

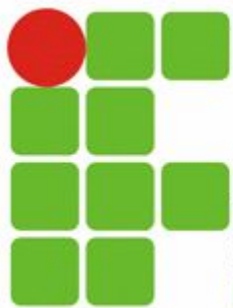
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
FLUMINENSE

PROGRAMAÇÃO ORIENTADA A OBJETOS

Interface Gráfica

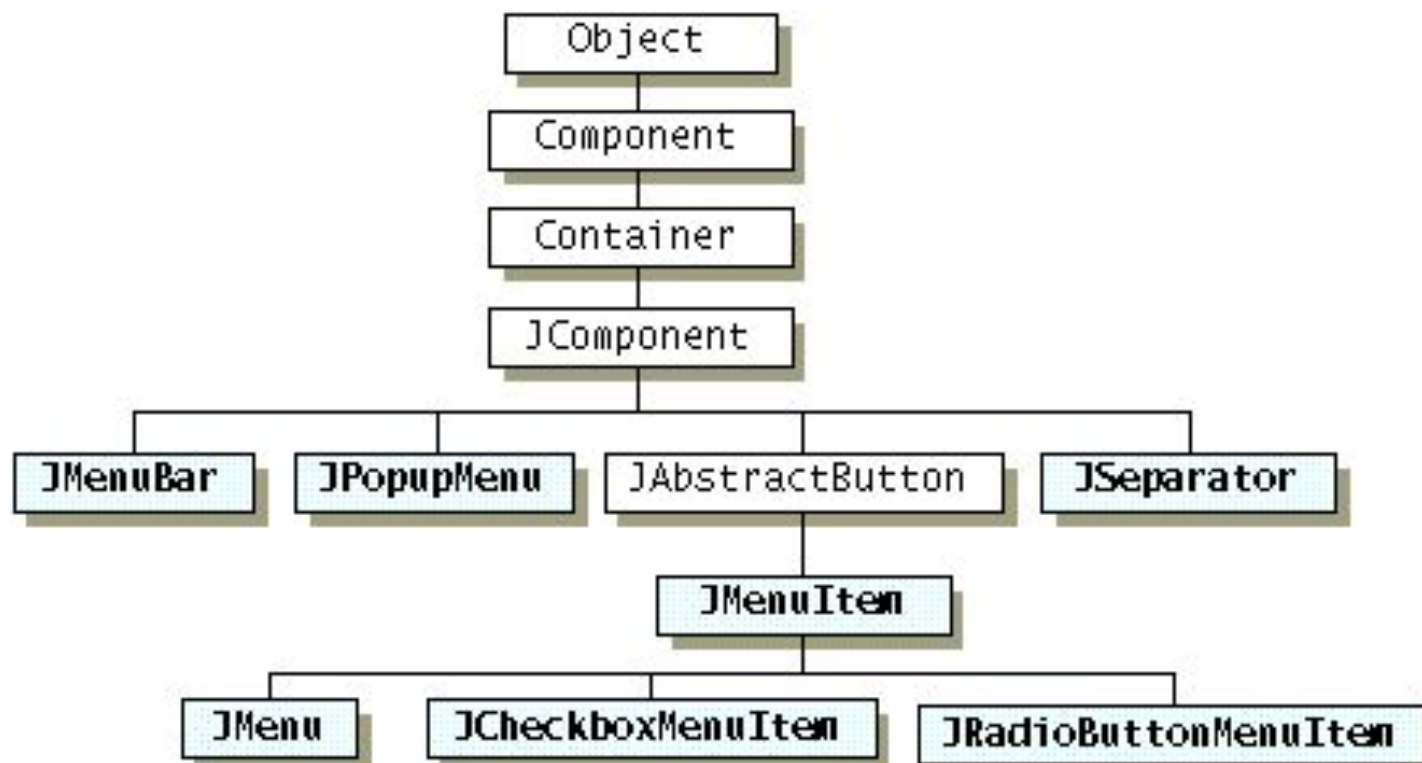
Profª Roberta B. Tôrres

Adaptação do Material de Profª Giselle Teixeira de Almeida



□ Componentes Avançados da GUI

- JMenuBar
- JMenu
- JMenuItem



Componentes GUI Avançados

Criação de Menus

Os menus podem ser anexados diretamente a JFrame através do método setJMenuBar.

- **JMenuBar** é a classe responsável por gerenciar uma barra de menu. Uma barra de menu é um container para menus.
- **JMenu** é a classe que gera as opções de menu e podem conter uma coleção de itens de menu.
- **JMenuItem** é a classe que gera os itens de menu.

Componentes GUI Avançados

JMenuBar, JMenu e JMenuItem

Exemplos:

```
JMenuBar barramenu = new JMenuBar();
```

```
JMenu menu1 = new JMenu("Cadastrar");
```

```
JMenu menu2 = new JMenu("Sair");
```

```
JMenuItem item1 = new JMenuItem("Inserir");
```

```
JMenuItem item2 = new JMenuItem("Listar");
```

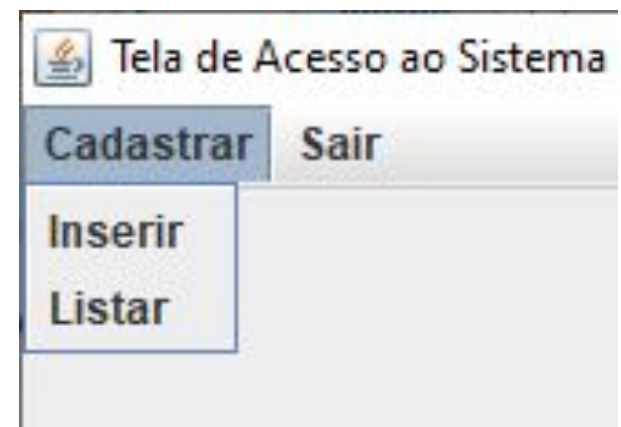
```
menu1.add( item1 );
```

```
menu1.add( item2 );
```

```
barramenu.add(menu1);
```

```
barramenu.add(menu2);
```

```
frame.setJMenuBar( barramenu );
```

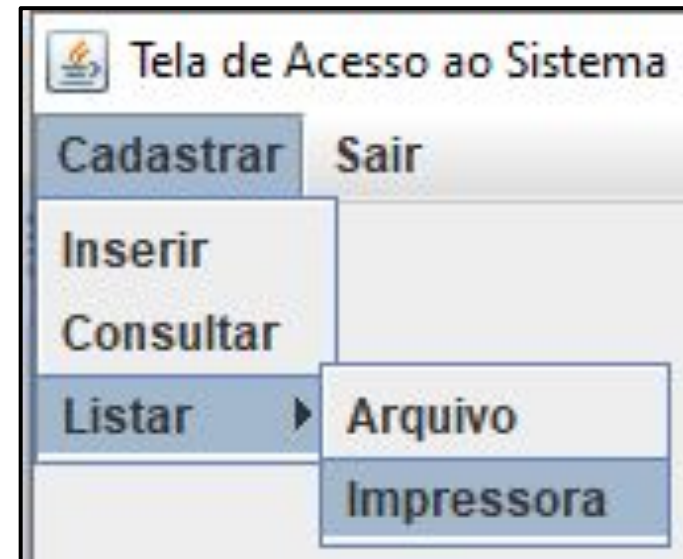


Componentes GUI Avançados

Criação de Submenu

```
JMenuBar barramenu = new JMenuBar();  
JMenu menu1 = new JMenu("Cadastrar");  
JMenu menu2 = new JMenu("Sair");  
JMenuItem item1 = new JMenuItem("Inserir");  
JMenuItem item2 = new JMenuItem("Consultar");  
  
menu1.add(item1);  
menu1.add(item2);  
  
JMenu submenu1 = new JMenu("Listar");  
JMenuItem item3 = new JMenuItem("Arquivo");  
JMenuItem item4 = new JMenuItem("Impressora");  
submenu1.add(item3);  
submenu1.add(item4);
```

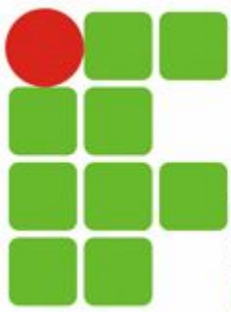
```
menu1.add(submenu1);  
barramenu.add(menu1);  
barramenu.add(menu2);  
frame.setJMenuBar(barramenu);
```



Componentes GUI Avançados

□ Criação de Menus e Submenus

Execute o código-fonte ***JanelaMenu.java*** para visualizar a criação de menus.



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
FLUMINENSE

GUI COMPONENTES AVANÇADOS

TRATAMENTO DE EVENTOS

Componentes GUI Avançados

Tratamento de Eventos

- Quando o usuário interage com um componente GUI, a interação - conhecida como **EVENTO** - guia o programa para realizar uma tarefa.
- Algumas interações comuns feitas pelos usuários com aplicativos incluem: clicar em botão, digitar um campo texto, selecionar um item de menu, fechar uma janela, mover o mouse.
- O código que realiza uma tarefa em resposta a um **EVENTO** é chamado de **HANDLER (manipulador) DE EVENTO**.
- O processo total de responder a **EVENTOS** é conhecido como **TRATAMENTO DE EVENTOS**.

Componentes GUI Avançados

Tratamento de Eventos

- Eventos são ações/tarefas realizadas quando o usuário interage com algum componente GUI.
- Os eventos estão associados aos componentes e refletem ações aplicadas sobre eles.
- Todo evento tem um objeto que é sua fonte. As Classes que tratam os Eventos estão no pacote **java.awt.event**.
- Os eventos são divididos em categorias:
 - **ActionEvent** (fonte: botão, lista, item de menu, caixa de texto).
 - **MouseEvent** (fonte: componentes afetados pelo mouse).
 - **ItemEvent** (fonte: checkBox, list).
 - **TextEvent** (fonte: componentes de texto).

Componentes GUI Avançados

Tratamento de Eventos

Cada evento tem uma interface (ouvinte) correspondente que possui métodos padrões para tratá-lo. Esta interface é chamada de **Listeners**.

Eventos x Listeners:

- ActionEvent x ActionListener
- MouseEvent x MouseListener
- ItemEvent x ItemListener
- TextEvent x TextListener

Execute o código-fonte ***JanelaEventos.java*** para visualizar o tratamento de eventos e chamada de Janelas sobrepostas.

Componentes GUI Avançados

Tratamento de Eventos

- Analise o código abaixo sobre o componente **JTextField**:

```
caixatexto.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        JOptionPane.showMessageDialog(null, "USUÁRIO: \n" +
            caixatexto.getText() );
    }
});
```

- Para implementar um Evento sobre o componente **JTextField** é necessário configurar a interface **ActionListener** e o método **addActionListener()** do componente.
- **actionPerformed** é o método que trata o evento.
- **ActionEvent** é executado quando se pressiona **ENTER** após preencher a caixa de texto.

