

ARA0066 - PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA



ARA0075

Aula - 05



Introdução - Interface gráfica (swing)

A biblioteca Swing possibilita o desenvolvimento de interfaces elaboradas para um ambiente de computação heterogêneo com interação mais agradável. As aplicações podem seguir uma aparência e comportamento de plataforma nativa, padrão Java ou mesmo personalizada. Permite o gerenciamento de layouts, tratamento de eventos, manipulação de imagens em duas dimensões - 2D e engloba diversos idiomas; entre outros.

Além disso, estende o benefício da Computação Java com a API da acessibilidade que funciona com dispositivos de entradas e saídas, como leitores de tela, terminais Braille e outros. Seus componentes são independentes de plataforma com o padrão MVC (Model-View-Controller ou Modelo-Visão-Controlador).

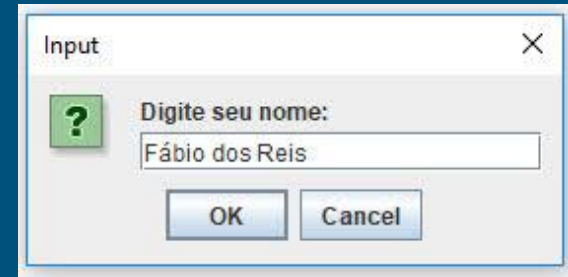


Introdução

Componentes

Na programação orientada a objetos, os componentes são classes que implementam a interface gráfica ao usuário. Os componentes básicos fornecidos pela biblioteca Java Swing são:

- JButton
- JCheckBoxName
- JLabelGenericName
- JTextAreaName
- JTextFieldName
- JPasswordField
- JRadioButtonName
- JOptionPane



Componentes

Para exibição de um texto simples ou imagem com texto, utilizamos o componente JLabel - Rótulos que também pode ser usado para exibir os resultados de um processo. Os métodos mais comuns são:

- `setText()` - Insere um valor de texto para exibir um resultado neste componente.
- `setFont()` - Define o tipo, tamanho e o estilo da fonte (negrito e/ou itálico) da fonte.
- `setIcon()` - Permite que um ícone ou imagem seja usado e exibido na tela.

Componentes

Figura 1: Caixa de texto (JTextField, JPasswordField e JTextArea)

Na figura 1, foram usados componentes para o usuário inserir um valor textual e capturá-lo. A classe `JTextField` possibilita a inserção de uma linha de texto, enquanto a classe `JPasswordField` permite inserir um valor, mas o exibe de modo camuflado, e por fim, a classe `JTextArea` foi utilizada para entrada de várias linhas de texto. Os métodos mais usados são:

- `setText()` - Insere um valor de texto para exibir um resultado neste componente.
- `getText()` - Usado para recuperar o valor da caixa de texto.
- `setFont()` - Define o tipo, tamanho e o estilo da fonte (negrito e/ou itálico) da fonte.
- `setEnabled()` - Permite que a escrita seja desabilitada, e pode ser usado para exibir um resultado como em um `JLabel`.

A imagem mostra uma janela de aplicativo Java Swing com o título "Formulario". A interface contém os seguintes elementos:

- Nombre:** Um campo de texto único (JTextField) contendo o nome "Jhon".
- Clave:** Um campo de senha (JPasswordField) representado por uma série de pontos para ocultar o conteúdo.
- Género:** Dois botões de opção (radio buttons) para "Masculino" (selecionado) e "Femenino".
- Idiomas:** Três caixas de seleção (checkboxes) para "Español", "Ingles" e "Italiano". "Español" e "Ingles" estão marcadas.
- Grado:** Um menu suspenso (JComboBox) com o valor "Primaria" selecionado.
- Comentarios:** Uma área de texto multi-linha (JTextArea) contendo texto de preenchimento (Lorem ipsum).
- Botões:** Dois botões de ação no rodapé: "Enviar" e "Salir".

Componentes

Figura 2: Botões (JRadioButton, JCheckBox e JButton)

Na figura 2 apresenta os componentes que permitem manipular diversas formas gráficas com as quais o usuário pode interagir. A classe JRadioButton possibilita escolher apenas uma opção dentro de um grupo de alternativas, enquanto a classe JCheckBox permite selecionar mais de uma dessas opções. Finalmente, a classe JButton é usada para executar um evento quando pressionados.

