

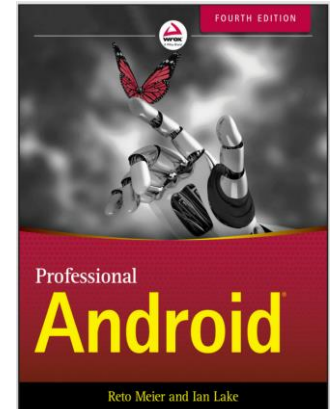
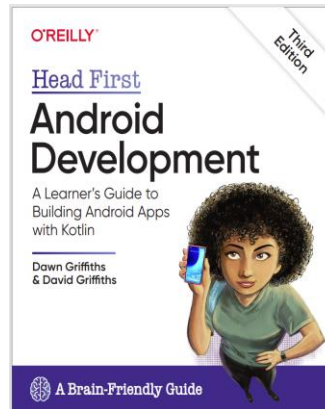
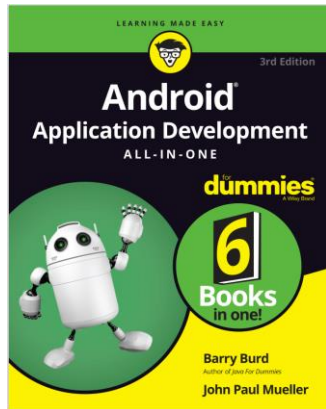
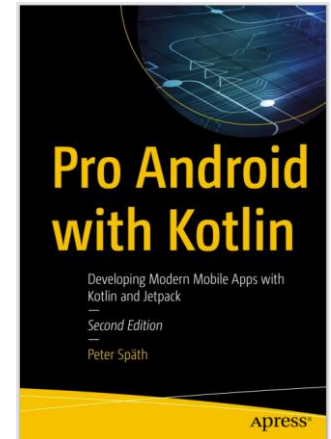
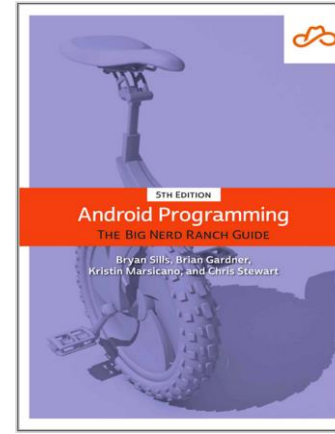
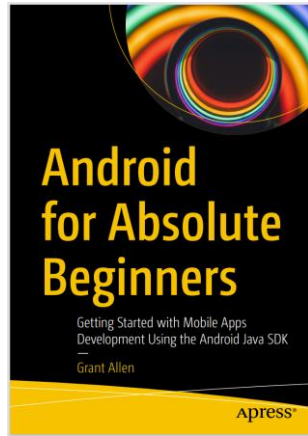
Desenvolvimento de Aplicativos

Unidade 4 – Segunda aplicação Android

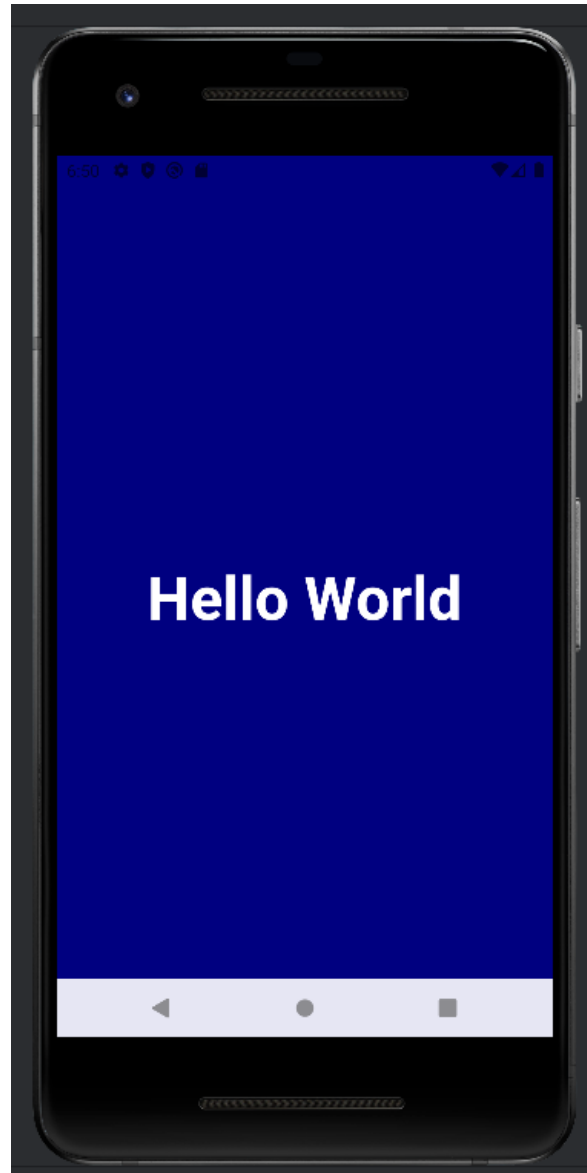


Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUVSP
aparecido.freitas@online.uscs.edu.br
aparecidovfreitas@gmail.com

Bibliografia

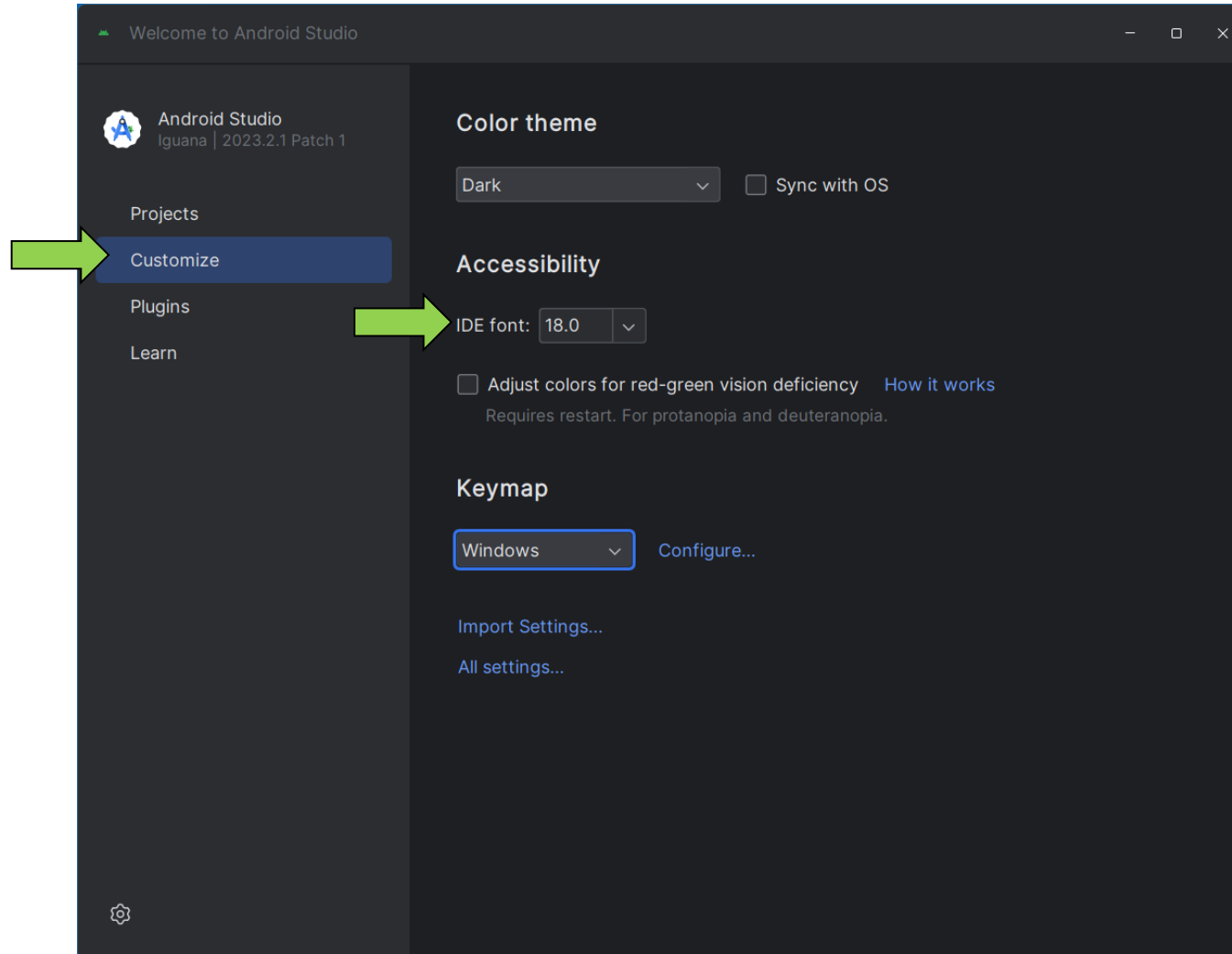


Segunda aplicação Android



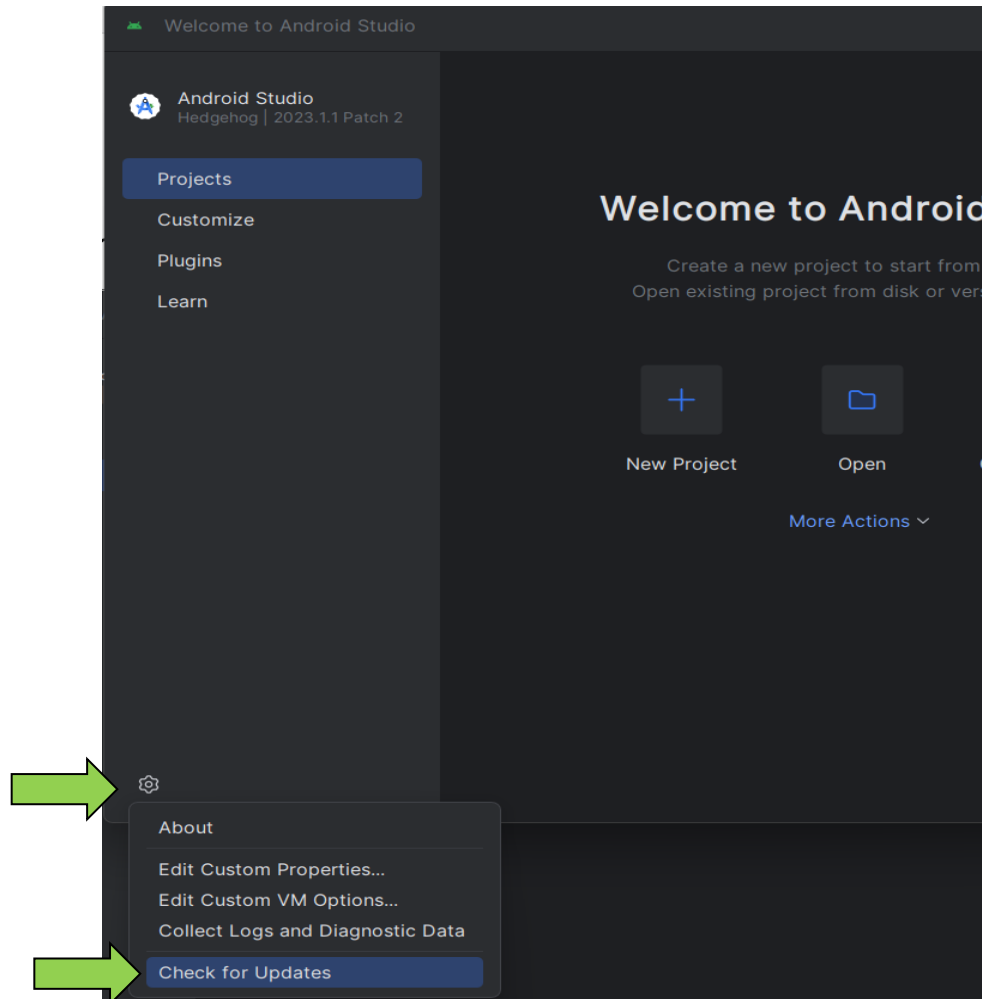
Segunda aplicação Android

✚ Abra o **Android Studio** e clique em “**Customize**”.



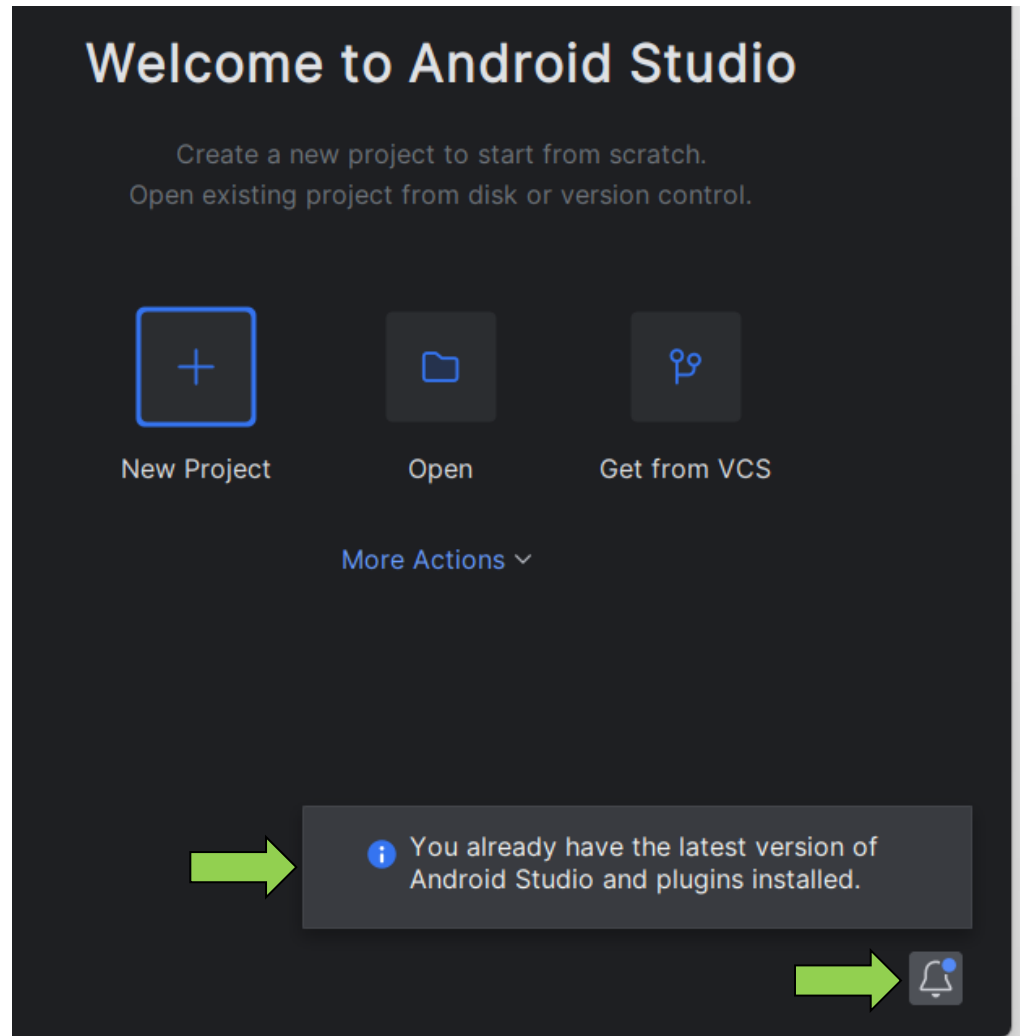
Segunda aplicação Android

- ✚ Abra o **Android Studio** e entre com a opção “**Check for Updates**”.



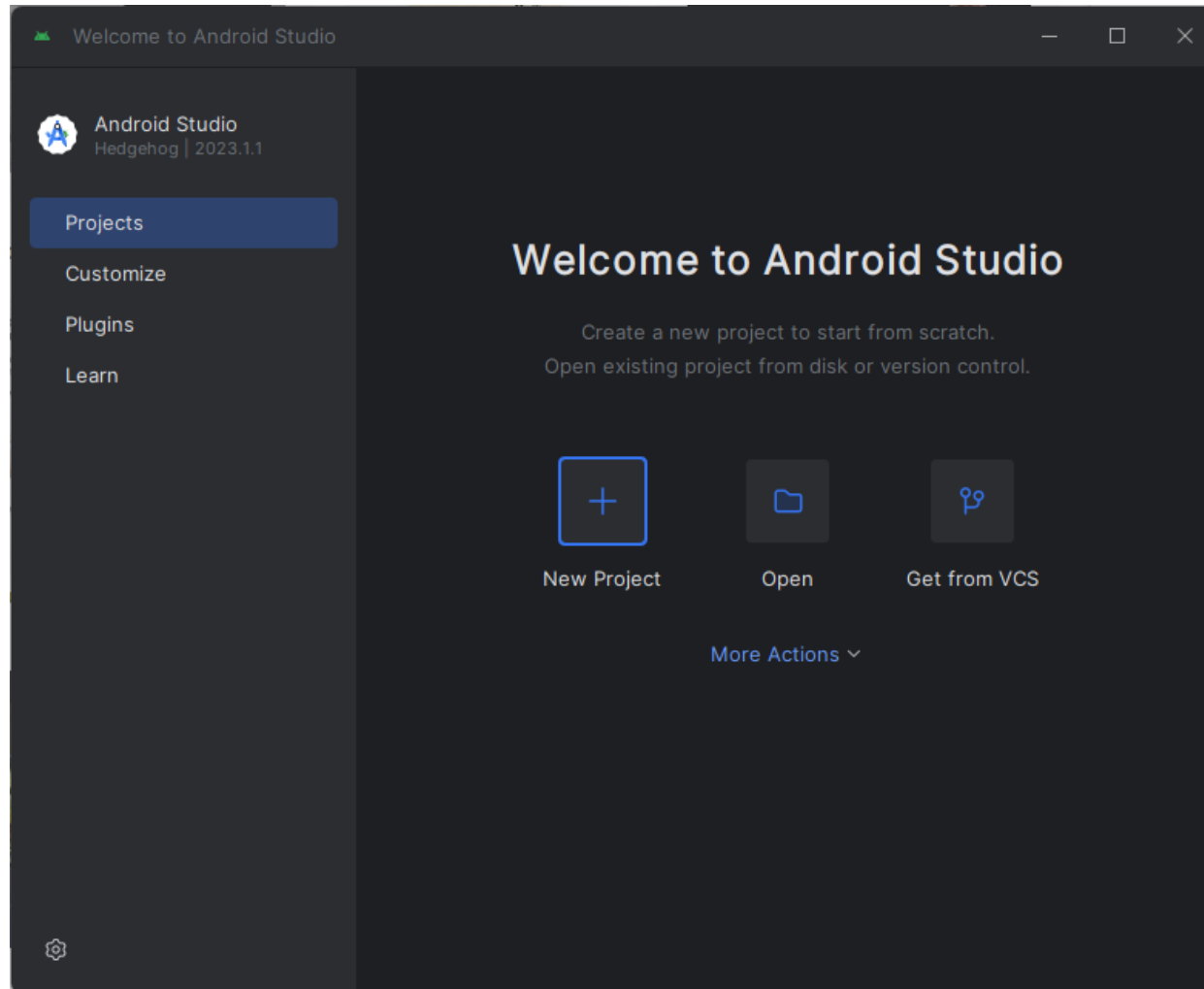
Segunda aplicação Android

- ⊕ Verifique a mensagem de retorno após “**Check for Updates**”.



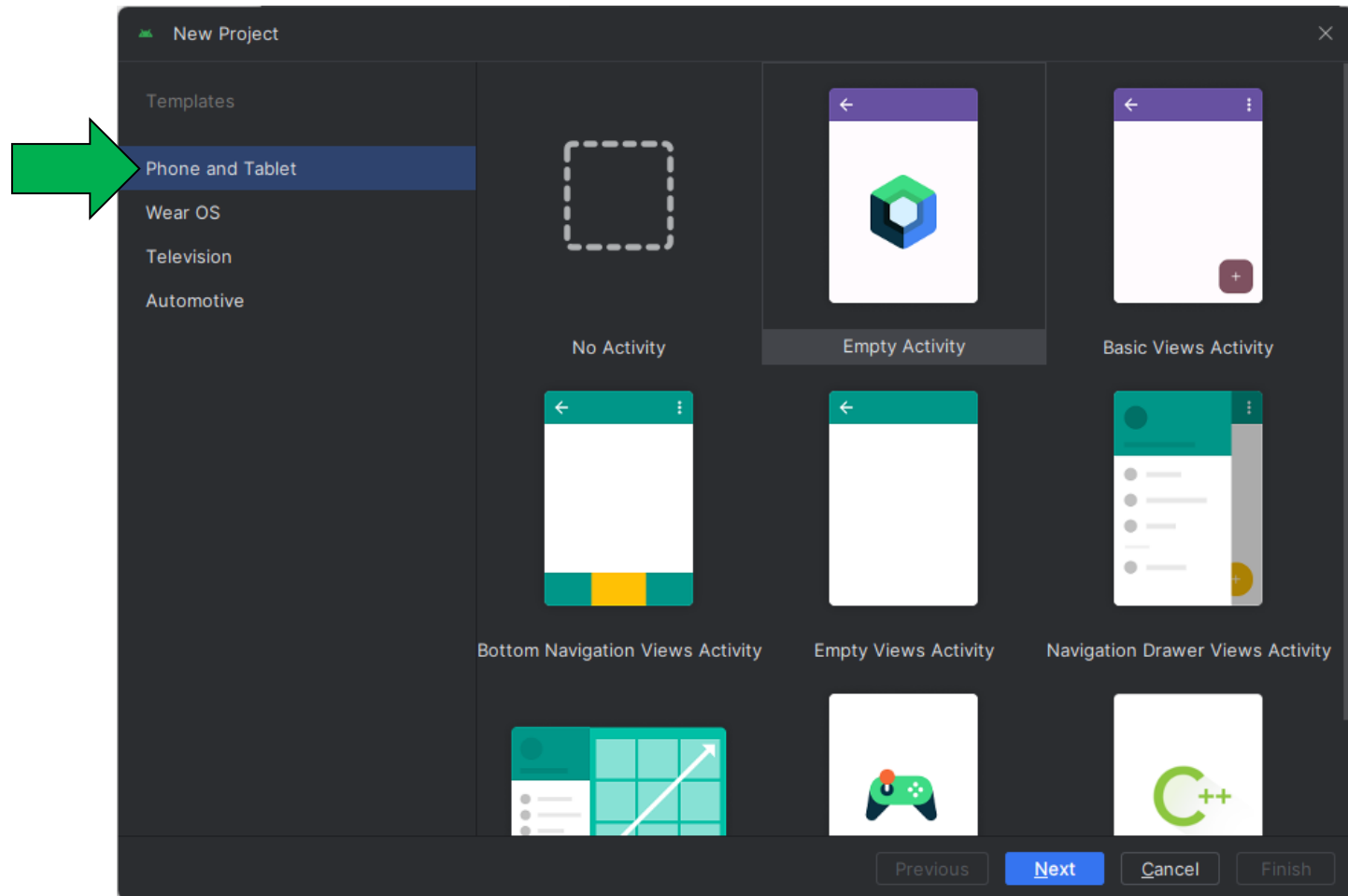
Segunda aplicação Android

✚ Abra o **Android Studio** e clique em “**New Project**”.



