**Detalles técnicos para la descarga**

1. **Introducción**

El presente documento explica los detalles técnicos utilizados para la descarga

1. **API utilizada**

Para el procedimiento de descarga se utilizó el API de python para Instagram, la cual está publicada en la página de desarrolladores de Instagram y en GitHub. Es totalmente gratuita.

Fuente: <http://instagram.com/developer/libraries/>

Fuente: <https://github.com/Instagram/python-instagram>

La cantidad limitada de posts es 20 por cada petición. Se utilizó sólo 2 servicios web mediante la API. Uno para obtener los posts relacionados a un tag y otro para obtener los posts relacionados a un usuario. Para poder obtener una mayor cantidad de tweets se tiene que realizar otra petición de tal forma que se obtengan los 20 siguientes y así continuamente. En cada JSON de respuesta de la API se obtine una variable la cual ayuda a realizar la paginación de posts.

Nuevamente, al igual que las APIs anteriores, la limitación es la cantidad de peticiones, la cual es 5000 por hora.

1. **Medio de Comunicación**

La entreda de datos se realiza a través de un JSON. Sin embargo, la herramienta Instagram de python 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://pastebin.com/FtZ8MZ8f>

A diferencia de la API de Twitter, en esta API se procedió a modificar la librería python de instagram, de tal forma que se logre obtener el json original sin necesidad de ser parseado a clases de manera manual.

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

Id\_usuario\_1;#hashtag11,#hashtag12

Id\_usuario2;#hashtag21,#hashtag22,#hashtag23

…

Se recomienda el uso de Ids\_de\_usuario para la extración de datos. De esta manera, se ahorrarían peticiones.

1. **Tipos de datos entregados**

Posterior de realizada la extracción de datos, los posts 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 posts 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

@id\_del\_usuario

json\_del\_post\_extraido1

json\_del\_post\_extraido2

…

#hashtag11

json\_del\_post\_extraido1

json\_del\_post\_extraido2

…

#hashtag12

json\_del\_post\_extraido1

json\_del\_post\_extraido2

…

@id\_del\_usuario

…

#...

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)

@id\_del\_usuario1

csv\_del\_post\_extraido1

csv\_del\_post\_extraido2

#Hastag1

csv\_del\_post\_extraido1

csv\_del\_post\_extraido2

…

Para mayor detalle de la estructura del JSON y CSV, puede ver los archivos Instagram\_Json.txt y Instagram\_csv.txt

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 el resultado del flujo completo con los datos obtenidos.