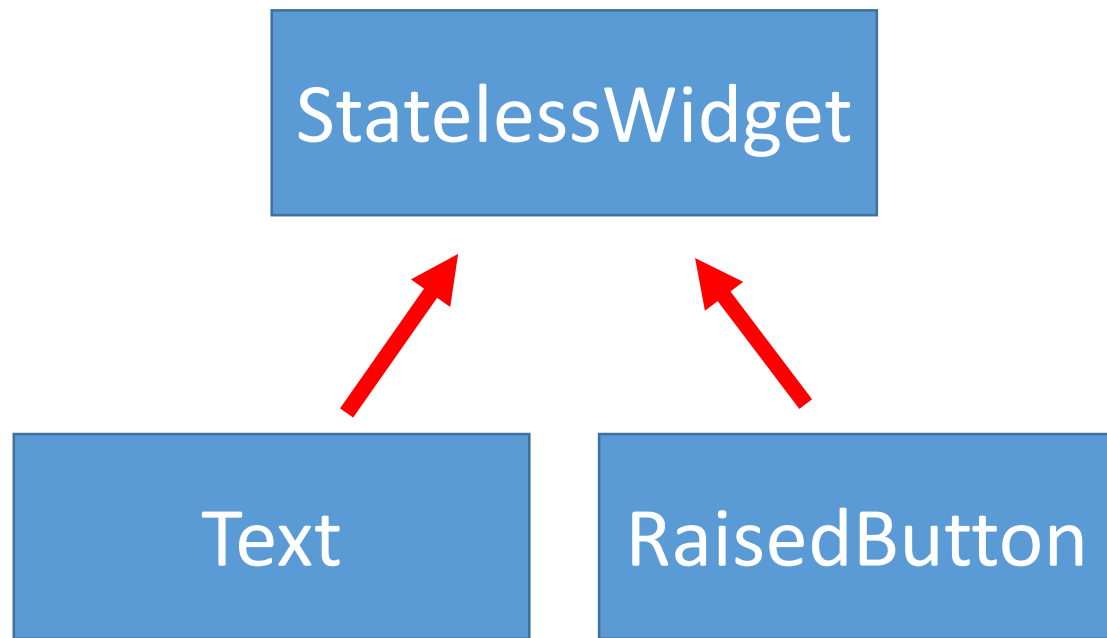
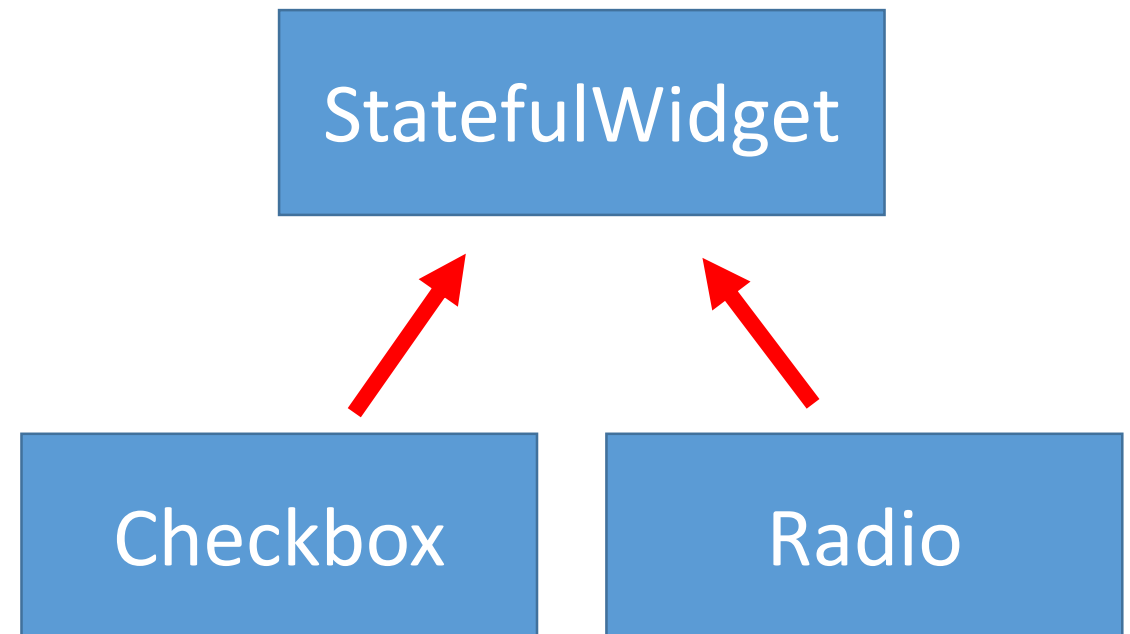