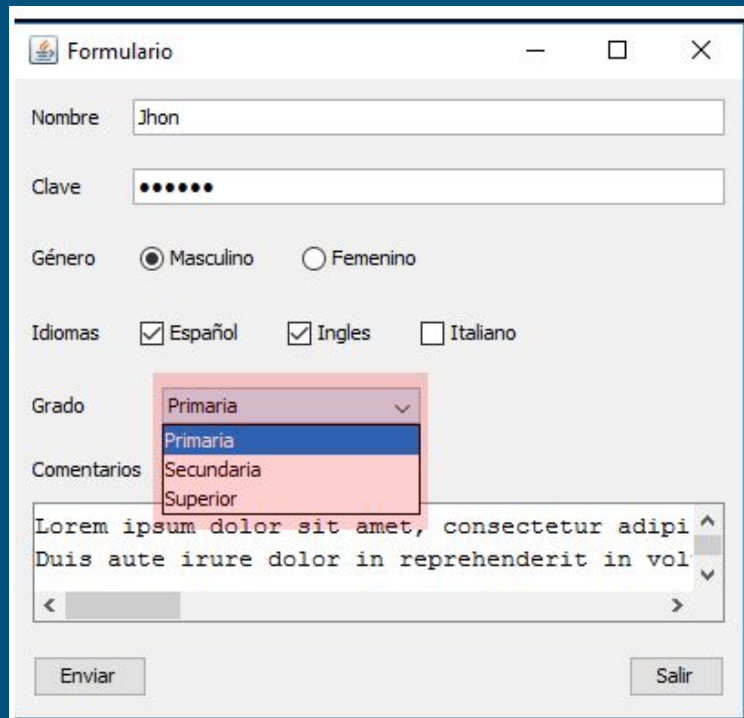
The image shows a Java Swing window titled "Formulario" with standard window controls (minimize, maximize, close). The form contains the following components:

- Nombre:** A text field containing the value "Jhon".
- Clave:** A text field containing seven dots, indicating a password.
- Género:** Two radio buttons labeled "Masculino" (selected) and "Femenino".
- Idiomas:** Three checkboxes labeled "Español" (checked), "Ingles" (checked), and "Italiano" (unchecked).
- Grado:** A dropdown menu currently showing "Primaria".
- Comentarios:** A text area containing the placeholder text "Lorem ipsum dolor sit amet, consectetur adipi" and "Duis aute irure dolor in reprehenderit in vol".
- Buttons:** Two buttons at the bottom, "Enviar" and "Salir", are highlighted with a yellow background.

Componentes

Figura 3: Listas suspensas (JComboBox)

Semelhante a classe JRadioButton, para selecionar apenas uma alternativa de um grupo, mas sem ocupar espaço na janela.



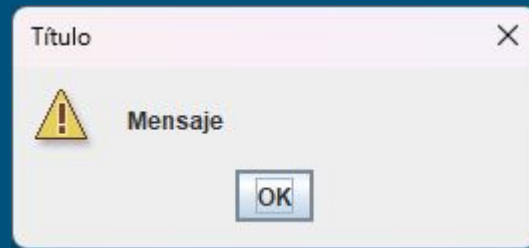
The screenshot shows a Java Swing window titled "Formulario" with standard window controls (minimize, maximize, close). The form contains the following elements:

- Nombre:** A text field containing the value "Jhon".
- Clave:** A password field represented by a series of dots.
- Género:** Two radio buttons labeled "Masculino" (selected) and "Femenino".
- Idiomas:** Three checkboxes labeled "Español" (checked), "Ingles" (checked), and "Italiano" (unchecked).
- Grado:** A JComboBox dropdown menu that is currently open, showing three options: "Primaria" (highlighted in blue), "Primaria" (in red), and "Superior".
- Comentarios:** A text area containing the placeholder text "Lorem ipsum dolor sit amet, consectetur adipi" and "Duis aute irure dolor in reprehenderit in vol".
- Buttons:** Two buttons at the bottom, "Enviar" and "Salir".

