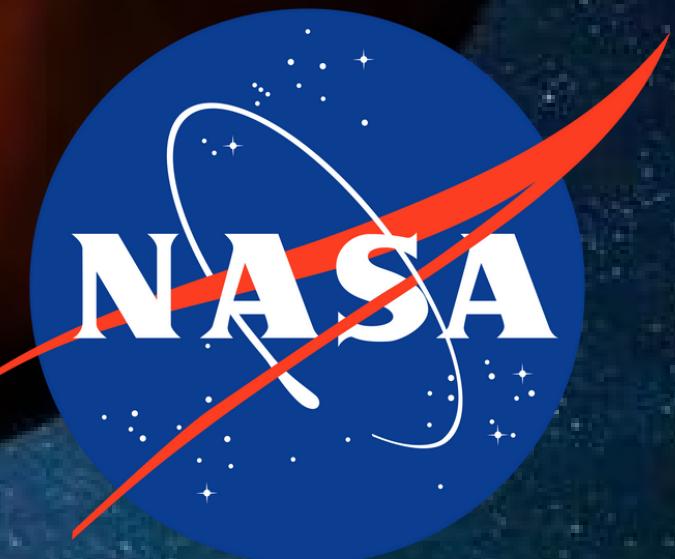
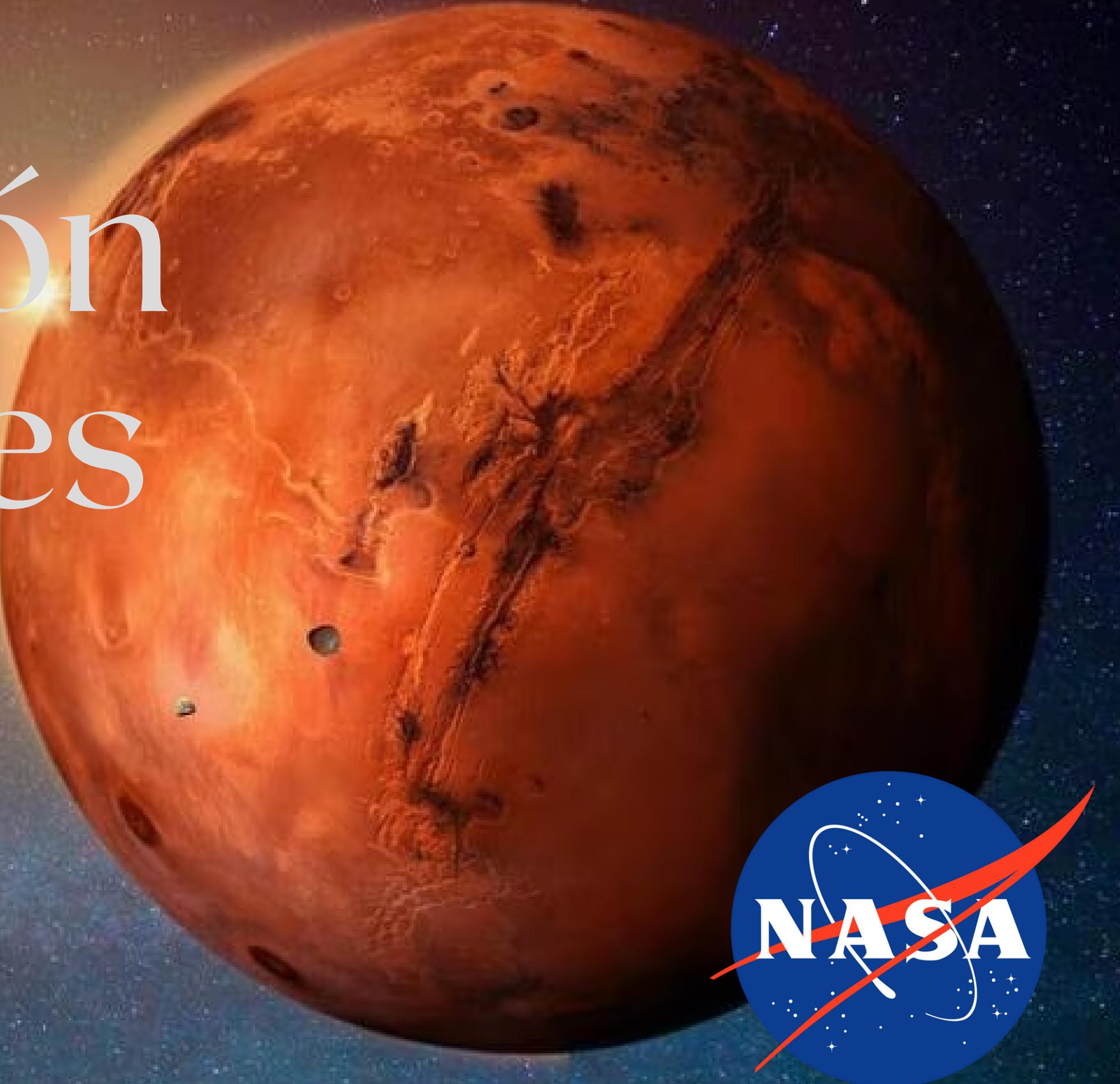


