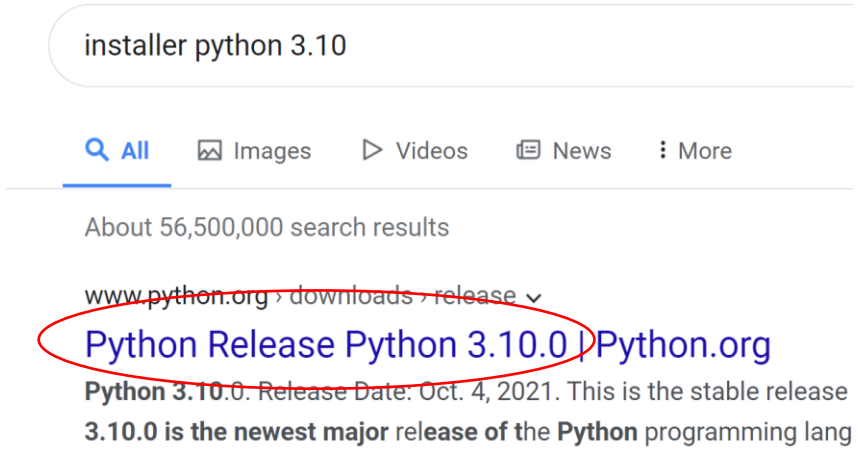
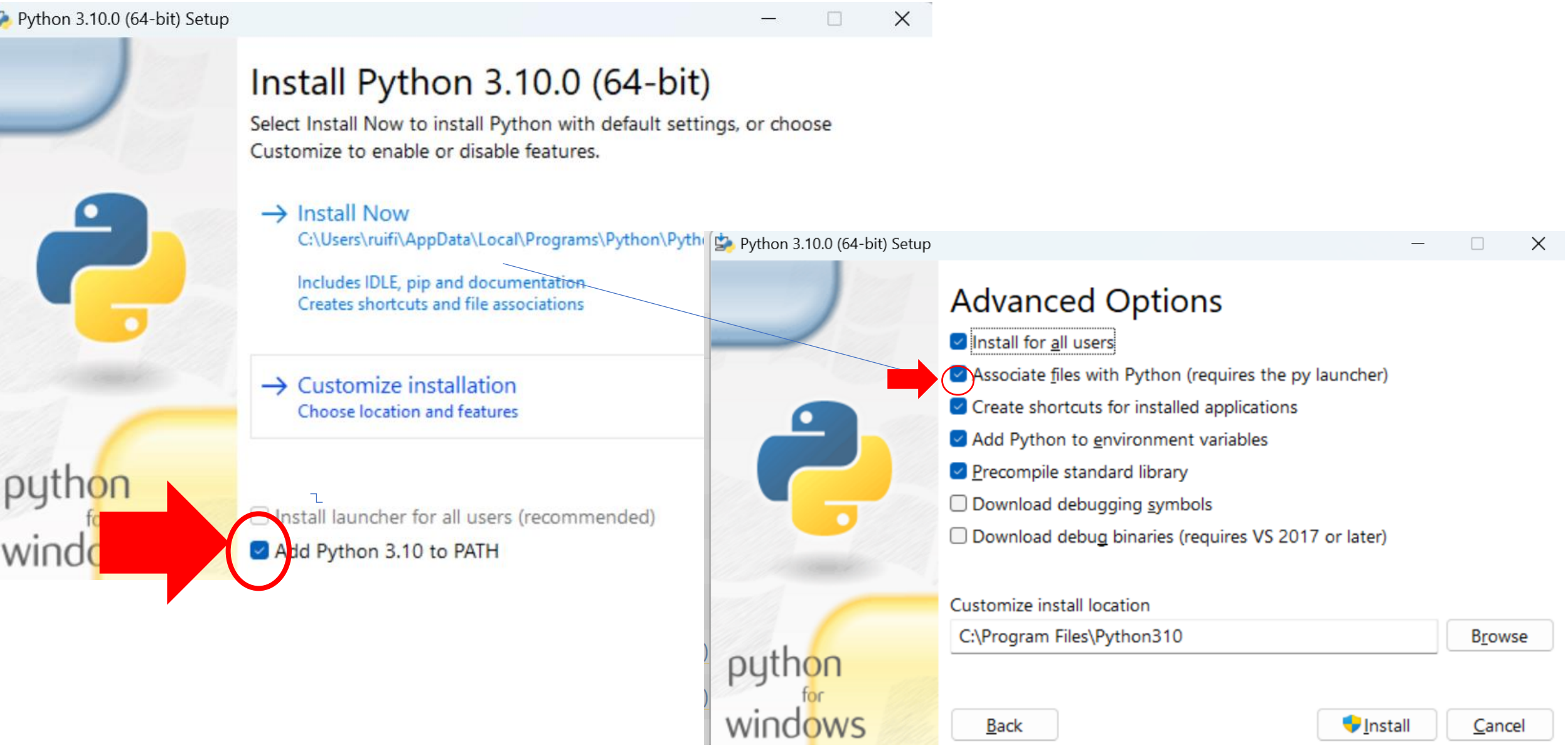
Ferramentas (para IMFVJ1)

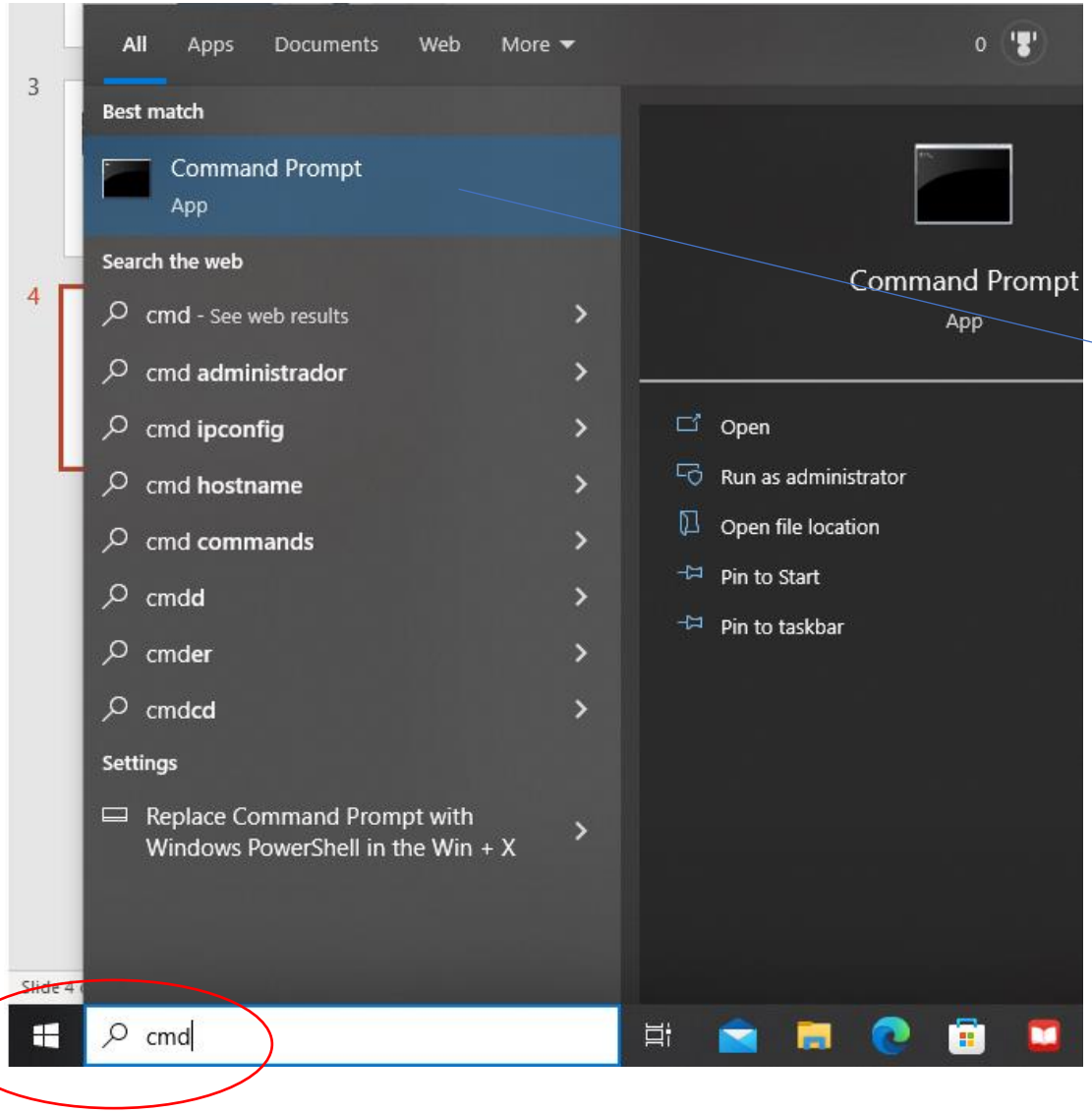
- Python 3.10
- Visual Studio Code
- P5.PyPI
- Biblioteca Python> Pygame
- Biblioteca Python> Numpy

