Taller 2

Mario Alejandro Muñetón Durango

Luis Fernando Londoño Londoño

Paola Andrea Vallejo

Juan David Martinez

Universidad EAFIT

Departamento de Sistemas e Informática

Proyecto Integrador 1

Medellín

2022

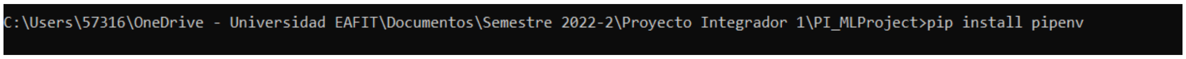
Tabla de Contenidos:

Enlace del repositorio en Github: https://github.com/mamunetond/PI_MLProject

Instalación de librerías:

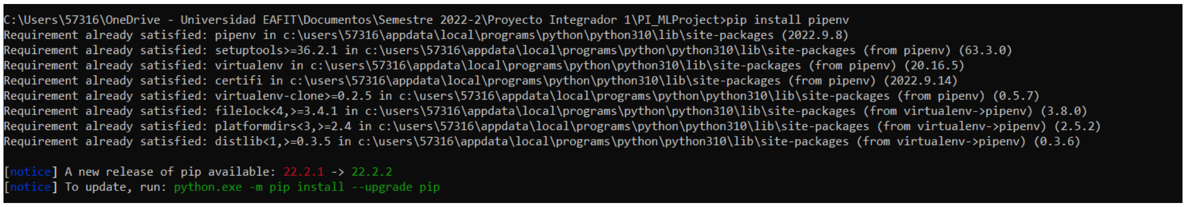
- **Crear un ambiente virtual:**

Para esto debemos de instalar pipenv, para hacer esto ejecutamos el siguiente comando en la cmd de Windows



```
C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject>pip install pipenv
```

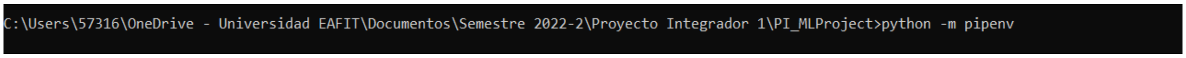
Figura 1. Instalar pipenv con el comando pip install pipenv.



```
C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject>pip install pipenv
Requirement already satisfied: pipenv in c:\users\57316\appdata\local\programs\python\python310\lib\site-packages (2022.9.8)
Requirement already satisfied: setuptools>=36.2.1 in c:\users\57316\appdata\local\programs\python\python310\lib\site-packages (from pipenv) (63.3.0)
Requirement already satisfied: virtualenv in c:\users\57316\appdata\local\programs\python\python310\lib\site-packages (from pipenv) (20.16.5)
Requirement already satisfied: certifi in c:\users\57316\appdata\local\programs\python\python310\lib\site-packages (from pipenv) (2022.9.14)
Requirement already satisfied: virtualenv-clone>=0.2.5 in c:\users\57316\appdata\local\programs\python\python310\lib\site-packages (from pipenv) (0.5.7)
Requirement already satisfied: filelock<4,>=3.4.1 in c:\users\57316\appdata\local\programs\python\python310\lib\site-packages (from virtualenv->pipenv) (3.8.0)
Requirement already satisfied: platformdirs<3,>=2.4 in c:\users\57316\appdata\local\programs\python\python310\lib\site-packages (from virtualenv->pipenv) (2.5.2)
Requirement already satisfied: distlib<1,>=0.3.5 in c:\users\57316\appdata\local\programs\python\python310\lib\site-packages (from virtualenv->pipenv) (0.3.6)

[notice] A new release of pip available: 22.2.1 -> 22.2.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

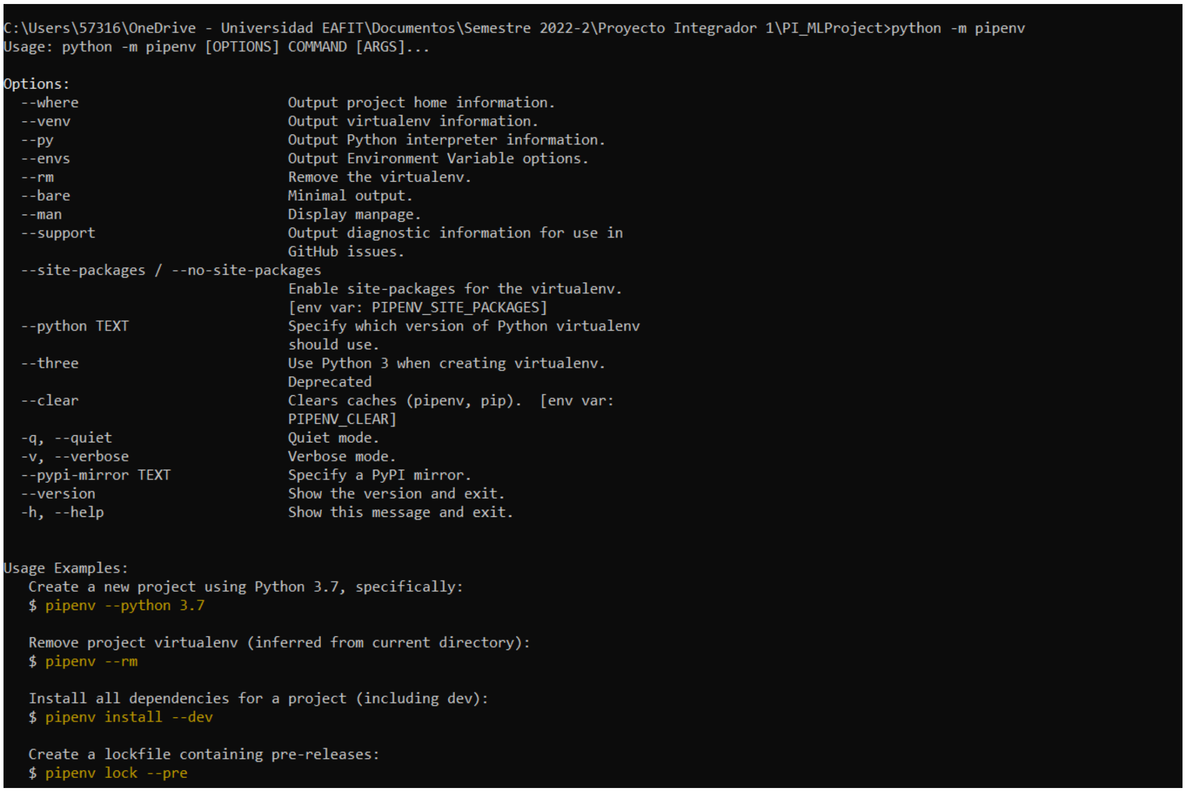
Figura 2. Se instaló correctamente pipenv.

Verificamos que pipenv se instaló correctamente con el siguiente comando



```
C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject>python -m pipenv
```

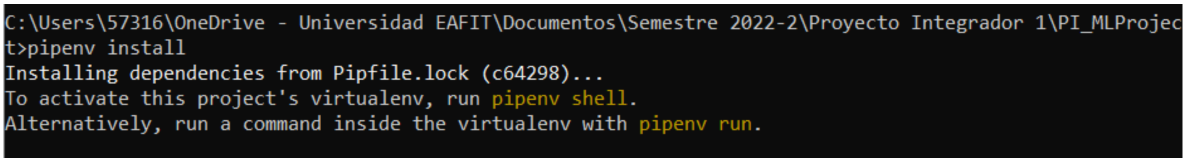
Figura 3. Verificamos que se instaló correctamente, con el comando python -m pipenv.



```
C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject>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.


Usage Examples:
   Create a new project using Python 3.7, specifically:
   $ pipenv --python 3.7

   Remove project virtualenv (inferred from current directory):
   $ pipenv --rm

   Install all dependencies for a project (including dev):
   $ pipenv install --dev

   Create a lockfile containing pre-releases:
   $ pipenv lock --pre
```

Figura 4. Se instaló correctamente pipenv.

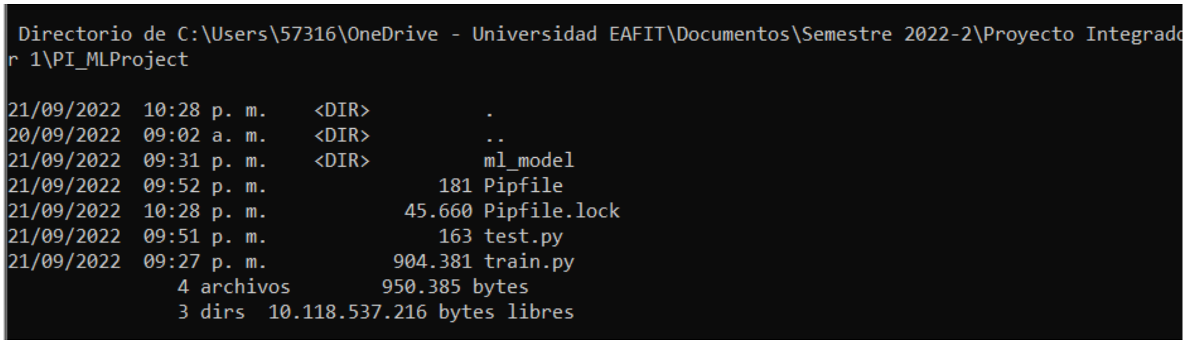
Después se crea el ambiente virtual con el siguiente comando.



```
C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProjec
t>pipenv install
Installing dependencies from Pipfile.lock (c64298)...
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
```

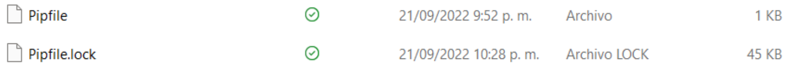
Figura 5. Se creó correctamente el ambiente virtual.

Se generan los archivos Pipfile y Pipfile.lock



```
 Directorio de C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrad
r 1\PI_MLProject

21/09/2022  10:28 p. m.    <DIR>          .
20/09/2022  09:02 a. m.    <DIR>          ..
21/09/2022  09:31 p. m.    <DIR>          ml_model
21/09/2022  09:52 p. m.               181 Pipfile
21/09/2022  10:28 p. m.            45.660 Pipfile.lock
21/09/2022  09:51 p. m.               163 test.py
21/09/2022  09:27 p. m.           904.381 train.py
               4 archivos        950.385 bytes
               3 dirs  10.118.537.216 bytes libres
```

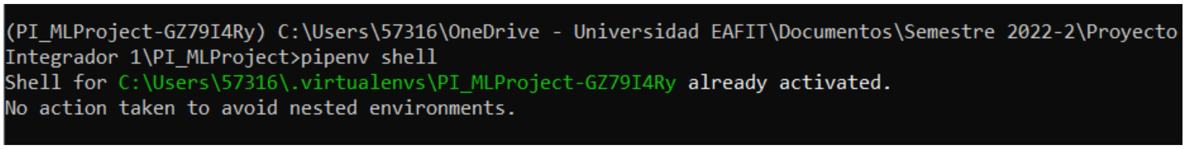
Figura 6. Con el comando dir podemos acceder a todo el contenido de la carpeta, podemos observar que efectivamente se encuentran los archivos Pipfile y Pipfile.lock.



Pipfile	✓	21/09/2022 9:52 p. m.	Archivo	1 KB
Pipfile.lock	✓	21/09/2022 10:28 p. m.	Archivo LOCK	45 KB

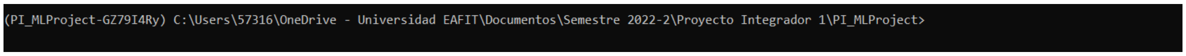
Figura 7. Archivo Pipfile y pipfile.lock

Luego se activa el ambiente virtual con el siguiente comando.



