

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA



UNAN-LEON

FACULTAD DE CIENCIAS Y TECNOLOGIA

SOTFWARE COMO SERVICIO

TEMA: GUÍA 8: RUBY ON RAILS CON TDD

DOCENTE: Ervin Ismael Montes Tellez

INTEGRANTES:

 **LENUEL GASTON PEREIRA HERNANDEZ 19-03199-0**

 **INGRID VALERIA RUIZ ULLOA 21-00483-0**

Guía 8: Ruby on Rails con TDD

Introducción

Test Driver Development (TDD) conocido como desarrollo en base a pruebas, es una de las principales técnicas de desarrollo ágil más utilizadas para trabajar con Ruby on Rails, se centra básicamente en escribir un pequeño test para un código que no existe todavía.

En esta guía el estudiante configurará un proyecto para poder trabajar con el desarrollo en base a pruebas y poder realizar una aplicación básica tomada de (Hartl, s.f.), se instalará un conjunto de gemas necesarias para el correcto funcionamiento del proyecto, dentro de las cuales se podría mencionar Rspec que es la gema principal para realizar las pruebas. De igual forma se introducirá más en el desarrollo de aplicaciones web utilizando el framework Ruby on Rails.

Objetivos

- Configurar el escenario para hacer test.
- Hacer uso de la herramienta Rspec.
- Crear una aplicación en RoR

Tiempo

- Una sesión de clase.

Requerimientos

Software	Hardware
Sistema debian 9 virtualizado en Virtual box con: <ul style="list-style-type: none">• Ruby versión 2.4.1• Rails 5.4.1• Sublimetext• Rspec	Computadora con características: <ul style="list-style-type: none">• Memoria ram mínimo 2GB• Procesador mínimo 2.1 GHz

1. Generar un nuevo proyecto.

\$ rails new App_RoR -T

```
lenuelp@Debian12:~/Desktop/Guia#8$ 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 from "."
```

Al generar un nuevo proyecto se escribe el -T, para que el framework no configure la herramienta de pruebas, que se instala por defecto al momento de generar el proyecto en Rails.

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

"rspec-rails", "~> 3.7"

```
gem 'capybara'
gem 'database_cleaner'
```

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

```
group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug', platforms: [:mri, :mingw, :x64_mingw]
  gem "rspec-rails", "~> 3.7"
  gem 'capybara'
  gem 'database_cleaner'
end
```

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

\$ bundle install

```
lenuelp@Debian12:~/Desktop/Guia8/App_RoR$ bundle install
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies.....
Using rake 13.0.6
Following files may not be writable, so sudo is needed:
  /usr/local/bin
  /var/lib/gems/3.1.0
Using concurrent-ruby 1.1.6
Using i18n 1.10.0
Using minitest 5.15.0
Using tzinfo 2.0.5
Using zeitwerk 2.6.1
Using activesupport 6.1.7.3
Using builder 3.2.4
Using erubi 1.9.0
Using racc 1.6.2
Fetching nokogiri 1.13.10 (x86_64-linux)
```

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

\$ rails g rspec:install

```
lenuelp@Debian12:~/Desktop/GUIA8/App_RoR$ rails g rspec:install
Running via Spring preloader in process 58575
  create  .rspec
  create  spec
  create  spec/spec_helper.rb
  create  spec/rails_helper.rb
```

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.

config.generators do |g|

```
  g.test_framework :rspec, fixtures: true, view_specs:
    false, helper_specs: false, routing_specs: false,
    controller_specs: true, request_specs: false
end
```

```
module Application
  class Application < Rails::Application
    # Initialize configuration defaults for originally generated Rails version.
    config.load_defaults 6.1

    # Configuration for the application, engines, and railties goes here.
    #
    # These settings can be overridden in specific environments using the files
    # in config/environments, which are processed later.
    #
    # config.time_zone = "Central Time (US & Canada)"
    # config.eager_load_paths << Rails.root.join("extras")

    # Don't generate system test files.
    config.generators.system_tests = nil
    config.generators do |g|
      g.test_framework :rspec,
        fixtures: true,
        view_specs: false,
        helper_specs: false,
        routing_specs: false,
        controller_specs: true,
        request_specs: false
    end
  end
end
```

