**Aula VIII - DevAppII**

**Widgets**

**Estados**

**Layout Stateless**

**Layout Stateful**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Fontes:***

<https://flutter.dev/>

<https://material.io/design>

<https://www.flutterparainiciantes.com.br/basico/stateless-stateful>

<https://www.treinaweb.com.br/blog/flutter-widgets-stateful-e-stateless>

<https://www.devmedia.com.br/flutter-widgets-statelesswidget-e-statefulwidget/40704>

<https://www.alura.com.br/artigos/flutter-diferenca-entre-stateless-e-statefull-widget>

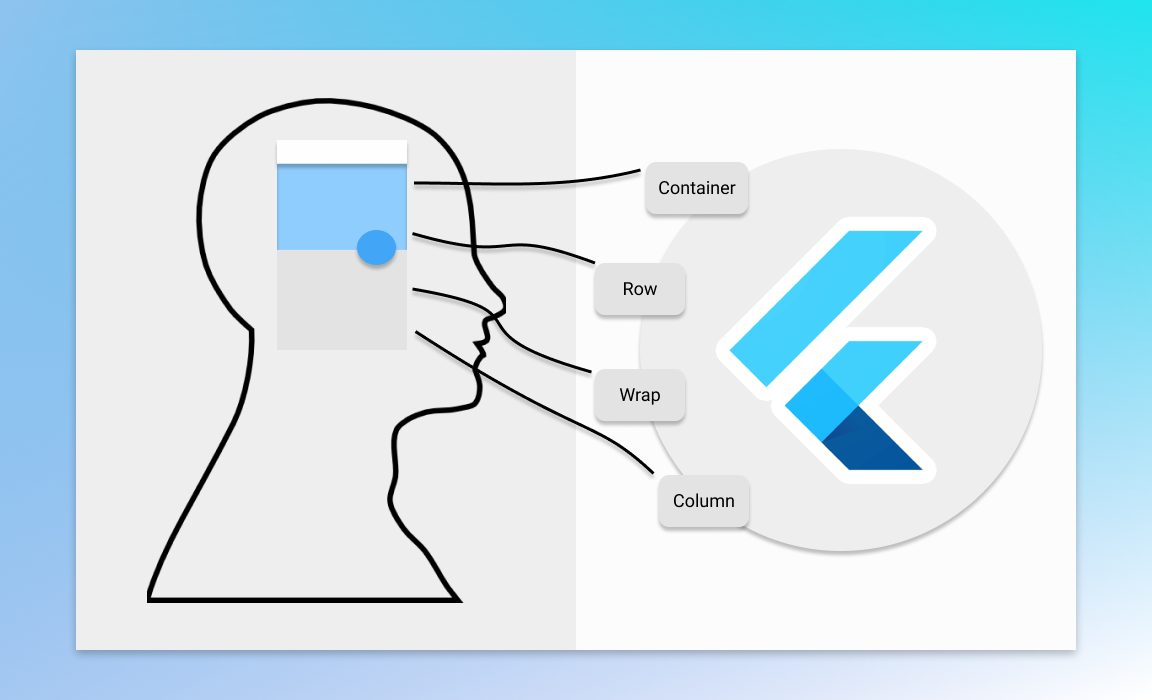
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**O que são stateless e stateful widgets?**

**A diferença mais importante é que o stateless é um widget estático, ou seja, não temos como gerenciar o estado dele. E o stateful é completamente dinâmico e nos dá o poder de torná-lo mutável através da gerência de estados.**

**Widgets**

Como vimos na aula anterior, Widget é um “componente” que usamos para montar a nossa aplicação e, antes de entrar a fundo nas diferenças entre stateless e stateful, que tal relembrar um pouco o que é widget? **Quando criamos uma aplicação inicial com Flutter, podemos observar a estrutura de código contida no arquivo main.dart.**



Rapidamente, é possível notar que existem itens que são filhos de outros itens a partir das propriedades **child ou children.** **Chamamos esses itens de widgets.** No exemplo abaixo, repare que o widget Align tem duas propriedades: alignment e child. Através do child é possível atribuir um filho, e esse widget filho é um elemento personalizado, criado por nós, chamado MeuWidget. Aqui entra um ponto interessante que já comentamos na aula anterior, o Flutter traz uma série de widgets prontos nas bibliotecas **material.dart e cupertino.dart.**

|  |
| --- |
| **Align(**  **alignment: Alignment.center,**  **child: MeuWidget(**  **width: 300,**  **height: 455,**  **decoration: BoxDecoration(**  **color: Colors.white,**  **borderRadius: BorderRadius.circular(12)**  **),**  **),**  **),** |

**Estados**

O que são os estados? Hoje em dia trabalhamos muito com “states” na parte de programação. Então, para conversarmos com os usuários, precisamos realizar mudanças no estado da aplicação. Afinal, o que são estados quando falamos de Flutter? **Basicamente, estado são os elementos que podem ser alterados quando um usuário realiza alguma ação.**

Basicamente, um estado é uma informação ou grupo de informações que são alteradas durante o tempo de execução do aplicativo. Imagine que você está no site de uma empresa e clica sobre o botão "Exibir na dashboard". **Neste momento, este botão realizará uma ação de deslizar para o lado esquerdo e modificará o texto de "Exibir na dashboard" para "Exibindo na dashboard", correto?**

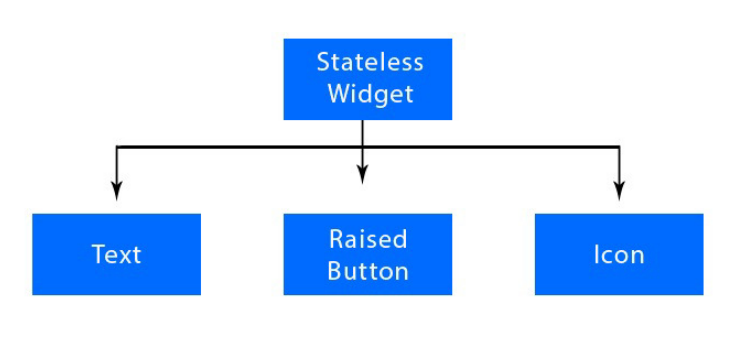
Esta alteração das informações em tempo de execução é chamada de mudança de estado, já que o valor atribuído a alguma variável de controle daquele elemento switch teve o seu valor alterado.

Em um aplicativo Flutter, a alteração de estados ocorre sempre que uma informação precisa ser alterada, seja porque um usuário realizou uma ação ou através de uma nova informação recebida de um servidor. Para criar o tipo de widget que aceita mudanças de estado, o Flutter faz uso do tipo stateful.

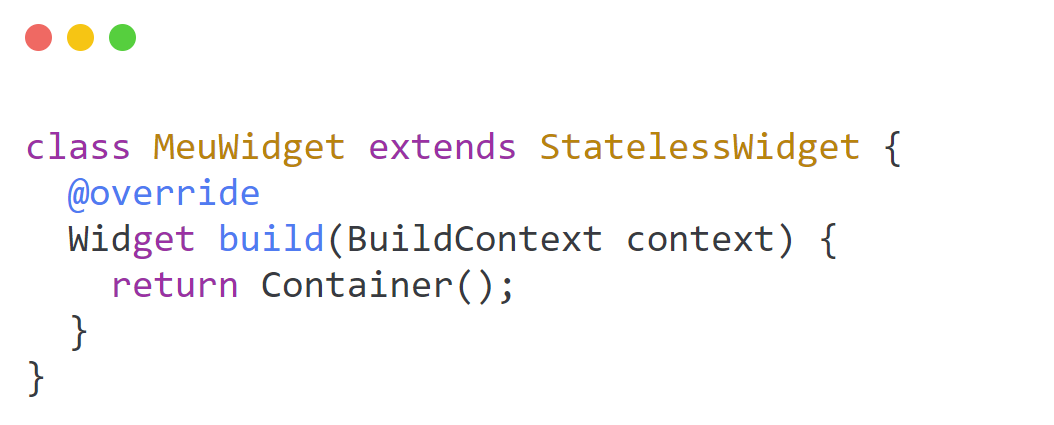
***“Porém, nem todo widget ou tela deve ter sua informação alterada em tempo de execução. Imagine a tela de política de privacidade de um app. Não tem o que mudar nesta tela, pois ela é estática e o usuário tem o poder de apenas ler o texto. Para este tipo de tela, é altamente recomendável utilizar stateless widget na sua construção, tendo em vista que o mecanismo de gerência de estados exige um pouco mais de trabalho do hardware que o stateless.”*** *(fonte:* [*https://www.alura.com.br/artigos/flutter-diferenca-entre-stateless-e-statefull-widget*](https://www.alura.com.br/artigos/flutter-diferenca-entre-stateless-e-statefull-widget) *)*

**Stateless widgets**

O Flutter contém alguns nomes diferentes para quem vem do mundo Web. Alguns são simples de compreender, mas outros nem tanto. Vamos ver agora o que é um **Stateless widget** da forma mais simples possível: **na prática, é um um widget sem o controle de estado.** Este tipo de widget não possibilita alterações dinâmicas, entenda-o como algo completamente estático. **Eles são amplamente utilizados para a criação de estruturas não mutáveis nos aplicativos (telas, menus, imagens etc.), ou seja, tudo que não envolva entradas de dados dos usuários, acessos a APIs e coisas que mudem ao longo do processo.**

.

Telas com informações jurídicas, mensagens de boas-vindas, uma imagem ou similares são exemplos de elementos que podem e devem ser construídos com Stateless. A estrutura padrão para a criação de um widget Stateless é a seguinte:



**Stateful widgets**

Os widgets Stateful são praticamente o oposto dos Stateless. Eles contêm estado e isso os torna mutáveis. **É por meio deles que construiremos boa parte das aplicações e componentes ao trabalhar com Flutter. São elementos-chave para o desenvolvimento móvel da forma interativa que conhecemos.** A sua criação pode parecer um pouco mais complexa que a dos Stateless, mas observando o processo passo a passo como faremos, você vai entender tudo perfeitamente e nem notará mais a diferença.

Ambos os tipos de widgets são bastante tranquilos de implementar. Basta sempre se perguntar se é hora de utilizar Stateless ou Stateful e seguir adiante e isso é feito por meio de um projeto. Mas, pense: ter ou não estado a ser gerenciado, o Stateful atende! Acabamos optando pelo Stateful somente quando realmente precisar gerenciar estados.

**Na criação de um widget Stateless, precisamos basicamente de uma classe estendendo a classe Stateless Widget.**

**Já para os Statefuls, precisamos criar mais um passo: temos de pensar também na estrutura que cuida do estado. O estado de um elemento basicamente controla o que é mutável nele.**

Sempre que uma variável, parâmetro ou similares mudam, automaticamente podemos comunicar ao Flutter que houve essa mudança através do método ´setState´, e ele executará uma reconstrução do widget alterado, atualizando na tela a exibição do elemento com o valor novo. A estrutura padrão para tornar um widget Stateful é:

