#-- primero instalar python3

#-- comandos pipenv

<https://luidasa.com/comandos-de-pipenv/>

#--crear proyecto

mkdir project1\_pipenv

cd project1\_pipenv/

#-- instalar entorno de desarrollo pipenv

pip3 install pipenv

#-- crear entorno de trabajo virtual de python o iniciar uno ya creado

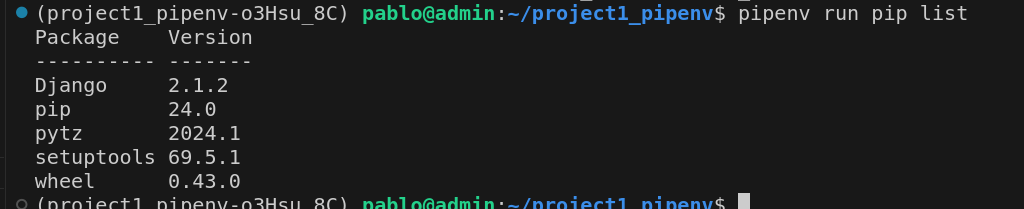
pipenv shell

#---Para activar el entorno virtual o crear uno nuevo vacio

$ pipenv run python

#-- ver paquetes instalados

pipenv run pip list



#-- para remover un entorno virtual creado con pipenv

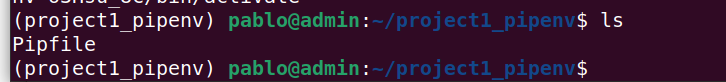
$ pipenv --rm

#-- instalar desde Pipfile

pipenv install

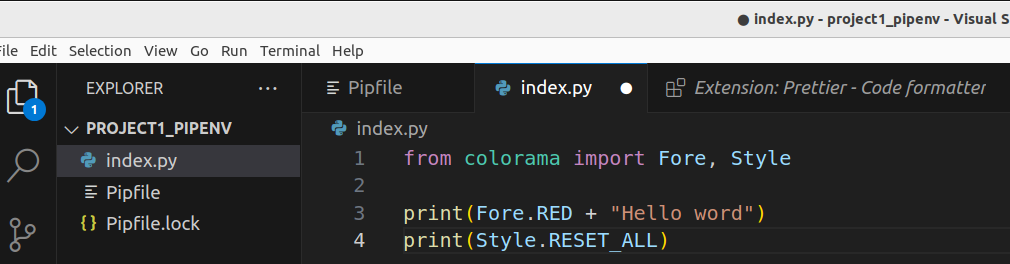
#--paquetes desarrollo

pipenv install --dev



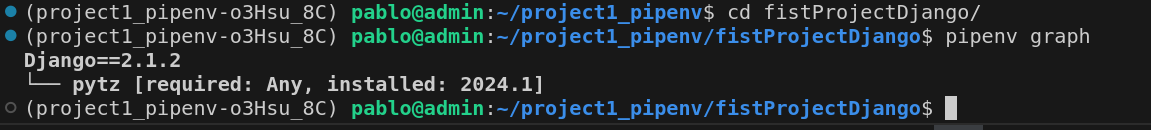
#-- instalar paquete en entorno proyecto cambio de colores terminal

pipenv install colorama



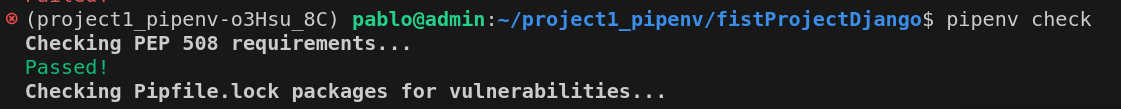
#-- dependencias instaladas

pipenv graph



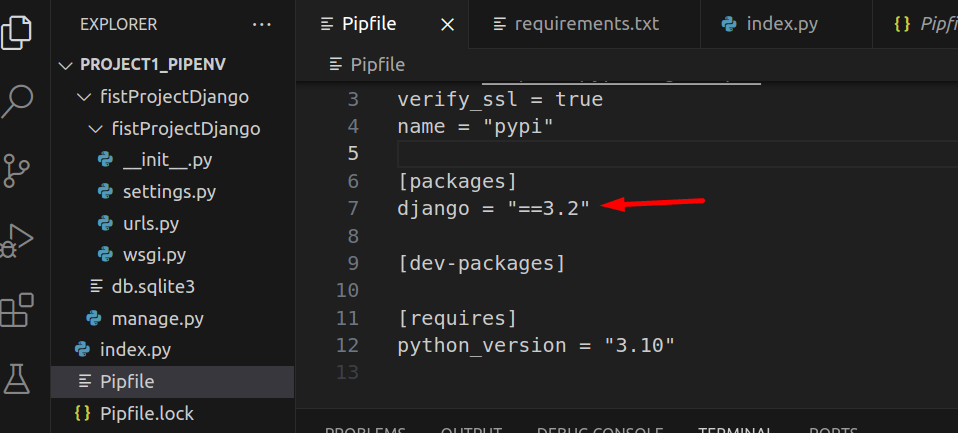
#--chequear vulnerabilidades y versiones recomendadas

pipenv check



#-- instalar paquetes actualizados

Primero modifico la nueva version a instalar en el archivo



#-- ejercuto el comando

pipenv install

#-- Como omitir el archivo PipFile y tomar las versiones de Pipfile.lock que son las versiones especificas que se esta trabajando.

