



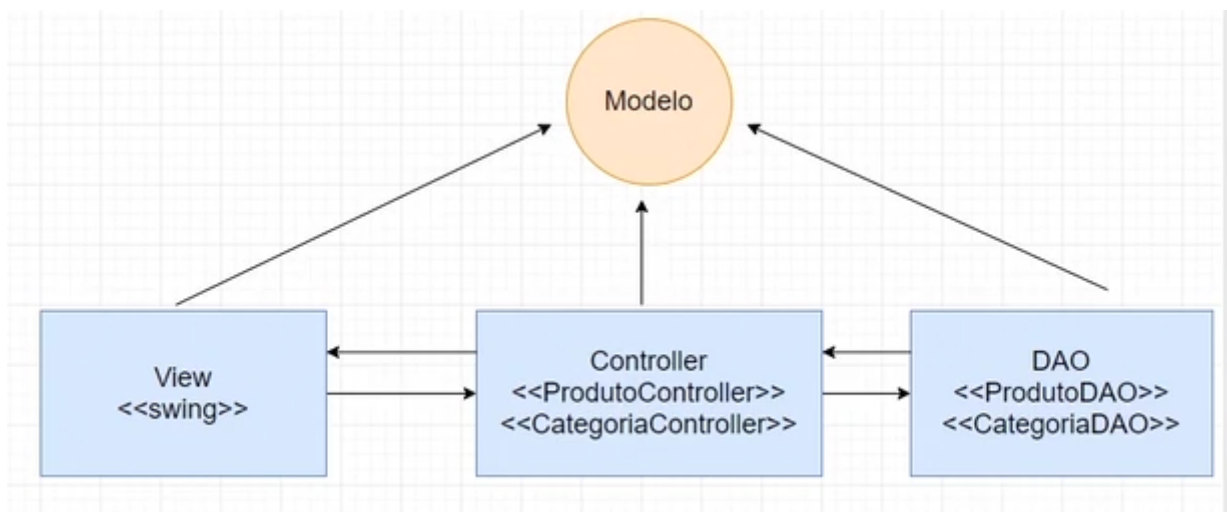
Apresentando as camadas da aplicação

Transcrição

[00:00] Olá, aluno. Tudo bom? Na aula passada, nós conhecemos a maneira como os nossos usuários vão fazer agora a requisição para o nosso *back-end*, a nossa aplicação. Nós vimos que será construída uma tela, onde o usuário poderá preencher campo para ser inserir, poderá excluir dados, poderá alterar produtos da nossa base de dados sem se preocupar como o código faz isso.

[00:25] É o dia a dia, o usuário, ele não vai acessar o código. Nós fizemos o curso para entendermos como funciona, mas agora expomos para o usuário a partir de uma *view*. E nós vimos então onde queremos chegar. Antes de partir para o desenvolvimento, precisamos entender um pouco sobre camadas, porque não é simplesmente a nossa *view* chegando na nossa DAO, fazendo a operação e pronto, acabou.

[00:54] Nós temos uma separação de camadas para que o nosso código, ele fique mais correto, que fique mais fácil de dar manutenção nesse código. Então, para isso, eu trouxe um desenho, onde nós vamos ver exatamente o que nós estamos fazendo. Então eu tenho uma *view*, aqui nós estamos utilizando o Swing, que é do próprio Java.



[01:17] Aqui eu abro um parênteses já para falar para vocês que o Swing, ele serve para aplicações Desktop. Só que hoje em dia, dificilmente vocês verão o desenvolvimento, começar um projeto para desenvolver para Desktop. Vamos ver para mobile, para web. Desktop já é algo mais antigo.

[01:38] Mas antes de passarmos para o próximo passo, que é conhecer o desenvolvimento web, que inclusive é até a nossa próxima formação, precisamos ter um conhecimento dessas camadas, de como funciona a comunicação entre as camadas e isso podemos fazer com o Swing, que vocês vão aproveitar esse conhecimento quando vocês forem para os próximos cursos, para o desenvolvimento web.

[02:04] Pensando nisso, com o Swing sendo o nosso *front-end*, a nossa *view*, temos que fazer a requisição para a nossa aplicação, que automaticamente vai no banco de dados para fazer alguma operação com o banco de dados. Só que a nossa *view*, ela não conecta diretamente com a nossa DAO.

[02:24] Nós não podemos atrelar essas duas pontas porque a *view*, ela serviria apenas para receber requisição e não deveria, pelo menos não deveria, ter lógicas, por exemplo, de abertura de conexão, porque a nossa DAO, ela precisa, no momento em que eu instancio o meu `ProdutoDAO`, a minha `CategoriaDAO`, eu preciso passar a minha conexão.

[02:49] Então eu não posso, ou então eu não deveria - não posso, não, não deveria - fazer isso na nossa *view*. Então, para isso, eu crio um controlador.

Esse controlador é quem vai receber a requisição da minha *view* e vai saber para onde tem que enviar para a nossa DAO. Ele fornece uma conexão para a nossa DAO. Quando a DAO processa a informação, devolve para a Controller, a Controller me devolve essa informação e a *view* apenas mostra.

[03:19] Então eu tenho agora um conceito de Controller, que é de fato um controlador. Nós vamos ver que tudo o que é trafegado entre essas caixas, entre as nossas camadas, é o nosso Modelo, que é o nosso `Categoria.java`, que é o nosso `Produto.java`, que tem os nossos atributos, que tem os nossos getters, os nossos setters.

[03:41] Então esse Controller, ele vai servir também para ser trafegado entre essas camadas. Nós já tínhamos as nossas DAOs, agora nós já vimos que nós temos a *view*, o que nós precisamos fazer é implementar a nossa Controller e, especialmente essa comunicação entre as camadas. É isso que é o nosso objetivo e é isso que vamos desenvolver nessa sessão do nosso curso.

[04:10] Então a ideia aqui era trazer para vocês o conhecimento dessas camadas, o porquê agora eu tenho uma Controller, o que trafega entre essas camadas e agora, o objetivo é fazermos de fato isso no código e botar a mão na massa, e ver a tela funcionando, mas agora com a nossa implementação. Então não deixem de assistir a próxima aula e vamos botar a mão na massa. Valeu.