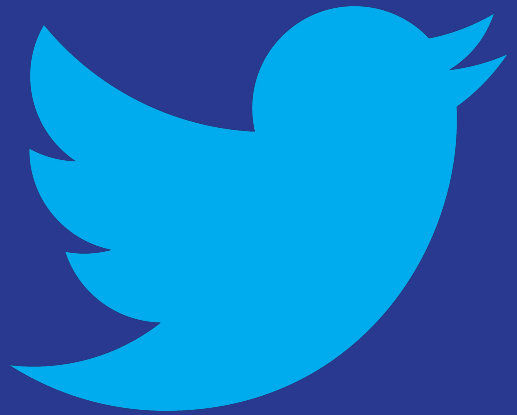
TWEEPY

Cuando Python escucha lo que dice un pajarito...



¿Quienes somos?

Nieves Borrero{

Licenciada en Hª del arte,

Estudiante 2º D.A.W,

Organizadora GDG Córdoba

}

@nievesborrero

@PabloLeonPsi

Pablo León{

Graduado en Psicología,

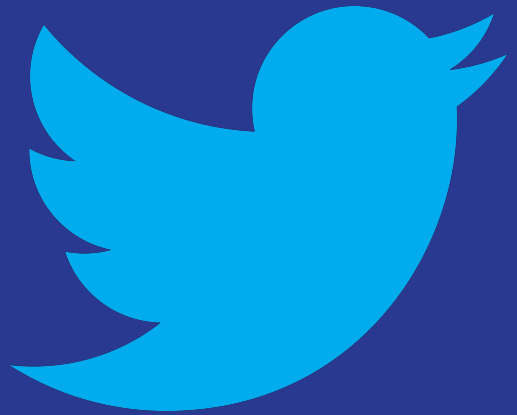
Estudiante 2º D.A.W,

Organizador GDG Córdoba

}

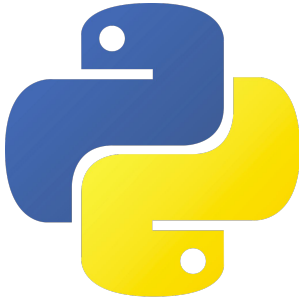
TWEEPY

Cuando Python escucha lo que dice un pajarito...



