



Flutter

Wesley Dias Maciel

2021/02

Prática 23

Lista

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

Objetivo: exemplificar a inserção de elementos em uma lista.

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

Nesta prática, você vai:

- a) Criar e inserir elementos numa lista.
- 1) Crie um novo projeto Flutter, usando:
 - a. Visual Studio Code, ou;
 - b. <https://dartpad.dev/>, ou;
 - c. <https://flutlab.io/ide>, 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.

OBS: neste exemplo a URL do produto foi considerada constante, sempre a mesma para todos os produtos.

```
import 'package:flutter/material.dart';  
  
main() => runApp(  
  // ...  
)
```

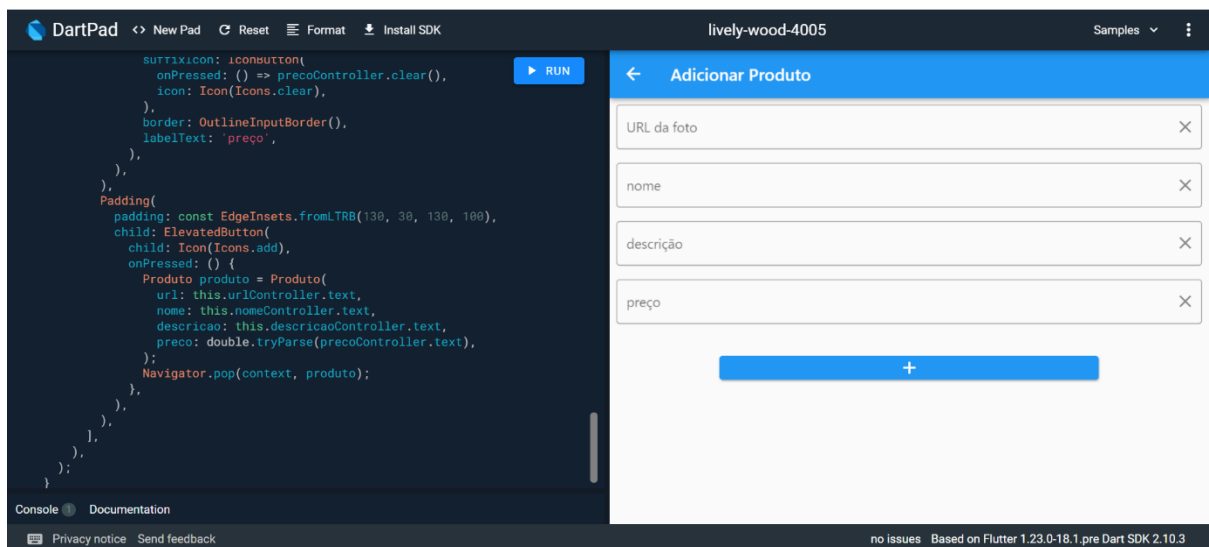
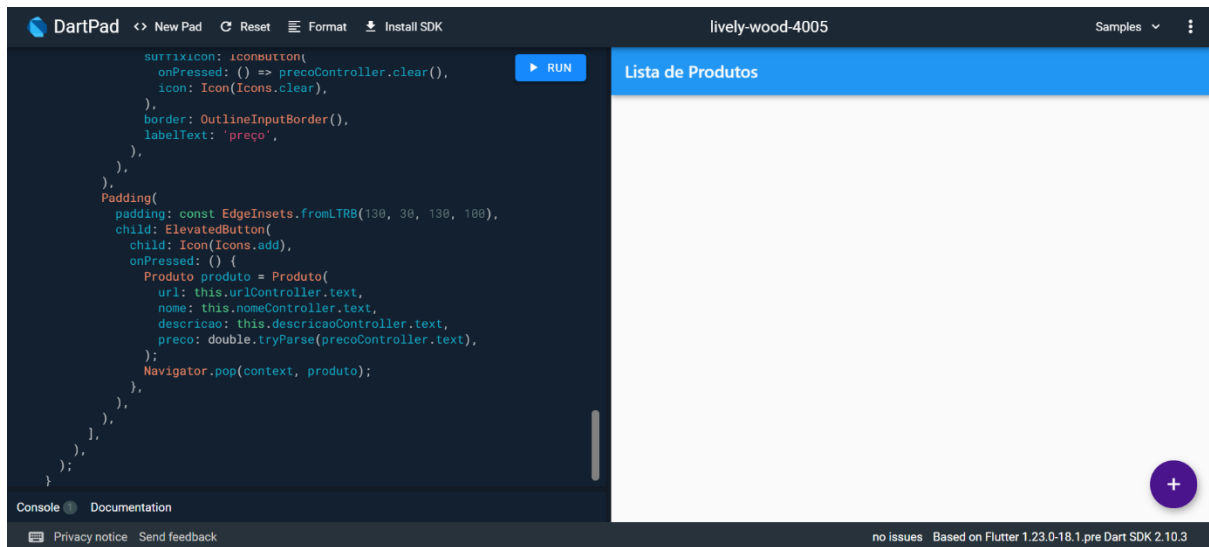
```
MaterialApp(  
  home: PrimeiraRota(),  
  debugShowCheckedModeBanner: false,  
),  
);  
  
class Produto {  
  String url, nome, descricao;  
  double preco;  
  
  Produto({  
    required this.url,  
    required this.nome,  
    required this.descricao,  
    required this.preco,  
  });  
}  
  
