IDENTIFICACIÓN DE ESPECIES DE SERPIENTES MEDIANTE VISIÓN POR COMPUTADOR

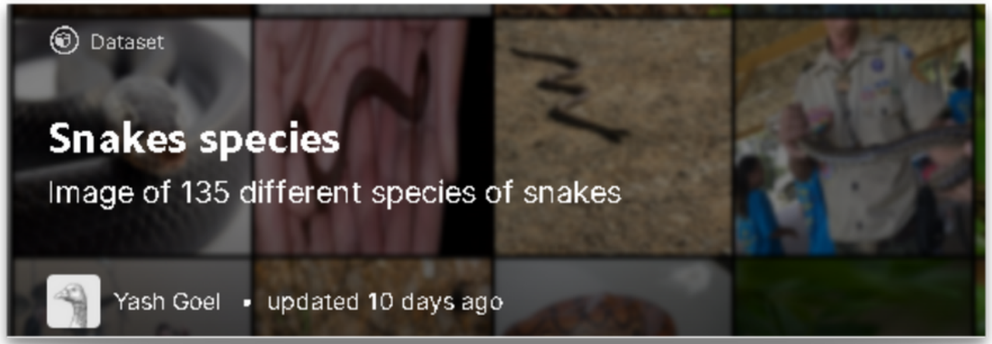
JESUS DANIEL LIZCANO CASTRO - 2171991
JUAN DAVID PORRAS GÓMEZ - 2172000

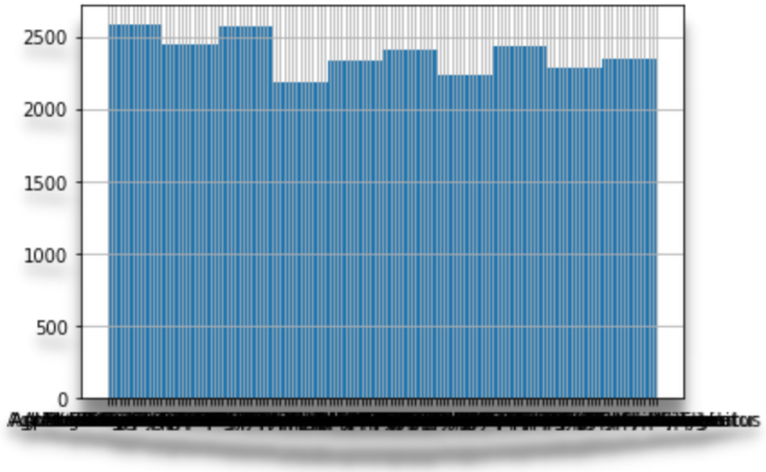
INTRODUCCIÓN

La mordedura de serpiente es una de los problemas tropicales mas mortales y desatendidos. Causa entre 81000 y 138000 muertes al año y alrededor de 400 000 víctimas de discapacidad y desfiguración en todo el mundo. Afecta a muchas comunidades, sobre todo rurales en países en desarrollo con una gran diversidad de serpientes venenosas y con experiencia y acceso a antídotos muy limitados.

Los antídotos pueden salvar vidas si se utilizan correctamente, pero muchas veces dependen de la identificación taxonómica correcta de la serpiente que muerde. Aquí es donde entra en juego el uso del Deep learning, identificando las serpientes a nivel de especie a partir de fotografías de estas.