Instalação do **Python 3.10**

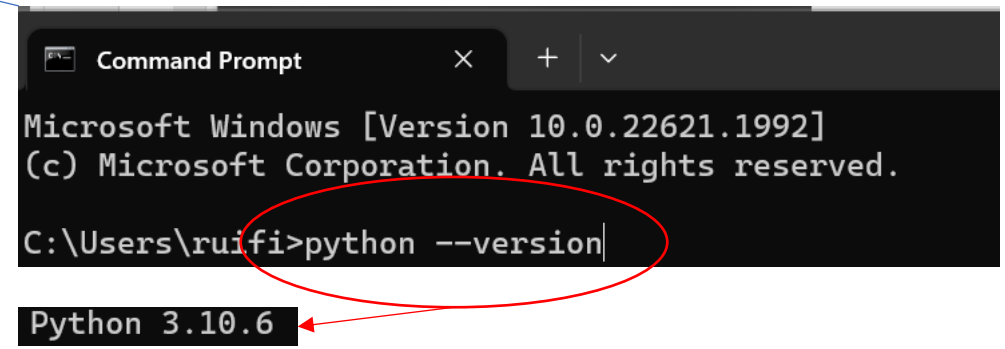
https://www.python.org/downloads/release/python-3100/




Python 3.10.0 (64-bit) Setup
Install Python 3.10.0 (64-bit)
Select Install Now to install Python with default settings, or choose
Customize to enable or disable features.
Install Now
C:\Users\ruifi\AppData\Local\Programs\Python\Pyth
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
Install launcher for all users (recommended)
Add Python 3.10 to PATH
python
for
windows
Python 3.10.0 (64-bit) Setup
Advanced Options
Install for all users
Associate files with Python (requires the py launcher)
Create shortcuts for installed applications
Add Python to environment variables
Precompile standard library
Download debugging symbols
Download debug binaries (requires VS 2017 or later)
Customize install location
C:\Program Files\Python310
Browse
Back
Install
Cancel

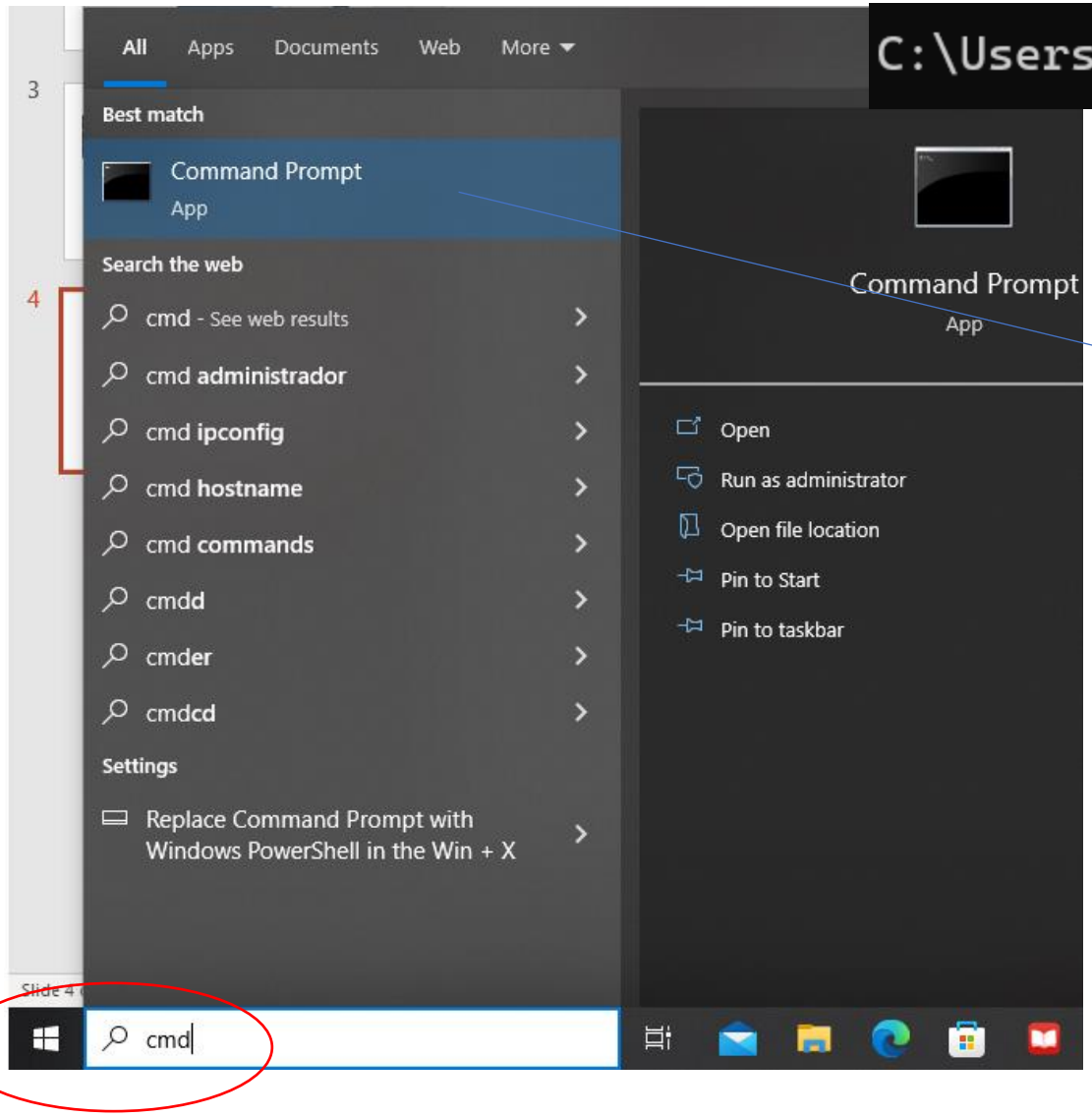


python --version



3.10

Instalação biblioteca Pygame e numpy



2- usar o comando **python -m pip install -U pygame --user**

```
C:\Users\ruifi>python -m pip install -U pygame --user
```

```
Collecting pygame
  Downloading pygame-2.1.2-cp310-cp310-win_amd64.whl (8.4 MB)
     ---------------------------------------- 8.4/8.4 MB 18.6 M/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.1.2
```

4-usar o comando **pip install numpy**

```
C:\Users\ruifi>pip install numpy
```

```
Collecting numpy
  Downloading numpy-1.23.3-cp310-cp310-win_amd64.whl (14.6 MB)
     ---------------------------------------- 14.6/14.6 MB 40.9 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-1.23.3
```

Testar

usar o comando **python -m pygame.examples.aliens**

```
C:\Users\ruifi>python -m pygame.examples.aliens
```

o output deverá ser uma janela com o jogo aliens

```
C:\Users\Administrator>python -m pygame.examples.aliens
pygame 2.1.2 (SDL 2.0.18, Python 3.10.7)
Hello from the pygame
community. https://www.pygame.org/contribute.html
```

Instalação da biblioteca P5

P5 for Python (p5.PyPI)

