



# Primer proyecto en Rails

Introducción a Rails (Parte II)

***Aplicar procedimiento de creación de un proyecto nuevo en Ruby on Rails que incorpora las vistas y controladores utilizando la línea de comando, helpers y layouts para dar solución a un requerimiento.***

- Unidad 1:  
Primer proyecto en Rails
- Unidad 2:  
Acelerar la construcción de un proyecto
- Unidad 3:  
Almacenar información en una base de datos
- Unidad 4:  
Despliegue de una aplicación en internet utilizando Heroku.
- Unidad 5:  
Disponibilizar un servicio web utilizando la arquitectura REST



Te encuentras aquí



## ¿Qué aprenderás en esta sesión?

- *Aplica procedimiento de creación de controllers y vistas utilizando generadores en Ruby on Rails para dar solución a un requerimiento.*

¿Cuáles son las  
tecnologías mínimas  
que domina un  
desarrollador Full Stack?



**/\* Estructura de los directorios de  
un proyecto nuevo en Rails \*/**

# Demostración "Portal de noticias en Rails"



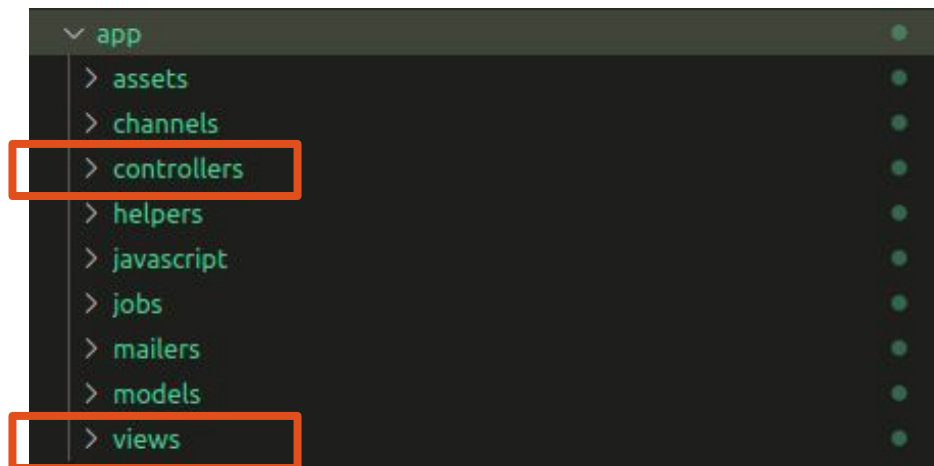
# Sigamos los pasos

A continuación, realizaremos un portal de noticias en **Ruby on Rails** donde agregaremos una vista para noticias de tecnología y otra para deportes. Para lograrlo, utilizaremos **generadores** de Rails.

- **Paso 1:** Crear un nuevo proyecto con el nombre portalNoticias:  
`rails new portalNoticias.`
- **Paso 2:** Inspeccionar la estructura de directorios y archivos que se generan al crear el proyecto.

# directorio /app

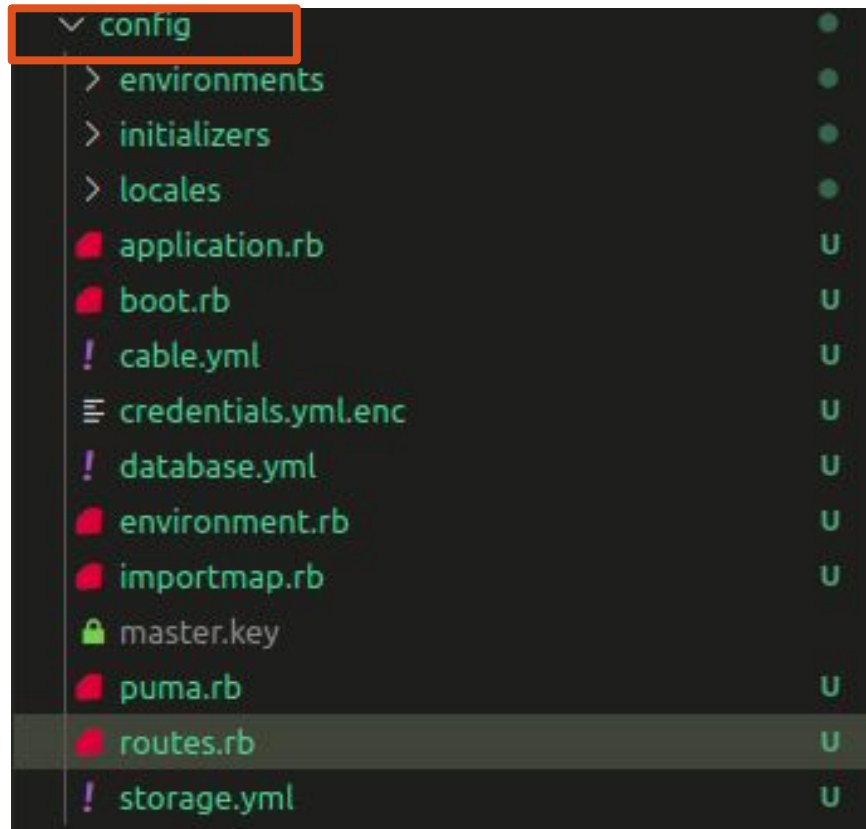
- Es uno de los directorios más importantes del proyecto.
- Es el directorio donde más trabajaremos al construir nuestras aplicaciones.
- Nos centraremos más adelante en los **subdirectorios** /controllers y /views.





# directorio /config

- Es un directorio que contiene configuraciones bases para que la aplicación funcione.
- En el modificaremos más adelante el archivo **routes.rb** para definir las rutas de nuestro blog.



# El gemfile

- Es un archivo fundamental en nuestra aplicación.
- Contiene el listado de dependencias iniciales para que la aplicación funcione.
- Nos muestra la versión de Ruby que posee nuestro proyecto y la de Rails.



***/\* Controladores, vistas y rutas \*/***

# Demostración

## "Agregar rutas, controladores y vistas a nuestro blog"



# Sigamos los pasos

- **Paso 3:** Dirigirnos al directorio `/config/routes.rb` para generar la ruta de news.  
`get '/technology'`.
- **Paso 4:** Acceder a la ruta `localhost:3000/technology` en el navegador.
- **Paso 5:** Analizar el mensaje de retorno de rails al acceder `localhost:3000/technology`



# Entendamos el código

- En el archivo **routes.rb** del directorio **config** agregamos las rutas que tendrá nuestra aplicación.
- **get '/technology**, le decimos que queremos obtener información a partir del verbo GET.
- Rails nos indica que está esperando un **controlador** para acceder a esta ruta.

## ArgumentError

Missing :controller key on routes definition, please check your routes.

Extracted source (around line #7):

```
5   # root "articles#index"
6
7   get "/technology"
8 end
```

# Controladores

- Para crear páginas en Rails tenemos que hacerlo bajo un controlador.
- Una simplificación del concepto es que su función es agrupar páginas.
- Estas páginas se conocen como vistas.
- Las vistas son las encargadas de desplegar la información que se mostrará en el cliente (navegador).

## Ejemplo de agrupación

- ❖ localhost:3000/news/tecnología
- ❖ localhost:3000/news/libros
- ❖ localhost:3000/news/moda
- ❖ localhost:3000/news/salud

# Sigamos los pasos

- **Paso 6:** En el directorio `/app/controllers` crear el archivo `news_controller.rb`.
- **Paso 7:** En el archivo del controlador escribir la clase `NewsController` que hereda de `ApplicationController`.
- **Paso 8:** Dentro de la clase definir un método `index`.
- **Paso 9:** Agregar en el `routes.rb` el controlador y el método que se encargará de mostrar la página de technology.  
`get "/technology", to: 'news#index'`





# Entendamos el código

- El archivo `news_controller.rb` contiene una clase llamada `NewsController`.
- Se utiliza la sintaxis de `PascalCase`.
- Hereda de la clase `ApplicationController`.
- Contiene un método `index`.

```
class NewsController < ApplicationController
  def index
  end
end
```

# Entendamos el código

- El archivo de **routes.rb** contiene la ruta que va a mostrar la información en el navegador.
- **to: 'news#index'**, le dice a rails que busque el controlador **news** y el método **index** que esté definido desde de ese controlador.

```
Rails.application.routes.draw do
  get "/technology", to: 'news#index'
end
```

# Entendamos el código

- Al recargar el navegador se muestra el siguiente mensaje de Rails.
- Está esperando un template en formato HTML que sea capaz de mostrar la información.
- No tenemos una vista declarada para retornar algo.

**No template for interactive request**

**NewsController#index is missing a template for request formats: text/html**

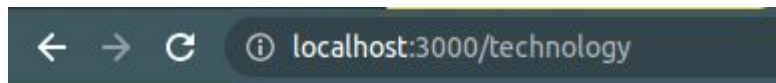
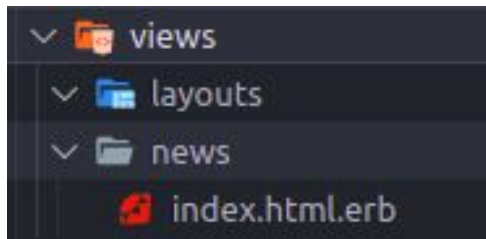
# Sigamos los pasos

- **Paso 10:** A continuación, buscaremos el directorio `/app/views`.
- **Paso 11:** Dentro de la carpeta de `/views` creamos un directorio nuevo llamado `/news`.
- **Paso 12:** En `/news` creamos un archivo llamado `index.html.erb`



# Entendamos el código

- En `/views` creamos el directorio de `/news`. Ese nombre debe tener similitud con el controlador generado.
- En `/news`, creamos una vista `index.html.erb` que corresponde al nombre del método creado en nuestro controlador.