# Clasificación de Imágenes

JHON RAMIREZ  
SEBASTIAN GIL  
NICOLÁS RODRÍGUEZ





# El Problema

---

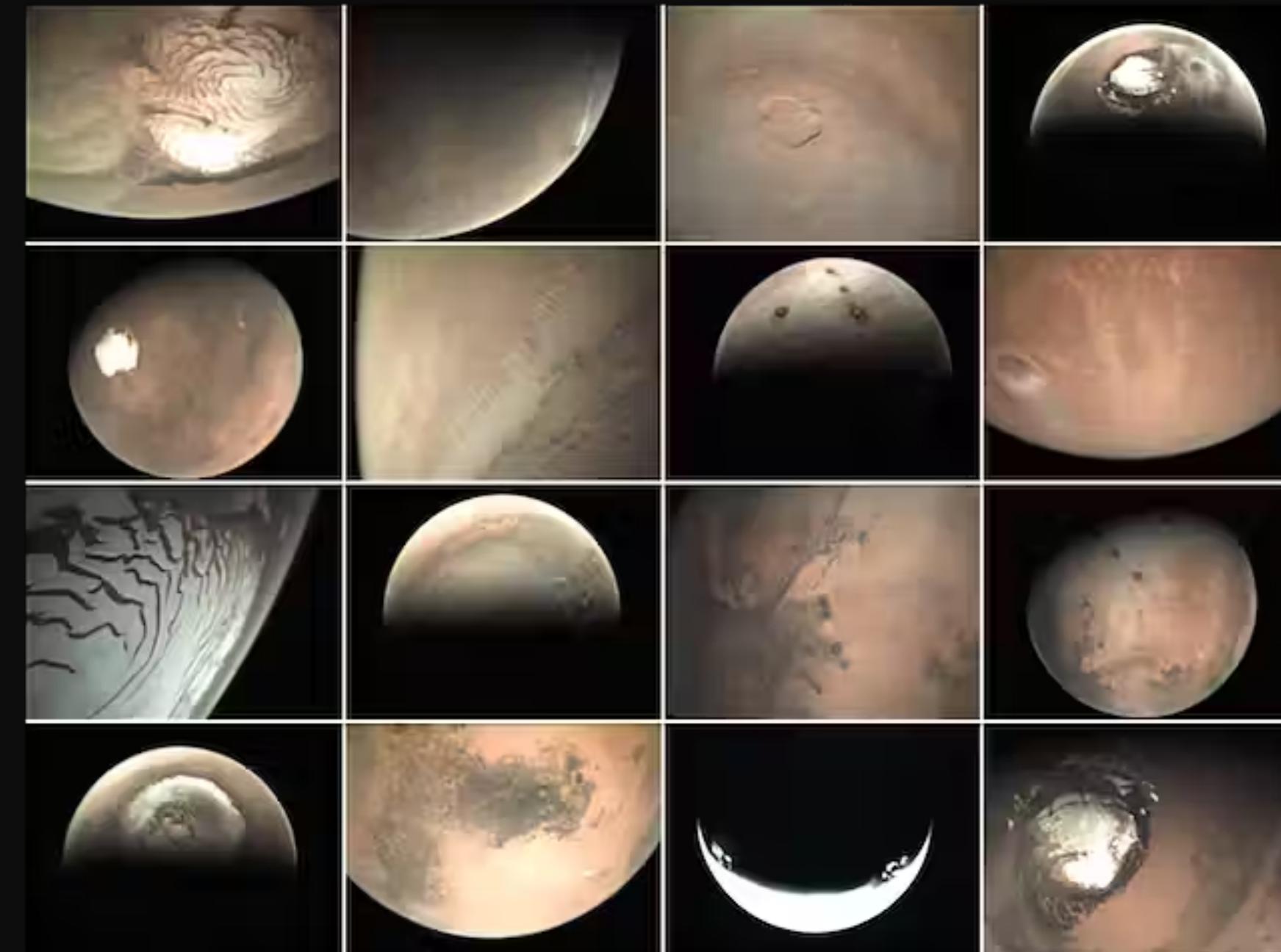
## PROBLEMAS DE CLASIFICACIÓN

La clasificación de imágenes astronómicas de Marte presenta varios desafíos debido a las características únicas del planeta y las condiciones en las que se capturan las imágenes

- Variabilidad en las condiciones de iluminación.
- Ruido y distorsión.
- Variabilidad en la topografía.
- Escala y resolución.
- Presencia de polvo.
- Limitaciones en la disponibilidad de datos etiquetados.

