

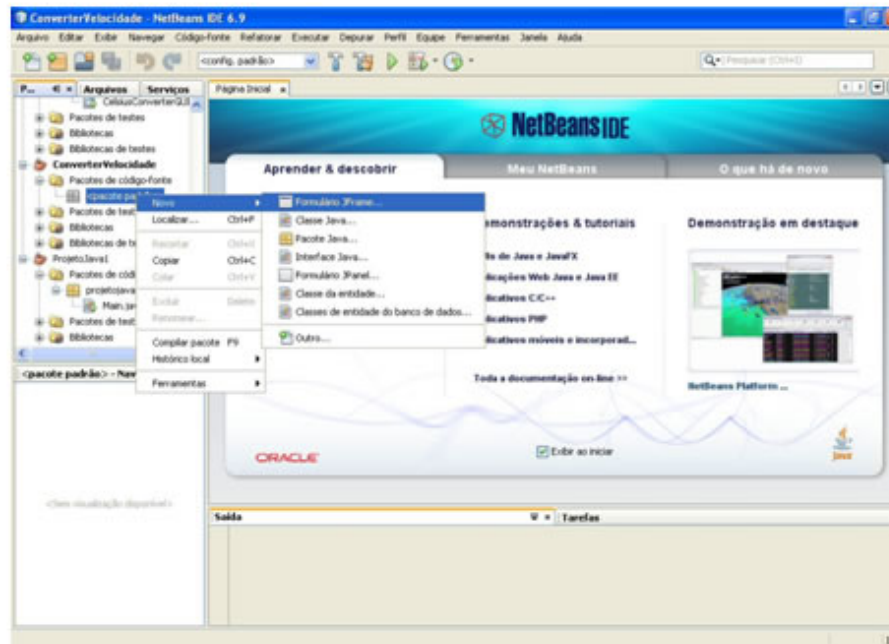
Interface Gráfica - Swing

Swing com o NetBeans

- Para a **construção** de **aplicativos gráficos**, o construtor de **interfaces gráficas** do **NetBeans** IDE, torna a **atividade uma simples tarefa** de **arrastar** e **soltar**;
- Sua **geração automática** de **código** simplifica bastante o **processo** de **desenvolvimento** das **GUIs** (**Graphical User Interface** ou Interface Gráfica com o Usuário), permitindo o **foco** do programador na **lógica da aplicação**, e não na **infraestrutura** necessária ao aplicativo.

