

**Proyecto 1:** Twitter Bot (@taggeame)

[https://github.com/tominaked1337/tagme\\_twitterBot](https://github.com/tominaked1337/tagme_twitterBot)

**Alumno:** Tomás Perez

**Materia:** Integración Tecnológica

**Año lectivo:** 2021



Profesorado  
Don Bosco

## Introducción

Este proyecto consta del desarrollo de un bot abocado a procesar la información que nos provee la API de Twitter. Consiste en utilizar los distintos métodos que la API contiene para analizar el contenido de un tweet, ya sea un evento disparado a través de una mención al bot (@taggeame en twitter), o al realizar una descarga de tweets de uno o más usuarios, a fines de analizar la existencia de palabras clave y en base a eso accionar según se desee.

El bot está escrito en Python 3.8, con la implementación de tweepy, pymongo y requests

## Funcionalidades

Los comandos que actualmente el bot comprende son:

- -usd  
Responde al usuario con la cotización para compra y venta del dólar americano al cambio en pesos en tiempo real.



- -btc

Responde al usuario la cotización del bitcoin al cambio en dólar americano en tiempo real.



The image is a screenshot of a Twitter thread on a dark background. The first tweet is from user **neiqued** (@toomiperez\_) dated May 4, replying to **@taggeame** with the text "-btc". It has 1 reply, 0 retweets, and 0 likes. The second tweet is from user **Botinho** (@taggeame) dated May 4, replying to the first tweet. The text of the second tweet is: "Cotización al 04/05/2021 02:02:40", "1 BTC = USD 55,878.09", and "Fuente". Below the second tweet is a large advertisement for CoinDesk 20. The ad features the CoinDesk logo, a Bitcoin symbol, and the text "Real-time BTC price, 24-hour charts and news." Below this, it says "Bitcoin Price Index — CoinDesk 20" and "The CoinDesk 20 provides real-time prices, news, videos and educational content on the digital assets moving the crypto market, from BTC to ...". At the bottom of the ad is a link to "coindesk.com". The Twitter interface at the bottom shows icons for replies, retweets, likes, and a share button.

**neiqued** @toomiperez\_ · May 4  
@taggeame -btc

1

**Botinho** @taggeame · May 4  
Cotización al 04/05/2021 02:02:40  
1 BTC = USD 55,878.09  
Fuente

**coindesk**® 20

**Real-time BTC price, 24-hour charts and news.**

Bitcoin Price Index — CoinDesk 20  
The CoinDesk 20 provides real-time prices, news, videos and educational content on the digital assets moving the crypto market, from BTC to ...  
[coindesk.com](https://coindesk.com)

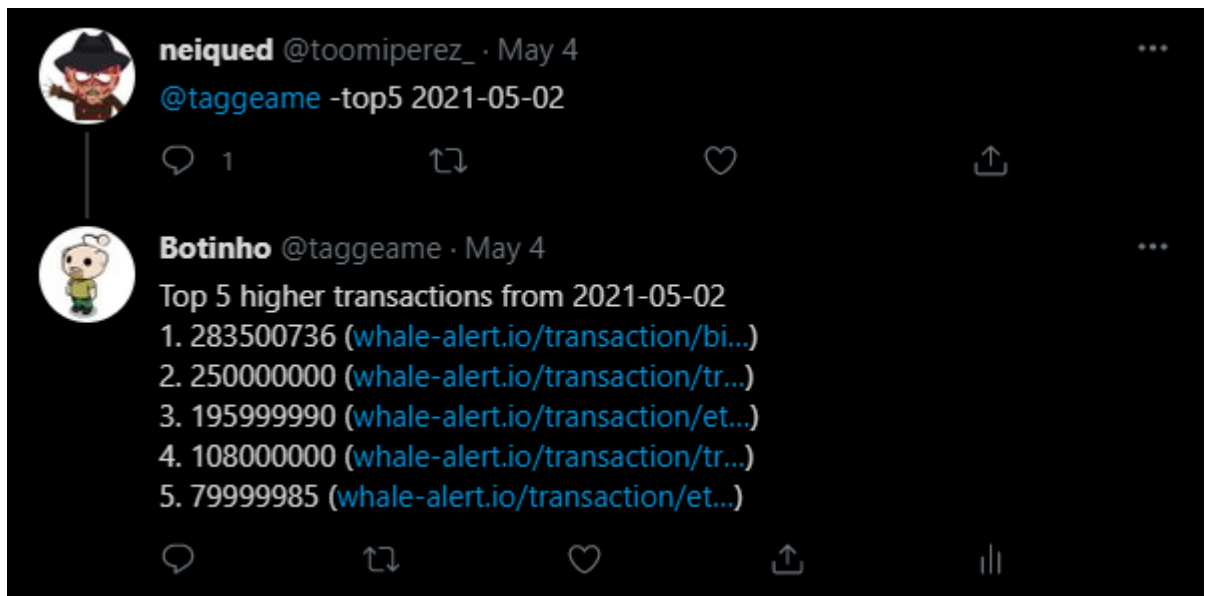
- -gif (palabra/s)

