Programação Orientada por Objetos

JavaFX – Controlos II

Prof. José Cordeiro,

Prof. Cédric Grueau,

Prof. Laercio Júnior

Departamento de Sistemas e Informática

Escola Superior de Tecnologia de Setúbal – Instituto Politécnico de Setúbal

2019/2020

Módulo Controlos II do JavaFX

- Sessão 1 – Controlos ListView e ComboBox
- Sessão 2 – Controlos Tabs, Accordion e TitledPane
- Sessão 3 – Menus em JavaFX
- Sessão 4 – Controlos Checkbox e Radiobutton

Módulo 13 – JavaFX – Controlos II

SESSÃO 1 – LISTVIEW E COMBOBOX

JavaFX - Controlos

As classes para criar controlos encontram-se no pacote

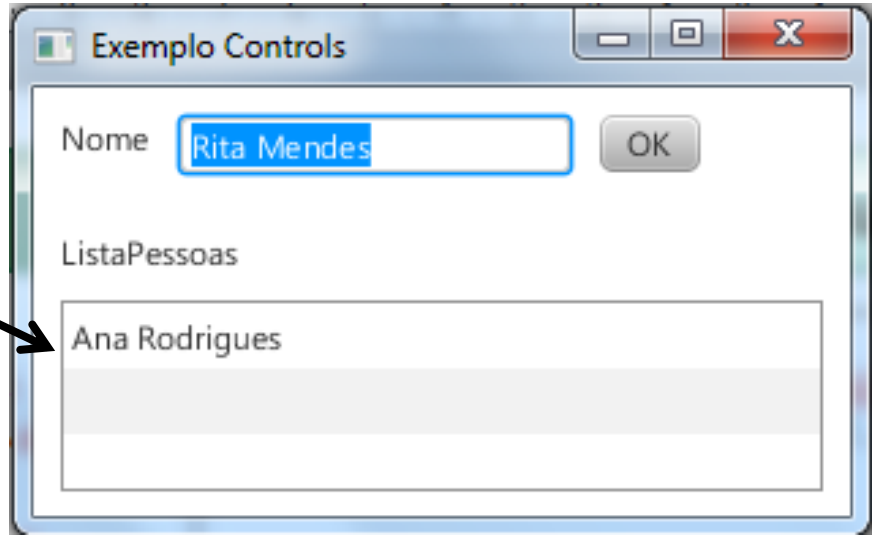
`javafx.scene.control`

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`**

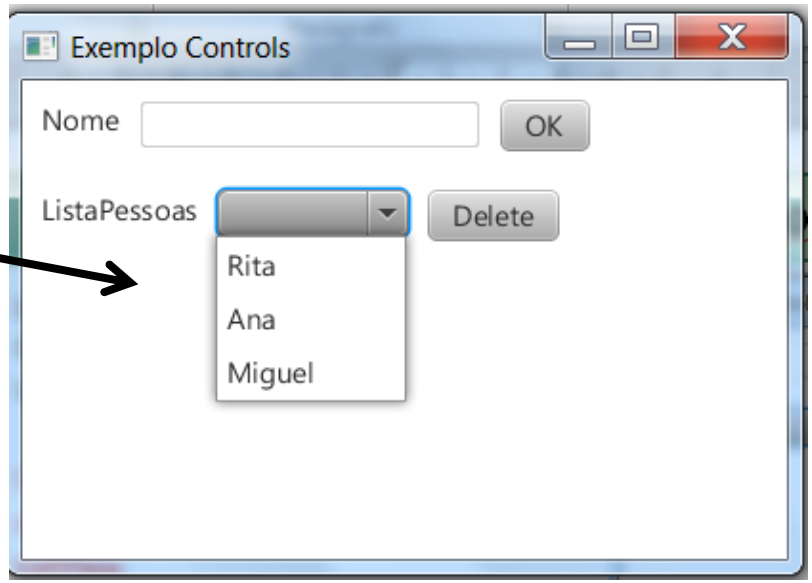
```
ListView<String> listaNomes= new ListView<>();

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

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

```
items.add(nome);
```

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**

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

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

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

