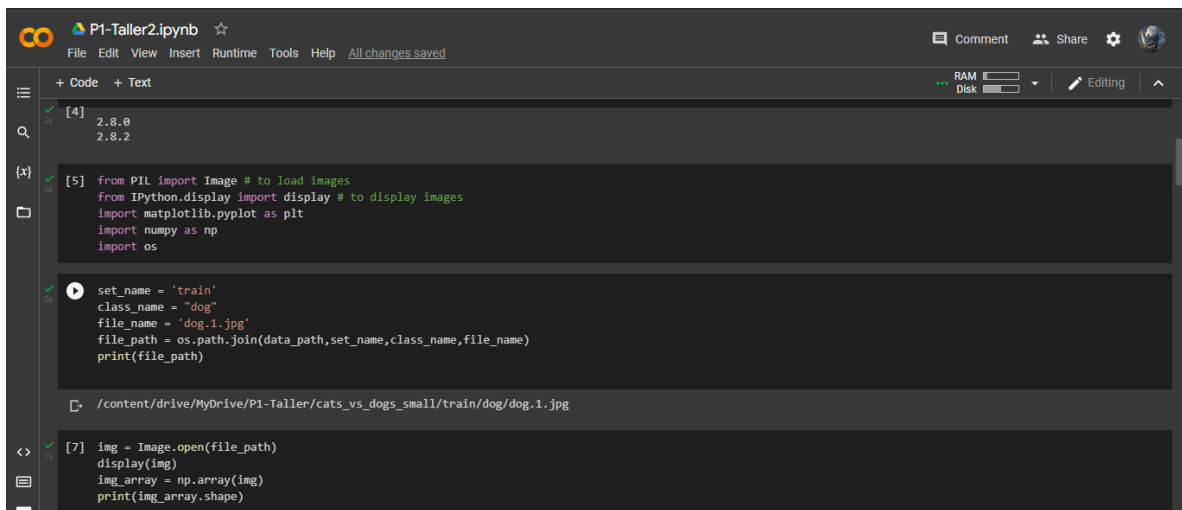


Presentado por:

- Miguel Angel Martinez Florez

## Taller 2

- pantallazos del archivo train.ipynb donde se define el nuevo modelo base.



The screenshot shows a Jupyter Notebook titled "P1-Taller2.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with options for Code, Text, Comment, Share, and settings. The notebook is in "Editing" mode. The code is organized into cells, with the first three cells being visible. The first cell (index 4) contains version information for Python (2.8.0 and 2.8.2). The second cell (index 5) contains import statements for PIL, IPython, matplotlib, and numpy. The third cell (index 6) contains code to define file paths and print them. The output of the third cell is a file path: "/content/drive/MyDrive/P1-Taller/cats\_vs\_dogs\_small/train/dog/dog.1.jpg". The fourth cell (index 7) contains code to load the image, display it, and print its shape. The output of the fourth cell is a visual representation of the image, which is a brown dog standing in a cage.

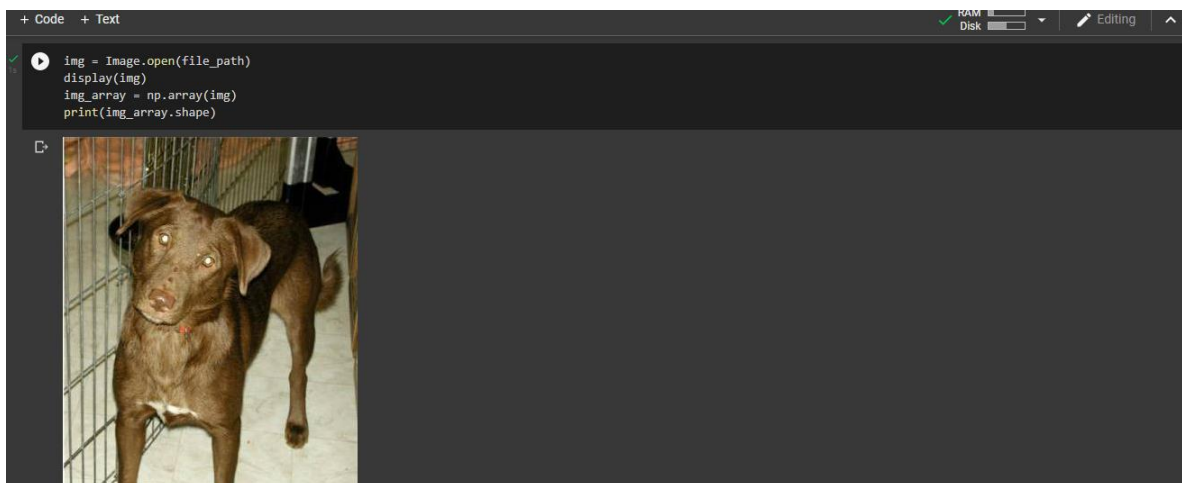
```
[4] 2.8.0
    2.8.2

[5] from PIL import Image # to load images
    from IPython.display import display # to display images
    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/P1-Taller/cats_vs_dogs_small/train/dog/dog.1.jpg

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



