

Índice

Figura 1 - Abrindo o NetBeans	8
Figura 2 - abrindo o NetBeans.....	9
Figura 3 - Tela do NetBeans	9
Figura 4 - Criando o Projeto	10
Figura 5 - nomeando o projeto	11
Figura 6 - renomeando o projeto.....	11
Figura 7 - criação de pacotes	12
Figura 8 - nomeando os pacotes	12
Figura 9 - nomeando os pacotes	13
Figura 10 - criando o JFrame	13
Figura 11 - nomeando o JFrame.....	14
Figura 12 - Tela do JFrame.....	14
Figura 13 - inserindo o botão JLabel	15
Figura 14 - botão JLabel inserido.....	15
Figura 15 - redimensionando o botão JLabel	16
Figura 16 - renomeando o JLabel.....	14
Figura 17 - renomeando o JLabel.....	17
Figura 18 - renomeando o JLabel.....	14
Figura 19 - renomeando o JLabel.....	18
Figura 20 - inserindo Imagem e centralizando	15
Figura 21 - inserindo Imagem e centralizando	19
Figura 22 - inserindo Imagem e centralizando	15
Figura 23 - inserindo Imagem e centralizando	20
Figura 24 - inserindo Imagem e centralizando	15
Figura 25 - inserindo Imagem e centralizando	21
Figura 26 - inserindo Imagem e centralizando	16
Figura 27 - inserindo Imagem e centralizando	22
Figura 28 - inserindo Imagem e centralizando	22
Figura 29 - inserindo e renomeando o botão JToggleButton	16
Figura 30 - inserindo e renomeando o botão JToggleButton	23
Figura 31 - inserindo e renomeando o botão JToggleButton	23
Figura 32 - inserindo e renomeando o botão JToggleButton	24
Figura 33 - inserindo o botão ButtonGroup.....	18
Figura 34 - inserindo o botão ButtonGroup.....	24

Figura 35- testando o programa	25
Figura 36 - inserindo um evento actionPerformed	25
Figura 37 - declarando as variáveis dos botões Ligar e Desligar	26
Figura 38 - declarando as variáveis dos botões Ligar e Desligar	27
Figura 39 - testando o programa	27
Figura 40 - testando o botão Ligar	28
Figura 41 - testando o botão Desligar	28
Figura 1criando uma classe Java	30
Figura 2 criando uma classe Java.....	31
Figura 3 usando o JFrame	31
Figura 4Jtexarea.....	32
Figura 5 Inserindo o jtexarea	32
Figura 6JtextArea	33
Figura 7 Inserindo o Label no projeto	33
Figura 8 editando o texto Label	34
Figura 9 um label após o nome editado	35
Figura 10 inserindo o Jtextfield	35
Figura 11 editando o Jtextfield	35
Figura 12 Inserindo ojbutton 1	36
Figura 13 Inserindo o Jbutton2.....	37
Figura 14 colocando o nome Inserir	37
Figura 15 colocando o nome Cancelar no J button2	38
Figura 16 O Jbutton1 e 2 já editados.....	38
Figura 17 visualizando o projeto	39
Figura 18 design do projeto	39
Figura 19 Inserindo uma borda no Jtexarea	40
Figura 20 colocando título e alterando a fonte	41
Figura 21dando funcionalidade aos botões.....	41
Figura 22 ativando actionperformance.....	42
Figura 23 declarando as variáveis.....	42
Figura 24 declarando variáveis	43
Figura 25 declarando variáveis	44
Figura 26 design do projeto	44
Figura 27 testando o botão.....	45
Figura 28 botão inserir funcionando corretamente	45
Figura 29 mostra as palavras digitadas inseridas no nosso projeto	46

Figura 1 - Criando projeto	48
Figura 2 - Escolha de Projeto.....	49
Figura 3 - Atribuir nome ao projeto.....	50
Figura 4 - Criação de Janelas	50
Figura 5 - Definindo o nome do JFrame.....	51
Figura 6 - Acessando as propriedades do JFrame	52
Figura 7 - Opção defaultCloseOperation DISPOSE.....	52
Figura 8 - Modificando o title	53
Figura 9 - Resizable.....	53
Figura 10 - Gerar Centralizado.....	53
Figura 11 - Visualizar Design.....	54
Figura 12 - Opções da paleta.....	54
Figura 13 - Usando o Pannel.....	55
Figura 14 - Segundo Pannel.....	56
Figura 15 - Bordas do Pannel.....	56
Figura 16 - Borda com Título	57
Figura 17 - Título da borda superior.....	57
Figura 18 - Cor da borda.....	58
Figura 19- Fonte da borda.....	58
Figura 20 - Opções fonte da borda	59
Figura 21 - Justificativa da borda	59
Figura 22 - Borda sobre borda.....	60
Figura 23 - Borda com Título - Borda	61
Figura 24 - Paleta AWT.....	62
Figura 25 - Propriedades do Pannel	63
Figura 26 - Background Pannel	64
Figura 27 - Background Pannel Inferior	64
Figura 28 - Botão de Rádio	65
Figura 29 - Opções jButton	65
Figura 30 - Opções de Envio.....	66
Figura 31 - PropriedadesjRadioButton.....	67
Figura 32 - FontjRadioButton.....	68
Figura 33 - Negrito jRadioButton.....	68
Figura 34 - Adicionando Botão	69
Figura 35 - Adicionando Label	69
Figura 36 - Textos JLabel	70

Figura 37 - Grupo de Botões.....	71
Figura 38 - Navegador do NetBeans	71
Figura 39 - Alterando Nome da Variável	72
Figura 40 - Propriedade buttonGroup.....	72
Figura 41 - actionPerformed.....	73
Figura 42 - Código PAC.....	74
Figura 43 - Código Encomenda Normal e Sedex.....	74
Figura 44 - Código Opção Seleccionada	75
Figura 45 - ActionCommand.....	76
Figura 46 - Código Botão Limpar Seleção.....	76
Figura 47 - Exemplo Radio Button.....	77
Figura 1 - Exemplo CheckBox	79
Figura 2 - Criando Projeto.....	80
Figura 3 - Escolha do Projeto.....	81
Figura 4 - Atribuir Nome ao Projeto	82
Figura 5 - Criação de Janelas	82
Figura 6 - Definindo o Nome do JFrame	83
Figura 7 - Acessando as Propriedades do JFrame	84
Figura 8 - Opção defaultCloseOperation DISPOSE.....	84
Figura 9 - Modificando o title	85
Figura 10 - Resizable.....	85
Figura 11 - Gerar Centralizado.....	85
Figura 12 - Visualizar Design.....	86
Figura 13 - Opções da Paleta	87
Figura 14 - Usando o Pannel.....	87
Figura 15 - Bordas do Pannel.....	88
Figura 16 - Borda com Título	88
Figura 17 - Título da Borda	89
Figura 18 - Cor da Borda	89
Figura 19 - Fonte da Borda	89
Figura 20 - Opções Fonte da Borda	90
Figura 21 - Justificativa da Borda.....	90
Figura 22 - Borda sobre Borda	91
Figura 23 - Borda com Título - Borda	91
Figura 24 - Propriedades do Pannel	92
Figura 25 - Background Pannel	93

Figura 26 - Adicionando Label	93
Figura 27 - Editar Texto	94
Figura 28 - Navegador.....	94
Figura 29 - Propriedades do JLabel.....	95
Figura 30 - Fonte do JLabel.....	95
Figura 31 - Adicionando Campo de Texto.....	96
Figura 32 - Editar TextojTextArea	96
Figura 33 - Adicionando jCheckBox	97
Figura 34 - Editar Texto jCheckBox	97
Figura 35 - Adicionando JButton	98
Figura 36 - Editar Texto JButton	98
Figura 37 - actionPerformed botão OK.....	99
Figura 38 - Código botão OK	100
Figura 39 - Projeto	100
Figura 40 - Código do botão Limpar	101
Figura 158 - Exemplo CheckBox.....	90

JToggleButton

Três tipos de botões de estado — `JToggleButton`, `JCheckBox` e `JRadioButton` — que têm valores ativados/desativados ou verdadeiro/falso. As classes `JCheckBox` e `JRadioButton` são subclasses de `JToggleButton`.

É uma variação bem básica do `JButton`. Trata-se de um botão que, uma vez pressionado, fica “afundado”, demonstrando que tal opção está selecionada ou que tal funcionalidade está ativada. O uso do `JToggleButton` torna-se possível por causa do método `isSelected()`, que retorna `True` ou `False` e nos diz se o botão está “afundado” ou não. Um Botão com duas opções.



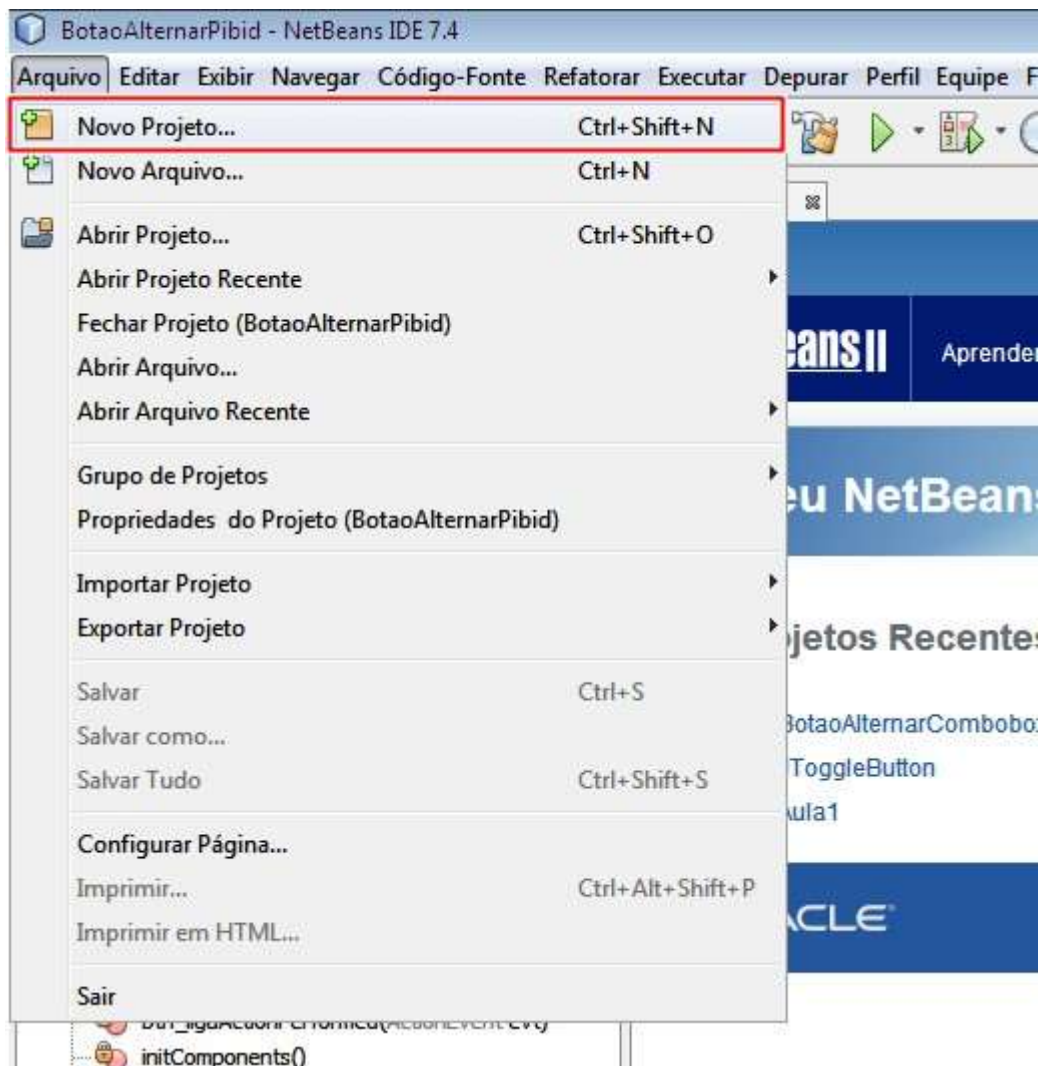


Figura 4 - Criando o Projeto.

Iremos nomear o projeto conforme a Figura 5:

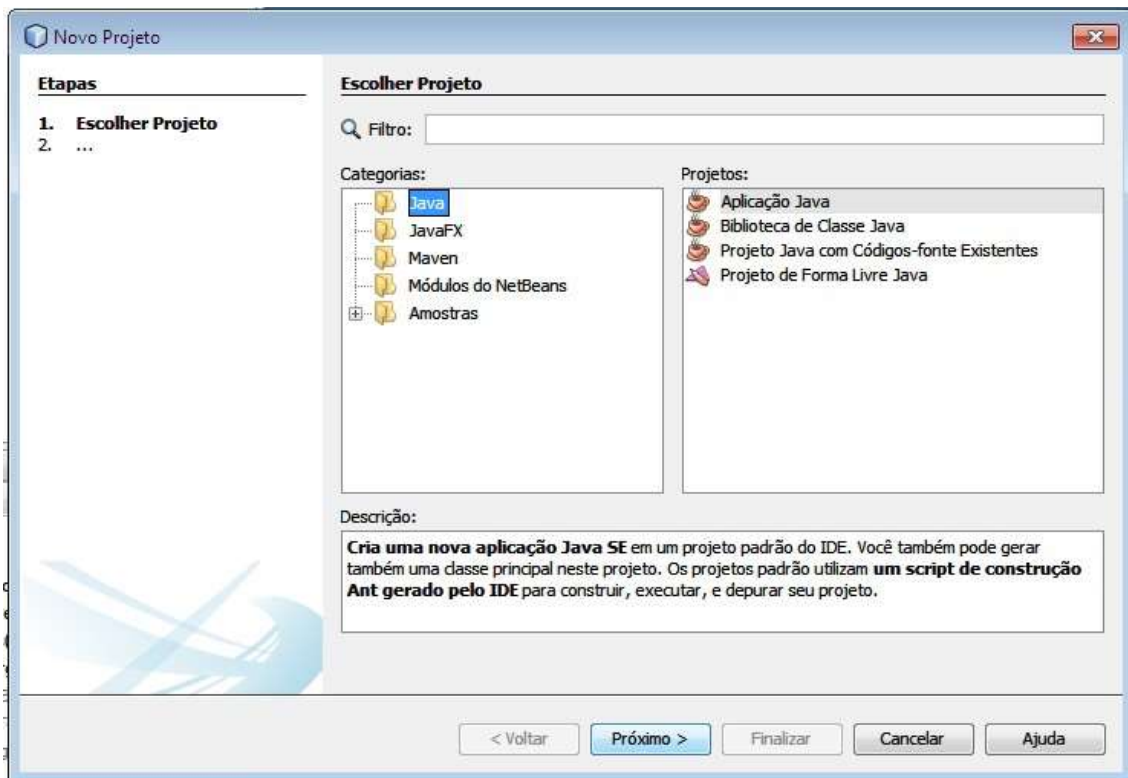


Figura 5 - Nomeando o projeto.

Após nomear o projeto clique em Finalizar conforme Figura 6:

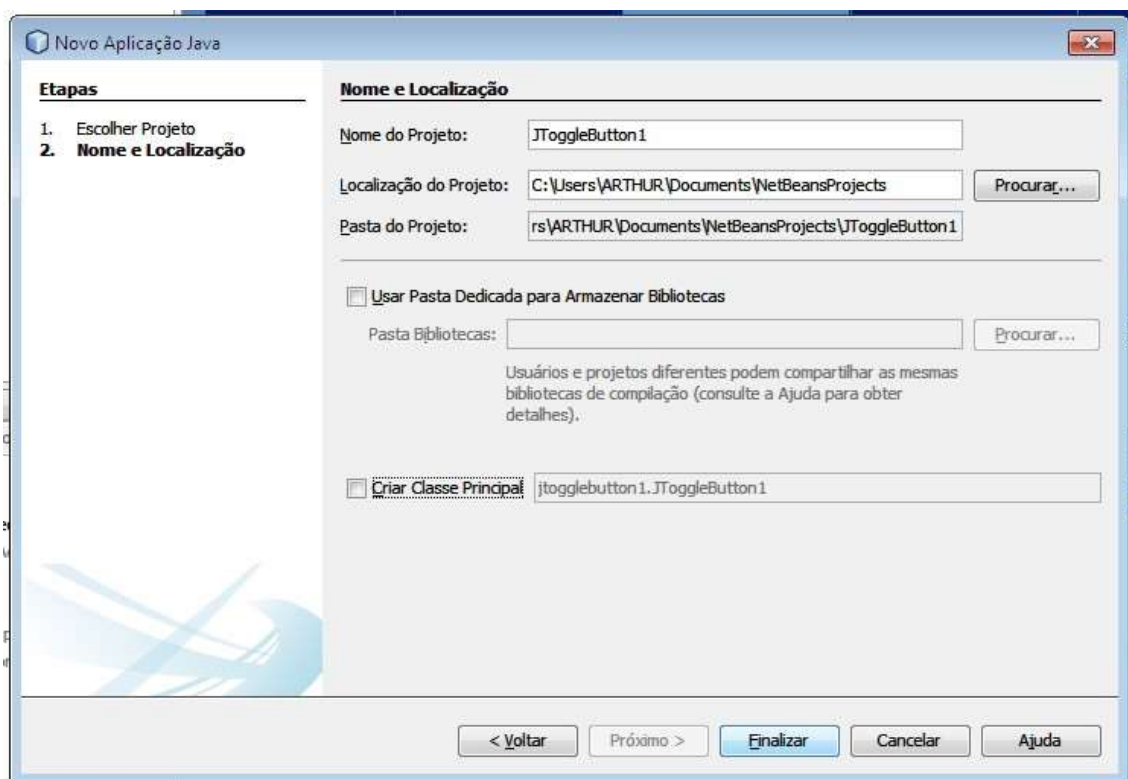


Figura 6 - Renomeando o projeto.

Projeto criado, agora iremos criar dois pacotes conforme Figura 7:

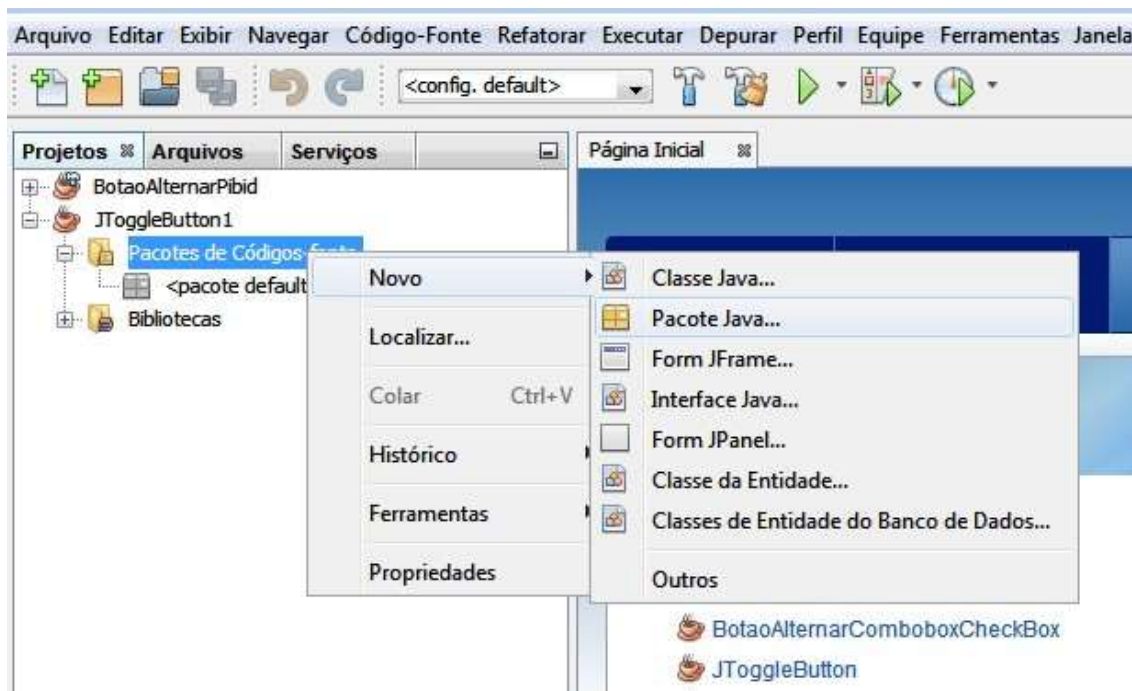


Figura 7 - Criação de pacotes.

O primeiro pacote será chamado de Telas conforme Figura 8:

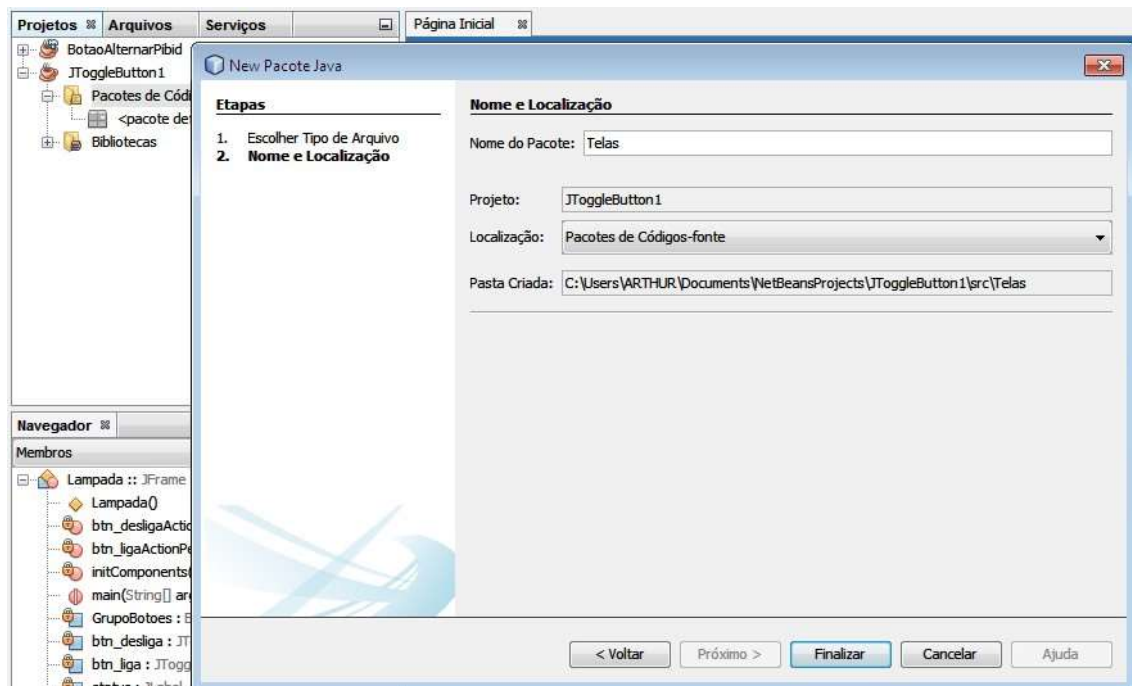


Figura 8 - Nomeando os pacotes.

O segundo pacote será chamado de Imagens conforme Figura 9:

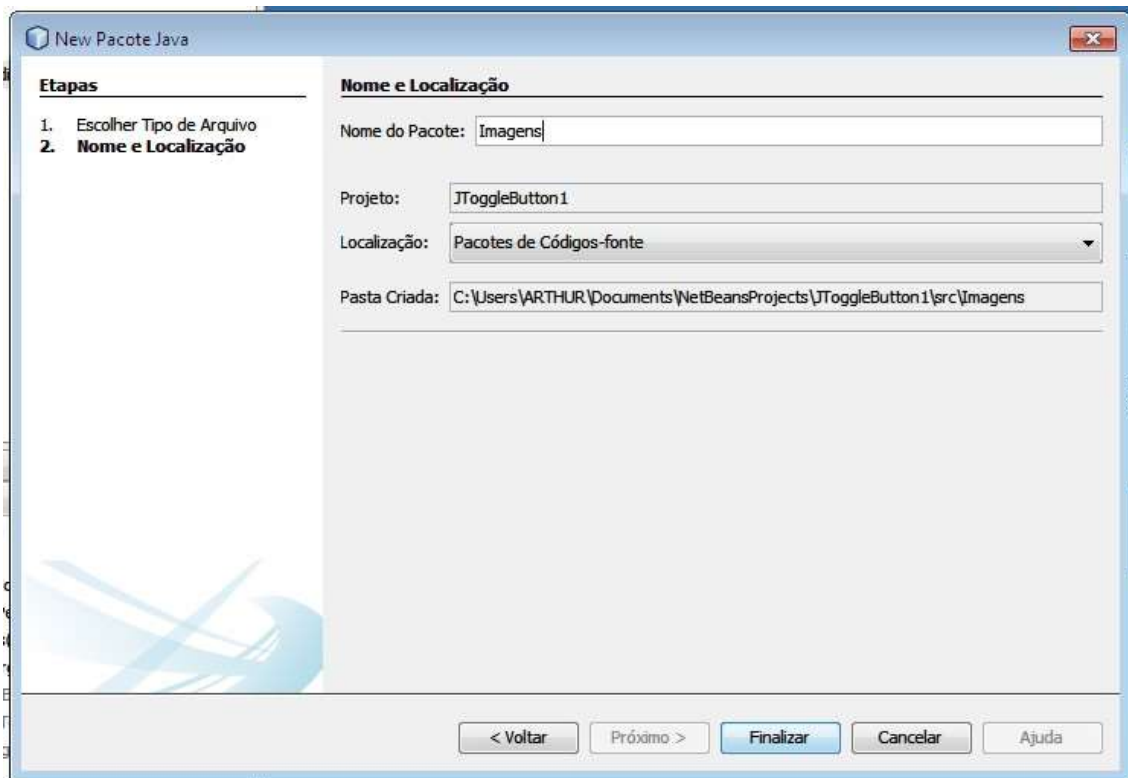


Figura 9 - Nomeando os pacotes.

Dentro do pacote Telas vamos criar um novo JFrame conforme Figura 10:

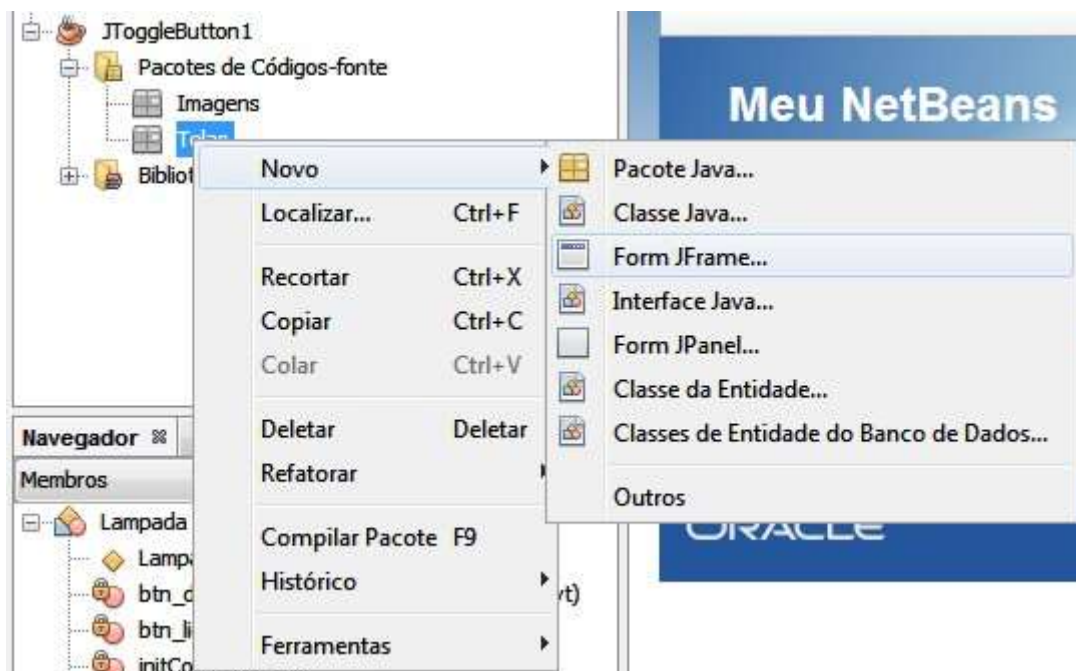


Figura 10 - Criando o JFrame.

O JFrame será chamado de TesteLampada conforme Figura 11:

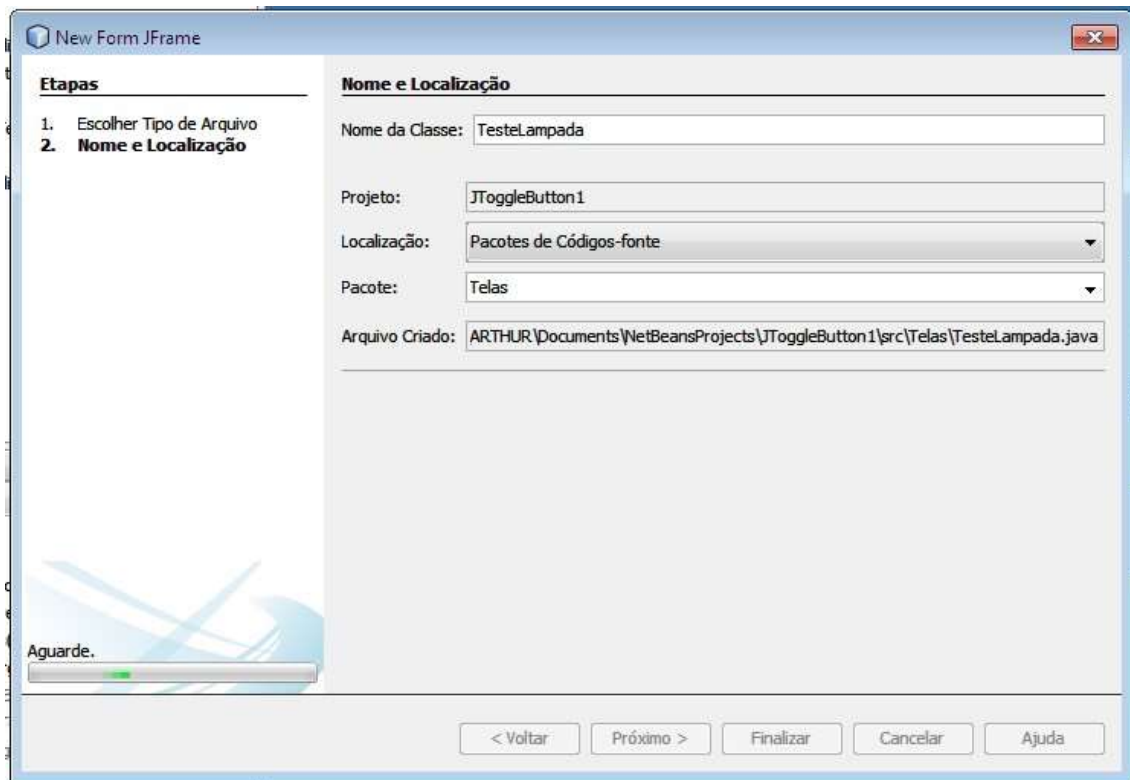


Figura 11 - nomeando o JFrame.

Tela do JFrame criado conforme Figura 12:

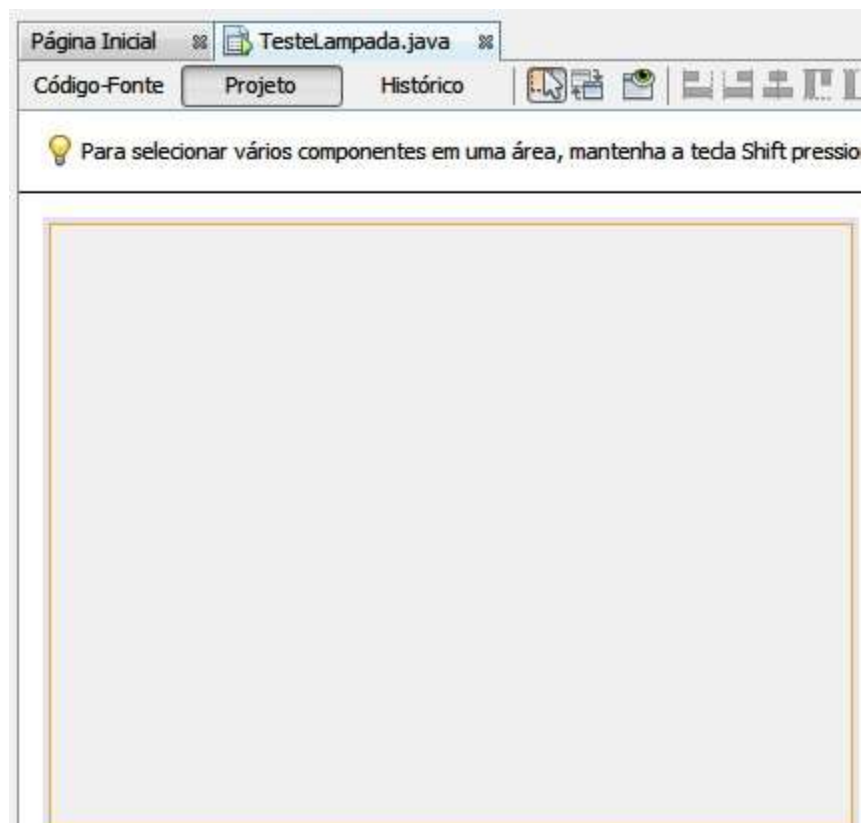


Figura 12 - Tela do JFrame.

Nesse teste da lâmpada iremos usar um botão JLabel, 1 botão ButtonGroup e dois botões JToggleButton, iremos inserir primeiro o botão JLabel, observe que quando selecionamos, o mesmo aparece uma informação destacada em amarelo conforme Figura 13:

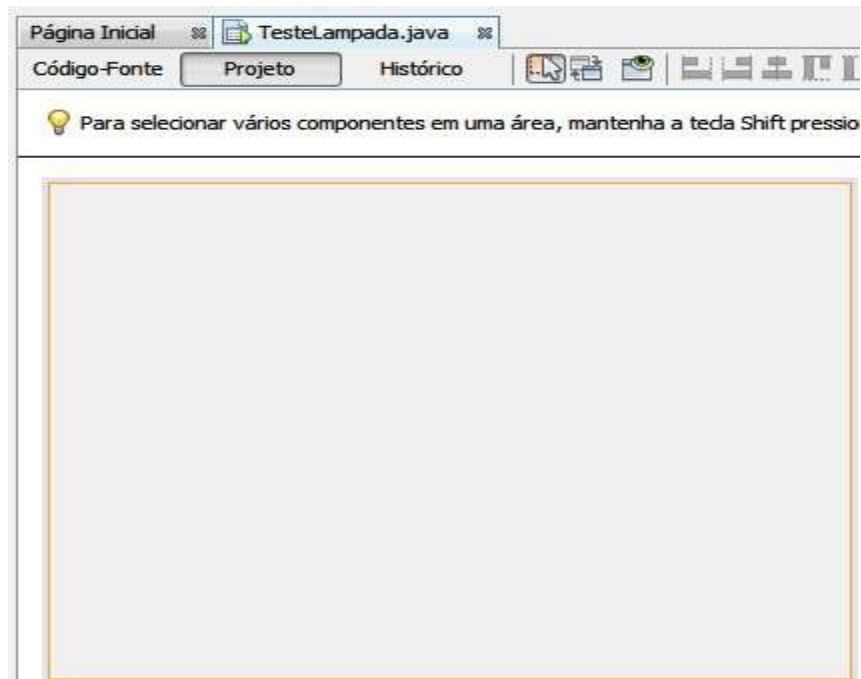


Figura 13 - Inserindo o botão JLabel.

Pronto o botão JLabel agora está inserido conforme Figura 14:

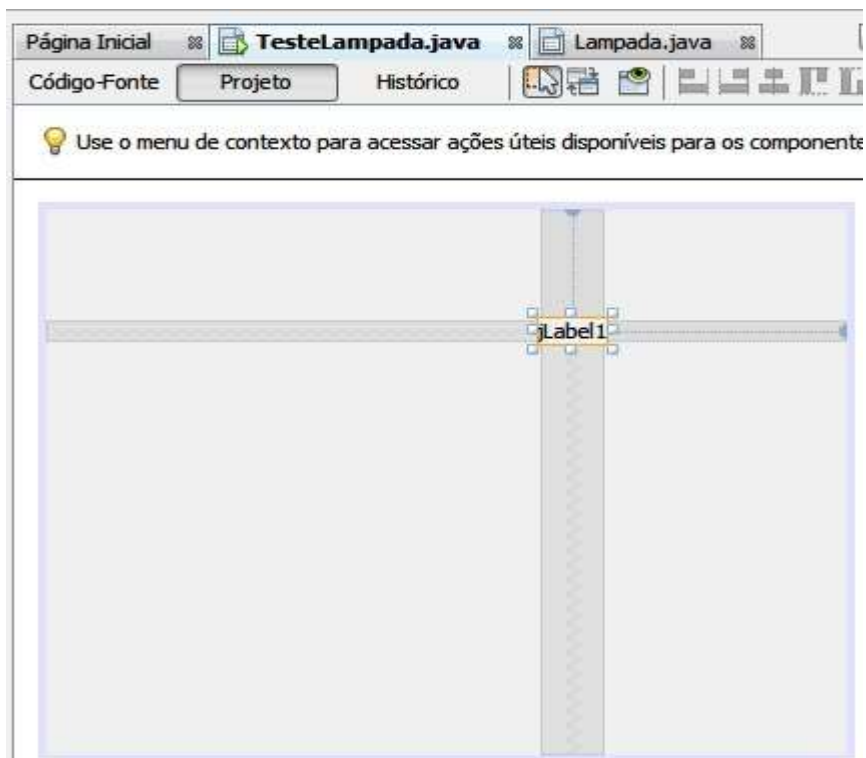


Figura 14 - Botão JLabel inserido.

Agora iremos redimensionar o tamanho deste JLabel conforme Figura 15:

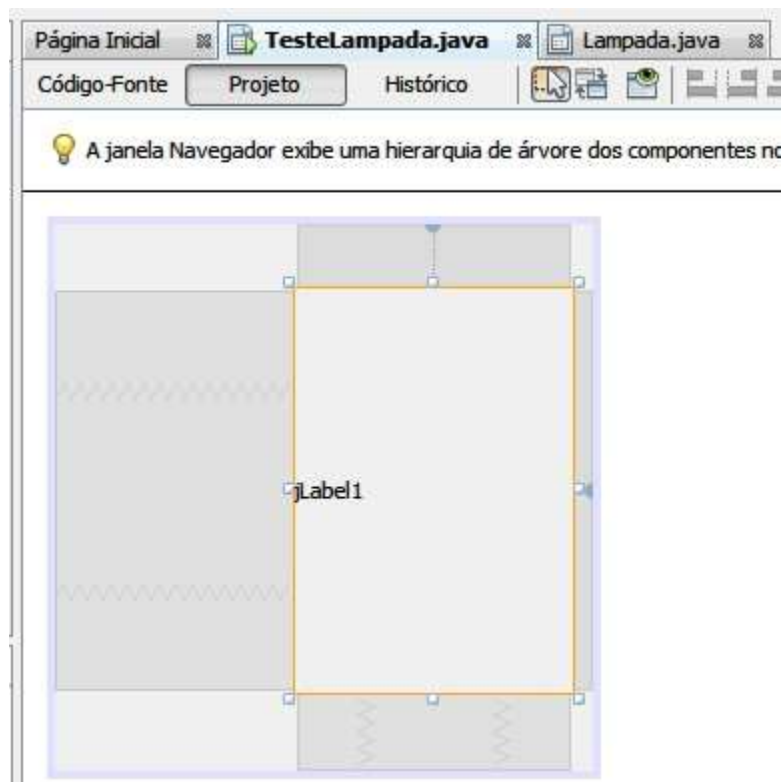


Figura 15 - Redimensionando o botão JLabel.

Iremos renomear “editar texto”, porém este não terá nenhum nome e sim uma imagem conforme as Figuras 16 à 19:



Figura 16 - Renomeando o JLabel.

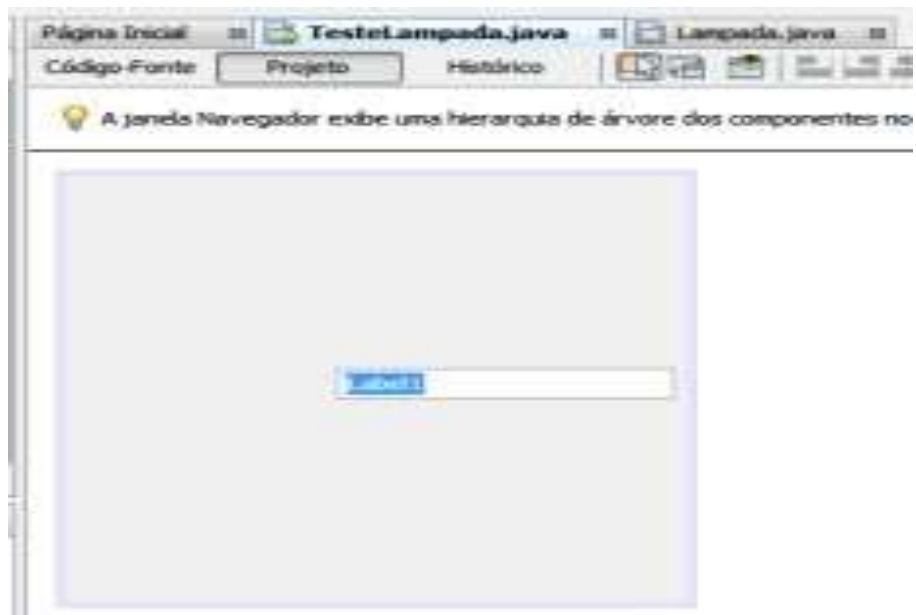


Figura 17 - Renomeando o JLabel.

