

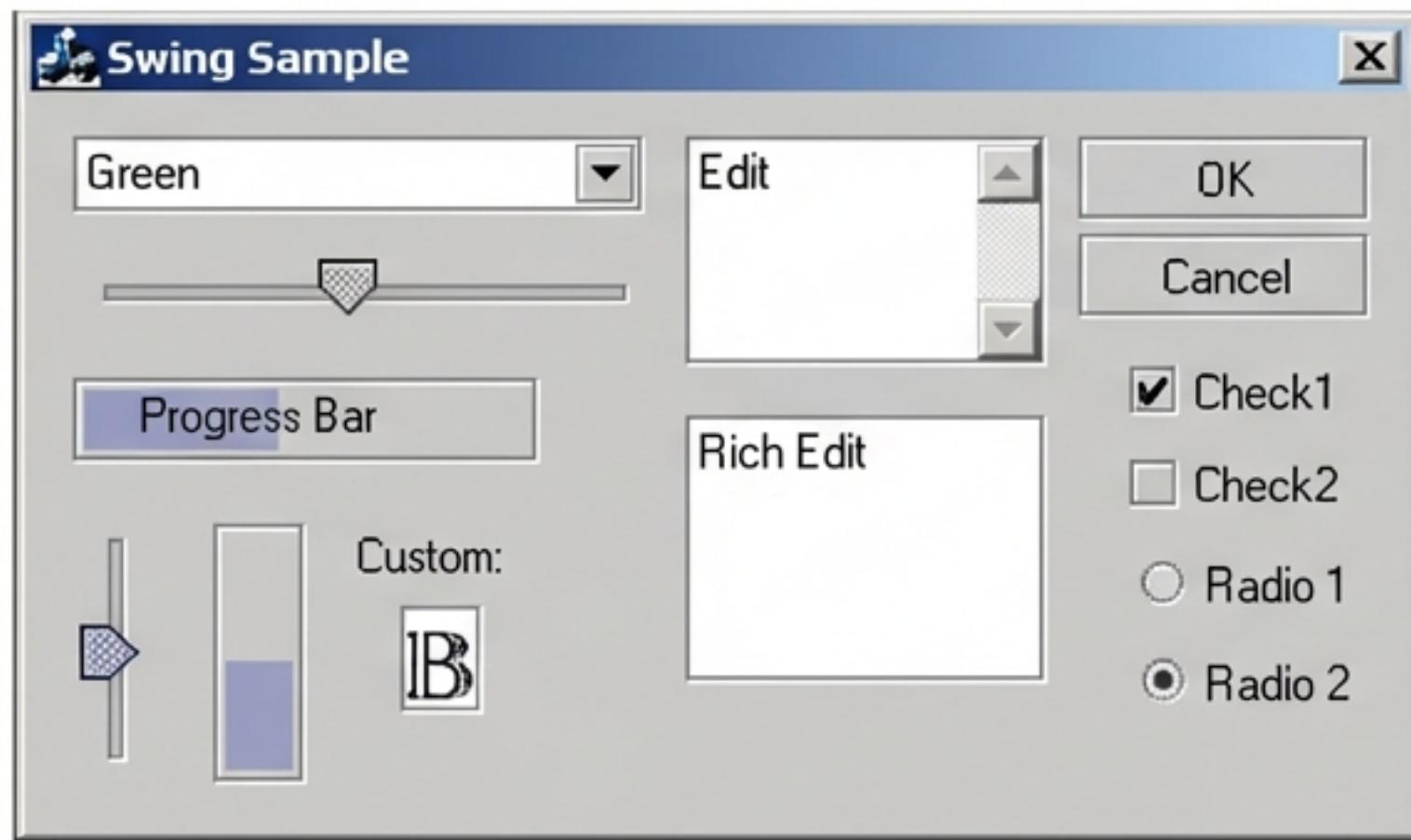
Dominando JavaFX: Criação de Interfaces Gráficas Modernas

Um guia prático da
arquitetura ao deploy



A Evolução da Interface Java: Por que JavaFX?