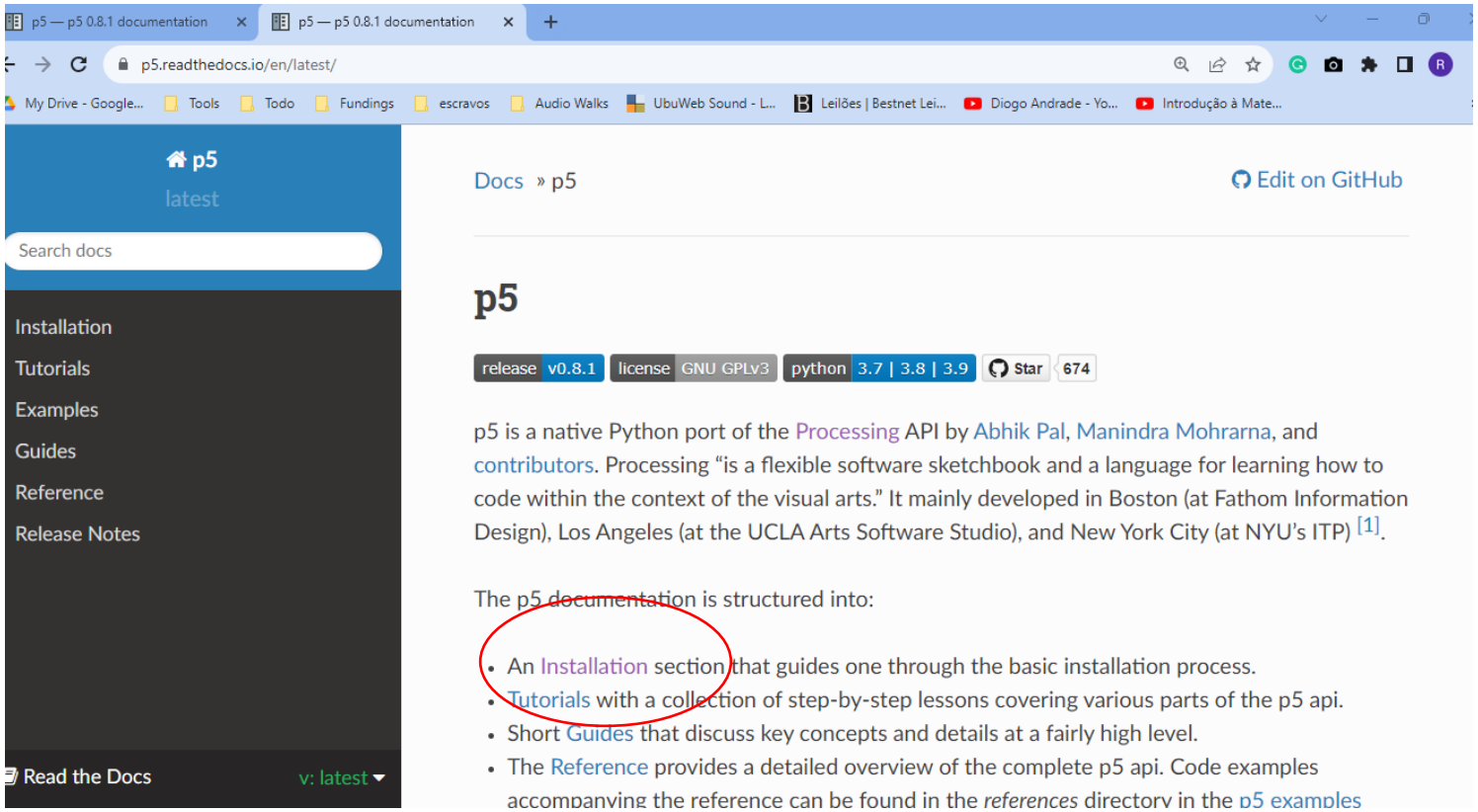
O que é o p5?

p5 is a native Python port of the Processing API by Abhik Pal, Manindra Mohrarna, and contributors. Processing “is a flexible software sketchbook and a language for learning how to code within the context of the visual arts.” It mainly developed in Boston (at Fathom Information Design), Los Angeles (at the UCLA Arts Software Studio), and New York City (at NYU’s ITP) [1].

Source: https://p5.readthedocs.io/en/latest/

pypi.org

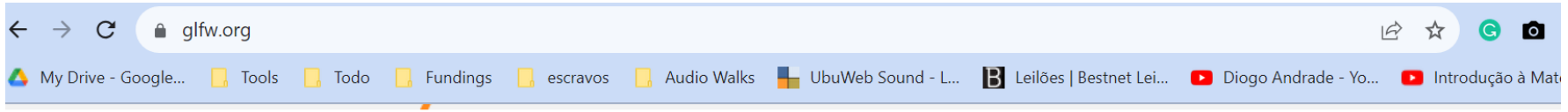
https://p5.readthedocs.io/en/latest/



https://p5.readthedocs.io/en/latest/install.html

Passo 1 .1– **Instalar GLFW** – download do binário para windows

GLFW is an Open Source, multi-platform library for OpenGL



GLFW is an Open Source, multi-platform library for OpenGL, OpenGL ES and Vulkan development on the desktop. It provides a simple API for creating windows, contexts and surfaces, receiving input and events.

GLFW is written in C and supports Windows, macOS, X11 and Wayland.

GLFW is licensed under the zlib/libpng license.

Gives you a window and OpenGL context with just two function calls

Support for OpenGL, OpenGL ES, Vulkan and related options, flags and extensions

Download GLFW 3.3.8
Released on July 22, 2022

Clone

GLFW 3.4 progres

Windows pre-compiled binaries

These packages contain the GLFW header files, documentation and release mode static libraries, DLLs and import libraries for Visual C++ 2010-2019 and the 2022 preview, MinGW-w64 and plain MinGW.

64-bit Windows binaries

32-bit Windows binaries

Binaries for Visual C++ 2010 and plain MinGW are only available in the 32-bit package.

Version 3.3.8 re

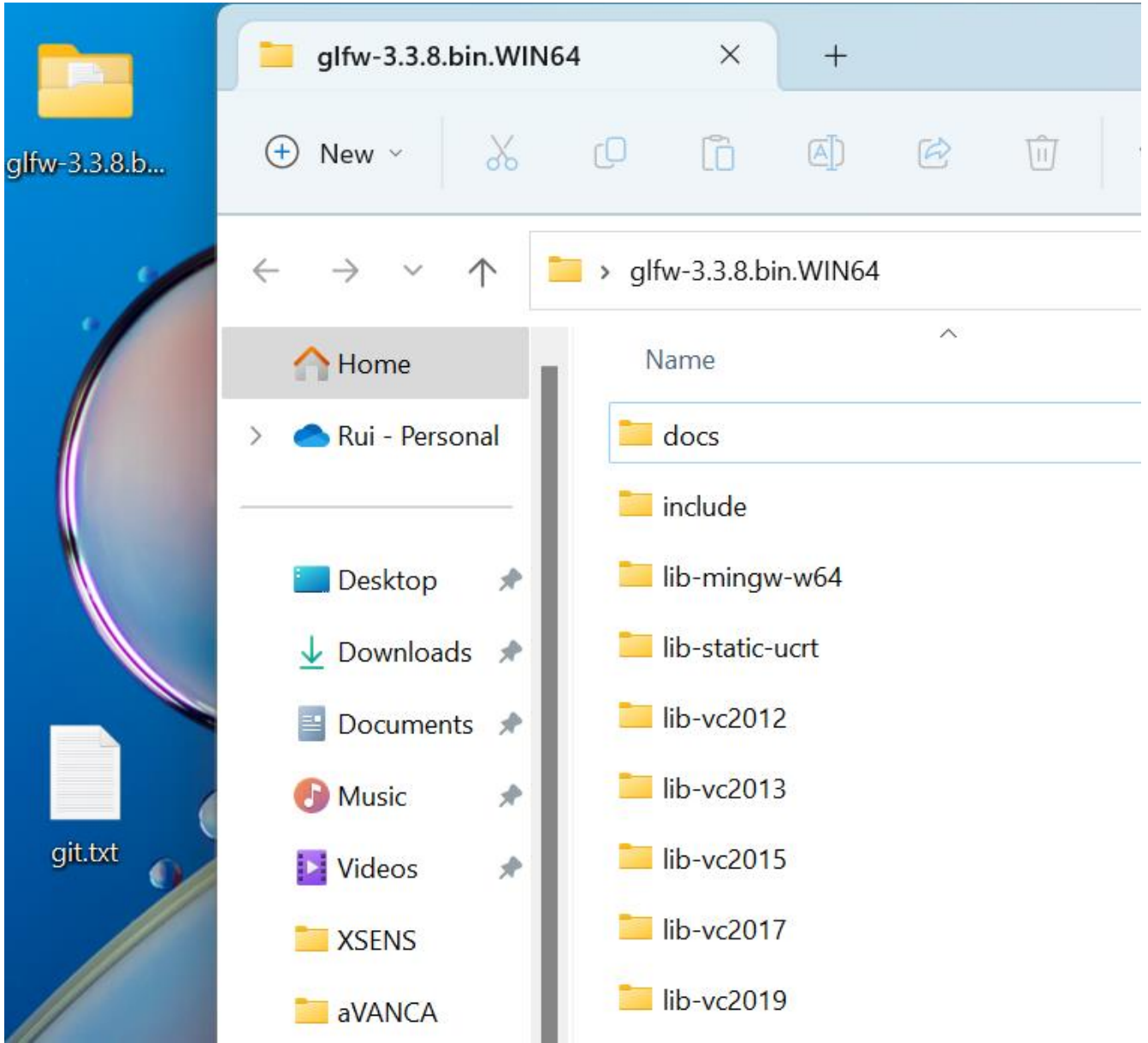
Posted on July 22, 2022

GLFW 3.3.8 is available for download.

