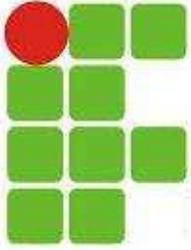

Programação Orientada a Objetos

Professor: Diego Oliveira



**Conteúdo 19:
Interface Gráfica 2**





Componentes

- Sabemos que uma janela do Java (JFrame) pode possuir diversos componentes
- Cada um desses componentes tem uma funcionalidade específica e uma forma de recuperar seus valores e também de alterar suas propriedades
- Nesta aula aprenderemos a utilizar os principais componentes de interface gráfica do Java passo a passo!





Componentes

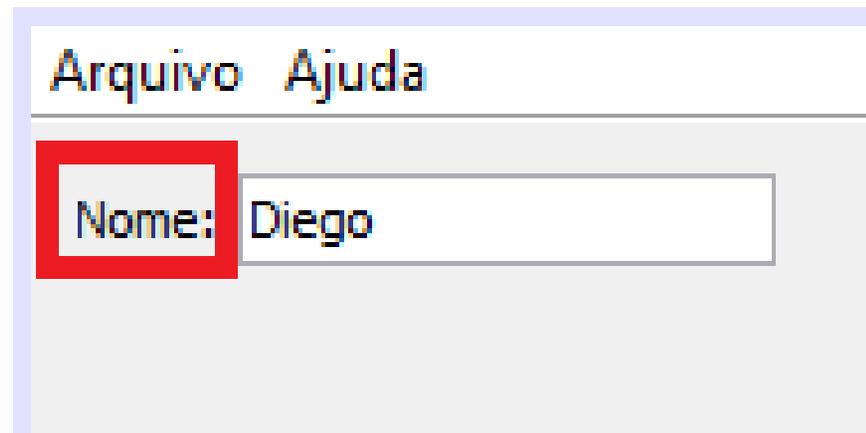
- Lembre-se: os componentes do javax.swing se dividem em Containers, Controls, Menus, Windows, Fillers e AWT (que é a sua base)
- Nesta aula focaremos nos Controls, que são os componentes mais variados e ativos
- Os Containers servem apenas para colocarmos os Controls dentro, como utilizamos os JPanel e JTabbedPane na aula passada





JLabel

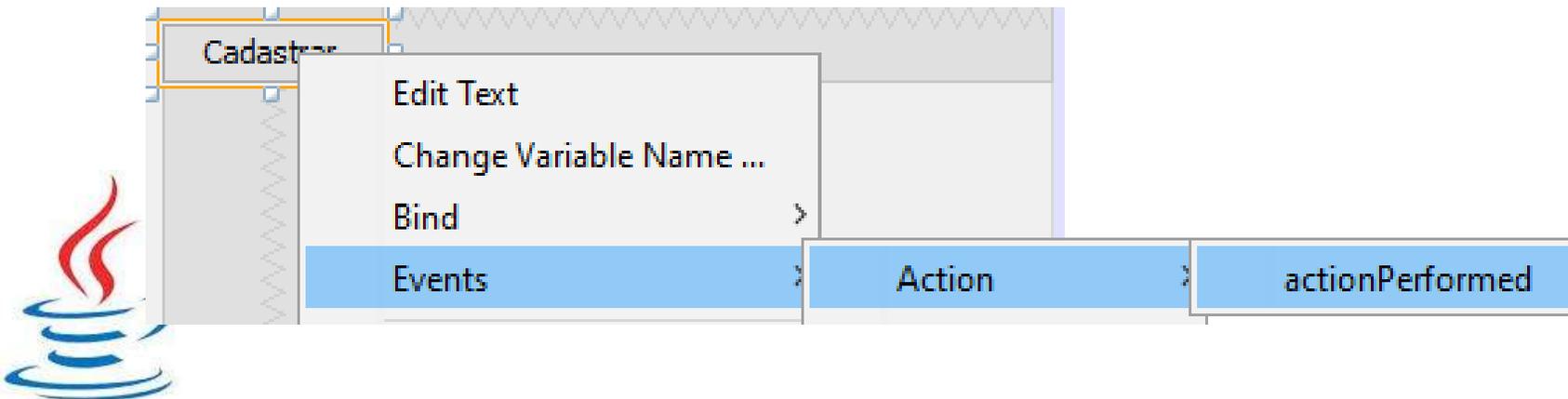
- JLabel é apenas um rótulo para ser colocado ao lado de componentes da janela, indicando o que deve ser digitado no campo por exemplo
- Aperte F2 para trocar o texto do JLabel





