



# Centro Universitário UNA

Usabilidade, Desenvolvimento Web,  
Mobile e Jogos

Graduação – TI e Engenharias

Práticas de Laboratório

Carlos Augusto dos Santos Pinheiro, Cristiano de Macedo Neto, Diego Augusto de Faria  
Barros, Wesley Dias Maciel

2020/02



Centro Universitário UNA  
Graduação – TI e Engenharias  
Usabilidade, Desenvolvimento Web, Mobile e Jogos  
Prática de Laboratório  
Carlos Augusto dos Santos Pinheiro, Cristiano de Macedo Neto, Diego Augusto  
de Faria Barros, Wesley Dias Maciel  
2020/02

# Flutter



# Prática 24

## Lista

Documentação: <https://api.flutter.dev/flutter/widgets/ListView-class.html>,  
<https://api.dart.dev/stable/2.10.3/dart-core/List-class.html>,  
<https://api.flutter.dev/flutter/material/AlertDialog-class.html>

**Objetivo:** exemplificar a remoção de elementos em uma lista.

Uma lista é uma coleção indexada de objetos que possui um tamanho.

Nesta prática, você vai:

- a) Criar, inserir e remover elementos numa lista.

- 1) Crie um novo projeto Flutter, usando:
  - a. Visual Studio Code, ou;
  - b. <https://dartpad.dev/>, ou;
  - c. <https://flutlab.io/>, ou;
  - d. <https://flutterstudio.app/>, ou;
  - e. <https://codemagic.io/>.

