



Curso de Java

GUI – Graphical User Interface
Componentes Swing
WindowBuilder Plugin
CRUD

Ministrado por: Mário Sergio

email: mariosergio30@gmail.com

<https://www.linkedin.com/profile/view?id=111098029>

GUI - Interface Gráfica

- Uma **interface gráfica (GUI)** apresenta um mecanismo amigável ao usuário para interagir com um aplicativo.
- Uma GUI dá ao aplicativo uma **aparência** e **comportamento** distintos.
- Uma boa **interface** facilita o aprendizado e utilização do aplicativo pelo usuário.
- As GUI são construídas a partir de **componentes** GUI.
- Um componente GUI é um **objeto** que permite ao usuário interagir via mouse, teclado, formulário, reconhecimento de voz, etc.
- Muitas **IDEs** fornecem ferramentas de projeto GUI para facilitar a construção da interface.

GUI - JOptionPane

Até agora só tínhamos utilizado GUI com caixas de dialogo que se sobrepõem (pop-ups)...

- É possível utilizar uma **caixa de diálogo** para interagir com o usuário.
- A caixa de diálogo é utilizada para receber informações do usuário ou exibir informações para o usuário.
- A classe **JOptionPane** do pacote **javax.swing** fornece caixas de diálogos pré-construídas tanto para entrada como para saída.
- Esses diálogos são exibidos invocando **métodos** `JOptionPane static`.
- Uma caixa de diálogo modal só deve ser usada quando for necessário impedir usuários de interagir com um restante de um aplicativo até que eles descartem o diálogo.
- Não é necessário a criação de um objeto para utilizar os métodos `showInputDialog`, `showMessageDialog` e `parseInt`. São métodos invocados a partir da classe.

GUI – Swing (interfaces ricas)

- Somente as caixas de diálogo não atendem todos os requisitos de uma boa interface.
- A maioria dos aplicativos exige interfaces mais elaboradas e personalizadas.
- Por isto, há muitos componentes **GUI Swing** no pacote javax.
- A maioria dos componentes são componentes Java puros (são escritos, manipulados e exibidos em Java).
- Estes componentes fazem parte das **Java Foundation Classes (JFC)** – bibliotecas para desenvolvimentos de GUI para múltiplas plataformas.

GUI – Swing (interfaces ricas)

Exemplos:

Cadastro de Livros

Adicionar Livros

Editora: Novatec

Livro: Google Android

ISBN: 978857522447

Lista de Livros

Código	Editora	Título	ISBN
1	Alta Books	Certificação Sun Para Progra...	978857608...
2	Novatec	HTML 5	978857522...
3	Alta Books	Use a Cabeça! Redes de Co...	978857608...
4	Alta Books	Use a Cabeça! Programação	978857608...
5	Novatec	Google Android	978857522...
6	9788575222386	Programação Java para a Web	978857522...
7	Alta Books	Domain-Driven Design	978857608...
8	Alta Books	Dominando Php e Mysql	978857608...
9	Novatec	Padrões Javascript	978857522...
10	Novatec	Google Android Para Tablets	978857522...

GUI – Swing (interfaces ricas)

Exemplos:

Cadastro de Produtos

Clientes Produtos Vendas Fornecedores Relatórios Ordem de serviço Pesquisar Sobre?

Id	Nome	Fornecedor	Preço comp...	Preço	quantidade	Categoria	Marca	Status	Data
1	HD320GB	HARDWARE	200.00	100.00	1	HD	HP	Novo	02/11/2014
2	MONITOR 1...	MARIA DA ...	300.00	200.00	2	MONITOR	SAMSUG	Novo	02/11/2014
3	BLUE RAY	HARDWARE	100.00	25.00	10	HD	HP	Novo	02/11/2014
4	CD	HARDWARE	10.00	5.00	6	ROTEADOR	HP	Novo	02/11/2014

Qtd:

Nome:

Preço de Venda:

Preço de compra:

Categoria:

Fornecedor:

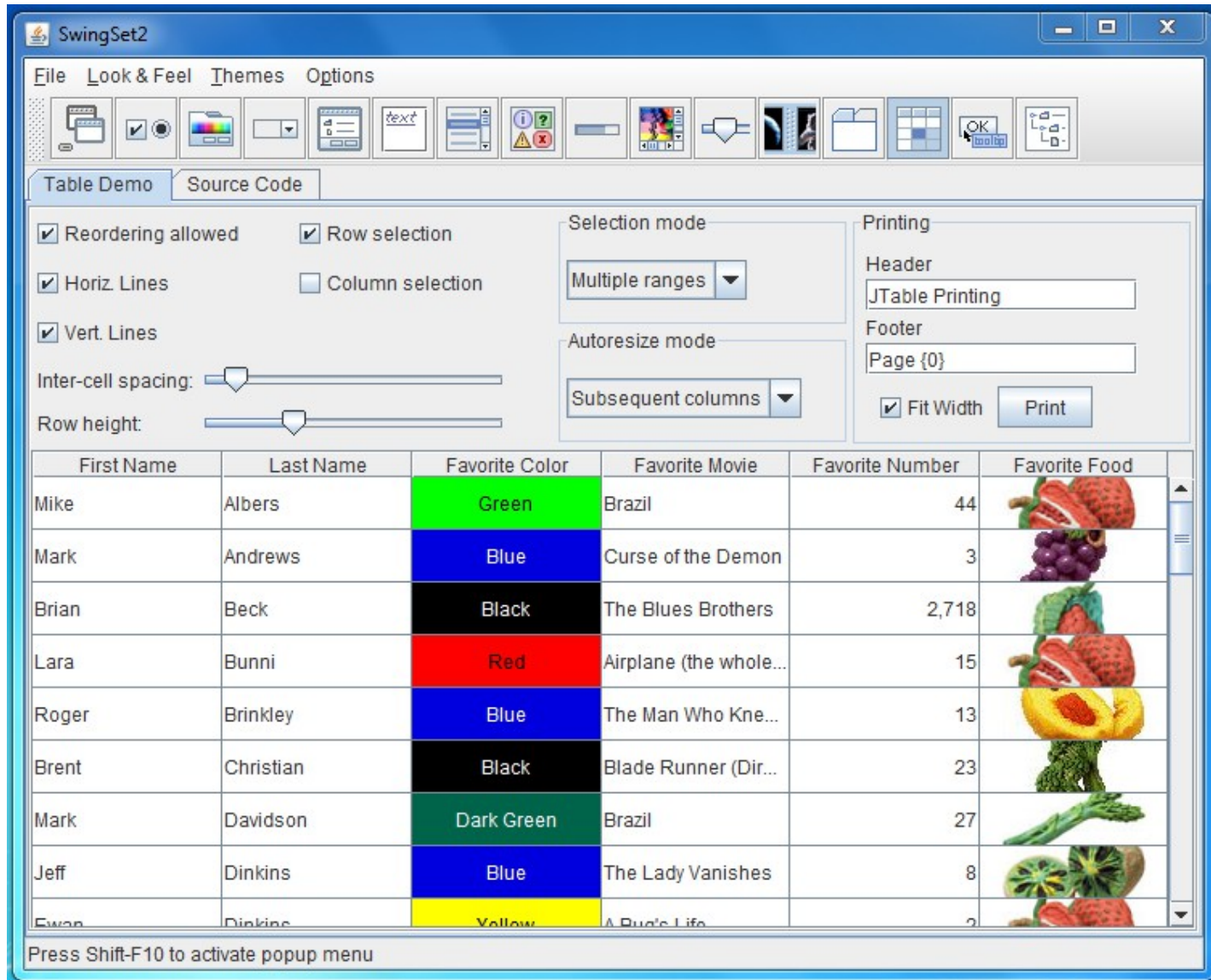
Status:

Modelo/Marca:

Salvar Limpar Alterar Sair

GUI – Swing (interfaces ricas)

Exemplos:



GUI – Swing (interfaces ricas)

- Há dois **pacotes** básicos que auxiliam na criação de componentes: **java.awt** e **javax.swing**.
- Os componentes java.awt são desenhados conforme a plataforma de execução.
- Os componentes java.awt dependem da plataforma local para determinar sua funcionalidade, aparência e comportamento. São componentes pesados.
- Os componentes javax.swing assumem a mesma aparência em qualquer plataforma. São componentes leves.
- A principal hierarquia de classes utilizadas na construção de GUI é descrita abaixo:
 - Object >> Component >> Container >> JComponent

GUI – Swing – Classe JFrame

Criando uma Tela GUI → Código de uma classe que herda um JFrame:

```
package visao;
import java.awt.*;
import javax.swing.*;

public class MinhaTela extends JFrame {

    public MinhaTela() {

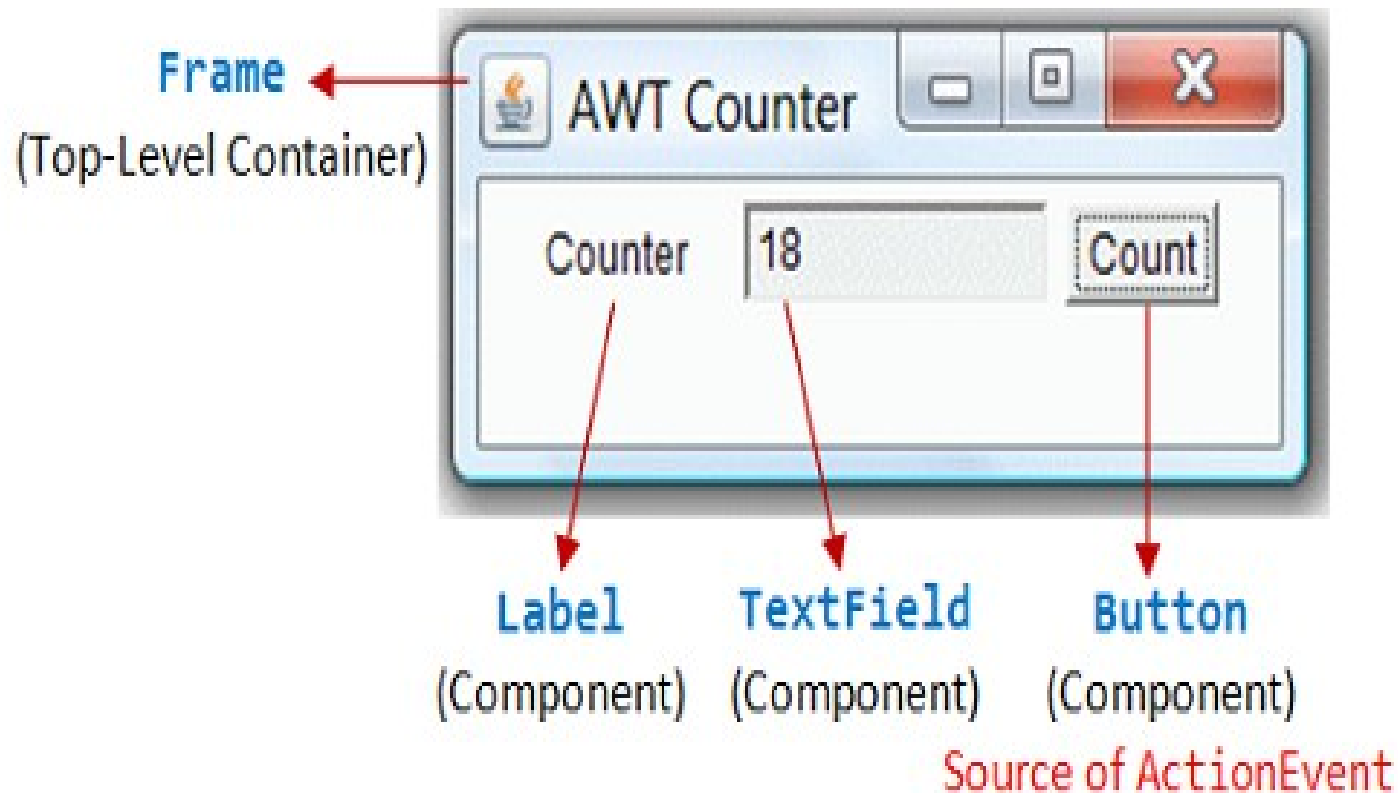
        this.setLayout(new FlowLayout());
        setBounds(100, 100, 450, 300);

        JButton botao = new JButton("Count");
        JLabel lb = new JLabel("Counter");
        JTextField edit = new JTextField("");
        edit.setColumns(10);

        this.add(lb);
        this.add(edit);
        this.add(botao);
    }
}
```

