

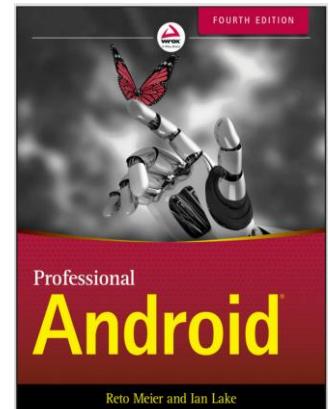
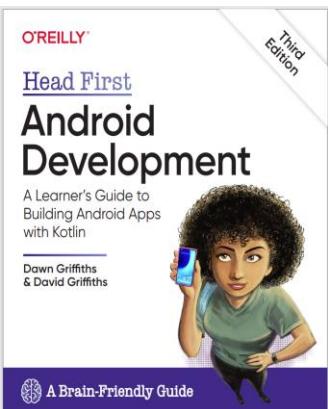
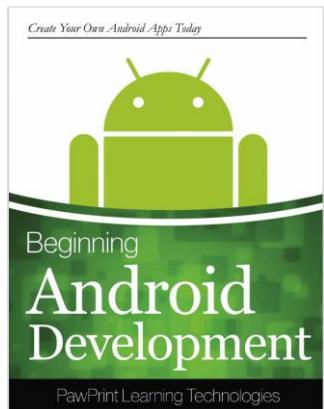
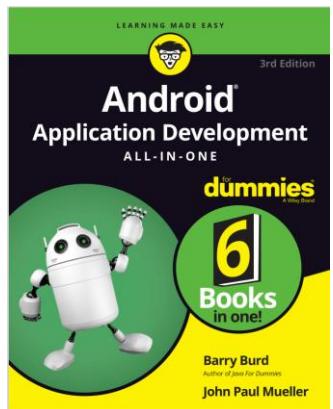
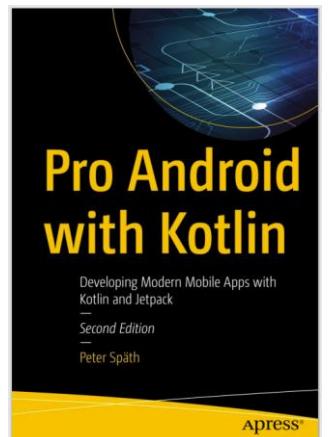
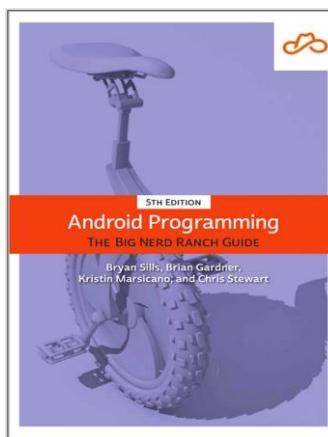
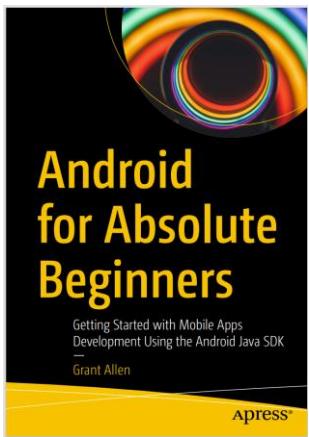
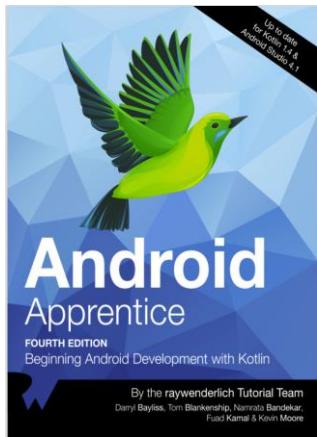
Desenvolvimento de Aplicativos

Unidade 2 – Primeira aplicação Android



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

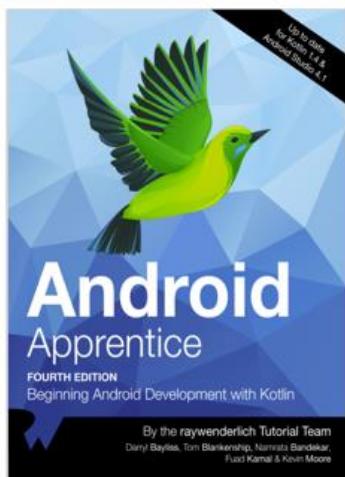
Bibliografia



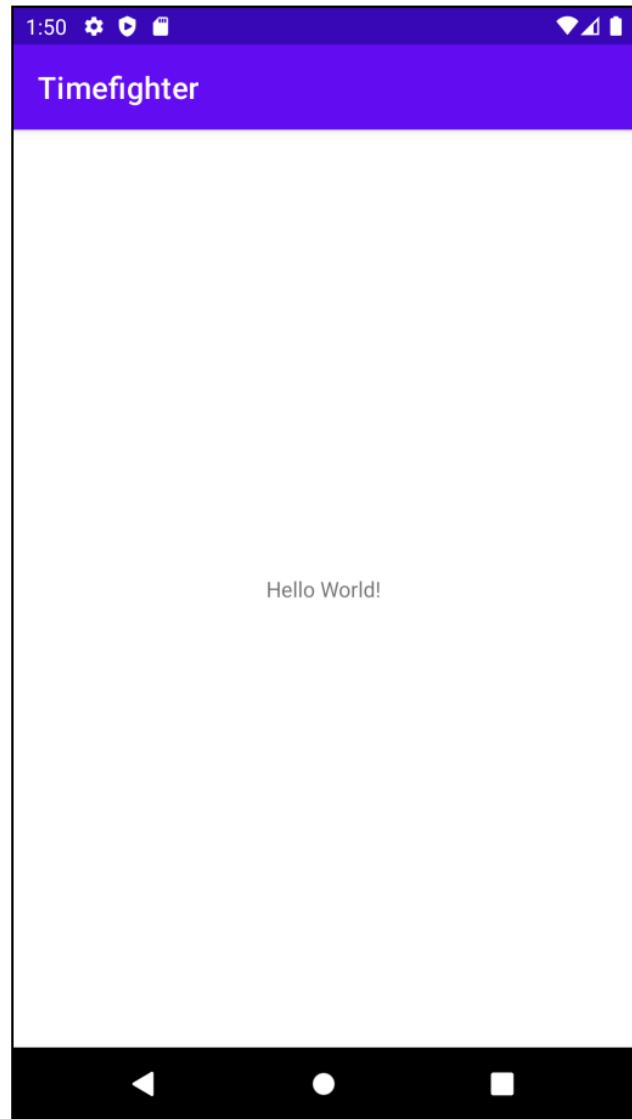
Primeiro Projeto Android

- ⊕ Agora o **Android Studio** está instalado, é hora de criar seu primeiro projeto;
- ⊕ É importante notar que esta unidade foca em fazer seu aplicativo funcionar o mais rápido possível;
- ⊕ Assim, você encontrará algumas telas que não entenderá; mas não se preocupe, você terá a chance de explorar essas telas em detalhes quando avançarmos nos próximos capítulos.

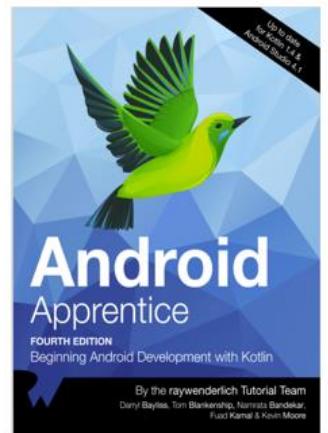
Fonte:



Primeira aplicação Android

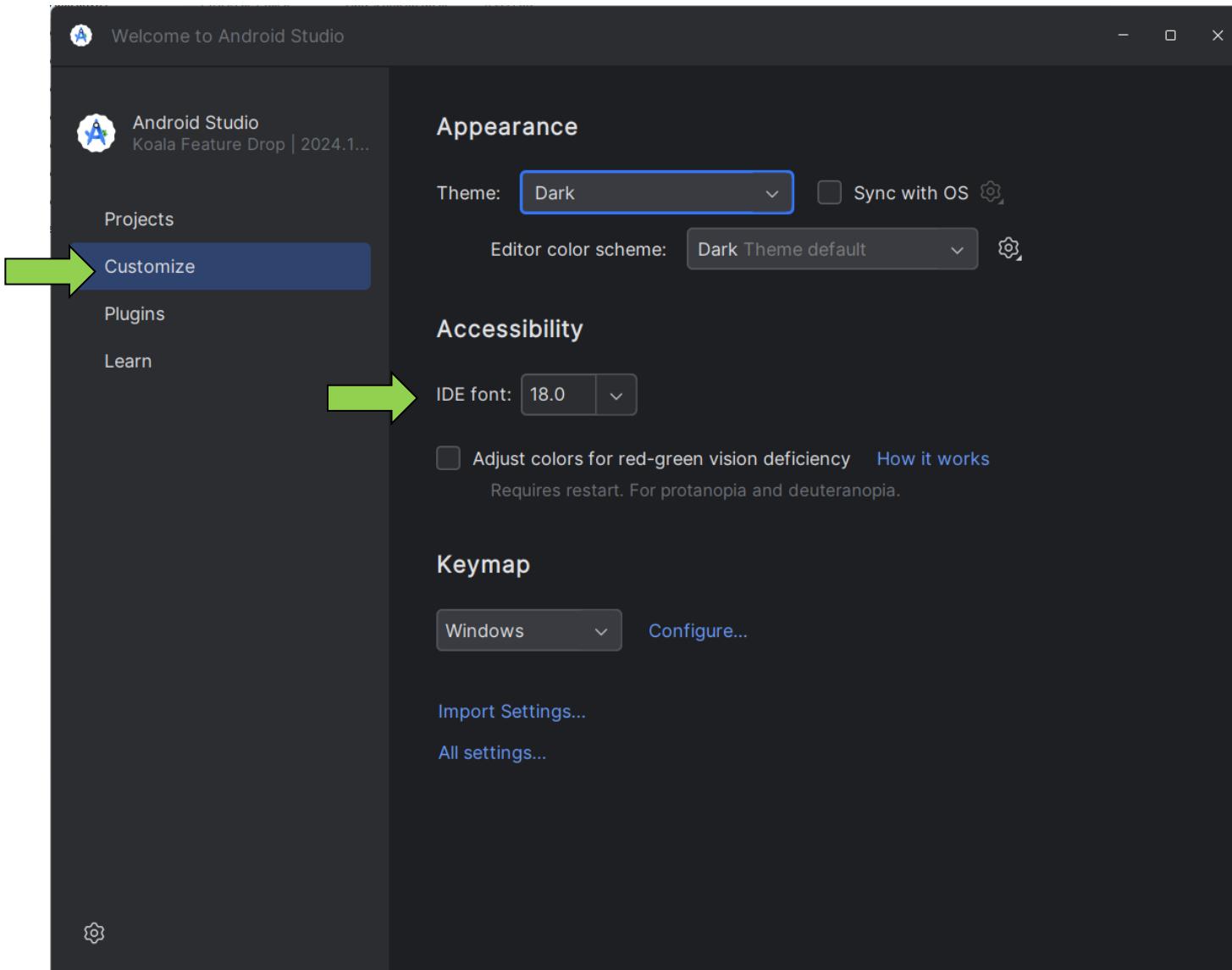


Fonte:



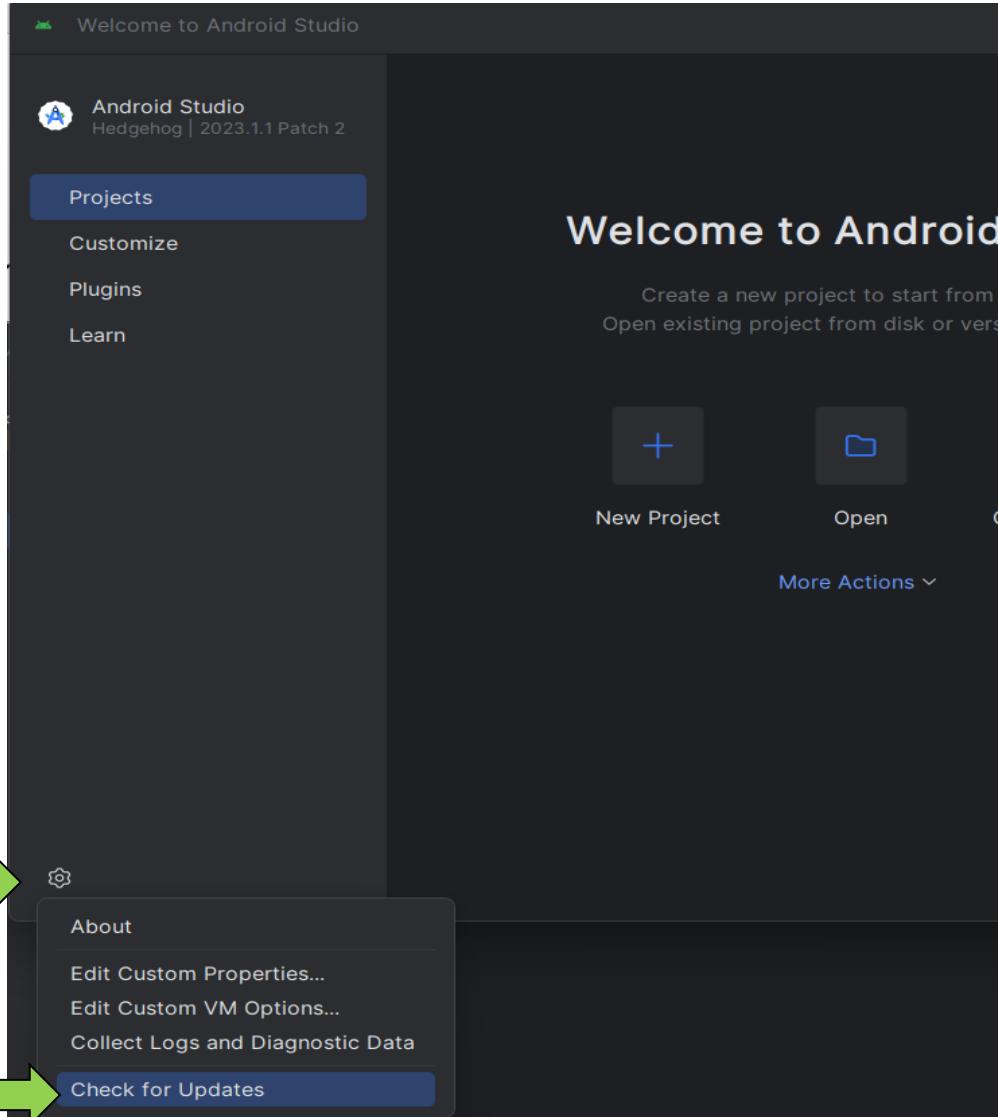
Primeira aplicação Android

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



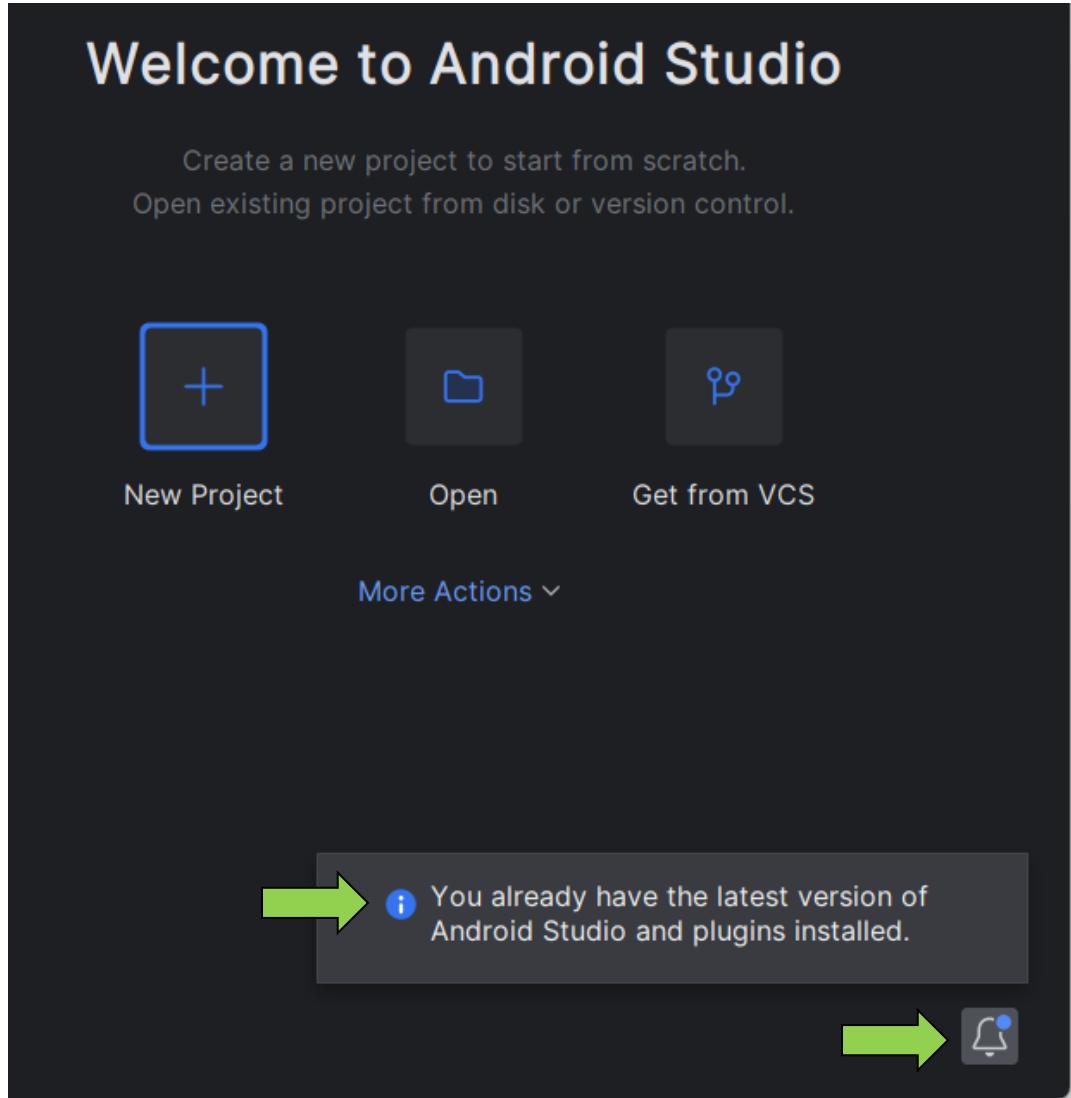
Primeira aplicação Android

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



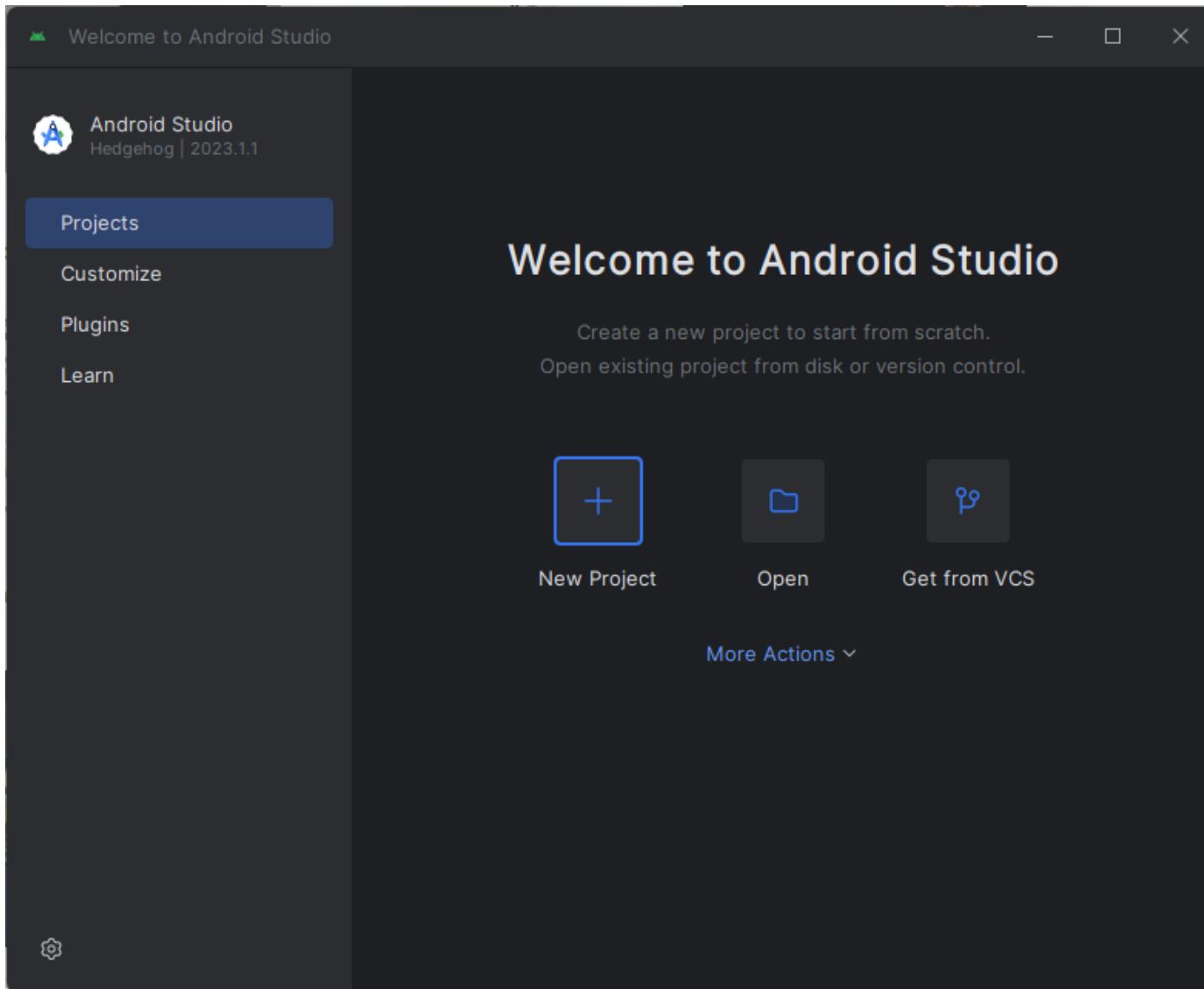
Primeira aplicação Android

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



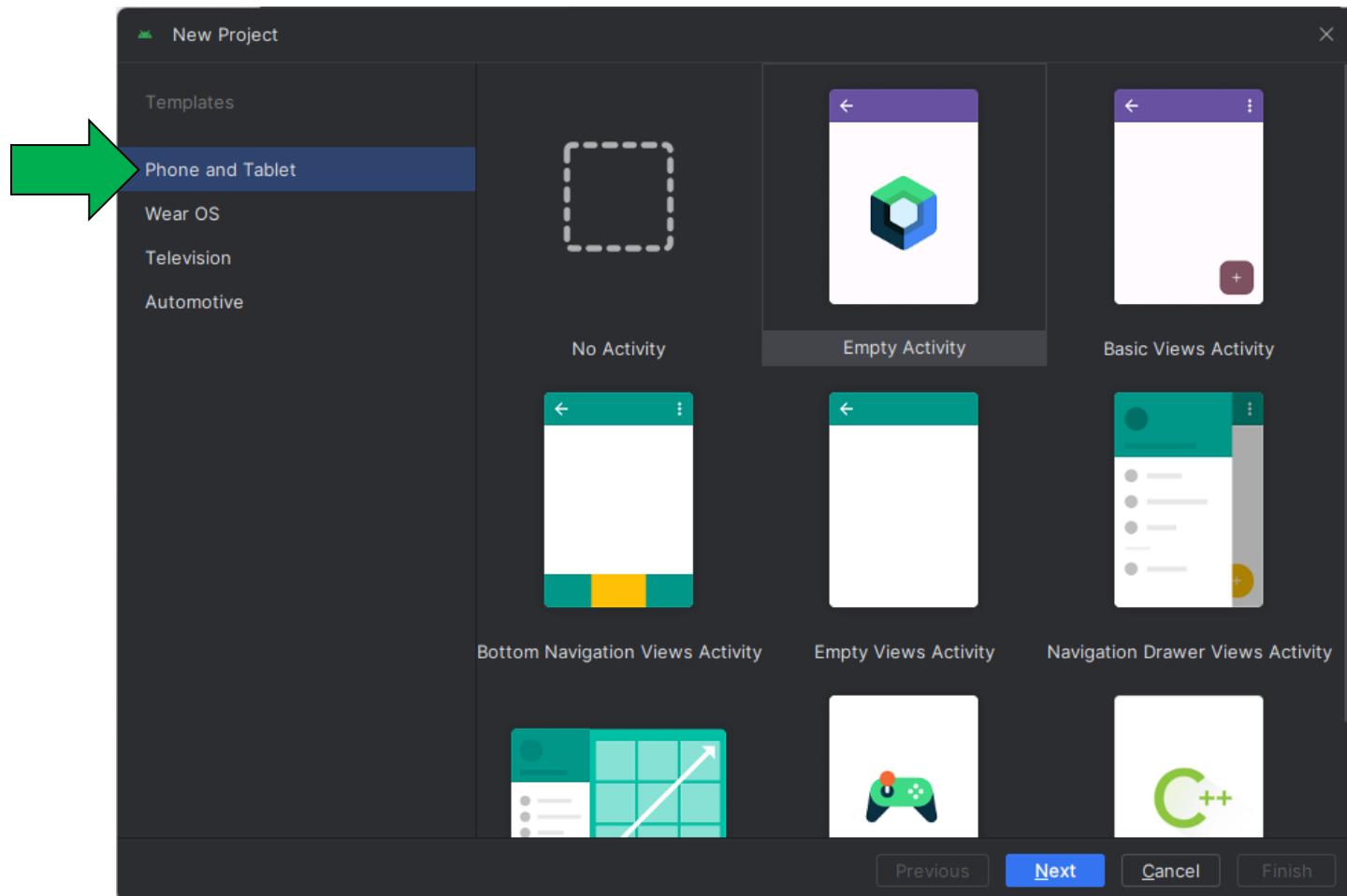
Primeira aplicação Android

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



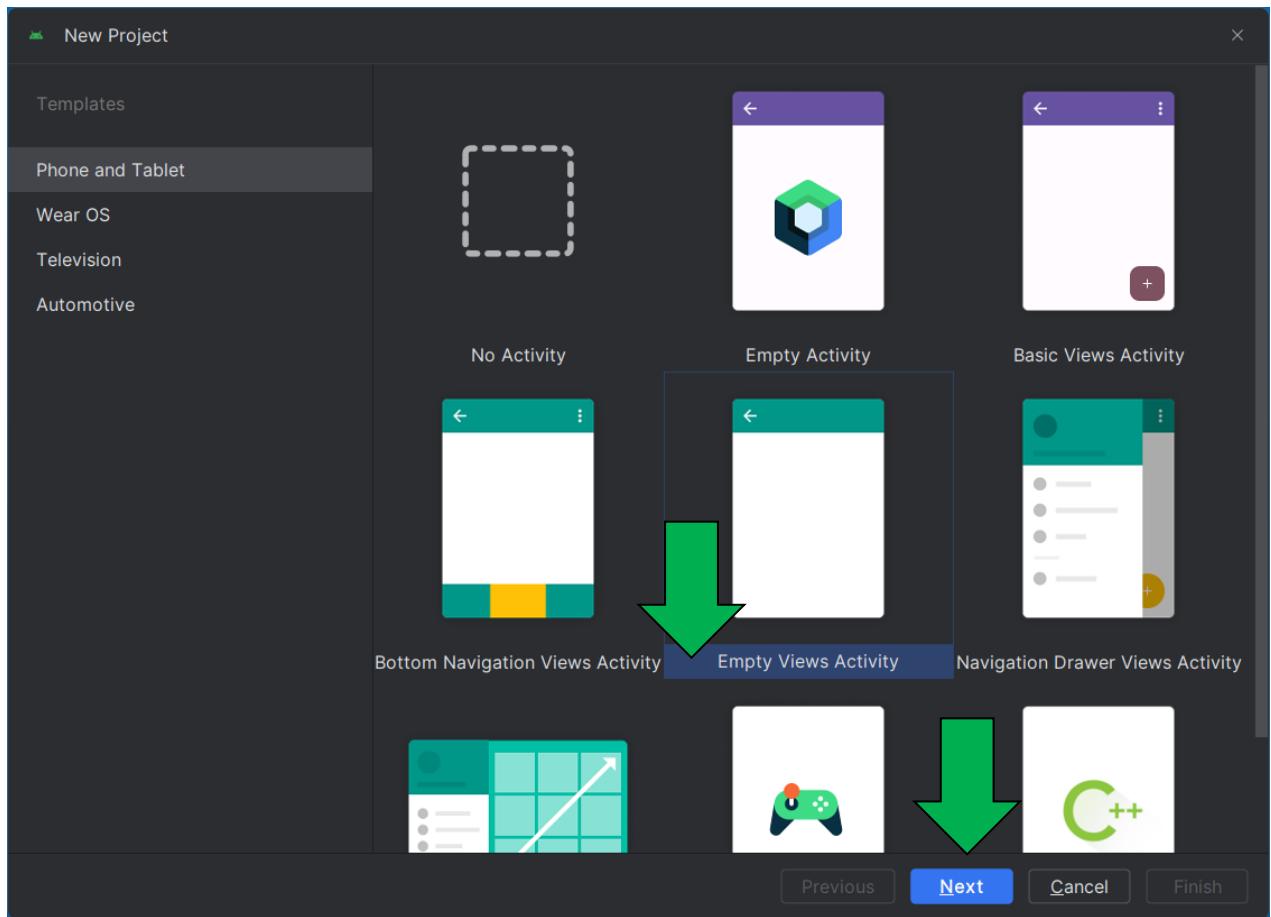
Primeira aplicação Android

- ⊕ 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:



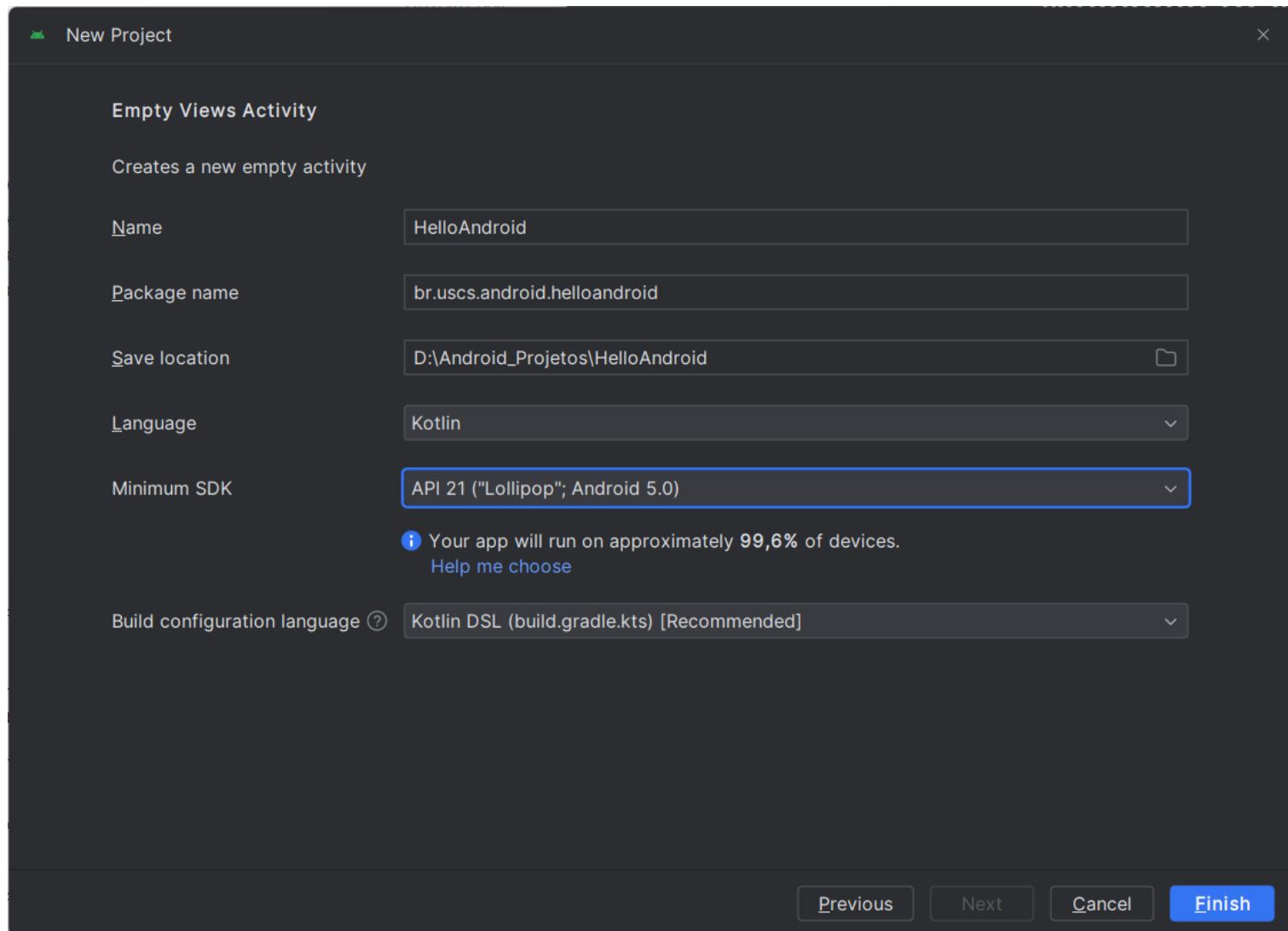
Primeira aplicação Android

- ⊕ 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.



Primeira aplicação Android

⊕ Preencha as informações do projeto:



Primeira aplicação Android

- ⊕ **Nome** é onde você insere o nome do aplicativo. Para esta unidade, nomeie o aplicativo como **HelloAndroid**.
- ⊕ **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.helloandroid**.
- ⊕ **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.

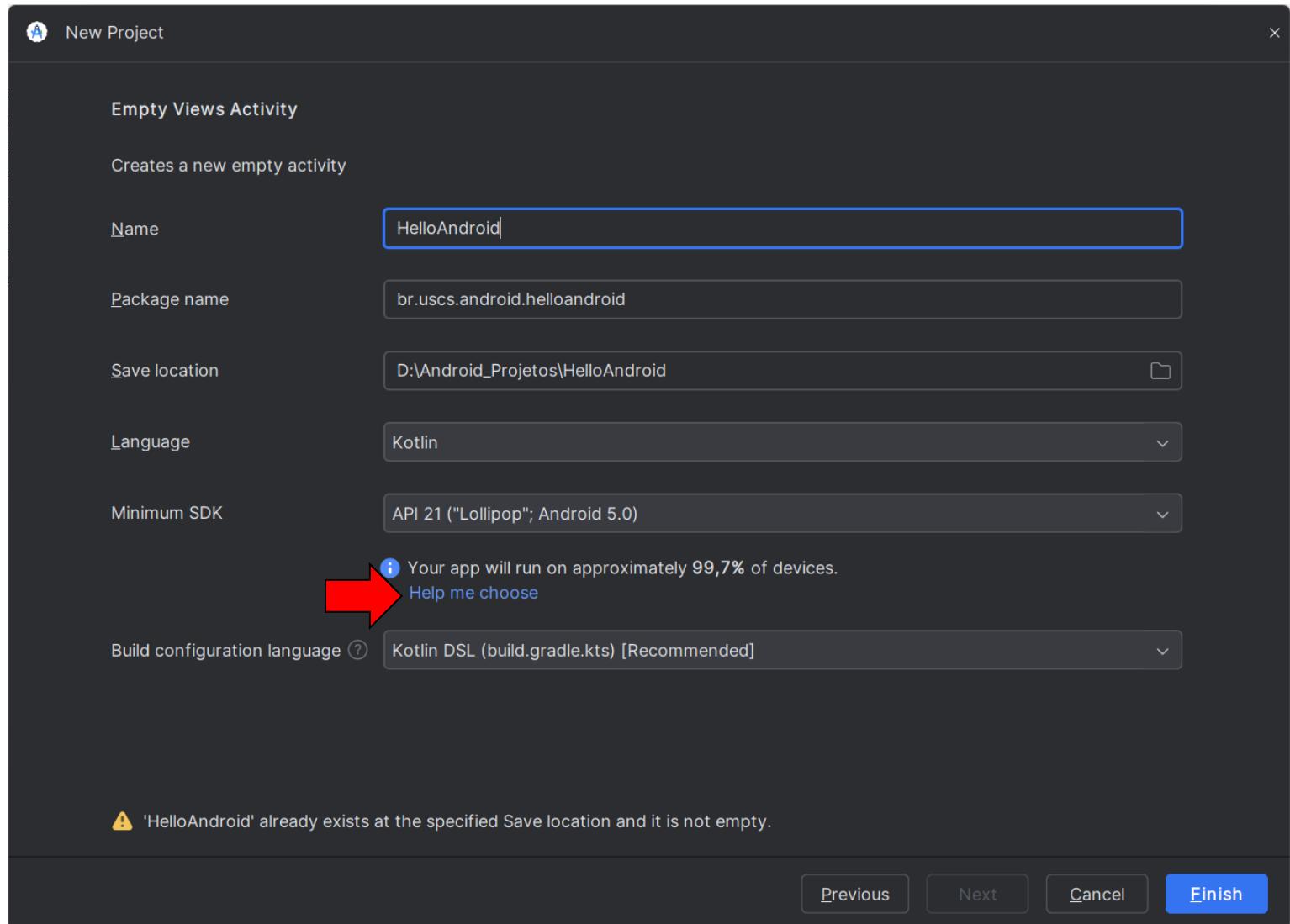


Primeira aplicação Android

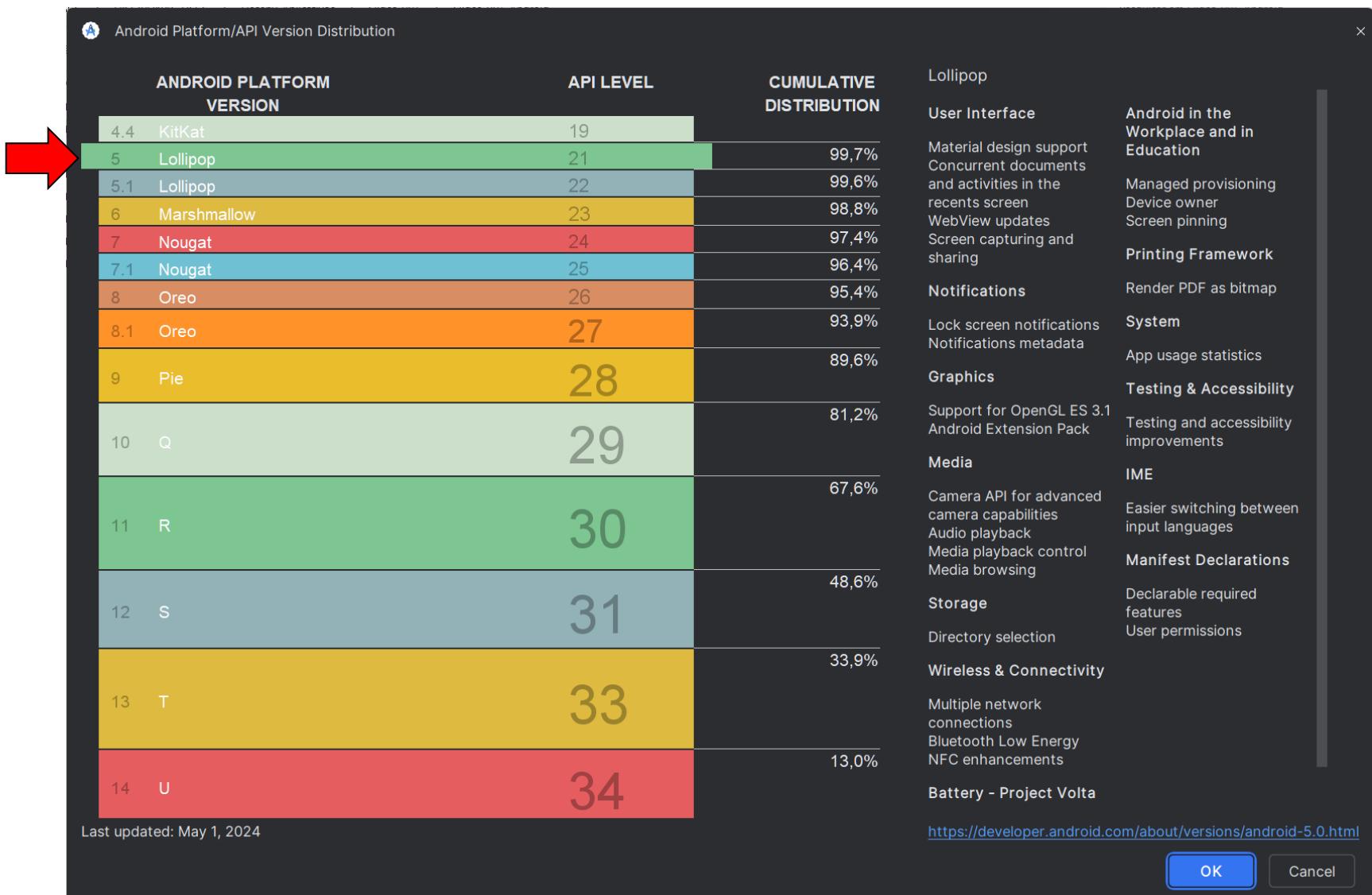
- ⊕ **Linguagem** informa ao **Android Studio** a linguagem que seu projeto requer. **Kotlin** já está selecionado, então nenhuma ação é necessária aqui (**Kotlin é padrão**);
- ⊕ **SDK Mínimo** define a versão mínima do **Android** que o aplicativo suportará. Neste curso, nossos aplicativos rodarão na **API 21**, ou Android **Lollipop** em inglês, portanto, certifique-se de que a **API 21**: Android 5.0 (Lollipop) esteja selecionada;
- ⊕ **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.



Primeira aplicação Android

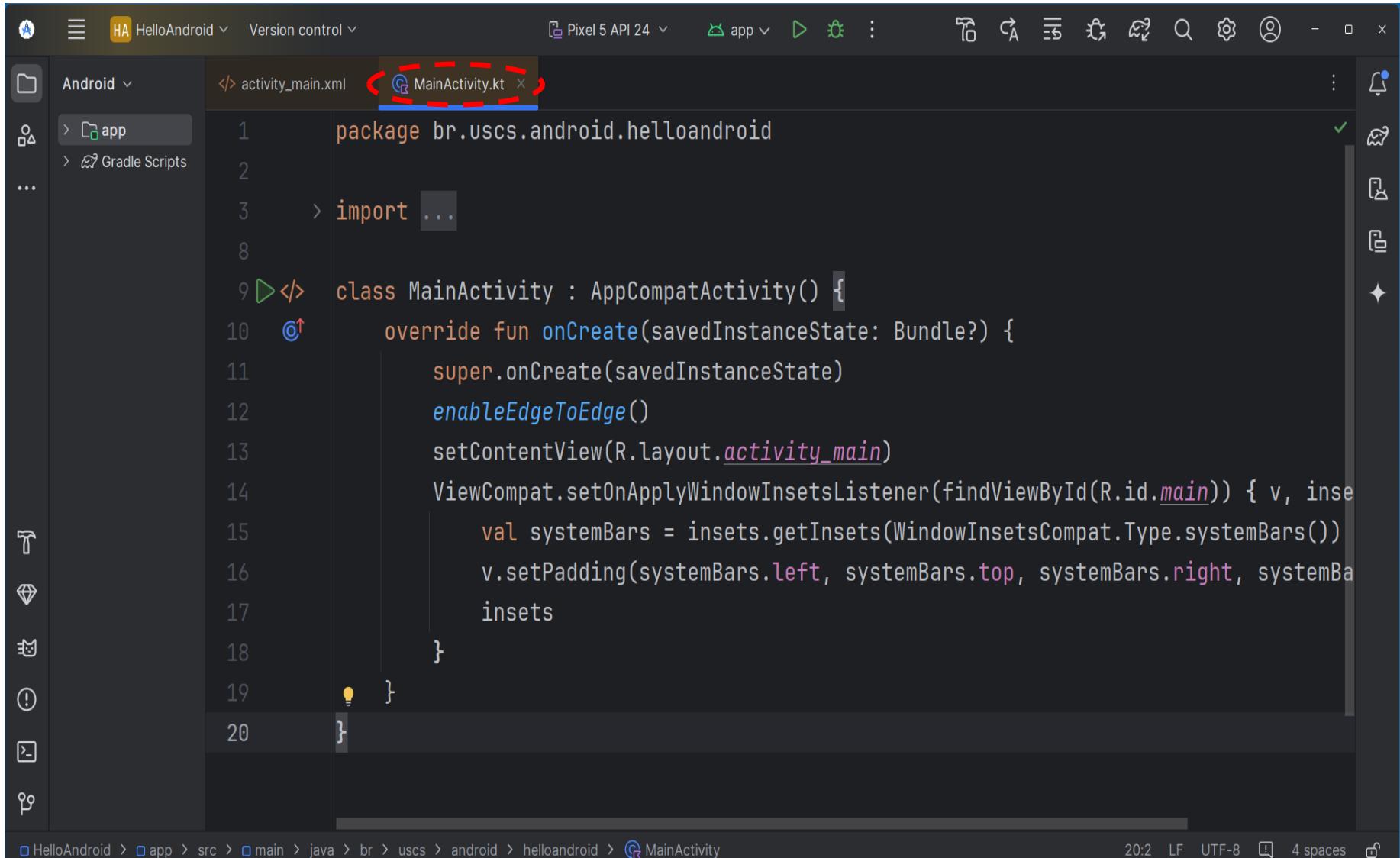


Primeira aplicação Android



Primeira aplicação Android

⊕ Projeto criado !



The screenshot shows the Android Studio interface with the project 'HelloAndroid' open. The code editor displays the 'MainActivity.kt' file, which contains the following Kotlin code:

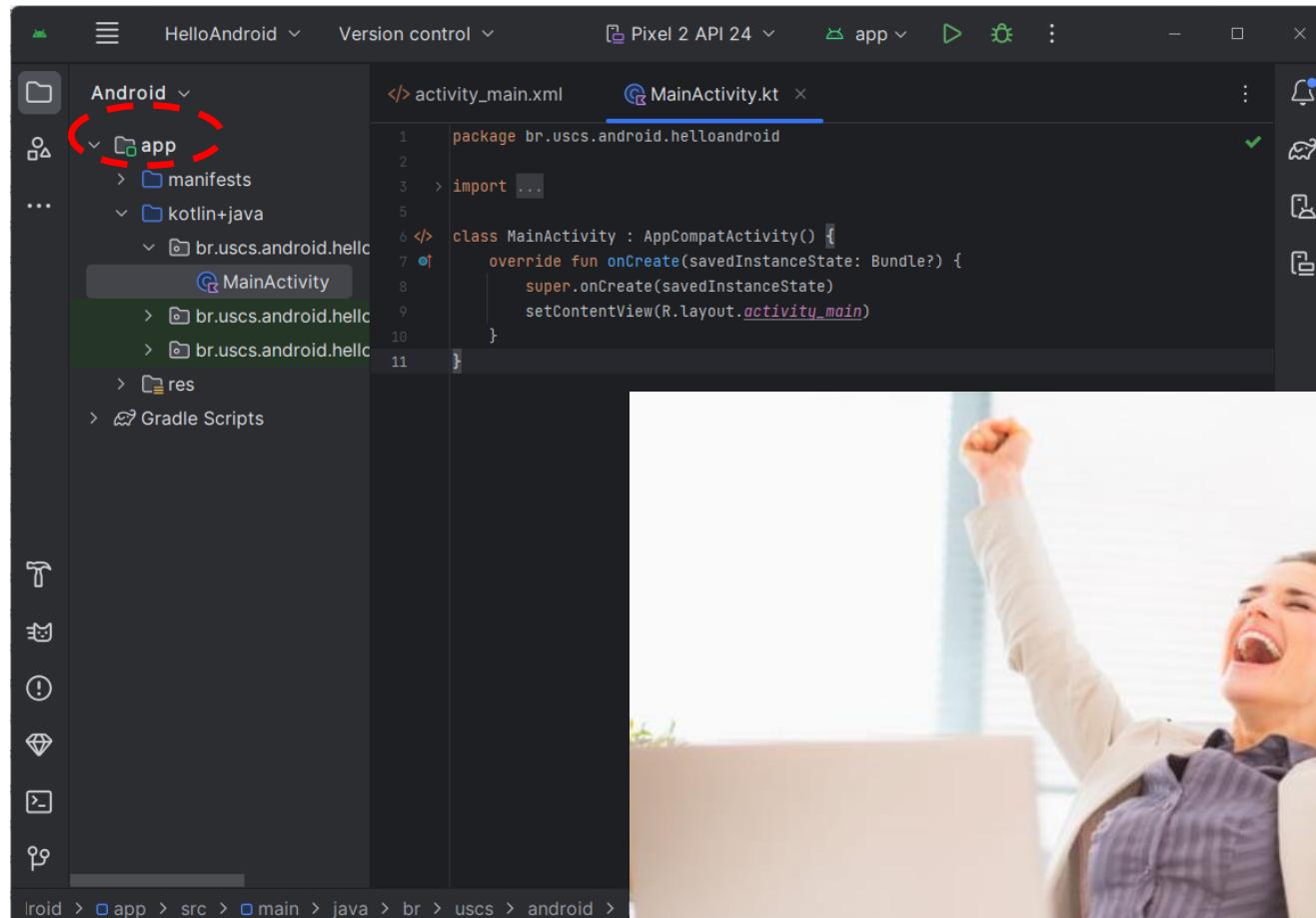
```
1 package br.uscs.android.helloandroid
2
3 > import ...
8
9 </> class MainActivity : AppCompatActivity() {
10    @Override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        enableEdgeToEdge()
13        setContentView(R.layout.activity_main)
14        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets -
15            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
16            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBa-
17            insets
18        }
19    }
20}
```

The code editor has several annotations: red dashed arrows pointing to the file tab and the title bar, a blue checkmark icon in the top right corner, and a yellow lightbulb icon on line 19 indicating a potential issue or suggestion.

At the bottom of the screen, the navigation bar shows the path: HelloAndroid > app > src > main > java > br > uscs > android > helloandroid > MainActivity.kt. The status bar at the bottom right indicates the time is 20:2, the line separator is LF, the encoding is UTF-8, there are 4 spaces, and the screen resolution is 16.

Primeira aplicação Android

⊕ Projeto criado !



The screenshot shows the Android Studio interface with a project named "HelloAndroid". A red circle highlights the "app" folder in the Project Structure sidebar. The code editor displays the MainActivity.kt file, which contains the following code:

```
1 package br.uscs.android.helloandroid
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_main)
9     }
10 }
```

The status bar at the bottom shows the path: android > app > src > main > java > br > uscs > android >





Android Studio exige que a virtualização esteja habilitada ?

Primeira aplicação Android - Virtualização

- ⊕ O **Android Studio**, por si só, **não** exige que a virtualização esteja habilitada;
- ⊕ No entanto, para usar o Android Emulator de forma eficiente e com melhor desempenho, recomend-se habilitar a virtualização;
- ⊕ A virtualização ajuda a acelerar a execução do emulador aproveitando recursos de hardware específicos;
- ⊕ Nos processadores **Intel**, a **Virtualização** de hardware é denominada **VT-X** e nos **AMD AMD-V**;
- ⊕ Esta ativação é feita na **BIOS** do ambiente.



Como saber se a virtualização está habilitada ?

Como saber se a virtualização está habilitada ?

⊕ Abra o Gerenciador de Tarefas

A screenshot of the Windows Task Manager. On the left, the navigation pane shows 'Desempenho' (Performance) selected. The main area displays performance metrics for various components:

- CPU: 2% 1,13 GHz
- Memória: 12/128 GB (9%)
- Disco 0 (H: R): SSD 0%
- Disco 1 (X): SSD 0%
- Disco 2 (Y): SSD 0%
- Disco 3 (C: E): SSD 0%
- Disco 4 (G): SSD 0%
- Disco 5 (D): SSD 0%
- Disco 6 (F): SSD 0%
- Disco 7 (K): USB 0%

The right side shows a graph of CPU usage over 60 seconds, with a peak at 1,13 GHz. Below the graph, detailed processor information is listed:

Utilização	Velocidade	Velocidade base:	3,60 GHz
2%	1,13 GHz	Sockets:	1
Processos	Threads	Núcleos:	12
260	4015	Identificadores	20
Tempo de atividade		Processadores lógicos:	Habilitado
0:02:22:37		Virtualização:	Cache L1: 1,0 MB
			Cache L2: 12,0 MB
			Cache L3: 25,0 MB

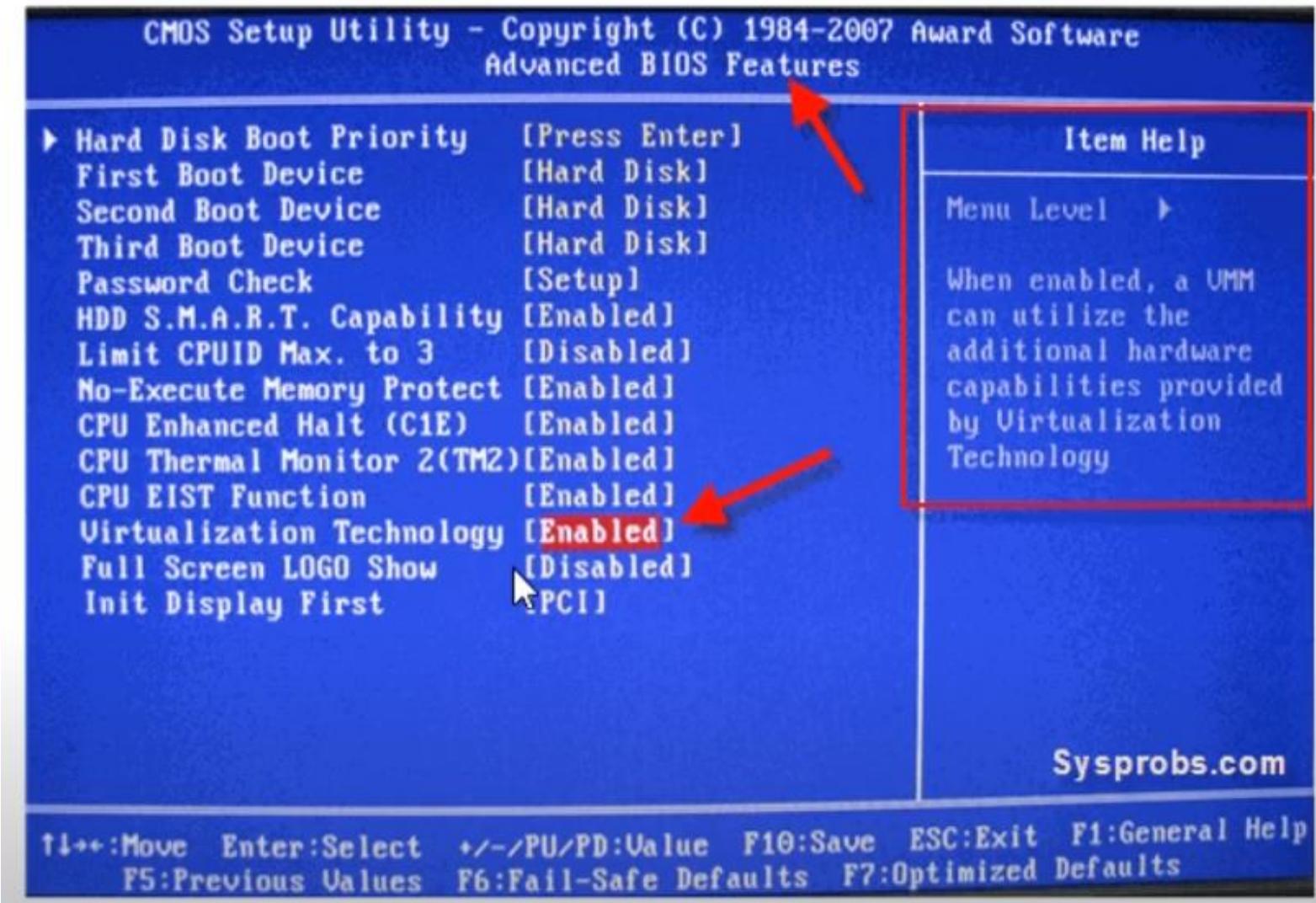
A green arrow points to the 'Desempenho' section in the navigation pane, and another green arrow points to the 'Virtualização:' row in the processor details table.



E se a virtualização NÃO estiver habilitada ?

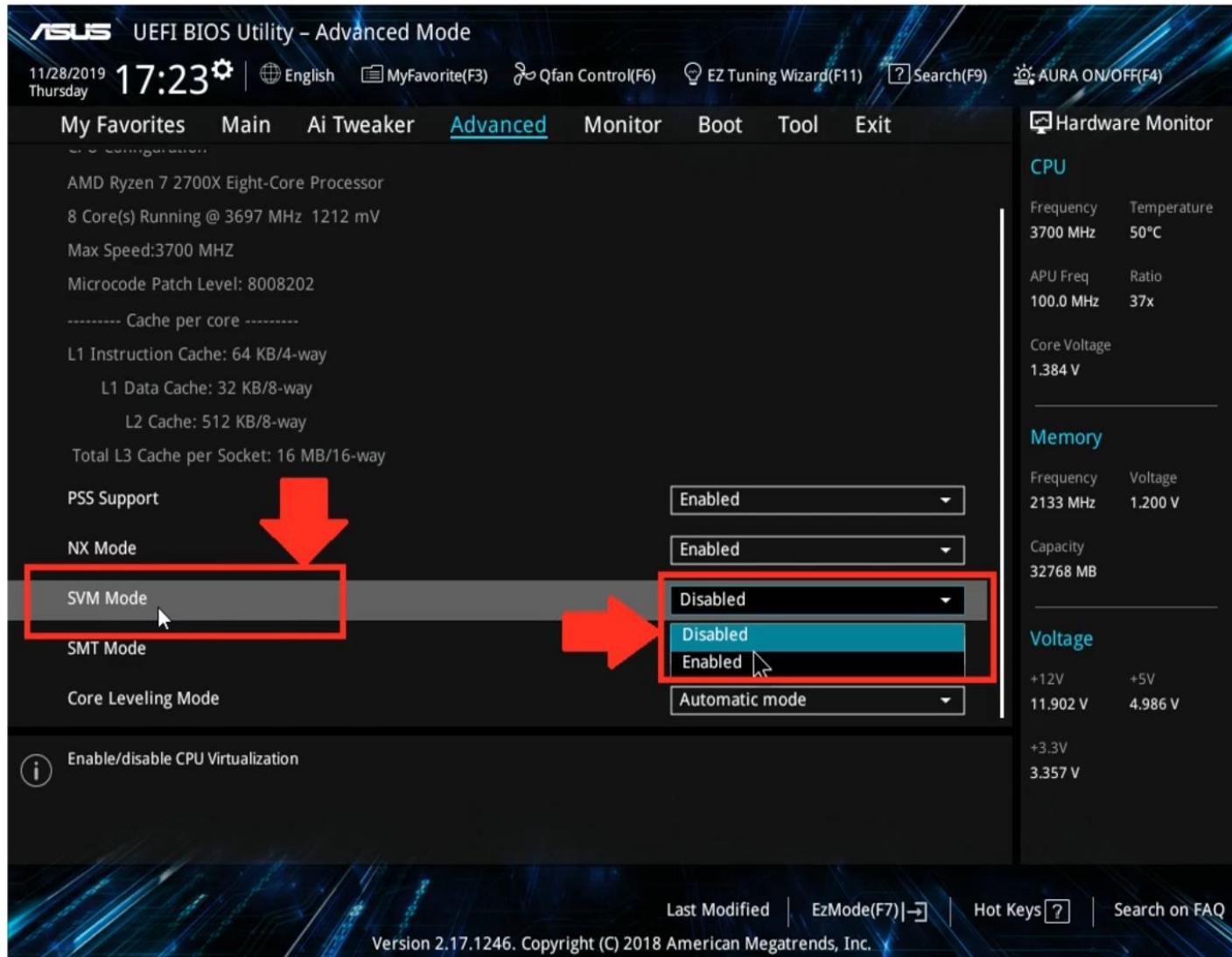
E se a virtualização NÃO estiver habilitada ?

- ⊕ Será necessário habilitar na BIOS



E se a virtualização NÃO estiver habilitada ?

⊕ Será necessário habilitar na BIOS



Classe MainActivity

- Após a criação do projeto, **Android Studio** exibe-o com os arquivos **MainActivity.kt** e **activity_main.xml** abertos.



```
</> activity_main.xml MainActivity.kt
1 package br.uscs.android.helloandroid
2
3 > import ...
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.getInsets(WindowInsetsCompat.Type.systemBars)
16            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.b
17            insets
18        }
19    }
20 }
```

The screenshot shows the Android Studio interface with the project 'HelloAndroid' open. The left sidebar shows the project structure with 'app' selected. The main editor window displays the 'MainActivity.kt' file, which contains the code for a Kotlin Activity. A large green arrow points downwards from the top of the slide towards the code editor area. The status bar at the bottom shows the file path 'HelloAndroid > app > src > main > java > br > uscs > android > helloandroid > MainActivity' and the status '20:2 LF UTF-8 4 spaces'.

Classe MainActivity

- Após a criação do projeto, **Android Studio** exibe-o com os arquivos **MainActivity.kt** e **activity_main.xml** abertos.



The screenshot shows the Android Studio interface. On the left is the Project Navigational Drawer, which includes sections for Android, Java, and Gradle Scripts. Under the Java section, the 'app' module is selected. In the center is the main code editor window. The title bar shows 'activity_main.xml' is the active file. The editor displays XML code for a ConstraintLayout. A large green arrow points downwards from the title bar towards the 'activity_main.xml' tab. The bottom of the screen shows the file path 'HelloAndroid > app > src > main > res > layout > activity_main.xml' and various status indicators like '18:1', 'LF', 'UTF-8', and '4 spaces'.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

