

Desenvolvimento de Software para a Web II – Roteiro 4

*Tema: Adicionando um título
dinâmico nas Páginas Estáticas*

Prof. Thiago Cavalcante

1. Os títulos (aqueles que aparecem nas abas do seu navegador) das páginas estáticas criadas no roteiro anterior (inicio, ajuda, sobre) estão todos iguais ao nome do seu app ("TerceiroApp"). Queremos que cada página exiba um título que a diferencie das outras. Em um projeto Rails, uma página que é exibida para o usuário é composta pelo **arquivo de layout** (que fica na pasta **app/views/layouts**) e pelos **arquivos de visualização** (que ficam na pasta **app/views/nome_do_controlador**). O arquivo de layout principal da aplicação, que é aplicado em todas as visualizações, fica em **app/views/layouts/application.html.erb**, criado automaticamente pelo comando **rails new**. Para implementar a modificação dos títulos das figuras e observar o funcionamento das visualizações, vamos ignorar (temporariamente) o arquivo de layout, renomeando-o com o comando a seguir

terminal, dentro da pasta do terceiro app

```
mv app/views/layouts/application.html.erb arq_layout
```

OBS.: Toda página HTML possui uma estrutura padrão, mostrada no quadro a seguir

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título da página</title>
  </head>
  <body>
    Conteúdo da página
  </body>
</html>
```

A seção **head** (cabeça) contém, entre outras informações, o título da página (que fica dentro de uma **tag <title>**) e a seção **body** (corpo) contém o conteúdo da página.

2. Cada página estática do seu app deve ter o título no formato "**(Nome) | Terceiro App Web II**", onde "**(Nome)**" é a parte variável do título (que depende da página) e "**Terceiro App Web II**" é a parte constante do título, que permanece sempre igual. Assim como no último roteiro, começaremos escrevendo o teste para o resultado que queremos obter. Para verificar o conteúdo de uma tag html, podemos usar a verificação **assert_select**. Com isso, o teste da página início pode ser atualizado com essa nova verificação.

test/controllers/paginas_estaticas_controller_test.rb

```
test "should get inicio" do
  get paginas_estaticas_inicio_url
  assert_response :success
  assert_select "title" , "Início | Terceiro App Web II"
end
```

Seguindo o modelo acima, atualize também os testes das páginas **ajuda** e **sobre**. Execute o comando **rails test**. Qual o resultado dos testes? **(responder por escrito)**

3. Inicialmente, vamos adicionar os títulos das páginas de forma **estática**. Você deve, no arquivo de visualização de cada página estática, adicionar o restante da estrutura básica de uma página html (como mostrado no quadro da primeira observação do roteiro). A visualização da página **início** fica como no quadro a seguir

app/views/paginas_estaticas/inicio.html.erb

```
<!--DOCTYPE html-->
<html>
  <head>
    <title>Início | Terceiro App Web II</title>
  </head>
  <body>
    <h1>Terceiro App</h1>
    <p>Este é o terceiro aplicativo Rails da disciplina
    <strong>Desenvolvimento de Software para a Web II</strong>
    do curso de Sistemas de Informação em
    <a href="http://ufal.edu.br/arapiraca/unidades-de-ensino/penedo">
      Penedo
    </a></p>
  </body>
</html>
```

Seguindo o modelo acima, modifique também as visualizações **ajuda.html.erb** e **sobre.html.erb**, trocando o título da página na seção **head** e também o conteúdo da página na seção **body**. Execute novamente o comando **rails test**. Qual o resultado dos testes? **(responder por escrito)**

4. **Exercício:** Elimine a repetição da parte constante do título no arquivo de testes do controlador de paginas estáticas adicionando uma definição para o método **setup** antes das definições dos testes, de acordo com o quadro a seguir

test/controllers/paginas_estaticas_controller_test.rb

```
def setup
  @titulo_base = "Terceiro App Web II"
end

test "should get inicio" do
  # Restante do código...
```

Substitua, então, **todas** as aparições do título base nos testes por **#{@titulo_base}**

test/controllers/paginas_estaticas_controller_test.rb

