

JavaFX Studies/ Aula01: Básico do FX e Stages

- Stage: é a janela principal do programa, o contêiner pai de todos, é similar a um JFrame, seria a janela que possui os botões de fechar, minimizar, etc.
- Scene: Similar a um JPanel, é onde eu posso desenhar algo, fazer a interface de algo.
- Scene-Graph: é onde eu posso armazenar diversos Nodes, nodes são os componentes do Scene, eles podem ser buttons, labels, etc.
- Nodes: Eles serão como uma árvore em grafos, possuindo sua raiz, sendo ela o root node que pode ser por exemplo um BorderPane, possuindo seus galhos sendo eles os seus branch nodes, e por último suas folhas sendo elas as Leaf nodes.

-Criando um novo Stage: `Stage stage = new Stage();`

É importante entender algumas coisas, a classe principal do javaFX é a Application, o que ocorre, eu faço a minha classe main herdar a classe Application, e a partir disso existe um método abstrato que eu preciso sobrescrever que é o public void start, que recebe um Stage como parâmetro e inicia a minha tela inicial, sendo assim eu também tenho que usar a partir do método main o launch(args) que inclusive é um método da Classe Application.

-Utilizando o `primaryStage.show();` eu consigo demonstrar o Stage, porém ele ainda não está modificado e não possui nada, nenhum Scene e nenhum Node nele, logo ele é vazio.

```

1  import javafx.application.Application;
2  import javafx.stage.Stage;
3
4  public class App extends Application {
5      Run | Debug
6      public static void main(String[] args){
7          launch(args);
8      }
9
10     @Override
11     public void start(Stage primaryStage) throws
12         Exception{
13         //Stage stage = new Stage();
14         primaryStage.show();
15     }
16 }

```

Porém para prosseguir preciso adicionar uma Scene a esse Stage, mas para instanciar uma Scene eu preciso do root raiz dela, logo eu preciso de um root node para instanciar ela.

Pense no root node como o o layout principal da página, podendo ser muitas coisas como um Border Pane por exemplo.

Um exemplo de root é o da Classe Group, que organiza o layout em um agrupamento de nodes.

```

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class App extends Application {
    Run | Debug
    public static void main(String[] args){
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws
    Exception{
        //Stage stage = new Stage();
        Group root = new Group();// Criando o root
        Scene scene = new Scene(root);// Criando
        a Scene e passando para ela o root

        primaryStage.setScene(scene);//Colocando
        e ligando a Scene ao Stage

        primaryStage.show();//Demostrando o Stage
        na tela.
    }
}

```

Um exemplo de uma simples inicialização de uma simples tela em FX.

-Posso também inicializar o fundo da minha Scene com outra cor, ex: Scene scene = new Scene(root, Color.BLUEVIOLET);

-Posso passar todos os argumentos de uma vez só na declaração do Scene da seguinte forma : Scene scene = new Scene(root,600,600,Color.AQUA);

Posso mudar o Ícone da meu Stage da seguinte forma:

```
Image icon = new Image("assets/teste.png");
```

```
primaryStage.getIcons().add(icon);
```

//Nesse caso a imagem está localizada em uma pasta assets que está dentro da pasta src.

Posso escolher também a largura e altura da janela da minha aplicação, da seguinte forma:

```
primaryStage.setWidth(400);  
primaryStage.setHeight(400);
```

//Posso também escolher se a janela será redimensionável ou não usando:
primaryStage.setResizable(true||false);, sendo true ela pode ser redimensionada e false não.

Posso decidir a posição da meu Stage em relação a tela, da seguinte forma:

```
primaryStage.setX(0);  
primaryStage.setY(0);
```

//Sendo 0 no X a esquerda e 0 no Y o topo.

Princípios da tela cheia:

Posso colocar o meu Stage em Tela cheia utilizando o seguinte comando:

```
primaryStage.setFullScreen(true);
```

//É importante que o Resizable esteja definido como true

Além disso posso utilizar dois comandos, um para informar a dica de sair da tela cheia, e um que muda a tecla que eu pressiono para sair da Tela cheia:

```
primaryStage.setFullScreen(true);  
primaryStage.setFullScreenExitHint("Você pode sair da tela cheia apertando Q");  
primaryStage.setFullScreenExitKeyCombination(KeyCombination.valueOf("q"));
```

//É importante ressaltar que na função setFullScreenExitKeyCombination eu preciso receber por parâmetro um objeto do tipo Exit Key Combination, ele pode ser Definido como KeyCombination.DOWN, que seria a setinha para baixo, ou eu posso passar para ele o valor correspondente da combinação de teclas como KeyCombination.valueOf("%"), nesse caso para sair da tela cheia preciso pressionar SHIFT + 5.

#ID