

Java Swing

Profº Frederico Durão

Departamento de Ciência da Computação

Universidade Federal da Bahia



Agenda

- ☐ O que é Java Swing
- ☐ Como usar o Swing
- ☐ Meu primeiro programa
 - ☐ Construindo a primeira tela - JFrame
 - ☐ Executando o programa
- ☐ Window Builder
 - ☐ Instalando o Window Builder no Eclipse
 - ☐ Abrindo o editor do Window Builder
 - ☐ Perspectiva
 - ☐ Como usar (drop & drag)
- ☐ JPanel
- ☐ Layouts
- ☐ JLabel
- ☐ JTextField
- ☐ Alinhamento

Agenda

- ☐ JPasswordField
- ☐ Simulação da tela em execução
- ☐ JScrollPane
- ☐ JTextArea
- ☐ Eventos
- ☐ Botões
 - ☐ JButton
 - ☐ JCheckBox
 - ☐ JRadioButton
 - ☐ JMenuItem
 - ☐ JToggleButton
- ☐ Aprofundando o conhecimento
 - ☐ Javadoc
 - ☐ Referências

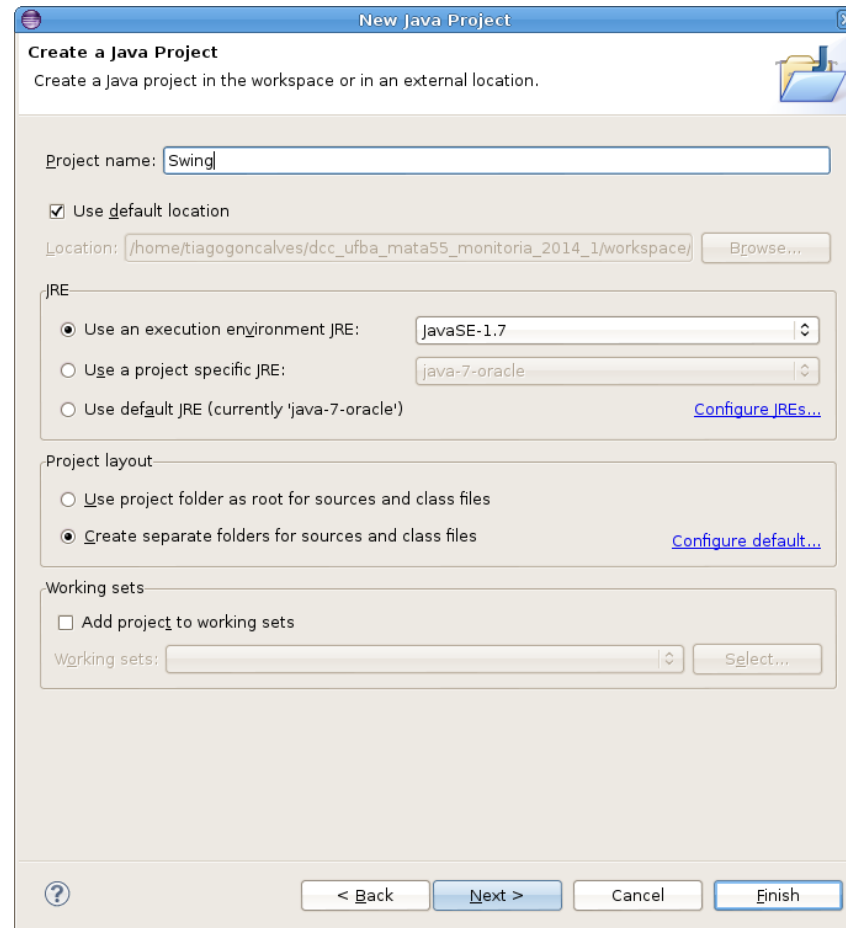
O que é Java Swing

- Swing é uma biblioteca oficial da linguagem Java.
- É utilizada para criar GUIs (Graphical User Interfaces).
- É uma GUI toolkit, ou seja, é formada por *widgets* básicos como botões, labels, scrollbars ou *widgets* avançados como trees e tables.
- Swing é uma parte do JFC, Java Foundation Classes, junto com o AWT, Acessability, Java 2D e Drag and Drop.

Como usar o Swing

- Editor de texto convencional do Eclipse.
- Plugins *drag and drop*(arrastar e soltar) .

Criando o projeto



The image shows a 'New Java Project' dialog box with the following sections:

- Create a Java Project**
Create a Java project in the workspace or in an external location.
- Project name:** Swing
- Use default location** (checked)
Location: /home/tiagogoncalves/dcc_ufba_mata55_monitoria_2014_1/workspace/ [Browse...](#)
- JRE**
 - ☒ Use an execution environment JRE: javaSE-1.7
 - ☐ Use a project specific JRE: java-7-oracle
 - ☐ Use default JRE (currently 'java-7-oracle') [Configure JREs...](#)
- Project layout**
 - ☐ Use project folder as root for sources and class files
 - ☒ Create separate folders for sources and class files [Configure default...](#)
- Working sets**
 - ☐ Add project to working sets
 - Working sets:** [Select...](#)

At the bottom, there is a help icon (?), a '< Back' button, a 'Next >' button, a 'Cancel' button, and a 'Finish' button.

Criando a classe principal

New Java Class

Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ default ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☒ `public static void main(String[] args)`

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

Meu primeiro programa

Construindo a primeira tela

```
public class PrimeiroPrograma extends JFrame{  
    public PrimeiroPrograma () {  
        }  
}
```

- A classe **PrimeiroPrograma** estende de **JFrame**.
- **JFrame** é uma janela **top-level** com um título e uma borda.

Meu primeiro programa

Executando o programa

- Necessário programar um pouco mais.

```
package br.ufba.mata55;
import ...
public class PrimeiroPrograma extends JFrame{
    public PrimeiroPrograma() {
        setSize(300,200); //largura,altura
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable(){
            @Override
            public void run(){
                PrimeiroPrograma pp;
                pp = new PrimeiroPrograma();
                pp.setVisible(true);
            }
        });
    }
}
```

Meu primeiro programa

Executando o programa

- O método **setSize(largura,altura)** define o tamanho da janela.
 - O tamanho é dado em pixels.
- A classe abstrata **SwingUtilities**
- O método **invokeLater**
- O método **setVisible** garante que a janela será visível na execução.

Meu primeiro programa

Executando o programa

- A execução do programa é igual a execução de um programa java convencional.

[imagem aqui]

Meu primeiro programa

Executando o programa

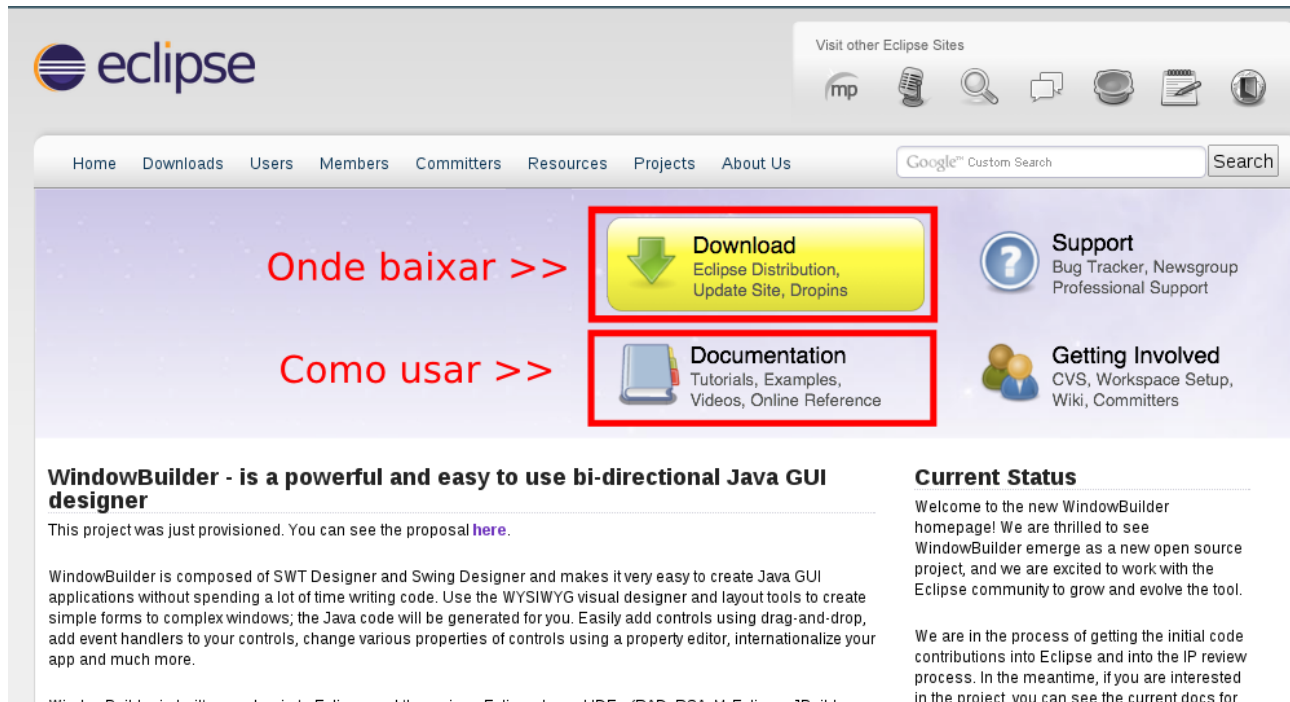
- O resultado será:

- [imagem aqui]

Window Builder

Instalando o Window Builder no Eclipse

- <https://www.eclipse.org/windowbuilder/>
- Clique em **Download**



The screenshot shows the Eclipse WindowBuilder website. At the top is the Eclipse logo and a navigation bar with links: Home, Downloads, Users, Members, Committers, Resources, Projects, and About Us. A Google Custom Search bar is also present. Below the navigation bar, there are two main sections: 'Onde baixar >>' (Where to download) and 'Como usar >>' (How to use). The 'Onde baixar >>' section has a yellow button labeled 'Download' with a green arrow icon, which is highlighted with a red box. The 'Como usar >>' section has a blue button labeled 'Documentation' with a book icon, also highlighted with a red box. To the right of these buttons are links for 'Support' (Bug Tracker, Newsgroup, Professional Support) and 'Getting Involved' (CVS, Workspace Setup, Wiki, Committers). Below these sections, there is a 'WindowBuilder - is a powerful and easy to use bi-directional Java GUI designer' section with a brief description and a link to the proposal. To the right of this is a 'Current Status' section with a welcome message and information about the project's progress.

