

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

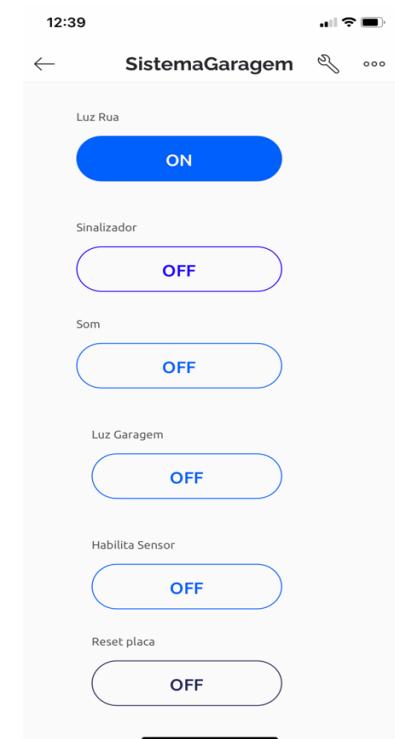
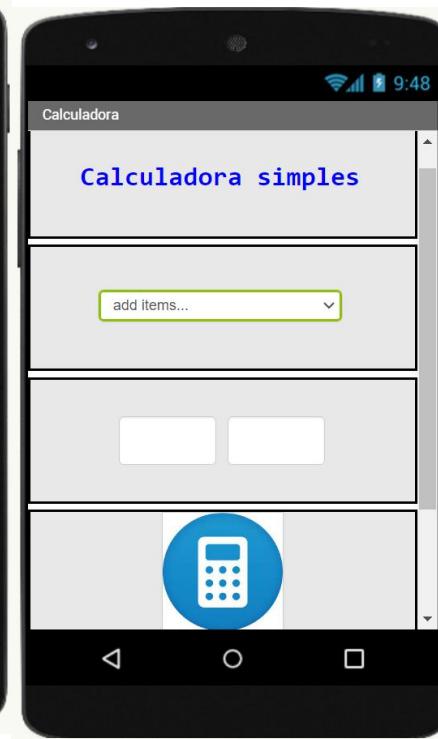
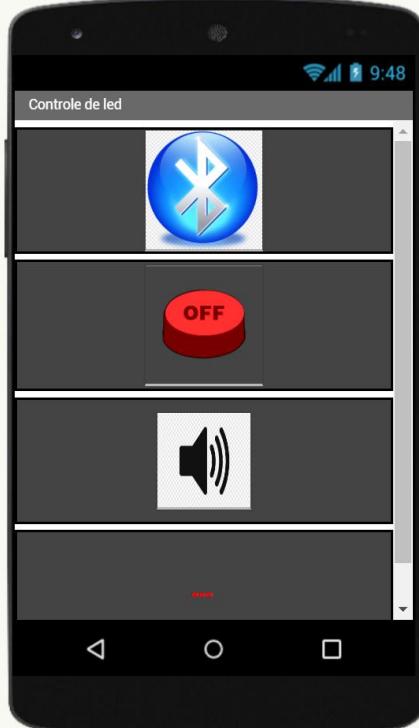
Hardware e firmware

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

Análise de dados

Pesquisa com IA, Machine Learning, Lógica Fuzzy

# Projetos realizados



# 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  
Persistência dos dados  
Banco de dados local  
Recursos Mobile  
Notificações locais  
Push Notifications

### Publicação da aplicação

Preparação e testes  
Geração da aplicação de produção (build)  
Técnica  
Manual de uso  
Hospedagem em lojas  
App Store  
Google

### Projetos

Hello world  
Escolha de combustível  
Conversor de moedas  
Calculadora  
Aplicativo para auxiliar no planejamento de tarefas  
Exercícios  
Atividade Formativa  
Atividade Somativa

# Capacidades técnicas/ Socioemocionais

1. Configurar o ambiente de desenvolvimento
2. Aplicar tecnologias Mobile de acordo com o sistema operacional
1. Elaborar programação de acordo com padrões de desenvolvimento
1. Definir os frameworks para programação Mobile em relação ao sistema operacional e aos recursos de hardware
5. Criar interfaces gráficas da aplicação baseadas em UX.
6. Teste do código com ferramentas automatizadas para garantia da qualidade do software, prevendo falhas
7. Documentar o sistema mobile para facilitar o uso e funcionamento

1. Demonstrar atenção a detalhes
2. Demonstrar capacidade de tomar decisão
3. Demonstrar organização
4. Demonstrar pensamento criativo
5. Demonstrar capacidade de valorização das diferenças sociais, culturais e sociais
6. Demonstrar capacidade de solucionar problemas

# Calendário das aulas

Mês	Julho								Agosto												Setembro							
Dia	21				28				04				11				18				25				01			
Nº aula	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
<b>CONTEÚDO FORMATIVO</b>	Apresentação da disciplina Exemplos de projetos Metodologia das aulas (Aulas expositivas e projetos Data Science e suas aplicações Configuração de ambiente VSCode + Dart + Android Studio Hello World com Flutter	Plataforma Mobile Android, IOS Ciclo de vida da aplicação Linguagem Dart Exercícios	Características do framework Flutter A Linguagem Dart e sua Sintaxe Variáveis Arrays Operadores Aritméticos Operadores relacionais e lógicos Controle de fluxo if, else. Exercícios	Controle de fluxo switch Loops – for e while Exercícios	Funções, Funções anônimas e parâmetros Exercícios	Classes Objeto Exercícios	O que são Widgets ? Elementos customizados Gerenciador de layouts Posicionamento dos elementos absoluto, relativo Exercícios																					

# Calendário das aulas

Mês	Setembro												Outubro								Novembro							
Dia	15				22				29				06				20				27				10			
Nº aula	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
<b>CONTEÚDO FORMATIVO</b>	Leiautes e Gráficos Labels Caixas de textos Botões Incremento com outros elementos Exercícios Desenvolvimento de um aplicativo para converter moedas Euro, Dólar para real	Criar aplicativo para selecionar o melhor combustível com base no valor de na distância percorrida	Princípios de UX para desenvolvimento Mobile Simplicidade, consistência, Feedback visual, Navegação intuitiva, foco no usuário. Responsividade Feedback claro de erro Performance otimizada Acessibilidade	Persistência de dados Estrutura de dados Preferência das aplicações Base de dados local Exercícios	Atividade Formativa	Mídias Image Viewer Video Player Áudio Player Desenvolvimento de um aplicativo com multi telas e botões para troca de imagens na tela	Publicação da aplicação, preparação e testes Geração da aplicação de produção(build). Documentação técnica e manual de uso. Hospedagem em lojas. Apo Stores Google																					

# Calendário das aulas

# Metodologias das aulas

Aulas teóricas, expositivas, exercícios e projetos

Aulas de quarta - Turma B e sexta - Turma A

Horário : 19:10 - 22:45

Intervalo: 21:00 - 21:15



# Critérios de avaliação

Natureza dos Critérios	Fundamentos Técnicos e Científicos ou Capacidades Técnicas	Critérios de avaliação		Alunos						O aluno foi capaz de a partir das características do projeto utilizar as bibliotecas disponíveis no Flutter para comunicação com recursos de hardware do dispositivo ?	F	S					
		Critico	Desejável														
Capacidades Técnicas	1. Configurar o ambiente de desenvolvimento para aplicações Mobile	O ambiente de desenvolvimento VSCode, Flutter, Dart foi configurado corretamente?		F						4. Definir os frameworks para programação Mobile em relação ao sistema operacional e aos recursos de hardware	O aluno foi capaz de a partir das características do projeto utilizar as bibliotecas disponíveis no Flutter para comunicação com recursos de hardware do dispositivo ?	F	S				
		O aluno foi capaz de solucionar problemas comuns na configuração do ambiente de desenvolvimento como: erros de variáveis de ambiente ?		S													
	2. Aplicar tecnologias Mobile de acordo com o sistema operacional	O aluno foi capaz de configurar o Framework Flutter para gerar o aplicativo para IOS ou Android seguindo as orientações?		F						5. Criar interfaces gráficas da aplicação Mobile baseadas em UX	O aluno foi capaz de desenvolver um aplicativo com princípios de UX para uma experiência gráfica positiva ao usuário ?	F	S				
		O aluno aplicou os princípios de UX para cria uma interface gráfica intuitiva e amigável ?		S													
	3. Elaborar programação Mobile de acordo com os padrões de desenvolvimento	O aluno foi capaz de elaborar o software legível, estruturado e limpo utilizando a linguagem Dart ?		F						6. Testar o código com ferramentas automatizadas (end to end) para garantia da qualidade do software, prevendo eventuais falhas.	O aluno foi capaz de realizar testes aplicados no aplicativo como testes unitários ? Testes de integração ?	F	S				
		O aluno foi capaz de desenvolver o código de forma organizada e modularizada?		S													
	4. Definir os frameworks para programação Mobile em relação ao sistema	O aluno foi capaz de a partir das características do projeto utilizar as bibliotecas disponíveis no Flutter para comunicação com recursos de hardware		F						7. Documentar o sistema mobile para facilitar o seu uso e funcionamento	O aluno foi capaz de criar um documento com todas as funcionalidades do aplicativo de forma clara e objetiva de modo a facilitar a refatoração do código ?	F	S				
				S													
										O aluno foi capaz de utilizar diagramas e fluxos para ilustrar visualmente o funcionamento do sistema mobile ?		F	S				

## Critérios de avaliação

1. Demonstrar atenção a detalhes	O aluno foi capaz de desenvolver o aplicativo com base nos requisitos do projeto ? (Entender, Interpretar e seguir os requisitos).	F			
	O aluno foi capaz de revisar e testar o aplicativo antes da entrega e documentou os testes e saídas do aplicativo ?	S			
2. Demonstrar capacidade de tomar decisão	O aluno foi capaz de analisar as alternativas propostas no projeto e escolher uma alternativa adequada ao budget reservado ?	F			
	O aluno foi capaz de comunicar efetivamente suas decisões de forma clara e persuasiva ?	S			
3. Demonstrar organização	O aluno foi capaz de desenvolver o aplicativo de forma estruturada e organizada com divisão de tarefas e classificação de prioridade de tarefas mais complexas e menos complexas ?	F			
	O aluno foi capaz de cumprir os prazos de entrega dos check points do aplicativo ?	S			
4. Demonstrar pensamento criativo	O aluno foi capaz de criar uma solução original e única para o projeto proposto ?	F			
	O aluno foi capaz de transformar suas ideias em um projeto viável e executável ?	S			

