



Introdução

ANDROID & IOS (MANAGED)

Windows

Linux

macOS

ANDROID (BARE & RN CLI)

Windows

Linux

macOS

IOS (BARE & RN CLI)

macOS

DISPOSITIVOS VIRTUAIS

Emulador Android

Simulator iOS

Android Emulator

ON THIS PAGE

Android Studio

Virtual Device Manager

Iniciando o Emulador

Executando app no Emulador

Expo Bare Workflow

React Native CLI

Android

Configurando o ambiente
de desenvolvimento Android

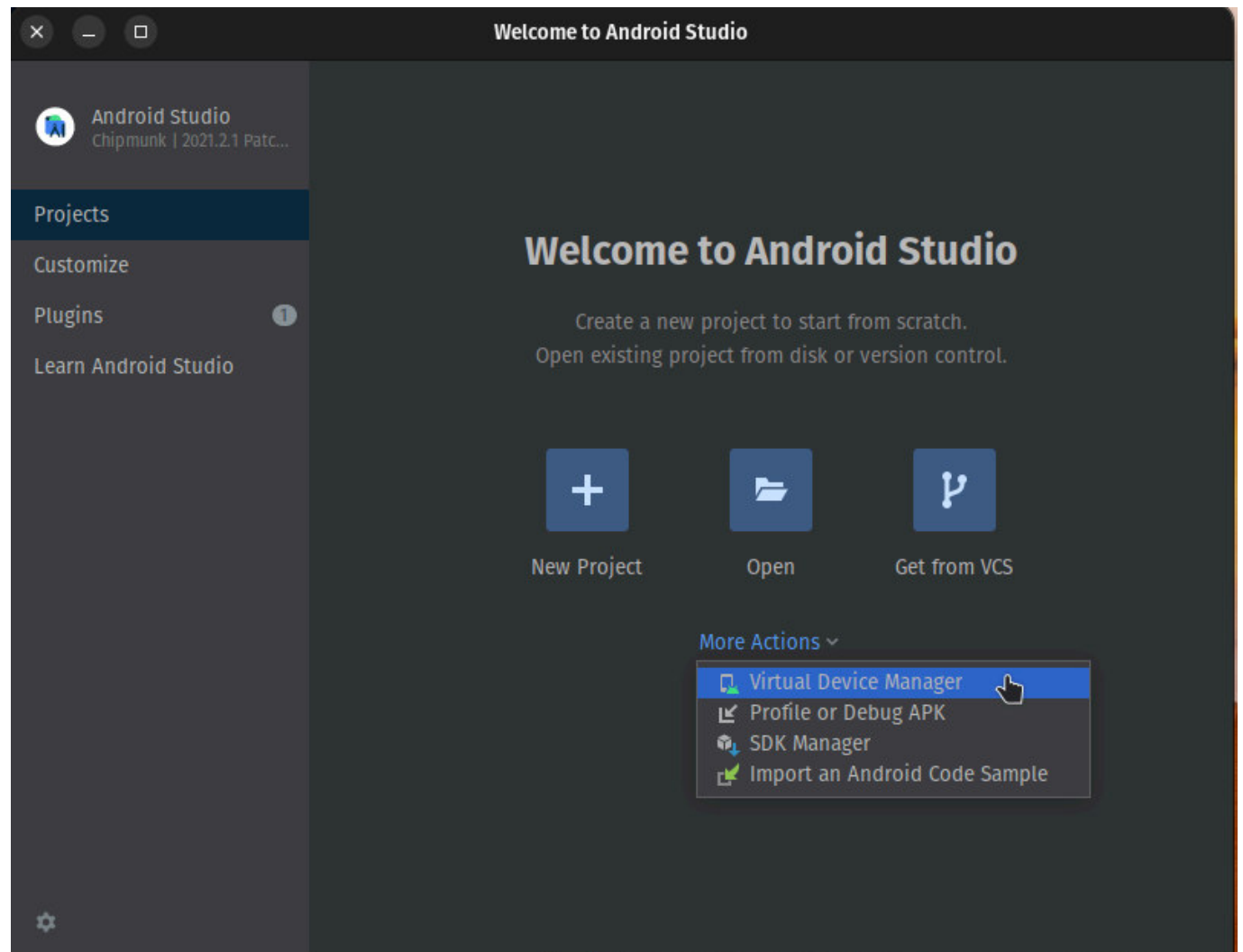
Caso você possua um dispositivo físico Android e seu computador tenha configurações modestas (ex.: ↓ i3, ↓ 4gb RAM), é recomendado executar a aplicação diretamente no dispositivo físico. Para configurar seu aparelho, clique **aqui**

Android Studio

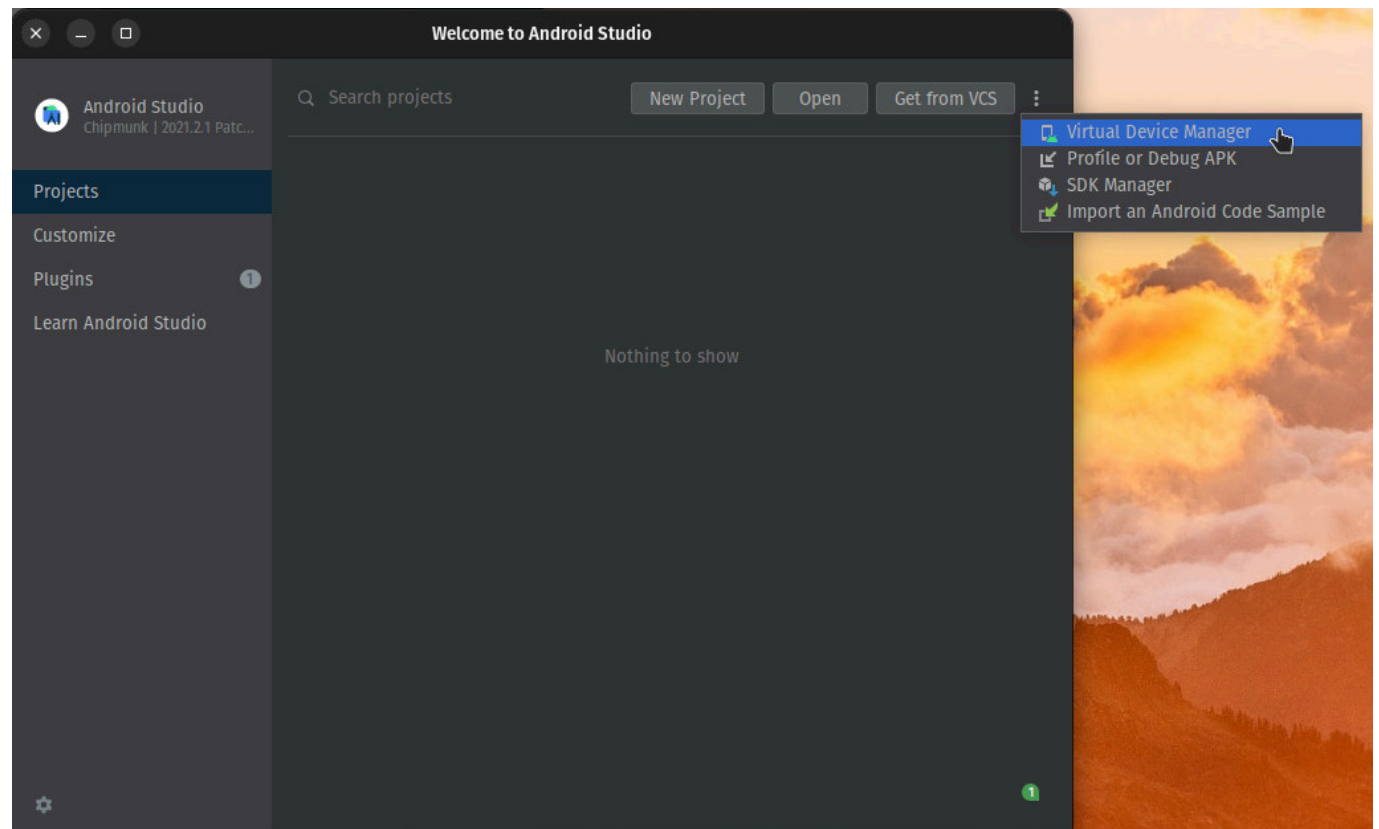
Para criar o seu emulador android, é preciso ter o Android Studio instalado. Abra o programa e, dependendo se você já abriu ou não um projeto android no Android Studio antes, a interface será diferente.

Se você não possui o Android Studio instalado, pode fazê-lo seguindo nossas docs de acordo com o seu sistema operacional (**Windows**, **Linux** e **macOS**).

Se nunca abriu um projeto android nele antes, basta clicar em **More Actions** no centro da tela e escolher a opção **Virtual Device Manager**.



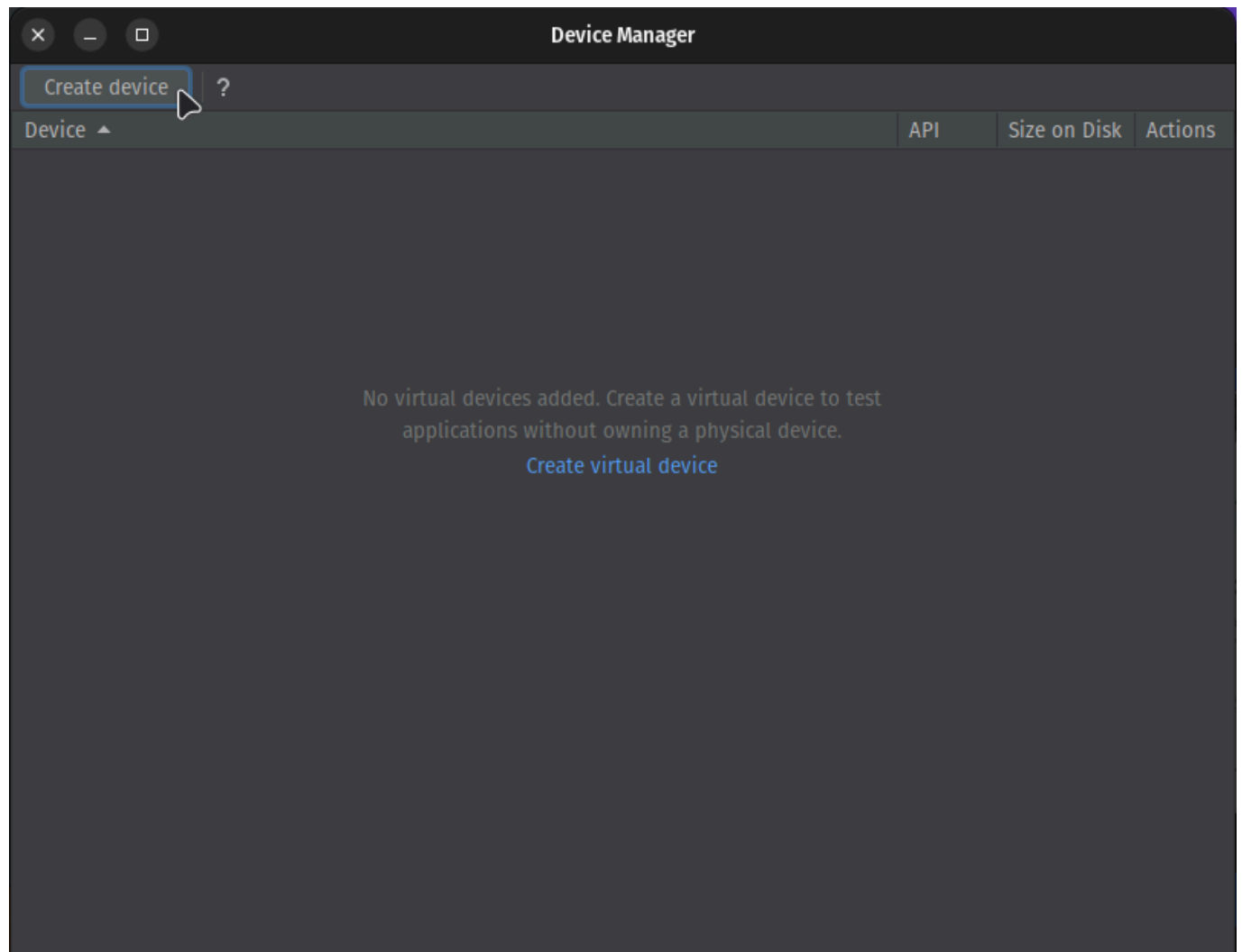
Se já abriu um projeto android antes, basta clicar nos três pontos no canto superior direito e escolher a opção **Virtual Device Manager**



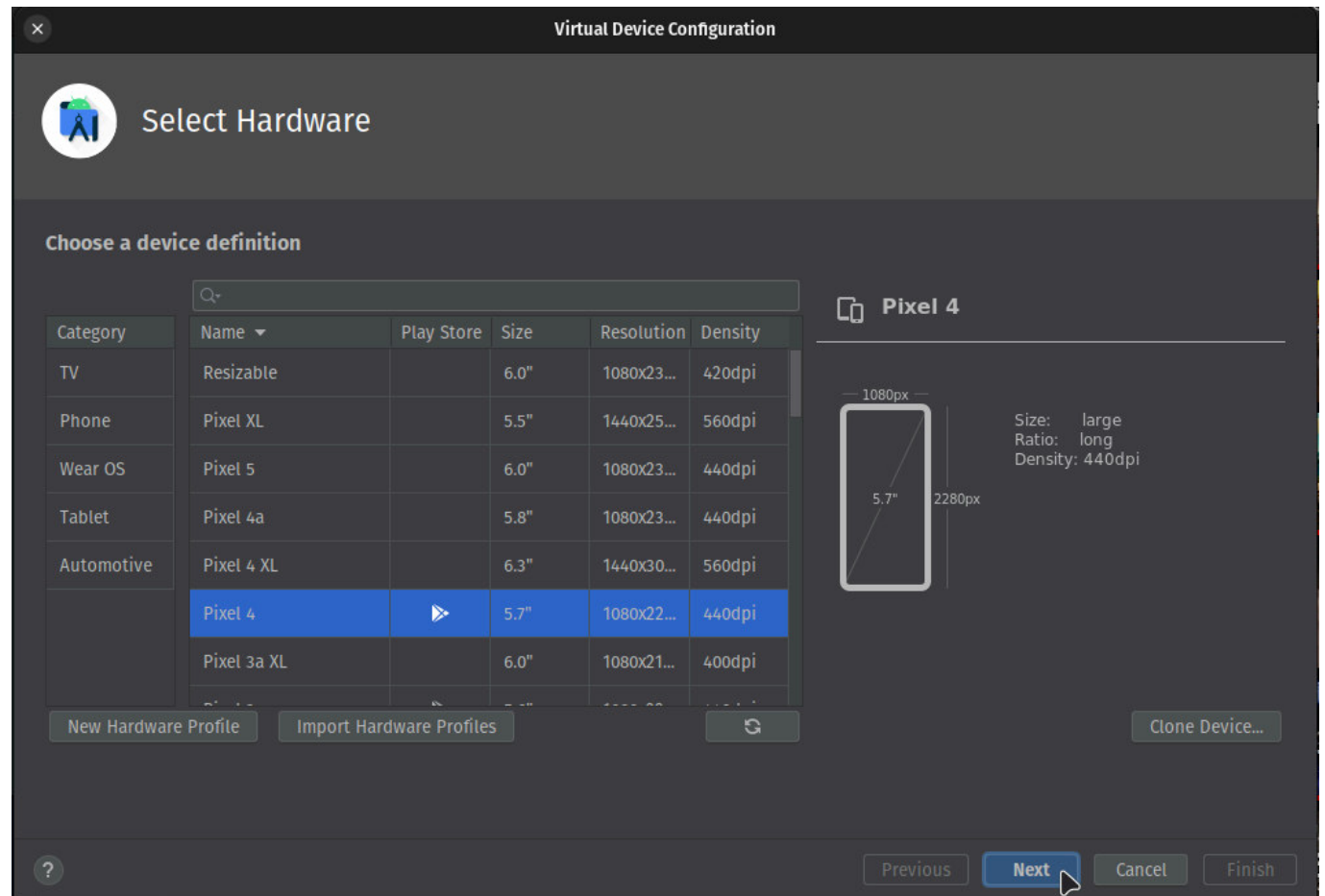
Virtual Device Manager

Na sequência, será apresentada uma tela com os emuladores instalados. Se você marcou a opção **Android Virtual Device** na etapa de configuração do Android Studio, deve aparecer um emulador já configurado e pronto para uso. Caso queira utilizar esse mesmo, pule para a próxima seção **clikando nesse link**.

Porém, caso queira criar seu próprio emulador, selecione a opção **Create Device** no canto superior esquerdo.



Nessa etapa, será perguntado qual Hardware você quer selecionar. Como exemplo, escolhi na aba **Phone** o **Pixel 4**. Após escolher, clique em **Next**.



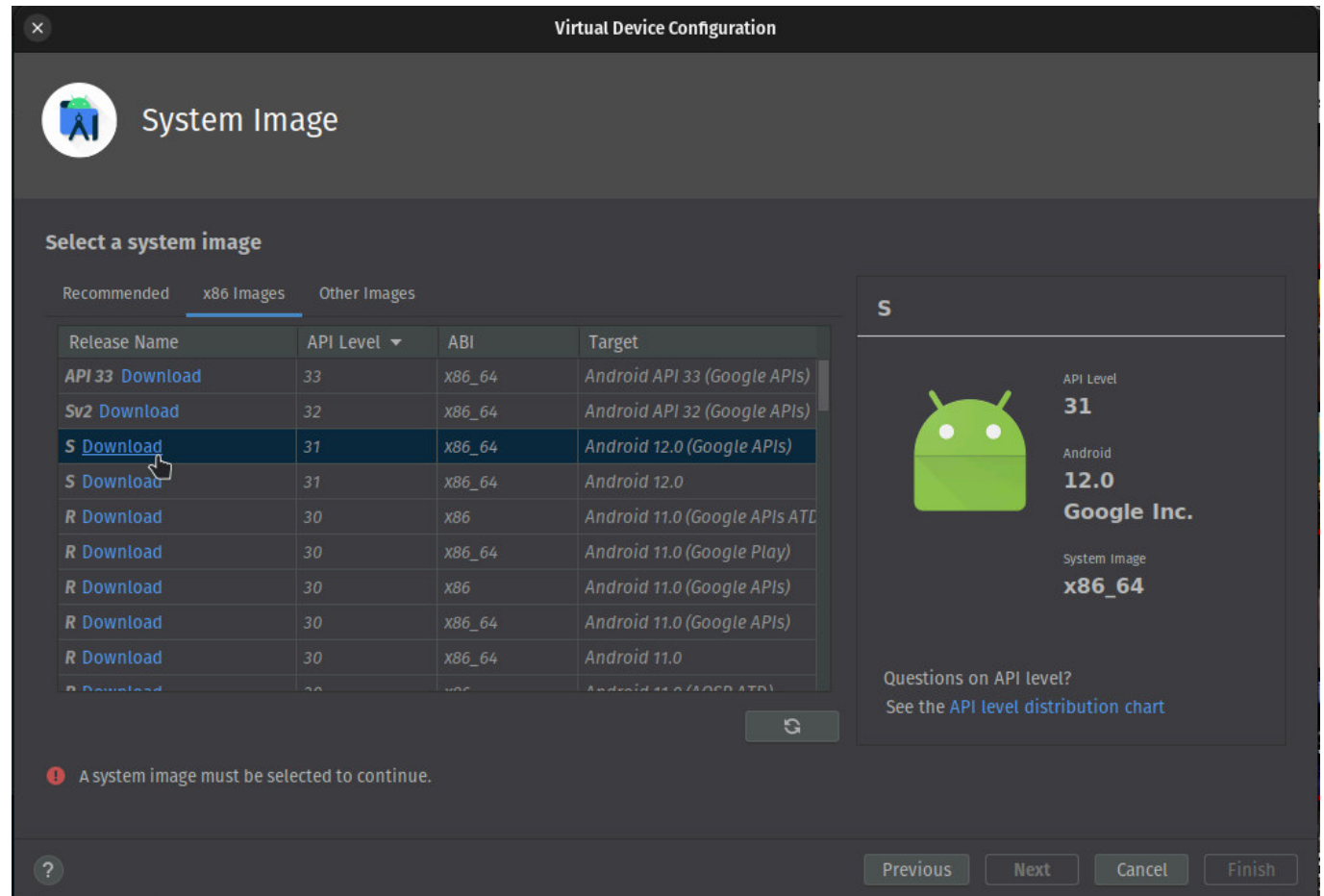
Em seguida, você deverá escolher a imagem do sistema (API) do emulador a partir de uma das três listas: **Recommended**, **x86 Images** e **Other Images**. Se essa é sua primeira vez, provavelmente nenhuma imagem estará baixada.

