

Universidad Nacional Autónoma de Nicaragua Unan León



Facultad de Ciencias y Tecnología.

Componente: Software como un servicio

Tema: Practica7

Grupo:1

Docente: Erving Montes

Integrante:

-Harold Steven Garcia Ramos.

Fecha: 25 de septiembre de 2024

1. Crear un nuevo proyecto en Rails.

```
harold@debian: ~/Guia7
harold@debian:~/Guia7$ su
Contraseña:
root@debian:/home/harold/Guia7# rails new App_mysql -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/Guia7/App_mysql/.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
```

2. Preparar el escenario de Test. En esta sección se trabajará con la gema rspec y shoulda-matcher para la parte de los tests; para esto, lo primero es configurar el escenario de la aplicación.

2.1 Abrir el archivo Gemfile y agregar las siguientes líneas al grupo development, test

```
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 ]
end
gem 'rspec-rails'
gem 'shoulda-matchers'
gem 'capybara'
█
group :development do
  # Use console on exceptions pages [https://github.com/rails/web-console]
  gem "web-console"

  # Add speed badges [https://github.com/MiniProfiler/rack-mini-profiler]

```

^G Ayuda	^O Guardar	^W Buscar	^K Cortar	^T Ejecutar	^C Ubicación
^X Salir	^R Leer fich.	^_ Reemplazar	^U Pegar	^J Justificar	^_ Ir a línea

2.2 Ejecutar el siguiente el comando en el terminal, para instalar las gemas agregadas anteriormente.

```
harold@debian: ~/Guia7
root@debian:/home/harold/Guia7/App_mysql# 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 rake 13.2.1
Using base64 0.2.0
Using bigdecimal 3.1.8
Using concurrent-ruby 1.3.3
Using connection_pool 2.4.1
Using drb 2.2.1
Using i18n 1.14.5
Using minitest 5.24.1
Using mutex_m 0.2.0
Using tzinfo 2.0.6
Using activesupport 7.1.3.4
Using builder 3.3.0
Using erubi 1.13.0
Using racc 1.8.1
Using nokogiri 1.16.7 (x86_64-linux)
Using rails-dom-testing 2.2.0
Using crass 1.0.6
Using loofah 2.22.0
```

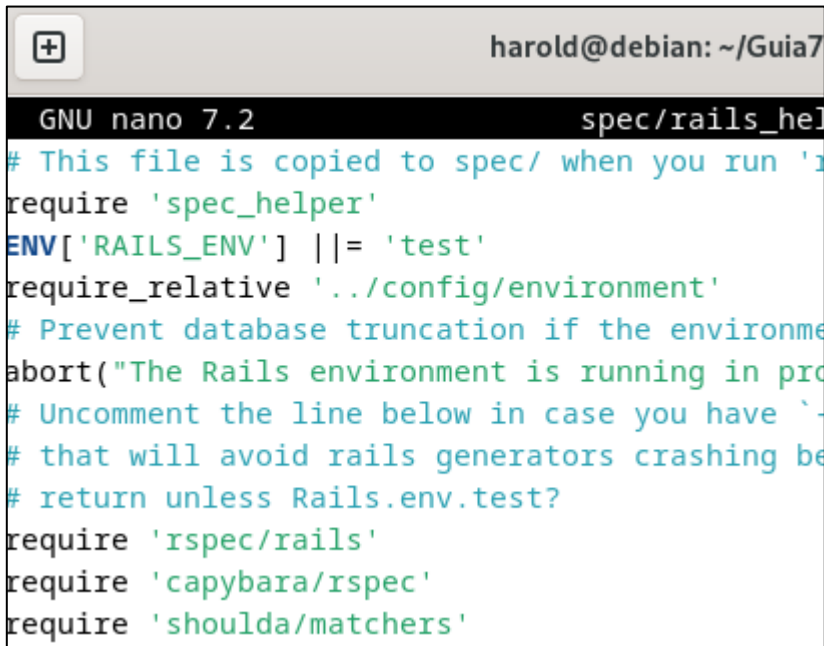
2.3 Ejecutar el siguiente comando en el terminal, para crear la base de datos.

```
harold@debian: ~/Guia7
root@debian:/home/harold/Guia7/App_mysql# rake db:create:all
Database 'storage/development.sqlite3' already exists
Database 'storage/test.sqlite3' already exists
Database 'storage/production.sqlite3' already exists
root@debian:/home/harold/Guia7/App_mysql#
```

2.4 Ejecutar el comando en el terminal, para generar los archivos de rspec.

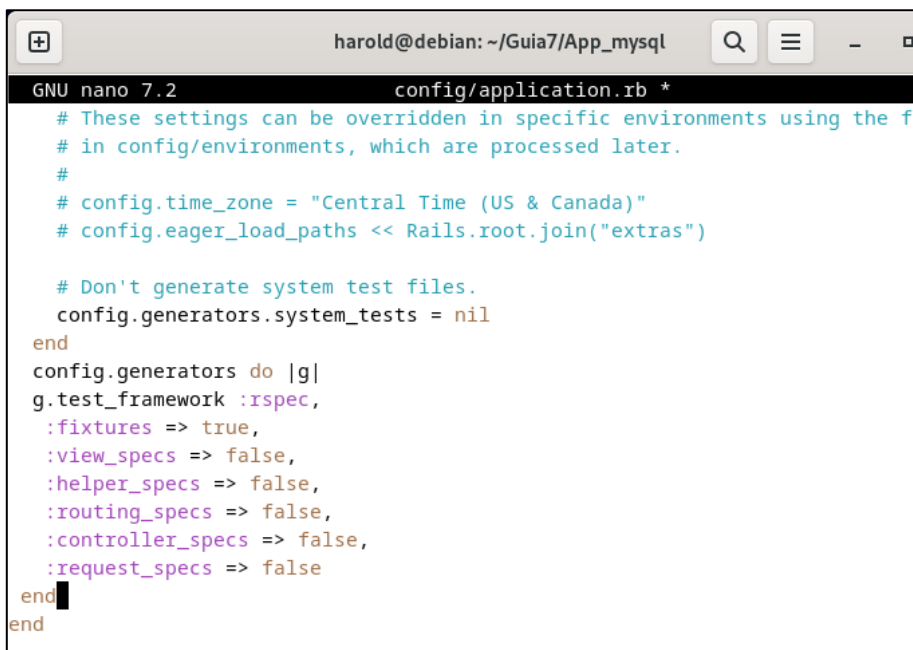
```
root@debian:/home/harold/Guia7/App_mysql# rails g rspec:install
create .rspec
create spec
create spec/spec_helper.rb
create spec/rails_helper.rb
root@debian:/home/harold/Guia7/App_mysql#
```

2.5 Como parte de la configuración hay que agregar las siguientes líneas dentro del archivo spec/rails_helper.rb que fue generado anteriormente.



```
harold@debian: ~/Guia7
GNU nano 7.2 spec/rails_helper.rb
# This file is copied to spec/ when you run 'rails generate rspec:install'
require 'spec_helper'
ENV['RAILS_ENV'] ||= 'test'
require_relative '../config/environment'
# Prevent database truncation if the environment is production
abort("The Rails environment is running in production mode!") if ENV['RAILS_ENV'] == 'production'
# Uncomment the line below in case you have `--skip-action-cable` in your Rails application's Gemfile
# require_relative '../config/action_cable'
# return unless Rails.env.test?
require 'rspec/rails'
require 'capybara/rspec'
require 'shoulda/matchers'
```

