



## Solução proposta

### Transcrição

[00:00] Então vamos ver como que ficou aquele desafio proposto. Vale ressaltar que essa é só a minha maneira de escrever aquele desafio. Não necessariamente você tem que seguir o que está aqui. Aqui é mais para que você tenha um comparativo de como você fez e de como eu fiz.

[00:21] É importante ressaltar também que as entidades que eu criei aqui, os nomes e as estruturas de entidades, vão ser importante para que sejam feitas as consultas. Então na hora que formos fazer consultas, se caso você utilizou algum nome diferente, tem que ser feita conforme as entidades que você criou. Então vamos lá. Nós já havíamos criado as entidades de Cargo.

[00:49] A entidade de Cargo, a CargoRepository e a CargoCrud. Então basicamente e essa é uma das vantagens que o Spring Data também nos traz, a padronização das coisas. Por quê? Porque basicamente as coisas vão ser muito parecidas umas com as outras. As coisas mais diferentes vão ser exatamente os relacionamentos do JPA. Então vamos ver como eu fiz.

[01:16] Vou primeiro mostrar a entidade de Cargo que já havíamos feito e vou mostrar aqui que eu fiz uma associação com funcionário utilizando a notação OneToMany. A entidade de Unidade de Trabalho também tem uma relação com funcionário e ela é basicamente a mesma coisa que Cargo.

[01:39] Agora a que tem um diferencial maior é a entidade de Funcionário porque dentro de Funcionário você vai ter as relações tanto com Cargo quanto com o Unidade de Trabalho fazendo uma ManyToOne e uma ManyToMany

porque o funcionário só pode ter um Cargo e um funcionário também pode trabalhar em N Unidade de Trabalho.

[02:04] Então ficou assim a minha relação entre as entidades. E vamos ver como que ficaram os nossos Repositorys. Esse era o Repository de Cargo. Não teve nenhuma mudança nele. Veja que nós estendemos o CrudRepository e tal. Aqui nós temos de Unidade de Trabalho.

[02:26] Deixa eu maximizar para que vocês possam ver que eu coloquei um `integer` porque o ID de Unidade de Trabalho é um `integer`. E repare que a estrutura é exatamente idêntica a de Cargo. Então você entendendo como que funciona, a estrutura continua a mesma. Então você tem uma padronização de como descrever as coisas.

[02:45] Você não tem que ficar memorizando muitas coisas, muita decoração. E de funcionário, exatamente também a mesma coisa. Você estende de um Repository e colocam lá as suas entidades e os seus IDs. Para finalizar vamos ver como ficou os nossos Crud. Aqui nós já havíamos feito o nosso Crud de Cargo. Ele continuou da mesma coisa, não teve nenhuma mudança.

[03:14] Vou mostrar para vocês também o Crud de Unidade de Trabalho e veja que ele é idêntico ao Crud de Cargo. Porque quando você está utilizando o Spring Data, as formas de fazer um *update*, de visualizar, deletar as coisas é muito parecida. Então as classes acabam tendo uma estrutura muito semelhante.

[03:36] A única classe que teve alguma alteração, algum diferencial e não por causa da forma de utilizar o Spring Data, mas sim da forma de dar a opção para o cliente salvar mais Unidade de Trabalho dentro de um Funcionário, foi exatamente o Crud de Funcionário. Por quê? Deixa eu vir aqui. Aqui eu tive que fazer uma sub-rotina para dar a liberdade de ele digitar várias Unidades de Trabalho.

[04:09] Por exemplo, ele pode vir aqui e pode entrar na Unidade 1, 2 e 3 e quando ele digitar 0, aí sim ele vai sair desse *loop* de Unidade de Trabalho. Mas basicamente é também a mesma estrutura que foi utilizada nos demais para fazer o Crud. Então não teve muita alteração na estrutura de código para fazer nenhum dos três. Essa foi a minha interpretação de como fazer esse desafio e nos vemos na próxima.