

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL

AULA 14

LINGUAGEM DE PROGRAMAÇÃO 2
JAVA



PROF. JANIHERYSON FELIPE

CONTEÚDO DESSA AULA

- CONHECER O TERMOS BASICOS DO JAVAFX;
- CONHECER O SCENE BUILDER;
- CRIAR A PRIMEIRA APLICAÇÃO JAVAFX;
- DISCUSSÕES E DÚVIDAS GERAIS.

JAVAFX - CARACTERÍSTICAS

- JavaFX é uma biblioteca de software utilizada para desenvolver aplicações de desktop e interfaces gráficas de usuário (GUI) em Java. Ela oferece uma variedade de componentes gráficos e ferramentas para criar interfaces de usuário interativas e visualmente atraentes.



JAVAFX - CONCEITOS BÁSICOS

- **Stage** (palco): O Stage é a janela principal da aplicação JavaFX. É onde toda a interface gráfica é exibida. Um aplicativo JavaFX geralmente tem apenas um Stage. O Stage contém uma ou mais Scene.



JAVAFX - CONCEITOS BÁSICOS

- **Scene** (cenário): A Scene representa o conteúdo real que é exibido dentro do Stage. Ela contém todos os elementos gráficos, como botões, caixas de texto, imagens, etc. A Scene é como um contêiner que organiza e exibe esses elementos dentro do Stage.



