

Paginação do Carrossel de Produtos

Transcrição

Agora que conseguimos exibir a primeira página do nosso catálogo, tentaremos exibir as demais. Voltaremos à *view* do carrossel, e remover o código do primeiro laço, feito com base nos produtos do modelo. Identificaremos a `div` de classe `item active`, que mostra cada uma das páginas do carrossel.

Faremos um laço com código C#, iniciando com `@` e, entre chaves (`{}`), colocaremos a declaração de uma constante que será o tamanho de cada página, isto é, `4`. Em seguida, calcularemos a quantidade de páginas que queremos exibir, para o qual dividiremos a quantidade de produtos pelo tamanho da página.

Porém, esta conta não é o suficiente, pois caso o resultado seja um número não inteiro, precisaremos obter o menor número inteiro acima deste valor. Para tal, utilizaremos uma função da biblioteca de matemática do C#, o teto, ou *ceiling*, em inglês. Também será necessário converter o parâmetro de `Ceiling()` para *double*. Ainda armazenaremos o retorno do seu valor na quantidade de páginas que teremos na aplicação, e faremos um *cast* para tornar o `Math.Ceiling()` um `int`.

Depois, faremos um laço `for`, e então varreremos os produtos desta página em específico. Incluiremos a `div` neste laço e formataremos o documento com "Ctrl + K + D".

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

    @{
        const int TAMANHO_PAGINA = 4;

        int paginas = (int)Math.Ceiling((double)Model.Count() / TAMANHO_PAGINA);
    }

    @for (int pagina = 0; pagina < paginas; pagina++)
    {
        <div class="item active">...</div>
    }

</div>
```

Assim, temos os produtos da página e, em vez de `Model.Take(4)`, deixaremos `Model.Take(TAMANHO_PAGINA)`. Além disso, precisaremos pegar os produtos daquela página específica, para o qual chamaremos um método do LINQ (Consulta Integrada à Linguagem) do .NET, com `Model.`, pulando a quantidade de elementos correspondente ao número de páginas por meio do método `Skip()`:

```
<div class="item active">
    <div class="row">

        @{
            var produtosDaPagina = Model.Skip(pagina * TAMANHO_PAGINA).Take(TAMANHO_PAGINA);
        }

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

```
    }  
    </div>  
</div>
```

Rodaremos a aplicação mais uma vez e veremos todos os produtos, mas não é o que gostaríamos. Queremos exibir apenas uma página por vez, a qual ficaria rodando pelo carrossel. Vamos modificar a `div` de classe `item active` do CSS, que indica a página que está sendo exibida no momento — no caso, todas.

Deixaremos somente a primeira página ativa, aquela que possui índice igual a zero. Removeremos `active` e usaremos uma expressão C# com `@` e, entre parênteses, o operador ternário para verificar se a página é igual a zero. Se sim, será retornada a *string* com o nome da classe do CSS, `active` e, caso contrário, o retorno será uma *string* vazia:

```
<div class="item @(pagina == 0 ? "active" : "")">
```

Vamos rodar a aplicação novamente com "Ctrl + F5". Agora sim, estamos exibindo uma página por vez e conseguimos mover o carrossel para a direita ou para a esquerda usando as setas, e acessar todos os produtos do nosso catálogo.