## Noticias de tecnología

Resultado al agregar la vista  
index.html.erb

**`/* MVC */`**

# Modelo - Vista - Controlador

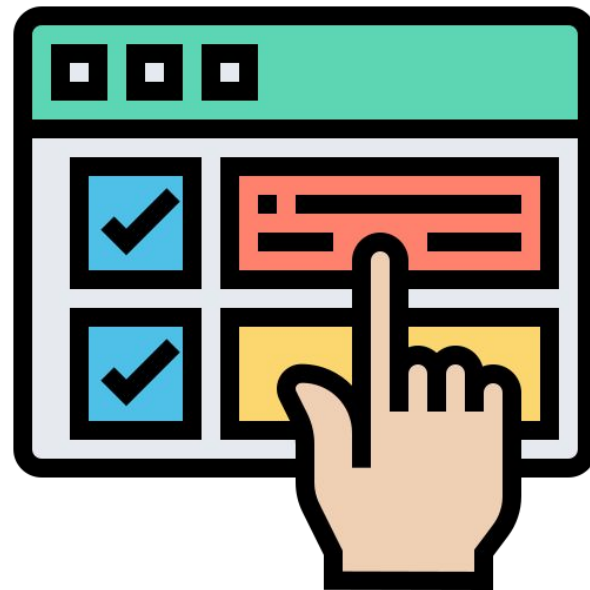
El MVC (Modelo Vista Controlador) es un patrón inspirado en la creación de sistemas que contienen:

1. Interfaz de usuario (**Vista**)
2. Datos (**Modelo**)
3. Lógica (**Controladores**)

Ruby On Rails es un Framework MVC por defecto y hasta este punto solo hemos trabajado la **Vista** y el **Controlador**. Más adelante veremos qué son los modelos.

# La vista

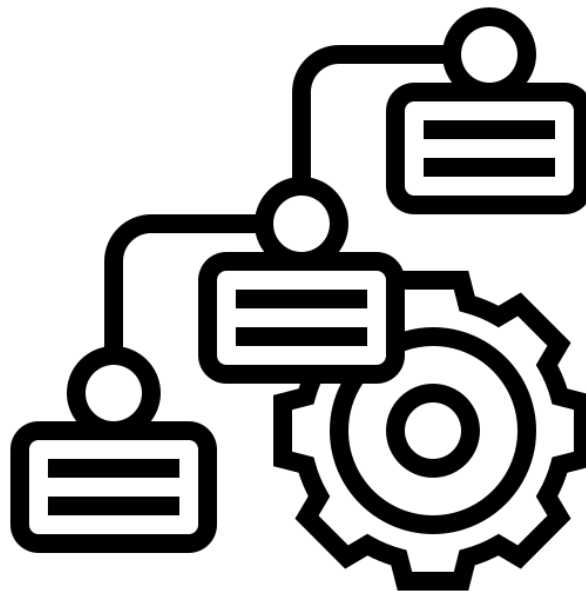
- Es donde se establece el cómo se mostrarán los datos.
- Principalmente se utiliza HTML para la muestra de datos en aplicaciones web.
- Es la información que verá el usuario en su navegador.





# El controlador

- Contiene la lógica que envía respuestas a partir de la interacción del usuario con el sistema.
- También posee la lógica capaz de modificar nuestra base de datos a partir de esa interacción. **(Esto lo veremos en siguientes apartados)**



# Ejercicio

## "Agrega la ruta de deportes al blog"



# Contexto

A continuación, deberás continuar con el blog y en esta ocasión agregar la ruta para noticias deportivas.

1. Puedes utilizar el mismo controlador y generar un nuevo método.
2. Modifica las rutas e incorpora la encargada de mostrar las noticias de deportes. *Recuerda declarar el controlador que manejará el flujo y su método.*
3. Crea la vista en HTML. *Recuerda que la vista debe coincidir con el nombre del método para los deportes.*



¿Qué te pareció lo aprendido hasta el momento?





## Próxima sesión...

- *Agrega nuevas páginas y rutas a un proyecto Ruby on Rails utilizando procedimiento manual para dar solución a un requerimiento.*
- *Construye vistas agregando código Ruby (ERB) dentro de las vistas para dar solución a un requerimiento.*

**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

