



Desenvolvimento
Mobile 3
Aula 01

Prof. Me Daniel Vieira

SENAI

Agenda

- 1- Apresentação
- 2 - Ementa do curso
- 3- Capacidades técnicas e Socioemocionais
- 4 - Calendário das aulas
- 5 - Critérios de avaliação
- 6 - Aplicativos Mobile
- 7 - Configuração do ambiente para desenvolvimento
(Flutter, Git, SDK Android)

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
5	Capacidades Técnicas 1. Configurar o ambiente de desenvolvimento para aplicações Mobile. Capacidades Socioemocionais 1. Demonstrar atenção a detalhes 2. Demonstrar organização	Apresentação da disciplina Exemplos de projetos Ementa do curso Metodologia das aulas (Aulas expositivas e projetos) -	Exposição Dialogada + Demonstração de como configurar o VSCode para criar projetos com Flutter. Instrumento de avaliação: Exercício para criar projeto no VSCode Composição da nota final da UC: Média das notas dos aplicativos desenvolvidos .	Quadro Branco/ Projetor multimídia
10	Capacidades Técnicas 1. Aplicar tecnologias de acordo com o sistema operacional 2. Elaborar programação Mobile de acordo com os padrões de desenvolvimento. Capacidades Socioemocionais 1. Demonstrar atenção a detalhes 2. Demonstrar capacidade de tomar decisão 3. Demonstrar organização 4. Demonstrar pensamento criativo	Princípios de UX para desenvolvimento Mobile 1.1 Simplicidade 1.2 Consistência 1.3 Feedback Visual 1.4 Navegação Intuitiva 1.5 Legibilidade 1.6 Foco no Usuário 1.7 Responsividade	Exposição dialogada + Desenvolvimento do aplicativo (Hands On) para permitir ao usuário a escolha do tipo de combustível de acordo com a distância percorrida e o valor do combustível com Radio Button, Check Box.	Quadro Branco/ Projetor Multimídia Computador com a IDE VSCode, Framework Flutter, extensão Dart instalada, emulador Android .

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
		1.8 Feedback claro de erro 1.9 Performance otimizada 1.10 Acessibilidade	Instrumento de avaliação: Teste do aplicativo e avaliação do código no GitHub conforme tabela de critérios críticos e desejáveis.	.

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
5	<p>Capacidades Técnicas</p> <p>1. Criar interfaces gráficas da aplicação Mobile baseadas em UX.</p> <p>Capacidades Socioemocionais</p> <p>1. Demonstrar atenção a detalhes</p> <p>2. Demonstrar capacidade de tomar decisão</p> <p>3. Demonstrar organização</p> <p>4. Demonstrar pensamento criativo</p> <p>5. Demonstrar capacidade de solucionar problemas</p>	<p>2. Integração com APIs</p> <p>2.1 Requisições</p> <p>2.1.1 HTTP</p> <p>2.1.2 HTTPS</p> <p>2.2 Consumo</p> <p>2.2.1 De APIs</p> <p>2.2.2 De Web Services</p>	<p>Exposição dialogada + Demonstração Criação de um APP para consumo de API</p> <p>Instrumento de avaliação: Teste do aplicativo e avaliação do código no GitHub conforme tabela de critérios críticos e desejáveis.</p> <p>.</p>	<p>Quadro Branco/ Projetor Multimídia,</p> <p>Computador com a IDE VSCode, Framework Flutter, extensão Dart instalada, emulador Android .</p>

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
10	Capacidades Técnicas 1. Elaborar programação Mobile de acordo com os padrões de desenvolvimento. 2. Criar interfaces gráficas da aplicação Mobile baseadas em UX.	3 Persistência de dados 3.1. Estrutura de dados persistentes 3.2. Preferências da aplicação	Exposição dialogada + Mentimeter (Nuvem de palavras) para início da aula para fazer o Check in antes de iniciar o conteúdo novo. Demonstração Desenvolvimento de aplicativo (Hands On)	Quadro Branco/ Projetor Multimídia, Computador com a IDE VSCode, Framework Flutter, extensão Dart instalada, emulador Android .
10	Capacidades Socioemocionais 1. Demonstrar atenção a detalhes 2. Demonstrar capacidade de tomar decisão 3. Demonstrar organização 4. Demonstrar pensamento criativo 5. Demonstrar capacidade de solucionar problemas		Criação de um APP para armazenar dados no banco de dados Instrumento de avaliação: Teste do aplicativo e avaliação do código no GitHub conforme tabela de critérios críticos e desejáveis.	
10	Capacidades Técnicas	4. Acesso a programação de periféricos 4.1 Câmera	Exposição dialogada	Quadro Branco/ Projetor Multimídia,

Plano de Ensino

Página 3 de 25

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
	<ol style="list-style-type: none">Aplicar tecnologias de acordo com o sistema operacionalElaborar programação Mobile de acordo com os padrões de desenvolvimento. <p>Capacidades Socioemocionais</p> <ol style="list-style-type: none">Demonstrar atenção a detalhesDemonstrar organizaçãoDemonstrar pensamento criativoDemonstrar capacidade de solucionar problemas	<p>4.2 Bluetooth</p> <p>4.3 GPS</p> <p>4.4 Sensores embarcados</p>	<p>Demonstração</p> <p>Desenvolvimento de aplicativo para consumo de API (Hands On)</p> <p>Instrumento de avaliação:</p> <p>Criar aplicativo para consumir API e postar o código no GitHub para testes e avaliação conforme tabela de critérios críticos e desejáveis.</p>	<p>Computador com a IDE VSCode, Framework Flutter, extensão Dart instalada, emulador Android .</p>
10	Capacidades Técnicas	5 Recursos Mobile	Exposição Dialogada + Demonstração	Quadro Branco/ Projetor Multimídia,

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
	<p>1. Elaborar programação Mobile de acordo com os padrões de desenvolvimento.</p> <p>Capacidades Socioemocionais</p> <ul style="list-style-type: none">1. Demonstrar atenção a detalhes2. Demonstrar capacidade de tomar decisão3. Demonstrar organização4. Demonstrar pensamento criativo5. Demonstrar capacidade de solucionar problemas	<p>5.1 Notificações locais</p> <p>5.2 Push Notifications</p>	<p>Desenvolvimento de aplicativo para exibir alertas e notificação (Hands On).</p> <p>Instrumento de avaliação:</p> <p>Realizar a configuração do banco de dados e mostrar as etapas realizadas por meio de uma entrega no Git Hub +</p> <p>Criar aplicativo para salvar informações para salvar informações no banco de dados e postar o código no Git Hub para testes e avaliação conforme tabela de critérios críticos e desejáveis.</p>	<p>Computador com a IDE VSCode, Framework Flutter, extensão Dart instalada, emulador Android , Firebase instalado.</p>
10	Capacidades Técnicas	5.3 Processamento Multi threads	Exposição dialogada +	Quadro Branco/ Projetor Multimídia,

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO

Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
	<p>1. Elaborar programação Mobile de acordo com os padrões de desenvolvimento.</p> <p>Capacidades Socioemocionais</p> <p>1. Demonstrar atenção a detalhes 2. Demonstrar capacidade de tomar decisão 3. Demonstrar organização 4. Demonstrar pensamento criativo 5. Demonstrar capacidade de solucionar problemas</p>	<p>5.4 Tarefas Background 5.5 Serviços Background 5.6 Mapas 5.7 SMS</p>	<p>Apresentação sobre Threads, tarefas background, mapas</p> <p>Desenvolvimento de projeto de app responsivo com threads</p> <p>Instrumento de avaliação: Situação de aprendizagem com entrega no GitHub e avaliação conforme tabela de critérios críticos e desejáveis.</p>	<p>Computador com a IDE VSCode, Framework Flutter, extensão Dart instalada, emulador Android, Firebase instalado.</p>
10	Capacidades Técnicas	2 Integração com APIs	Exposição dialogada + Demonstração	Quadro Branco/ Projetor Multimídia,

Plano de Ensino

Página 3 de 25

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
	<p>1. Elaborar programação Mobile de acordo com os padrões de desenvolvimento.</p> <p>Capacidades Socioemocionais</p> <ul style="list-style-type: none">1. Demonstrar atenção a detalhes2. Demonstrar capacidade de tomar decisão3. Demonstrar organização4. Demonstrar pensamento criativo5. Demonstrar capacidade de solucionar problemas	<p>Métodos de requisição de uma API</p> <p>2.2 Consumo</p> <p>2.2.1 De APIs</p> <p>2.2.2 De Web Services</p>	<p>Desenvolvimento de um aplicativo consumindo um Web Services e exibindo informações na tela do aplicativo</p> <p>Instrumento de avaliação: Postar o código no GitHub para testes e avaliação conforme tabela de critérios críticos e desejáveis.</p>	<p>Computador com a IDE VSCode, Framework Flutter, extensão Dart instalada, emulador Android, Firebase instalado .</p>
10	Capacidades Técnicas	1 Princípios de UX para desenvolvimento Mobile	Exposição dialogada sobre Web Services e início da situação de aprendizagem	Quadro Branco/ Projetor Multimídia,

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
	<p>1. Elaborar programação Mobile de acordo com os padrões de desenvolvimento.</p> <p>2. Criar interfaces gráficas da aplicação Mobile baseadas em UX.</p> <p>Capacidades Socioemocionais</p> <p>1. Demonstrar atenção a detalhes</p> <p>2. Demonstrar capacidade de tomar decisão</p> <p>3. Demonstrar organização</p> <p>4. Demonstrar pensamento criativo</p> <p>5. Demonstrar capacidade de solucionar problemas</p>	<p>1.1 Simplicidade</p> <p>1.2 Consistência</p> <p>1.3 Feedback Visual</p> <p>1.4 Navegação Intuitiva</p> <p>1.5 Legibilidade</p> <p>1.6 Foco no Usuário</p> <p>4 Acesso a programação de periféricos</p> <p>4.3 GPS</p> <p>5.6 Mapas</p>	<p>Desenvolvimento de um aplicativo com multi telas e acesso a programação de periféricos</p> <p>Instrumento de avaliação: Teste de funcionamento do aplicativo e avaliação do código no GitHub conforme tabela de critérios críticos e desejáveis.</p>	<p>Computador com a IDE VSCode, Framework Flutter, extensão Dart instalada, emulador Android, Firebase instalado.</p>
5	Capacidades Técnicas	<p>6 Publicação da aplicação</p> <p>6.1 Preparação e testes</p>	Exposição dialogada + Demonstração de comandos para gerar o apk no VSCode	Quadro Branco/ Projetor Multimídia,

Plano de Ensino

Página 10 de 25

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
	<p>3. Testar o código com ferramentas automatizadas (end to end) para garantia de qualidade do software, prevendo eventuais falhas.</p> <p>Capacidades Socioemocionais</p> <p>1. Demonstrar atenção a detalhes 2. Demonstrar capacidade de tomar decisão 3. Demonstrar organização</p>	<p>6.2 Geração da aplicação de produção (build)</p> <p>6.5 Hospedagem em lojas</p> <p>6.5.1 App Store</p> <p>6.5.2 Play Store</p>	<p>Nesta aula o objetivo é realizar testes no aplicativo e gerar o apk do aplicativo e publicar na Play Store.</p> <p>Instrumento de avaliação: apk instalado no celular e logs gerados no VSCode para gerar o apk.</p>	<p>Computador com a IDE VSCode, Framework Flutter, extensão Dart instalada, emulador Android, celular Android .</p>
10	Capacidades Técnicas	1. Princípios de UX para desenvolvimento Mobile	Exposição dialogada	Quadro Branco/ Projetor Multimídia, Computador com a IDE VSCode, Framework

