Taller 2

Camila Mejia Muñoz

Luis Fernando Londoño Londoño

Paola Andrea Vallejo

Juan David Martinez

Universidad EAFIT

Departamento de Sistemas e Informática

Proyecto Integrador 1

Medellín

2022

**Enlace Github:** [**https://github.com/Camila83/PI\_MLProject**](https://github.com/Camila83/PI_MLProject)

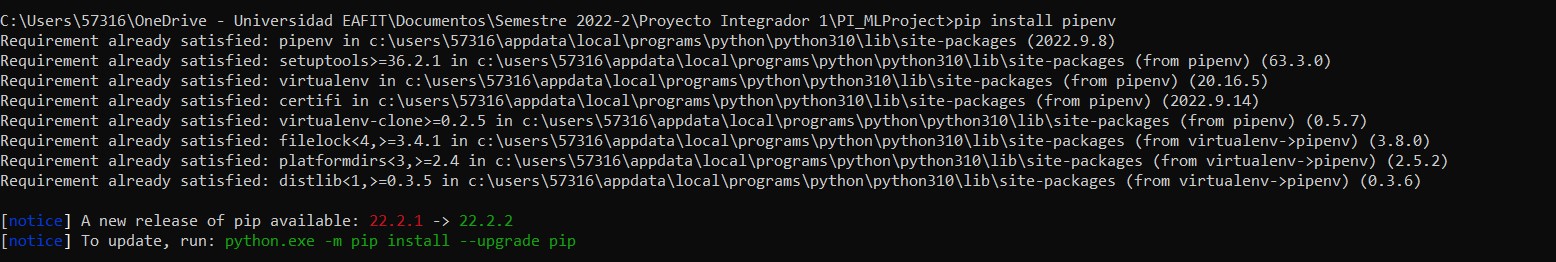
**Instalación de librerías:**

• **Crear un ambiente virtual:**

Instalar pipenv.



**Figura 1.** Instalar pipenv con el comando (pip install pipenv).

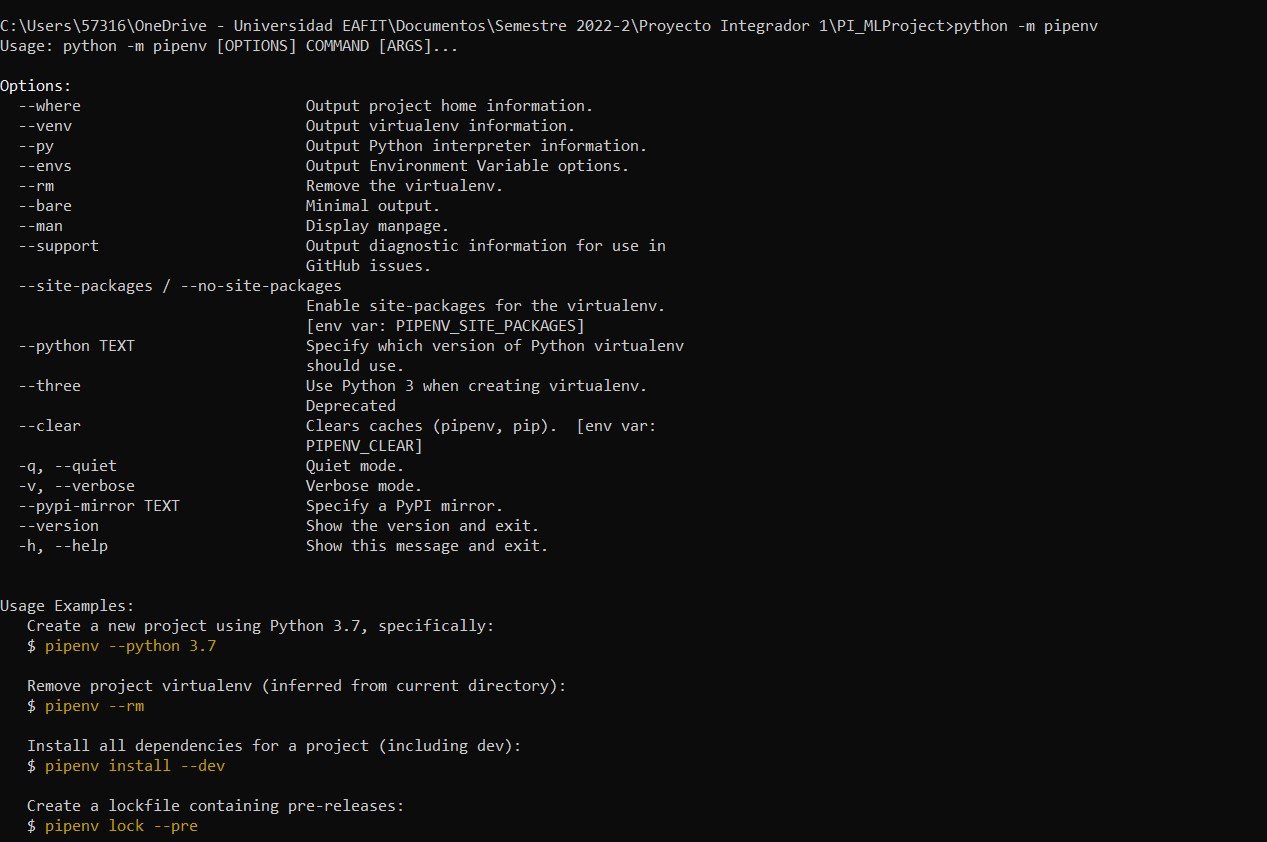


**Figura 2.** Se instaló correctamente pipenv.

Verificamos que pipenv se instaló correctamente con el siguiente comando

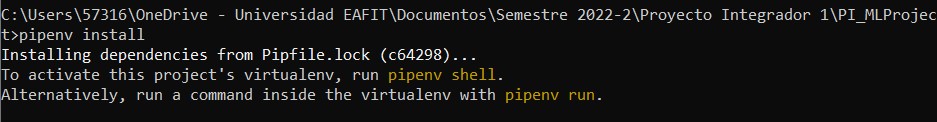


**Figura 3.** Verificamos que se instaló correctamente, con el comando python -m pipenv.



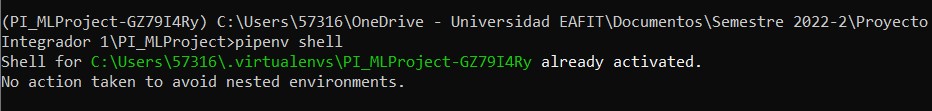
**Figura 4.** Se instaló correctamente pipenv.

Se crea el ambiente virtual con el siguiente comando.



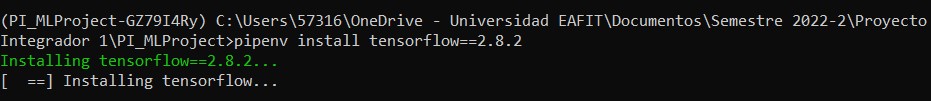
**Figura 5.** Se creó correctamente el ambiente virtual.

Se activa el ambiente virtual con el siguiente comando.

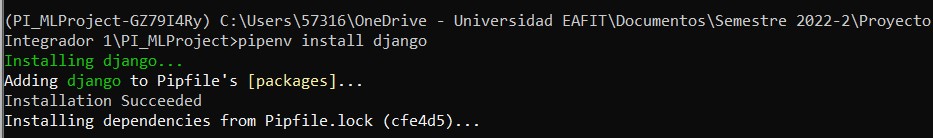


**Figura 6.** pipenv Shell, se activó correctamente el ambiente virtual.

Se instala el tensorflow .

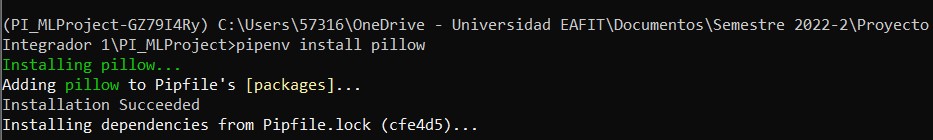


Luego se instala django en el ambiente virtual



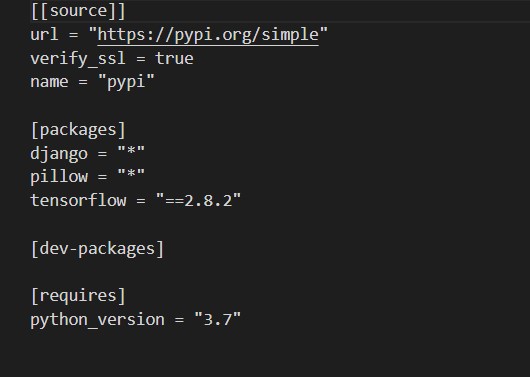
**Figura 7.** Se instaló correctamente Django.

Adicionalmente se instala la librería Pillow.



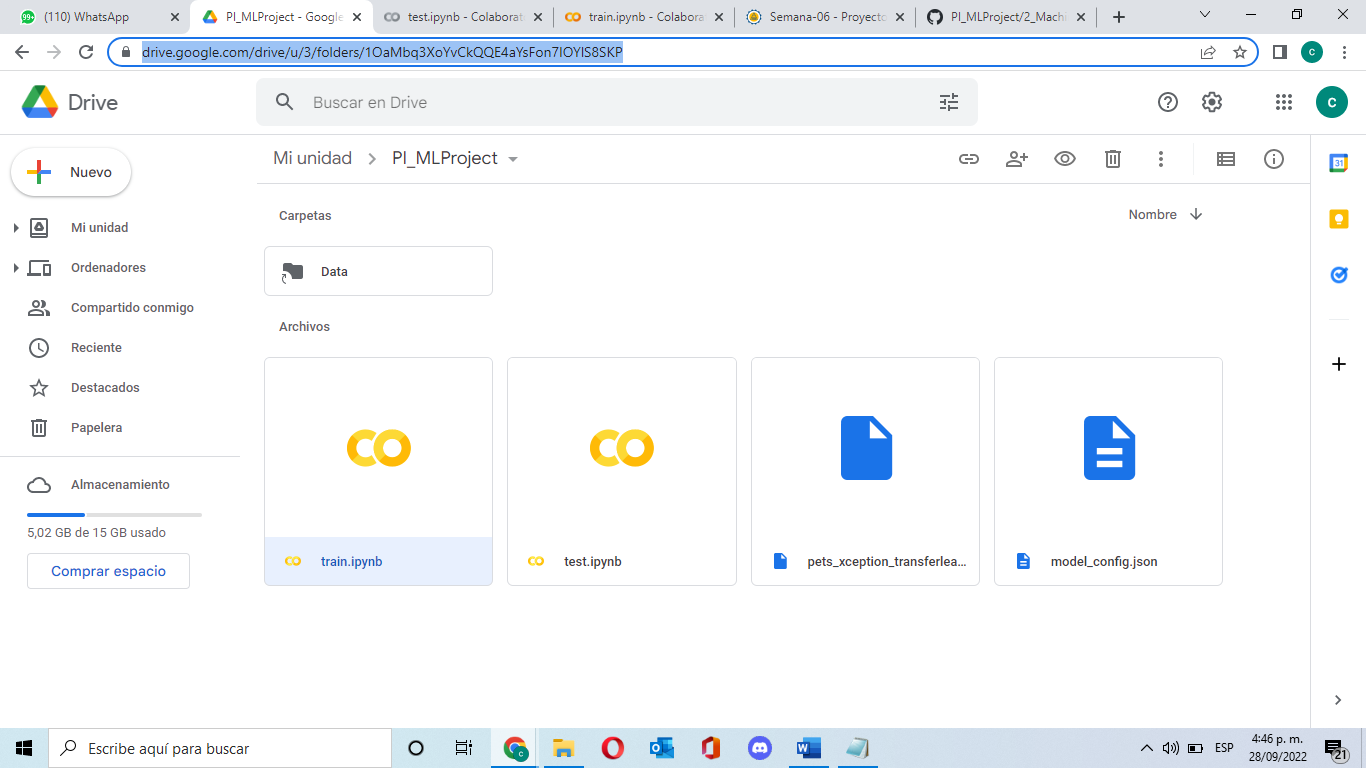
**Figura 8.** Se instaló correctamente Pillow

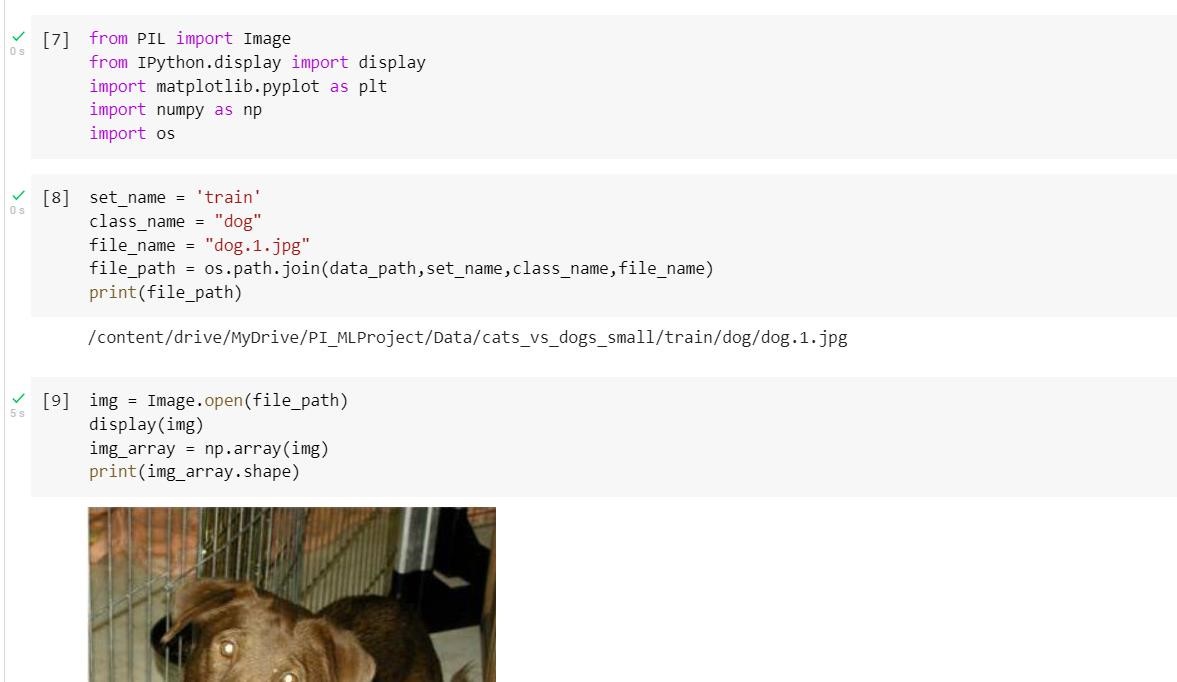
En el archivo Pipfile se muestran los paquetes instalados.