# Los Datos

---

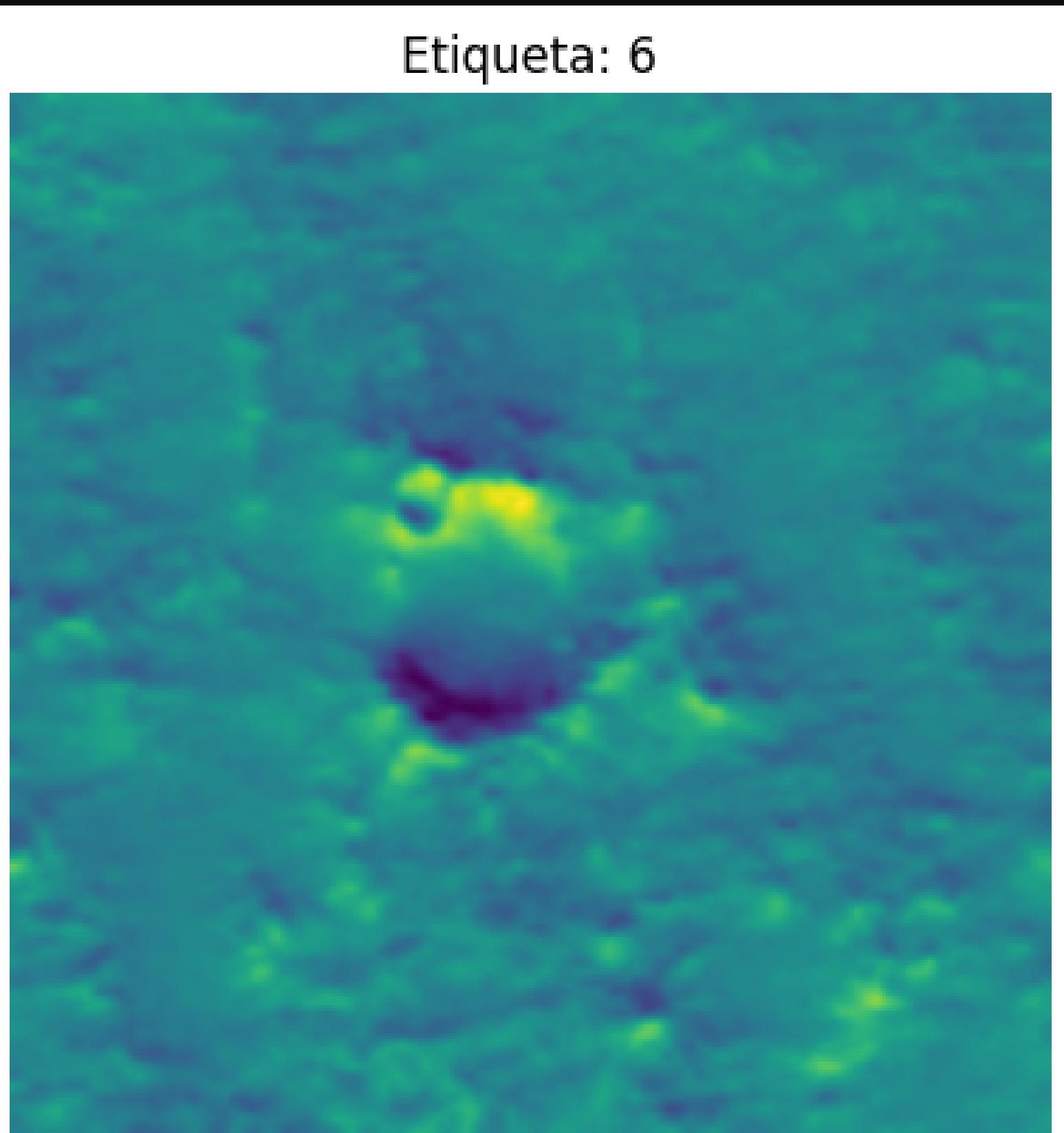
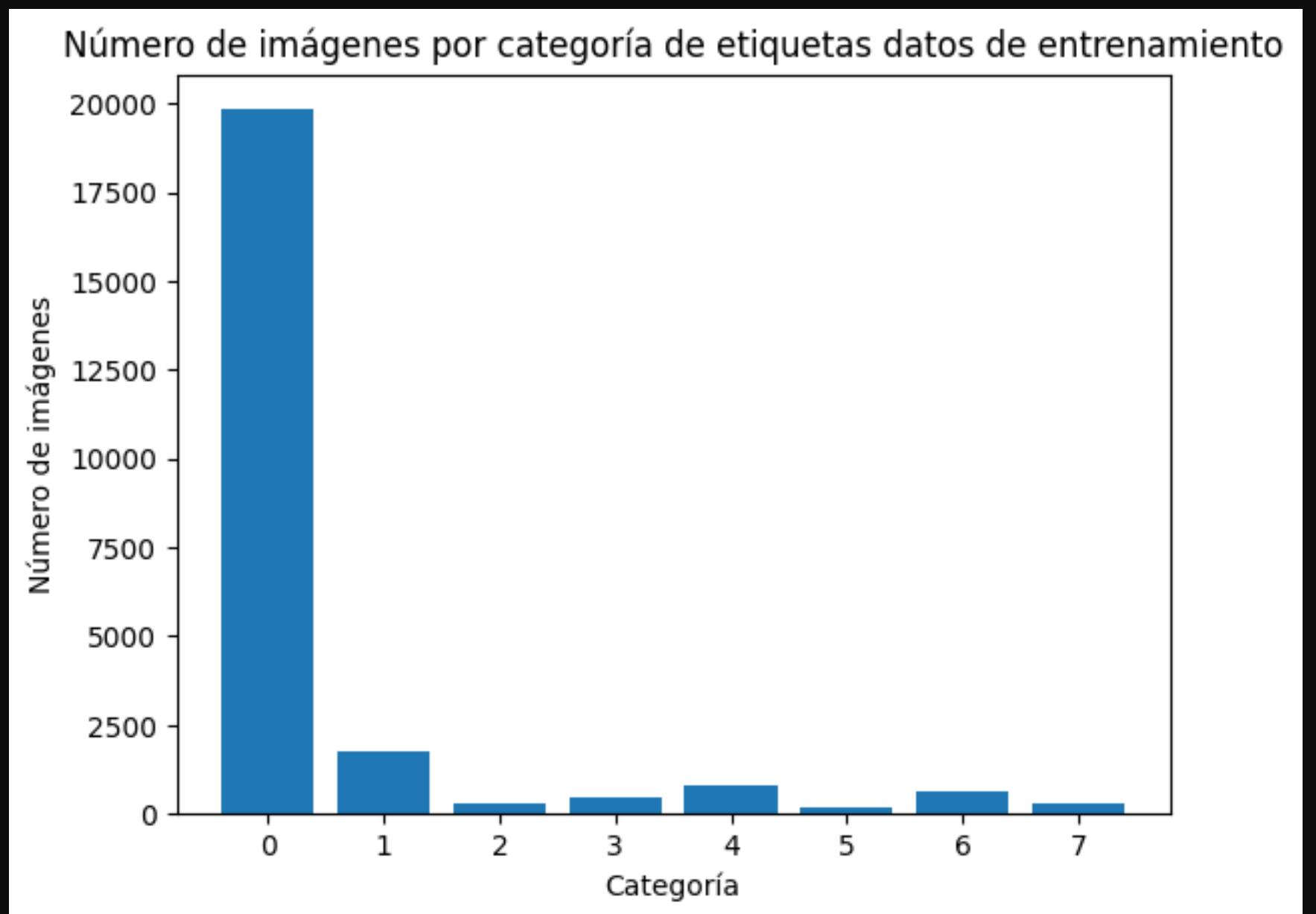


Este conjunto se formó a partir de 10.815 hitos originales. Cada punto de referencia original se recortó a un cuadro delimitador cuadrado que incluía la extensión total del punto de referencia más un margen de 30 píxeles a la izquierda, derecha, arriba y abajo. A continuación, se redimensionó cada punto de referencia a 227x227 píxeles. Luego, se aumentaron 9022 de estas imágenes para generar 6 puntos de referencia adicionales utilizando los siguientes métodos:

1. Rotación de 90 grados en el sentido de las agujas del reloj
2. Rotación de 180 grados en el sentido de las agujas del reloj
3. Rotación de 270 grados en el sentido de las agujas del reloj
4. Giro horizontal
5. Giro vertical
6. Ajuste de brillo aleatorio

Las 1.793 imágenes restantes no fueron aumentadas. Combinando estos con las 7\*9,022 imágenes, da un total de 64,947 imágenes separadas.