WindowBuilder - is a powerful and easy to use bi-directional Java GUI designer

This project was just provisioned. You can see the proposal [here](#).

WindowBuilder is composed of SWT Designer and Swing Designer and makes it very easy to create Java GUI applications without spending a lot of time writing code. Use the WYSIWYG visual designer and layout tools to create simple forms to complex windows; the Java code will be generated for you. Easily add controls using drag-and-drop, add event handlers to your controls, change various properties of controls using a property editor, internationalize your app and much more.

WindowBuilder is built as a plug-in to Eclipse and the various Eclipse-based IDEs (EAD, BSA, MyEclipse, JBuilder, etc.).

Current Status

Welcome to the new WindowBuilder homepage! We are thrilled to see WindowBuilder emerge as a new open source project, and we are excited to work with the Eclipse community to grow and evolve the tool.

We are in the process of getting the initial code contributions into Eclipse and into the IP review process. In the meantime, if you are interested in the project, you can see the current docs for

Window Builder

Instalando o Window Builder no Eclipse

- Copie o link de acordo com a versão do Eclipse

Installing WindowBuilder Pro

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.

Develop Java graphical user interfaces in minutes for Swing, SWT, RCP and XWT with WindowBuilder Pro's WYSIWYG, drag-and-drop interface. Use wizards, editors and intelligent layout assist to automatically generate clean Java code, with the visual design and source always in sync.

These instructions assume that you have already installed some flavor of Eclipse. If you have not, Eclipse can be downloaded from <http://www.eclipse.org/downloads/>. Instructions and system requirements for installing WindowBuilder can be found [here](#).

Update Sites

Eclipse Version	Release Version		Integration Version	
	Update Site	Zipped Update Site	Update Site	Zipped Update Site
4.4 (Luna)			link	link (MD5 Hash)
4.3 (Kepler)	link	link (MD5 Hash)	link	link (MD5 Hash)
4.2 (Juno)	link	link (MD5 Hash)	link	link (MD5 Hash)
3.8 (Juno)	link	link (MD5 Hash)	link	link (MD5 Hash)
3.7 (Indigo)	link	link (MD5 Hash)	link	link (MD5 Hash)
3.6 (Helios)	link	link (MD5 Hash)		
3.5 (Galileo)	link	link (MD5 Hash)		
3.4 (Ganymede)	link	link (MD5 Hash)		

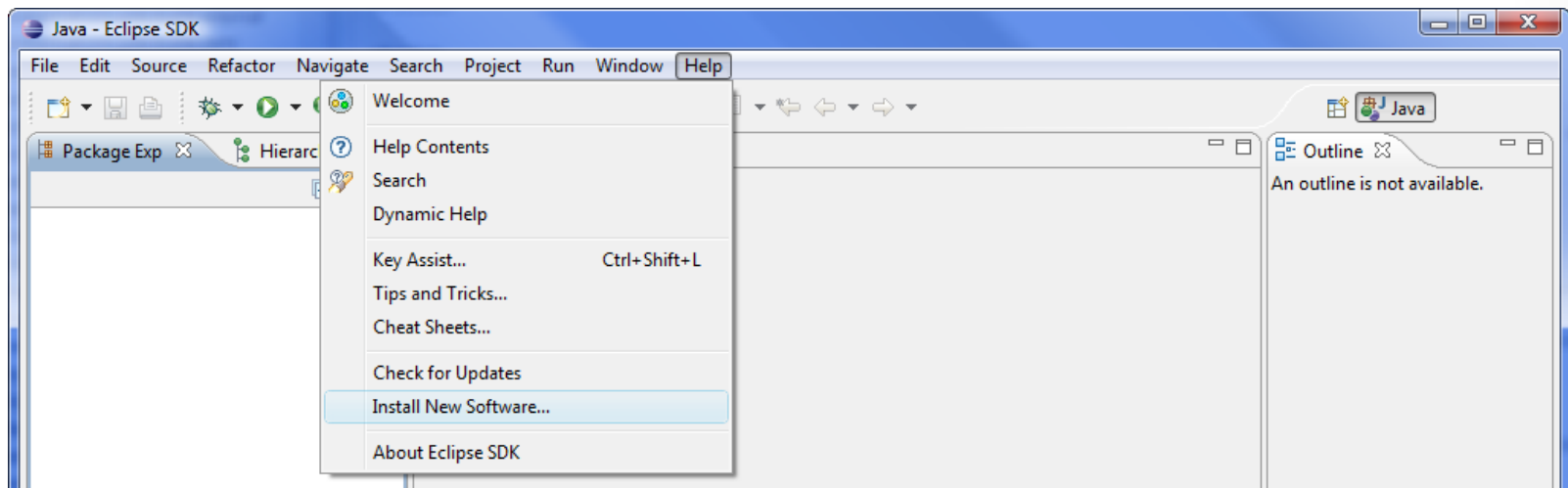
Copie o link
correspondente a
versão do Eclipse que
você está utilizando

Installing the Update Site or Zip editions for Eclipse 3.4 or above requires the full Eclipse SDK including the JDT and PDE. Use the Eclipse Classic, Java EE or RCP/Plug-in Developers distribution or install the JDT and PDE separately.

