

Aula 03: Ruby On Rails

Laboratório de Projeto de Sistemas



PUC Minas
Poços de Caldas

Luiz Alberto Ferreira Gomes

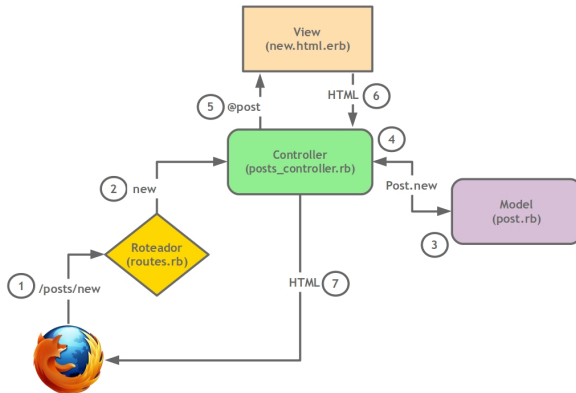
Curso de Ciência da Computação

30 de outubro de 2020

Agenda

- 1 Nova Postagem
- 2 Exibir Postagem
- 3 Atualizar Postagem
- 4 Reuso com Partial
- 5 Visão Destroy
- 6 Controle de Versão
- 7 Para Saber Mais

Ação: New (1)



Ação: New (2)

- Um novo objeto `@post` da classe `Post` é instanciado
- Procura pela visão `new.html.erb` para renderizar a resposta

Listing 1: `app/controllers/posts_controller.rb`

```
1 Class PostsController < ApplicationController
2   def new
3     @post = Post.new
4   end
5 end
```

Visão: New (1)

- Reinicie o servidor web e acesse a url `<http://localhost:3000/posts/new>`. Veja o erro que ocorreu.
- Implemente a visão `new.html.erb`:

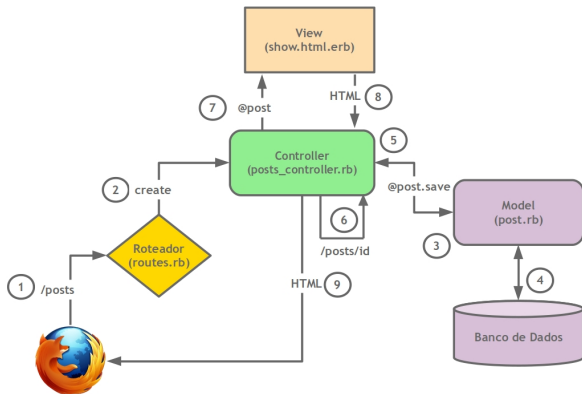
Listing 2: views/posts/new.html.erb

```
1 <h1>Novo Post</h1>
2 <%= form_with model: @post, local: true do |form| %>
3   <p>
4     <%= form.label :title %><br>
5     <%= form.text_field :title %>
6   </p>
7
8   <p>
9     <%= form.label :body %><br>
10    <%= form.text_area :body %>
11  </p>
```

Visão: New (2)

```
12
13     <p>
14         <%= form.submit %>
15     </p>
16 <% end %>
```

Ação: Create (1)



Ação: Create (2)

- Um novo objeto `@post` da classe `Post` é criado com os parâmetros que foram passados pelo formulário `new`
- Tenta `salvar` o objeto `@post` no `banco de dados`

Listing 3: controllers/posts_controller.rb

```
1 class PostsController < ApplicationController
2   def new
3     @post = Post.new
4   end
5
6   def create
7     @post = Post.new(post_params)
8
9     @post.save
10    redirect_to @post
11  end
12
```


Ação: Create (3)

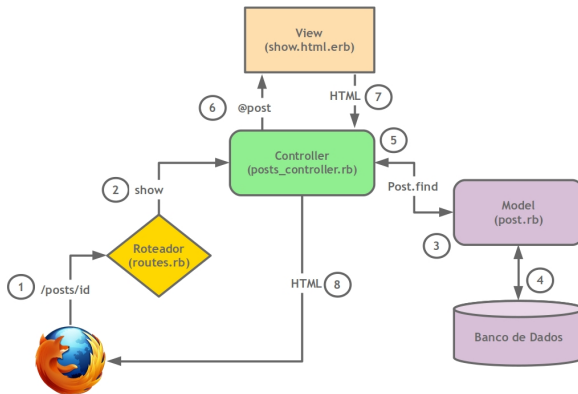
```
13 private
14   def post_params
15     params.require(:post).permit(:title, :body)
16   end
17 end
```

- a linha 15 implementa **strong parameters** para aumentar a segurança da aplicação
- Como a ação **Show** ainda não foi implementada, ocorrerá uma erro quando o botão **Submit** for pressionado.

