

**FCT/Unesp – Presidente Prudente**  
**Departamento de Matemática e Computação**

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

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

# Estratégia para evitar cadastro de produtos com mesmo código

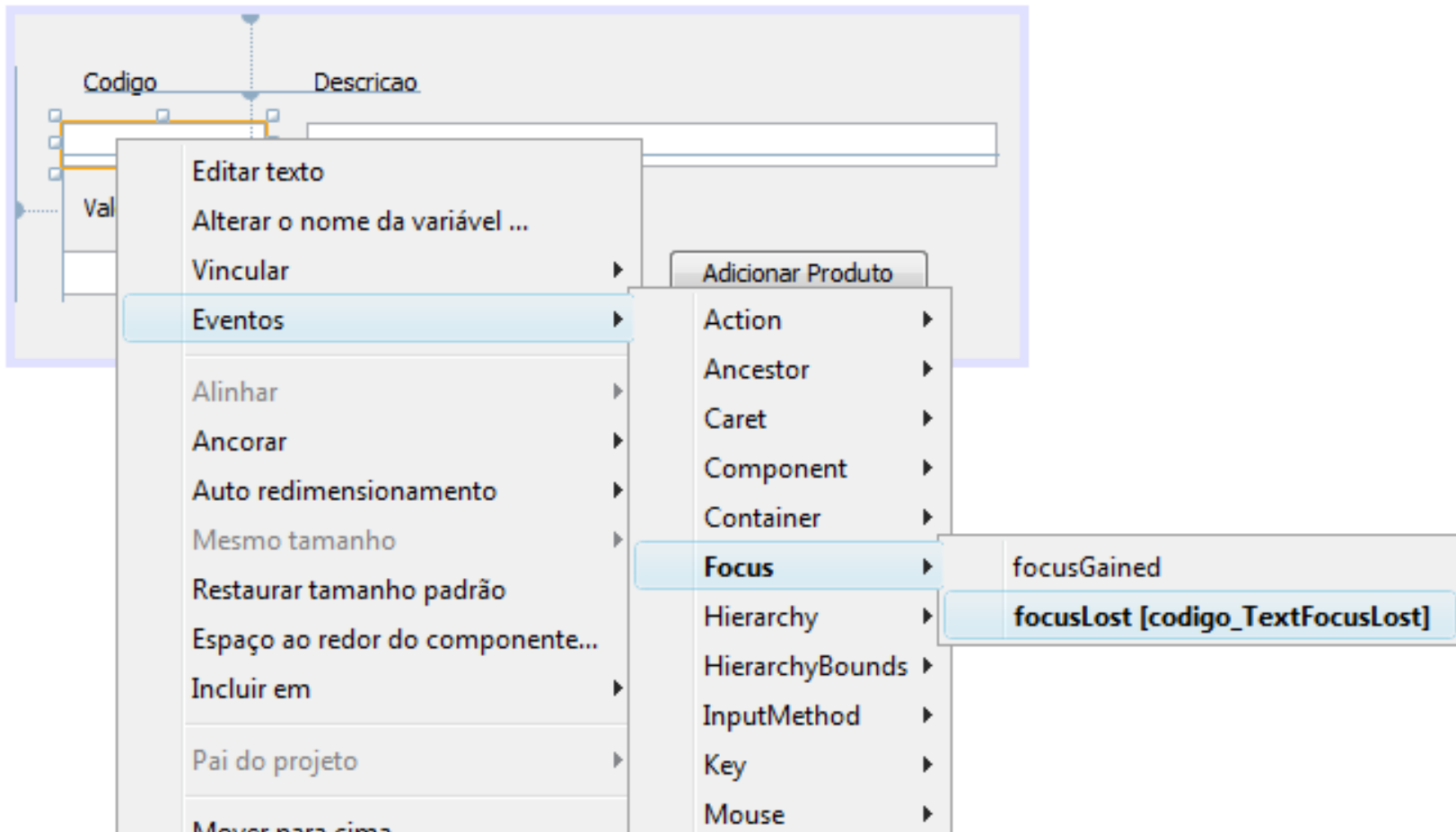
# Estratégia para evitar cadastro de produtos com mesmo código

- 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

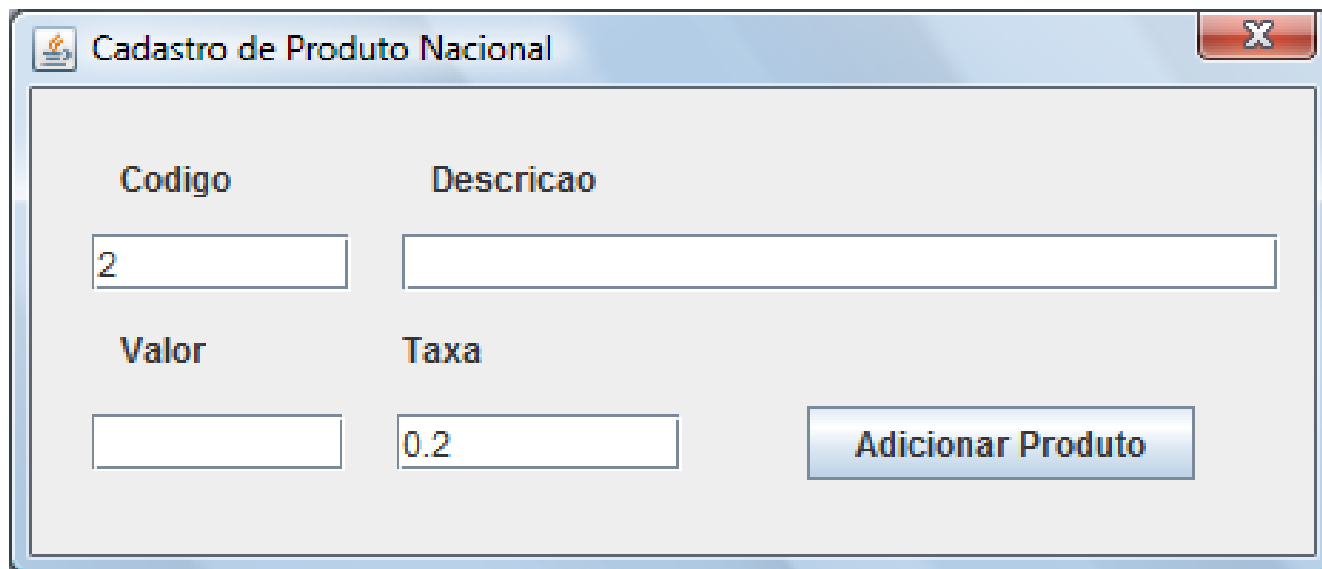


# Estratégia para evitar cadastro de produtos com mesmo código

```
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 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

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);  
}
```

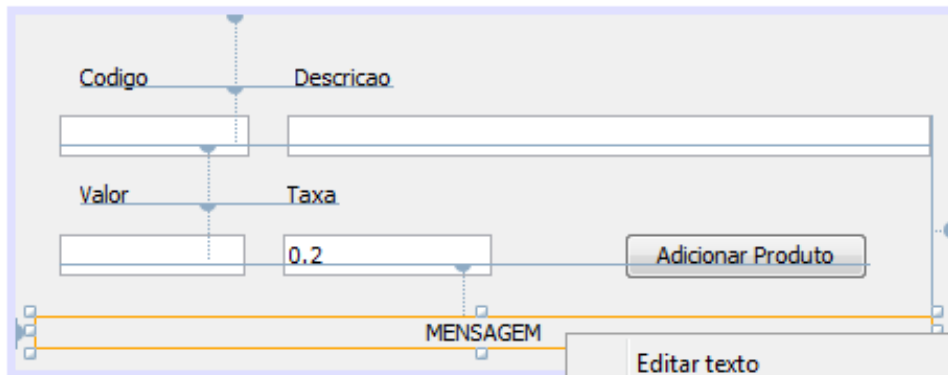
# Estratégia para evitar cadastro de produtos com mesmo código

- 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

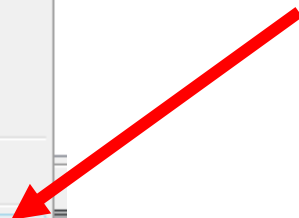
The screenshot shows a software interface with a form containing four input fields and one button. The fields are labeled 'Codigo', 'Descricao', 'Valor', and 'Taxa'. The 'Taxa' field contains the value '0.2'. A button labeled 'Adicionar Produto' is located to the right of the 'Taxa' field. Below the form, a message box labeled 'MENSAGEM' is visible, highlighted with an orange border. Dotted lines indicate connections between the labels and their respective input fields.

The screenshot shows a dialog box titled 'Renomear' with a close button (X) in the top right corner. The dialog contains a text input field labeled 'Novo nome:' with the text 'msg\_Label' entered. Below the input field are two buttons: 'OK' and 'Cancelar'.

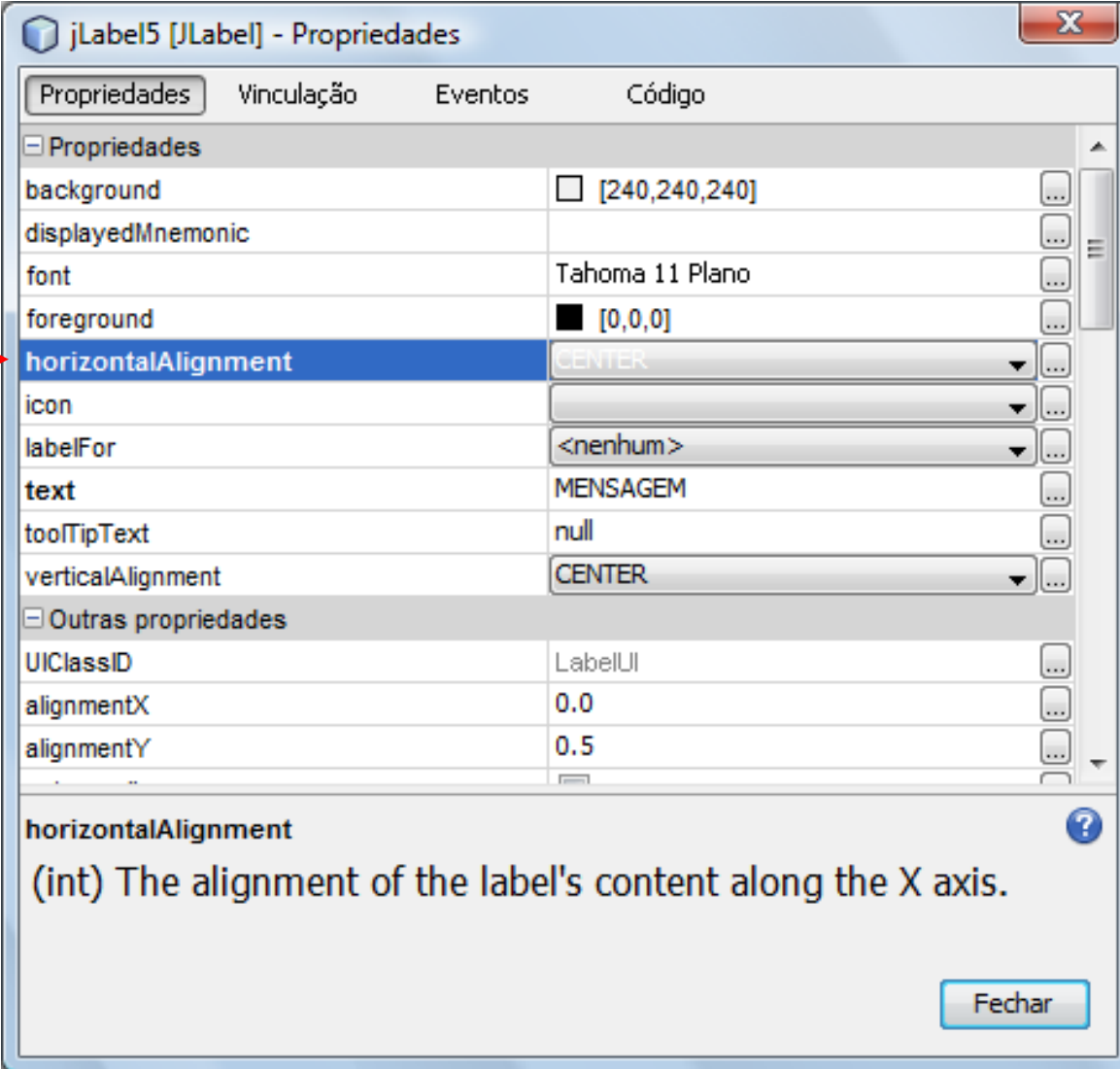


- Editar texto
- Alterar o nome da variável ...
- Vincular ▶
- Eventos ▶
- Alinhar ▶
- Ancorar ▶
- Auto redimensionamento ▶
- Mesmo tamanho ▶
- Restaurar tamanho padrão
- Espaço ao redor do componente...
- Incluir em ▶
- Pai do projeto ▶
- Mover para cima
- Mover para baixo
- Recortar
- Copiar
- Duplicar
- 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 a dialog box titled "jLabel5 [JLabel] - Propriedades" with four tabs: "Propriedades", "Vinculação", "Eventos", and "Código". The "Propriedades" tab is active, displaying a list of properties for the JLabel component. The "horizontalAlignment" property is selected and highlighted in blue, with a red arrow pointing to it. The tooltip for this property is visible at the bottom of the dialog, stating: "horizontalAlignment (int) The alignment of the label's content along the X axis." The dialog also includes a "Fechar" button at the bottom right.

