Uma breve introdução ao JavaFX

Programação Orientada a Objetos — QXD0007

Prof. Atílio Gomes Luiz

gomes.atilio@ufc.br

Universidade Federal do Ceará

2º semestre/2021

Objetivos

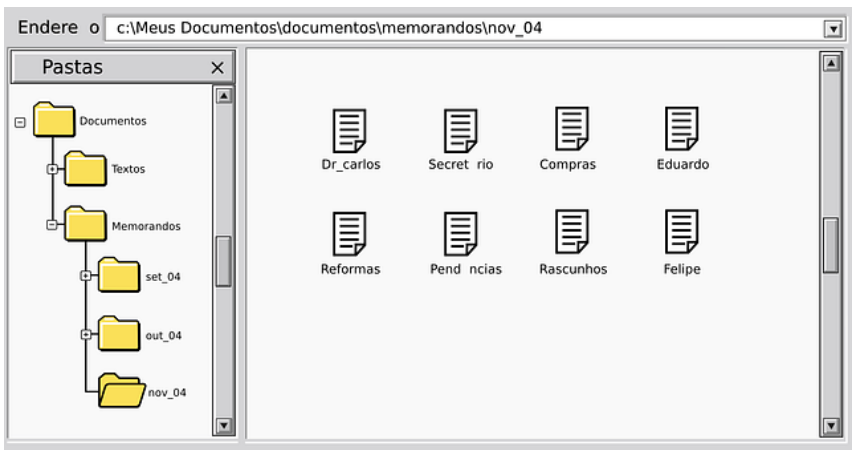
UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

- Instalar o JavaFX e o Scene Builder
- Configurar um projeto no VSCode para que ele reconheça uma aplicação JavaFX
- Criar janelas simples com alguns painéis e botões
- Projeto: construir uma calculadora simples usando o SceneBuilder + VsCode + JavaFX

Introdução

Interfaces Gráficas

- Uma interface gráfica do usuário, ou **GUI** (*graphical user interface*), consiste em um modelo de interface que permite ao usuário interagir com dispositivos eletrônicos usando ícones gráficos e componentes visuais (widgets)



pixabay.com

Eventos

- **Evento**: um objeto que representa a interação do usuário com a componente GUI.

- Tipos de eventos:
  - **Mouse:** mover/clicar/arrastar, pressionar/liberar botão
  - **Teclado:** pressionar/liberar tecla,
  - **Touchscreen:** pressionar/arrastar com o dedo
  - **Janela:** aumentar, minimizar, restaurar, fechar

- **Interface toolkits:** bibliotecas de objetos interativos (widgets) que usamos para auxiliar na construção GUIs e também no tratamento de eventos. Vamos ver o interface toolkit JavaFX.

JavaFX

- JavaFX é um novo framework para desenvolvimento de programas Java GUI
  - Funcionalidade gráfica é fornecida pela biblioteca
  - Inclui funcionalidades para GUI, gráficos e multimídia (imagens, animação, áudio e vídeo)

- História da GUI em Java
  - Java foi lançado com a AWT (Abstract Window Toolkit), que era dependente de plataforma.
  - Java 1.2 até Java 7: Swing — provavelmente, nunca morrerá, muitas aplicações ainda usam
  - A partir do Java 8: JavaFX
    - Diferentemente do Swing, JDK não vem no SDK, deve ser instalada separadamente

Ferramental

- Uma tela JavaFX pode ser montada via código Java, ou via código FXML
  - FXML é uma variação do XML, uma linguagem de marcação.
- Com o lançamento do Java 11, JavaFX não é mais parte do JDK
  - O JavaFX precisa ser baixado e configurado separadamente
  - É mantido pela Gluon: `https://gluonhq.com/products/javafx/`

Scene Builder

- Ferramenta WYSIWIG (What You See Is What You Get) — tem paletas para layouts, controles e propriedades, um editor para criar cenas.
- **Código fonte:** desenvolvido e mantido pelo OpenJFX Project da comunidade OpenJDK
- **Instalador:** a Oracle disponibiliza o código fonte do Scene Builder. Os builds para instalação são mantidos pela Gluon.
  - Download: `https://gluonhq.com/products/scene-builder/`

Configurando o JavaFX no VS Code

- Siga os passos no link:
  `https://github.com/Atilio-Luiz/poo_redes_2021/tree/master/javafx`

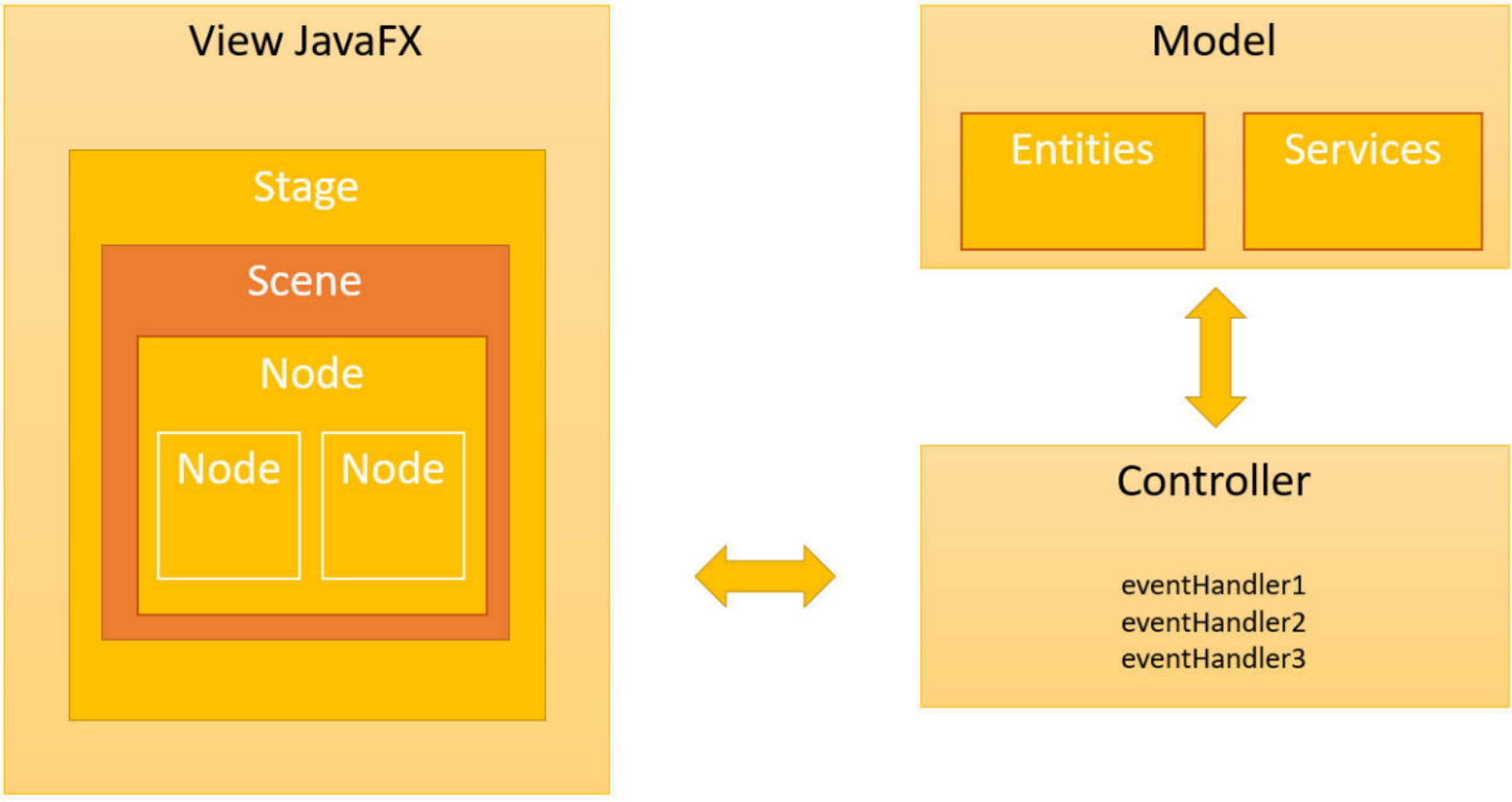
Visão geral do JavaFX e MVC

JavaFX é projetado sobre o padrão MVC (Model-View-Controller)

- Model - consiste nos dados de domínio e toda lógica de transformação desses dados
- View - São as telas de interação com o usuário (GUI)
- Controller - São as classes responsáveis por tratar as interações do usuário com as views (manipulação de eventos de interação com as telas)

Visão geral do JavaFX e MVC

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ



- Link para a hierarquia das classes do JavaFX:
  https://openjfx.io/javadoc/17/overview-tree.html

JavaFX — nossa primeira janela

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

```
1  import javafx.application.Application;
2  import javafx.scene.Scene;
3  import javafx.scene.layout.VBox;
4  import javafx.stage.Stage;
5
6  public class HelloWorld extends Application {
7      public static void main(String[] args){
8          System.out.println("Começando...");
9          launch(args);
10         System.out.println("Terminando...");
11     }
12
13     @Override
14     public void start(Stage stage) throws Exception {
15         Scene scene = new Scene(new VBox());
16         stage.setScene(scene);
17         stage.setTitle("Primeira Janela GUI!");
18         stage.show();
19     }
20 }
```

JavaFX — nossa primeira janela

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

```
1  import javafx.application.Application;
2  import javafx.scene.Scene;
3  import javafx.scene.layout.VBox;
4  import javafx.stage.Stage;
5
6  public class HelloWorld extends Application {
7      public static void main(String[] args){
8          System.out.println("Começando...");
9          launch(args);
10         System.out.println("Terminando...");
11     }
12
13     @Override
14     public void start(Stage stage) throws Exception {
15         Scene scene = new Scene(new VBox());
16         stage.setScene(scene);
17         stage.setTitle("Primeira Janela GUI!");
18         stage.show();
19     }
20 }
```

Classe Application para janela UI

main usa launch para chamar start

Um Stage é criado automaticamente pelo JavaFX
e passado como argumento para start()
Um Stage é uma janela
show() faz a janela aparecer

Classe `Application`

- Programas JavaFX incluem uma classe que estende `Application` do pacote `javafx.application.Application`

- Ao executar uma classe do tipo `Application`, JavaFX faz o seguinte:
  - Constrói uma instância da classe `Application`
  - Chama um método `init()` para iniciar a aplicação
    ... não construa um Stage ou um Scene no init()
  - Chama o método `start(javafx.stage.Stage)`
  - Espera a aplicação terminar: ou você chama `Platform.exit()` ou a última janela é fechada.
  - Chama o método `stop()` para liberar recursos. Os métodos `init()` e `stop()` possuem implementações default vazias.

JavaFX — nossa primeira janela

Então, temos uma janela (Stage),
o que vamos colocar nela?

JavaFX — nosso primeiro botão


```
1  import javafx.application.Application;
2  import javafx.scene.Scene;
3  import javafx.scene.control.Button;
4  import javafx.scene.layout.VBox;
5  import javafx.stage.Stage;
6
7  public class HelloButton extends Application {
8      public static void main(String[] args) {
9          launch(args);
10     }
11
12     @Override
13     public void start(Stage stage) throws Exception {
14         Button okButton = new Button("Olá");
15         okButton.setPrefSize(400, 200);
16         VBox pane = new VBox(okButton);
17
18         Scene scene = new Scene(pane);
19         stage.setScene(scene);
20         stage.setTitle("Primeiro Botão");
21         stage.show();
22     }
23 }
```

Terminologia

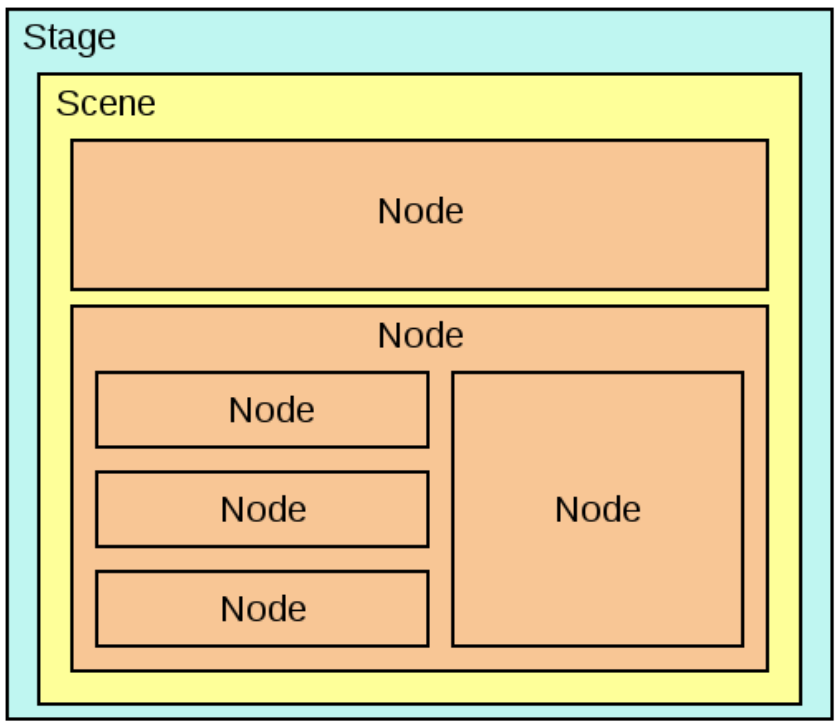
- Stage
  - representa janelas, top level **container**
  - possui vários métodos *setter*: `setTitle()`, `setWidth()`
  - um stage é criado por *default* pela `Application`
  - você pode ter múltiplos stages e setar uma delas como a sua *main stage*:
    construa uma `Stage` para cada janela na sua aplicação, e.g., para diálogos e pop-ups.

- Scene
  - cada `stage` tem uma `scene` (scene graph container)
  - em uma scene você pode colocar **controls** (Buttons, Labels, etc.)
  - você pode colocar controls diretamente em scenes, ou usar **Panes** para melhorar o layout e criar hierarquias de widgets.