## Critérios de avaliação

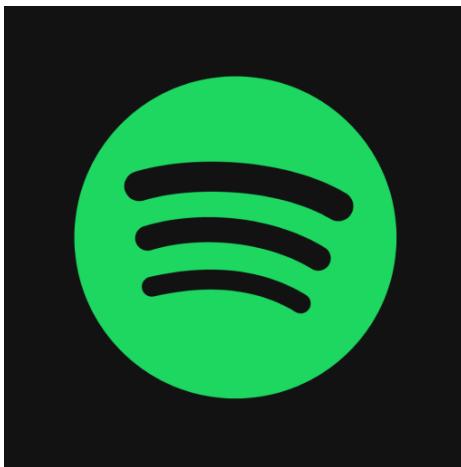
5. Demonstrar capacidade de valorização das diferenças sociais, culturais e raciais.	O aluno foi capaz de trabalhar em equipe respeitando o próximo e sua voz de fala ?	F				
	O aluno foi capaz de se adaptar a diferentes equipes durante o desenvolvimento dos projetos ?	S				
		F				
		S				

	6.Capacidade de solucionar problemas	O aluno foi capaz de solucionar os problemas encontrados durante o desenvolvimento do aplicativo utilizando debug, testes unitários ou testes de integração ?	F				
		O aluno foi capaz de gerenciar o tempo e recursos disponíveis para a resolução do problema ?	S				
			F				
			S				

# Critérios de avaliação

NÍVEIS DE DESEMPENHO	NÍVEIS	NOTA
Atingiu todos os critérios críticos e todos desejáveis	26	100
Atingiu todos os critérios críticos e 12 dos desejáveis	25	95
Atingiu todos os critérios críticos e 11 dos desejáveis	24	90
Atingiu todos os critérios críticos e 7 dos desejáveis	20	85
Atingiu todos os critérios críticos e 6 dos desejáveis	19	80
Atingiu todos os critérios críticos e 5 dos desejáveis	18	75
Atingiu todos os critérios críticos e 3 dos desejáveis	16	70
Atingiu todos os critérios críticos e 2 dos desejáveis	15	60
Atingiu todos os critérios críticos e nenhum desejável	13	50
Atingiu 8 critérios críticos e quaisquer desejáveis	9	40
Atingiu 7 critérios críticos e quaisquer desejáveis	8	30
Atingiu entre 1 e 4 critérios críticos e quaisquer desejáveis	5	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**.

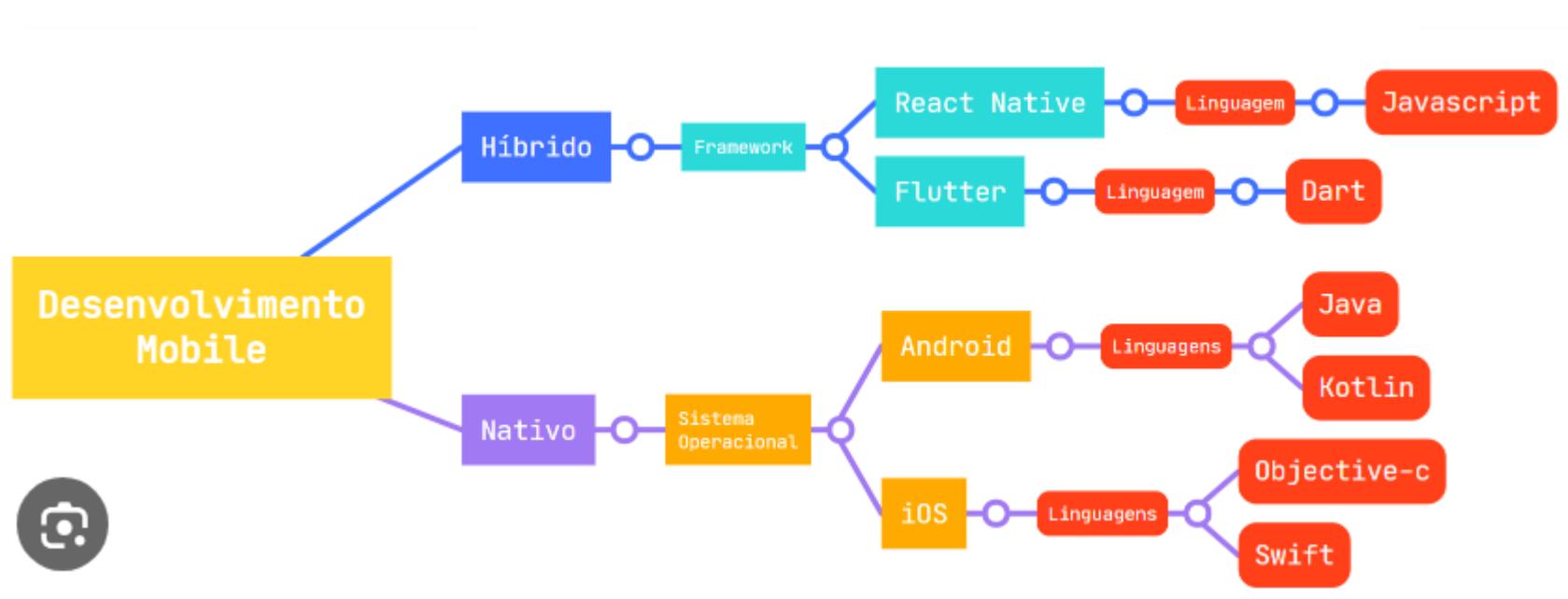
**Total do projeto: R\$ 18.663,00 / mês**

**Prazo Médio: 2 meses**

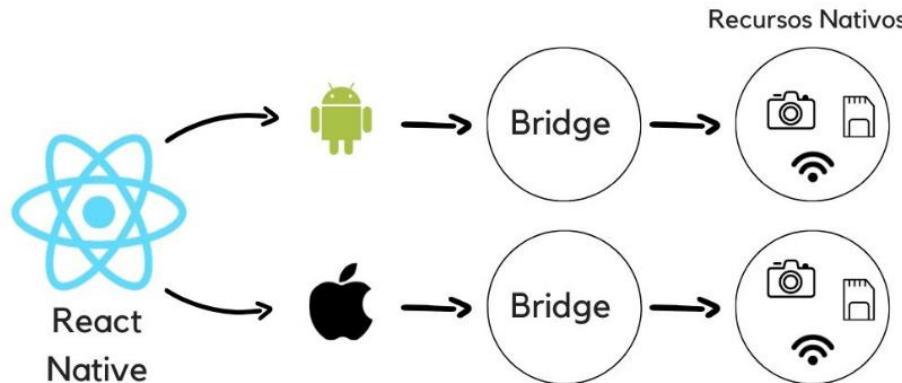
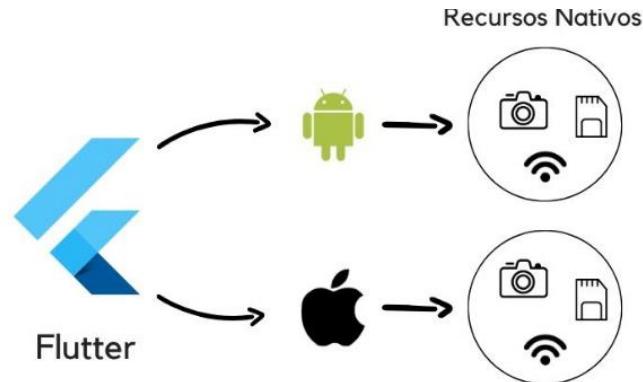
Subtotal: R\$ 37.326 + R\$ 7.100

