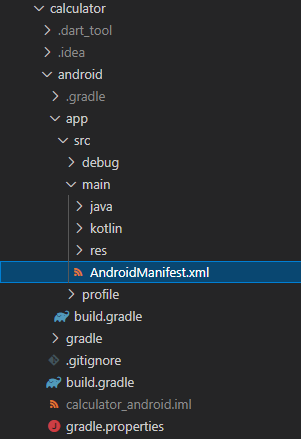
**Passo a Passo Publicação APP Flutter na loja (Android/iOS)**

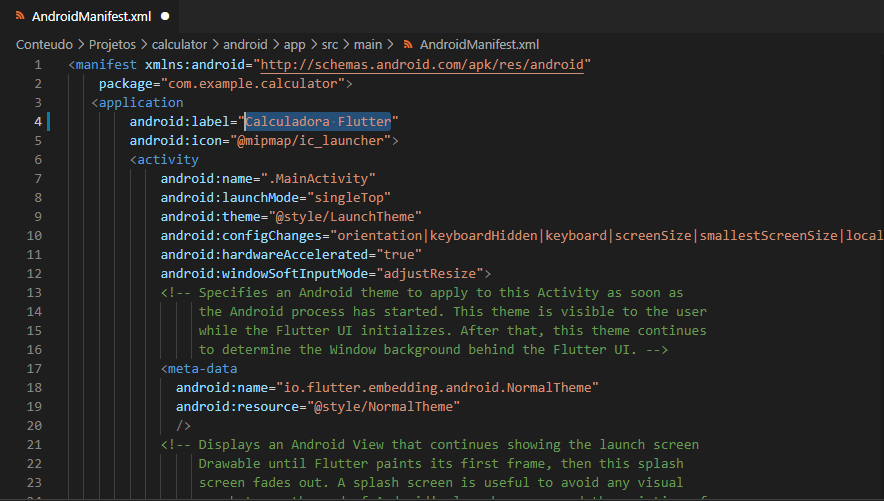
**Configurando ID da Aplicação**

**Android**

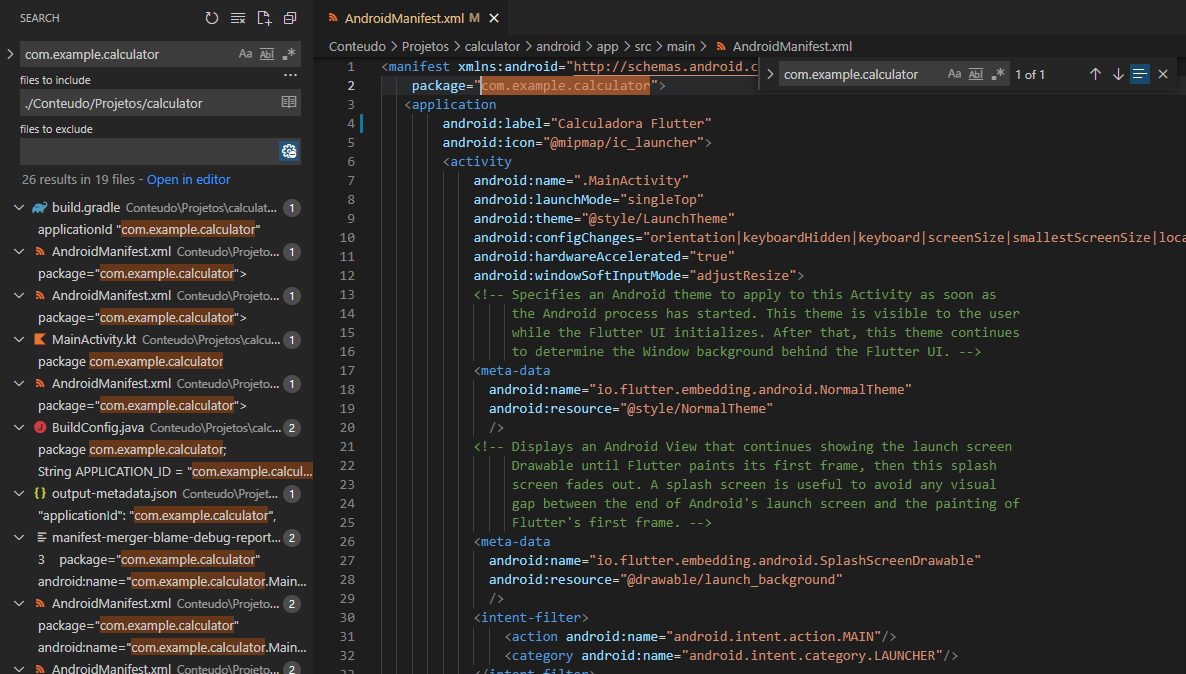
Acessar o arquivo AndroidManifest.xml no seguinte caminho



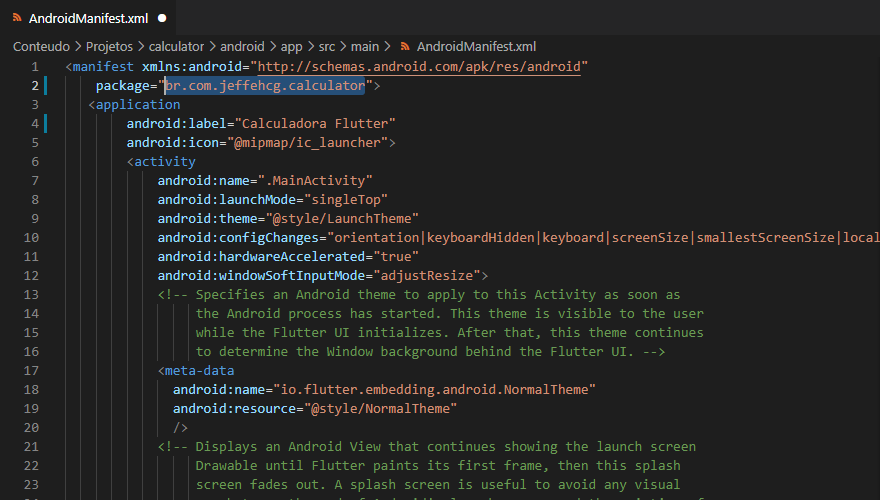
Defina o nome do projeto



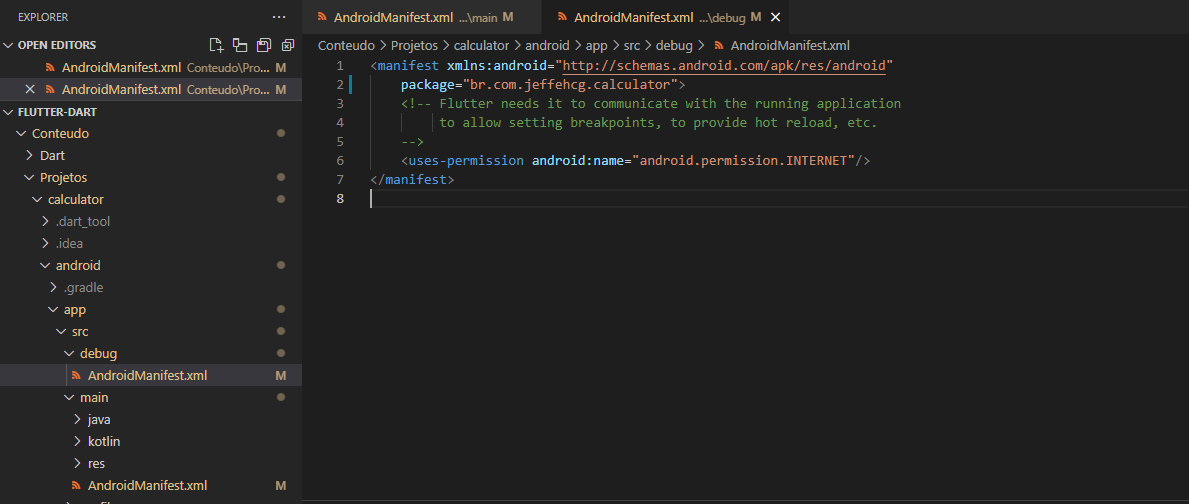
Pesquise os lugares onde o package é referenciado, pois a alteração a seguir devera ser efetuada em todos os lugares que o referencie.



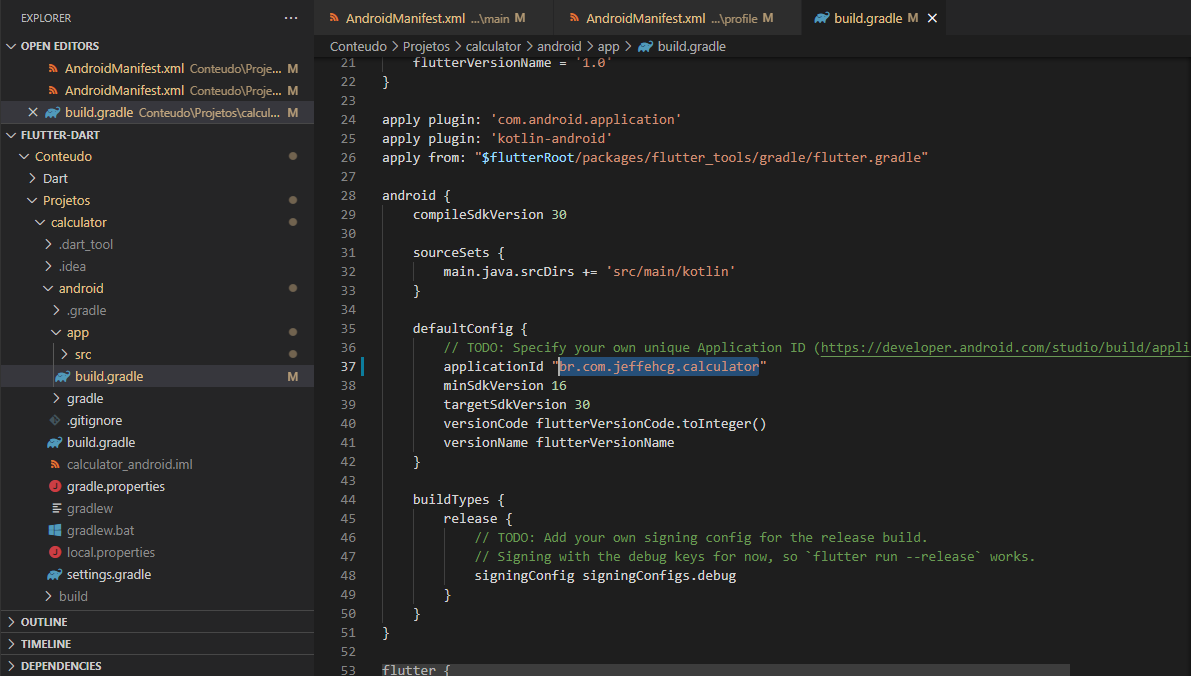
Defina o package, que será o id único da aplicação



Agora acesse o seguinte arquivo, e defina o package igual ao definido anteriormente.

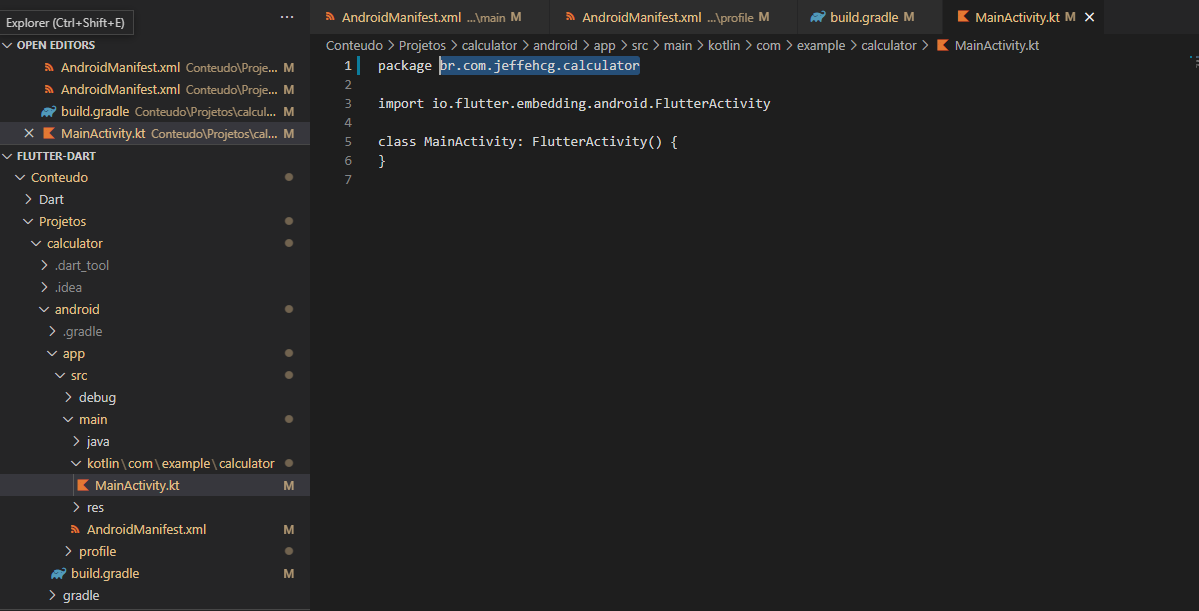


O próximo é o build.gradle

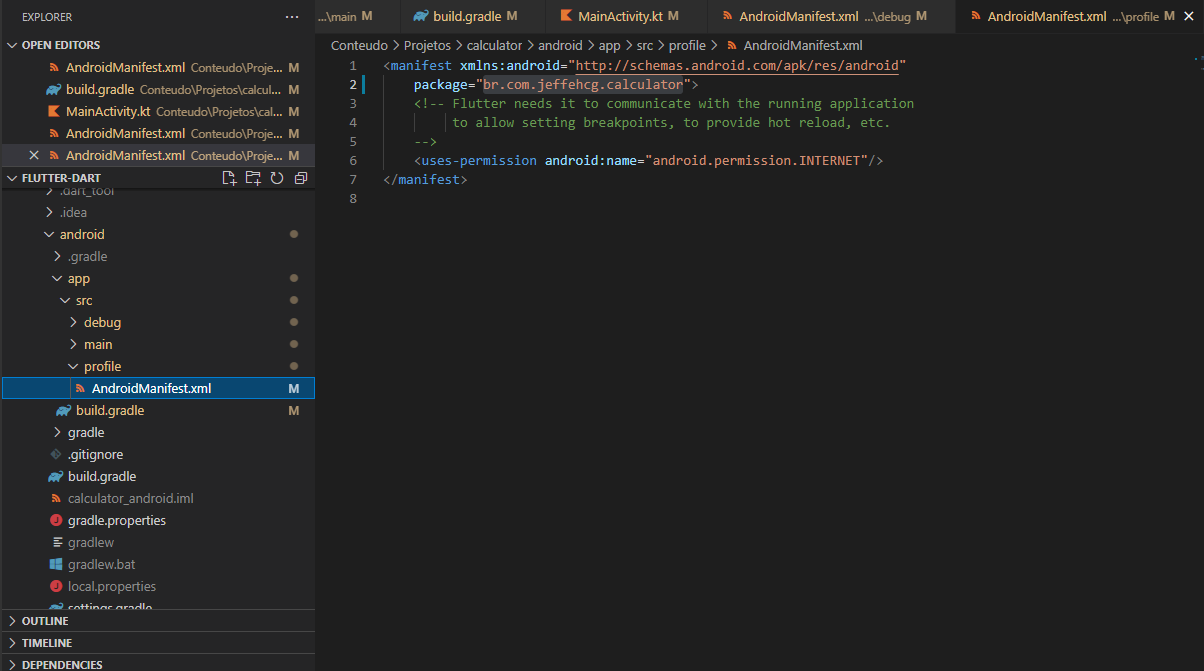


Nesse arquivo também é possível a versão mínima do android para aplicação, no caso esta na 16.

Próximo local a alterar o package é no arquivo MainActivity.kt



Agora no arquivo AndroidManifest.xml dentro de profile



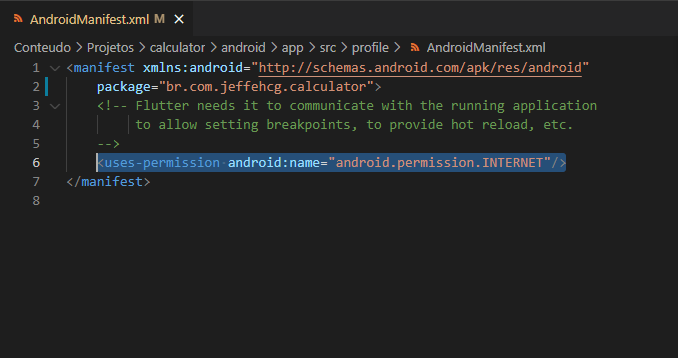
Após essas alterações efetue o redeploy da aplicação para ver se tudo irá funcionar corretamente.

\*\*Observação importante\*\* Durante a fase de desenvolvimento, quando a aplicação é buildada em modo de desenvolvedor, o flutter efetuar um merge de dois arquivos AndroidManifest.xml, um contido dentro da pasta debug, e outra dentro da pasta main.

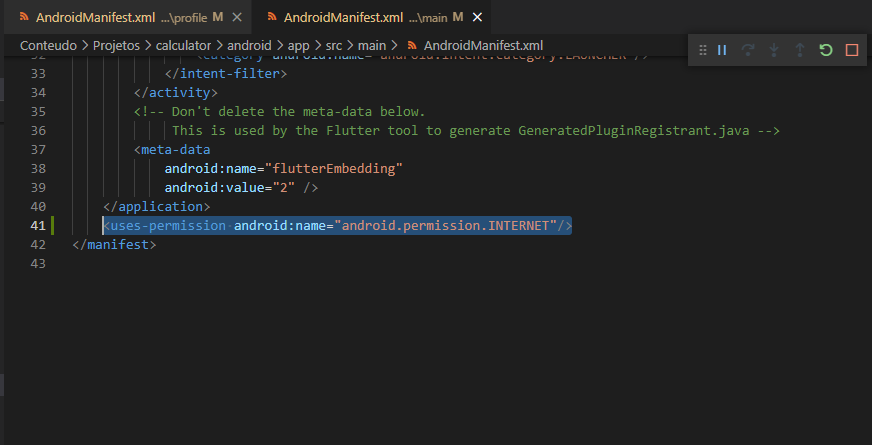


Porém quando e buildado para produção, é considerado apenas o arquivo contido na pasta main. Por tanto cuidado ao colocar configurações e permissões apenas no arquivo na pasta debug.

Por exemplo, no arquivo da pasta debug existe uma permissão para aplicação ter acesso a rede.