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
	<p>1. Elaborar programação Mobile de acordo com os padrões de desenvolvimento.</p> <p>2. Criar interfaces gráficas da aplicação Mobile baseadas em UX.</p> <p>Capacidades Socioemocionais</p> <p>1. Demonstrar atenção a detalhes</p> <p>2. Demonstrar capacidade de tomar decisão</p> <p>3. Demonstrar organização</p> <p>4. Demonstrar pensamento criativo</p> <p>5. Demonstrar capacidade de solucionar problemas</p>	<p>1.1 Navegação Intuitiva</p> <p>1.2 Legibilidade</p> <p>1.3 Foco no Usuário</p> <p>4.2 Consumo</p> <p>4.2.1 De APIs</p> <p>4.3 GPS</p> <p>5. Persistência de dados</p>	<p>Apresentação dos requisitos do projeto e início do desenvolvimento</p> <p>Instrumento de avaliação: Situação de aprendizagem com entrega no <u>GitHub</u> e avaliação conforme tabela de critérios críticos e desejáveis.</p> <p>.</p>	Flutter, extensão Dart instalada, emulador Android, Firebase instalado .
5	Capacidades Técnicas	<p>1. Princípios de UX para desenvolvimento Mobile</p> <p>1.1 Navegação Intuitiva</p>	Verificação dos <u>check</u> conforme requisitos da situação de aprendizagem. Instrumento de avaliação:	Quadro Branco/ Projetor Multimídia, Computador com a IDE VSCode, Framework

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
	<p>1. Aplicar tecnologias de acordo com o sistema operacional</p> <p>2. Elaborar programação Mobile de acordo com os padrões de desenvolvimento.</p> <p>3. Criar interfaces gráficas da aplicação Mobile baseadas em UX.</p> <p>Capacidades Socioemocionais</p> <p>1. Demonstrar atenção a detalhes</p> <p>2. Demonstrar capacidade de tomar decisão</p> <p>3. Demonstrar organização</p> <p>4. Demonstrar pensamento criativo</p> <p>5. Demonstrar capacidade de solucionar problemas</p>	<p>1.2 Legibilidade</p> <p>1.3 Foco no Usuário</p> <p>2.2 Consumo</p> <p>2.1 De APIs</p> <p>3. Persistência de dados</p>	<p>Demonstração do aplicativo em funcionamento.</p> <p>Situação de aprendizagem com entrega no GitHub e avaliação conforme tabela de critérios críticos e desejáveis.</p>	Flutter, extensão Dart instalada, emulador Android, Firebase instalado . .
5	<p>Capacidades Técnicas</p> <p>4. Testar o código com ferramentas automatizadas (end to end) para</p>	<p>6. Publicação da Aplicação</p> <p>6.1 Preparação e testes</p>	<p>Exposição dialogada</p> <p>Mentimeter para realizar o Check out de revisão da</p>	Quadro Branco/ Projetor Multimídia, Computador com a IDE VSCode, Framework

Plano de Ensino

ESTRATÉGIAS DE ENSINO, APRENDIZAGEM E AVALIAÇÃO				
Nº horas / aulas	Capacidades a serem trabalhadas	Conhecimentos relacionados	Estratégias de ensino e aprendizagem e instrumentos de avaliação	Recursos e ambientes pedagógicos
	garantia de qualidade do software, prevendo eventuais falhas. 5. Documentar o sistema mobile para facilitar seu uso e funcionamento Capacidades Socioemocionais 1. Demonstrar atenção a detalhes 2. Demonstrar capacidade de tomar decisão 3. Demonstrar organização	6.2 Geração da aplicação de produção (build) 6.3 Técnica 6.4.1 Manual de uso 6.5 Hospedagem em lojas 6.5.1 APP Store 6.5.2 Google	disciplina de Desenvolvimento Mobile 3 Preparação e testes do aplicativo Demonstração do aplicativo desenvolvido. Instrumento de avaliação: Situação de aprendizagem com entrega no GitHub , demonstração do aplicativo em funcionamento, relatório de logs do aplicativo desenvolvido com base nos requisitos e avaliação conforme tabela de critérios críticos e desejáveis.	Flutter, extensão Dart instalada, emulador Android, Firebase instalado .

Referências (livros, apostilas, sites, blog etc.):

1. <https://api.flutter.dev/>
2. <https://docs.flutter.dev/get-started/install>
3. <https://docs.flutter.dev/ui/widgets>
4. <https://docs.flutter.dev/ui/widgets/material2>
5. <https://developer.android.com/studio/intro?hl=pt-br>

Capacidades técnicas

Capacidades Técnicas	Critérios de avaliação	Avaliação			
		Aluno	Professor		
1. Configurar o ambiente de desenvolvimento para aplicações Mobile	O ambiente de desenvolvimento VSCode , Flutter , Dart foi configurado corretamente gerando as pastas lib , build , arquivos main.dart e pubspec.yaml ?			4. 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?
	O aluno foi capaz de solucionar problemas comuns na configuração do ambiente de desenvolvimento como: erros de variáveis de ambiente?			5. Testar o código com ferramentas automatizadas (end to end) para garantia da qualidade do software, prevendo eventuais falhas.	O aluno foi capaz de criar o repositório Git para versionamento do código?
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?			6. Documentar o sistema mobile para facilitar o seu uso e funcionamento	O aluno foi capaz de realizar testes no código e documentar os logs durante os testes?
	O aluno aplicou os princípios de UX para cria uma interface gráfica intuitiva e amigável?				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?
3. Elaborar programação Mobile de acordo com os padrões de desenvolvimento	O aluno foi capaz de elaborar o software legível, estruturado e limpo para consumir a API?				O aluno foi capaz de criar diagramas e fluxos para ilustrar visualmente o funcionamento do sistema mobile?
	O aluno foi capaz de desenvolver o código de forma organizada e modularizada?				

Capacidades Socioemocionais (SoftSkills)

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).		
	O aluno foi capaz de revisar e testar o aplicativo antes da entrega e documentou os testes e saídas do aplicativo?		
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?		
	O aluno foi capaz de comunicar efetivamente suas decisões de forma clara e persuasiva?		
3. Demonstrar organização	O aluno entregou o projeto conforme cronograma proposto?		
	O aluno foi capaz de dividir as tarefas complexas em subtarefas de menor complexidade?		

Capacidades Socioemocionais (SoftSkills)

4. Demonstrar pensamento criativo	O aluno foi capaz de criar uma solução original e única para o projeto proposto?		
	O aluno foi capaz de transformar suas ideias em um projeto viável e executável?		
5. Capacidade de solucionar problemas	O aluno foi capaz de resolver os problemas do aplicativo utilizando debug e teste?		
	O aluno foi capaz de gerenciar o tempo e recursos disponíveis para a resolução do problema?		

Níveis de desempenho

Página 5 de 15

TABELA DE NÍVEIS DE DESEMPENHO		Nível de desempenho	Conversão em notas
Critérios de Avaliação			
Atingiu todos os critérios críticos e todos desejáveis		22	100
Atingiu todos os critérios críticos e 10 dos desejáveis		21	95
Atingiu todos os critérios críticos e 9 dos desejáveis		20	90
Atingiu todos os critérios críticos e 7 dos desejáveis		18	85
Atingiu todos os critérios críticos e 6 dos desejáveis		17	80
Atingiu todos os critérios críticos e 5 dos desejáveis		16	75
Atingiu todos os critérios críticos e 3 dos desejáveis		14	70
Atingiu todos os critérios críticos e 2 dos desejáveis		13	60
Atingiu todos os critérios críticos e nenhum desejável		11	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

NÍVEL MÍNIMO DE DESEMPENHO ESPERADO | 11 |

Situação de aprendizagem



Você foi contratado pela empresa Mange Eats para desenvolver um aplicativo responsivo para entrega de comida

Este aplicativo deve possuir os seguintes requisitos:

A) Tela inicial

Login e senha realizando a autenticação por meio do usuário cadastrado no banco de dados Firebase do aplicativo

B) Tela Cadastro

Nesta tela o cliente irá criar seu usuário e senha para logar no aplicativo

C) Tela Cardápio

Nesta tela o aplicativo irá exibir todas as opções de comida disponível

Criação da tela de menu de comida onde os usuários podem escolher entre lanches e pizzas.;

Situação de aprendizagem



D) Tela Carrinho

Implementação do carrinho de compras para adicionar e remover itens

E) Confirmação de Pedido: Tela para confirmar o pedido e visualizar o total da compra e exibir o endereço de entrega obtido através da geocalização

F) Desenvolver a interface gráfica seguindo os princípios de UX, o código, realizar o versionamento do aplicativo através do sistema Git e realizar o upload do código no repositório GitHub.

Metodologias das aulas

Aulas teóricas, expositivas, exercícios e projetos

Aulas de Terça - Turma B e Quinta - Turma A

Horário : 18:45 - 23:10

Intervalo: 21:15 - 21:30

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

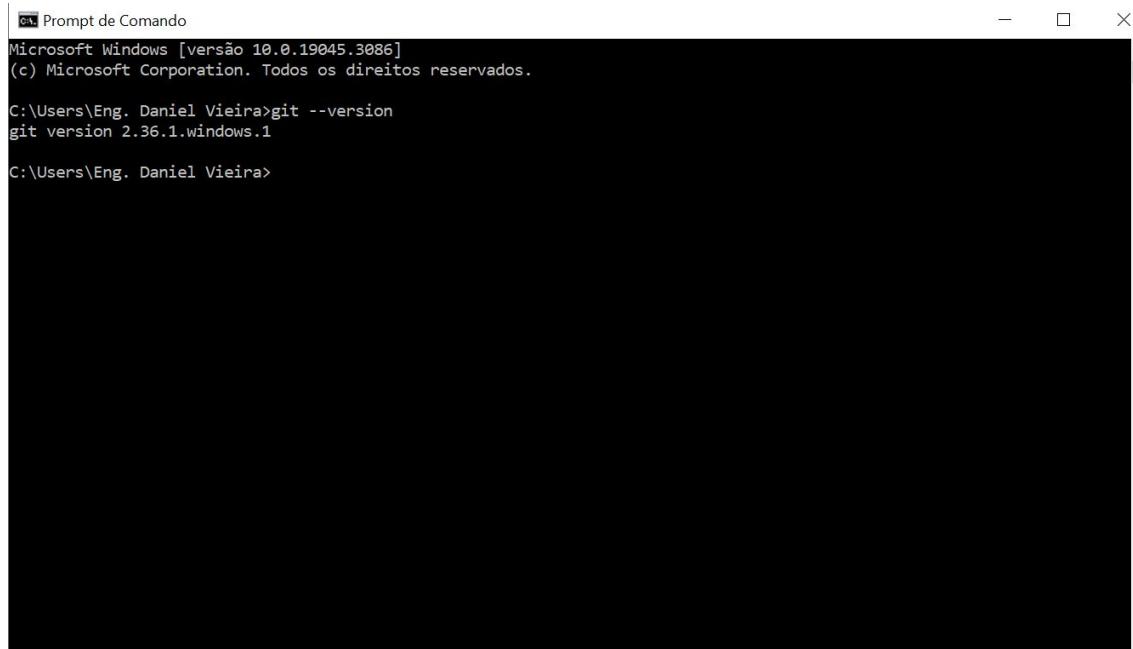
Git instalação

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 links for About, Documentation, Downloads (which is highlighted in red), and Community. A sidebar on the left promotes the "Pro Git book" by Scott Chacon and Ben Straub. The central content area is titled "Download for Windows". It features a prominent call-to-action button: "Click here to download the latest (2.41.0) 64-bit version of Git for Windows". Below this, there's a brief description of the latest release and a "Search entire site..." input field. Further down, there are sections for "Other Git for Windows downloads" (including Standalone Installer, 32-bit and 64-bit Setup options, and Portable editions), instructions for using the winget tool, and information about the current source code release (version 2.41.0). At the bottom, there's a "Now What?" section with a link to start using Git.

git --fast-version-control

About

Documentation

Downloads

GUI Clients

Logos

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.

Search entire site...

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](#)'. On the right side of the page, there is another 'Set up an editor' link with GitHub and VS Code icons.

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, 3. Test drive, 4. Write your first app, and 5. Learn more. Below this, there are 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'.

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 for Windows 2.x), provides a link to the 'Flutter SDK' (highlighted in blue), and includes a warning about installing to paths with special characters. The right sidebar contains links for 'Contents', 'System requirements', 'Get the Flutter SDK', 'Update your path', 'Run flutter doctor', 'Android setup', 'Windows setup', and a 'Next step' button.

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.

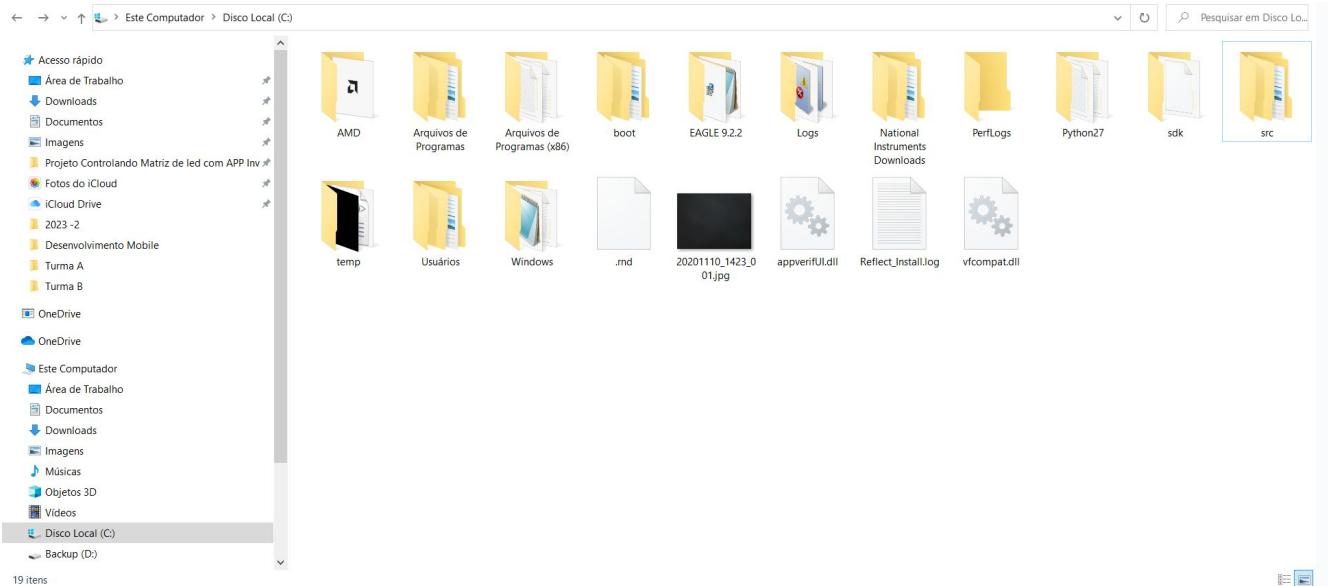
Instalação do Flutter

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.

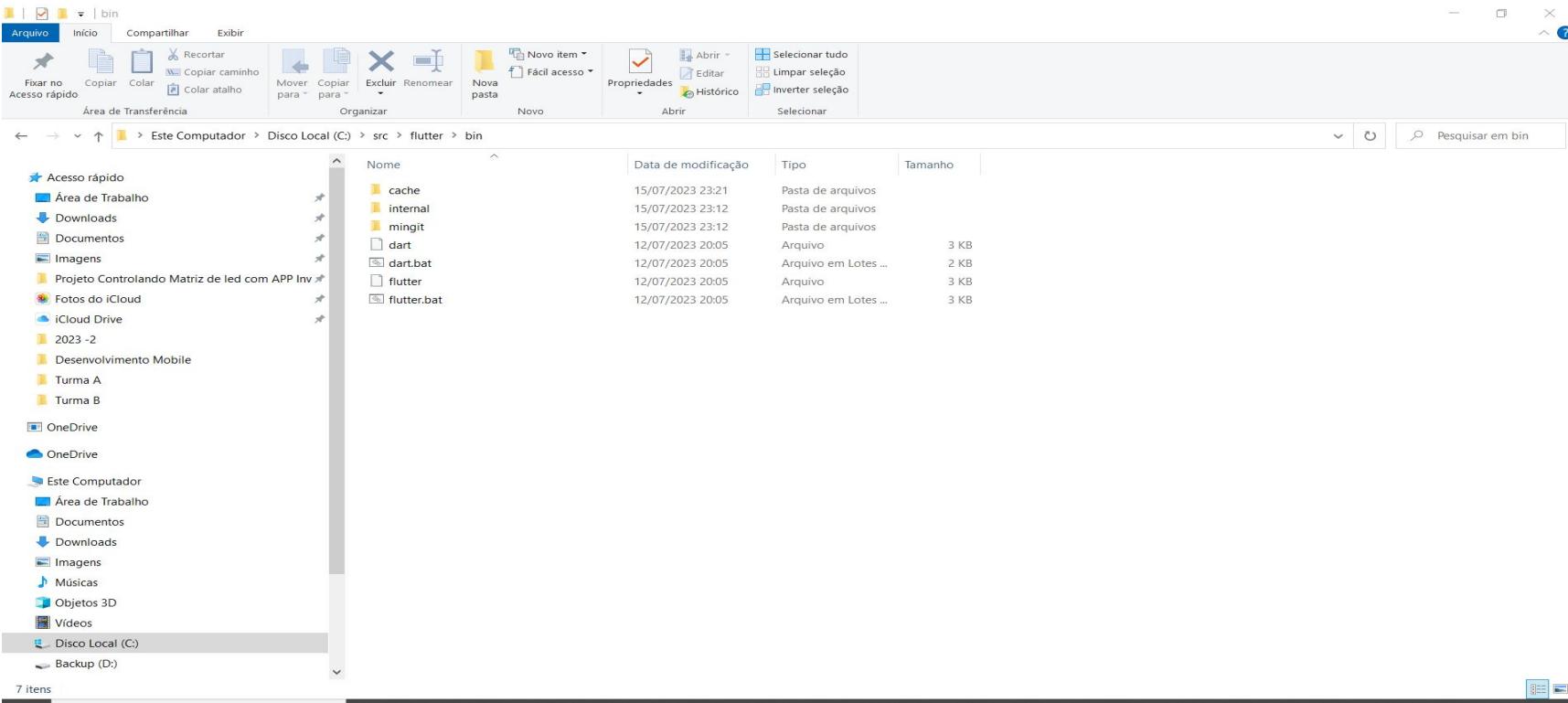
5º Passo: Copiar a pasta flutter para a pasta src



Instalação do Flutter

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

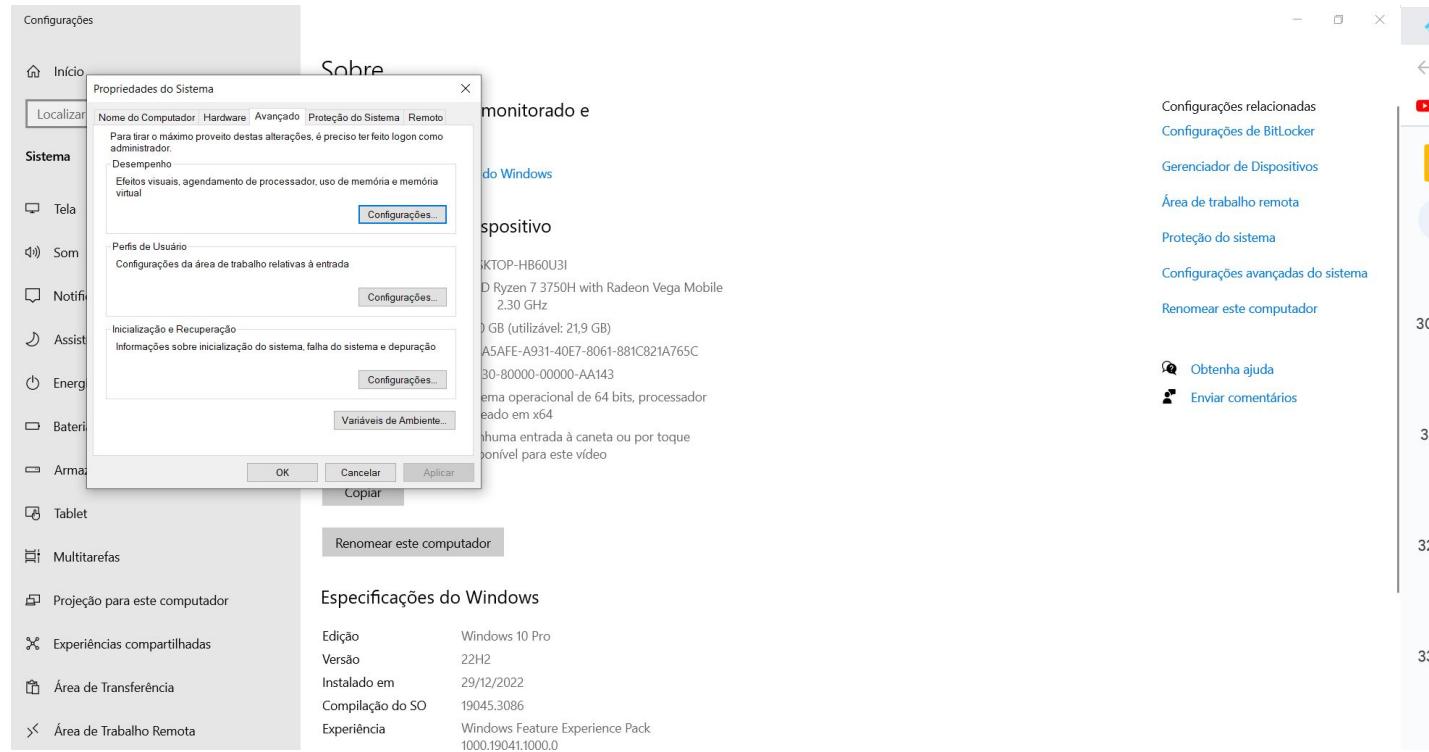
C:\src\flutter\bin



Instalação do Flutter

7º Passo : Configurar variáveis de ambiente

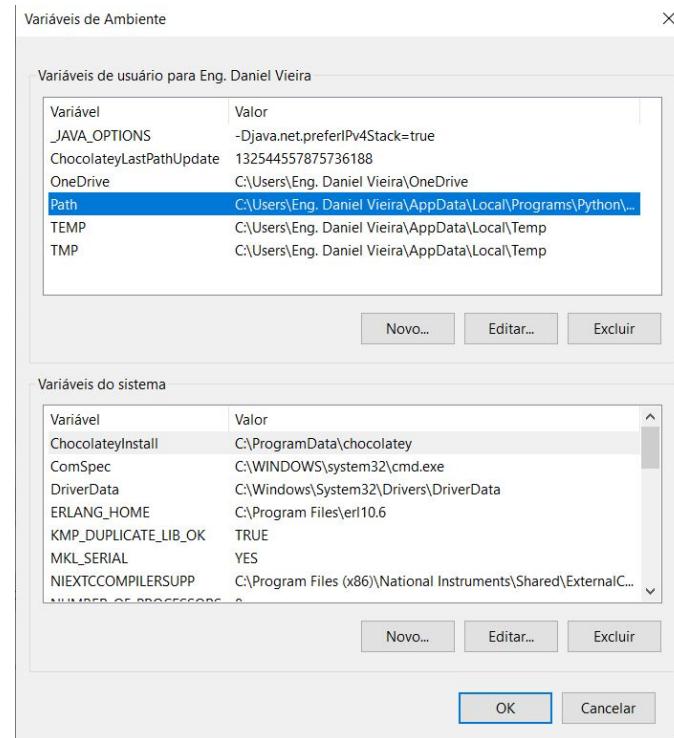
Meu Computador -> Propriedades -> Configurações avançadas do sistema