Aqui, a escolha da imagem do sistema vai depender da arquitetura do seu processador.

Sugerimos que escolham exatamente a imagem que recomendamos abaixo pois é a que


testamos por aqui. Outras imagens podem não funcionar corretamente.

- Caso o seu processador seja Intel ou AMD, escolha a aba **x86 Images** e clique no link de **Download** da imagem **S**, API Level 31, ABI **x86_x64** e Target **Android 12.0 (Google APIs)**.



- Caso o seu processador seja Apple ARM (ex.: M1), escolha a aba **Recommended** e clique no link de **Download** da imagem **S**, API Level 31, ABI **arm64-v8a** e Target **Android 12.0 (Google Play)**.

Virtual Device Configuration


System Image

Select a system image

RecommendedARM ImagesOther Images

Release Name	API Level	ABI	Target
<i>TiramisuPrivacySandbox</i>	<i>TiramisuPrivacy</i>	<i>arm64-v8a</i>	<i>Android API TiramisuPriv</i>
<i>API 33</i> Download	33	<i>arm64-v8a</i>	<i>Android API 33 (Google</i>
<i>API 32</i> Download	32	<i>arm64-v8a</i>	<i>Android API 32 (Google</i>
S Download	31	<i>arm64-v8a</i>	<i>Android 12.0 (Google Pla</i>
<i>R</i> Download	30	<i>arm64-v8a</i>	<i>Android 11.0 (Google Pla</i>
<i>Q</i> Download	29	<i>arm64-v8a</i>	<i>Android 10.0 (Google Pl</i>
<i>Pie</i> Download	28	<i>arm64-v8a</i>	<i>Android 9.0 (Google AR</i>

S



API Level

31

Android

12.0

Google Inc.

System image

arm64-v8a

We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?
[See the API level distribution chart](#)

?

A system image must be selected to continue.