```
items.add(nome);
```

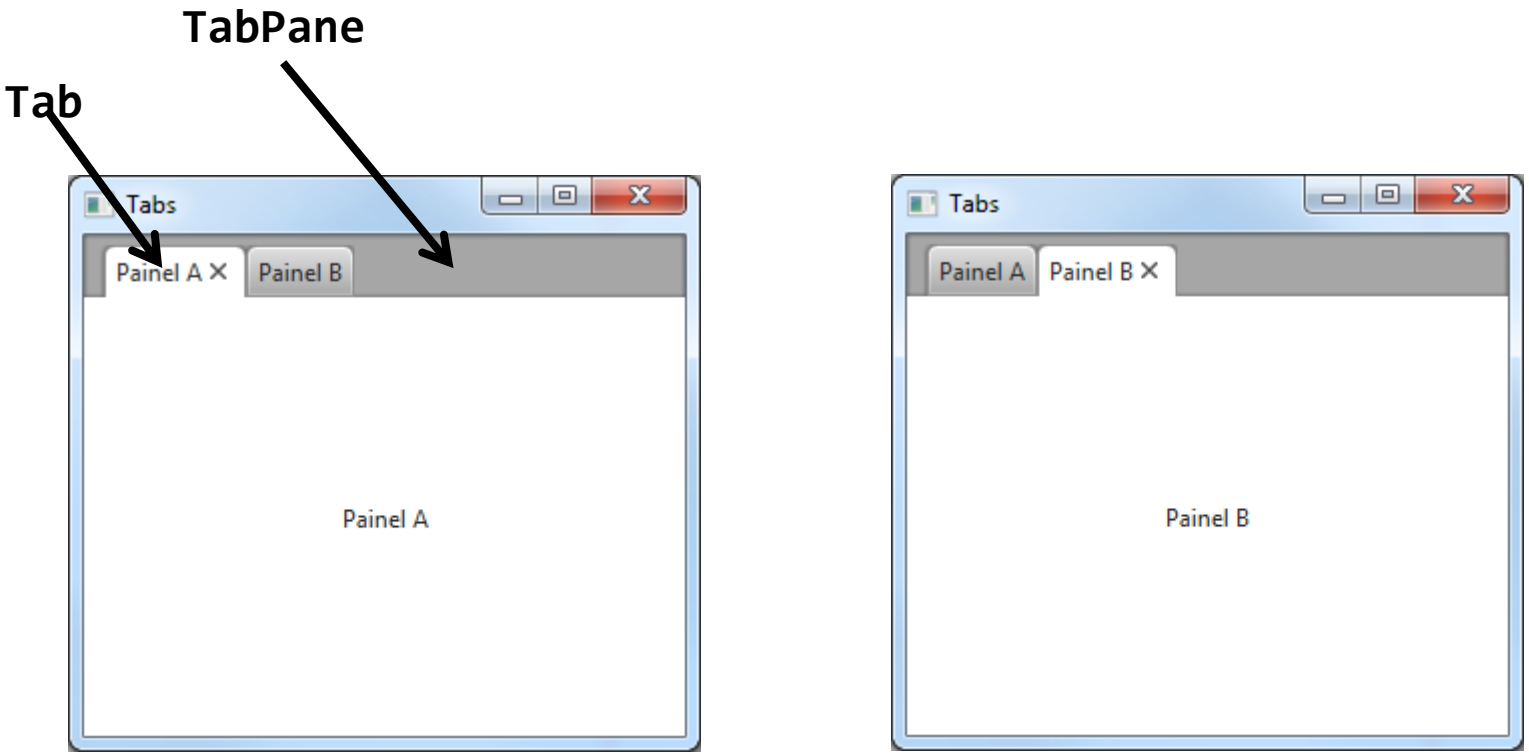
JavaFX- Eventos : Exemplo

Módulo 13 – JavaFX – Controlos II

SESSÃO 2 – TABS, ACCORDION E TITLEDPANE

JavaFX – Exemplo: Tab

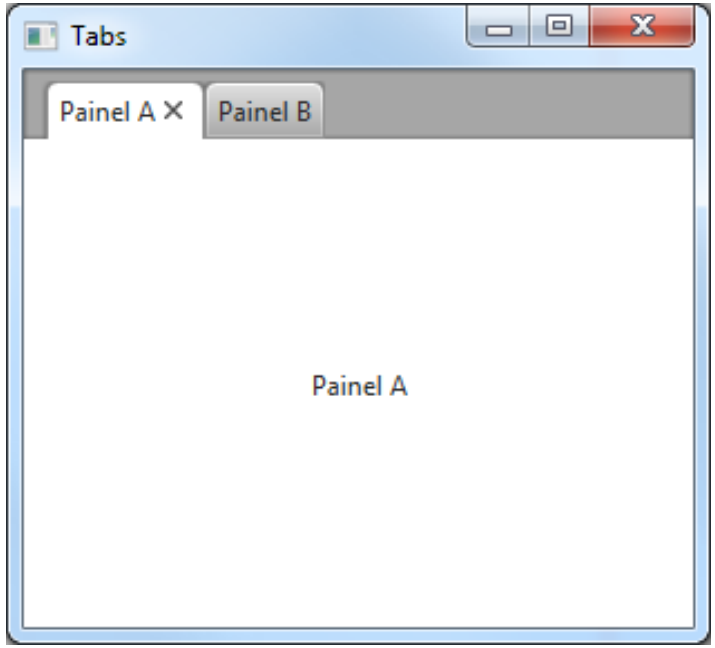
Uma forma simples de gerir a alternância entre vários painéis é através da utilização de um **TabPane** com vários **Tab** associados.



JavaFX – Exemplo: Tab

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

```
public class PainelComTab extends StackPane {

    public PainelComTab() {
        TabPane tabPane = new TabPane();
        Tab tabA= new Tab("Painel A");
        Tab tabB= new Tab("Painel B");
        tabA.setContent(new PainelA());
        tabB.setContent(new PainelB());
        tabPane.getTabs().addAll(tabA,tabB);
        this.getChildren().add(tabPane);
    }
}
```




```
public class PainelA extends StackPane {

    public PainelA() {
        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 PainelComTab extends StackPane {

    public PainelComTab() {
        TabPane tabPane = new TabPane();
        Tab tabA= new Tab("Painel A");
        Tab tabB= new Tab("Painel B");
        tabA.setContent(new PainelA());
        tabB.setContent(new PainelB());
        tabPane.getTabs().addAll(tabA,tabB);
        this.getChildren().add(tabPane);
    }
}


public class PainelA extends StackPane {

    public PainelA() {
        this.getChildren().add(new Label("Painel A"));
    }
}
```

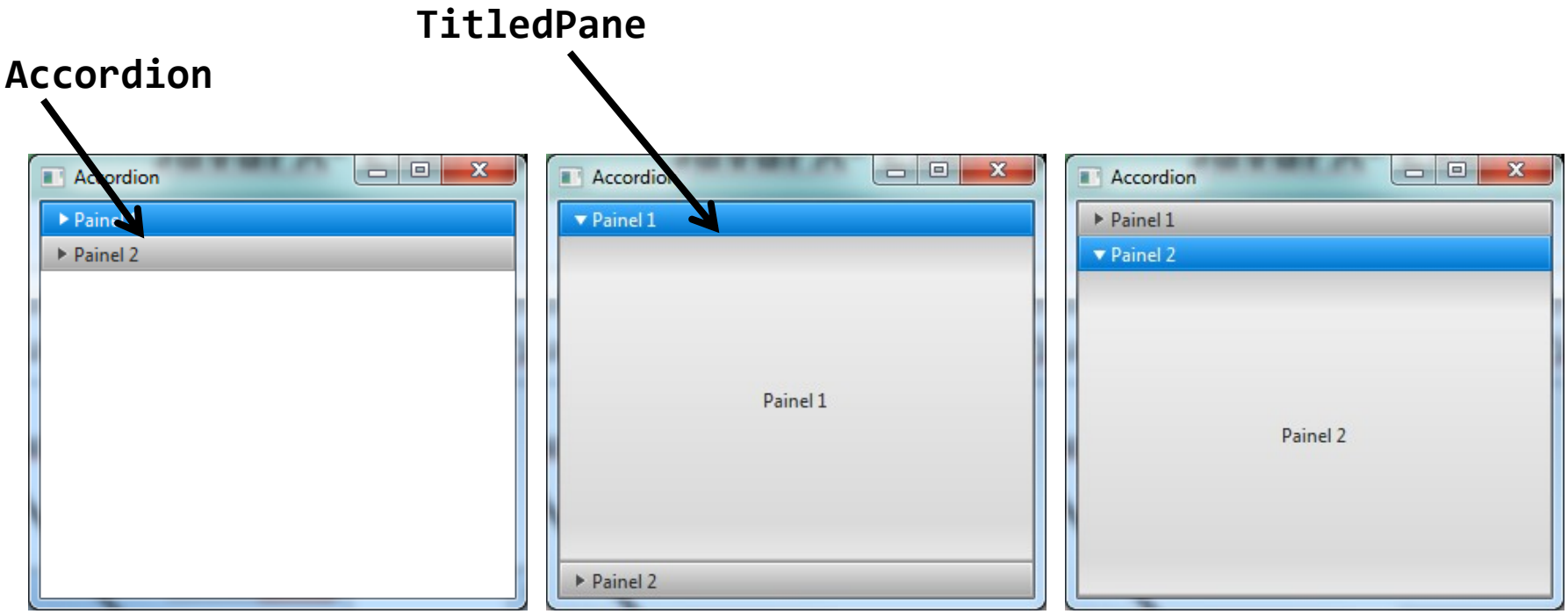
1. Criar a **TabPane**
2. Criar objetos do tipo **Tab**.
3. Associar a cada **Tab** o **Node** pretendido.
   Nota: **PainelA** e **PainelB** são classes derivadas de **Node**, normalmente painéis (**GridPane**, **HBox**, **VBox**, …).
4. Associar cada **Tab** ao **TabPane**

Para seleccionar um tab:

```
tabPane.getSelectionModel()
    .select(1);
```

JavaFX – Exemplo: Accordion

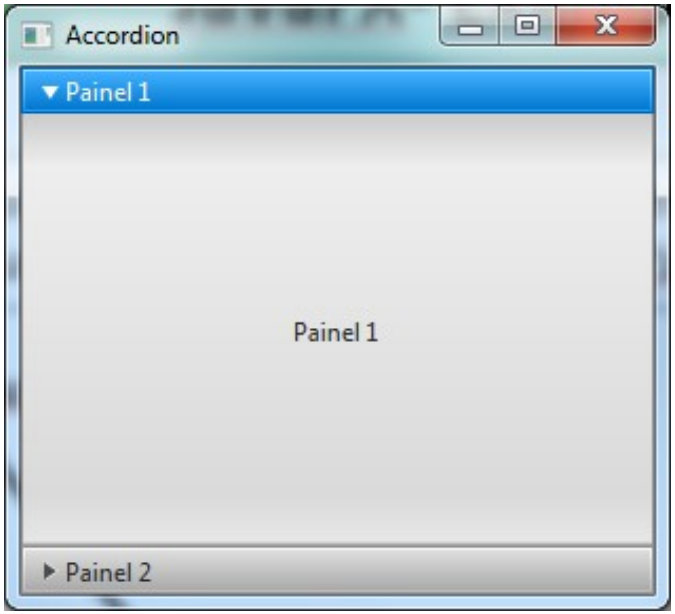
Outra forma simples de gerir a alternância entre vários painéis é através da utilização de um **`Accordion`** com vários **`TitledPane`** associados.



JavaFX – Exemplo: Accordion

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

```
public class PainelComAccordion extends StackPane {

    public PainelComAccordion() {
        Accordion accordion = new Accordion();
        TitledPane painel1 = new Painel1();
        TitledPane painel2 = new Painel2();
        accordion.getPanes().add(painel1);
        accordion.getPanes().add(painel2);
        this.getChildren().add(accordion);
    }
}
```




```
public class Painel1 extends TitledPane {

    public Painel1() {
        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 PainelComAccordion extends StackPane {

    public PainelComAccordion() {
        Accordion accordion = new Accordion();
        TitledPane painel1 = new Painel1();
        TitledPane painel2 = new Painel2();
        accordion.getPanes().add(painel1);
        accordion.getPanes().add(painel2);
        this.getChildren().add(accordion);
    }
}


public class Painel1 extends TitledPane {

    public Painel1() {
        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

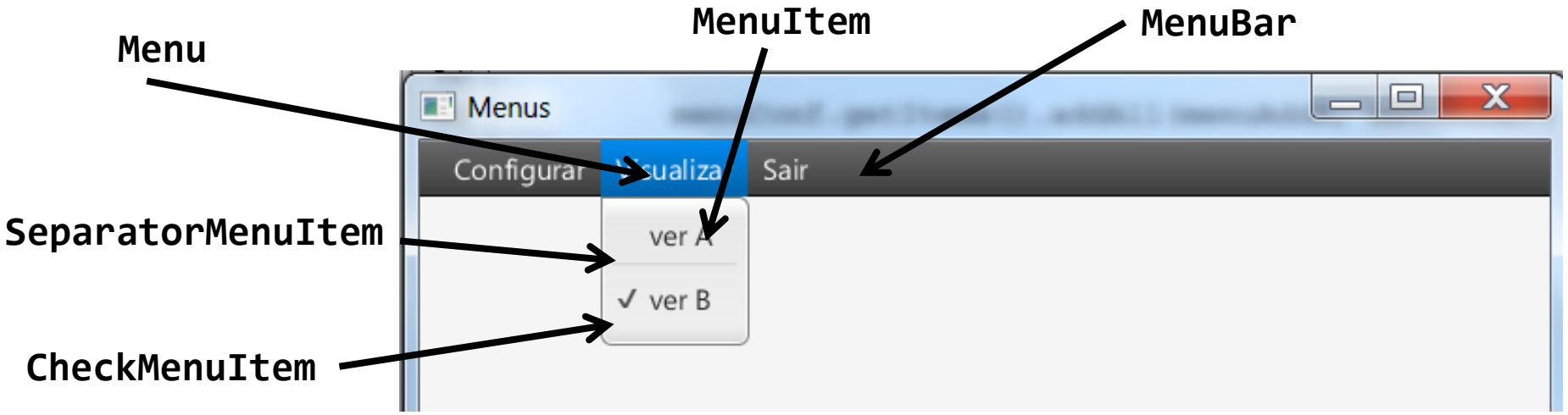
Módulo 13 – JavaFX – Controlos II

SESSÃO 3 – MENUS EM JAVAFX

JavaFX – Exemplo: Menus

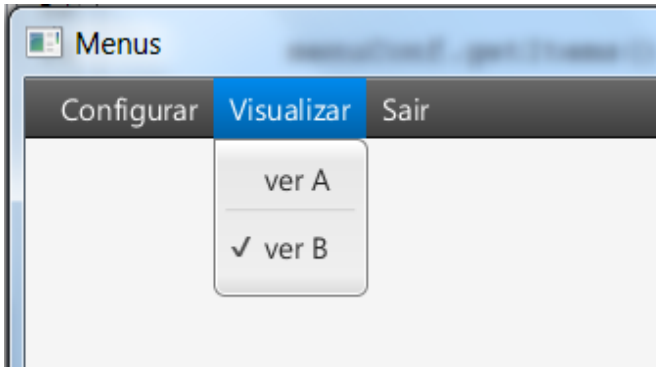
Em JavaFX podem usar-se as seguintes classes para definir **menus**:

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



Preparação do Layout

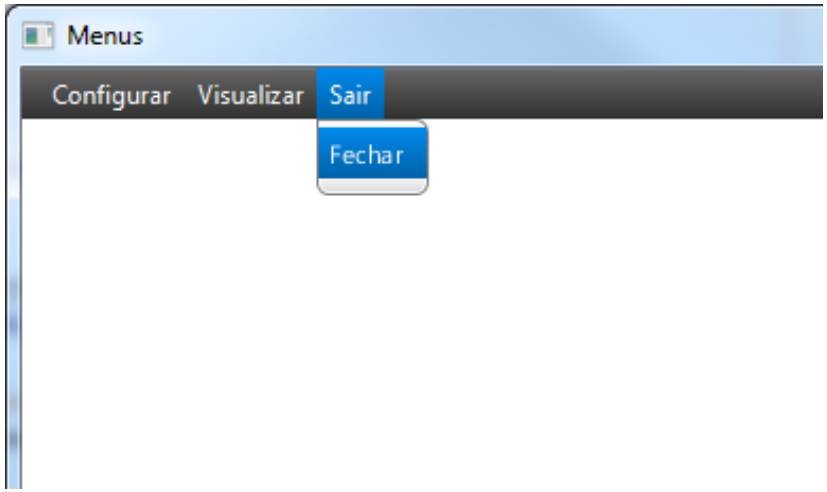
```
public class PainelComMenu extends BorderPane {
    public PainelComMenu() {
        MenuBar menuBar = new MenuBar();

        Menu menuConfigurar = new Menu("Configurar");
        MenuItem menuConfigurarA = new MenuItem("Adicionar A");
        MenuItem menuConfigurarB = new MenuItem("Adicionar B");
        menuConfigurar.getItems().addAll(menuConfigurarA, menuConfigurarB);

        Menu menuVisualizar = new Menu("Visualizar");
        MenuItem menuVerA = new MenuItem("ver A");
        CheckMenuItem menuVerB = new CheckMenuItem("ver B");
        menuVisualizar.getItems().addAll(menuVerA,new SeparatorMenuItem(), menuVerB);

        Menu menuSair = new Menu("Sair");
        MenuItem menuFechar = new MenuItem("Fechar");
        menuSair.getItems().add(menuFechar);

        menuBar.getMenus().addAll(menuConfigurar,
                menuVisualizar,
                menuSair);
        this.setTop(menuBar);
    }
}
```



Preparação do funcionamento

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

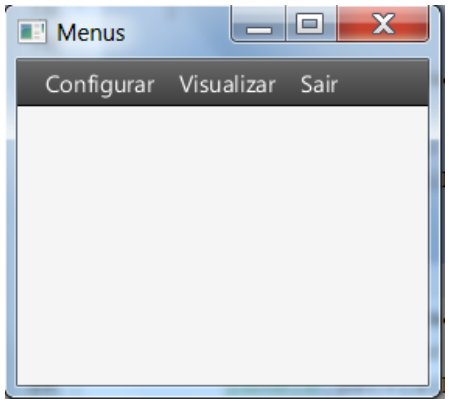
```
public class PainelComMenu extends BorderPane {
    public PainelComMenu() {
        MenuBar menuBar = new MenuBar();

        Menu menuConfigurar = new Menu("Configurar");
        MenuItem menuConfigurarA = new MenuItem("Adicionar A");
        MenuItem menuConfigurarB = new MenuItem("Adicionar B");
        menuConfigurar.getItems().addAll(menuConfigurarA, menuConfigurarB);

        Menu menuVisualizar = new Menu("Visualizar");
        MenuItem menuVerA = new MenuItem("ver A");
        CheckMenuItem menuVerB = new CheckMenuItem("ver B");
        menuVisualizar.getItems().addAll(menuVerA,new SeparatorMenuItem(), menuVerB);

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

        menuBar.getMenus().addAll(menuConfigurar,
                menuVisualizar, menuSair);
        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 (**painelA** e **painelB**), colocar na zona central o **painelA** ou o **painelB** consoante a opção do menu.

