

**UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA  
(UNAN-LEÓN)**

**FACULTAD DE CIENCIAS Y TECNOLOGÍAS.**

**INGENIERIA EN TELEMATICA.**



**COMPONENTE:** Gestión de Sistemas Unix.

**Grupo 2 Telemática**

**INTEGRANTES:**

➤ Hosting Stewart Lopez Martinez.

**DOCENTE:** Ervin Montes

**¡A LA LIBERTAD POR LA UNIVERSIDAD!  
“AÑO LECTIVO 2024”**

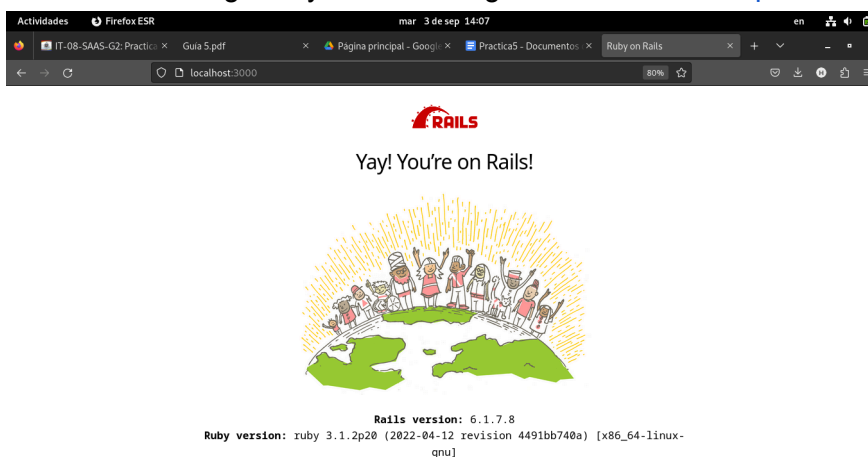
1. Creación de un proyecto nuevo
  - 1.1. Ubicarse en el directorio donde se va almacenar el proyecto.
  - 1.2. Generar un nuevo proyecto.

```
Hosting@debian 2024-09-03 13:42:11 ~ $ cd Proyectos_RoR
Hosting@debian 2024-09-03 13:42:29 Proyectos_RoR $ rails new my_app
create
create README.md
create Rakefile
create .ruby-version
create config.ru
create .gitignore
create .gitattributes
create Gemfile
run git init from "."
```

- 1.3. Ubicarse dentro del proyecto e iniciar el servidor para comprobar que funciona sin ningún problema.

```
Hosting@debian 2024-09-03 14:05:26 Proyectos_RoR $ cd my_app
Hosting@debian 2024-09-03 14:05:33 my_app $ rails s
=> Booting Puma
=> Rails 6.1.7.8 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.6.8 (ruby 3.1.2-p20) ("Birdie's Version")
* Min threads: 5
* Max threads: 5
* Environment: development
* PID: 5756
* Listening on http://127.0.0.1:3000
* Listening on http://[::1]:3000
Use Ctrl-C to stop
```

- 1.4. Abrir el navegador y escribir la siguiente dirección. <http://localhost:3000>



1. Generar un nuevo controlador al proyecto.

1.1. Verificar cuál es el controlador existente dentro del proyecto. De la siguiente forma se obtendrá el nombre de los controladores que existen hasta el momento.