**Total Estimado : R\$ 44.426**

# Aplicativos Mobile - Nativo ou híbrido



# Aplicativos Mobile - Frameworks



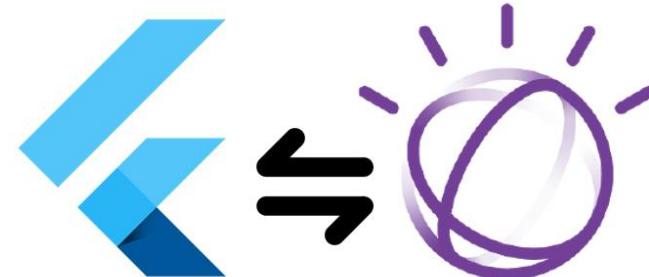
# Aplicativos Mobile - Flutter

- 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 - 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 interface. On the left, the code editor displays the `main.dart` file with Dart code for a simple counter application. On the right, a mobile phone emulator shows the resulting app with a red title bar and white body. The title bar says "Primeiro aplicativo em Flutter". The app's body contains the number "7" above three buttons: "Incrementar" (green), "Reset" (blue), and "Decrementar" (red).

```
main.dart x
projeto-flutter > projeto3 > lib > main.dart > 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-86pWq6tzA8zHcevK57S8Mx0aAl7eEALw\\_wcB&gclsrc=aw.ds](https://developer.android.com/studio?gclid=Cj0KCQjw2eilBhCCARIsAG0Pf8vTiRnMeJg5uKGukaJuvs-Y54bJas-86pWq6tzA8zHcevK57S8Mx0aAl7eEALw_wcB&gclsrc=aw.ds)

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

Extensões Flutter

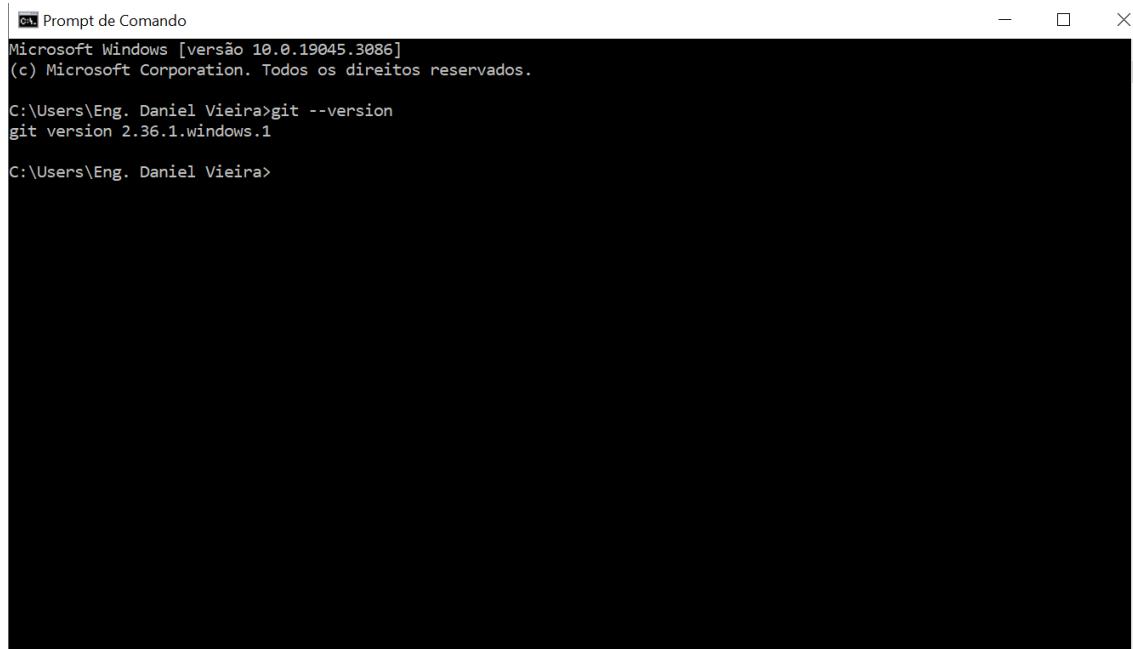
# Git instalação

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](https://git-scm.com/)) with a light beige background. At the top left is the Git logo (a red diamond with a white 'g') and the word 'git'. To its right is a search bar with the placeholder 'Search entire site...'. On the left side, there's a sidebar with links: 'About', 'Documentation', 'Downloads' (which is highlighted in red), 'GUI Clients', 'Logos', and 'Community'. Below this is a box containing text about the 'Pro Git book' by Scott Chacon and Ben Straub. The main content area has a heading 'Download for Windows'. It includes a call-to-action button 'Click here to download', a note about the latest version (2.41.0), and links for '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.'. It also shows how to use the 'winget' tool for installation. At the bottom, there's a section titled 'Now What?' with the text '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 dark blue header with the Flutter logo, navigation links like 'Multi-Platform', 'Development', 'Ecosystem', 'Showcase', 'Docs', and a search bar. A 'Get started' button is prominently displayed. Below the header, a banner says 'Read the announcement!' and 'Set up an editor'. On the left, a sidebar titled 'Get started' lists steps: 1. Install, 2. Set up an editor, 3. Test drive, 4. Write your first app, and 5. Learn more. It also includes sections for 'From another platform?' (Dart language overview), 'Stay up to date', 'Samples & tutorials', 'User interface', 'Navigation & routing', 'Data & backend', 'Accessibility & localization', 'Platform integration', and 'Packages & plugins'. The main content area is titled 'Install' and shows the path 'Get started > Install'. It asks to 'Select the operating system on which you are installing Flutter:' with four options: Windows (Windows logo), macOS (apple logo), Linux (Ubuntu logo), and ChromeOS (Chrome logo). A yellow callout box at the bottom left says 'Important: If you're in China, read [Using Flutter in China](#)'. A 'Set up an editor' link is located at the bottom right.

# Instalação do Flutter

## 2º Passo : Baixar a versão do Flutter

The screenshot shows the official Flutter website's "Get started" section. On the left, there's a sidebar with navigation links like "Get started", "Stay up to date", "Samples & tutorials", and various development categories. The main content area has a heading "Get the Flutter SDK". It includes a yellow box with an important note about China, a "Help" link, and a large blue button labeled "flutter\_windows\_3.10.6-stable.zip" which is highlighted with a blue border. Below this, there's a paragraph about other release channels and an "SDK archive". A second yellow box at the bottom contains a warning about installing to paths with special characters.

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](#).

Help

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.

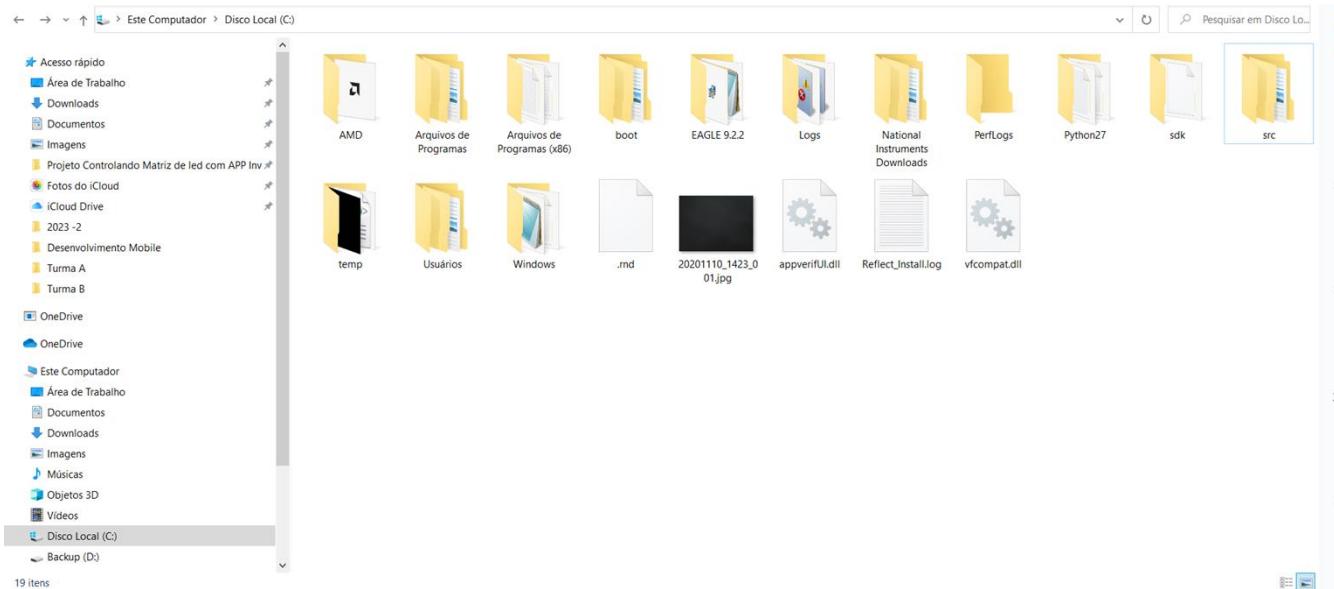
# Instalação do Flutter

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.

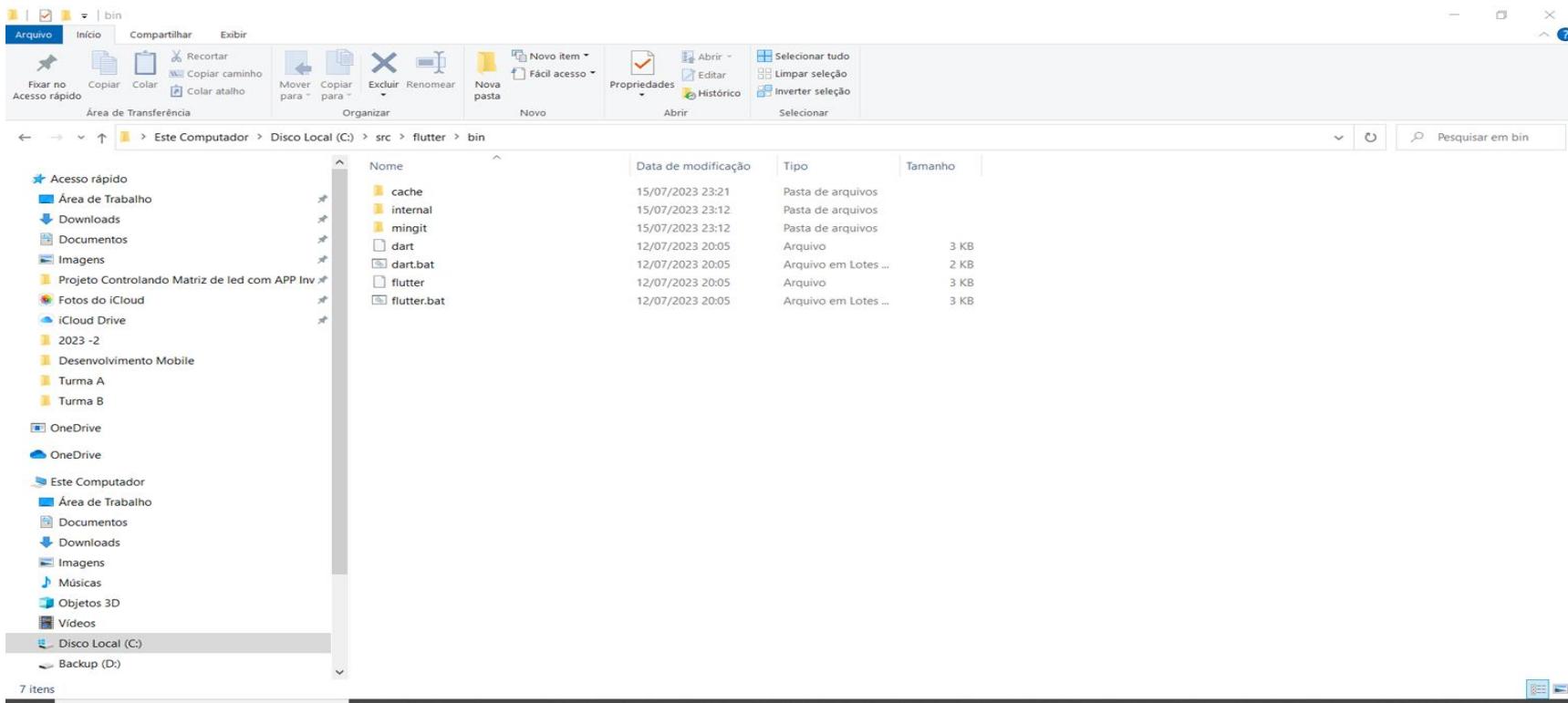
5º Passo: Copiar a pasta flutter para a pasta src



# Instalação do Flutter

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

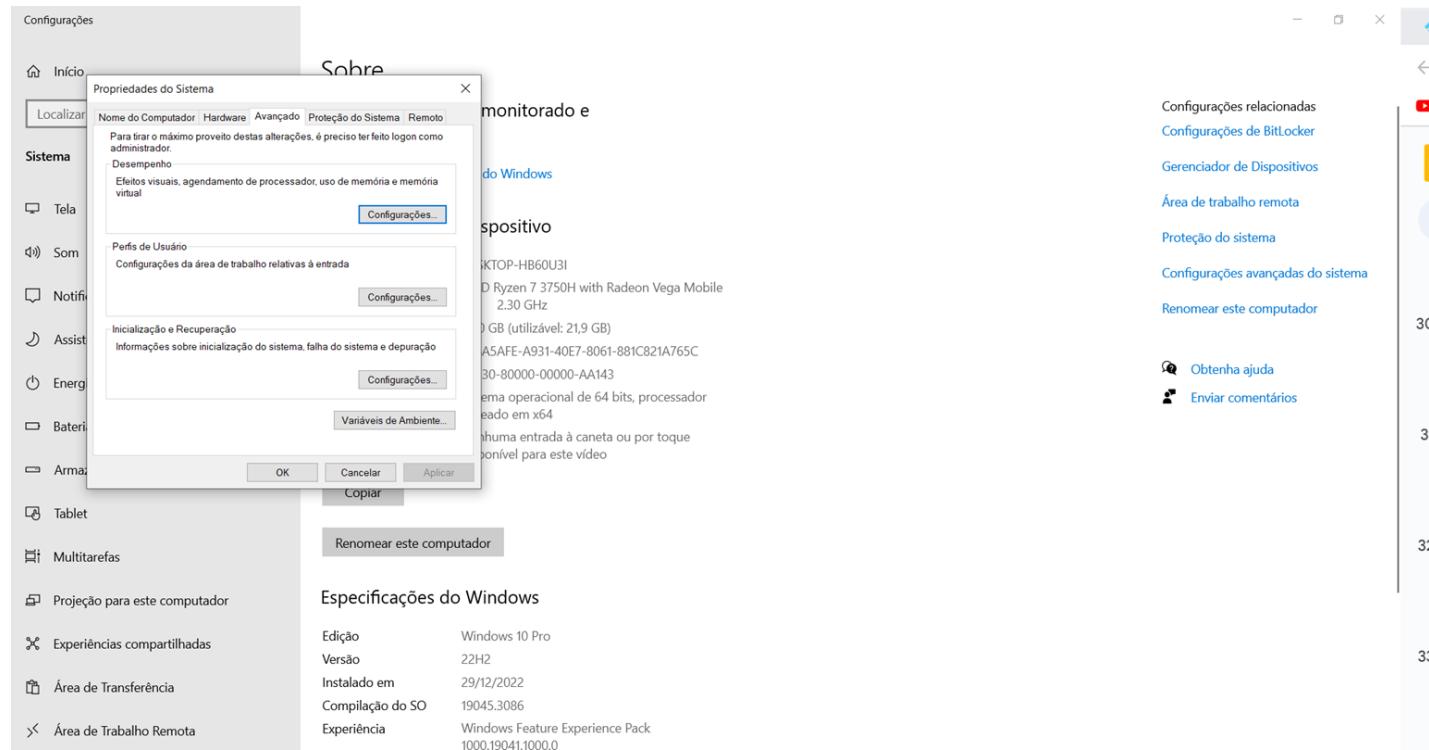
C:\src\flutter\bin



# Instalação do Flutter

## 7º Passo : Configurar variáveis de ambiente

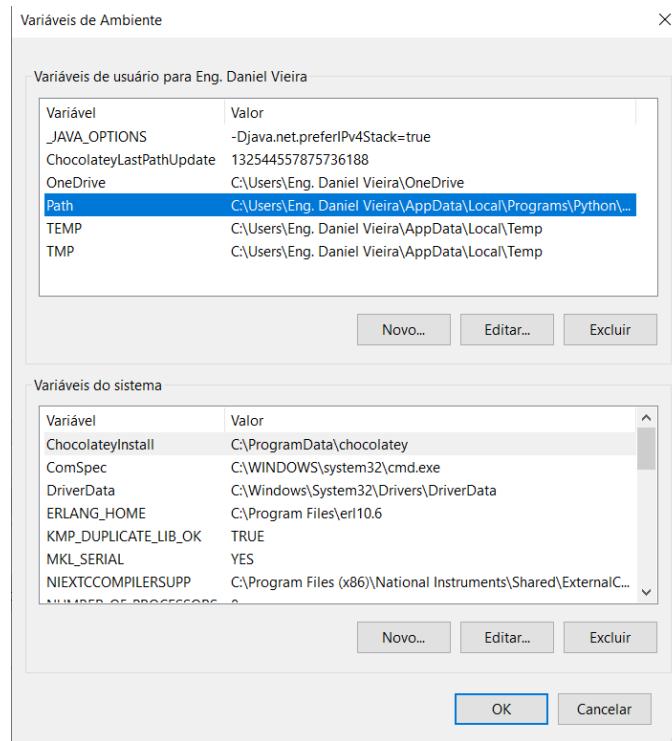
Meu Computador -> Propriedades -> Configurações avançadas do sistema



# Instalação do Flutter

## 8º Passo : Configurar variáveis de ambiente

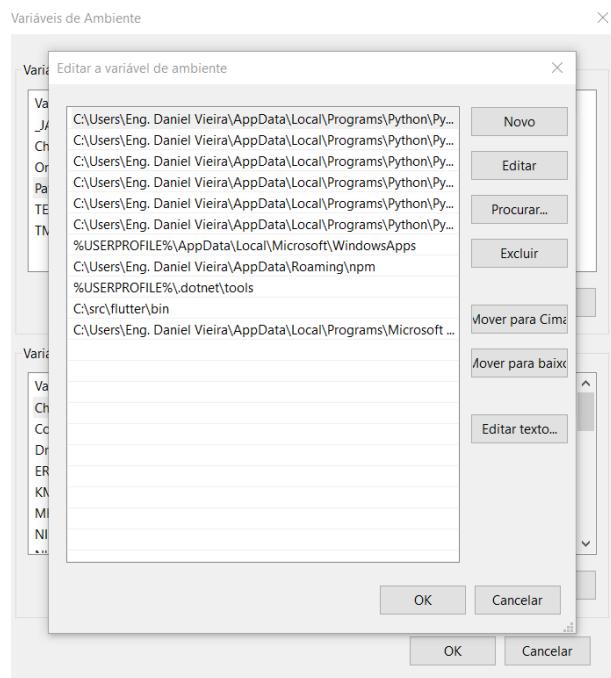
Clicar em variáveis de ambiente -> Procurar pelo campo Path e clicar em editar



# Instalação do Flutter

## 9º Passo : Configurar variáveis de ambiente

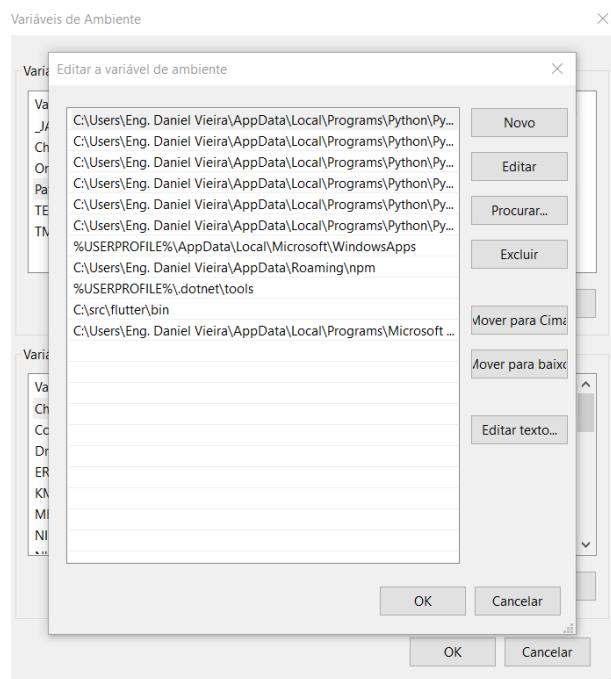
Procurar pelo campo Path e clicar em editar-> Novo e colar o caminho copiado anteriormente  
C:\src\flutter\bin



# Instalação do Flutter

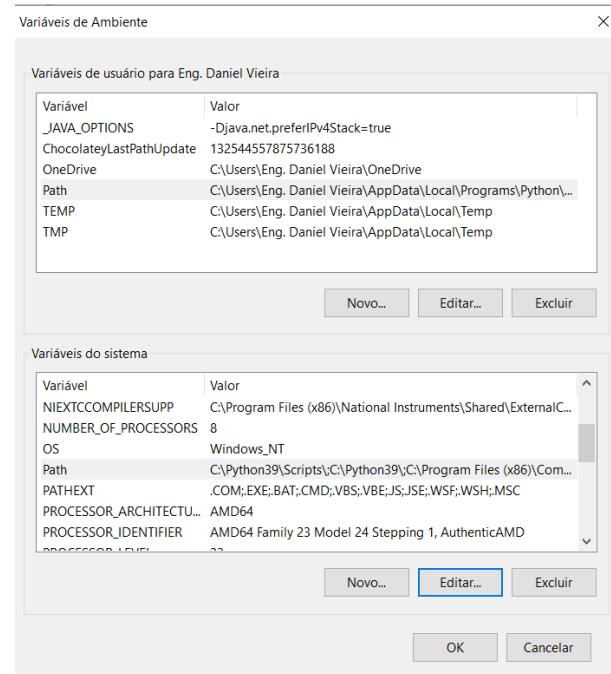
## 9º Passo : Configurar variáveis de ambiente

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

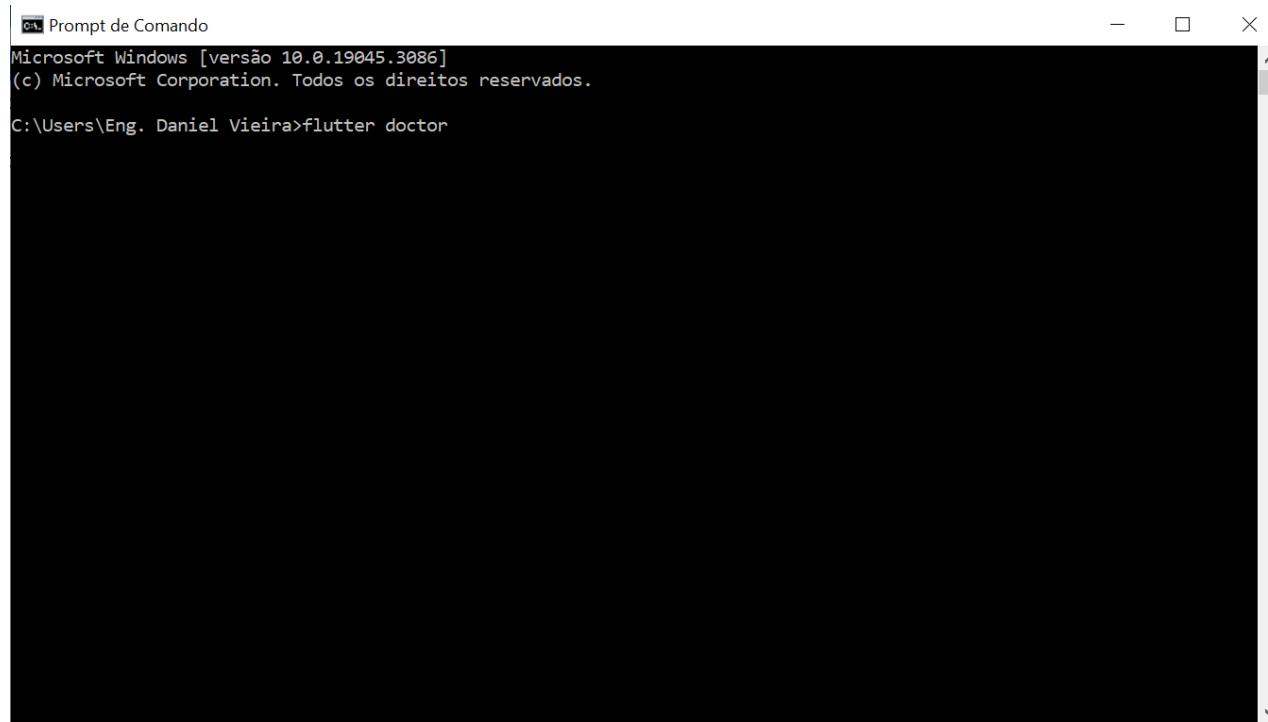


# Instalação do Flutter

10º Passo

Menu iniciar -> cmd

11º Passo digitar flutter doctor para verificar os componentes instalados

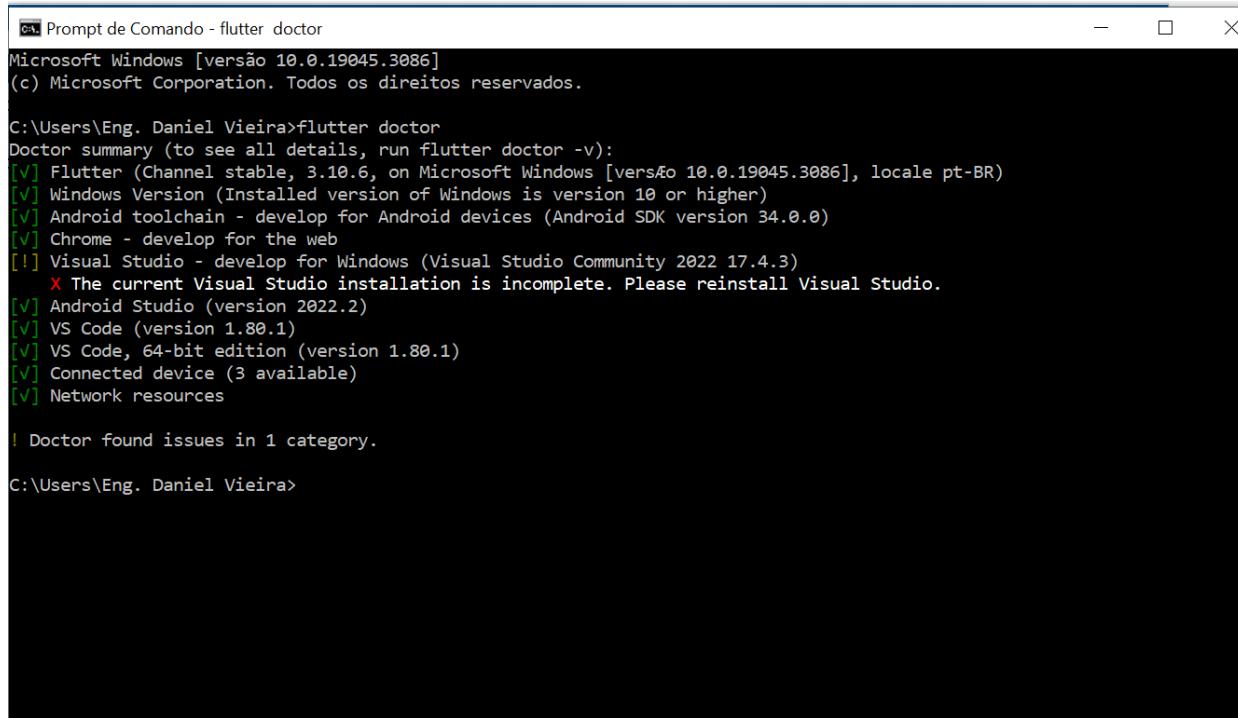


```
Prompt de Comando
Microsoft Windows [versão 10.0.19045.3086]
(c) Microsoft Corporation. Todos os direitos reservados.

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. At the top, there's a navigation bar with links for 'Essentials', 'Design & Plan', 'Docs' (which is underlined), and 'Google Play'. On the right side of the header are 'Search', 'Português - ...', 'Android Studio', and 'Fazer login'. Below the header, the page title 'ANDROID STUDIO' is displayed, followed by tabs for 'Download' (which is selected), '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 download icon. To the right of the description, there's a screenshot of the Android Studio interface showing code in the editor and a preview of the application in the design tab. At the bottom of the page, there's a section titled 'Read release notes' with a link and a small icon.

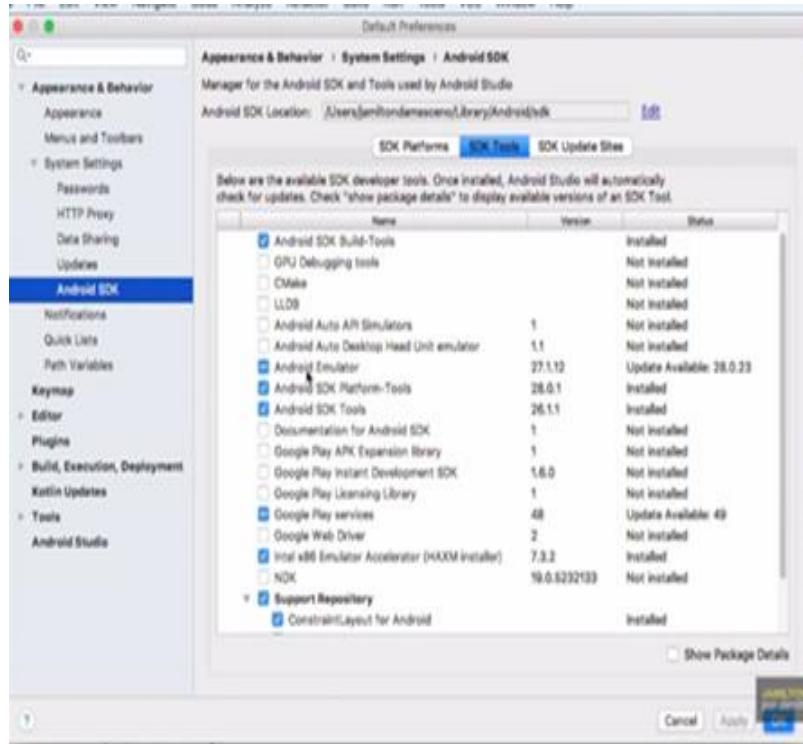
# Instalação do SDK Android

## 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 'Essentials', 'Design & Plan', 'Docs', and 'Google Play'. On the right side of the header are 'Search', 'Português - ...', 'Android Studio', and 'Fazer login' buttons. Below the header, the page title 'ANDROID STUDIO' is displayed, followed by tabs for 'Download', '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 download icon. To the right of the text, there's a preview window showing the Android Studio interface with code in the editor and a preview of the UI in the design tab. At the bottom of the page, there's a section titled 'Read release notes' with a link and a small icon.

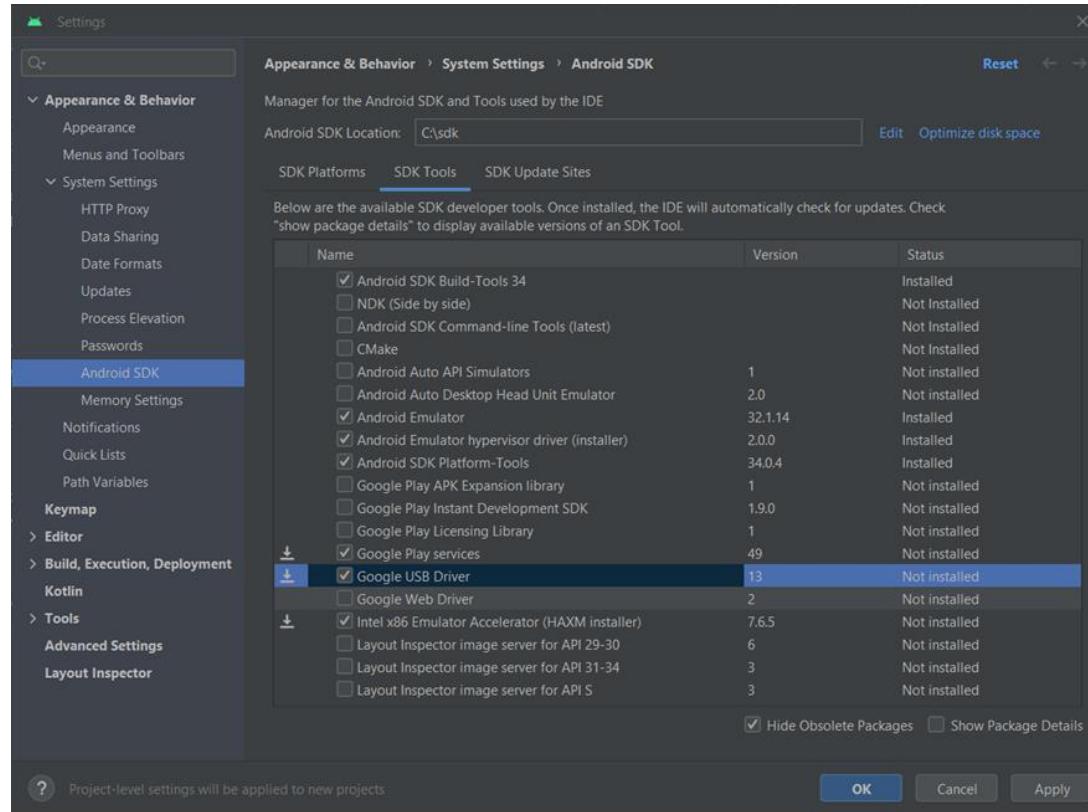
# Instalação do SDK Android

## 2º Instalar o Android Studio



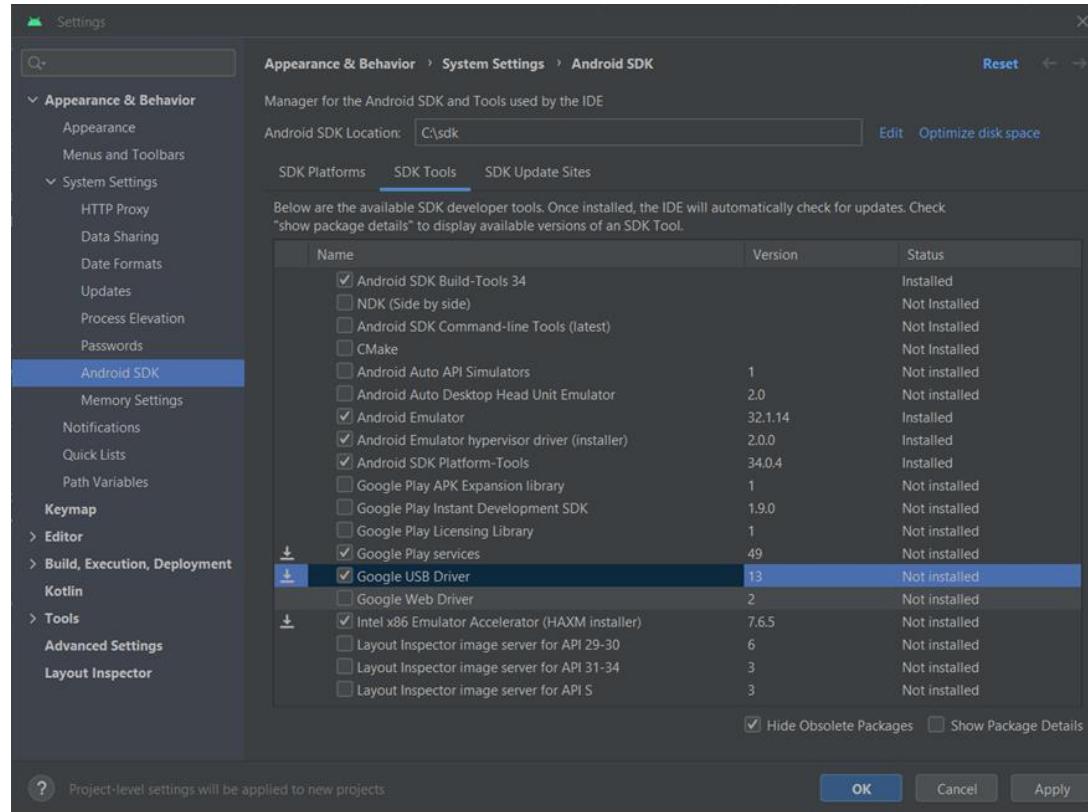
# Instalação do SDK Android

## 2º Instalar o SDK



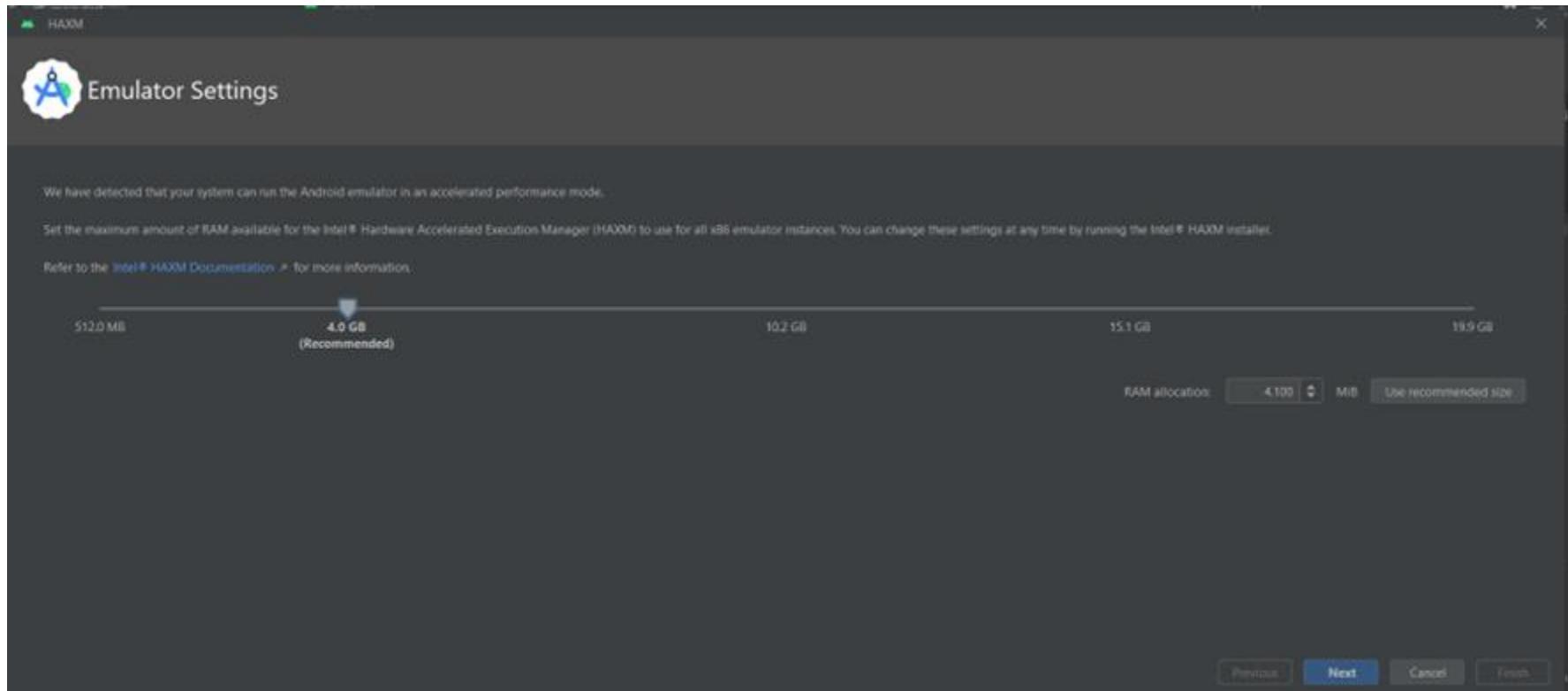
# Instalação do SDK Android

## 2º Instalar o SDK



# Instalação do SDK Android

## 3º Instalar o Emulador



# Instalação do SDK Android

## 4º Concordar com a licença

 License Agreement  
Read and agree to the licenses for the components which will be installed

Licenses

Intel® x86-Emulator Accelerator (HAXM installer)

Intel® Hardware Accelerated Execution Manager End-User License Agreement  
Copyright (c) 2012 Intel Corporation. All rights reserved.

Redistribution, Redistribution and use in binary form, without modifications, are permitted provided that the following conditions are met:

1. Redistributions must reproduce the above copyright notice and the following disclaimer in the documentation and/or other materials provided with the distribution.
2. Neither the name of Intel Corporation nor the names of its suppliers may be used to endorse or promote products derived from this software without specific prior written permission.
3. No reverse engineering, de-compilation, or disassembly of this software is permitted. Limited patent license. Intel Corporation grants a world-wide, royalty-free, non-exclusive license under patents it now or hereafter owns or controls to make, have made, use, import, offer to sell and sell ("Utilize") this software, but solely to the extent that any such patent is necessary to Utilize the software alone. The patent license shall not apply to any combinations which include this software. No hardware per se is licensed hereunder.

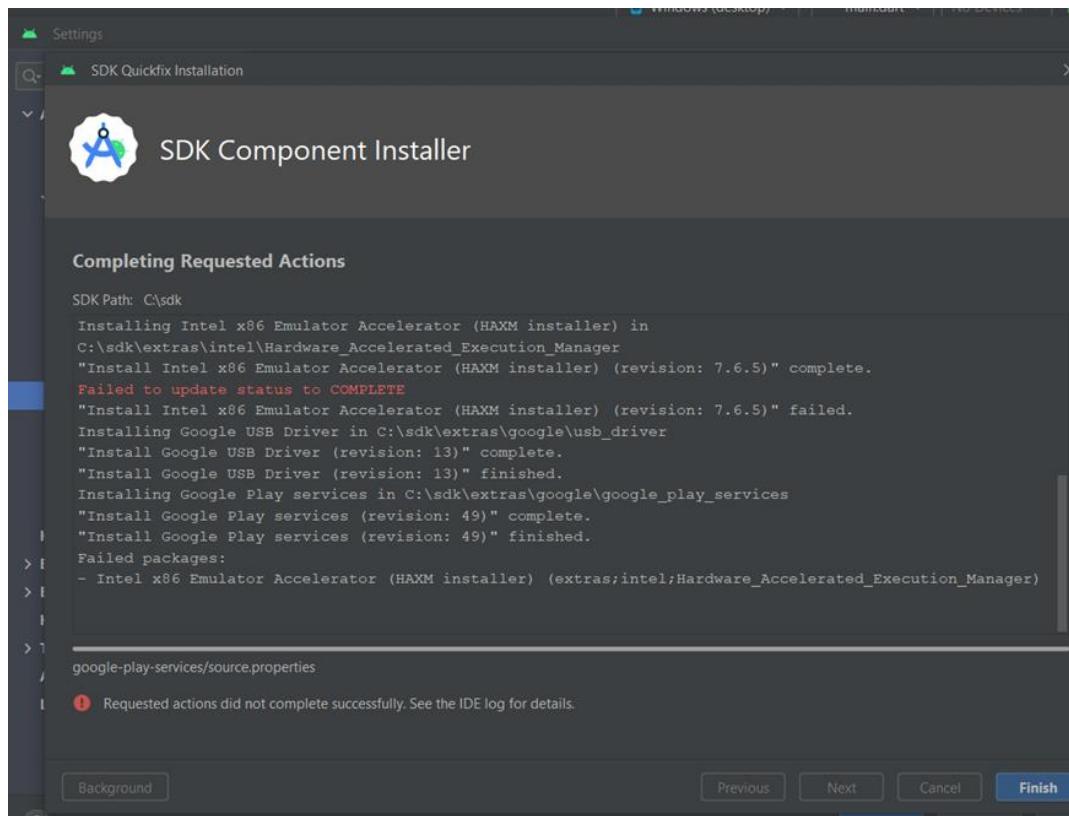
DISCLAIMER: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Decline  Accept

