

Universidade tecnológica Federal do Paraná - UTFPR

PAULO DE TACIO BASTOS PEREIRA

KAYLANY RAFAELA MARTINS DE PAULA

ALEC SANDER BITTENCOURT

Trabalho de Banco de Dados 2

Dois Vizinhos
2025

SUMÁRIO

1. INTRODUÇÃO	3
2. ESTRUTURA DO BANCO DE DADOS	4
2.1. Criação do Banco de Dados	4
3. INDEXAÇÃO	4
3.1. Criação de Índices	4
4. PROCESSAMENTO DE TRANSAÇÕES E CONTROLE DE CONCORRÊNCIA	5
4.1. Função de Simulação de Rollback	5
4.2. Transações Concorrentes	5
5. INTERFACE GRÁFICA	5
5.1. Telas:	5
6. BACKUP DO BANCO DE DADOS	7
7. CONSISTÊNCIA DE DADOS	8
8. SEGURANÇA	8
9. CONCLUSÃO	9
APÊNDICES	10
APÊNDICE A. Criação das Tabelas	10
A.1 Tabela de Fornecedores	10
A.2 Tabela de Produtos	10
A.3 Tabela de Funcionários	10
A.4 Tabela de Vendas	10
A.5 Tabela de Itens	11
APÊNDICE B. Criação da View e dos Índices	11
B.1 Criação da View	11
B.2 Criação do índice idx_fornecedores_descricao	11
B.3 Criação do índice dx_produtos_valor	11
APÊNDICE C. Criação dos roles e definição das permissões	11
C.1 Criação do user admin e suas permissões	11
C.2 Criação do role vendedor e suas permissões	11
C.3 Criação do role estoquista e suas permissões	12
C.4 Fornecendo permissão do admin sobre esses roles	12
APÊNDICE D. Criação das funções em PL/pgSQL	12

D.1 Função para simular ROLLBACK.....	12
D.2 Cadastro de fornecedores	13
D.3 Cadastro de funcionários	13
D.4 Cadastro de produtos	14
D.5 Remoção de produtos.....	16
D.6 Entrada de produtos	16
D.7 Realização de venda	17
APÊNDICE E. Batch files para backup.....	19
E.1 Backup automático.....	19
E.2 Backup manual.....	19
APÊNDICE F. Código-fonte da aplicação em Java	19
F.1 ConexaoBDPostgres.java	19
F.2 TelaAdmin.java	22
F.3 TelaAtualizarEstoque.java.....	28
F.4 TelaCadastroFornecedor.java.....	34
F.5 TelaCadastroProduto.java.....	37
F.6 TelaCadastroUsuario.java	42
F.7 TelaHistoricoVenda.java	49
F.8 TelaLogin.java.....	52
F.9 TelaRemocaoProduto.java	57
F.10 TelaVenda.java.....	61

Repositório do projeto:

<https://github.com/Kaylany-Rafaela/Banco-Dados>

1. INTRODUÇÃO

Este documento tem como objetivo descrever a estrutura e a implementação do banco de dados denominado "BancoTrabalho2", desenvolvido para a disciplina de Banco de Dados 2 do curso de Engenharia de Software da Universidade Tecnológica Federal do Paraná, Câmpus Dois Vizinhos. O banco de dados foi projetado para gerenciar informações relacionadas a fornecedores, produtos, funcionários, vendas e itens de vendas. Essa estrutura foi escolhida para garantir uma organização eficiente dos dados, facilitando a rastreabilidade e a integridade das informações. A relação entre essas entidades reflete a dinâmica de um sistema de controle de vendas, no qual fornecedores oferecem produtos, funcionários registram vendas e cada venda pode conter múltiplos itens, assegurando um modelo consistente e adequado para operações comerciais. O trabalho foi desenvolvido usando o ApacheNetBeans para a GUI e o PostgreSQL para o Banco de Dados. O GitHub foi usado como repositório, e todo o código fonte está disponível nesse link:

2. ESTRUTURA DO BANCO DE DADOS

2.1. Criação do Banco de Dados

O banco de dados foi criado utilizando o sistema de gerenciamento de banco de dados PostgreSQL, conforme a seguinte configuração:

```
CREATE DATABASE "BancoTrabalho2"  
WITH  
OWNER = postgres  
ENCODING = 'UTF8'  
LC_COLLATE = 'Portuguese_Brazil.1252'  
LC_CTYPE = 'Portuguese_Brazil.1252'  
TABLESPACE = pg_default  
CONNECTION LIMIT = -1;
```

As tabelas foram criadas de acordo com o Modelo Entidade Relacionamento disponível no Moodle.

3. INDEXAÇÃO

3.1. Criação de Índices

Para otimizar as consultas no banco de dados, foram criados índices com base em uma análise prévia dos padrões de acesso e do desempenho das consultas mais frequentes. Esses índices permitem uma busca mais eficiente por fornecedores e otimizam consultas financeiras, contribuindo para a melhoria do desempenho geral do sistema.

3.2. Índice de Buscas por Descrição de Produtos:

Facilita buscas frequentes por fornecedores pelo nome ou descrição.

```
CREATE INDEX idx_fornecedores_descricao ON tb_fornecedores (for_descricao);
```

3.3. Índice de Buscas para filtrar Produtos com uma faixa de Valor

Acelera consultas que filtram produtos por faixa de valor, como relatórios financeiros.

```
CREATE INDEX idx_produtos_valor ON tb_produtos (pro_valor);
```

4. PROCESSAMENTO DE TRANSAÇÕES E CONTROLE DE CONCORRÊNCIA

4.1. Função de Simulação de Rollback

```
CREATE OR REPLACE FUNCTION teste_rollback()
RETURNS void AS $$
BEGIN
    INSERT INTO tb_produtos (pro_codigo, pro_descricao, pro_valor, pro_quantidade,
tb_fornecedores_for_codigo)
    VALUES (999, 'Produto Inválido', 100.00, 10, 1);
    PERFORM 1 / 0; -- Simula erro
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'Erro encontrado. Rollback realizado.';
END;
$$ LANGUAGE plpgsql;
```

Essa função realiza uma inserção na tabela `tb_produtos`, e depois divide 1 por 0, simulando um erro. Quando a `EXCEPTION` é lançada, a transação sofre um rollback.

4.2. Transações Concorrentes

Para simular a realização de transações concorrentes, logamos um usuário no banco pelo `psql`. Primeiro desligamos o autocommit usando o comando `\SET AUTOCOMMIT OFF`, iniciamos a transação usando o `BEGIN` e chamamos a função `vender_produto`, que reserva as tabelas `tb_vendas` e `tb_produtos` para escrita. Ao chamar essa função novamente por outra interface `psql` ou pela aplicação Java, haverá uma trava até a primeira transação ser consolidada pelo `COMMIT`, garantindo a consistência dos dados.

5. INTERFACE GRÁFICA

5.1. Telas:

- TelaCadastroProduto

A classe “TelaCadastroProduto” é uma interface gráfica em Java, utilizando Swing, destinada ao cadastro de novos produtos no sistema. Esta tela permite que os usuários insiram

informações essenciais sobre um produto, como descrição, valor, quantidade e ID do fornecedor.

- Usabilidade:

Ao abrir a “TelaCadastroProduto”, o usuário verá um formulário onde pode inserir as informações do produto. Após preencher todos os campos, o usuário deve clicar no botão "Adicionar". Se todos os dados forem válidos, o produto será adicionado ao banco de dados, e uma mensagem de sucesso será exibida. Caso contrário, mensagens de erro apropriadas serão mostradas, orientando o usuário a corrigir os dados inseridos.

- TelaRemocaoProduto

A classe TelaRemocaoProduto é uma interface gráfica em Java, utilizando Swing, destinada à remoção de produtos do sistema. Esta tela permite que os usuários insiram o ID de um produto que desejam remover.

- TelaAtualizarEstoque

A classe TelaAtualizarEstoque é uma interface gráfica em Java, utilizando Swing, destinada à atualização do estoque de produtos no sistema. Esta tela permite que os usuários busquem um produto pelo seu ID e atualizem a quantidade disponível.

- TelaCadastroFornecedor

A classe TelaCadastroFornecedor é uma interface gráfica em Java, utilizando Swing, destinada ao cadastro de novos fornecedores no sistema. Esta tela permite que os usuários insiram informações essenciais sobre um fornecedor, como o nome.

- TelaCadastroUsuario

A classe TelaCadastroUsuario é uma parte fundamental do sistema, permitindo a inclusão de novos usuários de forma intuitiva e eficiente. A implementação atual garante que as informações sejam validadas antes de serem enviadas ao banco de dados, promovendo a integridade dos dados. No entanto, melhorias podem ser feitas, como a implementação de feedback visual mais detalhado e a adição de funcionalidades de segurança, como a criptografia de senhas.

- TelaLogin

A classe TelaLogin é uma parte fundamental do sistema, permitindo que os usuários se autenticuem antes de acessar as funcionalidades. A implementação atual garante que as credenciais sejam verificadas antes de permitir o acesso, promovendo a segurança do sistema. No entanto, melhorias podem ser feitas, como a implementação de feedback visual mais detalhado em caso de falhas de autenticação e a adição de funcionalidades de segurança, como a criptografia de senhas.

- TelaEstoque

A classe TelaEstoque é uma parte fundamental do sistema, permitindo a gestão eficiente do estoque de produtos. A implementação atual garante que os usuários possam buscar, adicionar, remover e atualizar produtos de forma intuitiva. No entanto, melhorias podem ser feitas, como a implementação de feedback visual mais detalhado e a adição de funcionalidades de segurança, como a confirmação de ações críticas (por exemplo, remoção de produtos).

- TelaAmdin

A classe TelaAdmin é uma parte fundamental do sistema, proporcionando uma interface centralizada para a gestão de funcionários e fornecedores. A implementação atual garante que os usuários possam visualizar e gerenciar informações de forma intuitiva e realizar backups manuais. No entanto, melhorias podem ser feitas, como a implementação de feedback visual mais detalhado e a adição de funcionalidades de segurança.

- TelaVenda

A classe TelaVenda é uma parte fundamental do sistema, permitindo a realização de vendas de forma intuitiva e eficiente. A implementação atual garante que os operadores possam adicionar produtos, verificar a quantidade disponível e concluir vendas de maneira organizada. No entanto, melhorias podem ser feitas, como a implementação de feedback visual mais detalhado e a adição de funcionalidades de segurança.

