

Programação Orientada a Objetos

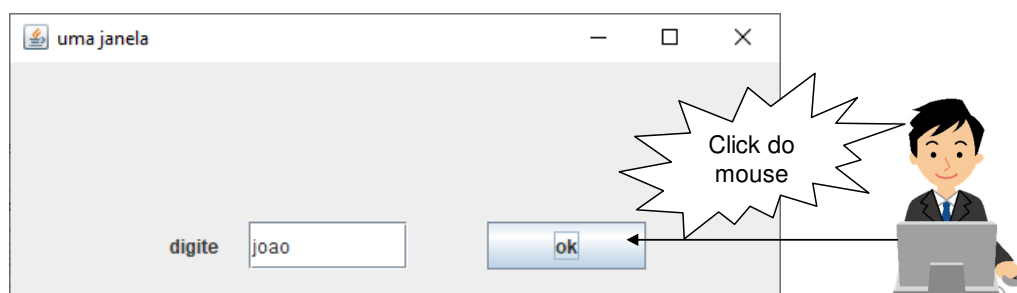
Fausto Maranhão Ayres

5

Aplicação gráfica (parte1)

Programação Orientada a Eventos

- Um programa é um conjunto de eventos
- Tipos de **eventos** que ocorrem durante a interação entre o usuário e a tela **comandam a execução** do programa:
 - Clicar na janela, botão, *checkbox*, etc
 - Digitar dentro de uma caixa de texto
 - Usar tecla de atalho



Biblioteca swing

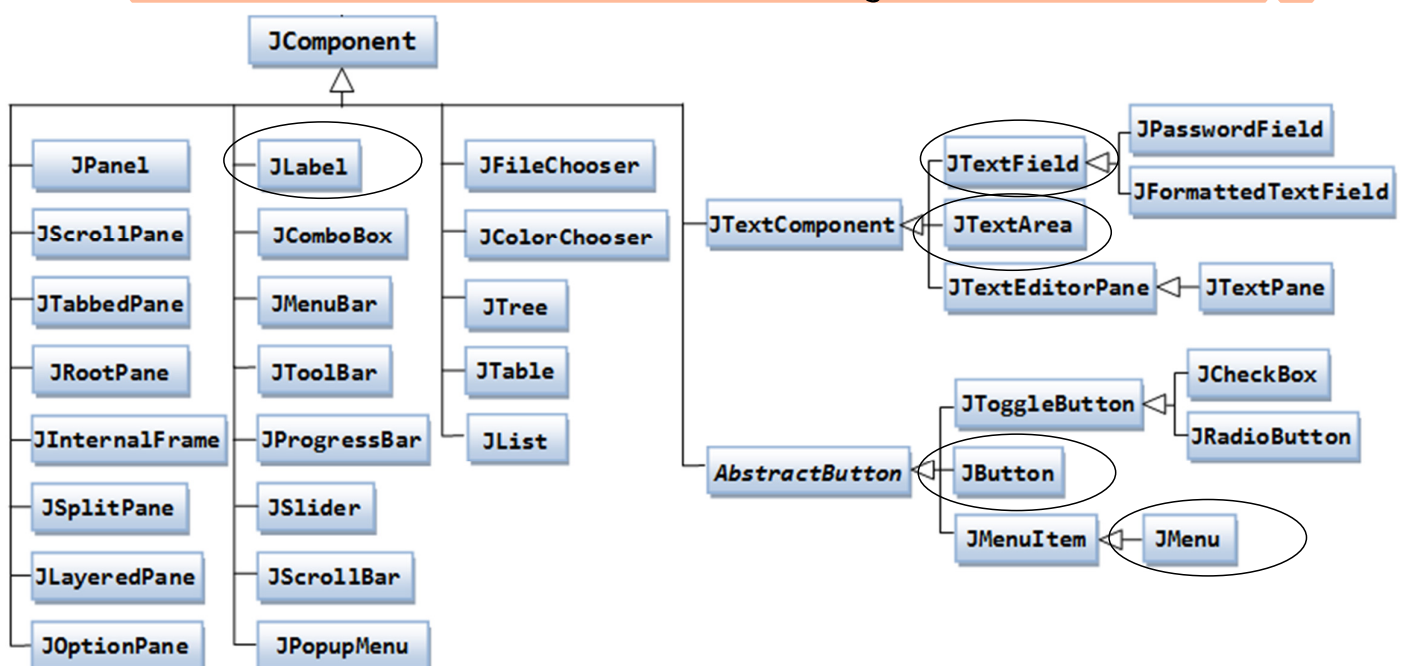
- Contém classes voltadas para a construção de tela desktop (Windows, Linux, MacOS)
- Cada classe deve ser importada com
`import javax.swing.nomeclasse;`
- Usaremos o plugin **WindowBuilder** do Eclipse como um assistente na construção das telas

fausto.ayres@ifpb.edu.br

3

Hierarquia Swing

- Os nomes das classes começam com a letra **J**



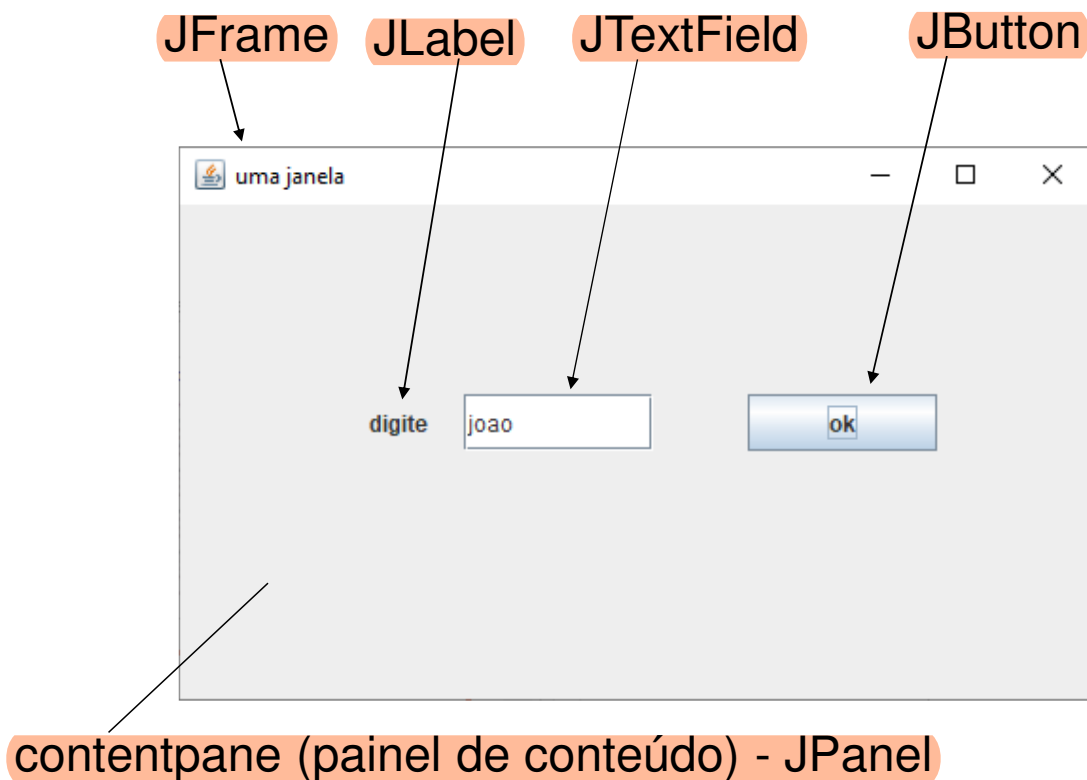
<https://docs.oracle.com/javase/tutorial/uiswing/components/index.html>

<http://web.mit.edu/6.005/www/sp14/psets/ps4/java-6-tutorial/components.html>

fausto.ayres@ifpb.edu.br

4

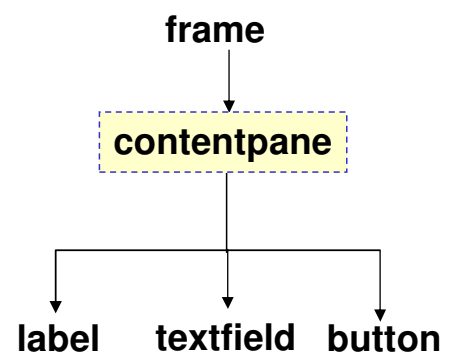
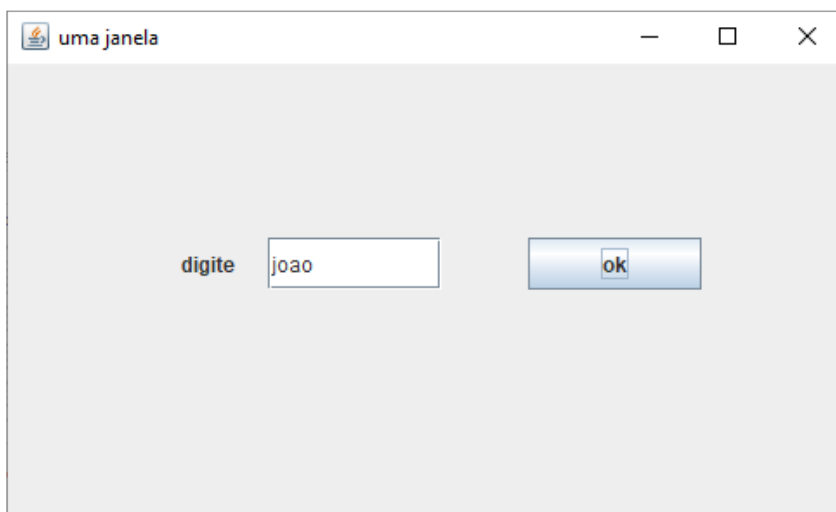
As principais classes do swing



