Classe Conexao.java

```
import java.sql.*;
import javax.swing.JOptionPane;

public class Conexao {
    public Connection conectaDB(){
        Connection conn = null;

        try {
            String url = "jdbc:mysql://localhost:3306/fatec?user=root&password=";
            conn = DriverManager.getConnection(url);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage());

        }
        return conn;
    }
}
```

```
/*-------------------------------------------------------------------------------------------------------------*/

//DTO - Transferência de Objetos
```

Classe ClientesDTO.java

```
public class ClientesDTO {
    private String nome_cliente, endereco_cliente, cidade_cliente, profissao_cliente;
    private int id_clientes;  // Criação da variável id_clientes, utilizada na listagem;

    public String getNome_cliente() {
        return nome_cliente;
    }

    public void setNome_cliente(String nome_cliente) {
        this.nome_cliente = nome_cliente;
    }

    public String getEndereco_cliente() {
        return endereco_cliente;
    }

    public void setEndereco_cliente(String endereco_cliente) {
        this.endereco_cliente = endereco_cliente;
    }

    public String getCidade_cliente() {
        return cidade_cliente;
    }

    public void setCidade_cliente(String cidade_cliente) {
        this.cidade_cliente = cidade_cliente;
    }

    public String getProfissao_cliente() {
        return profissao_cliente;
    }

    public void setProfissao_cliente(String profissao_cliente) {
        this.profissao_cliente = profissao_cliente;
    }
```

```
/*
     Criação dos métodos getId_clientes() e setId_clientes() utilizados na listagem.
*/
    public int getId_clientes() {
        return id_clientes;
    }

    public void setId_clientes(int id_clientes) {
        this.id_clientes = id_clientes;
    }
}
```

```
/*-------------------------------------------------------------------------------------------------------------*/

//DAO - Acesso ao Objeto
```

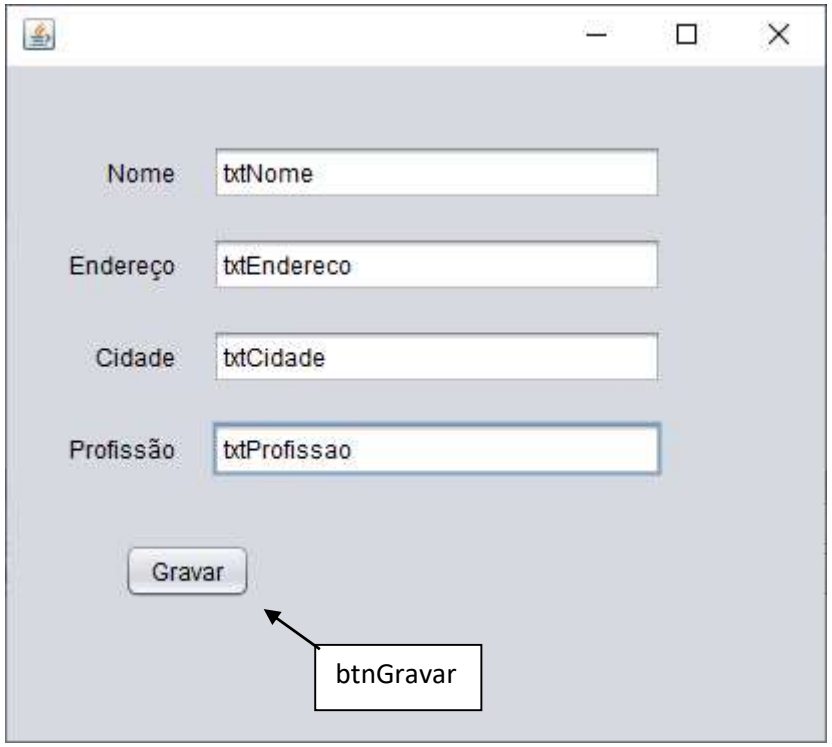
Classe ClientesDAO.java

```
import java.sql.*;
import javax.swing.JOptionPane;

public class ClientesDAO {
    Connection conn;
    PreparedStatement pstm;
    ResultSet rs;
    ArrayList<ClientesDTO> lista = new ArrayList<>();

     public void cadastrarCliente(ClientesDTO objClientesDTO){
        String sql = "insert into clientes (nome, endereco, cidade, profissao) values (?,?,?,?)";

        conn = new Conexao().conectaDB();
        try {
            pstm = conn.prepareStatement(sql);
            pstm.setString(1, objClientesDTO.getNome_cliente());
            pstm.setString(2, objClientesDTO.getEndereco_cliente());
            pstm.setString(3, objClientesDTO.getCidade_cliente());
            pstm.setString(4, objClientesDTO.getProfissao_cliente());

            pstm.execute();
            pstm.close();

        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }



/*
     O Método ListarClientes() foi criado para retornar o vetor lista que contém todos os registros selecionado na
     tabela do Banco de Dados.
*/
```

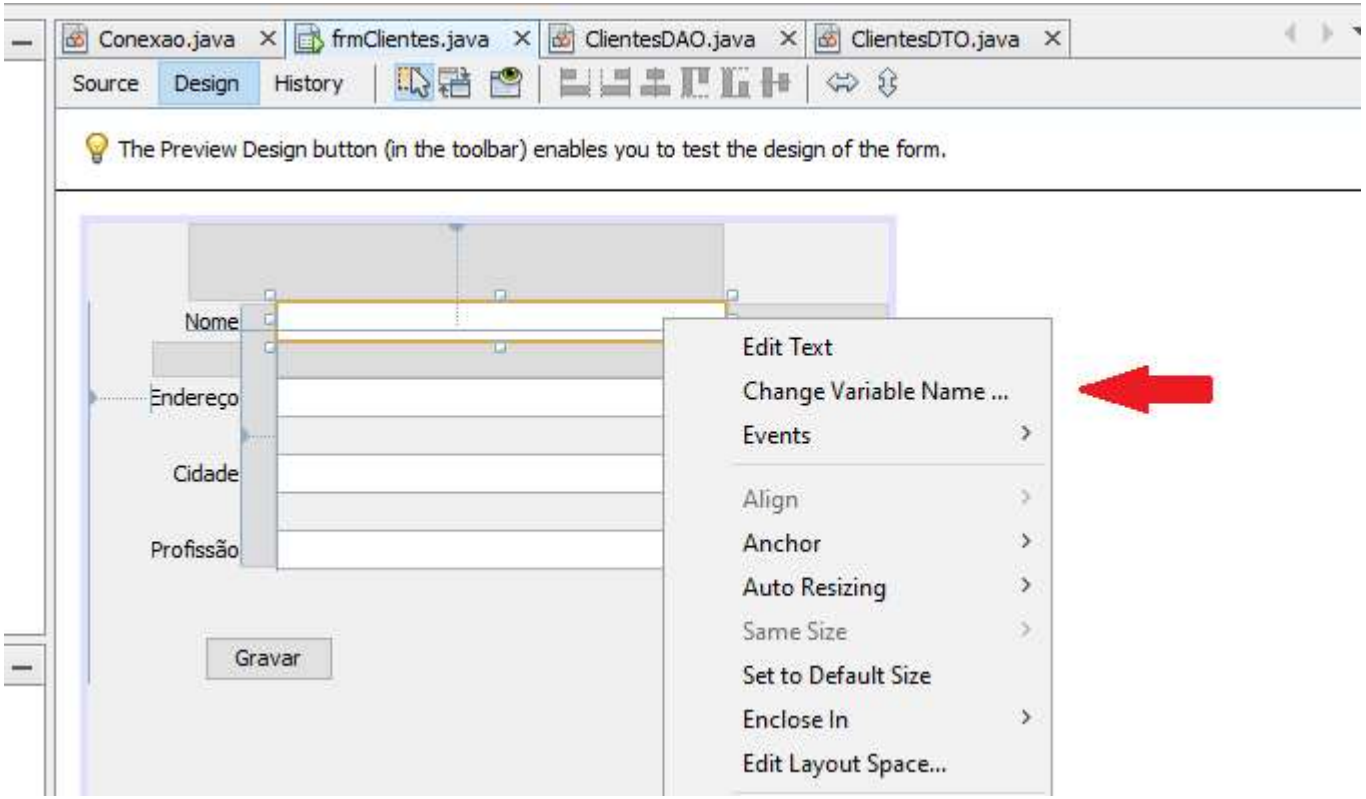
```
public ArrayList<ClientesDTO> ListarClientes() {
    String sql="select * from clientes";
    conn = new Conexao().conectaDB();

    try {
        pstm = conn.prepareStatement(sql);
        rs = pstm.executeQuery();

        while(rs.next()){
            ClientesDTO objClientesDTO = new ClientesDTO();
            objClientesDTO.setId_clientes(rs.getInt("id_clientes"));
            objClientesDTO.setNome_cliente(rs.getString("nome"));
            objClientesDTO.setEndereco_cliente(rs.getString("endereco"));
            objClientesDTO.setCidade_cliente(rs.getString("cidade"));
            objClientesDTO.setProfissao_cliente(rs.getString("profissao"));

            lista.add(objClientesDTO);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
    return lista;
}


public void alterarCliente(ClientesDTO objClientesDTO){
    String sql = "update clientes set nome = ?, endereco = ?, cidade = ?, profissao = ? where id_clientes = ?";

    conn = new Conexao().conectaDB();
    try {
        pstm = conn.prepareStatement(sql);
        pstm.setString(1, objClientesDTO.getNome_cliente());
        pstm.setString(2, objClientesDTO.getEndereco_cliente());
