

FCT/Unesp – Presidente Prudente
Departamento de Matemática e Computação

**Programação Orientada
a Objetos**
Interface Gráfica
Parte I

Prof. Dr. Danilo Medeiros Eler
danilo.eler@unesp.br

Aula de Hoje

- Interface de Usuário (IU)
 - *User Interface* (UI)
 - Interface Gráfica

Interface de Usuário

- Utilizaremos componentes do java para criar interfaces para entrada e saída de dados
- Faremos o exemplo de cadastro de Produto Nacional e Produto Importado
- O modelo de dados e a classe controladora já estão implementados, faremos somente a interface, invocando os métodos da classe controladora para efetuar o cadastro

Adicionar um JFrame

The screenshot shows an IDE interface with a project tree on the left and a code editor on the right. The project tree shows a package named 'exemploaula09' containing a 'Main.java' file. A context menu is open over the 'exemploaula09' package, with the 'Novo' (New) option selected. The 'Novo' submenu is also open, showing various options for creating new code elements. The 'Formulário JFrame...' option is highlighted. The code editor shows the following code:

```
6 package exemploaula09;  
7  
8 /**  
9 *  
10 * @author Danilo Medei
```

The context menu options are:

- Novo
 - Pacote Java...
 - Classe Java...
 - Formulário JDialog...
 - Enum Java...
 - Formulário JFrame...**
 - Formulário JPanel...
 - Interface Java...
 - Classe da entidade...
 - Classes de entidade do banco de dados...
 - Outro...
- Localizar... Ctrl+F
- Recortar Ctrl+X
- Copiar Ctrl+C
- Colar Ctrl+V
- Excluir Delete
- Refatorar
- Compilar pacote F9
- Histórico local
- Atualizar pasta
- Ferramentas

Adicionar um JFrame

Novo Formulário JFrame

Passos

1. Escolha o tipo de arquivo
2. **Nome e local**

Nome e local

Nome da classe: IUPrincipal

Projeto: ExemploAula09

Localização: Pacotes de código-fonte

Paquete: exemploaula09.iu

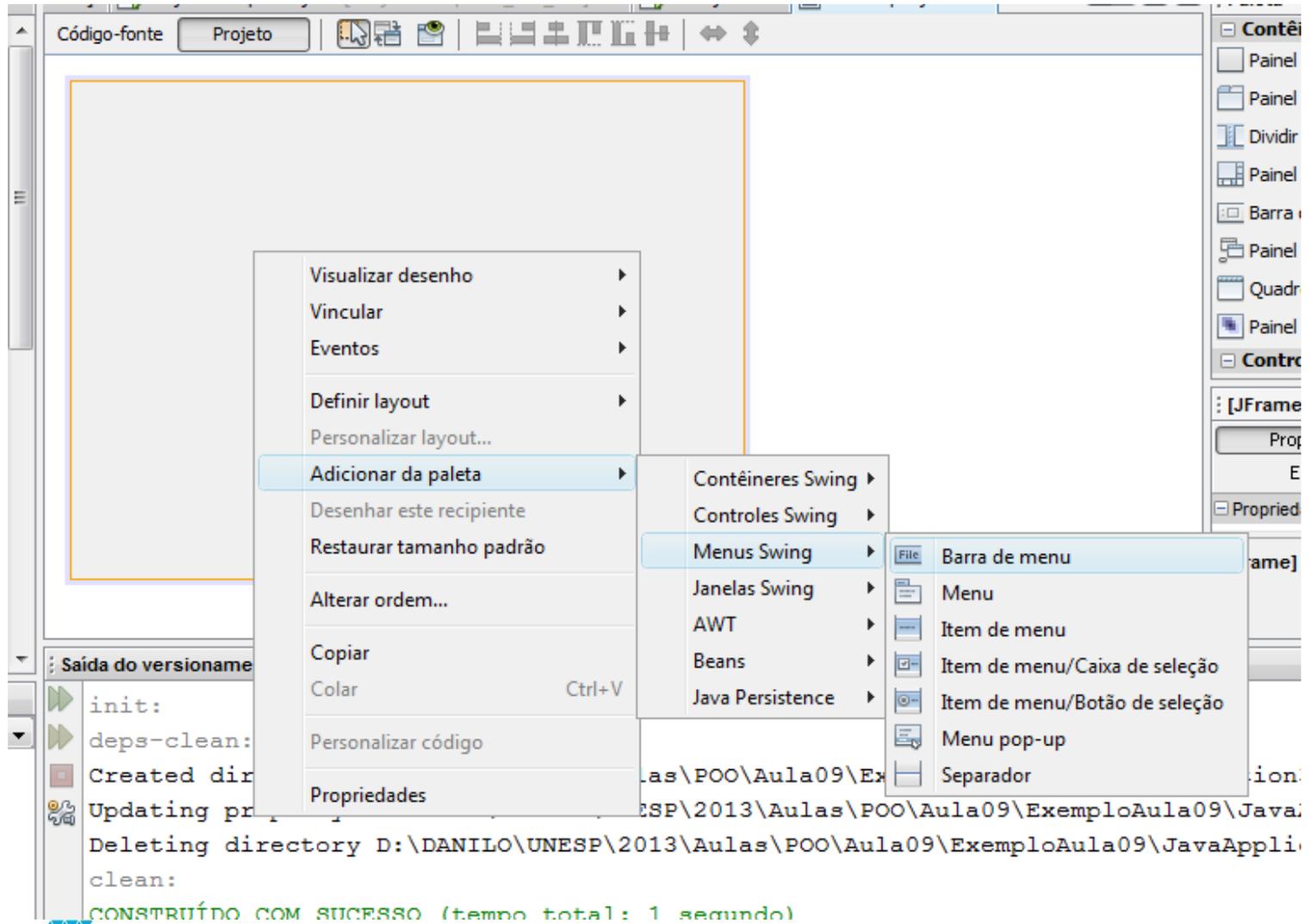
Arquivo criado: JNESP\2013\Aulas\POO\Aula09\ExemploAula09\src\exemploaula09\iu\IUPrincipal.java

< Voltar Próximo > Finalizar Cancelar Ajuda

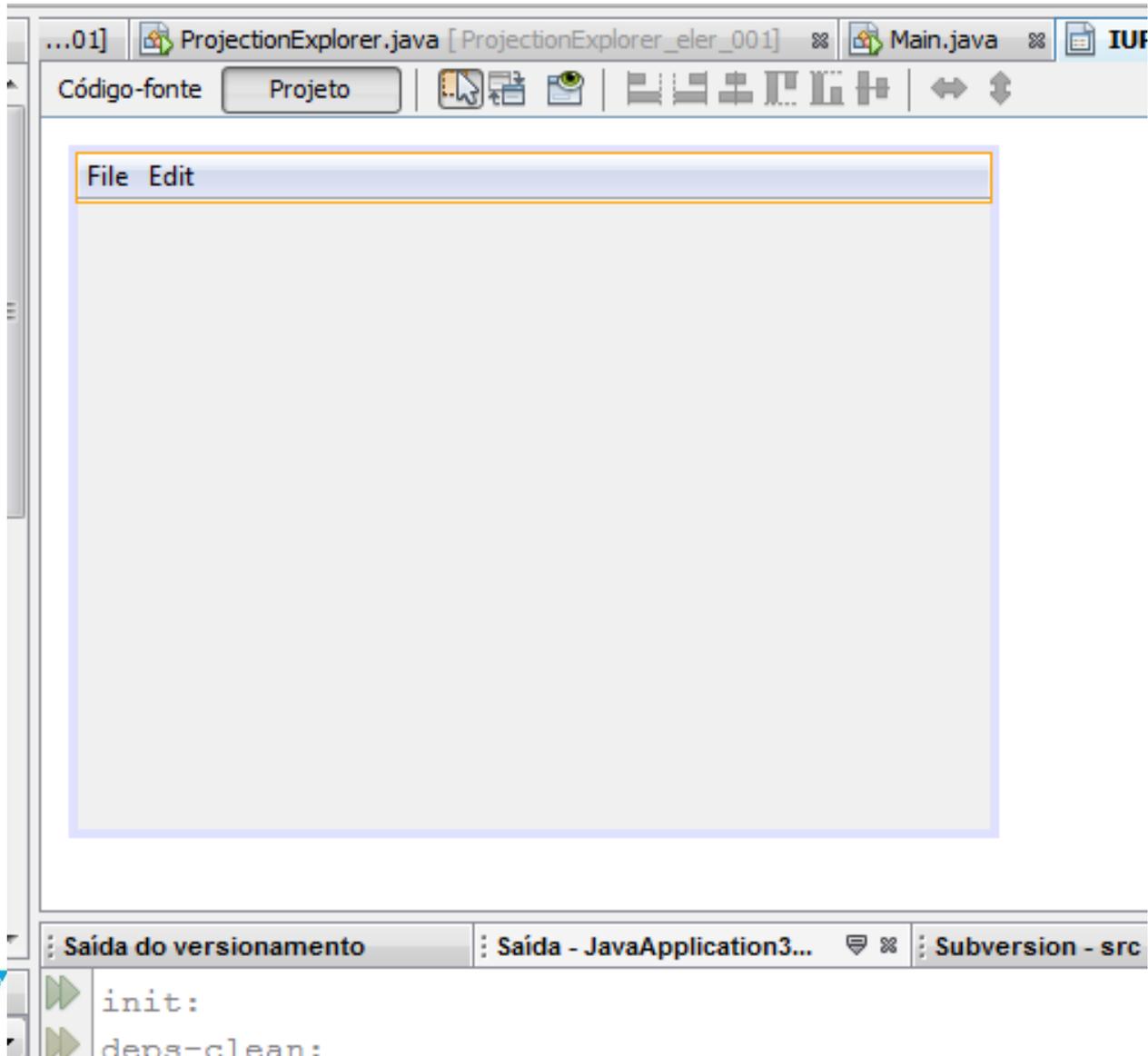
Adicionar um JFrame

The screenshot displays the NetBeans IDE 6.9.1 interface. The main editor window shows a large, empty rectangular area, indicating a new JFrame is being added. A red arrow points from the title 'Adicionar um JFrame' to this area. The left sidebar shows the project structure for 'ExemploAula09', including source code packages and test packages. The right sidebar shows the 'Paleta' (Palette) with various Swing components like 'Painel', 'Painel tabulado', and 'Controles Swing'. The bottom status bar shows the output of a build process, including the message 'CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)'. The bottom-left corner shows the 'Navegador' (Navigator) and 'Inspetor' (Inspector) tabs, with the 'IUPrincipal' class selected.

Adicionar uma Barra de Menu



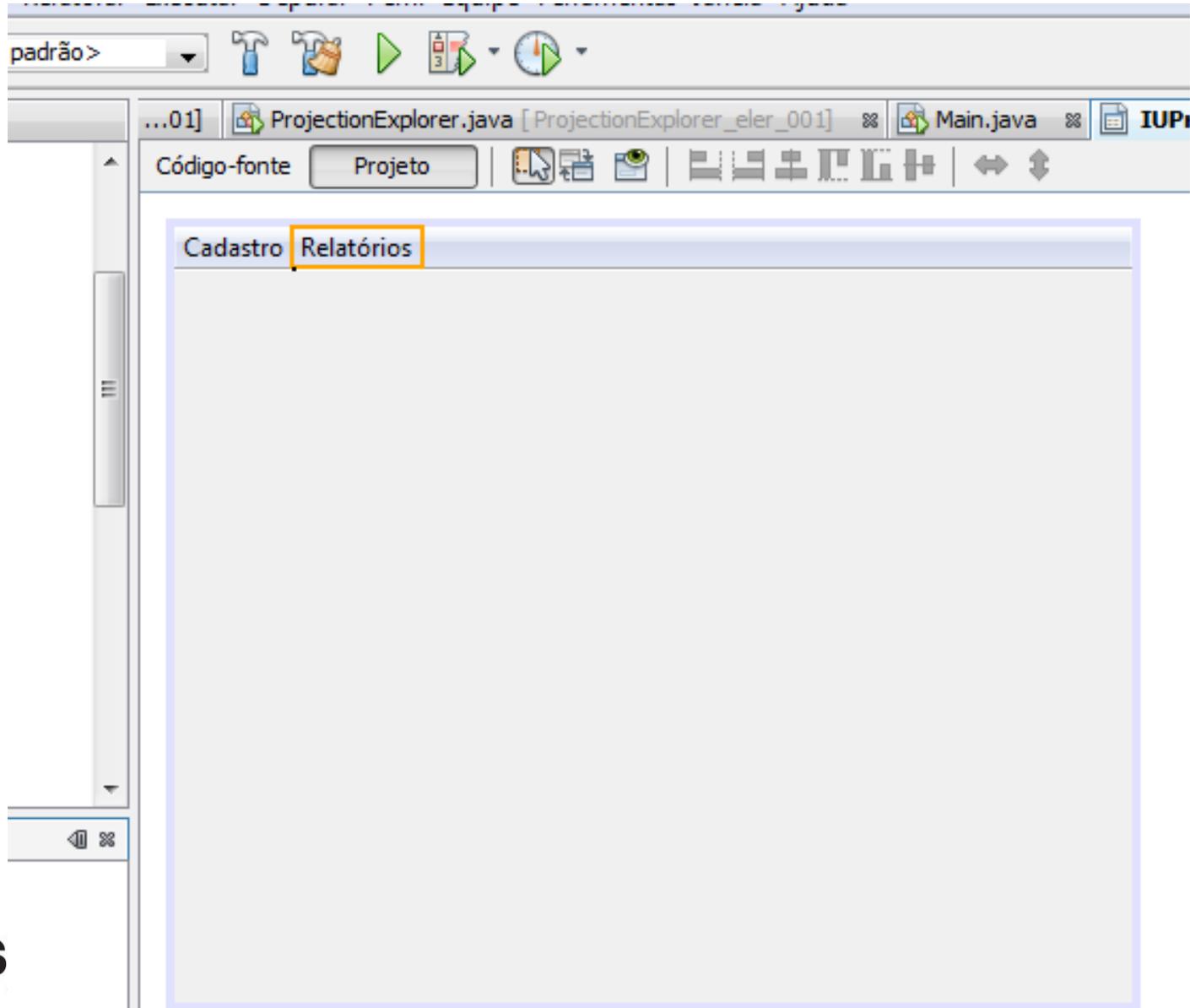
Adicionar uma Barra de Menu



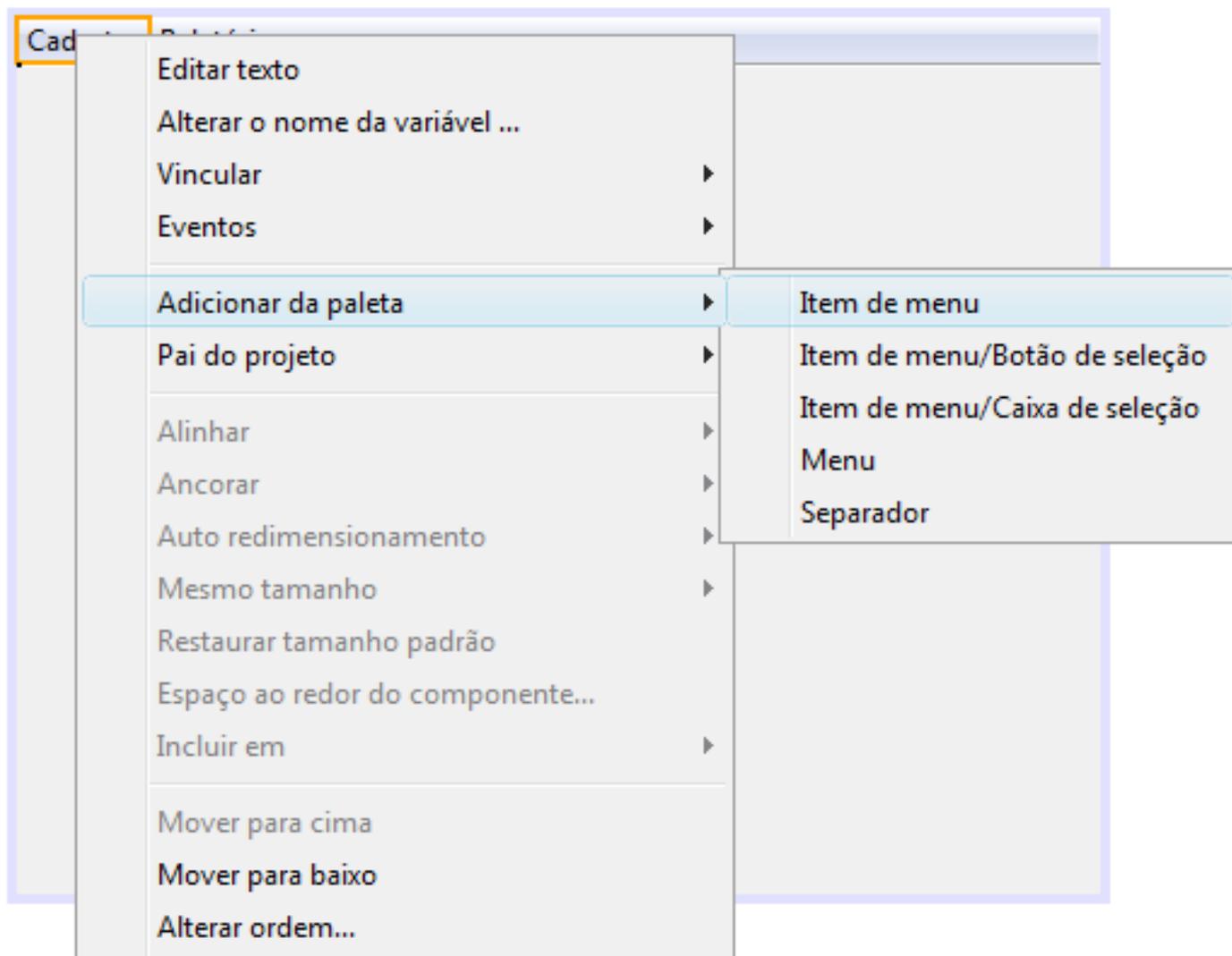
Editar Textos dos Menus

The image shows a screenshot of an IDE interface. On the left, a project tree displays a project named 'ExemploAula09' with sub-packages for source code, tests, and libraries. Below the tree is the 'Inspecor' (Inspector) window, which shows a hierarchy of components: 'Formulário IUPrincipal' containing 'JFrame', which contains 'JMenuBar1 [JMenuBar]', which in turn contains 'jMenu1 [JMenu]' and 'jMenu2 [JMenu]'. The 'jMenu1 [JMenu]' component is selected. A context menu is open over this component, listing various actions. The 'Editar texto' (Edit text) option is highlighted. Other options include 'Alterar o nome da variável ...', 'Vincular', 'Eventos', 'Adicionar da paleta', 'Pai do projeto', 'Alinhar', 'Ancorar', 'Auto redimensionamento', 'Mesmo tamanho', 'Restaurar tamanho padrão', 'Espaço ao redor do componente...', 'Incluir em', 'Mover para cima', 'Mover para baixo', 'Alterar ordem...', 'Recortar', 'Copiar', 'Colar' (with 'Ctrl+V' shortcut), 'Duplicar', 'Excluir', 'Personalizar código', and 'Propriedades'.