O Passado (Swing/AWT)



- Componentes pesados
- Dependentes do Sistema Operacional
- Visual datado

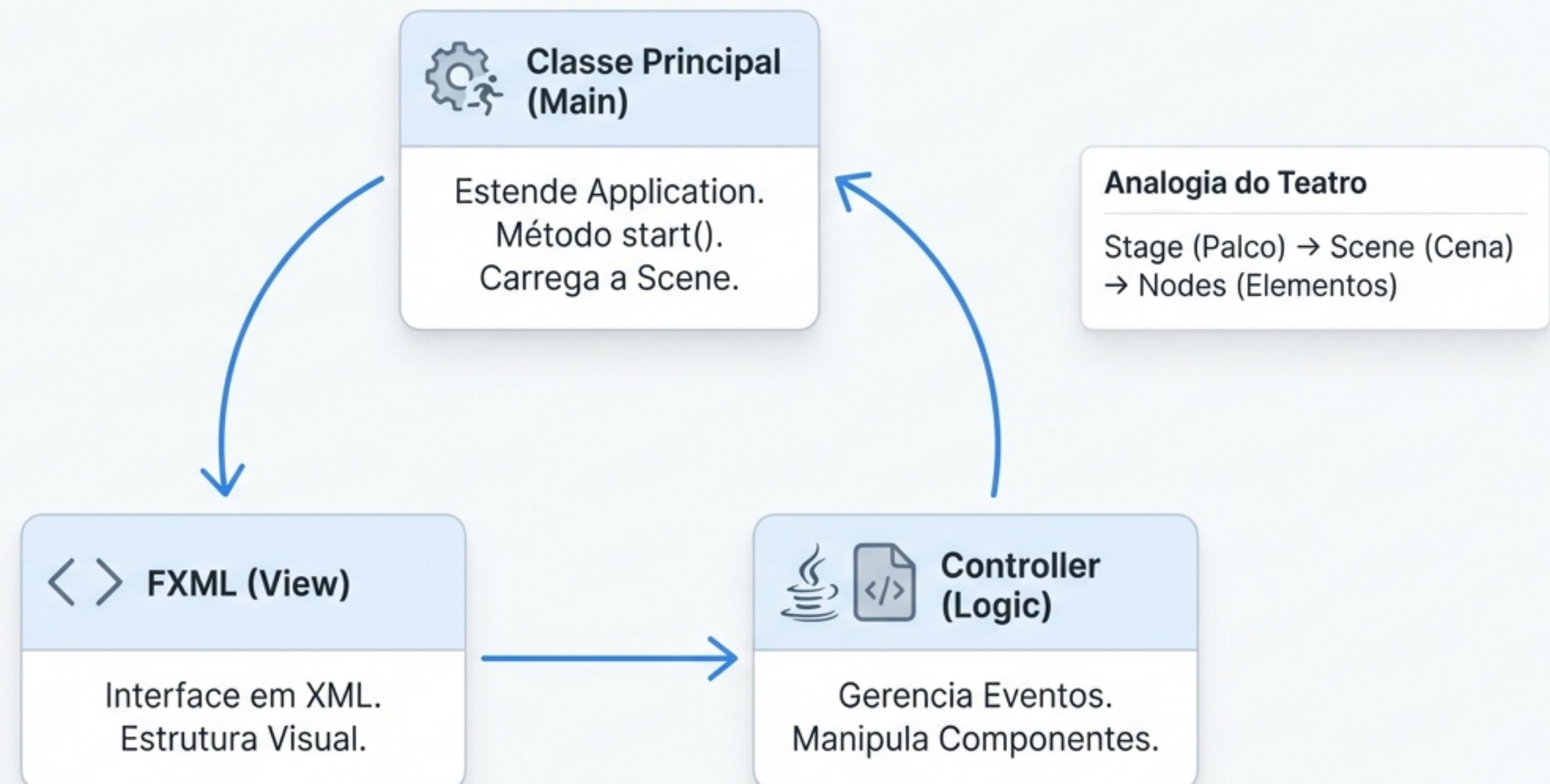
O Presente (JavaFX)