        pstm.setString(3, objClientesDTO.getCidade_cliente());
        pstm.setString(4, objClientesDTO.getProfissao_cliente());
        pstm.setInt(5, objClientesDTO.getId_clientes());

        pstm.execute();
        pstm.close();

    } catch (Exception e) {
```

```
            JOptionPane.showMessageDialog(null, e);
        }

    }

    public void excluirCliente(ClientesDTO objClientesDTO){
        String sql = "delete from clientes where id_clientes = ?";

        conn = new Conexao().conectaDB();
        try {
            pstm = conn.prepareStatement(sql);
            pstm.setInt(1, objClientesDTO.getId_clientes());

            pstm.execute();
            pstm.close();

        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }

    }
}
```

```
/*-------------------------------------------------------------------------------------------------------------*/

//Interface Gráfica; Formulário

Classe frmClientes.java
```

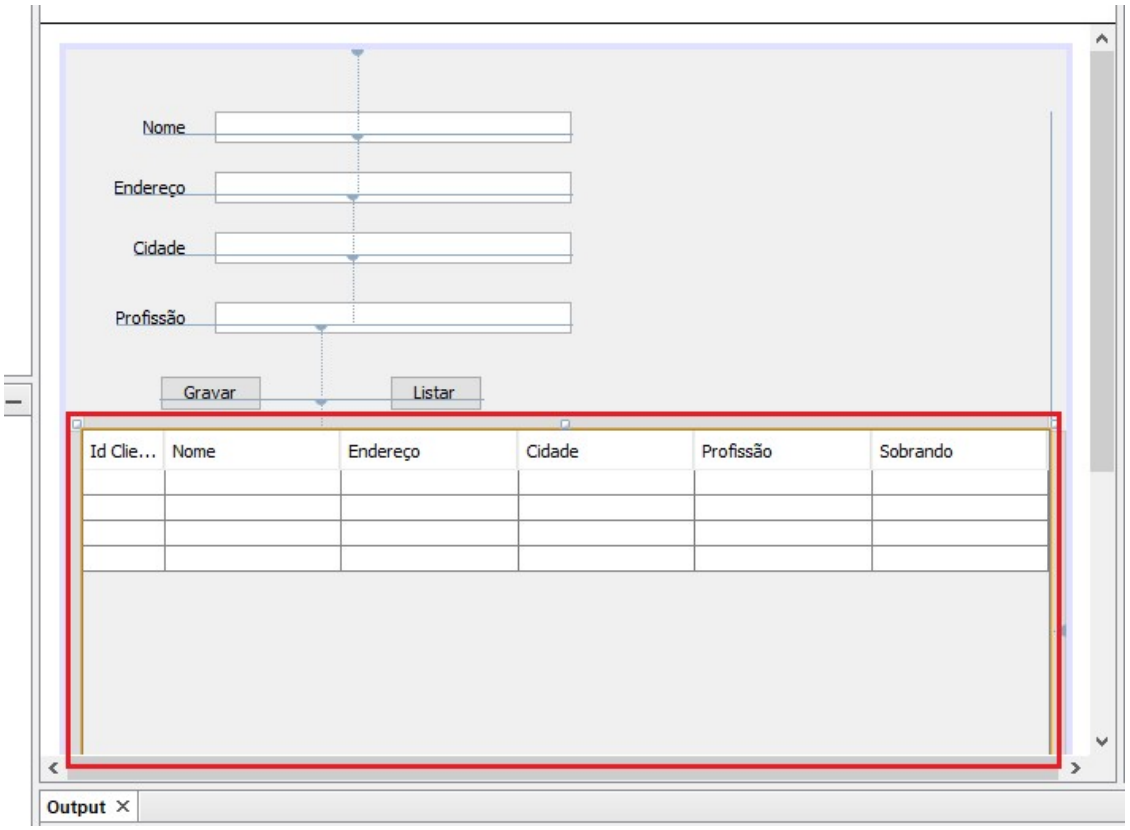


Para modificar o nome da variável do componente **JTextField**, clique com o botão direito do mouse sobre o componente desejado e selecione a opção **Change Variable Name**..., como mostra a imagem abaixo.