Tweepy

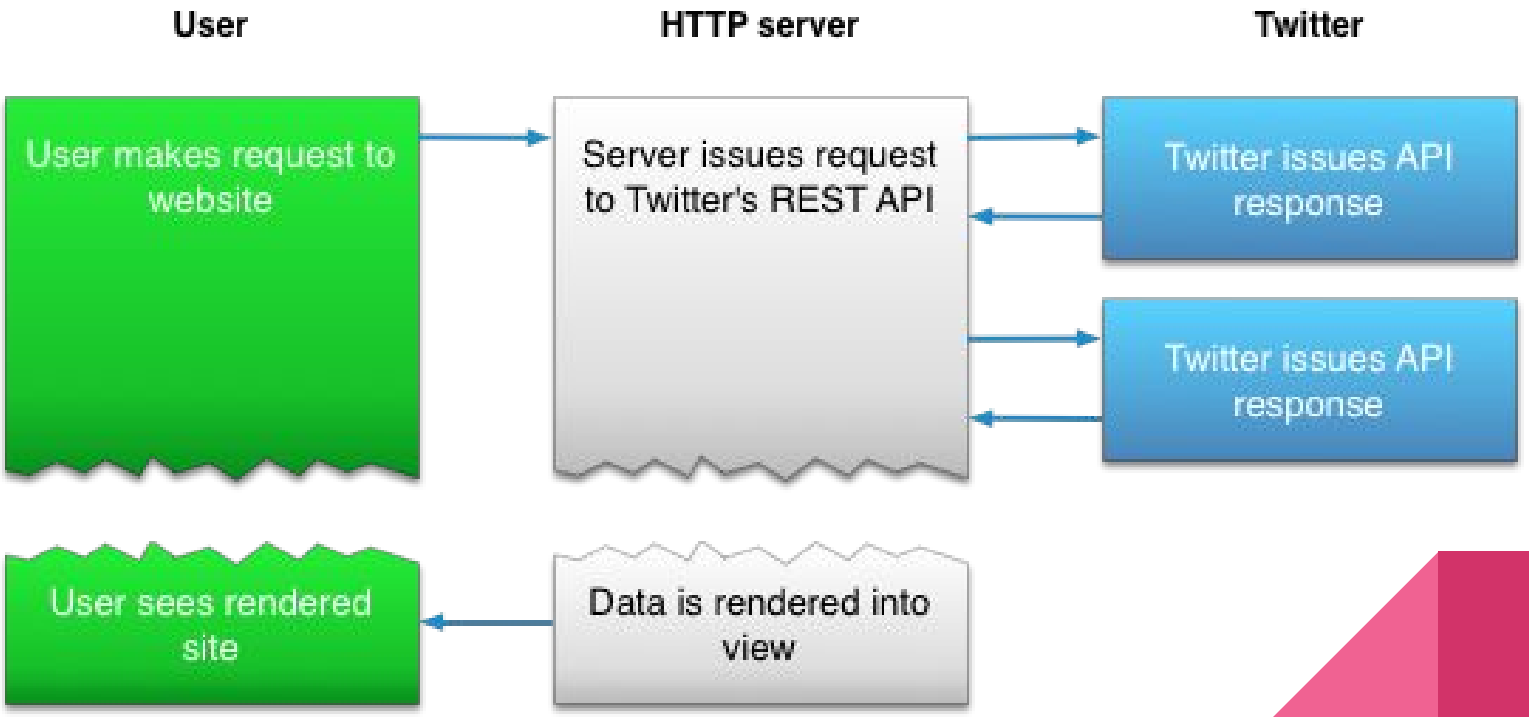
open source



- **Fácil integración con aplicaciones webs**
- **Fácil integración con bbdd**
- **Lenguaje sencillo y con corta curva de aprendizaje**
- **Gran número de librerías**

http://tweepy.readthedocs.io/en/v3.5.0/

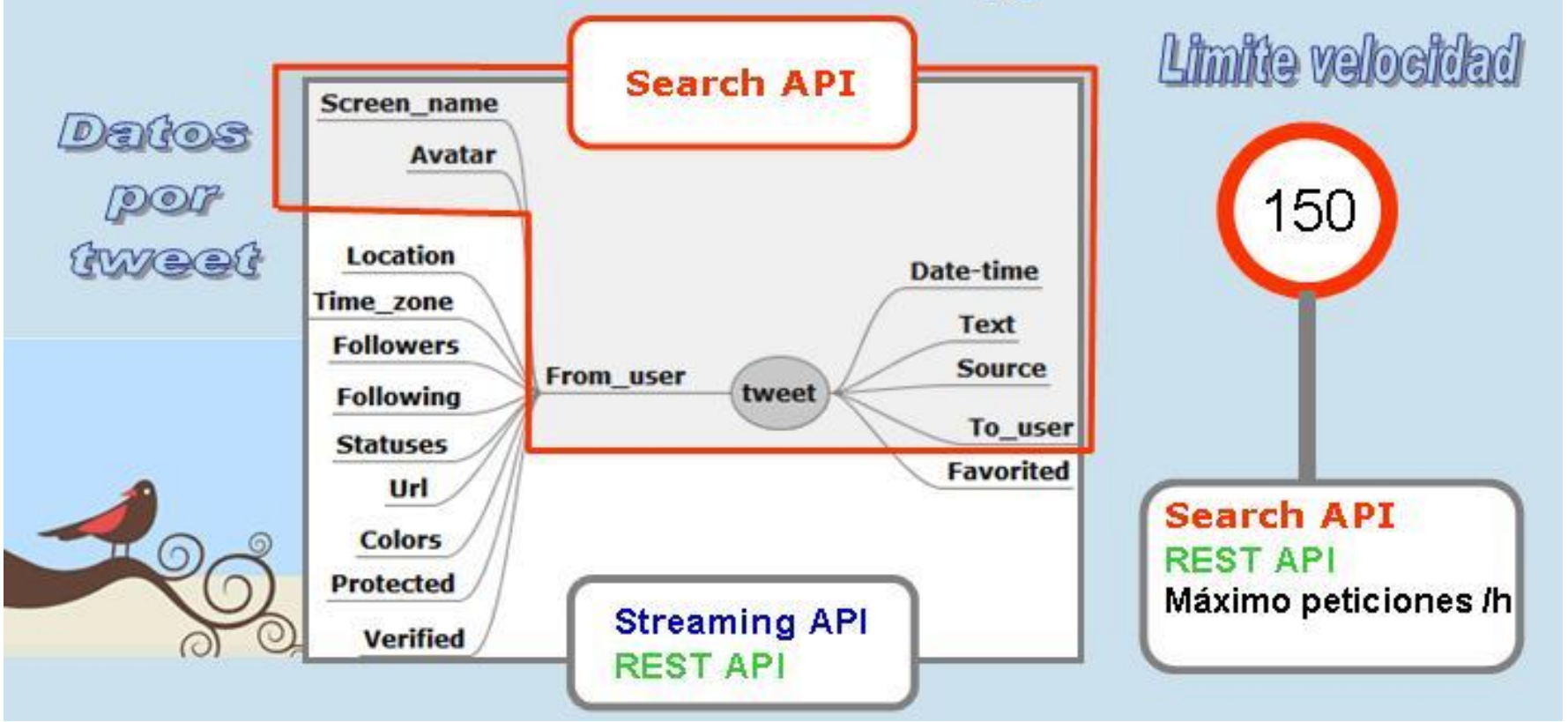
¿Cómo funciona?