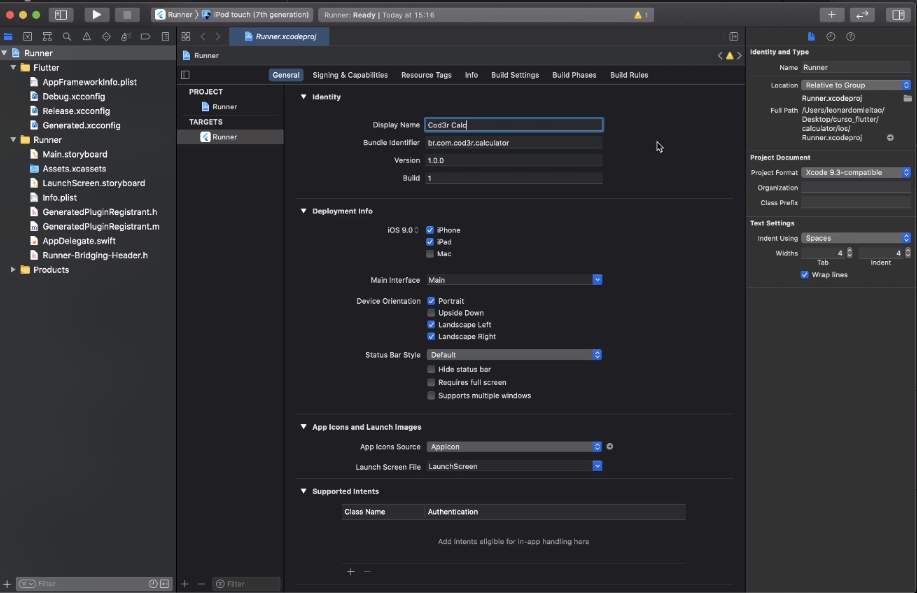


Caso sua aplicação precise acessar a rede para funcionar, adicione essa permissão também no arquivo contigo na pasta main.



**iOS**

Abra a aplicação com o XCode

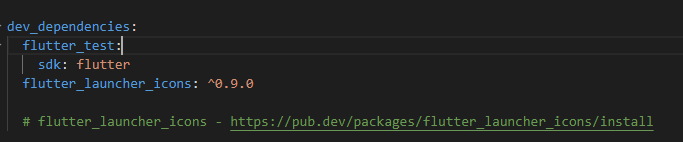


Defina o nome da aplicação, e o seu identificador único (Observação, os dados nesse print estão diferentes, pois peguei de um tutorial, não possuo um Mac).

**Adicionando Icones e Splash Screens**

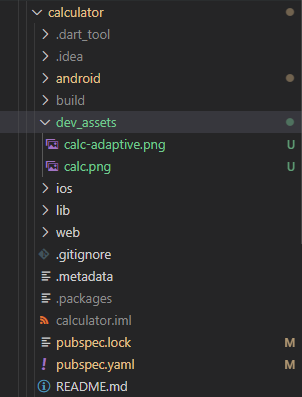
O passo a passo abaixo serve tanto pra android como para ios

Adicione a dependência “flutter\_launcher\_icons” ao projeto no arquivo pubspec.yaml



Observação, essa será uma dependência apenas de desenvolvimento, então adicione em dev\_dependencies.

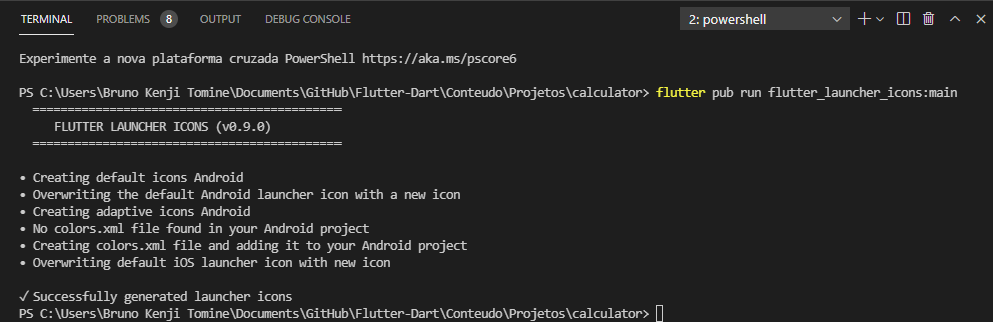
Crie uma pasta, e adicione os ícones desejados.



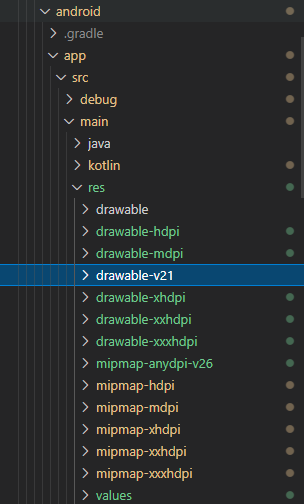
Adicione as configurações para que os ícones sejam gerados.



Após as configurações, execute o seguinte comando dentro do projeto para gerar as imagens dentro do projeto “flutter pub run flutter\_launcher\_icons:main”.



Agora dentro de res pode ser visto que os ícones foram configurados.

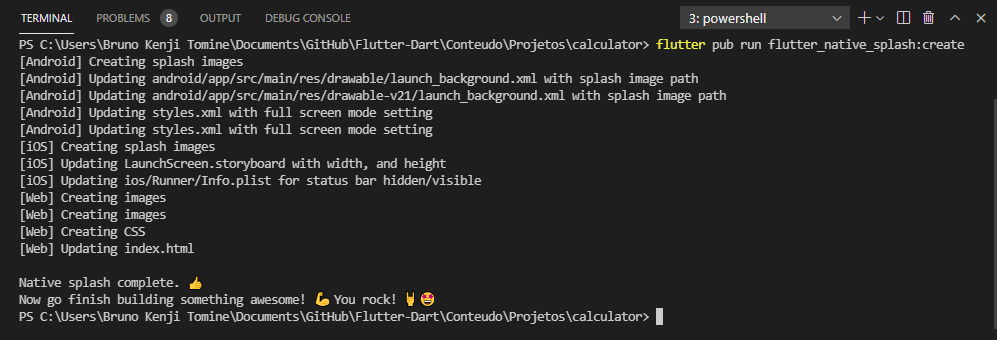


Agora instalaremos outra dependência, para efetuar a configuração da splash screen

Adicione a dependência “flutter\_native\_splash” e efetue as seguintes configurações.



Em seguida rode o comando “flutter pub run flutter\_native\_splash:create” no seu projeto



E prontinho, splash screen configurada.

**Assinando a APP**

**Criando a keystore**

Para efetuar a publicação na loja, é preciso gerar uma keystore, que é uma chave única, e só com ela será possível efetuar publicação de novas versão da sua aplicação.

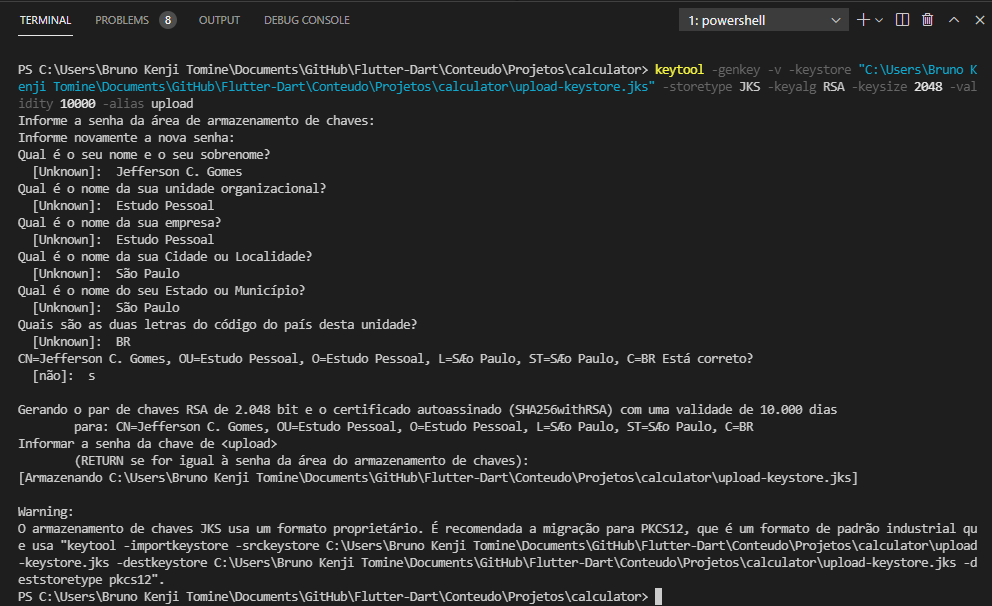
Importante, muito cuidado com essa chave, não a perca, e não salve em qualquer lugar, caso essa chave seja exposta, poderá gerar muitos problemas.

Rode o seguinte comando no terminal dentro da pasta desejada.

keytool -genkey -v -keystore c:\Users\USER\_NAME\upload-keystore.jks -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 -alias upload.

Em vez de “c:\Users\USER\_NAME\upload-keystore.jks” adicione a pasta que será salva a key

Será solicitado para adicionar uma senha, e diversos dados. Preencha os e confirme.



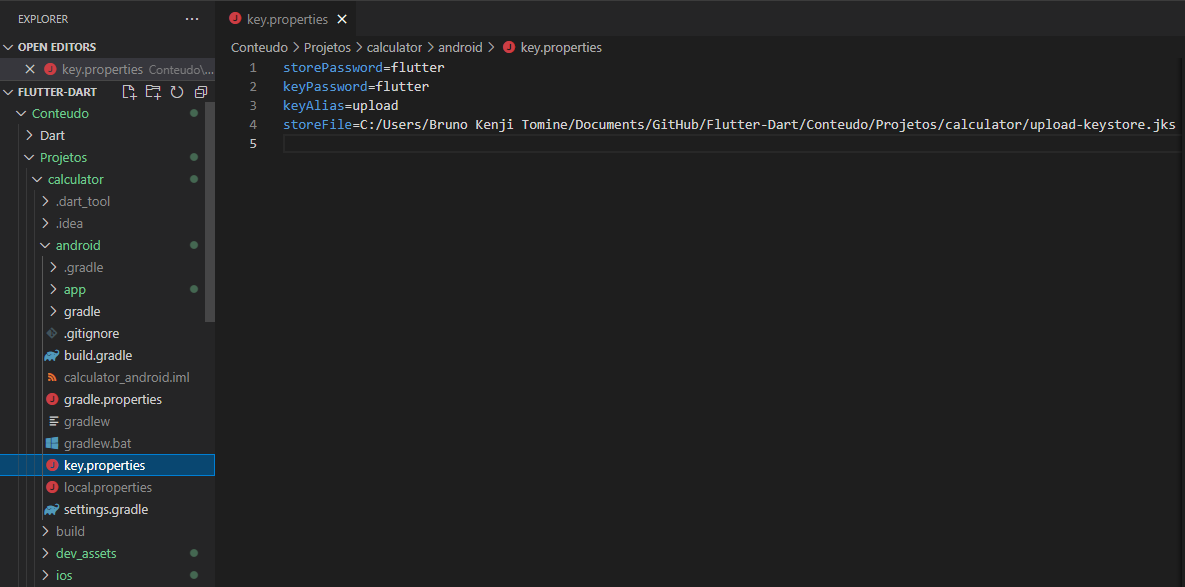
Dessa forma a chave foi gerada com sucesso. E o seguinte arquivo foi criado.



Observação, não é recomendado salvar esse arquivo na pasta do projeto criado, por questão de segurança.

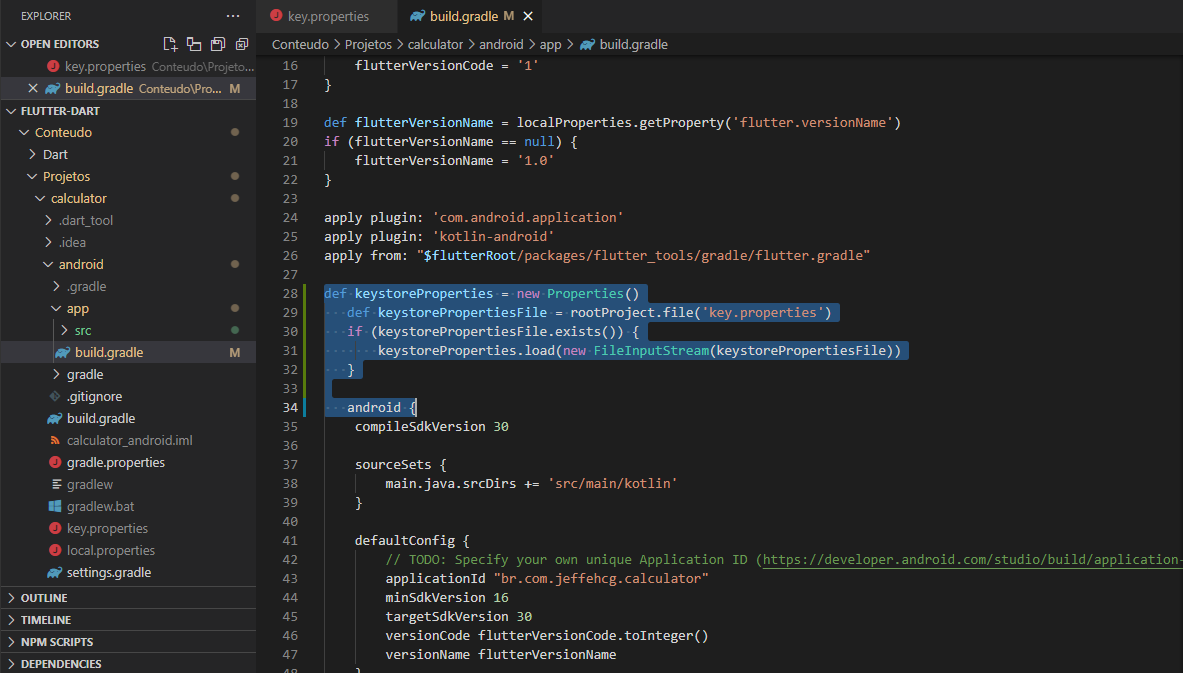
**Configuração para buildar aplicação em modo release e gerar o bundle assinado**

Crie o seguinte arquivo.

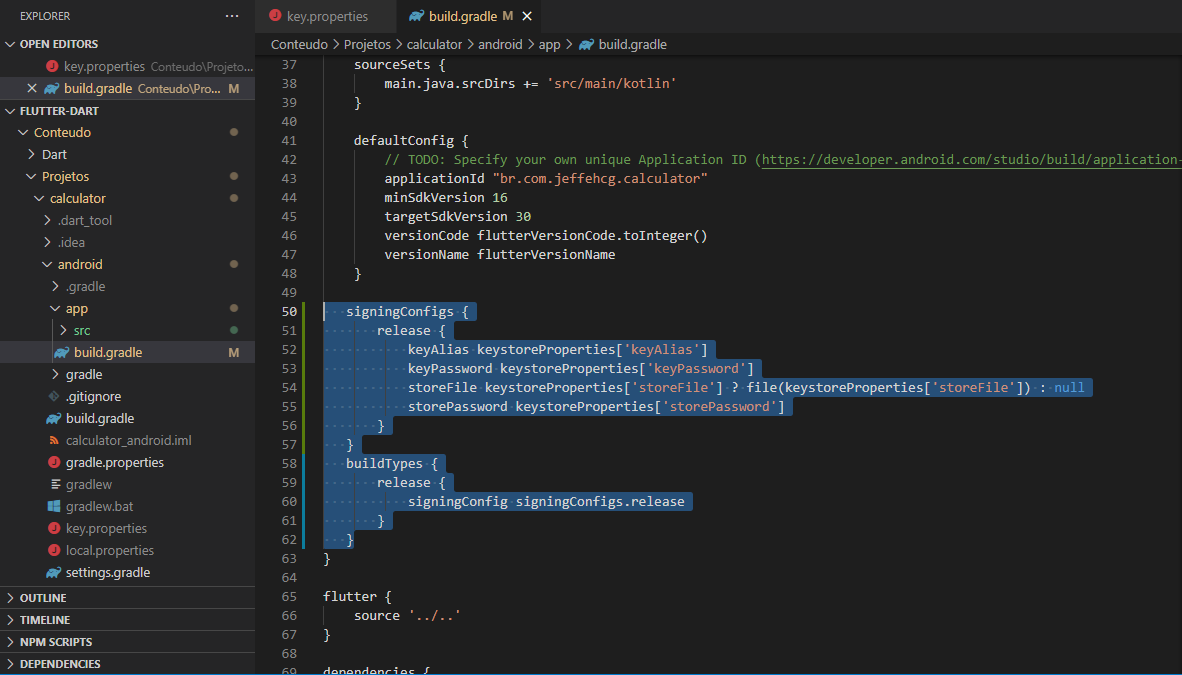


Nesse arquivo conterá a senha da key gerada anteriormente, nesse caso “flutter”, e o caminho em que a key esta localizada, obs... adicione esse arquivo no .gitignore para não ser publicado junto com o projeto no sistema de versionamento.

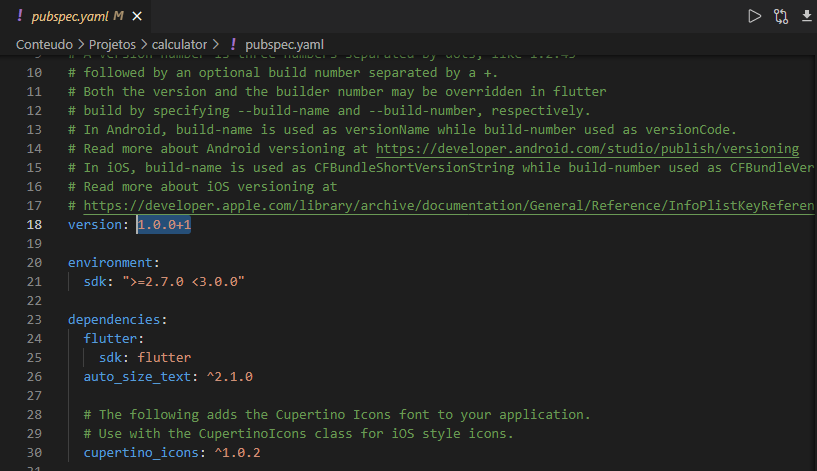
Agora atualize o arquivo build.gradle dentro app adicionando o seguinte código.



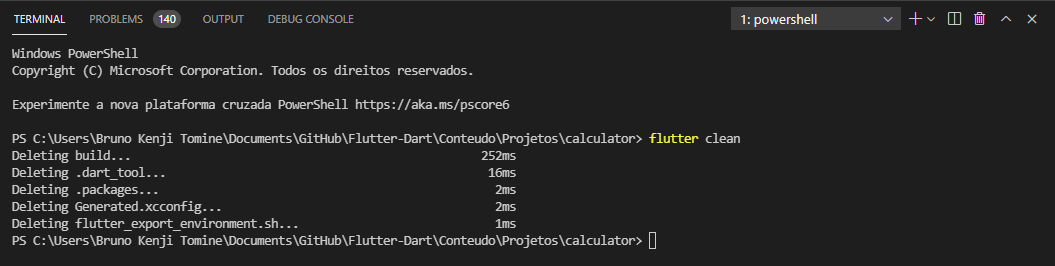
E também o abaixo.



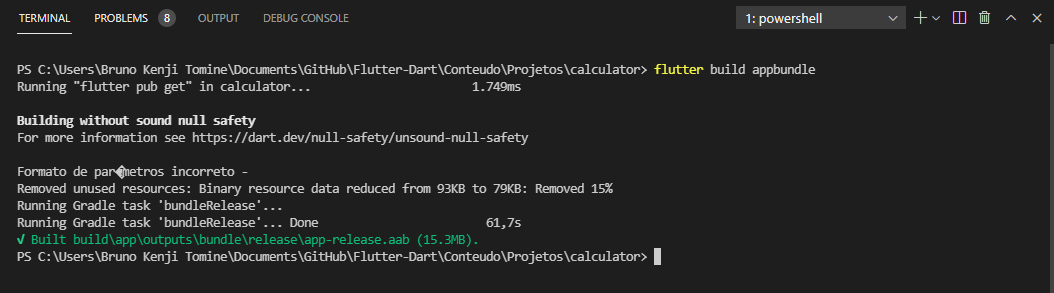
Em seguida altere o numero da versão no arquivo pubspec.yaml



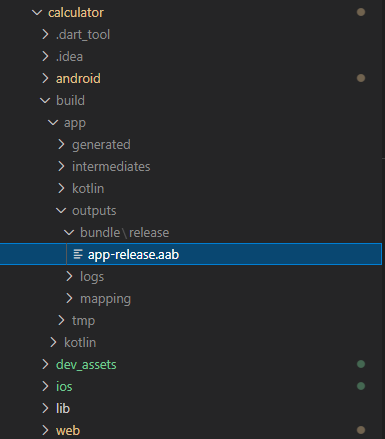
Agora rode o comando flutter clean



Agora rode o comando “flutter build appbundle”



Dessa forma o arquivo “app-release.aab” foi criado, e será publicado na loja da play store.



Para saber mais acesse o documento do flutter https://flutter.dev/docs/deployment/android

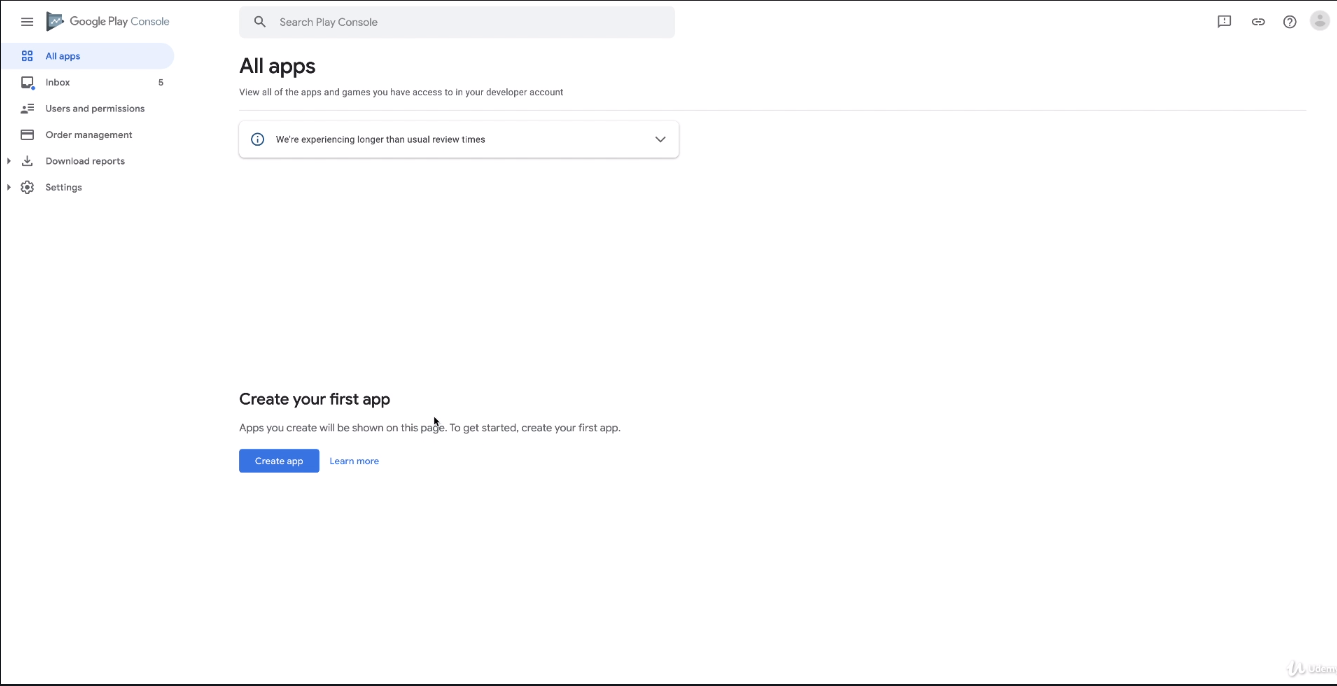
**Publicação na Play Store**

Acesse a Google Play Console <https://play.google.com/intl/pt-BR/console/about/>



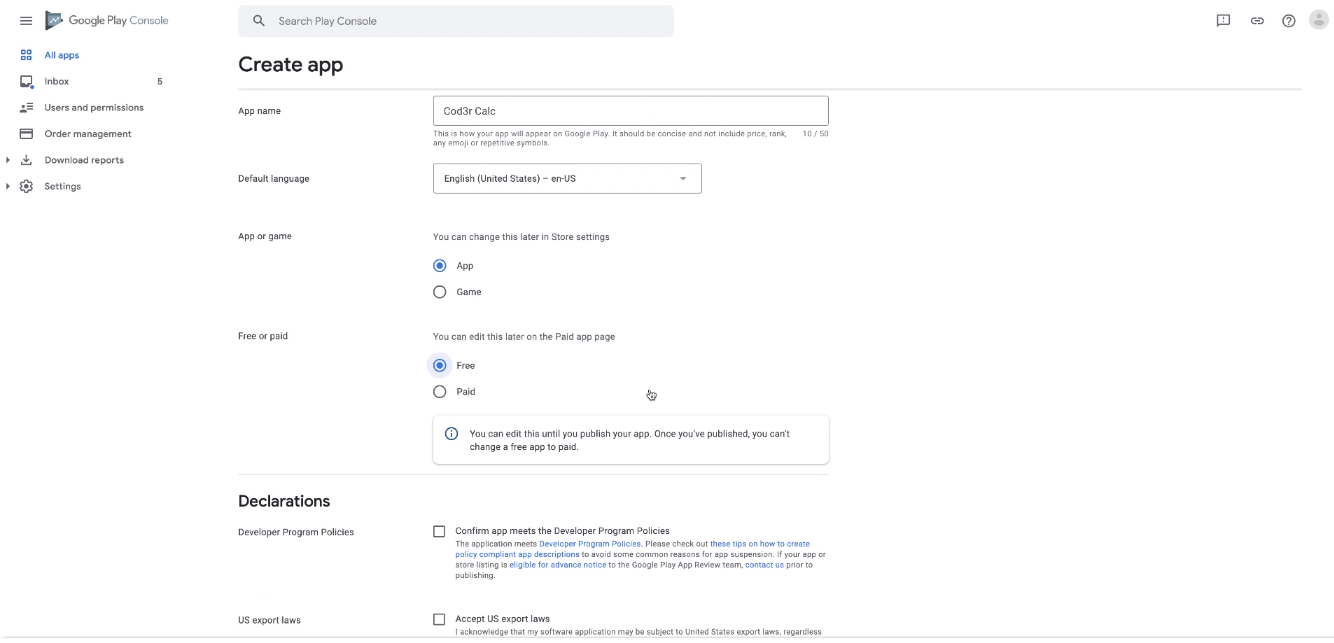
E clique em Acesso a Play Console

Observação: É preciso criar uma conta de desenvolvedor e pagar uma taxa única para a utilização do Console.

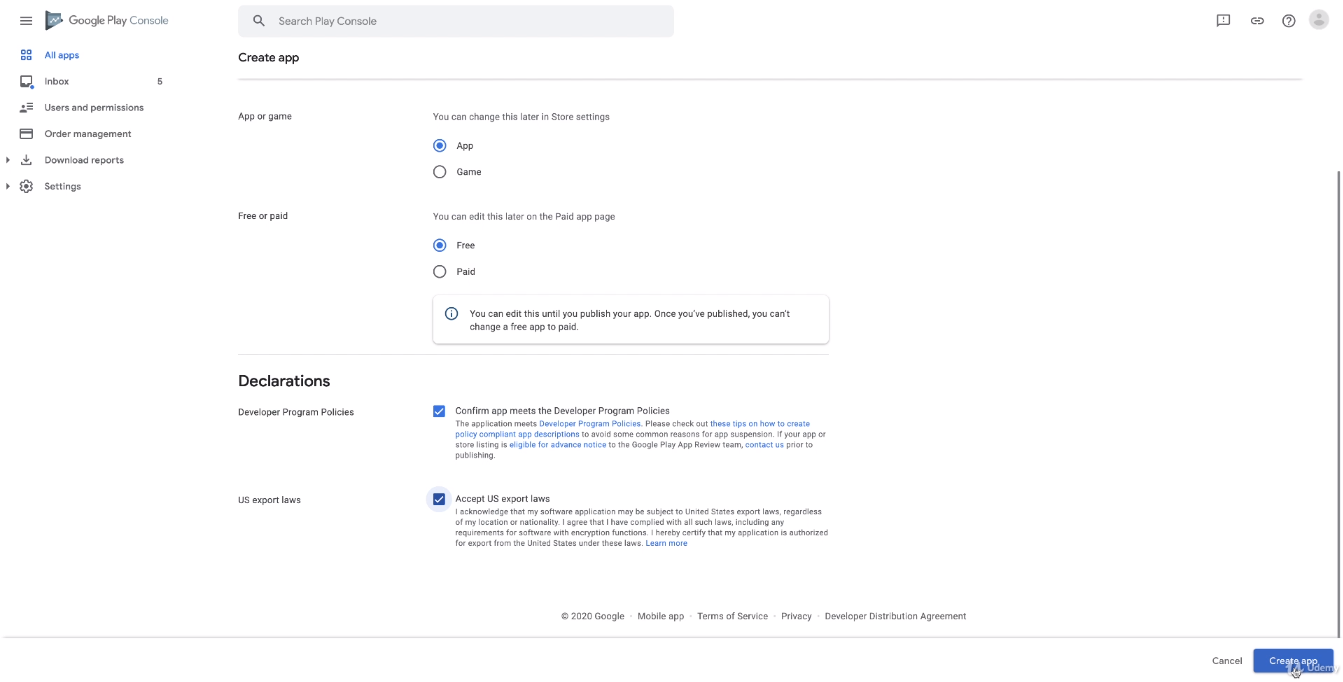


Clique em Create app.

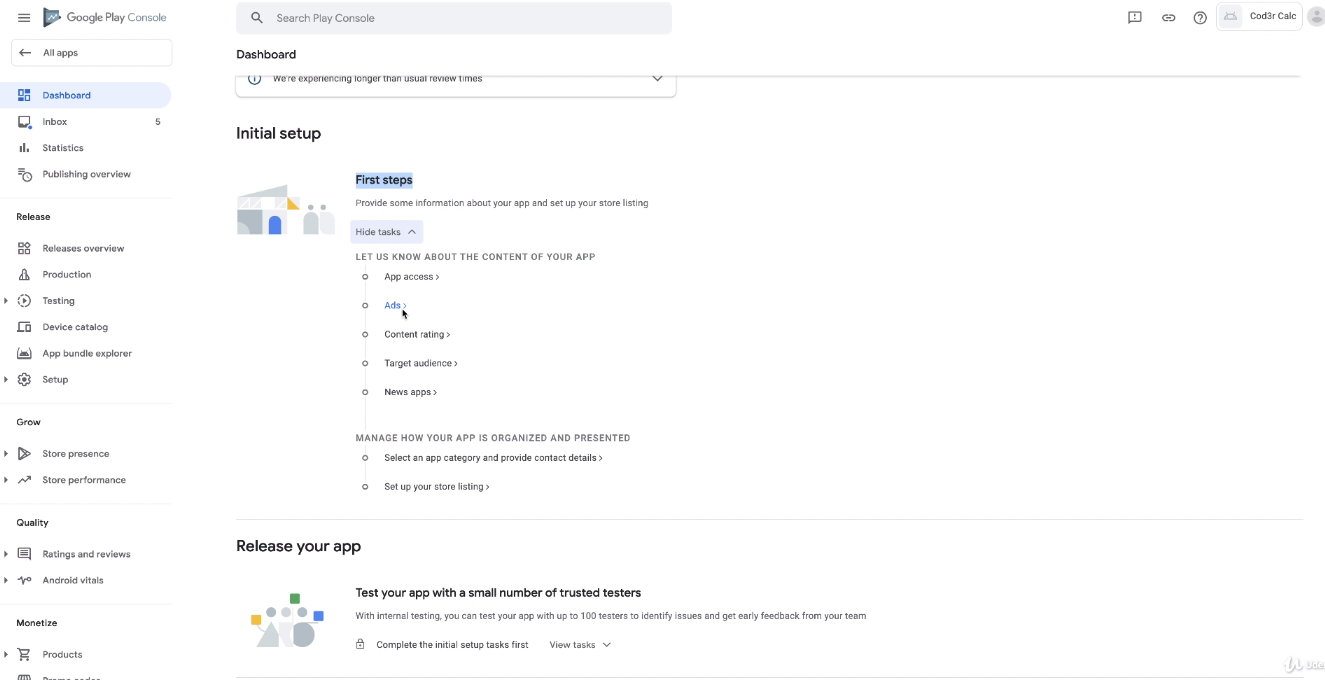
Preencha os dados solicitados.



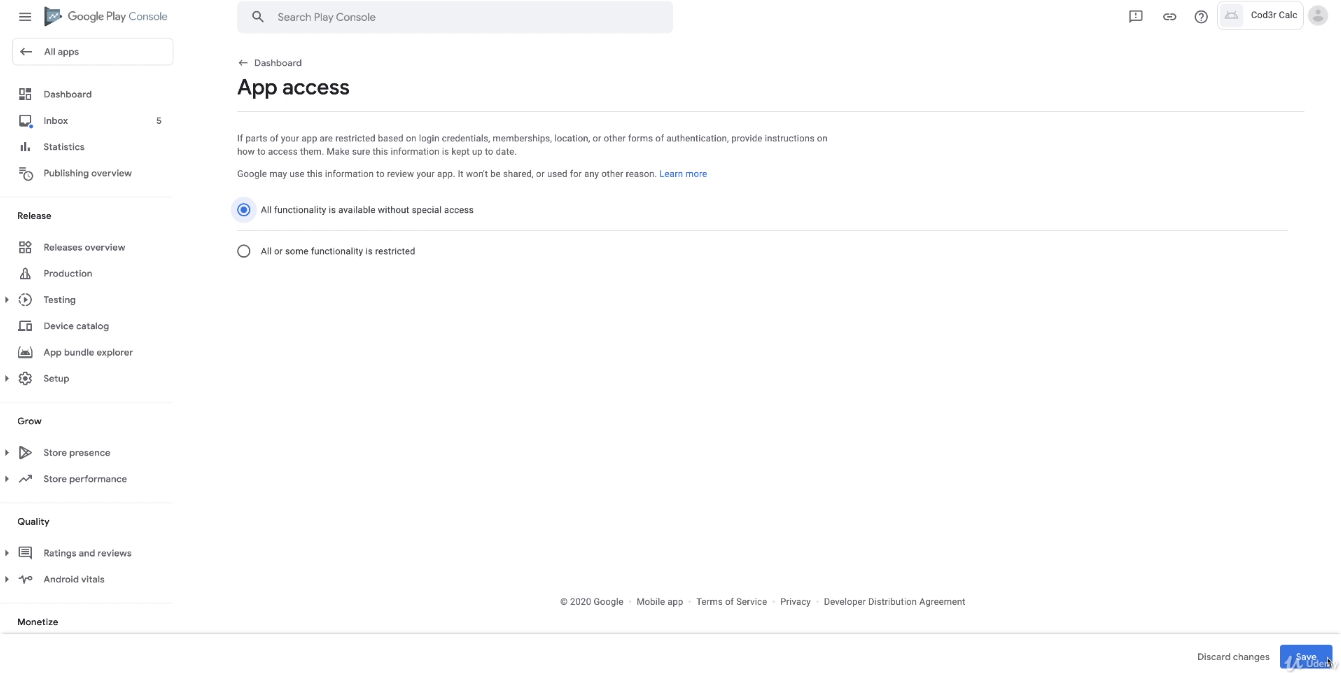
Aceite os termos de uso e clique em create app



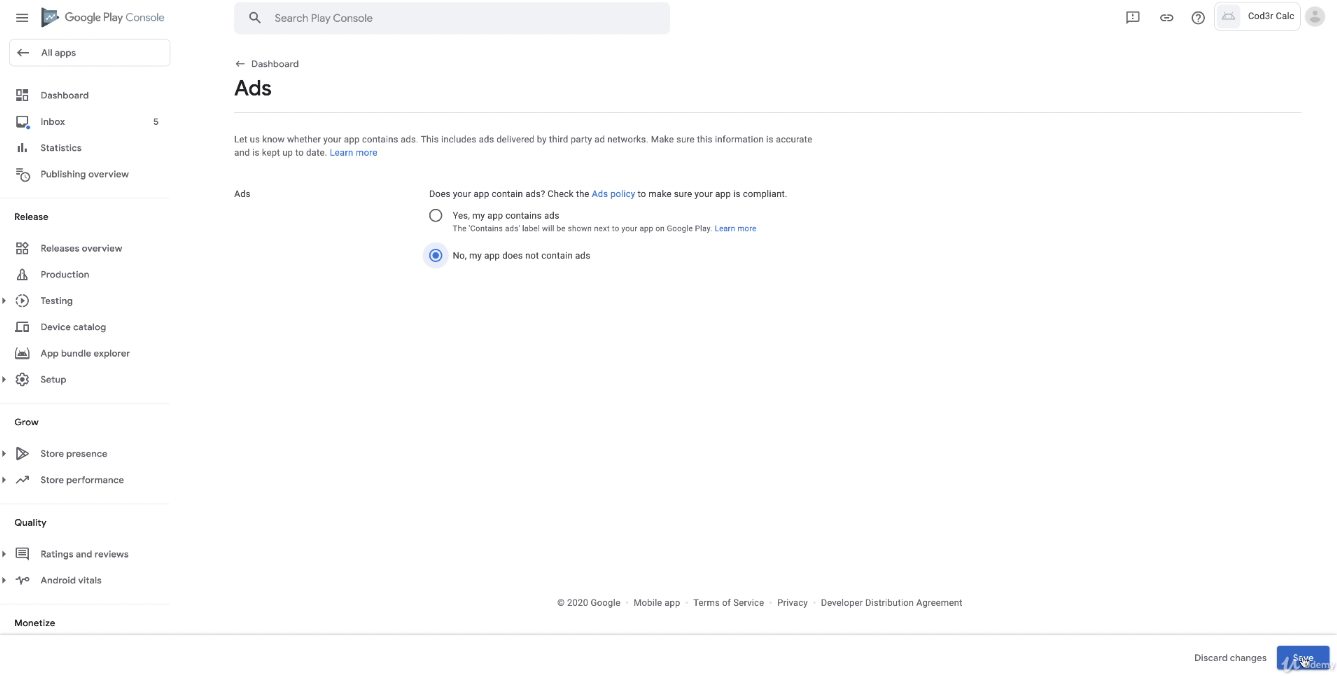
Agora que a dashboard foi aberta vá em “First steps”, e clique em view tasks, agora você irá selecionar as opções desejadas da aplicação, irei mostrar algumas como exemplo



Clique em App access. E defina que todas as funcionalidades estão disponíveis para qualquer acesso.



