

94. Edit and update: update existing articles

duda

¿Las rutas del resources son pre-definidas?

```
g > routes.rb
Rails.application.routes.draw do
  root "pages#home"
  get "about", to: "pages#about"
  resources :articles, only: [:show, :index, :new, :create, :edit, :update]
end
```

Ruta para editar

```
(i) localhost:3000/articles/1/edit
```

->patch handles the :update function

Automatic form to update articles

```
<%= form_with(model: @article, local: true) do |f| %>
```

Pasas una instancia del artículo y harás una form para actualizarlo. Ruby es inteligente y como estamos en la ruta edit, automáticamente hará submit a ese artículo en específico.

```
# redirect_to article_
redirect_to @article
```

Al parecer se puede hacer un redirect usando una instancia del modelo.

The process of editing an existing article and updating the article in the articles table utilizes the edit and update actions. The standard process is as follows:

1. Expose edit and update routes.
2. Add edit and update actions in the articles controller.
3. Create an edit template (form) in the app/views/articles folder.
4. Use the edit action to find the article to edit, display the existing article details in the edit form.
5. Use the update action to find the article in the db. Whitelist the new title and description fields and if there are no validation errors, then update the article in the articles table with the new data.

96. Delete: delete articles

```
› README.md
The front-end: Destroying (deleting) articles

REST – Representational state transfer – mapping HTTP verbs (get, post,
put/patch, delete) to CRUD actions

resources provides REST-ful routes to Rails resources
```

Rest is basically a mapping of http verbs with back end CRUD actions

```
g > routes.rb
Rails.application.routes.draw do
  root "pages#home"
  get "about", to: "pages#about"
  resources :articles
end
```

resources keyword in ruby means all the restful routes for a resource

Agregar link de accion en ruby

```
<td><%= link_to 'Delete', article_path(article), method: :delete %>
</td>
```

link_to, nombre del enlace, despues path a donde quieras mandarlo.

```
--[ Route 10 ]-----
valid Prefix           | 
Verb               | DELETE
URI                | /articles/:id(.:format)
Controller#Action | articles#destroy
--[ Route 11 ]-----
```

Ese delete es sacado de aqui, del verbo, ...?

En rails 7 es:

```
<td><%= link_to 'Delete', article_path(article), data: { turbo_method: :delete } %></th>
```

De donde salen esos paths

```
--[ Route 6 ]-----
Prefix           | edit_article
Verb             | GET
URI              | /articles/:id/edit(.:format)
Controller#Action | articles#edit
--[ Route 7 ]-----
```

Salen del prefijo, solo le agregar _path y se te devuelve el path

```
<td><%= article.description %></td>
<td><%= link_to "Edit", edit_article_path(article) %></th>
<td><%= link_to "Update", article_path(article) %></th>
<td><%= link_to 'Delete', article_path(article), data: { turbo_method: :delete } %></th>

```

Y si ocupas el id, pues entre parentesis le pasas el modelo o [modelo].id

Deleting articles can be accomplished using the destroy action in the articles controller. The standard deletion process would look like below:

1. Expose the destroy route.
2. Add the destroy action in the articles controller.
3. Find the article to delete and delete it using the destroy method within the destroy action.

98. User Interface - add layout links

```
<th colspan = "3">Actions</th>
```

Hace que la columna de la tabla use 3 celdas

Rails 7 redirect on delete

```
def destroy ()
  @article = Article.find params[:id]
  @article.destroy
  redirect_to articles_path, status: :see_other
end
```

303 - See Other

Rails HTTP Status code :see_other

When to use status code 303

The response to the request can be found under another URI using a GET method. When received in response to a POST (or PUT/DELETE), it should be assumed that the server has received the data and the redirect should be issued with a separate GET message.

100. DRY (Don't Repeat Yourself) code - refactoring and partials

Partial

Moveremos parte de la vista en otra vista, a esto se le llama partial.

```
before_action
```

```
  before_action :set_article, only: [:show, :edit, :update, :destroy]
```

Para esos métodos correrá un método que obtenga el artículo por id.

La keyword private afectará a todos los métodos debajo de ella.

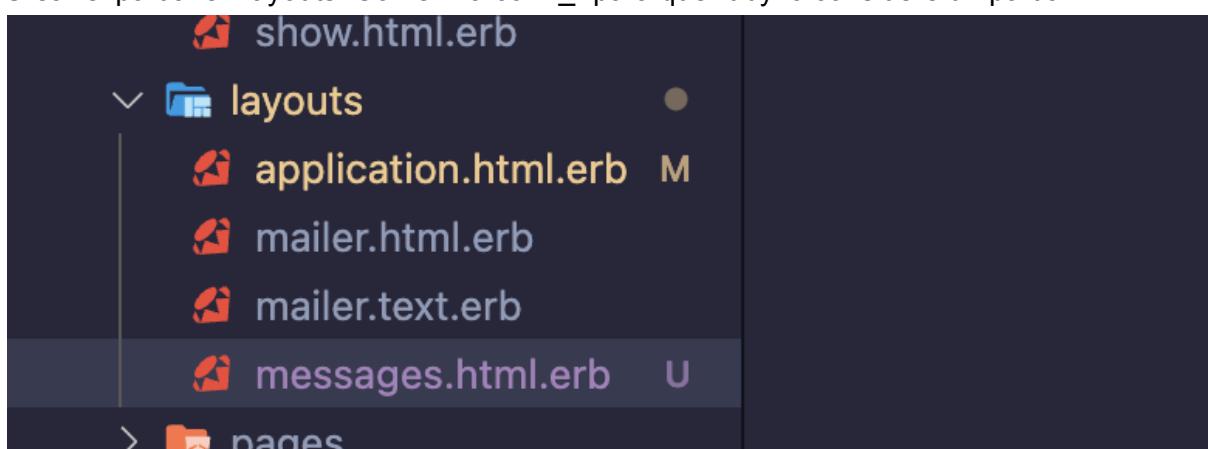
```
private
```

```
def set_article
  @article = Article.find(params[:id])
end
```

```
def article_params
  params.require(:article).permit(:title, :description)
end
```

Partial

Crear el partial en layouts. Comeienza con “_” para que ruby lo considere un partial



Llamarlo de esta forma.

```
<body>
  <%= render 'layouts/messages'%>
  <%= yield %>

```

```
3
4   <%= 'form'%> You, 1 second
5
```

Como ese partial esta en el mismo folder, no necesita poner el full path.

En el layout folder, es especial y no puedes ponerlo en la misma forma que el form.

```
<%= form_with(model: @article, local: true, data: { turbo: false } ) do |f| %>
```

Ese código del form funciona tanto para el new como el update.

Este form está respaldado por un modelo.

Detectara la ruta ?

102. Production deploy and wrap up section 4

Actualizar proyecto de heroku

Hara cambios del gemfile hacia gemfile.lock que sera leido por heroku

```
Exiting  
$ bundle install --without production
```

Ahora a pushear el codigo a heroku

```
git push heroku master
```

La bd en produccion no ha sido creada o modificada, pero podemos correr los arhicos de migracion.

Hay que hacer que heroku corra este comando

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

Abrir la app en el navegador

```
$ heroku open
```

We used the following steps to deploy our application to production (using Heroku). If you have not set up a Heroku account and deployed the application at the end of section 3 (including all the setup steps like installing Heroku CLI and creating a heroku app), you are encouraged to do so now before moving forward. If you would rather not deploy to production, then you can safely skip this step.

104. Introduction to Section 5 and styling

Usaremos bootstrap
existe tambien: materialize y semantic.

Para aprender mas sobre html

developer.mozilla.org/en-US/docs/Learn/HTML

Este sitio web también está muy bien



learn.shayhowe.com/html-css/

Sitio web recomendado por el autor de hacer mock ups para saber como nuestro sitio se vera

balsamiq®

PRODUCT LEARN TO WIREFRAME COMPANY CONTACT US BUY LOG IN Search...

Life's too short for bad software!

With [Balsamiq Wireframes](#), anyone can design great user interfaces.

Since 2008, our mission is to help rid the world of bad user interfaces.

El usa la versión de escritorio

Show_process

All Assets Big Buttons Common Containers Forms Icons iOS Layout Markup Media Symbols Text

Mockups

New Mockup 1 articles_listing New Mockup 3

A Web Page

Alpha Blog Articles Users Login

Alpha Blog Sign up today!

Some footer text

New Mockup 3

Notes Click here to edit notes.

Alternate Versions Click the '+' to create an alternate version of this mockup.

Pero este no es gratis

podemos buscar uno en internet gratuito o hacerlo en papel.

[Home](#) | [About](#) | [FAQ](#) | [Advertise](#) | [Contact](#)

Banner 460x60

[Sign Up or Log In](#)

[Home](#) | [About Us](#) | [Contact Us](#) | [Articles](#) | [Sign In](#)

Welcome to ...
Webinar Name is a community for users to showcase, discuss and learn everything about Webinars Name.
[Click here to find out more](#)

Advertisement
Banner 280x215

Post Title 1
11 November 2009
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

[12 comments](#)

Post Title 1
11 November 2009
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

[12 comments](#)

Post Title 1
11 November 2009
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

[12 comments](#)

Post Title 1
11 November 2009
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

[12 comments](#)

Post Title 1
11 November 2009
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

[12 comments](#)

Sponsors

Advertise Here	Advertise Here
Advertise Here	Advertise Here

Features Article
Post Title 1
11 November 2009
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

[12 comments](#)

Recent Discussion

- Username said:**
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- Username said:**
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- Username said:**
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Widget 1
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

Widget 2
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

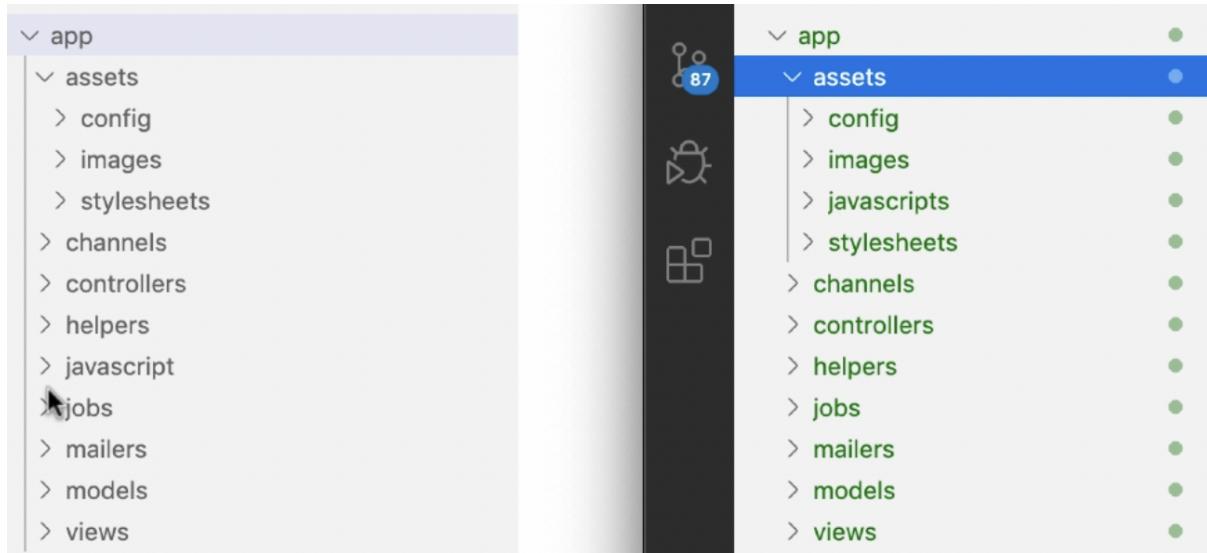
Widget 3
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

Widget 4
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit animi ut est.

