



Desenvolvimento
Mobile 1
Aula 01

Prof. Me Daniel Vieira

SENAI

Agenda

- 1- Apresentação
- 2 - Projetos realizados
- 3 - Ementa do curso
- 4- Capacidades técnicas e Socioemocionais
- 5 - Calendário das aulas
- 6 - Critérios de avaliação
- 7 - Aplicativos Mobile
- 8 - Ciclo de desenvolvimento de um aplicativo
- 9 - Custo de desenvolvimento de um aplicativo
- 10 - Aplicativo Mobile - Nativo ou Híbrido

Agenda

11- Framework

12 - Flutter

13 - Flutter - Desenvolvimento

14- Flutter - Recursos necessários

15 - Instalação do Git

16 - Instalação do Flutter

17 - Instalação do Android Studio

18 - Hello World

Apresentação

Doutorando e Mestre em Engenharia Elétrica - FEEC / Unicamp

Engenheiro Eletricista - UniMetrocamp Wyden

Técnico de Mecatrônica - Senai Roberto Mange

Áreas de atuação :

Projetos eletrônicos

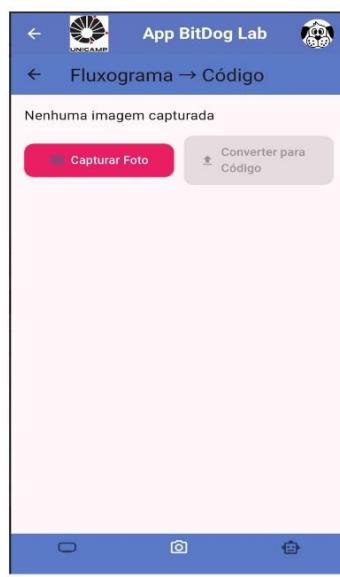
Hardware e firmware

Desenvolvimento de aplicativos C++, C#, Python, Mobile

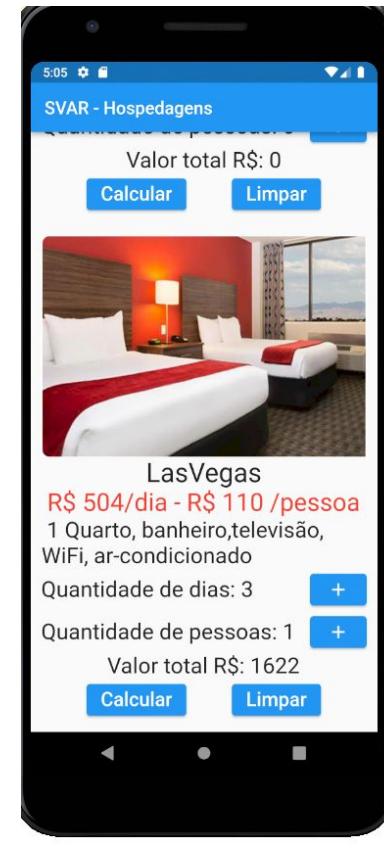
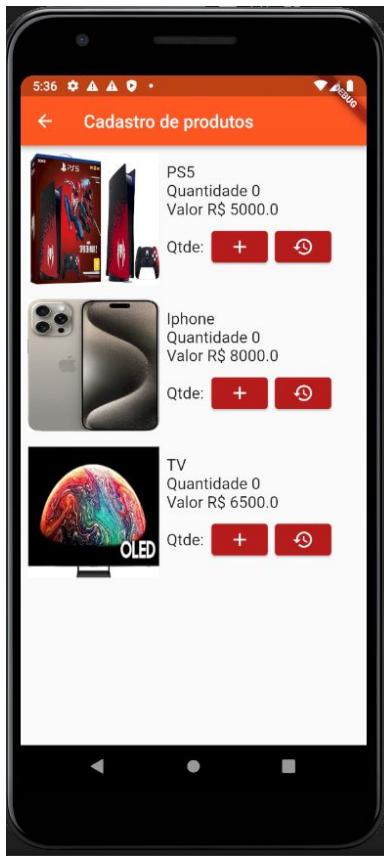
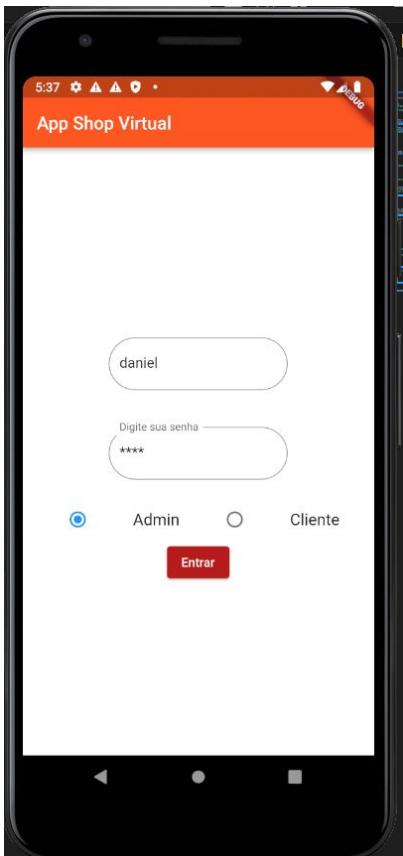
Análise de dados

Pesquisa com IA, Machine Learning, Lógica Fuzzy

Projetos realizados



Projetos realizados



Projetos realizados

The screenshot shows a Flutter development setup. On the left is a code editor with a dark theme, displaying Dart code for a 'main.dart' file. The code handles a 'main.dart' route named '_HomeState' and its 'publicarPost' method. It uses a Scaffold with an AppBar titled 'App post'. The body contains a Column with a main axis alignment of 'spaceEvenly' and children including a Text widget and four ElevatedButtons labeled 'Ler', 'Enviar', 'Deletar', and 'Alterar'. Below the code editor is a status bar with icons for battery, signal, and volume. At the bottom are navigation and search bars.

```
File Edit Selection View Go Run Terminal Help ← → Untitled (Workspace)
db.json main.dart 3
aula10_post_json > lib > main.dart > _HomeState > publicarPost
121
122
123 @override
124 Widget build(BuildContext context) {
125   return Scaffold(
126     appBar: AppBar(
127       title: Text("App post"),
128     ), // AppBar
129     body: Center(
130       mainAxisSize: MainAxisSize.spaceEvenly,
131       children: [
132         Text("Aplicativo post http", style: TextStyle(fontSize: 18)),
133         ElevatedButton(onPressed: _consultaapi, child: Text("Ler")),
134         ElevatedButton(onPressed: publicarPost, child: Text("Enviar")),
135         ElevatedButton(onPressed: delete_item, child: Text("Deletar")),
136         ElevatedButton(onPressed: modificar_item, child: Text("Alterar")),
137         Column(
138           children: [
139             Column(
140               children: produtos.map((produto) => Text("${produto.nome}- ${produto.valor}"), style: TextStyle(fontWeight: FontWeight.bold)).toList(),
141             ) // Column
142           ],
143         ) // Column
144       ],
145     ),
146   ); // Center // Scaffold
147 }
148
149 }
```

Aplicativo post http

Ler

Enviar

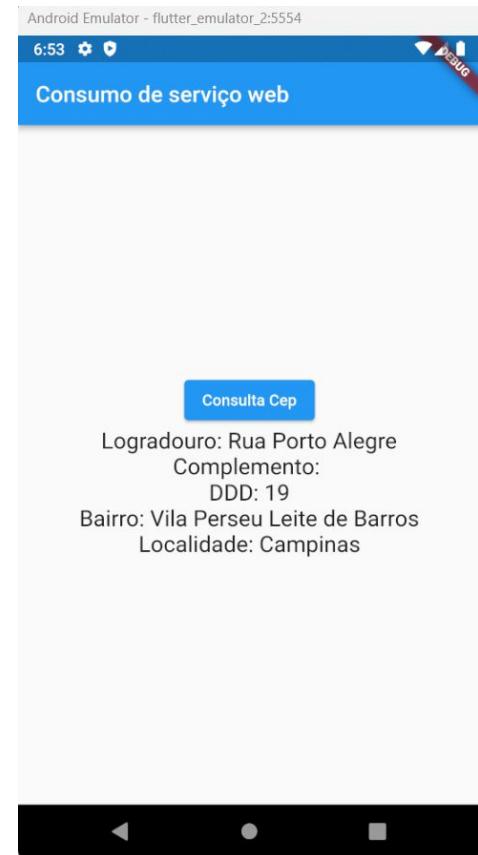
Deletar

Alterar

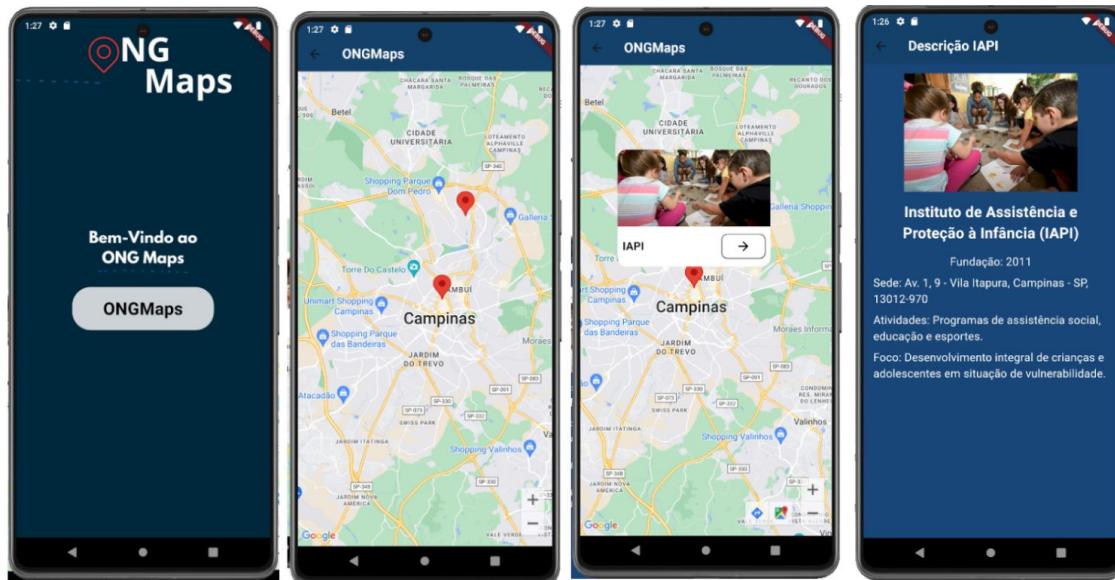
Ipad- 5000
Iphone- 8000
Notebook- 5500
TV- 4500

Ln 87, Col 16 Spaces: 2 UTF-8 CRLF {} Dart Emulador_novo (android-x86 emulator) ⚡ Prettier

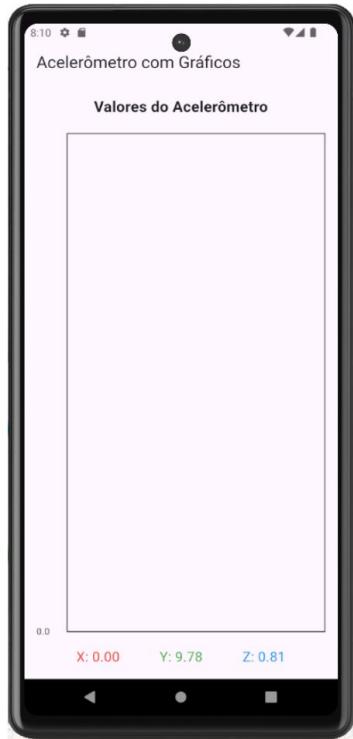
Projetos realizados



Projetos realizados



Projetos realizados



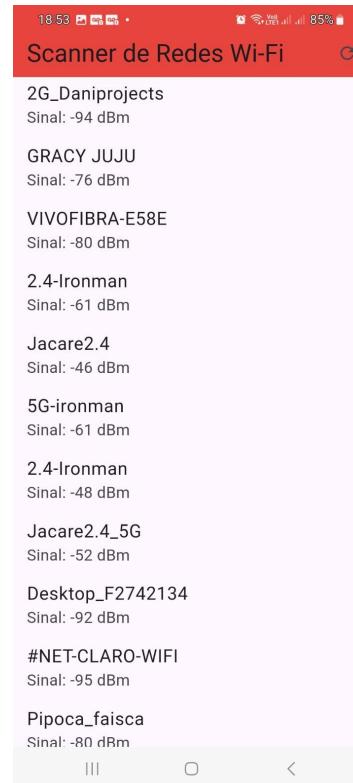
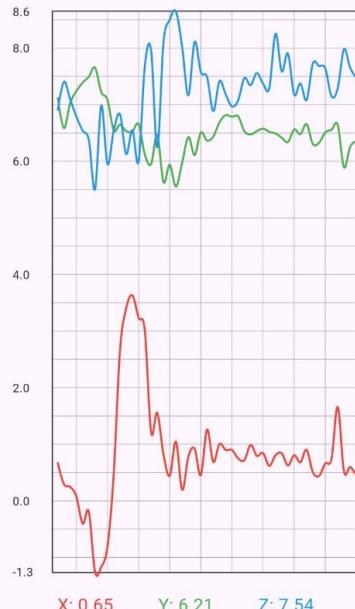
Acelerômetro com Gráficos

Valores do Acelerômetro



Acelerômetro com Gráficos

Valores do Acelerômetro



Ementa do curso

Carga horária: 80 horas

Desenvolvimento Mobile

Plataformas Mobile

Android
IOS
Ciclo de vida da aplicação
Flutter
Configuração do ambiente de desenvolvimento
Características
Android SDK e IOS
Emulador

Dart

Características do Framework
Flutter
Dart e sua sintaxe
Variáveis
Arrays
Operadores aritméticos e lógicos
Controle de fluxo
Repetição
Funções
Classes, Objetos

Princípios de UX

Simplicidade
Consistência
Feedback Visual
Navegação Intuitiva
Legibilidade
Foco no Usuário
Responsividade
Feedback claro de erro
Performance otimizada
Acessibilidade

Ementa do curso

Carga horária: 80 horas

Desenvolvimento Mobile

Flutter

Widgets
Labels,Caixas de texto, Botões
Posicionador de elementos
Mídias
Image Viewer
Video Player
API

Publicação da aplicação

Preparação e testes
Geração da aplicação de produção (build)
Técnica

Projetos

Hello world
Exercícios
Atividade Somativa

Capacidades técnicas/ Socioemocionais

1. Configurar o ambiente de desenvolvimento para aplicações Mobile.
2. Aplicar tecnologias Mobile de acordo com o sistema operacional.
3. Elaborar programação Mobile de acordo com padrões de desenvolvimento.
4. Definir os frameworks para programação Mobile em relação ao sistema operacional e aos recursos de hardware.

1. Aprendizagem ativa e estratégias de aprendizagem
2. Ética
3. Resolução de problemas complexos

Calendário das aulas

Calendário das aulas

Calendário das aulas

Calendário das aulas

Metodologias das aulas

Aulas teóricas, expositivas, exercícios e projetos

Aulas de Segunda - Turma B e Quinta - Turma A

Horário : 18:45 - 22:20

Intervalo: 20:50 - 21:05

Somativa 18/05 - 25/05 - Turma B

Somativa 21/05 e 28/05 Turma A

Recuperação 08/06 Turma B 11/06 Turma A

Situação de aprendizagem

Desenvolvimento de um Aplicativo para Monitoramento e Controle Agrícola

Com o aumento da demanda por práticas agrícolas mais eficientes e sustentáveis, o uso de tecnologia tem se tornado indispensável no campo. Uma das principais inovações nesse cenário é a integração da Internet das Coisas (IoT) e aplicativo mobile, que possibilita o monitoramento em tempo real de variáveis críticas que afetam diretamente a qualidade do plantio.

