Murilo Alves

Instruções Versão word.

\*\*Instruções para fazer a API funcionar\*\*

01.Primeiramente é necessário utilizar o backup do banco de dados no postgress.

02.Depois abrir a pasta com a API no netbeans(ou outro semelhante) para configurar a classe Conexao.java (do mesmo jeito que se configura na vídeo aula).

03.Depois de configurar a classe Conexao, a API está pronta para uso.

04.A API foi feita com base na vídeo aula da professora.

05.O menu consiste no mesmo exemplo apresentado na vídeo aula, mas com opções adicionais para o CRUD COMPLETO.

06.Números de 1 a 5 - Servem para inserir dados nas tabelas: cliente, funcionario, produto, pedido e cliente respectivamente.

07.Números de 6 a 10 - Servem para exibir os dados das tabelas: cliente, funcionario, produto, pedido e cliente respectivamente.

08.Números de 11 a 15 - Servem para atualizar os dados das tabelas: cliente, funcionario, produto, pedido e cliente respectivamente.

09.Números de 16 a 20 - Servem para deletar os dados das tabelas: cliente, funcionario, produto, pedido e cliente respectivamente.

10.Números de 21 a 24 - São operações de processo de negócio.

11.Números de 25 a 27 - São os relatórios.

Obs sobre os relatórios.

25 - Relatório sobre os produtos com material de qualidade mediano, a reposta será o código do produto e o código do material que possui qualidade mediano.

26 - Relatório sobre os pedidos de clientes chamados murilo(minúsculo), a reposta será o código do pedido e o código dos clientes que se chamam Murilo.

27 - Relatório sobre os pedidos com produtos de tamanho pequeno, a resposta será o código do pedido e o código do produto que possui o tamanho pequeno.

Existe um arquivo SQL na pasta para tirar a prova real dos relatórios no postgress.

Obs sobre o menu.

1.Toda vez que um input for acionado como por exemplo o relatório 25, a reposta ira aparecer e o menu também logo abaixo fazendo com que a resposta fique mais em cima.

2.As operações delete e update do crud foram feitas com base no inserir, logo eles funcionam de maneira semelhante.

Com um adendo, o delete precisa apagar os dados com base no código, ou seja se você quiser deletar um cliente, a API leva em conta apenas o codigoCliente para apagar o cliente, porém na hora de colocar o input a API vai pedir por todas as informações do cliente devido a lógica que utilizei, pode colocar qualquer outro valor aleatório nos outros campos(com base nos tipos certo).

Por exemplo, se quiser deletar o cliente com código 4, basta digitar 4 quando a API pedir e nos outros campos como "nome, telefone..." você pode colocar dados aleatórios.

3.O update possui um input semelhante ao inserir, você começa colocando o código daquilo que quer dar update e depois coloca os dados novos que você quer em cada lugar. Por exemplo, digamos que você quer atualizar um material, depois de selecionar o número devido no menu, a API vai pedir pelo código do material e em seguida basta colocar os dados que você, para manter o mesmo dado basta digitar ele novamente.