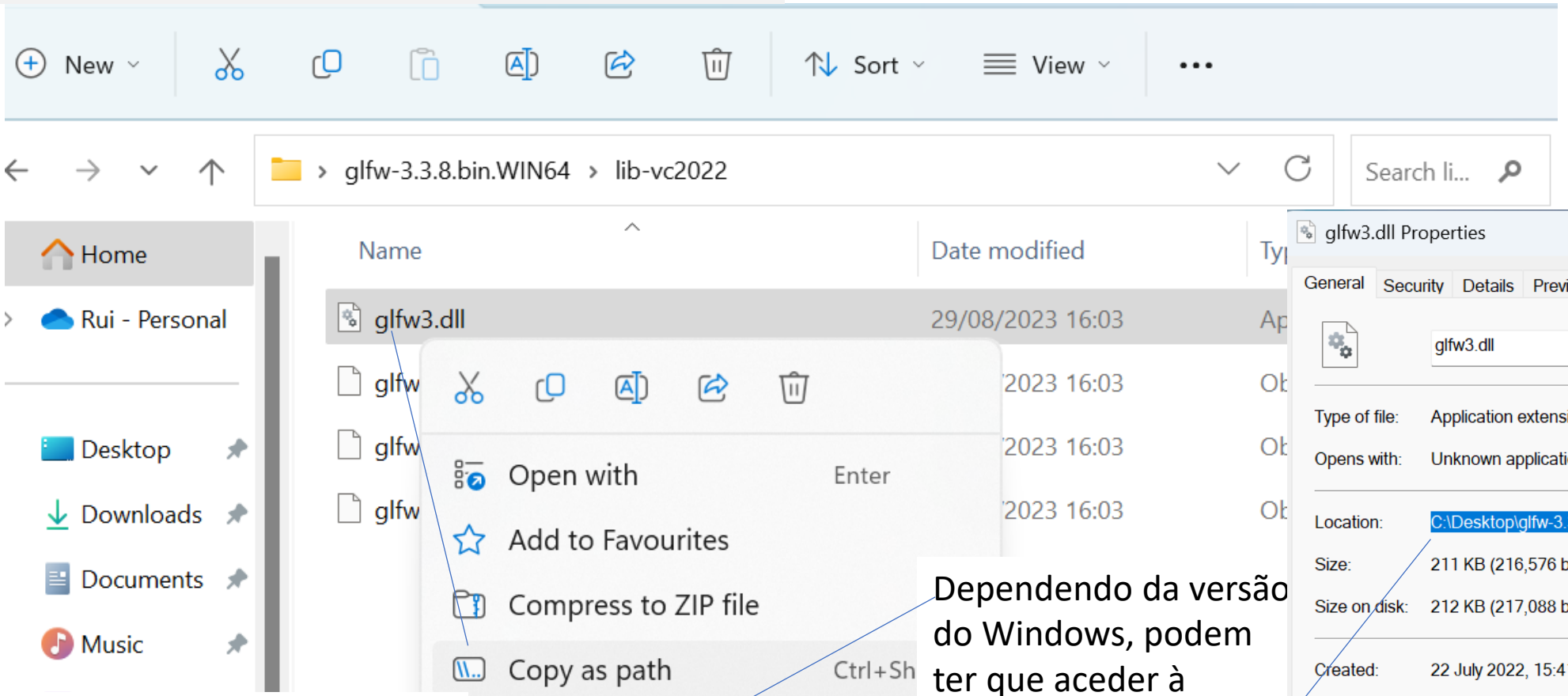
This is a bug fix release. It adds fixes for

Source: https://glfw.org

Passo 1.2 – Instalar **GLFW** - Descompactar para uma pasta no vosso sistema

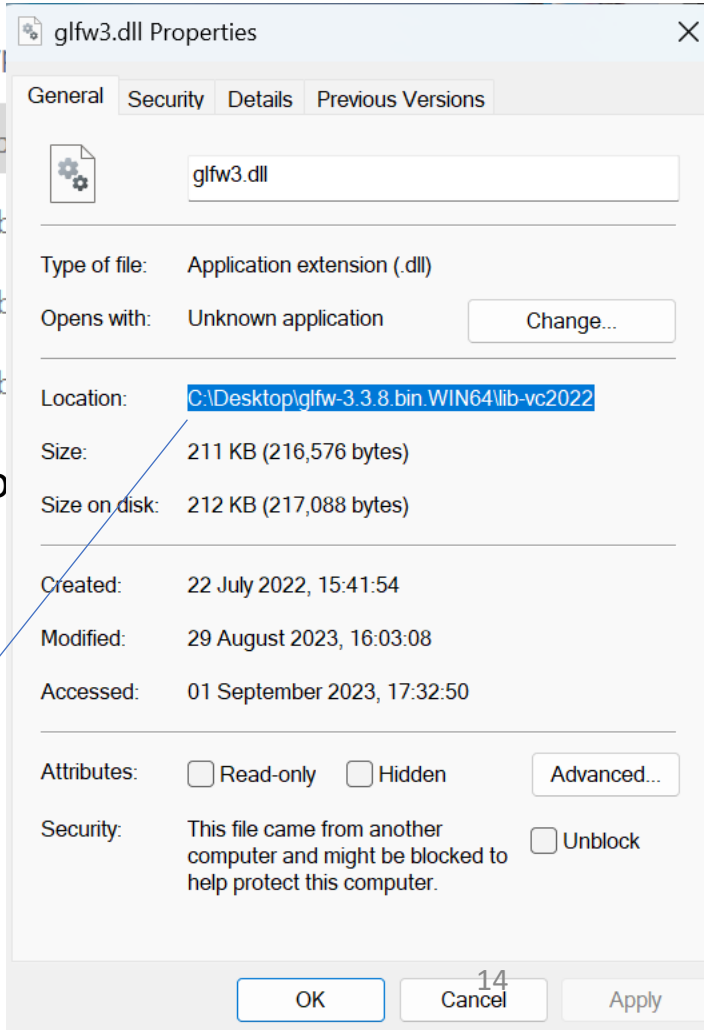


Passo 1.3– Instalar GLFW – copiar o directório de glfw3.dll

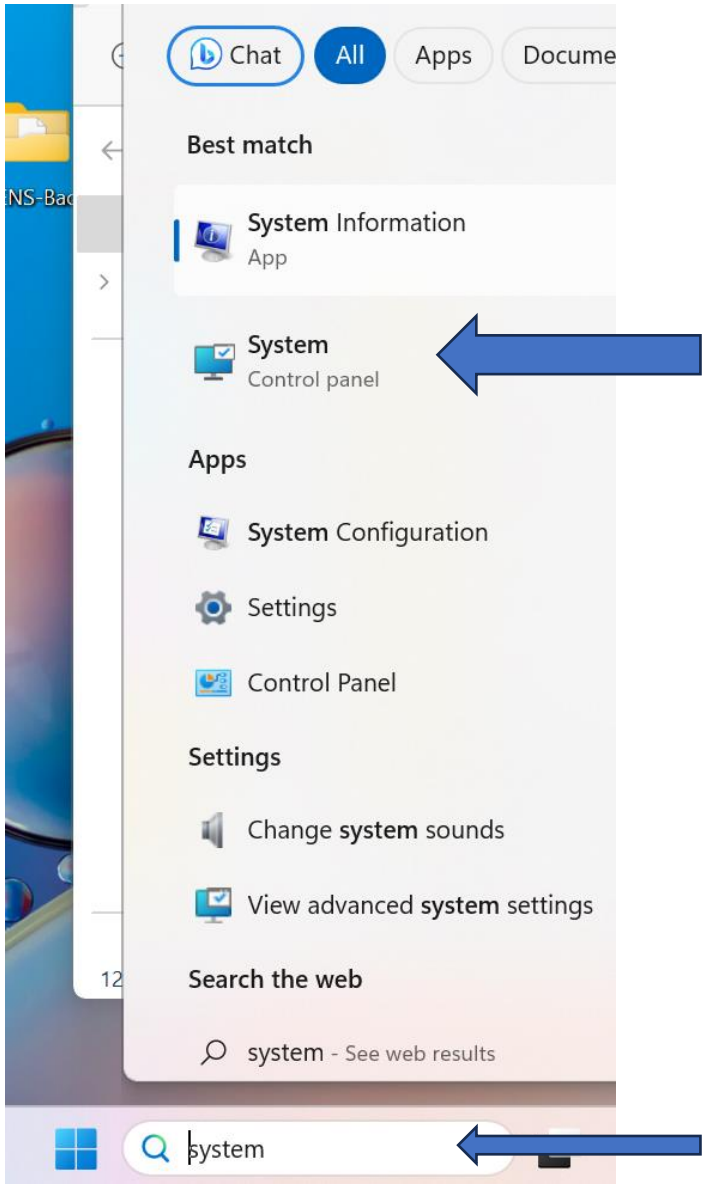


Na pasta “lib-vc2022”
Fazer “Botão direito” do rato em “glfw3.dll”
Selecionar “Copy as Path”

