

# Ruby On Rails: Laboratório 05

Laboratório Engenharia de Software



**PUC Minas**  
**Poços de Caldas**

Luiz Alberto Ferreira Gomes

Curso de Ciência da Computação

30 de abril de 2020

# Agenda

---

1 Páginas Estáticas

2 Estruturando a Aplicação

3 Para Saber Mais

# Controladores e Ações (1)

---

- Um controlador Rails pode ser gerado utilizando o comando `rails generate controller` ou `rails g controller`. Além dele, esse comando permite a geração de algumas ações.

---

```
1 ~/blog$ rails generate controller Home index help
2 route get 'home/help'
3 route get 'home/index'
4 invoke erb
5 create app/views/home
6 create app/views/home/index.html.erb
7 create app/views/home/help.html.erb
8 ...
```

---

# Arquivo de Roteamento: routes.rb (1)

---

- O conteúdo atualizado do arquivo **routes.rb** deverá conter as notas rotas para as ações recém-criadas.

Listing 1: config/routes.rb

```
1 Rails.application.routes.draw do
2   get 'home/index'
3   get 'home/help'
4   root 'home#index'
5   resources :posts
6 end
```

- **GET** é uma das operações do HTTP mais comuns, ela é utilizada para a leitura de dados na web.

## Arquivo de Roteamento: routes.rb (2)

---

- o navegador envia uma requisição **GET** toda vez de se deseja visitar um endereço, com por exemplo `<http://www.inf.pucpcaldas.br>`.
- As páginas vinculadas às ações **index** e **help** podem ser visitadas, primeiro, iniciando o servidor Rails com os comandos **rails server** ou, simplesmente, **rails s**.

---

```
1 $ cd ~/workspace/blog
2 $ rails server
```

---

- E depois, digitando as páginas os endereços `<http://localhost:3000/home/index>` e, depois, `<http://localhost:3000/home/help>`.

## Arquivo de Roteamento: routes.rb (3)

- Dica: Introdução às Rotas do Rails  
<<https://guides.rubyonrails.org/v3.2/routing.html>>

# Página Home (1)

---

- A páginas geradas pelo comando rails generate devem ser personalizadas para atenderem às especificações do projeto da aplicação. Por exemplo, para a página associada à ação principal pode-se reescrevê-la da seguinte forma:

Listing 2: app/views/home/index.html.erb

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Index | Blog App</title>
5   </head>
6   <body>
7     <h1>Blog App</h1>
8     <p>
9       Esta e a pagina index da aplicacao Blog desenvolvida durante
10      o Laboratorio de Engenharia de Software.
11     </p>
12   </body>
13 </html>
```

## Página Home (2)

---

- Já para a página associada à ação ajuda pode-se reescrevê-la, por exemplo, da seguinte forma:

Listing 3: app/views/home/help.html.erb

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Help | Blog App</title>
5   </head>
6   <body>
7     <h1>Blog App</h1>
8     <p>
9       Esta e a pagina help da aplicacao Blog desenvolvida durante
10      o Laboratorio de Engenharia de Software.
11     </p>
12   </body>
13 </html>
```



# Agenda

---

1 Páginas Estáticas

2 Estruturando a Aplicação

3 Para Saber Mais

# Criando Um Helper

---

- Um helper será criado para promover o reuso e evitar a repetição de código em uma aplicação.

Listing 4: app/helpers/application\_helper.rb

```
1 module ApplicationHelper
2   def preenche_titulo(titulo_da_pagina='')
3     titulo_padrao="Blog App"
4     if (titulo_da_pagina.empty?)
5       titulo_padrao
6     else
7       "#{titulo_da_pagina} | #{titulo_padrao}"
8     end
9   end
10 end
```

# Estruturando a Aplicação

---

- Uma aplicação web deve ser estruturada de forma a facilitar o seu uso. Uma ferramentas utilizadas para isso é o protótipo essencial. A ideia inicial para a aplicação Blog foi pensada.



# Navegação da Aplicação (1)

---

## Listing 5: app/views/layouts/application.html.erb

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title><%= preenche_titulo(yield(:titulo)) %></title>
5   <%= stylesheet_link_tag 'application', media: 'all',
6     'data-turbolinks-track' => true %>
7   <%= javascript_include_tag 'application', 'data-turbolinks-track' => true %>
8   <%= csrf_meta_tags %>
9 </head>
10 <body>
11   <%= link_to "Blog", "#", id: "logo" %>
12   <p>
13     <ul>
14       <li><%= link_to "Index", home_index_path %></li>
15       <li><%= link_to "Help", home_help_path %></li>
16     </ul>
17   </p>
18   <%= yield %>
19 </body>
20 </html>
```

# Refatorando as Páginas Index e Help (1)

---

## Listing 6: app/views/home/index.html.erb

```
1 <%= provide :titulo, "Index" %>
2 <div>
3   <h1>Blog App</h1>
4   <p>
5     Esta e a pagina principal da aplicacao Blog desenvolvida durante
6     o Laboratorio de Engenharia de Software.
7   </p>
8   <%= link_to "Posts", posts_path %>
9 </div>
```

# Refatorando as Páginas Index e Help (2)

---

## Listing 7: app/views/home/help.html.erb

```
1 <%= provide :titulo, "Help" %>
2 <div>
3   <h1>Blog App</h1>
4   <p>
5     Esta e a pagina de ajuda da aplicacao Blog desenvolvida durante
6     o Laboratorio de Engenharia de Software.
7   </p>
8 </div>
```

# Hora de Colocar a Mão na Massa (1)

---

1. Implemente os recursos apresentados na aula de hoje.
2. Registre as mudanças realizadas no repositório local:

```
$ git add .
```

3. Efetive as mudanças realizadas no repositório local:

```
$ git commit -m "#1: manutencao de posts"
```

4. Registre as alterações no repositório remoto:

```
$ git push -u origin master
```

# Agenda

---

1 Páginas Estáticas

2 Estruturando a Aplicação

**3 Para Saber Mais**



# Para Saber Mais

---

- `<https://www.ruby-lang.org/en/>`
  - referência oficial da linguagem Ruby onde a toda a sua documentação está disponível para ser consultada.
- `<http://rubyonrails.org/>`
  - referência oficial do framework Rails onde a toda a sua documentação está disponível para ser consultada.
- `<http://www.codecademy.com/pt/tracks/ruby>`
  - curso iterativo em português sobre a linguagem Ruby.
- `<https://gorails.com/setup/ubuntu/16.04>`
  - guia para instalação do Ruby on Rails no Ubuntu e no Mac OSX.