**Manual de instalación PAVI Back-End**

**Para Windows**

**Requisitos:**

* git 2.30.2
* python 3.8.10 (Se puede instalar directamente de la microsoft store. Aparece como "python 3.8")
* Microsoft c++ Build Tools
* Postman (opcional)
* [Certificados](https://github.com/HectorCosgalla/PAVIdocs/blob/main/manuales/ManualUsuarioWin10.md#posibles-errores) Let'sEncrypt (x1, x2 y r3)

probado en Windows 10

**paso 1: clonar el repositorio**

1.1.- Abrir la terminal.  
1.2.- Explorar las carpetas hasta llegar a la carpeta donde se desea clonar el repositorio.  
1.3.- escribimos el comando git clone https://github.com/HectorCosgalla/PAVI\_api y esperamos a que termine.

**paso 2: instalamos un entorno virtual**

2.1.- Accedemos a la carpeta del repositorio clonado.  
2.2.- Ejecutamos el comando python3.8 -m venv venv/ y esperamos a que termine (**IMPORTANTE**: En caso de tener más de una versión de python instalada, es primordial que cada que se quiera ejecutar algun comando de python sea con python3.8, pues es la última versión compatible con las demás dependencias).

**paso 3: instalar los requerimientos de ambiente**

3.1.- Ejecutamos el entorno virtual con venv\scripts\activate.bat  
3.2.- Ejecutamos el comando pip install -r requirements-dev.txt

**IMPORTANTE:** En este paso, si no se tiene instalado Microsoft visual c++ 14.00 o superior es probable que muestre un error, para solucionarlo consultar la sección "Posibles errores")

**paso 4: Ejecutamos la app**

4.1.- Primero creamos un archivo con el nombre .env y lo guardamos en la carpeta principal (Donde se encuentra el archivo run.py) con el siguiente contenido:

DB\_USER = admin-pavi

DB\_PASSWORD = paviPassword

DB\_HOST = pavi.eywsg.azure.mongodb.net

DB\_NAME = pavi

DB\_COLLECTION = videos

FLASK\_ENV = development

FLASK\_RUN\_PORT = 5000

**Paso opcional (En caso de querer ver como se procesa el video):** Abrir el archivo "yolov4" que se encuentra en la dirección "~/pavi/pavi/services" y en la última línea borrar la parte que dice , no show=True En caso de no estar, se puede agregar.  
4.2.- Ejecutamos la app con el comando python run.py

**Uso de la API**

**Procesando el video.**

1. Abrimos el navegador en la página http://localhost:5000/documented\_api/doc
2. Hacemos clic en la pestaña que dice **videos** *Video processing endpoint*
3. Seleccionamos la pestaña **POST** e inmediatamente en el botón **Try it out**
4. En la sección de **video** seleccionamos el video que deseamos procesar
5. En la sección **algorithm** no cambiamos nada
6. Presionamos **Execute** y esperamos a que termine de procesar el video.

**Obteniendo Los resultados**

1. Abrimos el navegador en la página http://localhost:5000/documented\_api/doc\
2. Hacemos click en la pestaña que dice **videos** *Video processing endpoint*
3. Seleccionamos la pestaña **GET** e inmediatamente en el botón **Try it out**
4. En la sección de **video\_id** escribimos en nombre de un video existente en la base de datos (Por ejemplo: Run).
5. Presionamos **Execute** y esperamos a que tener la lista de resultados.

**Posibles errores:**

1. Error: "Microsoft Visual C++ 14.0 is required. Get it with "Microsoft Visual C++ Build Tools":"  
   **Solución**: Paso 1: Entrar a la pagina: <https://visualstudio.microsoft.com/es/visual-cpp-build-tools/> y descargar Build Tools.  
   Paso 2: Una vez descargado, ejecutamos el instalador y una vez instalado y abierto seleccionamos la opcion Desarrollo para el escritorio con c++ e instalamos  
   Paso 3: Una vez finalizada la instalación reiniciamos el equipo y continuamos con la instalación.
2. Error: pymongo [SSL: CERTIFICATE\_VERIFY\_FAILED]: certificate has expired on Mongo Atlas  
   **Solucion**:  
   Paso 1: Descargar el zip del siguiente [link](https://github.com/HectorCosgalla/PAVIdocs/blob/main/archivos/Let'sEncryptCertificates.zip)  
   Paso 2: Descomprimirlo e instalar los 3 certificados siguiendo los pasos sin cambiar nada.  
   Paso 3: Una vez finalizada la instalación de los 3 certificados, reiniciar la computadora.

**Para Linux/Mac**

**Requisitos:**

* git 2.30.2
* python 3.8.11 (**IMPORTANTE**: Incluir en la instalación el comando "pip")
* Postman (Opcional)

probado en fedora 33 y 34

**paso 1: clonar el repositorio**

1.1.- Abrir la terminal.  
1.2.- Explorar las carpetas hasta llegar a la carpeta donde se desea clonar el repositorio.  
1.3.- escribimos el comando git clone https://github.com/HectorCosgalla/PAVI\_api y esperamos a que termine.

**paso 2: instalamos un entorno virtual**

2.1.- Accedemos a la carpeta del repositorio clonado.  
2.2.- Ejecutamos el comando python3.8 -m venv venv/ y esperamos a que termine (**IMPORTANTE**: En caso de tener más de una versión de python instalada, es primordial que cada que se quiera ejecutar algun comando de python sea con python3.8, pues es la última versión compatible con las demás dependencias).

**paso 3: instalar los requerimientos de ambiente**

3.1.- Ejecutamos el entorno virtual con source venv/bin/activate  
3.2.- Ejecutamos el comando pip install -r requirements-dev.txt

**paso 4: Ejecutamos la app**

4.1.- Primero creamos un archivo con el nombre .env y lo guardamos en la carpeta principal (Donde se encuentra el archivo run.py) con el siguiente contenido:

DB\_USER = admin-pavi

DB\_PASSWORD = paviPassword

DB\_HOST = pavi.eywsg.azure.mongodb.net

DB\_NAME = pavi

DB\_COLLECTION = videos

FLASK\_ENV = development

FLASK\_RUN\_PORT = 5000

**Paso opcional (En caso de querer ver como se procesa el video):** Abrir el archivo "yolov4" que se encuentra en la dirección "~/pavi/pavi/services" y en la última línea borrar la parte que dice , no\_show=True  
4.2.- Ejecutamos la app con el comando python3.8 run.py

**Uso de la API**

**Procesando el video.**

1. Abrimos el navegador en la página http://localhost:5000/documented\_api/doc
2. Hacemos clic en la pestaña que dice **videos** *Video processing endpoint*
3. Seleccionamos la pestaña **POST** e inmediatamente en el botón **Try it out**
4. En la sección de **video** seleccionamos el video que deseamos procesar
5. En la sección **algorithm** no cambiamos nada
6. Presionamos **Execute** y esperamos a que termine de procesar el video.

**Obteniendo Los resultados**

1. Abrimos el navegador en la página http://localhost:5000/documented\_api/doc\
2. Hacemos click en la pestaña que dice **videos** *Video processing endpoint*
3. Seleccionamos la pestaña **GET** e inmediatamente en el botón **Try it out**
4. En la sección de **video\_id** escribimos en nombre de un video existente en la base de datos (Por ejemplo: Run).
5. Presionamos **Execute** y esperamos a que tener la lista de resultados.

**Posibles errores:**

1.- Error "gcc: error fatal: cannot execute ‘cc1plus’: execvp: No existe el fichero o el directorio"  
**Solucion**: Instalar el paquete gcc-g++ con el comando sudo dnf install gcc-g++