Dependendo da versão do Windows, podem ter que aceder à localização do ficheiro através das propriedades após clicarem com o botão direito

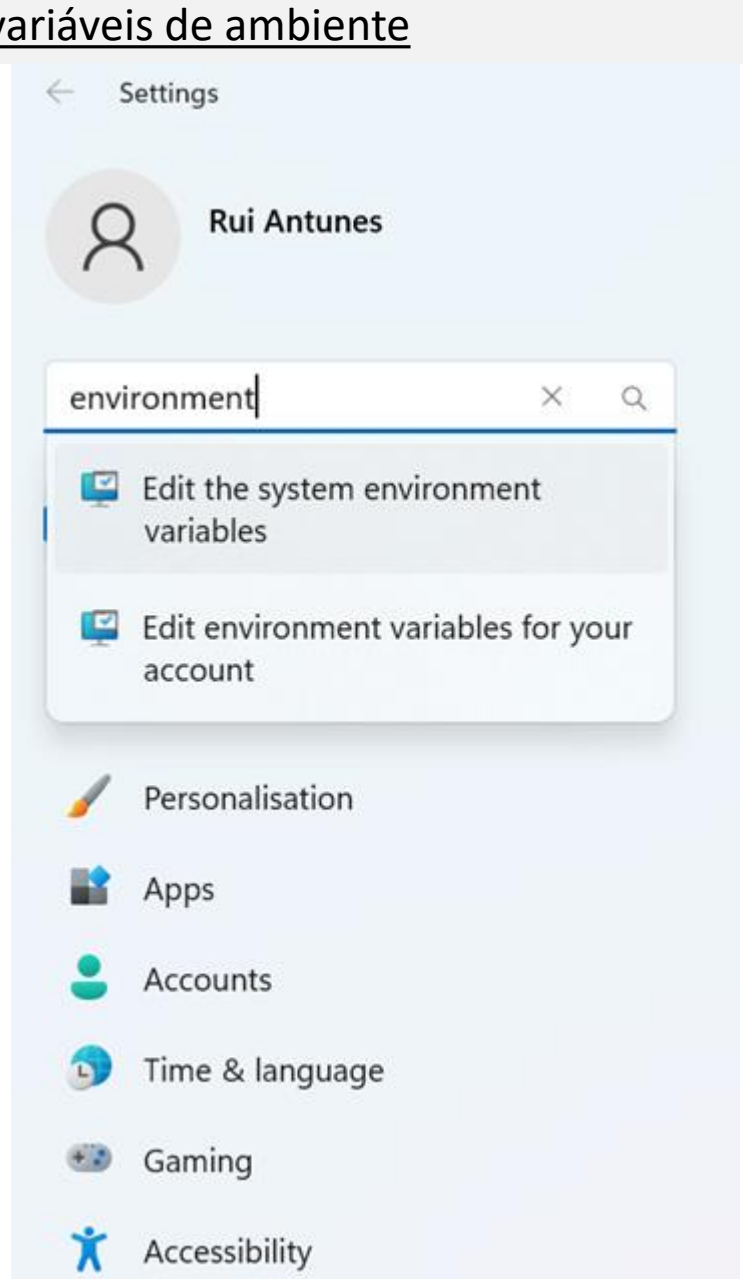


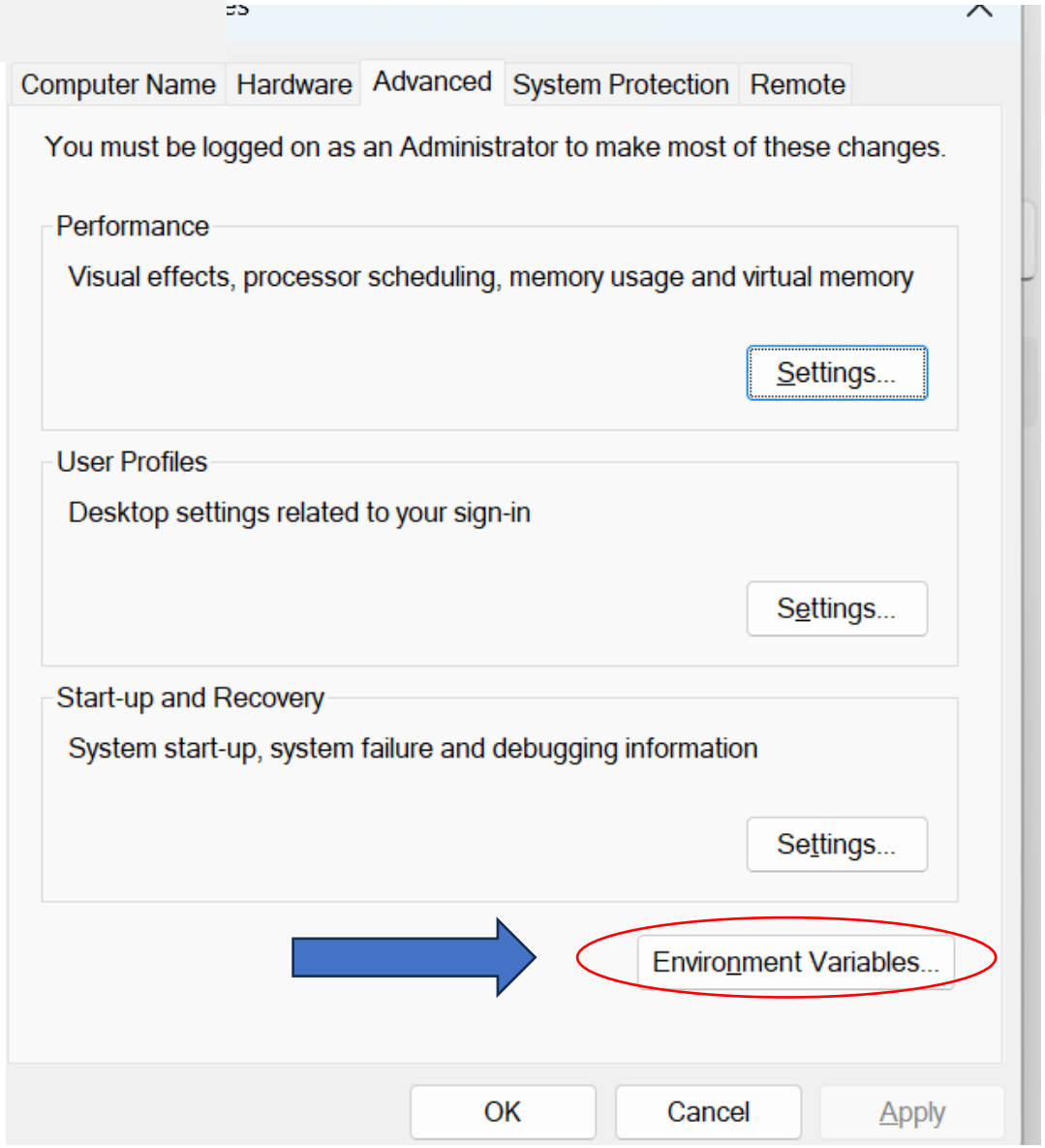
Passo 1.4– Instalar GLFW – adicionar o directório à **path** do sistema às **variáveis de ambiente**