```

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,
)

```

```

[9] training_set.class_names

['cat', 'dog']

```

```

[10] 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")

```



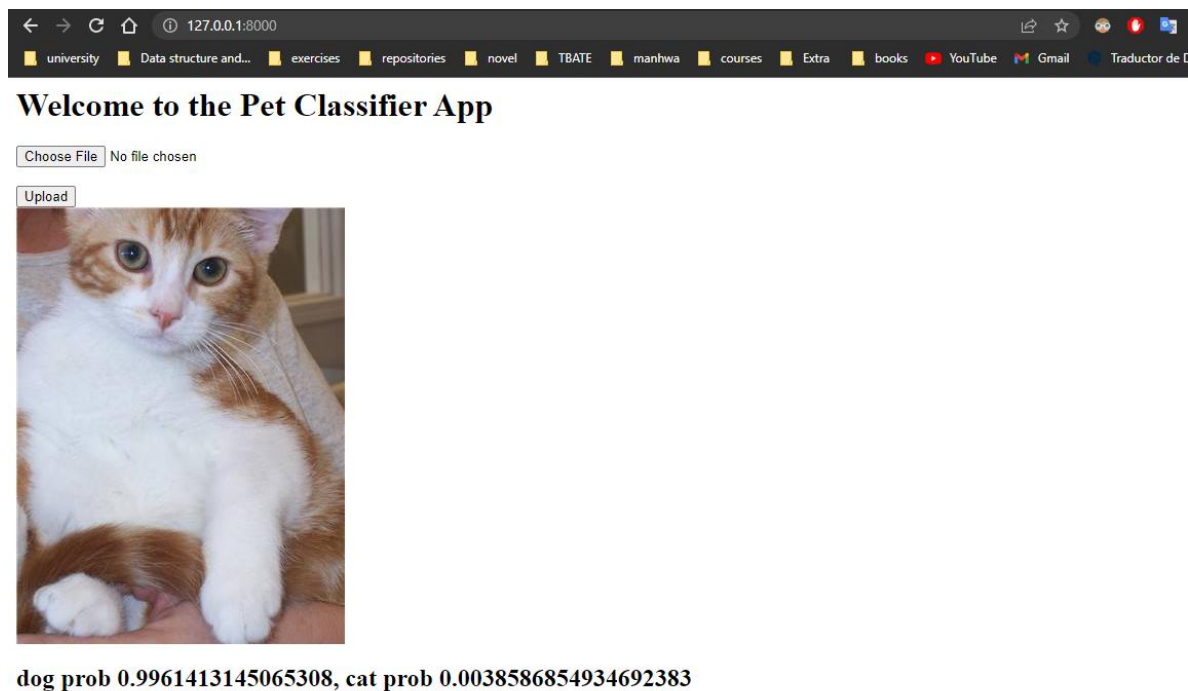
```
[11] base_model = keras.applications.Xception(
      weights = 'imagenet',
      input_shape = (150,150,3),
      include_top = False,
      )
      base_model.trainable = False

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

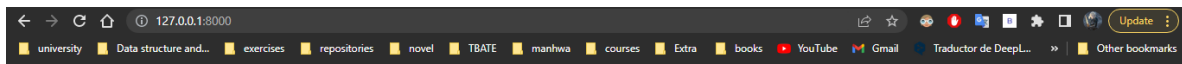
[12] inputs = keras.Input(shape = (150,150,3))
      x = tf.keras.applications.xception.preprocess_input(inputs)
      x = base_model(x, training=False)
      x = keras.layers.GlobalAveragePooling2D()(x)
      x = keras.layers.Dropout(0.2)(x)
      outputs = keras.layers.Dense(1)(x)
      model = keras.Model(inputs,outputs)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

- Pantallazos de la app en django donde se pruebe el modelo con alguna foto entregada por usted

Dada por la carpeta data

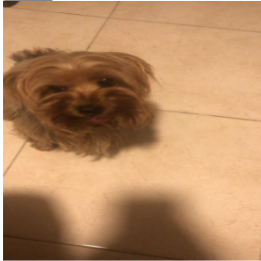


- imágenes dada por mi:

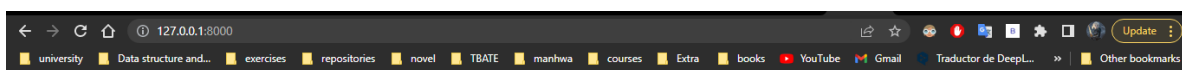


## Welcome to the Pet Classifier App

No file chosen

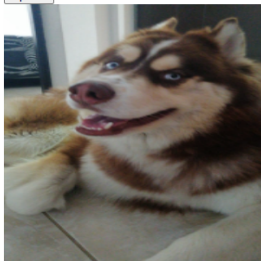


dog prob 0.8931819200515747, cat prob 0.10681807994842529



## Welcome to the Pet Classifier App

No file chosen



dog prob 0.21165083348751068, cat prob 0.7883491516113281

- Código de la implementación:

