

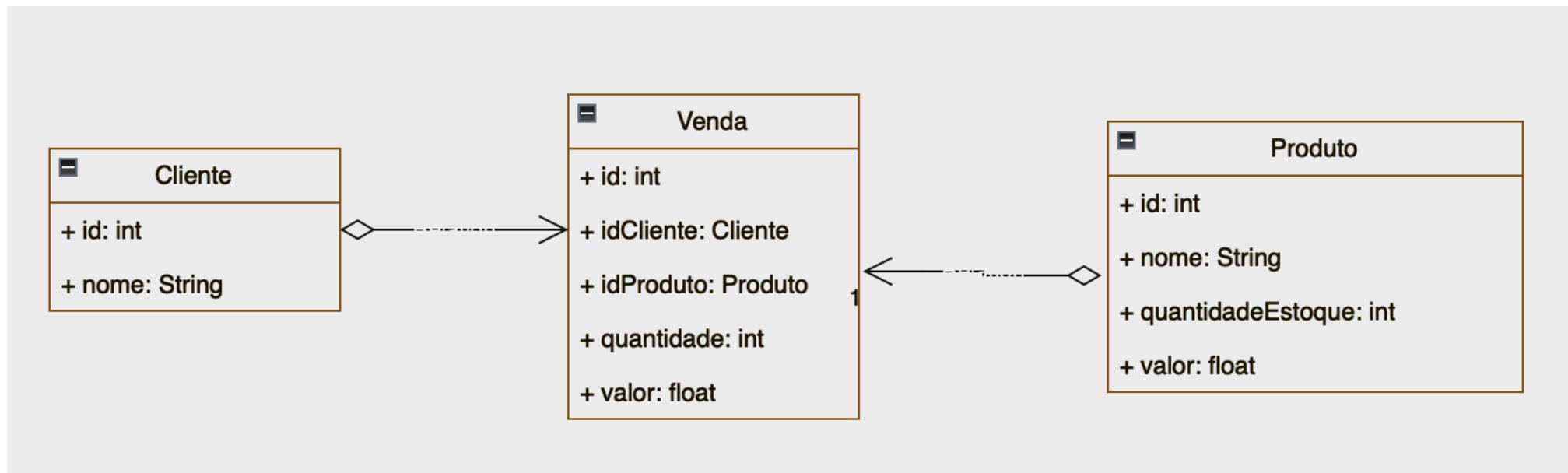
Venda Projetos - Java

Prof. Victor Peres

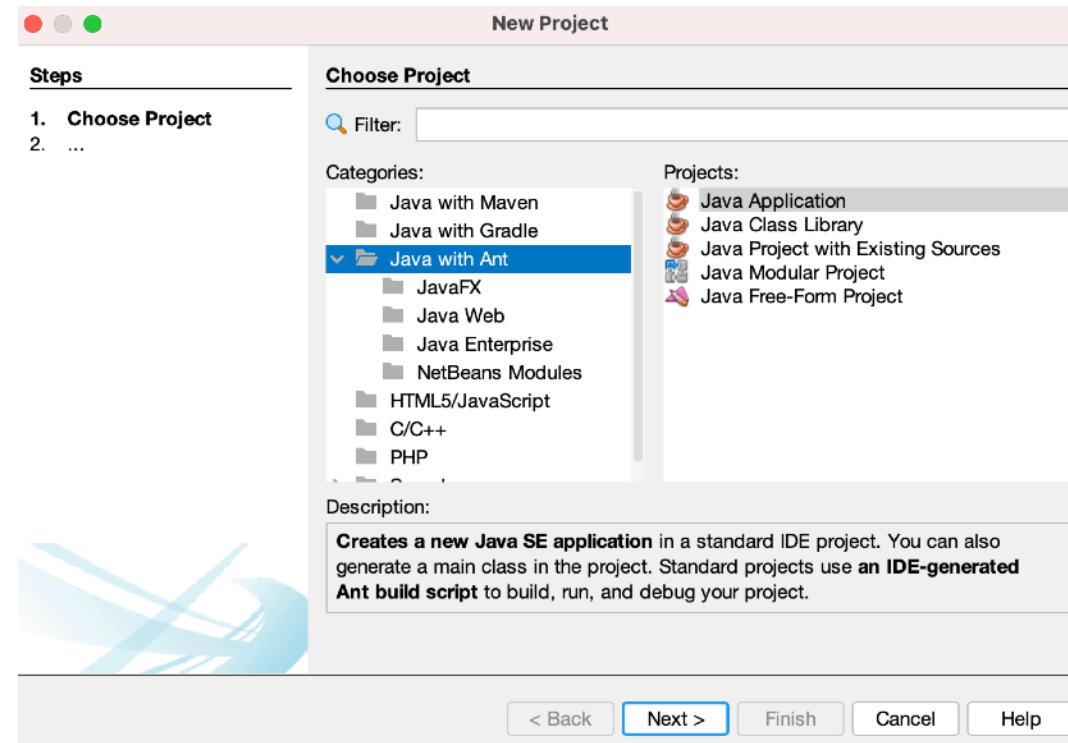
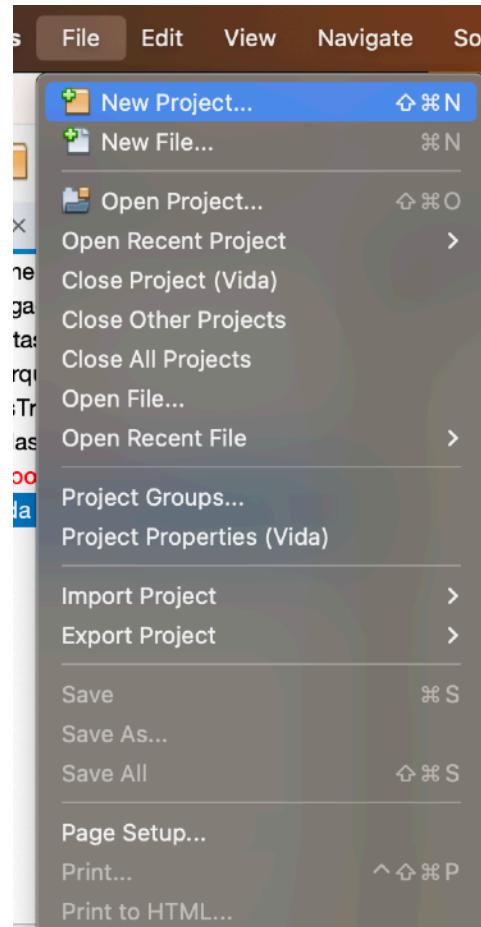
victor.peres@vacaria.ifrs.edu.br