```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto
Integrador 1\PI_MLProject>pipenv shell
Shell for C:\Users\57316\.virtualenvs\PI_MLProject-GZ79I4Ry already activated.
No action taken to avoid nested environments.
```

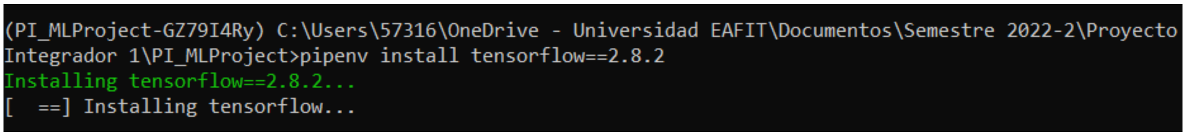
Figura 8. pipenv Shell, se activó correctamente el ambiente virtual.



```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject>
```

Figura 9. Cuando se activa el ambiente virtual, toma esta forma.

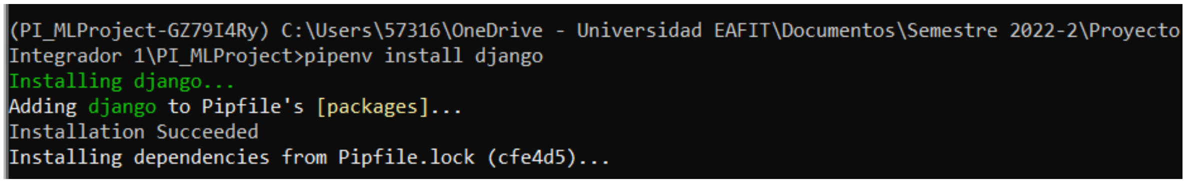
A continuación, se instala el tensorflow con el siguiente comando



```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto
Integrador 1\PI_MLProject>pipenv install tensorflow==2.8.2
Installing tensorflow==2.8.2...
[  ==] Installing tensorflow...
```

Figura 10. Con el comando pipenv install tensorflow==2.8.2 se instala la versión 2.8.2 de tensorflow

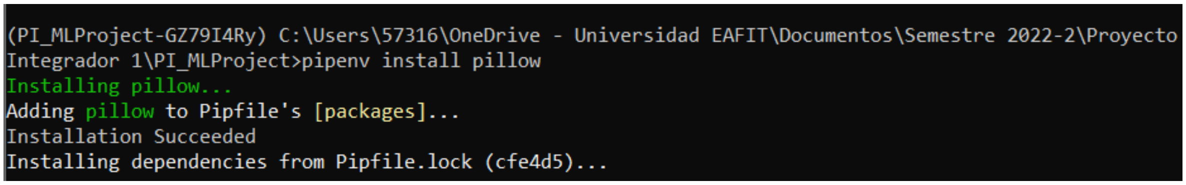
Luego se instala django en el ambiente virtual



```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto
Integrador 1\PI_MLProject>pipenv install django
Installing django...
Adding django to Pipfile's [packages]...
Installation Succeeded
Installing dependencies from Pipfile.lock (cfe4d5)...
```

Figura 11. Se instaló correctamente Django.

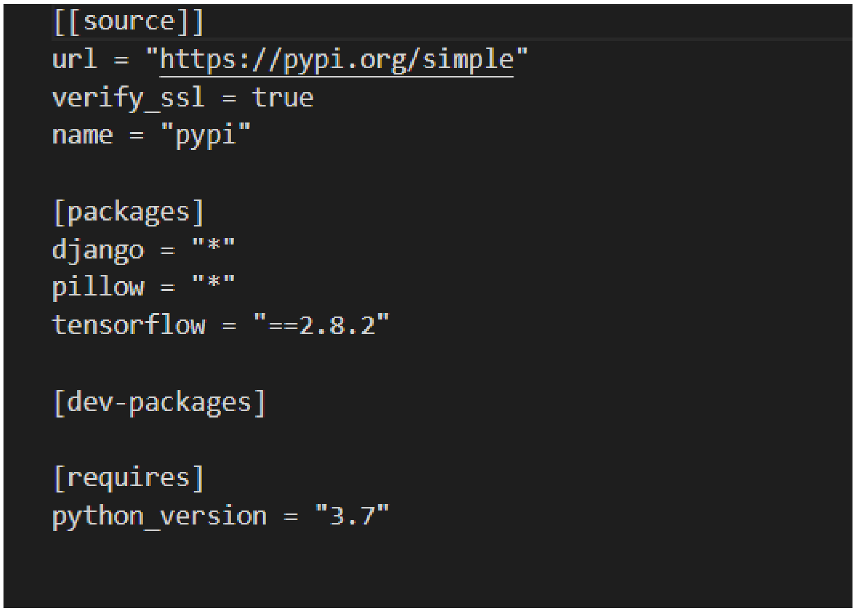
Adicionalmente se instala la librería Pillow para el manejo de imágenes.



```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto
Integrador 1\PI_MLProject>pipenv install pillow
Installing pillow...
Adding pillow to Pipfile's [packages]...
Installation Succeeded
Installing dependencies from Pipfile.lock (cfe4d5)...
```

Figura 12. Se instaló correctamente Pillow

En el archivo Pipfile se muestran los paquetes instalados.

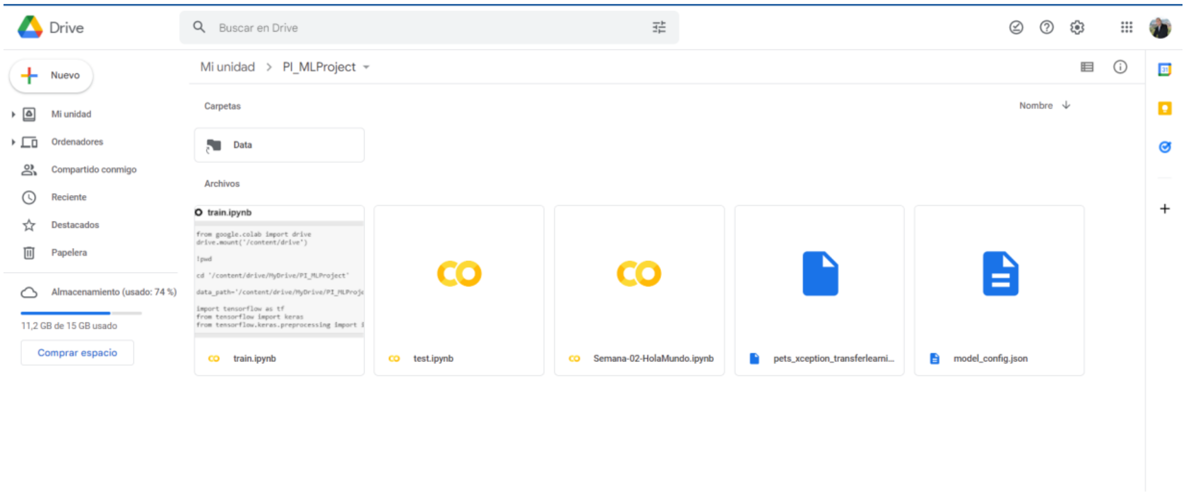


```
[[source]]
url = "https://pypi.org/simple"
verify_ssl = true
name = "pypi"

[packages]
django = "*"
pillow = "*"
tensorflow = "==2.8.2"

[dev-packages]

[requires]
python_version = "3.7"
```

Figura 13. Paquetes instalados django, pillow, tensorflow

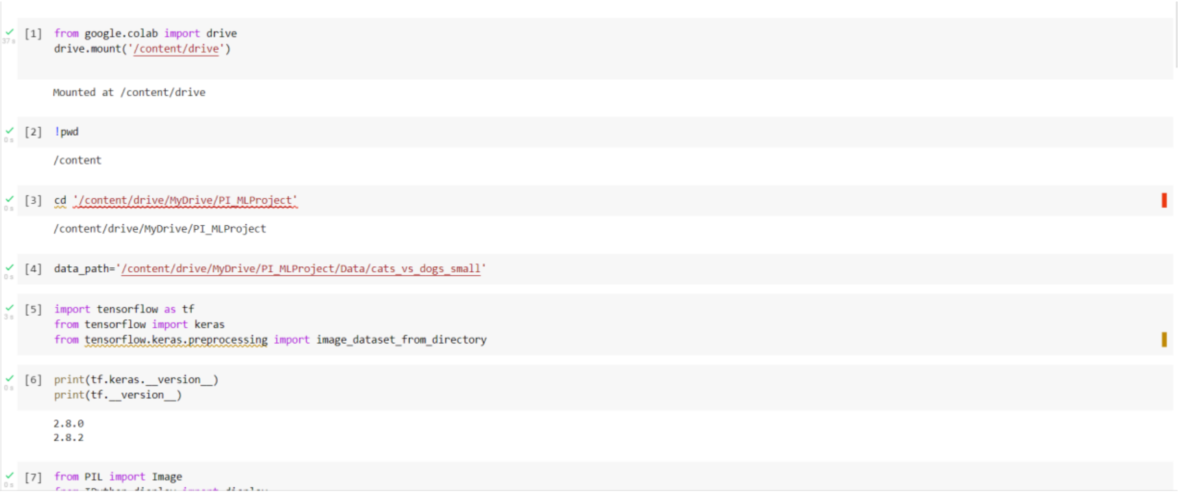
Entrenando un modelo de aprendizaje usando Google Colab

Enlace de Google Drive: https://drive.google.com/drive/u/0/folders/1hW9ACF9ToQ45YsjYmZPDG4QGryn_iW37

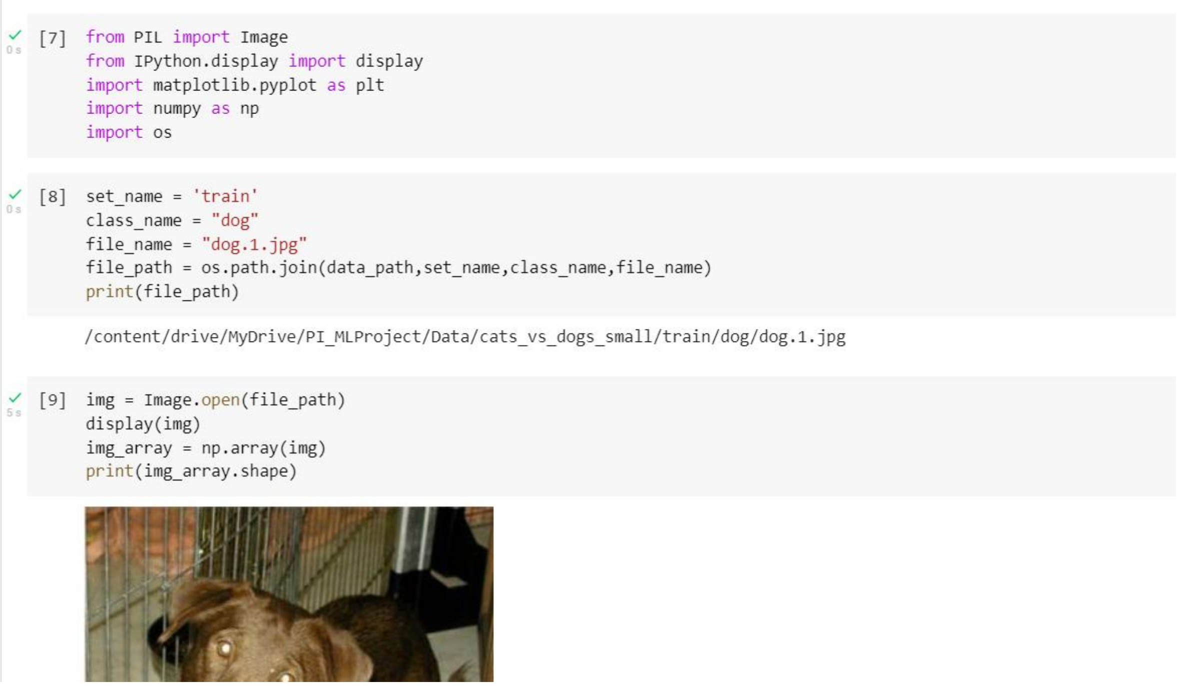


