
soluções mobile

prof. Thyerri Mezzari





Android Studio, projetos, ferramentas e Kotlin x Java



@thyerrimezzari

Grupo WhatsApp Mobile Seg





Lembrete: Nativo x Híbrido

Desenvolvimento híbrido consiste em utilizar linguagens não nativas (JavaScript, C#, Dart, etc) e transformar esse código em **código nativo** de cada sistema operacional, ou utilizar uma espécie de bridge (ponte) entre o código escrito nessa linguagem e a máquina virtual que executa os aplicativos que rodam em cima do sistema operacional utilizado.

Lembrando que as linguagens nativas de desenvolvimento mobile são: Java, Kotlin, Objective-C e Swift.



desenvolvimento para Android (nativo)

A principal ferramenta para criação de apps Android de forma nativa é o Android Studio.

Até é possível trabalhar com outras IDEs, editar os arquivos Java ou Kotlin e compilar/testar seu app usando diretamente a SDK do Android através da linha de comando, porém em quanto maior for o seu app, maior será o trabalho de seguir esse fluxo.

É neste momento que surge a devida necessidade de organização e agilidade de testes de seu app que o Android Studio comprovadamente se mostra com a principal ferramenta do ecossistema.



Android Studio

O Android Studio é o ambiente de desenvolvimento integrado (IDE) baseado no IntelliJ IDEA para o desenvolvimento de apps para Android.

Antigamente em suas primeiras versões o mesmo era baseado na IDE Java ECLIPSE.

Além do editor de código e das ferramentas de desenvolvedor avançadas do IntelliJ, o Android Studio oferece ainda mais recursos para aumentar sua produtividade na compilação de apps Android.

É a ferramenta oficial de desenvolvimento recomendada pela Google



Android Studio

Principais recursos focados em Android:

- Um sistema de compilação flexível baseado em Gradle
- Um emulador rápido (?) com inúmeros recursos
- Um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos Android
- A aplicação de alterações para enviar alterações de código e recursos ao aplicativo em execução sem reiniciar o aplicativo
- Ferramentas de lint para detectar problemas de desempenho, usabilidade, compatibilidade
- com versões, entre outros



Desempenho

Requirement	Minimum	Recommended
OS	64-bit Microsoft Windows 8	Latest 64-bit version of Windows
RAM	8 GB RAM	16 GB RAM or more
CPU	x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor Framework .	Latest Intel Core processor
Disk space	8 GB (IDE and Android SDK and Emulator)	Solid state drive with 16 GB or more
Screen resolution	1280 x 800	1920 x 1080



baixando o Android Studio

O Android Studio está disponível para os principais sistemas operacionais do mercado, ou seja, você o encontra nas versões Windows, Mac e Linux.

Para baixar o Android Studio escolha a versão adequada para seu OS no link abaixo:

<https://developer.android.com/studio>



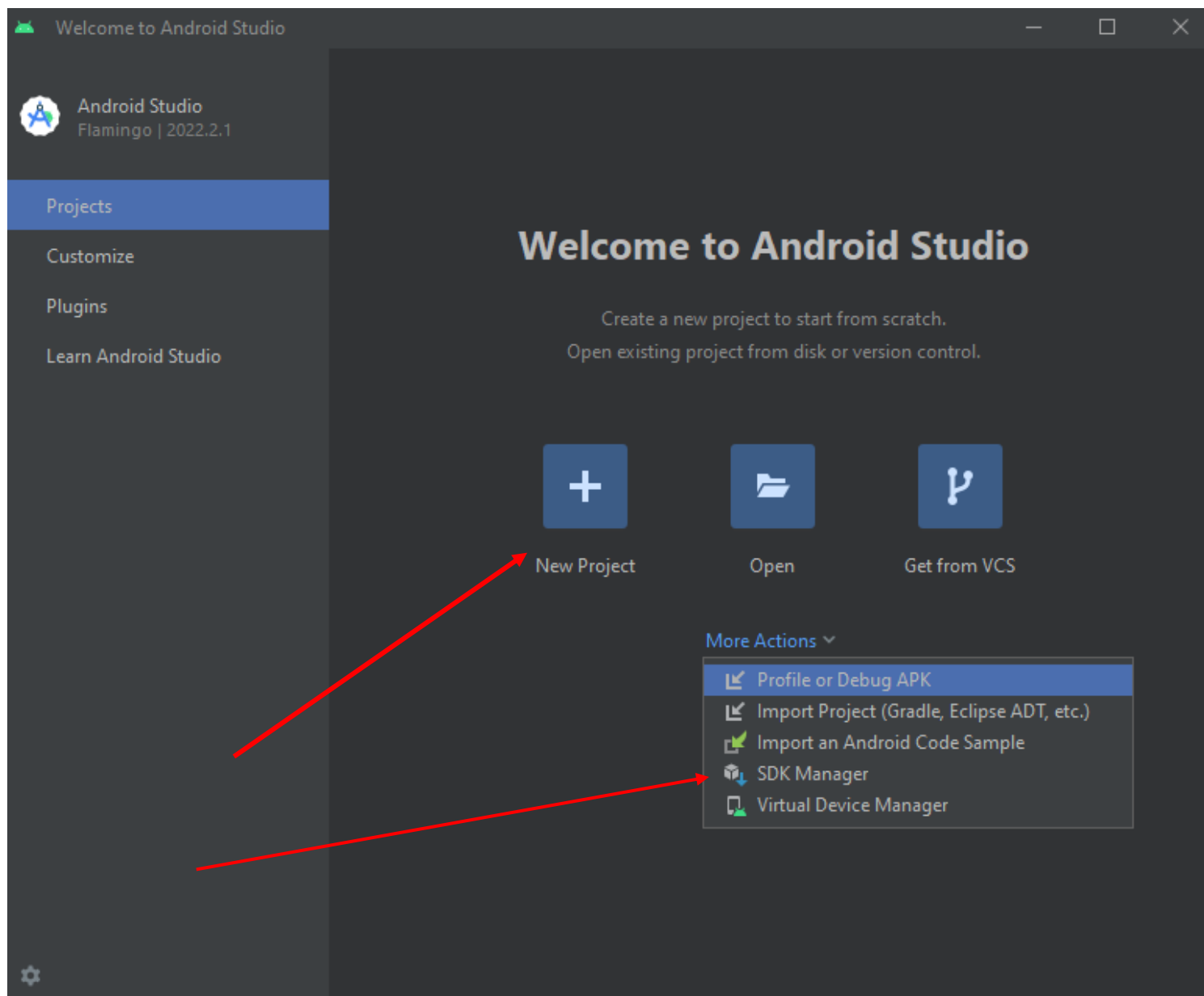
instalando o Android Studio

O processo de instalação é padrão, como qualquer outro programa de seu sistema operacional.

Fica como observação que nem sempre o que você baixa é o Android Studio final, durante o processo de instalação possivelmente outros pacotes adicionais serão baixados e isso pode demorar ainda mais.

Neste link existe um processo explicativo bem fácil sobre a instalação:

<https://developer.android.com/studio/install>





vale a pena dar uma lida

Configurações de Ambiente:

<https://developer.android.com/studio/intro/studio-config>

Atualizando o Android Studio:

<https://developer.android.com/studio/intro/update>

Fluxo de Trabalho:

<https://developer.android.com/studio/workflow>



Configurações Básicas

SDK

Kit de desenvolvimento de software, também conhecido como Software development kit, SDK ou "devkit", é tipicamente um conjunto de ferramentas de desenvolvimento de software que permite a criação de aplicativos.