GUI – Swing

Componentes básicos:



GUI – Swing

Explicando os Componentes básicos:

- Um **JFrame** é uma janela com uma barra de título e uma borda.
- A classe JFrame é uma subclasse de Frame. Frame é uma subclasse de Window.
- Uma janela gerada pelo JFrame é exibida conforme o padrão de janelas do sistema operacional em uso.
- Por padrão, quando o usuário fecha uma janela JFrame, ela se torna oculta, mas é possível controlar isto com o método **setDefaultCloseOperation()**.

GUI – Swing

Explicando os Componentes básicos:

- A maioria das janelas podem conter componentes Swing e são instâncias da classe **JFrame** ou uma subclasse de JFrame.
- JFrame é uma subclasse indireta de java.awt.Window e descende diretamente de java.awt.Frame.
- A classe java.awt.Window fornece atributos e comportamentos básicos de uma janela.
 - barra de título na parte superior
 - botões para minimizar, maximizar e fechar a janela.
- Um **JLabel** (rótulo) fornece um texto que declara a finalidade de cada componente dentro de uma janela. Pode exibir apenas texto, uma imagem ou uma combinação de imagem com texto.

GUI – Swing

Explicando os Componentes básicos:

- O gerenciador de layout **FlowLayout** organiza os componentes da esquerda para a direita. Sempre reorganiza os componentes quando as dimensões do container (JFrame) são alteradas.
- O método **setLayout** é utilizado para definir qual gerenciador de layout deve ser utilizado.
- Todos os componentes devem ser adicionados explicitamente dentro do contêiner.
- Um **Icon** é um objeto que implementa a interface `javax.swing.Icon`. A classe **ImageIcon** é um exemplo.
- A classe `ImageIcon` suporta vários formatos de imagem (GIF, PNG, JPEG).

GUI – Swing

Explicando os Componentes básicos:

- A classe **JText** estende a classe `JTextComponent` que fornece muitos recursos comuns aos componentes baseados em texto.
- A classe **JPasswordField** estende a classe `JTextField` e adiciona métodos de processamento de senhas.
- Ambos os componentes representam uma área de uma única linha onde o usuário pode inserir um texto.
- Um `JPasswordField` mostra que caracteres estão sendo digitados, mas oculta os caracteres reais.
- Quando o usuário digita um texto em um dos componentes e tecla **ENTER** um evento ocorre.
- O texto só pode ser digitado para o componente que estiver em **foco**.
- Um **componente** recebe o foco quando o usuário clica no componente. Isto é importante para capturar corretamente a origem do evento.

GUI – Swing

Principais Componentes:



Buttons



Combo Box



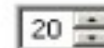
List



Menu



Slider



Spinner



Text Fields



Label



Progress Bar



Tool tip

GUI – Swing

Principais Componentes:



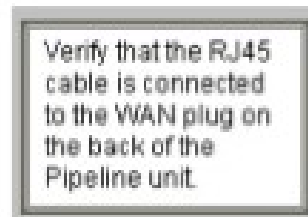
Color Chooser



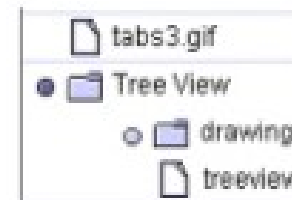
File Chooser

First Name	Last Name
Mark	Andrews
Tom	Ball
Alan	Chung
Jeff	Dinkins

Table



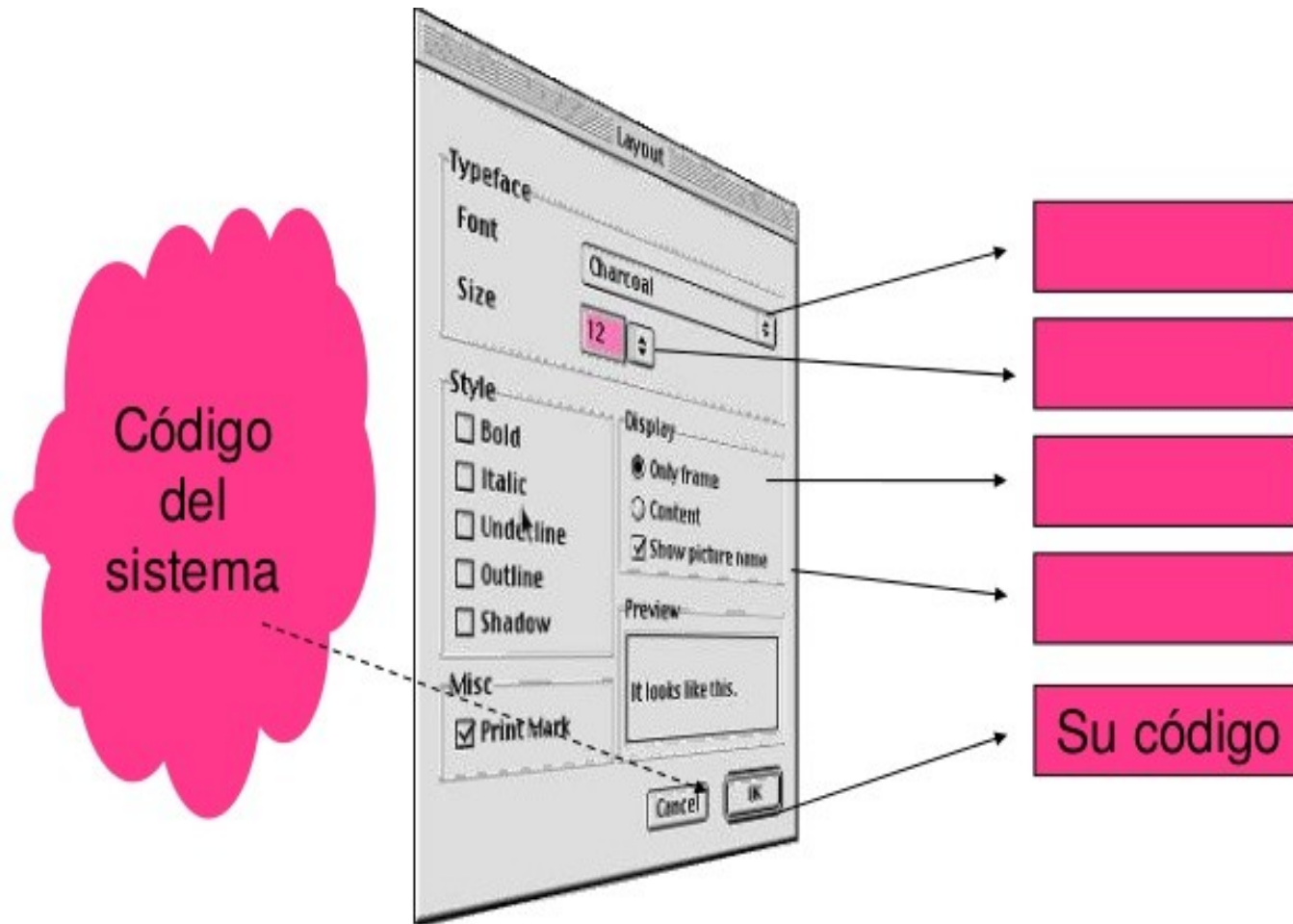
Text



Tree

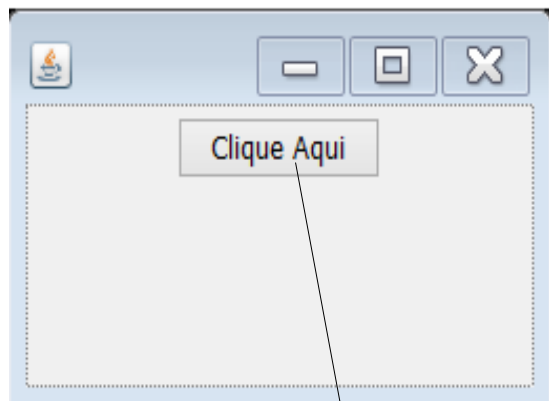
GUI – Swing - Eventos

A interface gráfica é só uma **Casca**, que serve para o usuário final utilizar o seu programa.

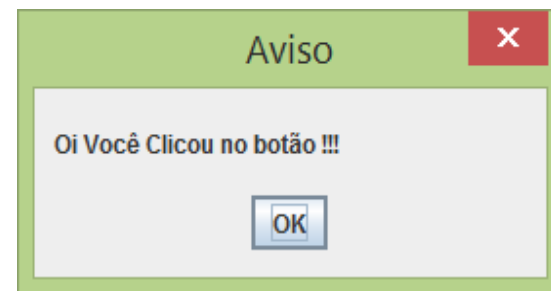


GUI – Swing - Eventos

Se você desejar que **ALGO OCORRA** em resposta a uma interação (ação) do usuário, é preciso **TRATAR OS EVENTOS**.



```
 JButton botao = new JButton("Count");  
  
 botao.addActionListener(new ActionListener() {  
     public void actionPerformed(ActionEvent e) {  
         JOptionPane.showMessageDialog(null, "Oi Você Clicou no botão !!!", "Aviso",  
             JOptionPane.PLAIN_MESSAGE);  
     }  
 });
```



Os Eventos são tratados a partir de **Listeners** (escutadores), que ficam o tempo todo **"ouvindo"** o que se passa na interface com o usuário. Os Listeners avisam para o seu programa que **ocorreu um Evento**, para que seu programa decida o que deve ser feito.