1.2. Generar el nuevo controlador y una acción correspondiente a ese controlador.

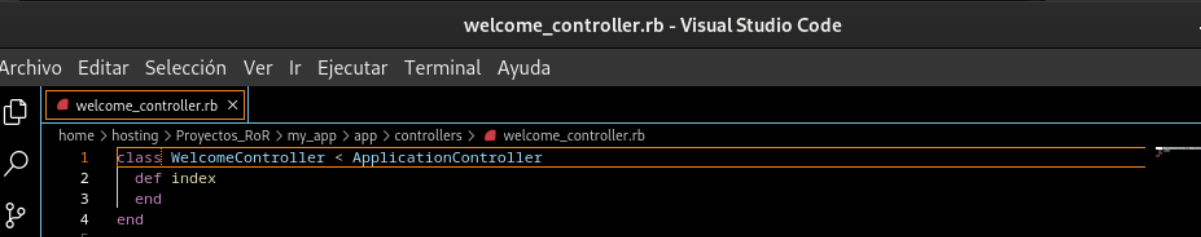
```
hosting@debian:~/Proyectos_RoR/my_app
Hosting@debian 2024-09-03 18:25:54 ~ $ cd Proyectos_RoR
Hosting@debian 2024-09-03 18:28:42 Proyectos_RoR $ cd my_app
Hosting@debian 2024-09-03 18:28:50 my_app $ ls app/controllers/*_controller.rb
app/controllers/application_controller.rb
Hosting@debian 2024-09-03 18:31:17 my_app $ rails generate controller welcome index
Running via Spring preloader in process 3580
  create  app/controllers/welcome_controller.rb
  route  get 'welcome/index'
  invoke  erb
  create  app/views/welcome
  create  app/views/welcome/index.html.erb
  invoke  test_unit
  create  test/controllers/welcome_controller_test.rb
  invoke  helper
  create  app/helpers/welcome_helper.rb
  invoke  test_unit
  invoke  assets
  invoke  scss
  create  app/assets/stylesheets/welcome.scss
Hosting@debian 2024-09-03 18:35:06 my_app $
```

1.3. Verificar nuevamente los controladores del proyecto. Podrá observar que ahora ya existe uno nuevo, y que se corresponde con el creado anteriormente.

```
  create  app/assets/stylesheets/welcome.scss
Hosting@debian 2024-09-03 18:35:06 my_app $ ls app/controllers/*_controller.rb
app/controllers/application_controller.rb  app/controllers/welcome_controller.rb
Hosting@debian 2024-09-03 18:36:46 my_app $
```

1.4. Abrir el archivo `welcome_controller.rb` que se encuentra en el directorio (`app/controllers/`) y verificar que existe el método `index`, que fue creado al momento de generar el controlador, este es el método (acción) que tiene relación con la vista `index.html.erb`.

```
Hosting@debian 2024-09-03 18:41:31 my_app $ code app/controllers/welcome_controller.rb
```

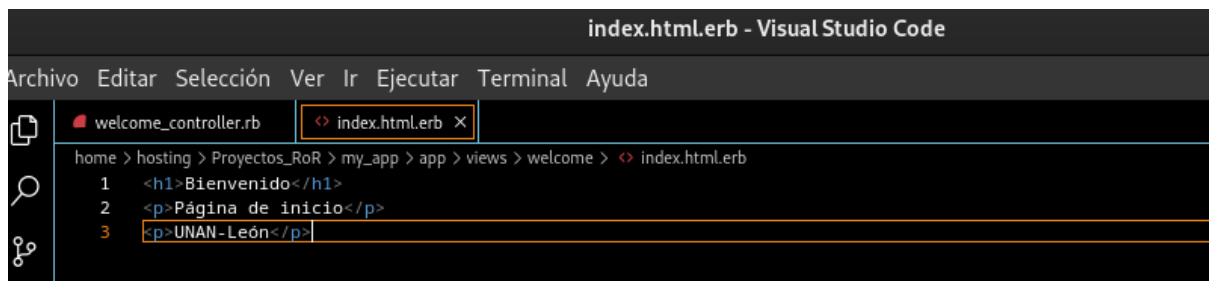


```
1 class WelcomeController < ApplicationController
2   def index
3   end
4 end
```

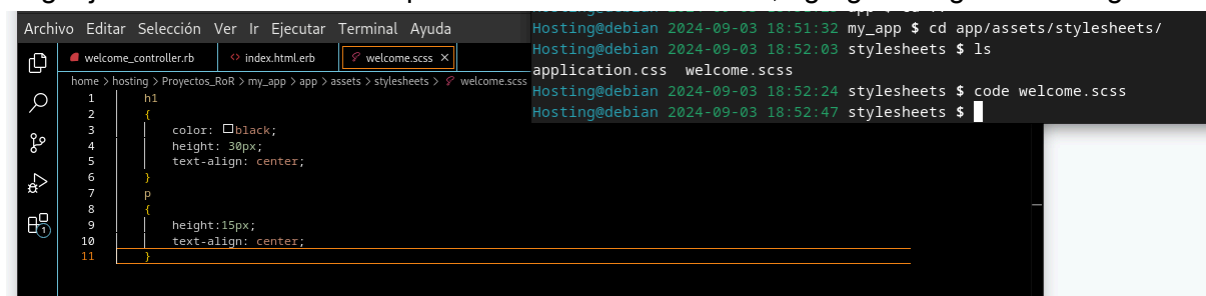
2. Trabajar con la vista Para comenzar a trabajar con las vistas se debe ubicar el directorio (app/view/), que es donde se almacenan las vistas del proyecto, ahí se observa un archivo llamado index.html.erb, este es por medio del cual los usuarios finales interactúan con la aplicación