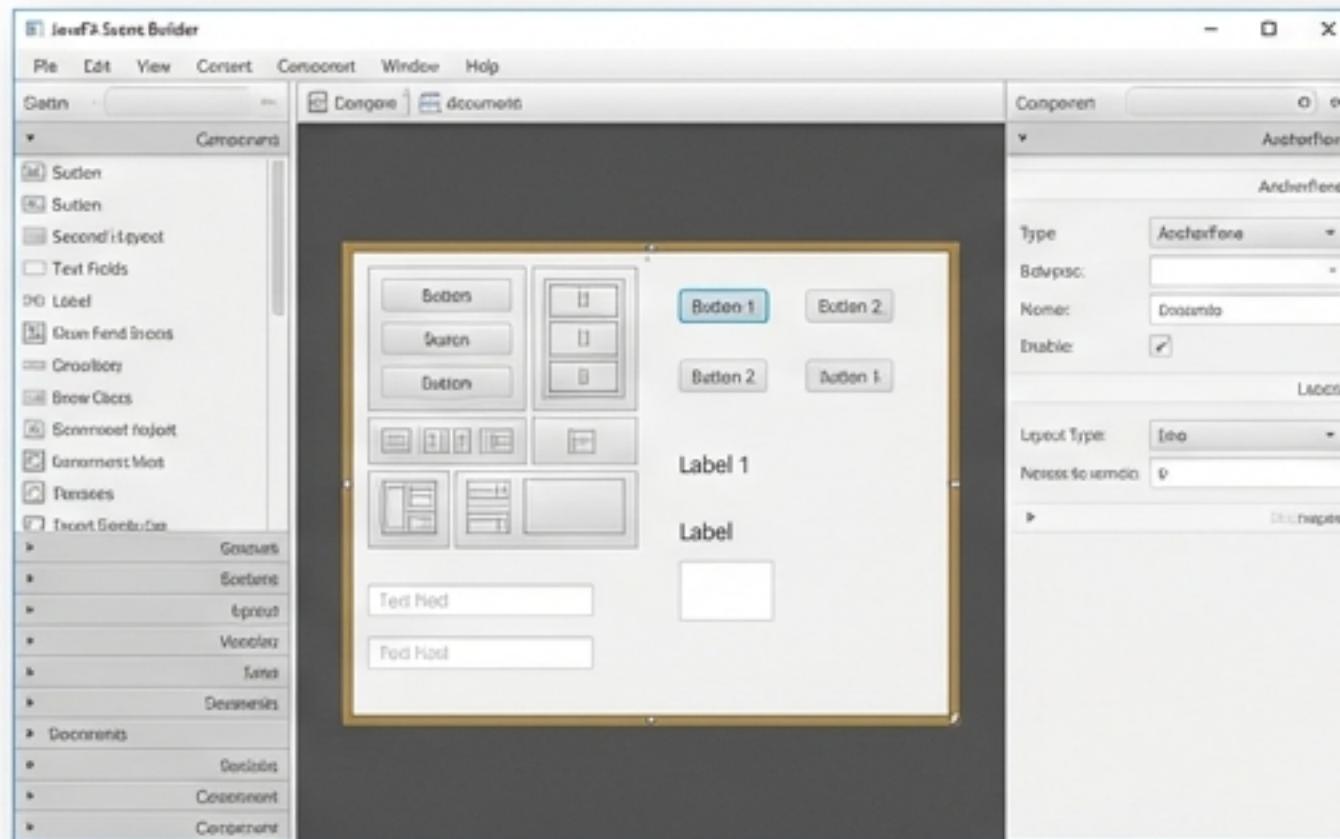
- Separação entre Design (FXML) e Lógica
- Estilização via CSS (Web-like)
- Aceleração por Hardware

“O JavaFX é a plataforma moderna do Java... substituindo o antigo Swing.”

