```
/** * Esta classe representa a interface de usuário para autenticação de login. * O código-fonte foi
descompilado de um arquivo .class usando o decompilador FernFlower. * * @author [Seu Nome] *
@version 1.0 * @since 2024-06-16 */// Source code is decompiled from a .class file using FernFlower
decompiler. package view; import java.awt.Component; import java.awt.Dimension; import java.awt.Font;
import java.awt.GridBagConstraints; import java.awt.GridBagLayout; import java.awt.Insets; import
java.sql.Connection; import java.sql.PreparedStatement; import java.sql.ResultSet; import
javax.swing.JButton; import javax.swing.JFrame; import javax.swing.JLabel; import
javax.swing.JOptionPane; import javax.swing.JPasswordField; import javax.swing.JTextField; import
javax.swing.SwingUtilities; import model.ConnectionFactory; import model.Funcionario; import
model.UsuarioLogado; public class LoginView extends JFrame { private JTextField loginField; private
JPasswordField passwordField; private JButton loginButton; /** * Cria uma nova instância da interface de
login. */ public LoginView() { this.setTitle("Login"); this.setSize(600, 400);
this.setDefaultCloseOperation(3); this.setLocationRelativeTo((Component)null); this.setLayout(new
GridBagLayout()); GridBagConstraints gbc = new GridBagConstraints(); gbc.fill = 2; gbc.insets = new
Insets(5, 5, 5, 5); JLabel textoLabel = new JLabel("Sistema da Biblioteca Leitura+"); textoLabel.setFont(new
Font("Verdana", 1, 18)); textoLabel.setHorizontalAlignment(0); gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth
= 2; gbc.insets = new Insets(10, 10, 20, 10); this.add(textoLabel, gbc); JLabel loginLabel = new
JLabel("Login:"); gbc.gridx = 0; gbc.gridy = 1; gbc.gridwidth = 1; gbc.anchor = 13; this.add(loginLabel,
gbc); this.loginField = new JTextField(15); gbc.gridx = 1; gbc.gridy = 1; gbc.anchor = 17;
this.add(this.loginField, gbc); JLabel passwordLabel = new JLabel("Senha:"); gbc.gridx = 0; gbc.gridy = 2;
gbc.anchor = 13; this.add(passwordLabel, gbc); this.passwordField = new JPasswordField(15); gbc.gridx =
1; gbc.gridy = 2; gbc.anchor = 17; this.add(this.passwordField, gbc); this.loginButton = new
JButton("Login"); this.loginButton.setPreferredSize(new Dimension(25, 25)); gbc.gridx = 1; gbc.gridy = 3;
gbc.anchor = 10; this.add(this.loginButton, gbc); this.loginButton.addActionListener(new 1(this)); } /** *
Autentica o usuário com as credenciais fornecidas. */ private void authenticateUser() { String login =
this.loginField.getText(); String password = new String(this.passwordField.getPassword()); try { Throwable
var3 = null; Object var4 = null; try { Connection conn = ConnectionFactory.getConnection(); try { String sql
= "SELECT * FROM funcionario WHERE login = ? AND senha = ?"; PreparedStatement stmt =
conn.prepareStatement(sql); stmt.setString(1, login); stmt.setString(2, password); ResultSet rs =
stmt.executeQuery(); if (rs.next()) { String nomeCompleto = rs.getString("nome_completo"); Funcionario
funcionario = new Funcionario(rs.getInt("id"), login, password, nomeCompleto);
UsuarioLogado.setFuncionario(funcionario); PrincipalView principalView = new
PrincipalView(funcionario); principalView.setVisible(true); this.dispose(); } else {
JOptionPane.showMessageDialog(this, "Login ou senha incorretos.", "Erro", 0); } } finally { if (conn !=
null) { conn.close(); } } } catch (Throwable var19) { if (var3 == null) { var3 = var19; } else if (var3 !=
var19) { var3.addSuppressed(var19); } throw var3; } } catch (Exception var20) { var20.printStackTrace();
JOptionPane.showMessageDialog(this, "Erro ao conectar ao banco de dados.", "Erro", 0); } } /** * O
método principal para iniciar a aplicação. * * @param args Argumentos de linha de comando (não utilizados
neste caso) */ public static void main(String[] args) { SwingUtilities.invokeLater(new 2()); } }
```