```
from google.colab import drive
drive.mount('/content/drive')

!pwd

cd '/content/drive/MyDrive/PI_MLProject'

data_path='/content/drive/MyDrive/PI_MLProj

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing import i
```

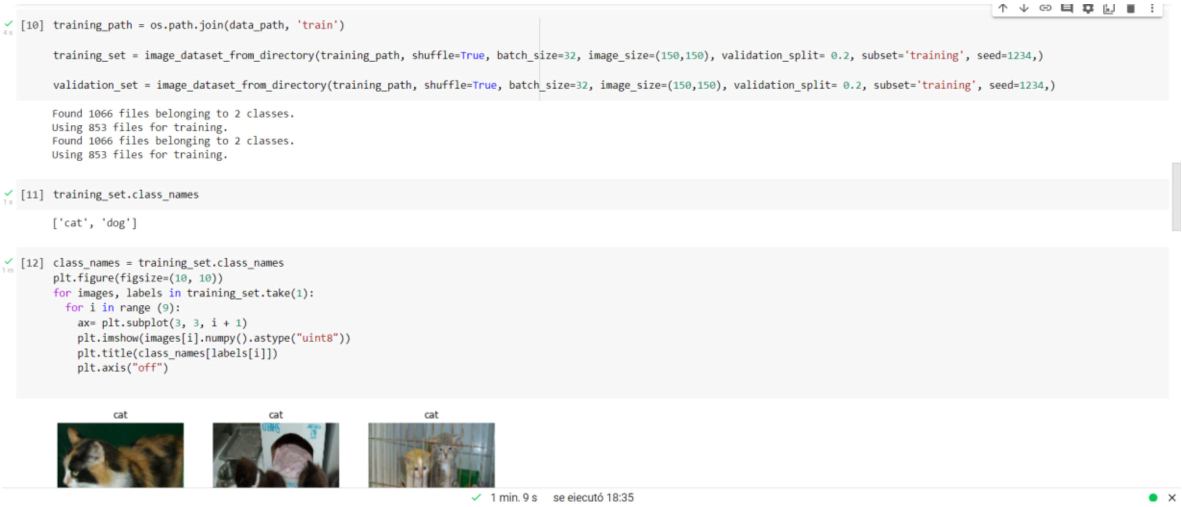
- **Desarrollo de la clase train.ipynb**



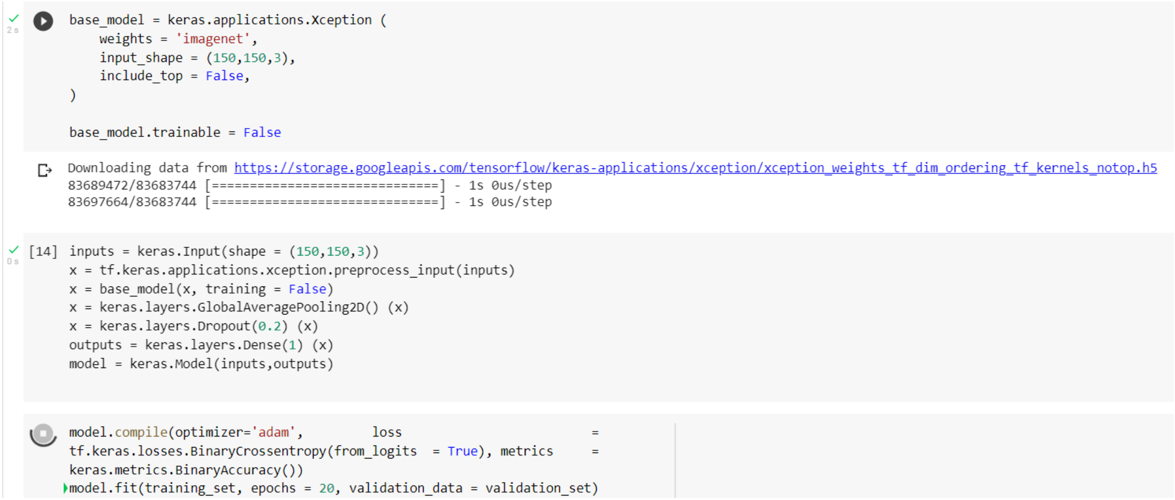
```
[1] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive

[2] !pwd

    /content

[3] cd '/content/drive/MyDrive/PI_MLProject'

    /content/drive/MyDrive/PI_MLProject

[4] data_path='/content/drive/MyDrive/PI_MLProject/Data/cats_vs_dogs_small'

[5] import tensorflow as tf
    from tensorflow import keras
    from tensorflow.keras.preprocessing import image_dataset_from_directory

[6] print(tf.keras.__version__)
    print(tf.__version__)

    2.8.0
    2.8.2

[7] from PIL import Image
```



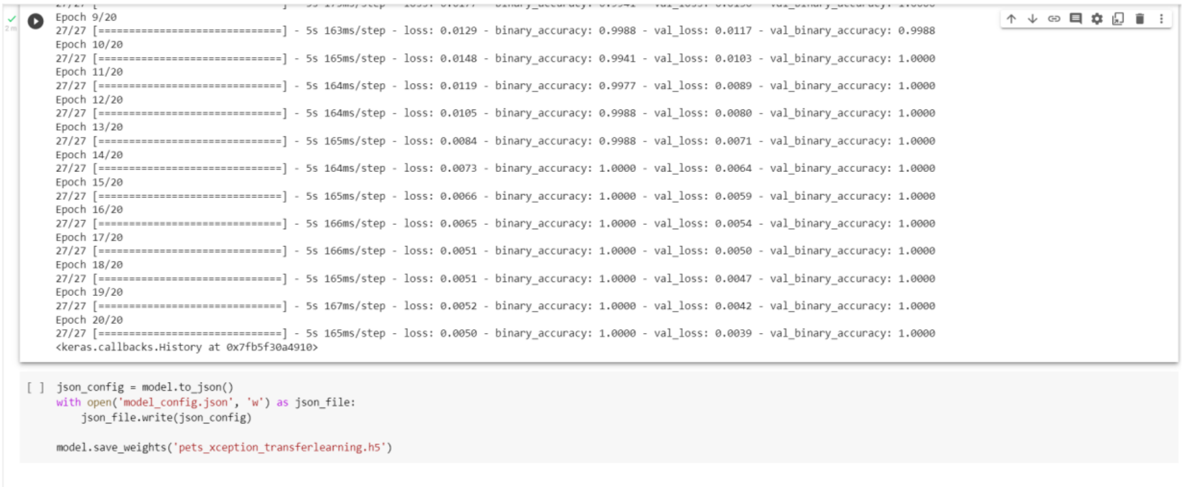
```
[7] from PIL import Image
    from IPython.display import display
    import matplotlib.pyplot as plt
    import numpy as np
    import os

[8] set_name = 'train'
    class_name = "dog"
    file_name = "dog.1.jpg"
    file_path = os.path.join(data_path,set_name,class_name,file_name)
    print(file_path)

    /content/drive/MyDrive/PI_MLProject/Data/cats_vs_dogs_small/train/dog/dog.1.jpg

[9] img = Image.open(file_path)
    display(img)
    img_array = np.array(img)
    print(img_array.shape)
```



```
[10] training_path = os.path.join(data_path, 'train')

     training_set = image_dataset_from_directory(training_path, shuffle=True, batch_size=32, image_size=(150,150), validation_split= 0.2, subset='training', seed=1234,)

     validation_set = image_dataset_from_directory(training_path, shuffle=True, batch_size=32, image_size=(150,150), validation_split= 0.2, subset='training', seed=1234,)

     Found 1066 files belonging to 2 classes.
     Using 853 files for training.
     Found 1066 files belonging to 2 classes.
     Using 853 files for training.

