Classificaço de Patologias usando Imagens Médicas

Carregar imagens do diretório

In [1]:
```
import os
current_dir = os.path.abspath(os.getcwd())
```

Converter base de dados para treino, validação e teste

In [2]:
```
#cria nova pasta para cachorros e gatos atendendo a estrutura do Keras/Tenso
folder = "/novo"
train_folder = current_dir + folder + "/train"
val_folder = current_dir + folder + "/val"
test_folder = current_dir + folder + "/test"

model_filepath = "keras/classificacao_02_03.keras"
conversao_path = "conversao/conversao_02_03"
```

Fazer o Tensorflow carregar as imagens para a RNA

In [3]:
```
import tensorflow as tf

print(tf.config.list_physical_devices('GPU'))
print(tf.__version__)
```

```
2022-06-28 22:00:50.274870: W tensorflow/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: l
ibcudart.so.11.0: cannot open shared object file: No such file or directory
2022-06-28 22:00:50.274885: I tensorflow/stream_executor/cuda/cudart_stub.cc:
29] Ignore above cudart dlerror if you do not have a GPU set up on your machi
ne.
[]
2.6.1
```

```
2022-06-28 22:00:50.976439: W tensorflow/stream_executor/platform/default/dso
_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcud
a.so.1: cannot open shared object file: No such file or directory
2022-06-28 22:00:50.976459: W tensorflow/stream_executor/cuda/cuda_driver.cc:
269] failed call to cuInit: UNKNOWN ERROR (303)
2022-06-28 22:00:50.976476: I tensorflow/stream_executor/cuda/cuda_diagnostic
s.cc:156] kernel driver does not appear to be running on this host (pc): /pro
c/driver/nvidia/version does not exist
```

In [4]:
```
from tensorflow.keras.utils import image_dataset_from_directory
#image_dataset_from_directory monta uma estrutura de dados com imagens 180x1
# de 32 em 32 imagens
train_dataset = image_dataset_from_directory(train_folder, image_size=(180, 1

validation_dataset = image_dataset_from_directory(val_folder,image_size=(180,
test_dataset = image_dataset_from_directory(test_folder, image_size=(180, 18
```

```
Found 34931 files belonging to 2 classes.
Found 16 files belonging to 2 classes.
Found 484 files belonging to 2 classes.
2022-06-28 22:00:51.507237: I tensorflow/core/platform/cpu_feature_guard.cc:1
42] This TensorFlow binary is optimized with oneAPI Deep Neural Network Libra
ry (oneDNN) to use the following CPU instructions in performance-critical ope
rations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate c
ompiler flags.
```

In [5]:

```
#
for data_batch, labels_batch in train_dataset:
    print("data batch shape:", data_batch.shape)
    print("labels batch shape:", labels_batch.shape)
    print(data_batch[0].shape)
    break
```

```
2022-06-28 22:00:51.562763: I tensorflow/compiler/mlir/mlir_graph_optimizatio
n_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered
2)
data batch shape: (32, 180, 180, 3)
labels batch shape: (32,)
(180, 180, 3)
```

Treinando o modelo

In [6]:

```
from tensorflow import keras
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.layers.experimental.preprocessing import Rescaling

#cria uma arquitetura de uma rede neural profunda vazia
model = keras.Sequential()
model.add(Rescaling(scale=1.0/255))
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(Flatten())
model.add(Dense(1, activation="sigmoid"))
model.compile(loss="binary_crossentropy",optimizer="adam",metrics=["accuracy"
#model.add(Dense(4, activation='softmax'))
#model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['ac
```