fausto.ayres@ifpb.edu.br

5

Grupamento dos objetos



fausto.ayres@ifpb.edu.br

6

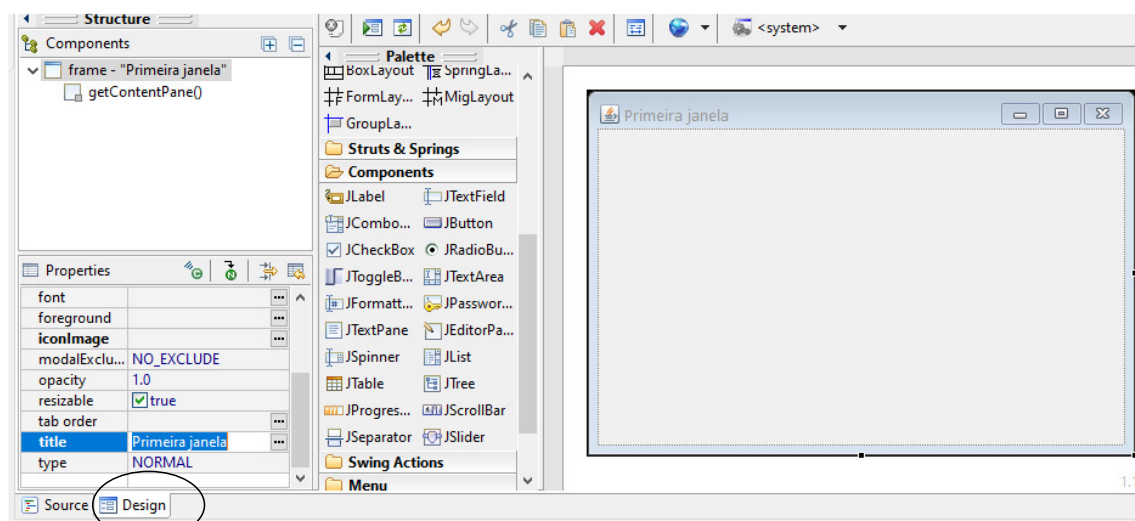
PLUGIN WINDOWBUILDER DO ECLIPSE

fausto.ayres@ifpb.edu.br

7

Plugin Windowbuilder

- Possui um **editor gráfico (aba Design)** que nos ajuda criar visualmente uma tela



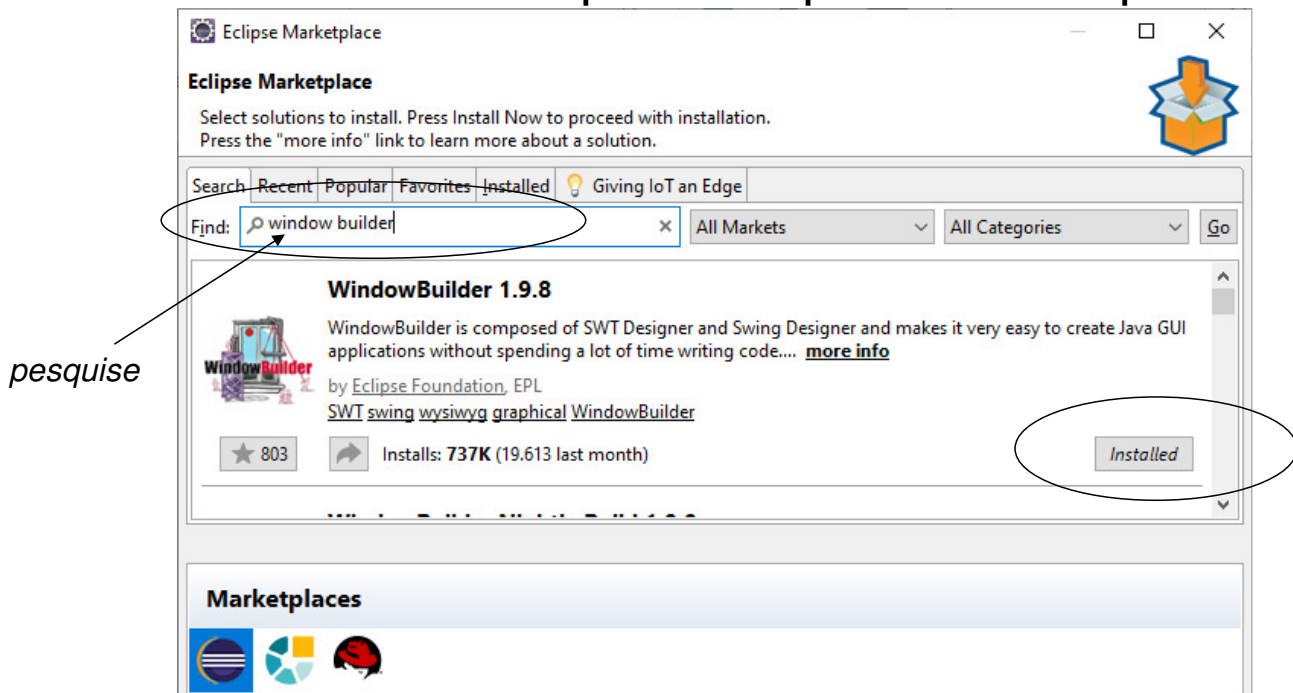
- Caso o eclipse não possua o plugin, ele deve ser instalado

fausto.ayres@ifpb.edu.br

8

Verificar se está instalado

- Acesse o menu Help/Marketplace do eclipse

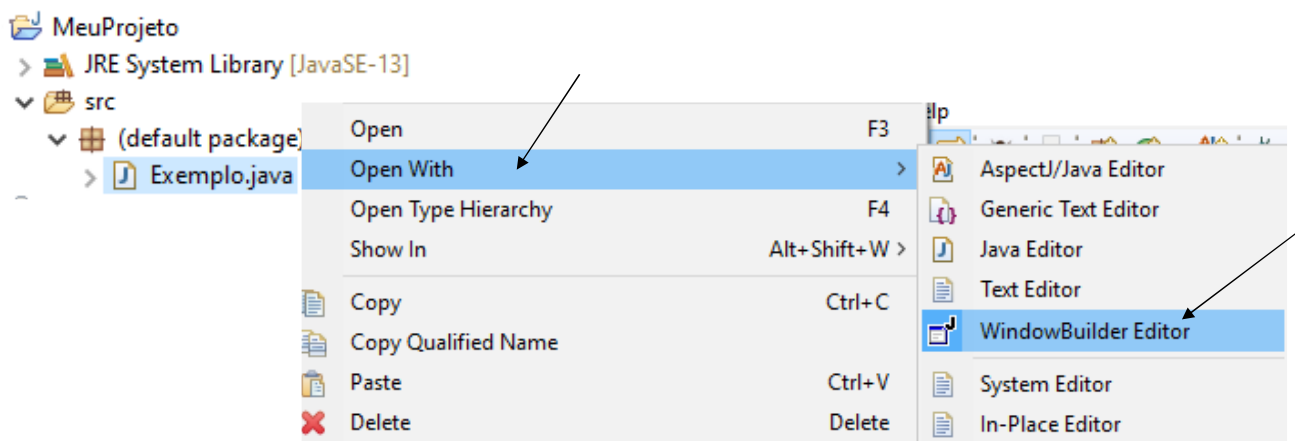


fausto.ayres@ifpb.edu.br

9

Editar um programa existente

- Ao clicar 2 vezes no programa existente, o eclipse abrirá automaticamente as **abas Source e Design** do editor Windowbuilder.
- Caso isto não ocorra, você poderá abri-las da seguinte forma:

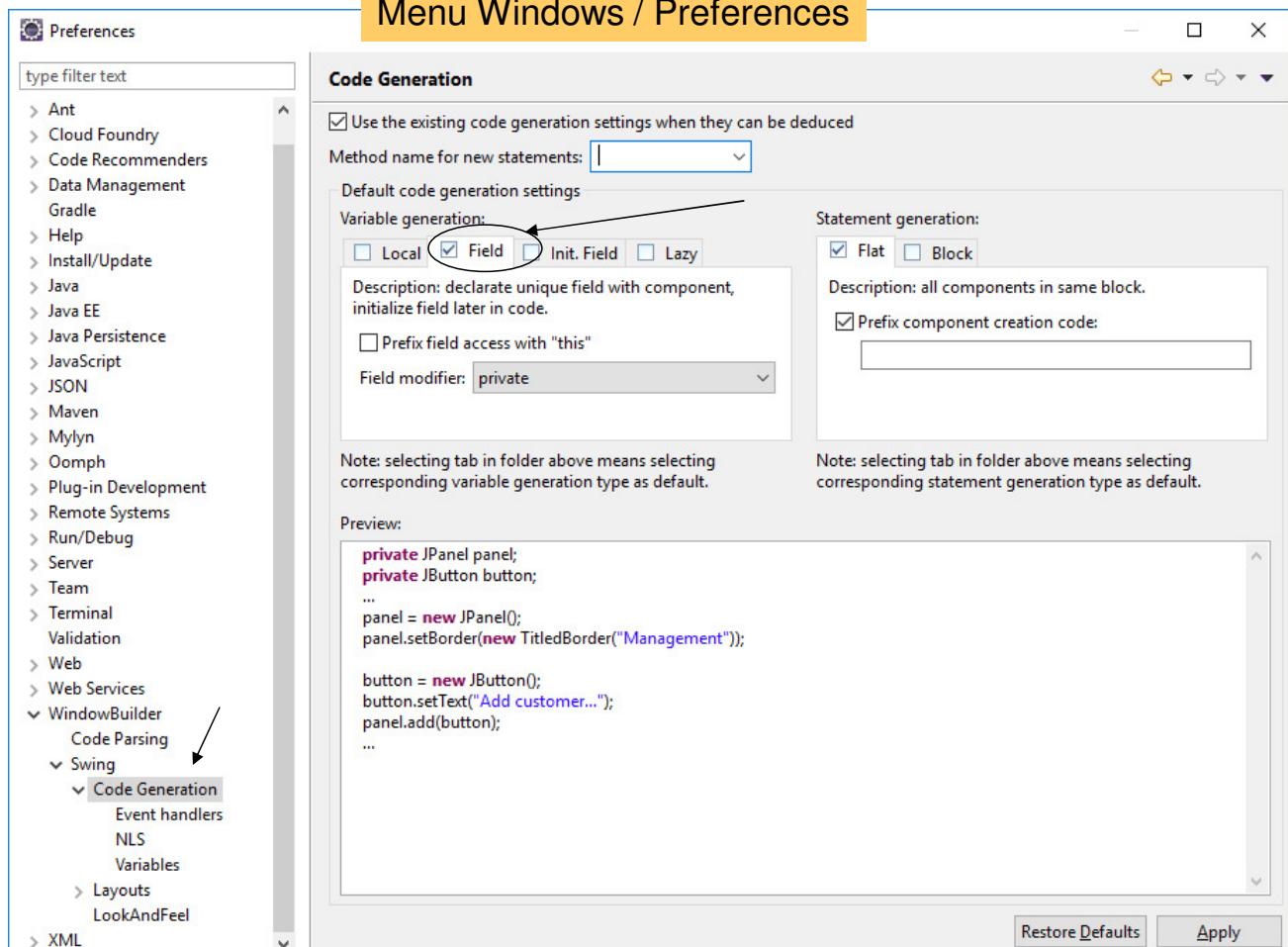


fausto.ayres@ifpb.edu.br

10

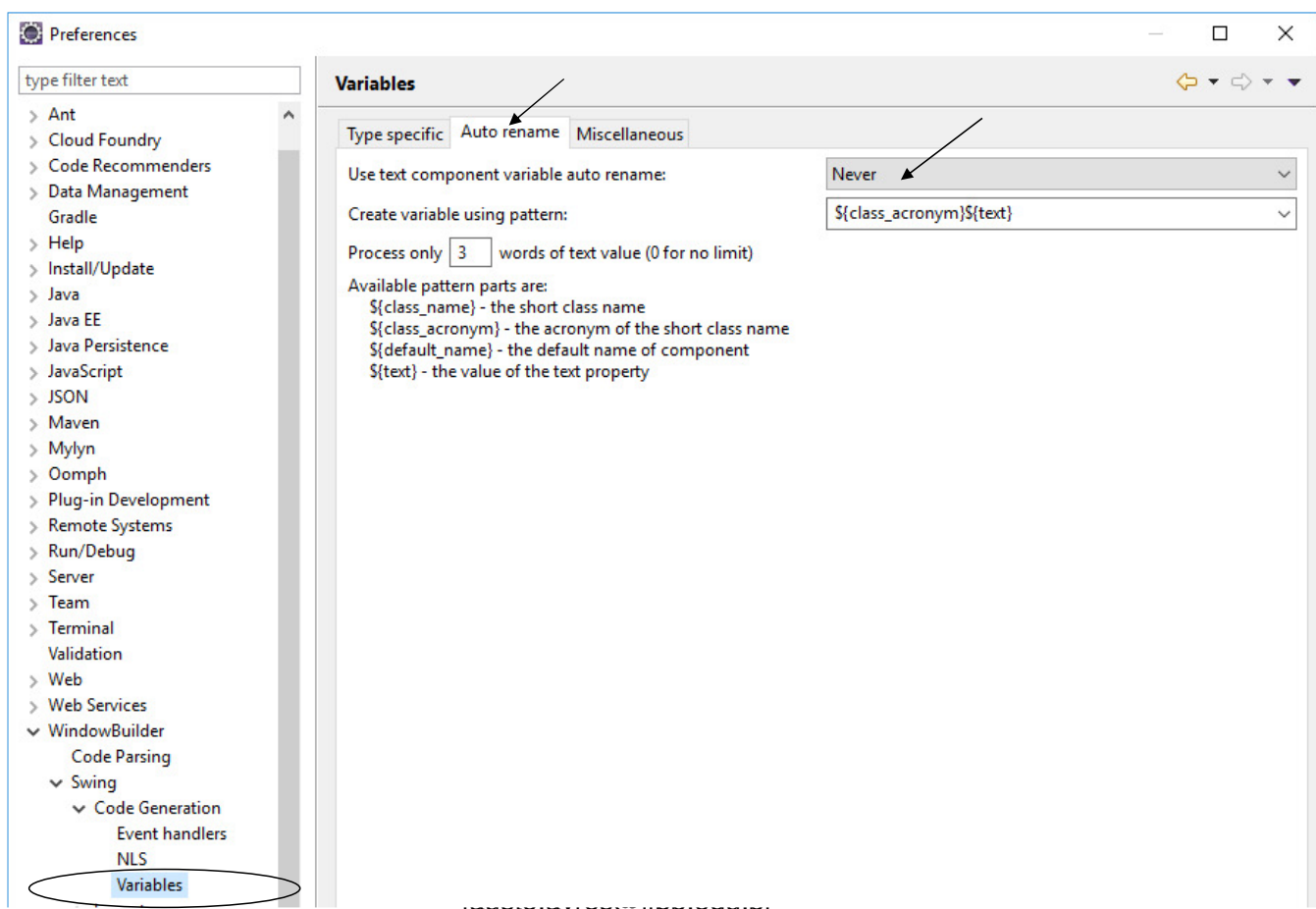
Configurar o editor windowbuilder (1)

Menu Windows / Preferences



11

Configurar o editor windowbuilder (2)



