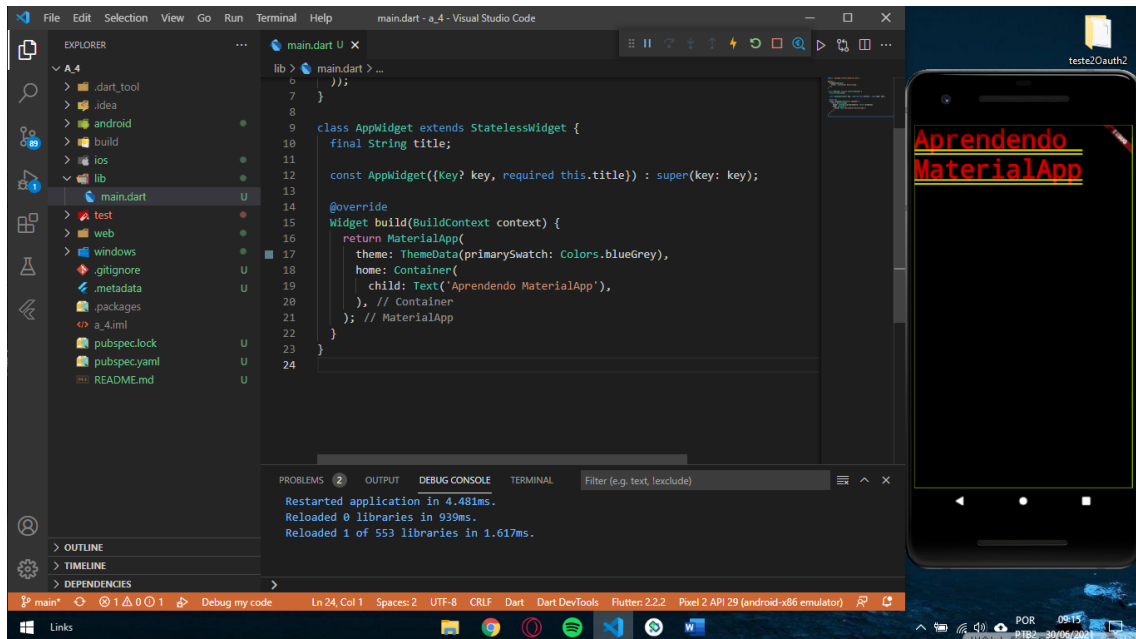
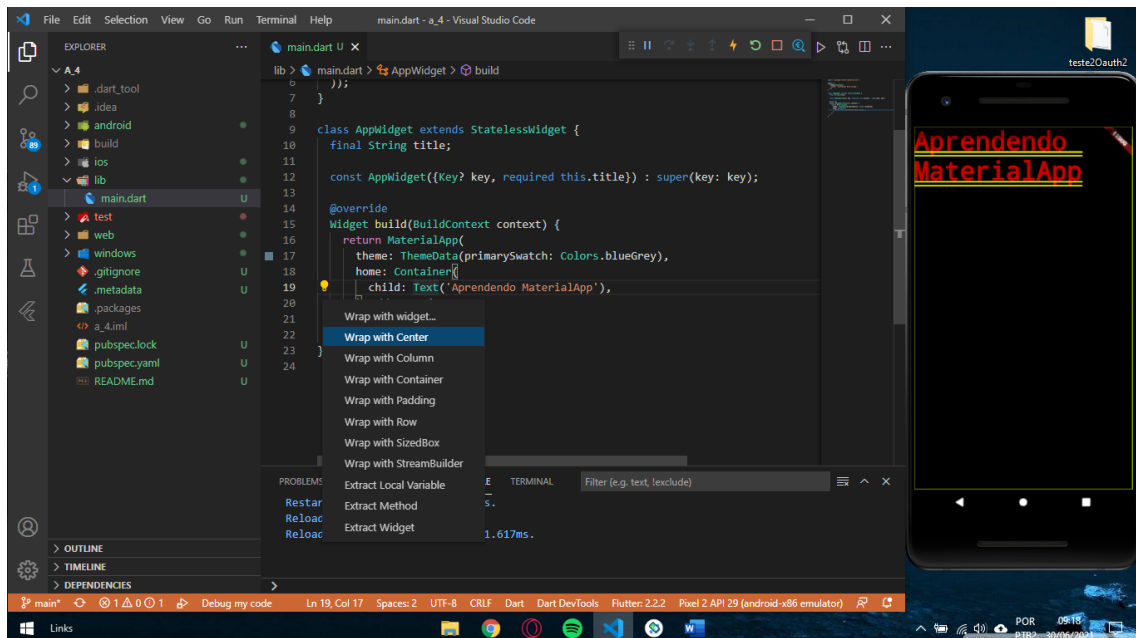


