

Elegí la moneda ethereum para analizar

Primero que todo para elegir qué api iba a usar empecé con la que me pareció más fácil de manejar, la cual fue google, ya que este es posible obtener los links de las noticias recientes sin la necesidad de hacer scraping en la página. Sin embargo con las otras 2 páginas elegidas, bing y cointelegraph, tuve que utilizar la herramienta BeautifulSoup para scrapear las noticias más recientes y obtener el link de cada una para poder acceder a la página de donde voy a conseguir el texto. Luego creé una función genérica para scrapear los párrafos de la noticia y así poder utilizarlos luego.

```
import requests
from textblob import TextBlob
from bs4 import BeautifulSoup

def scrape_url(url : str):
    polaridadTot = 0
    texto_com = ""
    response = requests.get(url)
    soup = BeautifulSoup(response.content, "html.parser")
    paragraphs = soup.find_all('p')
    titulo = soup.title.text.strip()
    texto_com = " ".join(paragraph.text for paragraph in paragraphs)
    return {"title": titulo, "body": texto_com, "url": url}
```

Después de obtener todos los párrafos de todas las noticias, los junte todos en una variable con el .join y empecé a analizar el texto:

Por un lado separe las palabras y no palabras con un word\_tokenize, luego filtre las palabras por sentimiento utilizando la librería text blob, la cual me permite definir el sentimiento de cada palabra, por lo que deseché las palabras sin sentimientos. Y por último conte la cantidad de cada tipo de

palabra y lo junte junto a su palabra para obtener el frequency y luego ordenar la lista por este numero(al final solo agarre las primeras 10 como principales).

Por otro lado utilizando text blob calculo el sentimiento del texto Total, obtengo el timestamp utilizando la libreria time y consigo sus ner con la librería spacy.

Obteniendo como resultado todos los datos necesarios para luego meterlos en la base de datos, la cual es un csv creado en el colab

```
listaTotal = scrape_bing() + scrape_google() + scrape_cointelegraph()
listaPalabras = []
textoTotal = " ".join(noticia["body"] for noticia in listaTotal)
for pagina in listaTotal:
    listaPalabras = listaPalabras + word_tokenize(textoTotal.lower())
listaFiltrada = filtrarPorPolaridad(listaPalabras)
palPrin = palabrasPrin(listaFiltrada,10)
sentimiento = sentimientoTexto(textoTotal)
timeStamp = time_Stamp()
palabrasNer = devolverNer(textoTotal)
cargarBaseDeDatos(sentimiento,timeStamp,palPrin,palabrasNer)
```

## Bibliografía

*Base de Datos.* (s/f). <https://www.delftstack.com/es/howto/python/python-append-to-csv/>

*Google.* (s/f). <https://pypi.org/project/googlesearch-python/>

*Ner.* (s/f). <https://www.wisecube.ai/blog/named-entity-recognition-ner-with-python/>

*Request.* (s/f). <https://pypi.org/project/requests/>

*TextBlob.* (s/f). <https://pypi.org/project/textblob/>

*Time Stamp.* (s/f). <https://flexiple.com/python/python-timestamp>