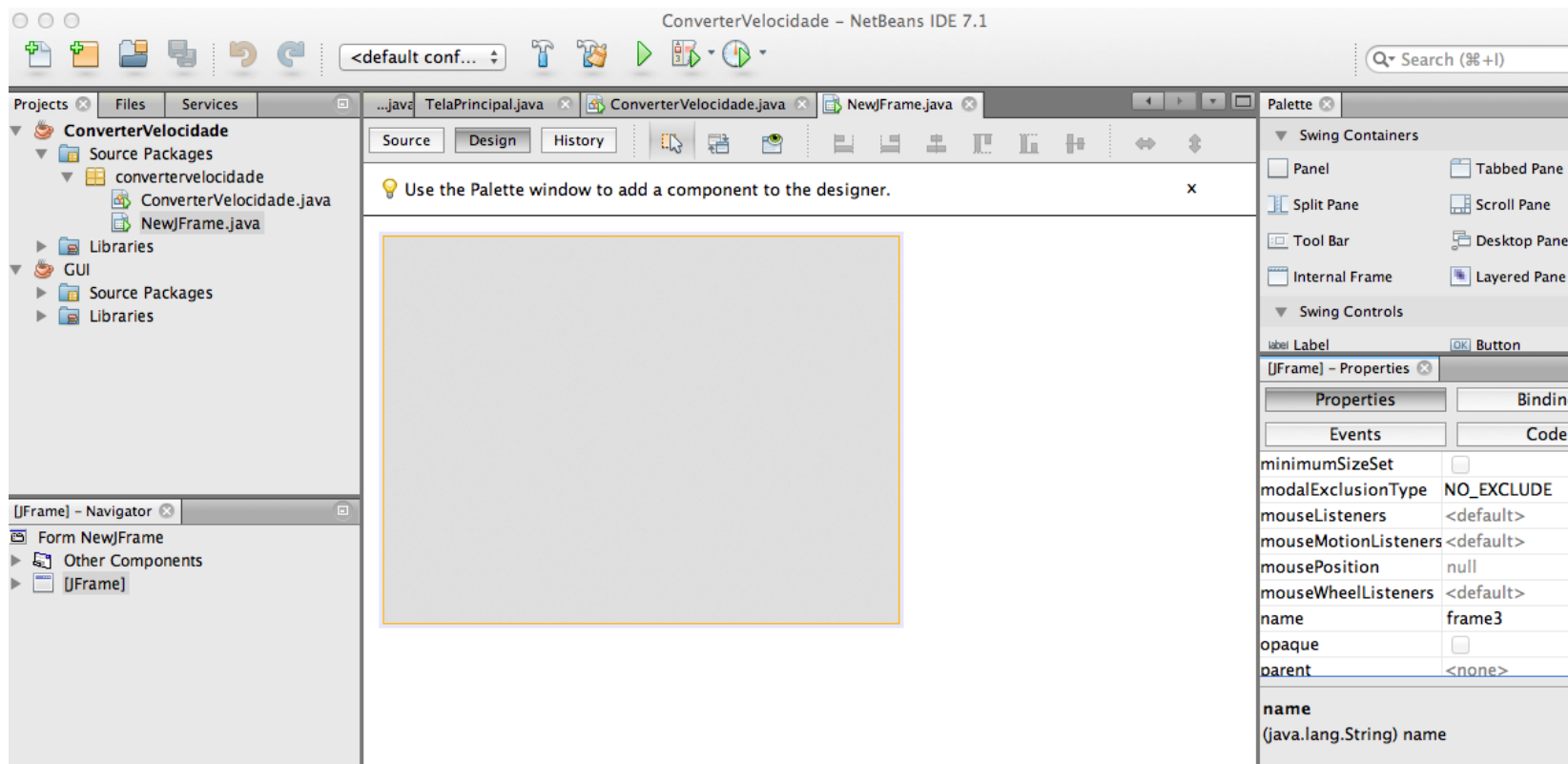
Aplicação Simples

- ❑ Criem um **novo projeto** com nome **ConverterVelocidade**;
- ❑ Certifique-se de que a opção (*check box*) **Criar classe principal** esteja **desmarcada**.
- ❑ Adicione um **novo** formulário **JFrame** clicando com o **botão** direito do **mouse** sobre o nome do **projeto** (**ConverterVelocidade**) e seguindo o caminho,



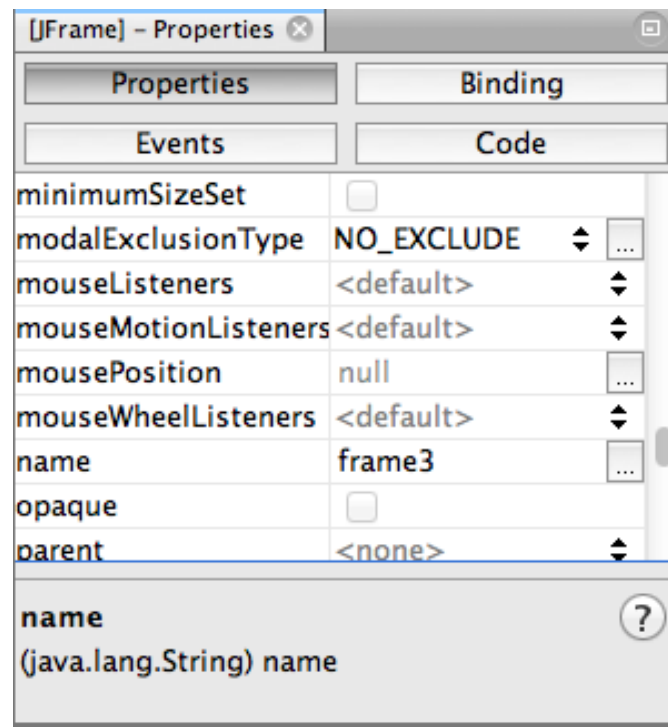
Opções de Visualização

- O **NetBeans** disponibiliza **duas opções** de **visualização**: a vista **gráfica** do projeto e a vista do **código** fonte;



Propriedades

- No modo de **visualização Projeto** clicando no **componente JFrame** aparecerá, no canto **inferior direito**, o painel de **propriedades** com várias **características** do componente que **poderão** ser **alteradas**



