

# Java

## Plataforma JavaFX

O objetivo do JavaFX é substituir o Swing.

Como visto, o AWT evoluiu para o Swing e o Swing será substituído pelo JavaFX.

Isso ocorre porque o AWT possuía a limitação do Sistema Operacional tomar decisões; com a Biblioteca Swing, essa possibilidade foi eliminada. Porém, agora, vem a Plataforma JavaFx para eliminar uma limitação do Swing.

O Swing só cria interfaces para Janelas (Mac, Windows, Linux,...). Com o JavaFx será criada uma Aplicação e, esta, funcionará para tudo.

A versão atual do JavaFX cria aplicativos para janela como padrão, mas, também, cria para **celulares e navegadores**.



Criar aplicações para janelas, celulares e navegadores já é garantido na versão atual do JavaFX. Na próxima versão, já está previsto suporte para Vídeo Games, Blu-ray e Sistemas de TV Interativas.

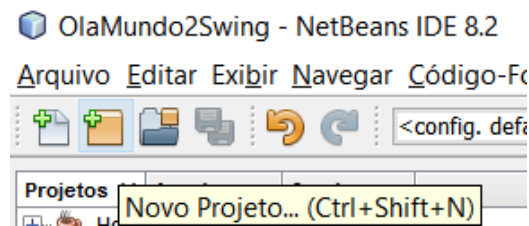


## Tarefa

Para nossa tarefa vamos utilizar a Plataforma, esta possui um SDK específico, baixado na primeira aula, este já vem com o SDK do JavaFX. Vamos realizar o mesmo programa de “Olá, Mundo!” utilizando JavaFX, ou seja, utilizando “FXML”, que é uma versão mais simplificada, onde utilizaremos um pouco HTML, XML e CSS em conjunto.

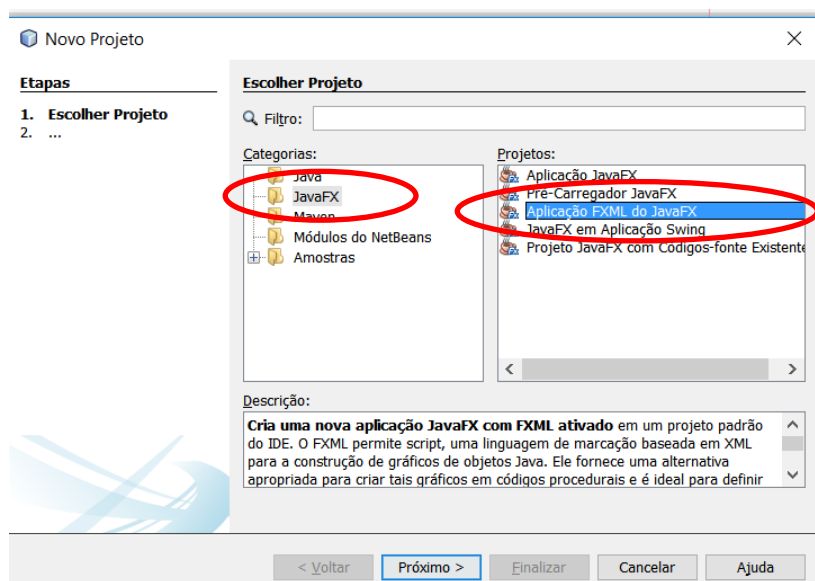
Abra o Netbeans.

Clique no botão Novo Projeto, localizado na Barra de Tarefas.

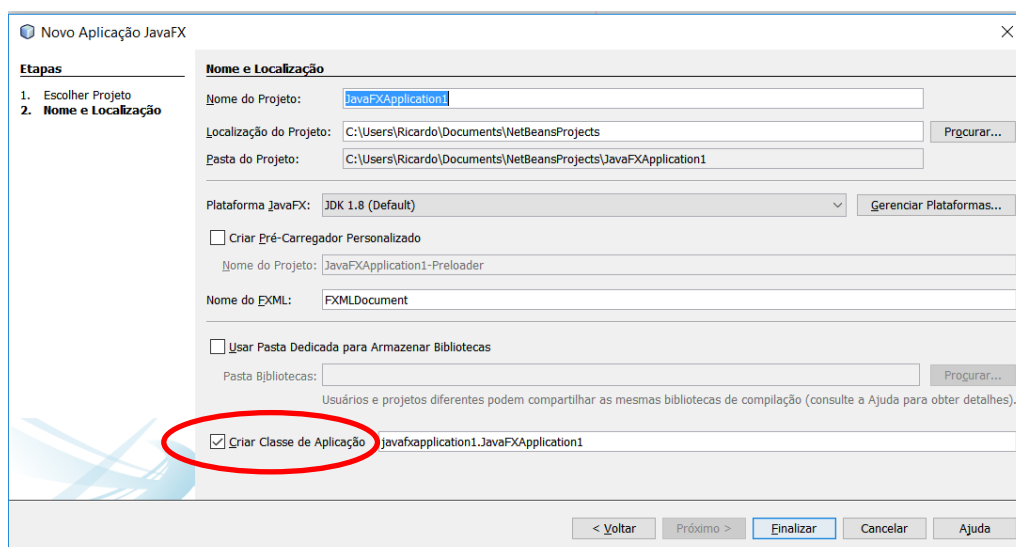


Selecione a **Categoria JavaFX** e em Projetos, selecione **Aplicação FXML** do JavaFX. Após clique em Próximo.

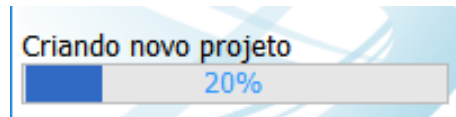
Como informado anteriormente FXML irá unir XML, HTML e CSS na programação em Java.



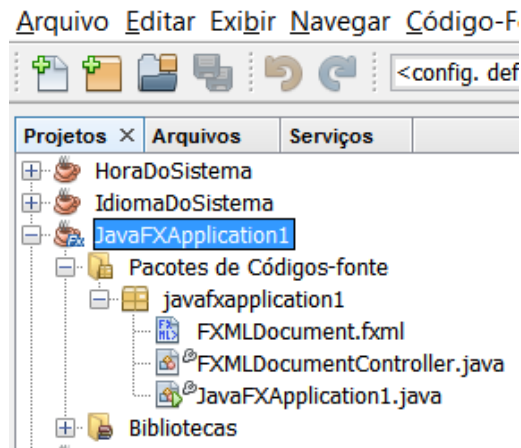
Nomeie o Projeto como OlaMundo3JavaFX. Após clique em Finalizar.



Aguarde o processo de criação do Pacote ser concluído.



Desta vez, ao invés de criar um arquivo, o Netbeans criou um Pacote com três arquivos.