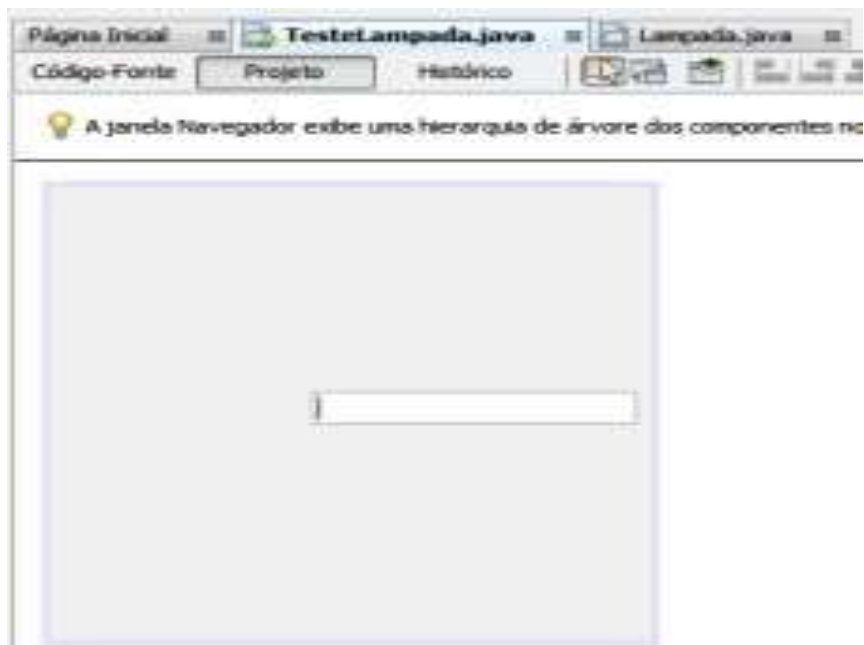


Figura 18 - Renomeando o JLabel.



Figura 19 - Renomeando o JLabel.

Agora no lugar do Texto que tinha no JLabel iremos inserir uma imagem e deixa-la centralizada conforme as Figuras 20 a 28:

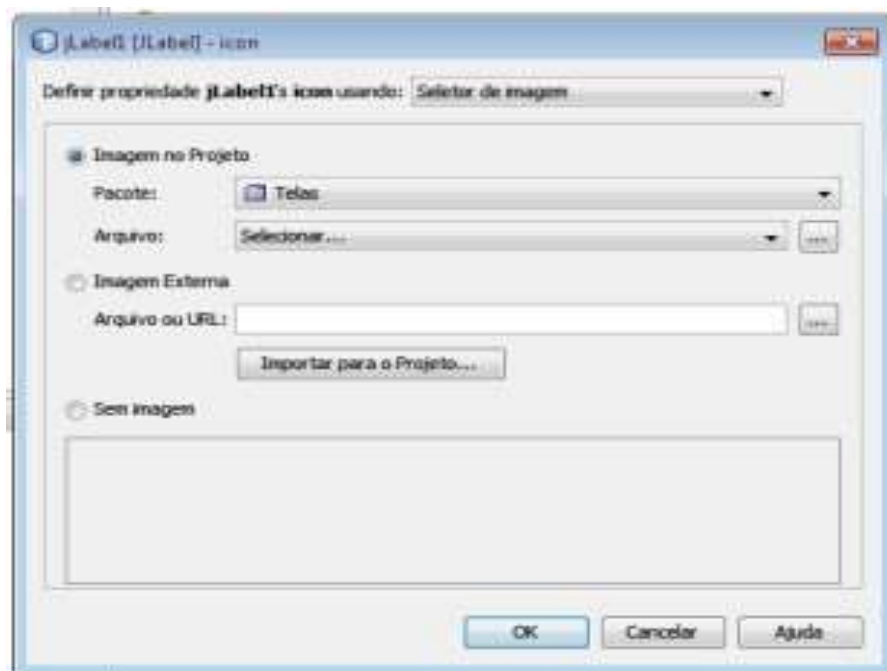


Figura 20 - Inserindo Imagem e centralizando.

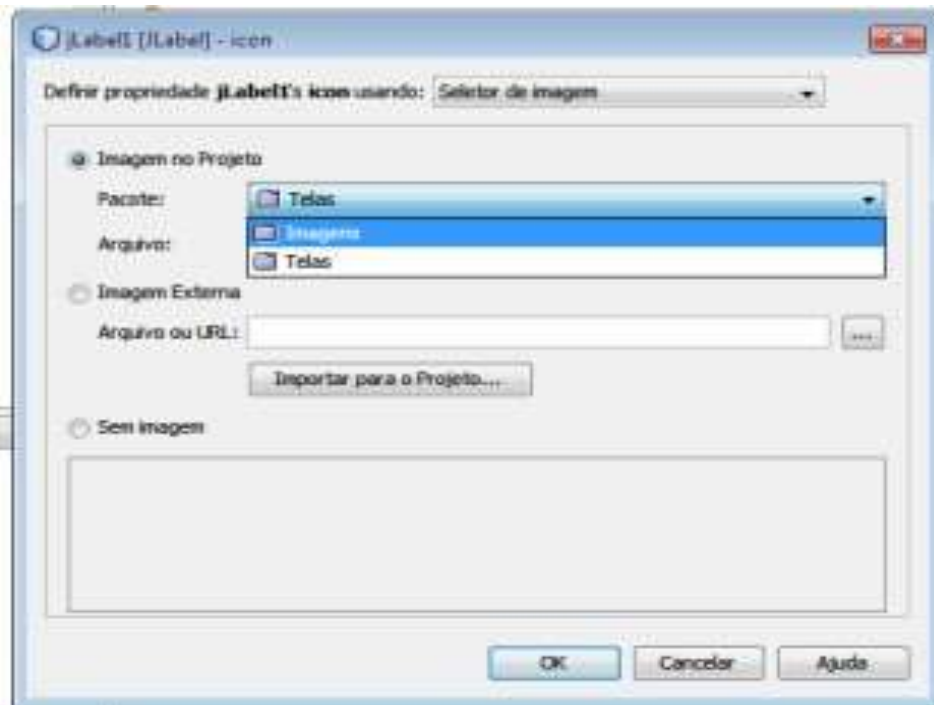


Figura 21 - Inserindo Imagem e centralizando.

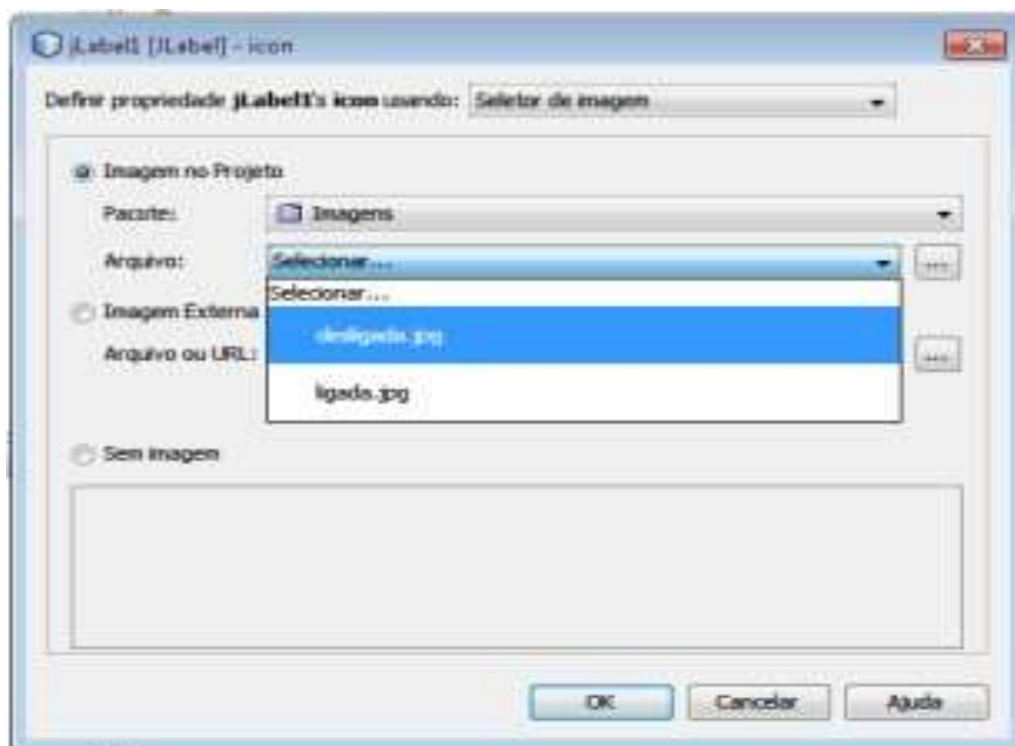


Figura 22 - Inserindo Imagem e centralizando.

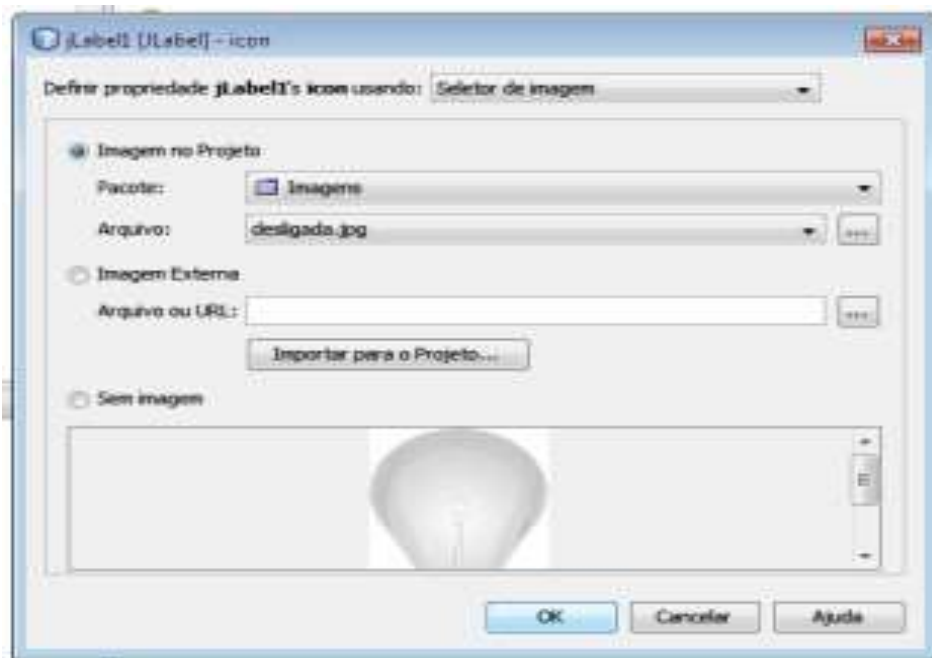


Figura 23 - Inserindo Imagem e centralizando.



Figura 24 - Inserindo Imagem e centralizando.

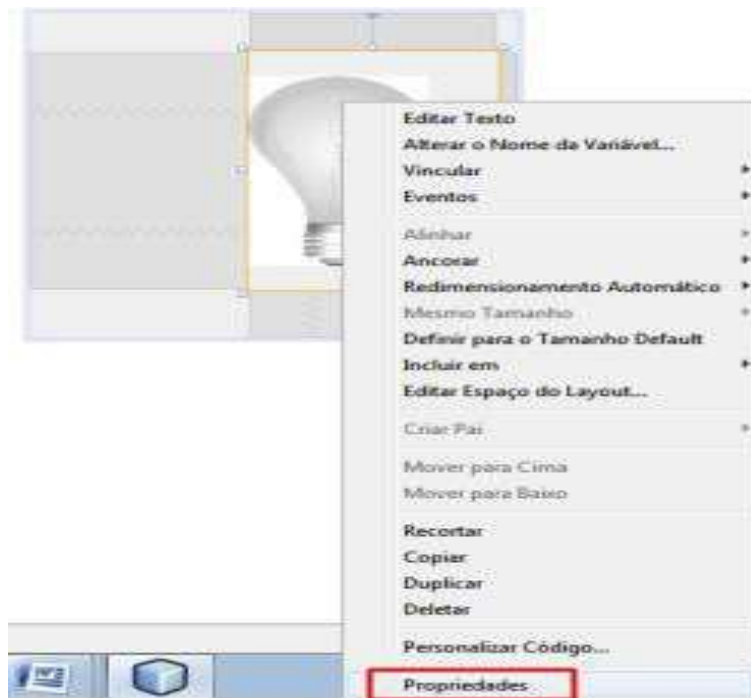


Figura 25 - Inserindo Imagem e centralizando.

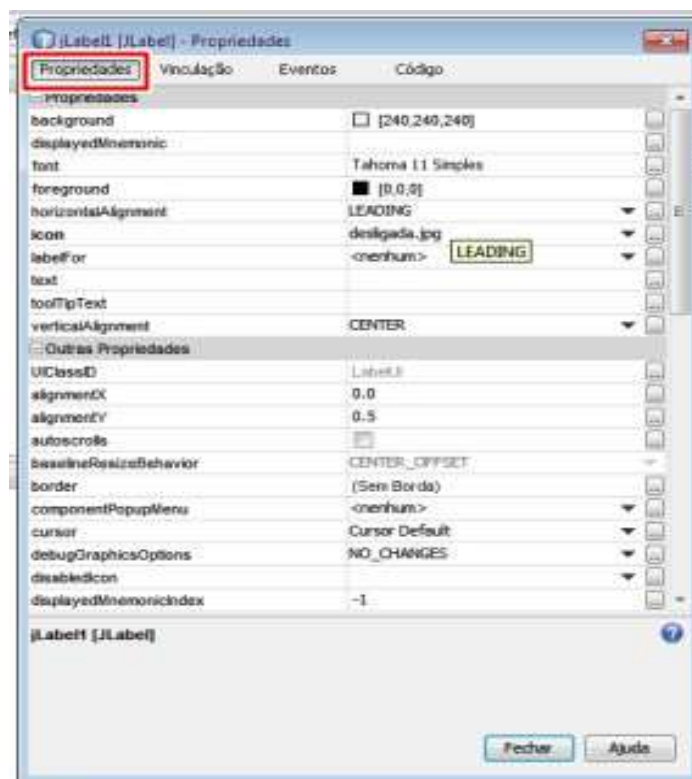


Figura 26 - Inserindo Imagem e centralizando.

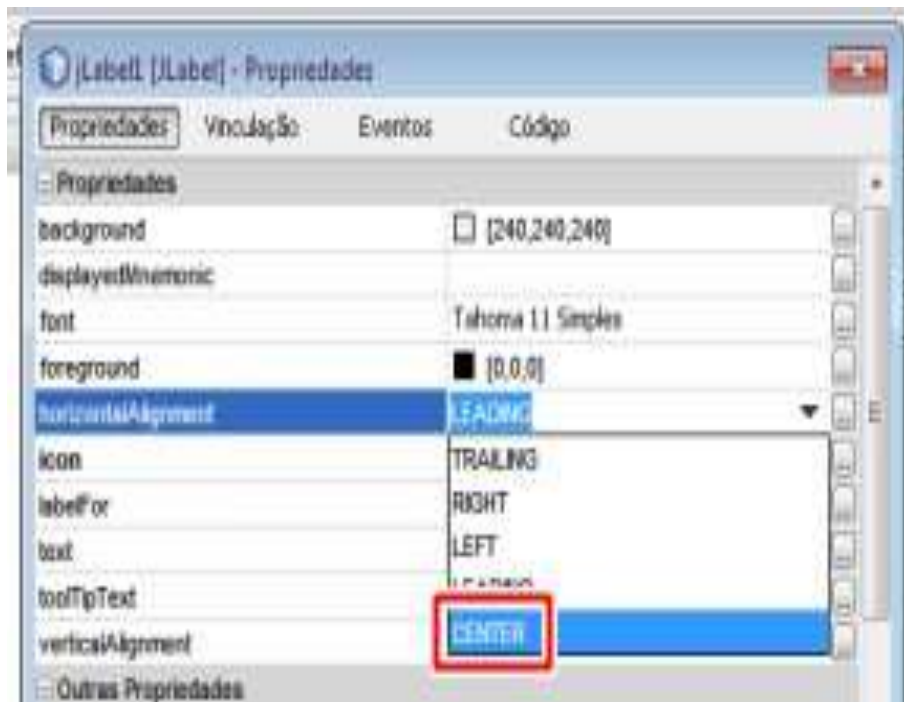


Figura 27 - Inserindo imagem e centralizando.



Figura 28 - Inserindo imagem e centralizando.

Após inserirmos o botão JLabel e configura-los iremos agora inserir os botões JToggleButton, lembrando que usaremos dois botões JToggleButton. O primeiro será renomeado e será chamado de “Ligar” conforme as Figuras 29 a 31:

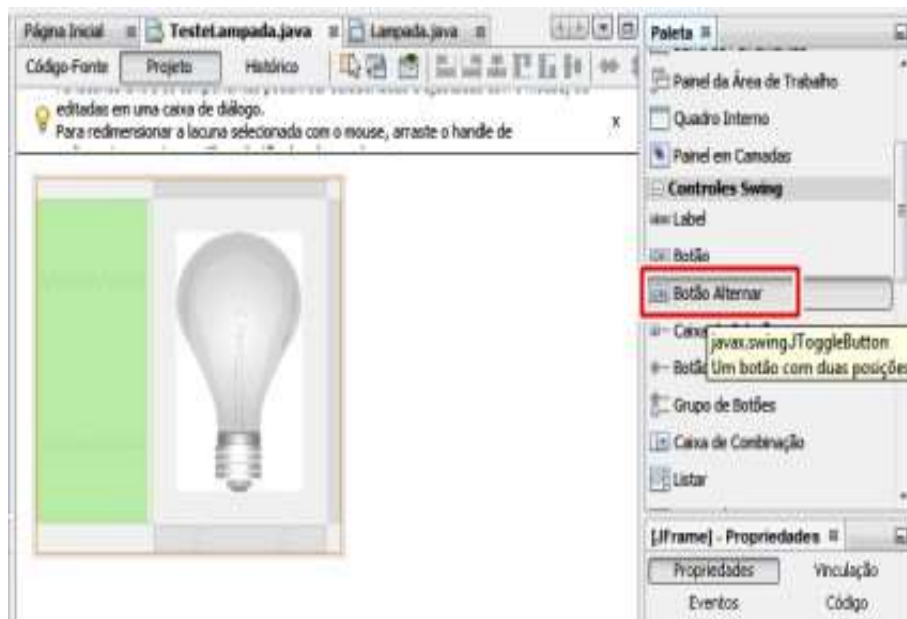


Figura 29 - Inserindo e renomeando o botão JToggleButton.



Figura 30 - Inserindo e renomeando o botão JToggleButton.

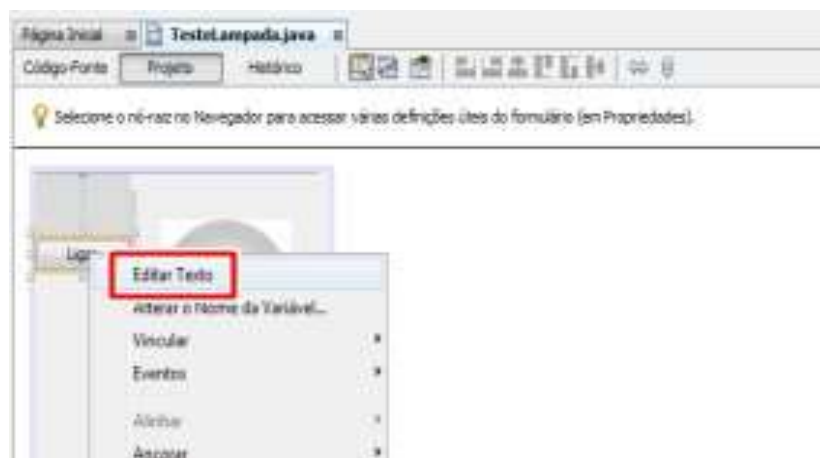


Figura 31 - Inserindo e renomeando o botão JToggleButton.

O primeiro botão foi inserido e renomeado, agora iremos inserir o segundo botão chamado de “Desligar”, conforme Figura 32:



Figura 32 - Inserindo e renomeando o botão JToggleButton.

Iremos inserir agora o botão ButtonGroup que serve para deixar os dois JToggleButton com a funcionalidade Desligar e Ligar conforme as Figuras 33 e 34:



Figura 33 - Inserindo o botão ButtonGroup.



Figura 34 - Inserindo o botão ButtonGroup.

Iremos fazer um teste para verificar como está o design do nosso programa para isso devemos pressionar Shift + F6 e irá executar e abrirá uma nova janela conforme Figura 35:

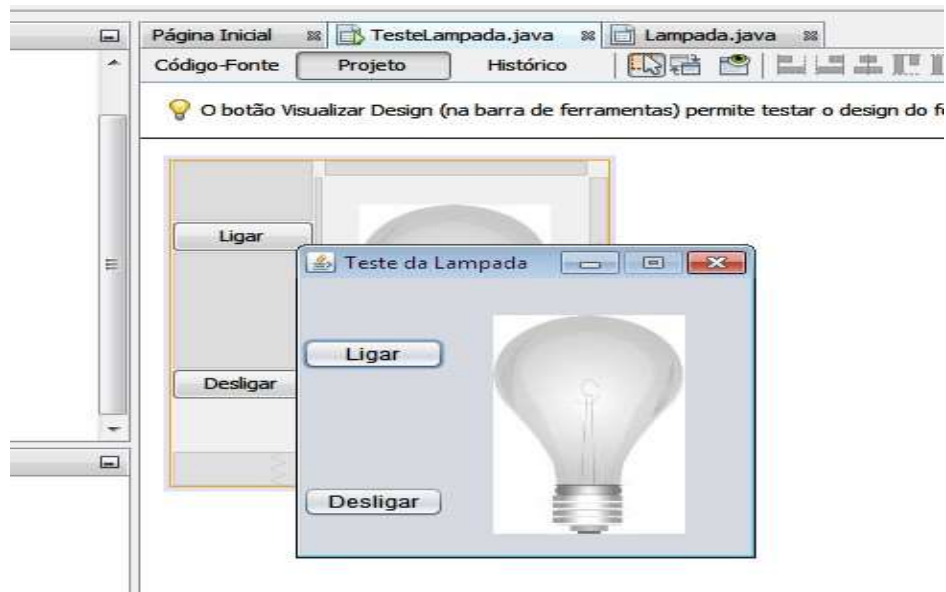


Figura 35 - Testando o programa.

Agora vamos dar funcionalidade aos JToggleButton “ Botão LIGAR e DESLIGAR” para isso clicamos em cima de um dos botões JToggleButton com o botão direito do mouse e iremos inserir um Evento - Action - actionPerformed conforme mostra a Figura 36:

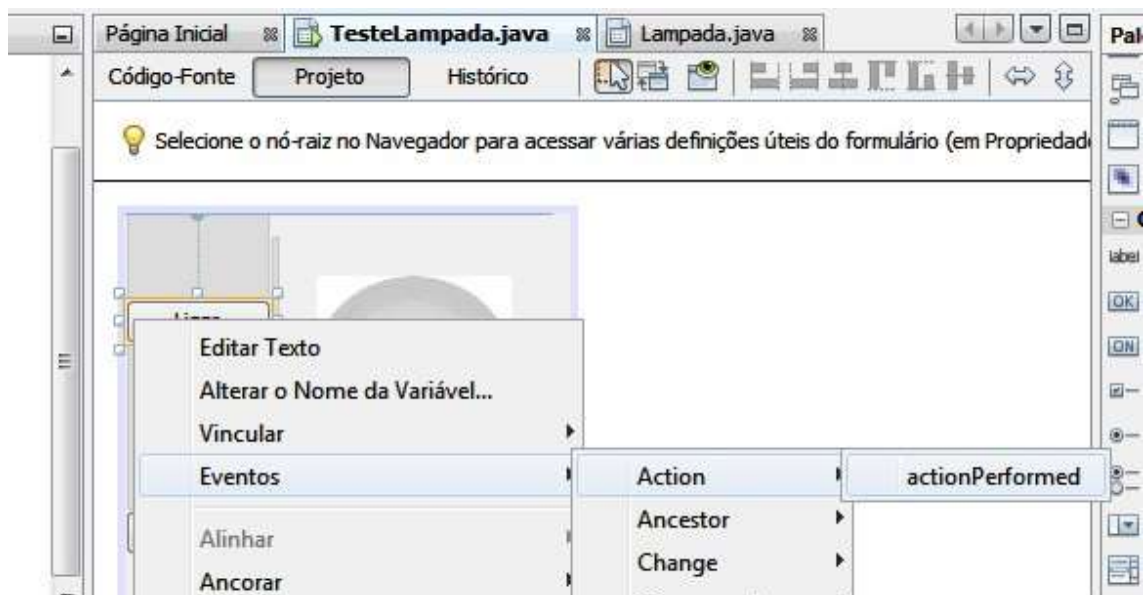
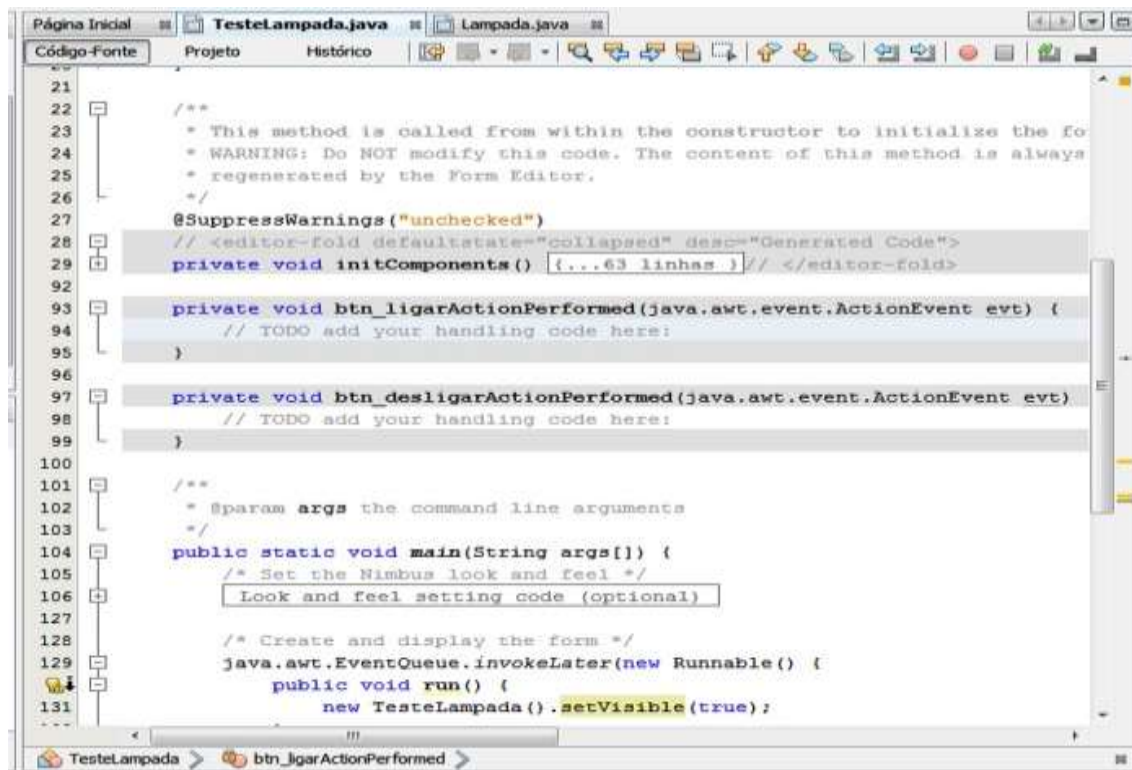


Figura 36 - Inserindo um evento actionPerformed.

Agora iremos declarar as variáveis do JToggleButton para a funcionalidade dos botões “Ligar e Desligar” conforme Figura 37. Observe que nas linhas 99 até 101 e 107 até a 109 as Variáveis são declaradas conforme a Figura 38:



```
21
22
23  /**
24   * This method is called from within the constructor to initialize the fo
25   * WARNING: Do NOT modify this code. The content of this method is always
26   * regenerated by the Form Editor.
27   */
28  @SuppressWarnings("unchecked")
29  // <editor-fold defaultstate="collapsed" desc="Generated Code">
30  private void initComponents() { ...63 linhas } // </editor-fold>
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93  private void btn_ligarActionPerformed(java.awt.event.ActionEvent evt) {
94      // TODO add your handling code here:
95  }
96
97  private void btn_desligarActionPerformed(java.awt.event.ActionEvent evt)
98      // TODO add your handling code here:
99  }
100
101  /**
102   * @param args the command line arguments
103   */
104  public static void main(String args[]) {
105      /* Set the Nimbus look and feel */
106      Look and feel setting code (optional)
107
108      /* Create and display the form */
109      java.awt.EventQueue.invokeLater(new Runnable() {
110          public void run() {
111              new TesteLampada().setVisible(true);
112          }
113      });
114  }
```

Figura 37 - Declarando as variáveis dos botões Ligar e Desligar.

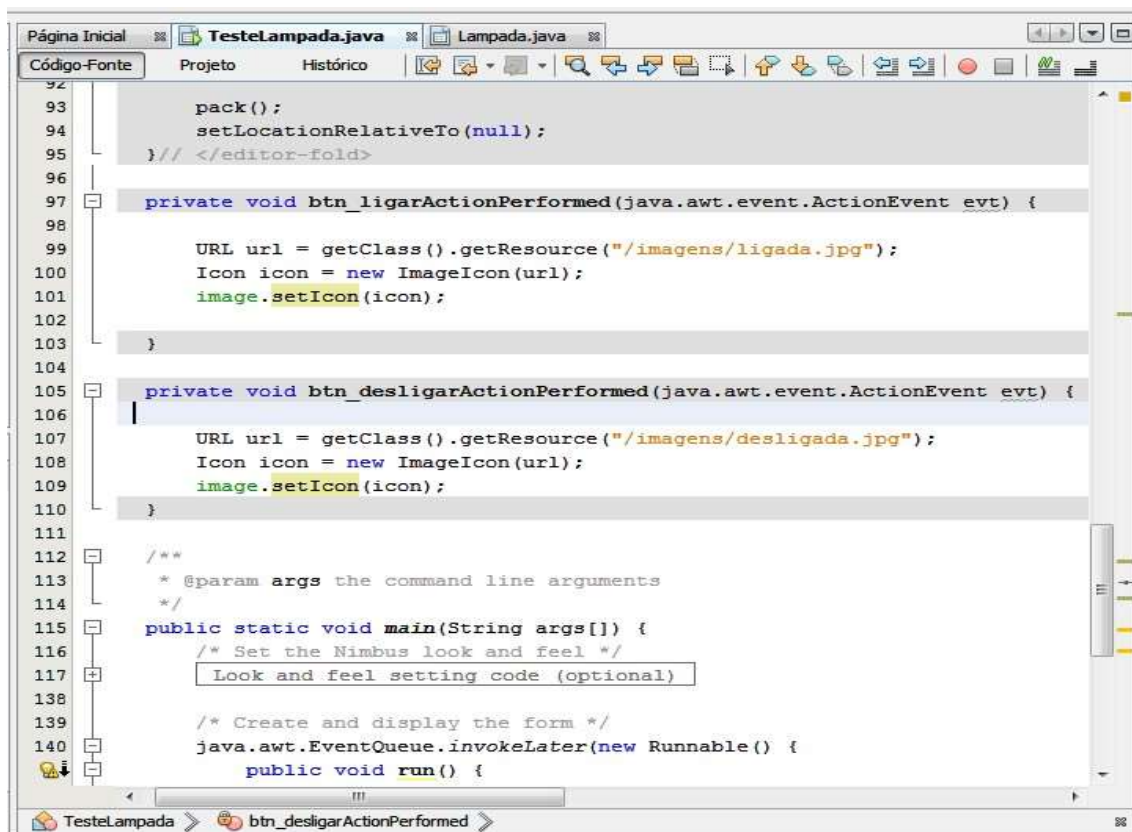


Figura 38 - Declarando as variáveis dos botões Ligar e Desligar.

Agora iremos testar nosso programa, para isso pressione a tecla Shift + F6, irá abrir uma nova Janela mostrando a lâmpada desligada conforme a Figura 39:

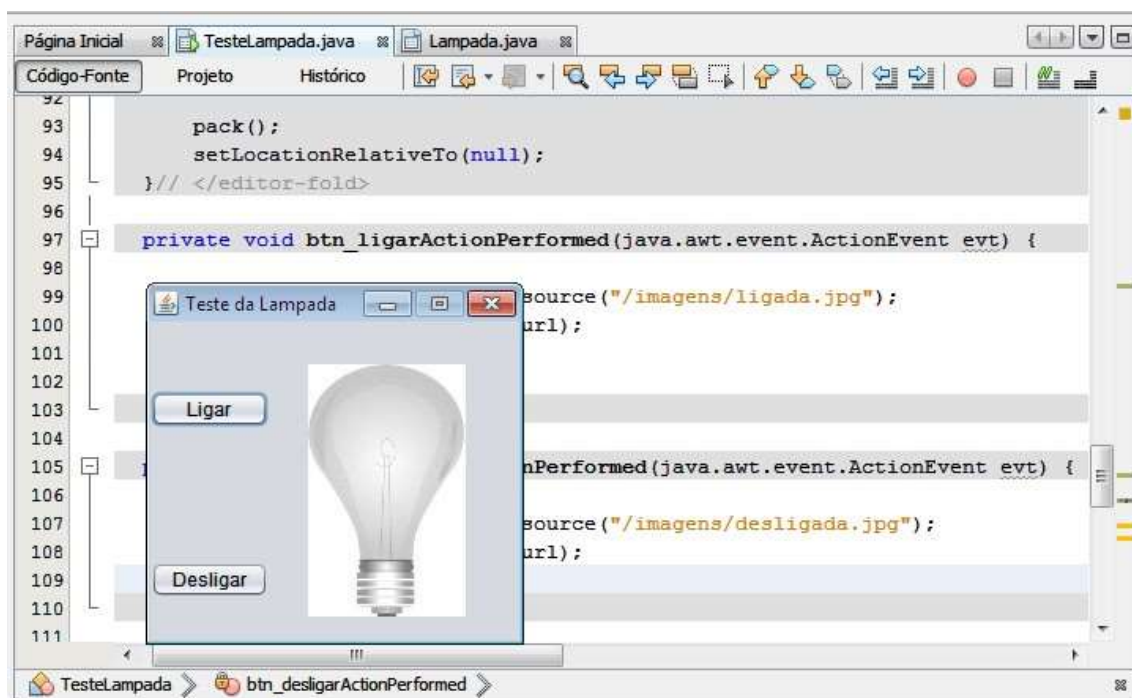


Figura 39 - Testando o programa.

Agora ao pressionar o botão “Ligar” a imagem da lâmpada mudará para a cor amarela que corresponde à ligada conforme Figura 40:

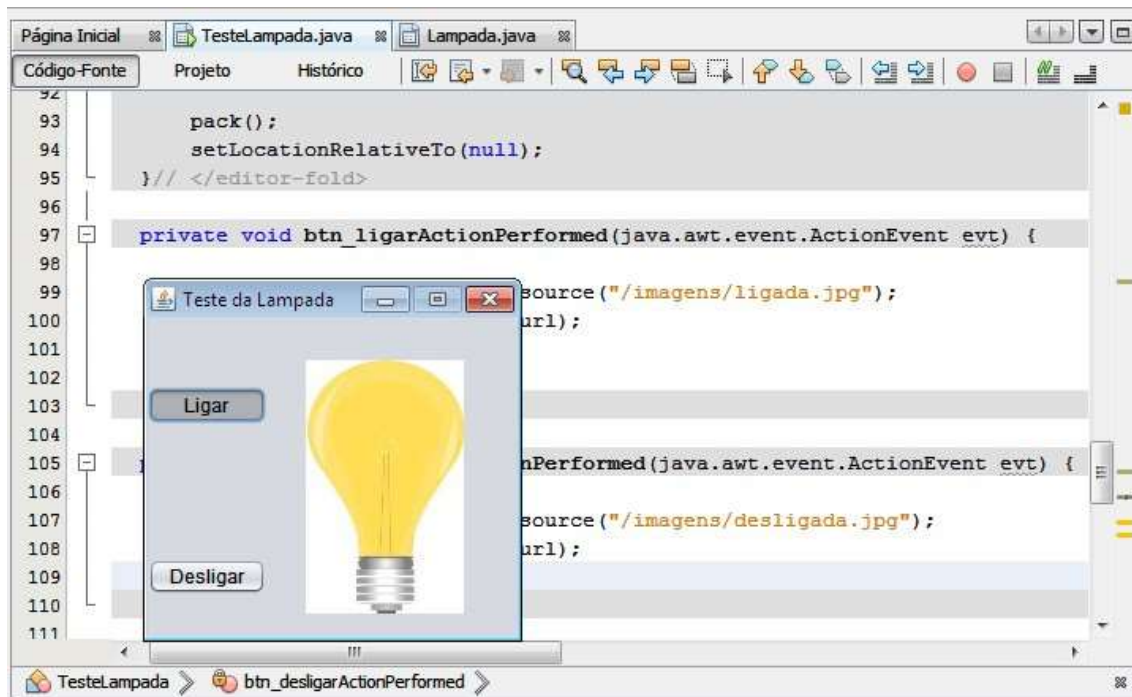


Figura 40 - Testando o botão Ligar.

Agora ao pressionar o botão “Desligar” a imagem da lâmpada mudará para a cor cinza que corresponde à desligada conforme Figura 41:

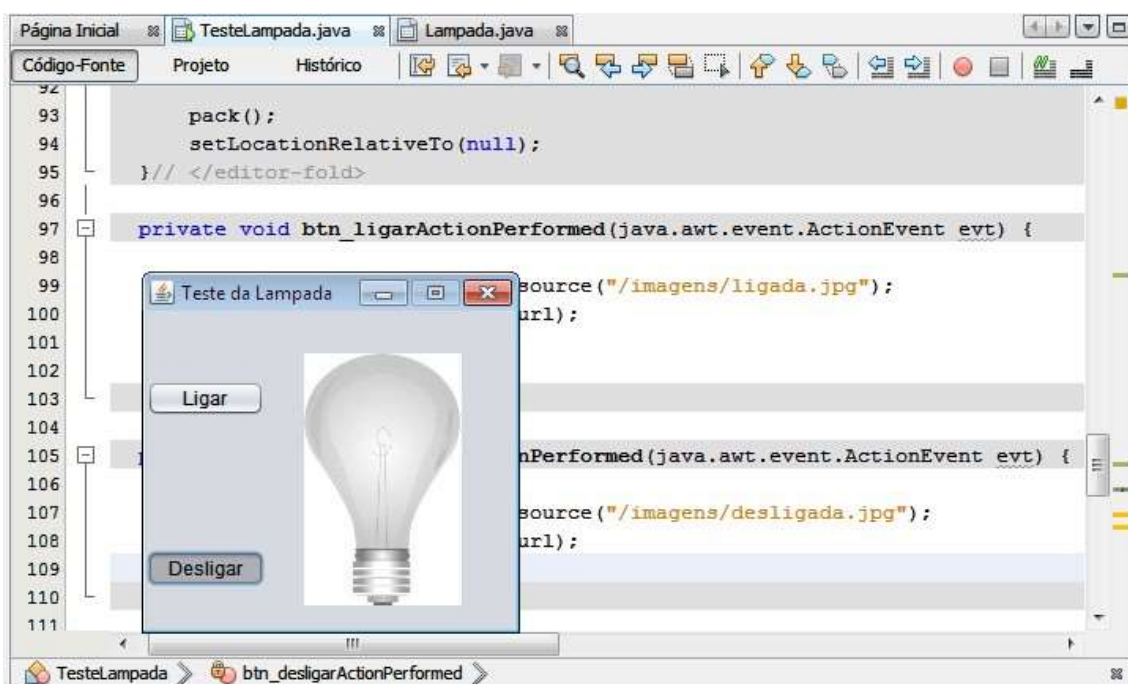


Figura 41 - Testando o botão Desligar.

Programa funcionando conforme instruções.