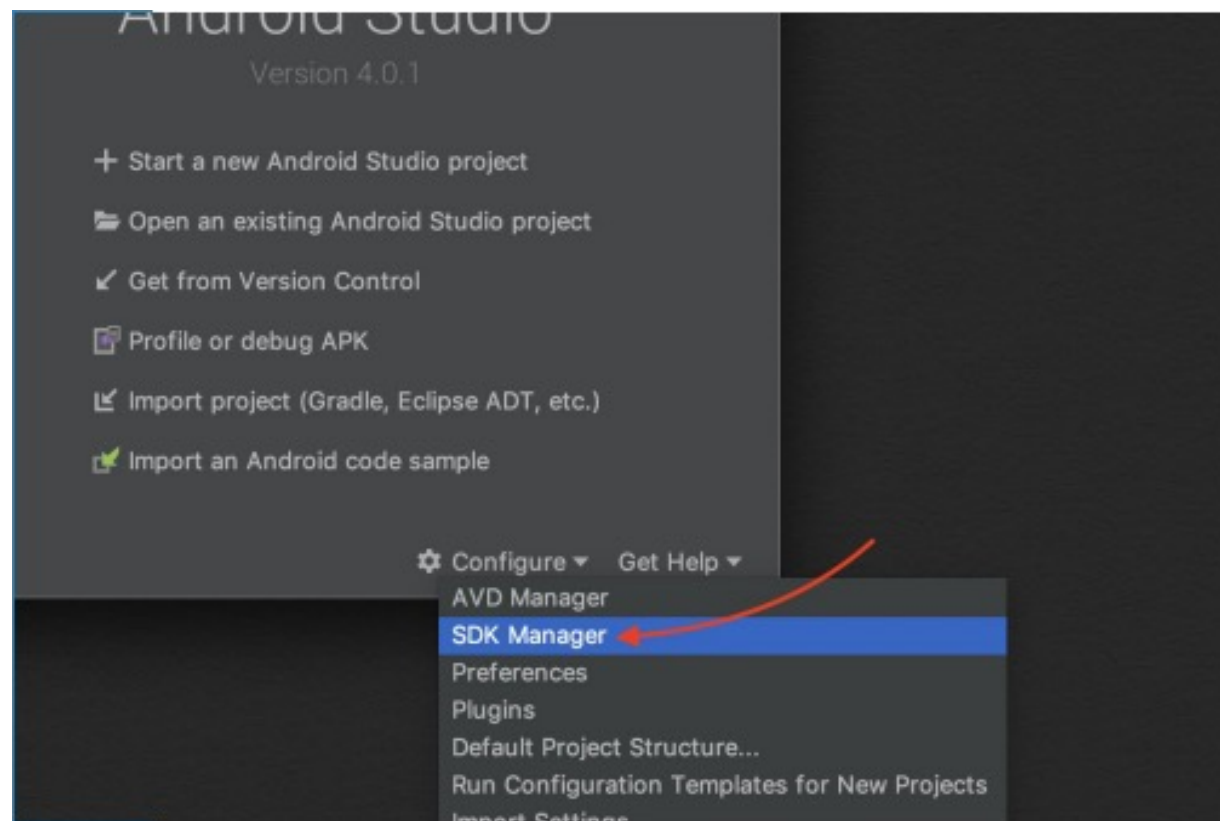
Android Studio != Android SDK.

Você pode baixar a SDK do Android separada do Android Studio



Configurações Básicas

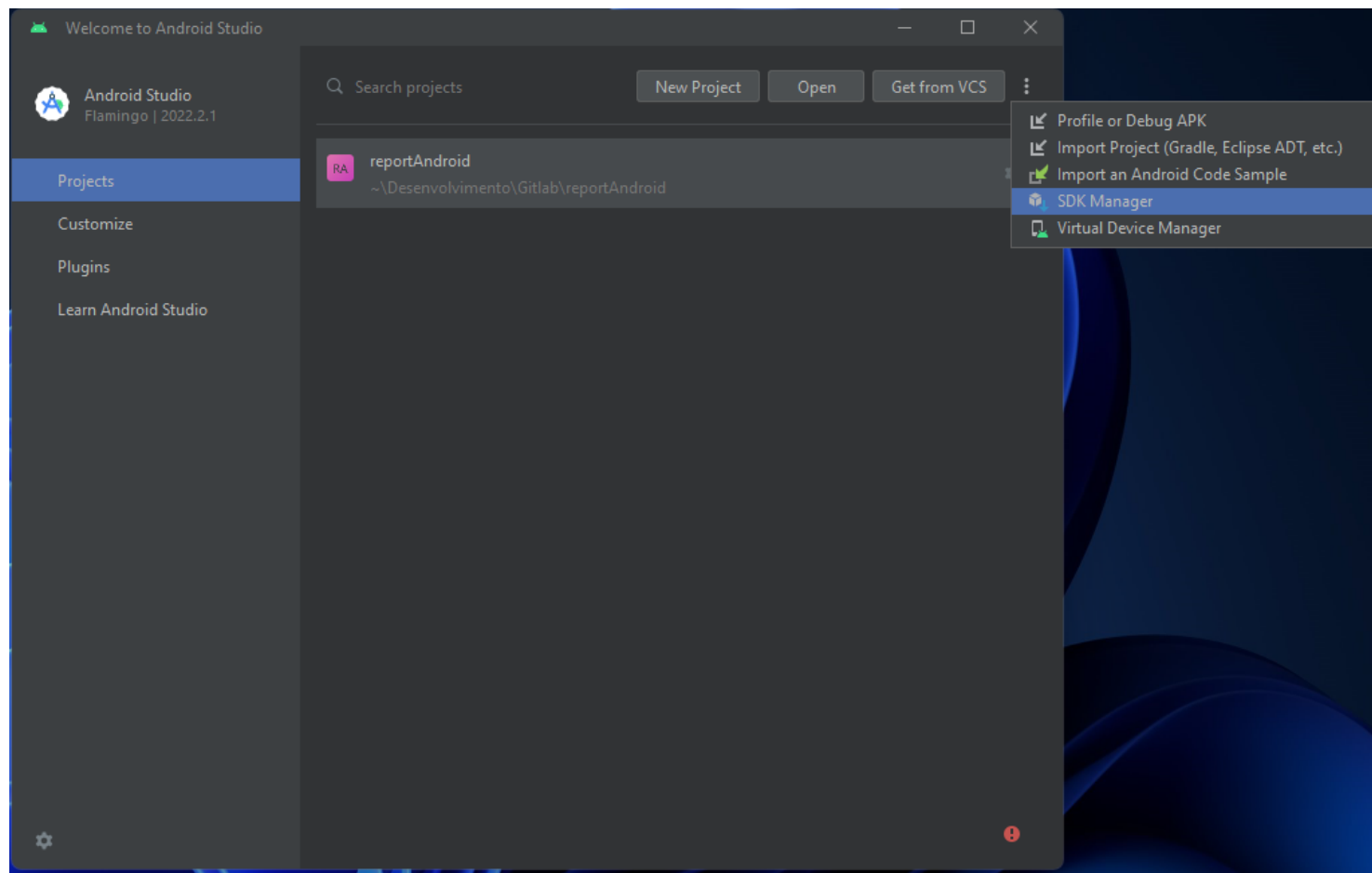
SDK





Configurações Básicas

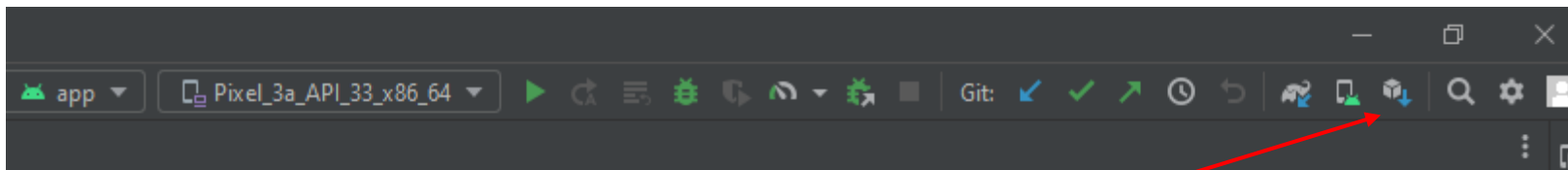
SDK





Configurações Básicas

SDK





Configurações Básicas

Após abrir o menu de configurações de SDK do Android Studio será possível configurar quais versões do Android você irá usar em sua máquina e seus projetos.



versões do android



Cupcake
1.5



Donut
1.6



Eclair
2.0/2.1



Froyo
2.2



Gingerbread
2.3



Honeycomb
3.0/3.1



Ice Cream Sandwich
4.0



Jelly Bean
4.1/4.2/4.3



KitKat
4.4



Lollipop
5.0



Marshmallow
6.0



Nougat
7.0



Oreo
8.0



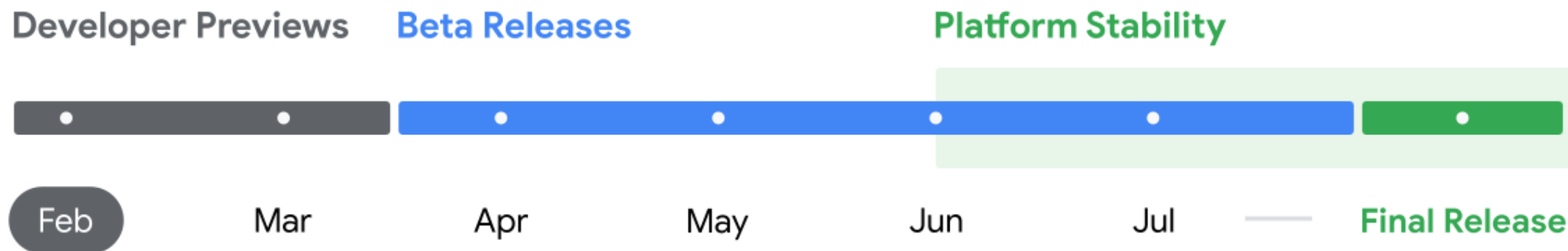
Pie
9.0





Como se preparar para o Android 15?