Window Builder

Instalando o Window Builder no Eclipse

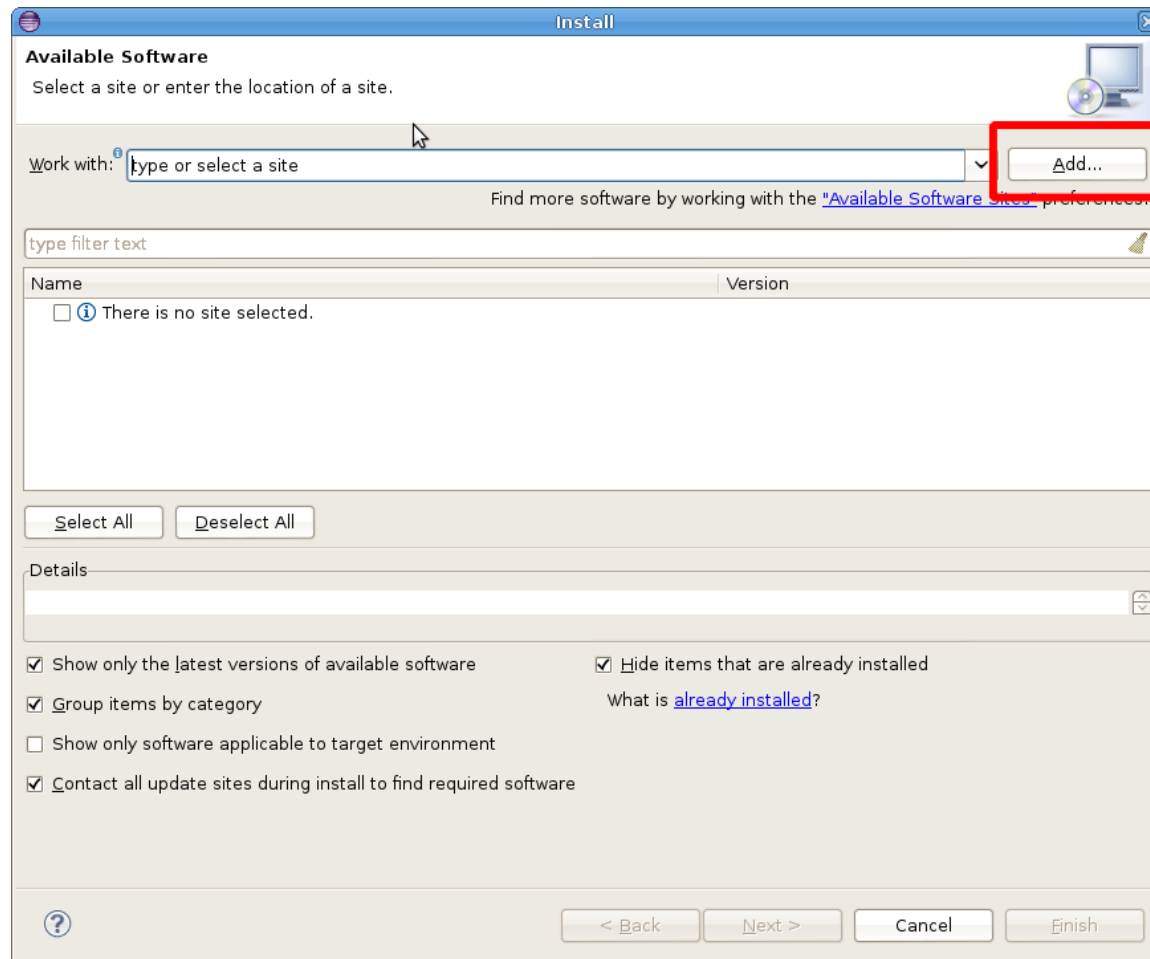
- Help > Install New Software



Window Builder

Instalando o Window Builder no Eclipse

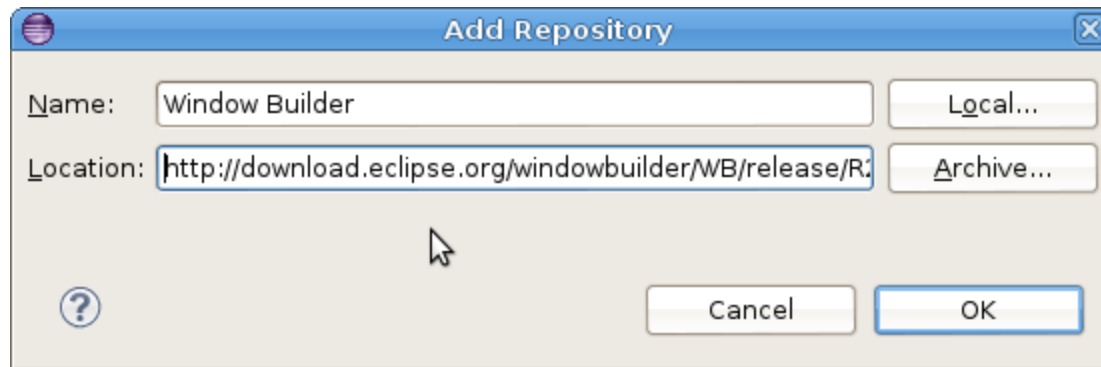
- Clique em **Add**

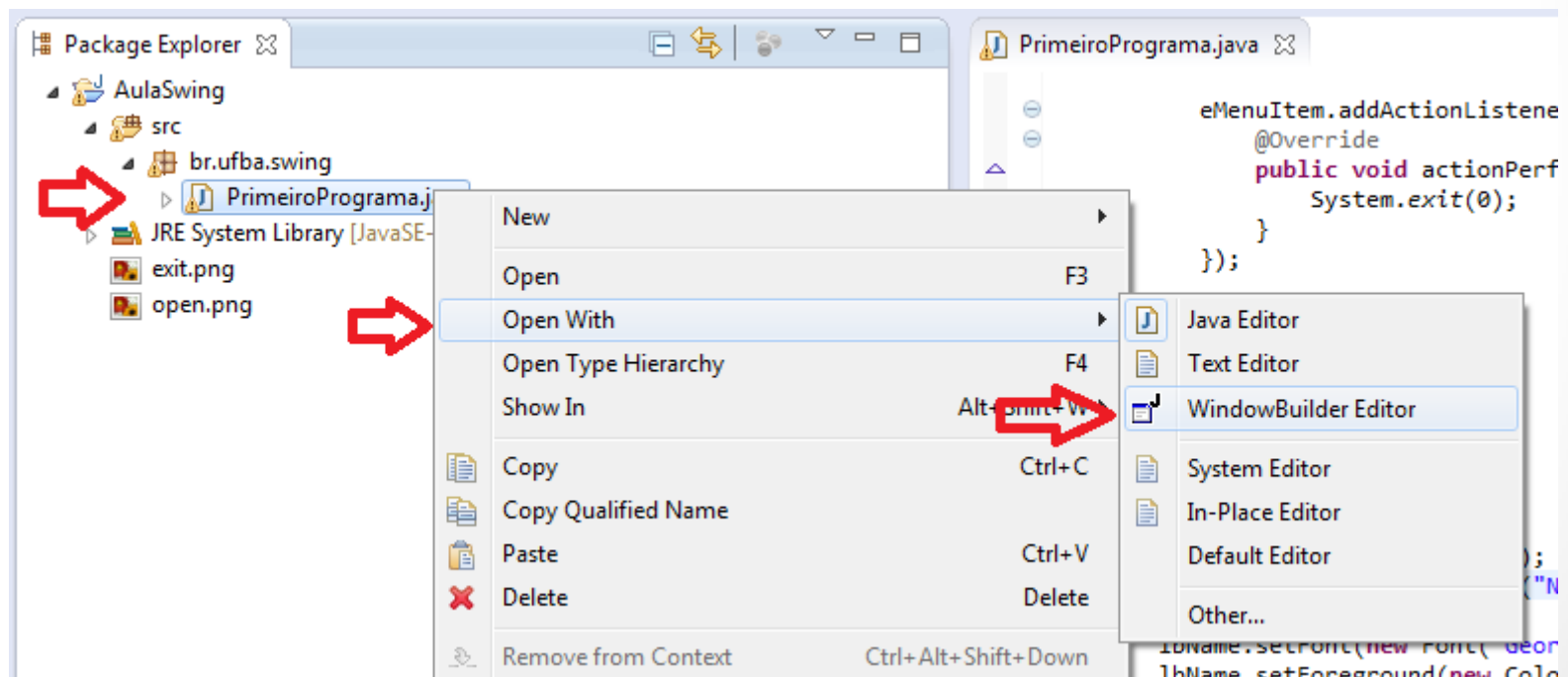


Window Builder

Instalando o Window Builder no Eclipse

- Em **Name** digite WindowBuilder(pode ser qualquer nome)
- Em **Location** cole a URL copiada da página de downloads
 - Clique em OK, aceite a licença e aguarde finalizar a instalação.

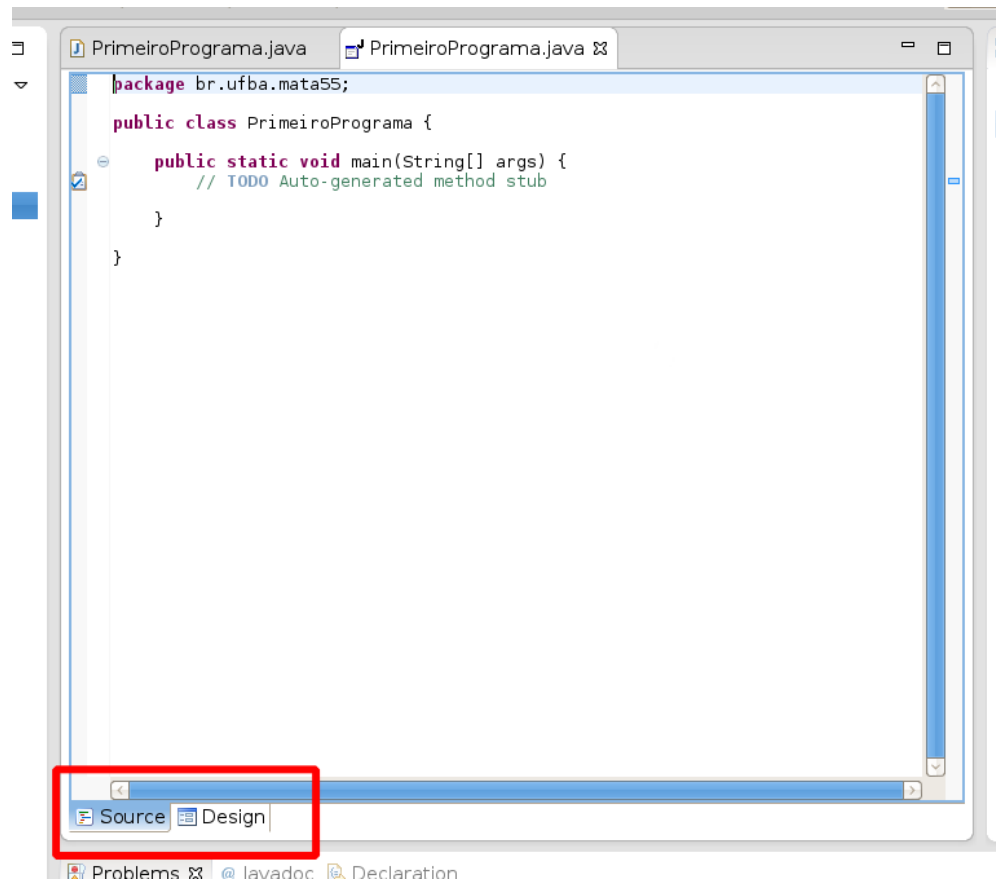




Window Builder

Perspectiva

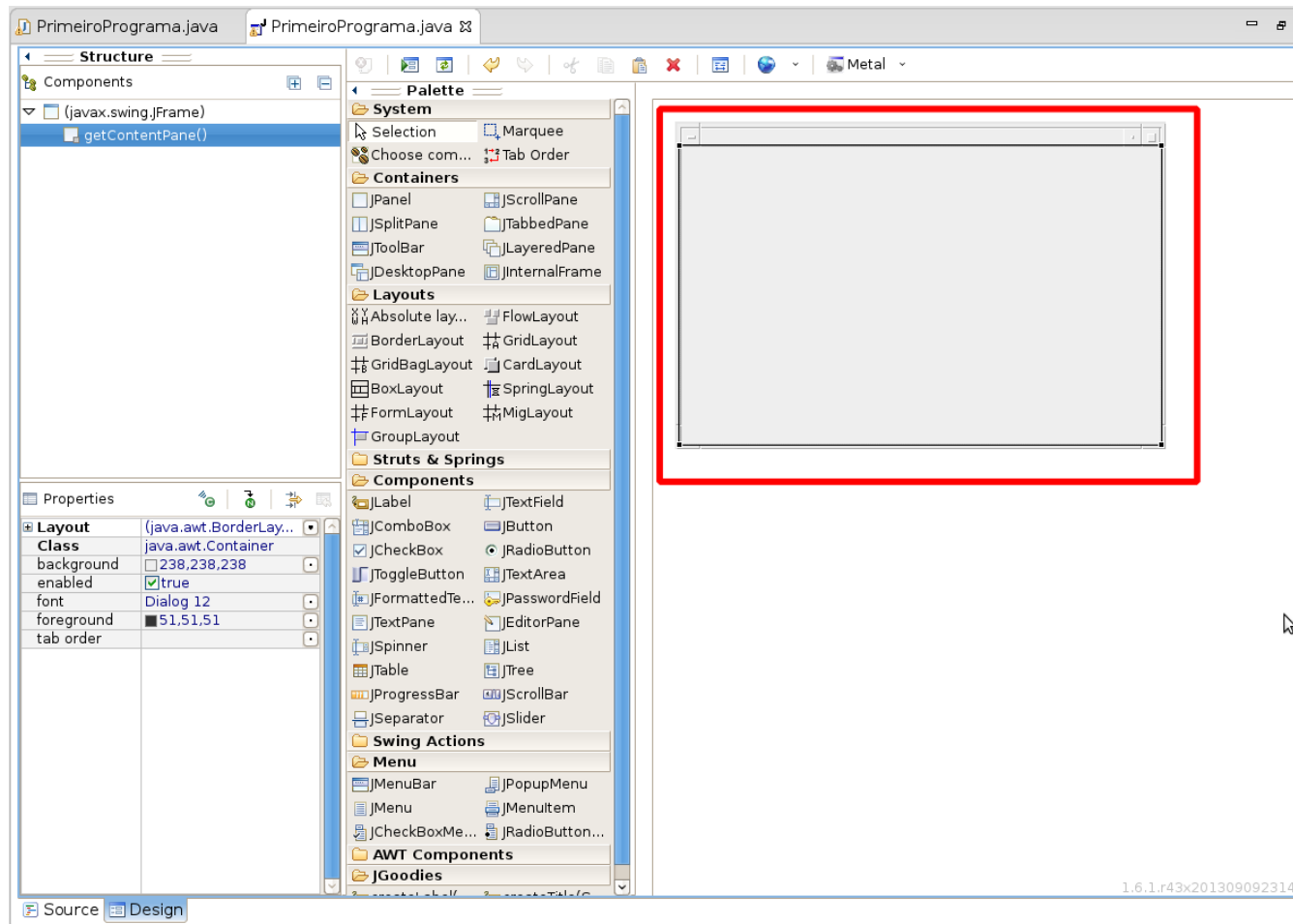
- **Source** – igual ao editor padrão do Eclipse
- **Design** – editor gráfico do Window Builder



