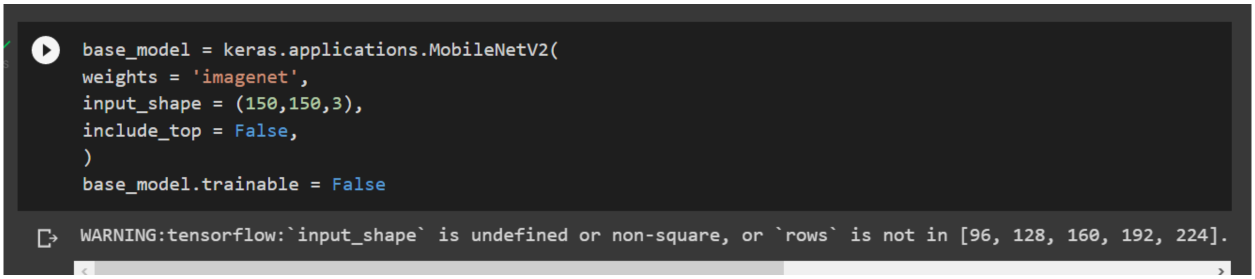
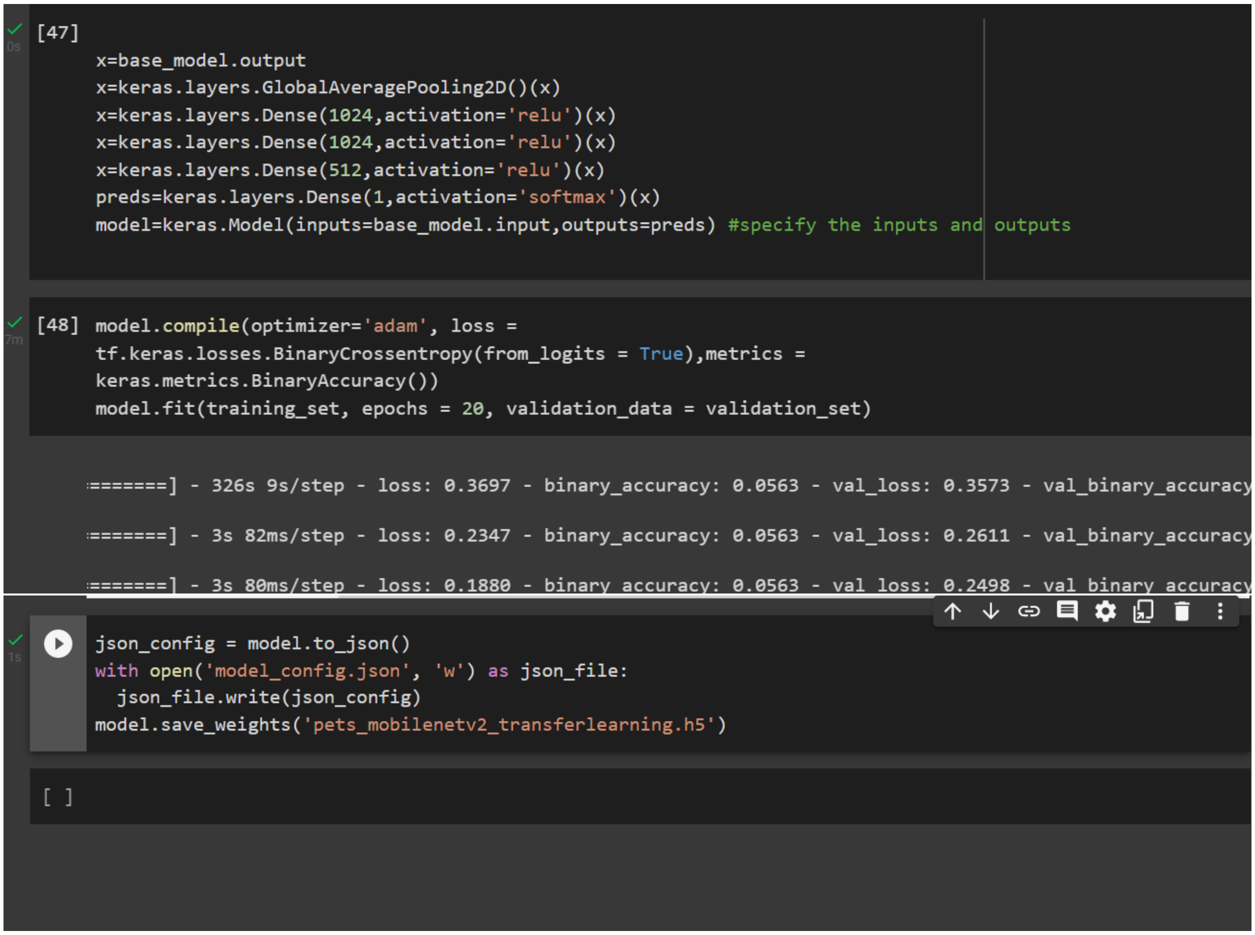
El modelo utilizado ha sido MobileNetV2.



```
base_model = keras.applications.MobileNetV2(
weights = 'imagenet',
input_shape = (150,150,3),
include_top = False,
)
base_model.trainable = False
```

```
WARNING:tensorflow:`input_shape` is undefined or non-square, or `rows` is not in [96, 128, 160, 192, 224].
```



```
[47]
 x=base_model.output
 x=keras.layers.GlobalAveragePooling2D()(x)
 x=keras.layers.Dense(1024,activation='relu')(x)
 x=keras.layers.Dense(1024,activation='relu')(x)
 x=keras.layers.Dense(512,activation='relu')(x)
 preds=keras.layers.Dense(1,activation='softmax')(x)
 model=keras.Model(inputs=base_model.input,outputs=preds) #specify the inputs and outputs
```

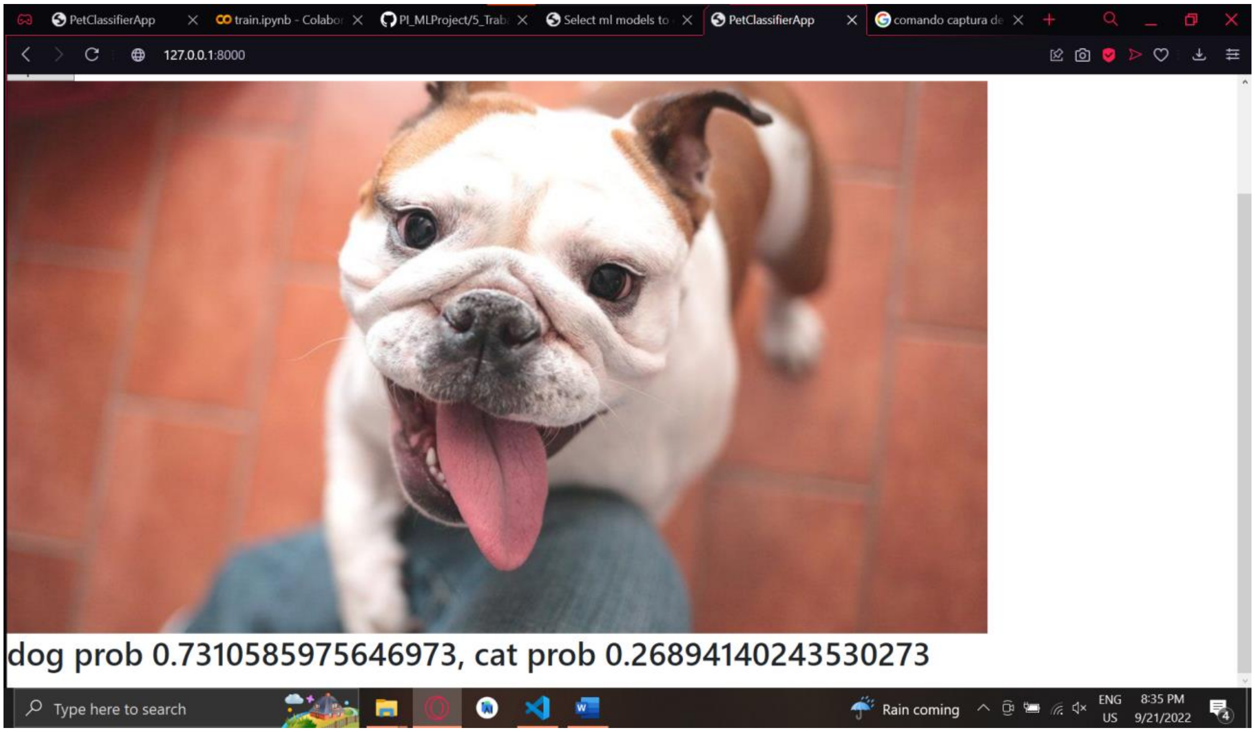
```
[48] model.compile(optimizer='adam', loss =
     tf.keras.losses.BinaryCrossentropy(from_logits = True),metrics =
     keras.metrics.BinaryAccuracy())
     model.fit(training_set, epochs = 20, validation_data = validation_set)
```

```
=======] - 326s 9s/step - loss: 0.3697 - binary_accuracy: 0.0563 - val_loss: 0.3573 - val_binary_accuracy

=======] - 3s 82ms/step - loss: 0.2347 - binary_accuracy: 0.0563 - val_loss: 0.2611 - val_binary_accuracy

=======] - 3s 80ms/step - loss: 0.1880 - binary_accuracy: 0.0563 - val_loss: 0.2498 - val_binary_accuracy
```

```
json_config = model.to_json()
with open('model_config.json', 'w') as json_file:
  json_file.write(json_config)
model.save_weights('pets_mobilenetv2_transferlearning.h5')
```

```
[ ]
```

Prueba del modelo en DJANGO



```
dog prob 0.7310585975646973, cat prob 0.26894140243530273
```