Situação de aprendizagem



Situação de aprendizagem

O aplicativo deve:

- 1. Exibir Dados em Tempo Real:**
 - a. Consumir dados de um dispositivo IoT que forneça informações sobre temperatura, umidade do ar e umidade do solo, vazão de agua.
 - b. Apresentar essas informações em uma interface amigável e organizada.
- 2. Controle Manual:**
 - a. Implementar um botão para acionamento manual de uma bomba de irrigação, simulando a interação com um sistema de automação.
- 3. Integração com ChatBot especialista em agricultura e irrigação**

O chatbot deve responder:

- “Como reduzir gasto de energia na irrigação?”

Situação de aprendizagem

- “Qual setor mais irrigou hoje e por quê?” (usa dados da API)
- “O que fazer quando a umidade não sobe mesmo irrigando?” (base técnica + hipóteses)
- “Sugira ações para economizar água mantendo produtividade” (regras + boas práticas)

Regra de qualidade: se não tiver dados/fonte, o bot deve dizer que **não encontrou** e sugerir a consulta correta.

Situação de aprendizagem

Requisitos Técnicos:

1. A interface deve ser clara e responsiva.
2. O consumo de dados da API deve ser realizado com boas práticas de programação, utilizando o pacote http ou equivalente no Flutter.

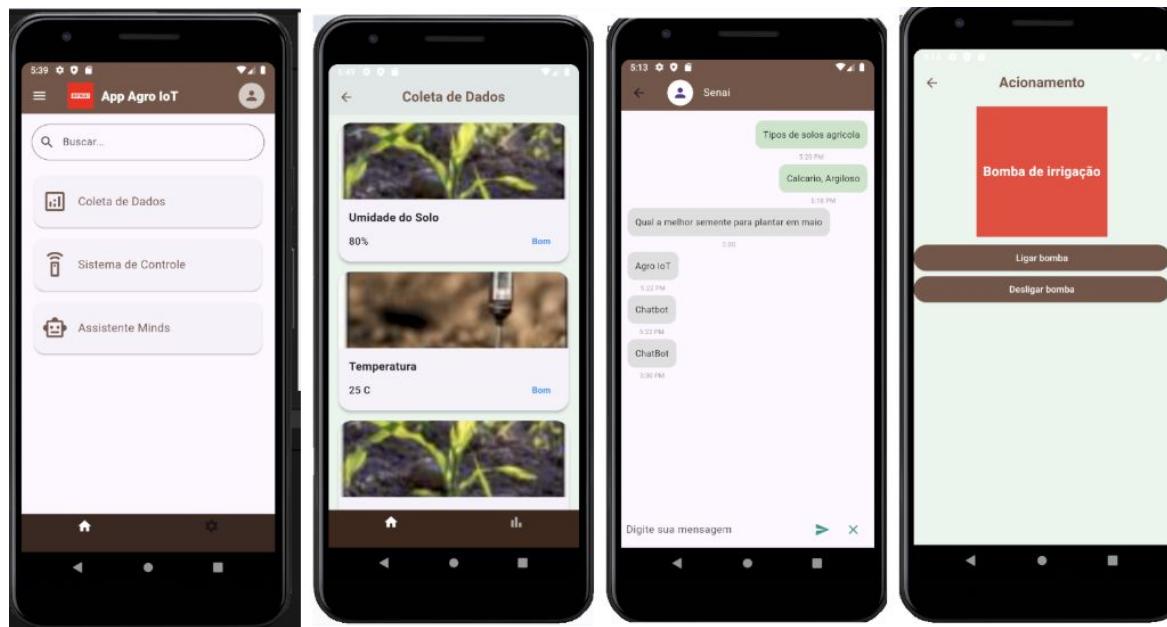
O acionamento da bomba deve ser controlado por meio de uma requisição HTTP (POST) para um endpoint específico.

Situação de aprendizagem

Entrega:

- O código-fonte do aplicativo desenvolvido em Flutter será entregue via repositório Git Hub.
- Um arquivo README explicando as funcionalidades do aplicativo e como configurá-lo para consumir a API

Subir o projeto no GitHub e entregar o código via TEAMS



Critérios de avaliação

INSTRUMENTO DE REGISTRO DE AVALIAÇÃO						
Natureza dos Critérios	Capacidades Técnicas	Critérios de avaliação		Avaliação		
		Crítico	Desejável	Aluno	Professor	
Competências Técnicas	1. Configurar o ambiente de desenvolvimento para aplicações Mobile	O ambiente de desenvolvimento VSCode, Flutter, Dart foi configurado corretamente?				O aluno criou a classe Stateless chamada Home e a classe Statefull para atualizar os dados dos sensores na tela ?
		O aluno foi capaz de solucionar problemas comuns na configuração do ambiente de desenvolvimento como: erros de variáveis de ambiente ?				
	2. Aplicar tecnologias Mobile de acordo com o sistema operacional	O aluno foi capaz de configurar o Framework Flutter na IDE VSCode para gerar o aplicativo para IOS ou Android seguindo as orientações?				O aluno foi capaz de criar um aplicativo com o uso de cores, tipografia, espaçamento, hierarquia visual com um visual amigável e de fácil utilização ?
		O aluno aplicou os princípios de UX para cria uma interface gráfica intuitiva e amigável ?				
	3. Elaborar programação Mobile de acordo com os padrões de desenvolvimento	O aluno foi capaz de elaborar o software legível, estruturado utilizou o Git para versionamento ?				
		O aluno foi capaz de desenvolver o código de forma organizada e modularizada?				

Critérios de avaliação

Página 2 de 21

Capacidades socioemocionais	1. Aprendizagem ativa e estratégias de aprendizagem	O aluno foi capaz de consultar a documentação do framework Flutter no site para resolver problemas ?		
		O aluno foi capaz de desenvolver o aplicativo utilizando os princípios de aprendizagem baseada em projetos ?		
	1. Ética	Os aplicativos são de design autoral ?		
		O aluno foi capaz de desenvolver o aplicativo seguindo as leis LGPD ?		
	2. Resolução de problemas complexos	O aluno foi capaz de solucionar os problemas encontrados durante o desenvolvimento do aplicativo de logica, design?		

Critérios de avaliação

TABELA DE NÍVEIS DE DESEMPENHO

Critérios de Avaliação	Nível de desempenho	Conversão em notas
Atingiu todos os critérios críticos e todos desejáveis	14	100
Atingiu todos os critérios críticos e 6 dos desejáveis	13	90
Atingiu todos os critérios críticos e 5 dos desejáveis	12	80
Atingiu todos os critérios críticos e 4 dos desejáveis	11	70
Atingiu todos os critérios críticos e 3 dos desejáveis	10	65
Atingiu todos os critérios críticos e 2 dos desejáveis	9	60
Atingiu todos os critérios críticos e 1 dos desejáveis	8	55
Atingiu todos os critérios críticos e nenhum desejável	7	50
Atingiu 5 critérios críticos e quaisquer desejáveis	6	40
Atingiu 4 critérios críticos e quaisquer desejáveis	5	30
Atingiu entre 1 e 3 critérios críticos e quaisquer desejáveis	4	20
Não atingiu nenhum critério crítico e quaisquer desejáveis	1	10
Não atingiu nenhum critério	0	0

Aplicativos Mobile



Etapas de desenvolvimento de um aplicativo



Qual o custo para desenvolver um aplicativo ?

Profissional	Quantidade de Horas	Valor/Hora	Total
Programador Android	100	75	R\$ 7.500
Programador iOS	100	80	R\$ 8.000
Gerente de Projetos	20	100	R\$ 2.000
Designer	30	35	R\$ 1.050
Desenvolvedor Web	10	50	R\$ 500
Desenvolvedor Backend	25	40	R\$ 1.000

Total do projeto: R\$ 20.050,00

Prazo Médio: 3 meses

Qual o custo para desenvolver um aplicativo ?

Profissional	Quantidade de Horas	Salário	Encargos Sociais	Tempo necessário	Total
Programador Android	130	R\$ 3.980	R\$ 2.100	1 mês	R\$ 6.080
Programador iOS	130	R\$ 5.000	R\$ 2.644	1 mês	R\$ 7.644
Gerente de Projeto	26	R\$ 8.972	R\$ 4.745	1/6 mês	R\$ 2.286
Designer	39	R\$ 3.009	R\$ 1.600	1/4 mês	R\$ 1.152
Desenvolvedor Web	13	R\$ 2.801	R\$ 1480	1/8 mês	R\$ 535
Desenvolvedor Backend	33	R\$ 2.500	R\$ 1.322	1/4 mês	R\$ 955

Custos de rescisão contratual aproximadamente **R\$ 7.100**.

Total do projeto: R\$ 18.663,00 / mês

Prazo Médio: 2 meses

Subtotal: R\$ 37.326 + R\$ 7.100

Total Estimado : R\$ 44.426

Aplicativos Mobile - Nativo ou híbrido

Dynatac 8000x Dynatac Flip IBM Simon Nokia S10 Sharp J –SH04 1º Iphone Android



Tablets Assistentes virtuais Ascensão dos pagamentos móveis 5G e Smartphones dobráveis IA + Smartphones



Aplicativos Mobile - Nativo ou híbrido

1983 – O Primeiro Telefone Móvel Chega ao Mercado

Em 1983, a Motorola revolucionou a comunicação ao lançar o DynaTAC 8000x, um aparelho robusto, com 33 cm de altura e pesando 800 gramas. Sua autonomia de bateria era de apenas uma hora de conversa, e seu custo de quase 4 mil dólares o tornava inacessível para muitos. Mesmo assim, ele marcou o início de uma nova era.

1989 – A Revolução do Flip

O MicroTAC 9800X, lançado em 1989, trouxe um design inovador com sua tampa flip, tornando os celulares mais compactos e práticos. Essa inovação abriu caminho para os celulares de bolso.

1994 – Telas Sensíveis ao Toque

O IBM Simon foi pioneiro ao introduzir telas sensíveis ao toque e funcionalidades como calendário e bloco de notas. Apesar da visão futurista, seu alto custo e baixa aceitação limitaram sua popularidade.

1996 – Primeiros Passos para a Internet Móvel

O Nokia 9000 Communicator foi o primeiro celular a oferecer acesso à internet, ainda que de forma limitada e cara. Seu design lembrava um pequeno notebook, antecipando funcionalidades dos smartphones.

1998 – Telas Coloridas Chegam aos Celulares

Com o lançamento do Siemens S10, em 1998, as telas coloridas fizeram sua estreia. A Nokia rapidamente superou essa inovação com o modelo 9210, introduzindo telas de alta qualidade.

Aplicativos Mobile - Nativo ou híbrido

2001 – A Era das Câmeras

O Sharp J-SH04 trouxe a câmera integrada aos celulares, permitindo tirar fotos e compartilhá-las instantaneamente. Foi o primeiro passo para transformar os celulares em ferramentas multimídia.

2007 – O iPhone e a Revolução dos Smartphones

Com o lançamento do iPhone 2G, a Apple redefiniu o mercado. Sua interface totalmente touchscreen, sem teclados físicos, e o sistema iOS trouxeram os aplicativos ao centro da experiência do usuário, criando o ecossistema que hoje domina o mercado.

2008 – O Android Surge como Alternativa

O HTC Dream introduziu o Android, um sistema operacional flexível e acessível a diversas fabricantes. Essa abertura acelerou a adoção de smartphones em todo o mundo.

Aplicativos Mobile - Nativo ou híbrido

2010 – Tablets e o Avanço dos Smartphones

Com o iPad e melhorias significativas nos smartphones, como câmeras aprimoradas e processadores mais rápidos, o mercado de dispositivos móveis se consolidou como indispensável para o dia a dia.

2013 – Assistentes Virtuais Redefinem a Interação

A Siri, seguida por Google Assistant e Alexa, transformou os smartphones em assistentes pessoais, simplificando o acesso a informações e automações.

2016 – Pagamentos Móveis se Popularizam

Serviços como Apple Pay e Google Pay tornaram os smartphones também carteiras digitais, trazendo mais conveniência e segurança às transações.

2019 – 5G e Smartphones Dobráveis

A chegada do 5G trouxe conectividade ultrarrápida, enquanto os smartphones dobráveis, como o Galaxy Fold, abriram novas possibilidades de design.

Aplicativos Mobile - Nativo ou híbrido

2020 e Além – IA e Integração Total

Nos últimos anos, a inteligência artificial consolidou os smartphones como ferramentas essenciais para produtividade, saúde e entretenimento. Recursos como câmeras avançadas com IA, reconhecimento facial e assistentes de voz redefiniram o que esperamos desses dispositivos.

2024 – Um Futuro Promissor

Hoje, os smartphones são assistentes pessoais completos, essenciais para nossa vida. A evolução não para, e o futuro promete ainda mais inovações que continuarão a transformar nossa relação com a tecnologia.

Aplicativos Mobile - Nativo ou híbrido

Os recursos e a IDE utilizada para desenvolvimento Nativo são:

Onde IDE – Integrated Development Environment (Ambiente integrado de desenvolvimento) é um software que ajuda os programadores a desenvolverem código de forma organizada e eficiente.

Para o sistema operacional Android:

- Linguagens: Java e Kotlin
- Java: Uma das linguagens mais populares e robustas, utilizada desde os primeiros dias do Android.
- Kotlin: Linguagem moderna, oficializada pelo Google em 2017, projetada para ser mais expressiva, segura e produtiva.
- IDE: Android Studio.
- Ferramenta oficial do Google para desenvolvimento Android.
- Oferece emuladores, depuração, gerenciamento de recursos, suporte a Gradle (para automação de builds) e design visual para interfaces.

Exemplos de aplicativos desenvolvidos no modo nativo: Spotify, Google Maps, WhatsApp

Aplicativos Mobile - Nativo ou híbrido

Para o sistema operacional IOS:

- Linguagens: Objective C e Swift
- Objective C: Uma linguagem mais antiga e amplamente utilizada antes do lançamento do Swift.
- Swift: Lançada pela Apple em 2014, é moderna, segura, rápida com sintaxe mais limpa.
- IDE: Xcode
- Ferramenta oficial da Apple, disponível exclusivamente para macOS.
- Inclui simuladores, editor de interface visual (Storyboard) gerenciamento de certificados e ferramentas de depuração.