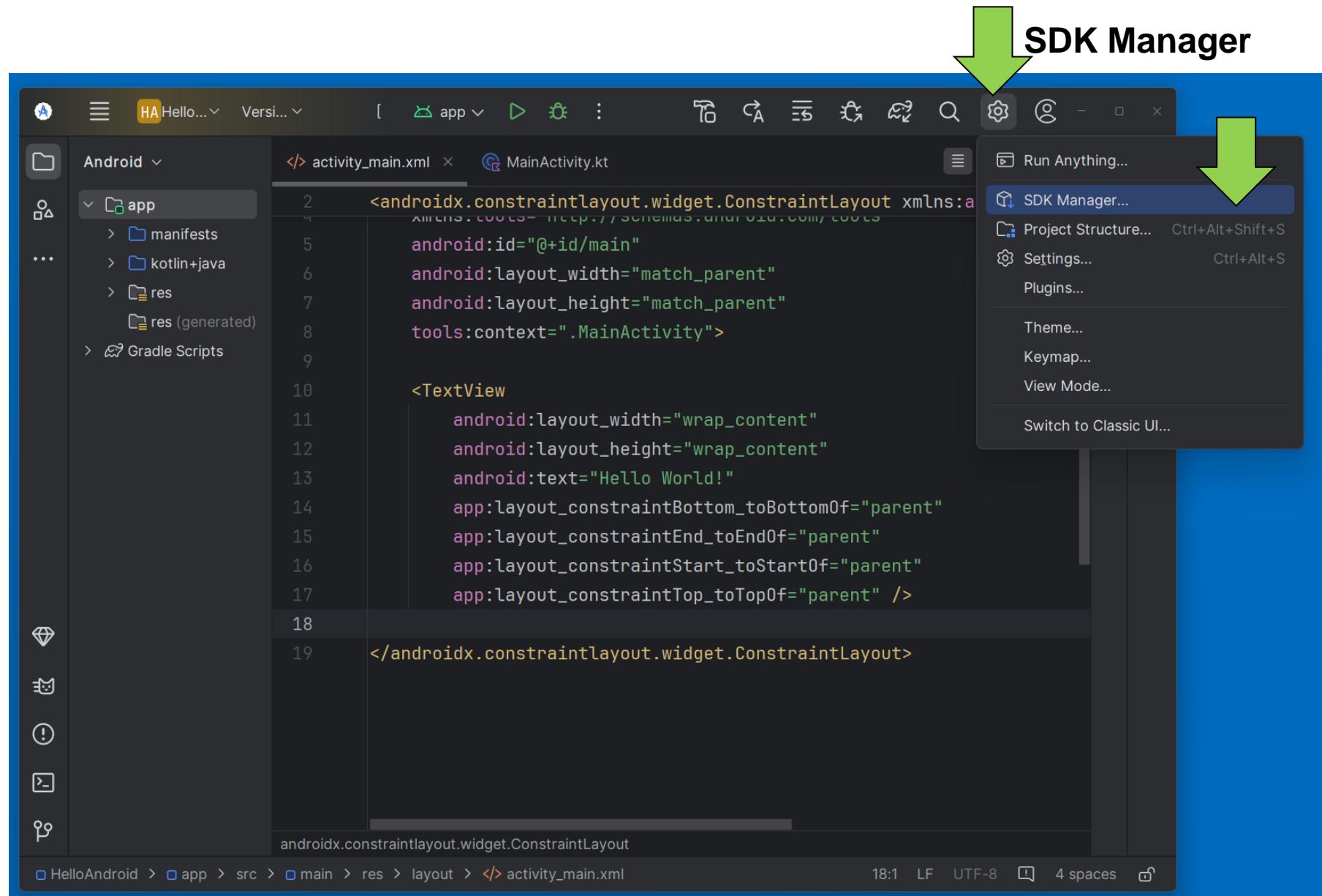
</androidx.constraintlayout.widget.ConstraintLayout>
```

Android Studio

- ⊕ Com o projeto criado, você está **pronto** para trabalhar nele;
- ⊕ Mas **não se apresse** tanto;
- ⊕ O **Android Studio** é um software **complexo**;
- ⊕ Se você mergulhar rápido demais, poderá se perder.
- ⊕ Antes de começar a construir seu aplicativo, reserve um momento para **revisar** o que o **Android Studio** tem a oferecer.



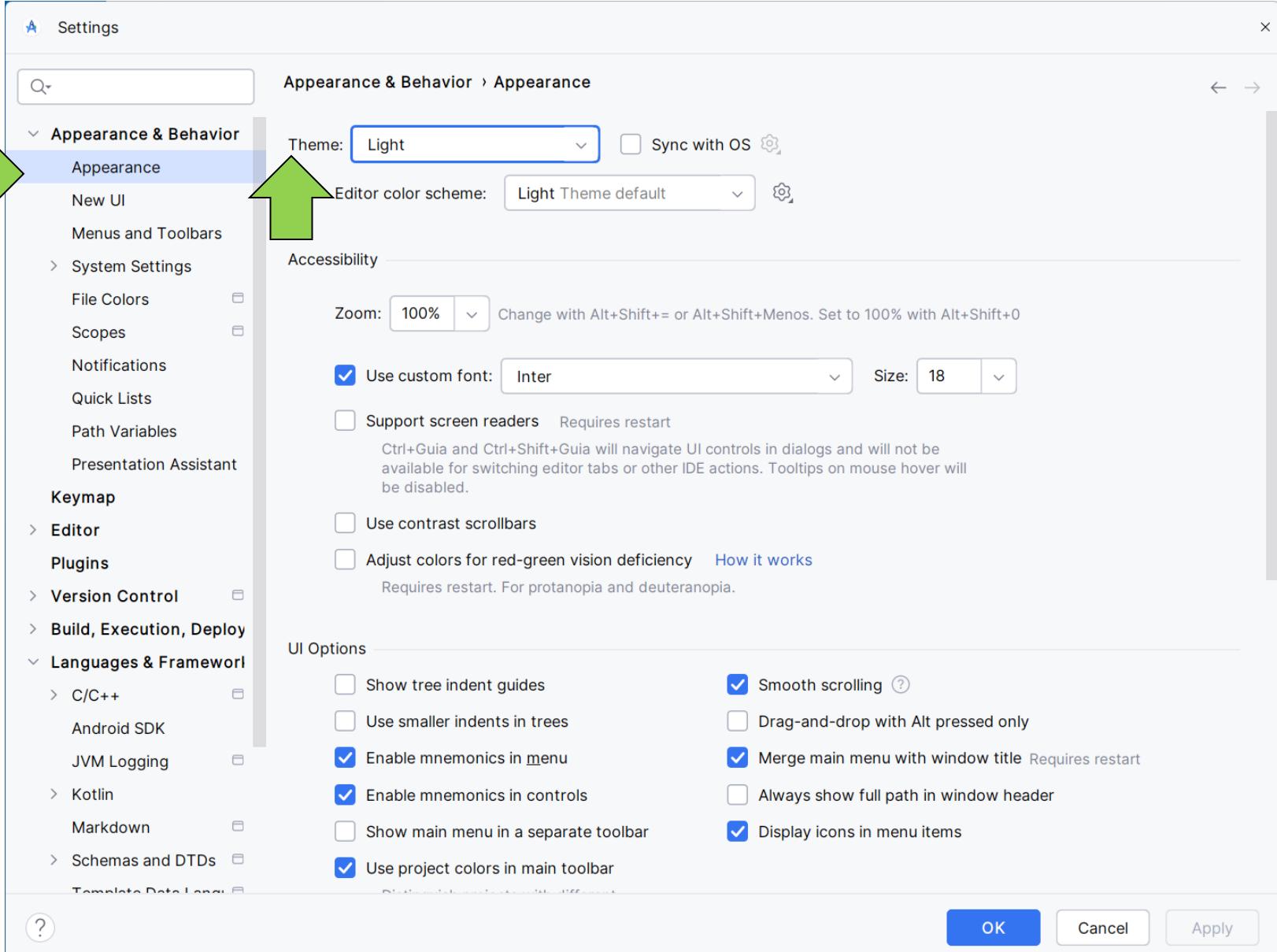
Android Studio – Configurações de Projeto



Android Studio – Configurações de Projeto

The screenshot shows the 'Appearance & Behavior > Appearance' settings in the Android Studio Settings dialog. The 'Theme' dropdown is set to 'Dark'. The 'Sync with OS' checkbox is unchecked. The 'Editor color scheme' dropdown is set to 'Dark Theme default'. The 'Accessibility' section includes a 'Zoom' slider at 100%, a note about keyboard navigation, and several checkboxes: 'Use custom font' (Inter, Size 18), 'Support screen readers' (disabled), 'Use contrast scrollbars' (disabled), and 'Adjust colors for red-green vision deficiency' (disabled). The 'How it works' link is visible. The 'UI Options' section contains several checkboxes: 'Show tree indent guides' (disabled), 'Smooth scrolling' (checked), 'Use smaller indents in trees' (disabled), 'Drag-and-drop with Alt pressed only' (disabled), 'Enable mnemonics in menu' (checked), 'Merge main menu with window title' (checked), 'Requires restart', 'Show main menu in a separate toolbar' (disabled), 'Always show full path in window header' (disabled), 'Enable mnemonics in controls' (checked), 'Display icons in menu items' (checked), and 'Use project colors in main toolbar' (checked). The bottom right of the dialog has 'OK', 'Cancel', and 'Apply' buttons.

Android Studio – Configurações de Projeto



The screenshot shows the 'Appearance & Behavior > Appearance' settings in the Android Studio Preferences. The 'Theme' dropdown is set to 'Light'. The 'Editor color scheme' dropdown is set to 'Light Theme default'. The 'OK' button at the bottom right is highlighted.

Appearance & Behavior

Theme: Light Sync with OS

Editor color scheme: Light Theme default

New UI

Menus and Toolbars

System Settings

File Colors

Scopes

Notifications

Quick Lists

Path Variables

Presentation Assistant

Keymap

Editor

Plugins

Version Control

Build, Execution, Deploy

Languages & Framework

C/C++

Android SDK

JVM Logging

Kotlin

Markdown

Schemas and DTDs

Template Data Languages

UI Options

Show tree indent guides