pipenv install --ignore-pipfile

#---------------------------------------------------------------

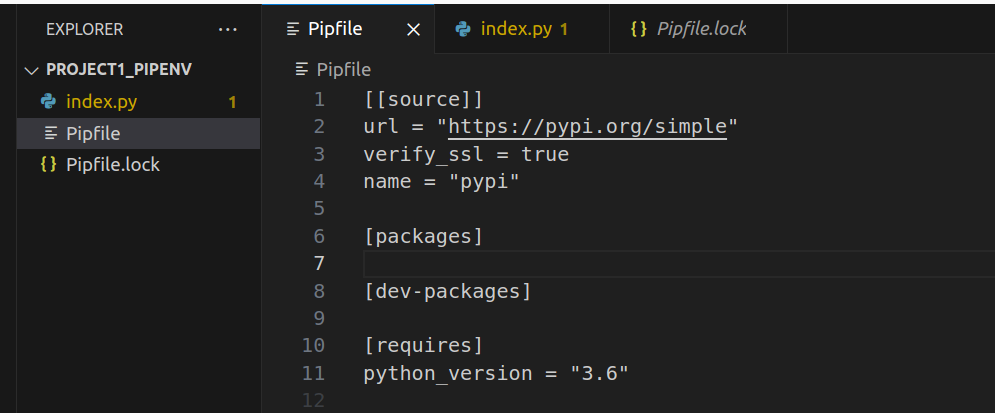
#-- corre python

python index.py

#-- quitar paquete

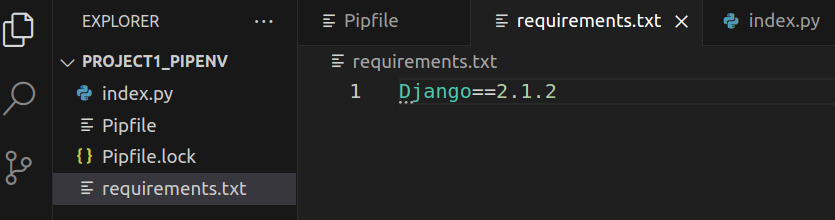
pipenv uninstall colorama

#---- instalar paquetes para solo desarrollo no en genereal – dev



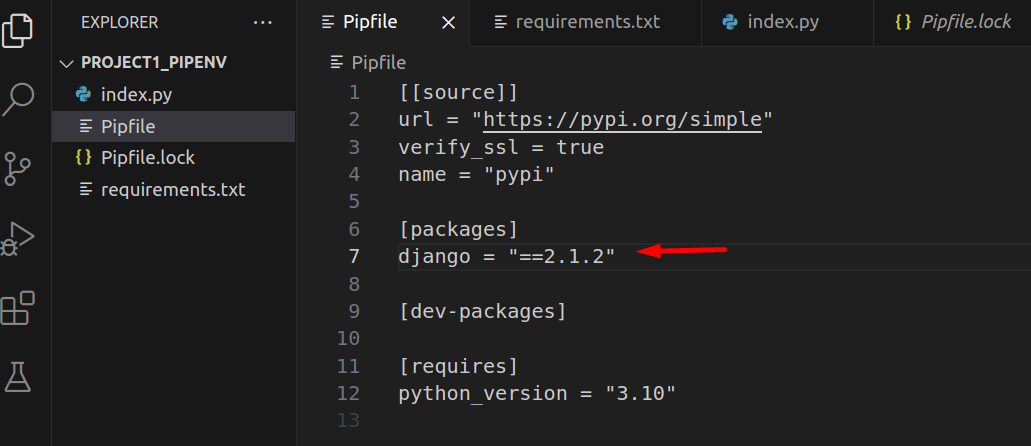
pipenv install colorama-dev

#------- como instalar de una lista de archivos de paqueetes



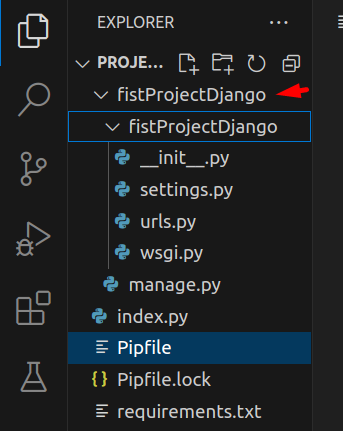
#-- instalar desde archivo

pipenv install -r requirements.txt



#-- crear nuevo proyecto en django

django-admin startproject fistProjectDjango



#-- correr proyecto django

cd fistProjectDjango/

python manage.py runserver