A prévia para desenvolvedores 1 já está disponível para testes com seus apps.



<https://android-developers.googleblog.com/2024/02/first-developer-preview-android15.html>



Settings

Q

Appearance & Behavior

Appearance

Menus and Toolbars

System Settings

HTTP Proxy

Data Sharing

Date Formats

Updates

Process Elevation

Passwords

Android SDK

Memory Settings

Notifications

Quick Lists

Path Variables

Keymap

Editor

Build, Execution, Deployment

Kotlin

Tools

Advanced Settings

Layout Inspector

Appearance & Behavior > System Settings > Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: C:\Users\User\AppData\Local\Android\Sdk

Edit Optimize disk space

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
<input type="checkbox"/>	Android VanillalceCream Preview	VanillalceCream	1	Not installed
<input type="checkbox"/>	Android UpsideDownCakePrivacySandbox Preview	UpsideDownCakePrivacySandbox	3	Not installed
<input type="checkbox"/>	Android API 34	34	3	Not installed
<input type="checkbox"/>	Android TiramisuPrivacySandbox Preview	TiramisuPrivacySandbox	9	Not installed
<input checked="" type="checkbox"/>	Android 13.0 (Tiramisu)	33	2	Update available
<input type="checkbox"/>	Android 12L (Sv2)	32	1	Partially installed
<input type="checkbox"/>	Android 12.0 (S)	31	1	Not installed
<input type="checkbox"/>	Android 11.0 (R)	30	3	Not installed
<input type="checkbox"/>	Android 10.0 (Q)	29	5	Not installed
<input type="checkbox"/>	Android 9.0 (Pie)	28	6	Not installed
<input type="checkbox"/>	Android 8.1 (Oreo)	27	3	Not installed
<input type="checkbox"/>	Android 8.0 (Oreo)	26	2	Not installed
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Not installed
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2	Not installed
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Not installed
<input type="checkbox"/>	Android 5.1 (Lollipop)	22	2	Not installed
<input type="checkbox"/>	Android 5.0 (Lollipop)	21	2	Not installed
<input type="checkbox"/>	Android 4.4W (KitKat Wear)	20	2	Not installed

☒ Hide Obsolete Packages ☐ Show Package Details

? Project-level settings will be applied to new projects

OK Cancel Apply



Settings



Appearance & Behavior

Appearance

Menus and Toolbars

System Settings

HTTP Proxy

Data Sharing

Date Formats

Updates

Process Elevation

Passwords

Android SDK

Memory Settings

Notifications

Quick Lists

Path Variables

Keymap

> Editor

> Build, Execution, Deployment

Kotlin

> Tools

Advanced Settings

Layout Inspector

Appearance & Behavior > System Settings > Android SDK



Manager for the Android SDK and Tools used by the IDE

Android SDK Location: C:\Users\User\AppData\Local\Android\Sdk

[Edit](#) [Optimize disk space](#)

SDK Platforms

SDK Tools

SDK Update Sites

Below are the available SDK developer tools. Once installed, the IDE will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

	Name	Version	Status
▼	Android SDK Build-Tools 35-rc1		
<input type="checkbox"/>	35.0.0-rc1	35.0.0 rc1	Not installed
<input type="checkbox"/>	34.0.0	34.0.0	Not installed
<input type="checkbox"/>	34.0.0-rc3	34.0.0 rc3	Not installed
<input type="checkbox"/>	34.0.0-rc2	34.0.0 rc2	Not installed
<input type="checkbox"/>	34.0.0-rc1	34.0.0 rc1	Not installed
<input type="checkbox"/>	33.0.3	33.0.3	Not installed
<input checked="" type="checkbox"/>	33.0.2	33.0.2	Installed
<input type="checkbox"/>	33.0.1	33.0.1	Not installed
<input type="checkbox"/>	33.0.0	33.0.0	Not installed
<input type="checkbox"/>	32.1.0-rc1	32.1.0 rc1	Not installed
<input type="checkbox"/>	32.0.0	32.0.0	Not installed
<input checked="" type="checkbox"/>	31.0.0	31.0.0	Installed
<input type="checkbox"/>	30.0.3	30.0.3	Not installed
<input type="checkbox"/>	30.0.2	30.0.2	Not installed
<input type="checkbox"/>	30.0.1	30.0.1	Not installed
<input type="checkbox"/>	30.0.0	30.0.0	Not installed
<input type="checkbox"/>	29.0.3	29.0.3	Not installed
<input type="checkbox"/>	29.0.2	29.0.2	Not installed

Looking for updates...