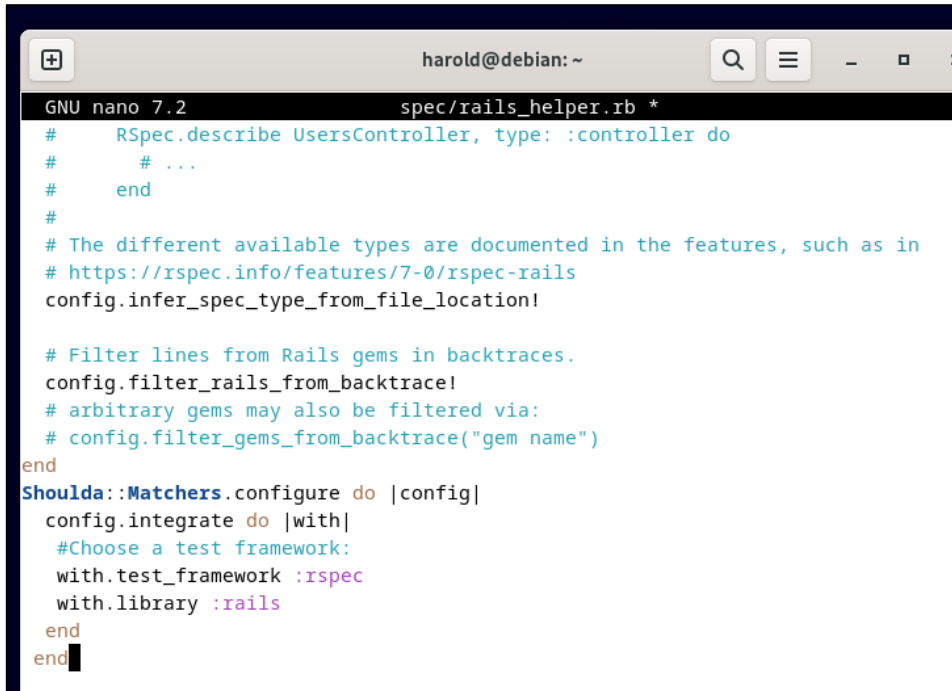
2.6 Como en esta guía solo se harán test para los modelos. Abrir el archivo Application.rb que está dentro del directorio config y agregar el siguiente código dentro de la clase applications.



```
harold@debian: ~/Guia7/App_mysql
GNU nano 7.2 config/application.rb *
# These settings can be overridden in specific environments using the files config/environments/*
# 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
end
config.generators do |g|
  g.test_framework :rspec,
    :fixtures => true,
    :view_specs => false,
    :helper_specs => false,
    :routing_specs => false,
    :controller_specs => false,
    :request_specs => false
end
end
```

2.7 Por ultimo solo queda escribir en siguiente código en el archivo `/spec/rails_helper.rb` para poder trabajar con la gema `shoulda-matcher` al momento de crear los tests. Agregarlo al final de todo el código que contiene el archivo.

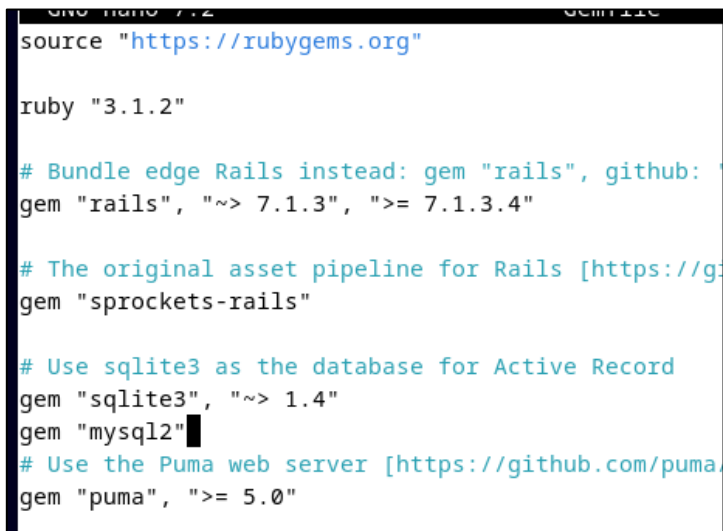


```
harold@debian: ~
GNU nano 7.2 spec/rails_helper.rb *
# RSpec.describe UsersController, type: :controller do
#   # ...
#   end
#
# The different available types are documented in the features, such as in
# https://rspec.info/features/7-0/rspec-rails
config.infer_spec_type_from_file_location!

# Filter lines from Rails gems in backtraces.
config.filter_rails_from_backtrace!
# arbitrary gems may also be filtered via:
# config.filter_gems_from_backtrace("gem name")
end
Shoulda::Matchers.configure do |config|
  config.integrate do |with|
    #Choose a test framework:
    with.test_framework :rspec
    with.library :rails
  end
end
```

3.Trabajando con mysql El framework RoR trabaja por defecto con `sqlite3` para almacenar los datos, pero también permite trabajar con distintos gestores de base de datos, en este caso se trabajará con `mysql`.

3.1 Abrir el archivo `Gemfile` y agregar la gema '`mysql2`'.



```
GNU nano 7.2 Gemfile
source "https://rubygems.org"

ruby "3.1.2"

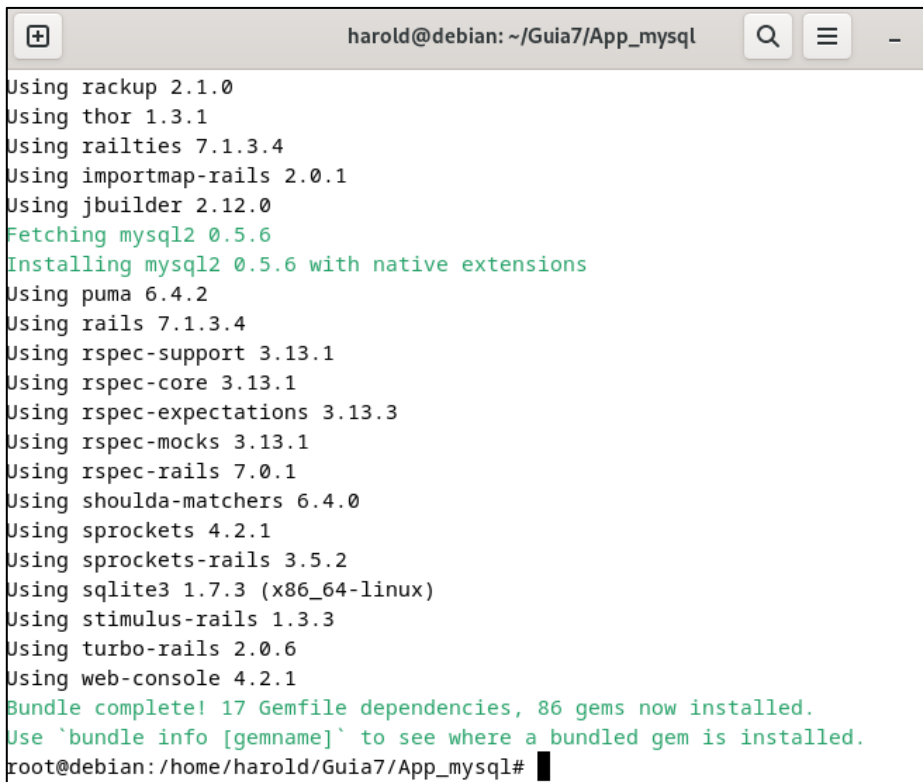
# Bundle edge Rails instead: gem "rails", github: "rails/rails"
gem "rails", "~> 7.1.3", ">= 7.1.3.4"

# The original asset pipeline for Rails [https://github.com/rails/sprockets-rails]
gem "sprockets-rails"

# Use sqlite3 as the database for Active Record
gem "sqlite3", "~> 1.4"
gem "mysql2"

# Use the Puma web server [https://github.com/puma/puma]
gem "puma", ">= 5.0"
```