```
assert_select "title" , "Início | #{@titulo_base}"
```

5. Nesse momento, você pode comparar os arquivos de visualização das páginas estáticas do seu app e ver que: 1) os títulos (na seção **head**) de cada uma são bem semelhantes, a não ser pela parte relativa ao nome de cada página; 2) todos contém a estrutura básica de uma página html. Toda essa repetição de informações viola o princípio **DRY** (*Don't Repeat Yourself*, Não repita a si mesmo). Para eliminar essa

repetição, podemos começar usando **código Ruby embutido** nas visualizações. Você pode introduzir a geração dinâmica de cada título usando o método **provide** na primeira linha dos arquivos.

app/views/paginas_estaticas/inicio.html.erb

```
<% provide(:title, "Início") %>
<!DOCTYPE html>
<html>
  <head>
    <title><%= yield(:title) %> | Terceiro App Web II</title>
  </head>
  <body>
    <h1>Terceiro App</h1>
    <p>Este é o terceiro aplicativo Rails da disciplina
    <strong>Desenvolvimento de Software para a Web II</strong>
    do curso de Sistemas de Informação em
    <a href="http://ufal.edu.br/arapiraca/unidades-de-ensino/penedo">
      Penedo
    </a></p>
  </body>
</html>
```

O comando **provide** associa ao símbolo **:title** a string **"Início"**. O comando **yield**, por sua vez, obtém o valor previamente associado ao símbolo **:title** e o exibe na página. Seguindo o modelo acima, faça as modificações nas páginas **ajuda** e **sobre**, adicionando os comandos **provide** e **yield**.

6. Com essas modificações, todas as visualizações agora devem estar com a estrutura a seguir

```
<% provide(:title, "Início"/"Ajuda"/"Sobre") %>
<!DOCTYPE html>
<html>
  <head>
    <title><%= yield(:title) %> | Terceiro App Web II</title>
  </head>
  <body>
    Conteúdo da página inicio/ajuda/sobre
  </body>
</html>
```

Excluindo a primeira linha (com o comando **provide**) e o conteúdo de cada página, o que resta nas visualizações é essencialmente o conteúdo do arquivo de layout. Podemos agora desfazer o que foi feito no início do roteiro e retornar o arquivo de layout para a sua localização original, de forma que ele seja carregado pelo app.

terminal, dentro da pasta do terceiro app

```
mv arq_layout app/views/layouts/application.html.erb
```

Para que o layout funcione como desejado, você deve substituir a linha do título por aquela que está presente nas visualizações.

app/views/layouts/application.html.erb

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= yield(:title) %> | Terceiro App Web II</title>

    # Restante do código...
  </head>

  <body>
    <%= yield %>
  </body>
</html>
```

Outra modificação necessária é a remoção da estrutura html dos arquivos de visualização, agora que essa estrutura já está presente no layout. O comando **<%= yield %>** carrega o arquivo de visualização requisitado pelo controlador. O quadro a seguir mostra a visualização da página início após a correção.

app/views/paginas_estaticas/inicio.html.erb

```
<% provide(:title, "Início") %>
<h1>Terceiro App</h1>
<p>Este é o terceiro aplicativo Rails da disciplina
<strong>Desenvolvimento de Software para a Web II</strong>
do curso de Sistemas de Informação em
<a href="http://ufal.edu.br/arapiraca/unidades-de-ensino/penedo">
  Penedo
</a></p>
```

Faça o mesmo com as páginas ajuda e sobre. Execute o comando **rails test** mais uma vez. Qual o resultado dos testes? (responder por escrito)

7. **Exercício:** Seguindo os mesmos passos do roteiro anterior para criar a página **sobre**, crie uma página **contato** para o seu app. Você pode usar o conteúdo no quadro abaixo para sua visualização.

app/views/paginas_estaticas/contato.html.erb

```
<% provide(:title, "Contato") %>
<h1>Contato</h1>
<p>
  <ul>
    <li><strong>Nome: </strong>Aluno</li>
    <li><strong>Email: </strong>Aluno@email.com</li>
    <li><strong>Telefone: </strong>987654321</li>
  </ul>
</p>
```

8. Adicione os arquivos para serem salvos no git com o comando **git add -A** e faça um commit com suas modificações com **git commit -m "Adicionar título dinâmico nas páginas estáticas"**. Atualize o repositório do GitHub com **git push origin master** e o Heroku com **git push heroku master**.