

Universidad Nacional Autónoma de Nicaragua Unan León



Facultad de Ciencias y Tecnología.

Componente: Software como un servicio

Tema: Practica 6

Grupo:1

Docente: Erving Montes

Integrante:

-Harold Steven Garcia Ramos.

Fecha: 11 de septiembre de 2024

1. Crear un nuevo proyecto en Rails.

1.1 Crear un proyecto y ubicarse en su interior haciendo uso del terminal.

```
harold@debian: ~/Proyectos_RoR
Contraseña:
root@debian:/home/harold/Proyectos_RoR# rails new App_scaffold

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 vacío en /home/harold/Proyectos_RoR/App_scaffold
it/
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
```

1.2 Generar un scaffold, con una tabla llamada Buy con los campos:

```
harold@debian: ~/Proyectos_RoR
root@debian:/home/harold/Proyectos_RoR/App_scaffold# rails generate scaffold Buy
category:string description:text amount:decimal
invoke  active_record
create  db/migrate/20240911134543_create_buys.rb
create  app/models/buy.rb
invoke  test_unit
create  test/models/buy_test.rb
create  test/fixtures/buys.yml
invoke  resource_route
route   resources :buys
invoke  scaffold_controller
create  app/controllers/buys_controller.rb
invoke  erb
create  app/views/buys
create  app/views/buys/index.html.erb
create  app/views/buys/edit.html.erb
create  app/views/buys/show.html.erb
create  app/views/buys/new.html.erb
create  app/views/buys/_form.html.erb
create  app/views/buys/_buy.html.erb
invoke  resource_route
invoke  test_unit
create  test/controllers/buys_controller_test.rb
create  test/system/buys_test.rb
```

2.Migrar los datos. Las migraciones son el modo más conveniente de cambiar el esquema de la base de datos a través del tiempo de una manera consistente y fácil, utilizan un lenguaje de definición de esquemas (DSL) en Ruby, por lo que no tiene que escribir SQL.

```
root@debian:/home/harold/Proyectos_RoR/App_scaffold# rake db:migrate
== 20240911134543 CreateBuys: migrating =====
-- create_table(:buys)
   -> 0.0029s
== 20240911134543 CreateBuys: migrated (0.0032s) =====
root@debian:/home/harold/Proyectos_RoR/App_scaffold#
```

3.Rutas Las rutas son una parte muy importante de la aplicación. Como se mencionó anteriormente scaffold genera las rutas, para que pueda existir una conexión entre las peticiones del usuario y la aplicación.

```
harold@debian: ~/Proyectos_RoR
GNU nano 7.2 routes.rb
Rails.application.routes.draw do
  resources :buys
  # 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 excep>
  # Can be used by load balancers and uptime monitors to verify that the app is>
  get "up" => "rails/health#show", as: :rails_health_check

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

3.1 Escribir el siguiente comando en el terminal y observar las rutas generadas.

```
harold@debian: ~/Proyectos_RoR
root@debian:/home/harold/Proyectos_RoR/App_scaffold# rails routes
Prefix Verb   URI Pattern
buys GET      /buys(.:format)
          buys#index
          POST   /buys(.:format)
          buys#create
new_buy GET    /buys/new(.:format)
          buys#new
edit_buy GET    /buys/:id/edit(.:format)
          buys#edit
buy GET       /buys/:id(.:format)
          buys#show
          PATCH  /buys/:id(.:format)
          buys#update
          PUT     /buys/:id(.:format)
          buys#update
          DELETE  /buys/:id(.:format)
          buys#destroy
rails_health_check GET    /up(.:format)
          rails/health#show
turbo_recede_historical_location GET    /recede_historical_location(.:format)
          turbo/native/navigation#recede
turbo_resume_historical_location GET    /resume_historical_location(.:format)
```

4.Verificar el controlador. Observar que se han creado los distintos métodos del controlador en el archivo buys_controller.rb dentro del directorio (app/controller/).

```
harold@debian: ~/Proyectos_RoR
GNU nano 7.2 app/controllers/buys_controller.rb
class BuysController < ApplicationController
  before_action :set_buy, only: [:show, :edit, :update, :destroy]

  # GET /buys or /buys.json
  def index
    @buys = Buy.all
  end

  # GET /buys/1 or /buys/1.json
  def show
  end

  # GET /buys/new
  def new
    @buy = Buy.new
  end

  # GET /buys/1/edit
  def edit
  end
end
```

5.Configurar el index generado por el scaffold como la página de inicio de la aplicación.

5.1 Escribir la siguiente línea de código en el archivo config/routes.rb.

