

Características das Linguagens de Programação I

Conceúdo: Projeto de aplicação - parte III

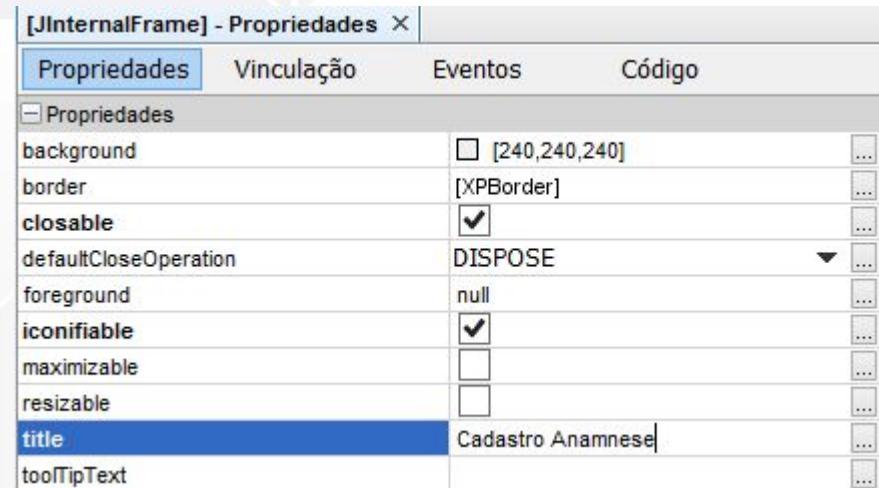
Prof. Dsc. Giomar Sequeiros
giomar@eng.uerj.br



GUI Anamnese

Criação da GUI Anamnese(1)

- No pacote apresentação, criar um novo **JInternalFrame**, com o nome fmAnamnese.
- Modificar a propriedade closable e iconifiable do **JInternalFrame** fmPaciente para true. Finalmente acrescente um título



- Finalmente ligar ao menu principal (fmPrincipal)

Criação da GUI Anamnese(1)

- Acrescente componentes de interface conforme a figura abaixo:

cbPacientes
cbPsicologos

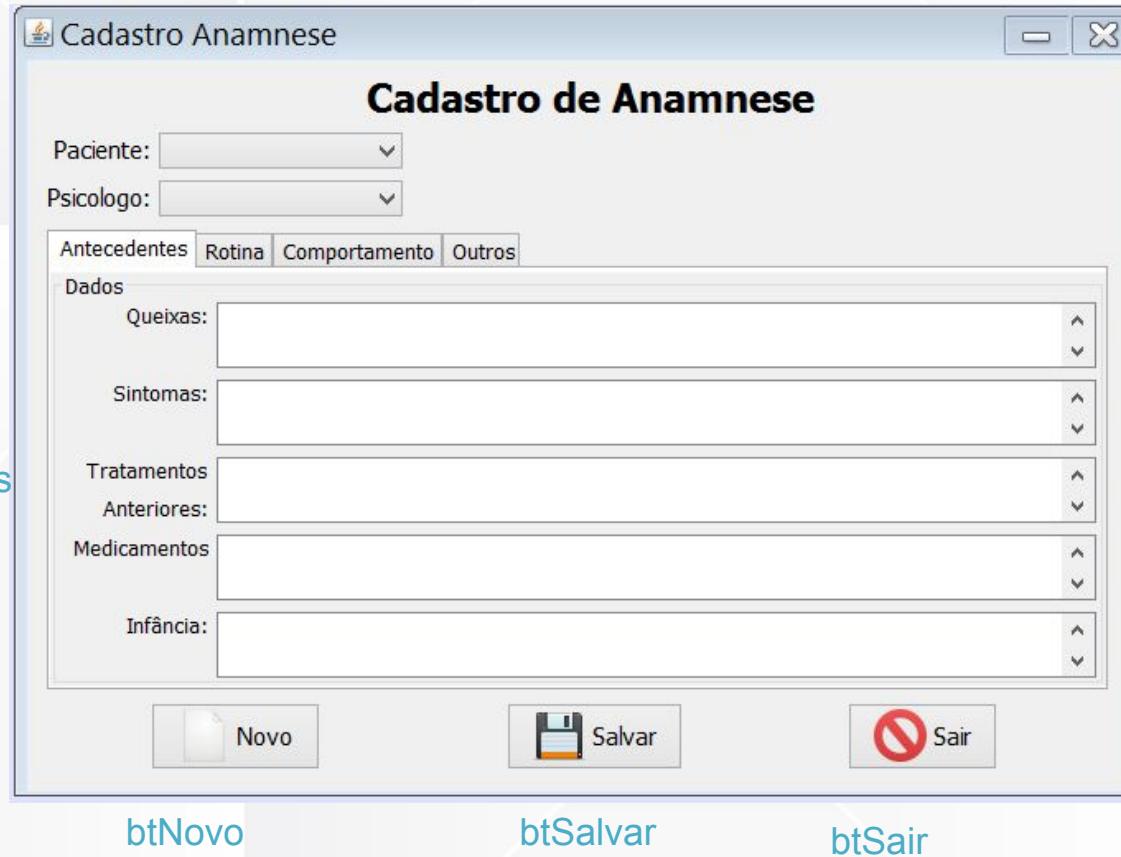
txtQueixas

txtSintomas

txtTratamentosAnteriores

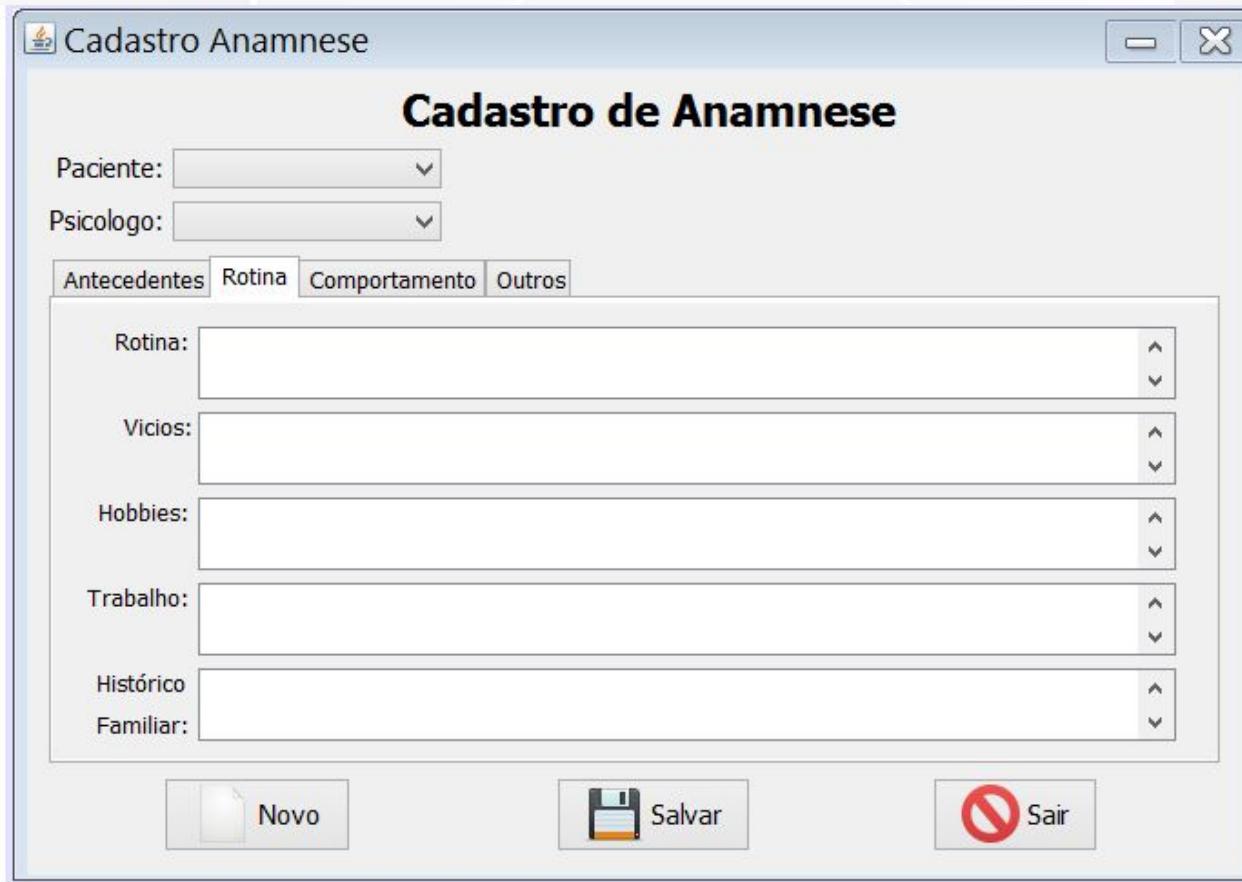
txtMedicamentos

txtInfancia



Criação da GUI Anamnese(3)

- Acrescente componentes de interface conforme a figura abaixo:



txtRotina

txtVicios

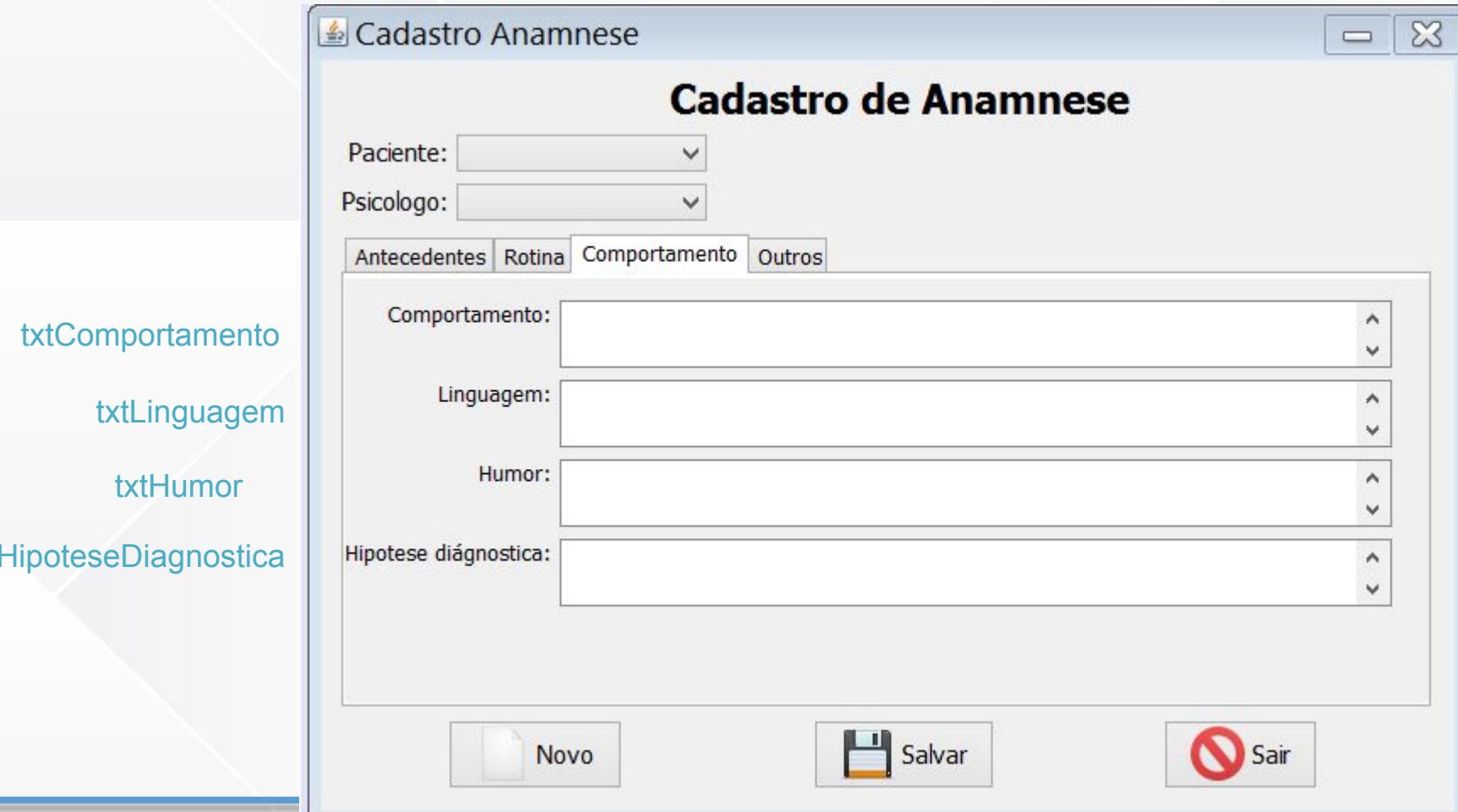
txtHobbies

txtTrabalho

txtHistoricoFamiliar

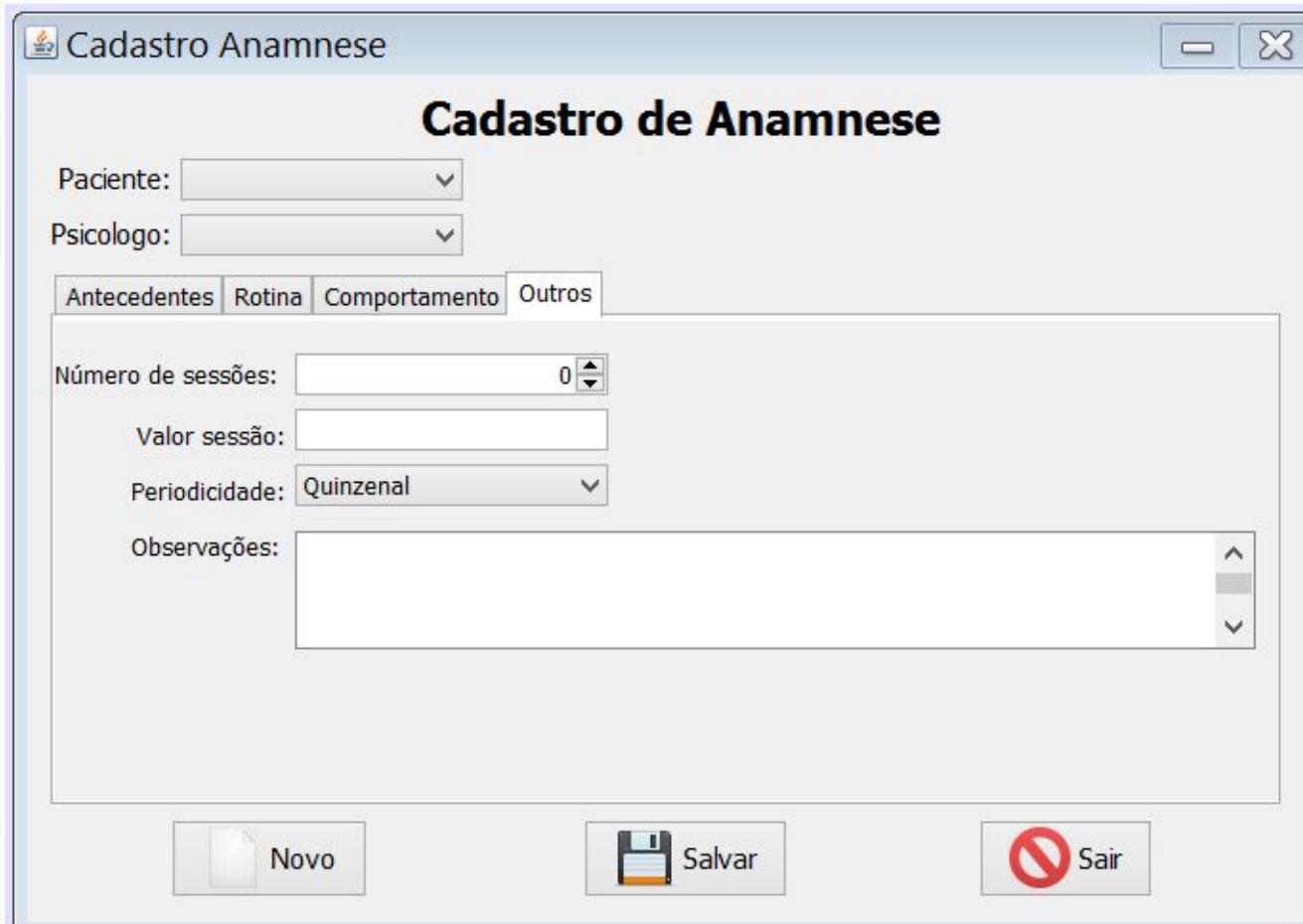
Criação da GUIAnamnese(4)

- Acrescente componentes de interface conforme a figura abaixo:



Criação da GUI Anamnese(5)

- Acrescente componentes de interface conforme a figura abaixo:



txtNumeroSessoes

txtValorSessao

cbPeriodicidade

txtObservacoes

Criação da GUI Anamnese(6)

- Adicione 3 pacientes ao banco de dados a GUI correspondente
- Adicione 2 psicólogos usando o script sql mostrado abaixo:

```
use dbconsultorio;

insert into psicologo(nome, cpf, crp, telefone, login, senha)
values('Psicólogo teste','12345678912', '987654', '(21)987987987', 'user','user123');
```

Criação da GUI Anamnese(7)

- Implemente o método `listarTodos()` na classe **PacienteDAO**. Esse método será invocado quando a GUI anamnese for iniciada.

```
try {
    List<Paciente> pacientes = new ArrayList<Paciente>();
    PreparedStatement stmt = this.connection.prepareStatement("select * from paciente");
    ResultSet rs = stmt.executeQuery();

    while (rs.next()) {
        // criando o objeto Contato
        Paciente paciente = new Paciente();
        paciente.setIdPaciente(rs.getInt("idPaciente"));
        paciente.setNome(rs.getString("nome"));
        paciente.setCpf(rs.getString("cpf"));

        // montando a data através do Calendar
        Calendar data = Calendar.getInstance();
        data.setTime(rs.getDate("data_nascimento"));
        paciente.setData_nascimento(data);

        paciente.setSexo(rs.getString("sexo"));
        paciente.setEndereco(rs.getString("endereco"));
        paciente.setTelefone(rs.getString("telefone"));
        paciente.setFoto(rs.getString("foto"));
        paciente.setPlano_saude(rs.getString("plano_saude"));
        paciente.setObservacoes(rs.getString("observacoes"));
        data.setTime(rs.getDate("dataCadastro"));
        paciente.setDataCadastro(data);
        // adicionando o objeto à lista
        pacientes.add(paciente);
    }
    rs.close();
    stmt.close();
    return pacientes;
} catch (SQLException e) {
    throw new RuntimeException(e);
}
```

Criação da GUI Anamnese(8)