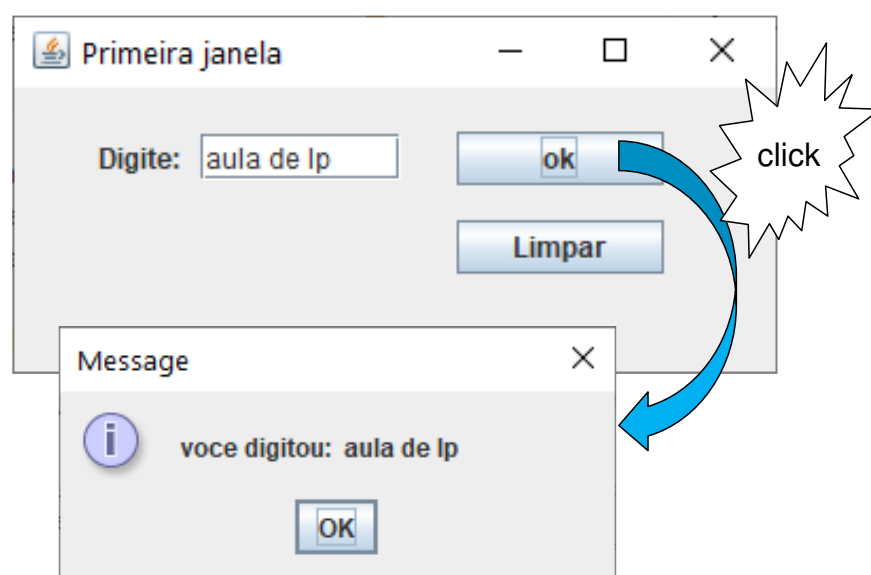
12

EXEMPLO 1

fausto.ayres@ifpb.edu.br

13

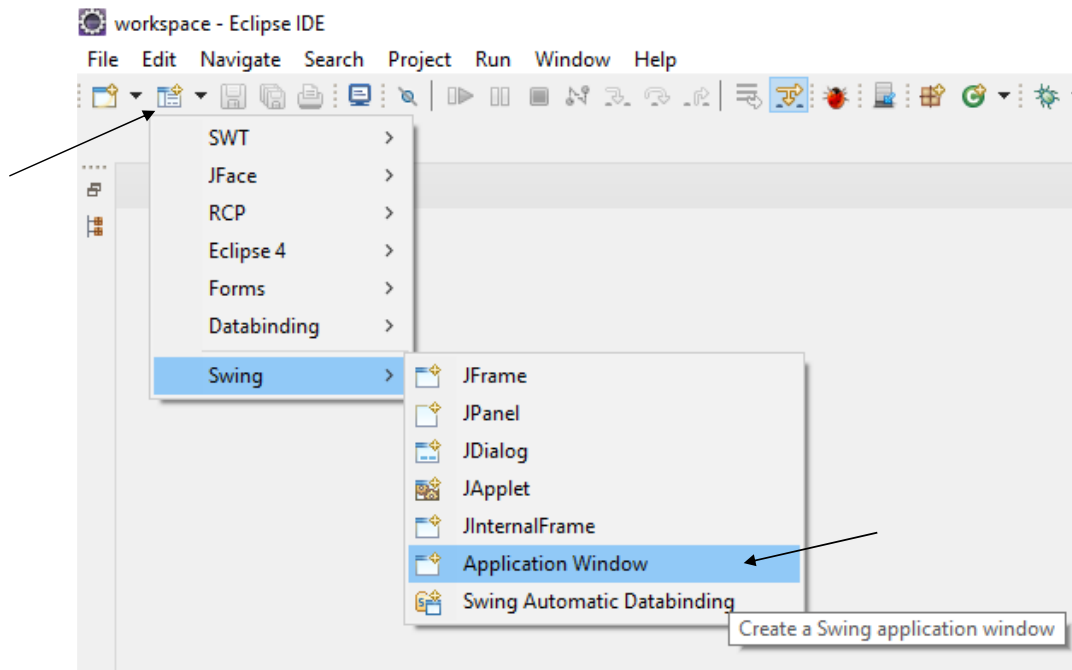
Exemplo.java



fausto.ayres@ifpb.edu.br

14

Criando o programa com o plugin



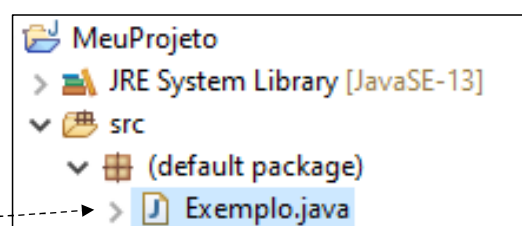
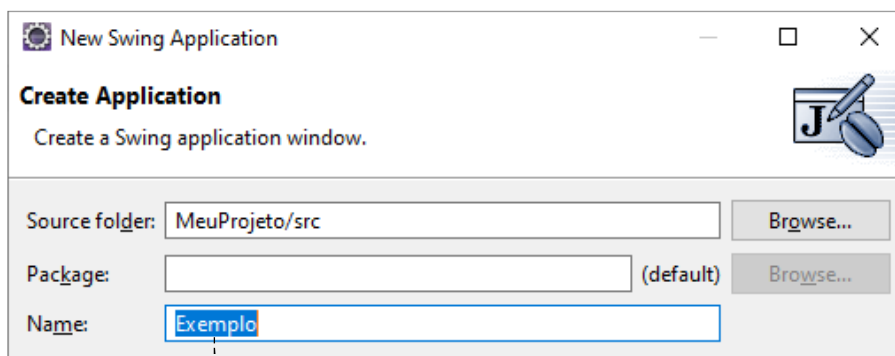
ou selecione o Menu:

File / New / Other / WindowBuider / Swing Designer / Application Window

fausto.ayres@ifpb.edu.br

15

Criando o programa com o plugin



fausto.ayres@ifpb.edu.br

16

As abas Source e Design

Variáveis do programa

propriedades do objeto selecionado

Sincronização: Tudo que é alterado na aba **Source** é alterado também na aba **Design** e vice-versa

MeuProjeto

- JRE System Library [JavaSE-13]
- src
 - (default package)
 - Exemplo.java

Design

```
1 import java.awt.EventQueue;
2
3 public class Exemplo {
4
5     private JFrame frame;
6
7     /**
8      * Launch the application.
9      */
10    public static void main(String[] args) {
11        EventQueue.invokeLater(new Runnable() {
12            public void run() {
13                try {
14                    Exemplo window = new Exemplo();
15                    window.frame.setVisible(true);
16                } catch (Exception e) {
17                    e.printStackTrace();
18                }
19            }
20        });
21    }
22 }
```

Source

fausto.ayres@ifpb.edu.br

17

Passo 1 - Configurar a janela

propriedades do objeto JFrame

Alterar o título da janela

Structure

- Components
 - frame - "Primeira janela"
 - getContentPane()

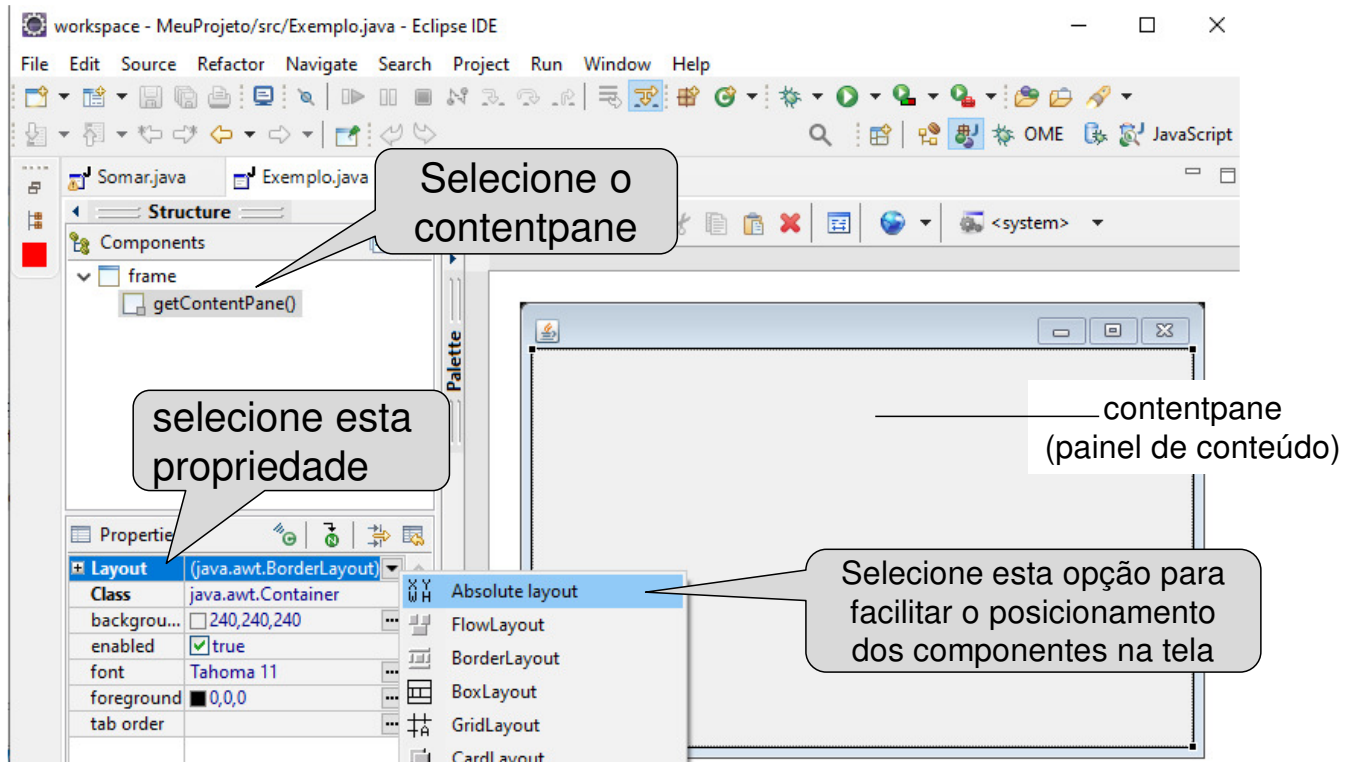
Properties

font	
foreground	
iconImage	
modalExclu...	NO_EXCLUDE
opacity	1.0
resizable	<input checked="" type="checkbox"/>
tab order	
title	Primeira janela
type	NORMAL

Palette

- BoxLayout
- SpringLa...
- FormLay...
- MigLayout
- GroupLa...
- Struts & Springs
- Components
 - JLabel
 - JTextField
 - JCombo...
 - JButton
 - JCheckBox
 - JRadioBu...
 - JToggleB...
 - JTextArea
 - JFormatt...
 - JPasswor...
 - JTextPane
 - JEditorPa...
 - JSpinner
 - JList
 - JTable
 - JTree
 - JProgres...
 - JScrollbar
 - JSeparator
 - JSlider
- Swing Actions
- Menu

Passo 2 – Configurar o painel de conteúdo



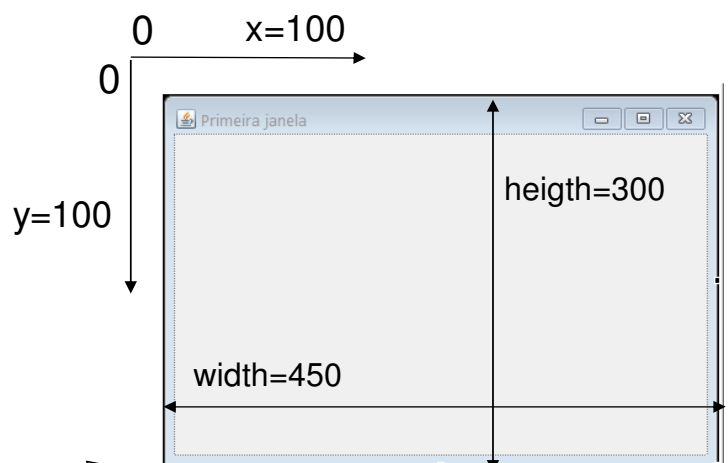
fausto.ayres@ifpb.edu.br

19

Ver o código fonte (aba Source)

Sincronização:

Tudo que é alterado na aba **Source** é alterado também na aba **Design** e vice-versa



```
private void initialize() {  
    frame = new JFrame();  
    frame.setTitle("Primeira janela");  
    frame.setBounds(100, 100, 450, 300);  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.getContentPane().setLayout(null);  
}
```

