

## Exibindo uma página de produtos

### Transcrição

Vamos colocar os dados acerca do catálogo de produtos no nosso carrossel, dando uma olhada na nossa *view*, com uma estrutura que nos mostra que este HTML fixo foi pensado para que quatro produtos seguidos estejam em um mesmo grupo, denominado "página".

No carrossel, conforme clicamos no ícone branco de seta nas laterais, temos a alternância entre os grupos de quatro produtos sendo exibidos simultaneamente. Representaremos isto modificando o HTML no código para conseguirmos injetar nosso modelo.

Removeremos a `div` de classe `item`, ficando apenas com a de `item active`. E no lugar de quatro produtos, manteremos apenas o primeiro, a ser repetido várias vezes com informações diferentes, isto é, cada vez que ele for exibido, estará associado a um produto diferente.

Portanto, faremos uma estrutura de repetição, o `foreach` do C#, usando `@`. A repetição terá cada produto em uma página de produtos ( `produtosDaPagina` ) mas, antes, incluiremos um código C# com a declaração dessa coleção entre chaves. Pegaremos o modelo e obteremos somente os quatro primeiros elementos desta coleção de produtos, por meio do método `Take()`.

Feito isso, incluiremos a `div` que representará cada um dos produtos do carrossel dentro do `foreach`:

```
<div class="carousel-inner" role="listbox">
  <div class="item active">
    <div class="row">

      @{
        var produtosDaPagina = Model.Take(4);
      }

      @foreach (var produto in produtosDaPagina)
      {
        // código referente à div omitido
      }
    </div>
  </div>
</div>
```

E então trocaremos as informações por aquelas do nosso modelo de produtos. Em relação à imagem de cada produto do carrossel, por exemplo, temos um arquivo `large_001.jpg`, em que `001` pode ser representado pela propriedade código do nosso produto, sendo substituído portanto por `@produto.Codigo`, e ficando `large_@produto.Codigo.jpg`.

Da mesma forma, mais abaixo, onde se encontra o nome do produto, ASP.NET Core MVC, trocaremos para `@produto.Nome`, e no lugar do preço, `R$ 69,90`, deixaremos `R$ @produto.Preco`. Com isso, poderemos rodar a aplicação, atualizando a página.

Ocorrerá um erro de compilação, pois o programa está considerando que `jpg` (a extensão do arquivo de imagem), faz parte do código C#. Envolveremos esta parte com parênteses, pois assim o delimitaremos. O trecho ficará assim:

```
large_@(produto.Codigo).jpg
```

Ao rodarmos a aplicação novamente, veremos que o carrossel traz as informações provenientes do nosso modelo para a apresentação de uma página de produtos.