



03

Formulário do pedido

Transcrição

Aqui também, segue o HTML completo desenvolvido nesse vídeo:

```
<html>
  <head>
    <meta charset="UTF-8" />
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0h"
crossorigin="anonymous">    </head>
  <body>
    <div class="container">
      <div class="jumbotron mb-0">
        <h1 class="display-4">Novo Pedido</h1>
      </div>

      <form action="/pedido/novo" method="POST">
        <div>
          <label for="nomeProduto">Produto</label>
          <input name="nomeProduto"
placeholder="nome do produto"/>
          <small>Informa qual o nome do produto.
</small>
        </div>
        <div>
          <label for="urlProduto">Url</label>
          <input name="urlProduto"
```

```
placeholder="url do produto"/>
        <small>Procure em um site o produto que
você deseja e cole a url da página aqui.</small>
    </div>
    <div>
        <label for="urlImagem">Imagem</label>
        <input name="urlImagem" placeholder="url
da imagem"/>
        <small>Copie a url de uma boa imagem do
produto e cole aqui para ajudar a conseguir compradores.
</small>
    </div>
    <div>
        <label for="descricao">Descrição</label>
        <textarea name="descricao"
placeholder="Adicione detalhes importantes para ajudar o
comprador."></textarea>
    </div>
    <button type="submit">Cadastrar</button>
</form>
</div>
</body>
</html>
```

[COPIAR CÓDIGO](#)

[00:00] Já temos o nosso menu, a primeira parte do topo. Ainda falta a logo marca lá em cima, mas nós vamos fazer em um outro momento. Ao clicar nesse botão “Novo”, nós somos redirecionados para uma requisição, que é “/pedido/formulario” - que ainda não existe.

[00:16] Não existe ainda nem o “Controller” que vai tratar todas as requisições para “Pedido”. Por exemplo: “criar pedido”, “atualizar pedido”, “remover pedido”. Nós vamos fazer um “Controller” específico. Vamos agrupar as “actions” de tudo o que tem haver com pedido nesse “Controller”.

[00:32] E a requisição para `/pedido/formulario` ainda não tem um `Controller` e não tem a página específica que deve ser aberta. Então o que vamos fazer agora é aqui no pacote de `Controllers`, nós vamos criar uma nova classe chamada de `PedidoController` e todas as requisições que forem para `/pedido`, vão ser direcionadas para esse `Controller`.

[00:54] Ou seja, esse é o `Controller` que já vai dizer que ele vai tratar todas as requisições para `Pedido`. Então, o que vamos fazer? Primeiramente, vamos integrar essa classe com o Spring. Nós dizemos que essa classe é um `Controller`, onde tem esse `stereotype` de `Controller` do próprio Spring MVC.

[01:12] E a outra coisa é utilizarmos o `request`. Nós usamos o `@GetMapping`. Nós vamos fazer o `@RequestMapping` porque daí podemos mapear todos os tipos de requisições. Então `get` é um tipo de requisição. Com o `@RequestMapping` nós estamos dizendo o quê? Todas as requisições que forem para pedido vão bater nesse `Controller` aqui. Nós vamos fazer um `action` específico para o formulário. Então, `public String`.

[01:36] `String` é o nome da página que deve ser aberta - `formulario` e aqui em cima nós vamos anotar com o `@GetMapping` dizendo que `/pedido` já está mapeada no `Controller` no nível da classe e `formulario` vai estar mapeado no nível da `action` do método. Então, `formulario/pedido/formulario`.

[01:58] E o que esse método tem que fazer? Ele tem que apenas retornar a página que deve ser renderizada para o usuário. Qual é o HTML? Então vamos retornar. No `homeController` nós usamos `home` apenas.

[02:11] Aqui dentro de `templates`, página `home`. No caso do `pedido`, a princípio ele pode ter várias páginas associadas ao tratamento. À criação do pedido? Tem formulário. Atualização do pedido? Tem outro formulário de atualização, ou pode ser o mesmo. E outras páginas específicas que podem surgir só para `pedido`.

[02:31] Então o que faremos é criar uma pasta específica e colocar todos os HTMLs aqui dentro. Então vamos retornar a página “formulario”, mas dentro da pasta “pedido”. Beleza? Nós vamos colocar todas as páginas dentro dessa pasta.

[02:53] Então, pronto! Agora dentro da pasta “pedido” nós vamos criar uma página nova chamada “formulario.html” e essa página especificamente é assim. Então vamos começar com o topo que nós já temos da página principal. Já temos o Jumbotron.

[03:15] Vou copiar do “home” toda a parte inicial do topo que envolve a importação do Bootstrap CSS, a “

” e o Jumbotron para o “Meus Pedidos”, que aqui, na verdade, é diferente. Olhando no exemplo, é “Novo Pedido”. Então eu vou alterar esse texto para “Novo Pedido”.

[03:37] Eu vou fechar a “container”, a “body” e a “html”. Lembrando que a “container” serve para centralizar o conteúdo e definir um tamanho padrão para a página que é redimensionável. De acordo com o tamanho, ele se ajusta com uma largura diferente. Então todo o conteúdo do site fica dentro desse “container”.

[03:56] Então nós já temos a página de “Novo Pedido”, mas ela só tem o topo. Só tem esse aqui, “Novo Pedido”. Se nós abrirmos de novo, então, “/pedido/formulario”, já está abrindo. Ou seja, a requisição já está vindo nesse método e essa “action” está dizendo: “abra a página ‘formulário’ “. Ou seja, ela não tem nenhuma lógica específica, senão redirecionar para essa tela de formulário.

[04:17] O que temos que fazer? Já está funcionando. Então, o que temos que fazer nessa página? Vamos primeiro nesse vídeo criar o HTML específico e depois nós continuaremos o desenvolvimento em outro momento. Então, o que temos que colocar de HTML aqui?

[04:32] Sem Bootstrap e sem nada, só o HTML em si para já conseguirmos receber uma requisição, para que o nosso sistema consiga receber uma requisição do usuário. Tem que ter um item para colocarmos o produto que envolve um “label”, um “input” e uma explicação embaixo, além do “placeholder” e do “input”. Tem também a mesma coisa para “URL”, a mesma coisa para “Imagem” e “Descrição” também. Só que descrição não é o “input”, é um “textarea” e não tem essas explicações embaixo.

[04:58] Então as estruturas desses três primeiros envolvem... Vou criar uma tag “form” e aqui dentro vou criar uma “div”, e dentro dessa “div” vou colocar os três componentes que envolvem o produto. “Label”, “input” e esse texto embaixo que eu vou colocar em uma tag “small”. Então primeiro o “label” com o conteúdo “Produto” e depois o “input”. Nós já vamos dar um “name=“nomeProduto” ”.

[05:25] Vou aproveitar que eu dei um “nome” e vou colocar no “label” - “label for = “nomeProduto” e embaixo, usando essa tag “small” do HTML, eu vou colocar “Informe qual o nome do produto”. Então, “Informe qual o nome do produto”.

[05:41] Beleza! Então isso já está sendo mostrado sem estilo. Então só tem o “input” e tudo na mesma linha. Não tem endentação, não tem posicionamento. Mas tudo bem, já tem o “input” aqui. Vou copiar esse aqui para fazer também... Lembrando que tem o “placeholder” que eu vou colocar também, “NomeProduto”.

[06:00] “placeholder” é um atributo do “input”, onde você pode colocar um texto “default” que não é um texto preenchido. “NomeProduto”. Como isso aparece no HTML? Aperte a tecla “F5”, aparece aqui, mas na verdade não está preenchido. Quando você começa a escrever, ele some. Então é só uma ajuda para a pessoa saber o que tem que colocar dentro desse campo.

[06:28] Depois tem o “URL”. Então vamos colocar “Url” no “label”. Em vez de “nome do produto” é “url do produto”. No “name” também coloco “url do

Produto” e ele tem o “placeholder”, que é “url do Produto” também.

[06:44] E no texto embaixo tem, “Procure em um site o produto que você deseja”. Então vamos lá, em partes. “Procure em um site o produto que você deseja”. Beleza, o que mais? “E cole a URL da página aqui”. Então a “url do Produto” já está ok. Depois tem a “url da Imagem”.