- Implemente o método `listarTodos()` na classe **PsicologoDAO**. Esse método será invocado quando a GUI anamnese for iniciada.

```
@Override  
public List<Psicologo> listarTodos() {  
    try {  
        List<Psicologo> psicologos = new ArrayList<Psicologo>();  
        PreparedStatement stmt = this.connection.  
        prepareStatement("select * from psicologo");  
        ResultSet rs = stmt.executeQuery();  
  
        while (rs.next()) {  
            // criando o objeto Contato  
            Psicologo psicologo = new Psicologo();  
            psicologo.setIdPsicologo(rs.getInt("idpsicologo"));  
            psicologo.setNome(rs.getString("nome"));  
            psicologo.setCpf(rs.getString("cpf"));  
            psicologo.setCrp(rs.getString("crp"));  
            psicologo.setTelefone(rs.getString("telefone"));  
            psicologo.setLogin(rs.getString("login"));  
            psicologo.setSenha(rs.getString("senha"));  
  
            // adicionando o objeto à lista  
            psicologos.add(psicologo);  
        }  
        rs.close();  
        stmt.close();  
        return psicologos;  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```

Criação da GUI Anamnese(9)

- No evento InternalFrameOpened da janela fmAnamnese adicione o código a seguir:

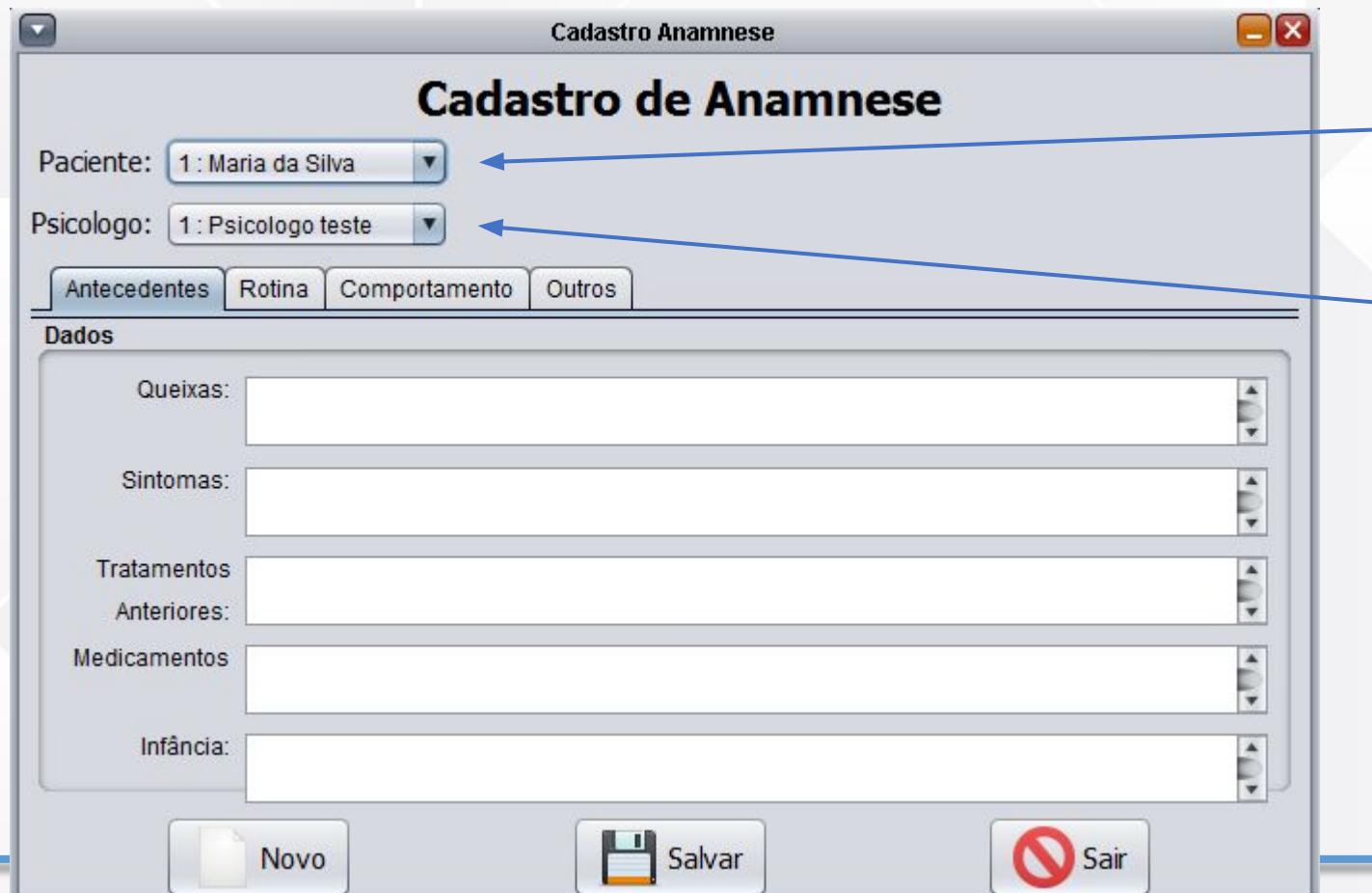
Carrega os pacientes cadastrados

Carrega os psicólogos cadastrados

```
private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {  
    try{  
        //Carregar pacientes dentro do jcomboBox  
        IPacienteDAO dao = new PacienteDAO();  
        List<Paciente> lista = dao.listarTodos();  
  
        DefaultComboBoxModel modelo = new DefaultComboBoxModel ();  
        for(Paciente paciente: lista) {  
            modelo.addElement(paciente.getIdPaciente()+" : "+paciente.getNome());  
        }  
        cbPacientes.setModel(modelo);  
  