[11] training_set.class_names

     ['cat', 'dog']

[12] class_names = training_set.class_names
     plt.figure(figsize=(10, 10))
     for images, labels in training_set.take(1):
       for i in range (9):
         ax= plt.subplot(3, 3, i + 1)
         plt.imshow(images[i].numpy().astype("uint8"))
         plt.title(class_names[labels[i]])
         plt.axis("off")
```



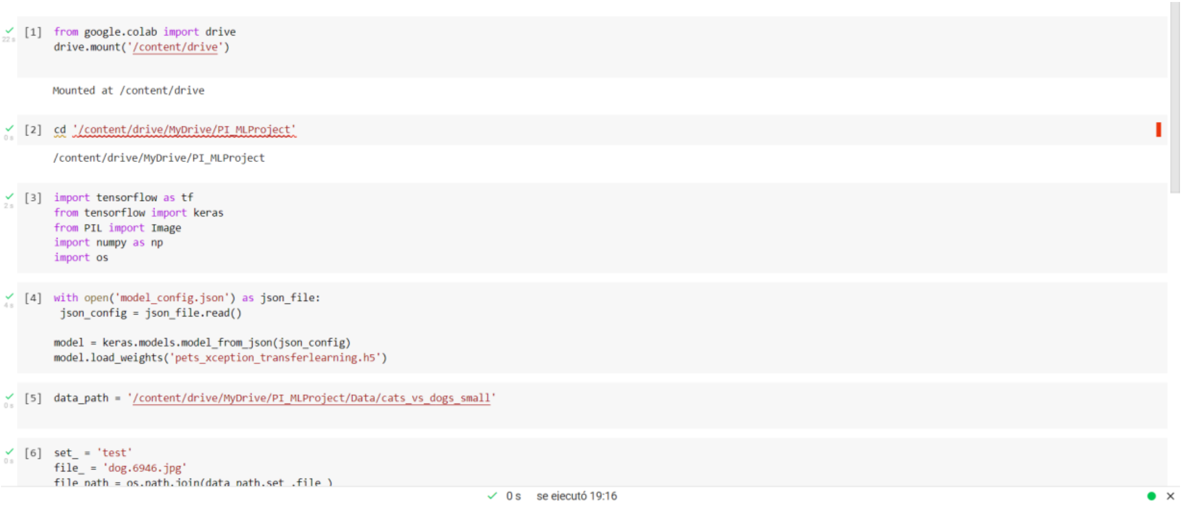
```
base_model = keras.applications.Xception (
    weights = 'imagenet',
    input_shape = (150,150,3),
    include_top = False,
)

base_model.trainable = False

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83689472/83683744 [==============================] - 1s 0us/step
83697664/83683744 [==============================] - 1s 0us/step

[14] inputs = keras.Input(shape = (150,150,3))
     x = tf.keras.applications.xception.preprocess_input(inputs)
     x = base_model(x, training = False)
     x = keras.layers.GlobalAveragePooling2D() (x)
     x = keras.layers.Dropout(0.2) (x)
     outputs = keras.layers.Dense(1) (x)
     model = keras.Model(inputs,outputs)

model.compile(optimizer='adam',         loss                      =
tf.keras.losses.BinaryCrossentropy(from_logits  = True), metrics     =
keras.metrics.BinaryAccuracy())
model.fit(training_set, epochs = 20, validation_data = validation_set)
```



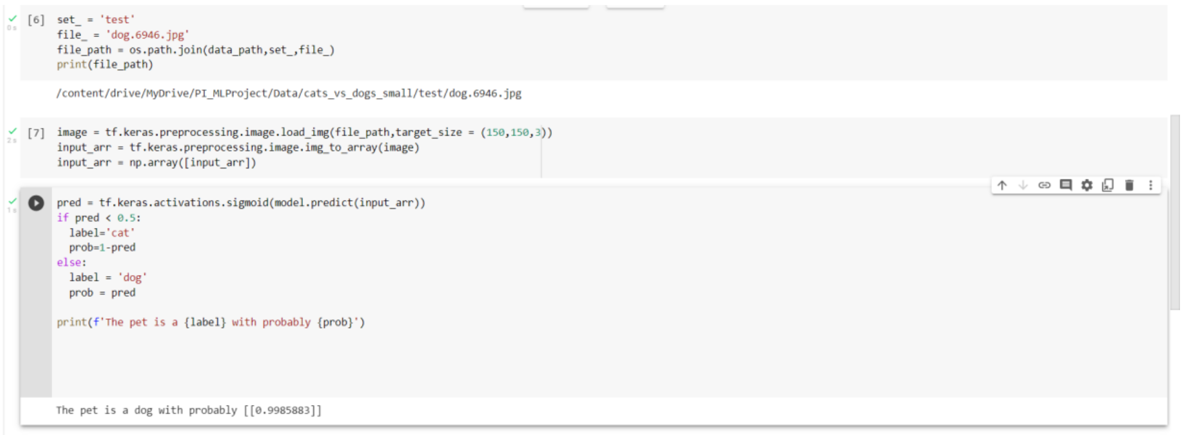
```
Epoch 9/20
27/27 [==============================] - 5s 163ms/step - loss: 0.0129 - binary_accuracy: 0.9988 - val_loss: 0.0117 - val_binary_accuracy: 0.9988
Epoch 10/20
27/27 [==============================] - 5s 165ms/step - loss: 0.0148 - binary_accuracy: 0.9941 - val_loss: 0.0103 - val_binary_accuracy: 1.0000
Epoch 11/20
27/27 [==============================] - 5s 164ms/step - loss: 0.0119 - binary_accuracy: 0.9977 - val_loss: 0.0089 - val_binary_accuracy: 1.0000
Epoch 12/20
27/27 [==============================] - 5s 164ms/step - loss: 0.0105 - binary_accuracy: 0.9988 - val_loss: 0.0080 - val_binary_accuracy: 1.0000
Epoch 13/20
27/27 [==============================] - 5s 165ms/step - loss: 0.0084 - binary_accuracy: 0.9988 - val_loss: 0.0071 - val_binary_accuracy: 1.0000
Epoch 14/20
27/27 [==============================] - 5s 164ms/step - loss: 0.0073 - binary_accuracy: 1.0000 - val_loss: 0.0064 - val_binary_accuracy: 1.0000
Epoch 15/20
27/27 [==============================] - 5s 165ms/step - loss: 0.0066 - binary_accuracy: 1.0000 - val_loss: 0.0059 - val_binary_accuracy: 1.0000
Epoch 16/20
27/27 [==============================] - 5s 166ms/step - loss: 0.0065 - binary_accuracy: 1.0000 - val_loss: 0.0054 - val_binary_accuracy: 1.0000
Epoch 17/20
27/27 [==============================] - 5s 166ms/step - loss: 0.0051 - binary_accuracy: 1.0000 - val_loss: 0.0050 - val_binary_accuracy: 1.0000
Epoch 18/20
27/27 [==============================] - 5s 165ms/step - loss: 0.0051 - binary_accuracy: 1.0000 - val_loss: 0.0047 - val_binary_accuracy: 1.0000
Epoch 19/20
27/27 [==============================] - 5s 167ms/step - loss: 0.0052 - binary_accuracy: 1.0000 - val_loss: 0.0042 - val_binary_accuracy: 1.0000
Epoch 20/20
27/27 [==============================] - 5s 165ms/step - loss: 0.0050 - binary_accuracy: 1.0000 - val_loss: 0.0039 - val_binary_accuracy: 1.0000
<keras.callbacks.History at 0x7fb5f30a4910>

[ ] json_config = model.to_json()
    with open('model_config.json', 'w') as json_file:
        json_file.write(json_config)

    model.save_weights('pets_xception_transferlearning.h5')
```

Testeando el modelo

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

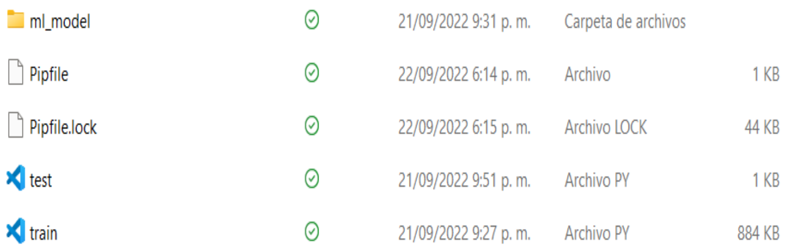


```
[1] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive

[2] cd '/content/drive/MyDrive/PI_MLProject'

    /content/drive/MyDrive/PI_MLProject

[3] import tensorflow as tf
    from tensorflow import keras
    from PIL import Image
    import numpy as np
    import os

[4] with open('model_config.json') as json_file:
     json_config = json_file.read()

    model = keras.models.model_from_json(json_config)
    model.load_weights('pets_xception_transferlearning.h5')

[5] data_path = '/content/drive/MyDrive/PI_MLProject/Data/cats_vs_dogs_small'

[6] set_ = 'test'
    file_ = 'dog.6946.jpg'
    file_path = os.path.join(data_path,set_,file_)
```



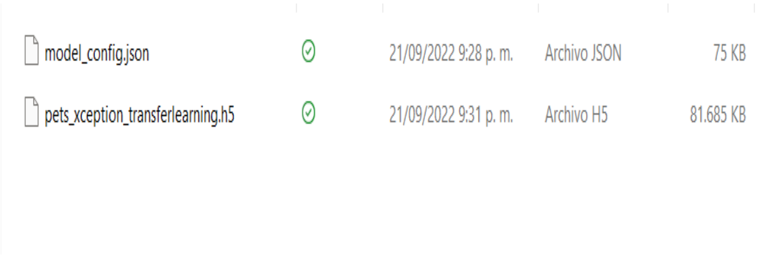
```
[6] set_ = 'test'
    file_ = 'dog.6946.jpg'
    file_path = os.path.join(data_path,set_,file_)
    print(file_path)

    /content/drive/MyDrive/PI_MLProject/Data/cats_vs_dogs_small/test/dog.6946.jpg

[7] image = tf.keras.preprocessing.image.load_img(file_path,target_size = (150,150,3))
    input_arr = tf.keras.preprocessing.image.img_to_array(image)
    input_arr = np.array([input_arr])

    pred = tf.keras.activations.sigmoid(model.predict(input_arr))
    if pred < 0.5:
      label='cat'
      prob=1-pred
    else:
      label = 'dog'
      prob = pred

    print(f'The pet is a {label} with probably {prob}')