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);  
      },  
    );  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Lista de Produtos'),  
      ),  
      body: Corpo(this.produtos),  
      floatingActionButton: FloatingActionButton(  
        onPressed: () {  
          Navigator.push(  
            context,  
            MaterialPageRoute(builder: (context) => SegundaRota()),  
          ).then(  

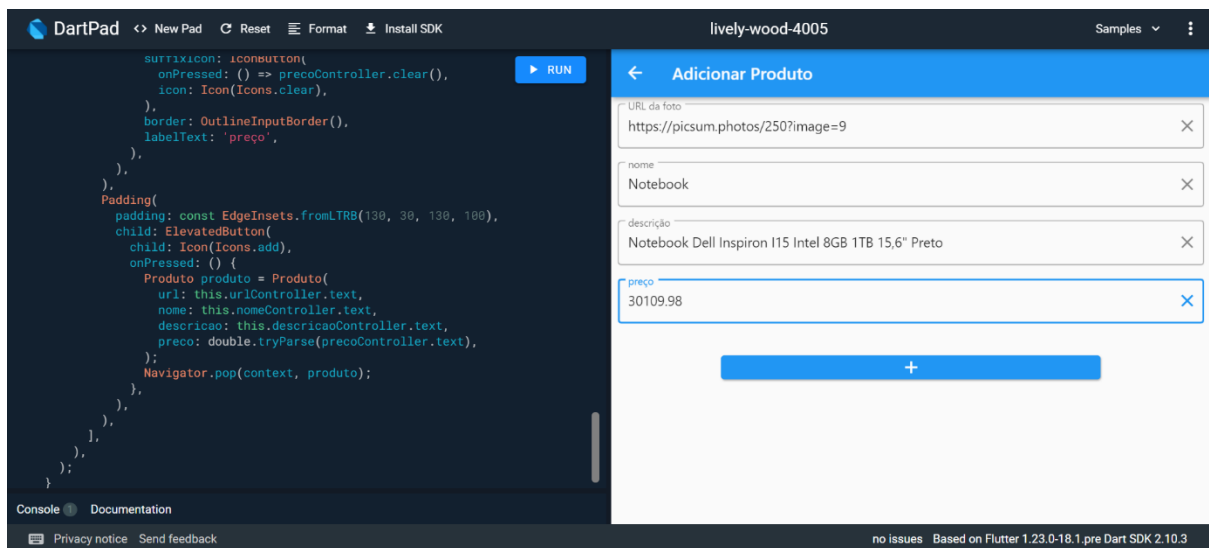
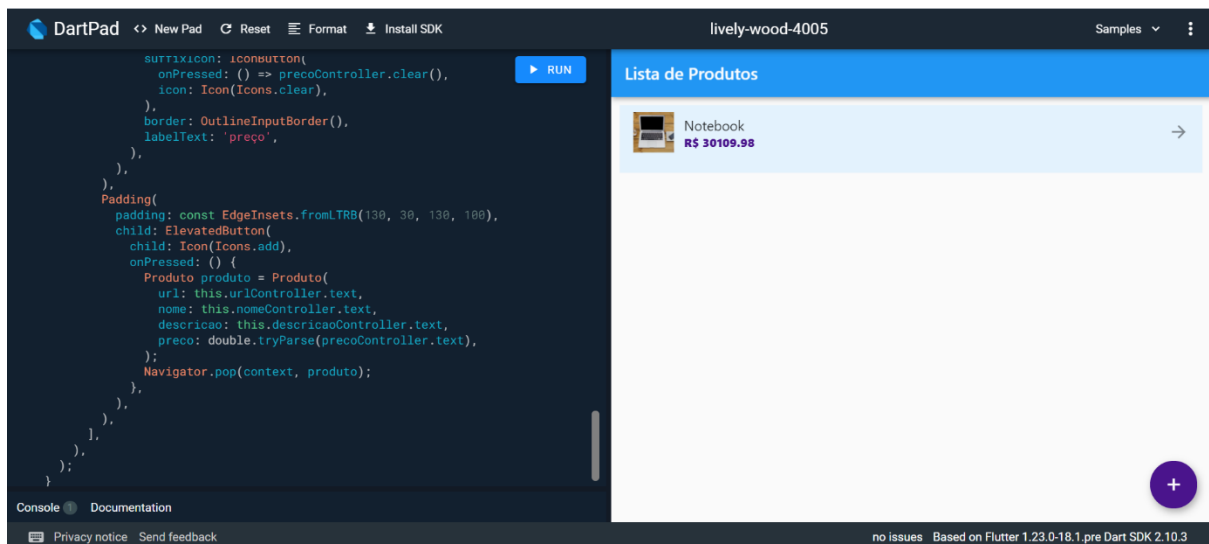
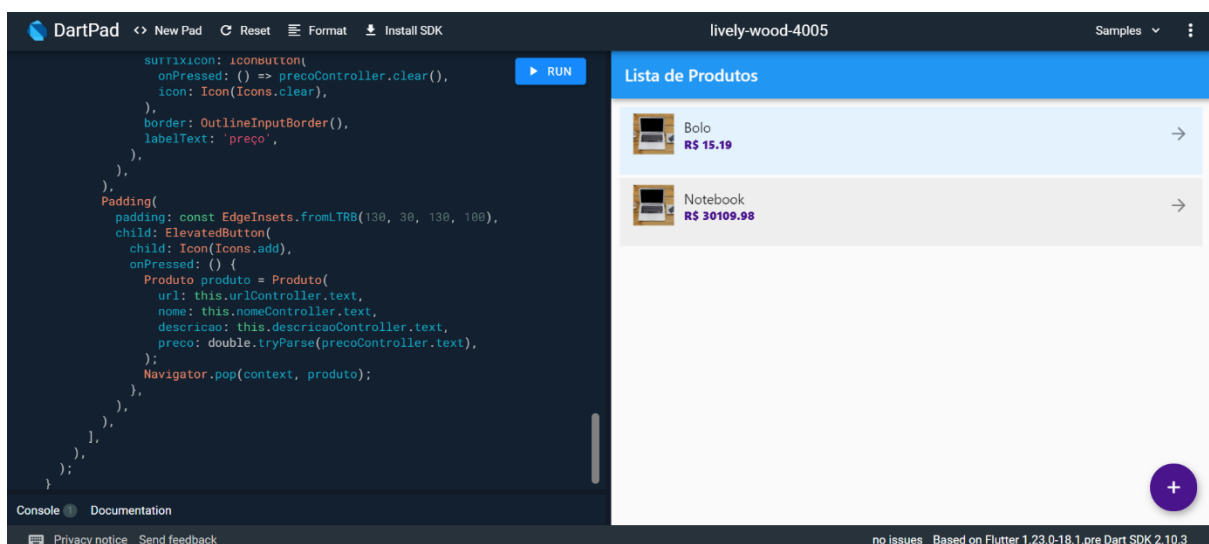
```

