

# Programação Gráfica em Java – Alguns Componentes javax.swing

## Objetivo:

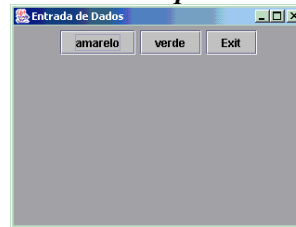
Ao final da aula o aluno deverá ser capaz de empregar alguns componentes visuais do pacote javax.swing para produzir interfaces gráficas.

Além dos componentes visuais, será abordado, de forma superficial, o listener do teclado, tratamento de erros e entrada e saída de arquivos texto.

## Componente

*JButton:*

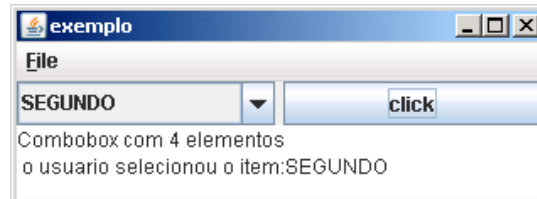
## Exemplo



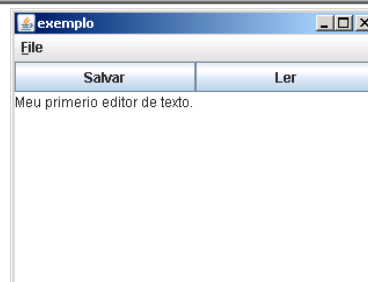
*Gerenciador de  
Layout:*



*JComboBox e JTextArea:*



*JTextArea;  
JScrollPane;  
Arquivos de Entrada e  
Saída e Tratamento de  
Erros:*



*JPasswordField e  
Listener do Teclado:*



# Sumário

1. Componentes de Interface Gráfica.....	24
Botão .....	24
Exemplo de código: .....	24
Sugestão de código para o programa de teste ou execução. ....	26
A execução: .....	26
Como definir o posicionamento dos componentes? .....	27
2. Gerente de Layout.....	27
a) FlowLayout .....	28
Exemplo de FlowLayout:.....	28
b) BorderLayout: .....	28
Como inserir componentes em uma específica região?.....	29
Exemplo de BorderLayout: .....	29
c) GridLayout: .....	29
Exemplo de definição de um GridLayout: .....	29
Exemplo de Execução do GridLayout .....	30
GridLayout com espaços. ....	30
3. Outros Componentes Gráficos.....	30
3.1 JComboBox / JTextArea .....	32
3.2 JTextArea / JScrollPane / Arquivos de Entrada e Saída / Tratamento de Erros .....	33
3.3 JPasswordField / Listener do Teclado .....	35
Desafio: .....	36
Desafio 2: .....	37