## Lista de Produtos

- 2) O exemplo abaixo possui duas rotas. A primeira rota exibe uma lista de produtos. Inicialmente, a lista está vazia. Na segunda rota, o usuário insere os dados de um produto. Então, o produto é inserido na lista e exibido na primeira rota. Neste exemplo, os elementos são inseridos no início da lista. A cor de fundo de cada posição da lista é alternada. A exclusão de um produto é executada ao pressionar e manter pressionada uma posição da lista



```
import 'package:flutter/material.dart';

main() => runApp(
  MaterialApp(
    home: PrimeiraRota(),
    debugShowCheckedModeBanner: false,
  ),
);

class ProdutoMenu {
  String url, nome;

  ProdutoMenu({this.url, this.nome});

  static List<ProdutoMenu> menu() {
    return [
      ProdutoMenu(
        url: 'https://picsum.photos/250?image=9',
        nome: 'Notebook',
      ),
      ProdutoMenu(
        url:
          'https://images.pexels.com/photos/213780/pexels-photo-213780.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500',
        nome: 'Bolo',
      ),
      ProdutoMenu(
        url:
          'https://images.pexels.com/photos/213798/pexels-photo-213798.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500',
        nome: 'Torre e aerogerador',
      ),
    ];
  }
}

class Produto {
  int id;
  String url, nome, descricao;
  double preco;

  Produto({this.id, this.url, this.nome, this.descricao, this.preco});
}
```



```
class PrimeiraRota extends StatefulWidget {  
  @override  
  PrimeiraRotaState createState() => PrimeiraRotaState();  
}  
  
class PrimeiraRotaState extends State<PrimeiraRota> {  
  final List<Produto> produtos = <Produto>[];  
  
  void adicionarProdutoNaLista(Produto produto) {  
    setState(  
      () {  
        produtos.insert(0, produto);  
      },  
    );  
  }  
  
  void removerProdutoNaLista(int indice) {  
    setState(  
      () {  
        this.produtos.removeAt(indice);  
      },  
    );  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Lista de Produtos'),  
      ),  
      body: ListView.builder(  
        shrinkWrap: true,  
        padding: new EdgeInsets.only(left: 25, top: 2.5, right: 25),  
        itemCount: produtos.length,  
        itemBuilder: (BuildContext context, int indice) {  
          return Container(  
            height: 80,  
            margin: EdgeInsets.all(2),  
            color: (indice % 2) == 0 ? Colors.blue[50] : Colors.grey[200],  
            child: ListTile(  
              leading: Image(  
                image: NetworkImage(this.produtos[indice].url),  
                height: 50,  
                width: 70,
```



```
    ),  
    title: Text('${this.produtos[indice].nome}'),  
    subtitle: Text(  
      'R$ ${this.produtos[indice].preco.toStringAsFixed(2)}',  
      style: TextStyle(  
        fontWeight: FontWeight.w900,  
        color: Colors.purple[900],  
      ),  
    ),  
    trailing: Icon(Icons.arrow_forward),  
    onTap: () {},  
    onLongPress: () {  
      this.removeProdutoNaLista(indice);  
    },  
  ),  
);  
},  
),  
floatingActionButton: FloatingActionButton(  
  onPressed: () {  
    Navigator.push(  
      context,  
      MaterialPageRoute(builder: (context) => SegundaRota()),  
    ).then((novoProduto) {  
      adicionarProdutoNaLista(novoProduto);  
    });  
  },  
  child: Icon(Icons.add),  
  backgroundColor: Colors.purple[900],  
),  
);  
}  
}  
  
List<DropdownMenuItem<ProdutoMenu>> montarListaProdutosMenu(List menu) {  
  List<DropdownMenuItem<ProdutoMenu>> listaProdutosMenu = List();  
  
  for (ProdutoMenu produtoMenu in menu) {  
    listaProdutosMenu.add(  
      DropdownMenuItem(  
        value: produtoMenu,  
        child: Text(produtoMenu.nome),  
      ),  
    );  
  };  
};
```



```
}

return listaProdutosMenu;
}

class SegundaRota extends StatefulWidget {
  SegundaRotaState createState() {
    return SegundaRotaState();
  }
}

class SegundaRotaState extends State<SegundaRota> {
  ProdutoMenu produtoSelecionado;
  List<DropDownMenuItem<ProdutoMenu>> listaProdutosMenu;
  final TextEditingController nomeController = TextEditingController();
  final TextEditingController descricaoController = TextEditingController();
  final TextEditingController precoController = TextEditingController();

  /*O método initState() gera o estado inicial do widget
  quando um objeto da classe for instanciado.*/
  @override
  void initState() {
    listaProdutosMenu = montarListaProdutosMenu(ProdutoMenu.menu());
    produtoSelecionado = listaProdutosMenu[0].value;
    super.initState();
  }

  aoSelecionarProduto(ProdutoMenu produtoSelecionado) {
    setState(
      () {
        this.produtoSelecionado = produtoSelecionado;
      },
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Adicionar Produto'),
      ),
      body: ListView(
        shrinkWrap: true,
        padding: new EdgeInsets.only(left: 25, right: 25),
      ),
    );
  }
}
```



```
children: [
  Padding(
    padding: EdgeInsets.all(8),
    child: DropdownButton(
      value: this.produtoSelecionado,
      items: this.listaProdutosMenu,
      onChanged: this.aoSelecionarProduto,
      icon: Icon(Icons.arrow_downward),
      isExpanded: true,
      iconSize: 24,
      elevation: 16,
      style: TextStyle(fontSize: 16, color: Colors.deepPurple),
      underline: Container(
        height: 2,
        color: Colors.deepPurpleAccent,
      ),
    ),
  ),
  Padding(
    padding: EdgeInsets.all(8),
    child: TextField(
      controller: nomeController,
      decoration: InputDecoration(
        suffixIcon: IconButton(
          onPressed: () => precoController.clear(),
          icon: Icon(Icons.clear),
        ),
        border: OutlineInputBorder(),
        labelText: 'nome',
      ),
    ),
  ),
  Padding(
    padding: EdgeInsets.all(8),
    child: TextField(
      controller: descricaoController,
      decoration: InputDecoration(
        suffixIcon: IconButton(
          onPressed: () => precoController.clear(),
          icon: Icon(Icons.clear),
        ),
        border: OutlineInputBorder(),
        labelText: 'descrição',
      ),
    ),
  ),
],
```