Smooth scrolling

Use smaller indents in trees

Drag-and-drop with Alt pressed only

Enable mnemonics in menu

Merge main menu with window title

Enable mnemonics in controls

Always show full path in window header

Show main menu in a separate toolbar

Display icons in menu items

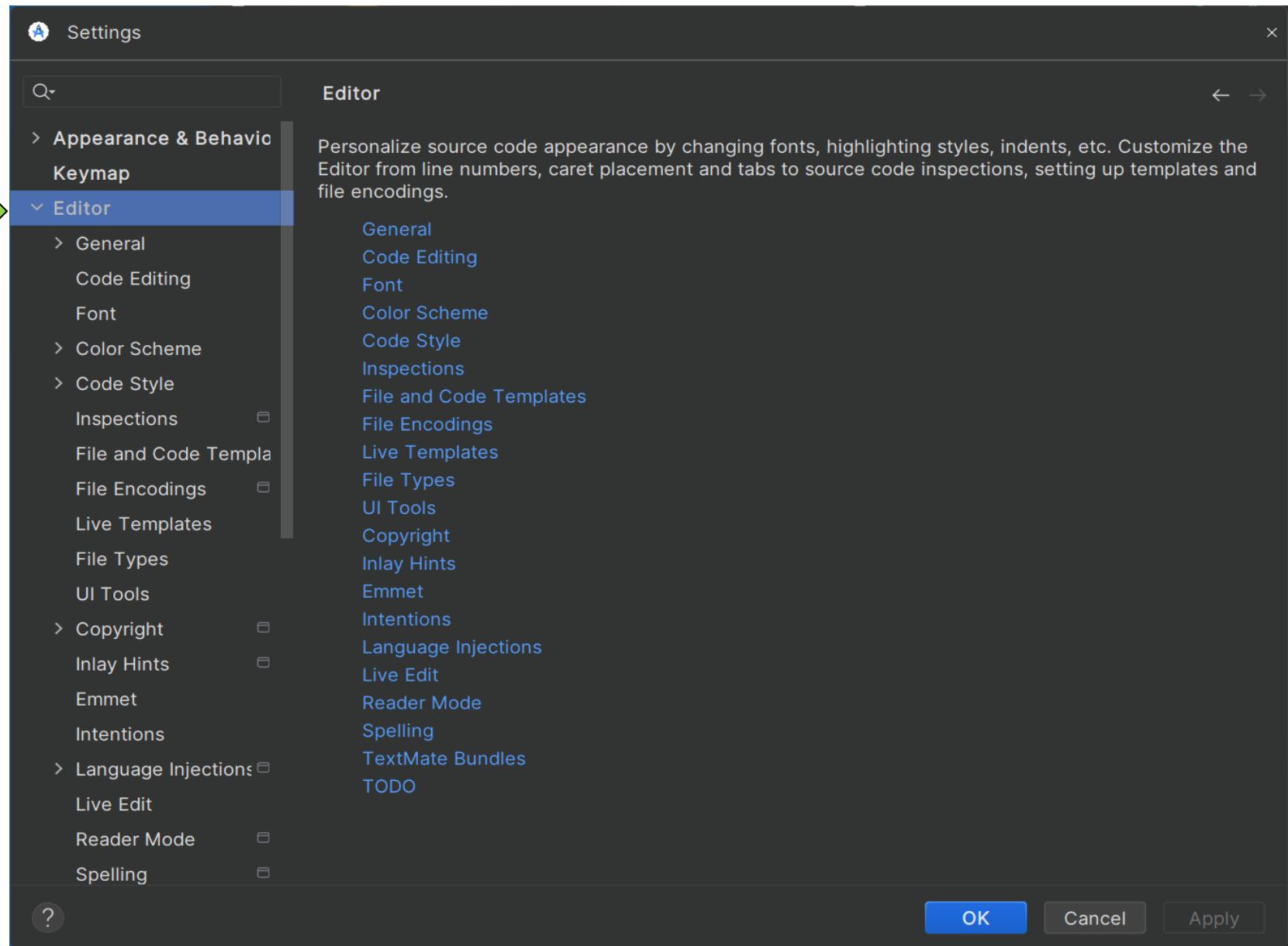
Use project colors in main toolbar

OK Cancel Apply

Android Studio – Configurações de Projeto

The screenshot shows the 'Appearance & Behavior > Appearance' settings in Android Studio. The left sidebar is expanded, showing categories like Appearance & Behavior, Appearance, System Settings, Editor, Plugins, etc. A large green arrow points from the top-left towards the sidebar. On the right, under the 'Appearance' tab, there's a 'Theme' dropdown set to 'VSCode Dark' and a 'Sync with OS' checkbox. Below that, 'Editor color scheme' is set to 'VSCode Dark Brighter'. Under 'Accessibility', the 'Zoom' is set to '100%'. A red oval highlights the 'Use custom font' section, which has a checked checkbox and a dropdown menu set to 'Inter'. To the right of this, a green arrow points towards the 'Size' dropdown, which is set to '20'. The 'UI Options' section contains several checkboxes, some of which are checked (e.g., 'Smooth scrolling', 'Enable mnemonics in menu', 'Display icons in menu items'). At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

Android Studio – Configurações de Projeto



The screenshot shows the 'Settings' dialog in Android Studio, specifically the 'Editor' configuration screen. The left sidebar lists various editor-related settings under the 'Editor' category, which is currently selected and highlighted with a blue background. A large green arrow points to this 'Editor' section. The main content area displays a list of configuration options: General, Code Editing, Font, Color Scheme, Code Style, Inspections, File and Code Templates, File Encodings, Live Templates, File Types, UI Tools, Copyright, Inlay Hints, Emmet, Intentions, Language Injections, Live Edit, Reader Mode, Spelling, TextMate Bundles, and TODO.

Settings

Editor

Appearance & Behavior

Keymap

Editor

General

Code Editing

Font

Color Scheme

Code Style

Inspections

File and Code Templates

File Encodings

Live Templates

File Types

UI Tools

Copyright

Inlay Hints

Emmet

Intentions

Language Injections

Live Edit

Reader Mode

Spelling

TextMate Bundles

TODO

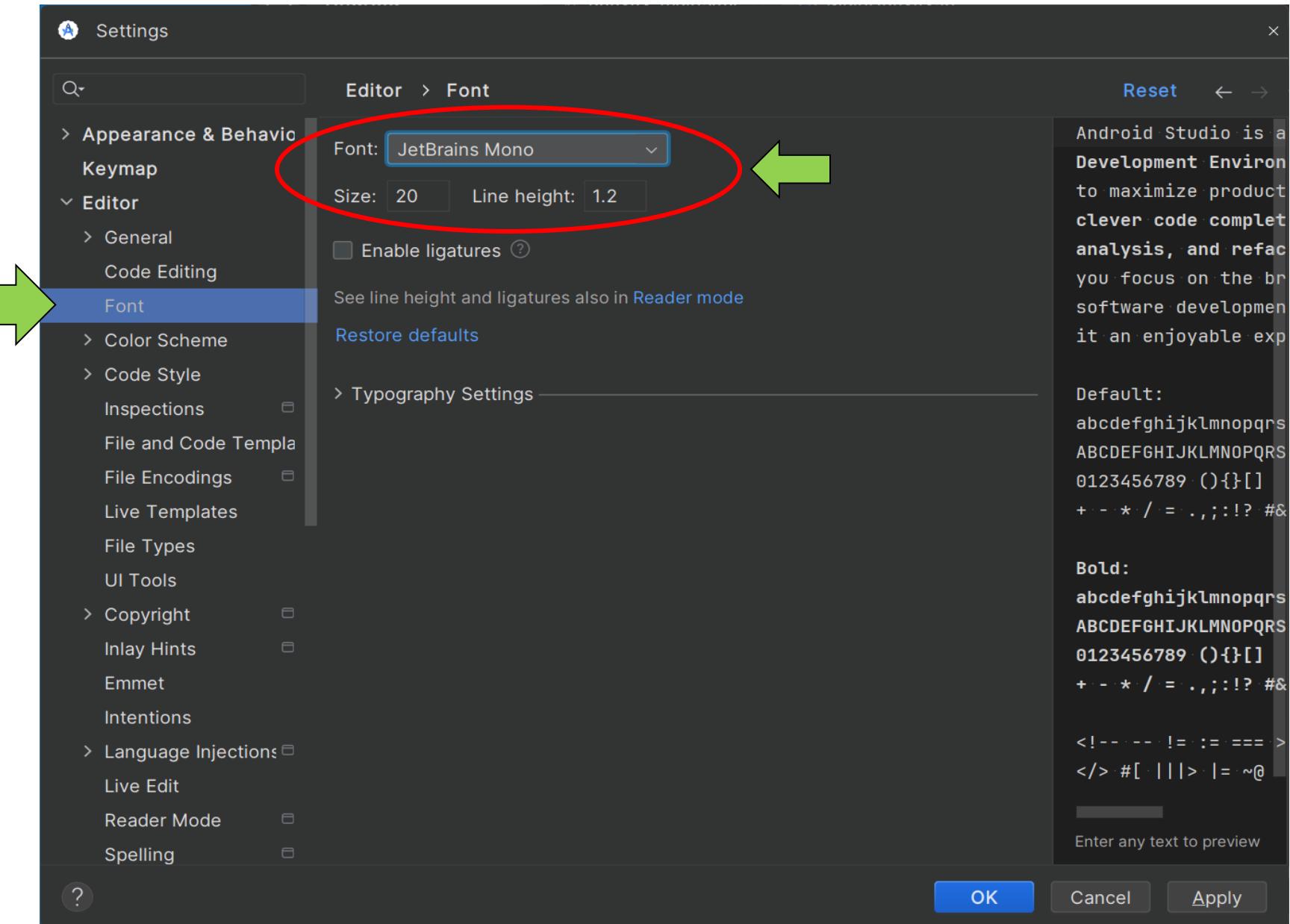
?

OK

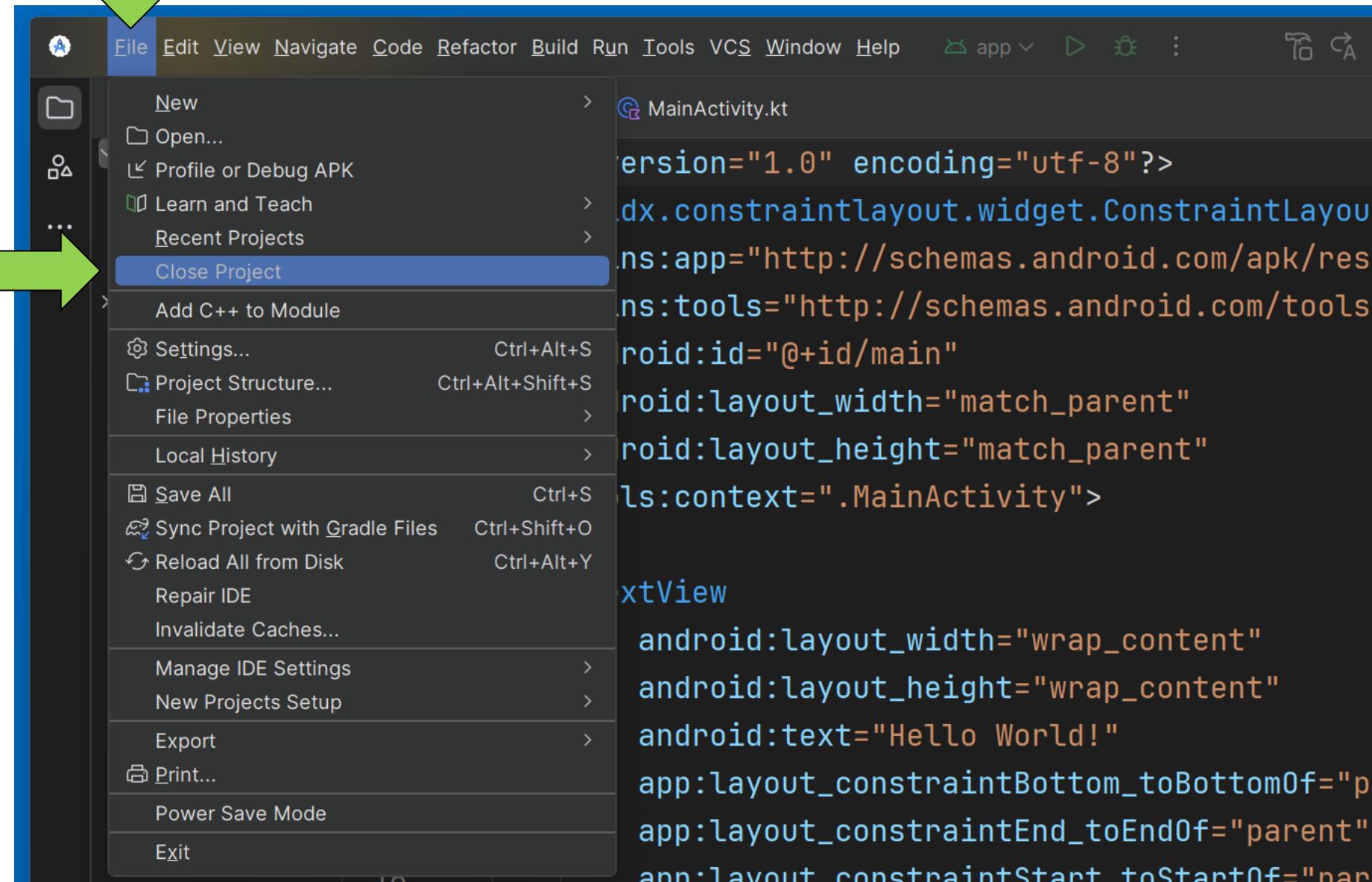
Cancel

Apply

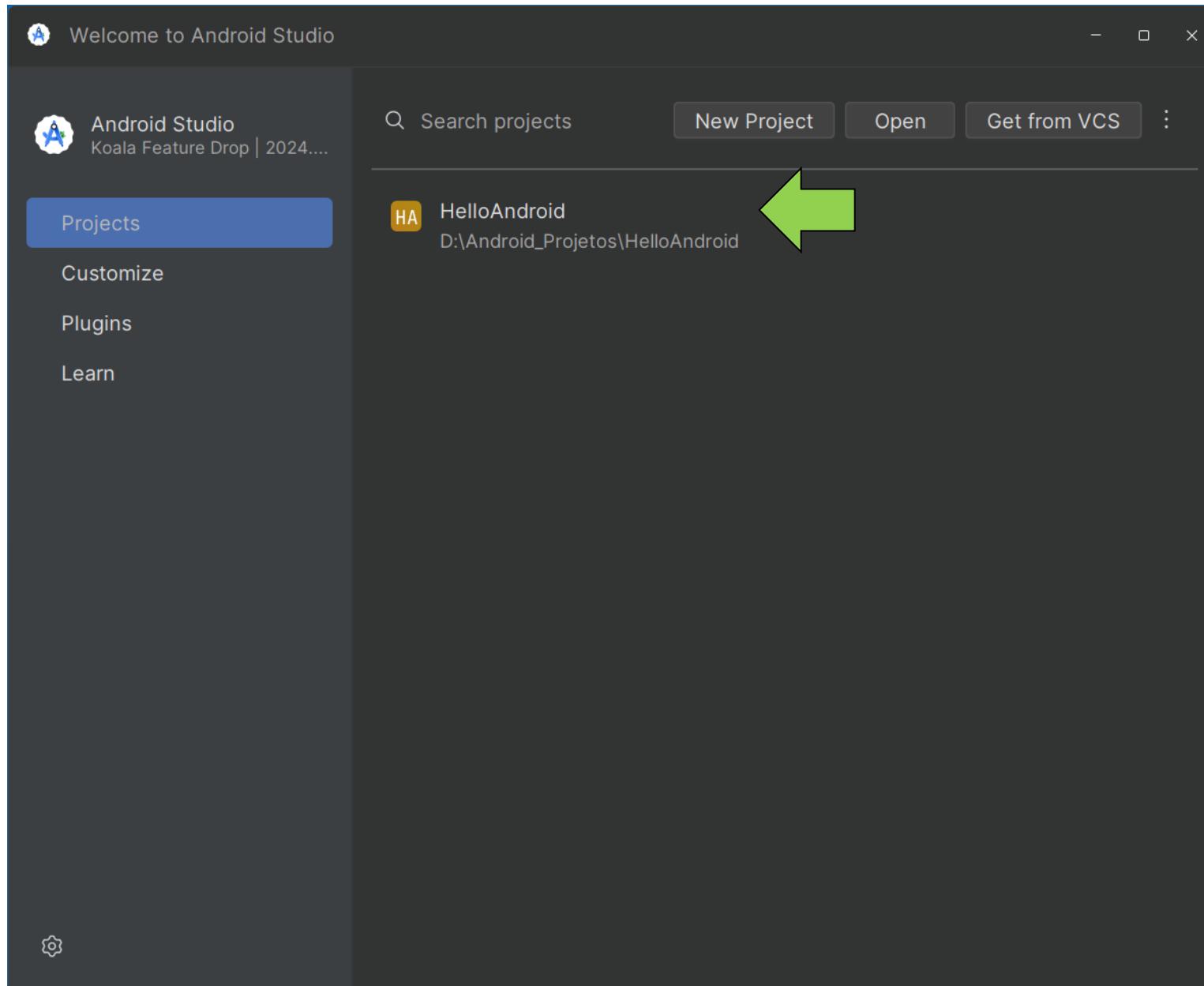