- ConexãoBDPostgres

A classe ConexaoBDPostgres é essencial para a interação entre a aplicação e o banco de dados PostgreSQL. Ela encapsula a lógica de conexão com o banco de dados, permitindo que outras partes do sistema se concentrem na lógica de negócios. A implementação atual é funcional, mas pode ser aprimorada com a adição de métodos para operações específicas de CRUD e melhorias na segurança.

6. BACKUP DO BANCO DE DADOS

O principal objetivo de criar um backup do banco de dados é garantir a segurança e a integridade dos dados. Entre as razões para sua realização, destacam-se a proteção contra perda de dados, a recuperação de desastres e a manutenção e atualização do sistema. Essa prática é essencial para o gerenciamento de dados, pois protege e preserva a integridade do banco. A implementação de backups regulares e sua automação são fundamentais para assegurar a recuperação eficiente das informações em caso de falhas, minimizando riscos e garantindo a continuidade do sistema.

Foi criada uma pasta chamada `backup_util`, onde guardamos cópia do executável `pg_dump`, suas bibliotecas e um batch file chamado `backup.bat` para a realização do backup. O batch file gera um backup em formato de diretório, chamado "backup". Outro batch file chamado `backup_manual.bat` gera um diretório chamado "backup_manual" após o comando do usuário pela aplicação Java. A automação do procedimento foi realizada por meio do Task Scheduler (Agendador de Tarefas) do Windows, ferramenta que permite a execução programada do batch file, eliminando a necessidade de acionamento manual e aumentando a confiabilidade do processo.

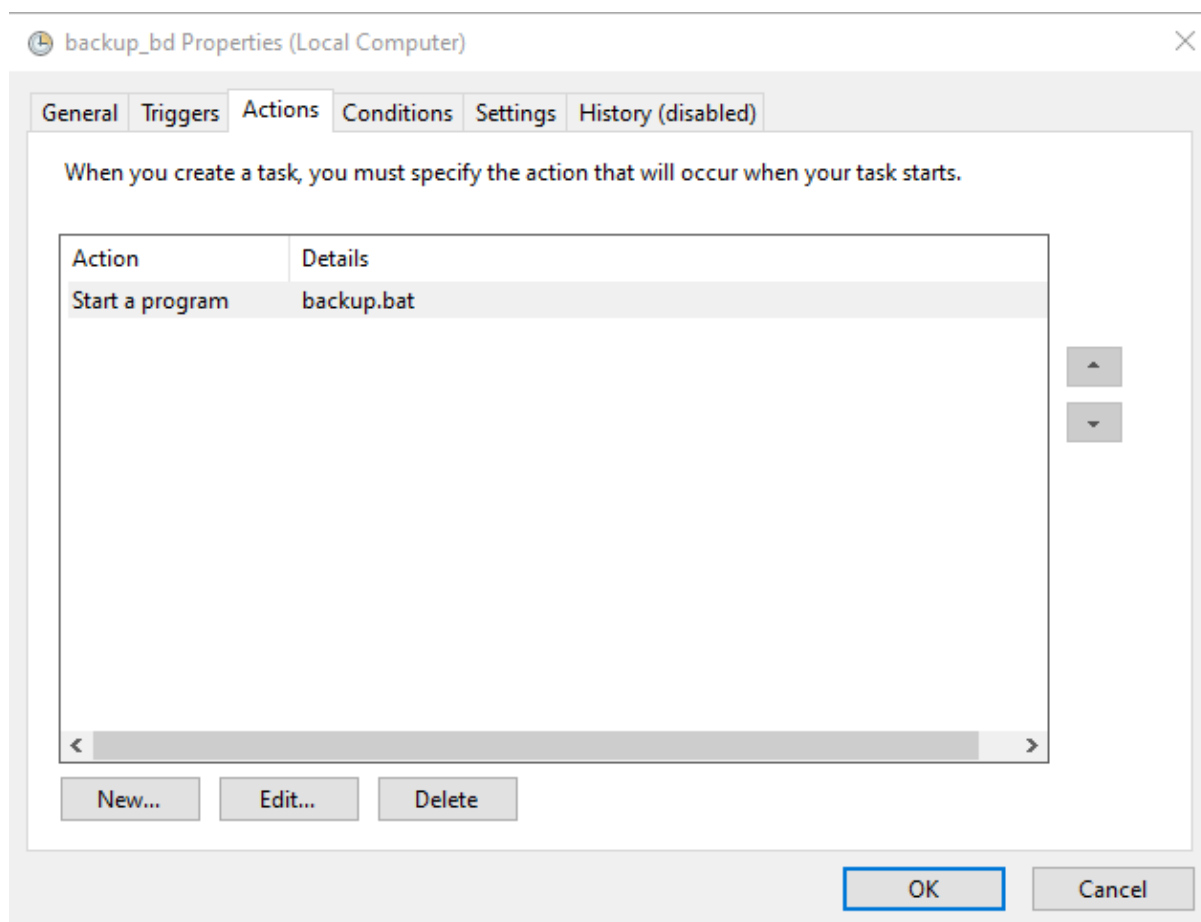


Figura: Ações da tarefa agendada backup_bd.

7. CONSISTÊNCIA DE DADOS

Todos os procedimentos do banco de dados verificam se as entradas são nulas, se são do tipo inválido ou se possuem valor inválido (ex: venda de -1 produtos); em todos os casos uma EXCEPTION é gerada e a transação sofre um ROLLBACK. Além disso, o programa java realiza as mesmas verificações antes iniciar qualquer transação no banco.

8. SEGURANÇA

Foi criada uma view chamada 'view_funcionarios_login', que oculta a senha dos funcionários mais ainda permite ver outras informações como nome e função.

Foram criados 3 roles:

'Admin', possui a permissão de criar outros usuários e dar a eles permissões.

'Vendedor' possui as permissões de visualizar a view de funcionários, a tabela de produtos e a tabela de vendas.

'Estoquista' possui apenas as permissões de visualizar a view de funcionários e a tabela de produtos.

Para modificar as tabelas, o Vendedor e Estoquista chamam funções do banco de dados que os concedem permissão temporária durante a transação através do SECURITY DEFINER, desde que as suas entradas foram válidas.

9. CONCLUSÃO

O banco de dados "BancoTrabalho2" foi estruturado para atender às necessidades de um sistema de vendas, garantindo normalização, integridade referencial e desempenho otimizado. O uso de chaves estrangeiras e índices aprimorou a eficiência das operações, e a documentação apresentada facilita manutenções futuras e expansões do sistema.

APÊNDICES

APÊNDICE A. Criação das Tabelas

A.1 Tabela de Fornecedores

```
CREATE TABLE tb_fornecedores (  
    for_codigo BIGINT PRIMARY KEY,  
    for_descricao VARCHAR(45) NOT NULL  
);
```

A.2 Tabela de Produtos

```
CREATE TABLE tb_produtos (  
    pro_codigo BIGINT PRIMARY KEY,  
    pro_descricao VARCHAR(45) NOT NULL,  
    pro_valor DECIMAL(10, 2) NOT NULL,  
    pro_quantidade INT NOT NULL,  
    tb_fornecedores_for_codigo BIGINT NOT NULL,  
    FOREIGN KEY (tb_fornecedores_for_codigo) REFERENCES tb_fornecedores  
    (for_codigo)  
);
```

A.3 Tabela de Funcionários

```
CREATE TABLE tb_funcionarios (  
    fun_codigo BIGINT PRIMARY KEY,  
    fun_nome VARCHAR(45) NOT NULL,  
    fun_cpf VARCHAR(45) NOT NULL UNIQUE,  
    fun_senha VARCHAR(50) NOT NULL,  
    fun_funcao VARCHAR(50) NOT NULL  
);
```

A.4 Tabela de Vendas

```
CREATE TABLE tb_vendas (  
    ven_codigo BIGINT PRIMARY KEY,  
    ven_horario TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,  
    ven_valor_total DECIMAL(10, 2) NOT NULL,  
    tb_funcionarios_fun_codigo BIGINT NOT NULL,  
    FOREIGN KEY (tb_funcionarios_fun_codigo) REFERENCES tb_funcionarios  
    (fun_codigo)  
);
```

A.5 Tabela de Itens

```
CREATE TABLE tb_itens (  
    ite_codigo BIGINT PRIMARY KEY,  
    ite_quantidade INT NOT NULL,  
    ite_valor_parcial DECIMAL(10, 2) NOT NULL,  
    tb_produtos_pro_codigo BIGINT NOT NULL,  
    tb_vendas_ven_codigo BIGINT NOT NULL,  
    FOREIGN KEY (tb_produtos_pro_codigo) REFERENCES tb_produtos (pro_codigo),  
    FOREIGN KEY (tb_vendas_ven_codigo) REFERENCES tb_vendas (ven_codigo)  
);
```

APÊNDICE B. Criação da View e dos Índices

B.1 Criação da View

```
CREATE OR REPLACE VIEW view_funcionarios_login  
AS SELECT fun_codigo, fun_nome, fun_cpf, fun_funcao FROM tb_funcionarios  
WITH CHECK OPTION;
```

B.2 Criação do índice idx_fornecedores_descricao

```
CREATE INDEX idx_fornecedores_descricao ON tb_fornecedores (for_descricao);
```

B.3 Criação do índice dx_produtos_valor

```
CREATE INDEX idx_produtos_valor ON tb_produtos (pro_valor);
```

APÊNDICE C. Criação dos roles e definição das permissões

C.1 Criação do user admin e suas permissões

```
DROP OWNED by admin;  
DROP user IF EXISTS admin;  
CREATE USER admin WITH PASSWORD 'admin' CREATEROLE;  
GRANT SELECT ON ALL TABLES IN SCHEMA public TO admin WITH GRANT OPTION;  
GRANT INSERT ON ALL TABLES IN SCHEMA public TO admin WITH GRANT OPTION;  
GRANT UPDATE ON ALL TABLES IN SCHEMA public TO admin WITH GRANT OPTION;  
GRANT DELETE ON ALL TABLES IN SCHEMA public TO admin WITH GRANT OPTION;
```

C.2 Criação do role vendedor e suas permissões

```
DROP OWNED BY vendedor;  
DROP ROLE IF EXISTS vendedor;  
CREATE ROLE vendedor;  
GRANT SELECT ON tb_produtos TO vendedor;
```

```
GRANT SELECT ON tb_vendas TO vendedor;  
GRANT SELECT ON view_funcionarios_login TO vendedor;
```

C.3 Criação do role estoquista e suas permissões

```
DROP OWNED BY estoquista;  
DROP ROLE IF EXISTS estoquista;  
CREATE ROLE estoquista;  
GRANT SELECT ON tb_produtos TO estoquista;  
GRANT SELECT ON view_funcionarios_login TO estoquista;
```

C.4 Fornecendo permissão do admin sobre esses roles

```
GRANT vendedor to admin WITH ADMIN OPTION;  
GRANT estoquista to admin WITH ADMIN OPTION;
```

APÊNDICE D. Criação das funções em PL/pgSQL

D.1 Função para simular ROLLBACK

```
CREATE OR REPLACE FUNCTION teste_rollback()  
RETURNS void AS $$  
BEGIN  
    -- Início da transação  
    INSERT INTO tb_produtos (pro_codigo, pro_descricao, pro_valor, pro_quantidade,  
tb_fornecedores_for_codigo)  
    VALUES (999, 'Produto Inválido', 100.00, 10, 1);  
  
    -- Erro proposital para testar o Rollback  
    PERFORM 1 / 0; -- Divisão por zero  
  
    -- Este código nunca será alcançado devido ao erro acima  
    COMMIT;  
EXCEPTION  
    WHEN OTHERS THEN  
        RAISE NOTICE 'Erro encontrado. Rollback realizado.';  
        -- O Rollback já foi executado automaticamente pelo PostgreSQL  
END;  
$$ LANGUAGE plpgsql;  
  
SELECT teste_rollback();
```

D.2 Cadastro de fornecedores

```
CREATE OR REPLACE PROCEDURE insert_tb_fornecedores(inputDescricao text)
AS $$
DECLARE
    codigo BIGINT;
    descricao VARCHAR(45);
BEGIN

    IF inputDescricao IS NULL THEN
        RAISE EXCEPTION 'Descricao nao pode ser nula';
    END IF;

    SELECT max(for_codigo) FROM tb_fornecedores INTO codigo;

    IF codigo IS NULL THEN
        codigo := 1;
    ELSE
        codigo := codigo + 1;
    END IF;

    BEGIN
        descricao := inputDescricao::VARCHAR(45);
    EXCEPTION WHEN others THEN
        RAISE EXCEPTION 'Descricao nao e um texto valido: %', inputDescricao;
    END;

    INSERT INTO tb_fornecedores VALUES (codigo, descricao);
END;
$$ LANGUAGE plpgsql;
```

D.3 Cadastro de funcionários

```
CREATE OR REPLACE PROCEDURE insert_tb_funcionarios(inputNome text, inputCPF
text, inputSenha text, inputFuncao text)
AS $$
DECLARE
    var_codigo BIGINT;
    var_nome VARCHAR(45);
    var_cpf VARCHAR(45);
    var_senha VARCHAR(50);
    var_funcao VARCHAR(50);
BEGIN
    IF inputNome IS NULL THEN
```

```
RAISE EXCEPTION 'Nome nao pode ser nulo';
ELSIF inputCPF IS NULL THEN
    RAISE EXCEPTION 'CPF nao pode ser nulo';
ELSIF inputSenha IS NULL THEN
    RAISE EXCEPTION 'Senha nao pode ser nula';
ELSIF inputFuncao IS NULL THEN
    RAISE EXCEPTION 'Funcao nao pode ser nula';
END IF;

IF inputFuncao != 'Vendedor' AND inputFuncao != 'Estoquista' AND inputFuncao !=
'vendedor' AND inputFuncao != 'estoquista' THEN
    RAISE EXCEPTION 'Nome invalido para Funcao';
END IF;

SELECT max(fun_codigo) FROM tb_funcionarios INTO var_codigo;

IF var_codigo IS NULL THEN
    var_codigo := 1;
ELSE
    var_codigo := var_codigo + 1;
END IF;

var_nome = inputNome;
var_cpf = inputCPF;
var_senha = inputSenha;
var_funcao = inputFuncao;

INSERT INTO tb_funcionarios VALUES (var_codigo, var_nome, var_cpf, var_senha,
var_funcao);
END;
$$ LANGUAGE plpgsql;
```

D.4 Cadastro de produtos

```
CREATE OR REPLACE PROCEDURE insert_tb_produtos(inputDescricao text, inputValor
text, inputQuantidade text, inputIDFornecedor text)
AS $$
DECLARE
    var_codigo BIGINT;
    var_descricao VARCHAR(45);
    var_valor DECIMAL(10, 2);
    var_quantidade INT;
    var_fornecedores_codigo BIGINT;
```

```
BEGIN
    IF inputDescricao IS NULL THEN
        RAISE EXCEPTION 'Descricao nao pode ser nula';
    ELSIF inputValor IS NULL THEN
        RAISE EXCEPTION 'Valor nao pode ser nulo';
    ELSIF inputQuantidade IS NULL THEN
        RAISE EXCEPTION 'Quantidade nao pode ser nula';
    ELSIF inputIDFornecedor IS NULL THEN
        RAISE EXCEPTION 'ID do Fornecedor nao pode ser nulo';
    END IF;

    BEGIN
        var_descricao := inputDescricao::VARCHAR(45);
    EXCEPTION WHEN others THEN
        RAISE EXCEPTION 'Descricao nao e um texto valido: %', inputDescricao;
    END;

    BEGIN
        var_valor := inputValor::DECIMAL(10, 2);
    EXCEPTION WHEN others THEN
        RAISE EXCEPTION 'Valor nao e um numero valido: %', inputValor;
    END;

    BEGIN
        var_quantidade := inputQuantidade::INT;
    EXCEPTION WHEN others THEN
        RAISE EXCEPTION 'Quantidade nao e um numero valido: %', inputQuantidade;
    END;

    BEGIN
        var_fornecedores_codigo := inputIDFornecedor::BIGINT;
    EXCEPTION WHEN others THEN
        RAISE EXCEPTION 'ID Fornecedor nao e um numero valido: %', inputIDFornecedor;
    END;

    IF(var_valor <= 0) THEN
        RAISE EXCEPTION 'Valor do produto nao pode ser menor ou igual a zero!';
    ELSIF (var_quantidade < 0) THEN
        RAISE EXCEPTION 'Quantidade do produto nao pode ser menor que zero!';
    END IF;

    SELECT max(pro_codigo) FROM tb_produtos INTO var_codigo;
```

```
IF var_codigo IS NULL THEN
    var_codigo := 1;
ELSE
    var_codigo := var_codigo + 1;
END IF;
```

```
INSERT INTO tb_produtos VALUES (var_codigo, var_descricao, var_valor,
var_quantidade, var_fornecedores_codigo);
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;
```

D.5 Remoção de produtos

```
CREATE OR REPLACE PROCEDURE remove_tb_produtos(inputCodigo text)
AS $$
DECLARE
    var_codigo BIGINT;
BEGIN
    IF inputCodigo IS NULL THEN
        RAISE EXCEPTION 'Codigo nao pode ser nulo';
    END IF;
    BEGIN
        var_codigo := inputCodigo::BIGINT;
    EXCEPTION WHEN others THEN
        RAISE EXCEPTION 'ID do Produto nao e um numero valido: %', inputCodigo;
    END;

    DELETE FROM tb_produtos WHERE pro_codigo = var_codigo;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;
```

D.6 Entrada de produtos

```
CREATE OR REPLACE PROCEDURE entrada_produto(inputCodigo text, inputQuantidade
text)
AS $$
DECLARE
    var_codigo BIGINT;
    var_quantidade INT;
BEGIN
    IF inputCodigo IS NULL THEN
        RAISE EXCEPTION 'Codigo nao pode ser nulo';
    ELSIF inputQuantidade IS NULL THEN
```



```
RAISE EXCEPTION 'Quantidade nao pode ser nula';  
END IF;
```

```
BEGIN  
    var_codigo := inputCodigo::BIGINT;  
EXCEPTION WHEN others THEN  
    RAISE EXCEPTION 'ID do Produto nao e um numero valido: %', inputCodigo;  
END;
```

```
BEGIN  
    var_quantidade := inputQuantidade::INT;  
EXCEPTION WHEN others THEN  
    RAISE EXCEPTION 'Quantidade do Produto nao e um numero valido: %',  
inputQuantidade;  
END;
```

```
UPDATE tb_produtos SET pro_quantidade = pro_quantidade + var_quantidade WHERE  
pro_codigo = var_codigo;  
END;  
$$ LANGUAGE plpgsql SECURITY DEFINER;
```

D.7 Realização de venda

```
CREATE OR REPLACE PROCEDURE vender_produto(inputCodigo text, inputDescricao  
text, inputQuantidade text, inputValorTotal text, inputCPFVendedor text)
```

```
AS $$
```

```
DECLARE
```

```
    var_codigo BIGINT;  
    var_quantidadeVenda INT;  
    var_quantidadeEstoque INT;  
    var_valorTotal DECIMAL(10, 2);  
    var_codVendedor BIGINT;  
    var_codVenda BIGINT;
```

```
BEGIN
```

```
-- Null check pra todas as entradas  
    IF inputCodigo IS NULL THEN  
        RAISE EXCEPTION 'Codigo nao pode ser nulo';  
    ELSIF inputQuantidade IS NULL THEN  
        RAISE EXCEPTION 'Quantidade nao pode ser nula';  
    ELSIF inputValorTotal IS NULL THEN  
        RAISE EXCEPTION 'Valor total nao pode ser nulo';  
    ELSEIF inputCPFVendedor IS NULL THEN  
        RAISE EXCEPTION 'Nome do vendedor nao pode ser nulo';  
    END IF;
```

```
-- Type check pra todas as entradas
BEGIN
    var_codigo := inputCodigo::BIGINT;
EXCEPTION WHEN others THEN
    RAISE EXCEPTION 'ID do Produto nao e um numero valido: %', inputCodigo;
END;

BEGIN
    var_quantidadeVenda := inputQuantidade::INT;
EXCEPTION WHEN others THEN
    RAISE EXCEPTION 'Quantidade do Produto nao e um numero valido: %',
inputQuantidade;
END;
BEGIN
    var_valorTotal := inputValorTotal::DECIMAL(10, 2);
EXCEPTION WHEN others THEN
    RAISE EXCEPTION 'Valor total da venda nao e um numero valido: %', inputValorTotal;
END;

IF (var_quantidadeVenda <= 0) THEN
    RAISE EXCEPTION 'Quantidade de produtos vendidos não pode ser menor ou igual a
zero!';
ELSIF(var_valorTotal <= 0) THEN
    RAISE EXCEPTION 'Valor total da venda não pode ser menor ou igual a zero!';
END IF;

-- Check se há suficiente no estoque
SELECT pro_quantidade FROM tb_produtos WHERE pro_codigo = var_codigo INTO
var_quantidadeEstoque;
IF (var_quantidadeEstoque - var_quantidadeVenda) < 0 THEN
    RAISE EXCEPTION 'Quantidade do produto % indisponivel no estoque',
inputDescricao;
END IF;

-- Atualizar a quantidade do produto
UPDATE tb_produtos SET pro_quantidade = pro_quantidade - var_quantidadeVenda
WHERE pro_codigo = var_codigo;

-- Registrar a venda na tabela

-- Atribui o codigo do funcionario
```

```
SELECT fun_codigo from view_funcionarios_login WHERE fun_cpf = inputCPFVendedor
INTO var_codVendedor;
SELECT max(ven_codigo) FROM tb_vendas INTO var_codVenda;
IF var_codVenda IS NULL THEN
    var_codVenda := 1;
ELSE
    var_codVenda := var_codVenda + 1;
END IF;
INSERT INTO tb_vendas(ven_codigo, ven_valor_total, tb_funcionarios_fun_codigo)
VALUES (var_codVenda, var_valorTotal, var_codVendedor);
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;
```

APÊNDICE E. Batch files para backup

E.1 Backup automático

: : Gera uma pasta chamada "backup", pode ser restaurada pelo pgAdmin como DIRECTORY.