Instalação do Flutter

8º Passo : Configurar variáveis de ambiente

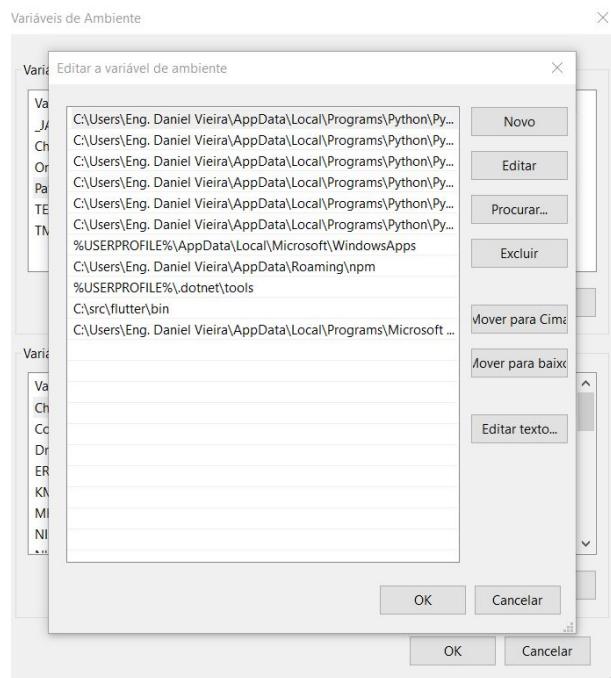
Clicar em variáveis de ambiente -> Procurar pelo campo Path e clicar em editar



Instalação do Flutter

9º Passo : Configurar variáveis de ambiente

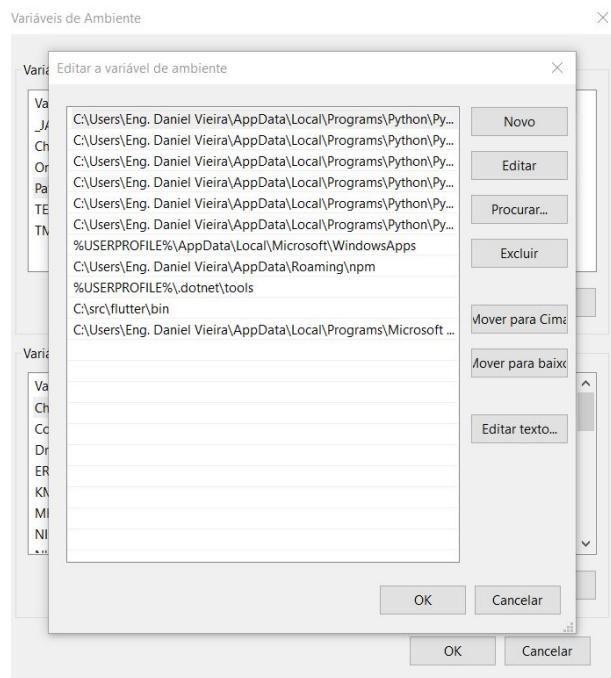
Procurar pelo campo Path e clicar em editar-> Novo e colar o caminho copiado anteriormente
C:\src\flutter\bin



Instalação do Flutter

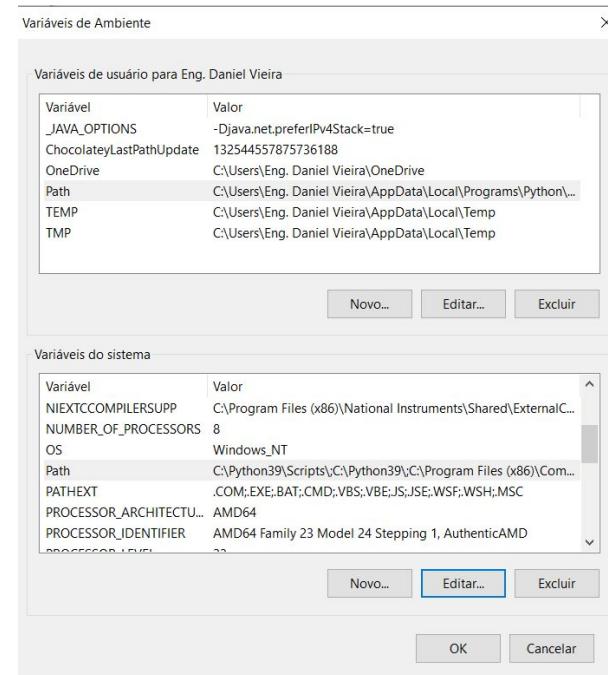
9º Passo : Configurar variáveis de ambiente

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

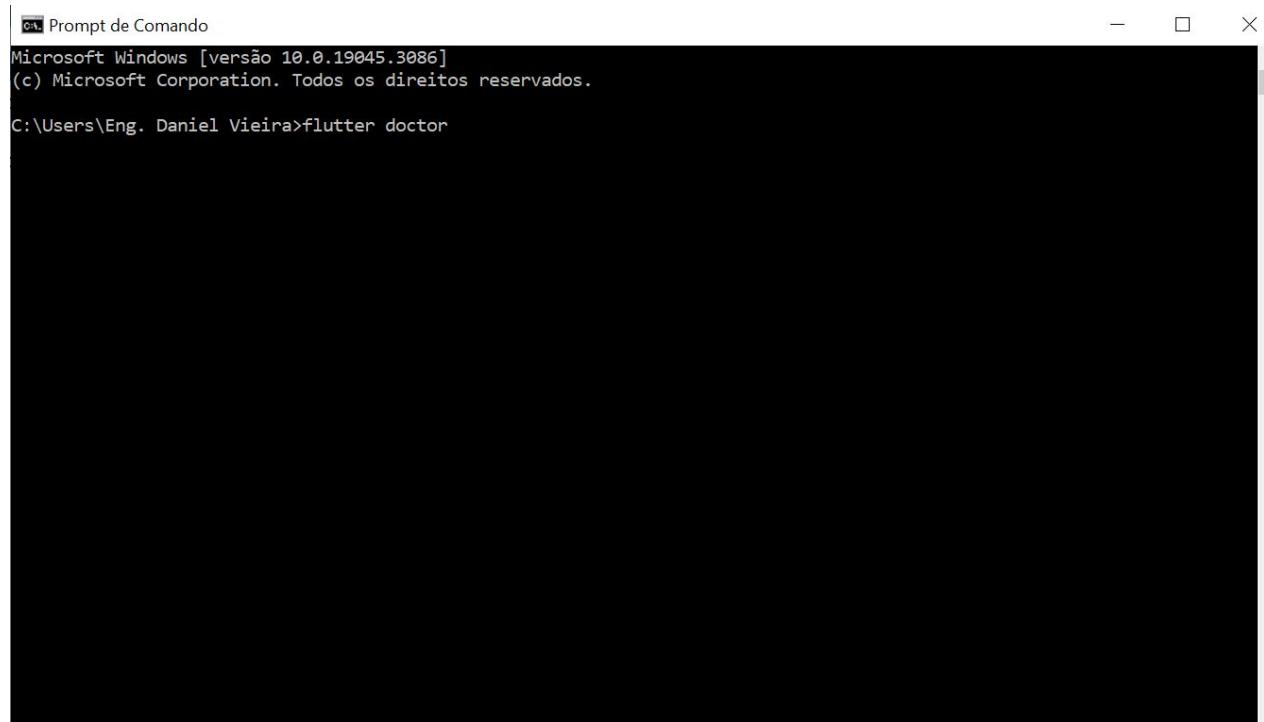


Instalação do Flutter

10º Passo

Menu iniciar -> cmd

11º Passo digitar flutter doctor para verificar os componentes instalados

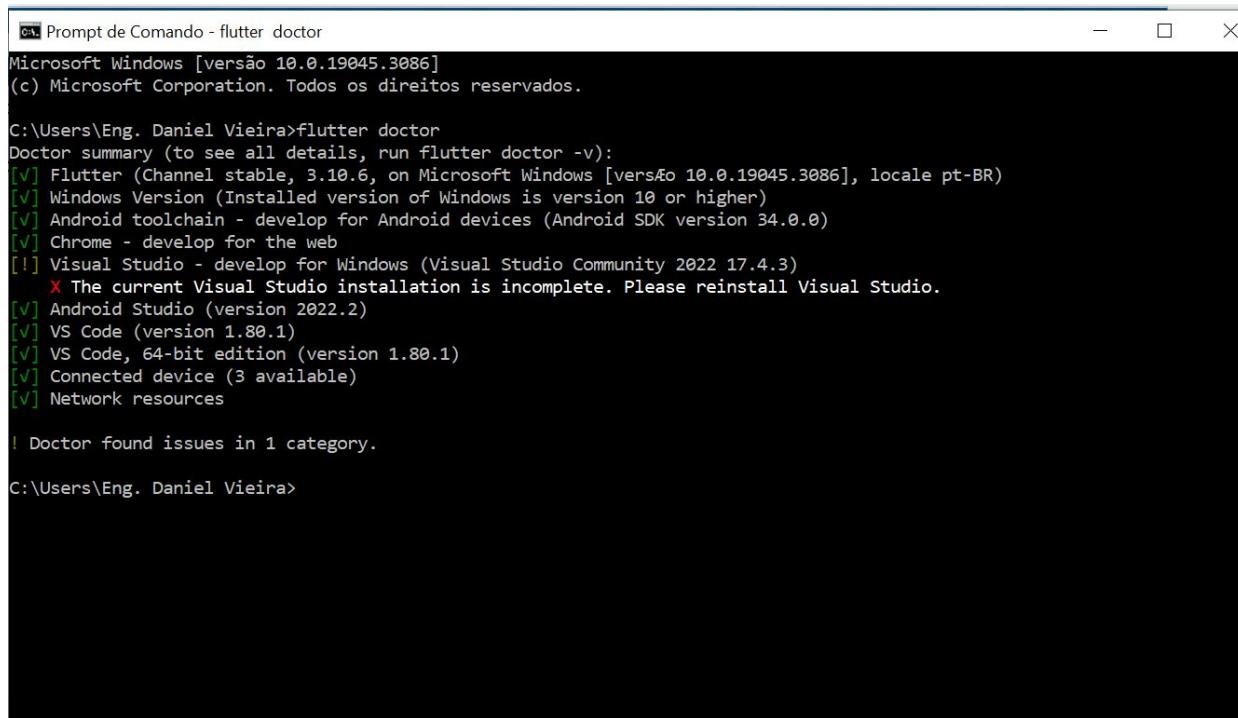


```
Prompt de Comando
Microsoft Windows [versão 10.0.19045.3086]
(c) Microsoft Corporation. Todos os direitos reservados.

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. At the top, there's a navigation bar with links for 'developers' (with a logo), 'Essentials', 'Design & Plan', 'Docs', 'Google Play', 'Search' (with a magnifying glass icon), 'Português - ...' (language dropdown), 'Android Studio' (selected), and 'Fazer login' (login). Below the navigation, the page title 'ANDROID STUDIO' is displayed. A horizontal menu bar includes 'Download' (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 downward arrow icon. To the right, the actual Android Studio IDE is shown. It displays a Java code snippet for a Composable function named 'TopicSelection'. The code uses 'rememberLazyGridState' and 'LazyHorizontalGrid' to create a scrollable grid of items. On the right side of the IDE, there's a 'Layout Inspector' tool showing a wireframe of a mobile device screen with various UI components like cards and icons. At the bottom of the IDE window, there's a 'App Quality Insights' section with a chart showing issues over time across different devices and operating systems.

developers

Essentials ▾ Design & Plan ▾ Docs Google Play

Search

Português - ...

Android Studio Fazer login

ANDROID STUDIO

Download Android Studio editor Android Gradle Plugin SDK tools Preview

Android Studio

Get the official Integrated Development Environment (IDE) for Android app development.

Download Android Studio Flamingo

Read release notes