JButton

- O JButton é o componente mais simples no qual podemos colocar um evento
- Para adicionar um evento a um botão aperte o botão direito do mouse, selecione Events -> Action -> actionPerformed ou **clique 2x sobre o botão**



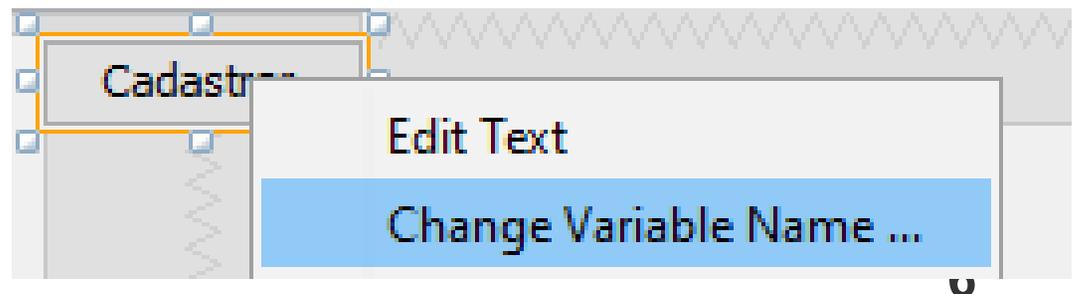


JButton

- No método criado pelo NetBeans, digite o código que será executado quando o botão for clicado

```
private void botaoCadastrarActionPerformed(java.awt.event.ActionEvent evt) {  
    JOptionPane.showMessageDialog(null, "Usuário cadastrado com sucesso");  
}
```

- Sempre renomeie os componentes para facilitar:

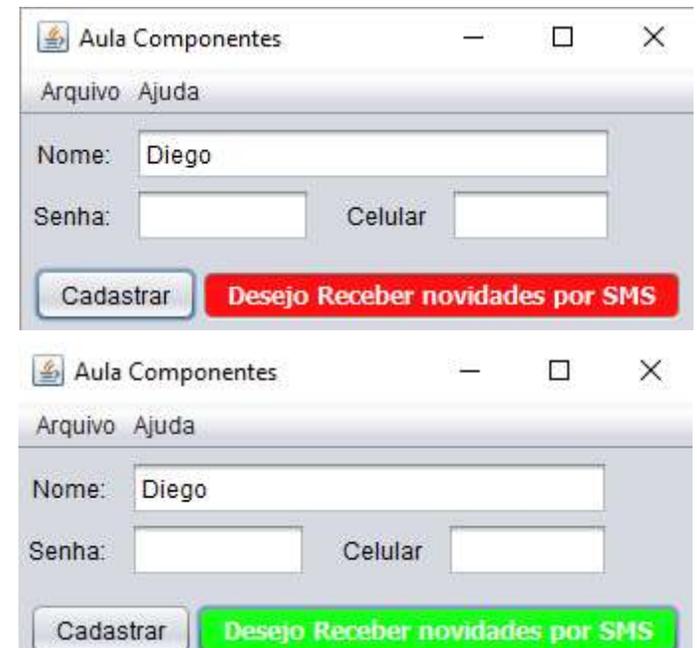




JToggleButton

- Este componente é similar ao botão, porém funciona como um interruptor de lâmpada, ao ser pressionado permanece ligado, só é desligado após ser apertado novamente:
- Utilize `isSelected()` para verificar se o botão está apertado.

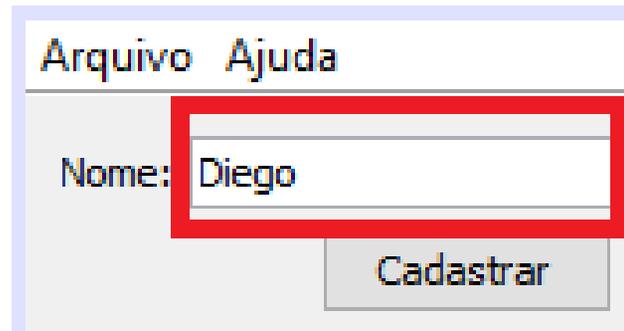
- Para trocar a cor utilize `isSelected()` e `setBackground(Color.RED)`





JTextField

- Este componente é o mais básico para inserção de informações
- É possível digitar um pequeno texto sem caractere de quebra de linha
- Para recuperar o texto digitado basta utilizar `nomeDaVariavel.getText()`





JPasswordField

- Este componente é similar ao anterior
- A diferença básica entre os dois é que neste componente o texto digitado não aparece no monitor, mas pode ser resgatado via código utilizando o método **getPassword()** que retorna um array de bytes. Utilize o **toString()** para converter para String.



Aula Componentes

Arquivo Ajuda

Nome:

Senha:



JTextArea

- Este componente é similar ao JTextField porém ele pode ser expandido e suportar diversas linhas

- O texto digitado pode ser recuperado utilizando-se o método

getText()

- Teste também o

getSelectedText()



A screenshot of a Java Swing window titled "Aula Componentes". The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar is a menu bar with "Arquivo" and "Ajuda". The main content area contains a form with the following elements:

- A text field labeled "Nome:" containing the text "Diego".
- A text field labeled "Senha:" containing six asterisks "*****".
- A text area labeled "Sobre Mim:" containing a list of five items:
 - 1-Professor
 - 2-Jogador de CS
 - 3-Gosto de Naruto
 - 4-Toco violino :P
 - 5-Futuro rico rs
- A button labeled "Cadastrar" at the bottom right.



JCheckBox

- O JCheckBox é muito útil quando precisamos saber apenas se a resposta é Sim ou Não:
- Utilize `isSelected()` para verificar se o componente foi selecionado

```
String interesses = "";  
if(checkMusica.isSelected()){  
    interesses+="Música,";  
}  
if(checkDormir.isSelected()){  
    interesses+="Dormir,";  
}
```



Aula Componentes

Arquivo Ajuda

Nome: Diego

Senha: Celular

Interesses

- Música
- Comida
- Jogos
- Dormir

Cadastrar



JRadioButton

- Este componente é utilizado quando desejamos prover opções mutuamente exclusivas
- Assim como o JCheckBox utilizamos `isSelected()` para verificar se foi marcado

A screenshot of a Java Swing window titled "Aula Componentes". The window contains a registration form with the following fields and sections:

- Nome:** Diego
- Senha:** [Empty field]
- Celular:** [Empty field]
- Interesses:** A group box containing four checkboxes: Música, Comida, Jogos, and Dormir.
- Escolaridade:** A group box containing four radio buttons: Fundamental, Médio, Superior, and Pós-Graduação. The "Pós-Graduação" radio button is selected. This section is highlighted with a red border.

At the bottom of the window, there are two buttons: "Cadastrar" and "Desejo Receber ...".