Propriedades do JFrame

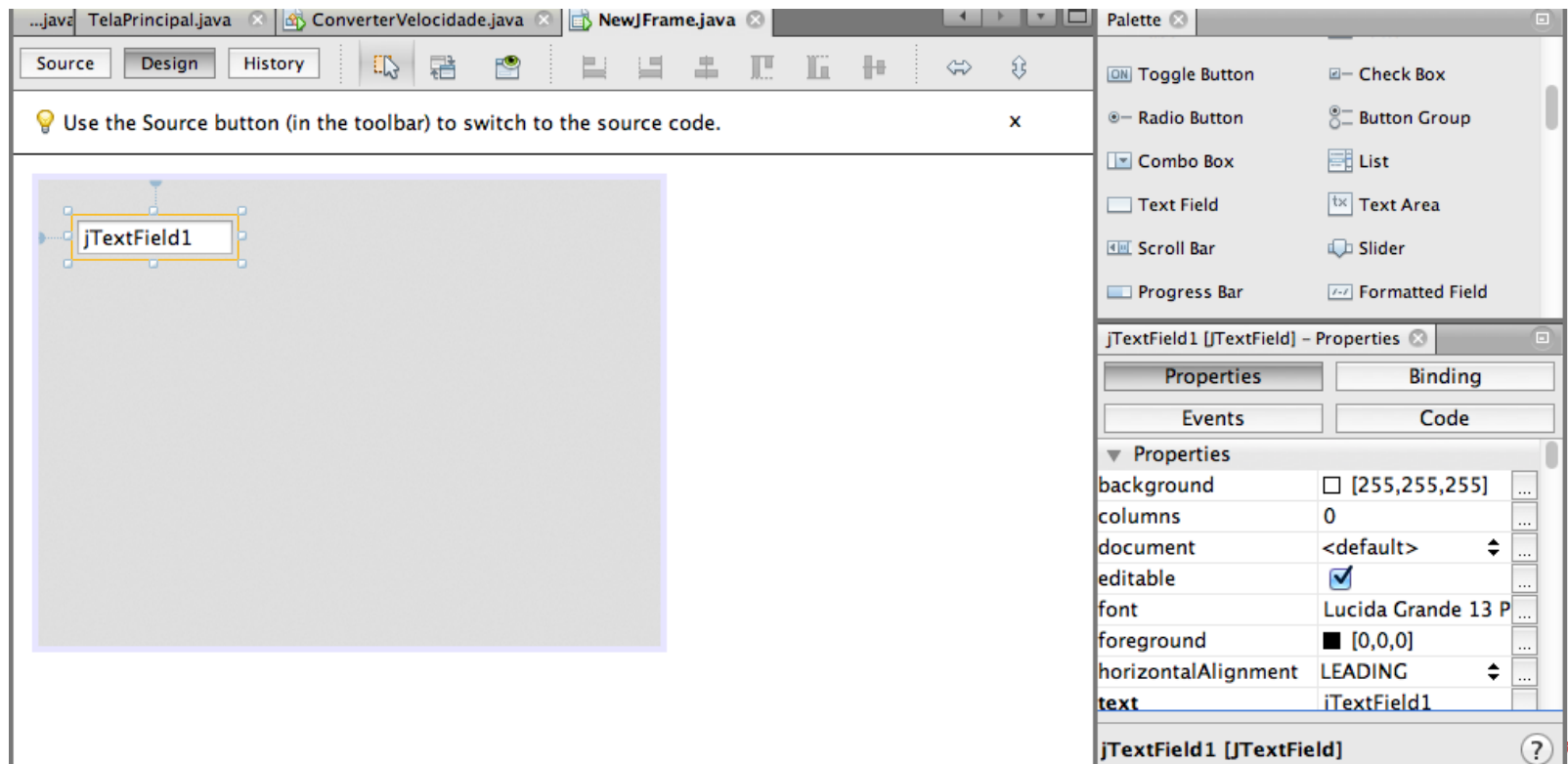
- ❑ No subitem **Propriedades** do menu, você poderá utilizar:
 - **title** – permite **inserir** um **texto** que aparecerá no **topo** da **janela** do **aplicativo**.

- ❑ Na seção '**Outras propriedades**', você poderá **alterar** também:
 - **reizable** – permite o usuário **redimensionar** a **janela** do **aplicativo**.

 - **undecorated** – define a **exibição** ou **não** da **borda** da **janela**. Por **padrão**, essa opção se apresenta **desmarcada**, exibindo a borda externa ao **JFrame**.

Campo de texto (TextField)

- Em seguida, na janela à **direita**, clique e **arraste** um campo de **texto** (**JTextField**) para **dentro** da área do **JFrame**.