## 1. Componentes de Interface Gráfica

<b>Botão</b>	<p>Um botão, ou JButton, é um componente gráfico que permite que o usuário solicite alguma ação.</p> <p>Um elemento JButton é inicializado de maneira similar a qualquer outro componente Java, através da palavra new. As linhas 13 e 14 do código abaixo criam um botão cujo rótulo é a palavra “amarelo”. As linhas 15, 16, 17 e 18 criam outros dois botões.</p> <p>Após a criação das instâncias dos botões, é recomendável, adicionar métodos <i>listeners</i> conforme ilustrado nas linhas 38 a 44, por exemplo.</p> <p>A instância de JButton pode ser adicionada no painel através da instrução add(Component) do componente JPanel. A linha 59 ilustra esse aspecto.</p>
<b>Exemplo de código:</b>	<pre>////////// **** arquivo meuFormulario.java /*01*/ package interfaceGrafica.formulario;  /*02*/ import javax.swing.*; /*03*/ import java.awt.event.*; /*04*/ import java.awt.*;  /*05*/ class Eventos extends WindowAdapter /*06*/ { /*07*/     public void windowClosing(WindowEvent e) /*08*/         {System.exit(0); } /*09*/ }  /*10*/ public class meuFormulario extends JFrame /*11*/ { /*12*/     private JPanel conteudo = new JPanel();  /*13*/     private JButton botao1 = new /*14*/         JButton("amarelo"); /*15*/     private JButton botao2 = new /*16*/         JButton("verde"); /*17*/     private JButton botaoSair = new /*18*/         JButton("Exit");  /*19*/     private void centraliza() /*20*/     { /*21*/         int x = /*22*/         ((Toolkit.getDefaultToolkit()) /*23*/             .getScreenSize()).width; /*24*/         int y = /*25*/         ((Toolkit.getDefaultToolkit()) /*26*/             .getScreenSize()).height; /*27*/         setLocation((x - getSize().width)/2 /*28*/             , (y-getSize().height)/2); /*29*/     }  /*30*/     public meuFormulario(String titulo) /*31*/     { /*32*/         addWindowListener( new Eventos()); /*33*/         conteudo.setBackground(new /*34*/             Color(159,154,151)); /*35*/         setTitle(titulo);</pre>

```
/*36*/ setSize(320,240);
/*37*/ centraliza();

/*38*/ botao1.addActionListener
/*39*/ (
/*40*/ new AbstractAction()
/*41*/ {public void actionPerformed(ActionEvent evt)
/*42*/     {conteudo.setBackground(Color.YELLOW);}
/*43*/     }
/*44*/ );

/*45*/ botao2.addActionListener
/*46*/ (
/*47*/ new AbstractAction()
/*48*/ {public void actionPerformed(ActionEvent evt)
/*49*/     {conteudo.setBackground(Color.GREEN);}
/*50*/     }
/*51*/ );

/*52*/ botaoSair.addActionListener
/*53*/ (
/*54*/ new AbstractAction()
/*55*/ {public void actionPerformed(ActionEvent evt)
/*56*/     {System.exit(0);}
/*57*/     }
/*58*/ );

/*59*/ conteudo.add(botao1);
/*60*/ conteudo.add(botao2);
/*61*/ conteudo.add(botaoSair);

/*62*/ Container fundo = getContentPane();
/*63*/ fundo.add(conteudo);
/*64*/ }
/*65*/ }
```

**Sugestão de código para o programa de teste ou execução.**

```
////***** arquivo testeMeuFormulario.java
package interfaceGrafica.formulario;

import javax.swing.*;

public class testeMeuFormulario
{
    public static void main(String[] a)
    {
        (new meuFormulario
            ("Entrada de Dados")).show();
    }
}
```

**A execução:**

A Figura 1 ilustra o resultado da execução do programa anterior. Pode-se observar que não foi definido no código do programa o posicionamento dos botões. Poderíamos dizer que os botões são colocados na ordem em que a função add (Component) foi chamada e que seu posicionamento na tela é aleatório.



Figura 1-Inserção aleatória de botões.

Esse posicionamento aleatório pode ser melhor observado se inserirmos uma quantidade elevada de componentes. Por exemplo, insira entre as linhas 61 e 62 do código anterior, a seguinte instrução:

```
/*61_1*/ for (int i = 4; i < 34; i++)
/*61_2*/     {
/*61_3*/         conteudo.add(
/*61_4*/             new JButton(
/*61_5*/                 "Botao #" + i));
/*61_6*/     }
```

A Figura 2 ilustra o resultado da execução do programa com as instruções das linhas 61\_1 a 61\_6 inseridas no arquivo meuFormulario.java.



Figura 2-Inserção aleatória de muitos botões.

Como definir o posicionamento dos componentes?

O posicionamento dos componentes na janela é definido através do conceito de Layouts.