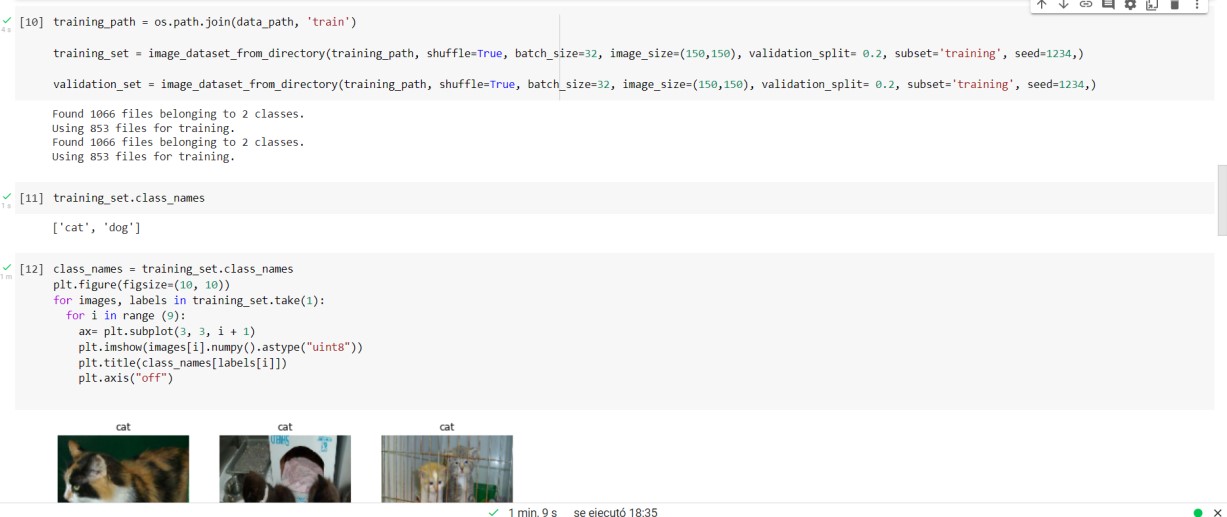


**Entrenamiento de un modelo de aprendizaje (Google Colab)**

**Enlace de Google Drive:**

<https://drive.google.com/drive/u/3/folders/1OaMbq3XoYvCkQQE4aYsFon7IOYlS8SKP>**Desarrollo de la clase train.ipynb**