    The pet is a dog with probably [[0.9985883]]
```

Posteriormente se descarga el archivo test.py como archivo de Python, adicionalmente, se descarga los archivos model_config.json y pets_xception_trasferlearning.h5

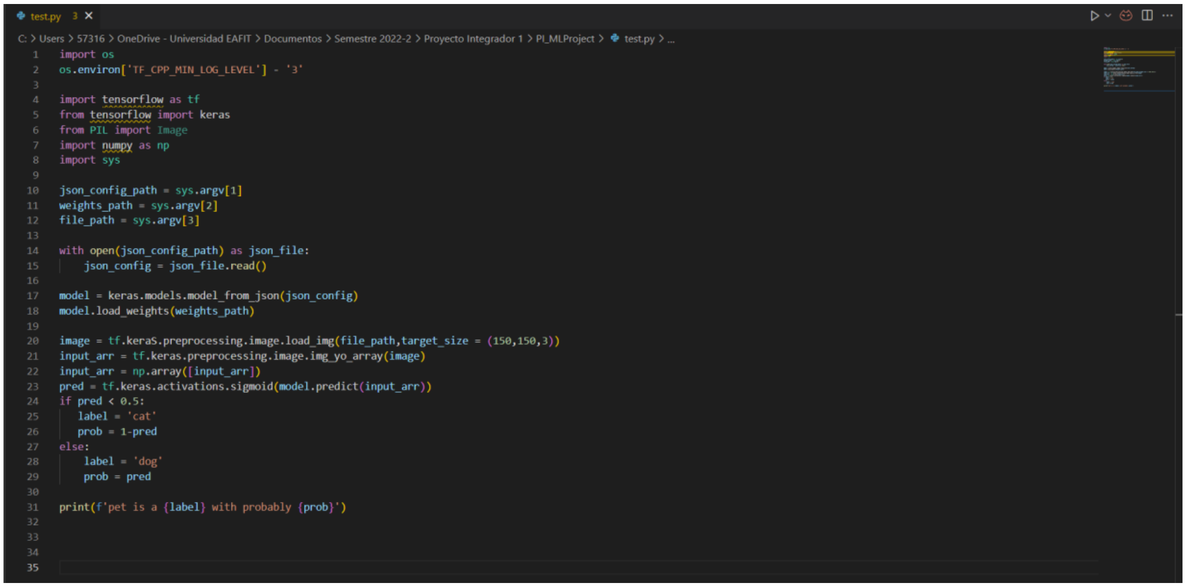


ml_model	21/09/2022 9:31 p. m.	Carpeta de archivos	
Pipfile	22/09/2022 6:14 p. m.	Archivo	1 KB
Pipfile.lock	22/09/2022 6:15 p. m.	Archivo LOCK	44 KB
test	21/09/2022 9:51 p. m.	Archivo PY	1 KB
train	21/09/2022 9:27 p. m.	Archivo PY	884 KB

Se crea una carpeta ml_model, esta contiene los archivos model_config.json y pets_xception_transferlearning.h5



model_config.json	21/09/2022 9:28 p. m.	Archivo JSON	75 KB
pets_xception_transferlearning.h5	21/09/2022 9:31 p. m.	Archivo H5	81.685 KB

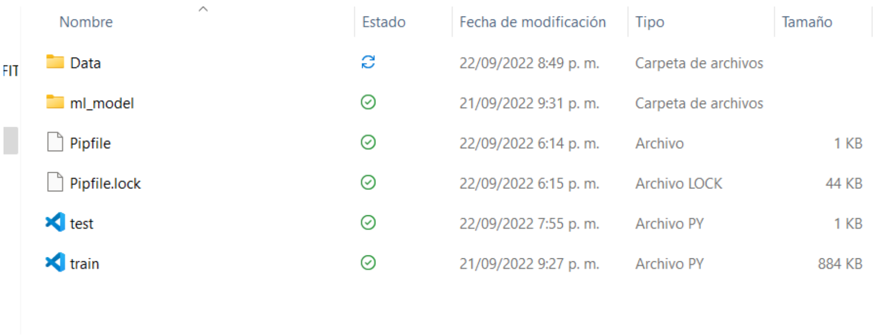


C: > Users > 57316 > OneDrive - Universidad EAFIT > Documentos > Semestre 2022-2 > Proyecto Integrador 1 > PI_MLProject > test.py > ...

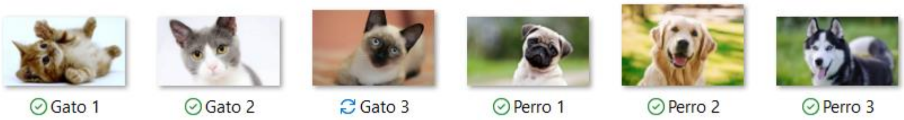
```python
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] - '3'

import tensorflow as tf
from tensorflow import keras
from PIL import Image
import numpy as np
import sys

json_config_path = sys.argv[1]
weights_path = sys.argv[2]
file_path = sys.argv[3]

with open(json_config_path) as json_file:
    json_config = json_file.read()

model = keras.models.model_from_json(json_config)
model.load_weights(weights_path)

image = tf.keraS.preprocessing.image.load_img(file_path,target_size = (150,150,3))
input_arr = tf.keras.preprocessing.image.img_yo_array(image)
input_arr = np.array([input_arr])
pred = tf.keras.activations.sigmoid(model.predict(input_arr))
if pred < 0.5:
   label = 'cat'
   prob = 1-pred
else:
    label = 'dog'
    prob = pred

print(f'pet is a {label} with probably {prob}')
```

Figura 14. Test.py con los cambios hechos.

Luego se crea una subcarpeta que se va a llamar Data

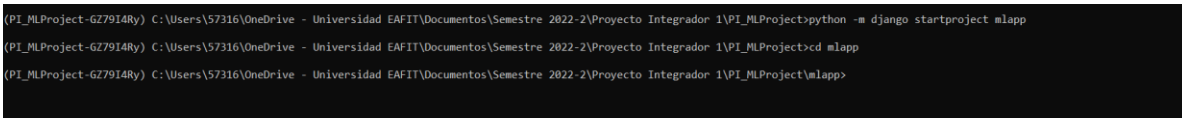


Nombre	Estado	Fecha de modificación	Tipo	Tamaño
Data	🔄	22/09/2022 8:49 p. m.	Carpeta de archivos	
ml_model	✓	21/09/2022 9:31 p. m.	Carpeta de archivos	
Pipfile	✓	22/09/2022 6:14 p. m.	Archivo	1 KB
Pipfile.lock	✓	22/09/2022 6:15 p. m.	Archivo LOCK	44 KB
test	✓	22/09/2022 7:55 p. m.	Archivo PY	1 KB
train	✓	21/09/2022 9:27 p. m.	Archivo PY	884 KB



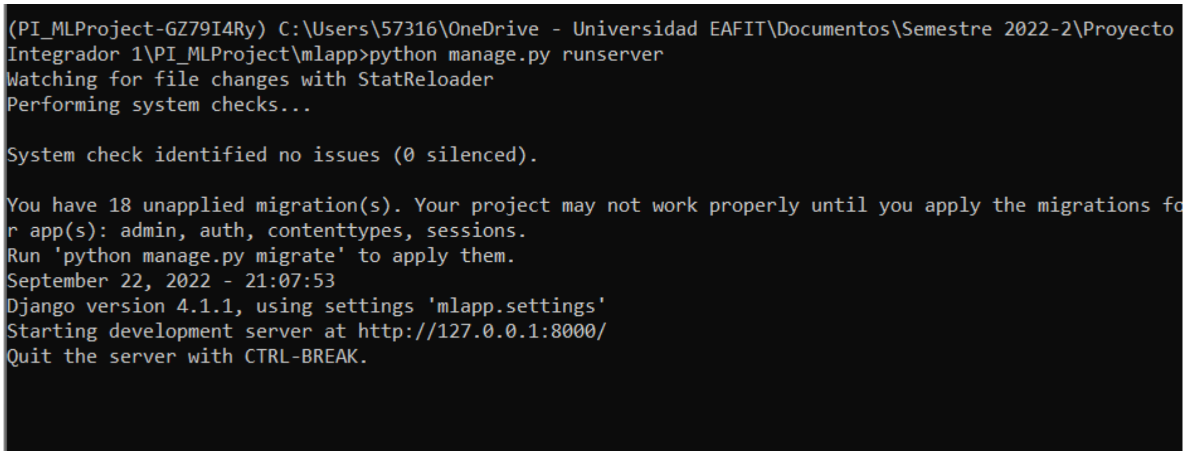
La carpeta va a contener imágenes de gatos y de perros.

Creando un proyecto en Django



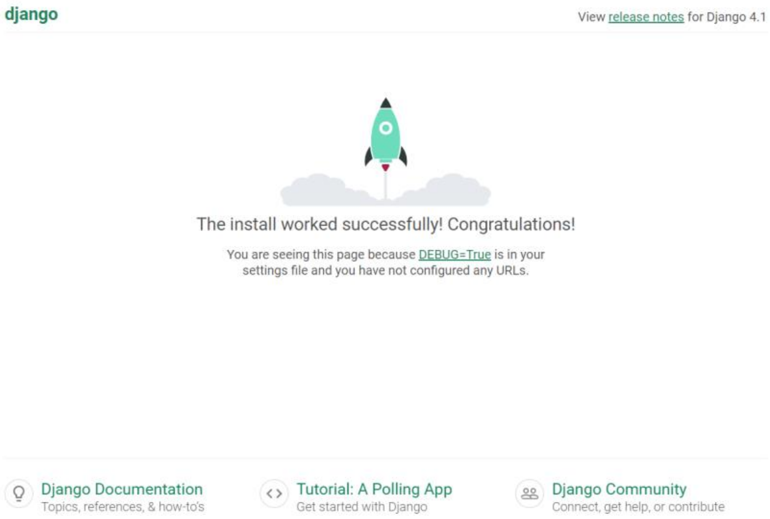
```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject>python -m django startproject mlapp

(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject>cd mlapp

(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject\mlapp>
```

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



```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto
Integrador 1\PI_MLProject\mlapp>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations fo
r app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
September 22, 2022 - 21:07:53
Django version 4.1.1, using settings 'mlapp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

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


django
View release notes for Django 4.1
Django Documentation
Topics, references, & how-to's
Tutorial: A Polling App
Get started with Django
Django Community
Connect, get help, or contribute

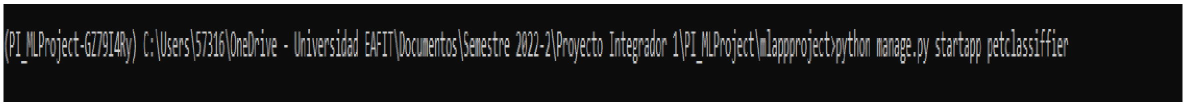
The install worked successfully! Congratulations!

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

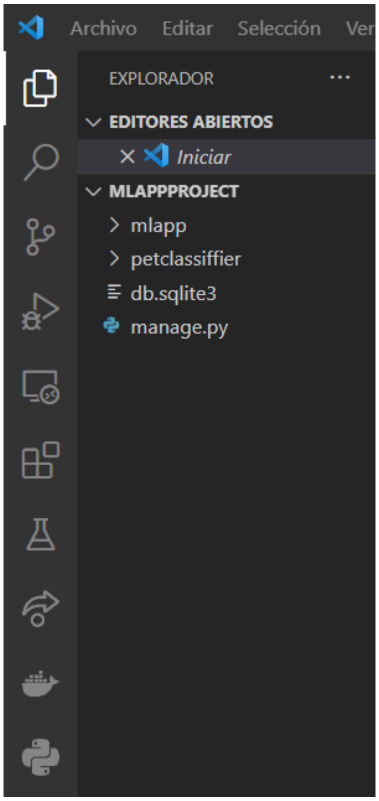
De esta manera se creó el proyecto de manera correcta.

Cambiamos el nombre de la carpeta mlapp por mlappproject.