## 2. Gerente de Layout.

Nesse material de apoio ao estudo, serão analisados três tipos de gerenciadores de layout, a saber:

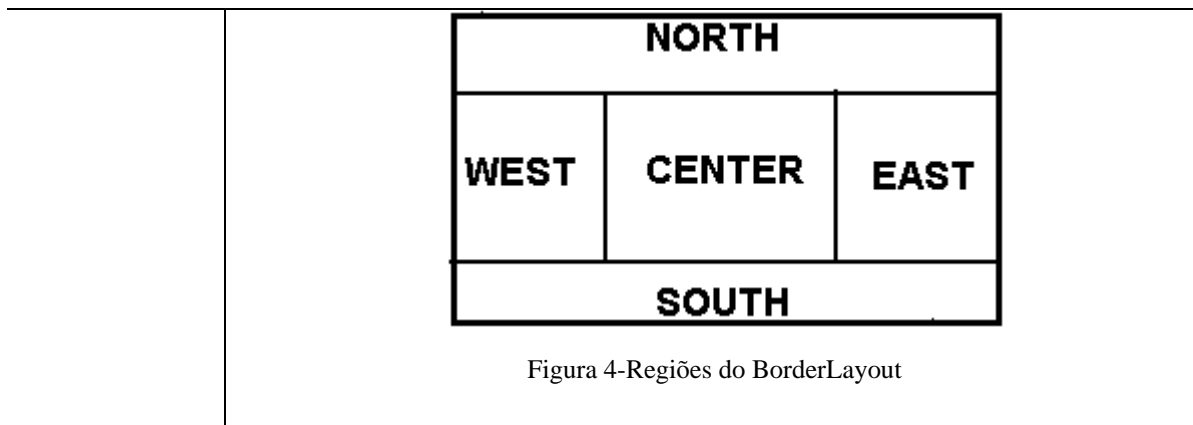
- a) FlowLayout
- b) BorderLayout
- c) GridLayout

a) FlowLayout	FlowLayout é um gerente de Layout que procura alinhar os componentes em um contexto de fluxo de elementos.
Exemplo de FlowLayout:	<p>O seguinte código ao ser inserido entre as linhas 58 e 59 permitirá definir o posicionamento dos componentes.</p> <pre><code>/*58_1*/ conteudo.setLayout(new FlowLayout(0,10,1));</code></pre> <p>Uma instância de FlowLayout é chamada através da palavra new. O construtor dessa classe pode receber três parâmetros inteiros a saber:</p> <pre><code>new FlowLayout(int p1, int p2, int p3).</code></pre> <ul style="list-style-type: none"> <li>• p1: define o alinhamento de acordo com os seguintes valores inteiros: { 0 -&gt; alinhamento a esquerda, 1 -&gt; alinhamento no centro e 2 -&gt; alinhamento a direita}.</li> <li>• p2: define o espaçamento horizontal em pixels.</li> <li>• p3: define o espaçamento vertical em pixels.</li> </ul> <p>A Figura 3 a seguir exemplifica a modificação no posicionamento dos componentes em função do valor de p1.</p>



Figura 3-Posicionamento do FlowLayout

b) BorderLayout:	<p>Outro LayOut que poderia ser utilizado é o BorderLayout (). No código manipulado até o presente, realize as seguintes modificações:</p> <ul style="list-style-type: none"> <li>• Exclua as linhas 61_1 a 61_6.</li> <li>• Modifique a linha /*58_1*/ para conter a seguinte instrução:  <pre><code>/*58_1*/ conteudo.setLayout(new BorderLayout());</code></pre> </li> </ul> <p>Esse Layout define quatro regiões, conforme ilustra a Figura 4. Nessas regiões, qualquer componente pode ser inserido.</p>
------------------	---



Como inserir componentes em uma específica região?