Copyright Your Company © 2009. All Rights Reserved.

105. Learn to use Bootstrap

106. Install Bootstrap, asset pipeline, JavaScript, webpack - Rails 6 vs 5



En 6 esta en app el folder en otra está en assets

En rails 5 js esta manejado por webpack.

Hay una gem "webpacker". en 6, antes estaba manejado por el asset pipeline.

Así que antes estaba en assets porque era manejado por el manejador de assets de rails. Y ahora en su propio folder porque es manejado por webpack.

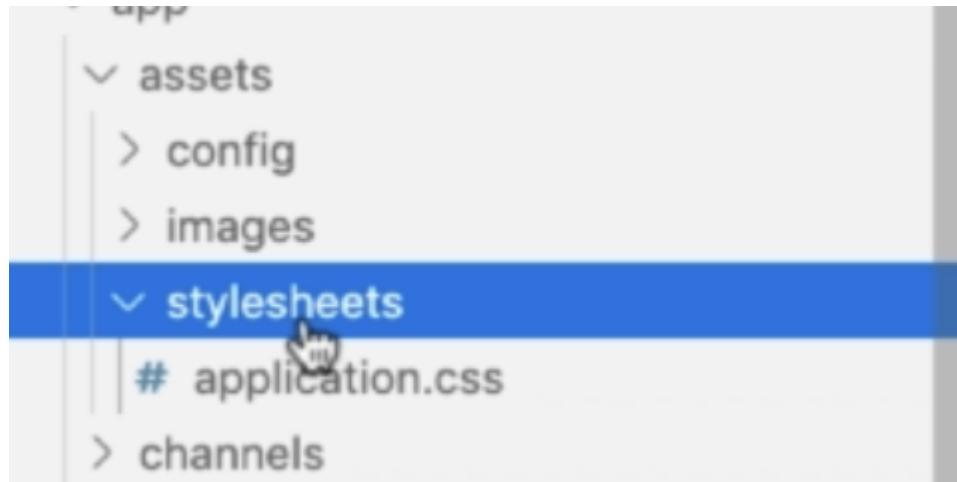
6:

```
application.html.erb — alpha-blog
Gemfile application.html.erb
app > views > layouts > application.html.erb
<!DOCTYPE html>
<html>
  <head>
    <title>AlphaBlog</title>
    <%= csrf_meta_tags %>
    <%= csp_meta_tag %>

    <%= stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-track': 'reload' %>
  </head>

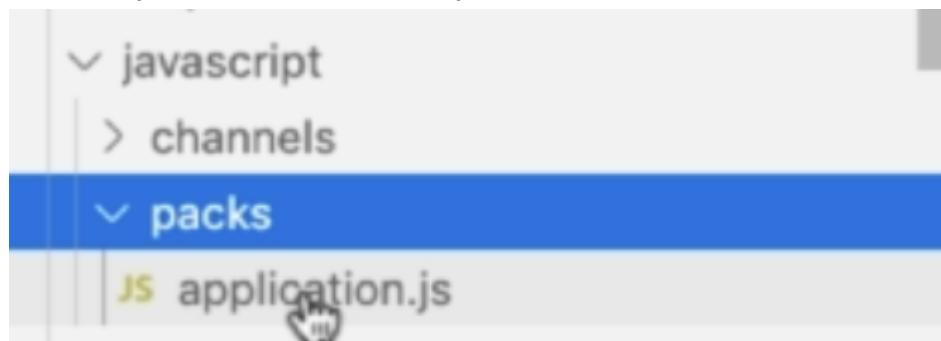
  <body>
    <%= render 'layouts/messages' %>
    <%= yield %>
  </body>
</html>
```

styleseeth_link_tag hace referencia a



Todo el css que crees en ese folder sera servidor por ese manifest.

El otro de js hace referencia a ese js



De nuevo, todo el js estara disponible en tu app a travez de este manifesto.

```
<%= csp_meta_tag %>

<%= stylesheet_link_tag    'application'
  <%= javascript_include_tag 'application'
</head>
```

un poco diferente...