JTextArea

O JTextArea fornece uma área para manipular múltiplas linhas de texto. JTextArea é uma subclasse de JTextComponent, que declara métodos comuns para JTextFields, JTextAreas e vários outros componentes GUI baseados em texto.

Você pode encontrar informações e exemplos de utilização de todos os componentes de texto em uso abaixo.

Para criar uma classe java com o JTextArea, primeiro é necessário no Netbeans clicar em arquivo como mostra a figura 1 depois criar novo projeto. figura 42:

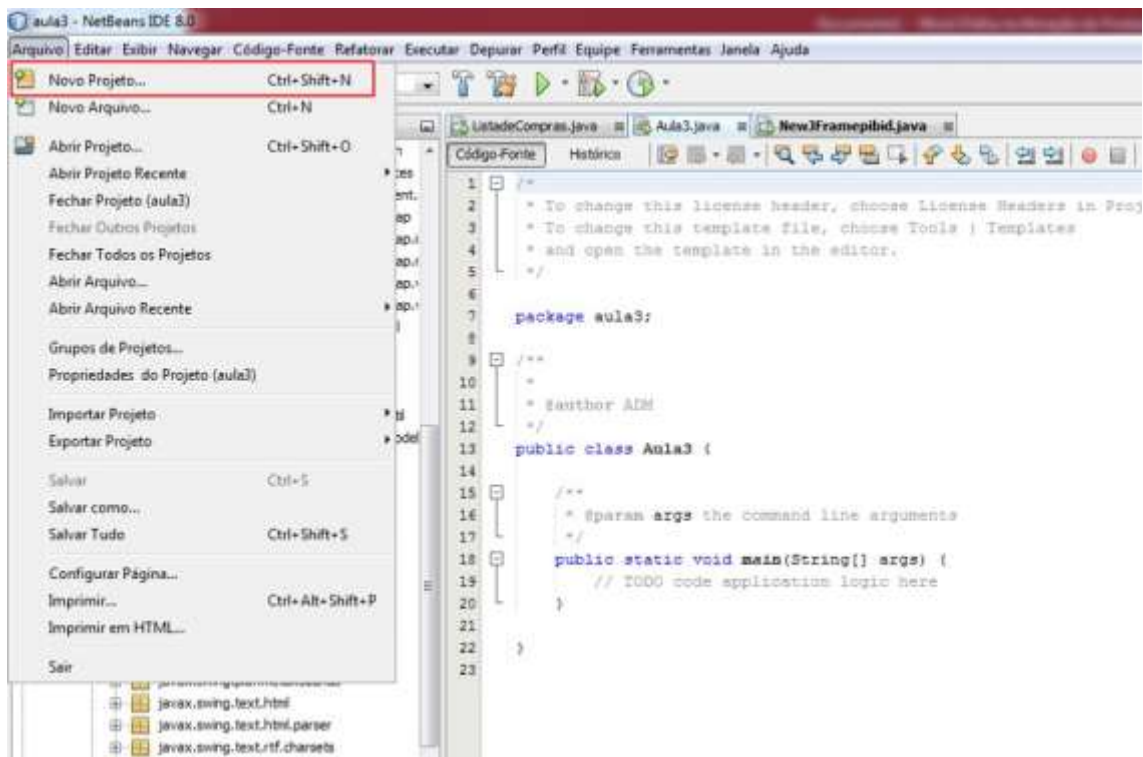


Figura 42 - Criando uma classe Java.

Criar classe java, figura 43:

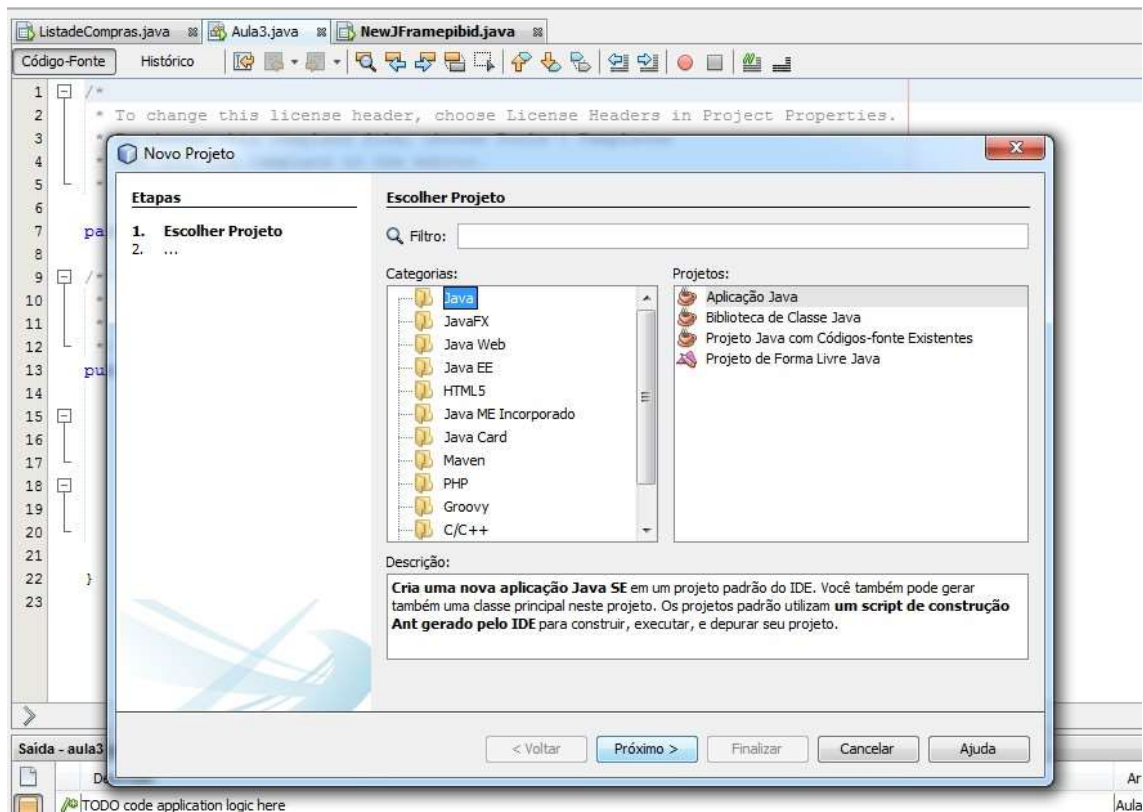
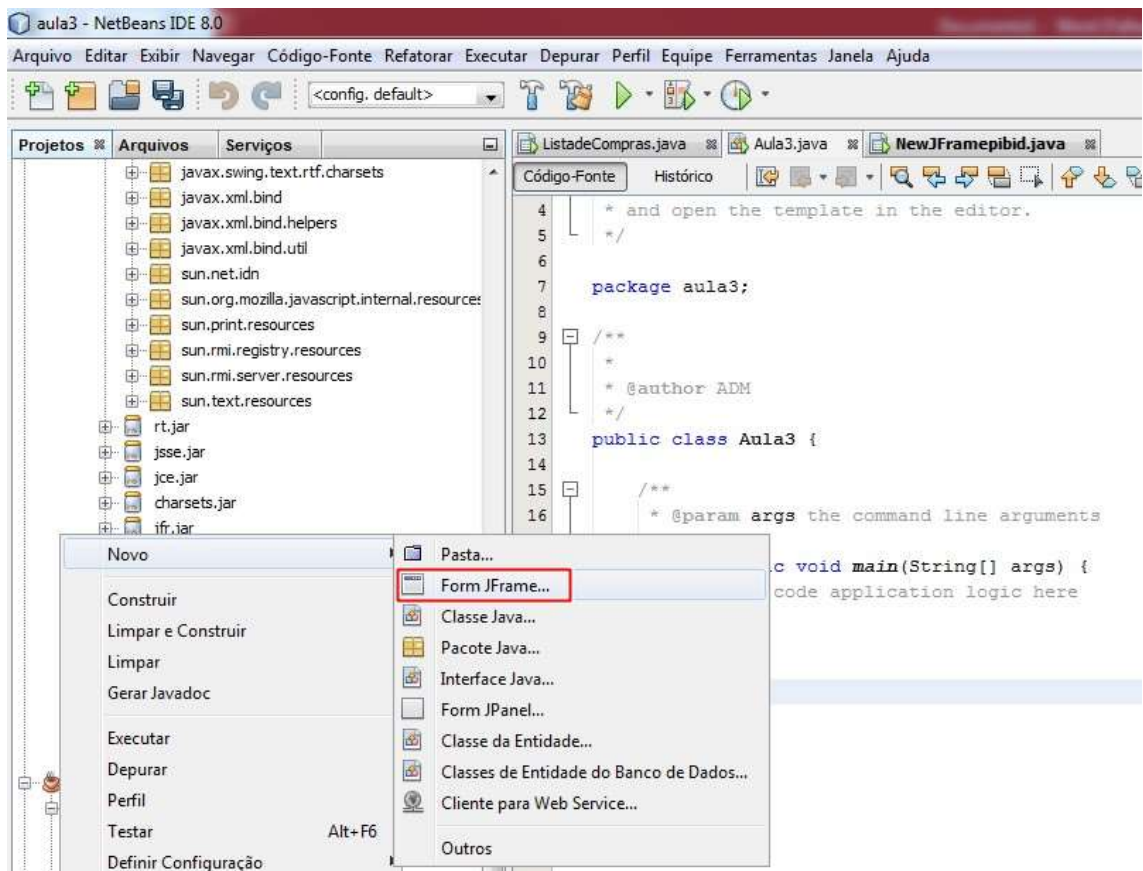


Figura 43 - Criando uma classe Java.

Agora vamos criar novo JFrame, figura 44:



Aparecerá uma tela semelhante a da figura abaixo, figura 45:

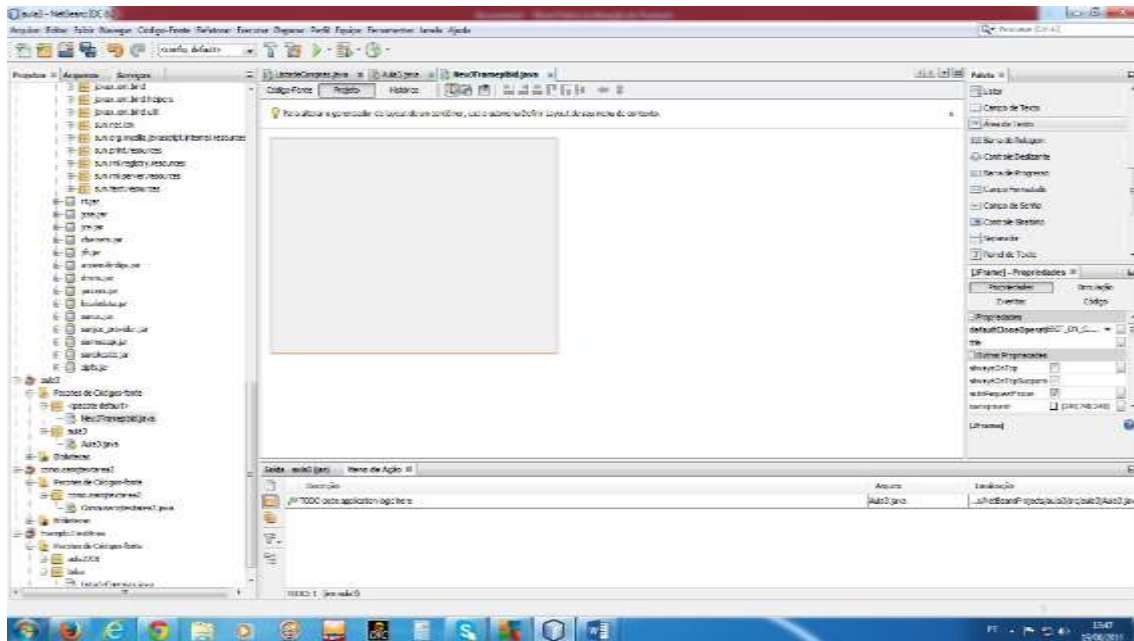


Figura 45 - JTextArea.

Para inserir o JTextArea clique na aba peleta sobre o campo de texto ou JTextArea e arraste para o projeto, figura 46:

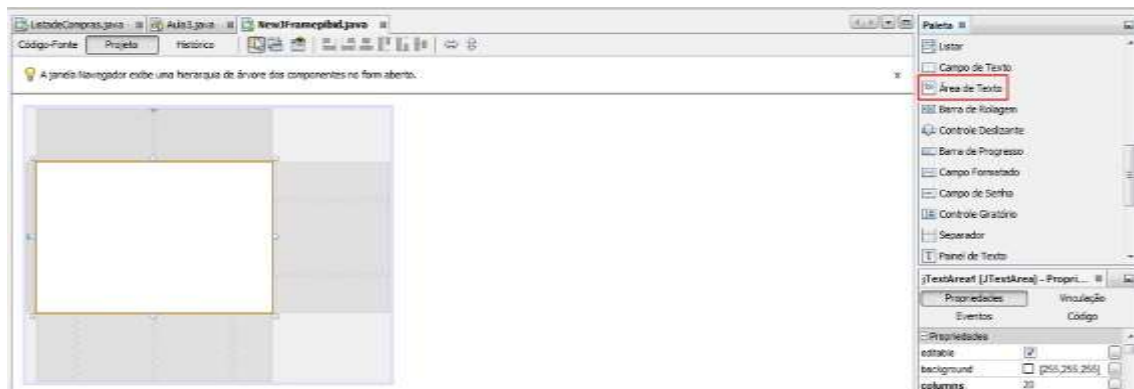


Figura 46 - Inserindo o JTextArea.

Aparecerá uma imagem semelhante a da figura abaixo, figura 47:

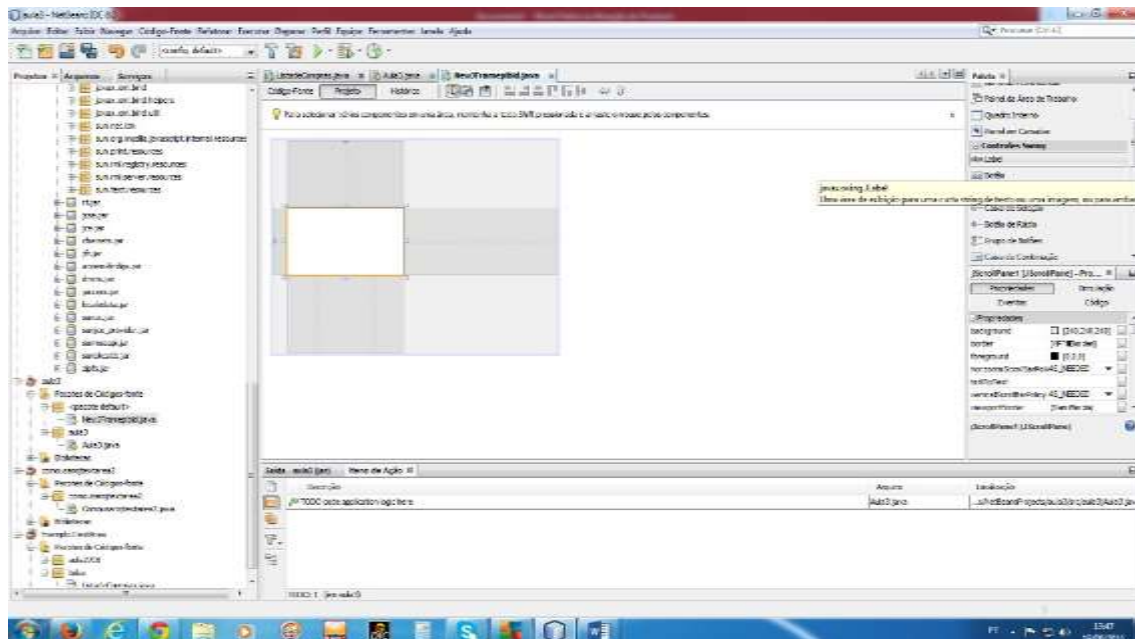


Figura 47 - JTextArea.

Vamos usar como exemplo a criação de um projeto chamado de lista de materias escolar, para isso usaremos alguns componetes como: Jtexarea, Jbutton, Jtextfield e JLabel.

Para receber a variável lista de compras usaremos um Label que está dentro da paleta na opção controles Swing, o JLabel receberá o nome lista de materias, figura 48:

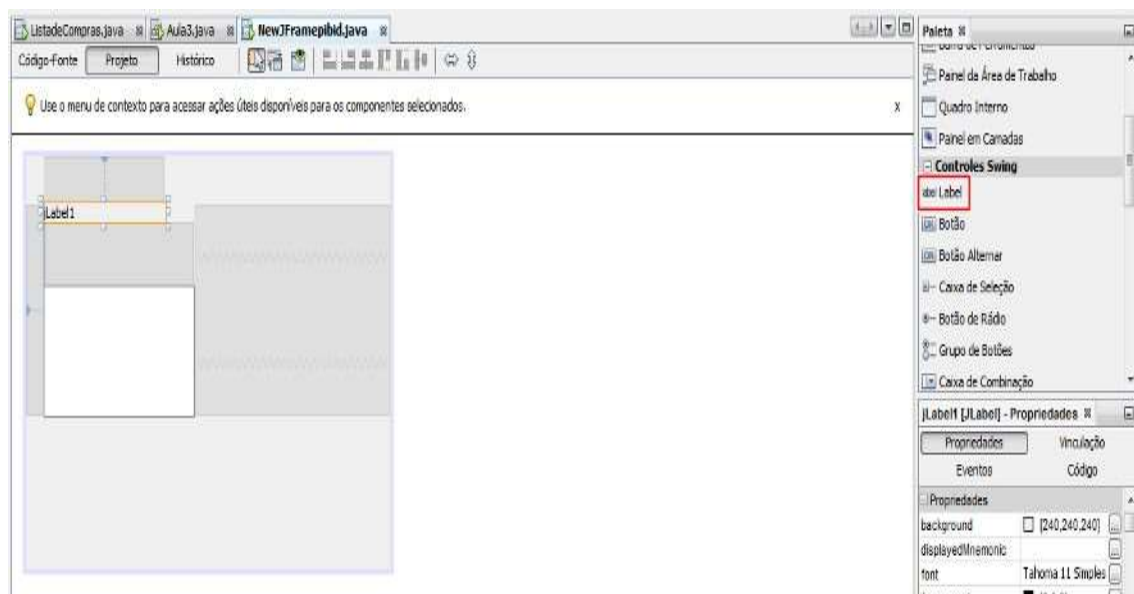


Figura 48 - Inserindo o Label no projeto.

Clique com botão direito do mouse em cima do Label para editar o texto, figura 49 e 50:

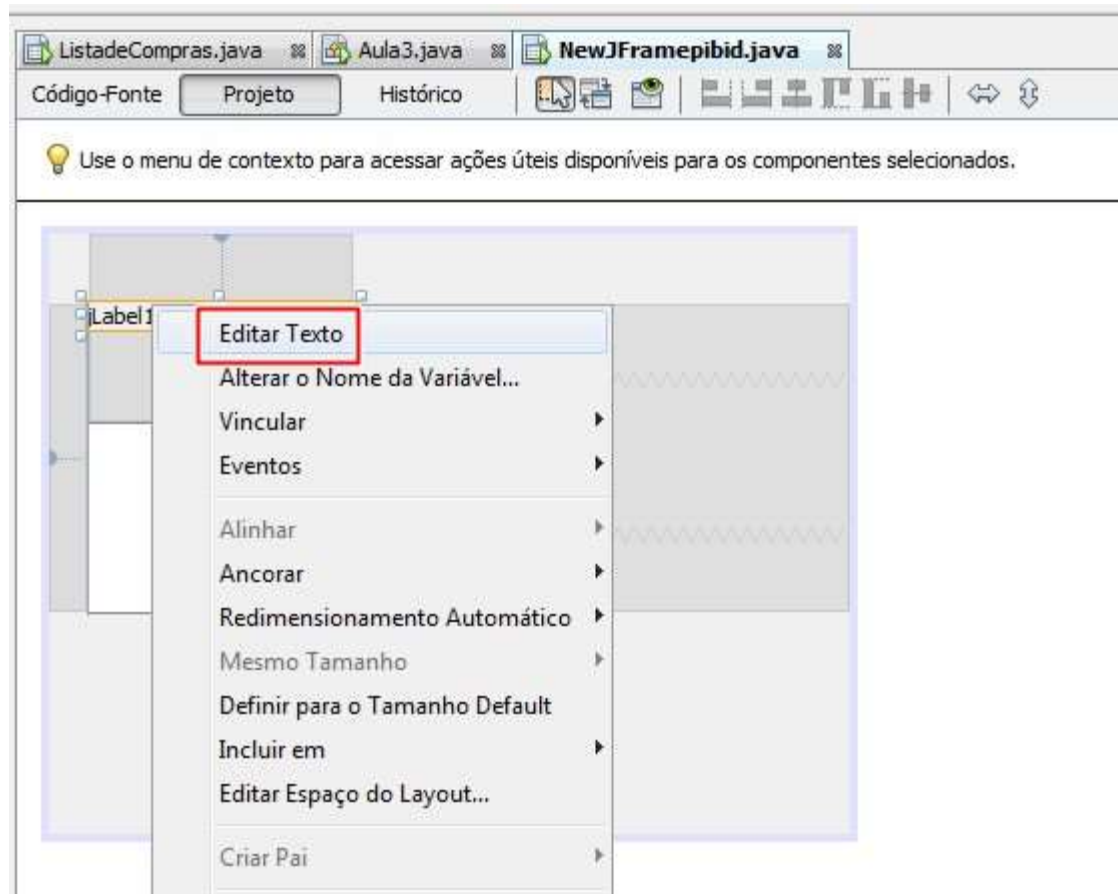


Figura 49 - Editando o texto Label.

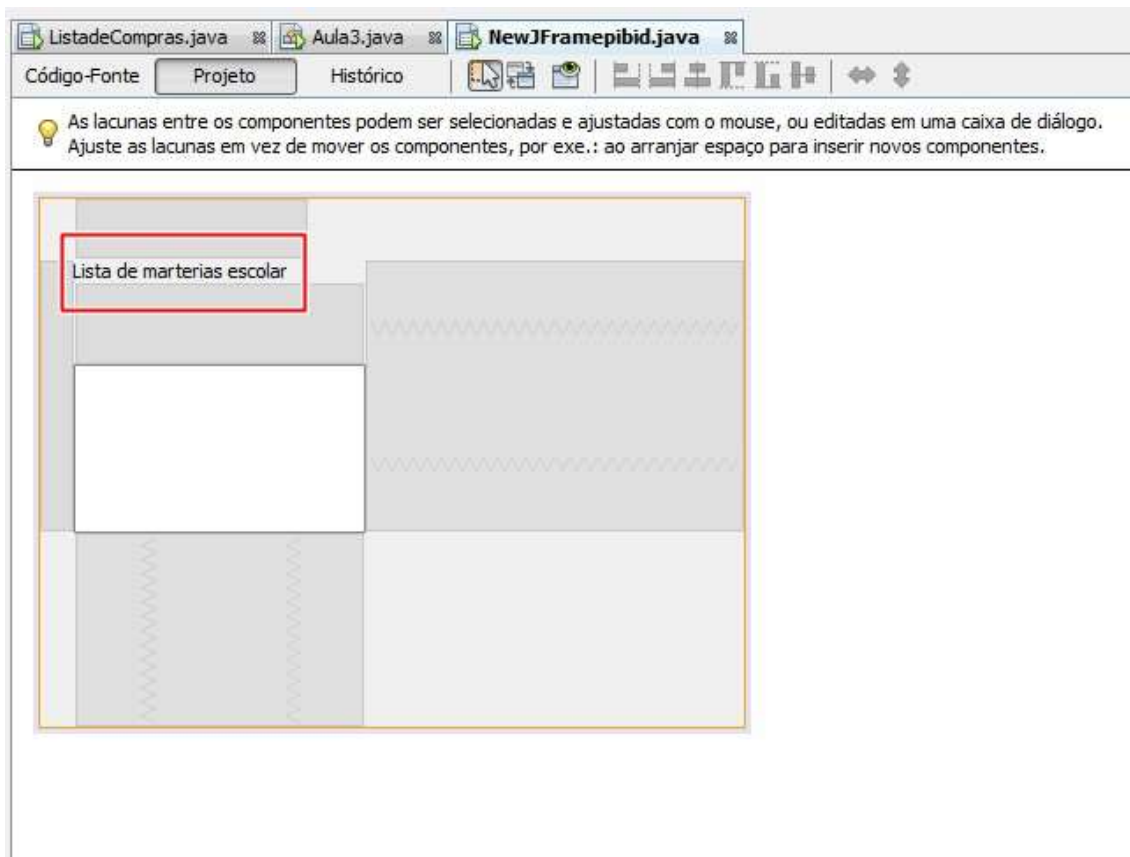


Figura 50 - Label após o nome editado.

Agora vamos inserir um JTextField para receber os nomes dos materiais digitados, figura 51:

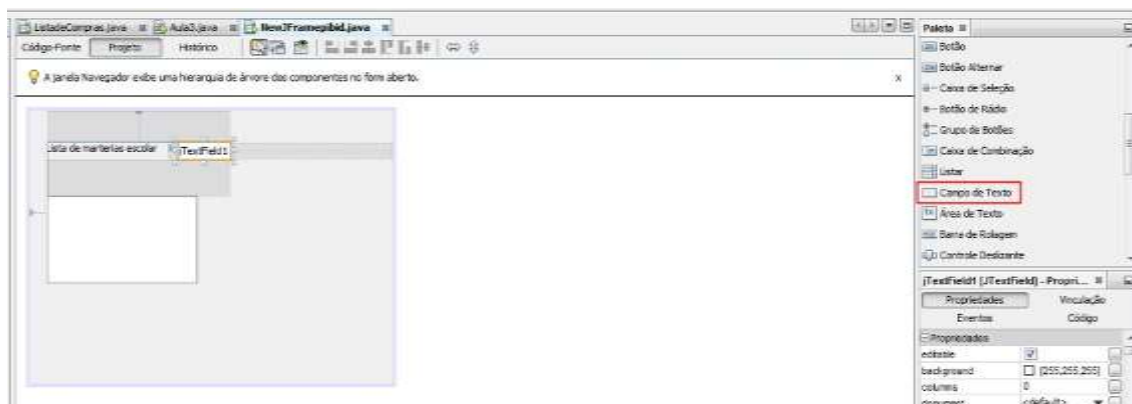


Figura 51 - Inserindo o Jtextfield.

Vamos clicar como o botão direito sobre o campo JTextField para editar texto e apertamos delete para apagar e o campo aparecem em branco como na figura abaixo, figura 52:

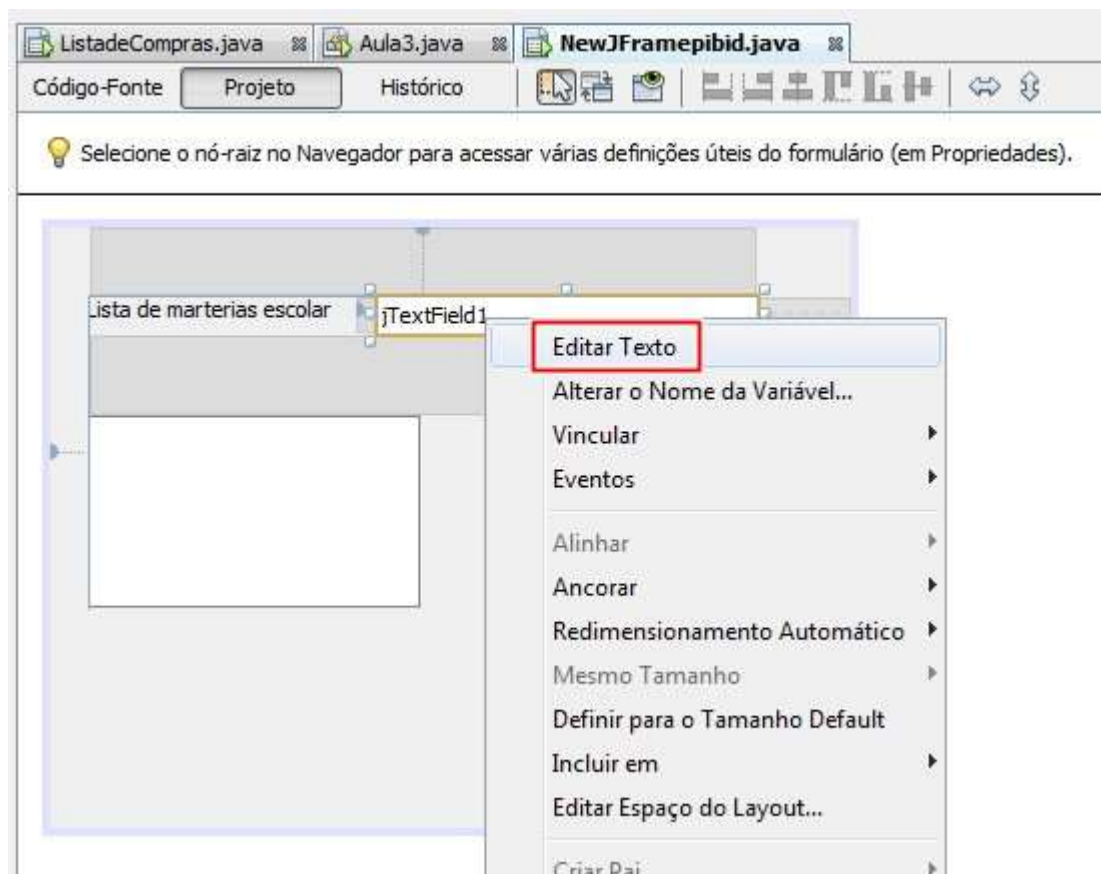


Figura 52 - Editar o jTextField.

Vamos inserir um botão jButton para recebe inserir como nas figuras a baixo, figura 53 e 54:

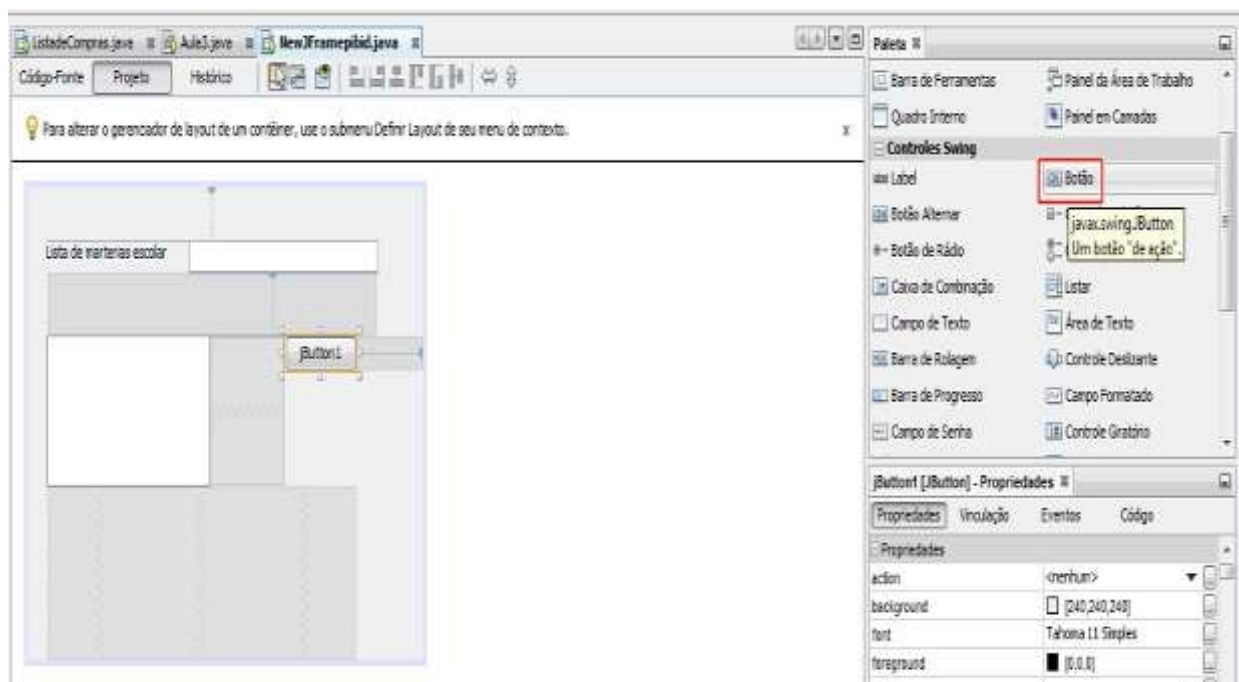


Figura 53 - Inserindo o jButton1.

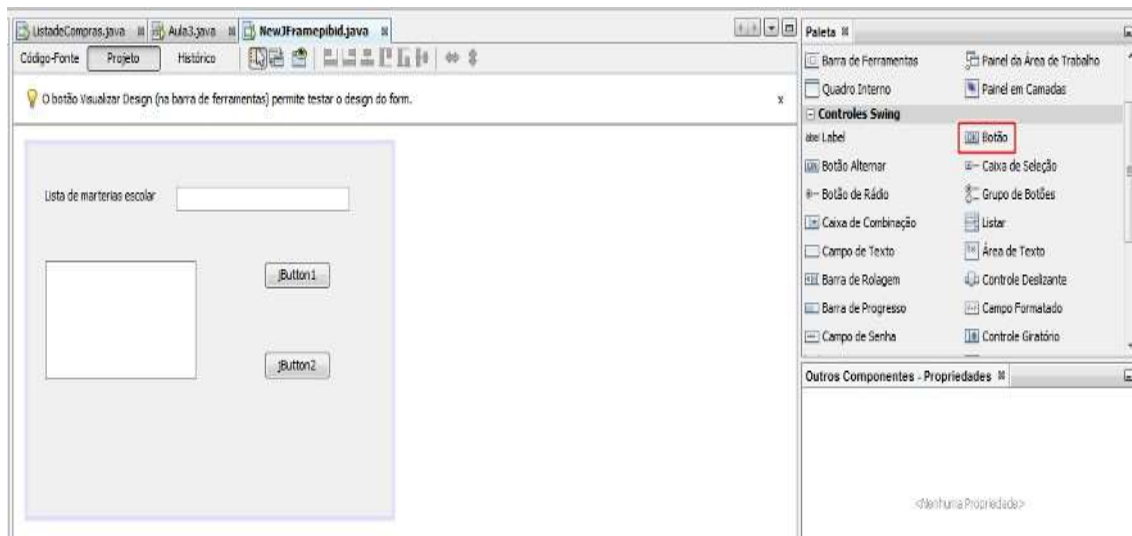


Figura 54 - Inserindo o JButton2.

Agora vamos clicar em alterar texto para colocarmos os nomes inserir e cancelar nos botões jButton1 e 2, figura 55 e 56:



Figura 55 - Colocando o nome Inserir.

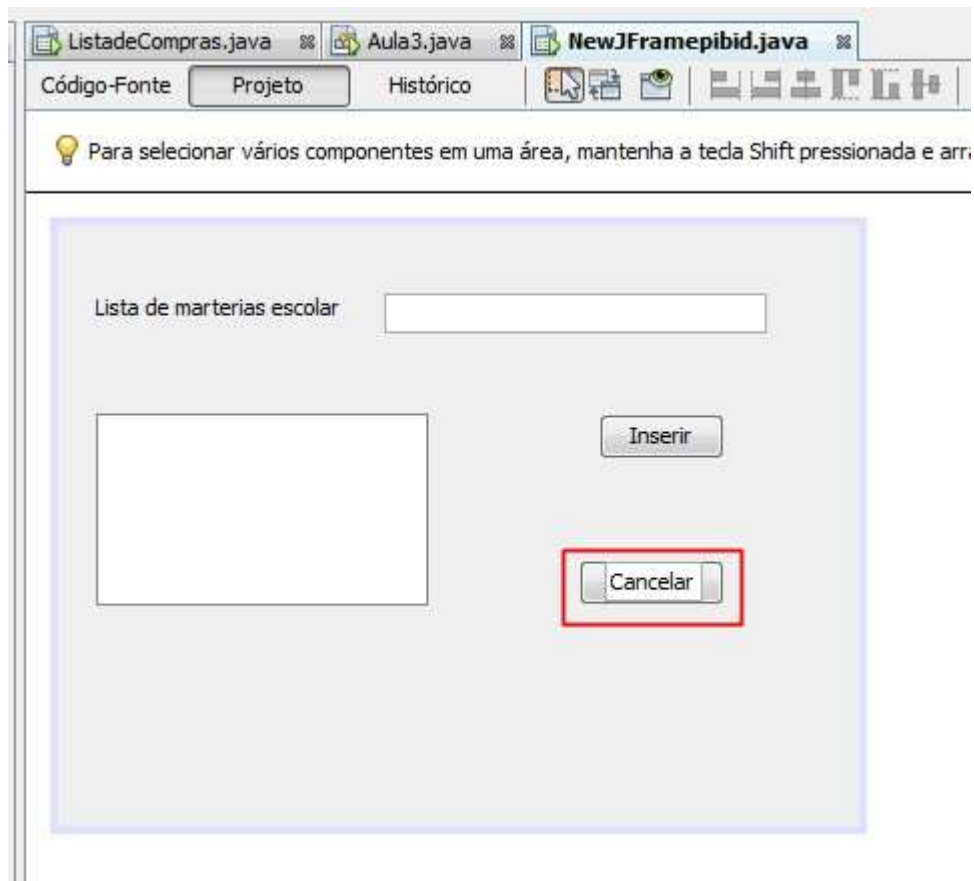


Figura 56 - Colocando o nome Cancelar no JButton2.

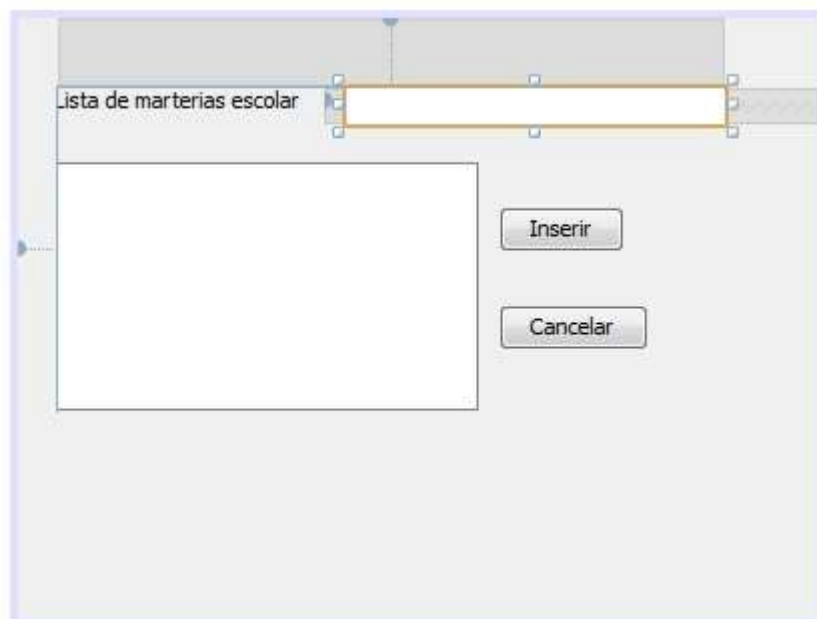


Figura 57 - O Jbutton 1 e 2 editados.

Vamos clicar em visualizar design para testar o funcionamento ver como ficará o nosso projeto, figura 58 e 59:

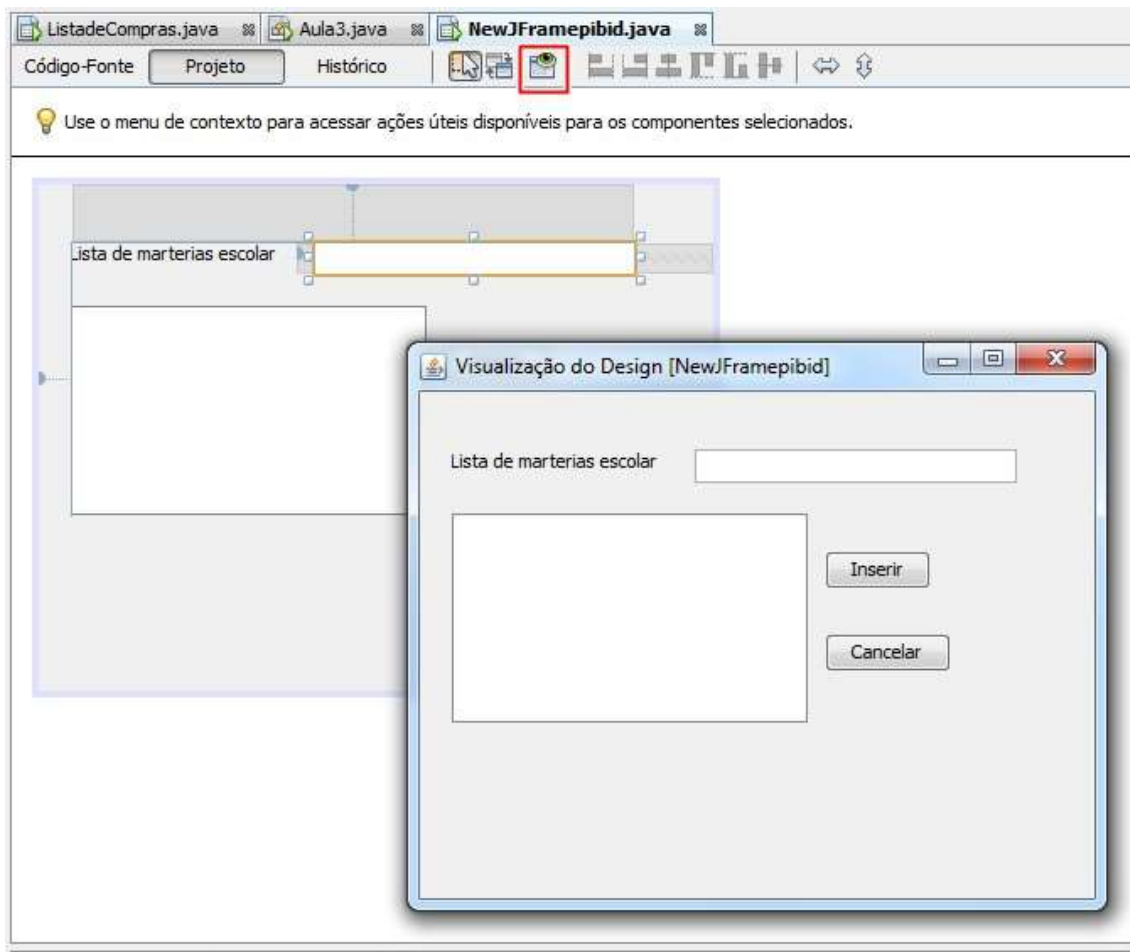


Figura 58 - Visualizando o projeto.



Figura 59 - Design do projeto.

The screenshot shows the NetBeans IDE interface. At the top, there are tabs for 'Página Inicial', 'ListadeCompras.java', 'Aula3.java', and 'NewJFramepibid.java'. Below the tabs is a toolbar with icons for 'Código-Fonte', 'Projeto', 'Histórico', and other development tools. The main window displays a Java Swing component, 'jTextArea1ListaDeMateriasEscolar [JTextArea]', with a text area containing the text 'Lista de matérias escolares'. Overlaid on this is the 'Propriedades' (Properties) window. The window has four tabs: 'Propriedades', 'Vinculação', 'Eventos', and 'Código'. The 'Propriedades' tab is selected, showing a list of properties and their values. The 'Outras Propriedades' (Other Properties) section is expanded, showing properties like 'UI', 'UIClassID', 'alignmentX', 'alignmentY', 'autoscrolls', 'baselineResizeBehavior', 'border', and 'caret'. The 'border' property is set to '[TitledBorder]'. The 'Fechar' (Close) and 'Ajuda' (Help) buttons are at the bottom right of the window.

Propriedades	Vinculação	Eventos	Código
editable	<input checked="" type="checkbox"/>		
background	<input type="checkbox"/>	[240,240,240]	
columns		20	
document		<default>	
font		Monospaced 13 Simples	
foreground	<input type="checkbox"/>	[0,0,0]	
lineWrap	<input type="checkbox"/>		
rows		5	
tabSize		8	
text			
toolTipText			
wrapStyleWord	<input type="checkbox"/>		
Outras Propriedades			
UI		<default>	
UIClassID		TextAreaUI	
alignmentX		0.5	
alignmentY		0.5	
autoscrolls	<input checked="" type="checkbox"/>		
baselineResizeBehavior		CONSTANT_ASCENT	
border		[TitledBorder]	
caret		<default>	

jTextArea1ListaDeMateriasEscolar [JTextArea]

Fechar Ajuda

Colocarmos o título e a fonte como na imagem abaixo, figura 61:

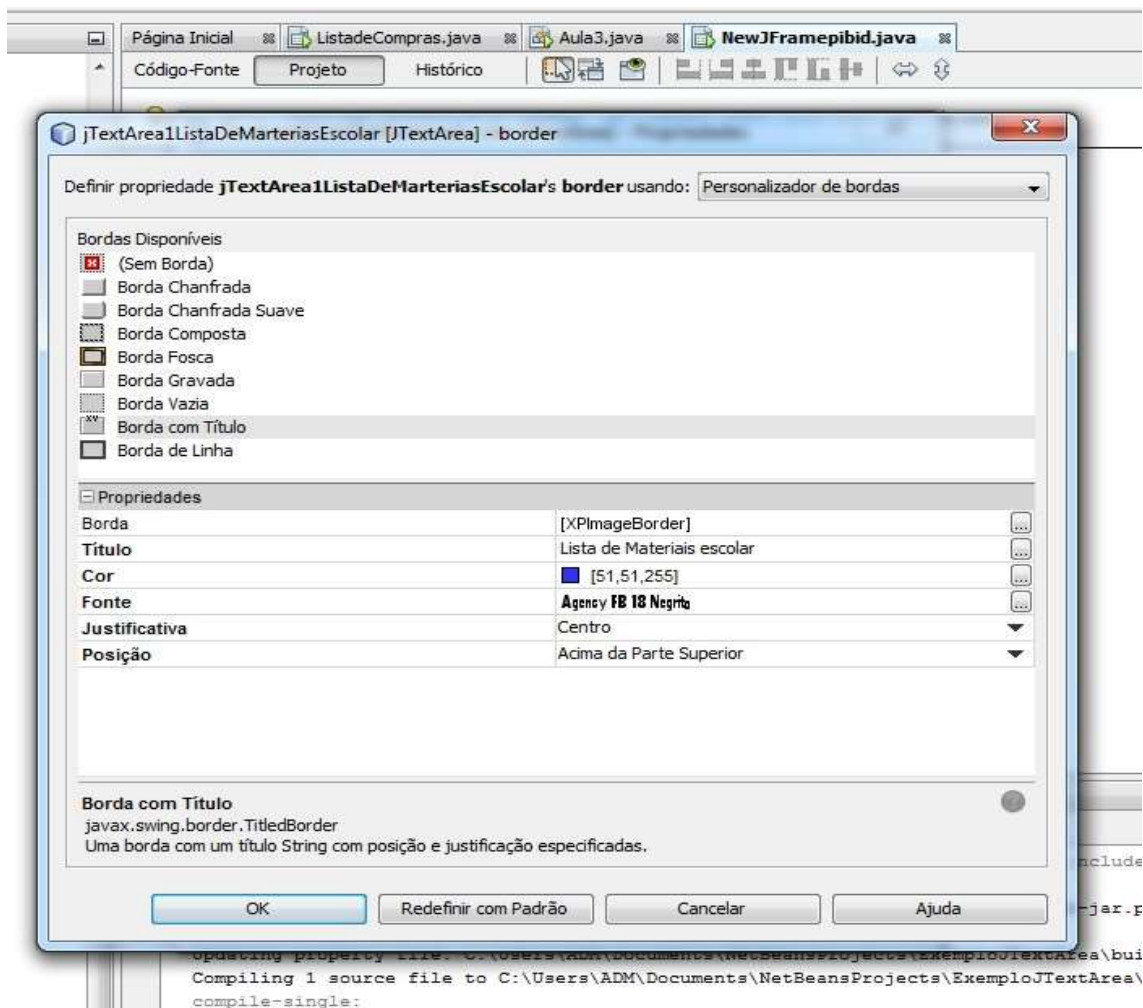


Figura 61 - Colocando título e alterando a fonte.

Para dar funcionalidade ao projeto iremos trabalhar com um evento chamado `ActionPerformed` como mostra as imagens abaixo das figuras 62 e 63:

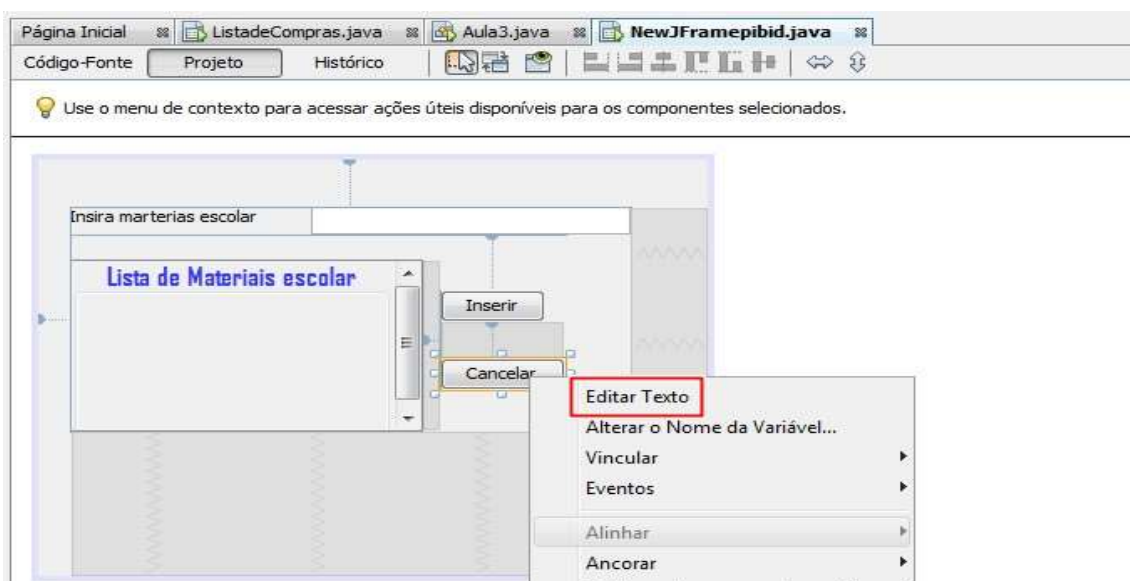


Figura 62 - Dando funcionalidade aos botões.

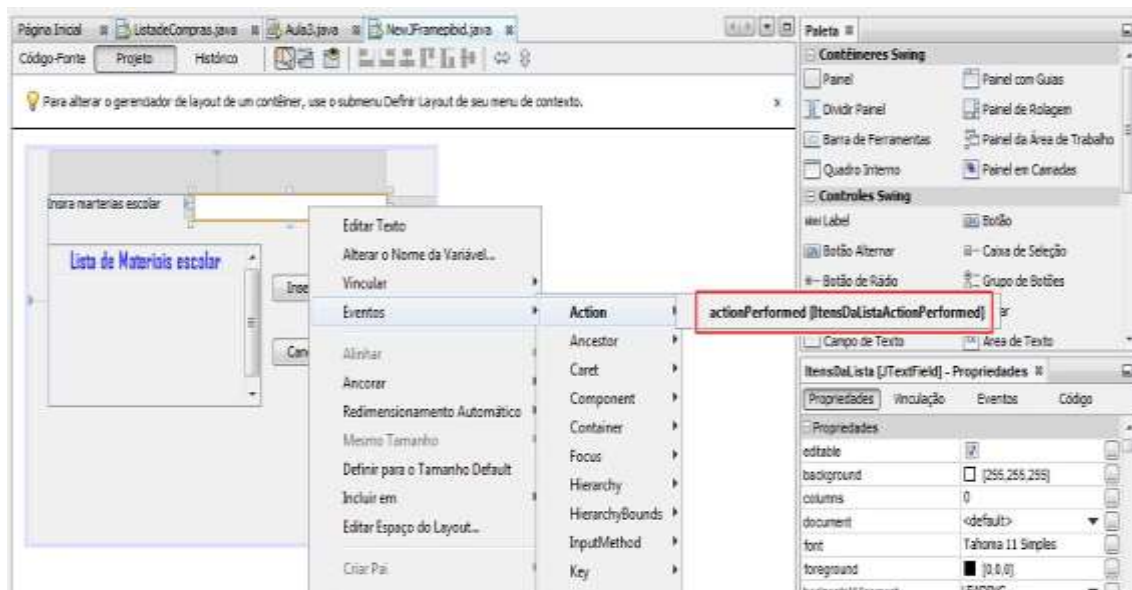


Figura 63 - Ativando actionPerformed.

Iremos declarar as variáveis no `jButton2ActionPerformed` que é o botão cancelar iremos inserir o comando `System.exit(0);` para que o botão cancelar possa funcionar corretamente, figura 64:

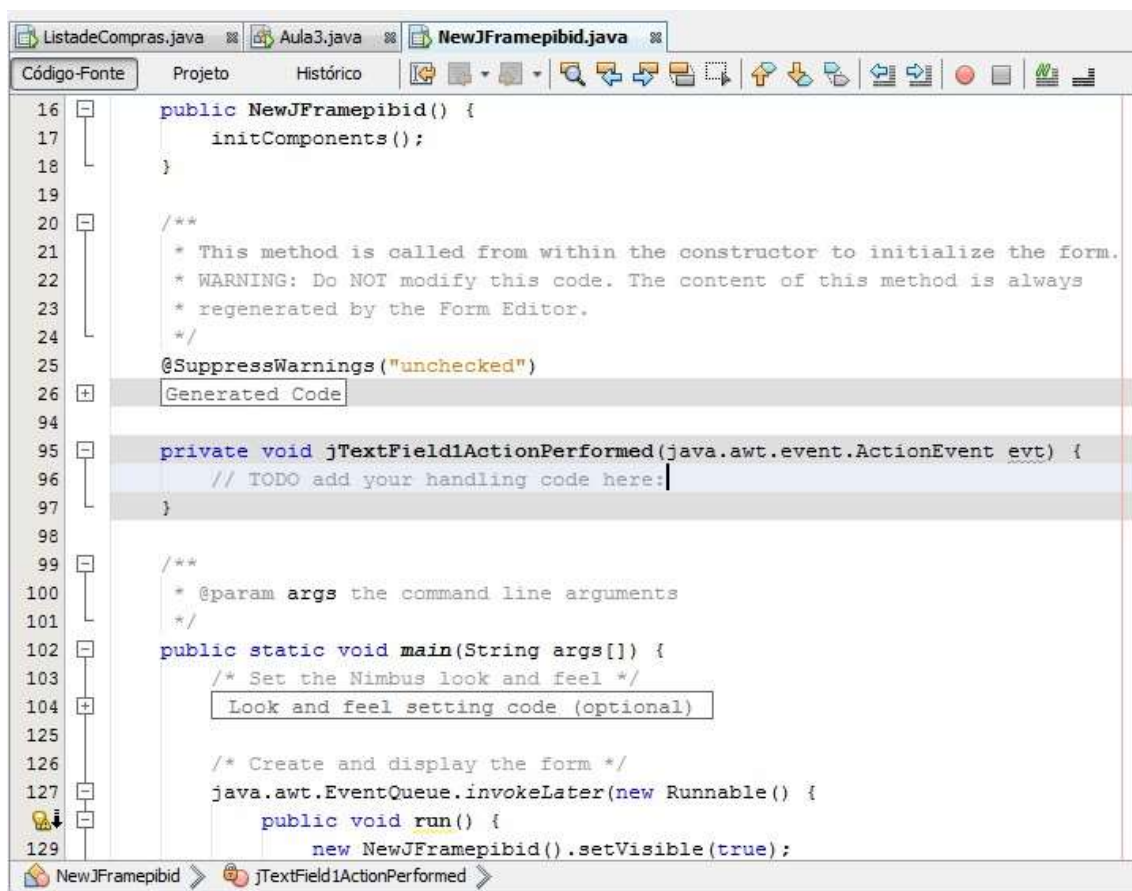


Figura 64 - Declarando as variáveis.

No `jButton1ActionPerformed` que é o botão Inserir iremos dar o:

```
String item = ItensDaLista.getText();
```

```
jTextArea1ListaDeMateriasEscolar.append(item+"\n");
```

```
ItensDaLista.setText("");
```

Conforme as figuras 65 e 66:

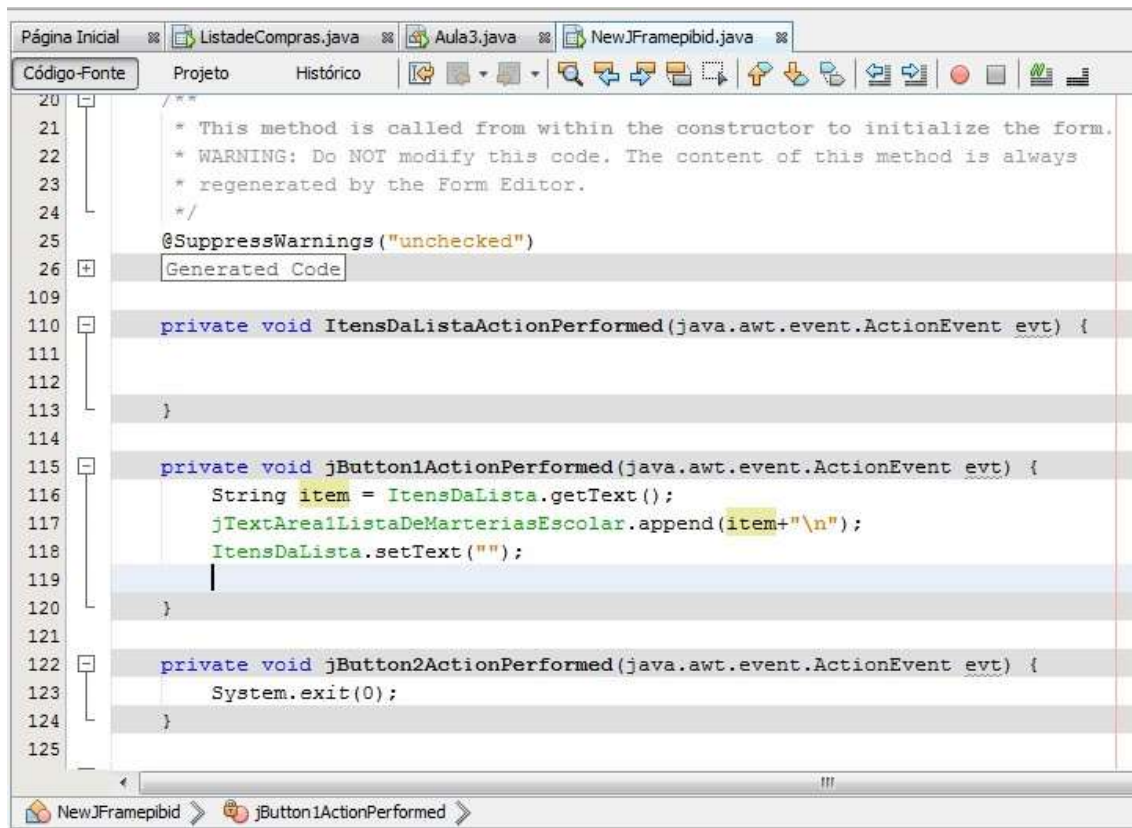


Figura 65 - Declarando variáveis.

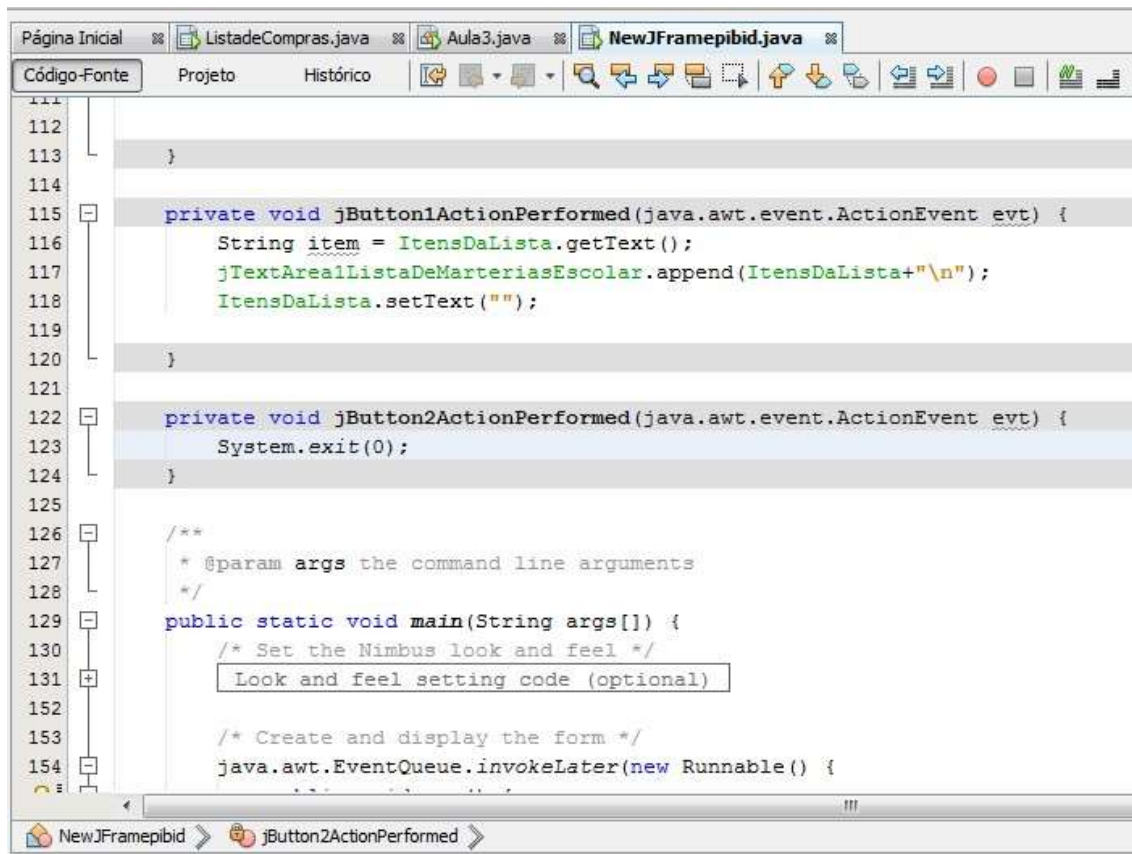


Figura 66 - Declarando variáveis.

Ao executar o projeto, com F6, o programa vai abrir a Figura 67:



Figura 67 - Design do projeto.

Agora vamos testar o nosso programa para ver se está funcionando corretamente vou digitar a palavra lápis e clicar em inserir figuras 68 e 69:



Figura 68 - Testando o botão.



Figura 69 - Botão inserir funcionando corretamente.

Agora digitaremos a palavra Borracha e clicar em inserir, figura 70:



Figura 70 - Mostra as palavras digitadas inseridas no nosso projeto.

Projeto funcionado corretamente.

jRadioButton e jButtonGroup

Botões de opção (declarados com a classe `JRadioButton`) são semelhantes a caixas de seleção no sentido de que têm dois estados —selecionado e não selecionado.

Os botões de opção normalmente aparecem como um grupo em que apenas um botão pode ser selecionado por vez, selecionar um botão de opção diferente força a remoção da seleção de todos os outros que estão selecionados, Utilizados para representar opções mutuamente exclusivas.

A relação lógica entre os botões de opção é mantida por um objeto `ButtonGroup` (pacote `javax.swing`), que organiza um grupo de botões e não é exibido em uma interface com o usuário, os botões de estado `JRadioButton` têm valores ativados/desativados ou verdadeiro/falso. No exemplo a seguir, um grupo de botões de rádio começa com todos os botões desmarcados, mas depois de um ser selecionado o programa irá mostrar os valores e a opção selecionada.

Para criar um novo projeto no Netbeans clique na aba **Arquivo**, **Novo Projeto**, Figura 71 - Criando projeto.

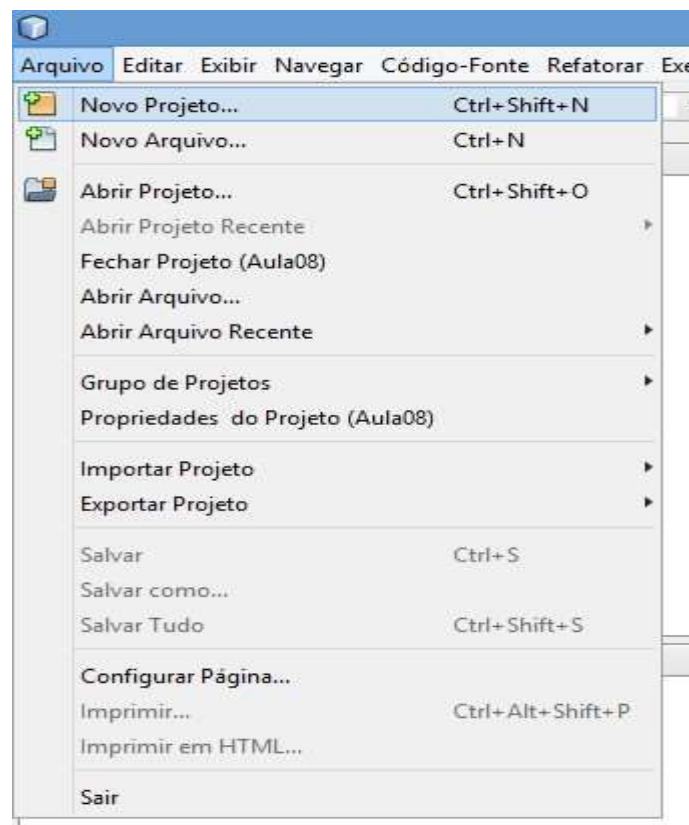


Figura 71- Criando projeto.

Em seguida aparecerá à tela para escolha do tipo de projeto, Figura 72 - Escolha de Projeto, escolha **Categoria, Java** e **Projetos, Aplicação Java** para desenvolver sistemas para Desktop.

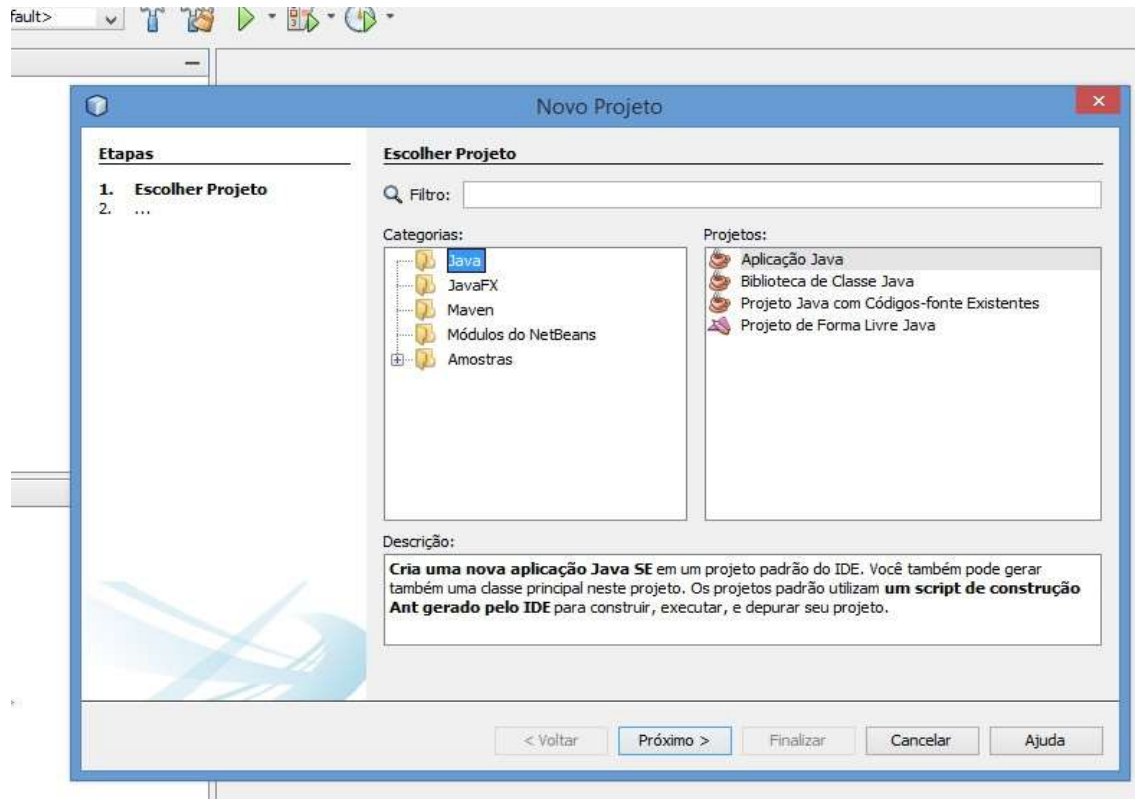


Figura 72 - Escolha de Projeto.

Próximo passo é definir o nome **ExemploRadioButton** para o projeto, Figura 73 - Atribuir nome ao projeto, neste caso devemos colocar um nome sem acentuação e espaços em branco, normalmente a primeira letra do projeto é maiúscula.

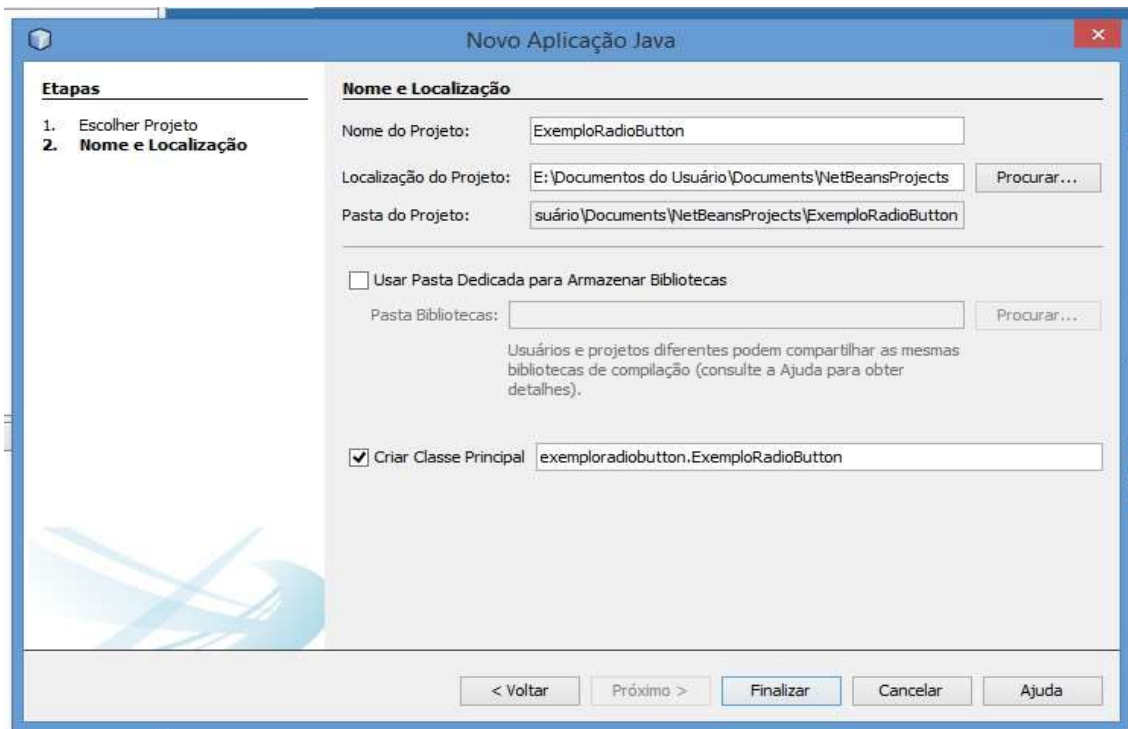


Figura 73 - Atribuir nome ao projeto.

Criar uma nova janela (JFrame): Clique com botão direito em cima da xícara (Projeto criado) e selecione **Novo, FormJFrame**, Figura 74 - Criação de Janelas:

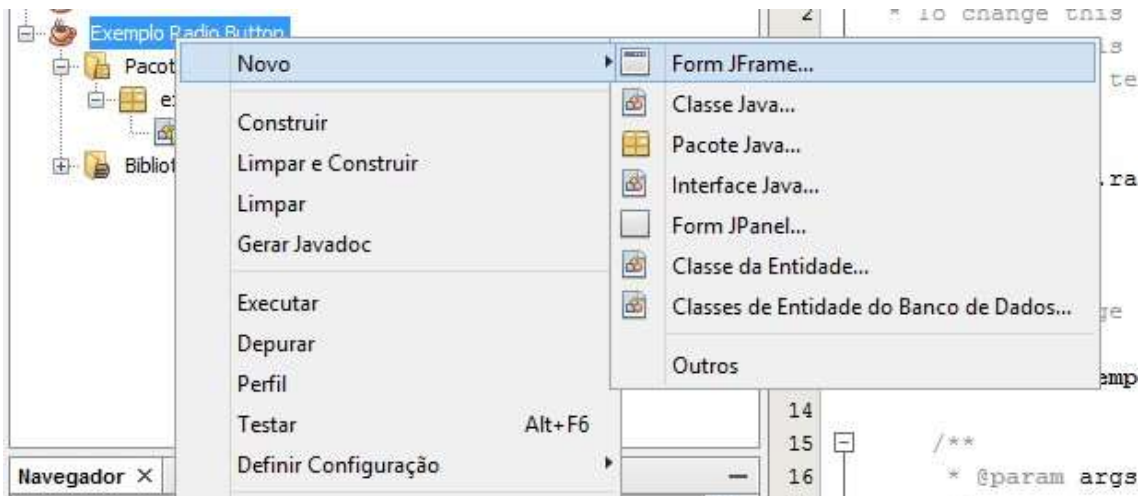


Figura 74 - Criação de Janelas.

Agora vamos definir um nome para a janela que será **radioButton**, no campo pacote, que não pode ficar vazio, temos que colocar esses JFrames dentro de um pacote, no caso vamos escrever **telas** que será nossa pasta que irá possuir

todos os JFrames/Janelas do sistema, Figura 75 - Definindo o nome do JFrame:

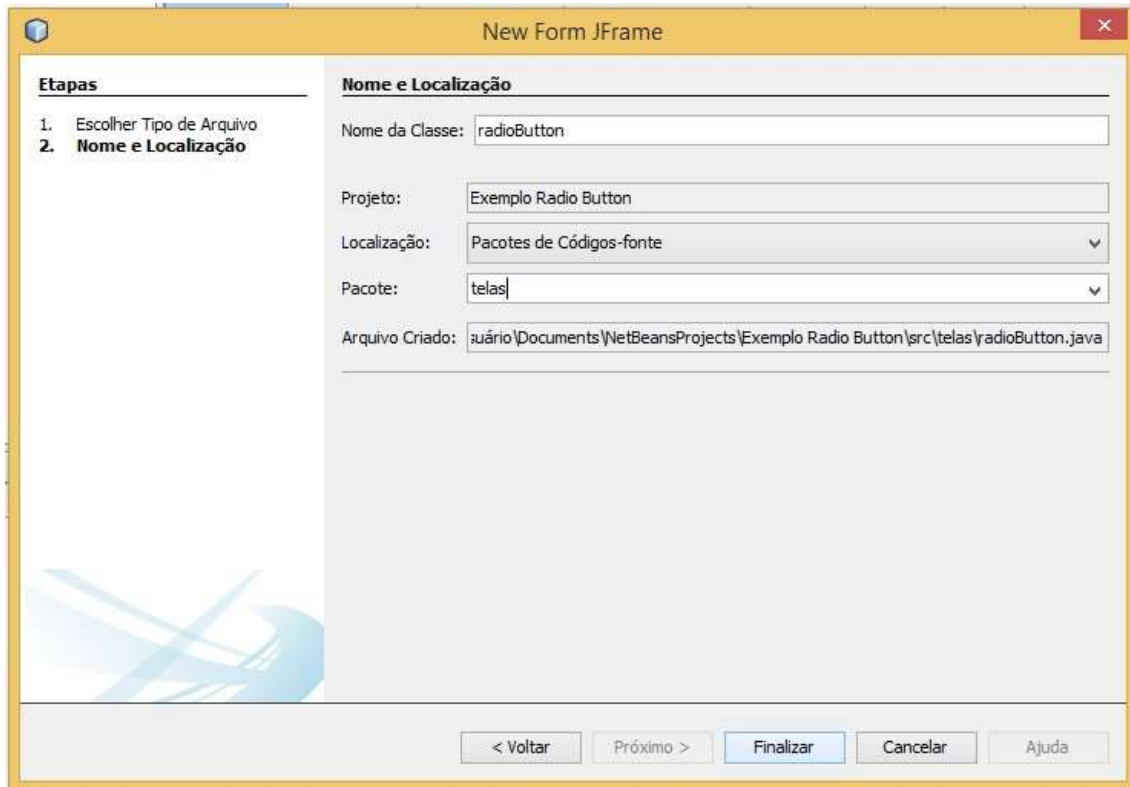


Figura 75 - Definindo o nome do JFrame.

Com o componente selecionado dá para alterar aparência, texto e tamanho dele, abaixo segue uns dos principais atributos que podemos modificar:

Background: cor de fundo;

Font: Tipo de letra e tamanho;

Foreground: Cor da letra;

Text: O texto que vai aparecer na tela.

Após a criação do JFrame vamos modificar algumas das suas propriedades, para melhorar seu visual. Clique com o botão direito do mouse sobre o JFrame, e clique em **propriedades**, Figura 46 - Acessando as propriedades do JFrame:

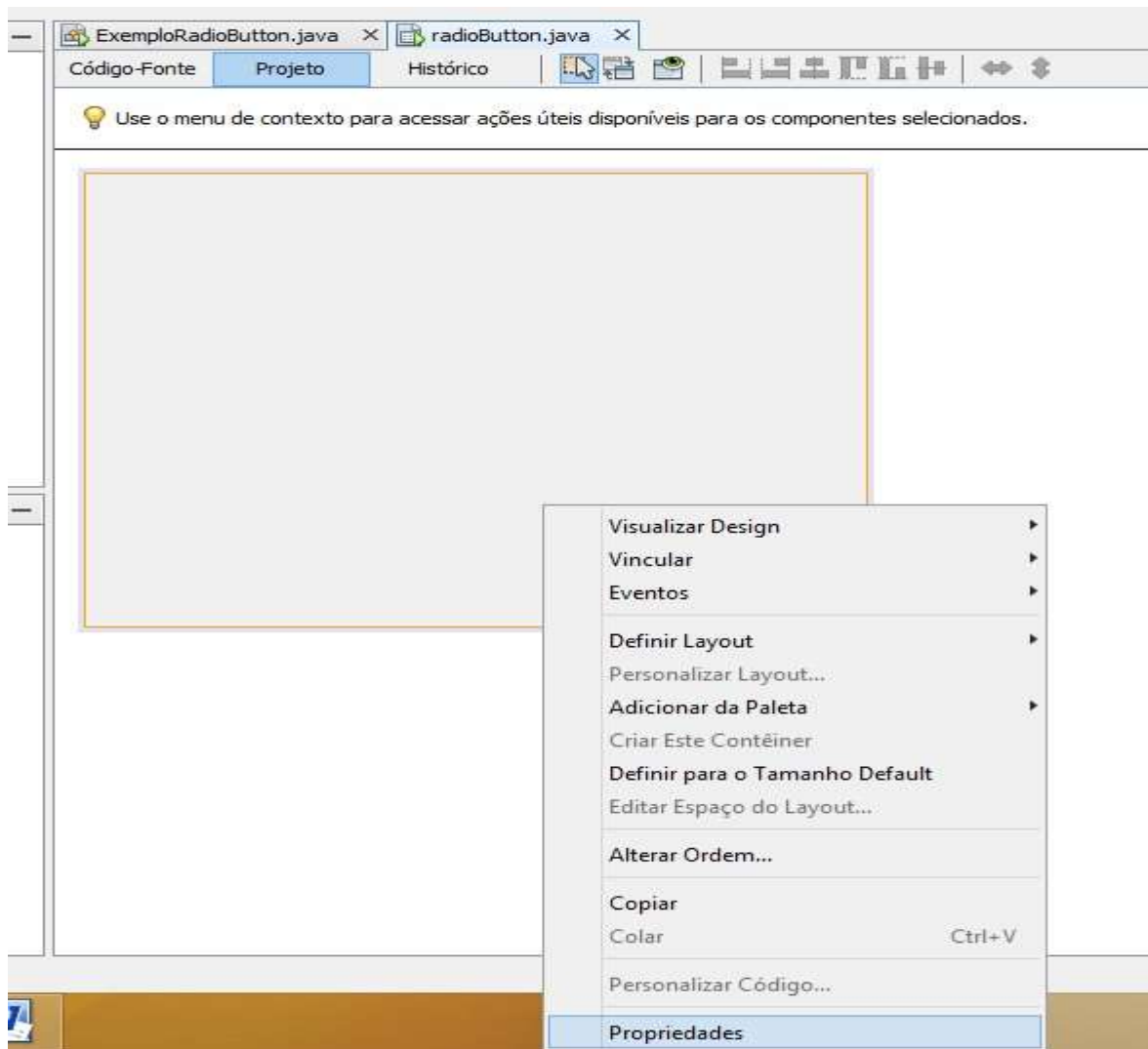


Figura 76 - Acessando as propriedades do JFrame.

Vamos primeiro colocar a opção **defaultCloseOperation** como **DISPOSE**, para permitir o fechamento do formulário(Form), Figura 77 - Opção **defaultCloseOperationDISPOSE**:



Figura 77 - Opção defaultCloseOperation DISPOSE.

