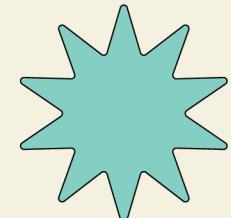
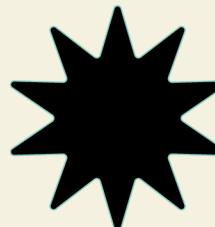
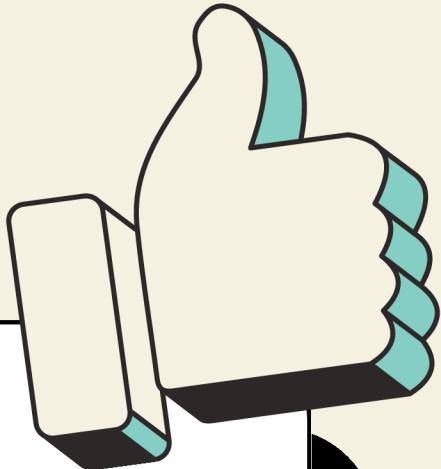
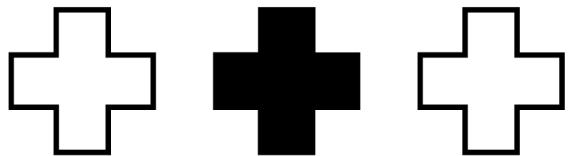




KOTLIN

# AGENDA



01

02

03

04

**HISTÓRIA DA LINGUAGEM;**

**CARACTERÍSTICA DA LINGUAGEM;**

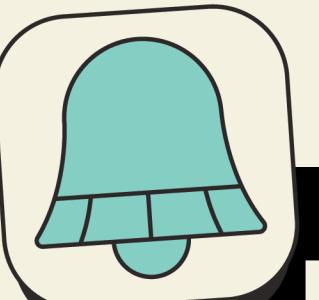
**TUTORIAL DE INSTALAÇÃO;**

**EXPLICAÇÃO DO CÓDIGO.**

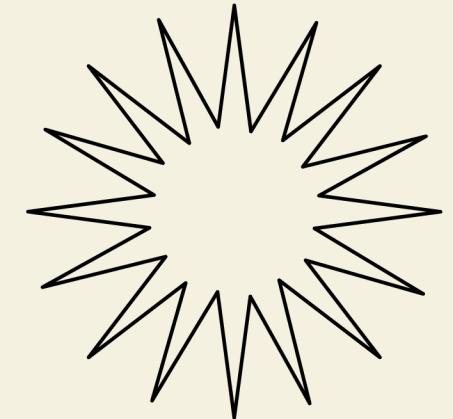




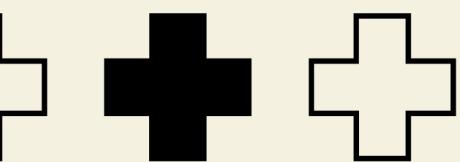
KOTLIN

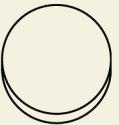


# INTRODUÇÃO

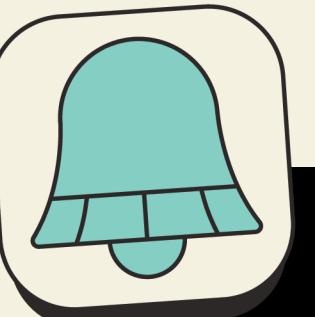


- **Kotlin foi desenvolvida pela JetBrains, uma empresa de software conhecida por suas ferramentas de desenvolvimento como o IntelliJ IDEA**
- **A linguagem começou a ser desenvolvida em 2010, com o primeiro lançamento oficial em 2011.**
- **A ideia por trás do Kotlin era criar uma linguagem moderna que pudesse funcionar de forma nativa na JVM (Java Virtual Machine) e que fosse mais concisa, segura e expressiva do que o Java, sem deixar de ser 100% interoperável com ele.**

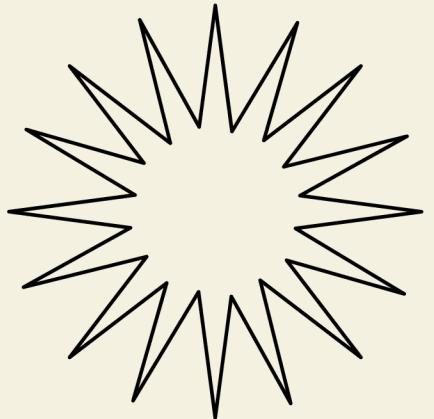




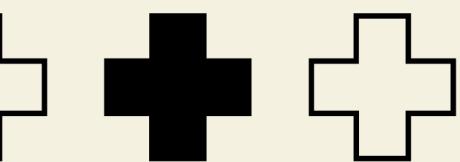
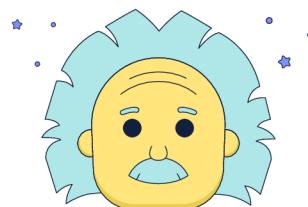
KOTLIN

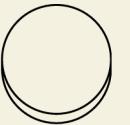


## CONTEXTO HISTÓRICO

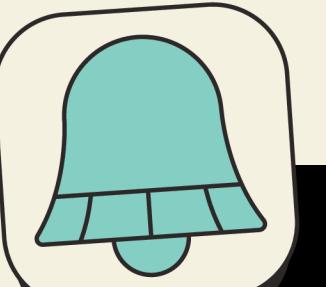


- **No início dos anos 2010, o desenvolvimento de software estava em uma fase de transição.**
- **O Java, embora amplamente utilizado, enfrentava críticas por sua verbosidade e por problemas de segurança.**
- **Desenvolvedores estavam buscando alternativas que pudessem manter a robustez e o ecossistema do Java, mas com uma sintaxe mais limpa e moderna.**

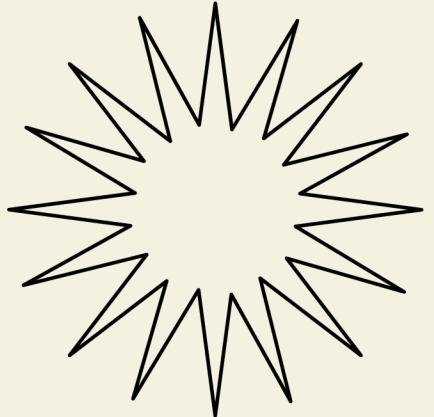
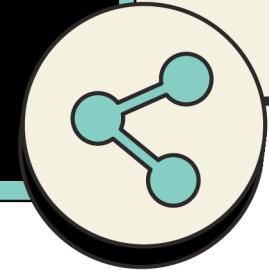




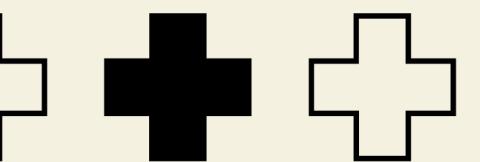
KOTLIN

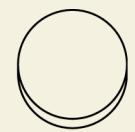


## POPULARIZAÇÃO

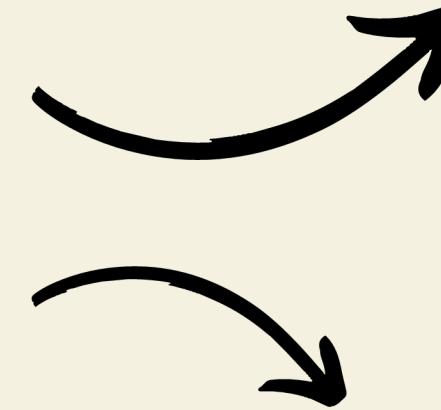
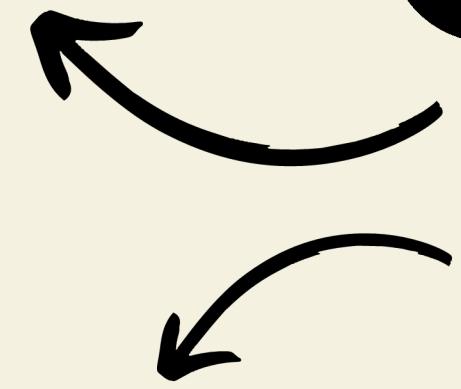
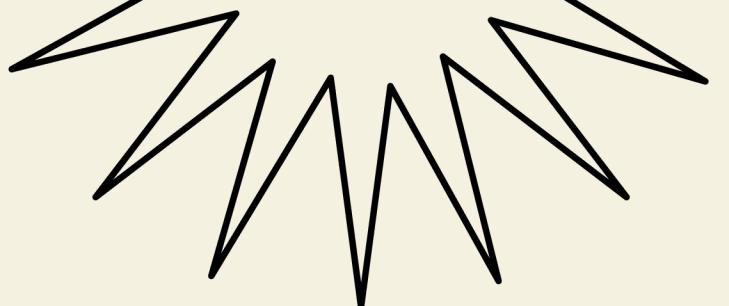


- **Começou a ganhar popularidade em 2016, após ser adotada como uma linguagem oficial para o desenvolvimento Android pelo Google.**
- **A escolha do Google foi um grande marco, pois muitos desenvolvedores estavam insatisfeitos com o Java para Android devido à sua complexidade e à batalha legal entre o Google e a Oracle em torno do uso do Java.**





KOTLIN





KOTLIN

# PARADIGMA



## QUAL O PARADIGMA?

A linguagem de programação Kotlin é multiparadigma, mas se destaca em dois paradigmas:

**Orientação a Objetos;**

**Paradigma funcional.**



## ORIENTAÇÃO A OBJ.

Como Java, Kotlin permite a criação de classes e objetos, encapsulamento, herança, e polimorfismo.

## FUNCIONAL

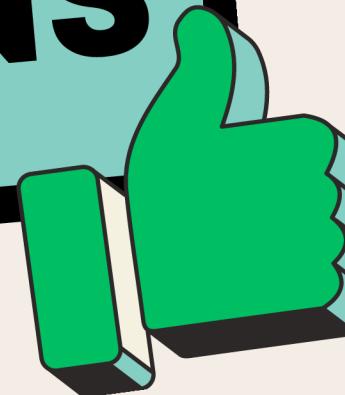
Kotlin facilita a programação funcional com recursos como funções de ordem superior, lambdas, imutabilidade, e a capacidade de tratar funções como cidadãos de primeira classe.