In [7]:

```
from tensorflow.keras.callbacks import ModelCheckpoint

callbacks = [
    ModelCheckpoint(
        filepath = model_filepath,
        save_best_only = True,
        monitor = "val_loss"
    )
]

history = model.fit(
    train_dataset,
    epochs=100,
    validation_data=validation_dataset,
    callbacks=callbacks)
```

```
Epoch 1/100
1092/1092 [==============================] - 331s 303ms/step - loss: 0.4803 -
accuracy: 0.7815 - val_loss: 0.2444 - val_accuracy: 0.8750
Epoch 2/100
1092/1092 [==============================] - 321s 294ms/step - loss: 0.3008 -
accuracy: 0.8782 - val_loss: 0.1097 - val_accuracy: 1.0000
Epoch 3/100
1092/1092 [==============================] - 319s 292ms/step - loss: 0.2479 -
accuracy: 0.9076 - val_loss: 0.2194 - val_accuracy: 0.8125
Epoch 4/100
1092/1092 [==============================] - 319s 292ms/step - loss: 0.1958 -
accuracy: 0.9287 - val_loss: 0.1275 - val_accuracy: 0.9375
Epoch 5/100
1092/1092 [==============================] - 319s 292ms/step - loss: 0.1402 -
accuracy: 0.9502 - val_loss: 0.0861 - val_accuracy: 0.9375
Epoch 6/100
1092/1092 [==============================] - 319s 292ms/step - loss: 0.0940 -
accuracy: 0.9688 - val_loss: 0.1164 - val_accuracy: 0.9375
Epoch 7/100
1092/1092 [==============================] - 319s 292ms/step - loss: 0.0581 -
accuracy: 0.9816 - val_loss: 0.0182 - val_accuracy: 1.0000
Epoch 8/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0333 -
accuracy: 0.9903 - val_loss: 0.0216 - val_accuracy: 1.0000
Epoch 9/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0265 -
accuracy: 0.9921 - val_loss: 0.0194 - val_accuracy: 1.0000
Epoch 10/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0207 -
accuracy: 0.9938 - val_loss: 8.6307e-04 - val_accuracy: 1.0000
Epoch 11/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0160 -
accuracy: 0.9950 - val_loss: 3.6635e-04 - val_accuracy: 1.0000
Epoch 12/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0165 -
accuracy: 0.9950 - val_loss: 5.0544e-04 - val_accuracy: 1.0000
Epoch 13/100
1092/1092 [==============================] - 317s 290ms/step - loss: 0.0149 -
accuracy: 0.9952 - val_loss: 5.5288e-04 - val_accuracy: 1.0000
Epoch 14/100
1092/1092 [==============================] - 317s 290ms/step - loss: 0.0105 -
accuracy: 0.9970 - val_loss: 1.3414e-04 - val_accuracy: 1.0000
Epoch 15/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0126 -
accuracy: 0.9965 - val_loss: 0.0018 - val_accuracy: 1.0000
Epoch 16/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0119 -
accuracy: 0.9960 - val_loss: 2.4300e-05 - val_accuracy: 1.0000
Epoch 17/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0109 -
accuracy: 0.9970 - val_loss: 1.4156e-04 - val_accuracy: 1.0000
Epoch 18/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0089 -
accuracy: 0.9979 - val_loss: 4.5876e-06 - val_accuracy: 1.0000
Epoch 19/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0075 -
accuracy: 0.9979 - val_loss: 1.9808e-05 - val_accuracy: 1.0000
Epoch 20/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0093 -
accuracy: 0.9974 - val_loss: 1.3595e-04 - val_accuracy: 1.0000
Epoch 21/100
```

```
1092/1092 [==============================] - 317s 290ms/step - loss: 0.0088 -
accuracy: 0.9976 - val_loss: 8.7040e-06 - val_accuracy: 1.0000
Epoch 22/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0094 -
accuracy: 0.9977 - val_loss: 1.0753e-04 - val_accuracy: 1.0000
Epoch 23/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0074 -
accuracy: 0.9985 - val_loss: 3.7298e-05 - val_accuracy: 1.0000
Epoch 24/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0095 -
accuracy: 0.9979 - val_loss: 1.6303e-05 - val_accuracy: 1.0000
Epoch 25/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0068 -
accuracy: 0.9987 - val_loss: 0.0012 - val_accuracy: 1.0000
Epoch 26/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0080 -
accuracy: 0.9977 - val_loss: 5.4656e-05 - val_accuracy: 1.0000
Epoch 27/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0075 -
accuracy: 0.9983 - val_loss: 4.9920e-07 - val_accuracy: 1.0000
Epoch 28/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0057 -
accuracy: 0.9987 - val_loss: 9.2779e-06 - val_accuracy: 1.0000
Epoch 29/100
1092/1092 [==============================] - 316s 290ms/step - loss: 0.0083 -
accuracy: 0.9976 - val_loss: 3.4694e-04 - val_accuracy: 1.0000
Epoch 30/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0038 -
accuracy: 0.9993 - val_loss: 0.0015 - val_accuracy: 1.0000
Epoch 31/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0076 -
accuracy: 0.9983 - val_loss: 8.2806e-04 - val_accuracy: 1.0000