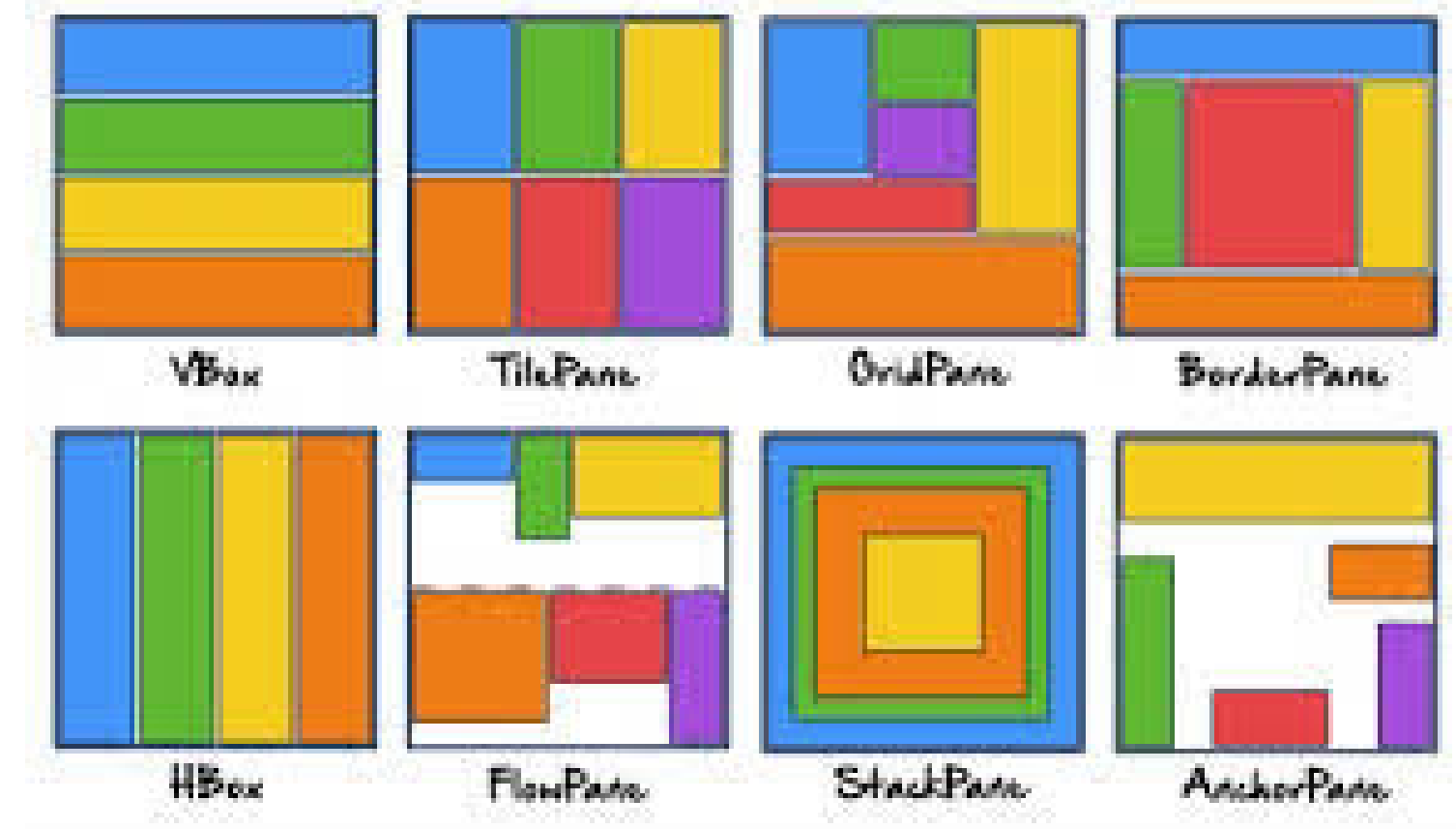
JAVAFX - CONCEITOS BÁSICOS

- **Node** (nó): Um Node é um elemento gráfico individual dentro de uma Scene. Isso pode ser qualquer coisa, desde um simples botão até um layout complexo. Os Nodes são organizados hierarquicamente para formar a estrutura visual da aplicação.