```
public class PainelComMenu extends BorderPane {

  private final PanelA painelA;
  private final PanelB painelB;

  public PainelComMenu() {

    this.painelA = new PanelA();
    this.painelB = new PanelB();


    Menu menuConfigurar = new Menu("Configurar");
    MenuItem menuConfigurarA = new MenuItem("Adicionar A");
    MenuItem menuConfigurarB = new MenuItem("Adicionar B");

    MenuBar menuBar = new MenuBar();
        ...
    this.setTop(menuBar);
  }
}
```



Preparação do layout:

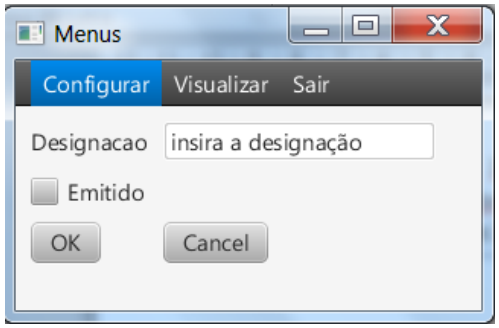
1. Usar o layout **BorderPane** para a Janela.
2. Declarar dois painéis como atributos e criá-los no construtor.
3. Criar o Menu Configurar e os MenuItem
4. Criar o MenuBar e associá-lo ao topo

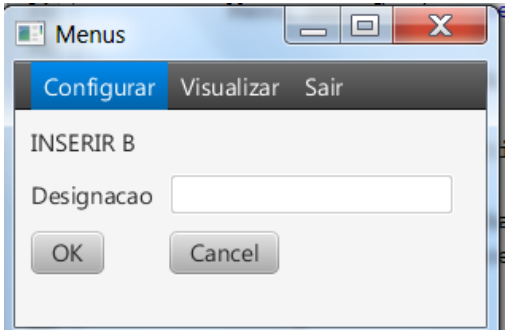
JavaFX – Exemplo: Menus

```
public class PainelComMenu extends BorderPane {

 private final PanelA painelA;
 private final PanelB painelB;

 public PainelComMenu() {
    this.painelA = new PanelA();
    this.painelB = new PanelB();
    MenuBar menuBar = new MenuBar();
    this.setTop(menuBar);
    Menu menuConfigurar = new Menu("Configurar");
    MenuItem menuConfigurarA = new MenuItem("Adicionar A");
    MenuItem menuConfigurarB = new MenuItem("Adicionar B");
    menuConfigurarA.setOnAction(e -> mostrarPainelA());
    menuConfigurarB.setOnAction(e -> mostrarPainelB());
    // (…)
    }

    public final void mostrarPainelA() {
        this.setCenter(painelA);
    }

    public final void mostrarPainelB() {
        this.setCenter(painelB);
    }
}
```

Preparação do Funcionamento:

1. Adicionar um evento ao **MenuItem menuConfigurarA**, que irá mostrar o **painelA**.
2. Adicionar um evento ao **MenuItem menuConfigurarB**, que irá mostrar o **painelB**.



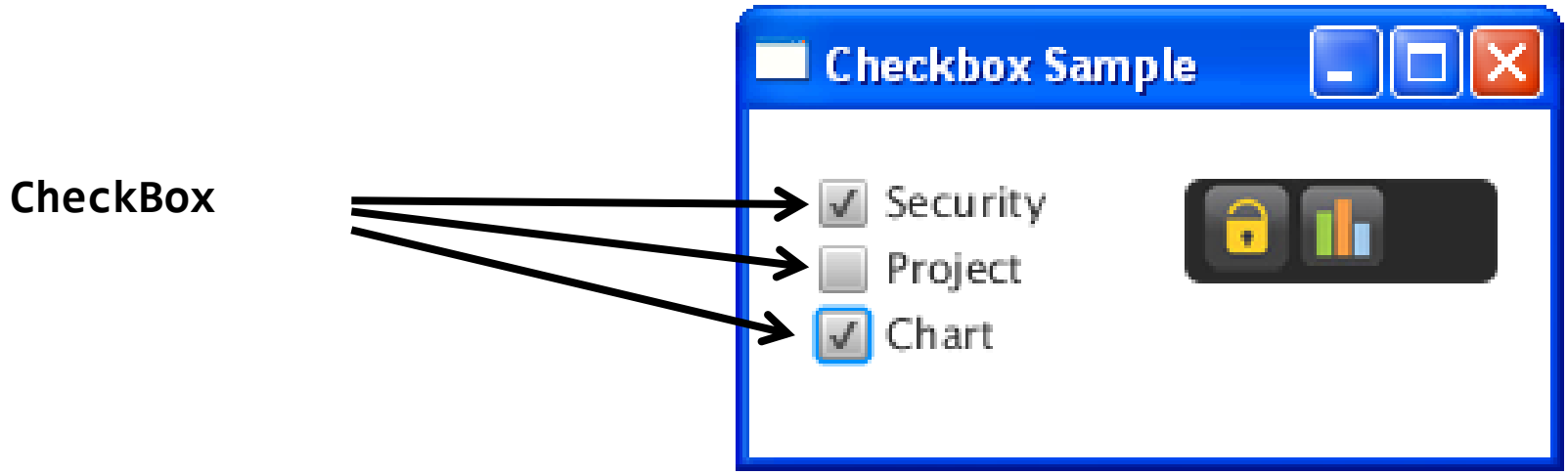


JavaFX- Eventos : Exemplo

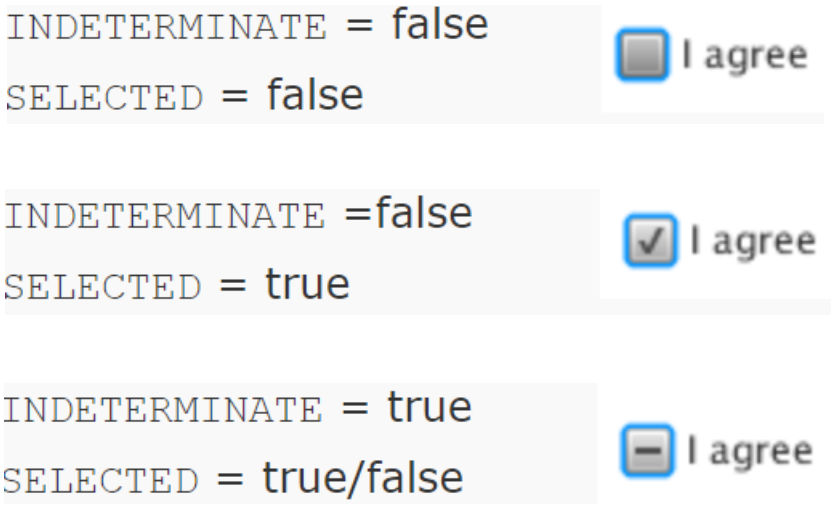
Módulo 13 – JavaFX – Controlos II

SESSÃO 4 – CHECKBOX E RADIOBUTTON

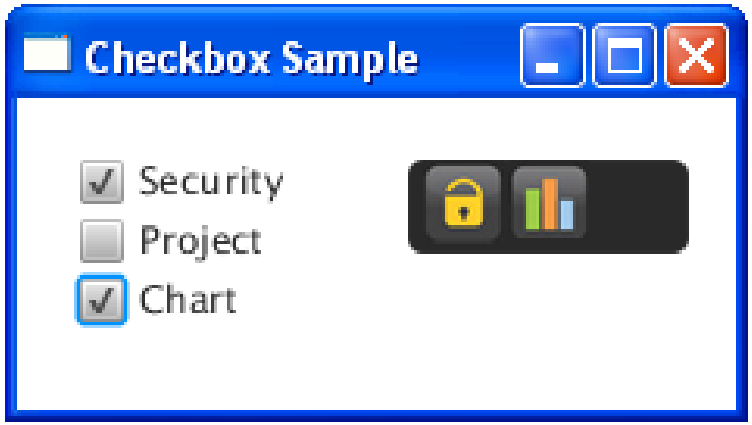
JavaFX – Exemplo: CheckBox



- Uma **CheckBox** pode estar em três estados:
  - Determinado e selecionada.
  - Determinada e não selecionada.
  - Indeterminado



JavaFX – Exemplo: CheckBox



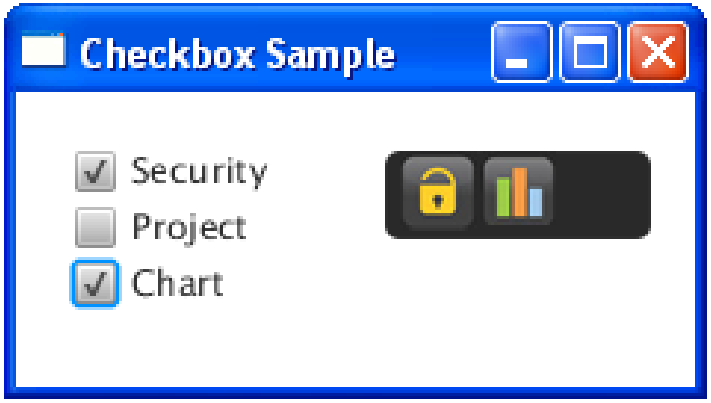
Objetivo: Mostrar um botão com um icon sempre que a **CheckBox** sassociada está selecionada.

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

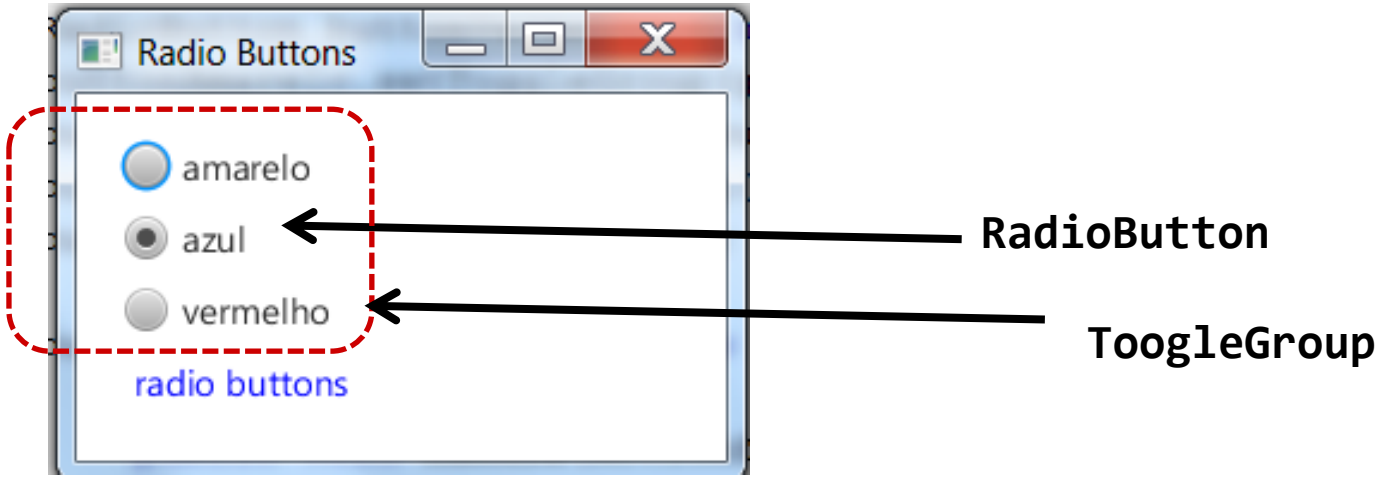


Passo 2

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 **`ToogleGroup`**.

JavaFX – Exemplo: Radio Buttons

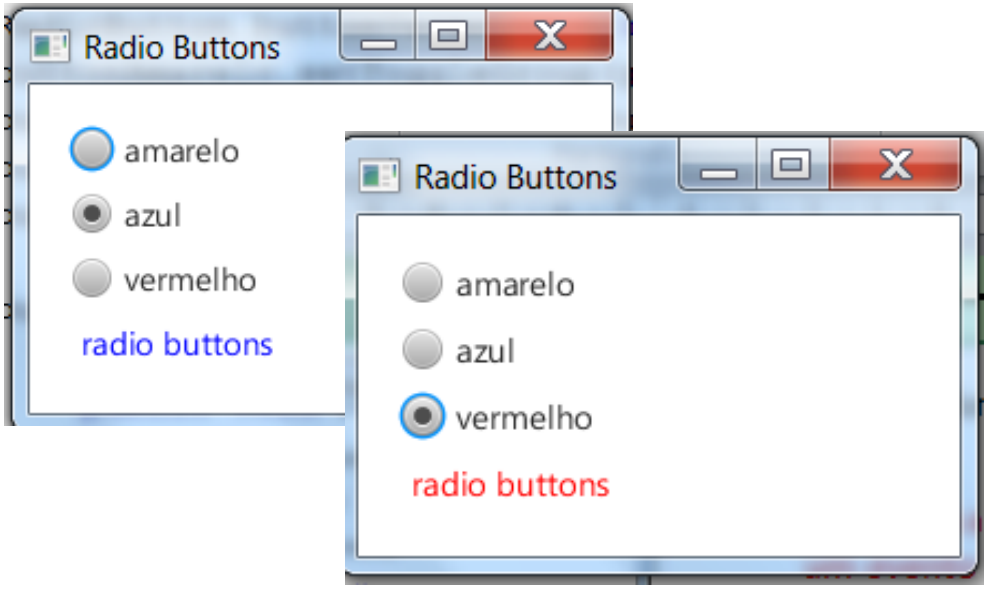
Objetivo: Ter uma aplicação que em função do **RadioButton** selecionado, muda a cor do texto apresentado.

Passo 1

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

```
RadioButton buttonAmarelo = new RadioButton("amarelo");
RadioButton buttonAzul = new RadioButton("azul");
RadioButton buttonVermelho = new RadioButton("vermelho");
final ToggleGroup group = new ToggleGroup();
buttonAmarelo.setToggleGroup(group);
buttonAzul.setToggleGroup(group);
buttonVermelho.setToggleGroup(group);

final Text texto = new Text("radio buttons");
texto.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.

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

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

buttonVermelho.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/combo-box.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/radio-button.htm#BABBJBDA

Controlos UI

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