Passo 1.4.B– Instalar GLFW – adicionar o directório à path do sistema

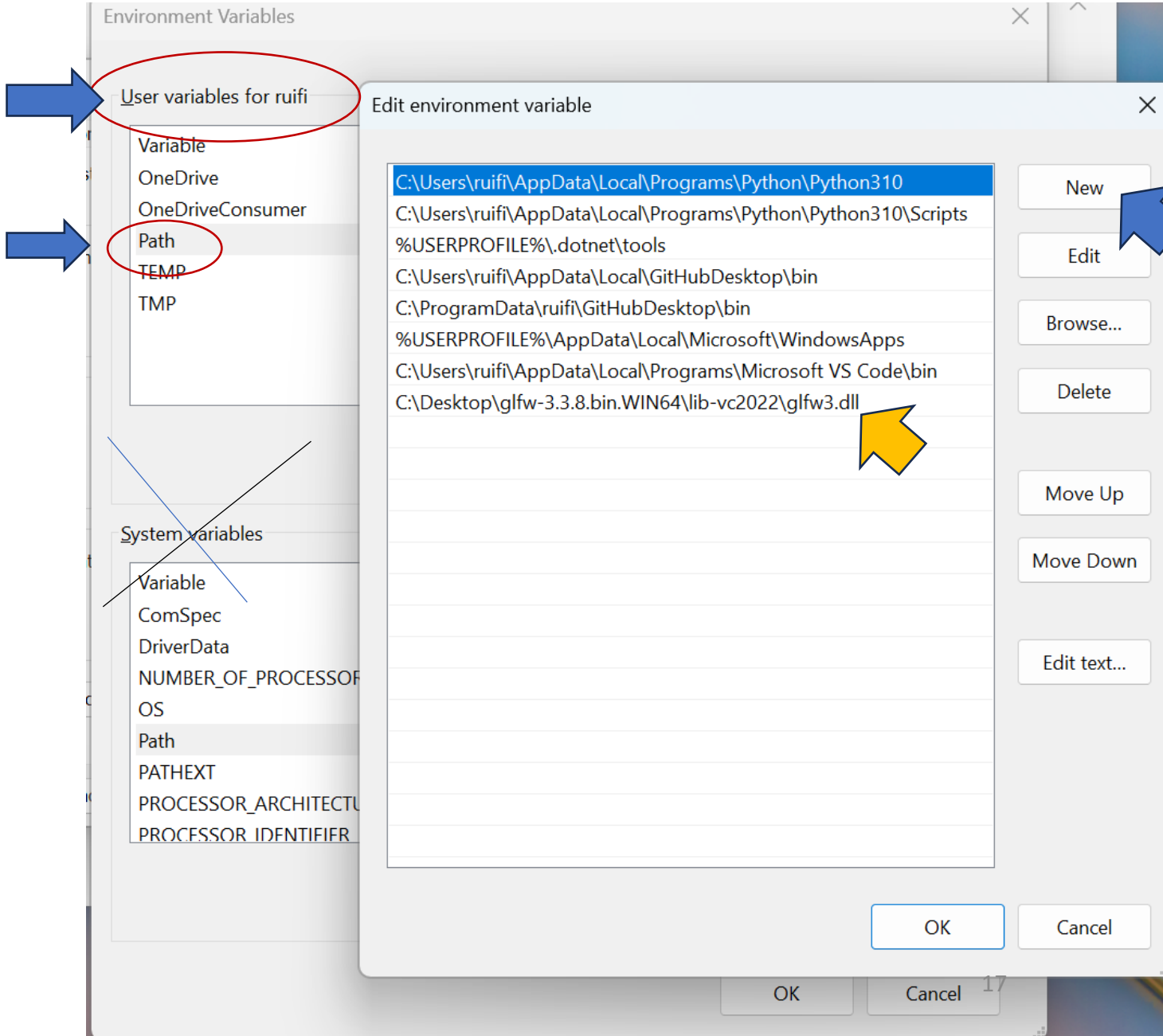
As variáveis de ambiente



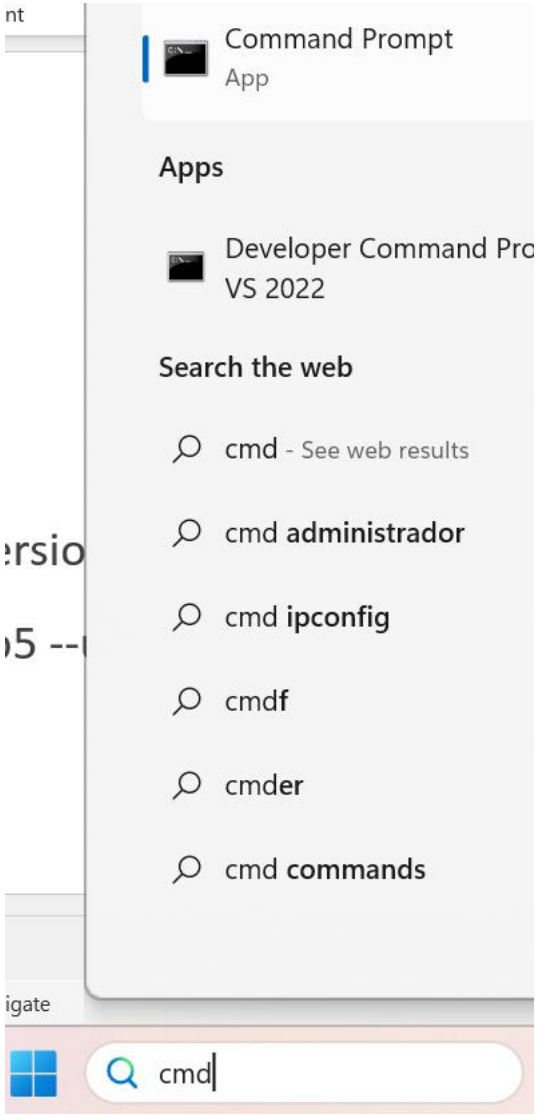


Passo 1.4.B– Instalar GLFW – adicionar o directório à path do sistema

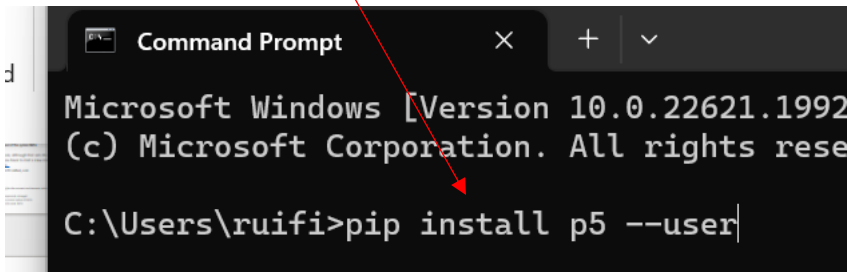
As variáveis de ambiente



Instalar o P5



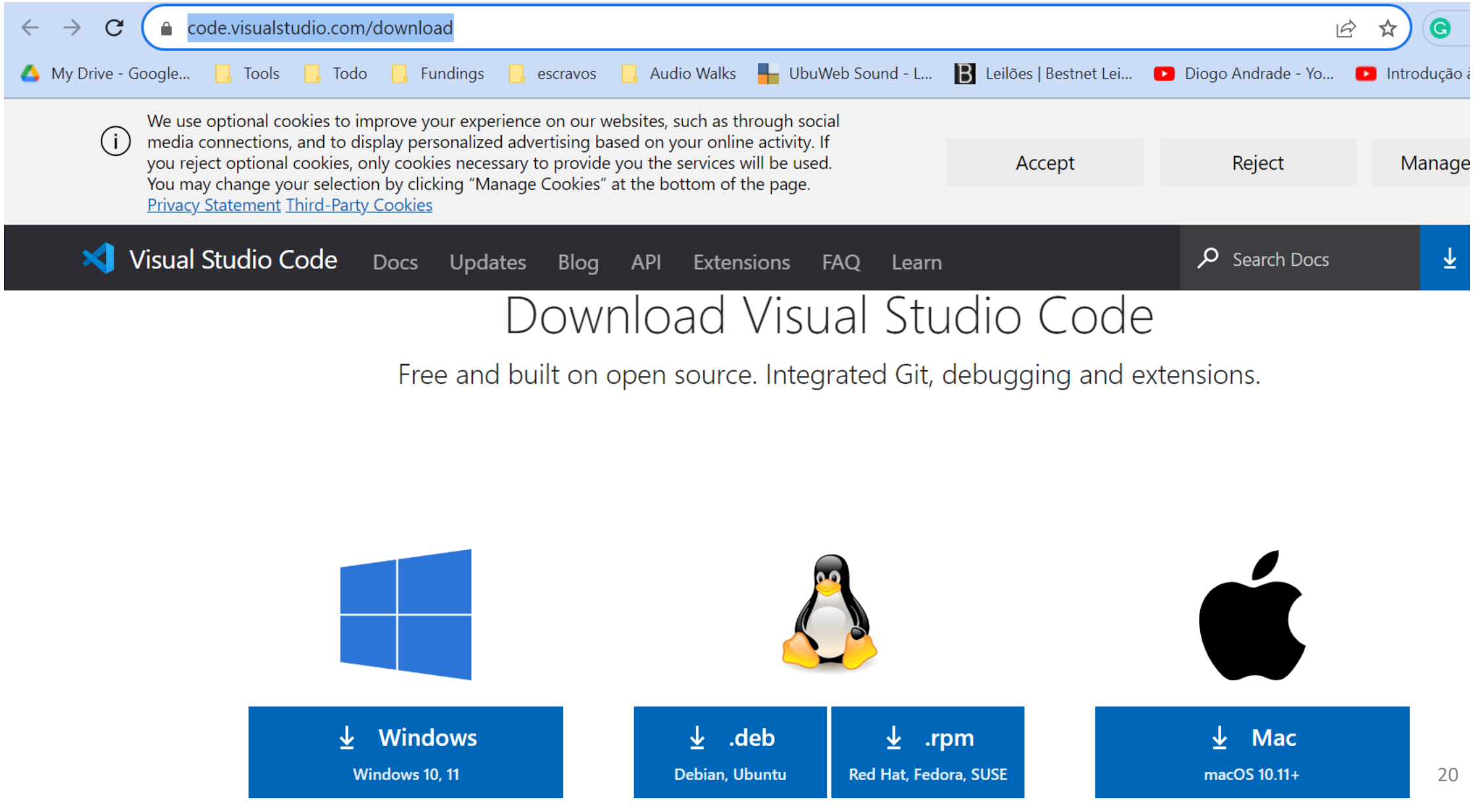
Pip install p5 --user

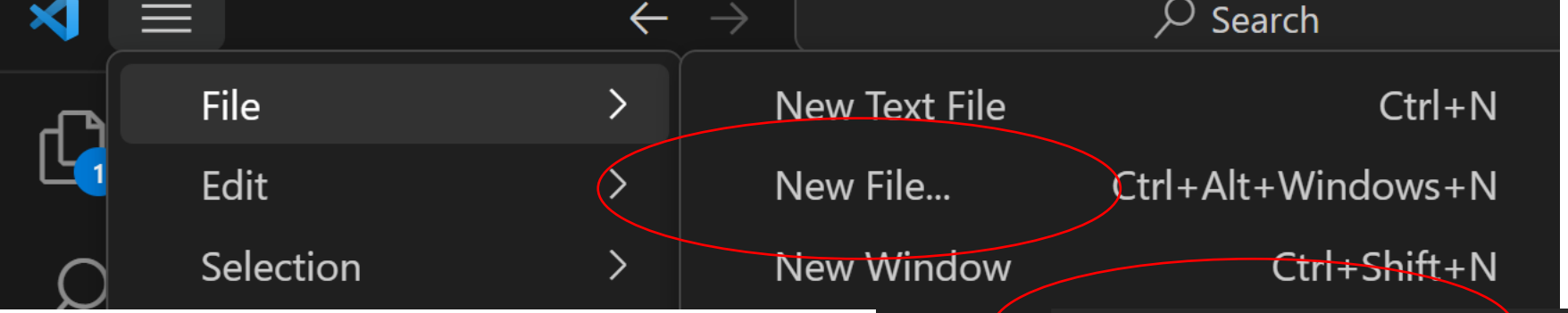


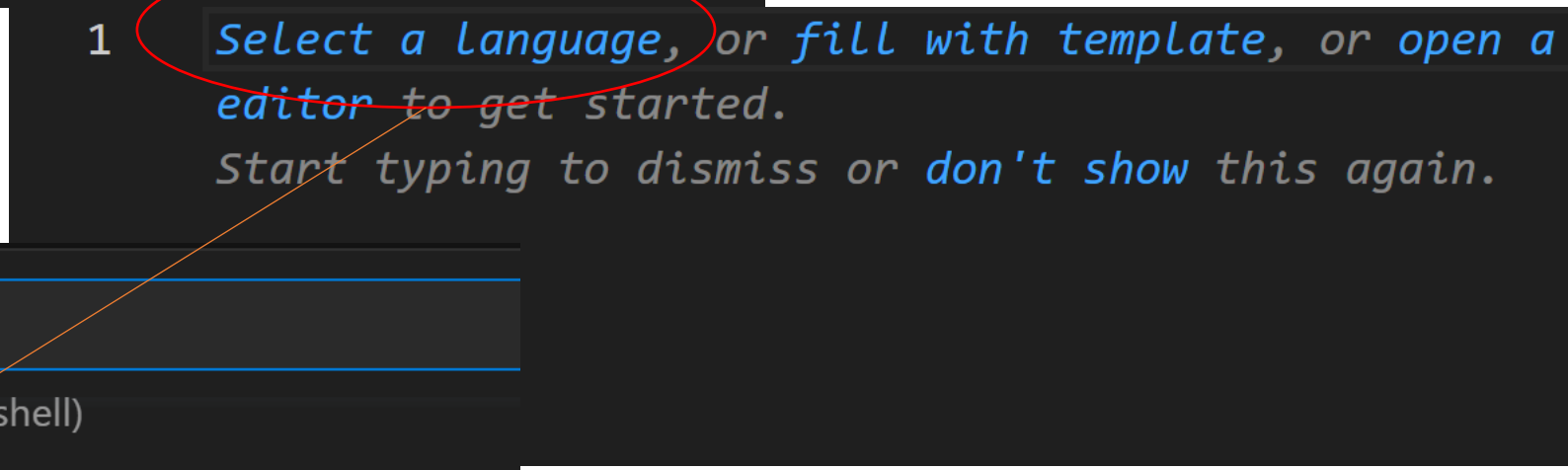