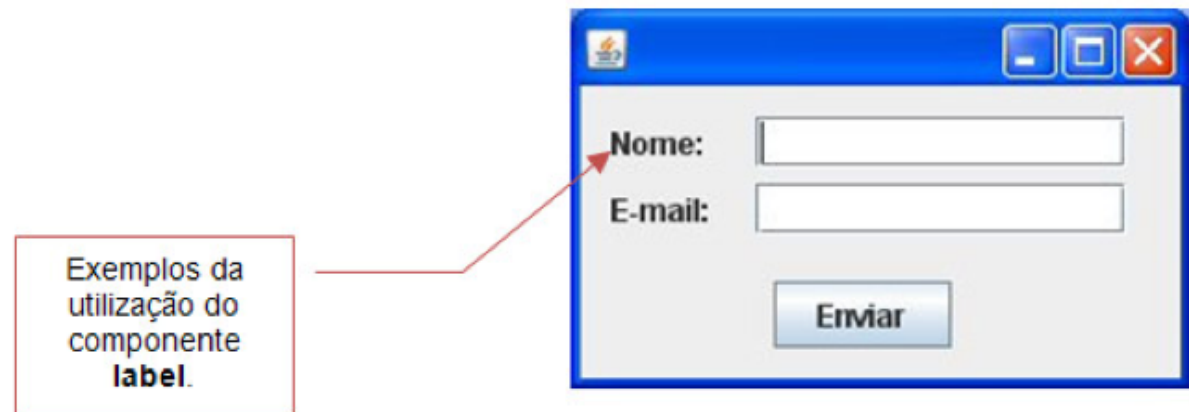


Campo de texto (TextField)

- Esse **componente** é muito **comum** em **formulários** onde o usuário **digita** um **texto** como uma **informação** a ser **enviada**: dados de nome, telefone e e-mail em um cadastro pessoal.
- Algumas de suas **propriedades** incluem
 - **text** – essa **propriedade** é utilizada para você **especificar** um **texto padrão** para o **componente** quando a aplicação for **executada**, ao invés de ficar em branco.
 - **background** – permite **alteração** da **cor de plano de fundo** do componente;

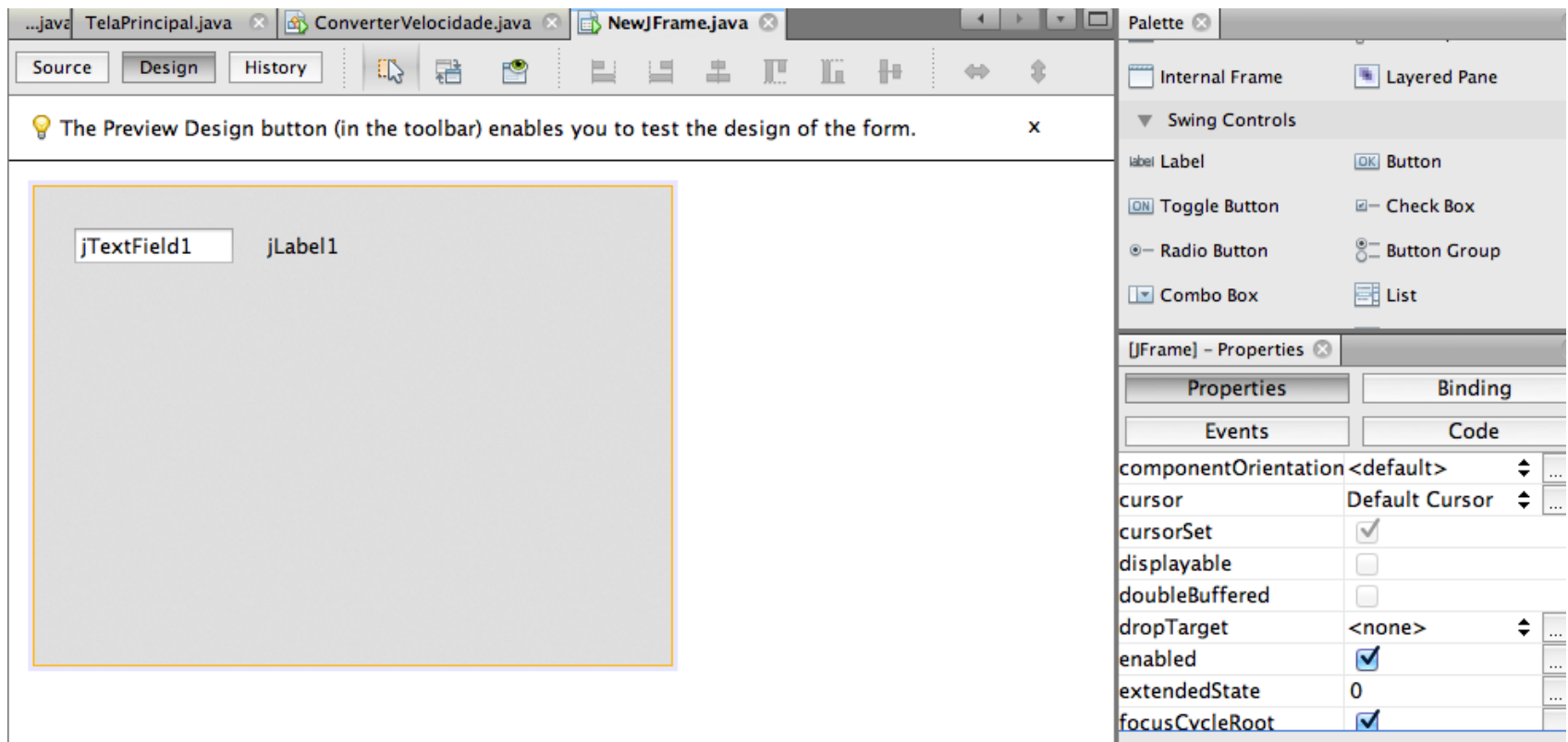
Rótulo (label)