☒ Hide Obsolete Packages ☒ Show Package Details

Project-level settings will be applied to new projects

OK

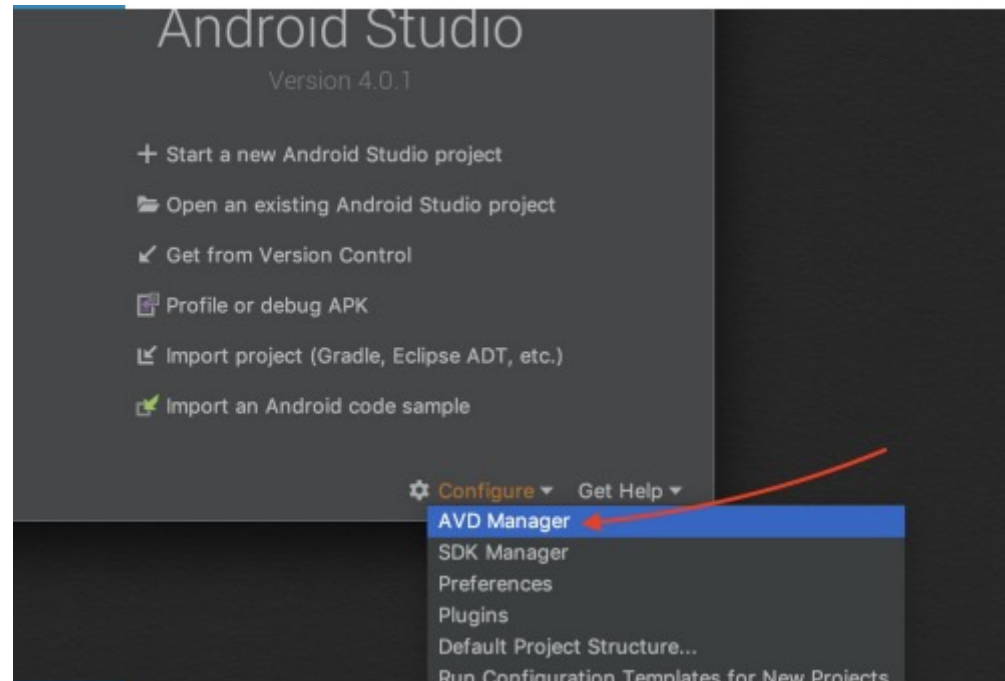
Cancel

Apply



Configurações Básicas

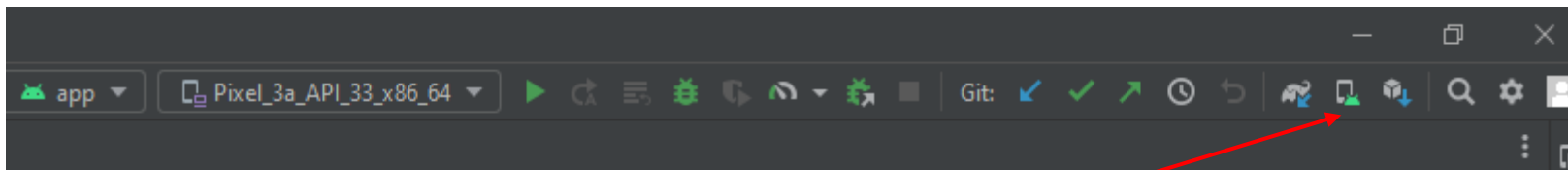
AVD MANAGER





Configurações Básicas

AVD MANAGER





Configurações Básicas

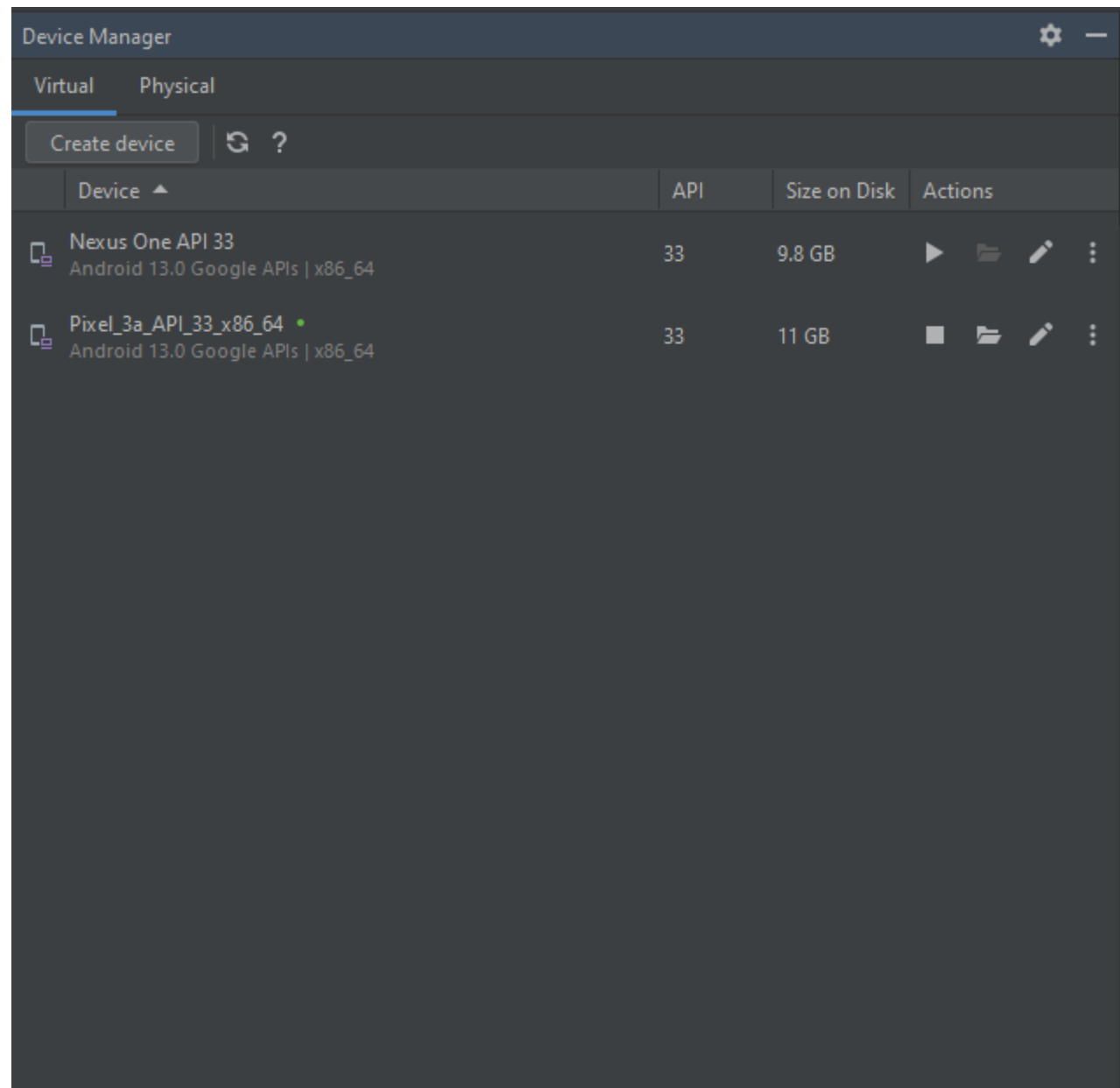
AVD MANAGER

Acessando o menu de configurações do AVD (AVD Manager) será possível criar novas imagens para emulação e também configurar detalhes dos dispositivos a serem emulados, tamanho de tela, resolução, simulação de câmera e etc.



Configurações Básicas

AVD MANAGER

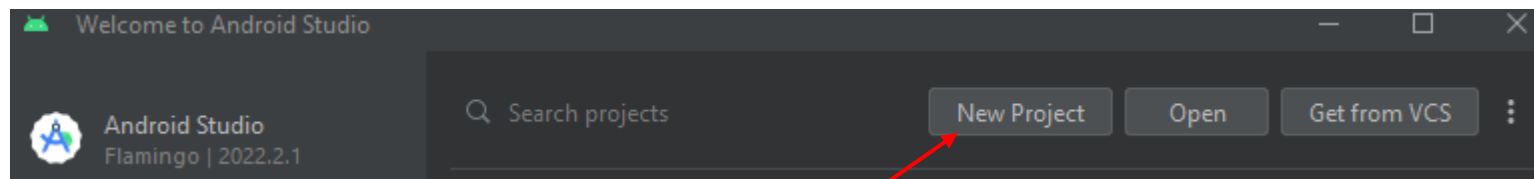




Criando um novo projeto

Tela inicial ou

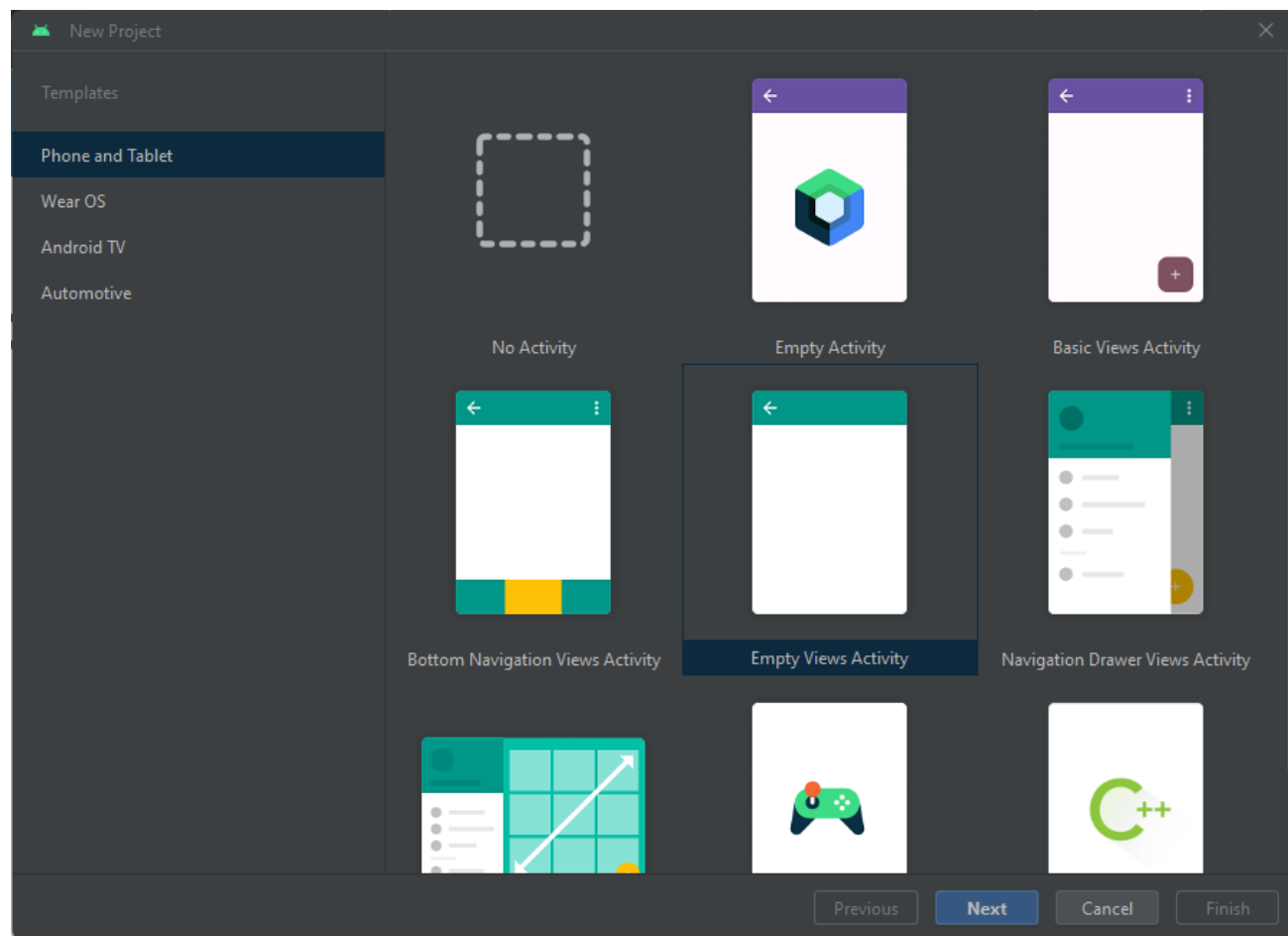
File -> New -> New Project





Criando um novo projeto

Selecionando um modelo de App





Criando um novo projeto

Configurando o novo projeto

Nome do App

Nome do pacote

New Project

Empty Views Activity

Creates a new empty activity

Name: My Application

Package name: com.example.myapplication

Save location: C:\Users\User\AndroidStudioProjects\MyApplication

Language: Kotlin

Minimum SDK: API 24: Android 7.0 (Nougat)

i Your app will run on approximately 96,3% of devices.
[Help me choose](#)

