

# Desenvolvimento de CRUD para a Tabela Clientes – Sistema E-commerce

**Integrantes:** João Pedro Natividade Ferreira e Luiz Phillipe de Souza Santos

## Estrutura

- **Cliente:** Representa a estrutura de dados do cliente.
- **ClienteADO:** Gerencia as operações CRUD no banco de dados.
- **Main:** Interface de usuário via console para interação com o sistema.

## Classe Cliente

Essa classe representa um cliente, encapsulando os dados e métodos relacionados ao gerenciamento das informações de cada cliente.

- **Atributos:**
  - **clienteId:** Um identificador exclusivo para cada cliente.
  - **nome, email, telefone:** Informações de contato do cliente.
  - **dataCadastro:** A data em que o cliente foi cadastrado no sistema.
- **Construtor:**
  - **Cliente(int clienteId, String nome, String email, String telefone, java.sql.Date dataCadastro):** Cria uma nova instância de Cliente com os dados fornecidos.
- **Métodos:**
  - **Getters e Setters:** Métodos acessores para manipulação de atributos, garantindo o encapsulamento. Facilita o uso da classe em operações CRUD, permitindo acesso controlado aos dados do cliente.

## Classe ClienteADO

Esta classe é responsável por realizar operações de acesso a dados relacionadas ao cliente no banco de dados, utilizando JDBC para comunicação com o banco.

- **Atributos:**
  - **connection:** Gerencia a conexão com o banco de dados para execução de operações SQL.
- **Construtor:**
  - **ClienteADO():** Inicializa a conexão usando propriedades armazenadas em um arquivo db.properties, que contém as configurações (URL, nome de usuário, senha) para configurar a conexão com o banco.

- **Métodos CRUD:**

- **cadastrarCliente:** Insere um novo cliente no banco de dados, utilizando PreparedStatement para garantir segurança e evitar SQL Injection.
- **listarClientes:** Retorna uma lista de todos os clientes cadastrados. Executa uma consulta para buscar todos os registros.
- **buscarCliente:** Pesquisa um cliente específico por ID, retornando um objeto Cliente se encontrado.
- **atualizarCliente:** Atualiza as informações de um cliente existente com base no ID. Usa PreparedStatement para garantir segurança.
- **excluirCliente:** Exclui um cliente pelo ID.
- **fecharConexao:** Fecha a conexão com o banco para liberar os recursos.

## Classe Main

Esta classe contém o ponto de entrada da aplicação, sendo responsável pela interface de usuário via console.

- **Objetivo:**
  - Oferecer uma interface básica que permite ao usuário realizar operações de gerenciamento de clientes por meio de um menu de opções.
- **Estrutura do Menu:**
  - Exibe um menu principal com as opções:
    - 1 - Cadastrar um novo cliente
    - 2 - Listar todos os clientes
    - 3 - Buscar um cliente por ID
    - 4 - Atualizar informações de um cliente
    - 5 - Excluir um cliente
    - 0 - Sair do programa
  - Cada opção chama o método correspondente na ClienteADO para realizar a operação desejada.
- **Exemplo de Uso:**

```
Run (Main) × Java_E-commerce - C:\Users\luizp\OneDrive\Documentos\NetBeansProjects\Java_E-commerce ×  
  
--- Menu Principal ---  
1. Cadastrar Cliente  
2. Listar Clientes  
3. Buscar Cliente  
4. Atualizar Cliente  
5. Excluir Cliente  
0. Sair  
  
Escolha uma opcao:
```

- **Fluxo Básico**
  - Ao selecionar uma opção, o programa exibe instruções para coletar os dados necessários.
- Exemplo de cadastro:

```
t - Run (Main) x
--- Menu Principal ---
1. Cadastrar Cliente
2. Listar Clientes
3. Buscar Cliente
4. Atualizar Cliente
5. Excluir Cliente
0. Sair

Escolha uma opcao: 1
Nome: James Baxter
Email: jamesdabolinhadepraia@gmail.com
Telefone: 17986175657
Cliente cadastrado com sucesso!
```