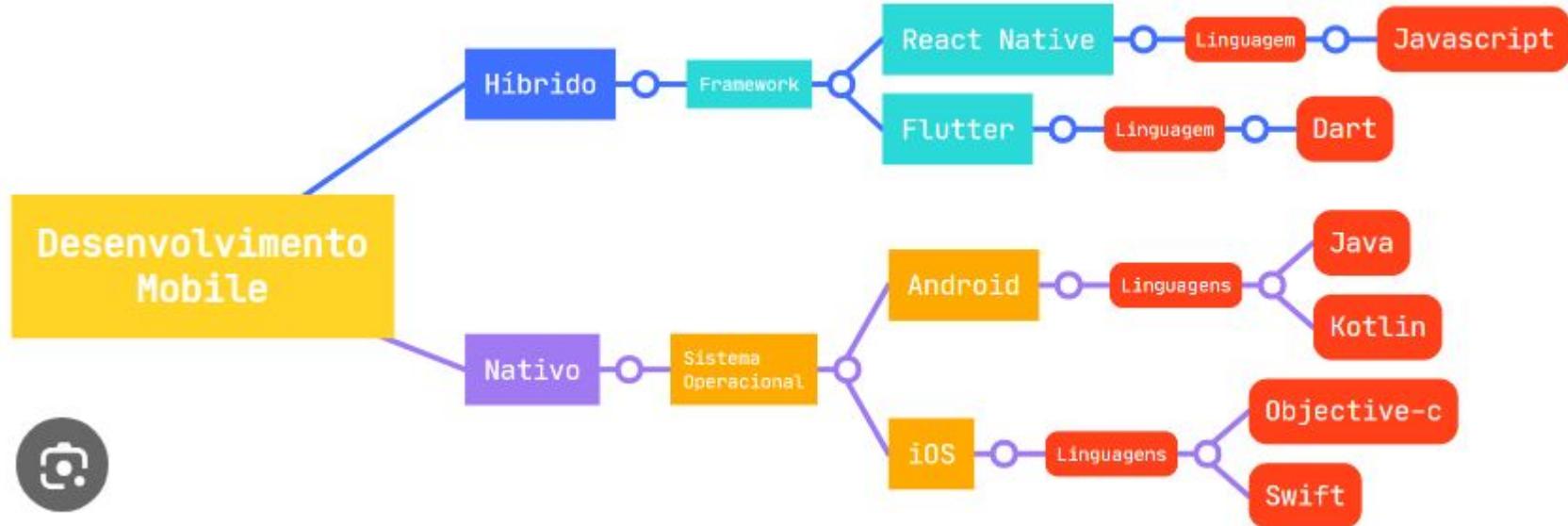
Exemplos de aplicativos desenvolvidos no modo nativo: Uber, Pinterest

Aplicativos Mobile - Nativo ou híbrido

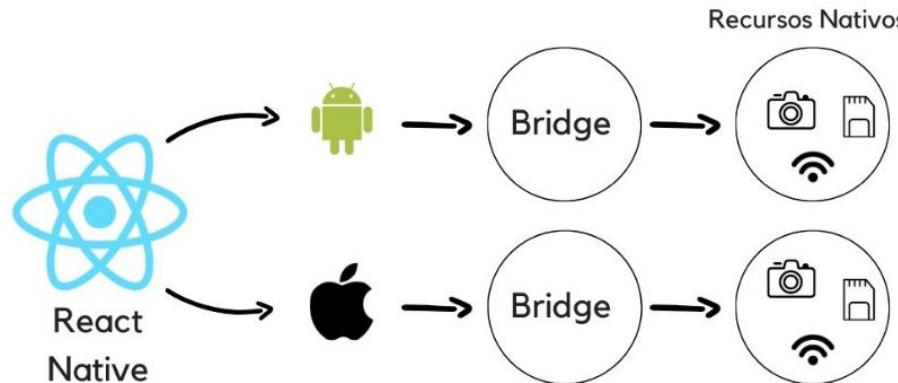
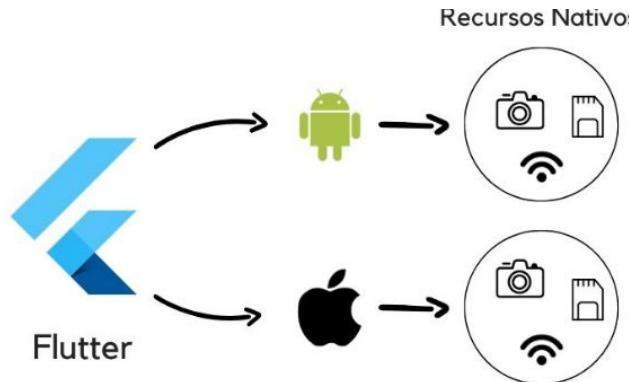
Tabela 1: Comparação entre os frameworks para desenvolvimento mobile

Aspecto	Nativo	Híbrido (React Native)	Híbrido (Flutter)	Híbrido Xamarin	Híbrido (Ionic)
Linguagem	Swift, Objective C, Kotlin, Java	JavaScript	Dart	C#, .NET	JavaScript/ TypeScript
Acesso ao hardware	Direto	Via Bridge	Direto	Direto	WebViews com Capacitor
Desempenho	Excelente	Bom, pode ser afetado pela Bridge	Excelente	Próximo ao nativo	Médio pode ser afetado pelo Webview
Interface	Totalmente nativa	Parcialmente nativa	Renderização própria	Nativa ou compartilhada	Componentes pré estilizados
Ferramenta de desenvolvimento	Xcode, Android Studio	VSCode, Android Studio	VSCode, Android Studio, IntelliJ	Visual Studio	Ionic Studio ou qualquer editor de código

Aplicativos Mobile - Nativo ou híbrido



Aplicativos Mobile - Frameworks



Aplicativos Mobile - Frameworks

No React Native, a perda de desempenho ocorre ao acessar os recursos nativos do smartphone devido à utilização da Bridge que realiza a conexão com os recursos nativos do smartphone. Dessa forma, o React Native pode não ser a melhor escolha para aplicativos que dependem de um uso intensivo dos recursos de hardware do smartphone, impactando a performance do aplicativo.

A bridge em React Native é o mecanismo que permite que o JavaScript interaja com os componentes nativos da plataforma (iOS ou Android). Ele funciona como uma ponte bidirecional, onde o JavaScript envia comandos para o código nativo e recebe respostas de volta. Essa arquitetura é essencial porque o React Native utiliza duas camadas distintas:

1. **Camada JavaScript:** Onde o código da interface do usuário e a lógica do aplicativo são escritos.
2. **Camada Nativa:** Onde os componentes específicos da plataforma são implementados.

A comunicação entre essas camadas é assíncrona, o que significa que as operações não bloqueiam a execução principal da aplicação.

Aplicativos Mobile - Frameworks

Como funciona a Bridge

1. **JavaScript envia mensagens para o código nativo:** Quando você precisa acessar funcionalidades nativas, como o GPS ou a câmera, o JavaScript solicita isso por meio de módulos nativos.
2. **Código nativo processa as solicitações:** O código nativo executa as operações necessárias e retorna os resultados ao JavaScript.
3. **Mensagens de volta para o JavaScript:** Os resultados são enviados de volta para o lado JavaScript, que os processa para continuar a execução.

Essa arquitetura oferece flexibilidade e permite que desenvolvedores adicionam funcionalidades nativas personalizadas ao aplicativo.

A seguir, serão apresentados os passos para configuração do ambiente de

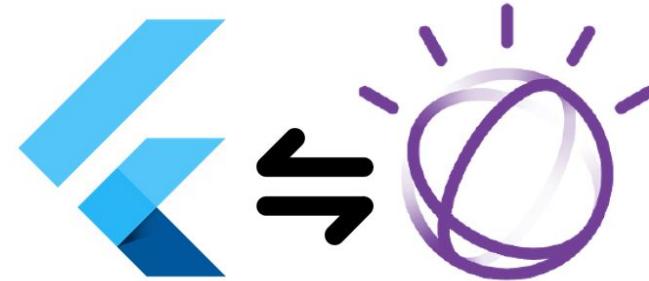
Aplicativos Mobile - Flutter

- Flutter é um kit de desenvolvimento de interface de usuário ou (UI toolkit - Um Kit de ferramentas de interface do usuário, de código aberto criado pelo Google em 2017
- Comunidade em franca expansão tanto no Brasil quanto no exterior.
- Linguagem de programação utilizada Dart
- Documentação disponível <https://docs.flutter.dev/reference/tutorials>

Aplicativos Mobile - Flutter

- Multiplataforma - Android, IOS, Linux e Mac OS
- Criação de aplicações nativas a partir de um único código base.
- Com o Flutter é possível desenvolver aplicações nativas para Android e IOS.
- Acesso direto aos recursos nativos do dispositivo como câmera, Wifi, memória e etc
- Maior desempenho - As aplicações criadas com Flutter possuem maior desempenho quando comparadas ao React Native, por exemplo, pois todo seu código fonte é transformado em código nativo.

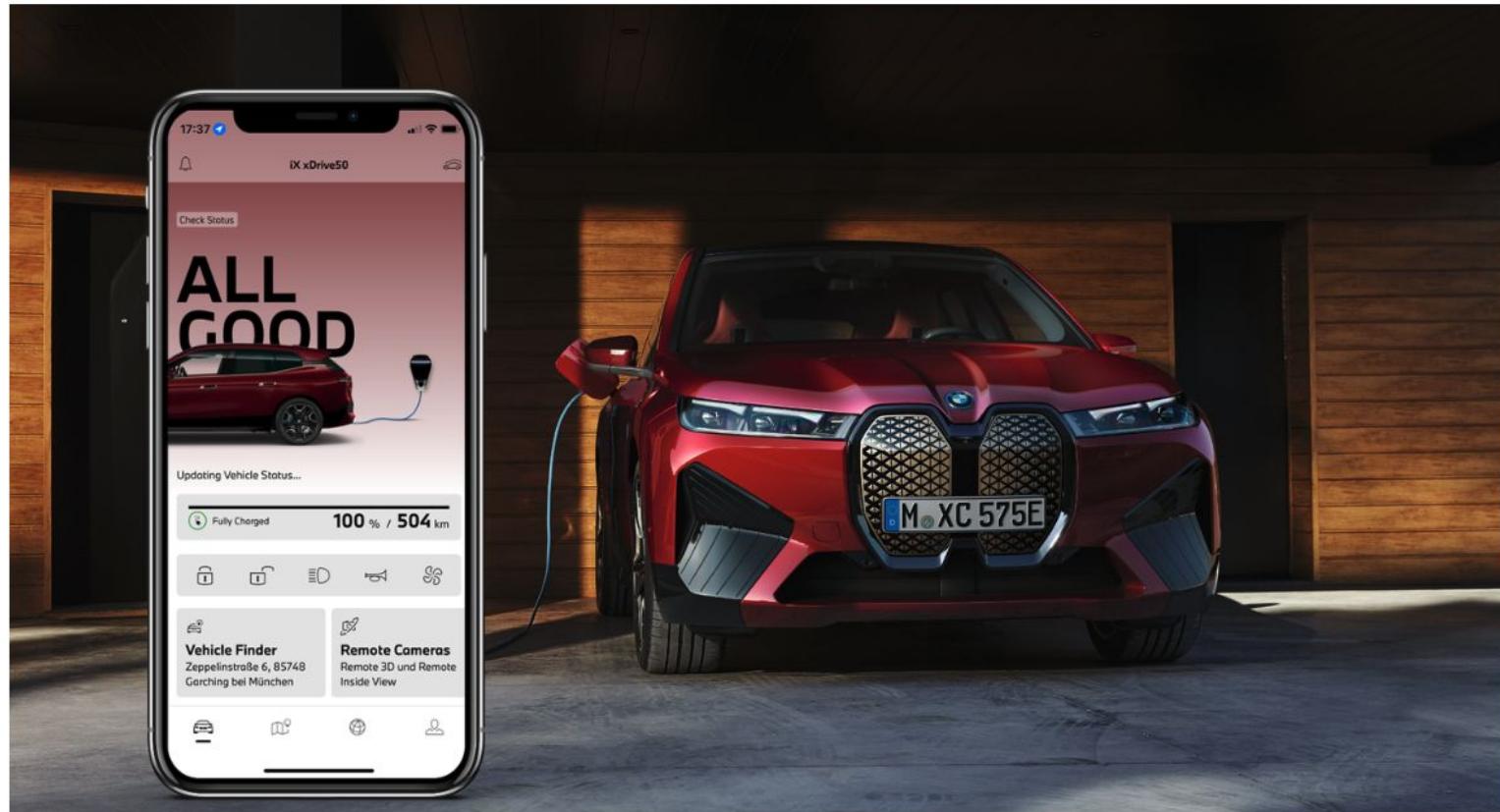
Aplicativos Mobile - Flutter - Empresas que utilizam



watson_assistant_v2
package



Aplicativos Mobile - Flutter - Empresas que utilizam



Aplicativos Mobile - Flutter - Vantagens

É gratuito, livre e aberto

Possui a funcionalidade stateful hot reload. Ela permite visualizar em tempo real as mudanças feitas na programação do aplicativo. Ou seja, o app é emulado enquanto se programa.

É capaz de executar animações em até 120 fps (frames por segundo)

Desempenho similar a de um app nativo

Grandes comunidades de desenvolvedores usam

Crescimento notório no Brasil.

Aplicativos Mobile - Flutter - Desenvolvimento

The screenshot shows a Flutter development setup. On the left, the `main.dart` file is open in a code editor, displaying the initial code for a Counter App. On the right, a mobile phone icon represents the emulator, showing the running application. The app has a red header bar with the text "Primeiro aplicativo em Flutter". The main screen displays a counter value of 7, with three buttons below it: "Incrementar" (green), "Reset" (blue), and "Decrementar" (red).

```
main.dart x
projeto-flutter > projeto3 > lib > main.dart > MyApp > build
  Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Counter App',
      theme: ThemeData(
        appBarTheme: AppBarTheme(
          backgroundColor: Colors.red,
        ), // AppBarTheme
      ), // ThemeData
      home: CounterScreen(),
    );
  }
}
```

Aplicativos Mobile - Flutter - Recursos necessários

Git <https://git-scm.com/download/win>

SDK Flutter <https://docs.flutter.dev/get-started/install>

Android Studio SDK + Emulador

https://developer.android.com/studio?gclid=Cj0KCQjw2eilBhCCARIsAG0Pf8vTiRnMeJg5uKGukaJuvs-Y54bJas-86pWq6tzA8zHcevK57S8Mx0aAI7eEALw_wcB&qclsrc=aw.ds