Instalar bootstrap en rails 6

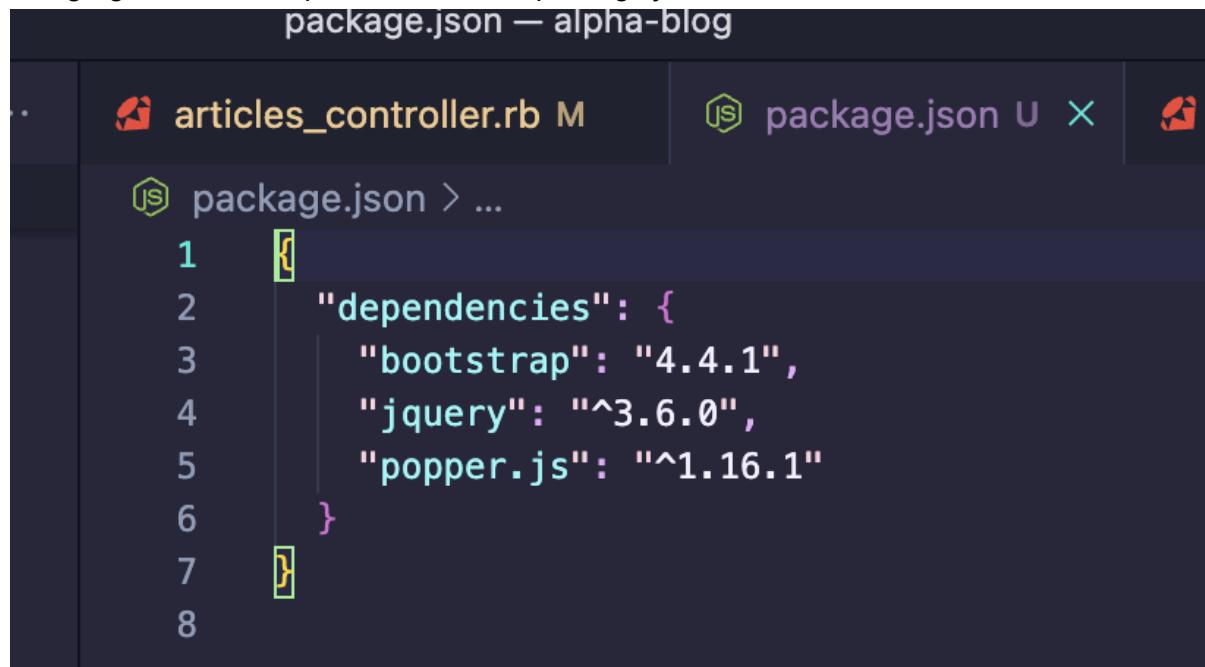
Ei tiene un blog donde lo explica

<https://www.mashrurhossain.com/blog/rails6bootstrap4>

Lo instala a travez de yarn

```
/Users/mashrurhossain/rails_projects_2020/rails_
$ yarn add bootstrap@4.4.1 jquery popper.js
yarn add v1.17.3
```

Se agregaran como dependencias en tu package.json



```
package.json — alpha-blog

articles_controller.rb M package.json U X

package.json > ...
1  {
2    "dependencies": {
3      "bootstrap": "4.4.1",
4      "jquery": "^3.6.0",
5      "popper.js": "^1.16.1"
6    }
7  }
8
```

Ahora tenemos que agregar esta linea dentro de application.css

```
*  
*= require bootstrap  
*= require_tree .  
*= require_self  
*/
```

Now onto CSS

Go to app/assets/stylesheets/application.css and add the following line above the require_tree and require_self lines:

```
= require bootstrap
```

Ahora sigue poner el js de bootstrap

Para ello tenemos que hacer que nuestra app entienda jquery y popper extensions.
Solo entiende plain js.

Cambiamos la config de webpack

Make the config/webpack/environment.js file look like below:

```
config > webpack > JS environment.js > Popper  
const { environment } = require('@rails/webpacker')  
  
const webpack = require("webpack")  
  
environment.plugins.append("Provide", new webpack.ProvidePlugin({  
  $: 'jquery',  
  jQuery: 'jquery',  
  Popper: ['popper.js', 'default']  
})  
  
module.exports = environment
```

Ahora agregamos un import en application.js (similar que como hicimos con css)
Agregamos el import “bootstrap”

The screenshot shows a code editor with two tabs open: 'environment.js' and 'application.js'. The 'environment.js' tab is active, displaying the following code:

```
app > javascript > packs > application.js
1 // This file is automatically compiled by Webpack, along with
2 // present in this directory. You're encouraged to place your
3 // a relevant structure within app/javascript and only use its
4 // that code so it'll be compiled.
5
6 import Rails from "@rails/ujs";
7 import Turbolinks from "turbolinks";
8 import * as ActiveStorage from "@rails/activestorage";
9 import "channels";
10
11 Rails.start();
12 Turbolinks.start();
13 ActiveStorage.start();
14
15 import "bootstrap";      You, 1 second ago • Uncommitted changes
```

El código del enviroment no funcionó, pero modificandolo hacia, sí funciona:

Saving this for later since I'm back for a second time.

Steps for setting up Bootstrap

Terminal/Powershell:

```
yarn add bootstrap @popperjs/core
```

```
config/webpack/environment.js:
1 const webpack = require("webpack")
2
3 environment.plugins.append("Provide", new webpack.ProvidePlugin({
4   Popper: ['popper.js', 'default']
5 }))
6

app/javascript/packs/application.js:
import "bootstrap"

app/assets/stylesheets/application.css:
*= require bootstrap
```

Run Rails server and test to make sure it's working.

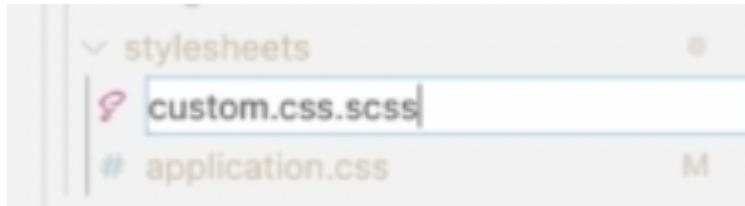
Extra - Import all of Bootstrap CSS libraries for modifying what we want:

Create `custom.css.scss` file under `app/assets/stylesheets` and add `@import 'bootstrap/dist/css/bootstrap'` at the top of the file.

Test modify some bootstrap css.

Si queremos agregar custom styles...

creas tu css dentro de stylesheets



Si queremos sobreescribir clases de bootstrap primero tenemos que importarlo.

```
p > assets > stylesheets > ⚡ custom.css.scss  
1   @import 'bootstrap/dist/css/bootstrap';
```

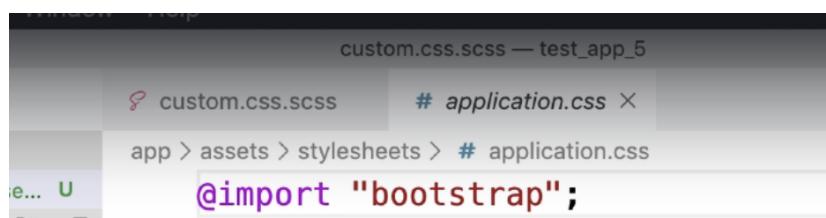
108. Install Bootstrap in Rails 5 (or earlier versions)

Tienes que instalarlo por gem. En gemfiles

```
# gem 'mini_racer', platforms: :r  
gem 'bootstrap', '~> 4.4.1'  
gem 'jquery-rails'
```

Tambien agregar jquery

Correr bundle-install

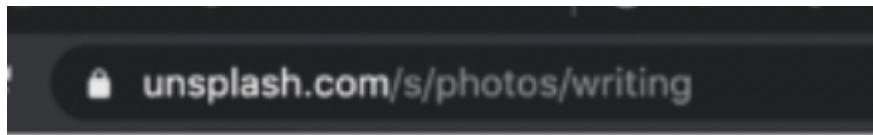


Despues crea tu propio css.