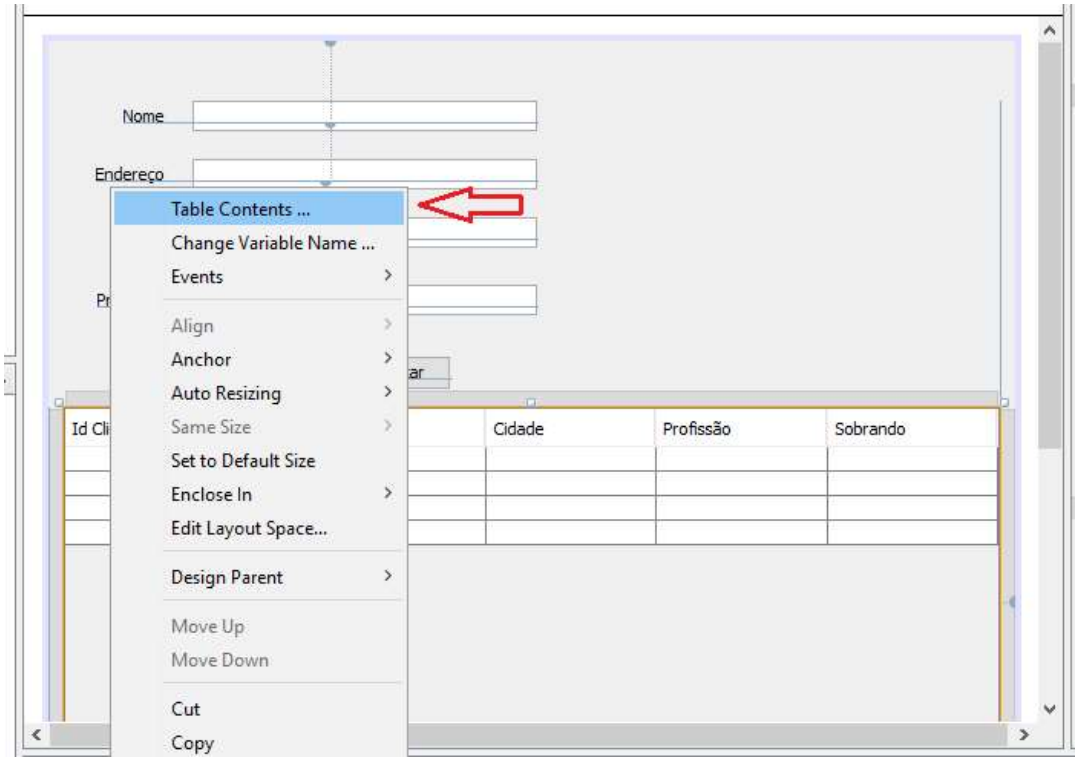

```
/*
      Programação do botão btnGravar. Para escrever a programação abaixo, clique duas vezes sobre o botão btnGravar.