JButtonGroup

- Como você provavelmente percebeu, ainda é possível marcar vários JComboBox de uma vez
- Para evitar isso utilizamos este componente e adicionamos a ele todas as opções mutuamente exclusivas:

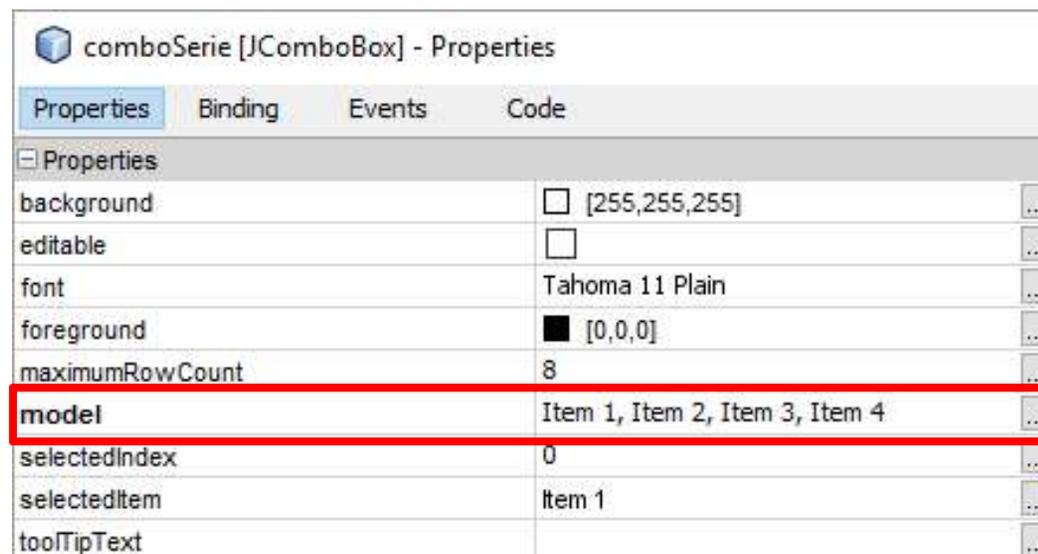
```
grupoEscolaridade.add(radioFundamental);  
grupoEscolaridade.add(radioMedio);  
grupoEscolaridade.add(radioSuperior);  
grupoEscolaridade.add(radioPos);
```





JComboBox

- Este componente é similar a um menu
- Ao clicar nele as opções são exibidas
- Na tela de propriedades é possível adicionar as opções:





JComboBox

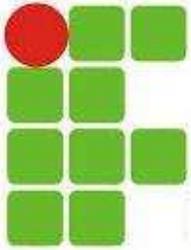
- Para verificar a opção selecionada pelo usuário utilize os métodos `getSelectedIndex()` ou `getSelectedItem()`:

```
String serie = comboSerie.getSelectedItem().toString();
```

OU

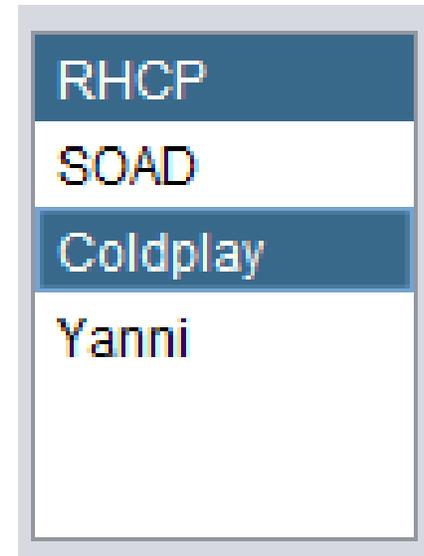


```
if (comboSerie.getSelectedIndex() == 1) {  
    serie = "Game of Thrones";  
}
```



JList

- Listas são utilizadas para selecionar mais de um elemento ao mesmo tempo
- As opções são adicionadas da mesma maneira que no JComboBox
- Para recuperar os valores:



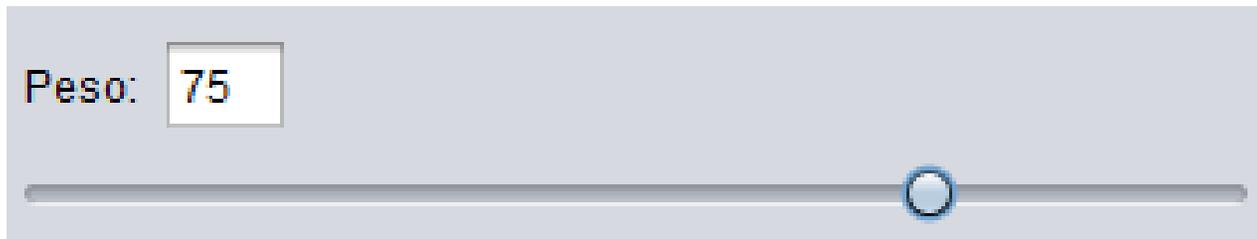
```
String bandas = "";  
List<String> bandasSelecionadas =  
    listaBandas.getSelectedValuesList();  
for (String banda : bandasSelecionadas) {  
    bandas += banda + ", ";  
}
```





JSlider

- Utilizado como barra de seleção de valores, como um controle de volume
- O valor padrão começa em 0 e vai até 100



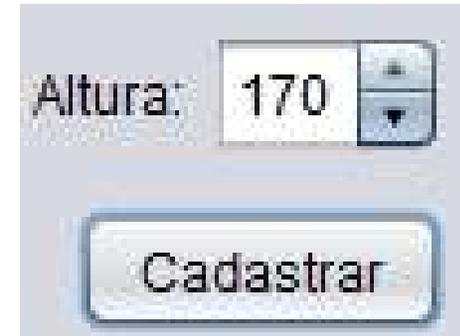
- Para recuperar o valor utilize `getValue()`
- Utilize o evento `ChangeListener->stateChanged()`:

```
private void sliderPesoStateChanged(javax.swing.event.ChangeEvent evt) {  
    campoPeso.setText(sliderPeso.getValue()+"");  
}
```



JSpinner

- O Spinner é similar ao Slider, porém apresenta dois botões, um para aumentar o valor e outro para diminuir:



- Para recuperar o valor utilize `getValue()` :

```
private void spinnerAlturaStateChanged(javax.swing.event.ChangeEvent evt) {  
    altura = spinnerAltura.getValue()+"";  
}
```





JProgressBar

- A barra de progressão é muito utilizada para indicar uma porcentagem ou tempo restante para conclusão de algo
- Para configurar seu valor utilize `setValue(VALOR)` e para resgatar `getValue()`
- Utilize o evento `focusLost()` dos componentes para ter uma barra proativa!

```
private void campoNomeFocusLost(java.awt.event.FocusEvent evt) {  
    if(campoNome.getText() != null) {  
        barra.setValue(barra.getValue()+10);  
    }  
}
```



Resultado dos Exemplos

- Utilizando `focusLost()` em cada componente da janela, quando este perder o foco, caso esteja preenchido, a barra de progressão será aumentada, até chegar o máximo de 100%



É possível ativar o botão cadastro somente nos 100%

Aula Componentes

Arquivo Ajuda

Nome: Diego

Senha: *** Celular: ***

The Walking Dead

RHCP: SOAD, Coldplay, Yanni

Escolaridade: Fundamental, Médio, Superior, Pós-Graduação

Interesses: Música, Comida, Jogos, Dormir

Peso: 75 Altura: 170

Sobre Mim:

1-Professor
2-Jogador de CS
3-Gosto de Naruto
4-Toco violino :P
5-Futuro rico rs

Desejo Receber novidades por SMS

Cadastrar



Exercício

- Utilizando cada um dos componentes mostrados na aula, monte um formulário sobre um tema de seu interesse.
- Após cada componente ser preenchido, vá aumentando a barra de progressão até 100%
- Quando a barra chegar em 100% ative o botão de MOSTRAR DADOS
- Mostre os dados de todos os campos preenchidos em um JOptionPane



• **Vale 1 PONTO EXTRA** 😊



Perguntas?