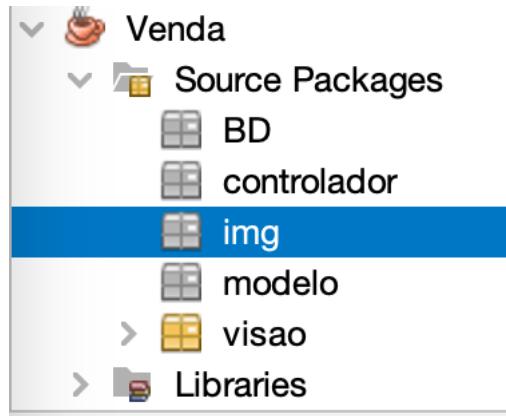
Projeto - Venda



Criando Projeto - Netbeans



Criando Projeto - Netbeans

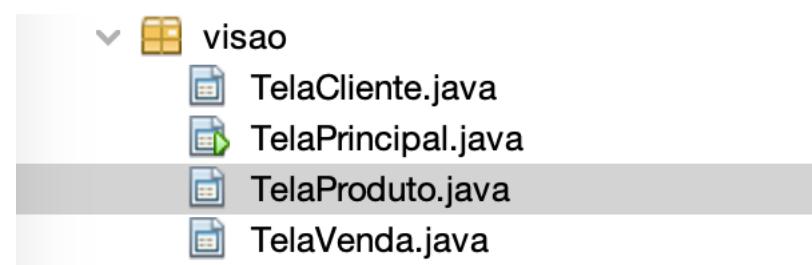
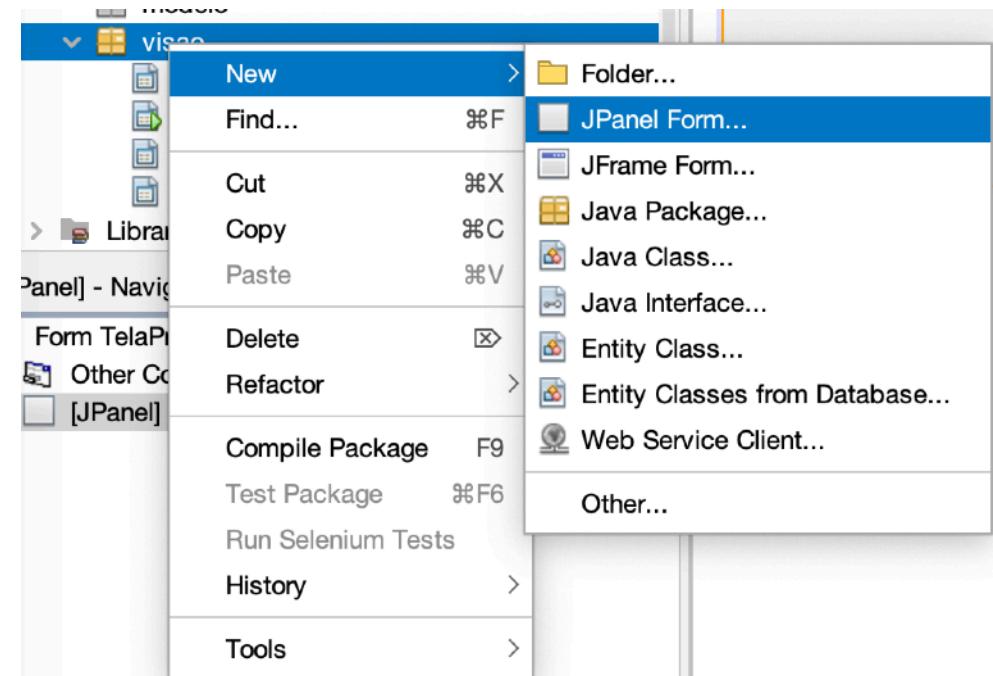
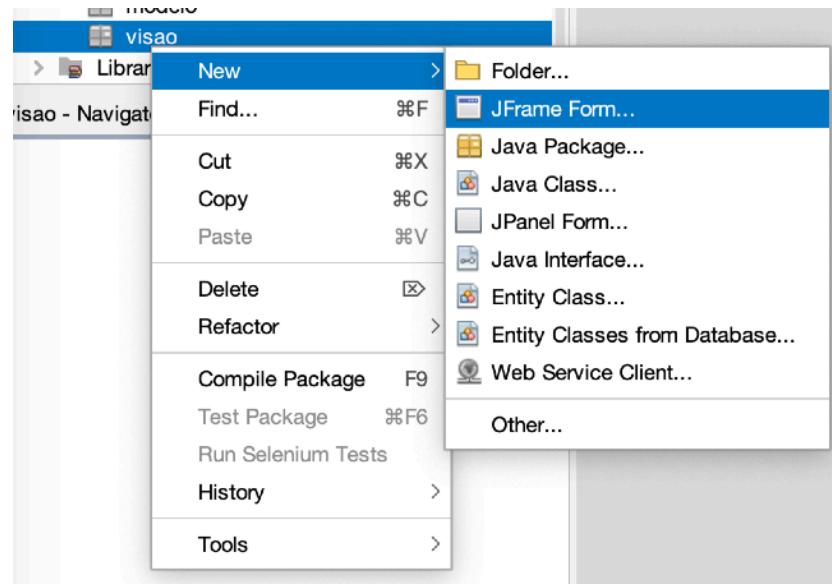


No pacote visão, vamos criar:

- JFrame com o nome de TelaPrincipal
- JPanel com o nome TelaCliente
- JPanel com o nome TelaVenda
- JPanel com o nome TelaProduto