*/
    private void btnGravarActionPerformed(java.awt.event.ActionEvent evt) {
        String nome, endereco, cidade, profissao;

        nome = txtNome.getText();
        endereco = txtEndereco.getText();
        cidade = txtCidade.getText();
        profissao = txtProfissao.getText();

        ClientesDTO objclientesdto = new ClientesDTO();
        objclientesdto.setNome_cliente(nome);
        objclientesdto.setEndereco_cliente(endereco);
```

```
        objclientesdto.setCidade_cliente(cidade);
        objclientesdto.setProfissao_cliente(profissao);

        ClientesDAO objclientesdao = new ClientesDAO();
        objclientesdao.cadastrarCliente(objclientesdto);

        listarClientes(); /* Chamada ao método que lista os clientes cadastrados no banco. Este método atualiza o
                             componente TabelaClientes (JTable)  */

    }


/*
      Inserção do componente JTable: tabelaClientes no formulário frmClientes. Poderia ser criado um novo formulário
      para conter apenas o JTable: tabelaClientes.
*/
```



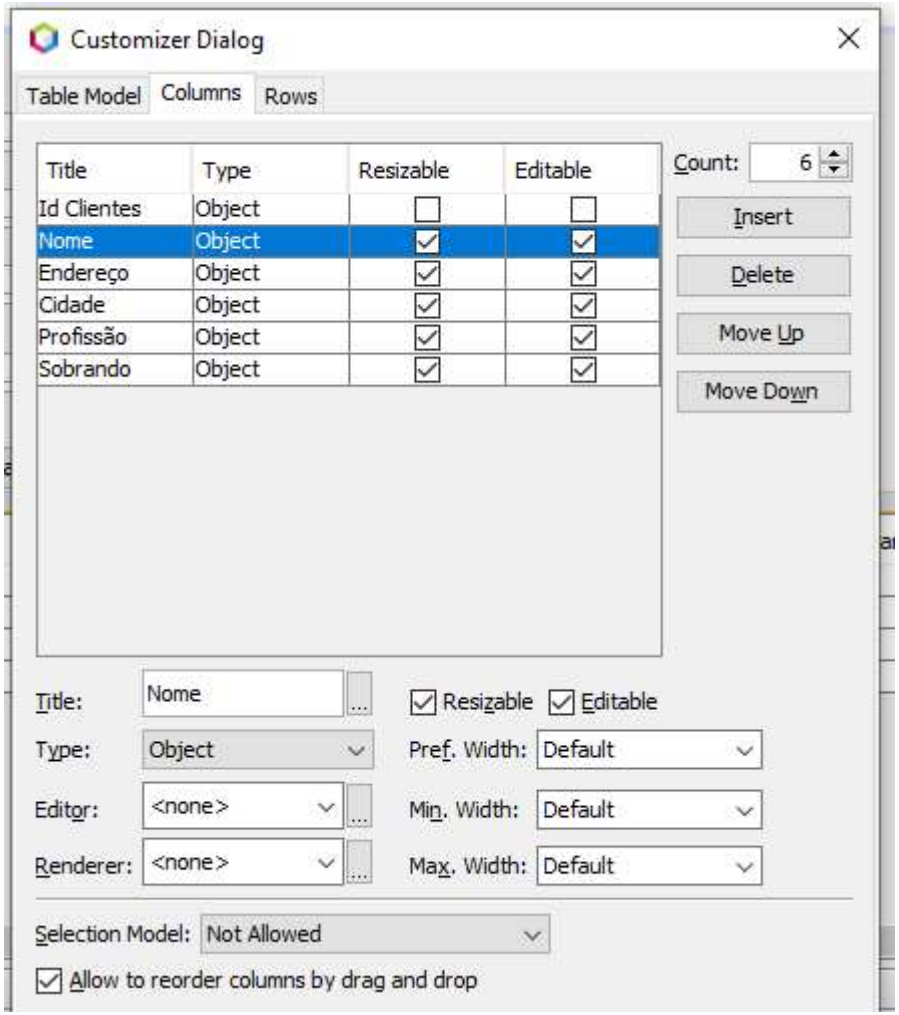
```
/*
     Criação do Método listarClientes() que lista os clientes cadastrados no banco. Este método atualiza o
     componente TabelaClientes (JTable).
     Este método deve ser criado no final da programação do formulário frmClientes.java.