# Preprocesamiento de los datos



# El Modelo

*En el presente estudio, se realizará la clasificación de imágenes utilizando redes neuronales convolucionales . Se explorarán dos enfoques distintos: la creación de una red desde cero y el reentrenamiento de una red preentrenada mediante transfer learning. Estos métodos se aplicarán a un conjunto de datos compuesto por imágenes y etiquetas.*

*Adicionalmente se usará un modelo RESNET para prevenir el problema de la degradación del rendimiento que se produce cuando se agregan capas adicionales a una red neuronal.*

*La creación de una red desde cero nos brinda la flexibilidad de diseñar la arquitectura a medida, mientras que el reentrenamiento de una red preentrenada permite aprovechar el conocimiento previo adquirido en conjuntos de datos masivos.*



# Primer Modelo

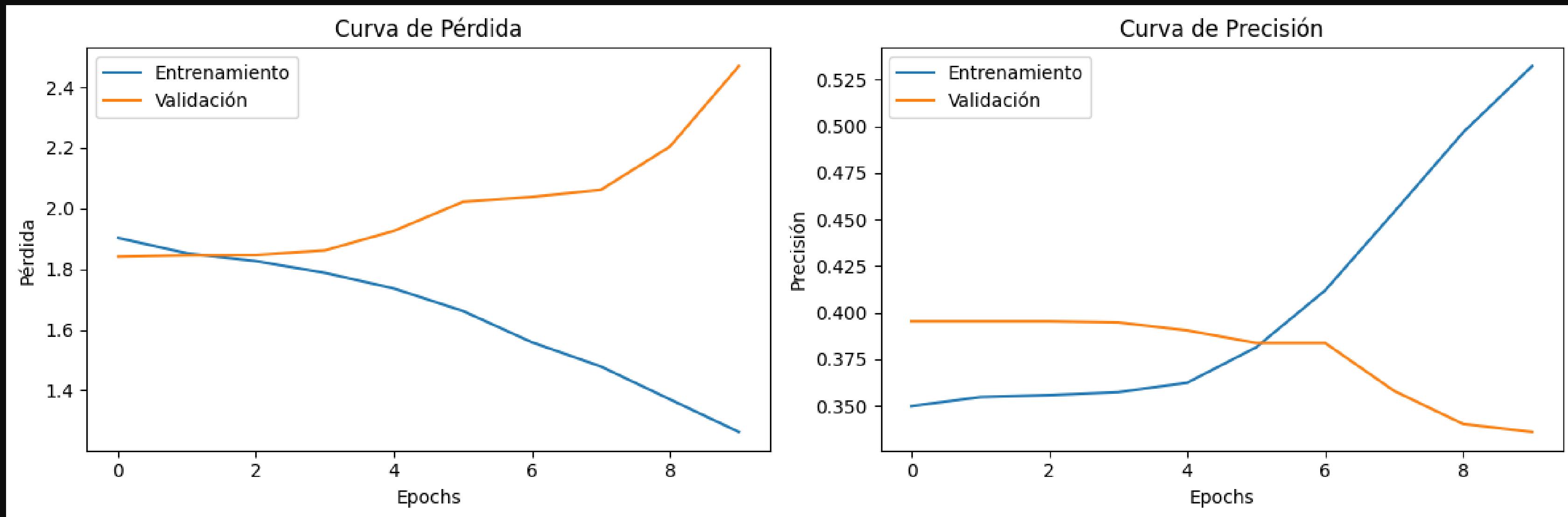
## RED NEURONAL CONVOLUCIONAL

```
Model: "sequential_4"
```

Layer (type)	Output Shape	Param #
<hr/>		
conv2d_8 (Conv2D)	(None, 225, 225, 16)	160
max_pooling2d_8 (MaxPooling 2D)	(None, 112, 112, 16)	0
conv2d_9 (Conv2D)	(None, 110, 110, 32)	4640
max_pooling2d_9 (MaxPooling 2D)	(None, 55, 55, 32)	0
flatten_4 (Flatten)	(None, 96800)	0
dense_8 (Dense)	(None, 32)	3097632
dense_9 (Dense)	(None, 8)	264
<hr/>		
Total params: 3,102,696		
Trainable params: 3,102,696		
Non-trainable params: 0		

# Primer Modelo

## RED NEURONAL CONVOLUCIONAL



# Segundo Modelo

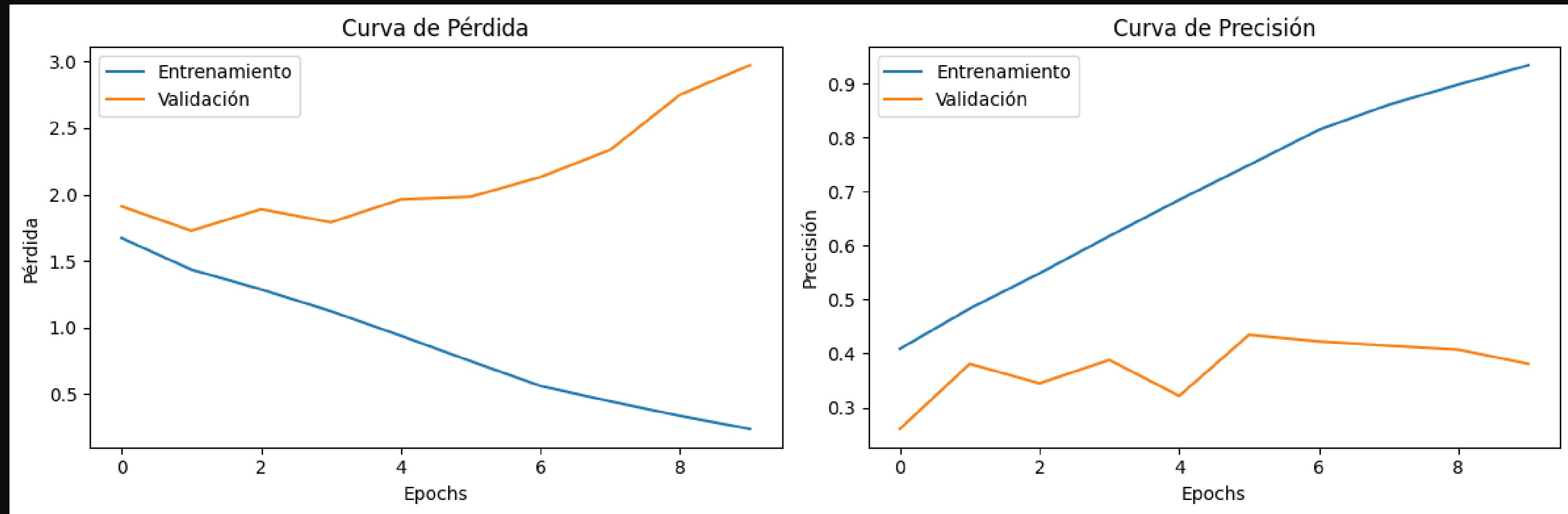
## RED NEURONAL CONVOLUCIONAL

```
Model: "sequential_6"

Layer (type)                 Output Shape              Param #
=====
conv2d_49 (Conv2D)           (None, 225, 225, 8)      80
max_pooling2d_27 (MaxPooling2D) (None, 112, 112, 8)    0
conv2d_50 (Conv2D)           (None, 110, 110, 16)     1168
max_pooling2d_28 (MaxPooling2D) (None, 55, 55, 16)      0
flatten_11 (Flatten)         (None, 48400)            0
dense_21 (Dense)             (None, 16)                774416
dense_22 (Dense)             (None, 8)                 136
=====
Total params: 775,800
Trainable params: 775,800
Non-trainable params: 0
```

