

RAILS: ROUTING Y CRUD

IIC2143 2022-2

¿Qué veremos?



Ruby on Rails



Routing

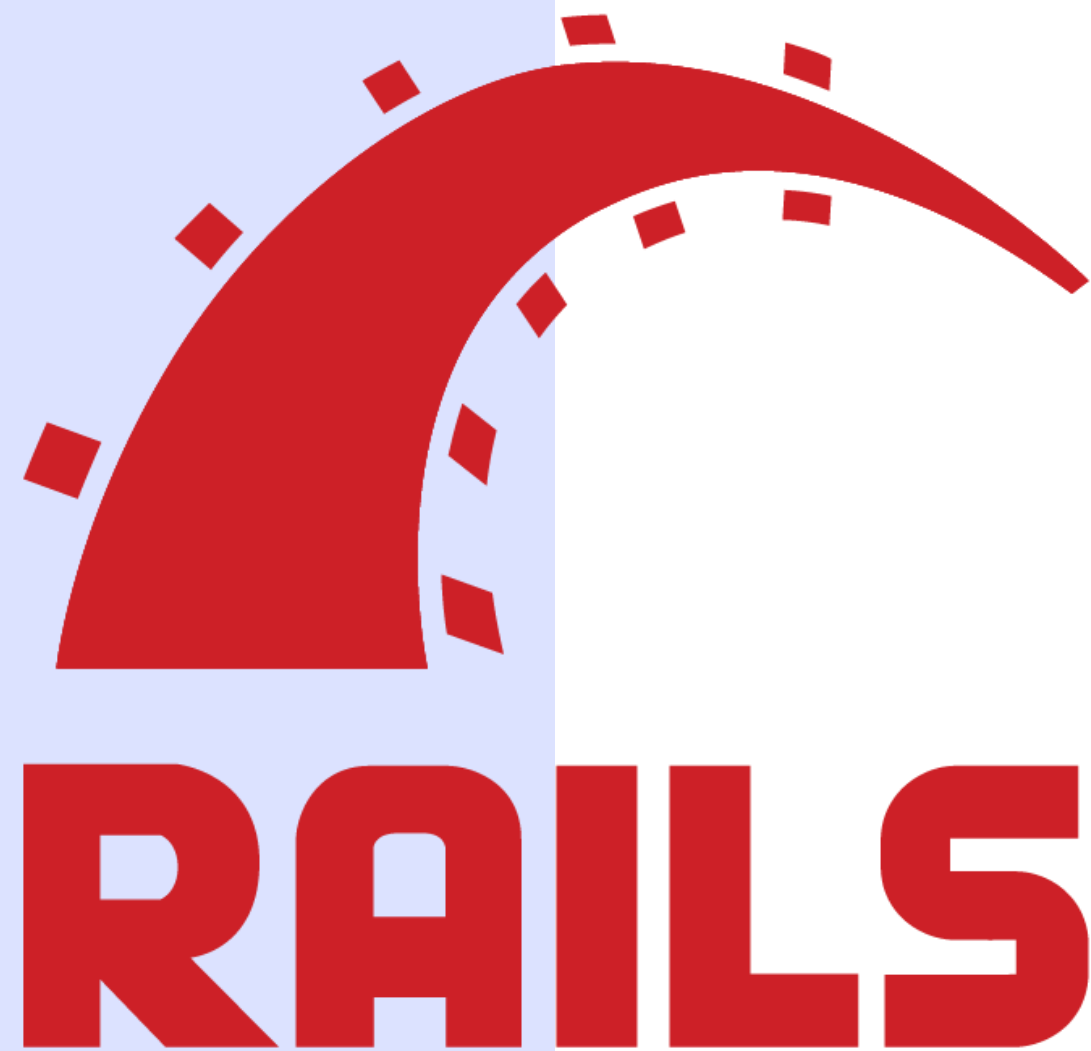


CRUD



Ruby on Rails





- Es un framework de código abierto escrito en el lenguaje de programación Ruby.
 - Nos permite **desarrollar aplicaciones web** con bastante rapidez y facilidad, trabajando tanto en el *back-end* como, como en el *front-end*.
 - Se estructura mediante el **patrón Modelo Vista Controlador**.
-