Epoch 32/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0060 -
accuracy: 0.9989 - val_loss: 1.4387e-04 - val_accuracy: 1.0000
Epoch 33/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0065 -
accuracy: 0.9985 - val_loss: 7.4121e-04 - val_accuracy: 1.0000
Epoch 34/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0061 -
accuracy: 0.9988 - val_loss: 1.5277e-04 - val_accuracy: 1.0000
Epoch 35/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0056 -
accuracy: 0.9986 - val_loss: 4.8540e-04 - val_accuracy: 1.0000
Epoch 36/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0048 -
accuracy: 0.9992 - val_loss: 8.7021e-05 - val_accuracy: 1.0000
Epoch 37/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0075 -
accuracy: 0.9989 - val_loss: 3.7087e-04 - val_accuracy: 1.0000
Epoch 38/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0050 -
accuracy: 0.9990 - val_loss: 4.6718e-06 - val_accuracy: 1.0000
Epoch 39/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0042 -
accuracy: 0.9991 - val_loss: 2.4948e-04 - val_accuracy: 1.0000
Epoch 40/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0050 -
accuracy: 0.9991 - val_loss: 4.1633e-08 - val_accuracy: 1.0000
Epoch 41/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0082 -
accuracy: 0.9987 - val_loss: 1.8814e-04 - val_accuracy: 1.0000
Epoch 42/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0046 -
```

```
accuracy: 0.9989 - val_loss: 1.5844e-06 - val_accuracy: 1.0000
Epoch 43/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0057 -
accuracy: 0.9989 - val_loss: 5.4654e-05 - val_accuracy: 1.0000
Epoch 44/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0050 -
accuracy: 0.9989 - val_loss: 4.2107e-06 - val_accuracy: 1.0000
Epoch 45/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0043 -
accuracy: 0.9993 - val_loss: 1.4065e-04 - val_accuracy: 1.0000
Epoch 46/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0059 -
accuracy: 0.9987 - val_loss: 0.0064 - val_accuracy: 1.0000
Epoch 47/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0048 -
accuracy: 0.9991 - val_loss: 4.0472e-04 - val_accuracy: 1.0000
Epoch 48/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0042 -
accuracy: 0.9995 - val_loss: 1.4891e-06 - val_accuracy: 1.0000
Epoch 49/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0059 -
accuracy: 0.9989 - val_loss: 9.2724e-04 - val_accuracy: 1.0000
Epoch 50/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0048 -
accuracy: 0.9992 - val_loss: 1.8490e-06 - val_accuracy: 1.0000
Epoch 51/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0039 -
accuracy: 0.9993 - val_loss: 3.4134e-06 - val_accuracy: 1.0000
Epoch 52/100
1092/1092 [==============================] - 317s 290ms/step - loss: 0.0042 -
accuracy: 0.9992 - val_loss: 2.5919e-05 - val_accuracy: 1.0000
Epoch 53/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0048 -
accuracy: 0.9991 - val_loss: 7.7228e-05 - val_accuracy: 1.0000
Epoch 54/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0058 -
accuracy: 0.9990 - val_loss: 3.3098e-04 - val_accuracy: 1.0000
Epoch 55/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0032 -
accuracy: 0.9996 - val_loss: 1.5359e-05 - val_accuracy: 1.0000
Epoch 56/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0045 -
accuracy: 0.9992 - val_loss: 3.0660e-06 - val_accuracy: 1.0000
Epoch 57/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0040 -
accuracy: 0.9993 - val_loss: 2.4181e-06 - val_accuracy: 1.0000
Epoch 58/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0040 -
accuracy: 0.9993 - val_loss: 0.0072 - val_accuracy: 1.0000
Epoch 59/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0042 -
accuracy: 0.9992 - val_loss: 6.9940e-05 - val_accuracy: 1.0000
Epoch 60/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0050 -
accuracy: 0.9991 - val_loss: 1.7703e-07 - val_accuracy: 1.0000
Epoch 61/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0047 -
accuracy: 0.9992 - val_loss: 1.0900e-04 - val_accuracy: 1.0000
Epoch 62/100
1092/1092 [==============================] - 314s 287ms/step - loss: 0.0035 -
accuracy: 0.9995 - val_loss: 2.3213e-06 - val_accuracy: 1.0000
Epoch 63/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0052 -
accuracy: 0.9993 - val_loss: 9.0748e-04 - val_accuracy: 1.0000
```

```
Epoch 64/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0043 -
accuracy: 0.9992 - val_loss: 5.5922e-04 - val_accuracy: 1.0000
Epoch 65/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0040 -
accuracy: 0.9992 - val_loss: 3.1599e-07 - val_accuracy: 1.0000
Epoch 66/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0051 -