Nota:
pip defaults to installing Python packages to a system directory (such as /usr/local/lib/python3.4). This requires root access.
--user makes pip install packages in your home directory instead, which doesn't require any special privileges.

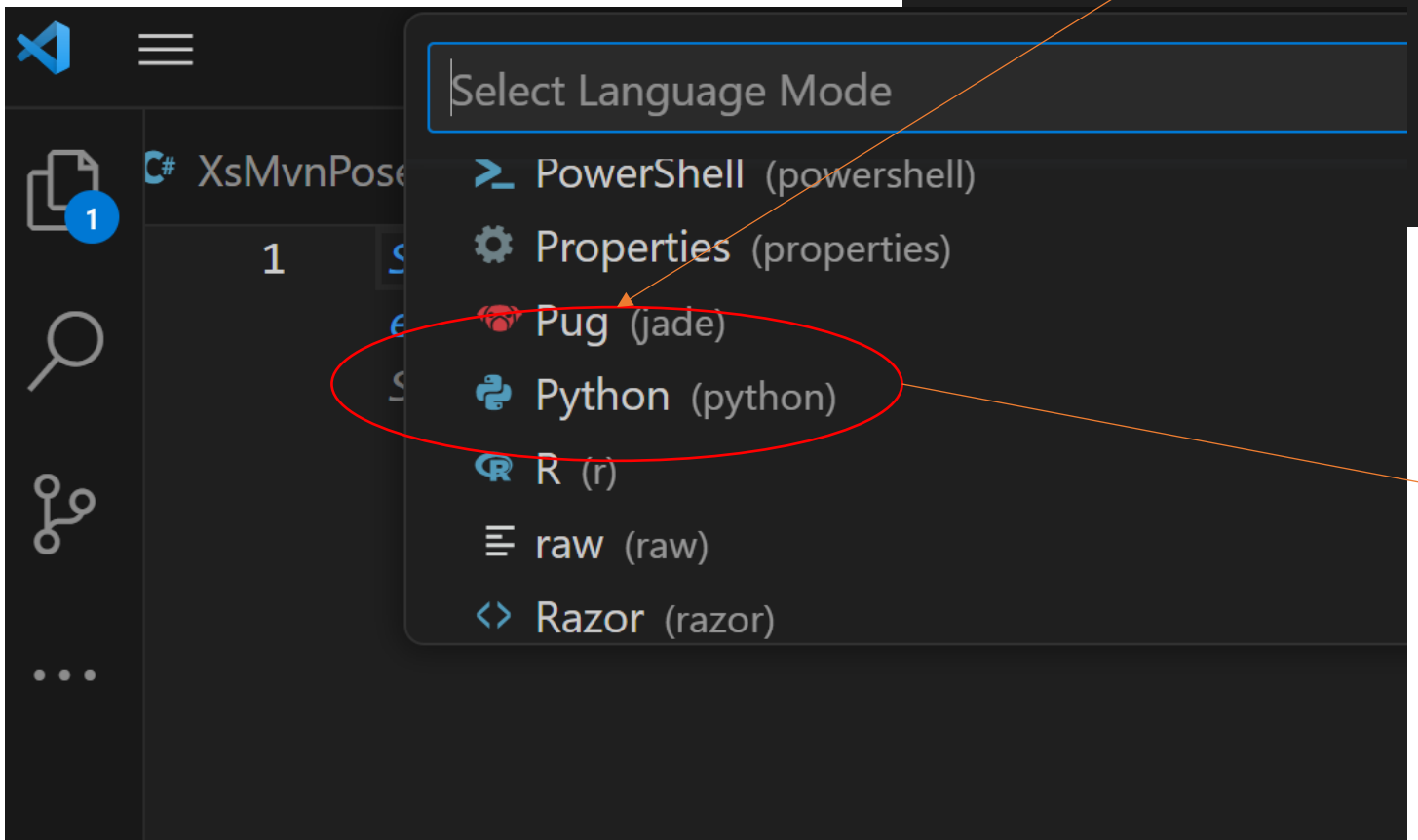
Instalação do **Visual Studio Code**

https://code.visualstudio.com/download









No canto inferior direito irá aparecer uma janela a dizer para instalar uma extensão.
Confirme a instalação.