# Segundo Modelo

## RED NEURONAL CONVOLUCIONAL



# Tercer Modelo

## RED NEURONAL RESNET

```
activation_20 (Activation)      (None, 28, 28, 128)  0          ['batch_normalization_20[0][0]']

conv2d_47 (Conv2D)             (None, 28, 28, 128)  147584    ['activation_20[0][0]']

conv2d_48 (Conv2D)             (None, 28, 28, 128)  8320     ['max_pooling2d_25[0][0]']

batch_normalization_21 (BatchN (None, 28, 28, 128)  512      ['conv2d_47[0][0]']

ormalization)

add_10 (Add)                  (None, 28, 28, 128)  0          ['conv2d_48[0][0]',
                                             'batch_normalization_21[0][0]']

activation_21 (Activation)     (None, 28, 28, 128)  0          ['add_10[0][0]']

max_pooling2d_26 (MaxPooling2D (None, 14, 14, 128)  0        ['activation_21[0][0]']

)

flatten_10 (Flatten)           (None, 25088)         0          ['max_pooling2d_26[0][0]']

dense_19 (Dense)              (None, 128)          3211392    ['flatten_10[0][0]']

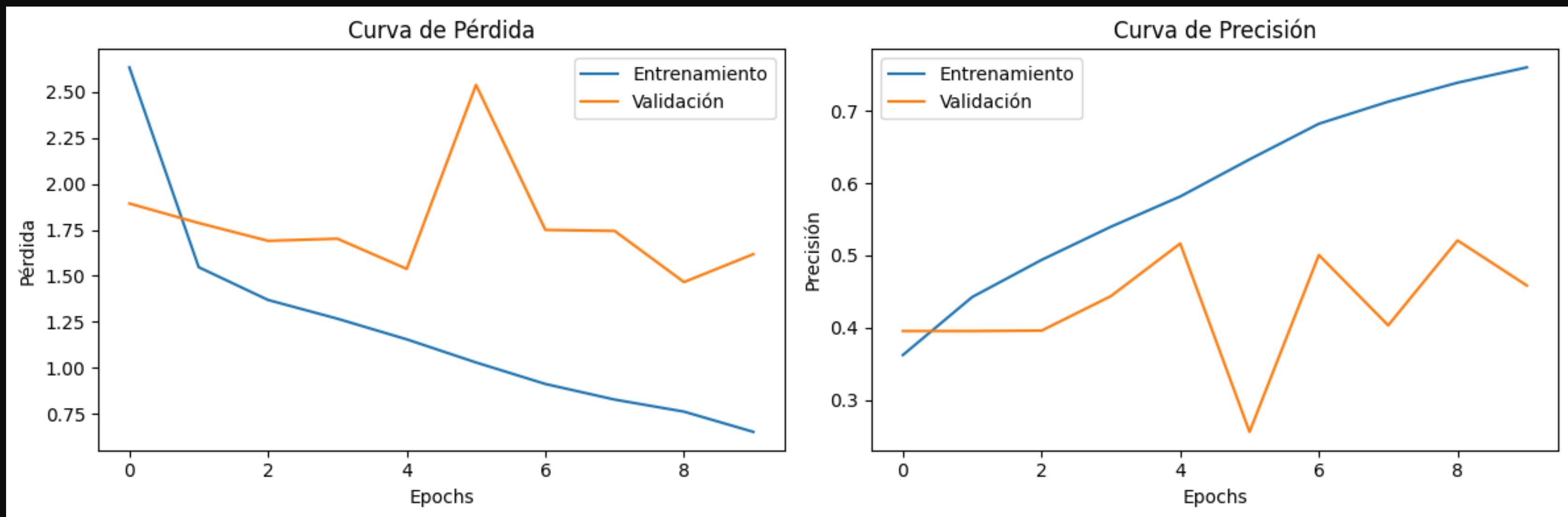
dropout_1 (Dropout)            (None, 128)          0          ['dense_19[0][0]']

dense_20 (Dense)              (None, 8)            1032       ['dropout_1[0][0]']

=====
Total params: 3,521,384
Trainable params: 3,520,488
Non-trainable params: 896
```

