



Flutter

Wesley Dias Maciel

2021/02

Prática 15

Retornar Valor de uma Rota para Outra

Documentação: <https://flutter.dev/docs/cookbook/navigation/returning-data>,
<https://api.flutter.dev/flutter/widgets/Navigator-class.html>

Objetivo: considerando um aplicativo com duas rotas, retornar um valor processado na segunda rota para a primeira através do método `Future.then ()`.

Nesta prática, você vai:

- 1) Criar as duas rotas.
- 2) Navegar da primeira para a segunda rota usando o método `Navigator.push ()`.
 - Usar o método `Future.then ()` para atualizar a primeira rota com a função `setState ()`.
- 3) Usar o método `Navigator.pop ()` para voltar para a primeira rota com o valor gerado na segunda.

- 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/>.

Criar duas Rotas

- 2) O exemplo abaixo cria as duas rotas.

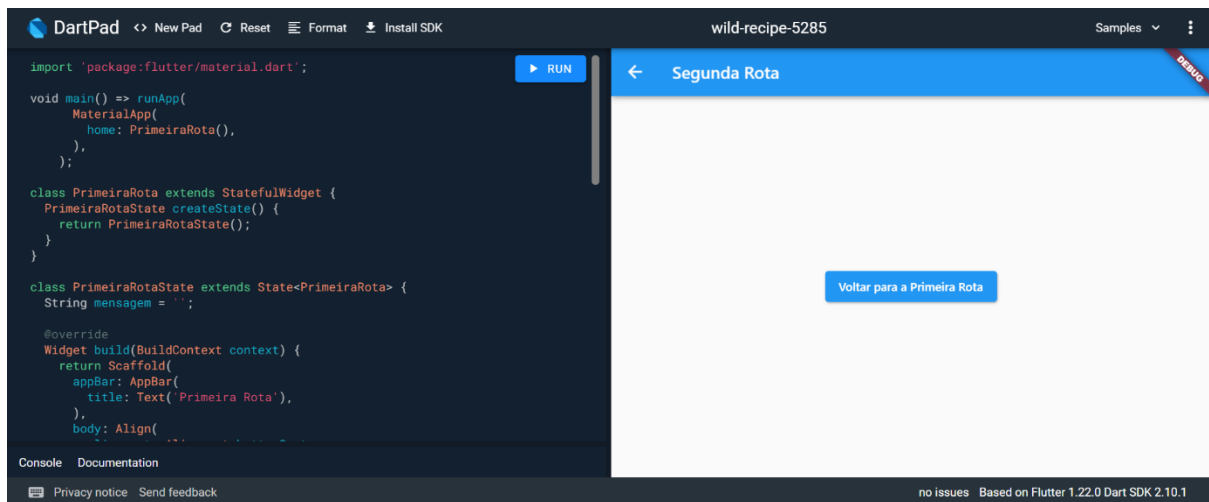
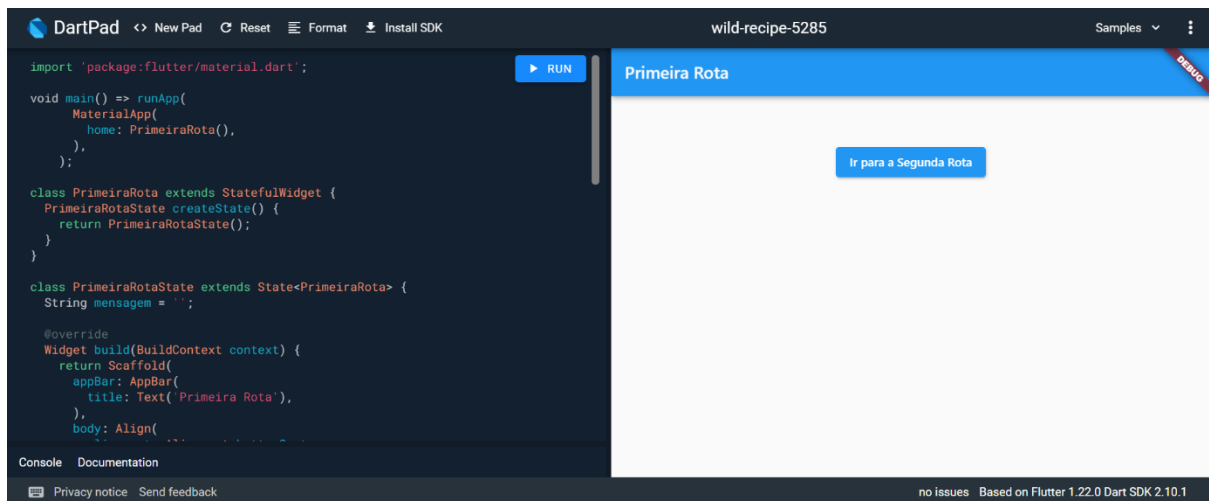
```
import 'package:flutter/material.dart';

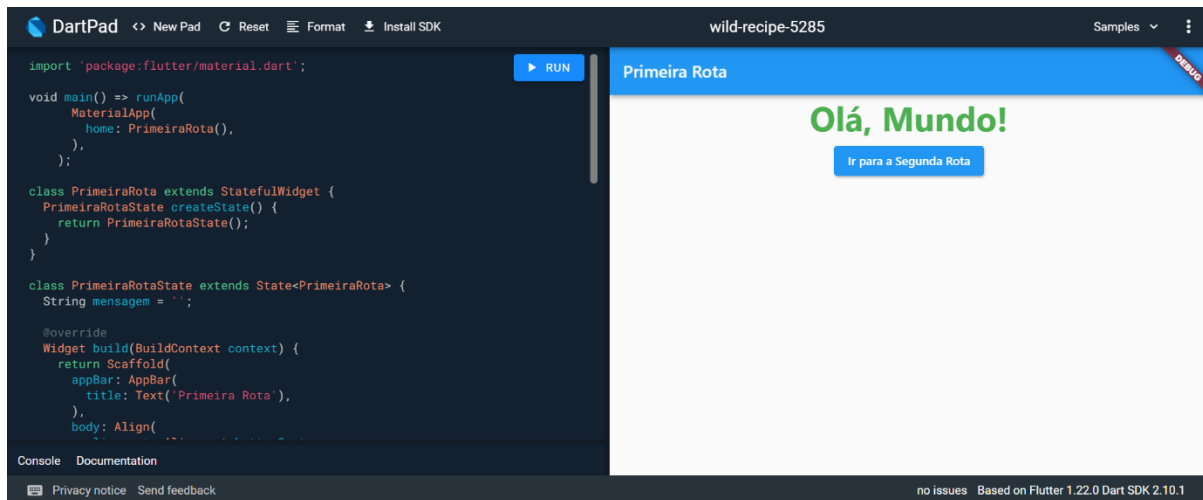
void main() => runApp(
  MaterialApp(
    home: PrimeiraRota(),
  ),
);

class PrimeiraRota extends StatefulWidget {
```

```
PrimeiraRotaState createState() {  
  return PrimeiraRotaState();  
}  
}  
  
class PrimeiraRotaState extends State<PrimeiraRota> {  
  String mensagem = '';  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Primeira Rota'),  
      ),  
      body: Center(  
        child: Column(  
          children: [  
            Text(  
              '$mensagem',  
              style: TextStyle(  
                fontSize: 40,  
                color: Colors.green,  
                fontWeight: FontWeight.bold,  
              ),  
            ),  
            ElevatedButton(  
              child: Text('Ir para a Segunda Rota'),  
              onPressed: () {  
                Navigator.push(  
                  context,  
                  MaterialPageRoute(builder: (context) => SegundaRota()),  
                ).then((resposta) {  
                  setState(() {  
                    mensagem = resposta;  
                  });  
                });  
              },  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}  
  
class SegundaRota extends StatelessWidget {  
  @override
```

