Capturas del archivo train.ipynb

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

    Mounted at /content/drive

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

    /content/drive/MyDrive/PI_MLProject

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

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

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

    2.8.0
    2.8.2

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

    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
```

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

```
    (499, 327, 3)
```

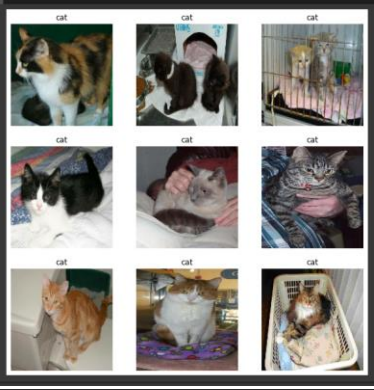
```
    training_path = os.path.join(data_path,'train')
    training_set = image_dataset_from_directory(training_path,
    shuffle=True,
    batch_size=32,
    image_size=(150, 150),
    validation_split = 0.2,
    subset = 'training',
    seed = 1234,
    )
    validation_set = image_dataset_from_directory(training_path,
    shuffle=True,
    batch_size=32,
    image_size=(150, 150),
    validation_split = 0.2,
    subset = 'validation',
    seed = 1234,
    )

    Found 1066 files belonging to 2 classes.
    Using 853 files for training.
    Found 1066 files belonging to 2 classes.
    Using 213 files for validation.

[ ] training_set.class_names

    ['cat', 'dog']
```

```
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 [==============================] - 0s 0us/step
83697664/83683744 [==============================] - 0s 0us/step
```

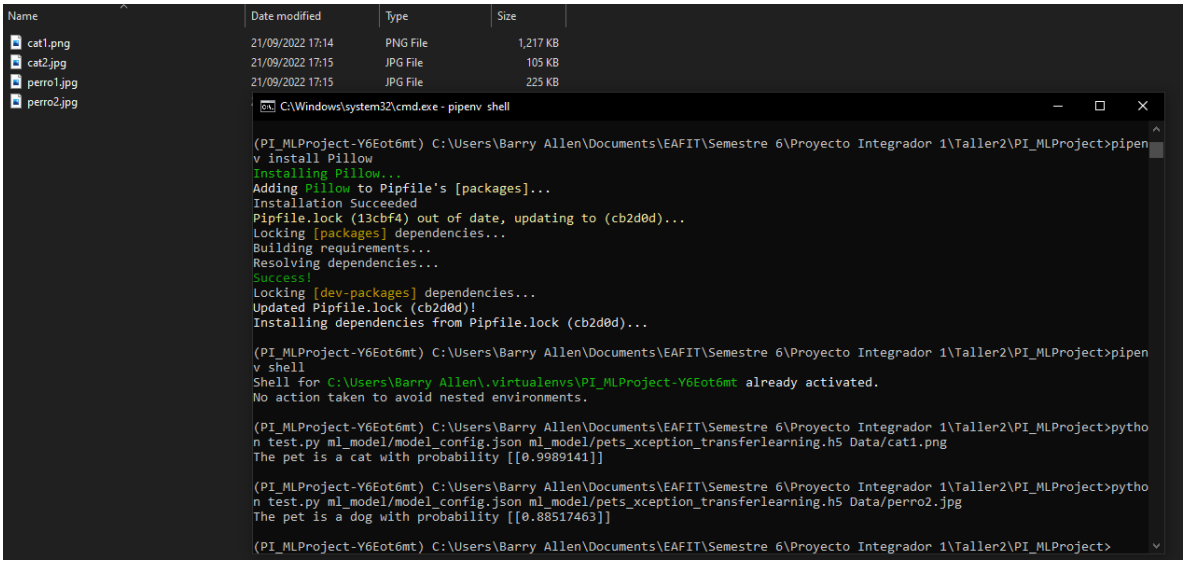
```
inputs = keras.Input(shape = (150,150,3))
x = tf.keras.applications.xception.preprocess_input(inputs)
x = base_model(x, training=False)
x = keras.layers.GlobalAveragePooling2D()(x)
x = keras.layers.Dropout(0.2)(x)
outputs = keras.layers.Dense(1)(x)
model = keras.Model(inputs,outputs)
```

