



ISNTITUTO TECNOLOGICO SUPERIOR DE VENUSTIANO CARRANZA

INGENERIA EN
SISTEMAS COMPUTACIONALES

INVESTIGACION

ACTIVIDAD

Reporte de instalación y configuración

ALUMNO:

GILBERTO URBINA LARIOS

DOCENTE

GUILLERMO GARCIA GARCIA

Podemos ejecutar el programa de dos formas distintas, instalar la versión compilada directo en Windows o usarla en modo de desarrollo, esta última es útil para modificar o cuando mejoramos el entrenamiento con más épocas

A continuación, veremos la forma de instalación en Windows que suele ser sencilla

Requisitos:

Sistema operativo Windows 11

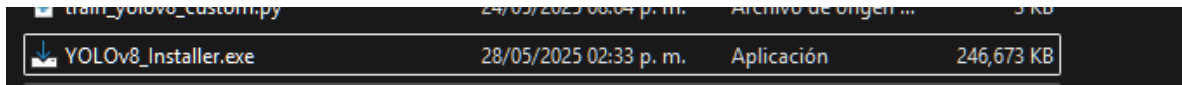
Programa WinRAR

1. Descargar el instalador

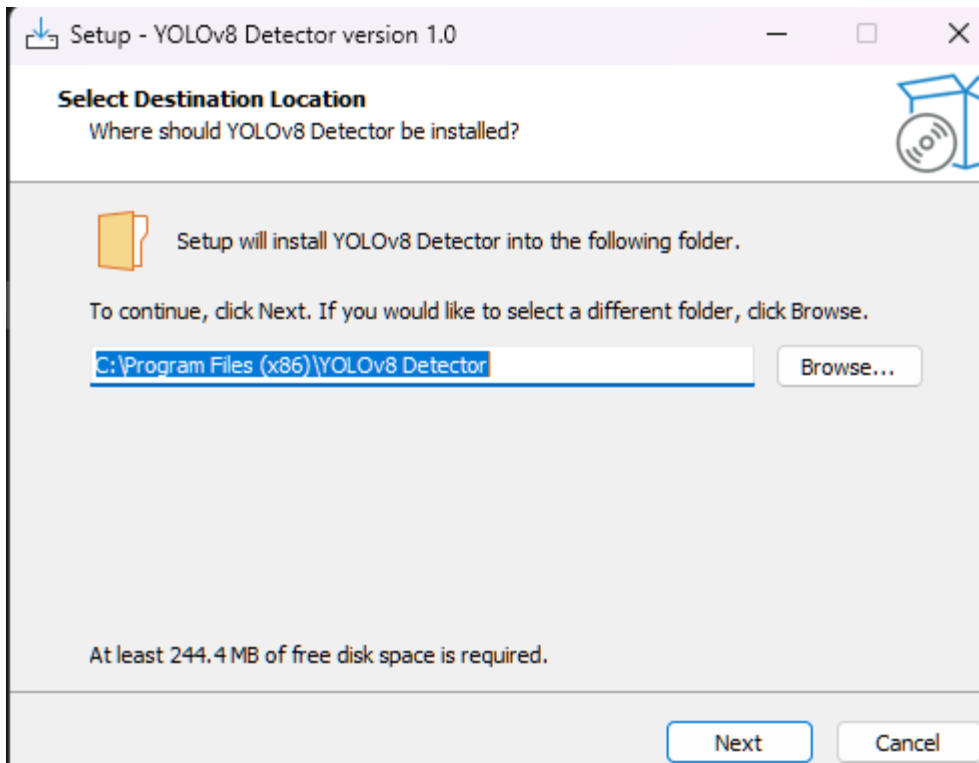
Link: https://drive.google.com/file/d/1zNTolNuoeiXYfqKkpCfz-nGngFu2gXPN/view?usp=drive_link

2. Proceso de instalación

Una vez descargado el instalador

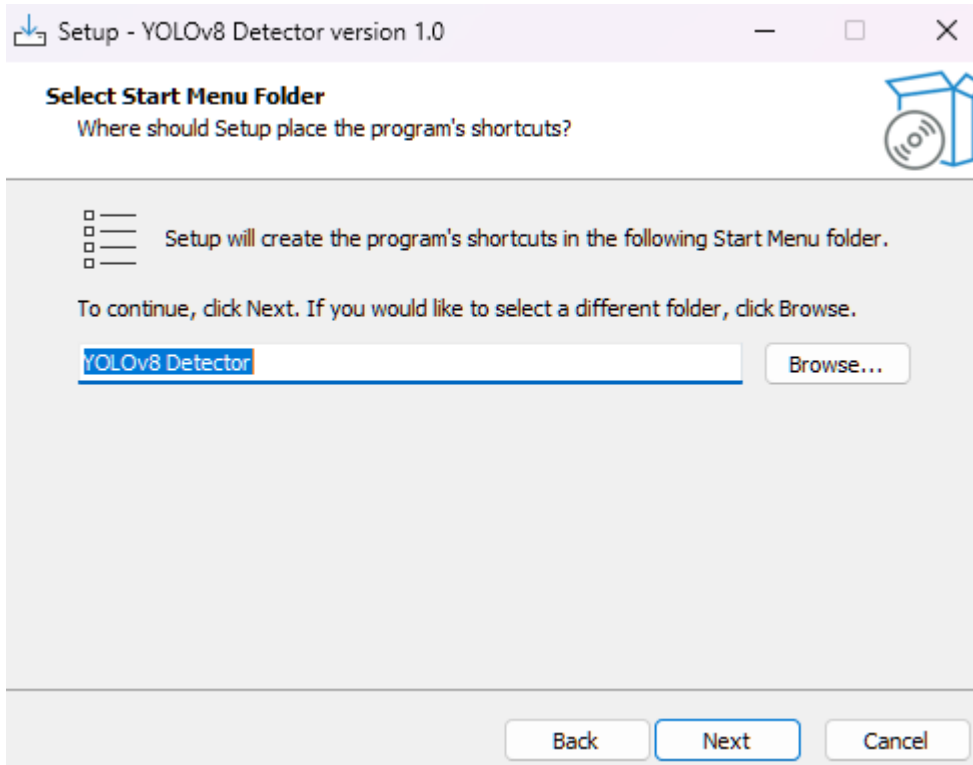


Damos doble clic sobre él y esperamos unos segundos a que se inicie

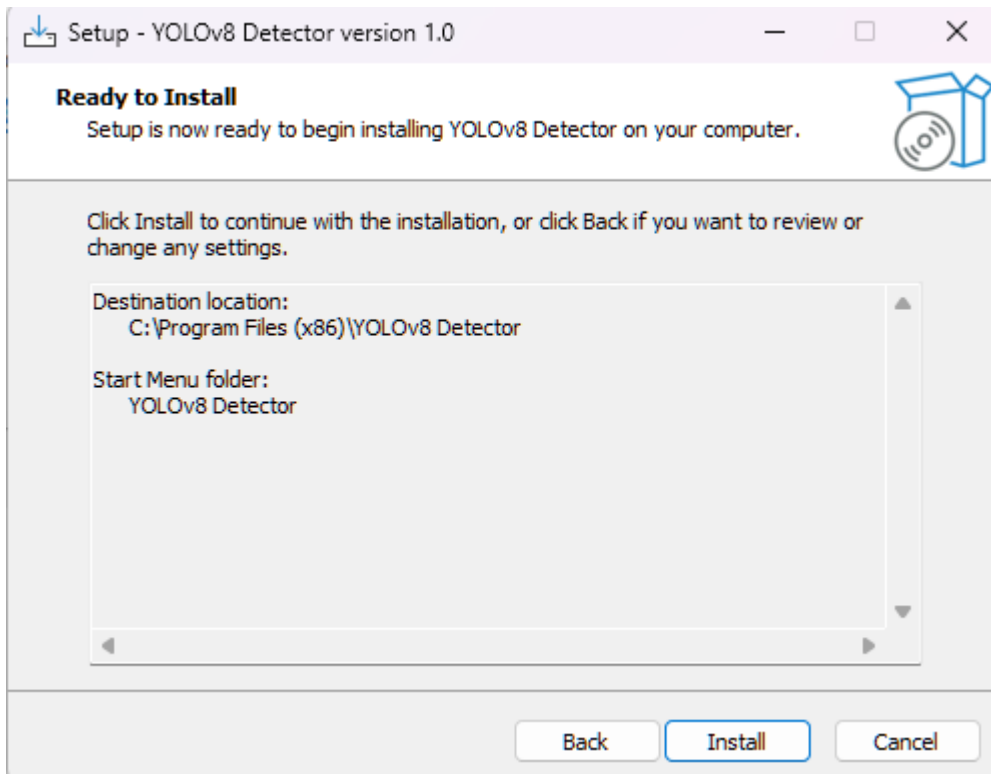


Nos preguntara la ruta de instalación, recomendando dejarla por defecto

Aquí nos preguntara el nombre de carpetas, igual recomendando dejarla por defecto



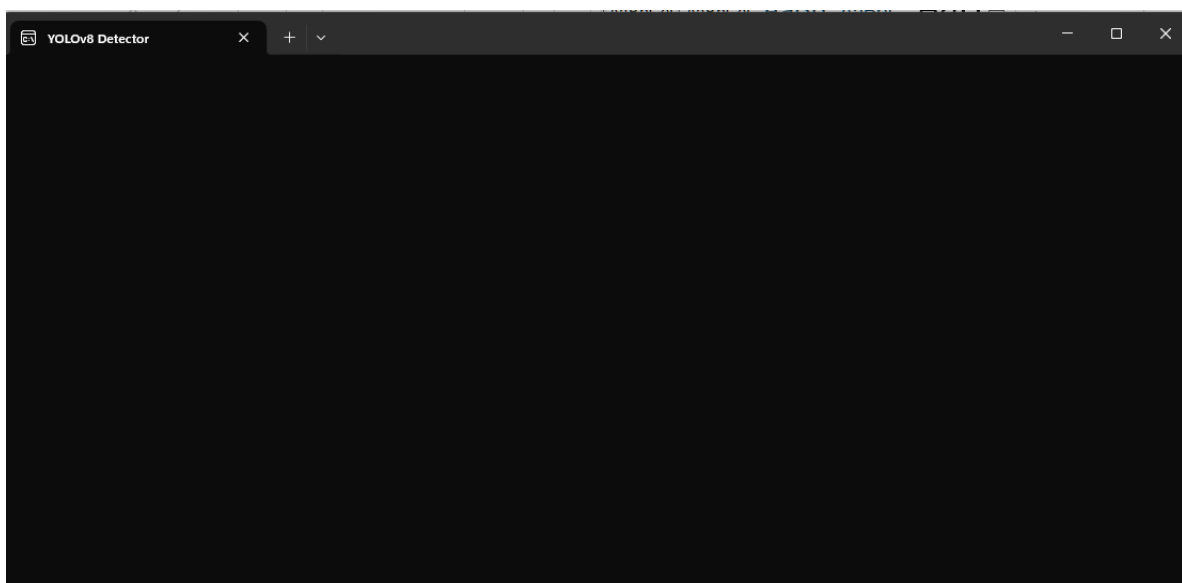
Por último, damos clic en instalar



Para probar el programa debemos buscarlo desde la barra de tareas, debe aparecer con el nombre YOLOv8 detector

3. Uso del programa

Al inicio de la ejecución nos saldrá una ventana CM



Cuando termine de cargar nos saldrá la ventana principal



El botón (Modelo 1: (60 épocas))

Sirve para seleccionar el entrenamiento deseado de épocas, recomendando usar la de 150 para mayor precisión

Seleccionamos una imagen desde el botón de selección y esperamos unos segundos a que cargue



Forma de desarrollo

Para poder ejecutar el proyecto correctamente debemos tener los siguientes requisitos

Requisitos:

- Python 3.13
- Pycharm
- Inno setup compiler

1. Descargar todo el proyecto

Link: https://drive.google.com/file/d/1zxSyWpFSUtMyB7VaPe4JG-YAFJRXLNWA/view?usp=drive_link

2. Proceso de abertura del proyecto

Al descargar el proyecto nos dejara un archivo .rar lo cual debemos descomprimirlo.

La carpeta debe incluir todo esto

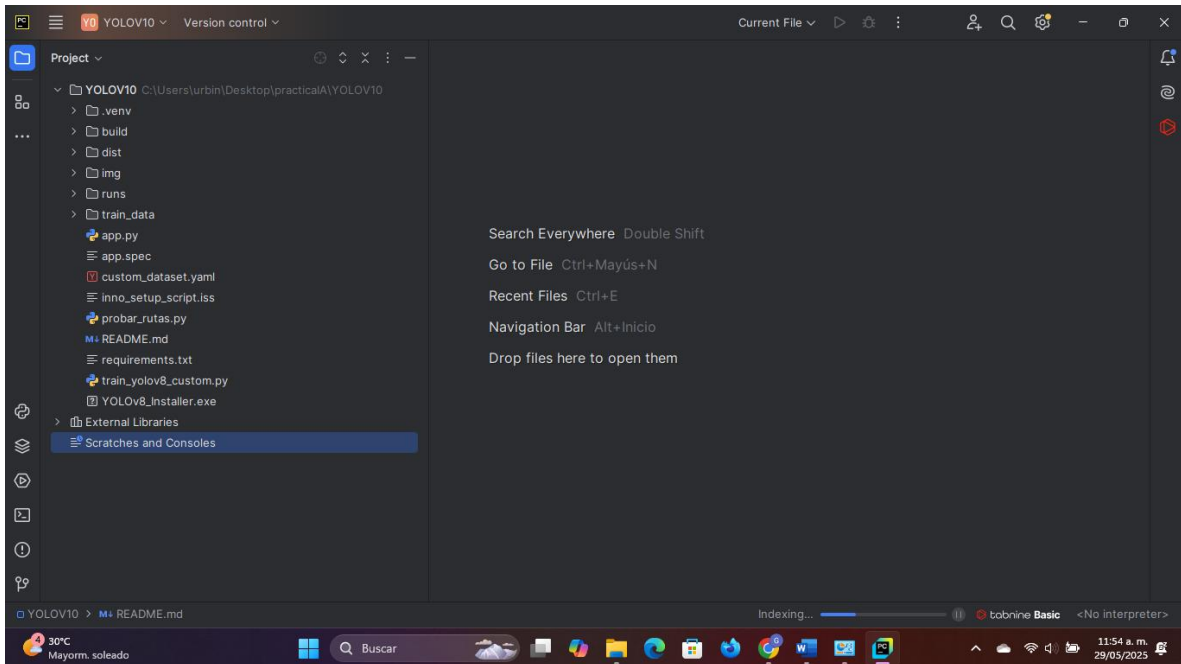
Nombre	Fecha de modificación	Tipo	Tamaño
.env	13/05/2025 02:20 p. m.	Carpeta de archivos	
build	25/05/2025 03:19 p. m.	Carpeta de archivos	
dist	28/05/2025 02:01 p. m.	Carpeta de archivos	
img	28/05/2025 04:45 p. m.	Carpeta de archivos	
runs	13/05/2025 03:30 p. m.	Carpeta de archivos	
train_data	12/05/2025 05:18 p. m.	Carpeta de archivos	
app.py	28/05/2025 01:57 p. m.	Archivo de origen ...	3 KB
app.spec	28/05/2025 01:58 p. m.	Archivo SPEC	1 KB
custom_dataset.yaml	13/05/2025 05:26 p. m.	Archivo de origen ...	1 KB
inno_setup_script.iss	28/05/2025 01:52 p. m.	Inno Setup Script	1 KB
probar_rutas.py	13/05/2025 05:12 p. m.	Archivo de origen ...	1 KB
README.md	28/05/2025 04:50 p. m.	Archivo de origen ...	3 KB
requirements.txt	28/05/2025 02:57 p. m.	Documento de tex...	2 KB
train_yolov8_custom.py	24/05/2025 08:04 p. m.	Archivo de origen ...	3 KB
YOLOv8_Installer.exe	28/05/2025 02:33 p. m.	Aplicación	246,673 KB

Aquí se encuentra todo el proyecto mas el instalador

3. Probar en entorno de Python

La carpeta ya incluye todo el entorno listo para su ejecución, sin embargo, en caso de algún error se incluye el archivo de requerimientos para instalar las librerías faltantes oh problemáticas

Abrimos la carpeta en el programa de PyCharm

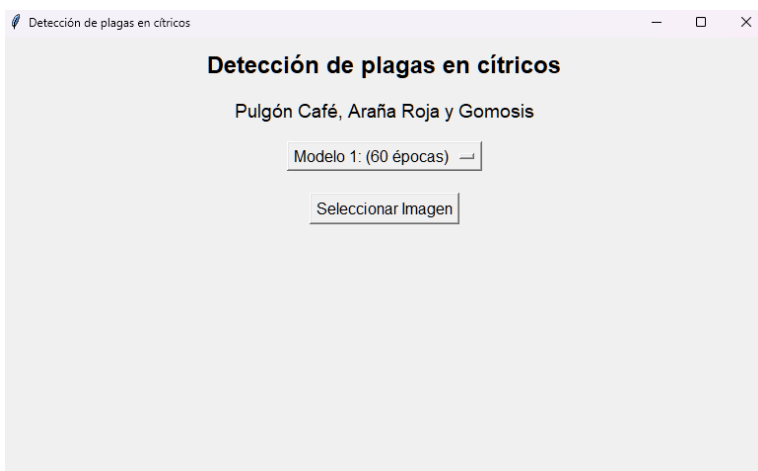


Para empezar a probar ejecutamos el siguiente comando en la terminal,

NOTA: asegúrate de estar dentro de la carpeta del proyecto

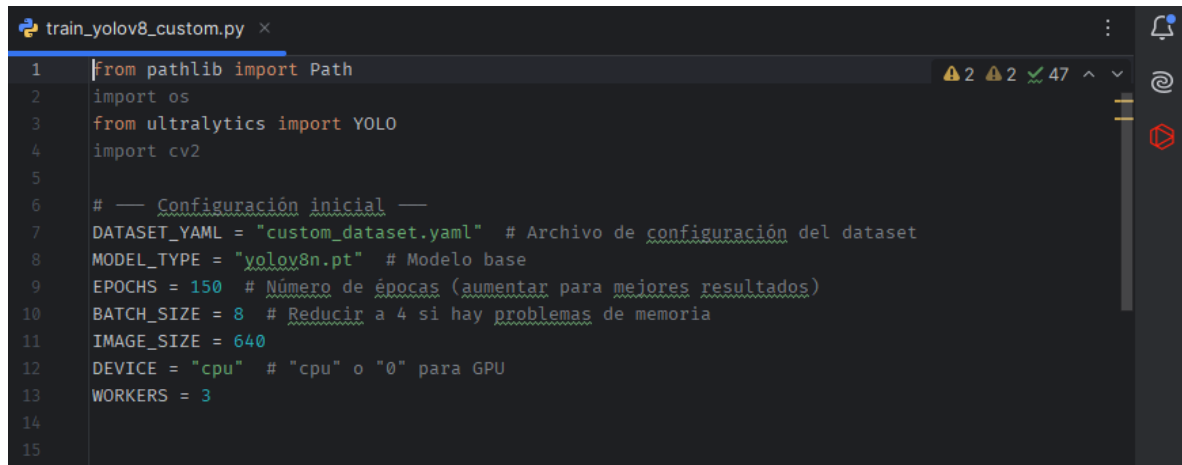
`python app.py`

Esto nos ejecutara el programa ya listo para comenzar a usarse



4. Entrenar con más épocas

En el proyecto ya se incluye un archivo para volver a entrenar con más épocas



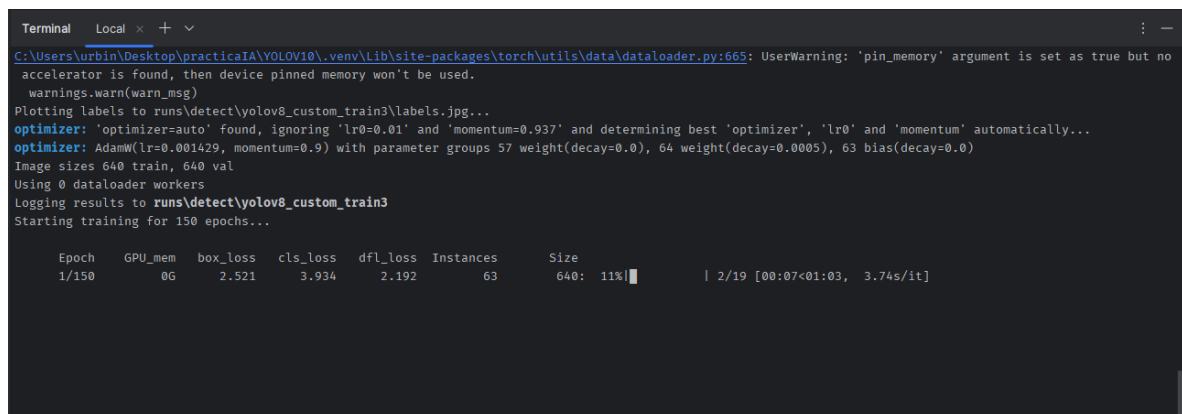
```
1 from pathlib import Path
2 import os
3 from ultralytics import YOLO
4 import cv2
5
6 # --- Configuración inicial ---
7 DATASET_YAML = "custom_dataset.yaml" # Archivo de configuración del dataset
8 MODEL_TYPE = "yolov8n.pt" # Modelo base
9 EPOCHS = 150 # Número de épocas (aumentar para mejores resultados)
10 BATCH_SIZE = 8 # Reducir a 4 si hay problemas de memoria
11 IMAGE_SIZE = 640
12 DEVICE = "cpu" # "cpu" o "0" para GPU
13 WORKERS = 3
14
15
```

Solo debemos cambiar los valores EPOCHS, todo lo demás recomiendo no tocarlo ya

Después de terminar la configuración podemos empezar el nuevo entrenamiento con el comando

`python train_yolov8_custom.py`

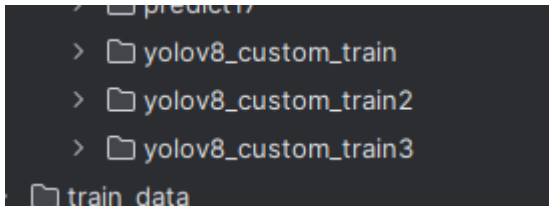
El proceso de entrenamiento va a durar dependiendo del equipo, se paciente



```
Terminal Local x + -
C:\Users\urbin\Desktop\practicaIA\YOLOV10\.venv\Lib\site-packages\torch\utils\data\dataloader.py:665: UserWarning: 'pin_memory' argument is set as true but no
accelerator is found, then device pinned memory won't be used.
warnings.warn(warn_msg)
Plotting labels to runs\detect\yolov8_custom_train3\labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.001429, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(decay=0.0)
Image sizes 640 train, 640 val
Using 0 dataloader workers
Logging results to runs\detect\yolov8_custom_train3
Starting training for 150 epochs...

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
1/150   0G       2.521    3.934    2.192    63         640: 11% | 2/19 [00:07<01:03, 3.74s/it]
```

Cuando finalice el proceso nos generara un nuevo entrenamiento, se encuentra en la carpeta runs y lo nombra como yolov8_custom_train3,4,5 dependiendo



Para usar el nuevo modelo, debemos colocar la ruta del entrenamiento en uno de los dos espacios de la lista

No incluir mas, serán ignorados

```
# RUTAS USADAS EN --add-data
MODELOS = {
    "Modelo 1: (60 épocas)": recurso_path("runs/detect/yolov8_custom_train/weights/best.pt"),
    "Modelo 2: (150 épocas)": recurso_path("runs/detect/yolov8_custom_train2/weights/best.pt")
}
```

Cuando tengamos todo listo, tenemos que volver a generar nuestro ejecutable y para ello ejecutamos el siguiente comando en la consola

pyinstaller --onefile --add-data

"runs/detect/yolov8_custom_train/weights/best.pt;runs/detect/yolov8_custom_train/weights" --add-data

"runs/detect/yolov8_custom_train2/weights/best.pt;runs/detect/yolov8_custom_train2/weights" app.py

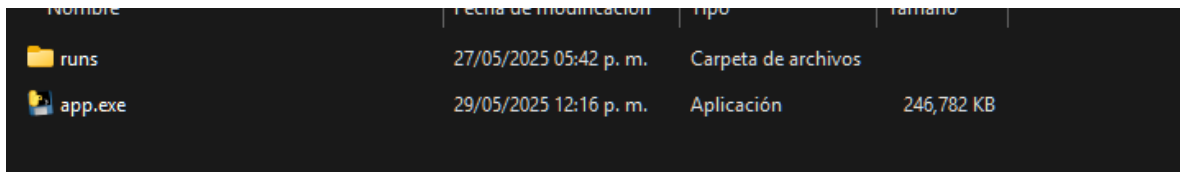
Se incluye el comando en el blog de notas “comandos” por si no se puede copiar correctamente

Esto nos regenerara el app.exe que se encuentra en la carpeta dist

```
Terminal Local x + v
375 INFO: Module search paths (PYTHONPATH):
['C:\\Users\\urbin\\Desktop\\practicaIA\\YOLOV10\\.venv\\Scripts\\pyinstaller.exe',
 'C:\\Python312\\python312.zip',
 'C:\\Python312\\DLLs',
 'C:\\Python312\\Lib',
 'C:\\Python312',
 'C:\\Users\\urbin\\Desktop\\practicaIA\\YOLOV10\\.venv',
 'C:\\Users\\urbin\\Desktop\\practicaIA\\YOLOV10\\.venv\\Lib\\site-packages',
 'C:\\Users\\urbin\\Desktop\\practicaIA\\YOLOV10\\.venv\\Lib\\site-packages\\setuptools\\_vendor',
 'C:\\Users\\urbin\\Desktop\\practicaIA\\YOLOV10']
1009 INFO: Appending 'datas' from .spec
1010 INFO: checking Analysis
1677 INFO: checking PYZ
2002 INFO: checking PKG
2246 INFO: Bootloader C:\\Users\\urbin\\Desktop\\practicaIA\\YOLOV10\\.venv\\Lib\\site-packages\\PyInstaller\\bootloader\\Windows-64bit-intel\\run.exe
2246 INFO: checking EXE
2351 INFO: Rebuilding EXE-00.toc because mtimes don't match
2352 INFO: Building EXE from EXE-00.toc
2355 INFO: Copying bootloader EXE to C:\\Users\\urbin\\Desktop\\practicaIA\\YOLOV10\\dist\\app.exe
2477 INFO: Copying icon to EXE
2541 INFO: Copying 0 resources to EXE
2541 INFO: Embedding manifest in EXE
LOV10 > app.py 26/25 CRLF UTF-8 kobline Basic 4 spaces Python 3.
```

Antes de empaquetar en instalador, tenemos que asegurar que funcione correctamente.

Desde el explorador de archivos, vamos a la carpeta y le damos doble clic



Esperamos unos segundos a que se ejecute y probamos con alguna imagen para ver si no tiene errores



Si todo esta listo, procedemos a empaquetar todo en un solo instalador.

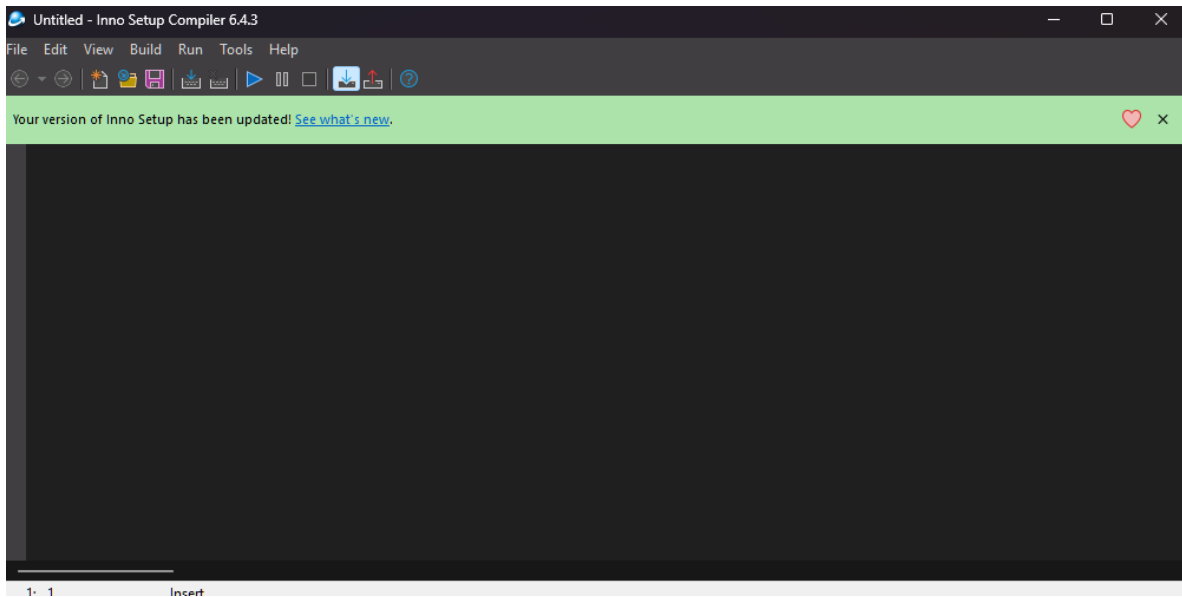
Necesitamos el inno setup compiler, podemos descargarlo desde aquí

<https://jrsoftware.org/download.php/is.exe?site=2>

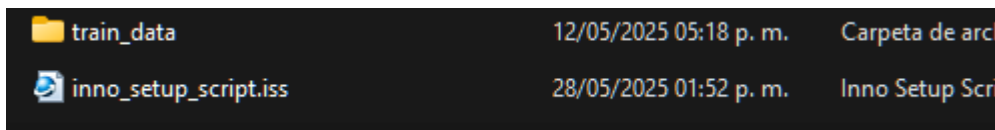
Instalarlo como cualquier otro programa, no necesita de configuraciones

5. Empaquetar en instalador

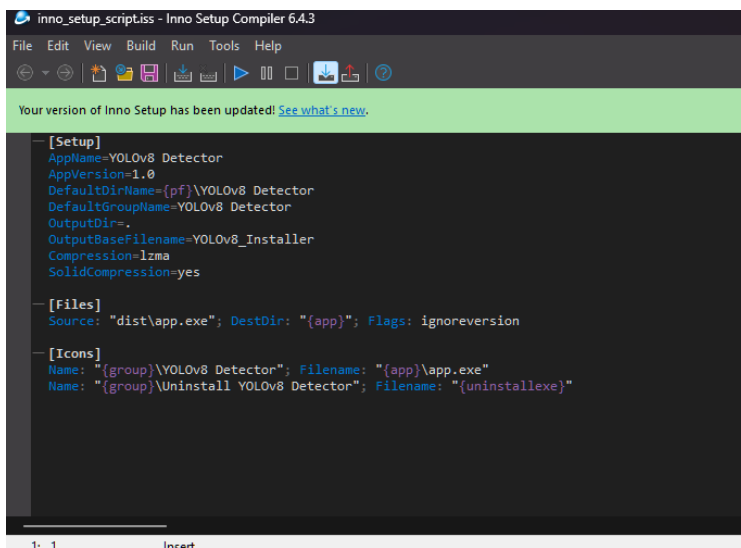
Ejecutamos el inno setup



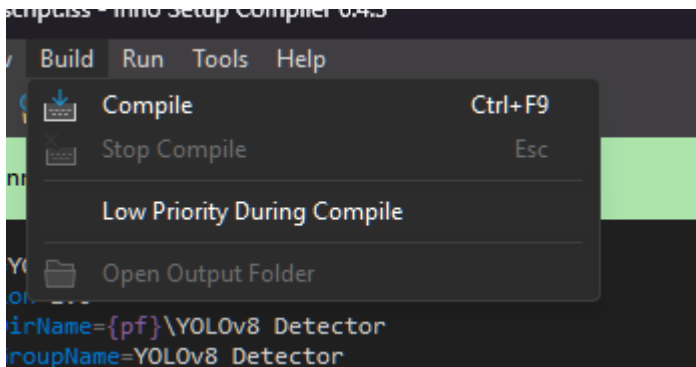
Damos clic en file -> open y vamos a la carpeta del proyecto donde seleccionamos el archivo inno_setup_script.iss



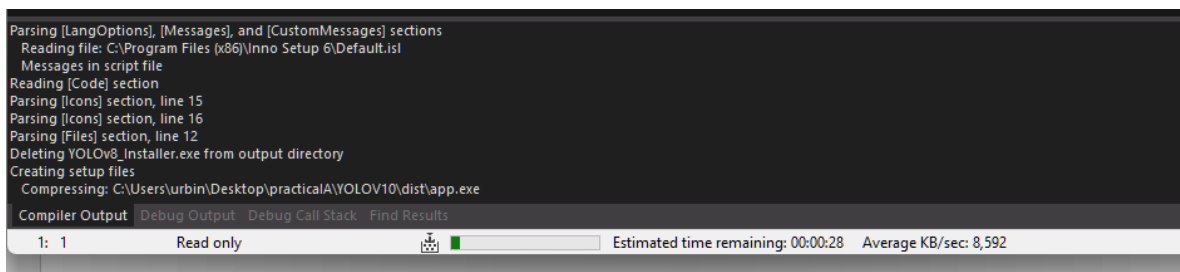
Después se cargará en el programa



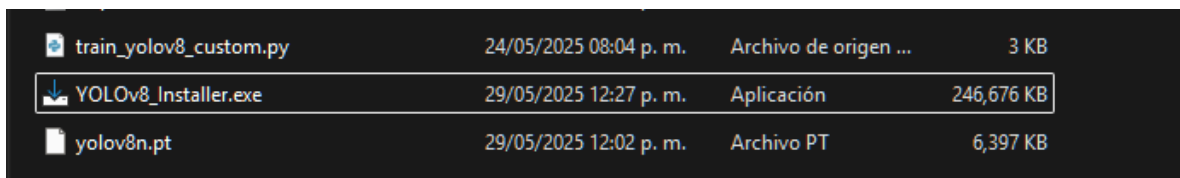
Solo queda ejecutarlo, dando clic en Build y luego en compile



Y empezara el proceso de empaquetado



Se creará el instalador en la misma carpeta del programa



Listo para instalarlo seguir los pasos del principio