Dentro de la carpeta del proyecto, crear la aplicación petclasiffier

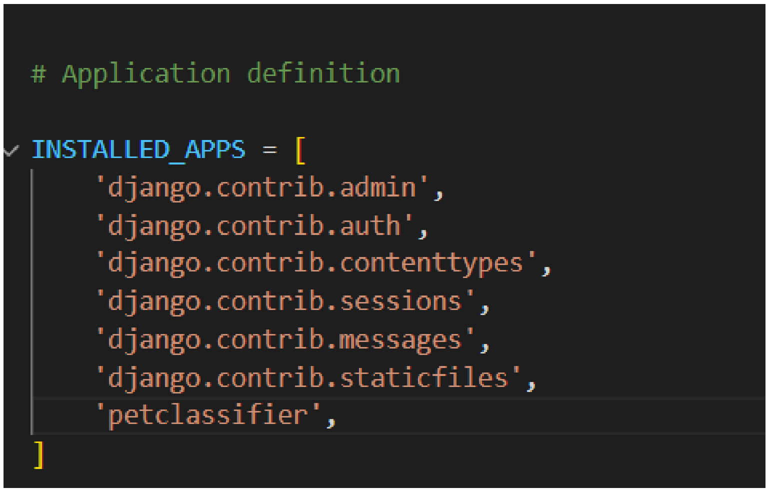


```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject\mlappproject>python manage.py startapp petclassiffier
```

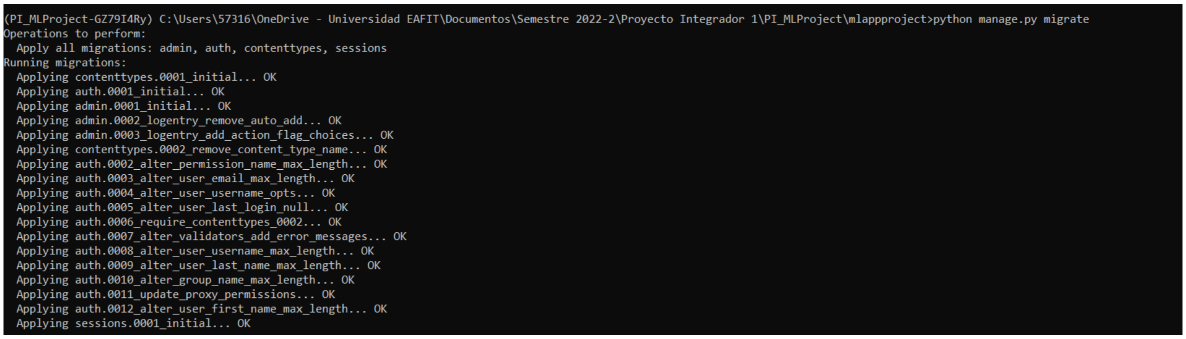


En efecto se creó petclassiffier

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



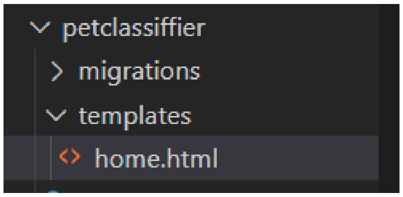
```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'petclassifier',
]
```

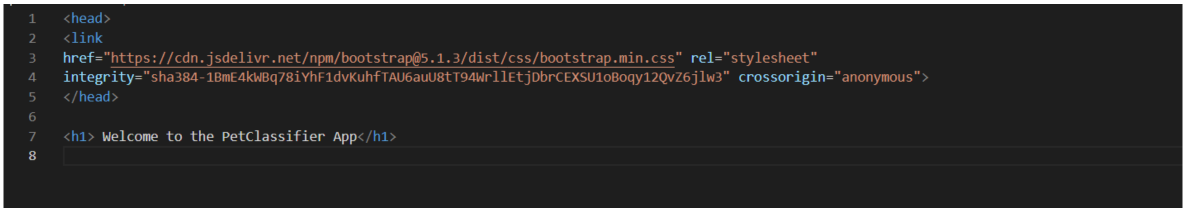


```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto Integrador 1\PI_MLProject\mlappproject>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

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.

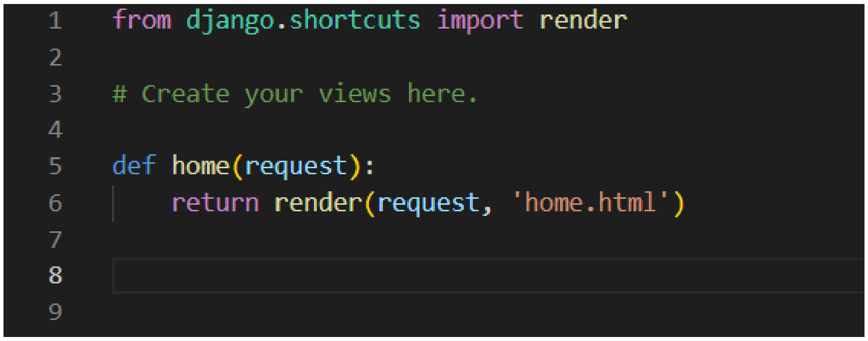




```html
<head>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrllEtjDbrCEXSU1oBoqy12QvZ6jlw3" crossorigin="anonymous">
</head>

<h1> Welcome to the PetClassifier App</h1>
```

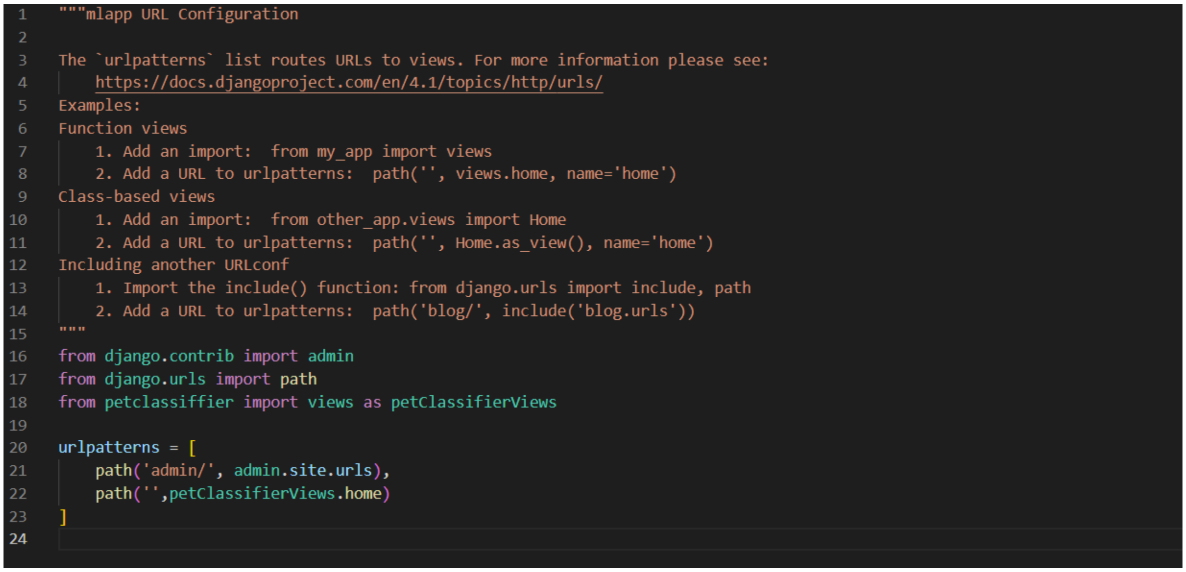
Figura 15. Archivo home.html

En views se crea la vista de home



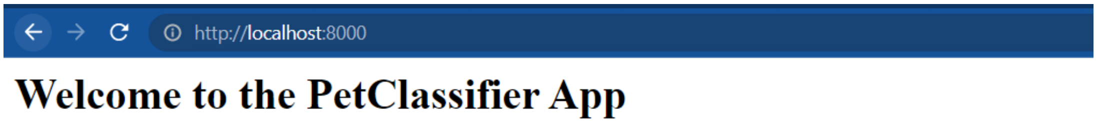
```python
from django.shortcuts import render

# Create your views here.

def home(request):
    return render(request, 'home.html')
```

En urls.py se agrega la ruta de petclassifier



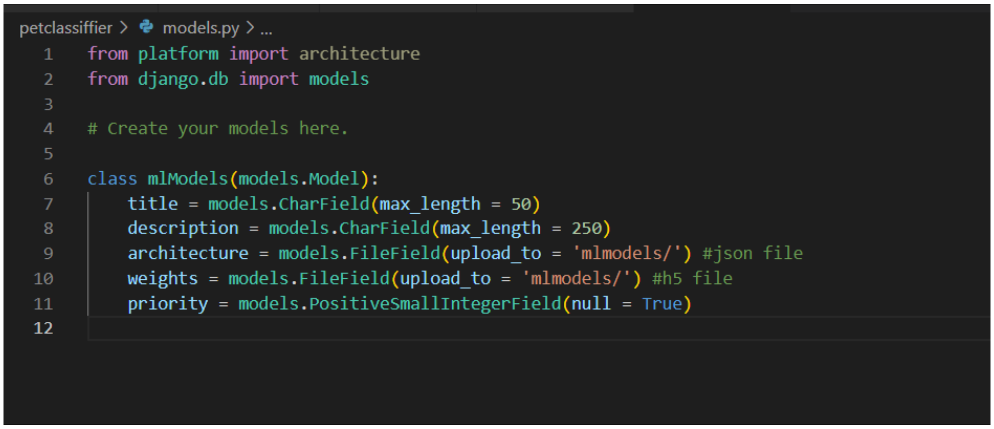
```
"""mlapp URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from petclassiffier import views as petClassifierViews

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',petClassifierViews.home)
]

```

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



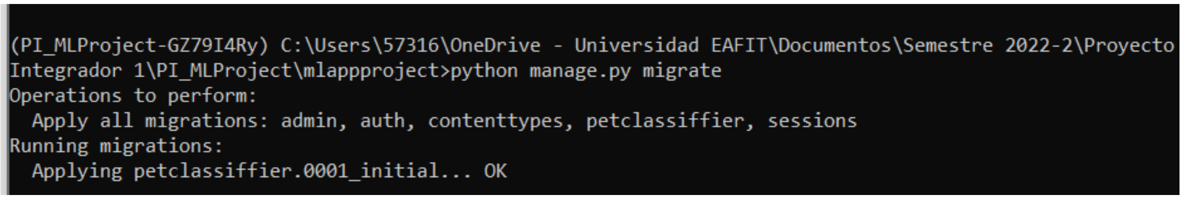
Welcome to the PetClassifier App

Se crea el modelo



```
from platform import architecture
from django.db import models

# Create your models here.

class mlModels(models.Model):
    title = models.CharField(max_length = 50)
    description = models.CharField(max_length = 250)
    architecture = models.FileField(upload_to = 'mlmodels/') #json file
    weights = models.FileField(upload_to = 'mlmodels/') #h5 file
    priority = models.PositiveSmallIntegerField(null = True)

```

Se hace las migraciones desde la consola

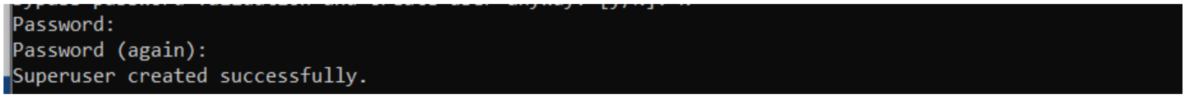


