

SEGUNDA PARTE – INICIO DO DESENVOLVIMENTO

A construção de uma aplicação em Rails segue a estrutura MVC (Model-View-Controller), e o fluxo ideal geralmente começa assim:

1. **Definição das rotas (routes.rb)** → Para mapear as URLs.
2. **Criação do modelo (Model)** → Define a estrutura do banco de dados e a lógica de negócios.
3. **Criação do controlador (Controller)** → Faz a ponte entre o modelo e a view.
4. **Criação das views (View)** → Para exibir os dados no navegador.

Porém, dependendo do seu estilo de desenvolvimento, você pode começar pelas views se já tiver uma ideia de como quer a interface. Mas, se começar pelas views sem os modelos e controladores prontos, você terá que lidar com muitas referências quebradas.

O caminho mais comum e seguro é:

1. Criar o **modelo** (Model) e a **migração** para estruturar o banco de dados.
2. Criar o **controlador** (Controller) e definir as ações principais (index, show, new, create, edit, update, destroy).
3. Criar as **rotas** (Routes) para direcionar as requisições.
4. Criar as **views** para exibir as informações.

VAMOS MONTAR UM INDEX SIMPLES -VISÃO DIDÁTICA

♦ Entendendo a Definição de Rotas no Rails

No Ruby on Rails, o arquivo **config/routes.rb** é responsável por definir as rotas da aplicação, ou seja, quais URLs estarão disponíveis e para onde elas devem direcionar o usuário.

📌 Explicando a Rota Criada

```
blog > config > routes.rb
1  Rails.application.routes.draw do
2
3    get '/', controller: 'home', action: 'index'
4
5  end
6
```

♦ **Rails.application.routes.draw do ... end**

Essa estrutura define todas as rotas da aplicação. Tudo que estiver dentro desse bloco será considerado como parte do sistema de rotas do Rails.

♦ `get '/'`


Aqui estamos dizendo ao Rails que queremos definir uma rota do tipo **GET**, que é usada para exibir páginas (como a página inicial de um site).

♦ `controller: 'home', action: 'index'`

Isso indica que quando um usuário acessar a **URL raiz** (`/`), o Rails deve direcioná-lo para o **controller** chamado **HomeController**, executando a **ação (método) index** dentro desse controller.

Como Essa Rota Funciona?

1. O usuário acessa o navegador e digita `http://localhost:3000/`.
2. O Rails verifica o arquivo `routes.rb` e encontra a rota `get '/'`.
3. O Rails chama o **controller HomeController** e executa o método **index**.
4. O método **index** dentro do controller renderiza uma **view**, que será mostrada no navegador.

 **Resumo:** Essa rota faz com que o Rails exiba uma página inicial (`/`), carregando o método **index** do **HomeController**.

♦ Entendendo o Controller no Rails

No Rails, o **controller** é responsável por receber as requisições do usuário, processar a lógica de negócios e decidir qual **view** deve ser renderizada como resposta. Ele também pode interagir com o **model** para acessar ou manipular dados.

Explicando o HomeController

```
class HomeController < ApplicationController
  def index
    @nome = "USER MASTER"
    # @nomes = ["Alice", "Bob", "Carlos", "Diana"]

    @devs = [
      { nome: 'Weslley', dev: 'Ruby on Rails' },
      { nome: 'André', dev: 'Next.js' },
      { nome: 'Kaike', dev: 'Mysql' }
    ]
  end
end
```

♦ `class HomeController < ApplicationController`

Aqui estamos criando o controller chamado **HomeController**, que herda de **ApplicationController**. Isso significa que ele tem acesso a todos os métodos e funcionalidades básicas definidas em **ApplicationController**. O nome do controller deve seguir a convenção **plural** da URL, mas como estamos tratando da página inicial, usamos **HomeController**.

♦ **def index**

O método `index` é o responsável por exibir a página inicial, que foi definida na rota. Quando o usuário acessar a URL raiz (`/`), esse método será chamado. Dentro desse método, você pode processar qualquer lógica que precise para essa página.

