



# INSTITUTO FEDERAL

Mato Grosso

Campus Cuiabá

Cel. Octayde Jorge da Silva

**Docente:** Pedro Neto

**Discente:** Arthur de Abreu Fernandes

**Disciplina:** Linguagem de Programação II **Curso:** 3ºA-7131 Informática

## Resumo dos Slides (42 ao 54)

### 42 ao 45:

Uma **GUI**, (Graphical User Interface), traduzida como Interface Gráfica com Usuário, é onde os resultados são apresentados em modo gráfico, para a melhor visualização para o usuário.

No slide 42 o Prof. mostra alguns problemas com a GUI extensa, como: dificulta identificar o objetivo de cada componente. No mesmo slide ele apresenta a classe **JLabel** e uma subclasse **JComponent**, que serve para exibir texto somente como leitura, uma imagem ou tanto texto como imagem.

No slide 43 ele nos mostra como **JFrame** funciona, associando a um contêiner, e explica que o Java fornece vários gerenciadores de layout, que nos ajuda a posicionar os componentes. No próximo slide, fala que muitas **IDE's** (Ambiente de desenvolvimento integrado), nos ajuda na criação de GUI, escrevendo-as.

No slide 45 o Prof. explica a forma com que os componentes são organizados, da esquerda para a direita, e fala que se caso precise de mais espaço, irá parecer um **FlowLayout**, que redimensiona os componentes, deixando-os com menos linhas.

### 46 ao 48:

Do slide 46 ao 47 o Prof. apenas nos dá 2 dicas : “Se você não adicionar explicitamente um componente GUI a um contêiner, o componente GUI não será exibido quando o contêiner aparecer na tela.” e “Utilize as dicas de ferramenta para adicionar texto descritivo aos componentes GUI. Esse texto ajuda o usuário a determinar o propósito do componente GUI na interface com o usuário”. No slide 48 ele nos fala sobre o método **setToolTipText** (herdado por JLabel de JComponent), que especifica a dica de ferramenta que é exibida quando o usuário posiciona o cursor do mouse sobre um JComponent (como um JLabel).

#### 49 ao 51:

Nos próximos slides ele fala sobre os ícones, que no Java são chamados de Icons, normalmente, um ícone é especificado com um argumento Icon para um construtor ou para o método **setIcon** do componente, outro método também utilizado, é o **ImageIcon**, que suporta mais extensões. Um JLabel pode exibir um Icon, isso facilita na hora de criar um aplicativo. A interface **SwingConstants** (pacote javax.swing) declara um conjunto de constantes de inteiro comuns (como SwingConstants.LEFT) que são utilizadas com muitos componentes Swing.

#### 52 ao 54:

Do slide 52 ao 54 o Prof. Volta a falar sobre o JLabel, mas sobre o posicionamento de cada método:

- O método setText configura o texto exibido no rótulo.
- O método getText recupera o texto atual exibido em um rótulo.
- O método setIcon especifica o Icon a ser exibido em um rótulo
- O método getIcon recupera o Icon atual exibido em um rótulo
- Os métodos setHorizontalTextPosition e setVerticalTextPosition especificam a posição do texto no rótulo.

Por fim, ele fala sobre alguns métodos: **setDefaultCloseOperation**, que indica que o programa deve terminar quando a janela for fechada pelo usuário. O método **setSize** especifica a largura e altura da janela em pixels. E o método **setVisible** com o argumento true exibe a janela na tela.