**Detalles técnicos para la descarga**

1. **Introducción**

El presente documento explica los detalles técnicos realizados para la extracción de datos.

1. **API utilizada**

Para el procedimiento de descarga de datos de Twitter, se contaba con el API de Twitter. Esta incluia una variedad de servicios web disponibles para el usuario, el cual lograba tener a su disposición para poder acceder a los datos de usuarios, tweets, etc. Sin embargo, el uso de esta API nos limitaba a tener que acceder como máximo a 100 tweets por usuario o hashtag. Por esta razón, se decidió utilizar el API Tweepy, la cual nos permitía acceder más de 3000 tweets.

La API Tweety es totalmente gratuita y funciona con el leguaje de programación Python. Para ello es necesaria la instalación de la librería de Tweepy, la cual se puede encontrar gratuitamente en GitHub.

Fuente: <https://github.com/tweepy/tweepy>

A pesar de la gran cantidad de tweets que se podrían sacar, la limitación se encuentra en que la máxima cantidad de peticiones que se pueden realizar son 100 por hora. Por lo tanto, se podrían analizar un máximo de 100 usuarios/hashtags cada hora.

Ante esta limitación existen otras herramientas adquiridas por Twitter, las cuales están orientadas a empresas y cuenta con un acceso ilimitado. Sin embargo, el precio es desconocido ya que se necesita hacer contacto directo con la empresa.

Fuente: <http://gnip.com/>

1. **Medio de Comunicación**

Como mencionamos en el punto anterior, la API de Twitter (llamémosla, la API original de Twitter para poder diferenciarla de Tweepy) cuenta con una cantidad de servicios web disponibles para los usuarios. La entrega de datos de la API original de Twitter la realizaba a través de un JSON. Sin embargo, la herramienta Tweepy maneja sus propias clases, las cuales tendrán que ser parseadas manualmente para poder ser trabajadas con JSON o cualquier otro tipo de dato.

Fuente: <http://tweepy.readthedocs.org/en/v2.3.0/>

Durante el proceso de la API, se realizó la creación de clases con los atributos que se establecieron como los necesarios para el proyecto. Para mayor información, se puede revisar el “Documento con especificación de retorno de datos por API” para Twitter.

Las clases propias que se crearon fueron posteriormente convertidas a JSON con el uso de las librerias “json” y “simplejson”, las cuales están incluidas durante la instalación del Python.

Por otro lado, los usuarios/hashtags de los cuales se debía realizar la extracción de tweets se encuentran almacenados en un fichero .csv. La estructura de ese archivo se defina a continuación:

Archivo: twt\_input\_token.txt

@usuario1;#hashtag11,#hashtag12

@usuario2;#hashtag21,#hashtag22,#hashtag23

…

1. **Tipos de datos entregados**

Posterior de realizada la extracción de datos, los tweets extraídos deberán de ser purificados. Estos criterios serán explicado posteriormente en el “Documento de detalles técnicos del porque (o no) de los registros duplicados”. Sin embargo, previo a la purificación los datos deberán ser almacenados en un fichero/base de datos. Por lo tanto, la purificación se realizará en base al fichero determinado.

Los tweets extraídos fueron almacenados en un fichero .txt. Los datos se encuentran en el formato JSON. Dicho fichero cuenta con la siguiente estructura

Archivo: twt\_output\_fecha.txt

@usuario1

json\_del\_tweet\_extraido1

json\_del\_tweet\_extraido2

…

#hashtag11

json\_del\_tweet\_extraido1

json\_del\_tweet\_extraido2

…

#hashtag12

json\_del\_tweet\_extraido1

json\_del\_tweet\_extraido2

…

@usuario2

…

#...

NO habrán líneas en blanco por ningún motivo. Cada dato explicado anteriormente ocupará una línea (usuario, hashtag o json).

Finalmente, se procederá a almacenar el json en un fichero CSV. Este tendrá la siguiente estructura:

Header (Estructura del CSV)

@Usuario1

csv\_del\_tweet\_extraido1

csv\_del\_tweet\_extraido2

#Hastag1

csv\_del\_tweet\_extraido1

csv\_del\_tweet\_extraido2

…

Para más detalle, ver los archivos Json\_twitter y Json\_csv

1. **Conclusión**

Se ha logrado definir el medio de comunicación con la API y con los datos necesarios por el proyecto. Finalmente, se logró obtener los ficheros los cuales muestran elresultado del flujo completo con los datos obtenidos.