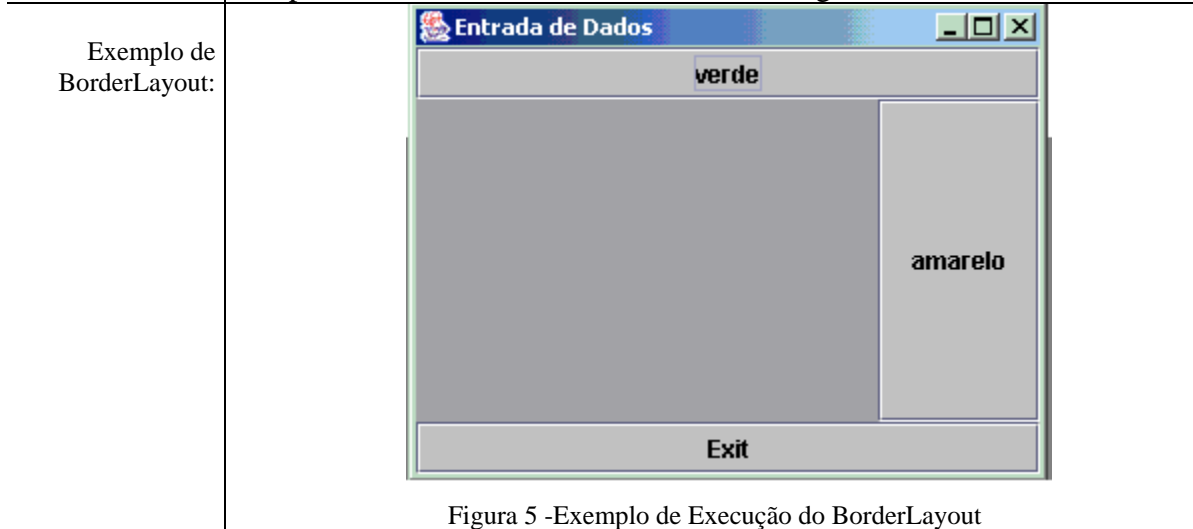
```

/*58_1*/  conteudo.setLayout(new BorderLayout());
/*59*/    conteudo.add(botao1, "East");
/*60*/    conteudo.add(botao2, "North");
/*61*/    conteudo.add(botaoSair, "South");

/*62*/    Container fundo = getContentPane();
/*63*/    fundo.add(conteudo);

```

Ao definir essas regiões, a execução fornecerá a disposição dos componentes de acordo com o ilustrado na Figura 5.



c) GridLayout: O GridLayout permite organizar a área em linhas e colunas. Cada célula definida no grid pode armazenar qualquer componente gráfico.

Exemplo de definição de um GridLayout:

A instrução `new GridLayout(int Linhas , int Colunas)` permite especificar como será dividida a área.


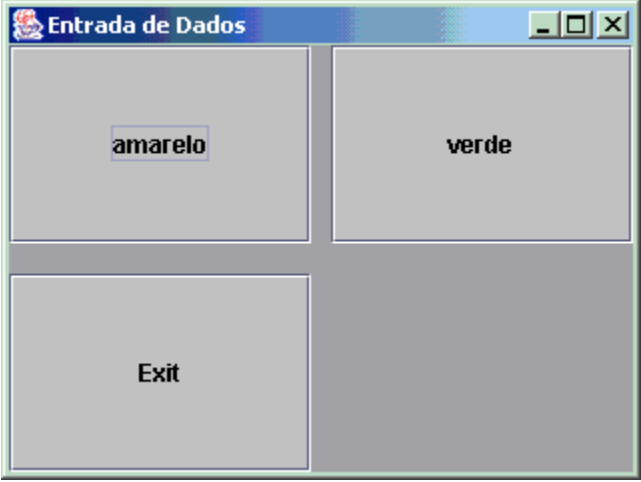
No código, realize as seguintes modificações:

```

/*58_1*/  conteudo.setLayout(new GridLayout(2,2));
/*59*/    conteudo.add(botao1);
/*60*/    conteudo.add(botao2);
/*61*/    conteudo.add(botaoSair);
/*62*/    Container fundo = getContentPane();
/*63*/    fundo.add(conteudo);