Modelos

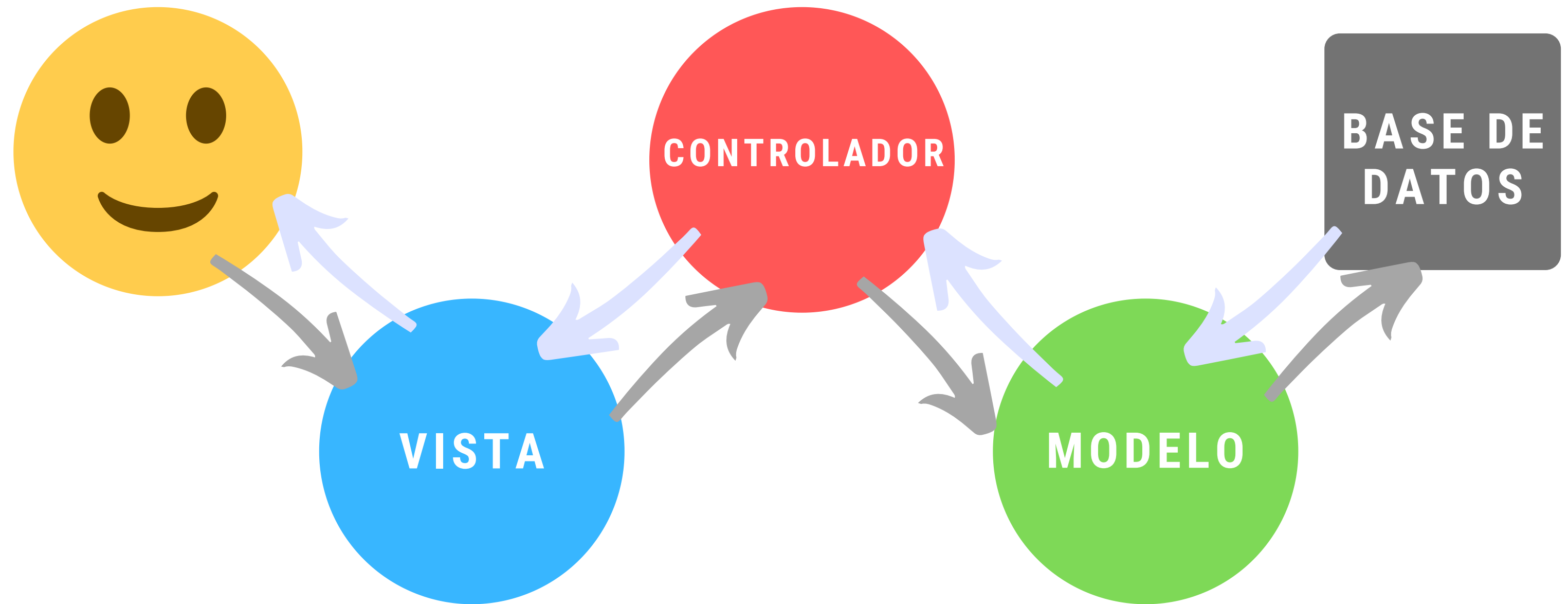
- Son las **entidades** de mi aplicación.
 - Interactúa con la base de datos.
 - Se comunica con el controlador.
-

Vistas

- Es con lo que el **usuario interactúa**.
 - Se comunica con el controlador, quién le puede entregar datos para que el usuario visualice.
 - Utiliza **motores de plantilla (ERB)**.
-

Controladores

- **Procesa acciones.**
 - Recibe datos de parte del modelo.
 - Le entrega datos a las vista.
-





Routing



Rails Router

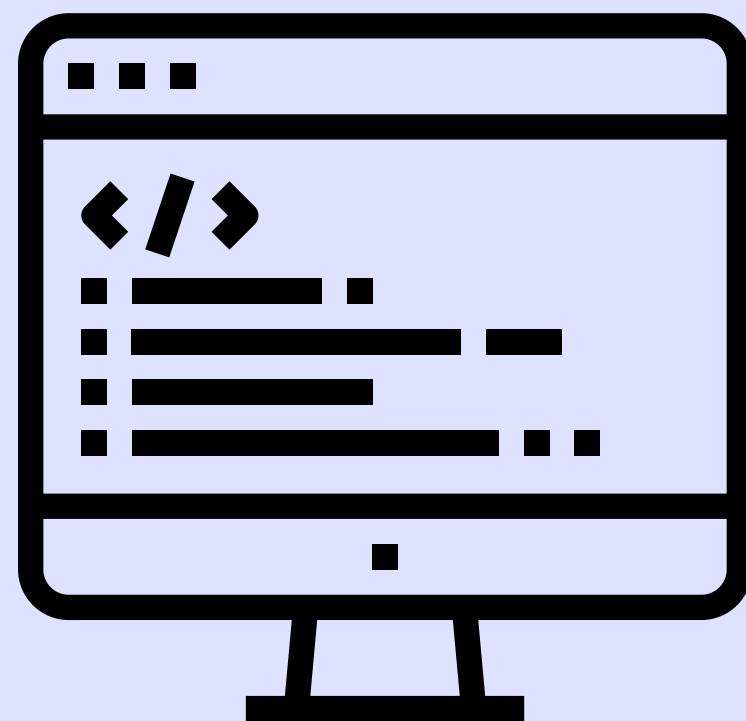
- Reconoce URL's y los envía a un método de un controlador.





```
# /config/routes.tb
```

```
Rails.application.routes.draw do  
  método_http 'url', to: 'controlador#método'  
end
```