*/

    private void listarClientes(){
        try {
            ClientesDAO objClientesDAO = new ClientesDAO();

            DefaultTableModel model = (DefaultTableModel) TabelaClientes.getModel();
            model.setNumRows(0);
            ArrayList<ClientesDTO> lista = objClientesDAO.ListarClientes();

            for(int num=0; num < lista.size(); num++){
                model.addRow(new Object[]{
                    lista.get(num).getId_clientes(),
                    lista.get(num).getNome_cliente(),
                    lista.get(num).getEndereco_cliente(),
                    lista.get(num).getCidade_cliente(),
                    lista.get(num).getProfissao_cliente()
                });
            }
        } catch (Exception e) {
        }
    }
```

/*
Para modificar o cabeçalho das colunas do componente **JTable: TabelaClientes,** clique com botão direito do mouse sobre o JTable e selecione a opção **Table Contents...**
*/



/*
Em seguida, clique na aba **Columns** da janela **Customize Dialog.**
*/

```
/*
     Clique sobre o titulo da coluna e informe o valor desejado.
*/
```



```
// Continuação da classe frmClientes.java

private void CarregarDados(){
        int setar = TabelaClientes.getSelectedRow();

        txtId_clientes.setText(TabelaClientes.getModel().getValueAt(setar, 0).toString());
        txtNome.setText(TabelaClientes.getModel().getValueAt(setar, 1).toString());
        txtEndereco.setText(TabelaClientes.getModel().getValueAt(setar, 2).toString());
        txtCidade.setText(TabelaClientes.getModel().getValueAt(setar, 3).toString());
        txtProfissao.setText(TabelaClientes.getModel().getValueAt(setar, 4).toString());
        txtNome.requestFocus();
     }

    private void CadastrarClientes(){

//Esta programação veio do botão btnGravar

        String nome, endereco, cidade, profissao;

        nome = txtNome.getText();
        endereco = txtEndereco.getText();
        cidade = txtCidade.getText();
        profissao = txtProfissao.getText();

        ClientesDTO objclientesdto = new ClientesDTO();
        objclientesdto.setNome_cliente(nome);
        objclientesdto.setEndereco_cliente(endereco);
        objclientesdto.setCidade_cliente(cidade);
        objclientesdto.setProfissao_cliente(profissao);

        ClientesDAO objclientesdao = new ClientesDAO();
        objclientesdao.cadastrarCliente(objclientesdto);
    }

    private void LimparCampos(){
        txtId_clientes.setText(""); // Limpa o valor o JTextField
        txtNome.setText("");
        txtEndereco.setText("");
        txtCidade.setText("");
        txtProfissao.setText("");
```

```
    txtNome.requestFocus(); // Seta o foco para
}

private void AlterarCliente(){
    int id_cliente;
    String nome_cliente, endereco_cliente, cidade_cliente, profissao_cliente;

    id_cliente = Integer.parseInt(txtId_clientes.getText());
    nome_cliente = txtNome.getText();
    endereco_cliente = txtEndereco.getText();
    cidade_cliente = txtCidade.getText();
    profissao_cliente = txtProfissao.getText();

    ClientesDTO objClientesDTO = new ClientesDTO();
    objClientesDTO.setId_clientes(id_cliente);
    objClientesDTO.setNome_cliente(nome_cliente);
    objClientesDTO.setEndereco_cliente(endereco_cliente);
    objClientesDTO.setCidade_cliente(cidade_cliente);
    objClientesDTO.setProfissao_cliente(profissao_cliente);

    ClientesDAO objClientesDAO = new ClientesDAO();
    objClientesDAO.alterarCliente(objClientesDTO);
}

private void ExcluirCliente(){
    int id_cliente;
    // String nome_cliente, endereco_cliente, cidade_cliente, profissao_cliente;

    id_cliente = Integer.parseInt(txtId_clientes.getText());

    ClientesDTO objClientesDTO = new ClientesDTO();
    objClientesDTO.setId_clientes(id_cliente);

    ClientesDAO objClientesDAO = new ClientesDAO();
    objClientesDAO.excluirCliente(objClientesDTO);
}
```

```
/*

     Última versão, completa, dor arquivo frmClientes.java;

*/



import java.util.ArrayList;
import javax.swing.table.DefaultTableModel;

public class frmClientes extends javax.swing.JFrame {

    public frmClientes() {
        initComponents();
        listarClientes();
    }

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

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        txtNome = new javax.swing.JTextField();
        txtEndereco = new javax.swing.JTextField();
        txtCidade = new javax.swing.JTextField();
        txtProfissao = new javax.swing.JTextField();
        btnGravar = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        TabelaClientes = new javax.swing.JTable();
        btnListar = new javax.swing.JButton();
        jLabel5 = new javax.swing.JLabel();
        txtId_clientes = new javax.swing.JTextField();
        btnCarregarDados = new javax.swing.JButton();
        btnLimpar = new javax.swing.JButton();