        //Carregar psicologos dentro do jcomboBox  
        IPsicologoDAO dao2 = new PsicologoDAO();  
        List<Psicologo> lista2 = dao2.listarTodos();  
  
        DefaultComboBoxModel modelo2 = new DefaultComboBoxModel ();  
        for(Psicologo psicologo: lista2) {  
            modelo2.addElement(psicologo.getIdPsicologo()+" : "+psicologo.getNome());  
        }  
        cbPsicologos.setModel(modelo2);  
  
    }  
    catch(Exception e){  
        System.out.println(e.toString());  
        JOptionPane.showMessageDialog(null, "Houve um erro ao carregar os dados...");  
    }  
}
```

Criação da GUI Anamnese(10)

- Após executar o programa os dados devem ser recuperados na interface conforme a figura abaixo:



Os pacientes cadastrados no banco de dados serão carregados no combo box.

Os psicólogos cadastrados no banco de dados serão carregados no combo box.

Criação da GUI Anamnese(10)

- No evento **actionPerformed** do botão salvar escreva o código abaixo

```
private void btSalvarActionPerfomed(java.awt.event.ActionEvent evt) {  
    int valor = JOptionPane.showConfirmDialog(null, "Tem certeza que deseja salvar?", "Sistema Consultório Médicos", 1);  
    if(valor==0){  
        //recuperar os dados inseridos  
        int idpaciente=Integer.parseInt(cbPacientes.getSelectedItem().toString().split(" : ")[0]);  
        int idpsicologo=Integer.parseInt(cbPsicologos.getSelectedItem().toString().split(" : ")[0]);  
  
        Anamnese anamnese = new Anamnese();  
        anamnese.setQueixas(txtQueixas.getText());  
        anamnese.setSintomas(txtSintomas.getText());  
        anamnese.setTratamentos_anteriores(txtTratamentosAnteriores.getText());  
        anamnese.setMedicamentos(txtMedicamentos.getText());  
        anamnese.setInfancia(txtInfancia.getText());  
        ...  
        anamnese.setIdPaciente(idpaciente);  
        anamnese.setIdPsicologo(idpsicologo);  
        // gravamos os dados  
        IAnamneseDAO dao = new AnamneseDAO();  
        dao.adiciona(anamnese);  
        JOptionPane.showMessageDialog(null, "Os dados foram gravados");  
    }  
}
```

Completar o código

Atividades

- Crie uma GUI funcional que permita salvar psicólogos
- Crie uma GUI funcional que permita salvar sessões.

Listar Paciente

DAO – Método listarTodos(1)

- Implementamos o método listarTodos() na classe PacienteDAO:

```
public List<Paciente> listarTodos() {
    try {
        List<Paciente> pacientes = new ArrayList<Paciente>();
        PreparedStatement stmt = this.connection.prepareStatement("select * from paciente");
        ResultSet rs = stmt.executeQuery();

        while (rs.next()) {
            // criando o objeto Contato
            Paciente paciente = new Paciente();
            paciente.setId(rs.getLong("id"));
            paciente.setNome(rs.getString("nome"));
            paciente.setNome_mae(rs.getString("nome_mae"));

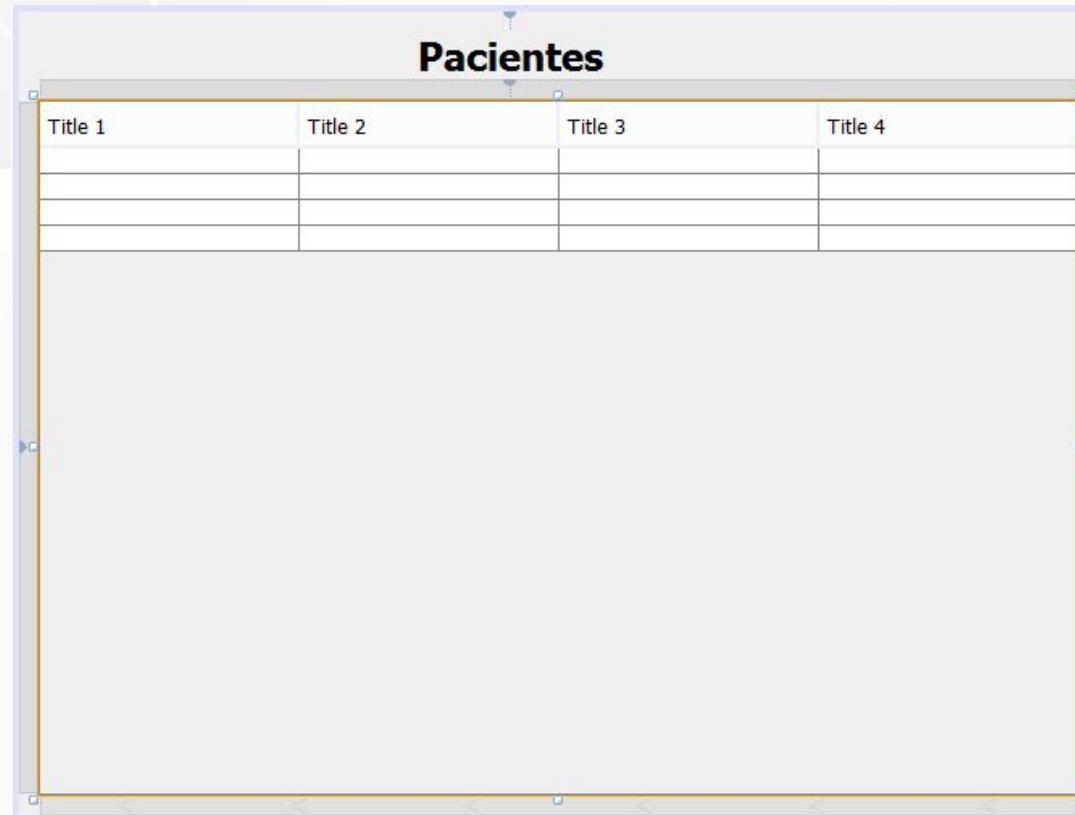
            // montando a data através do Calendar
            Calendar data = Calendar.getInstance();
            data.setTime(rs.getDate("data_nascimento"));
            paciente.setData_nascimento(data);

