

# Construindo interfaces com JavaFx Scene Builder

...

ecompcjr.

# Apresentação

- Bruno Vogel

Consultor de Projetos da EcompJr



- Kayo Costa

Diretor de Recursos Humanos da EcompJr

Bolsista Fapesb de Iniciação Científica (GICE)



- Thatianne Carvalho

Assessora Administrativa e Financeira da EcompJr



# Antes de tudo...

Este curso foi projetado para quem já possui conhecimento em programação em java.

Momentos teóricos e atividades práticas.

Padrões de Arquitetura.

Mini desafio (projeto).

# Objetivo

Ao final desta oficina, pretende-se que vocês estejam aptos para criar suas interfaces para aplicações java através do javafx scene builder.

# Uma nova ferramenta

Programar uma interface

# Uma nova ferramenta


Programar uma interface 



**CARA**


**QUE LOUCURA**


# Uma nova ferramenta

Programar uma interface  Uma das piores ideias



# Uma nova ferramenta

Programar uma interface  Uma das piores ideias

Form JFrame  Form JFrame...

**bug bug bug**

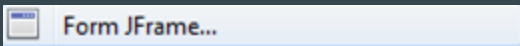


**bug bug bug**

# Uma nova ferramenta

Programar uma interface  Uma das piores ideias

Form JFrame



