

# Ayudantía Devise

IIC2143

# ¿Qué es una “Gema”?

En Ruby, las “Gemas” son:

1. Paquetes de códigos que permiten simplificar algunas tareas
2. La mayoría son **OpenSource**.
3. En **rubygems.org** pueden encontrar más información.

# Devise



## ¿Qué es?

Es una Gema de Ruby

## ¿Para que sirve?

Facilita la implementación de la autenticación para nuestras aplicaciones en Rails.

## Modularity concept



*"Usa solo lo que necesitas"*

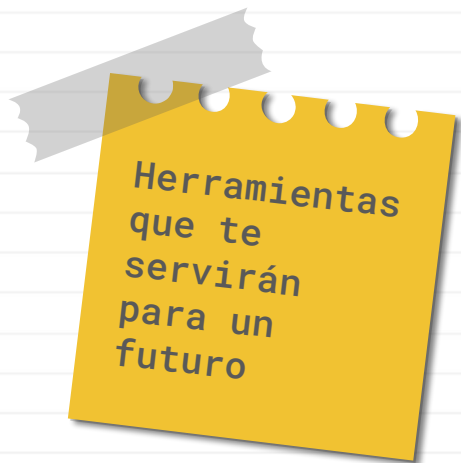
Son 10 módulos en total  
Entre ellos están:

- Registros
- Tracking
- Validación

## Más información

<https://github.com/heartcombo/devise>

## Helpers incluidos



**current\_user**



**user\_session**



**user\_signed\_in?**



**before\_action  
:authenticate\_user!**



01

# Setup

Pasos a seguir . . .


## Pasos a seguir

1. **Agrega la gema** a tu Gemfile
2. Descomenta la **gema bcrypt** para encriptar las contraseñas
3. Para instalar, correr en la terminal **bundle install**

## Gemfile

```
1 gem 'devise', '~> 4.7', '>= 4.7.3'
```

```
1 gem 'bcrypt', '~> 3.1.7'
```



## Pasos a seguir

Para instalar devise en nuestro proyecto corremos:

*rails generate devise:install*

De las 4 instrucciones, solo haremos la primera y tercera . . .

1. Copiar la siguiente línea para setear la url base en:

*config/environments/development.rb*

```
config/environments/development.rb

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

Para el ambiente de producción es distinto:

*config/environments/production.rb*

```
config/environments/production.rb

config.action_mailer.default_url_options = { host:
'url-de-app-heroku' }
```

## Pasos a seguir

3. Agregamos las alertas en el layout de las vistas del proyecto:

*app/views/layouts/application.html.erb*

```
app/views/layouts/application.html.erb

<body align='center'>

  <p class="notice"><%= notice %></p>
  <p class="alert"><%= alert %></p>

  <%= yield %>

</body>
```



Book Title. P52





02

## Devise Models

Preparamos el modelo *users*  
para trabajar con devise . . .

# Pasos a seguir

## 1. Ejecutamos:

```
bash  
rails g devise user
```

Esto crea la migración asociada al modelo *user*.

## 2. Ejecutamos:

```
bash  
rails db:migrate
```

Esto actualiza la base de datos en base a las migraciones nuevas.

### 3. Ejecutamos:

```
bash  
rails g migration AddUsernameToUsers username:string
```

Esto agrega el atributo *username* al modelo *users*.

### 4. Ejecutamos:

```
bash  
rails g migration AddPhoneToUsers phone:string
```

Esto agrega el atributo *phone* al modelo *users*.

### 5. Ejecutamos:

```
bash  
rails g migration AddAgeToUsers age:integer
```

Esto agrega el atributo *age* al modelo *users*.

### 6. Ejecutamos otra vez:

```
bash  
rails db:migrate
```

Esto actualiza nuevamente la base de datos.



03

# Devise Controllers

Creamos los controladores y  
configuramos las rutas para  
devise . . .

# Pasos a seguir

## 1. Ejecutamos:

```
bash  
rails g devise:controllers users
```

Esto crea los controladores que posee la gema devise.

## 2. Agregamos estas rutas al archivo *config/routes.rb*:

```
config/routes.rb  
  
devise_for :users, :controllers: {  
  sessions: 'users/sessions',  
  registrations: 'users/registrations'  
},  
path: '', path_names: {  
  sign_in: 'login',  
  sign_out: 'logout', sign_up: 'register'  
}
```

Esto agrega las rutas de devise para poder crear/actualizar un perfil de usuario.

## TIP

Podemos revisar las rutas que existen en nuestro proyecto con el comando *rails routes*

Prefix	Verb	URI Pattern	Controller#Action
photos	GET	/photos(.:format)	photos#index
	POST	/photos(.:format)	photos#create
new_photo	GET	/photos/new(.:format)	photos#new
edit_photo	GET	/photos/:id/edit(.:format)	photos#edit
photo	GET	/photos/:id(.:format)	photos#show
	PATCH	/photos/:id(.:format)	photos#update
	PUT	/photos/:id(.:format)	photos#update
	DELETE	/photos/:id(.:format)	photos#destroy

3. **Descomentamos** las siguientes líneas del archivo *app/controllers/users/registrations\_controller.rb*:

```
app/controllers/users/registrations_controller.rb

before_action :configure_sign_up_params, only: [:create]
before_action :configure_account_update_params, only: [:update]
```

Esto para que el controlador ejecute las funciones que permiten añadir los nuevos parámetros al modelo users.

```
app/controllers/users/registrations_controller.rb

# If you have extra params to permit, append them to the sanitizer.
def configure_sign_up_params
  devise_parameter_sanitizer.permit(:sign_up, keys: [:username, :phone, :age])
end

# If you have extra params to permit, append them to the sanitizer.
def configure_account_update_params
  devise_parameter_sanitizer.permit(:account_update, keys: [:username, :phone, :age])
end
```

Además descomentamos estas funciones y añadimos los parámetros nuevos del modelo User. Esto para que Devise maneje el crear/actualizar perfil de usuario con los nuevos atributos.

A graphic of a spiral-bound notebook with a white page, a red cover, and a green background. The page has a spiral binding at the top. On the left side, there are two horizontal tabs, one yellow and one orange. In the center of the page, the number '04' is displayed in a large, bold, black font, surrounded by a light green circular arrow. Below the number, the title 'Devise Views' is written in a bold, red font. Underneath the title, a paragraph of text in a monospaced font reads: 'Creamos y adaptamos las vistas que existen por defecto en devise. . .'.

04

## Devise Views

Creamos y adaptamos las  
vistas que existen por  
defecto en devise. . .



# Disponibilizar vistas

En *application.html.erb* podemos agregar un control de sesión:

app/views/layouts/application.html.erb

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