



Capacitação em Inteligência Artificial e Aplicações

Integração Tensorflow com Flutter

- Prof. Gerson Vieira Albuquerque Neto
- Prof. Rodrigo Carvalho Souza Costa
- Prof. Yves Augusto Romero



UNIVERSIDADE
ESTADUAL DO CEARÁ



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO





IA

Planejamento da Disciplina

D	S	T	Q	Q	S	S
26	27 Introdução ao curso	28 Áreas e aplicações de IA	29 Tipos e definições de Inteligência artificial	30 Revisão de álgebra e probabilidade	31 Laboratório Python 1	1
2	3 Introdução aos classificadores supervisionados	4 Aula teórica Naive Bayes	5 Aula prática Naive Bayes	6 Feriado Semana Santa	7 Feriado Semana Santa	8
9	10 KNN + Métricas de Avaliação	11 Regressão Linear e Introdução à árvores de decisão	12 Prática Regressão Linear + Árvores de Decisão	13 Feriado	14 Introdução à Clusterização + KMédias	15
16	17 Falta de Energia Campus Fortaleza	18 PCA / Hiperparâmetros	19 Introdução ao Perceptron Simples – Prática	20 MLP	21 Feriado Tiradentes	28
23	24 Introdução ao DeepLearning	25 Introdução ao TensorFlow / Keras	26 Introdução ao Pytorch	27 Tensorflow for android	28	29



IA

Objetivos da Aula

- Após a conclusão deste módulo, você será capaz de:
 - Compreender como realizar a integração do Flutter com o Tensorflow para criação de aplicações de inteligência artificial em dispositivos portáteis





IA

Flutter

- Flutter é um framework de desenvolvimento de aplicativos móveis que foi lançado pela primeira vez em 2017. Ele permite que os desenvolvedores criem aplicativos para iOS e Android a partir de uma única base de código.
- Uma das principais vantagens do Flutter é que ele é altamente personalizável e oferece uma ampla gama de widgets pré-construídos que podem ser usados para criar interfaces de usuário bonitas e intuitivas.
- Ao combinar o poder do TensorFlow com a flexibilidade do Flutter, os desenvolvedores podem criar aplicativos que processam imagens em tempo real.
- Por exemplo, um aplicativo pode usar um modelo TensorFlow para identificar objetos em uma imagem e exibir informações relevantes na tela.
- Além disso, o Flutter oferece uma variedade de ferramentas de animação e design que podem ser usadas para criar interfaces de usuário envolventes e interativas que respondem aos resultados do processamento de imagem.

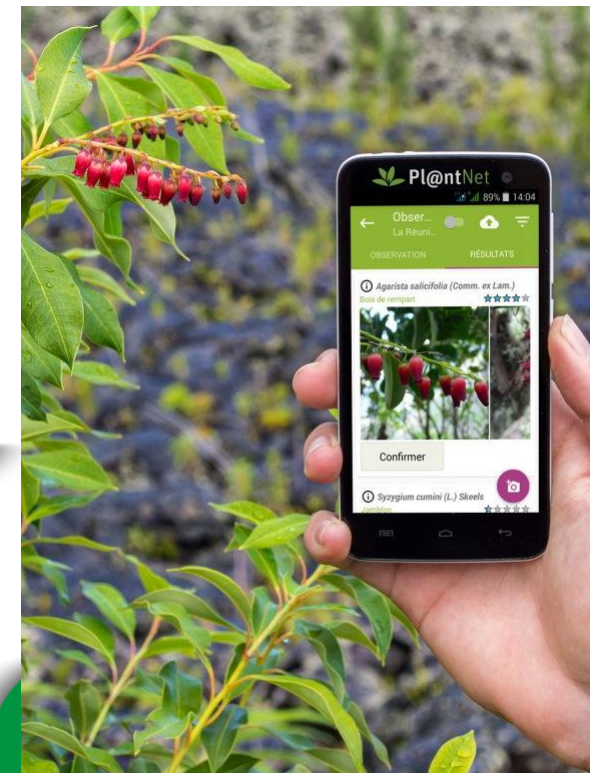


IA

Exemplo de Aplicativos que Usam Inteligência Artificial

- Há muitos exemplos de aplicativos móveis que usam o Flutter e o TensorFlow para criar experiências de usuário inovadoras e envolventes usando inteligência artificial.
- Aplicativos multiplataformas como o Plant.ID para identificação de plantas e doenças utilizando modelos de inteligência artificial