En el js application agregar estas lineas

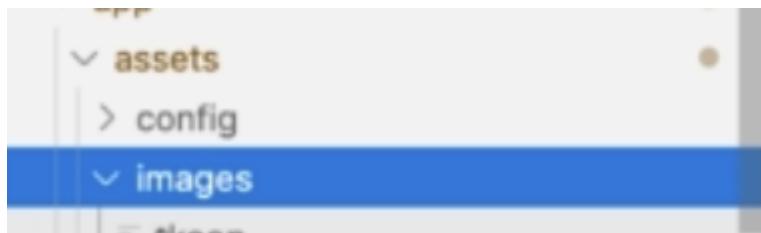
```
//= require jquery3  
//= require popper  
//= require bootstrap
```

110. Build homepage



El encuentra imagenes desde ahí.

Pones imágenes en imágenes de assets



Como está en carpeta images no necesitas darle ruta

```
.jumbotron {  
  background-image: asset-url('jumbo-copy.jpg')  
}
```

Link_to

```
nav class="navbar navbar-expand-lg navbar-light bg-light">  
  <%= link_to "ALPHA BLOG", root_path, class: "navbar-brand", id: "logo"%>  
  button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" data  
  Link to con clase y id.
```

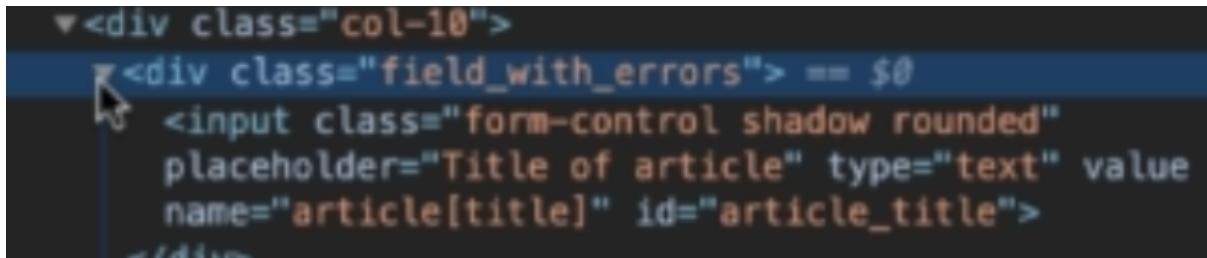
114. Style articles index page

```
<%= link_to article.title, article_path(article),  
= truncate(article.description, length: 100) %></p>  
  button-primary">Go somewhere</a>
```

Helper date method

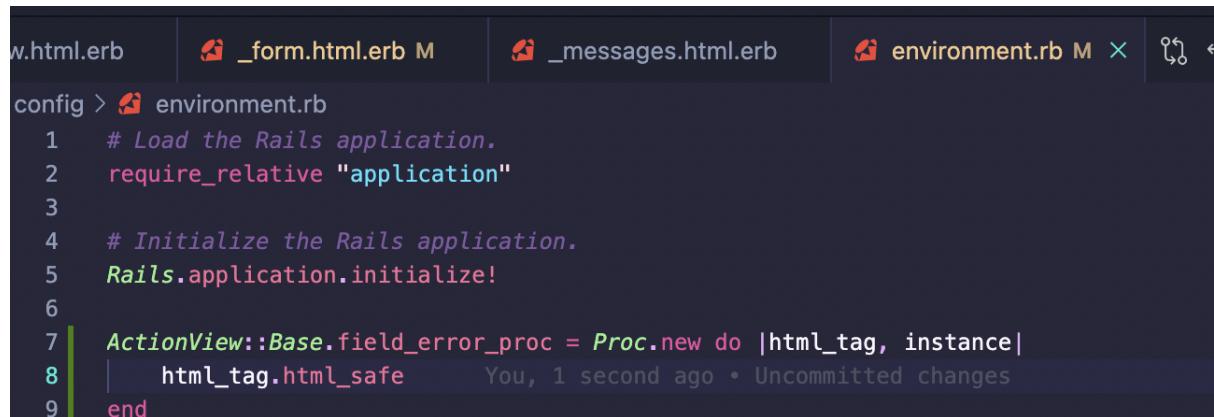
```
<div class="card-footer text-muted">  
  <small>Create <%= time_ago_in_words(article.created_at)%> ago,  
  edited <%= time_ago_in_words(article.updated_at)%> ago</small>  
</div>
```

118. Style validation and flash messages



```
<div class="col-10">
  <div class="field_with_errors"> == $0
    <input class="form-control shadow rounded"
      placeholder="Title of article" type="text" value
      name="article[title]" id="article_title">
</div>
```

En el form se agrega un espacio automatico al fallar por como funciona, lo vamos a aquitar.



```
w.html.erb | _form.html.erb M | _messages.html.erb | environment.rb M ✘ ↵
config > environment.rb
1   # Load the Rails application.
2   require_relative "application"
3
4   # Initialize the Rails application.
5   Rails.application.initialize!
6
7   ActionView::Base.field_error_proc = Proc.new do |html_tag, instance|
8     html_tag.html_safe
9   end
```

En enviroment de configuración, ponemos eso y se deshabilita esa sección de errores.

Reinicia el servidor después de hacer cambios en archivo de configuración.

120. Style show view

```
<p class="card-text"><%= simple_format(@article.description) %></p>
```

```
simple_format(text, html_options = {}, options = {})
public
```

Returns text transformed into [HTML](#) using simple formatting rules. Two or more consecutive newlines(\n\n or \r\n\r\n) are considered a paragraph and wrapped in <p> tags. One newline (\n or \r\n) is considered a linebreak and a
 tag is appended. This method does not remove the newlines from the text.