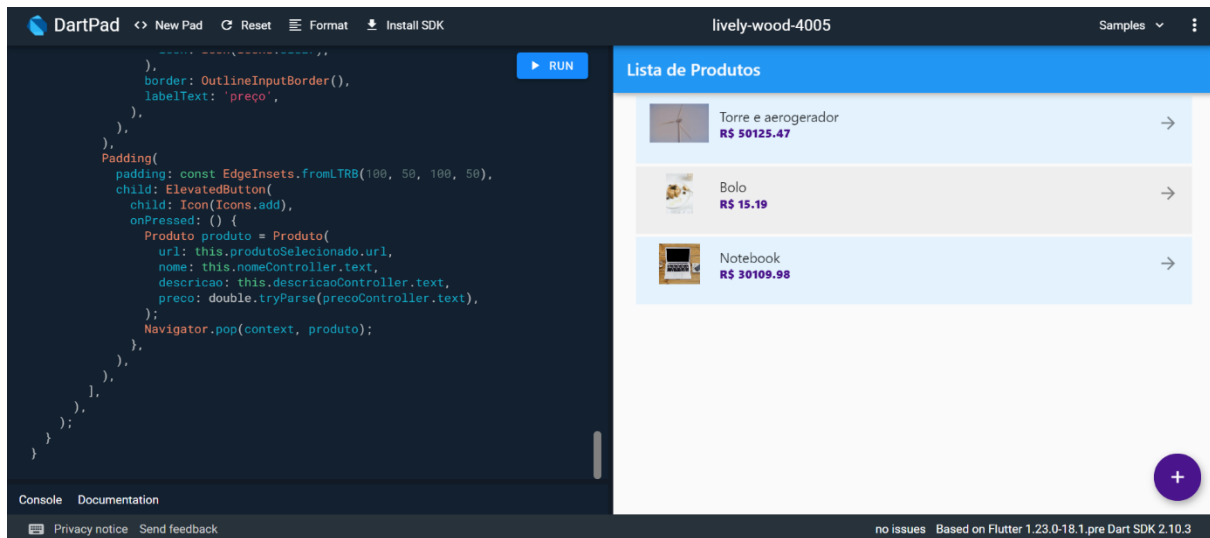


```
    ),  
  ),  
  Padding(  
    padding: const EdgeInsets.all(8),  
    child: TextField(  
      controller: precoController,  
      keyboardType: TextInputType.number,  
      decoration: InputDecoration(  
        suffixIcon: IconButton(  
          onPressed: () => precoController.clear(),  
          icon: Icon(Icons.clear),  
        ),  
        border: OutlineInputBorder(),  
        labelText: 'preço',  
      ),  
    ),  
  ),  
  Padding(  
    padding: const EdgeInsets.fromLTRB(100, 50, 100, 50),  
    child: ElevatedButton(  
      child: Icon(Icons.add),  
      onPressed: () {  
        Produto produto = Produto(  
          url: this.produtoSelecionado.url,  
          nome: this.nomeController.text,  
          descricao: this.descricaoController.text,  
          preco: double.tryParse(precoController.text),  
        );  
        Navigator.pop(context, produto);  
      },  
    ),  
  ),  
],  
),  
);  
}
```

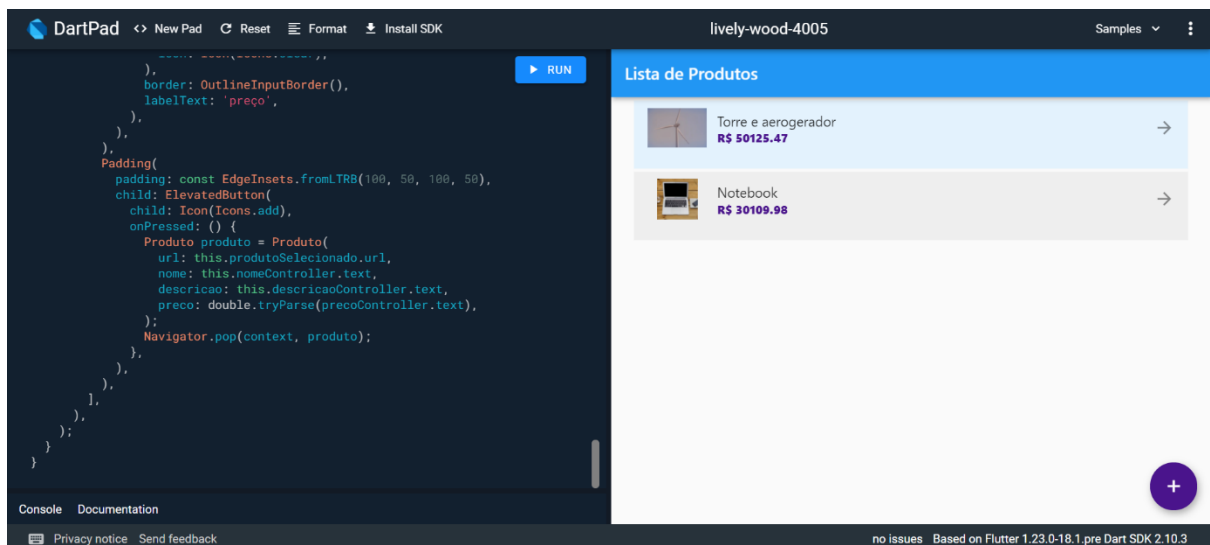


Centro Universitário UNA  
Graduação – TI e Engenharias  
Usabilidade, Desenvolvimento Web, Mobile e Jogos  
Prática de Laboratório  
Carlos Augusto dos Santos Pinheiro, Cristiano de Macedo Neto, Diego Augusto de Faria Barros, Wesley Dias Maciel  
2020/02

Antes da remoção:



Após a remoção:

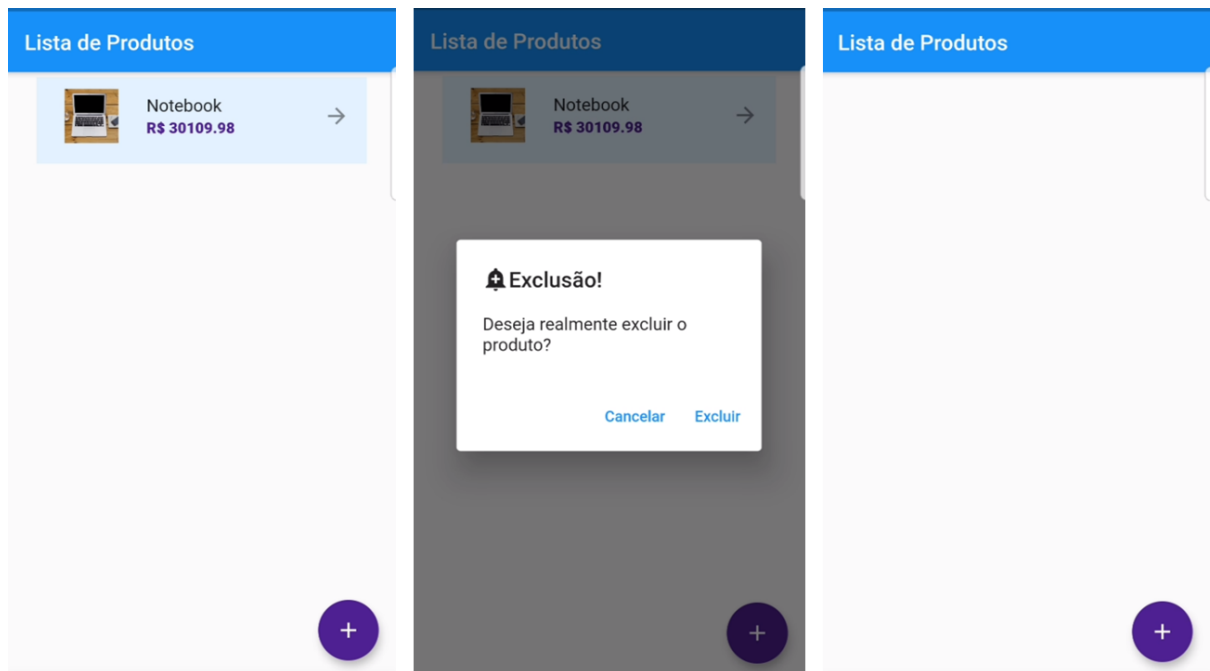


## Exercício

- 1) Altere o algoritmo apresentado nesta prática, para que ele exiba uma janela de diálogo que solicite uma confirmação para realizar a exclusão, como ilustrado abaixo.



Centro Universitário UNA  
Graduação – TI e Engenharias  
Usabilidade, Desenvolvimento Web, Mobile e Jogos  
Prática de Laboratório  
Carlos Augusto dos Santos Pinheiro, Cristiano de Macedo Neto, Diego Augusto de Faria Barros, Wesley Dias Maciel  
2020/02



Dica:

a) Janela de diálogo:

```
confirmarExclusao(BuildContext context, int indice) {  
  Widget cancelamento = FlatButton(  
    child: Text("Cancelar"),  
    onPressed: () {  
      Navigator.of(context).pop();  
    },  
  );  
  
  Widget confirmacao = FlatButton(  
    child: Text("Excluir"),  
    onPressed: () {  
      this.removerProdutoNaLista(indice);  
      Navigator.of(context).pop();  
    },  
  );  
  
  AlertDialog alerta = AlertDialog(  
    title: Row(  
      children: [
```



```
        Icon(Icons.add_alert),  
        Text("Exclusão!"),  
    ],  
),  
content: Text("Deseja realmente excluir o produto?"),  
actions: [  
    cancelamento,  
    confirmacao,  
],  
);  
  
showDialog(  
    context: context,  
    builder: (BuildContext context) {  
        return alerta;  
    },  
);  
}
```

b) Chamada da janela de diálogo:

```
onLongPress: () {  
    this.confirmarExclusao(context, indice);  
},
```

- 2) Altere o algoritmo do exercício anterior, para que ele permita atualizar o nome, descrição e preço do produto. A atualização deve ser implementada em uma nova rota.