Previous Next Cancel Finish



Criando um novo projeto

Configurando o novo projeto

Pasta do projeto

Linguagem padrão do projeto

New Project

Empty Views Activity

Creates a new empty activity

Name: My Application

Package name: com.example.myapplication

Save location: C:\Users\User\AndroidStudioProjects\MyApplication

Language: Kotlin

Minimum SDK: API 24: Android 7.0 (Nougat)

! Your app will run on approximately 96,3% of devices.
[Help me choose](#)

Previous Next Cancel Finish



Criando um novo projeto

Configurando o novo projeto

Versão mínima do Android suportada em seu projeto

New Project

Empty Views Activity

Creates a new empty activity

Name: My Application

Package name: com.example.myapplication

Save location: C:\Users\User\AndroidStudioProjects\MyApplication

Language: Kotlin

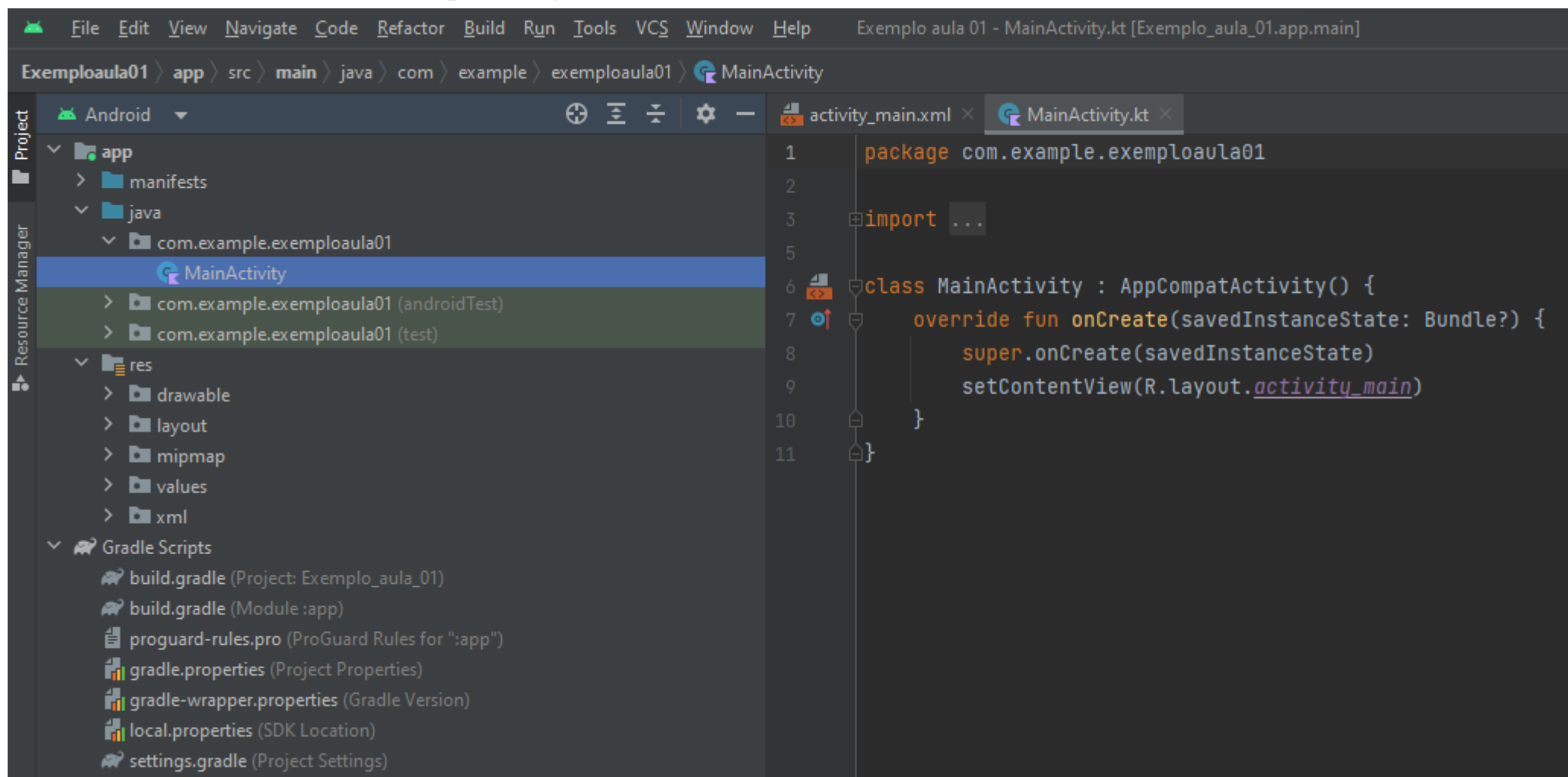
Minimum SDK: API 24: Android 7.0 (Nougat)

Information: Your app will run on approximately 96,3% of devices.
[Help me choose](#)

