

# ALPOO – Aplicações de Linguagem de Programação Orientada a Objetos

Prof. Ms. Gustavo Molina

[msc.gustavo.unip@gmail.com](mailto:msc.gustavo.unip@gmail.com)

Aula 03 – Swing Parte II

# JButton

Um objeto da classe **JButton** é um botão que o usuário pode clicar e, então, uma ação pode ser executada.

## CONSTRUTOR

---

String texto

Icon    imagem

```
JButton btn = new JButton(texto);
```

```
JButton btn = new JButton(texto, imagem);
```

```
JButton btn = new JButton(imagem);
```

# JButton



# JLabel

## JLabel

Um objeto da classe **JLabel** é um dos componentes mais simples, já que servem apenas para exibir textos na tela. O usuário não consegue editar o texto, porém, o programa pode alterar suas características.

### CONSTRUTOR

---

```
String texto  
Icon    imagem  
int     alinhamento // JLabel.LEFT, JLabel.CENTER ou JLabel.RIGHT
```

```
JLabel lbl = new JLabel(texto);  
JLabel lbl = new JLabel(texto, alinhamento);  
JLabel lbl = new JLabel(imagem);  
JLabel lbl = new JLabel(imagem, alinhamento);  
JLabel lbl = new JLabel(texto, imagem, alinhamento);
```

# JCheckBox

## JCheckBox

Um objeto da classe **JCheckBox** é um componente que representa dois estados: selecionado ou não selecionado. O usuário pode alterar o estado da caixa clicando sobre ela. O estado da caixa de seleção é representado por um valor do tipo boolean, sendo: **true** (caixa selecionada) e **false** (caixa não selecionada).

### CONSTRUTOR

---

```
String    texto;    // texto do CheckBox
boolean   estado;   // estado do CheckBox (true ou false)
```

```
JCheckBox cb = new JCheckBox(texto);
JCheckBox cb = new JCheckBox(texto, estado);
```

# JCheckbox



# JRadioButton

Um objeto da classe **JRadioButton** é um componente que representa um grupo de opções (**ButtonGroup**), onde apenas uma delas podem ser selecionada. Inicialmente os botões vêm desmarcados para o usuário que poderá, através do mouse, clicar sobre a sua opção.

Um **ButtonGroup** reúne uma quantidade (2 ou mais) de **JRadioButton**. Para tanto, é necessário criar um objeto do tipo **ButtonGroup** e a ele adicionar os **JRadioButtons** desejados.

```
ButtonGroup bgroup = new ButtonGroup();  
bgroup.add(radioButton1);  
bgroup.add(radioButton2);  
bgroup.add(radioButton3);
```

## CONSTRUTOR

---

```
String      texto;      // texto do RadioButton  
boolean     estado;     // estado do RadioButton (true ou false)  
Icon imagem;           // imagem associada ao RadioButton
```

```
JRadioButton rb = new JRadioButton(texto);  
JRadioButton rb = new JRadioButton(texto, estado);  
JRadioButton rb = new JRadioButton(imagem);  
JRadioButton rb = new JRadioButton(imagem, estado);  
JRadioButton rb = new JRadioButton(texto, imagem);  
JRadioButton rb = new JRadioButton(texto, imagem, estado);
```

# JRadioButton





# JScrollPane

## JScrollPane

Um objeto da classe **JScrollPane** é um componente que permite a exibição de um conteúdo em uma janela com barras de rolagem. Este objeto é útil quando temos uma porção pequena da tela e necessitamos exibir uma grande quantidade de informação ou imagem.

### CONSTRUTOR

```
JScrollPane sp = new JScrollPane(imagem);  
JScrollPane sp = new JScrollPane(imagem, vsbPolicy, hsbPolicy);  
JScrollPane sp = new JScrollPane(vsbPolicy, hsbPolicy);
```

Onde:

- **vsbPolicy** → VERTICAL\_SCROLLBAR\_AS\_NEEDED  
VERTICAL\_SCROLLBAR\_NEVER  
VERTICAL\_SCROLLBAR\_ALWAYS
- **hsbPolicy** → HORIZONTAL\_SCROLLBAR\_AS\_NEEDED  
HORIZONTAL\_SCROLLBAR\_NEVER  
HORIZONTAL\_SCROLLBAR\_ALWAYS

# JScrollPane



Ainda bem que  
é apenas um  
exemplo e não  
condiz com a  
realidade!!!!



# JTextFieldComponent

Um objeto da classe **JTextField** permite a edição de textos pelo usuário. Os componentes pertencentes à esta classe são:

- **JTextField**
- **TextArea**
- **EditorPane**
- **PasswordField**

## JTextField

Componentes **JTextField** têm dois tipos de uso:

ENTRADA → o usuário pode entrar com uma linha de texto (String)

SAÍDA → exibe o conteúdo de uma linha de texto

### CONSTRUTOR

---

```
JTextField tf = new JTextField(int colunas);  
JTextField tf = new JTextField(String inicial);  
JTextField tf = new JTextField(String inicial, int colunas);
```

# JTextComponent

\*\*\* CADASTRO DE ALUNOS \*\*\*



NOME  
[ ]

SEXO  
☐ Feminino ☐ Masculino

IDADE  
[ ]

OBSERVAÇÃO  
Rapazes! Alistem-se no serviço militar

# JTextArea

## JTextArea

Componentes **JTextArea** tem praticamente a mesma funcionalidade de um **TextField**, porém, pode conter múltiplas linhas. Sendo assim, muitos métodos são comuns aos dois componentes.

### CONSTRUTOR

---

```
JTextArea ta = new JTextArea(int linhas, int colunas)
```

```
JTextArea ta = new JTextArea(String conteúdo)
```

```
JTextArea ta = new JTextArea(String conteúdo, int linhas, int colunas)
```

# JTextArea



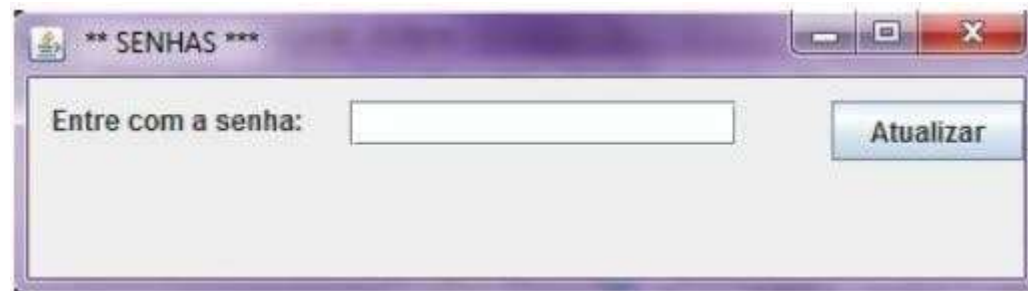
# JPasswordField

Componentes **JPasswordField** tem a mesma funcionalidade de um **JTextField**, porém, os caracteres digitados não aparecem (“ecoam”) na tela. O caractere de máscara padrão é o \* (asterisco), mas é possível modificar este caractere com o método apropriado.

## CONSTRUTOR

---

```
JPasswordField pf = new JPasswordField (int colunas)
JPasswordField pf = new JPasswordField (String texto)
JPasswordField pf = new JPasswordField (String texto, int colunas)
```



# JScrollBar

## JScrollBar

O **JScrollBar** permite que o usuário selecione graficamente um valor deslizando um botão dentro de um intervalo limitado.

Para trabalhar com eles, você deve utilizar um `AdjustmentListener` através do método `JScrollbar.addAdjustmentListener()`. Quando ocorrer algum evento de ajuste, o método ouvinte será chamado.

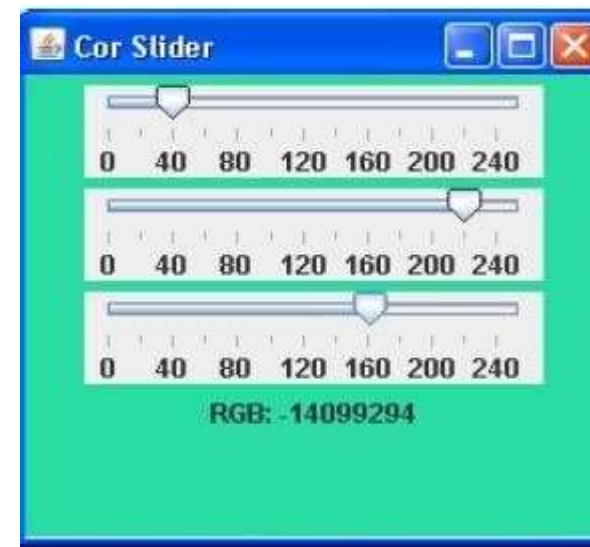


# JScrollBar



# JSlider

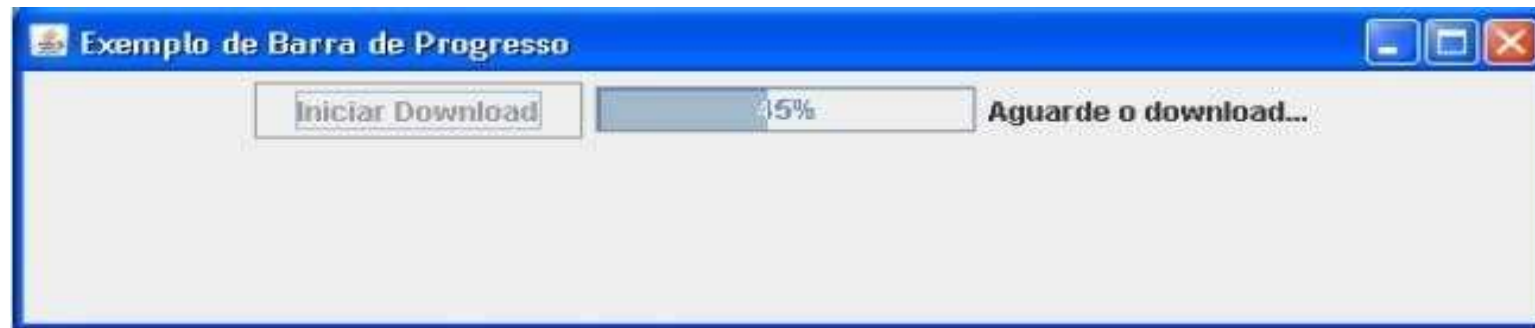
O **JSlider** permite que o usuário selecione graficamente um valor, através do deslizamento de um botão dentro de um intervalo limitado. As marcas da escala (principais e menores) podem ser mostradas, ambas com intervalos variáveis.