Componentes GUI Avançados

Tratamento de Eventos

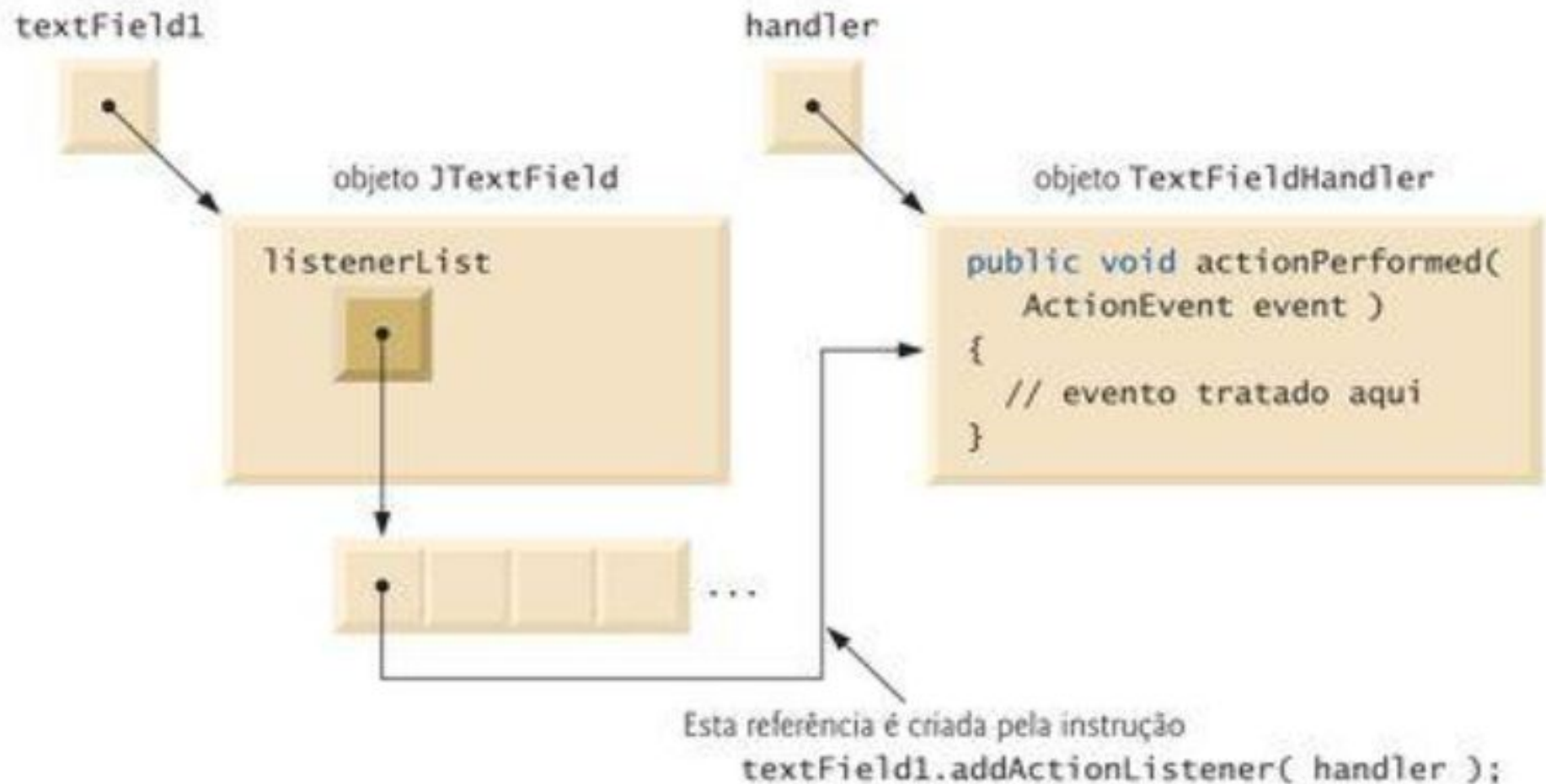


Figura 14.13 | Registro de evento para o `JTextField` `textField1`.

Componentes GUI Avançados

Tratamento de Eventos - Mouse

Métodos de interface `MouseListener`

```
public void mousePressed( MouseEvent event )
```

Chamado quando um botão do mouse é pressionado enquanto o cursor do mouse estiver sobre um componente.

```
public void mouseClicked( MouseEvent event )
```

Chamado 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`.

```
public void mouseReleased( MouseEvent event )
```

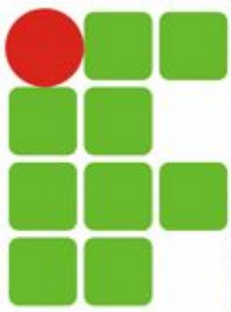
Chamado quando um botão do mouse é liberado depois de ser pressionado. Esse evento sempre é precedido por uma chamada para `mousePressed` e uma ou mais chamadas para `mouseDragged`.

```
public void mouseEntered( MouseEvent event )
```

Chamado quando o cursor do mouse entra nos limites de um componente.

```
public void mouseExited( MouseEvent event )
```

Chamado quando o cursor do mouse deixa os limites de um componente.



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
FLUMINENSE

GUI COMPONENTES AVANÇADOS

FONTES E CORES

Componentes GUI Avançados

Fontes e Cores

- Qualquer componente pode mudar a sua fonte e cor.

□ Fontes são instâncias da classe `java.awt.Font`.

```
Font f = new Font("SansSerif", Font.BOLD, 24);  
componente.setFont( f );
```

□ Cores são instâncias da classe `java.awt.Color`.

- **Cor de fundo:** `componente.setBackground(new Color(255,0,0));`
- **Cor da fonte:** `componente.setForeground(Color.yellon);`

Execute o código-fonte ***JanelaParte03.java*** para visualizar a mudança de cores nos componentes.

Gerenciadores de Layout

□ Gerenciadores de Layout

- FlowLayout
- BorderLayout
- GridLayout

□ Combinando Layouts

□ Exemplos e Exercícios

Gerenciadores de Layout

Gerenciadores de Layout auxiliam a organizar os componentes GUI em um contêiner para propósitos de apresentação, sendo possível determinar a posição e o tamanho exato de cada componente na tela.

Os principais Gerenciadores de Layout são:

- FlowLayout
- BorderLayout
- GridLayout

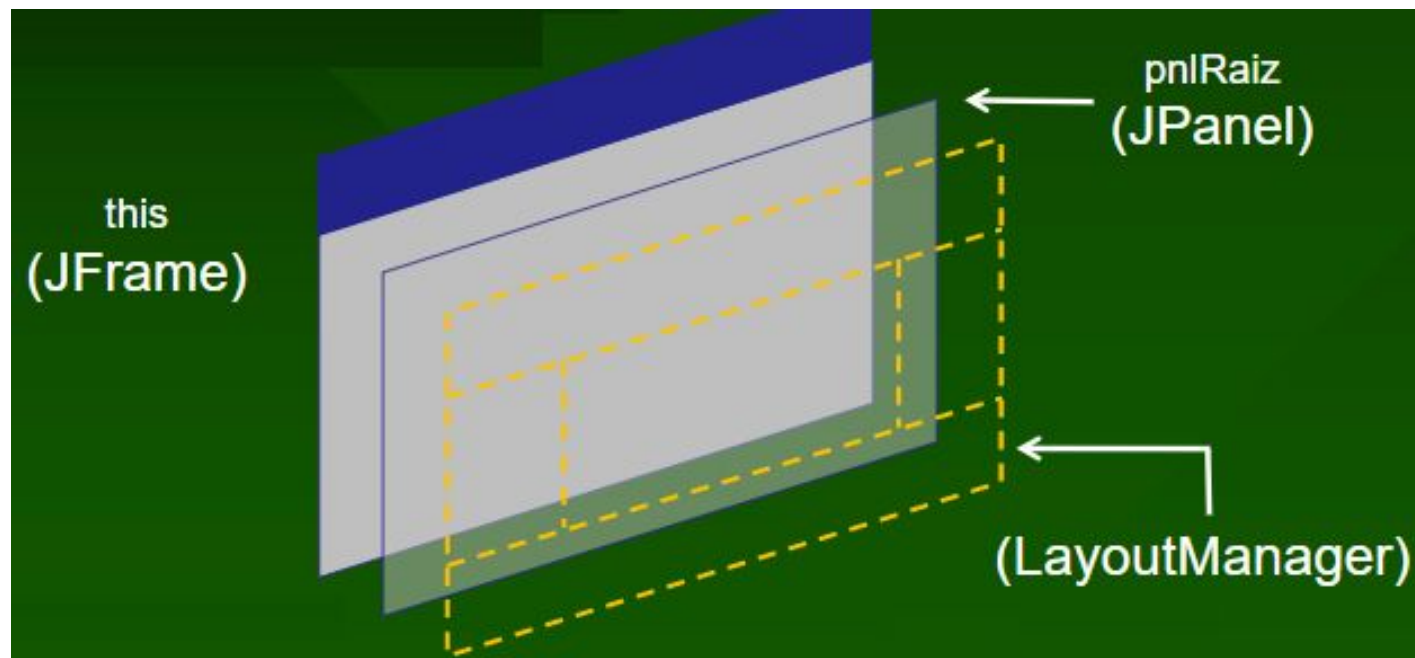
Gerenciadores de Layout

Há três maneiras de organizar os componentes em uma GUI:

- **Posicionamento Absoluto**: este modo fornece o maior nível de controle sobre a aparência da GUI. Para posicionar o componente na tela deve-se informar as coordenadas x e y da tela, bem como o tamanho e largura do componente.
- **Gerenciadores de Layout**: posicionar os componentes com gerenciadores é uma forma mais simples e rápida do que usar Posicionamento Absoluto. No entanto, perde-se algum controle sobre o tamanho e posição precisa dos componentes.
- **Programação visual em uma IDE** (ambiente de desenvolvimento integrado): as IDEs fornecem ferramentas que facilitam a criação de GUIs. Em geral, toda IDE fornece uma ferramenta de design GUI que permite arrastar e soltar os componentes.

Gerenciadores de Layout

Um **Gerenciador de Layout** é um objeto vinculado ao painel da janela (**JPanel**) e implementa a interface **LayoutManager** (do pacote `java.awt`). Existe o método **setLayout** da classe **Container (JFrame)** que aceita um objeto que implementa a interface **LayoutManager** como um argumento.



Gerenciadores de Layout

FlowLayout (Layout de Fluxo)

- É o gerenciador de layout mais básico.
- Organiza os componentes seqüencialmente da direita para esquerda, na ordem em que foram adicionados (layout seqüencial).
- Quando a borda do container/panel é alcançada, os componentes continuam na próxima linha.
- Permite que os componentes sejam alinhados à direita, ao centro e à esquerda.
- É possível especificar a ordem dos componentes utilizando o método *add*.

