

# Java SWING

## MATA55 - Programação Orientada a Objetos

Profº Frederico Durão  
freddurao@dcc.ufba.br

Instituto de Matemática  
Departamento de Ciência da Computação  
Universidade Federal da Bahia

23 de abril de 2014



## 1 Introdução

- O que é Java Swing
- Por que estudar Java Swing
  - Swing vs. others
- Como usar o Java Swing
  - Codificando no Editor convencional
  - Metodo Drop & Drag - Plugin WindowBuilder para Eclipse
  - WindowBuilder - Instalando
  - WindowBuilder - Perspectiva

## 2 Referências

# Introdução

## O que é Java Swing

- Swing é uma API Java para interfaces gráficas.
- Faz parte da JFC (Java Foundation Classes) que encapsula um grupo de 'features' para GUIs (Graphical User Interfaces).

# Introdução

Por que estudar Java Swing

# Introdução

Por que estudar Java Swing

Swing vs. others

# Introdução

## Como usar o Java Swing

- Editor convencional.
- Plugins Drop & Drag.

# Introdução I

## Como usar o Java Swing

### Codificando no Editor convencional

---

```
package br.ufba.mata55;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

public class HelloWorldSwing extends JFrame {

    public HelloWorldSwing() {
        initUI();
    }

    private void initUI() {

        JPanel panel = new JPanel();
        getContentPane().add(panel);
```

# Introdução II

## Como usar o Java Swing

```
panel.setLayout(null);

setTitle("Hello World!");
setSize(300, 200);
setLocationRelativeTo(null);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}

public static void main(String[] args) {

    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            HelloWorldSwing ex = new HelloWorldSwing();
            ex.setVisible(true);
        }
    });
}
```



# Introdução III

Como usar o Java Swing

}

---

# Introdução

Como usar o Java Swing

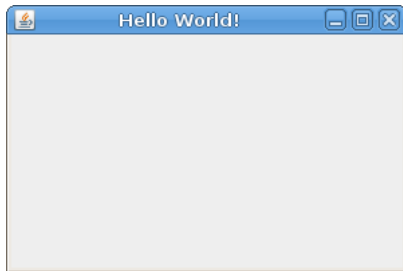


Figura: HelloWorldSwing.java

# Introdução

## Criando Projeto

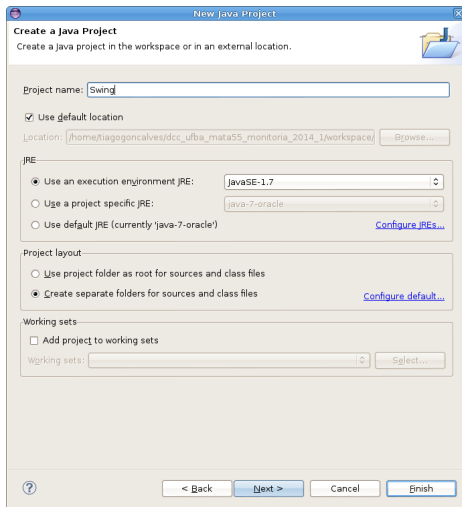


Figura: Projeto Java

# Introdução

## Criando Classe Principal

**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

Figura: PrimeiroPrograma.java

# Página principal do Window Builder para Eclipse

<https://www.eclipse.org/windowbuilder/>

**eclipse**

Visit other Eclipse Sites

Home Downloads Users Members Committers Resources Projects About Us

Google™ Custom Search Search

Onde baixar >>

Como usar >>

**Download**  
Eclipse Distribution,  
Update Site, Dropins

**Documentation**  
Tutorials, Examples,  
Videos, Online Reference

**Support**  
Bug Tracker, Newsgroup  
Professional Support

**Getting Involved**  
CVS, Workspace Setup,  
Wiki, Committers

**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 (RCP, RSE, MyEclipse, JBuilder, ...)

**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

Figura: Página principal do Window Builder

# WindowBuilder - Instalando

## Help > Instal New Software

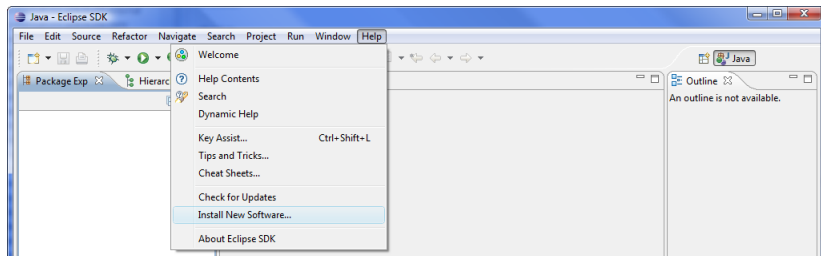


Figura: Instalando Window Builder

# WindowBuilder - Instalando

Clique em **Add**.