Estrutura Básica

Estrutura básica de um programa JavaFX

- Estenda a classe `Application`
- Sobreponha o método `start(Stage)`
- `Stage` ← `Scene` ← `Nodes` (Panes ou Controls)

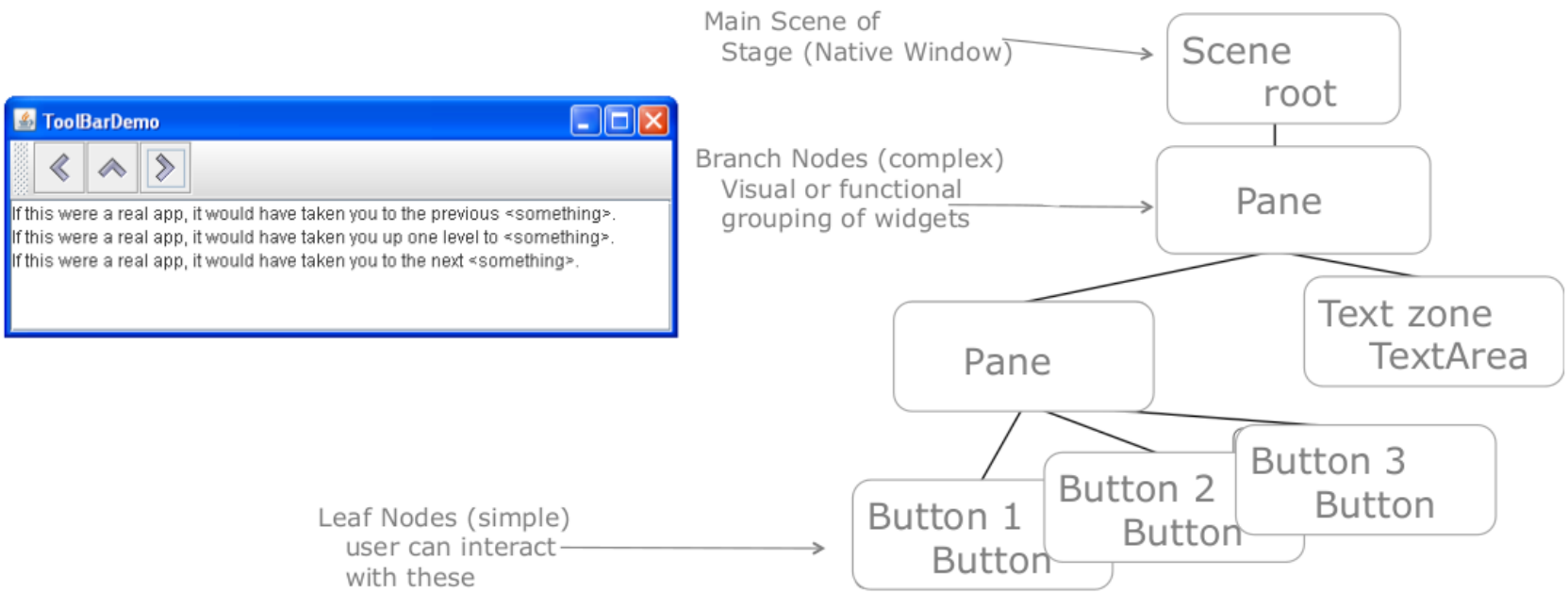


https://commons.wikimedia.org/wiki/File:Javafx-stage-scene-node.svg

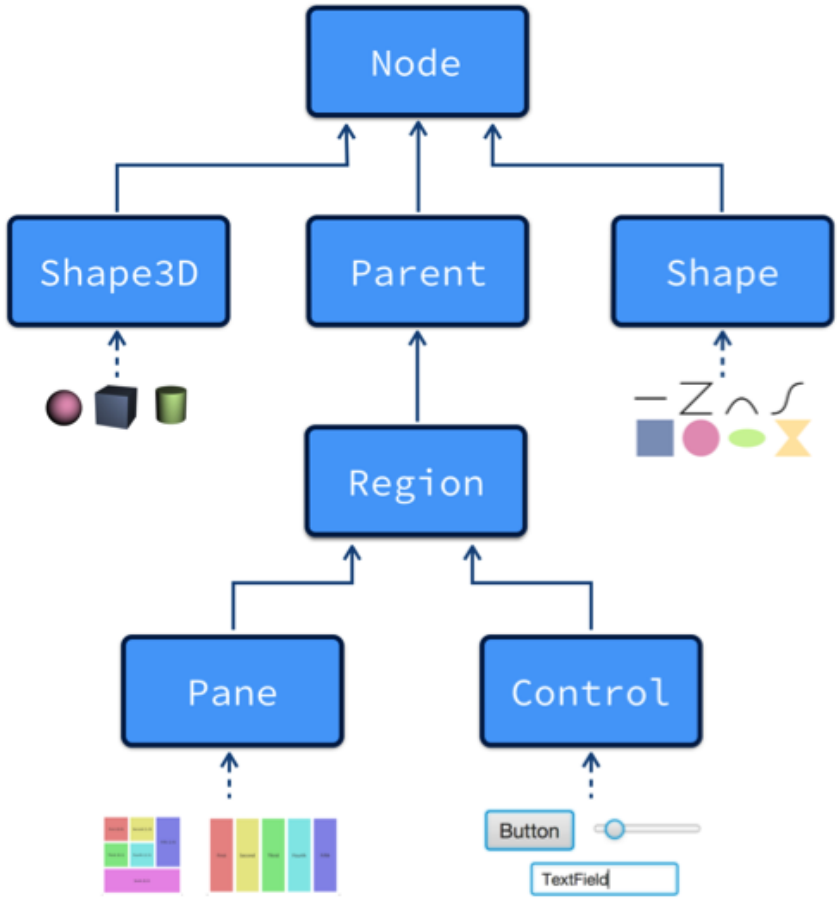
Estrutura Básica — Scene graph

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

- **Scene graph**: uma estrutura de dados hierárquica (árvore) que representa os conteúdos em uma scene.
- Um **node** é um objeto gráfico que está contido no scene graph.
  - Um node pode pertencer somente a um container (Pane)



Hierarquia Node

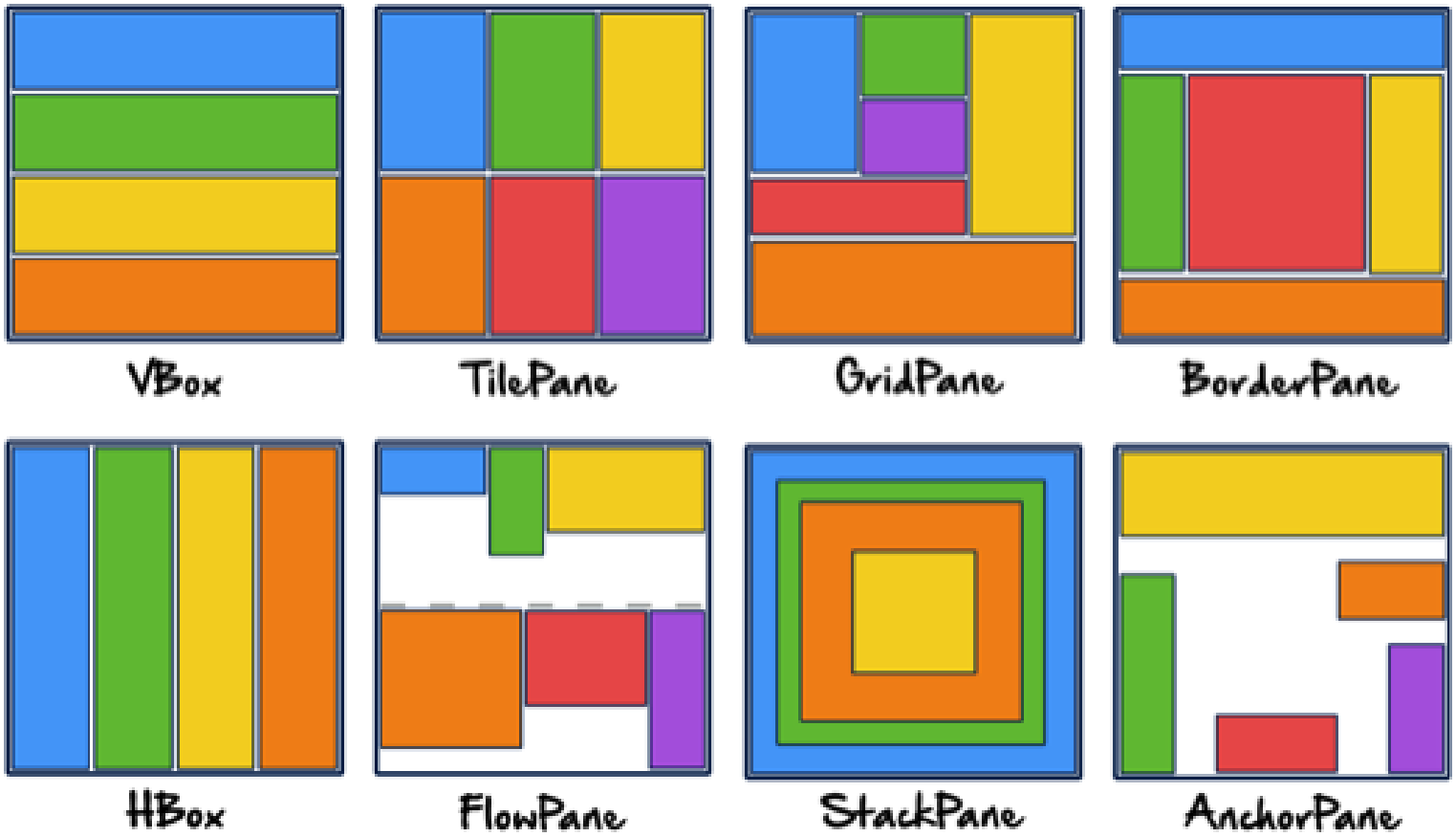


Interface toolkits

- Todos as *interfaces toolkits*, em particular o JavaFX, possuem:
  - uma coleção de controles UI (**Classes herdeiras de Control**)
    - TextFiled, Label, Button, ComboBox, RadioBox, etc.
  - um modo de organizar esses controles (**layout panes**)
  - um modo de gerenciar os eventos disparados pelos controles a partir da interação do usuário com esses controles

JavaFX layout Panes

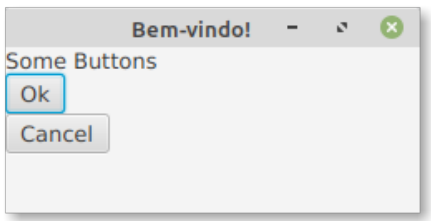
Layouts disponíveis no JavaFX



tutorial: `https://docs.oracle.com/javafx/2/layout/builtin_layouts.htm`
Imagens de https://dzone.com/refcardz/javafx-8-1

VBox — Exemplo


```
1  public class VBoxExample01 extends Application {
2      public static void main(String[] args) {
3          launch(args);
4      }
5
6      @Override
7      public void start(Stage stage) throws Exception {
8          Label descriptionLabel =
9              new Label("Some Buttons");
10         Button okButton = new Button("Ok");
11         Button cancelButton = new Button("Cancel");
12
13         VBox root = new VBox();
14         root.getChildren().addAll(descriptionLabel,
15             okButton, cancelButton);
16
17         Scene scene = new Scene(root, 250, 100);
18         stage.setScene(scene);
19         stage.setTitle("Bem-vindo!");
20         stage.show();
21     }
22 }
```

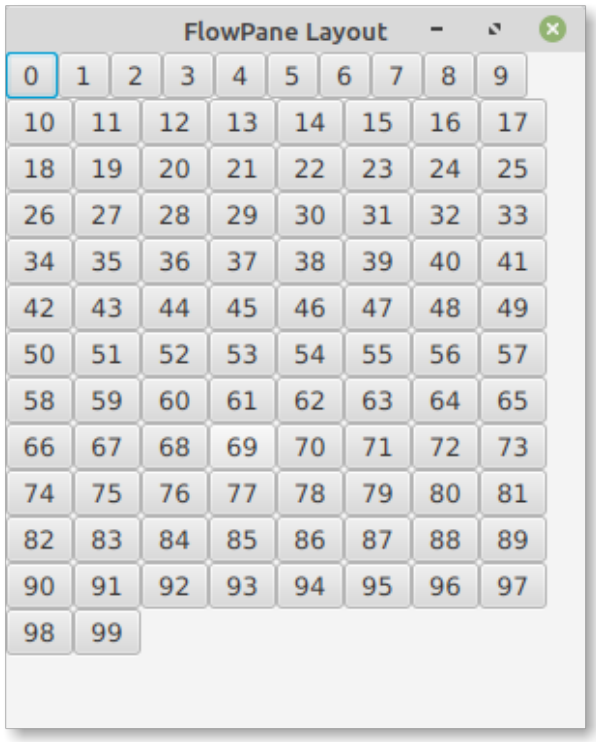


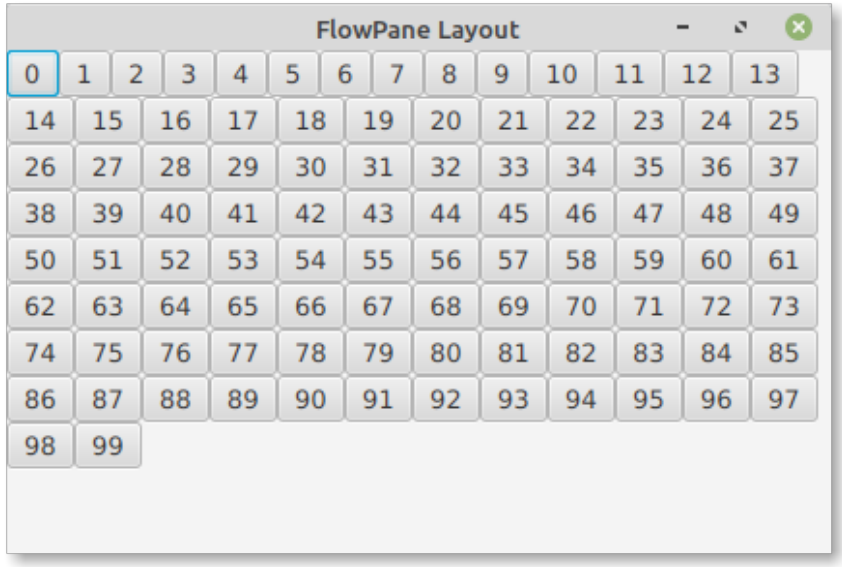
FlowPane — Exemplo


```
1  public class FlowPaneExample01 extends Application {
2      public static void main(String[] args) {
3          launch(args);
4      }
5
6      @Override
7      public void start(Stage stage) throws Exception {
8          FlowPane root = new FlowPane();
9
10         for(int i = 0; i < 100; i++) {
11             root.getChildren()
12                 .add(new Button(Integer.toString(i)));
13         }
14
15         Scene scene = new Scene(root, 300, 350);
16         stage.setTitle("FlowPane Layout");
17         stage.setScene(scene);
18         stage.show();
19     }
20 }
```

FlowPane — Exemplo

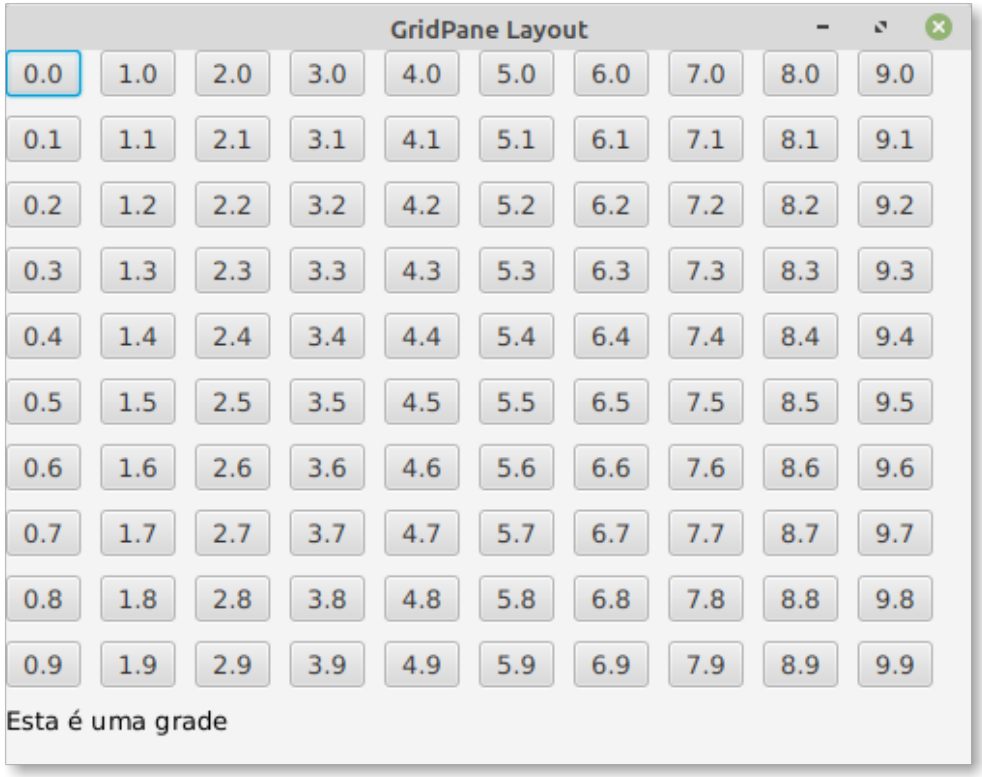
UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ





GridPane — Exemplo

- Ver arquivo GridPaneExample01.java

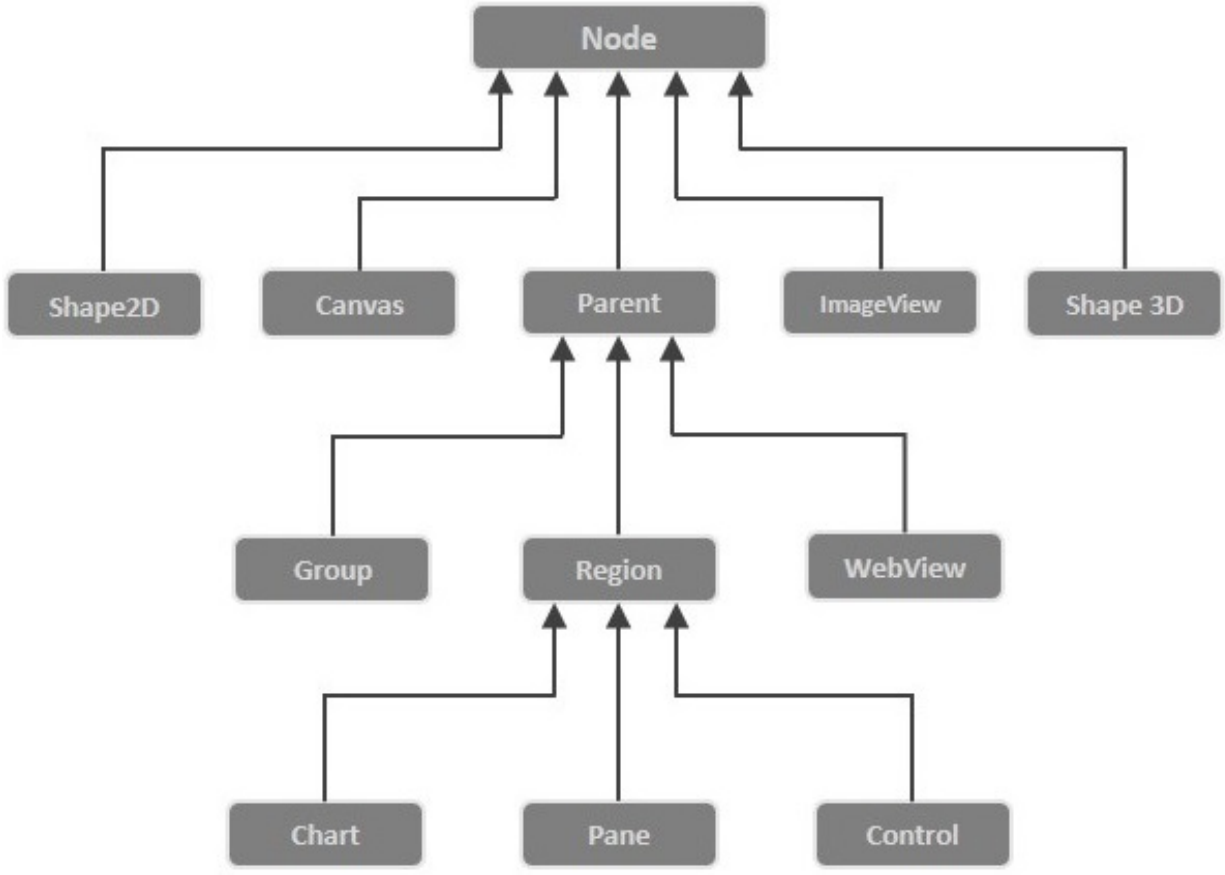


Classes de Layout

- **Pane**: Classe base para todas as classes de layout `Pane`. Ela contém o método `getChildren()` que retorna todos os nodes no Pane.
- **AnchorPane**: Nodes são ancorados dos lados ou ao centro do Pane.
- **BorderPane**: Nodes são colocados em uma de cinco regiões: top, bottom, center, left, right
- **FlowPane**: Nodes se movem a fim de preencher o espaço horizontal (vertical).
- **GridPane**: Node são colocados em células de uma grade.
- **StackPane**: Nodes são colocados um em cima do outro no centro do Pane.
- **HBox**: Nodes horizontalmente.
- **VBox**: Nodes verticalmente.

A classe `javafx.scene.Node`

Layout Panes são considerados nodes (assim como os Controls). Logo, eles podem ser adicionados a outros Panes.

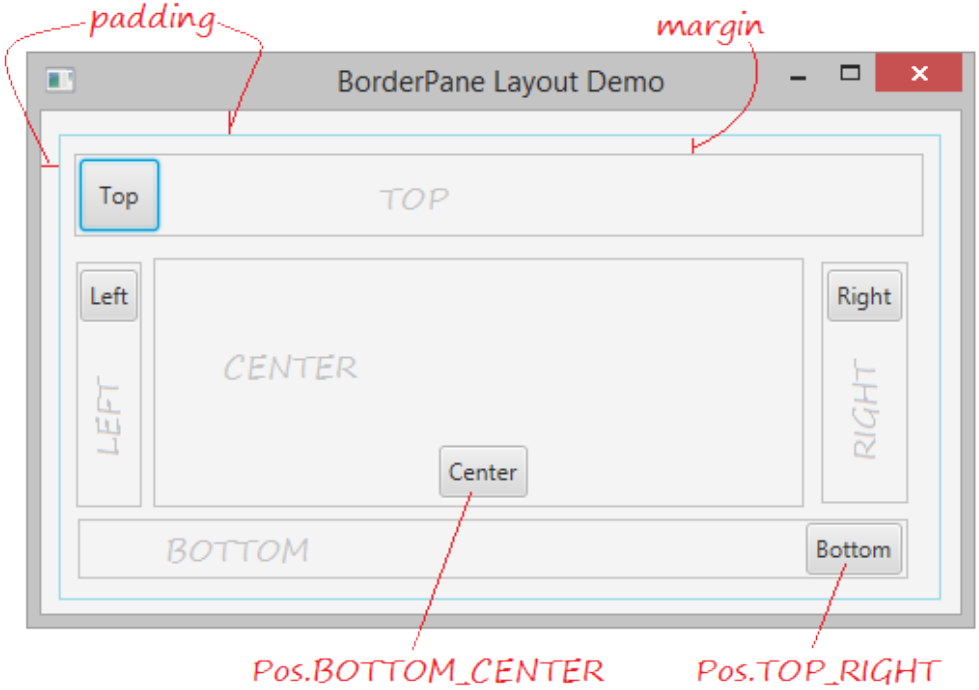


https://www.tutorialspoint.com/javafx/javafx_application.htm

Melhorando o layout

Universidade Federal do Ceará
Campus Quixadá

Layout Panes possuem diferentes propriedades que ajudam a criar layouts que persistem mesmo após um redimensionamento de tela.



https://o7planning.org/10629/javafx-borderpane

A classe `javafx.scene.Region`

Todos os Panes e Controls herdam a classe `Region`. Esta classe possui diversos métodos para configuração do layout, dentre os quais:

- `void setPadding(Insets value)`: O preenchimento superior, direito, inferior e esquerdo ao redor do conteúdo da região.

- `void setStyle(String value)`: Uma string contendo o estilo CSS associado a esse `Node`.

- Métodos para configuração de largura e altura:
  - `void setPrefHeight(double value)`
  - `void setPrefWidth(double value)`
  - `void setPrefSize(double value)`
  - `void setMinSize(double minWidth, double minHeight)`
  - `void setMaxSize(double maxWidth, double masHeight)`

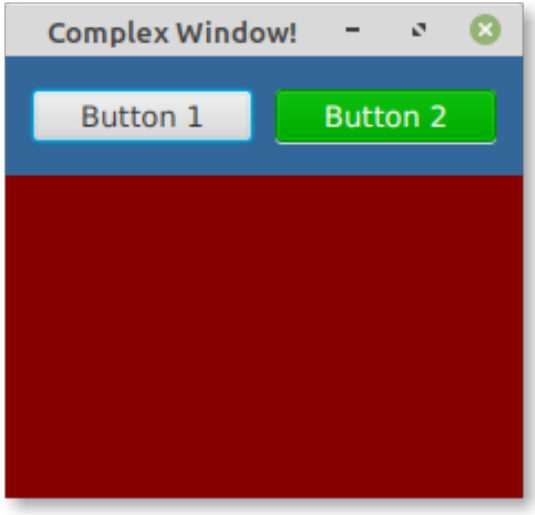
Classe `javafx.geometry.Insets`

- A classe Insets armazena os deslocamentos internos para os quatro lados da área retangular. Insets herda a classe `Object`.

- Construtores da classe:
  - `Insets(double a)`: Constrói uma nova instância de Insects com o mesmo valor para todos os quatro deslocamentos.
  - `Insets(double top, double right, double bottom, double left)`: Constrói uma nova instância de Insects com quatro deslocamentos diferentes.

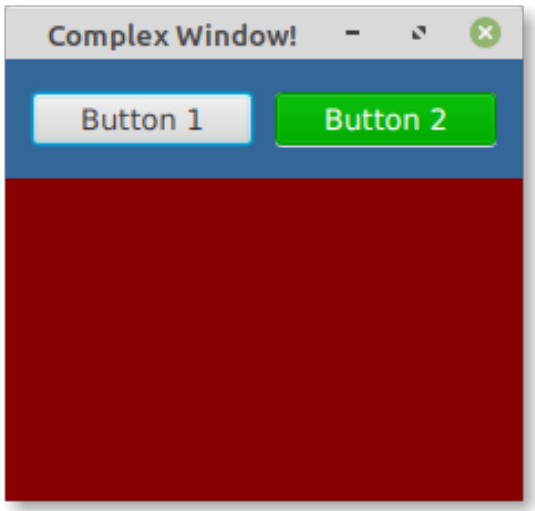
Exemplo

- Ver projeto `ImprovedLayouts`



Exemplo

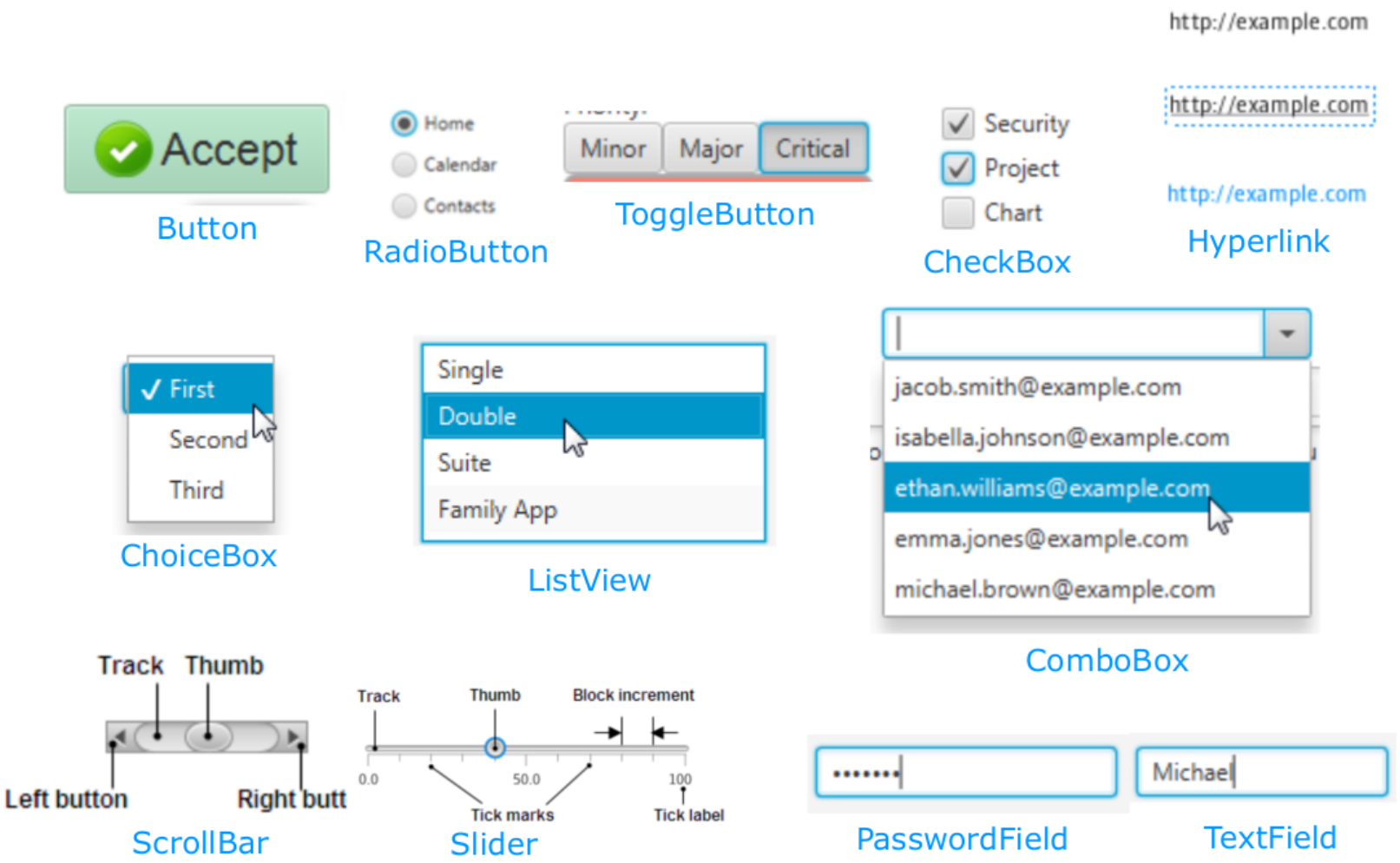
- Ver projeto `ImprovedLayouts`



Ver projeto BorderPane

JavaFX Controls

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ



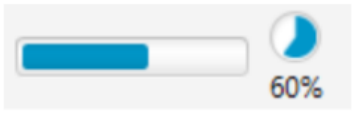
https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/ui_controls.htm#JFXUI336

JavaFX Controls (não editáveis)

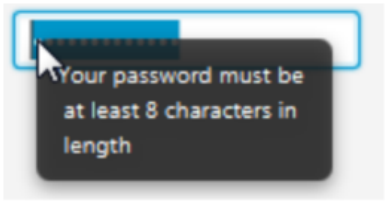
UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Label

Separator



ProgressBar
ProgressIndicator

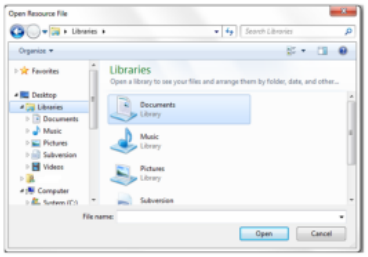


Tooltip

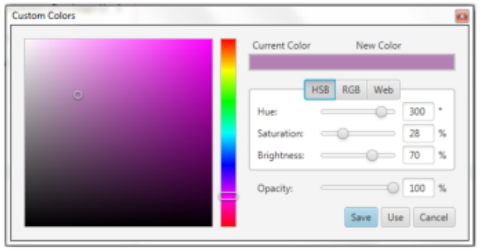
https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/ui_controls.htm#JFXUI336

JavaFX Controls (mais complexos)

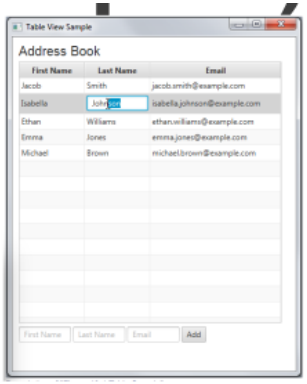
UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ



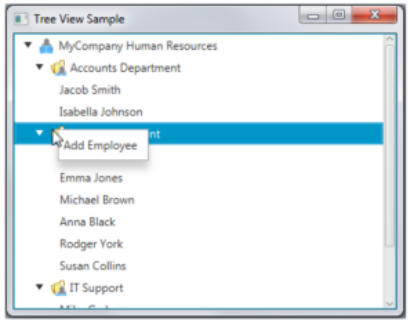
FileChooser



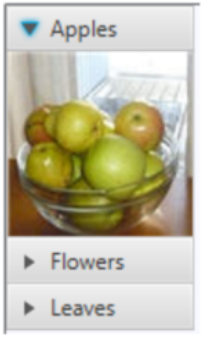
ColorPicker



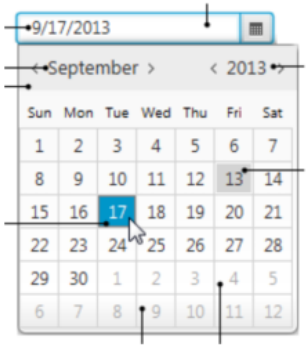
TableView,
TableColumn



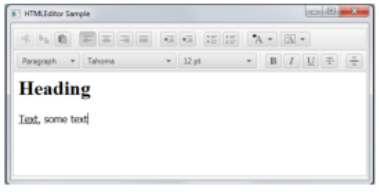
TreeView



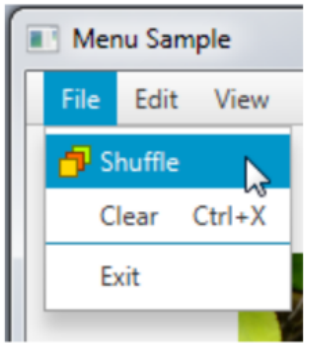
Accordion
TitlePane



DatePicker



HTMLEditor

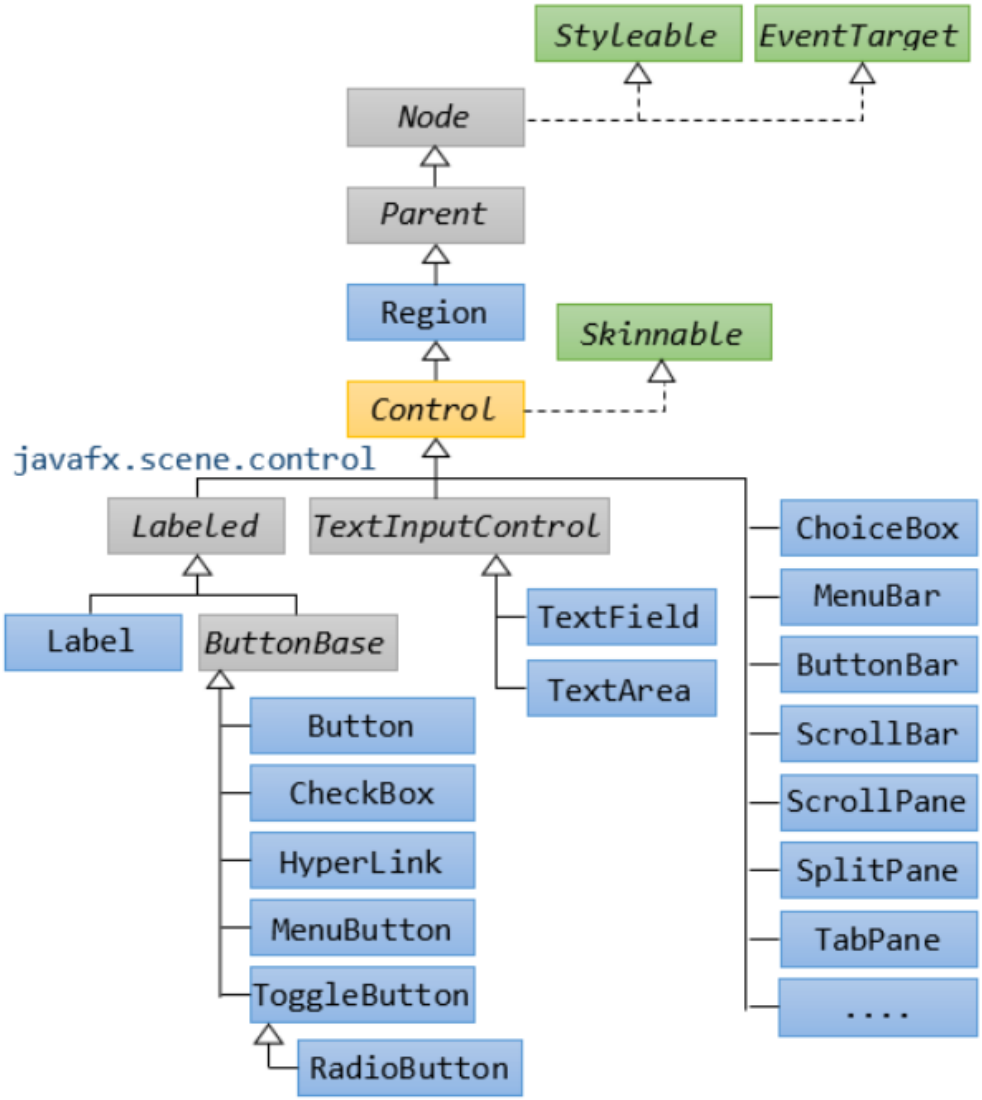


Menu, MenuItem

https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/ui_controls.htm#JFXUI336

JavaFX Control Classes

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ



Prática

- Fazer o checklist

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

JavaFX e CSS

Design Consistente

- Imagine que temos uma ou mais janelas e decidimos que queremos mudar o visual de todas elas ...

CSS (Cascading Style Sheets)

- CSS descreve como elementos da HTML podem ser mostrados na tela
- CSS econimiza muito o trabalho. Pode controlar o layout de múltiplas páginas web tudo de um vez em um só lugar.
- Link: `https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/css_tutorial.htm#JFXUI733`

Design Consistente

- No seu projeto, crie um arquivo CSS, por exemplo, style.css

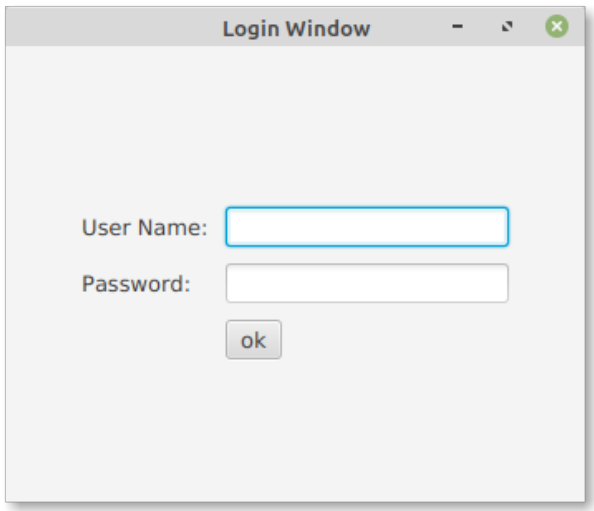
- Insira algumas propriedades de estilo no arquivo criado:

```
.root {
     -fx-background-image: url("background.jpeg");
}

.label {
    -fx-font-size: 12px;
    -fx-font-weight: bold;
    -fx-text-fill: #333333;
    -fx-effect: dropshadow( gaussian , rgba(255,255,255,0.5) , 0,0,0,1 );
}
.button {
    -fx-text-fill: white;
    -fx-font-family: "Arial Narrow";
    -fx-font-weight: bold;
    -fx-background-color: linear-gradient(#61a2b1, #2A5058);
    -fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );
}

.button:hover {
    -fx-background-color: linear-gradient(#2A5058, #61a2b1);
}
```

Design Consistente usando CSS

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

- Um modo simples de aplicar estilo a todas as janelas
- Ver o projeto LoginWindow



Links úteis

- Documentação do JavaFX: `https://openjfx.io/javadoc/17/`
- Breve tutorial da Oracle: `https://docs.oracle.com/javase/8/javase-clienttechnologies.htm`
- `https://docs.oracle.com/javase/8/javafx/layout-tutorial/size_align.htm#JFXLY133`
- Vários layouts: `https://o7planning.org/en/11009/javafx`
- `https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/css_tutorial.htm#JFXUI733`
- Tutorial construindo uma aplicação gráfica JavaFX do zero: `https://code.makery.ch/pt/library/javafx-tutorial/part1/`

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

FIM