Responde al usuario un gif relacionado a la palabra indicada después del comando



- -top5 (aaaa-mm-dd)

Responde al usuario un top de las 5 transacciones más altas de la fecha ingresada, ordenadas de mayor a menor, con los detalles de cada una provistos por <https://whale-alert.io/>



## Implementación de la API de Twitter

<https://developer.twitter.com/en/docs/twitter-api>

Debe realizarse una postulación a la misma, en donde se solicitan múltiples datos acerca de la finalidad del proyecto, del desarrollador y del uso que se le va a dar a la API.

Es importante considerar los **límites de requests** que se le harán a la API, ya sea cantidad de tweets a responder por segundo, descargas de tweets por segundo, etc. No cumplir con estas condiciones puede ser motivo de **penalización** del proyecto **restringiendo nuestro acceso a la API** y **declinación de las credenciales (API Keys)**

### Autenticación de credenciales

```
bot.py x keys.py x
1 CONSUMER_KEY = '****your key****'
2 CONSUMER_SECRET = '****your key****'
3 ACCESS_KEY = '****your key****'
4 ACCESS_SECRET = '****your key****'
```

```
#Auth With Tweepy
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
api = tweepy.API(auth, wait_on_rate_limit=True)
```

## Descripción de los comandos

Dentro del módulo principal “bot.py” existen dos funciones principales: `reply_mentions()` y `find_keywords()`. Ambas están condicionadas por el id de los tweets, quién será el indicador principal para trabajar. Para esto, existen otras dos funciones, `retrieve_last_seen_id(file_name)` y `store_last_seen_id(last_seen_id, file_name)`, cuyo labor es guardar y leer el id de cada tweet que esté siendo procesado para evitar analizar un tweet dos veces, no dar respuesta duplicadas, etc. **Se debe tener en cuenta que el ID de los tweets es incremental**, de modo que si evaluamos `if mention to reply.id > last seen id:` garantizamos responder menciones nuevas y no duplicar respuestas.

Básicamente, **`reply_mentions()`** se encarga de estar a la espera de nuevas menciones y aplicando un análisis del texto contenido en el tweet.

Para el comando **-usd** se realiza un scrapping a <https://www.dolarhoy.com/> para obtener cada vez que se menciones el comando el precio actualizado del dólar. Lo mismo se hace para el comando **-btc** solo que se efectua scrapping a <https://www.coindesk.com/price/bitcoin>.

En cuanto al comando **-top5** se implementa el uso de la función **`find_keywords()`**, para descargar todos los tweets del usuario @whale\_alert (notifica todas las transacciones de criptomonedas superiores a 1MM USD, entre otras cosas) que contengan la keyword ‘transferred’ para detectar que se trata de una transacción y no un minado u otro tipo de operación. Se guardan todos los datos útiles proporcionados por los métodos del objeto `Status()` (el tweet en concreto) en una base de datos MongoDB. Una vez que el bot detecta el comando **-top5** va a consultar por todas las transacciones de la fecha indicada, y devolverá las 5 más altas, junto con un enlace al detalle que provee whale-alert.io.

Por último, el comando **-gif** va a estar utilizando el módulo `search_gif.py`, el cual implementa la API de Tenor (<https://tenor.com/gifapi/documentation>) para realizar una query en base a la palabra ingresada siguiente al comando. Esto devuelve un json, dentro del cual tendremos múltiples formatos de gif (**tener en cuenta que Twitter gifs mayores a 3MB**) y su enlace directo para visualizar. Lo siguiente será hacer un request a dicho enlace para obtener el gif como archivo, guardarlo y finalmente devolverlo como adjunto en la respuesta al tweet (una vez efectuada la respuesta el archivo se elimina). Cabe destacar que Tenor otorga una API Key pública de libre uso, por lo que no fue necesario registrarse ni aplicar al uso de ella

(esto fue un gran alivio ya que se pierde mucho tiempo aplicando y aguardando a tener la aprobación).