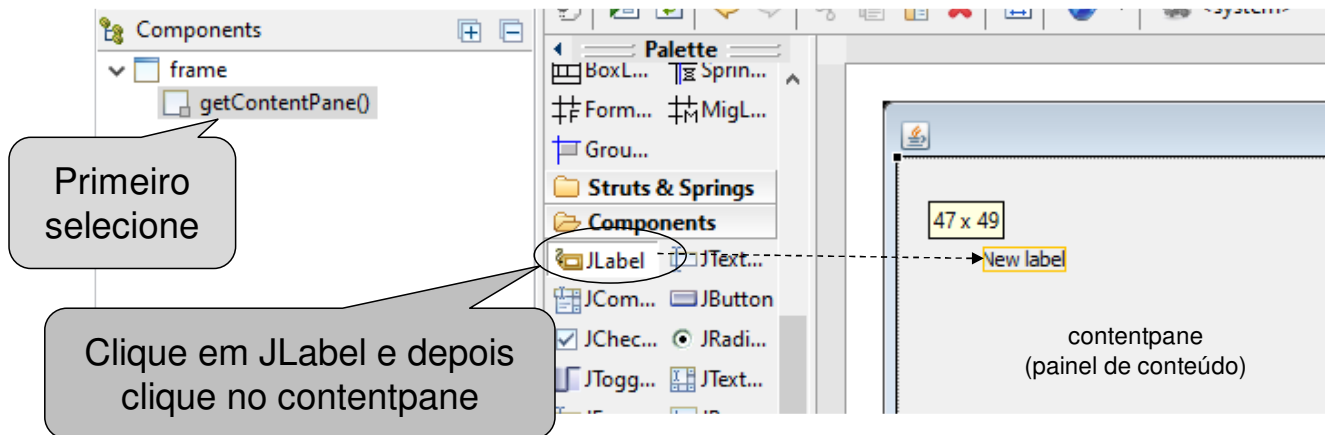
layout do painel de conteúdo

fausto.ayres@ifpb.edu.br

20

Passo 3 - Adicionar um label

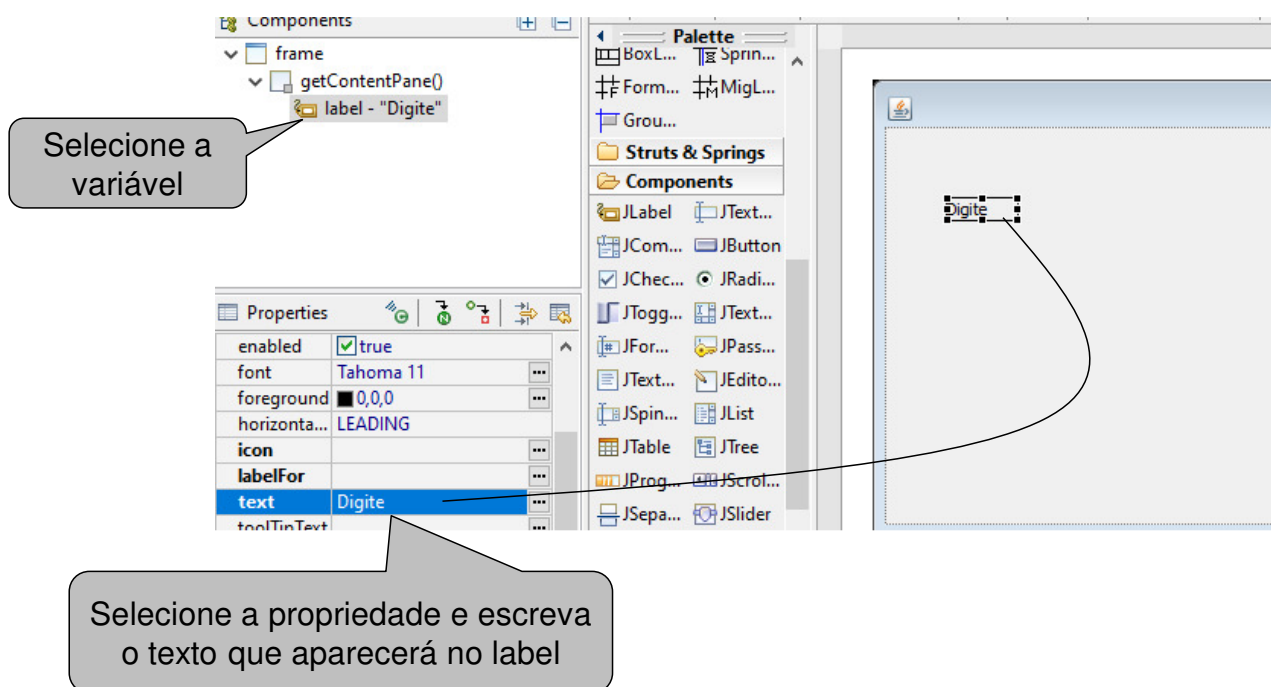
- Abrir a aba Components da Palette
- Inserir um objeto JLabel na janela



fausto.ayres@ifpb.edu.br

21

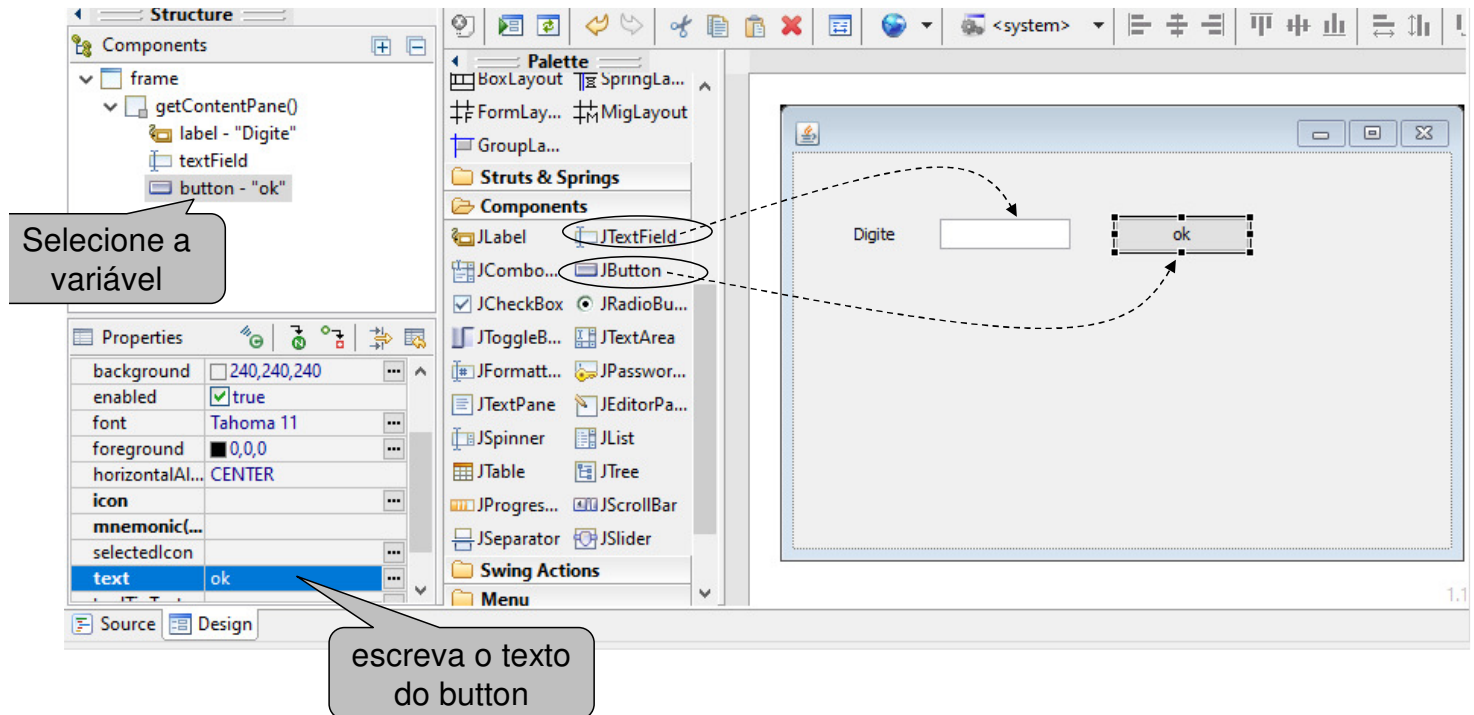
Escrever o texto dentro do label



fausto.ayres@ifpb.edu.br

22

Passo 4 - Adicionar textfield e button



fausto.ayres@ifpb.edu.br

23

Ver o código fonte (aba Source)

Observe que o plugin criou automaticamente as variáveis e os objetos das classes `JFrame`, `JLabel`, `JButton` e `JTextField` e configurou as suas propriedades usando métodos `set()`

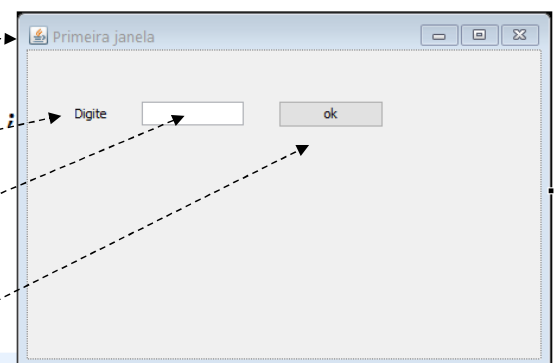
```
private void initialize() {  
    frame = new JFrame();  
    frame.setTitle("Primeira janela");  
    frame.setBounds(100, 100, 450, 300);  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.getContentPane().setLayout(null);  
  
    label = new JLabel("Digite");  
    label.setBounds(41, 47, 46, 14);  
    frame.getContentPane().add(label);  
  
    textField = new JTextField();  
    textField.setBounds(97, 44, 86, 20);  
    frame.getContentPane().add(textField);  
    textField.setColumns(10);  
  
    button = new JButton("ok");  
    button.setBounds(212, 43, 89, 23);  
    frame.getContentPane().add(button);  
}
```