Como o Flutter tem essa estrutura de containers dentro de containers podemos definir uma hierarquia de pai e filhos ou ancestrais e sucessores, e com essa hierarquia os containers filhos podem herdar por assim dizer características dos containers pai.

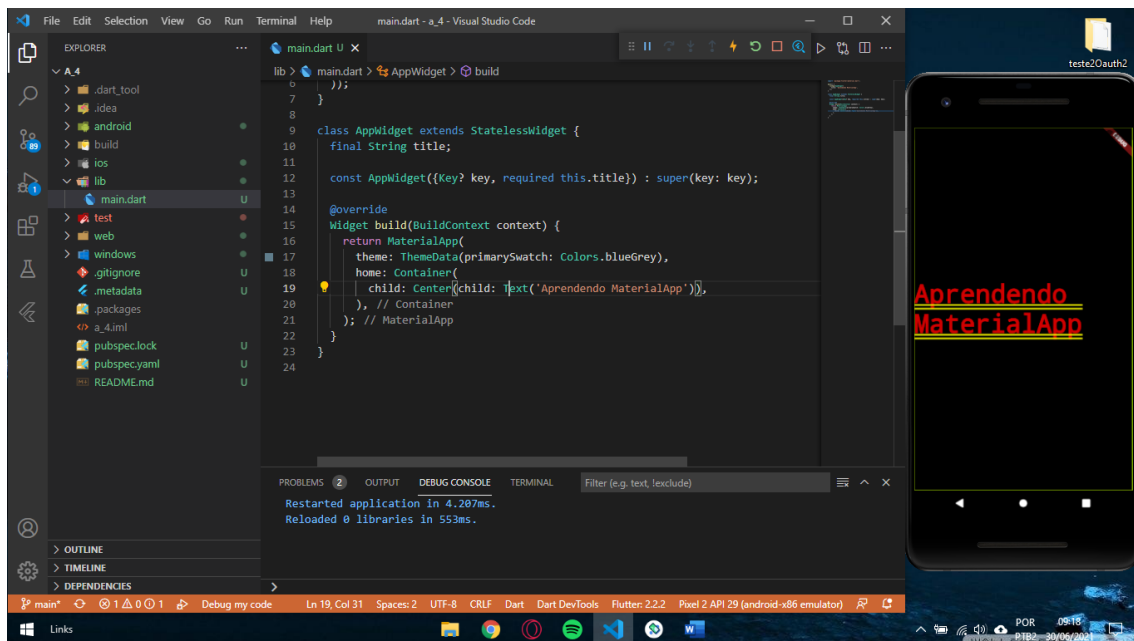
Agora vamos aprender a dar um ar mais de aplicativo ao seu aplicativo, usando o container global 'MaterialApp'.



Usamos o 'MaterialApp' definimos o tema que nesse caso seria a cor da barra superior, veremos ela mais pra frente, e para testarmos colocamos um texto na tela, note que com o 'MaterialApp' o texto apareceu na tela sem precisar de direcionais.

