

Implementando a busca

Capítulo 10 - Implementando a busca

Agora conseguimos incluir e excluir produtos da nossa lista. Se tivermos muitos produtos incluídos, pode ficar difícil encontrar algum específico. O que implementaremos, então, um campo para busca.

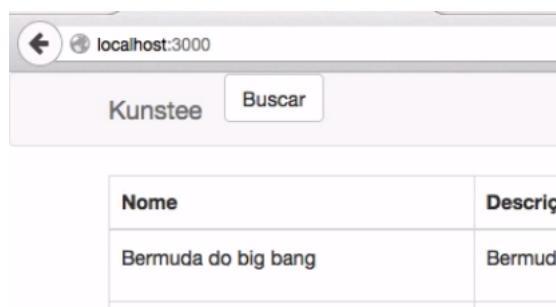
Poderemos acrescentar na barra de navegação uma *search engine*. A `nav_bar` pode receber um bloco de código usando `do` e fechando com `<% end %>`:

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

Vamos, primeiramente, criar um botão:

```
<%= nav_bar brand: "Kunstee", brand_link: root_url do %>
  <%= button_to "Buscar", nil, class: "btn btn-default" %>
<% end %>
```

Como não enviaremos nenhuma informação, passamos `nil`, e criamos um botão padrão (default). Por enquanto é assim que está nosso botão:



Implementamos o campo para a digitação e agrupamos os elementos (botão e campo):

```
<%= nav_bar brand: "Kunstee", brand_link: root_url do %>
  <div class="form-group">
    <%= text_field_tag :nome, nil, class: "form-control" %>
    <%= button_to "Buscar", nil, class: "btn btn-default" %>
  </div>
<% end %>
```

Como não estamos em um formulário de um objeto, usamos `text_field_tag`. O `nil` é o texto que digitaremos. Veja que, mesmo assim, está com um formato de formulário:

Nome	Descrição	Preço	Quantidade

Diferente do formulário completo que antes implementamos, aqui só usaremos sua *tag*:

```
<%= nav_bar brand: "Kunstee", brand_link: root_url do %>
<%= form_tag busca_produto_path, class: "navbar-form" do %>
  <div class="form-group">
    <%= text_field_tag :nome, nil, class: "form-control" %>
    <%= button_to "Buscar", nil, class: "btn btn-default" %>
  </div>
<% end %>
<% end %>
```

O `busca_produto_path` é a URL para qual queremos enviar as informações e usamos uma classe do Bootstrap para melhorar seu visual. Precisamos também criar uma rota para a busca, por padrão usamos get:

```
get "/produtos/busca" => "produtos#busca", as: :busca_produto
```

E no *controller*, para já podermos visualizar:

```
def busca
end
```

Com isso temos:

Nome	Descrição	Preço	Quantidade	
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre tambem a camiseta	120.3	13	<button>Remover</button>
Camiseta de gastronomia	Camiseta super bacana gourmet	37.66	8	<button>Remover</button>

Para jogar o campo de busca para a direita e definir seu método e busca:

```
<%= form_tag busca_produto_path,
            method: :get,
            class: "navbar-form navbar-right" do %>
```

Ainda não implementamos a busca em si. Precisamos criar a sua *regra de negócio*:

```
def busca
  nome = params[:nome]
end
```

No console conseguimos buscar produtos usando o método `where`:

```
irb(main):001:0> nome = "Bermuda do big bang"
=> "Bermuda do big bang"
irb(main):002:0> Produto.where(nome: nome)
```

O que retorna o item de mesmo nome com todas as informações (poderíamos usar os outros campos do produto). O problema é que esse método busca o valor inteiro. Se pesquisássemos apenas por "big bang", nada será retornado. Então usamos o `like`:

```
irb(main):003:0> resultado = Produto.where "nome like '%#{nome}%'"
```

ou

```
irb(main):003:0> resultado = Produto.where "nome like '%big bang%'"
```

O que retorna os produtos que tenham "big bang" no nome. Essa interpolação pode ser modificada por um usuário com má fé, com uma SQL injection, por exemplo. Então devemos tomar alguns cuidados:

```
irb(main):004:0> nome = %big bang%
```

```
irb(main):005:0> resultado = Produto.where "nome like ?", nome
```

Então temos duas maneiras de buscar no banco de dados:

- `where`: busca o nome, descrição, etc com valores idênticos ao fornecido
- `like`: busca os campos que contém o valor fornecido

E lembrando de usar o "?" para não corrermos riscos com a SQL injection. Vamos implementar isso no *controller*

```
def busca
  nome_a_buscar = "%#{params[:nome]}%"
  @produtos = Produto.where "nome like ?", nome_a_buscar
end
```

Vamos criar a *view* com nome de "busca.html.erb". Ela irá renderizar os resultados de pesquisa. Será também uma tabela:

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th>Nome</th>
      <th>Descrição</th>
      <th>Preço</th>
      <th>Quantidade</th>
    </tr>
  </thead>
  <tbody>
    <% @produtos.each do |produto| %>
```

```

<tr>
  <td><%= produto.nome %></%></td>
  <td><%= produto.descricao %></%></td>
  <td><%= produto.preco %></%></td>
  <td><%= produto.quantidade %></%></td>
  <td><%= button_to "Remover", produto,
                    method: :delete,
                    class: "btn btn-danger",
                    data: { confirm: "Tem certeza que deseja remover #{produto.nome}" } %></td>
</tr>
<% end %>
</tbody>
</table>

```



E agora sim, conseguimos fazer nossa busca:

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

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	

[Criar novo produto](#)

O que retorna:

Nome	Descrição	Preço	Quantidade	
Camiseta do Big Bang Theory	Camiseta super bacana do bbt	70.5	10	<button>Remover</button>
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre tambem a camiseta	120.3	13	<button>Remover</button>

O `:nome_a_buscar` está concatenado com o `"%"`. Poderíamos passar a interpolação direto para que o nome buscado permaneça na barra de escrita:

```

def busca
  @nome_a_buscar = params[:nome]
  @produtos = Produto.where "nome like ?", "%#{@nome_a_buscar}%""
end

```

E lá no arquivo "application", na `form_tag`:

```
<%= text_field_tag :nome, @nome_a_buscar, class: "form-control" %>
```

Agora o nome permanece no campo de busca depois dela ser feita:

The screenshot shows a web browser window with the URL `localhost:3000/produtos/busca?utf8=%E2%9C%93&nome=big+bang&authenticity_token=%2FJnznKuT3N7nY5FPGXoDzeoaTCk%2FmFODWE`. The search bar contains the text "big bang". Below the search bar is a table titled "Kunstee" showing two products: "Camiseta do Big Bang Theory" and "Bermuda do big bang". Each product has a "Remover" button in its respective row.

Nome	Descrição	Preço	Quantidade	
Camiseta do Big Bang Theory	Camiseta super bacana do bbt	70.5	10	<button>Remover</button>
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre tambem a camiseta	120.3	13	<button>Remover</button>