

Partial render convenção

Capítulo 12 - Partial render convenção

Nossas tabelas retornam o campo quantidade. Esta informação não é relevante para o usuário final. Pode parecer fácil de resolver esse problema, afinal bastaria excluir a coluna "Quantidade" de todas as tabelas. Este trabalho pode ser grande e cair em erro. Não parece uma boa prática se poderíamos fazer essa mudança em todas as tabelas ao mesmo tempo.

Existe um recurso que nos permite extrair todos os comportamentos que se repetem, que queremos reaproveitar em outras páginas, para um único arquivo. Ele só terá a parte do código que será copiada para as outras páginas, damos a ele o nome de *partial* e, para identificá-la usamos o símbolo de *underline*: `_produto.html.erb`. Salvamos dentro do diretório `/views/produtos` e, como é a tabela que nos interessa aqui, esse arquivo fica:

```
<% @produtos.each do |produto| %>
<tr>
  <td><%= produto.nome %></td>
  <td><%= produto.descricao %></td>
  <td><%= produto.preco %></td>
  <td><%= button_to "Remover", produto,
    method: :delete,
    class: "btn btn-danger",
    data: { confirm: "Tem certeza que
deseja remover #{produto.nome}?" } %></td>
</tr>
<% end %>
```

E onde antes estavam as tabelas (busca e raiz), renderizamos essa *partial*:

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th>Nome</th>
      <th>Descrição</th>
      <th>Preço</th>
    </tr>
  </thead>
  <tbody>
    <%= render partial: "produto" %>
  </tbody>
</table>
```

Collections

Isso, por enquanto, só irá funcionar para a tabela de busca, pois ela não tem uma ordem por quantidade ou por preço. Precisamos generalizar nossa *partial* para qualquer tipo de tabela. O primeiro passo é excluir a primeira linha do arquivo *partial*:

```

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

```

E nas tabelas do nosso `index` indicamos a coleção (ordem) que queremos renderizar:

```

<table class="table table-bordered">
  <thead>
    <tr>
      <th>Nome</th>
      <th>Descrição</th>
      <th>Preço</th>
    </tr>
  </thead>
  <tbody>
    <%= render partial: "produto", collection: @produtos_por_nome %>
  </tbody>
</table>

<table class="table table-bordered">
  <thead>
    <tr>
      <th>Nome</th>
      <th>Descrição</th>
      <th>Preço</th>
    </tr>
  </thead>
  <tbody>
    <%= render partial: "produto", collection: @produtos_por_preco %>
  </tbody>
</table>

```

A mesma coisa para a busca:

```

<table class="table table-bordered">
  <thead>
    <tr>
      <th>Nome</th>
      <th>Descrição</th>
      <th>Preço</th>
    </tr>
  </thead>
  <tbody>
    <%= render partial: "produto", collection: @produtos %>
  </tbody>
</table>

```

Usamos o método `collection` que traz o tipo de tabela que estamos querendo baseada nas variáveis do `controller` (`@produtos_por_nome` e `@produtos_por_preco`).

Uma forma mais resumida de renderizar as tabelas é fazendo:

```
<%= render @produtos %>

<%= render @produtos_por_nome %>

<%= render @produtos_por_preco %>
```

para cada caso.

Quando renderizamos as *partials* de coleções, a maneira mais tradicional é passar só cada coleção e seguir o padrão de nomear a *partial* com o *underline*.