Las APIs de Twitter

Ads API	para gestionar publicidad
Search tweets	nueva acepción del Search API
Filter realtime tweets	antes denominada Streaming API
Direct Message	API de nueva creación

https://developer.twitter.com/

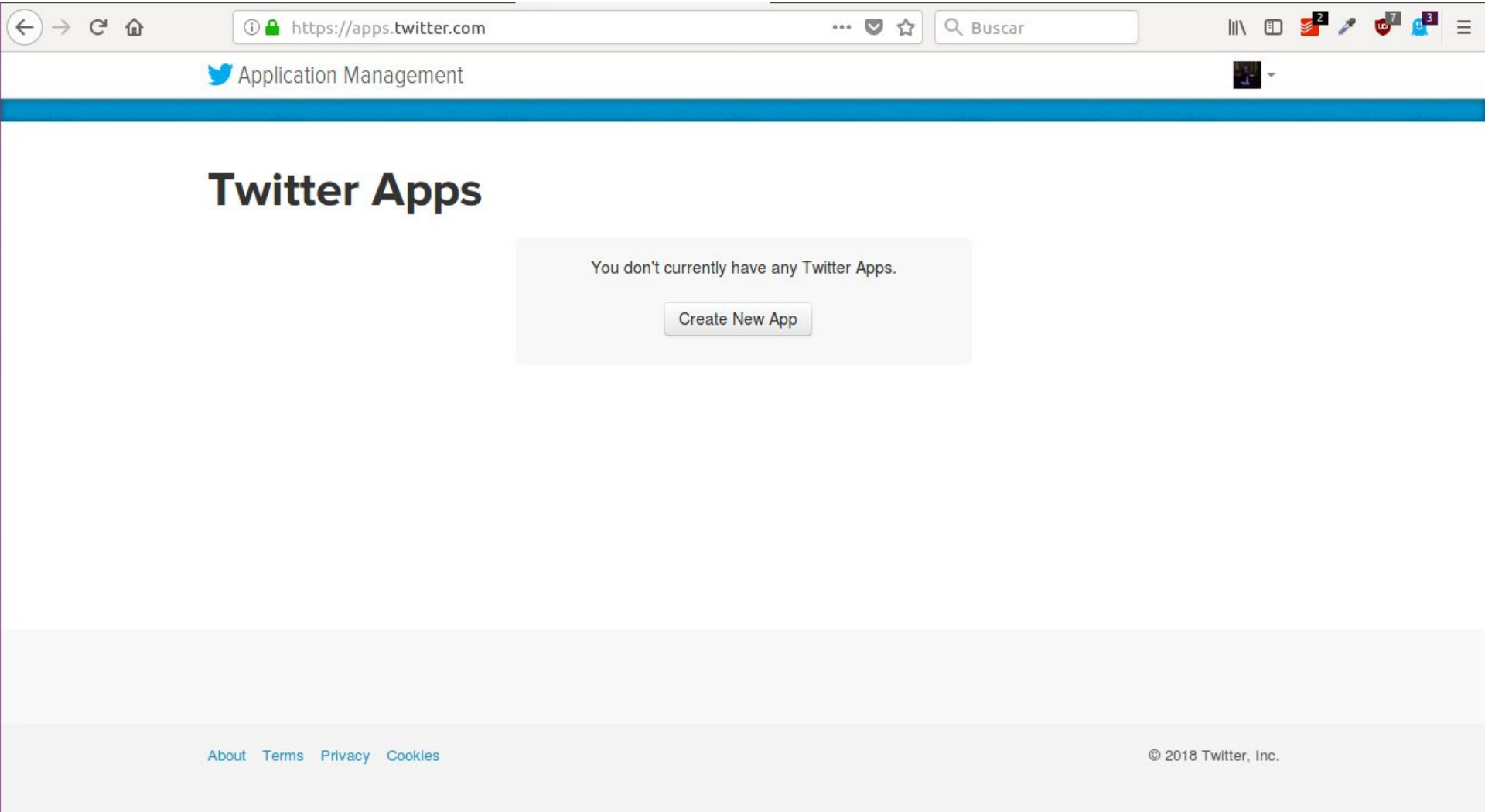


www.barriblog.com

zación
Retweets
Unfollow

Creando una aplicación de Twitter

https://apps.twitter.com/


https://apps.twitter.com
Buscar
Application Management
Twitter Apps
You don't currently have any Twitter Apps.
Create New App
About
Terms
Privacy
Cookies
© 2018 Twitter, Inc.

Create an application

Application Details

Name *

sysmana_app

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

proyecto para la sysmana 2018

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

www.google.com

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribut
tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To res
application from using callbacks, leave this field blank.

sysmana_app

Test OAuth

Details | Settings | Keys and Access Tokens | Permissions

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	D3jfsZ7 ... 03
Consumer Secret (API Secret)	ReBvmlNbqmu ... Alz
Access Level	Read and write (modify app permissions)
Owner	PabloLeonPsi
Owner ID	805349166

Application Actions

Regenerate Consumer Key and Secret | Change App Permissions

Your Access Token

Application Actions

Regenerate Consumer Key and Secret | Change App Permissions

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	8C … 3- nfIYFS0S[redacted]5vwXdT37WVmL92tc0jft2
Access Token Secret	zG… ZxDm[redacted]mU86zEVJaJYuepOSC
Access Level	Read and write
Owner	[redacted]
Owner ID	66

Token Actions

Regenerate My Access Token and Token Secret | Revoke Token Access

Instalando la librería Tweepy

sudo python get-pip.py

sudo pip install tweepy

```
root@Lenovo:/home# pip install tweepy
Collecting tweepy
  Downloading tweepy-3.5.0-py2.py3-none-any.whl
Requirement already satisfied: six>=1.7.3 in /usr/lib/python2.7/dist-packages (f