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

# 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 {

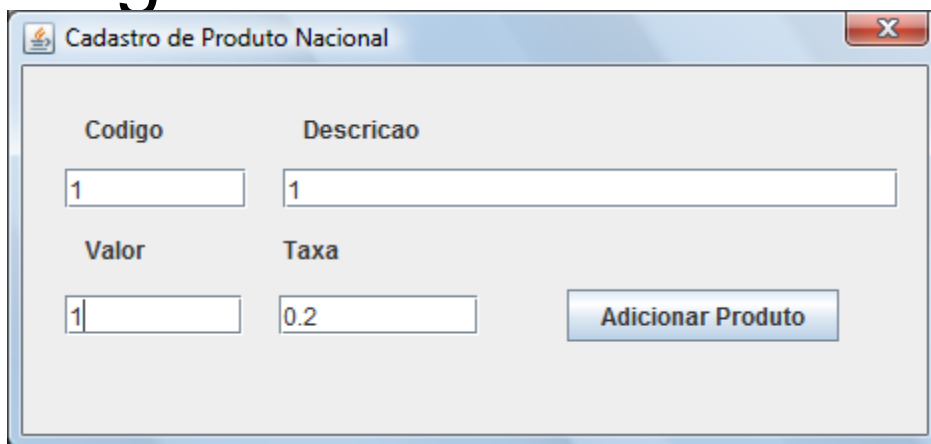
    /** Creates new form IUCadastroProdutoNacional */
    public IUCadastroProdutoNacional(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
        this.msg_Label.setVisible(false);
    }
}
```

# 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



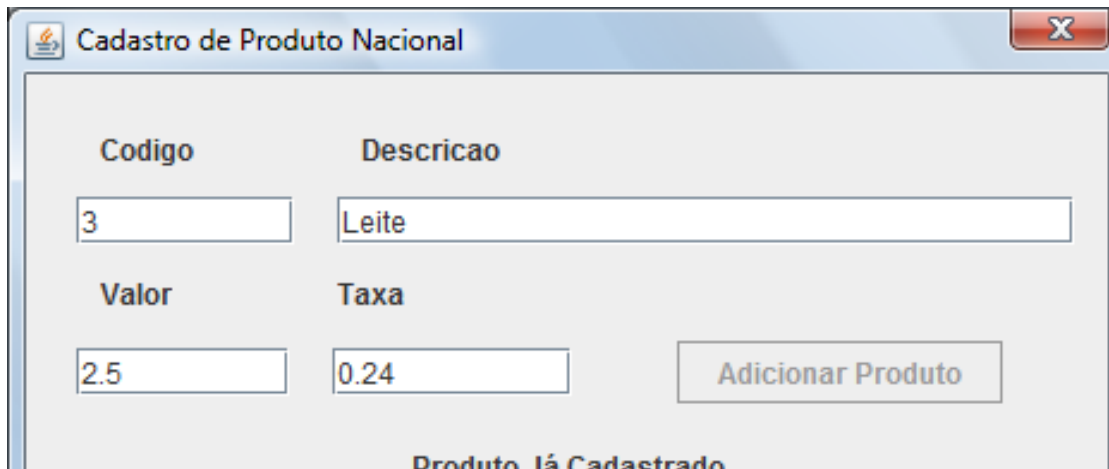
The screenshot shows a window titled "Cadastro de Produto Nacional". It has four text input fields arranged in a 2x2 grid. The top-left field is labeled "Codigo" and contains the number "1". The top-right field is labeled "Descricao" and contains the number "1". The bottom-left field is labeled "Valor"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

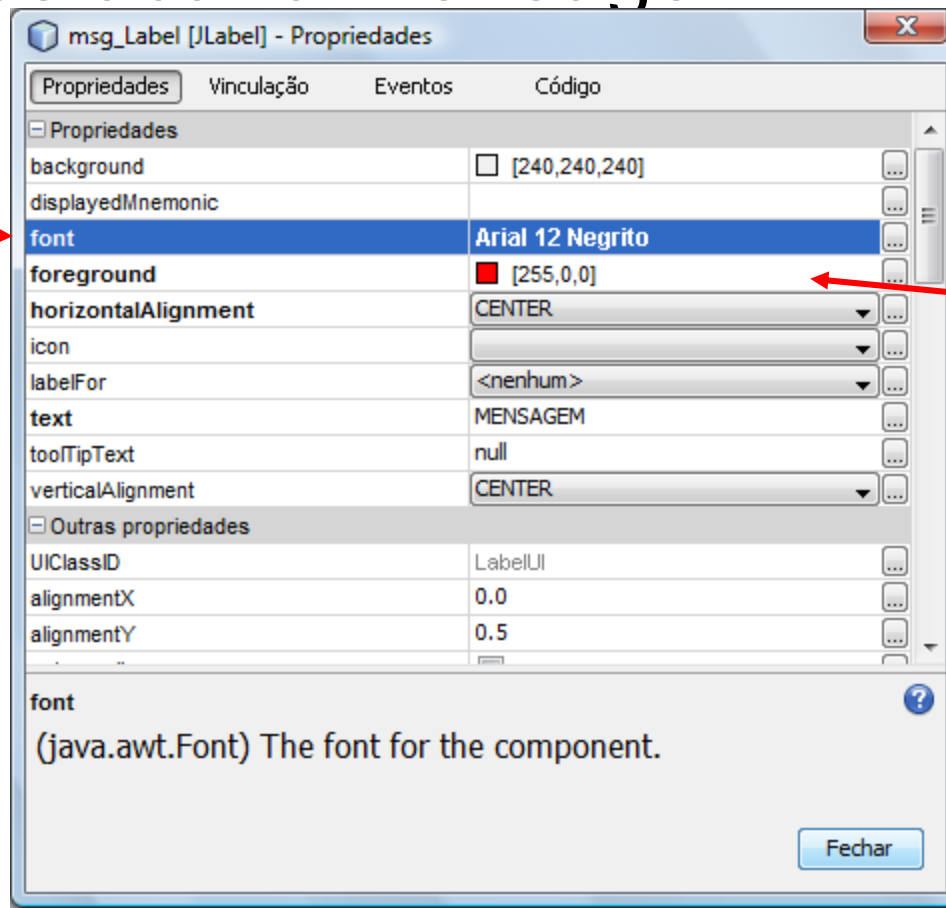
Adicionar Produto

Produto Já Cadastrado

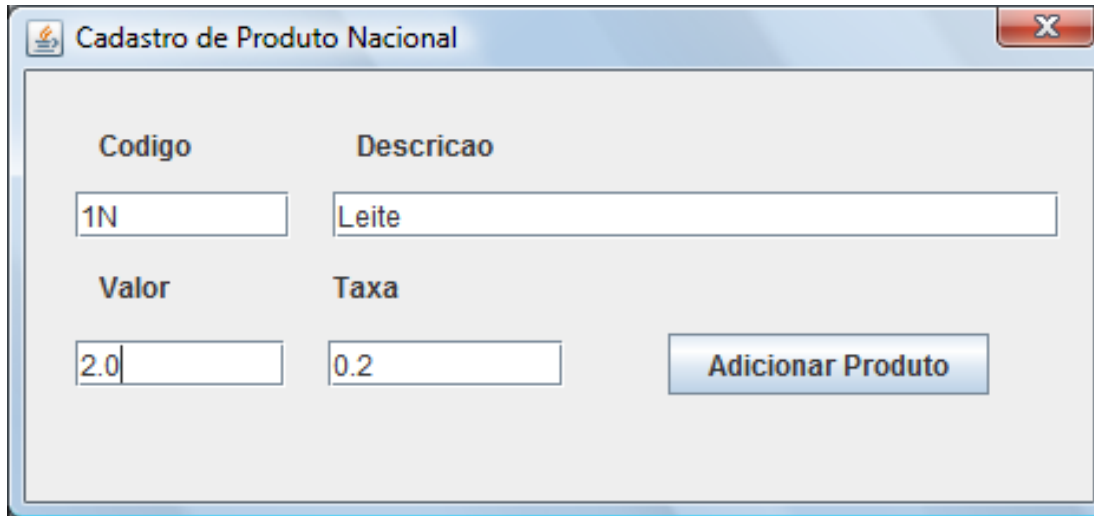
```
if (produto != null) {  
    descricao_Text.setText(produto.getDescricao());  
    valor_Text.setText( Float.toString( produto.getValor() ) );  
    float taxaImposto = ((ProdutoNacional) produto).getTaxaImposto();  
    taxa_Text.setText(Float.toString( taxaImposto ) );  
    adicionar_Button.setEnabled(false);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



