# 

# **Ingeniería de Sistemas de Información**

# Procesamiento del Lenguaje Natural

# 

Trabajo Práctico Individual

Detección de Emociones y Recuperación de la Información (Information Retrieval)

Daniel Agustín Brosio - 168.362-7  
02/07/2021

## Índice

[Descripcion General](#_b6btw5i7o604)

[Keywords](#_xvk3k5p6sp8p)

[Diagrama del Sistema](#_1cdv6jom0ajz)

[Information Retrieval](#_comglbc0nows)

[Redes sociales](#_mhj8l72o3c3j)

[Noticias](#_8yv0zp37d7cn)

[Mercado](#_5y4v9glofcvf)

[Sentiment Analysis](#_7ys0x9z2227r)

[Acerca del modelo](#_8rjvb52zpq8q)

[Parámetros](#_cxzieo92ubms)

[Resultados](#_2cy7jwhbv8gd)

[Tabla de puntajes](#_gfdglhrpxt22)

[Conclusión](#_8p7tjc8ukx9g)

## 

## Descripcion General

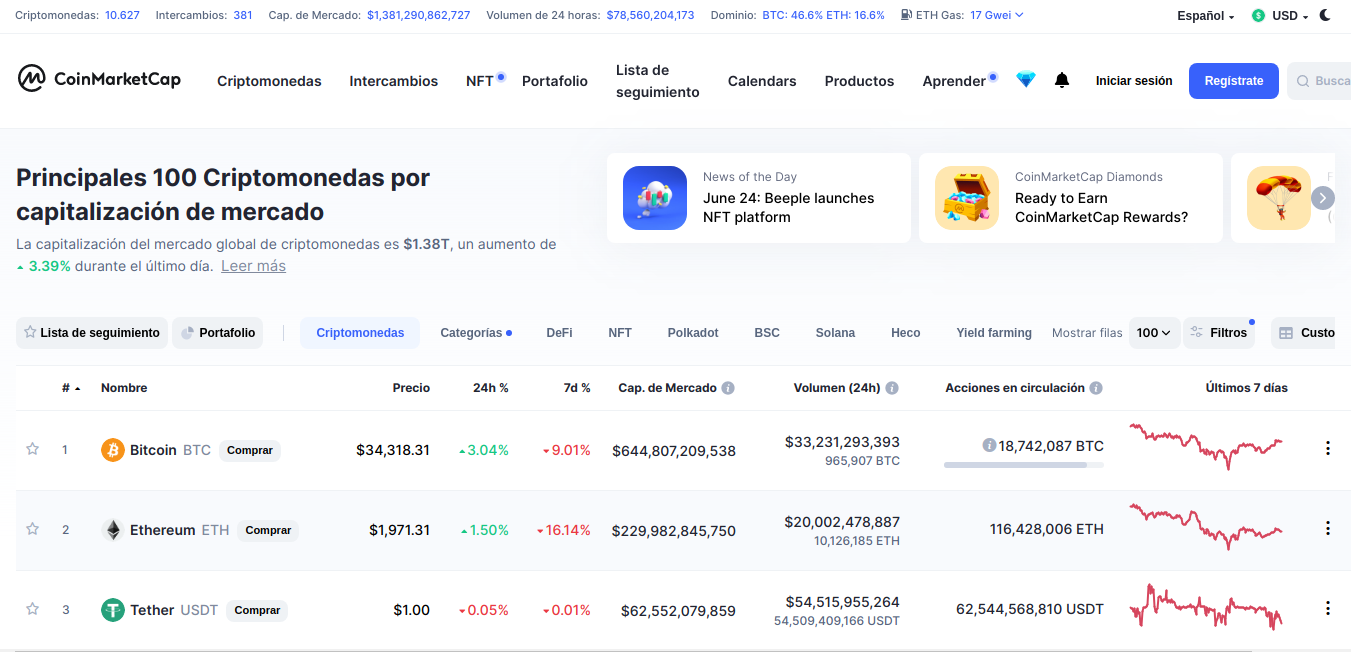
Una de las aplicaciones más frecuentes e interesantes del Procesamiento de Lenguaje Natural es el **análisis de sentimientos**, sea para analizar reviews masivamente o para generar respuestas más adecuadas en un chatbot. Si además lo combinamos con información provista de redes sociales las posibilidades se multiplican.

El objetivo de este trabajo es intentar predecir la suba o baja de precio de 20 criptomonedas a partir de información recopilada de la web y aplicando lo aprendido en la cátedra de Procesamiento de Lenguaje Natural.

Se desarrolló una aplicación en **NodeJS** que realiza IR (**Information Retrieval** o Búsqueda y Recuperación de Información) de distintas fuentes tanto redes sociales como también sitios de noticias para tener un panorama más amplio. Luego se aplica un modelo predictivo proporcionado por la librería **Tensor Flow JS** el cual califica el texto en el rango de 0-1 (0 = sentimiento negativo; 1 = sentimiento positivo). Todo este `batch` de datos se almacena en la base de datos en **MongoDB** para su posterior utilización.

