

03

Bootstrap e root url

Capítulo 7 - **Bootstrap e root URL**

Temos as duas listagens prontas em nossa página, porém a interface não foi ainda trabalhada. Vamos melhorá-la.

No site do *Bootstrap* (getbootstrap.com) existem alguns modelos de tabelas para nos basearmos em layouts já implementados. Acontece que o próprio *Ruby* já possui um pacote de plugins (gems) integrados com o *Rails*. Então, em vez de baixarmos os arquivos CSS e Javascript com o esqueleto do código no *Bootstrap* e direcioná-los para os diretórios específicos do nosso projeto, vamos instalar essas gems adicionando dependências Gemfile:

```
gem 'twitter-bootstrap-rails'
```

Salvamos o arquivo e agora vamos instalar. Para isso paramos o servidor e digitamos `bundle install` (download do código) e depois `rails g bootstrap:install static` (instalação nos diretórios). Pronto, gems instaladas! Levantamos o servidor (`rails s`) e vejamos a nova cara do site:

Nome	Descrição	Preço	Quantidade
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre tambem a camiseta	120.3	13
Camiseta de gastronomia	Camiseta super bacana gourmet	37.66	8
Camiseta do Big Bang Theory	Camiseta super bacana do bbt	70.5	10
Nome	Descrição	Preço	Quantidade
Camiseta de gastronomia	Camiseta super bacana gourmet	37.66	8
Camiseta do Big Bang Theory	Camiseta super bacana do bbt	70.5	10

Formatando a tabela

Apesar de mínima, já conseguimos notar alguma mudança. Vamos customizar com as Classes de exemplo do *Bootstrap*. A mais simples é a "table".

```
<html>
<body>
<table class="table">
  <thead>
    <tr>
      <th>Nome</th>
      <th>Descrição</th>
      <th>Preço</th>
      <th>Quantidade</th>
    </tr>
  </thead>
  <tbody>
    <% @produtos_por_nome.each do |produto| %>
    <tr>
      <td><%= produto.nome %></%></td>
      <td><%= produto.descricao %></%></td>
      <td><%= produto.preco %></%></td>
```

```

        <td><%= produto.quantidade %></%></td>
    </tr>
    <% end %>
</tbody>
</table>

<table class="table">
    <thead>
        <tr>
            <th>Nome</th>
            <th>Descrição</th>
            <th>Preço</th>
            <th>Quantidade</th>
        </tr>
    </thead>
    <tbody>
        <% @produtos_por_preco.each do |produto| %>
        <tr>
            <td><%= produto.nome %></%></td>
            <td><%= produto.descricao %></%></td>
            <td><%= produto.preco %></%></td>
            <td><%= produto.quantidade %></%></td>
        </tr>
        <% end %>
    </tbody>
</table>
</body>
</html>

```

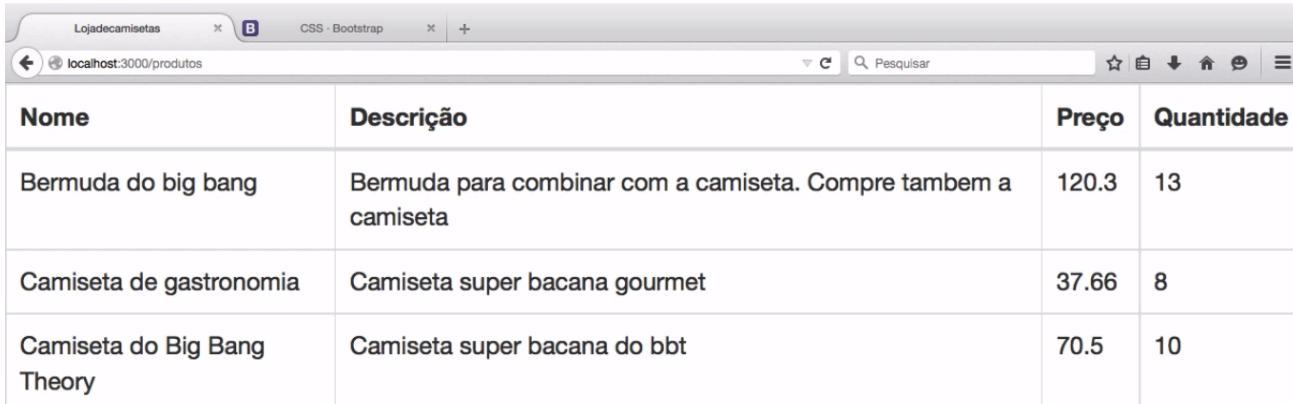
Vejamos o resultado:

Nome	Descrição	Preço	Quantidade
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre tambem a camiseta	120.3	13
Camiseta de gastronomia	Camiseta super bacana gourmet	37.66	8
Camiseta do Big Bang Theory	Camiseta super bacana do bbt	70.5	10

Nome	Descrição	Preço	Quantidade
Camiseta de gastronomia	Camiseta super bacana gourmet	37.66	8
Camiseta do Big Bang Theory	Camiseta super bacana do bbt	70.5	10

Podemos adicionar uma borda com

```
<table class="table table-bordered">
```



A screenshot of a web browser window titled 'Lojadecamisetas'. The address bar shows 'localhost:3000/produtos'. The page displays a table with four columns: Nome, Descrição, Preço, and Quantidade. The data in the table is:

Nome	Descrição	Preço	Quantidade
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre tambem a camiseta	120.3	13
Camiseta de gastronomia	Camiseta super bacana gourmet	37.66	8
Camiseta do Big Bang Theory	Camiseta super bacana do bbt	70.5	10

Nome	Descrição	Preço	Quantidade
Camiseta de gastronomia	Camiseta super bacana gourmet	37.66	8
Camiseta do Big Bang Theory	Camiseta super bacana do bbt	70.5	10

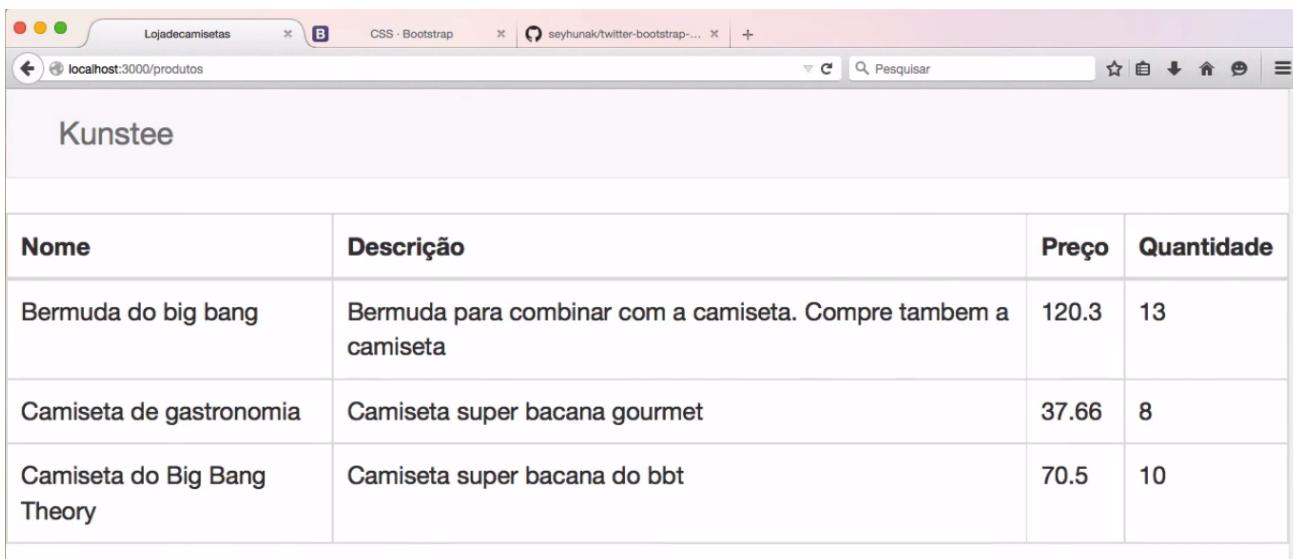
A barra de navegação - `nav_bar`

Estamos acostumados a ver em diversos sites uma barra de navegação no topo das páginas. Vamos adicionar essa `navbar`, com o nome da empresa. Ela deve ser responsiva para melhorar a navegação em diversas resoluções e será implementada pelo `Ruby`, o qual já possui métodos prontos. Então, no HTML:

```
<%= nav_bar brand: "Kunstee", brand_link: "/produtos" %>
```

- `brand`: acrescenta o nome;
- `brand_link`: acrescenta um destino ao clicarmos no nome.

O resultado é esse:



A screenshot of a web browser window titled 'Kunstee'. The address bar shows 'localhost:3000/produtos'. The page displays a table with four columns: Nome, Descrição, Preço, and Quantidade. The data in the table is identical to the previous screenshot. The 'Kunstee' logo is visible in the top left corner of the page area.

Nome	Descrição	Preço	Quantidade
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre tambem a camiseta	120.3	13
Camiseta de gastronomia	Camiseta super bacana gourmet	37.66	8
Camiseta do Big Bang Theory	Camiseta super bacana do bbt	70.5	10

Root

Para facilitar a navegação do usuário para nosso site, a página que criamos de produtos poderia ser a raiz, ou seja, seria melhor se não precisássemos escrever a URL "/produtos". Para tal fazemos no `routes`:

```
Rails.application.routes.draw do
  get "produtos" => "produtos#index"
  root "produtos#index"
end
```

Temos agora duas URL's para a mesma página:

- localhost:3000
- localhost:3000/produtos

Podemos confirmar quais rotas existem fazendo no Terminal `rake routes`:

```
rake routes
Prefix Verb URI Pattern      Controller#Action
produtos GET  /produtos(.:format) produtos#index
          root GET   /                  produtos#index
```

É basicamente o que vemos no arquivo de rotas. Para não termos duas rotas para a mesma página deletamos `get "produtos" => "produtos#index"`. Agora, além da rota deixar de existir, não conseguimos mais acessar "localhost:3000/produtos".

Havíamos implementado o link no nome da marca para que ele sempre leve para a página de produtos, que é a raiz. Acontece que se quisermos mudar a página que será raiz, o link estará quebrado (aliás neste momento está, pois excluímos sua rota), então generalizemos fazendo no HTML (e em todos que forem necessários):

```
<%= nav_bar brand: "Kunstee", brand_link: root_url %>
```

Temos inúmeras possibilidades para implemetar o design de nossa página. Teste-as e se utilize das documentações!