Gerenciadores de Layout

FlowLayout - Construtores

→ A classe FlowLayout possui alguns construtores pré-definidos:

■ `FlowLayout()`

- ◆ Centraliza os componentes deixando uma distância entre eles (**gap**) de 5 pixels.

■ `FlowLayout(int align)`

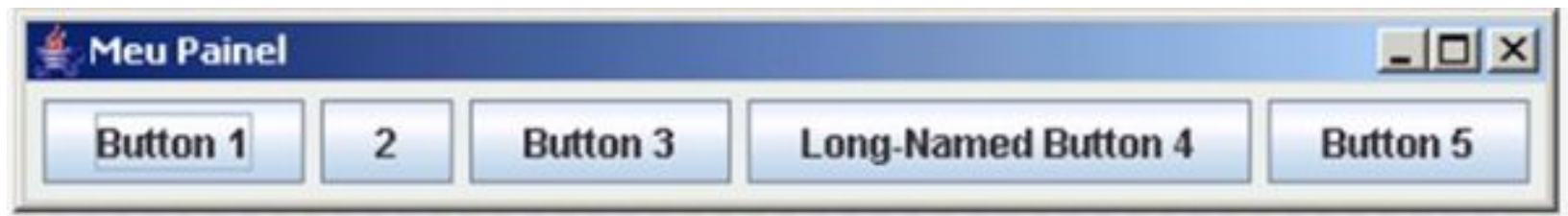
- ◆ Constrói um layout com alinhamento estabelecido (`FlowLayout.CENTER`, `FlowLayout.LEFT`, `FlowLayout.RIGHT`)

■ `FlowLayout(int align, int hgap, int vgap)`

- ◆ Define o alinhamento e a distância horizontal e vertical, respectivamente

Gerenciadores de Layout

FlowLayout - Exemplo



```
Container contentPane = janela.getContentPane();

contentPane.setLayout(new FlowLayout());

contentPane.add(new JButton("Button 1"));
contentPane.add(new JButton("2"));
contentPane.add(new JButton("Button 3"));
contentPane.add(new JButton("Long-Named Button 4"));
contentPane.add(new JButton("Button 5"));
```

Gerenciadores de Layout

FlowLayout

Código

```
Container c = this.getContentPane();  
FlowLayout layout = new FlowLayout();  
c.setLayout(layout);  
layout.setAlignment(FlowLayout.LEFT);
```

Código (Nenhum Layout)

```
c.setLayout(null);
```



Execute o código-fonte ***JanelaFlowlayout.java*** para visualizar a disposição dos componentes com este gerenciador.

Gerenciadores de Layout

BorderLayout (Layout de Borda)

- Gerenciador padrão para o painel de conteúdo (container) de todo JFrame e JApplets.
- Organiza os componentes em cinco áreas: 4 posições “cardeais” (norte, sul, leste, oeste) e ao centro.
- Norte e Sul têm prioridade sobre Leste e Oeste que, por sua vez, têm prioridade sobre Centro.
- Os componentes tem seu tamanho alterado, de modo a preencher todo o espaço disponível.
- Constantes: BorderLayout. NORTH / SOUTH / WEST, EAST, CENTER.

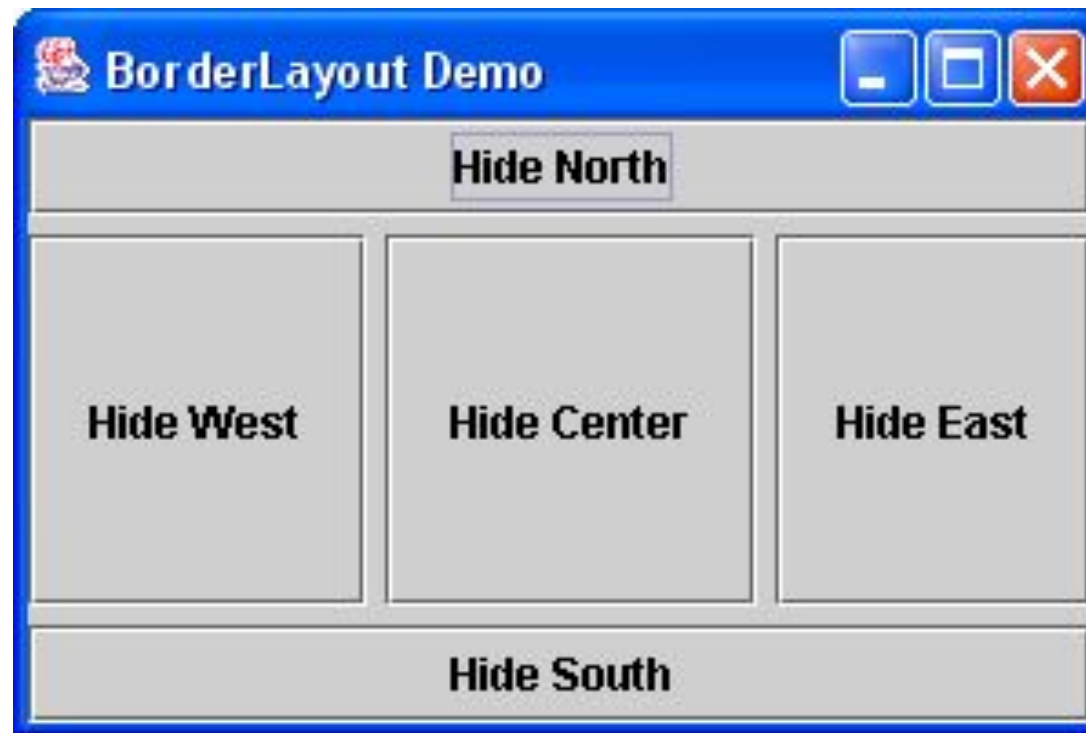
Gerenciadores de Layout

BorderLayout

```
Container c = this.getContentPane();  
BorderLayout layout = new BorderLayout(5,5);  
c.setLayout(layout);  
JButton botao1 = new JButton("TOPO");  
JButton botao2 = new JButton("RODAPÉ");  
JButton botao3 = new JButton("ESQUERDA");  
JButton botao4 = new JButton("CENTRO");  
JButton botao5 = new JButton("DIREITA");  
c.add (botao1,BorderLayout.NORTH);  
c.add (botao2,BorderLayout.SOUTH);  
c.add (botao3,BorderLayout.WEST);  
c.add (botao4,BorderLayout.CENTER);  
c.add (botao5,BorderLayout.EAST);
```

Gerenciadores de Layout

BorderLayout



Execute o código-fonte ***JanelaBorderlayout.java*** para visualizar a disposição dos componentes com este gerenciador.

Gerenciadores de Layout

GridLayout (layout de grade)

- Divide o container em uma grade, organizando os componentes em linhas e colunas, como elementos de uma tabela.
- Os componentes são adicionados da esquerda para direita.
- Cada componente em um **GridLayout** tem o mesmo tamanho.
- O tamanho individual dos componentes não é respeitado.
- Cada célula aceita um componente que a preenche totalmente.

Gerenciadores de Layout

GridLayout

```
Container c = this.getContentPane();  
// Grid de 3 linhas x 2 colunas, com espaçamento horizontal e vertical de 5px  
  
GridLayout layout = new GridLayout(3,2,5,5);  
c.setLayout(layout);  
  
JButton botao1 = new JButton("UM");  
JButton botao2 = new JButton("DOIS");  
JButton botao3 = new JButton("TRÊS");  
JButton botao4 = new JButton("QUATRO");  
JButton botao5 = new JButton("CINCO");  
JButton botao6 = new JButton("SEIS");  
  
c.add (botao1);      c.add (botao4);  
c.add (botao2);      c.add (botao5);  
c.add (botao3);      c.add (botao6);
```

Gerenciadores de Layout

GridLayout



Execute o código-fonte ***JanelaGridlayout.java*** para visualizar a disposição dos componentes com este gerenciador.

Gerenciadores de Layout

Outros Gerenciadores

Existem outros Gerenciadores de Layout que permitem layouts sofisticados com amplo controle e maior flexibilidade, são eles:

- BoxLayout
- CardLayout
- GridBagLayout

Saiba mais sobre estes Gerenciadores no link abaixo:

<https://pt.slideshare.net/DanielArndtAlves/gerenciadores-de-layout>

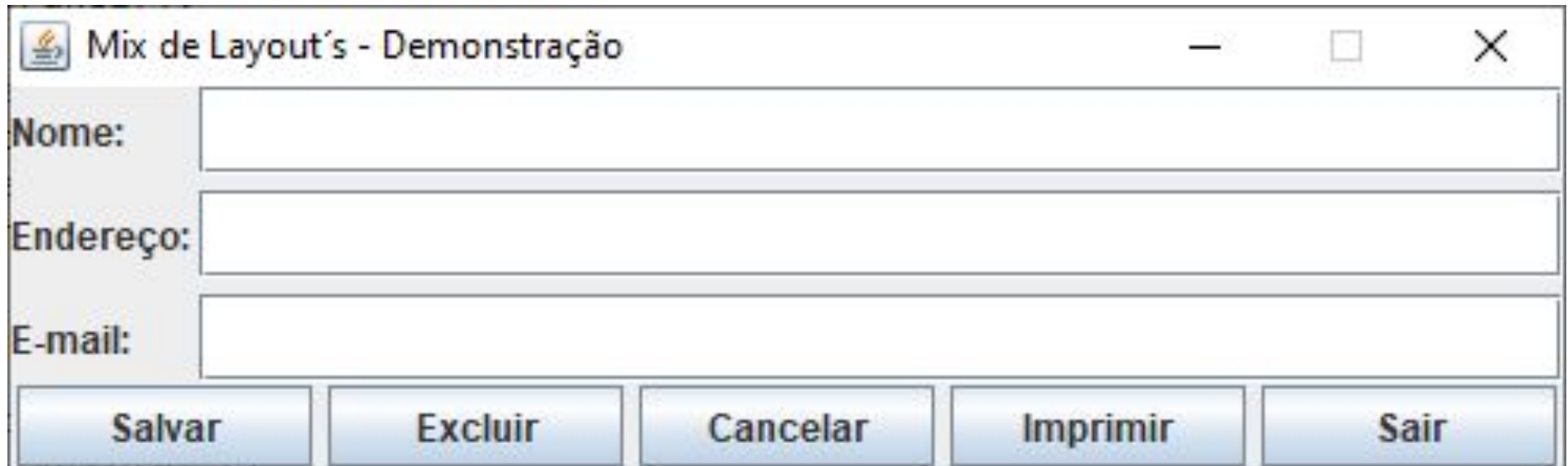
Gerenciadores de Layout

Combinando Layouts

- Layout's podem ser combinados para a construção de uma GUI mais elaborada.
- **Exemplo:** o frame pode ter um layout diferente do layout aplicado a um JPanel.
 - `Container c = this.getContentPane();`
 - `c.setLayout(new BorderLayout());`
 - `JPanel p = new JPanel();`
 - `p.setLayout(new GridLayout(3,1));`

Gerenciadores de Layout

Combinando Layouts



The screenshot shows a Java Swing window titled "Mix de Layout's - Demonstração". The window contains three text input fields labeled "Nome:", "Endereço:", and "E-mail:". Below these fields is a row of five buttons: "Salvar", "Excluir", "Cancelar", "Imprimir", and "Sair". The window has standard Windows-style window controls (minimize, maximize, close) in the title bar.