rom tweepy)
Collecting requests-oauthlib>=0.4.1 (from tweepy)
  Downloading requests_oauthlib-0.8.0-py2.py3-none-any.whl
Collecting requests>=2.4.3 (from tweepy)
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
    100% |                                        | 92kB 2.5MB/s
Collecting oauthlib>=0.6.2 (from requests-oauthlib>=0.4.1->tweepy)
  Downloading oauthlib-2.0.6.tar.gz (127kB)
    100% |                                        | 133kB 3.4MB/s
Requirement already satisfied: idna<2.7,>=2.5 in /usr/lib/python2.7/dist-package
s (from requests>=2.4.3->tweepy)
Collecting urllib3<1.23,>=1.21.1 (from requests>=2.4.3->tweepy)
```

streamSearch.py

https://github.com/tweepy/tweepy/tree/master/tweepy

https://github.com/tweepy-sysmana/sysmana2018/blob/master/estructura_status.json

```
#
# Escucha en tiempo real y almacena los tweets en un fichero de texto
#
import tweepy
import codecs
from secret import *

api = init_twitter('sysmanapy') #### PASO 1

class MyStreamListener(tweepy.StreamListener): #### PASO 2

    #cada vez que se publica un estado
    def on_status(self, status):    #### PASO 6
        autor = status.user.screen_name
        print('Autor: '+autor)
        print('Idioma: '+status.lang)
        print('Estado: \n'+status.text)

        api.create_favorite(status.id);
        api.update_status("Genial! soy el script de @nievesborrero y @PabloLeonPsi, encantado "+ autor , in_reply_to_status_id=status.id);

        print("-"*10)

        # Almacenamos en un documento
        with codecs.open("prueba.txt", "a", "utf-8") as myfile:
            myfile.write('Autor: '+status.user.screen_name+'\n')
            myfile.write('Estado: \n'+status.text+'\n')
            myfile.write('\n-----\n')


if __name__ == '__main__':

    myStreamListener = MyStreamListener() #### PASO 3
    myStream = tweepy.Stream(auth=api.auth, listener=myStreamListener) #### PASO 4
    myStream.filter(track=['probandoprobando']) #### PASO 5
```