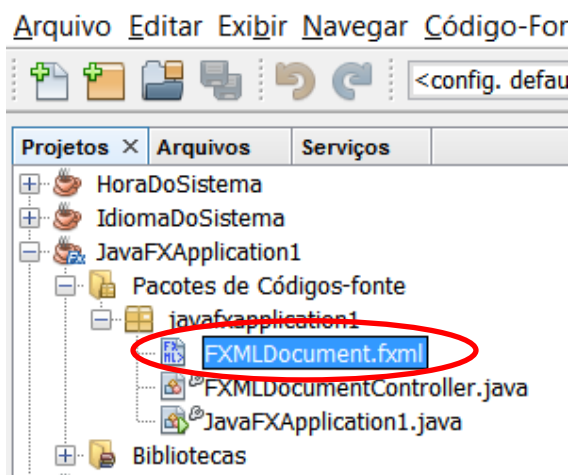
Observe que já foram importados diversos Pacotes externos, ou seja, Bibliotecas externas.

```
6 package javafxapplication1;
7
8 import javafx.application.Application;
9 import javafx.fxml.FXMLLoader;
10 import javafx.scene.Parent;
11 import javafx.scene.Scene;
12 import javafx.stage.Stage;
13
```

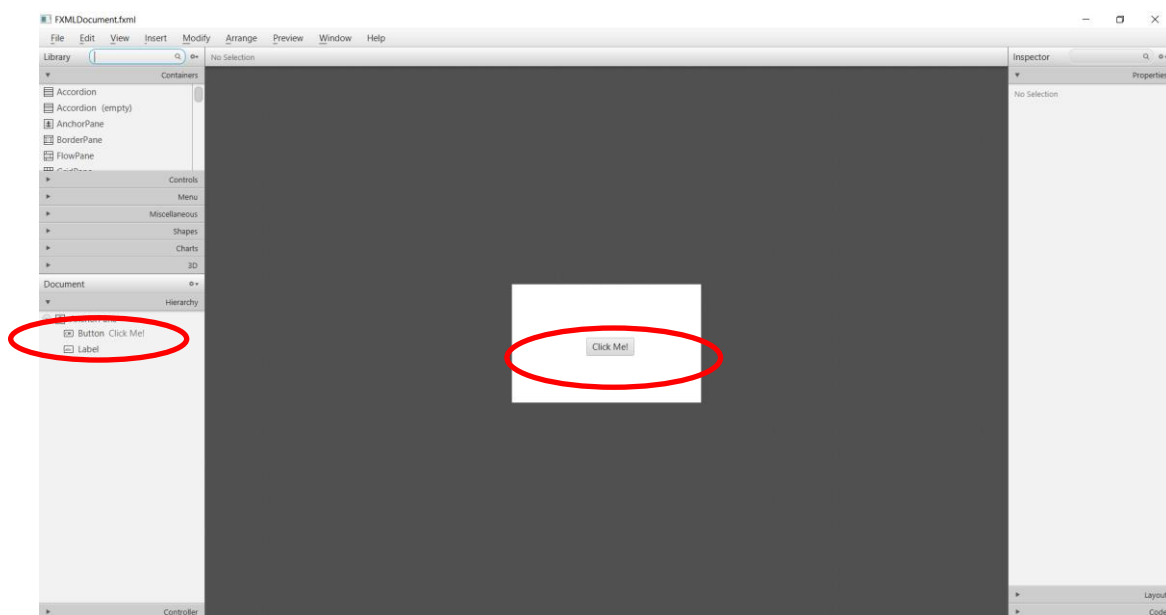
Utiliza “extends” para criar uma Herança, um Application, isso é herda tudo que a Aplicação possui.

```
6 package javafxapplication1;
7
8 import javafx.application.Application;
9 import javafx.fxml.FXMLLoader;
10 import javafx.scene.Parent;
11 import javafx.scene.Scene;
12 import javafx.stage.Stage;
13
14 /**
15  *
16  * @author Ricardo
17  */
18 public class JavaFXApplication1 extends Application {
19
```

Dando um duplo clique sobre FXMLElement.fxml irá abrir um programa que nós já instalamos na primeira aula, o Scenebuilder.

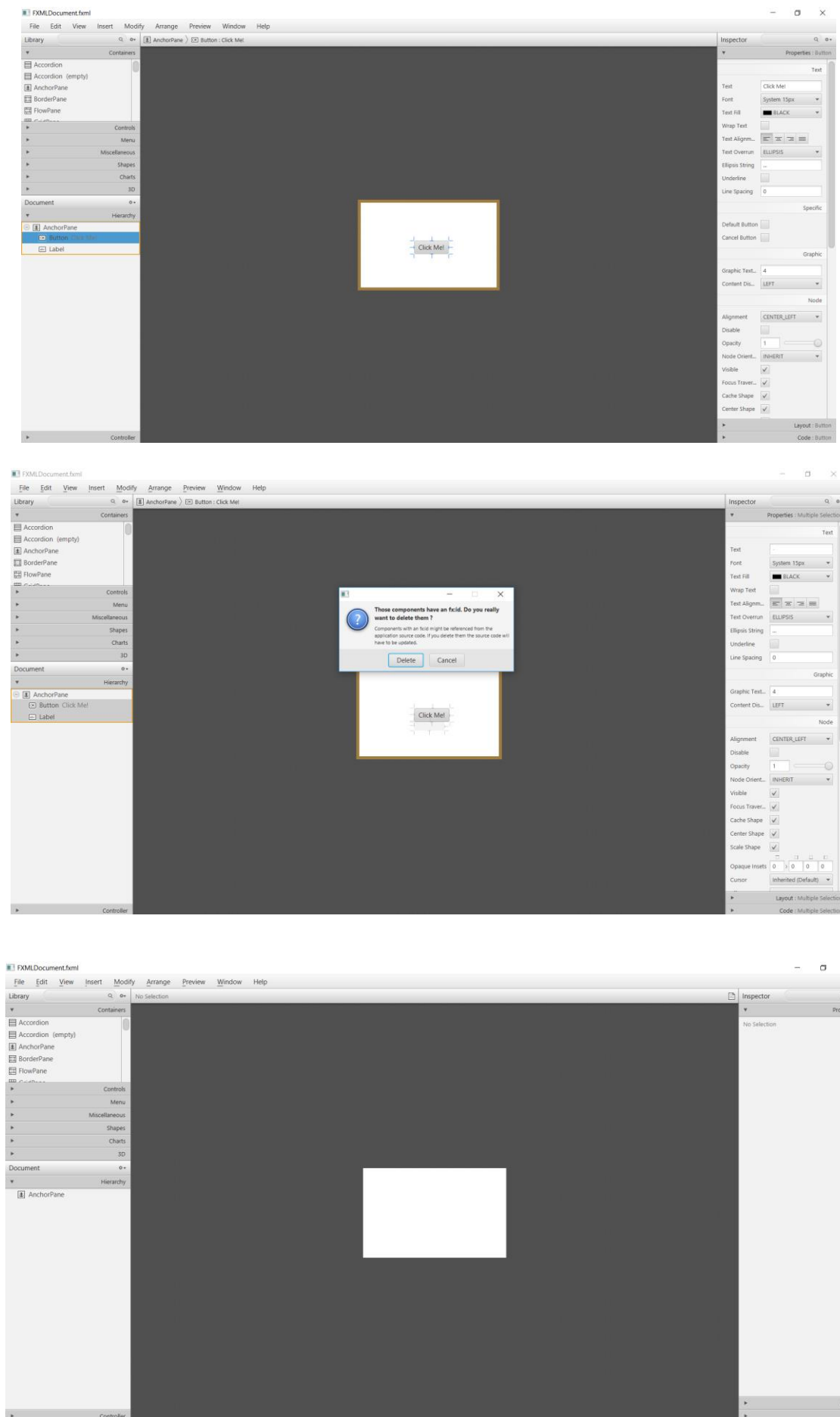


O Scenebuilder substitui a interface gráfica que vimos do Swing; funcionando como um ambiente gráfico externo.



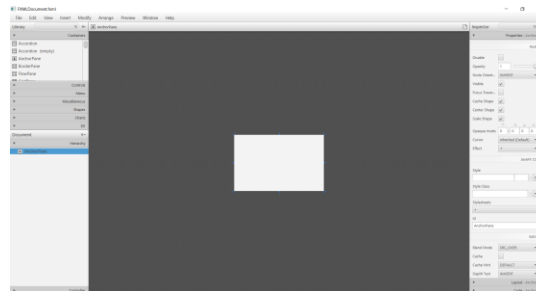
O FXMLDocument já vem com um Botão e um Label.

Para não ficar muito fácil e, assim, podemos aprender mais da aplicação, vamos excluir estes itens.

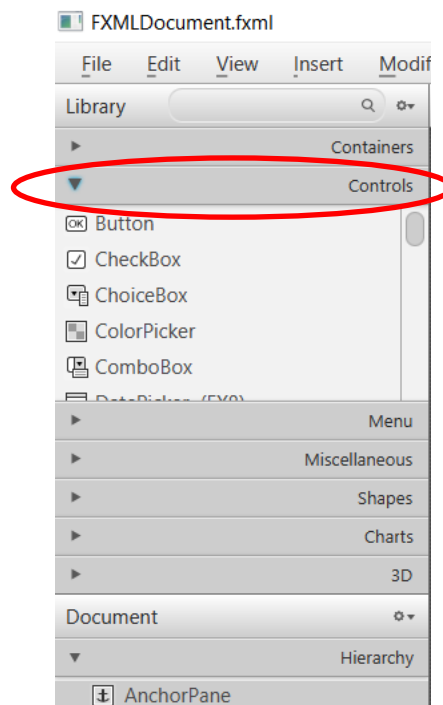


A tela que chamamos de JFrame quando utilizamos Swing, aqui é chamada de AnchorPane ou Painei de Ancoragem.

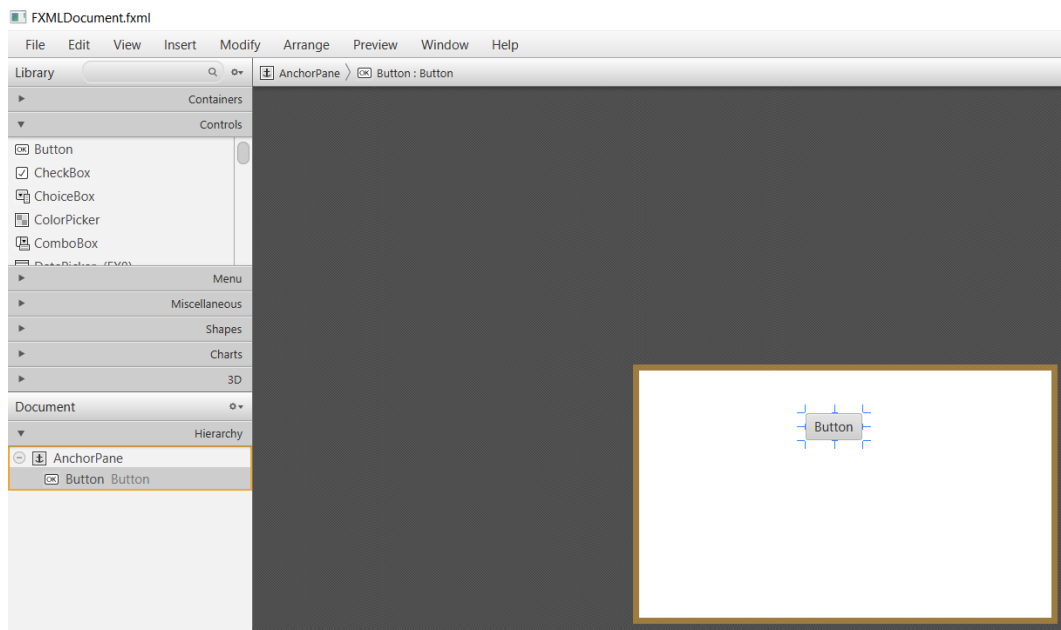
Não podemos chamar este Painei de Janela, pois, como dito anteriormente, ele irá servir para Celulares, Navegadores, Aplicações, Vídeo Games, TVs e etc. Por tanto, não podemos chamar de Janela, de maneira correta devemos chama-lo de Painei de Ancoragem.



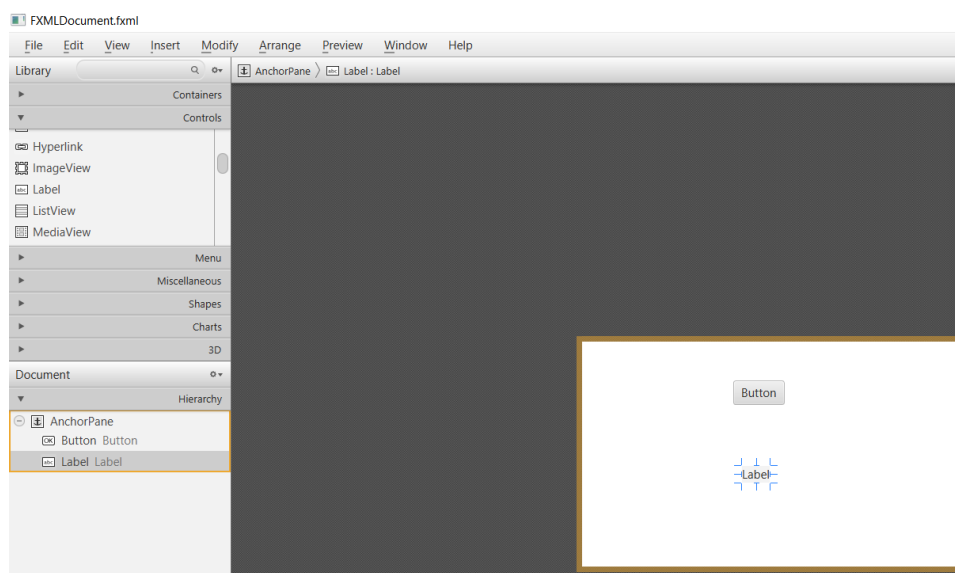
No Painei localizado ao lado esquerdo da tela, clique em Controls para expandir e mostrar as opções.



Clique e arraste, mantendo o botão direito do mouse pressionado, para adicionar e posicionar o Botão (Button).

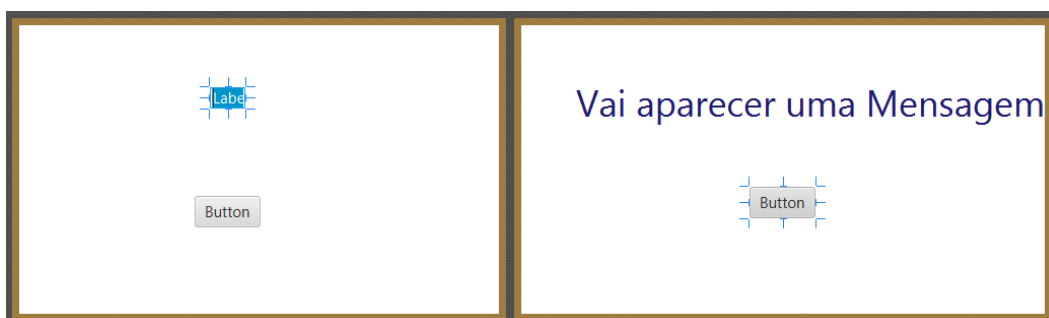


Utilize a Barra de Rolagem para descer entre as opções e adicione um Label.



Assim como anteriormente, é possível mover os itens para reposicioná-los e alterar o texto dando um duplo clique.

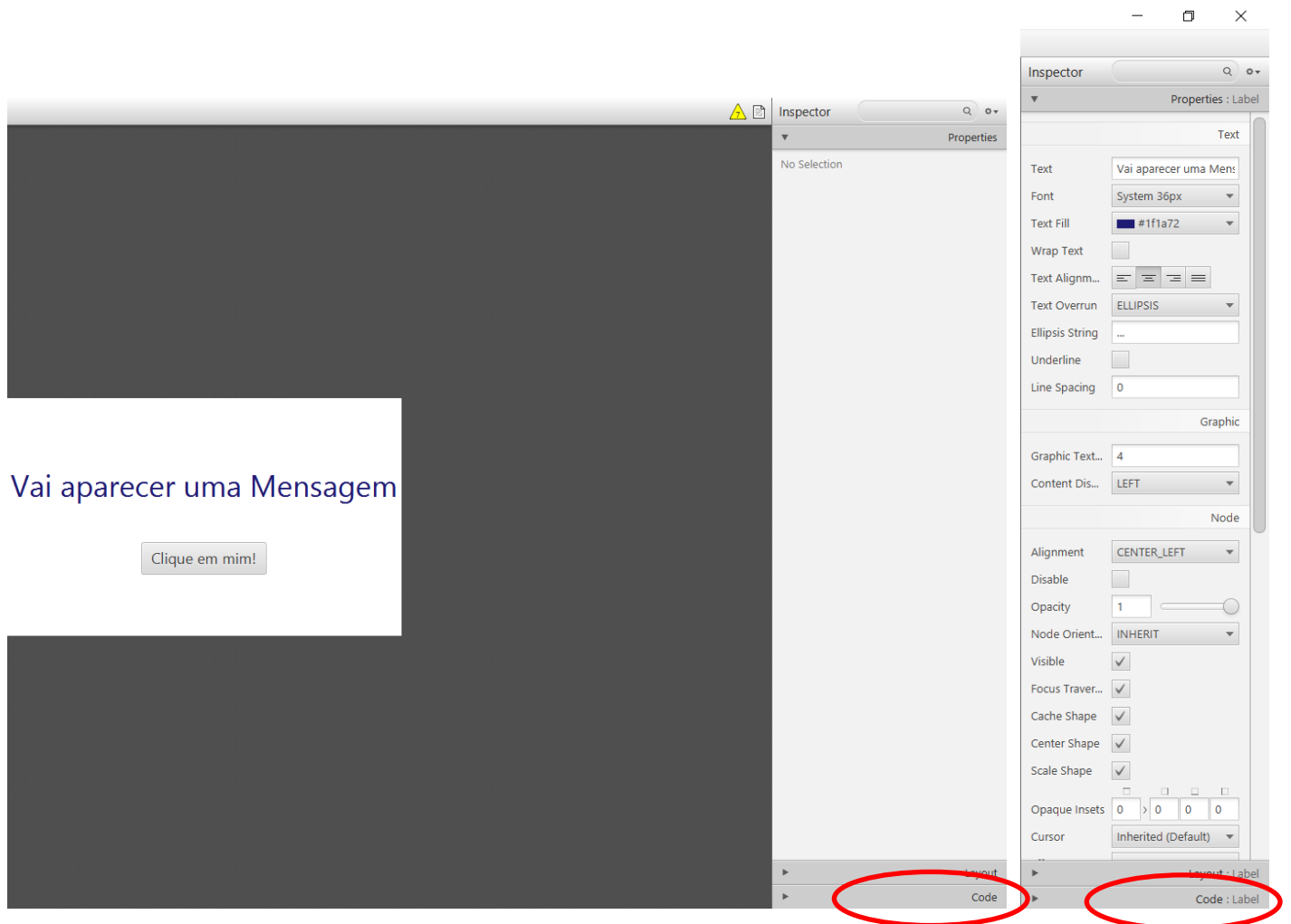
Digite “Vai aparecer uma Mensagem” e formate a fonte, como cor, tamanho, etc.



Da mesma forma, reconfigure o Botão.

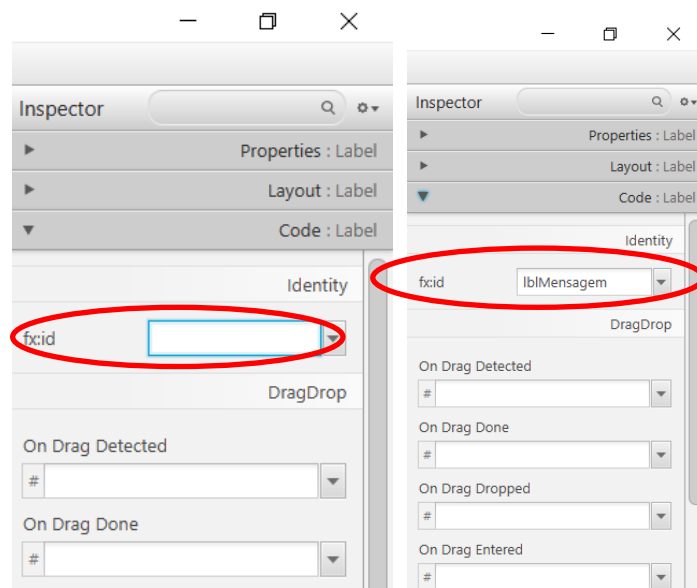


Para modificar o ID, selecione o Label e no painel localizado ao lado direito da tela clique em Code: Label para expandir o Painel.

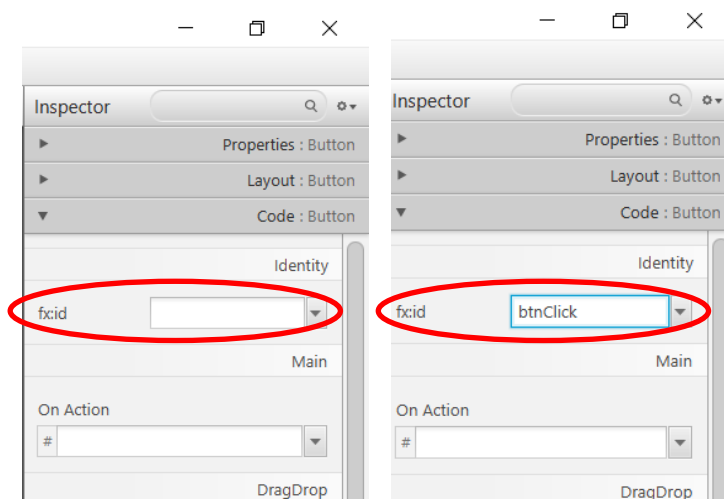




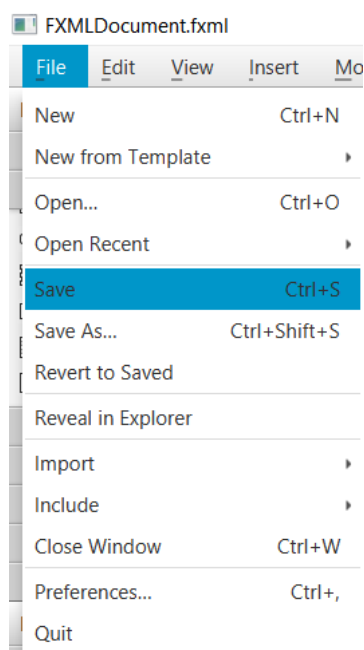
Na caixa de texto “fx:id”, digite “lblMensagem”.



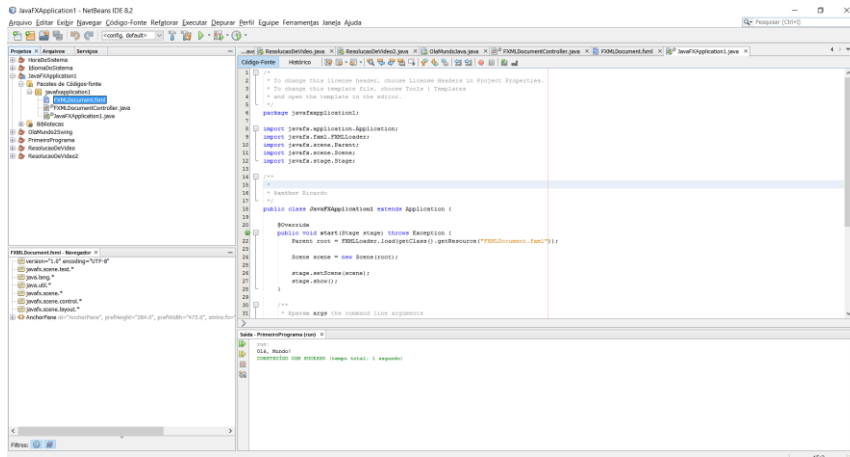
Faça a mesma coisa com o Botão, dando o ID de btnClick.