Window Builder

Perspectiva

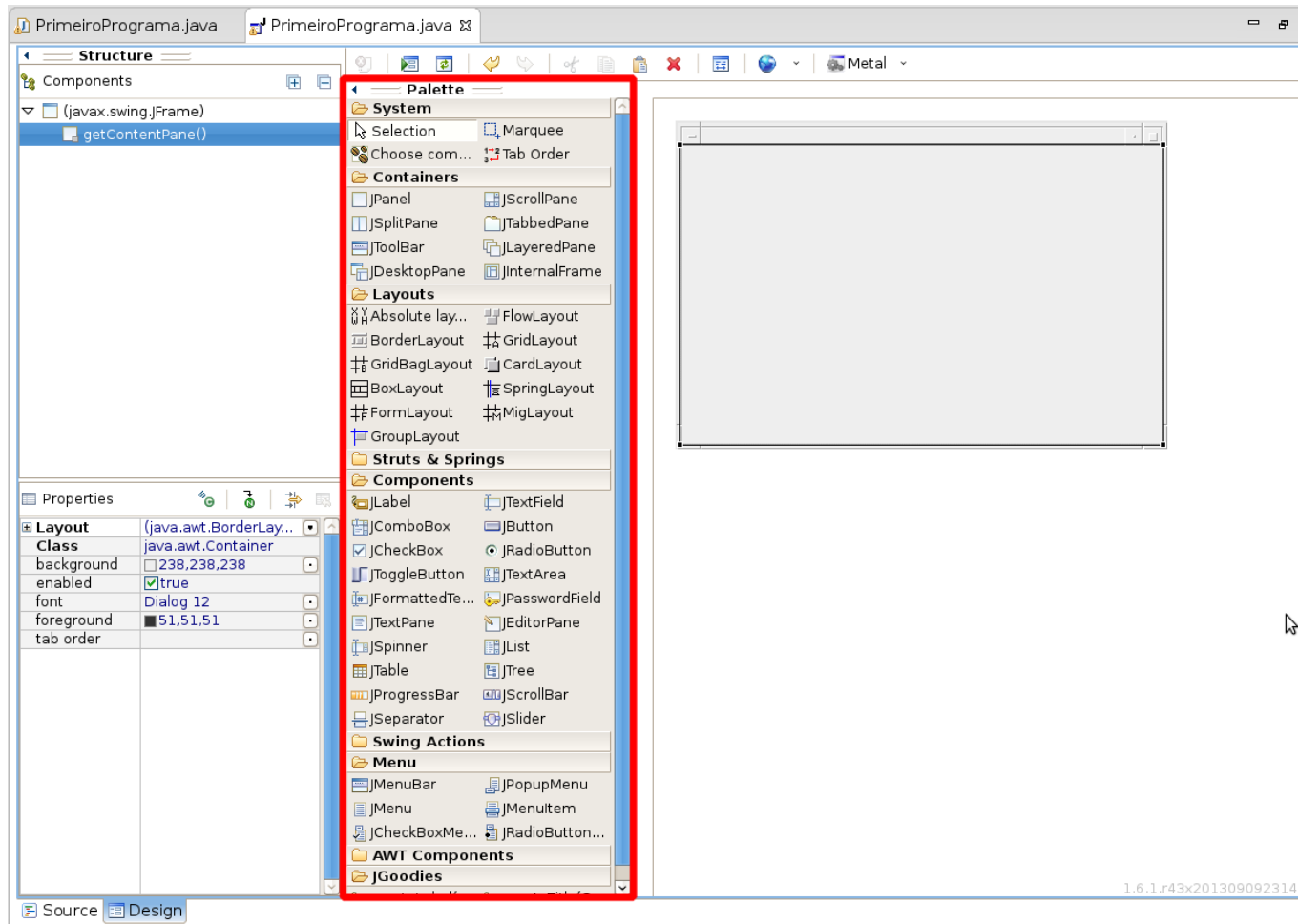
- Visualizador gráfico



Window Builder

Perspectiva

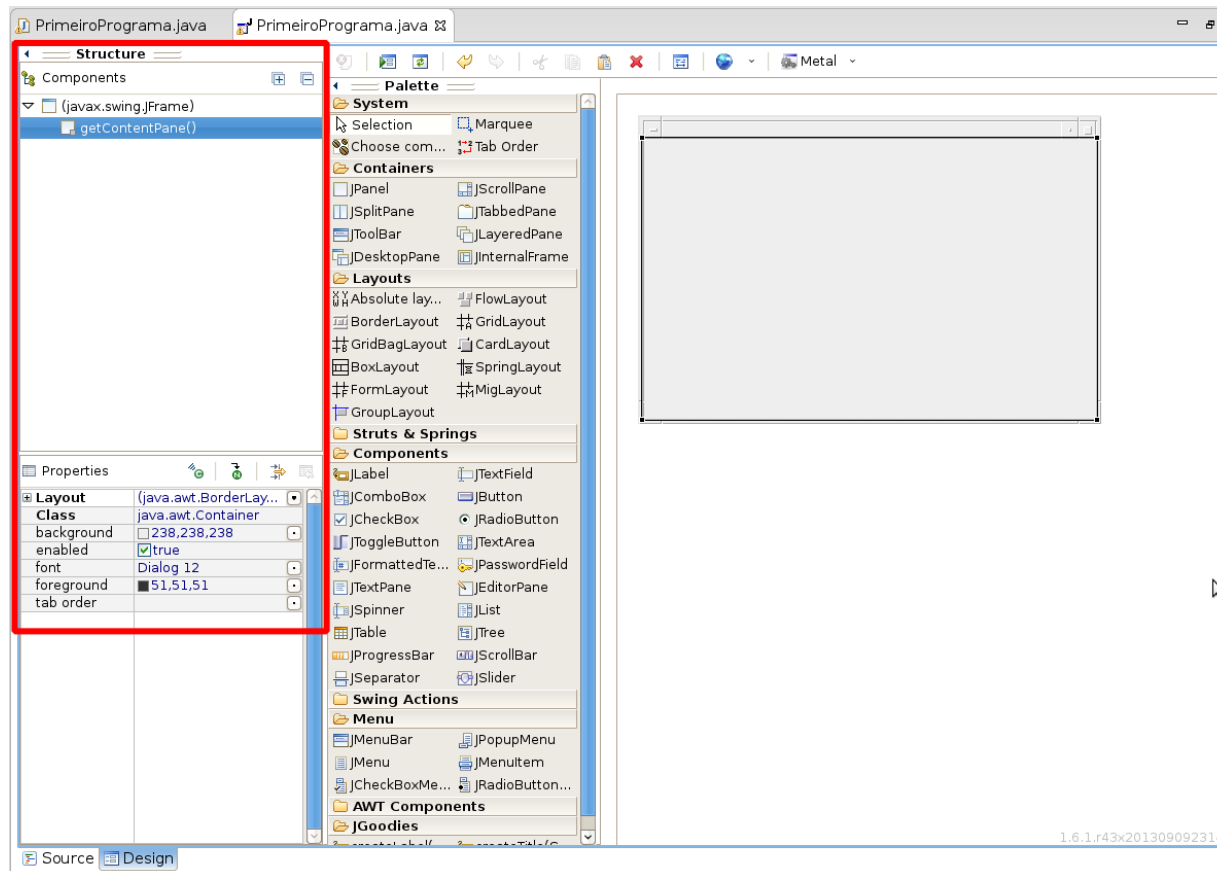
- Paleta com os principais componentes do Swing



Window Builder

Perspectiva

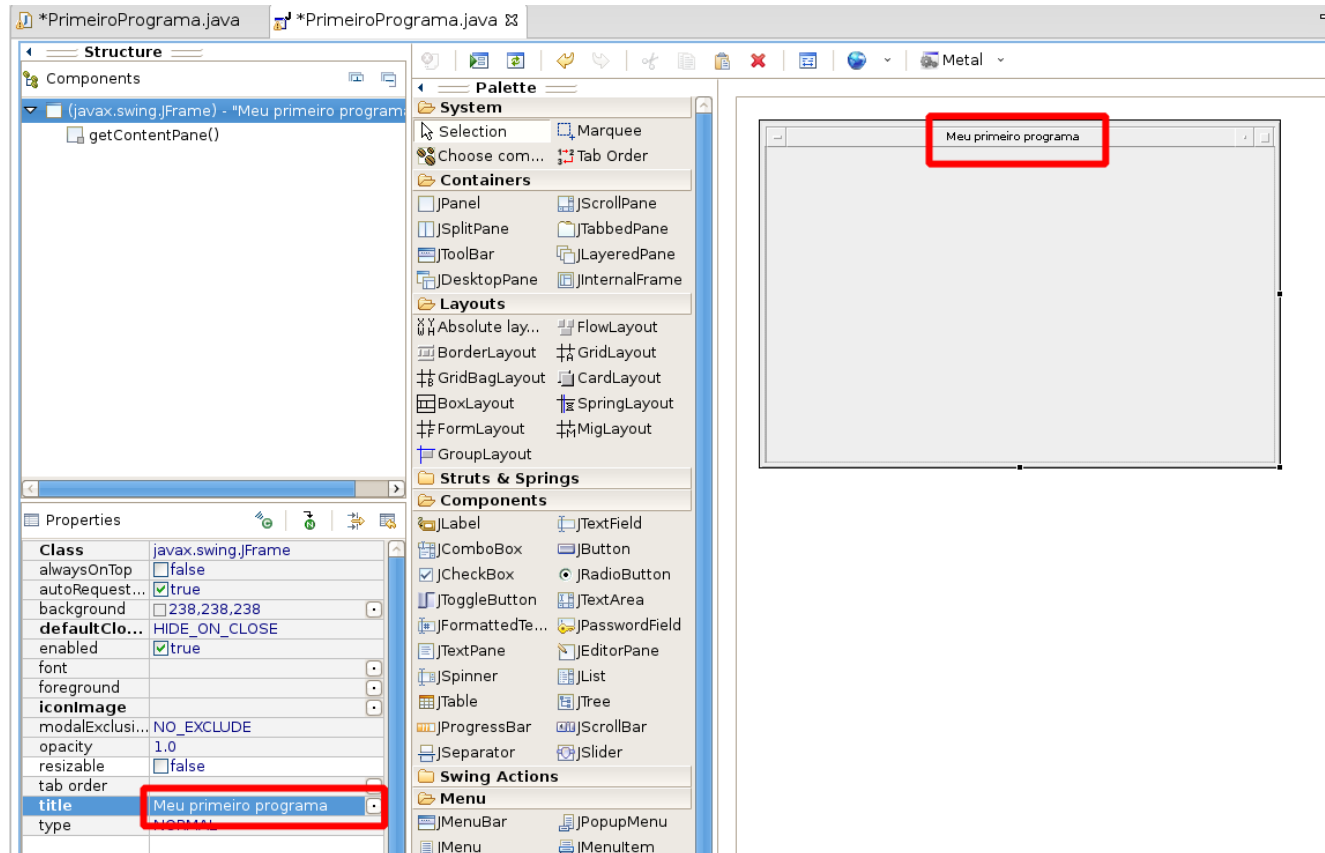
- **Structure** - estrutura da tela com seus componentes
- **Properties** – propriedades do componente selecionado.



Window Builder

Como usar

- Alterando as propriedades de um componente já inserido.
 - Selecione a janela e insira o título “Meu primeiro programa” em **Title** na aba **Properties**.



Window Builder

Como usar

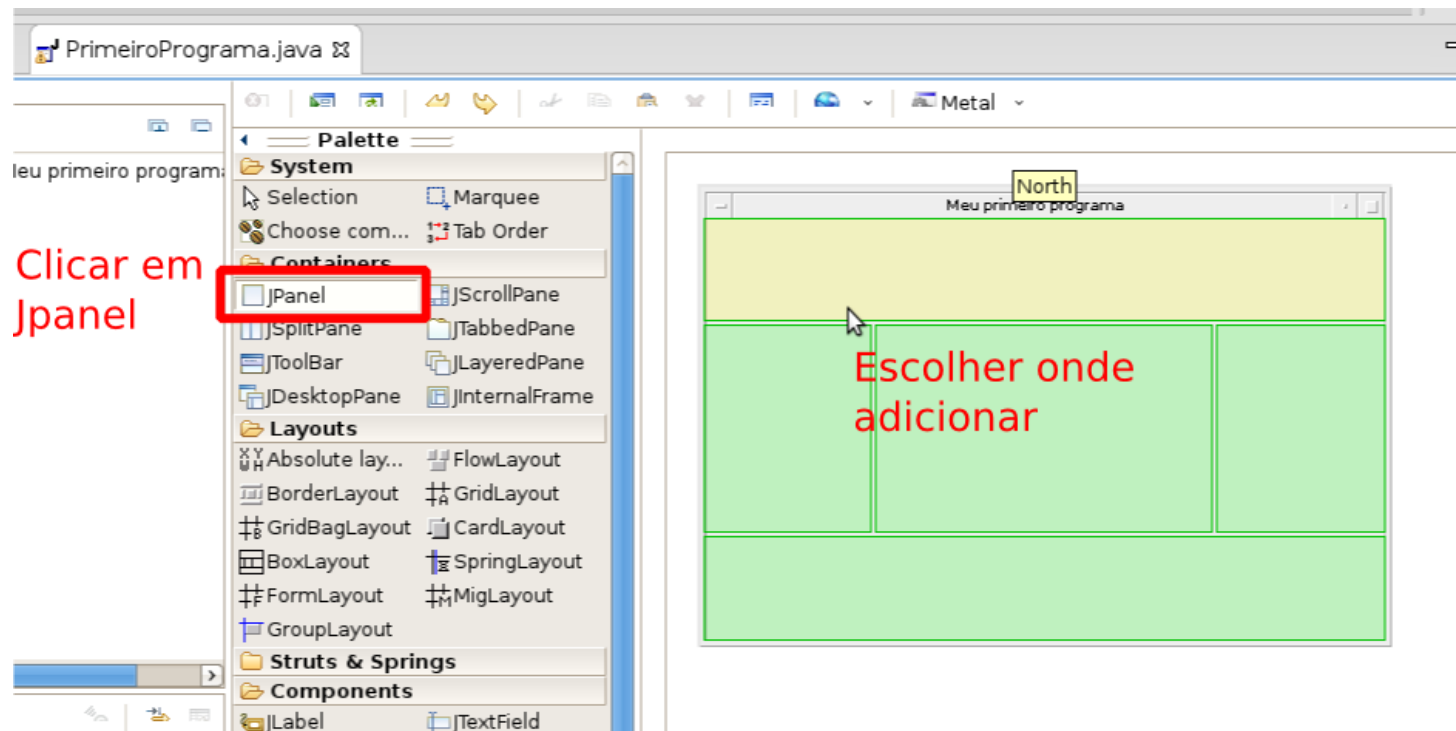
- Mude para a aba **Source** e veja o equivalente em código.

```
public class PrimeiroPrograma extends JFrame{  
    public PrimeiroPrograma() {  
        setTitle("Meu primeiro programa");  
        ...  
    }  
    ...  
}
```


Window Builder

Como usar

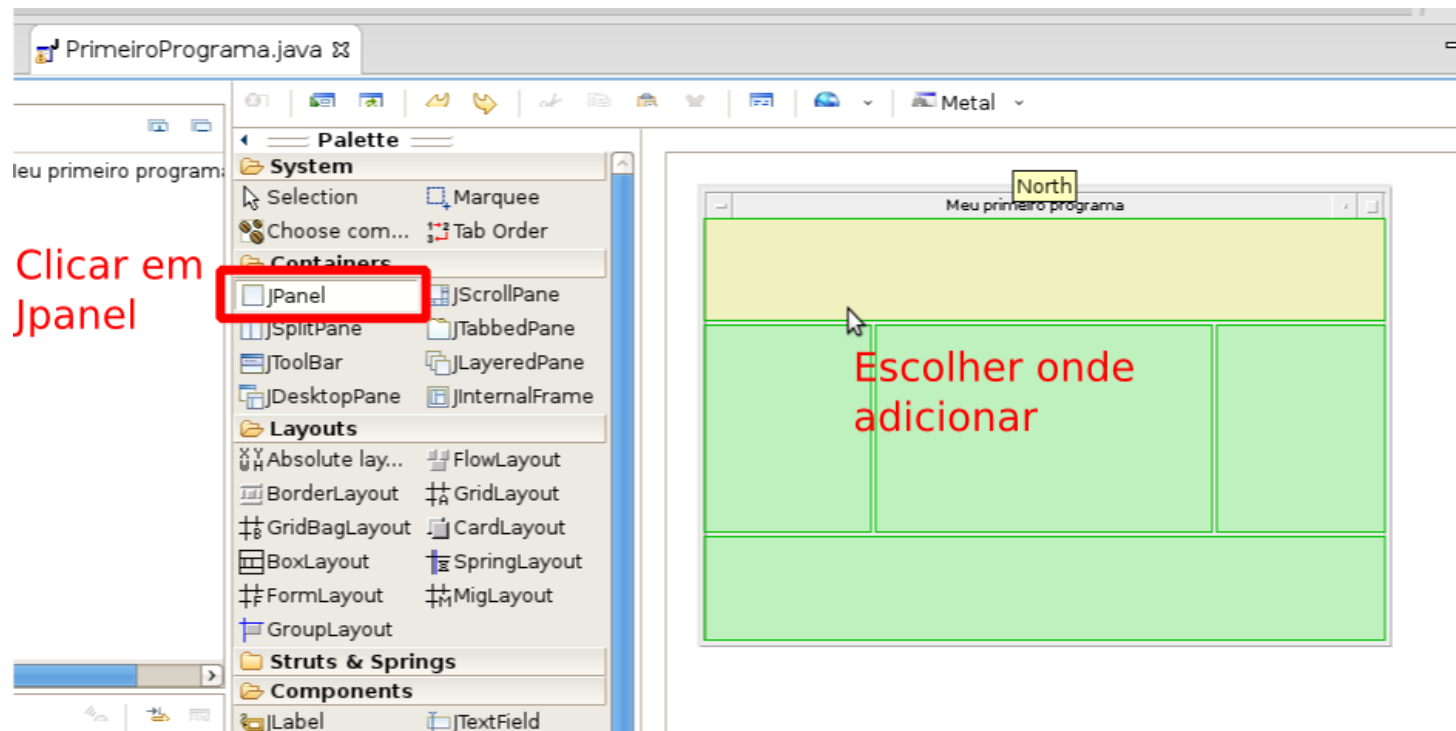
- Método **drag and drop** (clique e soltar).
- As áreas verdes são os locais onde o componente pode ser inserido.



Window Builder

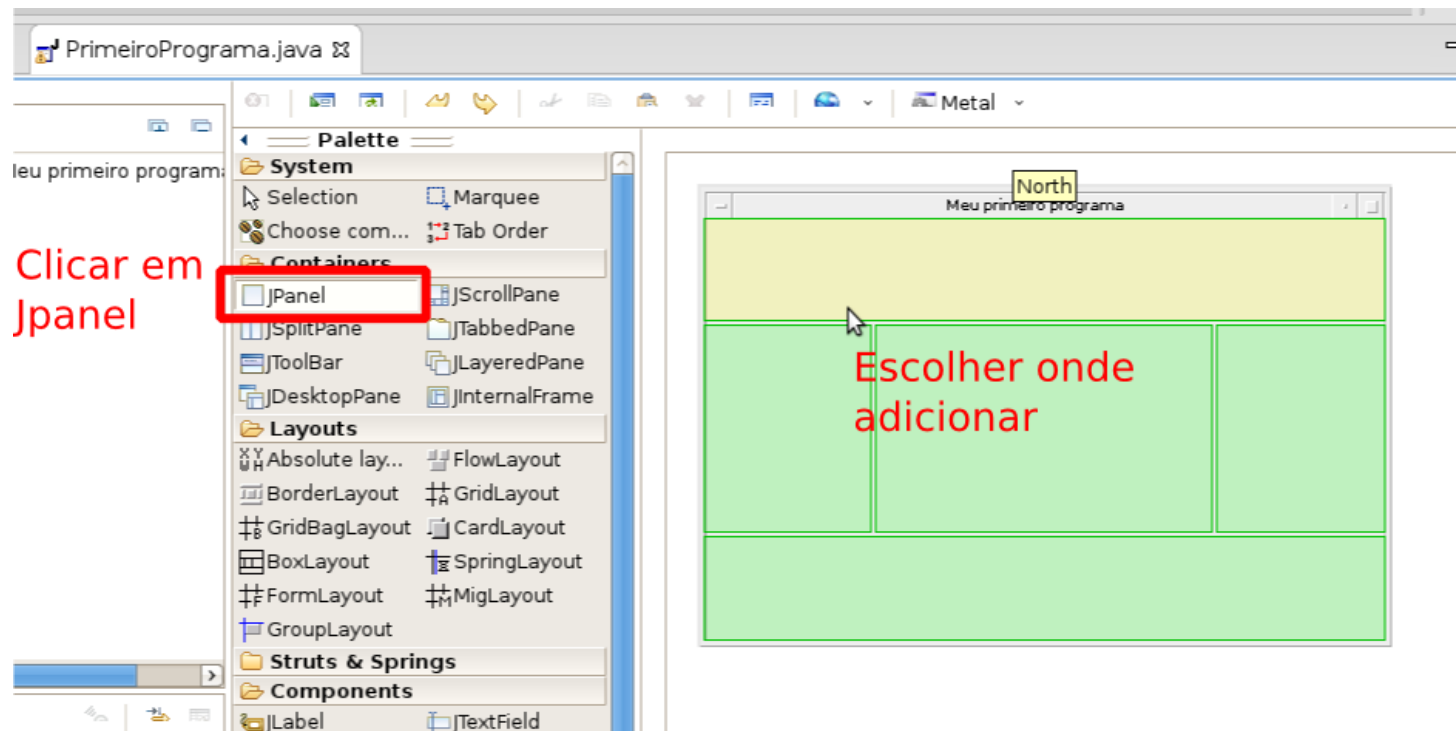
Como usar

- As divisões entre as áreas verdes são ditadas pelo **layout** escolhido (por enquanto o padrão).
- Por padrão o **layout** é o **BorderLayout** que divide a tela em **NORTH, SOUTH, WEST, CENTER** e **EAST**.