horasActivas.py

```python
import tweepy, codecs, sys
from secret import *

api = init_twitter('sysmanapy');

if len(sys.argv) != 2:
   sys.exit('Usage: '+sys.argv[0]+' twitter_user')
else:
    user = sys.argv[1]

manana = tarde = noche = 0

def horas(user):
    for tweets in api.user_timeline(screen_name=user, count=200): # Recorremos el timeline del usuario.
        global manana
        global tarde
        global noche
        dateTime = str(tweets.created_at) # Recogemos la fecha
        array = dateTime.split()
        time = array[1].split(":")
        if(int(time[0])<14):
            manana+=1
        else:
            if(int(time[0])>20):
                noche+=1
            else:
                tarde+=1
    manana = manana/2
    tarde = tarde/2
    noche = noche/2

if __name__ == '__main__':
    horas(user)
    print("--Estadística de twiteo de "+user+"--")
    print('madrugando: '+str(manana)+"%")
    print('tarde: '+str(tarde)+"%")
    print('trasnochando: '+str(noche)+"%")
```

control.py

https://github.com/tweepy/tweepy/blob/master/tweepy/cursor.py

```
# -*- coding: utf-8
import sys, tweepy, shelve, datetime, traceback
import itertools # Libreria de Python con métodos que devuelven iterables eficientes.
from secret import *

api = init_twitter('nieves')
followers = {}
friendsIds = []
lostFollowers = {}
newFollowers = {}
followerIds = []
inactivos = []
followersFile = 'followers_control.twt'
user = sys.argv[1]
inactivityTime = 90 #days
today = datetime.datetime.now()


def paginacion(iterable, pageSize):
    while True:
        i1, i2 = itertools.tee(iterable)
        iterable, page = (itertools.islice(i1, pageSize, None),
            list(itertools.islice(i2, pageSize)))
        if len(page) == 0:
            break
        yield page #devolvemos ese iterable

if len(sys.argv) != 2:
    sys.exit('Usage: '+sys.argv[0]+' twitter_user')


# Recoge la información de usuario
try:
    userInfo = api.get_user(user)
except tweepy.error.TweepError, e:
    traceback.print_exc()
    if e.reason == 'Not found':
        sys.exit('Error: Usuario no encontrado!!')
    elif e.reason.find('Rate limit exceeded'):
        sys.exit('Error: Limite excedido')
    else:
```

Old way vs Cursor way

First let's demonstrate iterating the statues in the authenticated user's timeline. Here is how we would do it the "old way" before Cursor object was introduced:

```
page = 1
while True:
    statuses = api.user_timeline(page=page)
    if statuses:
        for status in statuses:
            # process status here
            process_status(status)
    else:
        # All done
        break
    page += 1  # next page
```

As you can see we must manage the "page" parameter manually in our pagination loop. Now here is the version of the code using Cursor object:

```
for status in tweepy.Cursor(api.user_timeline).items():
    # process status here
    process_status(status)
```

retuitea.py

```
import sys
if 'idlelib.rpc' in sys.modules:
    sys.argv.extend(raw_input("Args: ").split())
import tweepy
from secret import *
# Se comprueba que reciba por parámetro un fichero donde leer los tweets
if len(sys.argv)<2:
    print   "Usage:",sys.argv[0],"file\n\n",
    sys.exit(1)

# Recogemos el fichero
if '-f' in sys.argv:    #fichero
    ftuits=sys.argv[sys.argv.index('-f')+1]
else:
    print "No hay fichero de tuits"
    sys.exit(1)

api = init_twitter('sysmanapy')
yo = api.me()

input_file=open(ftuits,'r')
file_lines=input_file.readlines()
# Extraemos el primer tweet de la lista
n=0
while n<len(file_lines):
    l=file_lines[n].split('/')
    # comprobamos que se corresponda con un tweet
    if l[0][:4]=="http" and l[2]=="twitter.com" and l[4]=="status":
        tuit=l[5].strip('\n')
        break
    n+=1
# Sobreescribimos el fichero extrayendo el tweet
file_lines=file_lines[n+1:]
input_file.close()
output_file=open(ftuits,'w')
output_file.writelines(file_lines)
output_file.close()

#Y se retwitea
api.retweet(tuit)
print "Retweet de estado ",tuit," realizado con éxito"
```

Más cosas chulas

- Recoger emojis en tiempo real http://emojitracker.com/
- Análisis de sentimientos https://www.meaningcloud.com/es/
- Monitorizar actividad potencialmente peligrosa
- Heat maps https://github.com/manugarri/tweets_map
- Búsqueda de patrones de riesgo de abuso, trastornos alimenticios…

¿Sugerencias?

https://github.com/tweepy-sysmana/sysmana2018

Gracias!

@nievesborrero

@PabloLeonPsi

NievesBorrero

pabloleonalcaide