**TÍTULO:** Projeto integrador Confeitaria. (Uc15)

**RESUMO:** sistema para cadastrar e armazenar dados dos clientes e pedidos diários de uma confeitaria em expansão.

**SOLID:** Exclui a classe DAO que tinha todas as funcionalidades, de todas as entidades.

- **S:** Crei novas classes DAOs com responsabilidade única, com funções para cada entidade

(ClienteDao, FuncionarioDao, PedidoDao, ProdutoDao, ItemPedidoDao).

- **O:** Pode ter novos métodos sem prejudicar ao alterar o código que já tem.

- **I:** Criei apenas métodos que estão sendo usados.

**ALTERAÇÕES/MODIFICAÇÕES:** Refatorei todas as classes, deixando as funções com nomes mais legíveis e fáceis de entender. Eliminei os comentários pelo código e os espaçamentos desnecessários.

**Evidências de refatoração:**

\*Alguns exemplos de métodos antes e depois de refatorar e separar em suas entidades específicas.

* **Cadastro de funcionários anteriormente(Classe DAO):**

// Cadastrar funcionários

public void adicionarFuncionario(Funcionario f){

String sql = "insert into funcionario(login, senha) values (?,?)";

try (Connection conn = Conexao.getConexao();

PreparedStatement ps = conn.prepareStatement(sql)){

ps.setString(1, f.getLogin());

ps.setString(2, f.getSenha());

ps.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

* **Classe funcionarioDao agora:**

**-**Melhorei a escrita deixando mais legível, removi os comentários do código e melhorando o tratamento de erro.

public boolean salvarFuncionario(Funcionario funcionario) {

String sql = "INSERT INTO funcionario(login, senha) VALUES (?, ?)";

try ( PreparedStatement ps = this.conn.prepareStatement(sql)) {

ps.setString(1, funcionario.getLogin());

ps.setString(2, funcionario.getSenha());

int linhaAfetada = ps.executeUpdate();

return linhaAfetada > 0;

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Erro ao cadastrar funcionário: " + e.getMessage(),

"Erro", JOptionPane.ERROR\_MESSAGE);

return false;

}

}

* **Cadastro de pedido anteriormente(Classe DAO):**

public void cadastrarPedido(Pedido pd){

String sql= "insert into pedido(nomeProduto,valor,nomeCliente,quantidade) values" + "(?,?,?,?)";

try{

PreparedStatement ps= this.conn.prepareStatement(sql);

ps.setString(1, pd.getNomeProduto());

ps.setDouble(2, pd.getValor());

ps.setString(3, pd.getNomeCliente());

ps.setInt(4, pd.getQuantidade());

ps.execute();

JOptionPane.showMessageDialog(null, "Pedido cadastrado com sucesso!");

}catch(Exception e){

System.out.println("Erro ao cadastrar pedido: " + e.getMessage());

}

}

public List<Pedido> getPedido(){

String sql= "SELECT \* FROM pedido";

List<Pedido> listaPedido= new ArrayList<>();

try{

PreparedStatement ps = this.conn.prepareStatement(sql);

ResultSet rs= ps.executeQuery();

while(rs.next()){

Pedido pd= new Pedido();

pd.setIdPedido(rs.getInt("idPedido"));

pd.setNomeProduto(rs.getString("nomeProduto"));

pd.setValor(rs.getDouble("valor"));

pd.setNomeCliente(rs.getString("nomeCliente"));

pd.setQuantidade(rs.getInt("quantidade"));

listaPedido.add(pd);

}

}catch(Exception e){

System.out.println("Erro: " + e.getMessage());

}

return listaPedido;

}

* **Classe pedidoDao agora:**
* Mantive os métodos de cadastrar e listar. E adicionei mais dois métodos, um de buscar por ID na tabela de listagem de pedidos e outro de excluir o pedido (botão).

public Pedido buscarPorId(int idPedido) {

String sql = "SELECT \* FROM pedido WHERE idPedido = ?";

try ( PreparedStatement ps = this.conn.prepareStatement(sql)) {

ps.setInt(1, idPedido);

try ( ResultSet rs = ps.executeQuery()) {

if (rs.next()) {

Pedido pedido = new Pedido();

pedido.setIdPedido(rs.getInt("idPedido"));

pedido.setNomeProduto(rs.getString("nomeProduto"));

pedido.setValor(rs.getDouble("valor"));

pedido.setNomeCliente(rs.getString("nomeCliente"));

pedido.setQuantidade(rs.getInt("quantidade"));

return pedido;

}

}

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Erro ao buscar pedido: " + e.getMessage(),

"Erro", JOptionPane.ERROR\_MESSAGE);

}

return null;

}

public boolean excluirPedido(int idPedido) {

String sql = "DELETE FROM pedido WHERE idPedido = ?";

try ( PreparedStatement ps = this.conn.prepareStatement(sql)) {

ps.setInt(1, idPedido);

int linhaAfetada = ps.executeUpdate();

return linhaAfetada > 0;

} catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Erro ao excluir pedido: " + e.getMessage(),

"Erro", JOptionPane.ERROR\_MESSAGE);

return false;

}

}

* **Evidências GitHub**

**- Link repositório:** https://github.com/NathanyCipriana/Confeitaria-1.1.git