



## Modelo e o DAO de categorias

### Transcrição

[00:00] Fala, aluno. Tudo bom? Voltando à nossa aplicação, agora com a tabela de categorias criadas e inclusive até com algumas categorias adicionadas, nós podemos trabalhar com aquelas informações que estão no banco de dados. A primeira coisa que eu vou fazer é criar uma classe, e eu vou botar a "TestaListagemDeCategorias".

[00:29] E eu quero que essa classe já tenha um main, porque nós vamos fazer o teste da nossa aplicação já referente às categorias. O que eu fazer, nós vamos usar aquele mesmo padrão que nós estávamos utilizando com produto, então nós vamos recuperar uma conexão, passar essa conexão no construtor de uma DAO de categorias e na DAO, vamos fazer algo semelhante ao que nós com produto também, o método de listar.

[01:06] Para isso vamos criar uma lista, botar por enquanto de categoria, ele não vai reconhecer ainda, vou botar `List<Categoria> listaDeCategorias = categoriaDAO.listar();` . Foi isso aqui mais ou menos que nós fizemos com Produto . Para `TestaListagemDeCategorias` , eu quero que agora tenha `CategoriaDAO categoriaDAO = new categoriaDAO(connection);` .

[02:03] Essas duas linhas vão estar dentro de um try com recursos, então já vamos trazer as duas linhas para dentro de `try()` . E eu vou fazer o seguinte, eu quero pegar uma `try(Connection connection = new ConnectionFactory.recuperarConexao())` . Nós fizemos isso, mas por enquanto não temos nada referente à categoria, não temos um `categoriaDAO` , não temos uma classe de modelo para a categoria.

[02:49] Então vamos fazer o seguinte, eu quero criar essa classe "CategoriaDAO" e eu quero botar ela no pacote "br.com.alura.jdbc.dao". Pronto, já criamos a CategoriaDAO . Ele vai receber, no seu construtor, uma public CategoriaDAO(Connection connection) e vai existir o this.connection = connection; .

[03:19] Lembrando que esse this.connection não existe, vamos criar uma connection . Agora o que nós queremos é devolver, no método de listar, uma lista de categorias. Então o que nós vamos fazer é um public List<Categoria> listar() , Categoria não existe eu vou chamar listar() . Já vamos instanciar uma lista de categorias, que nós vamos precisar, e eu vou colocar List<Categoria> categorias = new ArrayList<>(); .

[04:00] Esse ArrayList<> ainda não existe, então vou deixar ele vazio. Vamos escrever na nossa String sql = "SELECT ID, NOME FROM CATEGORIA"; . Criei essa string, agora eu já recebi a conexão, vamos preparar o Statement, então eu vou usar o try com recursos, vou fazer o seguinte: try(PreparedStatement pstmt = connection.prepareStatement(sql)) , bem semelhante ao que nós fizemos com Produto .

[04:44] Então já vou mandar executar, com pstmt.execute(); . Agora eu quero fazer o seguinte, eu quero pegar o resultado disso, então vamos fazer try(ResultSet rst = pstmt.getResultSet()) . O que eu tenho que fazer agora é fazer o laço, então enquanto eu tiver um próximo, um próximo resultado, uma próxima linha no meu banco de dados, cria uma categoria e vamos criar um novo objeto do tipo categoria.

[05:33] Então while(rst.next()) e Categoria categoria = new Categoria; . Categoria não existe ainda, vamos então criar a classe "Categoria". Vamos botar ela no pacote "br.com.alura.jdbc.modelo". Agora Categoria existe. O que nós vamos fazer nela é criar um private Integer id; , private String nome; e já vamos criar um construtor, que vai receber public Categoria(Integer id, String nome) .

[06:14] E esses atributos vão ser `this.id = id`, `this.nome = nome`; . Agora nós temos então uma Categoria. O que ele está reclamando aqui? Como não temos mais o construtor padrão, nós precisamos fazer aquele esquema que nós fizemos com `Produto`. Eu já vou pegar o `new Categoria(rst.getInt(1), rst.getString(2));`, o `getInt` na coluna 1, que é o ID e vou pegar o `getString` da coluna 2.

[07:05] O que ele está falando? Vamos adicionar o `throws SQLException`, porque senão ele não vai parar de reclamar. Com esse código em mãos, o que eu preciso fazer agora é adicionar `categorias.add(categoria);`. Então a cada laço, a cada iteração, ele vai pegando uma informação do banco de dados, transforma em um objeto do tipo categoria e salva na lista de categorias.

[07:34] Se eu mandar adicionar um `return Categorias;`, ele vai retornar a nossa lista, bem semelhante ao que fizemos para produto. E agora, o que nós precisamos fazer em `TestaListagemDeCategoria`? Importar categoria, que nós não temos ainda, adicionar o `throws SQLException`.

[07:53] Agora, com essa lista de categorias, eu quero pegar o nome da categoria. Então vamos fazer um `listaDeCategorias.string.forEach(ct -> ct.getNome());`. Esse `getNome()` ainda não existe, então vamos criar o `getNome()` de `Categoria`, com `public String getNome()` e `return nome;`.

[08:30] Com isso, agora eu faço o seguinte, um `listaDeCategorias.stream().foreach(ct -> System.out.println(ct.getNome()));` em `TestaListagemDeCategoria`. Pronto, com esse código agora, se eu mandar executar é para ele trazer as nossas categorias. Vamos ver. Temos agora a nossa categoria de eletrônicos, a nossa categoria de eletrodomésticos e a nossa categoria de móveis.

[09:06] Aparentemente tudo ok. Muito bom, aluno. A nossa aula era só para criarmos, como havíamos prometido nas últimas aulas, mais sobre essa parte que envolve categorias, era exatamente criar as nossas DAOs e a nossa classe

modelo. Objetivo cumprido, agora vamos ver os próximos passos. Espero que vocês tenham gostado e até o próximo vídeo.