



# JavaFX 8

## O Primeiro Projeto

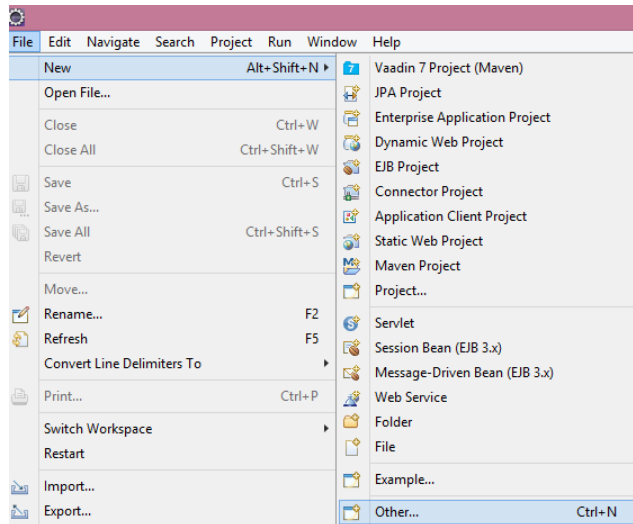
Prof. Vicente Paulo de Camargo

# O Primeiro Projeto JavaFX 8

Crie uma pasta para o workspace de projetos JavaFX

Abra o Eclipse

Selecione File / New / Other

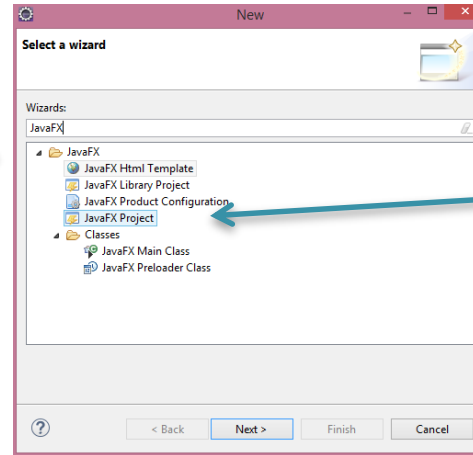


# O Primeiro Projeto JavaFX 8

Uma nova janela será apresentada

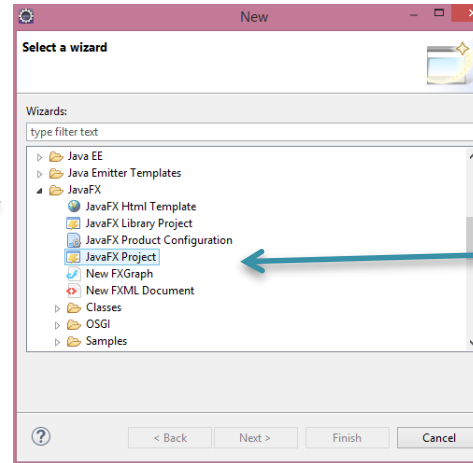
Selecione a opção JavaFX

1



2

Selecione a opção  
JavaFX Project

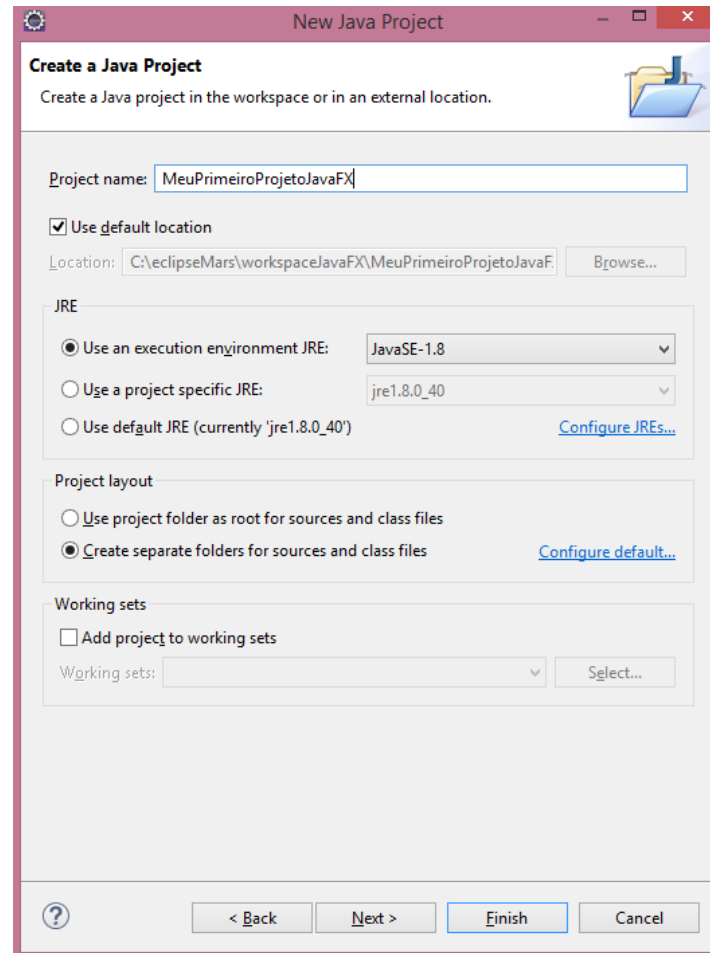


Pressione o botão Next

# O Primeiro Projeto JavaFX 8

Na nova janela **New Java Project**

Informe o nome do projeto **MeuPrimeiroProjetoJavaFX**

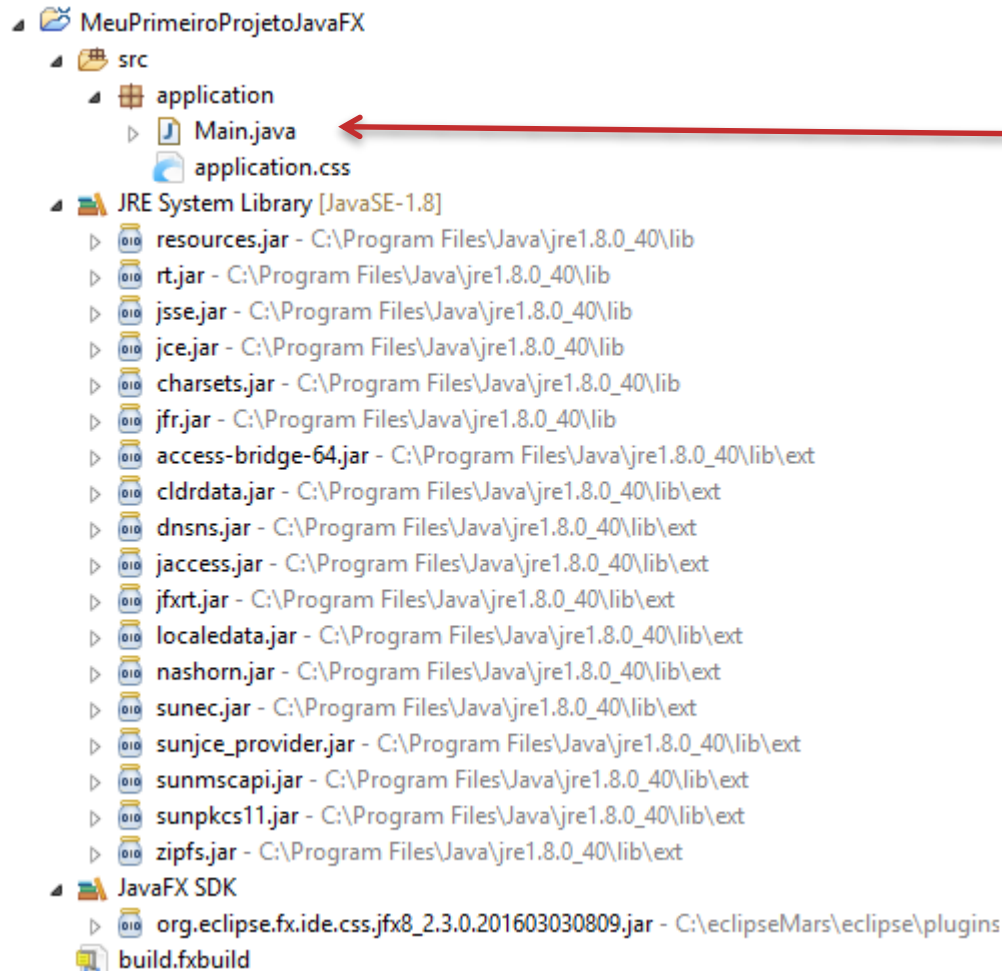


The screenshot shows the 'New Java Project' dialog box in the Eclipse IDE. The title bar reads 'New Java Project'. Inside the dialog, the 'Project name' field is filled with 'MeuPrimeiroProjetoJavaFX'. The 'Use default location' checkbox is checked, and the 'Location' field shows the path 'C:\eclipseMars\workspaceJavaFX\MeuPrimeiroProjetoJavaF'. Under the 'JRE' section, the radio button 'Use an execution environment JRE:' is selected, with 'JavaSE-1.8' chosen from the dropdown. The 'Project layout' section has the radio button 'Create separate folders for sources and class files' selected. The 'Working sets' section has the checkbox 'Add project to working sets' unchecked. At the bottom, there are buttons for '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