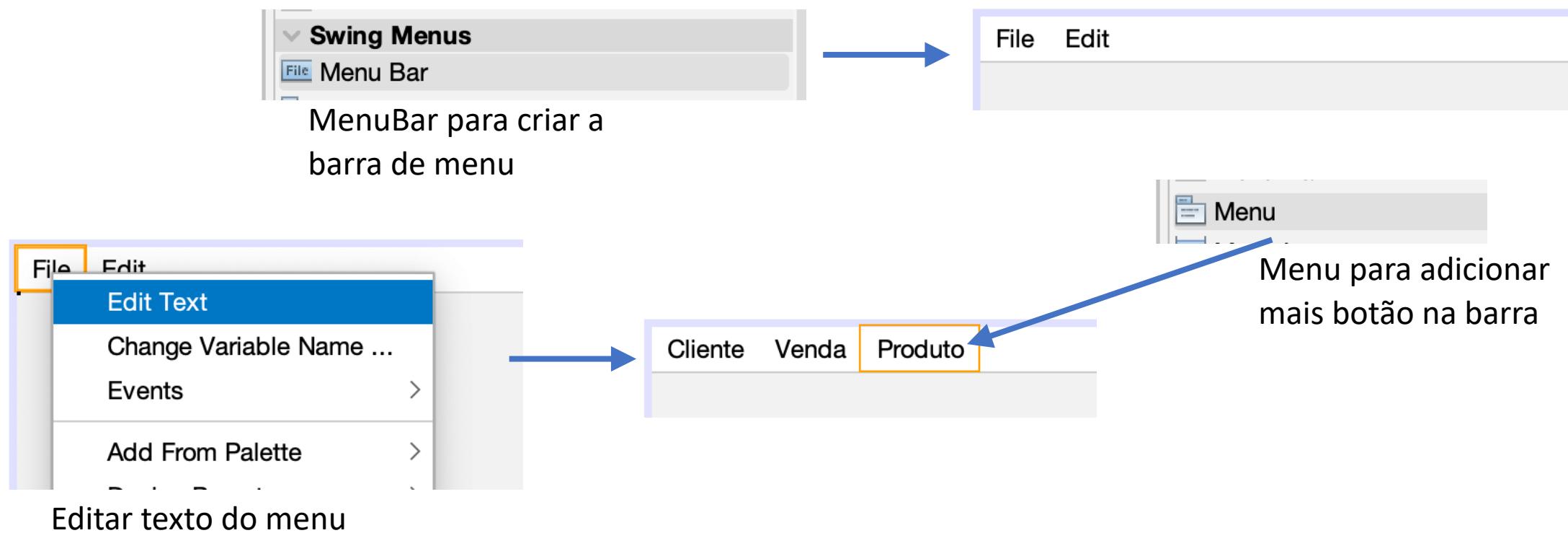
Criando Telas



TelaPrincipal:

- **MenuBar**
- **DesktopPane**

Criando TelaPrincipal

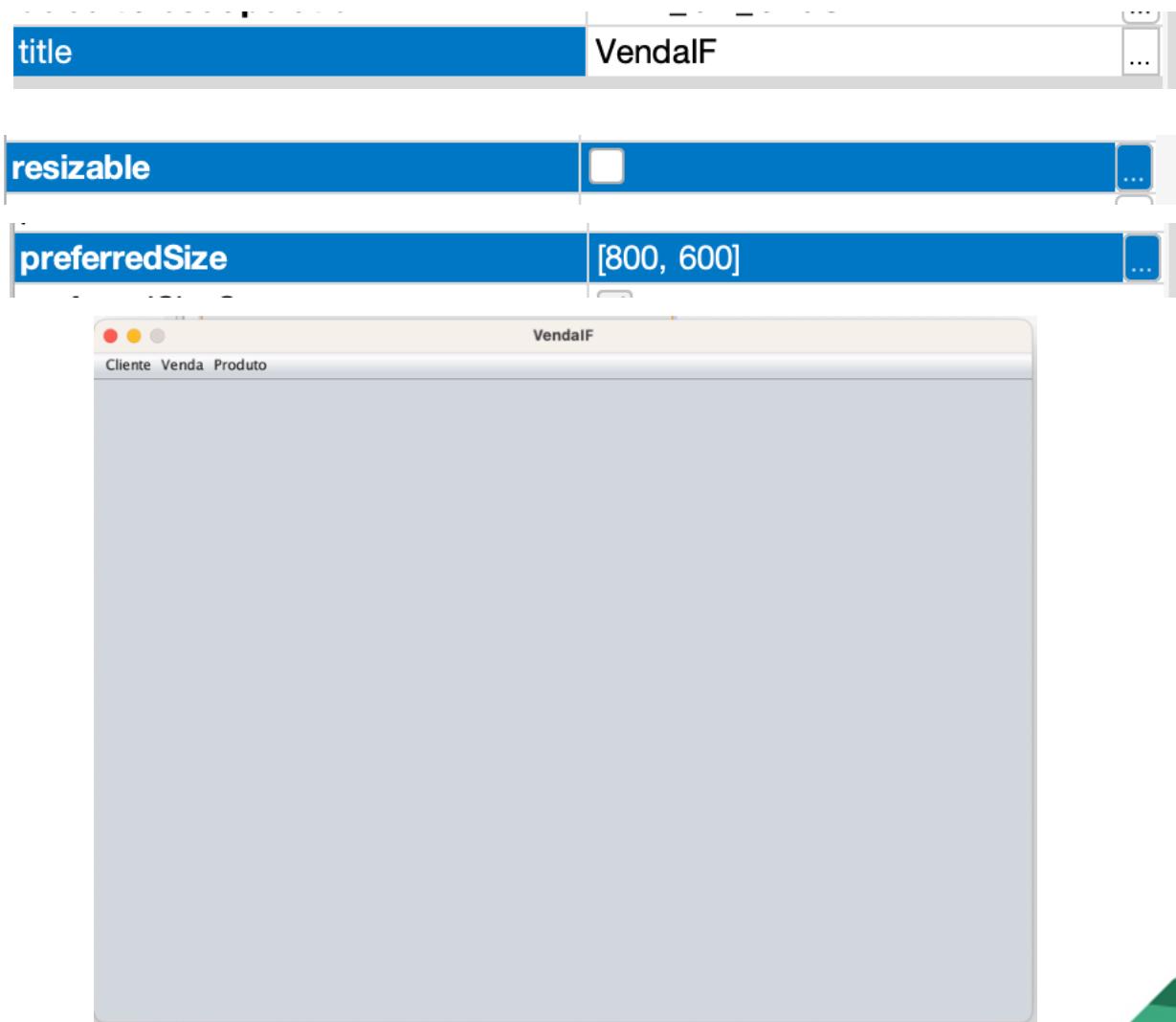
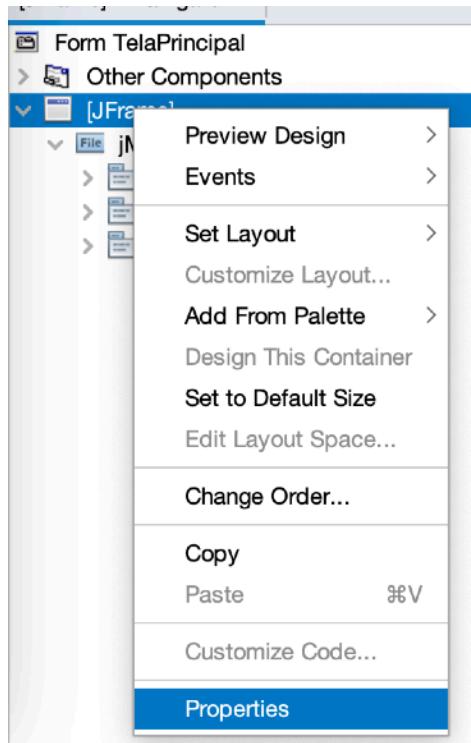