```



<p>Exemplo de Execução do GridLayout</p>	<p>Ao definir essas regiões, a execução fornecerá a disposição dos componentes de acordo com o ilustrado na Figura 6.</p>  <p>Figura 6-Exemplo de Execução do GridLayout</p> <p>Observer que o preenchimento da tabela é feita em ordem.</p>
<p>GridLayout com espaços.</p>	<p>O GridLayout permite ainda a definição de um espaçamento entre os componentes. Esse aspecto pode ser definido no seu construtor. Por exemplo, realize as seguintes modificações:</p> <pre data-bbox="488 972 1252 1035">/*58_1*/  conteudo.setLayout(new                                 GridLayout(2,3,10,15));</pre> <p>Nesse caso, o construtor o espaçamento horizontal é 10 pixels e o vertical 15 pixels.</p> <p>A Figura 7 ilustra a execução do GridLayout com espaços.</p>  <p>Figura 7-Exemplo de Execução do GridLayout com espaços</p>

### 3. Outros Componentes Gráficos

Nessa terceira parte, os seguintes componentes serão investigados:

- JComboBox
- JTextArea

- JScrollPane
- JPasswordField
- JLabel

Para essa análise, um arquivo auxiliar deve ser gerado. Nesse arquivo, uma classe genérica denominada Janela será definida com o posicionamento no centro da tela, com o componente JMenu permitindo ao usuário encerrar a aplicação.

A partir dessa classe Janela, através do conceito de Herança, outras janelas serão geradas.

```

////***** arquivo exemploComponentes.java
/*001*/ package interfaceGrafica.componentes;
/*002*/ import javax.swing.*;
/*003*/ import java.awt.event.*;
/*004*/ import java.awt.*;

/*005*/ class Acoes implements ActionListener
/*006*/ {
/*007*/     public void actionPerformed(ActionEvent evt)
/*008*/     {
/*009*/         if (evt.getSource() instanceof JMenuItem )
/*010*/         {JMenuItem quem = (JMenuItem)evt.getSource();
/*011*/             if (quem.getActionCommand() == "Exit")
/*012*/                 System.exit(0);
/*013*/         }
/*014*/     }
/*015*/ }

/*016*/ class Eventos extends WindowAdapter
/*017*/ {
/*018*/     public void windowClosing(WindowEvent e)
/*019*/         {System.exit(0); }
/*020*/ }

/*021*/ class Janela extends JFrame
/*022*/ {
/*023*/     private void centraliza()
/*024*/     {
/*025*/         int x = ((Toolkit.getDefaultToolkit())
/*026*/                 .getScreenSize()).width;
/*027*/         int y = ((Toolkit.getDefaultToolkit())
/*028*/                 .getScreenSize()).height;
/*029*/         setLocation((x - getSize().width)/2,
/*030*/                       (y-getSize().height)/2);
/*031*/     }

/*032*/     private JMenuBar criarMenus()
/*033*/     {
/*034*/         JMenuBar menu = new JMenuBar();
/*035*/         JMenu mFile = new JMenu("File");
/*036*/         mFile.setMnemonic('F');

/*037*/         JMenuItem mExit = new JMenuItem("Exit",'E');
/*038*/         mExit.addActionListener(new Acoes());

```

```

/*036*/      mFile.add(mExit);
/*037*/      menu.add(mFile);

/*038*/      return menu;
/*039*/      }

/*040*/      public Janela(){this("exemplo");}
/*041*/      public Janela(String titulo)
/*042*/      {
/*043*/          setJMenuBar(criarMenus());

/*044*/          addWindowListener( new Eventos());
/*045*/          setTitle(titulo);
/*046*/          setSize(320,240);
/*047*/          centraliza();
/*048*/      }
/*049*/  }

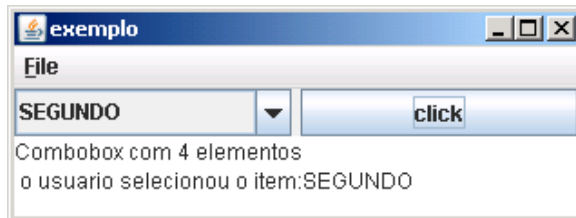
/*050*/public class exemploComponentes
/*051*/{
/*052*/      public static void main(String[] a)
/*053*/      { (new Janela()).show();}
/*054*/}

```

### 3.1 JComboBox / JTextArea

Componente que permite ao usuário selecionar um elemento em um conjunto de elementos.