Execute o código-fonte ***JanelaMixlayout.java*** para visualizar a disposição dos componentes com a combinação de layout.

Ferramenta Automatizada

Plugin WindowBuilder

O **WindowBuilder** é um plugin disponível para Eclipse, que tem o objetivo de simplificar o desenvolvimento das interfaces gráficas para aplicações desktop.

O artigo “**Construindo telas com Java WindowBuilder**” (link abaixo) ilustra como implementar uma Agenda Simples, usando este plugin.

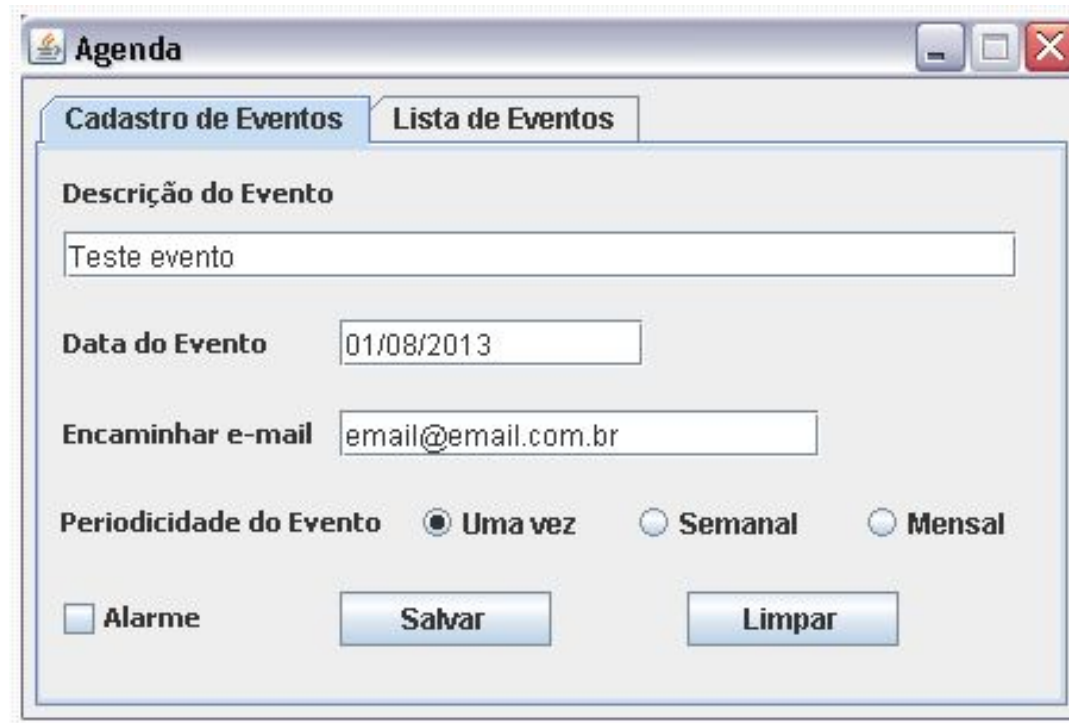
Maiores informações:

<https://www.devmedia.com.br/construindo-telas-com-java-windowbuilder/29312>

Ferramenta Automatizada

Plugin WindowBuilder

Tela da Aplicação Agenda do artigo “**Construindo telas com Java WindowBuilder**”.



The screenshot shows a Java Swing window titled "Agenda". It has two tabs: "Cadastro de Eventos" (active) and "Lista de Eventos". The "Cadastro de Eventos" tab contains the following fields and controls:

- Descrição do Evento:** A text field containing "Teste evento".
- Data do Evento:** A date field containing "01/08/2013".
- Encaminhar e-mail:** A text field containing "email@email.com.br".
- Periodicidade do Evento:** Three radio buttons: "Uma vez" (selected), "Semanal", and "Mensal".
- Alarme:** A checkbox that is currently unchecked.
- Buttons:** "Salvar" and "Limpar" buttons.

Maiores informações:

<https://www.devmedia.com.br/construindo-telas-com-java-windowbuilder/29312>

Interface Gráfica do Usuário - GUI

Plugin WindowBuilder

Canal Descompila (YouTube) - acesse a Playlist de Vídeos “**Interface Gráfica JavaSE com Eclipse**”.

Estes vídeos ilustram como criar uma Tela em Java usando o plugin WindowBuilder.

https://www.youtube.com/playlist?list=PLWd_VnthxxLdhkzJA1VoziSqP5la7B0zE

Interface Gráfica do Usuário - GUI

EXERCÍCIO DE FIXAÇÃO

- Aplique os conhecimentos sobre GUI apresentados até o momento e tente montar uma interface gráfica conforme a ilustração a seguir.
- Serão necessários os seguintes componentes: JFrame, JPanel, JLabel, JTextField, JButton, Gerenciador de Layout, etc...
- Use a sua criatividade.