A Arquitetura de uma Aplicação JavaFX



Scene Builder: A Ponte entre Design e Código



Gera Automaticamente

documento.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<AnchorPane
    <AnchorPane xmlns="http://javafx.com/javafx" xmlns:fx="http://javafx.com/fxml">
        <AnchorPane>
            <Label text="Label 1"/>
            <Label text="Label 2"/>
            <Button text="Button 1"/>
            <Button text="Button 2"/>
            <Button text="Button 3"/>
            <Button text="Button 4"/>
            <Label text="Label 3"/>
            <Label text="Label 4"/>
            <Textfield/>
            <Textfield/>
        </AnchorPane>
    </AnchorPane>
</AnchorPane>
```

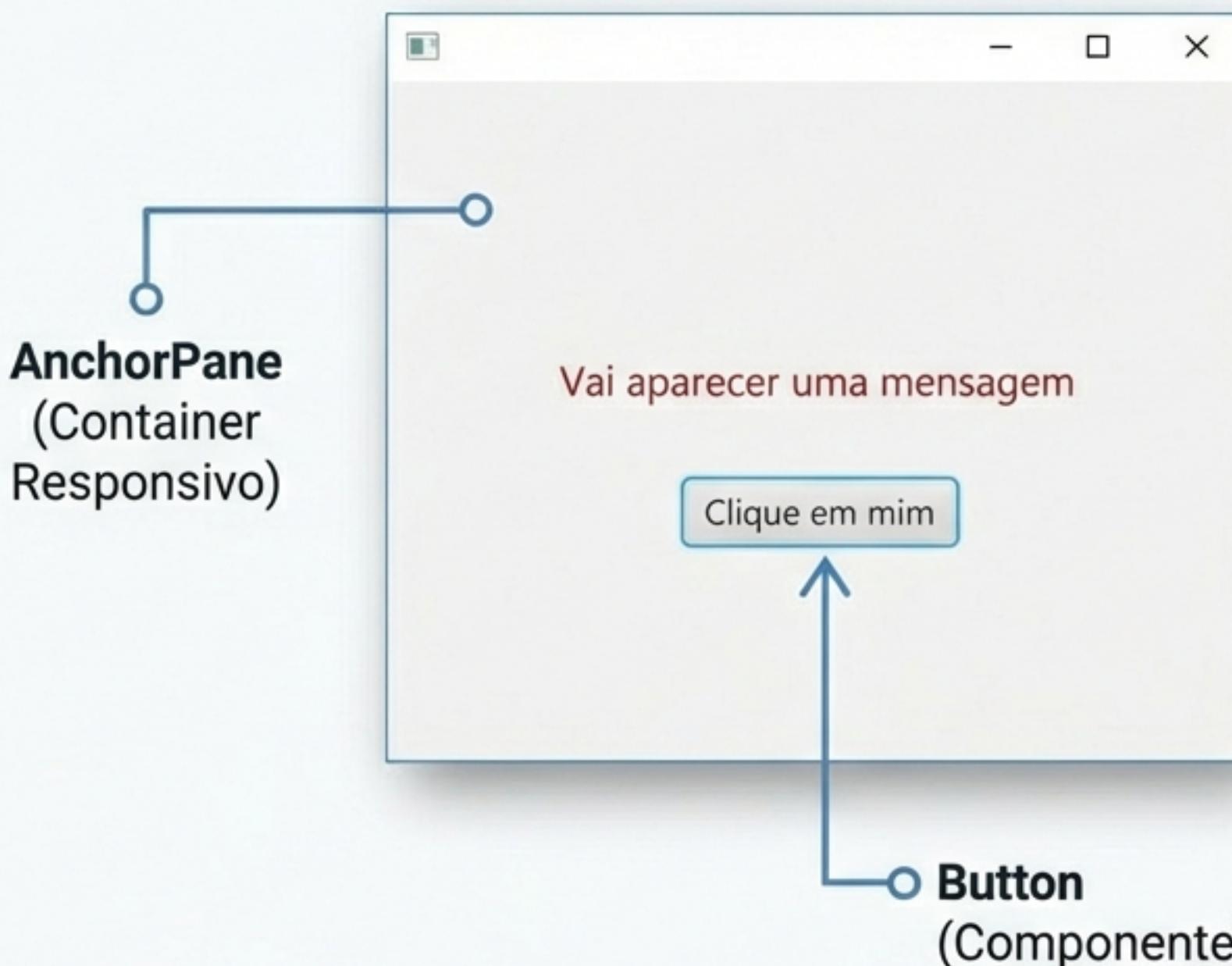
1. Desenhar a tela no Scene Builder (Drag & Drop)

2. Salvar o arquivo .fxml

3. Vincular ao Controller no NetBeans

Pro Tip: Essa ferramenta elimina a necessidade de escrever XML manualmente.

Primeiros Passos: Container e Identificação



O Container (AnchorPane)

Painel onde os componentes são 'ancorados' nas bordas. Ideal para layouts responsivos.

Identificação (*Critical*)

Para o Java 'enxergar' o visual, precisamos de um ID.

No Scene Builder: Code > fx:id

Exemplo: fx:id = 'buttonClick'

Injeção de Dependência: Conectando FXML ao Java

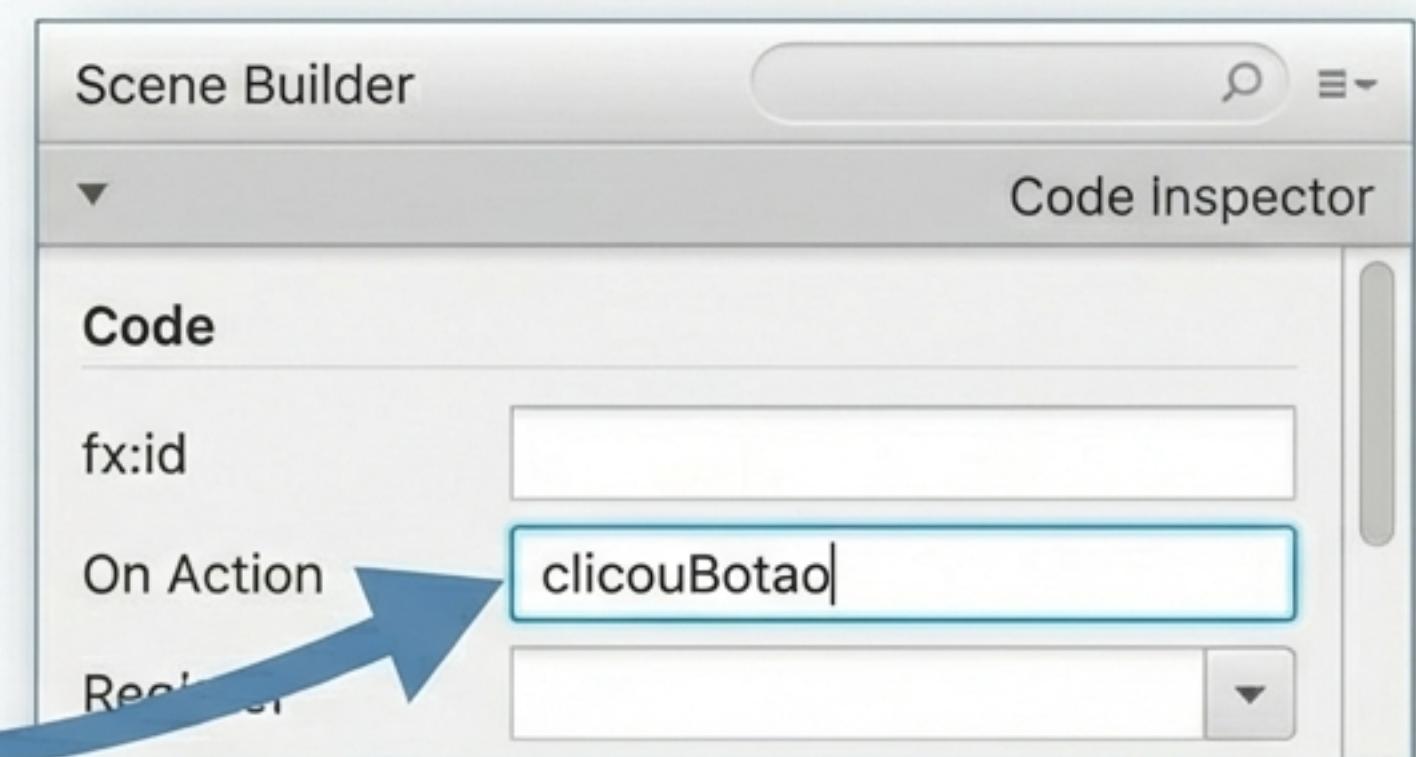
```
public class FXMLDocumentController implements Initializable {  
    @FXML ← Injeta o componente do FXML aqui  
    private Button btnClick;  
    @FXML ← Interface de inicialização  
    private Label lblMensagem;  
  