You can pass any [HTML](#) attributes into `html_options`. These will be added to all created paragraphs.

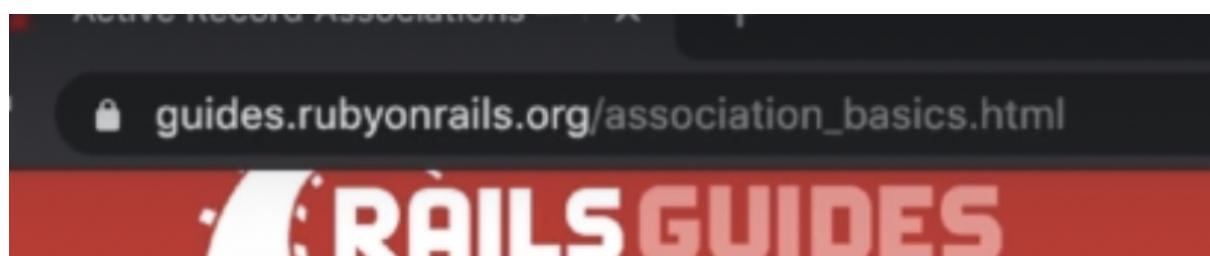
To deploy your application to heroku, first ensure you have checked `git status` and that all your changes have been committed to your git repo. If there are untracked/uncommitted changes then follow the usual commit route:

```
1 | $ git add -A  
2 | $ git commit -m "Appropriate commit message"  
3 | $ git push origin master # to deploy to your GitHub or other code  
repo
```

Then deploy your changes to heroku and run any pending migrations there:

```
1 | $ git push heroku master  
2 | .....after deployment completes successfully....  
3 | $ heroku run rails db:migrate
```

124. Introduction to section 6: users, associations, ERD and more

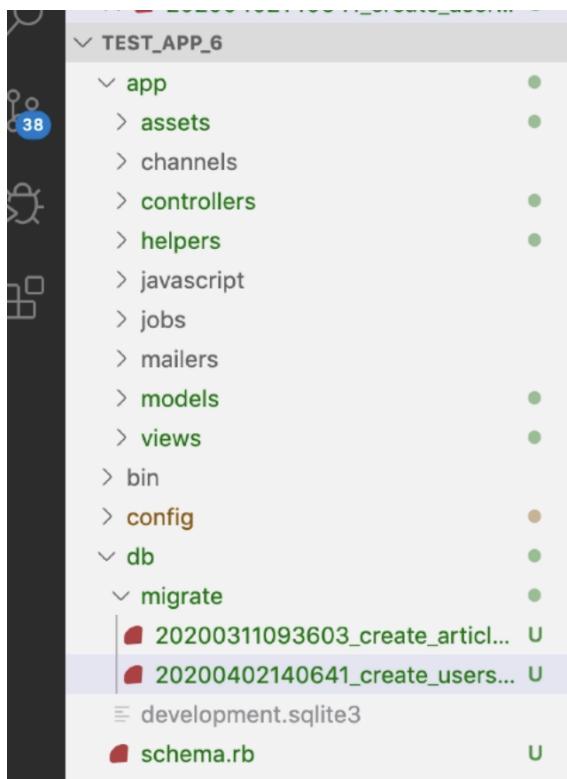


Esta página te habla sobre las asociaciones en rails.

125. One-to-many associations demo with the rails console

Create resource

```
rails generate scaffold User username:string
```



migramos...

- Creamos migracion, agregamos el id
- en los modelos agregamos el belongs_to y has_many

Rails es inteligente al crear y en lugar de poner user.id podemos pasar solamente el user

```
> Article.create(title: "some article", description: "desc  
some article", user: user_1)
```

Ver los articulos de un usuario

```
user_1.articles
```

Puedes crear los articulos de un usuario asi

```
|_at: "2020-04-02 14:08:14">  
> user_1.articles.build(title: "some new article", descrip  
ription of some new article")
```

Agregar un usuario a articulo usando el operador de pala

```
|at: "2020-04-02 14:08:24">  
> user_2.articles << article
```

126. Create users

Existen multiples workflows para trabajar con multiples branches en un equipo

```
migrate > 20220812193849_create_users.rb
class CreateUsers < ActiveRecord::Migration[6.1]
  def change
    create_table :users do |t|
      t.string :username
      t.string :email
      t.timestamps
    end
  end
end
```

El timestamps agrega la fecha de modificacion y creacion

Detalles

El modelo no require atributos para poder agregar o que funcione correctamente.

Tambien si hicimos una migracion en otra branch y despues cambiamos a otra branch, la migracion seguira estando ahí.

128. Add user validations

```
/ README.md
Resource
- Users - create users table and model
  - add validations
    * username must be present and unique, min 3 max 25
    * email address must be present and unique, max 105
    * email must be valid email format, check with email regex
```

```
Associations
- One-to-many
  between users and articles
```

```
REST for users
```

```
Authentication
- Login using secure password
```

```
Restriction of actions
- Based on logged in/logged out state
```

```
Security
- A verificación usando un rechazo que es la
```

En la sig pagina puedes acordarte y ver todas las validaciones

The screenshot shows a browser window with three tabs open:

- Active Record Validations — Ru
- Rubular: a Ruby regular expres
- udemyrailscourse/alpha-blog-1

The main content area displays the "1 Validations Overview" section of the guide. It includes a code snippet and a note about validation rules.

```
class Person < ApplicationRecord
  validates :name, presence: true
end

Person.create(name: "John Doe").valid? # => true
Person.create(name: nil).valid? # => false
```

As you can see, our validation lets us know that our `Person` is not valid without a `name` attribute. The second `Person` will not be persisted to the database.