        btnAlterar = new javax.swing.JButton();
        btnExcluir = new javax.swing.JButton();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText("Nome");

jLabel2.setText("Endereço");

jLabel3.setText("Cidade");

jLabel4.setText("Profissão");

btnGravar.setText("Gravar");
btnGravar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnGravarActionPerformed(evt);
    }
});

TabelaClientes.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null},
        {null, null, null, null, null}
    },
    new String [] {
        "Id Clientes", "Nome", "Endereço", "Cidade", "Profissão"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, true, true, true, true
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
jScrollPane1.setViewportView(TabelaClientes);
if (TabelaClientes.getColumnModel().getColumnCount() > 0) {
    TabelaClientes.getColumnModel().getColumn(0).setResizable(false);
    TabelaClientes.getColumnModel().getColumn(0).setPreferredWidth(10);
}
```

```
btnListar.setText("Listar Dados");
btnListar.setMaximumSize(new java.awt.Dimension(109, 23));
btnListar.setMinimumSize(new java.awt.Dimension(109, 23));
btnListar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnListarActionPerformed(evt);
    }
});

jLabel5.setText("Código");

txtId_clientes.setEnabled(false);

btnCarregarDados.setText("Carregar Dados");
btnCarregarDados.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCarregarDadosActionPerformed(evt);
    }
});

btnLimpar.setText("Liimpar Dados");
btnLimpar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLimparActionPerformed(evt);
    }
});

btnAlterar.setText("Alterar");
btnAlterar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAlterarActionPerformed(evt);
    }
});

btnExcluir.setText("Excluir");
btnExcluir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnExcluirActionPerformed(evt);
    }
});
```

```
        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(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 611, Short.MAX_VALUE)
                .addContainerGap())
            .addGroup(layout.createSequentialGroup()
                .addGap(31, 31, 31)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel4)
                    .addComponent(jLabel3)
                    .addComponent(jLabel2)
                    .addComponent(jLabel1)
                    .addComponent(jLabel5))
                .addGap(18, 18, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(txtNome, javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
                    .addComponent(txtEndereco)
                    .addComponent(txtCidade)
                    .addComponent(txtProfissao)
                    .addComponent(txtId_clientes, javax.swing.GroupLayout.PREFERRED_SIZE, 57,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(btnAlterar)
                    .addComponent(btnGravar)
                    .addComponent(btnExcluir))
                .addGap(177, 177, 177))
            .addGroup(layout.createSequentialGroup()
                .addGap(17, 17, 17)
                .addComponent(btnListar, javax.swing.GroupLayout.PREFERRED_SIZE, 109,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnCarregarDados)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnLimpar, javax.swing.GroupLayout.PREFERRED_SIZE, 109,
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()
                .addGap(3, 3, 3)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel5)
                    .addComponent(txtId_clientes, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(btnGravar))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(txtNome, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(btnAlterar))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(txtEndereco, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(btnExcluir))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(txtCidade, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(24, 24, 24)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel4)
                    .addComponent(txtProfissao, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(26, 26, 26)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(btnListar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(btnCarregarDados)
                    .addComponent(btnLimpar))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 444, Short.MAX_VALUE)
```

```
                .addContainerGap())
        );

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

    private void btnGravarActionPerformed(java.awt.event.ActionEvent evt) {
        CadastrarClientes();
        listarClientes();
        LimparCampos();
    }

    private void btnListarActionPerformed(java.awt.event.ActionEvent evt) {
        listarClientes();
    }

    private void btnCarregarDadosActionPerformed(java.awt.event.ActionEvent evt) {