```
(PI_MLProject-GZ79I4Ry) C:\Users\57316\OneDrive - Universidad EAFIT\Documentos\Semestre 2022-2\Proyecto
Integrador 1\PI_MLProject\mlappproject>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, petclassiffier, sessions
Running migrations:
  Applying petclassiffier.0001_initial... OK
```

Ejecutando los comandos

- Python manage.py makemigrations
- Python manage.py migrate

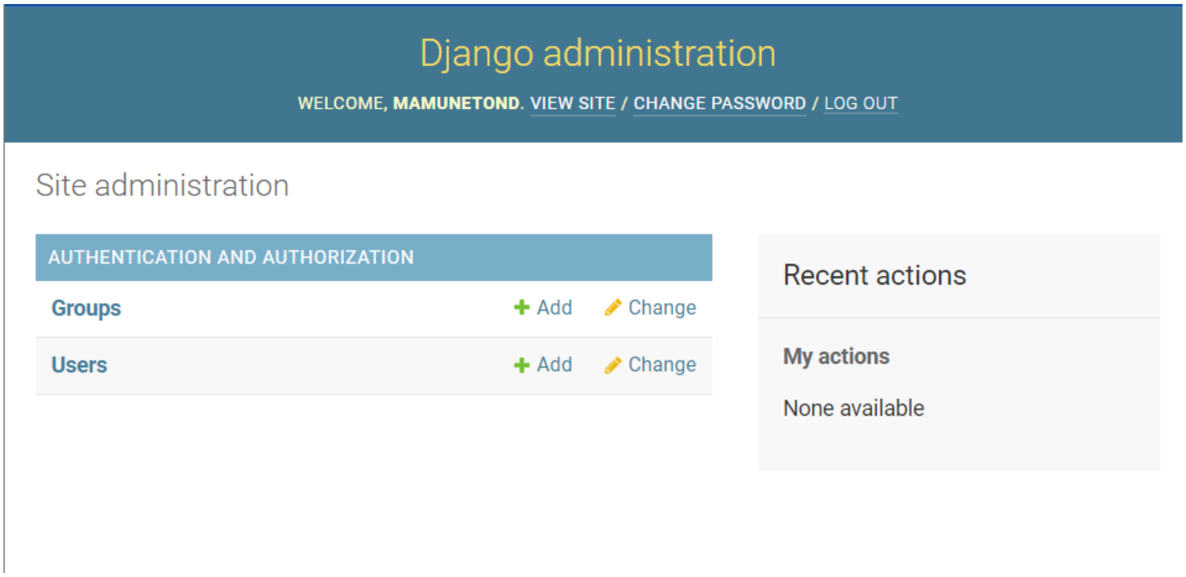
Luego en la terminal se ejecuta python manage.py createsuperuser



```
Password:
Password (again):
Superuser created successfully.
```

Se creó satisfactoriamente el superusuario.

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



Django administration

WELCOME, **MAMUNETOND**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

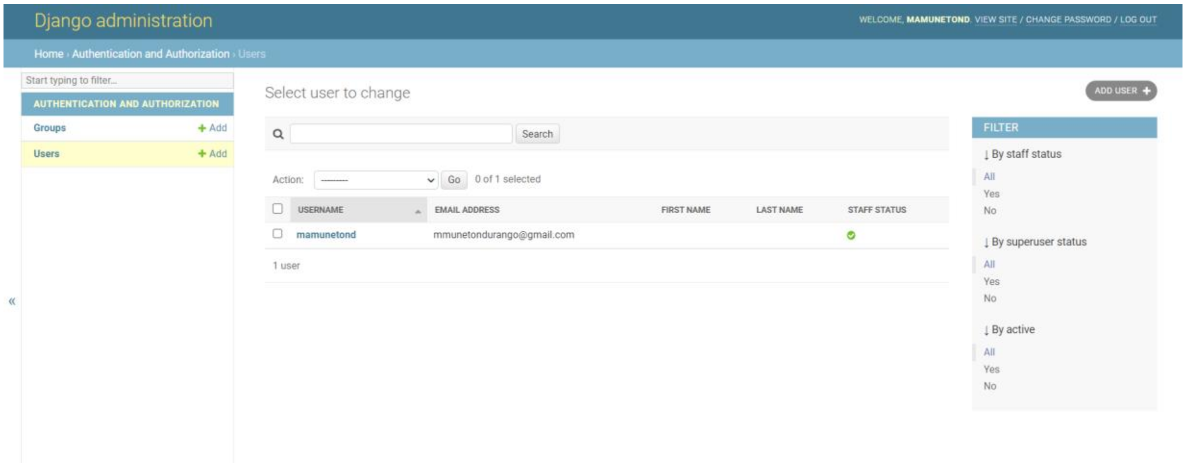
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

Recent actions

My actions

None available

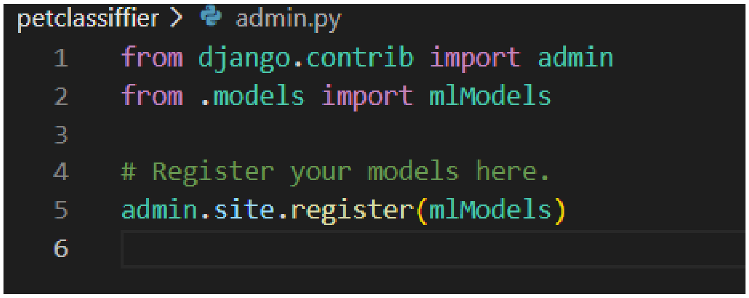
Vemos el panel de administrador de Django


Django administration
WELCOME, MAMUNETOND. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Authentication and Authorization › Users
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
«
Select user to change
ADD USER +
Search
Action: --------- Go 0 of 1 selected
FILTER
↓ By staff status
All
Yes
No
↓ By superuser status
All
Yes
No
↓ By active
All
Yes
No

□	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
□	mamunetond	mmunetondurango@gmail.com			✓

1 user

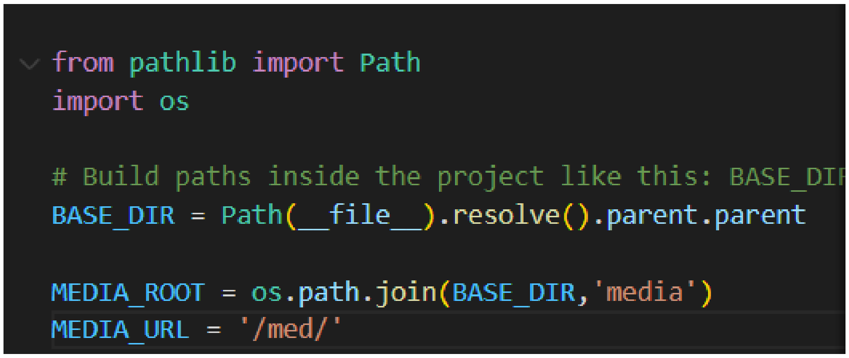
Está el usuario que creamos



petclassiffier > admin.py

```python
from django.contrib import admin
from .models import mlModels

# Register your models here.
admin.site.register(mlModels)

```

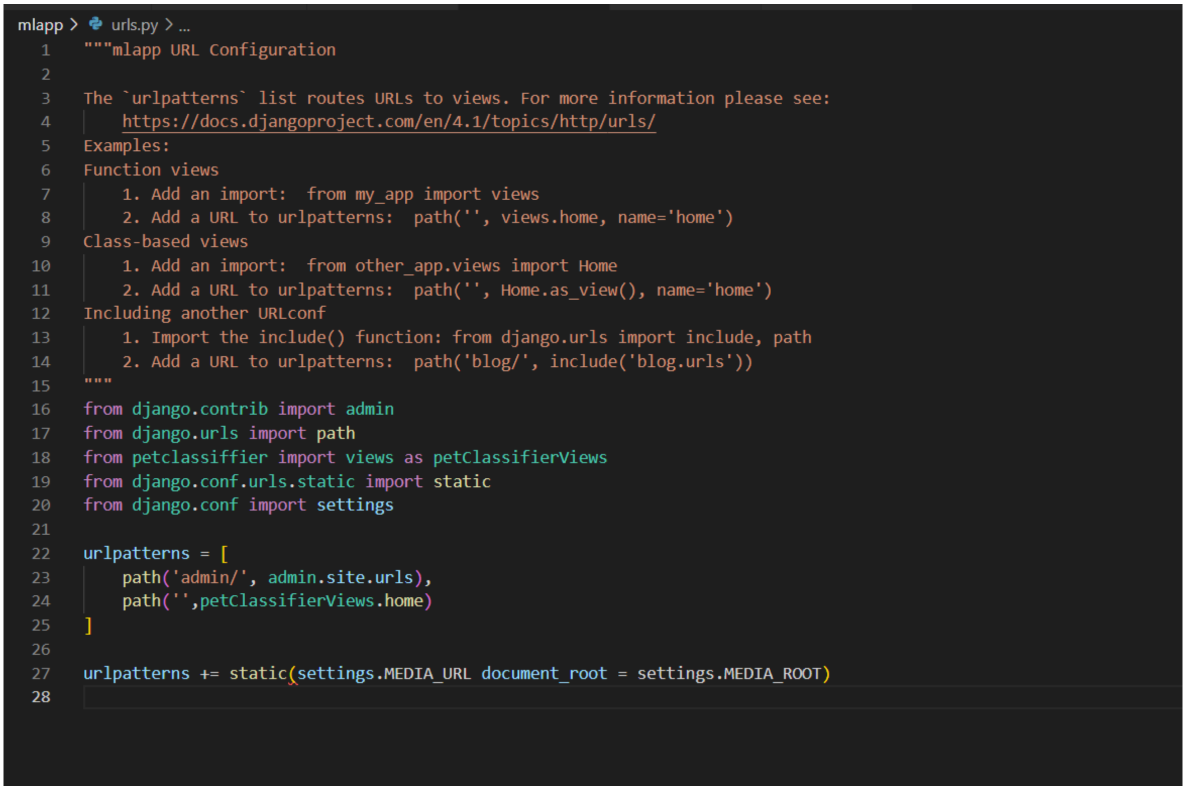
En admin.py agregamos el modelo.



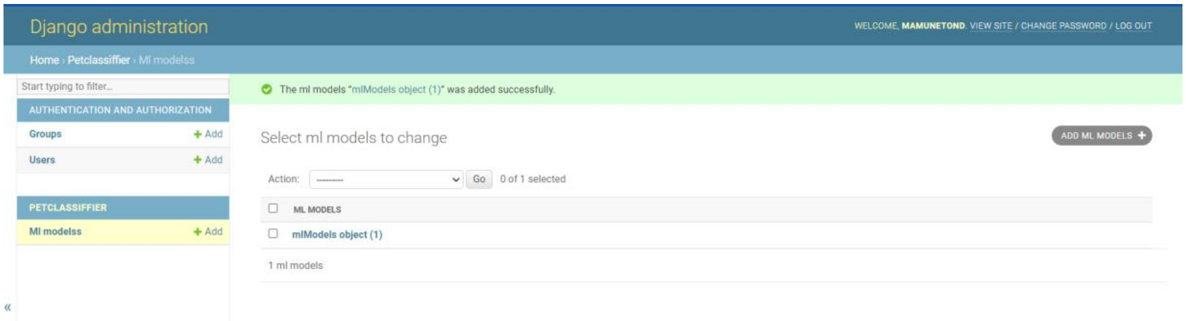
```python
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DI
BASE_DIR = Path(__file__).resolve().parent.parent