# Tercer Modelo

## RED NEURONAL RESNET



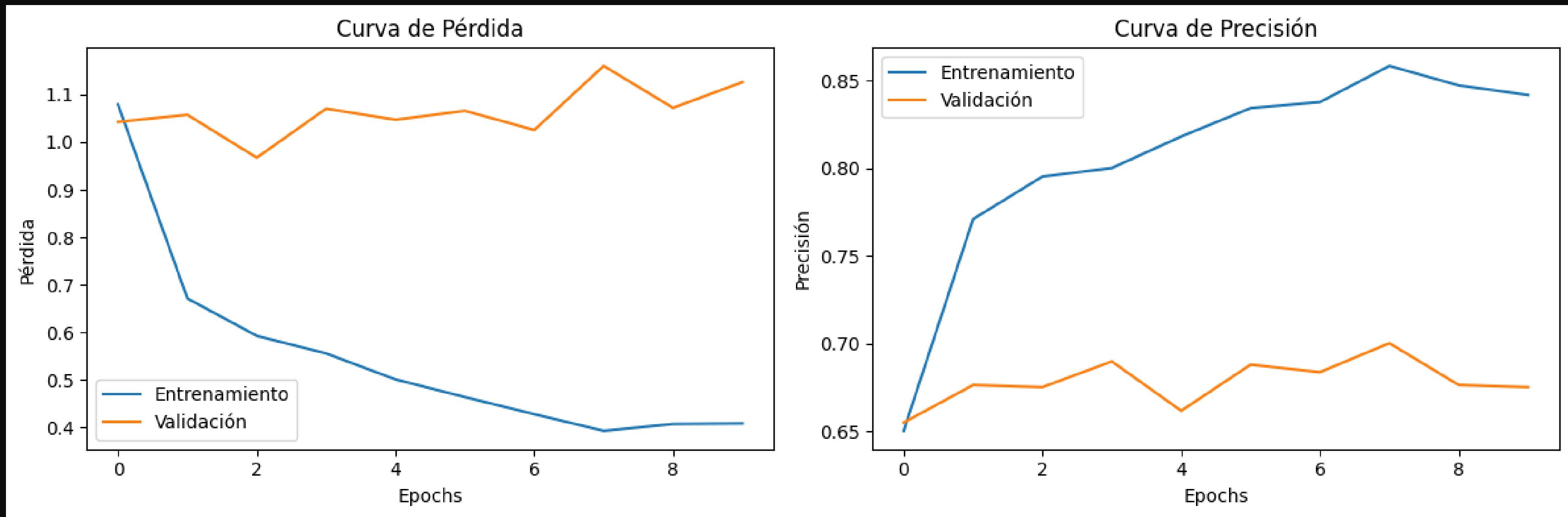
# Modelo final

## RED NEURONAL CONVOLUCIONAL VGG16

block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_2 (Flatten)	(None, 25088)	0
dense_2 (Dense)	(None, 128)	3211392
dropout (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 8)	1032
=====		
Total params: 17,927,112		
Trainable params: 3,212,424		
Non-trainable params: 14,714,688		

