

Desenvolvimento de Software para a Web II – Roteiro 5

*Tema: Preenchendo o layout;
Estilo; Visualizações parciais*

Prof. Thiago Cavalcante

1. Crie um novo branch no seu repositório para as modificações do roteiro com **git checkout -b preenchendo-layout**
2. Modifique a página inicial do seu app para a página estática **inicio** no arquivo de rotas

config/routes.rb

```
root 'paginas_estaticas#inicio'
```

3. Adicione um teste para a página inicial do seu app, para garantir que ela está sendo carregada corretamente

test/controllers/paginas_estaticas_controller_test.rb

```
test "should get root" do
  get root_url
  assert_response :success
end
```

4. Queremos melhorar a estrutura do site adicionando alguns elementos, como uma barra de navegação superior e um rodapé com links extras, que permitam ao usuário navegar entre as diferentes páginas criadas, e adicionando também estilos aos elementos da página usando CSS (*cascading style sheets*) e o *framework* Bootstrap. Para que esses novos elementos apareçam em todas as visualizações, devemos modificar o **layout** do app, no arquivo **app/views/layouts/application.html.erb**. Substitua o comando **<%= yield %>** no layout pelos comandos no quadro a seguir

app/views/layouts/application.html.erb

```
<header class="navbar navbar-fixed-top navbar-inverse">
  <div class="container">
    <%= link_to 'terceiro app', '#', id: 'logo' %>
    <nav>
      <ul class="nav navbar-nav navbar-right">
        <li><%= link_to, 'Início', '#' %></li>
        <li><%= link_to, 'Ajuda', '#' %></li>
        <li><%= link_to, 'Entrar', '#' %></li>
      </ul>
    </nav>
  </div>
</header>
<div class="container">
  <%= yield %>
</div>
```

A tag **header** contém os elementos que devem ficar no topo da página. Ela possui três classes CSS: **navbar**, **navbar-fixed-top** e **navbar-inverse**. As classes são **rótulos** que podem ser fornecidos a vários elementos em uma página HTML e são utilizados nos arquivos CSS para aplicar estilos. Essas classes da tag header (e todas as outras classes que começam com **navbar**) possuem um significado especial para o **Bootstrap**.

Dentro do elemento **header** temos um elemento **div**, que representa uma **divisão** na página. Ele possui a classe **container**, também especial para o **Bootstrap**.

Dentro do elemento **div**, temos um pedaço de **Ruby embutido**. Essa linha de código executa o método **link_to**, usado para criar uma tag HTML de *hyperlink*, **texto**. O primeiro parâmetro é o texto do link (**'terceiro app'**), o segundo parâmetro é o próprio link (**'#'**, apenas um substituto para o link de verdade que será adicionado posteriormente) e o terceiro parâmetro é um *hash* com opções adicionais.

Abaixo do código Ruby existe a tag **nav**, que indica o local dos links de navegação. Dentro desse elemento temos a tag **ul**, que representa uma lista não ordenada, e cada item da lista está em uma tag **li**. Cada item contém um link para uma outra página, criado também com o método **link_to**.

Por fim, abaixo do **header**, existe uma outra **divisão** na página, onde é inserido o conteúdo da visualização com o comando **yield**.

5. Para melhorar a estrutura da página início, substitua o seu conteúdo pelo conteúdo do quadro a seguir

app/views/paginas_estaticas/inicio.html.erb

```
<% provide(:title, "Início") %>

<div class="center jumbotron">
  <h1>Terceiro App</h1>

  <h2>
    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>
  </h2>

  <%= link_to 'Cadastre-se', '#', class: "btn btn-lg btn-primary" %>
</div>

<%= link_to image_tag("brasao.png", alt: "UFAL", width: "50px"),
  "https://ufal.br" %>
```

Essa nova página de início possui um botão com um link para que o usuário possa criar um novo cadastro. Esse link está na primeira chamada do método **link_to**, e as classes que começam com **btn**, em conjunto com o Bootstrap, garantem a aparência de um botão.

A segunda chamada do método **link_to** utiliza o método **image_tag** para incluir uma imagem com um link na página. A imagem escolhida foi o brasão da UFAL e o link leva o usuário para o site da UFAL. Para que a imagem apareça na sua página, você deve baixá-la nesse endereço (link clicável no PDF) e salvá-la na pasta **app/assets/images** do seu projeto.