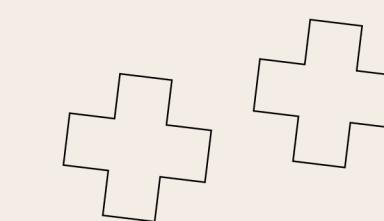
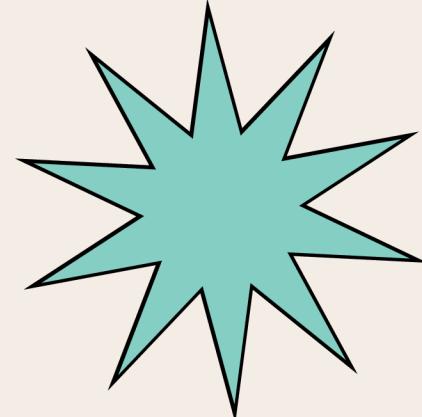
KOTLIN



# VANTAGENS

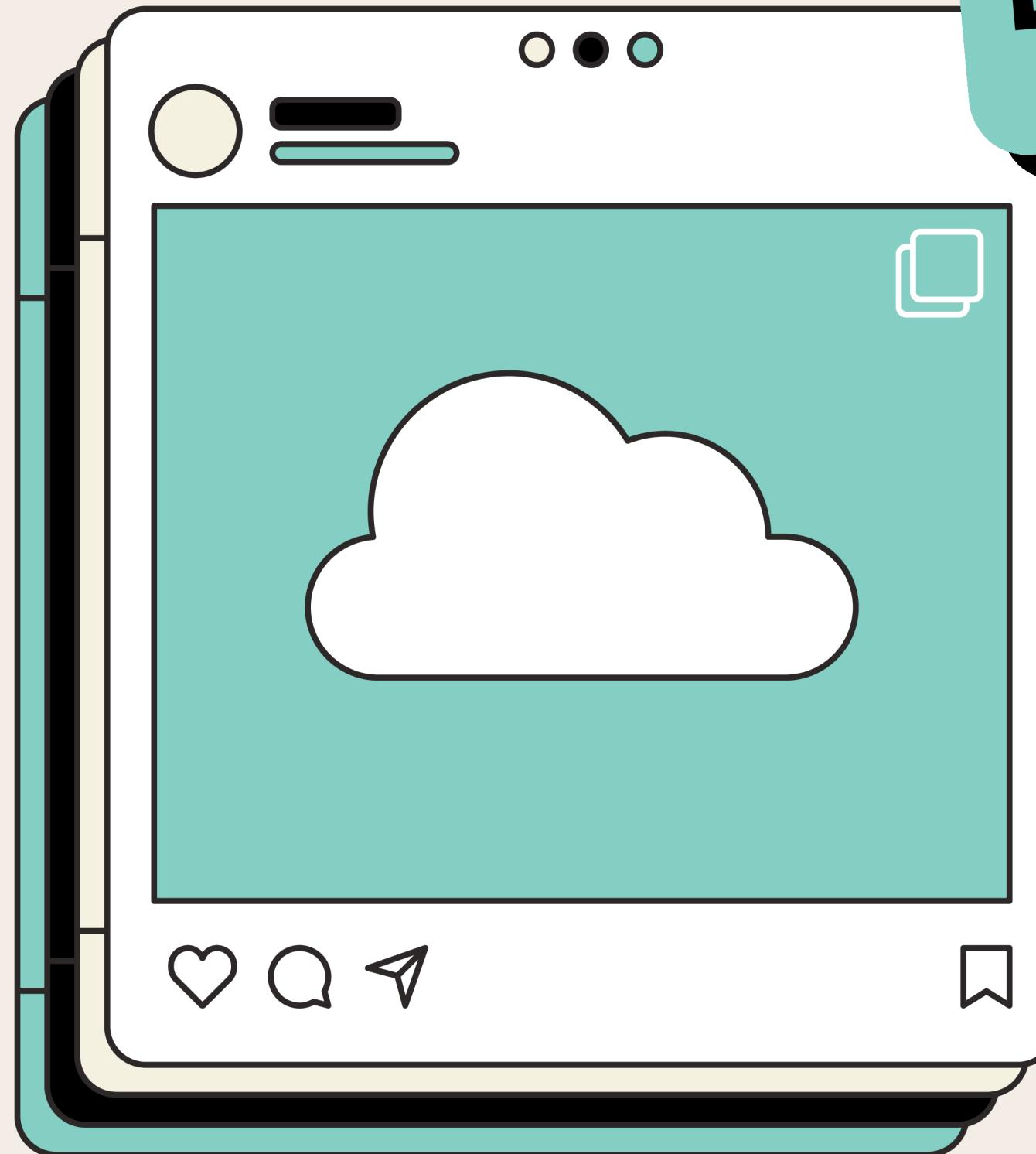


- 01** **Reduz a quantidade de código necessário, tornando-o mais legível e compacto.**
- 02** **100% compatível com Java, permitindo uso de bibliotecas já existentes.**
- 03** **Permite desenvolvimento em diferentes ambientes (JVM, Native, JS) com código compartilhado.**
- 04** **Facilita a criação de DSLs, tornando APIs mais intuitivas e expressivas.**
- 05** **Introduz nulidade segura.**





KOTLIN



## DESVANTAGENS



01

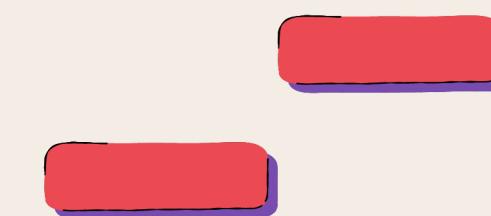
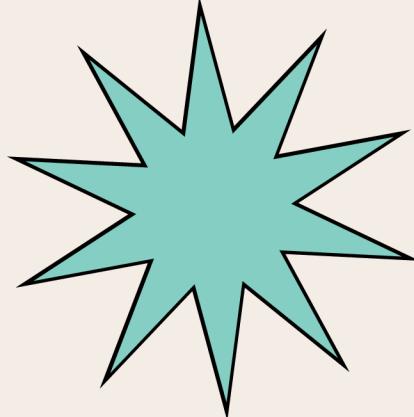
**Pode ser menos performática que o Java em alguns casos.**

02

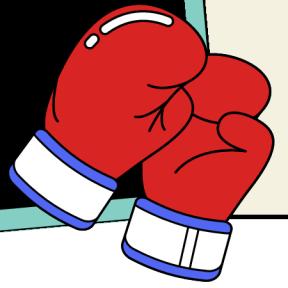
**Conceitos funcionais podem ser desafiadores para novos desenvolvedores, o que acarreta uma dificuldade na curva de aprendizado.**

03

**Comunidade ainda em desenvolvimento.**



# KOTLIN X JAVA



## KOTLIN



## JAVA

- Mais conciso, seguro contra nulidade;
- Mais expressivo, especialmente em termos de programação funcional;
- Facilita a criação de DSLs e tem suporte multiplataforma.

- Comunidade maior;
- Mais bibliotecas e frameworks;
- Mais acessível para novos programadores.

# KOTLIN X SCALA



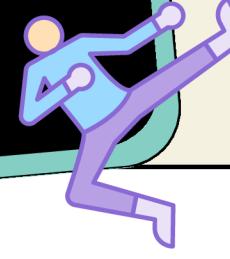
## KOTLIN

- Mais simples e tem melhor interoperabilidade com Java;
- Curva de aprendizado mais fácil em relação a linguagem de programação Scala.

## SCALA

- Mais poderosa e flexível para programação funcional;
- Oferece características mais avançadas, tais como implicações e tipo de dependência.

# KOTLIN X SWIFT



## KOTLIN

- Pode ser utilizado para o desenvolvimento Android;
- Oferece suporte multiplataforma.

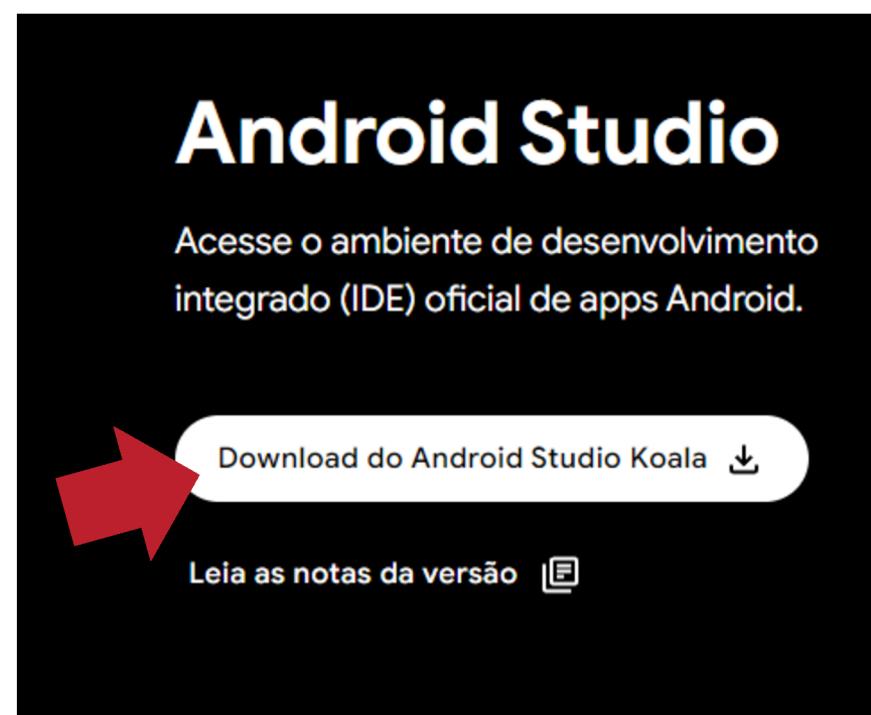
## SWIFT

- Projetado especificamente para o ecossistema da Apple;
- Integração profunda com APIs e frameworks iOS/macOS.

