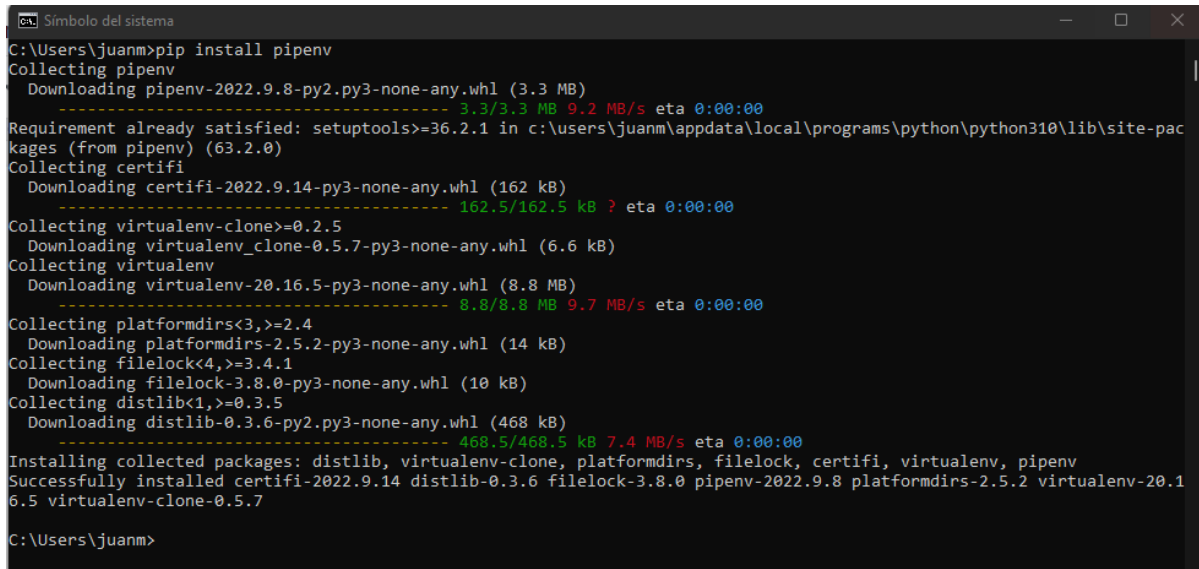


Este primer paso es opcional. Sin embargo, es recomendable crear un ambiente virtual para cada uno de los proyectos en los que se está trabajando.

1. Instale pipenv (también puede crearse con virtualenv, anaconda)

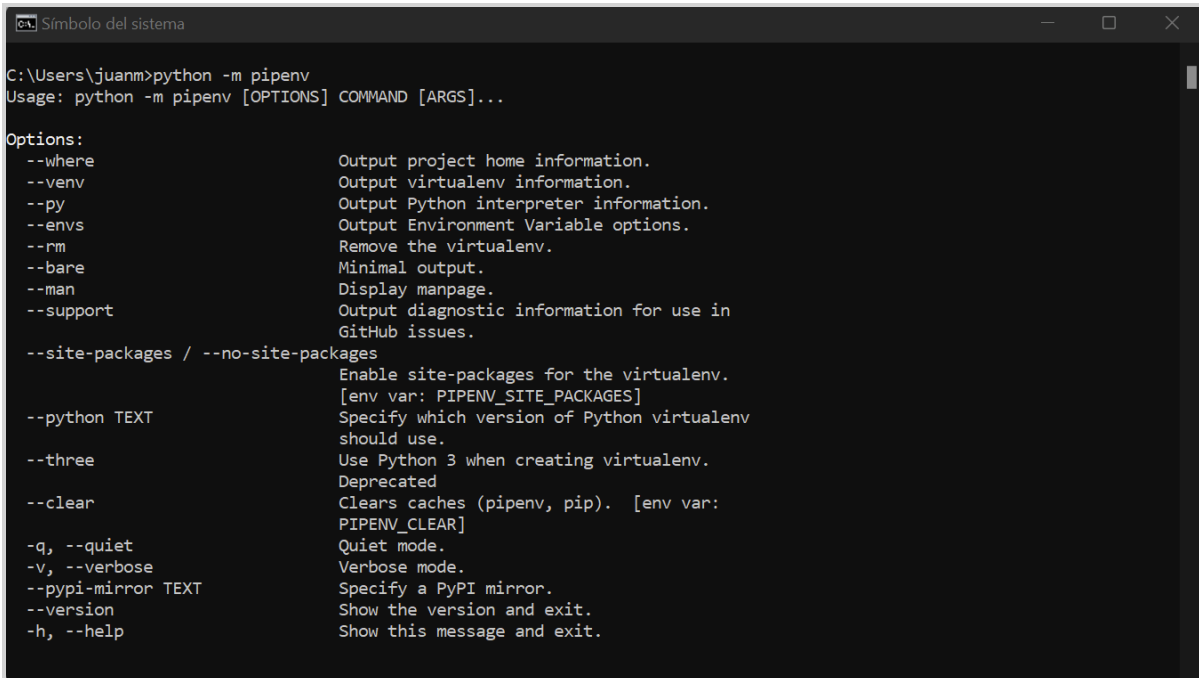
pip install pipenv



```
Símbolo del sistema
C:\Users\juanm>pip install pipenv
Collecting pipenv
  Downloading pipenv-2022.9.8-py2.py3-none-any.whl (3.3 MB)
----- 3.3/3.3 MB 9.2 MB/s eta 0:00:00
Requirement already satisfied: setuptools>=36.2.1 in c:\users\juanm\appdata\local\programs\python\python310\lib\site-packages (from pipenv) (63.2.0)
Collecting certifi
  Downloading certifi-2022.9.14-py3-none-any.whl (162 kB)
----- 162.5/162.5 kB ? eta 0:00:00
Collecting virtualenv-clone>=0.2.5
  Downloading virtualenv_clone-0.5.7-py3-none-any.whl (6.6 kB)
Collecting virtualenv
  Downloading virtualenv-20.16.5-py3-none-any.whl (8.8 MB)
----- 8.8/8.8 MB 9.7 MB/s eta 0:00:00
Collecting platformdirs<3,>=2.4
  Downloading platformdirs-2.5.2-py3-none-any.whl (14 kB)
Collecting filelock<4,>=3.4.1
  Downloading filelock-3.8.0-py3-none-any.whl (10 kB)
Collecting distlib<1,>=0.3.5
  Downloading distlib-0.3.6-py2.py3-none-any.whl (468 kB)
----- 468.5/468.5 kB 7.4 MB/s eta 0:00:00
Installing collected packages: distlib, virtualenv-clone, platformdirs, filelock, certifi, virtualenv, pipenv
Successfully installed certifi-2022.9.14 distlib-0.3.6 filelock-3.8.0 pipenv-2022.9.8 platformdirs-2.5.2 virtualenv-20.16.5 virtualenv-clone-0.5.7
C:\Users\juanm>
```

2. Verifique que pipenv se instaló correctamente

python -m pipenv



```
Símbolo del sistema
C:\Users\juanm>python -m pipenv
Usage: python -m pipenv [OPTIONS] COMMAND [ARGS]...

Options:
  --where                Output project home information.
  --venv                 Output virtualenv information.
  --py                  Output Python interpreter information.
  --envs                 Output Environment Variable options.
  --rm                  Remove the virtualenv.
  --bare                 Minimal output.
  --man                  Display manpage.
  --support              Output diagnostic information for use in
                        GitHub issues.
  --site-packages / --no-site-packages
                        Enable site-packages for the virtualenv.
                        [env var: PIPENV_SITE_PACKAGES]
  --python TEXT          Specify which version of Python virtualenv
                        should use.
  --three                Use Python 3 when creating virtualenv.
                        Deprecated
  --clear                Clears caches (pipenv, pip). [env var:
                        PIPENV_CLEAR]
  -q, --quiet            Quiet mode.
  -v, --verbose          Verbose mode.
  --pypi-mirror TEXT     Specify a PyPI mirror.
  --version              Show the version and exit.
  -h, --help             Show this message and exit.
```

3. Cree una carpeta para el proyecto y desde la consola ubíquese en esta carpeta. En este ejemplo se creó la carpeta PI_MLProject en el escritorio

```
cd C:\Users\juanm\OneDrive\Escritorio\PI_MLProject
```

4. Cree el ambiente virtual (quedará con el nombre del current path)

```
pipenv install
```

```
Símbolo del sistema - pipenv shell

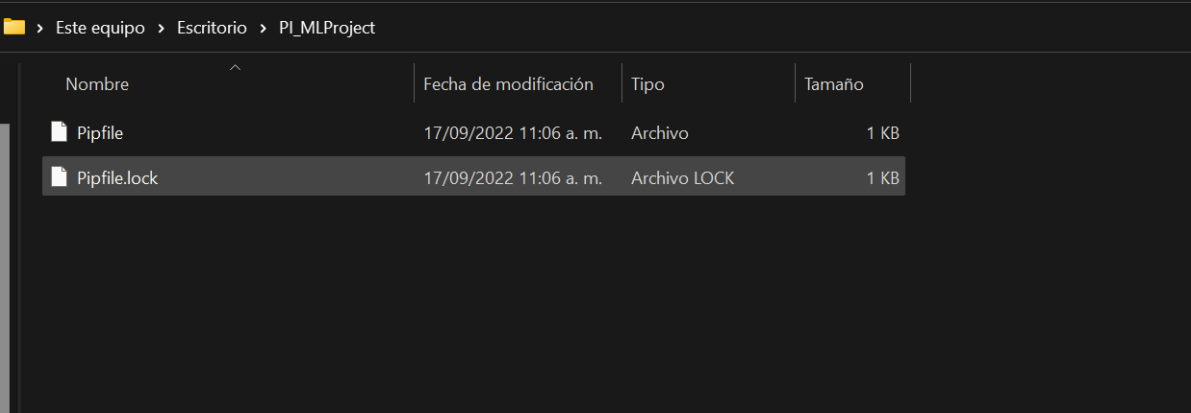
C:\Users\juanm>cd C:\Users\juanm\OneDrive\Escritorio\PI_MLProject

C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>pipenv install
Creating a virtualenv for this project...
Pipfile: C:\Users\juanm\OneDrive\Escritorio\PI_MLProject\Pipfile
Using C:/Users/juanm/AppData/Local/Programs/Python/Python310/python.exe (3.10.7) to create virtualenv...
[== ] Creating virtual environment...created virtual environment CPython3.10.7.final.0-64 in 4746ms
creator CPython3Windows(dest=C:\Users\juanm\.virtualenvs\PI_MLProject-xSoj08wY, clear=False, no_vcs_ignore=False, glob
al=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\juanm\
AppData\Local\pypa\virtualenv)
    added seed packages: pip==22.2.2, setuptools==65.3.0, wheel==0.37.1
    activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

Successfully created virtual environment!
Virtualenv location: C:\Users\juanm\.virtualenvs\PI_MLProject-xSoj08wY
Creating a Pipfile for this project...
Pipfile.lock not found, creating...
Locking [packages] dependencies...
Locking [dev-packages] dependencies...
Updated Pipfile.lock (e4eef2)!
Installing dependencies from Pipfile.lock (e4eef2)...
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.

C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>pipenv shell
Launching subshell in virtual environment...
Microsoft Windows [Versión 10.0.22000.856]
(c) Microsoft Corporation. Todos los derechos reservados.
```

En la carpeta del proyecto deberán verse los siguientes documentos:



Nombre	Fecha de modificación	Tipo	Tamaño
Pipfile	17/09/2022 11:06 a. m.	Archivo	1 KB
Pipfile.lock	17/09/2022 11:06 a. m.	Archivo LOCK	1 KB

5. Active el ambiente virtual

```
pipenv shell
```

En este punto la consola debe verse de la siguiente forma:

```
Símbolo del sistema - pipenv shell
C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>pipenv shell
Launching subshell in virtual environment...
Microsoft Windows [Versión 10.0.22000.856]
(c) Microsoft Corporation. Todos los derechos reservados.

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>
```

Desde este punto podremos instalar paquetes en el ambiente virtual usando pipenv en vez de pip.

Si no creó el ambiente virtual, puede seguir los pasos desde acá cambiando pipenv por pip.

6. Instale tensorflow 2.8.2 en el ambiente virtual dado que es la versión que se utiliza en Google colab (cualquier versión de tensorflow debería funcionar sin problema)

```
pipenv install tensorflow==2.8.2
```

7. Compruebe que django esté instalado en el ambiente virtual

```
python -m django
```

8. En caso de que no esté instalado (como en mi caso), instale django

```
Símbolo del sistema - pipenv shell

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>python -m django
C:\Users\juanm\virtualenvs\PI_MLProject-xSoj08WY\Scripts\python.exe: No module named django

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>
```

```
pipenv install django
```

```
Símbolo del sistema - pipenv shell

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>pipenv install django
Installing django...
Adding django to Pipfile's [packages]...
Installation Succeeded
Pipfile.lock (e4eef2) out of date, updating to (32d259)...
Locking [packages] dependencies...
Locking...
Resolving dependencies...
Success!
Locking [dev-packages] dependencies...
Updated Pipfile.lock (32d259)!
Installing dependencies from Pipfile.lock (32d259)...

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>
```

9. Adicionalmente, debe instalar la librería Pillow para manejo de imágenes

```
pipenv install Pillow
```

```
Simbolo del sistema - pipenv shell

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>pipenv install Pillow
Installing Pillow...
Adding Pillow to Pipfile's [packages]...
Installation Succeeded
Pipfile.lock (32d259) out of date, updating to (1cec49)...
Locking [packages] dependencies...
Locking...
Resolving dependencies...
Success!
Locking [dev-packages] dependencies...
Updated Pipfile.lock (1cec49)!
Installing dependencies from Pipfile.lock (1cec49)...

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>
```

Si abre el archivo Pipfile notará que aparece la lista de los paquetes instalados:

```
Pipfile  x  test.py  x
1  [[source]]
2  url = "https://pypi.org/simple"
3  verify_ssl = true
4  name = "pypi"
5
6  [packages]
7  tensorflow = "*"
8  django = "*"
9  pillow = "*"
10
11 [dev-packages]
12
13 [requires]
14 python_version = "3.10"
15
```