3.2 Desde el terminal instalar la gema de mysql.



```
harold@debian: ~/Guia7/App_mysql
Using rackup 2.1.0
Using thor 1.3.1
Using railties 7.1.3.4
Using importmap-rails 2.0.1
Using jbuilder 2.12.0
Fetching mysql2 0.5.6
Installing mysql2 0.5.6 with native extensions
Using puma 6.4.2
Using rails 7.1.3.4
Using rspec-support 3.13.1
Using rspec-core 3.13.1
Using rspec-expectations 3.13.3
Using rspec-mocks 3.13.1
Using rspec-rails 7.0.1
Using shoulda-matchers 6.4.0
Using sprockets 4.2.1
Using sprockets-rails 3.5.2
Using sqlite3 1.7.3 (x86_64-linux)
Using stimulus-rails 1.3.3
Using turbo-rails 2.0.6
Using web-console 4.2.1
Bundle complete! 17 Gemfile dependencies, 86 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
root@debian: /home/harold/Guia7/App_mysql#
```

4. Configurar la base de datos. Modificar el archivo config/database.yml para que quede de la siguiente manera, en la sección development.

```
#
default: &default
  adapter: mysql2
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
  timeout: 5000

development:
  <<: *default
  adapter: mysql2
  username: root
  password: harold123
  host: 127.0.0.1
  port: 3306
  database: RoR

# Warning: The database defined as "test" will be erased and
# re-generated from your development database when you run "rake".
# Do not set this db to the same as development or production.
test:
  <<: *default
  username: root
  password: harold123
```

5. Crear base de datos

5.1 Generar la base de dato. Se tiene que diferenciar que es lo que se quiere hacer, si crear una nueva base de datos o migrar los datos de un motor de base de datos a otro.

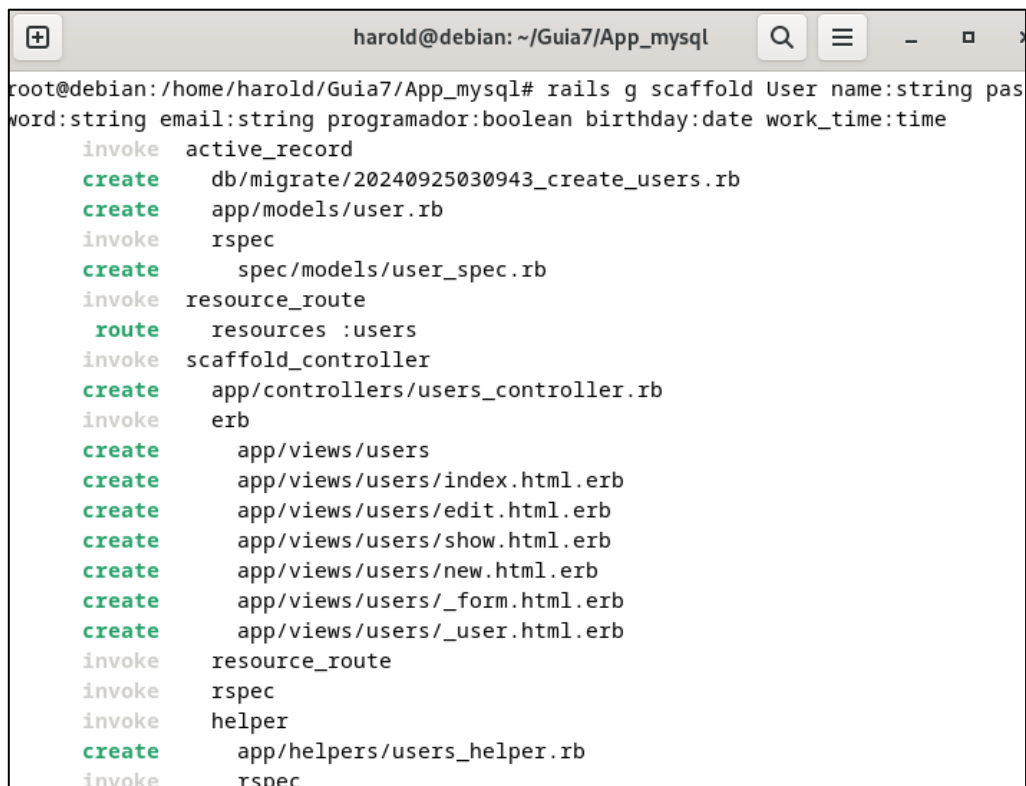
5.2 Crear nueva base de datos.

```
Created database 'RoR'  
Created database 'RoR_test'  
root@debian:/home/harold/Guia7/App_mysql#
```

5.3 Migrar los datos, para ello la base de datos ya debe existir.

```
root@debian:/home/harold/Guia7/App_mysql# rake db:migrate  
root@debian:/home/harold/Guia7/App_mysql#
```

6. Para probar el comportamiento se está usando scaffold para generar código.



```
harold@debian: ~/Guia7/App_mysql  
root@debian:/home/harold/Guia7/App_mysql# rails g scaffold User name:string password:string email:string programador:boolean birthday:date work_time:time  
  invoke  active_record  
    create  db/migrate/20240925030943_create_users.rb  
    create  app/models/user.rb  
  invoke  rspec  
    create  spec/models/user_spec.rb  
  invoke  resource_route  
    route  resources :users  
  invoke  scaffold_controller  
    create  app/controllers/users_controller.rb  
  invoke  erb  
    create  app/views/users  
    create  app/views/users/index.html.erb  
    create  app/views/users/edit.html.erb  
    create  app/views/users/show.html.erb  
    create  app/views/users/new.html.erb  
    create  app/views/users/_form.html.erb  
    create  app/views/users/_user.html.erb  
  invoke  resource_route  
  invoke  rspec  
  invoke  helper  
    create  app/helpers/users_helper.rb  
  invoke  rspec
```

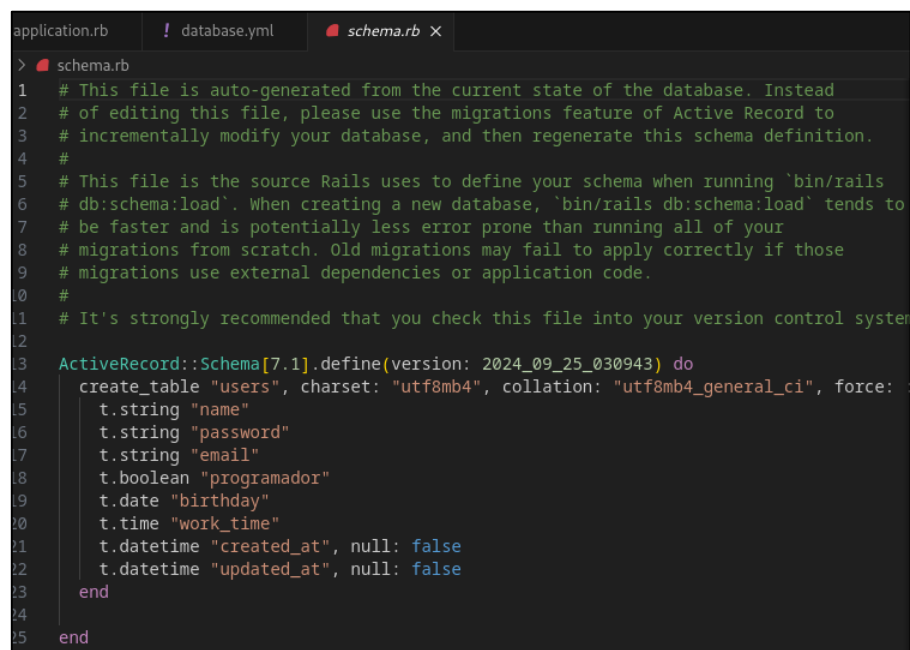
7. Migrar los datos generados con scaffold y la base de datos de los tests

```
root@debian:/home/harold/Guia7/App_mysql# rails db:migrate RAILS_ENV=test
== 20240925030943 CreateUsers: migrating =====
-- create_table(:users)
   -> 0.0217s
== 20240925030943 CreateUsers: migrated (0.0220s) =====
root@debian:/home/harold/Guia7/App_mysql#
```

8. Migrar los datos de la parte de desarrollo.

```
root@debian:/home/harold/Guia7/App_mysql# rails db:migrate RAILS_ENV=development
== 20240925030943 CreateUsers: migrating =====
-- create_table(:users)
   -> 0.0279s
== 20240925030943 CreateUsers: migrated (0.0281s) =====
root@debian:/home/harold/Guia7/App_mysql#
```

9. Ir al directorio db y abrir el archivo schema.rb que se ha generado anteriormente, el cual contiene el esquema actual de la base de datos donde se almacenarán los datos del usuario.



```
application.rb  ! database.yml  schema.rb X
> schema.rb
1  # This file is auto-generated from the current state of the database. Instead
2  # of editing this file, please use the migrations feature of Active Record to
3  # incrementally modify your database, and then regenerate this schema definition.
4  #
5  # This file is the source Rails uses to define your schema when running `bin/rails
6  # db:schema:load`. When creating a new database, `bin/rails db:schema:load` tends to
7  # be faster and is potentially less error prone than running all of your
8  # migrations from scratch. Old migrations may fail to apply correctly if those
9  # migrations use external dependencies or application code.
10 #
11 # It's strongly recommended that you check this file into your version control system
12
13 ActiveRecord::Schema[7.1].define(version: 2024_09_25_030943) do
14   create_table "users", charset: "utf8mb4", collation: "utf8mb4_general_ci", force: :cascade do |t|
15     t.string "name"
16     t.string "password"
17     t.string "email"
18     t.boolean "programador"
19     t.date "birthday"
20     t.time "work_time"
21     t.datetime "created_at", null: false
22     t.datetime "updated_at", null: false
23   end
24 end
```


10. Si observa, al momento de generar el código con scaffold, se ha creado un directorio models dentro del directorio spec, verificar como se muestra a continuación.

```
root@debian:/home/harold/Guia7/App_mysql# ls spec/  
models rails_helper.rb spec_helper.rb  
root@debian:/home/harold/Guia7/App_mysql#
```

10.1 Dentro de este directorio model se ha creado el archivo user_spec.rb que se necesita para poder hacer test a los modelos. Eliminar o comentar la siguiente línea, que no se ocupará.

```
spec > models > user_spec.rb  
1  require 'rails_helper'  
2  
3  RSpec.describe User, type: :model do  
4    #pending "add some examples to (or delete) #{__FILE__}"  
5  end  
6
```

11. Para empezar a crear los tests y validar el modelo user.

11.1 Agregar el código al archivo spec/models/user_spec.rb para hacer test al campo name y agregar validaciones.

```
application.rb  ! database.yml  user_spec.rb  
spec > models > user_spec.rb  
1  require 'rails_helper'  
2  
3  RSpec.describe User, type: :model do  
4    #pending "add some examples to (or delete) #{__FILE__}"  
5    context "name" do  
6      it {should validate_presence_of :name}  
7      it {should validate_length_of(:name).is_at_most(15) }  
8      it {should_not allow_value("123123").for(:name) }  
9      it {should allow_value("abcd").for(:name) }  
10   end  
11 end  
12
```

11.2 Ir al terminal y ejecutar el siguiente comando para hacer los tests.

```
harold@debian: ~/Guia7/App_mysql
root@debian: /home/harold/Guia7/App_mysql# rspec spec/models/user_spec.rb
FFF.

Failures:

  1) User name is expected to validate that :name cannot be empty/falsy
     Failure/Error: it {should validate_presence_of :name}

     Expected User to validate that :name cannot be empty/falsy, but this
     could not be proved.
     After setting :name to <" ">, the matcher expected the User to be
     invalid, but it was valid instead.
     # ./spec/models/user_spec.rb:6:in `block (3 levels) in <top (required)>'

  2) User name is expected to validate that the length of :name is at most 15
     Failure/Error: it {should validate_length_of(:name).is_at_most(15) }

     Expected User to validate that the length of :name is at most 15, but
     this could not be proved.
     After setting :name to <"xxxxxxxxxxxxxxxx">, the matcher expected the
     User to be invalid, but it was valid instead.
     # ./spec/models/user_spec.rb:7:in `block (3 levels) in <top (required)>'

  3) User name is expected not to allow :name to be <"123123">
```

12. Ir al archivo user.rb que se encuentra en el directorio app/model y agregar el siguiente código dentro de la clase.

```
nal Help
application.rb database.yml user_spec.rb user.rb
app > models > user.rb
1 class User < ApplicationRecord
2   validates_presence_of :name, presence: true
3   validates_format_of :name, :with=> /\A[a-zA-Z]+\z/
4   validates :name, length: {maximum: 15 }, on: :create
5 end
6
```

12. Si ejecuta el comando de los tests nuevamente, podrá observar que ya no se muestran los mensajes en rojo.

```
harold@debian: ~/Guia7/App_mysql
root@debian: /home/harold/Guia7/App_mysql# rspec spec/models/user_spec.rb
.....

Finished in 0.1555 seconds (files took 2.35 seconds to load)
4 examples, 0 failures
```