GUI – Swing - Eventos

- Cada interface listener de eventos especifica um ou mais métodos de **tratamento de evento** que devem ser declarados na classe que implementa a interface.
- Cada classe que implementa uma interface deve declarar (definir o código) todos os métodos dessa interface.
- Cada JComponent tem uma variável de instância chamada **listenerList** que referencia um objeto de classe **javax.swing.event.EventListenerList** para registrar os seus ouvintes (listeners).
- Um componente pode ter vários ouvintes registrados para processar os seus eventos.
- Cada tipo de evento é tratado por uma interface listener apropriada:
 - ActionEvent é tratado por ActionListener
 - MouseEvent é tratado por MouseListener e MouseMotionListener
 - KeyEvent é tratado por KeyListener

GUI – Swing - Eventos

Principais Eventos de Mouse:

- **mousePressed**(MouseEvent event) : quando um botão do mouse é pressionado enquanto o cursor do mouse estiver sobre um component.
- **mouseClicked**(MouseEvent event) : quando um botão do mouse é pressionado e liberado enquanto o cursor do mouse pairar sobre um componente. Esse evento é sempre precedido por uma chamada para mousePressed.
- **mouseReleased**(MouseEvent event) : quando um botão do mouse é liberado depois de ser pressionado. do mouse estiver sobre um component. Esse evento é sempre precedido por uma chamada para mousePressed e uma ou mais chamadas para mouseDragged.
- **mouseEntered**(MouseEvent event) : quando o cursor do mouse entra nos limites de um componente.
- **mouseExited**(MouseEvent event) : quando o curso do mouse deixa os limites de um componente.

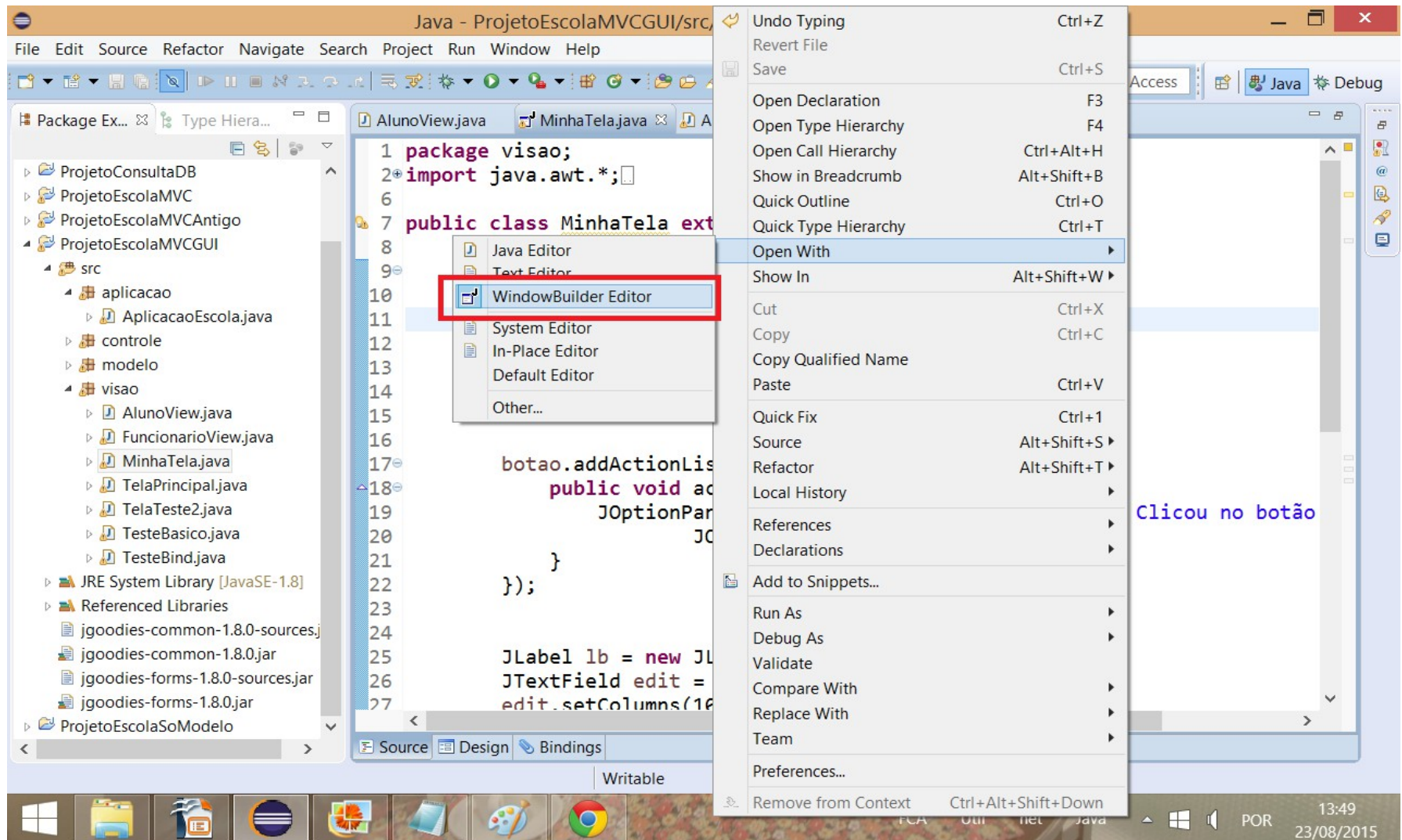
GUI – Swing - Eventos

Principais Eventos de Teclado:

- Os eventos do teclado são tratados por uma classe que implementa **KeyListener**.
- Eventos de teclado são gerados quando teclas são pressionadas e liberadas.
- Os métodos `keyPressed()`, `keyReleased()` e `keyTyped()` recebem um **KeyEvent** como argumento e definem o código para tratamento do evento.
 - **keyPressed()** é invocado quando qualquer tecla é pressionada.
 - **keyTyped()** é invocado quando uma tecla que não é de ação é invocada (home, end, setas, etc).
 - **keyReleased()** é chamado quando a tecla é lançada após um dos métodos anteriores.

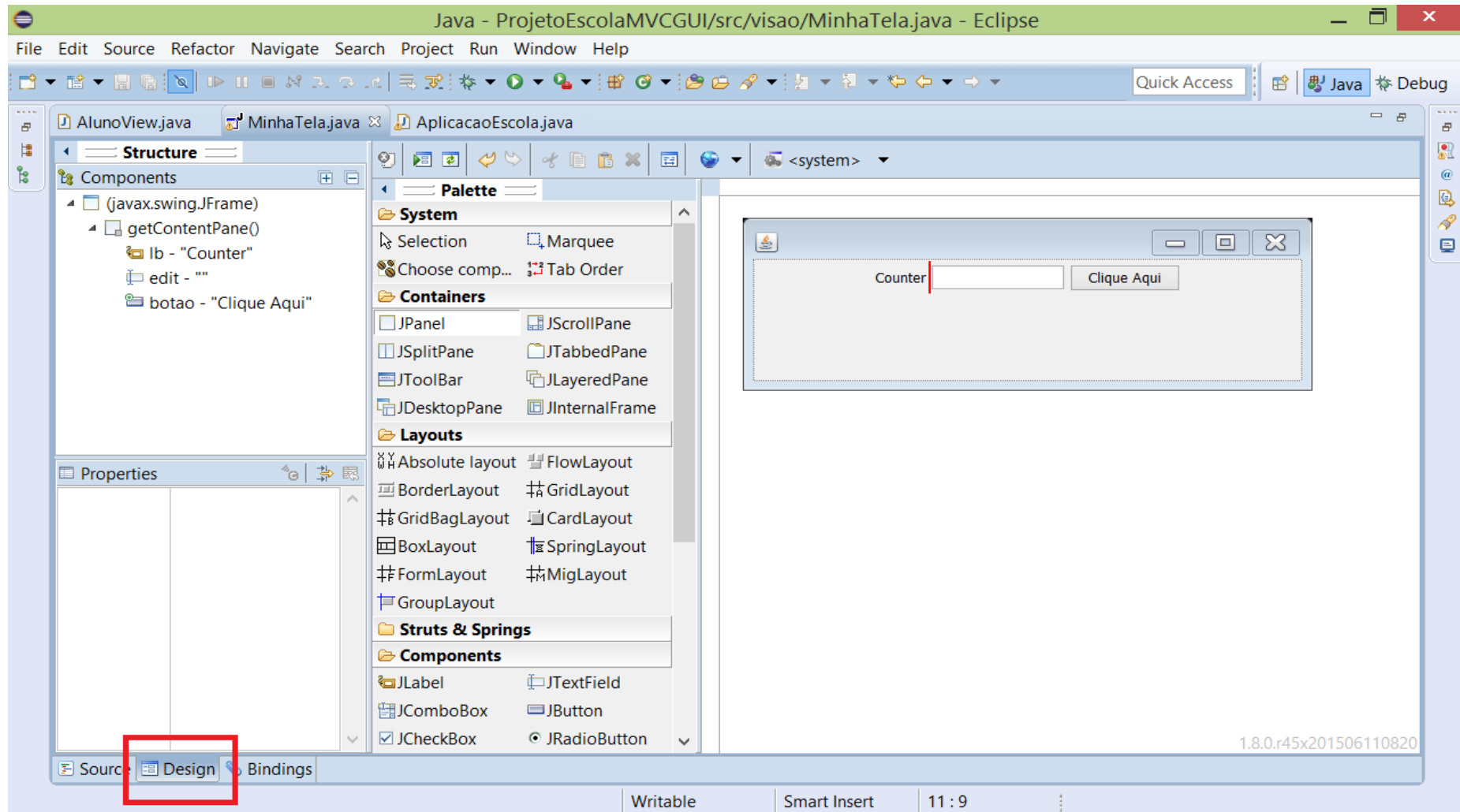
GUI – Swing - WindowBuilder

Interface gráfica sem sofrimento → Clique com o botão direito do mouse sobre o seu código fonte:



GUI – Swing - WindowBuilder

Interface gráfica sem sofrimento → Desenhando telas com o mouse:



GUI – Swing - WindowBuilder

O WindowBuilder Pro

É um framework que provê uma rica API para construção de interfaces gráficas. A empresa que iniciou o desenvolvimento do projeto foi incorporada ao Google no ano de 2010, que a partir de então começou a distribuir o conjunto de componentes visuais gratuitamente. O projeto atualmente está disponível em <http://www.eclipse.org/windowbuilder/>.

Além da rica API o que chama atenção é a vasta documentação de referência disponível para os desenvolvedores e pessoas interessadas em contribuir com o desenvolvimento do framework, sendo este composto por:

- Editor de código-fonte
- Janela de desenho de interface gráfica
- Árvore de componente ou Inspector
- Painel de propriedades simples e avançado
- Paleta de componentes visuais
- Wizards para facilitar a criação de componentes visuais
- Toolbars e menus de contexto

Excelente tutorial para instalação do WindowBuilder

<http://blog.coding4fun.com.br/2014/01/interfaces-graficas-swing-em-java-usando-windowbuilder-pro/>

Janelas do WindowBuilder

Palette of GUI components

Hierarchal list of all GUI components in use.

Click here to see events tied to the selected component.

Properties of the selected component.

Graphical view of design.

Select between graphical and textual views of GUI

Structure

- Components
 - (javax.swing.JFrame)
 - contentPane
 - panel
 - btnClickMe - "Click Me!"
 - lblText - "A label of text"
 - txtThisIsA - "This is a big text area"

Palette

- System
 - Selection
 - Choose com...
 - Containers
 - JPanel
 - JSplitPane
 - JToolBar
 - JDesktopPane
 - Layouts
 - Absolute lay...
 - BorderLayout
 - GridBagLayout
 - BoxLayout
 - FormLayout
 - GroupLayout
 - Struts & Springs
 - Components
 - JLabel
 - JComboBox
 - JCheckBox
 - JToggleButton
 - JFormattedT...
 - JTextPane
 - JSpinner
 - JTable
 - JProgressBar
 - JSeparator
 - Swing Actions
 - Menu
 - AWT Components
 - JGoodies
 - createLabel(...)
 - createTitle(S...

Properties

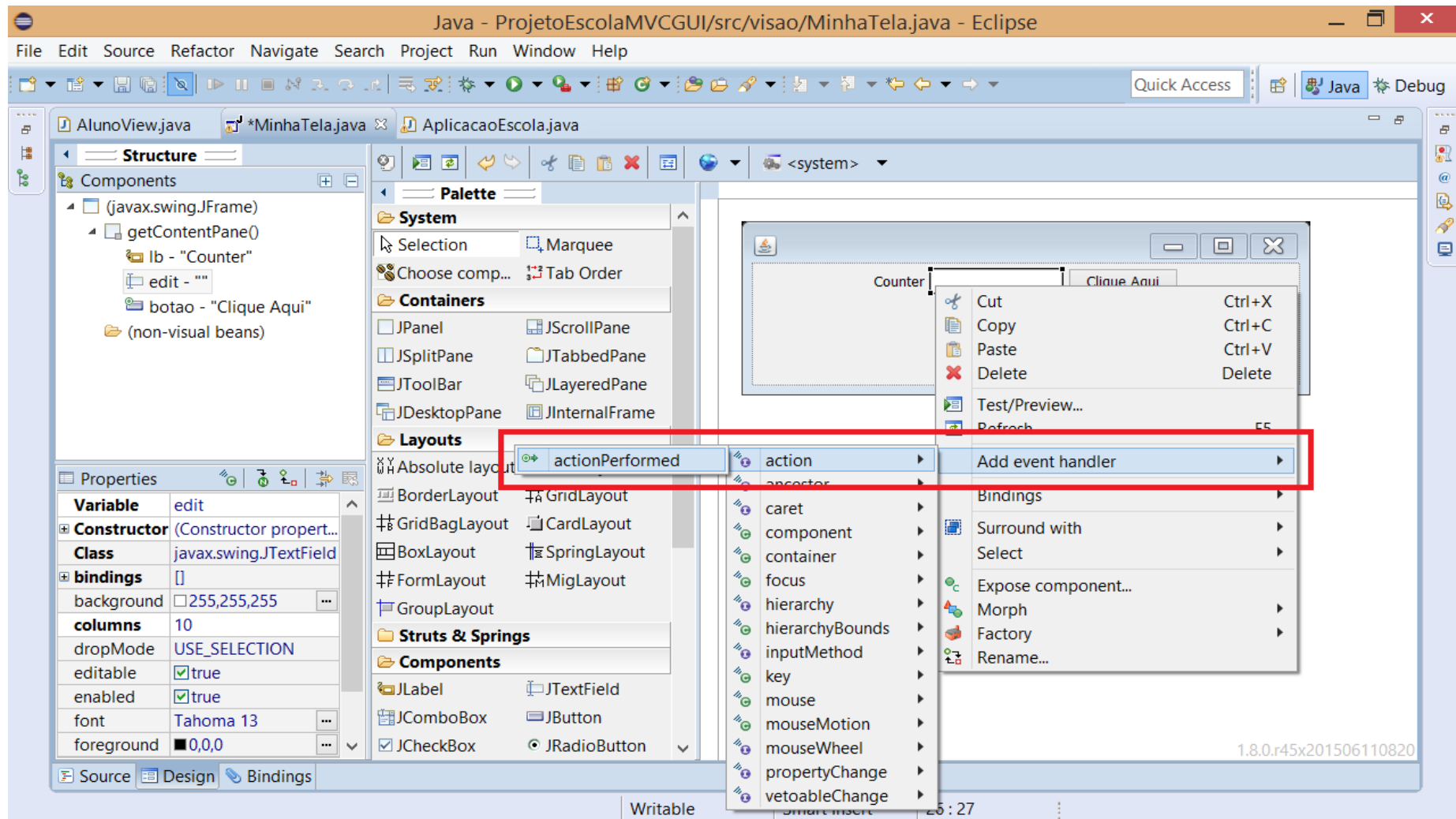
Variable	btnClickMe
Construc...	(Constructor properties)
Class	javax.swing.JButton
backgrou...	240,240,240
enabled	<input checked="" type="checkbox"/> true
font	Tahoma 11
foreground	0,0,0
horizonta...	CENTER
icon	
mnemon...	
selectedc...	
text	Click Me!
toolTipText	
verticalAl...	CENTER

Design

WindowBuilder
1.0.0.r37x201106081531

GUI – Swing - WindowBuilder

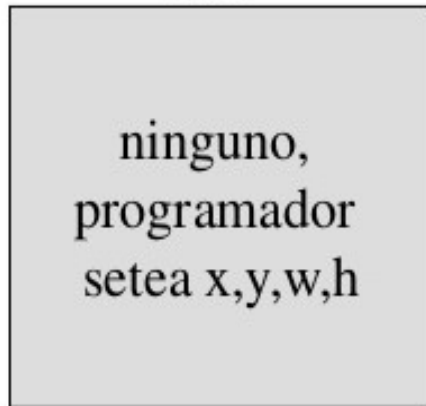
Adicionando Tratamento de Eventos → Clique com o botão direito do mouse sobre o componente desejado:



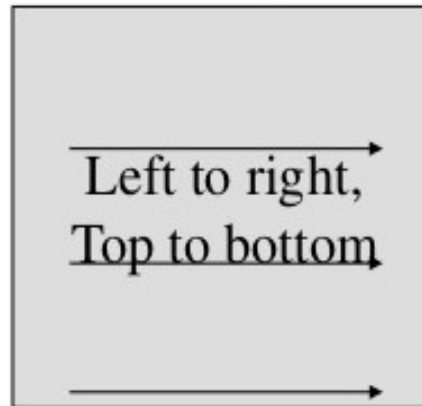
GUI – Swing - Layouts

Gerenciadores de Layout:

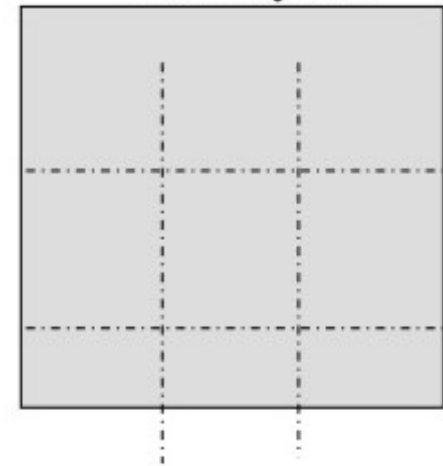
null



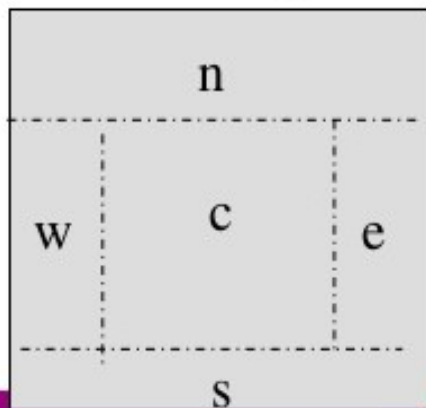
FlowLayout



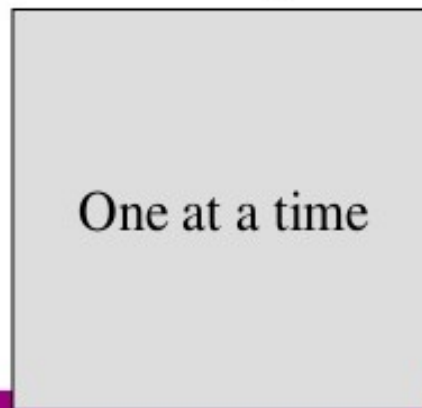
GridLayout



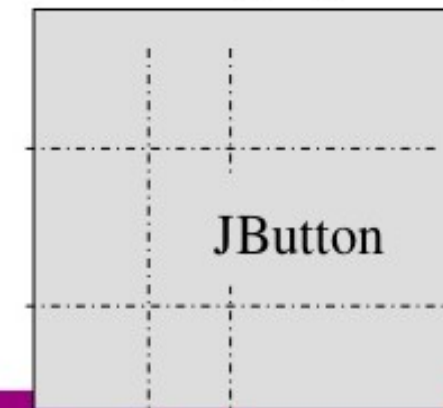
BorderLayout



CardLayout



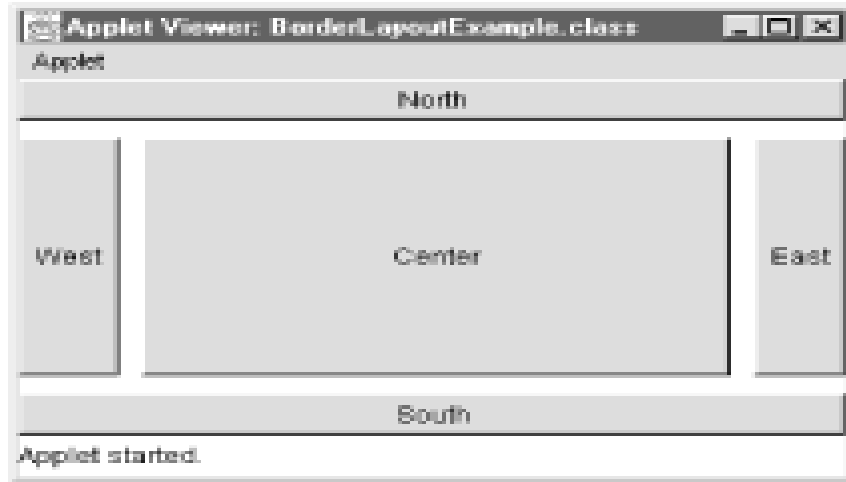
GridBagLayout



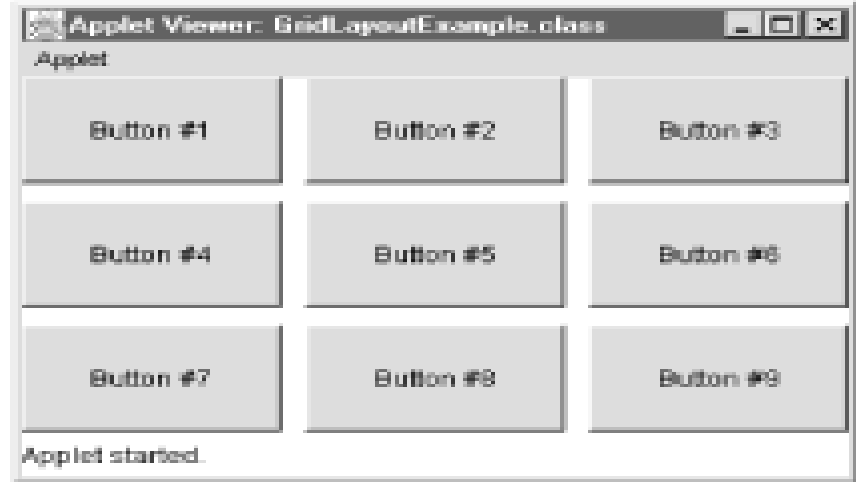
GUI – Swing - Layouts

Gerenciadores de Layout:

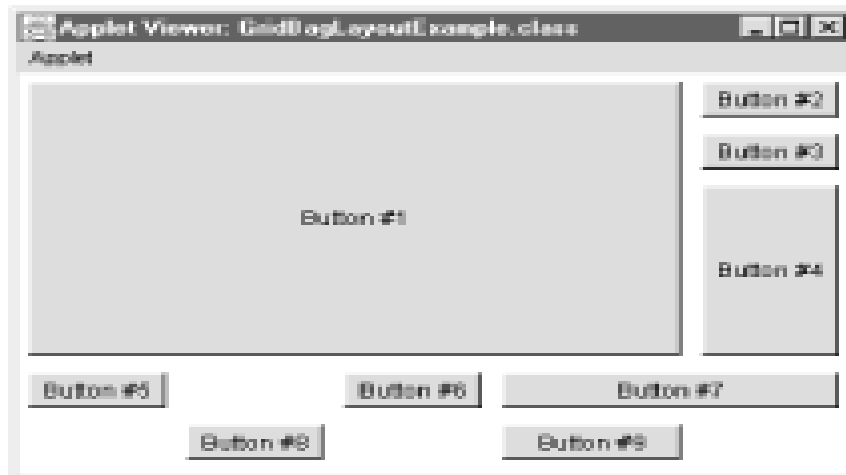
Exemplo da disposição dos componentes dentro dos Layouts:



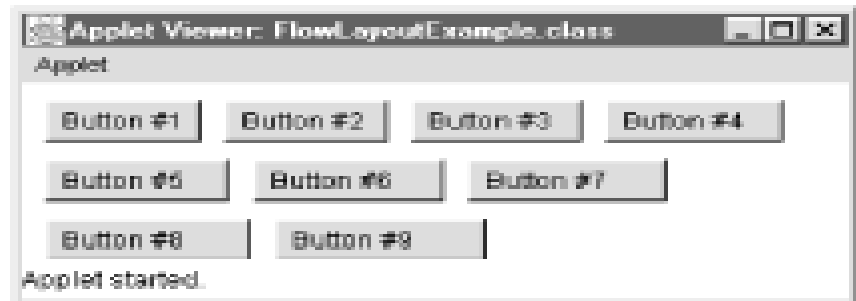
BorderLayout



GridLayout



GridBagLayout



FlowLayout

GUI – Swing - Layouts

Gerenciadores de Layout:

- Os **gerenciadores de layout** organizam os componentes GUI em um contêiner para propósito de apresentação.
- Os gerenciadores ajudam a organizar os elementos definindo de forma automática as dimensões e posição de cada componente.
- Todos os gerenciadores de layout implementam a interface **java.awt.LayoutManager**.
- O método **setLayout()** da classe Container aceita um objeto que implementa a interface LayoutManager como argumento.

GUI – Swing - Layouts

Gerenciadores de Layout:

- Há basicamente três maneiras de organizar componentes em uma GUI.
 - **Posicionamento absoluto:** você pode configurar o layout com null e utilizar os métodos setSize e setLocation ou setBounds para configurar cada componente. É mais trabalhoso.
 - **Gerenciadores de layout:** posiciona os elementos de forma mais simples e rápida, mas perdemos um pouco o controle sobre as dimensões e posicionamento dos elementos.
 - **Posicionamento visual em uma IDE:** algumas IDEs fornecem ferramentas para construir layout com arrastar-e-soltar. Além disto, é possível programar os eventos usando clique duplo sobre os componentes. Neste caso, a IDE gera o código Java necessário. A IDE fornece um controle bom sobre posição, tamanho e alinhamento dos elementos.

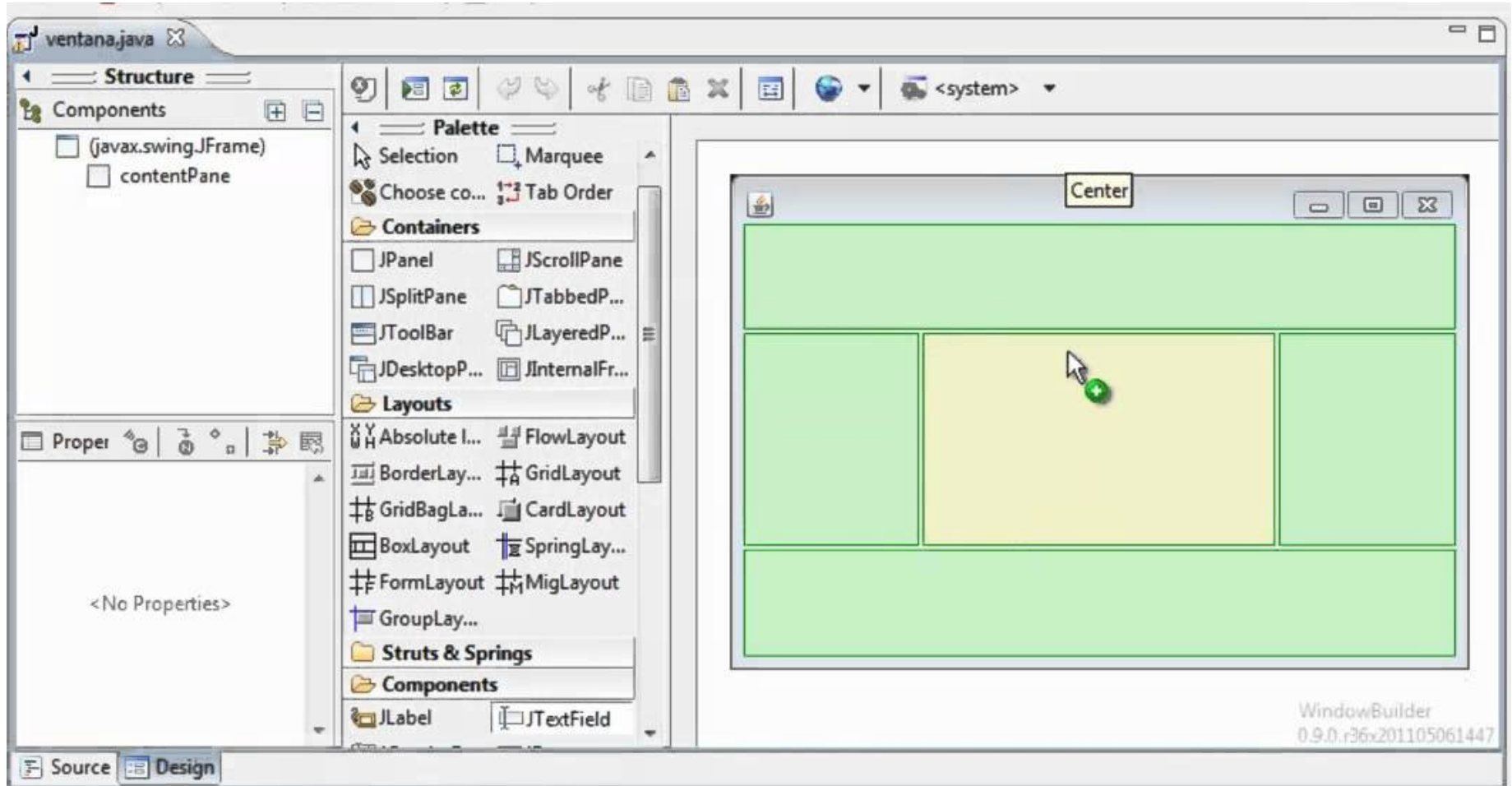
GUI – Swing - Layouts

Gerenciadores de Layout mais comuns:

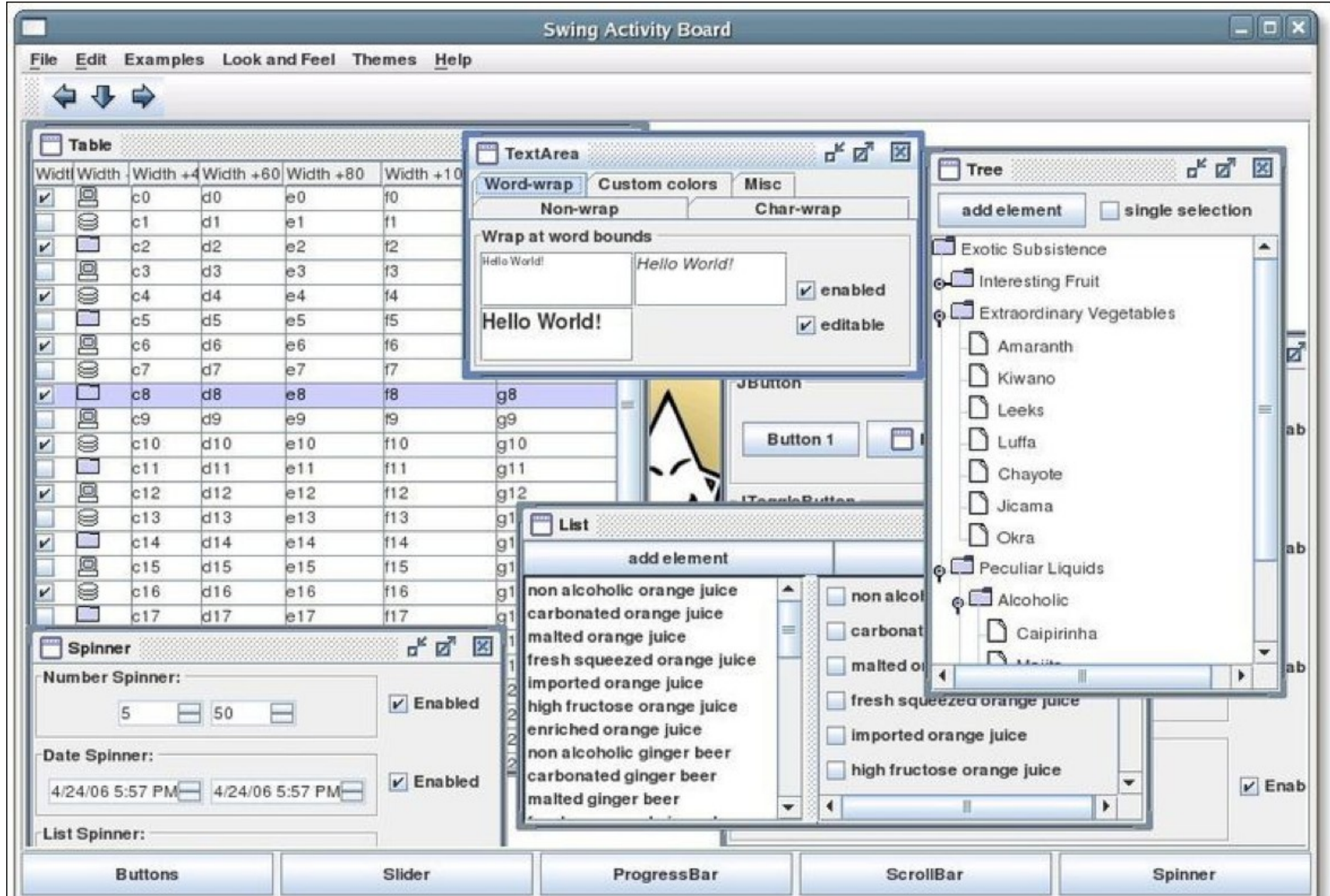
- **FlowLayout** : todos os componentes são colocados em sequência da esquerda para a direita na ordem em que foram adicionados. É o padrão para JPanel.
- **BorderLayout**: organiza os componentes em cinco áreas: NORTH, SOUTH, EAST, WEST, CENTER. É o padrão para JFrame.
- **GridLayout**: organiza os componentes em linhas e colunas.

GUI – Swing - Layouts

Exemplo do BorderLayout no WindowsBuilder:



GUI – Swing – Telas Complexas



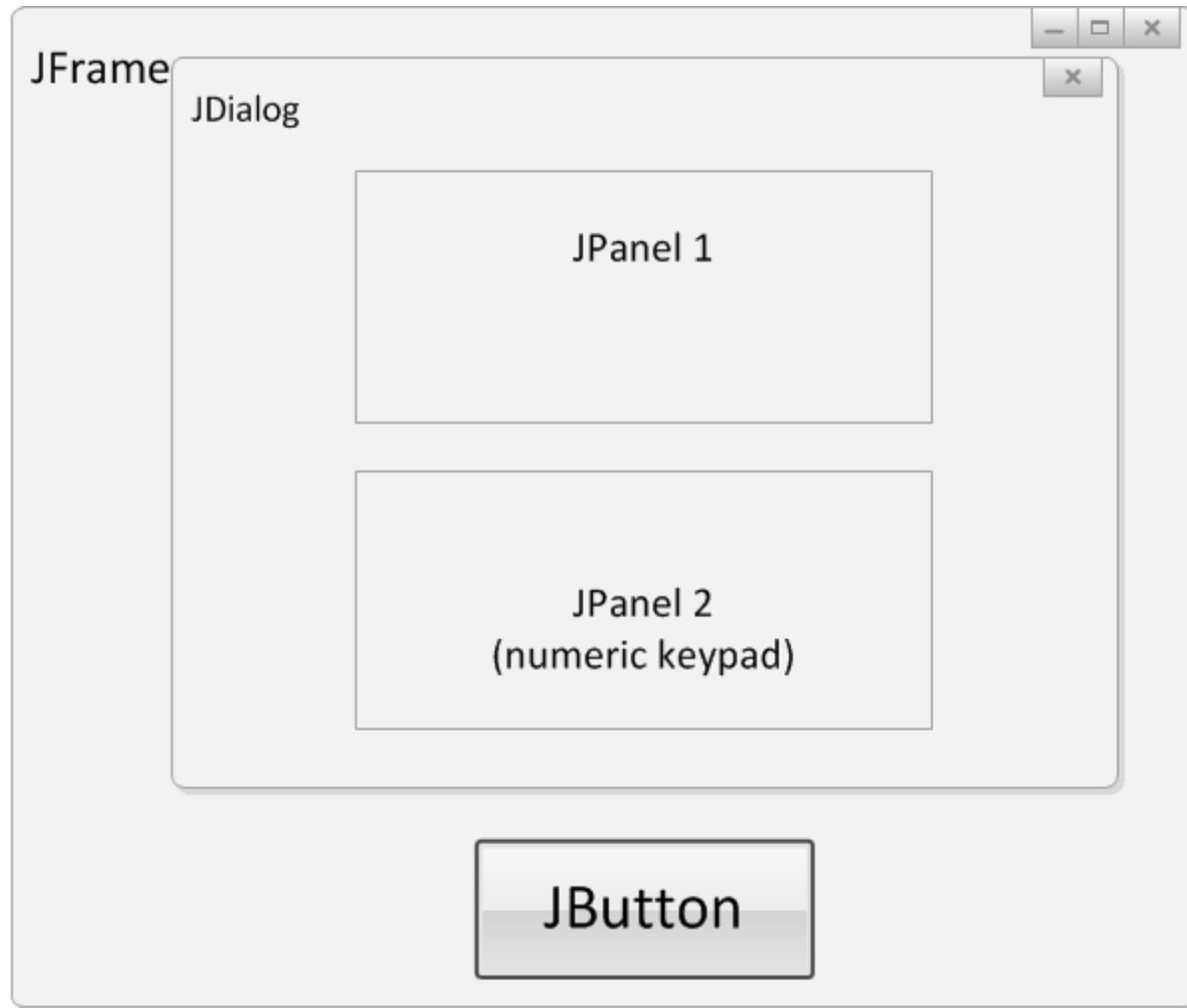
GUI – Swing – Telas Complexas

Além dos **Layouts** também precisamos utilizar **Containers**, eles servem para **conter** (delimitar o espaço) de outros componentes dentre dele.



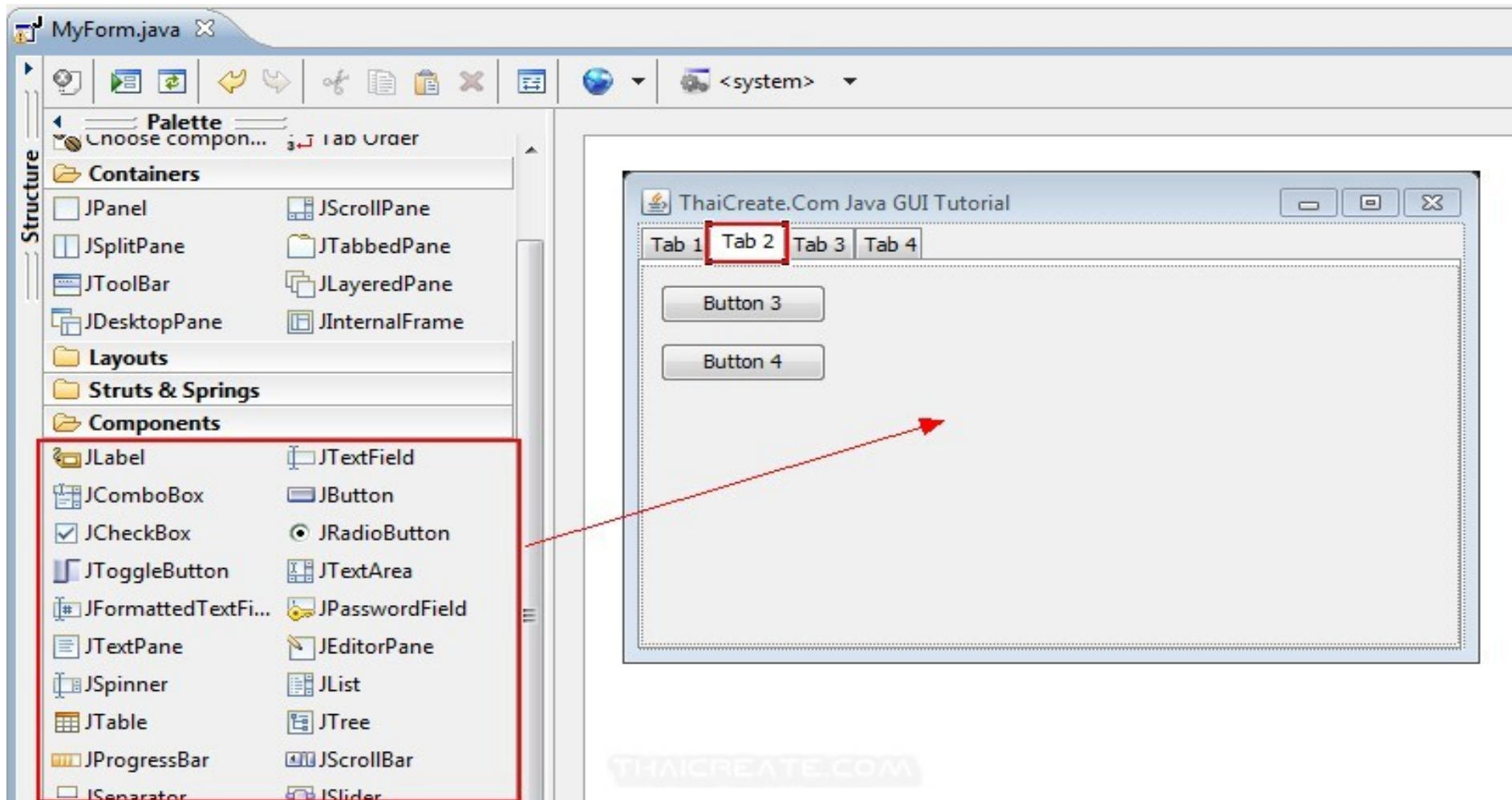
GUI – Swing – Telas Complexas

O **JPanel** (Painel) é um dos contêineres mais utilizados:



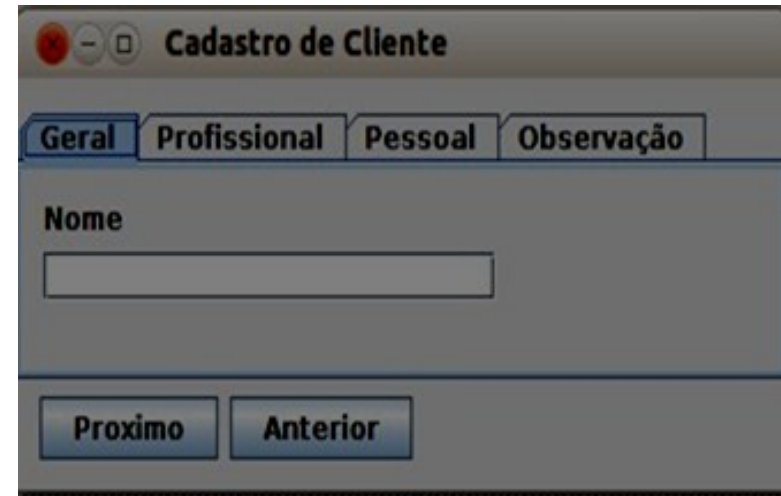
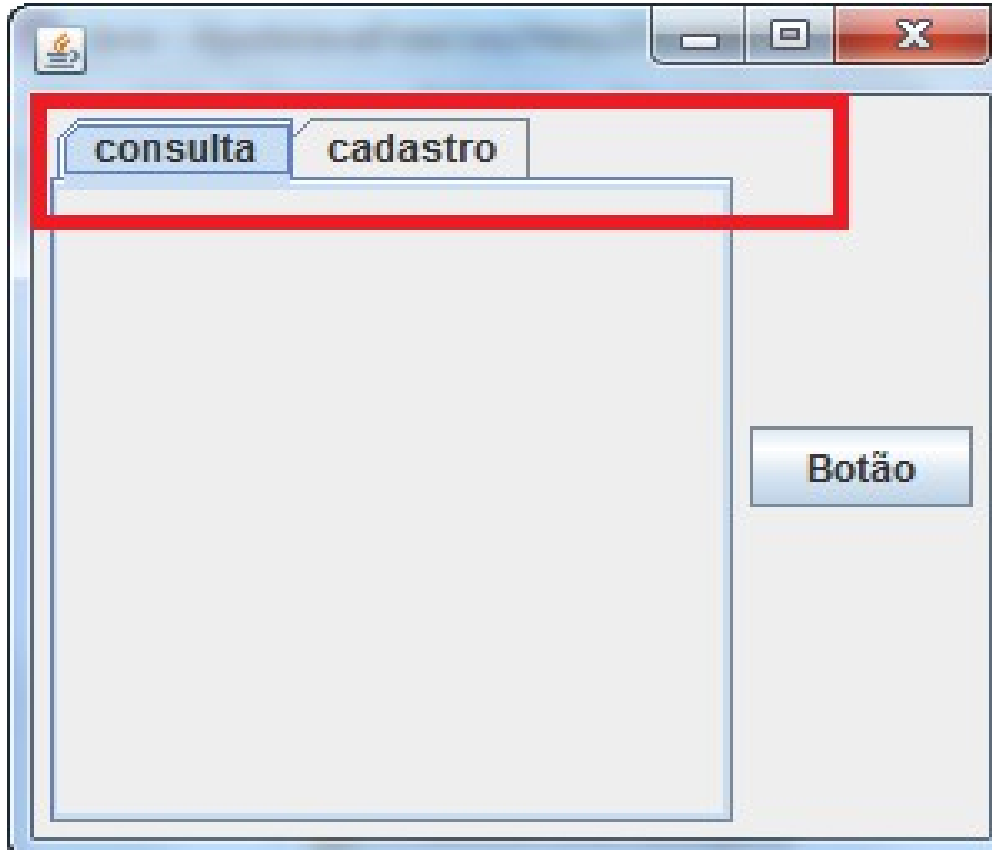
GUI – Swing – Telas Complexas

O **JTabbedPane** permite a exibição de muitos componentes em um mesmo Frame, acessíveis a partir do clique em suas **Abas**, diminuindo a necessidade de criar várias janelas.



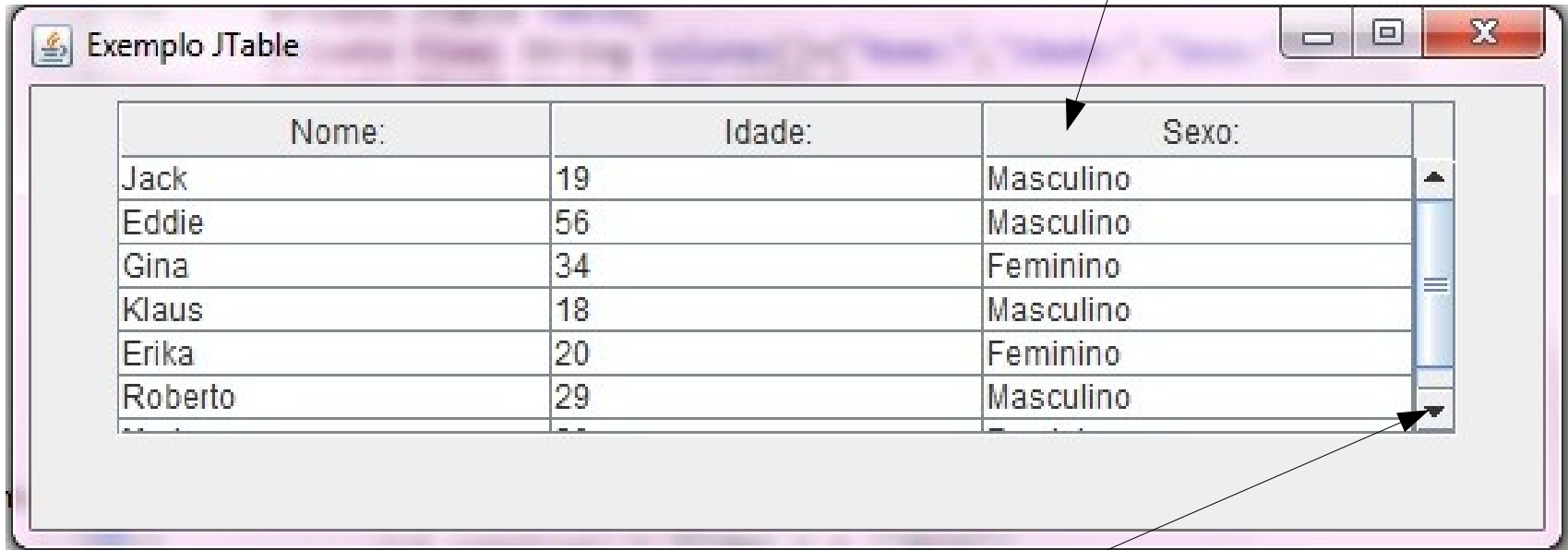
GUI – Swing – Telas Complexas

O **JTabbedPane** permite a exibição de muitos componentes em um mesmo Frame, acessíveis a partir do clique em suas **Abas**, diminuindo a necessidade de criar várias janelas.



GUI – Swing – Telas Complexas

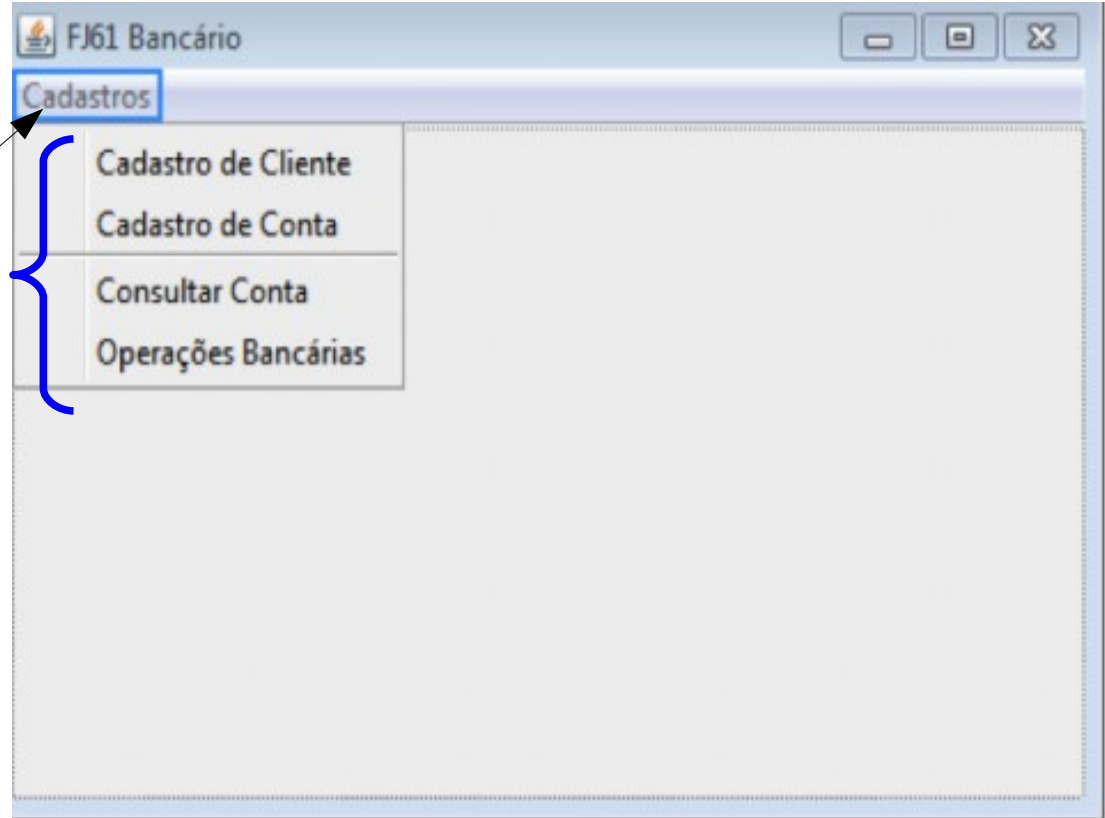
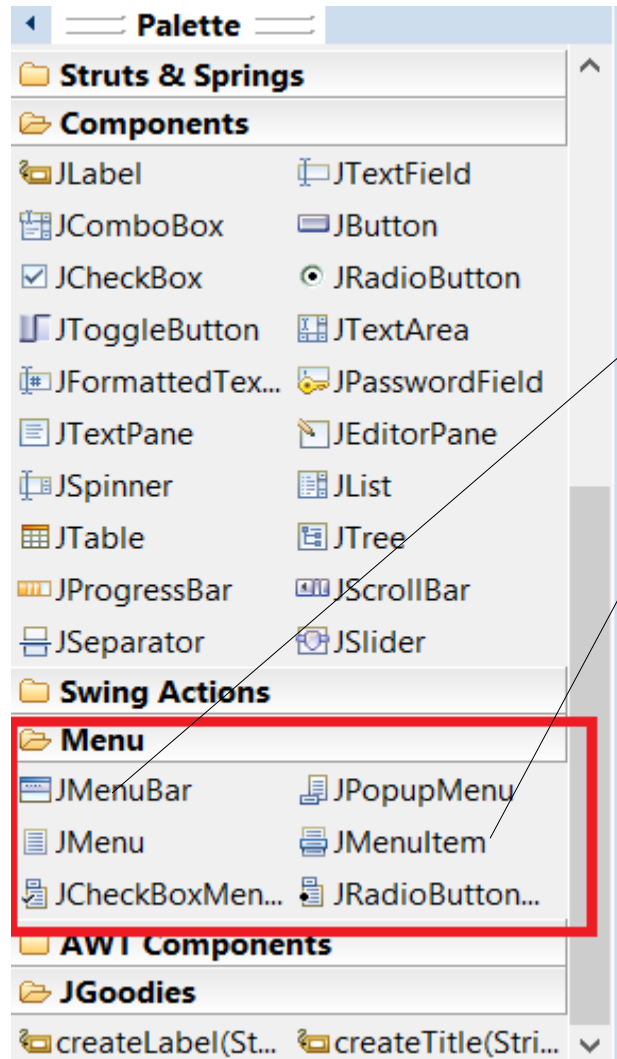
O **JScrollPane** fornece **Barras de Rolagem** verticais e horizontais. Por isso elas são ideais para conter componentes como o **JTable**.



O componente JTable está dentro de JScrollPane, somente por isso é que as Barras de Rolagem aparecem quando a tabela possui muitas linhas/colunas.

GUI – Swing – Telas Complexas

Menus:



GUI – Swing – Telas Complexas

Menus:

- A classe **JMenuBar** possui métodos para gerenciar uma barra de menus (container de menus).
- A classe **JMenu** possui métodos para gerenciar menus.
- Os menus contêm itens de menu e são adicionados a barras de menus ou a outros menus (submenus).
- Quando um menu é clicado, ele se expande para mostrar sua lista dos itens de menu.
- A classe **JMenuItem** contém os métodos para gerenciar itens de menu.
- Quando um item de menu é selecionado resulta em um evento de ação.

GUI – Swing – Telas Complexas

PopUpMenu:

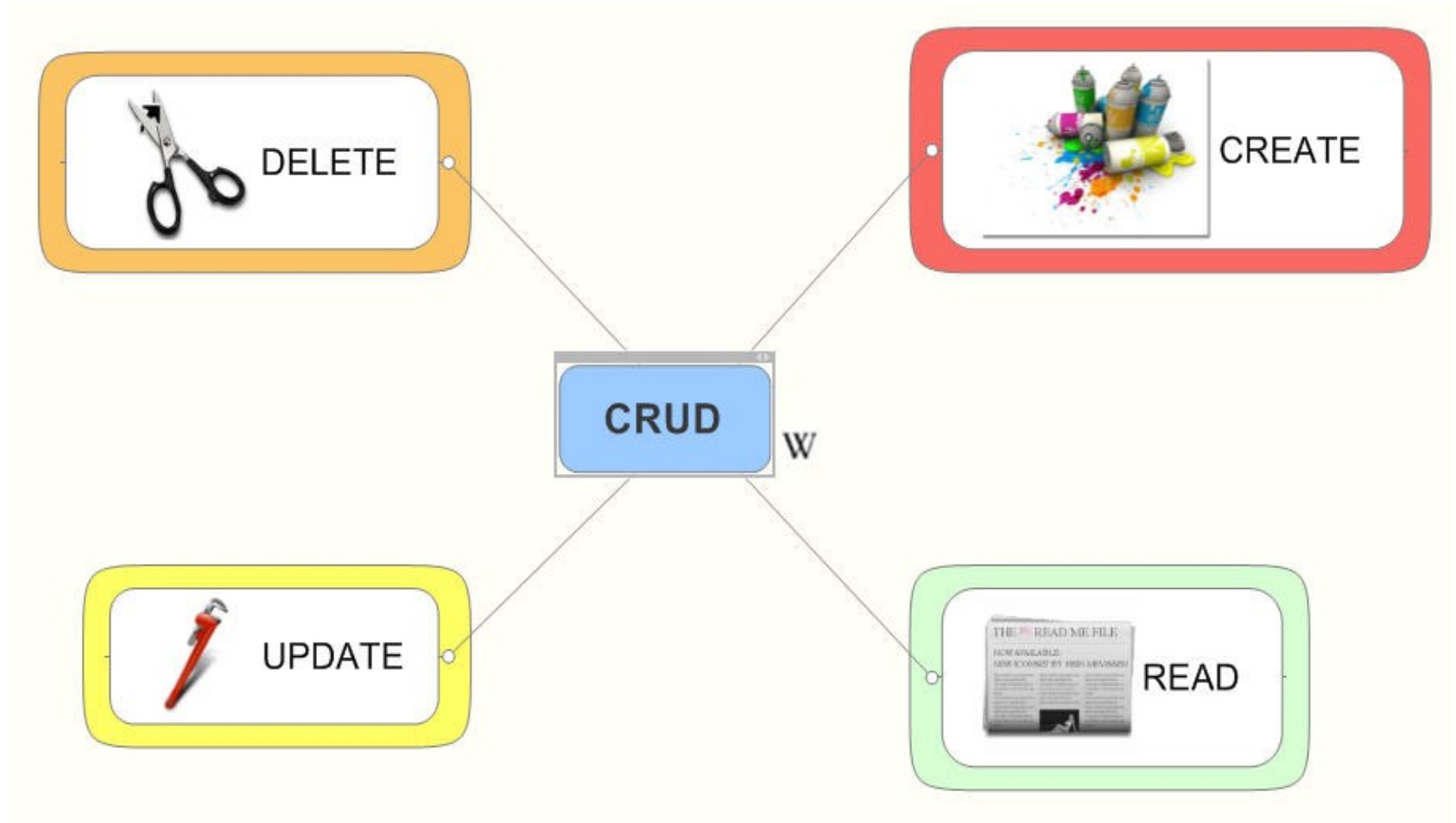
- Alguns aplicativos fornecem menus pop-up sensíveis ao contexto.
- No Swing, esses menus são criados com a classe **JPopupMenu**.
- Esses menus fornecem opções que são específicas do componente pelo qual o evento de gatilho pop-up foi gerado.
- Normalmente, o evento de acionamento pop-up ocorre quando o usuário pressiona e libera o botão direito do mouse.

GUI – Swing – Telas Complexas

PopUpMenu:

- Alguns aplicativos fornecem menus pop-up sensíveis ao contexto.
- No Swing, esses menus são criados com a classe **JPopupMenu**.
- Esses menus fornecem opções que são específicas do componente pelo qual o evento de gatilho pop-up foi gerado.
- Normalmente, o evento de acionamento pop-up ocorre quando o usuário pressiona e libera o botão direito do mouse.

CRUD



CRUD

CRUD (acrónimo de **C**reate, **R**ead, **U**psert e **D**estroy na língua Inglesa) para as **quatro operações básicas** utilizadas em bases de dados relacionais (RDBMS) ou em interface para utilizadores para criação, consulta, atualização e destruição de dados.

A abreviação CRUD mapeada para o padrão **ISO/SQL**:

Create	INSERT
Read (Retrieve)	SELECT
Update	UPDATE
Delete (Destroy)	DELETE

Exemplos de Interfaces CRUD em Java

The screenshot shows a Java application window titled "Cadastro de Livros". It contains two main sections: "Adicionar Livros" (Add Books) and "Lista de Livros" (List of Books).

Adicionar Livros

Form fields:

- Editora: Novatec
- Livro: Google Android
- ISBN: 978857522447

Buttons: Add (pencil icon), Delete (trash icon), Save (floppy disk icon), Edit (pencil icon), and Cancel (X icon).

Lista de Livros

Código	Editora	Título	ISBN
1	Alta Books	Certificação Sun Para Progra...	978857608...
2	Novatec	HTML 5	978857522...
3	Alta Books	Use a Cabeça! Redes de Co...	978857608...
4	Alta Books	Use a Cabeça! Programação	978857608...
5	Novatec	Google Android	978857522...
6	9788575222386	Programação Java para a Web	978857522...
7	Alta Books	Domain-Driven Design	978857608...
8	Alta Books	Dominando Php e Mysql	978857608...
9	Novatec	Padrões Javascript	978857522...
10	Novatec	Google Android Para Tablets	978857522...

Exemplos de Interfaces CRUD em Java

Cadastro de Produtos

Clientes Produtos Vendas Fornecedores Relatórios Ordem de serviço Pesquisar Sobre?

Id	Nome	Fornecedor	Preço comp...	Preço	quantidade	Categoria	Marca	Status	Data
1	HD320GB	HARDWARE	200.00	100.00	1	HD	HP	Novo	02/11/2014
2	MONITOR 1...	MARIA DA ...	300.00	200.00	2	MONITOR	SAMSUG	Novo	02/11/2014
3	BLUE RAY	HARDWARE	100.00	25.00	10	HD	HP	Novo	02/11/2014
4	CD	HARDWARE	10.00	5.00	6	ROTEADOR	HP	Novo	02/11/2014

Qtde:

Nome:

Preço de Venda:





Preço de compra:

Categoria:

Fornecedor:

Status:

Modelo/Marca:

 Salvar  Limpar  Alterar  Sair

Exemplos de Interfaces CRUD em Java

Cadastro de Clientes

ID: 1 Nome: Daniel Fernandes Martins

Tel. Residencial: (14)3333-3333 Tel. Comercial: (14)4444-4444 Tel. Celular: (14)5555-5555

E-mail: daniel.tritone@gmail.com

Inserir **Remover** **Sair**

Pergunta Deseja remover?

Yes **No**

Clientes cadastrados

ID	Nome	E-mail
1	Daniel Fernandes Martins	daniel.tritone@gmail.com

Sugestão de Interface CRUD em Java

Com uso de **JTabbedPane**, com duas ABAS, uma aba para exibir todos os itens já cadastrados, e outra aba para edição individual de cada registro selecionado pelo usuário na aba de Consulta.

Botões para Incluir, Editar, Deletar.

Lista de Livros

Código	Editora	Título	ISBN
1	Alta Books	Certificação Sun Para Progra...	978857608...
2	Novatec	HTML 5	978857522...
3	Alta Books	Use a Cabeça! Redes de Co...	978857608...
4	Alta Books	Use a Cabeça! Programação	978857608...
5	Novatec	Google Android	978857522...
6	9788575222386	Programação Java para a Web	978857522...
7	Alta Books	Domain-Driven Design	978857608...
8	Alta Books	Dominando Php e Mysql	978857608...
9	Novatec	Padrões Javascript	978857522...
10	Novatec	Google Android Para Tablets	978857522...

Botões para Salvar ou Abandonar a alteração/Inclusão.

Referências

Programação de computadores em
Java

Rui Rossi dos Santos

Java 8: Programação de Computadores
- Guia Prático de Introdução, Orientação
e Desenvolvimento - José Augusto N.
G. Manzano

Slides de Gabriel de Jesus

Sugestão de Material

<http://www.dialetodigital.com/blog/conteudos-programacao/>