



## Mão na massa: implementando o Cubit no contador do Flutter

*Chegou o momento colocar em prática o que aprendeu até agora e, assim, vou propor a você um desafio. E aí, topa?*

Quando você estava começando a aprender Flutter, muito provavelmente criou o famoso **contador** (projeto que vem por padrão quando criamos uma aplicação), é o nosso "Hello, World!" no Flutter, certo?

Como sabemos, o contador utiliza `StatefulWidget` e o `setState` para avisar ao Flutter que a pessoa usuária apertou o `FloatingActionButton` e que deve atualizar a interface da aplicação.

Com base nisso, meu desafio para você é o seguinte:

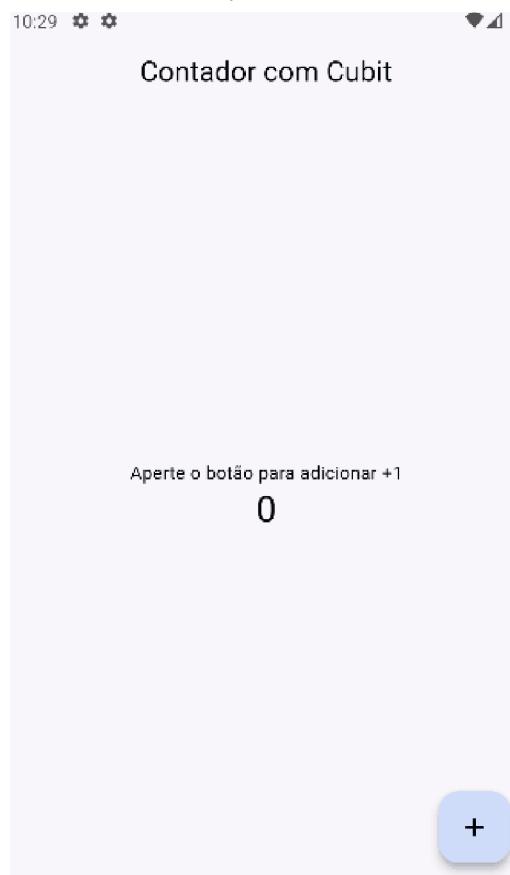
**Que tal viajar para o passado e usar seus novos conhecimentos em**

gerenciamento de estados para  
recriar a mesma aplicação,  
mas usando Cubit no lugar de  
`StatefulWidget` ?

Vou te deixar algumas dicas:

1. Crie o Cubit para o contador;
  2. Crie a instância desse Cubit na sua página de contador;
  3. Chame a instância do Cubit para:
    - a) Notificar que ocorreu uma mudança de estado;
    - b) Atualizar o valor do contador na tela.
- 
1. Por fim, a maior dica e conselho que posso dar:
    - a) Estude a [documentação](https://pub.dev/packages/flutter_bloc) ([https://pub.dev/packages/flutter\\_bloc](https://pub.dev/packages/flutter_bloc)) para entender melhor e aperfeiçoar seus conhecimentos.

Ao final, você deverá ter uma aplicação parecida com esta:



*Lembrando que a interface não precisa ser parecida ou igual! Apenas o funcionamento do contador. Ah, e compartilhe na nossa comunidade o resultado do seu desafio. Vamos adorar ver como ficou.*

## Opinião do instrutor



E aí, conseguiu resolver o desafio? Espero que sim!

E como também gosto de desafios, vou deixar aqui o passo a passo de como fiz:

1. Instalei as bibliotecas

```
bloc e flutter_bloc ;
```

2. Criei um arquivo chamado

```
counter_cubit.dart ;
```

3. Em `counter_cubit.dart`

criei uma classe

`CounterCubit` que vai estender de `Cubit` e vai lidar com estados do tipo inteiro ( `int` ).

Não se esqueça de

importar a biblioteca

```
bloc/bloc.dart ;
```

1. Além disso, a classe

`CounterCubit` deve começar com valor `0` e ter uma função que faz o `emit` (atualiza) do estado, incrementando +1 ao estado atual:

```
import 'package:bloc
```

```
class CounterCubit e  
CounterCubit() : s
```

```
void increment() =  
}
```

[COPIAR CÓDIGO](#)

1. Agora na tela do aplicativo,  
no arquivo `main.dart` ,  
criei uma instância de  
`CounterCubit` :

```
final counterCubit =
```

[COPIAR CÓDIGO](#)

1. E para atualizar o valor do  
`Text` , criei o  
`BlocBuilder` ,  
especificando o BLoC que  
queria utilizar (e não  
esqueça de importar a  
biblioteca  
`flutter_bloc/flutter_bloc.dart` ):

```
BlocBuilder<CounterC  
  bloc: counterCub  
  builder: (context  
    return Text(  
      state.toStri  
      style: Theme  
    );  
  },
```

```
),
```

[COPIAR CÓDIGO](#)

1. Por fim, dentro do

```
onPressed de
```

```
floatingActionButton ,
```

chamei a função de

incremento que criei

usando a instância de

```
CounterCubit :
```

```
counterCubit.increme
```

[COPIAR CÓDIGO](#)

Dessa maneira, consegui fazer o contador funcionar perfeitamente!

Se quiser conferir os arquivos da aplicação completos, vou deixá-los logo abaixo:

- Arquivo `main.dart` :

```
import 'package:flutter/
```

```
import 'package:flutter_
```

```
import 'counter_cubit.da
```

```
void main() {  
  runApp(const MyApp());  
}
```

```
class MyApp extends StatelessWidget {  
  const MyApp({super.key});  
  
  @override  
  Widget build(BuildContext) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      title: 'Flutter Counter',  
      theme: ThemeData(  
        useMaterial3: true,  
        colorSchemeSeed: Colors.teal,  
      ),  
      home: CounterPage(),  
    );  
  }  
}
```

```
class CounterPage extends StatelessWidget {  
  const CounterPage({super.key});  
  
  final counterCubit = CounterCubit();  
  
  @override  
  Widget build(BuildContext) {  
    return Scaffold(  
      appBar: AppBar(  
        centerTitle: true,  
        title: const Text('Flutter Counter'),  
      ),  
    );  
  }  
}
```

```
body: Center(  
  child: Column(  
    mainAxisAlignmentAlignm  
    children: <Wid  
    const Text(  
      'Aperte o  
    ),  
    BlocBuilder<  
    bloc: coun  
    builder: (  
      return T  
      state.  
      style:  
    );  
  },  
),  
],  
),  
),  
floatingActionButt  
  onPressed: () {  
    counterCubit.i  
  },  
  tooltip: 'Increm  
  child: const Ico  
),  
);  
}
```

[COPIAR CÓDIGO](#)



- Arquivo

```
counter_cubit.dart :
```

```
import 'package:bloc/blo
```

```
class CounterCubit exten  
  CounterCubit() : super
```

```
void increment() => em  
}
```

**COPIAR CÓDIGO**

Caso tenha ficado dúvidas  
na solução do desafio,  
procure nossa  
comunidade do Discord ou  
o fórum do curso.