VSCode <https://code.visualstudio.com/download>

Extensões Flutter

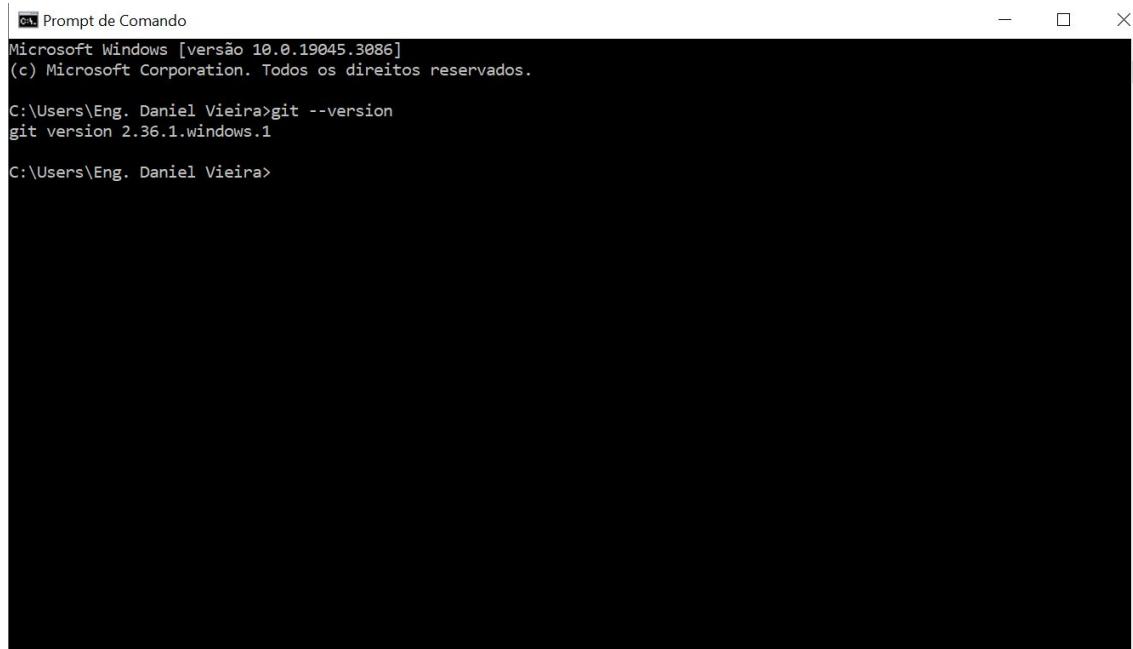
Git instalação

1º Passo : Verificar se o Git está instalado no computador

Menu Iniciar -> Digite cmd

Vai abrir prompt de comando

Digitar o comando git --version



```
Prompt de Comando
Microsoft Windows [versão 10.0.19045.3086]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Eng. Daniel Vieira>git --version
git version 2.36.1.windows.1

C:\Users\Eng. Daniel Vieira>
```

Git instalação

1º Passo : Verificar se o Git está instalado no computador

2º Passo entrar no site Git -> Fazer o download e executar a instalação

The screenshot shows the official Git website (git-scm.com/). The main navigation menu includes 'About', 'Documentation', 'Downloads' (which is highlighted in red), and 'Community'. A sidebar on the left provides information about the 'Pro Git book' by Scott Chacon and Ben Straub, available online for free. The central content area is titled 'Download for Windows' and features a prominent call-to-action: 'Click here to download the latest (2.41.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 8 days ago, on 2023-07-13.' Below this, there are links for 'Other Git for Windows downloads', including 'Standalone Installer', '32-bit Git for Windows Setup.', '64-bit Git for Windows Setup.', 'Portable ("thumbdrive edition")', '32-bit Git for Windows Portable.', and '64-bit Git for Windows Portable.'. Further down, instructions for using the 'winget tool' are provided, along with a command-line snippet: `winget install --id Git.Git -e --source winget`. A note states that the current source code release is version 2.41.0, with a link to the source code. At the bottom, a section titled 'Now What?' encourages users to start using Git.

git --fast-version-control

About Documentation Downloads Community

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Download for Windows

[Click here to download](#) the latest (2.41.0) 64-bit version of **Git for Windows**. This is the most recent [maintained build](#). It was released **8 days ago**, on 2023-07-13.

Other Git for Windows downloads

[Standalone Installer](#)
[32-bit Git for Windows Setup.](#)
[64-bit Git for Windows Setup.](#)
[Portable \("thumbdrive edition"\)](#)
[32-bit Git for Windows Portable.](#)
[64-bit Git for Windows Portable.](#)

Using winget tool
Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version 2.41.0. If you want the newer version, you can build it from [the source code](#).

Now What?

Now that you have downloaded Git, it's time to start using it.

Instalação do Flutter

1º Passo : Entrar no site flutter e selecionar de acordo com o sistema operacional: Windows, Linux, Mac

The screenshot shows the Flutter website's 'Install' page. At the top, there is a navigation bar with links for Multi-Platform, Development, Ecosystem, Showcase, Docs, and a search bar. To the right of the search bar is a 'Get started' button. Below the navigation bar, a blue banner says 'Read the announcement!' and has a 'Set up an editor' link with icons for GitHub and VS Code.

The main content area is titled 'Install' and shows the path 'Get started > Install'. It asks the user to 'Select the operating system on which you are installing Flutter:' and provides four options: Windows, macOS, Linux, and ChromeOS, each with its respective logo. A yellow callout box at the bottom left contains the text: 'Important: If you're in China, read [Using Flutter in China](#)'. To the right of this box is another 'Set up an editor' link. On the left side of the page, there is a sidebar with a 'Get started' section containing numbered steps (1. Install, 2. Set up an editor, etc.) and a 'From another platform?' section with a 'Dart language overview' link. Other sections listed in the sidebar include Stay up to date, Samples & tutorials, User interface, Navigation & routing, Data & backend, Accessibility & localization, Platform integration, and Packages & plugins.

Instalação do Flutter

2º Passo : Baixar a versão do Flutter

The screenshot shows the official Flutter website's 'Get started' page for Windows. The left sidebar has sections like 'Get started' (with '1. Install' selected), 'Stay up to date', 'Samples & tutorials', and various developer tools. The main content area discusses minimum requirements (Windows 10, 1.64 GB disk space, Git 2.x or PowerShell 5.0+), provides a link to the 'Flutter SDK' (highlighted in blue), and includes important notes for users in China and a warning about installation paths. The right sidebar contains links for 'Contents', 'System requirements', 'Get the Flutter SDK', 'Update your path', 'Run flutter doctor', 'Android setup', 'Install Android Studio', 'Set up your Android device', 'Set up the Android emulator', 'Agree to Android Licenses', 'Windows setup', 'Additional Windows requirements', and 'Next step'.

To install and run Flutter, your development environment must meet these minimum requirements:

- **Operating Systems:** Windows 10 or later (64-bit), x86-64 based.
- **Disk Space:** 1.64 GB (does not include disk space for IDE/tools).
- **Tools:** Flutter depends on these tools being available in your environment.
 - [Windows PowerShell 5.0](#) or newer (this is pre-installed with Windows 10)
 - [Git for Windows 2.x](#), with the [Use Git from the Windows Command Prompt](#) option.

If Git for Windows is already installed, make sure you can run `git` commands from the command prompt or PowerShell.

Get the Flutter SDK

Important: If you're in China, read [Using Flutter in China](#).

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

[flutter_windows_3.10.6-stable.zip](#)

For other release channels, and older builds, check out the [SDK archive](#).

2. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\src\flutter`).

Warning: Do not install Flutter to a path that contains special characters or spaces.

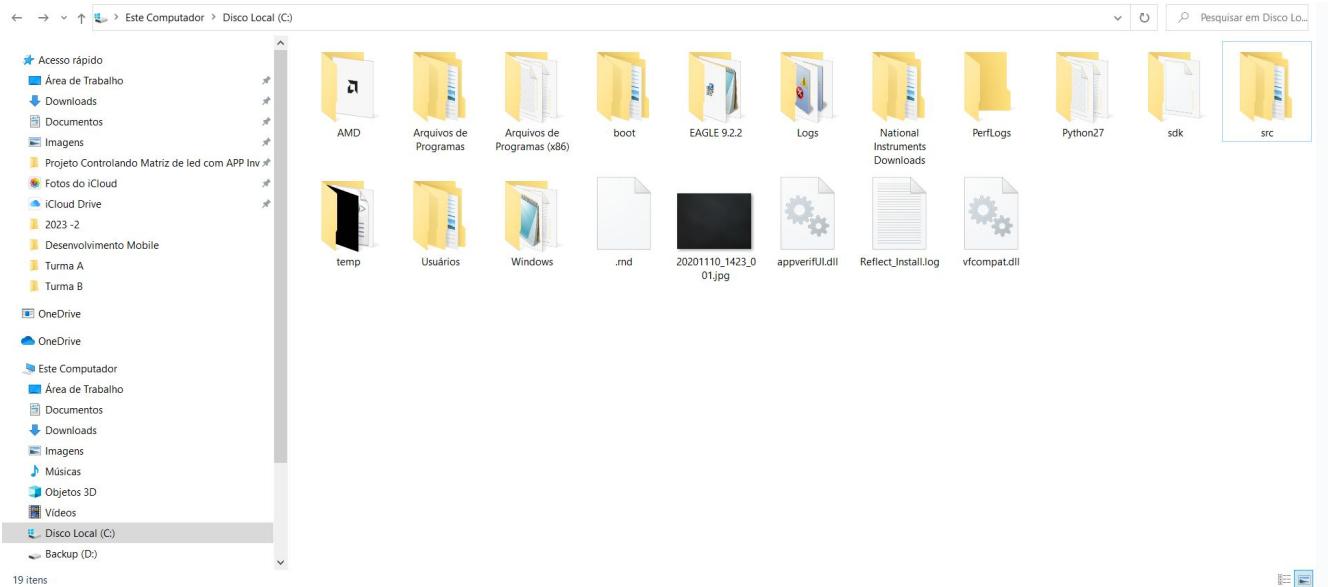
Instalação do Flutter

3º Passo : Após fazer o download, descompactá-lo.

4º Passo Criar uma pasta no disco C chamada src.

Não colocar a pasta flutter no diretório raiz C, pois pedirá permissão de administrador.

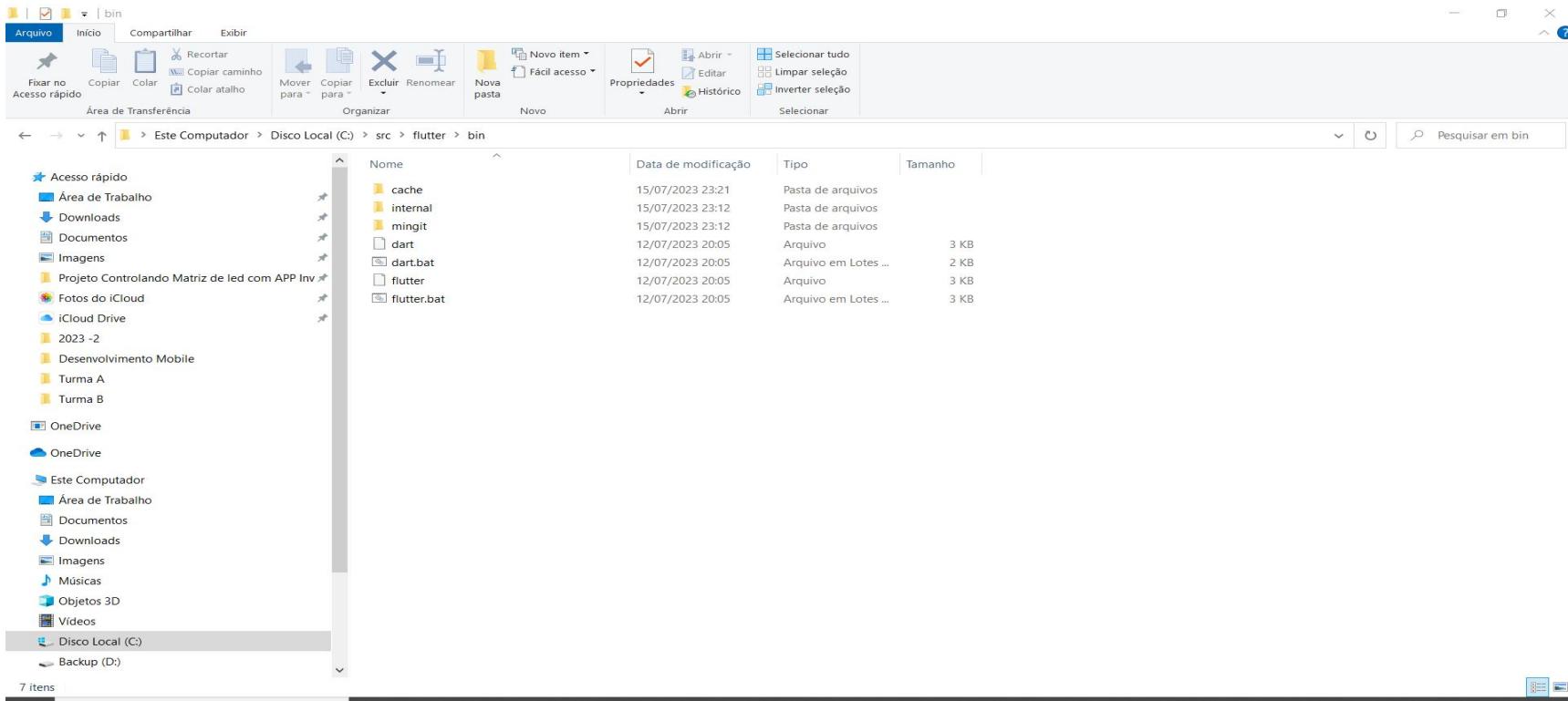
5º Passo: Copiar a pasta flutter para a pasta src



Instalação do Flutter

6º Passo : Entrar na pasta src ->flutter->bin e copiar esse caminho, pois iremos configurar as variáveis de ambiente