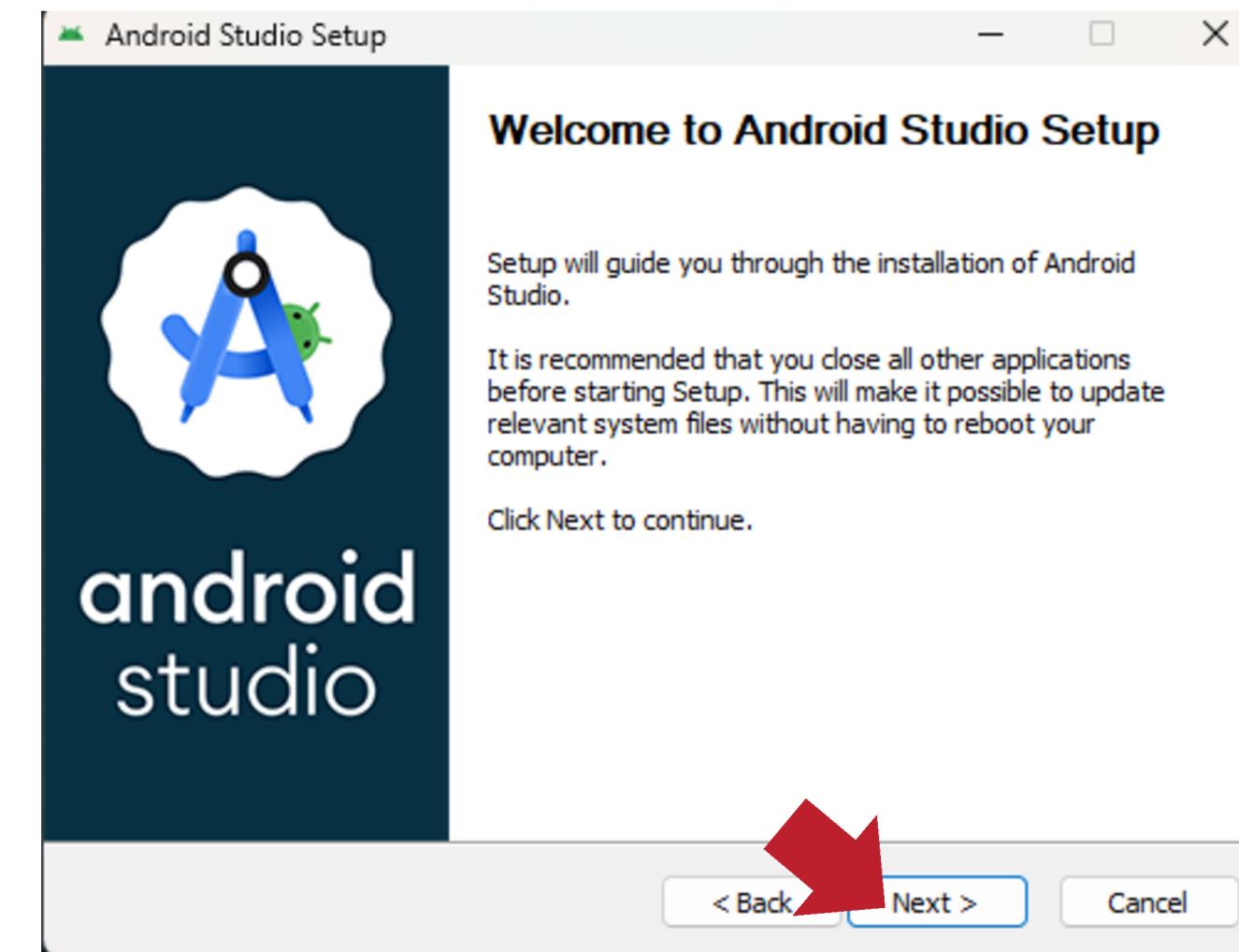
# TUTORIAL DE INSTALAÇÃO



## 1 - Download do Android Studio;



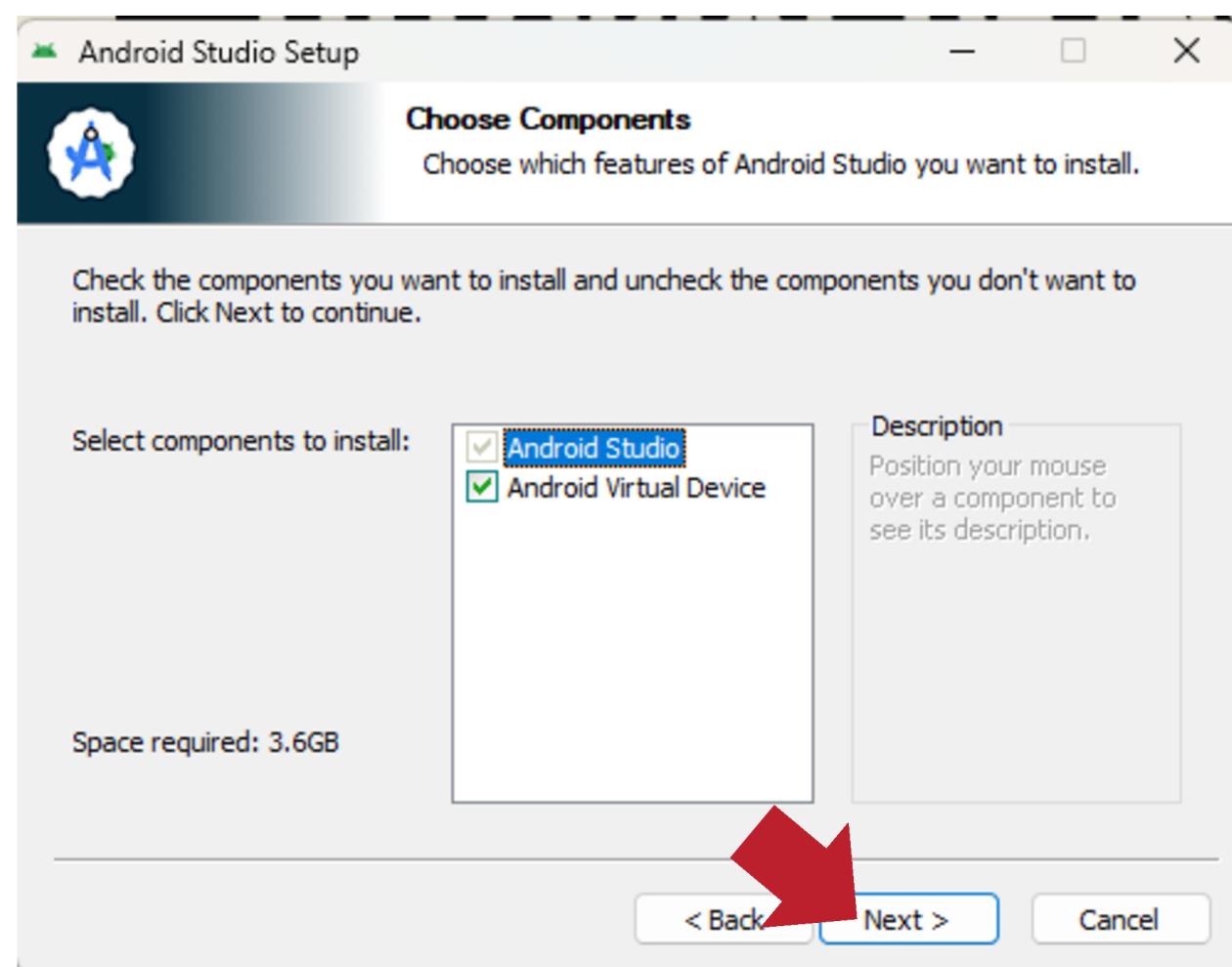
## 2 - Instalar o Android Studio



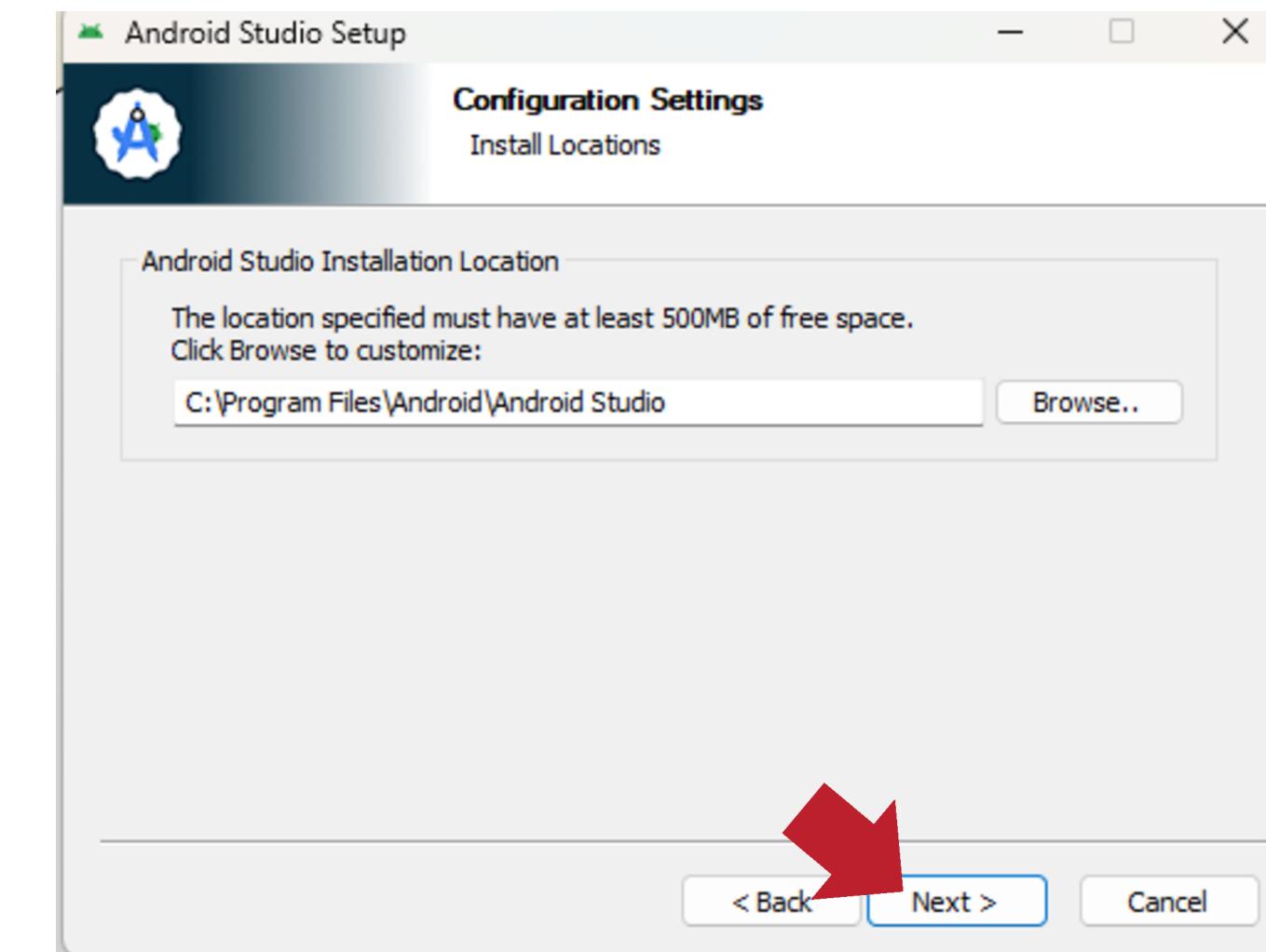
# TUTORIAL DE INSTALAÇÃO



## 2.1 - Instalando o Android Studio



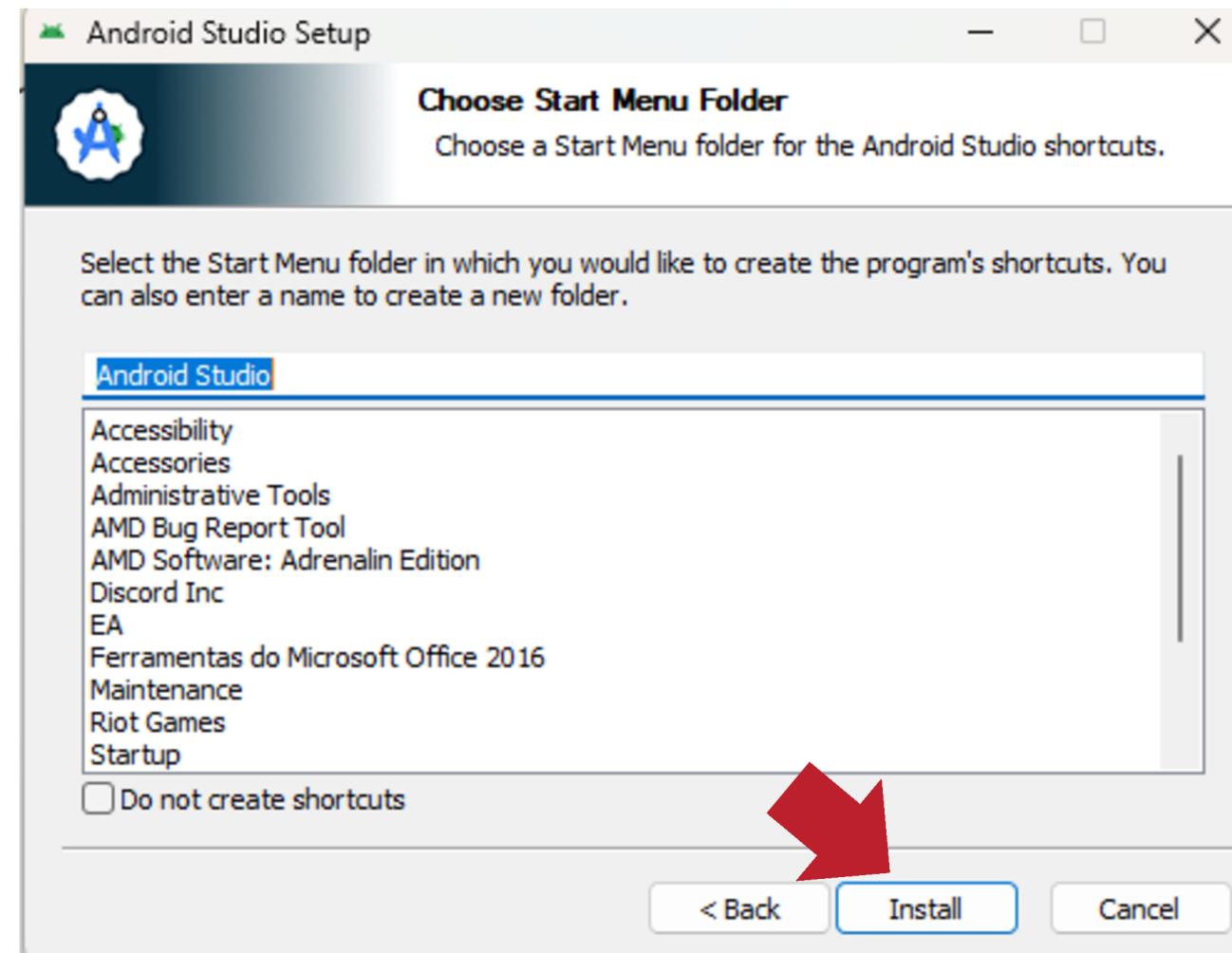
## 2.2 - Instalando o Android Studio



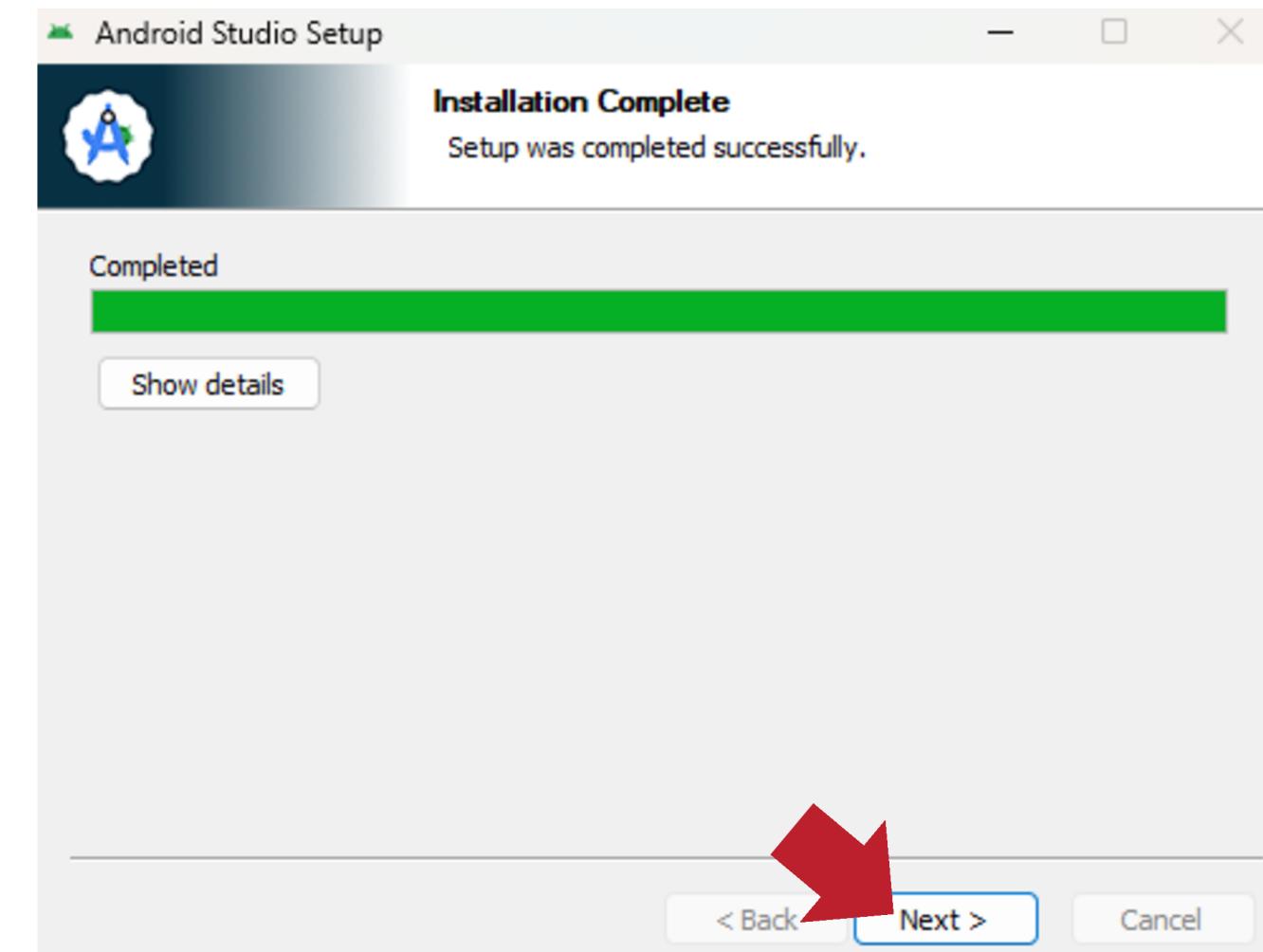