# Widget – Parte 2

# Conceitos de StatelessWidget e StatefulWidget

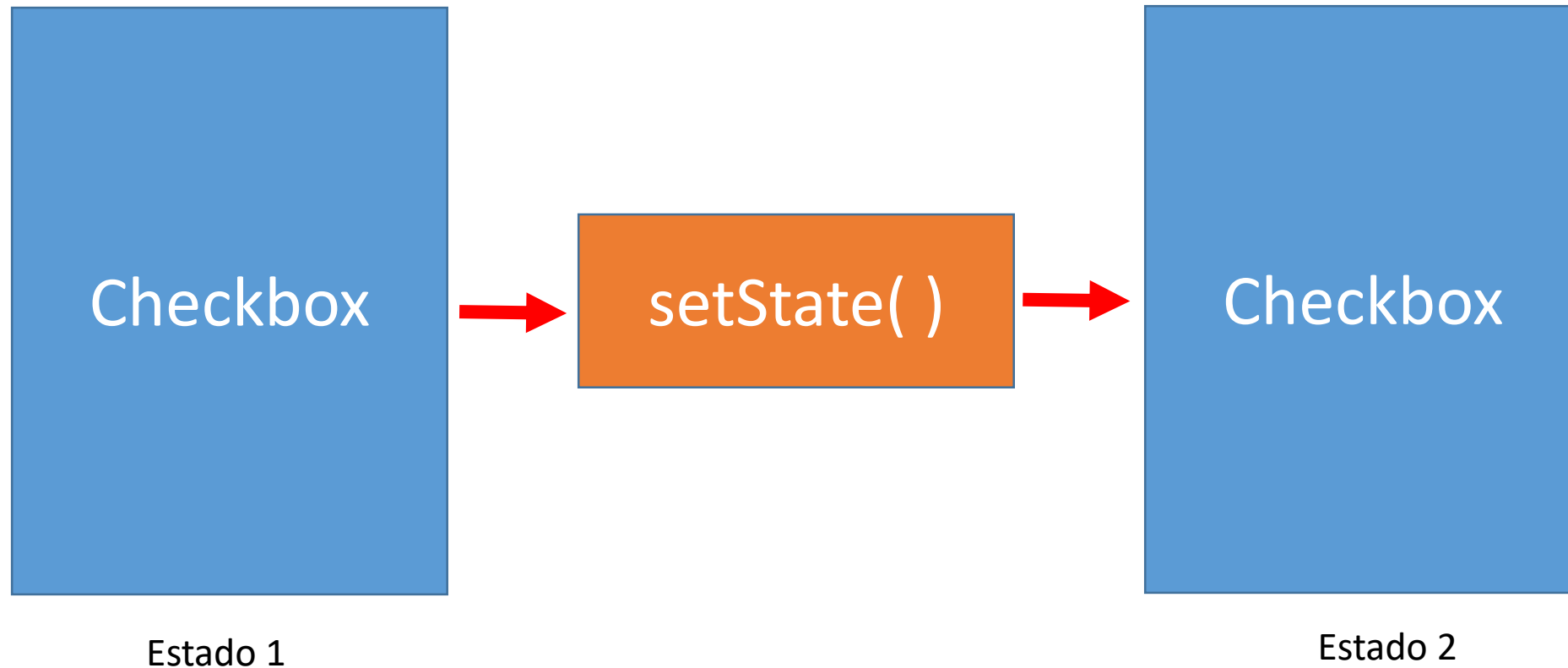


Esses widgets não permitem controlar o estado interno.