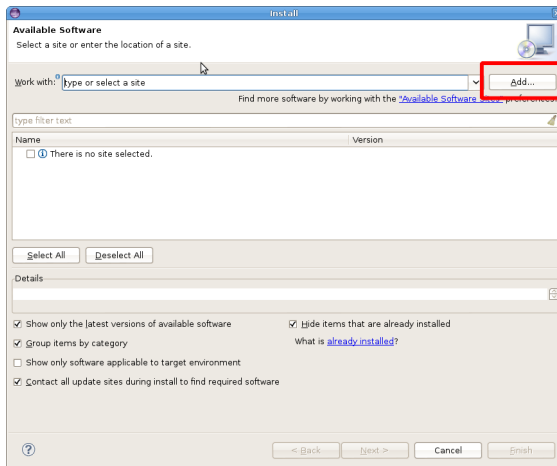


Figura: Instalando Window Builder

# WindowBuilder - Instalando

Em **Name** digite **WindowBuilder** (pode ser qualquer nome).

Em **Location** digite a url

<http://download.eclipse.org/windowbuilder/WB/release/R201309271200/4.3/>

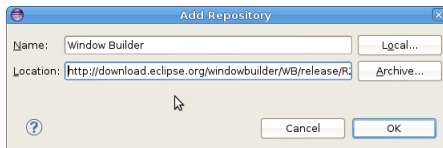


Figura: Instalando Window Builder



# Abrindo Editor do Window Builder

Clique com o botão direito na classe `PrimeiroPrograma.java` . Selecione **Open with** e depois **WindowBuilder Editor**.

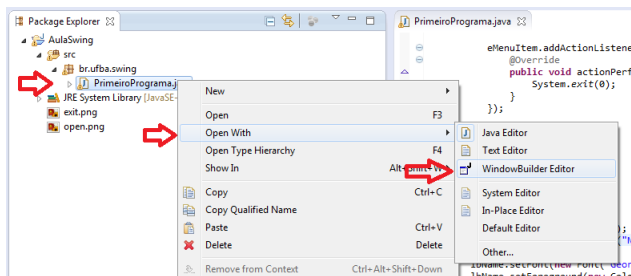


Figura: Abrindo Editor do Window Builder

# Abrindo Editor do Window Builder

- **Source** - Como o Editor padrão do Eclipse.
- **Design** - Editor gráfico do Window Builder.

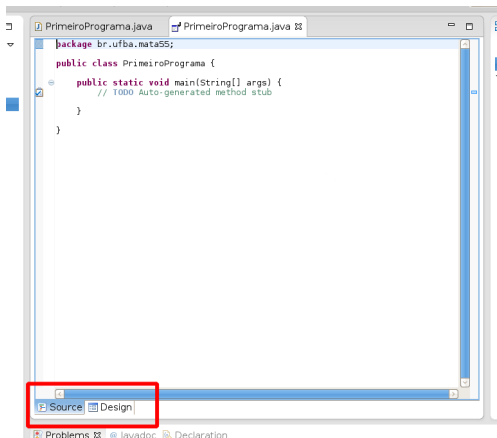


Figura: Editor do Windows Builder

# Aba Design do Editor do Window Builder

- Visualizador gráfico.

