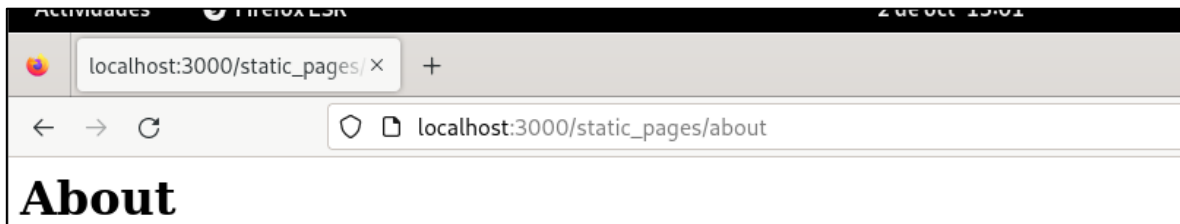
```
App_RoR > app > views > static_pages > help.html.erb
1  <%= provide(:title, "Help") %>
2  <h1>StaticPages#help</h1>
3  <h1>Help</h1>
4  <p>
5    Get help on the Ruby on Rails Tutorial at the
6    <a href="http://www.railstutorial.org/help">Rails Tutorial help page</a>
7    To get help on this sample app, see the
8    <a href="http://www.railstutorial.org/book"><em>Ruby on Rails Tutorial</em>
9    book</a>
10 </p>
```



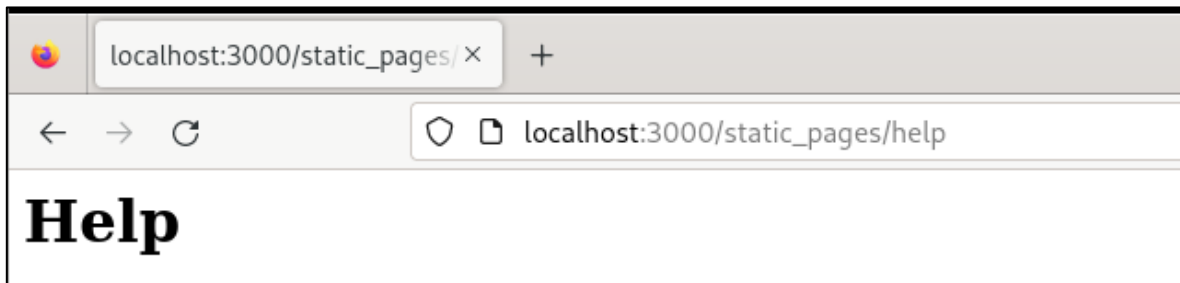
7.2. En el archivo about.html.erb, agregar la siguiente línea en la parte superior de todo el código.

```
App_RoR > app > views > static_pages > <> about.html.erb
1 <% provide(:title, "About") %>
2 <h1>About</h1>
3 <p>
4   The <a href="http://www.railstutorial.org/"><em>Ruby on Rails Tutorial</em></a> is a
5   <a href="http://www.railstutorial.org/book">book</a> and
6   <a href="http://screencasts.railstutorial.org/">screencasts series</a>
7   to teach web development with
8   <a href="http://rubyonrails.org/">Ruby on Rails</a>
9   This is the sample application for the tutorial.
10 </p>
```

7.3. Navegar en la dirección [http://localhost:3000/static\\_pages/about](http://localhost:3000/static_pages/about) y verificar que cambia el título de la página.



7.4. Navegar en la dirección [http://localhost:3000/static\\_pages/help](http://localhost:3000/static_pages/help) y verificar que cambia el título de la página.



#### Ejercicios Propuestos:

2. Copiar el código del recuadro, en el archivo `static_pages_controller_spec.rb` del directorio `spec/controllers/`.

```
describe "GET #contact" do
  it "returns http success" do
    get :contact
    expect(response).to have_http_status(:success)
  end
end
```

3. Ejecutar el test y deberá mostrar los siguientes mensajes.

```
root@debian:/home/harold/Guia8/App_RoR# bundle exec rspec spec/controllers/static_pages_controller_spec.rb

While loading ./spec/controllers/static_pages_controller_spec.rb a 'raise SyntaxError' occurred, RSpec will now quit.
Failure/Error: require 'rspec/rails'

SyntaxError:
  /home/harold/Guia8/App_RoR/app/helpers/application_helper.rb:3: syntax error, unexpected constant, expecting ')'
    base_title="Rails_App"
    ^~~~~~
  /home/harold/Guia8/App_RoR/app/helpers/application_helper.rb:10: unterminated string meets end of file
  /home/harold/Guia8/App_RoR/app/helpers/application_helper.rb:10: syntax error, unexpected end-of-input, expecting 'end'
# ./spec/rails_helper.rb:10:in `<top (required)>'
# ./spec/controllers/static_pages_controller_spec.rb:1:in `require'
# ./spec/controllers/static_pages_controller_spec.rb:1:in `<top (required)>'

Finished in 0.01696 seconds (files took 2.34 seconds to load)
0 examples, 0 failures, 1 error occurred outside of examples
```

4. Realizar los mismos pasos que realizo con la página about y hacer que el test pase.

```
j> routes.rb
Rails.application.routes.draw do
  get 'static_pages/home'
  get 'static_pages/help'
  get 'static_pages/about'
  get 'static_pages/contact'
  # Define your application routes per the DSL in https://guides.rubyonrails.org/routing.html

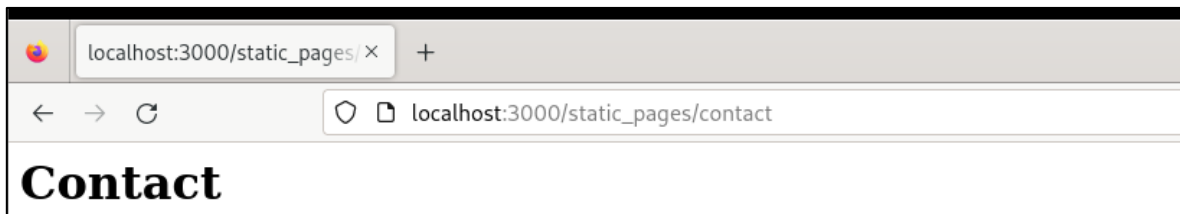
  # Reveal health status on /up that returns 200 if the app boots with no exception
  # Can be used by load balancers and uptime monitors to verify that the app is live
  get "up" => "rails/health#show", as: :rails_health_check

  # Defines the root path route ("/")
  # root "posts#index"
end
```

```
end
def contact
end
end
```

```
> views > static_pages > <> contact.html.erb
<% provide(:title, "Contact") %>
<h1>Contact</h1>
<p>
  The <a href="http://www.railstutorial.org/"><em>Ruby on Rails Tutorial</em></a>
  <a href="http://www.railstutorial.org/book">book</a> and
  <a href="http://screencasts.railstutorial.org/">screencasts series</a>
  to teach web development with
  <a href="http://rubyonrails.org/">Ruby on Rails</a>
  This is the sample application for the tutorial.
</p>
```

5. Navegar en la dirección [http://localhost:3000/static\\_pages/contact](http://localhost:3000/static_pages/contact).



6. Configurar el archivo config/routes.rb, para que la página de inicio del proyecto sea el archivo home que se encuentra dentro del directorio static\_pages, realizar la prueba navegando en la dirección localhost:3000, como se muestra en la figura 105.

