

IIC2143

Ayudantía CRUD + Routing



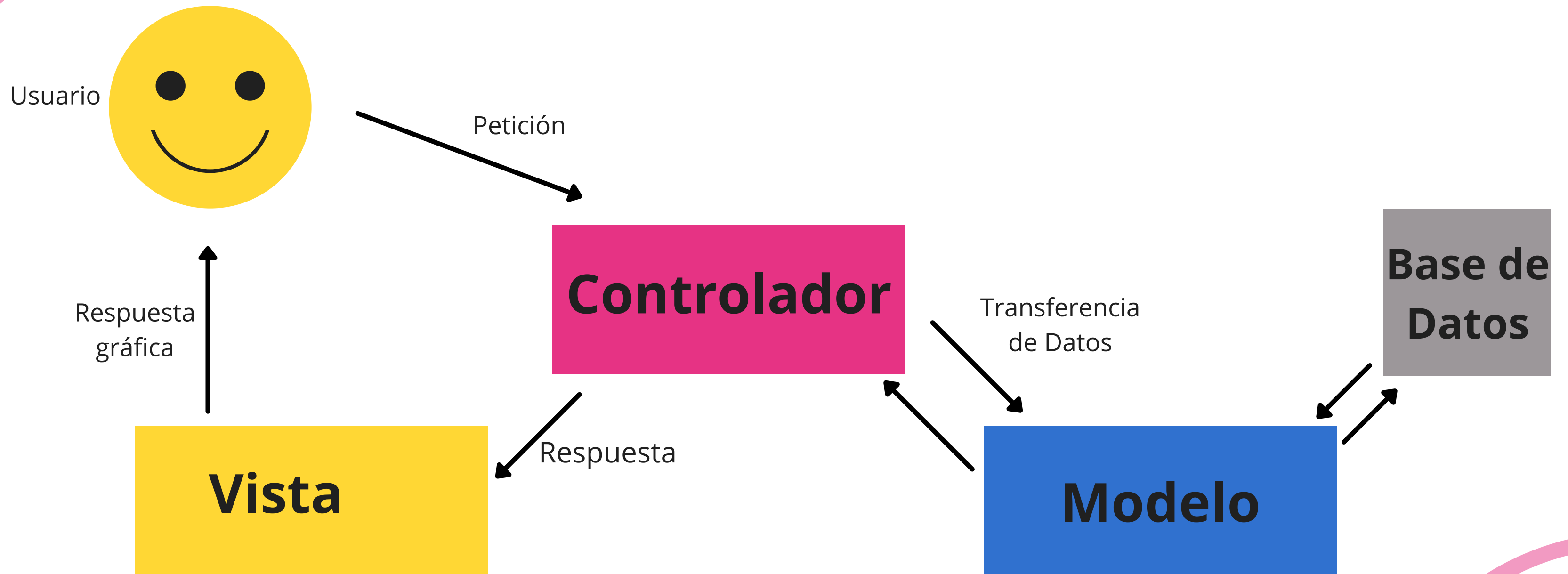


¿Que aprenderemos a hacer?



- Crear una aplicación básica con modelo vista y controlador
- Definir y utilizar rutas para la aplicación

Modelo-Vista-Controlador



CRUD

- CREATE: new, create
- READ: index, show
- UPDATE: edit, update
- DELETE: delete



●●● MODELO

- Guarda la información de sus atributos en una base de datos
- Nombre del modelo debe ir en singular





Crear Modelo

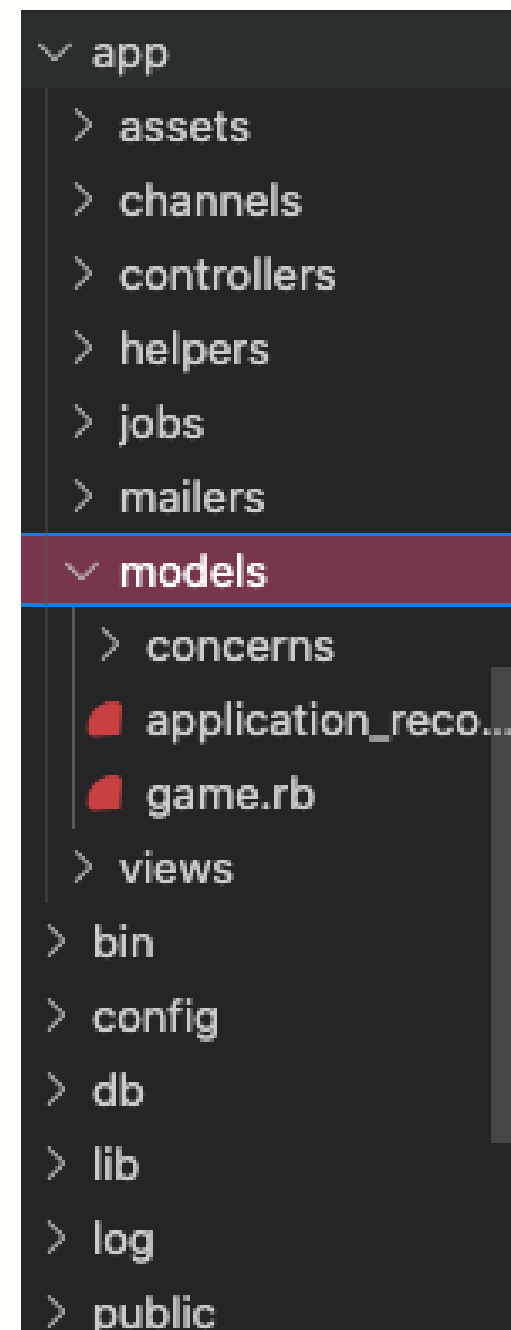
```
rails g model {Nombre_modelo} {atributo1}:{tipo_atributo1} {atributo2}:{tipo_atributo2} ...
```

- **Nombre_modelo:** mayúscula y singular
- **atributo i:** nombre del atributo
- **tipo_atributo i:** string, integer, boolean, etc.





Se crea el siguiente archivo



```
1  class Game < ApplicationRecord
2  end
3
4
5
```

Se crea el siguiente archivo



```
> bin
> config
✓ db
  ✓ migrate
    2021090316171...
    schema.rb
    seeds.rb
> lib
> log
> public
> storage
```

```
1 class CreateGames < ActiveRecord::Migration[5.2]
2   def change
3     create_table :games do |t|
4       t.string :name
5       t.float :ranking
6
7       t.timestamps
8     end
9   end
10 end
```




Migración

```
rails db:migrate
```

Se crea la tabla correspondiente al modelo creado y se puede trabajar con él mediante la aplicación.



CONTROLADOR

- Permite la interacción con la base de datos de una aplicación mediante las vistas
- El nombre del controlador debe ir en plural





Crear Controlador

```
rails g controller {Nombre_controlador} {metodo1} {metodo2} {metodo3} ...
```

- **Nombre_controlador:**
mayúscula y plural
- **metodo i:** nombre del metodo
(CRUD)



Se crea el siguiente archivo

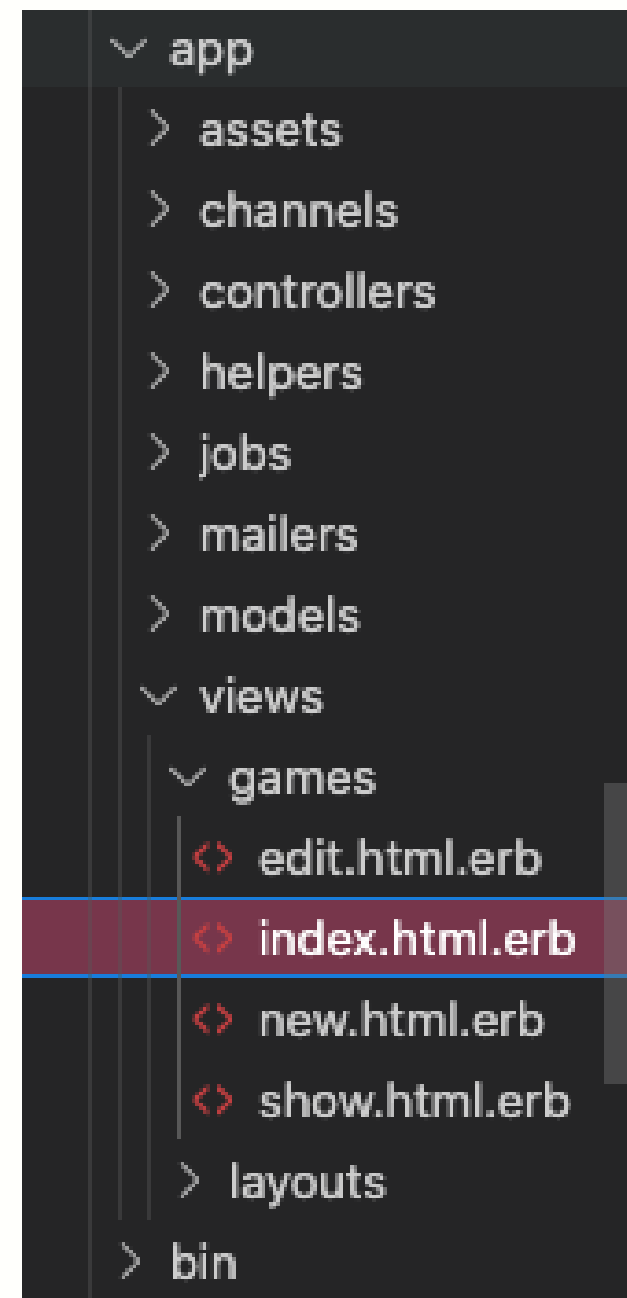


```
✓ app
  > assets
  > channels
  ✓ controllers
    > concerns
    application_cont...
    games_controlle...
  > helpers
  > jobs
  > mailers
  > models
  ✓ views
    > games
    > layouts
```

```
1  ✓ class GamesController < ApplicationController
2      def new
3      end
4
5      def index
6      end
7
8      def show
9      end
10
11     def edit
12     end
13 end
```



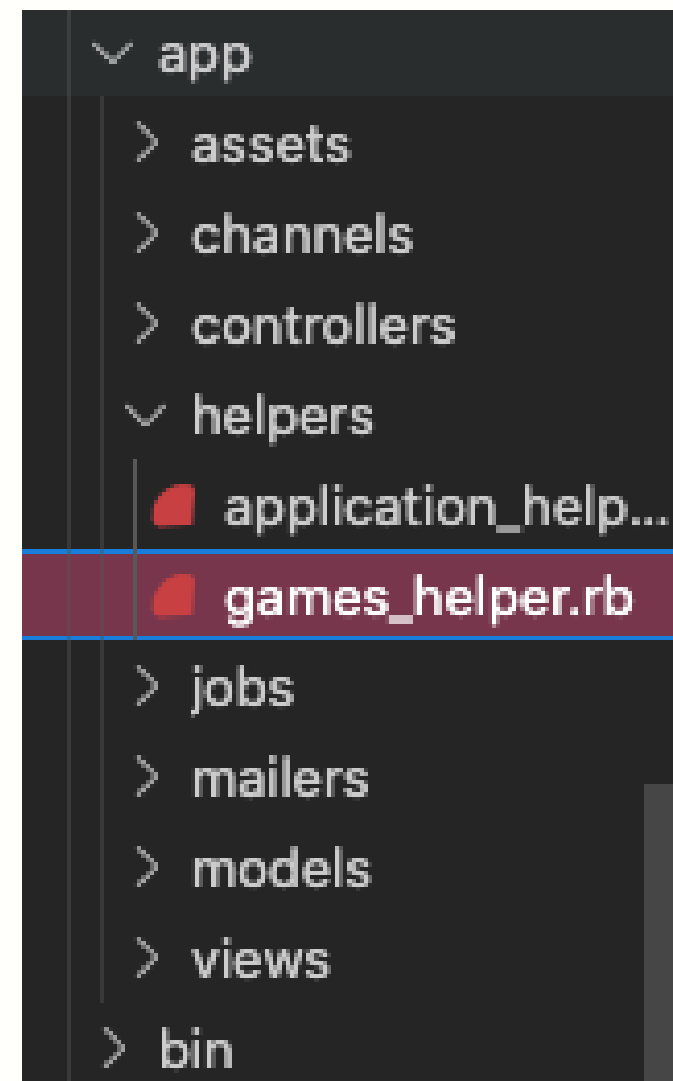
Se crea el siguiente archivo



- Archivo html
- Se creará una para cada método declarado
- ¿Cómo usar ruby en html?
<% {@variable}%> o
<%= {variable}%>



Se crea el siguiente archivo



```
1  module GamesHelper
2
3  end
```

- Opcional.
- Sirve para crear funciones que pueden ser luego llamadas en el controlador.




Scaffold

```
rails g scaffold {Nombre_modelo} {atributo1}:{tipo_atributo1} {atributo2}:{tipo_atributo2} ...
```

Ventajas

- Crea el modelo con su migración, la vista y el controlador con solo un comando.

Desventajas

- Puede generar archivos y funciones que nunca utilizemos.
 - No permite entender bien cómo funciona MVC ya que hace todo el trabajo.
- 



RUTAS

- Se crean automáticamente al crear el controlador
- Conectan las distintas vistas





Comando rails routes

	Prefix	Verb	URI Pattern	Controller#Action
	root	GET	/	games#index
	games_new	GET	/games/new(:format)	games#new
	games	POST	/games(:format)	games#create
	games_index	GET	/games/index(:format)	games#index
	games_show	GET	/games/show(:format)	games#show
	games_edit	GET	/games/edit(:format)	games#edit
	games_update	PATCH	/games/update(:format)	games#update
	games_delete	DELETE	/games/delete(:format)	games#delete
	rails_service_blob	GET	/rails/active_storage/blobs/:signed_id/*filename(:format)	active_storage/blobs#show
	rails_blob_representation	GET	/rails/active_storage/representations/:signed_blob_id/:variation_key/*filename(:format)	active_storage/representations#show
	rails_disk_service	GET	/rails/active_storage/disk/:encoded_key/*filename(:format)	active_storage/disk#show
	update_rails_disk_service	PUT	/rails/active_storage/disk/:encoded_token(:format)	active_storage/disk#update
	rails_direct_uploads	POST	/rails/active_storage/direct_uploads(:format)	active_storage/direct_uploads#create

- Muestra las rutas existentes





Rails Routes prefix



```
Prefix Verb
root GET
games_new GET
games POST
games_index GET
games_show GET
games_edit GET
games_update PATCH
games_delete DELETE
rails_service_blob GET
rails_blob_representation GET
rails_disk_service GET
update_rails_disk_service PUT
rails_direct_uploads POST
```



Como se usan

- Ruta sin parametros

```
<%= link_to 'Titulo boton', {helper_indicado_en_routes}_path %>
```

- Ruta con parametros

```
<%= link_to 'Titulo boton', {helper_indicado_en_routes}_path(:llave => valor) %>
```

