

**Documentación**

**IMPERIUS**

Contenido

[Proyecto IMPERIUS 2](#_Toc90138891)

[Conocimiento previos 2](#_Toc90138892)

[Estructural del proyecto 6](#_Toc90138893)

[Referencias 8](#_Toc90138894)

# Proyecto IMPERIUS

IMPERIUS es un sitio web , que nos permite utilizar algorimos para procesar imágenes , con el fin de utilizar para temas de estudio y practica en el tema de Procesamiento de imagenes.

Esta creado con el lenguaje de programacion python y la librería OpenCV , se implementa un api en el BackEnd y se cosume mediantes Axios desde FronEnd.

# Conocimiento previos

1. Python 3.7.0
2. Flask
3. Vue js
4. Java Script
5. BoostrapVue
6. git
7. Editor de texto Vs Code Plugin THUNDER CLIENT

Creacion de Ambiente Virtuales y ejecucion del programa

Para SO linux se recomienta utilizar conda

<https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html>

Pasos

1. Instalar Python 3.7.0

<https://www.python.org/>

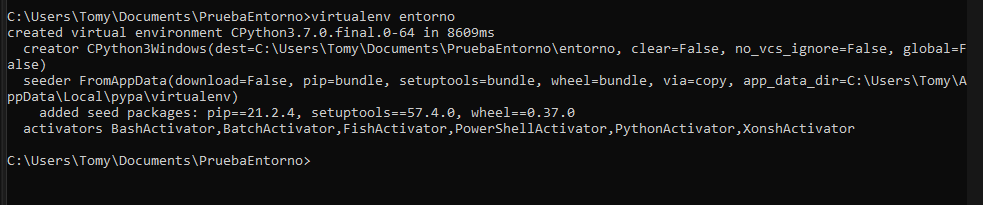
1. Instalar Virtualenv

<https://pypi.org/project/virtualenv/>

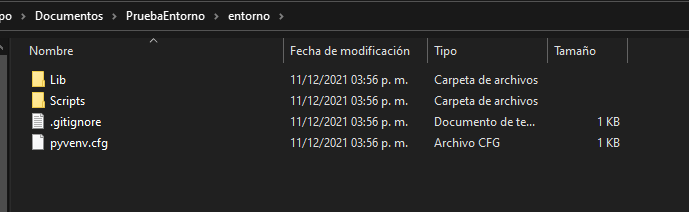
1. Abrir la terminal y ejecutar dentro de la carpeta en donde estará el proyecto

Virtualenv <nombre\_entorno>

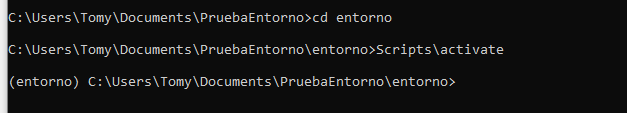
Windows



Se debe de crear una carpeta con la siguiente estructura



1. Movernos a la carpeta y ejecutar el comando Scripts\activate se activa el entorno



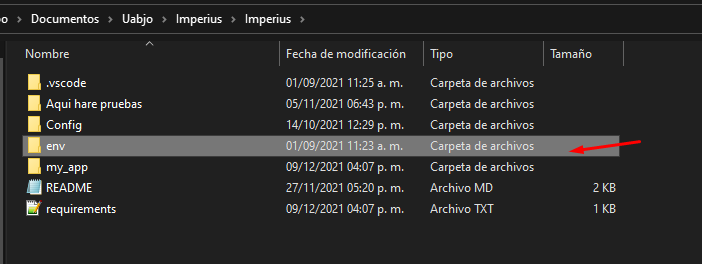
1. Dentro del entorno activado se debe de clonar el repositorio en la rama desarrolo

git clone -b <branchname> <remote-repo-url>

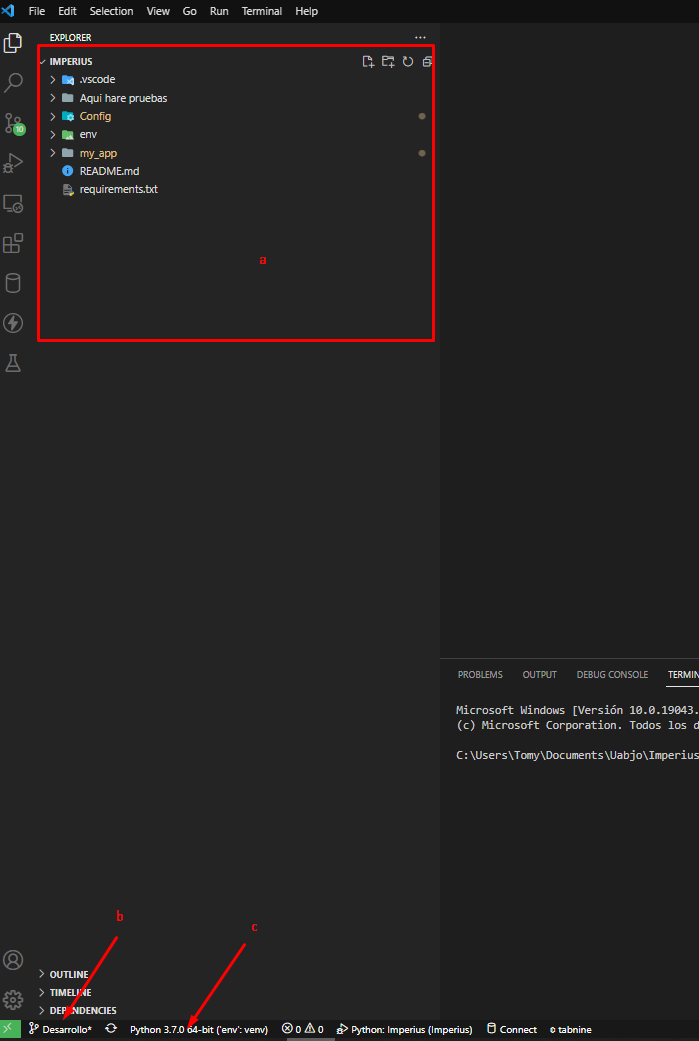
git clone –b Desarrollo <https://github.com/TomasGit-1/Imperius.git>

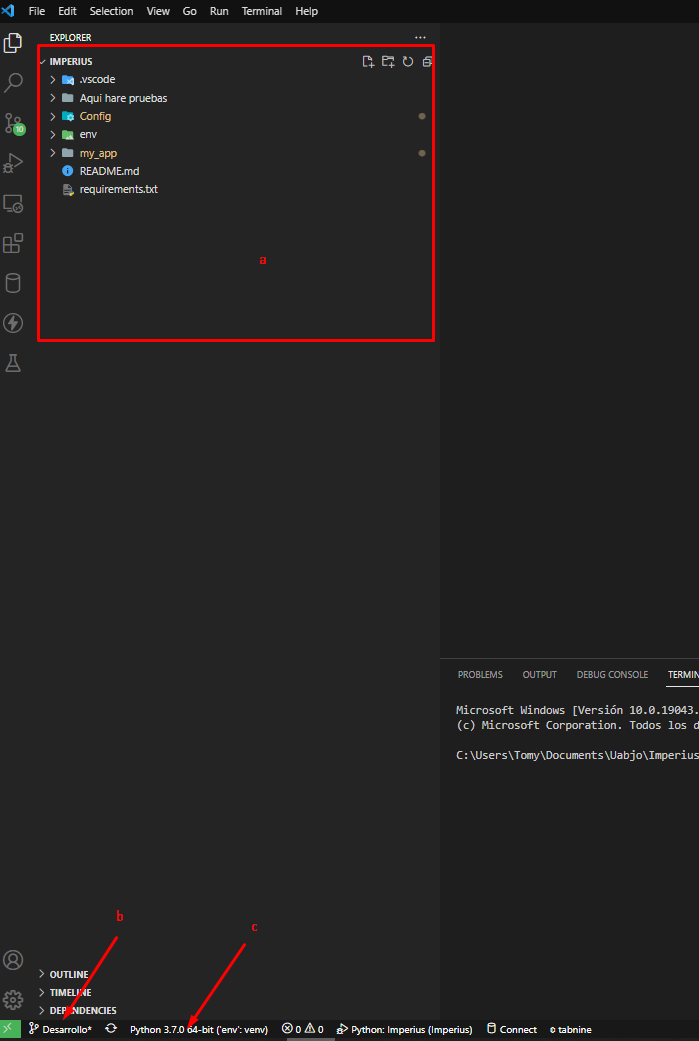
No utilicen la rama main

1. Una vez clonado el proyecto repetimos el paso 3 dentro de la carpeta del Proyecto

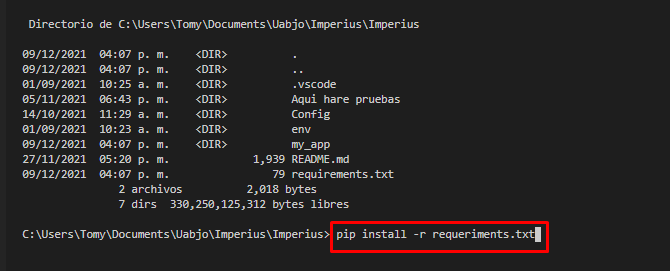


1. Abrir el proyecto desde vsCode
2. Estructura del proyecto
3. Rama en la que se trabaja
4. Entorno virtual

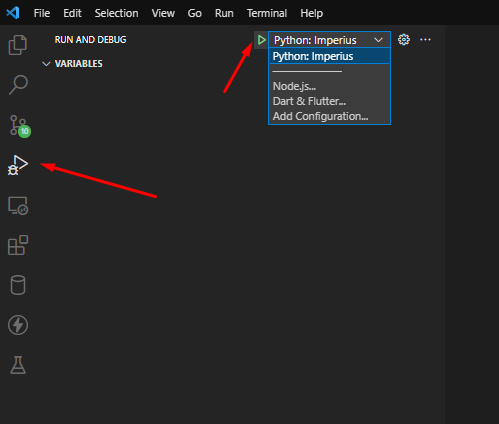




1. Abrimos la terminal de vsCode y ejecutamos pip install -r requeriments.txt para instalar las dependencias



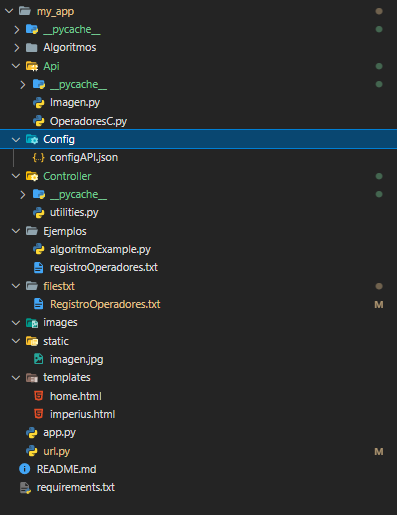
1. Nos vamos a debug y ejecutamos Imperius



1. Corriendo el Proyecto, flask se ejecuta en localhost



# Estructural del proyecto



Algoritmos

Carpeta donde irán almacenados los algoritmos del usuario

Api

Esta la clase de la Imagen aquí se cargan todos la lógica

Config

Es un archivo .json donde se guardaran las rutas de los archivos modelos o configuraciones locales, tener en cuenta que las rutas cambies dependiendo del sistema operativo.

Controller

Tiene un archivo .py donde se agregan funciones que se pueden reutilizar como lectura de un archivo

Ejemplos

Tiene ejemplos de los operadores o el algoritmo que debe de cargar el usuario

Filestxt

Esta el archivo en donde están almacenados los operadores que se utilizan el la aplicación

Images

Se almacenan las imágenes que se procesan

Static

Recursos de la aplicación como logos o archivos js

Templates

Se guardan todos los Html de la aplicación

App.py y url.py aquí se genera las peticiones HTTP que ingresan a la aplicación

Readme

Link de documentación etc..

Requeriments

Aquí iran agregadas las librerías que se ocupan en la aplicación , importante poner la version

# Referencias

<https://www.python.org/>

<https://vuejs.org/>

<https://bootstrap-vue.org/>

<https://pypi.org/project/pip/>

<https://git-scm.com/>

<https://github.com/TomasGit-1/Imperius>

<https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html>