- ❑ Esse é um outro **componente** bastante **comum**. Podemos dizer até que está **presente** em toda **interface gráfica**.
- ❑ O **Rótulo**, mais conhecido pelo termo em inglês **label**, é todo o texto que **encontramos** na **interface** que tem a **função** de informar o usuário.



Rótulo (label)

- Agora **adicione** o rótulo (**JLabel**) seguindo as linhas-guia, posicionando-o ao lado do **TextField**;



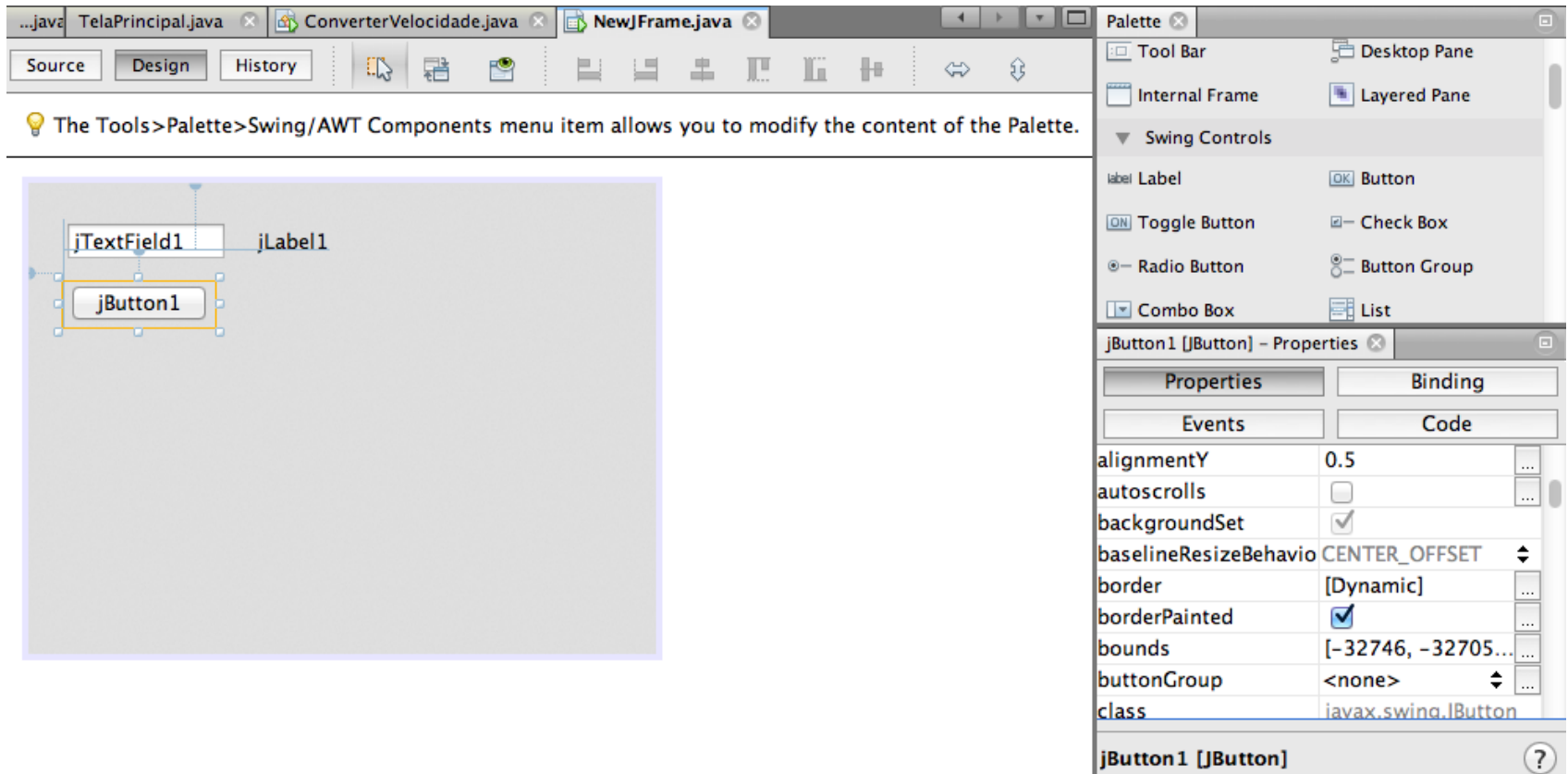
The screenshot shows an IDE window titled 'NewJFrame.java' in Design mode. The design view displays a window with a text field labeled 'jTextField1' and a label 'jLabel1' positioned to its right. The Palette on the right lists Swing Controls, including 'Label', 'Button', 'Toggle Button', 'Radio Button', 'Combo Box', 'Check Box', 'Button Group', and 'List'. The Properties window for the selected component shows various attributes such as 'componentOrientation', 'cursor', 'cursorSet', 'displayable', 'doubleBuffered', 'dropTarget', 'enabled', 'extendedState', and 'focusCycleRoot'.