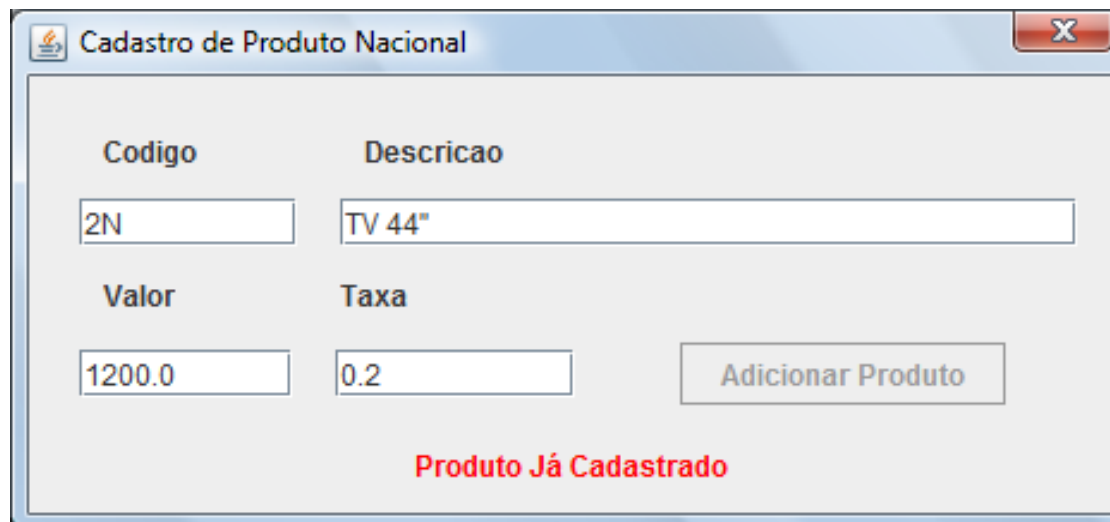
# Destacando a Mensagem



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.