



Material de Apoio do Curso

INTERFACE HOMEM MÁQUINA MICROCONTROLADA

Material:

Instalações Curso IHMM

Ministrantes do Curso:

Guilherme Franco

Reinan Lima

Data do Curso:

10/12/2024

Apresentação

Para esse curso, onde iremos criar interfaces, precisamos primeiro preparar o ambiente onde vamos trabalhar, e para isso temos que instalar algumas coisas na máquina. Nosso curso vai utilizar a linguagem de programação Python para construção da interface e funções aplicadas a ela. Por isso precisamos que nosso computador seja capaz de executar os códigos em python que escrevemos, e importante também, que possamos escrever o código de forma tranquila, usando um editor de texto adequado.

Instalação do Python

Para instalar o Python, basta ir na página oficial e ir na parte de download. Qualquer versão pode ser instalada, mas recomendamos a versão mais recente, a `3.13.0`

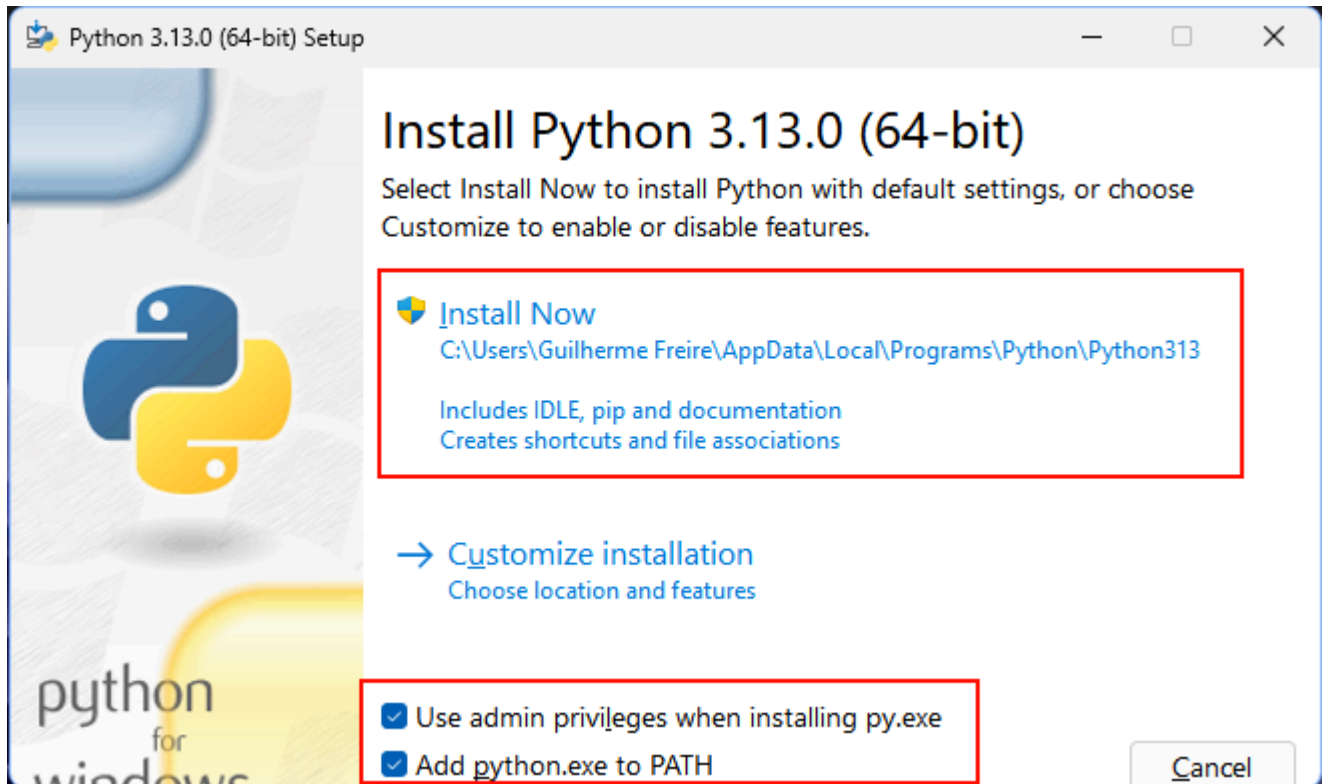
[PÁGINA DE DOWNLOAD](#)

Na página de download você vai baixar o instalador do python em alguma pasta do seu computador, e executar. Ele abrirá uma janela para configurar a instalação do python.

