

Programação Orientada por Objetos

JavaFX – Controlos II



Prof. Cédric Grueau

Prof. José Sena Pereira

Departamento de Sistemas e Informática

Escola Superior de Tecnologia de Setúbal

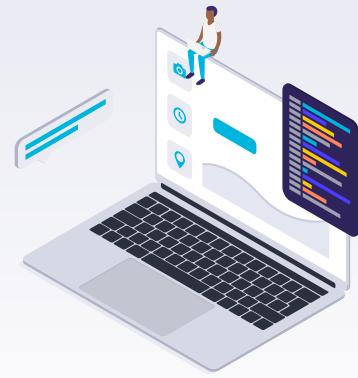
Instituto Politécnico de Setúbal

2022/2023

Sumário



- ▶ Controlos ListView e ComboBox
- ▶ Controlos Tabs, Accordion e TitledPane
- ▶ Menus em JavaFX
- ▶ Controlos Checkbox e Radiobutton



LISTVIEW E COMBOBOX

- ▶ JavaFX – Controlos II



JavaFX - Controlos

- ▶ As classes para criar controlos encontram-se no pacote
 - ▶ **`javafx.scene.control`**
- ▶ Exemplos anteriores:
 - ▶ **Button**
 - ▶ **TextField**
 - ▶ **Label**
 - ▶ **ListView**



JavaFX – Revendo o exemplo da ListView

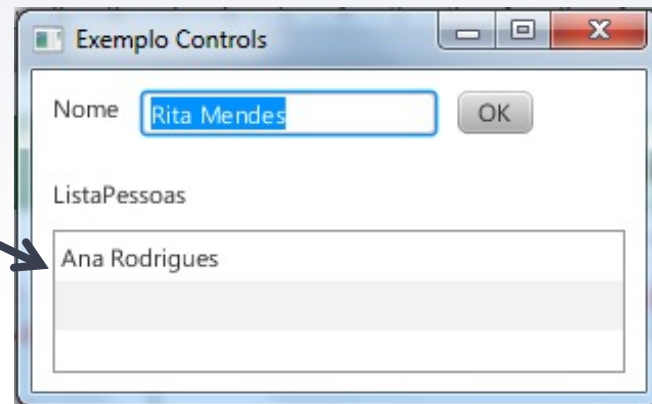
❑ **ListView** – Criação e Utilização

1. Criar um objeto do tipo
ListView.

2. Criar uma
ObservableList de
Strings e associá-la
à **ListView**

3. Adicionar linhas de texto
à **ObservableList**
para preencher a
ListView

ListView



```
ListView<String> listOfNames = new ListView<>();
```

```
ObservableList<String> items=FXCollections.observableArrayList();  
listOfNames.setItems(items);
```

```
items.add(name);
```