            paciente.setEstado_civil(rs.getString("estado_civil"));
            paciente.setCor(rs.getString("cor"));
            paciente.setTelefone(rs.getString("telefone"));
            paciente.setFotografia(rs.getString("fotografia"));

            // adicionando o objeto à lista
            pacientes.add(paciente);
        }
        rs.close();
        stmt.close();
        return pacientes;
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```

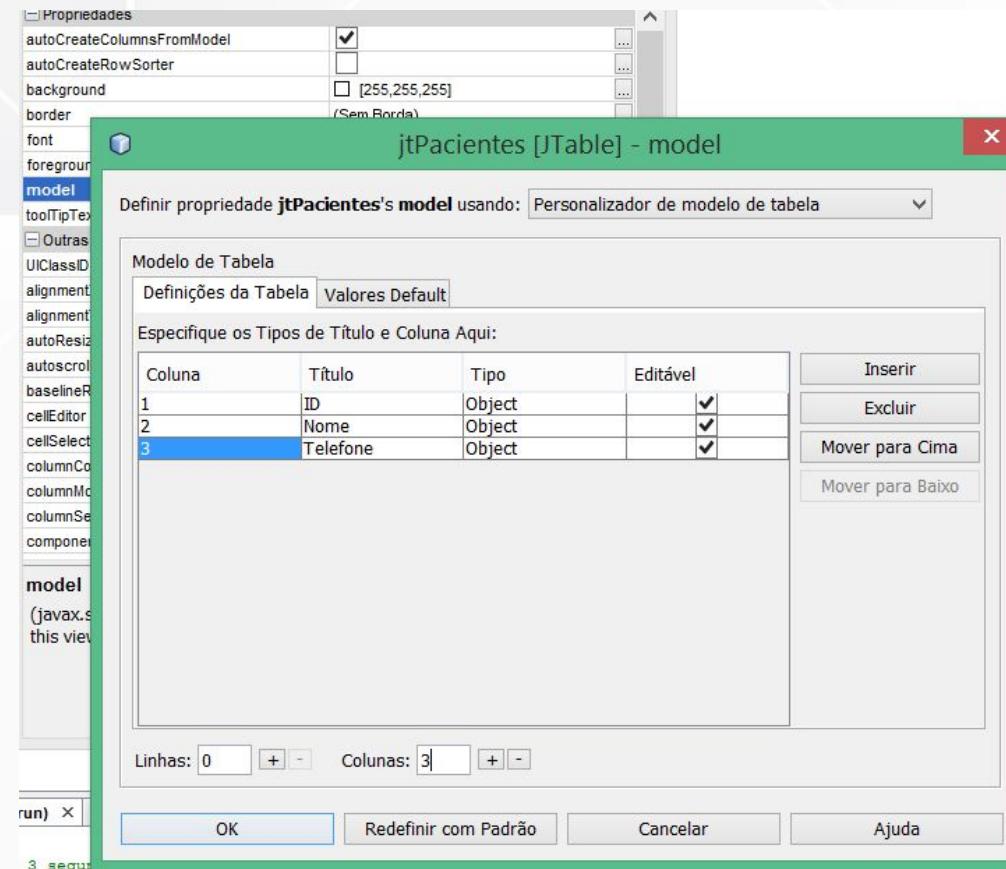
DAO – Método listarTodos(2)

- No pacote apresentação criamos um novo formulário (JFrame) chamado de **fmListarPaciente**.
- Adicionamos um rótulo e um componente JTable conforme a figura



DAO – Método listarTodos(3)

- Modificamos o nome de variável do JTable para jtPacientes e modificamos a propriedade **model** da tabela conforme a figura abaixo:



DAO – Método listarTodos(4)

- O resultado será:



DAO – Método listarTodos(5)

- No evento **windowsOpened** do formulário jtPacientes escrevemos o código abaixo:

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {
    //Chamamos ao método listar da classe DAO
    IPacienteDAO dao=new PacienteDAO();
    List<Paciente> lista = dao.listarTodos();

    //Criando um modelo para a tabela
    DefaultTableModel modelo=new DefaultTableModel();
    modelo.addColumn("ID");
    modelo.addColumn("Nome");
    modelo.addColumn("Telefone");

    //adicionamos os items à tabela
    for(Paciente paciente: lista)
    {
        modelo.addRow(new Object[]{paciente.getId(), paciente.getNome(), paciente.getTelefone()});
    }
    //Dar o modelo à tabela paciente
    jtPacientes.setModel(modelo);
}
```

DAO – Método listarTodos(6)

- Testando:



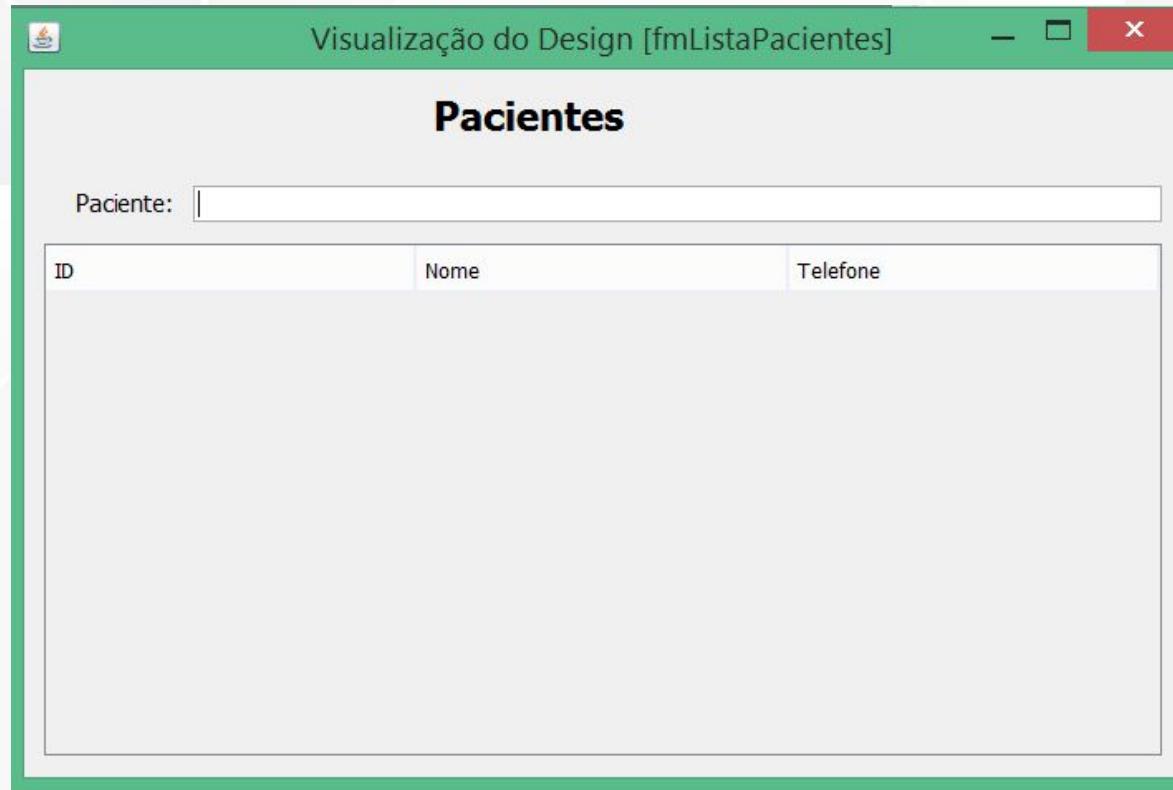
A screenshot of a Java Swing application window titled "Pacientes". The window has a green header bar with standard window controls (minimize, maximize, close). The main area is titled "Pacientes" in bold black font. Below the title is a table with three columns: "ID", "Nome", and "Telefone". A single row of data is visible, containing the values 1, Maria da Silva, and 987654321.

ID	Nome	Telefone
1	Maria da Silva	987654321

O componente mostrará todos os registros da tabela pacientes armazenados no banco de dados

DAO – Método listarTodos(7)

- Atividade Permitindo buscas: Adicione um método no DAO para permitir fazer buscas por alguma palavra chave.



Referências

BIBLIOGRAFIA BÁSICA:

- DEITEL, Harvey M.. Java : como programar. 10^a ed. São Paulo: Pearson - Prentice Hall, 2017.

- BORATTI, Isaías Camilo. Programação Orientada a Objetos em Java : Conceitos Fundamentais de Programação Orientada a Objetos. 1^a ed. Florianópolis: VisualBooks, 2007.

- SIERRA, Kathy; BATES, Bert. Use a Cabeça! Java. 2^a ed. Rio de Janeiro: Alta Books, 2007.