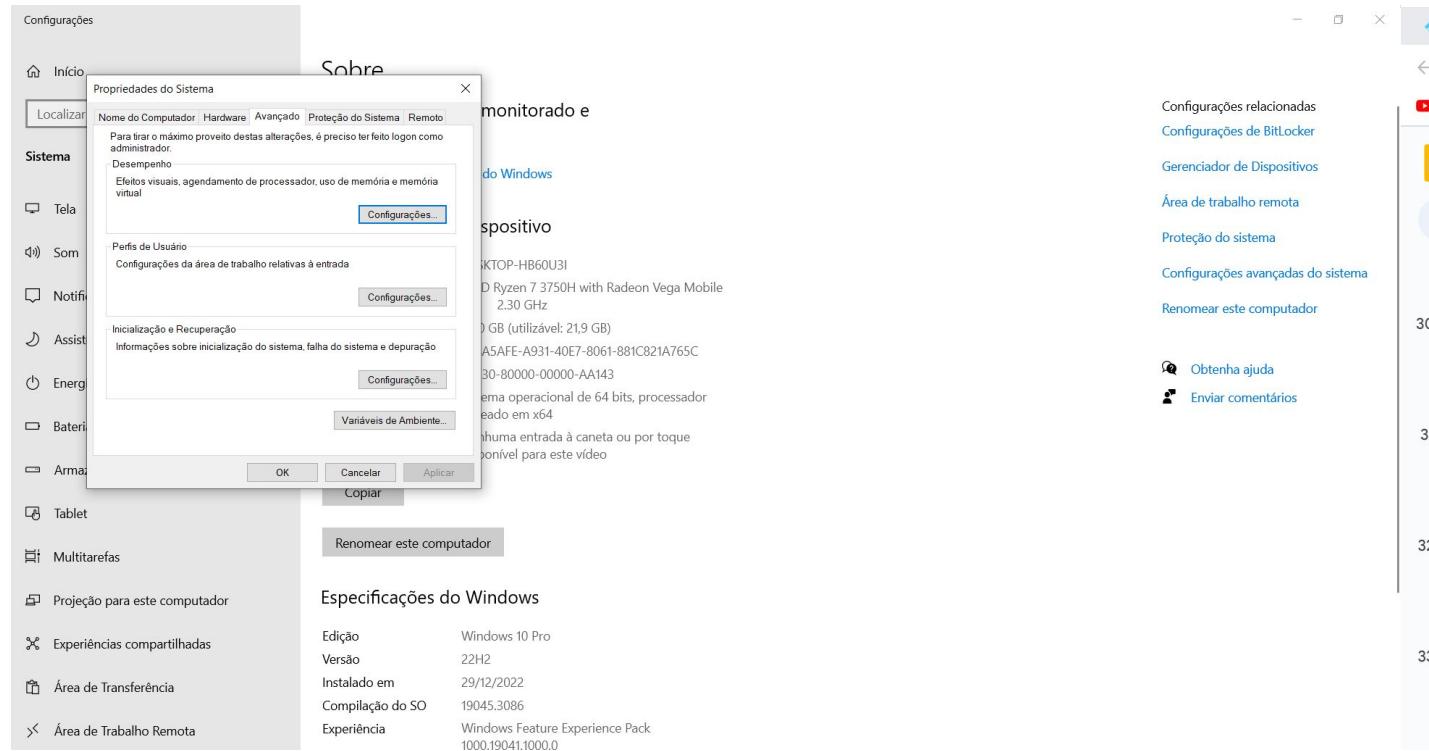
C:\src\flutter\bin



Instalação do Flutter

7º Passo : Configurar variáveis de ambiente

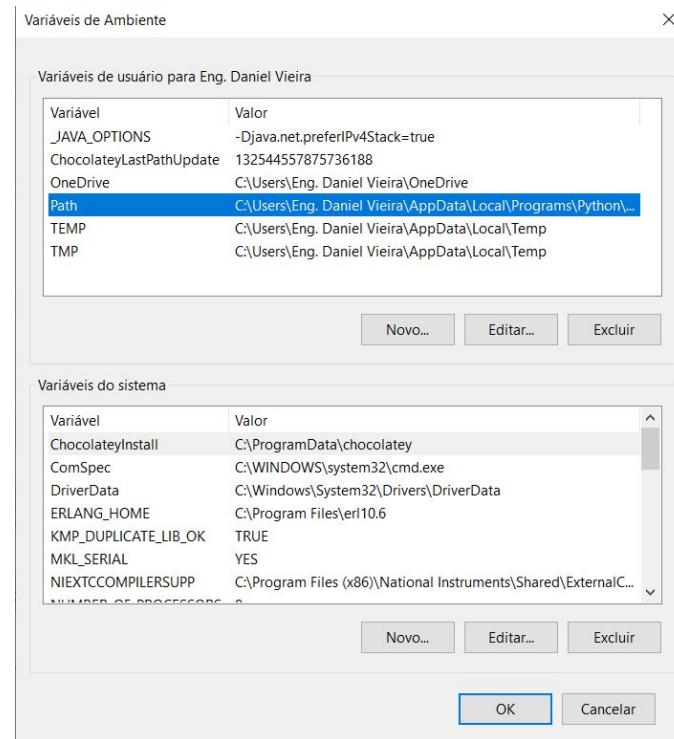
Meu Computador -> Propriedades -> Configurações avançadas do sistema



Instalação do Flutter

8º Passo : Configurar variáveis de ambiente

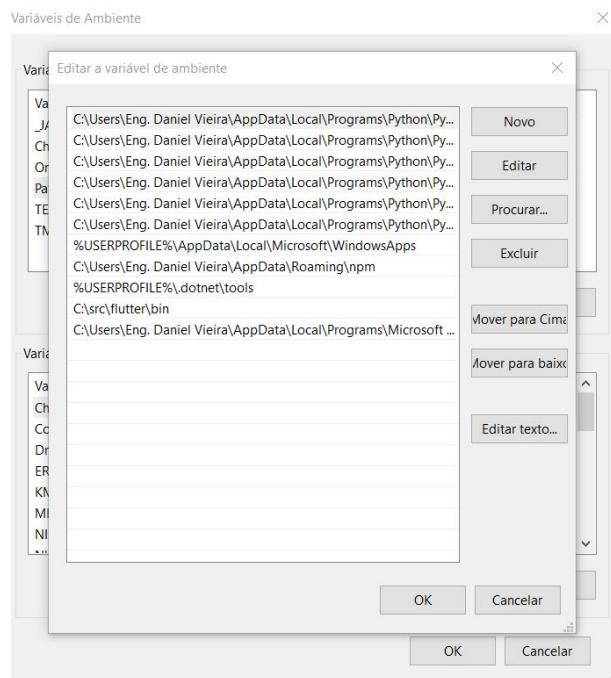
Clicar em variáveis de ambiente -> Procurar pelo campo Path e clicar em editar



Instalação do Flutter

9º Passo : Configurar variáveis de ambiente

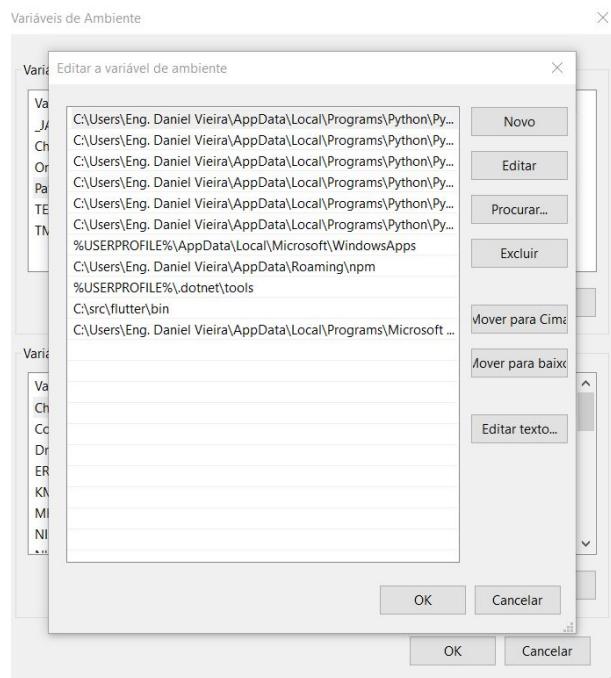
Procurar pelo campo Path e clicar em editar-> Novo e colar o caminho copiado anteriormente
C:\src\flutter\bin



Instalação do Flutter

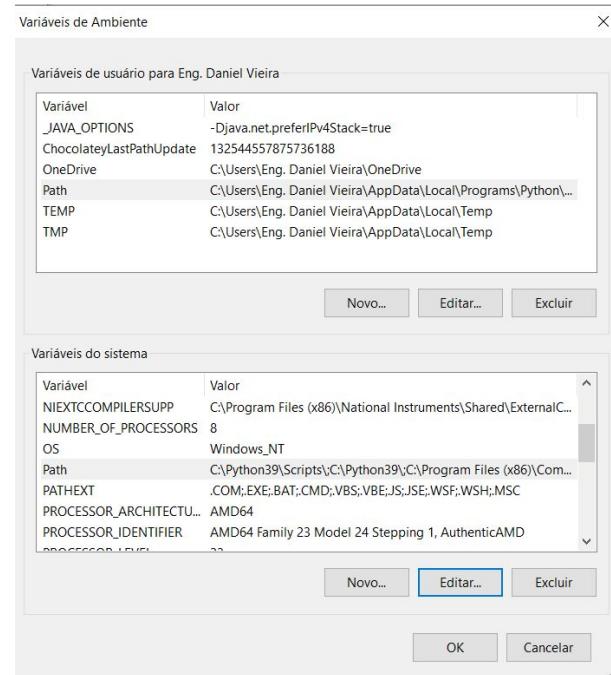
9º Passo : Configurar variáveis de ambiente

Procurar pelo campo Path e clicar em editar-> Novo e colar o caminho copiado anteriormente
C:\src\flutter\bin



Instalação do Flutter

Aqui na escola temos que colocar como variável do sistema, só do usuário dá conflito.
Procurar pelo campo Path e clicar em editar-> Novo e colar o caminho copiado anteriormente
C:\src\flutter\bin - Clicar ok e fecha as janelas

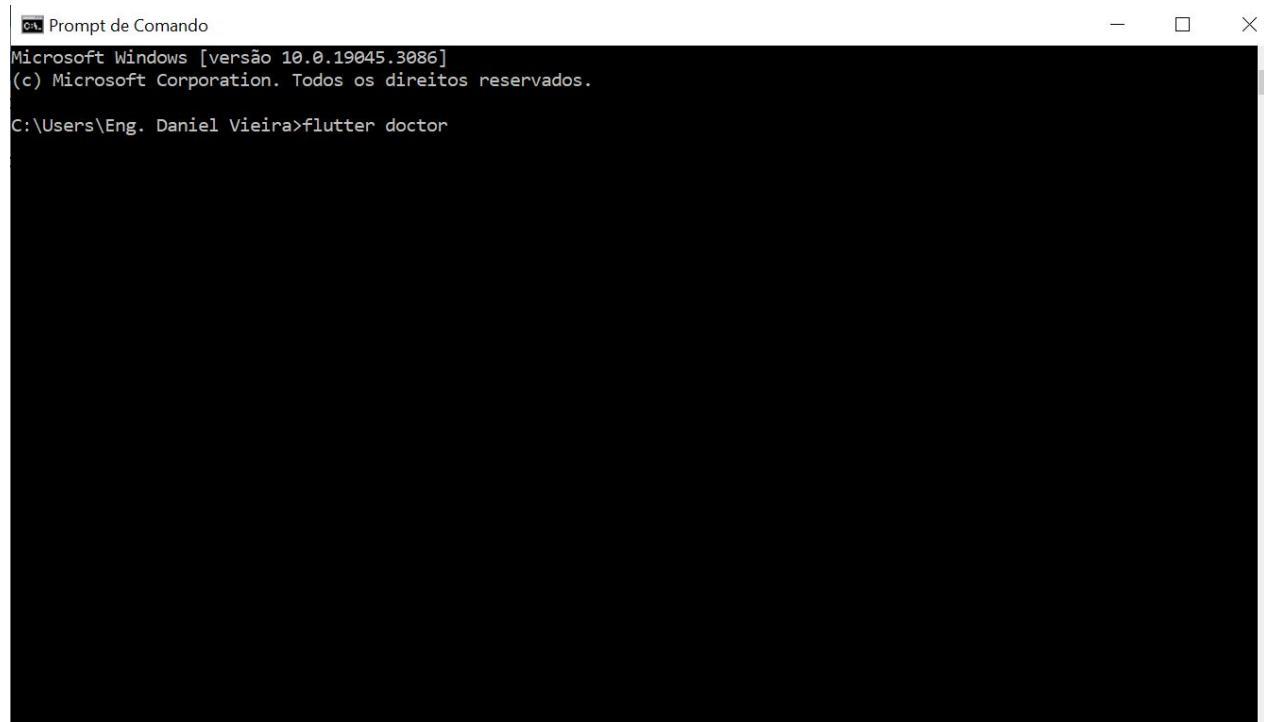


Instalação do Flutter

10º Passo

Menu iniciar -> cmd

11º Passo digitar flutter doctor para verificar os componentes instalados

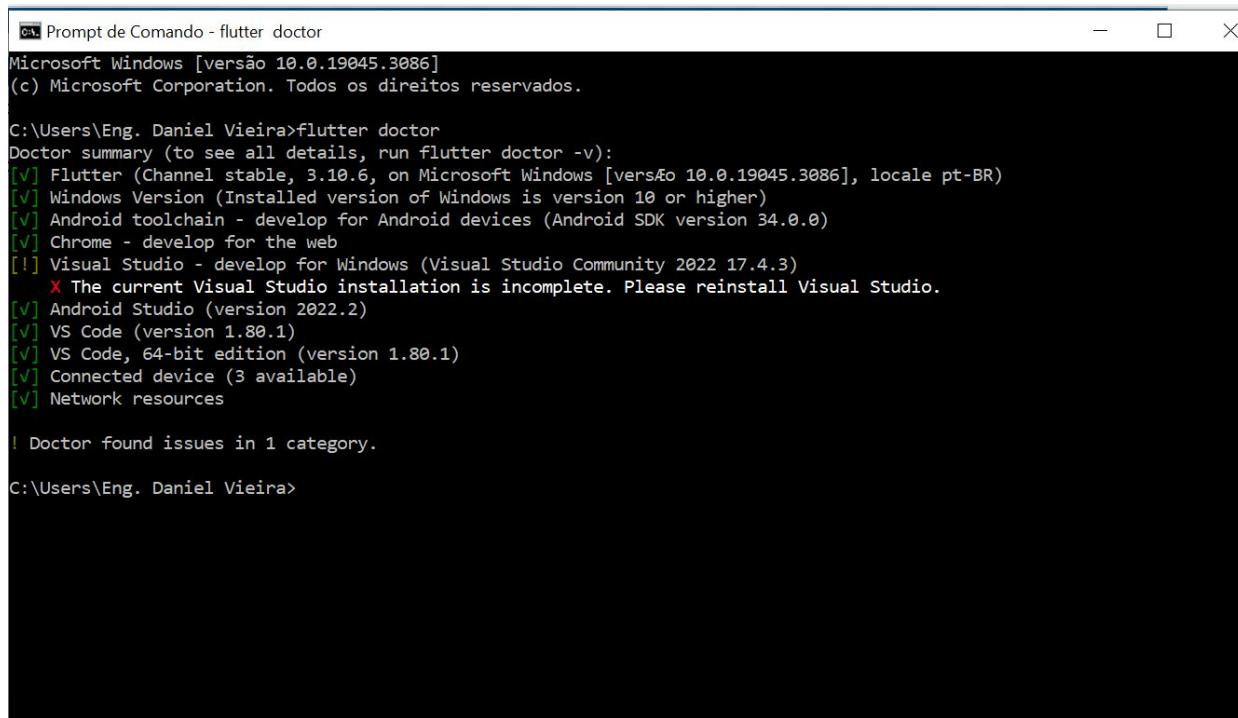


```
Prompt de Comando
Microsoft Windows [versão 10.0.19045.3086]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Eng. Daniel Vieira>flutter doctor
```

Instalação do Flutter

12º Se as configurações das variáveis de ambiente foram realizadas com sucesso, aparecerá a tela abaixo



```
Prompt de Comando - flutter doctor
Microsoft Windows [versão 10.0.19045.3086]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Eng. Daniel Vieira>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.10.6, on Microsoft Windows [versão 10.0.19045.3086], locale pt-BR)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[✓] Chrome - develop for the web
[!] Visual Studio - develop for Windows (Visual Studio Community 2022 17.4.3)
  X The current Visual Studio installation is incomplete. Please reinstall Visual Studio.
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.80.1)
[✓] VS Code, 64-bit edition (version 1.80.1)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 1 category.

C:\Users\Eng. Daniel Vieira>
```

Instalação do SDK Android

1º Fazer o download do Android Studio

The screenshot shows the official Android Studio download page and the Android Studio IDE interface.

Top Bar: developers android, Essentials, Design & Plan, Docs (highlighted), Google Play, Search, Português - ..., Android Studio, Fazer login.

Left Sidebar: ANDROID STUDIO, Download (highlighted), Android Studio editor, Android Gradle Plugin, SDK tools, Preview.

Middle Section: A large "Android Studio" logo with the text: "Get the official Integrated Development Environment (IDE) for Android app development." Below it is a green button: "Download Android Studio Flamingo" with a download icon.

Bottom Section: "Read release notes" with a link icon.

Right Side (IDE): Shows the Android Studio interface with the code editor displaying Kotlin code for a LazyGridState, the Layout Inspector showing a mobile device layout, and the App Quality Insights dashboard at the bottom.