```
harold@debian: ~/Proyectos_RoR
GNU nano 7.2 config/routes.rb
Rails.application.routes.draw do
  resources :buys
  # 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 "buys#index"
end
```

5.2 Iniciar el servidor.

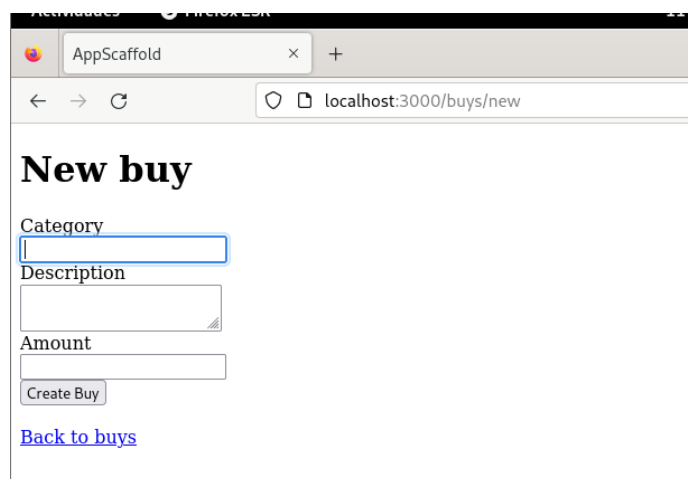
```
harold@debian: ~/Proyectos_RoR
root@debian:/home/harold/Proyectos_RoR/App_scaffold#
root@debian:/home/harold/Proyectos_RoR/App_scaffold# nano app/controllers/buys_controller

root@debian:/home/harold/Proyectos_RoR/App_scaffold# nano config/routes.rb
root@debian:/home/harold/Proyectos_RoR/App_scaffold# 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: 2705
* Listening on http://127.0.0.1:3000
* Listening on http://[::]:3000
Use Ctrl-C to stop
```

5.3 Abrir el navegador y acceder a la dirección.

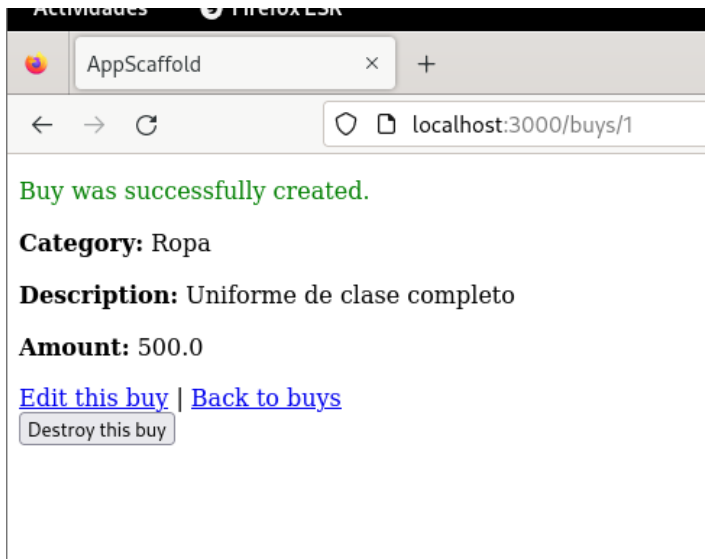
6.Verificar el funcionamiento de las vistas.

6.1 En la vista mostrada anteriormente, presiona new buy, se mostrará el formulario para poder agregar datos a la aplicación.

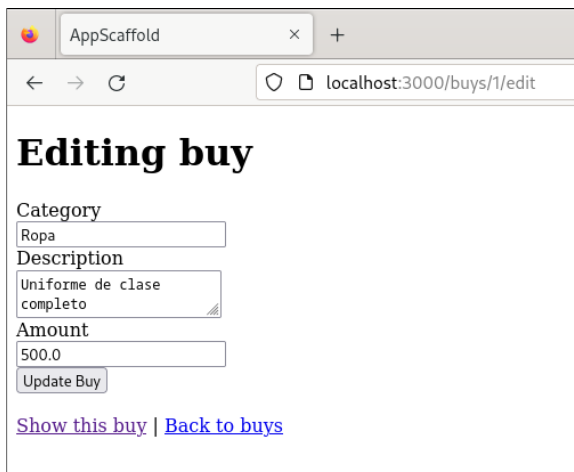


The screenshot shows a web browser window with the title 'AppScaffold'. The address bar displays 'localhost:3000/buys/new'. The page content includes a heading 'New buy' followed by a form with three input fields: 'Category', 'Description', and 'Amount'. Below these fields is a 'Create Buy' button and a blue link labeled 'Back to buys'.

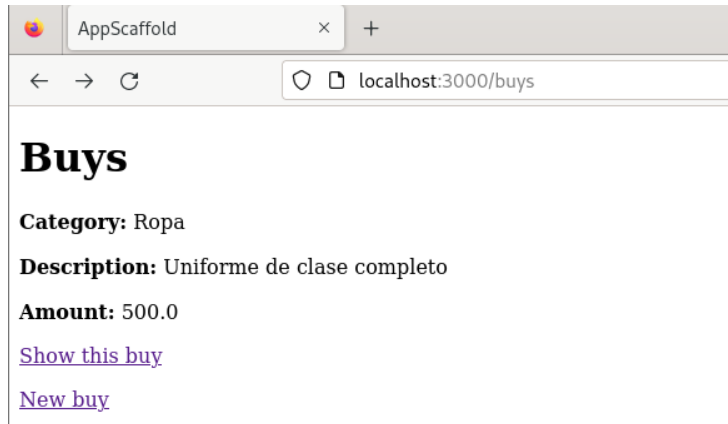
6.2 Ingresar una nueva compra llenando el formulario, al presionar el botón create buy, el dato se ha creado y almacenado correctamente.



1.1 Click en Edit, podrá observar que permite editar el dato anteriormente almacenado.

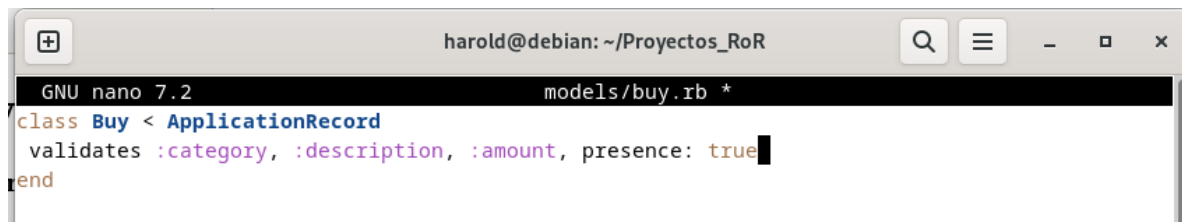


1.2 Verificar que se muestran los datos almacenados dentro del index de la aplicación.



2. Validaciones al modelo buy Si el usuario agrega un nuevo dato en la aplicación, podrá agregar datos vacíos debido a que no se le ha agregado ningún tipo de validaciones al modelo. Una de las ventajas de Rails es permite validar los campos del formulario de una manera más sencilla, en este caso solo se validará que los campos del modelo no estén vacíos para poder agregar un nuevo dato a la aplicación.

2.1 Abrir el archivo /model/buy.rb y agregar el siguiente código.



A screenshot of a web browser window at the URL `localhost:3000/buys/new`. The page has a title "New buy". Below the title, a red message states "3 errors prohibited this buy from being saved:". This is followed by a bulleted list of errors in red: "Category can't be blank", "Description can't be blank", and "Amount can't be blank". Below the list are three input fields labeled "Category", "Description", and "Amount", all of which are empty. At the bottom of the form is a "Create Buy" button and a link labeled "Back to buys".

Ejercicios propuestos.

2. Crear un proyecto nuevo, utilice scaffold para generar el código y crear una tabla de nombre Estudiante con los campos nombres, apellidos, carrera, carnet, fecha de nacimiento, edad, celular; deberá configurar el archivo routes.rb para que la página principal de la aplicación sea el index generado por el scaffold, validar el campo celular para que solo admita número y que no permita campos vacíos. La aplicación deberá mostrar un formulario parecido al de la figura 69, se puede observar en la figura como el framework crea automáticamente las cajas de texto, dependiendo del tipo de dato que se le especifica al generar el scaffold, como el campo fecha de nacimiento donde crea un input tipo date_select en el formulario para seleccionar la fecha de nacimiento del estudiante.