Pressione o botão Finish

# O Primeiro Projeto JavaFX 8

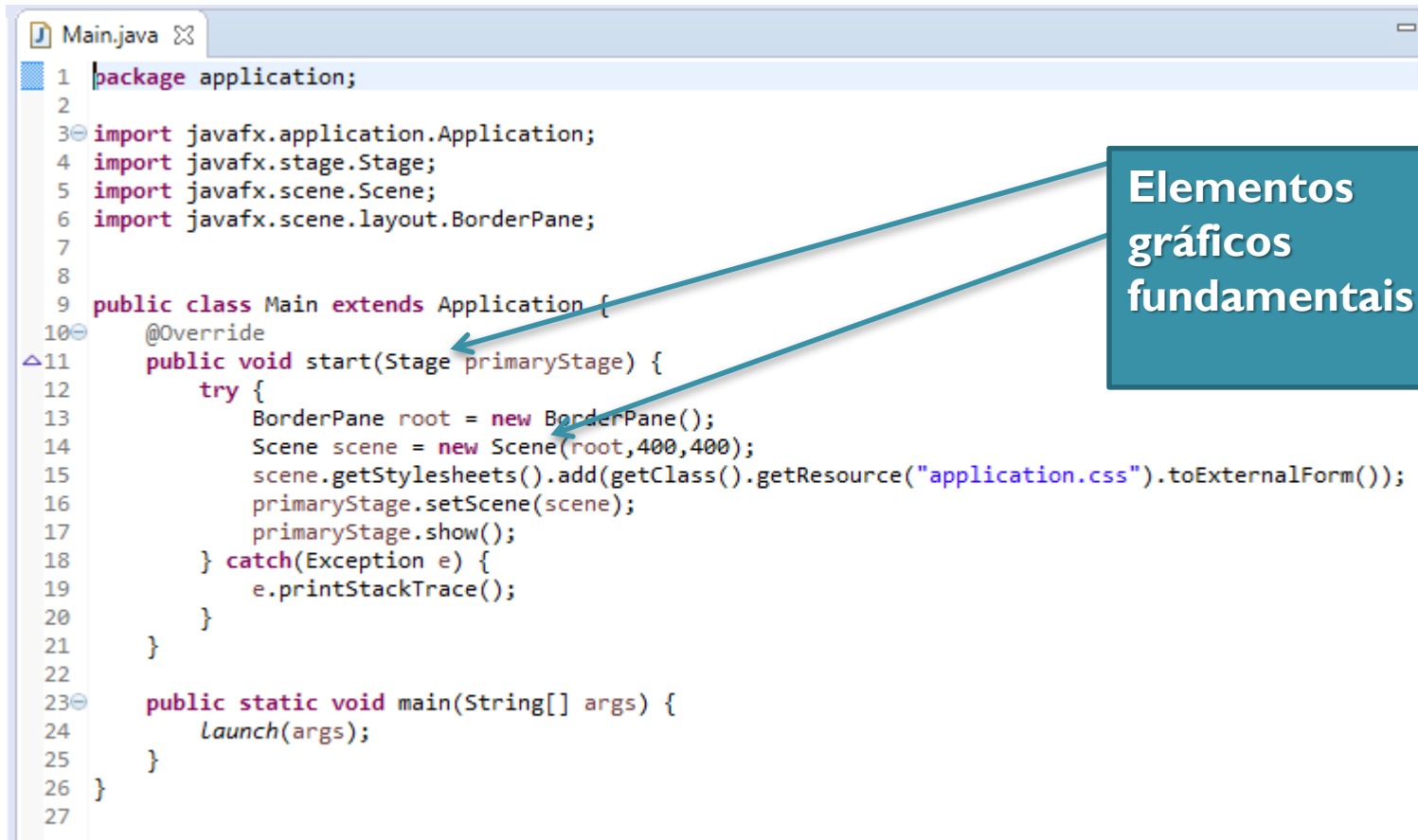
O projeto será criado com a seguinte estrutura



Programa principal  
da aplicação

# O Primeiro Projeto JavaFX 8

O programa principal da aplicação



```
1 package application;
2
3 import javafx.application.Application;
4 import javafx.stage.Stage;
5 import javafx.scene.Scene;
6 import javafx.scene.layout.BorderPane;
7
8
9 public class Main extends Application {
10     @Override
11     public void start(Stage primaryStage) {
12         try {
13             BorderPane root = new BorderPane();
14             Scene scene = new Scene(root, 400, 400);
15             scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
16             primaryStage.setScene(scene);
17             primaryStage.show();
18         } catch (Exception e) {
19             e.printStackTrace();
20         }
21     }
22
23     public static void main(String[] args) {
24         launch(args);
25     }
26 }
27
```

Elementos  
gráficos  
fundamentais

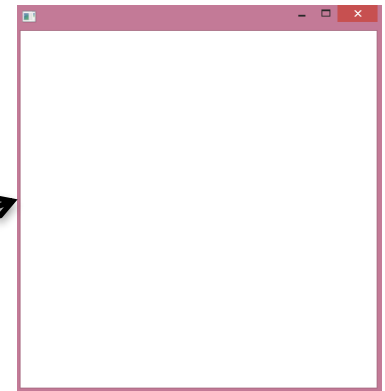
# O Primeiro Projeto JavaFX 8

O programa principal da aplicação

```
Main.java
1 package application;
2
3 import javafx.application.Application;
4 import javafx.stage.Stage;
5 import javafx.scene.Scene;
6 import javafx.scene.layout.BorderPane;
7
8
9 public class Main extends Application {
10     @Override
11     public void start(Stage primaryStage) {
12         try {
13             BorderPane root = new BorderPane();
14             Scene scene = new Scene(root, 400, 400);
15             scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
16             primaryStage.setScene(scene);
17             primaryStage.show();
18         } catch (Exception e) {
19             e.printStackTrace();
20         }
21     }
22
23     public static void main(String[] args) {
24         launch(args);
25     }
26 }
27
```

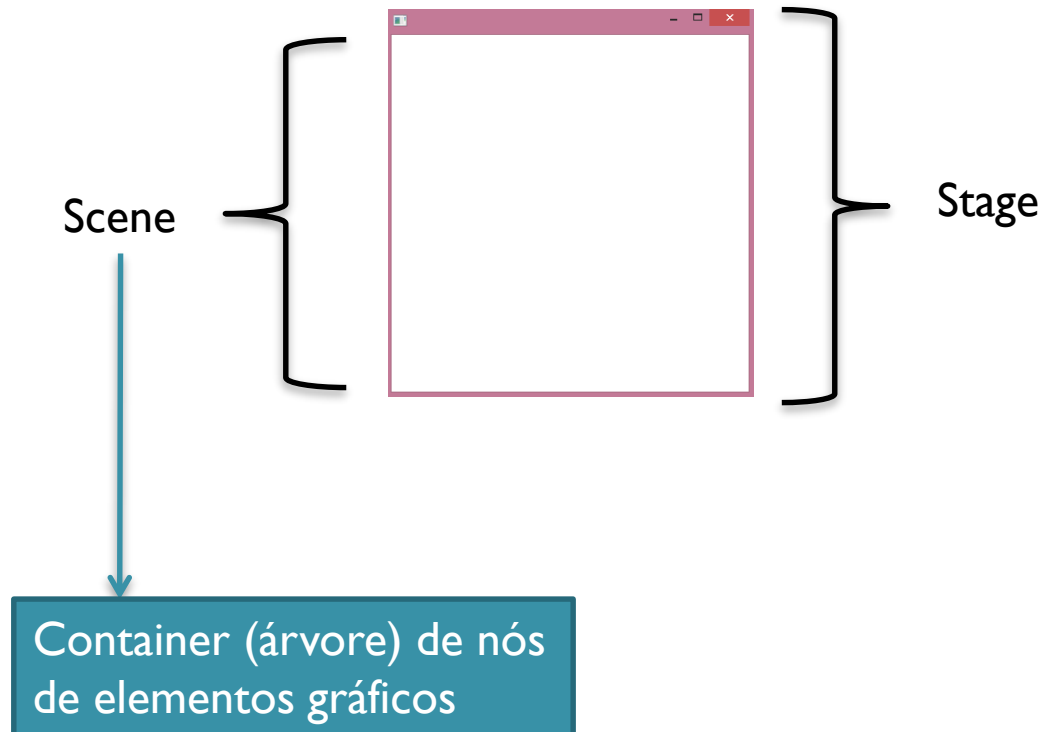
Elementos  
gráficos  
fundamentais

O Resultado



# O Primeiro Projeto JavaFX 8

O programa principal da aplicação

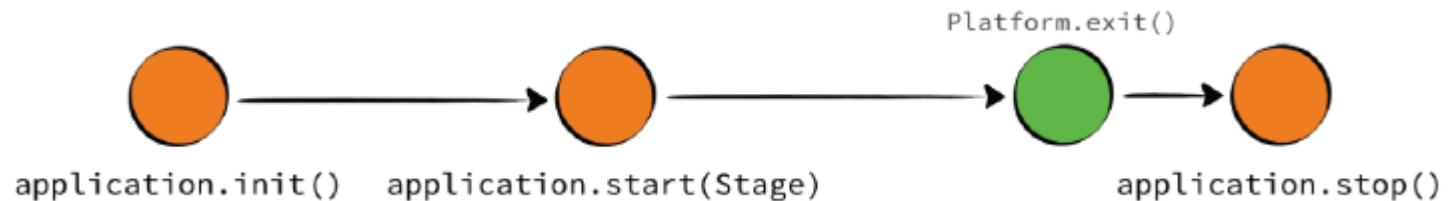




# O Primeiro Projeto JavaFX 8

O Dzone Refcardz de no. 219 (DZONE.COM) destaca a seguinte imagem:

Uma aplicação JavaFX inicia sua execução em uma classe Application. Assim, o método **init()** é o primeiro a ser acionado, como é mostrado no esquema abaixo.

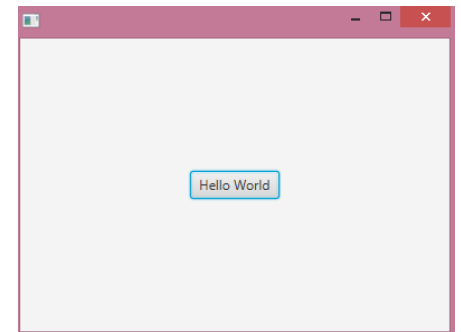


# O Primeiro Projeto JavaFX 8

Exemplo usando componente Button

```
1 package application;
2
3 import javafx.application.Application;
4 import javafx.scene.Scene;
5 import javafx.scene.control.Button;
6 import javafx.scene.layout.StackPane;
7 import javafx.stage.Stage;
8
9 public class Main extends Application {
10     @Override
11
12     public void start(Stage primaryStage) {
13         Button button = new Button("Hello World");
14         button.setOnAction(e -> System.out.println("Hello World"));
15         StackPane myPane = new StackPane();
16         myPane.getChildren().add(button);
17         Scene myScene = new Scene(myPane);
18         primaryStage.setScene(myScene);
19         primaryStage.setWidth(400);
20         primaryStage.setHeight(300);
21         primaryStage.show();
22     }
23
24     public static void main(String[] args) {
25         launch(args);
26     }
27 }
```

Expressão Lambda



# O Primeiro Projeto JavaFX 8

## Expressão Lambda

A expressão Lambda possui duas partes:

A primeira, à esquerda da seta ( $\rightarrow$ ), lista os seus parâmetros

A segunda, à direita da seta ( $\rightarrow$ ), contém o corpo do método

Exemplos:

`(int x, int y)  $\rightarrow$  { return x + y; }`

`()  $\rightarrow$  System.out.println("Olá");`

`(String s)  $\rightarrow$  { System.out.println(s); }`

`()  $\rightarrow$  42`

`()  $\rightarrow$  { return 3.1415 };`

`a  $\rightarrow$  a > 10`

# O Primeiro Projeto JavaFX 8

## Expressão Lambda – complemento

```
Principal.java
1 import java.util.ArrayList;
2
3 public class Principal {
4
5     public static void main(String[] args) {
6
7         //List inteiros = Arrays.asList(10,30, 40, 50, 60, 70);
8         ArrayList<Integer> inteiros = new ArrayList<Integer>();
9         inteiros.add(10); inteiros.add(20); inteiros.add(30); inteiros.add(5); inteiros.add(33);
10        for (Integer inteiro : inteiros)
11        {
12            System.out.println("valor: " + inteiro);
13        }
14        System.out.println("=====");
15        inteiros.forEach(valor -> System.out.println(valor));
16        System.out.println("#####");
17        inteiros.forEach(n -> {
18            if (n % 2 != 0)
19            {
20                System.out.println(n);
21            }
22        });
23    }
24 }
25 }
```


valor: 10  
valor: 20  
valor: 30  
valor: 5  
valor: 33  
=====  
10  
20  
30  
5  
33  
#####  
5  
33

Resultado

# O Primeiro Projeto JavaFX 8

## Expressão Lambda – Vantagem

Permite diminuir a quantidade de código, principalmente em certas funções que utilizam classes internas como é o caso de criar métodos para eventos para componentes gráficos



F I M