Android Studio – Configurações de Projeto



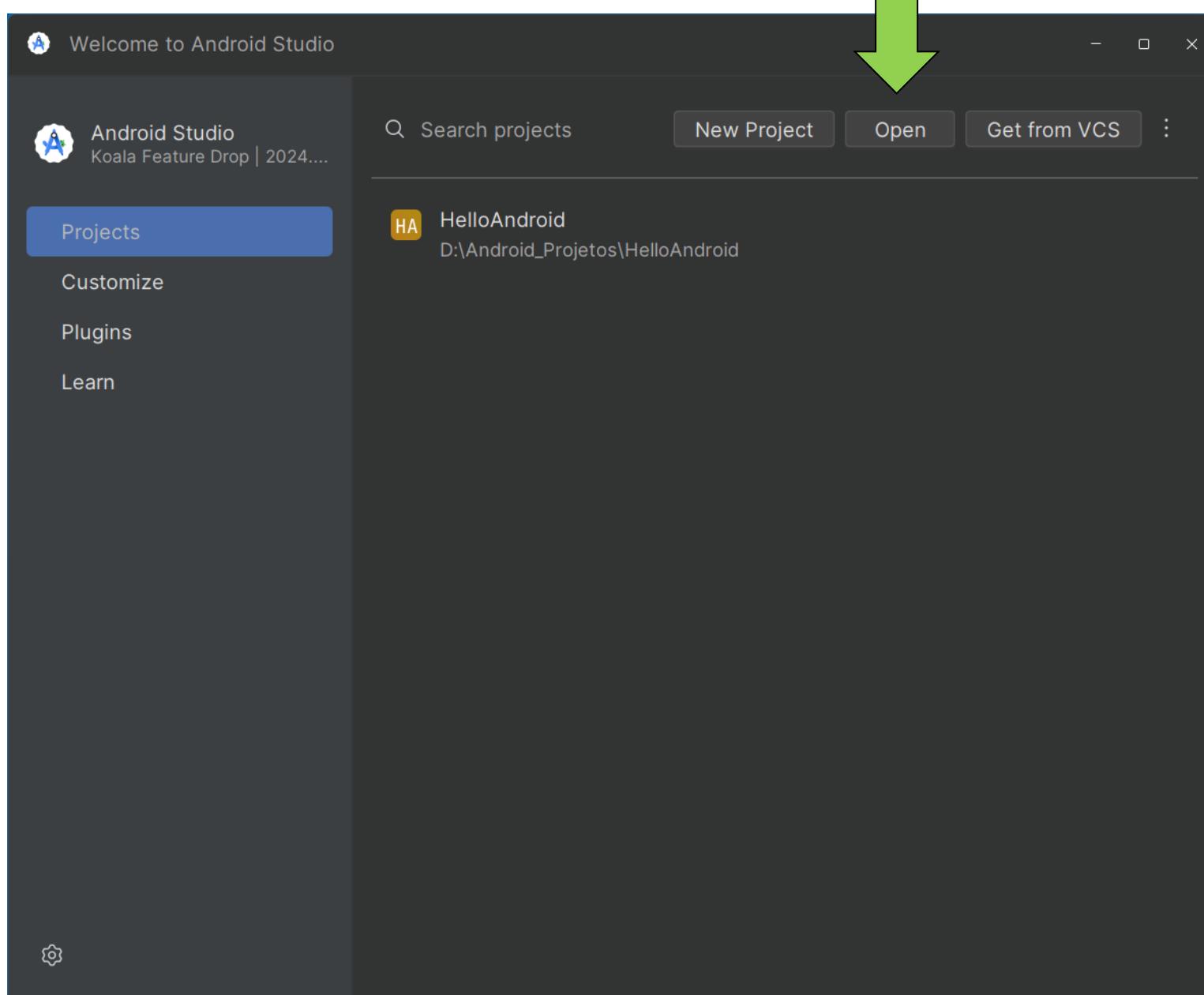
Android Studio – Fechando Projeto



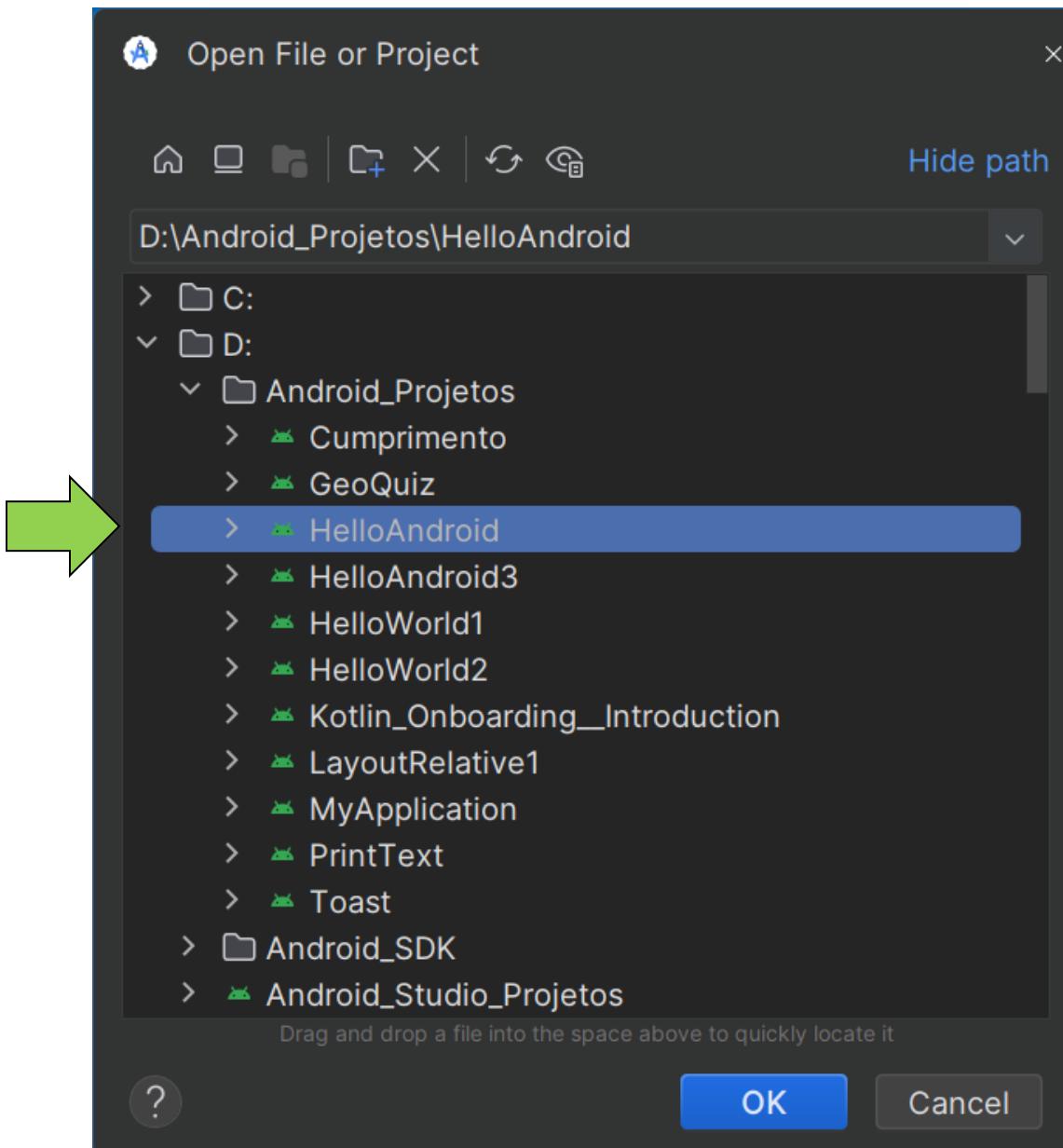
Android Studio – Abrindo Projeto



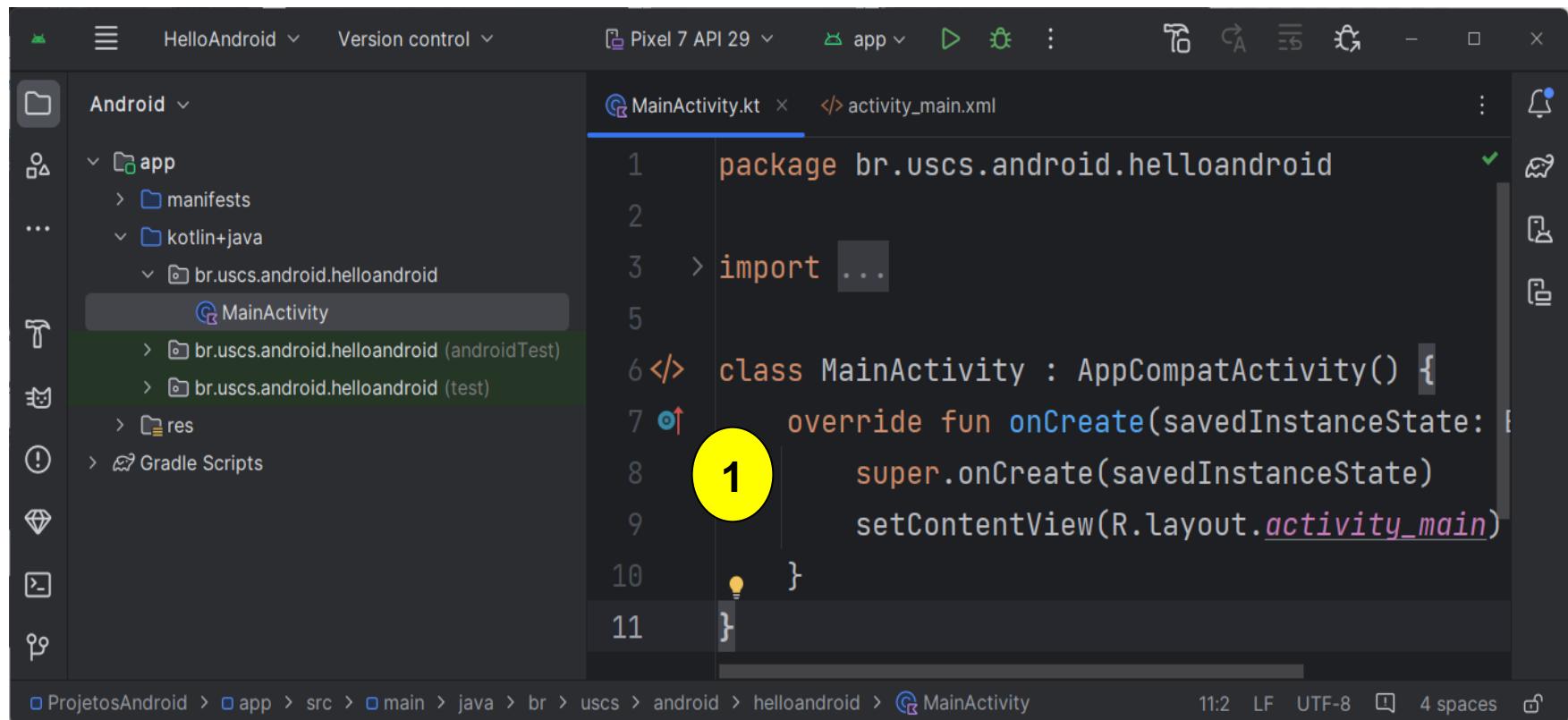
Android Studio – Abrindo Projeto



Android Studio – Abrindo Projeto



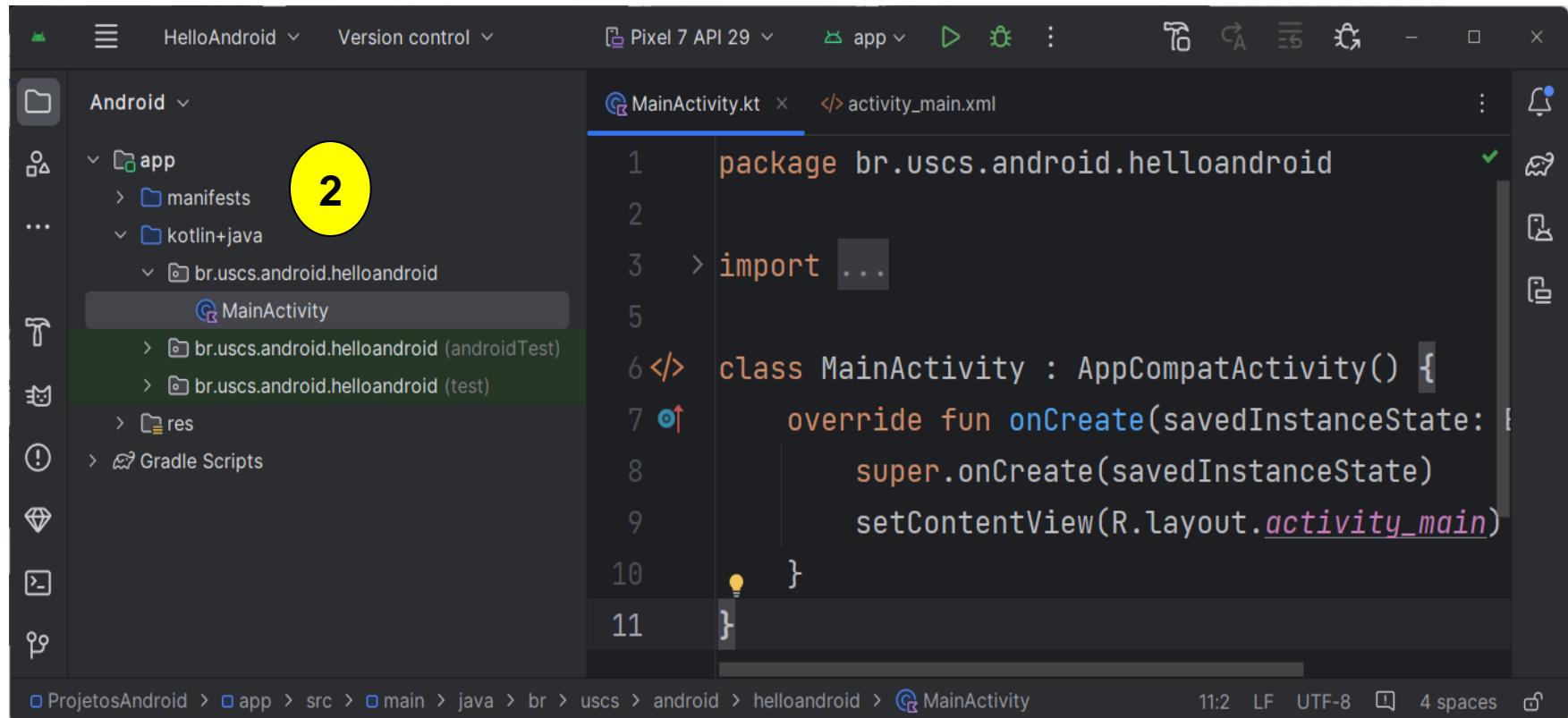
Android Studio



1

A primeira janela que aparece é o **Editor**. Esta janela fornece um espaço para editar o **código-fonte** do seu aplicativo. Inclui ferramentas como destaque de **sintaxe**, **auto-complete** para métodos e objetos, bem como a capacidade de definir **breakpoints** no seu código durante a depuração.

Android Studio



2

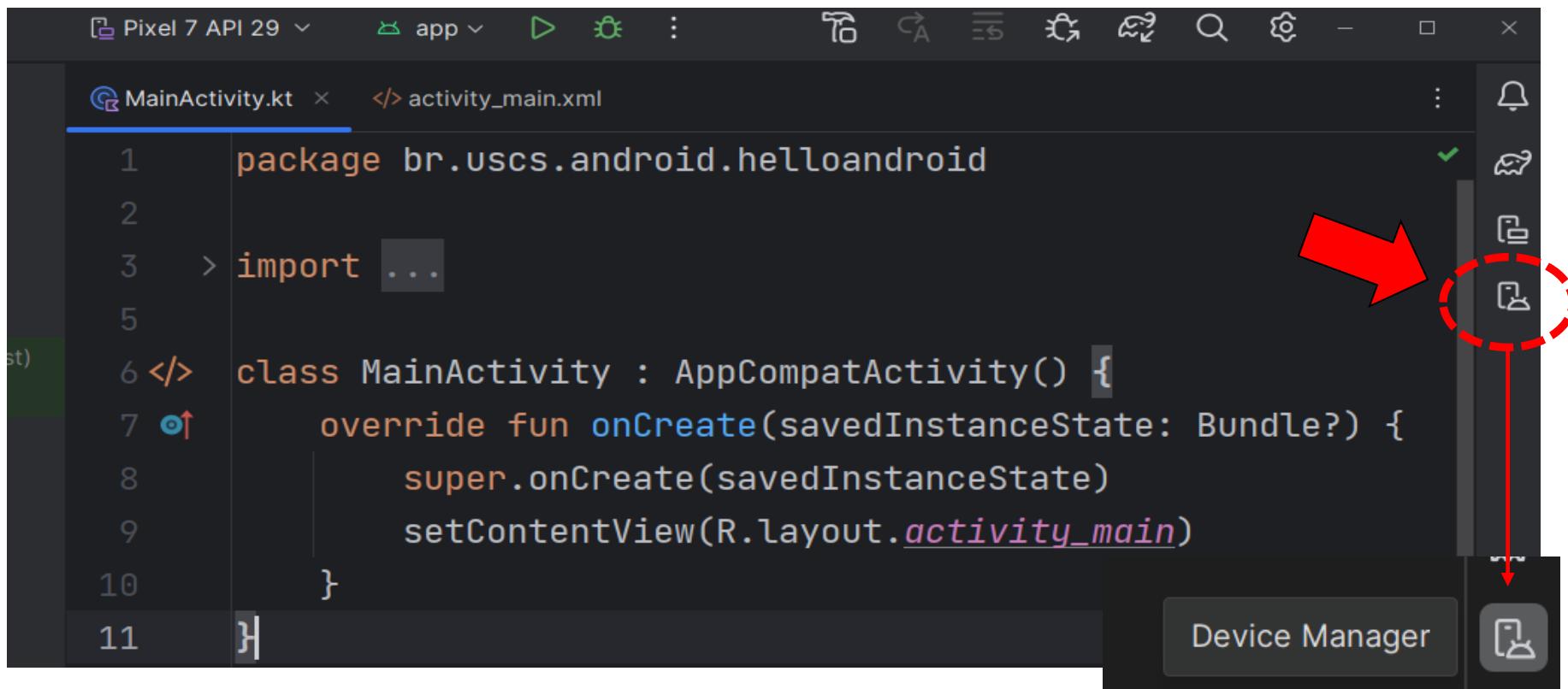
À esquerda do Editor está o **Project Navigator**. Esta janela mostra tudo o que seu projeto contém, incluindo código e ativos de imagem. O **Android Studio** oferece muito; você pode ver o que clicando na seta à esquerda dos itens no **Project Navigator**. Por enquanto, não se preocupe muito com esses arquivos; você se familiarizará bem com eles conforme avançar neste curso.

Configurando o Emulador



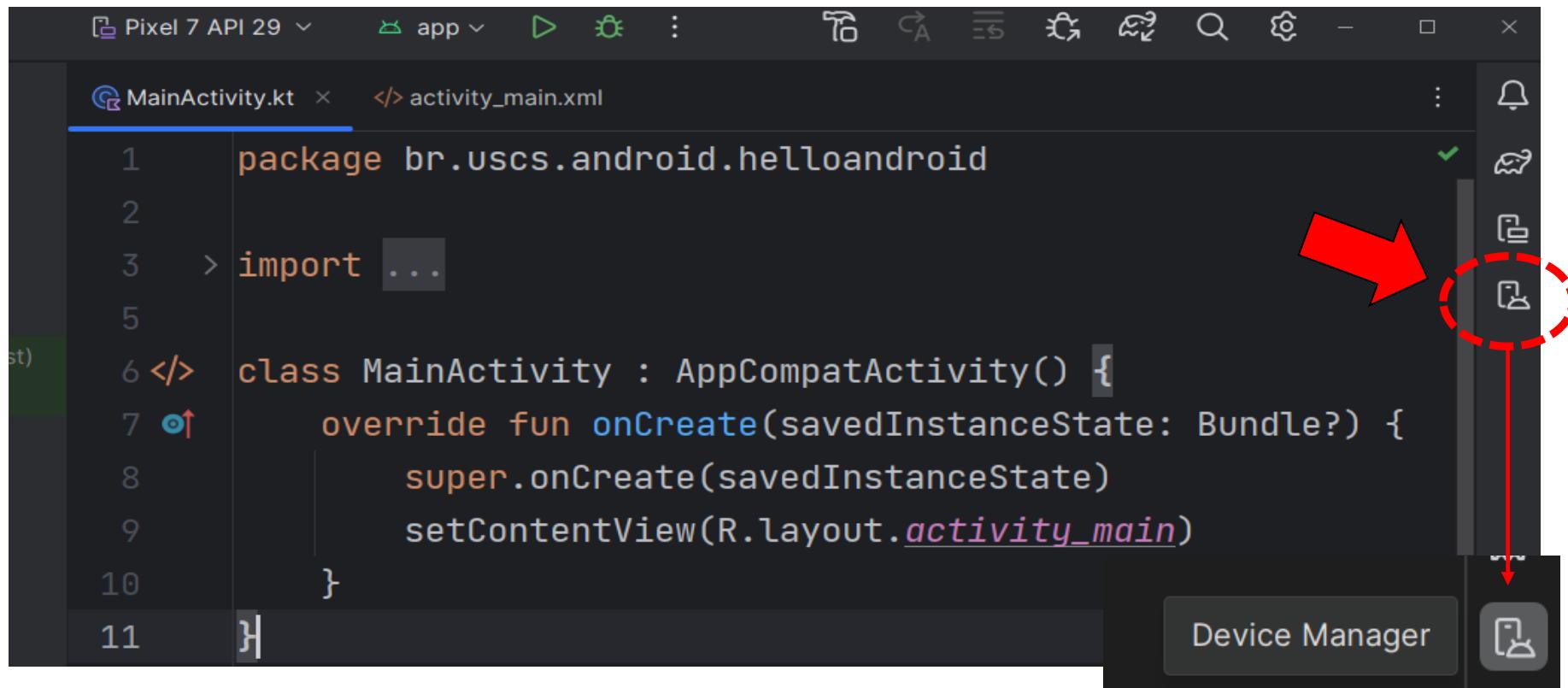
Criando um Device Virtual

- ⊕ Estudar a IDE é recomendado, 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.



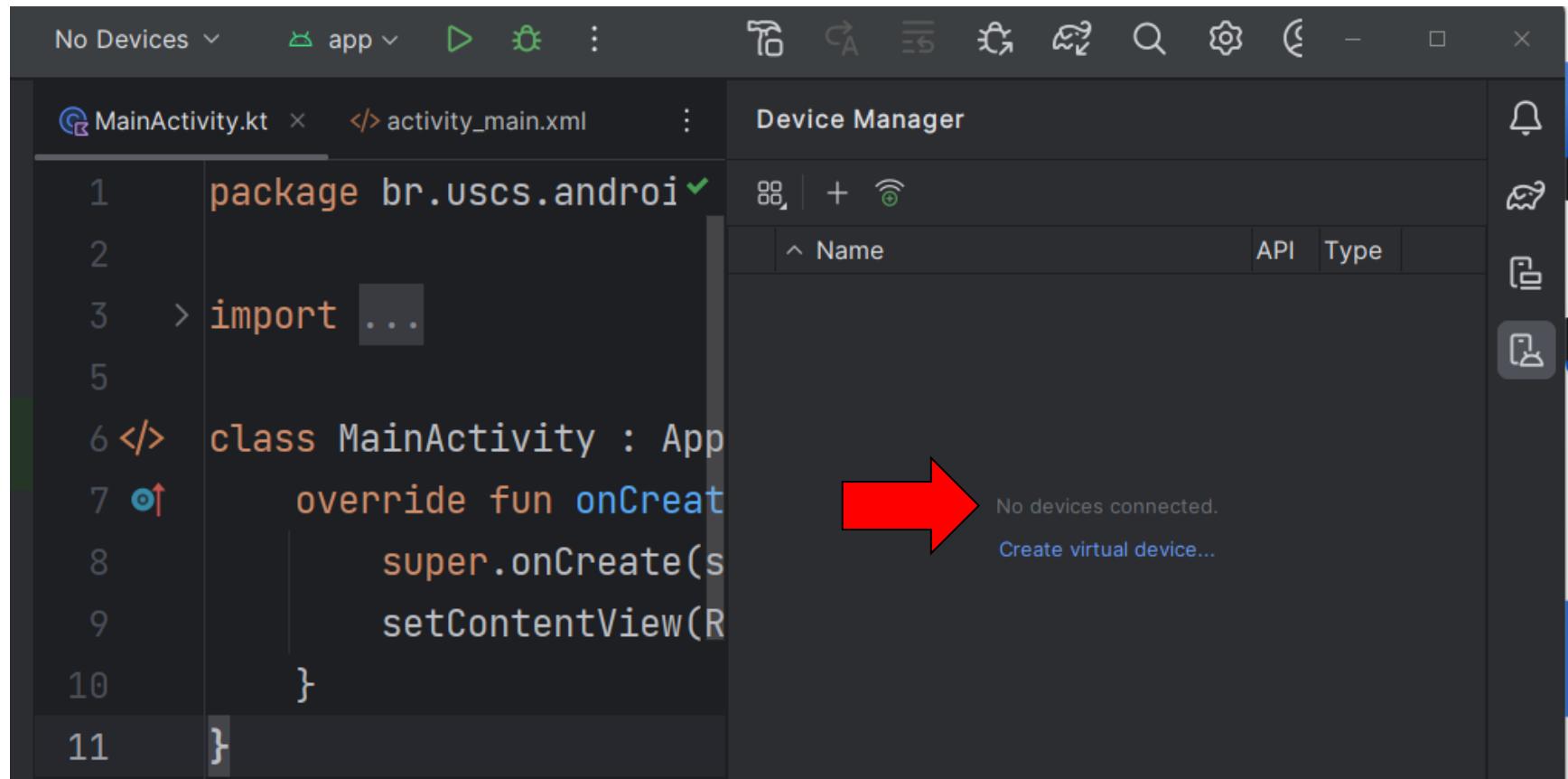
AVD Manager

- ⊕ 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.



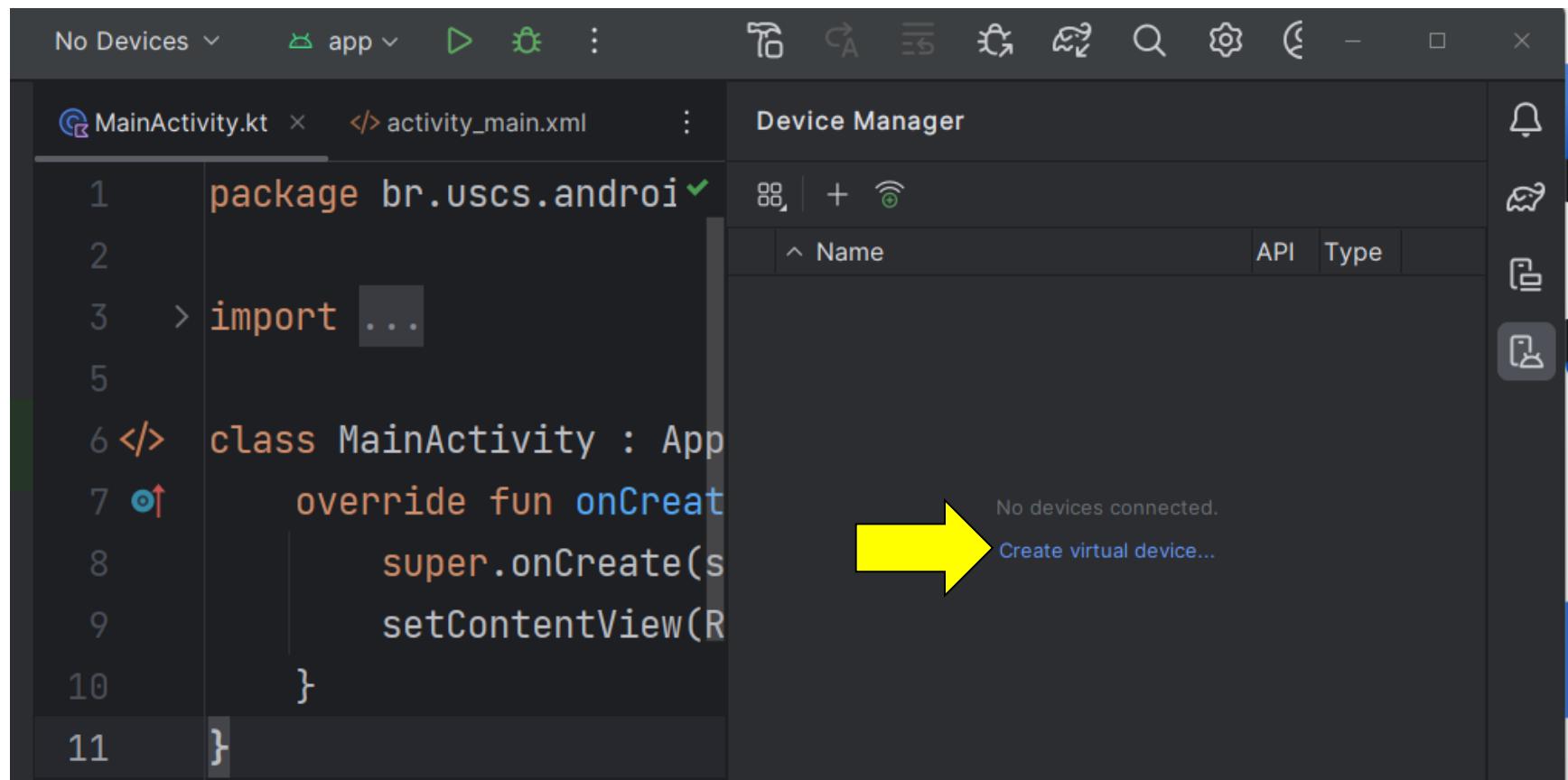
AVD Manager

- ⊕ 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**.



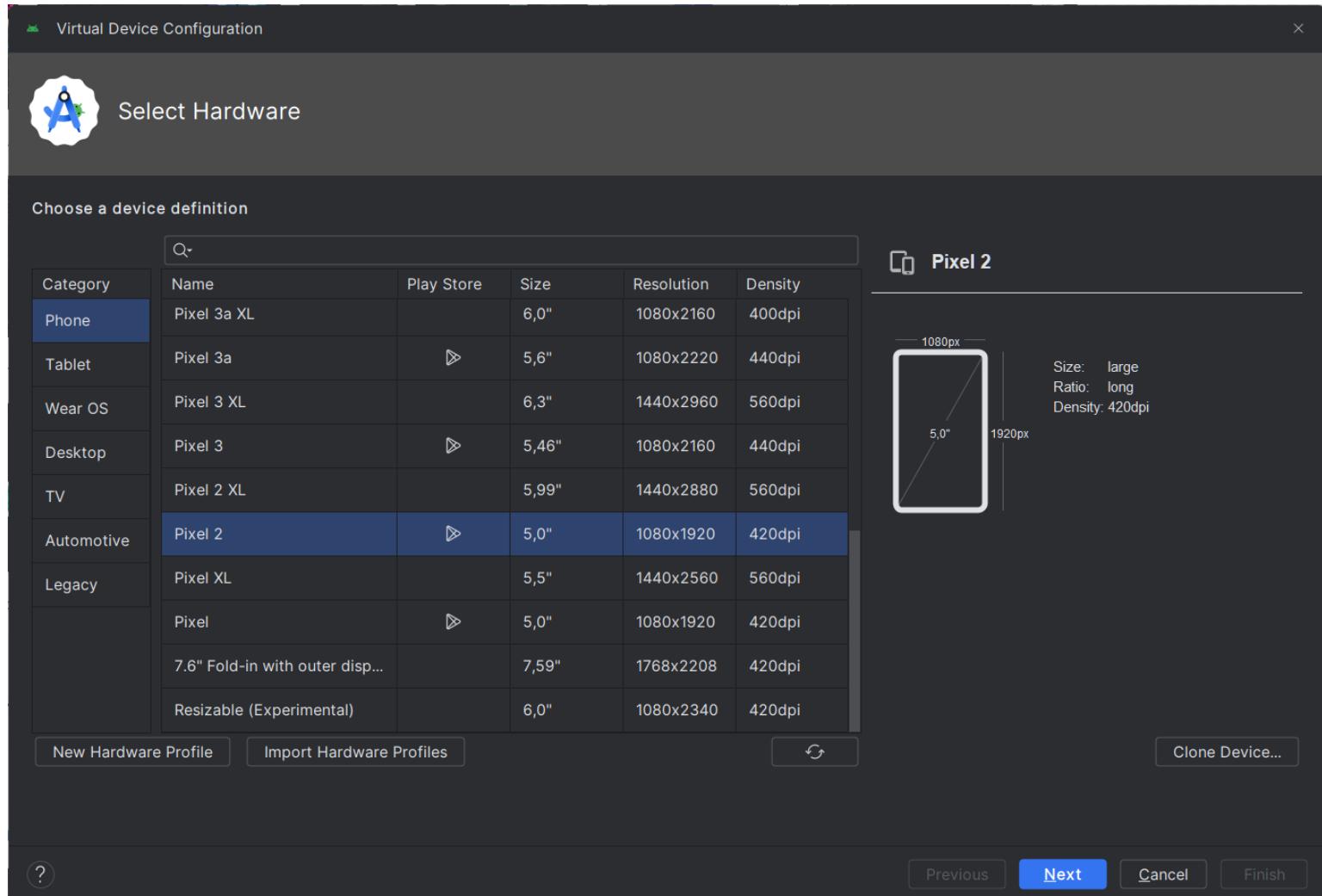
AVD Manager

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



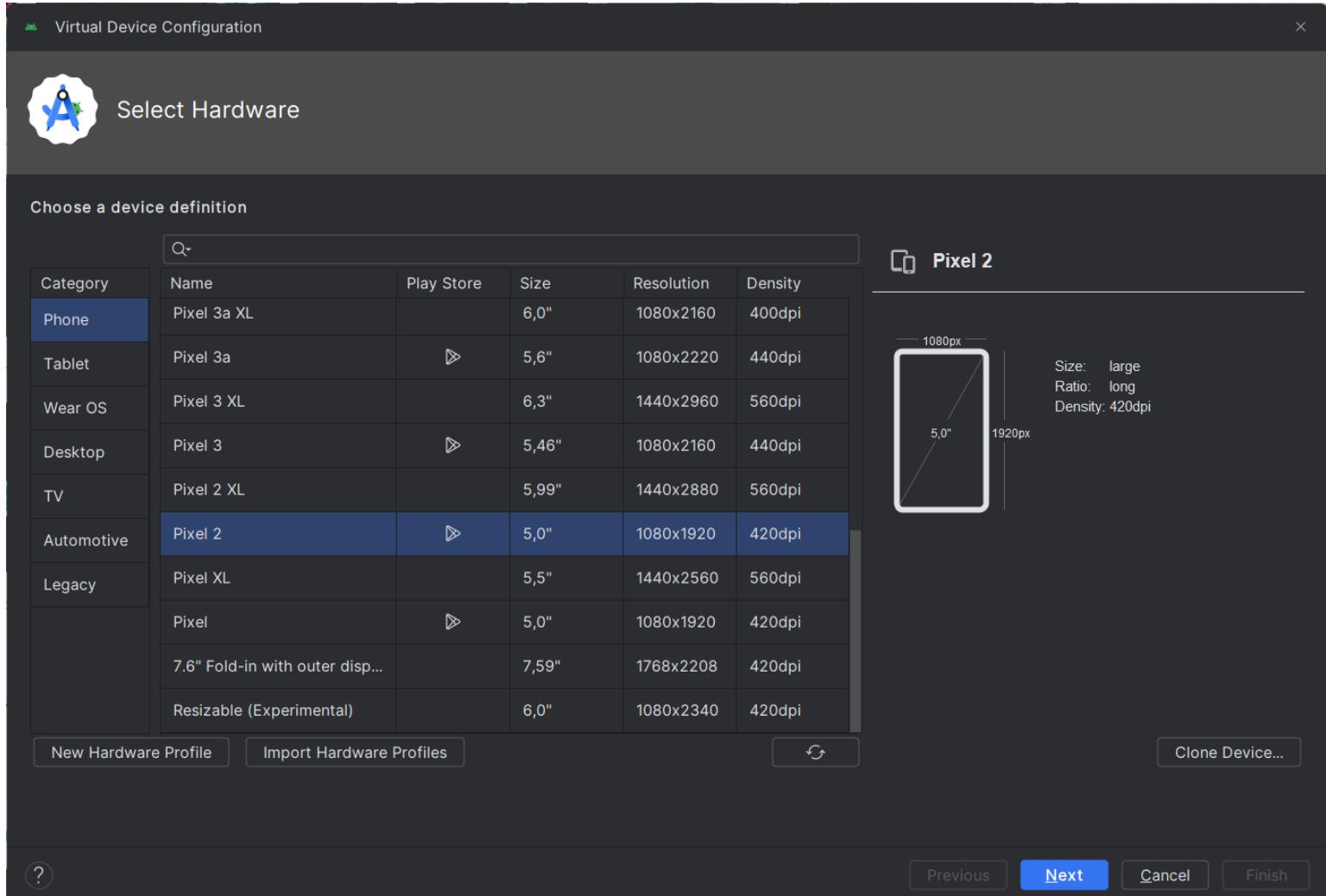
AVD Manager

- 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.





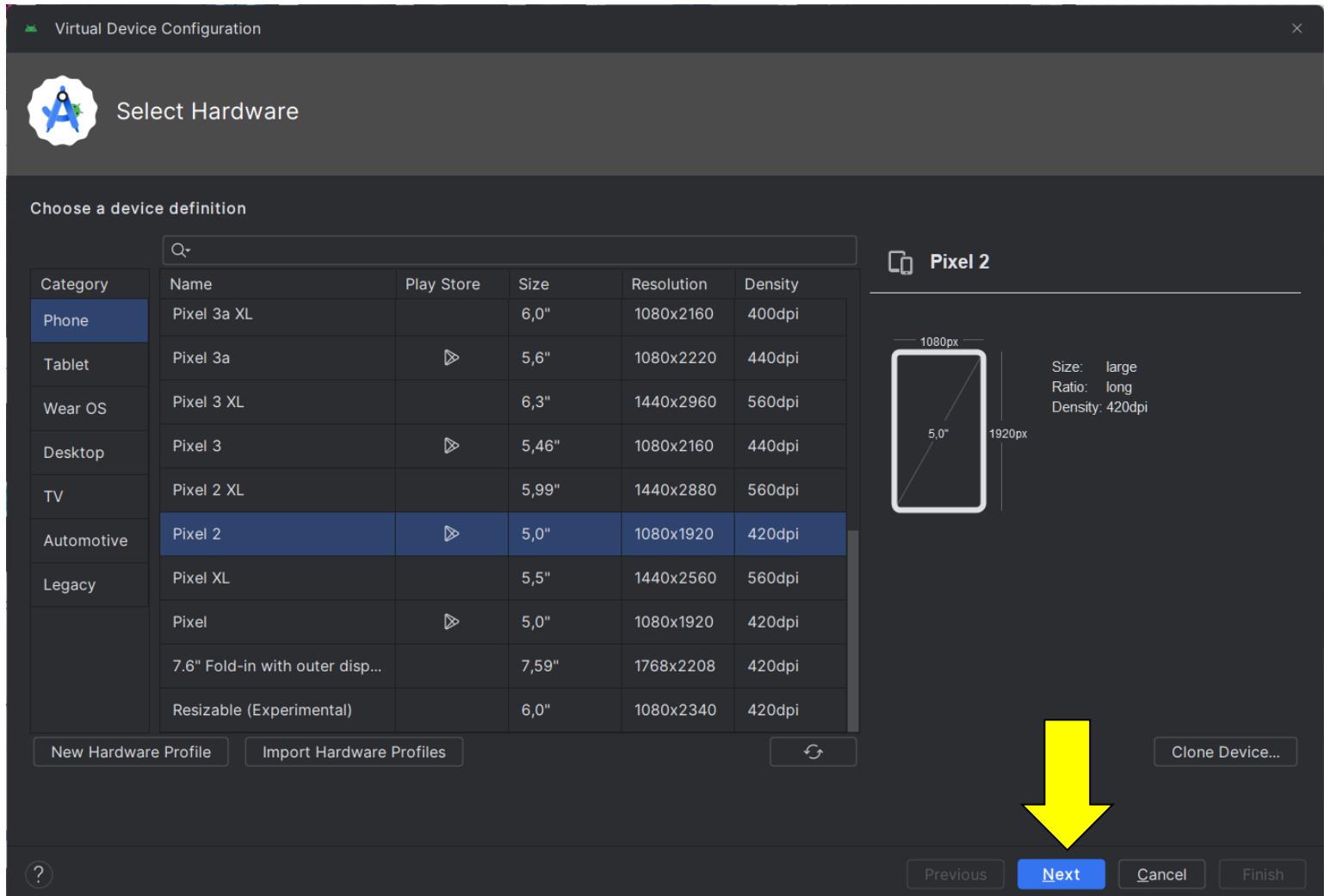
Porque Pixel 2 ?

Porque Pixel 2 ?

- ⊕ **Pixel 2** é uma escolha recomendada para configurar o emulador no **Android Studio**, especialmente por alguns motivos técnicos e de compatibilidade;
- ⊕ O **Pixel 2** oferece um conjunto de funcionalidades e versões do **Android** que são amplamente suportadas no Android Studio e é frequentemente utilizado pela equipe de desenvolvimento do Google como um dispositivo de referência;
- ⊕ O **Pixel 2** possui uma resolução de 1080x1920 pixels, o que é um padrão comum de tela para muitos dispositivos Android.
- ⊕ Ele possui boas configurações para emulação de diferentes características de hardware, como câmeras, sensores;