TelaPrincipal:

- **MenuBar**
- **DesktopPane**

Criando TelaPrincipal

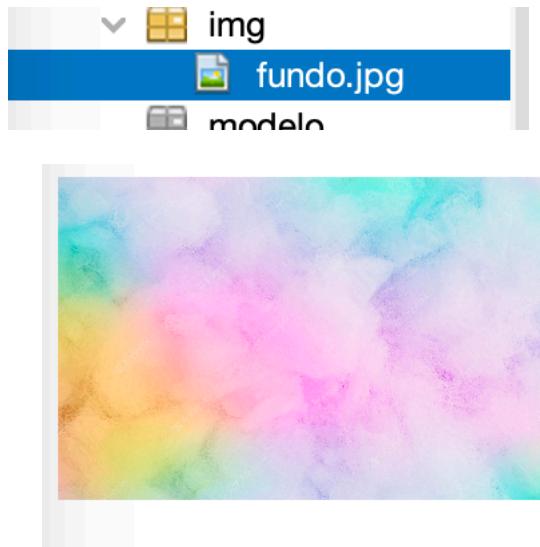
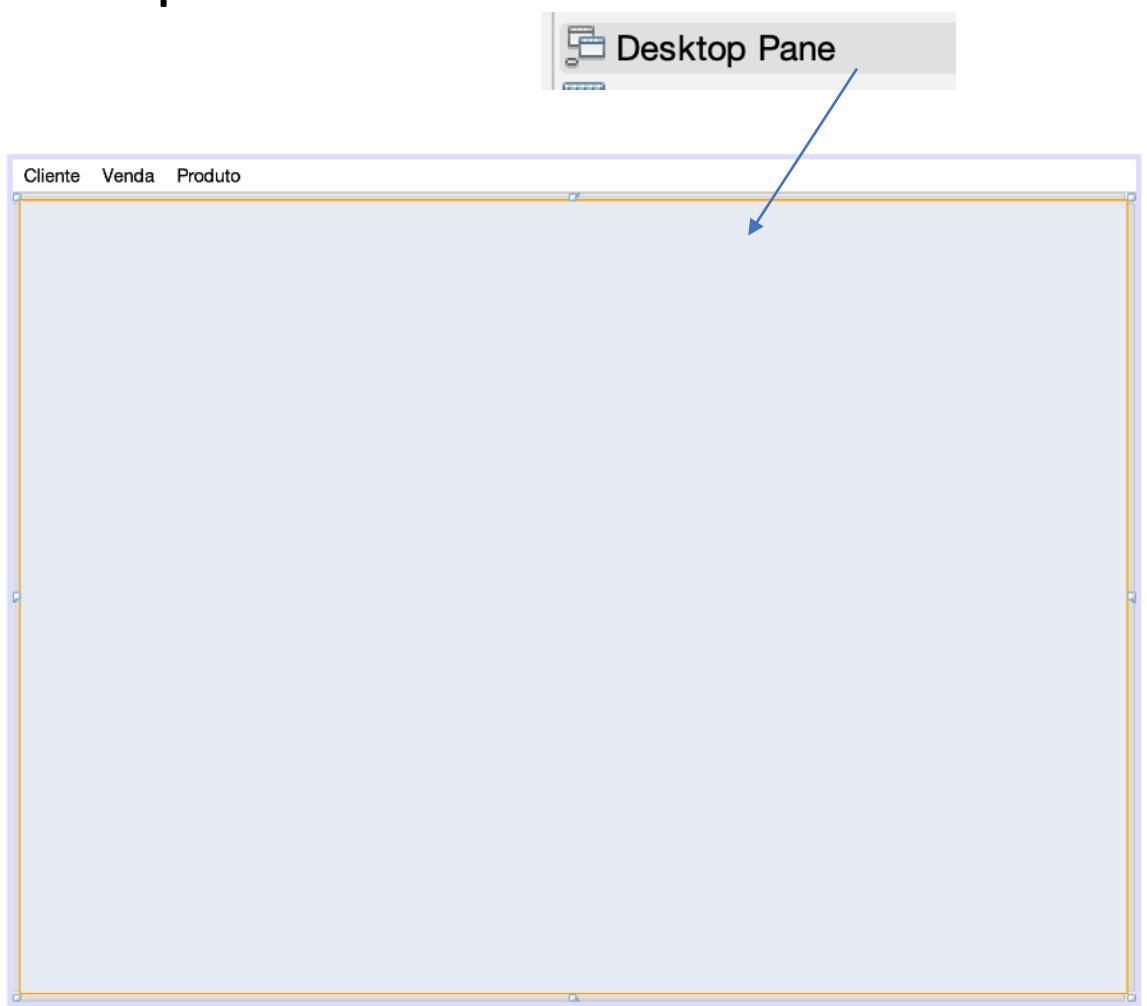
Propriedade do JFrame



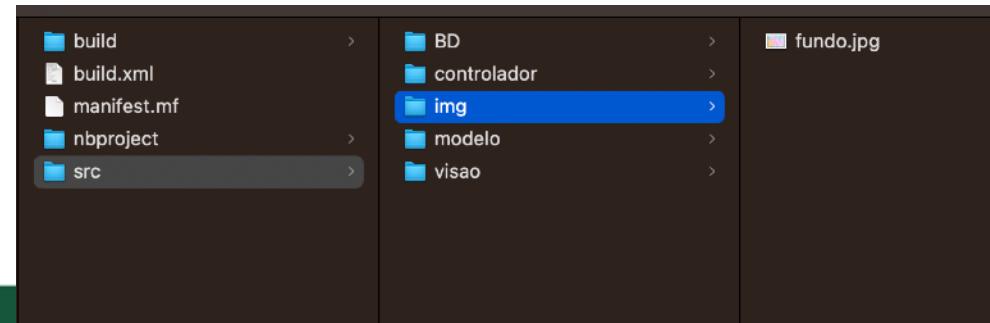
TelaPrincipal:

- **MenuBar**
- **DesktopPane**

Criando TelaPrincipal



Inserir a imagem no pacote img. (A imagem pode ser encontrada no Moodle). Para inserir abra a pasta do projeto, pasta src e depois para img.

