**INSTALACION PYTHON**

**Comandos Utilizados**

python

python3

exit() para salir de la interfaz de python

**Instalación**

apt update

sudo apt update

sudo apt -y upgrade

**Verificar Instalación de python**

python3 -V

**Instalación de gestor de paquetes de dependencias**

sudo apt insstall -y python3-pip

**Verificar Instalación del gestor**

pip3 -V

**Dependencias en entorno profesional**

apt install -y build-essential libssl-dev libffi-dev python3-dev

**CREACION DE AMBIENTES:**

**Verificar donde esta python y pip**

which python3

which pip3

**Instalar ambiente virtual, si estas en linus o wsl debes instalar**

sudo apt install -y python3-venv

**Poner cada proyecto en su propio ambiente, entrar en cada carpeta**.

python3 -m venv env

**Activar el ambiente**

source env/bin/activate

source env/Scripts/activate (para windows)

**Salir del ambiente virtual**

deactivate

**Podemos instalar las librerías necesarias en el ambiente virtual como, por ejemplo**

pip3 install matplotlib==3.5.0

**Verificar las instalaciones de las dependencias pip**

pip3 freeze

Requirements.txt = **Archivo que gestiona todas las dependencias y en que versiones se necesitan, para correr el proyecto.**

**Generar el archivo con todas las dependencias dentro con el siguiente comando**

pip3 freeze > requirements.txt

**Revisar lo que hay dentro del archivo**

cat requirements.txt

**Instalar las dependencias necesarias para contribuir más rápido en proyectos**

pip3 install -r requirements.txt

Preparar archivo para contribución

**PANDAS**

Es una de las librerías más utilizadas en Python y nos sirven para analizar y manipular datos de archivos duros.

**Activar ambiente del proyecto**

source env/bin/activate

source env/Scripts/activate

**Verificar**

which python3

**Ver que hay dentro del archivo en el cual se evidencia que no hay pandas**

cat requirements.txt

**Agregar nueva librería**

pip3 install pandas

**Verificar librerías instaladas**

pip3 freeze

**Actualizar el documento que contiene las librerías**

pip3 freeze > requirements.txt

**FASTAPI**

**Installation**

pip install fastapi

**Uvicorn Installation**

pip install "uvicorn[standard]"

**crear un archivo con main.py:**

from typing import Union

from fastapi import FastAPI

app = FastAPI()

@app.get("/")

def read\_root():

return {"Hello": "World"}

@app.get("/items/{item\_id}")

def read\_item(item\_id: int, q: Union[str, None] = None):

return {"item\_id": item\_id, "q": q}

**Con el siguiente comando puedes devolver una página web**

@app.get("/contact", response\_class=HTMLResponse)

def get\_list():

return """

<h1>¡Hola! Soy una pagina</h1>

<p>Soy un parrafo</p>

"""

**correr el programa en la Html://**

uvicorn main:app –reload

**DOCKER**

Es una herramienta que nos sirve para aislar entornos de ejecución, eso lo hace con contenedores que tiene Docker por detrás.

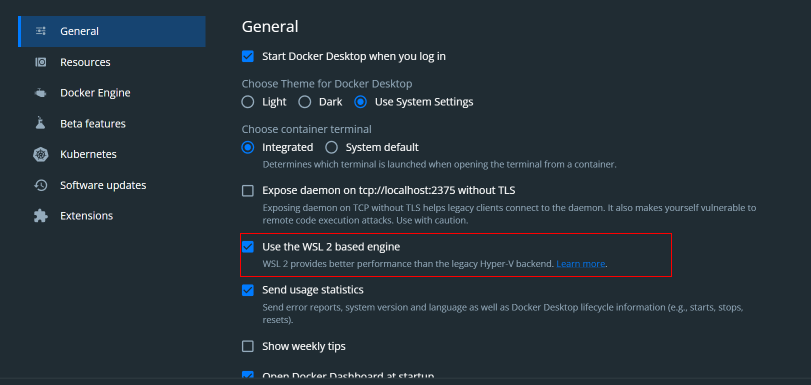
**Instalación de Docker**

Según el sistema operativo que utilices puede variar la instalación, así que a continuación te daré las indicaciones base para la instalación según tu sistema operativo:

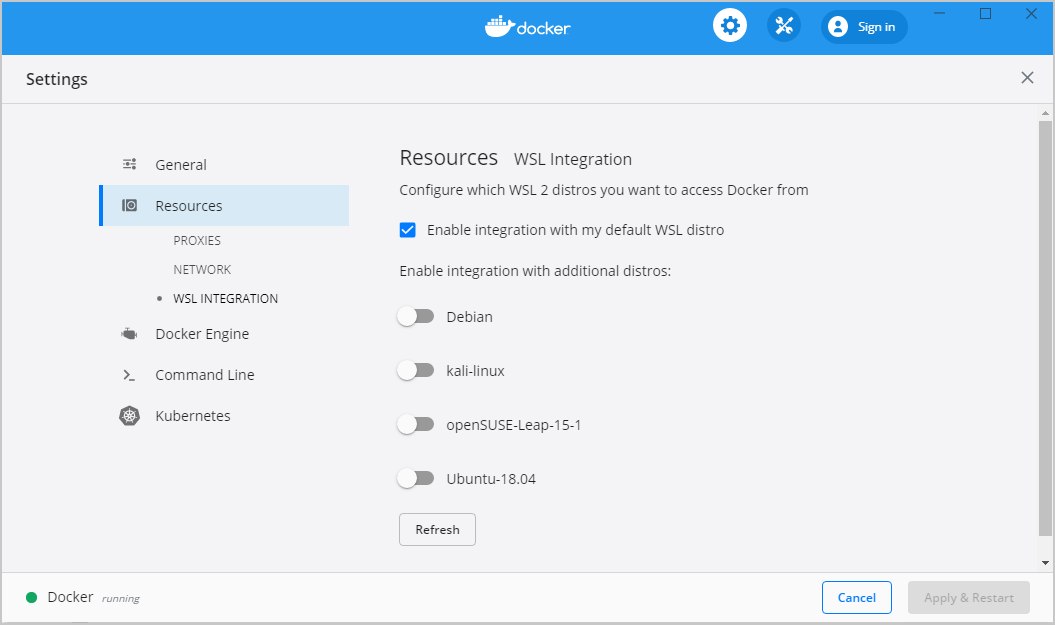
**Instalación en Windows con WSL (Recomendada) 🐧**

Debes descargar el instalador desde la página de [Docker Windows.](https://docs.docker.com/desktop/install/windows-install/)

Cuando ya tienes instalado Docker Desktop dentro de tus programas debes abrirlo y debes asegurarte que la opción “Use the WSL 2 based engine” está habilitada:



Luego en la sección “Resources > WSL Integration”, asegurarate que la opcion “Enable integration with my default WSL distro”, está habilitada:



**Resources**

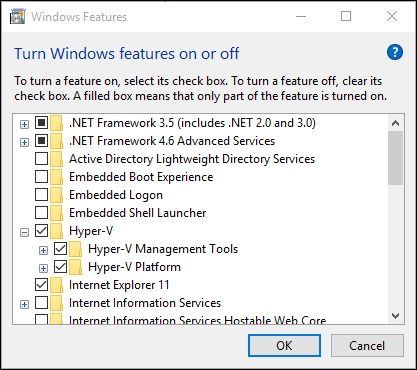
Puedes ver más detalles de Docker con WLS 👉 [Docker Desktop WSL 2 backend](https://docs.docker.com/desktop/windows/wsl/)

**Instalación en Windows 🪟**

Debes descargar el instalador desde la página de [Docker for Windows.](https://docs.docker.com/desktop/install/windows-install/)

Cuando ya tienes instalado Docker Desktop dentro de tus programas, una de las cosas que debes tener en cuenta en la instalación con Windows es que debes contar con Windows 10 de 64 Bits o superior y debes habilitar el Hyper-V de Windows.

Si quieres conocer los detalles, aquí te dejo el detalle como habilitar [habilitar **Hyper-V** desde la Interfaz de Windows](https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v)

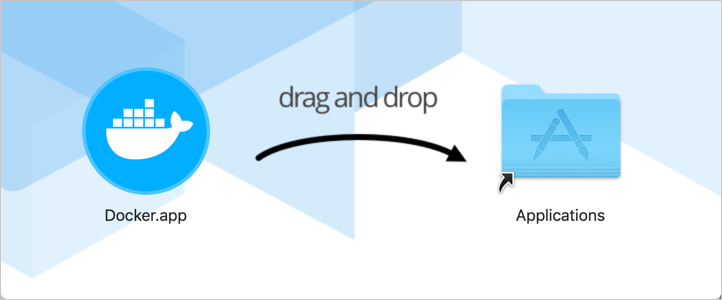


Instalación en macOS 🍎

En Mac tienes dos opciones. Todo dependerá si tienes los nuevos chips M1 o Intel, ya que hay un instalable apropiado para ambas arquitecturas de chip. Puedes escoger el instalable desde [Install Docker Desktop on Mac](https://docs.docker.com/desktop/install/mac-install/)

Adicionalmente, si cuentas con los nuevos chips M1, debes ejecutar la siguiente instrucción en tu terminal softwareupdate --install-rosetta

Una vez descargues el instalador adecuado, solo debes seguir los pasos y pasar Docker Desktop a tus aplicaciones.



**Instalación en Ubuntu 🐧**

Estos son los pasos para instalarlo dentro de Ubuntu, sin embargo, también puedes ver directamente  [Install Docker Engine on Ubuntu](https://docs.docker.com/engine/install/ubuntu/).

DIGITAR LOS SIGUIENTES COMANDOS:

------------------------------------------------------------------------------------------

sudo apt-get update

------------------------------------------------------------------------------------------

sudo apt-get install \

ca-certificates \

curl \

gnupg \

lsb-release

------------------------------------------------------------------------------------------

sudo mkdir -p /etc/apt/keyrings

------------------------------------------------------------------------------------------

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

------------------------------------------------------------------------------------------

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

------------------------------------------------------------------------------------------

sudo apt-get update

------------------------------------------------------------------------------------------

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

------------------------------------------------------------------------------------------

sudo docker run hello-world

------------------------------------------------------------------------------------------

Para otras distribuciones de Linux:

* [Install Docker Engine on CentOS](https://docs.docker.com/engine/install/centos/)
* [Install Docker Engine on Debian](https://docs.docker.com/engine/install/debian/)
* [Install Docker Engine on Fedora](https://docs.docker.com/engine/install/fedora/)

**CONSTRUIR NUESTRO CONTENEDOR**

**SE CONSTRUYE CON EL COMANDO:**

docker-compose build

**PARA LANZARLO:**

docker-compose up -d

**PARA VER EL ESTADO DE ESE CONTENEDOR:**

docker-compose ps

**Trabajar en ese ambiente:**

docker-compose exec app-csv bash

**Bajar el contendor donde está corriendo la aplicación:**

docker-compose down

**salir del contenedor:**

exit

**Ver los archivos y carpetas en mi Docker, estando corriendo:**

ls -l (muestra en forma de lista)

cat “nombre del archivo” (puedes ver el código del archivo)