```
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 1/20
27/27 [==============================] - 32s 697ms/step - loss: 0.2738 - binary_accuracy: 0.9215 - val_loss: 0.2586 - val_binary_accuracy: 0.9155
Epoch 2/20
27/27 [==============================] - 3s 109ms/step - loss: 0.1005 - binary_accuracy: 0.9601 - val_loss: 0.0894 - val_binary_accuracy: 0.9577
Epoch 3/20
27/27 [==============================] - 3s 110ms/step - loss: 0.0631 - binary_accuracy: 0.9695 - val_loss: 0.0804 - val_binary_accuracy: 0.9577
Epoch 4/20
27/27 [==============================] - 3s 110ms/step - loss: 0.0439 - binary_accuracy: 0.9836 - val_loss: 0.0654 - val_binary_accuracy: 0.9624
Epoch 5/20
27/27 [==============================] - 3s 110ms/step - loss: 0.0350 - binary_accuracy: 0.9894 - val_loss: 0.0604 - val_binary_accuracy: 0.9624
Epoch 6/20
27/27 [==============================] - 4s 111ms/step - loss: 0.0291 - binary_accuracy: 0.9894 - val_loss: 0.0576 - val_binary_accuracy: 0.9624
Epoch 7/20
27/27 [==============================] - 3s 109ms/step - loss: 0.0238 - binary_accuracy: 0.9906 - val_loss: 0.0497 - val_binary_accuracy: 0.9624
Epoch 8/20
27/27 [==============================] - 3s 110ms/step - loss: 0.0201 - binary_accuracy: 0.9941 - val_loss: 0.0571 - val_binary_accuracy: 0.9624
Epoch 9/20
27/27 [==============================] - 4s 112ms/step - loss: 0.0188 - binary_accuracy: 0.9941 - val_loss: 0.0489 - val_binary_accuracy: 0.9624
Epoch 10/20
27/27 [==============================] - 3s 111ms/step - loss: 0.0140 - binary_accuracy: 1.0000 - val_loss: 0.0500 - val_binary_accuracy: 0.9624
Epoch 11/20
27/27 [==============================] - 3s 109ms/step - loss: 0.0143 - binary_accuracy: 0.9953 - val_loss: 0.0503 - val_binary_accuracy: 0.9624
Epoch 12/20
27/27 [==============================] - 3s 110ms/step - loss: 0.0115 - binary_accuracy: 1.0000 - val_loss: 0.0487 - val_binary_accuracy: 0.9624
Epoch 13/20
27/27 [==============================] - 3s 111ms/step - loss: 0.0096 - binary_accuracy: 0.9988 - val_loss: 0.0539 - val_binary_accuracy: 0.9624
Epoch 14/20
27/27 [==============================] - 3s 109ms/step - loss: 0.0104 - binary_accuracy: 1.0000 - val_loss: 0.0485 - val_binary_accuracy: 0.9671
Epoch 15/20
27/27 [==============================] - 3s 109ms/step - loss: 0.0089 - binary_accuracy: 1.0000 - val_loss: 0.0450 - val_binary_accuracy: 0.9671
Epoch 16/20
27/27 [==============================] - 4s 113ms/step - loss: 0.0079 - binary_accuracy: 1.0000 - val_loss: 0.0487 - val_binary_accuracy: 0.9671
Epoch 17/20
27/27 [==============================] - 3s 110ms/step - loss: 0.0068 - binary_accuracy: 1.0000 - val_loss: 0.0544 - val_binary_accuracy: 0.9624
Epoch 18/20
27/27 [==============================] - 3s 109ms/step - loss: 0.0073 - binary_accuracy: 1.0000 - val_loss: 0.0473 - val_binary_accuracy: 0.9671
Epoch 19/20
27/27 [==============================] - 3s 111ms/step - loss: 0.0059 - binary_accuracy: 1.0000 - val_loss: 0.0480 - val_binary_accuracy: 0.9671
Epoch 20/20
27/27 [==============================] - 4s 115ms/step - loss: 0.0058 - binary_accuracy: 1.0000 - val_loss: 0.0475 - val_binary_accuracy: 0.9671
<keras.callbacks.History at 0x7f7a700d8cd0>
```

```
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')
```

Muestras de la ejecución del código con fotos bajadas de internet



Nota: Como le comenté al profe, tengo un error en el código, no entiendo qué me quiere decir el debugger de Django, entonces el profe me dijo que adjuntara todo lo que tenía hasta el momento y que luego sería revisado para orientarme en cuál es el error que se me presentó. El código funciona hasta la parte 4 al final del todo, sólo le faltaba la parte de la carga de fotos desde la aplicación web, pero desde la vista de admin, si deja.

El error