# TUTORIAL DE INSTALAÇÃO



## 2.3 - Instalando o Android Studio



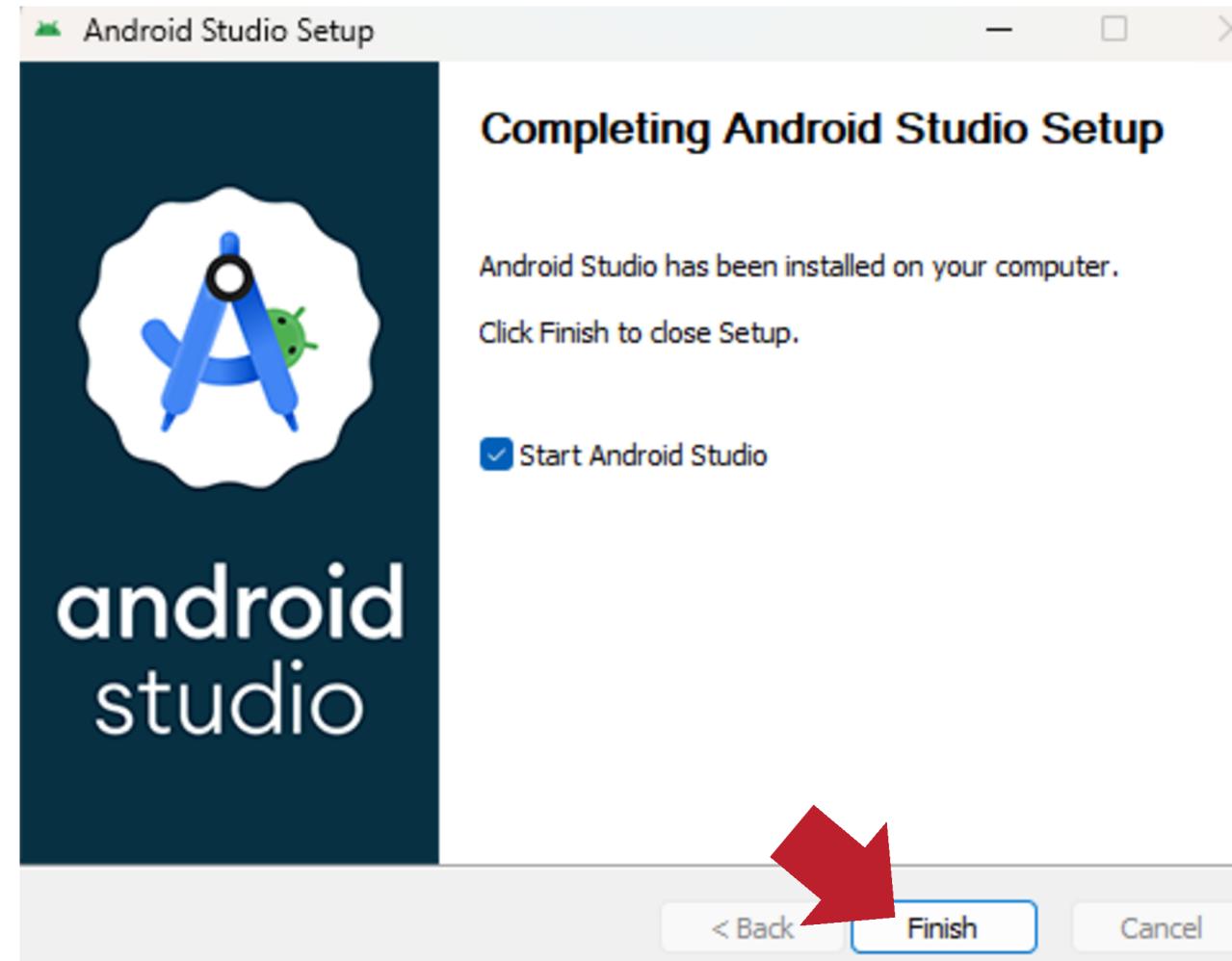
## 2.4 - Instalando o Android Studio



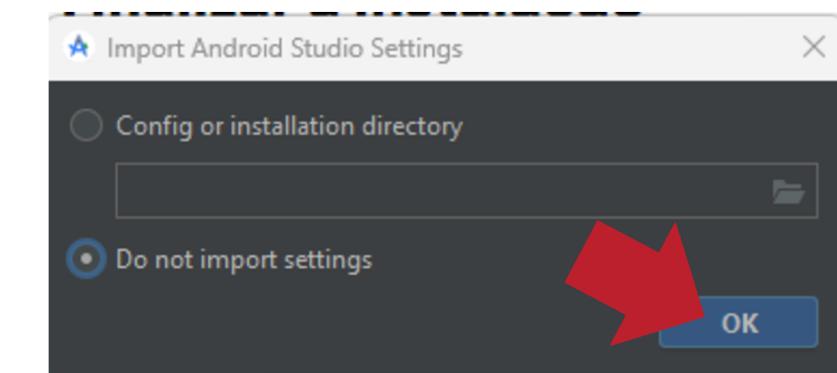
# TUTORIAL DE INSTALAÇÃO



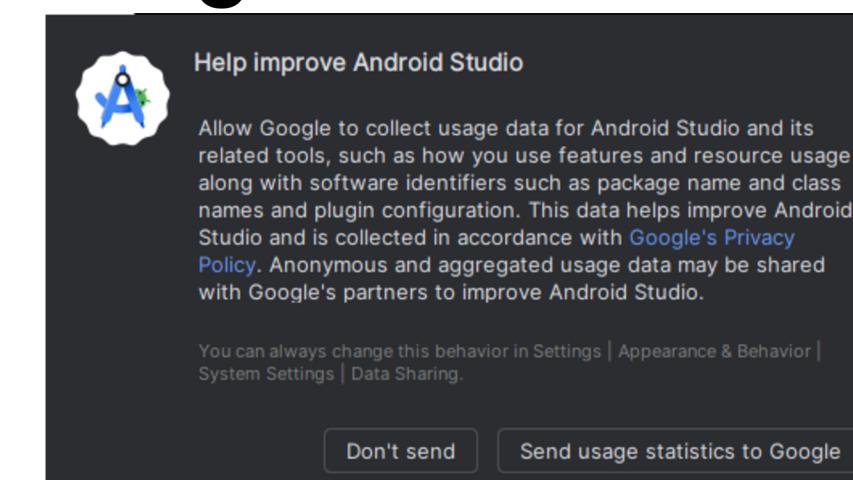
## 3 - Finalizar a instalação



## 4 - Configurando o ambiente



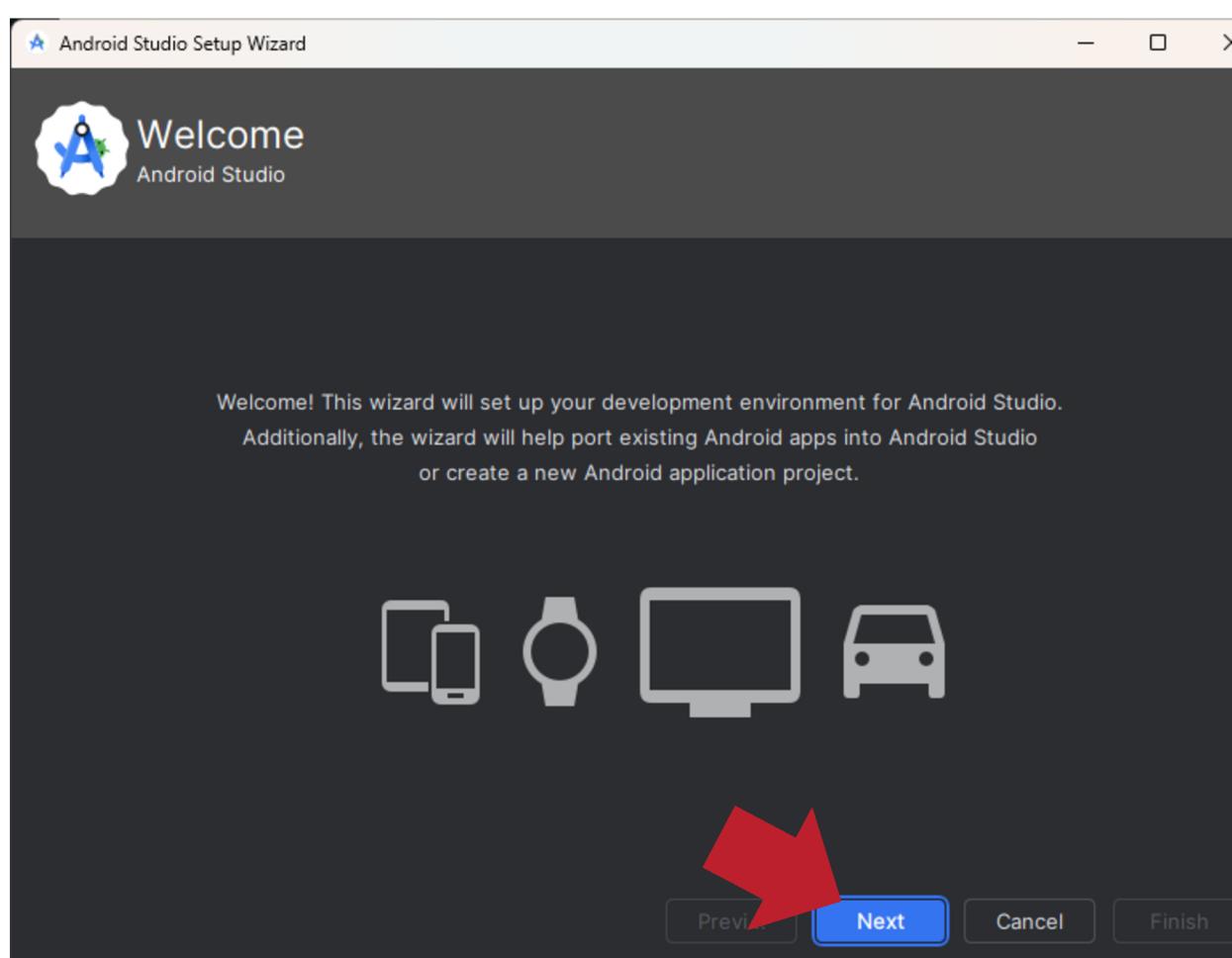
### 4.1 - Configurando o ambiente



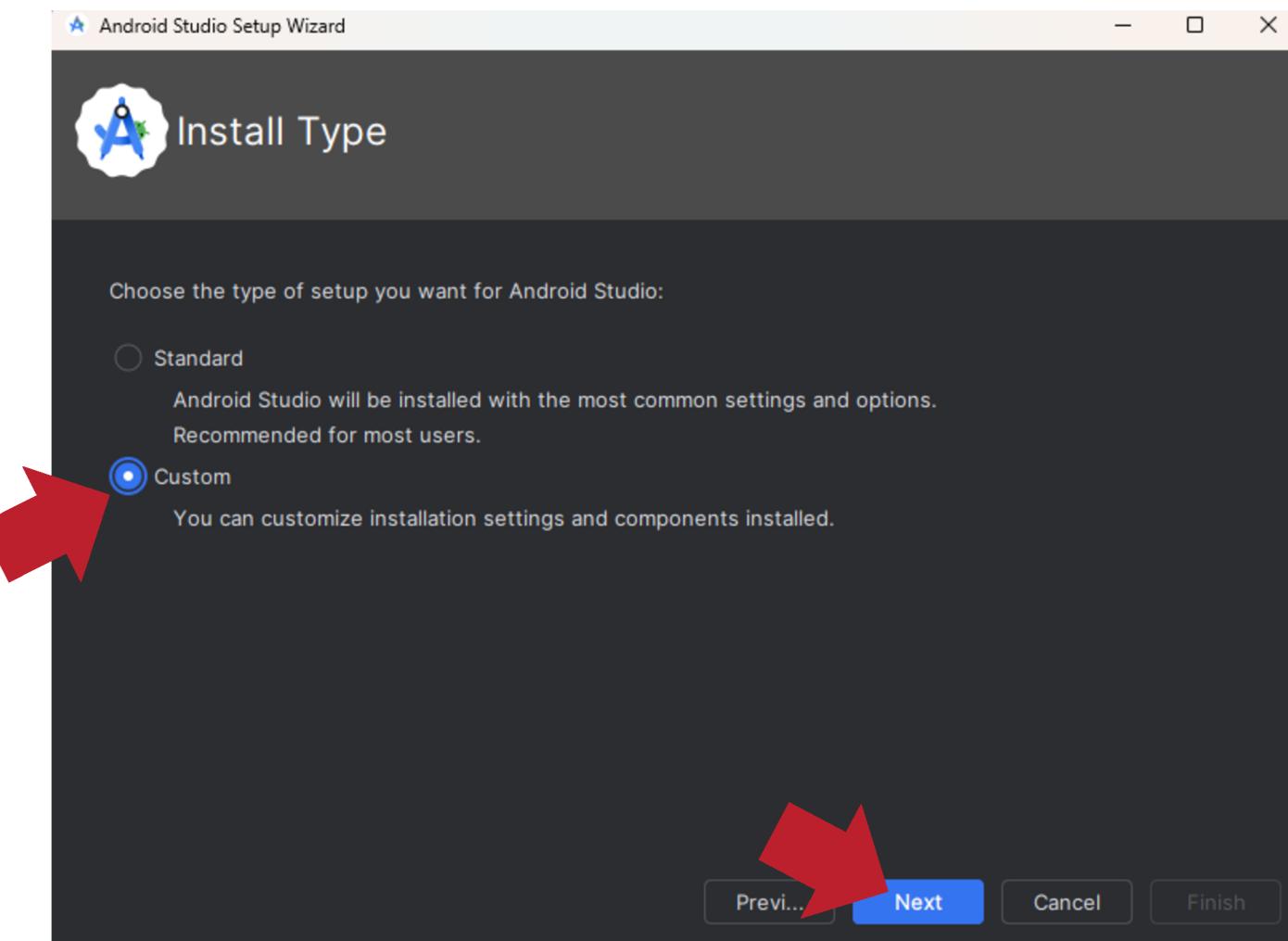
# TUTORIAL DE INSTALAÇÃO



## 4 - Configurando o ambiente



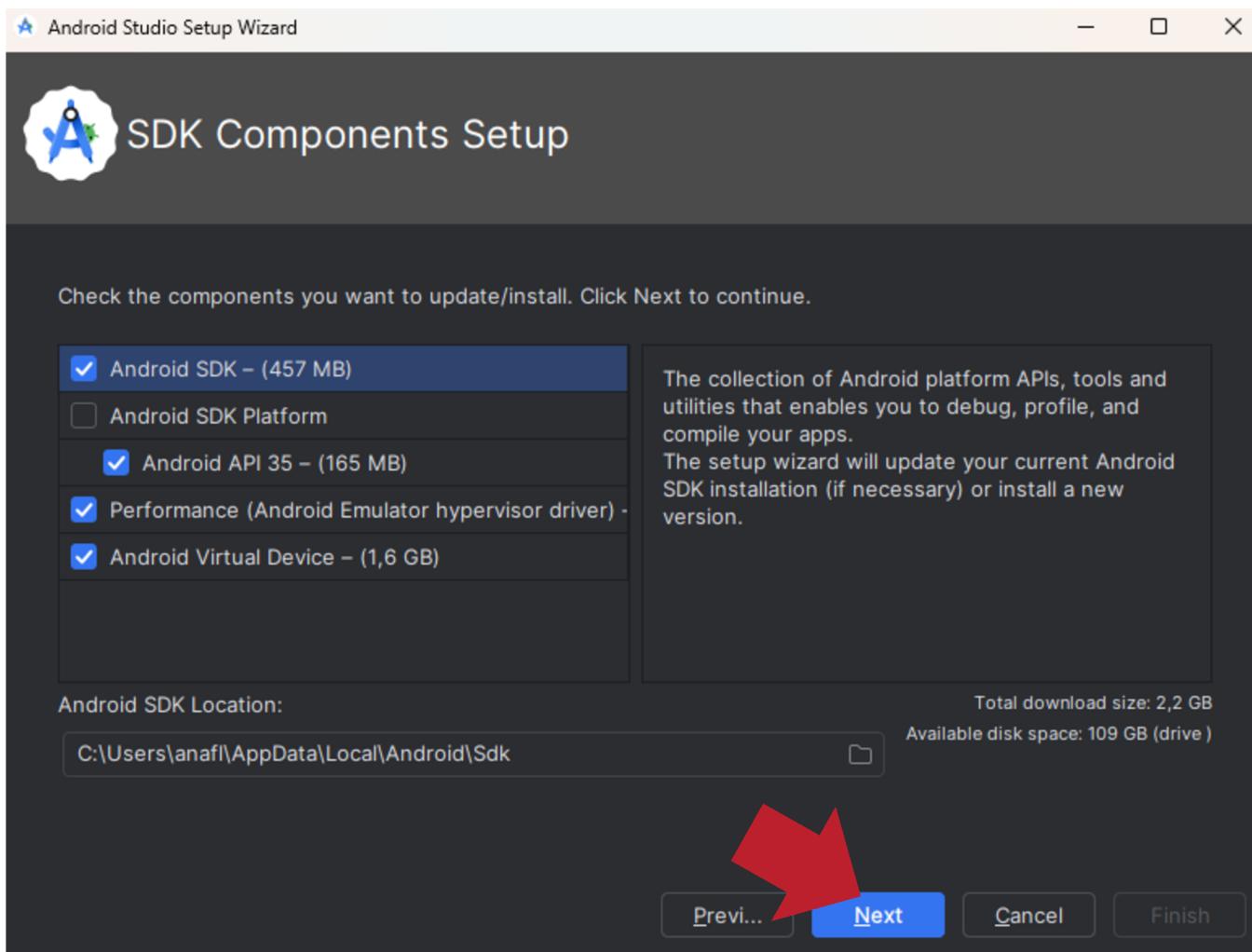