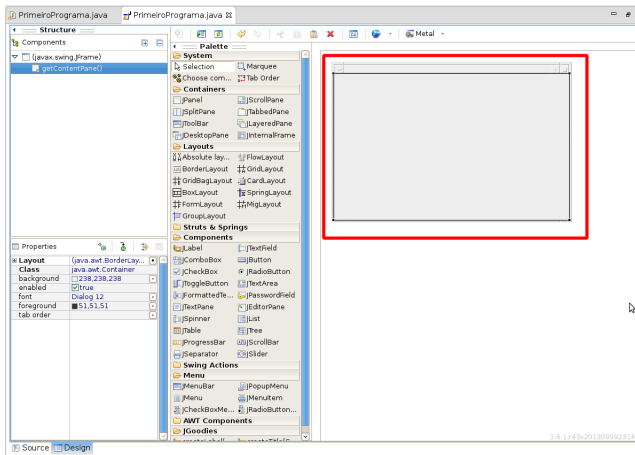
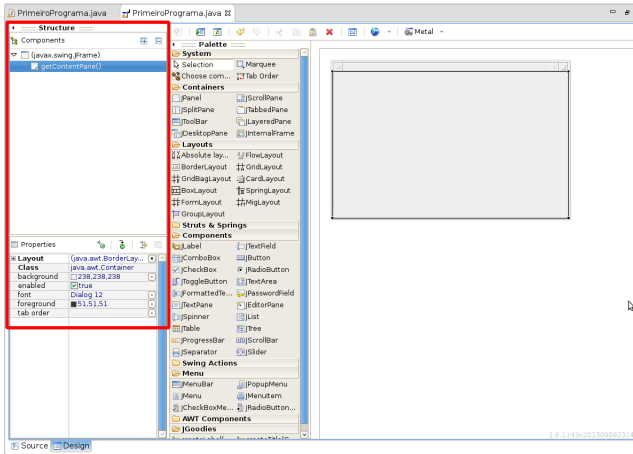


Figura: Aba Design

# Aba Design do Editor do Window Builder

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



### Figura: Estrutura e propriedades

# Construindo a janela

Inserindo um título à janela.

Selecione a janela e insira em **Title** na aba **Properties**.

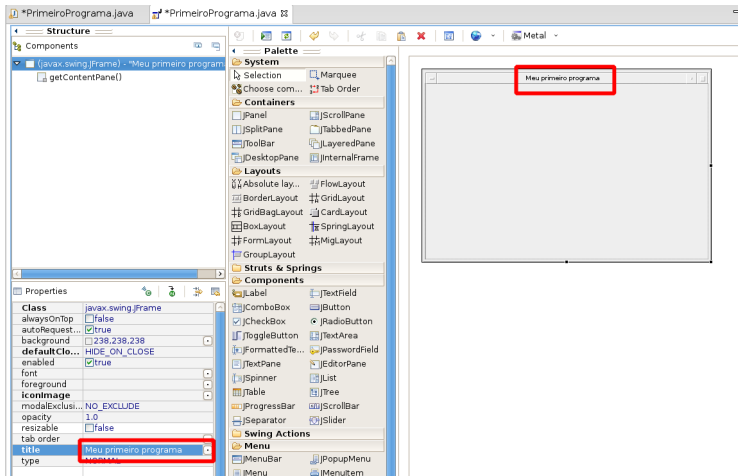


Figura: Estrutura e propriedades

# Construindo a janela

O equivalente em código seria.

---

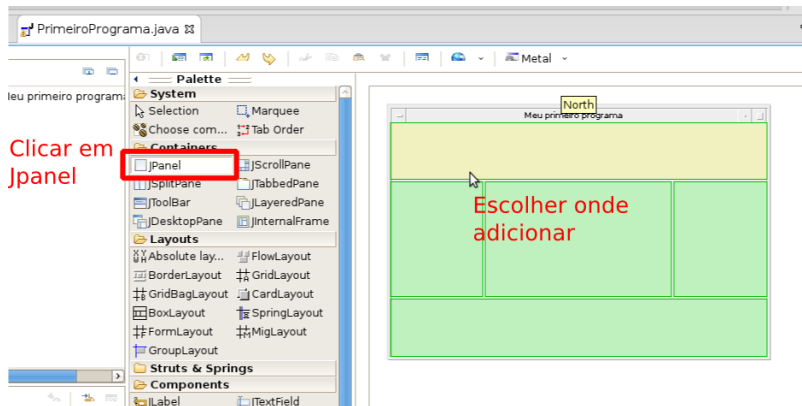
```
public PrimeiroPrograma() {  
    setTitle("Meu primeiro programa");  
}
```

---

# Construindo a janela - JPanel

# Construindo a janela - JPanel

- O Window Builder utiliza o método **Drag & Drop** (clica e solta)
- As áreas verdes são os locais onde o componente pode ser inserido
- As divisões entre as áreas verdes são ditadas pelo **layout** escolhido
- Por padrão o layout é o **BorderLayout** que divide a tela em **NORTH**, **SOUTH**, **WEST**, **CENTER** e **EAST**.