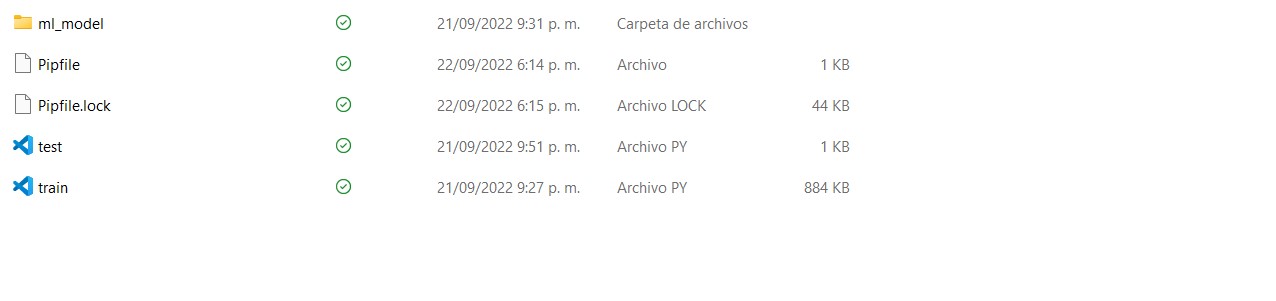
**Testeando el modelo**

* **Desarrollo de la clase test.ipynb**

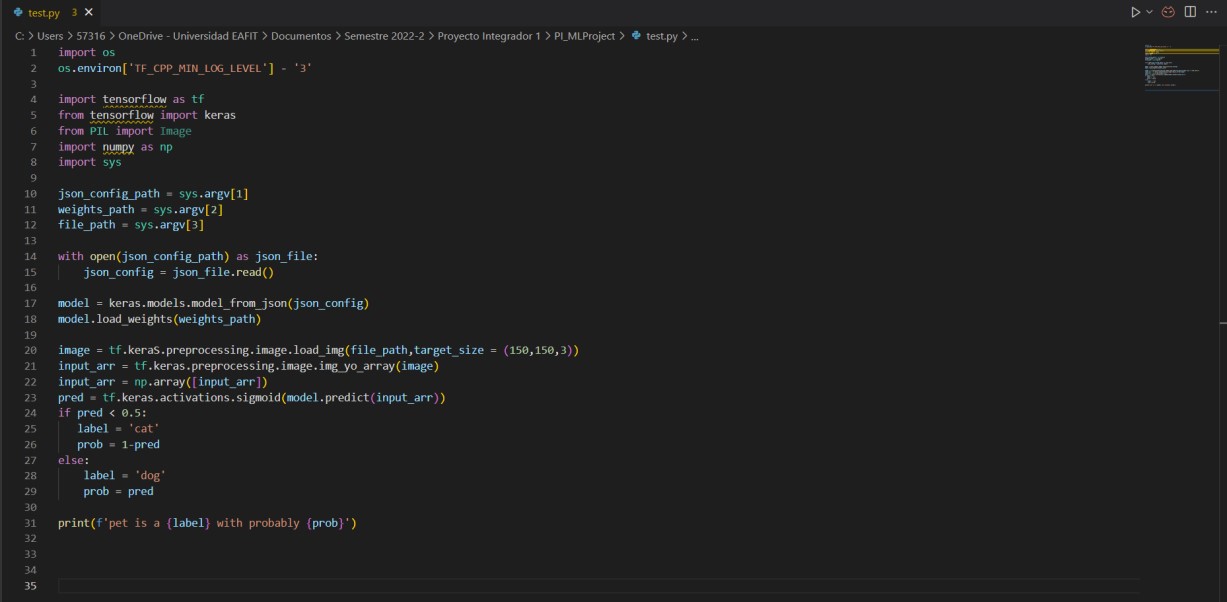




Posteriormente se descarga el archivo test.py como archivo de Python, adicionalmente, se descarga los archivos model\_config.json y pets\_xception\_trasferlearning.h5



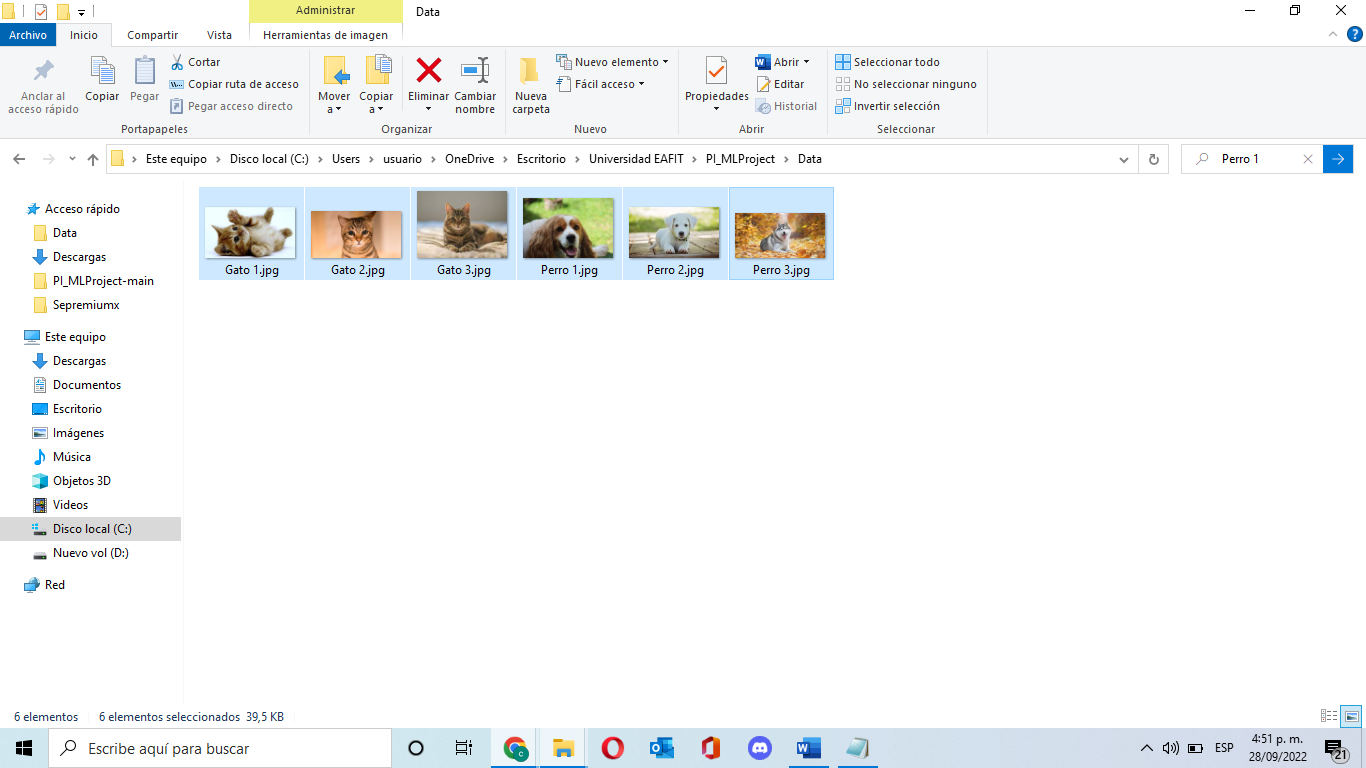
Se crea una carpeta ml\_model, esta contiene los archivos model\_config.json y pets\_xception\_transferlearning.h5



**Figura 14.** Test.py con los cambios hechos.

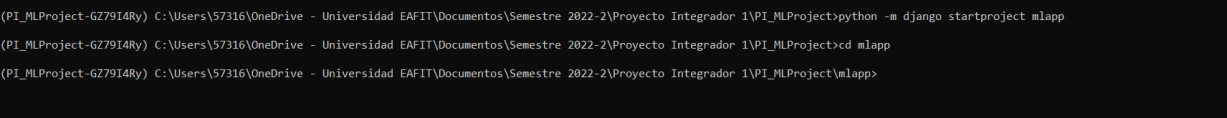
Luego se crea una subcarpeta que se va a llamar Data

# 11

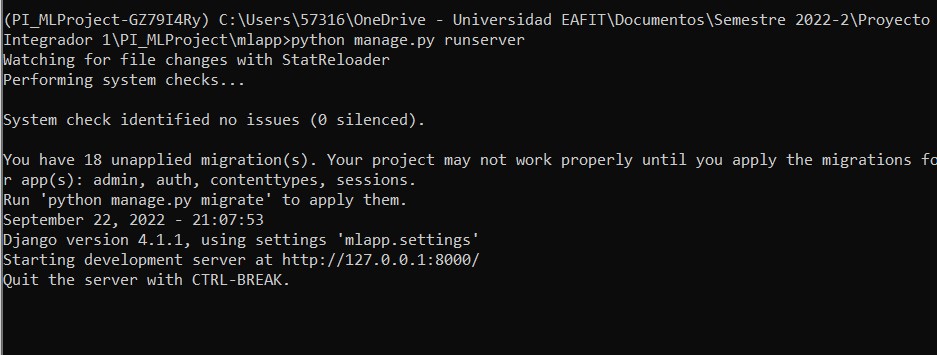
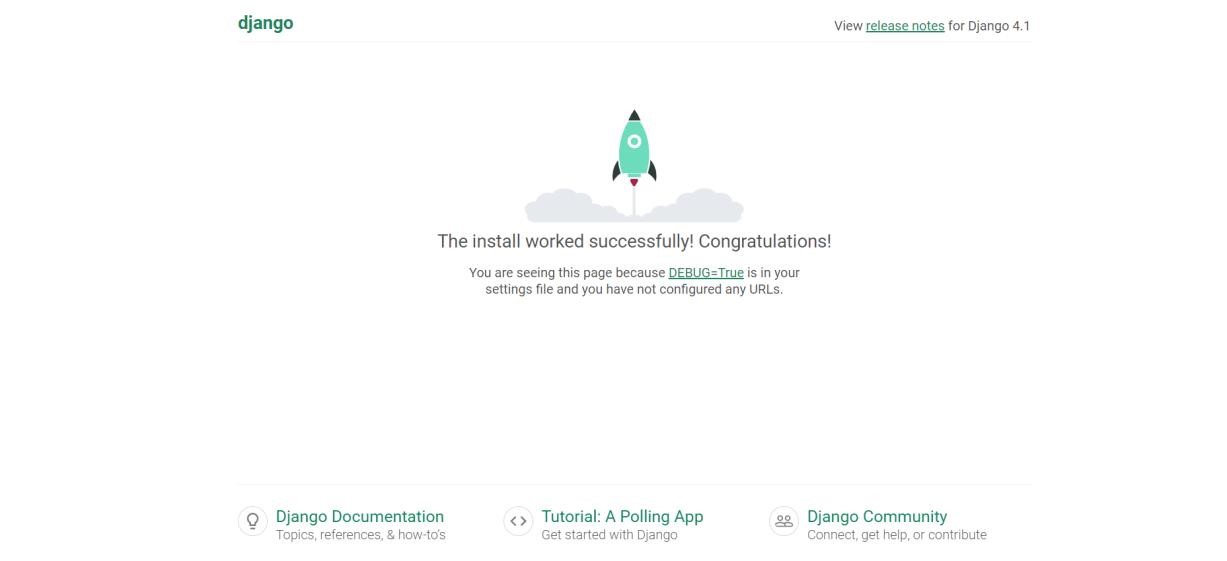


La carpeta va a contener imágenes de gatos y de perros.

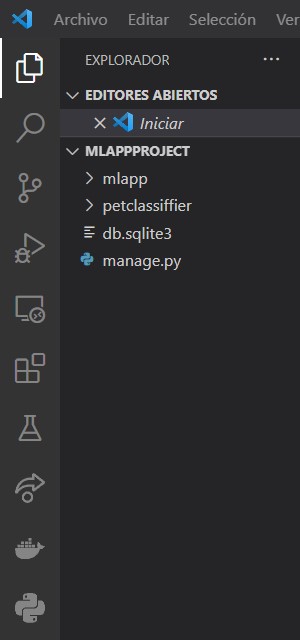
**Creando un proyecto en Django**



Se crea un proyecto en Django y se accede a la carpeta mlapp

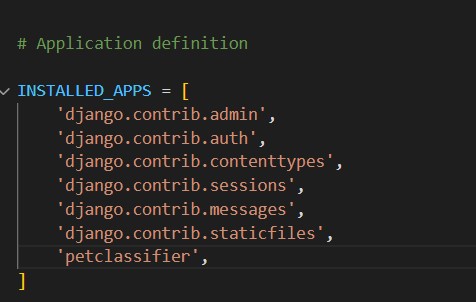
Se ejecuta el servidor de Django con el comando manage.py runserver 

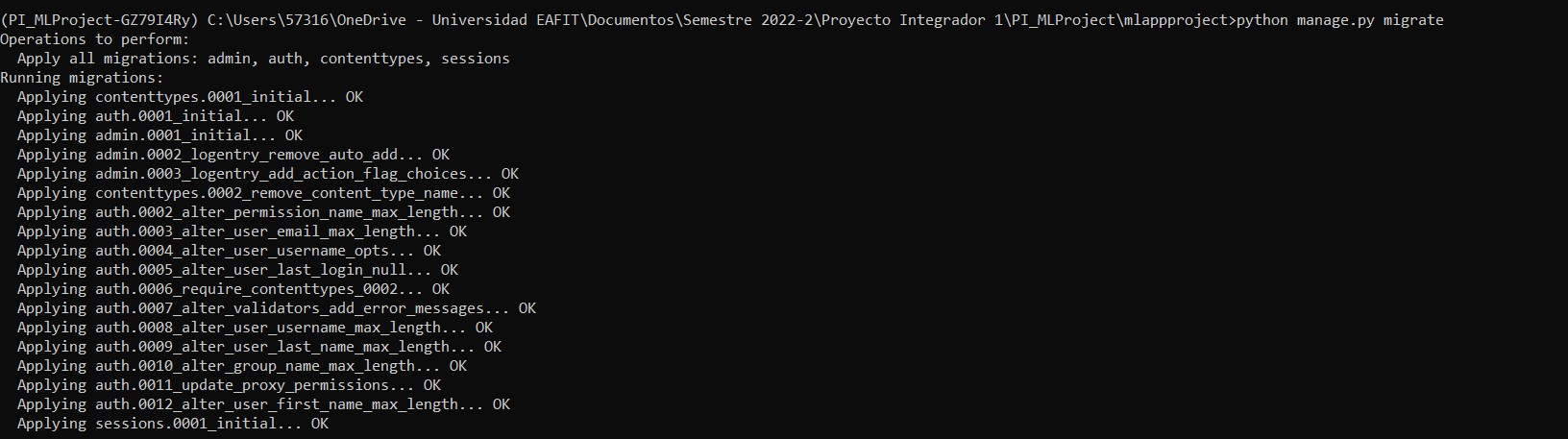
De esta manera se creó el proyecto de manera correcta.



En efecto se creó petclassiffier

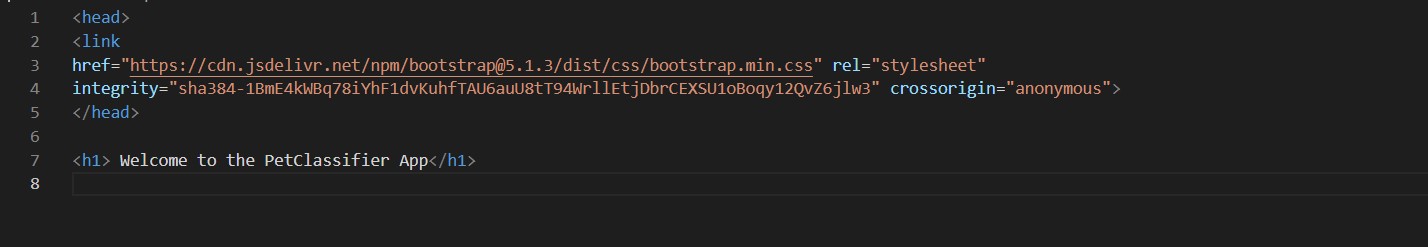
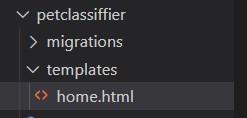
Luego, en el archivo settings.py agregar ‘petclassifier’





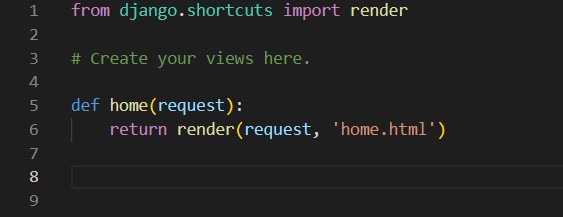
Se hace la migración con el comando python manage.py migrate.

Dentro de petclassifier se crea una carpeta llamada templates. Una vez dentro de la carpeta templates, se crea un archivo llamado home.html.

