

Reconocimiento de células infectadas por malaria

En este fichero trabajaremos con un dataset de imágenes de células. Nuestro objetivo será detectar si una célula está infectada por la malaria. Lo primero que hacemos es importar las librerías necesarias para desarrollar el trabajo.

```
In [1]: import tensorflow as tf
import os
import skimage.data as ind
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import collections
import keras
from keras import layers
from keras import models
keras.__version__

Using TensorFlow backend.
```

Out[1]: '2.2.4'

Preparación de los datos

A continuación, procedemos a la carga de los datos. Preparamos todas las imágenes y sus etiquetas.

```
In [2]: def load_ml_data():
labels = []
images = []
file_names = [os.path.join("cell_images/Parasitized", f)
               for f in os.listdir("cell_images/Parasitized")
               if f.endswith(".png")]

for f in file_names:
    images.append(ind.imread(f))
    labels.append(0)

file_names = [os.path.join("cell_images/Uninfected", f)
              for f in os.listdir("cell_images/Uninfected")
              if f.endswith(".png")]

for f in file_names:
    img=mpimg.imread(f)
    images.append(img)
    labels.append(1)

return images, labels
```

In [3]: images, labels = load_ml_data()

Dividimos el dataset en subjets de entrenamiento y validación.

```
In [4]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(images, labels, test_size=0.33, random_state=42)
```

In [5]: images = np.array(X_train)
labels = np.array(y_train)

La variables *images* y *labels* tienen una única dimensión, ya que son arrays.

In [6]: images.ndim

Out[6]: 1

Aunque el dataset está formado por 27558 imágenes, lo hemos dividido en dos subjets. De esta forma, tenemos una *subset de entrenamiento* y una *subset de test o validación*. El subset de entrenamiento está formado por 18463 imágenes.

In [7]: images.size

Out[7]: 18463

Vamos a ver qué dimensiones tienen nuestras imágenes.

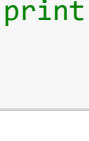
In [8]: # ver que se han almacenado correctamente los pixeles
images[2][45].shape

Out[8]: (148, 3)

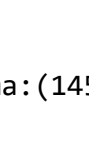
In [9]: import random
rand_signs = random.sample(range(0, len(labels)), 6)
rand_signs

Out[9]: [8568, 6014, 6501, 14380, 4405, 11362]

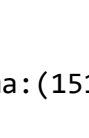
```
In [10]: import matplotlib.pyplot as plt
for i in range(len(rand_signs)):
    temp_im = images[rand_signs[i]]
    plt.subplot(3,3,i+1)
    plt.axis("off")
    imgplot = plt.imshow(temp_im)
    plt.subplots_adjust(wspace = 0.5)
    plt.show()
    print("Forma:{0}, min:{1}, max:{2}".format(temp_im.shape,
                                              temp_im.min(),
                                              temp_im.max()))
```



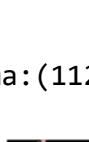
Forma:(145, 136, 3), min:0.0, max:0.8901960849761963



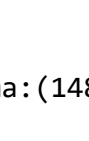
Forma:(151, 145, 3), min:0, max:215



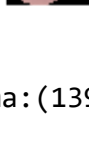
Forma:(112, 118, 3), min:0, max:205



Forma:(148, 127, 3), min:0, max:227

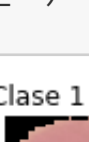
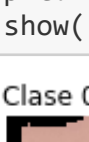


Forma:(139, 118, 3), min:0, max:224



Forma:(151, 118, 3), min:0, max:231

```
In [11]: unique_labels = set(labels)
plt.figure(figsize=(16,16))
i = 1
for label in unique_labels:
    temp_im = images[list(labels).index(label)]
    plt.subplot(3,3,i)
    plt.axis("off")
    plt.title("Clase {0} ({1})".format(label, list(labels).count(label)))
    i +=1
    plt.imshow(temp_im)
plt.show()
```



Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

Clase 0 (9214) Clase 1 (9249)

<