# **Índice:**

[Introducción a Tensor Flow:](#_ojkoutlxklu6)

[Yolov5 (v7)](#_os7360wqum8v)

[Yolov6](#_2rew8rx8ey4f)

[Entrenar Yolov5 (detección de mates)](#_6ssaogpou36e)

[Redes Recurrentes (Robo-Shakespeare)](#_yk23kq5bfu9r)

[BERT: análisis de sentimientos](#_r78yjypfmc12)

[RoBERTuito](#_mops6ng9ev57)

[Stable Diffusion](#_kx9x85mzsiwe)

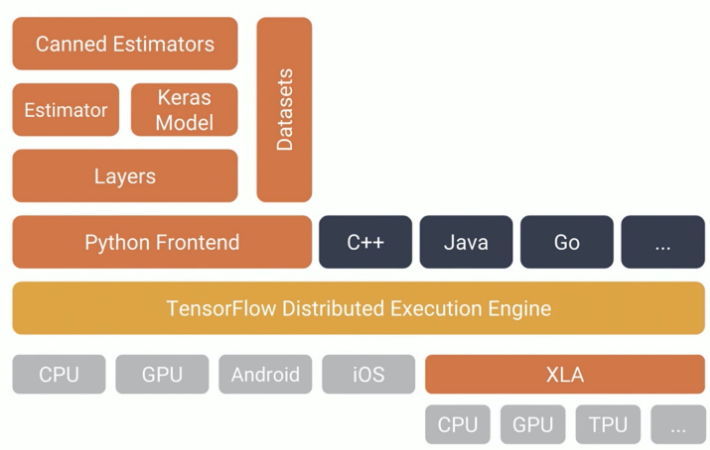
# 

# 

# **Introducción a Tensor Flow:**

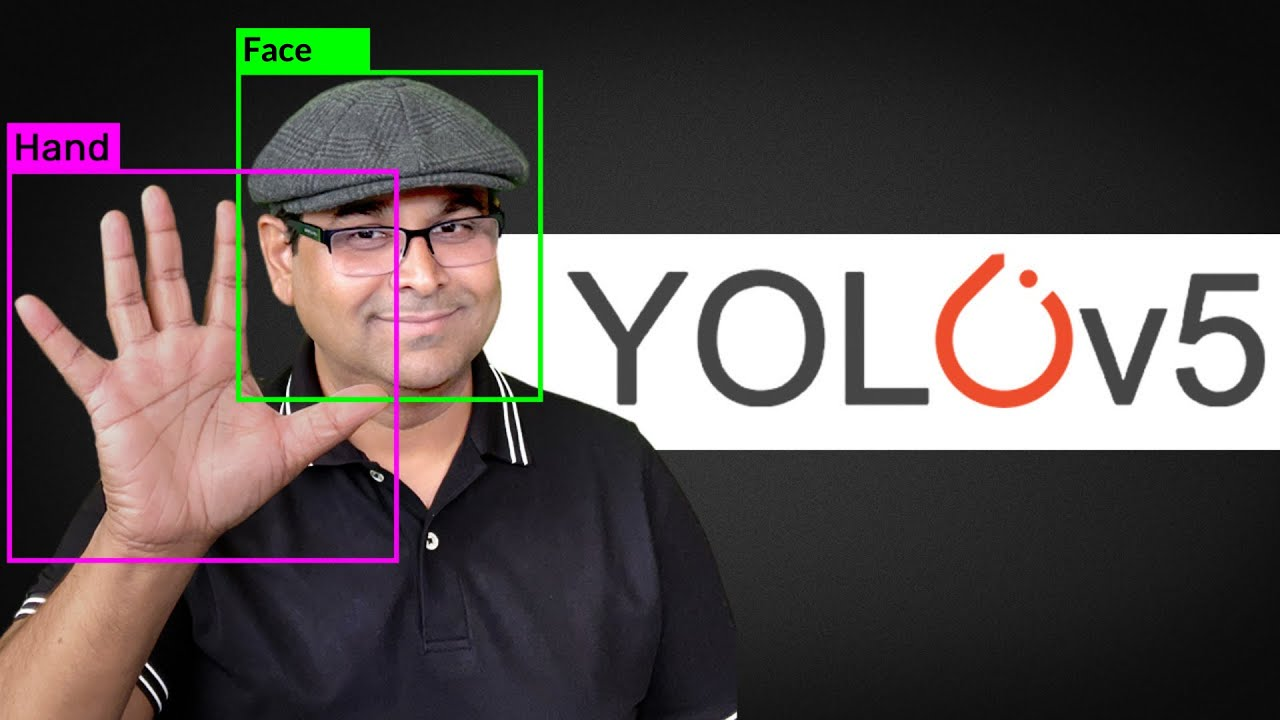
Breve explicación de que es y cómo funciona TensorFlow en la siguiente máquina Colab:

<https://colab.research.google.com/drive/1ar9BsIgiBZSV4cnmfRvbhUkj38dxAwLv?usp=sharing>



# **Yolov5 (v7)**

red convolucional preentrenada para la detección de elementos en imágenes:

Sitio oficial: [GitHub - ultralytics/yolov5: YOLOv5 🚀 in PyTorch > ONNX > CoreML > TFLite](https://github.com/ultralytics/yolov5)

**Ejecutarlo de forma local**:

1. hacer un *clone* del repositorio y ejecutarlo de forma local, el repo incluye una máquina Colab ([tutorial.ipynb](https://github.com/ultralytics/yolov5/blob/master/tutorial.ipynb)).

git clone https://github.com/ultralytics/yolov5

cd yolov5

pip install -r requirements.txt

1. De forma local se corre (sin entrenar) con el siguiente comando desde la consola:

python detect.py --source 0 # webcam o

python detect.py --source img.jpg # image o

python detect.py --source vid.mp4 # video

#NOTa: Los resultados estarán en el subdir: runs/detect

**Maquina Colab:** En el sitio anterior, hacer click en el siguiente enlace:

****

Al igual que en el caso anterior hay que instalar las dependencias. Pero desde python lo corremos de la siguiente forma:

import torch

# Model

model = torch.hub.load('ultralytics/yolov5', 'yolov5s')

# Images (cualquier path, local o URL a una imagen)

img = 'https://ultralytics.com/images/zidane.jpg'

# Inference

results = model(img)

# Results

results.print() # or .show(), .save(), .crop(), .pandas(), etc.

# **Yolov6**

Sitio oficial: [YOLOv6: a single-stage object detection framework dedicated to industrial applications.](https://github.com/meituan/YOLOv6)

**Ejecutarlo de forma local**:

1. hacer un *clone* del repositorio y ejecutarlo de forma local, el repo incluye una máquina Colab ([tutorial.ipynb](https://github.com/ultralytics/yolov5/blob/master/tutorial.ipynb)).

git clone https://github.com/meituan/YOLOv6

cd YOLOv6

pip install -r requirements.txt

1. De forma local se corre (sin entrenar) con el siguiente comando desde la consola:

python tools/infer.py --weights yolov6s.pt --source img.jpg

**Maquina Colab:**

En el sitio anterior hacer click aquí:



# **Entrenar Yolov5 (detección de mates)**

Con esta máquina de Colab podrán entrenar la red anterior (Yolov5) para que etiquete objetos nuevos, en este caso mates.

****

**Maquina Colab:**

<https://colab.research.google.com/drive/18ng3hBRZL87SDrLUTI6m6Q4UtqX6pHmf?usp=sharing>

**Repositorio de imagenes de Mates etiquetadas:** <https://github.com/75-70-Robots/mate-detection/tree/main/images/mates>

**Modelo entrenado:** mate\_pesos.pt

Lo encontrarán en la carpeta de recursos de Material-Alumnos en GoogleDrive

<https://drive.google.com/drive/folders/1xsQvHK92-zb_CXeh6iSdnZoLewAFKZ97?usp=share_link>

# 

# **Redes Recurrentes (Robo-Shakespeare)**

Esta máquina Colab permite construir un generador de texto shakesperiano. El modelo generará automáticamente cadenas cortas de texto

**Maquina Colab:**

<https://colab.research.google.com/drive/1DoG4YlEmN1-QV1trBPotF8802fdxoHrG?usp=sharing>