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

require "capybara/rspec"

```
6 require 'fileutils'
7
8 # Prevent database truncation if the environment is production
9 abort("The Rails environment is running in production mode!") if Rails.env.production?
10 require 'rspec/rails'
11 require "capybara/rspec"
12 # Add additional requires below this line. Rails is not loaded until this point!
13
14 # Requires supporting ruby files with custom matchers and macros, etc, in
15 # spec/support/ and its subdirectories. Files matching `spec/**/*_spec.rb` are
16 # run as spec files by default. This means that files in spec/support that end
17 # in _spec.rb will both be required and run as specs, so you should
```

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

\$ rspec

```
lenuelp@Debian12:~/Desktop/GUIA8/App_RoR$ rspec
No examples found.
```

```
Finished in 0.00145 seconds (files took 0.75219 seconds to load)
0 examples, 0 failures
```

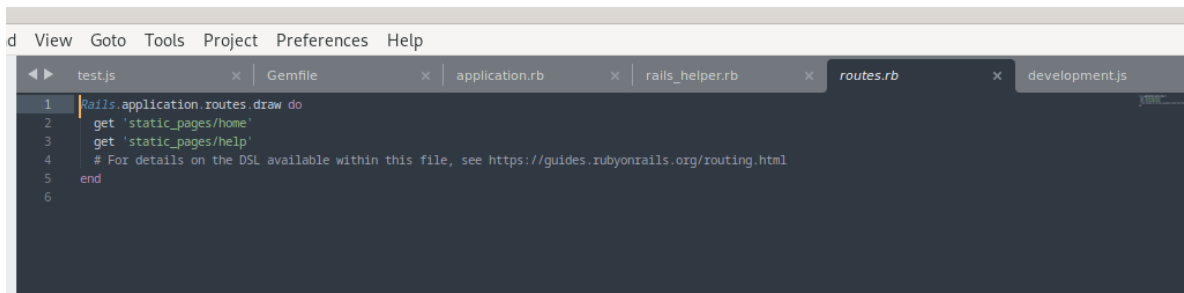
3. Static pages

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

```
$ rails generate controller StaticPages home help
```

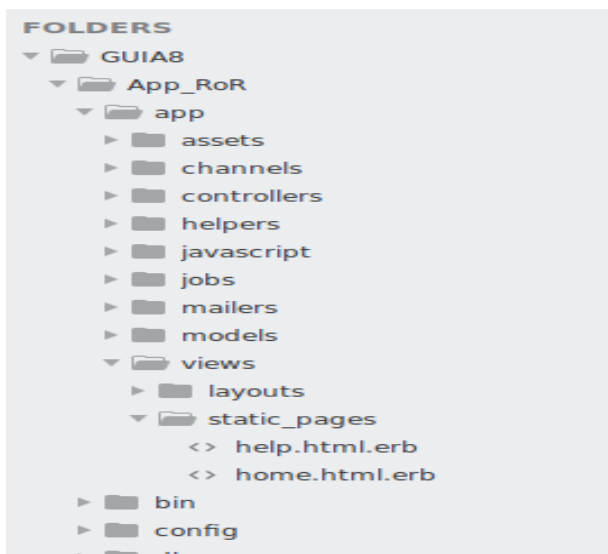
```
lenuelp@Debian12:~/Desktop/GUIA8/App_RoR$ rails generate controller StaticPages
home help
Running via Spring preloader in process 59020
  create  app/controllers/static_pages_controller.rb
  route   get 'static_pages/home'
         get 'static_pages/help'
  invoke  erb
  create  app/views/static_pages
  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
  invoke  assets
  invoke  scss
  create  app/assets/stylesheets/static_pages.scss
lenuelp@Debian12:~/Desktop/GUIA8/App_RoR$
```