Após identificar os IDs, salve o documento.

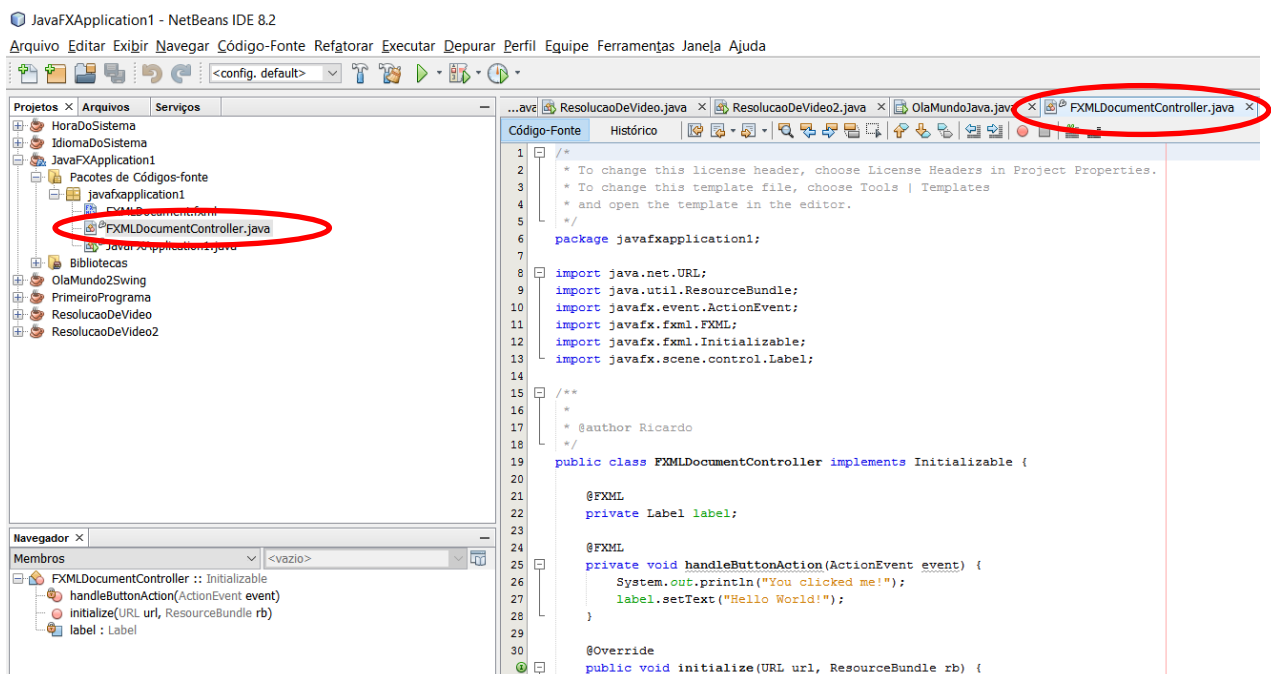


Retorne ao Netbeans.

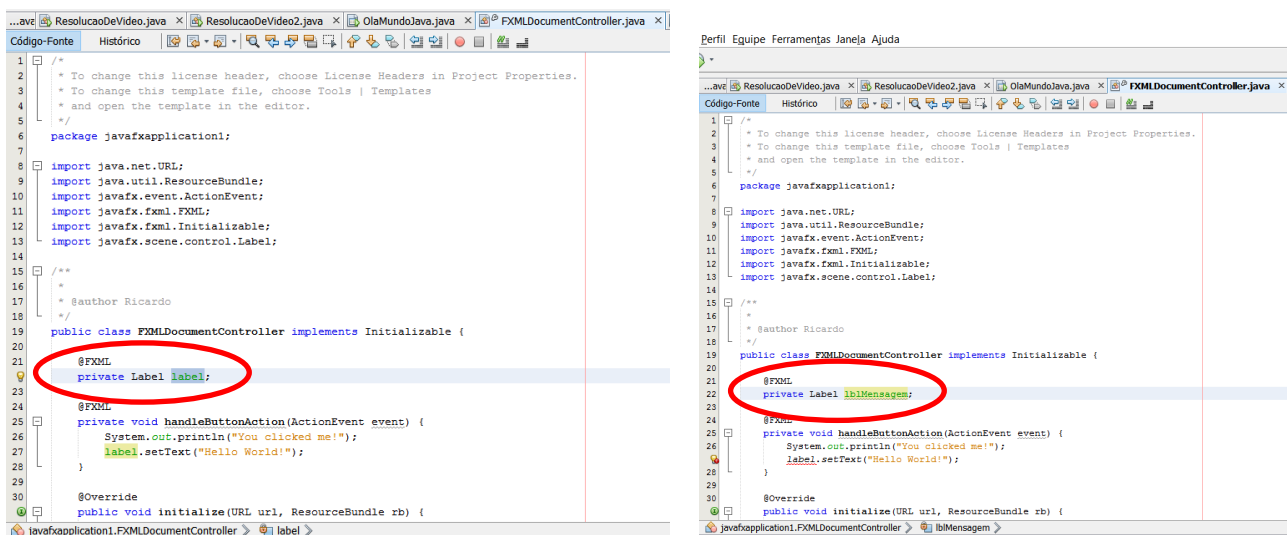


Clique sobre a guia FXMLDocumentController.java.

O “@FXML” é para identificar os controles.

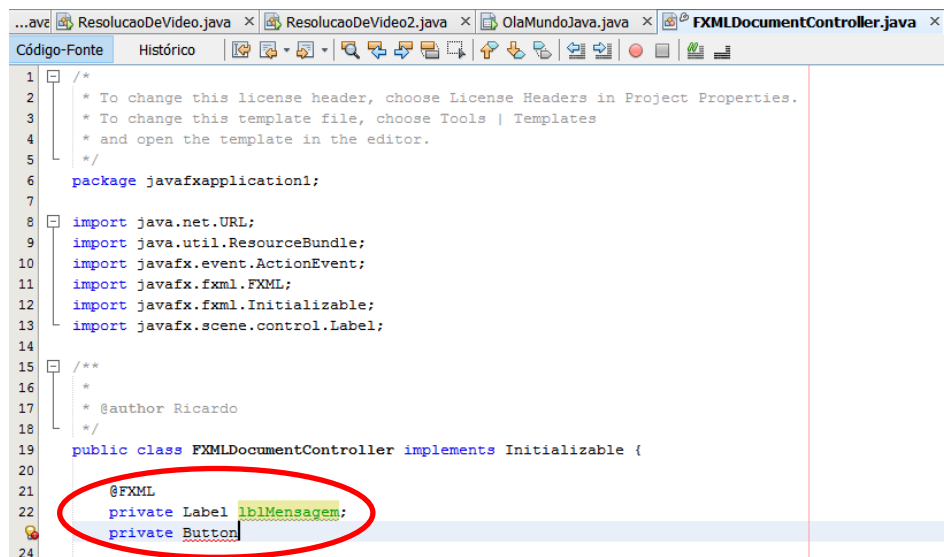


Onde está escrito “private Label label”, altere “label” para o nome do nosso controle, “lblMensagem”.



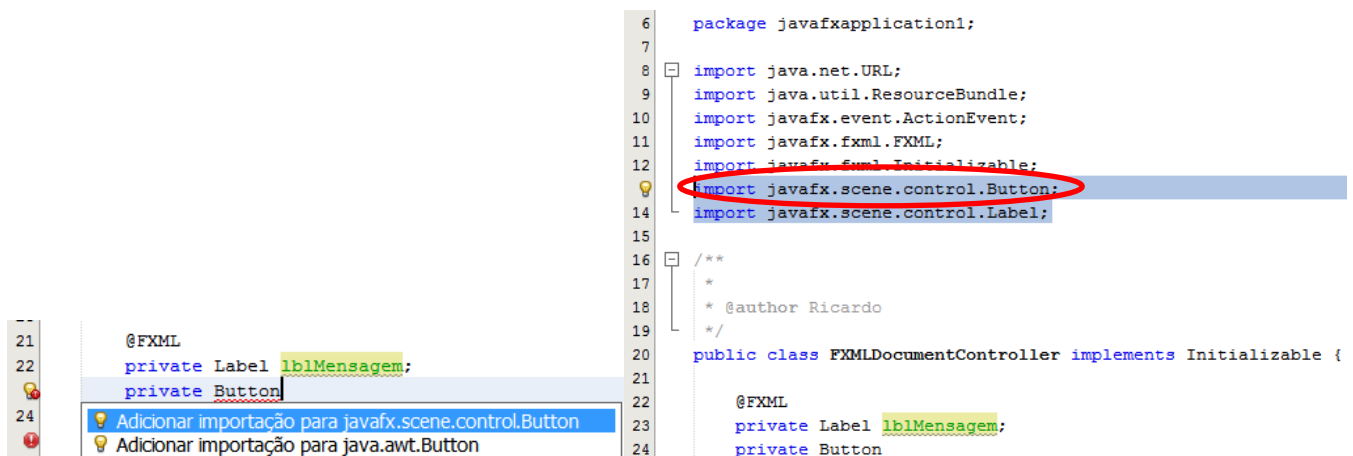
Digite “private Button”.

Aparecerá a lâmpada ao lado.



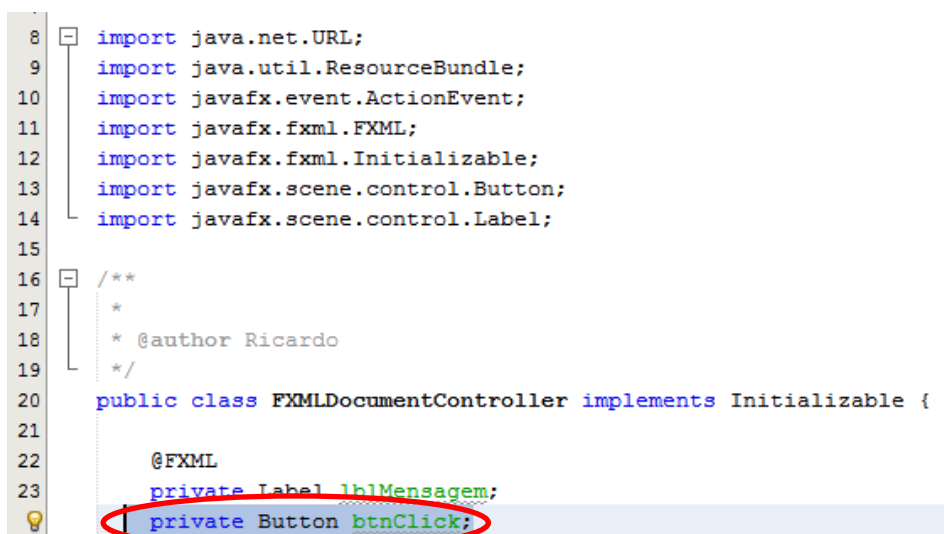
```
1  /**
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package javafxapplication1;
7
8   import java.net.URL;
9   import java.util.ResourceBundle;
10  import javafx.event.ActionEvent;
11  import javafx.fxml.FXML;
12  import javafx.fxml.Initializable;
13  import javafx.scene.control.Label;
14
15  /**
16   *
17   * @author Ricardo
18   */
19  public class FXMLDocumentController implements Initializable {
20
21      @FXML
22      private Label lblMensagem;
23      private Button
24  }
```

Observe que aparece a lâmpada informativa. Há a necessidade de importar a biblioteca; para isso, clique na lâmpada e após clique em “Adicionar importação para javafx.scene.control.Button”.



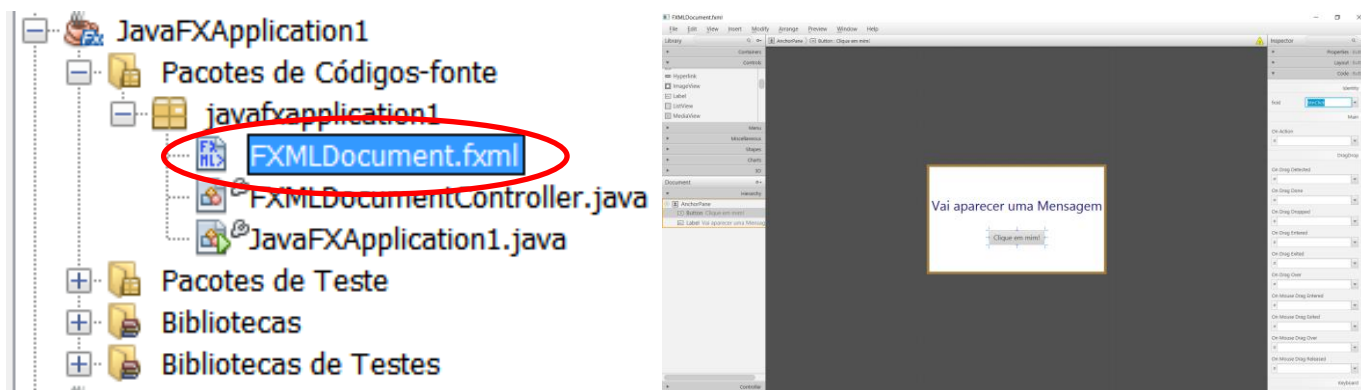
```
6   package javafxapplication1;
7
8   import java.net.URL;
9   import java.util.ResourceBundle;
10  import javafx.event.ActionEvent;
11  import javafx.fxml.FXML;
12  import javafx.fxml.Initializable;
13  import javafx.scene.control.Button;
14  import javafx.scene.control.Label;
15
16  /**
17   *
18   * @author Ricardo
19   */
20  public class FXMLDocumentController implements Initializable {
21
22      @FXML
23      private Label lblMensagem;
24      private Button
```

Digite o nome do Botão, btnClick.

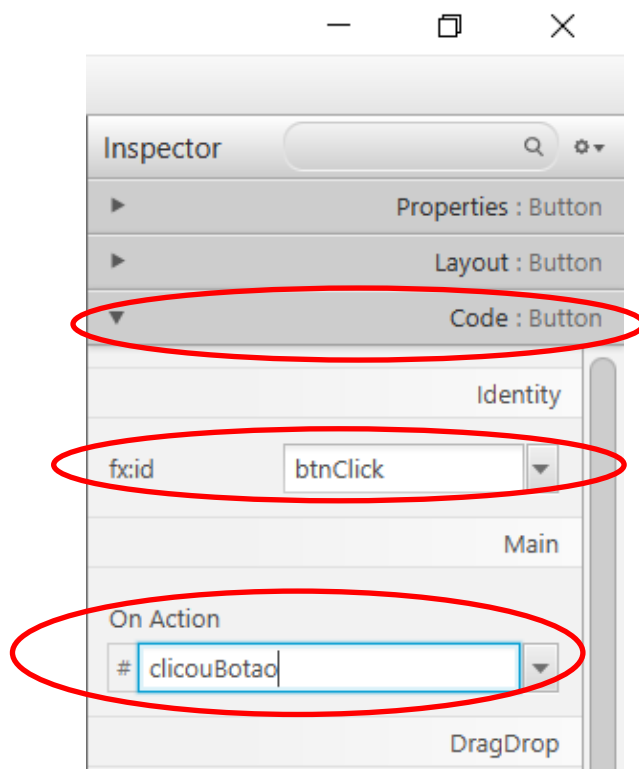


```
8   import java.net.URL;
9   import java.util.ResourceBundle;
10  import javafx.event.ActionEvent;
11  import javafx.fxml.FXML;
12  import javafx.fxml.Initializable;
13  import javafx.scene.control.Button;
14  import javafx.scene.control.Label;
15
16  /**
17   *
18   * @author Ricardo
19   */
20  public class FXMLDocumentController implements Initializable {
21
22      @FXML
23      private Label lblMensagem;
24      private Button btnClick;
25  }
```

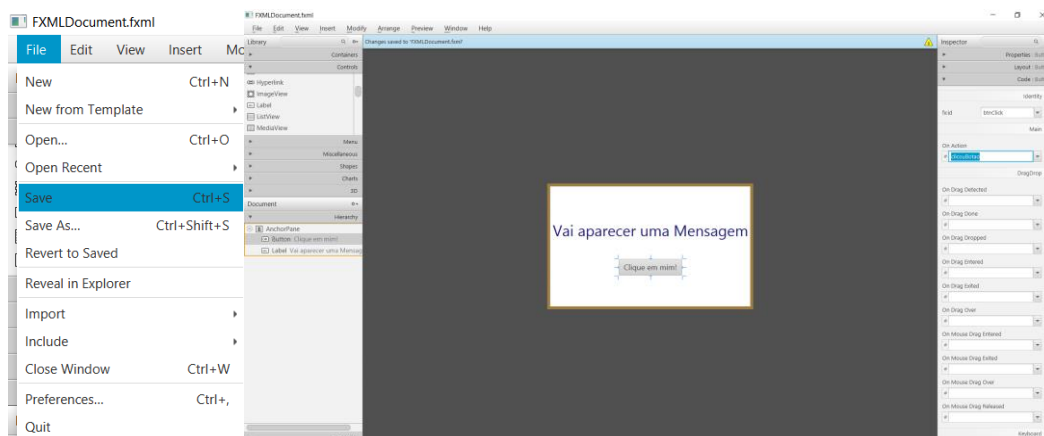
Retorne ao Scencebuilder dando um duplo clique sobre FXMLElement.fxmll, localizado no painel esquerdo.



Em “On Action”, localizado no painel ao lado direito, digite “cliqueuBotao”. Ou seja, um Evento “On Action” que é o “cliqueuBotao”.



Salve as alterações.



Retorne ao Netbeans para programar o Evento.

Vamos definir o Evento apagando “handleButtonAction” e digitando o nome da ação que acabamos de nomear no Scencebuilder, “cliqueuBotao”.

```
26 @FXML
27 private void handleButtonAction(ActionEvent event) {
28     System.out.println("You clicked me!");
29     label.setText("Hello World!");
30 }
```

```
26 @FXML
27 private void cliqueuBotao(ActionEvent event) {
28     System.out.println("You clicked me!");
29     label.setText("Hello World!");
30 }
```

Apague os comandos que estão no interior.

```
26 @FXML
27 private void cliqueuBotao(ActionEvent event) {
28     System.out.println("You clicked me!");
29     label.setText("Hello World!");
30 }
```

```
26 @FXML
27 private void cliqueuBotao(ActionEvent event) {
28
29 }
```

Digite o mesmo comando do Swing:

“lblMensagem.” e aguarde a ajuda do Netbens. Digite “setT” e pressione a tecla enter.

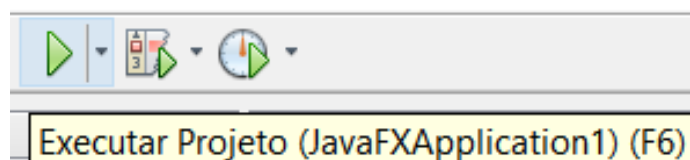
```
11 import javafx.fxml;
12 import javafx.scene.control.Label;
13 import javafx.scene.control.Button;
14 import javafx.scene.layout.HBox;
15
16 public final void setText(String value)
17     Sets the value of the property text.
18     Property description:
19     The text to display in the label. The text may be null.
20
21 public class FXML
22 {
23     @FXML
24     private Label
25     private Button
26
27     @FXML
28     private void
29     lblMensagem.setText
30 }
31
32 @Override
33
34 javaApplication1.FXMLDocu
35
36 run:
37 Olá, Mundo!
38 CONSTRUÍDO COM SUCESSO
39
40 Membros da Instância; Pressione 'Ctrl+SPACE' Novamente para Todos os Itens
```

```
26 @FXML
27 private void cliqueuBotao(ActionEvent event) {
28     lblMensagem.setText(value);
29 }
```

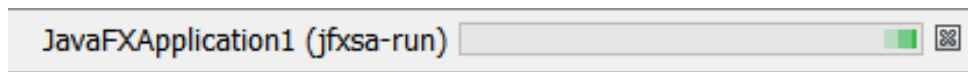
Configure o texto para “Olá, Mundo!”.

```
26 @FXML
27 private void cliqueuBotao(ActionEvent event) {
28     lblMensagem.setText("Olá, Mundo!");
29 }
```

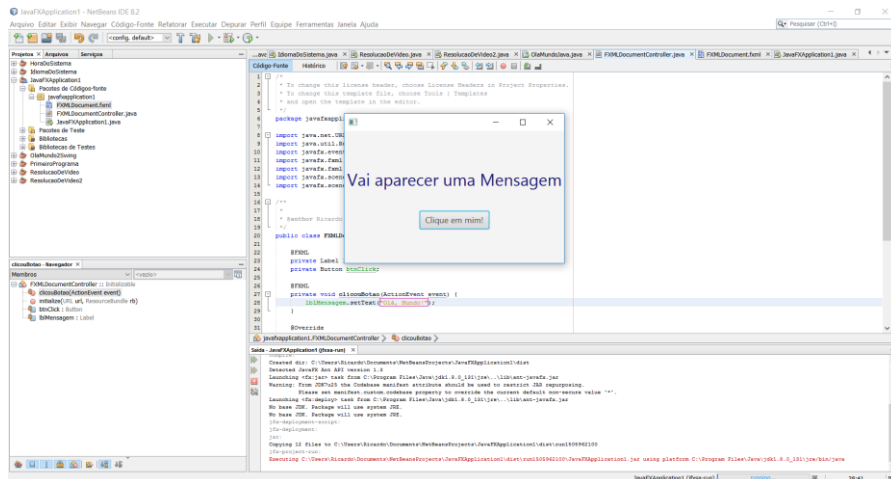
Clique no botão Executar Projeto, localizado na Barra de Tarefas.



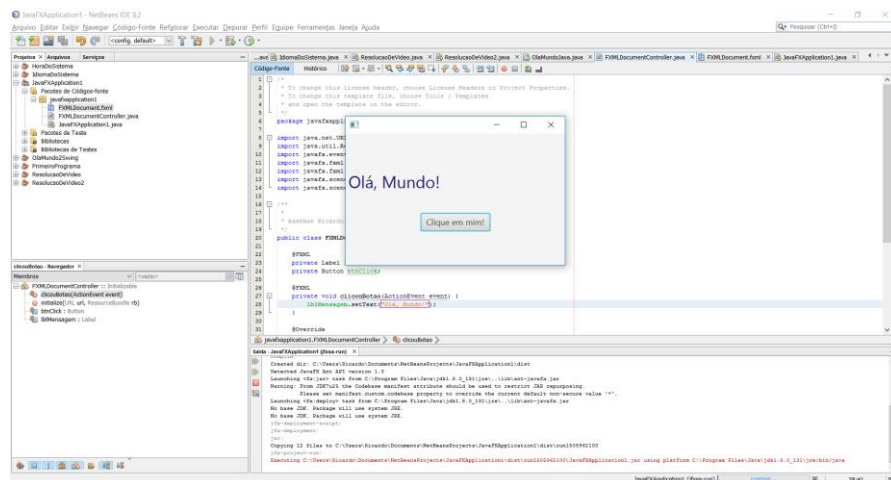
Aguarde o processamento.



