



## Conclusão

### Transcrição

[00:00] Olá, pessoal! **Parabéns!!** Chegamos ao final do nosso treinamento. Espero que vocês tenham gostado. Nesse treinamento nós continuamos o primeiro de JPA, pegamos o nosso projeto e complementamos, estudamos mais recursos da JPA.

[00:15] Nós focamos bastante na parte de consultas, então vimos como fazer consultas com funções de agregação - *sum*, *max*, *avg*, *min*, enfim. Consultas com `SELECT new`, principalmente utilizadas em relatórios. Como seria essa consulta sem o `SELECT new`, usando um *List* de *array* de objeto, que era um pouco meio complicado.

[00:33] Vimos também a questão do JOIN FETCH para fazer uma *query* planejada. Mudamos as nossas entidades para os relacionamentos *to one* serem todos como *lazy*, então só carregamos as informações quando precisar, evitamos *joins* desnecessários que podem gerar gargalos na aplicação.

[00:51] Aprendemos a fazer *queries* com parâmetros dinâmicos, do jeito tradicional com JPQL e utilizando API de *criteria*, que é meio confuso, eu particularmente não gosto muito. Aprendemos esses detalhes de performance, aprendemos também a parte de herança. Na nossa classe "Produto", vimos as principais estratégias para herança.

[01:13] Criamos algumas subclasses, como "Livro" e "Informatica", que estão herdando dessa entidade. Vimos também, na classe "Cliente", como dar uma organizada no código, extrair atributos para classes específicas, usando o `@Embedded` e o `@Embeddable`, principalmente para quando você tem campos

como endereço, telefone, e não ficar enchendo uma entidade com zilhões de atributos.

[01:39] Só porque estão na mesma tabela não precisam estar na mesma classe, podem estar quebrados, então nem sempre uma classe, uma entidade, é um espelho exatamente igual do que está na tabela, nós podemos seguir boas práticas de orientação a objetos no mundo Java, no mundo da orientação a objetos.

[01:54] Por fim, vimos também a parte de chave composta, usando o `@EmbeddedId` e o `@Embeddable` também. Então acabamos fazendo esses recursos, inclusive no último vídeo, ficou dando um problema na classe "Produto", por causa da `@NamedQuery`, que foi mais um recurso que nós vimos no treinamento.

[02:11] Essa *named query*, por conta da chave composta, deu problema no *WHERE*, que antes estava `p.categoria.nome`. Só que agora o nome está dentro do ID da categoria, então é `p.categoria.id.nome`. Feito isso, resolveu o problema. Então diversos recursos que nós aprendemos aqui durante esse treinamento, complementando os estudos sobre JPA.

[02:32] Espero que vocês tenham gostado, agradeço bastante por vocês terem acompanhado até aqui. Espero que vocês tenham aprendido bastante sobre a parte de JPA, como é que funciona a JPA por baixo dos panos. É muito importante conhecermos, principalmente quem está trabalhando com JDBC e quer sair do JDBC, quer entender como funciona a JPA, só sabe o básico.

[02:52] Ou quem utiliza *frameworks*, como Spring Boot ou outros *frameworks* que usam a JPA por baixo dos panos, mas que também criam mais camadas de abstração. Às vezes não entendemos muito bem o que está acontecendo ali por baixo, porque o *framework* está abstraindo muitos conceitos.

[03:08] Então aqui foi esse justamente o foco, por isso que o nosso projeto não tem nenhum *framework*, não é uma aplicação *web*, não depende de servidor,

para entendermos na raça como funciona a JPA e os seus recursos. E tomar cuidado com questões de problemas, questões de performance e entender certo a parte mapeamento. Vejo vocês em outros treinamentos aqui da Alura. Muito obrigado, um abraço e até lá.