objeto

objeto

objeto

objeto



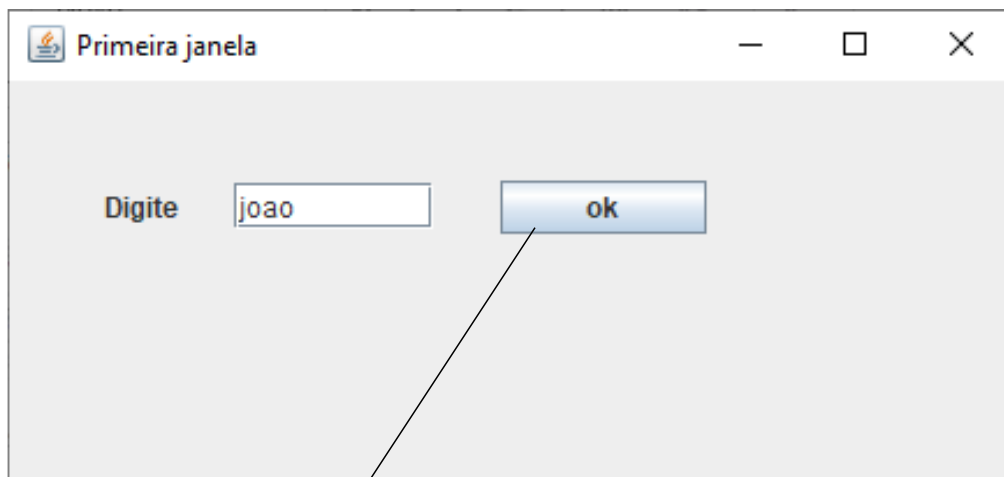
Atributos da aplicação

```
public class Exemplo {  
    private JFrame frame;  
    private JLabel label;  
    private JTextField textField;  
    private JButton button;  
}
```

fausto.ayres@ifpb.edu.br

24

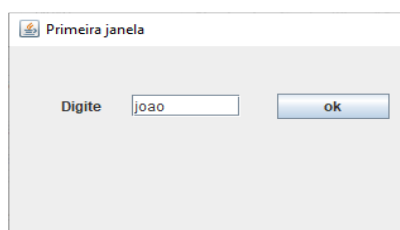
Executar o programa (Run)



Ainda falta programar o click no botão

Action Listener (Ouvinte de Ação)

- Um ouvinte de ação contém a ação a ser executada pelo windows, quando clicar o mouse



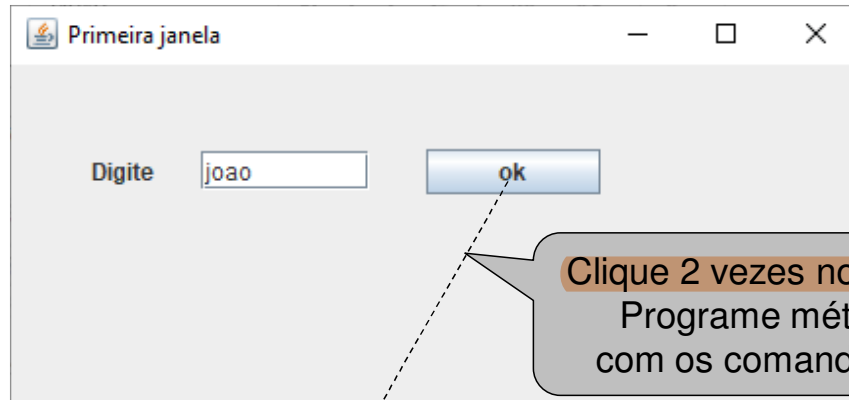
Ouvinte de ação

- Cada *button* possui seu ouvinte de ação

```
button = new JButton("ok");  
button.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        Programar a ação a ser executada...  
    }  
});
```

Criar um ActionListener

Passo 5 – Programar ouvinte de ação



Clique 2 vezes no button para criar o ouvinte
Programa método **actionPerformed()**
com os comandos que serão executados

```
button.addActionListener(  
    new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            String mensagem = "vc escreveu: " + textField.getText() ;  
            JOptionPane.showMessageDialog(null, mensagem);  
        }  
    }  
);
```

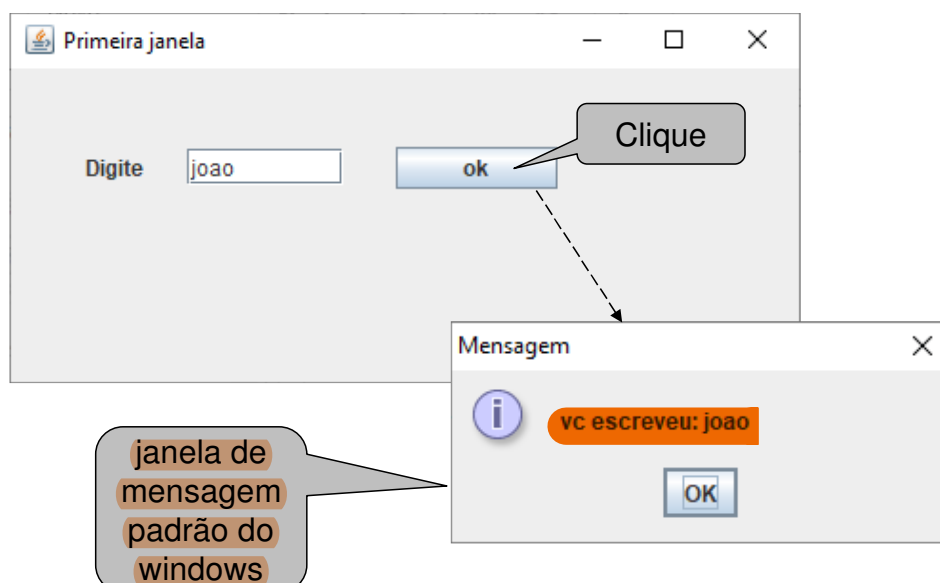
Abrir uma
janela de
mensagem
padrão

acessar o texto
que está dentro
do textfield

fausto.ayres@ifpb.edu.br

27

Executar o programa (Run)



fausto.ayres@ifpb.edu.br

28

Passo 6 - Adicionar um novo button

Observe que a nova variável foi numerada automaticamente

Components

- frame - "Primeira janela"
 - getContentPane()
 - label - "Digite"
 - textField
 - button - "ok"
 - button_1 - "Limpar"

Properties

background	240,240,240
enabled	<input checked="" type="checkbox"/>
font	Tahoma 11
foreground	0,0,0
horizontalAl...	CENTER
icon	
mnemonic(...	
selectedIcon	
text	Limpar
toolTipText	

Palette

- BoxLayout
- SpringLa...
- FormLay...
- MigLayout
- GroupLa...
- Struts & Springs
- Components
 - JLabel
 - JTextField
 - JCombo...
 - JButton
 - JCheckBox
 - JRadioBu...
 - JToggleB...
 - JTextArea
 - JFormat...
 - JPasswor...
 - JTextPane
 - JEditorPa...
 - JSpinner
 - JList
 - JTable
 - JTree
 - JProgres...
 - JScrollBar
 - JSeparator
 - JSlider
 - Swing Actions
 - Menu

Primeira janela

Digite [] [ok]

Limpar

tarefa é limpar e mover o cursor para o textfield

Escreva o texto do button_1

1.10.1.20200

fausto.ayres@ifpb.edu.br

29

Passo 7 - Programar ouvinte de ação

Primeira janela

Digite [] [ok]

Limpar

Clique 2 vezes no button_1 para criar o ouvinte Programme o método actionPerformed()

```
button_1.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        textField.setText("");  
        textField.requestFocus();  
    }  
});
```

setText() altera o texto e requestFocus() solicita o cursor

fausto.ayres@ifpb.edu.br

30

Manipulação das propriedades

- As propriedades dos componentes podem ser manipuladas via métodos **get** / **set**

Exemplos:

```
frame.setTitle("Primeira janela");
label.setText("Digite uma palavra");
button.setText("ok");
textField.setText(""); //vazio





String titulo = frame.getTitle();
String s1 = label.getText(); // "Digite..."
String s2 = button.getText(); // "ok"
```

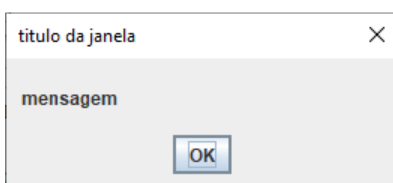
fausto.ayres@ifpb.edu.br

31

Customizando a Janela de mensagem

```
JOptionPane.showMessageDialog(null, "mensagem",  
"titulo da janela", JOptionPane.PLAIN_MESSAGE);
```

JOptionPane.ERROR_MESSAGE		Exibe um diálogo que indica um erro para o usuário.
JOptionPane.INFORMATION_MESSAGE		Exibe um diálogo com uma mensagem de informação para o usuário. O usuário poderá simplesmente dispensar o diálogo
JOptionPane.WARNING_MESSAGE		Exibe um diálogo que adverte o usuário sobre um problema em potencial.
JOptionPane.QUESTION_MESSAGE		Exibe um diálogo que impõe uma pergunta para o usuário. Este diálogo normalmente exige uma resposta, Tal como clicar em um botão Yes ou No