Previous Next Cancel Finish

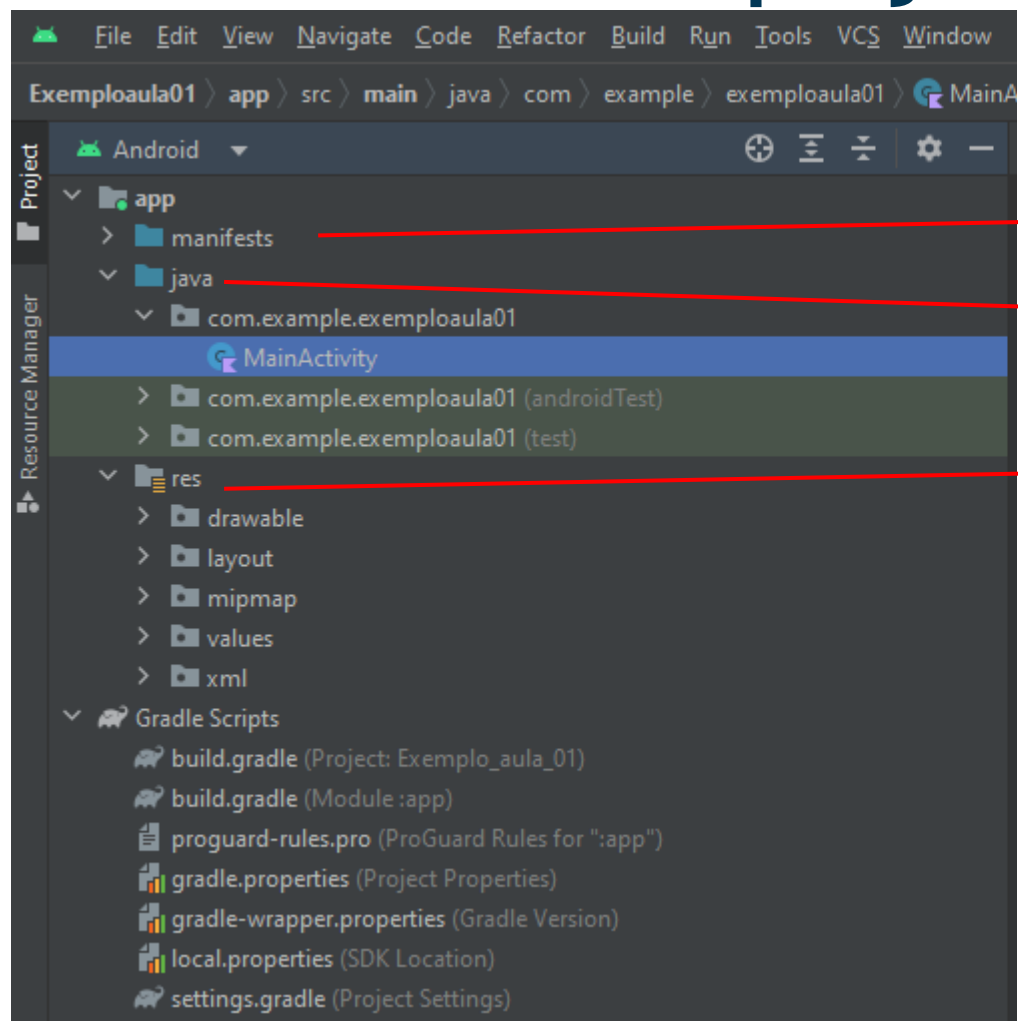


Estrutura de um projeto





Estrutura de um projeto



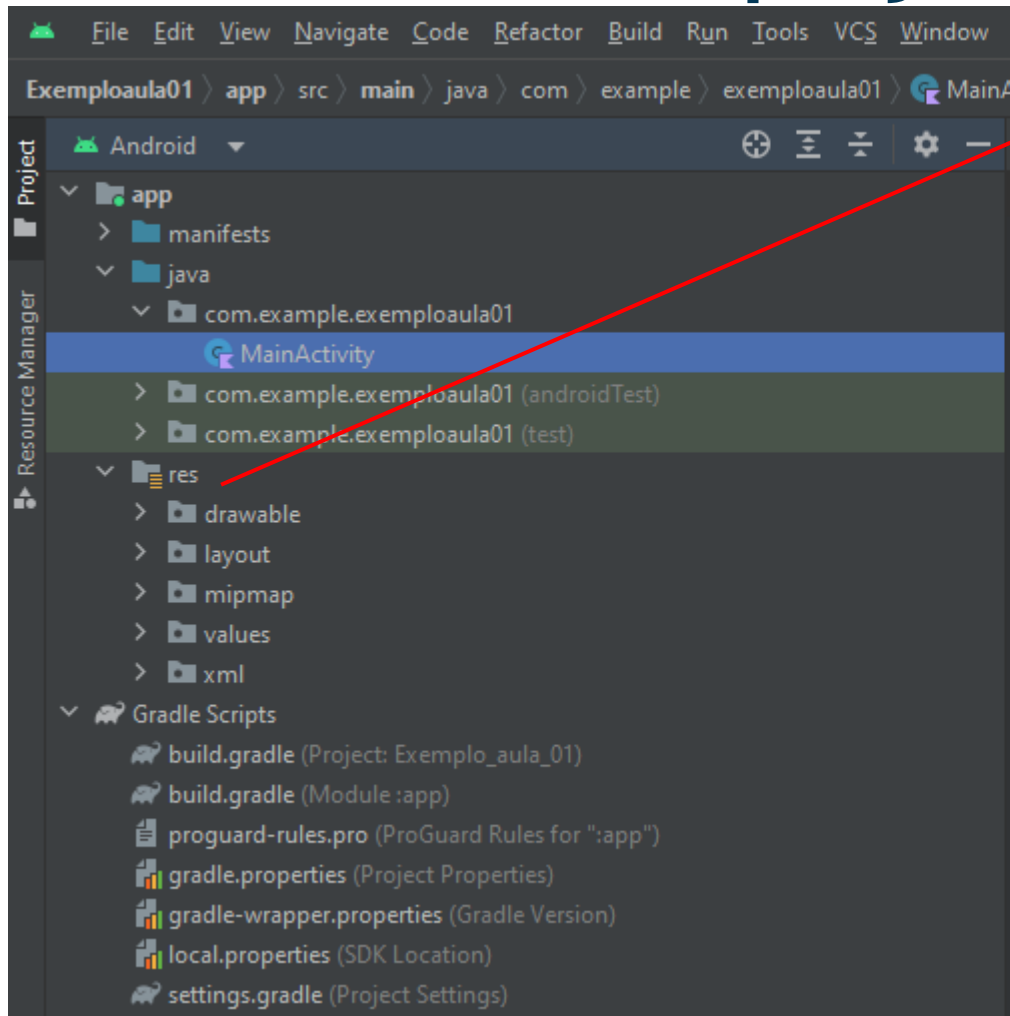
Configurações do App

Código fonte em Java ou Kotlin

Resources (recursos)



Estrutura de um projeto



Resources (recursos)

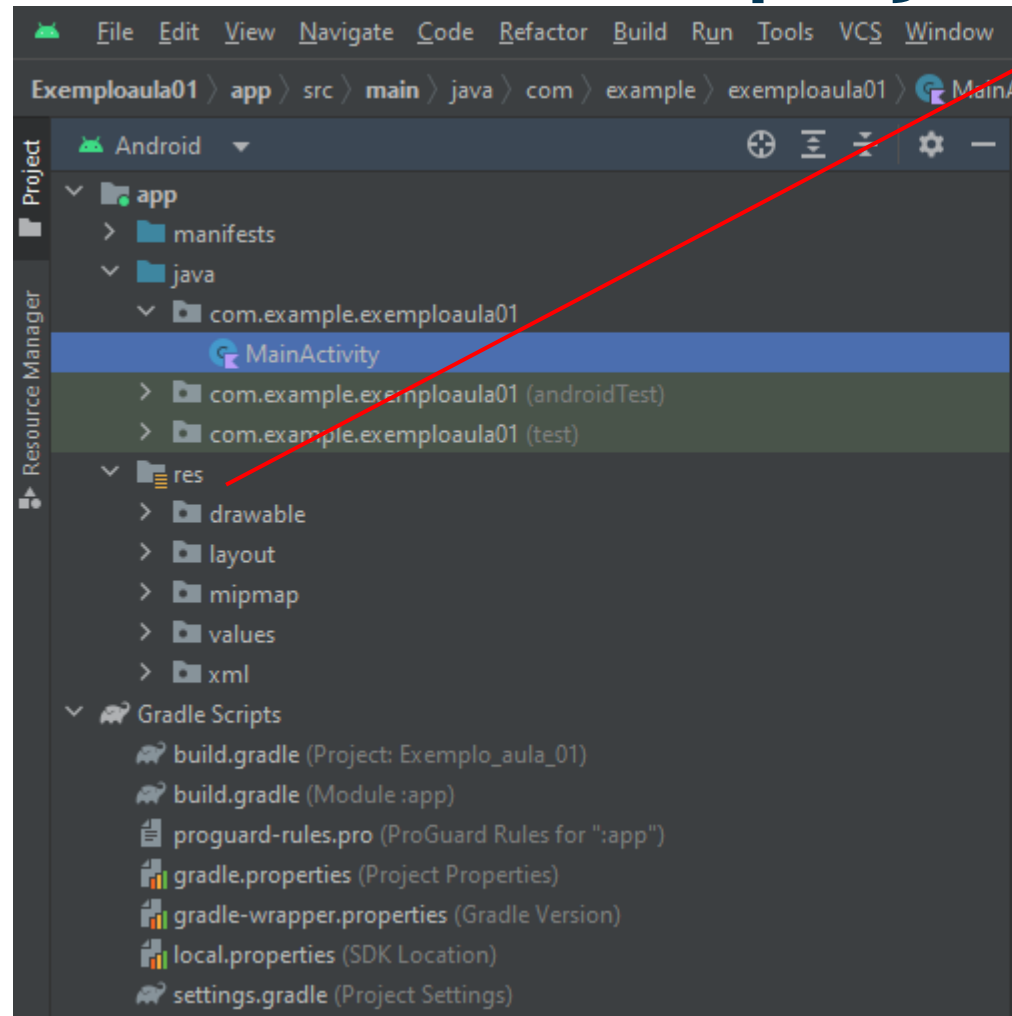
Pasta bem importante para seu projeto Android.

Tudo que não for código-fonte OU configurações deve ficar nessa pasta.

Imagens, ícones, textos/strings, catálogo de cores, layouts e menus visuais em formato XML e etc.



Estrutura de um projeto



Resources (recursos)

drawable: imagens, botões, pedaços de layout e etc

layout: conjunto maior de telas ou boa parte delas

menu: itens de menu

mipmap: uma espécie de drawable porém focado mais em ícones

navigation: guia de navegação entre telas (costura)

values: catálogos de strings, cores, tamanhos, estilos e etc



ferramentas

Uma das principais ferramentas aliadas de seu projeto e que trabalham junto do Android Studio é o Graddle.

O **Graddle** é uma ferramenta de build (Build Tool) com foco na automatização e configuração simplificada e que trabalha com diversas linguagens de programação.

Junto ao Android Studio o Graddle tem foco em compilar seu código Kotlin ou Java e organizar de maneira binária para que o sistema operacional Android possa rodar seu app.