        CarregarDados();
    }

    private void btnLimparActionPerformed(java.awt.event.ActionEvent evt) {
        LimparCampos();
    }

    private void btnAlterarActionPerformed(java.awt.event.ActionEvent evt) {
        AlterarCliente();
        listarClientes();
        LimparCampos();
    }

    private void btnExcluirActionPerformed(java.awt.event.ActionEvent evt) {
        ExcluirCliente();
        listarClientes();
        LimparCampos();
    }

    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(frmClientes.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(frmClientes.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(frmClientes.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(frmClientes.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new frmClientes().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JTable TabelaClientes;
    private javax.swing.JButton btnAlterar;
    private javax.swing.JButton btnCarregarDados;
    private javax.swing.JButton btnExcluir;
    private javax.swing.JButton btnGravar;
    private javax.swing.JButton btnLimpar;
    private javax.swing.JButton btnListar;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
```

```
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextField txtCidade;
private javax.swing.JTextField txtEndereco;
private javax.swing.JTextField txtId_clientes;
private javax.swing.JTextField txtNome;
private javax.swing.JTextField txtProfissao;
// End of variables declaration

private void listarClientes(){
    try {
        ClientesDAO objClientesDAO = new ClientesDAO();

        DefaultTableModel model = (DefaultTableModel) TabelaClientes.getModel();
        model.setNumRows(0);
        ArrayList<ClientesDTO> lista = objClientesDAO.ListarClientes();

        for(int num=0; num < lista.size(); num++){
            model.addRow(new Object[]{
                lista.get(num).getId_clientes(),
                lista.get(num).getNome_cliente(),
                lista.get(num).getEndereco_cliente(),
                lista.get(num).getCidade_cliente(),
                lista.get(num).getProfissao_cliente()
            });
        }
    } catch (Exception e) {
    }
}

private void CarregarDados(){
    int setar = TabelaClientes.getSelectedRow();

    txtId_clientes.setText(TabelaClientes.getModel().getValueAt(setar, 0).toString());
    txtNome.setText(TabelaClientes.getModel().getValueAt(setar, 1).toString());
    txtEndereco.setText(TabelaClientes.getModel().getValueAt(setar, 2).toString());
    txtCidade.setText(TabelaClientes.getModel().getValueAt(setar, 3).toString());
    txtProfissao.setText(TabelaClientes.getModel().getValueAt(setar, 4).toString());
    txtNome.requestFocus();
```

```
 }

private void CadastrarClientes(){
    String nome, endereco, cidade, profissao;

    nome = txtNome.getText();
    endereco = txtEndereco.getText();
    cidade = txtCidade.getText();
    profissao = txtProfissao.getText();

    ClientesDTO objclientesdto = new ClientesDTO();
    objclientesdto.setNome_cliente(nome);
    objclientesdto.setEndereco_cliente(endereco);
    objclientesdto.setCidade_cliente(cidade);
    objclientesdto.setProfissao_cliente(profissao);

    ClientesDAO objclientesdao = new ClientesDAO();
    objclientesdao.cadastrarCliente(objclientesdto);
}

private void LimparCampos(){
    txtId_clientes.setText(""); // Limpa o valor o JTextField
    txtNome.setText("");
    txtEndereco.setText("");
    txtCidade.setText("");
    txtProfissao.setText("");
    txtNome.requestFocus(); // Seta o foco para
}

private void AlterarCliente(){
    int id_cliente;
    String nome_cliente, endereco_cliente, cidade_cliente, profissao_cliente;

    id_cliente = Integer.parseInt(txtId_clientes.getText());
    nome_cliente = txtNome.getText();
    endereco_cliente = txtEndereco.getText();
    cidade_cliente = txtCidade.getText();
    profissao_cliente = txtProfissao.getText();

    ClientesDTO objClientesDTO = new ClientesDTO();
    objClientesDTO.setId_clientes(id_cliente);
```

```
        objClientesDTO.setNome_cliente(nome_cliente);
        objClientesDTO.setEndereco_cliente(endereco_cliente);
        objClientesDTO.setCidade_cliente(cidade_cliente);
        objClientesDTO.setProfissao_cliente(profissao_cliente);

        ClientesDAO objClientesDAO = new ClientesDAO();
        objClientesDAO.alterarCliente(objClientesDTO);
    }

    private void ExcluirCliente(){
        int id_cliente;
        // String nome_cliente, endereco_cliente, cidade_cliente, profissao_cliente;

        id_cliente = Integer.parseInt(txtId_clientes.getText());

        ClientesDTO objClientesDTO = new ClientesDTO();
        objClientesDTO.setId_clientes(id_cliente);

        ClientesDAO objClientesDAO = new ClientesDAO();
        objClientesDAO.excluirCliente(objClientesDTO);
    }
}
```