6. Com as alterações realizadas até agora, as estruturas do seu layout e página inicial estão prontas. Se você rodar o servidor do seu app e entrar na página inicial, no entanto, vai perceber que a aparência da página não está organizada de uma forma agradável. Isso acontece pois não há nenhum estilo associado à página. Para isso, é necessária a utilização de CSS em conjunto com o framework Bootstrap. Para instalar o Bootstrap, adicione a linha **gem 'bootstrap-sass', '3.4.1'** no **Gemfile** do seu projeto, logo abaixo da linha **gem 'rails'** e

rode o comando **bundle install -without production** no terminal, dentro da pasta do projeto.

7. Para poder usar CSS no seu app, você deve criar um novo arquivo CSS na pasta **app/assets/stylesheets**, o qual será carregado pela aplicação quando o servidor for reiniciado. Você pode criar esse arquivo manualmente, ou pelo terminal usando o comando **touch app/assets/stylesheets/custom.scss**.
8. Com o Bootstrap instalado, você precisa incluir as linhas a seguir no seu arquivo **custom.scss** para carregar o Bootstrap em seu projeto.

app/assets/stylesheets/custom.scss

```
@import "bootstrap-sprockets";  
@import "bootstrap";
```

Se você reiniciar o servidor do seu app e atualizar a página inicial, vai ver uma mudança considerável na aparência da página.

9. Adicione as linhas a seguir no arquivo **custom.scss** para fazer mais ajustes no estilo do layout e da página inicial.

app/assets/stylesheets/custom.scss

```
body {  
  padding-top: 60px;  
}  
  
section {  
  overflow: auto;  
}  
  
textarea {  
  resize: vertical;  
}  
  
.center {  
  text-align: center;  
}  
  
.center h1 {  
  margin-bottom: 10px;  
}  
  
h1,  
h2,  
h3,
```

```

h4,
h5,
h6 {
  line-height: 1;
}

h1 {
  font-size: 3em;
  letter-spacing: -2px;
  margin-bottom: 30px;
  text-align: center;
}

h2 {
  font-size: 1.2em;
  letter-spacing: -1px;
  margin-bottom: 30px;
  text-align: center;
  font-weight: normal;
  color: #777;
}

p {
  font-size: 1.1em;
  line-height: 1.7em;
}

#logo {
  float: left;
  margin-right: 10px;
  font-size: 1.7em;
  color: #fff;
  text-transform: uppercase;
  letter-spacing: -1px;
  padding-top: 9px;
  font-weight: bold;
}

#logo:hover {
  color: #fff;
  text-decoration: none;
}

```

10. Para finalizar a estrutura do app, falta apenas criar o rodapé com a descrição do criador do site e alguns links adicionais (sobre e contato). Da mesma forma que a tag **header** cria um elemento na parte superior do site, a tag **footer** cria um elemento de rodapé no site. Adicione os comandos do quadro a seguir no seu arquivo de layout, **logo abaixo** do comando `<%= yield %>`.

app/views/layouts/application.html.erb

```
<footer class="navbar navbar-fixed-top navbar-inverse">
  <small>
    Terceiro App criado por Thiago
  </small>
  <nav>
    <ul class="nav navbar-nav navbar-right">
      <li><%= link_to, 'Sobre', '#' %></li>
      <li><%= link_to, 'Contato', '#' %></li>
    </ul>
  </nav>
</footer>
```

11. Faça o ajuste do estilo do seu rodapé adicionando o código CSS abaixo no arquivo **custo.scss**

app/assets/stylesheets/custom.scss

```
footer {
  margin-top: 45px;
  padding-top: 5px;
  border-top: 1px solid #eaeaea;
  color: #777;
}

footer a {
  color: #555;
}

footer a:hover {
  color: #222;
}

footer small {
  float: left;
}

footer ul {
  float: right;
  list-style: none;
}

footer ul li {
  float: left;
  margin-left: 15px;
}
```

12. Reinicie o seu servidor, atualize a página e observe o resultado das modificações feitas no roteiro, tanto na estrutura HTML quanto no estilo das visualizações. Você pode ver como estão as outras páginas, além da página inicial. Faça um commit com todas as modificações, usando **git add -A** e **git commit -m "Adicionando estilo e estrutura ao layout"**. Junte as modificações do seu

branch ao branch principal (**master**) do seu repositório com **git checkout master** e **git merge preenchendo-layout**. Atualize o GitHub com **git push origin master** e o Heroku com **git push heroku master**.