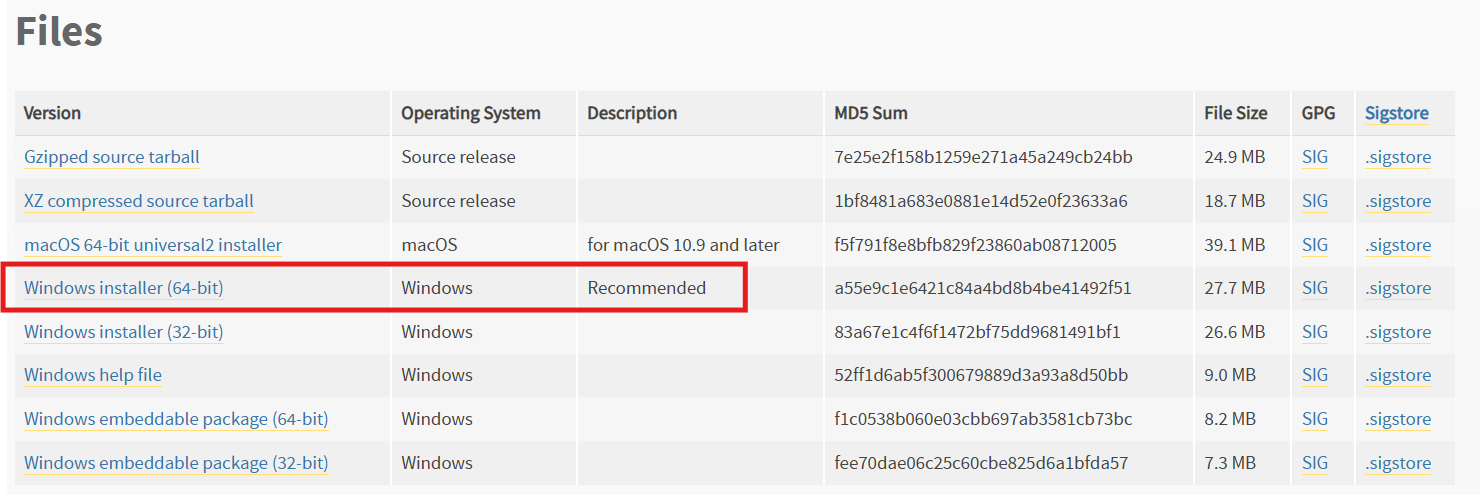
1.-Intalación del Sowtware del Kinet For Windows Developer Toolkit v1.8.0

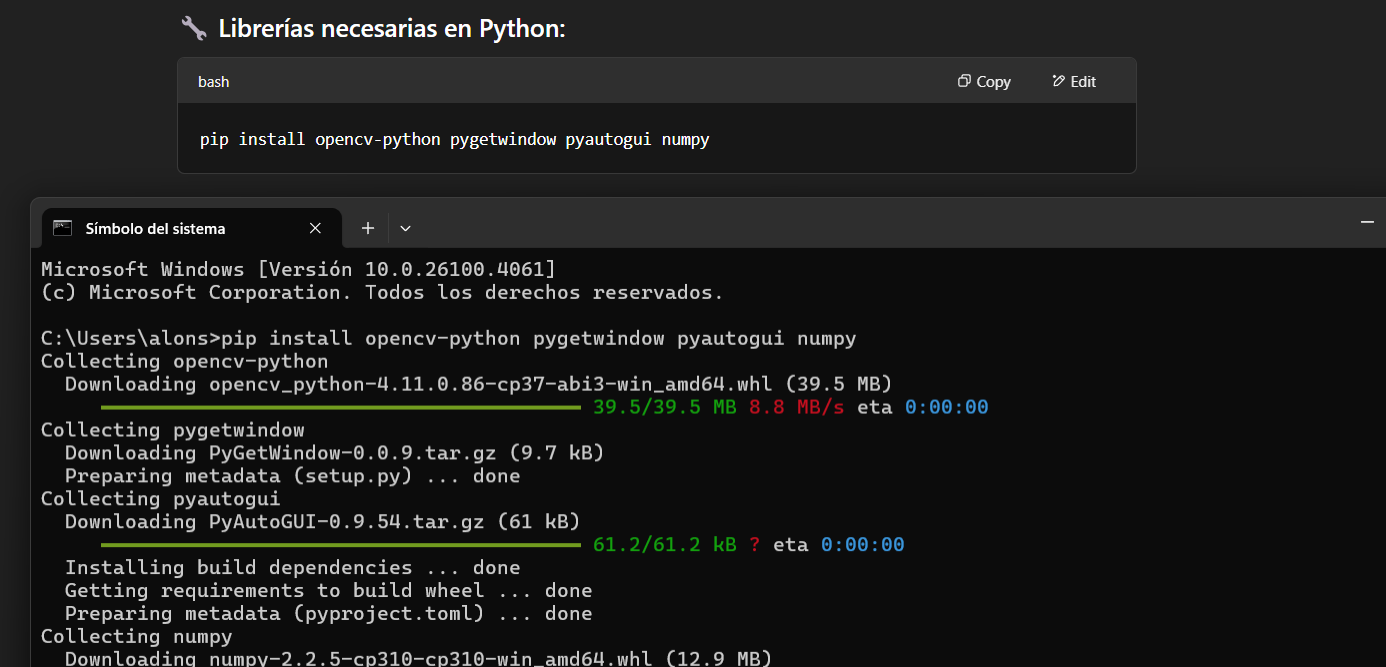
* 2.-Adaptación de Kinet, si no detecta la cámara del kinet debes de irte a configuraciones (settings). Turn the **Memory Integrity**setting**Off** if it isn’t already. You'll need to restart your computer for the changes to take effect.
* https://support.microsoft.com/en-us/windows/a-driver-can-t-load-on-this-device-8eea34e5-ff4b-16ec-870d-61a4a43b3dd5
* You can also open the Core isolation page by selecting **Start** > **Settings** > **Update & Security** > **Windows Security** > **Device Security** and then under **Core isolation**, selecting **Core isolation details**.

2.-Instalación de Python 3.10.11 desde <https://www.python.org/downloads/release/python-31011/> checar la arquitectura de tu computadora (en la actualidad ya la mayoría son x64), yo estoy con el sistema operativo de Windows



3.-Intalación de la librería de Python que nos permite visualizar el contenido de nuestra pantalla para poder detectar los puntos verdes (nodos) para poder realizar el algoritmo A\* y detectar patrones.

pip install opencv-python pygetwindow pyautogui numpy



3.-Compilar el siguiente código de Python desde tú IDE favorito, o compilandolo desde la misma consola:

import cv2

import numpy as np

import pyautogui

# captura pantalla completa

screenshot = pyautogui.screenshot()

# convierte a formato compatible con OpenCV

frame = cv2.cvtColor(np.array(screenshot), cv2.COLOR\_RGB2BGR)

# convierte a espacio de color HSV

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

# rango de verde (ajustable)

lower\_green = np.array([40, 40, 40])

upper\_green = np.array([80, 255, 255])

# máscara para detectar color verde

mask = cv2.inRange(hsv, lower\_green, upper\_green)

# detectar contornos verdes

contours, \_ = cv2.findContours(mask, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

# dibujar contornos

for cnt in contours:

area = cv2.contourArea(cnt)

if area > 100: # ignora ruido

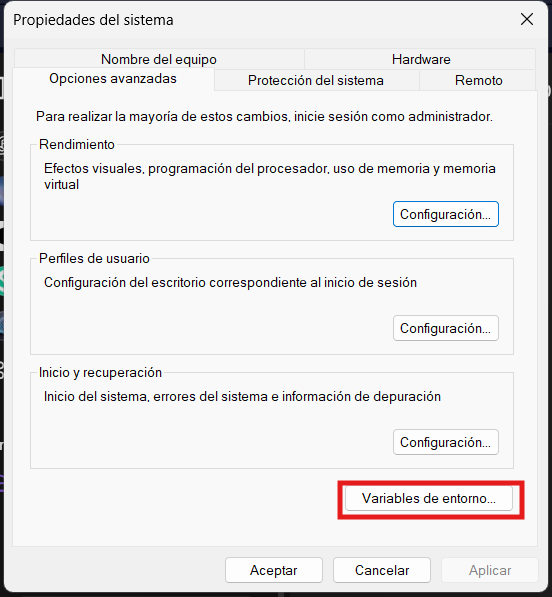
cv2.drawContours(frame, [cnt], -1, (0, 255, 0), 2)

# mostrar resultado

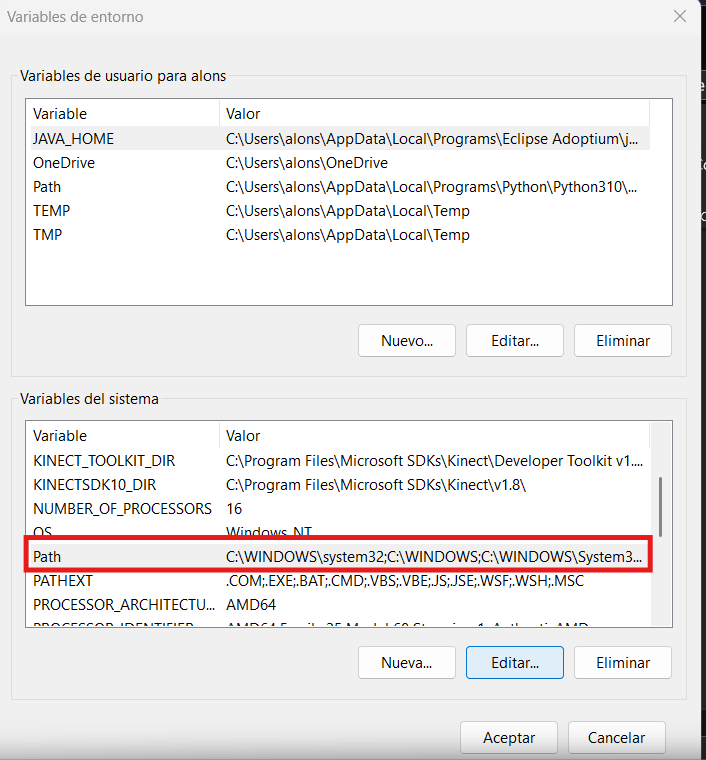
cv2.imshow("Detección de Verde", frame)

cv2.waitKey(0)

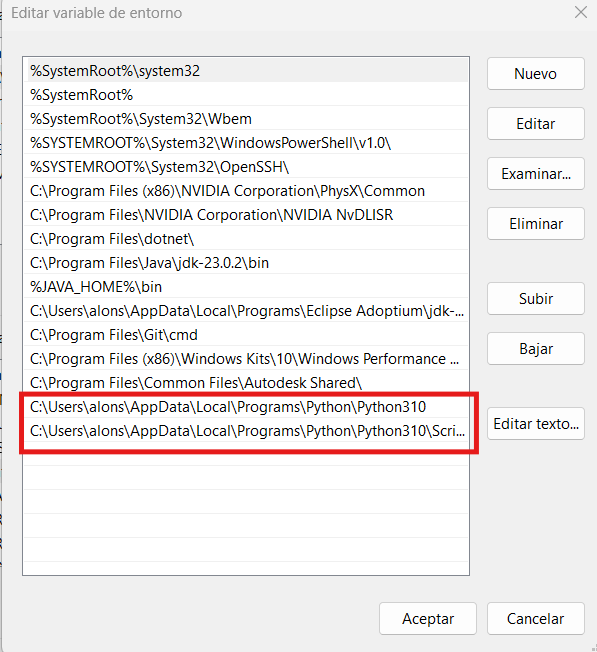
cv2.destroyAllWindows()

4.-Si les causa problema su compilación debes de “editar las variables de entorno” del sistema, así lo buscas en el buscador de window

Buscas Path en las variables del sistema y le das en editar:



Agregas la ruta de donde está tu .exe y ahí donde está tú .exe agregas también la carpeta Scripts para darle aceptar, aceptar y aceptar. Si tienes visual studio te pedirá el interprete, le das click en seleccionar interprete (en caso de que desinstalaste una versión de Python e instalaste la versión 3.10.11) y seleccionas el .exe de Python 3.10.11



5.- **✅ OBJETIVO**

* Detectar en tiempo real si hay color **verde** en pantalla.
* Extraer sus coordenadas (x, y).
* Guardarlas en una base de datos SQLite con marca de tiempo.

**🧱 PASO 1: Instalación de dependencias**

Abre tu terminal o consola y ejecuta:

bash

CopyEdit

pip install opencv-python pyautogui numpy sqlite3