JavaFX – Usando uma ComboBox

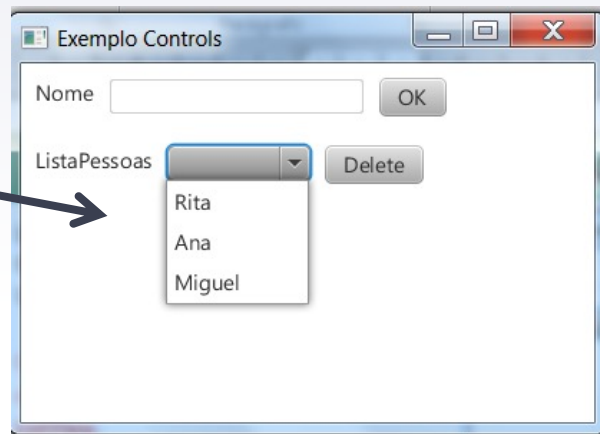
❑ ComboBox – Criação e Utilização

1. Criar um objeto do tipo
ComboBox.

2. Criar uma
ObservableList de
Strings e associá-la
à ComboBox

3. Adicionar linhas de texto
à ObservableList
para preencher a
ComboBox

ComboBox

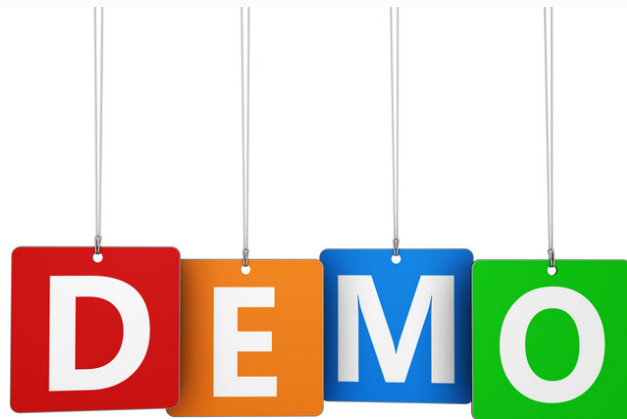


```
ComboBox<String> listOfNames= new ComboBox<>();
```

```
ObservableList<String> items=FXCollections.observableArrayList();  
listOfNames.setItems(items);
```

```
items.add(name);
```

JavaFX- Eventos : Exemplo



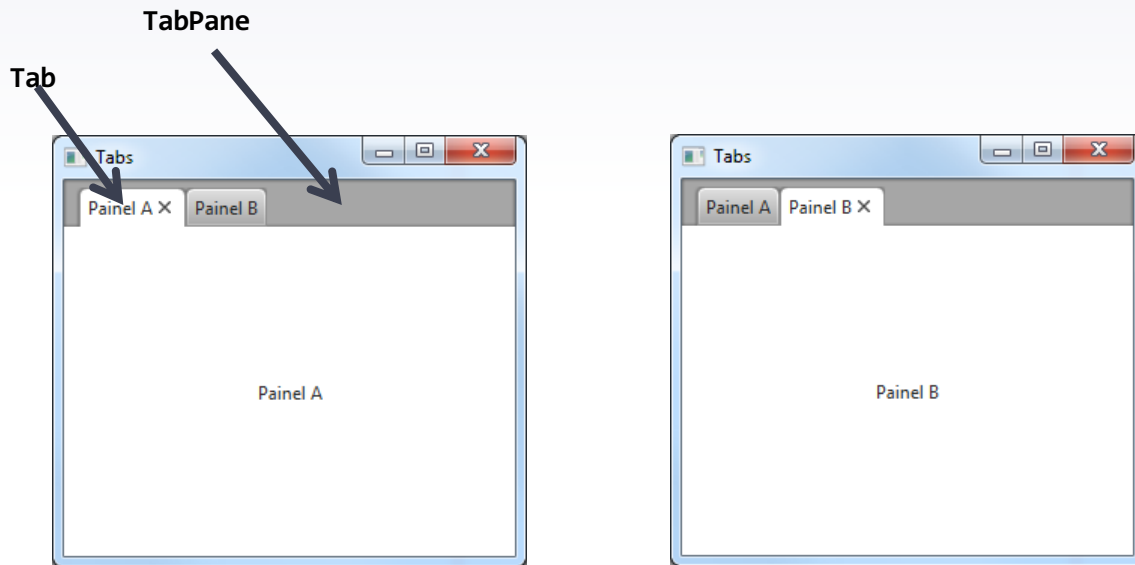
Tabs, Accordion e TitledPane

- ▶ JavaFX – Controles II



JavaFX – Exemplo: Tab

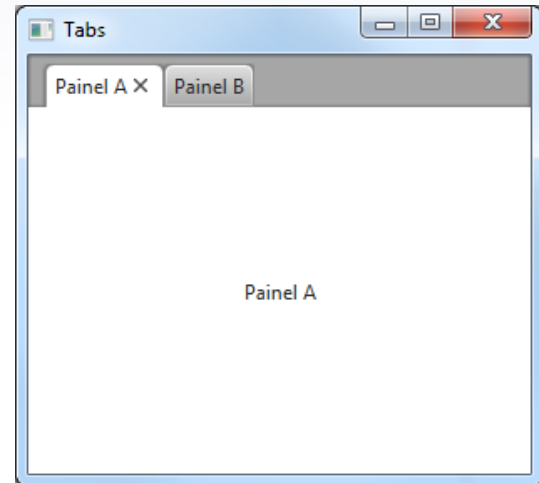
Uma forma simples de gerir a alternância entre vários painéis é através da utilização de um **TabPane** com vários **Tab** associados.



JavaFX – Exemplo: Tab

Objetivo: Ter dois Painéis A e B, e alternar entre um e outro consoante o **Tab** selecionado.

```
public class PanelWithTab extends StackPane {  
  
    public PanelWithTabs () {  
        TabPane tabPane = new TabPane();  
        Tab tabA= new Tab("Painel A");  
        Tab tabB= new Tab("Painel B");  
        tabA.setContent(new PanelA());  
        tabB.setContent(new PanelB());  
        tabPane.getTabs().addAll(tabA,tabB);  
        this.getChildren().add(tabPane);  
    }  
}  
  
public class PanelA extends StackPane {  
  
    public PanelA() {  
        this.getChildren().add(new Label("Painel A"));  
    }  
}
```



JavaFX – Exemplo: Tab

Objetivo: Ter dois Painéis A e B, e alternar entre um e outro consoante o **Tab** selecionado.

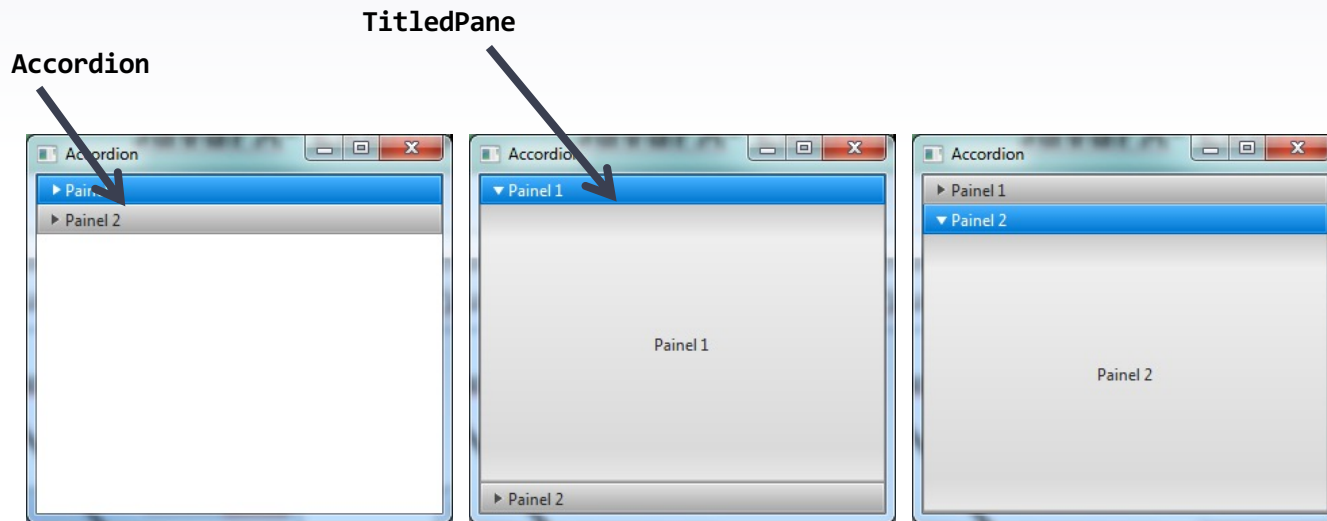
```
public class PanelWithTab extends StackPane {  
  
    public PanelWithTab () {  
        TabPane tabPane = new TabPane();  
        Tab tabA= new Tab("Painel A");  
        Tab tabB= new Tab("Painel B");  
        tabA.setContent(new PanelA());  
        tabB.setContent(new PanelB());  
        tabPane.getTabs().addAll(tabA,tabB);  
        this.getChildren().add(tabPane);  
    }  
}  
  
public class PanelA extends StackPane {  
  
    public PanelA() {  
        this.getChildren().add(new Label("Painel A"));  
    }  
}
```

1. Criar a **TabPane**
2. Criar objetos do tipo **Tab**.
3. Associar a cada **Tab** o **Node** pretendido.
Nota: **PanelA** e **PanelB** são classes derivadas de **Node**, normalmente painéis (**GridPane**, **HBox**, **VBox**, ...).
4. Associar cada **Tab** ao **TabPane**