JavaFX 2





**Para NooOossa  
alegria!**

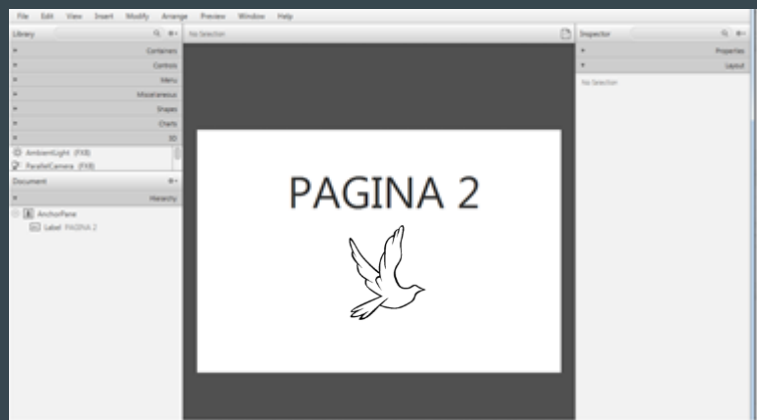
# JavaFX 2

Lançado pela Sun - Chris Oliver - Projeto F3

Perfil Comum

Manipulação gráfica por meio do Prisma

Integração com programas terceiros

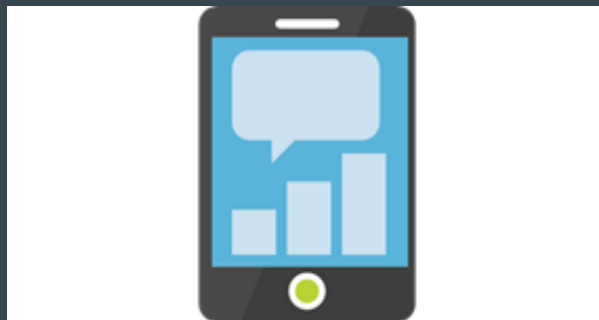


Animações



3D

Mobile



Exemplos de Utilizações do JavaFx

# JavaFX 2 X Swing

Swing  AWT

JavaFX  Livre

Semelhanças entre as bibliotecas

# Instalação

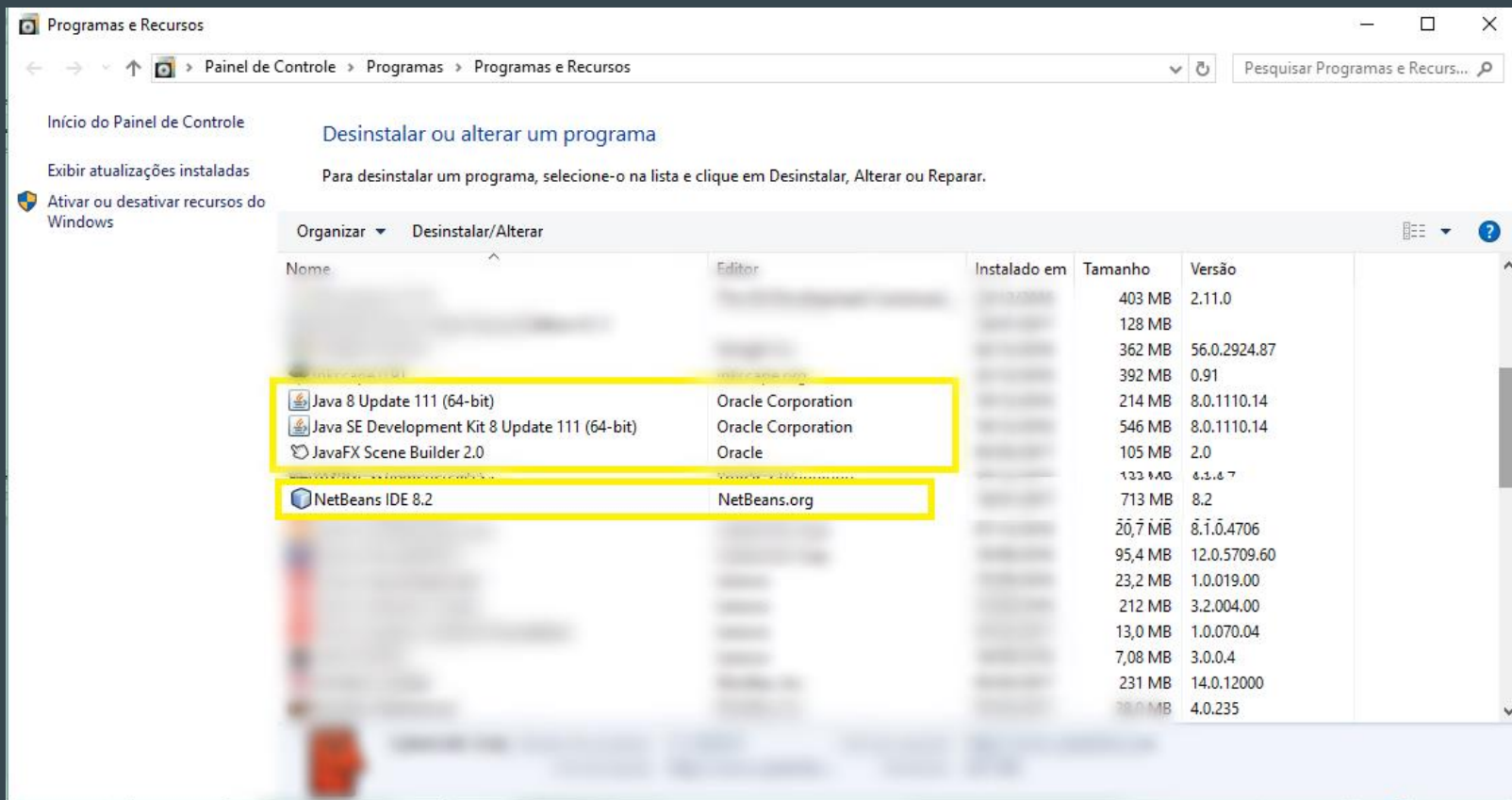
Netbeans

JDK 8

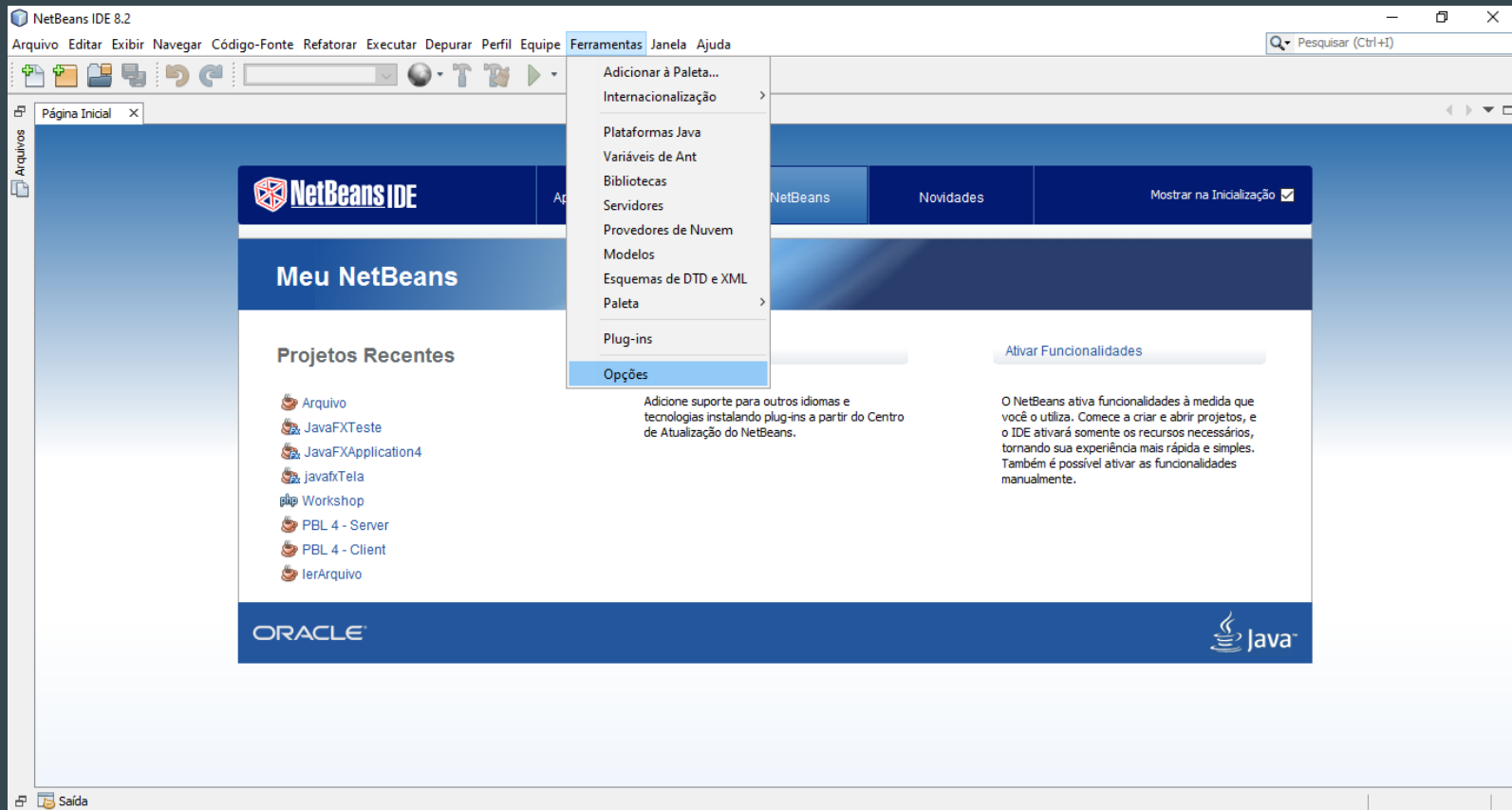
JavaFX Scene Builder 2.0

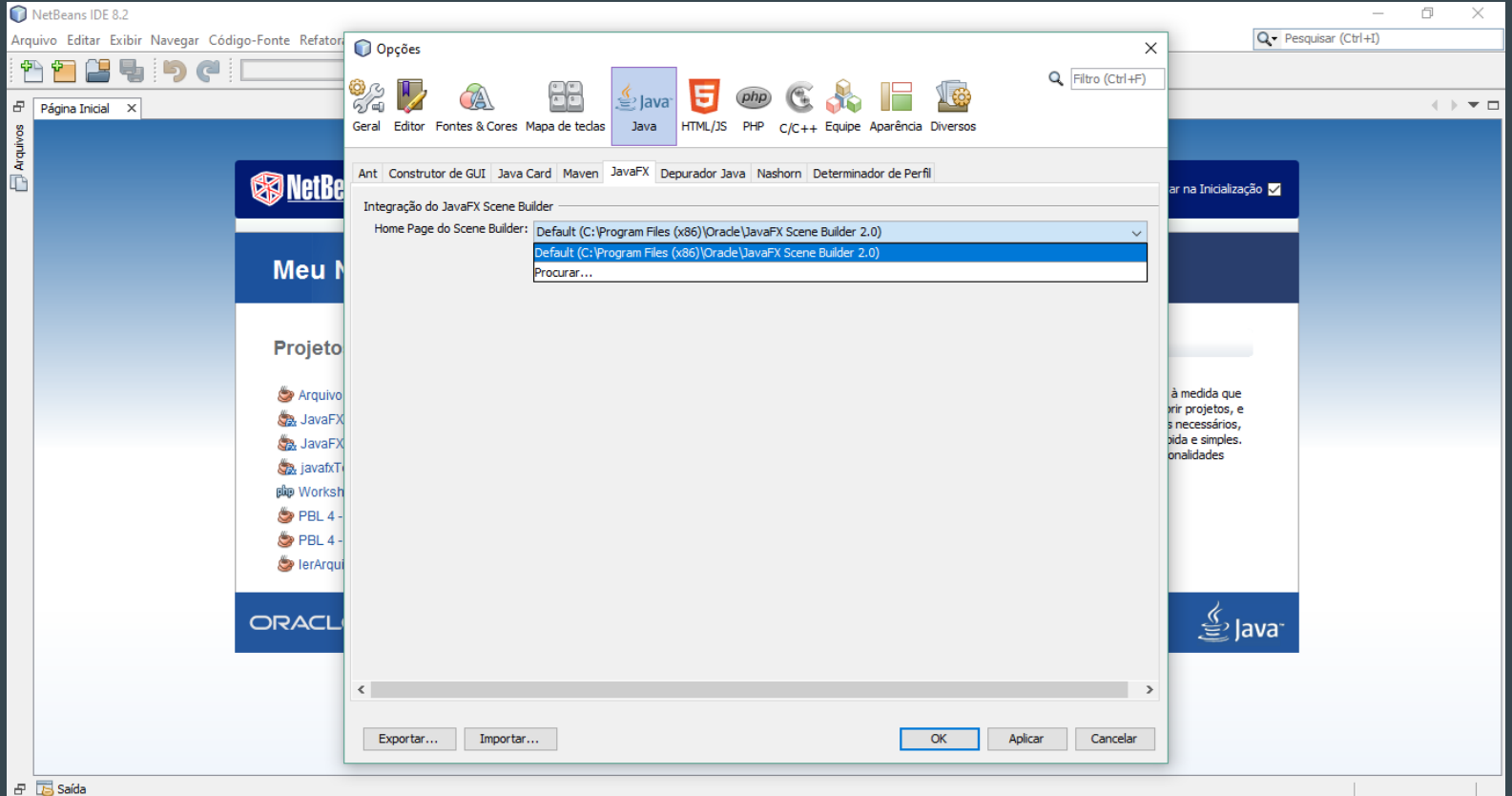


# Recursos Instalados



# Configurando







Meu N

Projeto

Arquivo

JavaFX

JavaFX

javafxT

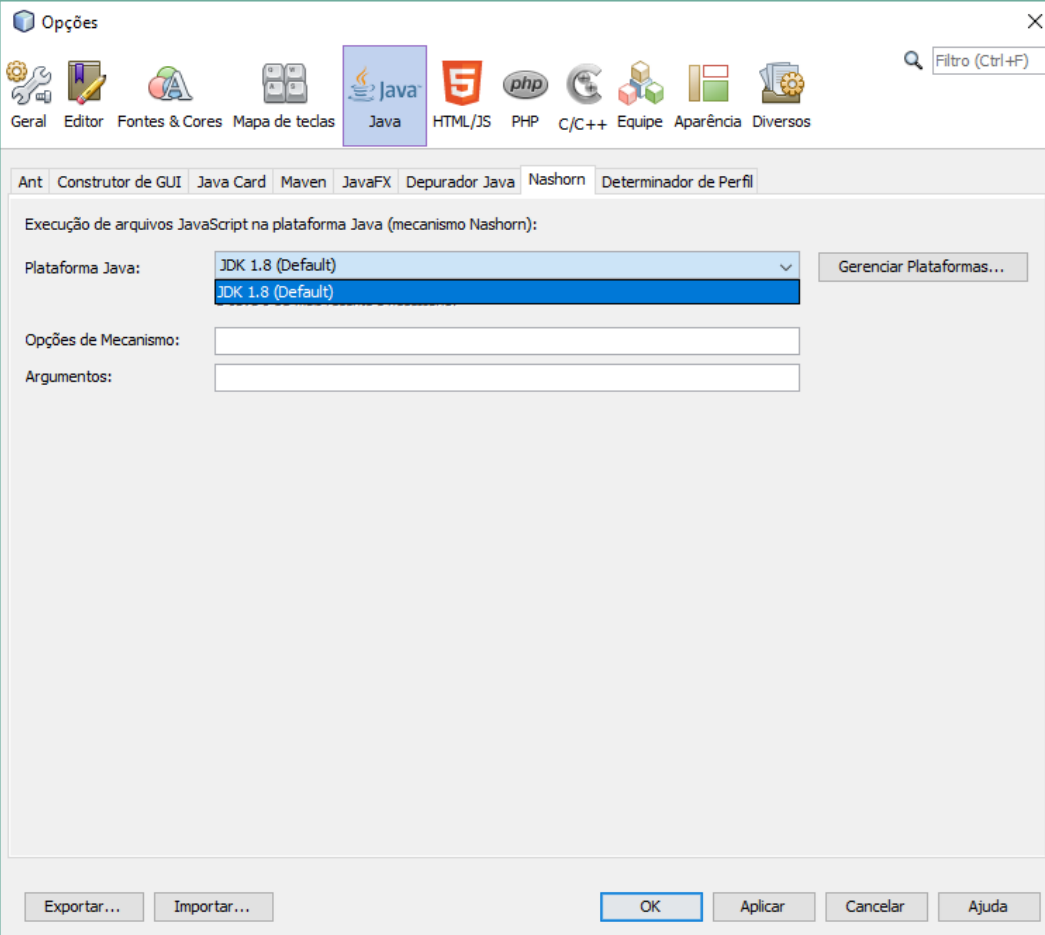
php Worksh

PBL 4 -

PBL 4 -

lerArqui

ORACLE



# Instalação No Linux Debian

Download do JDK 8u121 with NetBeans 8.2 (extensão .sh)

Download JavaFX Scene Builder 2.0 (extensão .deb)

No terminal:

```
chmod 744 jdk-8u121-nb-8_2-linux-x64.sh
```

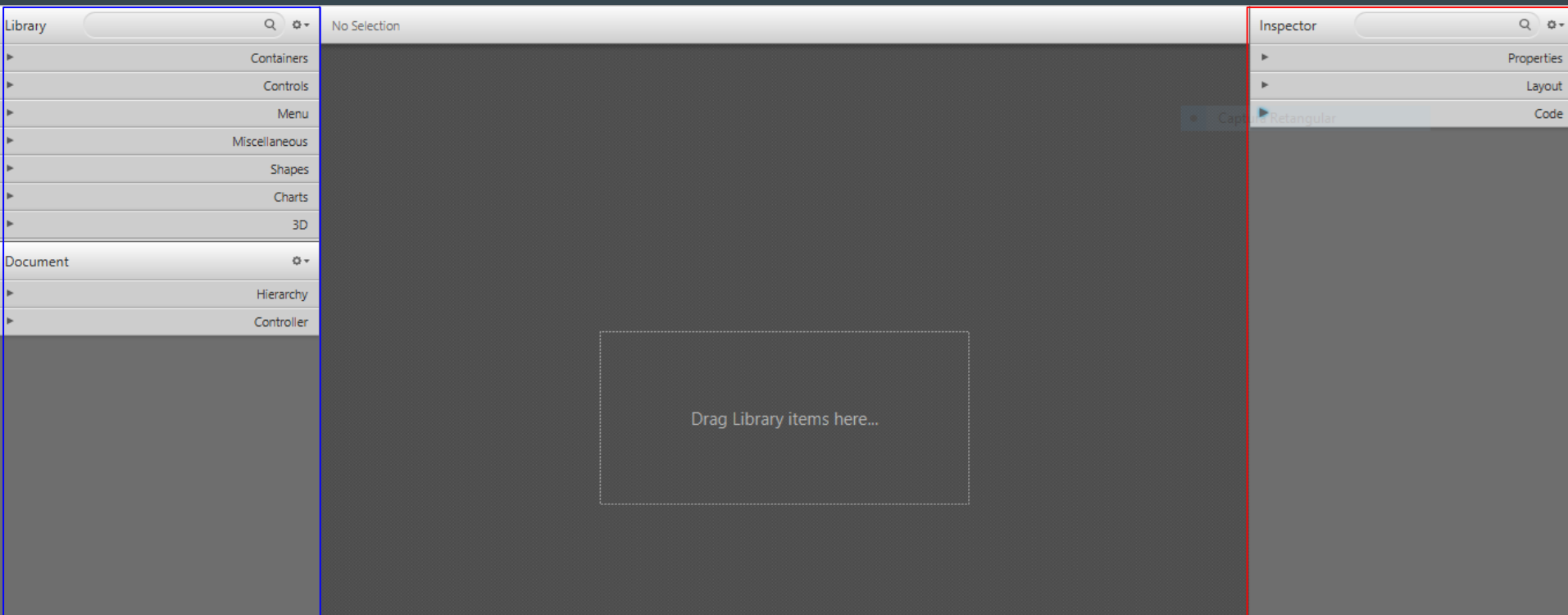
```
./jdk-8u121-nb-8_2-linux-x64.sh
```

Dois cliques no javaFX Scene Builder, instalar....