Rótulo (label)

- Vejamos mais algumas **propriedades** que podemos **manipular** para **personalizar** o **componente**.
 - **font** – com essa **propriedade**, é possível **configurar diversas** outras **propriedades** em relação à fonte, como **tipo**, **estilo** e **tamanho**.
 - **foreground** – utilizado para **alterar** a **cor** da **fonte**.

Botão (Jbutton)

- Agora, **adicione** um **botão (JButton)** logo abaixo do **JTextField**. Lembre-se de **utilizar** as linhas guia para **alinhar** os **componentes**.



The Tools>Palette>Swing/AWT Components menu item allows you to modify the content of the Palette.

jButton1 [JButton] - Properties	
Properties	Binding
Events	Code
alignmentY	0.5
autoscrolls	<input type="checkbox"/>
backgroundSet	<input checked="" type="checkbox"/>
baselineResizeBehavio	CENTER_OFFSET
border	[Dynamic]
borderPainted	<input checked="" type="checkbox"/>
bounds	[-32746, -32705...]
buttonGroup	<none>
class	javax.swing.JButton

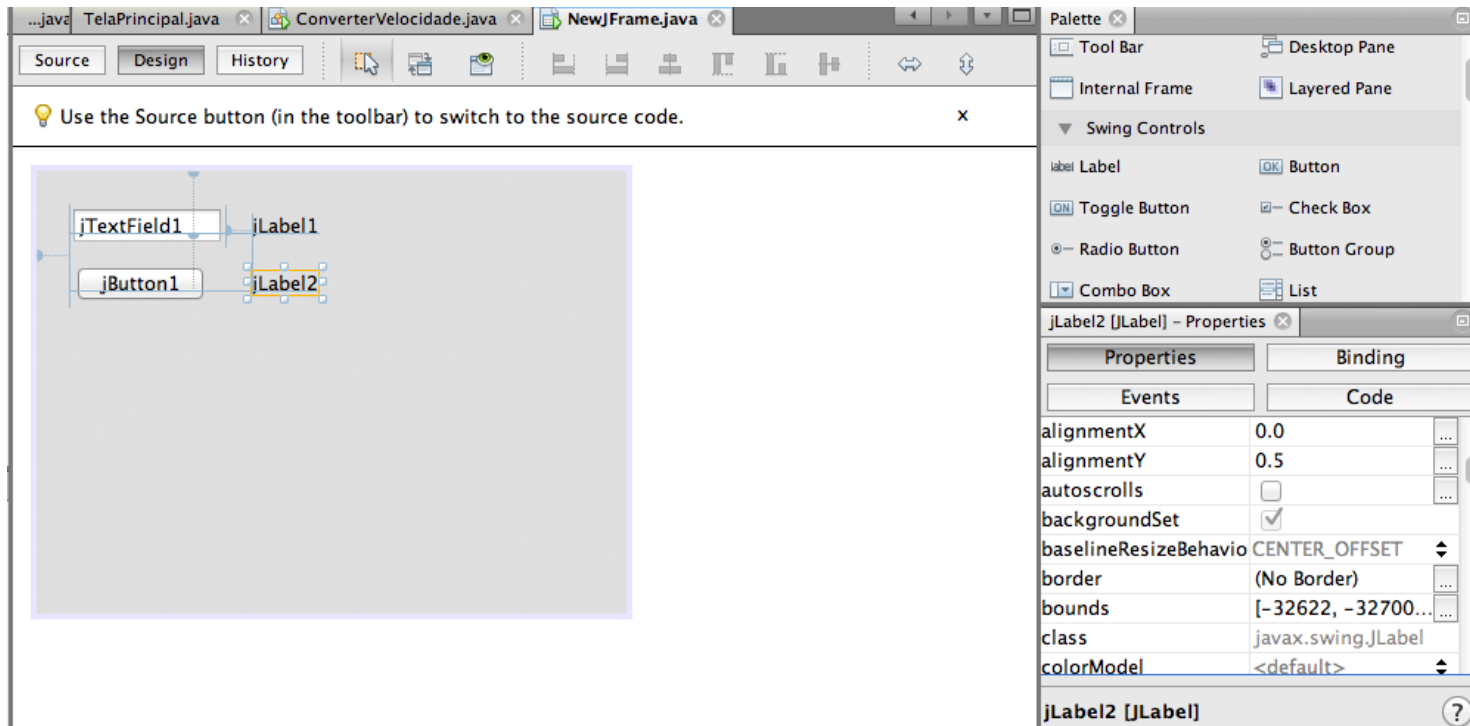
jButton1 [JButton]