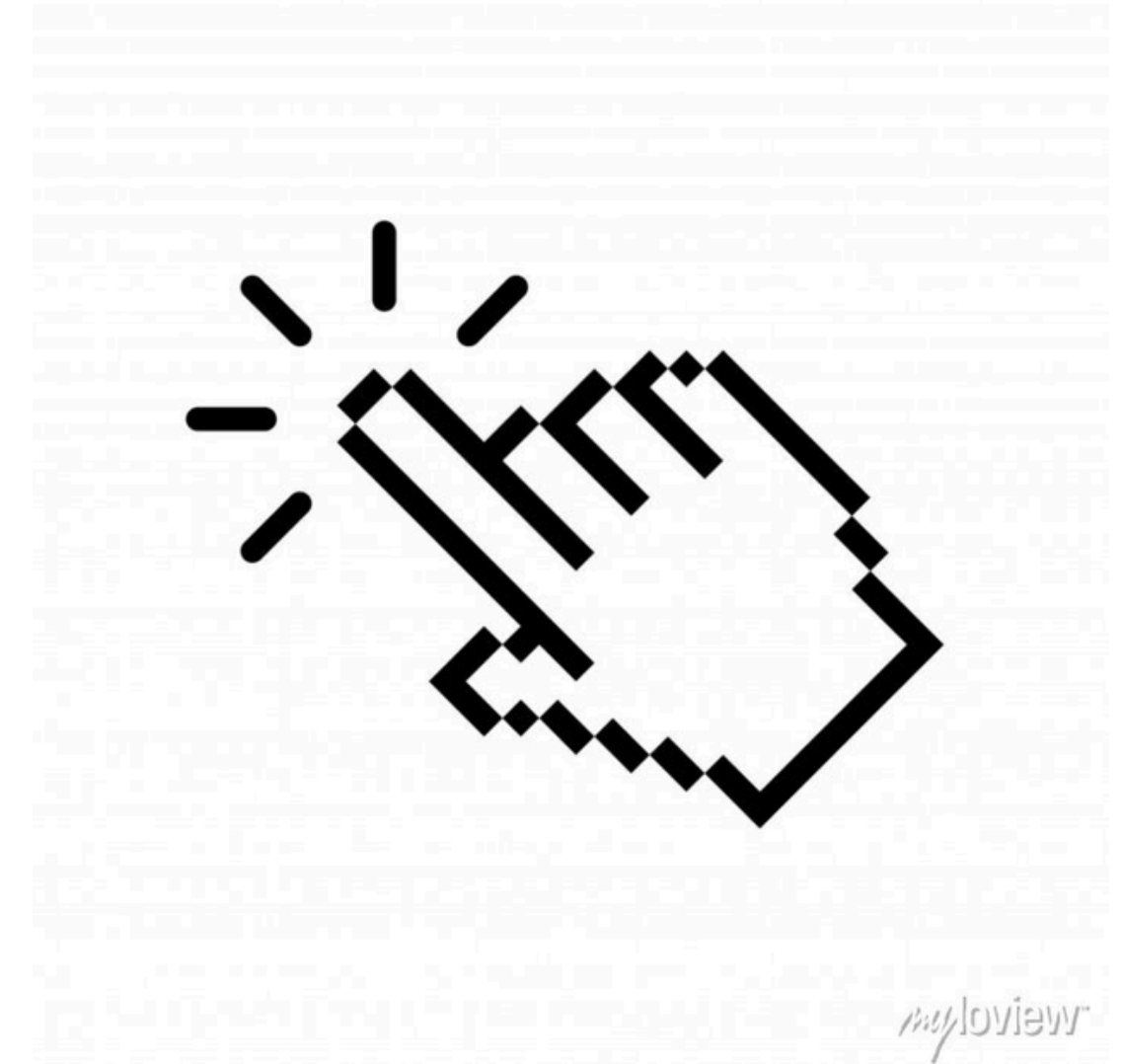
JAVAFX - CONCEITOS BÁSICOS

- **Layouts** (layouts): Em JavaFX, os Layouts são usados para organizar e posicionar os Nodes dentro de uma Scene. Existem diferentes tipos de Layouts, como VBox, HBox, BorderPane, GridPane, etc., cada um com seu próprio comportamento de organização.

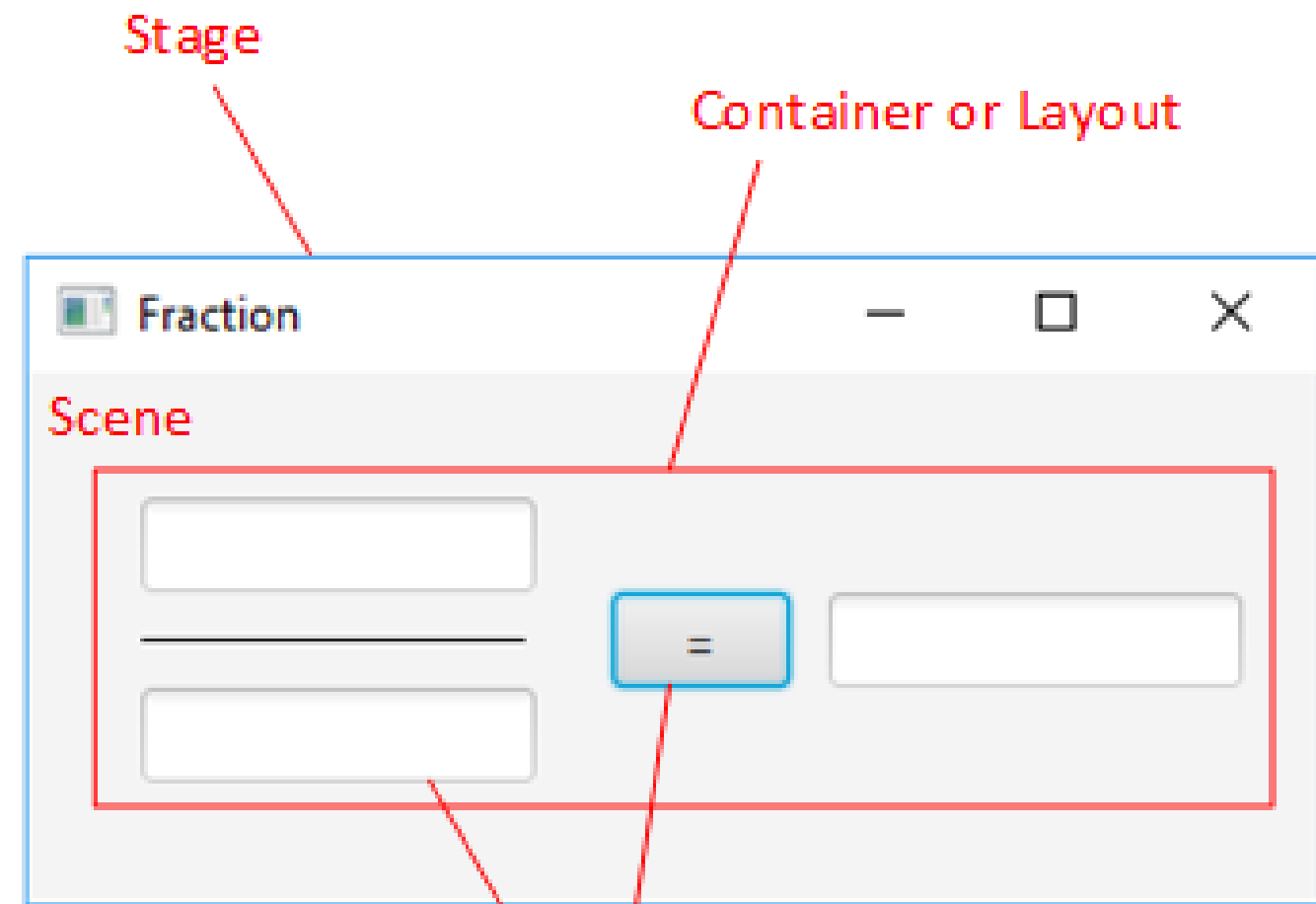


JAVAFX - CONCEITOS BÁSICOS

- **Event Handling** (manipulação de eventos):
JavaFX fornece um sistema de tratamento de eventos para lidar com interações do usuário, como cliques de mouse, pressionamentos de tecla, movimentos de mouse, etc. Isso é feito usando classes como `EventHandler` e `ActionEvent` para responder a esses eventos.



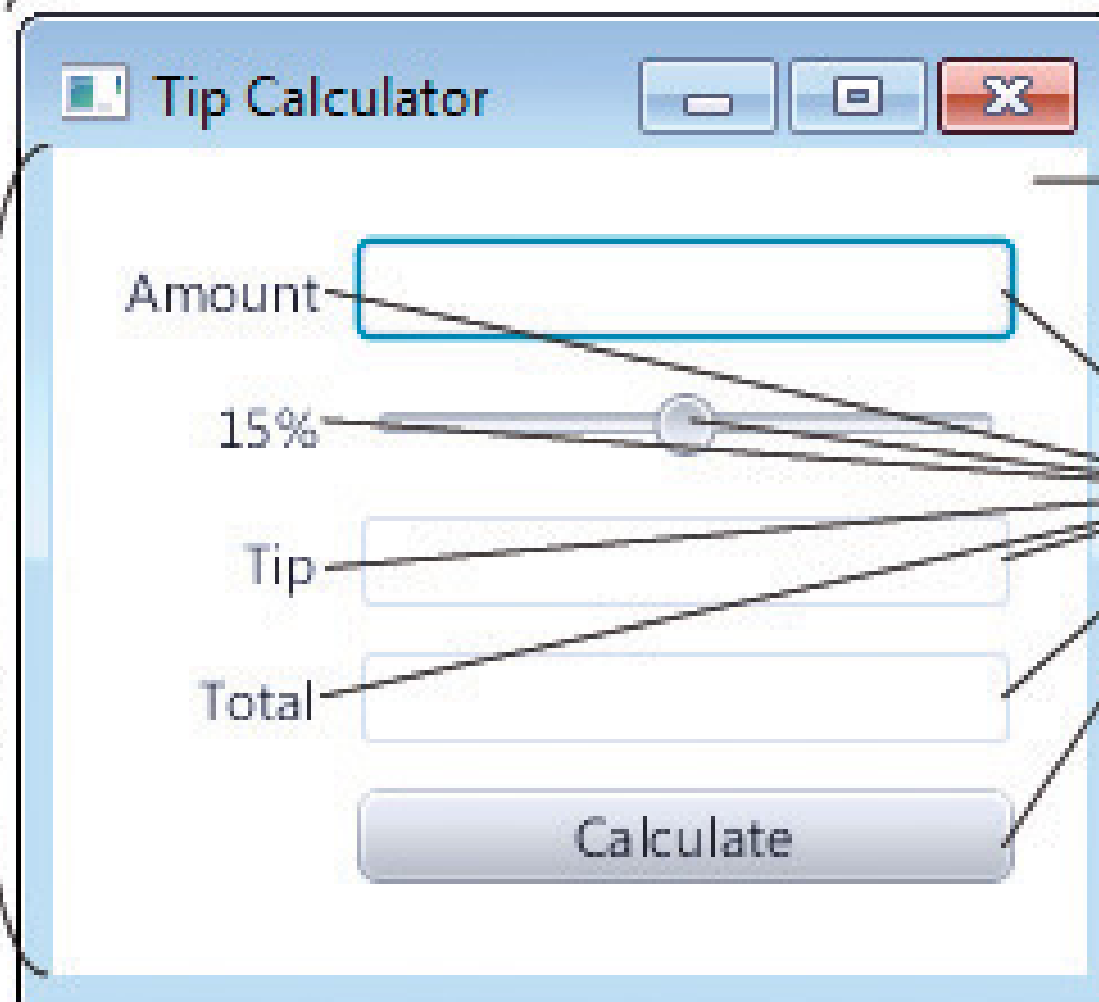
JAVAFX - CONCEITOS BÁSICOS



Controls

The stage's scene contains a scene graph of nodes

The window is known as the stage



The root node of this scene's scene graph is a layout container that arranges the other nodes

Each of the JavaFX components in this GUI is a node in the scene graph

JAVAFX - CLASSE PRINCIPAL

```
public class App extends Application {  
  
    private static Scene scene;  
  
    @Override  
    public void start(Stage stage) throws IOException {  
        Parent root = FXMLLoader.load(getClass().getResource("primary.fxml"));  
        scene = new Scene(root, width:640, height:480);  
        stage.setScene(scene);  
        stage.show();  
    }  
}
```

Run | Debug

```
public static void main(String[] args) {  
    launch();  
}
```

PRINCIPAL.FXML - ARQUIVO DA TELA PRINCIPAL

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.VBox?>

<VBox alignment="CENTER" prefHeight="214.0" prefWidth="192.0" spacing="20.0" xmlns="http://javafx.com/javafx/21"
    <children>
        <Label text="Primary View" />
        <Button fx:id="primaryButton" onAction="#switchToSecondary" text="Switch to Secondary View" />
        <Button fx:id="bt_msg" mnemonicParsing="false" onAction="#imprimirMensagem" text="Mostrar mensagem" />
        <Label fx:id="text_info" text="Texto aqui" />
    </children>
    <padding>
        <Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />
    </padding>
</VBox>
```

PRINCIPALCONTROLLER - CLASSE CONTROLADORA

```
public class PrimaryController {  
  
    @FXML  
    private void switchToSecondary() throws IOException {  
        App app = new App();  
        app.setRoot(fxml: "secondary.fxml");  
    }  
  
    @FXML  
    private void imprimirMensagem(ActionEvent event){  
        System.out.println(x: "Clicou no botão");  
        text_info.setText("Clicou no botão");  
    }  
  
    @FXML  
    private Label text_info;  
}
```