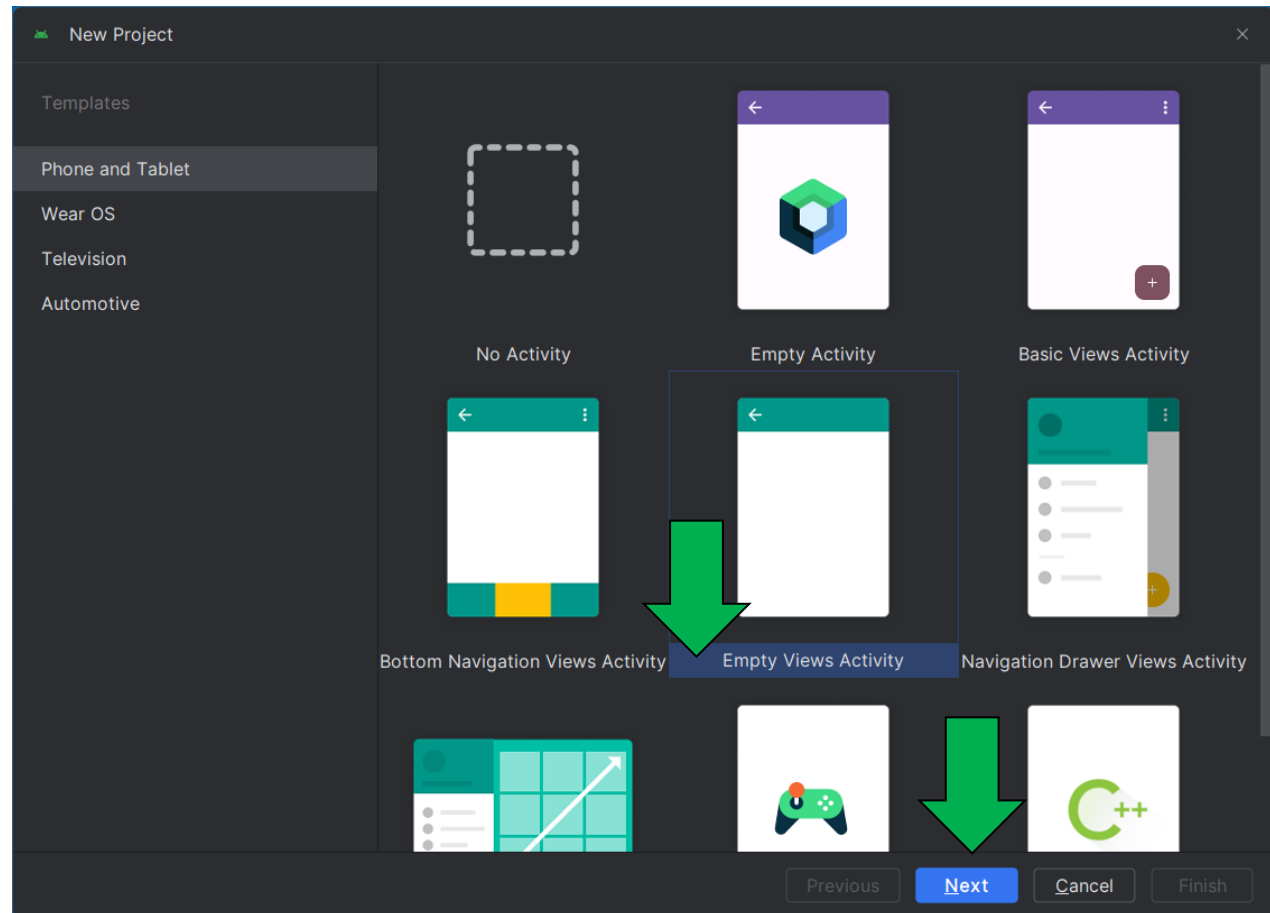
Segunda aplicação Android

- ⊕ Selecione agora o tipo de dispositivo a ser suportado pelo projeto;
- ⊕ Por **default**, **Phone and Tablet** é selecionado, exibindo uma variedade de tipos de telas:



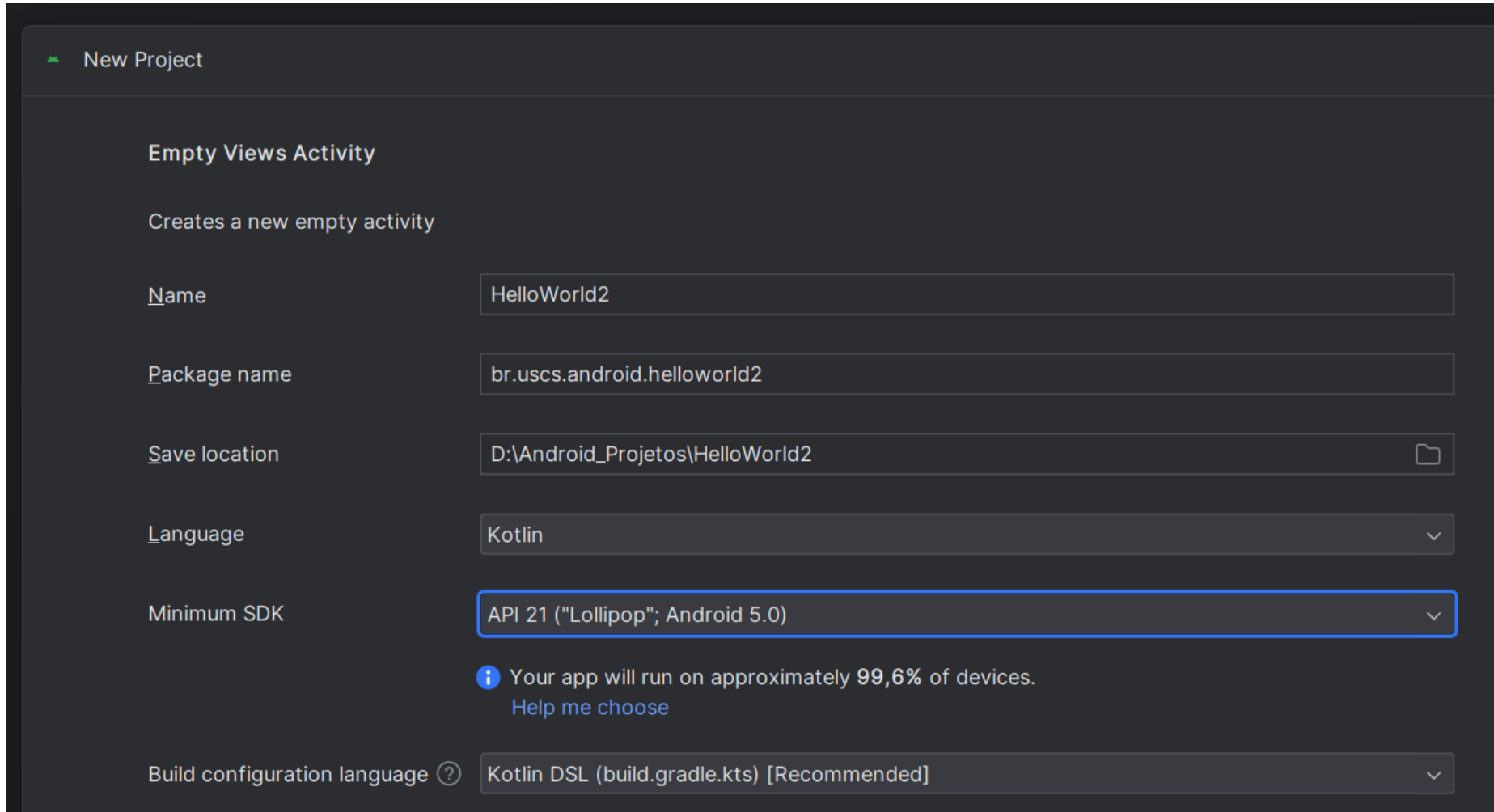
Segunda aplicação Android

- ✚ Clique em "**Empty Views Activity**" para escolhê-la como a configuração de projeto preferencial;
- ✚ Clique em "**Next**" no canto inferior direito da janela para prosseguir para a próxima etapa.



Segunda aplicação Android

✚ Preencha as informações do projeto:



New Project

Empty Views Activity

Creates a new empty activity

Name: HelloWorld2

Package name: br.uscs.android.helloworld2

Save location: D:\Android_Projetos\HelloWorld2

Language: Kotlin

Minimum SDK: API 21 ("Lollipop"; Android 5.0)

Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

Information: Your app will run on approximately 99,6% of devices. [Help me choose](#)

Segunda aplicação Android

- ⊕ **Nome** é onde você insere o nome do aplicativo. Para esta unidade, nomeie o aplicativo como **HelloWorld2**.
- ⊕ **Nome do Pacote** fornece ao aplicativo um nome de pacote, um conceito com o qual você pode estar familiarizado do **Java** ou **Kotlin**. Certifique-se de que este campo esteja preenchido como **br.uscs.android.helloworld2**.
- ⊕ **Local de Salvamento** informa ao **Android Studio** onde salvar o projeto. Você pode criar seu projeto em qualquer lugar que desejar. O botão de pasta à direita do campo fornece um navegador do sistema para ajudar na sua seleção.



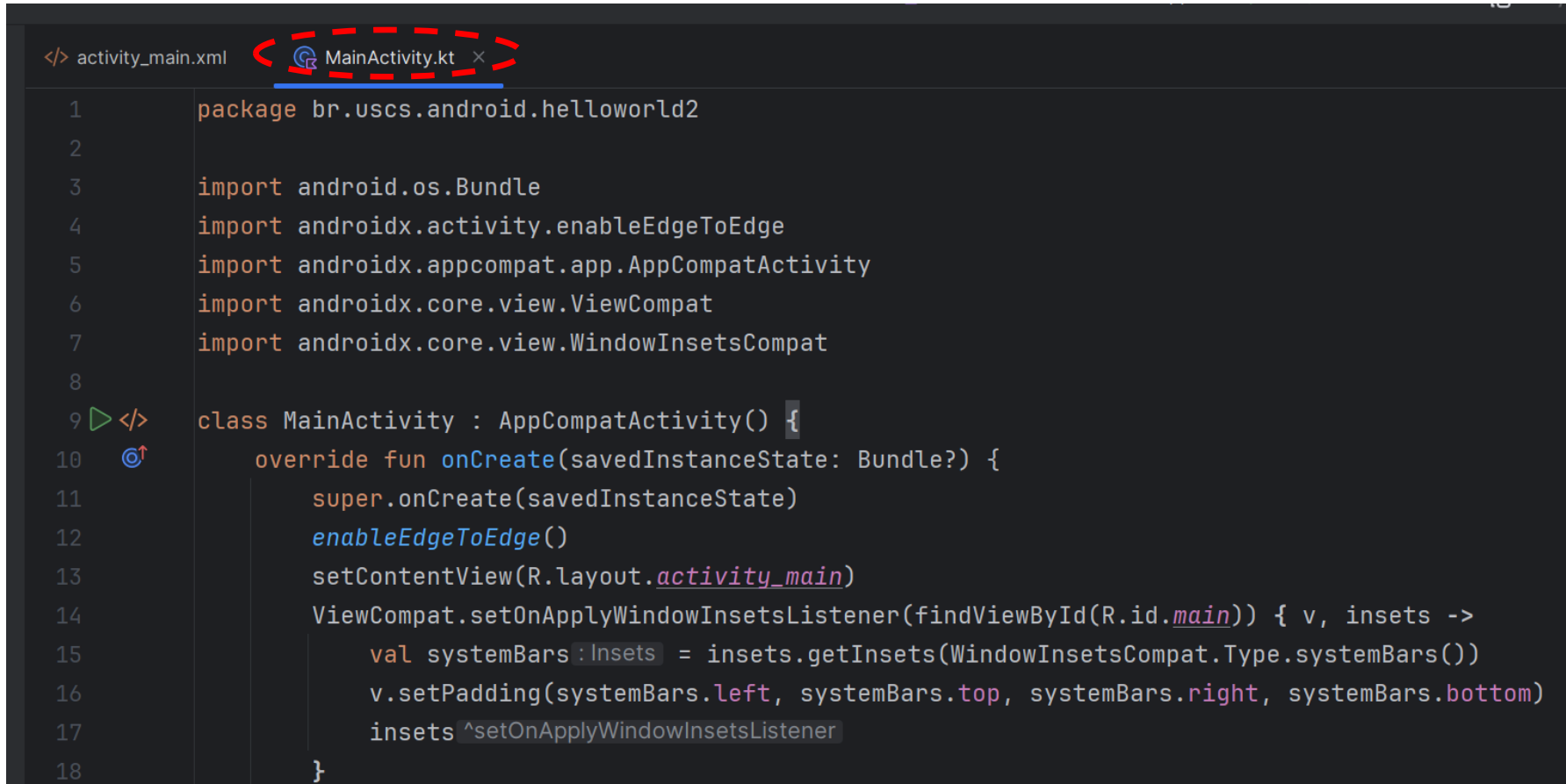
Segunda aplicação Android

- ⊕ **SDK Mínimo** define a versão mínima do **Android** que o aplicativo suportará. Mantenha a versão recomendada. Na versão que estou usando neste material, **API 21 (“Lollipop”; Android 5.0)**;
- ⊕ **Linguagem** informa ao **Android Studio** a linguagem que seu projeto requer. **Kotlin** já está selecionado, então nenhuma ação é necessária aqui;
- ⊕ **Concluir**, o botão no canto inferior direito da janela, completa a configuração do projeto. Clique neste quando estiver pronto para prosseguir;
- ⊕ **Android Studio** utilizará estas informações e alocará as bibliotecas e recursos necessários para gerar um novo projeto.



Segunda aplicação Android

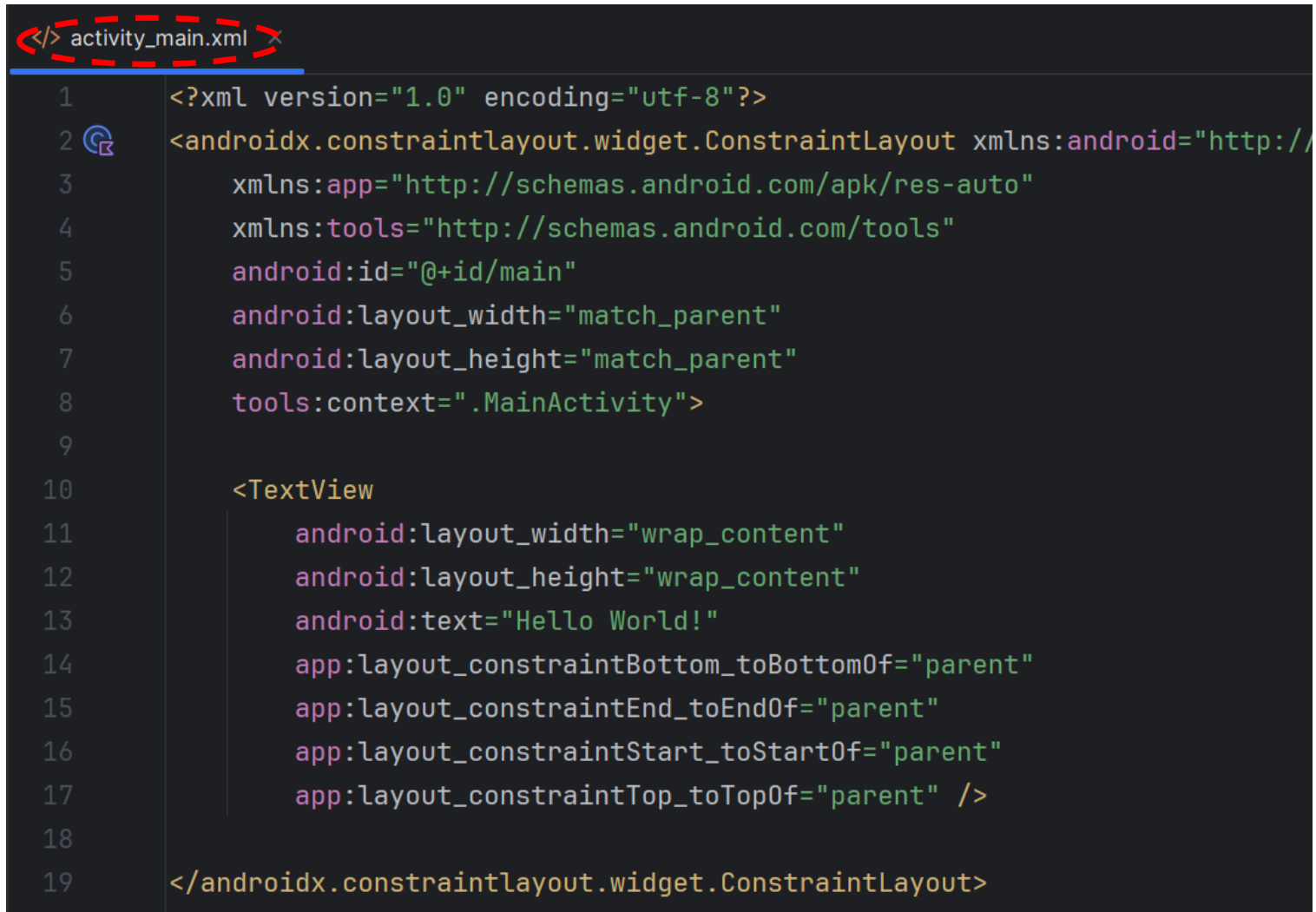
⊕ Projeto criado !



```
1 package br.uscs.android.helloworld2
2
3 import android.os.Bundle
4 import androidx.activity.enableEdgeToEdge
5 import androidx.appcompat.app.AppCompatActivity
6 import androidx.core.view.ViewCompat
7 import androidx.core.view.WindowInsetsCompat
8
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         enableEdgeToEdge()
13         setContentView(R.layout.activity_main)
14         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
15             val systemBars : Insets = insets.getInsets(WindowInsetsCompat.Type.systemBars())
16             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
17             insets ^setOnApplyWindowInsetsListener
18         }
```

Primeira aplicação Android

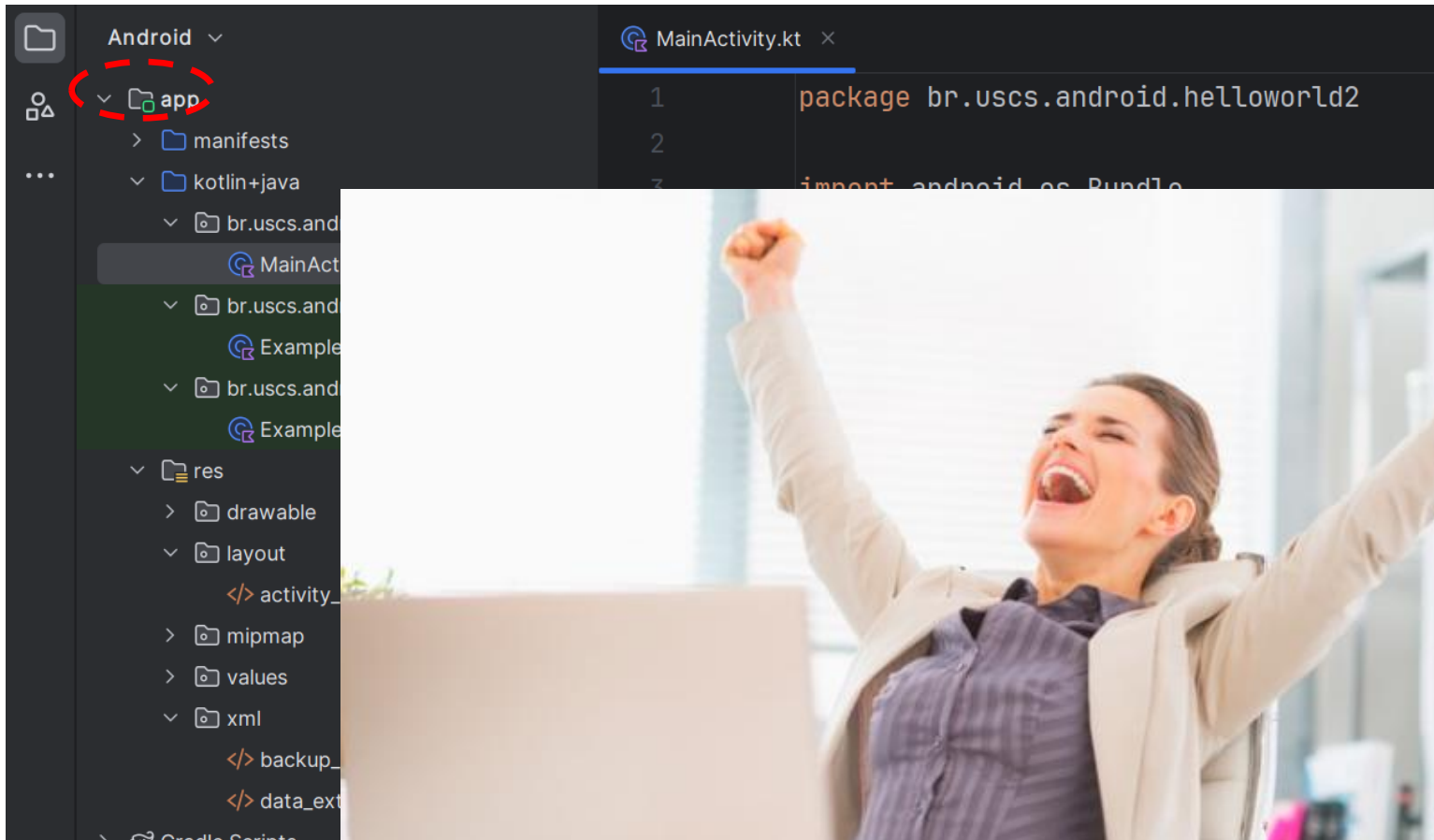
⊕ Projeto criado !



```
</> activity_main.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/main"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <TextView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Hello World!"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintEnd_toEndOf="parent"
16         app:layout_constraintStart_toStartOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

Primeira aplicação Android

📍 Projeto criado !



Segunda aplicação Android

- ⊕ Criaremos um aplicativo simples em **Kotlin** que exiba "**Hello World**" com um **fundo azul escuro** e o texto em **negrito** e de tamanho grande;



Segunda aplicação Android

- ⊕ Iremos modificar o Arquivo **activity_main.xml**;
- ⊕ Este arquivo se encontra no diretório **res/layout**;
- ⊕ Substitua o conteúdo do arquivo pelo código exibido no próximo slide.



activity_main.xml;

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000080">

    <TextView
        android:id="@+id/helloWorldTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/Hello"
        android:textSize="30sp"
        android:textStyle="bold"
        android:layout_centerInParent="true"
        android:textColor="#FFFFFF" />

</RelativeLayout>
```

strings.xml;