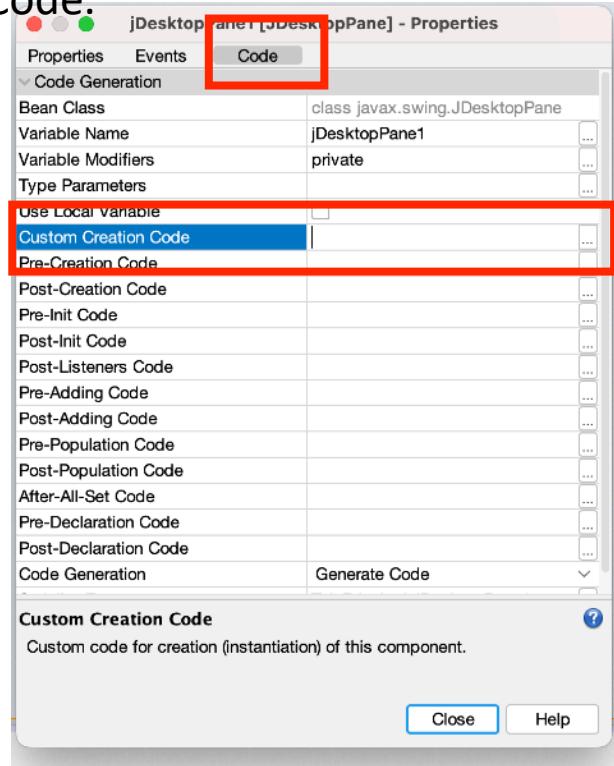


TelaPrincipal:

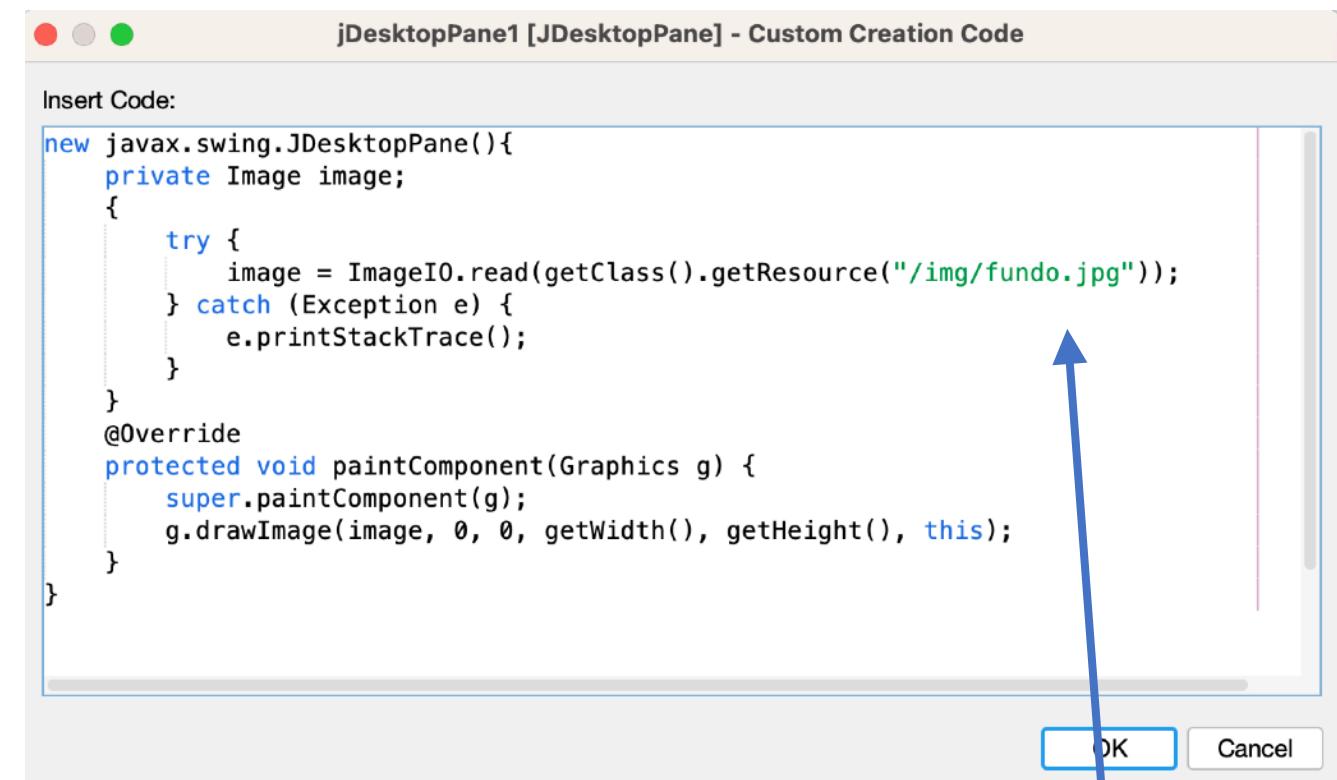
- **MenuBar**
- **DesktopPane**

Inserir a imagem no DesktopPane.

(Abra a propriedade do DesktopPane e depois aba code e Custom Creation Code.)



Criando TelaPrincipal



Endereço da imagem (Pacote img/
nome da imagem.jpg)

TelaPrincipal:

- **MenuBar**
- **DesktopPane**

Pode dar erro de importação no código fonte da TelaPrincipal

```
4  */
5  package visao;
6
7  /**
8  * @author victorperes
9  */
10 public class TelaPrincipal extends javax.swing.JFrame {
11
12     /**
13      * Creates new form TelaPrincipal
14      */
15     public TelaPrincipal() {
16         initComponents();
17     }
18
19
20     /**
21      * This method is called from within the constructor to initialize the form.
22      * WARNING: Do NOT modify this code. The content of this method is always
23      * regenerated by the Form Editor.
24      */
25     @SuppressWarnings("unchecked")
26     // Generated Code
```

Criando TelaPrincipal



Clique na linha vermelha para ver onde
estar o erro!

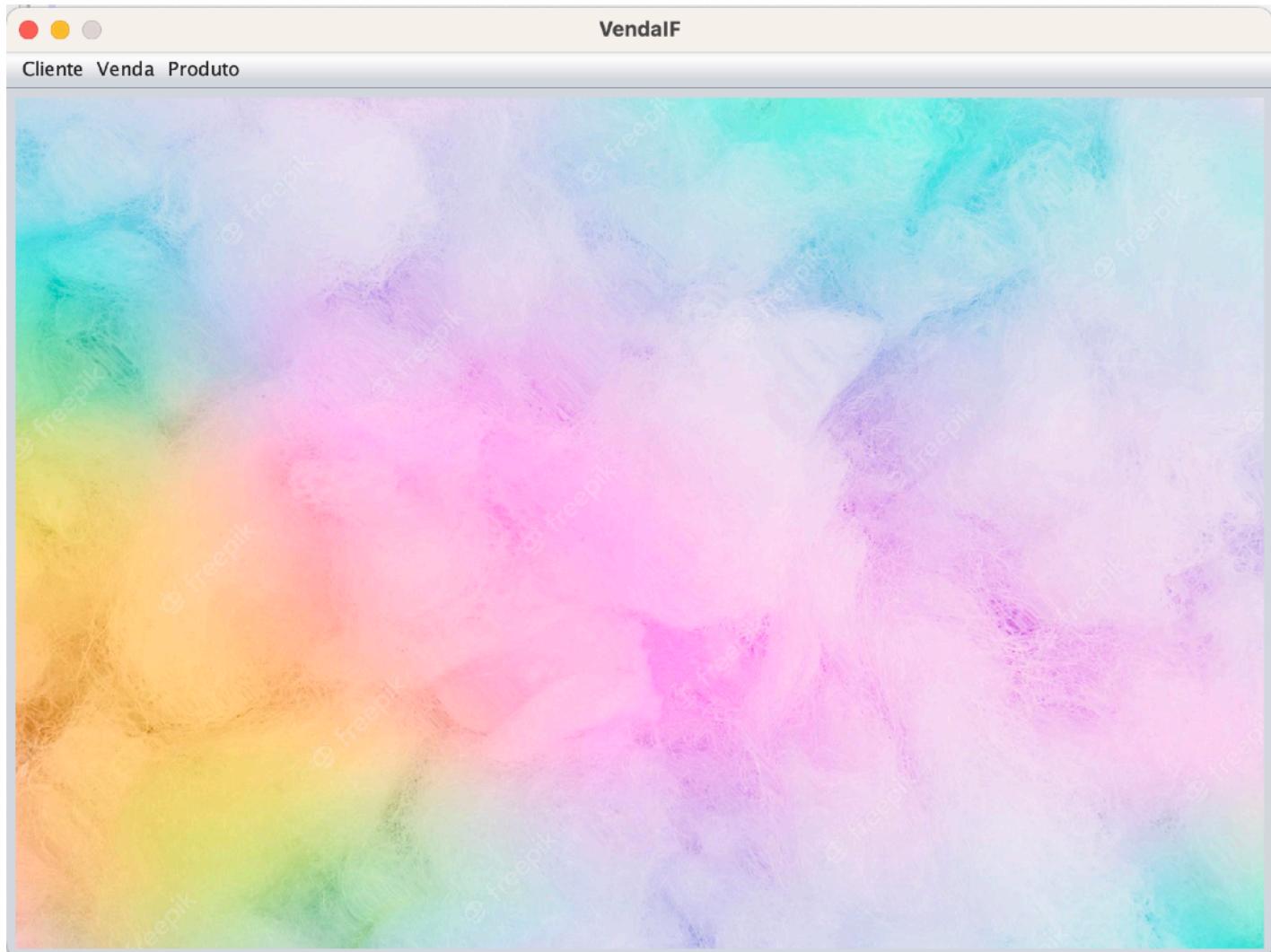
```
29  private javax.swing.JPanel panel = new javax.swing.JPanel();
30
31  private Image image;
32
33  image = ImageIO.read(getClass().getResource("/img/fundo.jpg"));
34
35  protected void paintComponent(Graphics g) {
36
```

Clique para fazer a importação

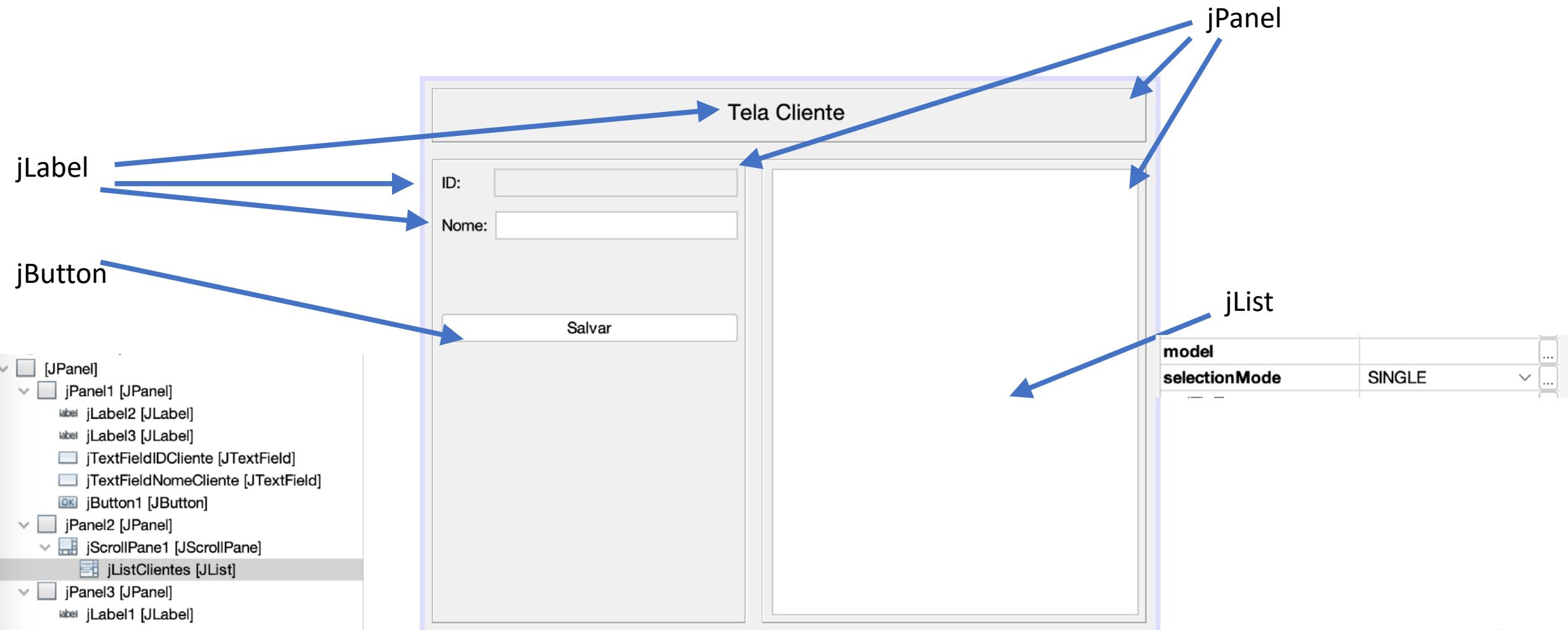
TelaPrincipal:

- **MenuBar**
- **DesktopPane**

Criando TelaPrincipal



Criando TelaCliente



Criando Modelo Cliente

Declarar as variáveis

```
public class Cliente implements Serializable{
```

Construtor

```
int id;  
String nome;
```

```
public Cliente() {  
}
```

```
public Cliente(int id, String nome) {  
    this.id = id;  
    this.nome = nome;  
}
```