Esses widgets permitem controlar o estado interno.

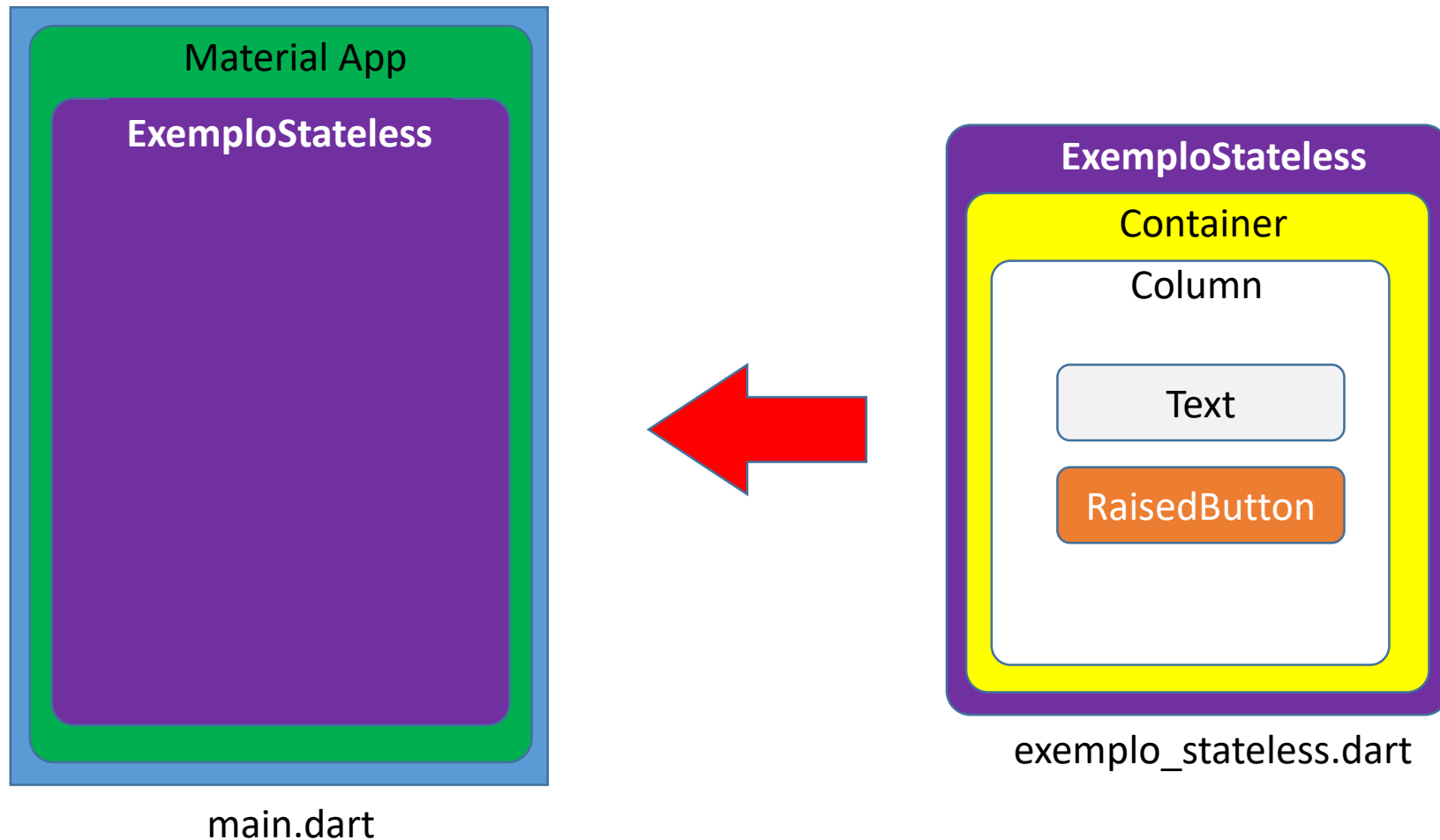
# Conceitos de StatelessWidget e StatefulWidget



# Conceitos de StatelessWidget e StatefulWidget

- Podemos construir nossos widgets simplesmente estendendo uma das classes:
  - StatelessWidget ;
  - StatefulWidget.
- As ferramentas para desenvolvimento flutter fornece a partir de um comando toda estrutura base para construir um widget.
- Uma boa prática de programação flutter é criar nossos widgets em arquivos separados.

# Criando um widget com StatelessWidget (1)



# Criando um widget com StatelessWidget (2)

```
void main(){  
  runApp(  
    MaterialApp(  
      home: ExemploStateless(),  
    ) // MaterialApp  
  );  
}
```

# Criando um widget com StatelessWidget (3)

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

class ExemploStateless extends StatelessWidget {
  double _valor = 0;

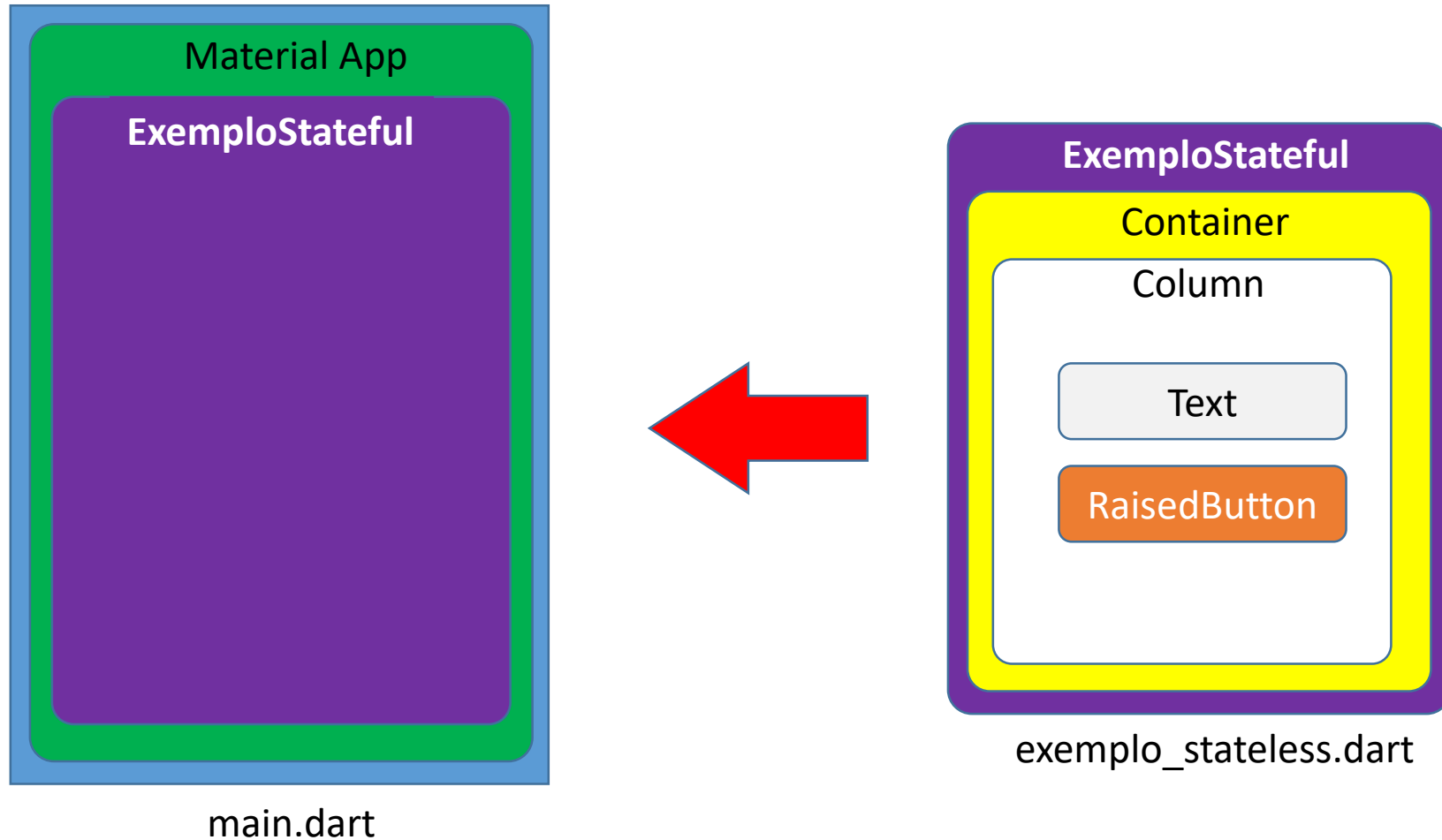
  void _acrecetar() {
    _valor += 100;
    print("R\$ $_valor");
  }

  @override
  Widget build(BuildContext context) {
    return Container(
      color: Colors.white,
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Text("R\$ $_valor",
            style: TextStyle(
              fontSize: 30,
              decoration: TextDecoration.none
            ), // TextStyle
        ), // Text
```

Continuação

```
          RaisedButton(child: Text("Rico",
            style: TextStyle(
              fontSize: 30,
            ), // TextStyle
          ), // Text
        ), // RaisedButton
      ], // <Widget>[]
    ), // Column
  ); // Container
}
```

# Criando um widget com StatefulWidget (1)





# Criando um widget com StatefulWidget (2)

```
void main(){  
  runApp(  
    MaterialApp(  
      home: ExemploStateful(),  
    ) // MaterialApp  
  );  
}
```

# Criando um widget com StatefulWidget (3)

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

class ExemploStateful extends StatefulWidget {
  @override
  _ExemploStatefulState createState() => _ExemploStatefulState();
}

class _ExemploStatefulState extends State<ExemploStateful> {
  double _valor = 0;

  void _acrecantar() {
    setState(() {
      _valor += 100;
      print("R\$ $_valor");
    });
  }

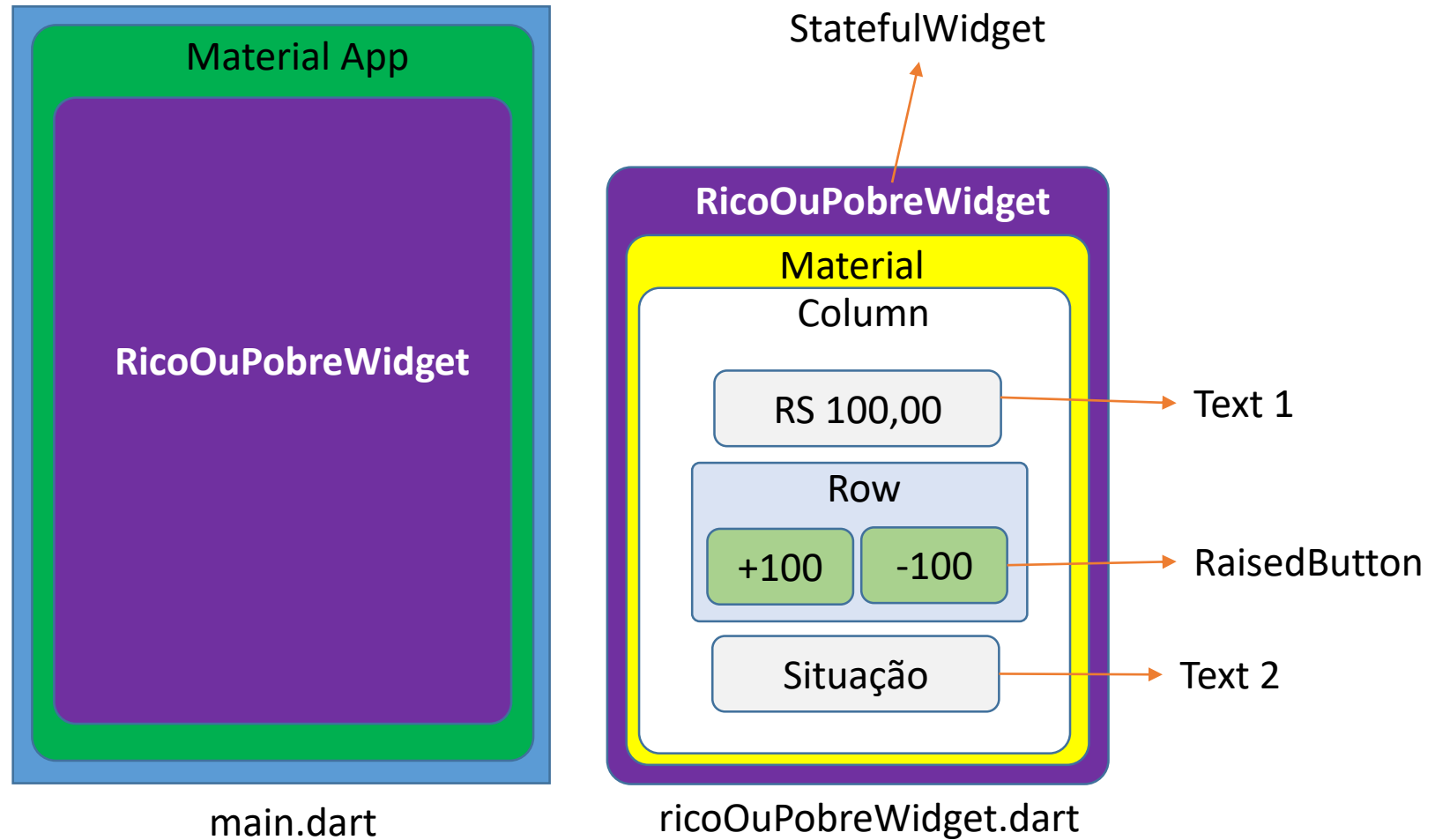
  @override
  Widget build(BuildContext context) {
    return Container(
      color: Colors.white,
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Text(
            "R\$ $_valor",
```

Continuação



```
            "R\$ $_valor",
            style: TextStyle(
              fontSize: 30,
              decoration: TextDecoration.none
            ), // TextStyle
          ), // Text
          RaisedButton(
            child: Text(
              "Rico",
              style: TextStyle(
                fontSize: 30,
              ), // TextStyle
            ), // Text
            onPressed: () => _acrecantar(),
          ), // RaisedButton
        ], // <Widget>[]
      ), // Column
    ); // Container
  }
}
```

# Exercício 1 - AppRicoOuPobre



# Exercício 1 - AppRicoOuPobre

- As cores dos componentes visuais ficam a critério do aluno;
- Todos os componentes no widget Column devem ficar centralizados;
- Os widgets RaisedButton fazem:
  - Deixar o usuário 100 reais mais rico
  - Deixar o usuário 100 reais menos rico
- O widget Text 1 apresenta o valor total do usuário;
- O widget Text 2 apresenta a situação do usuário;
  - Caso o valor maior que 1000, usuário rico;
  - Caso o valor menor que 0, usuário pobre;
  - Do contrario, usuário normal