O parâmetro **null** pode ser substituído pela janela principal (frame). Neste caso, a mensagem será exibida no centro da janela principal

fausto.ayres@ifpb.edu.br

32

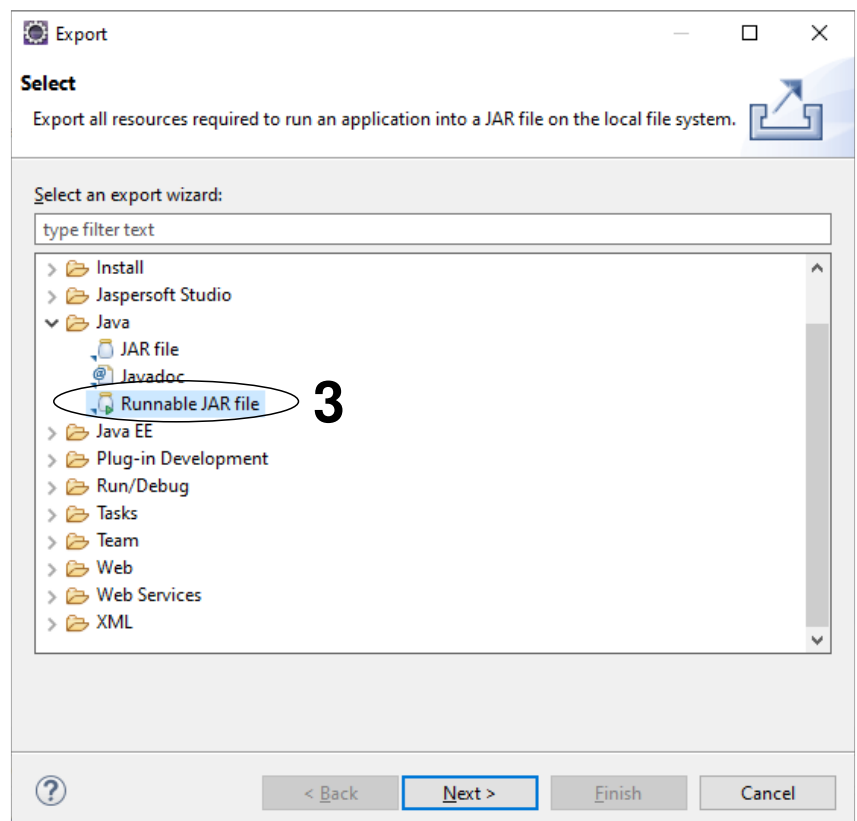
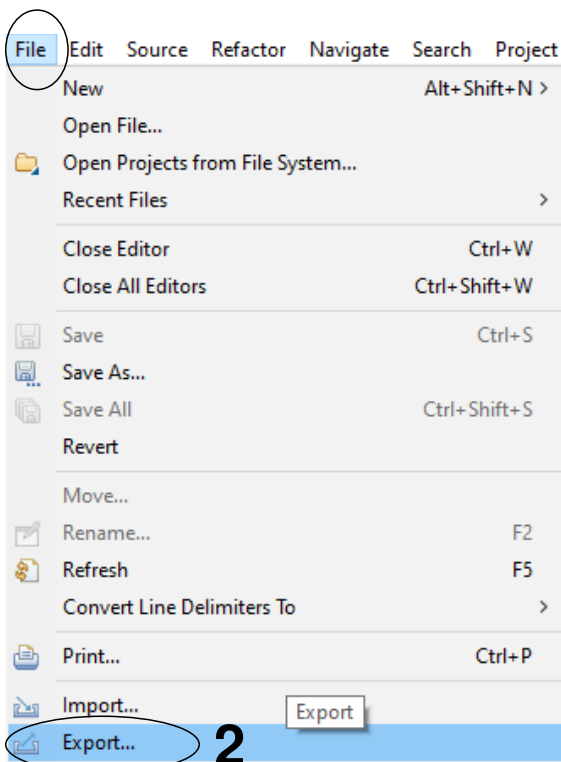
CRIANDO UM ARQUIVO EXECUTÁVEL PARA WINDOWS / LINUX / MAC

fausto.ayres@ifpb.edu.br

33

Menu File / Export / Runnable Jar file

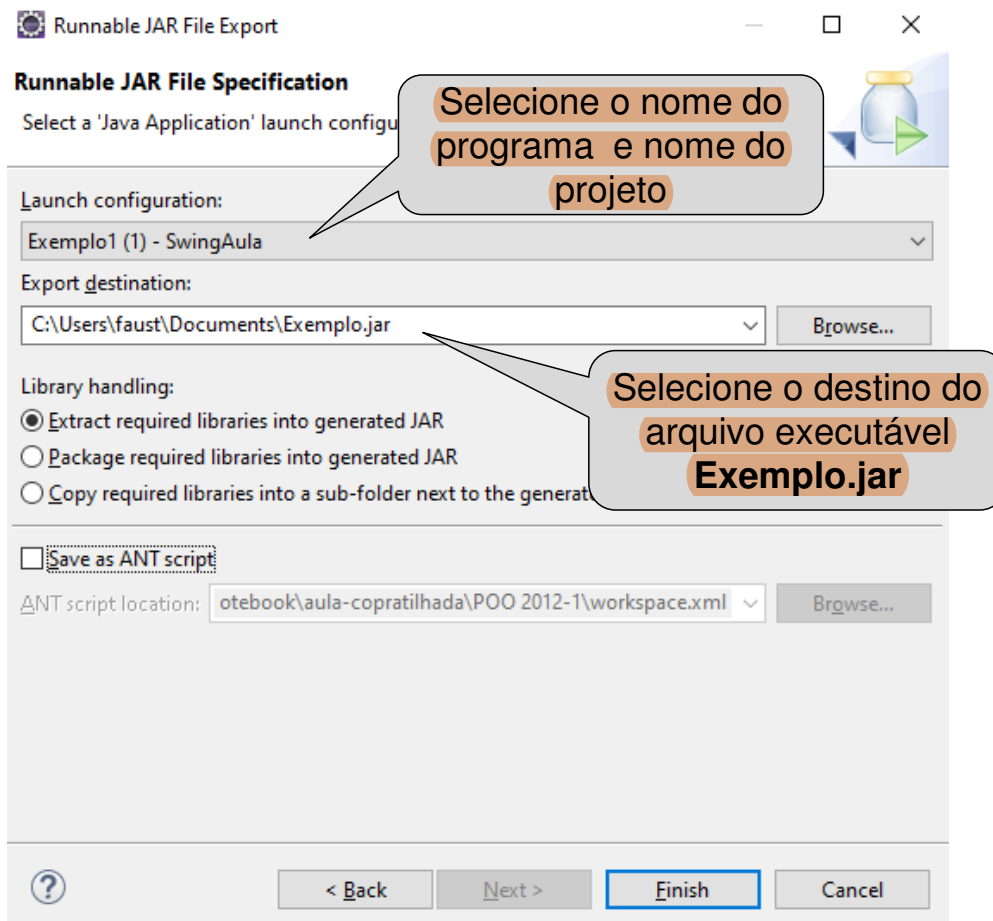
1



fausto.ayres@ifpb.edu.br

34

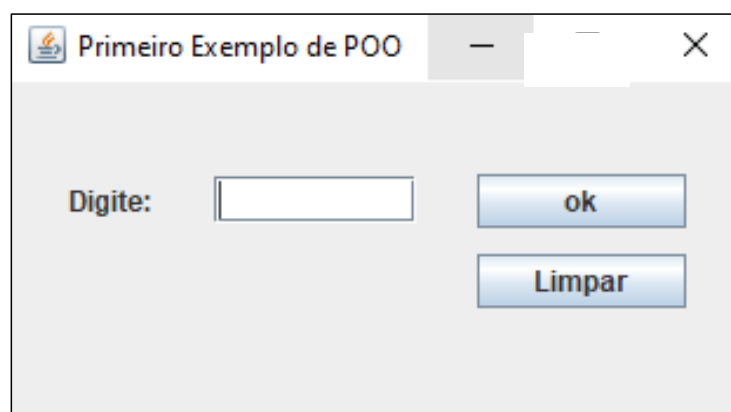
Configuração



35

Executando o programa no windows

- Localize o arquivo **Exemplo.jar** exportado
- Clique com botão direito do mouse no arquivo



36

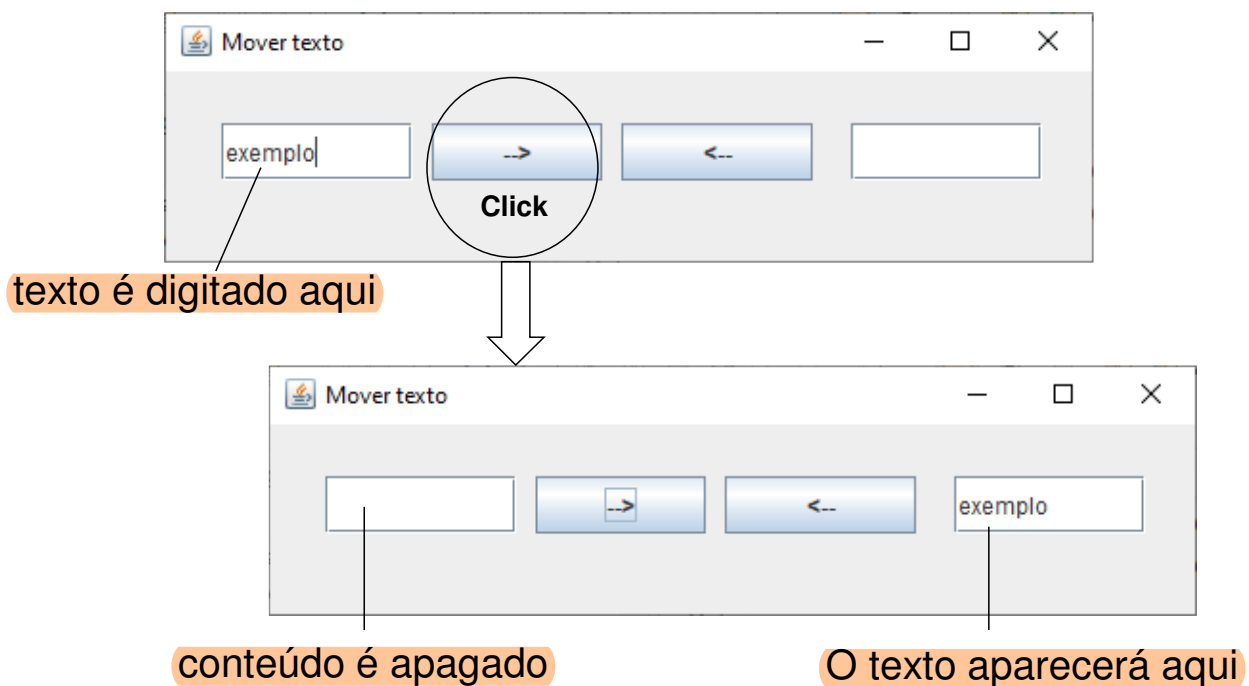
EXEMPLO 2

fausto.ayres@ifpb.edu.br

37

MoverTexto.java

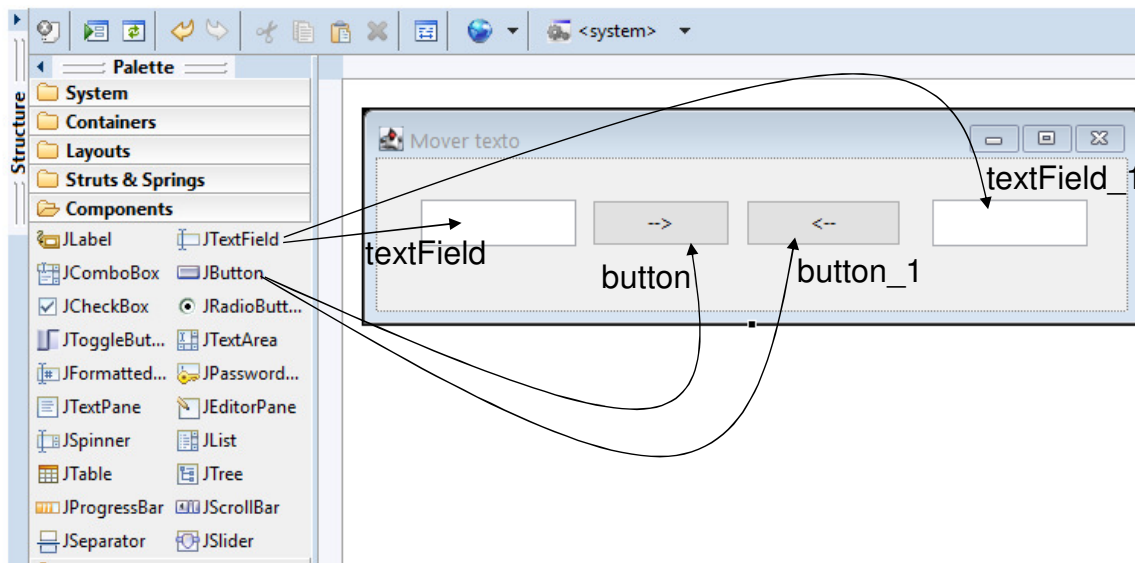
- Mover texto entre dois *textFields* (nos dois sentidos)



fausto.ayres@ifpb.edu.br

38

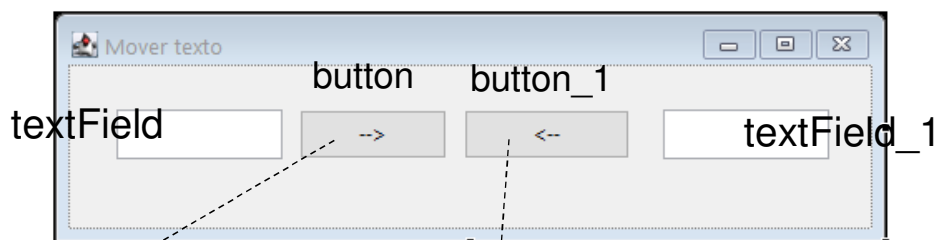
Adicionar 4 componentes



fausto.ayres@ifpb.edu.br

39

Programar os dois ouvintes de ação



```
button.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        textField_1.setText(textField.getText());  
        textField.setText("");  
    }  
});  
  
button_1.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        textField.setText(textField_1.getText());  
        textField_1.setText("");  
    }  
});
```

fausto.ayres@ifpb.edu.br

40

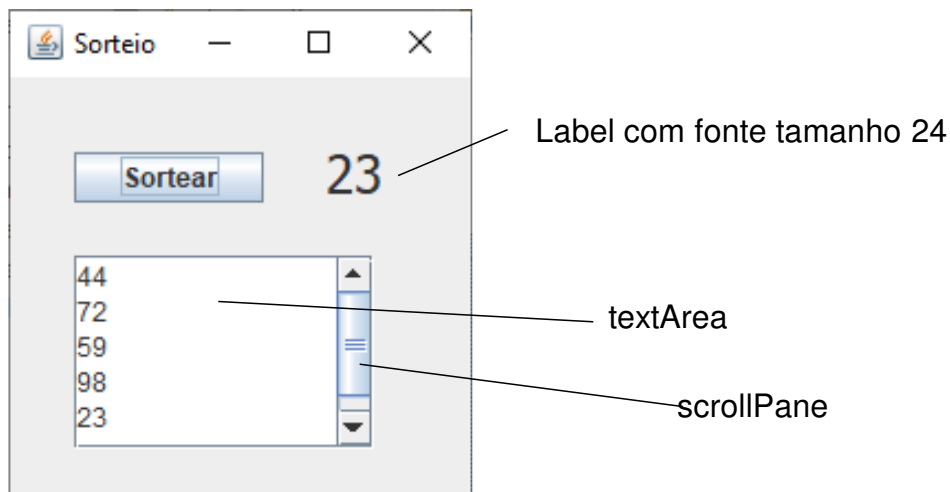
EXEMPLO 3

fausto.ayres@ifpb.edu.br

41

Sorteio.java

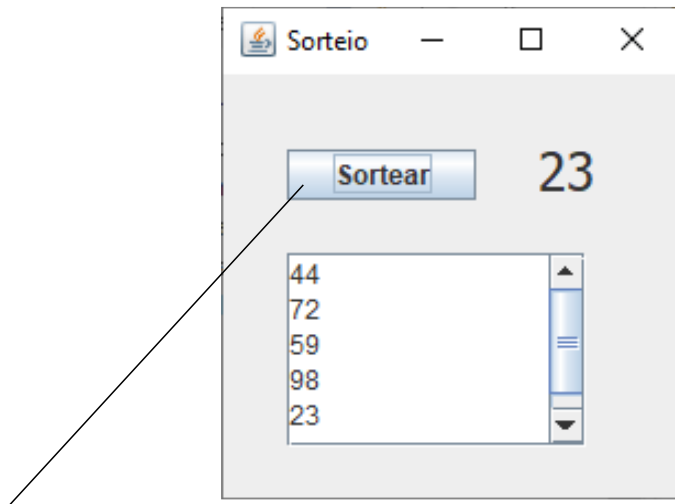
- Sortear número aleatório entre 0 e 99 e exibi-lo no *label* e o histórico num *textArea*



fausto.ayres@ifpb.edu.br

42

Ouvinte de ação



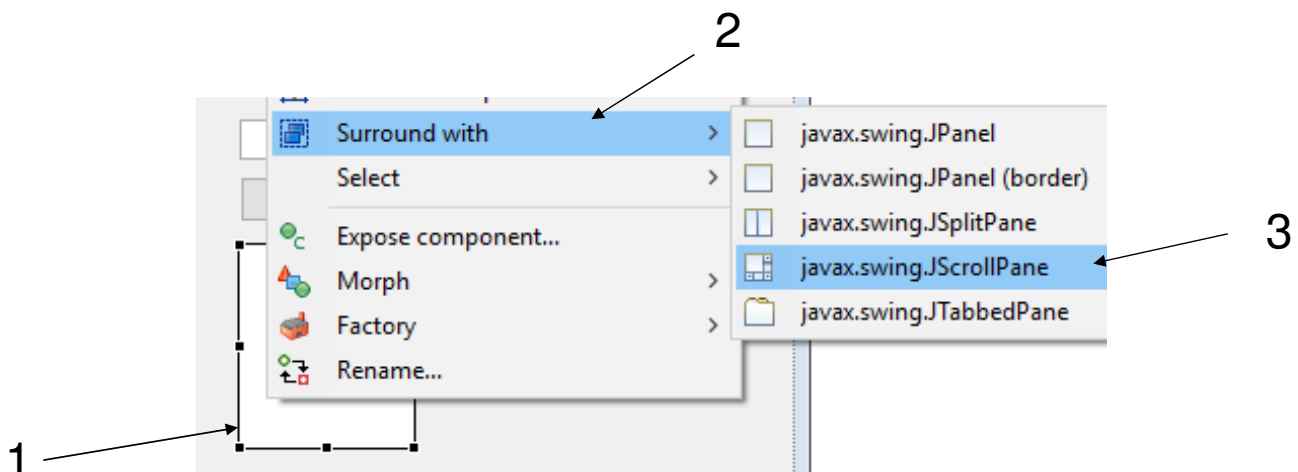
```
button.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        Random sorteio = new Random();  
        int numero = sorteio.nextInt(100);  
        String texto = numero + "  
        label.setText(texto);  
        textArea.append(texto + "\\n");  
    }  
});
```

tausto.ayres@itpb.edu.br

43

Painel de rolamento (JScrollPane)

1. Selecione o componente textArea
2. Use o botão direito do mouse, selecione **Surround with** (envolver) e depois **JScrollPane** (painel de rolamento)



fausto.ayres@ifpb.edu.br

44