Para poder salvar a Classe em arquivo

```
public int getId() {  
    return id;  
}
```

```
public void setId(int id) {  
    this.id = id;  
}
```

```
public String getNome() {  
    return nome;  
}
```

```
public void setNome(String nome) {  
    this.nome = nome;  
}
```

Get e Set

Criando BD_Cliente

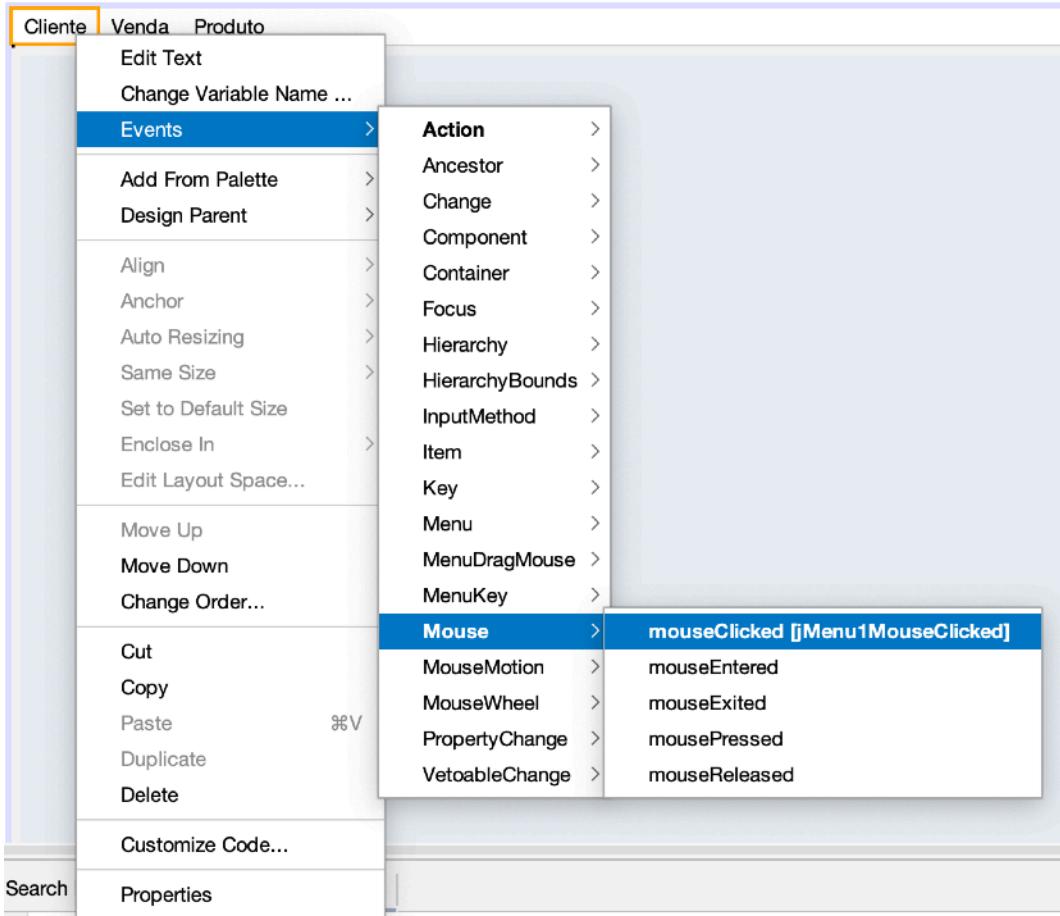
```
19 public class BD_Cliente {  
20  
21     public int salvarBanco(ArrayList<Cliente> dados) {  
22         try {  
23             FileOutputStream fs = new FileOutputStream("ClienteBanco.vcp");  
24             ObjectOutputStream os = new ObjectOutputStream(fs);  
25             os.writeObject(dados);  
26             os.close();  
27             return 0;  
28         } catch (Exception ex) {  
29             ex.printStackTrace();  
30             return -1;  
31         }  
32     }  
33  
34     public ArrayList<Cliente> carregarBanco() {  
35         ArrayList<Cliente> dados = new ArrayList<>();  
36         try {  
37             FileInputStream fis = new FileInputStream("ClienteBanco.vcp");  
38             ObjectInputStream ois = new ObjectInputStream(fis);  
39             Object ob = ois.readObject();  
40  
41             dados = (ArrayList<Cliente>) ob;  
42             ois.close();  
43             return dados;  
44         } catch (Exception ex) {  
45             ex.printStackTrace();  
46             return dados;  
47         }  
48     }  
49 }  
50 }
```

Controlador de Telas

```
public class ControladorTelas {  
    JDesktopPane desktopPane;  
  
    public ControladorTelas(JDesktopPane desktopPane) {  
        this.desktopPane = desktopPane;  
    }  
  
    public void TelaCliente() {  
        desktopPane.removeAll();  
        desktopPane.updateUI();  
        Dimension resolucao = desktopPane.getSize(); //captura resolução do container  
        TelaCliente t = new TelaCliente();  
        t.setSize(resolucao);  
        t.setLocation(0, 0);  
        desktopPane.add(t);  
        t.setVisible(true);  
    }  
  
    public void inicio() {  
        desktopPane.removeAll();  
        desktopPane.updateUI();  
    }  
}
```