Ejemplo



CRUD



CRUD

- **Create, Read, Update, Delete** (crear, leer, actualizar, eliminar).
 - Cuatro funciones básicas llevadas a cabo en una base de datos.
-

CREATE

- **new:** renderiza un form de HTML para instanciar un modelo.
 - **create:** corre al hacer "submit" del form del método new. Hace el cambio en la base de datos.
-

READ

- **index:** muestra una lista de todas las instancias de un modelo.
 - **show:** muestra información de una instancia específica
-

UPDATE

- **edit:** renderiza un form de HTML editar la instancia de un modelo.
 - **update:** corre al hacer "submit" del form del método edit. Hace el cambio en la base de datos.
-

DELETE

- **destroy:** elimina la instancia de un modelo.
-

HTTP Verb	Path	Controller#Action	Used for
GET	/photos	photos#index	display a list of all photos
GET	/photos/new	photos#new	return an HTML form for creating a new photo
POST	/photos	photos#create	create a new photo
GET	/photos/:id	photos#show	display a specific photo
GET	/photos/:id/edit	photos#edit	return an HTML form for editing a photo
PATCH/PUT	/photos/:id	photos#update	update a specific photo
DELETE	/photos/:id	photos#destroy	delete a specific photo



```
# /config/routes.tb
```

```
Rails.application.routes.draw do  
  get '/photos', to: 'photos#index'  
  get '/photos/:new', to: 'photos#new'  
  post '/photos', to: 'photos#create'  
  get '/photos/:id', to: 'photos#show'  
  get '/photos/:id/edit', to: 'photos#edit'  
  patch '/photos/:id', to: 'photos#update'  
  delete '/photos/:id', to: 'photos#destroy'  
end
```



```
# /config/routes.tb
```

```
Rails.application.routes.draw do  
  resources :photos  
end
```

PATH HELPERS

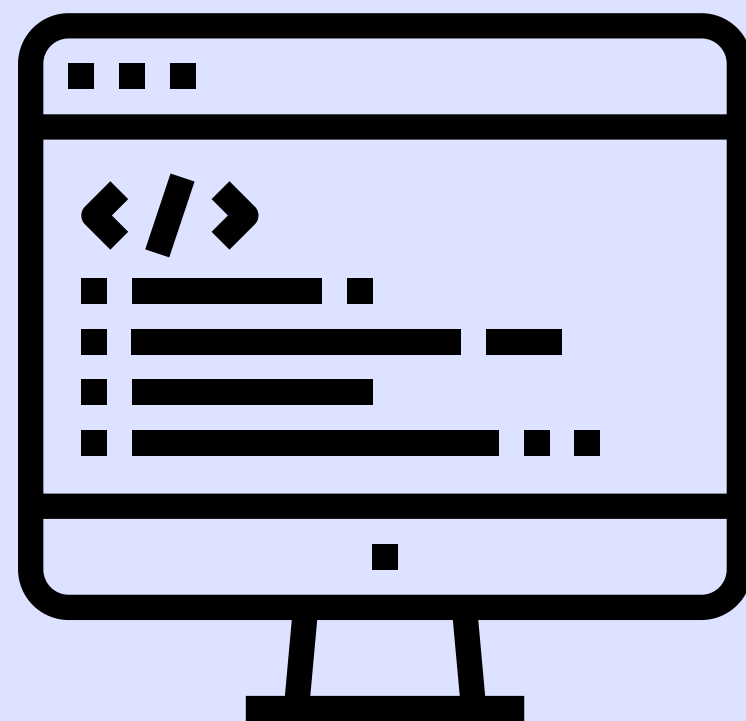
- **photos_path** → /photos
 - **new_photo_path** → /photos/new
 - **edit_photo_path(:id)** → /photos/:id/edit
 - **photo_path(:id)** → /photos/:id
-



```
<%= link_to 'View photo', '/photos/' + photo.id.to_s %>
```



```
<%= link_to 'View photo', photo_path(photo) %>
```



Ejemplo

rails g model game name:string year:integer

Crea el modelo Game con los atributos name y year, junto con la migración correspondiente

rails g controller games new index show edit

Crea el controlador Games junto con los métodos y views correspondientes

rails g scaffold game name:string year:integer

Crea tanto el modelo como el controlador, además de generar el código de cada método y el contenido de las views

rails db:migrate

Corre las migraciones faltantes