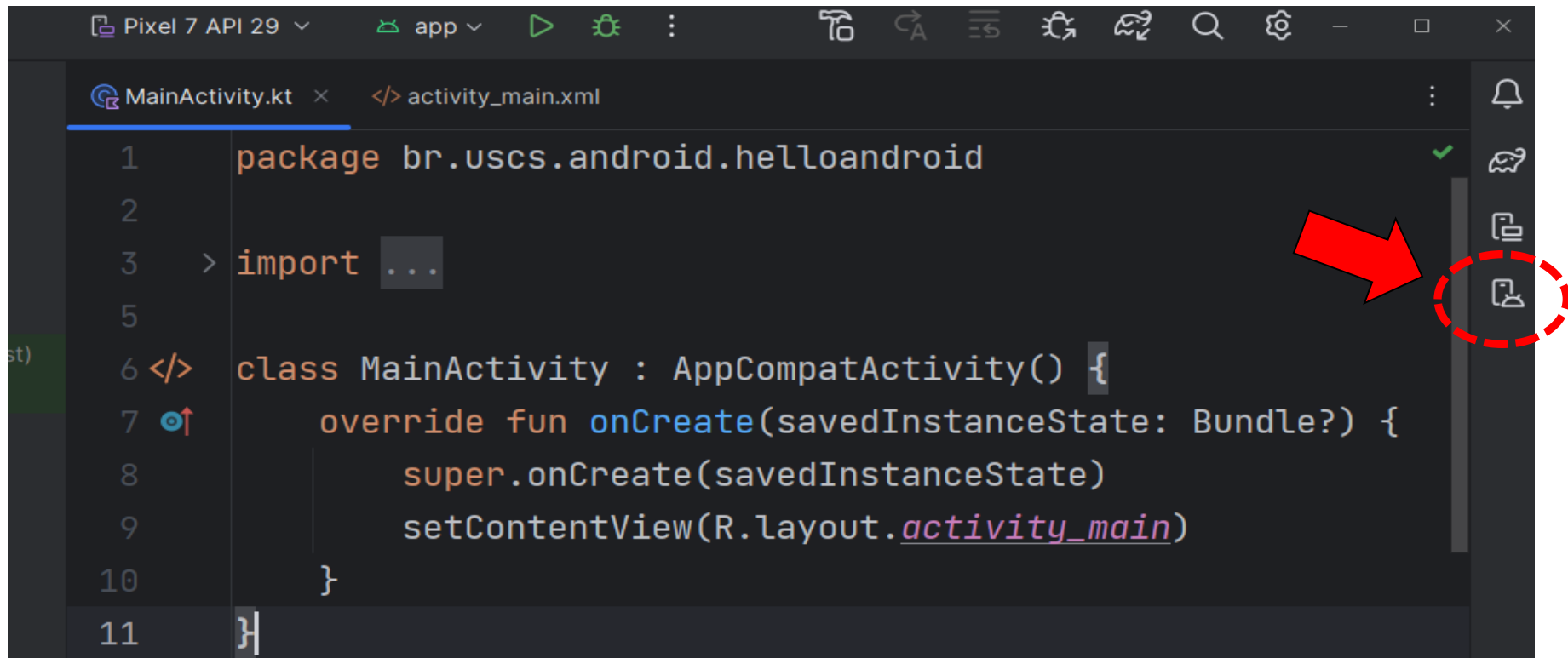
```
app
├── src
│   ├── main
│   │   ├── res
│   │   │   ├── values
│   │   │   │   └── strings.xml
```

strings.xml

```
<resources>
    <string name="app_name">HelloWorld2</string>
    <string name="Hello">Hello World</string>
</resources>
```

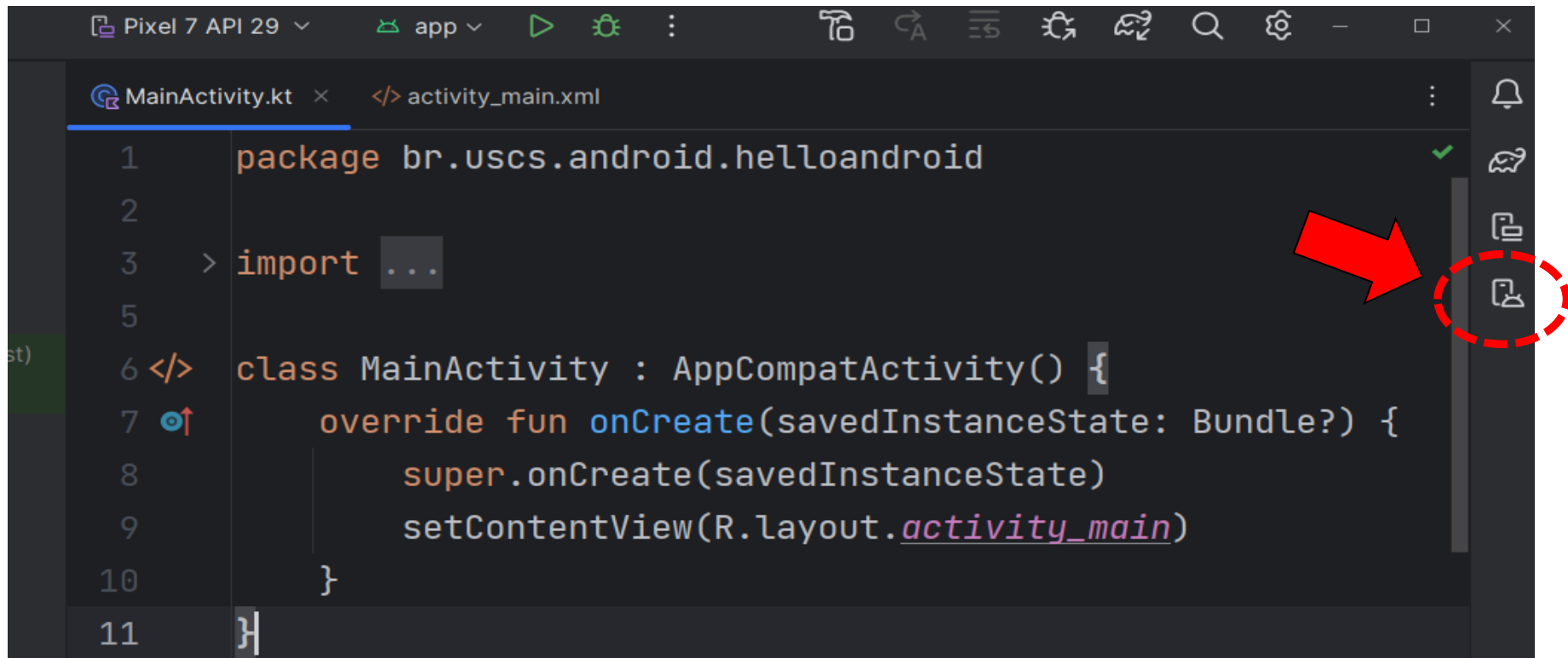
Criando um Device Virtual

- ✦ Olhar para o seu editor e arquivos é bom, mas uma vez que você terminou de escrever o código, **você precisa executar o aplicativo** para vê-lo em ação;
- ✦ Antes que você possa fazer isso, você precisa de um dispositivo - seja ele **real** ou **virtual**. **Olhe para o botão** destacado na imagem a seguir.



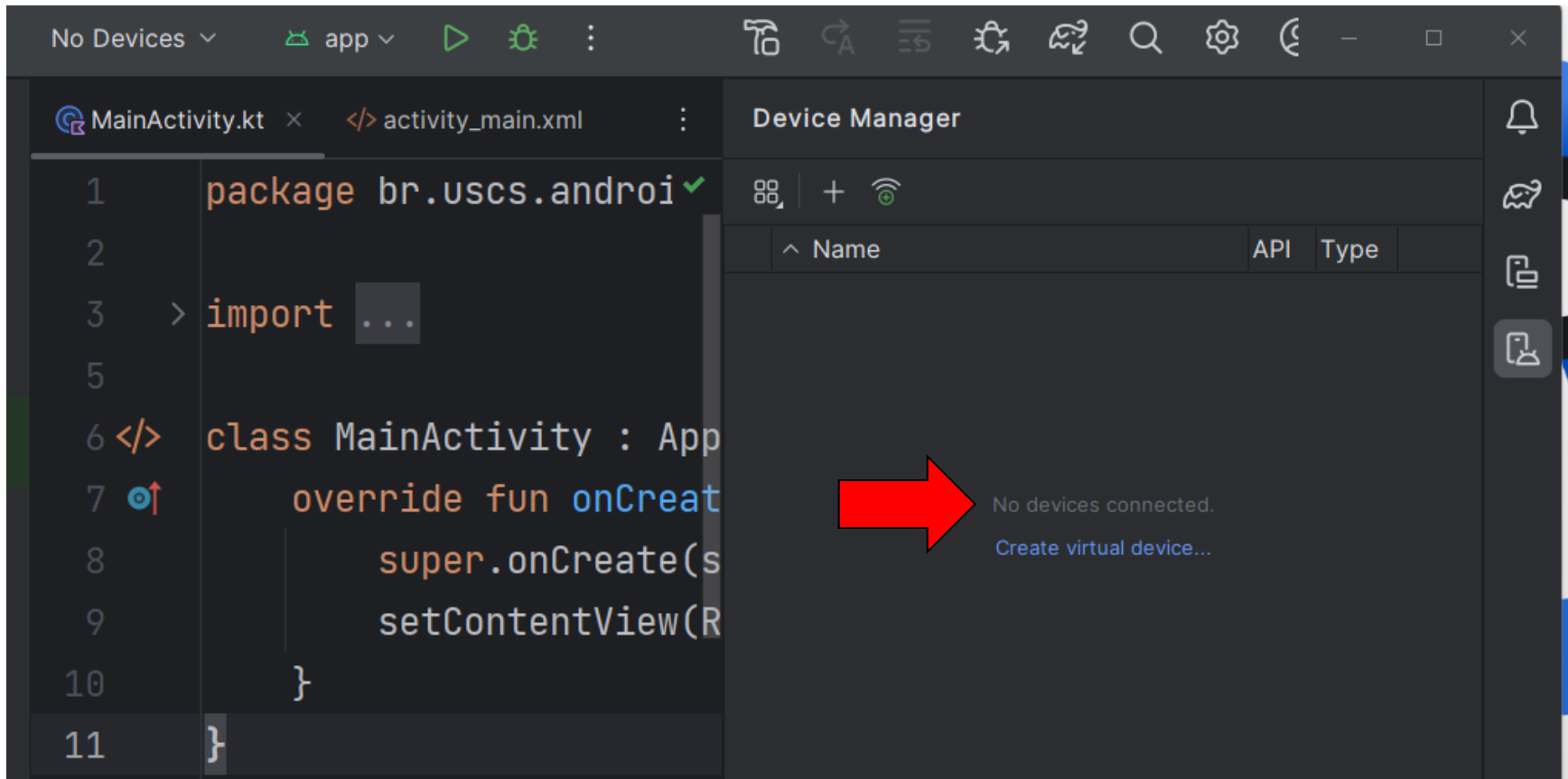
AVD Manager

- ⊕ Este botão abre uma nova janela para exibir os **Android Virtual Devices** disponíveis, usualmente abreviados por **AVD**; Um **AVD** é um **emulador** que age como um dispositivo em seu computador; Isto permite que se teste o aplicativo sem a necessidade de um dispositivo físico real.



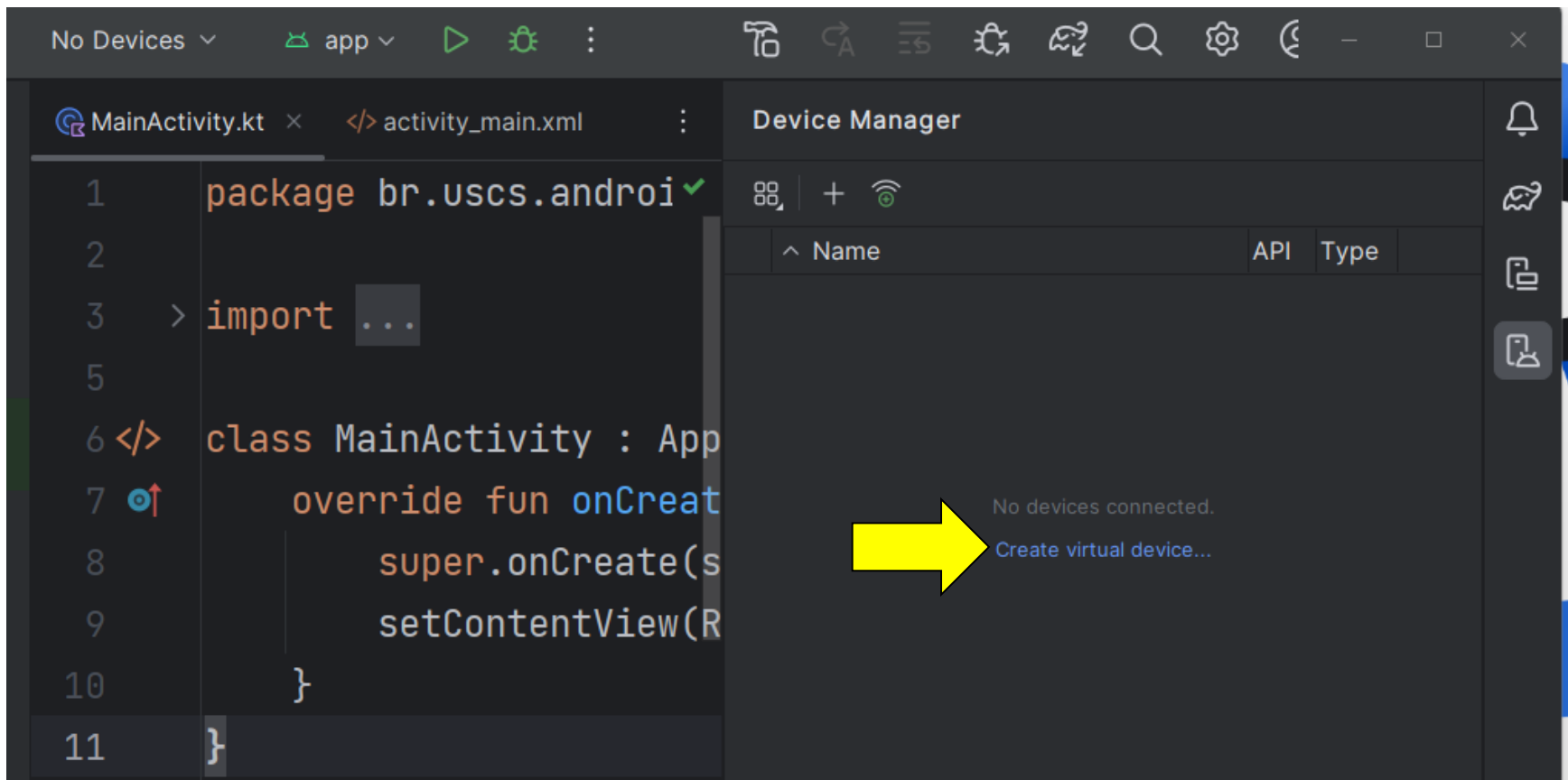
AVD Manager

- ✦ Esta janela exibe todos os **AVD's disponíveis** que existem em sua máquina;
- ✦ Até o momento, **nenhum** emulador está disponível, uma vez que acabamos de instalar o **Android Studio**.



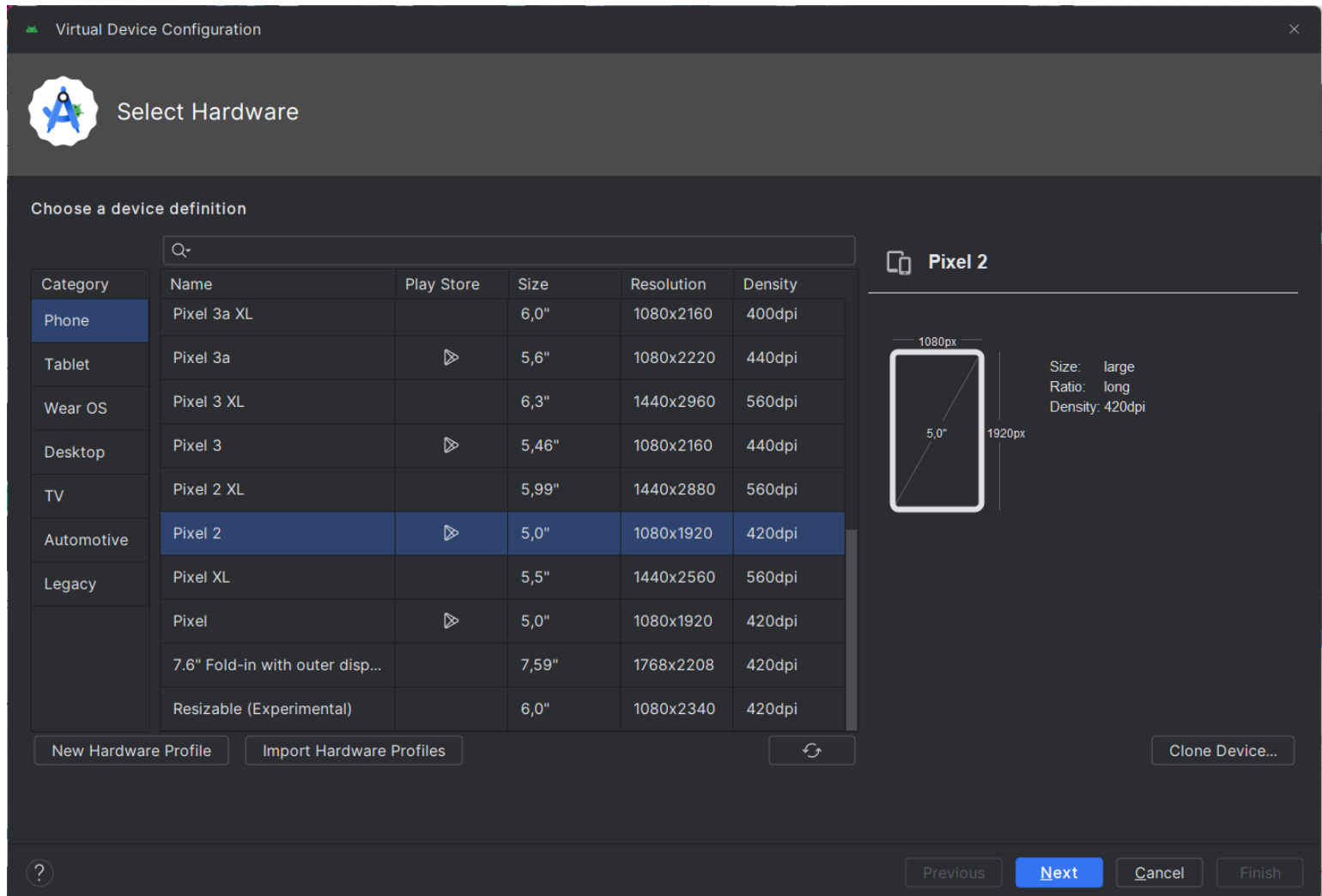
AVD Manager

- ✦ Clique agora em “**Create Virtual Device**” para que seja exibida a janela de “**Select Hardware**”.



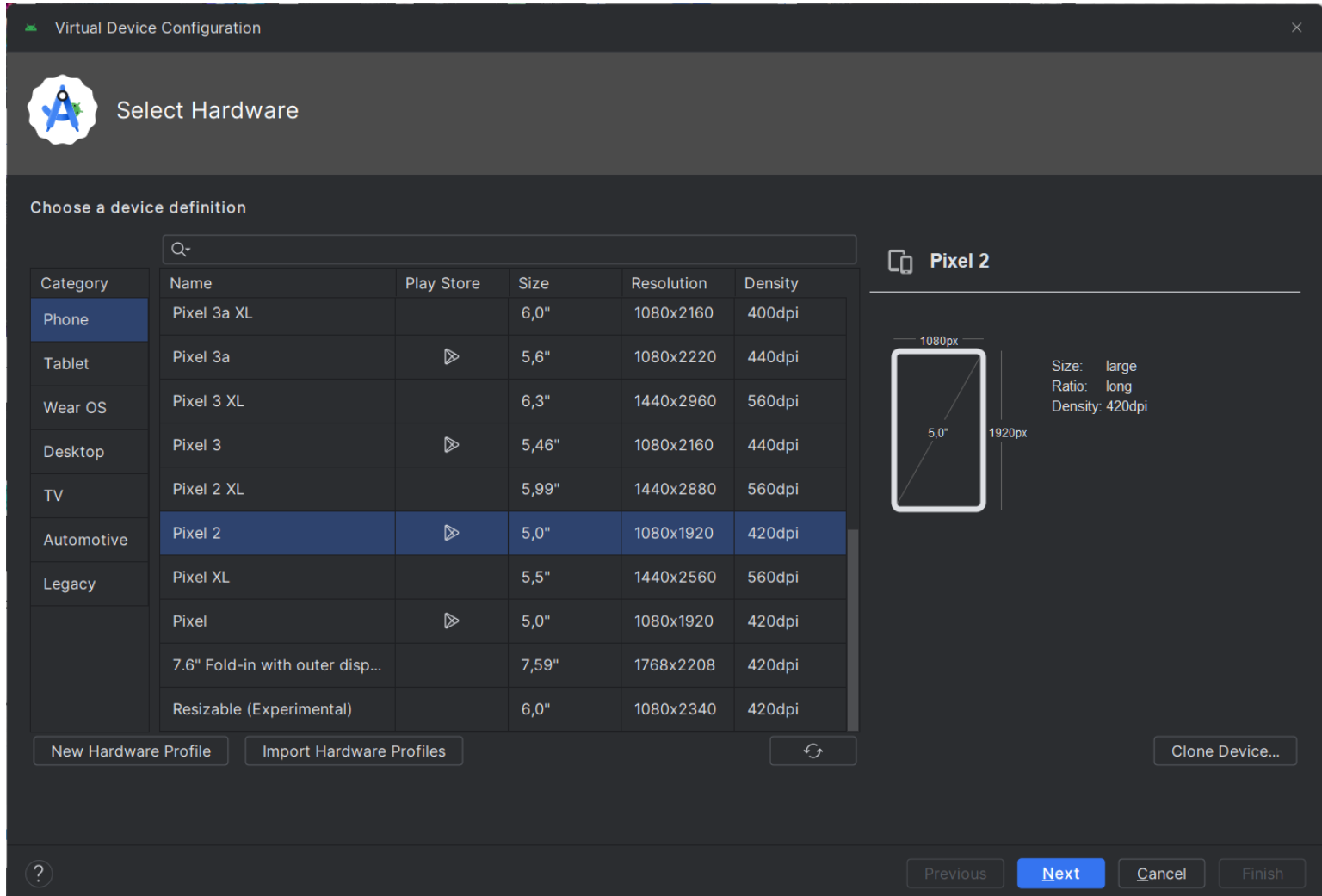
AVD Manager

- Na janela “**Select Hardware**” pode-se selecionar o tipo de **Device** que se deseja emular.



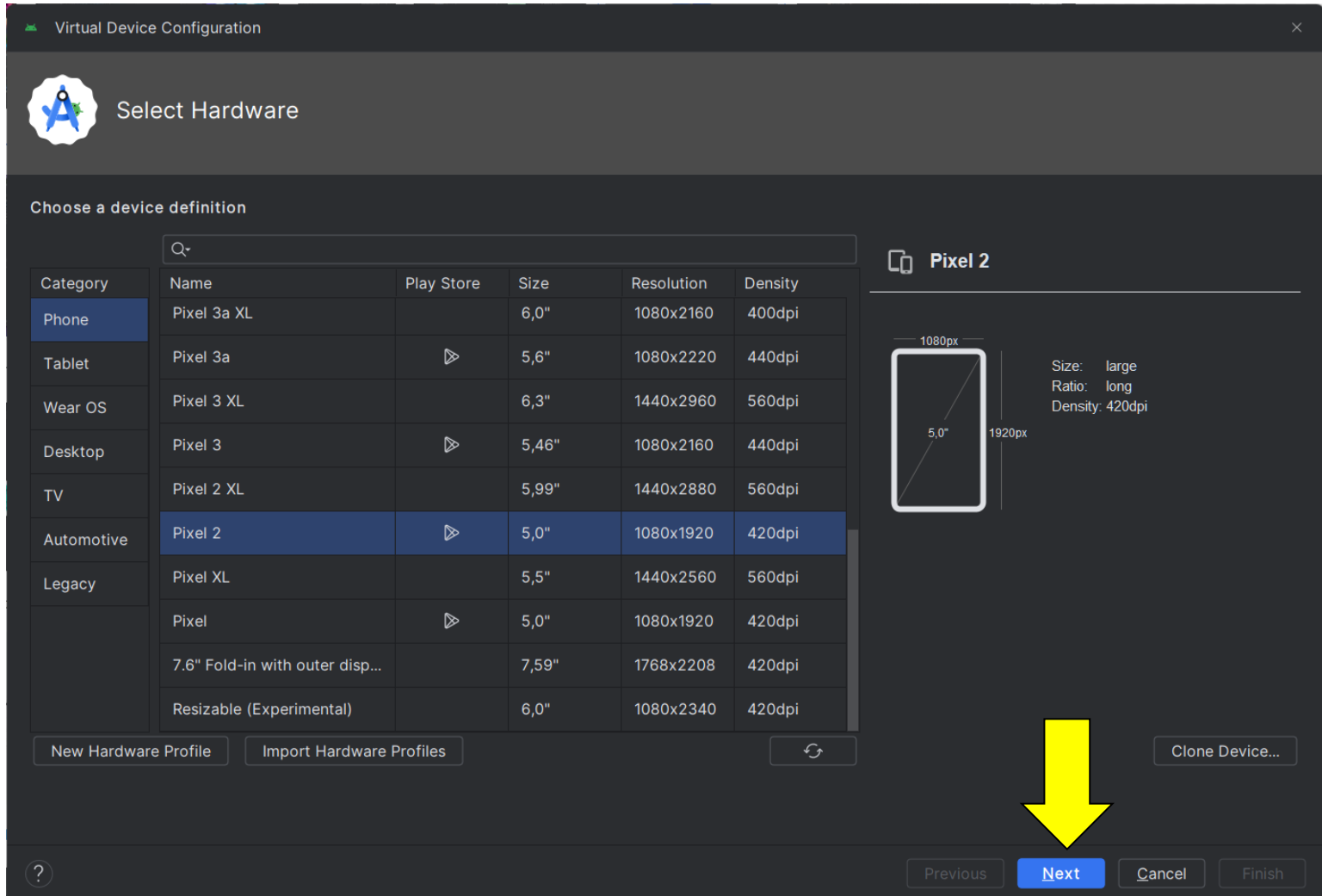
AVD Manager

- Uma boa opção é **Pixel 2**, pois emula um **dispositivo real** que é usado por muitas pessoas.



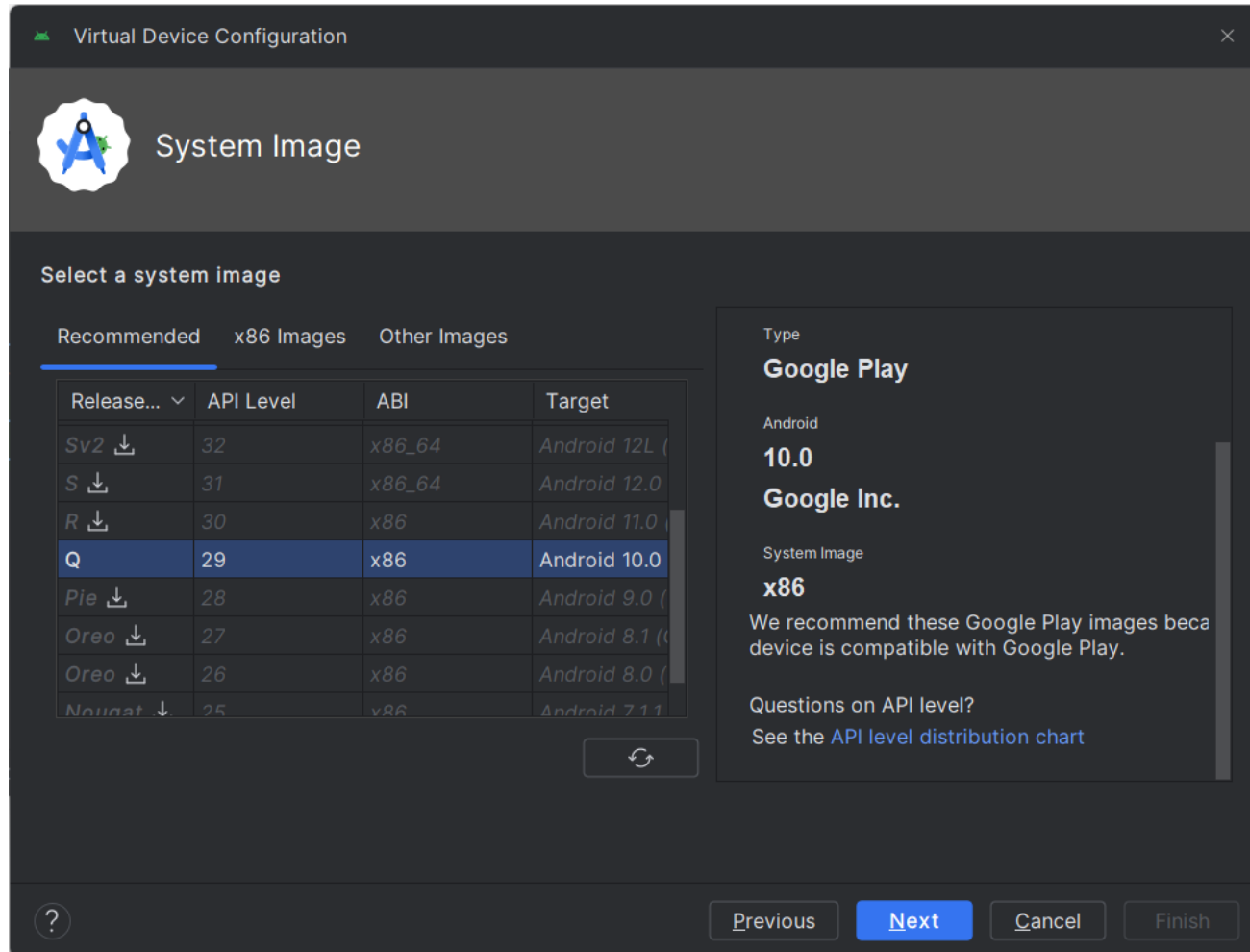
AVD Manager

✚ No canto inferior direito da janela, clique em **Next**.



AVD Manager

- ✚ A janela “**System Image**” será exibida. Nesta janela, pode-se **selecionar a versão do Android** que será executada pelo emulador.



AVD Manager

- ⊕ Existem algumas **abas** que ficam no **topo** da lista;
- ⊕ A mais interessante é a “**Recommended**”, que mostra uma lista de versões do **Android** que o **Google** recomenda que você use ao testar seus aplicativos;
- ⊕ As **versões em cinza claro não** estão instaladas na sua máquina e você poderá instalá-las.



AVD Manager



System Image

Select a system image

Recommended x86 Images Other Images

Release Name ▾	API Level	ABI	Target
S	31	x86_64	Android 12.0 (Google)
R	30	x86	Android 11.0 (Google)
Q	29	x86	Android 10.0 (Google)
Pie	28	x86	Android 9.0 (Google)
Oreo	27	x86	Android 8.1 (Google)
Oreo	26	x86	Android 8.0 (Google)
Nougat	25	x86	Android 7.1.1 (Google)
Nougat	24	x86	Android 7.0 (Google)



API Level

25

Type

Google Play

Android

7.1.1

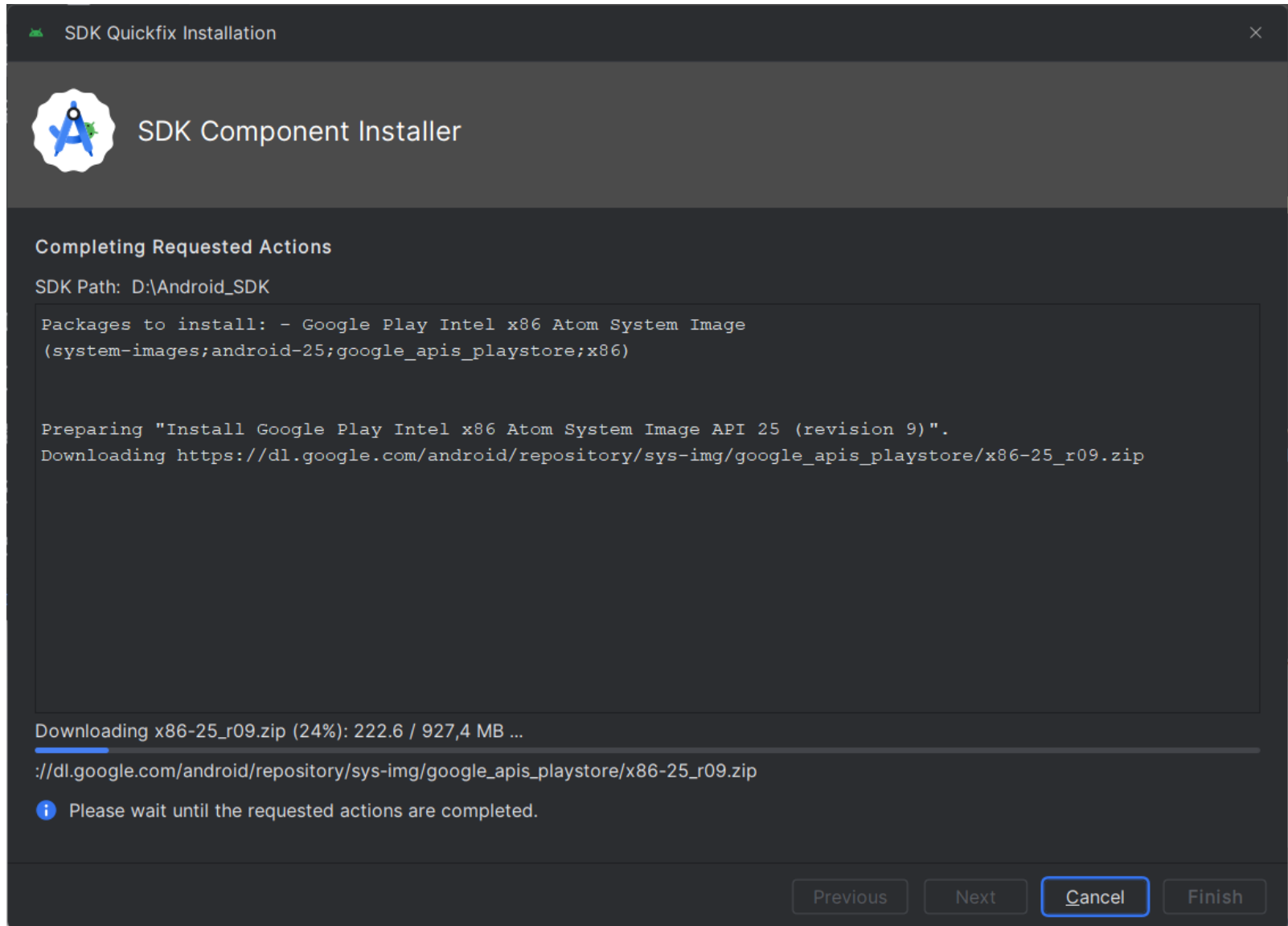
Google Inc.

System Image

x86

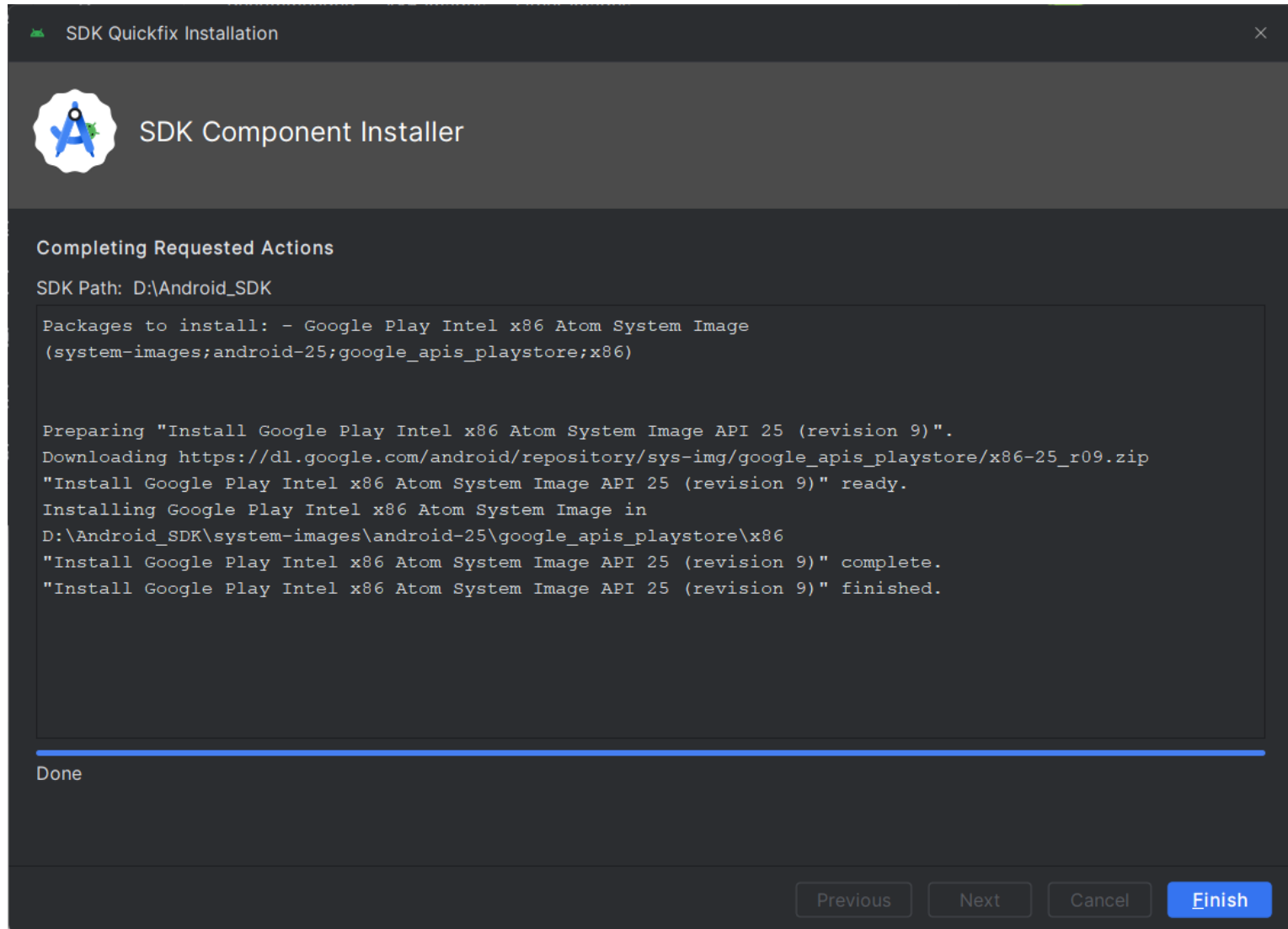
We recommend these Google Play images because your device is compatible with Google Play.

AVD Manager



AVD Manager

📍 Ao encerrar o **download**, clique em **Finish**.



AVD Manager

✚ Após a seleção da imagem do sistema, clique em **Next**.

Virtual Device Configuration

System Image

Select a system image

Recommended x86 Images Other Images

Release Name	API Level	ABI	Target
S	31	x86_64	Android 12.0 (Google)
R	30	x86	Android 11.0 (Google)
Q	29	x86	Android 10.0 (Google)
Pie	28	x86	Android 9.0 (Google)
Oreo	27	x86	Android 8.1 (Google)
Oreo	26	x86	Android 8.0 (Google)
Nougat	25	x86	Android 7.1.1 (Google)
Nougat	24	x86	Android 7.0 (Google)

API Level
24

Type
Google Play

Android
7.0

Google Inc.

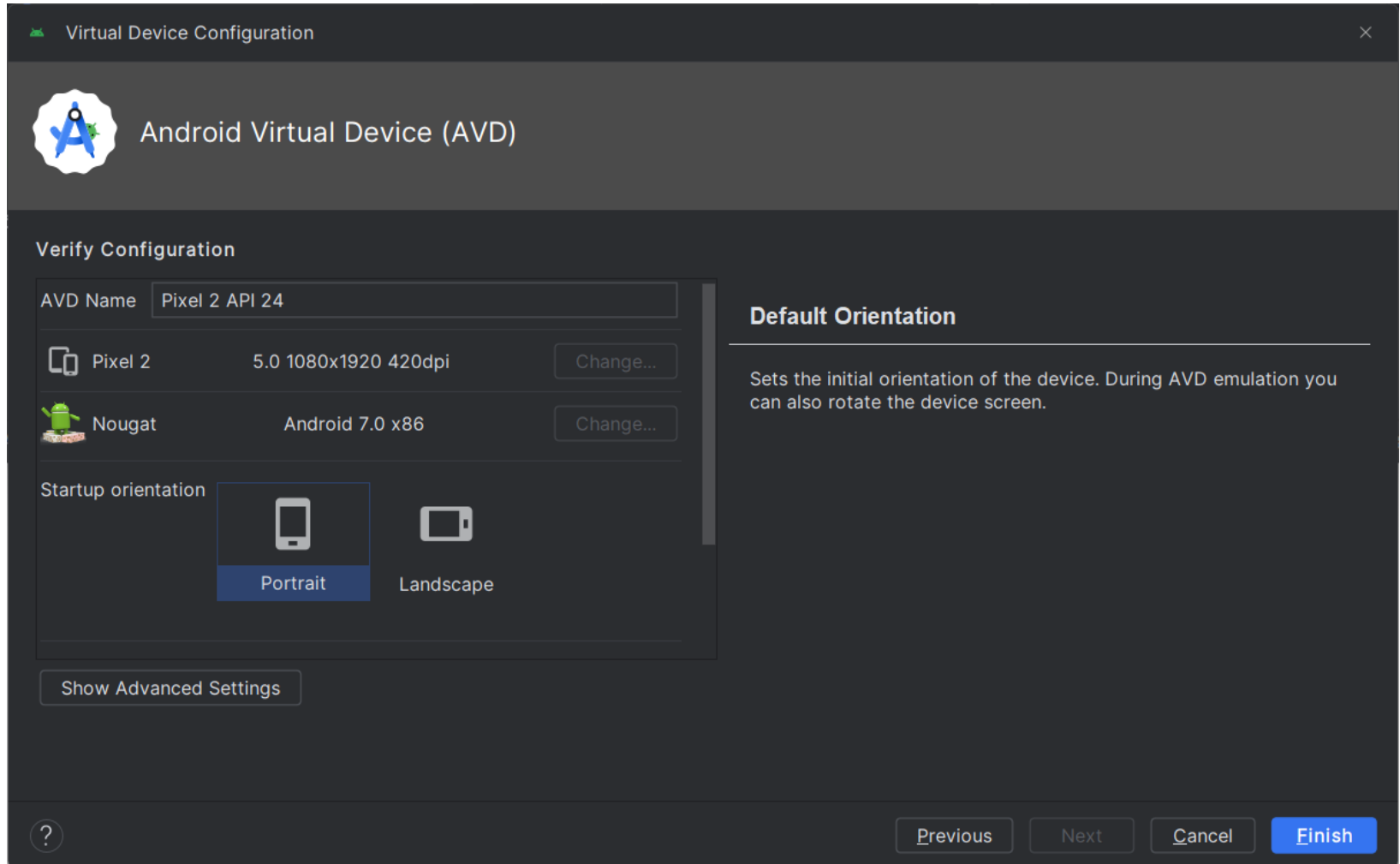
System Image
x86

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

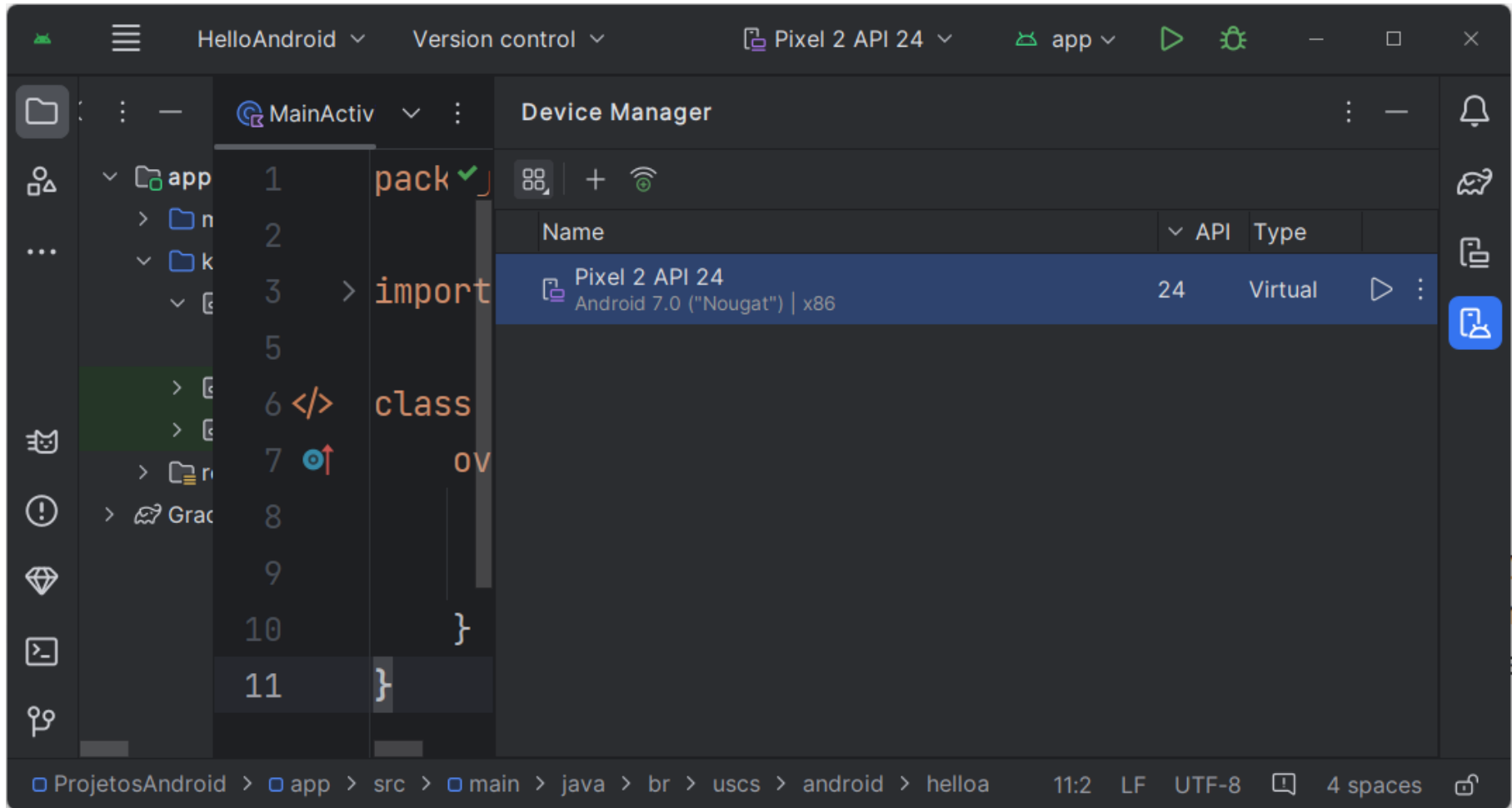
Previous Next Cancel Finish

AVD Manager

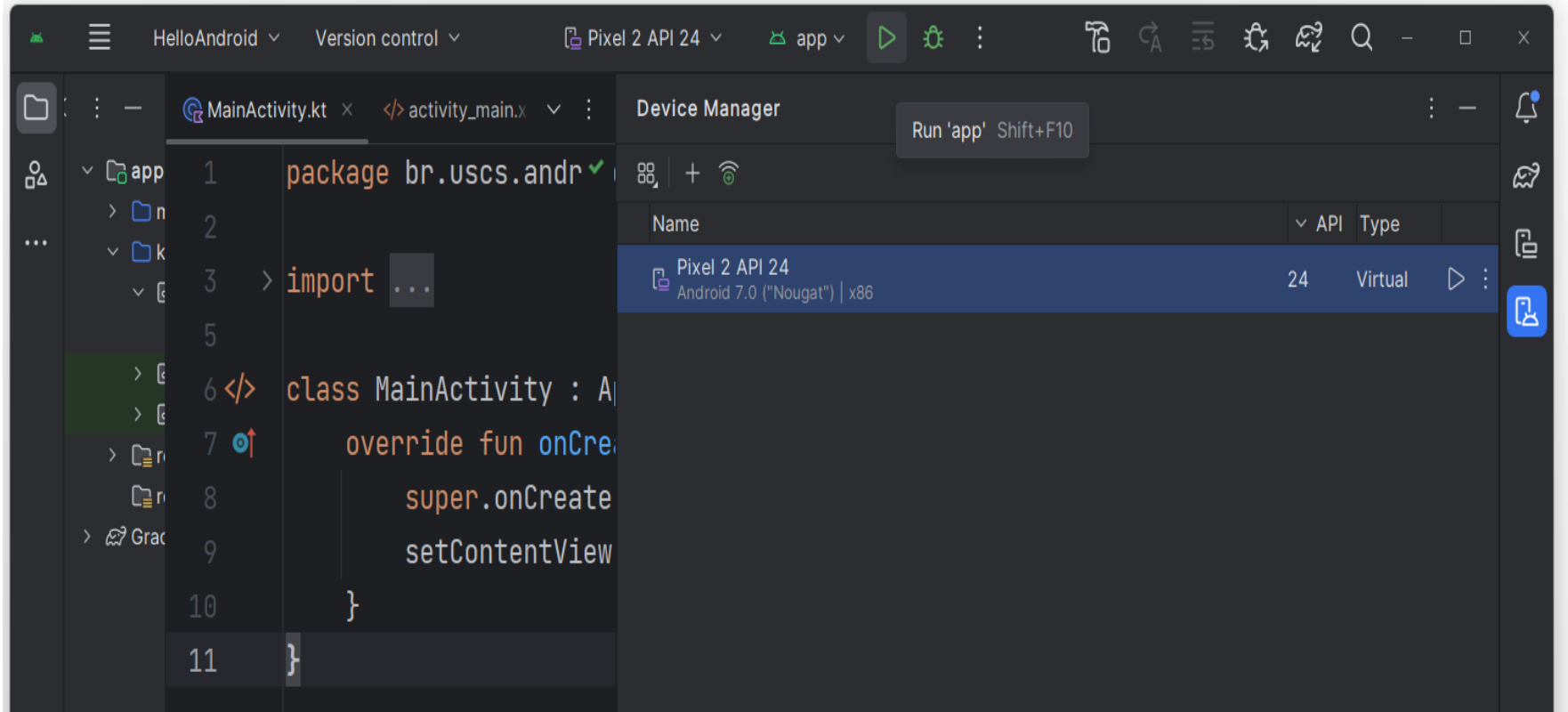
- ✚ A janela final exibe um **resumo** das características do device;
- ✚ Tecle em **Finish**.



AVD Manager



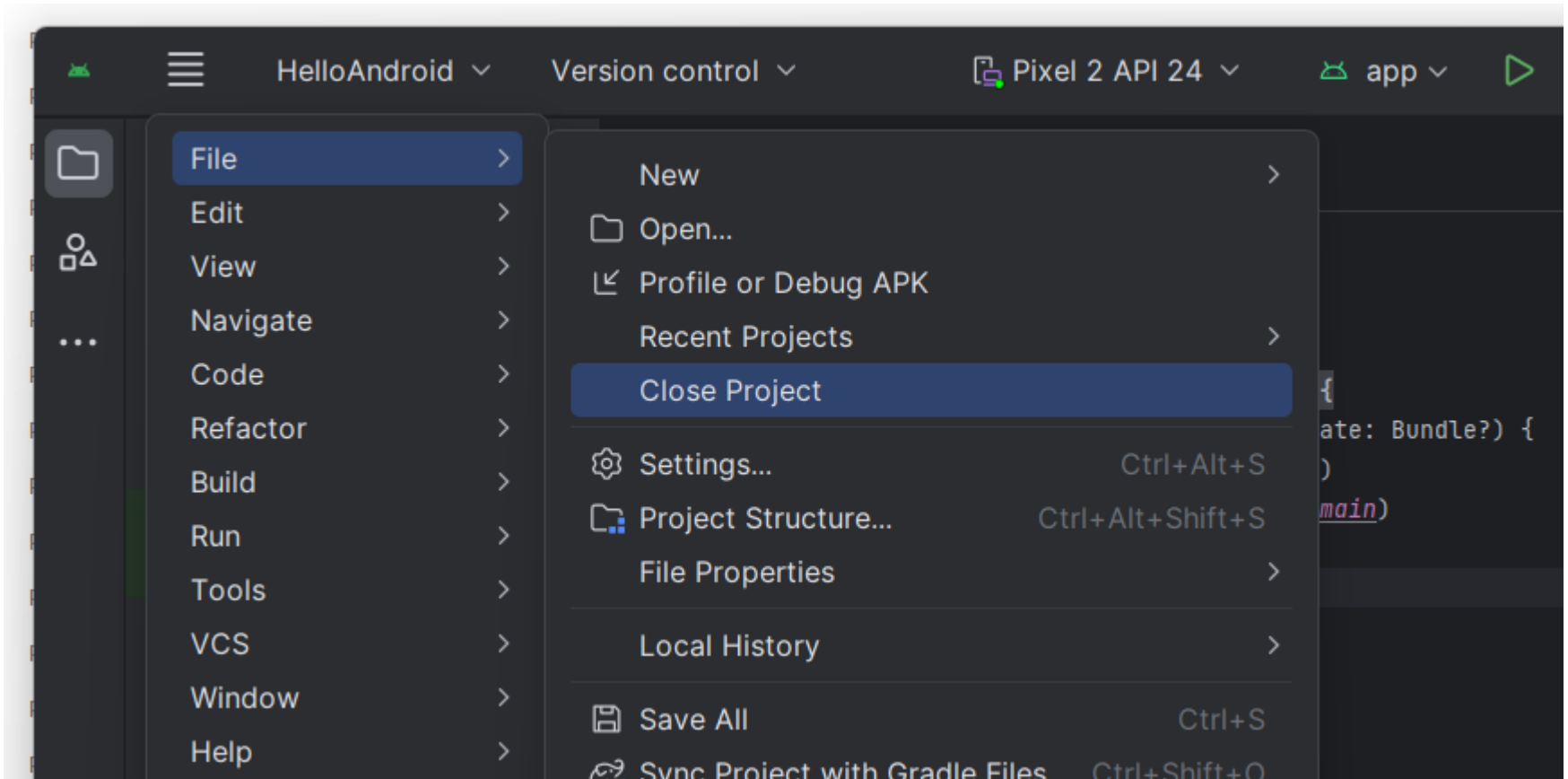
Executando a Aplicação sob AVD



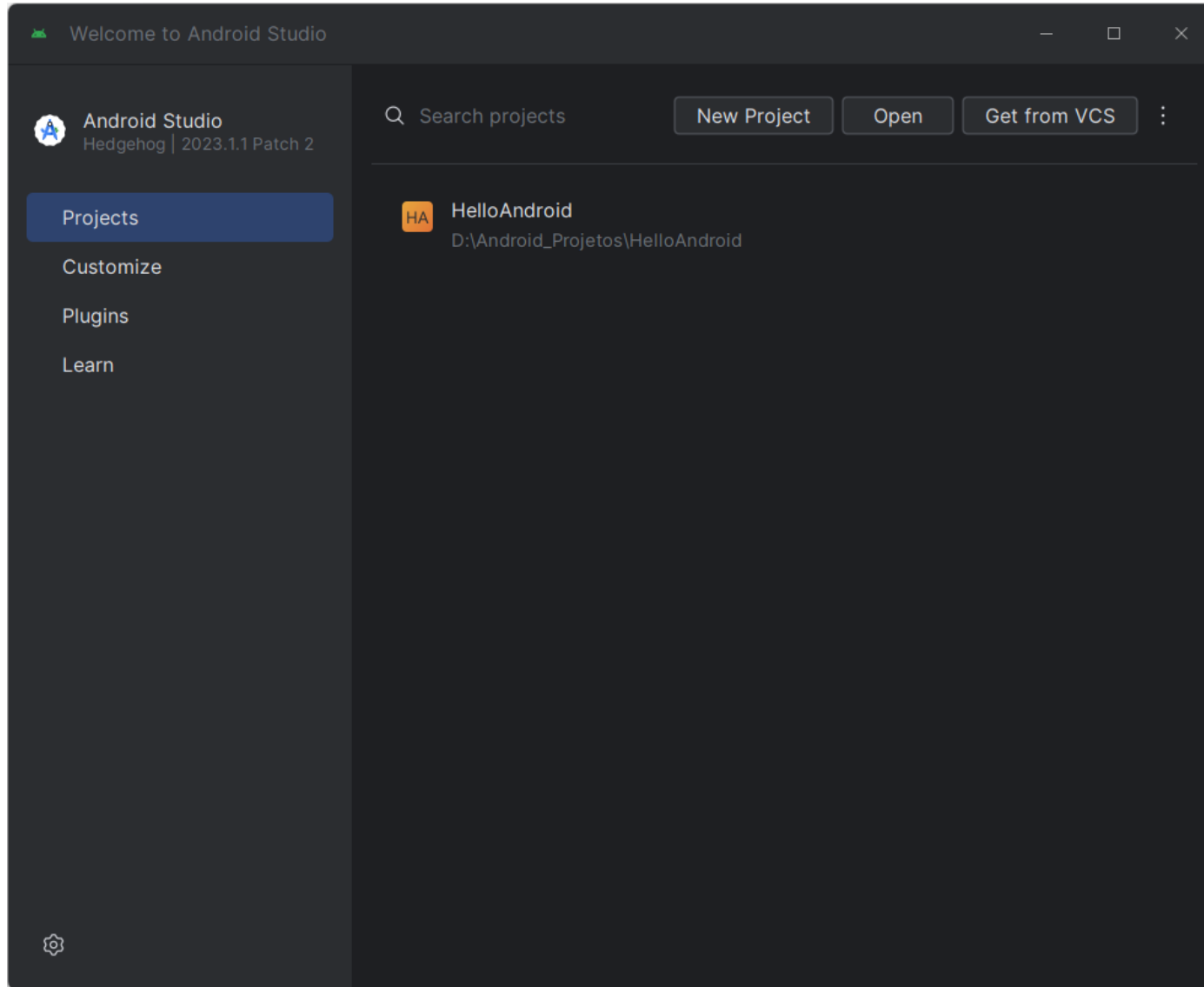
Executando a Aplicação sob AVD



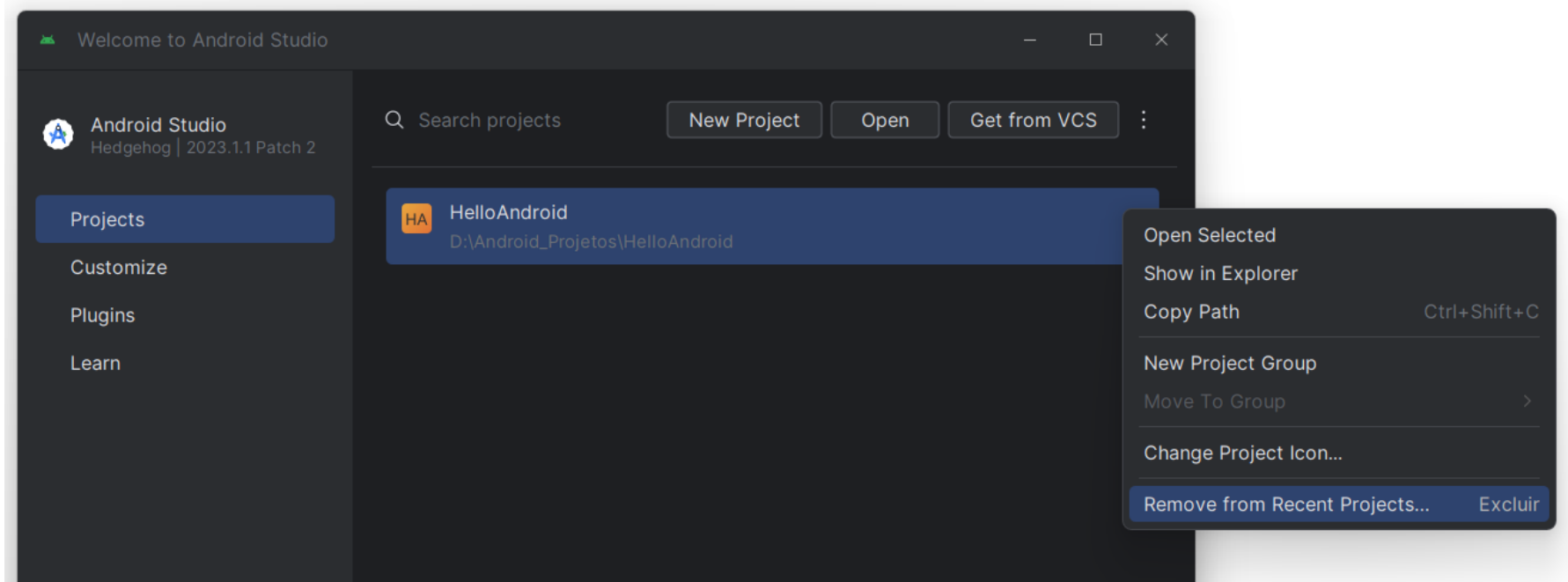
Fechando o Projeto



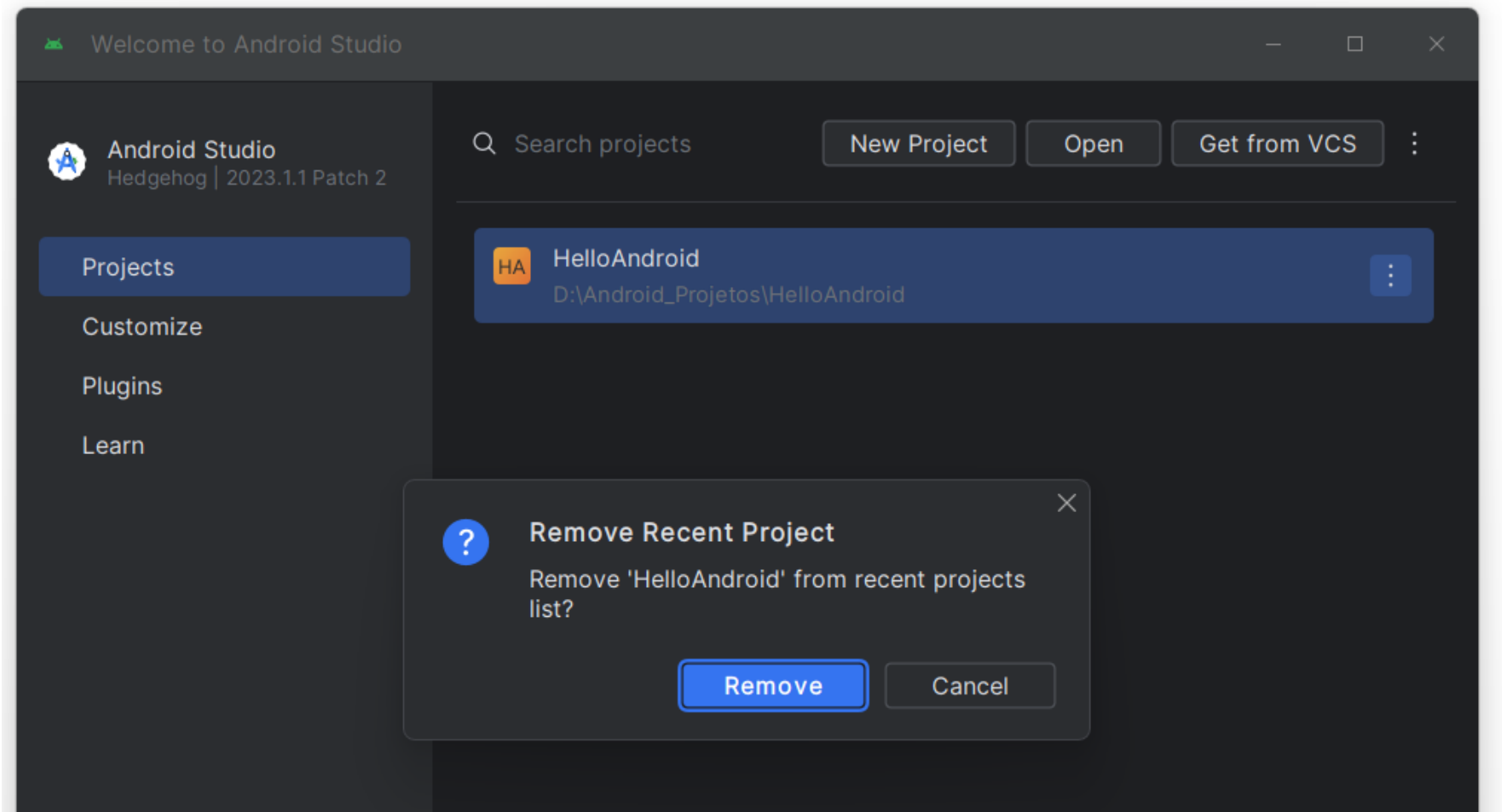
Fechando o Projeto



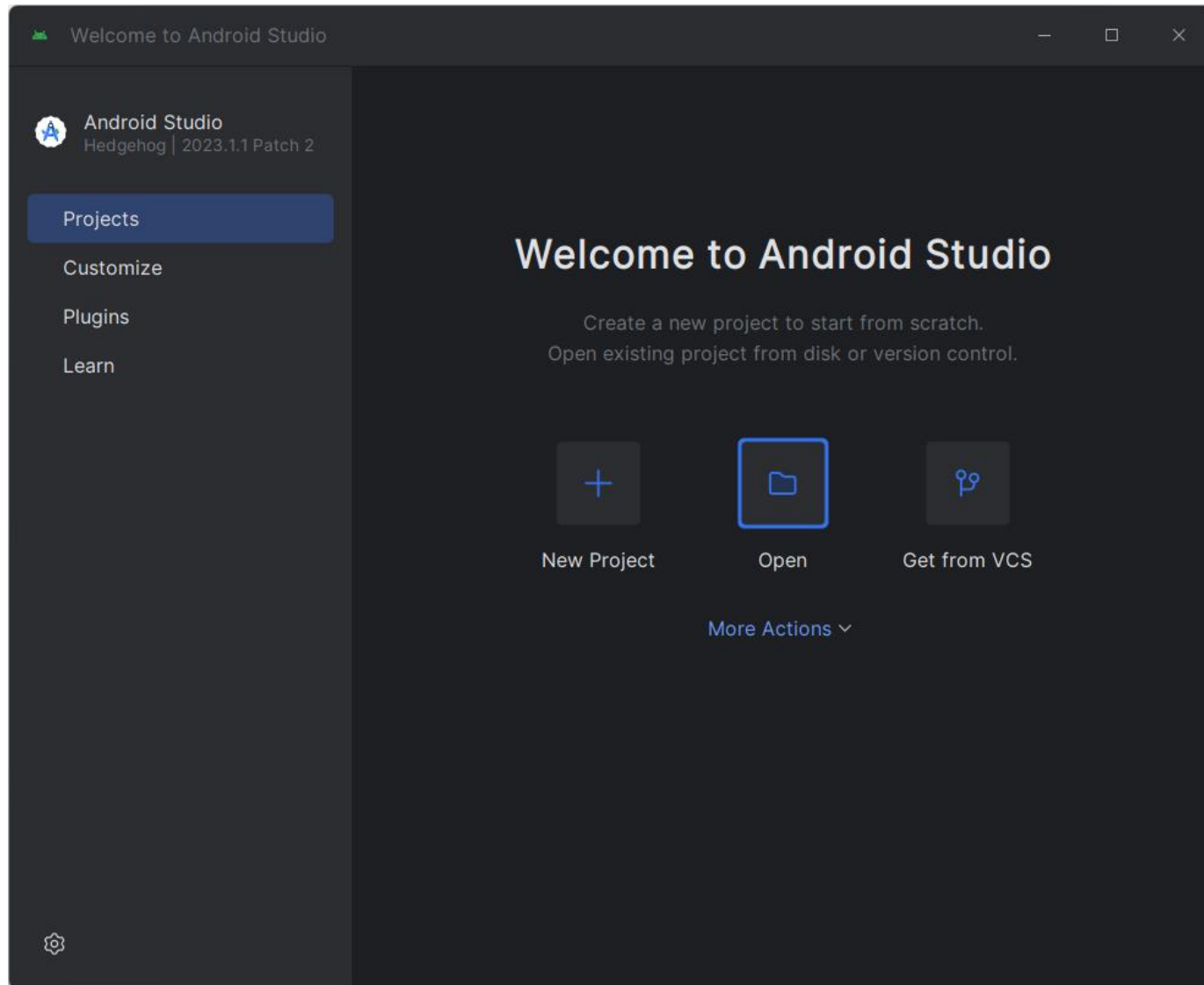
Retirando da Lista de Projetos Recentes



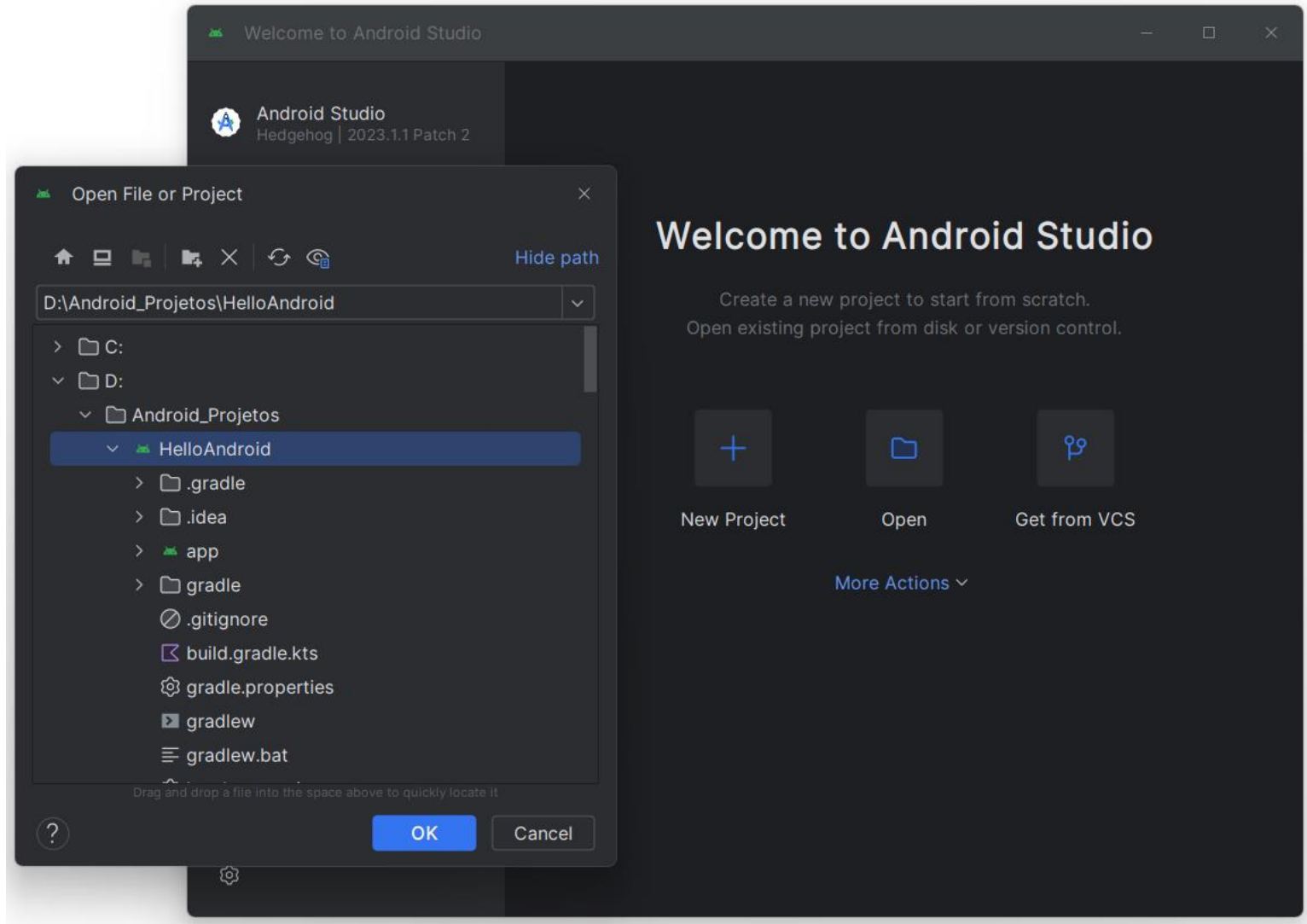
Retirando da Lista de Projetos Recentes



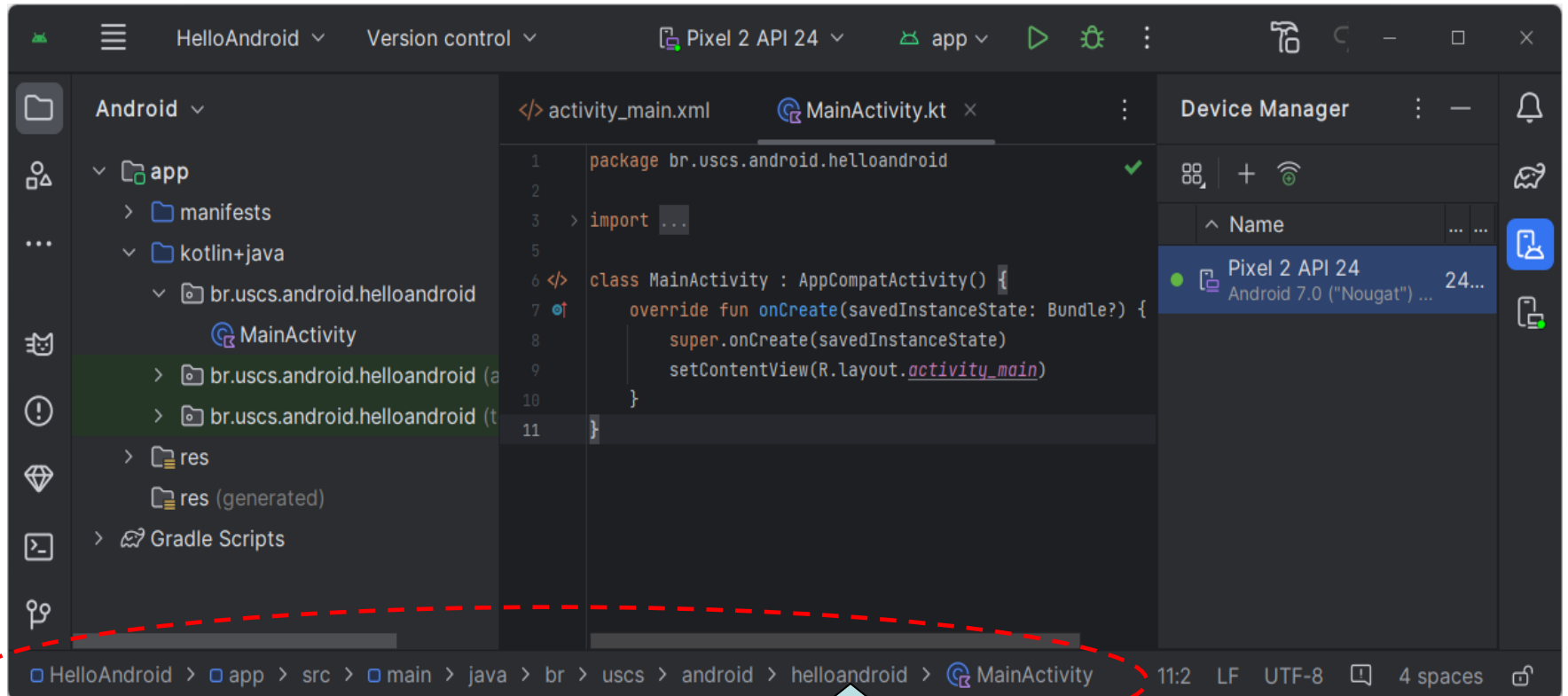
Retirando da Lista de Projetos Recentes



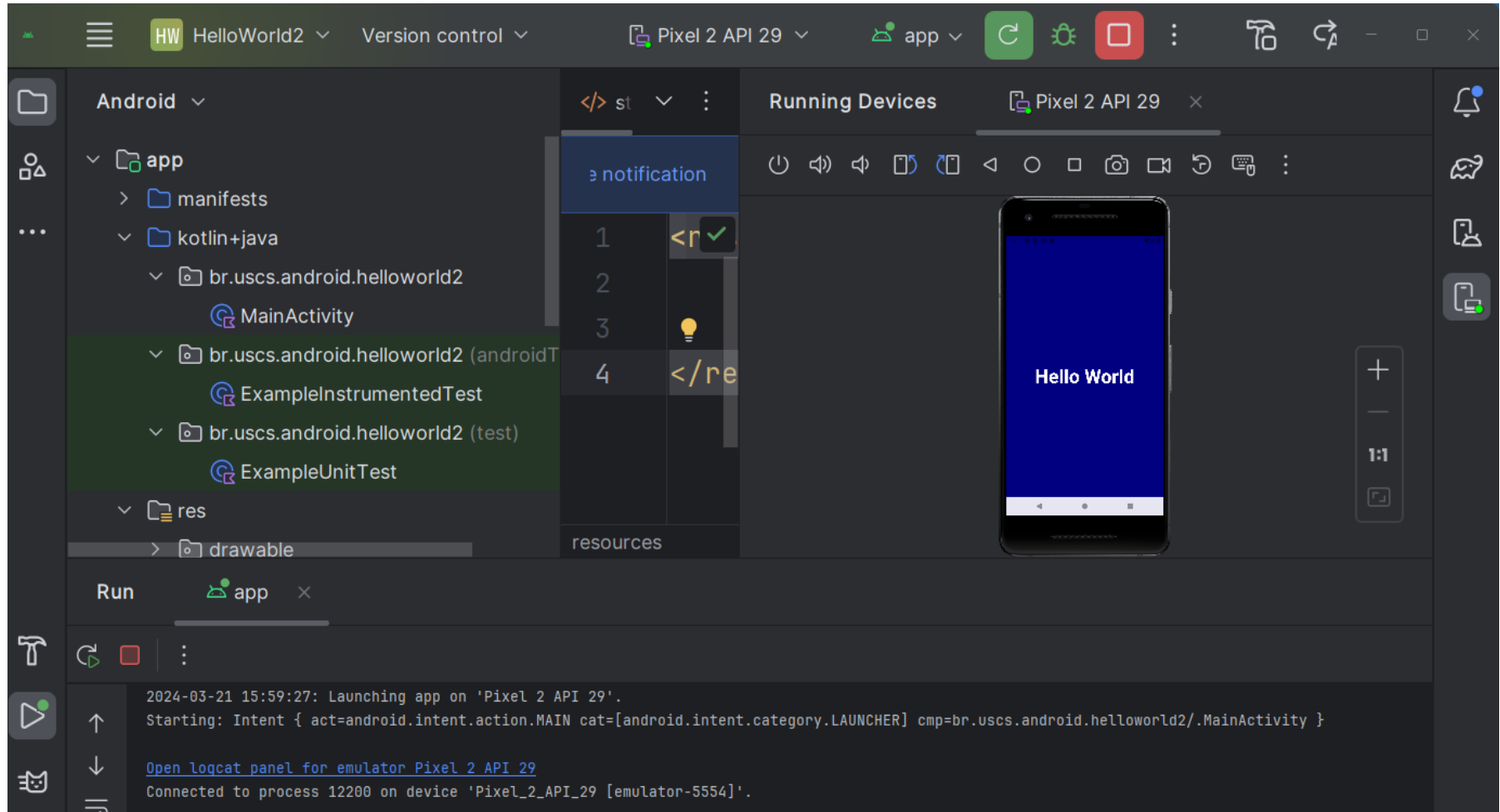
Reabrindo o Projeto



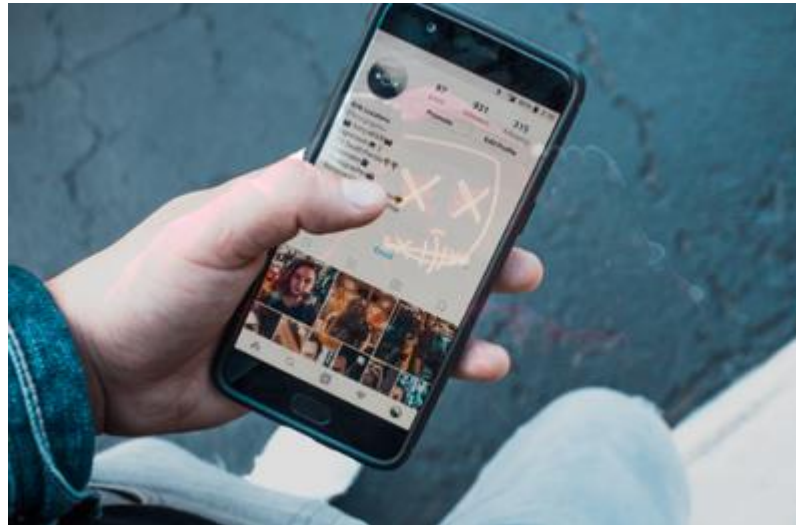
Reabrindo o Projeto



Reprocessando o Projeto

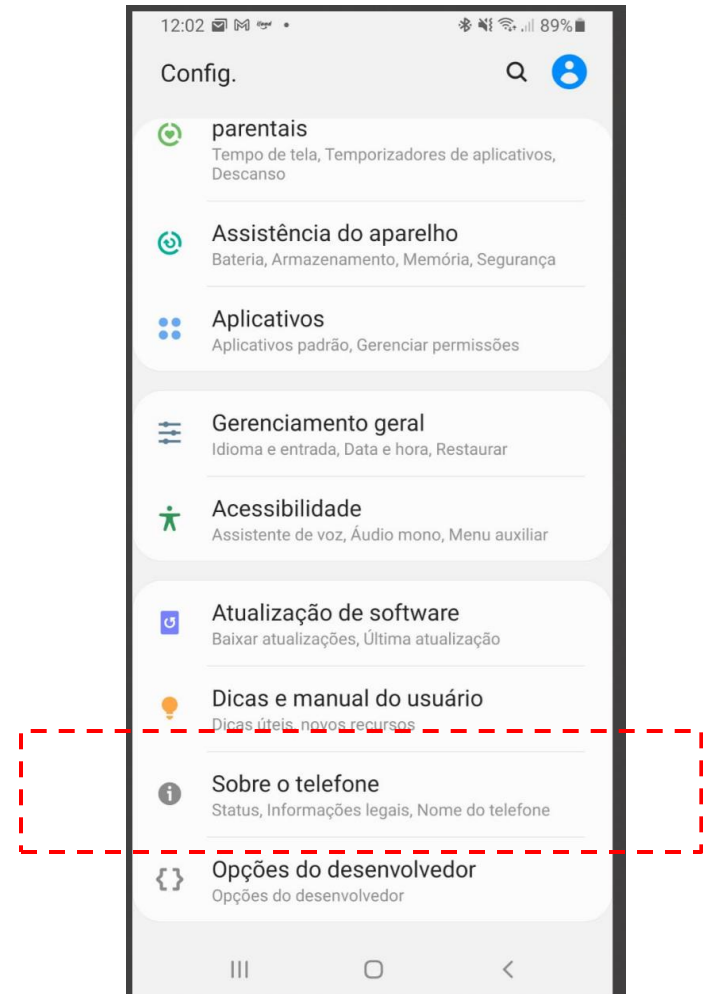
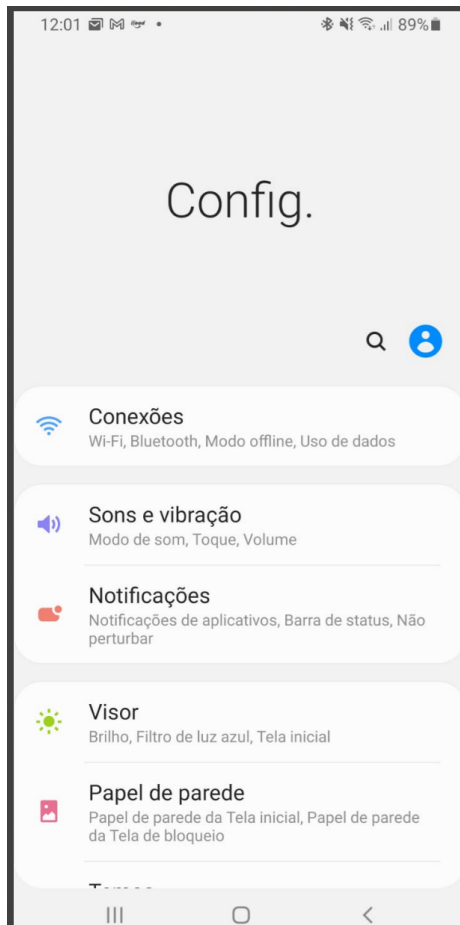


Executando aplicativo num device Físico



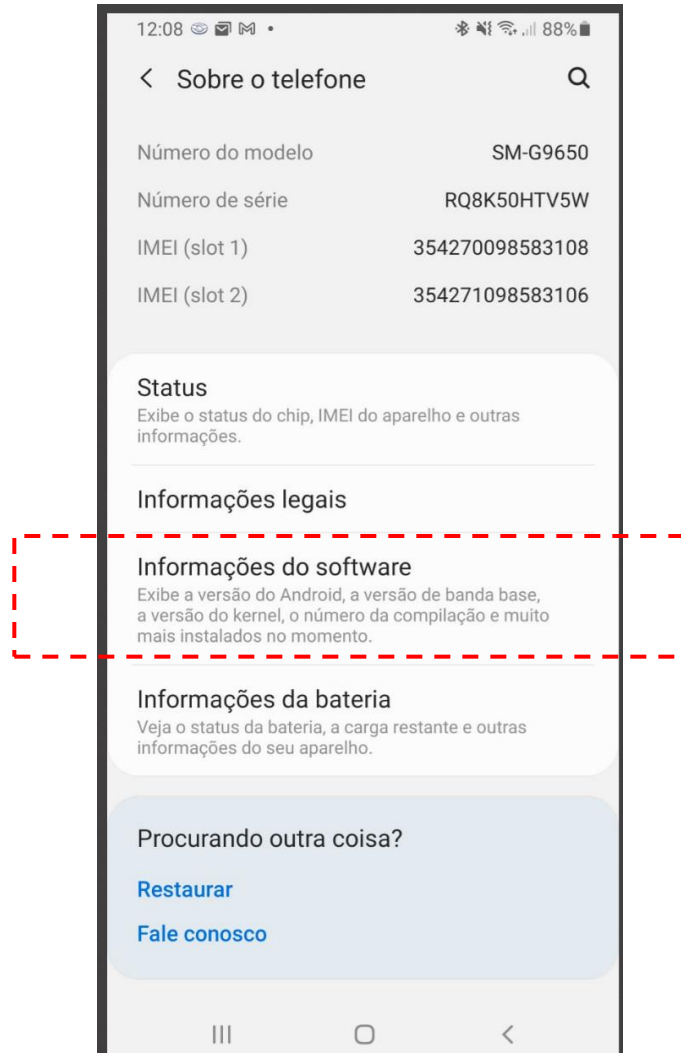
Executando a Aplicação num dispositivo Android

- ✚ Em seu **dispositivo**, entre na opção **Settings** (Configurações) e role a tela até **About Phone** (Sobre o telefone);



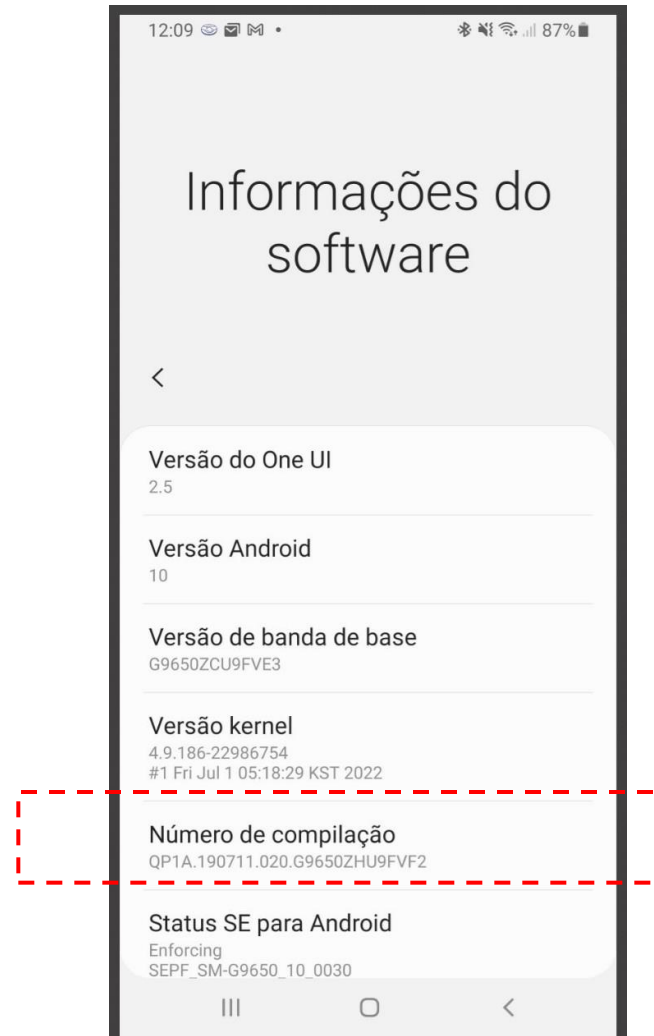
Executando a Aplicação num dispositivo Android

☳ Role a tela até o final e entre na opção “**Informações do Software**”:



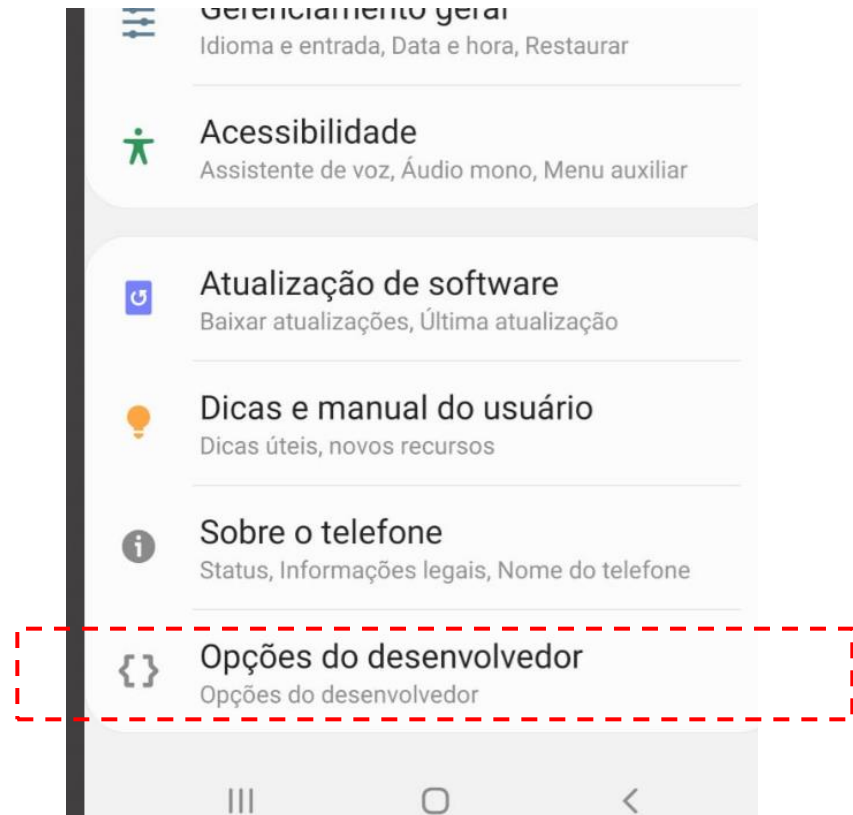
Executando a Aplicação num dispositivo Android

- ✦ Agora, a **parte mágica**: Role a tela até o final e verifique o **Build Number** (Número da Compilação):



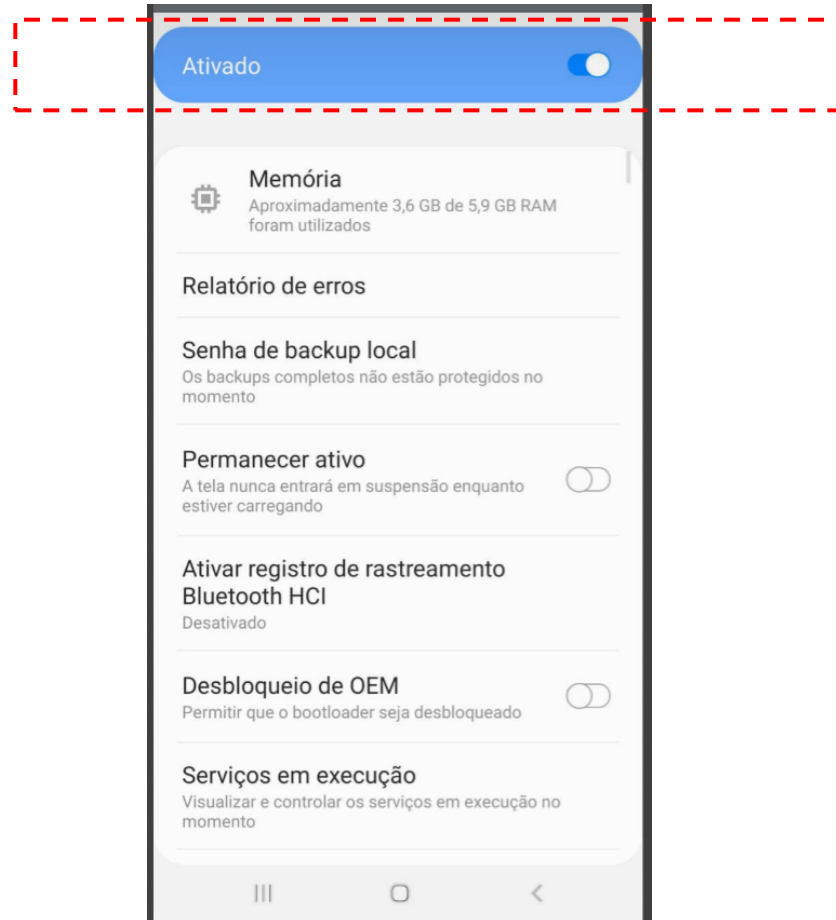
Executando a Aplicação num dispositivo Android

- ✚ Quando você ver o item Número da Compilação, toque nele várias vezes até que uma mensagem apareça informando que você está a alguns passos de se tornar um **desenvolvedor**;
- ✚ Continue tocando até que outra mensagem apareça, avisando que você se tornou um **desenvolvedor**.



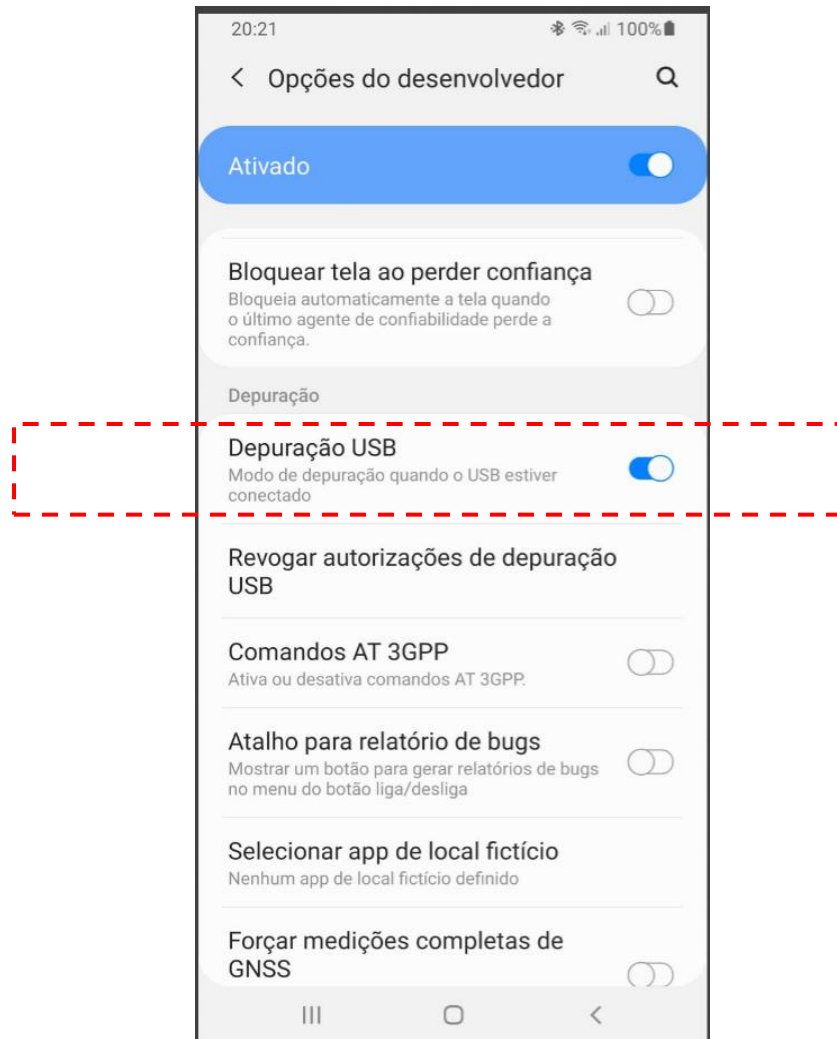
Developer Options

- ✚ Um novo item no menu irá aparecer que corresponde ao **Developer Options**;
- ✚ Tecle nesta opção para revisar todas os recursos disponíveis.



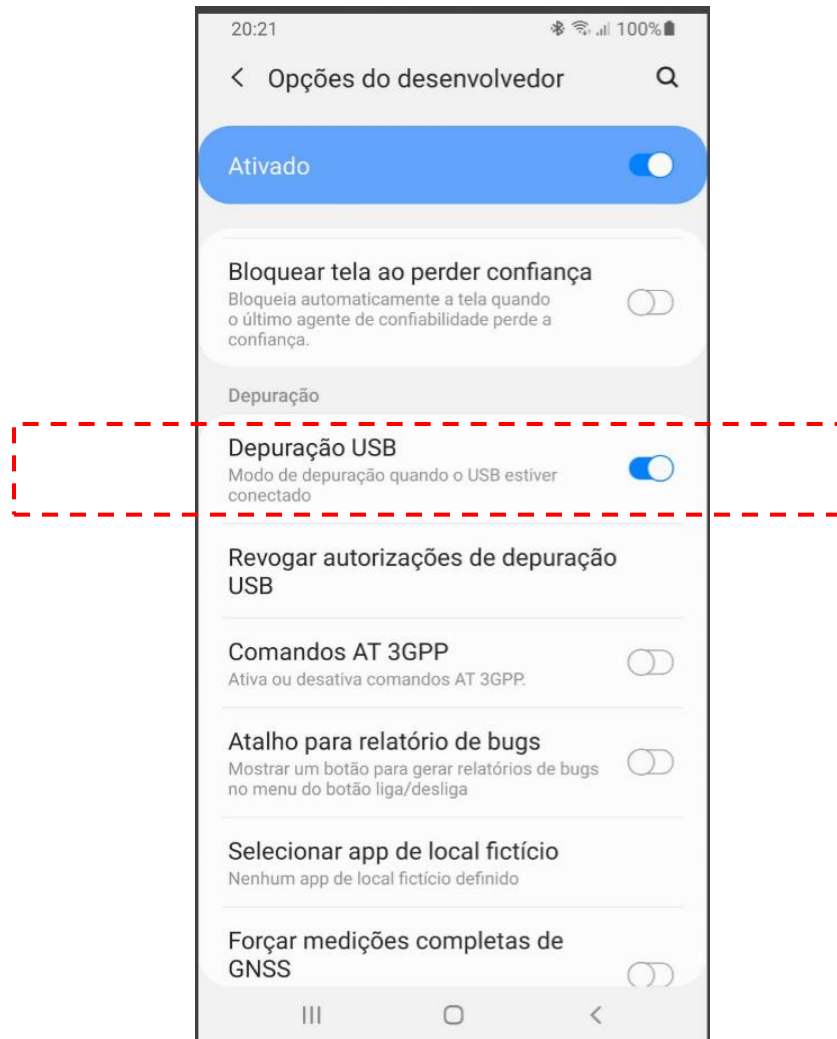
Developer Options

- ⊕ Há uma série de opções disponíveis. A única opção que iremos utilizar agora é **USB Debugging** (Depuração **USB**).



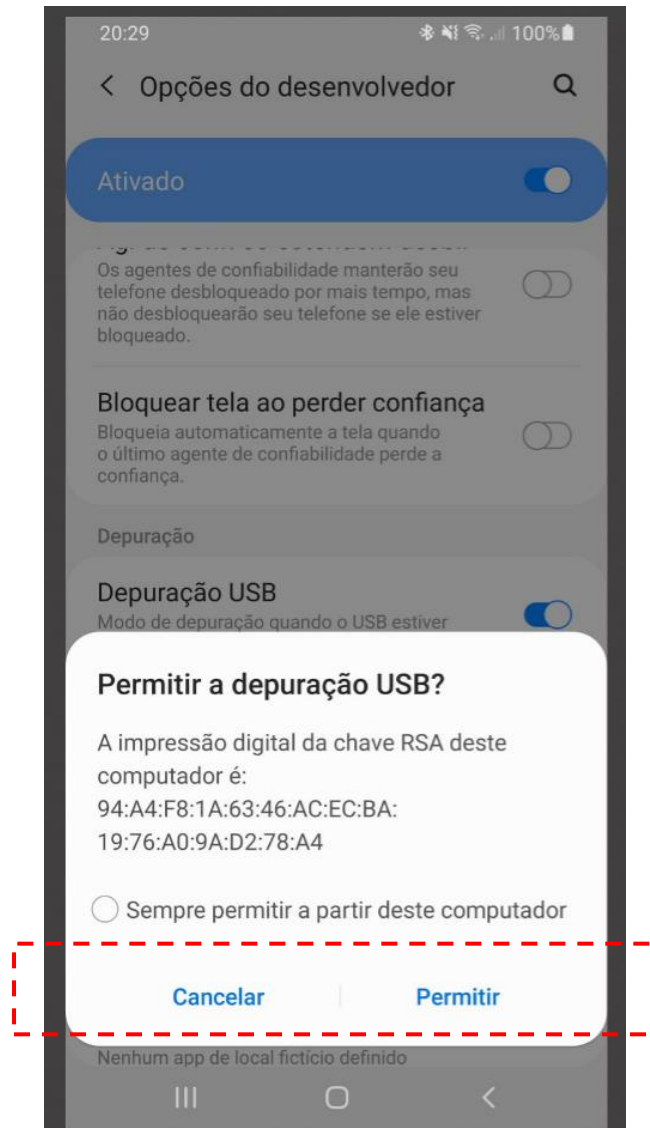
Rodando o aplicativo

- ✚ Há uma série de opções disponíveis. A única opção que iremos utilizar agora é **USB Debugging** (Depuração **USB**).



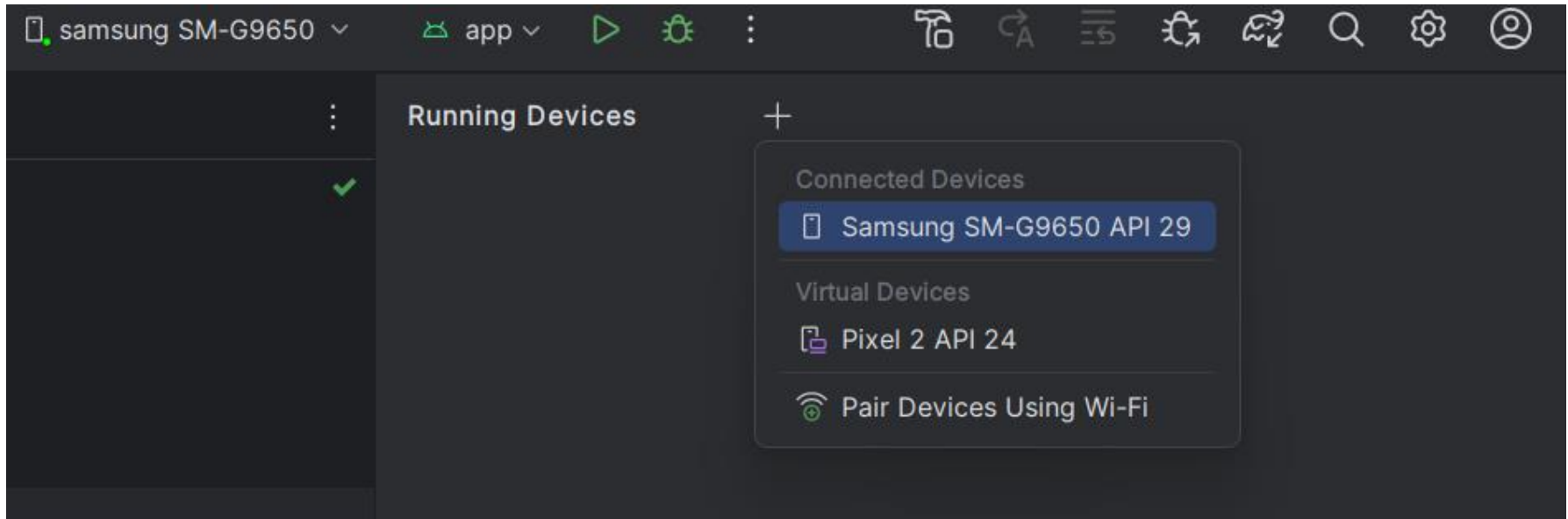
Rodando o aplicativo

⊕ Responda Permitir depuração USB:



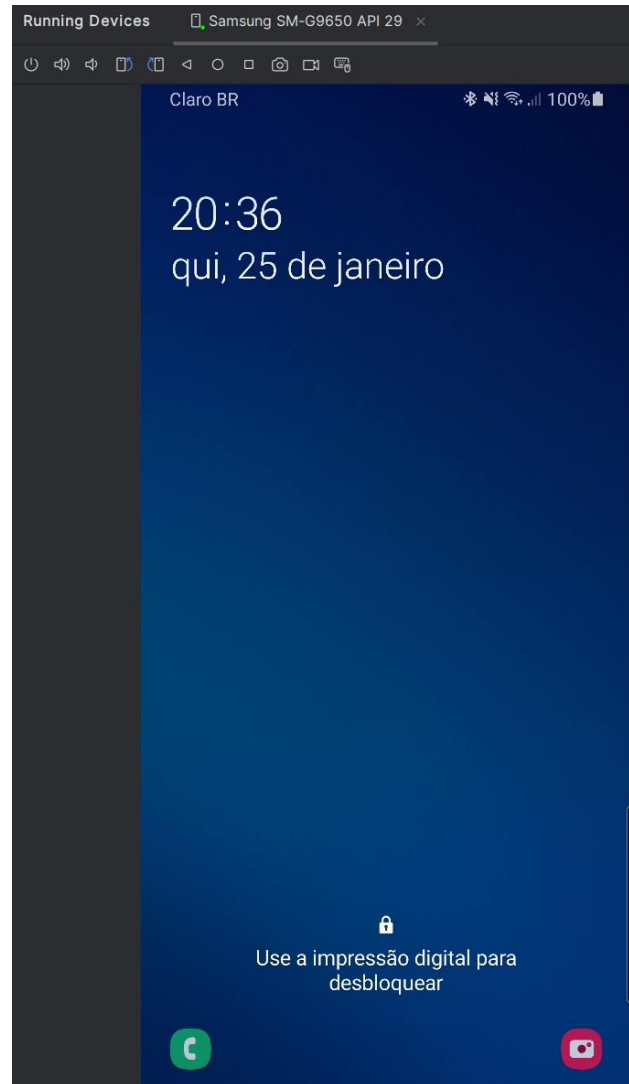
Rodando o aplicativo

- ⊕ Uma vez que o dispositivo está conectado pela porta **USB**, no **Android Studio**, selecione o device do celular conectado:



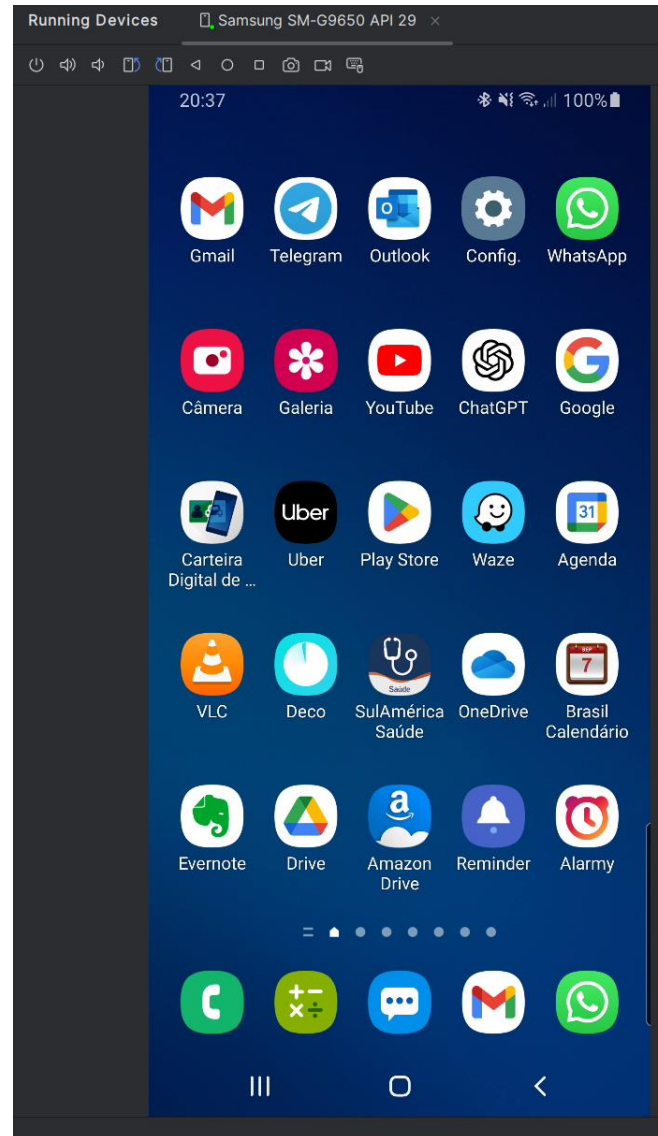
Rodando o aplicativo

✚ **Android Studio**, exibirá a imagem do dispositivo físico:



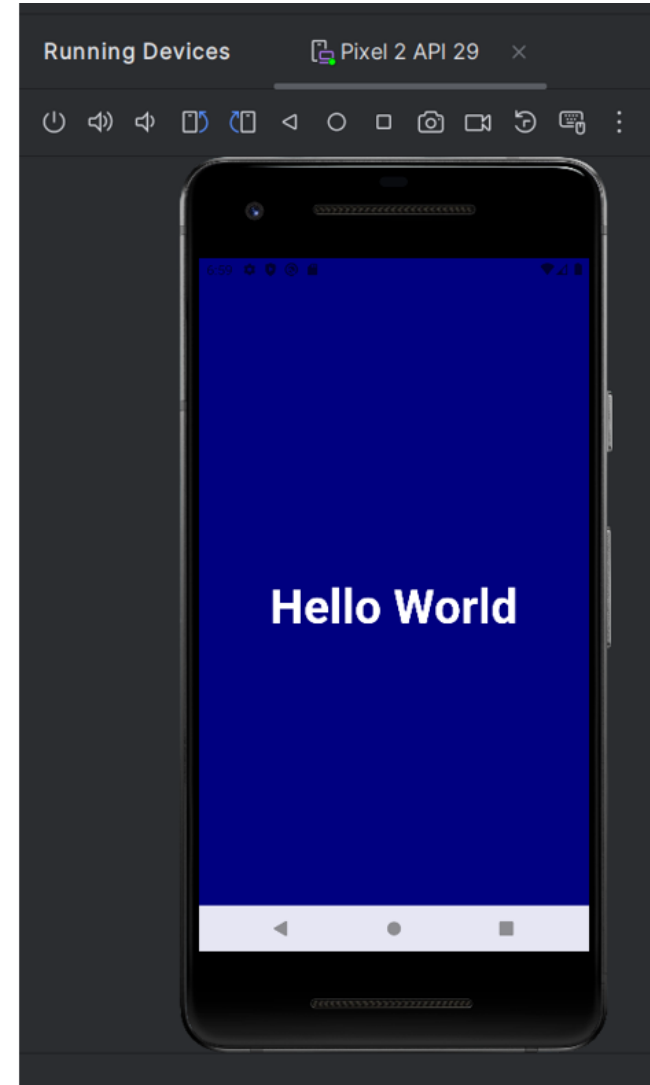
Rodando o aplicativo

✦ **Android Studio**, exibirá a imagem do dispositivo físico:



Rodando o aplicativo

- ✚ No canto superior esquerdo, tecele no botão **verde** de execução:



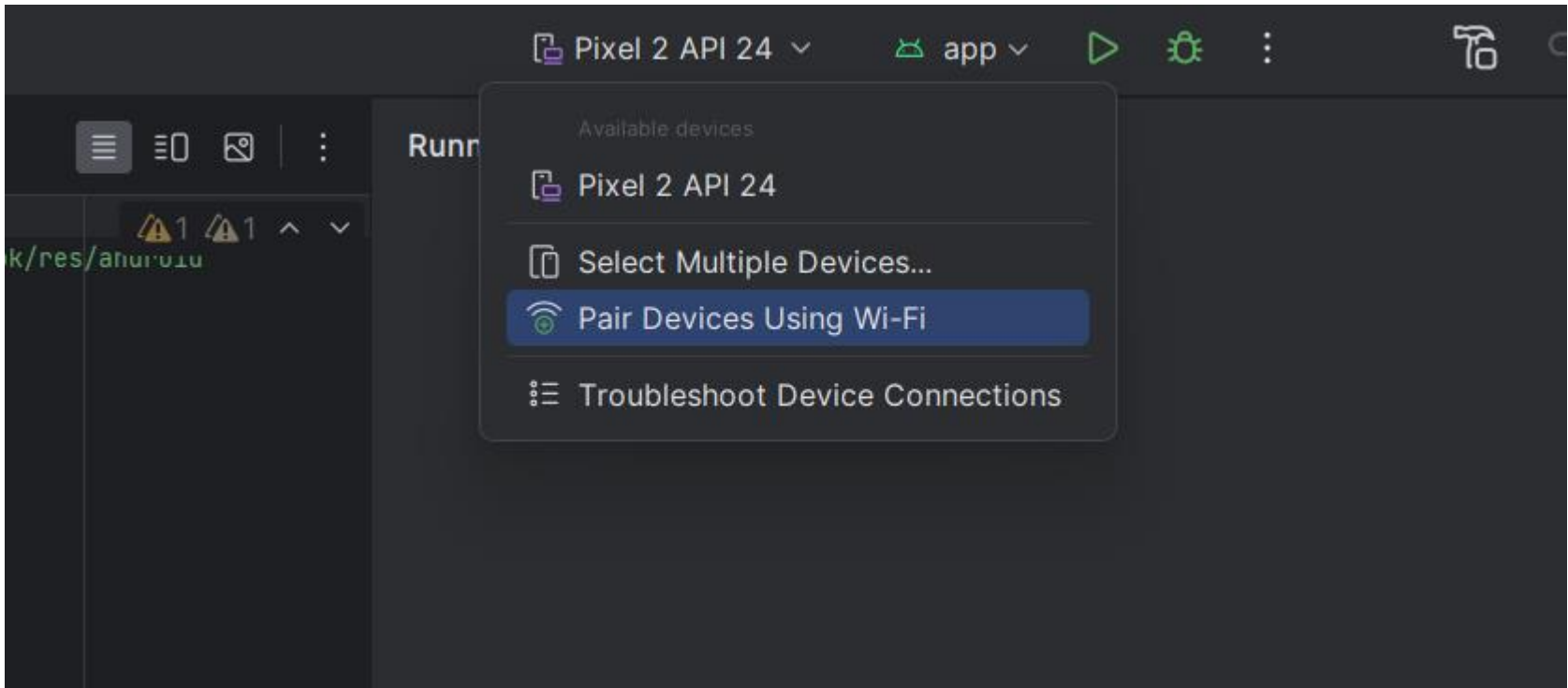
Depuração por Wi-Fi

- ✦ É possível executar uma aplicação **Android** em um dispositivo físico conectado via **Wi-Fi**;
- ✦ Esta funcionalidade é particularmente útil para desenvolvedores que desejam testar suas aplicações diretamente em um dispositivo real **sem** a necessidade de uma conexão **USB**;
- ✦ A funcionalidade exige que o dispositivo esteja instalado com a versão **Android 11**, no mínimo.



Depuração por Wi-Fi

- ✚ Abra o projeto e selecione no menu a opção: **Pair Devices Using Wi-Fi**



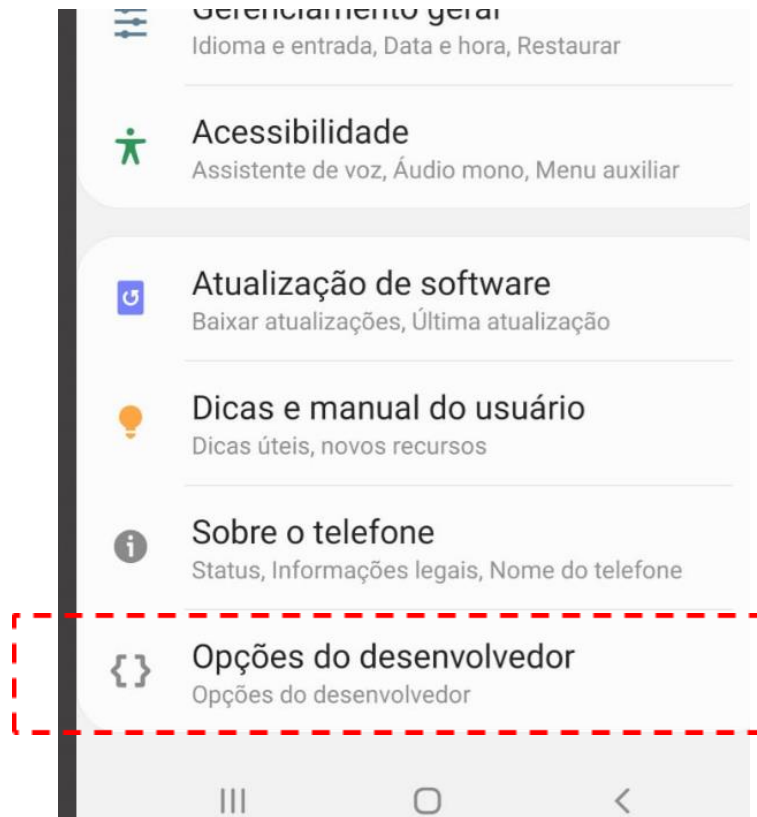
Depuração por Wi-Fi

- Abra o projeto e selecione no menu a opção: **Pair Devices Using Wi-Fi**



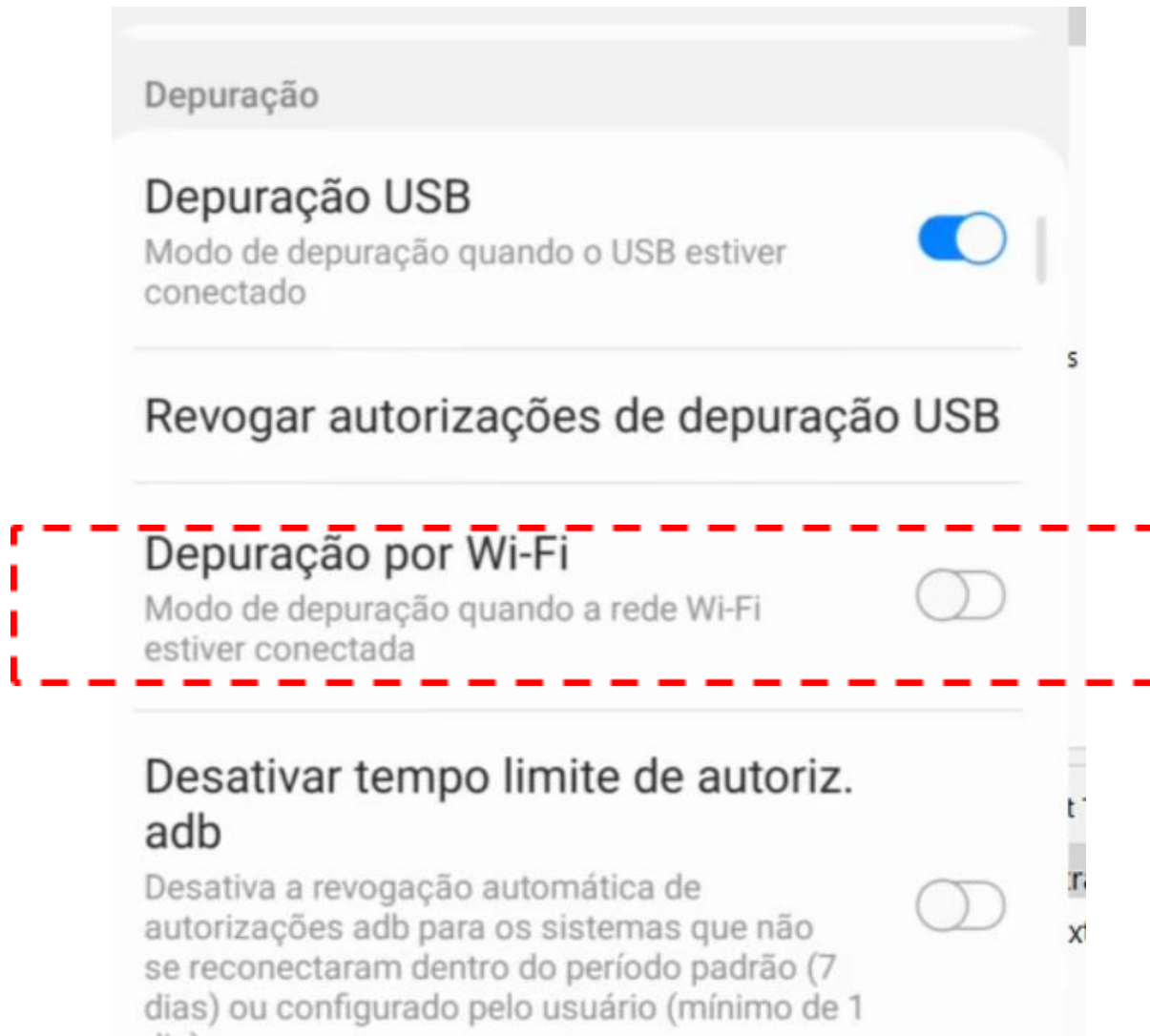
Depuração por Wi-Fi

- ✚ O celular precisa estar na mesma rede do **Android Studio**;
- ✚ Acessar no celular “**Opções do Desenvolvedor**”.



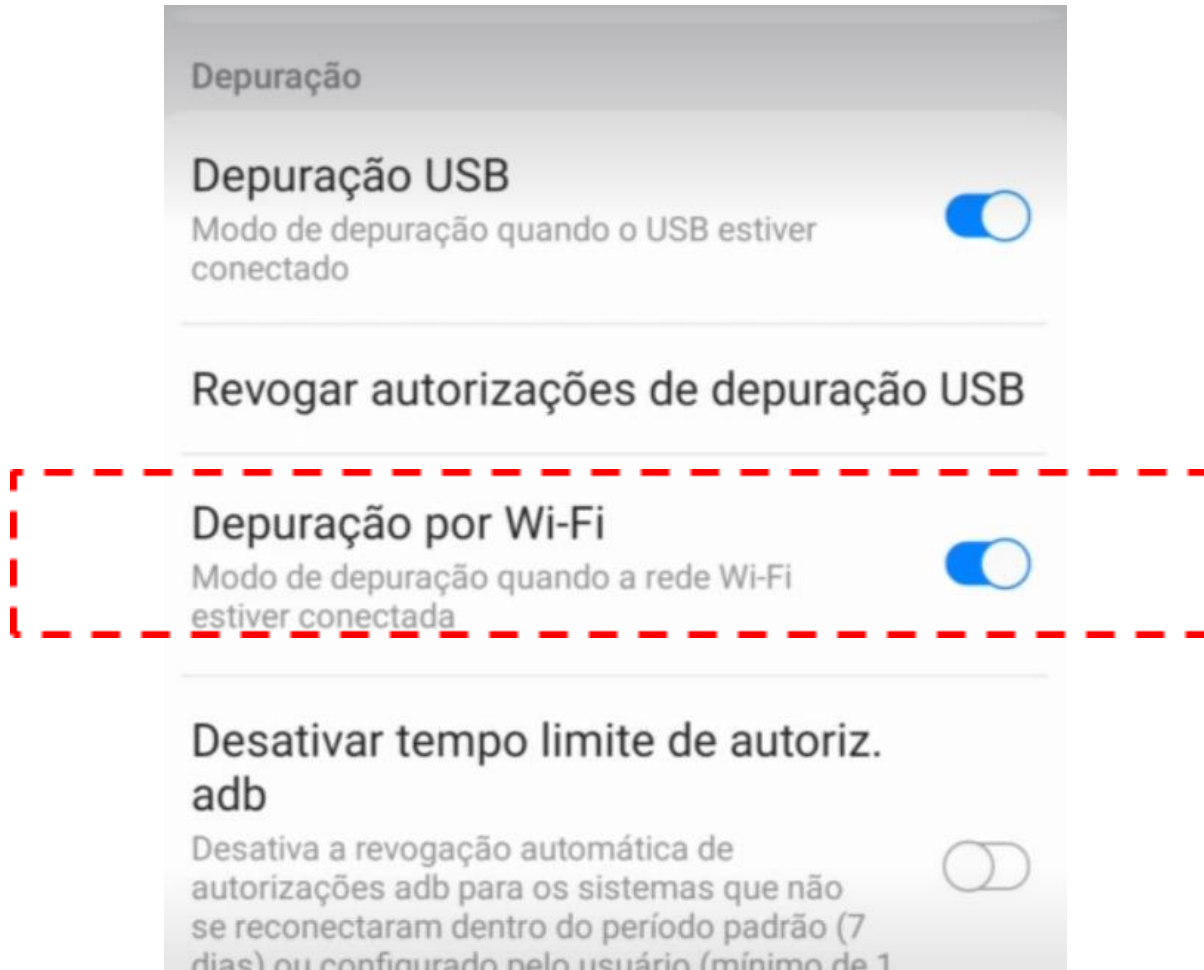
Depuração por Wi-Fi

- ⊕ Acessar no celular a opção “**Depuração por Wi-Fi**”.



Depuração por Wi-Fi

- ✚ Habilitar a opção “**Depuração por Wi-Fi**”.



Depuração por Wi-Fi

- ✚ Em seguida parear o dispositivo.

Device name

Galaxy A71

Endereço IP e porta

192.168.1.6:45297



Parear o dispositivo com um código QR

Parear novos dispositivos usando um leitor de código QR

Parear o dispositivo com um código de pareamento

Parear novos dispositivos usando um código de seis dígitos

Depuração por Wi-Fi

📶 Celular conectado!

Pair new devices over Wi-Fi

Pair devices to enable wireless debugging. Pair camera-enabled devices using a QR code. Other devices can be paired using a pairing code. [Learn more](#)

Pair using QR code Pair using pairing code



Depuração por Wi-Fi

⊕ Dispositivo exibido no **Android Studio** !