: : Automatize com o Task Scheduler do Windows.

set PGPASSWORD=postgres

pg_dump -Fd --username=postgres --dbname=BancoTrabalho2 -f backup

E.2 Backup manual

: : Gera uma pasta chamada "backup_manual", pode ser restaurada pelo pgAdmin como DIRECTORY.

cd backup_util

set PGPASSWORD=postgres

pg_dump -Fd --username=postgres --dbname=BancoTrabalho2 -f backup_manual

APÊNDICE F. Código-fonte da aplicação em Java

F.1 ConexaoBDPostgres.java

```
package br.com.sistemas.model.database;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import javax.swing.JOptionPane;
```

```
/**
 * Classe responsável por gerenciar a conexão com o banco de dados PostgreSQL.
 */
public class ConexaoBDPostgres {
    private Connection conexao;
    private static final String DRIVER = "org.postgresql.Driver";
    private static final String HOST = "localhost";
    private static final int PORTA = 5432;

    protected String cpf, senha, nomeBanco;

    public ConexaoBDPostgres(String cpf, String senha, String nomeBanco){
        this.cpf = cpf;
        this.senha = senha;
        this.nomeBanco = nomeBanco;
    }

    public boolean conectar() {

        String url = "jdbc:postgresql://" + HOST + ":" + PORTA + "/" + nomeBanco;

        try {
            Class.forName(DRIVER);
            // Conexao sem prefixo para admins
            if(cpf.equals("postgres") || cpf.equals("admin")){
                conexao = DriverManager.getConnection(url, cpf, senha);
            } else{
                conexao = DriverManager.getConnection(url, "u" + cpf + "u", senha);
            }

            return true;
        } catch (ClassNotFoundException e) {
            JOptionPane.showMessageDialog(null, "Erro: Driver JDBC não encontrado!", "Erro de Conexão", JOptionPane.ERROR_MESSAGE);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Erro ao conectar ao banco: " + e.getMessage(), "Erro de Conexão", JOptionPane.ERROR_MESSAGE);
        }

        return false;
    }
}
```

/**

```
* Obtém a conexão atual com o banco de dados.
*
* @return Objeto Connection representando a conexão ativa.
*/
public Connection getConexao() {
    return conexao;
}

public String getCPF(){
    return this.cpf;
}

// Retorna a funcao do usuário com o cpf informado
public String getCargo(String cpf){
    String cargo = "nulo";
    String sql = "SELECT fun_funcao FROM view_funcionarios_login WHERE fun_cpf = "
+ cpf + " ";
    try
    {
        PreparedStatement ps = conexao.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            cargo = rs.getString("fun_funcao");
        }
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
    return cargo;
}

/**
* Método responsável por fechar a conexão com o banco de dados.
*/
public void disconnect() {
    if (conexao != null) {
        try {
            conexao.close();
            System.out.println("Conexão fechada com sucesso.");
        } catch (SQLException erro) {
            System.out.println("Erro ao fechar a conexão: " + erro.getMessage());
        }
    }
}
}
```

F.2 TelaAdmin.java

```
package br.com.sistemas.userInterface;

import br.com.sistemas.model.database.ConexaoBDPostgres;
import java.io.File;
import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

//registro de teste 123321
/**
 *
 * @author Bastos
 */
public class TelaAdmin extends javax.swing.JFrame {

    /**
     * Creates new form TelaHome
     */
    ConexaoBDPostgres conexao;
    DefaultTableModel modeloFuncionario = new DefaultTableModel(new Object [][] { },
        new String [] { "ID Funcionario", "Nome", "CPF", "Funcao" });

    DefaultTableModel modeloFornecedor = new DefaultTableModel(new Object [][] { },
        new String [] { "ID Fornecedor", "Nome" });

    public TelaAdmin(ConexaoBDPostgres conexao) {
        this.conexao = conexao;
        initComponents();
        carregarFuncionarios();
        carregarFornecedores();
    }

    public final void carregarFuncionarios() {
        String sql = "SELECT * FROM tb_funcionarios;";
        try (PreparedStatement ps = conexao.getConexao().prepareStatement(sql);
            ResultSet rs = ps.executeQuery()) {
```

```

        while(modeloFuncionario.getRowCount() > 0) {
            modeloFuncionario.removeRow(0);
        }
        while (rs.next()) {
            Vector linha = new Vector();
            linha.add(rs.getInt("fun_codigo"));
            linha.add(rs.getString("fun_nome"));
            linha.add(rs.getString("fun_cpf"));
            linha.add(rs.getString("fun_funcao"));
            modeloFuncionario.addRow(linha);
        }
        ps.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

public final void carregarFornecedores() {
    String sql = "SELECT * FROM tb_fornecedores;";
    try (PreparedStatement ps = conexao.getConexao().prepareStatement(sql);
        ResultSet rs = ps.executeQuery()) {
        while(modeloFornecedor.getRowCount() > 0) {
            modeloFornecedor.removeRow(0);
        }
        while (rs.next()) {
            Vector linha = new Vector();
            linha.add(rs.getInt("for_codigo"));
            linha.add(rs.getString("for_descricao"));
            modeloFornecedor.addRow(linha);
        }
        ps.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">

```

```
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jScrollPane2 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jScrollPane1 = new javax.swing.JScrollPane();
    tabelaMostrarFuncionarios = new javax.swing.JTable();
    jButtonCadastrarFuncionarios = new javax.swing.JButton();
    jButtonCadastrarFornecedores = new javax.swing.JButton();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu1 = new javax.swing.JMenu();
    jMenuItemBackup = new javax.swing.JMenuItem();
    jMenuItemRefresh = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Administração");

    jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));

    jTable1.setModel(modeloFornecedor);
    jTable1.setEnabled(false);
    jScrollPane2.setViewportView(jTable1);

    tabelaMostrarFuncionarios.setModel(modeloFuncionario);
    tabelaMostrarFuncionarios.setEnabled(false);
    tabelaMostrarFuncionarios.setFocusable(false);
    tabelaMostrarFuncionarios.setRowSelectionAllowed(false);
    jScrollPane1.setViewportView(tabelaMostrarFuncionarios);

    jButtonCadastrarFuncionarios.setFont(new java.awt.Font("Segoe UI", 1, 14)); //
NOI18N
    jButtonCadastrarFuncionarios.setText("Cadastrar Funcionários");
    jButtonCadastrarFuncionarios.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonCadastrarFuncionariosActionPerformed(evt);
        }
    });

    jButtonCadastrarFornecedores.setFont(new java.awt.Font("Segoe UI", 1, 14)); //
NOI18N
    jButtonCadastrarFornecedores.setText("Cadastrar Fornecedores");
```



```

jButtonCadastrarFornecedores.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCadastrarFornecedoresActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
497, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButtonCadastrarFuncionarios)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        )
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
288, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButtonCadastrarFornecedores)
            .addContainerGap(15, Short.MAX_VALUE)
        )
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 524,
Short.MAX_VALUE)
            .addComponent(jScrollPane2)
        )
);

```

```
.addContainerGap()
);

jMenu1.setText("Sistema");

jMenuItemBackup.setText("Backup");
jMenuItemBackup.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItemBackupActionPerformed(evt);
    }
});
jMenu1.add(jMenuItemBackup);

jMenuItemRefresh.setText("Refresh");
jMenuItemRefresh.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItemRefreshActionPerformed(evt);
    }
});
jMenu1.add(jMenuItemRefresh);

jMenuBar1.add(jMenu1);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap()
            )
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            )
        );
```

```
        pack();
        setLocationRelativeTo(null);
    } // </editor-fold>

    private void jMenuItemRefreshActionPerformed(java.awt.event.ActionEvent evt) {
        carregarFuncionarios();
        carregarFornecedores();
    }

    private void jMenuItemBackupActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            File backup = new File("backup_manual.bat");
            if(backup.isFile()){
                Runtime.getRuntime().exec(new String[] { "cmd.exe", "/c", "backup_manual.bat" }
);
                JOptionPane.showMessageDialog(this, "Realizando backup!");
            } else{
                JOptionPane.showMessageDialog(this, "Erro! Batch file para backup não
encontrado!");
            }
        } catch (IOException e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }

    private void jButtonCadastrarFuncionariosActionPerformed(java.awt.event.ActionEvent
evt) {
        TelaCadastroUsuario telaCadastroUsuario = new TelaCadastroUsuario(conexao);
        telaCadastroUsuario.setVisible(true);
    }

    private void jButtonCadastrarFornecedoresActionPerformed(java.awt.event.ActionEvent
evt) {
        TelaCadastroFornecedor telaCadastroFornecedor = new
TelaCadastroFornecedor(conexao);
        telaCadastroFornecedor.setVisible(true);
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButtonCadastrarFornecedores;
    private javax.swing.JButton jButtonCadastrarFuncionarios;
    private javax.swing.JMenu jMenuItem1;
    private javax.swing.JMenuBar jMenuItemBar1;
```

```
private javax.swing.JMenuItem jMenuItemBackup;
private javax.swing.JMenuItem jMenuItemRefresh;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTable1;
private javax.swing.JTable tabelaMostrarFuncionarios;
// End of variables declaration
}
```