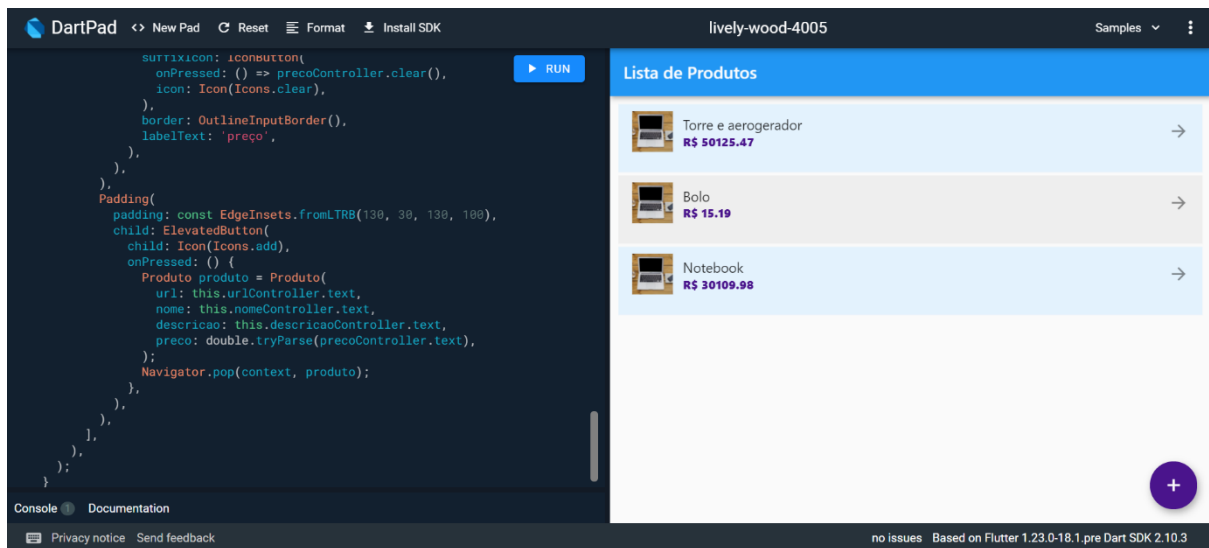
```
(novoProduto) {  
    adicionarProdutoNaLista(novoProduto);  
  },  
);  
},  
child: Icon(Icons.add),  
backgroundColor: Colors.purple[900],  
),  
);  
}  
}  
  
class Corpo extends StatelessWidget {  
  final List<Produto> produtos;  
  
  Corpo(this.produtos);  
  
  @override  
  Widget build(BuildContext context) {  
    return ListView.builder(  
      padding: EdgeInsets.all(8),  
      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(  
                'https://picsum.photos/250?image=9',  
              ),  
            ),  
            title: Text(  
              '${produtos[indice].nome}',  
            ),  
            subtitle: Text(  
              'R$ ${produtos[indice].preco.toStringAsFixed(2)}',  
              style: TextStyle(  
                fontWeight: FontWeight.w900,  
                color: Colors.purple[900],  
              ),  
            ),  
            trailing: Icon(Icons.arrow_forward),  
            onTap: () {},  
          ),  
        );  
      },  
    );  
  }  
}
```

```
    },  
  );  
}  
}  
  
class SegundaRota extends StatelessWidget {  
  final TextEditingController urlController = TextEditingController();  
  final TextEditingController nomeController = TextEditingController();  
  final TextEditingController descricaoController = TextEditingController();  
  final TextEditingController precoController = TextEditingController();  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Adicionar Produto'),  
      ),  
      body: ListView(  
        children: [  
          Padding(  
            padding: EdgeInsets.all(8),  
            child: TextField(  
              controller: this.urlController,  
              decoration: InputDecoration(  
                suffixIcon: IconButton(  
                  onPressed: () => this.urlController.clear(),  
                  icon: Icon(Icons.clear),  
                ),  
                border: OutlineInputBorder(),  
                labelText: 'URL da foto',  
              ),  
            ),  
          ),  
          Padding(  
            padding: EdgeInsets.all(8),  
            child: TextField(  
              controller: this.nomeController,  
              decoration: InputDecoration(  
                suffixIcon: IconButton(  
                  onPressed: () => this.nomeController.clear(),  
                  icon: Icon(Icons.clear),  
                ),  
                border: OutlineInputBorder(),  
                labelText: 'nome',  
              ),  
            ),  
          ),  
        ],  
      ),  
    ),  
  ),  
);
```

```
Padding(  
  padding: EdgeInsets.all(8),  
  child: TextField(  
    controller: descricaoController,  
    decoration: InputDecoration(  
      suffixIcon: IconButton(  
        onPressed: () => this.descricaoController.clear(),  
        icon: Icon(Icons.clear),  
      ),  
      border: OutlineInputBorder(),  
      labelText: 'descrição',  
    ),  
  ),  
),  
Padding(  
  padding: const EdgeInsets.all(8),  
  child: TextField(  
    controller: this.precoController,  
    keyboardType: TextInputType.number,  
    decoration: InputDecoration(  
      suffixIcon: IconButton(  
        onPressed: () => this.precoController.clear(),  
        icon: Icon(Icons.clear),  
      ),  
      border: OutlineInputBorder(),  
      labelText: 'preço',  
    ),  
  ),  
),  
Padding(  
  padding: const EdgeInsets.fromLTRB(130, 30, 130, 100),  
  child: ElevatedButton(  
    child: Icon(Icons.add),  
    onPressed: () {  
      Produto produto = Produto(  
        url: this.urlController.text,  
        nome: this.nomeController.text,  
        descricao: this.descricaoController.text,  
        preco: double.parse(this.precoController.text),  
      );  
      Navigator.pop(context, produto);  
    },  
  ),  
),  
],  
,  
);
```