Componentes

Figura 4: Caixas de diálogo (JOptionPane)

Um código para exibir uma caixa com uma mensagem, um ícone e botões seria:



```
JOptionPane.showMessageDialog( parentComponent: null, message: "Mensaje", title: "Título", JOptionPane.WARNING_MESSAGE);
```


Hierarquia de componentes

Agora que vimos sobre os componentes, podemos falar sobre a hierarquia existente entre eles dentro da biblioteca Swing. Tal hierarquia é utilizada na funcionalidade desses elementos, e também é utilizada para o entendimento de uma aplicação Swing. Essa hierarquia é comumente visualizada como uma árvore de componentes.

A árvore de componente trabalha com três tipos de elementos em uma aplicação:

- Recipiente de nível superior: são os elementos usados geralmente como base, isto é, fornece um lugar para se utilizar outros elementos. Exemplos desse tipo seriam as classes JFrame, JDialog e JApplet.
- Recipiente intermediário: são os elementos usados somente para manusear o posicionamento de elementos como o botão e o rótulo. Exemplos desse tipo seriam as classes JPanel, JScrollPane e JTabbedPane.
- Recipiente atômico: são os elementos que não armazenam outros elementos. Por sua vez, trata-se de uma entidade auto suficiente, utilizada para apresentar informações ao usuário ou para receber informações fornecidas pelo usuário. Utilizam as classes: JButton, JLabel, JComboBox, JTextField e JTable, entre outras.

Hierarquia de componentes

A classe JFrame comporta outros três componentes do tipo intermediário que são utilizados para configurações avançadas da interface gráfica: Root Pane, Layered Pane e Glass Pane.

Neste artigo foi pensado em introduzir o Swing, logo, não vamos aprofundar sobre esses componentes. Mas para ilustrar essa hierarquia, apresentamos um diagrama com os componentes:

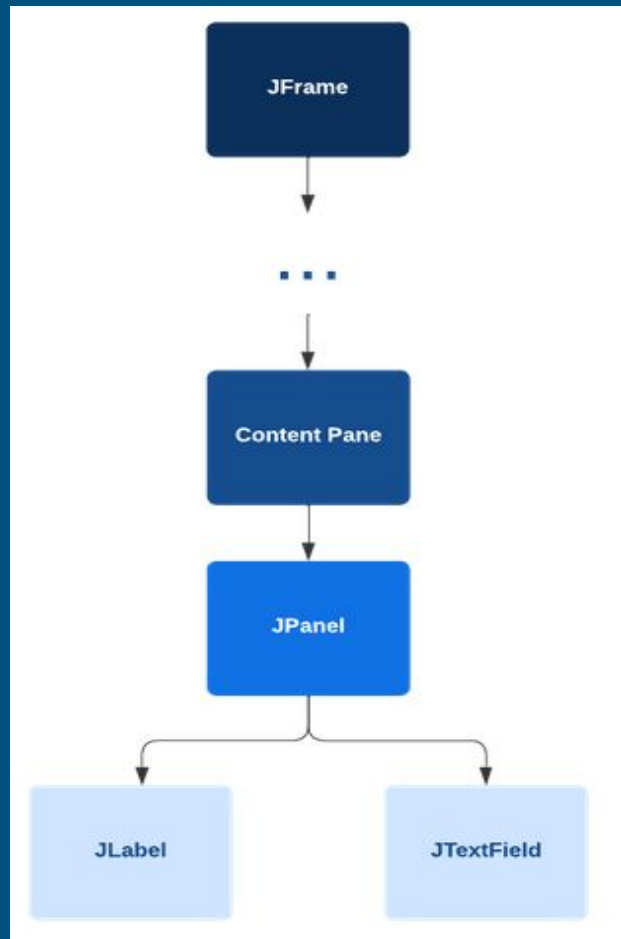


Figura 5: Diagrama de um exemplo de hierarquia dos componentes

Atividade

Converta (desenvolva) um sistema de de cadastro de cliente para uma empresa de prestação de serviços, que possui tanto clientes pessoa física quanto jurídica. Utilizando a interface gráfica Swing (Atividade anterior).