On the right side, there is a sidebar with a table of contents:

• errors
2. Validation
• accept
• validad
• confir
• exclus
• format
• inclus
• length
• numeri
• presen
• absenc

Acomodar codigo

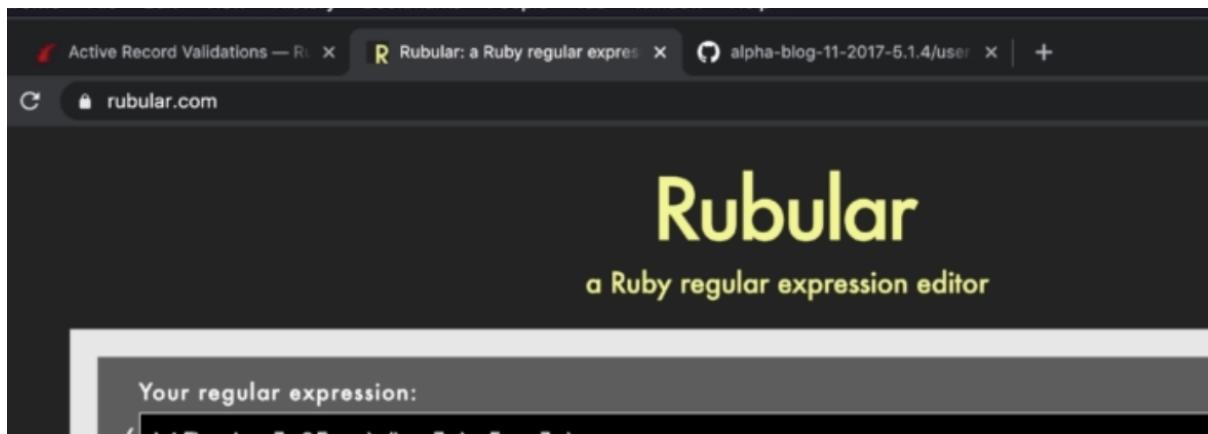
Si es muy largo ponlo en diferentes linea, hazlo despues de la coma

```
models > user.rb
class User < ApplicationRecord
  validate :username, present: true,
            uniqueness: { case_sensitive: false }, length: { minimum: 3, maximum: 25 }
  validate :email, present: true,
            uniqueness: { case_sensitive: false } length: { maximum: 105 } You, now
end
```

Quizo poner la expresion regular en una constante

```
p > models > user.rb
class User < ApplicationRecord
  validates :username, presence: true,
            uniqueness: { case_sensitive: false },
            length: { minimum: 3, maximum: 25 }
  VALID_EMAIL_REGEX = /\A[\w+\.-]+@[a-z\d\.-]+\.[a-z]+\z/i
  validates :email, presence: true,
            uniqueness: { case_sensitive: false },
            length: { maximum: 105 },
            format: { with: VALID_EMAIL_REGEX }
end
```

Página para checar expresiones regulares de ruby



130. One to many association

- Agregar el campo de id en artículos
- correr migración
- agregar el belongs_to y has_many

```
o > models > article.rb
1   class Article < ApplicationRecord
2     belongs_to :user      You, now
3
4
5
```

Singular

```
> models > user.rb
1   class User < ApplicationRecord
2     has_many :articles
3
4
5
```

plural

Checar la otra tabla

```
article.user
user.articles
```

Agregar a usuarios artículos

```
2>]
> user.articles << Article.all
Load (0.3ms)  SELECT "articles" * FROM "arti
```

Update all

```
development environment (Rails 6.0.2.1)
> Article.update_all(user_id: User.first.id)
   (0.3ms)  render 'new'
```

134. Alter object state before_save

Para poner en minúsculas el email podemos hacer

```
before_save { self.email = email.downcase}
```

Dentro del user model. usamos self porque esta haciendo referencia al email del objeto a guardar.

136. Add secure password

Devise -> gem popular para autentificación.

Pero para que sea mas facil de usar y entender que esta pasando no la usaremos y la crearemos manualmente

Hay que hashear las contraseñas para que sean mas seguras

users table	
e	password
m	\$2a\$12\$eOvhIQY4zy/j79bMvtDWgub8zPf/kz/
as	vwPda5VileM9L6aaY.a4BC

No podemos convertir un hash a palabra de vuelta.

No es prueba de balas pues si el hacker consigue el hash u sabe el algoritmo, puede hacer fuerza bruta y buscar ese hash en una bd de hashes. (Rainbow attack)

Salt

Laa solucion a un ataque arcoiris es el hash. **Es un pedazo random de texto que se agrega a cada contra antes de hashearla**

Descomentamos esta gema (para hacer hashes)

```
# Use Active Model has_secure_password
gem 'bcrypt', '~> 3.1.7'
```

y corremos **bundle install**

```
o > models > user.rb
1   class User < ApplicationRecord
2
3     before_save { self.email = email.downcase}
4
5     has_many :articles
6
7     validates :username, presence: true,
8           uniqueness: { case_sensitive: false }, length: {minimun: 3, maximum:25}
9
0     VALID_EMAIL_REGEX = /\A[\w+\.-]+@[a-z\d\-\-]+(\.[a-z\d\-\-]+)*\.[a-z]+\z/i
1     validates :email, presence: true,
2           uniqueness: { case_sensitive: false }, length: { maximum: 105},
3           format: {with: VALID_EMAIL_REGEX}
4
5     has_secure_password
6   end
7
```

Agregamos el campo de contraseña con **has_secure_password**

Despues le agregamos el campo “password_digest”

<https://api.rubyonrails.org/classes/ActiveModel/SecurePassword/ClassMethods.html>

De ahí saca del porque todo

Podemos referirnos al campo de password_digest como password aunque se llame de otra forma

BCrypt tiene el mecanismo de que cuando le pasamos un campo hash, este saber que parte es la salt

```
$2a$12$qo5Hc5Xf6Nbpt7Zg714enUE5KA2uQ8qwnI2vgf3Tv  
Q1G9cHo/Lyy"  
3.0.4 :004 > BCrypt::Password.create("password")  
=> "$2a$12$tLOMyPWiKmUAG55sFpeCIelUBlk/n.46xuaXXQtD7t  
P4M2vsF7wKG"  
3.0.4 :005 > password =  
=> "$2a$12$tLOMyPWiKmUAG55sFpeCIelUBlk/n.46xuaX...  
3.0.4 :006 > password.salt  
=> "$2a$12$tLOMyPWiKmUAG55sFpeCIe"  
3.0.4 :007 > [REDACTED]
```

Basicamente bcrypt usa esa salt como una llave, se la pone a la otra contra que comparamos y ve si coinciden.