Interface Gráfica do Usuário - GUI

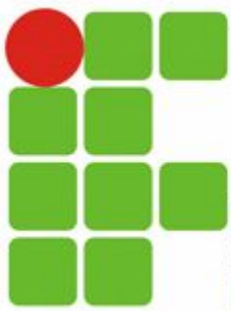
EXERCÍCIO DE FIXAÇÃO



The image shows a screenshot of a graphical user interface (GUI) window titled "Cadastro de Clientes". The window has a blue title bar with standard Windows window controls (minimize, maximize, close) on the right. The main area is light gray and contains several input fields and buttons. The fields are labeled on the left and have corresponding input boxes on the right. The labels are in a dark blue font. The input boxes are white with a thin gray border. The "Tipo" field is a dropdown menu with "Pessoa Física" selected. At the bottom, there are four buttons: "Cancelar", "Salvar", "Enviar", and "Sair", each in a separate gray box with black text.

Label	Input Type	Value
Codigo	Text	
Nome	Text	
Tipo	Dropdown	Pessoa Física
Endereco	Text	
Bairro	Text	
Cidade	Text	
Estado	Text	
CEP	Text	

Buttons: Cancelar, Salvar, Enviar, Sair



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
FLUMINENSE

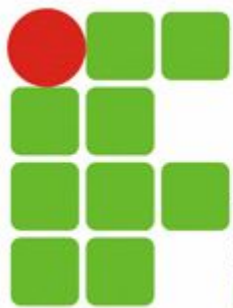
EXERCÍCIO DE FIXAÇÃO

- Tente fazer o exercício proposto no link abaixo para fixar o conteúdo de GUI
 - https://www.inf.pucrs.br/~michael/aulas/algo_progII/apresentacoes/Exerc%C3%ADcios%20Interface%20Gr%C3%A1fica.htm

- Aplicação
Calculadora
de Poupança

The screenshot shows a Java Swing window titled "Poupex" with a standard Mac OS X-style title bar (blue with red, yellow, and green buttons). The window contains a savings calculator interface with the following elements:

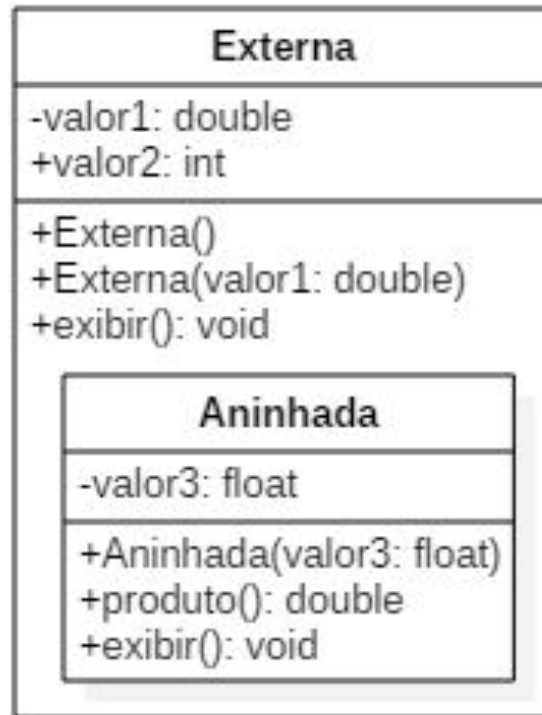
- Four text input fields arranged vertically on the right side of the window.
- Labels for each field on the left:
 - "Juros ao mês %:" (Interest per month %)
 - "Num. de anos:" (Number of years)
 - "Depósito mensal R\$:" (Monthly deposit R\$)
 - "Total poupado R\$:" (Total saved R\$)
- An "OK" button located at the bottom right of the window.



Classes Internas

- Leitura Obrigatória para compreender melhor o conceito e como implementar **Classes Internas**

<http://tecnopode.blogspot.com/2018/04/poo-p-25.html>



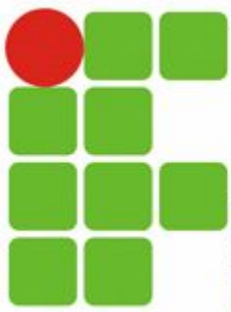
Gerenciadores de Layout

→ MATERIAL COMPLEMENTAR – EXEMPLOS

Conhecendo Gerenciadores de Layout GUI do Java

Veja neste artigo como os gerenciadores de layout organizam os componentes GUI em um contêiner para propósitos de apresentação. Saiba as capacidades básicas de layout para determinar a posição e o tamanho exato de cada componente GUI.

<https://www.devmedia.com.br/conhecendo-gerenciadores-de-layout-gui-do-java/25869>



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
FLUMINENSE

Material Complementar

- <https://slideplayer.com.br/slide/358129/>
- <https://www.devmedia.com.br/trabalhando-com-eventos-na-gui-do-java/25898>
- Gerenciador de Layout
 - <http://www.deinf.ufma.br/~mario/grad/poo/aulas/Layout.pdf>
 - <https://www.incc.br/~rogerio/poo/05a%20-%20Gerenciamento%20de%20Layout.pdf>
 - <https://www.caelum.com.br/apostila-java-testes-xml-design-patterns/mais-swing-layout-managers-mais-componentes-e-detalhes/>
- Exemplo de Aplicação Agenda de Celular
 - <https://slideplayer.com.br/slide/14154557/>