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: Text("Segunda Rota"),  
    ),  
    body: Center(  
      child: ElevatedButton(  
        onPressed: () {  
          Navigator.pop(context, 'Olá, Mundo!');  
        },  
        child: Text('Voltar para a Primeira Rota'),  
      ),  
    ),  
  );  
}
```





Navigator.push () e Future.then ()

- 3) No atributo onPressed do botão da primeira rota, permitir a navegação através do método Navigator.push (). Além disso, usar o método Future.then () para atualizar a primeira rota com a função setState ().

O método Future.then () só será executado depois que a segunda rota retornar uma resposta através de Navigator.pop (), nesse exemplo a resposta é do tipo String. Essa resposta é atribuída ao objeto mensagem, que é usado na atualização da primeira rota.

```
onPressed: () {  
  Navigator.push(  
    context,  
    MaterialPageRoute(builder: (context) => SegundaRota()),  
  ).then((resposta) {  
    setState(() {  
      mensagem = resposta;  
    });  
  });  
},
```

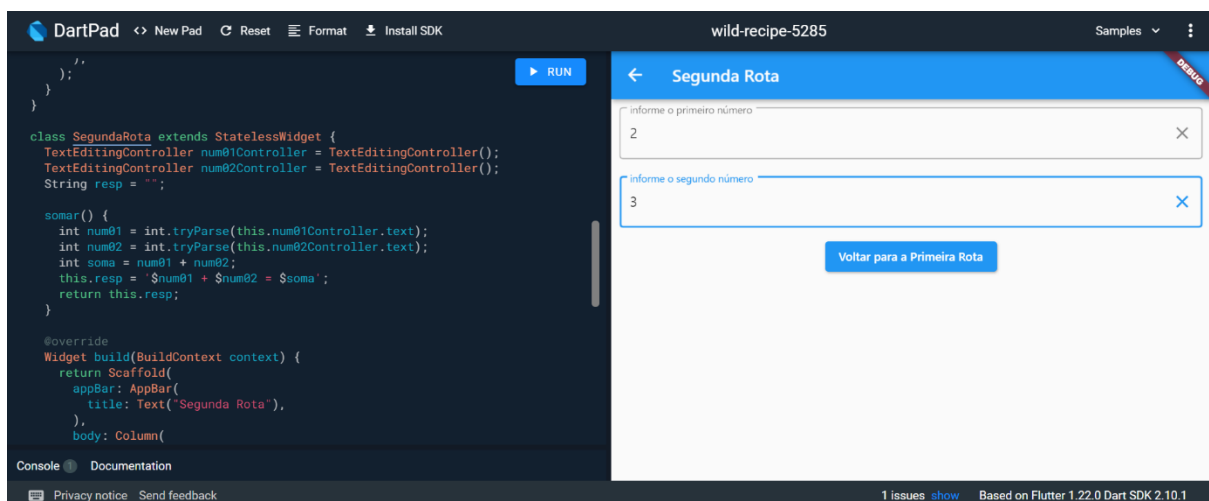
