

Desafio Master: Sistema de Logística de Entregas

Objetivo

Criar uma aplicação Java via terminal utilizando JDBC + Maven que simula a gestão de entregas de uma empresa de e-commerce, com foco em cadastro de clientes, motoristas, pedidos e entregas, além de relatórios analíticos.

Entidades e Campos

- Cliente: id, nome, cpf_cnpj, endereco, cidade, estado
- Motorista: id, nome, cnh, veiculo, cidade_base
- Pedido: id, cliente_id, data_pedido, volume_m3, peso_kg, status (PENDENTE, ENTREGUE, CANCELADO)
- Entrega: id, pedido_id, motorista_id, data_saida, data_entrega, status (EM_ROTA, ENTREGUE, ATRASADA)
- HistoricoEntrega: id, entrega_id, data_evento, descricao

Regras de Negócio

- Um cliente pode ter vários pedidos.
- Cada pedido pode ou não estar associado a uma entrega.
- Um motorista pode realizar várias entregas.
- Cada entrega pode ter múltiplos eventos registrados no histórico.

Menu de Funcionalidades

- 1 - Cadastrar Cliente
- 2 - Cadastrar Motorista
- 3 - Criar Pedido
- 4 - Atribuir Pedido a Motorista (Gerar Entrega)
- 5 - Registrar Evento de Entrega (Histórico)
- 6 - Atualizar Status da Entrega
- 7 - Listar Todas as Entregas com Cliente e Motorista
- 8 - Relatório: Total de Entregas por Motorista
- 9 - Relatório: Clientes com Maior Volume Entregue
- 10 - Relatório: Pedidos Pendentes por Estado
- 11 - Relatório: Entregas Atrasadas por Cidade
- 12 - Buscar Pedido por CPF/CNPJ do Cliente

Desafio Master: Sistema de Logística de Entregas

- 13 - Cancelar Pedido
- 14 - Excluir Entrega (com validação)
- 15 - Excluir Cliente (com verificação de dependência)
- 16 - Excluir Motorista (com verificação de dependência)
- 0 - Sair

Requisitos Técnicos

- Projeto Maven com driver JDBC (MySQL ou PostgreSQL)
- Uso de PreparedStatement e ResultSet
- Camadas: model, dao, view, service
- Entrada via Scanner
- Manipulação de datas com java.time
- LEFT JOIN, GROUP BY, filtros e condições
- Boas práticas de modelagem relacional

Critérios de Avaliação Sugeridos

- Funcionamento dos relacionamentos - 30
- Uso correto de JOINS e GROUP BY - 20
- Relatórios úteis e organizados - 20
- Estrutura de código em camadas - 20
- Criatividade e recursos extras - 10