F.3 TelaAtualizarEstoque.java

```
package br.com.sistemas.userInterface;
```

```
import br.com.sistemas.model.database.ConexaoBDPostgres;
import java.awt.Color;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

```
/**
```

```
*
```

```
* @author Bastos
```

```
*/
```

```
public class TelaAtualizarEstoque extends javax.swing.JFrame {
```

```
    ConexaoBDPostgres conexao;
```

```
    DefaultTableModel modeloTabelaBuscaEstoque = new DefaultTableModel(new Object
[] { },
        new String [] { "Código", "Descrição", "Quantidade" });
```

```
    public TelaAtualizarEstoque(ConexaoBDPostgres conexao) {
        this.conexao = conexao;
        initComponents();
    }
```

```
    public final void carregarTabelaBuscaEstoque() {
        String busca = jTextFieldBuscar.getText();
        if(busca.isBlank()){
            jLabelMensagemErroCodigo.setForeground(Color.red);
        }
```

```

    } else{
        jLabelMensagemErroCodigo.setForeground(new Color(214, 217, 223));
        String sql = "SELECT * FROM tb_produtos WHERE pro_codigo = " + busca + ";";
        try (PreparedStatement ps = conexao.getConexao().prepareStatement(sql);
            ResultSet rs = ps.executeQuery()) {
            while(modeloTabelaBuscaEstoque.getRowCount() > 0) {
                modeloTabelaBuscaEstoque.removeRow(0);
            }
            while (rs.next()) {
                Vector linha = new Vector();
                linha.add(rs.getInt("pro_codigo"));
                linha.add(rs.getString("pro_descricao"));
                linha.add(rs.getInt("pro_quantidade"));
                modeloTabelaBuscaEstoque.addRow(linha);
            }
            ps.close();
        } catch(SQLException e){
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }
}

public void entradaDeProduto(){
    int quantidadeEntrada = (int)jSpinnerSeletorQuantidadeAdd.getValue();
    String sql = "CALL entrada_produto(" + (int)modeloTabelaBuscaEstoque.getValueAt(0,
0) + ", " + quantidadeEntrada + "));";
    try{
        PreparedStatement ps = conexao.getConexao().prepareStatement(sql);
        ps.executeUpdate();
        ps.close();
        JOptionPane.showMessageDialog(this, "Atualizado com sucesso!");
    } catch (SQLException e){
        JOptionPane.showMessageDialog(this, "Erro" + e.getMessage());
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">

```

```
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jButtonAdicionar = new javax.swing.JButton();
    jTextFieldBuscar = new javax.swing.JTextField();
    jButtonBuscar = new javax.swing.JButton();
    jLabelMensagemErroCodigo = new javax.swing.JLabel();
    jSpinnerSeletorQuantidadeAdd = new javax.swing.JSpinner();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("Atualização de Estoque");

    jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));

    jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
    jLabel1.setText("ID do Produto");

    jTable1.setModel(modeloTabelaBuscaEstoque);
    jTable1.setEnabled(false);
    jScrollPane1.setViewportView(jTable1);

    jButtonAdicionar.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    jButtonAdicionar.setText("Adicionar");
    jButtonAdicionar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonAdicionarActionPerformed(evt);
        }
    });

    jTextFieldBuscar.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N

    jButtonBuscar.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    jButtonBuscar.setText("Buscar");
    jButtonBuscar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonBuscarActionPerformed(evt);
        }
    });
}
```

```
jLabelMensagemErroCodigo.setForeground(new java.awt.Color(214, 217, 223));
jLabelMensagemErroCodigo.setText("Insira o ID do Produto");

jLabelMensagemErroCodigo.setHorizontalTextPosition(javax.swing.SwingConstants.LEADING);

jSpinnerSeletorQuantidadeAdd.setFont(new java.awt.Font("Segoe UI", 1, 14)); //
NOI18N
jSpinnerSeletorQuantidadeAdd.addChangeListener(new
javax.swing.event.ChangeListener() {
    public void stateChanged(javax.swing.event.ChangeEvent evt) {
        jSpinnerSeletorQuantidadeAddStateChanged(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(jSpinnerSeletorQuantidadeAdd)
                .add(jButtonAdicionar)
            )
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .add(jTextFieldBuscar)
        )
        .add(jButtonBuscar)
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .add(jSpinnerSeletorQuantidadeAdd)
            .add(jButtonAdicionar)
            .add(jTextFieldBuscar)
            .add(jButtonBuscar)
        )
);
```

```
.addComponent(jLabel1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jLabelMensagemErroCodigo,
javax.swing.GroupLayout.PREFERRED_SIZE, 148,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
242, javax.swing.GroupLayout.PREFERRED_SIZE))
.addContainerGap()
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel1)
.addComponent(jLabelMensagemErroCodigo))
.addGap(7, 7, 7)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jTextFieldBuscar,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButtonBuscar))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jSpinnerSeletorQuantidadeAdd,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButtonAdicionar))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 86,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap()
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
```



```
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE))
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(10, Short.MAX_VALUE))
        );

pack();
setLocationRelativeTo(null);
} // </editor-fold>

private void jButtonAdicionarActionPerformed(java.awt.event.ActionEvent evt) {
    if(modeloTabelaBuscaEstoque.getRowCount() == 0){
        jLabelMensagemErroCodigo.setForeground(Color.red);
    } else{
        entradaDeProduto();
        carregarTabelaBuscaEstoque();
    }
}

private void
jSpinnerSeletorQuantidadeAddStateChanged(javax.swing.event.ChangeEvent evt) {
    if(((int)jSpinnerSeletorQuantidadeAdd.getValue() < 0){
        jSpinnerSeletorQuantidadeAdd.setValue(0);
    }
}

private void jButtonBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    carregarTabelaBuscaEstoque();
}
```

```
// Variables declaration - do not modify
private javax.swing.JButton jButtonAdicionar;
private javax.swing.JButton jButtonBuscar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabelMensagemErroCodigo;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JSpinner jSpinnerSeletorQuantidadeAdd;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextFieldBuscar;
// End of variables declaration
}
```

F.4 TelaCadastroFornecedor.java

```
package br.com.sistemas.userInterface;

import br.com.sistemas.model.database.ConexaoBDPostgres;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;

/**
 *
 * @author Bastos
 */
public class TelaCadastroFornecedor extends javax.swing.JFrame {

    ConexaoBDPostgres conexao;

    public TelaCadastroFornecedor(ConexaoBDPostgres conexao) {

        this.conexao = conexao;
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
```

```
private void initComponents() {

    jButtonOk = new javax.swing.JButton();
    jPanel1 = new javax.swing.JPanel();
    jLabelNomeFornecedor = new javax.swing.JLabel();
    jTextFieldDescricaoFornecedor = new javax.swing.JTextField();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("Cadastro de Fornecedor");

    jButtonOk.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
    jButtonOk.setText("Cadastrar Fornecedor");
    jButtonOk.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonOkActionPerformed(evt);
        }
    });

    jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));

    jLabelNomeFornecedor.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    jLabelNomeFornecedor.setText("Nome Fornecedor:");

    jLabelNomeFornecedor.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

    jTextFieldDescricaoFornecedor.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabelNomeFornecedor, javax.swing.GroupLayout.DEFAULT_SIZE, 200, true)
                    .addComponent(jTextFieldDescricaoFornecedor, javax.swing.GroupLayout.DEFAULT_SIZE, 200, true))
                .addContainerGap(10, Short.MAX_VALUE))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(10, 10, 0)
                .addComponent(jButtonOk, javax.swing.GroupLayout.DEFAULT_SIZE, 100, true)
                .addContainerGap())
    );
}
```

```
.addGap(0, 0, Short.MAX_VALUE))
.addComponent(jLabelNomeFornecedor,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
.addContainerGap()
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addContainerGap()
.addComponent(jLabelNomeFornecedor)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jTextFieldDescricaoFornecedor,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap()
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(jButtonOk, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addContainerGap()
.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jButtonOk)
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
```

```
);

pack();
setLocationRelativeTo(null);
} // </editor-fold>

private void jButtonOkActionPerformed(java.awt.event.ActionEvent evt) {
    String fornecedor = jTextFieldDescricaoFornecedor.getText();
    try {
        PreparedStatement ps = null;
        String sql = null;

        sql = "CALL insert_tb_fornecedores('" + fornecedor + "');";
        ps = conexao.getConnection().prepareStatement(sql);
        ps.executeUpdate();

        // Encerra a transação e mostra mensagem de sucesso
        ps.close();
        JOptionPane.showMessageDialog(this, "Registrado fornecedor " + fornecedor + "
com sucesso!");
        dispose(); // Auto-fecha janela de cadastro;
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

// Variables declaration - do not modify
private javax.swing.JButton jButtonOk;
private javax.swing.JLabel jLabelNomeFornecedor;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextFieldDescricaoFornecedor;
// End of variables declaration
}
```

F.5 TelaCadastroProduto.java

```
package br.com.sistemas.userInterface;

import br.com.sistemas.model.database.ConexaoBDPostgres;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
```

```
import javax.swing.JOptionPane;

/**
 *
 * @author Bastos
 */

public class TelaCadastroProduto extends javax.swing.JFrame {

    ConexaoBDPostgres conexaoBD;
    public TelaCadastroProduto(ConexaoBDPostgres conexaoBD) {
        this.conexaoBD = conexaoBD;
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        buttonAdicionar = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jLabelFornecedor = new javax.swing.JLabel();
        jLabelPreco = new javax.swing.JLabel();
        jTextFieldDescricaoProduto = new javax.swing.JTextField();
        jTextFieldValorProduto = new javax.swing.JTextField();
        jTextFieldQuantidadeProduto = new javax.swing.JTextField();
        jLabelQuantidade = new javax.swing.JLabel();
        jLabelDescricao = new javax.swing.JLabel();
        jTextFieldIdFornecedorProduto = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Cadastro de Produtos");
        setResizable(false);

        buttonAdicionar.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
        buttonAdicionar.setText("Cadastrar Produto");
        buttonAdicionar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {

```



```
.addComponent(jTextFieldDescricaoProduto)
.addComponent(jTextFieldValorProduto)
.addComponent(jTextFieldQuantidadeProduto,
javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addContainerGap()
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addContainerGap()
.addComponent(jLabelQuantidade)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jTextFieldQuantidadeProduto,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jLabelPreco)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jTextFieldValorProduto,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jLabelDescricao)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jTextFieldDescricaoProduto,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jLabelFornecedor)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jTextFieldIdFornecedorProduto,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addContainerGap()
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addContainerGap()
```



```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(buttonAdicionar, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(buttonAdicionar)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

pack();
setLocationRelativeTo(null);
} // </editor-fold>
```

```
private void buttonAdicionarActionPerformed(java.awt.event.ActionEvent evt) {

    Connection conexao = conexaoBD.getConexao();

    // Pegando os valores dos campos de texto
    String descricaoProduto = jTextFieldDescricaoProduto.getText();
    double valorProduto;
    int quantidadeProduto;
    int idFornecedorProduto;

    // Verificação de tipo das entradas
    try {
        valorProduto = Double.parseDouble(jTextFieldValorProduto.getText());
        quantidadeProduto = Integer.parseInt(jTextFieldQuantidadeProduto.getText());
        idFornecedorProduto = Integer.parseInt(jTextFieldIdFornecedorProduto.getText());
    } catch (NumberFormatException e) {
```

```
JOptionPane.showMessageDialog(this, "Erro: Certifique-se de que o preço e a
quantidade são números válidos.");
    return;
}

String sql = "CALL insert_tb_produtos('" + descricaoProduto + "', '" + valorProduto + "',
"
    + quantidadeProduto + "', '" + idFornecedorProduto + "')";

try (PreparedStatement ps = conexao.prepareStatement(sql)) {
    ps.executeUpdate();
    JOptionPane.showMessageDialog(this, "✓ Produto " + descricaoProduto + "
inserido com sucesso!");
    ps.close();
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, e.getMessage());
}
}

// Variables declaration - do not modify
private javax.swing.JButton buttonAdicionar;
private javax.swing.JLabel jLabelDescricao;
private javax.swing.JLabel jLabelFornecedor;
private javax.swing.JLabel jLabelPreco;
private javax.swing.JLabel jLabelQuantidade;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextFieldDescricaoProduto;
private javax.swing.JTextField jTextFieldIdFornecedorProduto;
private javax.swing.JTextField jTextFieldQuantidadeProduto;
private javax.swing.JTextField jTextFieldValorProduto;
// End of variables declaration
}
```

F.6 TelaCadastroUsuario.java

```
package br.com.sistemas.userInterface;

import javax.swing.JOptionPane;
import br.com.sistemas.model.database.ConexaoBDPostgres;
import java.sql.PreparedStatement;
import java.sql.SQLException;

/**
```

```
*
* @author Larissa
*/
public class TelaCadastroUsuario extends javax.swing.JFrame {

    /**
     * Creates new form TelaLogin
     */

    ConexaoBDPostgres conexao;

    public TelaCadastroUsuario(ConexaoBDPostgres conexao) {
        this.conexao = conexao;
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButtonCadastrar = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jLabelNome = new javax.swing.JLabel();
        jTextFieldUsername = new javax.swing.JTextField();
        jLabelSenha = new javax.swing.JLabel();
        jPasswordFieldSenha = new javax.swing.JPasswordField();
        jLabelCPF = new javax.swing.JLabel();
        jTextFieldCPF = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        jComboBoxFuncao = new javax.swing.JComboBox<>();
        jLabelErroCPF = new javax.swing.JLabel();
        jLabelErroNome = new javax.swing.JLabel();
        jLabelErroSenha = new javax.swing.JLabel();
        jLabelError = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Cadastro de Usuario");
        setBackground(new java.awt.Color(255, 255, 255));
    }
}
```

```
setMaximumSize(new java.awt.Dimension(652, 440));  
setMinimumSize(new java.awt.Dimension(652, 440));  
setResizable(false);
```

```
jButtonCadastrar.setFont(new java.awt.Font("Segoe UI", 1, 36)); // NOI18N  
jButtonCadastrar.setText("Cadastrar");  
jButtonCadastrar.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButtonCadastrarActionPerformed(evt);  
    }  
});
```

```
jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));
```

```
jLabelNome.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N  
jLabelNome.setText("Nome");
```

```
jTextFieldUsername.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
```

```
jLabelSenha.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N  
jLabelSenha.setText("Senha");
```

```
jPasswordFieldSenha.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
```

```
jLabelCPF.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N  
jLabelCPF.setText("CPF");
```

```
jTextFieldCPF.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N
```

```
jLabel5.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N  
jLabel5.setText("Função");
```

```
jComboBoxFuncao.setFont(new java.awt.Font("Segoe UI", 1, 24)); // NOI18N  
jComboBoxFuncao.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {  
    "Vendedor", "Estoquista" }));
```

```
jLabelErroCPF.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N  
jLabelErroCPF.setHorizontalAlignment(javax.swing.SwingConstants.TRAILING);  
jLabelErroCPF.setText(" ");
```

```
jLabelErroNome.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
```

```
jLabelErroNome.setHorizontalAlignment(javax.swing.SwingConstants.TRAILING);
jLabelErroNome.setText(" ");

jLabelErroSenha.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
jLabelErroSenha.setHorizontalAlignment(javax.swing.SwingConstants.TRAILING);
jLabelErroSenha.setText(" ");

jLabelError.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabelError, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(jTextFieldUsername)
        .addComponent(jPasswordFieldSenha)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jComboBoxFuncao, 0, 539, Short.MAX_VALUE))
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jLabelSenha)
            .addGap(32, 32, 32)
            .addComponent(jLabelErroSenha,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            .addComponent(jTextFieldCPF)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jLabelCPF, javax.swing.GroupLayout.PREFERRED_SIZE,
82, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(jLabelErroCPF, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jLabelNome)
```

```
.addGap(31, 31, 31)
.addComponent(jLabelErroNome, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
.addContainerGap()
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
NE)
.addComponent(jComboBoxFuncao)
.addComponent(jLabel5))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
NE)
.addComponent(jLabelCPF)
.addComponent(jLabelErroCPF))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jTextFieldCPF, javax.swing.GroupLayout.PREFERRED_SIZE,
42, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
NE)
.addComponent(jLabelNome)
.addComponent(jLabelErroNome))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jTextFieldUsername,
javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
NE)
.addComponent(jLabelSenha)
.addComponent(jLabelErroSenha))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(jPasswordFieldSenha,
javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
```



```
String senha = jPasswordFieldSenha.getText();
// Verificacao de nulo para toda caixa de texto
if(cpf.isBlank()){
    jLabelErroCPF.setText("Não pode ser nulo");
} else{
    jLabelErroCPF.setText(" ");
}
if(username.isBlank()){
    jLabelErroNome.setText("Não pode ser nulo");
} else{
    jLabelErroNome.setText(" ");
}
if(senha.isBlank()){
    jLabelErroSenha.setText("Não pode ser nulo");
} else{
    jLabelErroSenha.setText(" ");
}

// Caso nenhum campo for nulo, execute:
if(!cpf.isBlank() && !username.isBlank() && !senha.isBlank()){
    //log-in como admin para registrar usuários
    try {
        conexao.conectar();
        PreparedStatement ps;
        String sql;

        // Cria o usuário e dá as permissões da funcao (role) pra ele.
        // Prefixada e sufixada a letra u no nome do usuário pois o postgre não aceita
usuários completamente numéricos
        sql = "CREATE USER u" + cpf + "u WITH PASSWORD '" + senha + "';"
        + "GRANT " + funcao + " to u" + cpf + "u;"
        + "CALL insert_tb_funcionarios('" + username + "', '" + cpf + "', '" + senha + "', '" +
funcao + "');";
        ps = conexao.getConexao().prepareStatement(sql);
        ps.executeUpdate();

        // Encerra a transação e mostra mensagam de sucesso
        ps.close();
        JOptionPane.showMessageDialog(this, "Registrado usuário " + cpf + " (" +
username + ") " + " com sucesso!");
        dispose(); // Auto-fecha janela de cadastro;
    } catch (SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}
```



```
    }  
}  
}  
// Variables declaration - do not modify  
private javax.swing.JButton jButtonCadastrar;  
private javax.swing.JComboBox<String> jComboBoxFuncao;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabelCPF;  
private javax.swing.JLabel jLabelErroCPF;  
private javax.swing.JLabel jLabelErroNome;  
private javax.swing.JLabel jLabelErroSenha;  
private javax.swing.JLabel jLabelError;  
private javax.swing.JLabel jLabelNome;  
private javax.swing.JLabel jLabelSenha;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPasswordField jPasswordFieldSenha;  
private javax.swing.JTextField jTextFieldCPF;  
private javax.swing.JTextField jTextFieldUsername;  
// End of variables declaration  
}
```

F.7 TelaHistoricoVenda.java

```
package br.com.sistemas.userInterface;  
  
import javax.swing.table.DefaultTableModel;  
import br.com.sistemas.model.database.ConexaoBDPostgres;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.Vector;  
import javax.swing.JOptionPane;  
  
/**  
 *  
 * @author user  
 */  
public class TelaHistoricoVenda extends javax.swing.JFrame {  
  
    /**  
     * Creates new form TelaHistoricoVenda  
     */  
}
```

```
ConexaoBDPostgres conexao;
DefaultTableModel modelo = new DefaultTableModel(new Object [][] { },
    new String [] { "Código", "Horário", "Valor Total", "Funcionário"});

public TelaHistoricoVenda(ConexaoBDPostgres conexao) {
    this.conexao = conexao;
    initComponents();
    carregarVendas();
}

public final void carregarVendas() {
    String sql = "SELECT tb_vendas.ven_codigo, tb_vendas.ven_horario,
tb_vendas.ven_valor_total, view_funcionarios_login.fun_nome FROM tb_vendas "
        + "LEFT JOIN view_funcionarios_login ON tb_funcionarios_fun_codigo =
fun_codigo;";
    try (PreparedStatement ps = conexao.getConexao().prepareStatement(sql);
        ResultSet rs = ps.executeQuery()) {
        while(modelo.getRowCount() > 0) {
            modelo.removeRow(0);
        }
        while (rs.next()) {
            Vector linha = new Vector();
            linha.add(rs.getInt("ven_codigo"));
            linha.add(rs.getString("ven_horario"));
            linha.add(rs.getString("ven_valor_total"));
            linha.add(rs.getString("fun_nome"));
            modelo.addRow(linha);
        }
        ps.close();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {
```

```
jPanel1 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Historico de Venda");
```

```
jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));
```

```
jTable1.setModel(modelo);
jScrollPane1.setViewportView(jTable1);
```

```
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 721,
Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jTable1, javax.swing.GroupLayout.DEFAULT_SIZE, 416,
Short.MAX_VALUE)
            .addContainerGap())
);
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,  
Short.MAX_VALUE)  
);  
  
pack();  
} // </editor-fold>
```

```
// Variables declaration - do not modify  
private javax.swing.JPanel jPanel1;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable jTable1;  
// End of variables declaration  
}
```

F.8 TelaLogin.java