Código Exemplo:



```

/////***** arquivo exemploComboBox.java
/*001*/ package interfaceGrafica.componentes;
/*002*/ import javax.swing.*;
/*003*/ import java.awt.event.*;
/*004*/ import java.awt.*;

/*004*/ public class exemploJComboBox extends Janela
/*005*/ {
/*006*/     private JPanel conteudo = new JPanel();
/*007*/     private JPanel conteudo2 = new JPanel();

/*008*/     private JButton bt = new JButton("click");
/*009*/     private JTextArea msg = new JTextArea();
/*010*/     private JComboBox combo;

/*011*/     public exemploJComboBox()
/*012*/     {
/*013*/         conteudo.setLayout(new GridLayout(1,2,5,10));
/*014*/         Object [] op = {"PRIMEIRO", "SEGUNDO", "TERCEIRO"};
/*015*/         combo = new JComboBox(op);
/*016*/         combo.addItem("QUARTO");

```

```

/*017*/ bt.addActionListener(new AbstractAction()
/*018*/ { public void actionPerformed(ActionEvent evt)
/*019*/ { int q = combo.getItemCount();
/*020*/     msg.setText("Combobox com "+ q + " elementos"
/*021*/     + "\n o usuario selecionou o item: "
/*022*/     + combo.getSelectedItem());}
/*023*/ }
/*024*/ );

/*025*/ setSize(320,120);
/*026*/ conteudo.add(combo);
/*027*/ conteudo.add(bt);

/*028*/ conteudo2.setLayout(new BorderLayout());
/*029*/ conteudo2.add(conteudo, "North");
/*030*/ msg.setEditable(false);
/*031*/ conteudo2.add(msg, "Center");
/*032*/ Container fundo = getContentPane();
/*033*/ fundo.add(conteudo2);
/*034*/ }

/*035*/ public static void main(String[] a)
/*036*/ {
/*037*/     (new exemploJComboBox()).setVisible(true);
/*038*/ }

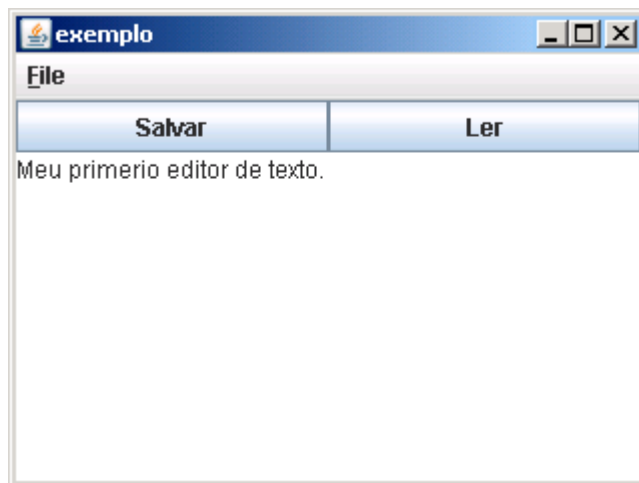
```

### 3.2 JTextArea / JScrollPane / Arquivos de Entrada e Saída / Tratamento de Erros

Permite que o usuário edite texto.

No exemplo a seguir, também é apresentado como acessar arquivos texto para leitura e escrita. Outro aspecto que merece destaque é a rotina de tratamento de erros das linhas / 20 e 27 / 28.

Código Exemplo:



```

/////***** arquivo exemploJTextArea.java

/*001*/ package interfaceGrafica.componentes;
/*002*/ import javax.swing.*;
/*003*/ import java.awt.event.*;
/*004*/ import java.awt.*;
/*005*/ import java.io.*;

/*006*/ public class exemploJTextArea extends Janela
/*007*/ {
/*008*/     private JTextArea texto =new JTextArea(80,80);

```

```

/*009*/ private JPanel conteudo = new JPanel();
/*010*/ private JPanel conteudo2 = new JPanel();
/*011*/ private JButton btSalvar = new JButton("Salvar");
/*012*/ private JButton btLer = new JButton("Ler");
/*013*/ private void lerArquivo(File name)
/*014*/ {
/*015*/     if (name.exists())
/*016*/     {
/*017*/         if (name.isFile())
/*018*/         {
/*019*/             try
/*020*/             {
/*021*/                 RandomAccessFile r = new RandomAccessFile(name, "r");
/*022*/                 StringBuffer buf = new StringBuffer();
/*023*/                 String text;
/*024*/                 texto.setText("");
/*025*/                 while ( (text=r.readLine()) != null)
/*026*/                     texto.append(text + "\n");
/*027*/             }
/*028*/             catch (IOException erro)
/*029*/             {JOptionPane.showMessageDialog(null
/*030*/                 , "File Error"
/*031*/                 , "Nao foi possivel ler"
/*032*/                 , JOptionPane.ERROR_MESSAGE);}
/*033*/             }
/*034*/         }
/*035*/     }

/*036*/ private void salvarArquivo(File name)
/*037*/ {
/*038*/     try{
/*039*/         if (name.exists()) {name.delete();}
/*040*/         RandomAccessFile r = new
/*041*/             RandomAccessFile(name, "rw");
/*042*/         StringBuffer buf = new StringBuffer();
/*043*/         r.writeBytes(texto.getText());
/*044*/     }
/*045*/     catch (IOException erro)
/*046*/     {JOptionPane.showMessageDialog(this
/*047*/         , "File Error"
/*048*/         , "Erro de Escrita"
/*049*/         , JOptionPane.ERROR_MESSAGE);}
/*050*/ }

/*051*/ public exemploJTextArea()
/*052*/ {
/*053*/     conteudo.setLayout(new BorderLayout());
/*054*/     conteudo2.setLayout(new GridLayout(1,2));

/*055*/     btSalvar.addActionListener(new AbstractAction()
/*056*/     { public void actionPerformed(ActionEvent evt)
/*057*/       {salvarArquivo(new File("exemplo.txt"));}
/*058*/     });

/*059*/     btLer.addActionListener(new AbstractAction()
/*060*/     { public void actionPerformed(ActionEvent evt)
/*061*/       {lerArquivo(new File("exemplo.txt"));}
/*062*/     });
/*063*/     conteudo2.add(btSalvar);

```

```

/*064*/      conteudo2.add(btLer);

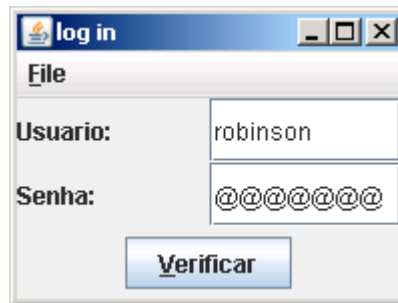
/*065*/      conteudo.add(conteudo2,"North");
/*066*/      conteudo.add(new JScrollPane(texto),"Center");
/*067*/      Container fundo = getContentPane();
/*068*/      fundo.add(conteudo);
/*069*/      }

/*070*/      public static void main(String[] a)
/*071*/      {
/*072*/          (new exemploJTextArea()).setVisible(true);
/*073*/      }
/*074*/  }

```

### 3.3 JPasswordField / Listener do Teclado

JPasswordField permite que o usuário entre com uma password.



```

//////***** arquivo exemploJPassword.java
/*001*/ package interfaceGrafica.componentes;
/*002*/ import javax.swing.*;
/*003*/ import java.awt.event.*;
/*004*/ import java.awt.*;

/*005*/ public class exemploJPassword extends Janela
/*006*/      implements KeyListener
/*007*/ {
/*008*/     //define que a senha terá tamanho de 8.
/*009*/     private JPasswordField senha = new JPasswordField(8); //
/*010*/     private JPanel conteudo = new JPanel();
/*011*/     private JPanel conteudo2 = new JPanel();
/*012*/     private JPanel conteudo3 = new JPanel();
/*013*/     private JButton btOk = new JButton("Verificar");
/*014*/     private JTextField editUsuario = new JTextField();

/*015*/     public void keyReleased(KeyEvent evt) {}
/*016*/     public void keyTyped(KeyEvent evt) {}

/*017*/     public void keyPressed(KeyEvent evt)
/*018*/     {if (evt.getKeyCode() == java.awt.event.KeyEvent.VK_ENTER)
/*019*/         { conteMe();}
/*020*/     }

/*021*/     private void conteMe()
/*022*/     { System.out.println("Usuario = "
/*023*/         + editUsuario.getText()
/*024*/         //o compilador vai fornecer um warning para getText().
/*025*/         + "\n Senha=" + senha.getText()); }

/*026*/     private void funcButton()

```

```

/*026*/ {
/*027*/     btOk.setMnemonic('V');
/*028*/     btOk.addActionListener(new AbstractAction()
/*029*/     { public void actionPerformed(ActionEvent evt)
/*030*/         { conteMe(); } });
/*031*/ }

/*032*/ private void monitorTeclado()
/*033*/ {
/*034*/     senha.addKeyListener(this);
/*035*/     btOk.addKeyListener(this);
/*036*/ }

/*037*/ public exemploJPassword(String texto)
/*038*/ {
/*039*/     super(texto);
/*040*/     setSize(200,150);
/*041*/     // carrega as funcoes do listener.
/*042*/     monitorTeclado();
/*043*/     //modifica o caracter de retorno visual.
/*044*/     senha.setEchoChar('@');

/*045*/     conteudo.setLayout(new BorderLayout());
/*046*/     conteudo2.setLayout(new GridLayout(2,2));
/*047*/     conteudo3.setLayout(new FlowLayout(FlowLayout.CENTER));

/*048*/     conteudo2.add(new JLabel("Usuario:"));
/*049*/     conteudo2.add(editUsuario);
/*050*/     conteudo2.add(new JLabel("Senha:"));
/*051*/     conteudo2.add(senha);
/*052*/     // carrega funcoes no botao
/*053*/     funcButton();
/*054*/     conteudo3.add(btOk);

/*055*/     conteudo.add(conteudo2,"Center");
/*056*/     conteudo.add(conteudo3,"South");
/*057*/     Container fundo = getContentPane();
/*058*/     fundo.add(conteudo);
/*059*/     // seta o foco para o primeiro componente de edicao.
/*060*/     editUsuario.requestFocus();
/*061*/ }

/*062*/ public static void main(String[] a)
/*063*/ {
/*064*/     (new exemploJPassword("log in")).setVisible(true);
/*065*/ }

```

## Desafio:

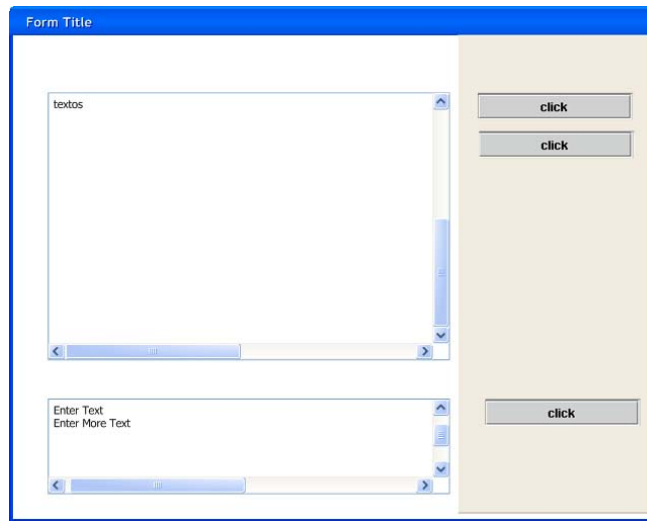
Pesquisar/Criar exemplos para os seguintes componentes visuais:

- JCheckBox
- JRadioButton
- JButtonGroup
- JList

## Desafio 2:

Codificar em Java interfaces similares a:

a)



b)