♦ **Variáveis de Instância**

- **@nome = "USER MASTER"**

Aqui estamos criando uma variável de instância chamada `@nome`, que será disponibilizada para a **view** correspondente ao método `index`. Esse valor pode ser usado na página para personalizar o conteúdo que será exibido.

- **@devs**

Esse é um array de hashes que contém informações sobre desenvolvedores. Cada hash tem dois pares de chave-valor: o nome do desenvolvedor e a tecnologia que ele usa. Como a variável é de instância (`@`), ela também estará disponível na view.

♦ **O que Acontece Aqui?**

Quando o usuário acessar a URL raiz (`/`), o Rails:

1. Executa o método `index` do `HomeController`.
2. Define as variáveis `@nome` e `@devs`.
3. Essas variáveis ficam disponíveis para a view que será renderizada, permitindo exibir os dados na interface do usuário.

♦ **Entendendo a View no Rails**

No Rails, a **view** é o arquivo responsável por renderizar o conteúdo que será exibido ao usuário no navegador. Ela usa **ERB (Embedded Ruby)** para embutir código Ruby no HTML, permitindo exibir variáveis, executar loops, condições e mais.

📌 **Explicando a View `index.html.erb`**

```
<header>
  <h1>Bem-vindo à Página Inicial, <%= @nome %></h1>
</header>
<main>
  <h2>Lista dos desenvolvedores</h2>
  <%= @devs.each do |dev| %>
    <p>
      <%= dev[:nome] %> - <%= dev[:dev] %>
    </p>
  <%= end %>
</main>
```

♦ **<header>**

Aqui estamos criando um **cabeçalho** para a página. Dentro dele, exibimos um título que diz "**Bem-**

vindo à Página Inicial" e o nome do usuário, que vem da variável de instância `@nome` definida no controller.

- ♦ `<%= @nome %>`

A sintaxe `<%= %>` é usada para **inserir valores de Ruby no HTML**. Nesse caso, o valor de `@nome` (que foi definido como "USER MASTER" no controller) será mostrado dentro da tag `<h1>`, então o título da página vai ser: **"Bem-vindo à Página Inicial, USER MASTER"**.

- ♦ `<main>`

Essa seção contém o conteúdo principal da página, onde estamos listando os desenvolvedores.

- ♦ `<% @devs.each do |dev| %>`

Aqui estamos iniciando um **loop** (each) para iterar sobre o array `@devs` que foi definido no controller. O código entre `<% %>` é Ruby puro, ou seja, o código será executado, mas não será exibido na página.

- ♦ `<%= dev[:nome] %>` - `<%= dev[:dev] %>`

Dentro do loop, estamos acessando cada elemento do array `@devs`. Cada item é um **hash** com chaves `:nome` e `:dev`.

- `<%= dev[:nome] %>` vai exibir o nome do desenvolvedor (ex: "Wesley").
- `<%= dev[:dev] %>` vai exibir a tecnologia que o desenvolvedor usa (ex: "Ruby on Rails").

- ♦ `<% end %>`

Essa linha finaliza o loop. Ou seja, ele vai repetir o código dentro do bloco para cada item na lista `@devs`. No final, todos os desenvolvedores serão listados na página.



Como tudo funciona junto?

1. Quando o usuário acessa a URL / (definida na rota), o **Rails** executa o método `index` no **controller** `HomeController`.
2. O **controller** define as variáveis `@nome` e `@devs`, que são disponibilizadas para a **view**.
3. A **view** usa **ERB** para renderizar essas variáveis em HTML:
 - O nome do usuário é exibido no título da página.
 - A lista de desenvolvedores é exibida em formato de texto, mostrando o nome e a tecnologia de cada um.



Resumo: A rota chama o controller, que prepara os dados, e a view exibe esses dados de forma dinâmica, permitindo que a página inicial mostre um conteúdo personalizado com a lista de desenvolvedores.