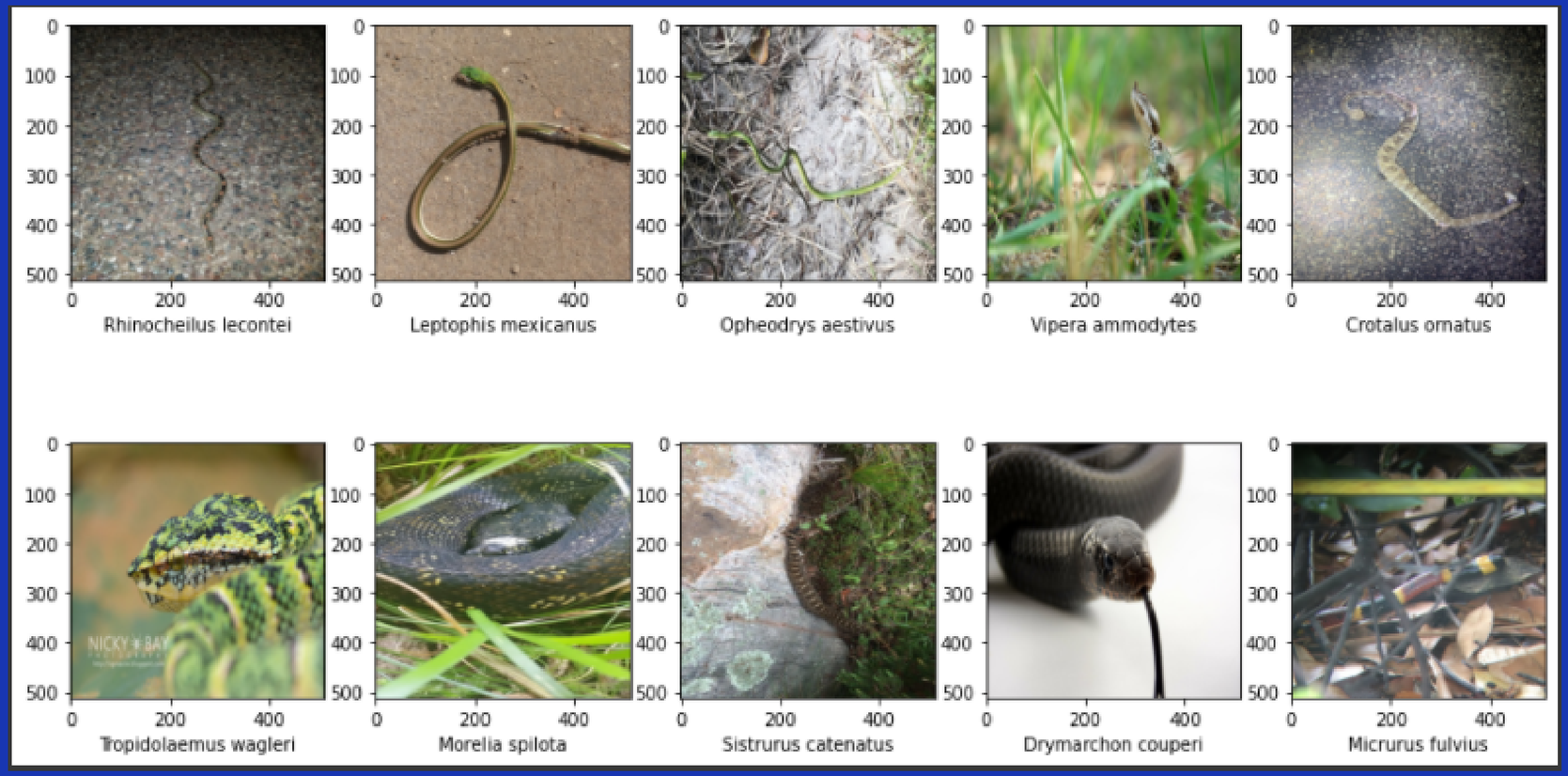
DATASET





binomial	country	continent	genus	family	UUID	class_id	snake_sub_family	poisonous	X	Y	height	width
Agkistrodon contortrix	United States of America	North America	Agkistrodon	Viperidae	20e23008100d4e249fd757c11fe059fe	18	Agkistrodon contortrix	1	0.507412	0.546939	0.916220	0.951425
Agkistrodon contortrix	United States of America	North America	Agkistrodon	Viperidae	0c6d14f33f404013ab116ab09880c523	18	Agkistrodon contortrix	1	0.503115	0.392086	0.835316	0.965363
Agkistrodon contortrix	United States of America	North America	Agkistrodon	Viperidae	3a31a32de0434653b4a82a30806f7a6d	18	Agkistrodon contortrix	1	0.353031	0.573312	0.652700	0.709286
Agkistrodon contortrix	United States of America	North America	Agkistrodon	Viperidae	1c5a3b2953c84d698fad8a40db91323e	18	Agkistrodon contortrix	1	0.436368	0.447794	0.918880	0.834457
Agkistrodon contortrix	United States of America	North America	Agkistrodon	Viperidae	e10c99a58c2546dab2c0d998de1f7c1b	18	Agkistrodon contortrix	1	0.512284	0.519067	0.923202	0.631036

Un pequeño vistazo...



FILTRO DE CLASES

```
def filter(labels,data_df):
  data = pd.DataFrame()
  for i in labels:
    datalabel = data_df[ data_df['label'] == i ]
    data = pd.concat([data,datalabel])
  return data
```

```
N = 100
labels = np.array(list(set(train_DF['label'])))
lbl = np.random.choice(labels, size=N, replace=False)

train_df = filter(lbl,train_DF)
test_df = filter(lbl,test_DF)

train_df.shape, test_df.shape
```

GENERATOR

	path	label
0	test/18/ca23ee722e5e4b6ca4b130291b8428f1.jpg	Agkistrodon contortrix
1	test/18/23dfe346fcbf40e7a465b0febf3620be.jpg	Agkistrodon contortrix
2	test/18/2c96953bd06645bdbfa87b180eb72b0f.jpg	Agkistrodon contortrix
3	test/18/e2bcec60b133466380d5c253088765ae.jpg	Agkistrodon contortrix
4	test/18/fd6a488d065e47978c9996734fc1732c.jpg	Agkistrodon contortrix

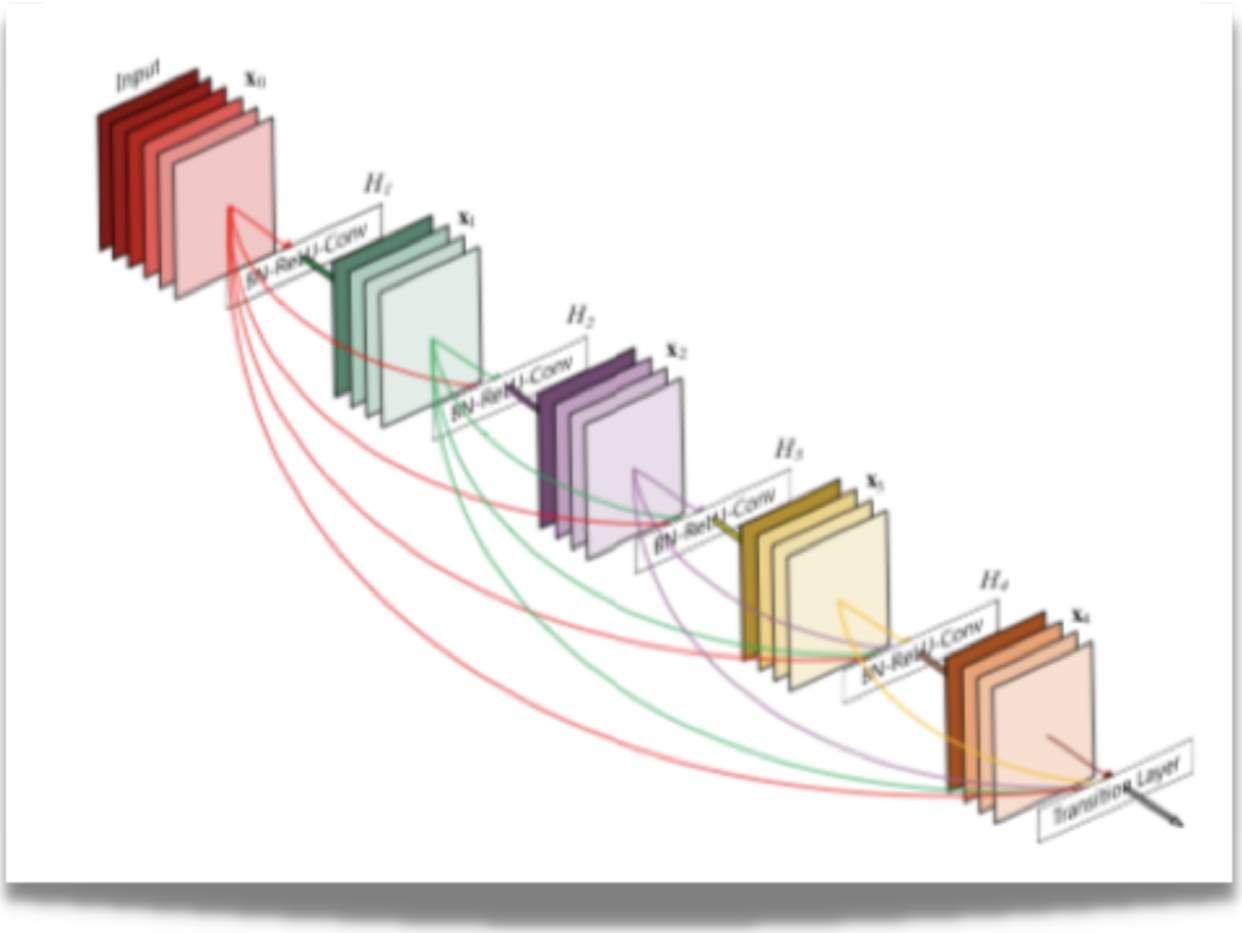
```
generator = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255.,
    horizontal_flip=True,
    vertical_flip=True,
    zoom_range=0.2,
    rotation_range=40,
    brightness_range=[0.4,1.5],
    validation_split=0.2
    )
```

```
train_generator = generator.flow_from_dataframe(
    dataframe=train_df,
    directory=None,
    subset='training',
    x_col='path',
    y_col='label',
    shuffle=True,
    batch_size=N*10,
    target_size=(200,200)
)
```

```
s = next(train_generator)                #Se genera una nueva imagen con su clase
print( speciesTrain[np.argmax(s[1])] ) #Se imprime la clase correspondiente
print(s[1])                              #Se imprime la clase correspondiente en OneHotEncoding
np.argmax(s[1])                          #Indice correspondiente a la clase
```

```
Crotalus molossus
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]]
26
```

