

```

/** * Esta classe representa a interface grÃfica para cadastrar um novo livro no sistema de biblioteca. * Os
dados sÃ£o inseridos em uma tabela do banco de dados. * O cÃdigo-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.LayoutManager; import java.sql.Connection; import
java.sql.PreparedStatement; import javax.swing.JButton; import javax.swing.JFrame; import
javax.swing.JLabel; import javax.swing.JOptionPane; import javax.swing.JTextField; import
model.ConnectionFactory; import model.Funcionario; public class CadastroLivroView extends JFrame {
private final Funcionario funcionario; private final ListarLivrosView listarLivrosView; private final
JTextField tituloField; private final JTextField autorField; private final JTextField generoField; private final
JTextField quantidadeField; private final JButton cadastrarButton; /** * Construtor da classe
CadastroLivroView. * * @param funcionario O funcionÃrio responsÃvel pelo cadastro. * @param
listarLivrosView A instÃncia da ListarLivrosView associada. */ public CadastroLivroView(Funcionario
funcionario, ListarLivrosView listarLivrosView) { this.funcionario = funcionario; this.listarLivrosView =
listarLivrosView; this.setTitle("Sistema de Biblioteca - Cadastro de Livro"); this.setSize(400, 300);
this.setDefaultCloseOperation(2); this.setLocationRelativeTo((Component)null);
this.setLayout((LayoutManager)null); JLabel tituloLabel = new JLabel("T\u00edtulo:");
tituloLabel.setBounds(20, 20, 100, 25); this.add(tituloLabel); this.tituloField = new JTextField();
this.tituloField.setBounds(120, 20, 200, 25); this.add(this.tituloField); JLabel autorLabel = new
JLabel("Autor:"); autorLabel.setBounds(20, 60, 100, 25); this.add(autorLabel); this.autorField = new
JTextField(); this.autorField.setBounds(120, 60, 200, 25); this.add(this.autorField); JLabel generoLabel =
new JLabel("G\u00eanero:"); generoLabel.setBounds(20, 100, 100, 25); this.add(generoLabel);
this.generoField = new JTextField(); this.generoField.setBounds(120, 100, 200, 25);
this.add(this.generoField); JLabel quantidadeLabel = new JLabel("Quantidade:");
quantidadeLabel.setBounds(20, 140, 100, 25); this.add(quantidadeLabel); this.quantidadeField = new
JTextField(); this.quantidadeField.setBounds(120, 140, 200, 25); this.add(this.quantidadeField);
this.cadastrarButton = new JButton("Cadastrar"); this.cadastrarButton.setBounds(120, 180, 100, 25);
this.add(this.cadastrarButton); this.cadastrarButton.addActionListener(new 1(this)); } /** * Realiza o
cadastro do livro no banco de dados. */ private void cadastrarLivro() { String titulo =
this.tituloField.getText(); String autor = this.autorField.getText(); String genero = this.generoField.getText();
int quantidade = Integer.parseInt(this.quantidadeField.getText()); try { Throwable var5 = null; Object var6 =
null; try { Connection conn = ConnectionFactory.getConnection(); try { String sql = "INSERT INTO livro
(titulo, autor, genero, quantidade) VALUES (?, ?, ?, ?)"; PreparedStatement stmt =
conn.prepareStatement(sql); stmt.setString(1, titulo); stmt.setString(2, autor); stmt.setString(3, genero);
stmt.setInt(4, quantidade); stmt.executeUpdate(); JOptionPane.showMessageDialog(this, "Livro cadastrado
com sucesso."); this.listarLivrosView.carregarLivros(); } finally { if (conn != null) { conn.close(); } } } catch
(Throwable var17) { if (var5 == null) { var5 = var17; } else if (var5 != var17) { var5.addSuppressed(var17);
} throw var5; } } catch (Exception var18) { var18.printStackTrace(); JOptionPane.showMessageDialog(this,
"Erro ao conectar ao banco de dados.", "Erro", 0); } } }

```