3.2. Verificar la configuración del archivo routes.rb.



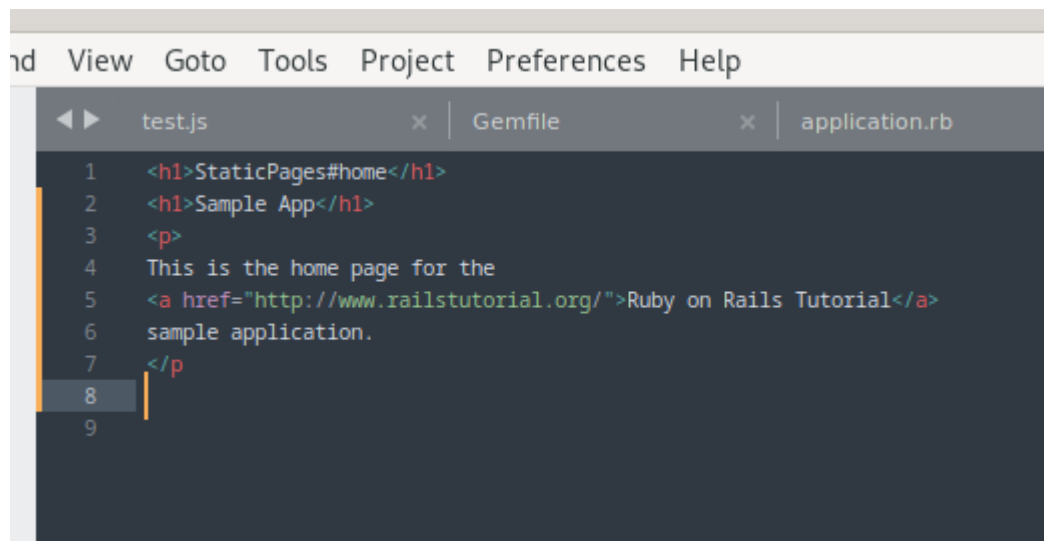
```
1 Rails.application.routes.draw do
2   get 'static_pages/home'
3   get 'static_pages/help'
4   # For details on the DSL available within this file, see https://guides.rubyonrails.org/routing.html
5 end
6
```

3.3. Verificar que se han creado las vistas en **app/views/static_pages/**.



4. Modificar las vistas.

4.1. Agregar al archivo 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
9
```

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

```
test.js  x  Gemfile  x  application.rb  x  rails_h
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
8  Tutorial</em>
9  book</a>.
10 </p>
11
```

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.

```
$ rails db:migrate
```

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

```
$ rspec spec/controllers/static_pages_controller_spec.rb
```

```
Finished in 0.44448 seconds (files took 7.75 seconds to load)
4 examples, 0 failures
```

```
root@debian12: /home/lenuel/Escritorio/guia8/App_RoR#
```

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

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

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

```
$ rspec spec/controllers/static_pages_controller_spec.rb
```

```
Failures:

  1) StaticPagesController GET #home 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:12:in `block (4 levels) in <top (required)>'

Finished in 0.45554 seconds (files took 7.8 seconds to load)
5 examples, 1 failure

Failed examples:

rspec ./spec/controllers/static_pages_controller_spec.rb:11 # StaticPagesController GET #home GET #about returns http success

root@debian12:/home/lenuel/Escritorio/guia8. Software
```

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

```
get 'static_pages/about'
```

```
static_pages_controller_spec.rb x | about.html.erb x | contact.html.erb x | static_pages_controller.rb x | application_helper.rb x | application.html.erb x | routes.rb x | Gemfile x | html.erb

1 Rails.application.routes.draw do
2   get "static_pages/home"
3   get "static_pages/help"
4   get 'static_pages/about'
5   get 'static_pages/contact'
6   root 'static_pages#home'
7
8
9   # Define your application routes per the DSL in https://guides.rubyonrails.org/routing.html
10
11   # Reveal health status on /up that returns 200 if the app boots
12   # Can be used by load balancers and uptime monitors to verify that the app is up
13   get "up" => "rails/health#show", as: :rails_health_check
14
15   # Render dynamic PWA files from app/views/pwa/*
16   get "pwa/:manifest" => "rails/pwa#render_manifest", as: :rails_pwa_manifest
17 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.

```
Failures:

  1) StaticPagesController GET #home 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:12:in 'block (4 levels) in <top (required)>'

Finished in 0.45235 seconds (files took 7.59 seconds to load)
3 examples, 1 failure

Failed examples:

rspec ./spec/controllers/static_pages_controller_spec.rb:11 # StaticPagesController GET #home GET #about returns http success

root@debian12: /home/lenuel/Escritorio/guia8/App_RoR#
```

5.7. Agregar el método en el archivo `controllers/static_pages_controller.rb` y luego ejecutar el test.

```
def about
end
```

```
static_pages_controller_spec.rb | about.html.erb | contact.html.erb | static_pages_controller.rb | application_helper.rb | application.html.erb | routes.rb | Gemfile
1 class StaticPagesController < ApplicationController
2   def about
3   end
4 end
5
6
```

```
Failures:

  1) StaticPagesController GET #home 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:12:in 'block (4 levels) in <top (required)>'

Finished in 0.47078 seconds (files took 7.52 seconds to load)
3 examples, 1 failure

Failed examples:

rspec ./spec/controllers/static_pages_controller_spec.rb:11 # StaticPagesController GET #home GET #about returns http success

root@debian12: /home/lenuel/Escritorio/guia8/App_RoR#
```

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.

```
static_pages_controller_spec.rb x about.html.erb x contact.html.erb x static_pages_controller.rb x application_helper.rb x application.html.erb x routes.rb x home
1 <% provide(:title, "About") %>
2
3 <h1>About</h1>
4 <p>
5   The <a href="http://www.railstutorial.org/"><em>Ruby on Rails
6   Tutorial</em></a> is a
7   <a href="http://www.railstutorial.org/book">book</a> and
8   <a href="http://screencasts.railstutorial.org/">screencast
9   series</a>
10  to teach web development with
11  <a href="http://rubyonrails.org/">Ruby on Rails</a>.
12  This is the sample application for the tutorial.
13 </p>
```

5.9.

Para comprobar que todo funciona, ejecutar el test nuevamente.

```
.....
Finished in 0.46677 seconds (files took 8.32 seconds to load)
5 examples, 0 failures

root@debian12: /home/lenuel/Escritorio/guia8/App_RoR#
```

6. Configurar el título de la página.

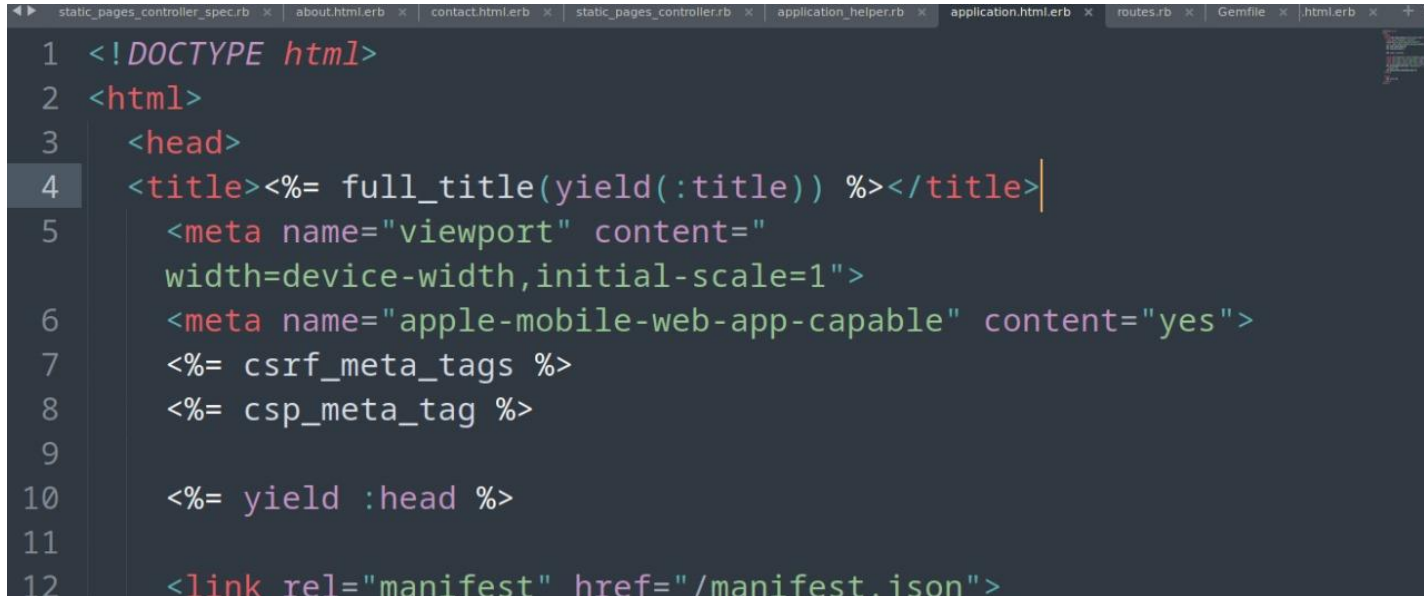
6.1. Para que el título de la página cambie por cada una de las vistas a la que se accede en la aplicación, copiar el código en el archivo **app/helpers/application_helper.rb** dentro del módulo.

```
static_pages_controller_spec.rb x about.html.erb x contact.html.erb x static_pages_controller.rb x application_helper.rb x application.html.erb
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.

```
<title><%= full_title(yield(:title)) %></title>
```

Deberá quedar a como se muestra en la figura .

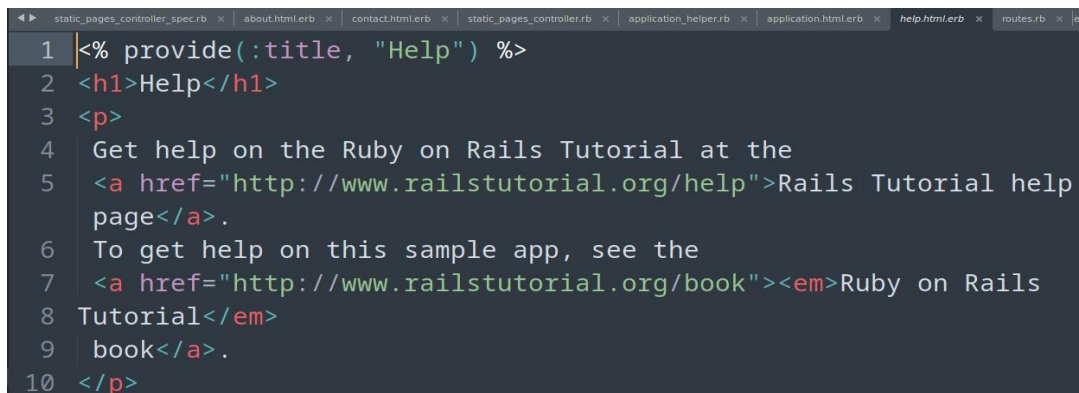


```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title><%= full_title(yield(:title)) %></title>
5     <meta name="viewport" content="
6       width=device-width,initial-scale=1">
7     <meta name="apple-mobile-web-app-capable" content="yes">
8     <%= csrf_meta_tags %>
9     <%= csp_meta_tag %>
10    <%= yield :head %>
11
12    <link rel="manifest" href="/manifest.json">
```

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.

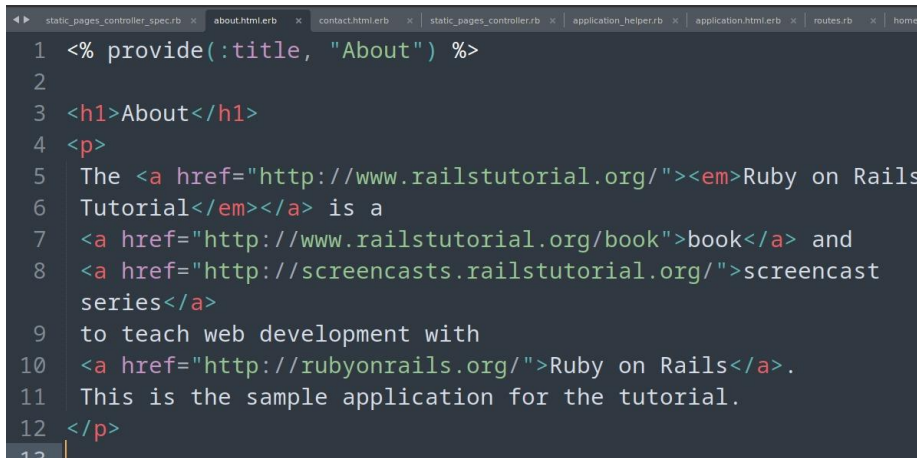
```
<%= provide(:title, "Help") %>
```



```
1 <%= provide(:title, "Help") %>
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
6   page</a>.
7   To get help on this sample app, see the
8   <a href="http://www.railstutorial.org/book"><em>Ruby on Rails
9   Tutorial</em>
10  book</a>.
11 </p>
```

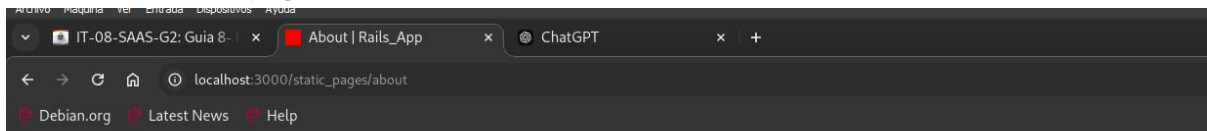
7.2. En el archivo **about.html.erb**, agregar la siguiente línea en la parte superior de todo el código.

```
<% provide(:title, "About") %>
```



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

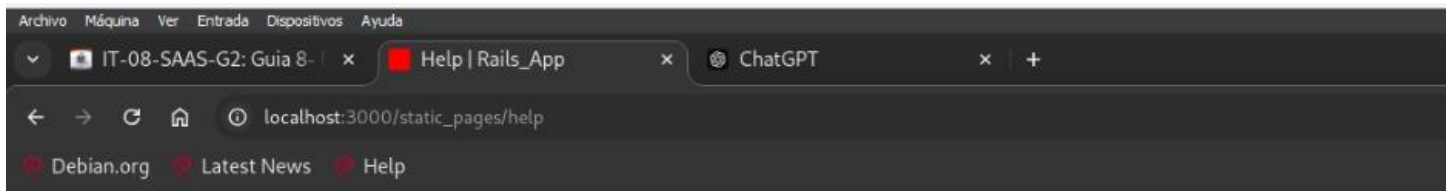
7.3. Navegar en la dirección http://localhost:3000/static_pages/about y verificar que cambia el título de la página.



About

The [Ruby on Rails Tutorial](http://www.rubyonrails.org/) is a [book](http://www.railstutorial.org/book) and [screencast series](http://screencasts.railstutorial.org/) to teach web development with [Ruby on Rails](http://rubyonrails.org/). This is the sample application for the tutorial.

7.4. Navegar en la dirección http://localhost:3000/static_pages/help y verificar que cambia el título de la página.



Help

Get help on the Ruby on Rails Tutorial at the [Rails Tutorial help page](http://www.rails-tutorial.org/help). To get help on this sample app, see the [Ruby on Rails Tutorial book](http://www.rubyonrails.org/).

Ejercicios propuestos para ser entregados al docente

1. Realizar y analizar cada uno de los enunciados de la guía.
2. Copiar el código del recuadro, en el archivo `static_pages_controller_spec.rb` del directorio `spec/controllers/`.

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

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

```
Failures:

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

     ActionController::UrlGenerationError:
       No route matches {:action=>"contact", :controller=>"static_pages"}
     # ./spec/controllers/static_pages_controller_spec.rb:6:in `block (3 levels) in <top (required)>'

Finished in 0.47816 seconds (files took 7.55 seconds to load)
6 examples, 1 failure

Failed examples:

rspec ./spec/controllers/static_pages_controller_spec.rb:5 # StaticPagesController GET #contact returns http success

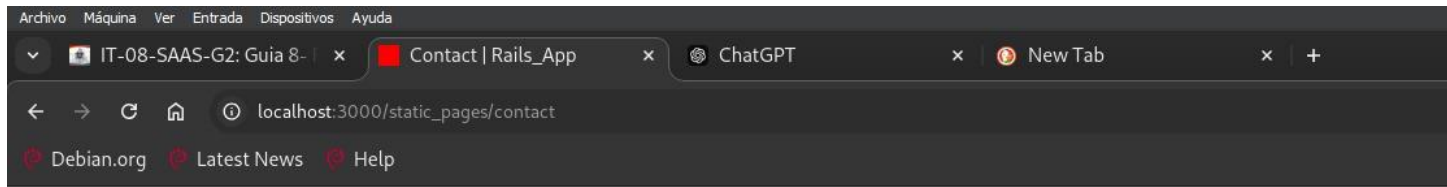
root@debian12:/home/lenuel/Escritorio/guia8/App_RoR#
```

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

```
Finished in 0.51126 seconds (files took 7.55 seconds to load)
6 examples, 0 failures

root@debian12:/home/lenuel/Escritorio/guia8/App_RoR#
```

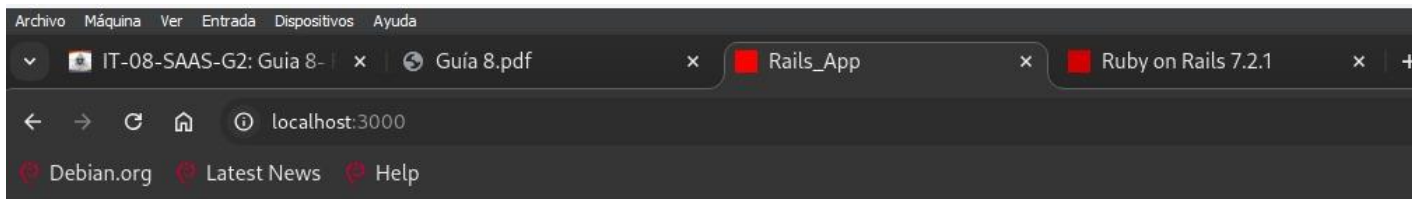
5. Navegar en la dirección `http://localhost:3000/static_pages/contact`.



Contact

The [Ruby on Rails Tutorial](#) is a [book](#) and [screencast series](#) to teach web development with [Ruby on Rails](#). This is the sample application for the tutorial.

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.



StaticPages#home

Find me in `app/views/static_pages/home.html.erb`