DENSENET

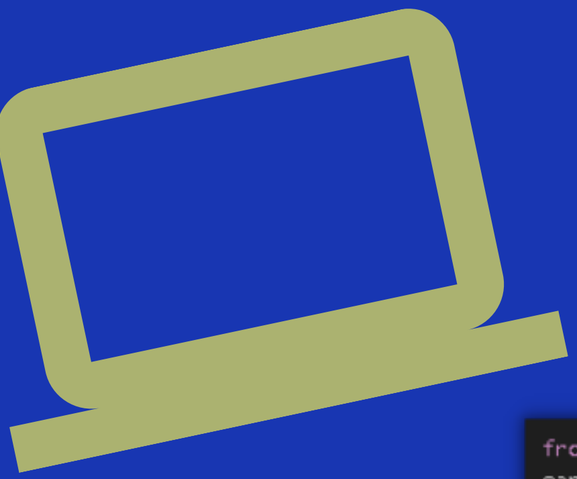

Input
$\mathbf{x}_0$
H_1
BN-ReLU-Conv
$\mathbf{x}_1$
H_2
BN-ReLU-Conv
$\mathbf{x}_2$
H_3
BN-ReLU-Conv
$\mathbf{x}_3$
H_4
BN-ReLU-Conv
$\mathbf{x}_4$
Transition Layer

TRANSFER LEARNING

```
model_Imp =  tf.keras.applications.DenseNet121(input_shape=(200,200, 3), weights='imagenet', include_top=False)
model_Imp.trainable = False
model_Imp.summary()
```

```
prediction_layer = tf.keras.layers.Dense( N , activation='softmax')
flatten_layer = tf.keras.layers.Flatten()
global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
```

```
modeloAgg = tf.keras.Sequential([
  model_Imp,
  global_average_layer,
  prediction_layer
])
```

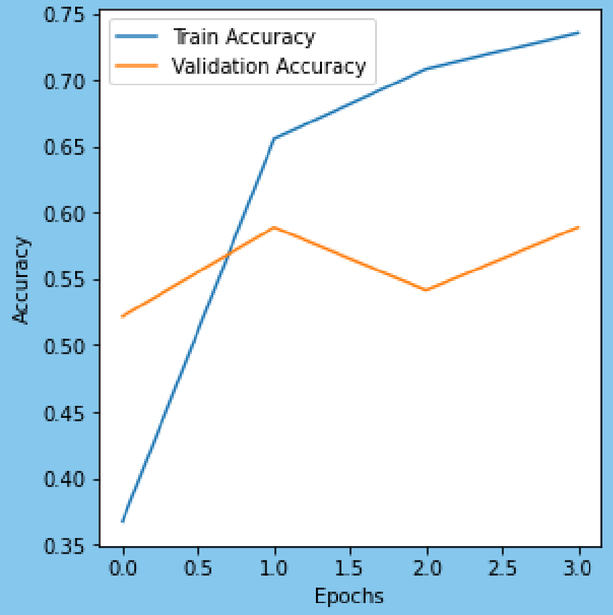


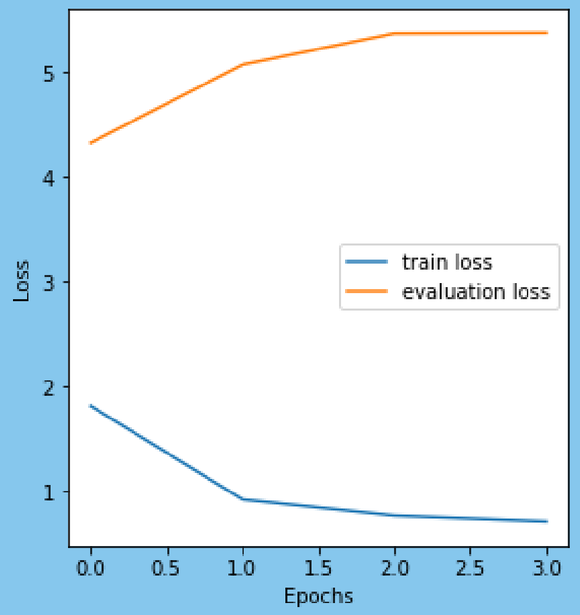
```
from tensorflow.keras.callbacks import EarlyStopping
early_stopping = EarlyStopping(patience=3, restore_best_weights = True)

#learning_rate= 0.1
opt = tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.9)#Adam(learning_rate=learning_rate)
modeloAgg.compile(optimizer=opt, loss='categorical_crossentropy',metrics=['accuracy'])
history = modeloAgg.fit(train_generator, epochs=30, validation_data=(test_images, test_labels), callbacks=[early_stopping])
```

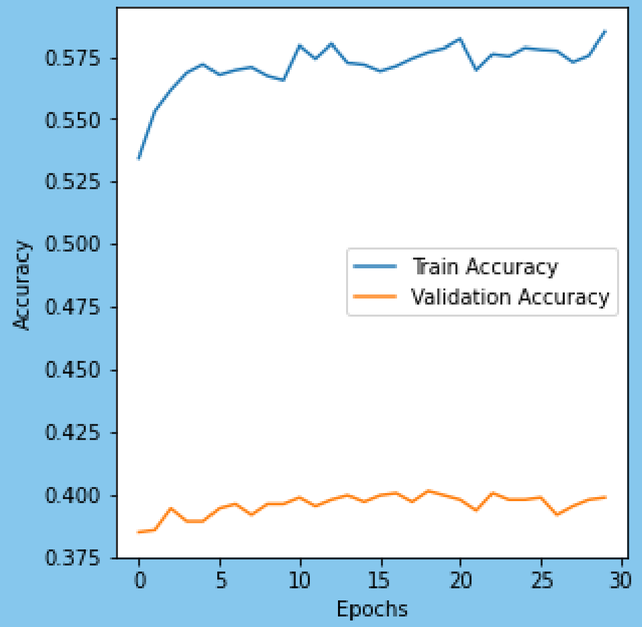
RESULTADOS PARA CADA CANTIDAD DE CLASES:

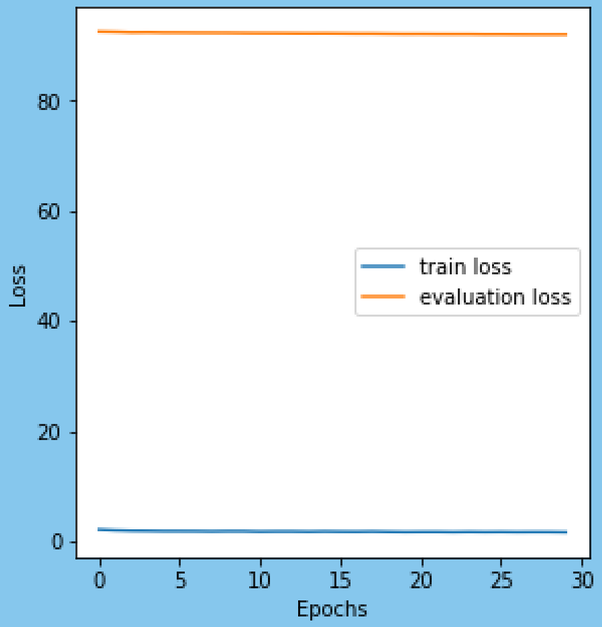
10 CLASES





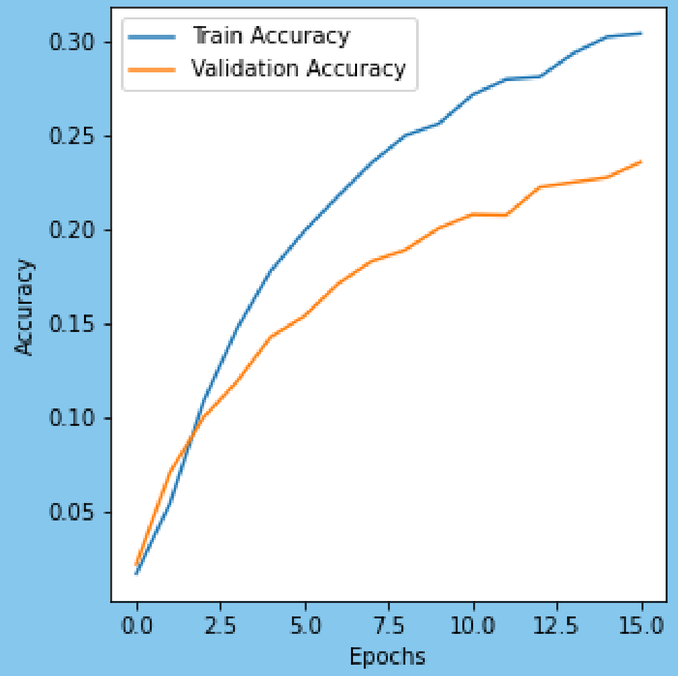
50 CLASES

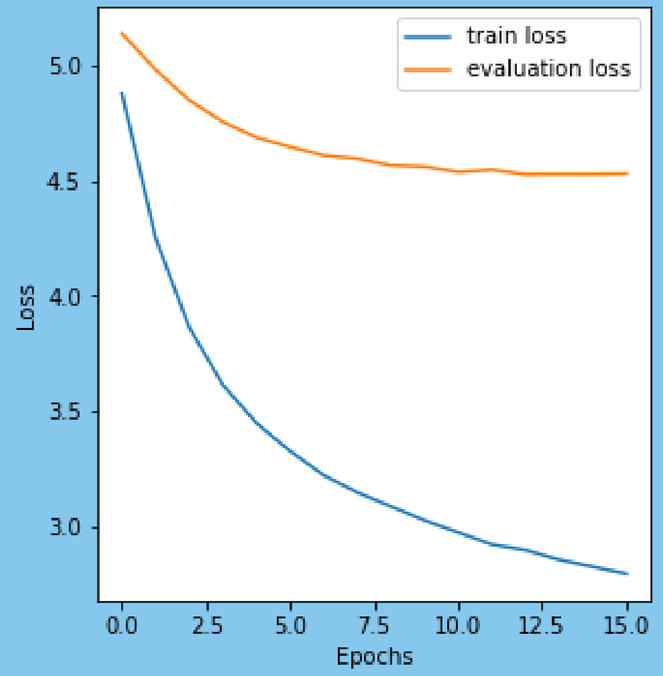




RESULTADOS PARA CADA CANTIDAD DE CLASES:

100 CLASES





PREDICCIONES