Antes de prosseguir, você vai marcar as duas caixas de seleção na parte inferior da janela.

☒ Use admin privileges when installing py.exe

☒ Add python.exe to PATH



Adicionar o python.exe to PATH, quer dizer adicionar o caminho onde o python será instalado, nas variáveis de ambiente. Autorizar uma ação como essa, possibilita que você chame o interpretador do python de qualquer lugar do seu computador. Por exemplo, abrindo o prompt de comando e enviar `python`, é o mesmo que você executar o `.exe` do python.

```
Windows PowerShell
PS C:\Users\Guilherme Freire> python
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr  2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World")
Hello World
>>> for i in range(5):
...     print(i)
...
0
1
2
3
4
>>>
```

✓ Com isso temos a instalação do python concluída com sucesso.

Instalação do editor de texto

Agora precisamos de um editor de texto para escrever nosso código python. Claro, podemos usar o bloco de notas para escrever o código normalmente, mas ele não nos oferece as mesmas capacidades que um editor de texto robusto tem, como coloração da sintaxe, ajuda na escrita, sugestões, definições, e movimentações do cursor.

Eu recomendo usar o VS Code para códigos python, pela versatilidade, além de o ambiente de escrita python poder ser altamente customizado com a ajuda de extensões.

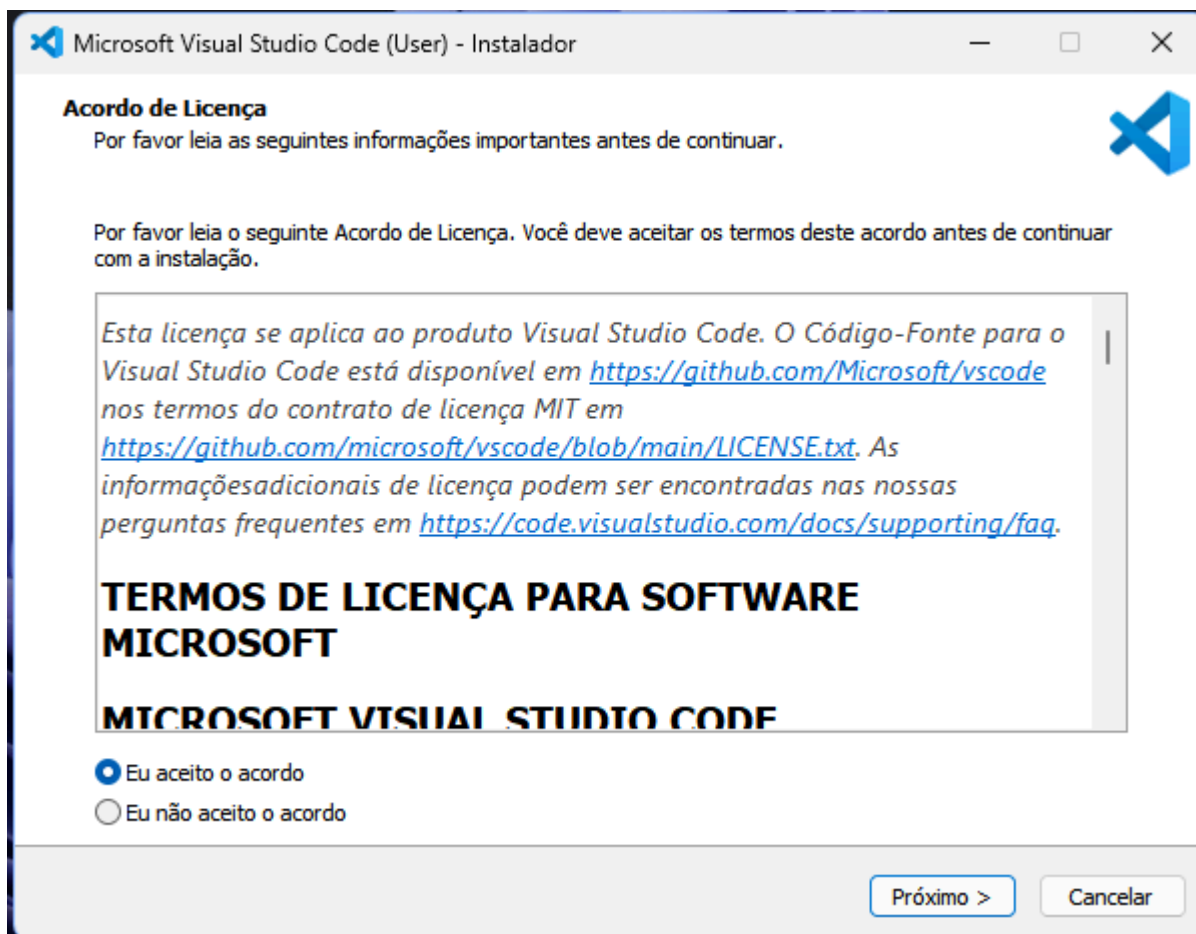
Outras ferramentas podem ser usadas, é claro, mas nós que iremos ministrar o curso usaremos o VS Code como editor de código principal.