Modificar o **title**(título) do nosso formulário para: **Exemplo de uso Radio Button**, Figura 78 - Modificando o title:



Figura 78 - Modificando o title.

Em seguida **desmarcamos** a opção **resizable**, para impedir que o usuário redimensione a tela do programa, ou seja, maximize ou minimize. Figura 79 - Resizable:

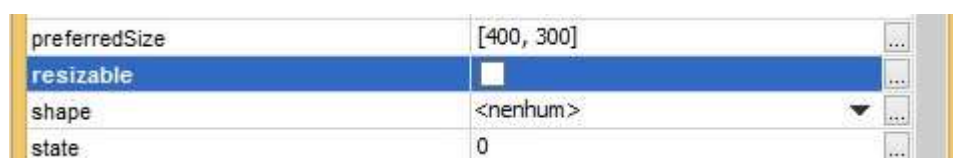


Figura 79 - Resizable.

Agora na aba **código** vamos selecionar a opção **Gerar Centralizado**, para que o programa sempre que for executado inicie no centro do monitor. Figura 80 - Gerar Centralizado:



Figura 80 - Gerar Centralizado.

Depois disso você pode ir à opção **Visualizar Design** do Netbeans (destacado em vermelho na Figura 81 - Visualizar Design), para ver como está o visual do seu programa, através dele você pode visualizar uma simulação da sua aplicação rodando, contudo essa opção se restringe a visualização do layout do seu programa já que as suas funcionalidades não poderão ser executados.

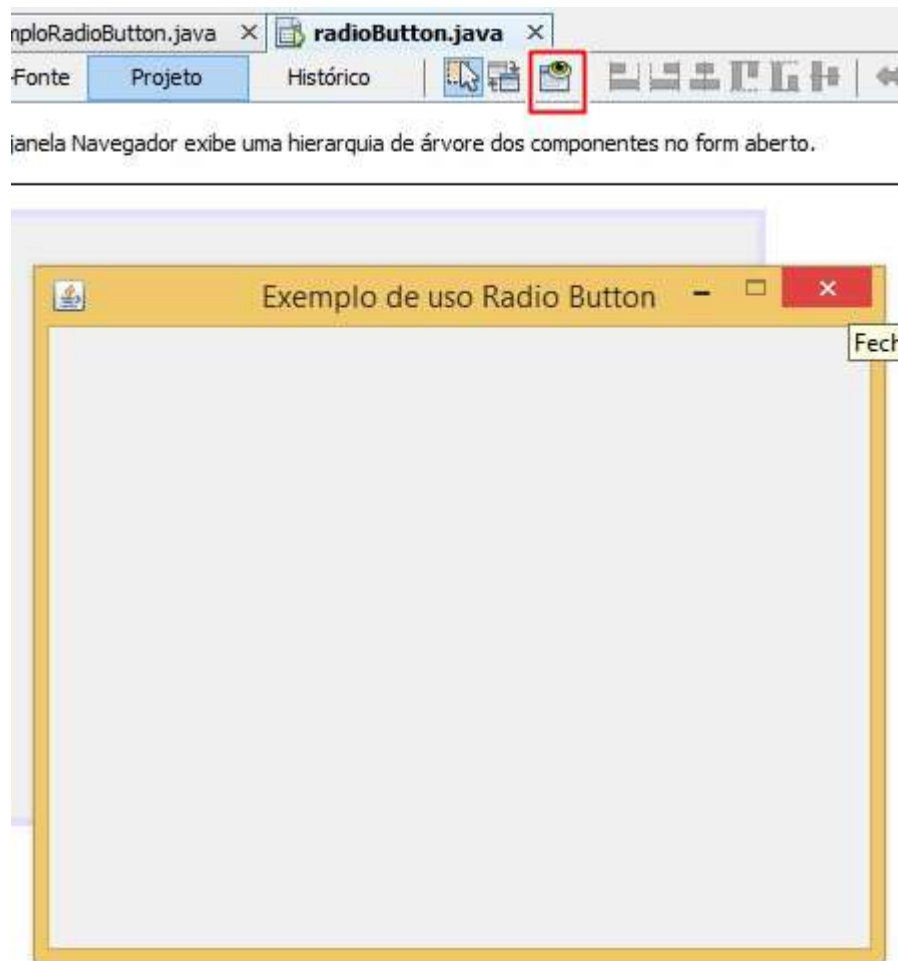


Figura 81 - Visualizar Design.

Agora vamos configurar alguns itens, você irá à **paleta**, canto direito do NetBeans, Figura 82 - Opções da paleta, e selecionaria na opção **ContêineresSwing** um **Painel**, pressionando o botão esquerdo do mouse, até o JFrame, depois irá redimensioná-lo como na Figura 83 - Usando o Painel:



Figura 82 - Opções da paleta

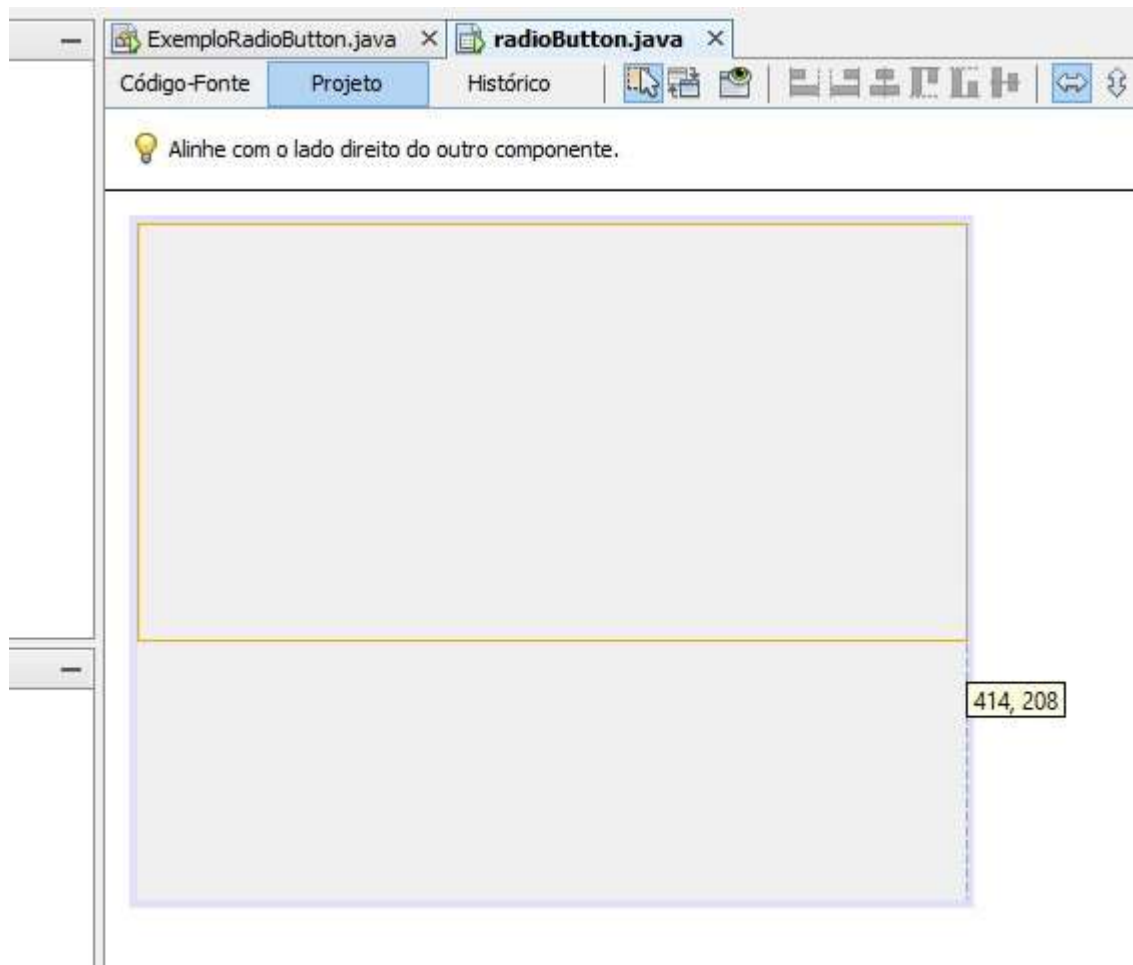


Figura 83 - Usando o Painel.

Agora para melhorar o agrupamento e o visual do nosso aplicativo vamos adicionar outro Painel no JFrame, só desta vez ele irá ficar na parte inferior ,
Figura 84 - Segundo Painel:

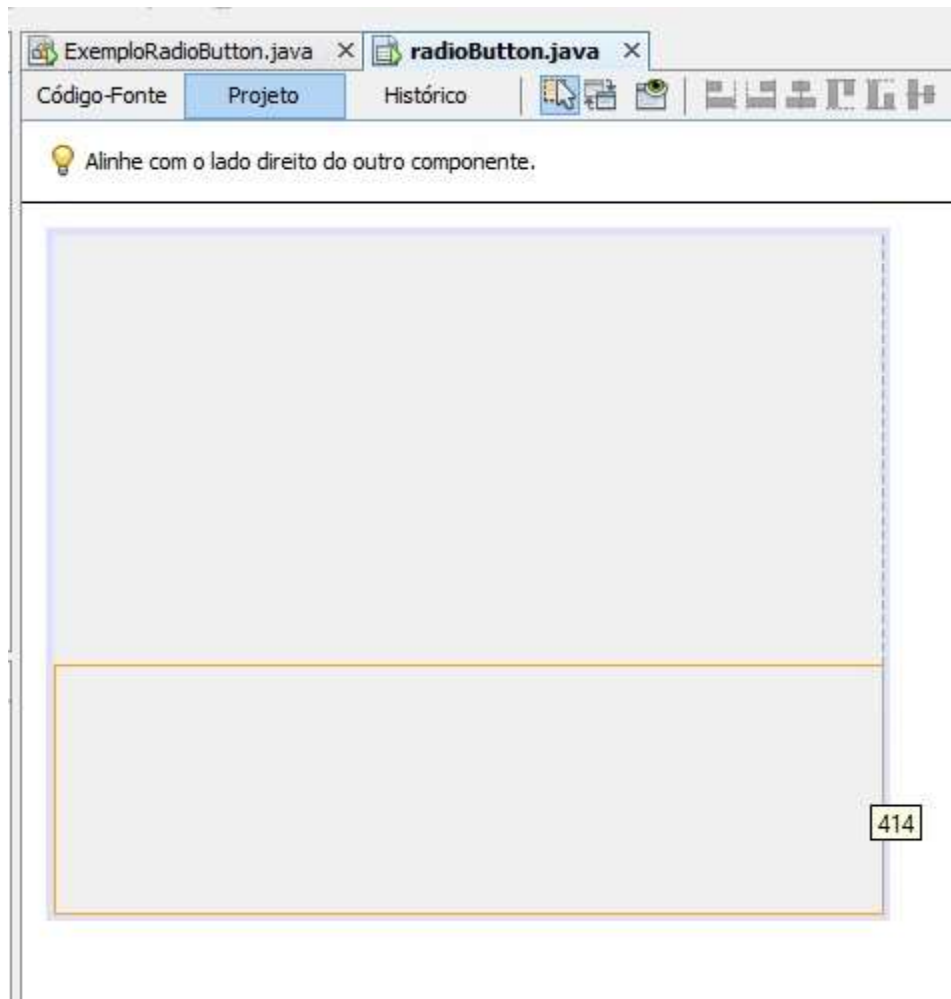


Figura 84 - Segundo Painel.

Vamos agora adicionar algumas opções para mudar a aparência do aplicativo, vamos selecionar a painel superior, e clicar com o botão direito do mouse para acessar as **propriedades** do painel, e então iremos clicar nas reticências (destacadas em vermelho), Figura 85 - Bordas do Painel:

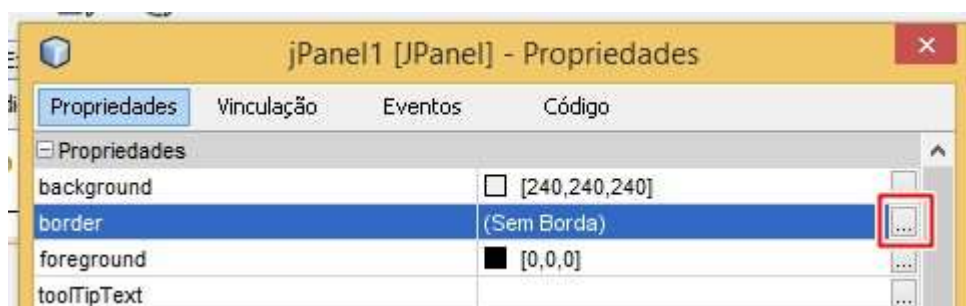


Figura 85 - Bordas do Painel.

No menu borda você irá selecionar a opção **BordacomTítulo**, Figura 86 - Borda com Título:

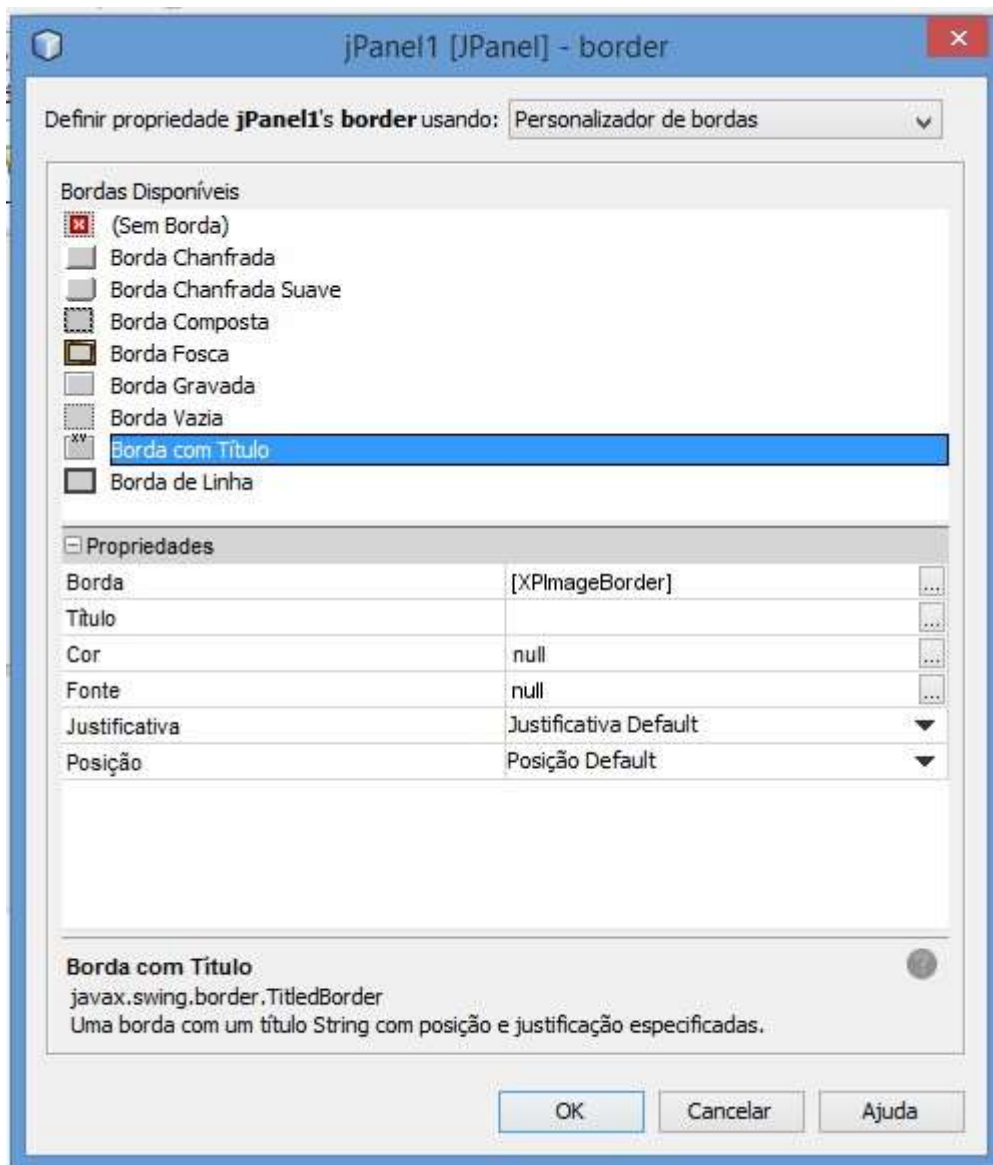


Figura 86 - Borda com Título.

Agora vamos por um título nessa borda, digitando na opção **Título** a frase: “Escolha uma opção de envio:” como na Figura 87 - Título da borda superior:



Figura 87 - Título da borda superior.

Na opção **Cor** iremos selecionar Branco, mas você pode experimentar outras cores, Figura 88 - Cor da borda:

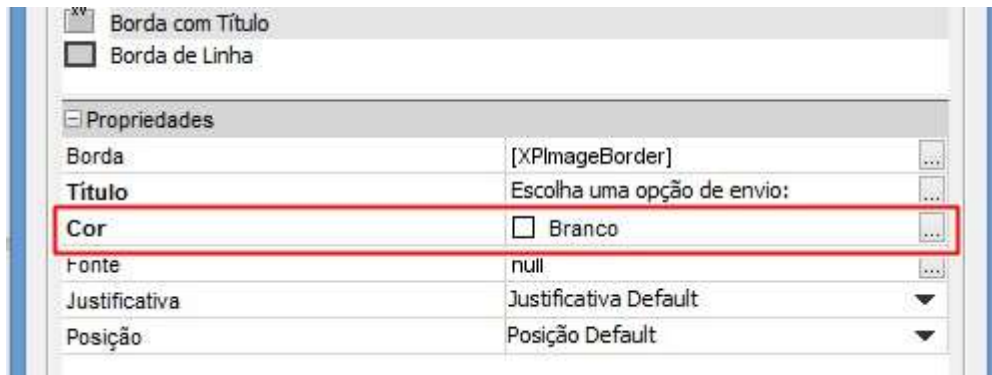


Figura 88 - Cor da borda.

Na opção **Fonte** você irá clicar nas reticências (destacada em vermelho), Figura 89 - Fonte da borda:



Figura 89- Fonte da borda.

Você pode selecionar as opções que quiser no menu Borda com Título - Fonte, mas no nosso exemplo selecionamos Tamanho: 14, Estilo da Fonte: Negrito e Fonte: Aharoni. Figura 90 - Opções fonte da borda:



Figura 90 - Opções fonte da borda.

E a **Justificativa** será a opção **Centro**, Figura 91 - Justificativa da borda:

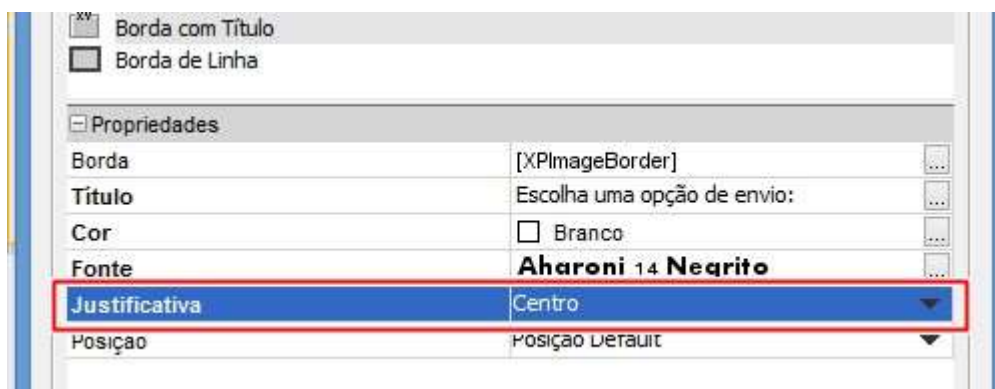


Figura 91 - Justificativa da borda.

Após esta primeira parte iremos pôr uma borda dentro da borda selecionada, você irá clicar nas reticências da opção **Borda** (destacada em vermelho), Figura 92 - Borda sobre borda:

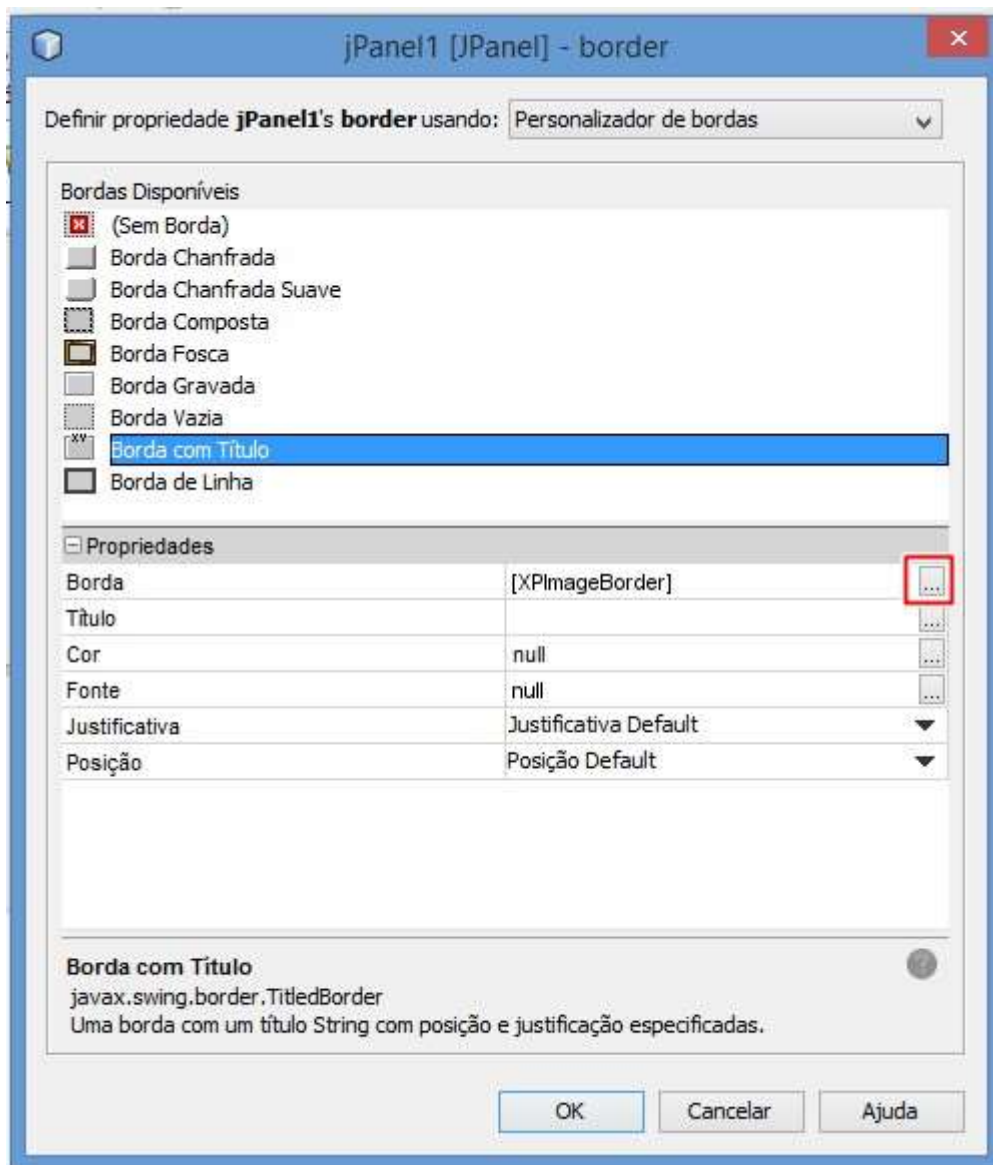


Figura 92 - Borda sobre borda.

Após isso aparecerá uma nova tela como a Figura 93 - Borda com Título - Borda, nela você pode mudar algumas características dessa nova borda. Neste exemplo iremos deixar a opção **CantosArredondados**, selecionada e mude a **Cor** clicando nas reticências, na nova tela você irá clicar na aba **PaletaAWT** (destaque em vermelho) e selecionar a cor branca, como na Figura 94 - Paleta AWT:

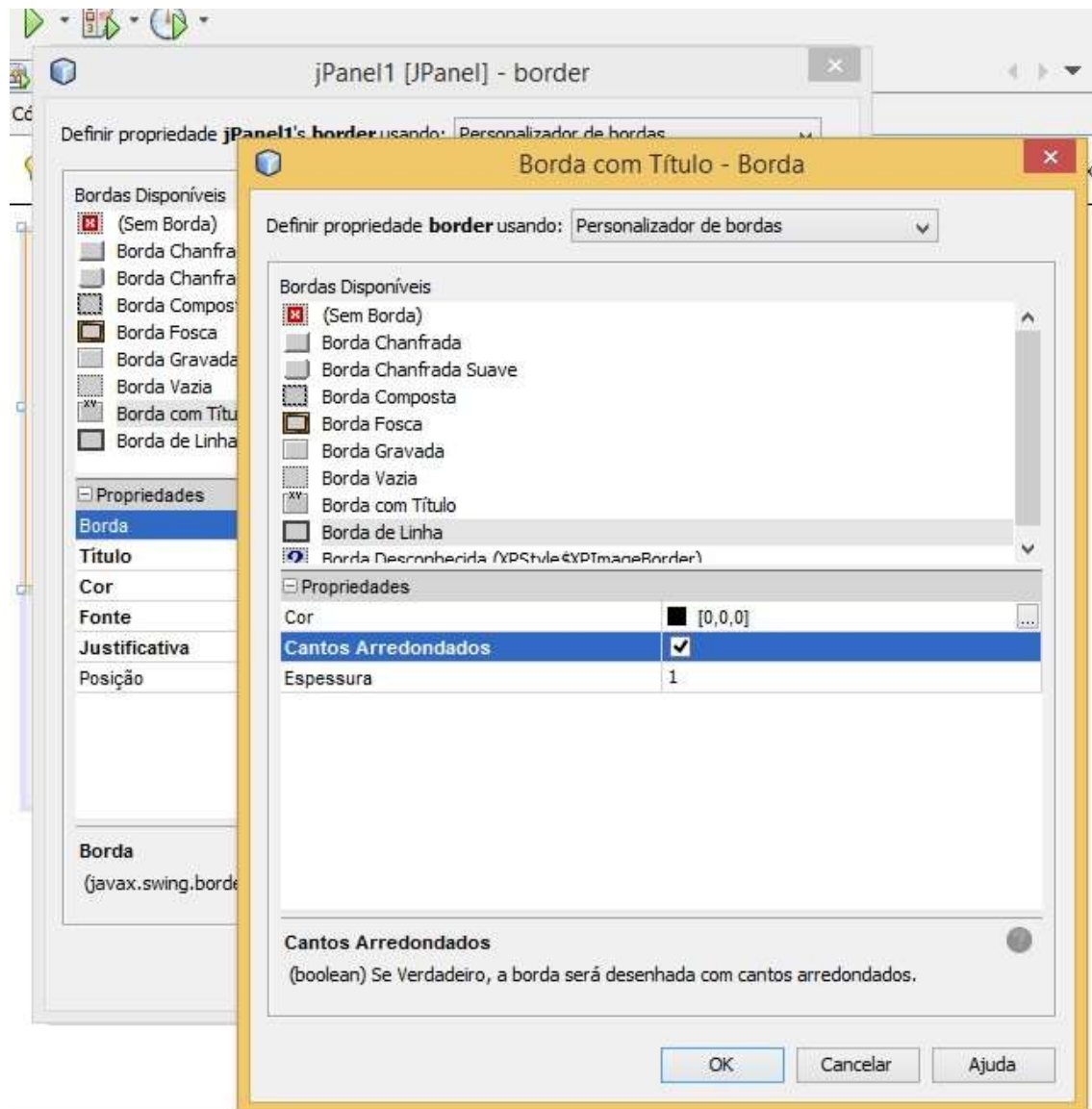


Figura 93 - Borda com Título - Borda.

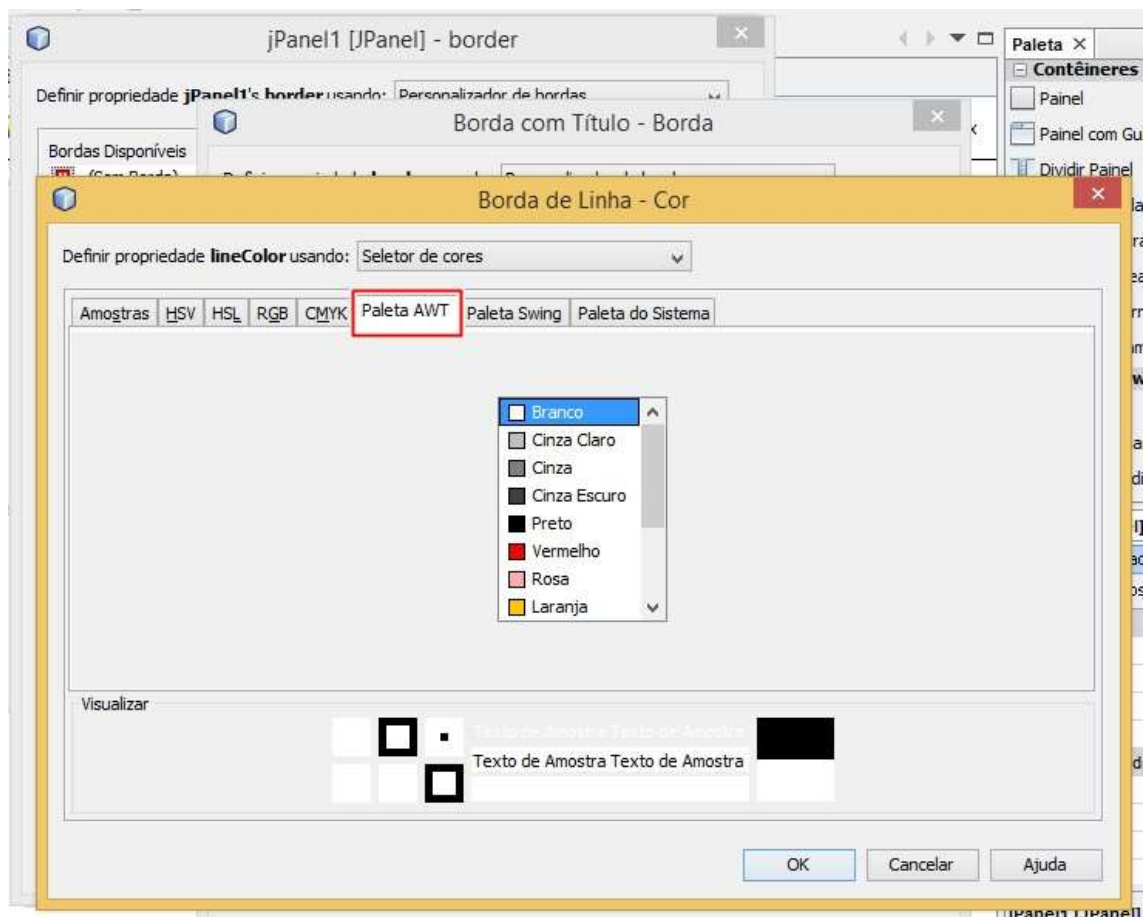


Figura 94 - Paleta AWT.

Agora vamos escolher uma cor de fundo (**Background**) para o label superior, na qual o Branco realce. Você voltará para as propriedades do painel, Figura 95 - Propriedades do Painel:

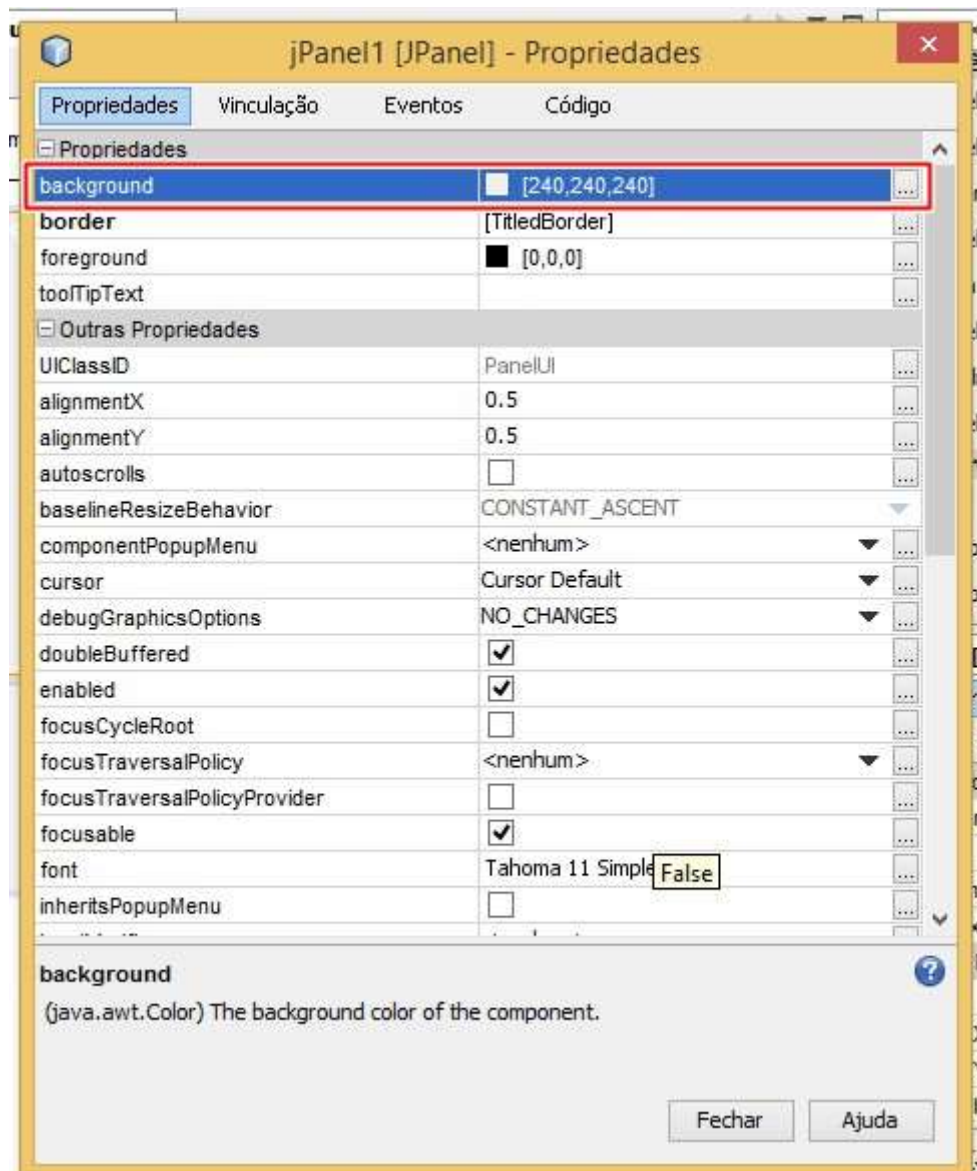


Figura 95 - Propriedades do Painei.

Na **Paleta AWT** você irá selecionar a opção cinza escuro, Figura 96 - Background Painei:

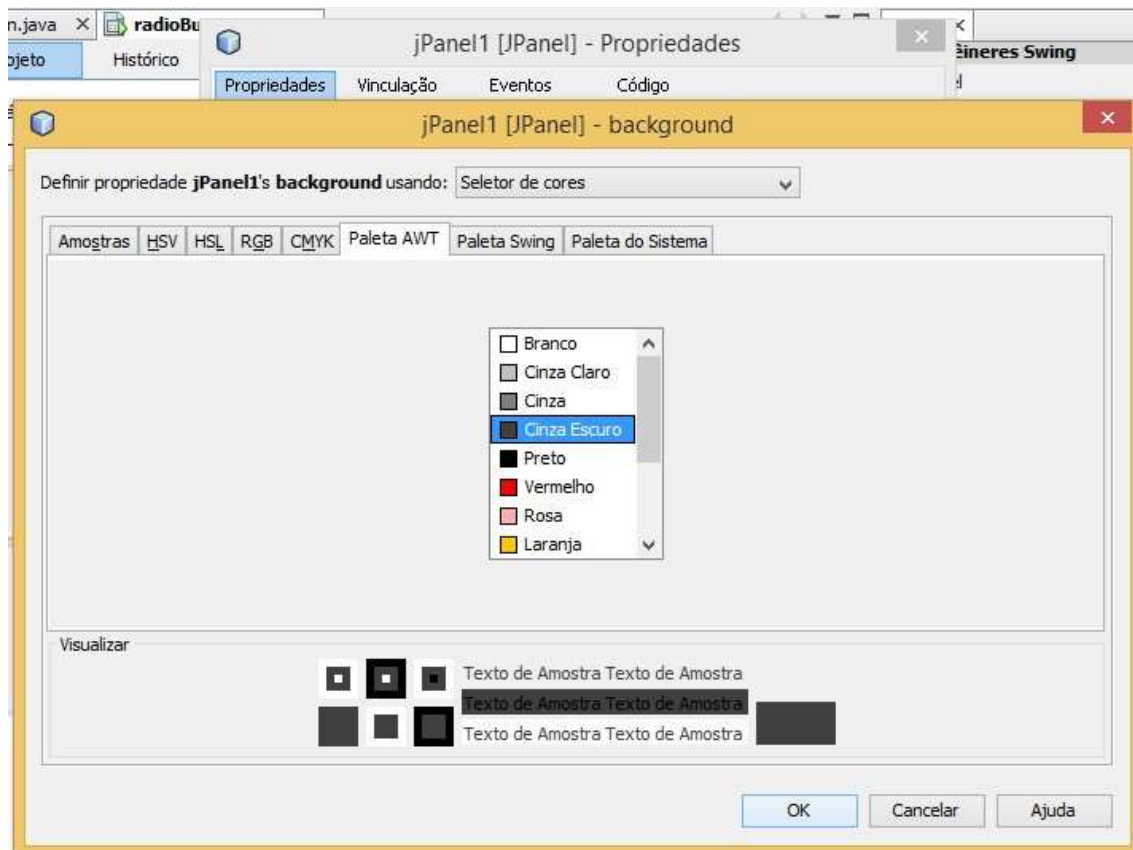


Figura 96 - Background Panel.

Em seguida vamos modificar o painel inferior e iremos escolher a cor de fundo (Background) cinza claro, Figura 27 - Background Painel Inferior.



Figura 97 - Background Painel Inferior.

Agora vamos adicionar os componentes **JRadioButton** no nosso painel. Primeiro vamos visualizar na paleta a opção **Controle Swing** (destaque em vermelho) e nela selecionar o **Botão de Rádio** (destaque em preto) e arrastar até o painel superior, você irá arrastar até o painel três botões de rádio que iram compor as opções de envio que teremos no nosso programa, como na Figura 98 - Botão de Rádio:

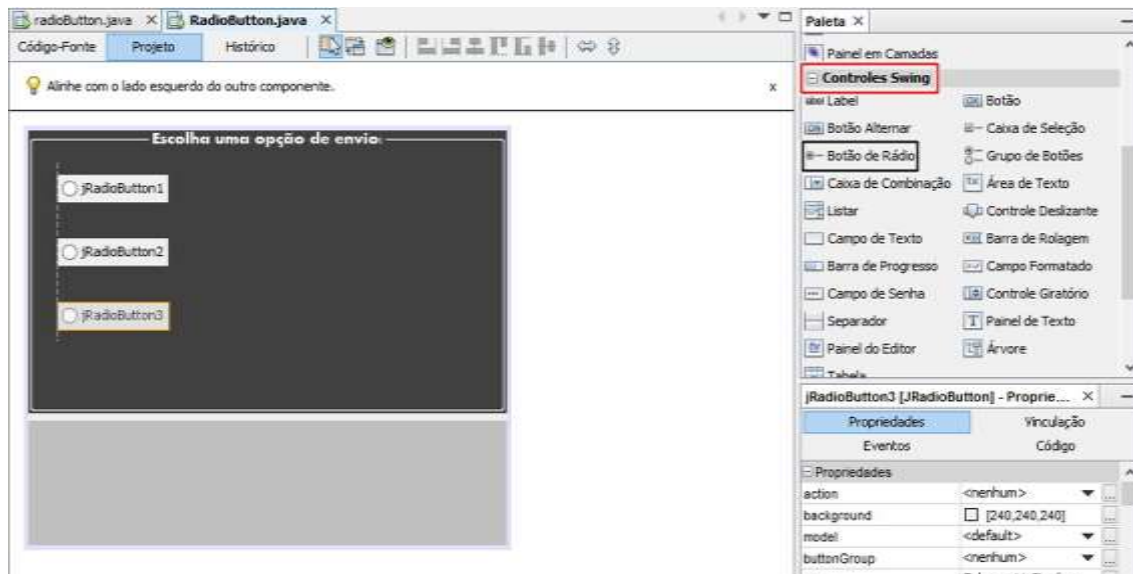


Figura 98 - Botão de Rádio.