# Construindo a janela - JPanel

Em código:

---

```
package br.ufba.mata55;

import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.BorderLayout;

public class PrimeiroProgramaPura extends JFrame {
    public PrimeiroProgramaPura() {

        JPanel panel = new JPanel();
        getContentPane().add(panel, BorderLayout.NORTH);
    }
}
```

---

# Construindo a janela - **JLabel**

# Construindo a janela - JLabel

- Ao inserir o JLabel já defina o nome!

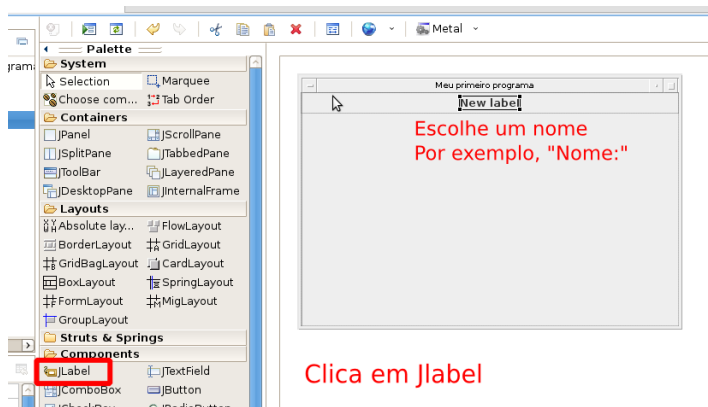


Figura: JLabel

# Construindo a janela - **JTextField**

# Construindo a janela - JTextField

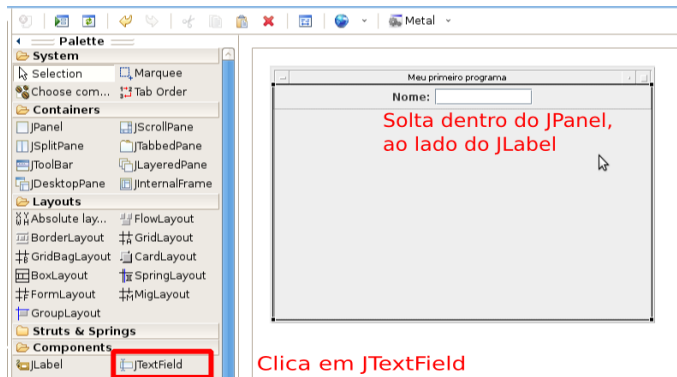


Figura: JTextField

# Construindo a janela - Alinhando os componentes

- Os tipos de alinhamento disponíveis são : **LEFT**, **RIGHT**, **CENTER**, **LEADING**(bottom) e **TRAILING** (top)

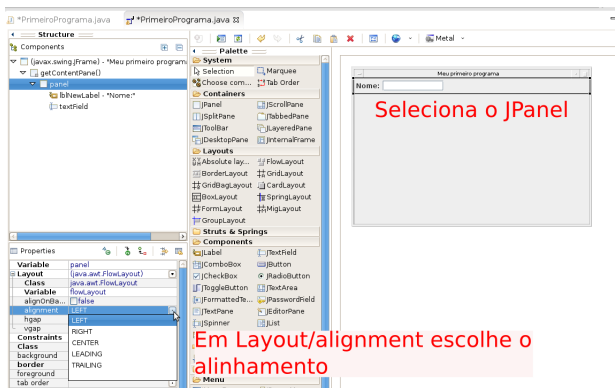


Figura: Alinhando componentes

# Construindo a janela - Alinhamento

Em código:

---

```
JPanel panel = new JPanel();
FlowLayout flowLayout = (FlowLayout) panel.getLayout();
flowLayout.setAlignment(FlowLayout.LEFT);
getContentPane().add(panel, "1, 1, fill, top");

JLabel lblNewLabel = new JLabel("Nome:");
panel.add(lblNewLabel);

textField = new JTextField();
panel.add(textField);
textField.setColumns(10);
```

---

- Window Builder - <https://www.eclipse.org/windowbuilder/>
- Swing tutorial - <http://www.tutorialspoint.com/swing/>
- Swing tutorial(verboso) -  
<http://www.wikihow.com/Create-a-Swing-GUI-in-Java>
- Swing tuto mais completo - <http://zetcode.com/tutorials/javaswingtutorial/>
- Swing oficial - <http://docs.oracle.com/javase/tutorial/uiswing/components/>
- Swing oficial - <http://docs.oracle.com/javase/tutorial/uiswing/>
- Swing javadoc -  
<http://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>