[PyCharm](#) é uma boa ferramenta também, mas ela é muito grande e pesada, em computadores de capacidade mais baixa, ela pode travar, o que acaba mais atrapalhando. [Sublime Text](#), [Notepad++](#), são outros editores que podem ser usados.

Para instalar o **VS Code** basta ir na página de downloads e instalar a versão mais recente compatível com seu computador.

[PÁGINA DE DOWNLOAD](#)

Na página de download, você irá baixar o instalador e executá-lo.

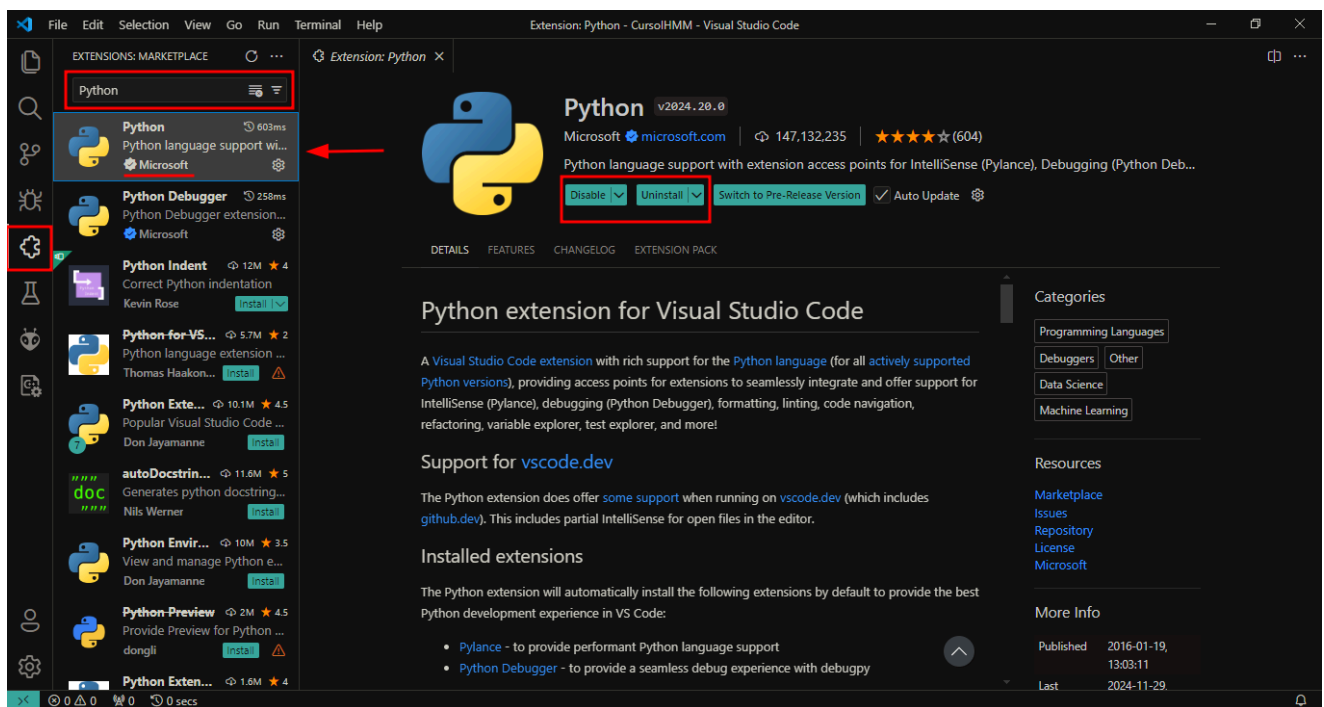


Você tem que marcar a opção "Eu aceito o acordo" e apertar o botão "Próximo" para prosseguir a instalação. Basta ir prosseguindo na instalação normalmente. Lembre de adicionar ao PATH caso o instalador peça.

✓ Sendo assim já temos instalado nosso editor de texto com sucesso.

Adicionando suporte ao Python no Vs Code

Ao baixar e executar o Vs Code pela primeira vez e tentar abrir algum arquivo python, ele não vai saber o que fazer, isso é porque precisamos adicionar o suporte ao python primeiro. Para fazer isso precisamos ir na aba de extensões e baixar a extensão que adiciona diversas funcionalidades que facilitam nossa vida ao programar em python.



Na barra lateral, clique no ícone de peça de quebra cabeça, isso levará para a aba de extensões, onde você pode baixar não só a extensão para suporte a linguagem python, mas qualquer outra funcionalidade, temas, ícones disponíveis.

Quando você clicar na aba de extensões, abrirá uma nova tela, haverá um espaço onde você pode pesquisar por extensões, clique lá e escreva `Python`. Aparecerá uma lista de extensões com Python no nome, clique na extensão oficial disponibilizada pela Microsoft.

Ao clicar, as informações da extensão abrirão na tela principal, clique no botão `Install` para instalar a extensão. Na imagem acima, mostra a minha tela e eu já tenho essa extensão instalada, e por isso o botão `Install` foi substituído pelos botões de `Disable` e `Uninstall`.

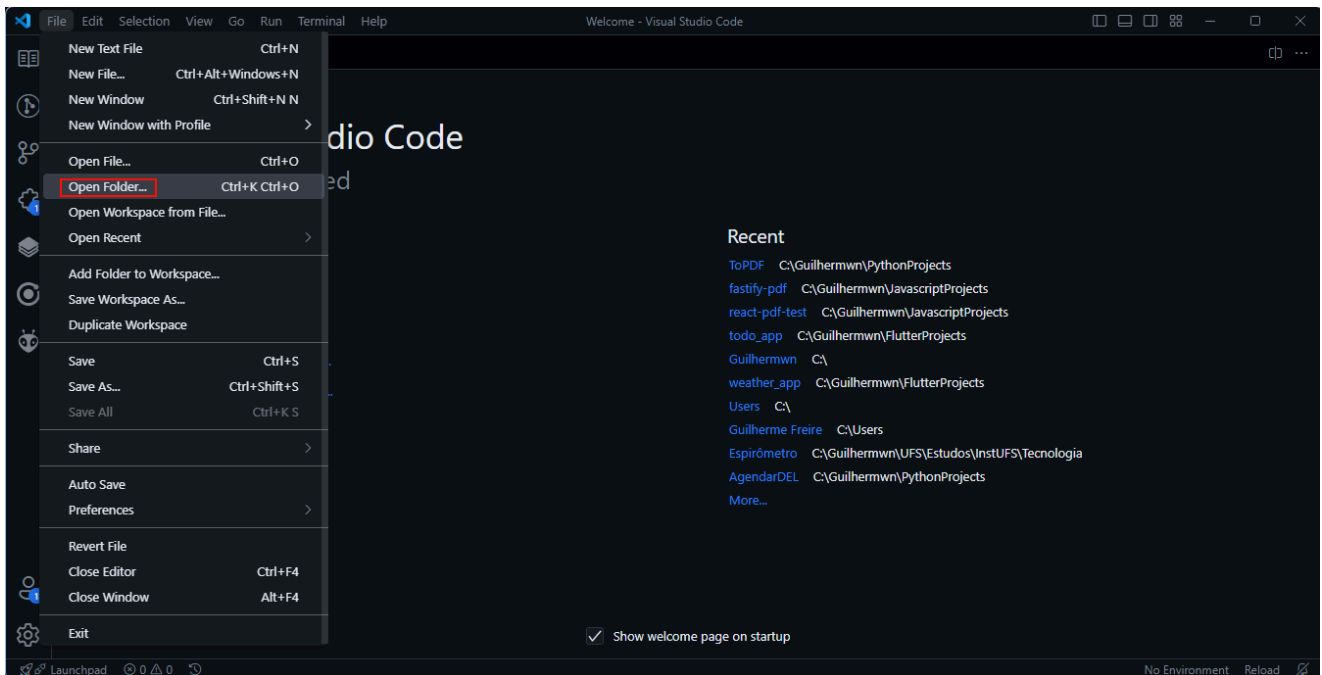
Essa extensão adiciona suporte a execução de scripts, suporte a linguagem, sugestão de funções e comandos, debug, formatação do código, navegação, e mais.

Iniciando um projeto

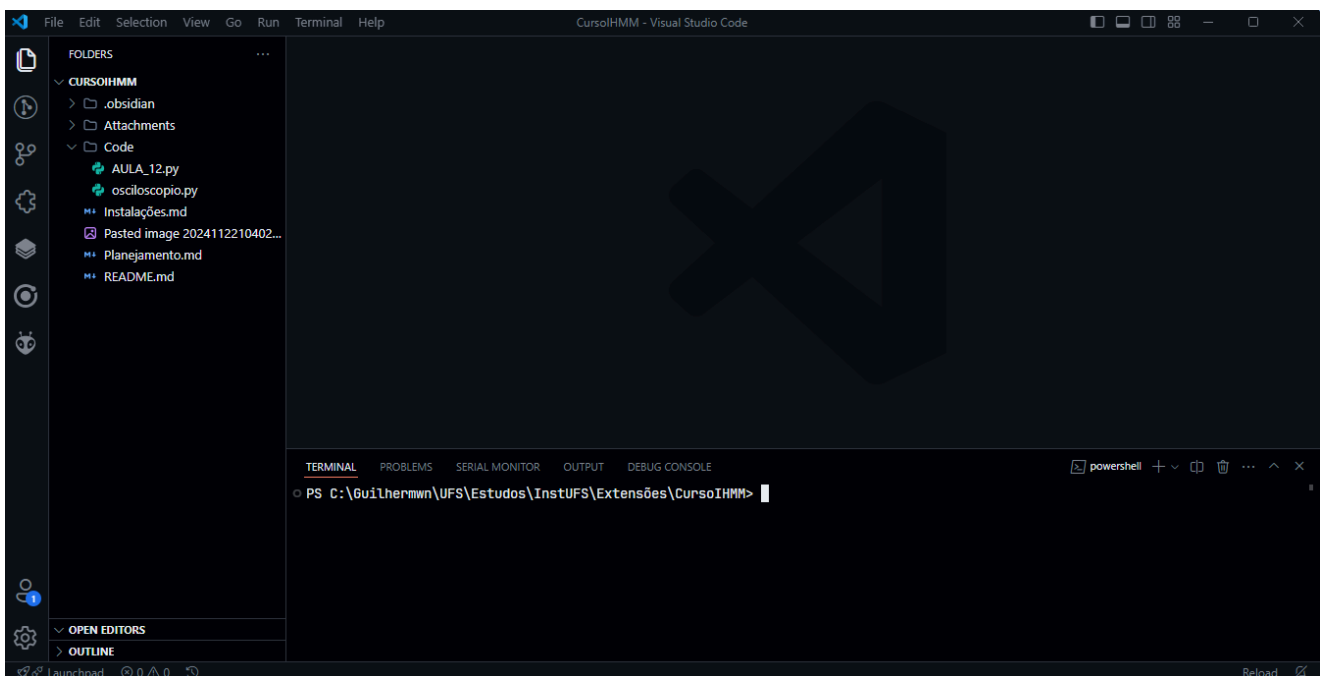
Ao abrir o VS Code, você vai no menu superior, clica na opção `File`, irá abrir um submenu com várias opções. Procure pela opção `Open Folder...` e clique nela. Essa opção irá abrir uma janela do explorador de arquivos, onde você poderá escolher uma pasta para abrir no editor.

Crie uma pasta dedicada aos arquivos do curso, e abra no editor através da opção `Open Folder`

 **File > Open Folder...**

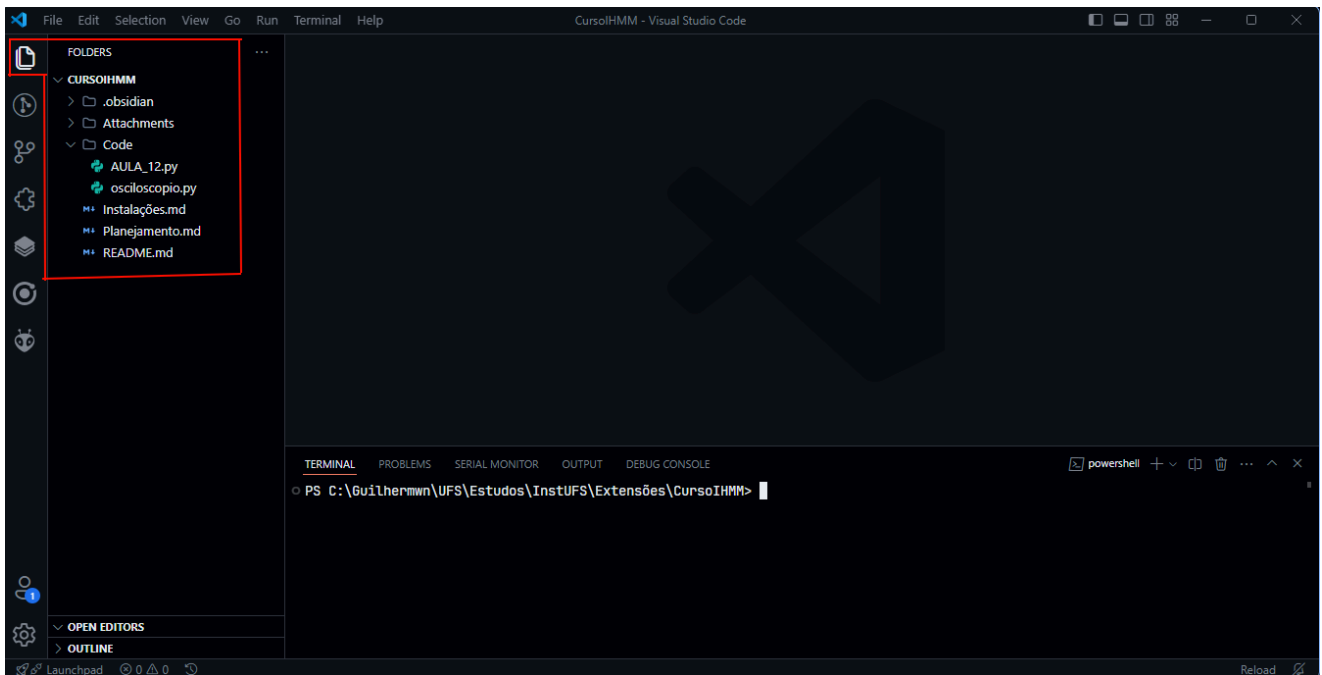


Ao abrir a pasta selecionada, você terá uma tela parecida com essa:

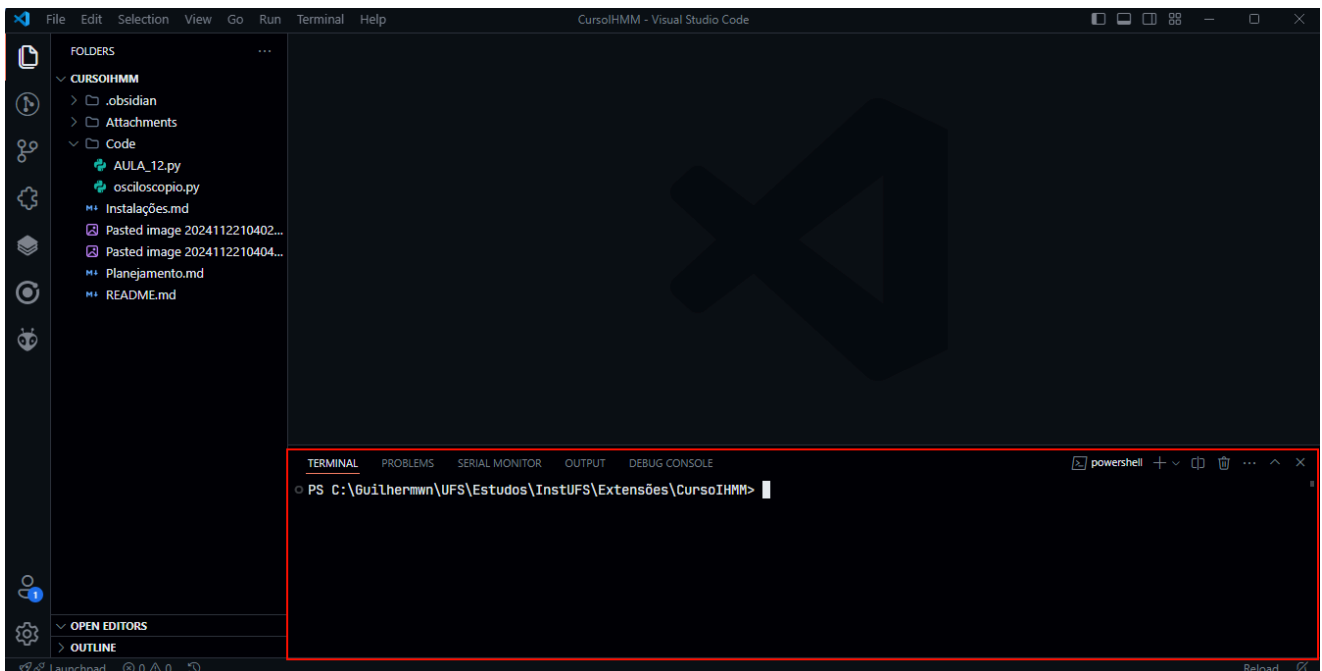


Na barra lateral esquerda, temos as abas dedicadas a extensões e funcionalidades extras, não precisa se preocupar com isso para o curso.

A parte selecionada em vermelho, é onde vemos a hierarquia de arquivos dentro da pasta aberta com todos os arquivos.



Na parte inferior em vermelho, temos o terminal integrado onde podemos executar comandos, ele sendo a mesma coisa que o terminal que abrimos para testar o python.



Instalação das bibliotecas

Para o nosso curso, precisaremos instalar algumas bibliotecas externas para o python, para adicionarmos mais funcionalidades a nossa interface.

O python possui um instalador de pacotes integrado, que já vem instalado, que acessa o repositório de pacotes [PyPi](https://pypi.org/).

Usamos o comando `pip` para instalar, desinstalar e ver informações dos pacotes.

As bibliotecas que usaremos no nosso curso serão:

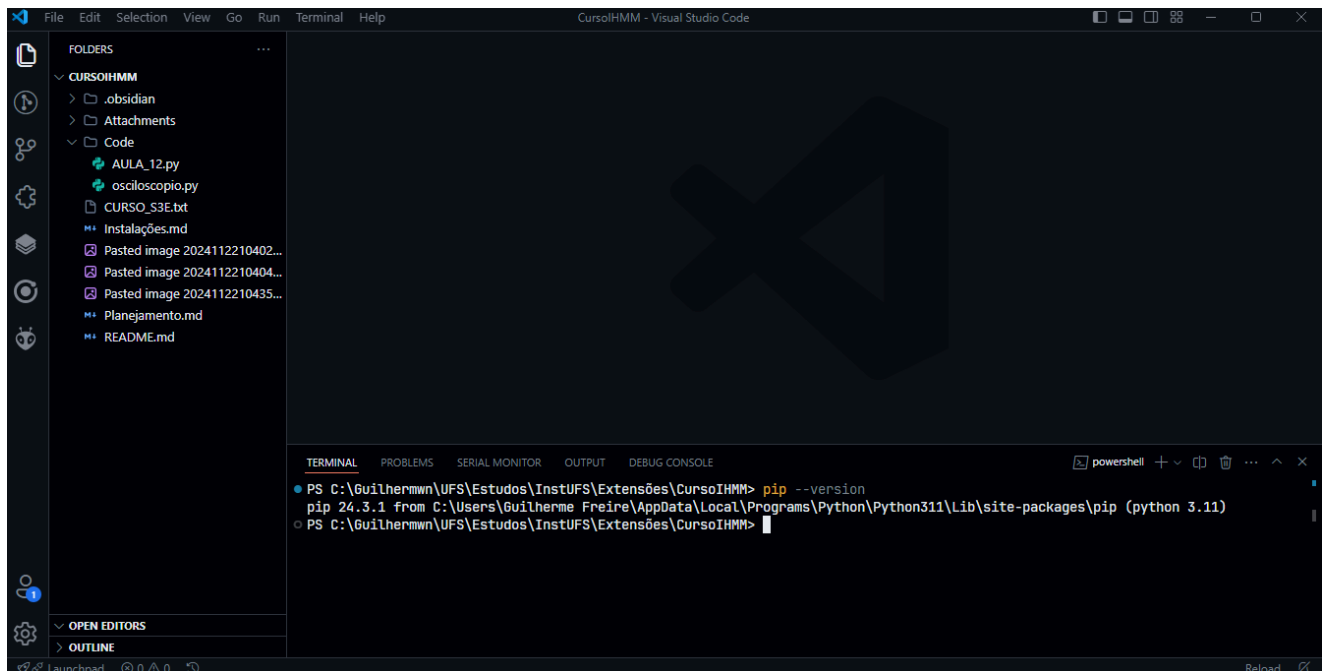
- [Tkinter](#)
- [Matplotlib](#)

- [Numpy](#)
- [PySerial](#)
- [CustomTkinter](#)

Antes de instalar vamos verificar se o `pip` está funcionando como deveria, para isso, inserimos no terminal o comando:

```
pip --version
```

O comando deve executar sem erros, e retornar a localização onde o arquivo do `pip` foi instalado.



Com isso verificado, podemos instalar as bibliotecas. O comando que usamos para instalar bibliotecas usando o pip é:

```
pip install <pacote>
```

Onde vamos substituir o termo `<pacote>` pelo nome no repositório do pacote que queremos usar. Sendo assim podemos enviar no terminal o comando de instalação para instalar as bibliotecas:

```
pip install matplotlib numpy pyserial customtkinter
```

Isso irá instalar as bibliotecas em um único comando. Caso queira instalar uma de cada vez, pode usar:

```
pip install matplotlib
pip install numpy
pip install pyserial
pip install customtkinter
```


The screenshot shows the Visual Studio Code interface. The left sidebar contains the 'FOLDERS' pane with a tree view of the project 'CURSOIHHM'. The tree includes folders like '.obsidian', 'Attachments', and 'Code', and files like 'AULA_12.py', 'osciloscopio.py', 'CURSO_S3E.txt', 'Instalações.md', 'Pasted image 2024112210400...', 'Pasted image 2024112210404...', 'Pasted image 2024112210435...', 'Pasted image 2024112210544...', 'Planejamento.md', and 'README.md'. The bottom of the sidebar has tabs for 'OPEN EDITORS' and 'OUTLINE'. The main editor area is titled 'CURSOIHHM - Visual Studio Code' and shows a terminal window. The terminal has a title bar with 'powershell' and standard window controls. The terminal content shows the command 'PS C:\Guilhermn\UFS\Estudos\InstUFS\Extensões\CursoIHHM> pip install matplotlib numpy pyserial' and its output. The output lists the requirements for matplotlib and the versions installed or available. The terminal text is as follows:

```
PS C:\Guilhermn\UFS\Estudos\InstUFS\Extensões\CursoIHHM> pip install matplotlib numpy pyserial
Requirement already satisfied: matplotlib in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-package
s (3.9.2)
Requirement already satisfied: numpy in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-packages (2.
0.1)
Requirement already satisfied: pyserial in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-packages
(3.5)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-p
ackages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-packa
ges (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-p
ackages (from matplotlib) (4.53.1)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-p
ackages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-pa
ckages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-packages
 (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\guilherme freire\appdata\local\programs\python\python311\lib\site-p
ackages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\guilherme freire\appdata\roaming\python\python311\site-packages
 (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\guilherme freire\appdata\roaming\python\python311\site-packages (from pytho
n-dateutil>=2.7->matplotlib) (1.16.0)
PS C:\Guilhermn\UFS\Estudos\InstUFS\Extensões\CursoIHHM>
```