Após isso precisamos mudar o nome dos `JRadioButton`, você irá selecionar um `JButton` e clicar com o botão direito do mouse, quando abrir as opções você irá clicar em **Editar Texto**, Figura 99 - Opções `JRadioButton`. E irá editar os textos de todos os três `JRadioButton`, que serão as opções de envio que teremos no nosso programa: PAC, Encomenda Normal e SEDEX, como na Figura 100 - Opções de Envio:

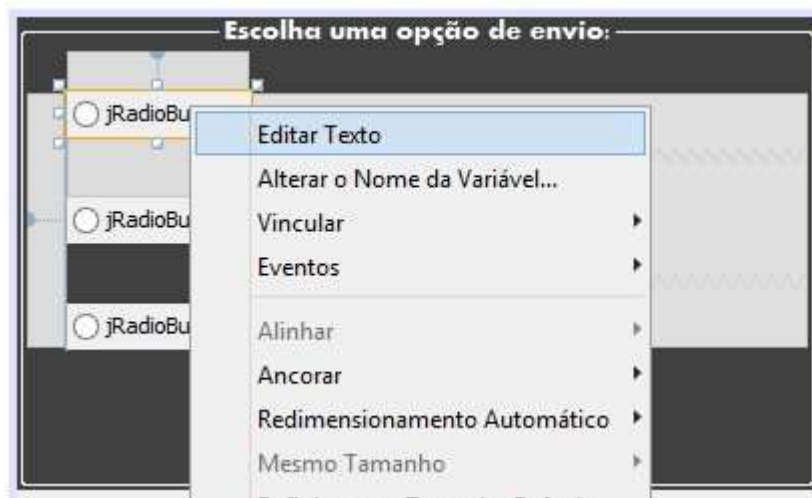


Figura 99 - Opções `JButton` .

Escolha uma opção de envio:

☐ PAC

☐ Encomenda Normal

☐ SEDEX

Figura 100 - Opções de Envio.

Agora vamos editar as propriedades desses botões como fizemos com os painéis, como a propriedade que pretendemos mudar é a mesma para os três botões podemos selecionar os três clicar com o botão direito e acessar as propriedades, desta forma as mudanças feitas serão vista no três botões selecionados, Figura 101 - Propriedades jButton:

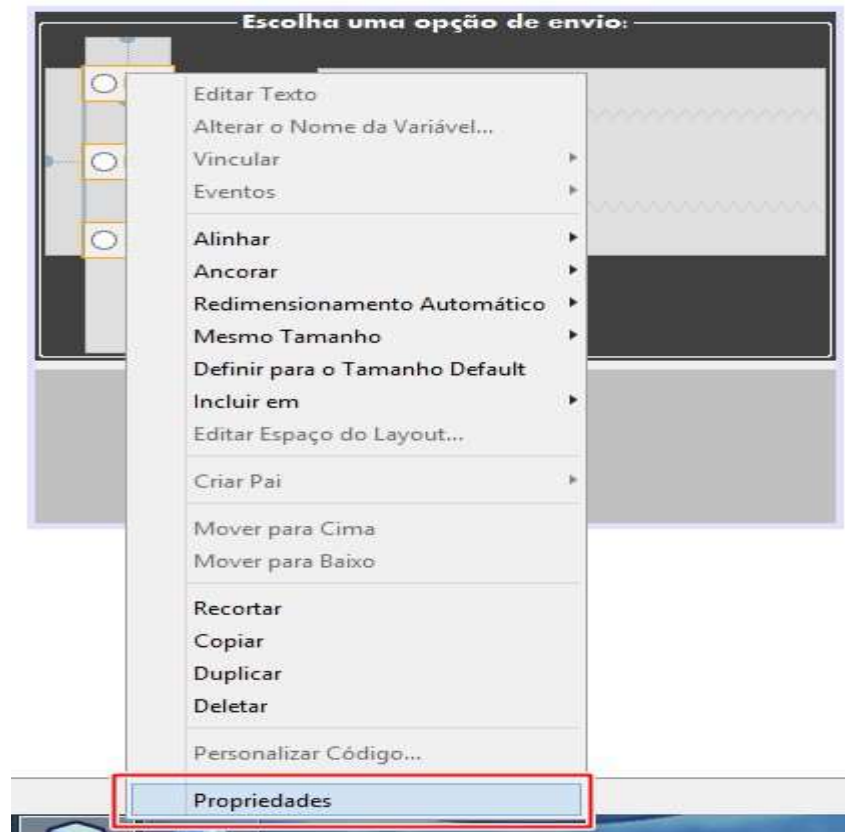


Figura 101 - Propriedades jRadioButton.

Neste caso vamos somente editar a fonte dos textos dos botões para negrito, clique nas reticências na opção **font**, Figura 102 - Font jRadioButton. E escolha o estilo de fonte: negrito, Figura 103 - Negrito jRadioButton:

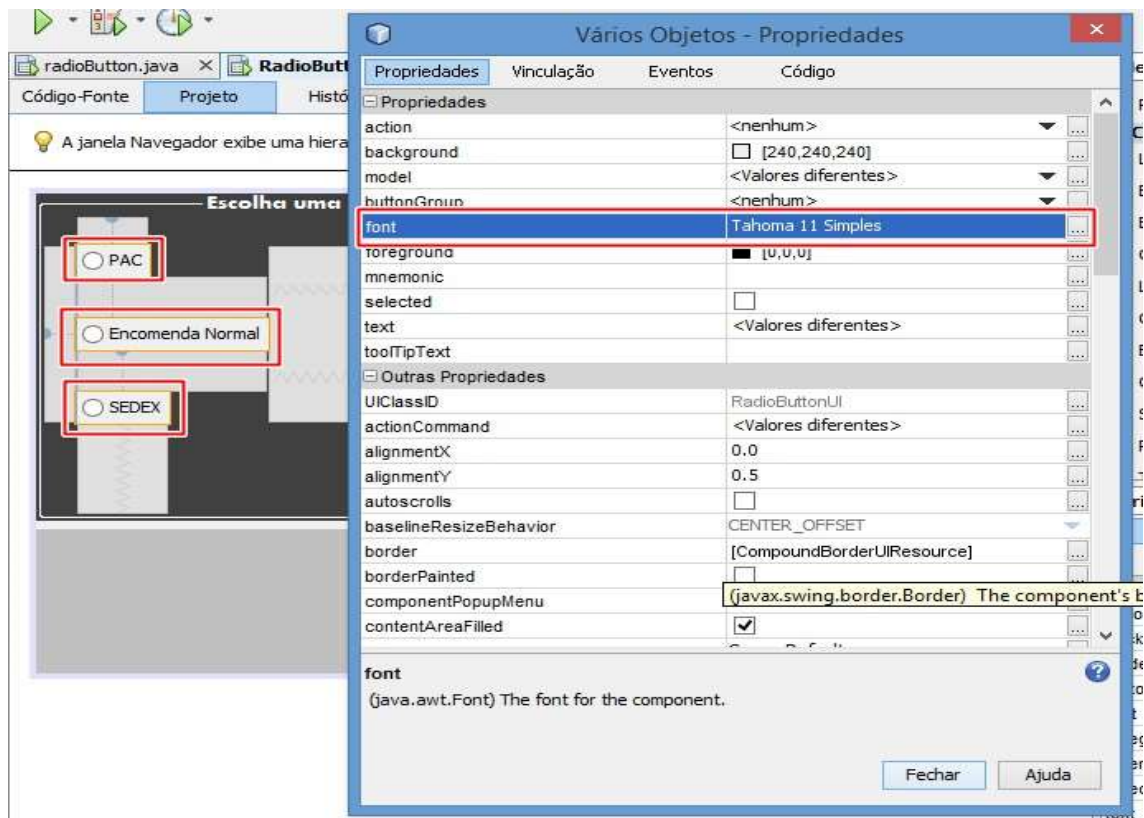


Figura 102 - Font jRadioButton.

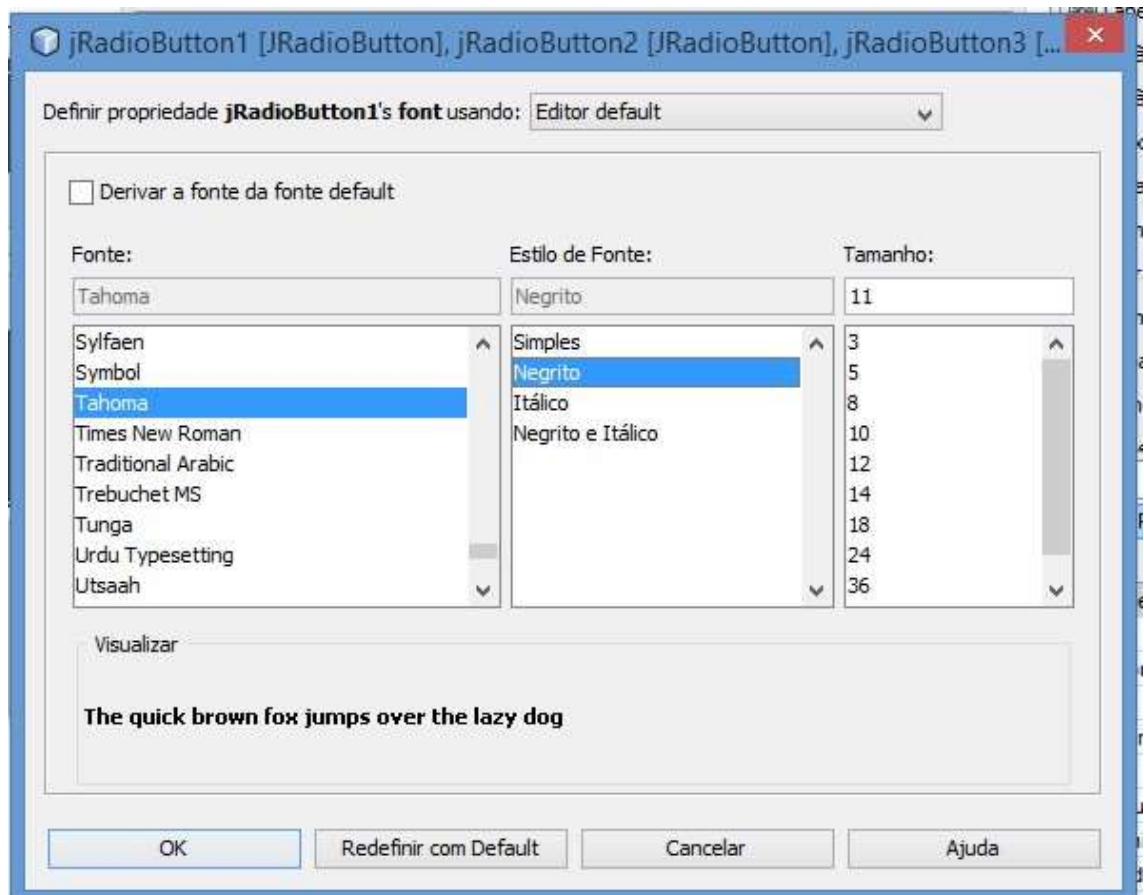


Figura 103 - Negrito jRadioButton.

Precisamos de um botão (JButton) para enviar o comando para limpar a seleção feita no programa, para isso vamos selecionar um **Botão** em **Controles Swing** e arrastá-lo até o painel inferior, como fizemos anteriormente com os botões de rádio, Figura 104 - Adicionando Botão:

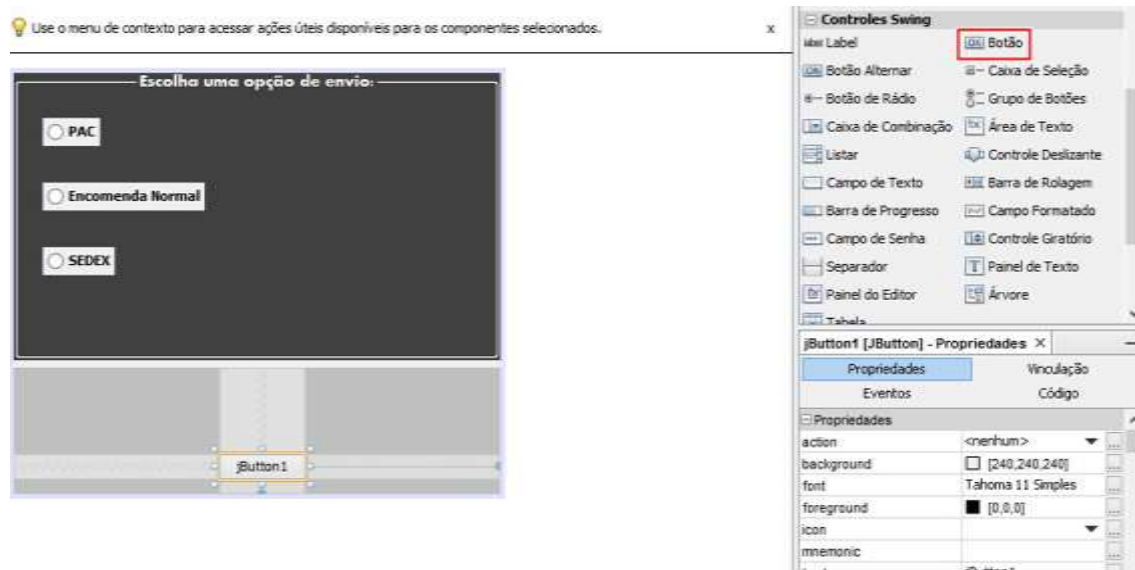


Figura 104 - Adicionando Botão.

Em seguida você irá editar o texto desse botão assim como fez para alterar os textos dos botões de rádio e irá escrever: Limpar Seleção. Agora vamos adicionar o **Label**, você irá mais uma vez na em **Controles Swing** e irá selecionar e arrastar os labels até o painel inferior, no total precisaremos de quatro labels que iram ficar dispostos como na Figura 105 - Adicionando Label.

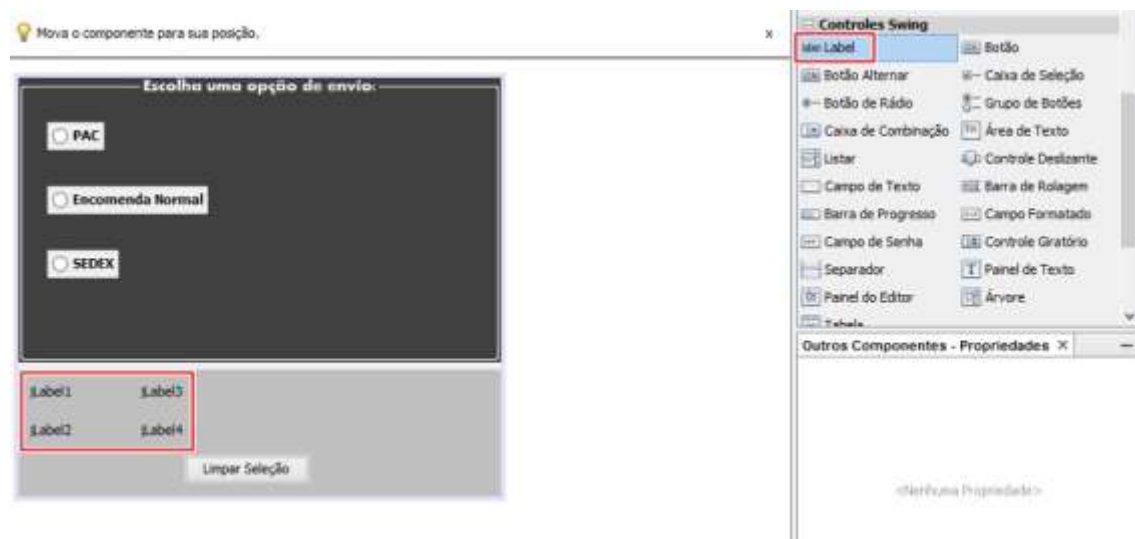


Figura 105 - Adicionando Label.

Após isso você irá editar os textos dos JLabel, como na Figura 106 - Textos JLabel:

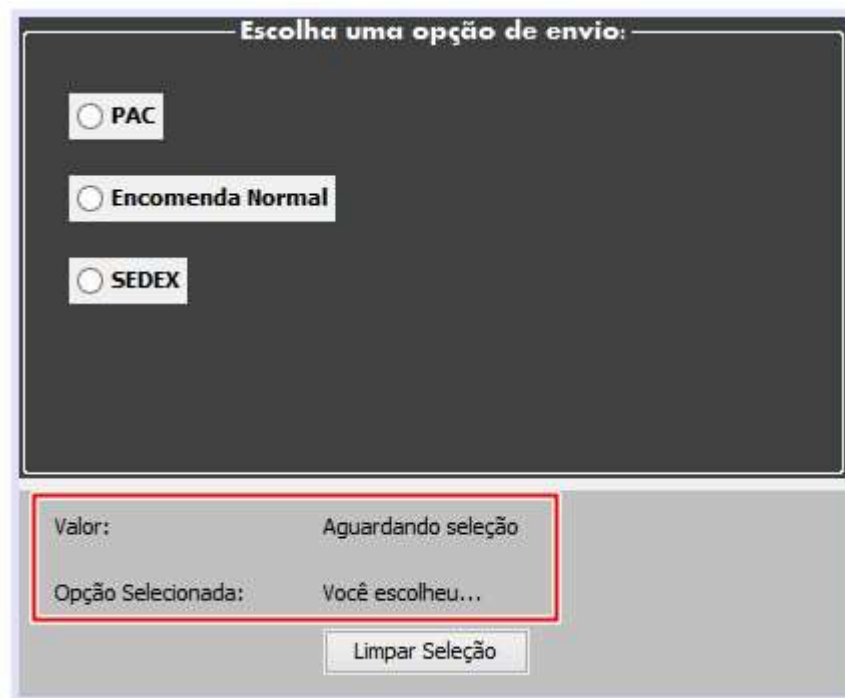


Figura 106 - Textos JLabel.

Após essa primeira etapa na construção do programa, se você clicar em Visualiza Design, irá perceber que é possível selecionar todas as opções do programa, no nosso programa queremos que apenas uma opção seja selecionada por vez, então vamos precisar acrescentar mais um componente neste o programa o **Grupo de Botões (buttonGroup)**. Você irá selecioná-lo em **Controles Swing** e arrastá-lo até painel superior e soltá-lo em qualquer espaço do painel superior como na Figura 107 - Grupo de Botões:

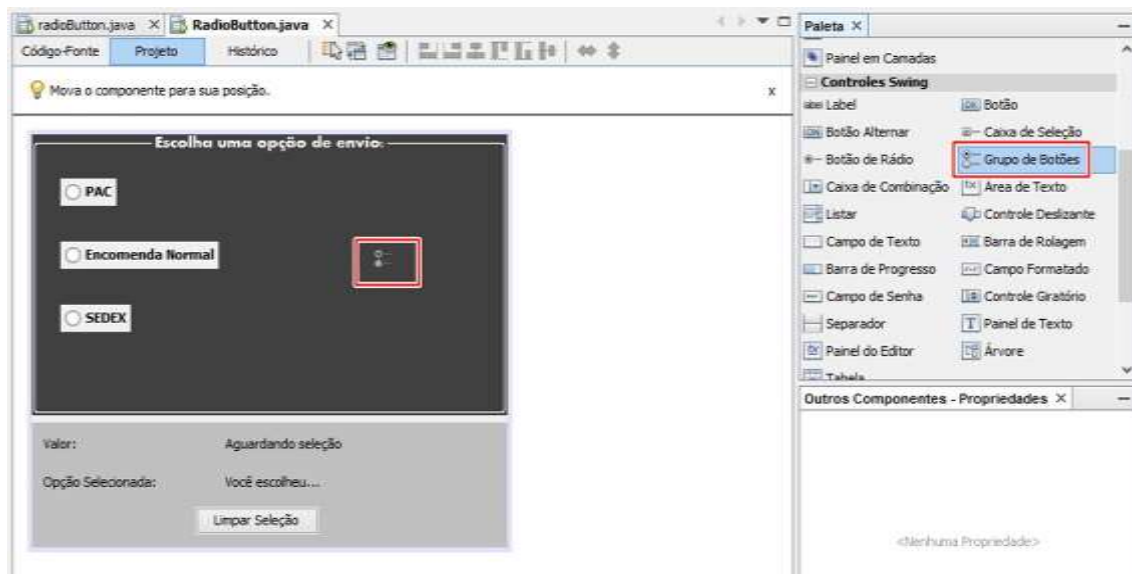


Figura 107 - Grupo de Botões.

Como esse é um componente “invisível” ele não será mostrado no **form** para configurá-lo precisamos ir ao **navegador** (canto direito inferior, destaque em preto) e lá veremos o **buttonGroup** (Destaque em vermelho) que usamos, Figura 108 - Navegador do NetBeans:

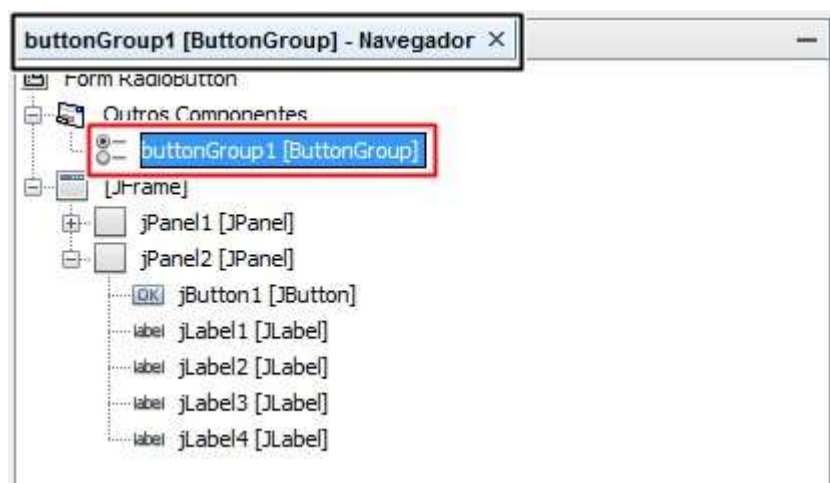


Figura 108 - Navegador do NetBeans.

Agora vamos renomear ele você irá clicar com o botão direito sobre o **buttonGroup** (destaque em vermelho, Figura 108 - Navegador do NetBeans) e clicar na opção: “Alterar o Nome da Variável...” e escrever: “buttonGroupOpcaoEnvio”, Figura 109 - Alterando Nome da Variável.



Figura 109 - Alterando Nome da Variável.

Agora que você viu como renomear o nome das variáveis, vamos renomear os outros componentes.

```
jRadioButton1 = jRadioButtonPac;
```

```
jRadioButton2 = jRadioButtonEncomendaNormal;
```

```
jRadioButton3 = jRadioButtonSedex;
```

```
jLabel3 = jLabelValor;
```

```
jLabel4 = jLabelOpcaoSelecionada;
```

```
jButton1 = jButtonLimparSelecao.
```

Em seguida vamos resolver o problema de selecionar apenas uma opção de envio por vez, para isso vamos clicar como botão direito sobre botão de rádio PAC, e iremos à opção propriedades. Em propriedades na opção **buttonGroup** selecione o Grupo de Botões que usamos, **buttonGroupOpcaoEnvio**, Figura 110 - Propriedade buttonGroup.

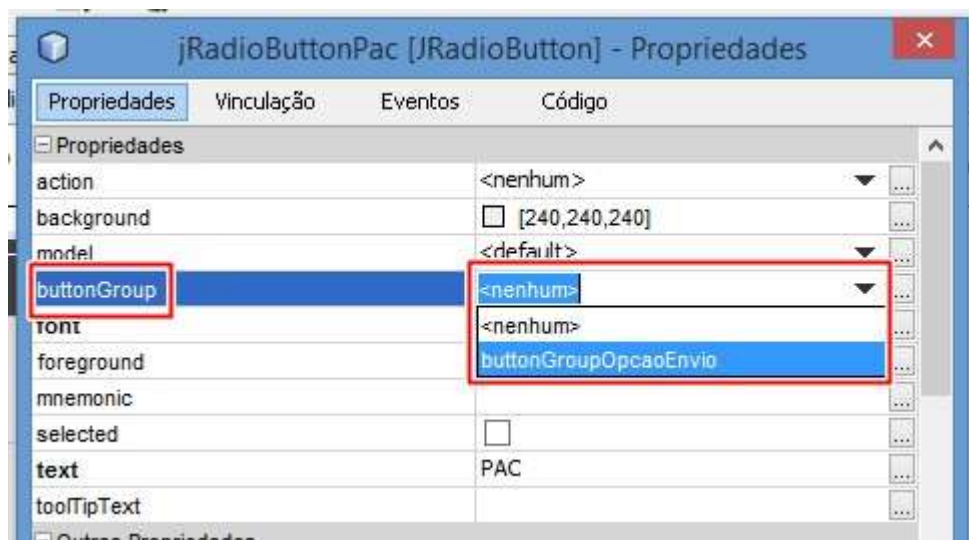


Figura 110 - Propriedade buttonGroup.

Você irá fazer isso com os outros dois botões de rádio o Encomenda Normal e o Sedex. Após isso, se você clicar em visualizar design perceberá que agora apenas uma das opções será selecionada.

Agora vamos dar mais funcionalidades ao nosso programa iremos fazer com que quando selecionarmos uma opção de envio no programa ele mostre o valor e a opção selecionada. Para isso clicamos com o botão direito em cima do primeiro botão de rádio, vamos à opção eventos, action, actionPerformed. Figura 111 - actionPerformed:

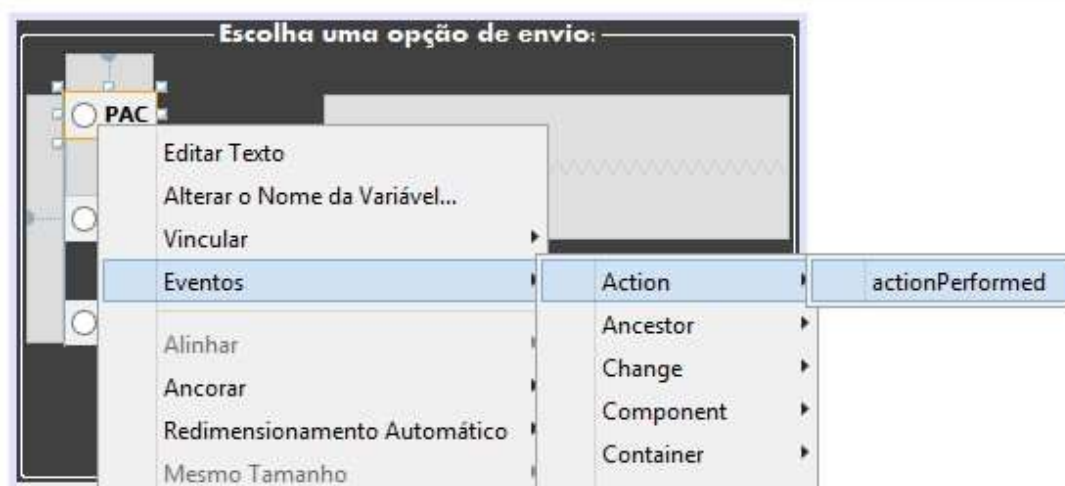
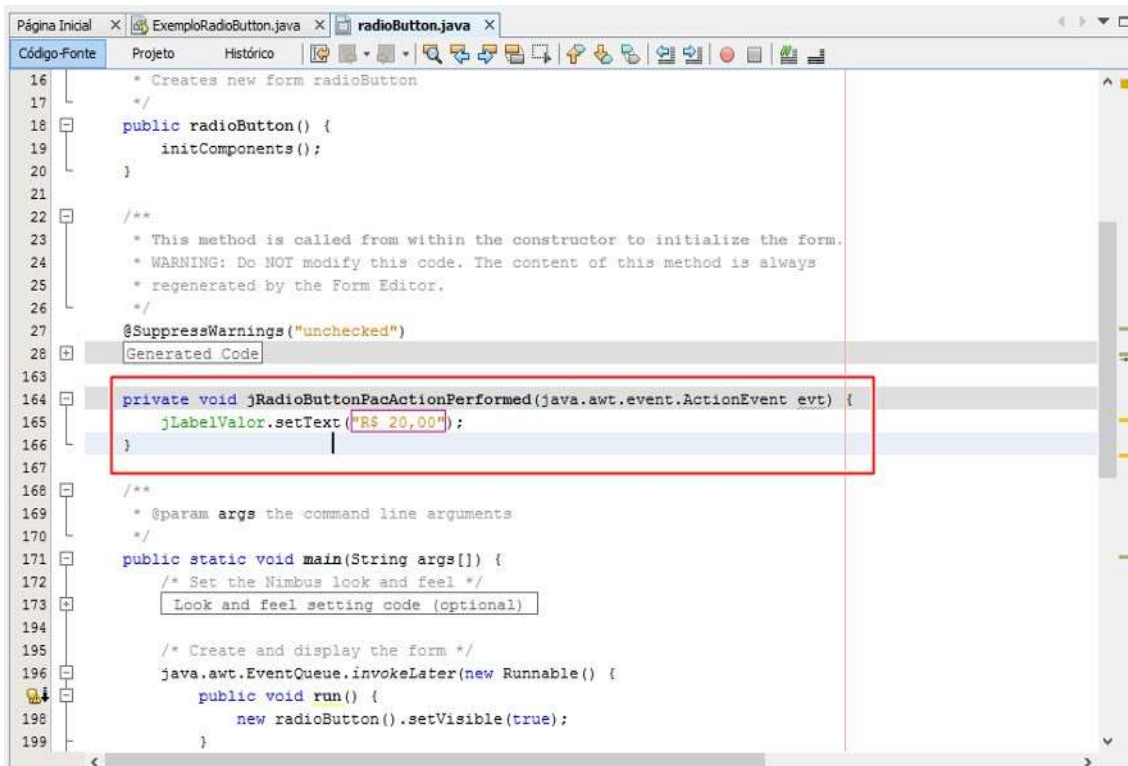


Figura 111 - actionPerformed.

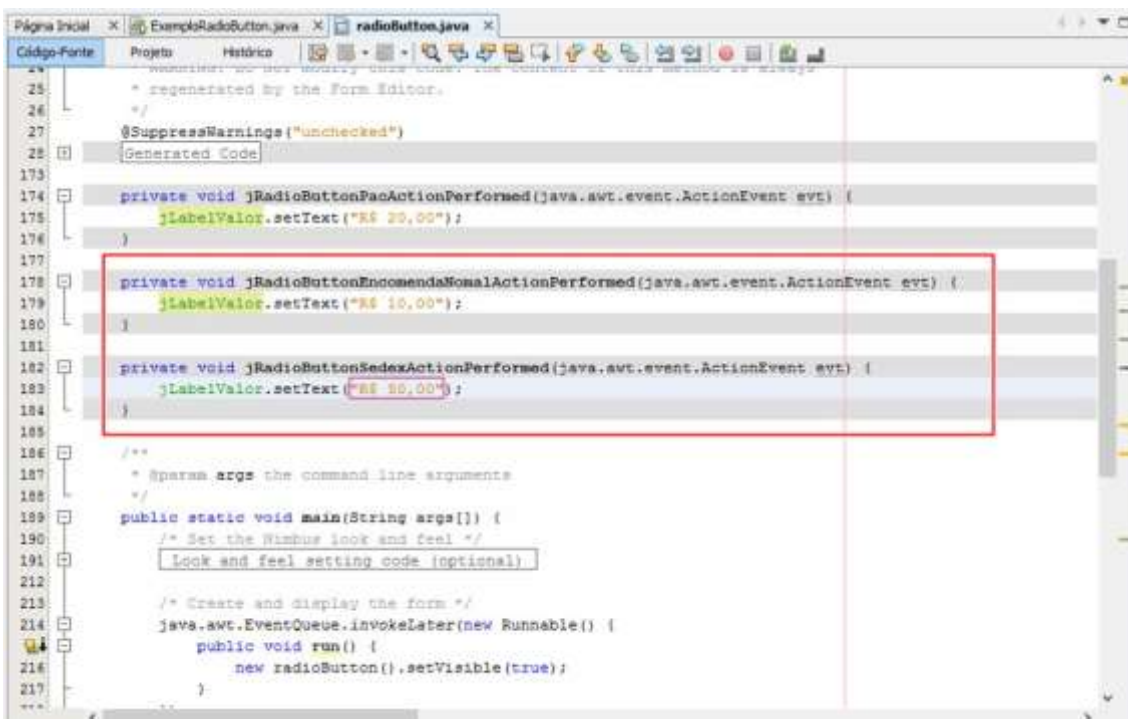
Ao fazermos isto iremos para a área de Código-Fonte do programa, no método que surgiu vamos deixá-lo como na Figura 112 - Código PAC.



```
16      * Creates new form radioButton
17      */
18      public radioButton() {
19          initComponents();
20      }
21
22      /**
23       * This method is called from within the constructor to initialize the form.
24       * WARNING: Do NOT modify this code. The content of this method is always
25       * regenerated by the Form Editor.
26       */
27      @SuppressWarnings("unchecked")
28      Generated Code
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64      private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
65          jLabelValor.setText("R$ 20,00");
66      }
67
68      /**
69       * @param args the command line arguments
70       */
71      public static void main(String args[]) {
72          /* Set the Nimbus look and feel */
73          Look and feel setting code (optional)
74
75          /* Create and display the form */
76          java.awt.EventQueue.invokeLater(new Runnable() {
77              public void run() {
78                  new radioButton().setVisible(true);
79              }
80          });
81      }
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Figura 112 - Código PAC.

Iremos fazer o mesmo para os outros dois botões de rádio, Encomenda Normal e Sedex, só mudando o código, veja Figura 113 - Código Encomenda Normal e Sedex:



```
25      * regenerated by the Form Editor.
26      */
27      @SuppressWarnings("unchecked")
28      Generated Code
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64      private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
65          jLabelValor.setText("R$ 20,00");
66      }
67
68      /**
69       * @param args the command line arguments
70       */
71      public static void main(String args[]) {
72          /* Set the Nimbus look and feel */
73          Look and feel setting code (optional)
74
75          /* Create and display the form */
76          java.awt.EventQueue.invokeLater(new Runnable() {
77              public void run() {
78                  new radioButton().setVisible(true);
79              }
80          });
81      }
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Figura 113 - Código Encomenda Normal e Sedex.

Se você executar seu programa agora irá ver que o problema de mostrar os valores já está resolvido.

Iremos agora resolver o problema de mostrar a opção selecionada, para isso vamos ter que pegar o texto dos botões de rádio e mostra no label as opção selecionadas, vamos precisar os métodos `getSelection()` e `getActionCommand()`, o código ficará como a Figura 114 -Código Opção Selecionada:

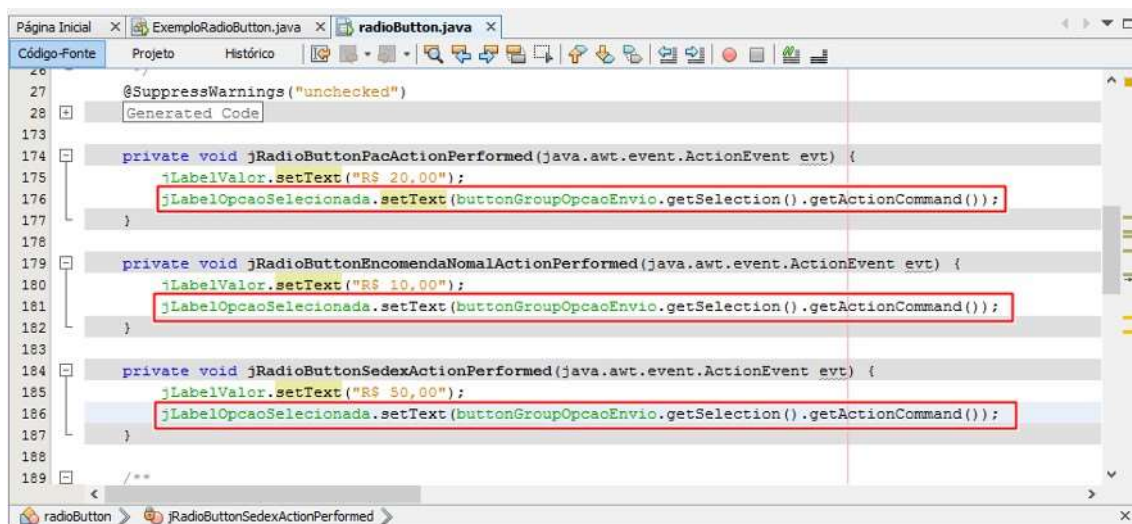


Figura 114 - Código Opção Selecionada.

Ainda precisamos fazer mais uma coisa para que o programa faça o que queremos, que é mostrar a opção selecionada, para isso vamos selecionar o botão de rádio PAC e clicar com o botão direito e acessar as suas propriedades, na opção **actionCommand** você verá que ele está com o nome PAC em maiúsculo, precisamos colocá-lo em minúsculo (destaque em vermelho Figura 115 - ActionCommand) pois o actionCommand não funciona com letras maiúsculas.

Iremos fazer o mesmo com os demais botões de rádio, Encomenda Normal e Sedex, colocar o actionCommand minúsculo.

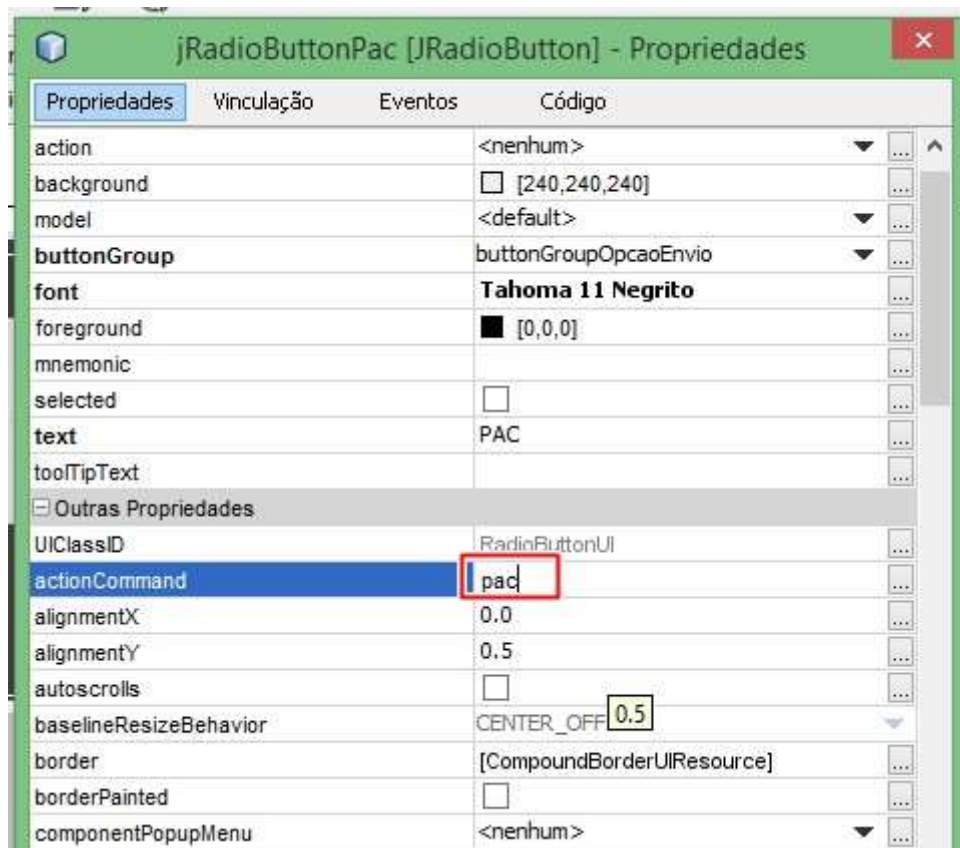


Figura 115 - ActionCommand.

Por fim vamos adicionar a função de limpar seleção. Para isso vamos clicar com o botão direito sobre o botão Limpar Seleção e acionar o actionPerformed como fizemos com os botões de rádio. No Código-Fonte vamos acrescentar o código da Figura 116 - Código Botão Limpar Seleção:

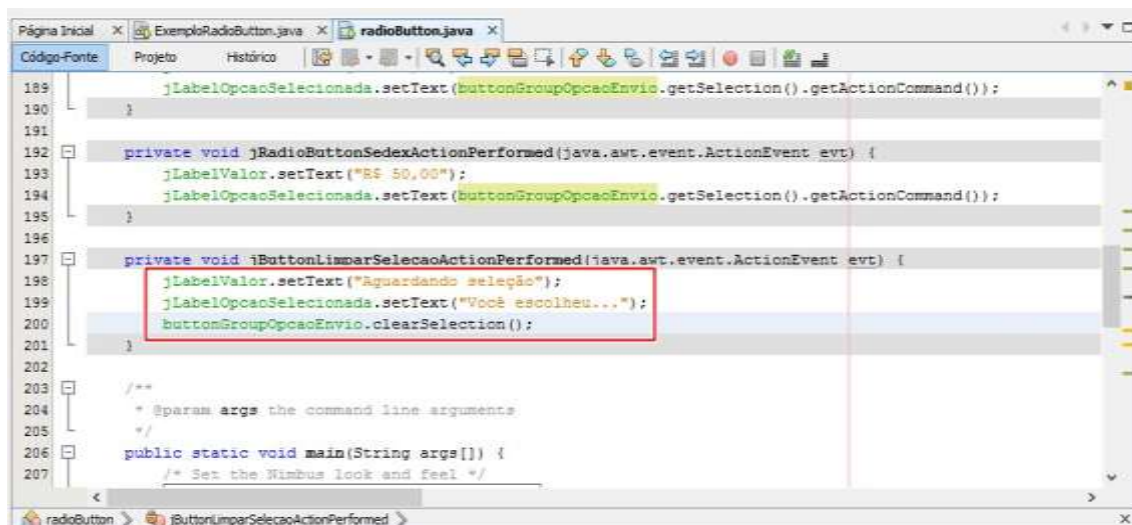


Figura 116 - Código Botão Limpar Seleção.

Agora temos o nosso código com todas as funcionalidades que queríamos adicionar, execute seu programa para ver como ficou, Figura 117 - Exemplo Radio Button.



Exemplo de uso Radio Button

Escolha uma opção de envio:

☐ PAC

☒ Encomenda Normal

☐ SEDEX

Valor: R\$ 10,00

Opção Seleccionada: encomenda normal

Limpar Seleção

Figura 117 - Exemplo Radio Button.

jcheckBox

Uma caixa de seleção é um componente gráfico que possui dois estados: "verdadeiro" (selecionada) e estado "falso" (sem seleção). Ao clicar em uma caixa de seleção muda seu estado de "verdadeiro" para "falso", ou em "falso" para "verdadeiro".

Ele é parecido com Botão de Rádio de selecionar as opções que são verdadeiras, mas com a diferença de poder selecionar várias opções ao mesmo tempo, por exemplo, vamos desenvolver um aplicativo que verifica qual é o gosto musical das pessoas, conforme Figura 118 - Exemplo CheckBox, repare que há várias opções e que o usuário pode selecionar todas, nenhuma ou qualquer uma.



Figura 118 - Exemplo CheckBox.

Para criar um novo projeto no Netbeans clique na aba **Arquivo**, **NovoProjeto**, Figura 119 - Criando Projeto.

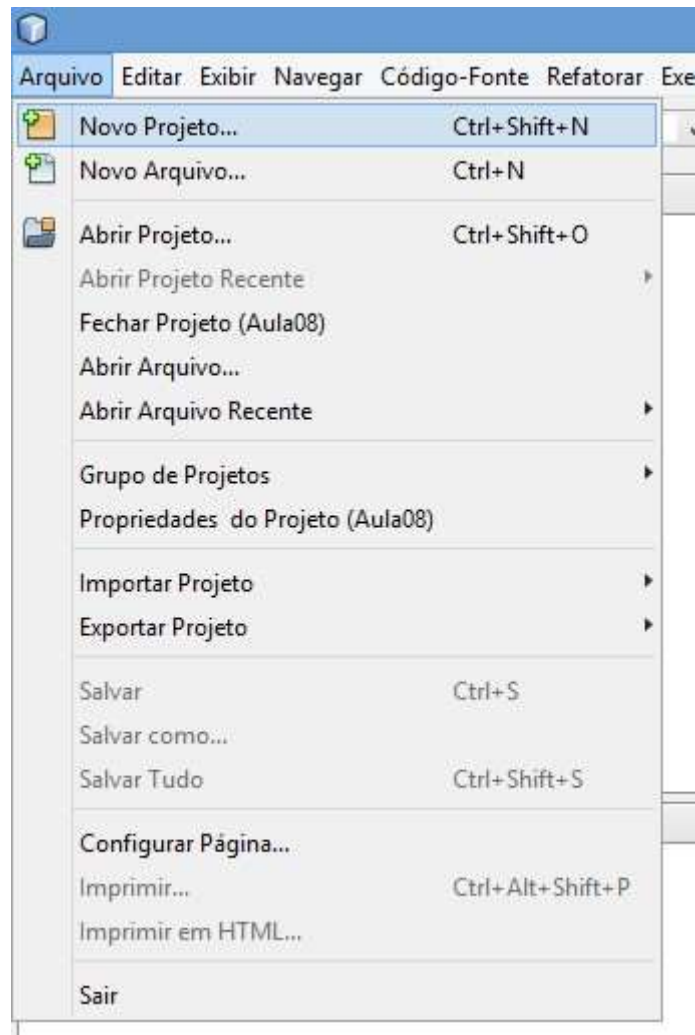


Figura 119 - Criando Projeto.

Em seguida aparecerá à tela para escolha do tipo de projeto, Figura 120 - Escolha do Projeto, escolha **Categoria, Java** e **Projetos, Aplicação Java** para desenvolver sistemas para Desktop.

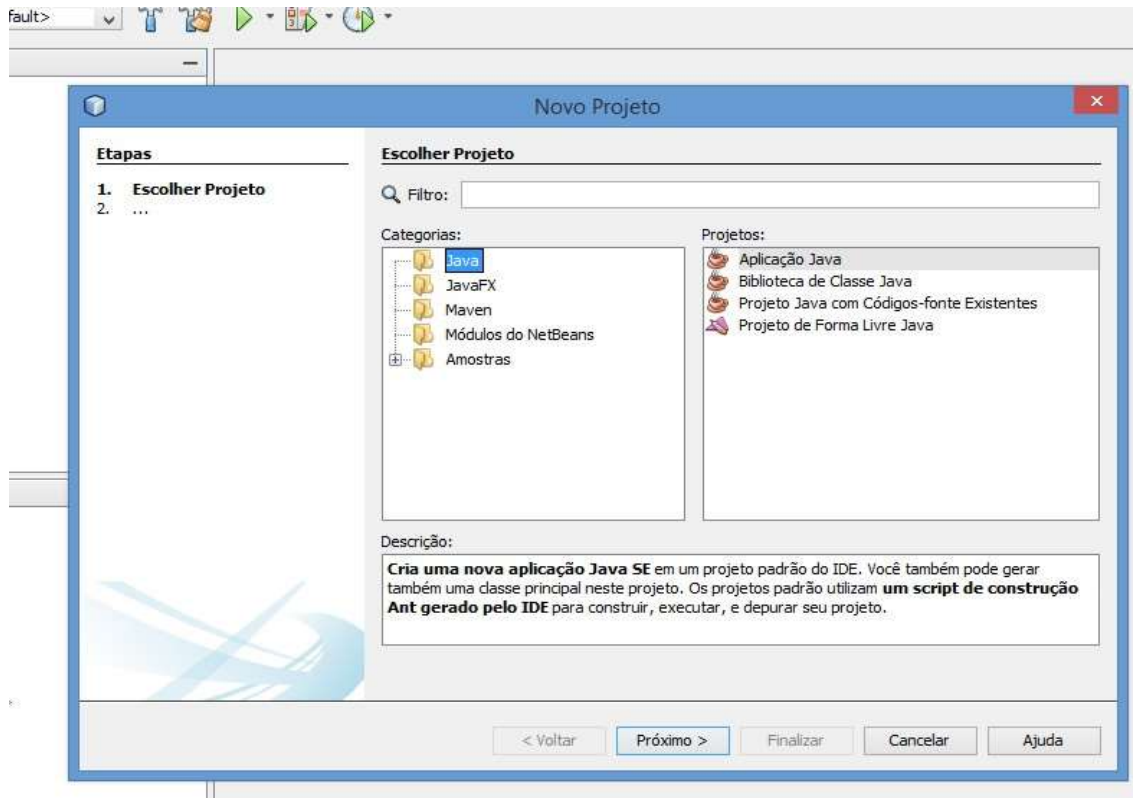


Figura 120 - Escolha do Projeto.

Próximo passo é definir o nome **ExemploCheckBox** para o projeto, Figura 121 - Atribuir Nome ao Projeto, neste caso devemos colocar um nome sem acentuação e espaços em branco, normalmente a primeira letra do projeto é maiúscula.

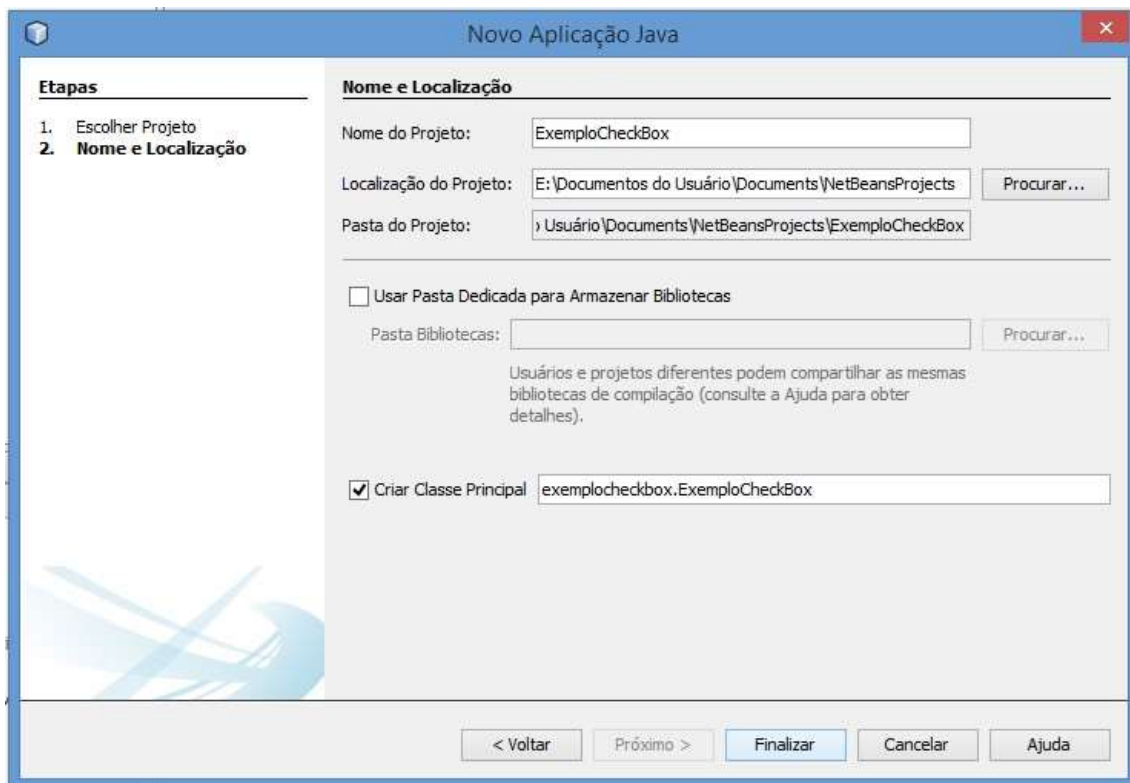


Figura 121 - Atribuir Nome ao Projeto

Criar uma nova janela (JFrame): Clique com botão direito em cima da xícara (Projeto criado) e selecione **Novo, Form JFrame**, Figura 122 - Criação de Janelas:

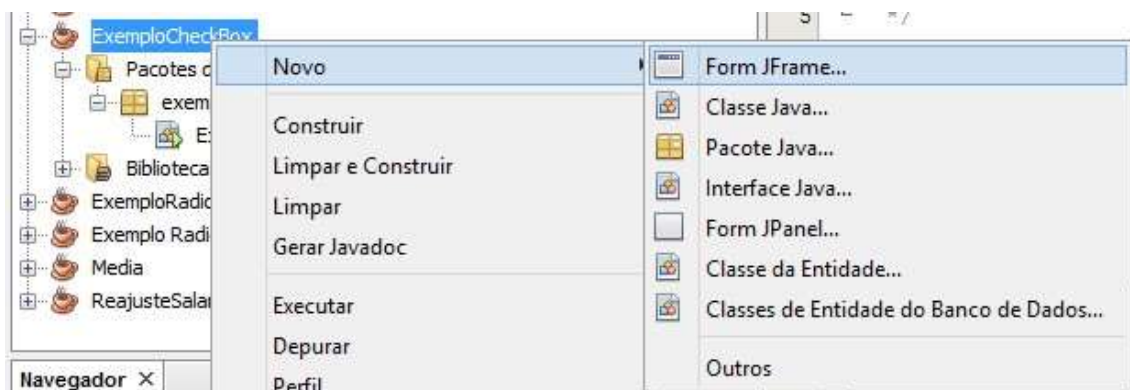


Figura 122 - Criação de Janelas.

Agora vamos definir um nome para a janela que será **checkBox**, no campo pacote, que não pode ficar vazio, temos que colocar esses JFrames dentro de um pacote, no caso vamos escrever **telas** que será nossa pasta que irá possuir todos os JFrames/Janelas do sistema, Figura 123 - Definindo o Nome do JFrame:

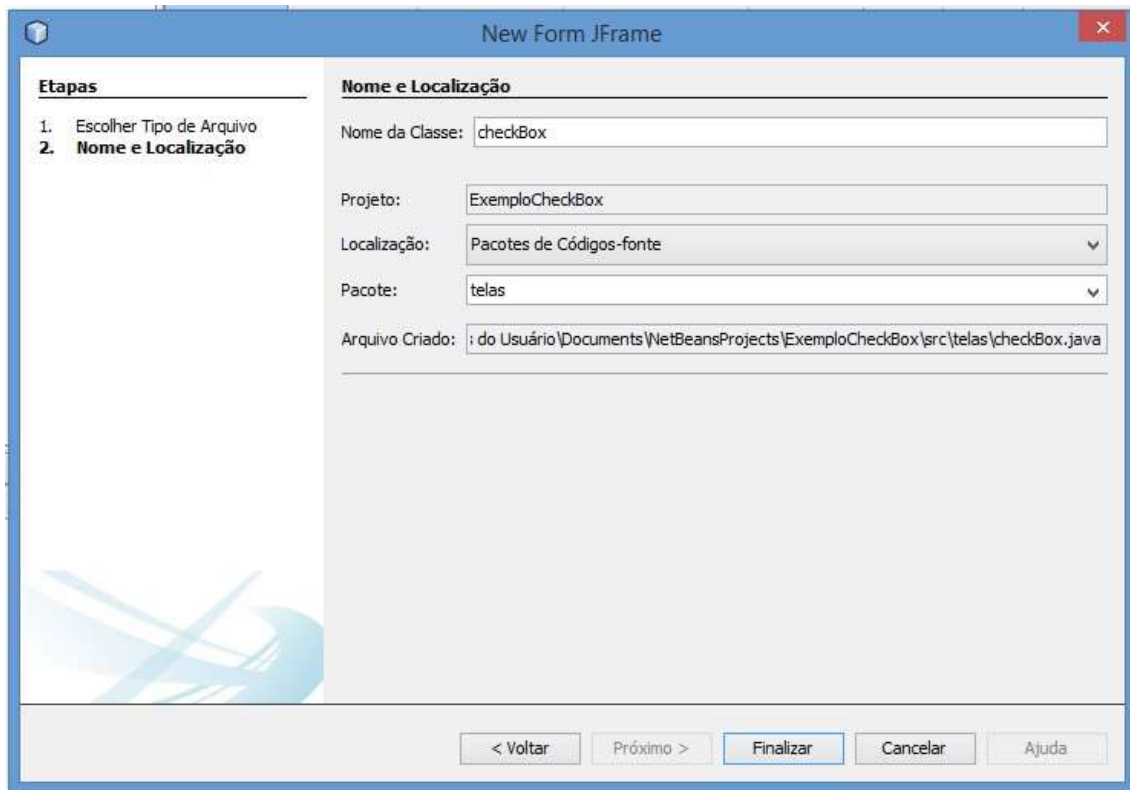


Figura 123 - Definindo o Nome do JFrame.

Com o componente selecionado dá para alterar aparência, texto e tamanho dele, abaixo segue uns dos principais atributos que podemos modificar:

Background: cor de fundo;

Font: Tipo de letra e tamanho;

Foreground: Cor da letra;

Text: O texto que vai aparecer na tela.

Após a criação do JFrame vamos modificar algumas das suas propriedades, para melhorar seu visual. Clique com o botão direito do mouse sobre o JFrame, e clique em **propriedades**, Figura 124 - Acessando as Propriedades do JFrame:

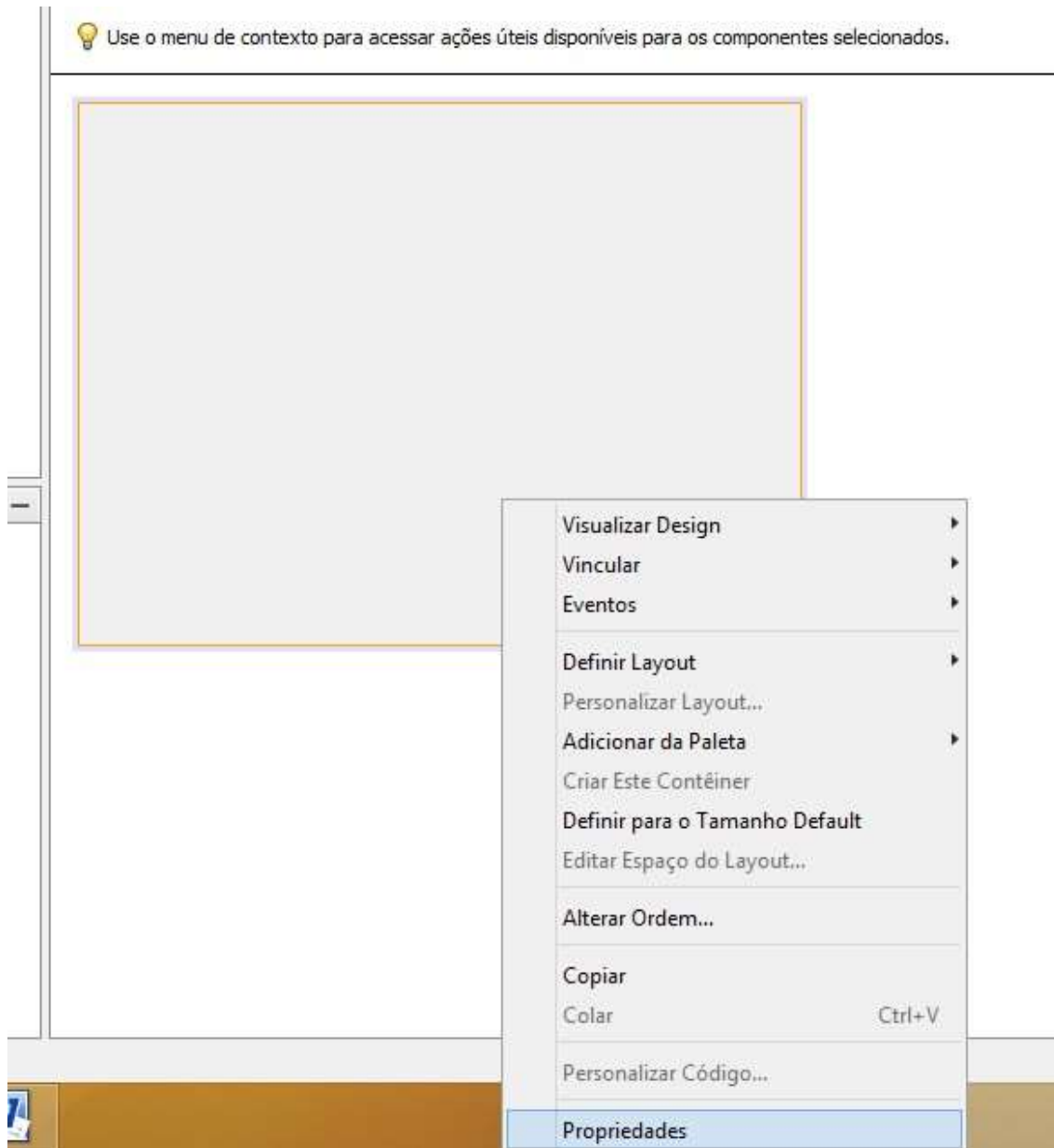


Figura 124 - Acessando as Propriedades do JFrame.

Vamos primeiro colocar a opção **defaultCloseOperation** como **DISPOSE**, para permitir o fechamento do formulário (Form), Figura 125 - Opção **defaultCloseOperationDISPOSE**:

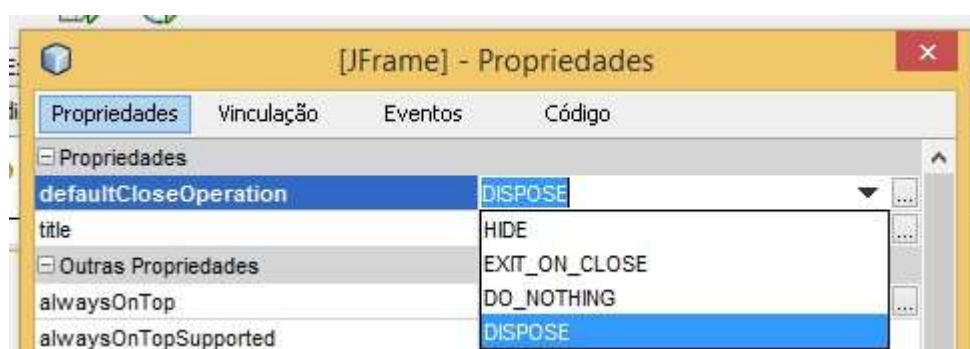


Figura 125 - Opção defaultCloseOperation DISPOSE.

Modificar o **title** (título) do nosso formulário para: **Exemplo de uso Radio Button**, Figura 126 - Modificando o title:

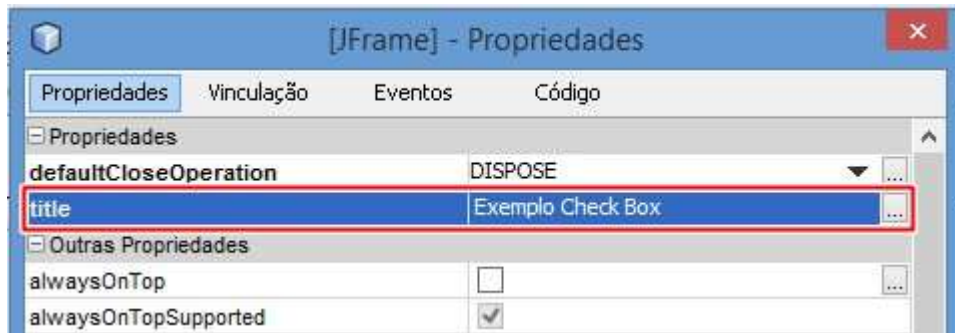


Figura 126 - Modificando o title.

Em seguida **desmarcamos** a opção **resizable**, para impedir que o usuário redimensione a tela do programa, ou seja, maximize ou minimize. Figura 127 - Resizable:

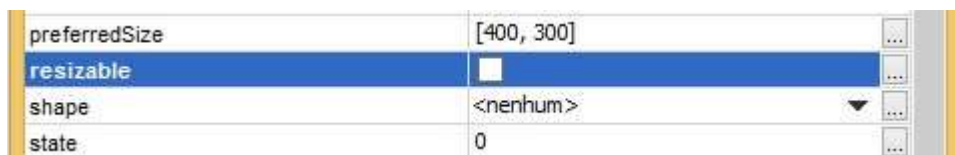


Figura 127 - Resizable.

Agora na aba **código** vamos selecionar a opção **Gerar Centralizado**, para que o programa sempre que for executado inicie no centro do monitor. Figura 128 - Gerar Centralizado:

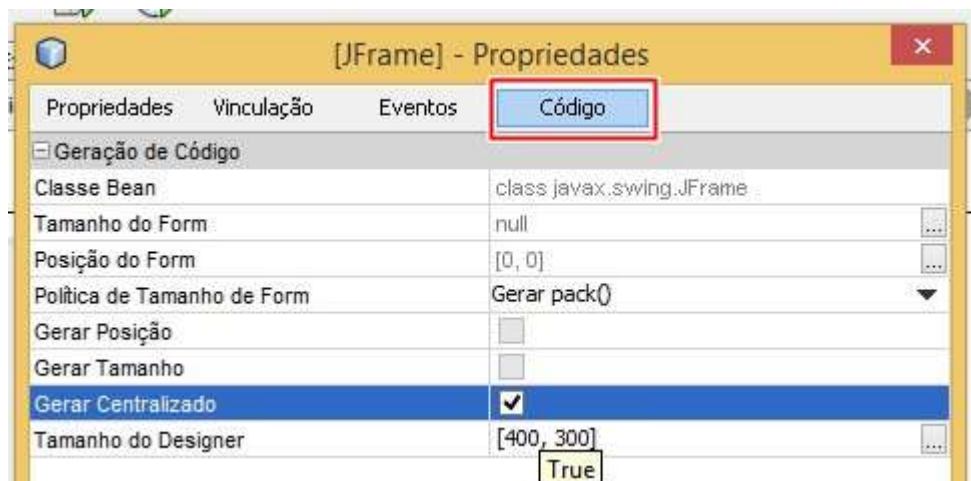


Figura 128 - Gerar Centralizado.

Depois disso você pode ir à opção **Visualizar Design** do Netbeans (destacado em vermelho na Figura 129 - Visualizar Design), para ver como está o visual

do seu programa, através dele você pode visualizar uma simulação da sua aplicação rodando, contudo essa opção se restringe a visualização do layout do seu programa já que as suas funcionalidades não poderão ser executados.

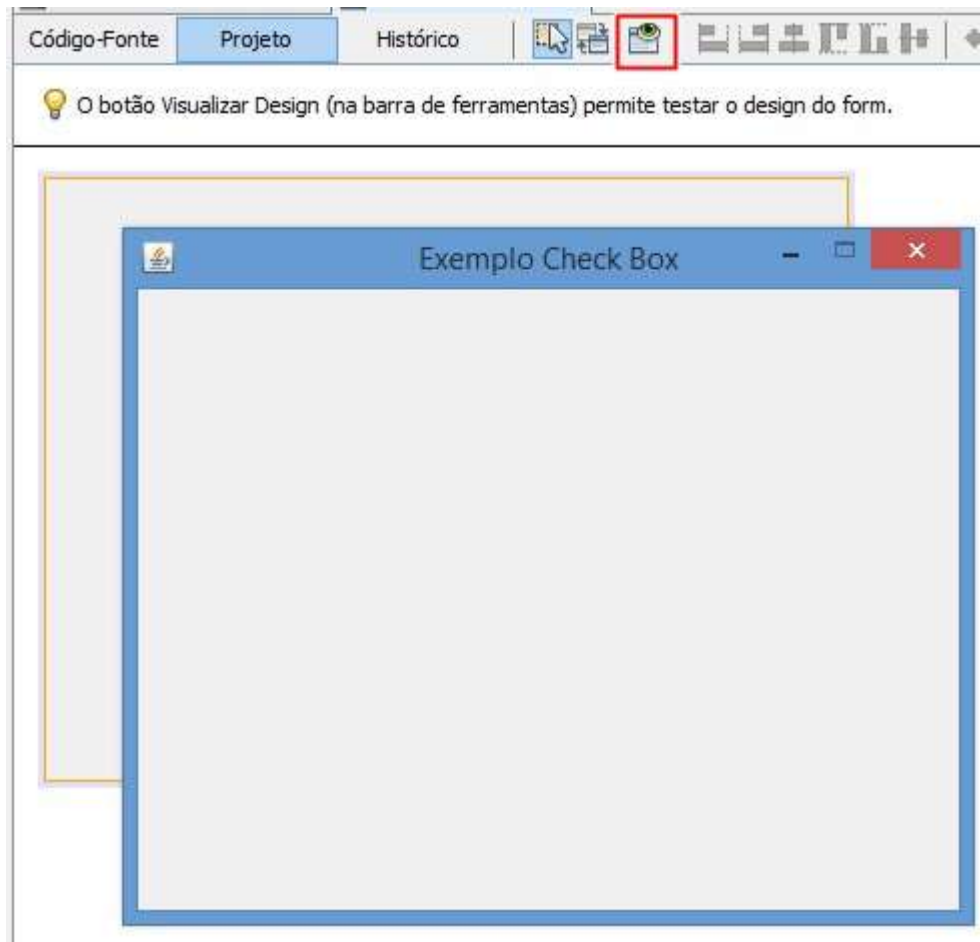


Figura 129 - Visualizar Design.

Agora vamos configurar alguns itens, você irá à **paleta**, canto direito do NetBeans, Figura 130 - Opções da Paleta, e selecionaria na opção **ContêineresSwing** um **Painel**, pressionando o botão esquerdo do mouse, até o JFrame, depois irá redimensioná-lo como na Figura 131 - Usando o Painel:



Figura 130 - Opções da Paleta.

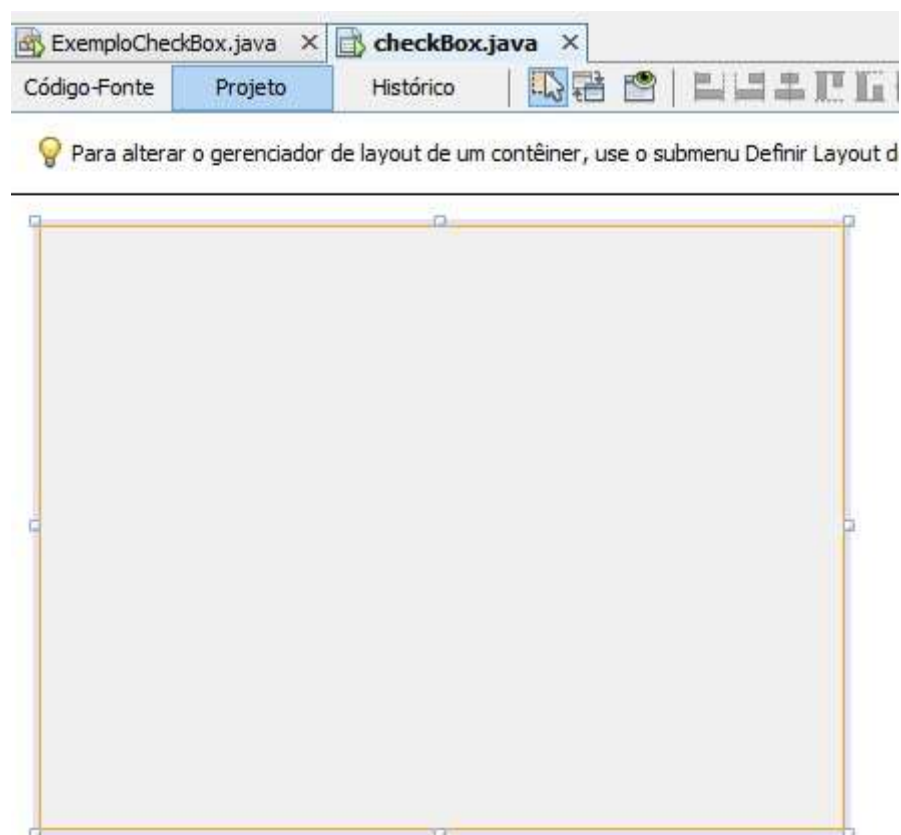


Figura 131 - Usando o Painel.

Vamos agora adicionar algumas opções para mudar a aparência do aplicativo, vamos selecionar a painel superior, e clicar com o botão direito do mouse para acessar as **propriedades** do painel, e então iremos clicar nas reticências (destacadas em vermelho), Figura 132 - Bordas do Painel:

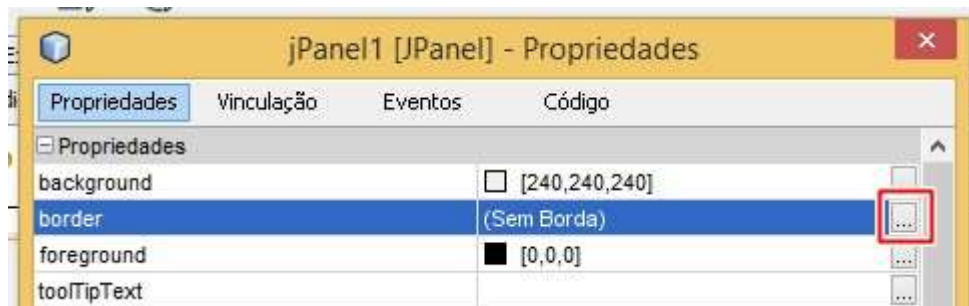


Figura 132 - Bordas do Painei.

No menu borda você irá selecionar a opção **BordacomTítulo**, Figura 133 - Borda com Título:

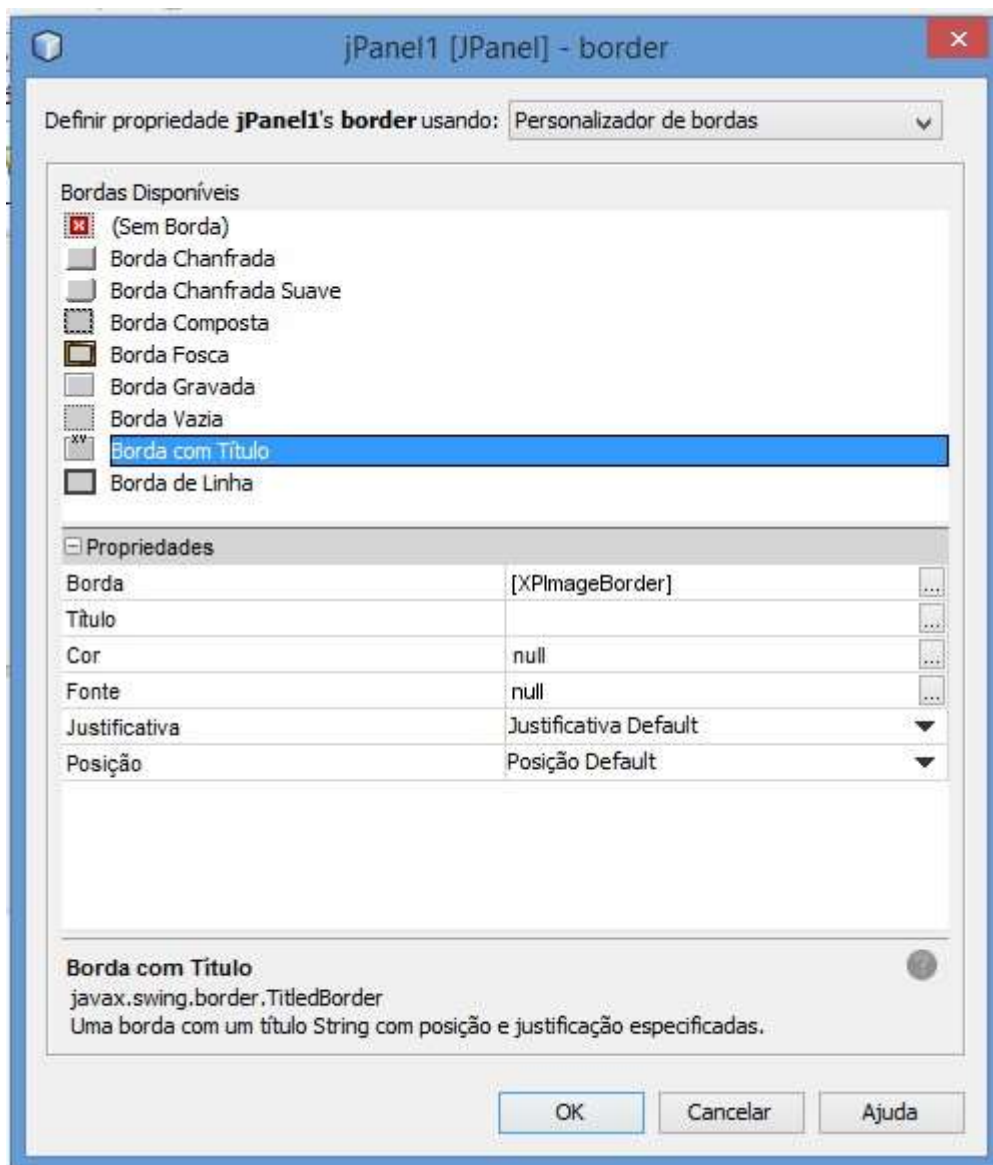


Figura 133 - Borda com Título.

Agora vamos por um título nessa borda, digitando na opção **Título** a frase: “Qual p o seu gosto musical ?” como na Figura 134 - Título da Borda:

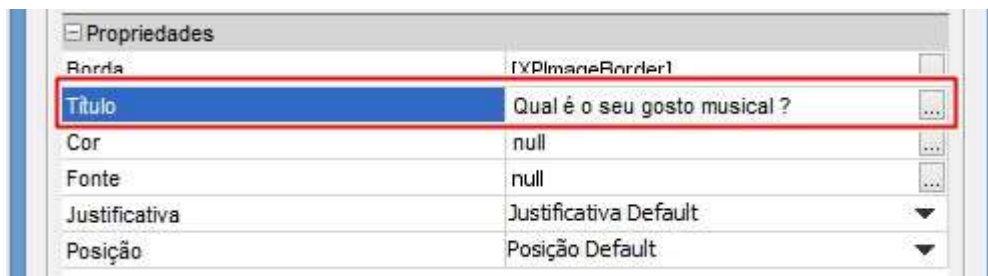


Figura 134 - Título da Borda.

Na opção **Cor** iremos selecionar Preto, mas você pode experimentar outras cores, Figura 135 - Cor da Borda:

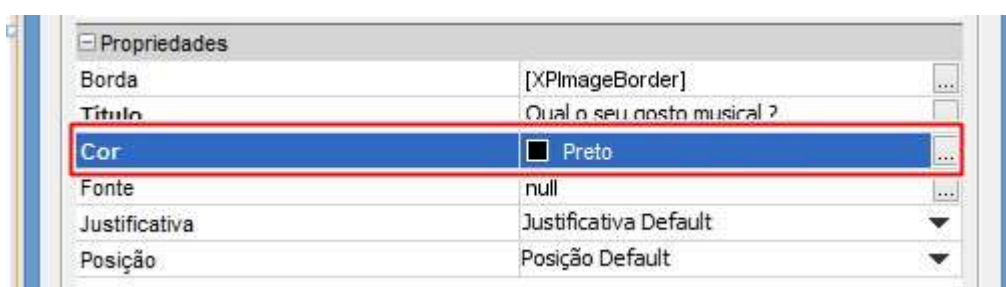


Figura 135 - Cor da Borda.

Na opção **Fonte** você irá clicar nas reticências (destacada em vermelho), Figura 136 - Fonte da Borda:

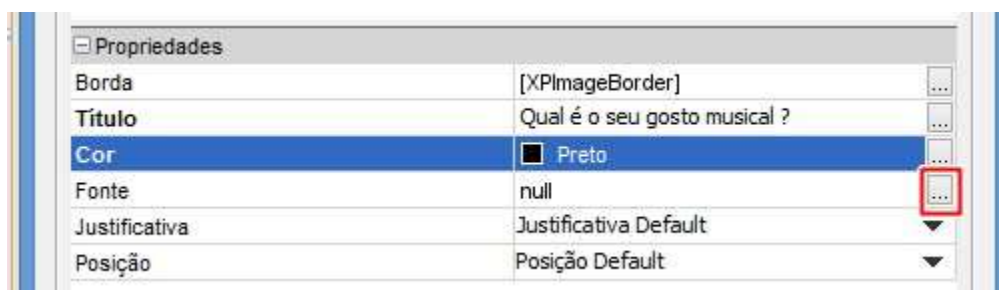


Figura 136 - Fonte da Borda.

Você pode selecionar as opções que quiser no menu Borda com Título - Fonte, mas no nosso exemplo selecionamos Tamanho: 14, Estilo da Fonte: Negrito e Fonte: Aharoni. Figura 20 - Opções Fonte da Borda:



Figura 137 - Opções Fonte da Borda.

E a **Justificativa** será a opção **Centro**, Figura 138 - Justificativa da Borda:

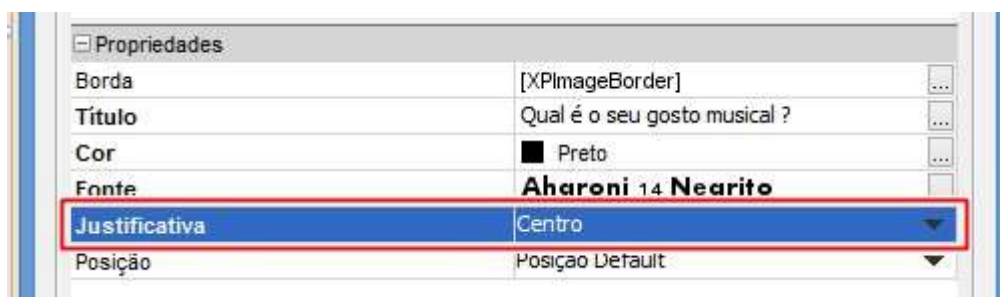


Figura 138 - Justificativa da Borda.

