

TI-220 Java Orientado a Objetos

ANTONIO CARVALHO - TREINAMENTOS

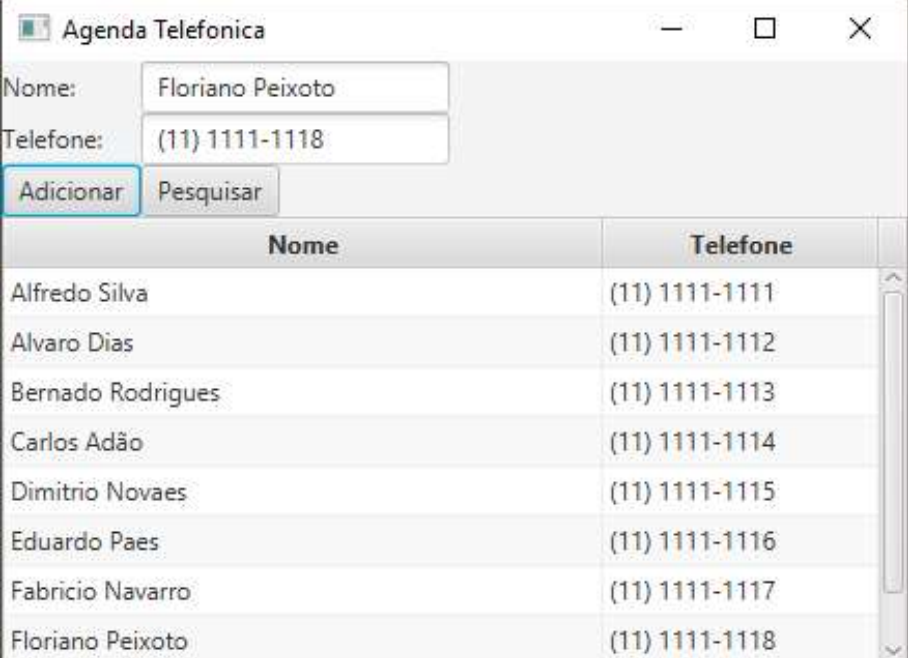
A solid blue horizontal bar spanning the width of the slide, located at the bottom.

TableView

TableView

Através do Node TableView é possível montar uma tabela no formulário JFX para mostrar linhas e colunas.

O TableView é o componente responsável por consultar os dados das colunas e coloca-los de maneira organizada na tela.



Nome	Telefone
Alfredo Silva	(11) 1111-1111
Alvaro Dias	(11) 1111-1112
Bernado Rodrigues	(11) 1111-1113
Carlos Adão	(11) 1111-1114
Dimitrio Novaes	(11) 1111-1115
Eduardo Paes	(11) 1111-1116
Fabricio Navarro	(11) 1111-1117
Floriano Peixoto	(11) 1111-1118

TableView

O componente TableView, precisa de um modelo de dados baseado em uma classe **Observable** do tipo **javafx.collections**

O modelo de dados segue esta classe **Observable**, pois desta forma quais quer mudanças nas propriedades dos objetos da Collection vão causar atualizações no componente TableView.

Para que as alterações sejam propagadas de maneira automática é necessário que:

- Os objetos guardados no elemento do tipo **javafx.collections** tenham suas propriedades baseadas em objetos do tipo **javafx.beans.property**, ou seja baseado em `StringProperty`, `BooleanProperty`, `IntegerProperty`, etc...

TableView

Se os objetos contidos no **model** do **TableView** não forem do tipo **javafx.beans.property** eles podem ser mapeados para colunas, porém as mudanças de valores nas propriedades não afetarão diretamente o componente **TableView**

Ao mapear a propriedade na coluna, será preciso envolvê-la com uma classe do tipo **ReadOnlyStringWrapper** ou do tipo **PropertyValueFactory**

TableView

Ao mapear a propriedade na coluna, deve-se criar uma TableColumn definindo o CellValueFactory para a propriedade desejada envolvida em uma classe do tipo **ReadOnlyStringWrapper**

- Envolver uma propriedade com uma classe **ReadOnlyStringWrapper**

```
TableColumn<T, String> <objeto Coluna> =  
    new TableColumn<>("<Cabeçalho da Coluna>");  
<objeto Coluna>.setCellValueFactory( itemData ->  
    new ReadOnlyStringWrapper(itemData.getValue().get<Propriedade>()) );
```

- Exemplo:

```
TableColumn<Agenda, String> telefoneColumn = new TableColumn<>("Telefone");  
telefoneColumn.setCellValueFactory( itemData ->  
    new ReadOnlyStringWrapper(itemData.getValue().getTelefone()) );
```

TableView

Outra alternativa é mapear a propriedade na coluna TableColumn definindo o CellValueFactory para a propriedade desejada envolvida em uma classe do tipo **PropertyValueFactory**

- Envolver uma propriedade com uma classe **PropertyValueFactory**

```
TableColumn<T, S> <objeto Coluna> =  
    new TableColumn<>("<Cabeçalho da Coluna>");  
<objeto Coluna>.setCellValueFactory(  
    new PropertyValueFactory<T, S>("<propriedade>"));
```

- Exemplo:

```
TableColumn<Agenda, String> telefoneColumn = new TableColumn<>("Telefone");  
telefoneColumn.setCellValueFactory(  
    new PropertyValueFactory<Agenda, String>("telefone") );
```

TableView

Uma vez com as colunas criadas, elas podem ser adicionadas ao TableView

□ Adicionar as colunas no TableView

```
<objeto TableView>.getColumns().addAll(  
    <objeto TableColumn1>,  
    <objeto TableColumn2>, ... );
```

■ Exemplo:

```
TableColumn<Agenda, String> telefoneColumn = new TableColumn<>("Telefone");  
telefoneColumn.setCellValueFactory(  
    new PropertyValueFactory<Agenda, String>("telefone") );  
TableView<Agenda> table = new TableView<>();  
table.getColumns().addAll(telefoneColumn);
```


TableView

Agora resta apenas definir a fonte de dados da tabela

□ Definir qual é a fonte de dados da tabela

```
<objeto TableView>.setItems(<objeto ObservableList>);
```

■ Exemplo:

```
ObservableList<Personagem> tableData =  
    FXCollections.observableArrayList();  
table.setItems(tableData);
```

Bibliografia

DEA, C., GRUNWALD, G., PEREDA, J., PHILLIPS, S., HECKLER, M., JavaFX 9 by Example 3rd Edition, Apress, 2017

EBBERS, H., Mastering JavaFX 8 Controls, McGrawHill Education, 2014

HOMMEL, S., JavaFX Properties and Binding Tutorial Release 8, Oracle Press, 2014

MORRIS, S., JavaFX in Action, MANNING Greenwich, 2010

ORACLE, JavaFX 2.0 Programming Model, Oracle Press, 2011

VOS, J., CHIN, S., WEIQI, G., WEAVER, J., IVERSON, D., Pro JavaFX 9 - A Definitive Guide to Building Desktop, Mobile, and Embedded Java Clients 4th edition, Apress, 2018