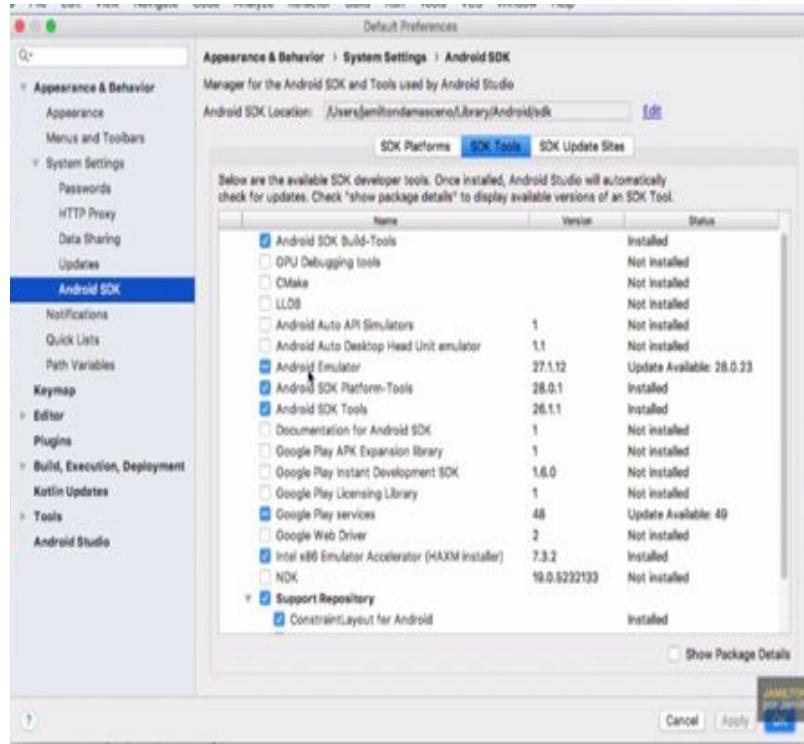
Instalação do SDK Android

1º Fazer o download do Android Studio

The screenshot shows the official Android Studio download page. At the top, there's a navigation bar with links for 'developers' (with a logo), 'Essentials', 'Design & Plan', 'Docs' (which is underlined in green), and 'Google Play'. To the right of the navigation is a search bar with the placeholder 'Search', a language dropdown set to 'Português - ...', and links for 'Android Studio' and 'Fazer login'. Below the navigation, there's a section titled 'ANDROID STUDIO' with tabs for 'Download' (which is selected and highlighted in blue), 'Android Studio editor', 'Android Gradle Plugin', 'SDK tools', and 'Preview'. The main content area features a large 'Android Studio' logo and a brief description: 'Get the official Integrated Development Environment (IDE) for Android app development.' Below this is a green button with the text 'Download Android Studio Flamingo' and a downward arrow icon. At the bottom left, there's a link 'Read release notes' with a document icon. The bottom right portion of the screenshot shows a screenshot of the Android Studio IDE interface, displaying code in the editor and a preview of an app running on a 'Google Pixel 7 Pro' device.

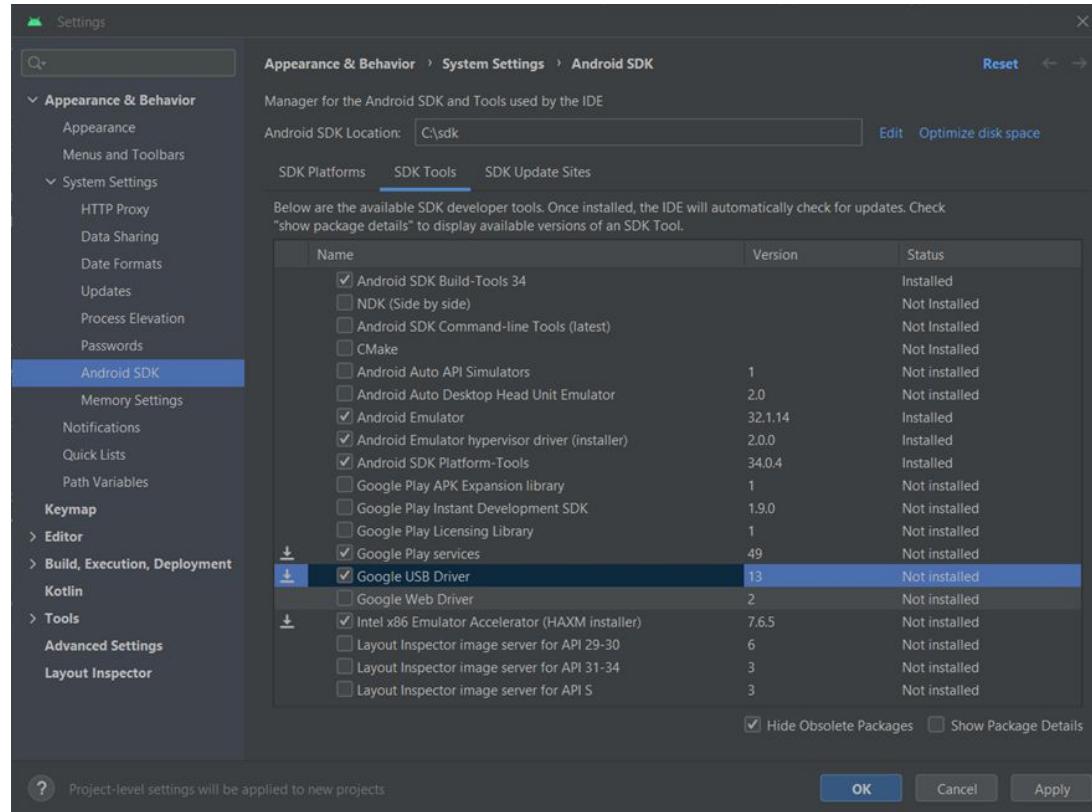
Instalação do SDK Android

2º Instalar o Android Studio



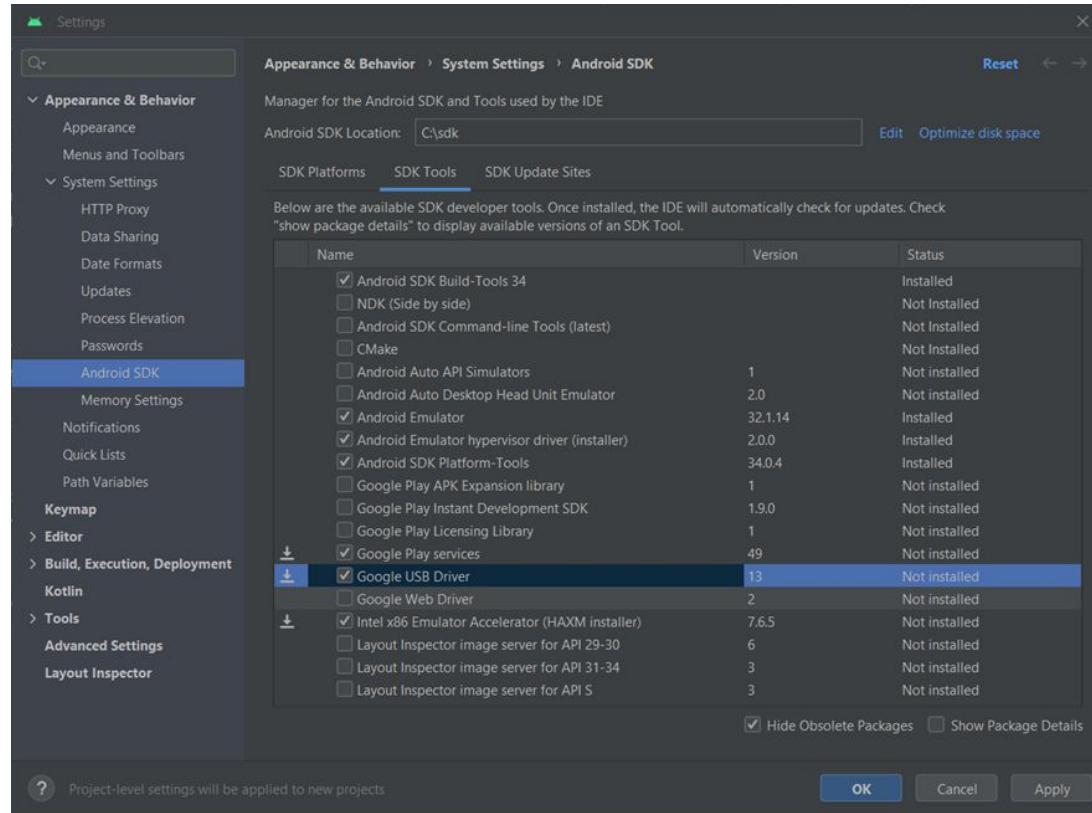
Instalação do SDK Android

2º Instalar o SDK



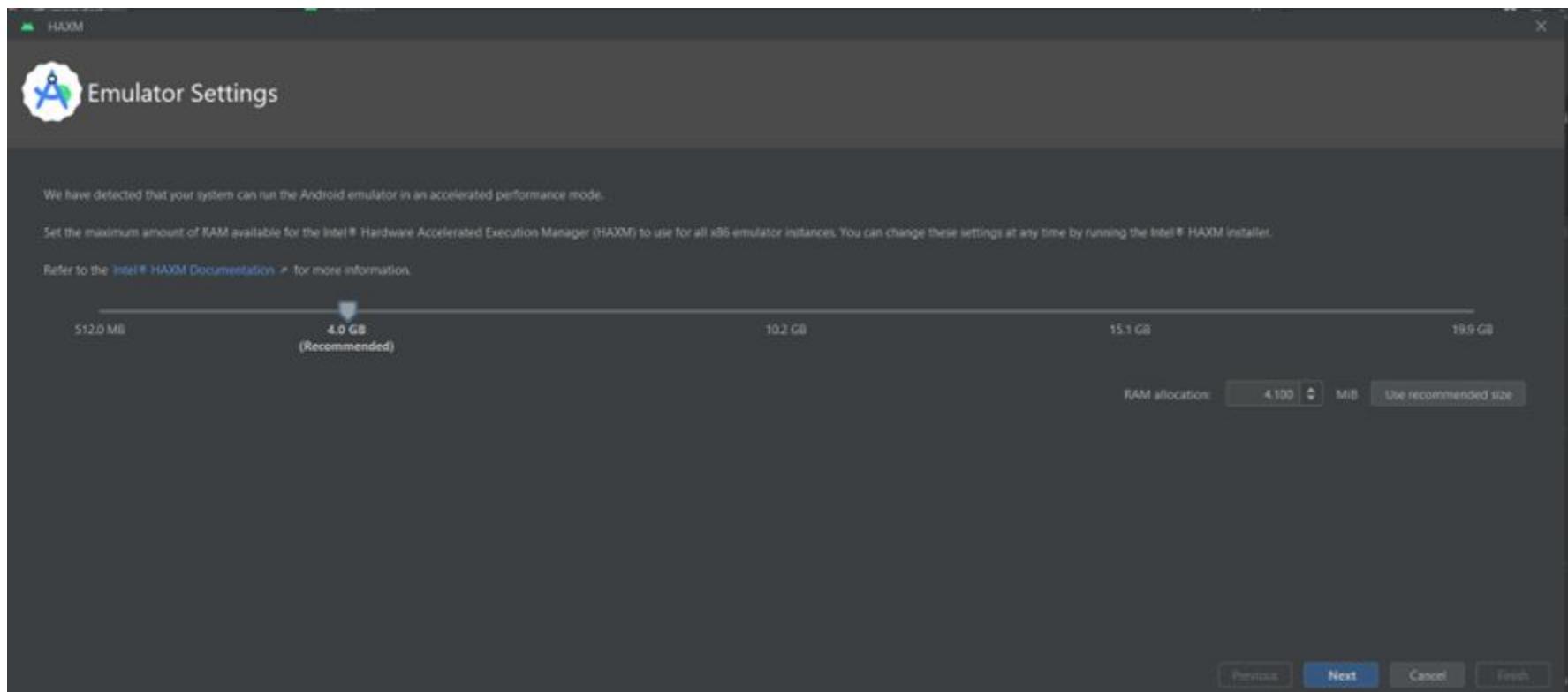
Instalação do SDK Android

2º Instalar o SDK



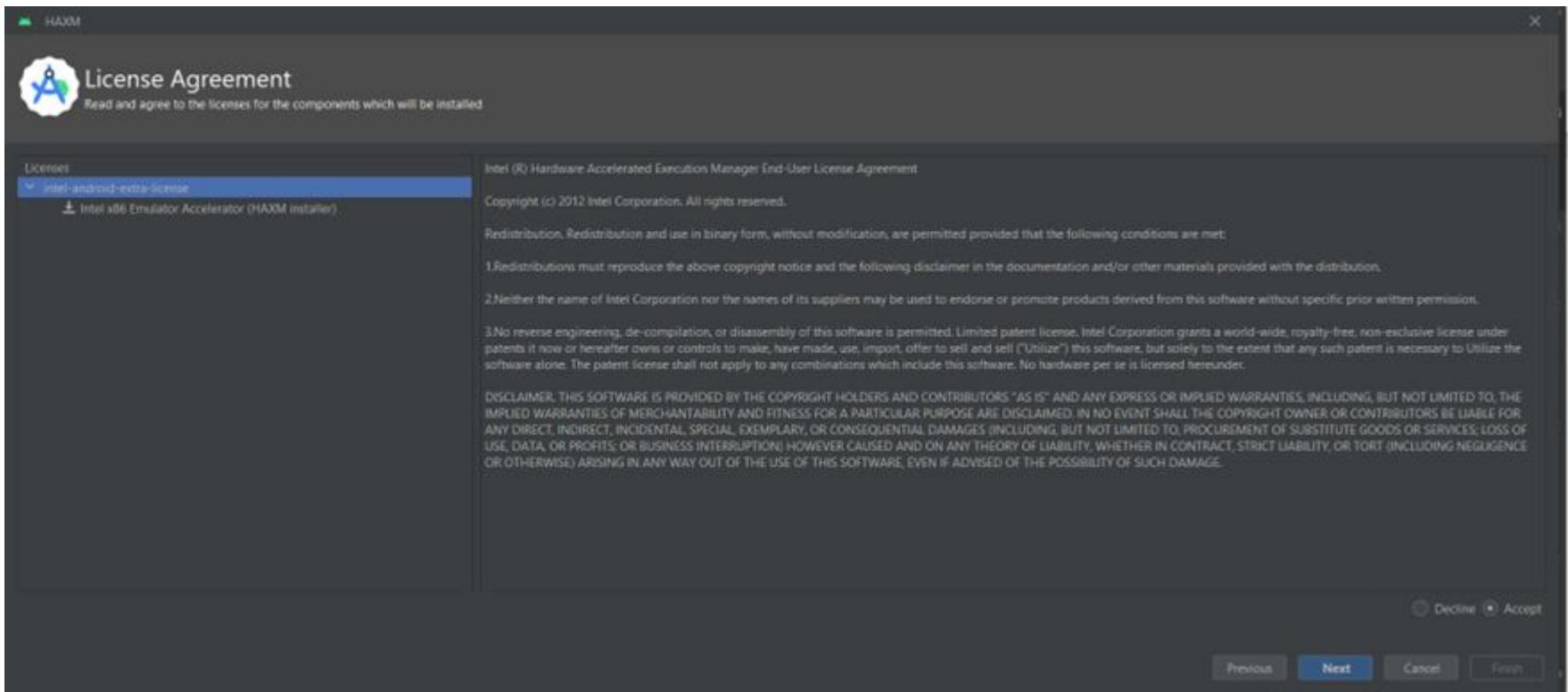
Instalação do SDK Android

3º Instalar o Emulador



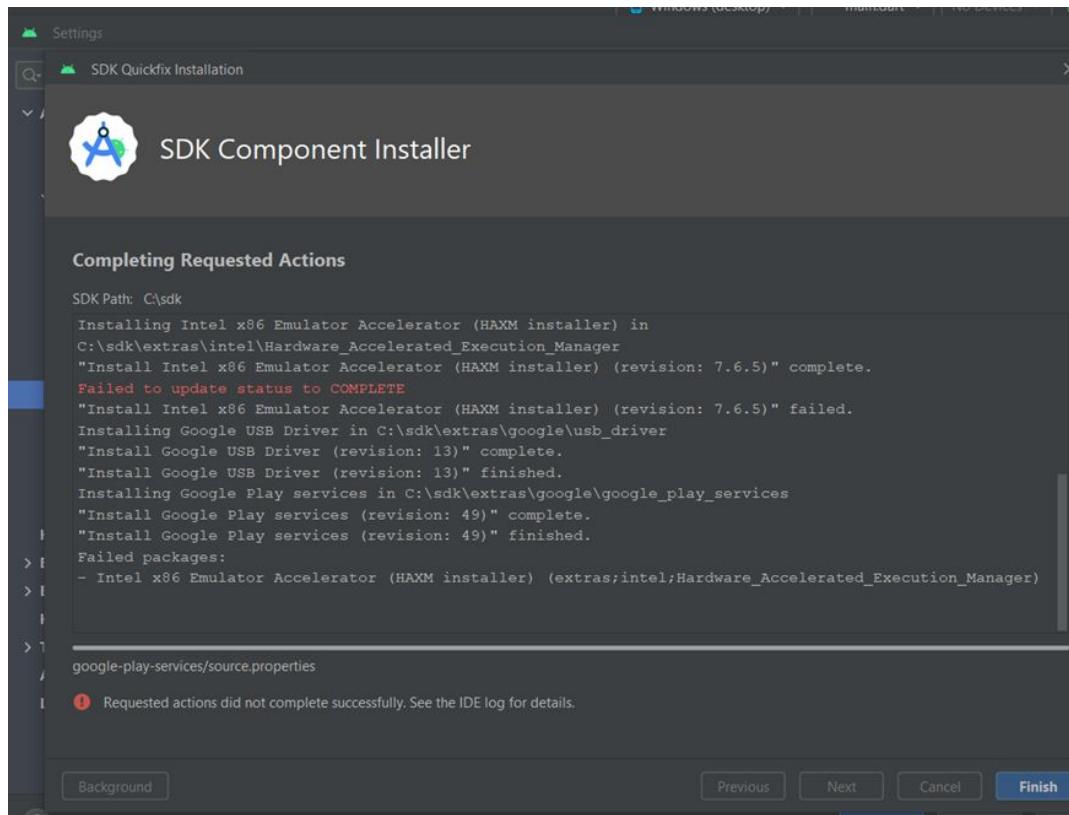
Instalação do SDK Android

4º Concordar com a licença



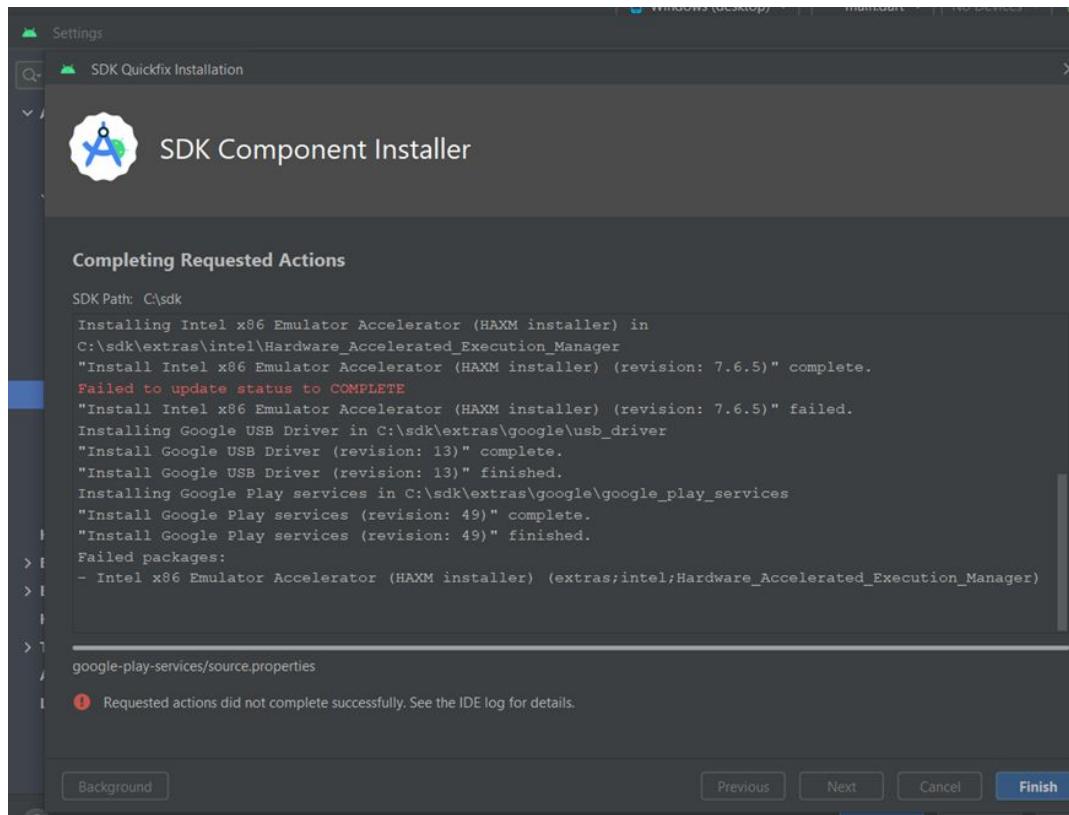
Instalação do SDK Android

5º Instalação do Emulador



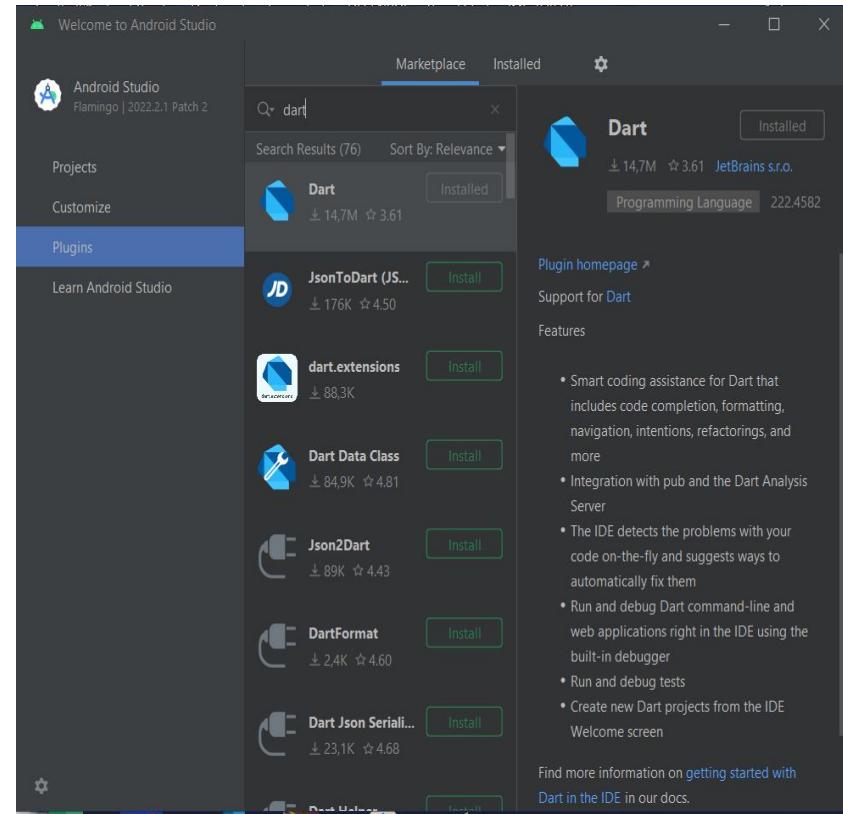
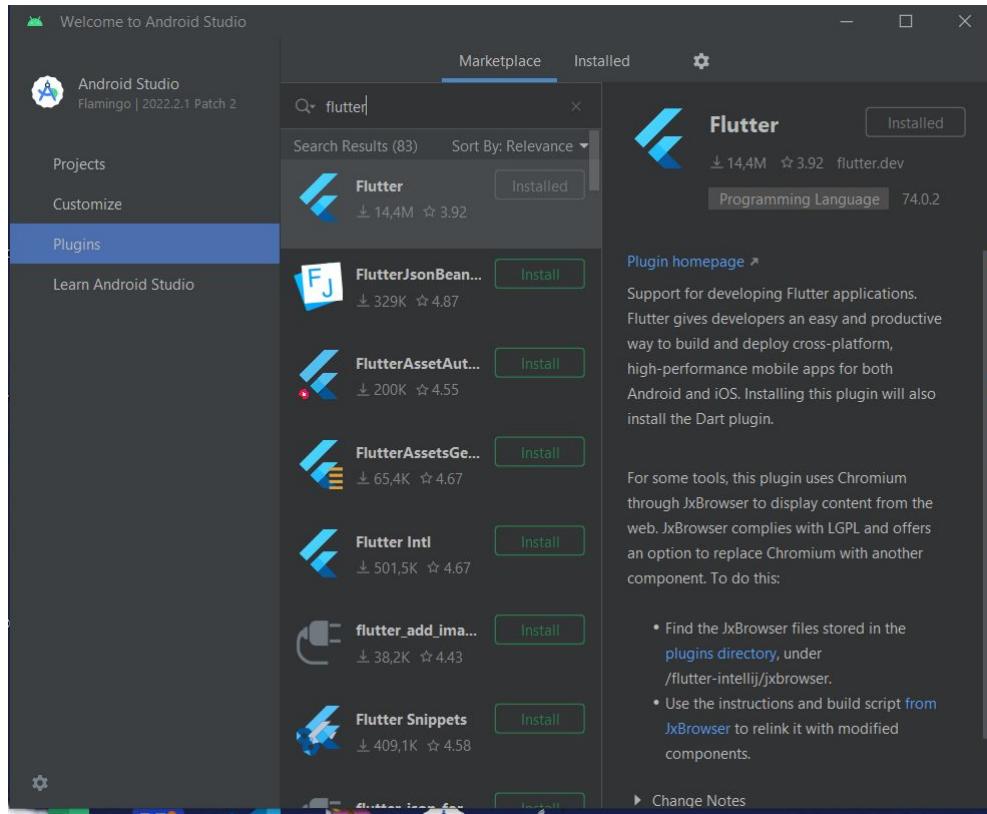
Instalação do SDK Android

