DOCUMENTAÇÃO: SISTEMA DE GERENCIAMENTO E-COMMERCE

1. Tratativas do Banco de Dados

1.1 Entidade clientes

- Descrição: Armazena informações sobre os clientes da loja.
- Atributos:
  - id_cliente (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único do cliente.
  - nome_cliente (VARCHAR(200), NOT NULL): Nome completo do cliente.
  - cpf (VARCHAR(11), UNIQUE, NOT NULL): CPF do cliente.
  - email (VARCHAR(100), UNIQUE, NOT NULL): Endereço de e-mail do cliente.
  - senha (VARCHAR(50), NOT NULL): Senha de acesso do cliente (armazenada de forma segura).
  - data_nascimento (DATE): Data de nascimento do cliente.
- Relações:
  - Um cliente pode ter múltiplos endereços (enderecos).
  - Um cliente pode fazer múltiplos pedidos (pedidos).
  - Um cliente pode utilizar vários cupons de desconto (cupons_desconto).

1.2 Entidade produtos

- Descrição: Armazena informações sobre os produtos disponíveis na loja.
- Atributos:
  - id_produto (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único do produto.
  - nome_produto (VARCHAR(100), NOT NULL): Nome do produto.
  - cod_produto (VARCHAR(10), UNIQUE, NOT NULL): Código único do produto.
  - preco_unitario (DECIMAL(10, 2), NOT NULL): Preço unitário do produto.
  - qtd_estoque (INT, NOT NULL): Quantidade do produto em estoque.
  - id_categoria (INT, FOREIGN KEY): Referência à categoria do produto.
- Relações:
  - Um produto pertence a uma categoria (categorias).
  - Um produto pode estar em múltiplos itens de pedido (itens_pedido).

1.3 Entidade pedidos

- Descrição: Armazena informações sobre os pedidos realizados pelos clientes.
- Atributos:
    - id_pedido (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único do pedido.
    - id_cliente (INT, FOREIGN KEY, NOT NULL): Referência ao cliente que fez o pedido.
    - data_pedido (DATE, NOT NULL): Data em que o pedido foi realizado.
    - status (VARCHAR(50), NOT NULL): Status atual do pedido (e.g., "Pendente", "Em processamento", "Enviado", "Entregue").
- Relações:
    - Um pedido pertence a um cliente (clientes).
    - Um pedido pode conter múltiplos itens (itens_pedido).
    - Um pedido possui um ou mais pagamentos (pagamentos).

1.4 Entidade itens_pedido

- Descrição: Representa cada item dentro de um pedido, com detalhes sobre a quantidade e preço de cada produto.
- Atributos:
    - id_item_pedido (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único do item de pedido.
    - id_pedido (INT, FOREIGN KEY, NOT NULL): Referência ao pedido que contém o item.
    - id_produto (INT, FOREIGN KEY, NOT NULL): Referência ao produto incluído no pedido.
    - quantidade (INT, NOT NULL): Quantidade do produto no pedido.
    - preco_unitario (DECIMAL(10, 2), NOT NULL): Preço unitário do produto no momento do pedido.
- Relações:
    - Um item de pedido pertence a um pedido (pedidos).
    - Um item de pedido referencia um produto específico (produtos).

1.5 Entidade categorias

- Descrição: Armazena as categorias dos produtos, permitindo a organização e o agrupamento de produtos semelhantes.
- Atributos:
    - id_categoria (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único da categoria.

    - nome_categoria (VARCHAR(100), NOT NULL): Nome da categoria.
    - descricao (TEXT): Descrição da categoria.
- Relações:
    - Uma categoria pode conter múltiplos produtos (produtos).

1.6 Entidade pagamentos

- Descrição: Armazena informações sobre os pagamentos de cada pedido.
- Atributos:
    - id_pagamento (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único do pagamento.
    - id_pedido (INT, FOREIGN KEY, NOT NULL): Referência ao pedido pago.
    - data_pagamento (DATE, NOT NULL): Data em que o pagamento foi realizado.
    - valor_pago (DECIMAL(10, 2), NOT NULL): Valor pago no pedido.
    - metodo_pagamento (VARCHAR(50), NOT NULL): Método de pagamento utilizado (e.g., "Cartão", "Boleto", "Pix").
- Relações:
    - Um pagamento está vinculado a um pedido (pedidos).

1.7 Entidade funcionarios

- Descrição: Armazena informações sobre os funcionários responsáveis pelo gerenciamento da loja.
- Atributos:
    - id_funcionario (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único do funcionário.
    - nome_funcionario (VARCHAR(200), NOT NULL): Nome completo do funcionário.
    - cpf (VARCHAR(11), UNIQUE, NOT NULL): CPF do funcionário.
    - email (VARCHAR(100), UNIQUE, NOT NULL): E-mail do funcionário.
    - senha (VARCHAR(50), NOT NULL): Senha de acesso do funcionário.
    - data_admissao (DATE, NOT NULL): Data de admissão do funcionário.
- Relações:
    - Funcionários podem ter permissões para gerenciar diferentes entidades, como produtos e pedidos.