Plant.id



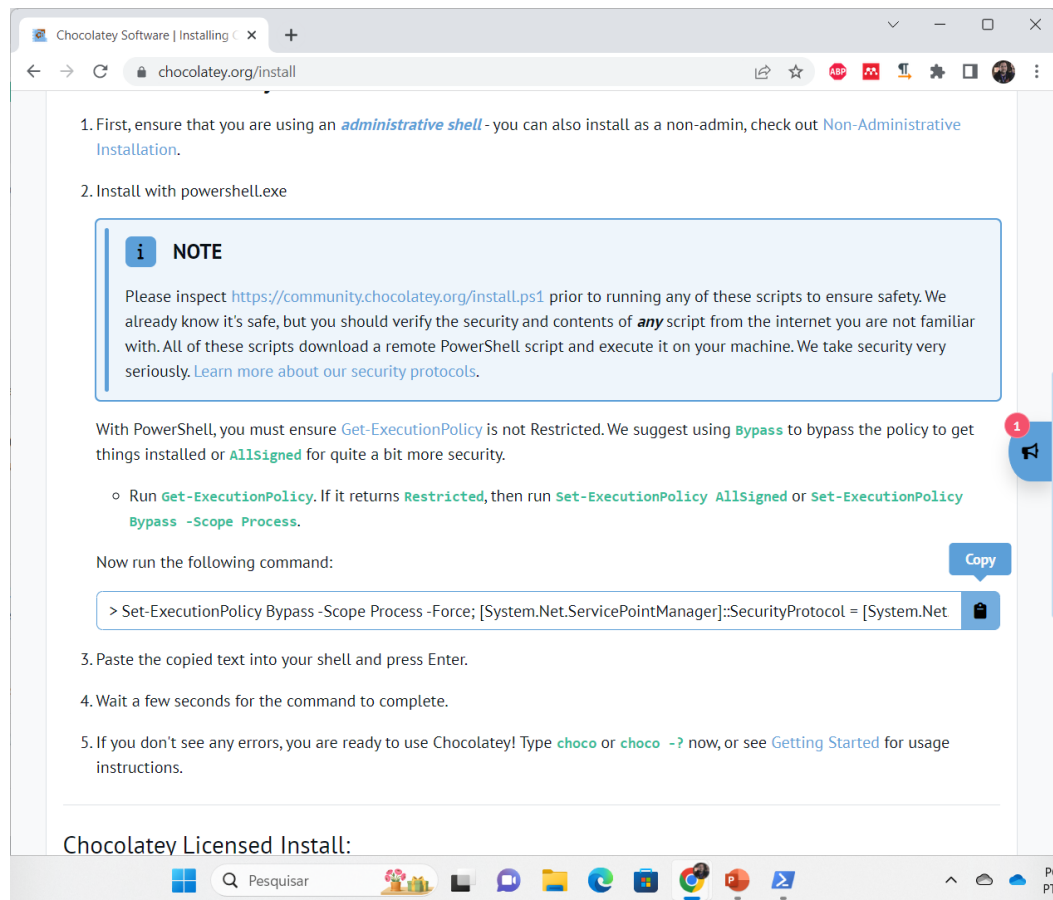


IA

Configurando o Flutter no Windows

Abrir a página de instalação do chocolatey

<https://chocolatey.org/install>



Chocolatey Software | Installing

chocolatey.org/install

1. First, ensure that you are using an **administrative shell** - you can also install as a non-admin, check out [Non-Administrative Installation](#).
2. Install with powershell.exe

NOTE

Please inspect <https://community.chocolatey.org/install.ps1> prior to running any of these scripts to ensure safety. We already know it's safe, but you should verify the security and contents of **any** script from the internet you are not familiar with. All of these scripts download a remote PowerShell script and execute it on your machine. We take security very seriously. [Learn more about our security protocols](#).

With PowerShell, you must ensure **Get-ExecutionPolicy** is not Restricted. We suggest using **Bypass** to bypass the policy to get things installed or **AllSigned** for quite a bit more security.

- Run **Get-ExecutionPolicy**. If it returns **Restricted**, then run **Set-ExecutionPolicy AllSigned** or **Set-ExecutionPolicy Bypass -Scope Process**.

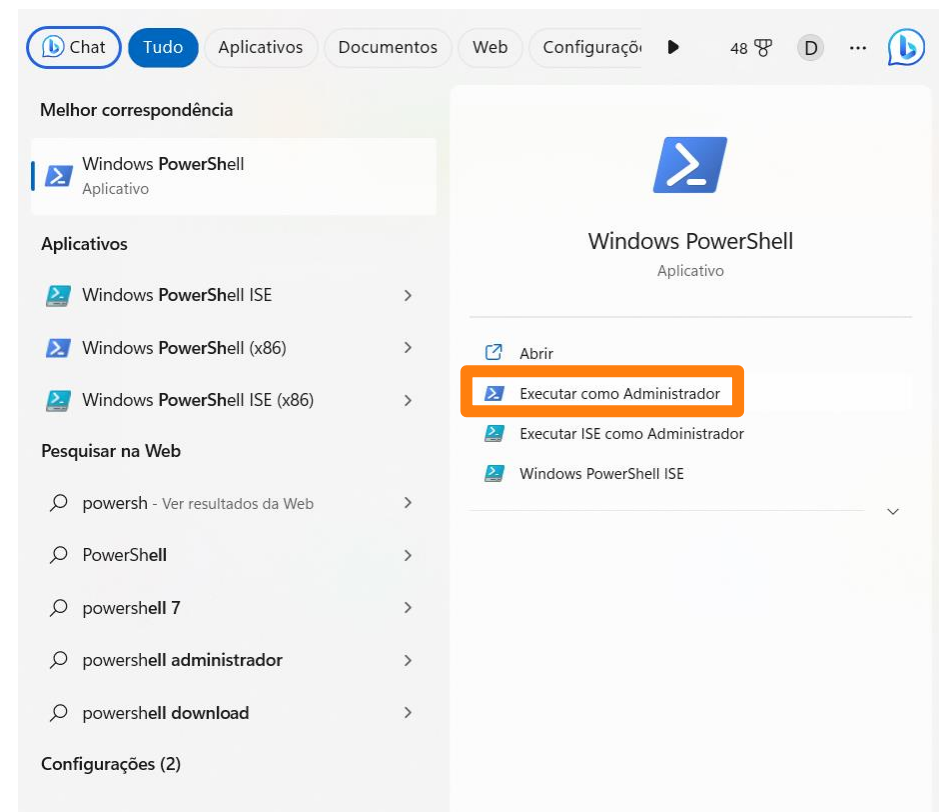
Now run the following command:

```
> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net
```

Copy

3. Paste the copied text into your shell and press Enter.
4. Wait a few seconds for the command to complete.
5. If you don't see any errors, you are ready to use Chocolatey! Type **choco** or **choco -?** now, or see [Getting Started](#) for usage instructions.

Chocolatey Licensed Install:



Chat Tudo Aplicativos Documentos Web Configurações 48 D ...

Melhor correspondência

Windows PowerShell
Aplicativo

Aplicativos

- Windows PowerShell ISE
- Windows PowerShell (x86)
- Windows PowerShell ISE (x86)

Pesquisar na Web

- powersh - Ver resultados da Web
- PowerShell
- powershell 7
- powershell administrador
- powershell download

Configurações (2)

Windows PowerShell
Aplicativo

- Abrir
- Executar como Administrador**
- Executar ISE como Administrador
- Windows PowerShell ISE