**Modelo entrenado:**

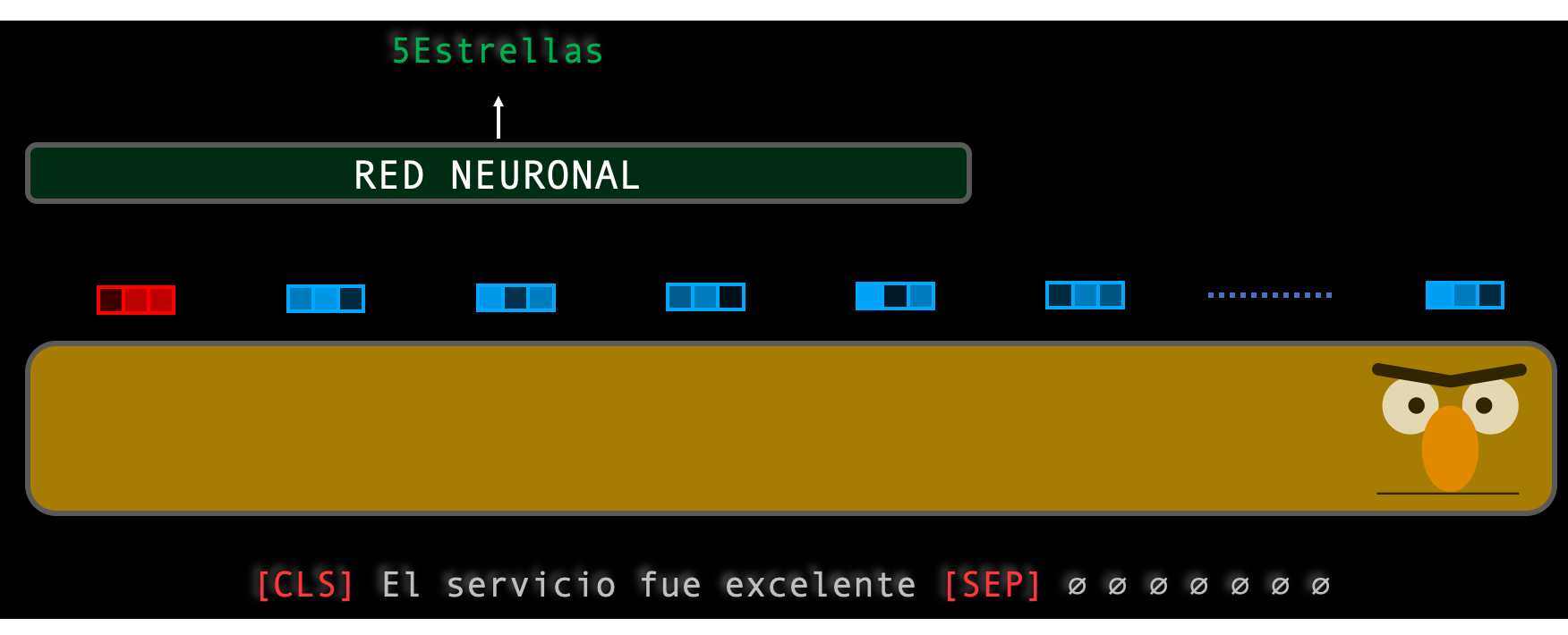
Lo encontrarán en la carpeta de recursos de Material-Alumnos en GoogleDrive (rnr\_shakespeare\_text\_generator.h5)

<https://drive.google.com/drive/folders/1xsQvHK92-zb_CXeh6iSdnZoLewAFKZ97?usp=share_link>

# 

# **BERT: análisis de sentimientos**

Con esta máquina de Google Colab, podrán entrenar un red neuronal que haga análisis de sentimientos, utilizando para ello un modelo lenguaje preentrenado llamado BERT.



**Maquina de Google Colab:**

<https://drive.google.com/file/d/1pDb7XUiHq7YznPudLHt0vx7RKE9b_mkN/view?usp=sharing>

**Conjunto de datos de entrenamiento:** BERT\_sentiment\_IMDB\_Dataset.csv.zip

Lo encontrarán en la carpeta de recursos de Material-Alumnos en GoogleDrive

<https://drive.google.com/drive/folders/1xsQvHK92-zb_CXeh6iSdnZoLewAFKZ97?usp=share_link>

**Modelo entrenado:** BERTSentimentClassifier.pt (para 2 epocas)

Lo encontrarán en la carpeta de recursos de Material-Alumnos en GoogleDrive

<https://drive.google.com/drive/folders/1xsQvHK92-zb_CXeh6iSdnZoLewAFKZ97?usp=share_link>

# **RoBERTuito**

Modelo de lenguaje similar a BERT pero entrenado para idioma español:

**Repositorio oficial:** <https://github.com/pysentimiento/robertuito>

**Análisis de sentimiento on-line con el modelo anterior:**

<https://huggingface.co/pysentimiento/robertuito-sentiment-analysis?text=Esperando+la+Carroza+es+una+obra+maestra%2C+no+hay+un+solo+momento+en+el+que+no+me+encuentre+riendo.+Es+una+imagen+cruda+de+nuestra+cultura>.

# **Stable Diffusion**

Stable diffusion es un modelo pre-entrenado (un pipeline entero en verdad), que permite generar imágenes a partir de texto. También imágenes complejas a partir de imágenes simples.

|  |  |
| --- | --- |

**Herramienta de edición Web (para probarlo):** <https://beta.dreamstudio.ai/dream>

Es paga, pero se puede usar gratis hasta que se acaben los créditos de regalo.

**Repositorio oficial:** <https://github.com/CompVis/stable-diffusion>

**Maquina Colab**:

<https://colab.research.google.com/github/huggingface/notebooks/blob/main/diffusers/stable_diffusion.ipynb>