



Apresentação

Transcrição

[00:00] No treinamento anterior terminamos a aplicação com essa estrutura de pacotes, com esses *templates*, então estamos trabalhando com o Spring e no front-end estamos trabalhando com o Thymeleaf, então temos Bootstrap aqui para fazer em todo o *layout*. Esse *layout* é responsivo, nossa aplicação está com a interface gráfica bem adaptável e a dispositivos com tamanhos de telas diferentes.

[00:27] Temos aqui no back-end a "HomeController.java", principalmente, e o "PedidoController.java"; onde conseguimos criar os pedidos, cadastrar novos pedidos e apresentar pedidos do usuário. Aqui já vamos começar evoluindo a questão de segurança de autenticação e autorização da aplicação.

[00:47] Vamos ter páginas: além do usuário logado, tela de login, salvar usuário com senha criptografada no banco de dados e vamos configurar páginas que são públicas e páginas que são privadas.

[00:59] Hoje temos, por exemplo, essa *home* aqui, onde estamos listando todos os pedidos, estamos chamando-a de "Meus Pedidos" - o que não faz muito sentido porque estamos buscando todos os pedidos que estamos salvando no banco.

[01:11] A partir do momento em que criarmos o usuário logado, vamos conseguir criar pedidos e associar esses pedidos ao usuário que está autenticado na aplicação, e aí sim essa página de "Meus Pedidos" vai se tornar a página principal do usuário logado.

[01:26] Depois vamos criar uma nova *home*, que vai ser a *home* pública - em que o usuário não vai precisar estar autenticado, é a página principal da aplicação. Inclusive, onde serão listados os últimos pedidos entregues, como propaganda mesmo para o nosso sistema, mostrando que ele funciona, que ele tem pessoas utilizando, etc.

[01:50] Vamos evoluir então essa camada de segurança. Estamos utilizando o Thymeleaf para fazermos o nosso front-end, nós vamos mudar isso também. Nós vamos criar uma funcionalidade de geração de ofertas para esses pedidos usando o Vue.js, que é um *framework* de front-end.

[02:10] Então nós vamos criar uma aplicação de front-end se comunicando com a nossa aplicação Mudi aqui em Java, de back-end; e a aplicação de front-end vai se comunicar com a aplicação de back-end utilizando tecnologia REST, usando uma biblioteca que faz requisições AJAX chamada Axios.

[02:31] Vamos entender o que é essa tecnologia REST, também vamos entender o que é o Vue.js, qual é a ideia desses *frameworks* como o React, Angular e o próprio Vue, que é o mais fácil de usar. Vamos entender como funciona isso, qual a diferença de trabalhar com Thymeleaf para trabalhar com uma aplicação de front-end.

[02:55] E no final nós vamos ter uma aplicação. Eu vou até fazer um *checkout*, vou até atualizar aqui para o estado final da aplicação, para você ver a atualização que vai ter aqui a nível de pacotes, a nível de interface gráfica também nos nossos *templates*. Então a nossa aplicação vai crescer bem, vai ter uma página de oferta, página para o usuário, a *home* do usuário, a *home* pública mesmo e página de pedidos.

[03:23] Aqui no "oferta", à esquerda, estamos utilizando o Vue.js. Vocês vão ver no decorrer do treinamento. Vamos trabalhar com *interceptor*, vamos trabalhar com *cache* também, paginação usando o Spring Data e vamos integrar. No final vamos ter um produto com todas essas tecnologias bem integradas e você vai entender como tudo funciona.

[03:47] Por mais que seja uma solução toda funcional, ela vai ter várias oportunidades para você evoluir. Como é título de exercício mesmo, próprio em casa, você pode decidir, por exemplo: "Vou continuar desenvolvendo essa aplicação com Thymeleaf" ou "vou migrar tudo para Vue.js e fazer uma aplicação de front-end se comunicando com tudo via REST, com aplicação do back-end" - você decide.

[04:16] Você vai ter então uma solução aqui, uma aplicação, no final, com todas as possibilidades de evoluir em várias funcionalidades diferentes. Isso vai ser um bom portfólio também, vai agregar no seu portfólio de maneira bastante positiva. Então é isso, vamos lá! Até o próximo vídeo!