```
package br.com.sistemas.userInterface;
```

```
import br.com.sistemas.model.database.ConexaoBDPostgres;  
import javax.swing.JOptionPane;
```

```
/**  
 *  
 * @author Larissa  
 */  
public class TelaLogin extends javax.swing.JFrame {
```

```
/**  
 * Creates new form TelaLogin  
 */  
public TelaLogin() {  
    initComponents();  
}
```

```
/**  
 * This method is called from within the constructor to initialize the form.  
 * WARNING: Do NOT modify this code. The content of this method is always  
 * regenerated by the Form Editor.  
 */
```

```
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jButtonEntrar = new javax.swing.JButton();
    jPanel1 = new javax.swing.JPanel();
    jLabelSenha = new javax.swing.JLabel();
    jPasswordFieldSenha = new javax.swing.JPasswordField();
    jLabelCPF = new javax.swing.JLabel();
    jTextFieldCPF = new javax.swing.JTextField();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Login");
    setMinimumSize(new java.awt.Dimension(241, 246));
    setResizable(false);

    jButtonEntrar.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
    jButtonEntrar.setText("Entrar");
    jButtonEntrar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonEntrarActionPerformed(evt);
        }
    });

    jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));

    jLabelSenha.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    jLabelSenha.setText("Senha");

    jPasswordFieldSenha.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N

    jLabelCPF.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    jLabelCPF.setText("CPF");

    jTextFieldCPF.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .add(jPasswordFieldSenha)
                    .add(jLabelCPF)
                    .add(jTextFieldCPF)
                    .add(jButtonEntrar)
                )
                .addContainerGap())
    );
}
```

```
.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jTextFieldCPF, javax.swing.GroupLayout.PREFERRED_SIZE,
213, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jPasswordFieldSenha,
javax.swing.GroupLayout.PREFERRED_SIZE, 213,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabelSenha)
    .addComponent(jLabelCPF, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap()
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
    .addContainerGap()
    .addComponent(jLabelCPF)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jTextFieldCPF, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addComponent(jLabelSenha)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jPasswordFieldSenha,
javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap()
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButtonEntrar, javax.swing.GroupLayout.PREFERRED_SIZE,
229, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap()
    );
```

```
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButtonEntrar)
            .addContainerGap(9, Short.MAX_VALUE))
        );

setSize(new java.awt.Dimension(257, 207));
setLocationRelativeTo(null);
} // </editor-fold>

private void jButtonEntrarActionPerformed(java.awt.event.ActionEvent evt) {
    // Coleta o CPF e senha das caixas de texto da interface
    String nomeDoBanco = "BancoTrabalho2";
    String cpf = jTextFieldCPF.getText();
    String senha = jPasswordFieldSenha.getText();

    ConexaoBDPostgres conexao = new ConexaoBDPostgres(cpf, senha, nomeDoBanco);
    // Tenta conectar ao banco com os dados inseridos pelo usuário
    Boolean connectionStatus = conexao.conectar();
    if (connectionStatus){
        // Se admin, abre tela Home
        if(cpf.equals("admin") || cpf.equals("postgres")){
            TelaAdmin telaHome = new TelaAdmin(conexao);
            telaHome.setVisible(true);
        } else {
            // Senão, busca no banco pela função desse usuário
            String cargo = conexao.getCargo(cpf);
            // Se vendedor, abre tela Venda
            if(cargo.equals("vendedor")){
                TelaVenda telaVenda = new TelaVenda(conexao);
                telaVenda.setVisible(true);
            } //Se estoquista, abre tela Estoque
        } else if(cargo.equals("estoquista")){
            TelaEstoque telaEstoque = new TelaEstoque(conexao);
            telaEstoque.setVisible(true);
        }
    }
}
```

```

    } else{
        // Se não encontrar o cargo, desconecta e avisa o usuário
        conexao.disconnect();
        JOptionPane.showMessageDialog(this, "Erro! Cargo não encontrado");
    }
}
}

/* ----- estabelecer lógica para autenticar login e senha -----
--- if estabelecido numa tentativa falha de testar o login ----
if (result.next() && result.getInt(1) > 0) {
    System.out.println("Login e senha são válidos!");
    TelaHome telaInicial = new TelaHome();
    telaInicial.setVisible(true);
} else {
    System.out.println("Erro: login ou senha inválidos!");
}*/

}

// Para teste
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
    feel.
    * For details see
    http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(TelaLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

```



```
java.util.logging.Logger.getLogger(TelaLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(TelaLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(TelaLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>
```

```
/* Create and display the form */
java.awt.EventQueue.invokeLater(() -> {
    new TelaLogin().setVisible(true);
});
}
```

```
// Variables declaration - do not modify
private javax.swing.JButton jButtonEntrar;
private javax.swing.JLabel jLabelCPF;
private javax.swing.JLabel jLabelSenha;
private javax.swing.JPanel jPanel1;
private javax.swing.JPasswordField jPasswordFieldSenha;
private javax.swing.JTextField jTextFieldCPF;
// End of variables declaration
}
```

F.9 TelaRemocaoProduto.java

```
package br.com.sistemas.userInterface;
```

```
import br.com.sistemas.model.database.ConexaoBDPostgres;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
```

```
/**
 *
 * @author Bastos
 */
```

```
public class TelaRemocaoProduto extends javax.swing.JFrame {

    ConexaoBDPostgres conexao;

    public TelaRemocaoProduto(ConexaoBDPostgres conexao) {

        this.conexao = conexao;
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButtonOk = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jLabelNomeFornecedor = new javax.swing.JLabel();
        jTextFieldIdProduto = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Remover produto");

        jButtonOk.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
        jButtonOk.setText("Remover Produto");
        jButtonOk.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonOkActionPerformed(evt);
            }
        });

        jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));

        jLabelNomeFornecedor.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabelNomeFornecedor.setText("ID do Produto:");

        jTextFieldIdProduto.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jTextFieldIdProdutoActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jTextFieldIdProduto,
                        javax.swing.GroupLayout.PREFERRED_SIZE, 300,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(0, 0, Short.MAX_VALUE))
                .addComponent(jLabelNomeFornecedor,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE))
                .addContainerGap())
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabelNomeFornecedor)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jTextFieldIdProduto,
                    javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jTextFieldIdProduto,
                        javax.swing.GroupLayout.PREFERRED_SIZE, 300,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(0, 0, Short.MAX_VALUE))
                .addComponent(jLabelNomeFornecedor,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE))
                .addContainerGap())
    );
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jButtonOk, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButtonOk)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

pack();
setLocationRelativeTo(null);
} // </editor-fold>

private void jTextFieldIdProdutoActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jButtonOkActionPerformed(java.awt.event.ActionEvent evt) {
    String idProduto = jTextFieldIdProduto.getText();
    try {
        PreparedStatement ps = null;
        String sql = null;

        sql = "CALL remove_tb_produtos('" + idProduto + "');";
        ps = conexao.getConexao().prepareStatement(sql);
        ps.executeUpdate();

        // Encerra a transação e mostra mensagem de sucesso
        ps.close();
        JOptionPane.showMessageDialog(this, "Removido produto " + idProduto + " com
sucesso!");
    }
```

```
        dispose(); // Auto-fecha janela de remocao;
    } catch (SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}
```

```
// Variables declaration - do not modify
private javax.swing.JButton jButtonOk;
private javax.swing.JLabel jLabelNomeFornecedor;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextFieldIdProduto;
// End of variables declaration
}
```

F.10 TelaVenda.java

