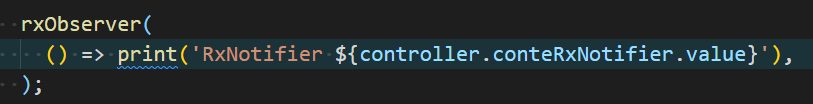
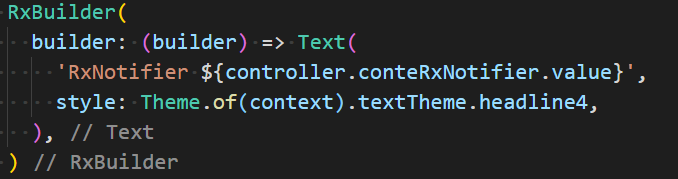
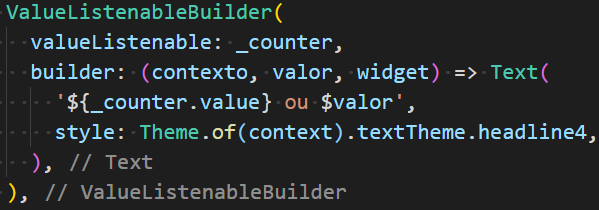
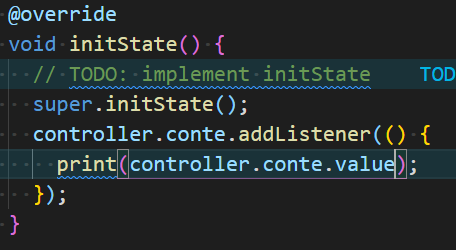
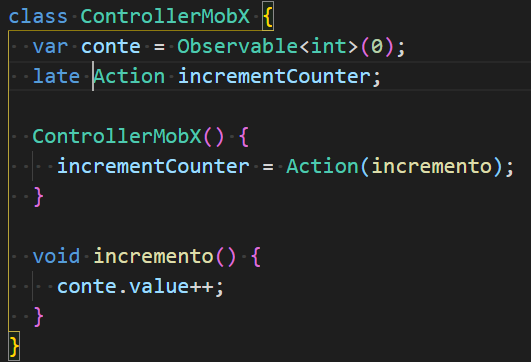
* **RxNotifier** é uma extenção do ValueNotifier(reatividade muito comum no flutter) que dá mais poderes ao ValueNotifier
  + Dá a possibilidade do valuenotifier ser transparente, igual ao mobx, sendo que não temos um limite arquitetural de precisar de uma action alí.
  + É bom mas é ruim
  + Isso não é uma gerencia de estado, mas vc pode gerenciar o estado usando ele.
  + Transparência: pq eu não preciso realemte definir o cara que eu quero escutar, so de eu ter ele dentro do escopo já é o suficiente para ele rebuidar para cada alteração.
  + Declarando: 
  + Para escutar vc pode usar na forma de ValueNotifier tbm, pq tudo é isso. Mas ele possibilita escutar desta forma
  + 
  + 
* **Server com dart**
  + Dart create server
* **Package comum**
  + Dart create –template=package-simple common
  + Se estivermos trabalhando com o flutter no front e o dart no back, podemos criar classes que são comuns, para evitar repetições
* **valueNotifier**
  + objeto em mutação: 
  + objeto que observa: 
  + outra forma de escutar:
  + 
* **COMPARATIVO COM O MOBx**
  + O mobx impõe um limite arquitetural que é pertinente pq força ter uma ação para fazer alguma alteração.
  + Mas comparado ao RxNotifier é a mesma coisa.
  + 
  + Pq o MobX faz isso? Para proibir de alterar uma observable sem ter uma ação, isso facilita a manutenção posteriormente e tbm os logs .
  + É um gerenciador de estado como o valueNotifier
* **SISTEMA DE MENSAGERIA**
  + **Socket io**
    - Socket\_io para o server

**DOCKER**

* **Se funcionar na sua máquina, vai funcionar no servidor - premissa**
* **Docker:** é uma espécie de virtualização que vc cria imagens, pequenos sistemas operacionais , vc consegue colocar um sistema operacional para rodar o seu código especifico ali.
* **Comandos**
  + Dart compile exe bin/arquivo.dart ::: basicamente compila o arquivo dart para um formato binário
  + docker build -t dart-server -f Dockerfile ../