Previous Next Cancel Finish

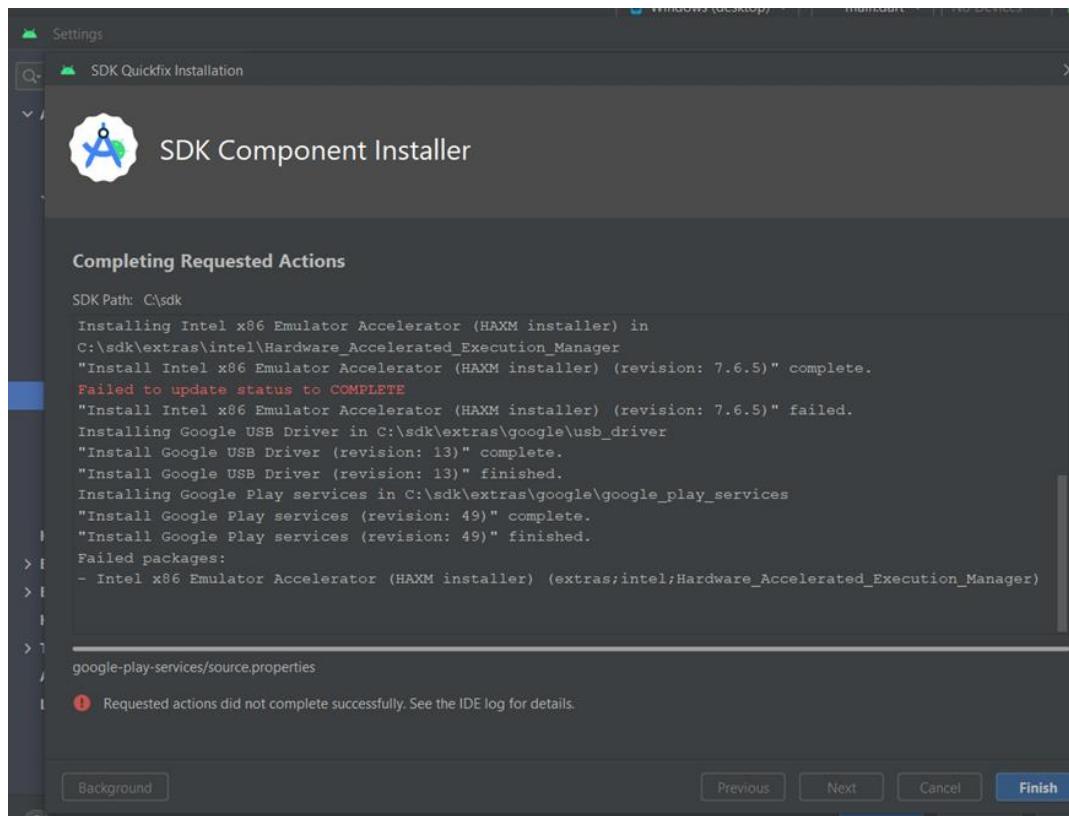
# Instalação do SDK Android

## 5º Instalação do Emulador



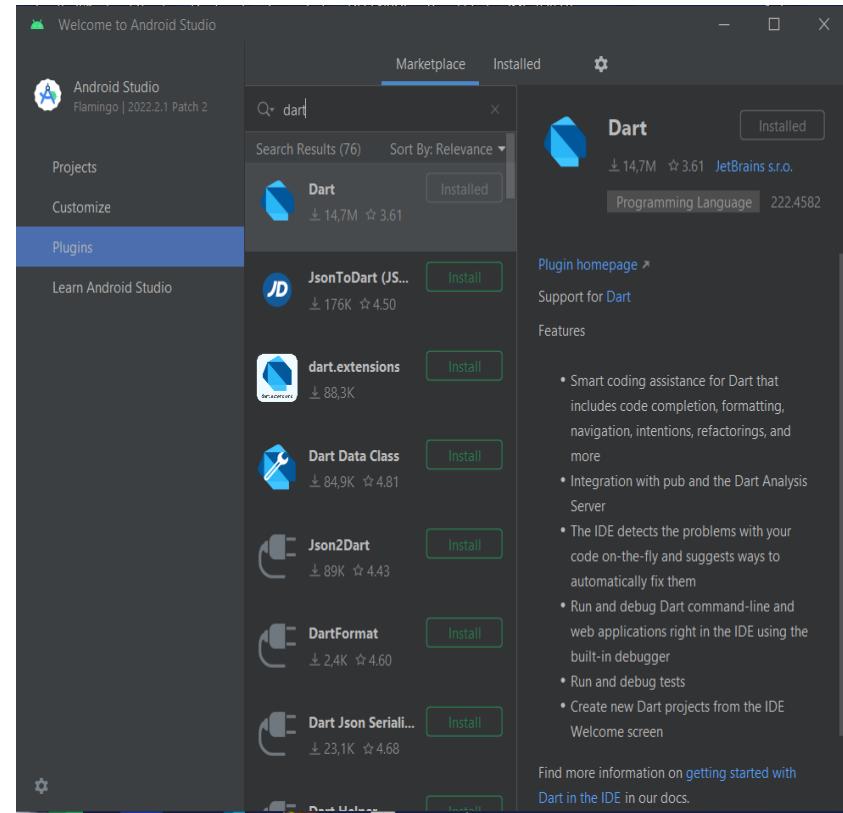
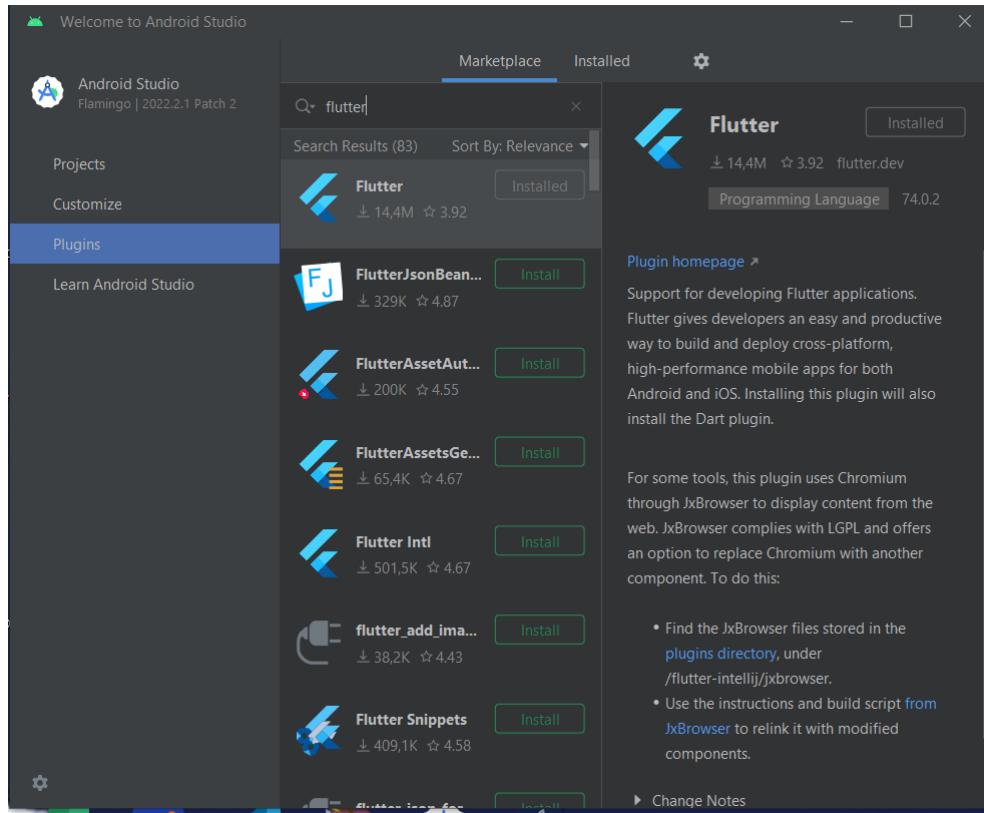
# Instalação do SDK Android

## 5º Instalação do Emulador



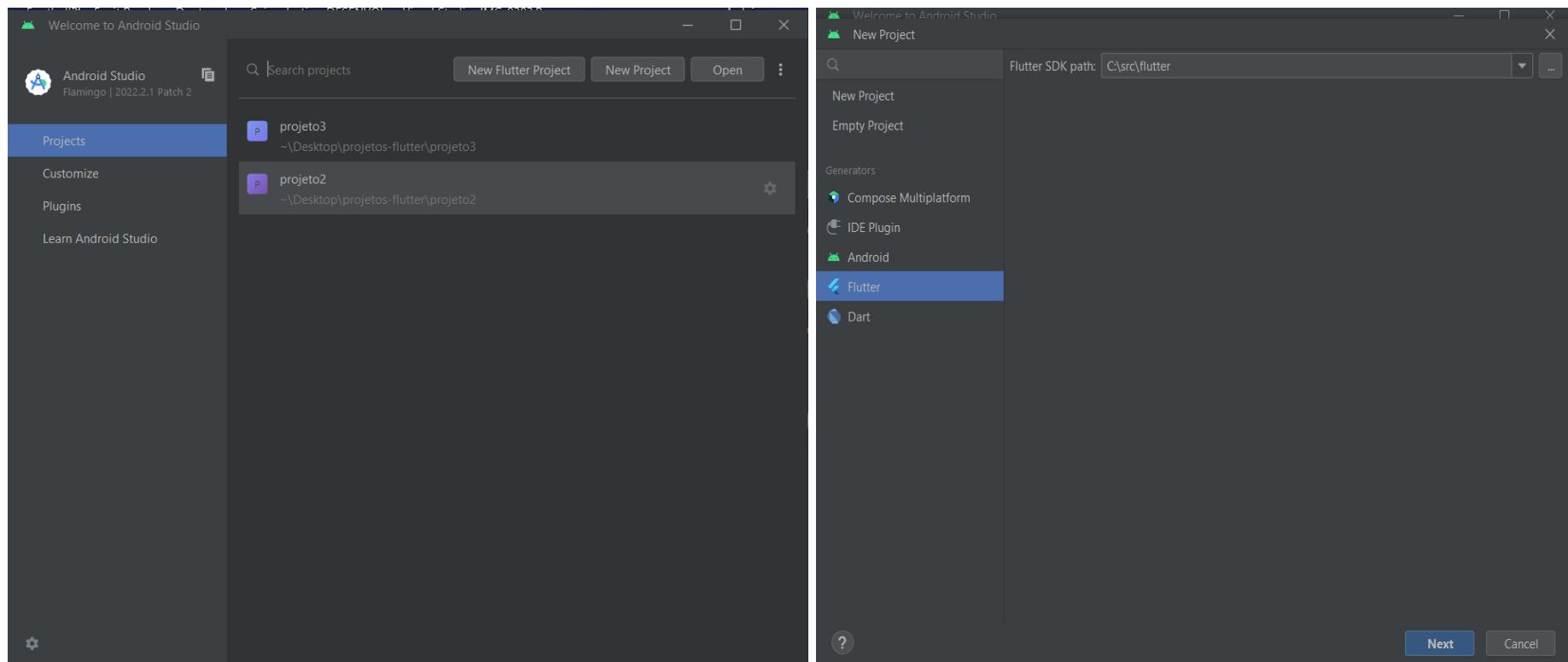
# Instalação do SDK Android

## 6º Instalação dos plugins Flutter e Dart



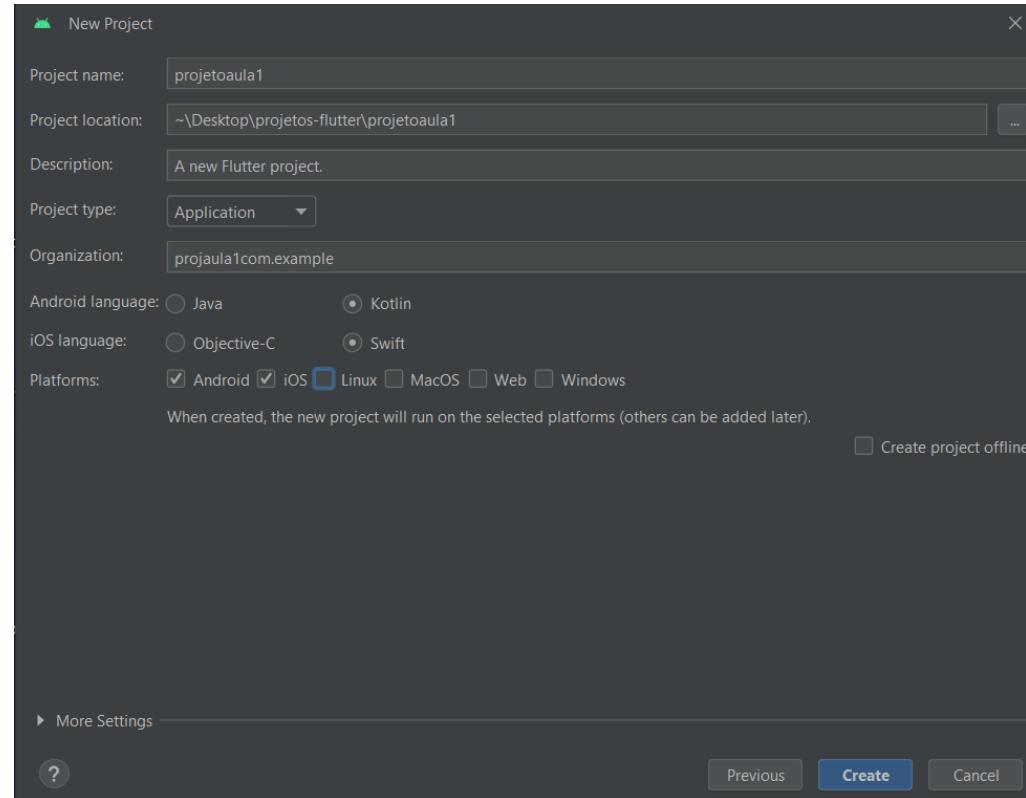
# Instalação do SDK Android

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



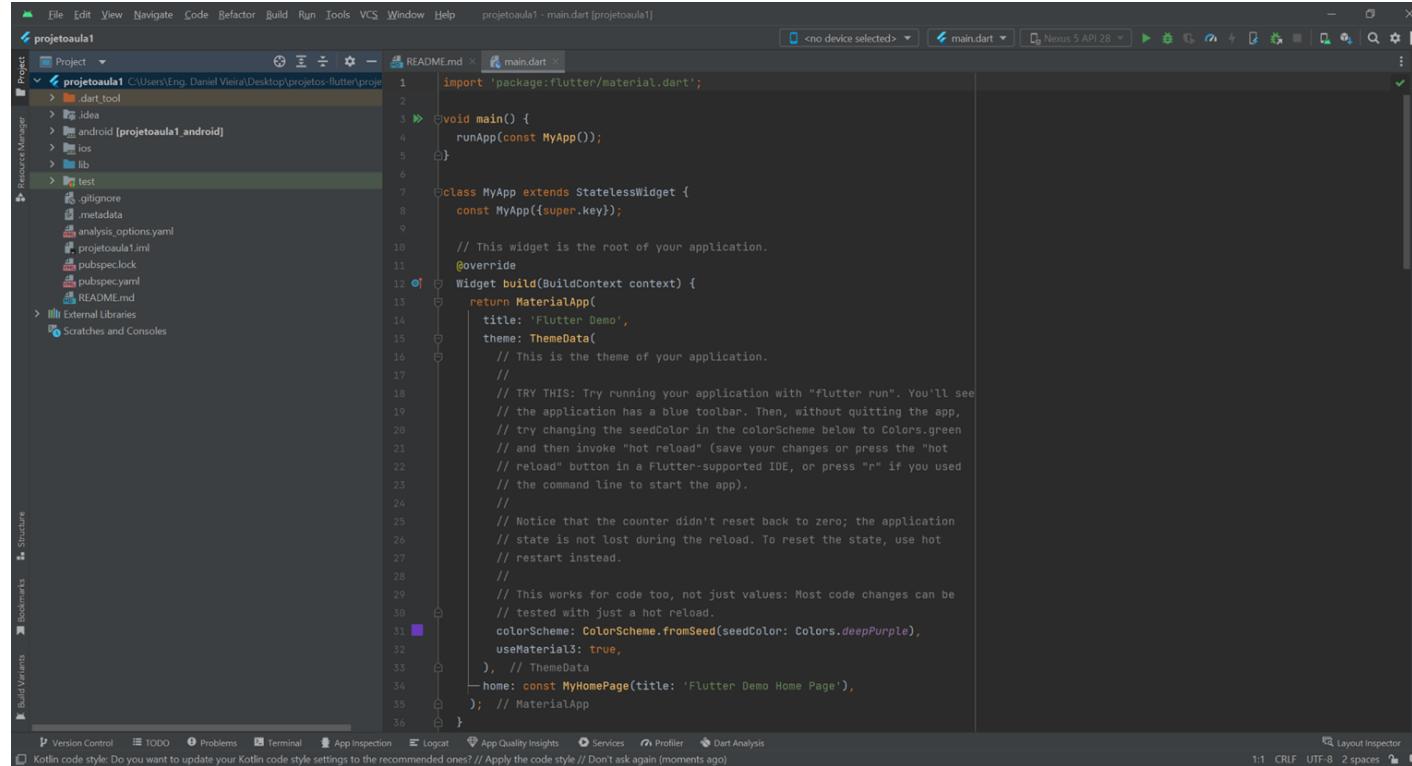
# Instalação do SDK Android

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



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

- Project Bar:** Shows "projetoaula1 - main.dart [projetoaula1]"
- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help
- Toolbar:** Includes icons for Device Manager, Layout Inspector, and others.
- Project Tree:** Shows the project structure with folders like .dart\_tool, .idea, android, ios, lib, test, and subfolders like .gitignore, .metadata, analysis\_options.yaml, projetoaula1.iml, pubspec.lock, pubspec.yaml, README.md, External Libraries, and Scratches and Consoles.
- Code Editor:** The main.dart file is open, showing the following code:

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a blue toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
        // and then invoke "hot reload" (save your changes or press the "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you used
        // the command line to start the app).
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
        //
        // This works for code too, not just values: Most code changes can be
        // tested with just a hot reload.
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ), // ThemeData
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    ); // MaterialApp
}
```

**Bottom Navigation Bar:** Version Control, TODO, Problems, Terminal, App Inspection, Logcat, App Quality Insights, Services, Profiler, Dart Analysis.

**Status Bar:** Kotlin code style: Do you want to update your Kotlin code style settings to the recommended ones? // Apply the code style // Don't ask again (moments ago)

**Bottom Right:** 1:1, CR LF, UTF-8, 2 spaces, Layout Inspector.

## Instalação do SDK Android

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

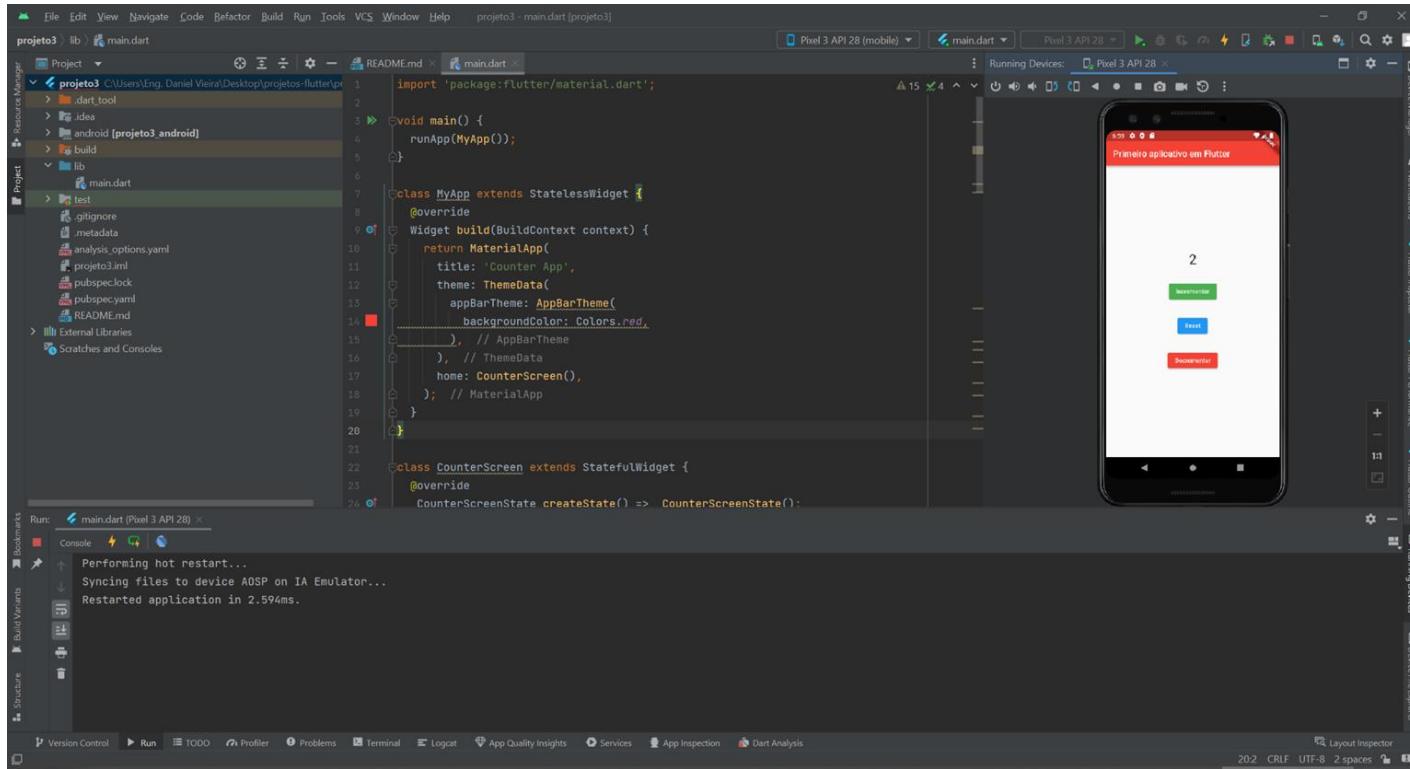
The screenshot shows the Android Studio interface with the following details:

- File Structure:** The left sidebar displays the project structure with files like README.md and main.dart.
- Code Editor:** The main editor shows the Dart code for the main application entry point. The code initializes a MaterialApp with a title of 'Flutter Demo' and a green theme. It includes comments about running with 'flutter run' and a note that the counter doesn't reset to zero.
- Flutter Outline:** A floating window titled "Flutter Outline" shows the current state of the application, indicating it's running on a Pixel 3 API 28 device.
- Running Device:** A large preview window shows a smartphone screen with the app's home page. The page has a pink header with the text "Flutter Demo Home Page". Below the header, there's a message: "You have pushed the button this many times: 0".
- Run Tab:** The bottom navigation bar shows the "Run" tab is active, with the command "main.dart (Pixel 3 API 28)" selected.
- Console Output:** The bottom-left panel shows the terminal output for a hot reload:

```
Performing hot reload...
Syncing files to device AOSP on IA Emulator...
Reloaded 1 of 665 libraries in 1.002ms (compile: 51 ms, reload: 402 ms, reassemble: 329 ms).
```

# 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](mailto:danielvieira2006@gmail.com)

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