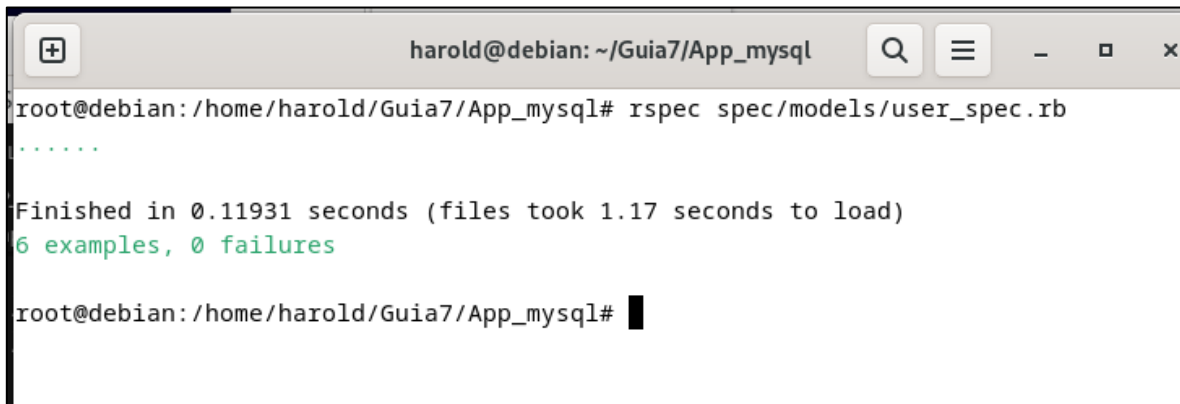
13. Agregar siguiente código al archivo spec/models/user_spec.rb para hacer test al campo password del modelo.

```
spec > models > user_spec.rb
1  require 'rails_helper'
2
3  RSpec.describe User, type: :model do
4    #pending "add some examples to (or delete) #{__FILE__}"
5    context "name" do
6      it {should validate_presence_of :name}
7      it {should validate_length_of(:name).is_at_most(15) }
8      it {should_not allow_value("123123").for(:name) }
9      it {should allow_value("abcd").for(:name) }
10   end
11   context "Password" do
12     it {should validate_presence_of :password }
13     it {should validate_length_of(:password).is_equal_to(8) }
14   end
15 end
16
```

14. Agregar las validaciones en el archivo app/model/user.rb, correspondientes para el campo password del modelo.

```
models > user.rb
class User < ApplicationRecord
  validates_presence_of :name, presence: true
  validates_format_of :name, :with=> /\A[a-zA-Z]+\z/
  validates :name, length: {maximum: 15 }, on: :create
  validates_presence_of :password, presence: true
  validates_length_of :password, is: 8
end
```

14.1 Una vez escrito el código, volver a ejecutar el test para poder observar que ahora muestra el mensaje en verde, indicando que ha pasado los test a como se muestra en la siguiente figura.

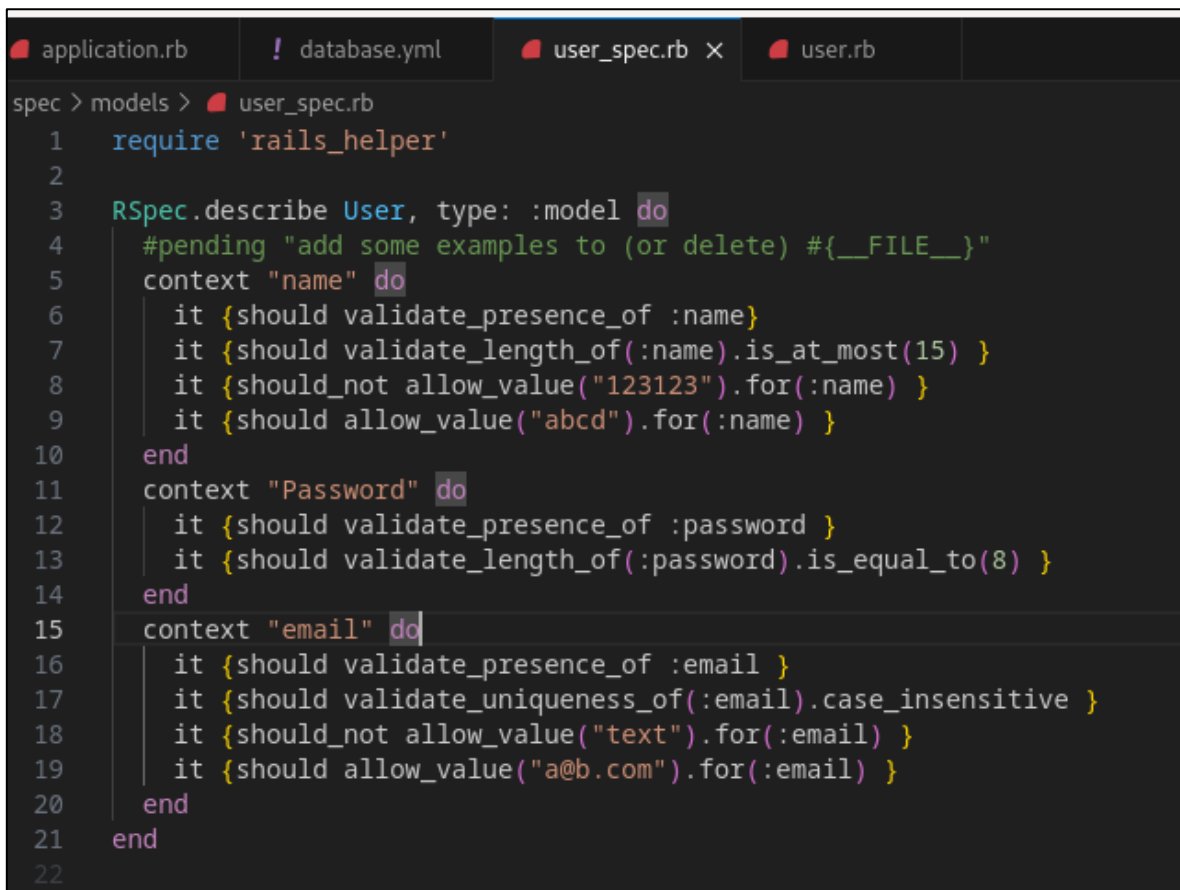


```
harold@debian: ~/Guia7/App_mysql
root@debian: /home/harold/Guia7/App_mysql# rspec spec/models/user_spec.rb
.....

Finished in 0.11931 seconds (files took 1.17 seconds to load)
6 examples, 0 failures

root@debian: /home/harold/Guia7/App_mysql#
```

15. El último test será para el campo email del modelo, agregar el código al archivo spec/models/user_spec.rb.



```
application.rb  ! database.yml  user_spec.rb x  user.rb
spec > models > user_spec.rb
1  require 'rails_helper'
2
3  RSpec.describe User, type: :model do
4    #pending "add some examples to (or delete) #{__FILE__}"
5    context "name" do
6      it {should validate_presence_of :name}
7      it {should validate_length_of(:name).is_at_most(15) }
8      it {should_not allow_value("123123").for(:name) }
9      it {should allow_value("abcd").for(:name) }
10   end
11   context "Password" do
12     it {should validate_presence_of :password }
13     it {should validate_length_of(:password).is_equal_to(8) }
14   end
15   context "email" do
16     it {should validate_presence_of :email }
17     it {should validate_uniqueness_of(:email).case_insensitive }
18     it {should_not allow_value("text").for(:email) }
19     it {should allow_value("a@b.com").for(:email) }
20   end
21 end
22
```

16. Ejecutar el test en el terminal y mostrará los mensajes en rojo nuevamente.

```
harold@debian: ~/Guia7/App_mysql
root@debian: /home/harold/Guia7/App_mysql# rspec spec/models/user_spec.rb
.....FFF.

Failures:

  1) User email is expected to validate that :email cannot be empty/falsy
     Failure/Error: it {should validate_presence_of :email }

     Expected User to validate that :email cannot be empty/falsy, but this
     could not be proved.
     After setting :email to <"">, the matcher expected the User to be
     invalid, placing a validation error on :email. The Example was
     invalid, but it had errors involving other attributes:

       * name: ["can't be blank", "is invalid"]
       * password: ["can't be blank", "is the wrong length (should be 8
         characters)"]
     # ./spec/models/user_spec.rb:16:in `block (3 levels) in <top (required)>'

  2) User email is expected to validate that :email is case-sensitively unique
     Failure/Error: it {should validate_uniqueness_of :email }

     Expected User to validate that :email is case-sensitively unique, but
     this could not be proved.
```