Após esta primeira parte iremos pôr uma borda dentro da borda selecionada, você irá clicar nas reticências da opção **Borda** (destacada em vermelho), Figura 139 - Borda sobre Borda:

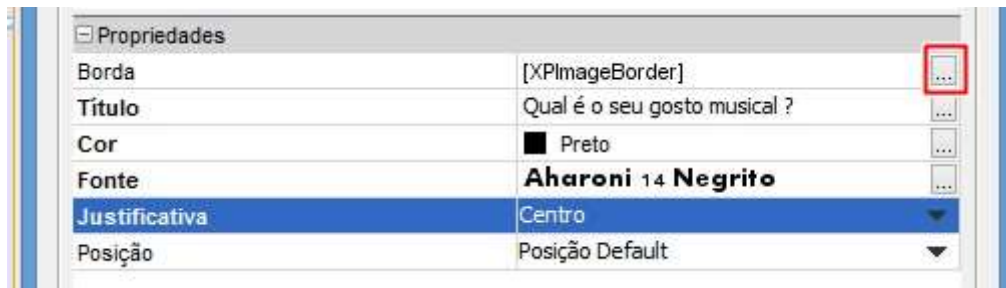


Figura 139 - Borda sobre Borda

Após isso aparecerá uma nova tela como a Figura 140 - Borda com Título - Borda, nela você pode mudar algumas características dessa nova borda. Neste exemplo iremos deixar a opção **Cantos Arredondados** selecionada:

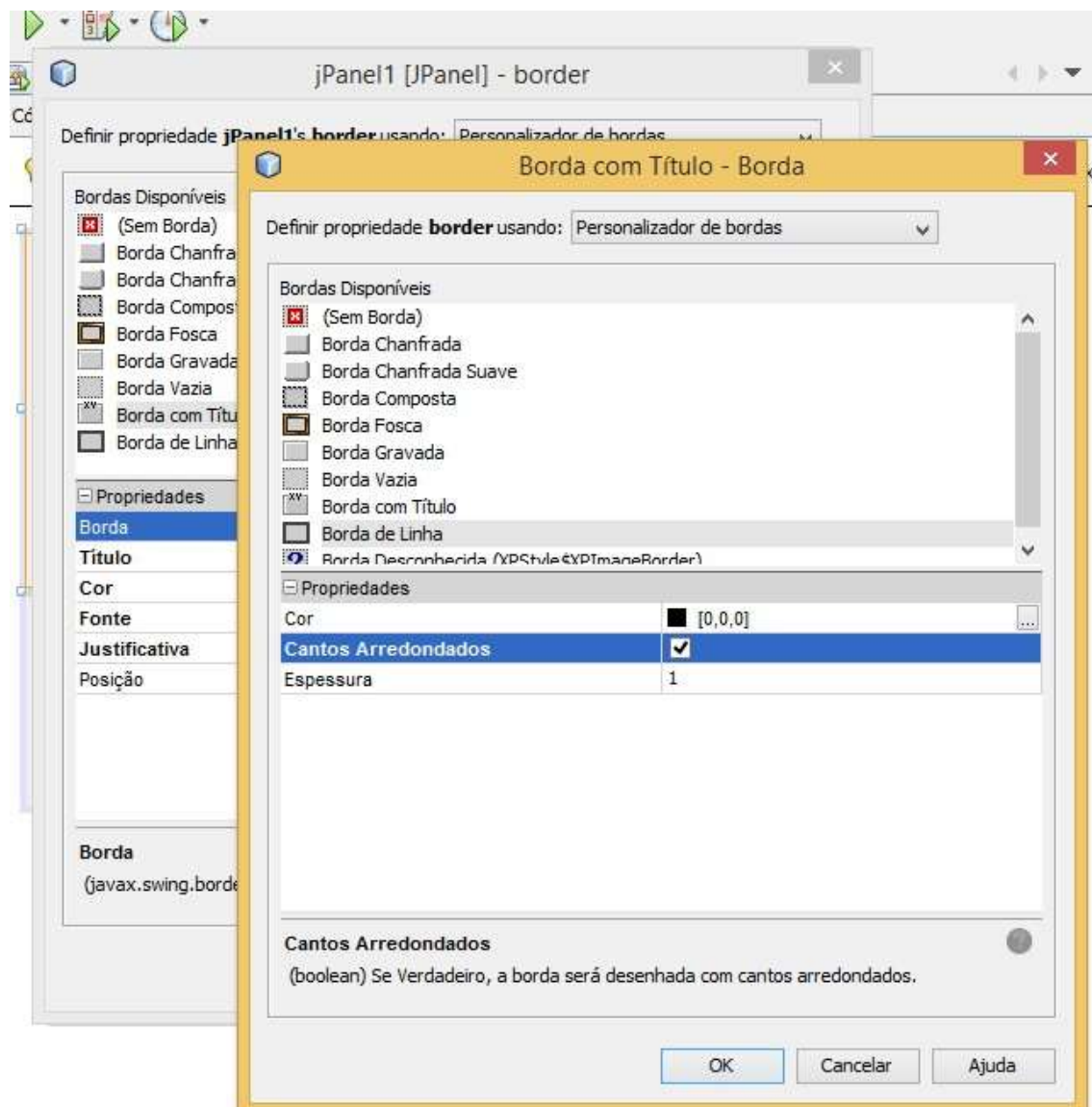


Figura 140 - Borda com Título - Borda.

Agora vamos escolher uma cor de fundo (**Background**) para o label superior, na qual o Branco realce. Você voltara para as propriedades do painel, Figura 141 - Propriedades do Painel:

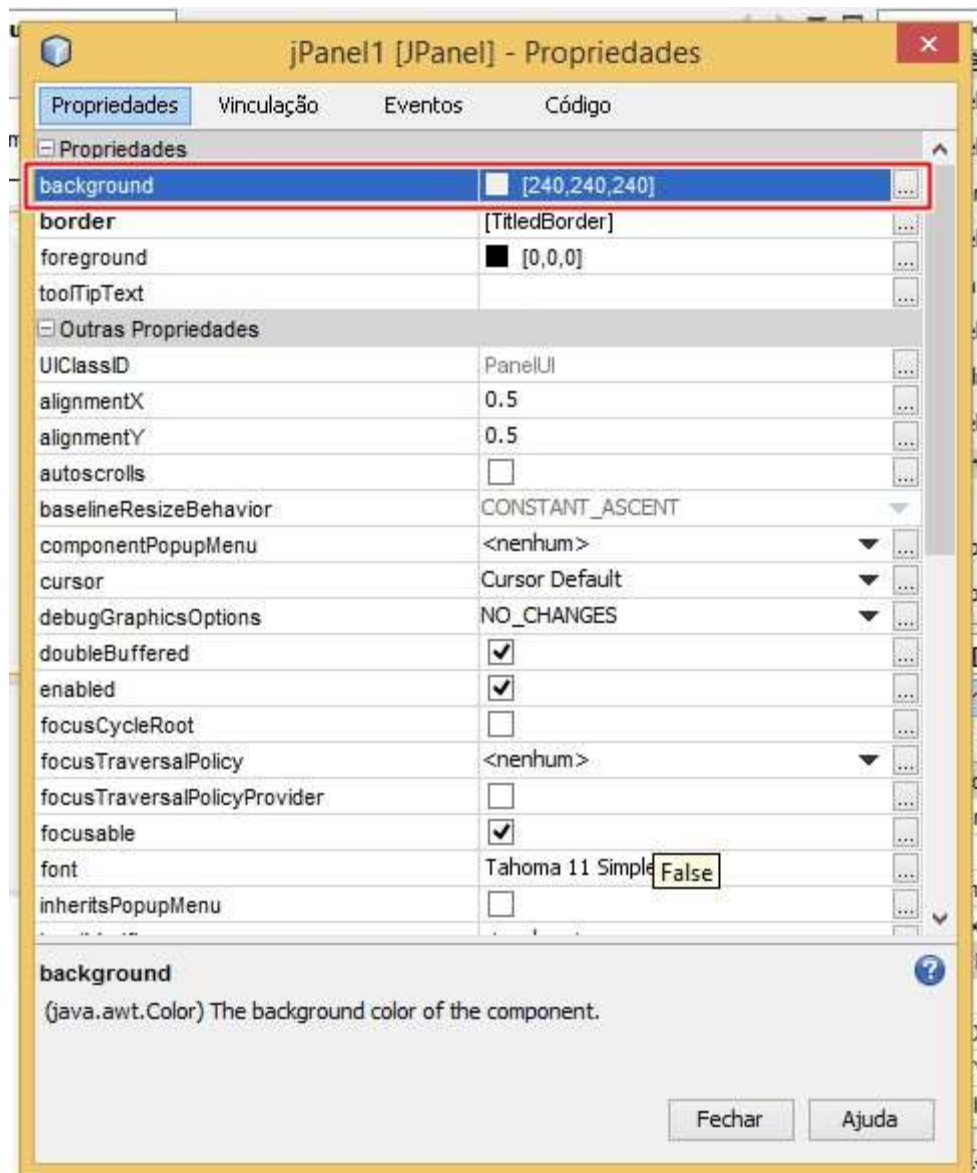


Figura 141 - Propriedades do Painel.

Na **PaletaAWT** você irá selecionar a opção Cinza Claro, Figura 142 - Background Painel:

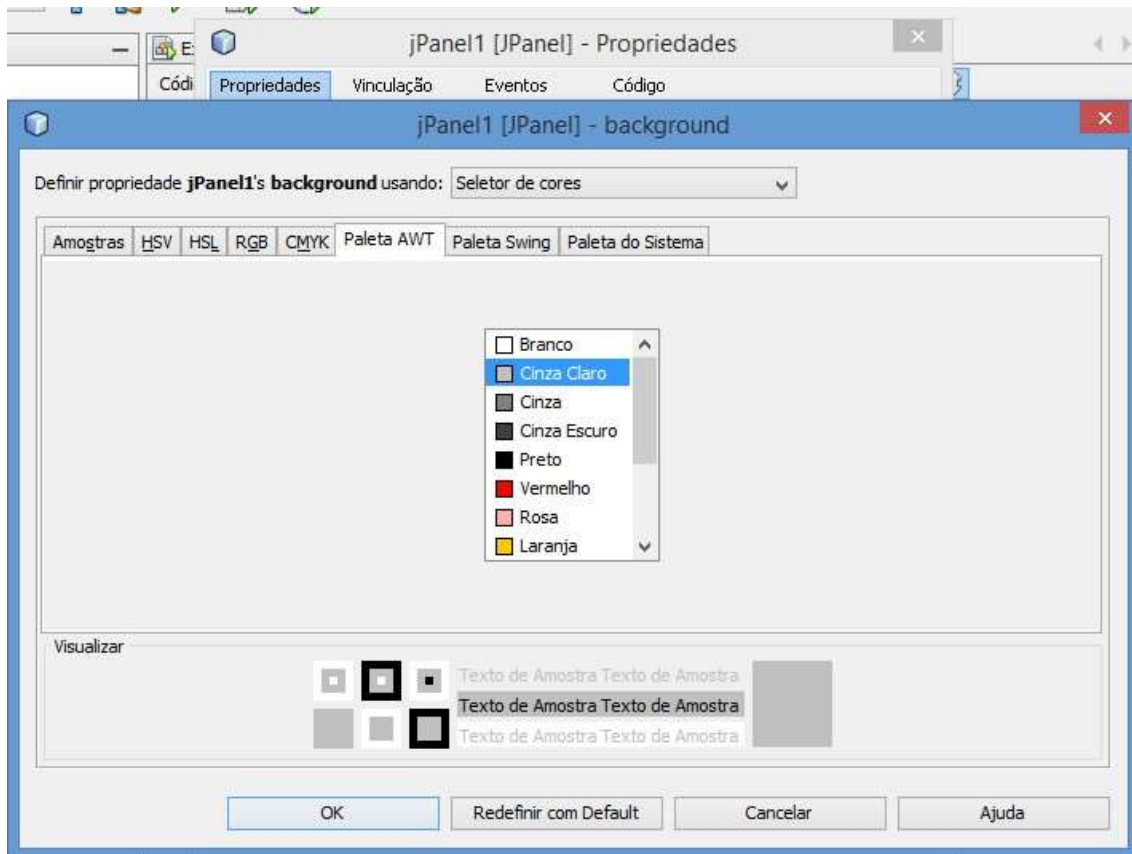


Figura 142 - Background Painel.

Agora vamos adicionar os componentes **JLabel** no nosso painel. Primeiro vamos visualizar na paleta a opção **Controle Swing** (destaque em vermelho) e nela selecionar o **Label** (destaque em preto) e arrastar até o painel, você irá arrastar até o painel dois labels, como na Figura 143 - Adicionando Label:

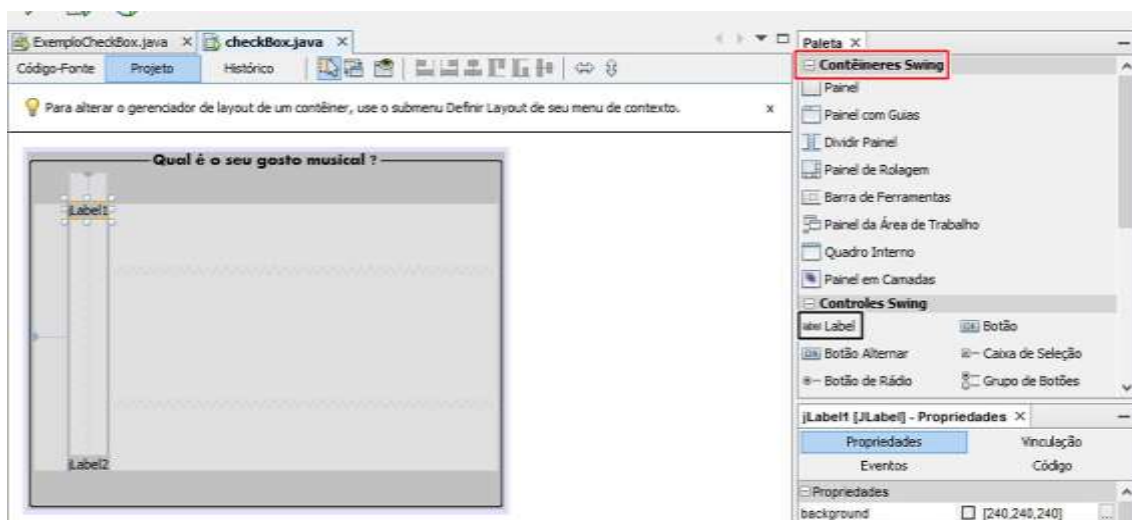


Figura 143 - Adicionando Label.

Para alterar o texto mostrado no componente selecione-o e clique com o botão direito do mouse, na nova aba clique em Editar Texto, Figura 144 - Editar Texto:

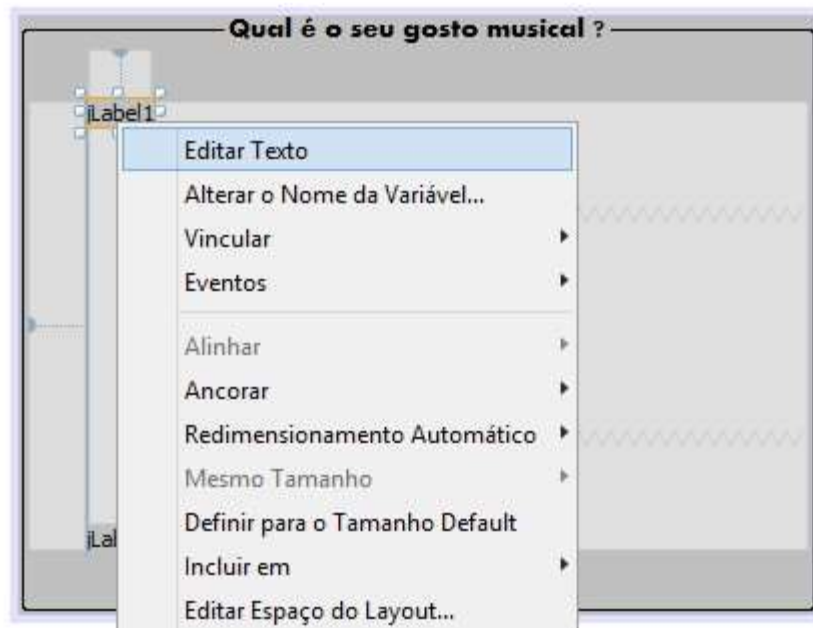


Figura 144 - Editar Texto.

O primeiro JLabel será alterado para: “Nome:”, no segundo JLabel você deve apagar todo o seu conteúdo e redimensionar para a mesma largura do frame (janela) para que possa mostrar o máximo possível de opções selecionadas, pois ele será usado apenas para saída das informações do usuário. Apesar de o label não aparecer no seu Form (Janela), você pode manipulá-lo através do navegador (destaque em vermelho), veja na Figura 145 - Navegador:

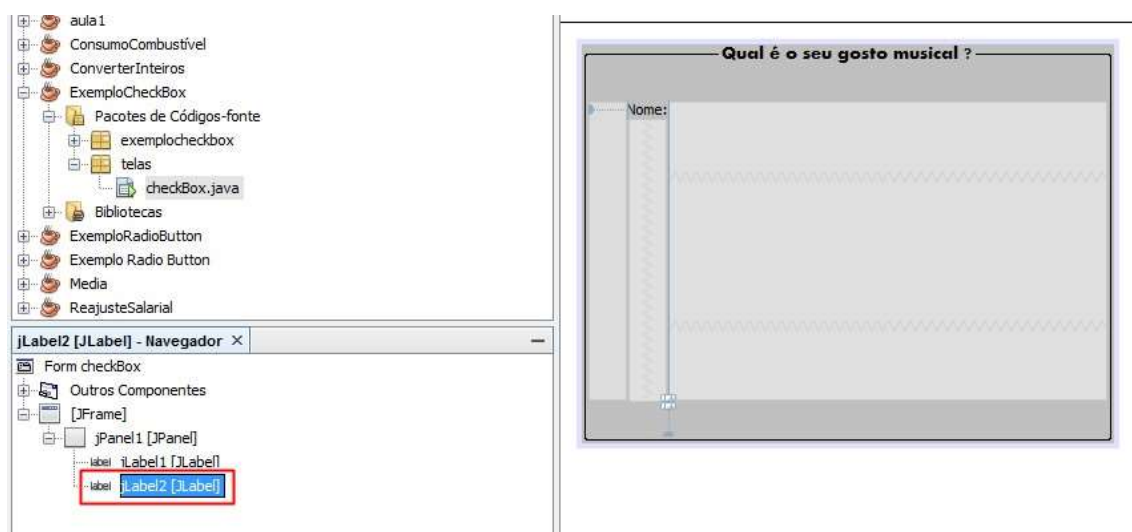


Figura 145 - Navegador.

No navegador selecione os dois JLabels, JLabel1 e JLabel2 (destaque em vermelho), e clique com o botão direito do mouse, em seguida abra as propriedades (destaque em preto), Figura 146 - Propriedades do JLabel. Dessa forma podemos manipular as propriedades dos dois labels. Isso é eficaz para o caso de querer mudar as mesmas propriedades em componentes iguais:

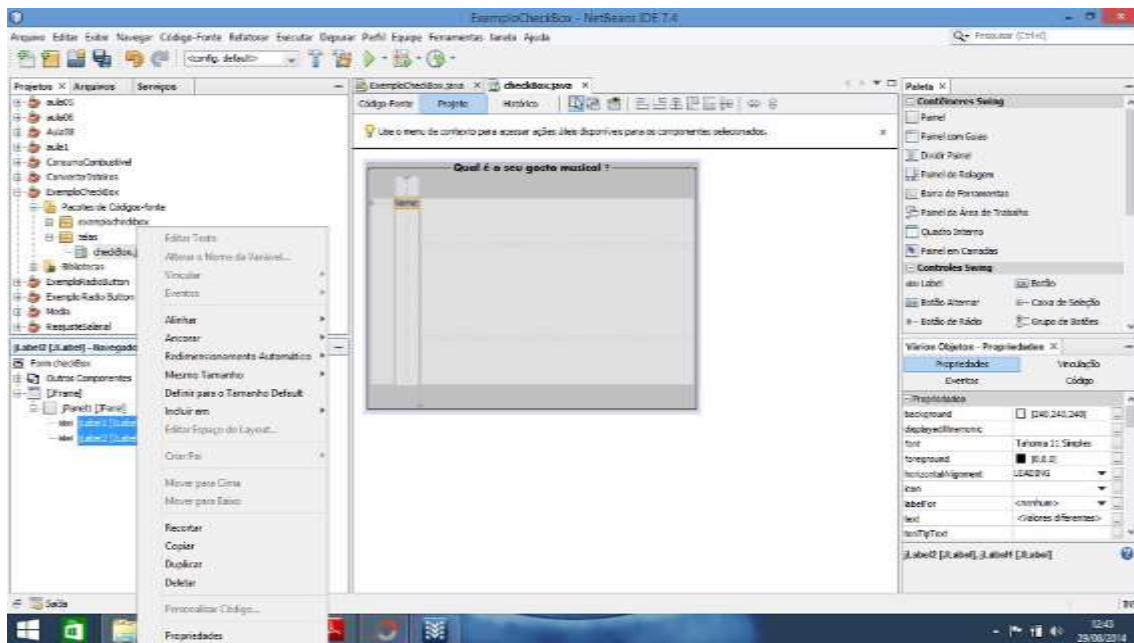


Figura 146 - Propriedades do JLabel.

Em seguida clique nas reticências na opção **font** você pode selecionar o estilo que quiser, neste exemplo usamos o tamanho da fonte: 14, estilo da fonte: negrito, e fonte: Tahoma, Figura 147 - Fonte do JLabel.

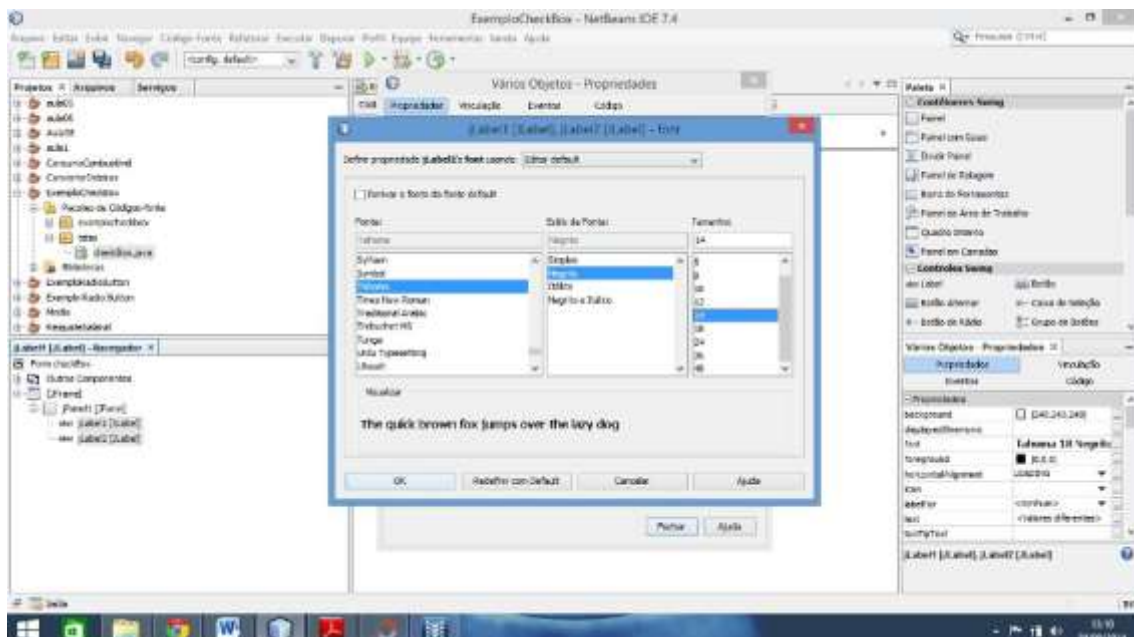


Figura 147 - Fonte do JLabel.

Próximo passo é adicionar um Campo de Texto - `TextArea` (destaque em vermelho) que será usado para receber o nome do usuário do aplicativo, Figura 148 - Adicionando Campo de Texto:

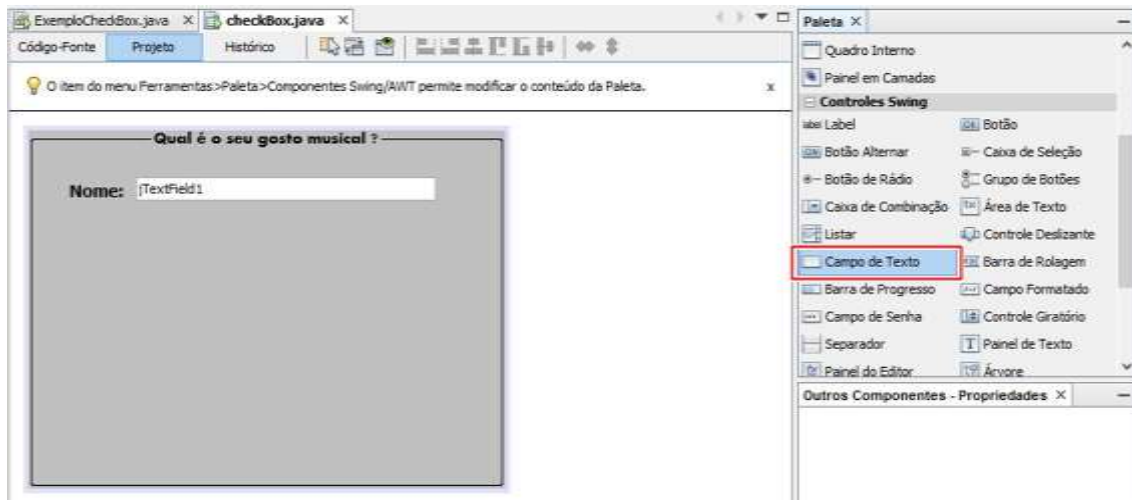


Figura 148 - Adicionando Campo de Texto.

Em seguida você irá acessar as propriedades do campo de texto adicionado, clique com o botão direito do mouse, na nova aba Editar Texto, você irá apagar o conteúdo que está escrito dentro campo de texto, Figura 149 - Editar Texto `TextField`:

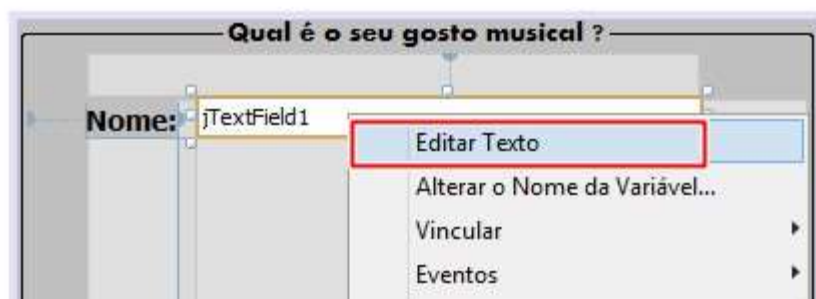


Figura 149 - Editar Texto `TextField`.

Agora vamos adicionar o componente que estamos estudando neste capítulo, a Caixa de Seleção - `CheckBox`, selecione uma de cada vez e arraste-as até o Frame (Janela), Figura 150 - Adicionando `CheckBox`:

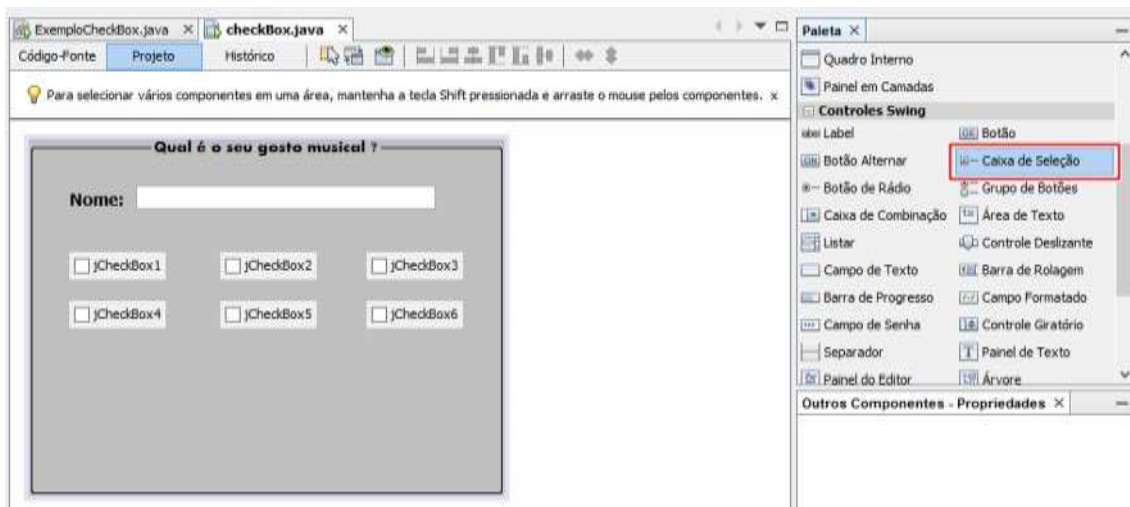


Figura 150 - Adicionando jCheckBox.

Agora vamos mudar os nomes das Caixas de Seleção para as opções de musicais que o usuário poderá selecionar, clique com o botão direito sobre o componente, depois clique em Editar Texto, e mude os nomes de todos as Caixas de Seleção, como na Figura 151 - Editar Texto jCheckBox:



Figura 151 - Editar Texto jCheckBox.

Agora vamos adicionar os botões (jButton) que enviaram as ações que o programa deve tomar, para isso vamos adicionar dois um para enviar e mostrar no programa o nome do usuário e seu gosto musical, e outro para limpar as informações adicionadas pelo usuário no programa.

Selecione um botão por ver, na **paleta**, **Controles Swing**. E arrastá-los até o painel. Figura 152 - Adicionando jButton:

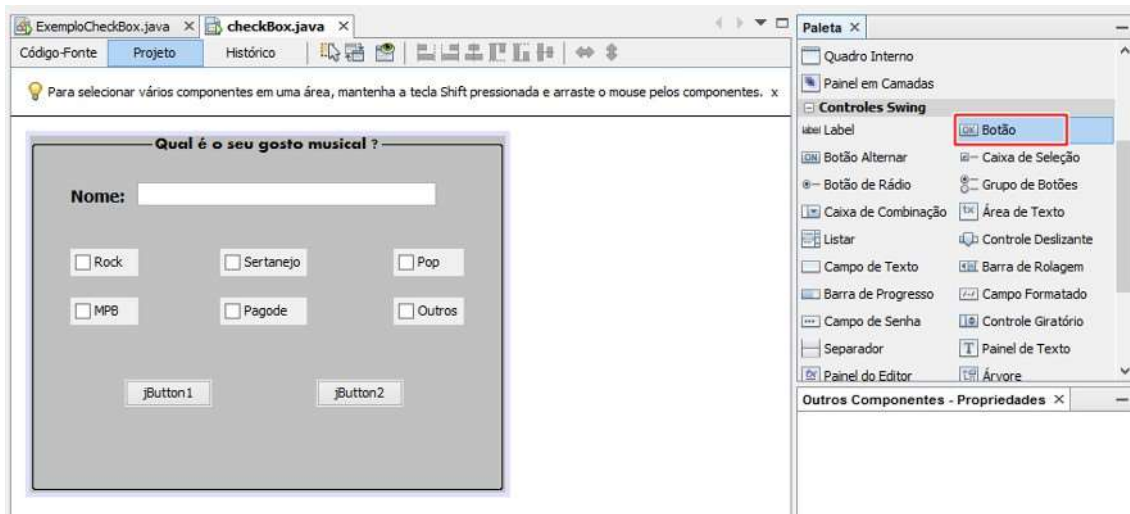


Figura 152 - Adicionando jButton.

Também vamos editar o texto dos botões, o primeiro será: “OK” e outro: “Limpar”, Figura 153 - Editar Texto jButton:



Figura 153 - Editar Texto jButton.

Vamos mudar os nomes das variáveis do programa, para ficar facilitar a codificação. Clique com o botão direito do mouse sobre o componente e clique em Alterar o Nome da Variável...

Os nomes das variáveis serão:

jTextField1 você mudará para jTextFieldNome;

jCheckBox1 para jCheckBoxRock;

jCheckBox2 para jCheckBoxSertanejo;

jCheckBox3 para jCheckBoxPop;

jCheckBox4 para jCheckBoxMpb;

jCheckBox5 para jCheckBoxPagode;

jCheckBox6 para jCheckBoxOutros;

jButton1 para jButtonOk;

jButton2 para jButtonLimpar;

jLabel2 para jLabelSaida.

Agora vamos adicionar o código do nosso programa para que depois que os usuários fizeram as seleções e clicar em OK aparecerá seu nome e seu gosto musical. Primeiro vamos selecionar o botão OK e clicar com o botão direito do mouse, na nova aba clique em **eventos**, **action**, **actionPerformed**, Figura 154 - actionPerformed botão OK.

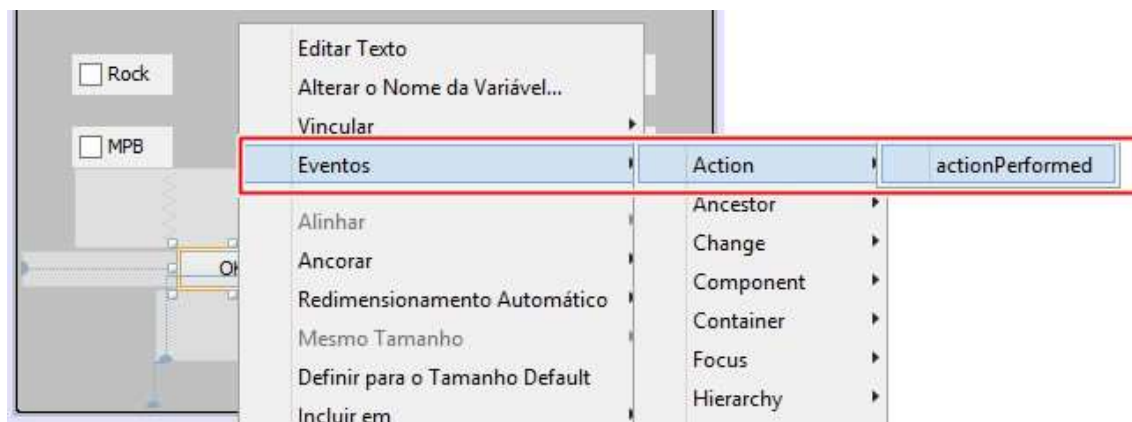


Figura 154 - actionPerformed botão OK.

Na hora de programar temos que testar todas as Caixas de Seleção para ver se alguma caixa esta selecionada, Figura 155 - Código botão OK.

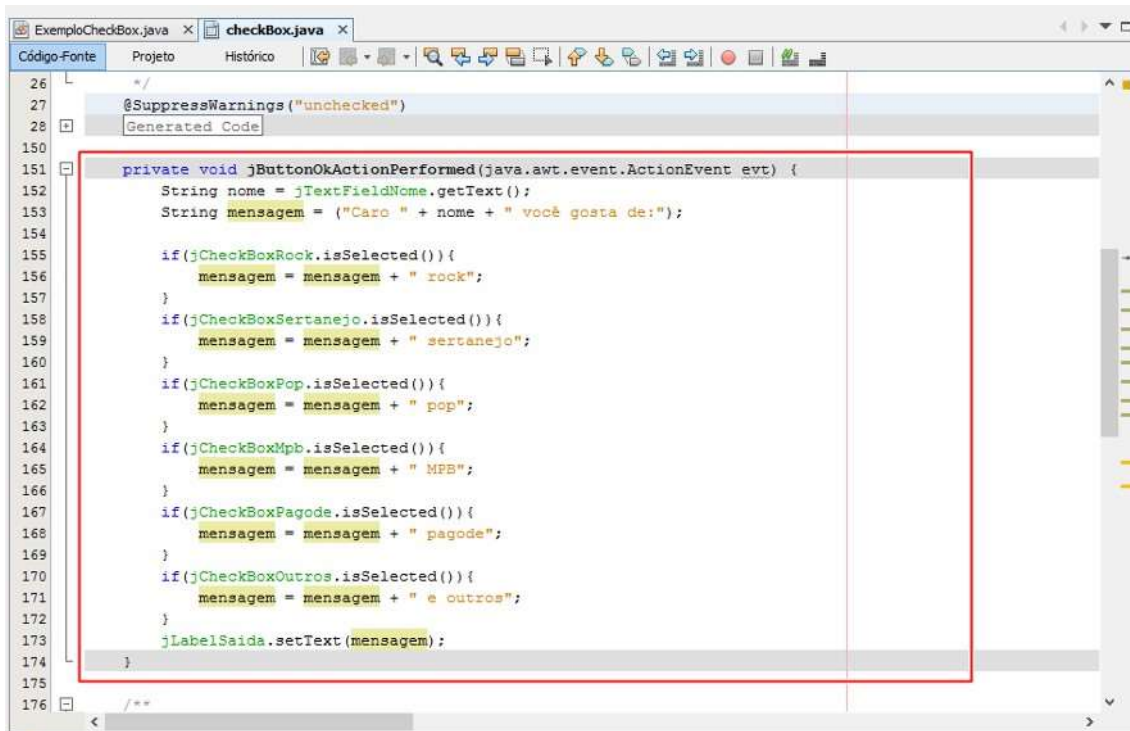


Figura 155 - Código botão OK.

Voltar para o projeto (destaque em vermelho), Figura 156 - Projeto, selecionar o botão Limpar e clicar com o botão direito do mouse, na nova aba clique em **eventos**, **action**, **actionPerformed**.

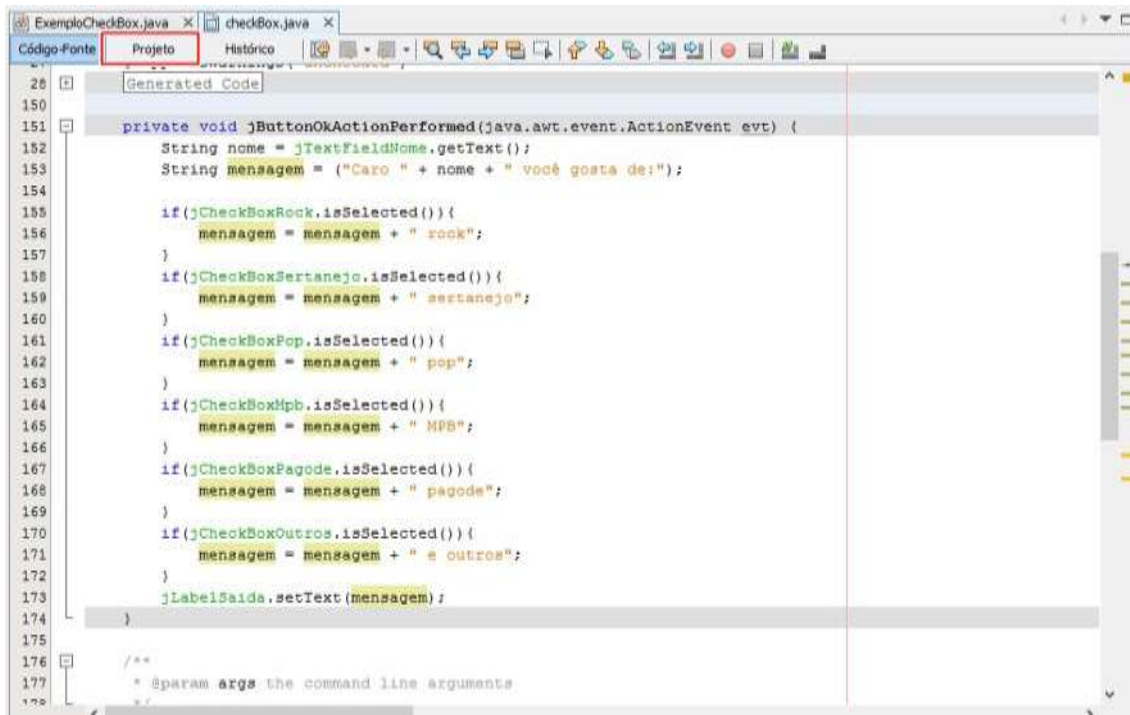


Figura 156 - Projeto.

Para codificar o botão Limpar você tem que informar que todas as caixas de seleção terão que assumir que não ficaram selecionadas, Figura 157 - Código do botão Limpar.

```
180
181 private void jButtonLimparActionPerformed(java.awt.event.ActionEvent evt) {
182     jLabelSaida.setText("");
183     jTextFieldNome.setText("");
184     jCheckBoxRock.setSelected(false);
185     jCheckBoxSertanejo.setSelected(false);
186     jCheckBoxPop.setSelected(false);
187     jCheckBoxMpb.setSelected(false);
188     jCheckBoxPagode.setSelected(false);
189     jCheckBoxOutros.setSelected(false);
190 }
191
192 /**
```

Figura 157 - Código do botão Limpar.

O seu programa terá a seguinte tela, figura 158, agora você pode testar para verificar se está tudo funcionando corretamente.

Qual é o seu gosto musical ?

Nome:

☐ Rock ☐ Sertanejo ☐ Pop

☐ MPB ☐ Pagode ☐ Outros

Figura 158 - Exemplo CheckBox.