```
narold@debian: ~  
root@debian: /home/harold/Proyectos_RoR# rails new EstudianteApp  
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/Proyectos_RoR/EstudianteApp/.  
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
```



```

root@debian:/home/harold/Proyectos_RoR/EstudianteApp# rails generate scaffold Es
tudiante nombres:string apellidos:string carrera:string carnet:string fecha_naci
miento:date edad:integer celular:string
  invoke  active_record
  create  db/migrate/20240911185649_create_estudiantes.rb
  create  app/models/estudiante.rb
  invoke  test_unit
  create  test/models/estudiante_test.rb
  create  test/fixtures/estudiantes.yml
  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

```

```

root@debian:/home/harold/Proyectos_RoR/EstudianteApp# rails db:migrate
== 20240911185649 CreateEstudiantes: migrating =====
-- create_table(:estudiantes)
   -> 0.0024s
== 20240911185649 CreateEstudiantes: migrated (0.0027s) =====

root@debian:/home/harold/Proyectos_RoR/EstudianteApp#

```

```

GNU nano 7.2                                config/routes.rb *
Rails.application.routes.draw do
  resources :estudiantes
  # 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
  # Can be used by load balancers and uptime monitors to verify that the app is up
  get "up" => "rails/health#show", as: :rails_health_check

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

```

```

harold@debian: ~
GNU nano 7.2                                app/models/estudiante.rb *
class Estudiante < ApplicationRecord
  validates :nombres, :apellidos, :carrera, :carnet, :fecha_nacimiento, :edad, :celular, presence: true
  validates :celular, format: {with: /\A\d+\z/, message: "Solo se permiten numeros en este campo" }
end

```

EstudianteApp

localhost:3000/estudiantes/new

New estudiante

Nombres

Harold

Apellidos

Garcia

Carrera

Ingenieria Telematica

Carnet

20-02315-0

Fecha nacimiento

05 / 04 / 2002

Edad

22

Celular

87964521

Create Estudiante

[Back to estudiantes](#)