|  |
| --- |
| **USTM**  UNIVERSIDADE SÃO TOMÁS DE MOÇAMBIQUE - USTM  FACULDADE DE CIÊNCIAS E TECNOLOGIAS DE INFORMAÇÃO    **Curso:** Licenciatura em Desenvolvimento de Software - LDS  **Cadeira:** POOA    **Discentes:**  António Manuel Sitóe (2022 010 61)  **Turma:** 4P7LDS1  Maputo, Junho de 2025 |

package view;

import controller.ControllerFuncionario;

import model.Funcionario;

import javax.swing.*\**;

import javax.swing.border.TitledBorder;

import javax.swing.table.DefaultTableModel;

import java.awt.*\**;

import java.awt.event.*\**;

import java.sql.SQLException;

import java.util.ArrayList;

public class ViewFuncionario implements *ActionListener*, *MouseListener* {

private JFrame frame;

private JTable tabela;

private JTextField txtCodigo, txtNome, txtContacto, txtDiasTrabalhados, txtSalarioDiario;

private JComboBox<String> cbDepartamento, cbEstadoCivil;

private JRadioButton rbMasculino, rbFeminino;

private ButtonGroup grupoGenero;

private JButton btnAdicionar, btnListar, btnEditar, btnRemover, btnLimpar, btnNovo, btnFiltrar;

public static *void* main(String[] *args*) {

EventQueue.invokeLater(() *->* {

try {

ViewFuncionario window = **new** ViewFuncionario();

window.frame.setVisible(true);

} catch (Exception *e*) {

e.printStackTrace();}});}

public ViewFuncionario() {initialize();}

private *void* initialize() {

frame = **new** JFrame();

frame.setTitle("Gestão de Funcionários");

frame.setBounds(100, 100, 1050, 600);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JPanel panel = **new** JPanel();

panel.setBorder(**new** TitledBorder(null, "Dados do Funcionário", TitledBorder.LEADING, TitledBorder.TOP, null, null));

panel.setBounds(15, 15, 680, 270);

frame.getContentPane().add(panel);

panel.setLayout(null);

JLabel lblCodigo = **new** JLabel("Código:");

lblCodigo.setBounds(15, 30, 80, 20);

panel.add(lblCodigo);

JLabel lblNome = **new** JLabel("Nome:");

lblNome.setBounds(15, 65, 80, 20);

panel.add(lblNome);

JLabel lblContacto = **new** JLabel("Contacto:");

lblContacto.setBounds(15, 100, 80, 20);

panel.add(lblContacto);

JLabel lblDepartamento = **new** JLabel("Departamento:");

lblDepartamento.setBounds(15, 135, 100, 20);

panel.add(lblDepartamento);

JLabel lblGenero = **new** JLabel("Gênero:");

lblGenero.setBounds(15, 170, 80, 20);

panel.add(lblGenero);

JLabel lblEstadoCivil = **new** JLabel("Estado Civil:");

lblEstadoCivil.setBounds(15, 205, 80, 20);

panel.add(lblEstadoCivil);

JLabel lblDias = **new** JLabel("Dias Trabalhados:");

lblDias.setBounds(15, 240, 120, 20);

panel.add(lblDias);

JLabel lblSalario = **new** JLabel("Salário Diário:");

lblSalario.setBounds(350, 240, 100, 20);

panel.add(lblSalario);

txtCodigo = **new** JTextField();

txtCodigo.setBounds(120, 25, 180, 30);

panel.add(txtCodigo);

txtNome = **new** JTextField();

txtNome.setBounds(120, 60, 400, 30);

panel.add(txtNome);

txtContacto = **new** JTextField();

txtContacto.setBounds(120, 95, 250, 30);

panel.add(txtContacto);

cbDepartamento = **new** JComboBox<>(**new** String[] { "RH", "Financeiro", "TI", "Marketing", "Operacoes" });

cbDepartamento.setBounds(120, 130, 250, 30);

panel.add(cbDepartamento);

rbMasculino = **new** JRadioButton("Masculino");

rbMasculino.setBounds(120, 165, 120, 30);

panel.add(rbMasculino);

rbFeminino = **new** JRadioButton("Feminino");

rbFeminino.setBounds(250, 165, 120, 30);

panel.add(rbFeminino);

grupoGenero = **new** ButtonGroup();

grupoGenero.add(rbMasculino);

grupoGenero.add(rbFeminino);

cbEstadoCivil = **new** JComboBox<>(**new** String[] { "Solteiro", "Casado", "Divorciado", "Viúvo" });

cbEstadoCivil.setBounds(120, 200, 250, 30);

panel.add(cbEstadoCivil);

txtDiasTrabalhados = **new** JTextField();

txtDiasTrabalhados.setBounds(140, 235, 180, 30);

panel.add(txtDiasTrabalhados);

txtSalarioDiario = **new** JTextField();

txtSalarioDiario.setBounds(440, 235, 180, 30);

panel.add(txtSalarioDiario);

JPanel panelOperacoes = **new** JPanel();

panelOperacoes.setBorder(**new** TitledBorder(null, "Operações", TitledBorder.LEADING, TitledBorder.TOP, null, null));

panelOperacoes.setBounds(710, 15, 300, 270);

frame.getContentPane().add(panelOperacoes);

panelOperacoes.setLayout(null);

btnAdicionar = **new** JButton("Adicionar");

btnAdicionar.setBounds(50, 15, 200, 35);

btnAdicionar.addActionListener(*this*);

panelOperacoes.add(btnAdicionar);

btnListar = **new** JButton("Listar Todos");

btnListar.setBounds(50, 55, 200, 35);

btnListar.addActionListener(*this*);

panelOperacoes.add(btnListar);

btnFiltrar = **new** JButton("Filtrar");

btnFiltrar.setBounds(50, 215, 200, 35);

btnFiltrar.addActionListener(*this*);

panelOperacoes.add(btnFiltrar);

btnEditar = **new** JButton("Editar");

btnEditar.setBounds(50, 95, 200, 35);

btnEditar.addActionListener(*this*);

panelOperacoes.add(btnEditar);

btnRemover = **new** JButton("Remover");

btnRemover.setBounds(50, 135, 200, 35);

btnRemover.addActionListener(*this*);

panelOperacoes.add(btnRemover);

btnLimpar = **new** JButton("Limpar");

btnLimpar.setBounds(50, 175, 200, 35);

btnLimpar.addActionListener(*this*);

panelOperacoes.add(btnLimpar);

JPanel panelTabela = **new** JPanel();

panelTabela.setBorder(**new** TitledBorder(null, "Listagem de Funcionários", TitledBorder.LEADING, TitledBorder.TOP, null, null));

panelTabela.setBounds(15, 300, 1000, 250);

frame.getContentPane().add(panelTabela);

panelTabela.setLayout(**new** BorderLayout());

JScrollPane scrollPane = **new** JScrollPane();

panelTabela.add(scrollPane, BorderLayout.CENTER);

tabela = **new** JTable();

tabela.setModel(**new** DefaultTableModel(**new** Object[][] {}, **new** String[] {

"Código", "Nome", "Contacto", "Departamento", "Gênero", "Estado Civil", "Dias Trabalhados", "Salário Diário"

}));

tabela.addMouseListener(*this*);

scrollPane.setViewportView(tabela);

}

@Override

public *void* actionPerformed(ActionEvent *e*) {

try {

if (*e*.getSource() == btnAdicionar) {

Funcionario funcionario = criarFuncionarioDoFormulario();

ControllerFuncionario.adicionarFuncionario(funcionario);

listar();

limparCampos();

JOptionPane.showMessageDialog(null, "Funcionário adicionado com sucesso.");

} else if (*e*.getSource() == btnEditar) {

Funcionario funcionario = criarFuncionarioDoFormulario();

ControllerFuncionario.actualizarFuncionario(funcionario);

listar();

limparCampos();

JOptionPane.showMessageDialog(null, "Funcionário atualizado com sucesso.");