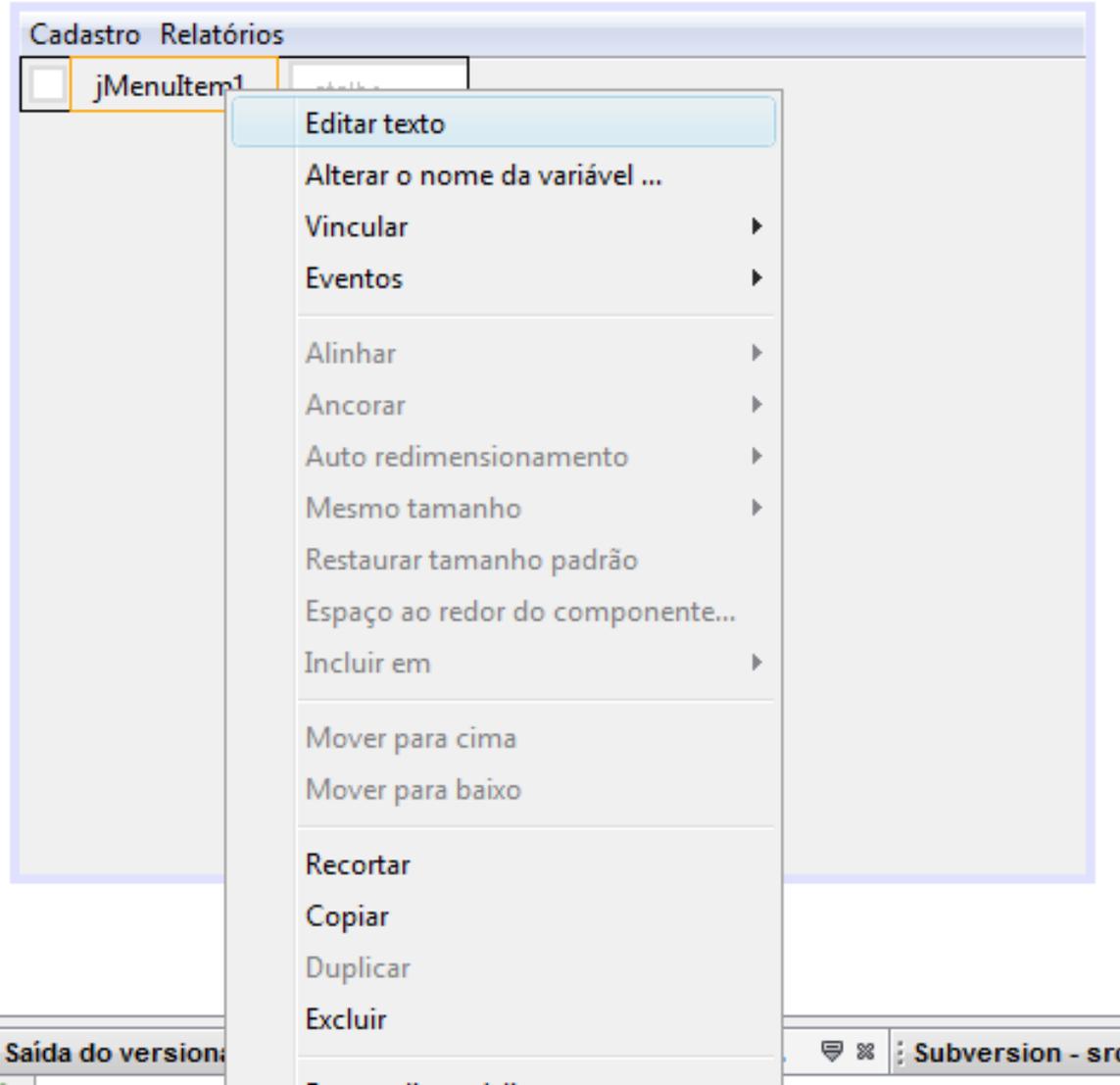
Editar Textos dos Menus



Adicionar Itens aos Menus



Editar Texto dos Items



Cadastro		Relatórios	
<input type="checkbox"/>	Produto Nacional	atalho	
<input type="checkbox"/>	Produto Importado	atalho	

Cadastro **Relatórios**

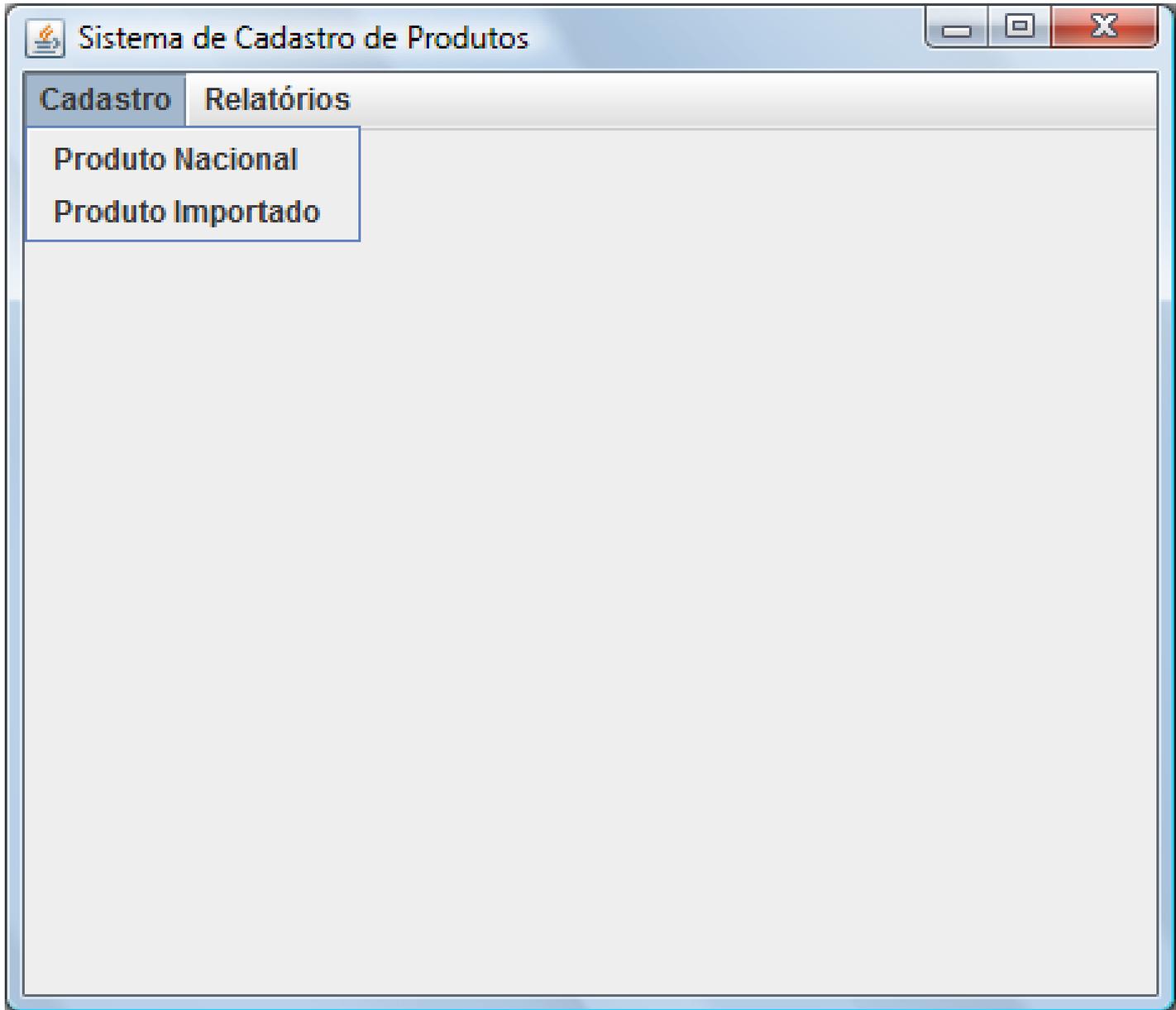
<input type="checkbox"/>	Todos Produtos	atalho
<input type="checkbox"/>	Produtos Nacionais	atalho
<input type="checkbox"/>	Produtos Importados	atalho

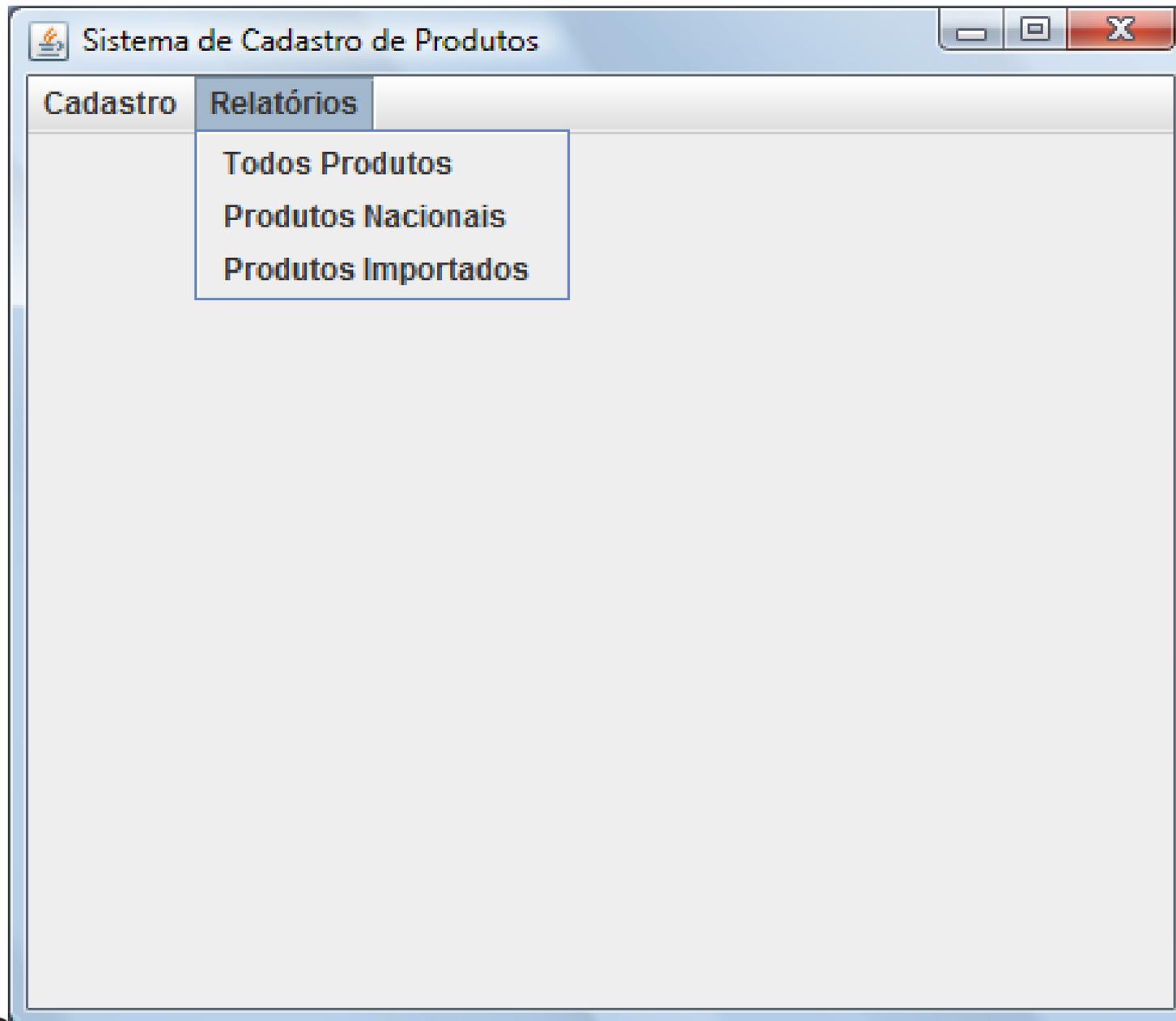
Instanciar IU na classe principal

```
package exemploaula09;

import exemploaula09.iu.IUPrincipal;

/**
 * @author Danilo Medeiros Eler
 */
public class Main {
    public static void main(String[] args) {
        // TODO code application logic here
        IUPrincipal principal = new IUPrincipal();
        principal.setVisible(true);
        principal.setTitle("Sistema de Cadastro de Produtos");
        principalToFront();
    }
}
```





IU de Cadastros

Cadastro de Produto Nacional

The image shows a screenshot of an IDE (likely NetBeans) with a project named 'AppSistemaBancario'. The project structure on the left includes 'Pacotes de código-fonte', 'Bibliotecas', and 'ExemploAula09'. Under 'ExemploAula09', there are sub-packages 'exemploaula09' and 'exemploaula09...'. The 'exemploaula09' package contains files like 'Main.java', 'IUPrincipal', and 'ProdutoNa...'. The 'exemploaula09...' package contains 'ProdutoIm...', 'ProdutoNa...', and 'Sistema.ja...'. A context menu is open over the 'exemploaula09...' package, showing options like 'Novo', 'Localizar...', 'Recortar', 'Copiar', 'Colar', 'Excluir', 'Refatorar', 'Compilar pacote', 'Histórico local', 'Atualizar pasta', and 'Ferramentas'. The 'Novo' option is expanded, showing a list of new elements: 'Classe Java...', 'Formulário JFrame...', 'Pacote Java...', 'Formulário JDialog...' (highlighted), 'Enum Java...', 'Formulário JPanel...', 'Interface Java...', 'Classe da entidade...', 'Classes de entidade do banco de dados...', and 'Outro...'. The code editor on the right shows a snippet of Java code:

```
38     }
39 }
40
41 public Produto busca:
42     for (int i=0; i
43         if (produto
```

Cadastro de Produto Nacional

Novo Formulário JDialog

Passos

1. Escolha o tipo de arquivo
2. **Nome e local**

Nome e local

Nome da classe:

Projeto:

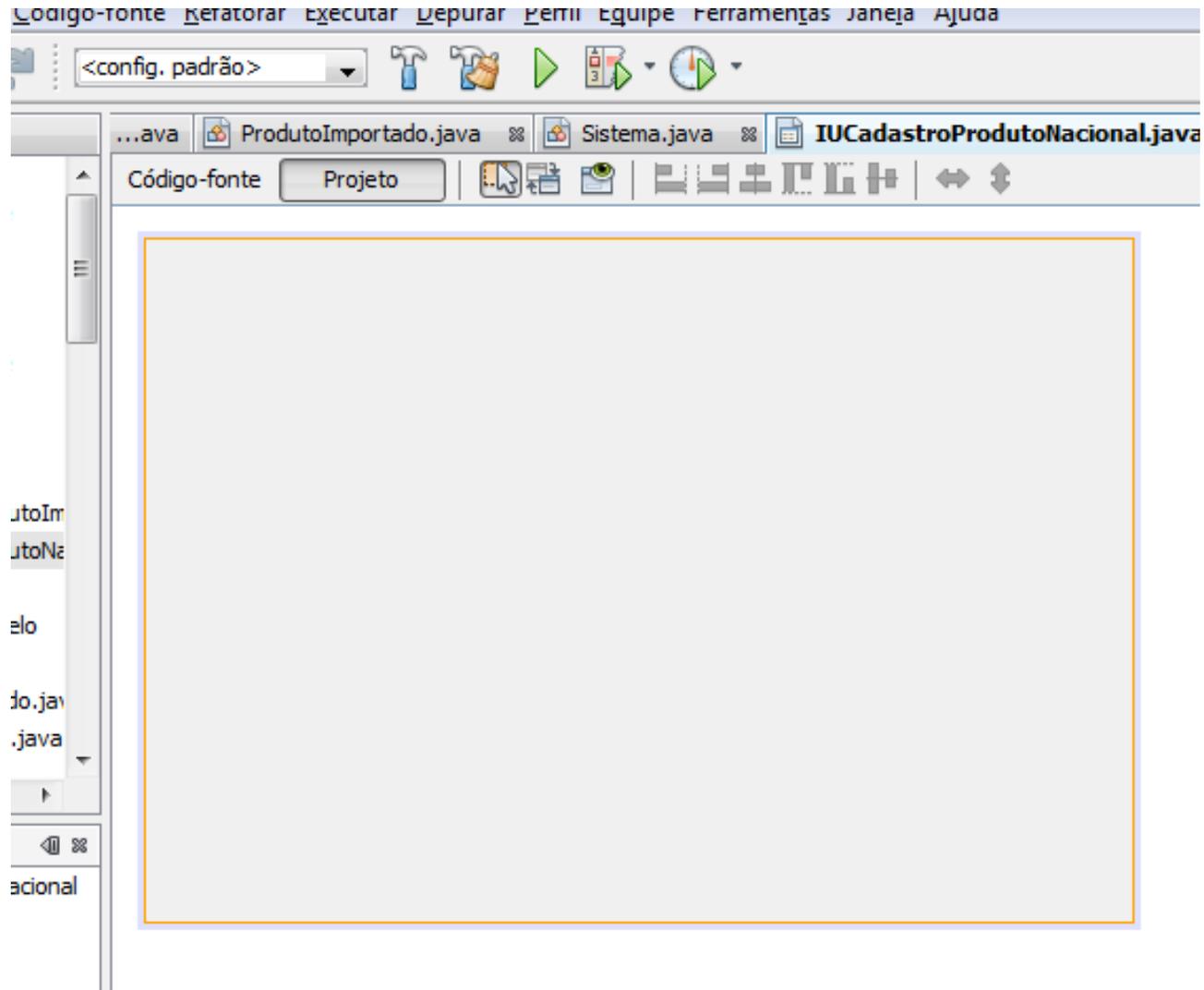
Localização:

Paquete:

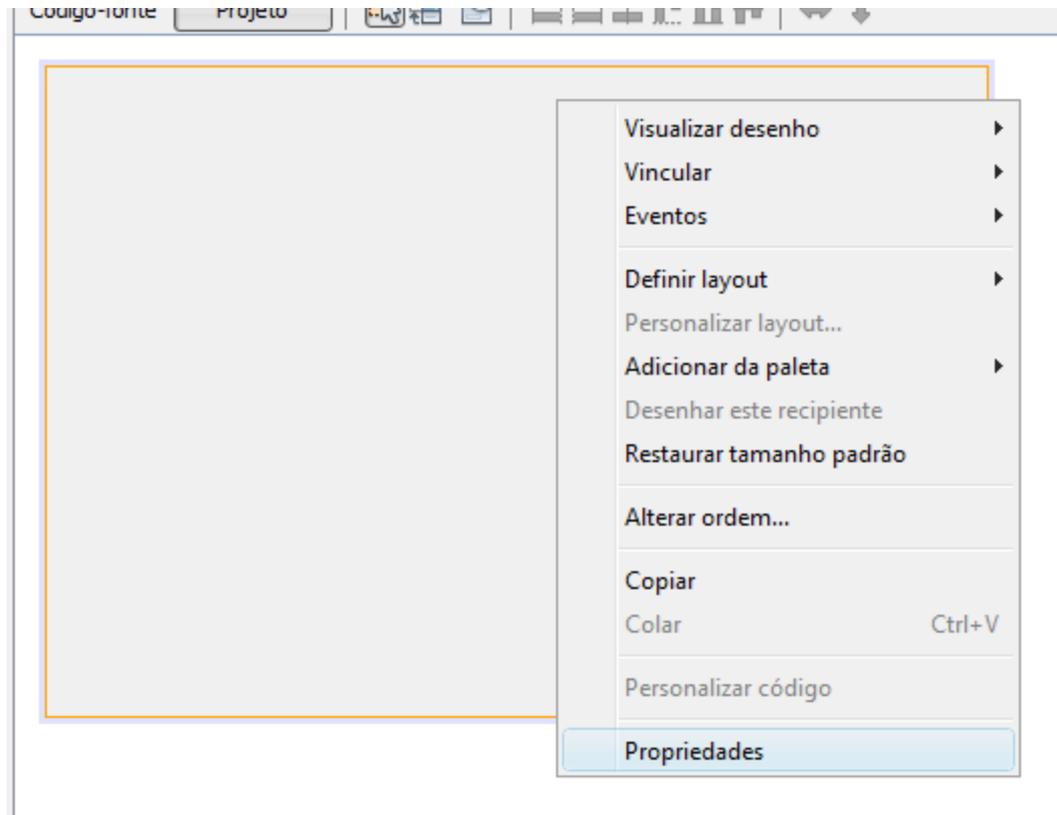
Arquivo criado:

< Voltar Próximo > Finalizar Cancelar Ajuda

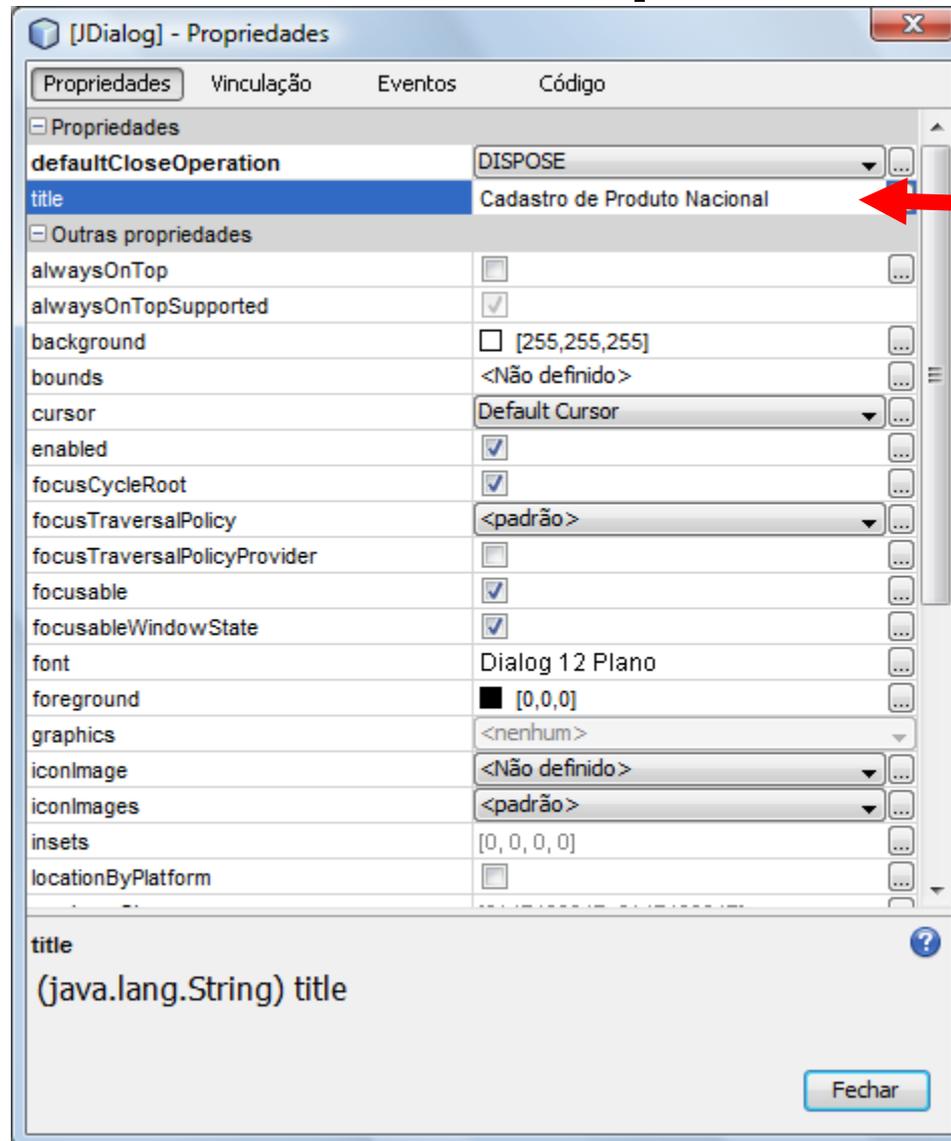
Cadastro de Produto Nacional



Alterando Propriedades

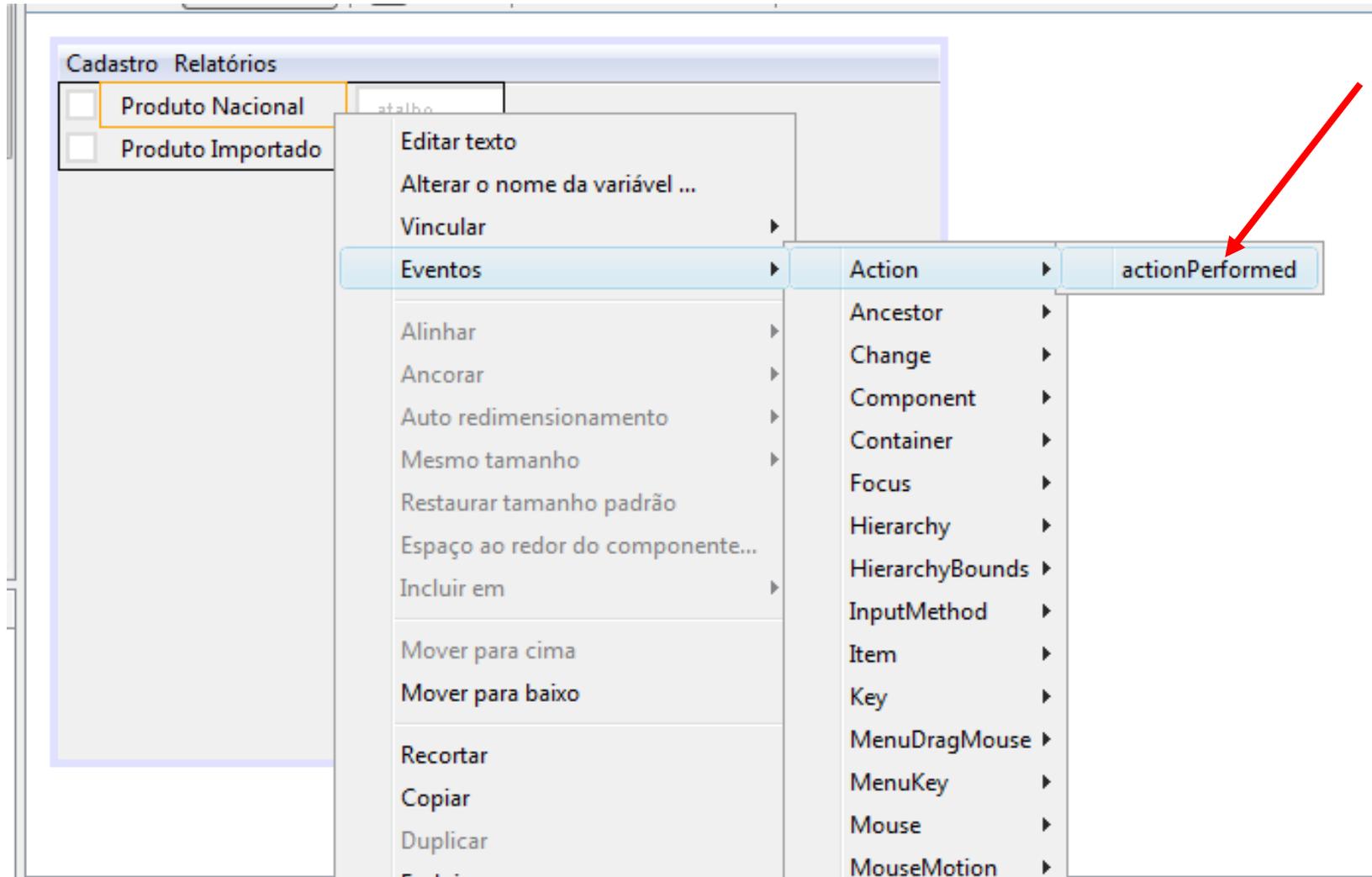


Alterando Propriedades



título

Adicionar Evento ao Item de Menu



Adicionar Evento ao Item de Menu

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    IUCadastroProdutoNacional IUCad = new IUCadastroProdutoNacional(this, true);  
    IUCad.setLocationRelativeTo(this);  
    IUCad.setVisible(true);  
}
```

Parent – para indicar qual janela a invocou

Adicionar Evento ao Item de Menu

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    IUCadastroProdutoNacional IUCad = new IUCadastroProdutoNacional(this, true);  
    IUCad.setLocationRelativeTo(this);  
    IUCad.setVisible(true);  
}
```

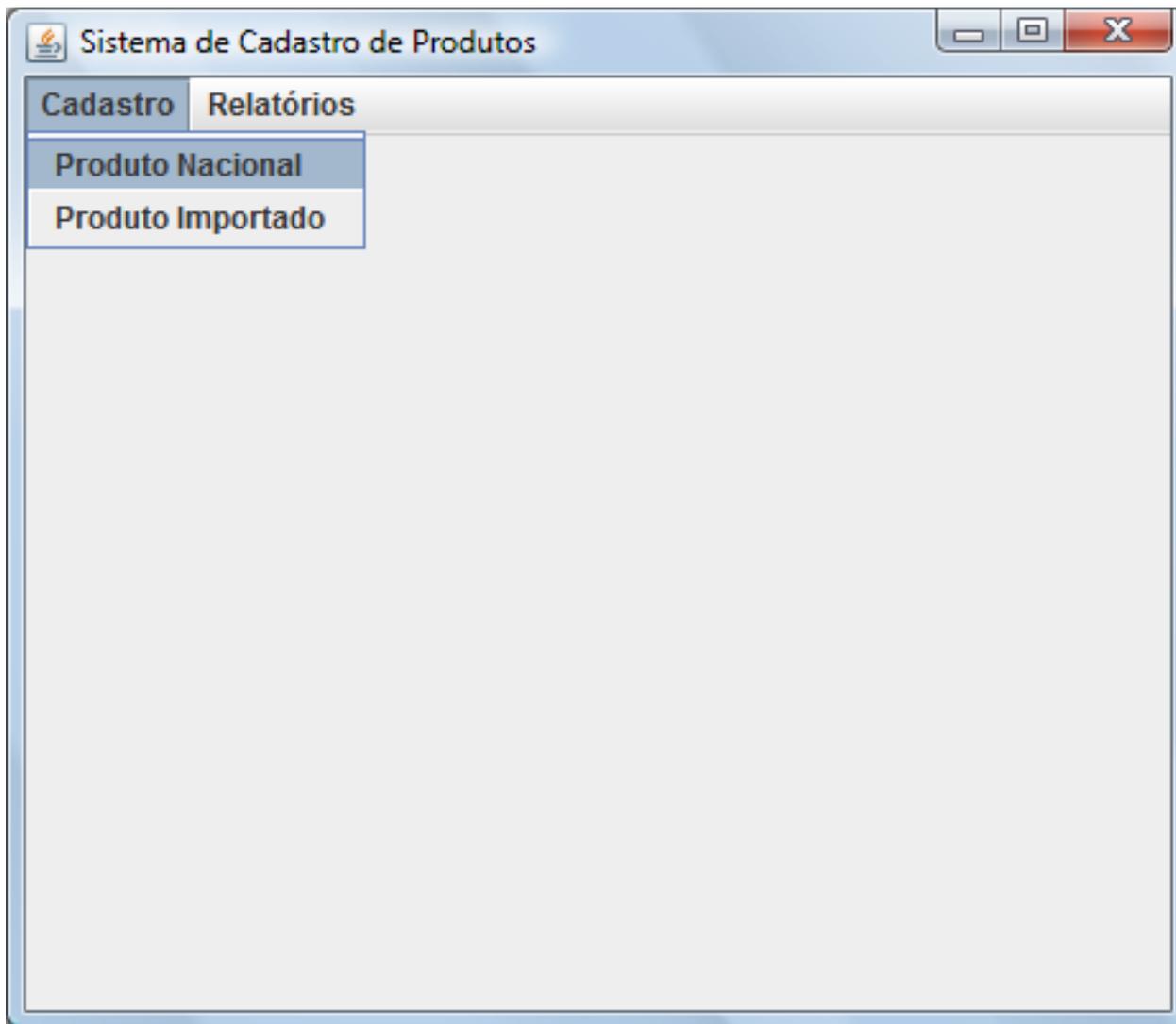
Modal = true – fica preso à janela, só volta à anterior se fechar a atual
Modal = false – não fica preso à janela, pode interagir com a janela anterior

Adicionar Evento ao Item de Menu

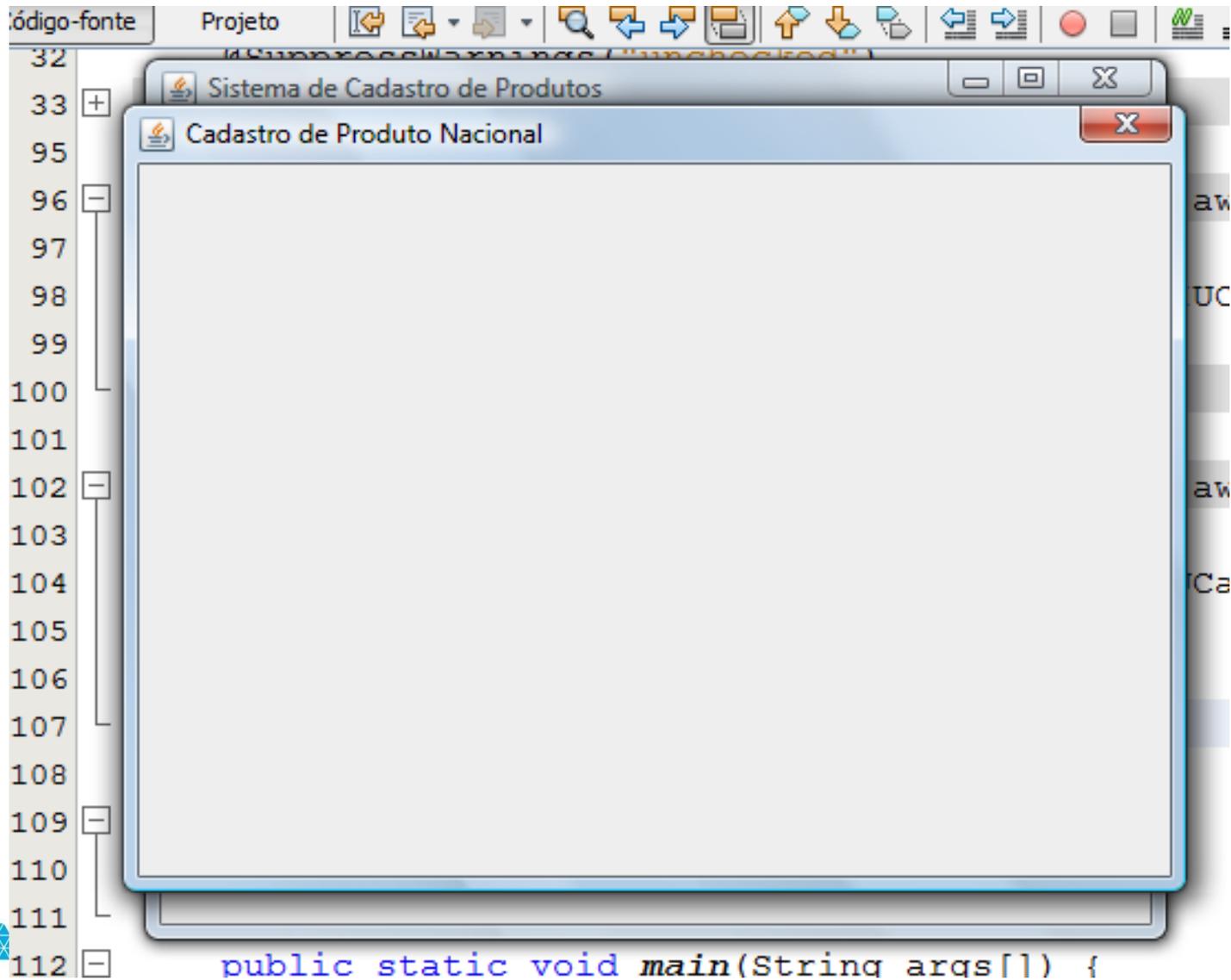
```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    IUCadastroProdutoNacional IUCad = new IUCadastroProdutoNacional(this, true);  
    IUCad.setLocationRelativeTo(this);  
    IUCad.setVisible(true);  
}
```

Para sempre exibir a janela (JDialog) próxima à janela que a invocou

Executando

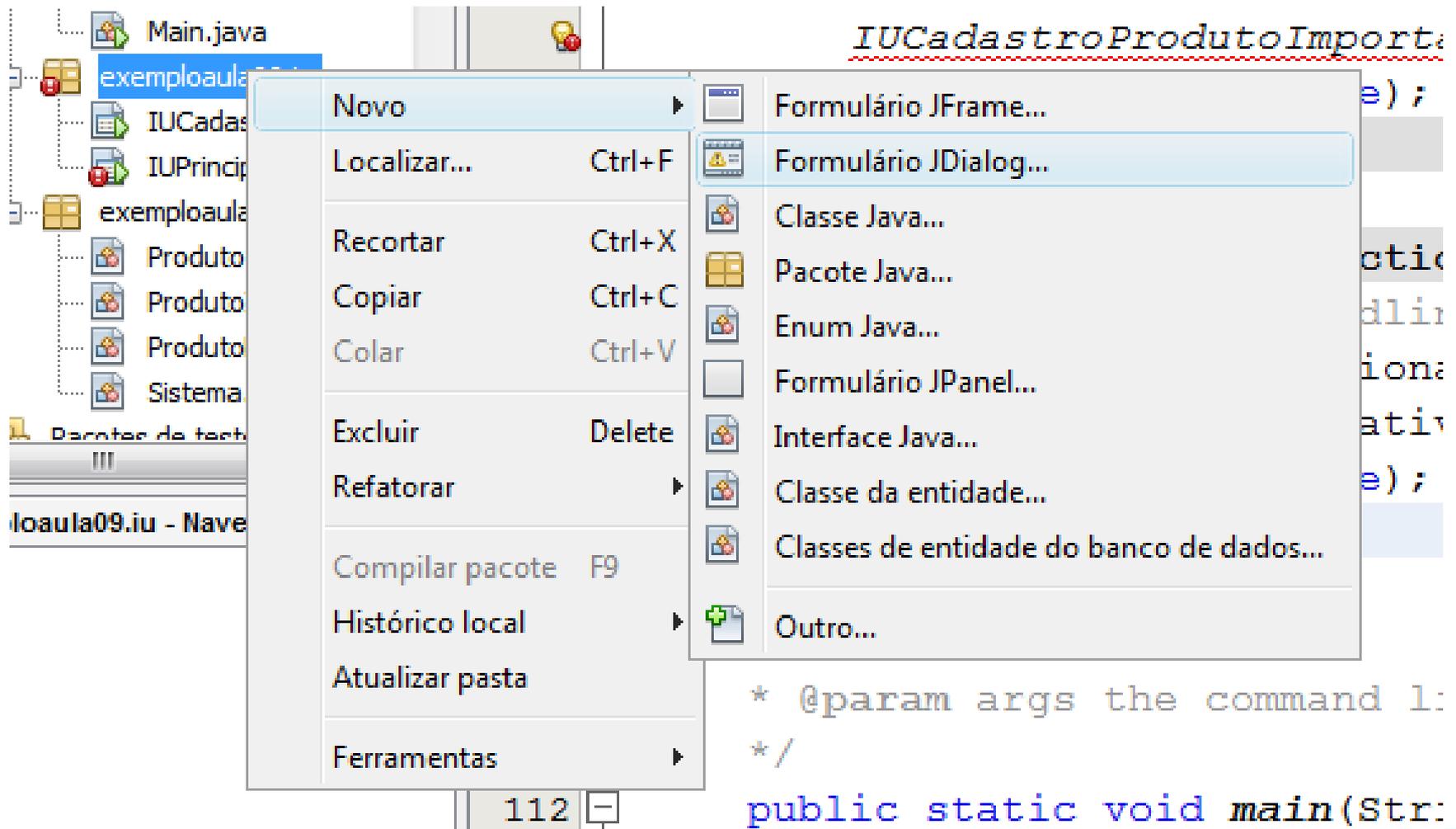


Executando



```
32  
33  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112 public static void main(String args[]) {
```

Cadastro de Produto Importado



IUCadastroProdutoImportado

```
);  
ctio  
dlin  
iona  
ativ  
e);
```

```
* @param args the command line arguments  
*/  
public static void main(String[] args) {
```

112

Cadastro de Produto Importado

Novo Formulário JFrame

Passos

1. Escolha o tipo de arquivo
2. **Nome e local**

Nome e local

Nome da classe:

Projeto:

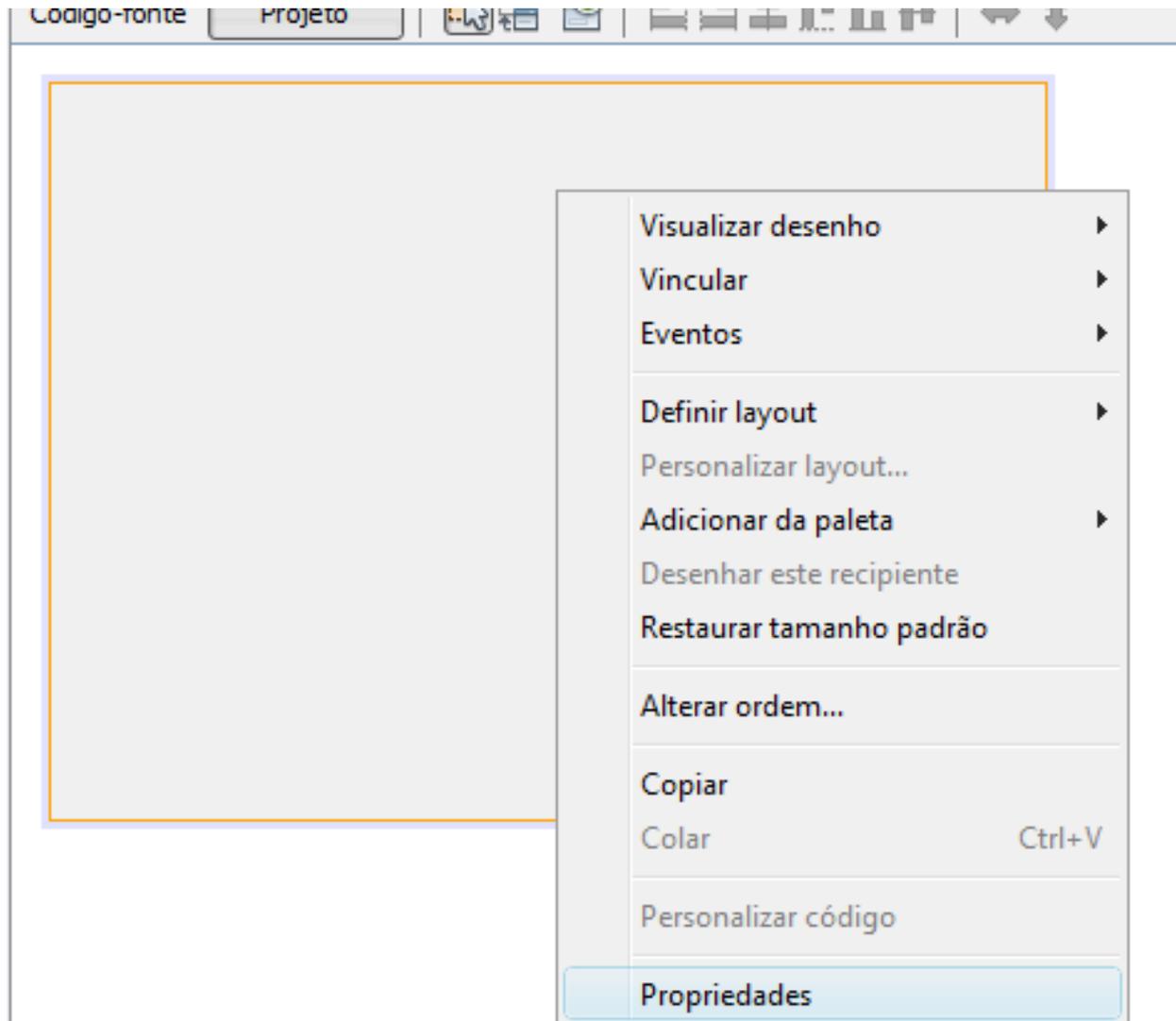
Localização:

Paquete:

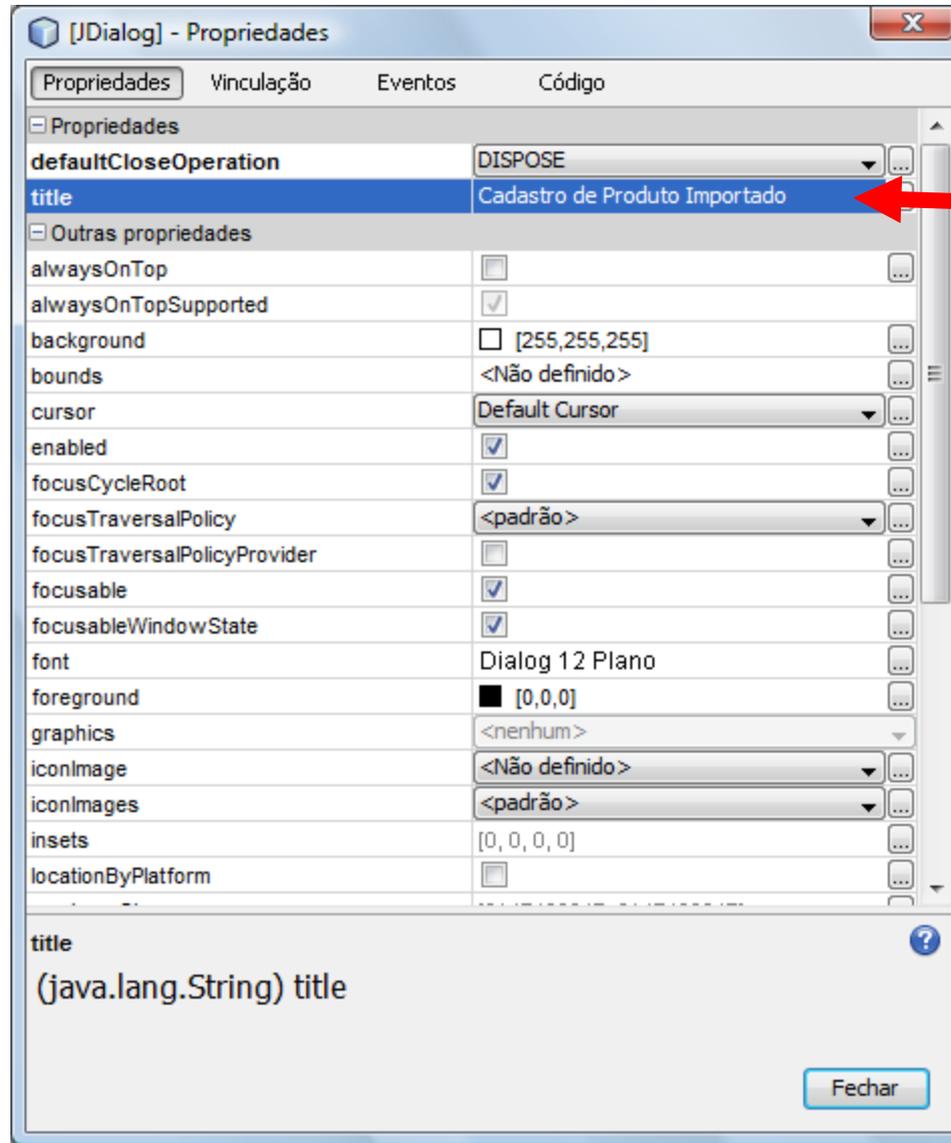
Arquivo criado:

< Voltar Próximo > Finalizar Cancelar Ajuda

Alterando Propriedades

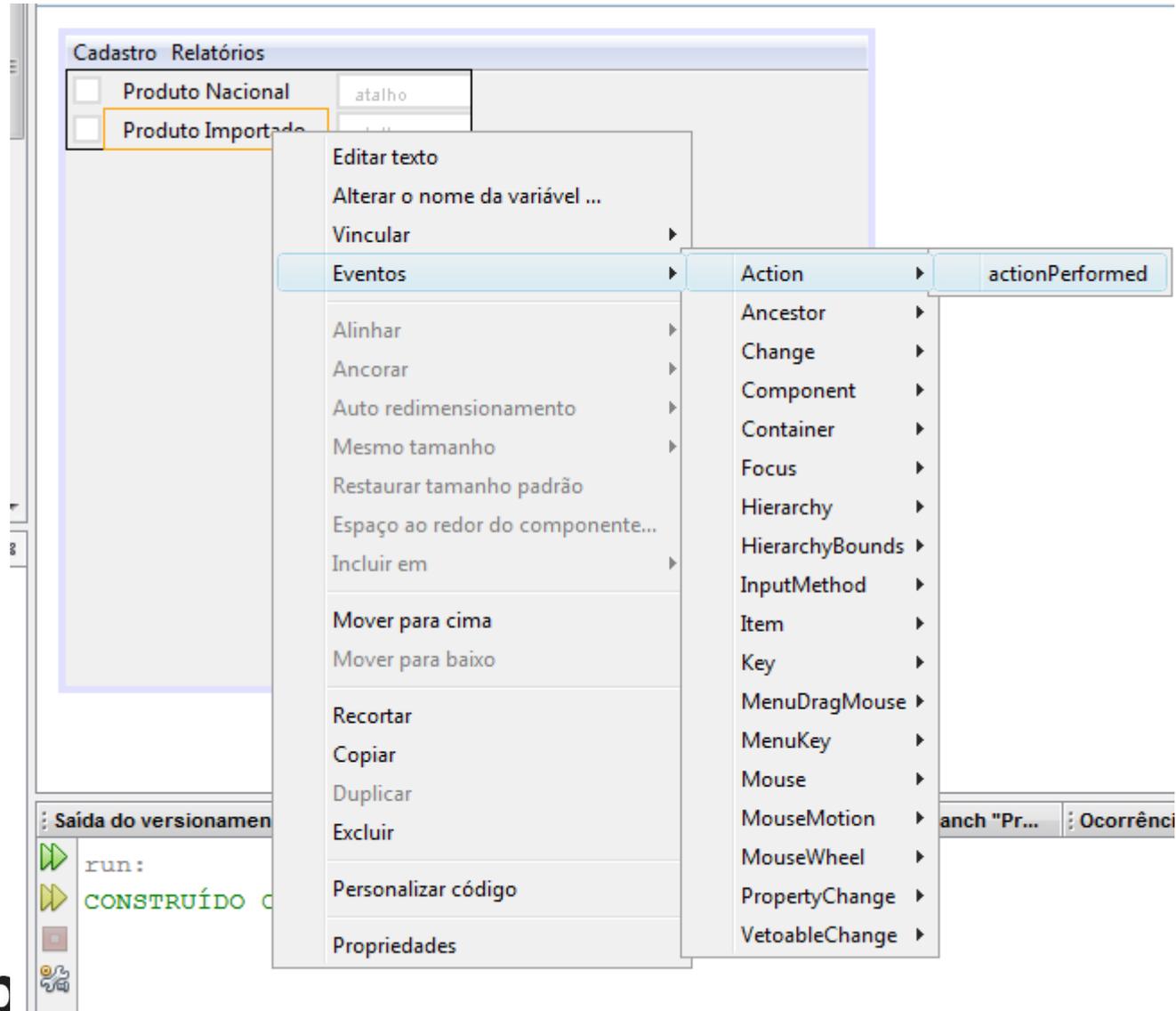


Alterando Propriedades



título

Adicionar Evento no Item de Menu

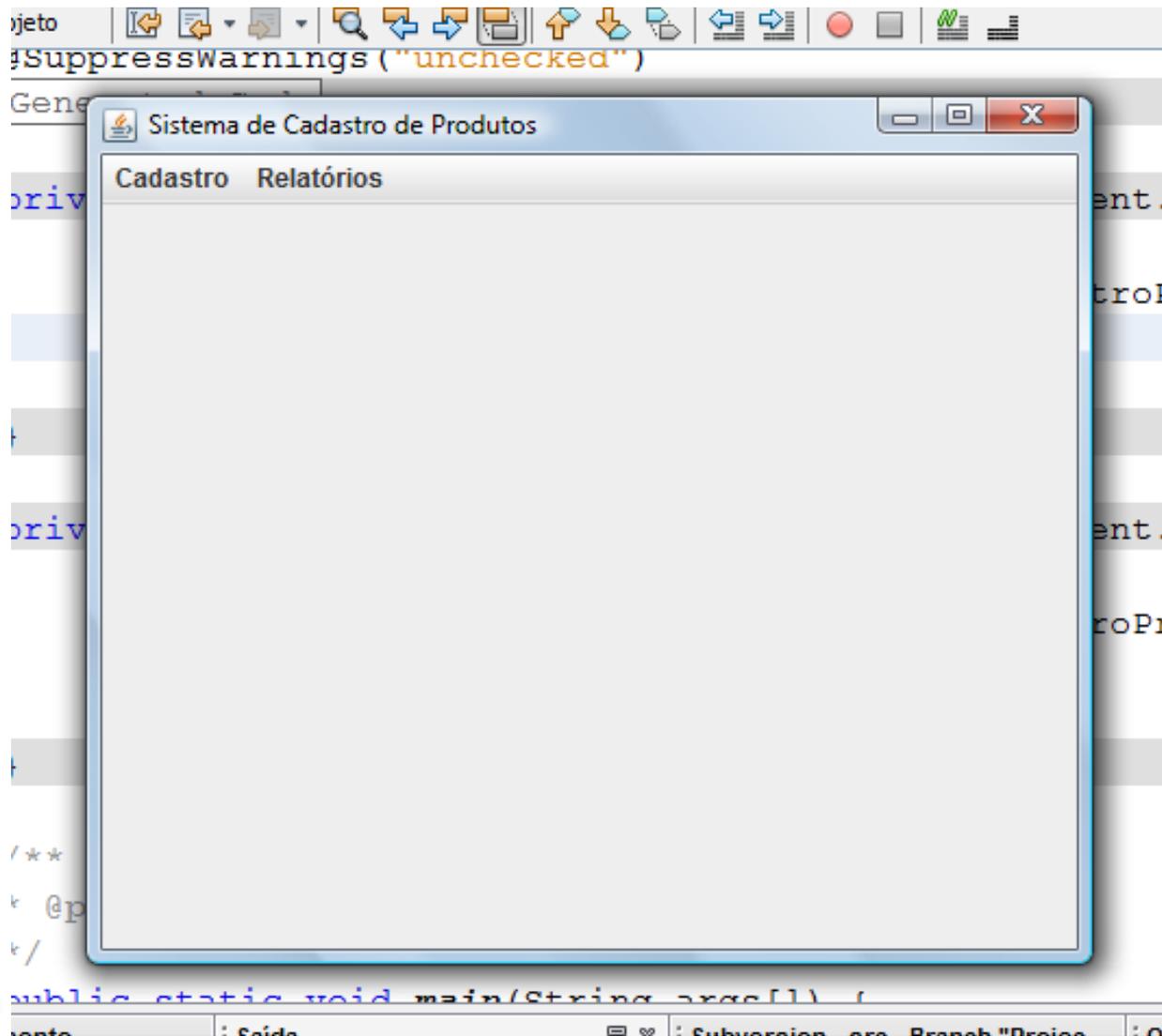


Adicionar Evento no Item de Menu

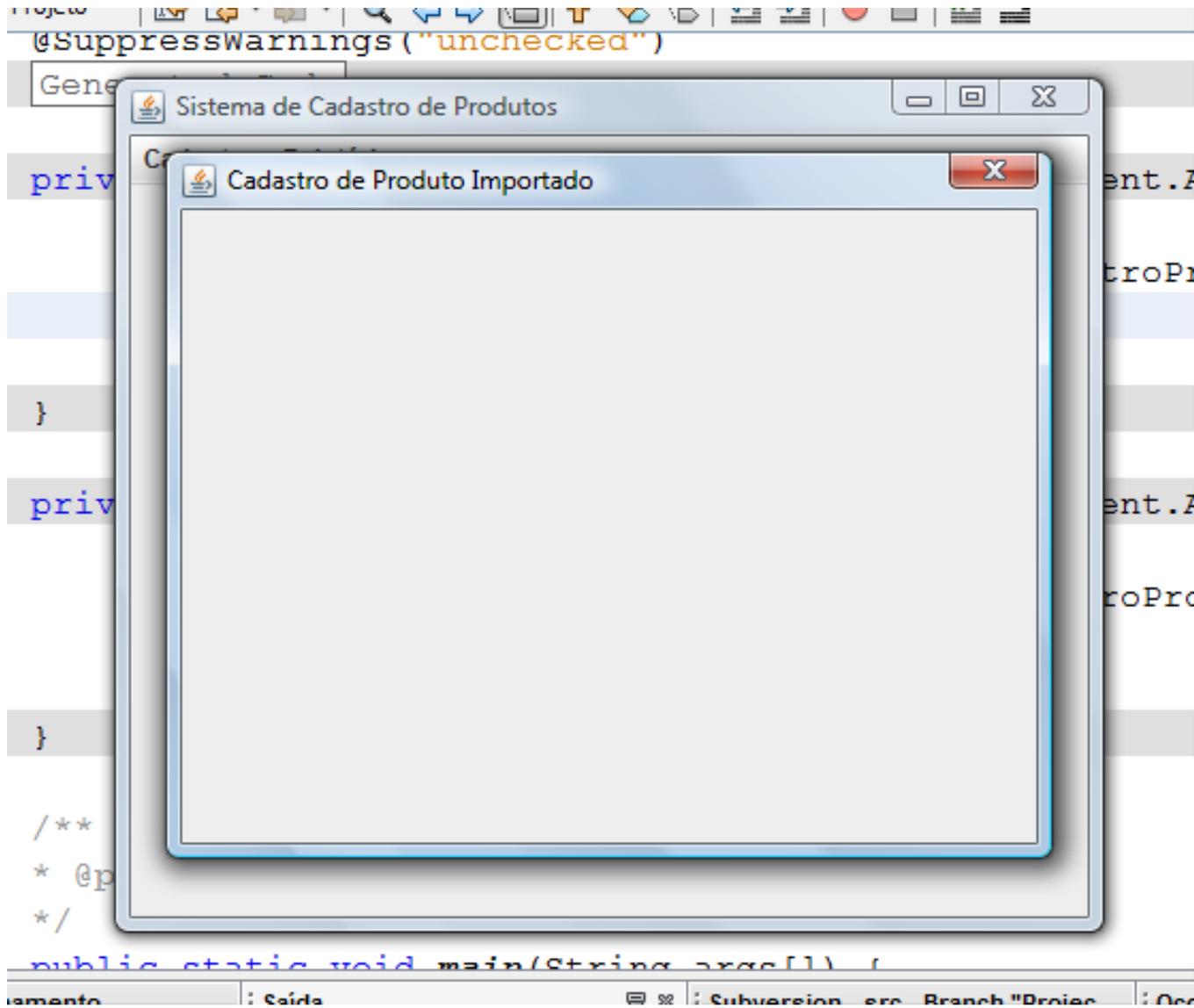
- Instanciar IU de cadastro e torná-la visível

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    IUCadastroProdutoImportado IUCad = new IUCadastroProdutoImportado(this, true);  
    IUCad.setLocationRelativeTo(this);  
    IUCad.setVisible(true);  
}
```

Executando

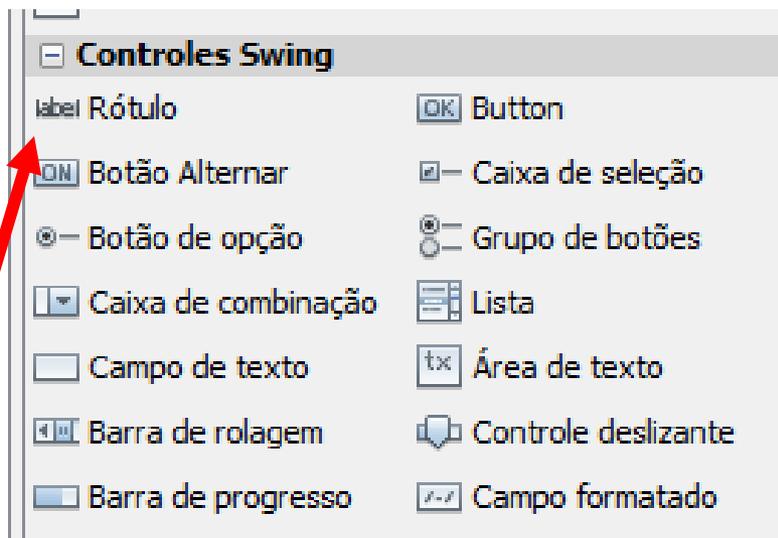


Executando



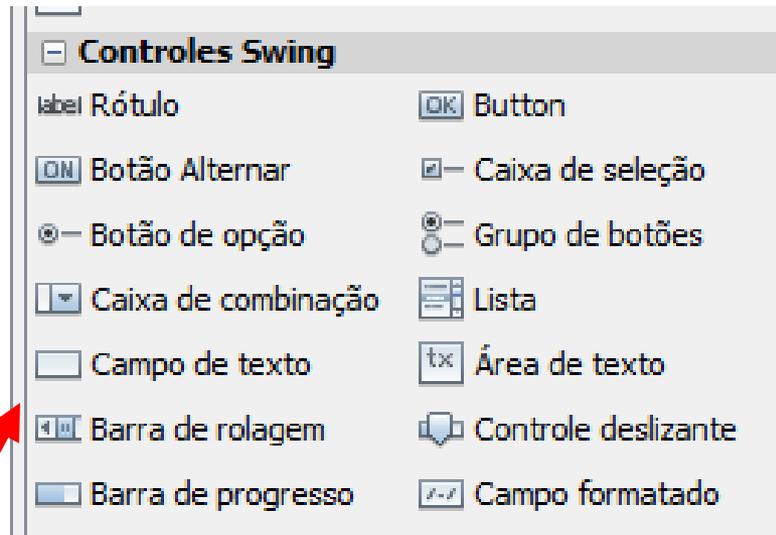
Adicionando Caixas de Texto e Botões

Componentes



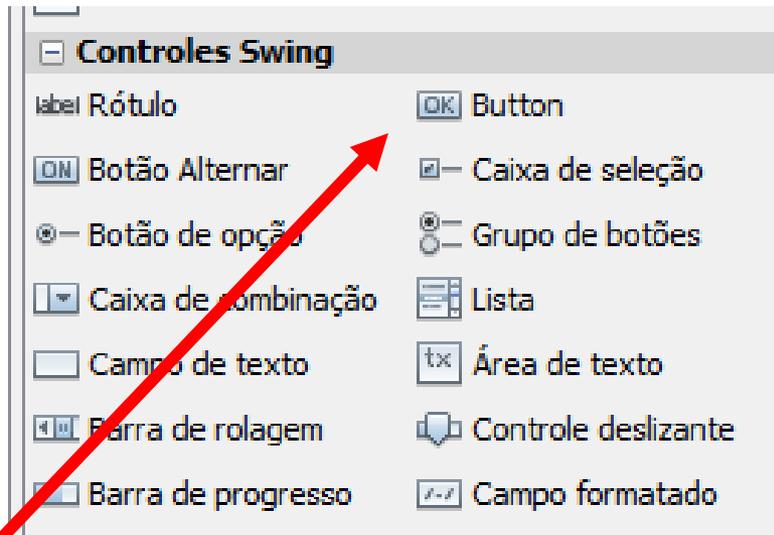
Utilizado para identificar itens no formulário
Pode ser utilizado para saída de dados
O usuário não pode alterar o conteúdo dele

Componentes



Utilizado para entrada e saída de dados
O usuário pode alterar o conteúdo dele

Componentes

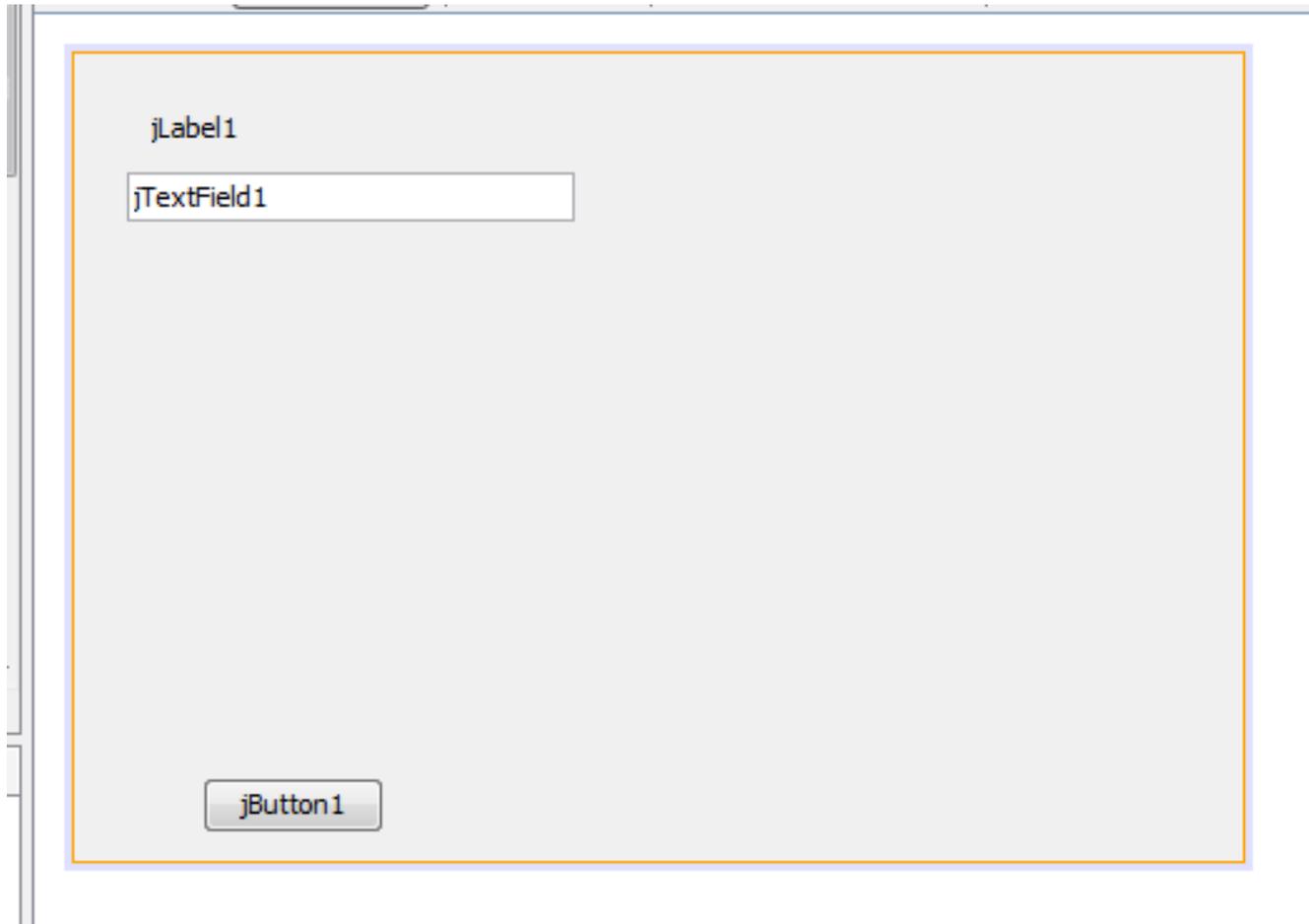


Botão – utilizado para executar alguma ação

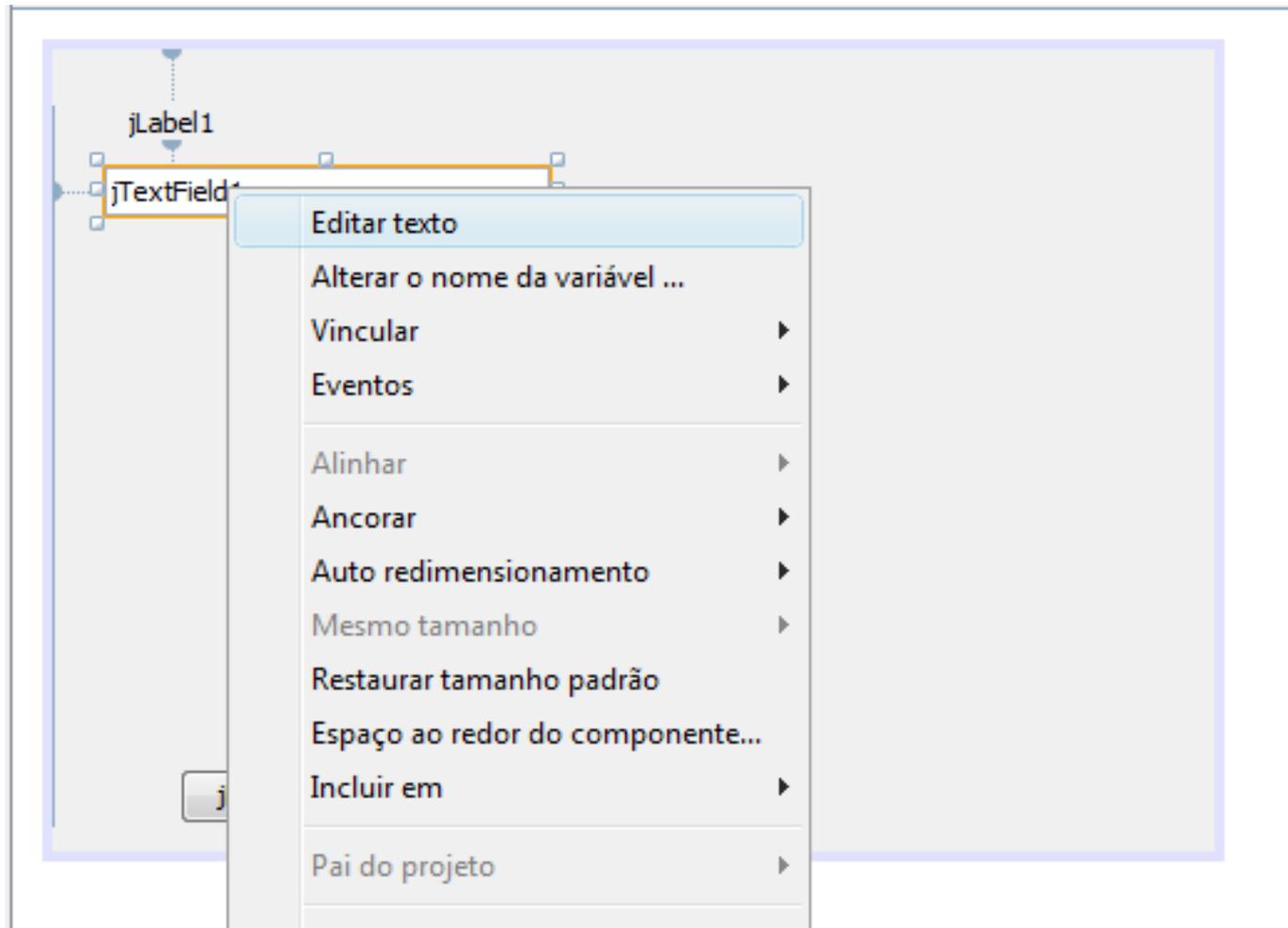
Adicionando Componentes

- Clique no componente desejado e arraste até o formulário
- Altere o conteúdo dele
- Altere o nome da variável (altamente recomendável para facilitar a programação)
 - Tudo isso é realizado por meio do clique com o botão direito sobre o componente e depois escolhendo a opção desejada

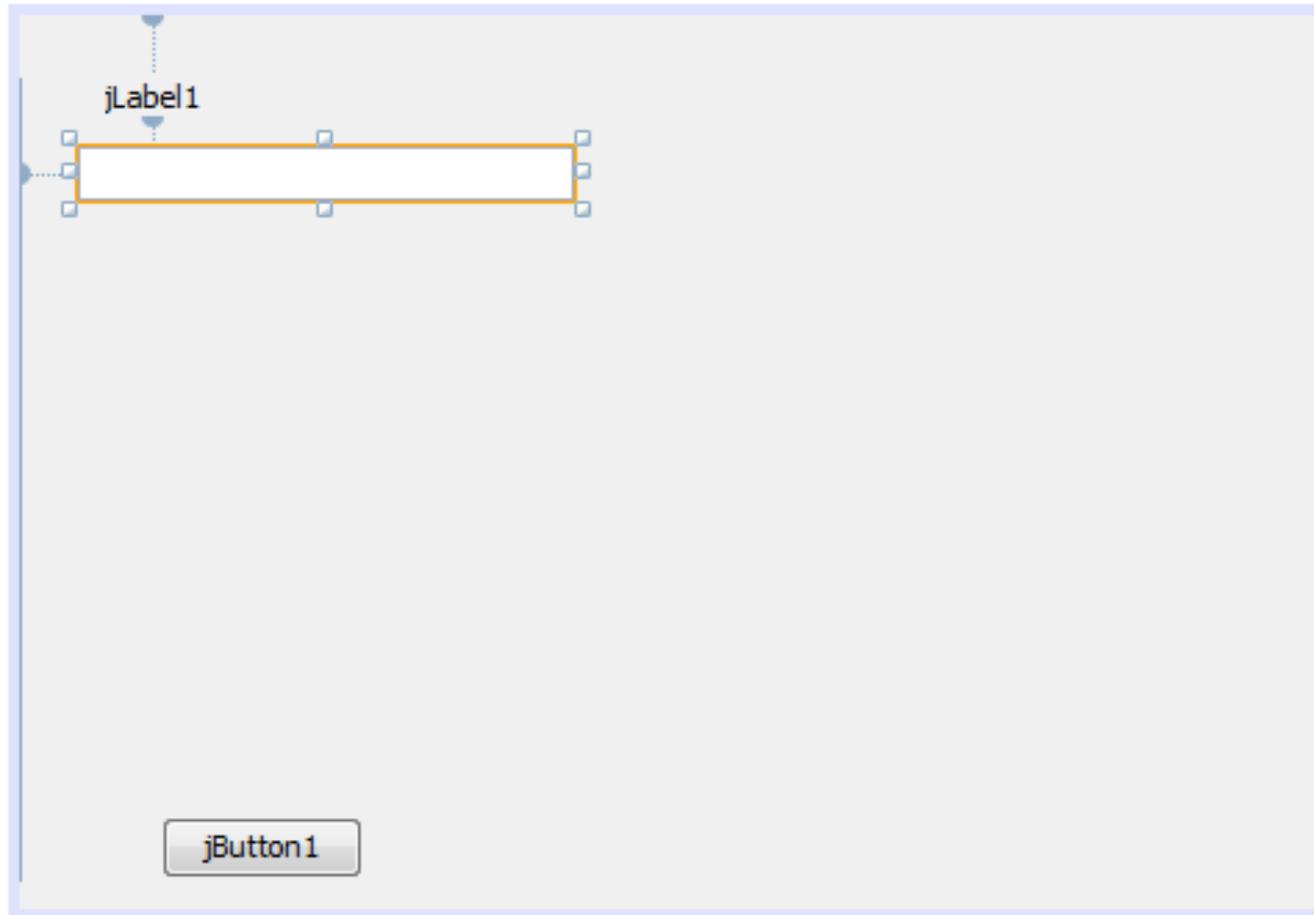
Adicionando Componentes



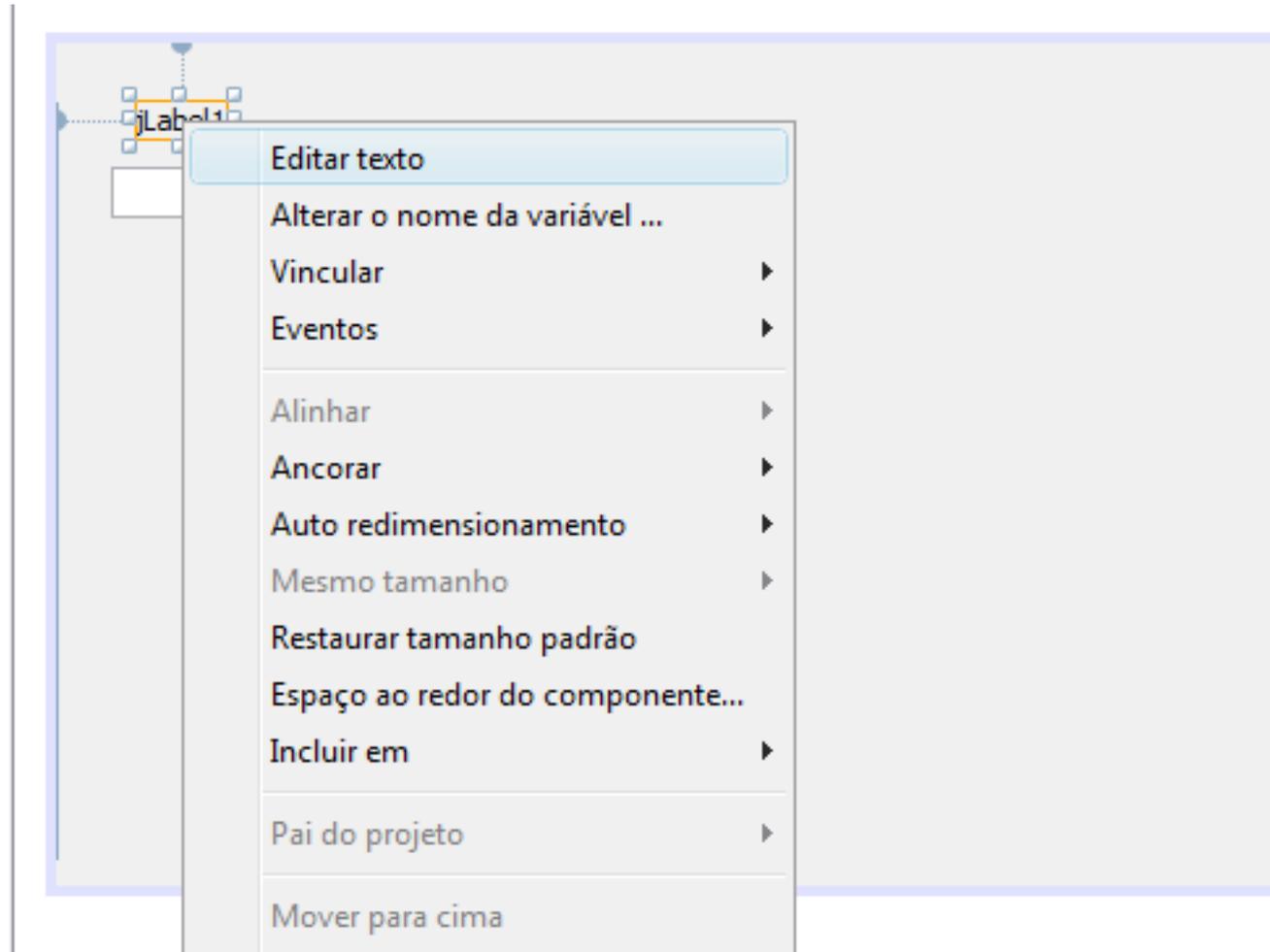
Adicionando Componentes



Adicionando Componentes



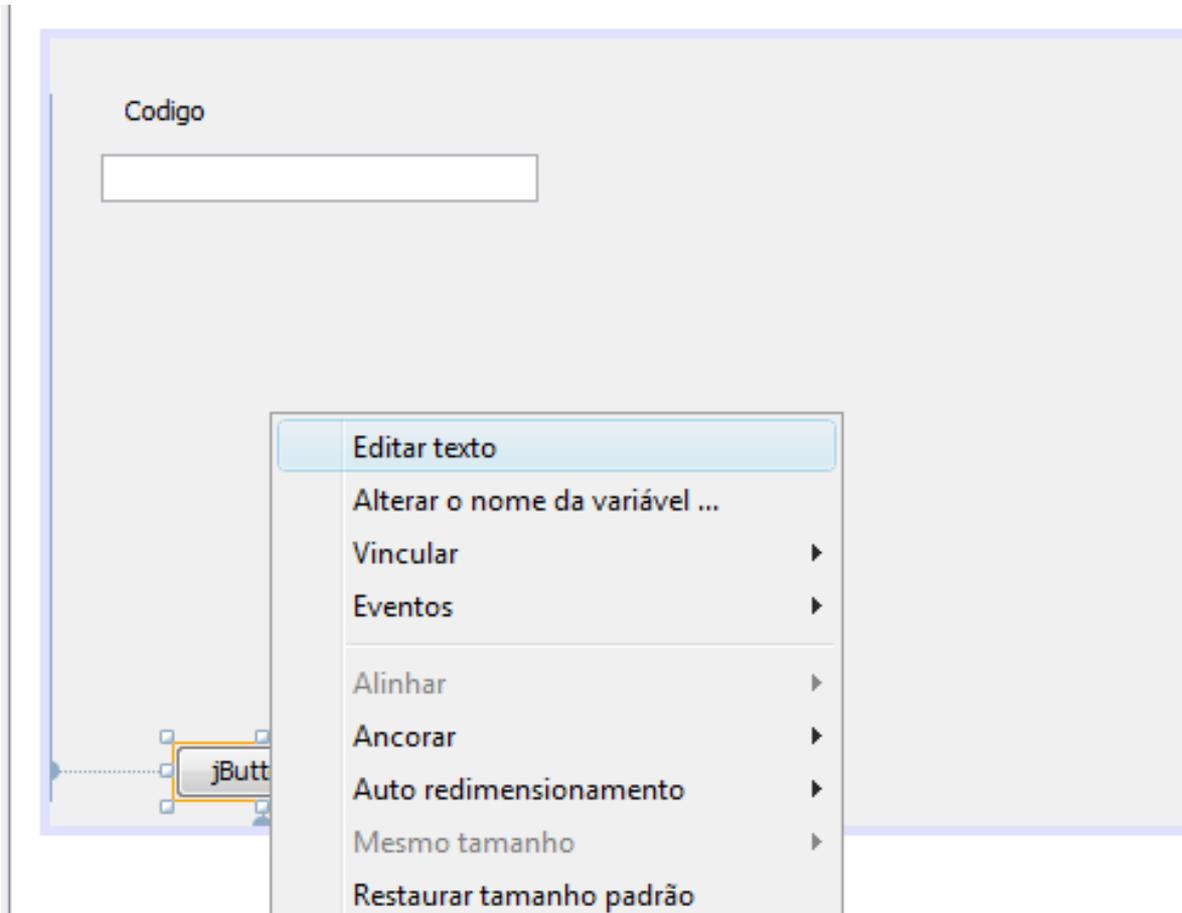
Adicionando Componentes



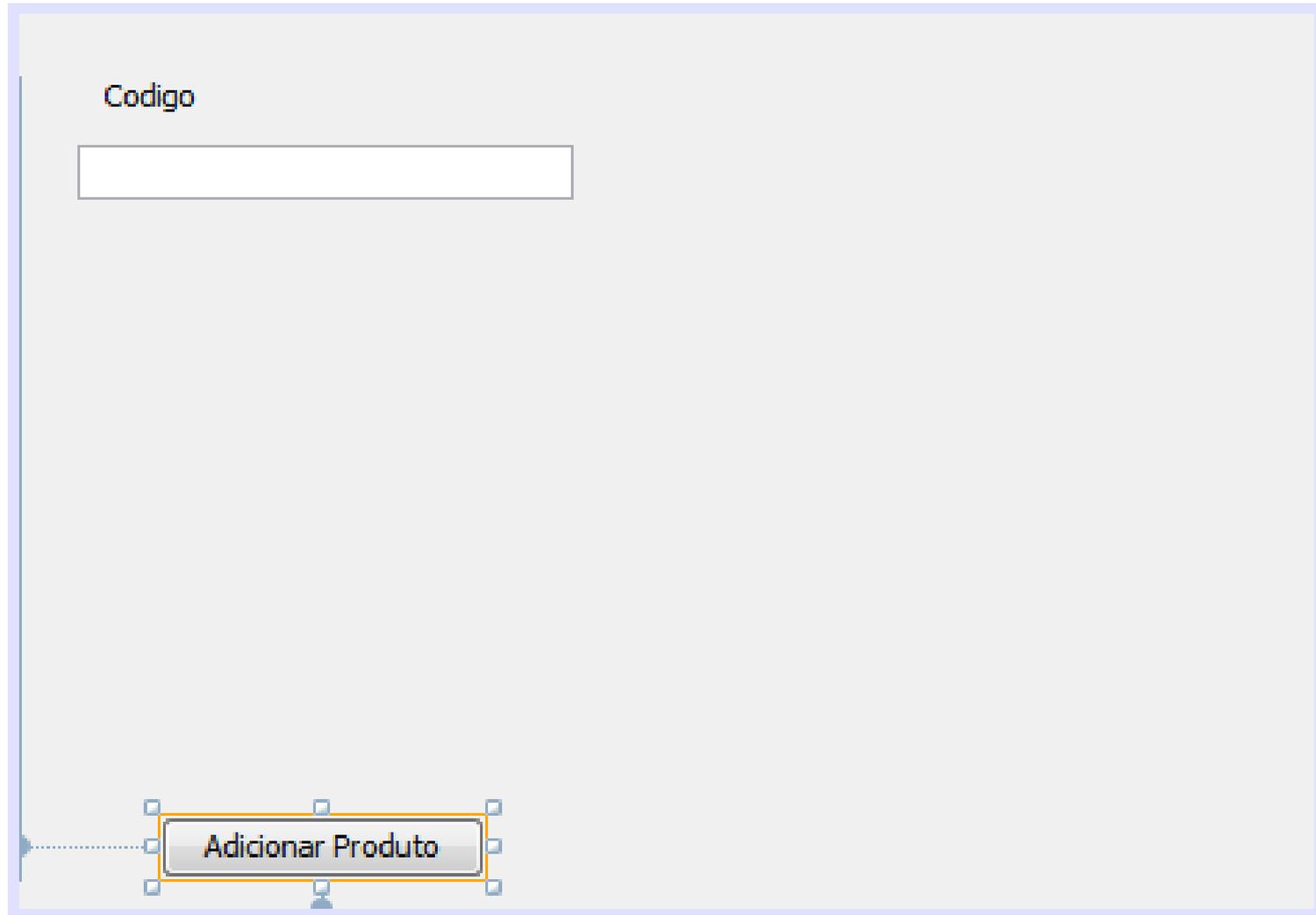
Adicionando Componentes



Adicionando Componentes

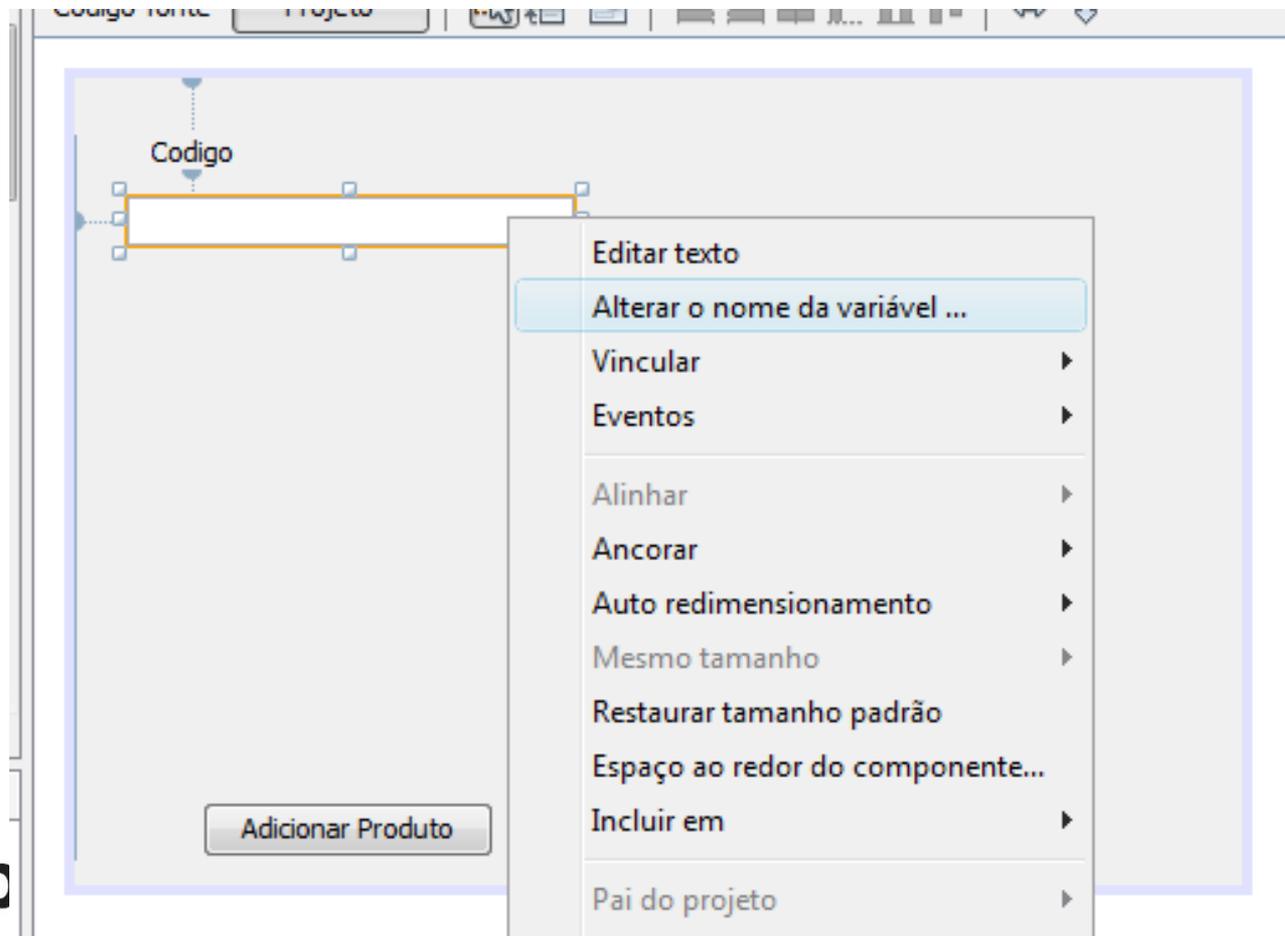


Adicionando Componentes



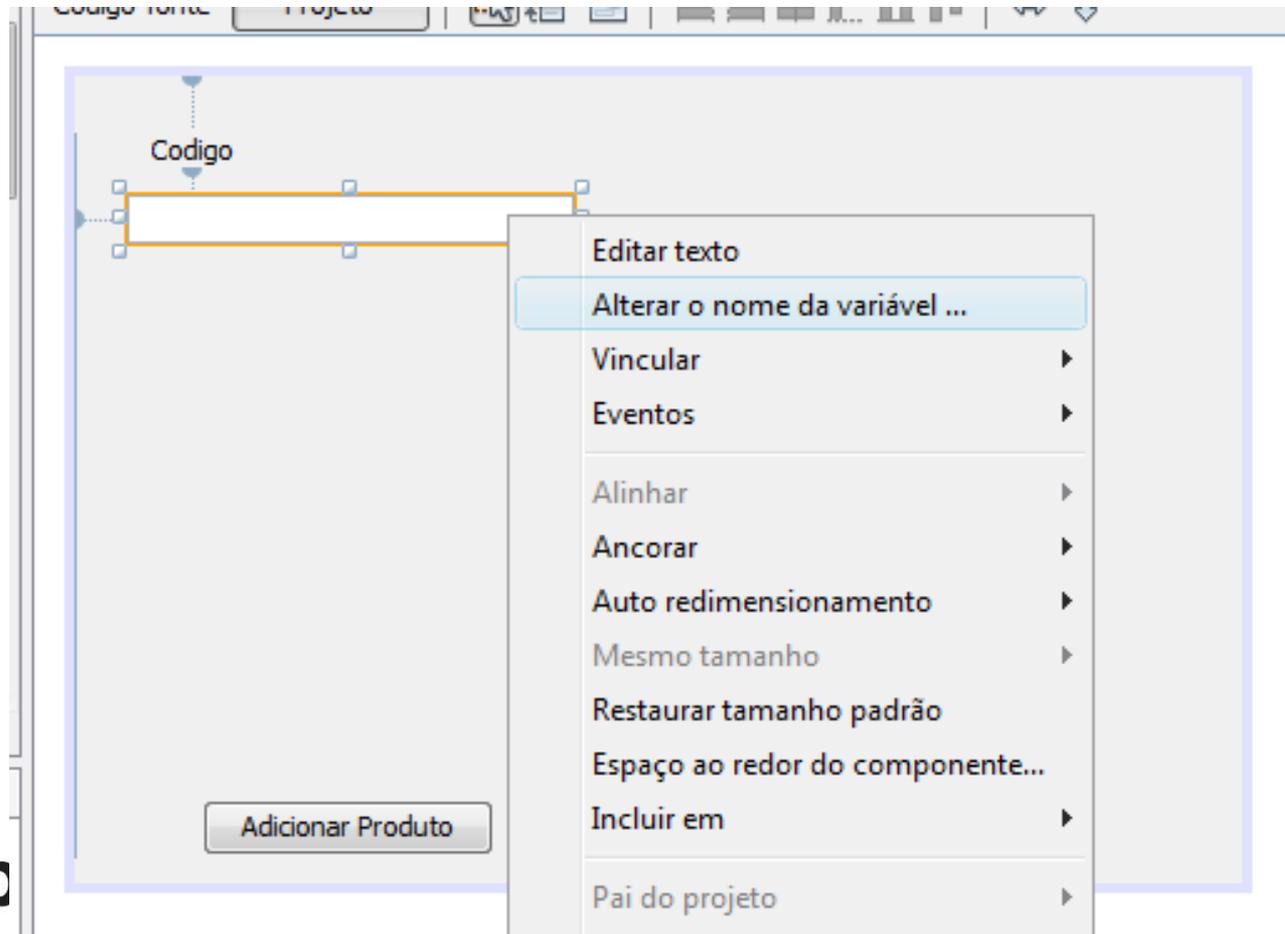
Adicionando Componentes

- Alterar nome das variáveis que serão utilizadas na programação



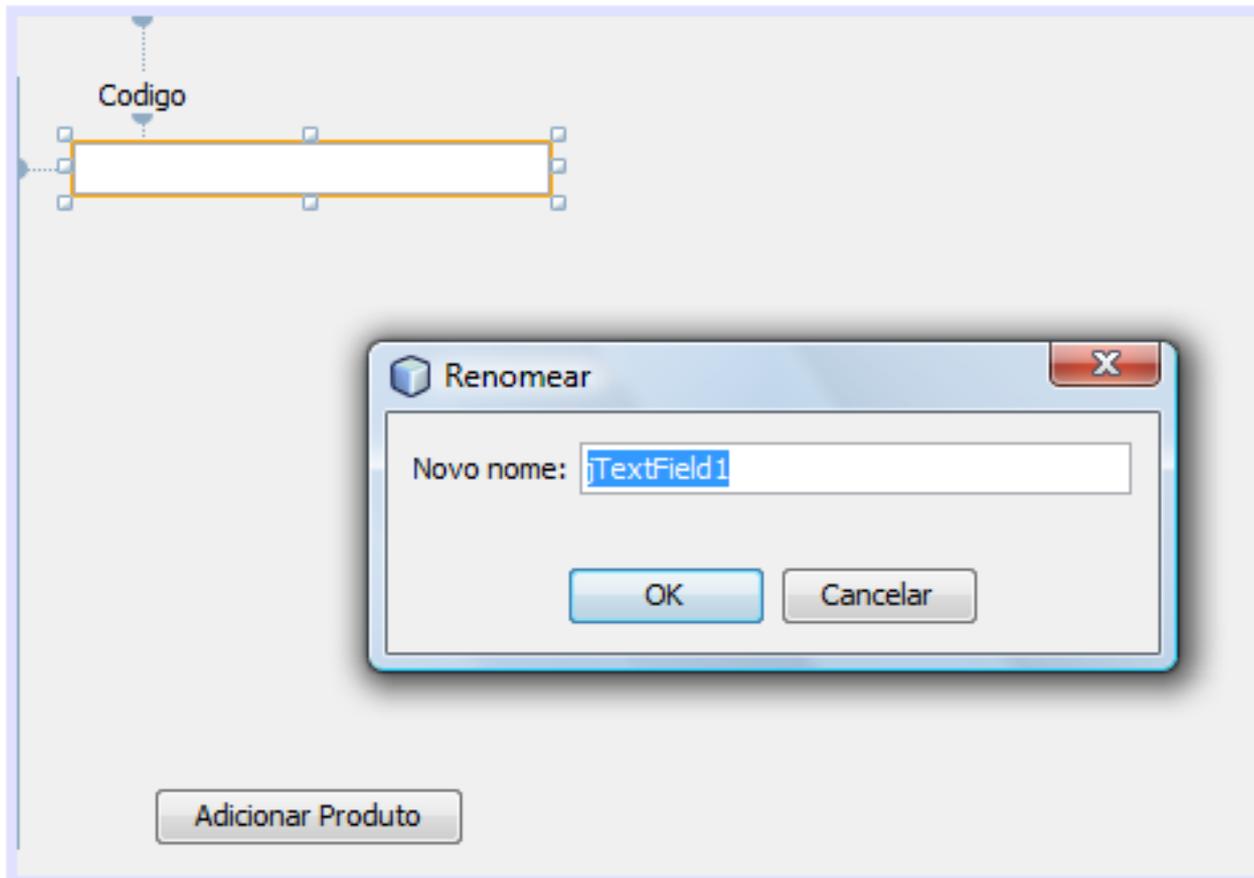
Adicionando Componentes

- Alterar nome das variáveis que serão utilizadas na programação



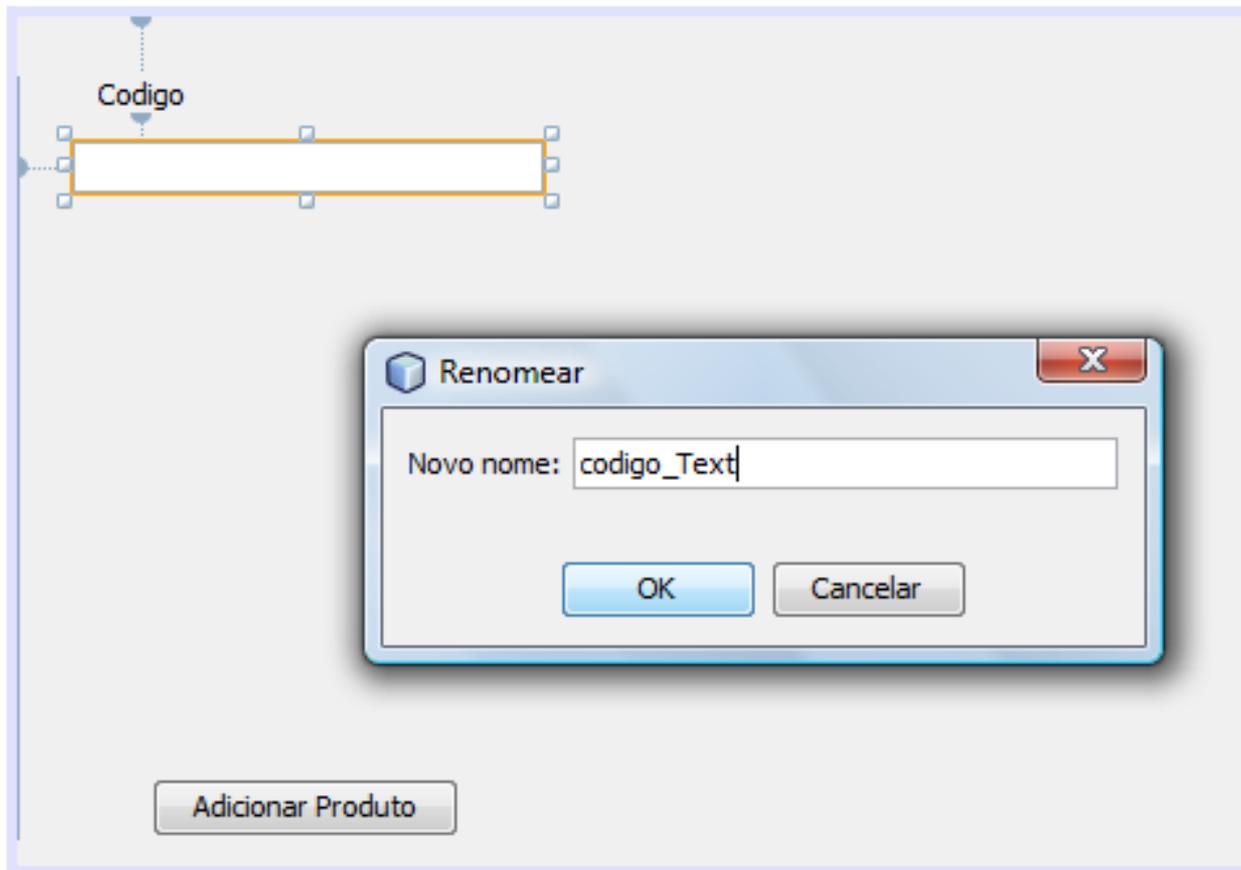
Adicionando Componentes

- Alterar nome das variáveis que serão utilizadas na programação



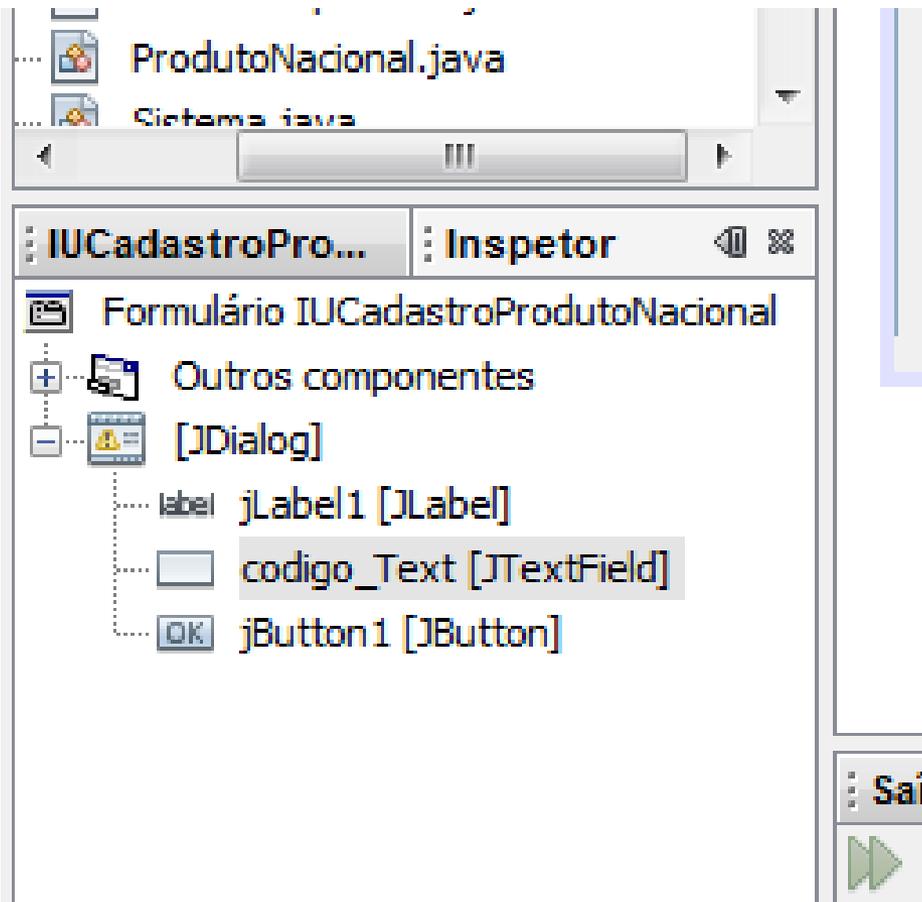
Adicionando Componentes

- Alterar nome das variáveis que serão utilizadas na programação



Adicionando Componentes

- Os componentes também são exibidos aqui aqui



temaBancario
cotes de código-fonte
liotecas
:caJFuzzyLogic
pl0Aula09
cotes de código-fonte
exemploaula09
Main.java
exemploaula09.iu
IUCadastroProdutoImportado.java
IUCadastroProdutoNacional.java
IUPrincipal.java
exemploaula09.modelo
Produto.java
ProdutoImportado.java
ProdutoNacional.java
Sistema.java

Código-fonte Projeto

Código

Adicionar Produto

IUCadastroPro... : Inspeção

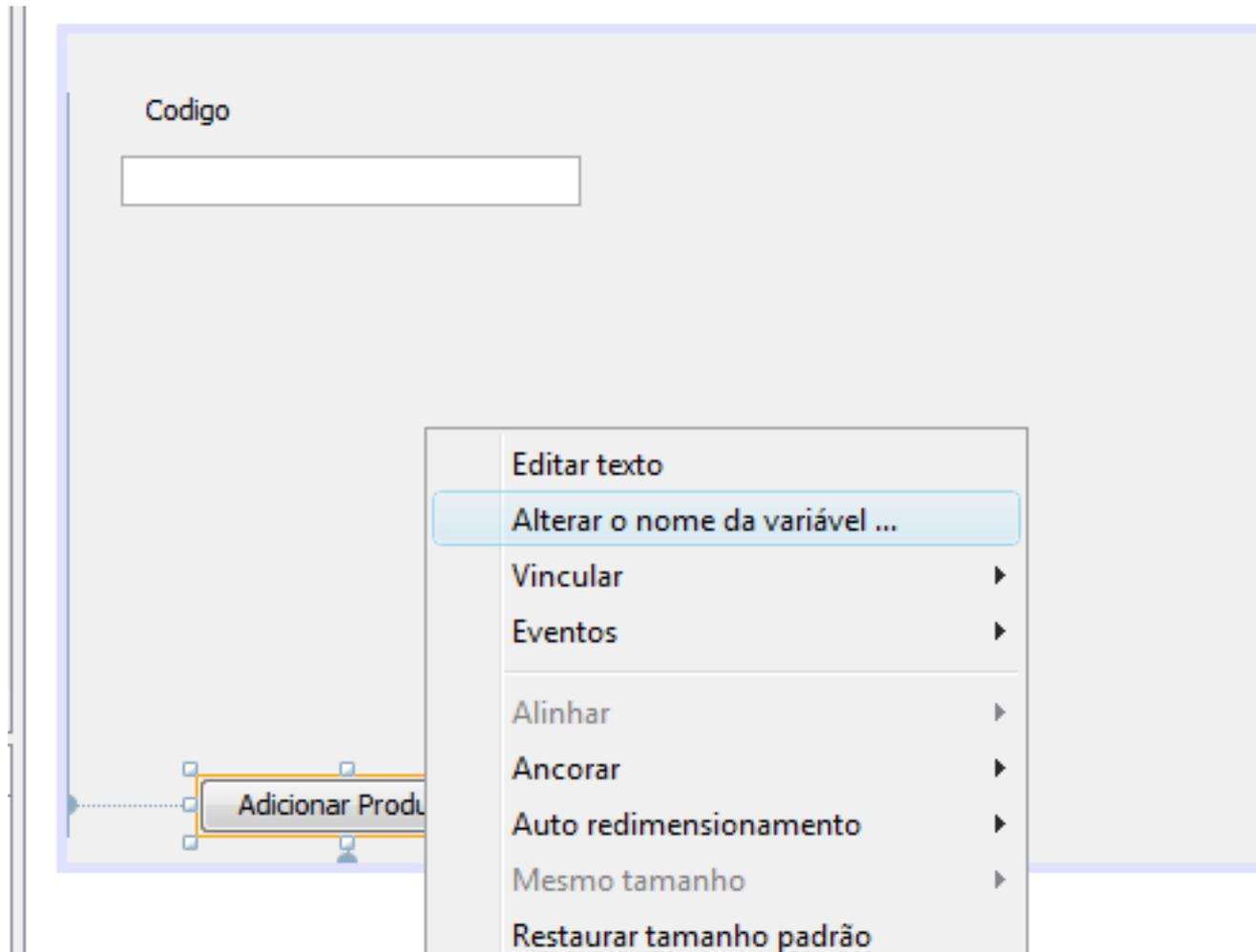
Formulário IUCadastroProdutoNacional

- Outros componentes
- [JDialog]
- label jLabel1 [JLabel]
- codigo_Text [JTextField]
- OK jButton1 [JButton]

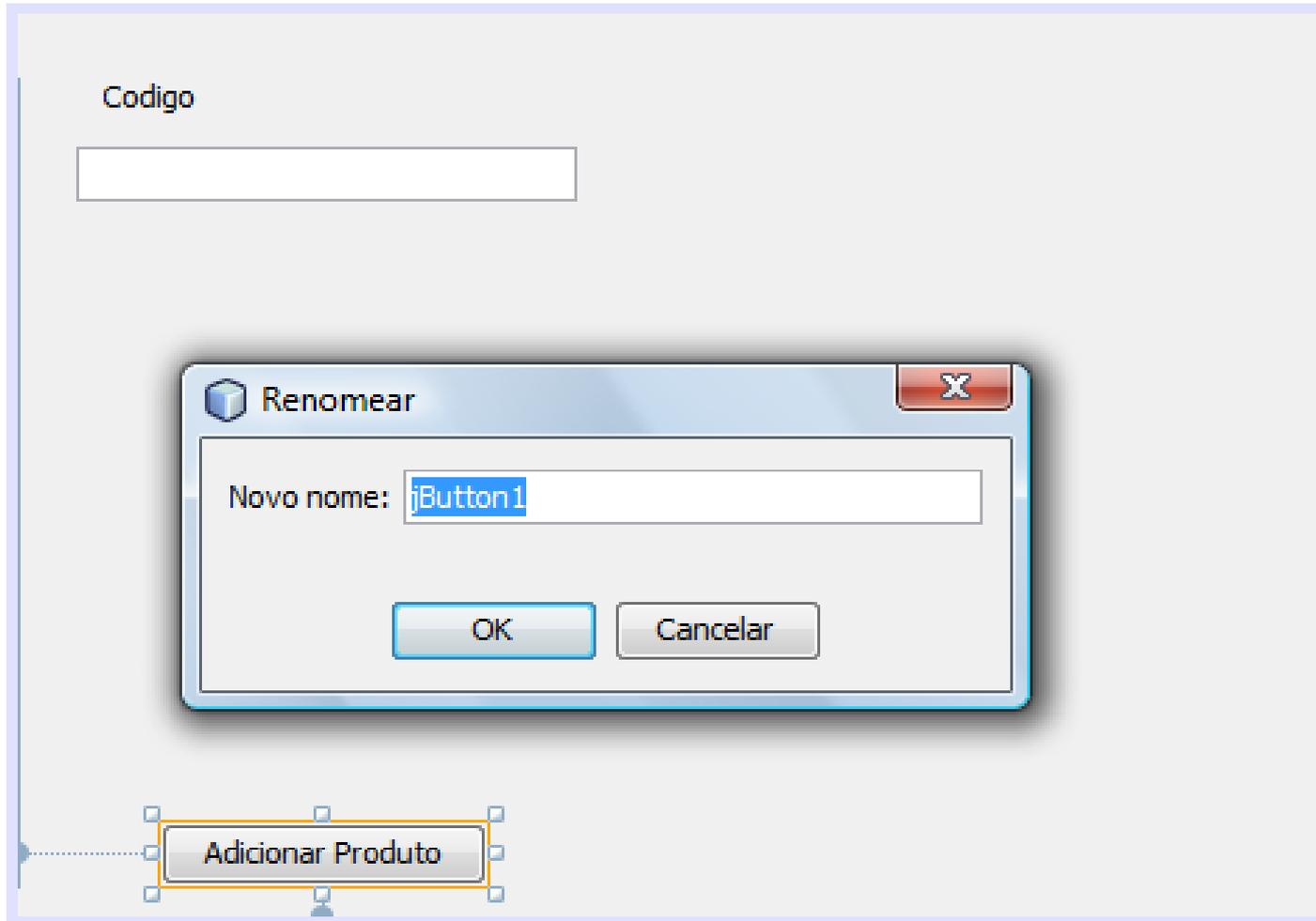
Saída do versionamento Saída

S

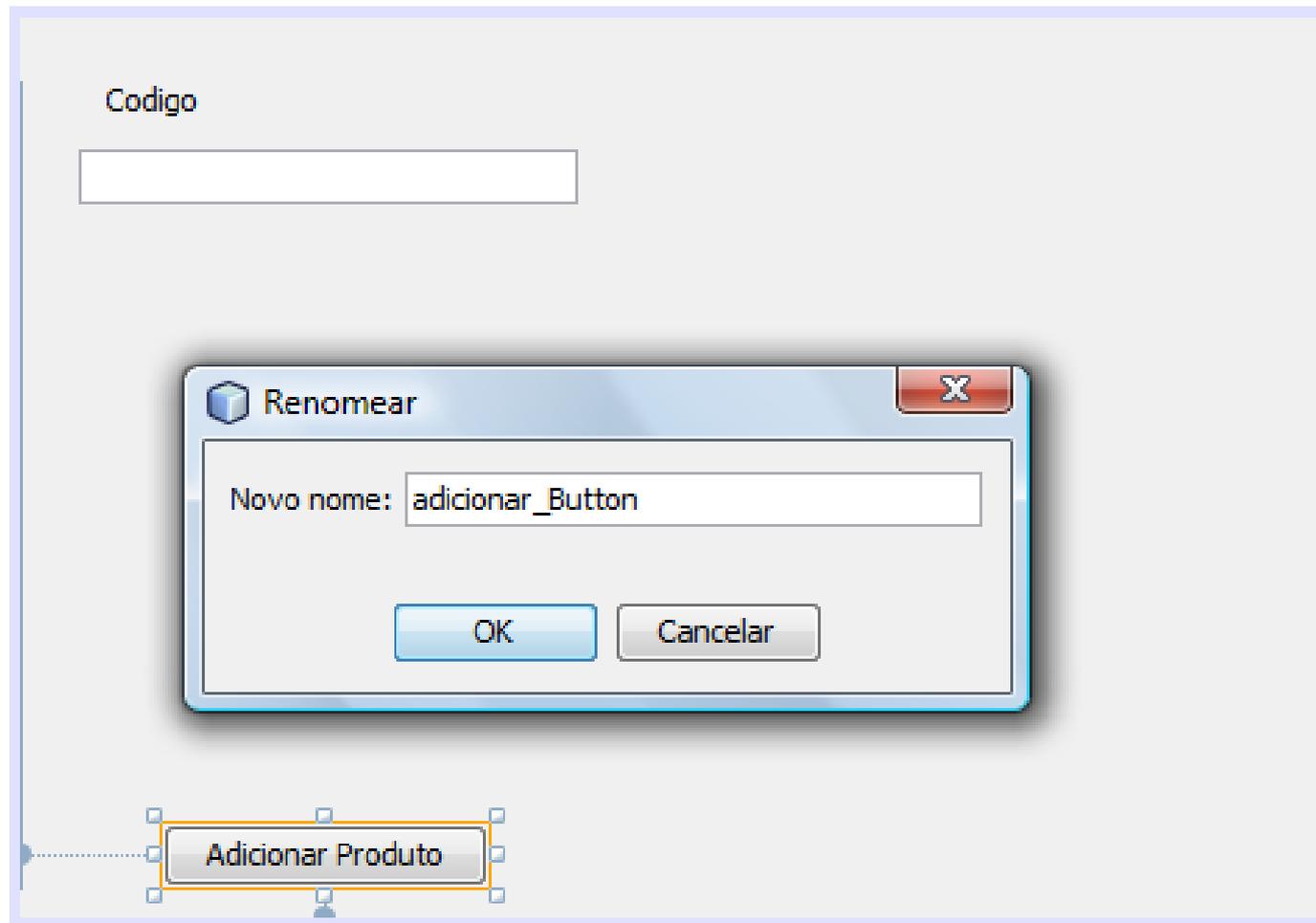
Adicionando Componentes



Adicionando Componentes



Adicionando Componentes



istemaBancario

acotes de código-fonte

ibliotecas

tecaJFuzzyLogic

mploAula09

acotes de código-fonte

- exemploaula09
 - Main.java
- exemploaula09.iu
 - IUCadastroProdutoImportado.java
 - IUCadastroProdutoNacional.java
 - IUPrincipal.java
- exemploaula09.modelo
 - Produto.java
 - ProdutoImportado.java
 - ProdutoNacional.java
 - Sistema.java

IUCadastroPro... Inspeção

Formulário IUCadastroProdutoNacional

Outros componentes

- [JDialog]
 - label jLabel1 [JLabel]
 - codigo_Text [JTextField]
 - adicionar_Button [JButton]

Código-fonte Projeto

Codigo

Adicionar Produto

Saída do versionamento Saída

Componetes Produto Nacional

Sistema.java
Pacotes de testes

IUCadastroPro... : Inspetor

Formulário IUCadastroProdutoNacional

- Outros componentes
- [JDialog]
 - label jLabel1 [JLabel]
 - codigo_Text [JTextField]
 - adicionar_Button [JButton]
 - label jLabel2 [JLabel]
 - nome_Text [JTextField]
 - label jLabel3 [JLabel]
 - valor_Text [JTextField]
 - label jLabel4 [JLabel]
 - taxa_Text [JTextField]

Codigo Descricao

Valor Taxa

Adicionar Produto

Saída do versionamento Saída Subversi

ExemploAula09 (run) ExemploAula09 (run) #2 *

run:

Componentes Produto Importado

The component navigator shows the following structure:

- Formulário IUCadastroProdutoImportado
 - Outros componentes
 - [JDialog]
 - label jLabel4 [JLabel]
 - valor_Text [JTextField]
 - label jLabel3 [JLabel]
 - label jLabel2 [JLabel]
 - taxa_Text [JTextField]
 - nome_Text [JTextField]
 - codigo_Text [JTextField]
 - adicionar_Button [JButton]
 - label jLabel1 [JLabel]
 - label jLabel5 [JLabel]
 - taxaImportacao_Text [JTextField]

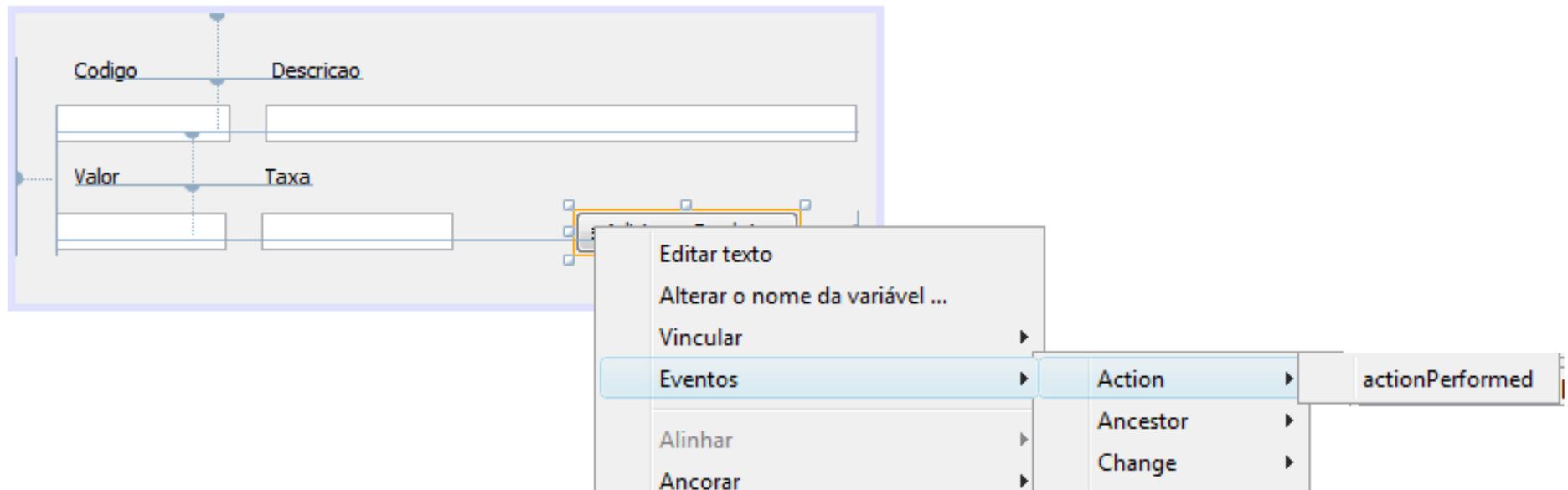
The window contains the following components:

- Label: Código
- Text Field: [Empty]
- Label: Descrição
- Text Field: [Empty]
- Label: Valor
- Text Field: [Empty]
- Label: Taxa
- Text Field: [Empty]
- Label: Taxa Importacao
- Text Field: [Empty]
- Button: Adicionar Produto

The console window shows the following output:

```
run:
```

Adicionando Evento do Botão Cadastro Nacional



Adicionando Evento do Botão Cadastro Nacional

- Recupera os dados das caixas de texto e invoca o método `addProdutoNacional` da classe `ControladorProdutos` (variável *sistema*), que é uma instância estática na classe `IUPrincipal`

```
private void adicionarButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    String codigo = codigoText.getText();  
    String descricao = descricaoText.getText();  
    float valor = Float.parseFloat(valorText.getText());  
    float taxaImposto = Float.parseFloat(taxaText.getText());  
    Controlador control = new Controlador();  
    control.addProdutoEstadual(codigo, valor, taxaImposto);  
}
```

Adicionando Evento do Botão Cadastro Importado

The image shows a Java Swing IDE interface. On the left, a form is displayed with the following fields:

- Codigo**: A text input field.
- Descricao**: A text input field.
- Valor**: A text input field.
- Taxa**: A text input field.
- Taxa Importacao**: A text input field.

Below the form is a button labeled **Adicionar Produto**. A context menu is open over this button, showing the following options:

- Editar texto
- Alterar o nome da variável ...
- Vincular
- Eventos** (highlighted) → **Action** (highlighted) → **actionPerformed** (highlighted)
- Ancestor
- Classes
- Alinhar

On the right side of the IDE, there is a palette titled **Controles Swing** with the following items:

- label Rótulo
- Botão Alternar
- Botão de opção
- Caixa de combinação
- Campo de texto
- Barra de rolagem
- Barra de progresso

Below the palette, the component **adicionar_Button [JButton]** is selected, and its **Propriedades** (Properties) window is open, showing the **actionPerformed** property.

Adicionando Evento do Botão Cadastro Importado

- Recupera os dados das caixas de texto e invoca o método `addProdutoImportado` da classe `ControladorProdutos` (variável *sistema*), que é uma instância estática na classe `IUPrincipal`

```
private void adicionarButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    String codigo = codigoText.getText();  
    String descricao = descricaoText.getText();  
    float valor = Float.parseFloat(valorText.getText());  
    float taxaImposto = Float.parseFloat(taxaText.getText());  
    float taxaImportacao = Float.parseFloat(taxaImportacaoText.getText());  
    Controlador control = new Controlador();  
    control.addProdutoNacional(codigo, descricao, valor, taxaImposto, taxaImportacao);  
}
```

Adicionando Evento do Botão

- Da maneira que o código está especificado é possível inserir vários produtos iguais, clicando várias vezes no botão
- Para evitar podemos:
 - Fechar a janela ao cadastrar
 - `this.setVisible(false);` --- esconde a janela, mas o objeto ainda existe
 - `this.dispose();` --- elimina o objeto da memória
 - Limpar os campos ao cadastrar e limitar para não inserir campos vazios
 - Verificar se o código do produto já existe, utilizando o método de busca implementado na classe Sistema

Exemplo com dispose

```
private void adicionarButtonActionPerformed(java.awt.event.ActionEvent evt) {  
    String codigo = codigoText.getText();  
    String descricao = descricaoText.getText();  
    float valor = Float.parseFloat(valorText.getText());  
    float taxaImposto = Float.parseFloat(taxaText.getText());  
    float taxaImportacao = Float.parseFloat(taxaImportacaoText.getText());  
    Controlador control = new Controlador();  
    control.addProdutoNacional(codigo, descricao, valor, taxaImposto, taxaImportacao);  
    this.dispose();  
}
```

Estratégia para evitar cadastro de produtos com mesmo código

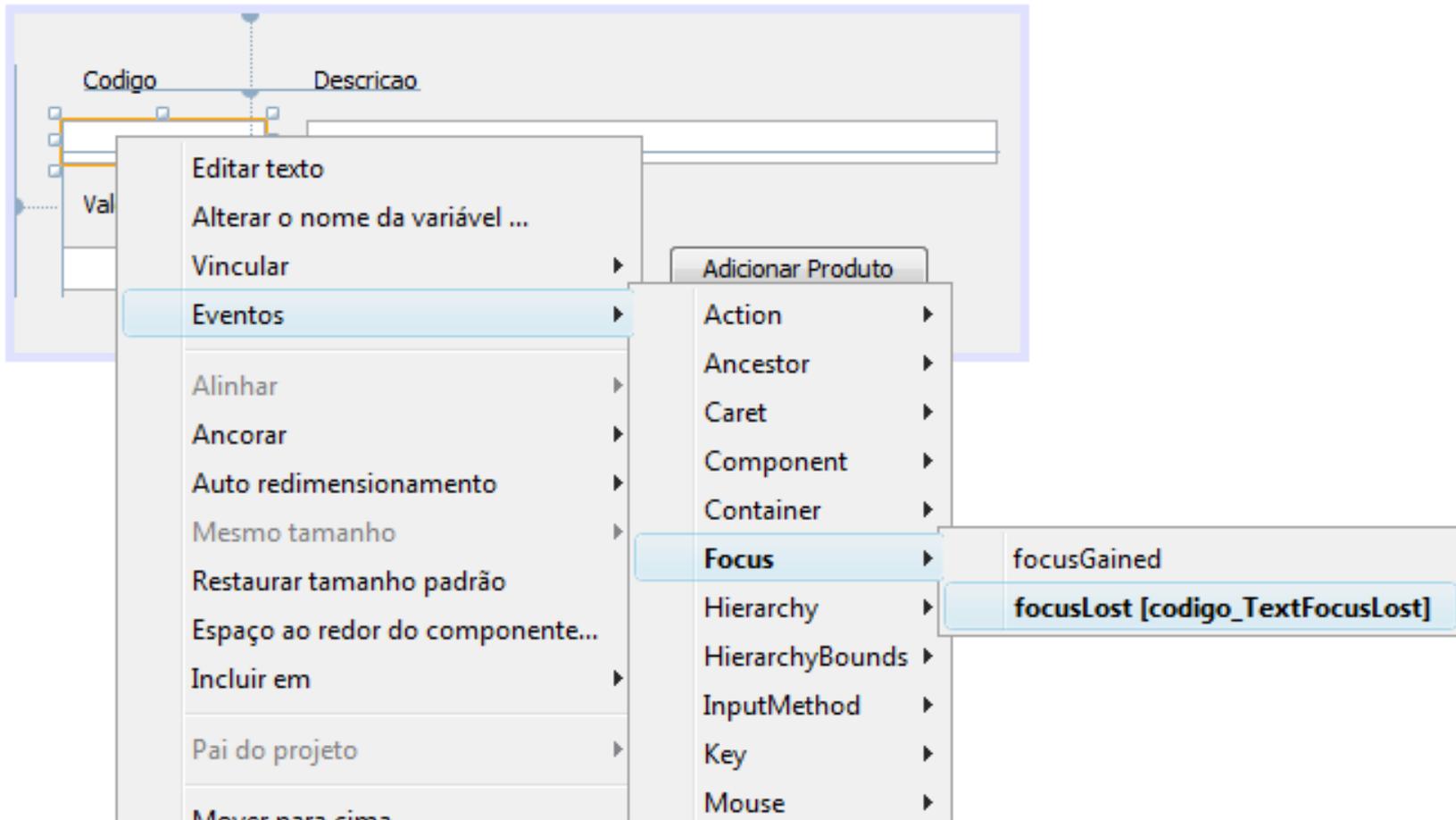
Estratégia para evitar cadastro de produtos com mesmo código

- Quando for digitado o código de um produto faremos uma busca pelo código
 - Se o código existir, exibiremos os seus dados e bloquearemos o botão Adicionar
 - `adicionar_Button.setEnabled(false);`
 - Caso contrário, limparemos os campos e habilitaremos o botão Adicionar
 - `adicionar_Button.setEnabled(true);`

Estratégia para evitar cadastro de produtos com mesmo código

- Utilizaremos o evento LostFocus da caixa de texto do botão
- Sempre que ela perder o foco, faremos a busca pelo código digitado

Estratégia para evitar cadastro de produtos com mesmo código

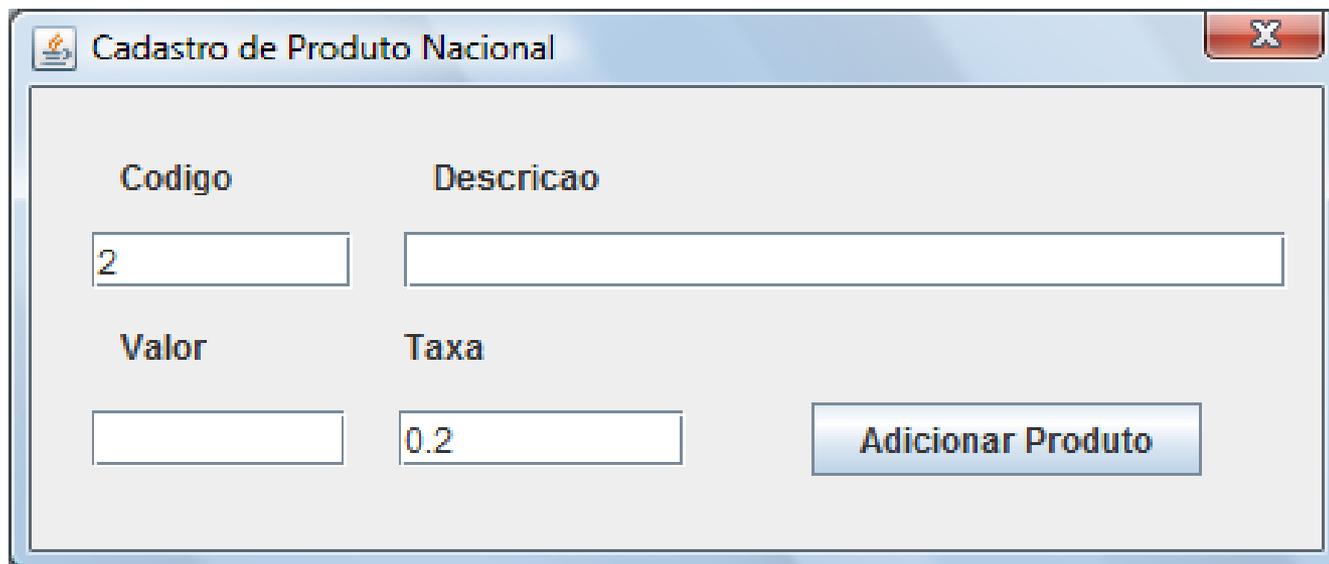


Estratégia para evitar cadastro de produtos com mesmo código

```
private void codigo_TextFocusLost(java.awt.event.FocusEvent evt) {  
    // TODO add your handling code here:  
    String codigo = codigo_Text.getText();  
    Controlador control = new Controlador();  
    Produto produto = control.buscarProdutoPorCodigo(codigo);  
    if (produto != null){  
        descricao_Text.setText(produto.getDescricao());  
        valor_Text.setText( Float.toString( produto.getValor() ) );  
        float taxaImposto = ((ProdutoNacional) produto).getTaxaImposto();  
        taxa_Text.setText(Float.toString( taxaImposto) );  
        adicionar_Button.setEnabled(false);  
    }else{  
        valor_Text.setText("");  
        descricao_Text.setText("");  
        adicionar_Button.setEnabled(true);  
    }  
}
```

Executando

- Produto não cadastrado



The image shows a screenshot of a software application window titled "Cadastro de Produto Nacional". The window contains a form with the following fields and controls:

Codigo	Descricao
<input type="text" value="2"/>	<input type="text"/>
Valor	Taxa
<input type="text"/>	<input type="text" value="0.2"/>

Adicionar Produto

Executando

- Produto não cadastrado

Cadastro de Produto Nacional

Codigo	Descricao
2	
Valor	Taxa
	0.2

Adicionar Produto

```
}else{  
    valor_Text.setText("");  
    descricao_Text.setText("");  
    adicionar_Button.setEnabled(true);  
}
```

Executando

- Produto nacional já cadastrado

Cadastro de Produto Nacional

Codigo	Descricao
1	PRoduto 1
Valor	Taxa
1.0	0.2

Adicionar Produto