accuracy: 0.9991 - val_loss: 9.5662e-06 - val_accuracy: 1.0000
Epoch 67/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0052 -
accuracy: 0.9993 - val_loss: 6.2262e-04 - val_accuracy: 1.0000
Epoch 68/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0050 -
accuracy: 0.9992 - val_loss: 2.4832e-04 - val_accuracy: 1.0000
Epoch 69/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0030 -
accuracy: 0.9997 - val_loss: 3.3305e-05 - val_accuracy: 1.0000
Epoch 70/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0027 -
accuracy: 0.9996 - val_loss: 1.1397e-04 - val_accuracy: 1.0000
Epoch 71/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0037 -
accuracy: 0.9992 - val_loss: 3.9482e-07 - val_accuracy: 1.0000
Epoch 72/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0054 -
accuracy: 0.9990 - val_loss: 6.1107e-05 - val_accuracy: 1.0000
Epoch 73/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0027 -
accuracy: 0.9997 - val_loss: 7.8678e-05 - val_accuracy: 1.0000
Epoch 74/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0028 -
accuracy: 0.9995 - val_loss: 2.0989e-06 - val_accuracy: 1.0000
Epoch 75/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0027 -
accuracy: 0.9996 - val_loss: 2.1671e-04 - val_accuracy: 1.0000
Epoch 76/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0033 -
accuracy: 0.9994 - val_loss: 0.0012 - val_accuracy: 1.0000
Epoch 77/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0052 -
accuracy: 0.9993 - val_loss: 1.8847e-06 - val_accuracy: 1.0000
Epoch 78/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0021 -
accuracy: 0.9998 - val_loss: 6.6884e-06 - val_accuracy: 1.0000
Epoch 79/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0045 -
accuracy: 0.9992 - val_loss: 8.4854e-06 - val_accuracy: 1.0000
Epoch 80/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0025 -
accuracy: 0.9997 - val_loss: 1.0245e-06 - val_accuracy: 1.0000
Epoch 81/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0036 -
accuracy: 0.9993 - val_loss: 1.8646e-07 - val_accuracy: 1.0000
Epoch 82/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0045 -
accuracy: 0.9994 - val_loss: 5.0891e-06 - val_accuracy: 1.0000
Epoch 83/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0025 -
accuracy: 0.9996 - val_loss: 2.9979e-08 - val_accuracy: 1.0000
Epoch 84/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0038 -
accuracy: 0.9993 - val_loss: 1.0366e-06 - val_accuracy: 1.0000
Epoch 85/100
```

```
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0033 -
accuracy: 0.9995 - val_loss: 1.8609e-07 - val_accuracy: 1.0000
Epoch 86/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0031 -
accuracy: 0.9994 - val_loss: 6.1276e-07 - val_accuracy: 1.0000
Epoch 87/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0027 -
accuracy: 0.9997 - val_loss: 1.8801e-06 - val_accuracy: 1.0000
Epoch 88/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0019 -
accuracy: 0.9996 - val_loss: 3.1015e-09 - val_accuracy: 1.0000
Epoch 89/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0034 -
accuracy: 0.9993 - val_loss: 3.1217e-06 - val_accuracy: 1.0000
Epoch 90/100
1092/1092 [==============================] - 314s 288ms/step - loss: 0.0042 -
accuracy: 0.9995 - val_loss: 1.7480e-08 - val_accuracy: 1.0000
Epoch 91/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0026 -
accuracy: 0.9997 - val_loss: 1.2252e-05 - val_accuracy: 1.0000
Epoch 92/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0033 -
accuracy: 0.9994 - val_loss: 1.2314e-06 - val_accuracy: 1.0000
Epoch 93/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0025 -
accuracy: 0.9996 - val_loss: 1.5645e-05 - val_accuracy: 1.0000
Epoch 94/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0029 -
accuracy: 0.9996 - val_loss: 4.6160e-04 - val_accuracy: 1.0000
Epoch 95/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0026 -
accuracy: 0.9996 - val_loss: 0.0052 - val_accuracy: 1.0000
Epoch 96/100
1092/1092 [==============================] - 315s 289ms/step - loss: 0.0028 -
accuracy: 0.9994 - val_loss: 2.7932e-08 - val_accuracy: 1.0000
Epoch 97/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0025 -
accuracy: 0.9996 - val_loss: 1.0433e-06 - val_accuracy: 1.0000
Epoch 98/100
1092/1092 [==============================] - 316s 289ms/step - loss: 0.0023 -
accuracy: 0.9997 - val_loss: 5.6527e-08 - val_accuracy: 1.0000
Epoch 99/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0024 -
accuracy: 0.9996 - val_loss: 5.1649e-05 - val_accuracy: 1.0000
Epoch 100/100
1092/1092 [==============================] - 315s 288ms/step - loss: 0.0025 -
accuracy: 0.9996 - val_loss: 4.9131e-06 - val_accuracy: 1.0000
```

In [8]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling (Rescaling)        (None, 180, 180, 3)       0
_________________________________________________________________
conv2d (Conv2D)              (None, 178, 178, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 89, 89, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 87, 87, 64)        18496
_________________________________________________________________
flatten (Flatten)            (None, 484416)            0
```

```
dense (Dense)                (None, 1)                 484417
=================================================================
Total params: 503,809
Trainable params: 503,809
Non-trainable params: 0

_________________________________________________________________
```

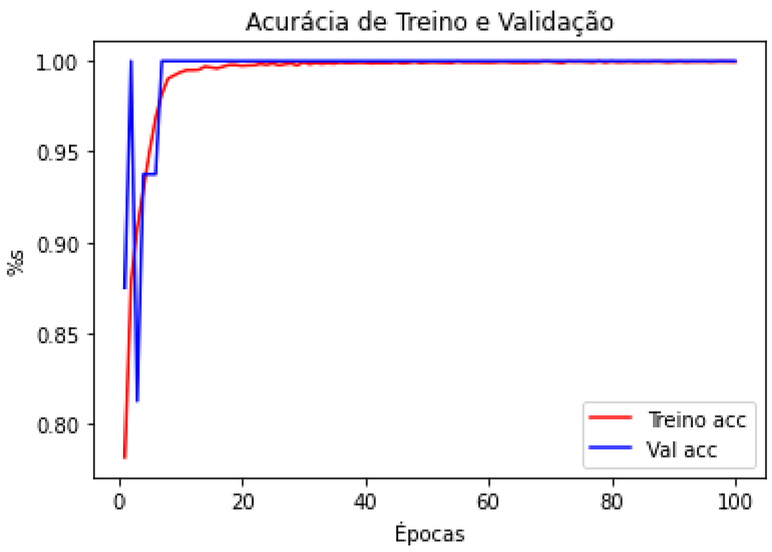
In [9]:

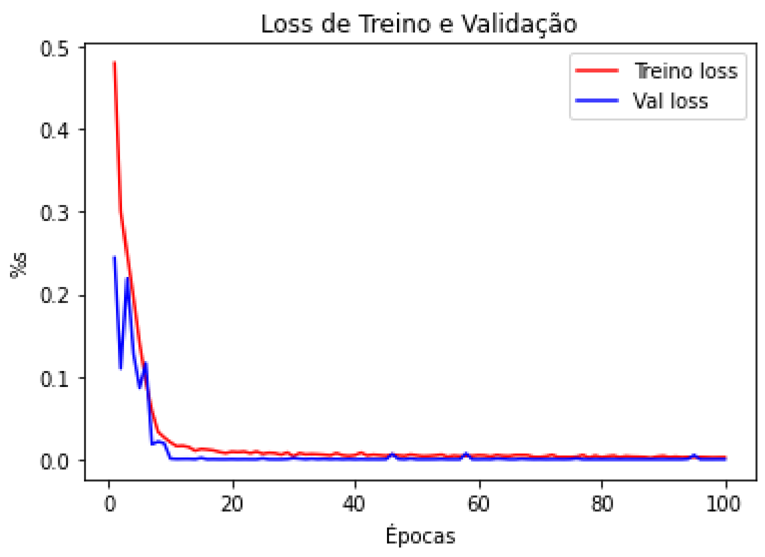
```
#https://www.tensorflow.org/js/tutorials/conversion/import_keras?hl=pt-br#alt
import tensorflowjs as tfjs
tfjs.converters.save_keras_model(model, conversao_path)
```

Visualização de Resultados

In [10]:

```
import matplotlib.pyplot as plt
accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
loss = history.history["loss"]
val_loss = history.history["val_loss"]
epochs = range(1, len(accuracy) + 1)
plt.plot(epochs, accuracy, "r", label="Treino acc")
plt.plot(epochs, val_accuracy, "b", label="Val acc")
plt.xlabel("Épocas")
plt.ylabel("%s")
plt.title("Acurácia de Treino e Validação")
plt.legend()
plt.figure()
plt.plot(epochs, loss, "r", label="Treino loss")
plt.plot(epochs, val_loss, "b", label="Val loss")
plt.xlabel("Épocas")
plt.ylabel("%s")
plt.title("Loss de Treino e Validação")
plt.legend()
plt.show()
```





Resultados do Conjunto de Teste

In [11]:
```
from tensorflow import keras
model = keras.models.load_model(model_filepath)
```

In [12]:
```
test_loss, test_acc = model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")
```

```
16/16 [==============================] - 1s 56ms/step - loss: 0.0961 - accura
cy: 0.9773
Test accuracy: 0.977
```

Referências

- https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/
- https://stackoverflow.com/questions/3430372/how-do-i-get-the-full-path-of-the-current-files-directory
- https://www.geeksforgeeks.org/python-list-files-in-a-directory/
- https://pynative.com/python-random-sample/
- https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/
- https://www.mygreatlearning.com/blog/keras-tutorial/
- https://www.machinecurve.com/index.php/2020/03/30/how-to-use-conv2d-with-keras/
- https://www.pyimagesearch.com/2021/06/30/how-to-use-the-modelcheckpoint-callback-with-keras-and-tensorflow/