1.8 Entidade avaliacoes

- Descrição: Armazena as avaliações feitas pelos clientes sobre os produtos.
- Atributos:

    - id_avaliacao (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único da avaliação.
    - id_cliente (INT, FOREIGN KEY, NOT NULL): Referência ao cliente que fez a avaliação.
    - id_produto (INT, FOREIGN KEY, NOT NULL): Referência ao produto avaliado.
    - nota (INT, NOT NULL): Nota dada pelo cliente ao produto (e.g., de 1 a 5).
    - comentario (TEXT): Comentário opcional sobre o produto.
    - data_avaliacao (DATE, NOT NULL): Data em que a avaliação foi realizada.
- Relações:
    - Uma avaliação é feita por um cliente (clientes) sobre um produto específico (produtos).

1.9 Entidade enderecos

- Descrição: Armazena os endereços dos clientes, utilizados para entregas.
- Atributos:
    - id_endereco (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único do endereço.
    - id_cliente (INT, FOREIGN KEY, NOT NULL): Referência ao cliente.
    - logradouro (VARCHAR(200), NOT NULL): Logradouro (rua, avenida, etc.).
    - numero (VARCHAR(10), NOT NULL): Número do endereço.
    - complemento (VARCHAR(100)): Complemento do endereço.
    - bairro (VARCHAR(100), NOT NULL): Bairro.
    - cidade (VARCHAR(100), NOT NULL): Cidade.
    - estado (VARCHAR(50), NOT NULL): Estado.
    - cep (VARCHAR(8), NOT NULL): Código postal.
- Relações:
    - Um cliente pode ter múltiplos endereços (clientes).

1.9.1 Entidade cupons_desconto

- Descrição: Armazena cupons de desconto que podem ser aplicados pelos clientes em seus pedidos.
- Atributos:
    - id_cupom (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único do cupom.
    - codigo_cupom (VARCHAR(10), UNIQUE, NOT NULL): Código do cupom.
    - percentual_desconto (DECIMAL(5, 2), NOT NULL): Percentual de desconto aplicado pelo cupom.
    - data_validade (DATE, NOT NULL): Data de validade do cupom.
- Relações:
    - Um cupom pode ser utilizado em múltiplos pedidos.

1.9.2 Resumo das Relações

- 1
  entre clientes e enderecos: Um cliente pode ter múltiplos endereços.
- 1
  entre clientes e pedidos: Um cliente pode realizar múltiplos pedidos.
- 1
  entre pedidos e itens_pedido: Cada pedido pode conter múltiplos itens.
- N:1 entre produtos e categorias: Cada produto pertence a uma categoria.
- 1
  entre clientes e avaliacoes: Cada cliente pode fazer múltiplas avaliações.
- 1
  entre produtos e avaliacoes: Cada produto pode ter múltiplas avaliações.
- 1
  entre pedidos e pagamentos: Cada pedido pode ter múltiplos pagamentos.

2. Tratativas da programação procedural da Aplicação JAVA

2.1 Estrutura do Projeto Java

Pacotes principais:

1. MODEL - Contém as classes que representam as entidades do banco de dados (também conhecidas como POJOs ou Data Transfer Objects - DTOs).
2. DAO - Contém as classes de acesso a dados (Data Access Objects - DAOs) responsáveis pela comunicação com o banco de dados.
3. SERVICE - Contém as classes de serviço que encapsulam a lógica de negócios e manipulam os dados.
4. VIEW - Contém as classes para a interface gráfica, se for implementado um front-end.
5. UTIL - Contém utilitários, como configuração de conexão com o banco de dados e manipuladores de exceções.

2.1.0 Pacote UTIL: Classes de conexão com o banco de dados utilizando o DriverManager

2.1.1 Pacote MODEL: Contém as classes que representam a entidades (tabelas) do banco de dados. cada atributo dessas classes correspondem às colunas de cada tabela, além de possuir construtores getters e setters.

2.1.2 Pacote Data Access Object (DAO): No pacote DAO, deverá existir para cada entidade da tabela uma classe, de modo que essa classe conterá todos os métodos que realizam operações CRUD no banco de dados sobre essa entidade.

ClienteDAO -> Contém os métodos insertCliente, getAllClientes, updateCliente, deletteCliente, getClienteById, etc.

Assim como haverão classes produtosDAO, pedidosDAO, itens_pedidoDAO...

2.1.3 Pacote SERVICE: As classes de serviço encapsulam a lógica de negócios e chamam os métodos dos DAOs para manipular os dados. Um exemplo seria ClienteService, que validaria dados antes de inserir no banco.

A classe Service linka os métodos DAO e cria uma estrutura de "menu" para cada um destes métodos.

2.1.4 Pacote VIEW: Interface Gráfica. Para criar uma interface gráfica, você pode usar o Swing ou JavaFX. A interface gráfica deve interagir com as classes de serviço, para que cada ação do usuário chame a lógica de negócios e, em última análise, manipule o banco de dados.