```
if (produto != null){
    descricao_Text.setText (produto.getDescricao());
    valor_Text.setText( Float.toString( produto.getValor() ) );
    float taxaImposto = ((ProdutoNacional) produto).getTaxaImposto();
    taxa_Text.setText(Float.toString( taxaImposto) );
    adicionar_Button.setEnabled(false);
}
```

Estratégia para evitar cadastro de produtos com mesmo código

- Um JLabel pode ser utilizado para exibir mensagens para o usuário, assim evitamos de utilizar o JOptionPane
 - A mensagem pode ser passada para ele e torná-lo visível
 - `msg_Label.setVisible(true);`
 - Caso não seja necessário uma mensagem, ele se torna invisível
 - `msg_Label.setVisible(false);`

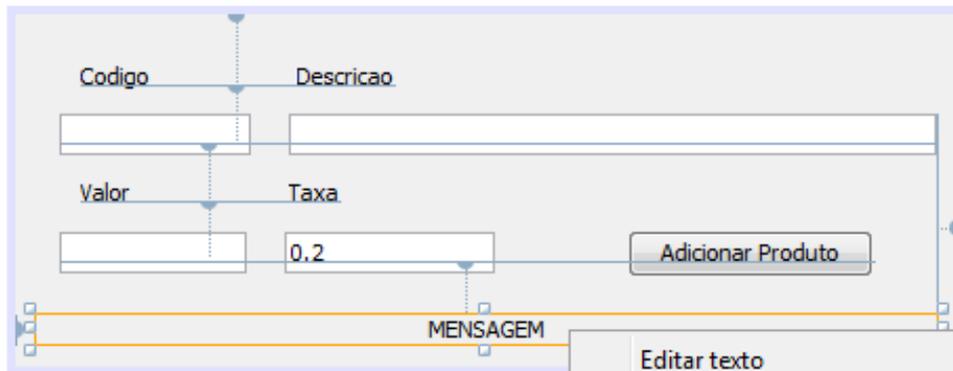
Alterar Nome de Variável e Conteúdo

The screenshot shows a software interface with a form containing the following elements:

- Field labeled "Codigo" with an empty text input box.
- Field labeled "Descricao" with an empty text input box.
- Field labeled "Valor" with an empty text input box.
- Field labeled "Taxa" with a text input box containing the value "0.2".
- A button labeled "Adicionar Produto" to the right of the "Taxa" field.
- A horizontal orange line at the bottom of the form area with the label "MENSAGEM" centered below it.

The screenshot shows a dialog box titled "Renomear" with the following elements:

- A text input field labeled "Novo nome:" containing the text "msg_Label".
- An "OK" button.
- A "Cancelar" button.

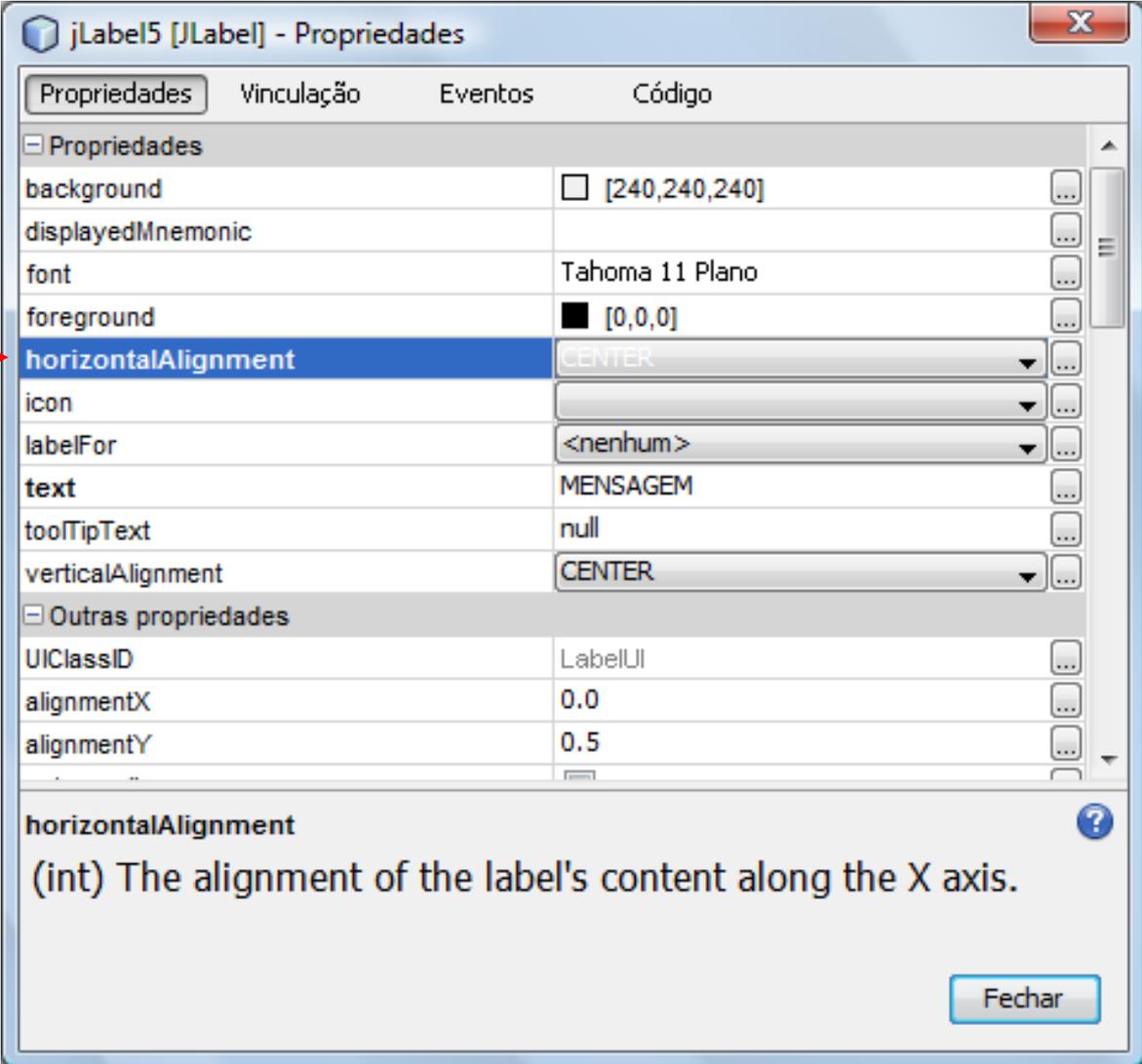


- Editar texto
- Alterar o nome da variável ...
- Vincular ▶
- Eventos ▶
- Alinhar ▶
- Ancorar ▶
- Auto redimensionamento ▶
- Mesmo tamanho ▶
- Restaurar tamanho padrão
- Espaço ao redor do componente...
- Incluir em ▶
- Pai do projeto ▶
- Mover para cima
- Mover para baixo
- Recortar
- Copiar
- Duplicar
- Excluir
- Personalizar código
- Propriedades

Saída do versionamento : Saída - Exem

```
> Taxa: 0.2  
> Valor Final: R$ 2.4  
Codigo: 3  
Descricao: 3  
Valor: R$ 3.0  
Taxa: 0.1  
Taxa Importacao: 0.2
```

Mudar Propriedades



The screenshot shows the 'jLabel5 [JLabel] - Propriedades' dialog box. The 'horizontalAlignment' property is selected and highlighted in blue, with a red arrow pointing to it. The dialog displays various properties and their values:

Propriedade	Valor
background	[240,240,240]
displayedMnemonic	
font	Tahoma 11 Plano
foreground	[0,0,0]
horizontalAlignment	CENTER
icon	
labelFor	<nenhum>
text	MENSAGEM
toolTipText	null
verticalAlignment	CENTER
Outras propriedades	
UIClassID	LabelUI
alignmentX	0.0
alignmentY	0.5

horizontalAlignment
(int) The alignment of the label's content along the X axis.

Fechar

Mudar Propriedades

- No construtor da classe IUCadastroNacional colocar como invisível

– `msg_Label.setVisible(false);`

```
/**
 *
 * @author Danilo Medeiros Eler
 */
public class IUCadastroProdutoNacional extends javax.swing.JDialog {

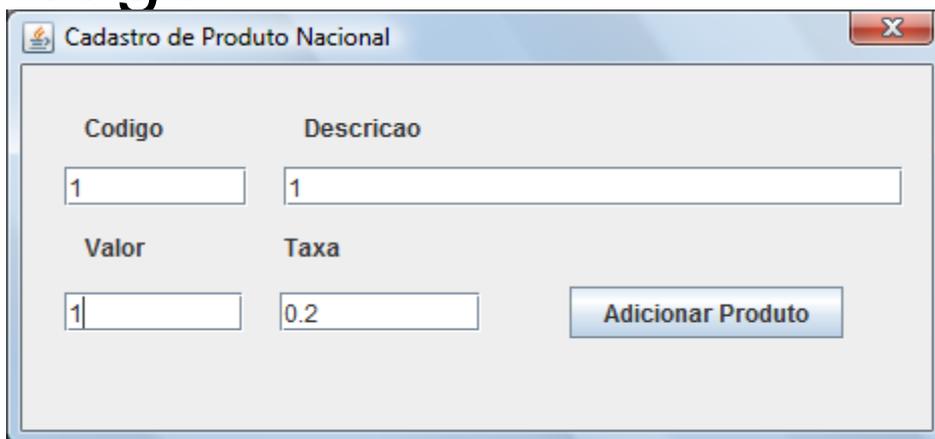
    /** Creates new form IUCadastroProdutoNacional */
    public IUCadastroProdutoNacional(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
        this.msg_Label.setVisible(false);
    }
}
```

Exibindo a Mensagem

```
private void codigo_TextFocusLost(java.awt.event.FocusEvent evt) {  
    // TODO add your handling code here:  
    String codigo = codigo_Text.getText();  
    Controlador control = new Controlador();  
    Produto produto = control.buscarProdutoPorCodigo(codigo);  
    if (produto != null) {  
        descricao_Text.setText(produto.getDescricao());  
        valor_Text.setText( Float.toString( produto.getValor() ) );  
        float taxaImposto = ((ProdutoNacional) produto).getTaxaImposto();  
        taxa_Text.setText(Float.toString( taxaImposto ) );  
        adicionar_Button.setEnabled(false);  
        this.msg_Label.setText("Produto Já Cadastrado");  
        this.msg_Label.setVisible(true);  
    }else{  
        valor_Text.setText("");  
        descricao_Text.setText("");  
        adicionar_Button.setEnabled(true);  
        this.msg_Label.setVisible(false);  
    }  
}
```

Executando

- Produto não cadastrado – Não precisa de mensagem

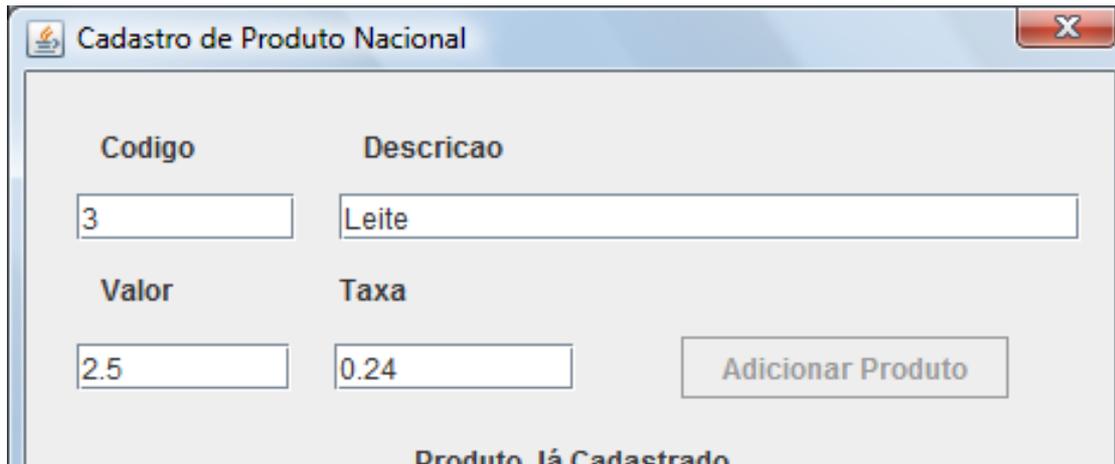


The screenshot shows a window titled "Cadastro de Produto Nacional". It has four text input fields arranged in a 2x2 grid. The top-left field is labeled "Codigo" and contains the number "1". The top-right field is labeled "Descricao" and contains the number "1". The bottom-left field is labeled "Valor" and contains the number "1". The bottom-right field is labeled "Taxa" and contains the number "0.2". To the right of the "Taxa" field is a button labeled "Adicionar Produto".

```
}else{  
    valor_Text.setText("");  
    descricao_Text.setText("");  
    adicionar_Button.setEnabled(true);  
    this.msg_Label.setVisible(false);  
}
```

Executando

- Produto cadastrado – Precisa de mensagem



Cadastro de Produto Nacional

Codigo	Descricao
3	Leite
Valor	Taxa
2.5	0.24

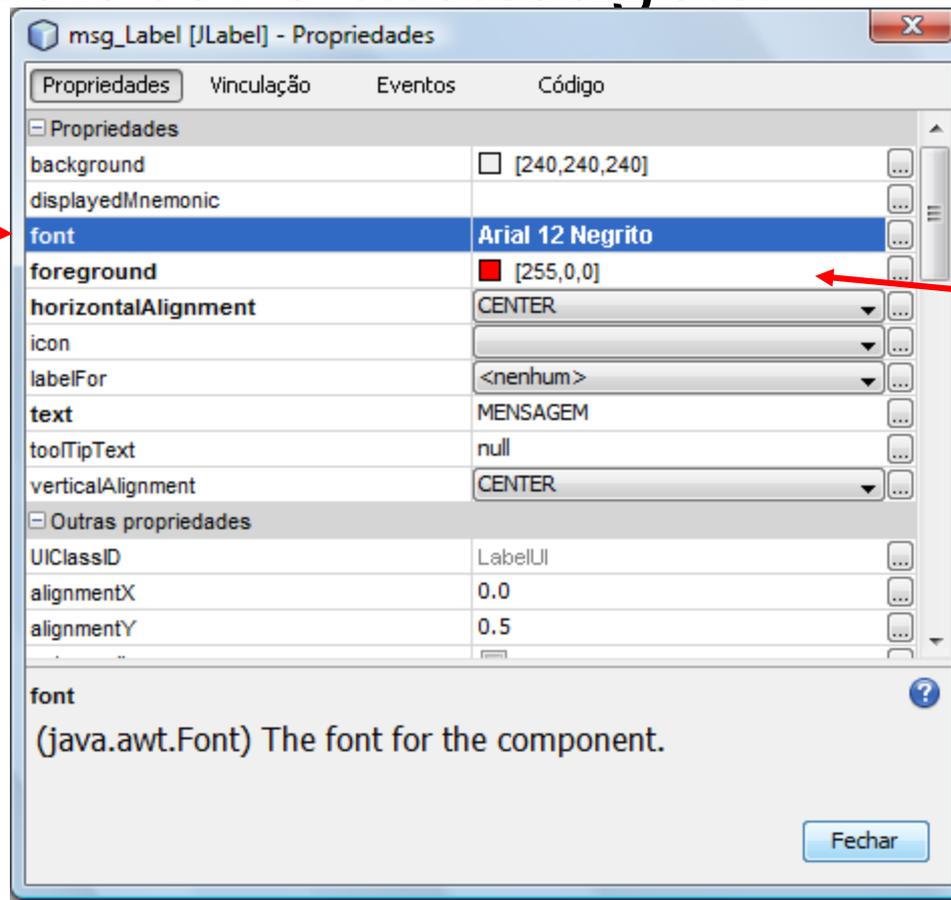
Adicionar Produto

Produto Já Cadastrado

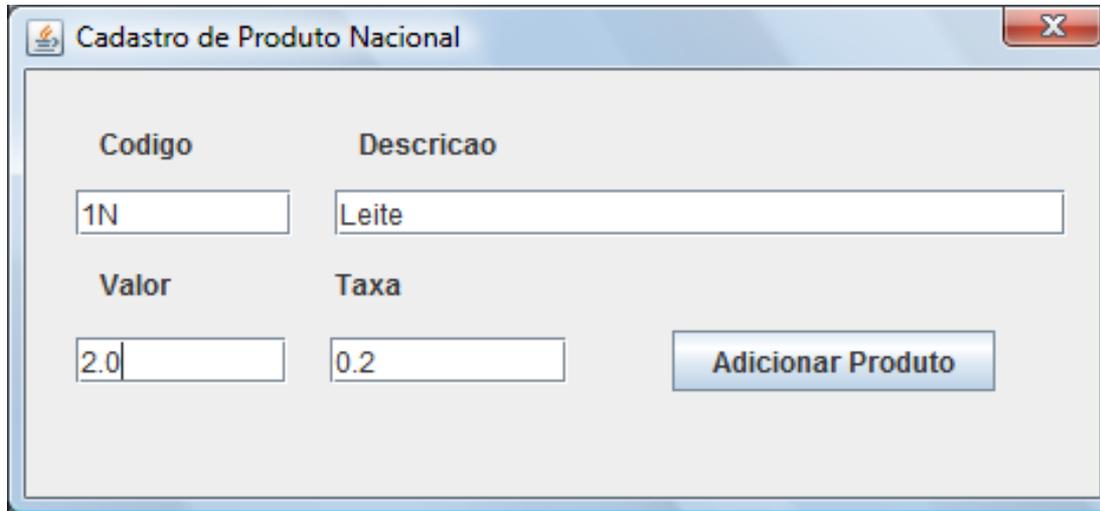
```
if (produto != null) {  
    descricao_Text.setText(produto.getDescricao());  
    valor_Text.setText( Float.toString( produto.getValor() ) );  
    float taxaImposto = ((ProdutoNacional) produto).getTaxaImposto();  
    taxa_Text.setText(Float.toString( taxaImposto ) );  
    adicionar_Button.setEnabled(false);  
    this.msg_Label.setText("Produto Já Cadastrado");  
    this.msg_Label.setVisible(true);  
}
```

Destacando a Mensagem

- Podemos alterar a cor e fonte do JLabel para destacar a mensagem



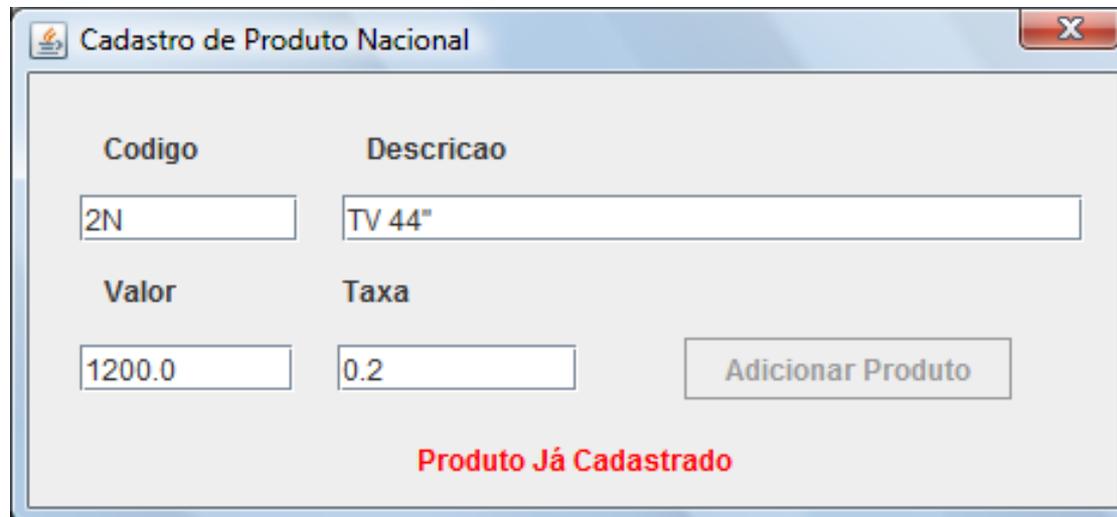
Destacando a Mensagem



Cadastro de Produto Nacional

Codigo	Descricao
1N	Leite
Valor	Taxa
2.0	0.2

Adicionar Produto



Cadastro de Produto Nacional

Codigo	Descricao
2N	TV 44"
Valor	Taxa
1200.0	0.2

Adicionar Produto

Produto Já Cadastrado

Referências

BIBLIOGRAFIA BÁSICA

1. SINTES, A., Aprenda programação orientada a objetos em 21 dias, Pearson Education do Brasil, 2002.
2. VAREJÃO, F., Linguagens de programação : Java, C e C++ e outras : conceitos e técnicas, Campus, 2004.
3. DEITEL, H. M., DEITEL, P. J., **Java:** como programar, São Paulo: Pearson Education do Brasil, 2010. 1144p.
4. DEITEL, H. M., DEITEL, P. J., **Java:** como programar, Porto Alegre: Bookman, 2003. 1386p.
5. SAVITCH, W. J., C++ absoluto, Pearson Education : Addison Wesley, 2004.

BIBLIOGRAFIA COMPLEMENTAR

1. BERMAN, A. M. *Data Structures via C++: Objects by Evolution*, Oxford University Press Inc., 1997.
2. BARNES, D.J. & KÖLLING, M., Programação orientada a objetos com Java, Pearson Education : Prentice Hall, 2004.
3. DEITEL, H. M. e DEITEL, P. J. *C++: Como Programar*, Bookman, 2001.
4. GILBERT, R. F. e FOROUZAN, B. A. *Data Structures: A Pseudo Approach with C++*, Brooks/Cole Thomson Learning, 2001.
5. MUSSER, D. R. e SAINI, A. *STL Tutorial and Reference Guide: Programming with the Standard Template Library*, Addison-Wesley, 1996.
6. SEBESTA, R. W. *Conceitos de Linguagem de Programação*, 4ª Ed., Bookman, 2003.
7. SEDGEWICK, R. *Algorithms in C++*, Addison-Wesley, 2002.
8. STROUSTRUP, B. *A Linguagem de Programação C++*, 3ª Ed., Bookman, 2000.