5º Instalação do Emulador



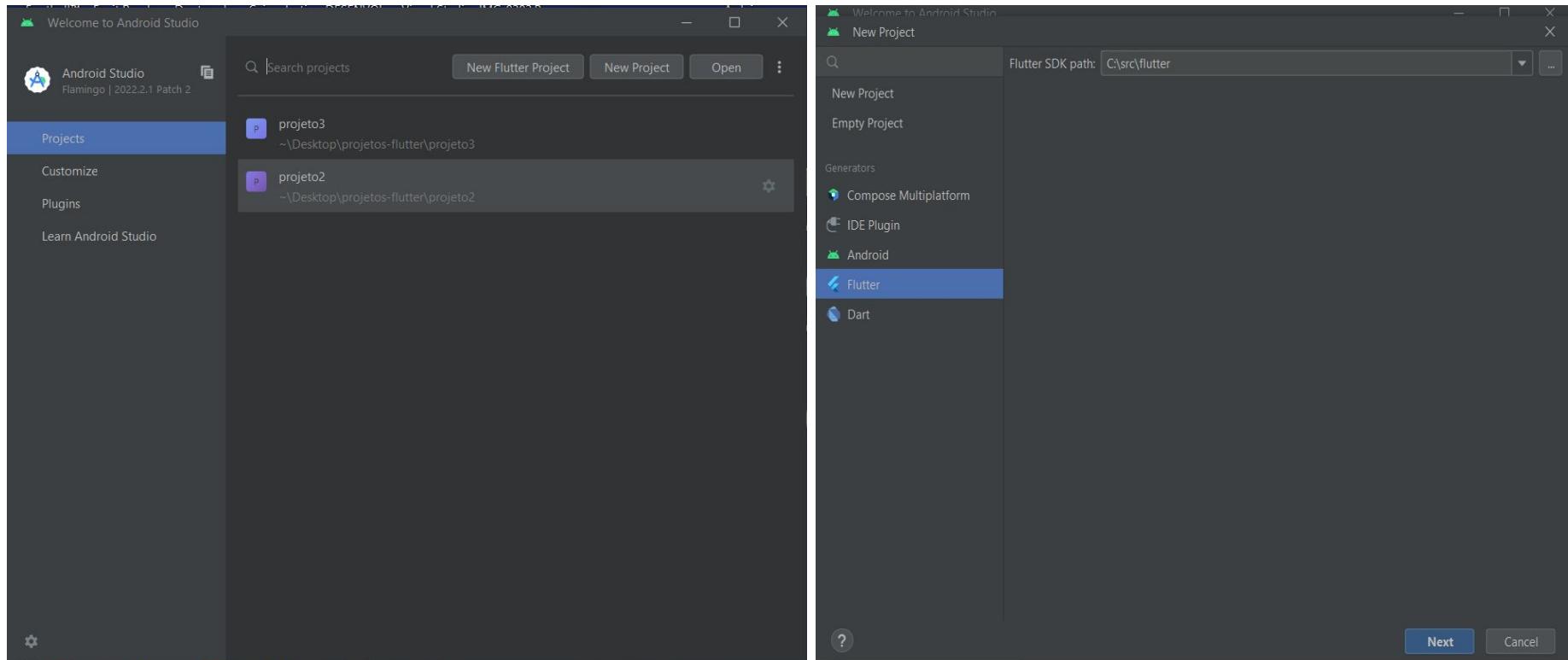
Instalação do SDK Android

6º Instalação dos plugins Flutter e Dart



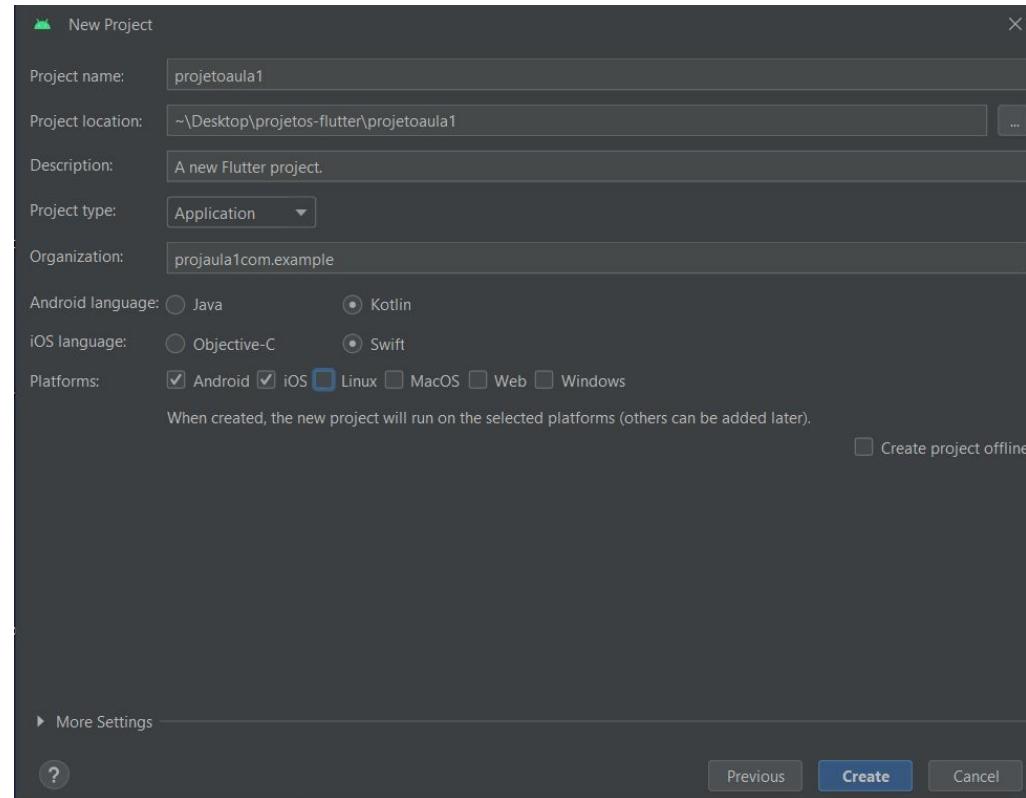
Instalação do SDK Android

7º Criando um projeto Flutter no Android Studio
Clicar em New Flutter Project



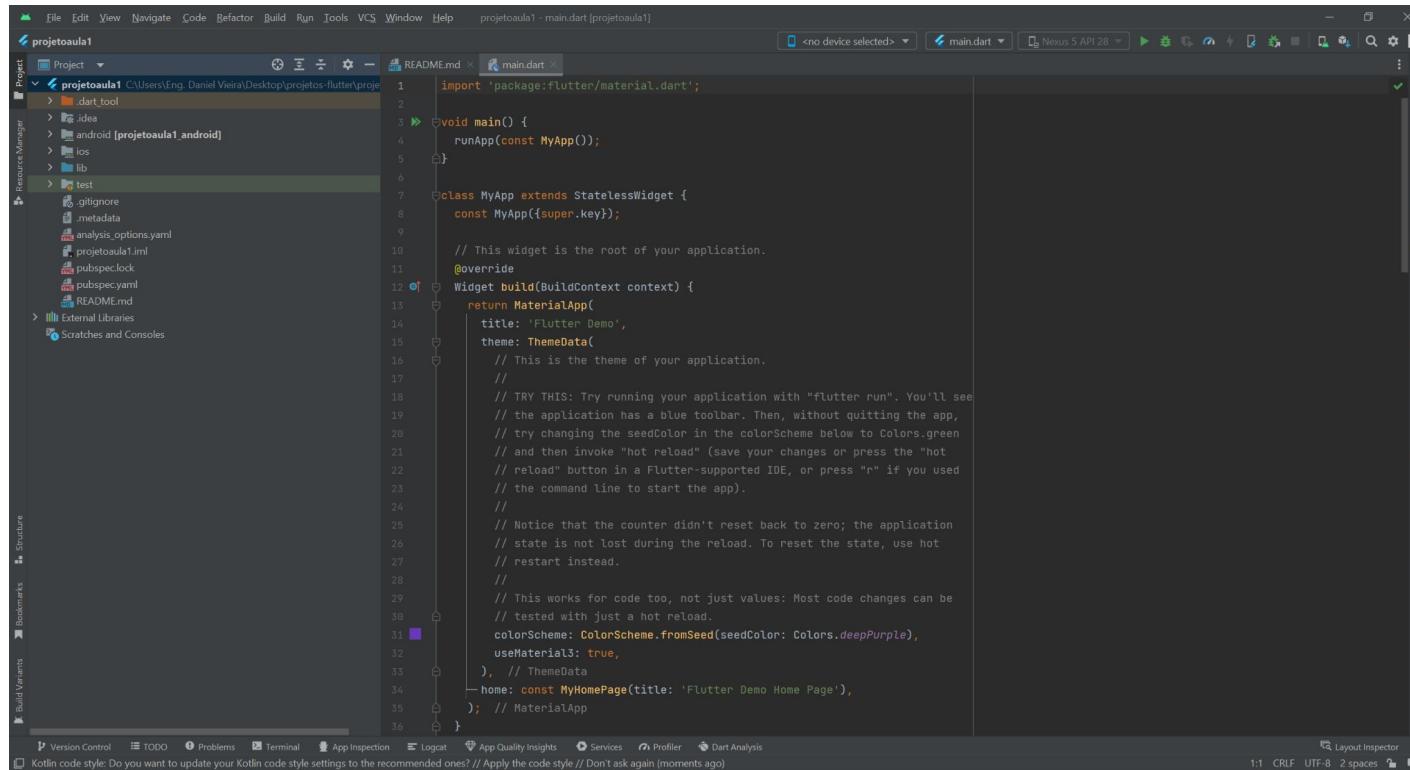
Instalação do SDK Android

7º Criando um projeto Flutter no Android Studio Clicar em New Flutter Project



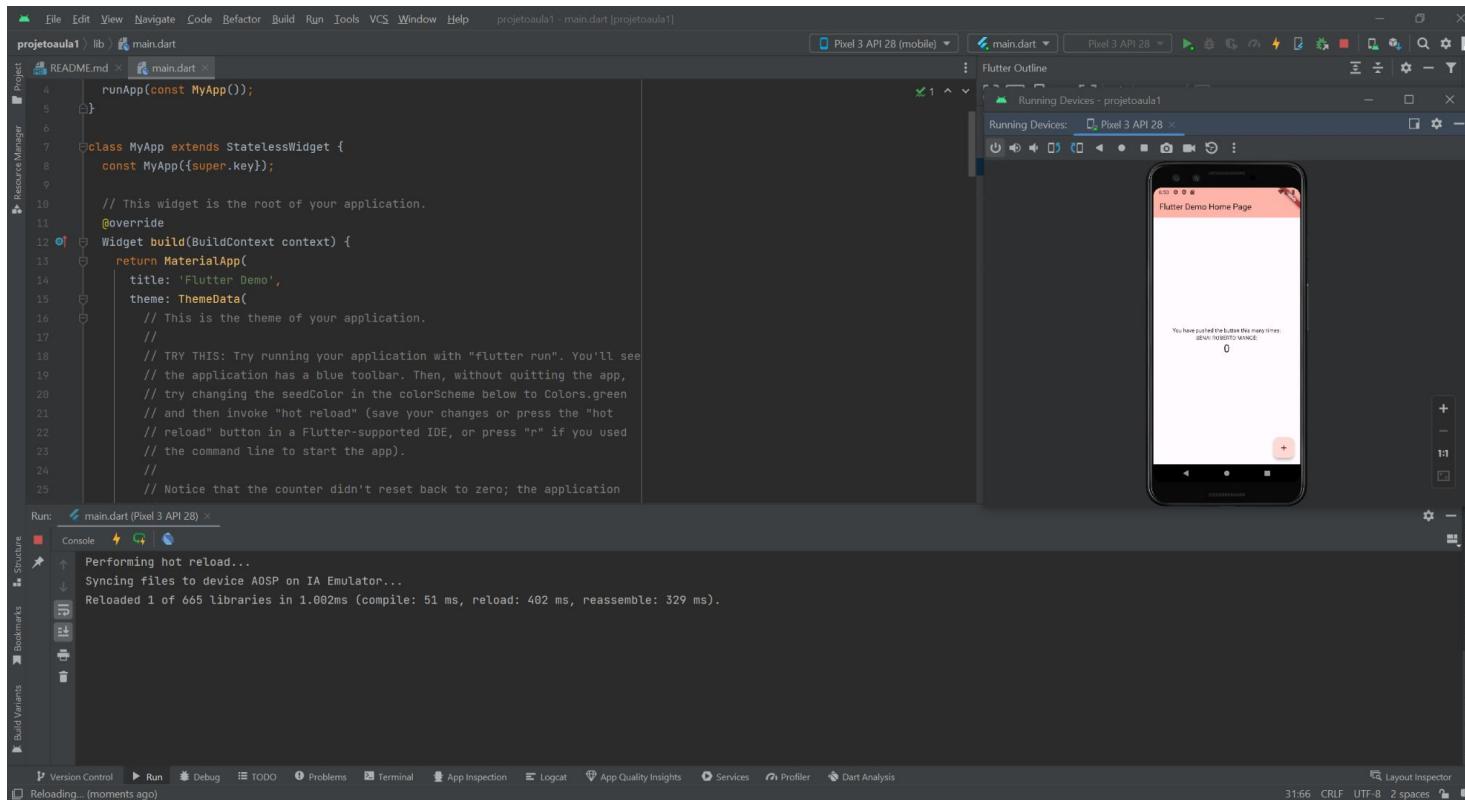
Instalação do SDK Android

7º Criando um projeto Flutter no Android Studio Clicar em New Flutter Project



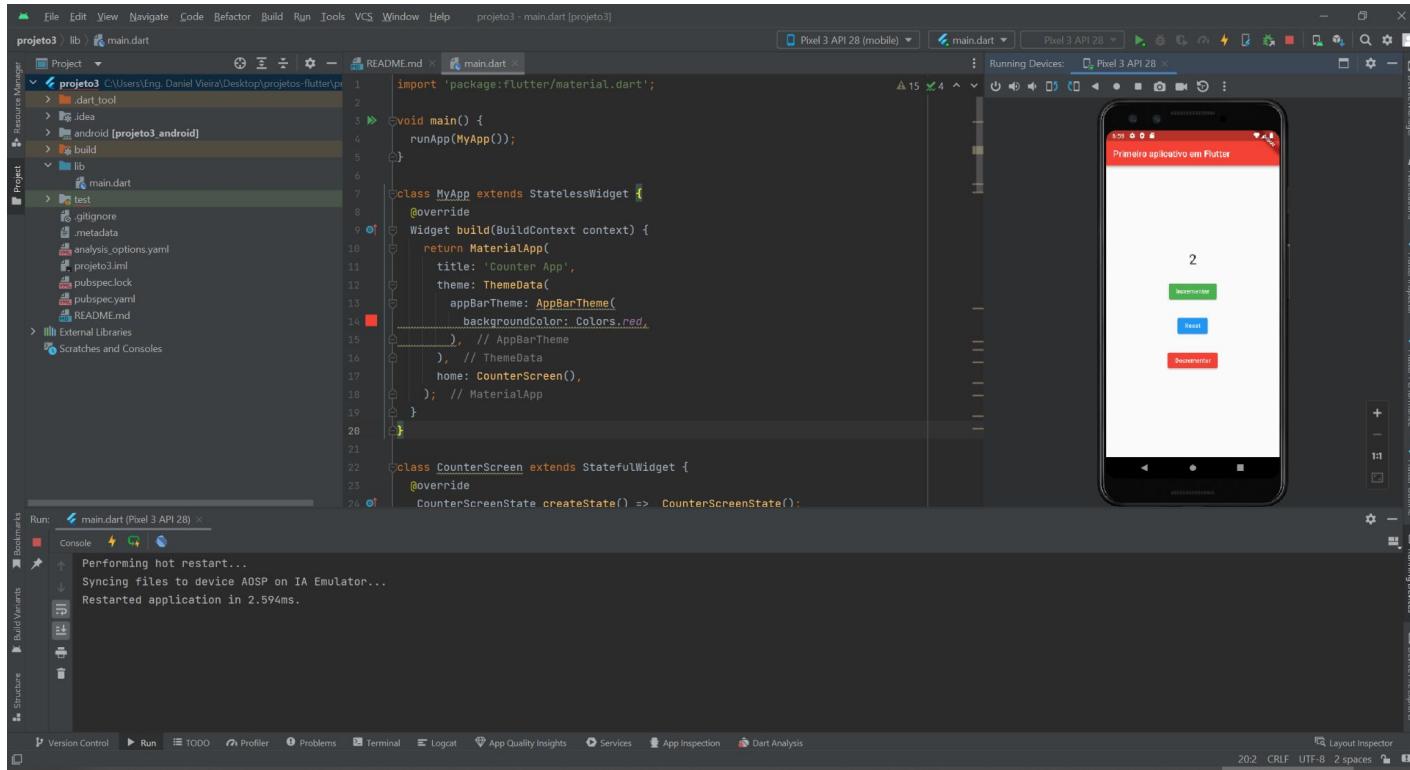
Instalação do SDK Android

7º Criando um projeto Flutter no Android Studio Clicar em New Flutter Project



Instalação do SDK Android

7º Criando um projeto Flutter no Android Studio Clicar em New Flutter Project



Código Flutter

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Counter App',
      theme: ThemeData(
        appBarTheme: AppBarTheme(
          backgroundColor: Colors.red,
        ),
        ),
      home: CounterScreen(),
    );
  }
}
```

Código Flutter

```
class CounterScreen extends StatefulWidget {  
  @override  
  _CounterScreenState createState() => _CounterScreenState();  
  
}  
  
class _CounterScreenState extends State<CounterScreen> {  
  int _counter = 0;  
  
  void _incrementCounter() {  
    setState(() {  
      _counter++;  
    });  
  }  
}
```

Código Flutter

```
void _resetCounter() {  
    setState(() {  
        _counter = 0;  
    });  
}  
void _decrementCounter() {  
    setState(() {  
        _counter = _counter -1;  
    });  
}
```

Código Flutter

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Primeiro aplicativo em Flutter'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            '_$_counter',
            style: TextStyle(fontSize: 30),
          ),
          SizedBox(height: 30),
          ElevatedButton(
            onPressed: _incrementCounter,
            child: Text('Incrementar'),
            style: ButtonStyle(
              backgroundColor: MaterialStateProperty.all<Color>(Colors.green),
            ),
          ),
        ],
      ),
    ),
  );
}
```

Código Flutter

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Primeiro aplicativo em Flutter'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            '_$_counter',
            style: TextStyle(fontSize: 30),
          ),
          SizedBox(height: 30),
          ElevatedButton(
            onPressed: _incrementCounter,
            child: Text('Incrementar'),
            style: ButtonStyle(
              backgroundColor: MaterialStateProperty.all<Color>(Colors.green),
            ),
          ),
        ],
      ),
    ),
  );
}
```

Código Flutter

```
SizedBox(height: 30),  
    ElevatedButton(  
        onPressed: _resetCounter,  
        child: Text('Reset'),  
        style: ButtonStyle(  
            backgroundColor: MaterialStateProperty.all<Color>(Colors.blue),  
        ),  
    ),
```

Código Flutter

```
SizedBox(height: 30),  
    ElevatedButton(  
        onPressed: _decrementCounter,  
        child: Text('Decrementar'),  
        style: ButtonStyle(  
            backgroundColor: MaterialStateProperty.all<Color>(Colors.red),  
        ),  
    ),  
,  
],  
,  
,  
);  
}  
}
```

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