Agora clique em Ads, e defina que não conterá anúncios.

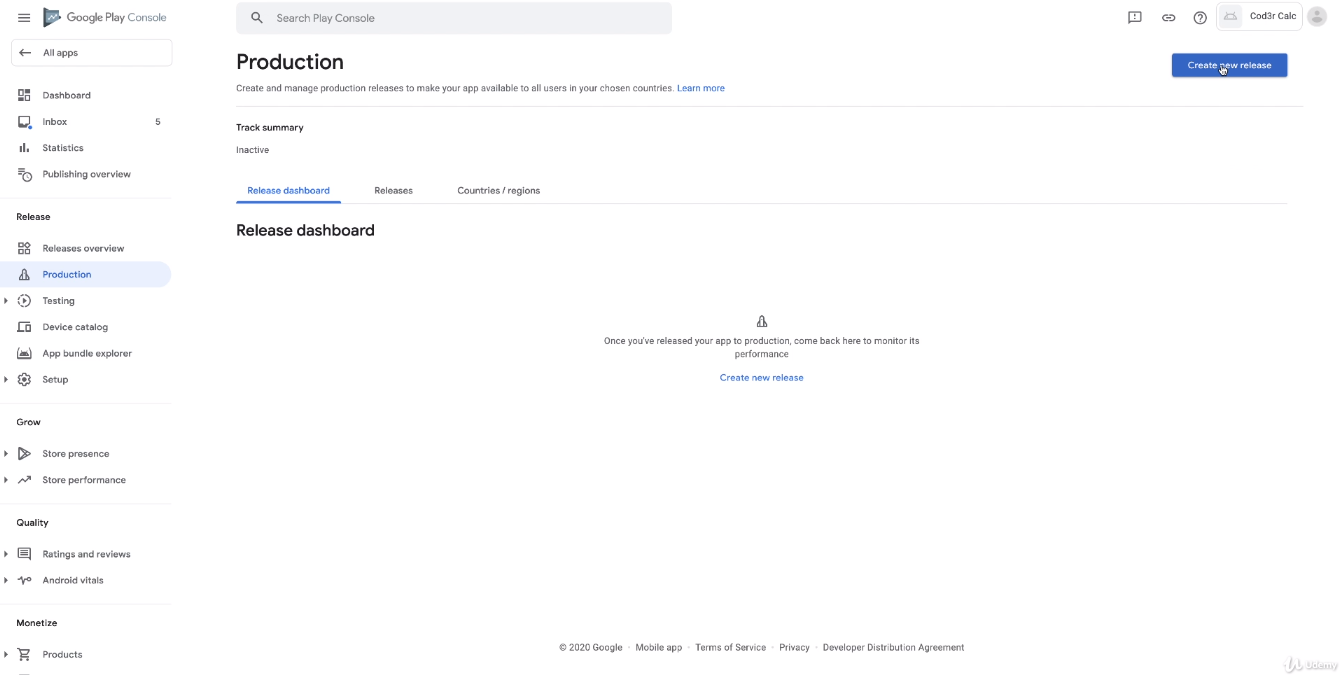


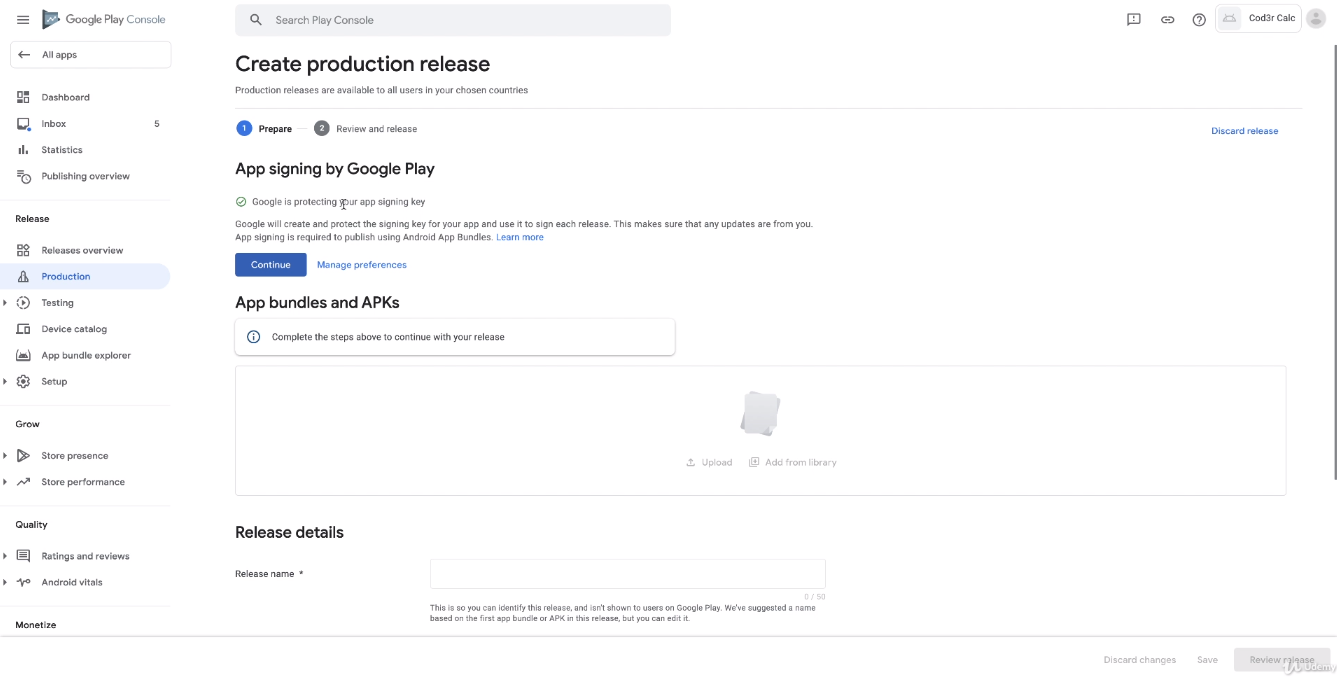
Repare que a cada passo feito, é exibida quais já foram concluídas, e quais não.



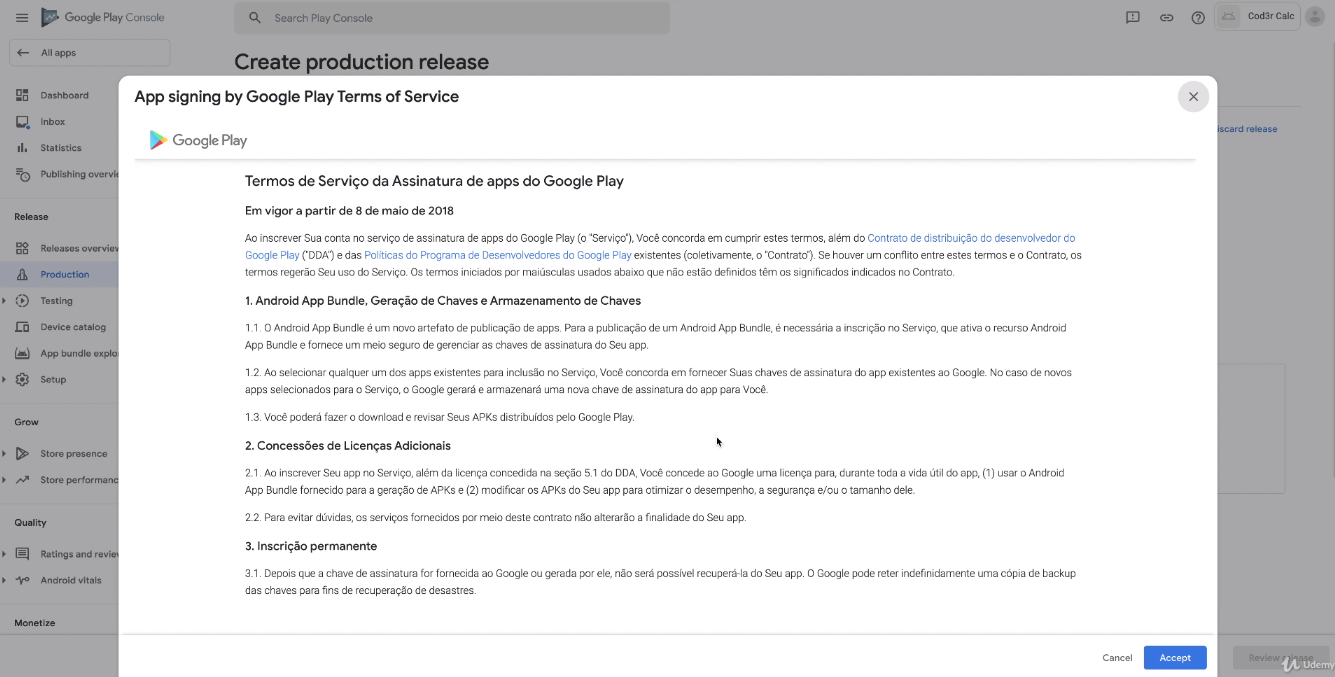
Como esses passos iniciais são bem simples, é vai de acordo com as opções de cada aplicação, irei pular os próximos.

Agora para gerar a release acesse a aba production e clique em create new release

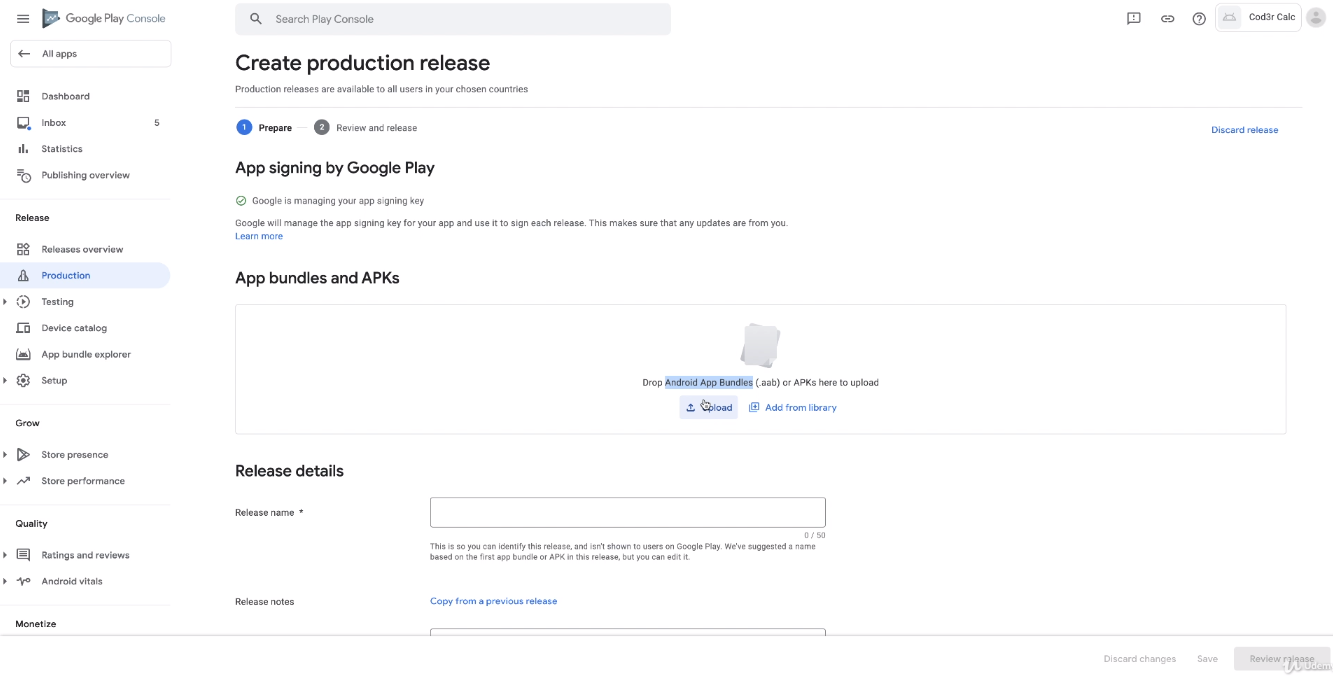




Clique em continue e aceite os termos.

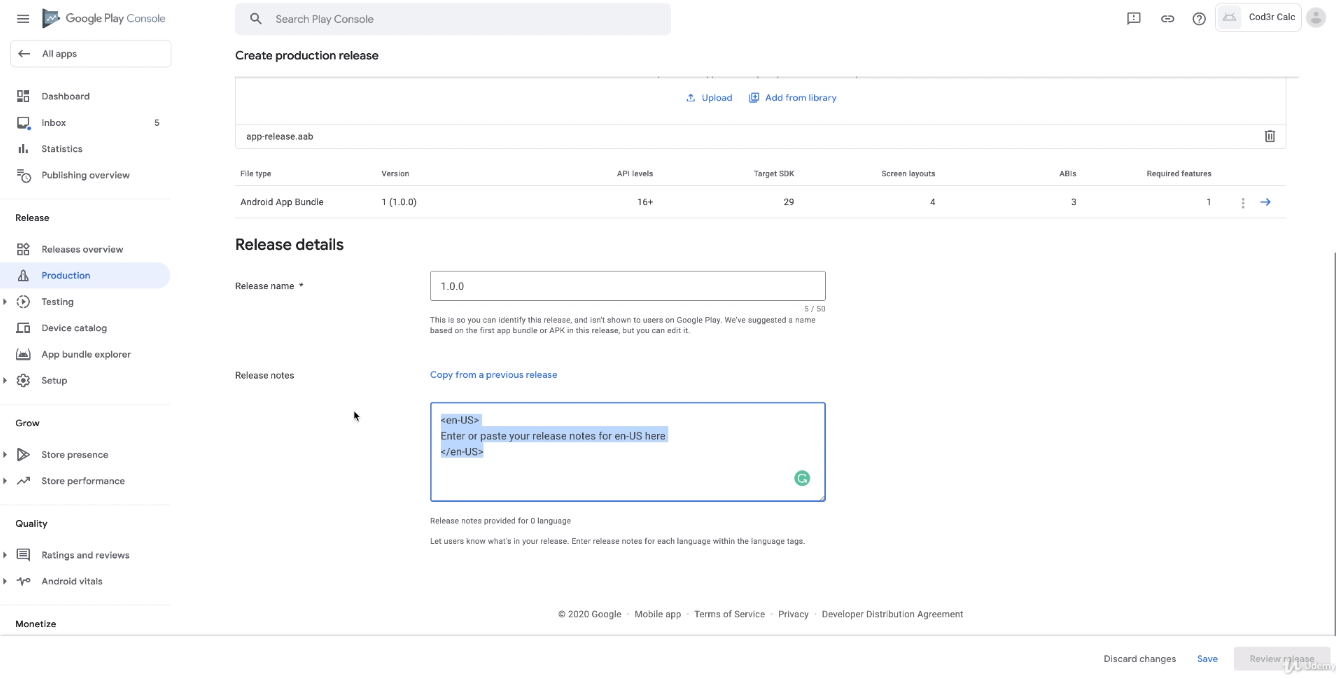


Agora clique em upload



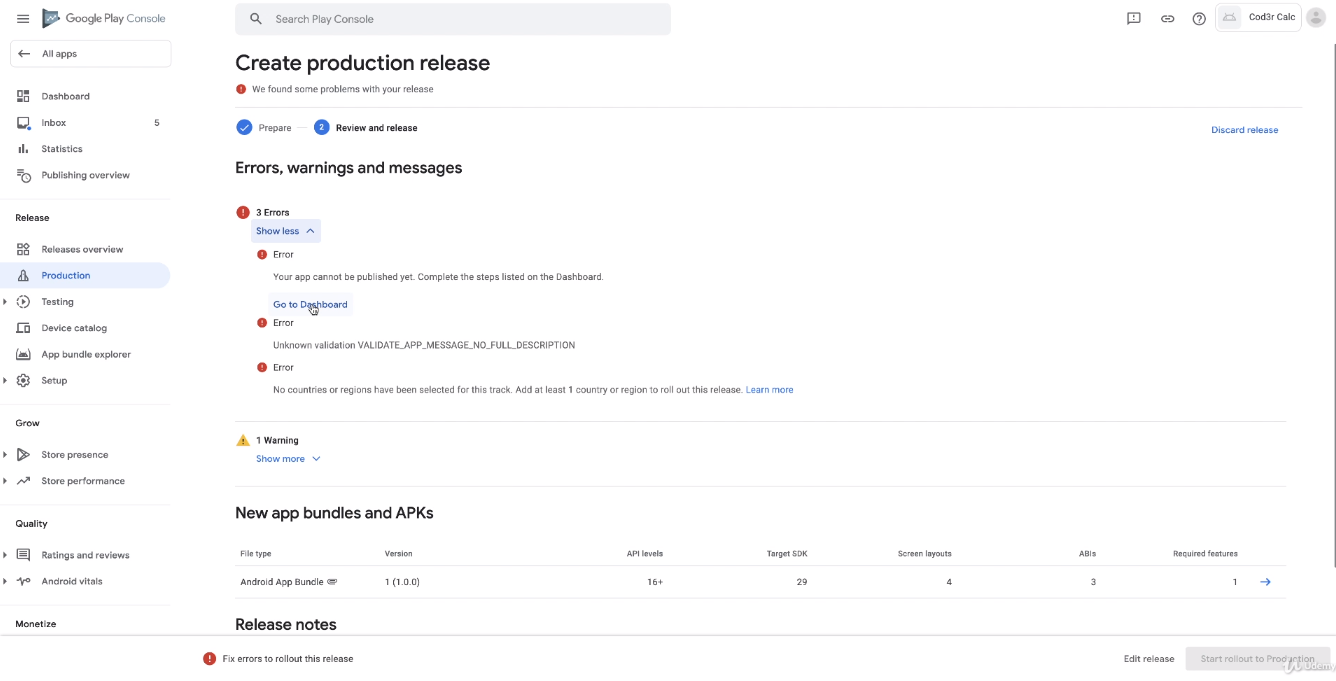
E selecione o arquivo bundle “app-release.aab” gerado anteriormente.

E também adicione o nome da release e notas de alteração caso deseje



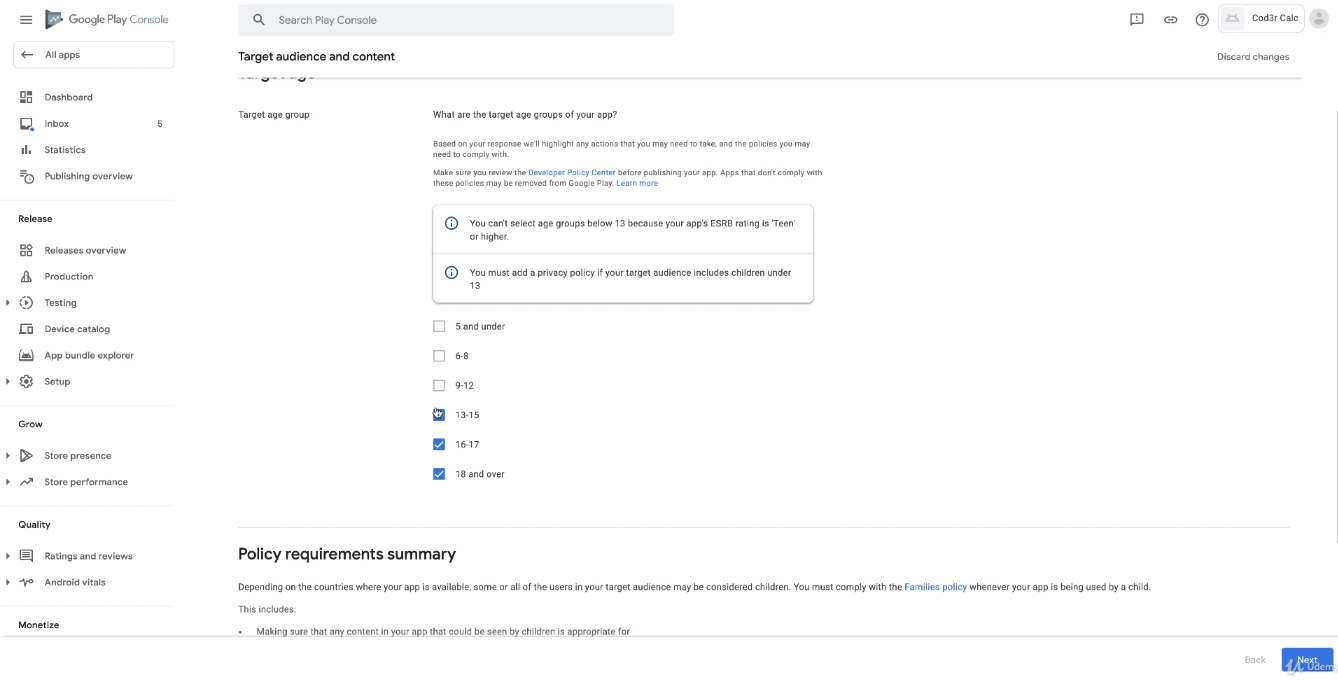
Em seguida clique em “Save”

Agora será exibida a release pronta para ser publicada, e caso possua algum erro para corrigir, também será exibida.

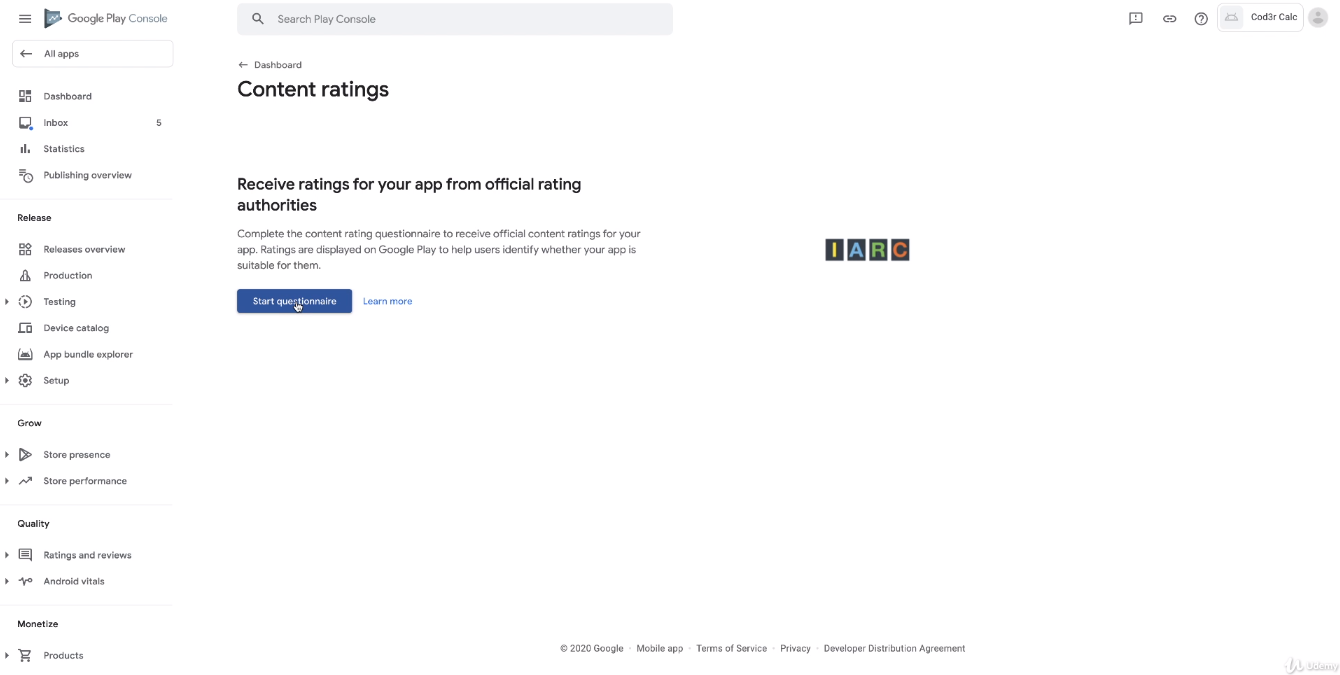


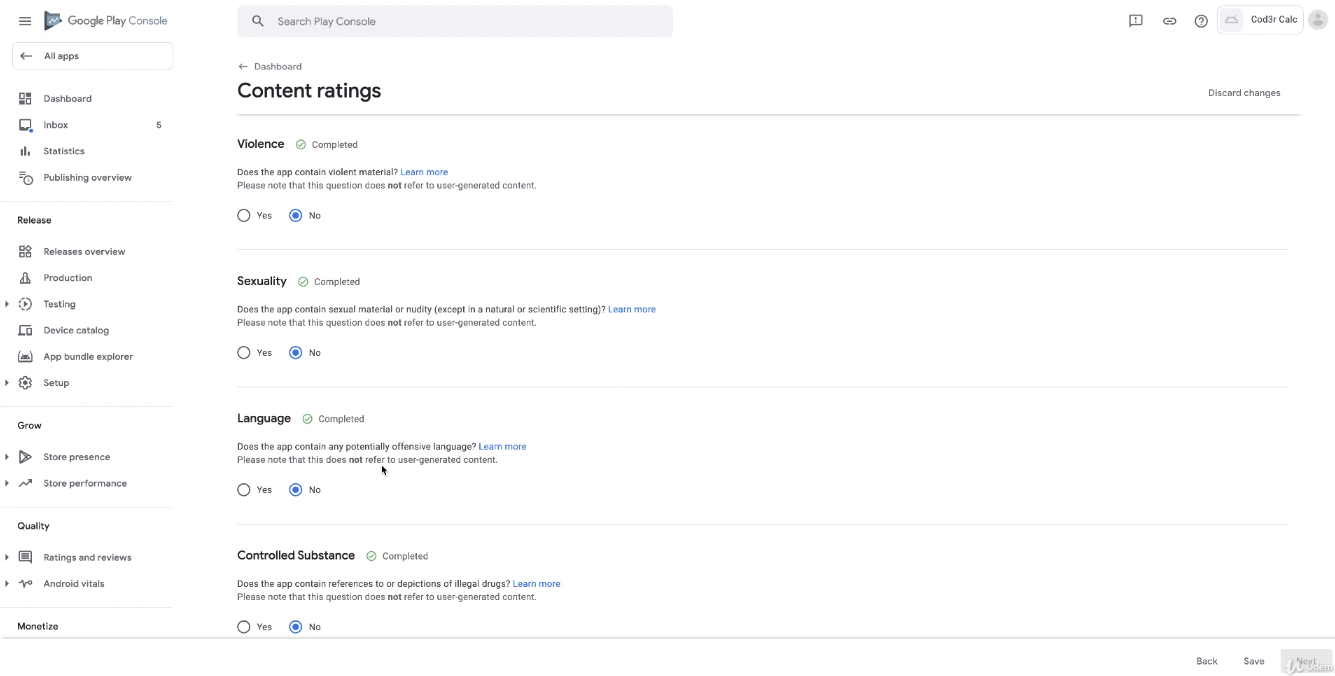
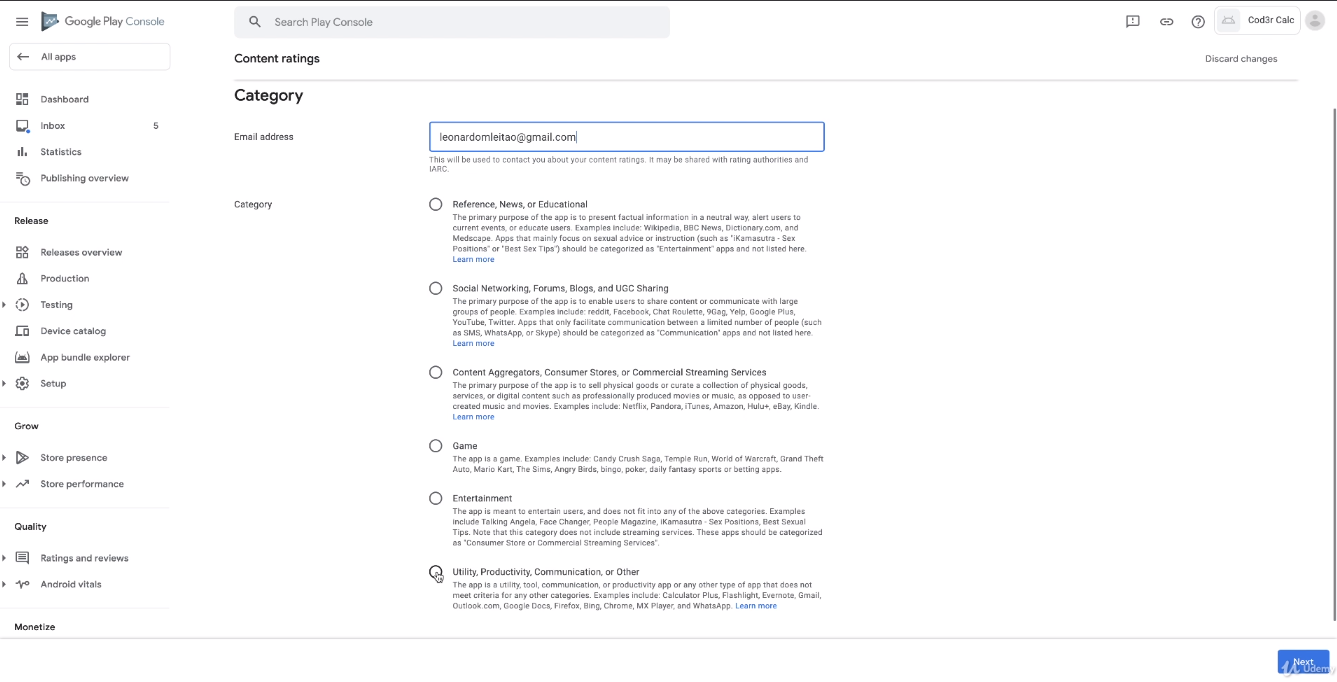
Pode ser visto, que primeiramente é necessário terminar os passos iniciais.

Definindo a audiência da aplicação por idade.