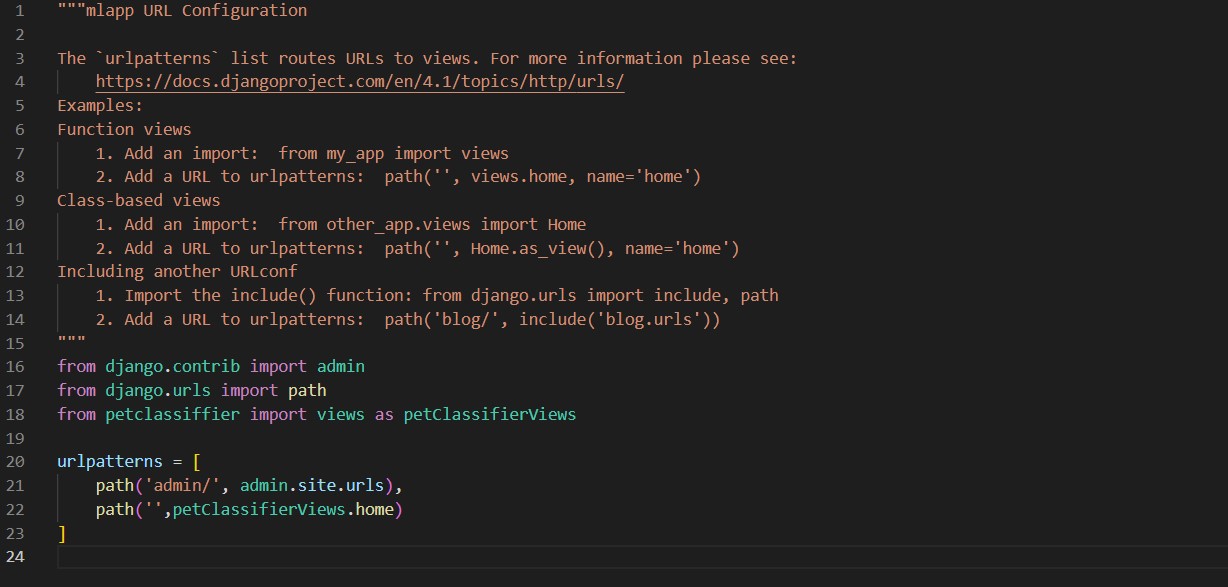


**Figura 15.** Archivo home.html

En views se crea la vista de home



En urls.py se agrega la ruta de petclassifier



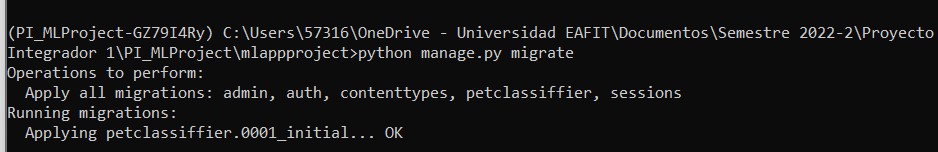
Al ejecutar el comando python manage.py runserver, al abrir el localhost:8000 se tiene lo siguiente



Se crea el modelo



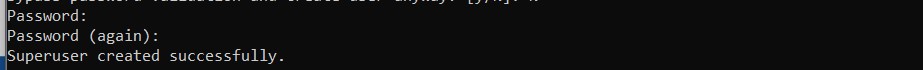
Se hace las migraciones desde la consola



Ejecutando los comandos

* Python manage.py makemigrations
* Python manage.py migrate

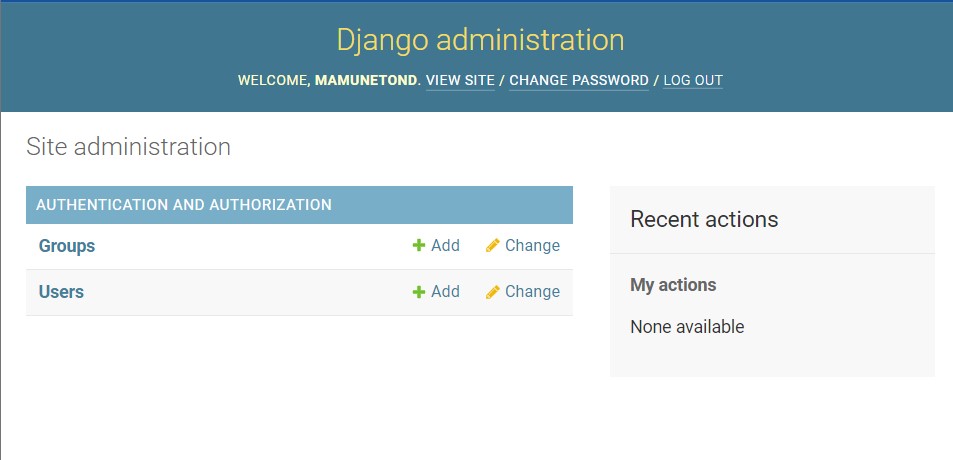
Luego en la terminal se ejecuta python manage.py createsuperuser



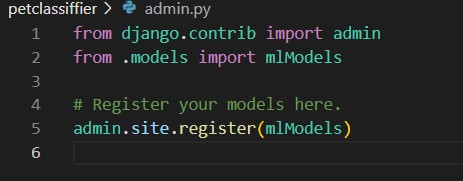
Se creó satisfactoriamente el superusuario.

Al ingresar a <http://localhost:8000/admin>

Vemos el panel de administrador de Django



Está el usuario que creamos



En admin.py agregamos el modelo.

En settings.py se agrega MEDIA\_ROOT y MEDIA\_URL

Después se habilita el servidor para almacenar datos