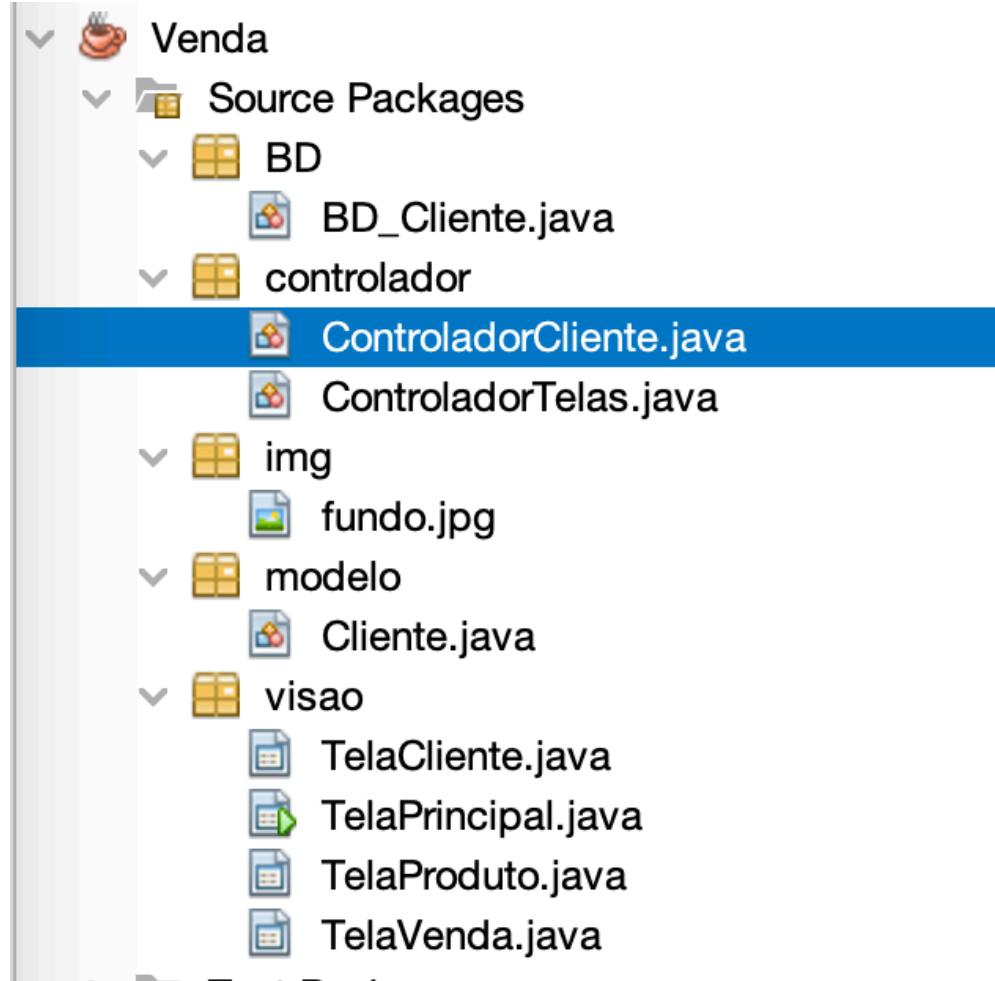


Ação do Cliente



```
ControladorTelas controlTelas ;  
public void inicio(){  
    controlTelas = new ControladorTelas(jDesktopPane1);  
}  
  
private void jMenu1MouseClicked(java.awt.event.MouseEvent evt) {  
    controlTelas.TelaCliente();  
    // TODO add your handling code here:  
}
```

Projeto



Classe de BD (Onde será
chamado o método para salvar
e carregar em arquivo)

Controlador Cliente

Declarar as variáveis que vamos
utilizar na TelaCliente

```
public class ControladorCliente {  
    JTextField jTId;  
    JTextField jTNome;  
    JList<String> jlClientes;  
    JButton jBSalvarEditar;  
  
    ArrayList<Cliente> BDClientes = new ArrayList<>();  
  
    DefaultListModel listaTelaCarro = new DefaultListModel();  
  
    BD_Cliente bd_cliente = new BD_Cliente();  
  
    public ControladorCliente(JTextField jTId, JTextField jTNome, JList<String> jlClientes, JButton jBSalvarEditar) {  
        this.jTId = jTId;  
        this.jTNome = jTNome;  
        this.jlClientes = jlClientes;  
        this.jBSalvarEditar = jBSalvarEditar;  
    }  
}
```

Lista de clientes (Banco de
dados) Aqui vai ficar salvo os
objetos Clientes

Variável para inserir dados na
JList (Lista de Clientes na Tela
Cliente)

Construtor para passar as
variáveis da telaCliente para o
Controlador)

Controlador Cliente - Ação do botão Novo

```
public ControladorCliente(JTextField jTId, JTextField jTNome, JList<String> jLClientes, JButton jBSalvarEditar) {  
    this.jTId = jTId;  
    this.jTNome = jTNome;  
    this.jLClientes = jLClientes;  
    this.jBSalvarEditar = jBSalvarEditar;  
  
    novoID();  
    carregarLista();  
}  
  
public void novoID() {  
    BDClientes = bd_cliente.carregarBanco();  
    int tamanhoArray = BDClientes.size();  
    jTId.setText((tamanhoArray + 1) + "");  
    limpar();  
}  
  
public void limpar() {  
    jTNome.setText("");  
    jTNome.setBackground(Color.white);  
    jBSalvarEditar.setText("Salvar");  
}
```



Controlador Cliente - Ação do botão Salvar

```
public void salvarEditarCliente() {  
    String nId = jTId.getText();  
    String nome = jTNome.getText();  
    boolean verificador = true;  
  
    if (campoVazio(nome)) {  
        JOptionPane.showMessageDialog(null, "O campo Nome não pode ser vazio");  
        jTNome.setBackground(Color.red);  
        verificador = false;  
    }  
  
    if (verificador) {  
        //SALVAR  
        Cliente cliente = new Cliente(Integer.parseInt(nId), nome);  
  
        if (jBSalvarEditar.getText().compareToIgnoreCase("Salvar") == 0) {  
            BDClientes.add(cliente);  
  
        } else {  
            int index = pegarIndex(cliente);  
  
            BDClientes.set(index, cliente);  
        }  
        limpar();  
        carregarLista();  
        bd_cliente.salvarBanco(BDClientes);  
    }  
}
```

```
public int pegarIndex(Cliente cliente) {  
    int index = 0;  
    for (Cliente cli : BDClientes) {  
        if (cli.getId() == cliente.getId()) {  
            break;  
        }  
        index++;  
    }  
    return index;  
}
```

```
public boolean campoVazio(String temp) {  
    return temp.isEmpty();  
}
```

```
public void carregarLista(){  
    listaTelaCliente = new DefaultListModel();  
    ArrayList<String> ListaTelaAtualizada = new ArrayList<>();  
    for (Cliente cliente1 : BDClientes) {  
        ListaTelaAtualizada.add(cliente1.getNome()+" ("+cliente1.getId()+")");  
    }  
    Collections.sort(ListaTelaAtualizada);  
    listaTelaCliente.addAll(ListaTelaAtualizada);  
    jLClientes.setModel(listaTelaCliente);  
}
```

Controlador Cliente - Editar ao clicar na JList

```
public void editar(){
    String temp = jLClientes.getSelectedValue();
    int indexInicioConsulta = temp.indexOf("(", 2);
    int indexFinalConsulta = temp.indexOf(")", 2);
    String res = temp.substring(indexInicioConsulta + 1, indexFinalConsulta);
    int id = Integer.parseInt(res);
    Cliente clienteEditar = buscar(id);

    jTId.setText(clienteEditar.getId()+"");
    jTNome.setText(clienteEditar.getNome());

    jBSalvarEditar.setText("Editar");
}

public Cliente buscar(int id){
    for (Cliente cliente1 : BDClientes) {
        if(cliente1.getId()==id){
            return cliente1;
        }
    }
    return null;
}
```