Menu iniciar > powershell > executar como administrador



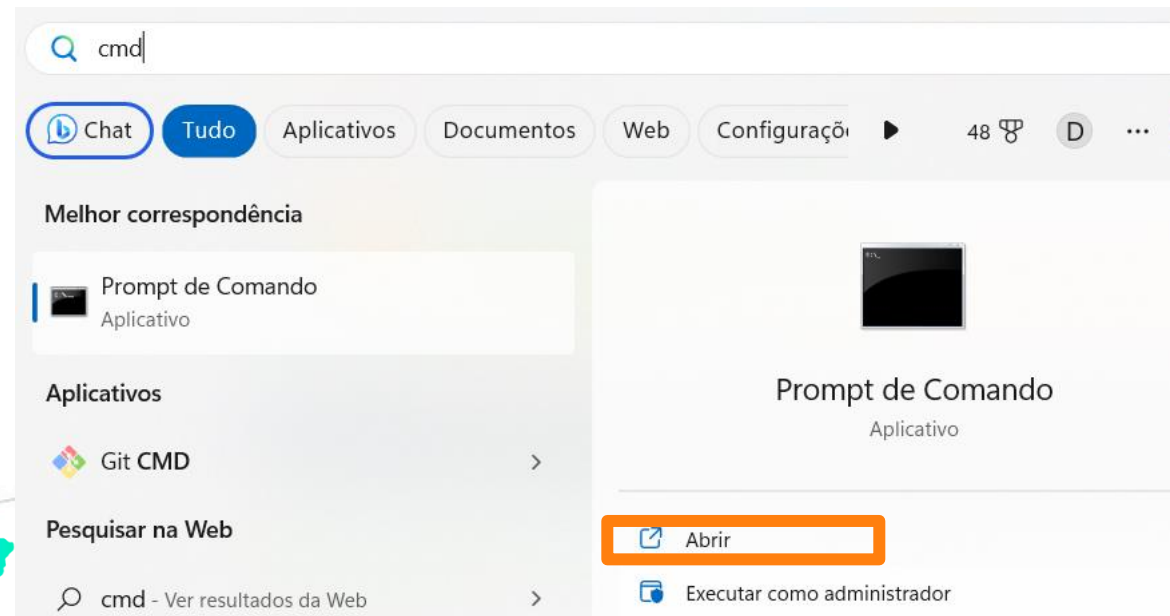
IA

Habilitando o Chocolatey no Windows

- Colar o código de inicialização do chocolatey

```
Administrador: Windows PowerShell
PS C:\windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

- Após a conclusão, para começar a usar o chocolatey, deve-se fechar o powershell e abrí-lo novamente através do powershell ou cmd como administrador





IA

Instalando pacotes através do chocolatey

- `choco install vlc android-sdk visualstudio2022buildtools visualstudio2022-workload-visualstudioextension visualstudio2019buildtools openjdk`

```
Administrador: Prompt de Comando - choco install vscode flutter
Microsoft Windows [versão 10.0.22621.1555]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Windows\System32>choco install vscode flutter
Chocolatey v1.2.1
Installing the following packages:
vscode;flutter
By installing, you accept licenses for the packages.
vscode v1.74.3 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.
Progress: Downloading flutter 3.7.10... 100%

flutter v3.7.10 [Approved]
flutter package files install completed. Performing other installation steps.
The package flutter wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
```




IA

Instalando o Flutter

- Já em modo de usuário baixe a versão do [site](#) e extraia para o diretório /tools/flutter

Flutter

Multi-Platform ▾ Development ▾ Ecosystem ▾ Showcase Docs ▾ 🔍 🐦

Get started ^

- 1. Install
- 2. Set up an editor
- 3. Test drive
- 4. Write your first app
- 5. Learn more

► From another platform?

Dart language overview ☑

Stay up to date ▾

Samples & tutorials ▾

User interface ▾

Data & backend ▾

Accessibility & localization ▾

Platform documentation

Set up an editor >

Install

[Get started](#) > Install

Select the operating system on which you are installing Flutter:

Windows

macOS

Linux

ChromeOS

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



IA

Configurando o ambiente

- Verificar se o sdk está já instalado

```
C:\Windows\System32>c:\Android\android-sdk\tools\bin\sdkmanager --install "cmdline-tools;latest"
```

- Instalar a plataforma android-29 ou superior

```
C:\Users\rodrigo.costa\flutter>c:\Android\android-sdk\tools\bin\sdkmanager.bat "platforms;android-29" "platform-tools" "build-tools;29.0.3"  
Warning: File C:\Users\rodrigo.costa\.android\repositories.cfg could not be loaded.  
[====] 11% Downloading platform-29 r05.zip...
```

- Aceitar as licenças do flutter

```
C:\Windows\System32> flutter doctor --android-licenses  
[=====] 100% Computing updates...  
6 of 7 SDK package licenses not accepted.  
Review licenses that have not been accepted (y/N)? y  
  
1/6: License android-googletv-license:  
-----  
Terms and Conditions  
  
This is the Google TV Add-on for the Android Software Development Kit License Agreement.
```



IA

Verificando os pactoes instalados