2.1. Editar el archivo index.html.erb del nuevo controlador y escribir el siguiente código html.

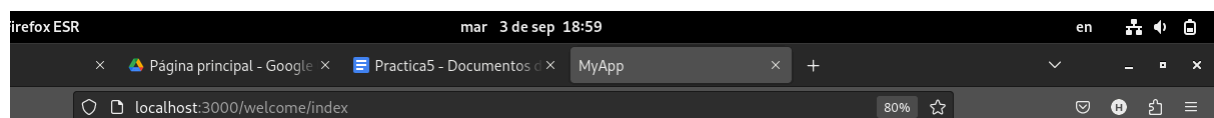
```
Hosting@debian 2024-09-03 18:42:32 my_app $ cd app/views
Hosting@debian 2024-09-03 18:44:24 views $ ls
layouts  welcome
Hosting@debian 2024-09-03 18:44:32 views $ cd welcome
Hosting@debian 2024-09-03 18:45:12 welcome $ ls
index.html.erb
Hosting@debian 2024-09-03 18:45:17 welcome $ code index.html.erb
Hosting@debian 2024-09-03 18:45:44 welcome $
```



2.2. Editar el archivo welcome.scss, que se encuentra en el directorio (app/assets/stylesheets/), en este directorio es donde se almacenan todas las hojas de estilos que contiene el proyecto. Todos los archivos de estilos en rails terminan en ".scss" debido a que el framework permite utilizar el lenguaje de estilos llamado Sass (Syntactically Awesome StyleSheets), es una extensión de css que agrega potencia y elegancia al lenguaje básico. Para darle un poco de formato a las vistas, agregar el siguiente código.



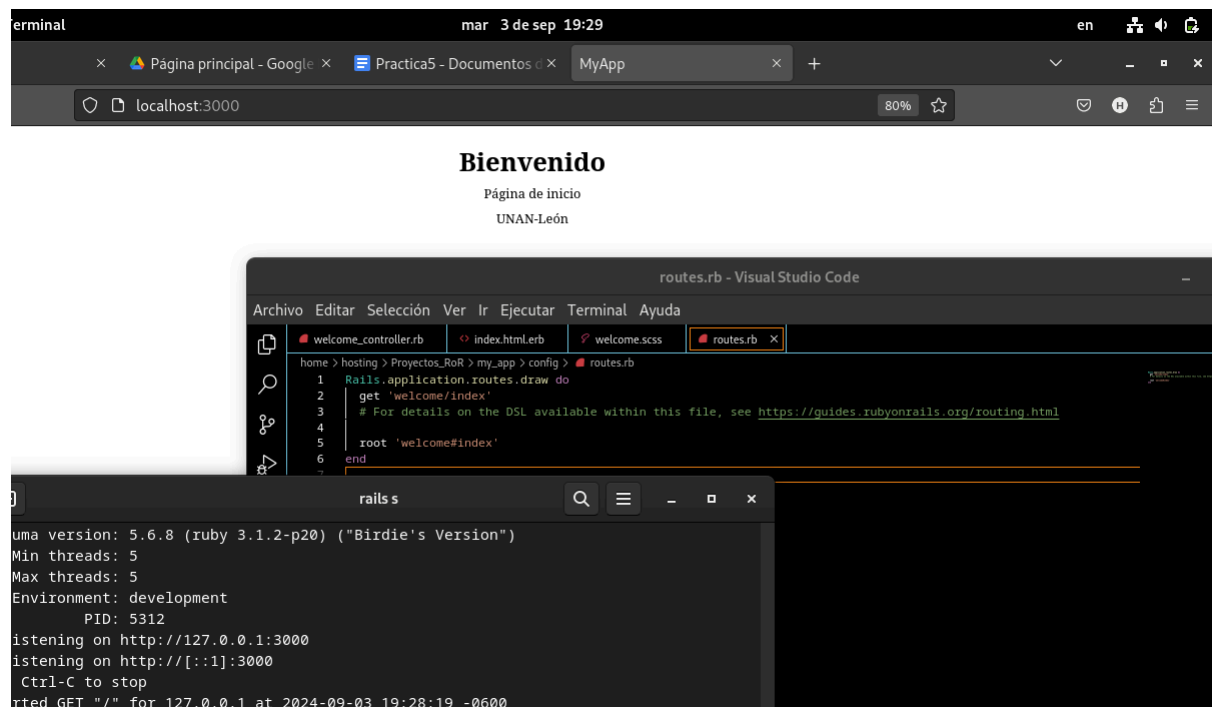
2.3. Ir a la dirección (<http://localhost:3000/welcome/index>), podrá obtener una vista como la



## Bienvenido

Página de inicio  
UNAN-León

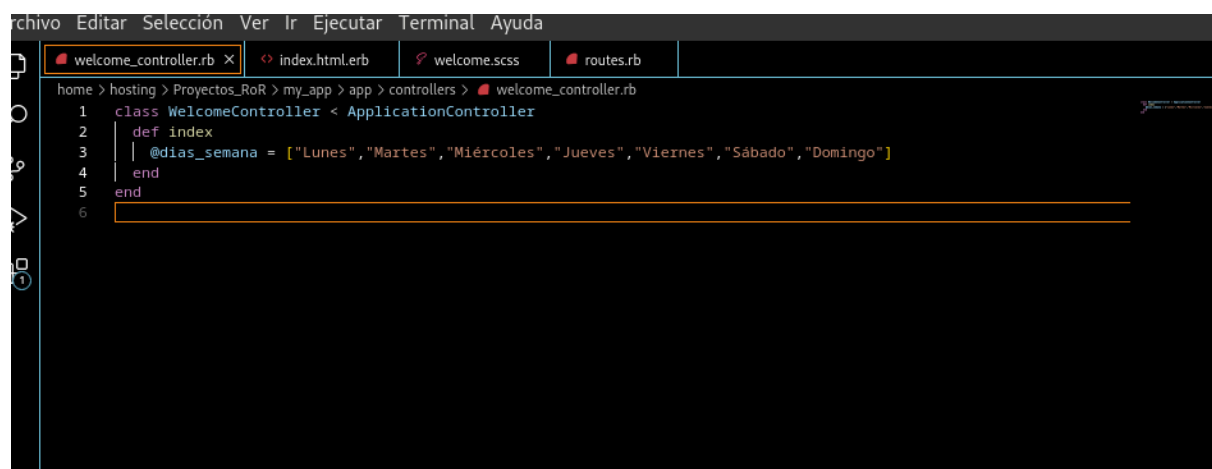
3. Si navega directamente en `http://localhost:3000/` no se tendrá acceso a la vista antes modificada, para esto, se debe modificar el archivo `routes.rb`, de manera que inicie (root) con la vista de nombre `index.html.erb`.



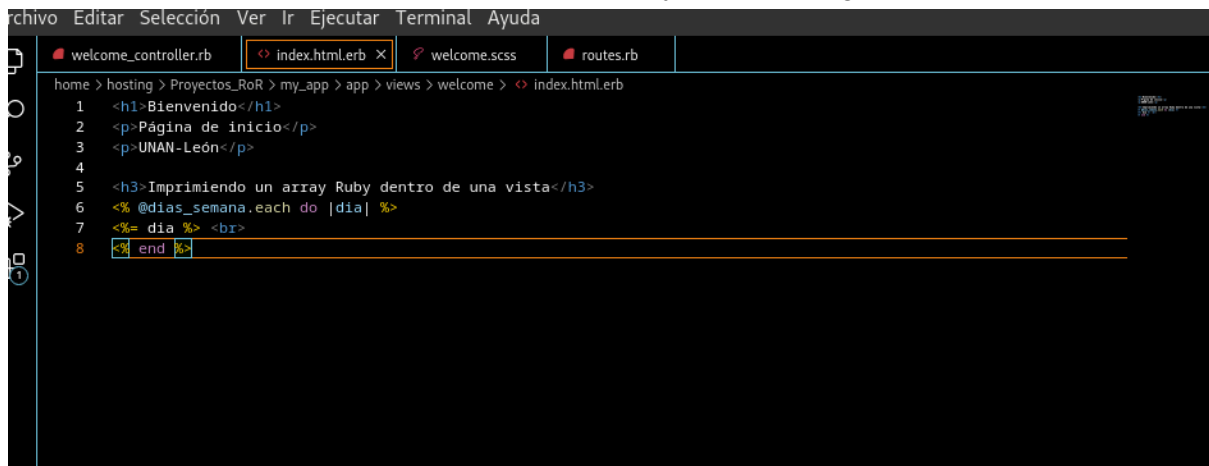
## 5. Uso del lenguaje Ruby en el framework Rails.

Dentro del framework se trabaja con lenguaje Ruby; de igual forma en las vistas se puede agregar código Ruby a través del lenguaje Embedded Ruby, conocido como eRuby o ERB, haciendo uso de etiquetas propias del lenguaje ERB.

5.1. Para ver el funcionamiento ir al archivo `welcome_controller.rb` y crear un array global con los días de la semana dentro del método `index`.



5.2. Para poder hacer uso del array dentro de la vista, se debe utilizar lenguaje ERB en el archivo de la vista index.html.erb; editar el archivo y escribir el siguiente.



```
home > hosting > Proyectos_RoR > my_app > app > views > welcome > index.html.erb
1 <h1>Bienvenido</h1>
2 <p>Página de inicio</p>
3 <p>UNAN-León</p>
4
5 <h3>Imprimiendo un array Ruby dentro de una vista</h3>
6 <%= @dias_semana.each do |dia| %>
7 <%= dia %> <br>
8 <%= end %>
```

Se verá de la siguiente manera:



5.3. De igual forma se puede hacer uso de los distintos métodos propios de los array, en el caso anterior se utilizó el método each para recorrer el array. Escribir el siguiente código en el archivo index.html.erb.



Ejercicios propuestos para ser entregados al docente

2. Muestre en la vista index.html.erb el primer dato almacenado y el tamaño completo del array, haga uso de métodos similares a los vistos anteriormente.

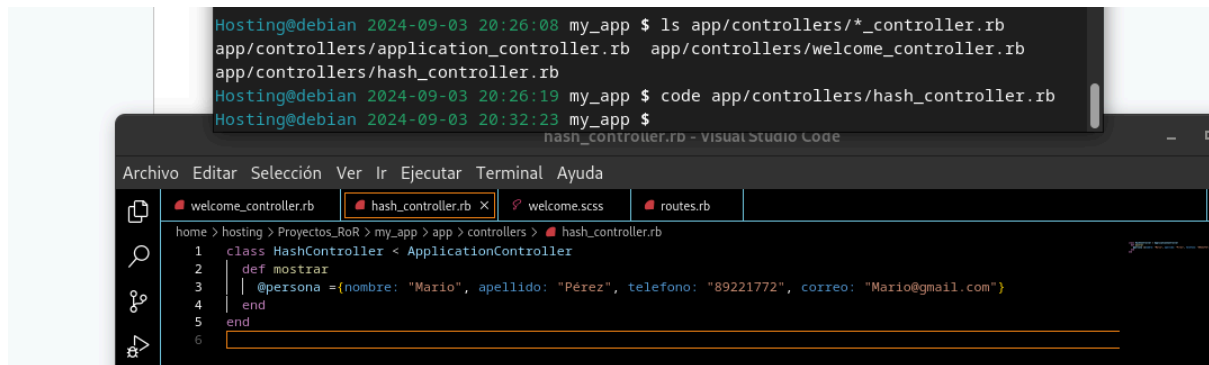


3. Generar un nuevo controlador de nombre “hash” y una acción que es la que tendrá relación con la vista llamada “mostrar”, similar al realizado al inicio de esta guía.

```
Hosting@debian 2024-09-03 20:23:05 my_app $ rails generate controller hash mostrar
Running via Spring preloader in process 6149
  create  app/controllers/hash_controller.rb
  route   get 'hash/mostrar'
  invoke  erb
  create  app/views/hash
  create  app/views/hash/mostrar.html.erb
  invoke  test_unit
  create  test/controllers/hash_controller_test.rb
  invoke  helper
  create  app/helpers/hash_helper.rb
  invoke  test_unit
  invoke  assets
  invoke  scss
  create  app/assets/stylesheets/hash.scss
Hosting@debian 2024-09-03 20:23:55 my_app $
```



4. Dentro de la acción mostrar que se creó junto al nuevo controlador del proyecto, crear un hash con los datos de una persona: nombre, apellido, teléfono, correo.



The image shows a terminal window at the top with the following commands and output:

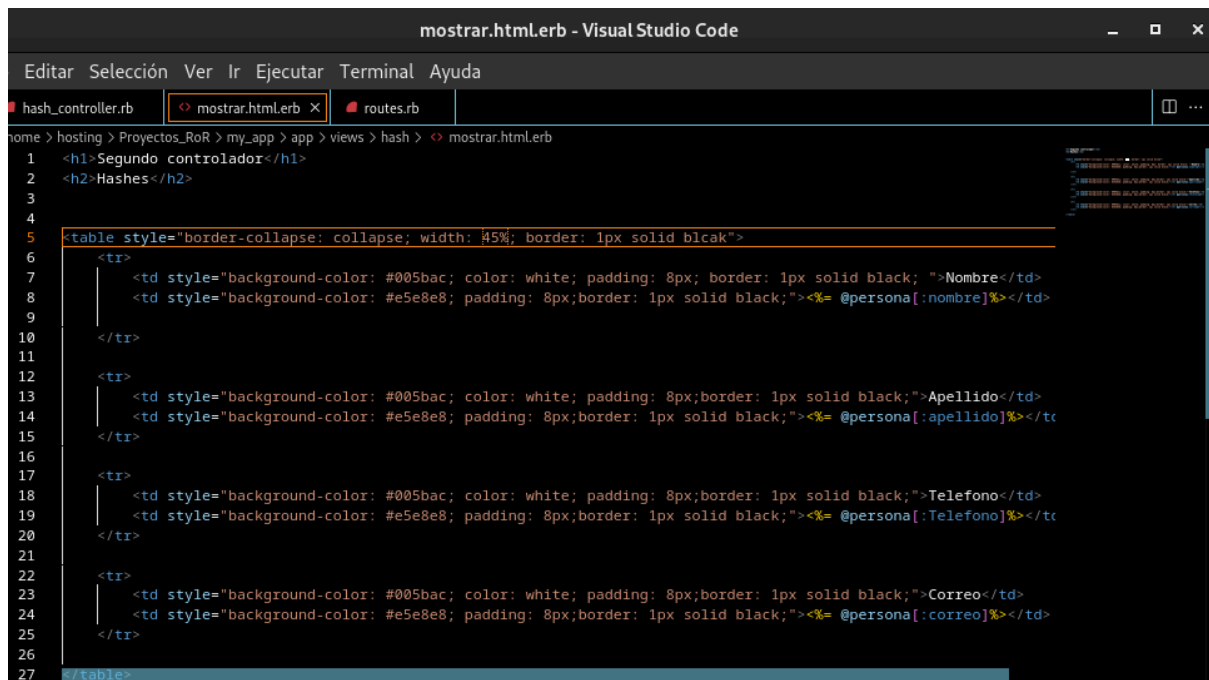
```
Hosting@debian 2024-09-03 20:26:08 my_app $ ls app/controllers/*_controller.rb
app/controllers/application_controller.rb  app/controllers/welcome_controller.rb
app/controllers/hash_controller.rb
Hosting@debian 2024-09-03 20:26:19 my_app $ code app/controllers/hash_controller.rb
Hosting@debian 2024-09-03 20:32:23 my_app $
```

Below the terminal is a VS Code editor window titled "hash\_controller.rb - Visual Studio Code". The editor shows the following code in the "hash\_controller.rb" file:

```
1 class HashController < ApplicationController
2   def mostrar
3     @persona = {nombre: "Mario", apellido: "Pérez", telefono: "89221772", correo: "Mario@gmail.com"}
4   end
5 end
```

5. Modificar la vista del controlador creado anteriormente y mostrar los datos del hash en una tabla html.

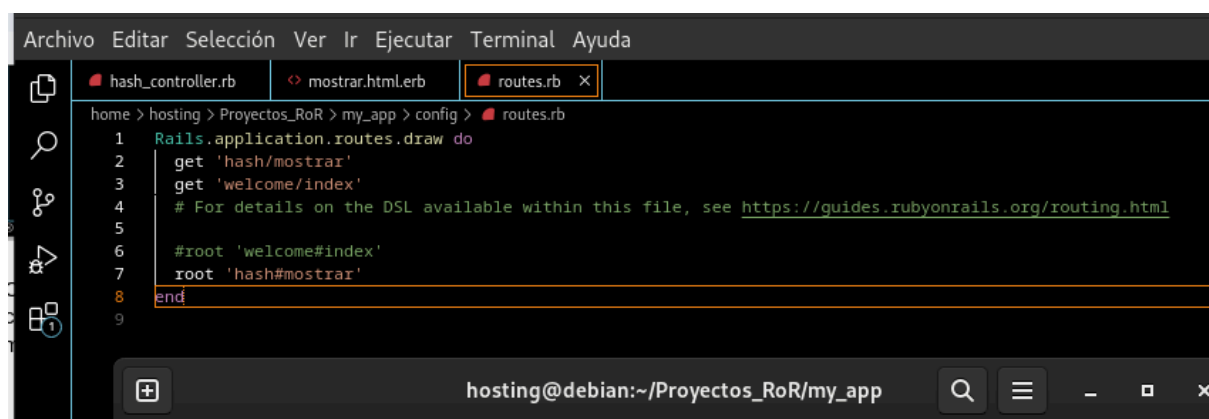
6. Agregar código css a la tabla, se mostrará en la vista como en la figura que se puede ver que la tabla contiene formato, colores, bordes.



The image shows a VS Code editor window titled "mostrar.html.erb - Visual Studio Code". The editor shows the following code in the "mostrar.html.erb" file:

```
1 <h1>Segundo controlador</h1>
2 <h2>Hashes</h2>
3
4
5 <table style="border-collapse: collapse; width: 45%; border: 1px solid black">
6   <tr>
7     <td style="background-color: #005bac; color: white; padding: 8px; border: 1px solid black;">Nombre</td>
8     <td style="background-color: #e5e8e8; padding: 8px; border: 1px solid black;"><%= @persona[:nombre]%></td>
9   </tr>
10
11   <tr>
12     <td style="background-color: #005bac; color: white; padding: 8px; border: 1px solid black;">Apellido</td>
13     <td style="background-color: #e5e8e8; padding: 8px; border: 1px solid black;"><%= @persona[:apellido]%></td>
14   </tr>
15
16   <tr>
17     <td style="background-color: #005bac; color: white; padding: 8px; border: 1px solid black;">Telefono</td>
18     <td style="background-color: #e5e8e8; padding: 8px; border: 1px solid black;"><%= @persona[:telefono]%></td>
19   </tr>
20
21   <tr>
22     <td style="background-color: #005bac; color: white; padding: 8px; border: 1px solid black;">Correo</td>
23     <td style="background-color: #e5e8e8; padding: 8px; border: 1px solid black;"><%= @persona[:correo]%></td>
24   </tr>
25 </table>
```

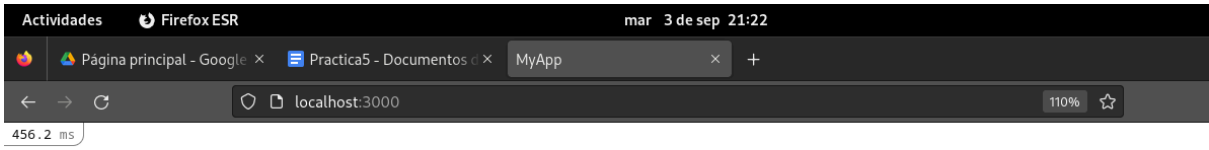
7. Configurar el archivo routes.rb para cambiar la página de inicio del proyecto y ahora hacerlo apuntar al nuevo controlador junto con su respectiva vista, la cual deberá quedar como se observa en la figura



The image shows a VS Code editor window titled "routes.rb - Visual Studio Code". The editor shows the following code in the "routes.rb" file:

```
1 Rails.application.routes.draw do
2   get 'hash/mostrar'
3   get 'welcome/index'
4   # For details on the DSL available within this file, see https://guides.rubyonrails.org/routing.html
5
6   #root 'welcome#index'
7   root 'hash#mostrar'
8 end
```

Finalmente se veria asi :



## Segundo controlador

### Hashes

Nombre	Mario
Apellido	Pérez
Telefono	89221772
Correo	Mario@gmail.com