17. Agregar las validaciones al modelo user en el archivo app/model/user.rb.

```
application.rb  database.yml  user_spec.rb  user.rb  x
app > models > user.rb
1  class User < ApplicationRecord
2    validates_presence_of :name, presence: true
3    validates_format_of :name, :with=> /\A[a-zA-Z]+\z/
4    validates :name, length: {maximum: 15 }, on: :create
5    validates_presence_of :password, presence: true
6    validates_length_of :password, is: 8
7    validates_presence_of :email, presence: true
8    validates_uniqueness_of :email
9    validates_format_of :email, with: /@/
10 end
11
```

```
harold@debian: ~/Guia7/App_mysql
root@debian:/home/harold/Guia7/App_mysql# rspec spec/models/user_spec.rb
.....

Finished in 0.09992 seconds (files took 1.31 seconds to load)
10 examples, 0 failures

root@debian:/home/harold/Guia7/App_mysql#
```

18. Hasta el momento no se ha trabajado con la parte web del framework; como se sabe, cuando se utiliza scaffold para generar el código, este ya ha generado las vistas para el modelo user.

18.1 Ir al directorio view y observar que existen todas las vistas necesarias para el correcto funcionamiento del modelo user.

```
> models
  > views
    > layouts
  > users
    <> _form.html.erb
    <> _user.html.erb
    ≡ _user.json.jbuilder
    <> edit.html.erb
    <> index.html.erb
    ≡ index.json.jbuilder
    <> new.html.erb
    <> show.html.erb
    ≡ show.json.jbuilder
  > bin
```

18.2 El siguiente paso es configurar el index de la aplicación, para eso ir al archivo config/routes.rb y escribir el código para el index sea la página principal del proyecto. El archivo routes quedará como se muestra en la figura 82.

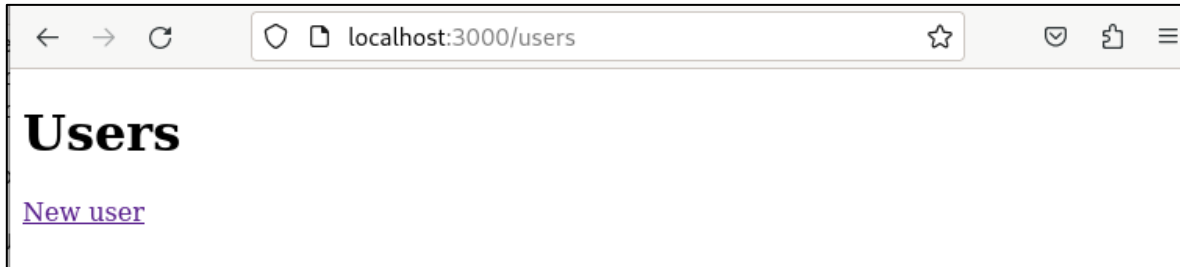
```
config > routes.rb
1  Rails.application.routes.draw do
2    root 'users#index'
3    resources :users
4    # Define your application routes per the DSL in https://guides.rubyonrails.org
5
6    # Reveal health status on /up that returns 200 if the app boots with no
7    # Can be used by load balancers and uptime monitors to verify that the
8    get "up" => "rails/health#show", as: :rails_health_check
9
10   # Defines the root path route ("/")
11   # root "posts#index"
12 end
```

19. Para probar la aplicación, debe arrancar el servidor en el terminal, ejecutando el siguiente comando

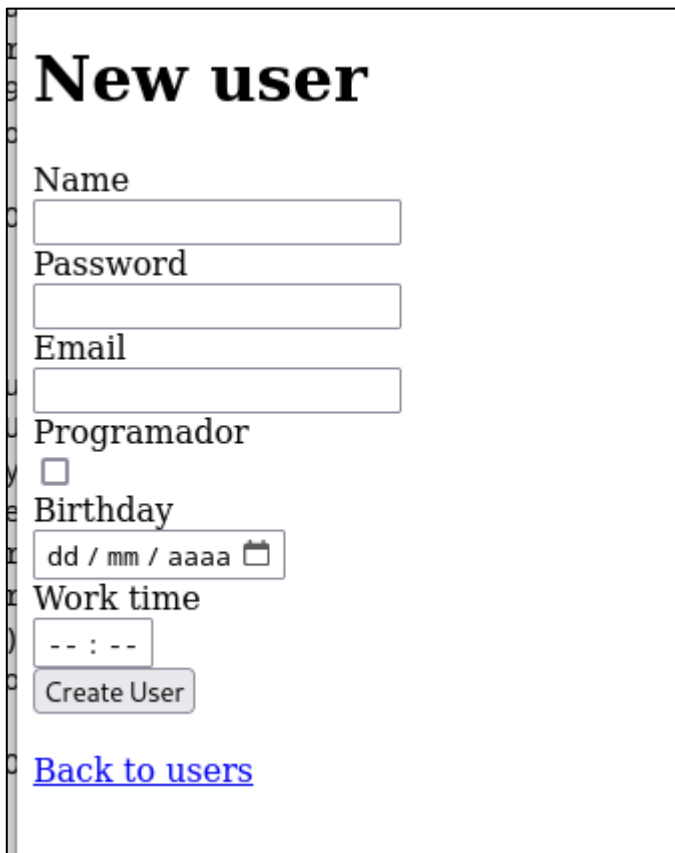
```
root@debian:/home/harold/Guia7/App_mysql# rails s
=> Booting Puma
=> Rails 7.1.4 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 6.4.3 (ruby 3.1.2-p20) ("The Eagle of Durango")
* Min threads: 5
* Max threads: 5
* Environment: development
* PID: 10962
* Listening on http://127.0.0.1:3000
* Listening on http://[::1]:3000
Use Ctrl-C to stop
```

20.Prueba en la aplicación web

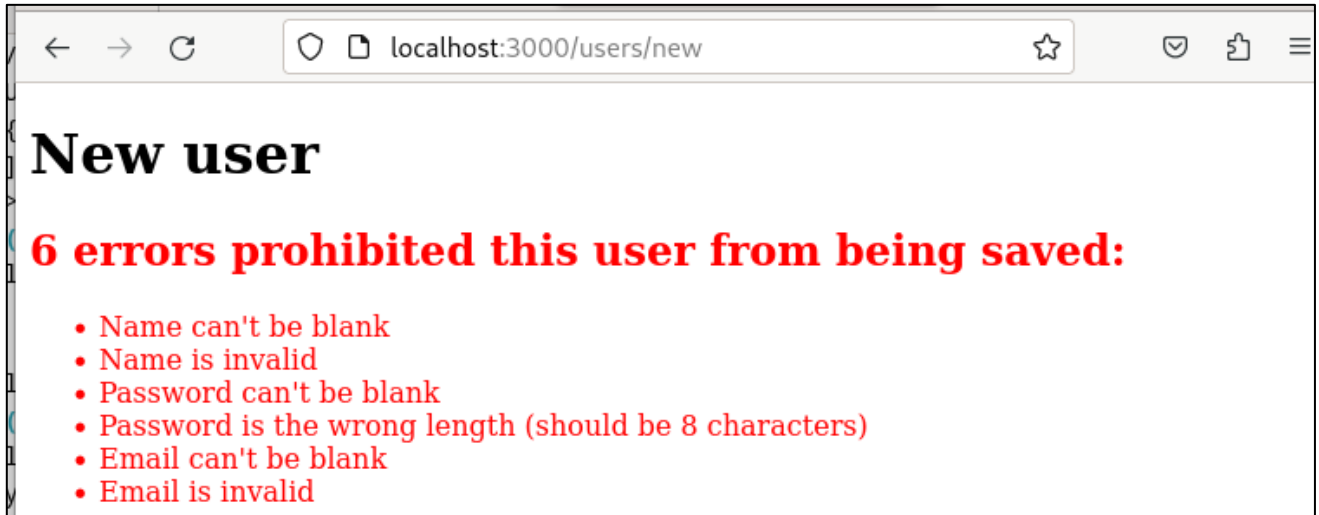
20.1Abrir el navegador en la dirección localhost:3000