[07:13] Então vou trocar para “imagem” como está lá. No “label” fora vou colocar a “urlImagem” e vou copiar isso e colar no “name” do “input”. E o “placeholder” a “url da Imagem”. Então muda aqui para “imagem”. Troquei no errado. Estou fazendo tudo no “url” em vez de na “imagem”.

[07:45] “url da imagem”, “urlImagem”, “Imagem” e o texto, “Copie a URL de uma boa imagem do produto e cole aqui para ajudar a conseguir compradores”. A ideia aqui é que se você for pedir para alguém trazer um produto para você, coloque a imagem lá que dê para ver, por exemplo, mais ou menos o tamanho.

[08:13] Imagine se você coloca lá um produto e quando a pessoa vai ver é cinco vezes o tamanho. Então se você colocar uma imagem que ajude a pessoa a entender é mais fácil de conseguir um comprador. Beleza? Então esse é o objetivo de colocar uma imagem e essa dica aqui embaixo.

[08:32] Mais uma “div” que vou colocar aqui. Vai ser a “Descricao”, aqui “descricao”. No “name” eu vou colocar “descricao” também. No “placeholder”, “Adicione detalhes importantes para ajudar o comprador” e ele não tem essa tag “small”.

[09:00] Para terminarmos, vamos colocar um “button”. Esse sim do tipo “submit” porque o objetivo dele é submeter todas as informações que a pessoa colocou nos “inputs”. Aliás, submeteremos isso para o formulário. Esse último de “descricao” está errado.

[09:13] “Cadastrar” no botão, mas não é um “input”, é um “textarea”. No “textarea” você precisa fechar assim. Então, vamos ver como ficou? “Produto”, “Url”, “Imagem”, “Descricao” e “Cadastrar”. Beleza! Todas as informações que você colocar aqui vão ser enviadas quando eu clicar no botão “Cadastrar”.

[09:40] Mas enviadas para onde? Então isso não vai submeter tudo o que você preencheu no formulário? Vai submeter para onde? Você tem que dizer, “Submeta isso para a “/pedido/novo”. Pronto! E aí você põe só o método aqui porque quando nós vamos criar uma coisa assim. Nós usamos o método “post”. Se não as informações vão na “URL” e aí é ruim. É ruim por vários motivos. Mas vamos lá!

[10:08] “**Cadastrar**”. Nós já conseguimos receber essas informações no “Controller”. Então nós já temos a página. Atualizou. Ela já está fazendo a requisição, “pedido/novo” e ele mandou as requisições para lá. Pronto!

[10:27] Se você olha aqui e aperta a tecla “F12”... Deixe-me só abrir de novo, só para ter certeza. Eu apertei a tecla “F12” no Chrome e ele abre essa parte. Normalmente ele abre na lateral. Eu é que jogo ele para baixo porque eu acho mais fácil de ver.

[10:39] Na aba “networks” você consegue ver as requisições que você faz. Então, por exemplo: cliquei em “Cadastrar” e aqui está a requisição que fizemos e lá embaixo aparece todos os parâmetros que nós mandamos, que são os parâmetros que preenchemos no formulário. Aqui não está preenchido.

[10:54] Se eu colocar aqui algumas informações, ele vai aparecer lá embaixo. Eu vou clicar de novo. Pronto, todos os valores que eu preenchi para cada “input” estão aqui mapeados em chave e valor! Onde a chave é o “name” do “input” que colocamos aqui e o valor é o “valor” que o usuário preencheu dentro do “input”. Então, pronto! É assim que nós mandamos as informações para o back-end.

[11:16] Então essa página vai ser renderizada no Browser do usuário e quando ele clicar em “Cadastrar” ele vai fazer uma requisição que vai chamar o nosso sistema. Então vamos ter que fazer um método para receber essas informações que nós preenchemos no formulário.

[11:30] Então vamos fazer no próximo, a recepção dessas informações e salvar esse pedido no banco de dados e depois aplicaremos o layout aqui. Continuaremos também fazendo validações de erro.