## 4 - Configurando o ambiente



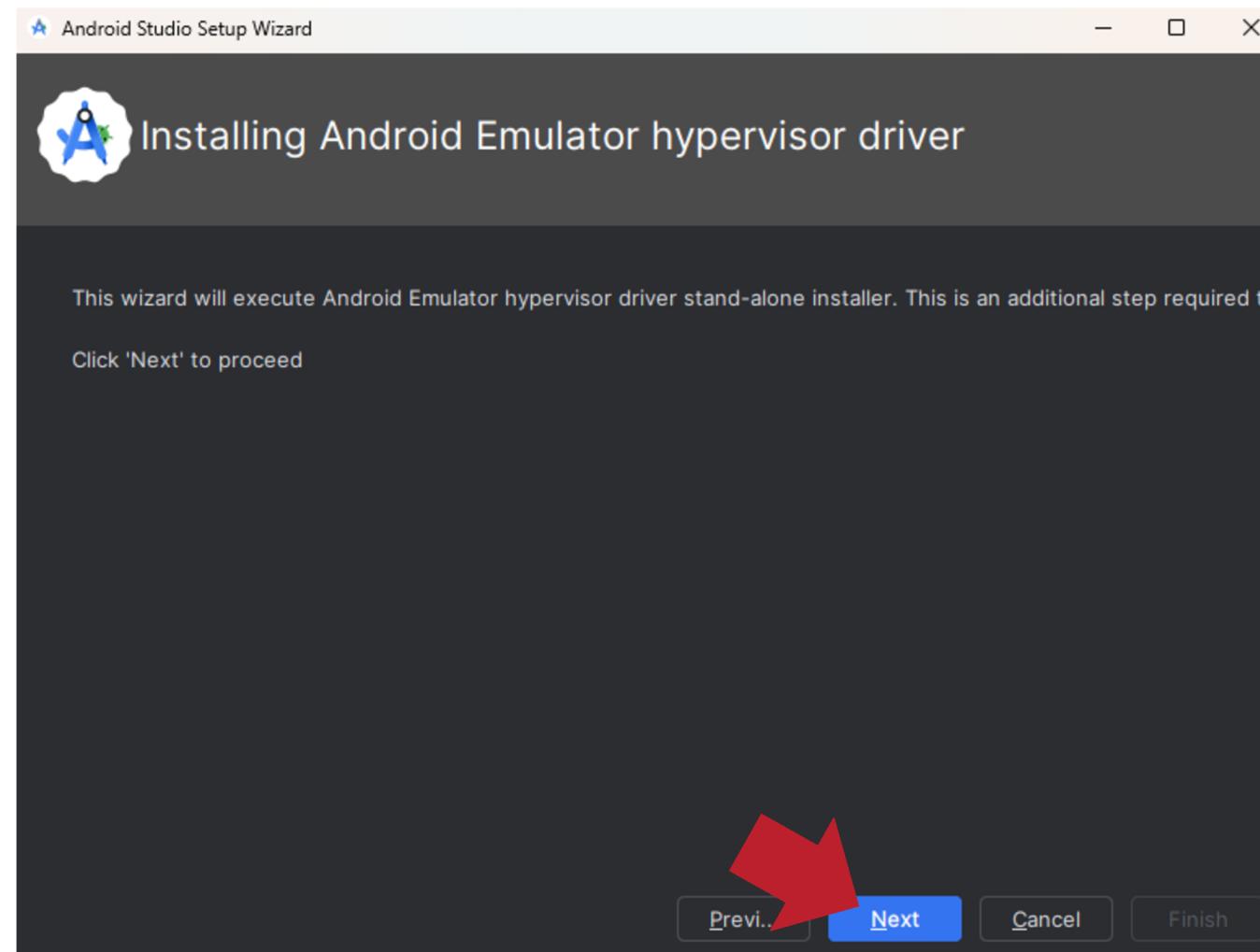
# TUTORIAL DE INSTALAÇÃO



## 4.2 - Configurando o ambiente



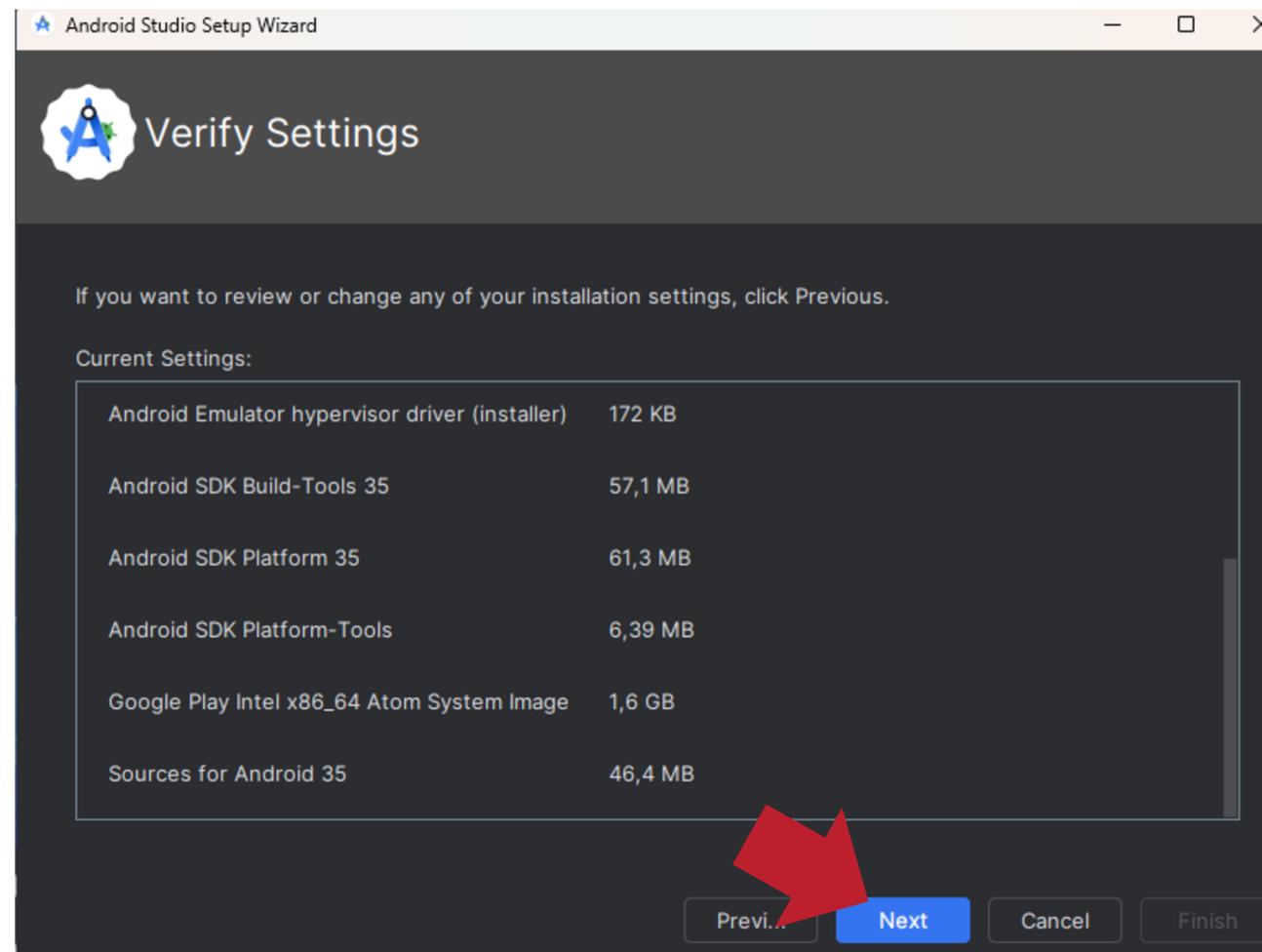
## 4.3 - Configurando o ambiente



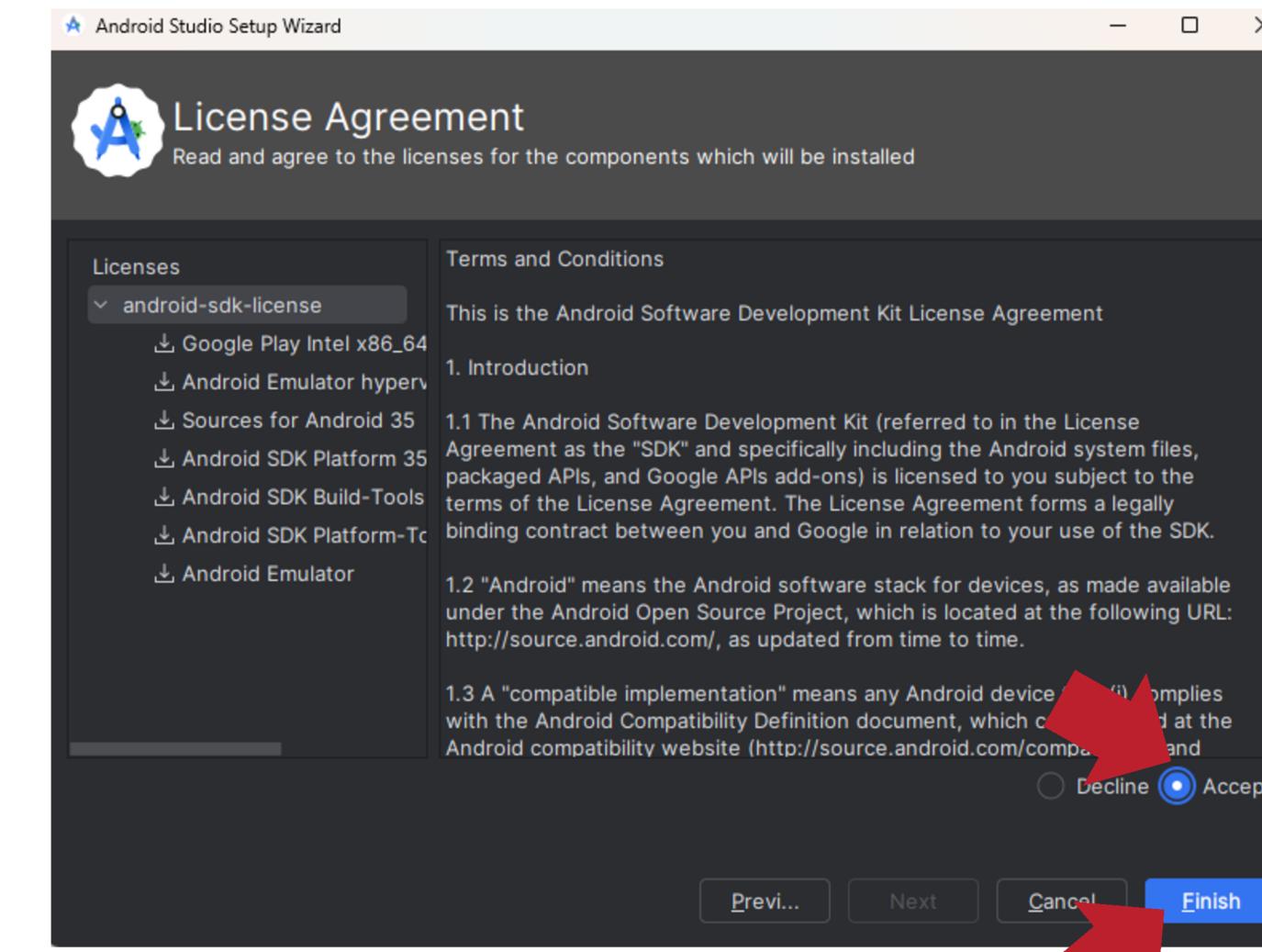
# TUTORIAL DE INSTALAÇÃO



## 4.4 - Configurando o ambiente



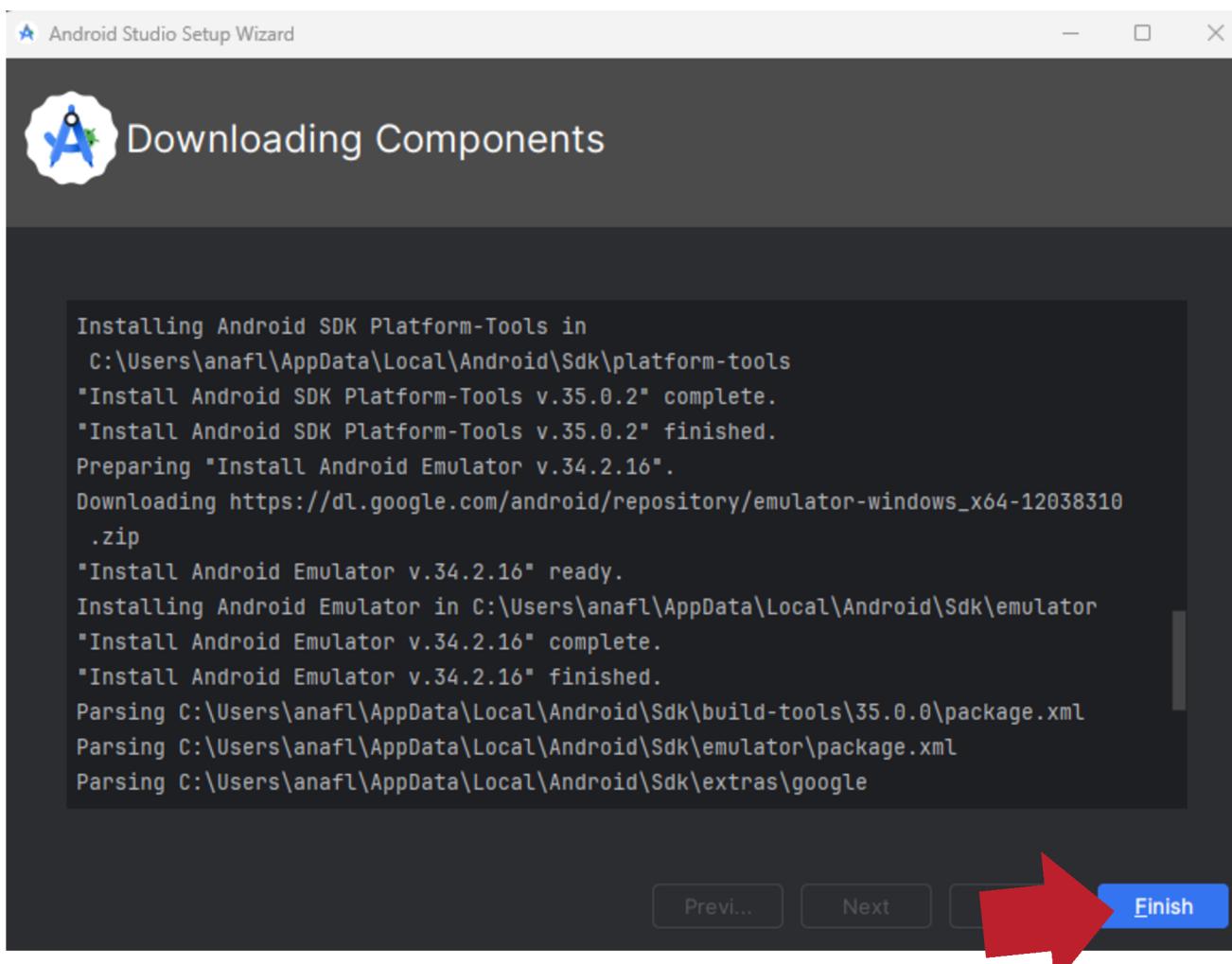
## 4.5 - Configurando o ambiente



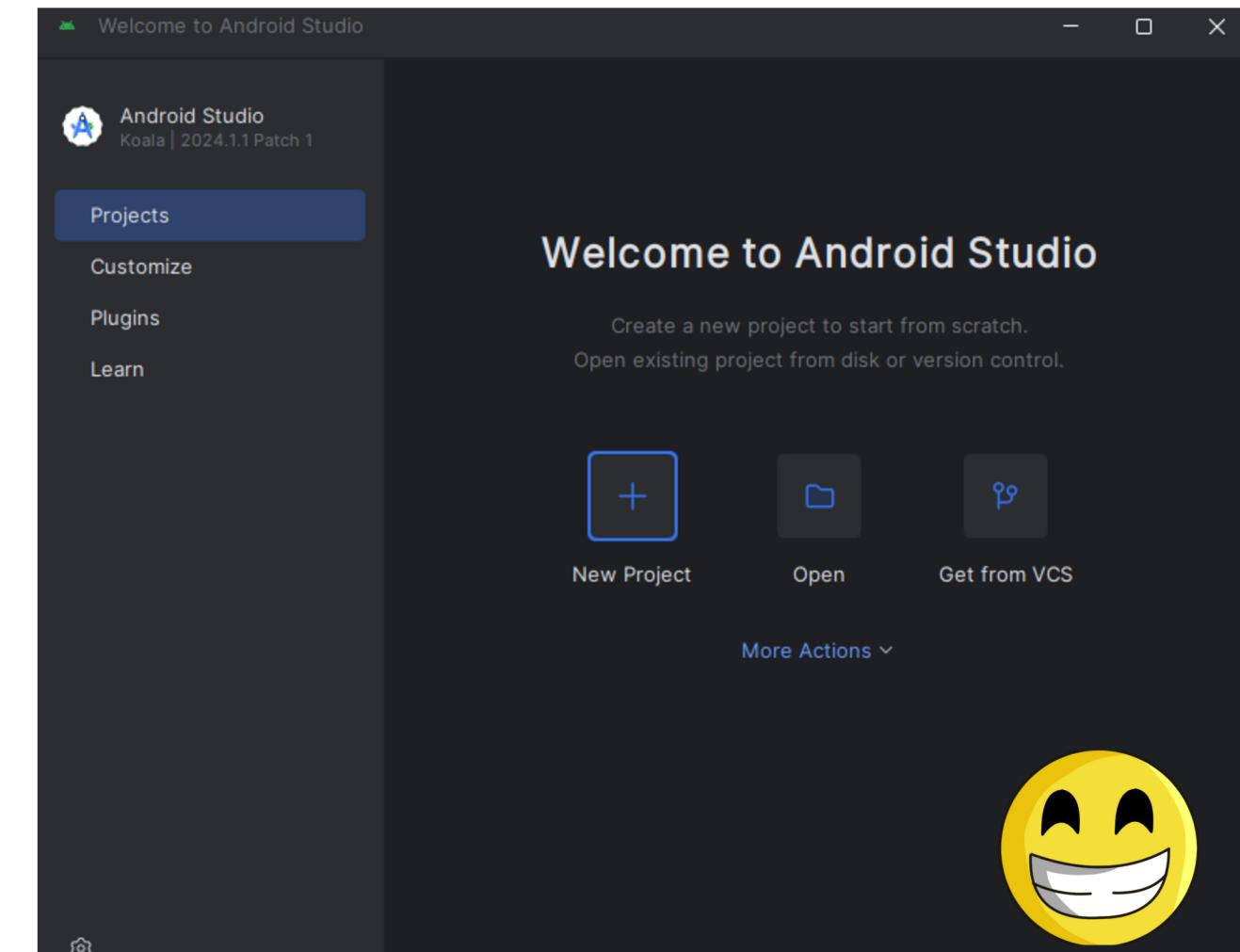
# TUTORIAL DE INSTALAÇÃO



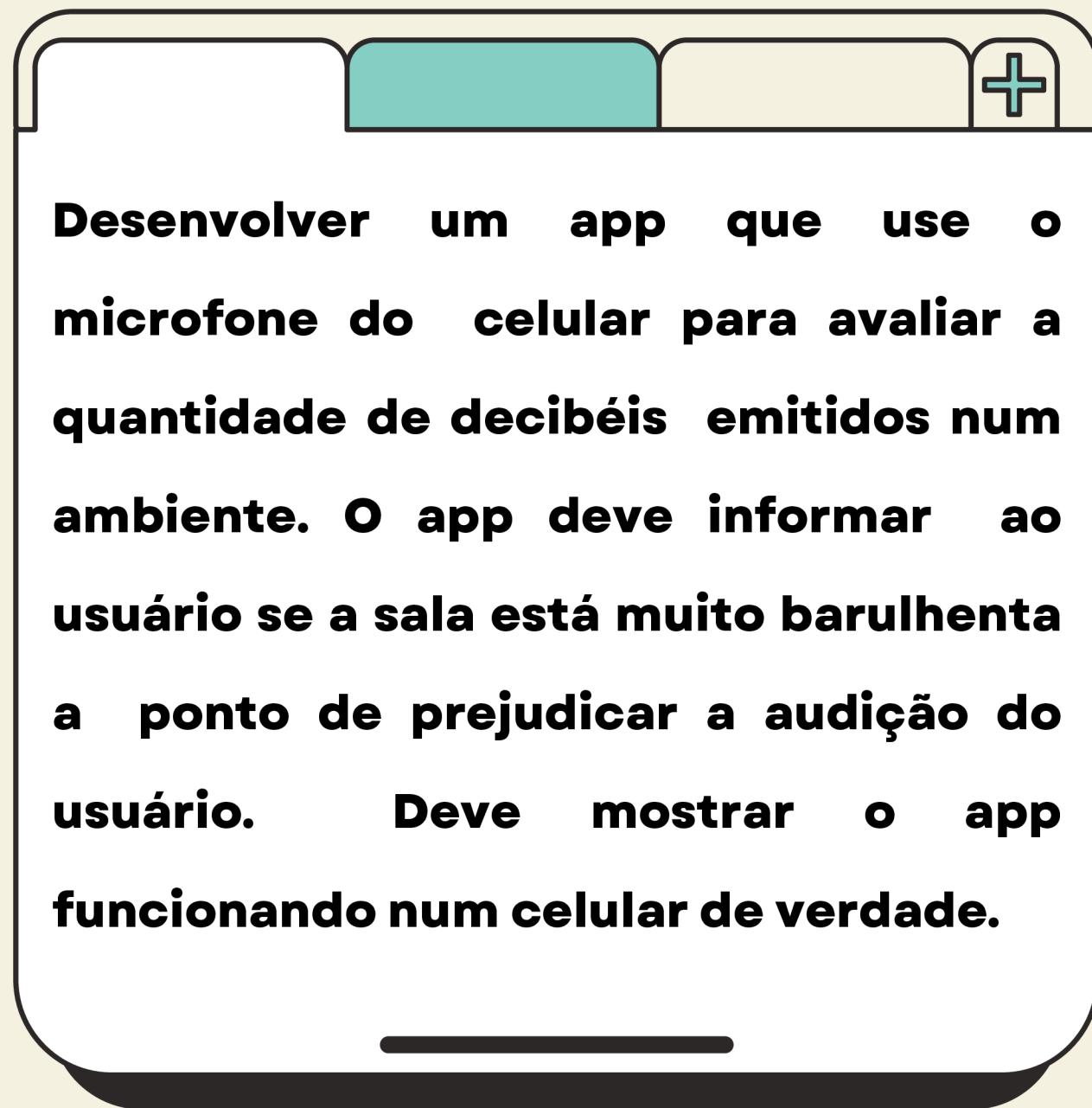
## 4.6 - Configurando o ambiente



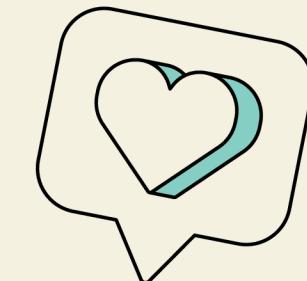
## 5 - Instalação concluída!

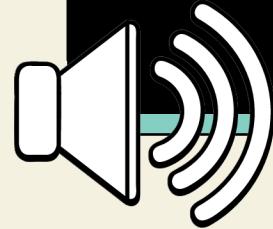


# DESAFIO



**Desenvolver um app que use o microfone do celular para avaliar a quantidade de decibéis emitidos num ambiente. O app deve informar ao usuário se a sala está muito barulhenta a ponto de prejudicar a audição do usuário. Deve mostrar o app funcionando num celular de verdade.**





# TABELA DE DECIBÉIS



Intensidade do ruído em dB.

TEMPO MÁXIMO DE EXPOSIÇÃO (SEM PROTECÇÃO) ANTES QUE SE PRODUZA LESÃO
- De 120 a 140 dB : Alguns segundos bastam para produzir lesão irreversível
- 107 dB : 1 minutos por dia
- 101 dB : 4 minutos por dia
- 95 dB : 15 minutos por dia
- 92 dB : 30 minutos por dia
- 86 dB : 2h horas por dia
- 80 dB : 8h horas por dia



■ Sons excepcionais:  
Lesão irreversível.

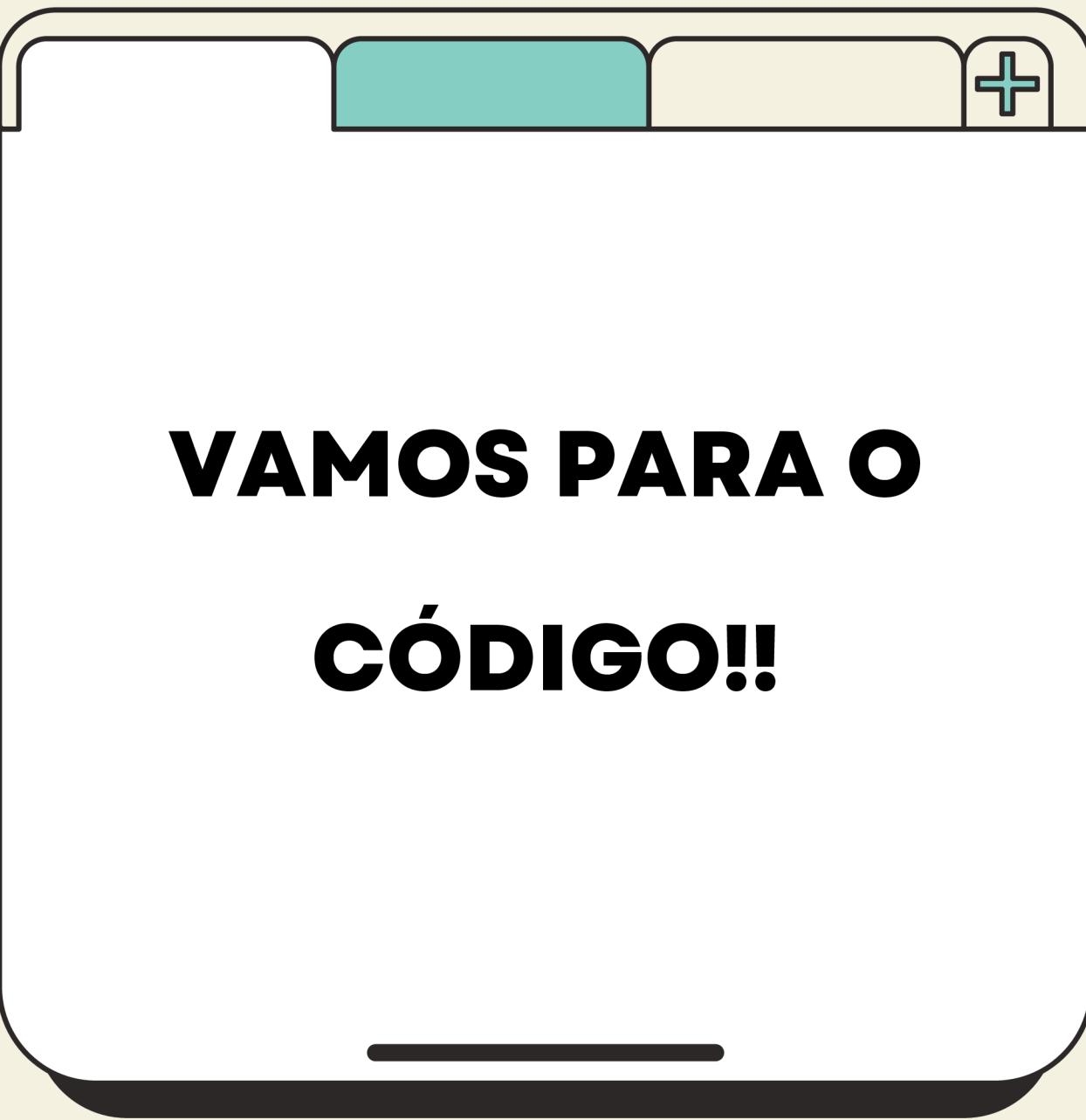
■ Perigo: Sons lesivo.

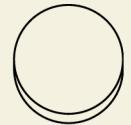
■ Limiar do som lesivo

■ Sem risco.

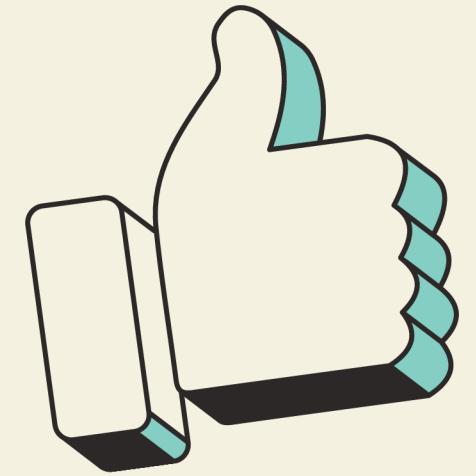
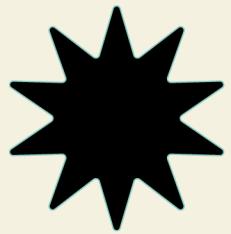
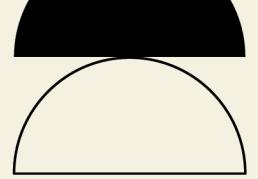
Fonte: <https://www.cochlea.org/po/ruido>

# DESAFIO

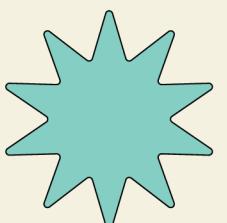
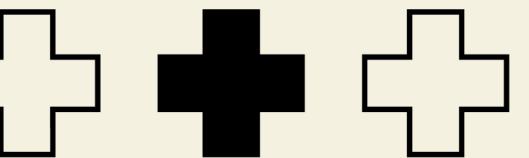


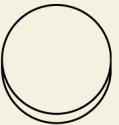


KOTLIN



# CONCLUSÃO





KOTLIN

