

Tutorial Devise

Que es devise

Devise es una herramienta que les simplificará mucho el manejo de usuarios en sus proyectos. Es ocupada para no tener que crear el manejo de sesiones de forma manual, además de esto provee algunos *helpers* que facilitan chequear si un usuario se encuentra registrado, quien es el usuario y otras tareas.

La documentación completa la pueden encontrar [acá](#).

Instalación

El primer paso es añadir la siguiente línea a su Gemfile:

```
gem 'devise'
```

Y descomentar la gema `bcrypt`. Si no está, agregarla:

```
gem 'bcrypt'
```

Luego en consola correr lo siguiente:

```
docker-compose build
```

```
docker-compose run web rails generate devise:install
```

Es probable que deban hacer unos cambios para que les funcione bien. El primero es abrir `config/environments/development.rb` y añadir justo antes de `end` lo siguiente:

```
config.action_mailer.default_url_options = { host: 'localhost', port: 3000 }
```

Lo siguiente es abrir `app/views/layouts/application.html.erb` y justo arriba de `<%= yield %>` agregar lo siguiente:

```
<% if notice %>
  <p class="alert alert-success"><%= notice %></p>
<% end %>
<% if alert %>
  <p class="alert alert-danger"><%= alert %></p>
<% end %>
```

Luego se puede crear el modelo de usuario. Para esto correr en consola:

```
docker-compose run web rails generate devise user
```

```
docker-compose run web rails db:migrate
```

Ahora tienen todo lo necesario para que se puedan crear cuentas, log in, log out, etc.

Comandos básicos

Notarán que Devise no genera los archivos de las vistas ni de los controladores, por lo que para editarlos es necesario generarlos por nuestra cuenta.

Para generar todos los controladores:

```
docker-compose run web rails g devise:controllers users
```

No olvidar ingresar la ruta a los nuevos controladores en su archivo `routes.rb`, por ejemplo:

```
Rails.application.routes.draw do
  devise_for :users, controllers: {
    sessions: 'users/sessions',
    registrations: 'users/registrations'
  }
end
```

Para generar un controlador en específico se incluye un `-c={nombre del controlador}`, por ejemplo:

```
docker-compose run web rails g devise:controllers users -c=registrations
```

Para generar todas las vistas:

```
docker-compose run web rails g devise:views
```

En caso de contar con más de un modelo de Devise (por ejemplo users y moderator_users), pueden especificar a cuál modelo generar las vistas:

```
docker-compose run web rails g devise:views users
```

Para generar views en específico se ocupa `-v`

```
docker-compose run web rails g devise:views -v registrations confirmations
```

Extendiendo Devise

Una vez incluida la gema en sus proyectos, podrán notar que el manejo de sesión que viene por defecto incluye solo correo y contraseña. Sin embargo, en sus proyectos puede que necesiten manejar más datos acerca de los usuarios y, por tanto, será necesario modificar lo hecho por Devise. Agreguemos un username. Para esto es necesario realizar una migración, por ejemplo:

```
docker-compose run web rails generate migration AddUsernameToUsers username:string
```

Lo anterior debió haber generado una migración como la siguiente (pueden editarlo si estiman conveniente):

```
class AddUsernameToUsers < ActiveRecord::Migration
  def change
    add_column :users, :username, :string
  end
end
```

Corremos las migraciones

```
docker-compose run web rails db:migrate
```

Pese a lo anterior, los usuarios se seguirán creando con un username vacío, dado que en ningún momento se les pide ingresarlos. Para esto debemos modificar tanto las vistas como los controladores.

Una vez generados los controladores, abrimos el archivo `registrations_controller.rb` y descomentamos (o reemplazamos todo directamente) tal que queda lo siguiente:

```
class Users::RegistrationsController < Devise::RegistrationsController
  before_action :configure_sign_up_params, only: [:create]
  before_action :configure_account_update_params, only: [:update]

  protected

  def configure_sign_up_params
    devise_parameter_sanitizer.permit(:sign_up, keys: [:username])
  end

  def configure_account_update_params
    devise_parameter_sanitizer.permit(:account_update, keys: [:username])
  end
end
```

De esta forma decimos que tanto para el sign up como para el update considere los parámetros contenidos en `keys` que en este caso es el username.

Finalmente es necesario agregar a los formularios en el directorio `views/devise/registration` el input para el username:

```
<div class="field">
  <%= f.label :username %><br />
  <%= f.text_field :username, autocomplete: "username" %>
</div>
```

Control de usuario registrado en vistas 😊

Es posible saber si un usuario esta logueado en la aplicación con un simple `current_user`, que se puede ocupar de la siguiente forma en una vista:

```
<% if current_user %>
  <%= link_to "Cerrar Sesion", destroy_user_session_path, method: :delete %>
<% else %>
  <%= link_to "Registrarse", new_user_registration_path %> |
  <%= link_to "Iniciar Sesion", new_user_session_path %>
<% end %>
```

Consideraciones ⚠

- Recuerden al momento de hacer deploy en Heroku que deben correr las migraciones:

```
heroku run rails db:migrate
```

Creado por Alberto Busch, Daniel Galloso, Pablo Olea, Jerónimo Salazar