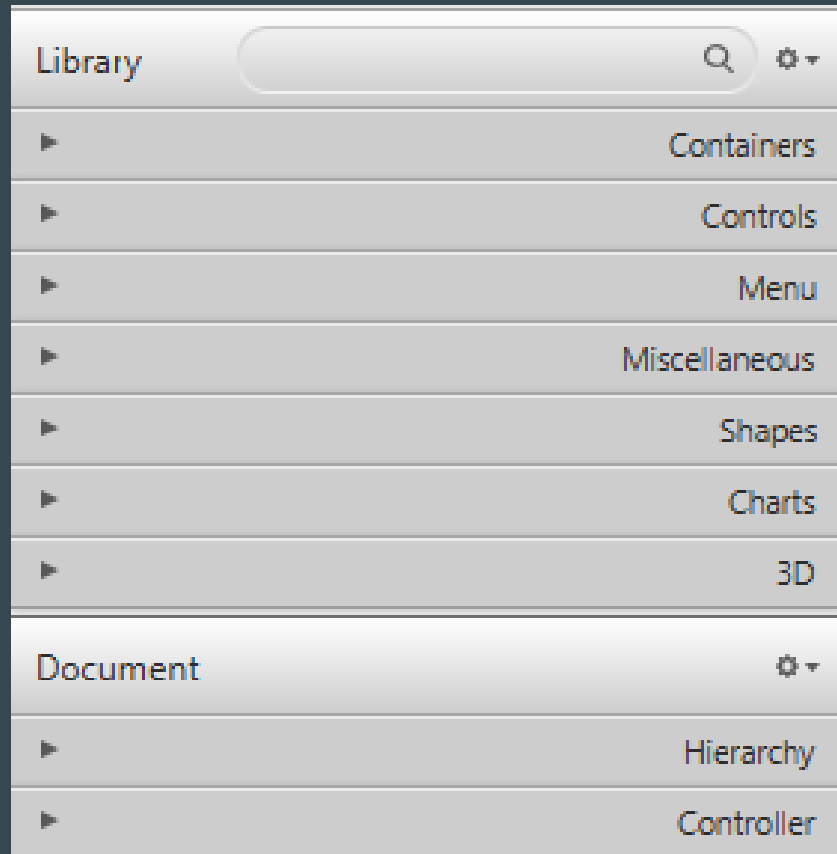
**PRONTO!!!**

# JavaFx Scene Builder

# Workspace

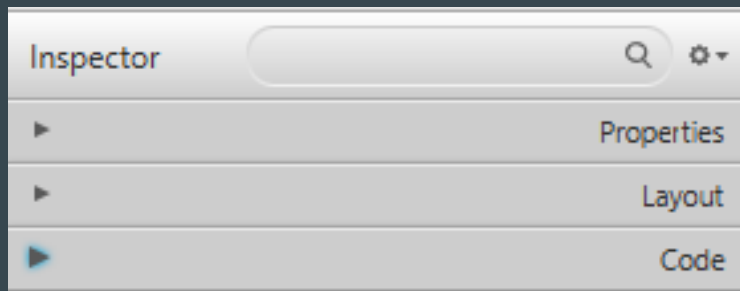


# Menu (Library e Document)



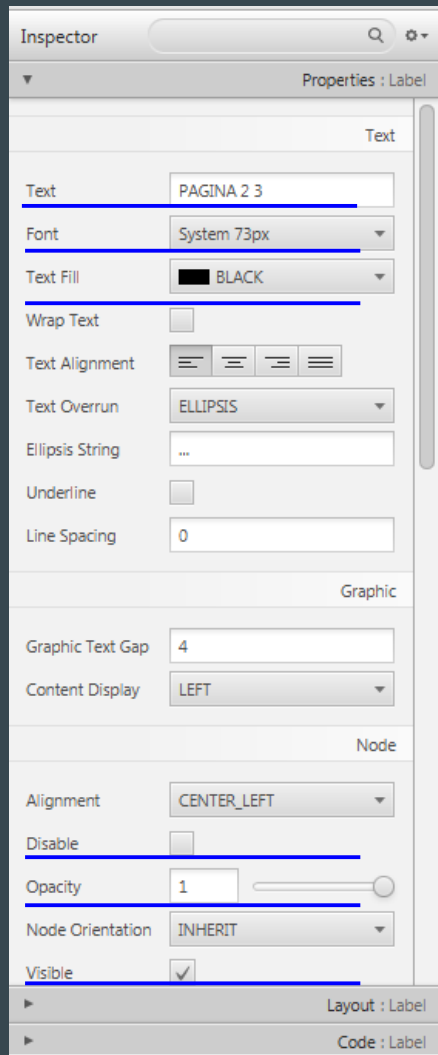


# Menu (Properties, Layout e Code)



# Menu

## Properties: Label



# Menu

## Layout: Label

Layout : Label

Min Width	USE_COMPUTED_SIZE
Min Height	USE_COMPUTED_SIZE
Pref Width	215
Pref Height	168
Max Width	USE_COMPUTED_SIZE
Max Height	USE_COMPUTED_SIZE
Width	302
Height	286

Position

Layout X	44
Layout Y	59

Transforms

Rotate 0

	X	Y	Z
Rotation Axis	0	0	1
Scale X	1		
Scale Y	1		
Scale Z	1		
Translate X	0		

# Menu

Code: Label

▼ Code : Label

Identity

fxid

DragDrop

On Drag Detected

#

On Drag Done

#

On Drag Dropped

#

On Drag Entered

#

On Drag Exited

#

On Drag Over

#

On Mouse Drag Entered

#

On Mouse Drag Exited

#

On Mouse Drag Over

#

# Primeiro projeto

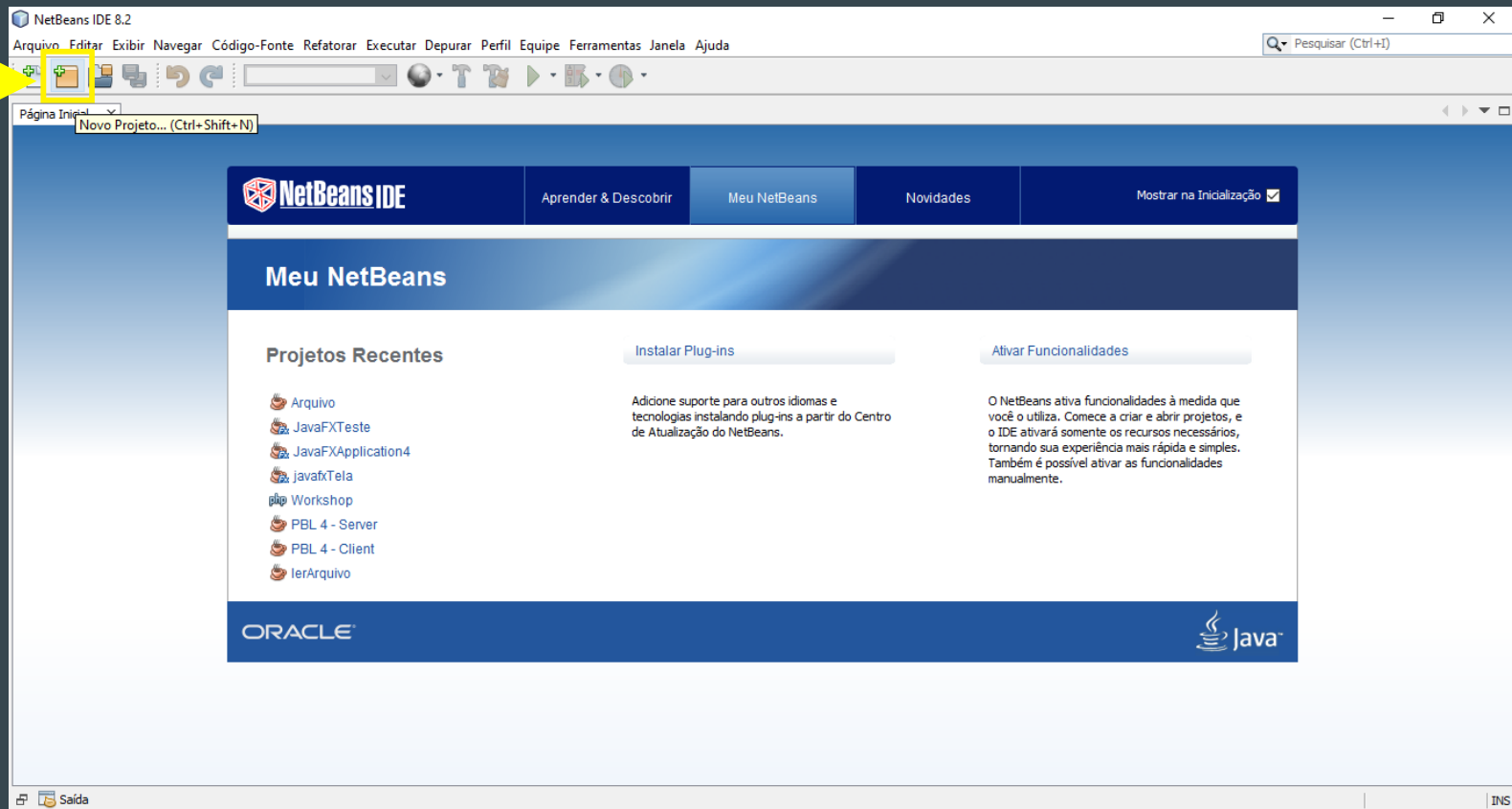


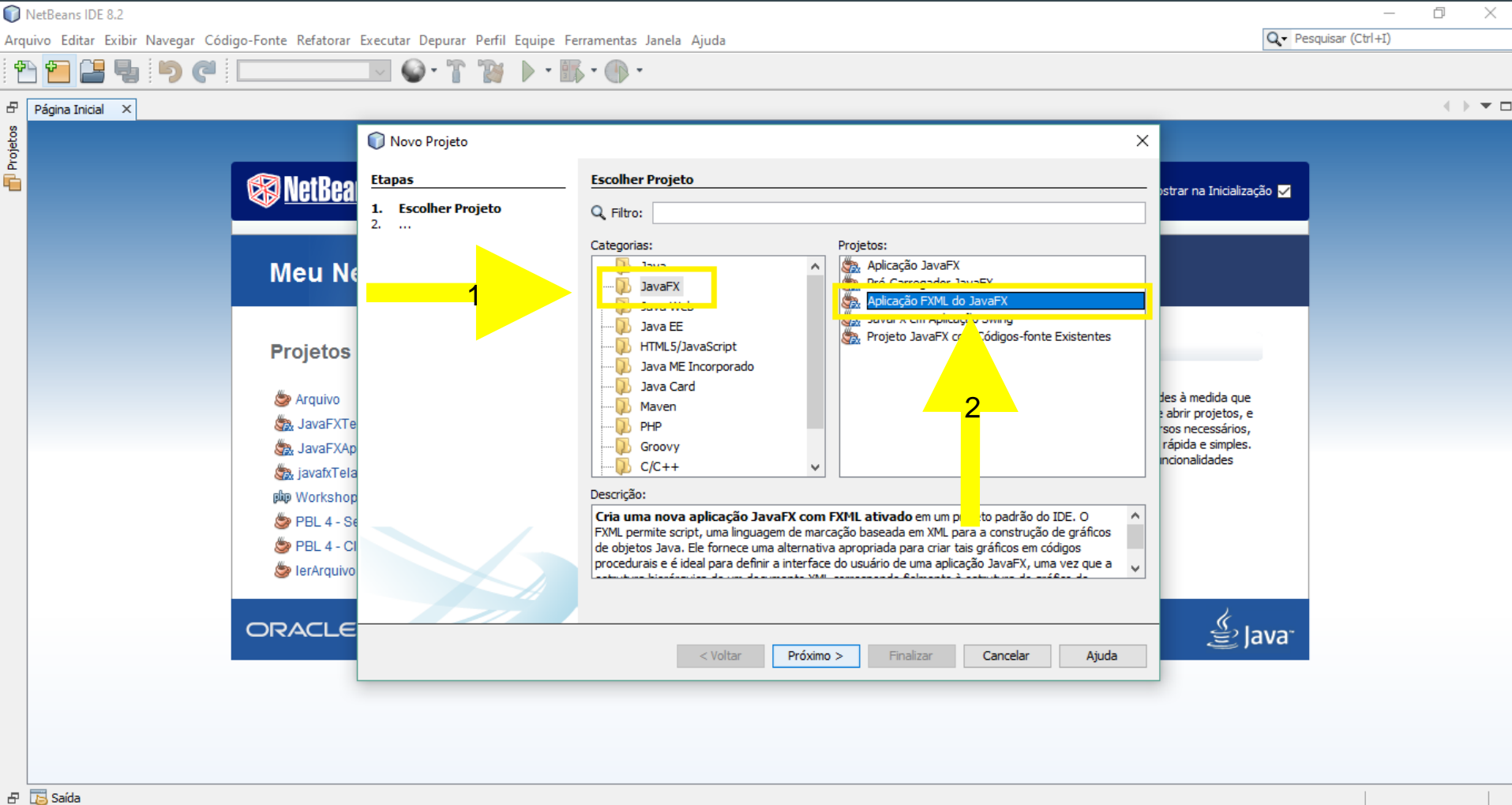
**Calma...**

**pra quê tanta empolgação?**

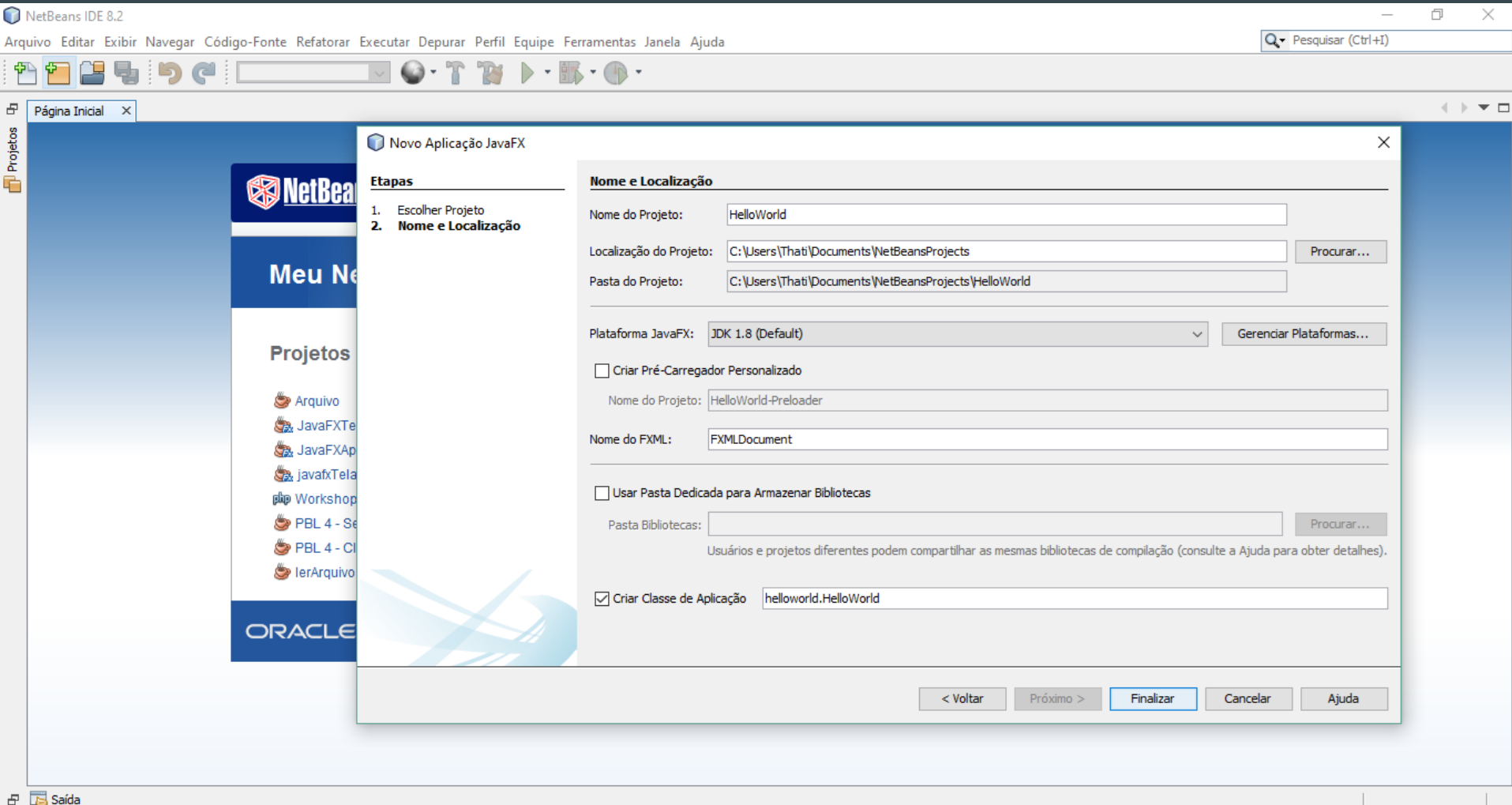
Crifar Meme

# Criação do Projeto









NetBeans IDE 8.2

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Pesquisar (Ctrl+I)

Projetos


- HelloWorld
  - Pacotes de Códigos-fonte
    - helloworld
      - FXMLDocument.fxml
      - @FXMLDocumentController.java
      - @HelloWorld.java
  - Bibliotecas
    - JDK 1.8 (Default)
      - resources.jar
      - rt.jar
      - jsse.jar
      - jce.jar
      - charsets.jar
      - jfr.jar
      - access-bridge-64.jar
      - cdldata.jar
      - dnsns.jar
      - laccess.jar
      - ifxrt.jar
      - localedata.jar
      - nashorn.jar
      - sunec.jar
      - sunjce\_provider.jar
      - sunmscapi.jar
      - sunpkcs11.jar
      - zipfs.jar

Código-Fonte

```
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 helloworld;
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 Thati
17   */
18  public class HelloWorld extends Application {
19
20      @Override
21      public void start(Stage stage) throws Exception {
22          Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));
23
24          Scene scene = new Scene(root);
25
26          stage.setScene(scene);
27          stage.show();
28      }
29
30      /**
31       * @param args the command line arguments
32       */
33  }
```

# Mas então, como fazer minha interface?

JavaFX Scene Builder  FXML

NetBeans(IDE)  FXMLController

FXML  FXMLController

# Apresentação do Controller

Atributos/Componentes

Métodos/Eventos

Initialize

# Alterar Texto da Label

Quando clicar no botão alterar o texto da label para “Seu nome”.

# Eventos

Mouse

Action

Keyboard

Outros

## Segundo projeto

Clicar no botão e mudar o texto da label

Clicar na label e mudar o texto do botão

# Terceiro projeto

Simular um sistema de Login

Digitar nome

Digitar senha

Confirmar se nome=admin e senha=1234

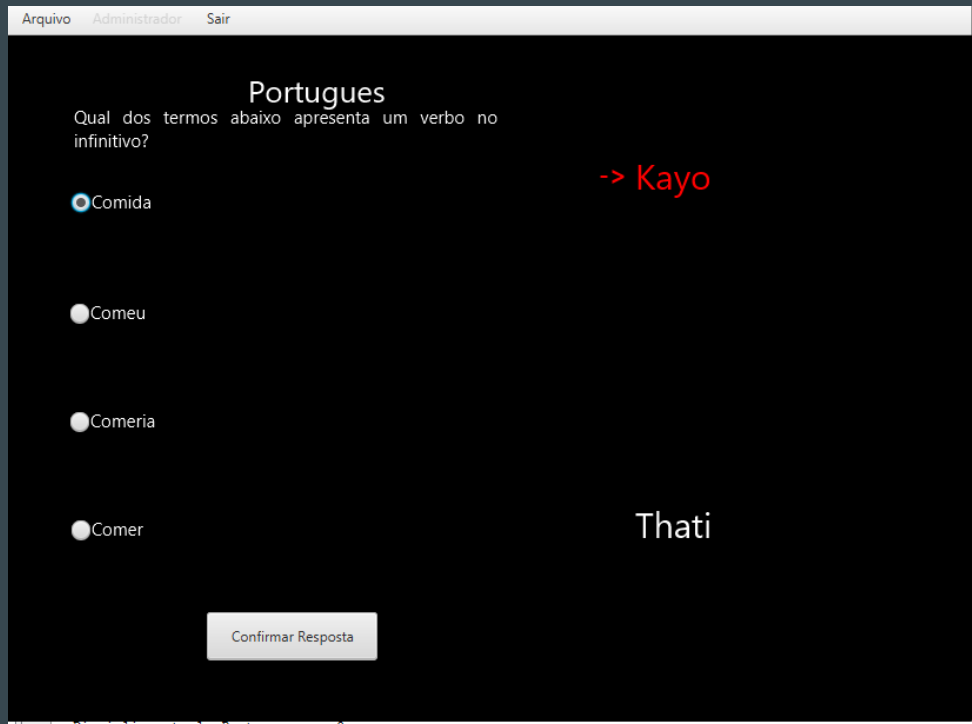
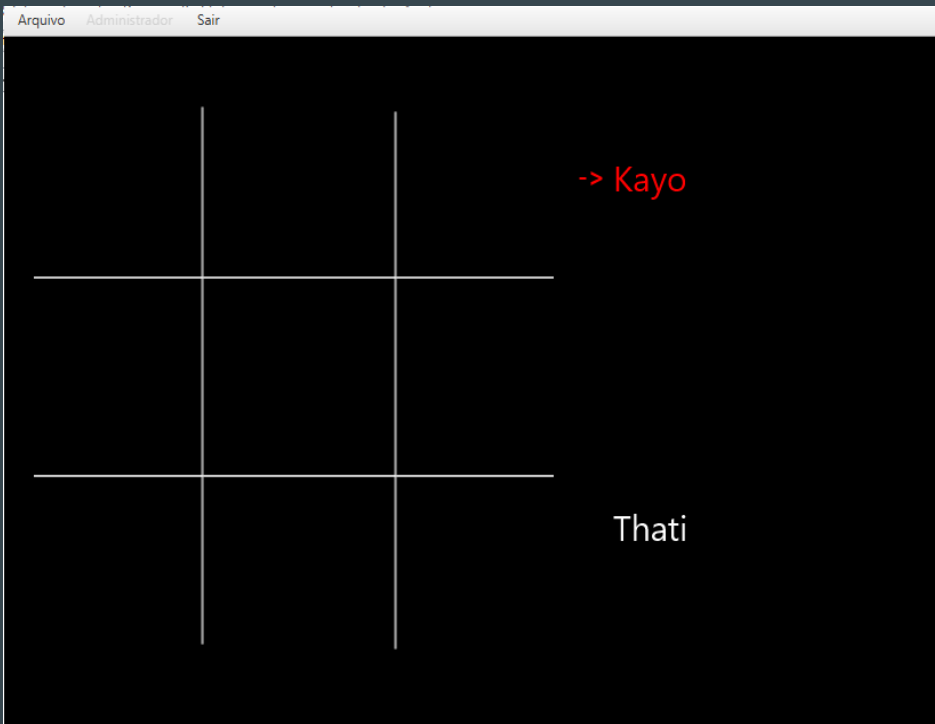
Botão de confirmar

Ao digitar enter na senha -> confirmar



# Linkagem de Páginas

Através da visibilidade dos componentes



# Linkagem de Páginas

Através da visibilidade dos componentes

Através de scenes

```
Parent adm_parent = FXMLLoader.load(getClass().getResource("novaTela.fxml"));  
Scene adm_page = new Scene(adm_parent);  
Stage app_stage = (Stage) componente.getScene().getWindow();  
app_stage.hide();  
app_stage.setScene(adm_page);  
app_stage.show();
```

**Mas como passar os dados entre as telas?**

# Mas como passar os dados entre as telas?

- > Atributos e/ou Métodos (Static)
- > Getters e Setters

## Terceiro projeto V2

Após validado o Login, ir para outra tela que exibirá “Olá, nome do usuário”

Opção de mostrar senha

# Projeto Final (Calculadora)

Científica / Normal (\*Foco na normal)

Interface Simples

Operandos (0~9) através entrada do teclado ou botão.

Para cada tecla apertada o botão correspondente deverá também receber foco.

Operadores (+ - \* / =) através do clique e implementar as operações

# Como serão avaliados

Funcionamento do fluxo do programa (operador1 + operação + operador2)

Resultados Coerentes para as 4 operações matemáticas

Implementação da Calculadora Científica

Análise de Interface

Análise de Funcionamento dos Eventos

A nota final será definida pelos 3 tutores



**JÁ ACABOU,  
JÉSSICA?**



A large, faint watermark of the FIM logo is centered in the background. It consists of a circle containing a stylized 'SC' monogram.

**FIM**

ecomprjr.