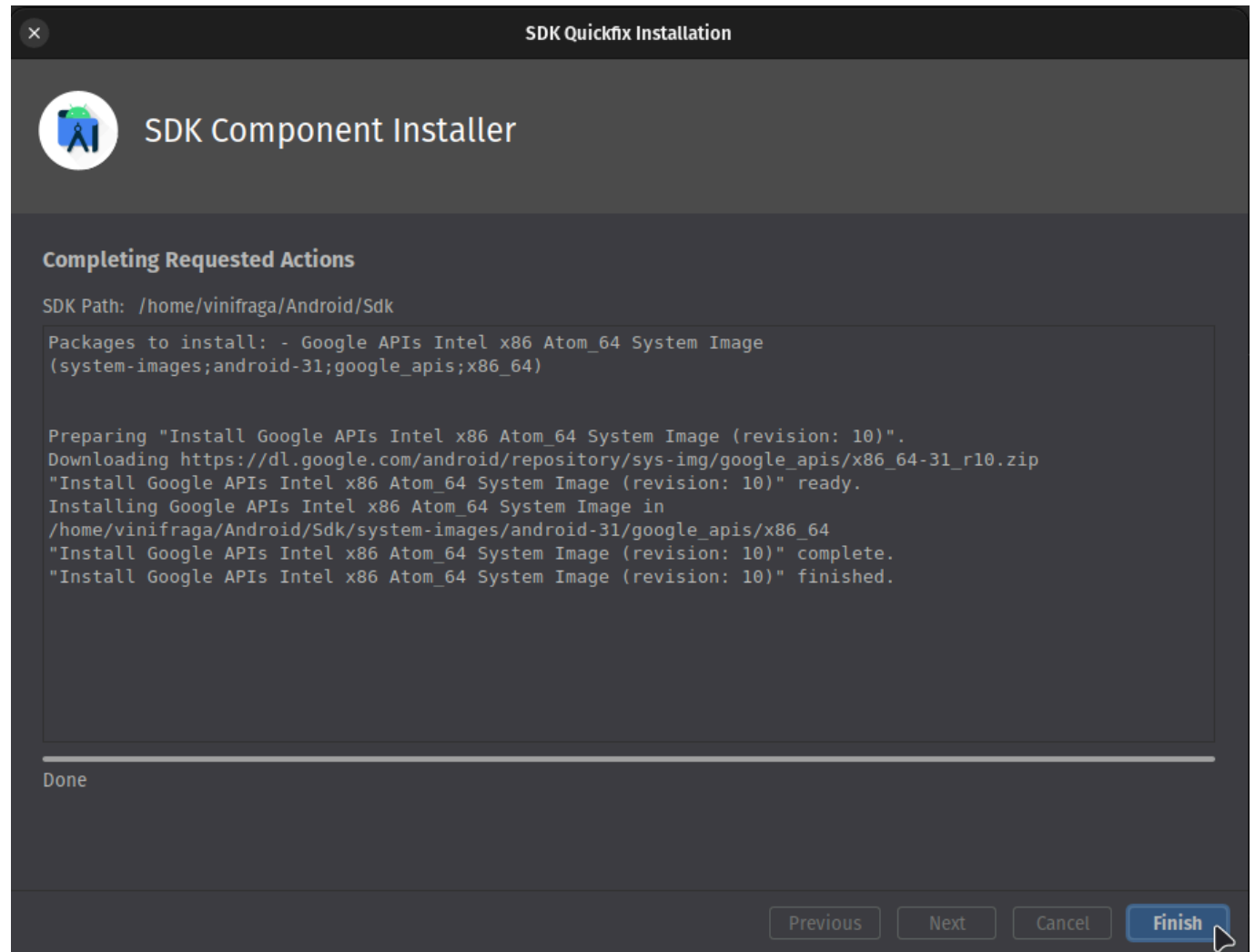
Cancel

Previous

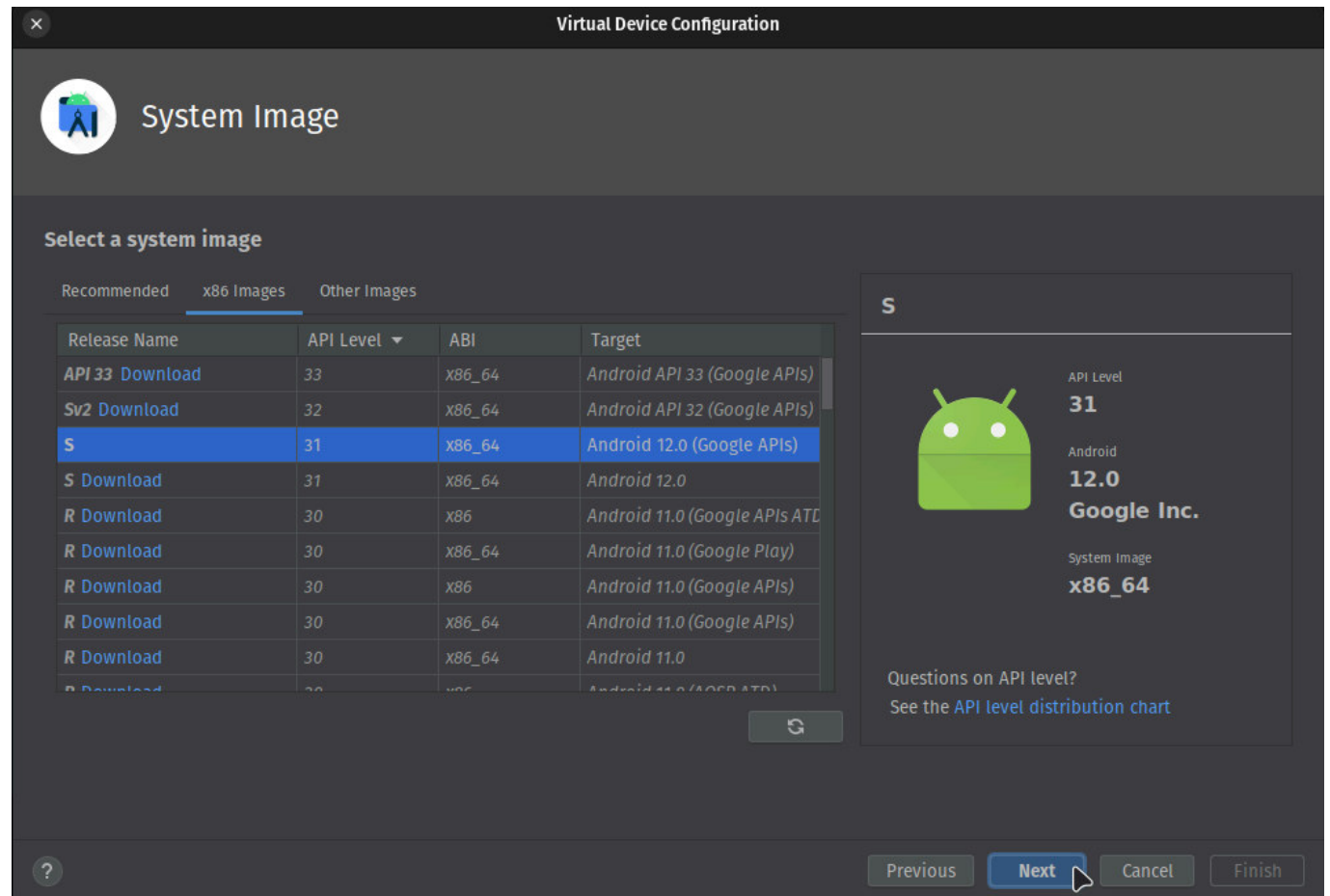
Next

Finish

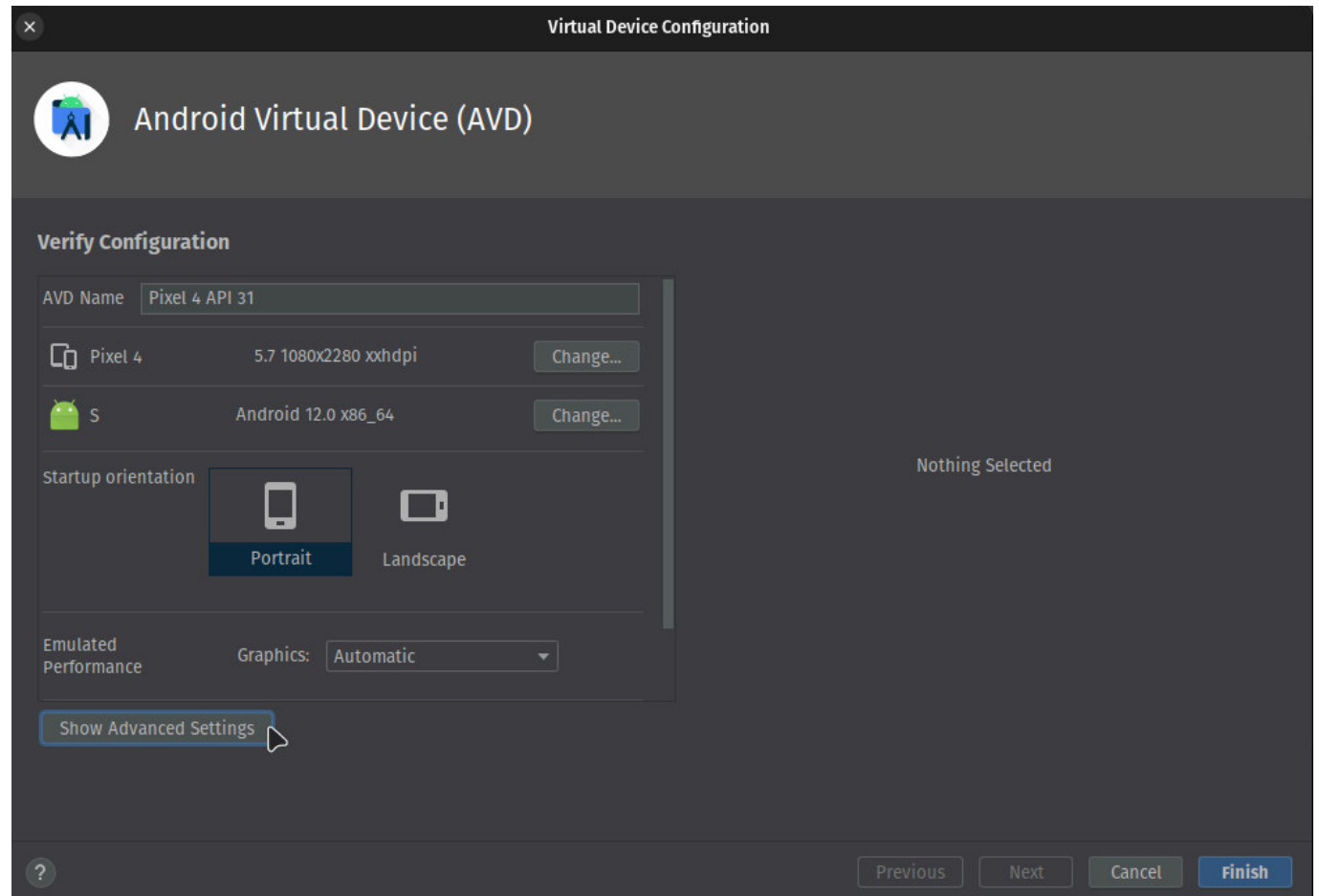
Uma nova janela irá abrir. Aceite as licenças e aguarde a instalação. Quando finalizar, clique em **Finish**