JPanel

- É um **simples container** (recipiente) **genérico**.
- É um **receptáculo** onde os componentes podem ser agrupados.
- Pode ter seu próprio **layout**, portanto suas próprias regras.



JPanel

- Mude para a aba **Source** e veja o equivalente em código.

```
public class PrimeiroPrograma extends JFrame{  
    public PrimeiroPrograma() {  
        ...  
        JPanel panel = new JPanel();  
        getContentPane().add(panel, BorderLayout.NORTH);  
        ...  
    }  
    ...  
}
```

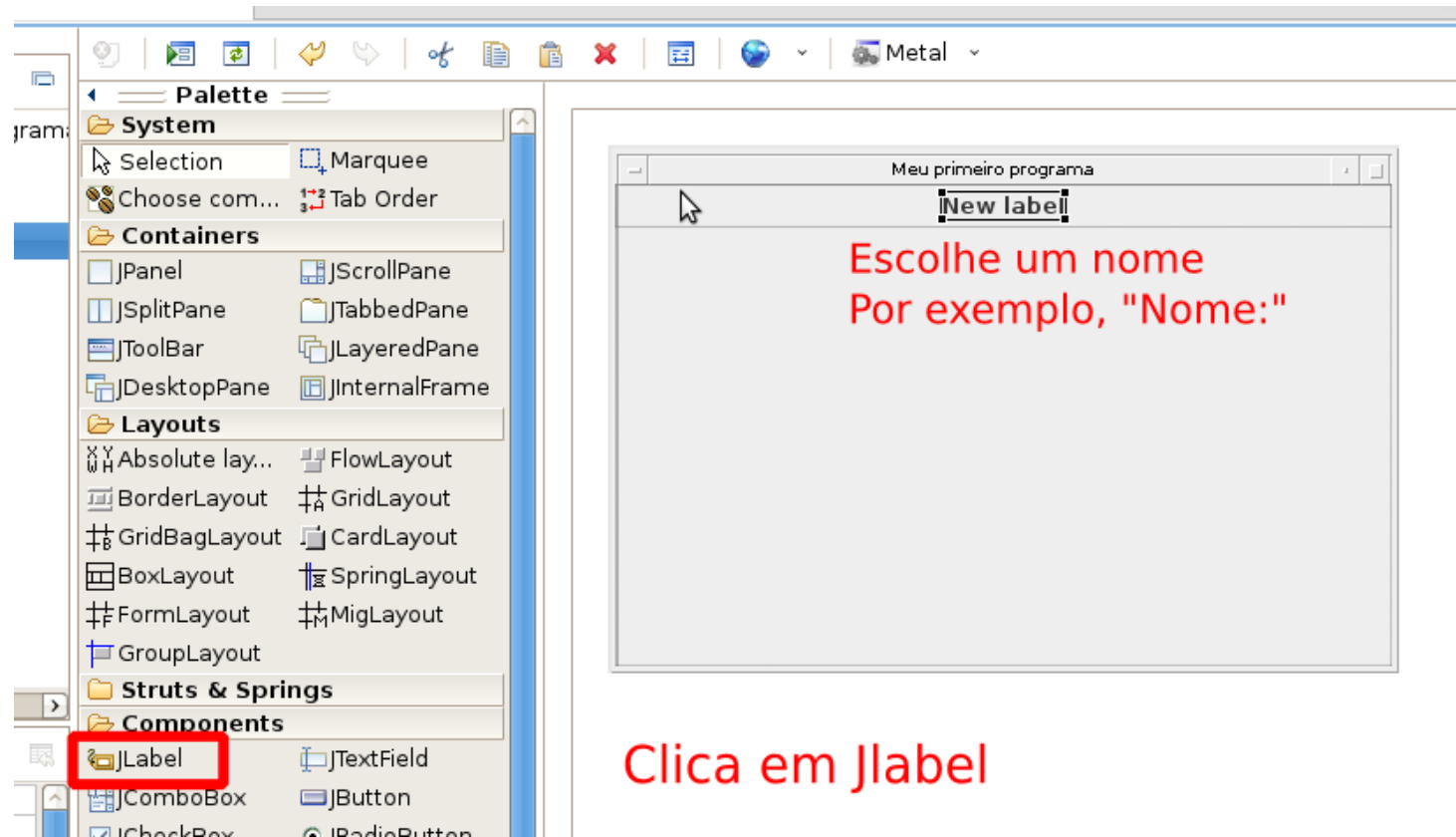
JPanel

- O método **getContentPane**
- O parâmetro **BorderLayout.NORTH** define onde o JPanel ficará.

Layouts

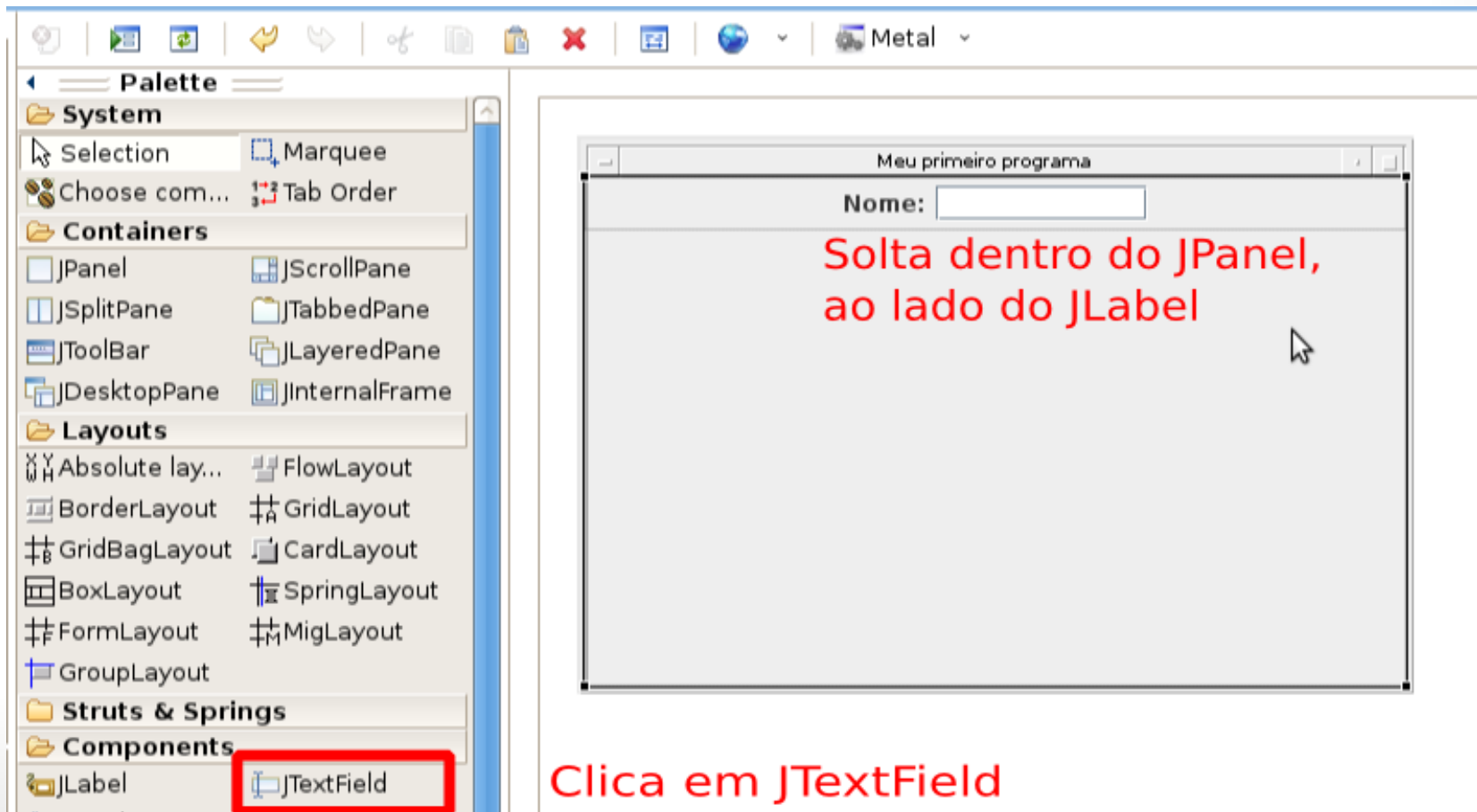
JLabel

- É um rótulo.
- Pode exibir um texto, uma imagem ou ambos.



JTextField

- É um componente que nos permite a edição de uma simples linha de texto.

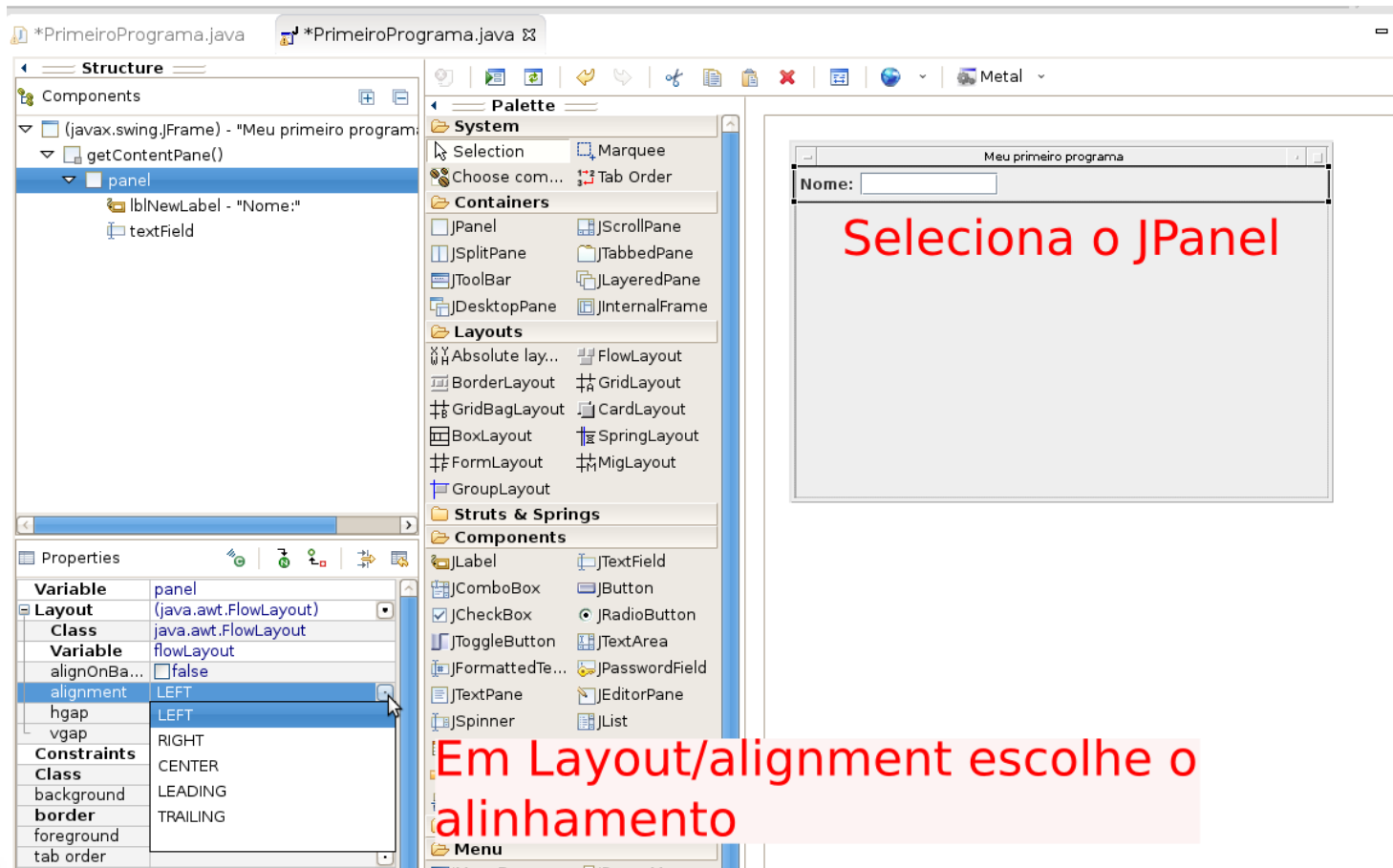


JTextField

- Mude para a aba **Source** e veja o equivalente em código.

Alinhamento

- Os tipos de alinhamento disponíveis são: **LEFT**, **RIGHT**, **CENTER**, **LEADING** (rodapé) e **TRAILING** (topo).

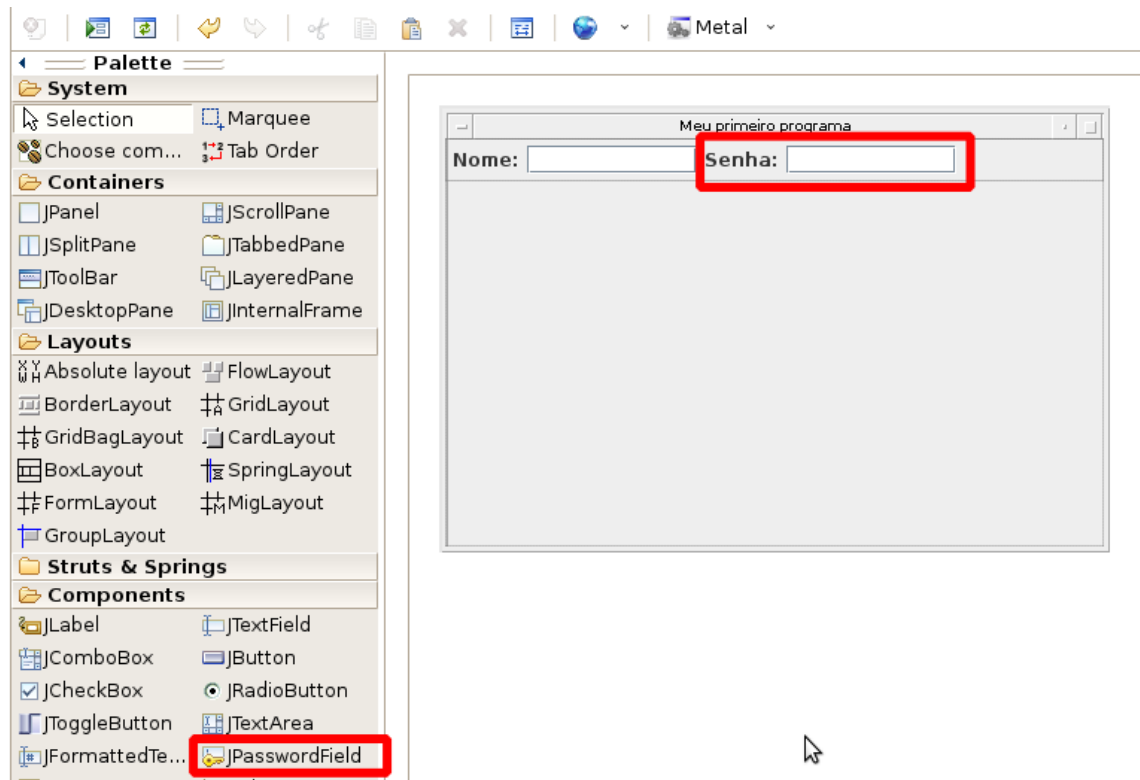


Alinhamento

- Mude para a aba **Source** e veja o equivalente em código.

JPasswordField

- Um **JTextField** não é o componente mais adequado para um campo de senha, por exemplo.
- Para isso existe o componente **JPasswordField**.



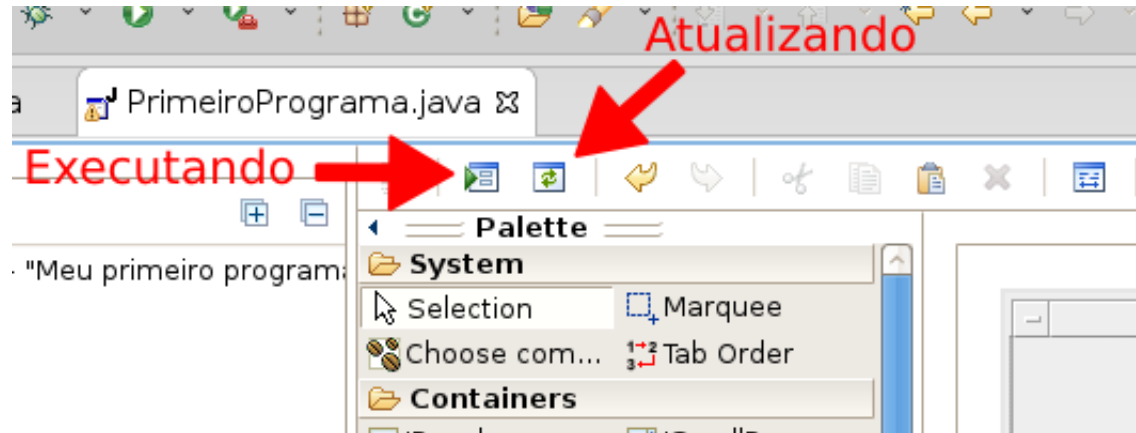
JPasswordField

- Mude para a aba **Source** e veja o equivalente em código.

```
public class PrimeiroPrograma extends JFrame{  
    public PrimeiroPrograma() {  
        ...  
        JPasswordField passwordField;  
        passwordField = new JPasswordField();  
        passwordField.setColumns(10);  
        panel.add(passwordField);  
        ...  
    }  
    ...  
}
```

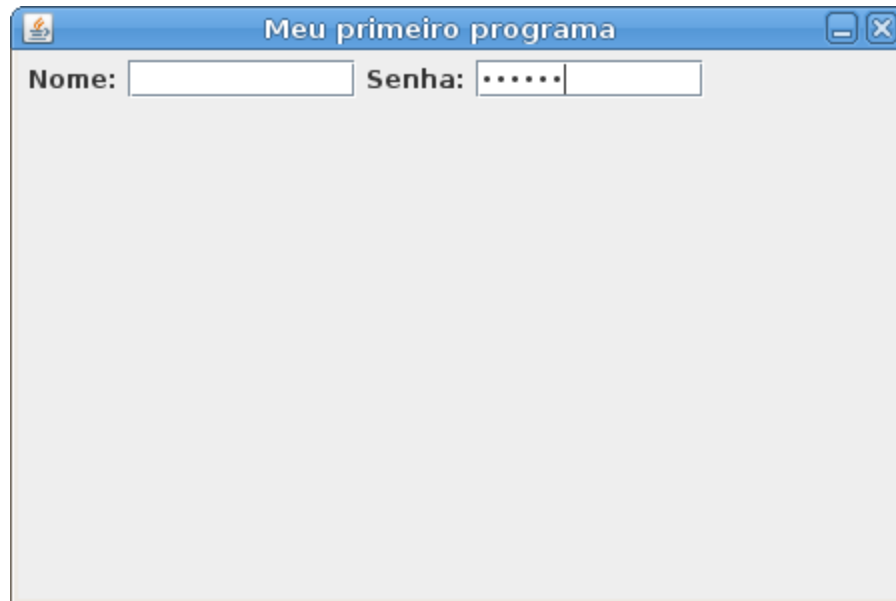
Simulação da tela em execução

- É possível simular o funcionamento da tela antes mesmo de executar o código.



Simulação da tela em execução

- Simulação da tela



Meu primeiro programa

Nome: Senha:

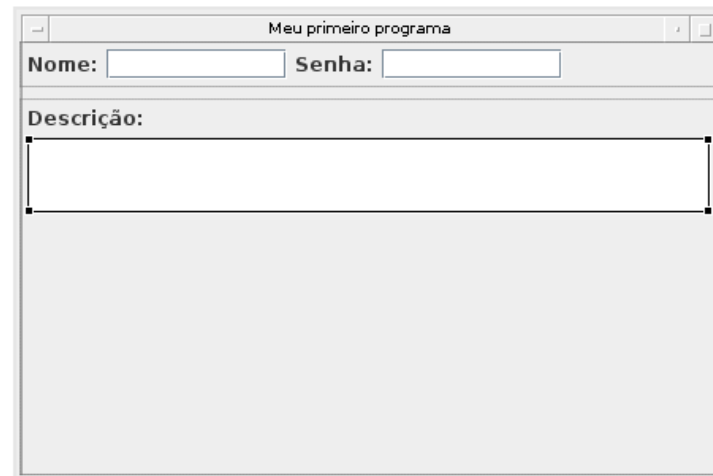
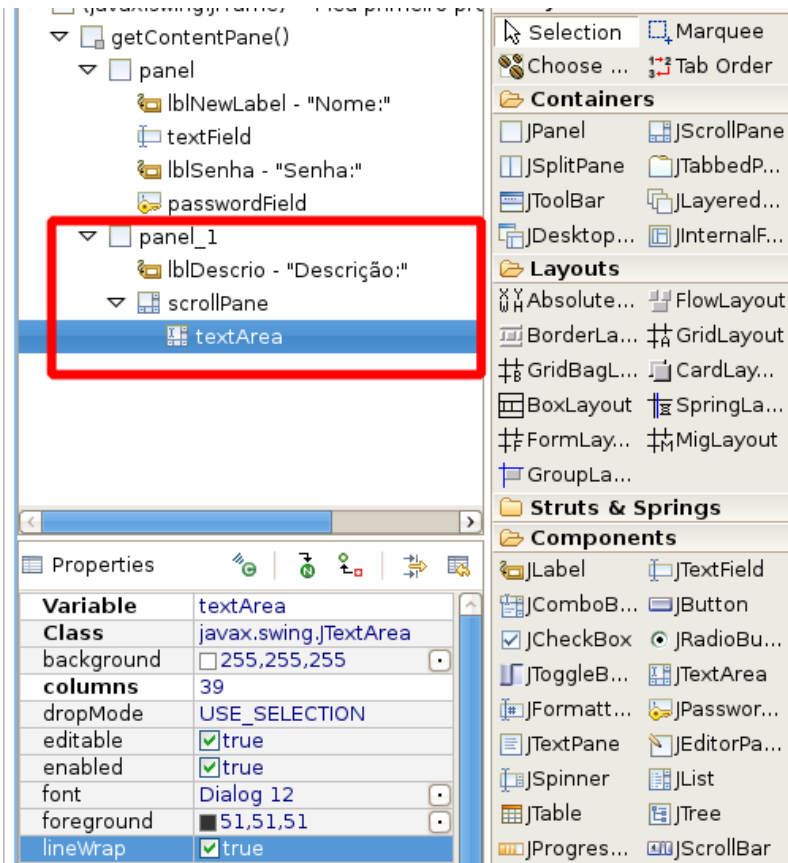
JScrollPane

- Assim como o **JPanel** é um container que agrupa componentes.
- A diferença é que o mesmo implementa uma barra de rolagem.
- Utilizaremos um **JScrollPane** combinado com o próximo componente.

JTextArea

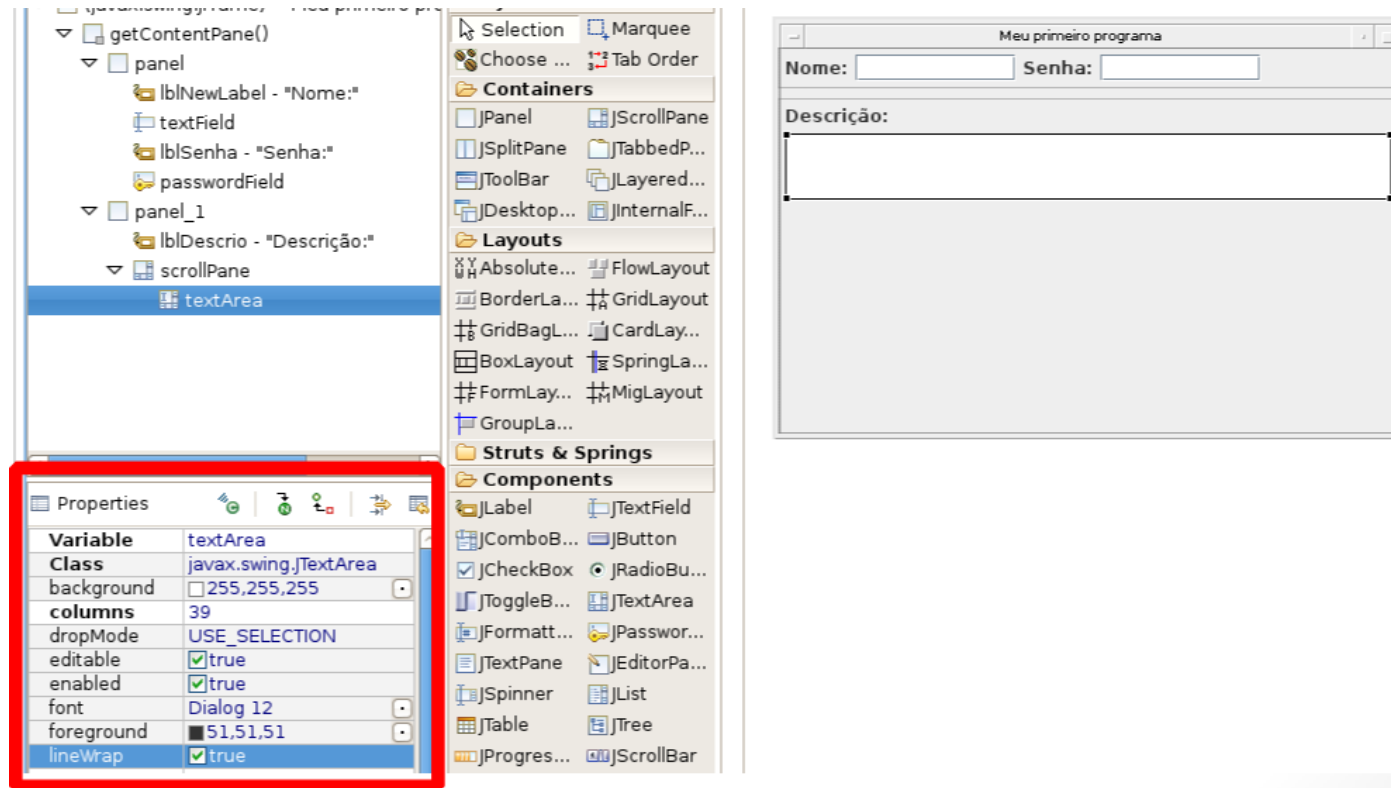
- É uma área multilinha que mostra um texto.
- Adicione um **JPanel**.
- Dentro desse **JPanel** adicione um **JLabel** Descrição.
- Adicione um **JScrollPane** a esse **JPanel**.
- Finalmente adicione o **JTextArea**.

JTextArea



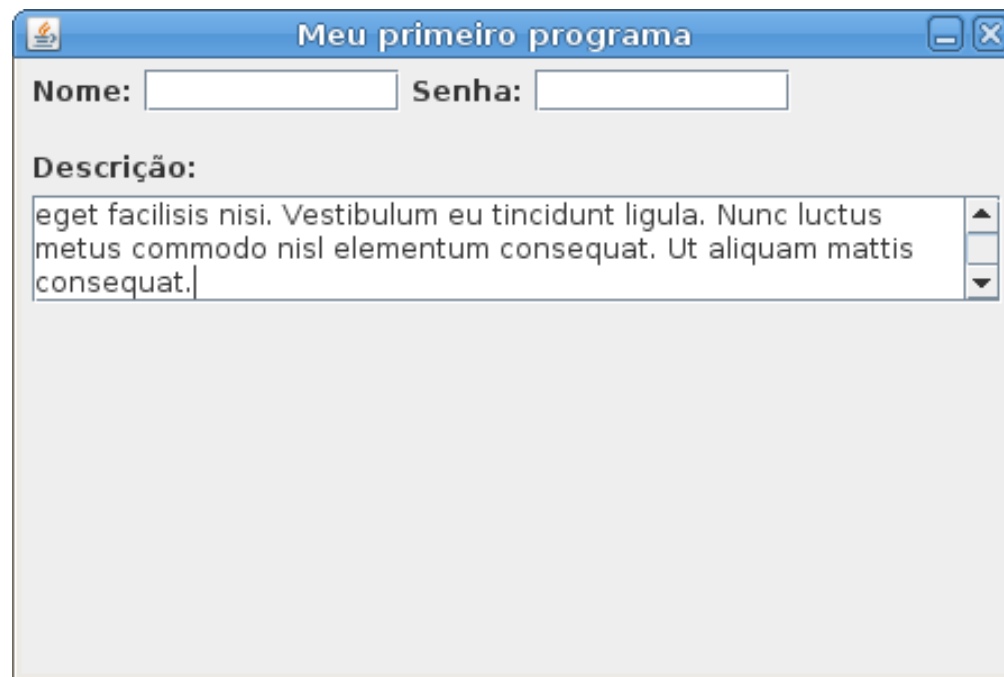
JTextArea

- Em **Properties** > **columns** foi definida a 'largura' do **JTextArea**.
- Em **Properties** > **lineWrap** foi definida a quebra de linha do texto.



JTextArea

- Simulação



The image shows a Java Swing window titled "Meu primeiro programa". Inside the window, there is a login form. At the top, there are two text input fields: "Nome:" followed by an empty text box, and "Senha:" followed by an empty text box. Below these, there is a label "Descrição:" followed by a JTextArea. The JTextArea contains the text "eget facilisis nisi. Vestibulum eu tincidunt ligula. Nunc luctus metus commodo nisl elementum consequat. Ut aliquam mattis consequat." and has a vertical scrollbar on its right side.

JTextArea

- Mude para a aba **Source** e veja o equivalente em código.

```
public class PrimeiroPrograma extends JFrame{
    public PrimeiroPrograma() {
        ...
        JScrollPane scrollPane = new JScrollPane();
        panel_1.add(scrollPane);
        JTextArea textArea = new JTextArea();
        textArea.setLineWrap(true);
        textArea.setWrapStyleWord(true);
        textArea.setRows(3);
        textArea.setColumns(39);
        scrollPane.setViewportViewView(textArea);
        ...
    }
    ...
}
```

Eventos

Botões

Botões

JButton

Botões

JCheckBox

Botões

JRadioButton

Botões

JMenuItem

Botões

JToggleButton

Aprofundando o conhecimento

Javadoc

Aprofundando o conhecimento

Referências