- ⊕ O **Pixel 2** possui configurações de **RAM** que permitem rodar o emulador em máquinas de desenvolvimento sem sobrecarregar tanto o sistema, o que é útil para quem não possui um hardware extremamente poderoso.

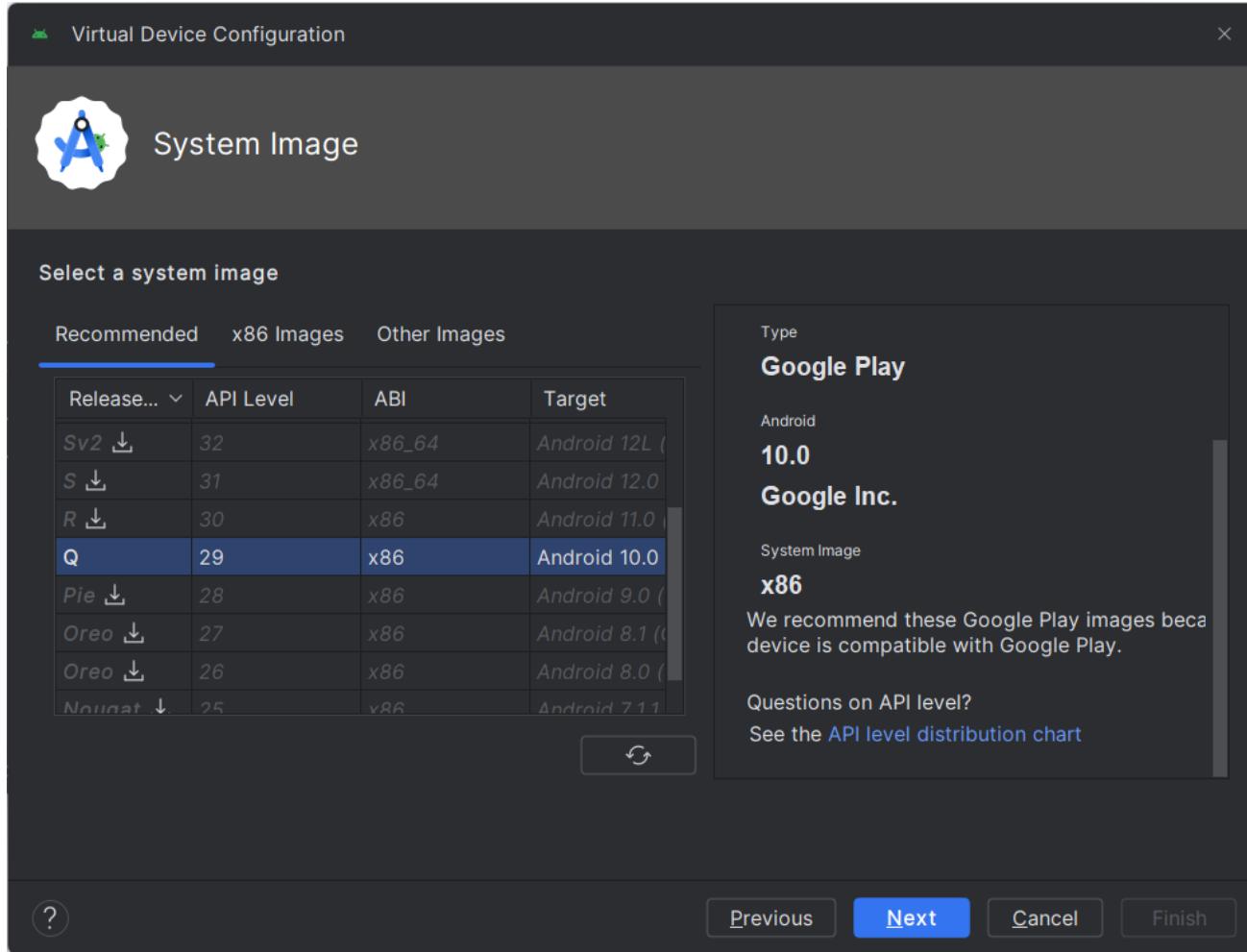
AVD Manager

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



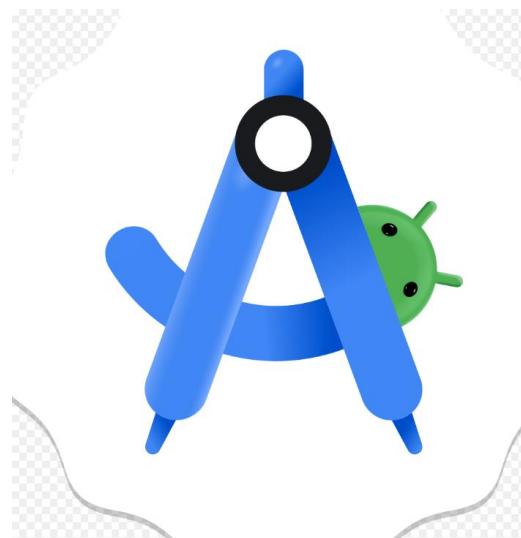
AVD Manager

- ⊕ 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

Virtual Device Configuration

System Image

Select a system image

Recommended x86 Images Other Images

Releas...	API	ABI	xABI	Target
VanillaIceCream	x86_64			Android 15
UpsideDownCak...	x86_64			Android 15
TiramisuPrivacy...	x86_64			Android 14
VanillaIceCream	35	x86_64		Android 15
VanillaIceCream	35	x86_64		Android 15
UpsideDownCak...	34	x86_64		Android 14
TiramisuPrivacy...	33	x86_64		Android 13
Sv2	32	x86_64		Android 12
S	31	x86_64		Android 12
R	30	x86		Android 11
Q	29	x86		Android 10
Pie	28	x86		Android 9.
Oreo	27	x86		Android 8

?

Previous Next Cancel Finish

Q

API Level
29

Type
Google Play

Android
10.0

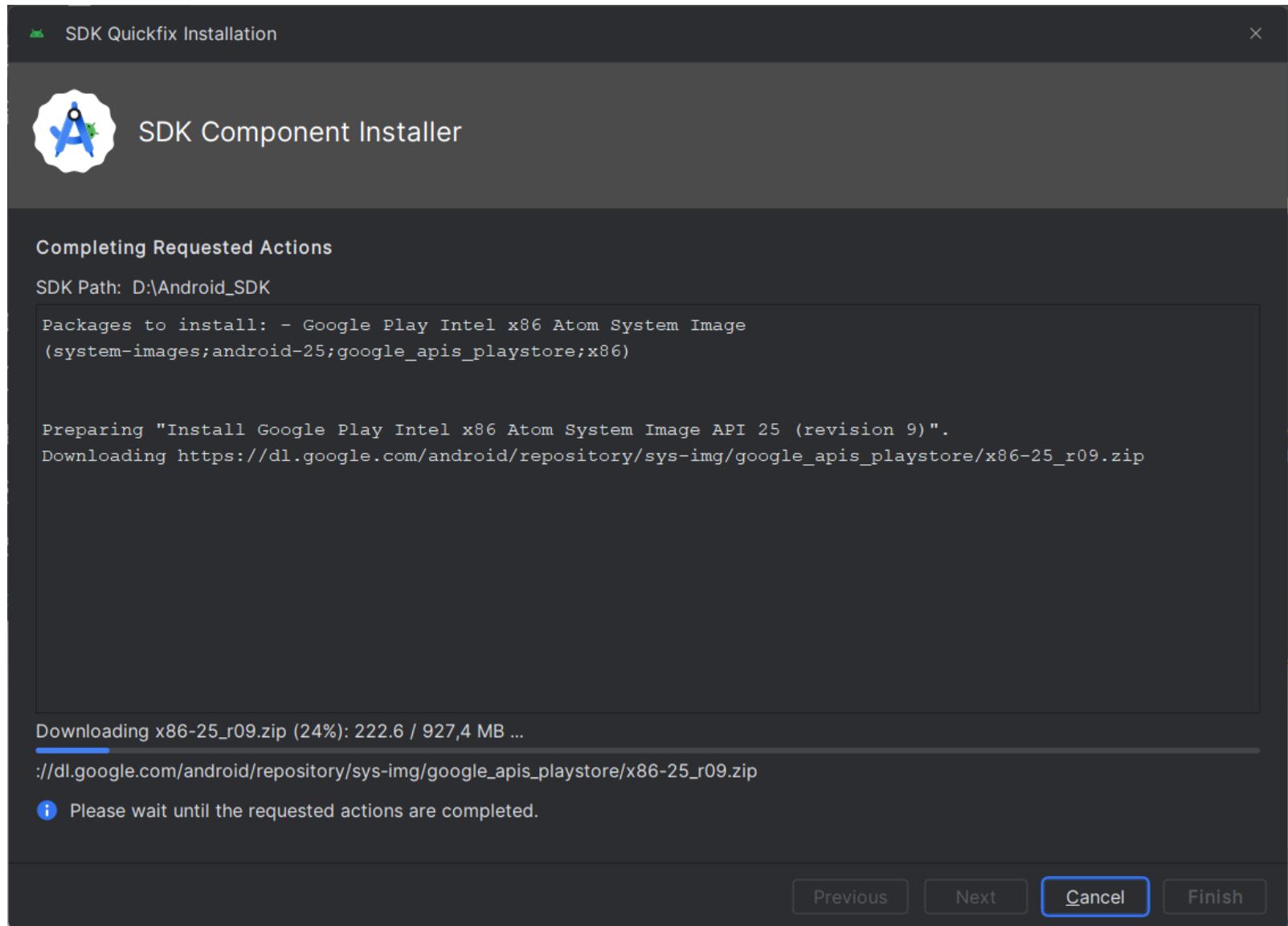
Google Inc.

System Image
x86

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

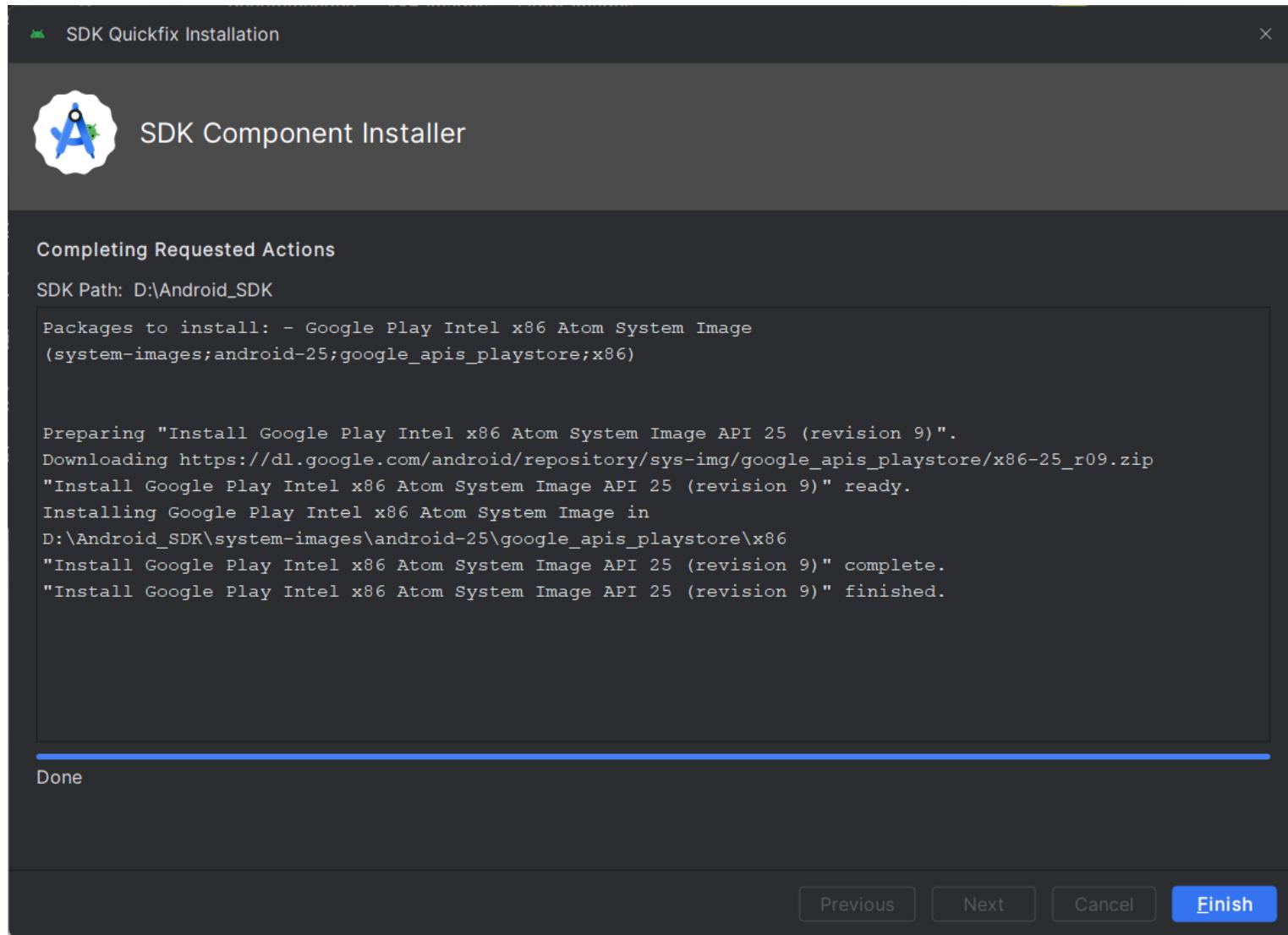
Questions on API level?
See the [API level distribution chart](#)

AVD Manager



AVD Manager

- ⊕ Ao encerrar o **download**, clique em **Finish**.



AVD Manager

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

A Virtual Device Configuration X

 System Image

Select a system image

Recommended x86 Images Other Images

Releas...	API	ABI	xABI	Target
Vani... ↴	35	x86_64		Android 15
Upsi... ↴	34	x86_64		Android 14
Tira... ↴	33	x86_64		Android 13
Sv2 ↴	32	x86_64		Android 12
S ↴	31	x86_64		Android 11
R ↴	30	x86		Android 10
Q	29	x86		Android 10
Pie ↴	28	x86		Android 9.

↻

Q

API Level
29

Type
Google Play

Android
10.0

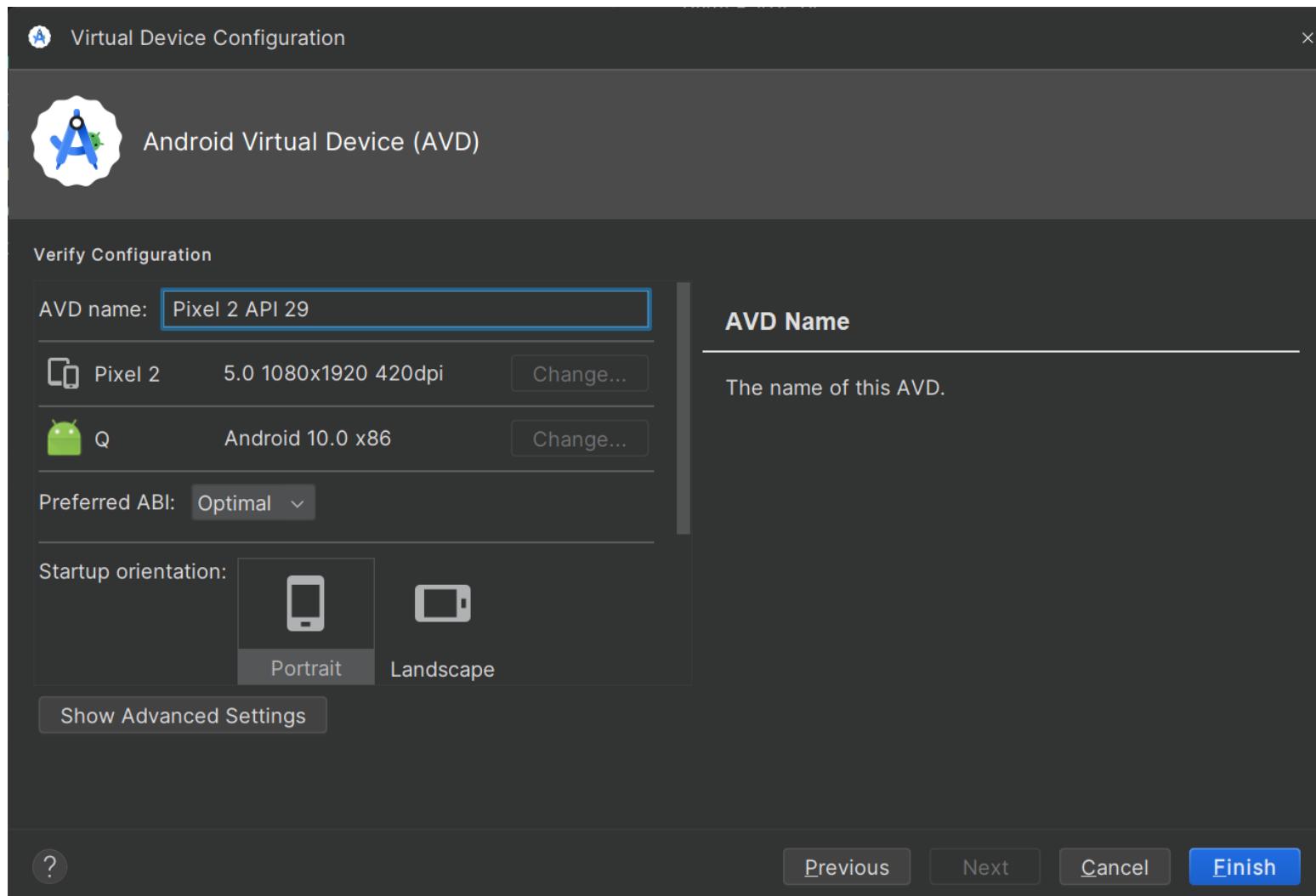
Google Inc.

System Image
x86

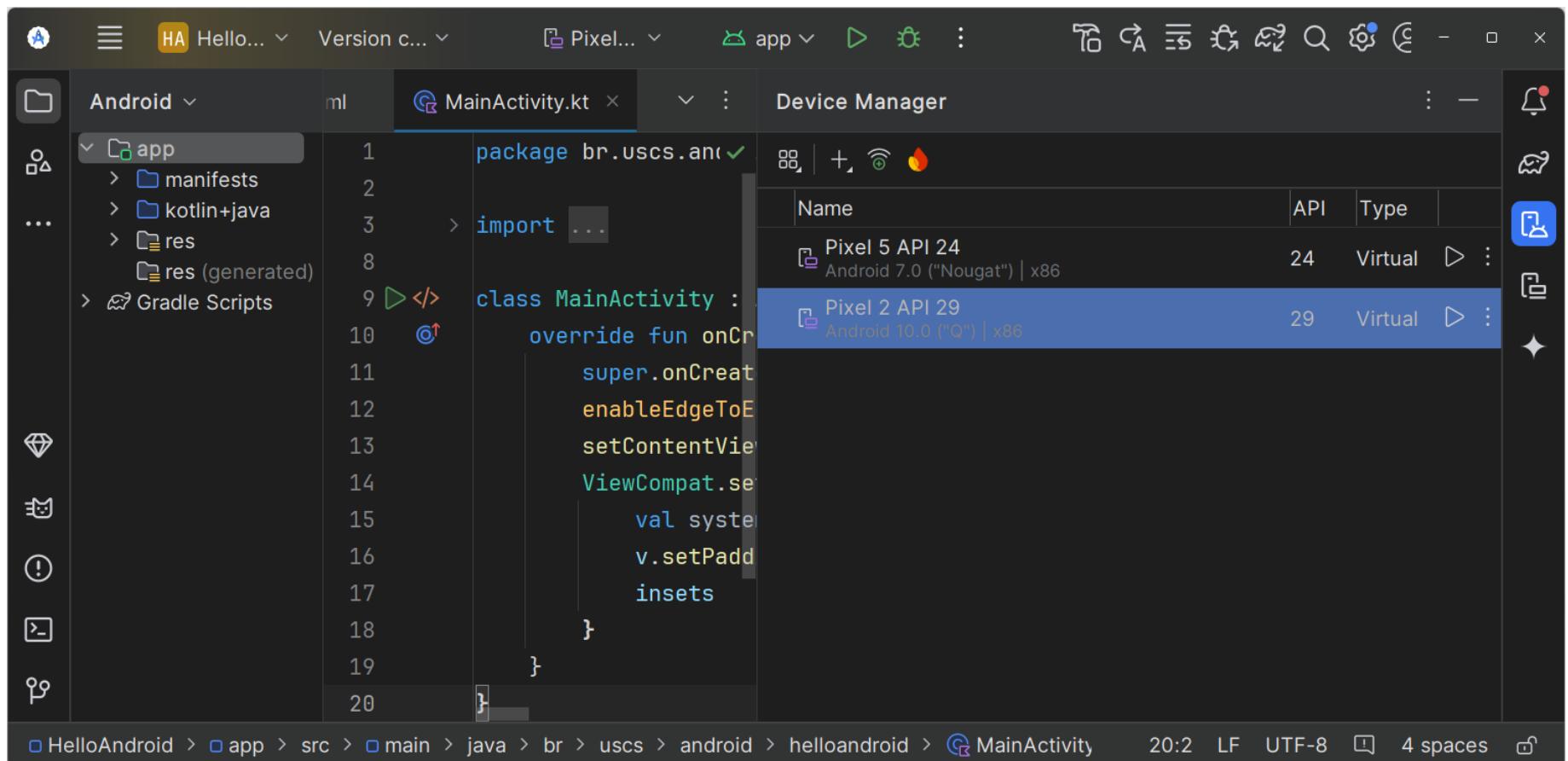
? Previous Next Cancel Finish

AVD Manager

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



AVD Manager



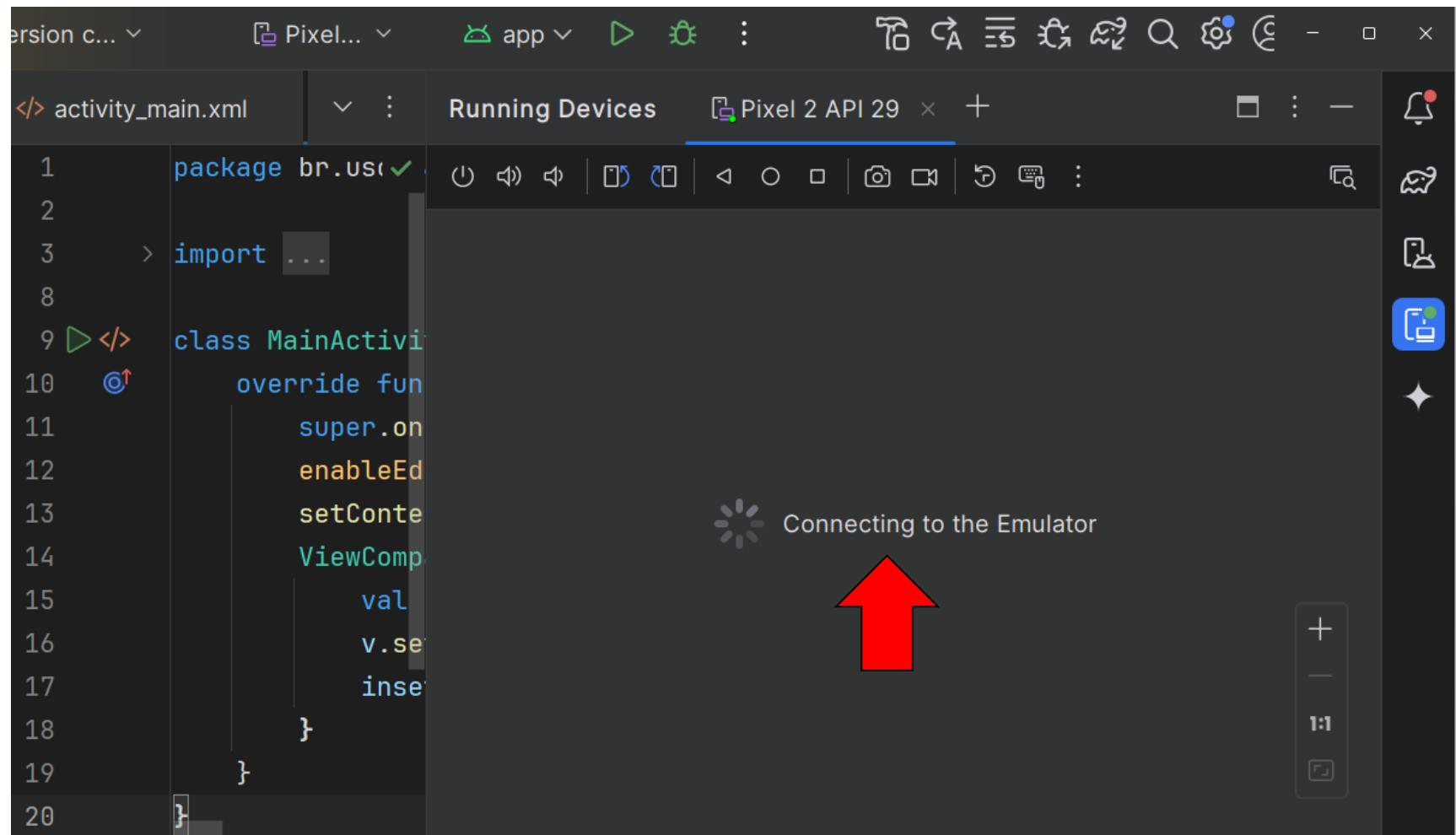
The screenshot shows the Android Studio interface with the following details:

- Title Bar:** Shows "HelloAndroid" as the project name, "Version c..." as the branch, and "Pixel 5 API 24" as the selected device.
- Side Navigation:** Includes icons for Home, Recent Projects, and various settings.
- Main Area:**
 - Device Manager:** A table listing available devices:

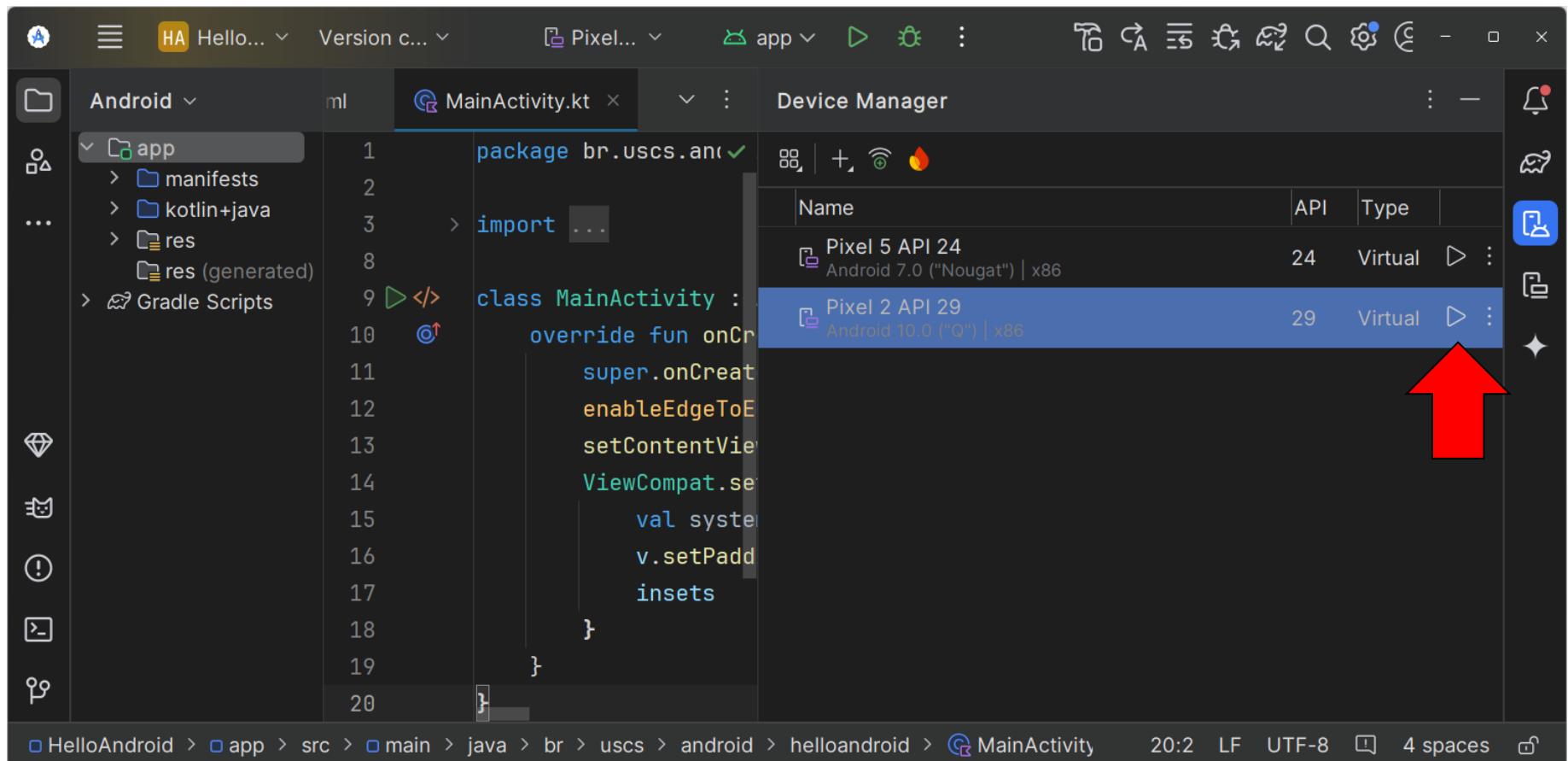
Name	API	Type
Pixel 5 API 24	24	Virtual
Pixel 2 API 29	29	Virtual
 - Code Editor:** Displays the `MainActivity.kt` file content:

```
package br.uscs.android
import ...
class MainActivity : ...
override fun onCreate...
    super.onCreate...
    enableEdgeToE...
    setContentView...
    ViewCompat.set...
        val system...
        v.setPadd...
        insets...
    }
}
}
```
- Bottom Bar:** Shows the file path "HelloAndroid > app > src > main > java > br > uscs > android > helloandroid > MainActivity", and status indicators like "20:2 LF UTF-8 4 spaces".

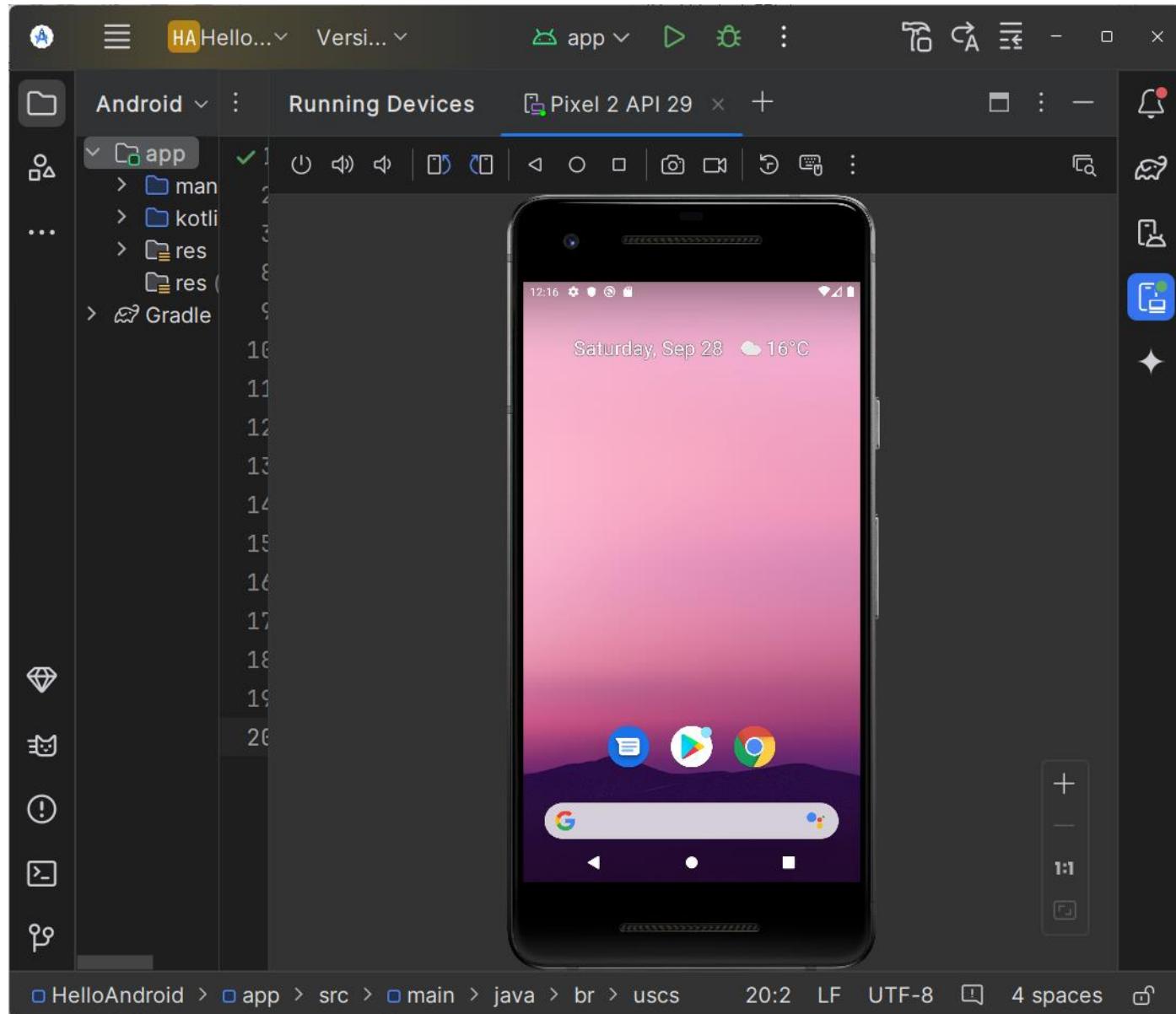
AVD Manager - Start



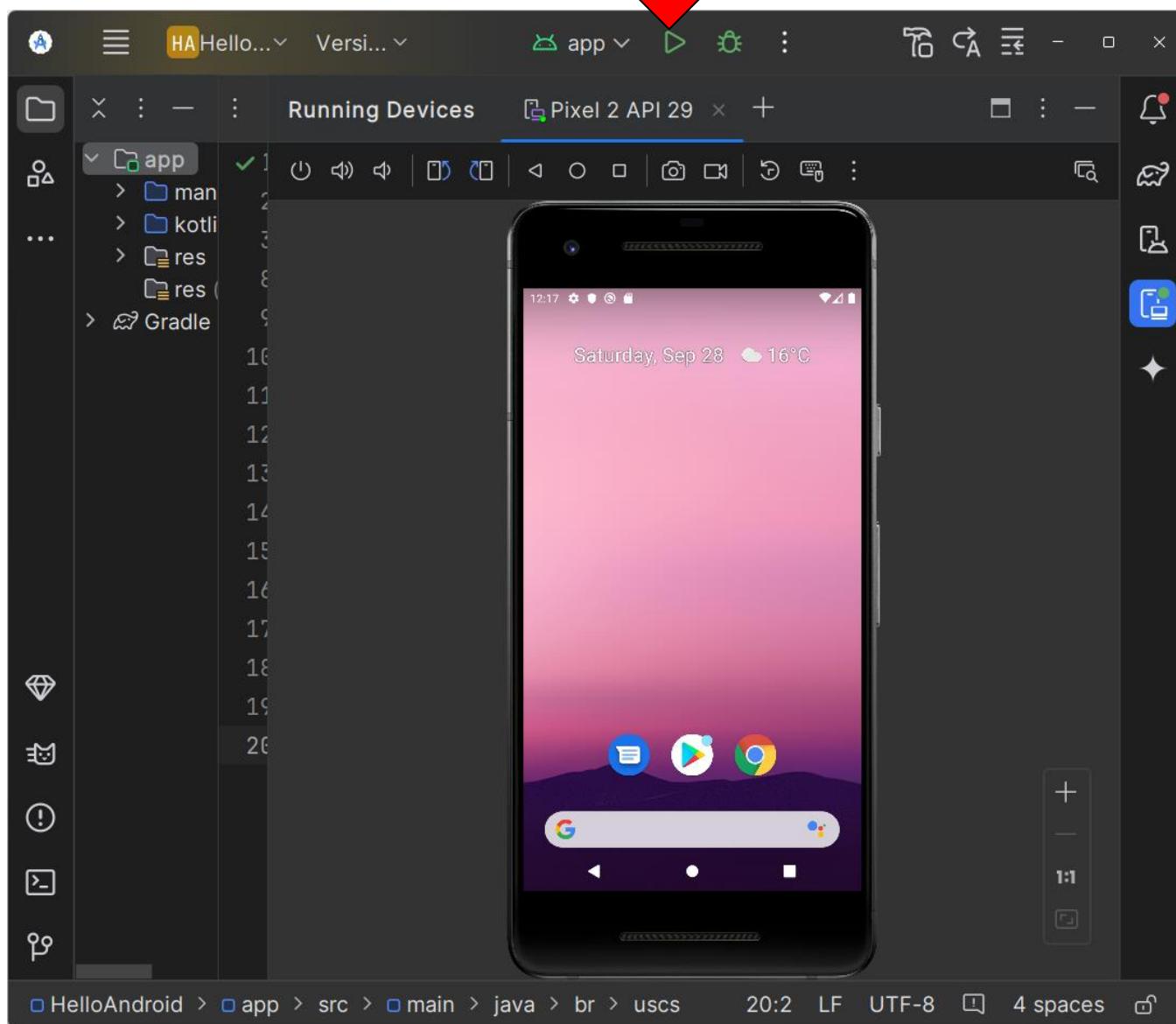
AVD Manager - Start



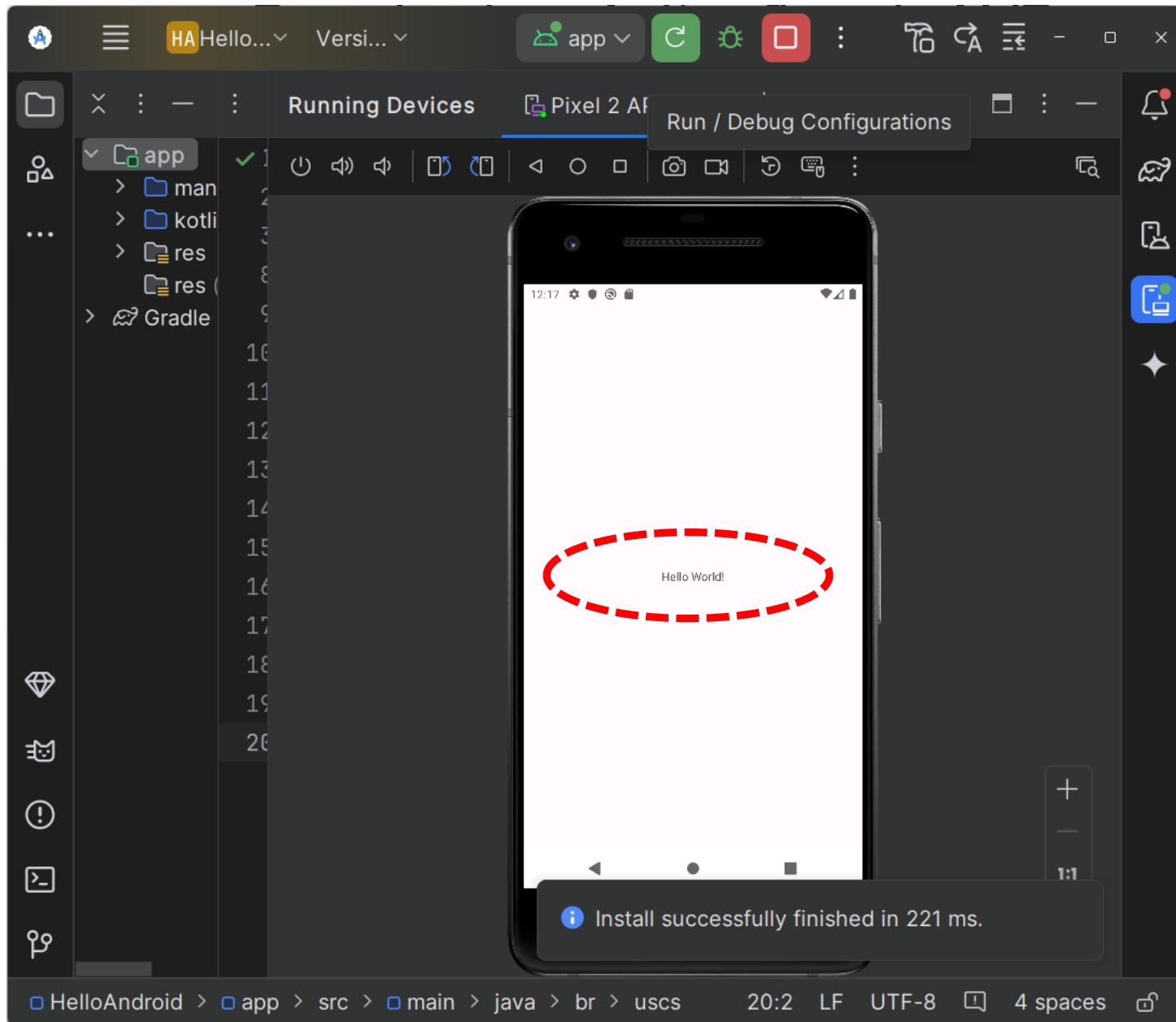
AVD Manager – running



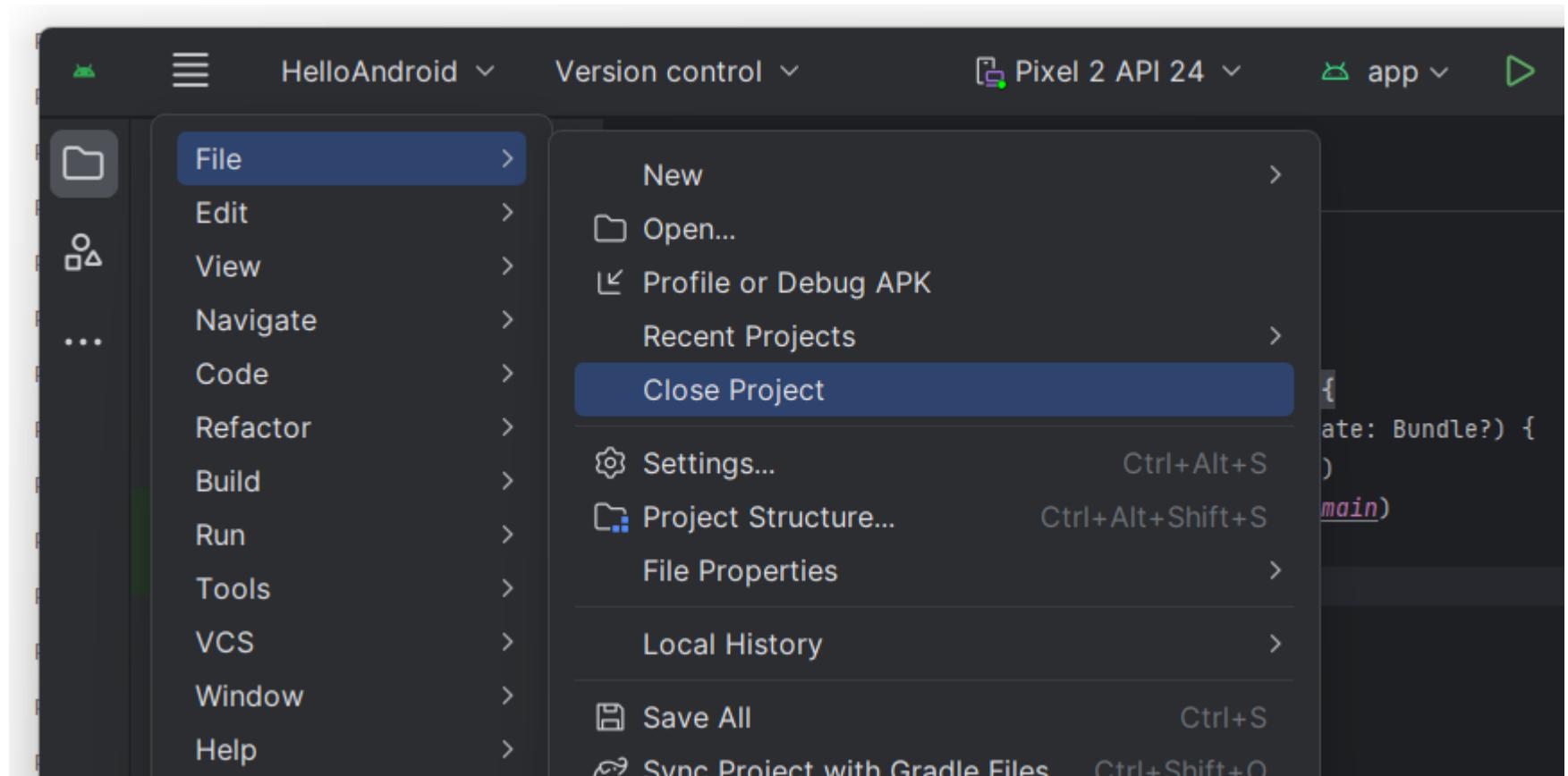
Executando a Aplicação sob AVD



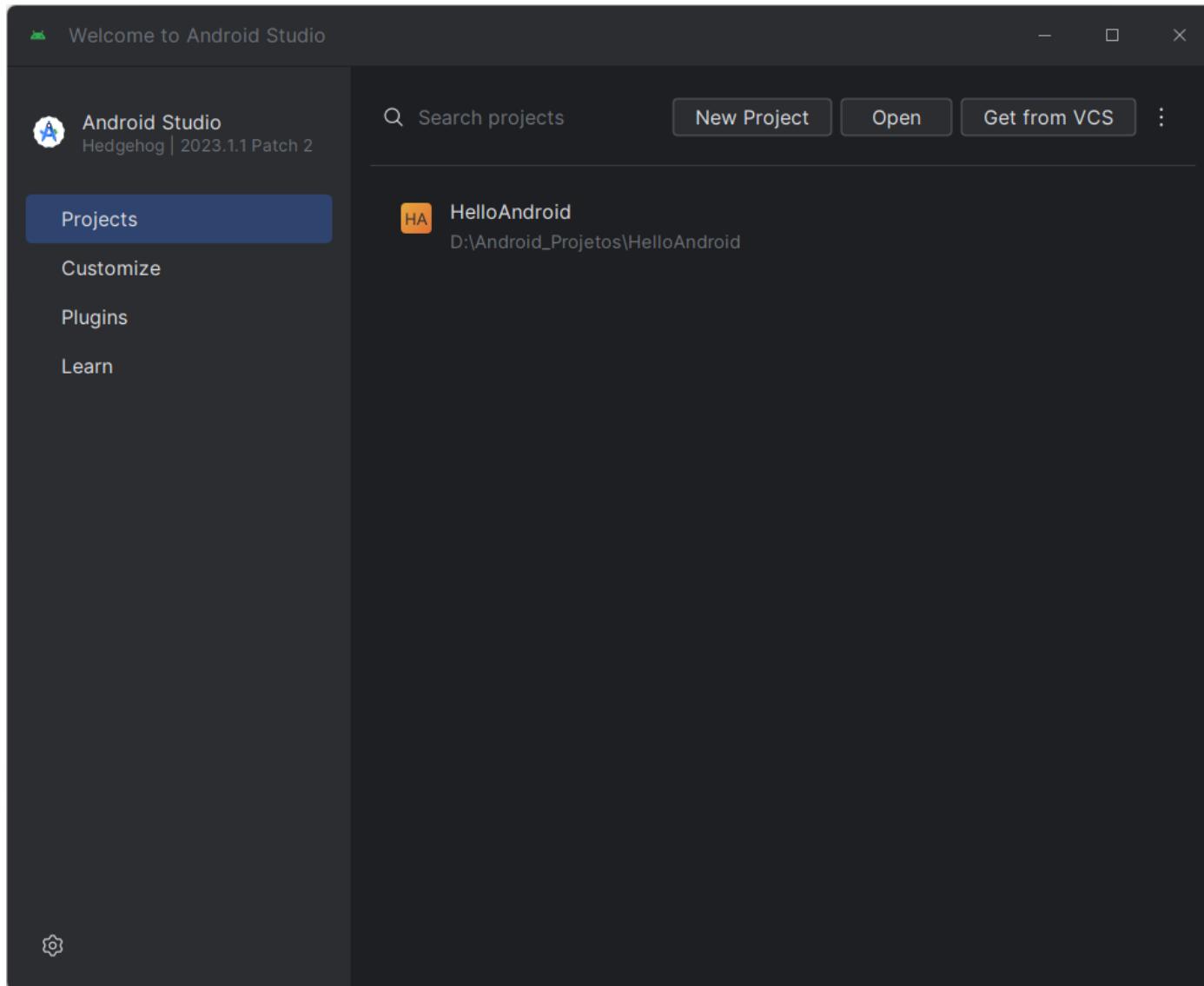
Executando a Aplicação sob AVD



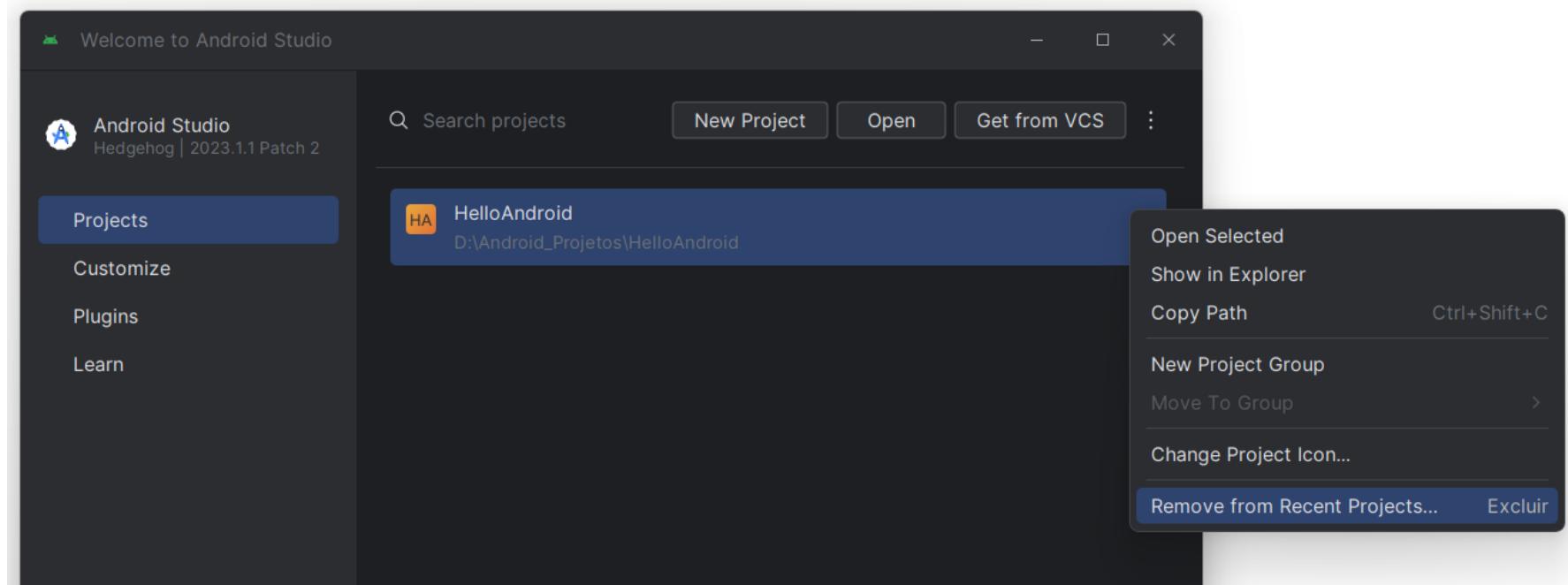
Fechando o Projeto



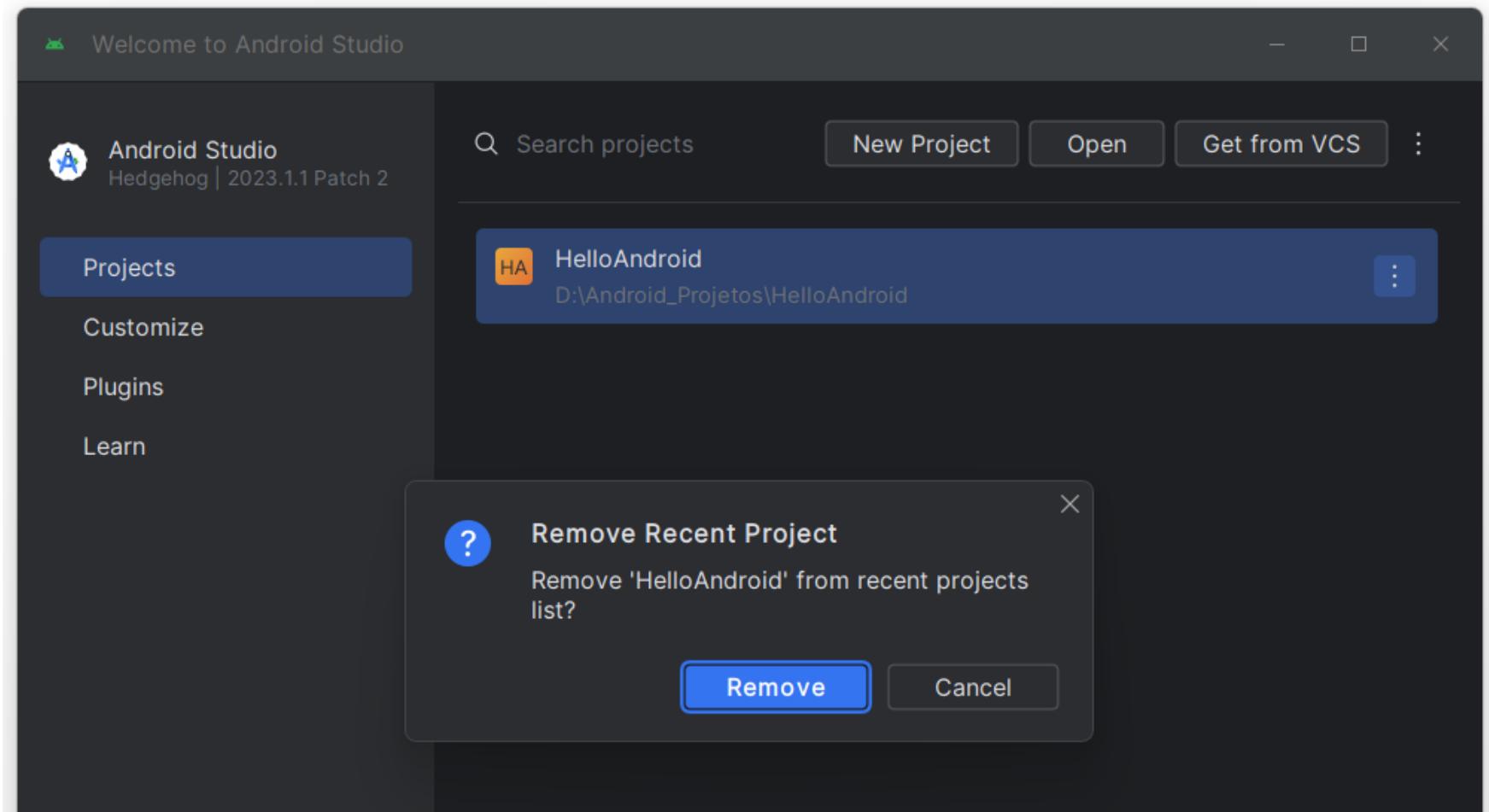
Fechando o Projeto



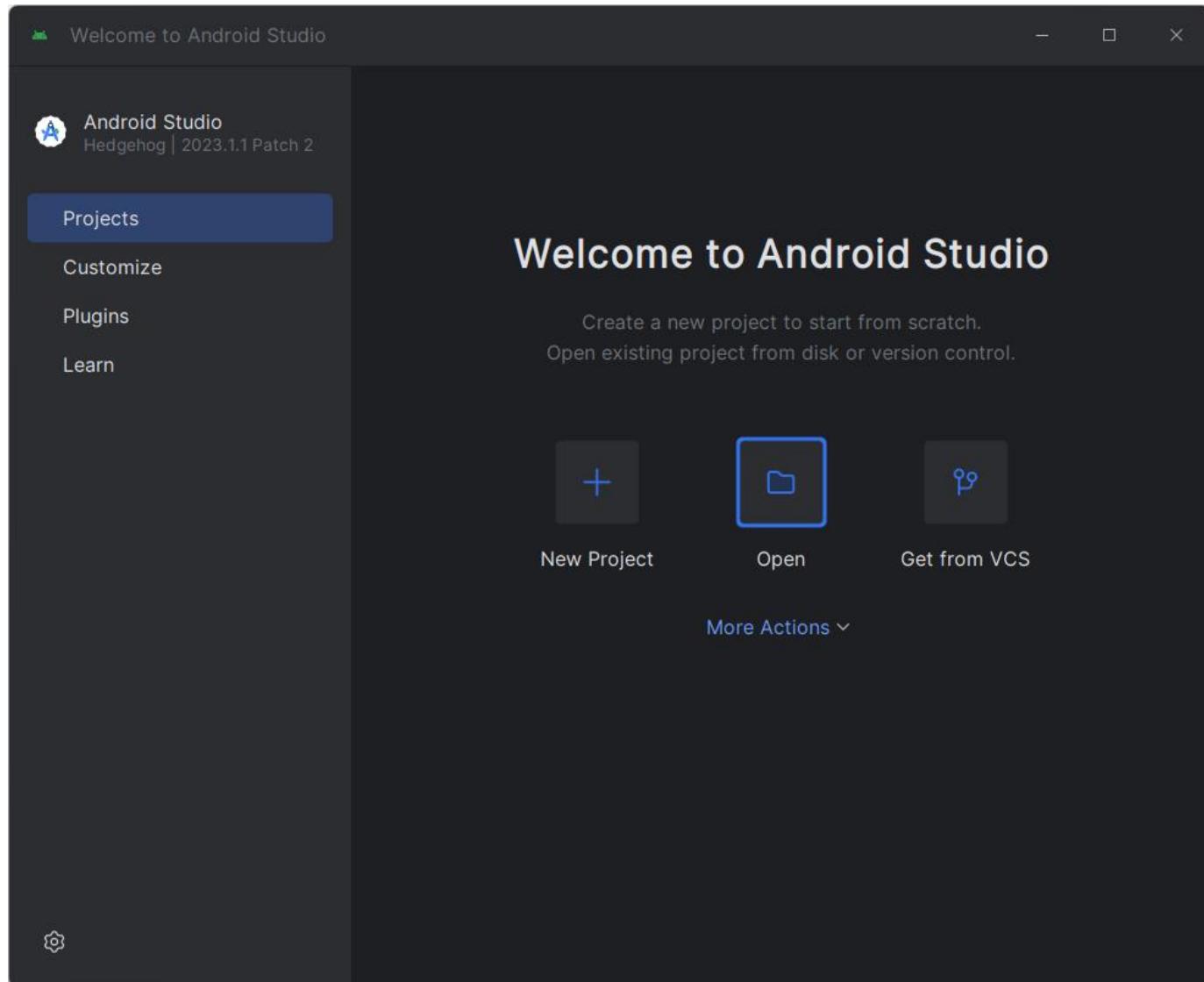
Retirando da Lista de Projetos Recentes



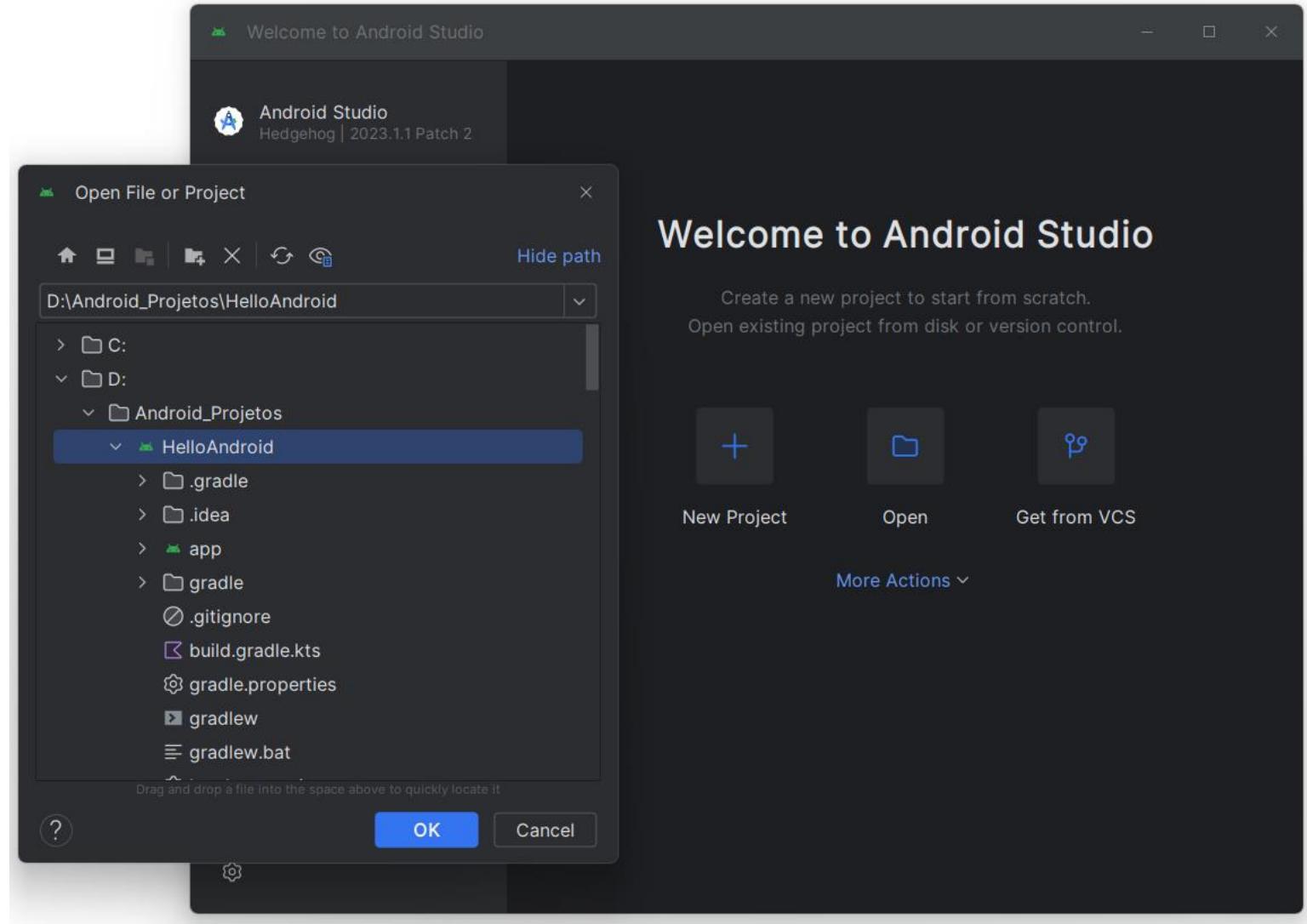
Retirando da Lista de Projetos Recentes



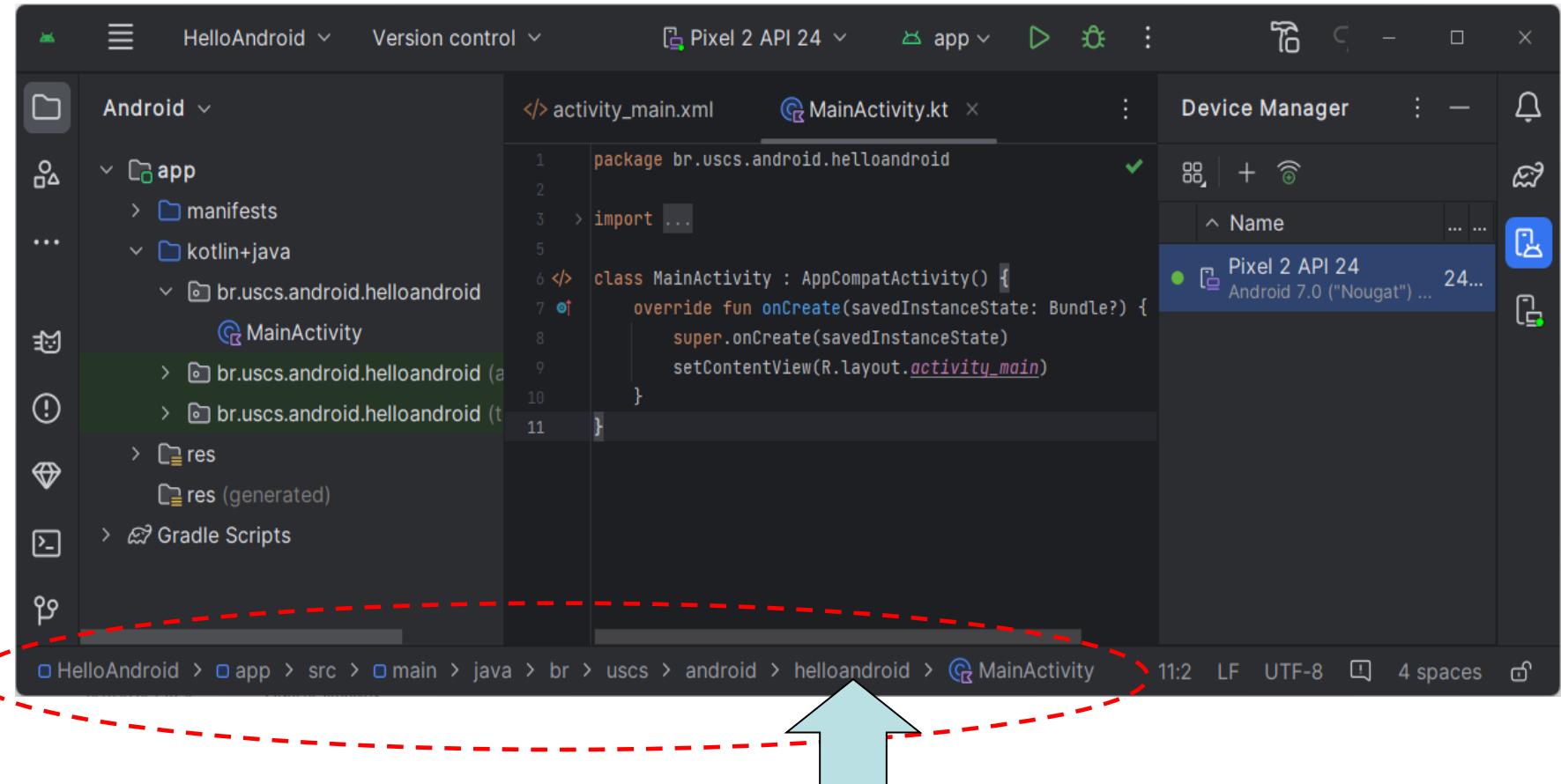
Retirando da Lista de Projetos Recentes



Reabrindo o Projeto



Reabrindo o Projeto



Reprocessando o Projeto