} else if (*e*.getSource() == btnRemover) {

*int* codigo = Integer.parseInt(txtCodigo.getText());

ControllerFuncionario.removerFuncionario(codigo);

listar();

limparCampos();

JOptionPane.showMessageDialog(null, "Funcionário removido com sucesso.");

} else if (*e*.getSource() == btnListar) {

listar();

}else if (*e*.getSource() == btnFiltrar) {

mostrarFuncionariosFiltrados();

} else if (*e*.getSource() == btnLimpar) {

limparCampos();

} else if (*e*.getSource() == btnNovo) {

gerarNovoCodigo();

}

} catch (SQLException | NumberFormatException *ex*) {

JOptionPane.showMessageDialog(null, "Erro: " + ex.getMessage());

}

}

private *void* mostrarFuncionariosFiltrados() {

try {

ArrayList<Funcionario> lista = ControllerFuncionario.listarFuncionariosFiltrados();

DefaultTableModel modelo = (DefaultTableModel) tabela.getModel();

modelo.setRowCount(0);

for (Funcionario f : lista) {

modelo.addRow(**new** Object[]{

f.getCodigo(),

f.getNome(),

f.getContacto(),

f.getDepartamento(),

f.getGenero(),

f.getEstadoCivil(),

f.getDiasTrabalhados(),

f.getSalarioDiario(),

f.calcularSalario()

});

}

} catch (SQLException *e*) {

JOptionPane.showMessageDialog(frame, "Erro ao carregar funcionários filtrados: " + e.getMessage());

}

}

private Funcionario criarFuncionarioDoFormulario() {

String genero = rbMasculino.isSelected() ? "Masculino" : rbFeminino.isSelected() ? "Feminino" : "";

return **new** Funcionario(

Integer.parseInt(txtCodigo.getText()),

txtNome.getText(),

txtContacto.getText(),

(String) cbDepartamento.getSelectedItem(),

genero,

(String) cbEstadoCivil.getSelectedItem(),

Integer.parseInt(txtDiasTrabalhados.getText()),

Double.parseDouble(txtSalarioDiario.getText()));

}

private *void* listar() throws SQLException {

DefaultTableModel modelo = (DefaultTableModel) tabela.getModel();

modelo.setRowCount(0);

ArrayList<Funcionario> lista = ControllerFuncionario.listarFuncionarios();

for (Funcionario f : lista) {

modelo.addRow(**new** Object[] {

f.getCodigo(),

f.getNome(),

f.getContacto(),

f.getDepartamento(),

f.getGenero(),

f.getEstadoCivil(),

f.getDiasTrabalhados(),

f.getSalarioDiario()

});}}

private *void* limparCampos() {

txtCodigo.setText("");

txtNome.setText("");

txtContacto.setText("");

cbDepartamento.setSelectedIndex(0);

grupoGenero.clearSelection();

cbEstadoCivil.setSelectedIndex(0);

txtDiasTrabalhados.setText("");

txtSalarioDiario.setText("");

txtCodigo.setEnabled(true);

}

private *void* gerarNovoCodigo() throws SQLException {

ArrayList<Funcionario> lista = ControllerFuncionario.listarFuncionarios();

*int* novoCodigo = lista.isEmpty() ? 1 : lista.get(lista.size() - 1).getCodigo() + 1;

txtCodigo.setText(String.valueOf(novoCodigo));

txtCodigo.setEnabled(false);

}

@Override

public *void* mouseClicked(MouseEvent *e*) {

if (tabela.getSelectedRow() != -1) {

*int* row = tabela.getSelectedRow();

txtCodigo.setText(tabela.getValueAt(row, 0).toString());

txtNome.setText(tabela.getValueAt(row, 1).toString());

txtContacto.setText(tabela.getValueAt(row, 2).toString());

cbDepartamento.setSelectedItem(tabela.getValueAt(row, 3).toString());

String genero = tabela.getValueAt(row, 4).toString();

if (genero.equals("Masculino")) {

rbMasculino.setSelected(true);

} else if (genero.equals("Feminino")) {

rbFeminino.setSelected(true);

} else {

grupoGenero.clearSelection();}

cbEstadoCivil.setSelectedItem(tabela.getValueAt(row, 5).toString());

txtDiasTrabalhados.setText(tabela.getValueAt(row, 6).toString());

txtSalarioDiario.setText(tabela.getValueAt(row, 7).toString());

txtCodigo.setEnabled(false);}}

@Override

public *void* mousePressed(MouseEvent *e*) {}

@Override

public *void* mouseReleased(MouseEvent *e*) {}

@Override

public *void* mouseEntered(MouseEvent *e*) {}

@Override

public *void* mouseExited(MouseEvent *e*) {}

}

package model;

import java.io.*Serializable*;

public class Funcionario implements *Comparable*<Funcionario>, *Serializable* {

private static final *long* serialVersionUID = 1L;

private *int* codigo;

private String nome;

private String contacto;

private String departamento;

private String genero;

private String estadoCivil;

private *int* diasTrabalhados;

private *double* salarioDiario;

public Funcionario(*int* *codigo*, String *nome*, String *contacto*, String *departamento*,

String *genero*, String *estadoCivil*, *int* *diasTrabalhados*, *double* *salarioDiario*) {

*this*.codigo = *codigo*;

*this*.nome = *nome*;

*this*.contacto = *contacto*;

*this*.departamento = *departamento*;

*this*.genero = *genero*;

*this*.estadoCivil = *estadoCivil*;

*this*.diasTrabalhados = *diasTrabalhados*;

*this*.salarioDiario = *salarioDiario*;

}

public *int* getCodigo() {

return codigo;

}

public *void* setCodigo(*int* *codigo*) {

*this*.codigo = *codigo*;

}

public String getNome() {

return nome;

}

public *void* setNome(String *nome*) {

*this*.nome = *nome*;

}

public String getContacto() {

return contacto;

}

public *void* setContacto(String *contacto*) {

*this*.contacto = *contacto*;

}

public String getDepartamento() {

return departamento;

}

public *void* setDepartamento(String *departamento*) {

*this*.departamento = *departamento*;

}

public String getGenero() {

return genero;

}

public *void* setGenero(String *genero*) {

*this*.genero = *genero*;

}

public String getEstadoCivil() {

return estadoCivil;

}

public *void* setEstadoCivil(String *estadoCivil*) {

*this*.estadoCivil = *estadoCivil*;

}

public *int* getDiasTrabalhados() {

return diasTrabalhados;

}

public *void* setDiasTrabalhados(*int* *diasTrabalhados*) {

*this*.diasTrabalhados = *diasTrabalhados*;

}

public *double* getSalarioDiario() {

return salarioDiario;

}

public *void* setSalarioDiario(*double* *salarioDiario*) {

*this*.salarioDiario = *salarioDiario*;

}

public *double* calcularSalario() {

return diasTrabalhados \* salarioDiario;

}

public *int* compareTo(Funcionario *outro*) {

return Integer.compare(*this*.codigo, *outro*.codigo);

}

public String toString() {

return codigo + " - " + nome + " - " + contacto + " - " + departamento + " - " +

genero + " - " + estadoCivil + " - " + diasTrabalhados + " dias - Salário: " + calcularSalario();

}

}

package controller;

import java.sql.*\**;

import java.util.ArrayList;

import model.Funcionario;

public class ControllerFuncionario {

public static *void* adicionarFuncionario(Funcionario *funcionario*) throws SQLException {

String sql = "INSERT INTO funcionario (codigo, nome, contacto, departamento, genero, estado\_civil, dias\_trabalhados, salario\_diario) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

try (*Connection* con = Conexao.conectar(); *PreparedStatement* stmt = con.prepareStatement(sql)) {

stmt.setInt(1, *funcionario*.getCodigo());

stmt.setString(2, *funcionario*.getNome());

stmt.setString(3, *funcionario*.getContacto());

stmt.setString(4, *funcionario*.getDepartamento());

stmt.setString(5, *funcionario*.getGenero());

stmt.setString(6, *funcionario*.getEstadoCivil());

stmt.setInt(7, *funcionario*.getDiasTrabalhados());

stmt.setDouble(8, *funcionario*.getSalarioDiario());

stmt.executeUpdate();

}

}

public static ArrayList<Funcionario> listarFuncionariosFiltrados() throws SQLException {

ArrayList<Funcionario> lista = **new** ArrayList<>();

String sql = "SELECT \* FROM funcionario WHERE (departamento = 'Operacoes' OR departamento = 'RH') AND (dias\_trabalhados \* salario\_diario) > 50000 AND dias\_trabalhados >= 30 AND genero = 'Masculino' AND estado\_civil IN ('Casado', 'Solteiro', 'Viúvo')";

try (*Connection* con = Conexao.conectar();

*PreparedStatement* stmt = con.prepareStatement(sql);

*ResultSet* rs = stmt.executeQuery()) {

while (rs.next()) {

lista.add(**new** Funcionario(

rs.getInt("codigo"),

rs.getString("nome"),

rs.getString("contacto"),

rs.getString("departamento"),

rs.getString("genero"),

rs.getString("estado\_civil"),

rs.getInt("dias\_trabalhados"),

rs.getDouble("salario\_diario")

));

}

}

return lista;

}

public static ArrayList<Funcionario> listarFuncionarios() throws SQLException {

ArrayList<Funcionario> lista = **new** ArrayList<>();

String sql = "SELECT \* FROM funcionario";

try (*Connection* con = Conexao.conectar();

*PreparedStatement* stmt = con.prepareStatement(sql);

*ResultSet* rs = stmt.executeQuery()) {

while (rs.next()) {

lista.add(**new** Funcionario(

rs.getInt("codigo"),

rs.getString("nome"),

rs.getString("contacto"),

rs.getString("departamento"),

rs.getString("genero"),

rs.getString("estado\_civil"),

rs.getInt("dias\_trabalhados"),

rs.getDouble("salario\_diario")));

}

}

return lista;

}

public static *void* actualizarFuncionario(Funcionario *funcionario*) throws SQLException {

String sql = "UPDATE funcionario SET nome=?, contacto=?, departamento=?, genero=?, estado\_civil=?, dias\_trabalhados=?, salario\_diario=? WHERE codigo=?";

try (*Connection* con = Conexao.conectar(); *PreparedStatement* stmt = con.prepareStatement(sql)) {

stmt.setString(1, *funcionario*.getNome());

stmt.setString(2, *funcionario*.getContacto());

stmt.setString(3, *funcionario*.getDepartamento());

stmt.setString(4, *funcionario*.getGenero());

stmt.setString(5, *funcionario*.getEstadoCivil());

stmt.setInt(6, *funcionario*.getDiasTrabalhados());

stmt.setDouble(7, *funcionario*.getSalarioDiario());

stmt.setInt(8, *funcionario*.getCodigo());

stmt.executeUpdate();

}

}

public static *void* removerFuncionario(*int* *codigo*) throws SQLException {

String sql = "DELETE FROM funcionario WHERE codigo=?";

try (*Connection* con = Conexao.conectar(); *PreparedStatement* stmt = con.prepareStatement(sql)) {

stmt.setInt(1, *codigo*);

stmt.executeUpdate();

}

}

}

package controller;

import java.sql.*Connection*;

import java.sql.DriverManager;

import java.sql.SQLException;

import javax.swing.JOptionPane;

public class Conexao {

public static *Connection* conectar() {

*Connection* conexao = null;

try {

conexao = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/crudfuncionario?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true",

"appuser",

"senha");

} catch (SQLException *e*) {

JOptionPane.showMessageDialog(null, "Erro ao conectar: " + e.getMessage());

}

return conexao;

}

public static *void* main(String[] *args*) {

try (*Connection* conn = Conexao.conectar()) {

if (conn != null) {

System.out.println("Conexão bem-sucedida!");

} else {

System.out.println("Falha ao conectar.");

}

} catch (Exception *e*) {

e.printStackTrace();

}

}