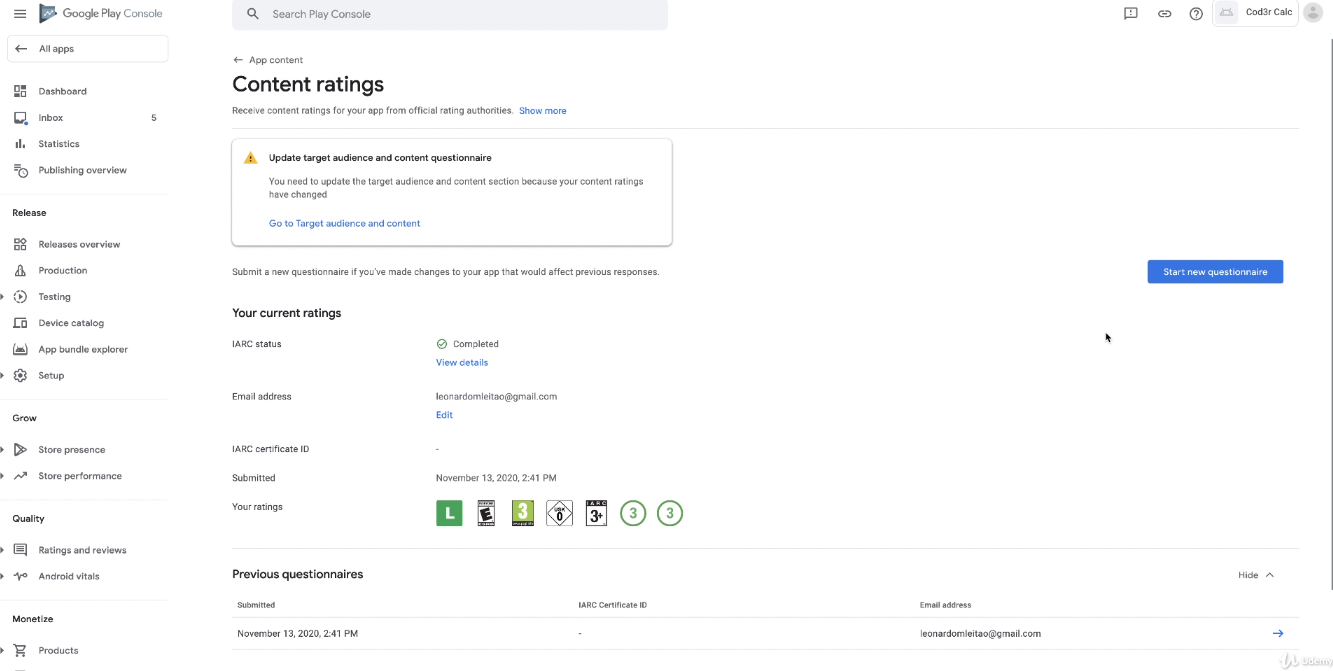


Configurando Content rating

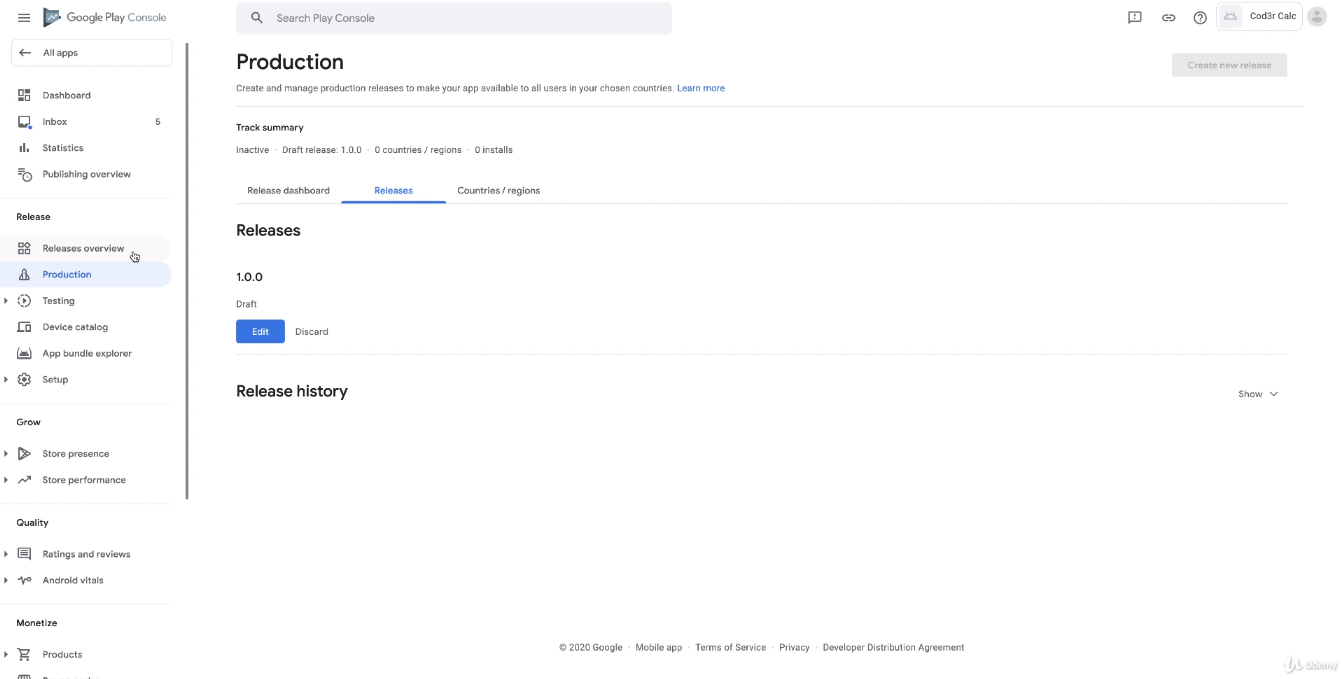




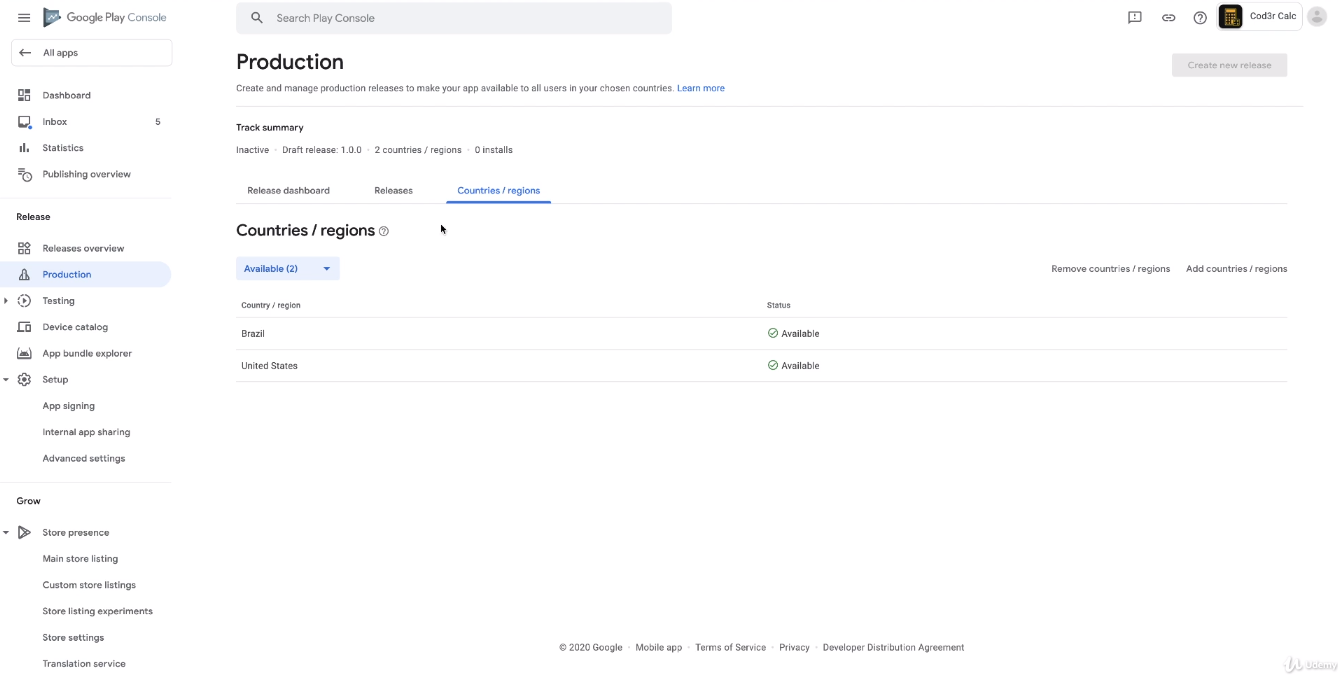
E por final clique em submit



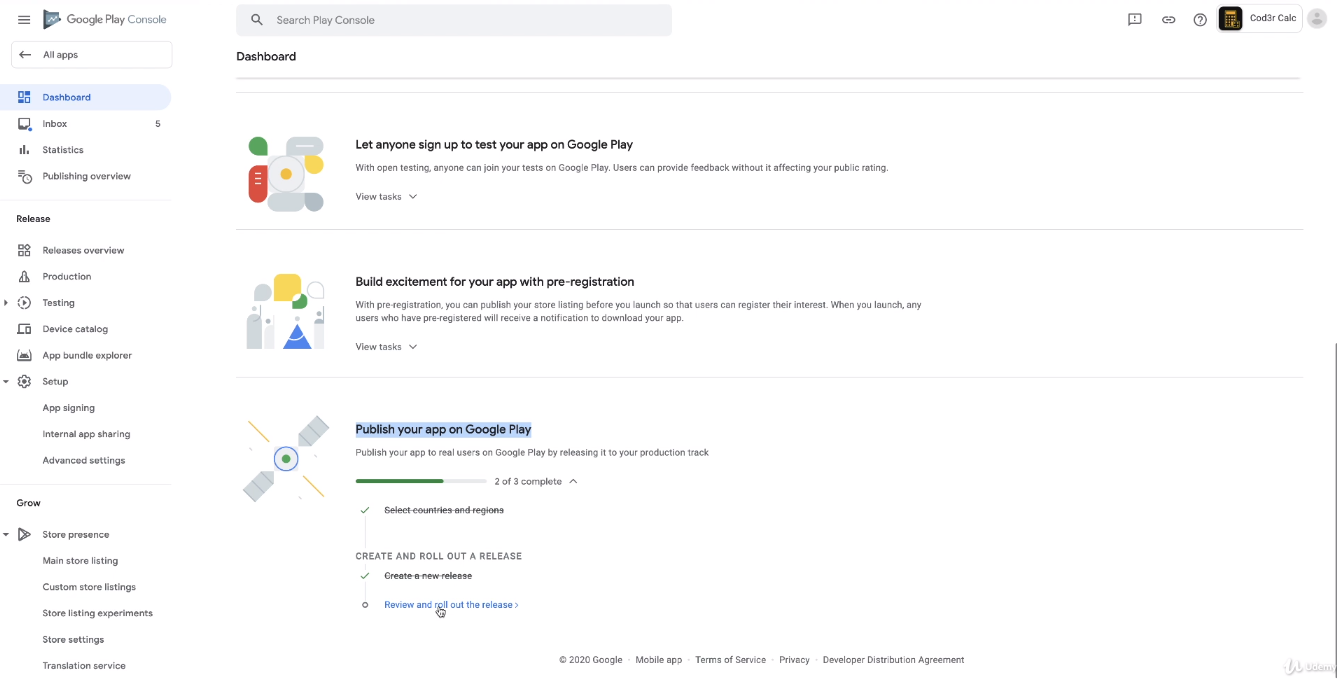
Em seguida é basicamente preencher os dados restantes solicitados, adicionar as imagens etc. e publicar a release criada.



Também é preciso selecionar as regiões

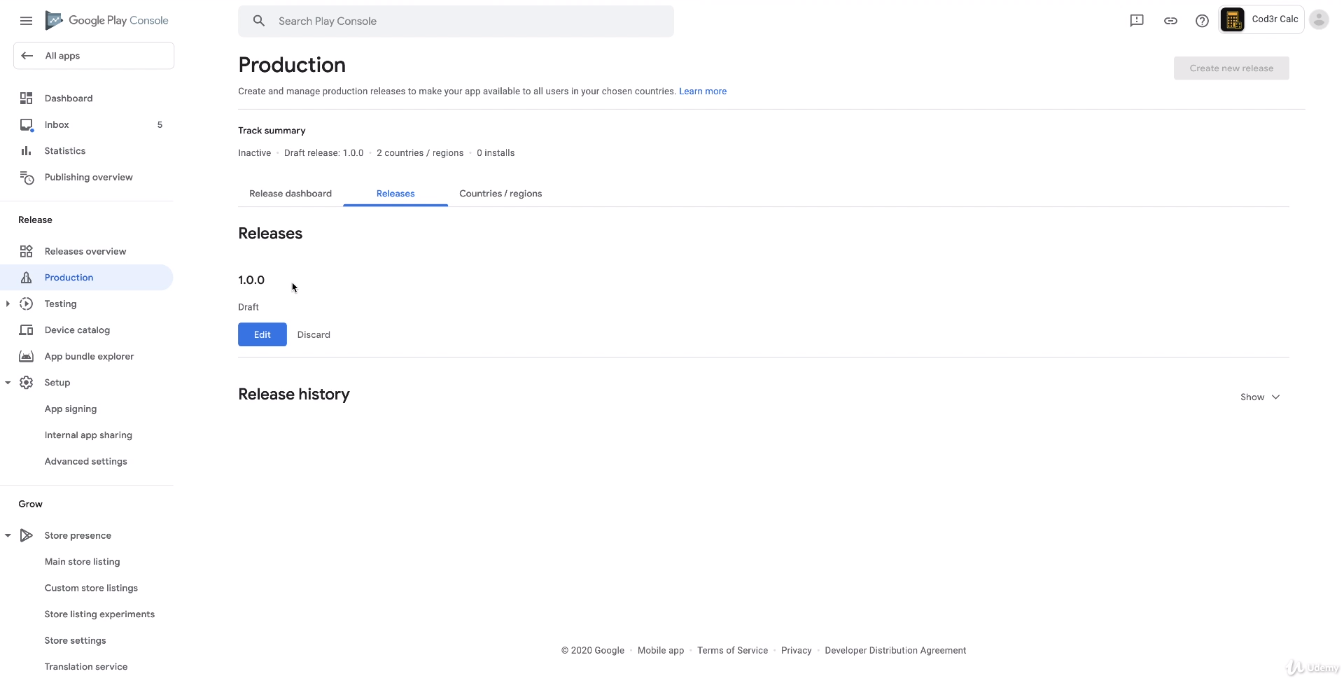


Então na dashboard vá em Publish your app on Google Play

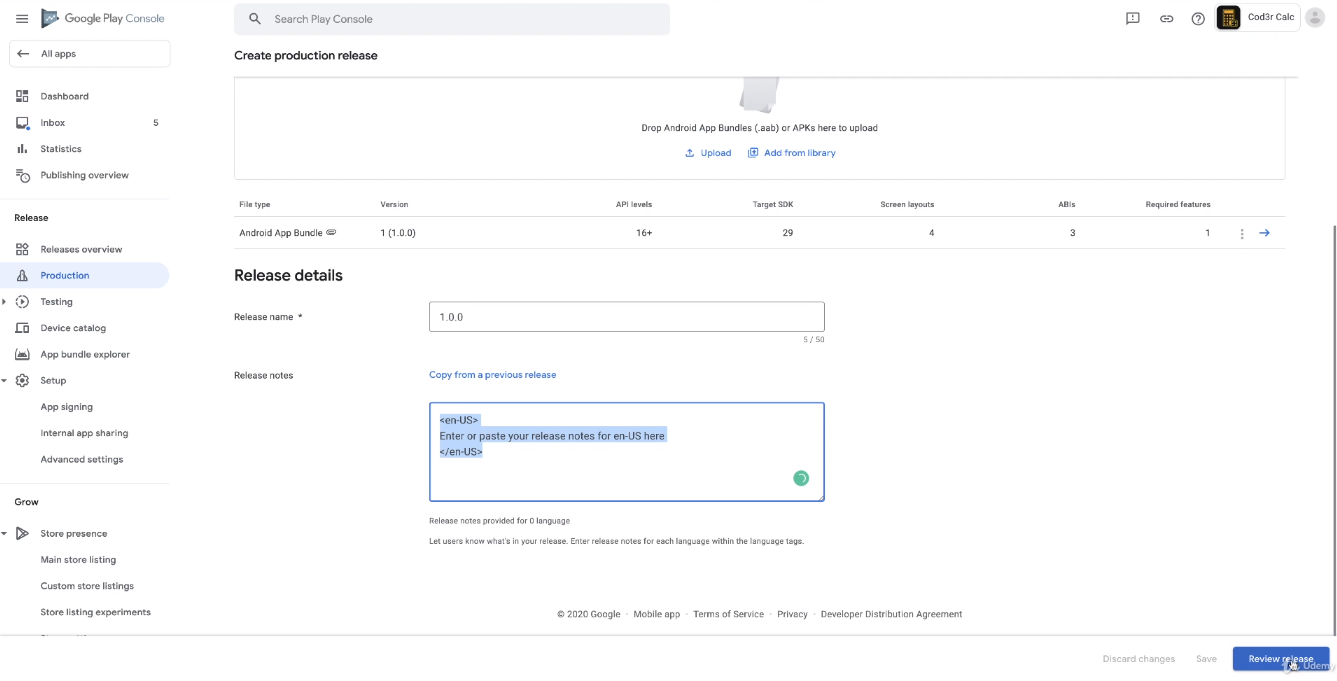


E clique em Review and roll out the release

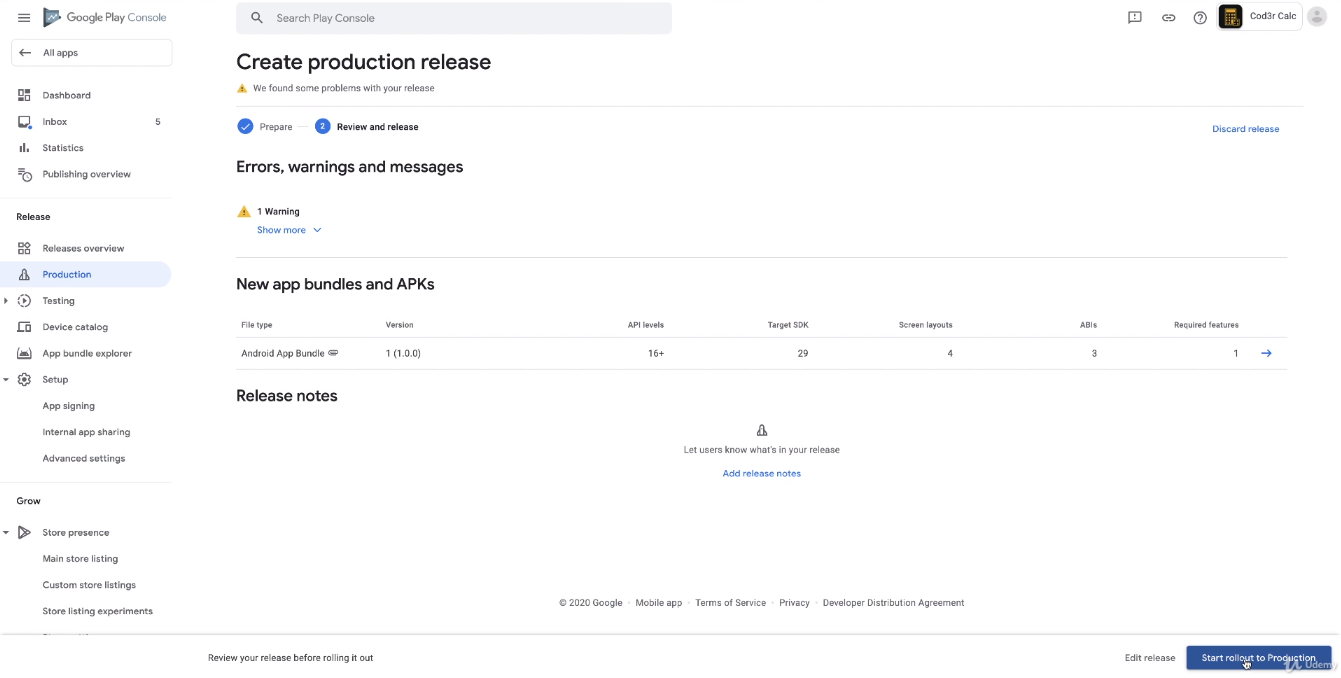
Clique em Edit

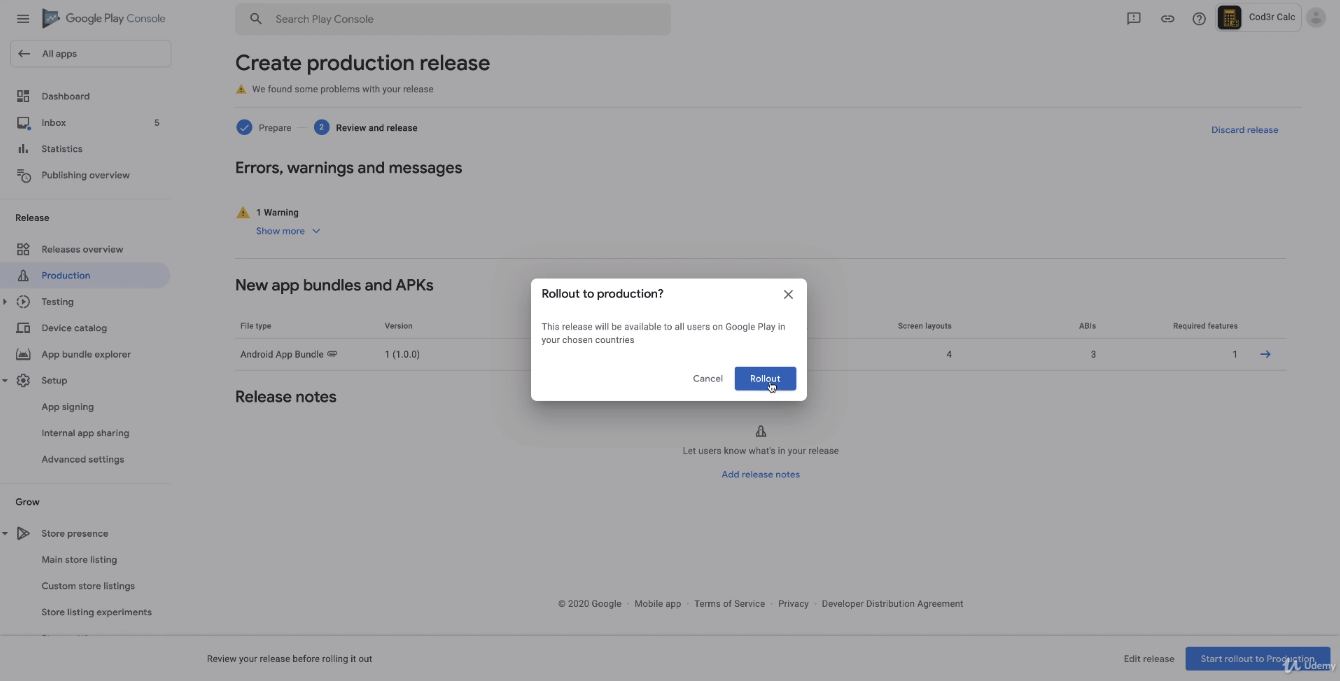


Clique em Review Release



Então finalmente clique em start rollout to production





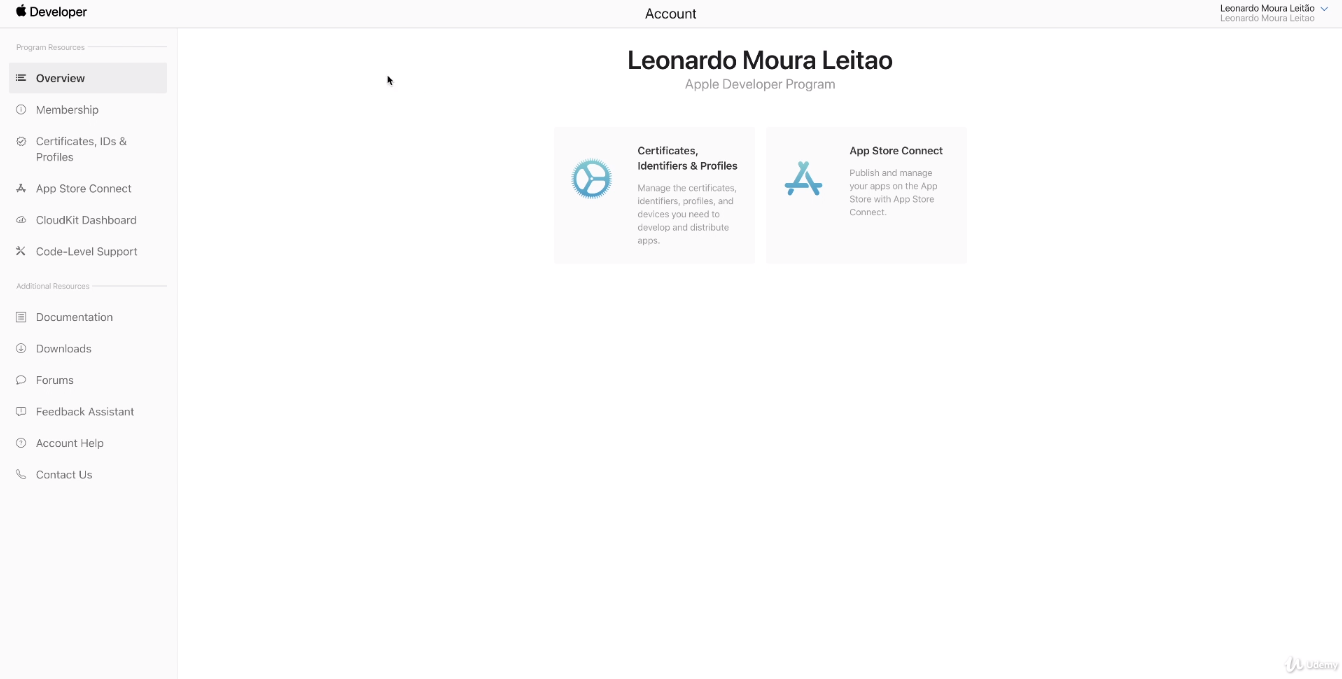
Pronto, agora sua APP entrara em revisão, e logo será publicada

Também é possível a publicação de versões de teste.

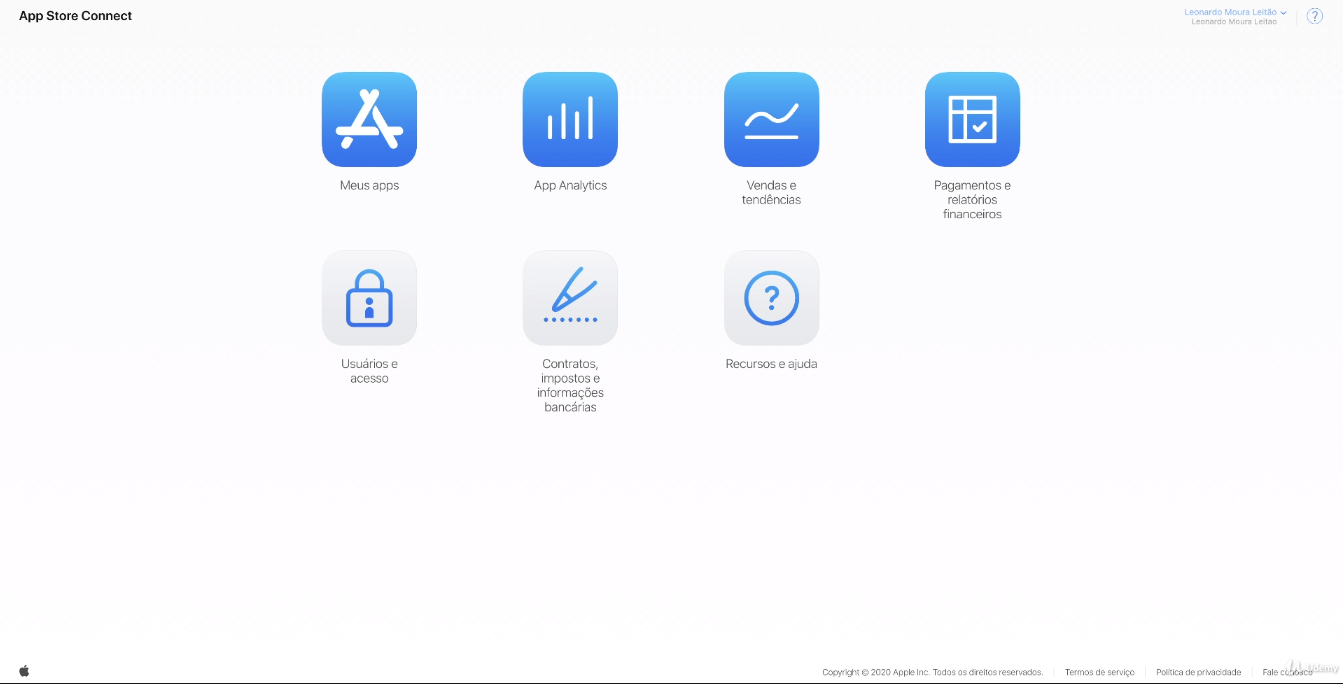
**Publicação na Apple Store**

Sera preciso acessar dois sites específicos.

<https://developer.apple.com/account/> - acessar sua conta.

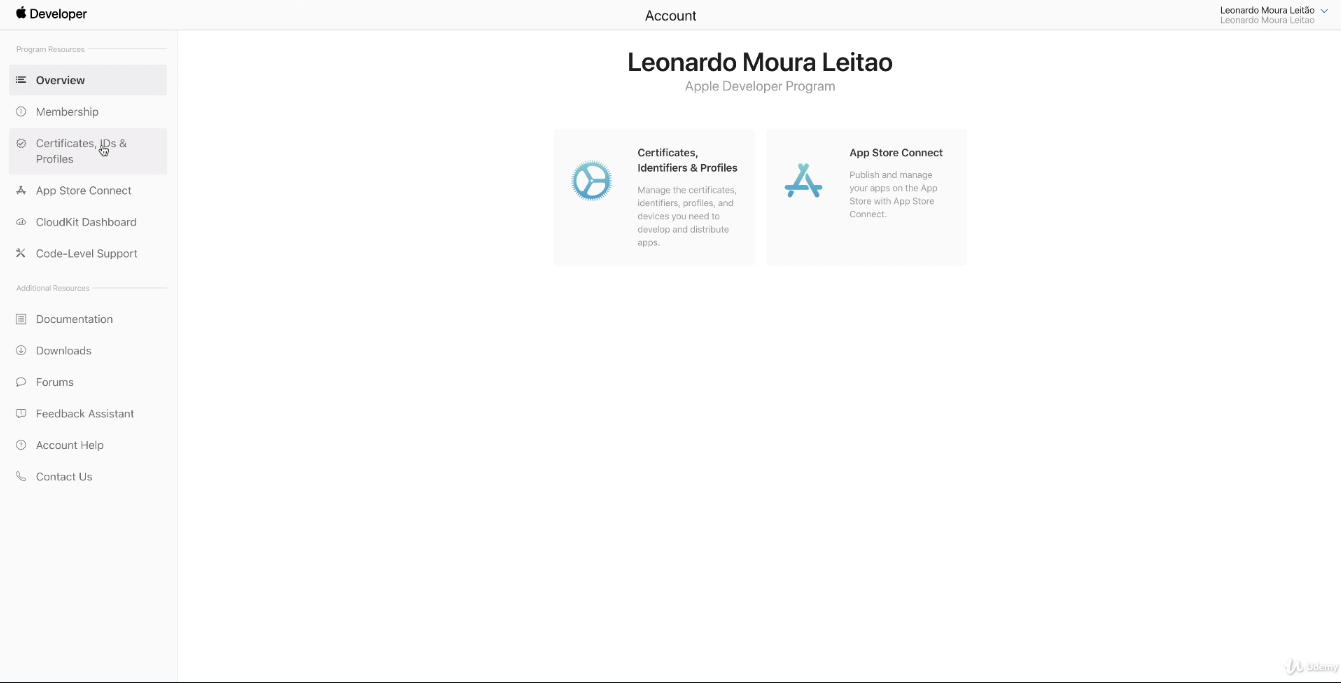


<https://appstoreconnect.apple.com/login> - Acesse com sua conta

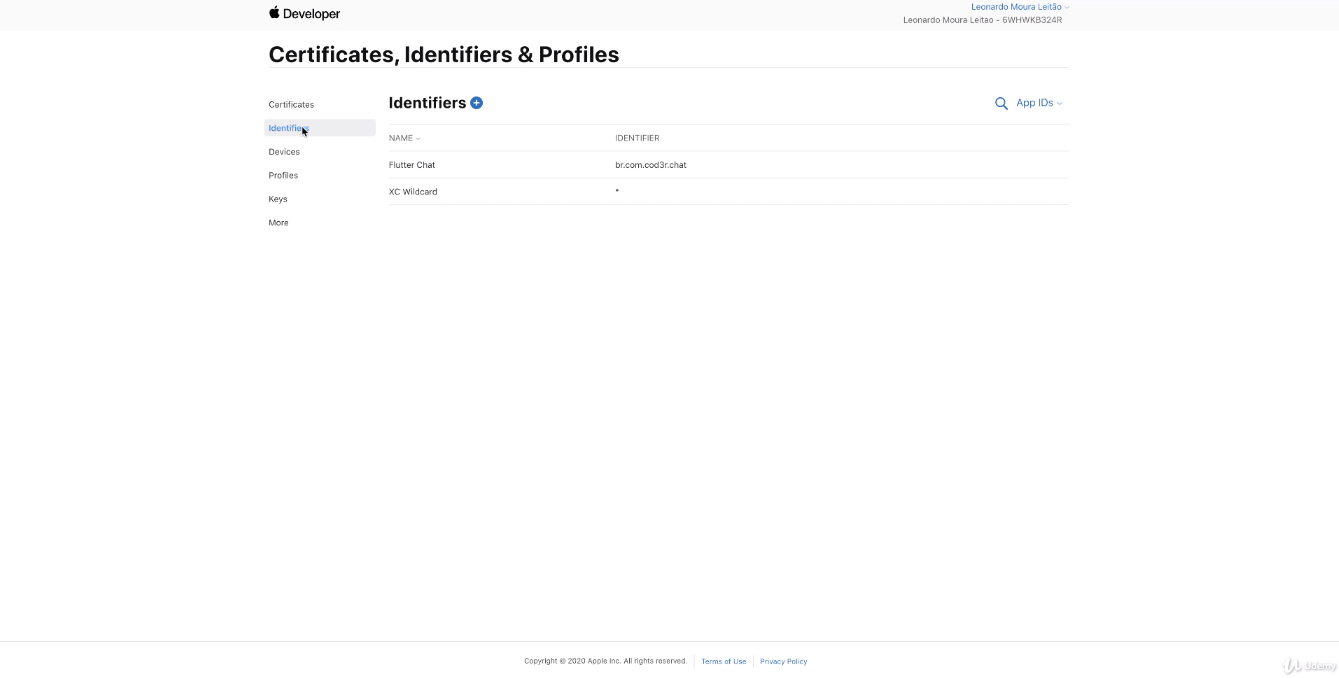


**Registrando o Bundle ID**

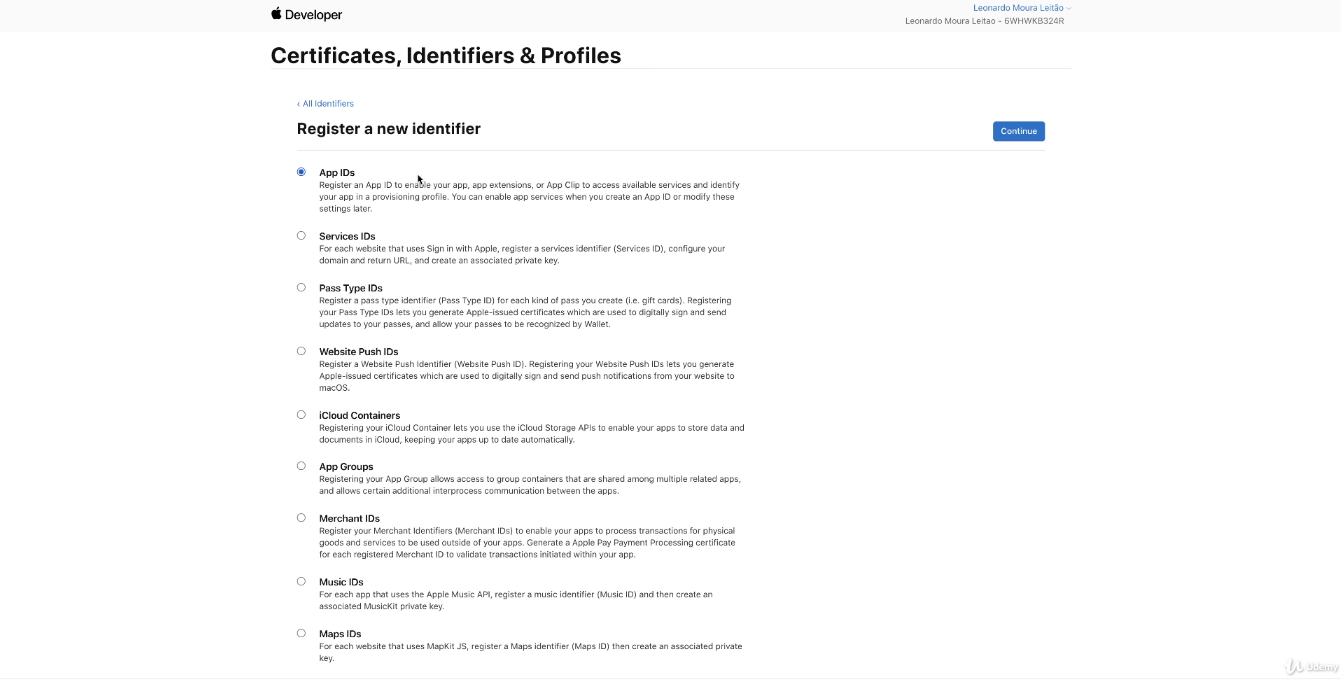
Selecione “Certificates, IDs & Profiles”



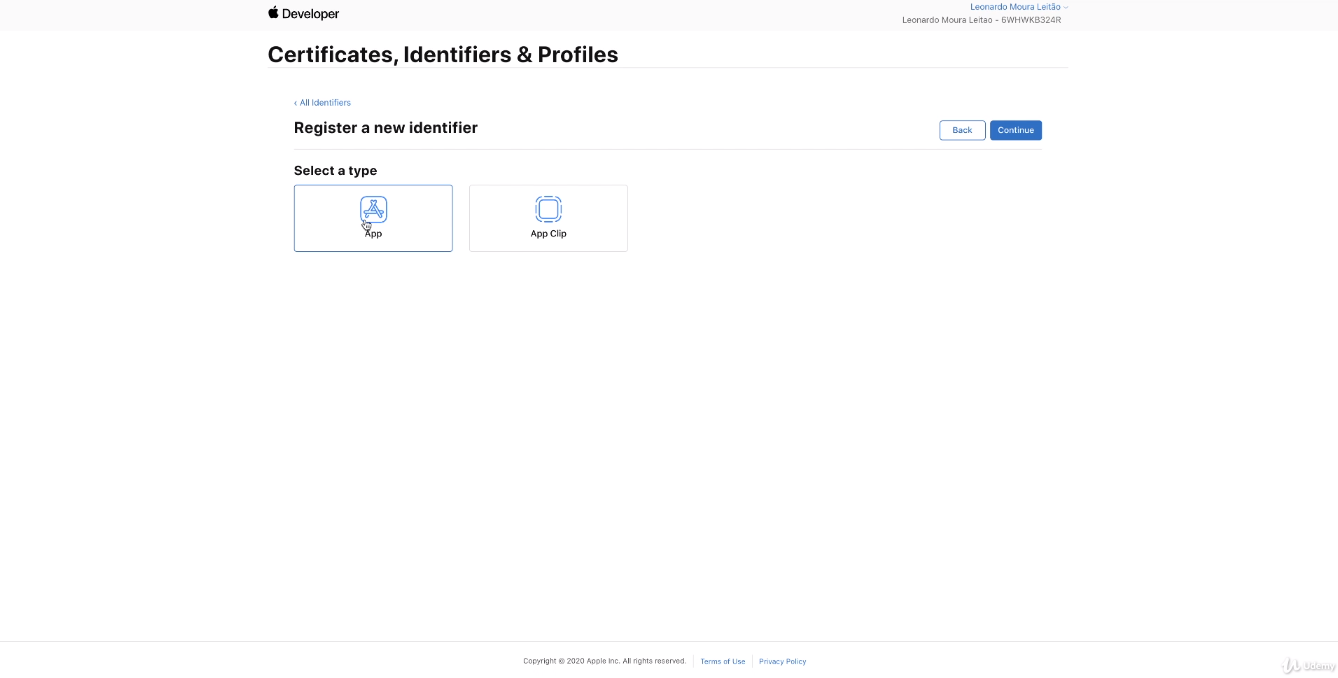
Selecione Identifiers



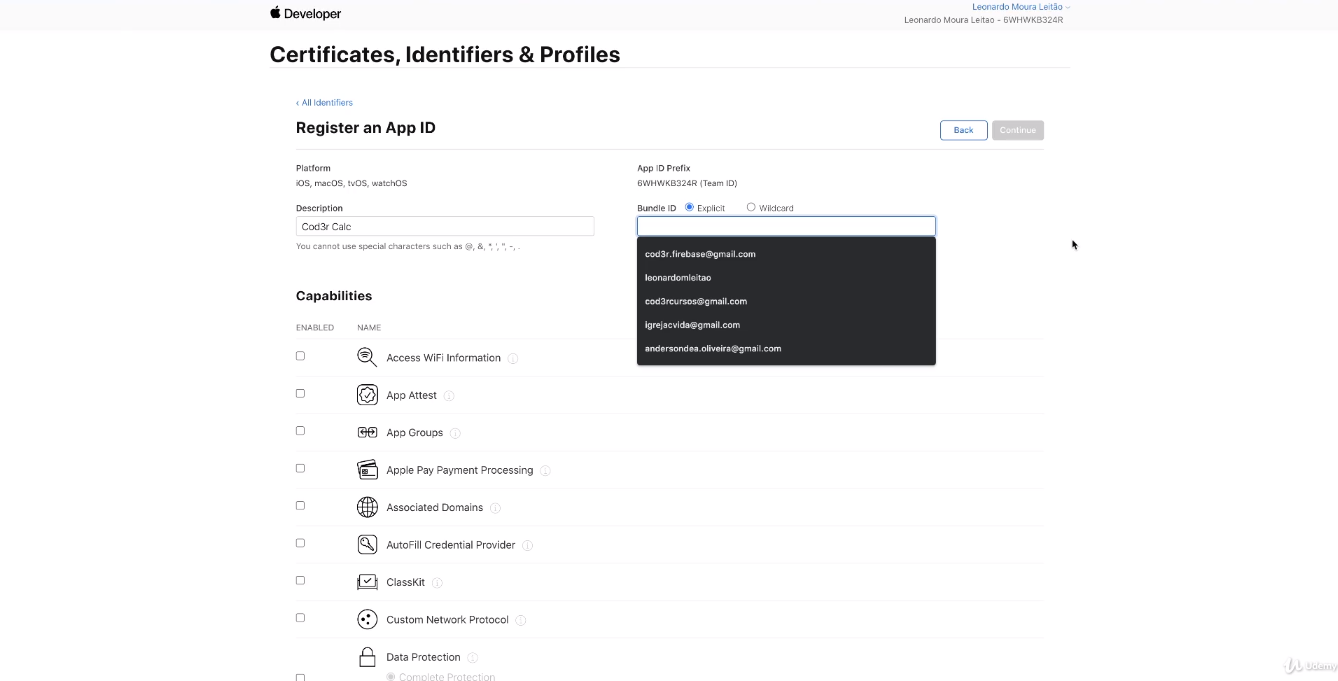
Selecione App IDs e clique em continue



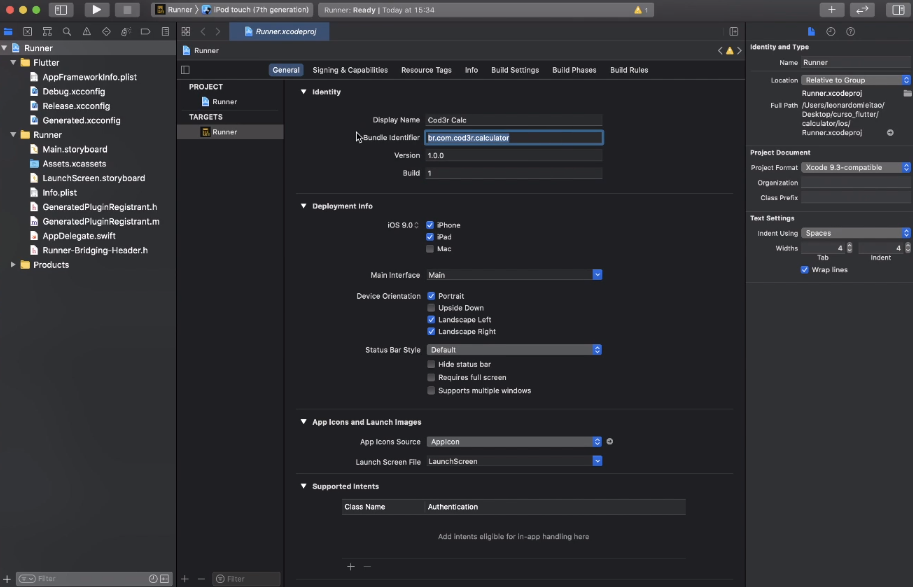
Selecione App e continue



Preencha a descrição e o bundle ID

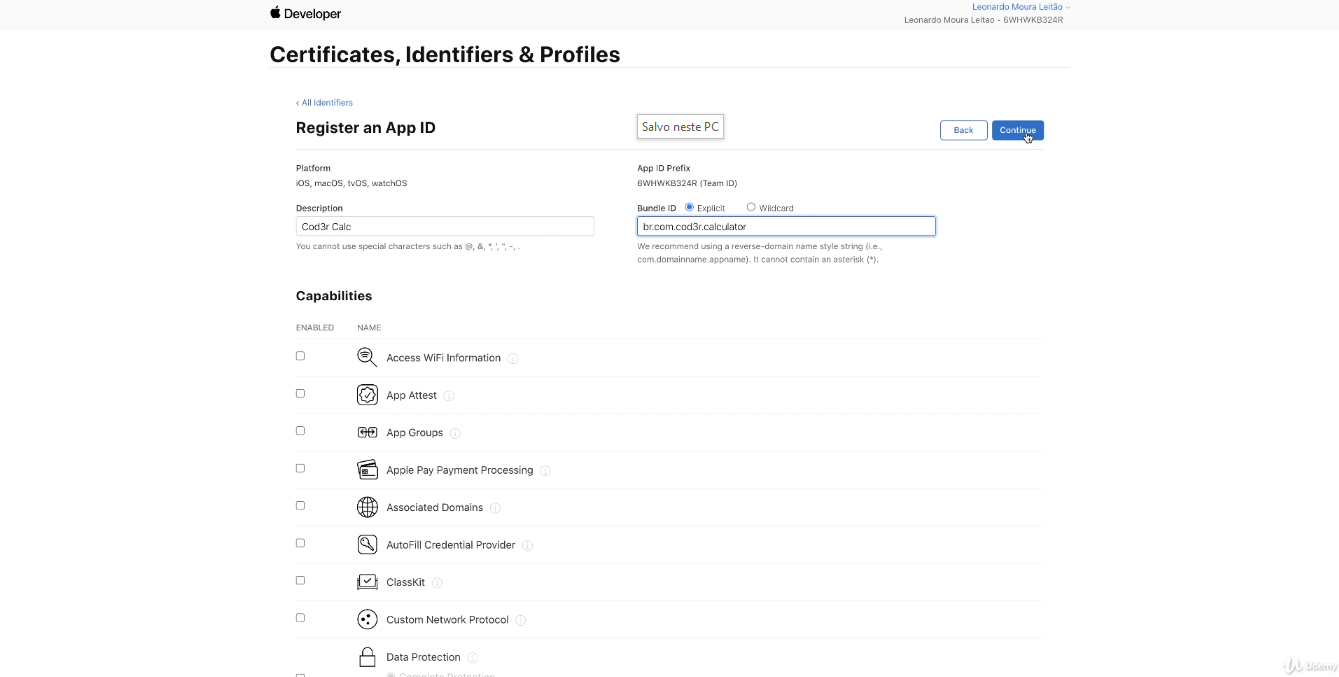


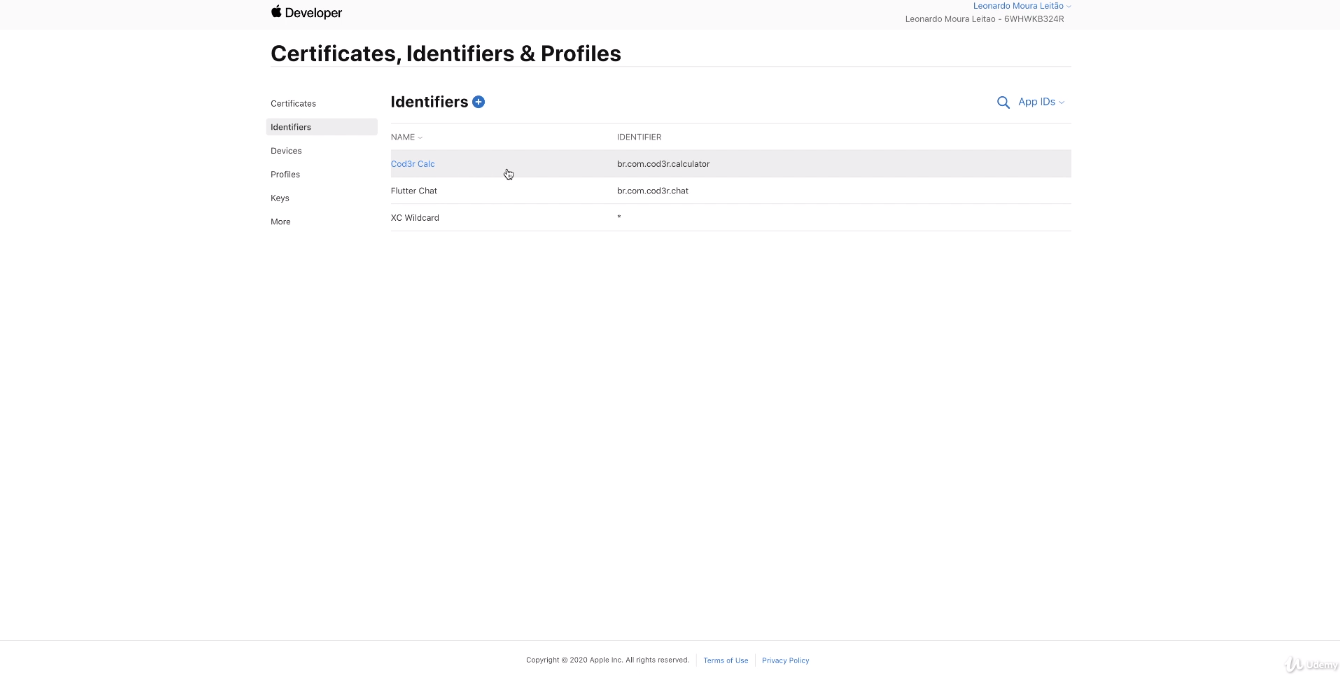
O bundle ID é encontrado no projeto, abrindo com o XCode



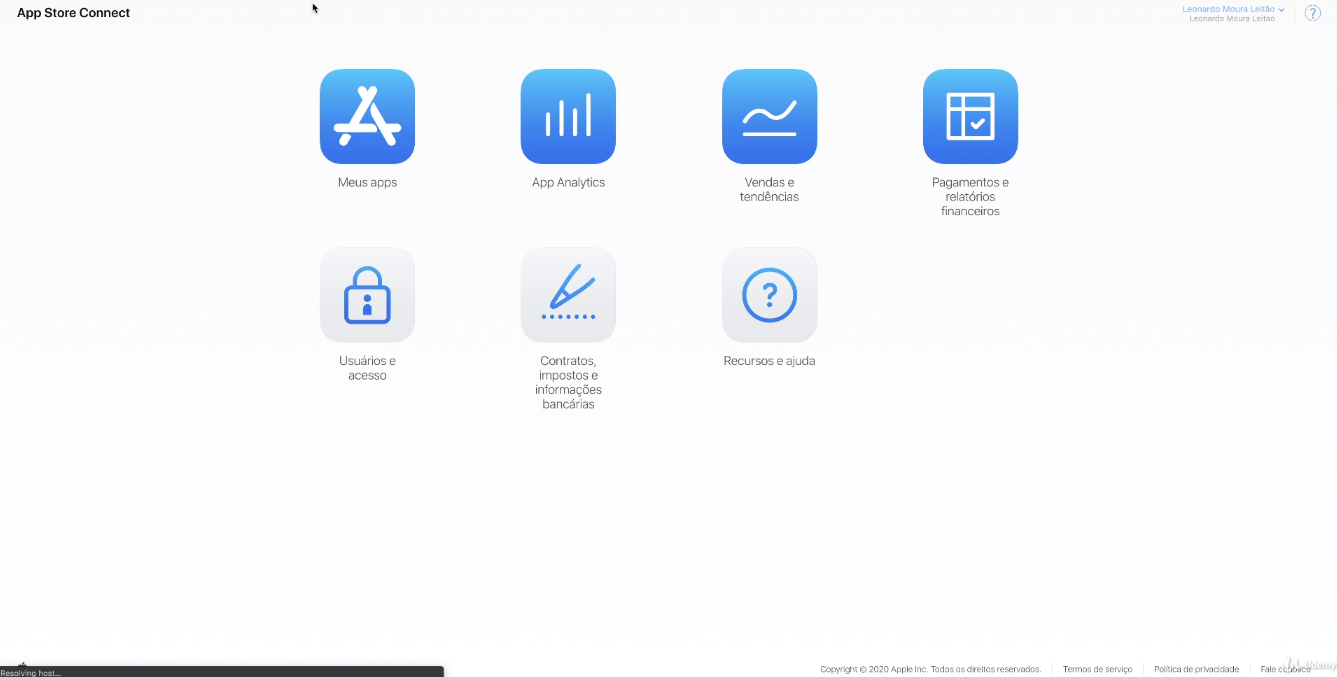
Selecione a Capabilities necessárias (Por exemplo: A aplicação utiliza push notification)

Em seguida clique em continue e Register

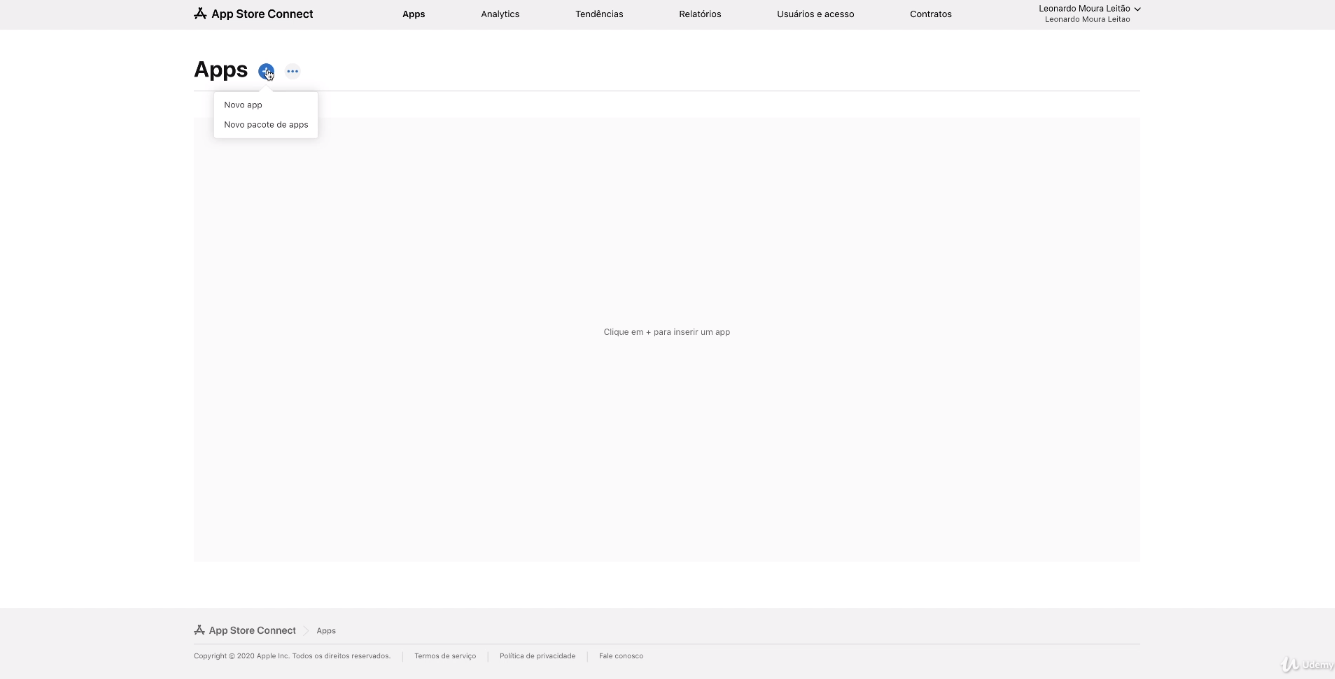




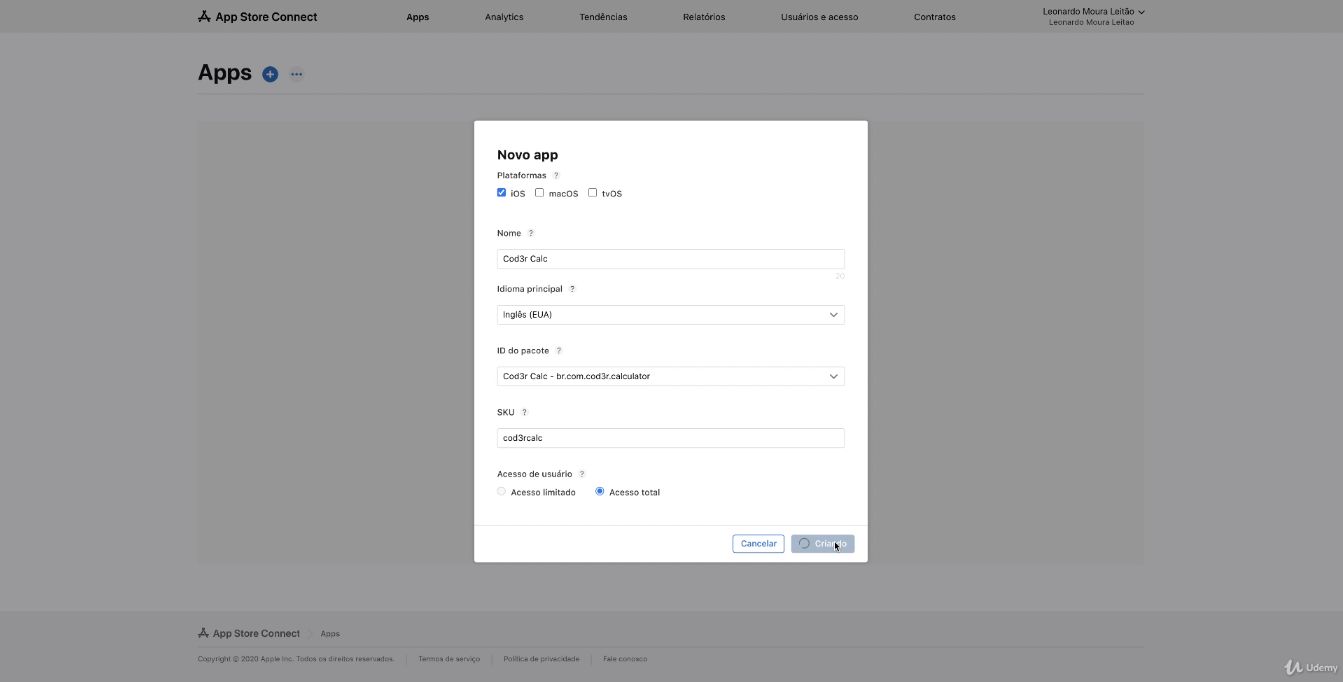
Agora em Apple Store Conect clique em Meus APPs



Clique em novo app



Preencha os dados e clique em criar



É preciso selecionar o Bundle ID gerado anteriormente, e em SKU definir uma chave única.