    @Override  
    public void initialize(URL url, ResourceBundle rb) {  
        // Executado ao carregar a tela  
    }  
}
```

Tratamento de Eventos: O Método onAction

The Code

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

The Tool



Ao vincular o método no campo **On Action**, o clique do usuário dispara a lógica Java imediatamente.

Feedback ao Usuário: Trabalhando com Alertas

```
Alert alert = new Alert(Alert.AlertType.INFORMATION);
alert.setTitle("Título do Alerta");
alert.setHeaderText("Cabeçalho");
alert.setContentText("Você clicou no botão!");
alert.show();
```

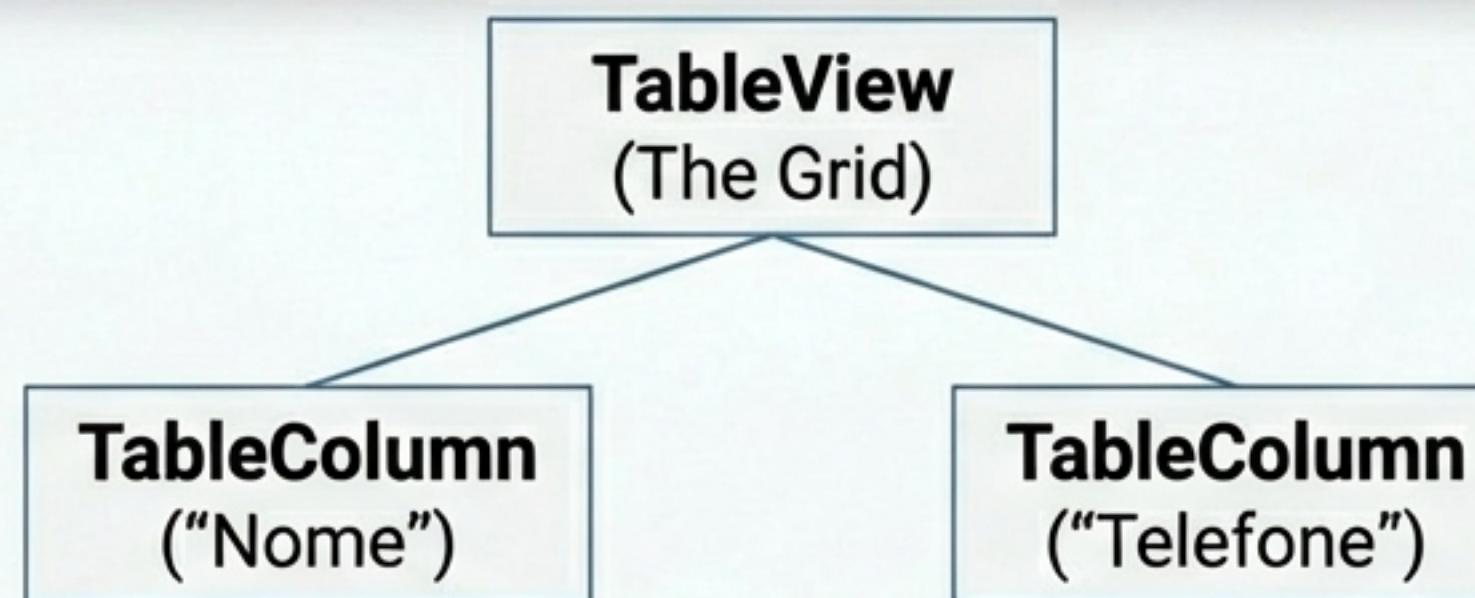
Alert Types

-  INFORMATION (Info Icon)
-  CONFIRMATION (Sim/Não)
-  ERROR (Crítico)



Exibição de Dados: Estruturando o TableView

Nome	Telefone



Setup no Scene Builder:

1. Arrastar componente TableView.
2. Definir TableColumns.
3. Column Resize Policy:
CONSTRINED_RESIZE_POLICY
(Ajuste automático de largura, sem scroll horizontal).

O Modelo de Dados: ObservableList e Factory