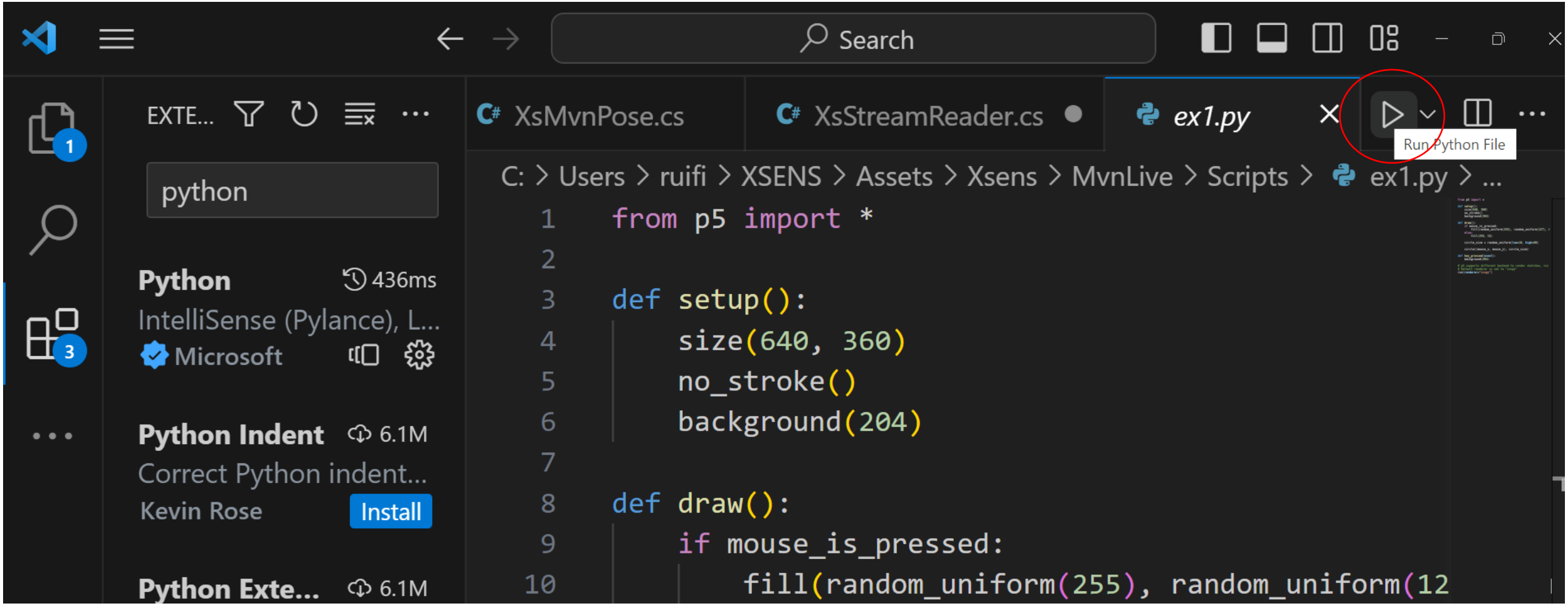
O código do programa

Copiar os seguintes comandos para o editor, depois gravar o ficheiro com a extensão “.py”

```
from p5 import *


def draw():
        background(255)
        stroke(0)
        fill(150)
        rect((50, 50), 75, 100)
if __name__ == '__main__’:
        run()
```

https://p5.readthedocs.io/en/latest/tutorials/color.html


Search
EXTE...
python
Python
436ms
IntelliSense (Pylance), L...
Microsoft
Python Indent
6.1M
Correct Python indent...
Kevin Rose
Install
Python Exte...
6.1M
XsMvnPose.cs
XsStreamReader.cs
ex1.py
Run Python File
C: > Users > ruifi > XSENS > Assets > Xsens > MvnLive > Scripts > ex1.py > ...
from p5 import *
def setup():
size(640, 360)
no_stroke()
background(204)
def draw():
if mouse_is_pressed:
fill(random_uniform(255), random_uniform(12

https://pypi.org/project/p5/

Guardar por exemplo

como ex1.py

em C:/Users

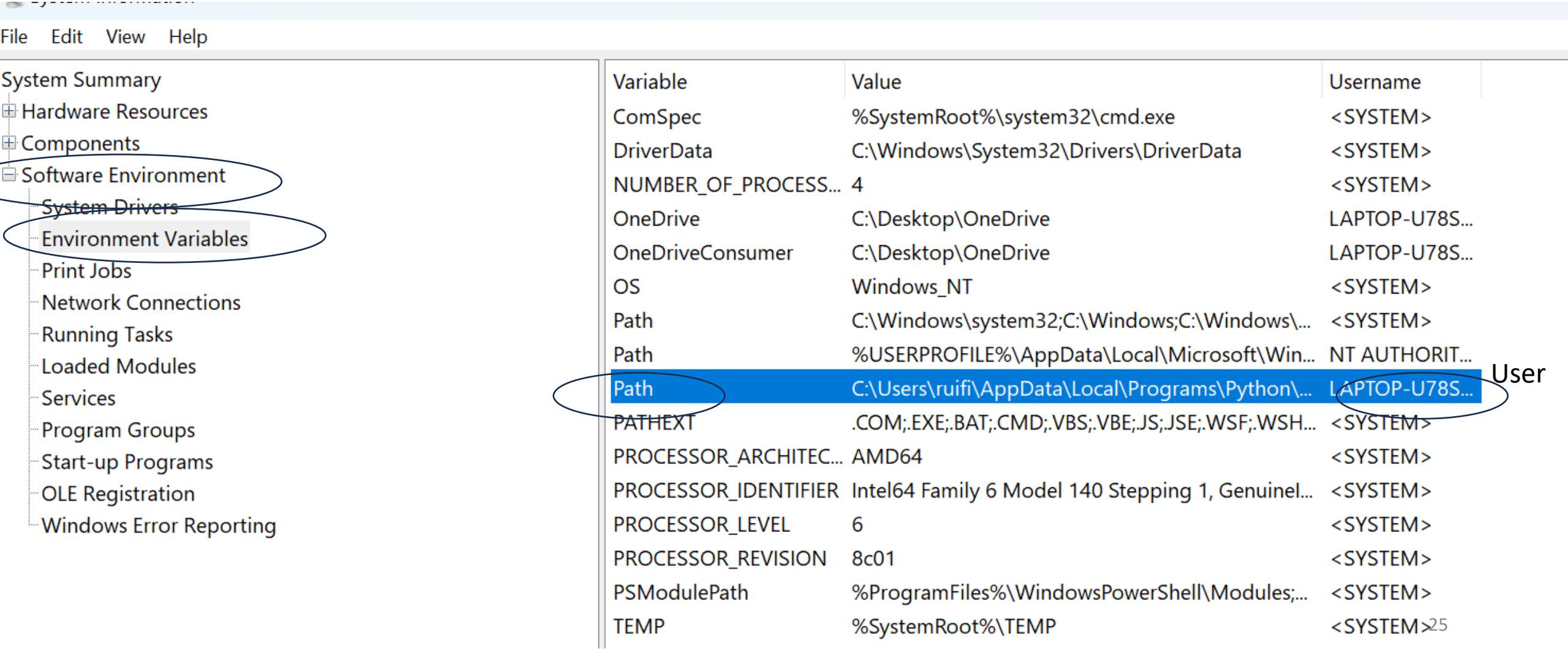
```
from p5 import *
def setup():
    size(640, 360)
    no_stroke()
    background(204)

def draw():
    if mouse_is_pressed:
        fill(random_uniform(255), random_uniform(127), random_uniform(51), 127)
    else:
        fill(255, 15)
    circle_size = random_uniform(low=10, high=80)
    circle((mouse_x, mouse_y), circle_size)

def key_pressed(event):
    background(204)
# p5 supports different backend to render sketches, viz "vispy" for both 2D and 3D sketches a
sketches # Default renderer is set to "vispy"
run(renderer="vispy")
```

Verificar as variáveis de sistema
Em Propriedades do Sistema:

A



User