```
C:\Windows\System32>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.7.10, on Microsoft Windows [vers o 10.0.22621.1555], locale pt-BR)
[X] Windows Version (Unable to confirm if installed Windows version is 10 or greater)
[✓] Android toolchain - develop for Android devices (Android SDK version 29.0.3)
[✓] Chrome - develop for the web
[!] Visual Studio - develop for Windows (Ferramentas de Build do Visual Studio 2022 17.5.4)
    X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop development with C++"
      workload, and include these components:
        MSVC v142 - VS 2019 C++ x64/x86 build tools
          - If there are multiple build tool versions available, install the latest
        C++ CMake tools for Windows
        Windows 10 SDK
[!] Android Studio (not installed)
[✓] VS Code, 64-bit edition (version 1.74.3)
[✓] Connected device (3 available)
[✓] HTTP Host Availability
```

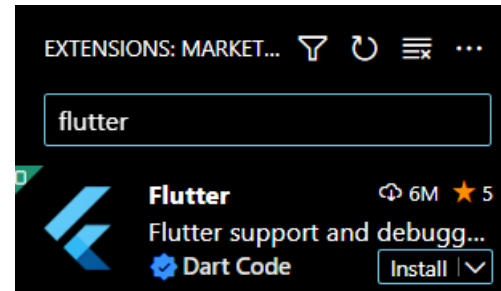


IA

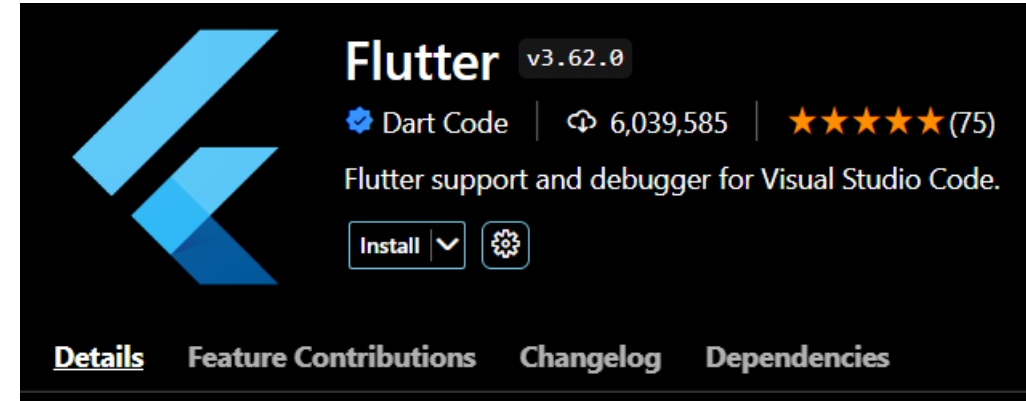
Instalando o plugin do flutter no vscode



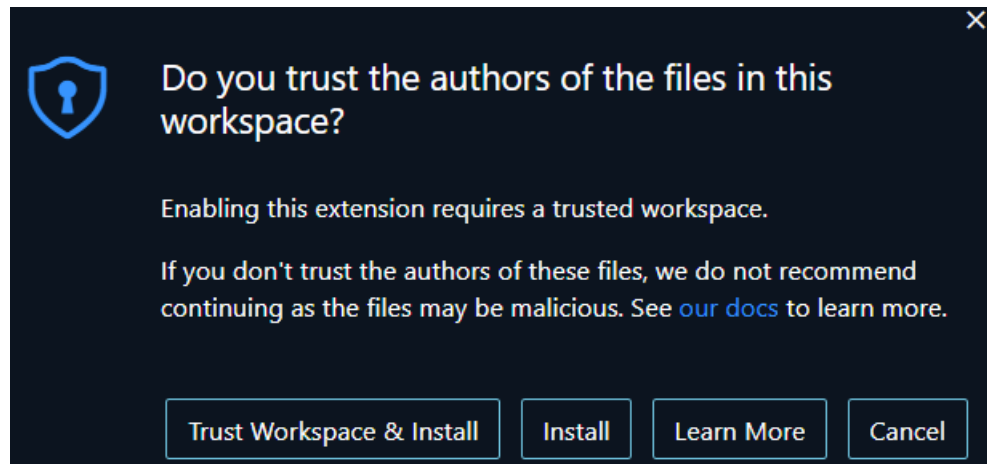
Clique no ícone de extensões



Digite flutter



Instale o plugin do flutter



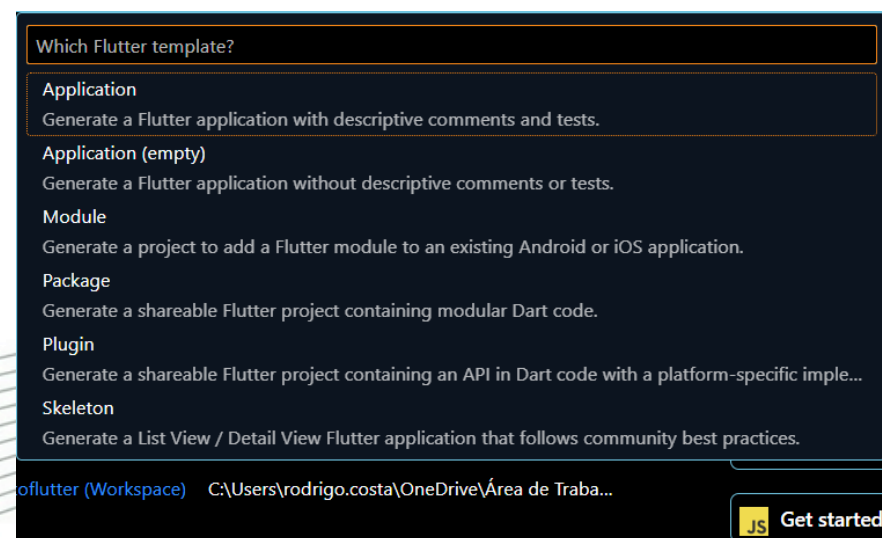
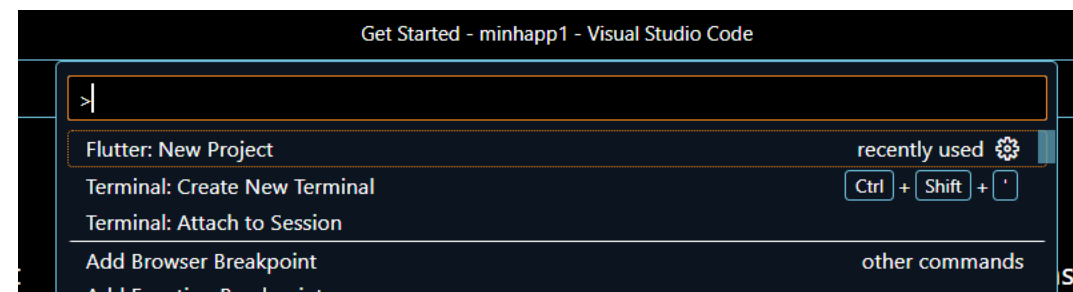
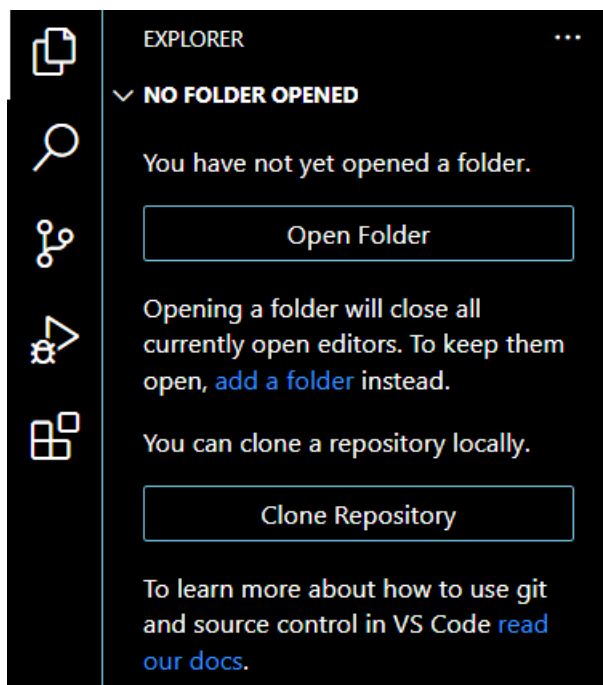
Autorize o vscode para confiar no workspace para habilitar o uso do plugin do flutter



IA

Criando um projeto flutter

- Retornar ao explorer e incluir uma pasta para trabalhar com o flutter ou criar o projeto via prompt

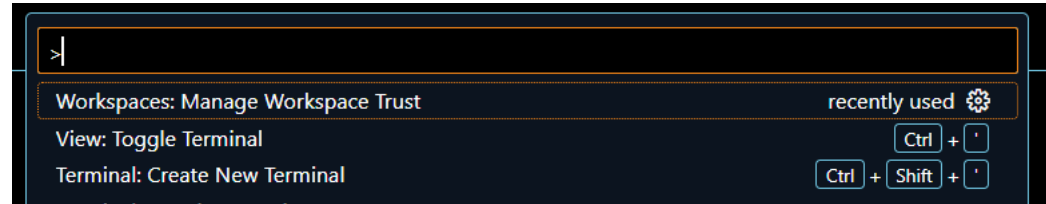




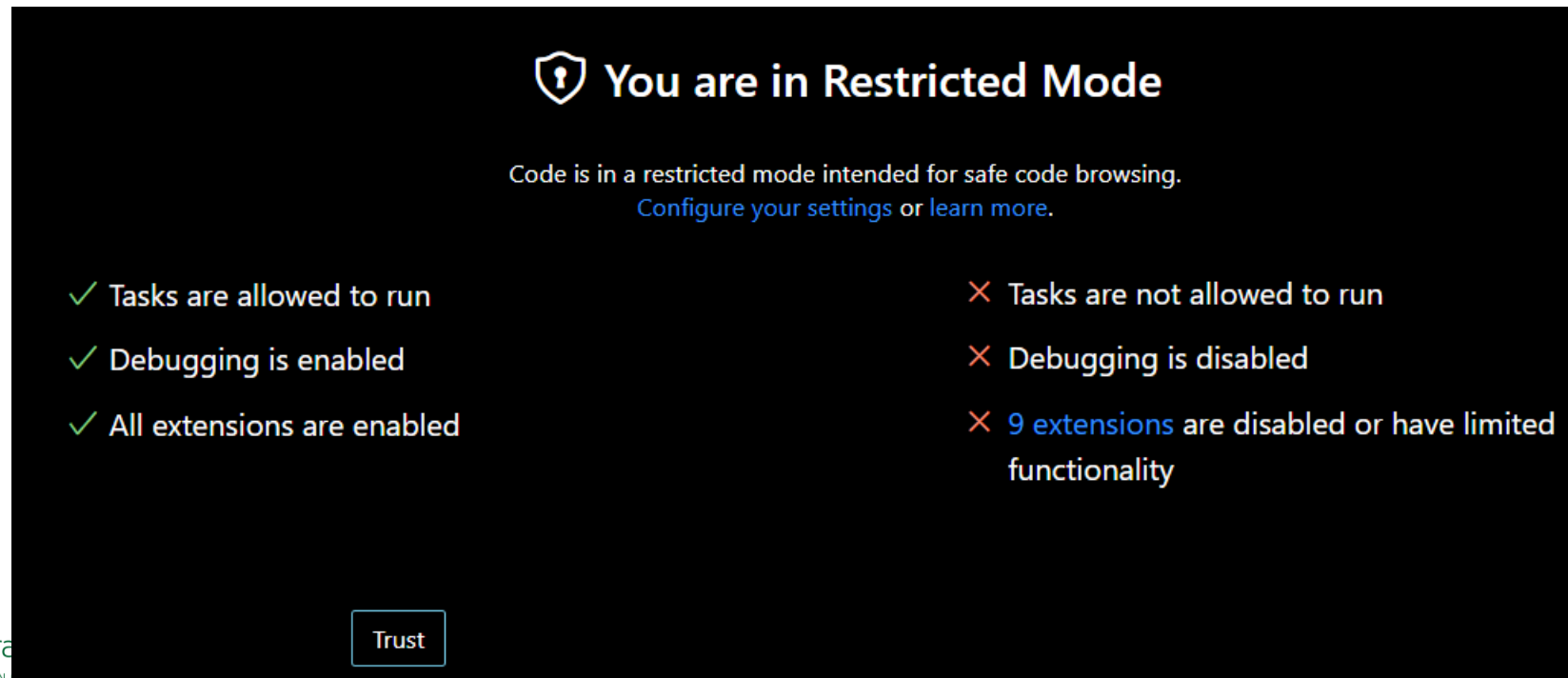
IA

Se o comando não aparecer, vc deverá tornar a janela confiável

- Aperte Ctrl + shift + P e digite workspace trust



- E clicar em trust





IA

Criando o projeto flutter

- Você também pode fazer isso no prompt de comando:

`flutter create --org com.yourdomain your_app_name`