```
colunaNome.setCellValueFactory(new PropertyValueFactory<>("nome"));
// A string "nome" deve bater com o atributo da classe Cliente
```

Interatividade na Tabela: Seleção e Remoção

Detectando o Clique:

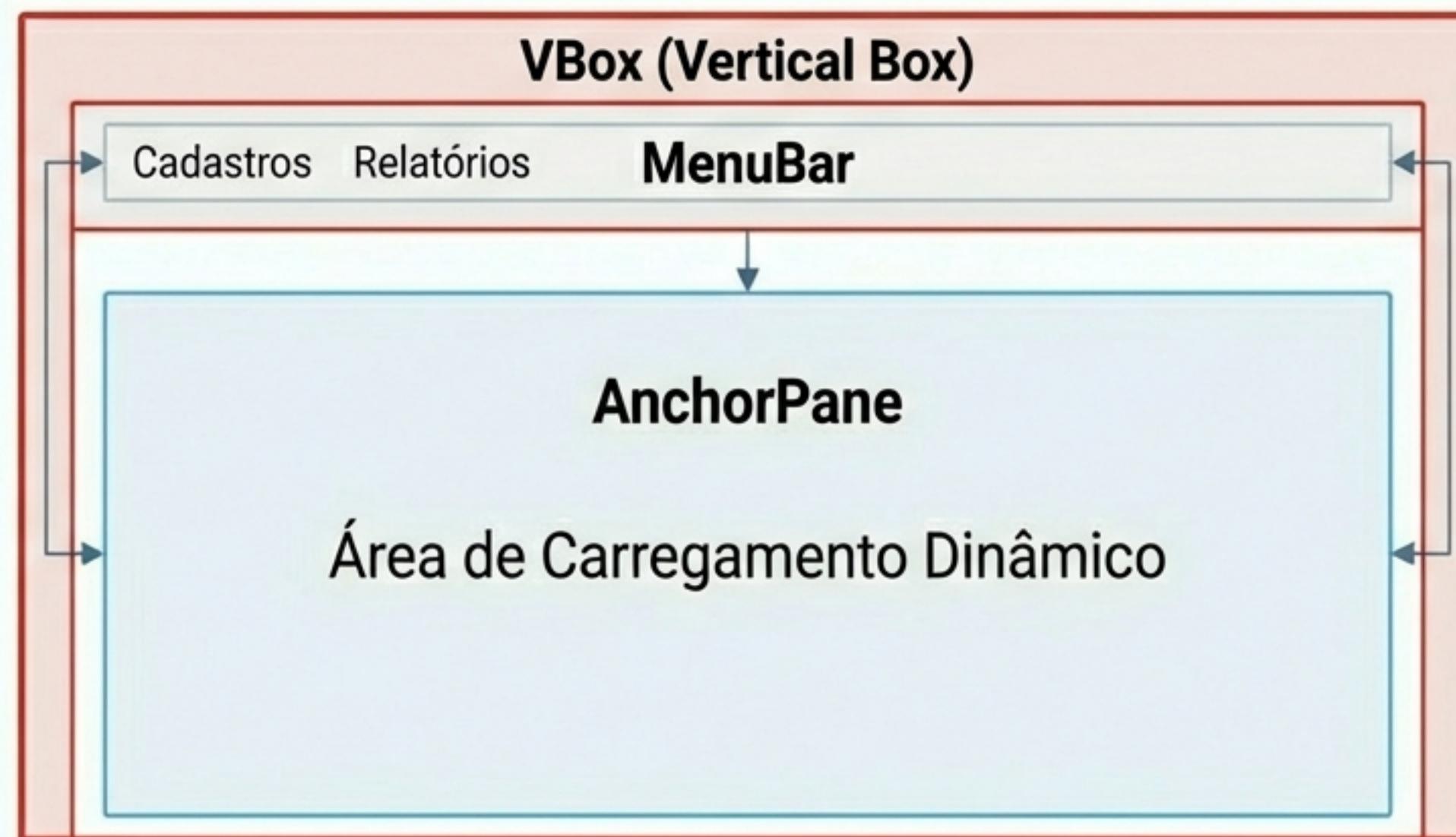
```
tableView.getSelectionModel().selectedItemProperty()
    .addListener((obs, oldSelection, newSelection) -> {
        if (newSelection != null) {
            System.out.println("Selecionado: " + newSelection.getNome());
        }
    }
});
```

Removendo Itens:

```
Cliente selecionado =
tableView.getSelectionModel().getSelectedItem();
observableListClientes.remove(selecionado);
// A tabela atualiza visualmente sozinha!
```

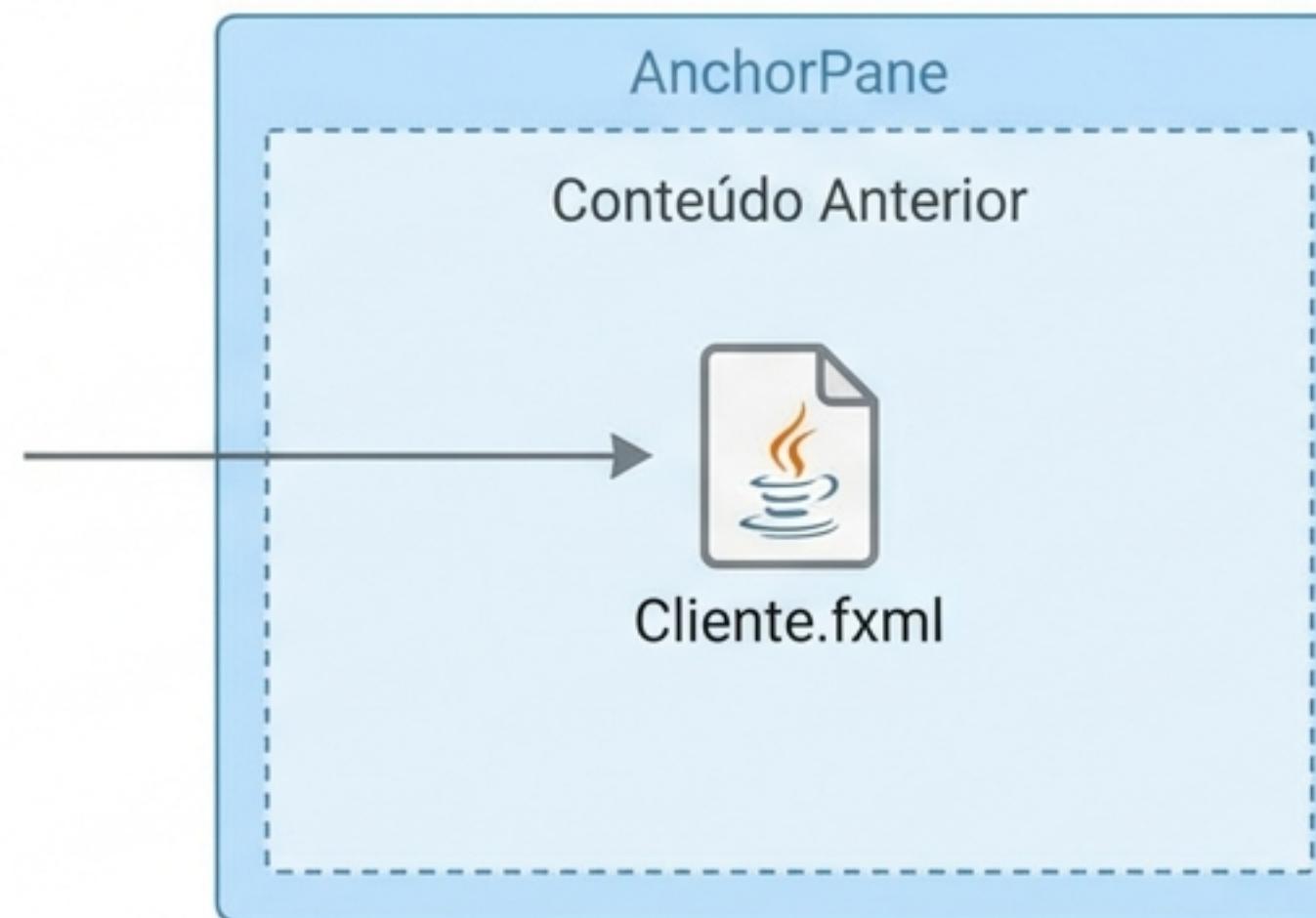


Arquitetura de Layout: Construindo a Tela Principal



O VBox organiza os elementos verticalmente. O Menu fica fixo no topo, e o AnchorPane expande para ocupar o resto da tela.

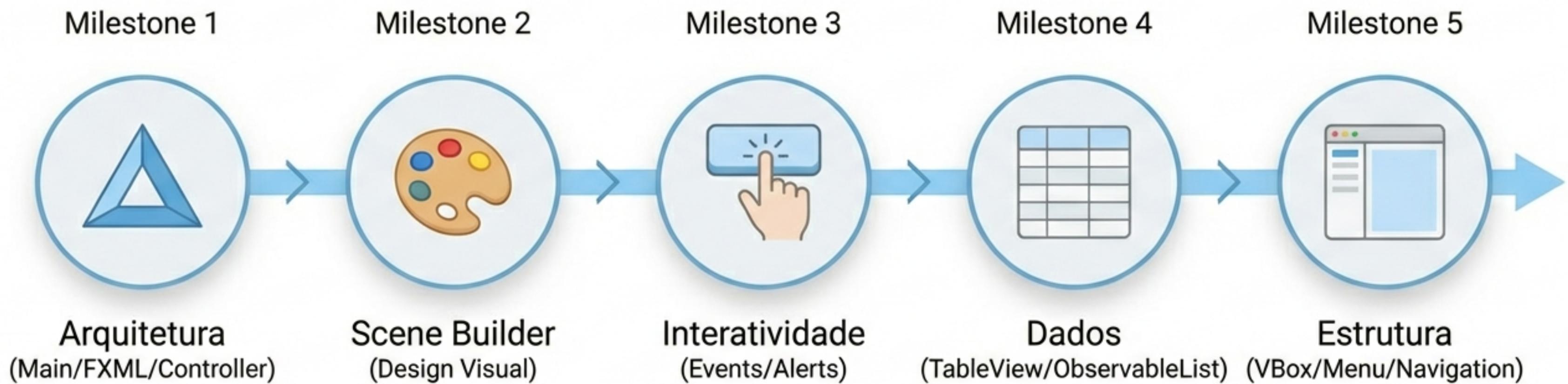
Navegação Dinâmica: Carregamento de Telas



```
// Método para trocar a tela central  
public void carregarTela(String fxml) {  
    AnchorPane a = FXMLLoader.load(getClass().getResource(fxml));  
    anchorPanePrincipal.getChildren().setAll(a);  
}
```

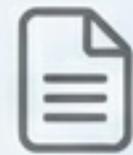
Navegação tipo SPA (Single Page Application). Não abrimos novas janelas (Stages), apenas trocamos o miolo (AnchorPane).

Resumo da Jornada JavaFX



JavaFX oferece o poder do Java com a flexibilidade visual da Web.

Referências e Recursos



Documentação Oficial Oracle: JavaFX UI Controls
docs.oracle.com/javafx



Curso em Vídeo: Integração Swing/JavaFX
video.curso/swing-javafx



Material Base: Aulas de POO II - Rafael Vargas Mesquita (IFES)
ifes.edu.br/poo2

Referências Bibliográficas

- Douglas E. Duffel. Objetos JavaFX UI Controls. douglasduffel.com/2014/07/01/javafx-uicontrols/
- Gilmar C. Góes. Páginas Empresariais. Cortes em Nível Integrado das Aplicações. gilmargoes.com.br/cortes-em-nivel-integrado-das-aplicacoes/
- Douglas E. Duffel. Desenvolvendo Aplicativos JavaFX com JavaFX SceneBuilder. douglasduffel.com/2014/07/01/javafx-scenebuilder/
- Douglas E. Duffel. ModelViewController na Arquitetura JavaFX. douglasduffel.com/2014/07/01/javafx-mvc/
- Douglas E. Duffel. Introdução ao JavaFX. douglasduffel.com/2014/07/01/introducao-ao-javafx/
- Douglas E. Duffel. Objetos. Básico Orientado a Objetos. douglasduffel.com/2014/07/01/objetos/
- Douglas E. Duffel. Introdução à POO. douglasduffel.com/2014/07/01/introducao-a-poo/