Será exibida a janela.



Clique no botão “Clique em mim!” e será exibida a mensagem “Olá, Mundo!”.



Compreendendo um pouco mais do código.

Tudo inicia com algumas importação como a “javafx.fxml”.

```
import javafx.fxml.FXML;
```

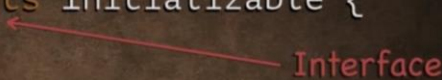
Depois temos a Classe utilizando Controlador que implementa Initializable.

```
import javafx.fxml.FXML;  
public class FXMLDocumentController  
    implements Initializable {
```

Neste ponto, ao invés de utilizarmos “extends” se utiliza “implements”, que é um outro conceito de Programação Orientada a Objeto que veremos mais a frente.

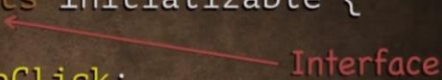
Sempre que visualizar a palavra “implements” saberá que está fazendo uma interface.

```
import javafx.fxml.FXML;  
public class FXMLDocumentController  
    implements Initializable {
```



Em seguida temos “@FXML” e dois comandos para adição dos controles. Um para o Botão e outro para o Label.

```
import javafx.fxml.FXML;  
public class FXMLDocumentController  
    implements Initializable {  
    @FXML  
    private Button btnClick;  
    private Label lblMensagem;
```





Após, a criação do Código do Evento. Este código é semelhante ao do Swing, mudando apenas a sintaxe.

```
import javafx.fxml.FXML;

public class FXMLDocumentController
    implements Initializable {

    @FXML
    private Button btnClick;
    private Label lblMensagem;

    @FXML
    private void clicouBotao(...) {

    }

}
```

Interface

E, igualmente em Swing, é adicionado o comando “lblMensagem.setText(“Olá, Mundo!”).”.

Note que é muito semelhante o código gerado em Swing e em JavaFX.

```
public class TelaSwing extends javax.swing.JFrame {
    private javax.swing.JButton btnClick;
    private javax.swing.JLabel lblMensagem;
    private void btnClickActionPerformed(...) {
        lblMensagem.setText("Olá, Mundo!");
    }
}

import javafx.fxml.FXML;
public class FXMLDocumentController
    implements Initializable {

    @FXML
    private Button btnClick;
    private Label lblMensagem;

    @FXML
    private void clicouBotao(...) {
        lblMensagem.setText("Olá, Mundo!");
    }

}
```

Swing

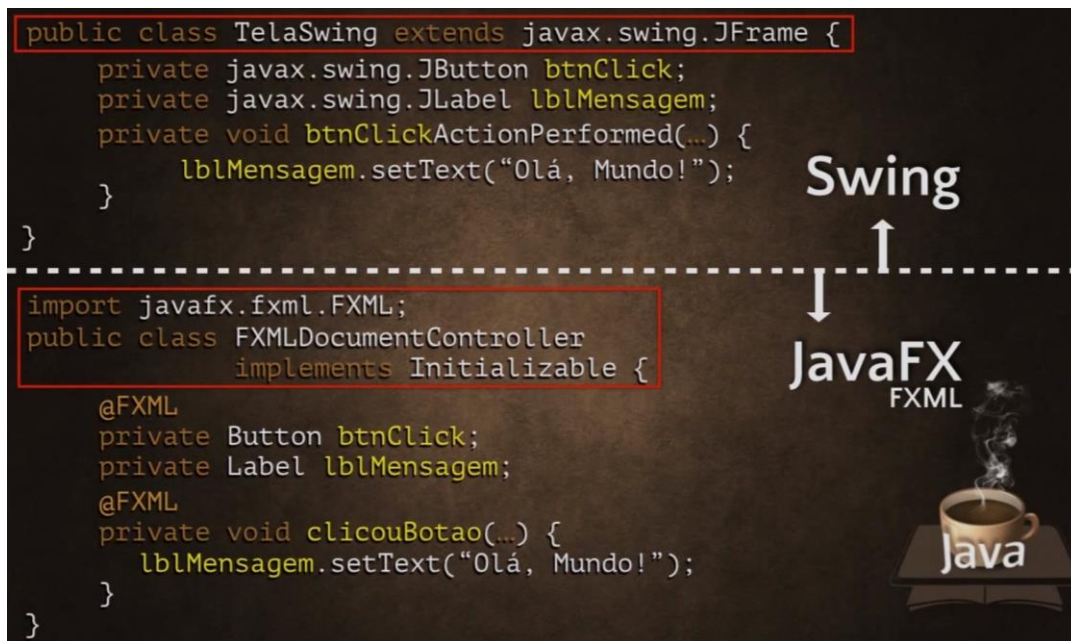
↓

JavaFX  
FXML

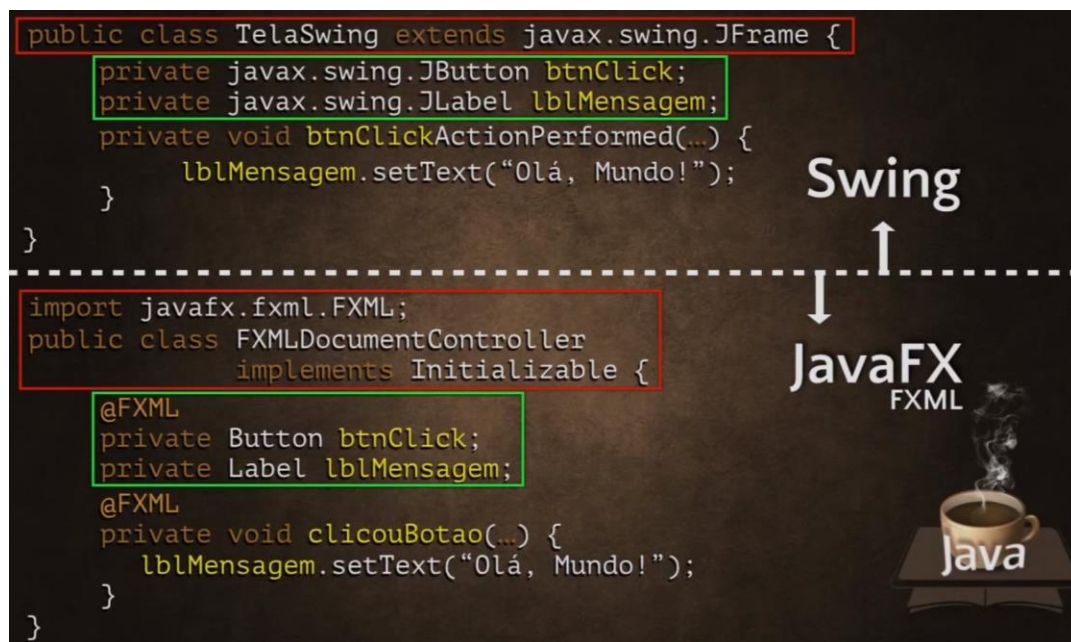
Java



No Cabeçalho, Swing usa a Herança de JFrame e o JavaFX usa a Importação da Biblioteca FXML.



Em seguida há a declaração dos Controles, uma para o Botão e outra para o Label.



E por fim, temos o Método do Botão.

