

SUMÁRIO

1. GERENCIAMENTO DE ESTADOS NO FLUTTER	1
2. CLASSIFICAÇÃO DE WIDGETS POR ESTADO	1
2.1. STATELESSWIDGET	1
2.1.1. QUANDO E COMO CRIAR UM STATELESSWIDGET	1
2.2. STATEFULWIDGET	2
2.2.1. QUANDO E COMO CRIAR UM STATEFULWIDGET	2

1. GERENCIAMENTO DE ESTADOS NO FLUTTER

Basicamente, um estado é uma informação ou grupo de informações que são alteradas durante o tempo de execução do aplicativo.

Essa alteração das informações em tempo de execução é chamada de mudança de estado. Podemos entender o estado como uma variável que persiste os dados atribuídos durante todo o tempo de execução, e enquanto ela não sofrer uma alteração.

Um estado, sendo uma variável, pode conter um conteúdo que por si só é expresso no aplicativo, ou ser um valor referencial utilizados em condicionais. Por exemplo, um state pode ser utilizado para reter o tipo e a validade de um token JWT, e servir de valor referencial para a permitir ou não o acesso de um usuário ao sistema.

2. CLASSIFICAÇÃO DE WIDGETS POR ESTADO

2.1. STATELESSWIDGET

Vamos ver agora o que é um Stateless widget da forma mais simples possível: na prática, é um widget sem o controle de estado. Este tipo de widget não possibilita alterações dinâmicas em tempo de execução. Entenda-o como algo completamente estático.

Imagine-os como tijolos em uma construção: são a base da construção, é apenas esse papel totalmente imutável que irão desempenhar durante todo o tempo de existência dela.

Podemos definir assim, para evitar todas as confusões possíveis: uma classe sem estado é uma que, após a primeira instância, não irá sofrer mudanças. Ou seja, pode receber variáveis na sua instanciação, e por isso ter uma formação variável, mas que se conclui em uma forma estática.

2.1.1. QUANDO E COMO CRIAR UM STATELESSWIDGET

Todas as classes (widgets) criadas no código do aplicativo Flutter que servem como blocos de construção, mas que **não possuem atributos que irão se alterar após a instância**, devem ser StatelessWidget.

```

Class Example extends StatelessWidget {

    //Atributos, se houverem

    //Construtor, se houverem atributos

    @override

    Widget build(BuildContext context) {

        return

    }

}

```

2.2. STATEFULWIDGET

Os widgets Stateful são o oposto dos Stateless. Eles contêm estado e isso os torna mutáveis. São eles que criam os dinamismos visuais da aplicação, como botões que alteram sua cor quando clicados, ou containers que aparecem e desaparecem em uma mesma página.

Os widgets com controle de estado são os elementos-chave para o desenvolvimento móvel da forma interativa que conhecemos.

Podemos definir assim, para evitar todas as confusões possíveis: uma classe com estado é uma que, após a instância, poderá sofrer mudanças internas em tempo de execução.

2.2.1. QUANDO E COMO CRIAR UM STATEFULWIDGET

Todas as classes (widgets) criadas no código do aplicativo Flutter que servem como blocos do seu visual, mas que possuem atributos que irão se alterar após a instância, devem ser StatefulWidget.

```

Class Example extends StatefulWidget{

    //Atributos, se houverem

    // Construtor, se os atributos forem preenchidos na instância

    @override

    State<StatefulWidget> createState() {

        return ExampleState();

    }

}

```

Um StatefulWidget ainda possui mais um passo. Um widget Stateless, que não possui estado, apenas retorna diretamente o seu conteúdo, mas um que possui, retorna a partir de uma outra classe intermediária, que é a sua state class.

```
Class ExampleState extends State<Example> {  
  
  @override  
  
  Widget build(BuildContext context) {  
  
    return  
  
  }  
  
}
```