Outras informações sobre o Graddle em:

<https://developer.android.com/studio/build/gradle-tips?hl=pt-br>

<https://gradle.org/>



ferramentas

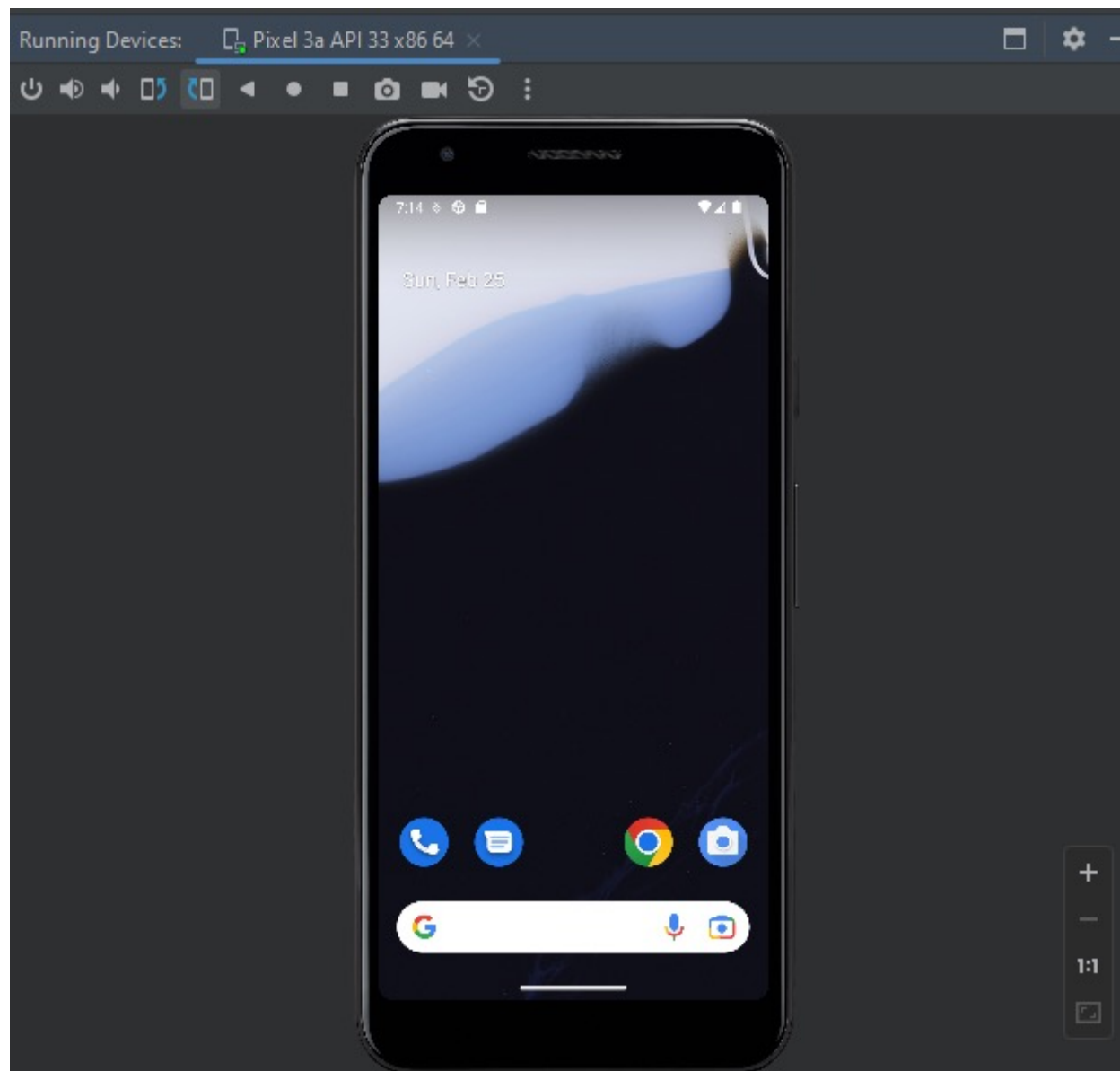
Além do Graddle e de outras ferramentas de debug (que iremos falar em outra aula) outro recurso muito importante disponível de forma fácil e ágil no Android Studio é o acesso a emulação e testes de nossos apps.

Depois de criarmos e configurações nosso primeiro dispositivo de testes no AVD ou disponibilizar o acesso físico (via USB ou WI-FI) a um smartphone real rodando Android* podemos focar no início dos testes de um app.



ferramentas

Emulador Android





ferramentas

Vale observar que o Emulador Android baseado em AVD é o "veículo" de mais fácil acesso no desenvolvimento nativo para Android. Porém ele não é o único tipo de emulador disponível no mercado.

Antes do Google ampliar o suporte a aceleração de hardware através dos processadores Intel, o emulador oficial do Android era considerado muito lento, nesta época surgiram outras alternativas mais rápidas como

o **Genymotion**:

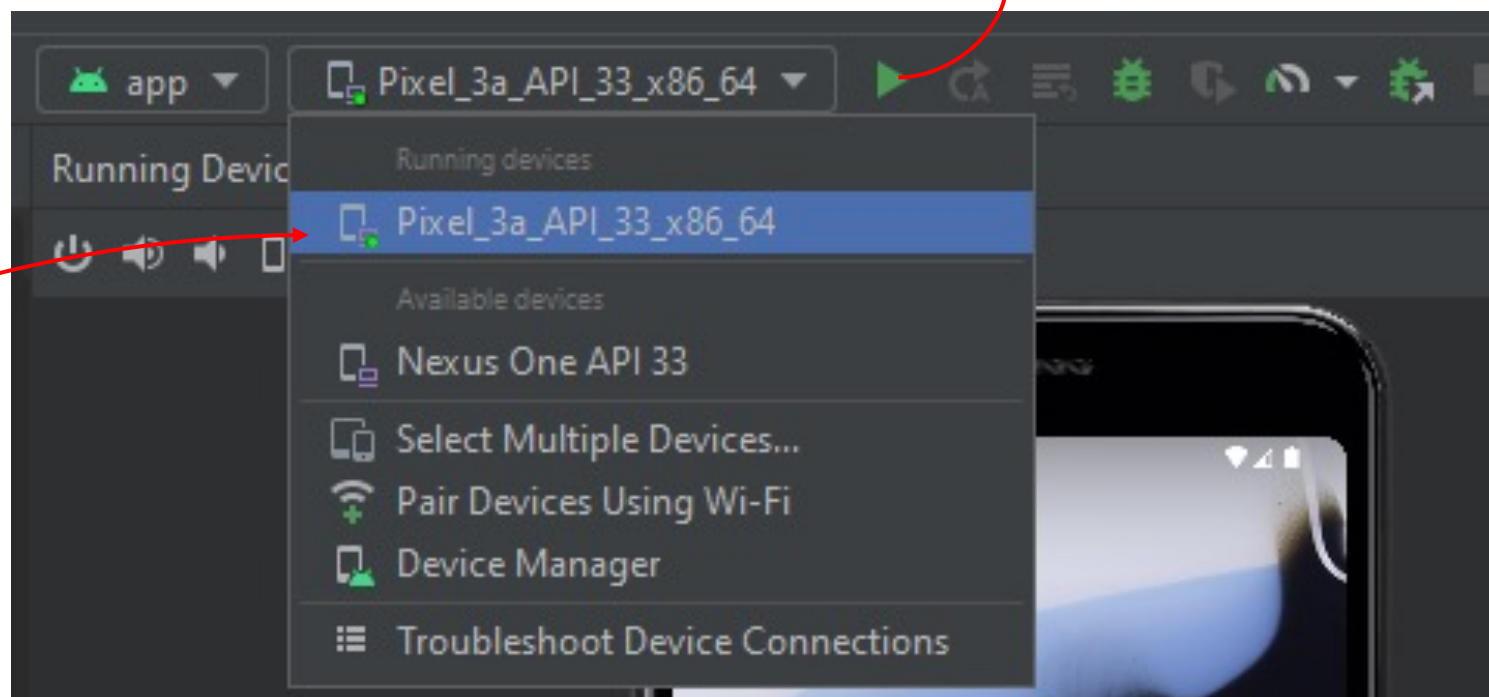
<https://www.genymotion.com/>



rodando um projeto

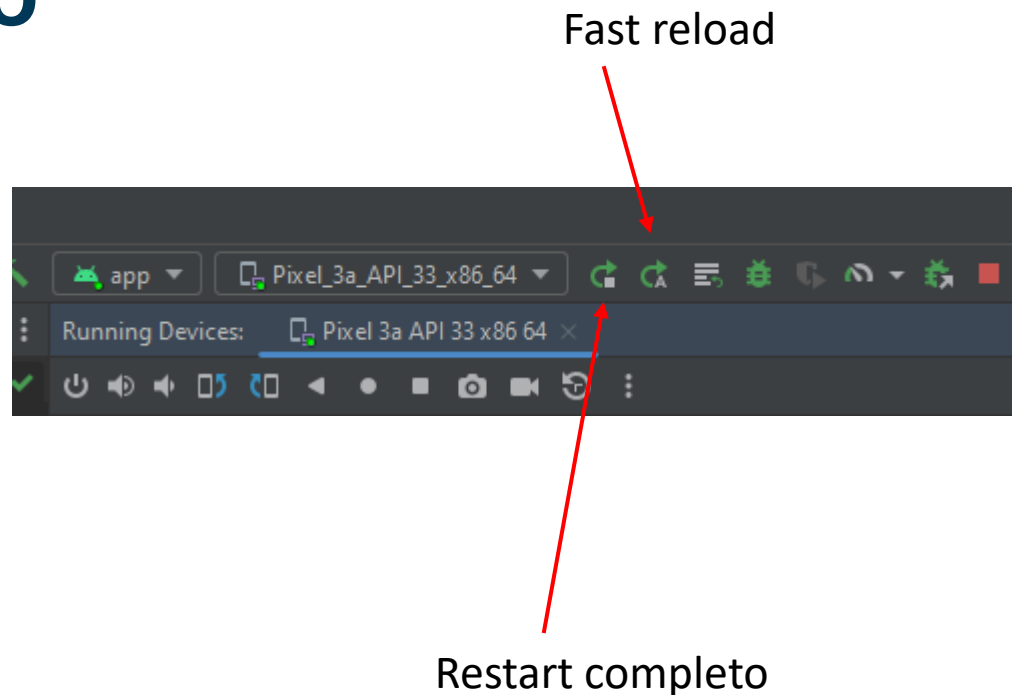
Dispositivos
disponíveis

Rodar projeto





rodando um projeto

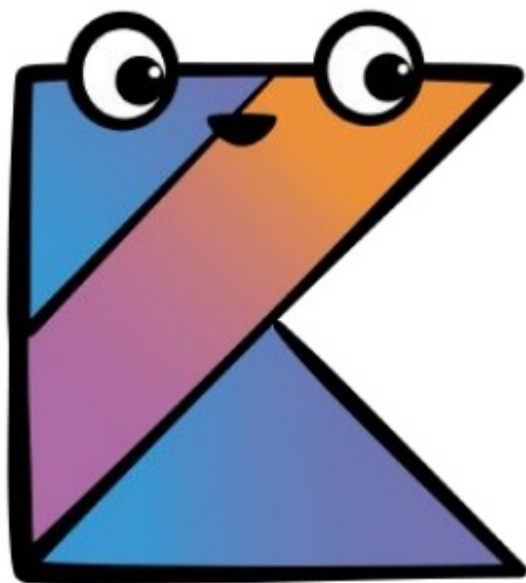


Uma vez que seu projeto esteja rodando em seu emulador ou dispositivo, é possível continuar desenvolvendo.

Usar o fast reload permite testar mais rapidamente pequenas alterações.



kotlin...





Kotlin *versus* Java

O Java é notoriamente conhecido como uma das linguagens líder de mercado no mundo corporativo e também foi durante muitos anos a única forma de desenvolver aplicações nativas para Android.

Em 2010 isso começou a mudar, fora criada a linguagem Kotlin (que também roda na JVM - Java Virtual Machine) para resolver alguns problemas que os desenvolvedores enfrentam usando apenas a linguagem Java (principalmente para Mobile).

Desde o anúncio oficial do Google sobre o suporte da linguagem Kotlin para o desenvolvimento Android, no Google I/O 2017, cada vez mais cresce o número de aplicações Android escritas parcial ou totalmente em Kotlin, o que significa que cada vez mais desenvolvedores no mercado estão fazendo o uso da mesma.

A partir do Android Studio 3.0, o suporte à linguagem Kotlin foi adicionado nativamente na IDE, facilitando ainda mais o desenvolvimento de aplicações com o uso da mesma.



Criação do Kotlin

JetBrains, empresa de desenvolvimento de software famosa por desenvolver ferramentas para desenvolvedores, como a IDE IntelliJ, estava procurando uma linguagem que tivesse algumas funcionalidades que Java não possuía.

Na época, uma solução pensada foi a linguagem Scala, que roda na JVM, mas ela logo foi descartada, pois sua compilação é muito lenta. Em 2010, a JetBrains criou o Kotlin e em 2016 - 2017 a linguagem se tornou pública e foi sendo adotada por vários desenvolvedores.

A empresa (JetBrains), além de criar a linguagem Kotlin também criou as melhores IDEs existentes atualmente no mercado, como: PhpStorm, CLion, PyCharm e IntelliJ, que é também a base do Android Studio, IDE oficial da Google para desenvolvimento de aplicações Android.



Vantagens do Kotlin

Interoperabilidade

Desde o início, a intenção do projeto foi usar o conhecimento e a experiência existentes para disponibilizar todas as bibliotecas aos programadores da Kotlin.

Os desenvolvedores podem simplesmente escrever módulos no Kotlin que funcionam perfeitamente dentro do código Java existente.



Vantagens do Kotlin

Sintaxe Limpa e Expressiva

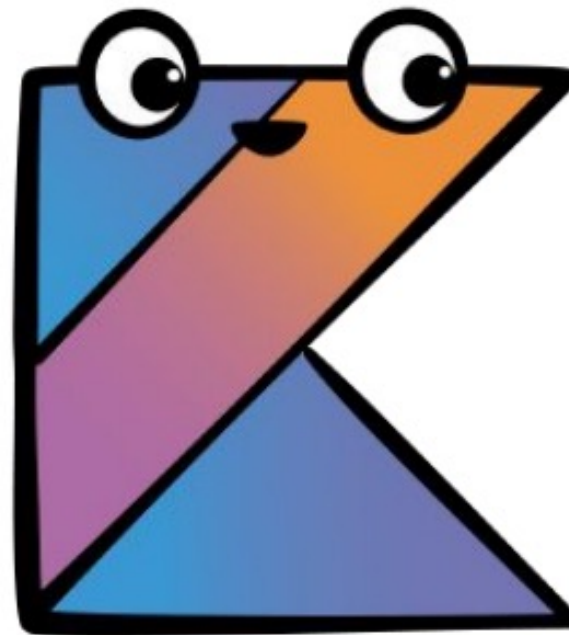
Kotlin introduz várias melhorias que causam um impacto significativo na legibilidade do código em relação ao Java, aumentando a produtividade e trazendo mais facilidade para manutenção do código gerado.

Alguns pontos:

- Ponto e vírgula são opcionais
- Chaves "{}" são opcionais nos casos em que não são úteis
- Getter/Setters são opcionais
- Tudo é objeto - primitivos são usados nos bastidores automaticamente, se necessário



Comparativo JAVA



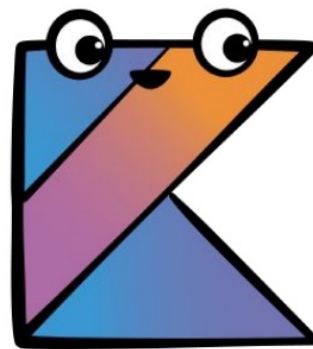


```
public class Usuario {  
  
    private String id;  
    private String firstName;  
    private String lastName;  
    private int age;  
  
    public Usuario(final String id, final String firstName, final String lastName, final int age) {  
        this.id = id;  
        this.firstName = firstName;  
        this.lastName = lastName;  
        this.age = age;  
    }  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(final String id) {  
        this.id = id;  
    }  
  
    public String getFirstName() {  
        return firstName;  
    }  
  
    public void setFirstName(final String firstName) {  
        this.firstName = firstName;  
    }  
  
    public String getLastName() {  
        return lastName;  
    }  
  
    public void setLastName(final String lastName) {  
        this.lastName = lastName;  
    }  
}
```





```
data class KotlinUser(var id: String,  
    var firstName: String,  
    var lastName: String,  
    var age: Int)
```



Por padrão, data classes já possuem uma implementação útil de `equals()`, `hashCode()` e `toString()`, que usa como parâmetros as propriedades definidas no construtor da classe e, por isso, não é necessário re-implementar estes métodos.

Fonte (1): <https://www.zup.com.br/blog/java-vs-kotlin-vantagens-desvantagens>

Fonte (2): <https://mobile.blog/motivos-para-trocar-o-java-pelo-kotlin-ainda-hoje>



Homework

Para quem tiver tempo, a tarefinha

- Baixar e Instalar o Android Studio
- Configurar as SDKs do Android
- Criar e "dar play" em sua primeira AVD de emulação
- Criar um projeto em branco baseado em um modelo
- Rodar este projeto
- Olhar com mais calma os links disponibilizados como extra nestes slides



Obrigado 🚀