Para seleccionar um tab:
`tabPane.getSelectionModel().select(1);`

JavaFX – Exemplo: Accordion

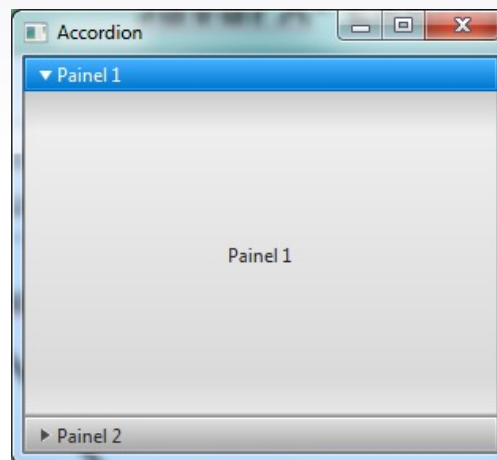
Outra forma simples de gerir a alternância entre vários painéis é através da utilização de um **Accordion** com vários **TitledPane** associados.



JavaFX – Exemplo: Accordion

Objetivo: Ter dois Painéis 1 e 2, e alternar entre um e outro consoante a seleção.

```
public class PanelWithAccordion extends StackPane {  
  
    public PanelWithAccordion() {  
        Accordion accordion = new Accordion();  
        TitledPane panel1 = new Panel1();  
        TitledPane panel2 = new Panel2();  
        accordion.getPanes().add(panel1);  
        accordion.getPanes().add(panel2);  
        this.getChildren().add(accordion);  
    }  
}  
  
public class Panel1 extends TitledPane {  
  
    public Panel1() {  
        this.setText("Painel 1");  
        this.setContent(new Label("Painel 1"));  
    }  
}
```



JavaFX – Exemplo: Accordion

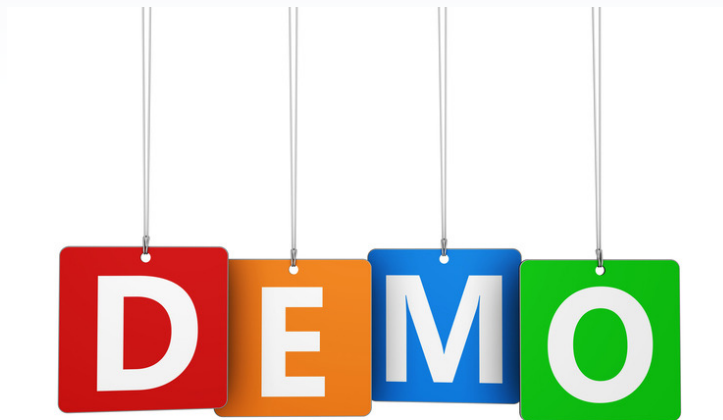
Objetivo: Ter dois Painéis 1 e 2, e alternar entre um e outro consoante a seleção.

```
public class PanelWithAccordion extends StackPane {  
  
    public PanelWithAccordion() {  
        Accordion accordion = new Accordion();  
        TitledPane panel1 = new Panel1();  
        TitledPane panel2 = new Panel2();  
        accordion.getPanes().add(panel1);  
        accordion.getPanes().add(panel2);  
        this.getChildren().add(accordion);  
    }  
}  
  
public class Panel1 extends TitledPane {  
  
    public Panel1() {  
        this.setText("Painel 1");  
        this.setContent(new Label("Painel 1"));  
    }  
}
```

1. Criar o **Accordion**
2. Criar objetos do tipo **TitledPane**.
Nota: É preciso definir o título (**setText**) e o conteúdo (**setContent**) dos **TitledPane**.
3. Associar cada **TitledPane** ao **Accordion**.

Para selecionar um **TitledPane** :
`accordion.setExpandedPane (painel2);`

JavaFX- Eventos : Exemplo



Menus em JavaFX

- ▶ JavaFX – Controlos II



JavaFX – Exemplo: Menus

Em JavaFX podem usar-se as seguintes classes para definir **menus**:

- **MenuBar**
 - **Menu**
 - **MenuItem**
 - **CheckMenuItem**
 - **RadioMenuItem**
 - **SeparatorMenuItem**
 - **ContextMenu**



JavaFX – Exemplo: Menus

Preparação do Layout

```
public class PanelWithMenu extends BorderPane {  
    public PanelWithMenu() {  
        MenuBar menuBar = new MenuBar();  
  
        Menu setupMenu = new Menu("Configurar");  
        MenuItem setupMenuA = new MenuItem("Adicionar A");  
        MenuItem setupMenuB = new MenuItem("Adicionar B");  
        setupMenu.getItems().addAll(setupMenuA, setupMenuB);  
  
        Menu showMenu = new Menu("Visualizar");  
        MenuItem showMenuA = new MenuItem("ver A");  
        CheckMenuItem showMenuB = new CheckMenuItem("ver B");  
        showMenu.getItems().addAll(showMenuA, new SeparatorMenuItem(), showMenuB);  
  
        Menu exitMenu = new Menu("Sair");  
        MenuItem closeMenu = new MenuItem("Fechar");  
        exitMenu.getItems().add(closeMenu);  
  
        menuBar.getMenus().addAll(setupMenu,  
                                   showMenu,  
                                   exitMenu);  
        this.setTop(menuBar);  
    }  
}
```



JavaFX – Exemplo: Menus

Preparação do funcionamento

Ex: Sair da aplicação através do **MenuItem** Fechar do **Menu** Sair

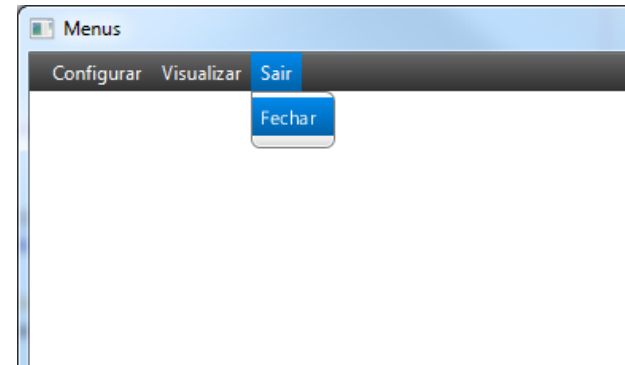
```
public class PanelWithMenu extends BorderPane {
    public PanelWithMenu() {
        MenuBar menuBar = new MenuBar();

        Menu setupMenu = new Menu("Configurar");
        MenuItem setupMenuA = new MenuItem("Adicionar A");
        MenuItem setupMenuB = new MenuItem("Adicionar B");
        setupMenu.getItems().addAll(setupMenuA, setupMenuB);

        Menu showMenu = new Menu("Visualizar");
        MenuItem showMenuA = new MenuItem("ver A");
        CheckMenuItem showMenuB = new CheckMenuItem("ver B");
        showMenu.getItems().addAll(showMenuA, new SeparatorMenuItem(), showMenuB);

        Menu exitMenu = new Menu("Sair");
        MenuItem closeMenu = new MenuItem("Fechar");
        exitMenu.getItems().add(closeMenu);
        closeMenu.setOnAction(
            e -> Platform.exit()
        );

        menuBar.getMenus().addAll(setupMenu,
                                   showMenu,
                                   exitMenu);
        this.setTop(menuBar);
    }
}
```



JavaFX – Exemplo: Menus

Objetivo: Alternar entre painéis, consoante a opção do **Menu** selecionada.

Solução: Usar o layout **BorderPane** para a Janela. Definir dois painéis (**panelA** e **panelB**), colocar na zona central o **panelA** ou o **panelB** consoante a opção do menu.

```
public class PanelWithMenu extends BorderPane {
```

```
    private final PanelA panelA;  
    private final PanelB panelB;
```

```
    public PanelWithMenu() {
```

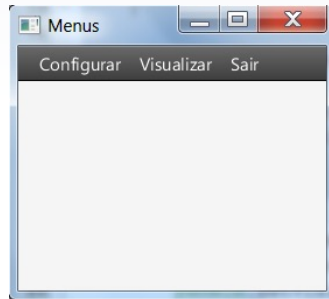
```
        this.panelA = new PanelA();  
        this.panelB = new PanelB();
```

```
        Menu setupMenu = new Menu("Configurar");  
        MenuItem setupMenuA = new MenuItem("Adicionar A");  
        MenuItem setupMenuB = new MenuItem("Adicionar B");
```

```
        MenuBar menuBar = new MenuBar();
```

```
        ...  
        this.setTop(menuBar);
```

```
    }  
}
```



Preparação do layout:

1. Usar o layout **BorderPane** para a Janela.
2. Declarar dois painéis como atributos e criá-los no construtor.
3. Criar o Menu **Configurar** e os **MenuItem**
4. Criar o **MenuBar** e associá-lo ao topo

JavaFX – Exemplo: Menus

```
public class PaineWithMenu extends BorderPane {

    private final PanelA panelA;
    private final PanelB panelB;

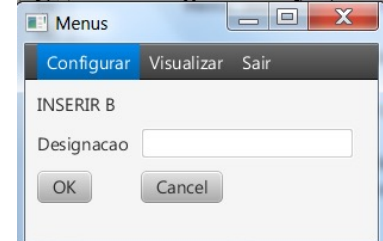
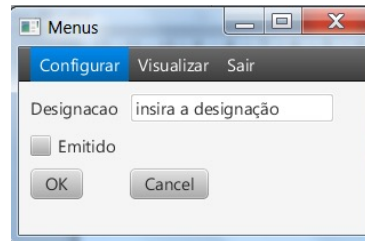
    public PaineWithMenu() {
        this.panelA = new PanelA();
        this.panelB = new PanelB();
        MenuBar menuBar = new MenuBar();
        this.setTop(menuBar);
        Menu setupMenu = new Menu("Configurar");
        MenuItem setupMenuA = new MenuItem("Adicionar A");
        MenuItem setupMenuB = new MenuItem("Adicionar B");
        setupMenuA.setOnAction(e -> showPanelA());
        setupMenuB.setOnAction(e -> showPanelB());
        // (...)
    }

    public final void showPanelA() {
        this.setCenter(panelA);
    }

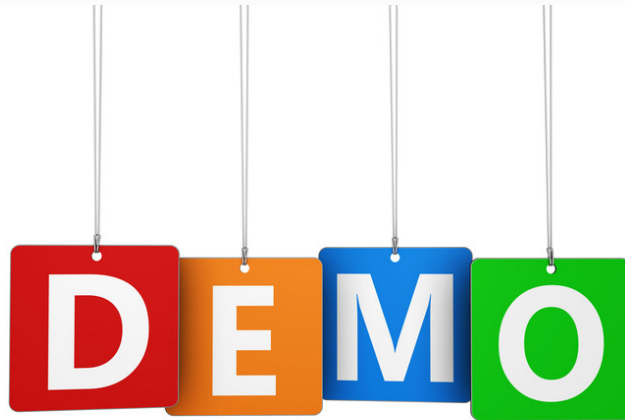
    public final void showPanelB() {
        this.setCenter(panelB);
    }
}
```

Preparação do Funcionamento:

1. Adicionar um evento ao **MenuItem setupMenuA**, que irá mostrar o **panelA**.
2. Adicionar um evento ao **MenuItem setupMenuB**, que irá mostrar o **panelB**.

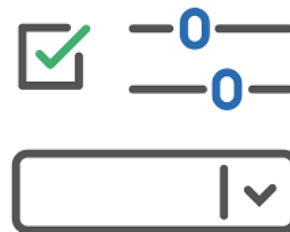


JavaFX- Eventos : Exemplo



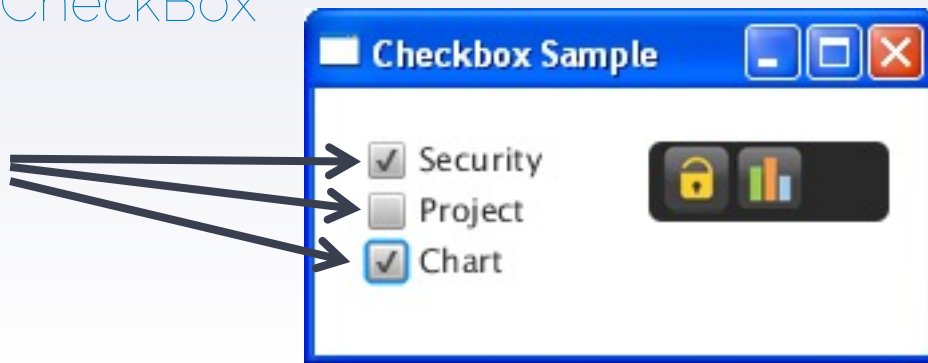
CheckBox e RadioButton

- ▶ JavaFX – Controlos II



JavaFX – Exemplo: CheckBox

CheckBox



- Uma **CheckBox** pode estar em três estados:

- Determinado e selecionada.

INDETERMINATE = false

SELECTED = false



- Determinada e não selecionada.

INDETERMINATE = false

SELECTED = true



- Indeterminado

INDETERMINATE = true

SELECTED = true/false



JavaFX – Exemplo: CheckBox



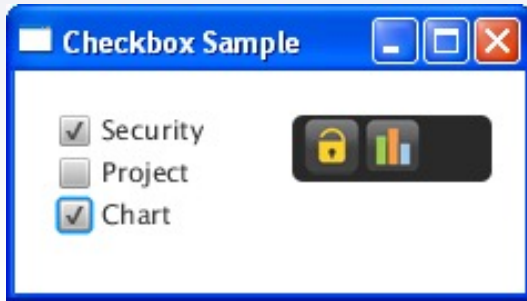
Objetivo: Mostrar um botão com um icon sempre que a **CheckBox** sassociada está selecionada.

Passo 1

1. Definir os vários arrays
2. Definir as imagens a mostrar
3. Definir as **CheckBox**

```
String[] names = new String[]{"Security", "Project", "Chart"};
Image[] images = new Image[names.length];
ImageView[] icons = new ImageView[names.length];
CheckBox[] cbs = new CheckBox[names.length];
for (int i = 0; i < names.length; i++) {
    Image image = images[i]
        = new Image(getClass().getResourceAsStream(names[i] + ".png"));
    ImageView icon = icons[i] = new ImageView();
    CheckBox cb = cbs[i] = new CheckBox(names[i]);
}
```

JavaFX – Exemplo: CheckBox

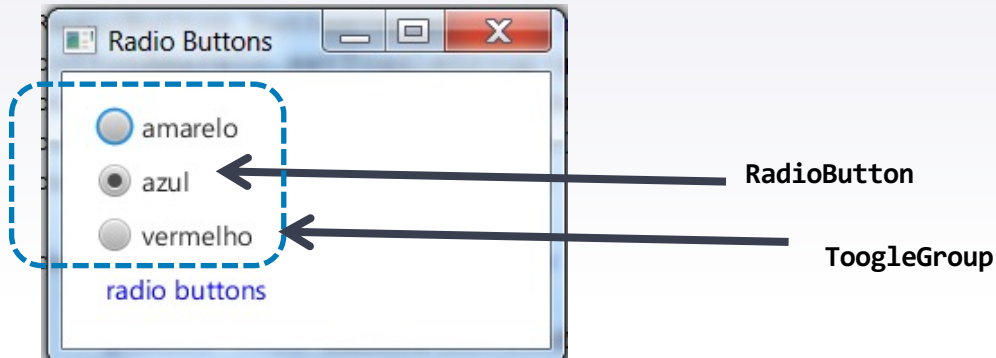


Passo 2

Definir a ação a realizar quando se altera a seleção de uma **CheckBox** recorrendo à utilização de “listeners” da propriedade **selected**

```
for (int i = 0; i < names.length; i++) {  
    CheckBox cb = cbs[i];  
    ImageView icon = icons[i];  
    Image image = images[i];  
    cb.selectedProperty().addListener(new ChangeListener<Boolean>() {  
        public void changed(ObservableValue ov,  
                            Boolean old_val, Boolean new_val) {  
            icon.setImage(new_val ? image : null);  
        }  
    });  
}
```

JavaFX – Exemplo: Radio Buttons



- Um **RadioButton** pode estar selecionado ou não selecionado.
- De forma a podermos ter um grupo de botões a trabalhar “sincronizados” só um poderá estar selecionado de cada vez
 - para implementarmos escolhas exclusivas) temos que definir um **ToggleGroup**.

JavaFX – Exemplo: Radio Buttons



Objetivo: Ter uma aplicação que em função do **RadioButton** selecionado, muda a cor do texto apresentado.

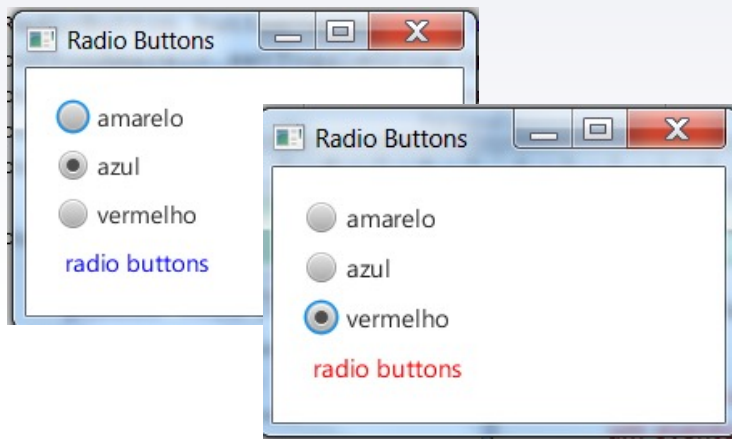
Passo 1

1. Definir os **RadioButton**
2. Definir o **ToggleGroup**
3. Associar os **RadioButton** ao **ToggleGroup**
4. Criar o **Text**

```
RadioButton yellowButton = new RadioButton("amarelo");
RadioButton blueButton = new RadioButton("azul");
RadioButton redButton = new RadioButton("vermelho");
final ToggleGroup group = new ToggleGroup();
yellowButton.setToggleGroup(group);
blueButton.setToggleGroup(group);
redButton.setToggleGroup(group);

final Text text = new Text("radio buttons");
text.setFill(Color.BLUE);
buttonAzul.setSelected(true);
```

JavaFX – Exemplo: Radio Buttons



Passo 2

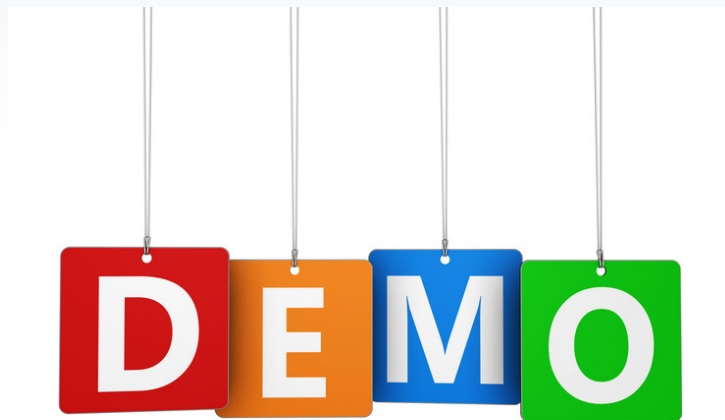
Definir as ações a realizar quando existe um evento nos botões. Uma ação para cada botão.

```
yellowButton.setOnAction(e -> texto.setFill(Color.YELLOW));
```

```
blueButton.setOnAction(e -> texto.setFill(Color.BLUE));
```

```
redButton.setOnAction(e -> texto.setFill(Color.RED));
```

JavaFX- Eventos : Exemplo



Resumindo

▶ **ListView e ComboBox**

- ▶ Permitem apresentar e manipular coleções de elementos.
- ▶ A **ListView** permite seleção Múltipla ou Simples.
- ▶ A **ComboBox** só permite a seleção de um elemento (Simples).

▶ **Tab**

- ▶ Através de um controlo **TabPane**, podemos implementar a alternância entre vários painéis.
- ▶ Associando um painel a um **Tab** e por sua vez os vários **Tabs** a um **TabPane**.

▶ **Accordion**

- ▶ Através de um controlo **Accordion**, também é possível implementar a alternância entre painéis.
- ▶ Criando os vários **TitledPane** e associando-os ao **Accordion**.

▶ **Menu**

- ▶ É possível definir uma hierarquia de menus e submenus, usando as classes **MenuBar**, **Menu**, **MenuItem**, **CheckMenuItem**, **RadioMenuItem**, **SeparatorMenuItem**.

▶ **CheckBox**

- ▶ Através de um controlo **CheckBox**, podemos implementar a seleção de uma opção.

▶ **RadioButton**

- ▶ Para implementar a escolha exclusiva através de **RadioButton**, temos que definir um **ToggleGroup** e associar cada **RadioButton** ao grupo criado.

Leitura Complementar

- ▶ Chapter 4 – Layouts and UI Controls Pgs 101 a 108

<http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/combobox.htm#BABJCCIB>

http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/menu_controls.htm#BABGHADI

https://docs.oracle.com/javafx/2/ui_controls/checkbox.htm

<http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/radiobutton.htm#BABBJBDA>

- ▶ Controlos UI

https://docs.oracle.com/javafx/2/ui_controls/overview.htm