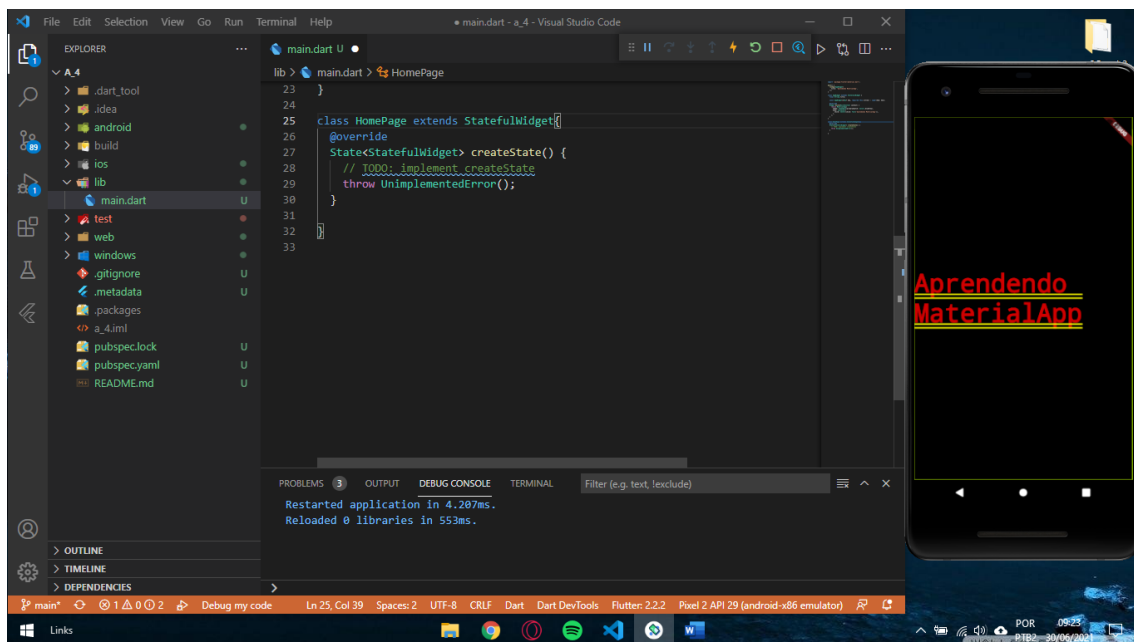


Clicando no nosso container text e na lâmpada, podemos automaticamente chamar um container pai para esse container e todos os seus sucessores.

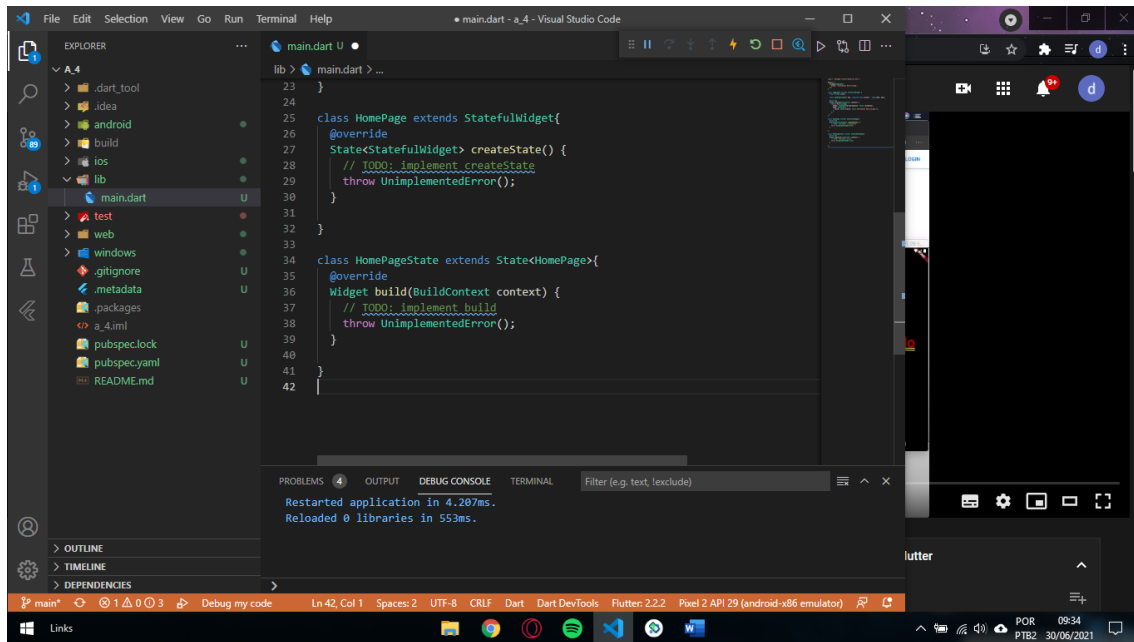


Com o nosso texto no meio da tela...

Para trabalharmos com widgets do tipo stateful para podermos manipular estados na aplicação devemos criar uma classe e obviamente herdar as características de um widget do tipo stateful.

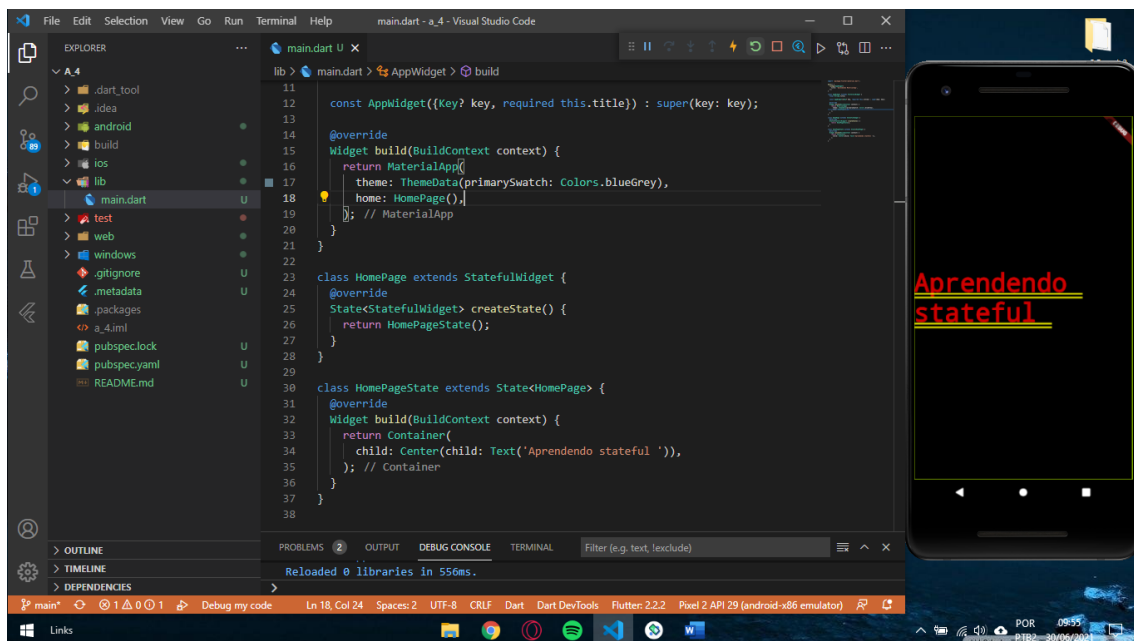


Mas os widgets do tipo stateful precisam de algo a mais para funcionarem como precisamos; precisamos criar uma classe para “gerenciar” esse estado, da verdade vai funcionar como um estado que sera modificado para podermos operar esse widget, fica mais claro praticando eu garanto.

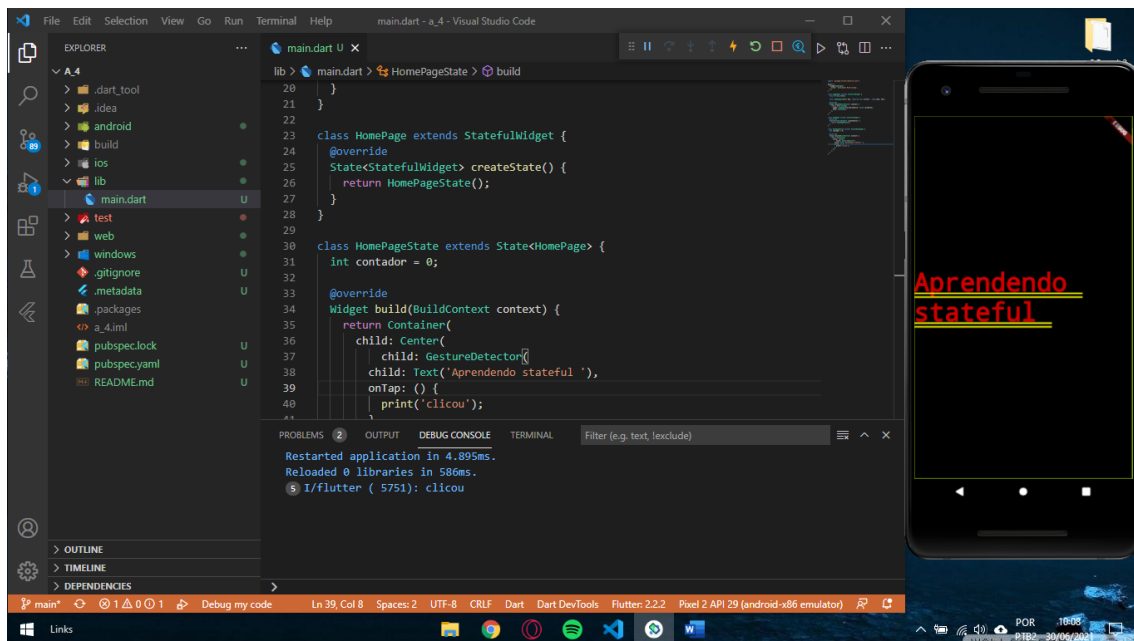


Criamos a classe de estado, e passamos a referencia da “HomePage” no ‘State’.

Devemos sempre criar para as classes do tipo stateful uma classe estado para trabalharmos com os estados que ela vai operar, criando as classes e referenciando qual classe o estado deve referenciar.

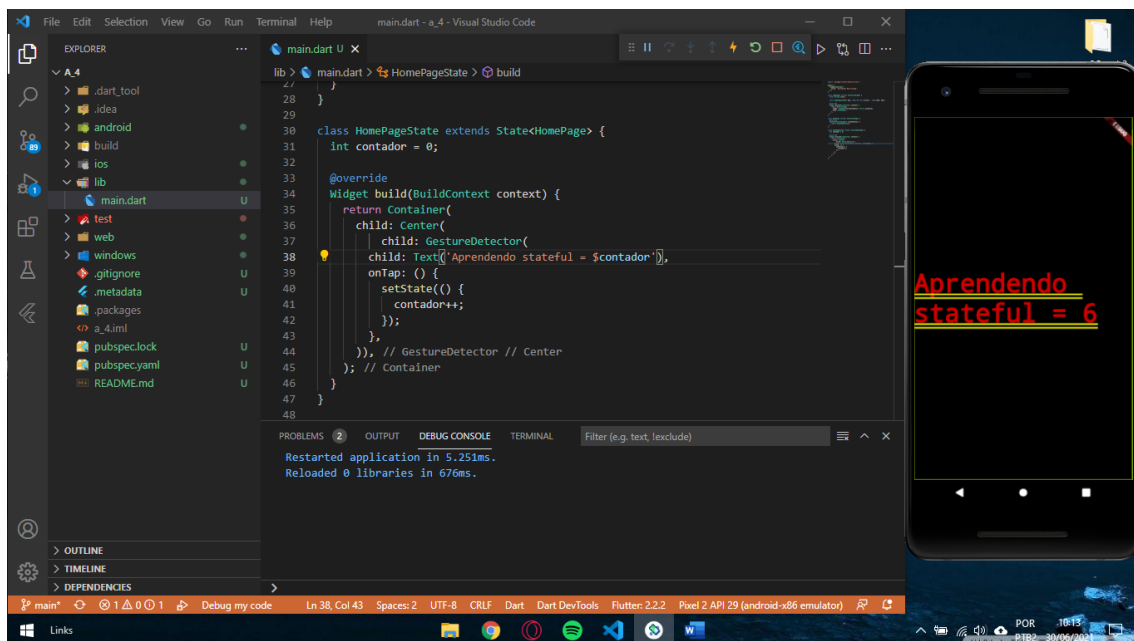


Criamos as classes, referenciamos como deveriamos, e para testar usamos o mesmo código de texto já usado, colocamos na nossa classe estado que é a responsável por construir para nós pois o build está nela, e para que ela apareça na tela referenciamos a ‘HomePage’ no ‘home’ no nosso ‘MaterialApp’.



Para testar a comunicação entre as classes criamos o método 'onTap' com o 'gestureDetector' que ira identificar cada vez que clicarmos sob o texto na tela e ira printar no console que foi clicado, como podemos ver esta tudo certo.

Mas, e se quisermos mostrar no nosso app quantas vezes foi clicado?



Primeiro fizemos a concatenação do texto na tela com a variável contador, nesse caso com o dart foi '\$'nomedavaraivel'' no nosso texto e para alterar o estado da nossa variável usamos o 'setState' se não usarmos ele o estado não sera reconstruído e não sera apresentado na tela, somente no console as alterações, usando o 'setState' colocamos para o contador incrementar e testamos.