Com a sua imagem do sistema baixada, basta escolhê-la e clicar em **Next**

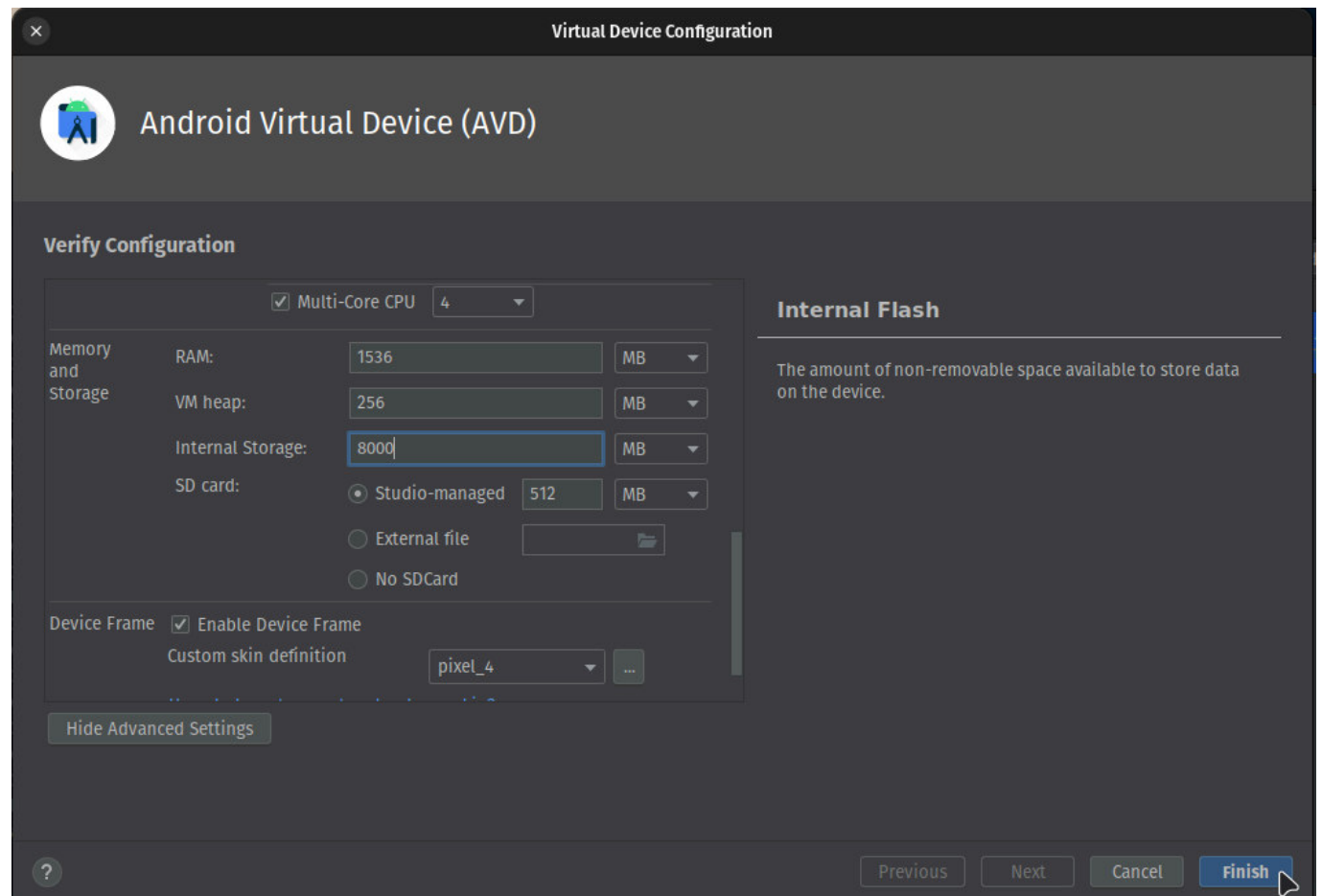


Por fim, serão apresentadas algumas configurações do seu emulador (Android Virtual Device - AVD). No canto inferior esquerdo, clique na opção **Show Advanced Settings**.

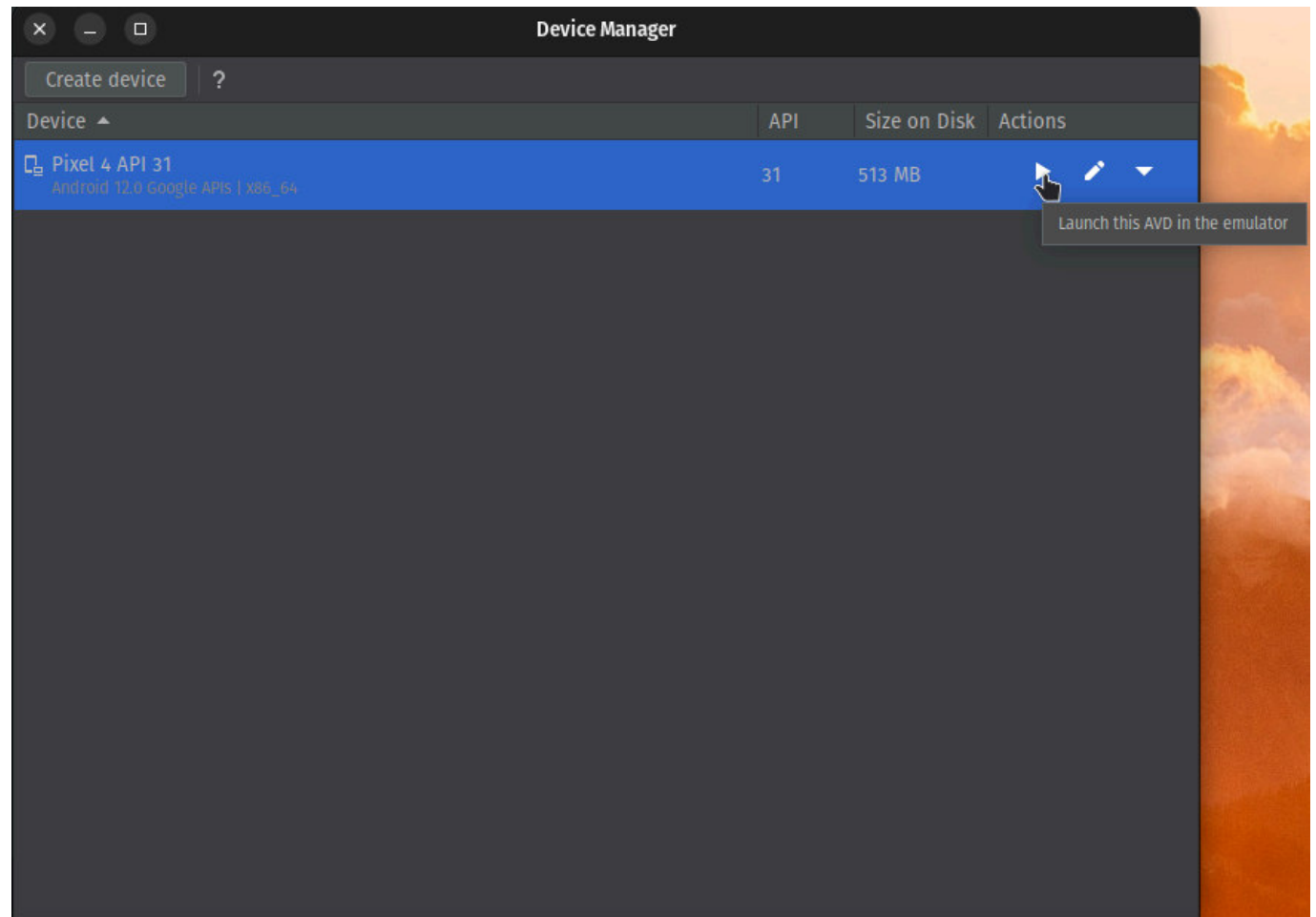


Deslize a tela até encontrar a opção **Memory and Storage**. Nela, aumente o **Internal Storage** até o valor desejado. Por padrão, esse valor vem como 800 MB, o que é muito pouco e acaba causando erro na instalação dos apps React Native no dispositivo.

Sugerimos escolher um valor acima de 4000 MB, no exemplo escolhemos 8000 MB. Definido esse valor, clique em **Finish**.



Agora, o seu emulador deve estar aparecendo na lista.



Iniciando o Emulador

Com o emulador pronto, basta clicar no botão **Play** e aguardar o AVD iniciar. Esse processo pode demorar, principalmente na primeira execução.

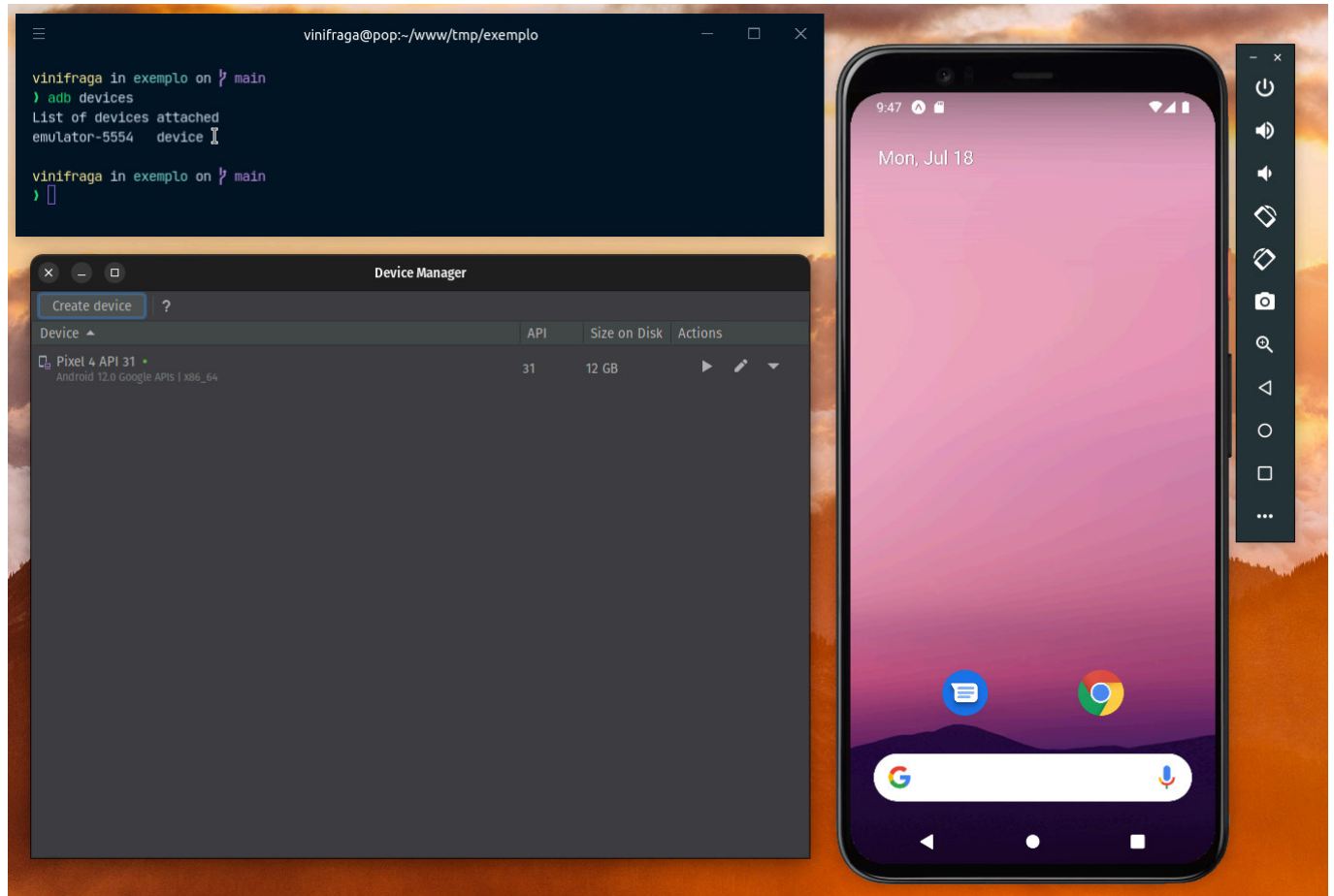
Quando o emulador terminar de carregar, abra o seu terminal e execute o comando:

```
BASH
```

[Copy](#)

```
adb devices
```

Se estiver tudo certo, deve aparecer uma lista com o nome do seu emulador android aberto (normalmente **emulator-5554**) e status **device**.



Executando app no Emulador

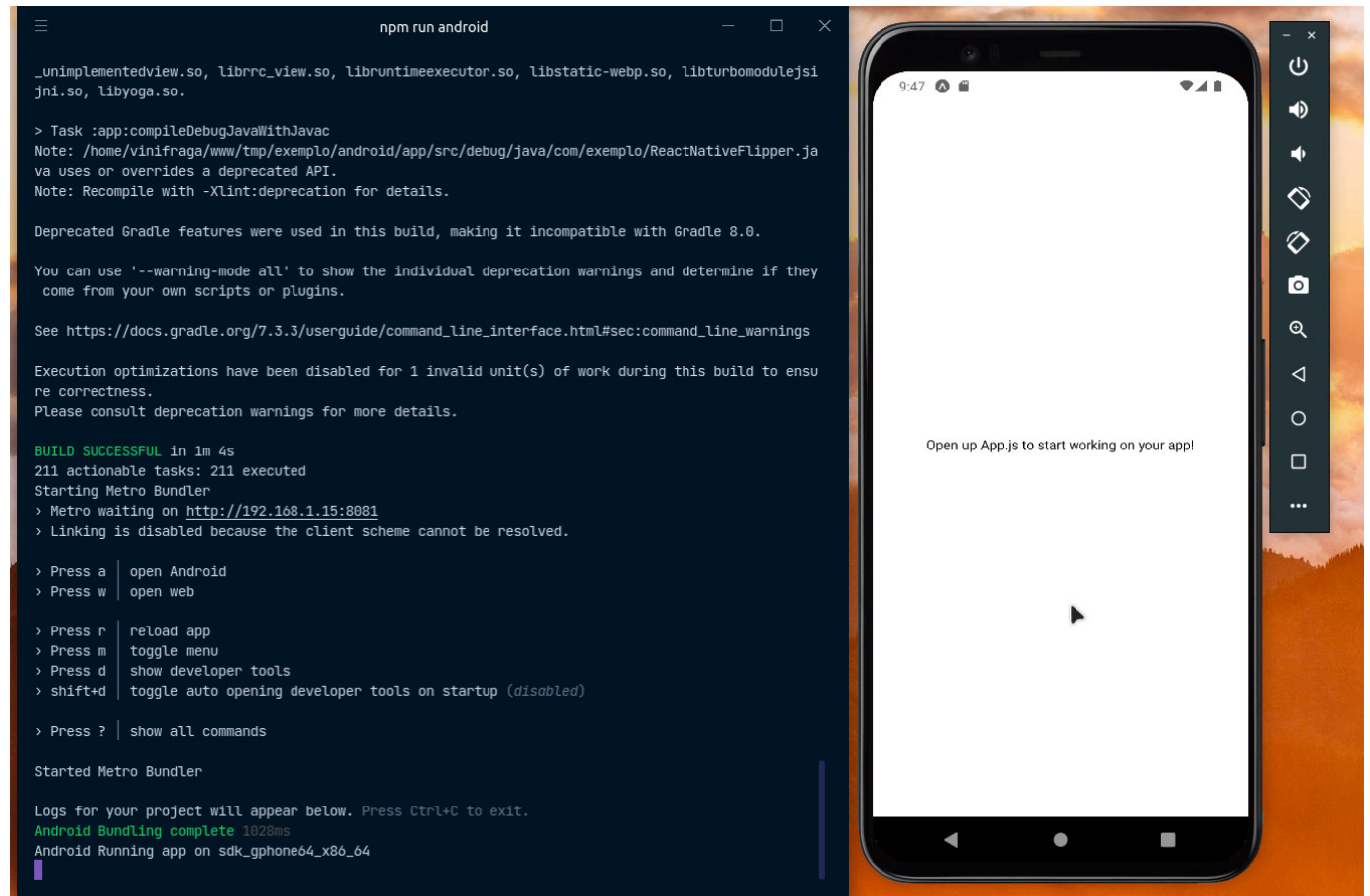
Expo Bare Workflow

Para esse passo é preciso que você já tenha criado o seu projeto Expo Bare Workflow. Caso não tenha criado ainda, você pode criar com o comando `npx create-expo-app --template` e escolher o template `Blank` (Bare).

Agora, com o emulador aberto, basta abrir um terminal, acessar a pasta do seu projeto e executar o comando:

```
BASH  
  
npm run android
```

[Copy](#)



React Native CLI

Para esse passo é preciso que você já tenha criado o seu projeto React Native CLI. Caso não tenha criado ainda, você pode criar com o comando `npx react-native init nomeDoSeuApp`

Agora, com o emulador aberto, basta abrir dois terminais e acessar a pasta do seu projeto em cada um deles. Em seguida, execute os comandos:

• Terminal 1

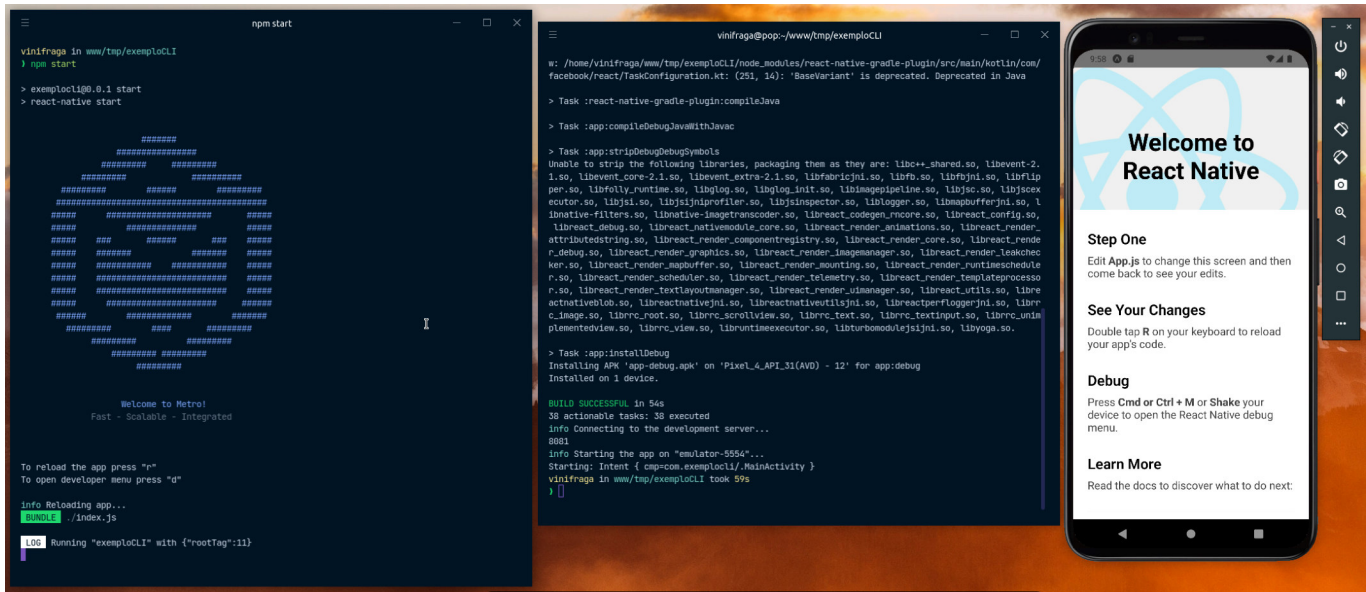
```
BASH  
  
npm start
```

Copy

• Terminal 2

```
BASH  
  
npm run android
```

Copy



[Edit this page on GitHub](#)



PREV

NEXT



macOS

Simulator iOS