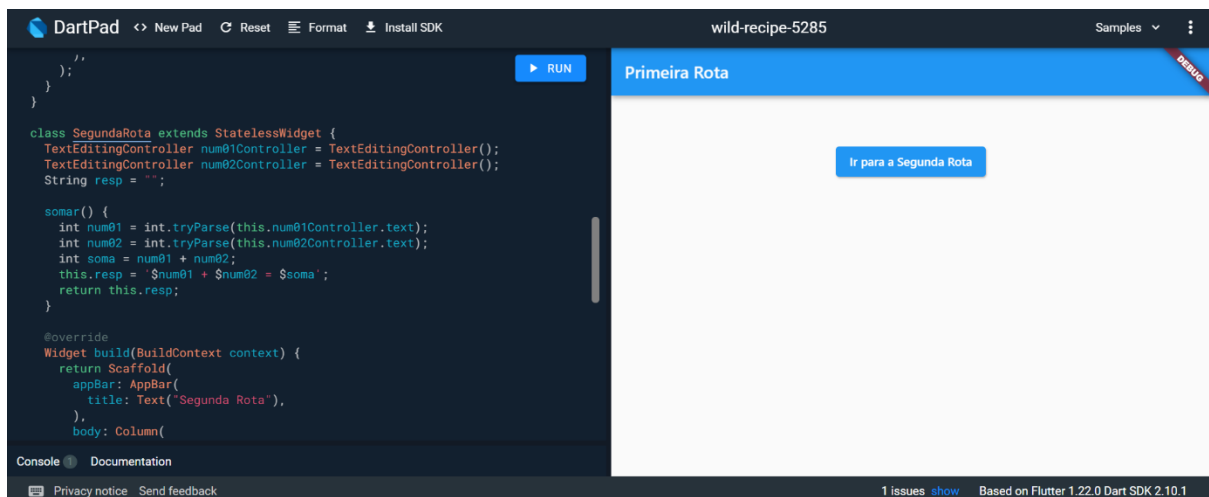
Navigator.pop ()

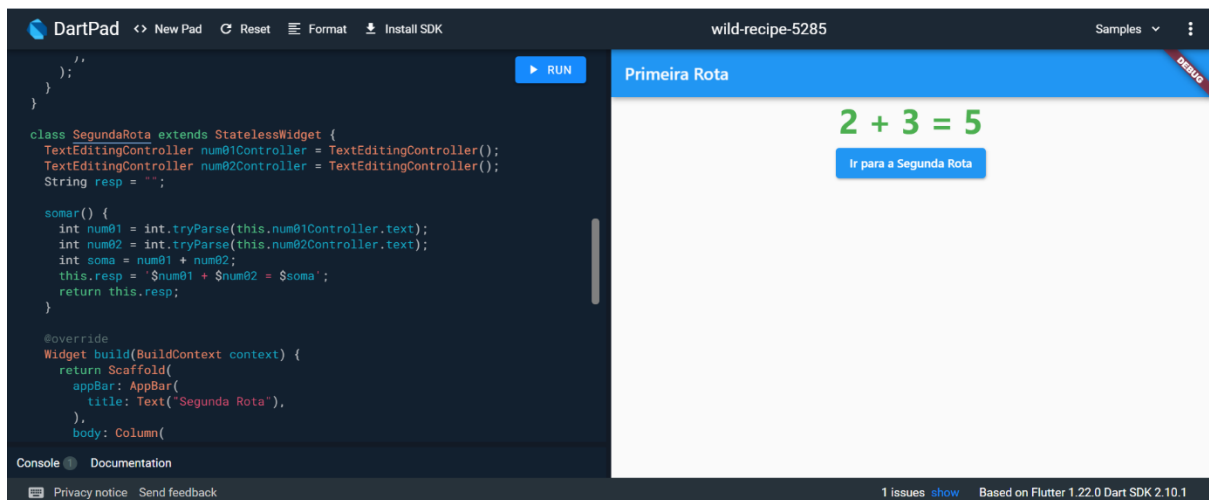
- 4) No atributo onPressed do botão da segunda rota, usar o método Navigator.pop () para voltar para a primeira rota, retornando o valor gerado. Nesse exemplo, o valor retornado é o String 'Olá, Mundo!'.

```
onPressed: () {  
  Navigator.pop(context, 'Olá, Mundo!');  
},
```

Exercício

- 1) Altere o algoritmo apresentado nesta prática, de forma que:
 - a. A primeira rota tenha um botão que direcione o usuário para a segunda rota.
 - b. A segunda rota permita que o usuário informe dois números. Ao clicar no botão da segunda rota, o usuário é direcionado para a primeira rota juntamente com a soma dos dois números informados.
 - c. Ao retornar, a primeira rota deve apresentar o resultado da soma dos números informados.





- 2) Altere o algoritmo da questão anterior, criando uma refatoração com os seguintes arquivos:
- lib/main.dart: arquivo com a função "main".
 - lib/rota/rota.dart: arquivo com as definições de PrimeiraRota, PrimeiraRotaState e SegundaRota.

Estrutura de diretórios:

