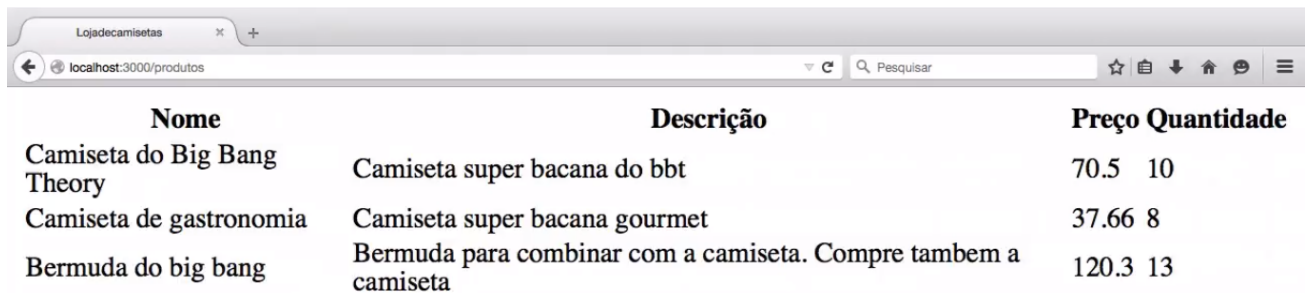


Controller Order Limit

Capítulo 5 - Controller Order Limit

Ordenando as lista de produtos

Já temos a nossa listagem no site.



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

O que queremos agora é dar uma ordem a ela, por exemplo, pelo nome do produto. No banco fazemos isso com:

```
irb(main):013:0> todos = Produto.all.order :nome
```

ou

```
irb(main):013:0> todos = Produto.order :nome
```

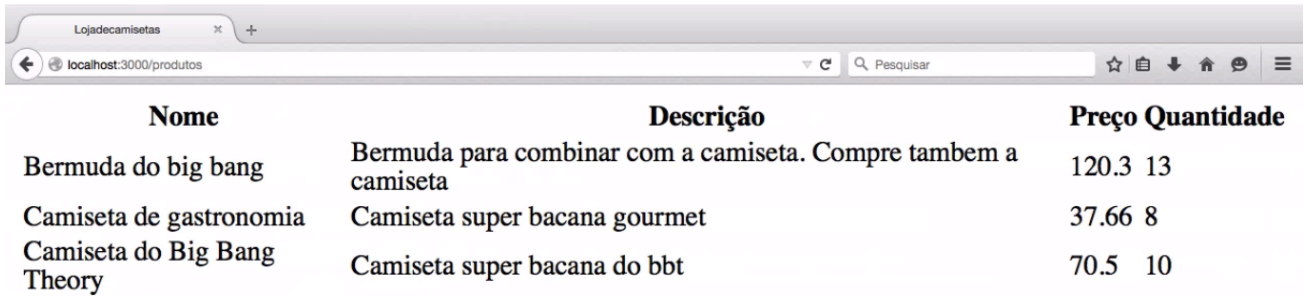
O que retorna o `array` ordenado no console. Para que o mesmo seja renderizado no navegador devemos usar a mesma ideia no código HTML:

```
<html>
<body>
<table>
  <thead>
    <tr>
      <th>Nome</th>
      <th>Descrição</th>
      <th>Preço</th>
      <th>Quantidade</th>
    </tr>
  </thead>
  <tbody>
    <% todos = Produto.order :nome %>
    <% todos.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>
```

```

</tbody>
</table>
</body>
</html>

```



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

Obviamente, poderíamos organizar pelo preço ou por qualquer outra informação do produto:

```

<html>
<body>
<table>
  <thead>
    <tr>
      <th>Nome</th>
      <th>Descrição</th>
      <th>Preço</th>
      <th>Quantidade</th>
    </tr>
  </thead>
  <tbody>
    <% todos = Produto.order :preco %>
    <% todos.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>

```

E, de fato, vendo os dois casos (copiamos o corpo da tabela inicial em baixo da mesma):

Nome	Descrição	Preço	Quantidade
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre também 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
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre também a camiseta	120.3	13

Regras de negócio

Se quisermos pegar tais informações e as colocarmos em outros lugares do site, não é uma boa prática deixarmos o pedaço de *query* `<% todos = Produto.order :[informação] %>` apenas na página da tabela. Esta é uma lógica que muito provavelmente se repetirá.

Antes de extrairmos a *query* para um local que será acessado por outras páginas vamos relembrar como funciona a URL `"/produtos"`. Quem faz toda a lógica da página *index* é o controlador de produtos:

```
get "produtos" => "produtos#index"
```

A *query* irá para dentro de um método com o mesmo nome da página (*index*) que, por sua vez, está dentro desse *controller* que foi criado no diretório `"app/controllers"`. Este será o lugar onde agruparemos as **regras de negócio**. Sem esse método, a rota nos mandava direto para a página. Após criado, a rota passa primeiro por ele, dentro do *controller*:

```
class ProdutosController < ApplicationController

  def index
    @produtos_por_nome = Produto.order :nome
    @produtos_por_preco = Produto.order :preco
  end

end
```

O `"@"` serve para dizer que a variável é de *instância* e estará disponível para visualização em todo o site. No *index* ela deve ter o mesmo nome que no *controller* (mudamos para melhor especificar):

```
<html>
<body>
<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>
  <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>

```

Vamos revisar o que vimos até agora:

- Digitamos "localhost:3000/produtos";
- A URL acessa as rotas e procura o controlador "produtos" com o método *index*;
- O controlador nos leva ao "ProdutosController", o qual possui tal método;
- O método executa a lógica de negócios (até agora temos apenas a ordenação);
- O método nos leva para a página que contém a tabela de produtos e a imprime.

Limitando o tamanho das listas

Se a lista de produtos for muito extensa (e será), não é bom que todos eles apareçam, precisamos limitar a nossa busca. Queremos, por exemplo, que na listagem por nome apareçam os primeiros cinco produtos e na por preço os dois primeiros. Fazemos isso no método *index* dentro do *controller*:

```

class ProdutosController < ApplicationController

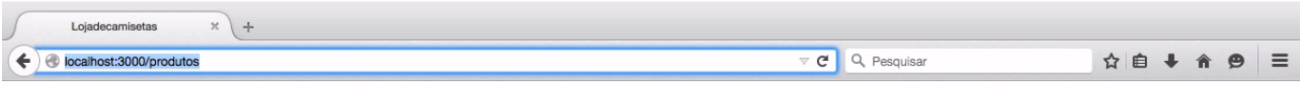
  def index

```

```
@produtos_por_nome = Produto.order(:nome).limit 5
@produtos_por_preco = Produto.order(:preco).limit 2
end

end
```

E, de fato:



Nome	Descrição	Preço	Quantidade
Bermuda do big bang	Bermuda para combinar com a camiseta. Compre também 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

Perceba que agora separamos os problemas e ficou mais fácil de saber onde buscar o quê no código:

- Se tivermos questões para resolver em relação à visualização, sabemos que devemos nos focar nos arquivos HTML no diretório "view".
- As questões que envolvem as lógicas de negócios serão tratadas dentro do "controller".
- Os modelos representam os produtos;
- O *Rails* conecta todas essas informações.