# JProgressBar

## JProgressBar

Um objeto do tipo **JProgressBar** cria uma barra que mostra o progresso da execução de uma determinada tarefa/atividade.



# JComboBox

## JComboBox

Um objeto do tipo **JComboBox** cria uma caixa de combinação (lista drop-down) onde o usuário pode escolher apenas uma opção. Esta lista pode ser editável ou não.



# JList

## JList

Um objeto do tipo **JList** cria uma lista suspensa onde o usuário pode selecionar um ou mais itens.



# Menus

## Menus

A classe Swing oferece uma série de componentes para a construção de menus. Os itens de um menu, na verdade, tem seu funcionamento e estrutura semelhantes aos botões (JButton).

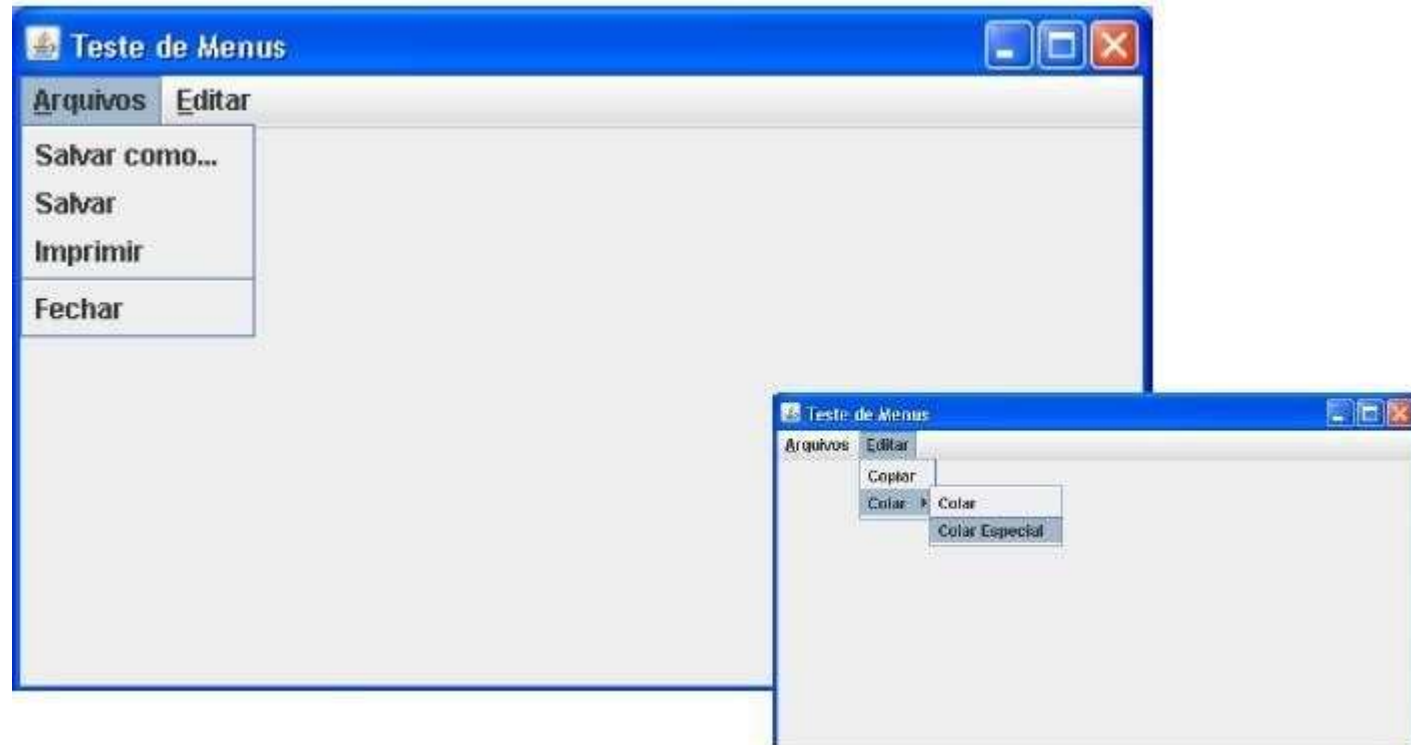
Observações:

- Um item de um menu pode ter texto e ícones
- Os itens de um menu podem ser RadioButton ou CheckBox
- Cada item de menu pode ter uma combinação de teclas de atalho associado à ele

Os componentes são:

**JMenuItem**  
**JCheckBoxMenuItem**  
**JMenu**  
**JMenuBar**

# Menus



# Dúvidas?