Agenda

- 1 Nova Postagem
- 2 Exibir Postagem
- 3 Atualizar Postagem
- 4 Reuso com Partial
- 5 Visão Destroy
- 6 Controle de Versão
- 7 Para Saber Mais

Ação: Show (1)



Ação: Show (2)

- Recupera **uma** postagem específica no parâmetro **id** passado como parte da URL
- (Implicitamente) procura pelo **show.html.erb** para renderizar a resposta

Listing 4: controllers/posts_controller.rb

```
1 class PostsController < ApplicationController
2   def show
3     @post = Post.find(params[:id])
4   end
5
6   def new
7     @post = Post.new
8   end
9
10  def create
```

Ação: Show (3)

```
11     @post = Post.new(post_params)
12
13     @post.save
14     redirect_to @post
15 end
16 private
17 def post_params
18     params.require(:post).permit(:title, :body)
19 end
20 end
```

Visão: Show (1)

- Implemente a visão `show.html.erb`:

Listing 5: views/posts/show.html.erb

```
1 <p>
2   <strong>Title:</strong>
3   <%= @post.title %>
4 </p>
5 <p>
6   <strong>Body:</strong>
7   <%= @post.body %>
8 </p>
```

Agenda

- 1 Nova Postagem
- 2 Exibir Postagem
- 3 Atualizar Postagem**
- 4 Reuso com Partial
- 5 Visão Destroy
- 6 Controle de Versão
- 7 Para Saber Mais

Ação: Edit

- Recupera uma postagem específica no parâmetro **id** passado como parte da URL
- (Implicitamente) procura pelo **edit.html.erb** para renderizar a resposta

Listing 6: controllers/posts_controller.rb

```
1 Class PostsController < ApplicationController
2   def edit
3     @post = Post.find(params[:id])
4   end
```


Agenda

- 1 Nova Postagem
- 2 Exibir Postagem
- 3 Atualizar Postagem
- 4 Reuso com Partial**
- 5 Visão Destroy
- 6 Controle de Versão
- 7 Para Saber Mais

Partials (1)

- Rails encoraja o princípio **DRY**
- O layout da aplicação é mantida em um único local no arquivo **application.html.erb**
- O código comum dos templates ser reutilizado em **múltiplos templates**
- Por exemplo, os formulários do **edit** e do **new** - são realmente muito diferentes ?
- Partials são similares aos templates regulares, mas ele possuem capacidades mais **refinadas**
- Nomes de partials começam com **underscore** (**_**)
- Partials são renderizados com **render 'partialname'** (sem underscore)

Partials (2)

■ `_form.html.erb`

Listing 7: `views/posts/_form.html.erb`

```
1  <%= form_with  model: @post, local: true do |form| %>
2    <p>
3      <%= form.label :title %><br>
4      <%= form.text_field :title %>
5    </p>
6
7    <p>
8      <%= form.label :body %><br>
9      <%= form.text_area :body %>
10   </p>
11
12   <p>
13     <%= form.submit %>
14   </p>
15 <% end %>
```

Partials (3)

```
16 <%= link_to 'Back', posts_path %>
```

View: Edit (1)

- `edit.html.erb`:

Listing 8: view/posts/edit.html.erb

```
1 <h1>Edit Post</h1>
2 <%= render 'form' %>
```

Ação: Update (1)

- Recupera um objeto **post** utilizando o parâmetro **id**
- Atualiza o objeto **post** com os parâmetros que foram passados pelo formulário **edit**
- Tenta **atualizar** o objeto no **banco de dados**

Listing 9: posts_controller.rb

```
1 Class PostsController < ApplicationController
2   def update
3     @post = Post.find(params[:id])
4     @post.update(post_params)
5     redirect_to @post
6   end
7   ...
```

Agenda

- 1 Nova Postagem
- 2 Exibir Postagem
- 3 Atualizar Postagem
- 4 Reuso com Partial
- 5 Visão Destroy**
- 6 Controle de Versão
- 7 Para Saber Mais

Ação: Destroy (1)

- Remove uma postagem específica pelo parâmetro **id** passado como parte da URL

Listing 10: posts_controller.rb

```
1  class PostsController < ApplicationController
2    def destroy
3      @post = Post.find(params[:id])
4      @post.destroy
5      redirect_to posts_path
6    end
  end
```


Agenda

- 1 Nova Postagem
- 2 Exibir Postagem
- 3 Atualizar Postagem
- 4 Reuso com Partial
- 5 Visão Destroy
- 6 Controle de Versão**
- 7 Para Saber Mais

Hora de Colocar a Mão na Massa (1)

1. Implemente as visões e ações Index, Edit, Update, Partials e Destroy
2. Registre as mudanças realizadas no repositório local:

```
$ git add .
```

3. Efetive as mudanças realizadas no repositório local:

```
$ git commit -m "#2: manutencao de posts"
```

4. Registre as alterações no repositório remoto:

```
$ git push origin main
```

Agenda

- 1 Nova Postagem
- 2 Exibir Postagem
- 3 Atualizar Postagem
- 4 Reuso com Partial
- 5 Visão Destroy
- 6 Controle de Versão
- 7 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.