Ahora nosotros no tenemos que usar bcrypt manualmente pues lo configuramos a que sea automatico

Ver si la contraseña es correcta

```
=> "$2a$12$EdxbRjHLSxkzJwoFxBcbbeULduxbzU08B0AljEQ.Ie  
3.0.4 :024 > user.authenticate("root")  
=>
```

El metodo regresa el usuario si true, regresa false si contra equivocada.

138. New user signup form

```
resources :articles
get 'signup', to: 'user#new'
end
```

recuerda que aunque en la bd se llama password_digest, el nombre virtual del atributo es password

Parámetros de partials

```
<div class="col-10">
  <%= render "shared/errors", obj: @article%>
  <%= form_with(model: @article, class: "shadow">
<ul>
  <% obj.errors.full_messages.each do |msg| %>
    <li><%= msg %></li>
  <% end %>
</ul>
```

El nuevo form dara error, porque todavia no tenemos la ruta del post

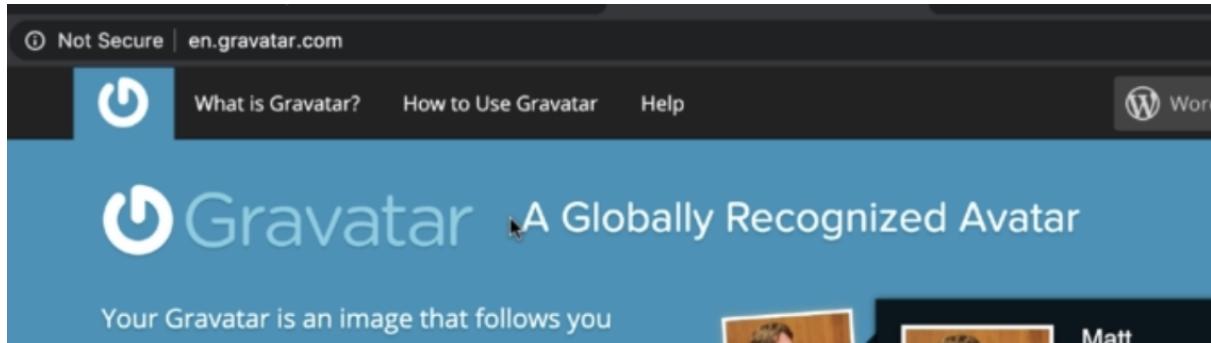
```
resources :articles
get 'signup', to: 'users#new'
post 'users', to: 'users#create'
end
```

u

```
resources :articles
get 'signup', to: 'users#new'
resources :users, except: [:new]
end
```

140. Create new users (back-end)

144. Show user and profile image



Imagenes de perfil. Asocias un email con una imagen. Si tienes una cuenta de wordpress puedes usarlo gratis.

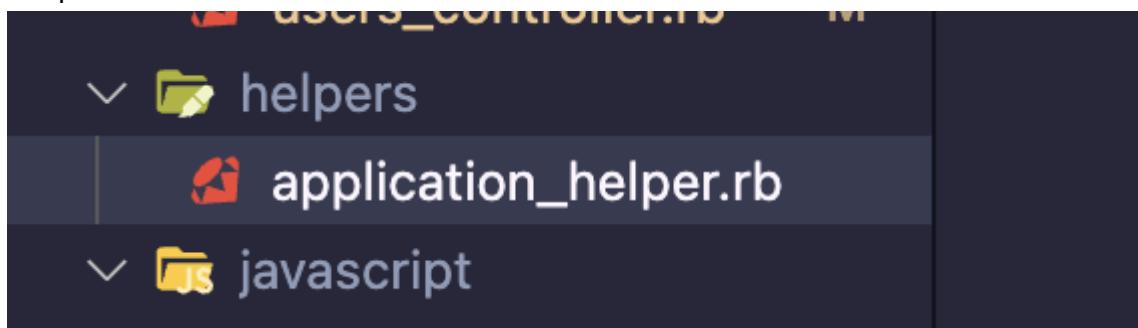
Funciona dándole un email, crea un hash, ese hash lo usas para hacer un gte y obtener la imagen.

<https://www.gravatar.com/avatar/d3c0f82202fb60604386603c682a1d50?s=200>

Helpers directory

Cualquier metodo que queramos usar en nuestras vsitas podemos meterlo en nuestro folder de helpers

Los ponemos dentro del



Helper image tag

```
image_src = "https://www.gravatar.com/avatar/#{\$hash}"  
return image_tag(gravatar_url, alt: user.username)
```

Parametro opcional?

```
module ApplicationController
```

```
  def gravatar_for (user, options = {size: 80})
    |   email_address = user.email.downcase
2
3   <%= gravatar_for @user, size: 200 %>
```

```
<%= render 'articles/article' %>
```

Como el partial esta en otra capeta usamos esto.

146. Add users index

El codigo para mostrar una card de usuario es similar al del articulo, asi que haremos copy paste y lo modificaremos.

Pluralize

```
  > class="card-text"><%= pluralize(user.articles.count, "article" %></p>
  => link_to "View" article_path(article) class: "btn btn-outline-success"
```

Si el numero es 1 o 0 sera singular, si muchos entonces plural.

148. Cleanup layout