Aparte se realizó otra aplicación Node JS que recopila los datos de la base y genera un archivo .csv para mostrar los resultados tanto del análisis como del desarrollo del valor de la criptomoneda en cuestión. De esta manera se puede analizar si existe correlación (o no) entre el intento de predicción y la suba/baja de precio.

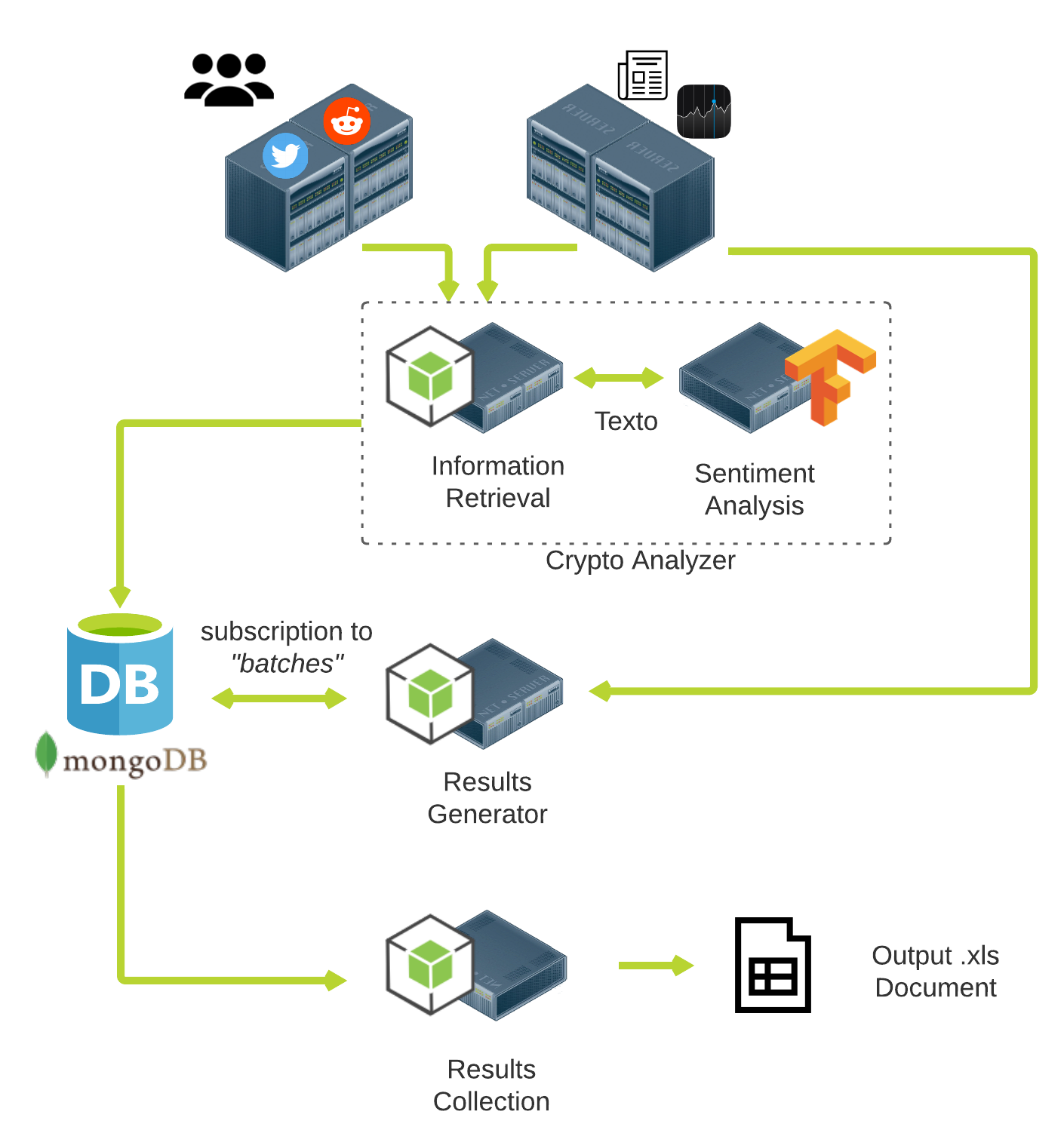
¿Qué monedas? Las principales según [coinmarketcap.com](https://coinmarketcap.com/es/) a la hora de seleccionarlas.



### Keywords

|  |  |
| --- | --- |
| 0: Bitcoin 1: Ethereum 2: Tether 3: Cardano 4: Binance Coin 5: XRP 6: Dogecoin 7: USD Coin 8: Polkadot 9: Uniswap | 10: Internet Computer 11: Bitcoin Cash 12: Chainlink 13: Litecoin 14: Polygon 15: Stellar 16: Binance USD 17: Solana 18: Ethereum classic 19: VeChain |

## Diagrama del Sistema



## Information Retrieval

### Redes sociales

* **Twitter:**

Se utiliza la librería [twitter-v2](https://github.com/HunterLarco/twitter-v2#readme) como wrapper de la API oficial de twitter. La misma es utilizada para obtener, como máximo 1000, tweets recientes (últimos 30 minutos), en inglés y que contengan el nombre de la criptomoneda o su correspondiente cashtag.

* **Reddit**

Usando [snoowrap](https://github.com/not-an-aardvark/snoowrap) se obtienen comentarios recientes de los subreddits de cada criptomoneda. En este caso se optó por esta estrategia porque subreddits centrales como ‘r/wallstreetbets’ rara vez mencionan a las monedas menos populares.

### Noticias

Debido a que los principales sitios de noticias (nyt, bloomberg, wsj, ft, etc…) contaban con protección anti scraping, fue necesario seleccionar una fuente menos popular, en este caso utilizamos `coinpedia`. Este proceso, a su vez consta de dos etapas:

* **Bing’s custom search**

Se utiliza [customsearch.ai](https://www.customsearch.ai/) de bing para buscar los resultados de noticias del día limitando la búsqueda a <https://coinpedia.org/>. Esto arroja varios resultados de noticias, pero desafortunadamente el `snippet` o `preview` de la misma no es suficiente como para realizar un correcto análisis. Para eso es la siguiente etapa.

* **Scraping**

Una vez obtenido el resultado se navega a la nota y se scrapping sus títulos y párrafos con la ayuda de la librería [puppeteer](https://pptr.dev/).

### Mercado

Por último se obtienen los resultados de valores de criptomonedas de la API de [CryptoCompare](https://www.cryptocompare.com/). Esto se realiza una vez al realizar el análisis de sentimiento y una vez más al generar los resultados, 30 minutos después de la búsqueda de información.

## 

## Sentiment Analysis

La función que realiza el análisis de sentimientos es `performSentimentAnalysis` la cual se encarga del seteo y configuración del `SentimentPredictor` basado en el [ejemplo](https://github.com/tensorflow/tfjs-examples/tree/master/sentiment) proporcionado en el repositorio de Tensor Flow JS.

También fue de gran ayuda el siguiente [blog post](https://bensonruan.com/twitter-sentiment-analysis-with-tensorflowjs/) de Benson Ruan.

### Acerca del modelo

Como explica Benson Ruan, el modelo está basado en reviews de películas, series, shows, etc. provenientes de IMDB.

*“Two model variants are provided (CNN and LSTM). These were trained on a set of 25,000 movie reviews from IMDB, labeled as having positive or negative sentiment. This dataset is* [*provided by Python Keras*](https://keras.io/datasets/#imdb-movie-reviews-sentiment-classification)*, and the models were trained in Keras as well, based on the* [*imdb\_cnn*](https://github.com/keras-team/keras/blob/master/examples/imdb_cnn.py) *and* [*imdb\_lstm*](https://github.com/keras-team/keras/blob/master/examples/imdb_lstm.py) *examples.”*

Fuente: <https://github.com/tensorflow/tfjs-examples/tree/master/sentiment#tensorflowjs-example-sentiment-analysis>

### Parámetros

Un parámetro configurable son los umbrales de clasificación de sentimientos. El sistema arroja un puntaje de 0 a 1 y arbitrariamente decidí que la clasificación sea -

* “POSITIVE”: Si el puntaje es mayor que 0.75
* “NEGATIVE”: Si el puntaje es menor que 0.25
* “NEUTRAL”: Cualquier otro caso

## Resultados

Considero un “*win*” haber predecido que el valor subirá o bajará y haber acertado. Un “*loose*” es lo contrario. De esta manera se observa que la predicción no ofrece mucha confiabilidad a la hora de predecir los movimientos del mercado.  
  
Si el programa corriese durante varios días se podría obtener mas claridad sobre la precisión del sistema. Esto requeriría deployar el proyecto en un servidor, pero eso sería el proximo paso.

## Tabla de puntajes

|  |  |  |  |
| --- | --- | --- | --- |
| **Compilation** | **wins** | **losses** | **score** |
| BTC | 93 | 251 | -158 |
| ETH | 16 | 45 | -29 |
| USDT | 51 | 5 | 46 |
| ADA | 46 | 10 | 36 |
| BNB | 30 | 23 | 7 |
| XRP | 48 | 3 | 45 |
| DOGE | 41 | 5 | 36 |
| USDC | 20 | 24 | -4 |
| DOT | 37 | 3 | 34 |
| UNI | 34 | 5 | 29 |
| ICP | 28 | 5 | 23 |
| BCH | 2 | 9 | -7 |

## Conclusión

La cantidad de resultados obtenidos no es suficiente como para tener claro si el sistema está prediciendo correctamente o es simplemente azar. Para obtener mas resultados estos programas deben quedar corriendo durante períodos largos de tiempo, lo cual requiere que el proyecto esté hosteado en un servidor.  
  
Sin embargo, este proyecto fue muy útil como prueba de concepto y fundamentalmente fue positivo para ganar experiencia en el manejo de APIs, tecnicas de scraping y herramientas de AI.