```
rowinandroid - 10 main - ForYouScreen.kt
```

```
@Composable
private fun TopicSelection(
    onBoardingUIState: OnboardingUIState,
    onTopicCheckedChanged: (String, Boolean) -> Unit,
    modifier: Modifier = Modifier
) = tree(sectionsName = "TopicSelection") {
    val lazyGridState = rememberLazyGridState()
    val topicSelectionTestTag = "ForYou:topicSelection"
    TrackScrollViewLazyGridState(lazyGridState, stateName = topicSelectionTestTag)

    LazyHorizontalGrid(
        state = lazyGridState,
        rows = GridCells.Fixed(count = 3),
        horizontalArrangement = Arrangement.spacedBy(12.dp),
        verticalArrangement = Arrangement.spacedBy(12.dp),
        contentPadding = PaddingValues(16.dp),
        modifier = modifier
            .heightIn(max(240.dp, with(LocalDensity.current) { 240.sp.toDp() }))
            .fillMaxWidth()
            .testTag(topicSelectionTestTag)
    )
}
```

App Quality Insights Google Firebase Crashlytics

Issue	Last 60 days	Events	Users	Details
!hasExtras()	24	7	Google 29%	
!app:nowInAndroid/u0a	12	4	Other 8%	
!@ed system_server and system app only, unless they are annotated with @Readable.	10	3	Android Versions	
!long IndianaOutOfMemoryException - Index: 4, size: 4	7	1	Android (12) 58%	
!java.SQLiteDatabaseException - Database or disk is full (code 13 SQLITE_FULL)	1	1	Android (13) 25%	
!long IndianaOutOfMemoryException - consensus	1	1	Android (9) 8%	
!long IndianaOutOfMemoryException - consensus	1	1	Other 8%	

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', 'Google Play', 'Search' (with a magnifying glass icon), 'Português - ...' (language dropdown), 'Android Studio' (selected), and 'Fazer login' (login). Below the navigation, the page title 'ANDROID STUDIO' is displayed. A horizontal menu bar includes 'Download' (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 downward arrow icon. To the right, the actual Android Studio IDE is shown. It displays a Java code snippet for a Composable function named 'TopicSelection'. The code uses 'rememberLazyGridState' and 'LazyHorizontalGrid' to create a scrollable grid of items. On the right side of the IDE, there's a 'Layout Inspector' tool showing a wireframe of a mobile device screen with various UI components like cards and icons. At the bottom of the IDE window, there's a 'App Quality Insights' section with a chart showing issues over time across different devices and operating systems.

developers

Essentials ▾ Design & Plan ▾ Docs Google Play

Search

Português - ...

Android Studio Fazer login

ANDROID STUDIO

Download Android Studio editor Android Gradle Plugin SDK tools Preview

Android Studio

Get the official Integrated Development Environment (IDE) for Android app development.

Download Android Studio Flamingo

Read release notes

```
rowinandroid - 10 main - ForYouScreen.kt
```

```
@Composable
private fun TopicSelection(
    onBoardingUIState: OnboardingUIState,
    onTopicCheckedChanged: (String, Boolean) -> Unit,
    modifier: Modifier = Modifier
) = tree(sectionsName = "TopicSelection") {
    val lazyGridState = rememberLazyGridState()
    val topicSelectionTestTag = "ForYou:topicSelection"
    TrackScrollViewLazyGridState(lazyGridState, stateName = topicSelectionTestTag)

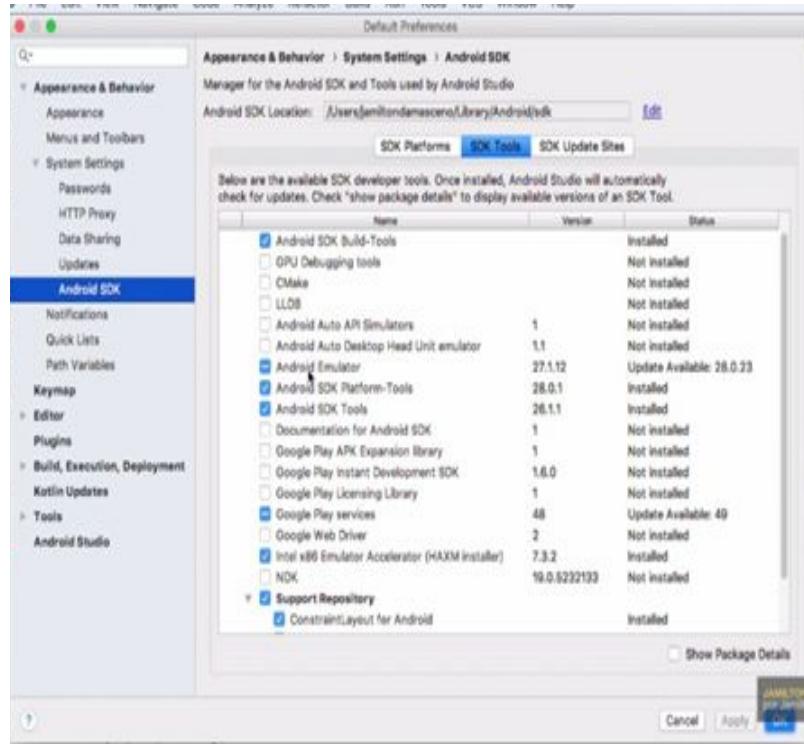
    LazyHorizontalGrid(
        state = lazyGridState,
        rows = GridCells.Fixed(count = 3),
        horizontalArrangement = Arrangement.spacedBy(12.dp),
        verticalArrangement = Arrangement.spacedBy(12.dp),
        contentPadding = PaddingValues(16.dp),
        modifier = modifier
            .heightIn(max(240.dp, with(LocalDensity.current) { 240.sp.toDp() }))
            .fillMaxWidth()
            .testTag(topicSelectionTestTag)
    )
}
```

App Quality Insights Google Firebase Crashlytics

Issue	Last 60 days	Events	Users	Details
! has extras	24	7	Google 29% Other 8%	
! apps.newinandroid/u0a	12	4	Other 8%	
! add system_server and system app only, unless they are annotated with @Readable.	10	3	Android Versions	
! long IndianaOrphansException - Index 4, size: 4	7	1	Android (12) 58% Android (13) 25% Android (9) 8% Other 8%	
! java.sql.SQLException - Database or disk is full (code 13 SQLITE_FULL)	1	1		
! long IndianaOrphansException - consensus	1	1		

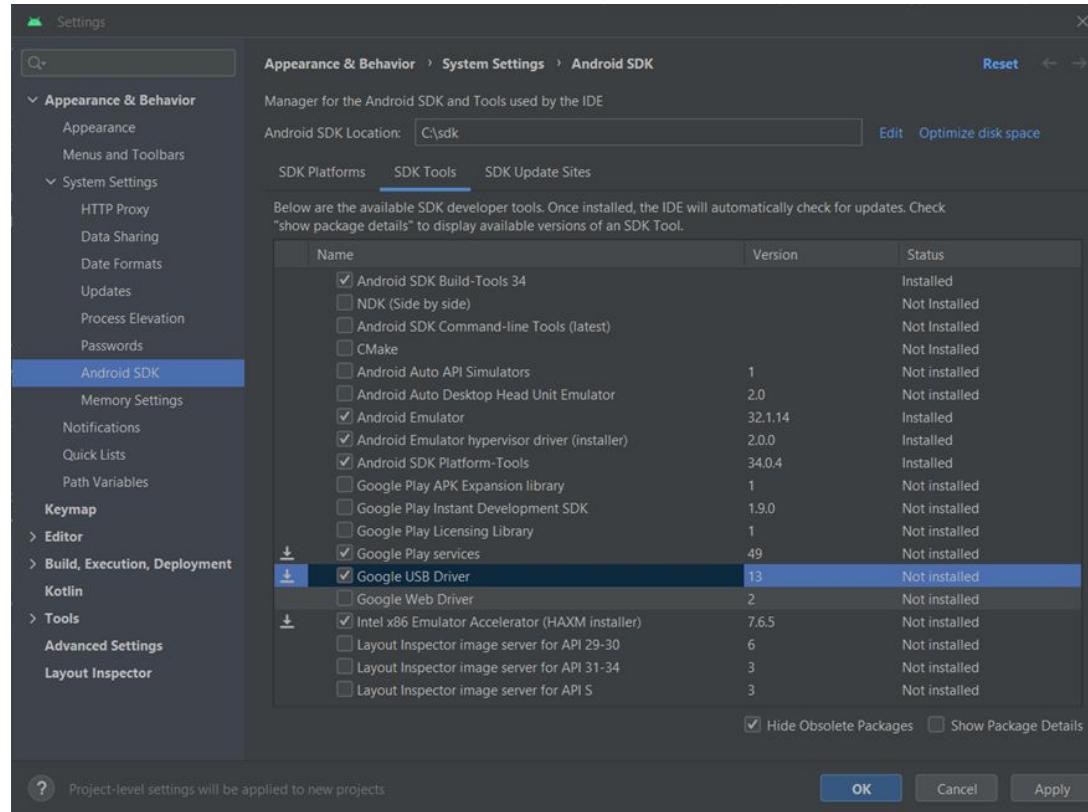
Instalação do SDK Android

2º Instalar o Android Studio



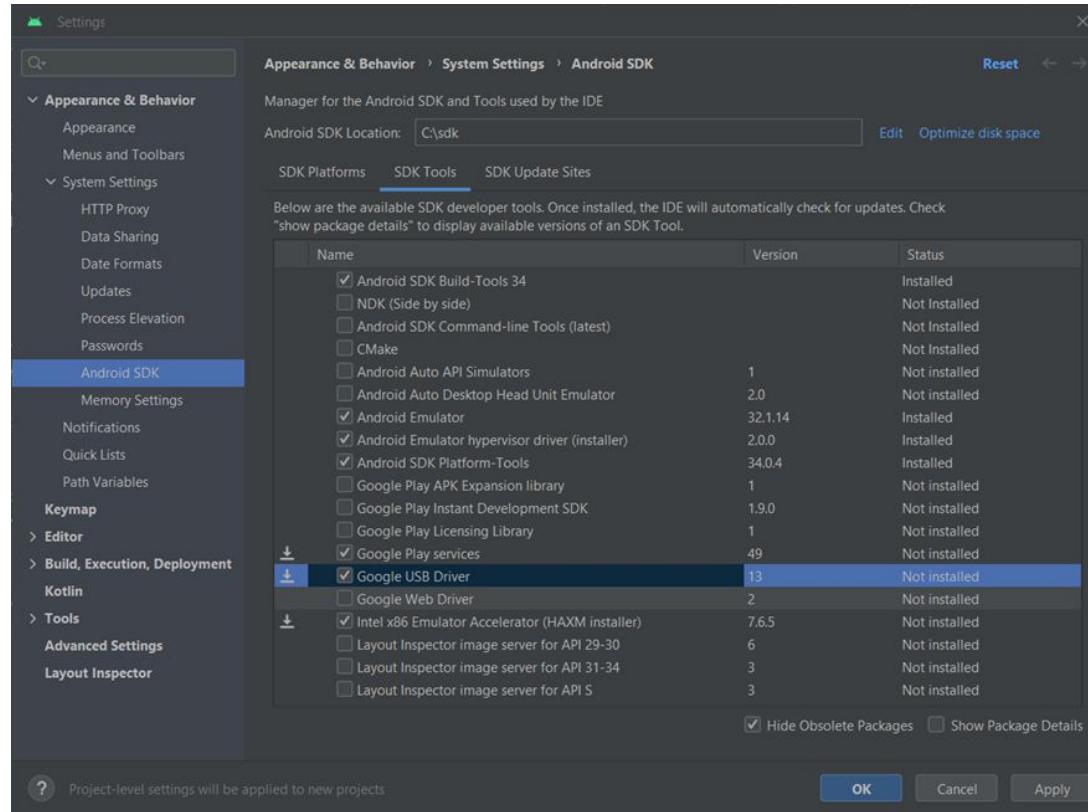
Instalação do SDK Android

2º Instalar o SDK



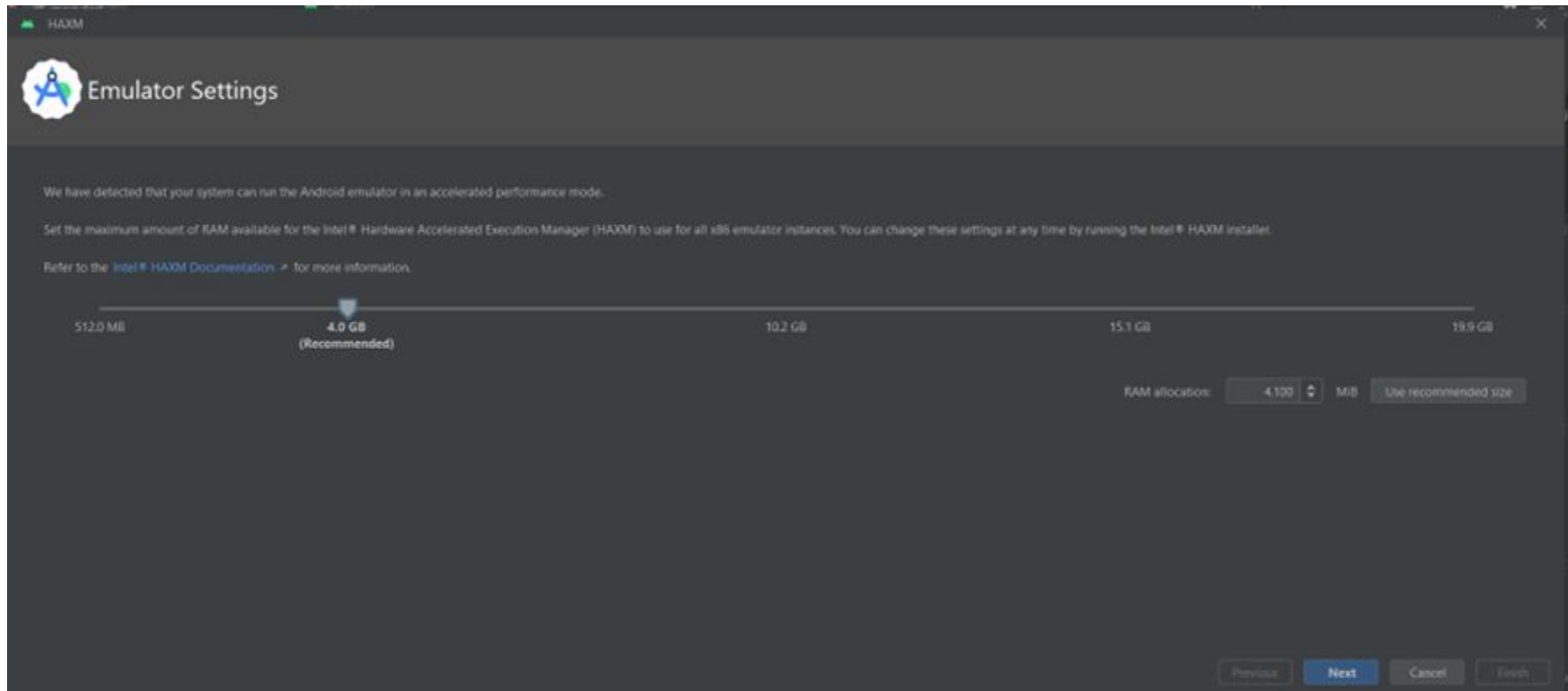
Instalação do SDK Android

2º Instalar o SDK



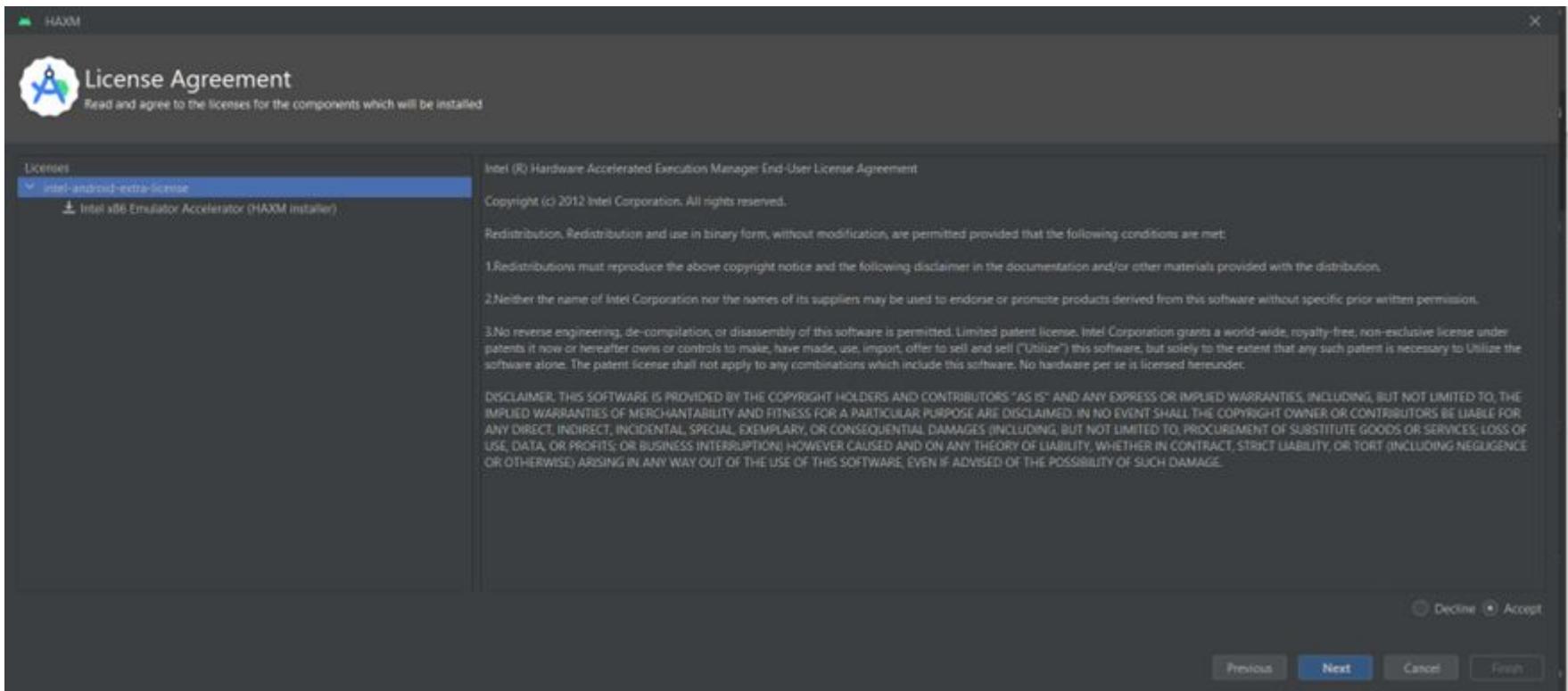
Instalação do SDK Android

3º Instalar o Emulador



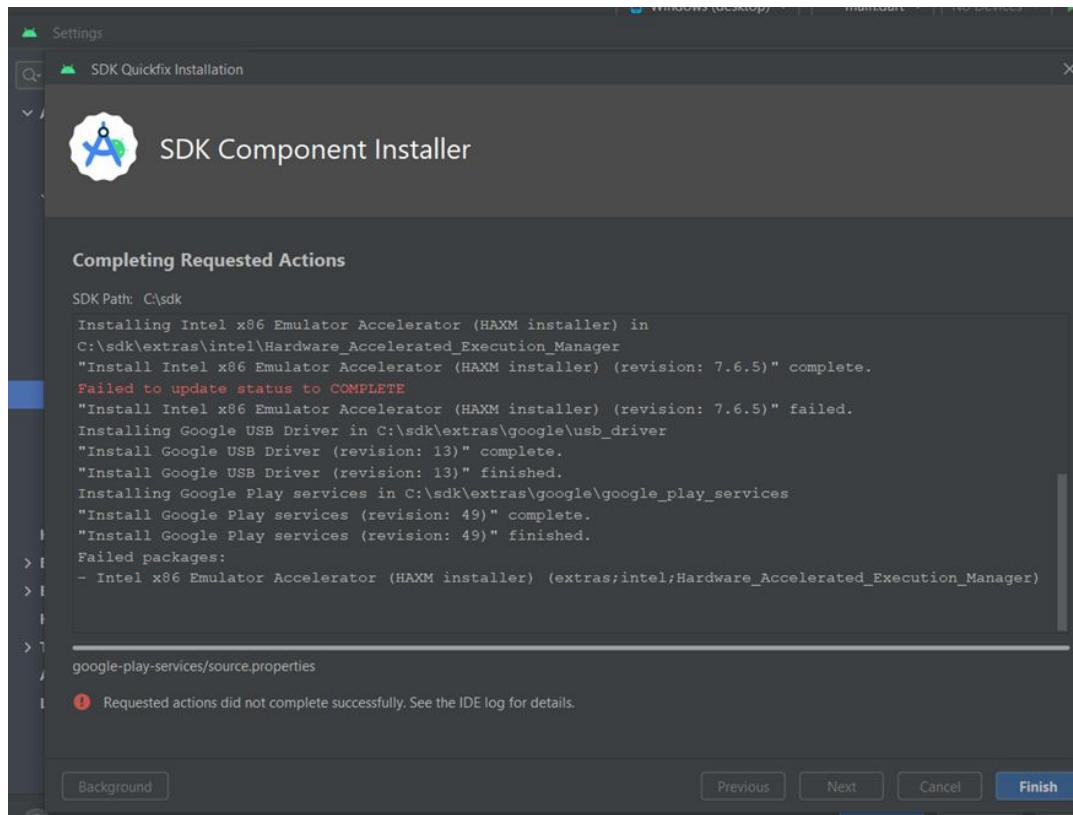
Instalação do SDK Android

4º Concordar com a licença



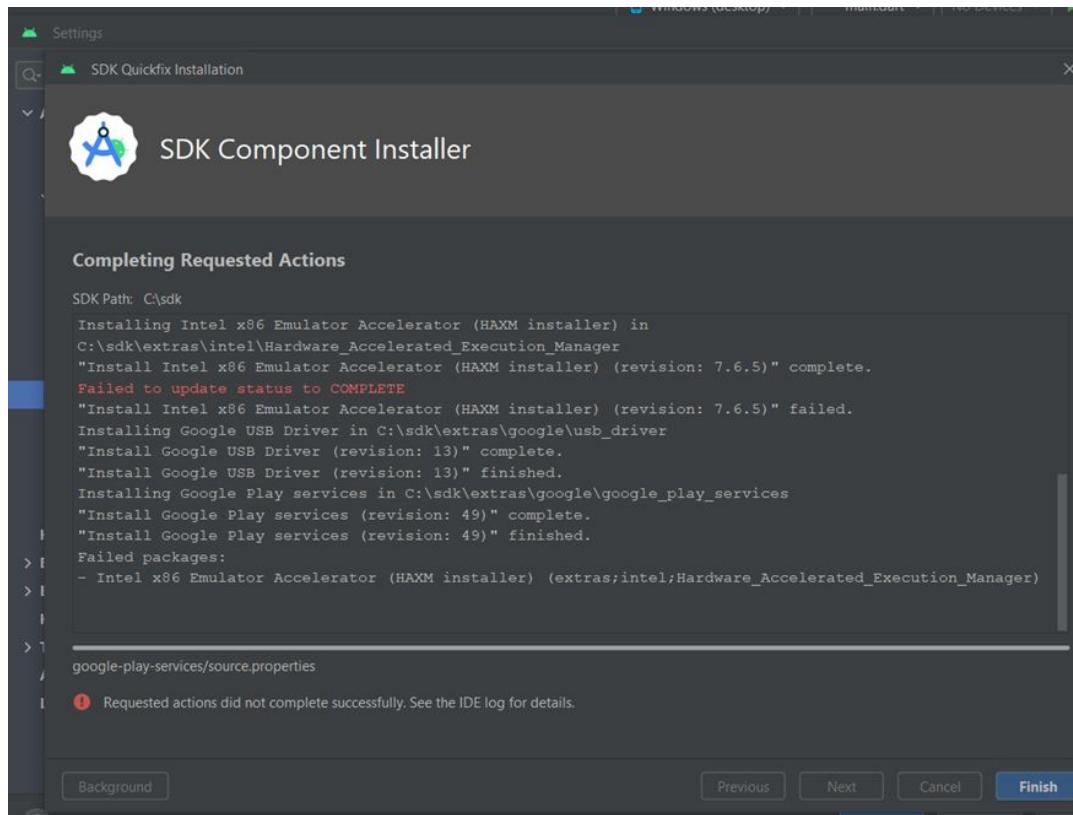
Instalação do SDK Android

5º Instalação do Emulador



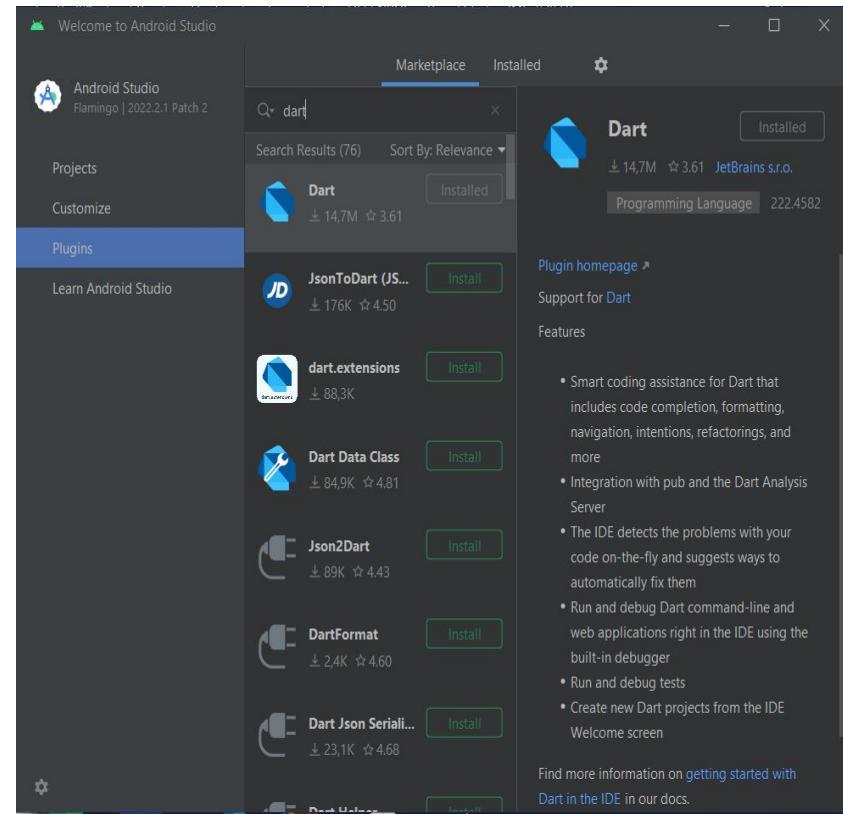
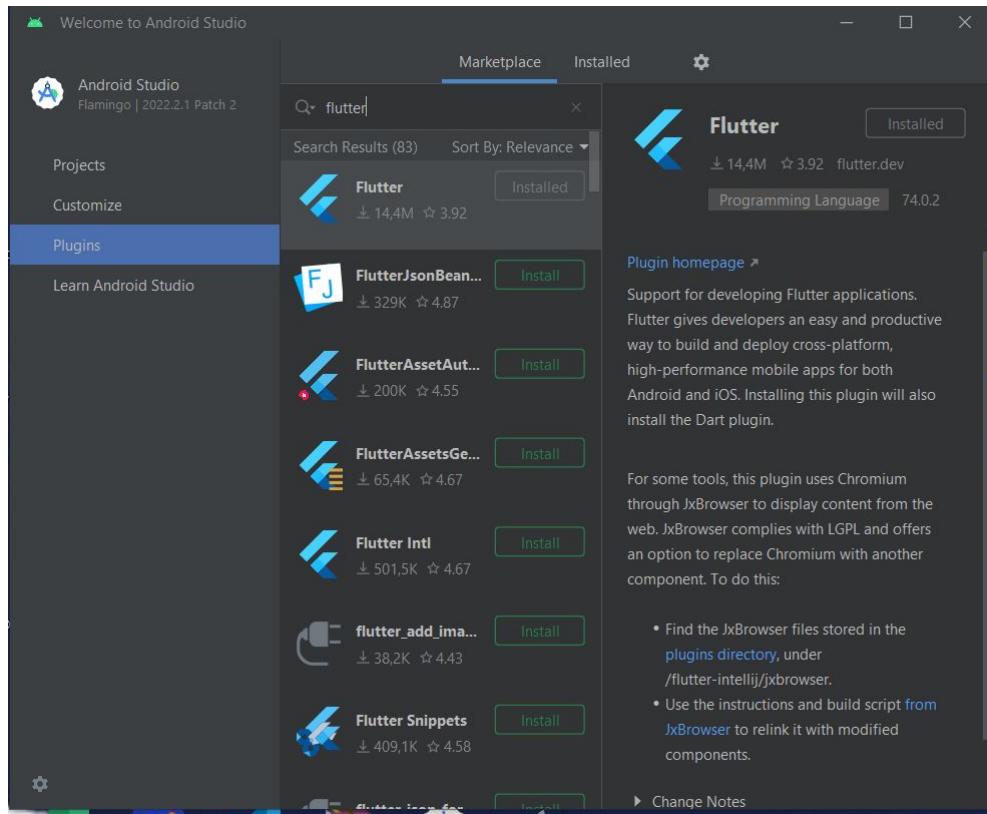
Instalação do SDK Android

5º Instalação do Emulador



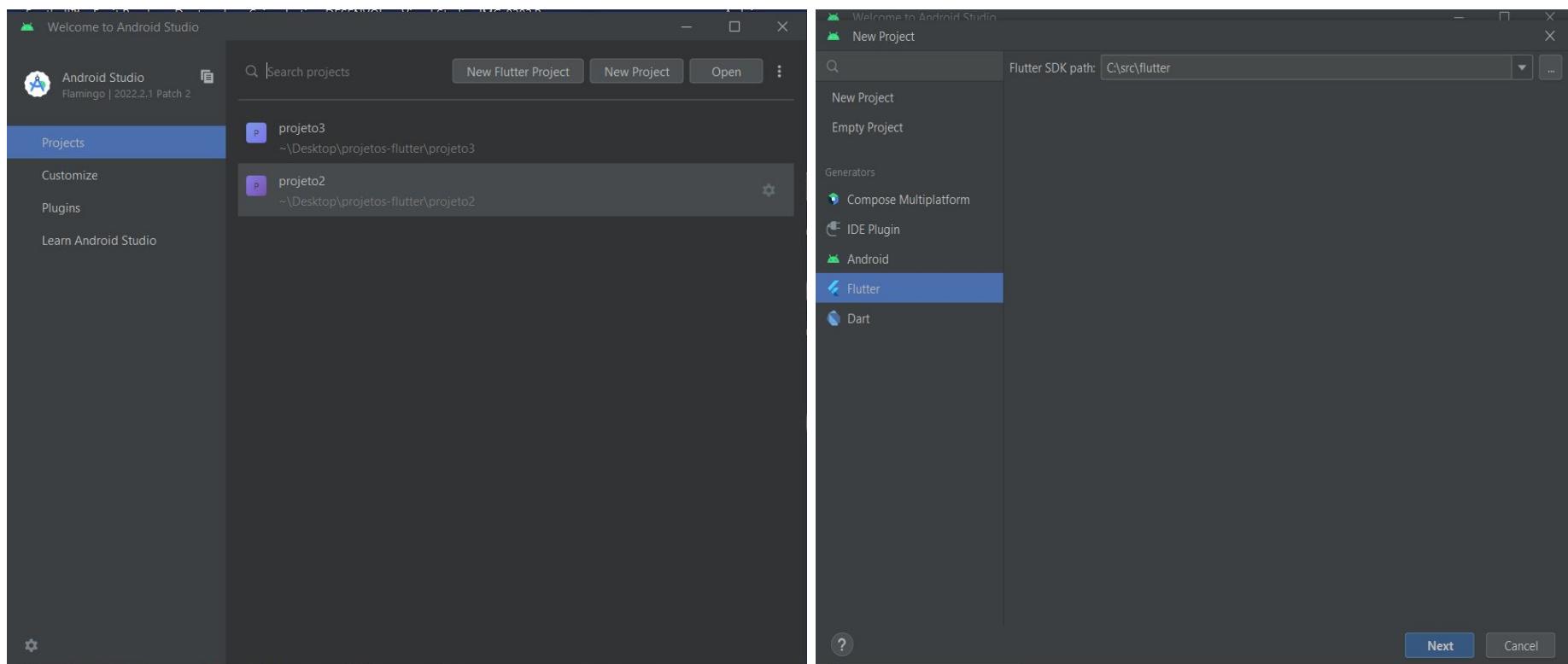
Instalação do SDK Android

6º Instalação dos plugins Flutter e Dart



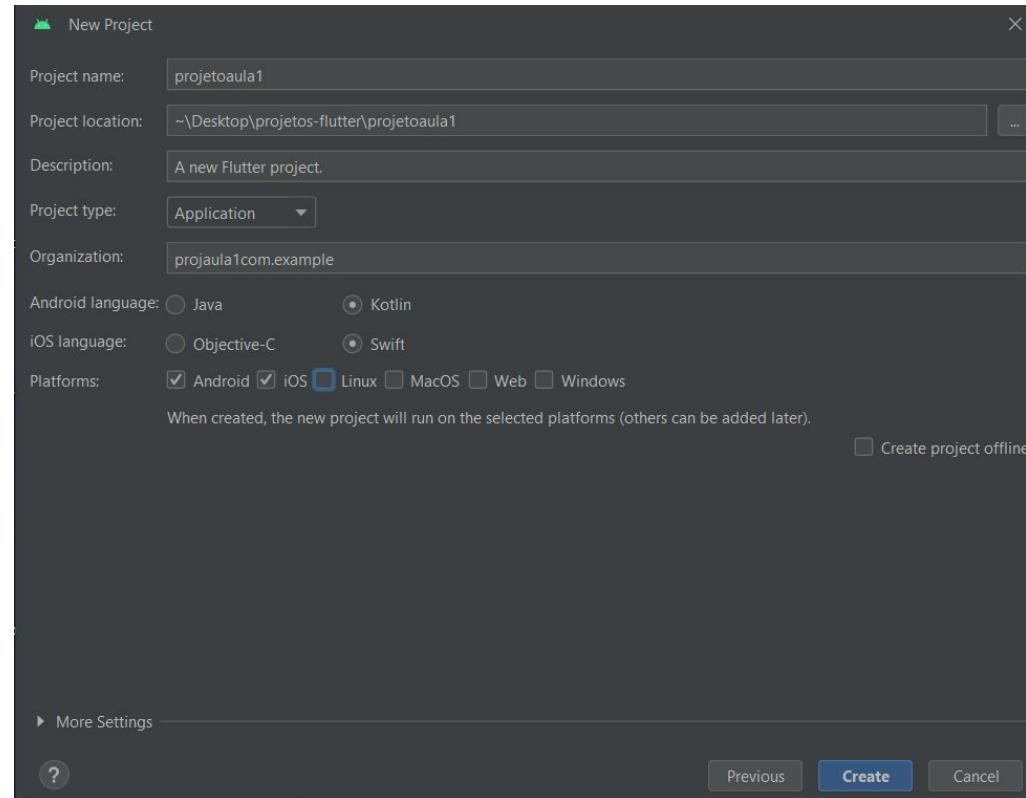
Instalação do SDK Android

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



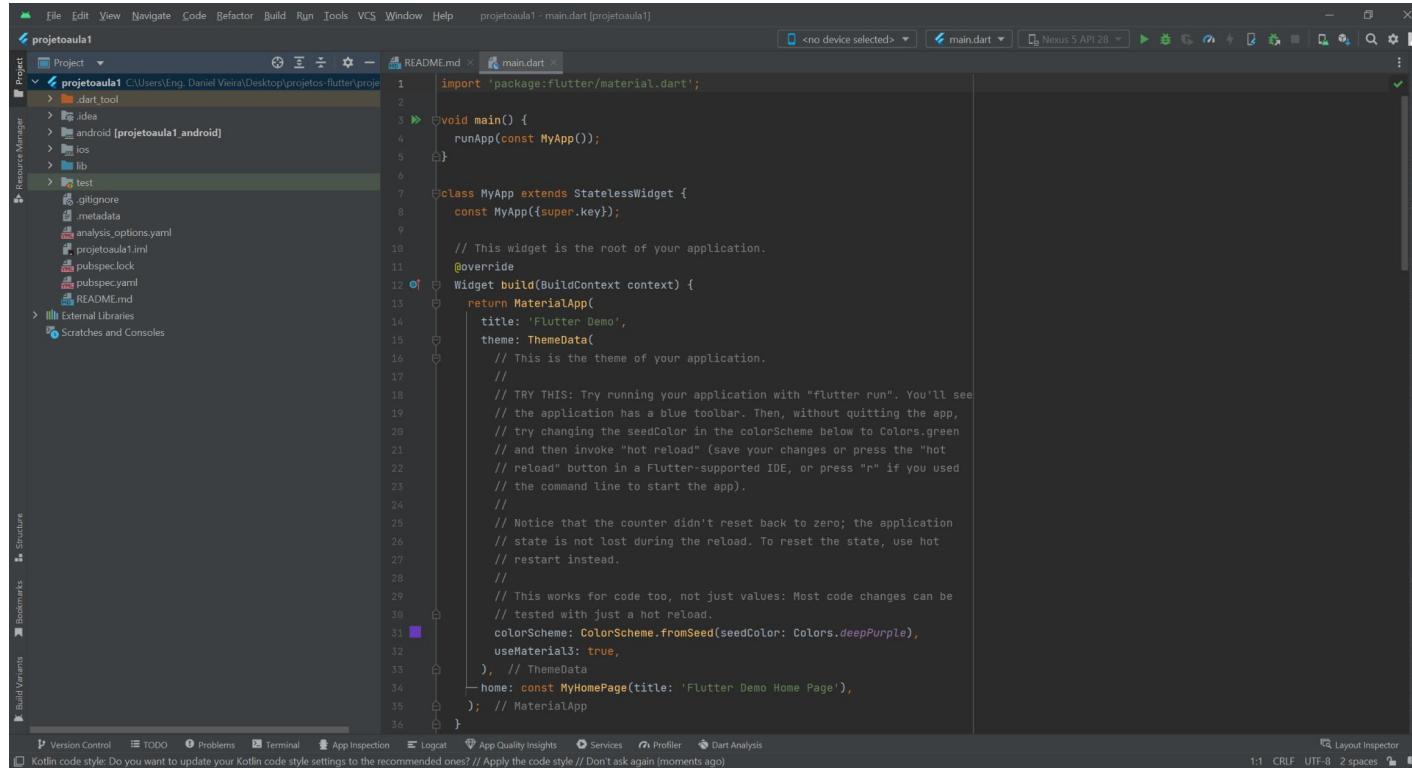
Instalação do SDK Android

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



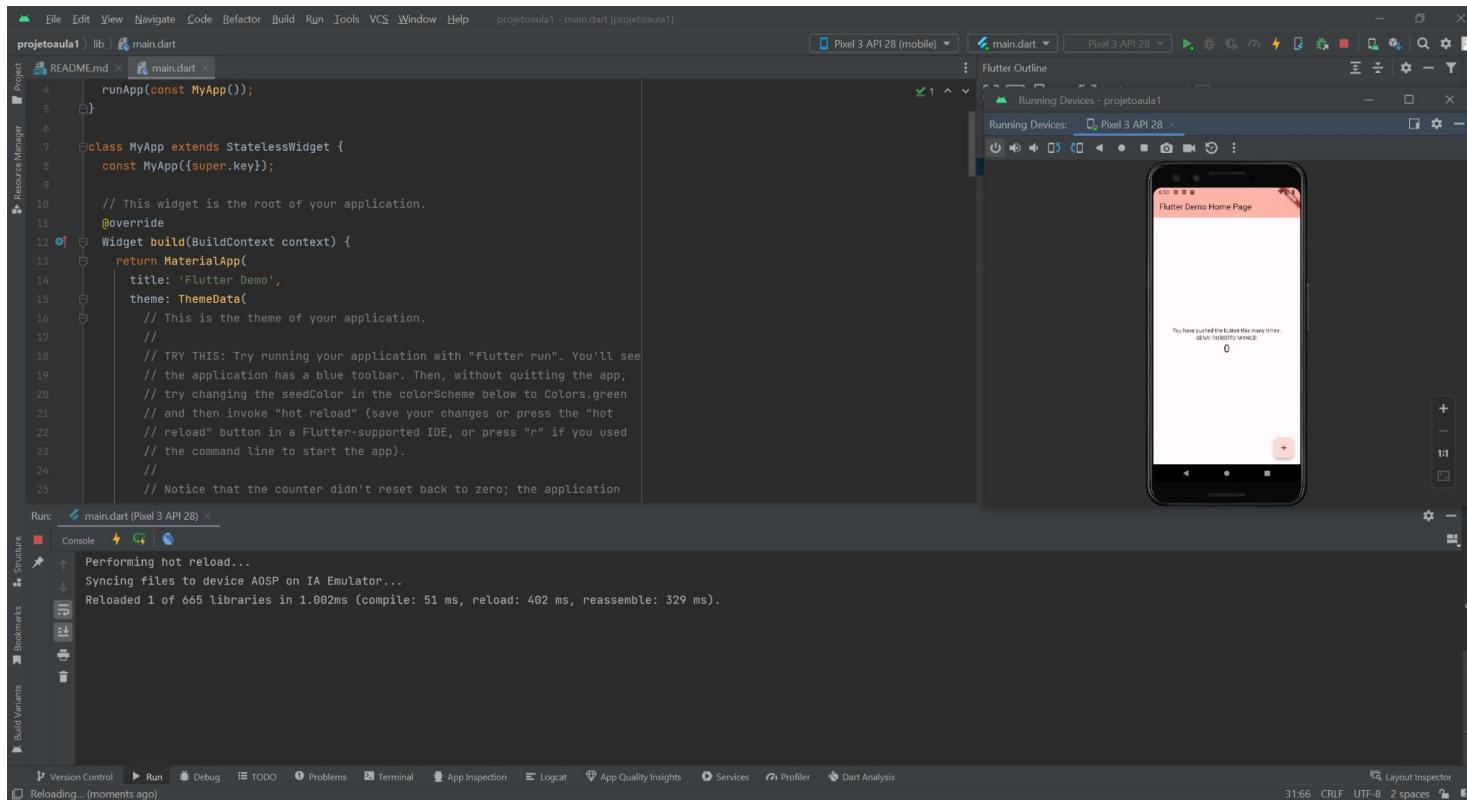
Instalação do SDK Android

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



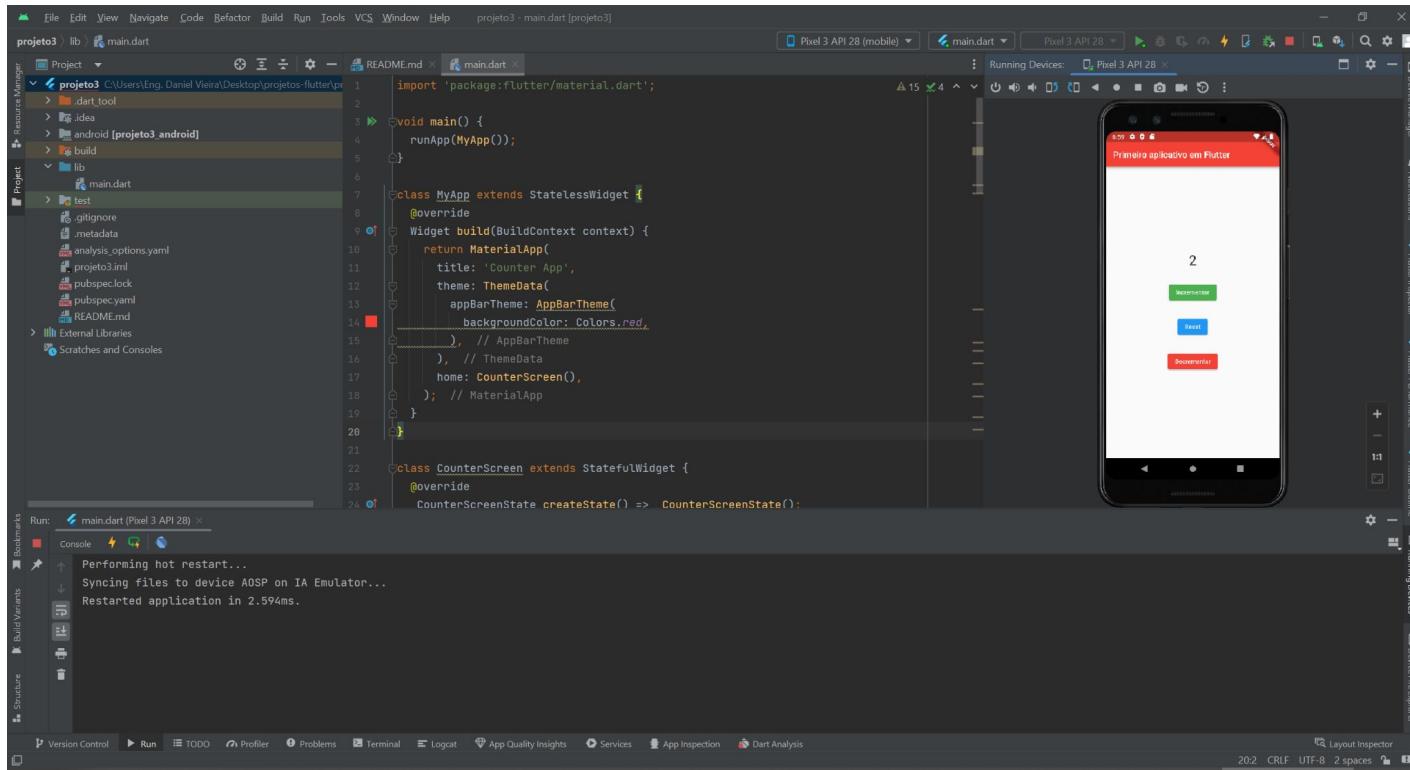
Instalação do SDK Android

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