```
Prompt de Comando - flutter x + v

C:\Users\rodrigo.costa\flutter>flutter create --org br.org.institutoiracema tensorflow
Creating project tensorflow...
Running "flutter pub get" in tensorflow...
Resolving dependencies in tensorflow... (1.4s)
+ async 2.10.0 (2.11.0 available)
+ boolean_selector 2.1.1
+ characters 1.2.1 (1.3.0 available)
+ clock 1.1.1
+ collection 1.17.0 (1.17.1 available)
+ cupertino_icons 1.0.5
+ fake_async 1.3.1
+ flutter 0.0.0 from sdk flutter
+ flutter_lints 2.0.1
+ flutter_test 0.0.0 from sdk flutter
+ js 0.6.5 (0.6.7 available)
+ lints 2.0.1
+ matcher 0.12.13 (0.12.15 available)
+ material_color_utilities 0.2.0 (0.3.0 available)
+ meta 1.8.0 (1.9.1 available)
+ path 1.8.2 (1.8.3 available)
+ sky_engine 0.0.99 from sdk flutter
+ source_span 1.9.1 (1.10.0 available)
+ stack_trace 1.11.0
+ stream_channel 2.1.1
+ string_scanner 1.2.0
+ term_glyph 1.2.1
+ test_api 0.4.16 (0.5.2 available)
+ vector_math 2.1.4
Changed 24 dependencies in tensorflow!
```



IA

Começando com o tensorflow

- Incluir as bibliotecas necessárias, no caso a tensorflow lite e a image_picker (para poder abrir imagens no flutter)

```
pubspec.yaml X main.dart
! pubspec.yaml
27 # dependencies can be managed
28 # the latest version available
29 # versions available, run
30 dependencies:
31   flutter:
32     sdk: flutter
33   image_picker: ^0.8.3+2
34   tflite: ^1.1.2
```

- Ir no console e dar o comando flutter pub get para baixar as novas dependências (ele executa automaticamente após salvar)

```
C:\Users\rodrigo.costa\OneDrive\Área de Trabalho\meuprojetoflutter\flutter_application_1>flutter pub get
Running "flutter pub get" in flutter_application_1...
Resolving dependencies... (1.0s)
```




IA

Alterando o projeto de exemplo criado alterando o main.darth

- O modelo criado fica incrementando uma variável a cada clique na tela.
- Vamos mudar o código fonte criado para permitir o uso do modelo criado no tensorflow
- Primeiramente vamos importar as bibliotecas necessárias
 - darth.io para manipulação de arquivos
 - image_picker: carregamento de imagens
 - tfite: tensorflow
- Incluir as permissões no manifest

```
! pubspec.yaml  main.dart  AndroidManifest.xml
lib > main.dart > _MyHomePageState > build
1  import 'package:flutter/material.dart';
2  //inclur as bibliotecas
3  import 'dart:io';
4  import 'package:image_picker/image_picker.dart';
5  import 'package:tfite/tfite.dart';
```

```
! pubspec.yaml  main.dart  AndroidManifest.xml
android > app > src > main > AndroidManifest.xml
1  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2    package="com.example.flutter_application_1">
3    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
4    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
5    <application
```

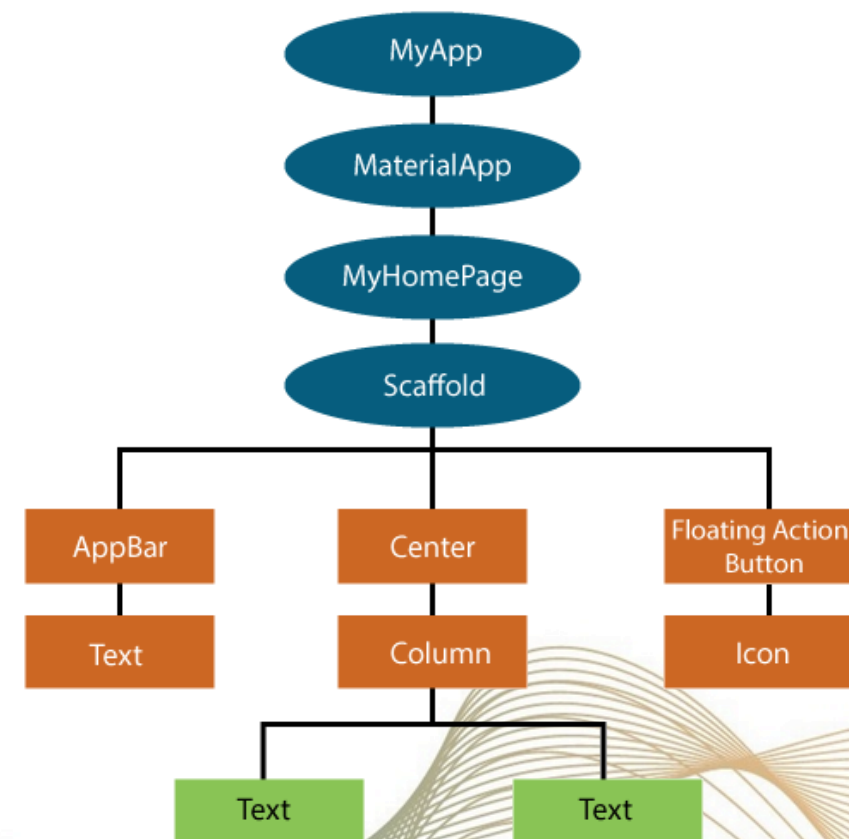


IA

Vamos personalizar a tela

- o método **Widget Build**, que é responsável por construir o método a interface gráfica do aplicativo.
 - recebe como parâmetro um objeto chamado **BuildContext**, que contém informações sobre o contexto em que o widget está sendo construído,
 - retorna uma árvore de widgets que representam a interface do usuário.
- **Scaffold** é um dos widgets mais importantes do Flutter.
 - fornece uma estrutura básica para o layout do aplicativo, incluindo a barra de navegação, a barra de título e o menu lateral.

```
Widget build(BuildContext context) {  
  return ....  
}
```

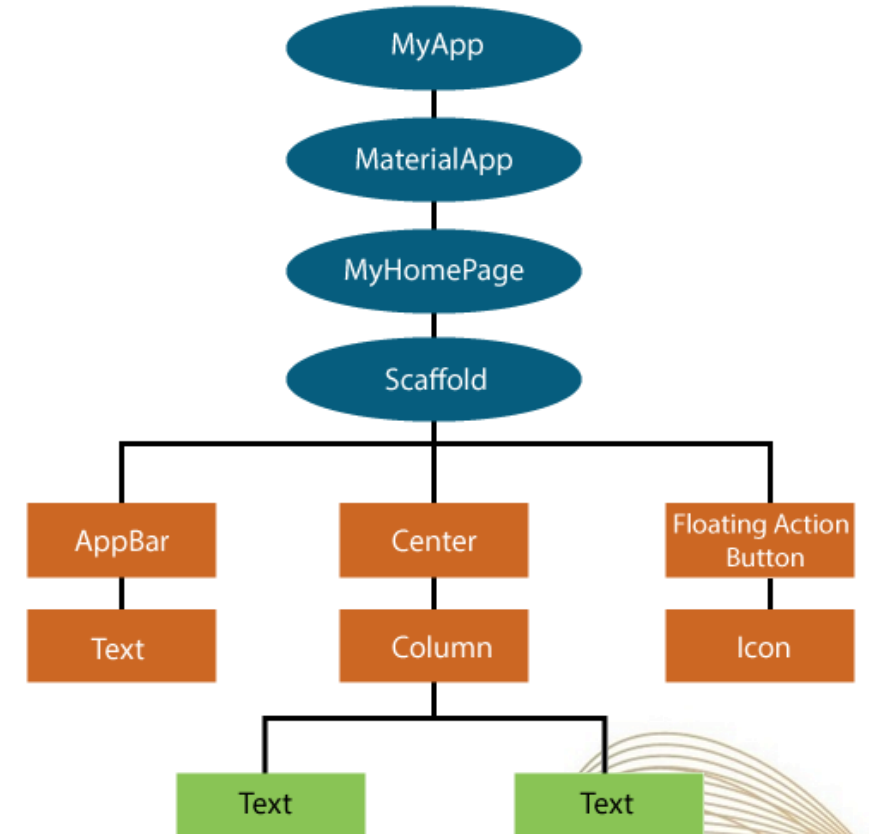


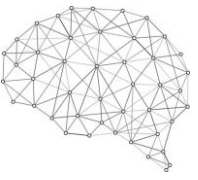


IA

Interface do código criado automaticamente

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: Text(widget.title),  
    ),  
    body: Center(  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: <Widget>[  
          const Text(  
            'You have pushed the button this many times:',  
          ),  
          Text(  
            '$_counter',  
            style: Theme.of(context).textTheme.headlineMedium,  
          ),  
        ],  
      ),  
    ),  
    floatingActionButton: FloatingActionButton(  
      onPressed: _incrementCounter,  
      tooltip: 'Increment',  
      child: const Icon(Icons.add),  
    ),  
  );  
}
```





IA

Interface do código criado automaticamente

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: Text(widget.title),  
    ),  
    body: Center(  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: <Widget>[  
          _image == null? Text("No image selected.") : Image.file(_image), SizedBox(height: 20),  
          ElevatedButton(  
            onPressed: _getImage, child: Text("Select Image"),  
          ),  
          SizedBox(height: 20),  
          ElevatedButton(  
            onPressed: _processImage, child: Text("Process"),  
          ),  
          SizedBox(height: 20),  
          _classification == null?Text(""): Text( "Classification: $_classification", style: TextStyle(fontSize: 18),  
            )  
        ],  
      ),  
    ),  
  );  
}
```

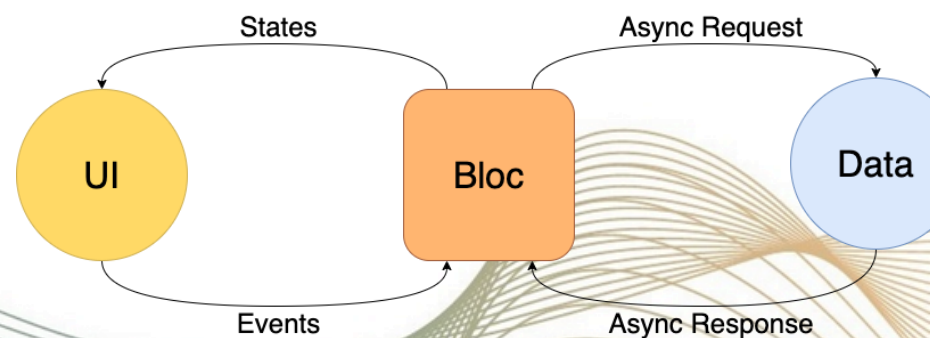
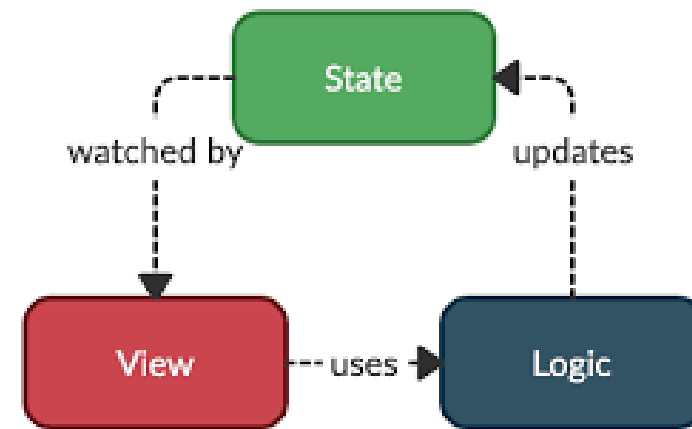




IA

Criando os call-backs chamados ao clicar nos botões

```
class _MyHomePageState extends State<MyHomePage>
{
  late File _image;
  late String _classification;
  final picker = ImagePicker();
  void _getImage() async {
    final pickedFile = await picker.getImage(source:ImageSource.gallery);
    if (pickedFile != null) {
      setState(() {_image = File(pickedFile.path);
    });
  }
}
void _processImage() async {
  String? res = await Tflite.loadModel(
    model: "assets/fruits.tflite", labels: "assets/labels.txt");
  var output = await Tflite.runModelOnImage(
    path: _image.path,
    numResults: 1, threshold: 0.05, imageMean: 0.0, imageStd: 255.0,
  );
  setState(() {
    _classification = output?[0]["label"] ?? "Unable to classify";
  });
}
```

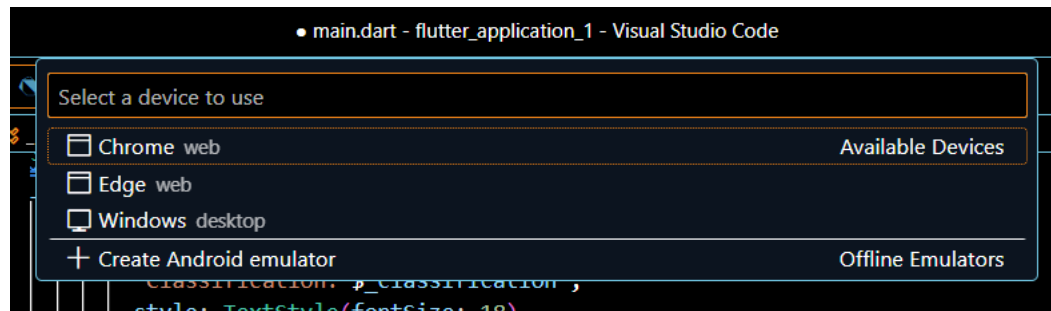
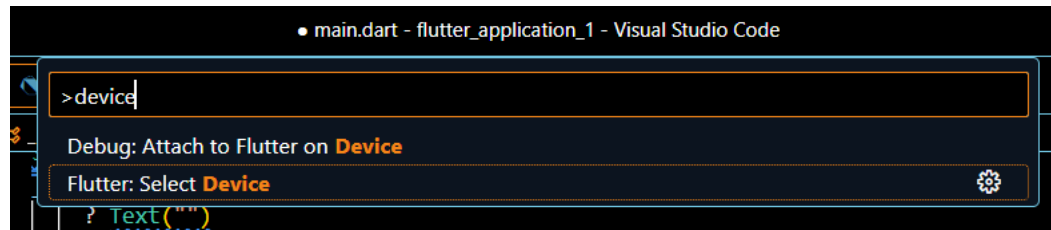




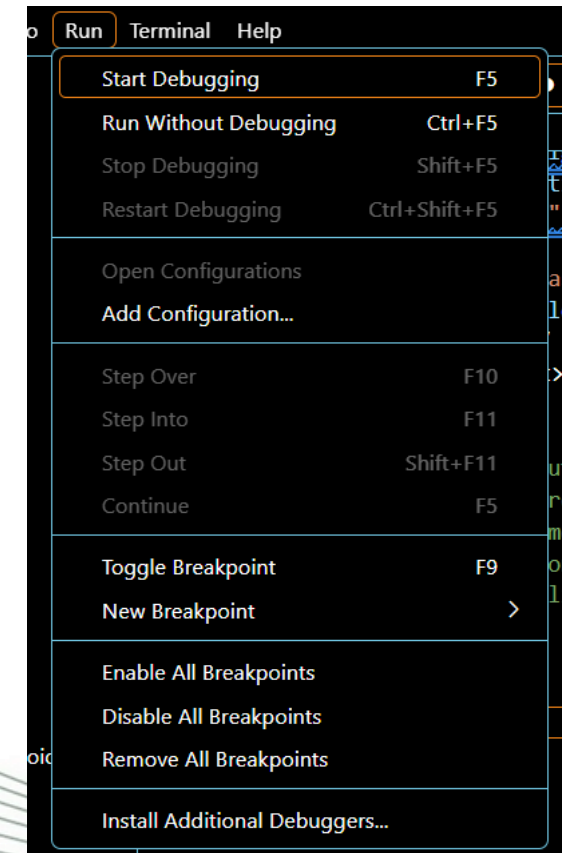
IA

Executando o programa

- Selecione-se o device para rodar



- Em seguida pode-se iniciar a execução

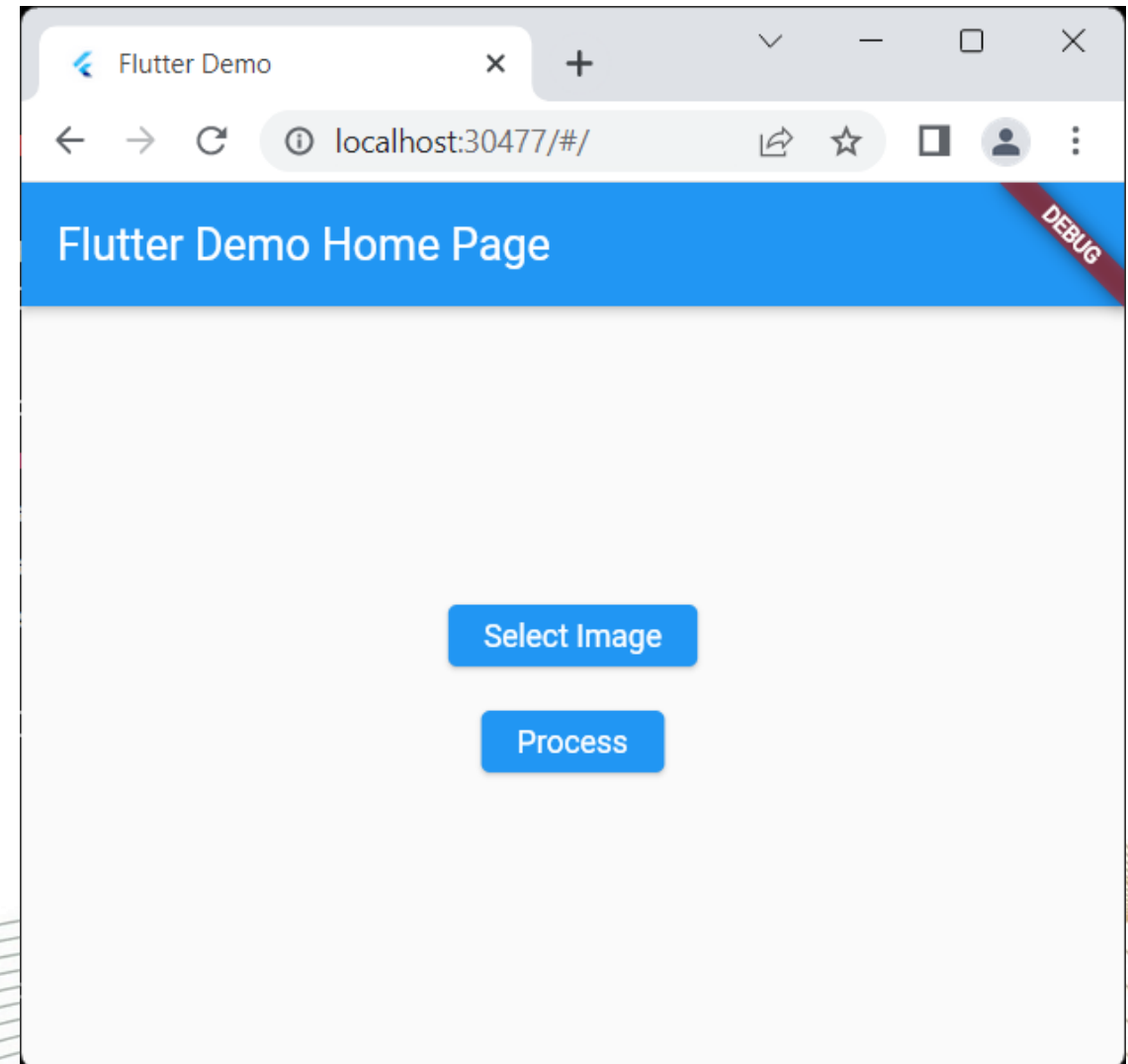




IA

Você pode testar a aplicação

- O Código fonte está em desenvolvimento e está disponível em
- <https://github.com/rodccosta/fruits360fluttertensorflow>



Dúvidas?

Módulo de Inteligência Artificial



UNIVERSIDADE
ESTADUAL DO CEARÁ



Instituto Iracema
PESQUISA E INOVAÇÃO



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