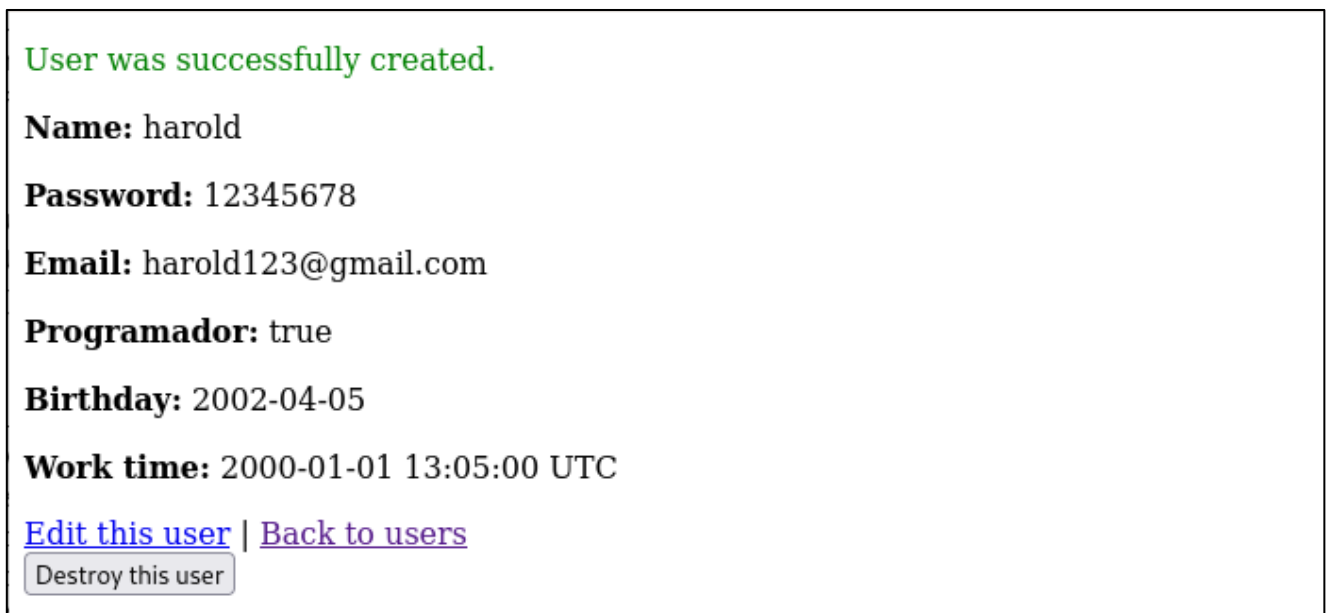
20.2Para crear nuevo usuario y agregar datos, presionar click en New User y mostrará el formulario para ingresar los datos.

A screenshot of a web form titled 'New user' in a large, bold, black font. The form contains several input fields and a checkbox. The fields are labeled 'Name', 'Password', 'Email', 'Programador', 'Birthday', and 'Work time'. The 'Programador' field has a checkbox. The 'Birthday' field has a date input format 'dd / mm / aaaa' and a calendar icon. The 'Work time' field has a time input format '-- : --'. Below the fields is a 'Create User' button. At the bottom of the form, there is a link 'Back to users' in a blue font.

20.3 Una vez abierto el formulario, se podrá poner a prueba las validaciones que se agregaron anteriormente, si presiona el botón create user observará que muestra unos mensajes en relación a las validaciones de los campos del modelo user. Estos mensajes son generados automáticamente por el framework.



20.4 Al llenar los datos del formulario correctamente, ya permite guardar los datos.



20.5 Verificar que se haya creado correctamente el usuario en mysql, abrir el phpmyadmin y hacer un select a la tabla user, de la base de datos en la que se está trabajando.

```
MariaDB [RoR]> SELECT *FROM users
-> ;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | name | password | email | programador | birthday | work_time | created_at | updated_at |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | harold | 12345678 | harold123@gmail.com | 1 | 2002-04-05 | 13:05:00 | 2024-09-25 04:46:05.643784 | 2024-09-25 04:46:05.643784 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

21. Pruebas por comportamiento BDD. Para trabajar con el comportamiento de la aplicación y comprobar que las validaciones anteriormente agregadas se comportan de la manera correcta sin necesidad de hacer uso del navegador web, se puede hacer uso de la gema capybara, que ayuda a probar aplicaciones web simulando como un usuario real interactúa con la aplicación, esta trabaja en conjunto con rspec.

21.1 Para hacer testing primero crear el directorio en donde se guardan los features.

```
root@debian:/home/harold/Guia7/App_mysql# rails g rspec:feature user
create spec/features/user_spec.rb
root@debian:/home/harold/Guia7/App_mysql#
```

21.2 Abrir el archivo .rspec que está en la raíz de la aplicación y agregar la siguiente línea.

```
! .rspec
1 --require spec_helper
2 -f documentation
```

21.3 Abrir el archivo spec/features/user_spec.rb y agregar el siguiente código, para hacer test al comportamiento al momento de crear un nuevo usuario.

```
ec > features > user_spec.rb
1  require 'rails_helper'
2
3  RSpec.feature "Users", type: :feature do
4    #pending "add some scenarios (or delete) #{__FILE__}"
5    feature "User want to create a new user successfully" do
6      scenario "Create a new user account" do
7        visit new_user_path
8        fill_in "Name", :with=> "Pedro"
9        fill_in "Password", :with=> "12345678"
10       fill_in "Email", :with=> "pedro@gmail.com"
11       click_button "Create User"
12       expect(page).to have_text("User was successfully created")
13     end
14   end
15 end
16
```

21.4 Ejecutar el test con rspec, escribiendo el siguiente comando en el terminal.

```
root@debian:/home/harold/Guia7/App_mysql# rspec spec/features/user_spec.rb

Users
  User want to create a new user successfully
    Create a new user account

Finished in 1.1 seconds (files took 2.11 seconds to load)
1 example, 0 failures

root@debian:/home/harold/Guia7/App_mysql#
```

22. Otra prueba que se puede realizar en relación al comportamiento, es cuando el usuario presione click en el link back se debe mostrar la página principal de la aplicación donde se muestran todos los usuarios almacenados.

22.1 Copiar el siguiente código en el archivo spec/features/user_spec.rb.

```
pec > features > user_spec.rb
1  require 'rails_helper'
2
3  RSpec.feature "Users", type: :feature do
4    #pending "add some scenarios (or delete) #{__FILE__}"
5    feature "User want to create a new user successfully" do
6      scenario "Create a new user account" do
7        visit new_user_path
8        fill_in "Name", :with=> "Pedro"
9        fill_in "Password", :with=> "12345678"
10       fill_in "Email", :with=> "pedro@gmail.com"
11       click_button "Create User"
12       expect(page).to have_text("User was successfully created")
13     end
14   end
15   feature "User want to see all users created" do
16     scenario "User is on index page" do
17       visit new_user_path
18       click_link "Back"
19       expect(page).to have_content("Users")
20     end
21   end
22 end
23
```

22.2 Ejecutar el test nuevamente y podrá observar que se ejecuta sin ningún problema.

```
root@debian:/home/harold/Guia7/App_mysql# rspec spec/features/user_spec.rb

Users
  User want to create a new user successfully
    Create a new user account
  User want to see all users created
    User is on index page

Finished in 0.34342 seconds (files took 1.64 seconds to load)
2 examples, 0 failures
```

Ejercicios propuestos.

2. Crear una aplicación en Rails utilizando mysql; configurar una base de datos de nombre Datos_estudiante y utilice scaffold para generar el código y crear una tabla Estudiante con los campos nombres, apellidos, carrera, carnet y validar todos los campos necesarios.

```
2 development:
3   database: RoR
4
5 test:
6   <<: *default
7   username: root
8   password: harold123
9   database: RoR_test
10
11 estudiante_development:
12   <<: *default
13   username: root
14   password: harold123
15   host: 127.0.0.1
16   port: 3306
17   database: Datos_estudiante
18
19 estudiante_test:
20   <<: *default
21   username: root
22   password: harold123
23   host: 127.0.0.1
24   port: 3306
25   database: Datos_estudiante_test
```

```
root@debian:/home/harold/Guia7/App_mysql# RAILS_ENV=estudiante_development
rake db:create
config.eager_load is set to nil. Please update your config/environments/*.rb
files accordingly:

* development - set it to false
* test - set it to false (unless you use a tool that preloads your test e
nvironment)
* production - set it to true

Created database 'Datos_estudiante'
root@debian:/home/harold/Guia7/App_mysql# RAILS_ENV=estudiante_test rake db
:create
config.eager_load is set to nil. Please update your config/environments/*.r
b files accordingly:

* development - set it to false
* test - set it to false (unless you use a tool that preloads your test e
nvironment)
* production - set it to true

Created database 'Datos_estudiante_test'
root@debian:/home/harold/Guia7/App_mysql#
```

```
harold@debian: ~  
root@debian:/home/harold/Guia7/App_mysql# rails generate scaffold Estudiante nombre:string apellidos:string carrera:string carnet:string  
invoke active_record  
create db/migrate/20240925155729_create_estudiantes.rb  
create app/models/estudiante.rb  
invoke rspec  
create spec/models/estudiante_spec.rb  
invoke resource_route  
route resources :estudiantes  
invoke scaffold_controller  
create app/controllers/estudiantes_controller.rb  
invoke erb  
create app/views/estudiantes  
create app/views/estudiantes/index.html.erb  
create app/views/estudiantes/edit.html.erb  
create app/views/estudiantes/show.html.erb  
create app/views/estudiantes/new.html.erb  
create app/views/estudiantes/_form.html.erb  
create app/views/estudiantes/_estudiante.html.erb  
invoke resource_route  
invoke rspec  
invoke helper  
create app/helpers/estudiantes_helper.rb  
invoke rspec
```

```
harold@debian: ~  
root@debian:/home/harold/Guia7/App_mysql# rails db:migrate  
== 20240925155729 CreateEstudiantes: migrating =====  
====  
<-- create_table(:estudiantes)  
-> 0.0558s  
== 20240925155729 CreateEstudiantes: migrated (0.0574s) =====  
====  
  
root@debian:/home/harold/Guia7/App_mysql#
```

```
class Estudiante < ApplicationRecord  
  validates_presence_of :nombre, presence: true  
  validates_format_of :nombre, with: /\A[a-zA-Z\s]+\z/, message: "solo puede contener letras y espacios"  
  validates_length_of :nombre, maximum: 30  
  
  validates_presence_of :apellidos, presence: true  
  validates_format_of :apellidos, with: /\A[a-zA-Z\s]+\z/, message: "solo puede contener letras y espacios"  
  validates_length_of :apellidos, maximum: 30  
  
  validates_presence_of :carrera, presence: true  
  validates_length_of :carrera, maximum: 50  
  
  validates_presence_of :carnet, presence: true  
  validates_length_of :carnet, maximum: 20  
end
```

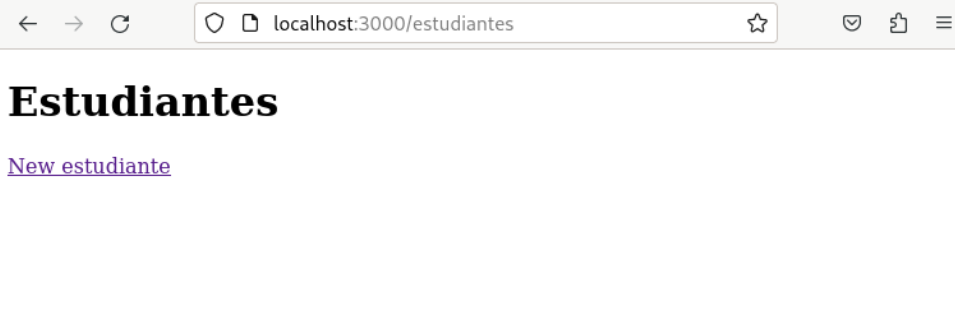
```
require 'rails_helper'

RSpec.describe Estudiante, type: :model do
  #pending "add some examples to (or delete) #{__FILE__}"
  context "nombre" do
    it { should validate_presence_of(:nombre) }
    it { should validate_length_of(:nombre).is_at_most(30) }
    it { should_not allow_value("12345").for(:nombre) }
    it { should allow_value("Juan").for(:nombre) }
    it { should allow_value("Juan Antonio").for(:nombre) }
  end

  context "apellidos" do
    it { should validate_presence_of(:apellidos) }
    it { should validate_length_of(:apellidos).is_at_most(30) }
    it { should_not allow_value("67890").for(:apellidos) }
    it { should allow_value("García").for(:apellidos) }
    it { should allow_value("García López").for(:apellidos) }
  end

  context "carrera" do
    it { should validate_presence_of(:carrera) }
    it { should validate_length_of(:carrera).is_at_most(50) }
  end

  context "carnet" do
```



New estudiante

6 errors prohibited this estudiante from being saved:

- Nombres can't be blank
- Nombres solo puede contener letras
- Apellidos can't be blank
- Apellidos solo puede contener letras
- Carrera can't be blank
- Carnet can't be blank

Nombres

Apellidos

Carrera

Carnet

Create Estudiante

[Back to estudiantes](#)

Estudiante was successfully created.

Nombres: Juan Jose

Apellidos: Lopex Perez

Carrera: Ingenieria

Carnet: 20987456

[Edit this estudiante](#) | [Back to estudiantes](#)

Destroy this estudiante