Botão (Jbutton)

- Na lista de propriedades **disponíveis**, você poderá **personalizar** o seu **botão** e deixá-lo do seu jeito. Algumas **propriedades** deste componente **incluem**:
 - **toolTipTex** – essa propriedade permite **incluir** um **texto explicativo** que aparece para o **usuário** quando o **cursor** passa por cima do **componente**;
 - **enabled** – se esse **check box** estiver **marcado** significa que o componente ficará **habilitado**, ou seja, no caso do **botão** poder ser **clicado**, caso contrário, ficará **desabilitado** e não será **executada nenhuma ação** definida para esse componente. Por padrão, esse **check box já é marcado**.

Fazendo Aplicação

- ▣ Agora **adicione** um segundo **JLabel** ao lado do **botão**;



Use the Source button (in the toolbar) to switch to the source code.

Palette

- Tool Bar
- Desktop Pane
- Internal Frame
- Layered Pane
- Swing Controls
 - Label
 - Button
 - Toggle Button
 - Check Box
 - Radio Button
 - Button Group
 - Combo Box
 - List

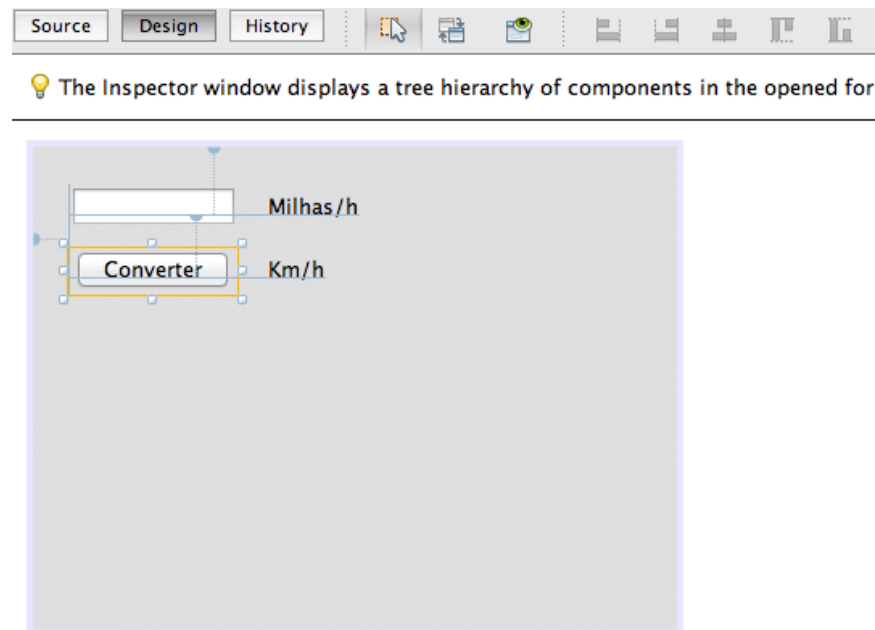
JLabel2 [JLabel] - Properties

Properties	Binding
Events	Code
alignmentX	0.0
alignmentY	0.5
autoscrolls	<input type="checkbox"/>
backgroundSet	<input checked="" type="checkbox"/>
baselineResizeBehavior	CENTER_OFFSET
border	(No Border)
bounds	[-32622, -32700...
class	javax.swing.JLabel
colorModel	<default>

jLabel2 [JLabel]

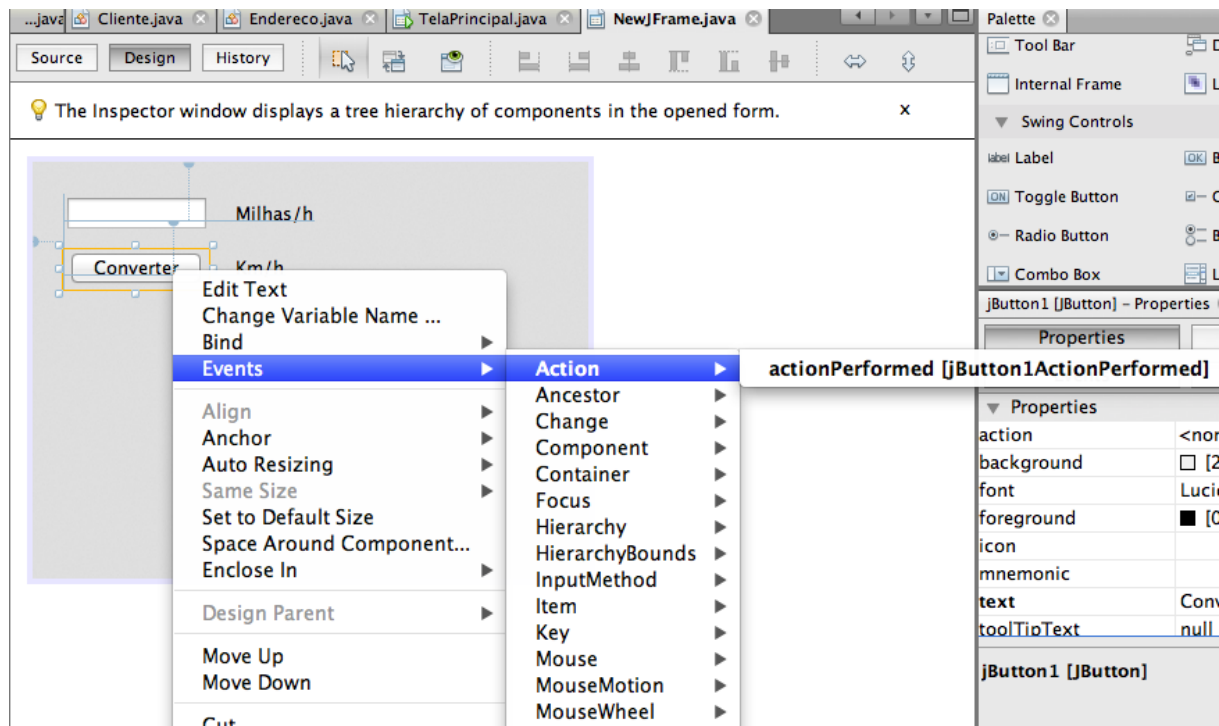
Fazendo Aplicação

- ❑ Para alterar o texto do **JTextField**, clique no componente e procure no painel de propriedades o campo **text**. Ao apagar o texto padrão **JTextField1**, que constava anteriormente,
- ❑ Como não queremos nenhum texto inicial no **JTextField**, manteremos esse campo vazio. Agora, de forma similar, altere o campo **text** dos outros componentes adicionando os nomes (**JLabel1** "milhas/h", **JButton** "Converter" e **JLabel2** "Km/h").



Fazendo Aplicação

- ❑ A parte realizada na GUI está completa. Vamos, agora, concluir o aplicativo inserindo os comandos de conversão ao botão.
- ❑ Clique no botão **Converter** com o botão direito do mouse.



Fazendo Aplicação

- **Aparecerá** na tela a **área** onde deverá ser **incluído** o **código** que será **executado** quando o **botão** for **apertado**.
- Insira a **linha** de comando **especificada** abaixo
- **int tempFahr = (int)**
((Double.parseDouble(jTextField1.getText(
))) * 1.609344);
jLabel2.setText(tempFahr + " Km/h");