```
package br.com.sistemas.userInterface;
```

```
import br.com.sistemas.model.database.ConexaoBDPostgres;
import java.awt.Color;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

```
/**
 *
 * @author Bastos
 */
```

```
public class TelaVenda extends javax.swing.JFrame {
```

```
    /**
     * Creates new form TelaInicial
     */
```

```
    ConexaoBDPostgres conexao;
    DefaultTableModel modelo = new DefaultTableModel(new Object [][] { },
        new String [] { "Código", "Descrição", "Quantidade", "Valor Un", "Valor Total" });
```

```
    public TelaVenda(ConexaoBDPostgres conexao) {
```

```
this.conexao = conexao;
 initComponents();
}

public final void atualizarLista() {
    int quantidadeVenda = (int)seletorDeQuantidadeDeVenda.getValue();
    String busca = textoCodigoProduto.getText();
    String sql = "SELECT * FROM tb_produtos WHERE pro_codigo = " + busca + ";";
    try (PreparedStatement ps = conexao.getConexao().prepareStatement(sql);
        ResultSet rs = ps.executeQuery()) {
        while (rs.next()) {
            if(((rs.getInt("pro_quantidade") - quantidadeVenda) < 0)){
                JOptionPane.showMessageDialog(this, "Quantidade indisponível no estoque!
Disponível: " + rs.getInt("pro_quantidade"));
                ps.close();
                return;
            }
            Vector linha = new Vector();
            linha.add(rs.getInt("pro_codigo"));
            linha.add(rs.getString("pro_descricao"));
            linha.add(quantidadeVenda);
            linha.add(rs.getDouble("pro_valor"));
            linha.add(rs.getDouble("pro_valor") * quantidadeVenda);
            modelo.addRow(linha);
        }
        ps.close();
        if(modelo.getRowCount() == 0){
            JOptionPane.showMessageDialog(this, "Produto com ID " + busca + " não
encontrado!");
            return;
        }
    } catch (SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    mostrarSubtotal.setText("R$ " + calcularSubtotal() + "0");
}

public double calcularSubtotal(){
    double subtotal = 0.00;
    for(int i = 0; i < modelo.getRowCount(); i++){
        subtotal = subtotal + (double)modelo.getValueAt(i, 4);
    }
    return subtotal;
}
```

```
}

public void realizarVenda(){
    double subtotal = calcularSubtotal();
    String sql = "";
    int cons_cod;
    String cons_desc;
    int cons_quantidade;
    String cpf = conexao.getCPF();
    for(int i = 0; i < modelo.getRowCount(); i++){
        cons_cod = (int)modelo.getValueAt(i, 0);
        cons_desc = (String)modelo.getValueAt(i, 1);
        cons_quantidade = (int)modelo.getValueAt(i, 2);
        sql = sql.concat("CALL vender_produto(" + cons_cod + ", " + cons_desc + ", " +
cons_quantidade + ", " + subtotal + ", " + cpf + ");");
    }
    try {
        PreparedStatement ps = conexao.getConnection().prepareStatement(sql);
        ps.executeUpdate();
        ps.close();
        JOptionPane.showMessageDialog(this, "Venda realizada! Subtotal: R$ " + subtotal +
"0");
        modelo.setRowCount(0);
    } catch (SQLException e){
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jFrame1 = new javax.swing.JFrame();
    jFrame2 = new javax.swing.JFrame();
    jFrame3 = new javax.swing.JFrame();
    menuBar1 = new java.awt.MenuBar();
    menu1 = new java.awt.Menu();
    menu2 = new java.awt.Menu();
}
```

```
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenu2 = new javax.swing.JMenu();
jSeparator1 = new javax.swing.JSeparator();
jPanel1 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
tabelaMostrarProdutosVendas = new javax.swing.JTable();
jLabel1 = new javax.swing.JLabel();
buttonAdicionar = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();
textoCodigoProduto = new javax.swing.JTextField();
seletorDeQuantidadeDeVenda = new javax.swing.JSpinner();
jLabel4 = new javax.swing.JLabel();
mostrarSubtotal = new javax.swing.JTextField();
buttonConcluirVenda = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jLabelMensagemErroCodigo = new javax.swing.JLabel();
jLabelMensagemErroQuantidade = new javax.swing.JLabel();
jButtonAbrirHistorico = new javax.swing.JButton();
jLabelMensagemErroTabelaVazia = new javax.swing.JLabel();

javax.swing.GroupLayout jFrame1Layout = new
javax.swing.GroupLayout(jFrame1.getContentPane());
jFrame1.getContentPane().setLayout(jFrame1Layout);
jFrame1Layout.setHorizontalGroup(
    jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 400, Short.MAX_VALUE)
);
jFrame1Layout.setVerticalGroup(
    jFrame1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 300, Short.MAX_VALUE)
);

javax.swing.GroupLayout jFrame2Layout = new
javax.swing.GroupLayout(jFrame2.getContentPane());
jFrame2.getContentPane().setLayout(jFrame2Layout);
jFrame2Layout.setHorizontalGroup(
    jFrame2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 400, Short.MAX_VALUE)
);
jFrame2Layout.setVerticalGroup(
    jFrame2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 300, Short.MAX_VALUE)
);
```



```
);

javax.swing.GroupLayout jFrame3Layout = new
javax.swing.GroupLayout(jFrame3.getContentPane());
jFrame3.getContentPane().setLayout(jFrame3Layout);
jFrame3Layout.setHorizontalGroup(
    jFrame3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 400, Short.MAX_VALUE)
);
jFrame3Layout.setVerticalGroup(
    jFrame3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 300, Short.MAX_VALUE)
);

menu1.setLabel("File");
menuBar1.add(menu1);

menu2.setLabel("Edit");
menuBar1.add(menu2);

jMenu1.setText("File");
jMenuBar1.add(jMenu1);

jMenu2.setText("Edit");
jMenuBar1.add(jMenu2);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Venda");
setResizable(false);

jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));
jPanel1.setRequestFocusEnabled(false);
jPanel1.setVerifyInputWhenFocusTarget(false);

tabelaMostrarProdutosVendas.setModel(modelo);
tabelaMostrarProdutosVendas.setEnabled(false);
jScrollPane1.setViewportView(tabelaMostrarProdutosVendas);
if (tabelaMostrarProdutosVendas.getColumnModel().getColumnCount() > 0) {
    tabelaMostrarProdutosVendas.getColumnModel().getColumn(1).setResizable(false);
    tabelaMostrarProdutosVendas.getColumnModel().getColumn(3).setResizable(false);
}
```

```
jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
jLabel1.setText("Produtos:");

buttonAdicionar.setText("Adicionar");
buttonAdicionar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        buttonAdicionarActionPerformed(evt);
    }
});

jLabel3.setText("Código do Produto:");

textoCodigoProduto.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        textoCodigoProdutoActionPerformed(evt);
    }
});

seletorDeQuantidadeDeVenda.setInheritsPopupMenu(true);
seletorDeQuantidadeDeVenda.addChangeListener(new
javax.swing.event.ChangeListener() {
    public void stateChanged(javax.swing.event.ChangeEvent evt) {
        seletorDeQuantidadeDeVendaStateChanged(evt);
    }
});

jLabel4.setText("Subtotal");

mostrarSubtotal.setEditable(false);
mostrarSubtotal.setBackground(new java.awt.Color(255, 255, 255));
mostrarSubtotal.setText("R$ 0.00");
mostrarSubtotal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mostrarSubtotalActionPerformed(evt);
    }
});

buttonConcluirVenda.setText("Concluir Venda");
buttonConcluirVenda.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        buttonConcluirVendaActionPerformed(evt);
    }
});
```

```
});

jLabel2.setText("Quantidade");

jLabelMensagemErroCodigo.setBackground(new java.awt.Color(214, 217, 223));
jLabelMensagemErroCodigo.setForeground(new java.awt.Color(214, 217, 223));
jLabelMensagemErroCodigo.setText("Insira o código!");

jLabelMensagemErroQuantidade.setForeground(new java.awt.Color(214, 217, 223));
jLabelMensagemErroQuantidade.setText("Insira uma quantidade!");

jButtonAbrirHistorico.setText("Histórico");
jButtonAbrirHistorico.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonAbrirHistoricoActionPerformed(evt);
    }
});

jLabelMensagemErroTabelaVazia.setForeground(new java.awt.Color(214, 217, 223));
jLabelMensagemErroTabelaVazia.setText("Adicione um produto!");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(jLabelMensagemErroCodigo)
                .add(jLabelMensagemErroQuantidade)
                .add(jLabelMensagemErroTabelaVazia)
            )
            .addContainerGap())
        .addGroup(jPanel1Layout.createSequentialGroup()
            .add(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 586, javax.swing.GroupLayout.PREFERRED_SIZE)
            .add(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 126, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
        .addGroup(jPanel1Layout.createSequentialGroup()
            .add(jLabelMensagemErroCodigo)
            .add(jLabelMensagemErroQuantidade)
            .add(jLabelMensagemErroTabelaVazia)
            .addContainerGap())
        .addGroup(jPanel1Layout.createSequentialGroup()
            .add(jScrollPane1)
            .add(jLabel1)
            .addContainerGap())
);
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(mostrarSubtotal)
.addComponent(buttonConcluirVenda,
javax.swing.GroupLayout.DEFAULT_SIZE, 169, Short.MAX_VALUE)
.addComponent(jButtonAbrirHistorico,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
.addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
169, Short.MAX_VALUE)
.addComponent(textoCodigoProduto)
.addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabelMensagemErroQuantidade)
.addGroup(jPanel1Layout.createSequentialGroup()
.addComponent(jLabel2)
.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(buttonAdicionar)
.addComponent(selectorDeQuantidadeDeVenda,
javax.swing.GroupLayout.PREFERRED_SIZE, 81,
javax.swing.GroupLayout.PREFERRED_SIZE))))
.addGroup(jPanel1Layout.createSequentialGroup()
.addGap(27, 27, 27)
.addComponent(jLabelMensagemErroTabelaVazia))
.addContainerGap(16, Short.MAX_VALUE))))
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel3))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(textoCodigoProduto,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabelMensagemErroCodigo)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(seletorDeQuantidadeDeVenda,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(buttonAdicionar)
    .addGap(3, 3, 3)
    .addComponent(jLabelMensagemErroQuantidade)
    .addGap(18, 18, 18)
    .addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(mostrarSubtotal,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addComponent(buttonConcluirVenda)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jLabelMensagemErroTabelaVazia)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jButtonAbrirHistorico))
    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
456, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(6, 6, 6))
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
);

pack();
setLocationRelativeTo(null);
} // </editor-fold>

private void nomeOperadorActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void menuArquivosActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void menuEstoqueActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void mostrarSubtotalActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void
seletorDeQuantidadeDeVendaStateChanged(javax.swing.event.ChangeEvent evt) {
```

```
        if((int)seletorDeQuantidadeDeVenda.getValue() < 0){
            seletorDeQuantidadeDeVenda.setValue(0);
        }
    }

    private void textoCodigoProdutoActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO: Implementar lógica para remover letras automaticamente
    }

    private void buttonAdicionarActionPerformed(java.awt.event.ActionEvent evt) {
        String codProduto = textoCodigoProduto.getText();

        if((int)seletorDeQuantidadeDeVenda.getValue() <= 0 ){
            jLabelMensagemErroQuantidade.setForeground(Color.red);
        } else{
            jLabelMensagemErroQuantidade.setForeground(new Color(214, 217, 223));
        }

        if(codProduto.isBlank()){
            jLabelMensagemErroCodigo.setForeground(Color.red);
        } else
        {
            jLabelMensagemErroCodigo.setForeground(new Color(214, 217, 223));
        }

        if(!((int)seletorDeQuantidadeDeVenda.getValue() <= 0 || codProduto.isBlank())){
            atualizarLista();
        }
    }

    private void buttonConcluirVendaActionPerformed(java.awt.event.ActionEvent evt) {
        if(modelo.getRowCount() == 0){
            jLabelMensagemErroTabelaVazia.setForeground(Color.red);
        } else{
            jLabelMensagemErroTabelaVazia.setForeground(new Color(214, 217, 223));
            realizarVenda();
        }
    }

    private void jButtonAbrirHistoricoActionPerformed(java.awt.event.ActionEvent evt) {
        TelaHistoricoVenda telaHistoricoVenda = new TelaHistoricoVenda(conexao);
        telaHistoricoVenda.setVisible(true);
    }
```

```
// Variables declaration - do not modify
private javax.swing.JButton buttonAdicionar;
private javax.swing.JButton buttonConcluirVenda;
private javax.swing.JButton jButtonAbrirHistorico;
private javax.swing.JFrame jFrame1;
private javax.swing.JFrame jFrame2;
private javax.swing.JFrame jFrame3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabelMensagemErroCodigo;
private javax.swing.JLabel jLabelMensagemErroQuantidade;
private javax.swing.JLabel jLabelMensagemErroTabelaVazia;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JSeparator jSeparator1;
private java.awt.Menu menu1;
private java.awt.Menu menu2;
private java.awt.MenuBar menuBar1;
private javax.swing.JTextField mostrarSubtotal;
private javax.swing.JSpinner seletorDeQuantidadeDeVenda;
private javax.swing.JTable tabelaMostrarProdutosVendas;
private javax.swing.JTextField textoCodigoProduto;
// End of variables declaration
}
```