MEDIA_ROOT = os.path.join(BASE_DIR,'media')
MEDIA_URL = '/med/'
```

En settings.py se agrega MEDIA_ROOT y MEDIA_URL

Después se habilita el servidor para almacenar datos



```python
"""mlapp URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from petclassiffier import views as petClassifierViews
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',petClassifierViews.home)
]

urlpatterns += static(settings.MEDIA_URL document_root = settings.MEDIA_ROOT)

```


Django administration
WELCOME, MAMUNETOND. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Petclassiffier › Ml modelss
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
PETCLASSIFFIER
Ml modelss + Add
«

The ml models "mlModels object (1)" was added successfully.

Select ml models to change

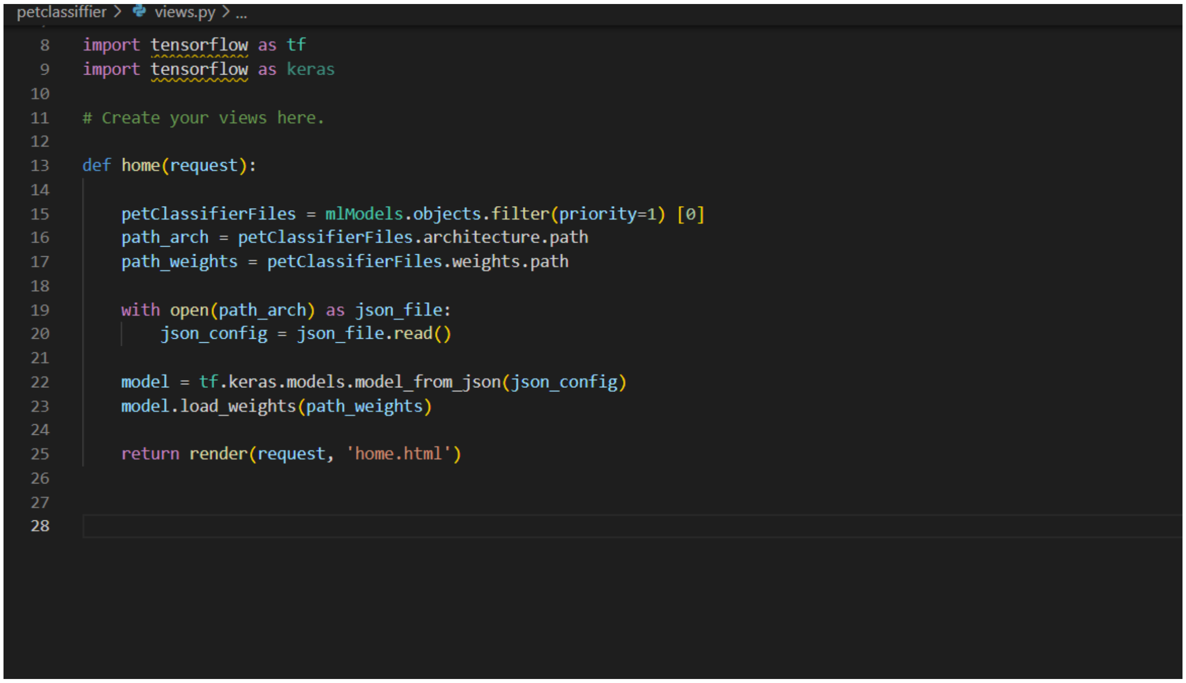
Action: --------- Go 0 of 1 selected

☐ ML MODELS

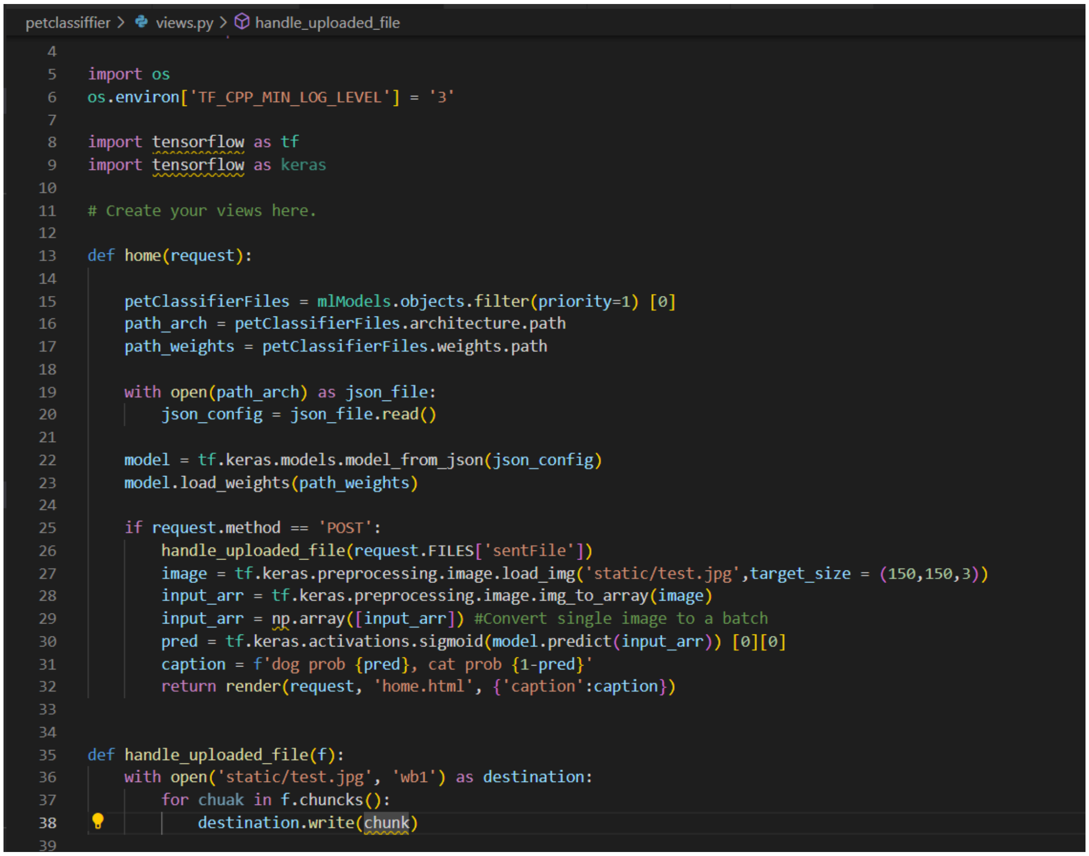
☐ mlModels object (1)

1 ml models

Se agregó de manera correcta.

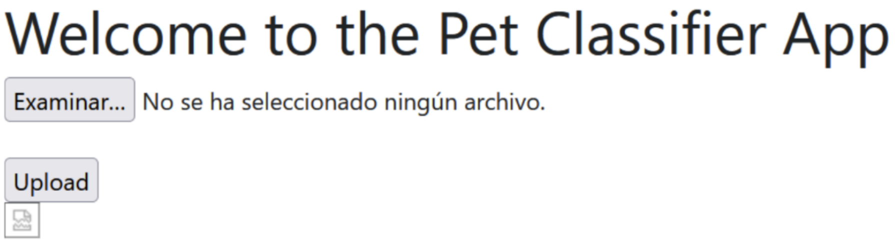


```python
import tensorflow as tf
import tensorflow as keras

# Create your views here.

def home(request):

    petClassifierFiles = mlModels.objects.filter(priority=1) [0]
    path_arch = petClassifierFiles.architecture.path
    path_weights = petClassifierFiles.weights.path

    with open(path_arch) as json_file:
        json_config = json_file.read()

    model = tf.keras.models.model_from_json(json_config)
    model.load_weights(path_weights)

    return render(request, 'home.html')


```



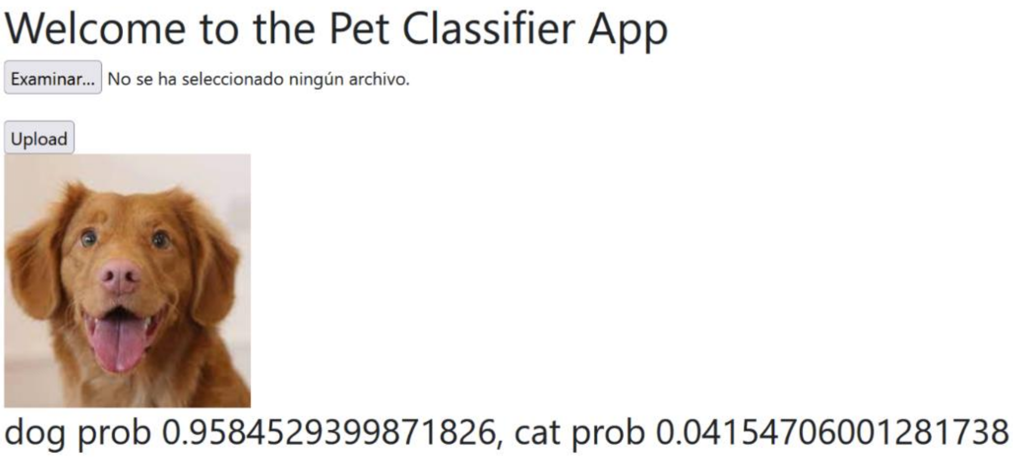
```python

import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

import tensorflow as tf
import tensorflow as keras

# Create your views here.

def home(request):

    petClassifierFiles = mlModels.objects.filter(priority=1) [0]
    path_arch = petClassifierFiles.architecture.path
    path_weights = petClassifierFiles.weights.path

    with open(path_arch) as json_file:
        json_config = json_file.read()

    model = tf.keras.models.model_from_json(json_config)
    model.load_weights(path_weights)

    if request.method == 'POST':
        handle_uploaded_file(request.FILES['sentFile'])
        image = tf.keras.preprocessing.image.load_img('static/test.jpg',target_size = (150,150,3))
        input_arr = tf.keras.preprocessing.image.img_to_array(image)
        input_arr = np.array([input_arr]) #Convert single image to a batch
        pred = tf.keras.activations.sigmoid(model.predict(input_arr)) [0][0]
        caption = f'dog prob {pred}, cat prob {1-pred}'
        return render(request, 'home.html', {'caption':caption})


def handle_uploaded_file(f):
    with open('static/test.jpg', 'wb1') as destination:
        for chuak in f.chuncks():
            destination.write(chunk)

```

Finalmente



Welcome to the Pet Classifier App

No se ha seleccionado ningún archivo.



Welcome to the Pet Classifier App

No se ha seleccionado ningún archivo.

```
dog prob 0.9584529399871826, cat prob 0.04154706001281738
```

3