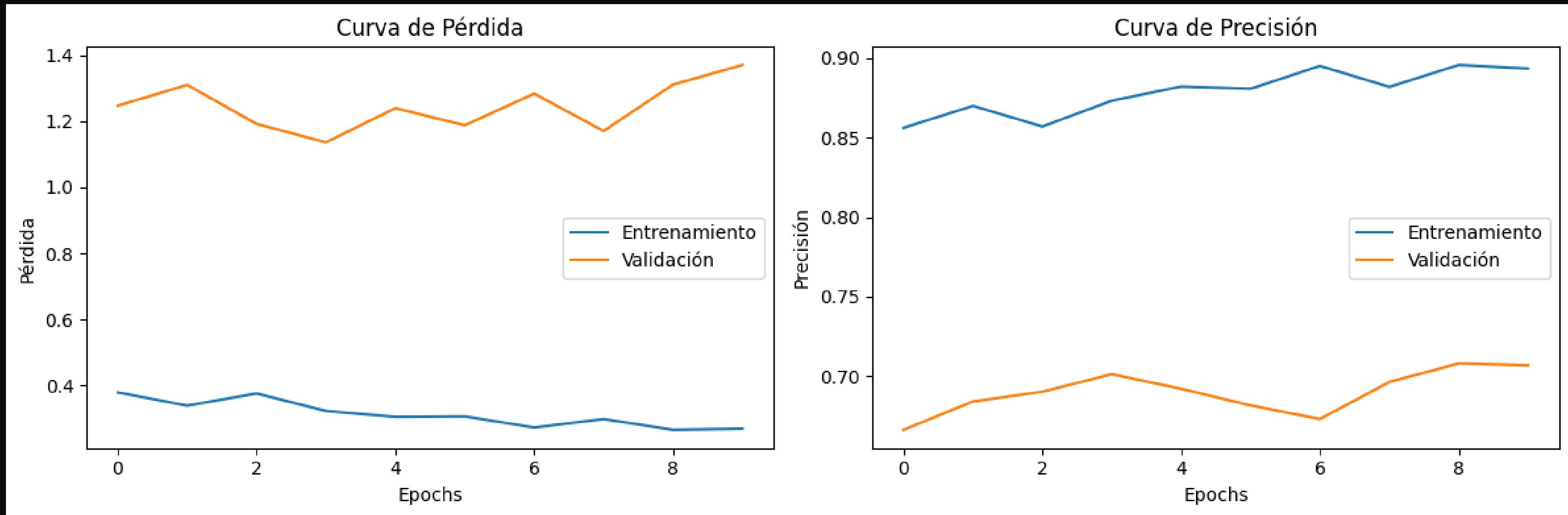
# Modelo final

## RED NEURONAL CONVOLUCIONAL VGG16

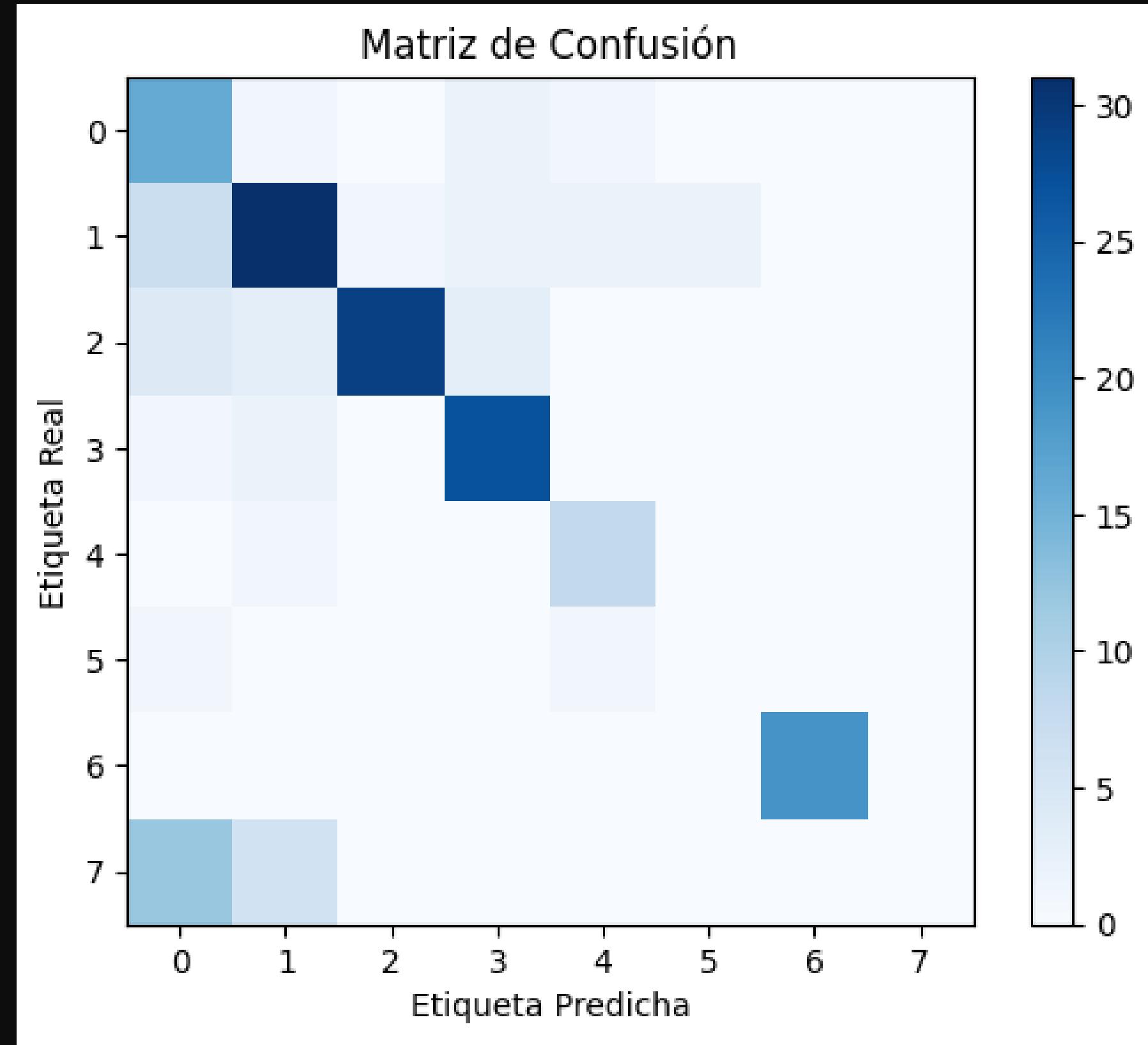


# Modelo final

## RED NEURONAL CONVOLUCIONAL VGG16

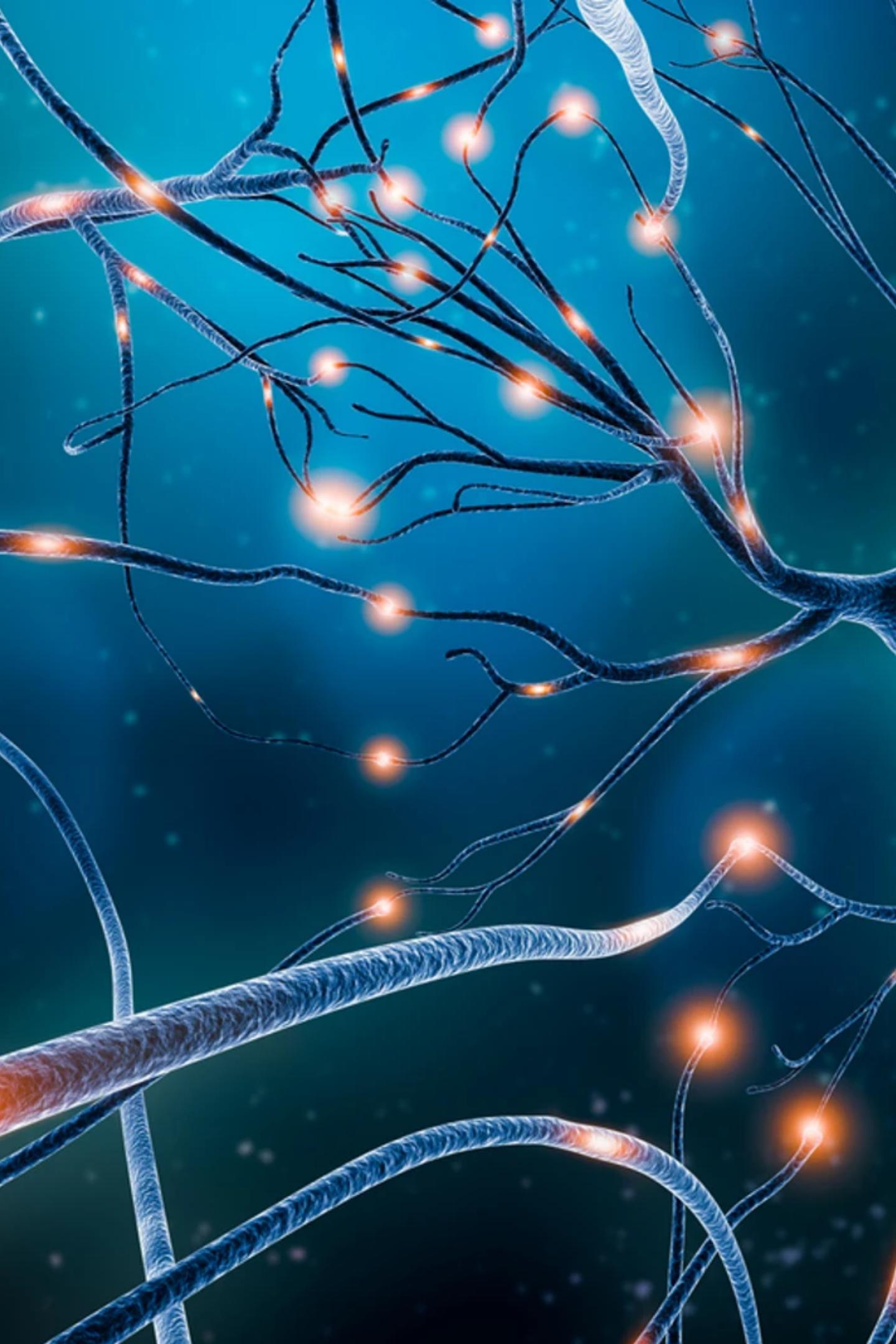


# Evaluación del modelo

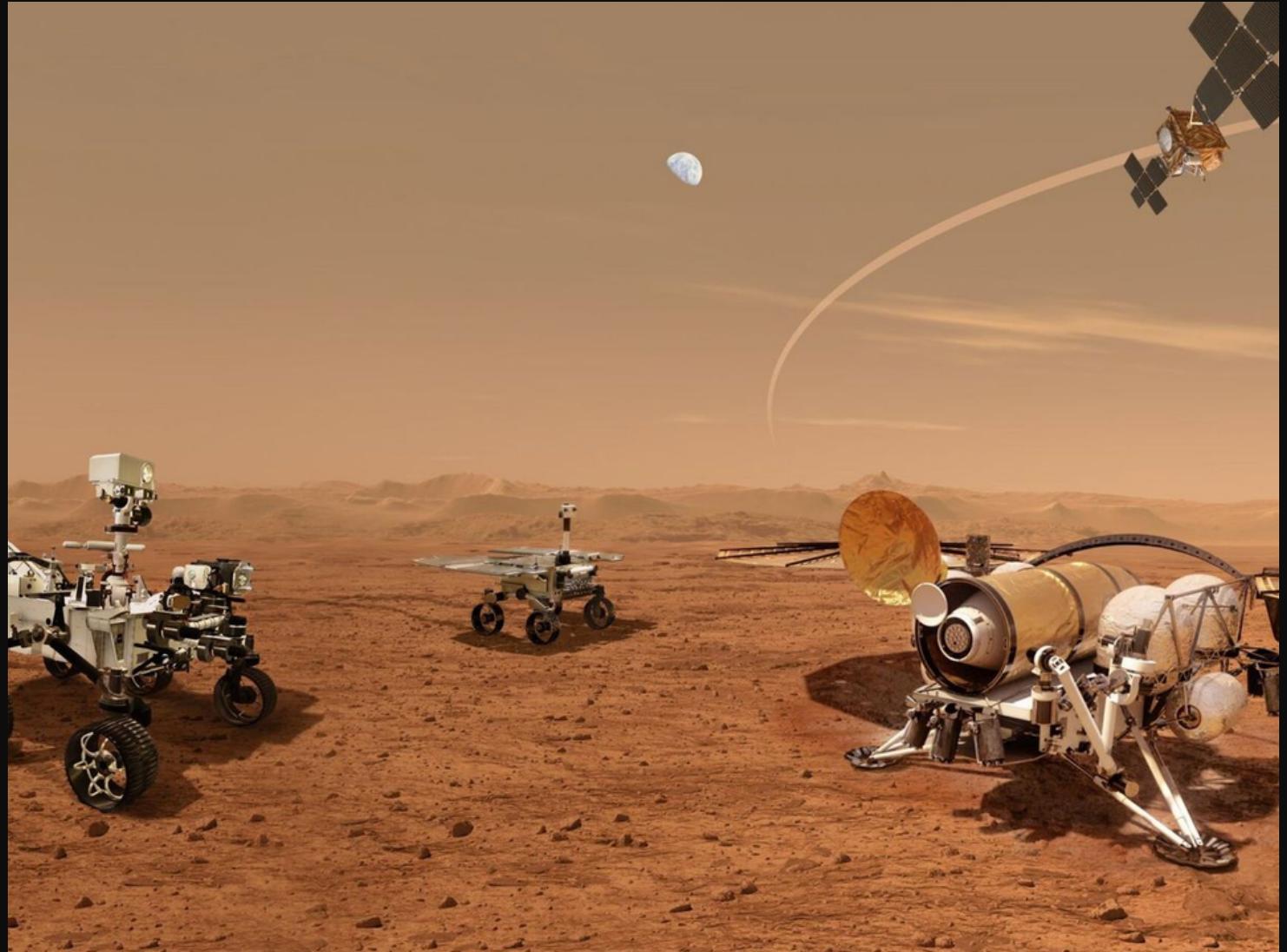


# Resultados

- Los dos primeros modelos, las redes convolucionales creadas por nosotros, mantuvieron la precisión debajo de 0.5
- El modelo Resnet mejoró pero no fue lo suficiente preciso
- El modelo VGG16 tuvo mejor ajuste debido a que es una red convolucional profunda y usaba menos parámetros que los otros modelos



# Conclusiones



- El preprocessamiento de los datos mejora significativamente el entrenamiento y precisión del modelo .
- Usar modelos realizados anteriormente por alguien más es una buena base de partida para lograr una buena aplicación
- Las modificaciones a las imágenes, como por ejemplo al blanco y negro hace que se pierda información para ajustar el modelo
- El último modelo puede llegar a ser mejor con más épocas de entrenamiento, aproximadamente 30

# Referencias



---

## *Base de datos*

Gary Doran, Emily Dunkel, Steven Lu y Kiri Wagstaff. (2020). Conjunto de datos etiquetados de imagen orbital de Marte (HiRISE) versión 3.2 (3.2.0) [Conjunto de datos]. Zenodo. <https://doi.org/10.5281/zenodo.4002935>

---

## *Modelo Resnet*

Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun}, Deep Residual Learning for Image Recognition, arXiv preprint arXiv:1512.03385, 2015

---

## *Redes convolucionales muy profundas*

Karen Simonyan, [arXiv:1409.1556v6](#), <https://doi.org/10.48550/arXiv.1409.1556>

# Gracias