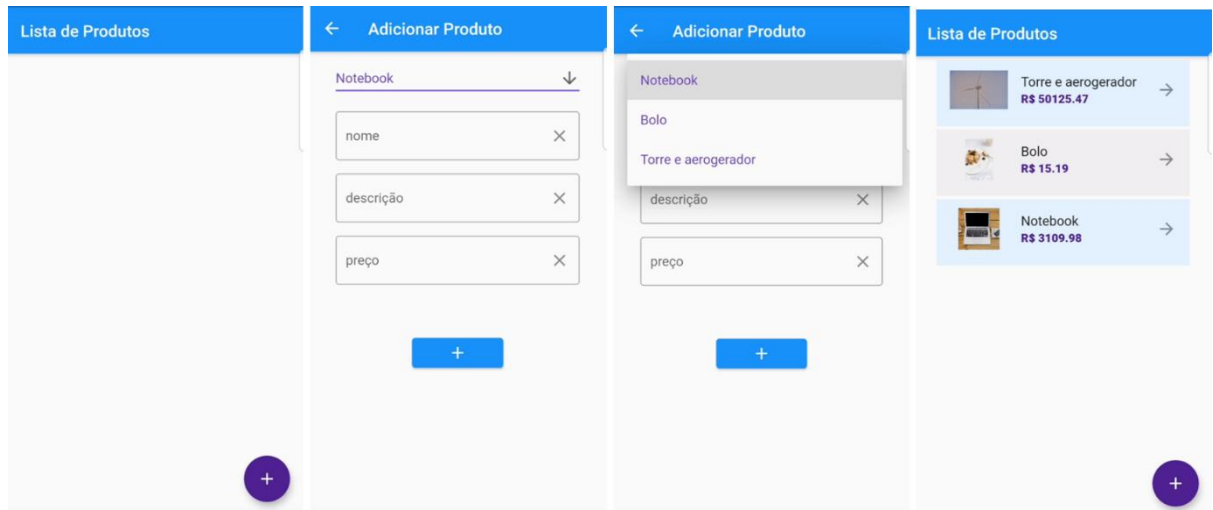
$$\left. \begin{array}{l} \{ \\ \{ \end{array} \right\}$$




Exercício

1) Altere o algoritmo apresentado nesta prática, para que ele gere as telas abaixo.



Dica:

a) Classe Menu:

```
class Menu {
  static List<ItemMenu> getItens() {
    return [
      ItemMenu(
        url: 'https://picsum.photos/250?image=9',
        nome: 'Notebook',
      ),
    ];
  }
}
```

```
    ),  
    ItemMenu(  
      url:  
        'https://images.pexels.com/photos/213780/pexels-photo-  
213780.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500',  
      nome: 'Bolo',  
    ),  
    ItemMenu(  
      url:  
        'https://images.pexels.com/photos/213798/pexels-photo-  
213798.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500',  
      nome: 'Torre e aerogerador',  
    ),  
  ],  
};  
}  
  
static List<DropdownMenuItem<ItemMenu>> getDropdownMenuItens(List itens) {  
  List<DropdownMenuItem<ItemMenu>> listaItensMenu = [];  
  
  for (ItemMenu item in itens) {  
    listaItensMenu.add(  
      DropdownMenuItem(  
        value: item,  
        child: Text(item.nome),  
      ),  
    );  
  }  
  
  return listaItensMenu;  
}  
  
static List<DropdownMenuItem<ItemMenu>> getListaItens() {  
  return getDropdownMenuItens(getItens());  
}  
}
```

b) Corpo da primeira rota:

```
class Corpo extends StatelessWidget {  
  final List<Produto> produtos;  
  
  Corpo(this.produtos);  
  
  @override  
  Widget build(BuildContext context) {  
    return ListView.builder(  

```

```
shrinkWrap: true, //Ocupar apenas o espaço necessário.
padding: 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: () {},
    ),
  );
},
);
}
```

c) Segunda rota:

```
class SegundaRotaState extends State<SegundaRota> {
  ItemMenu? itemSelecionado;
  List<DropDownMenuItem<ItemMenu>>? listaItensMenu;
  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() {
    listaItensMenu = Menu.getListaItens();
    itemSelecionado = listaItensMenu![0].value;
    super.initState();
  }
}
```

```
}

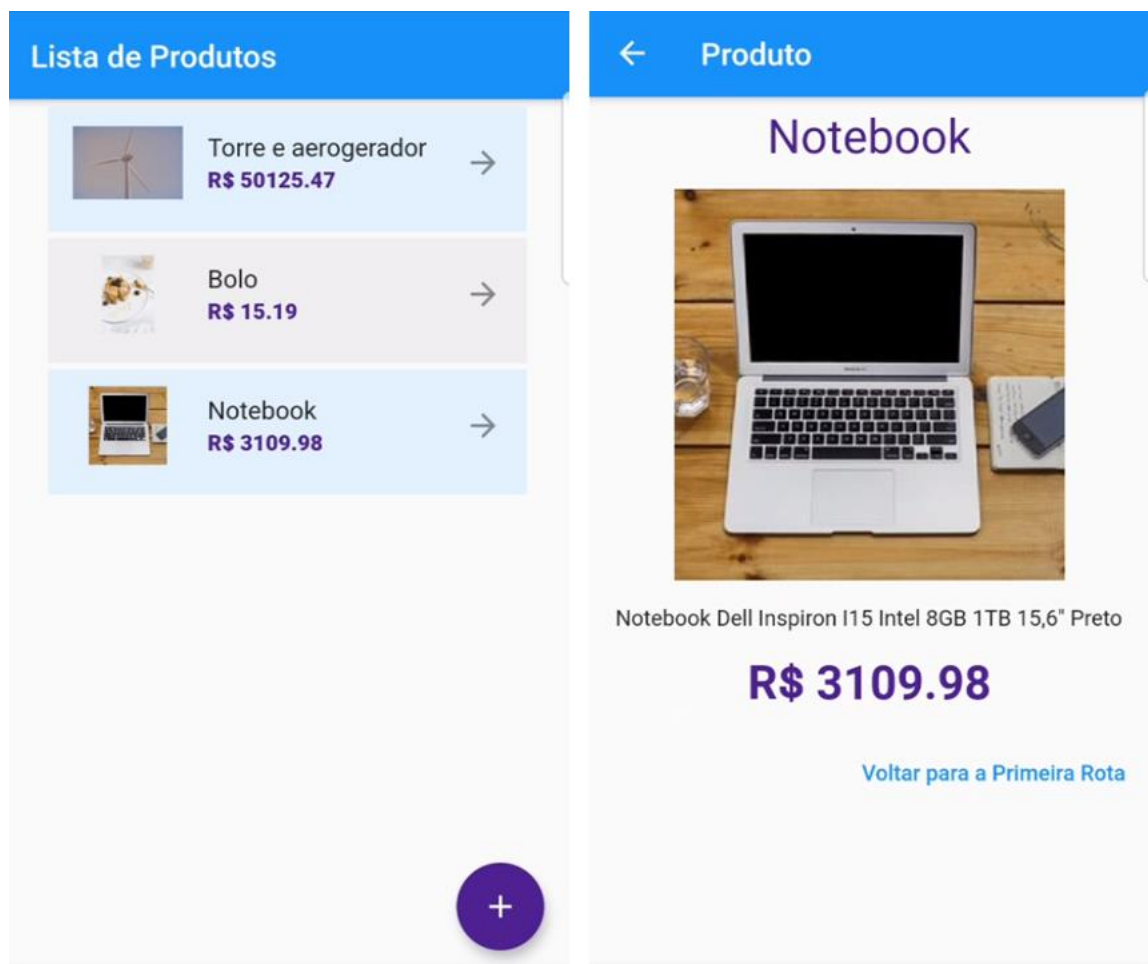
aoSelecionarItem(ItemMenu? itemSelecionado) {
  setState(
    () {
      this.itemSelecionado = itemSelecionado;
    },
  );
}

@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.itemSelecionado,
            items: this.listaItensMenu,
            onChanged: this.aoSelecionarItem,
            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: this.nomeController,
            decoration: InputDecoration(
              suffixIcon: IconButton(
                onPressed: () => this.nomeController.clear(),
                icon: Icon(Icons.clear),
              ),
            ),
            border: OutlineInputBorder(),
          ),
        ),
      ],
    ),
  );
}
```

```
        labelText: 'nome',
      ),
    ),
  ),
  Padding(
    padding: EdgeInsets.all(8),
    child: TextField(
      controller: this.descricaoController,
      decoration: InputDecoration(
        suffixIcon: IconButton(
          onPressed: () => this.descricaoController.clear(),
          icon: Icon(Icons.clear),
        ),
        border: OutlineInputBorder(),
        labelText: 'descrição',
      ),
    ),
  ),
  Padding(
    padding: const EdgeInsets.all(8),
    child: TextField(
      controller: this.precoController,
      keyboardType: TextInputType.number,
      decoration: InputDecoration(
        suffixIcon: IconButton(
          onPressed: () => this.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.itemSelecionado!.url,
          nome: this.nomeController.text,
          descricao: this.descricaoController.text,
          preco: double.parse(this.precoController.text),
        );
        Navigator.pop(context, produto);
      },
    ),
  ),
```



- 2) Altere o algoritmo da questão anterior, para que o usuário seja direcionado para uma rota 03 de descrição do produto ao clicar sobre uma das posições da lista apresentada na rota 01. A rota 03 também deve possuir um botão que permita retornar para a rota 01. Exemplo:



- 3) Altere o algoritmo do exercício anterior, para que os produtos sejam inseridos no final da lista apresentada na rota 01.

Dica: alterar o método `adicionarProdutoNaLista()` para:

```
void adicionarProdutoNaLista(Produto produto) {
```

```
setState(  
  () {  
    produtos.insert(produtos.length, produto);  
  },  
);  
}
```