BIG DATA
LAB SESSION

Streaming data from Twitter

Seguir la pipeline i fer-ho per a l'event que volguem i hashtag que volguem. Omplir la DB amb tot el que trobem sobre aquest event i el que l'envolta. Al final analitzarem aquestes dades. Ser ambiciós. Com més relevant siguin el tipus de tweets que seguim, i com més important sigui la pregunta que fem a les dades, molt millor. Què hem après de les dades? (Main goal of the course).

Amb el que tenim aquí farem el posterior analysis, model, etc. Activitat incremental.

Idea: Tenir una capçalera de diari amb el que hem trobat a les dades.

PREREQUISITES

- Regarding Python:
  - Version 2.7.x installed
  - Python packages:
    - pymongo
    - tweetpy
    - matplotlib
    - numpy

- Regarding Twitter:
  - Must have a Twitter account
  - Application registered (you have the keys)

- Regarding the database:
  - Mongodb installed and working

PYTHON 2.7.x

- Install the appropriate python for your platform:

  `https://www.python.org/downloads/`

- Install the necessary packages

```
sudo pip install tweepy        (mac osx)
```

funcions python per a twitter

```
pymongo
```

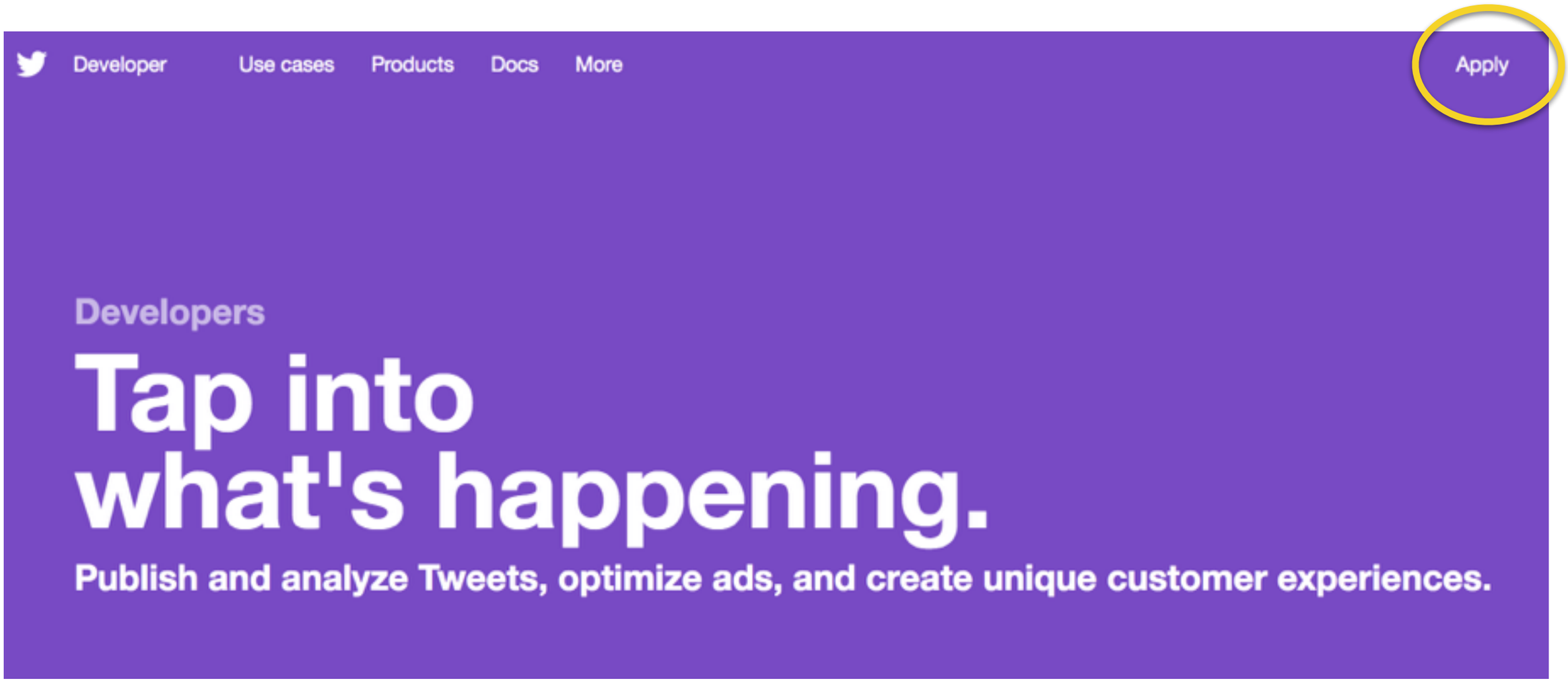
connexió python i mongoDB

```
matplotlib

numpy
```

TWITTER API

- Go to the developer area from Twitter and log in with your Twitter user:

  `https://developer.twitter.com/`



TWITTER API

Twitter enterprise APIs

Our enterprise APIs offer the highest level of access and reliability to those who depend on Twitter data.

Apply for enterprise access >

Twitter standard APIs

Our free, standard APIs are great for getting started, testing an integration, or validating a concept.

Get started with standard access >

Twitter Ads API

The Ads API gives partners a programmatic way to integrate with the Twitter Ads platform.

Apply for Ads API access >

Twitter publisher tools and SDKs

Bring live conversation into your website or app with tools and SDKs available in Twitter for Websites and Twitter Kit.

Get started with publisher tools >

Get started: Build an app on Twitter

Twitter's API platform includes numerous endpoints to help you build an app and solution on Twitter. Our basic endpoints are available for free. As your app or solution needs grow, you'll also find enterprise APIs that include increased levels of access.

Get started with the basic REST and Streaming APIs

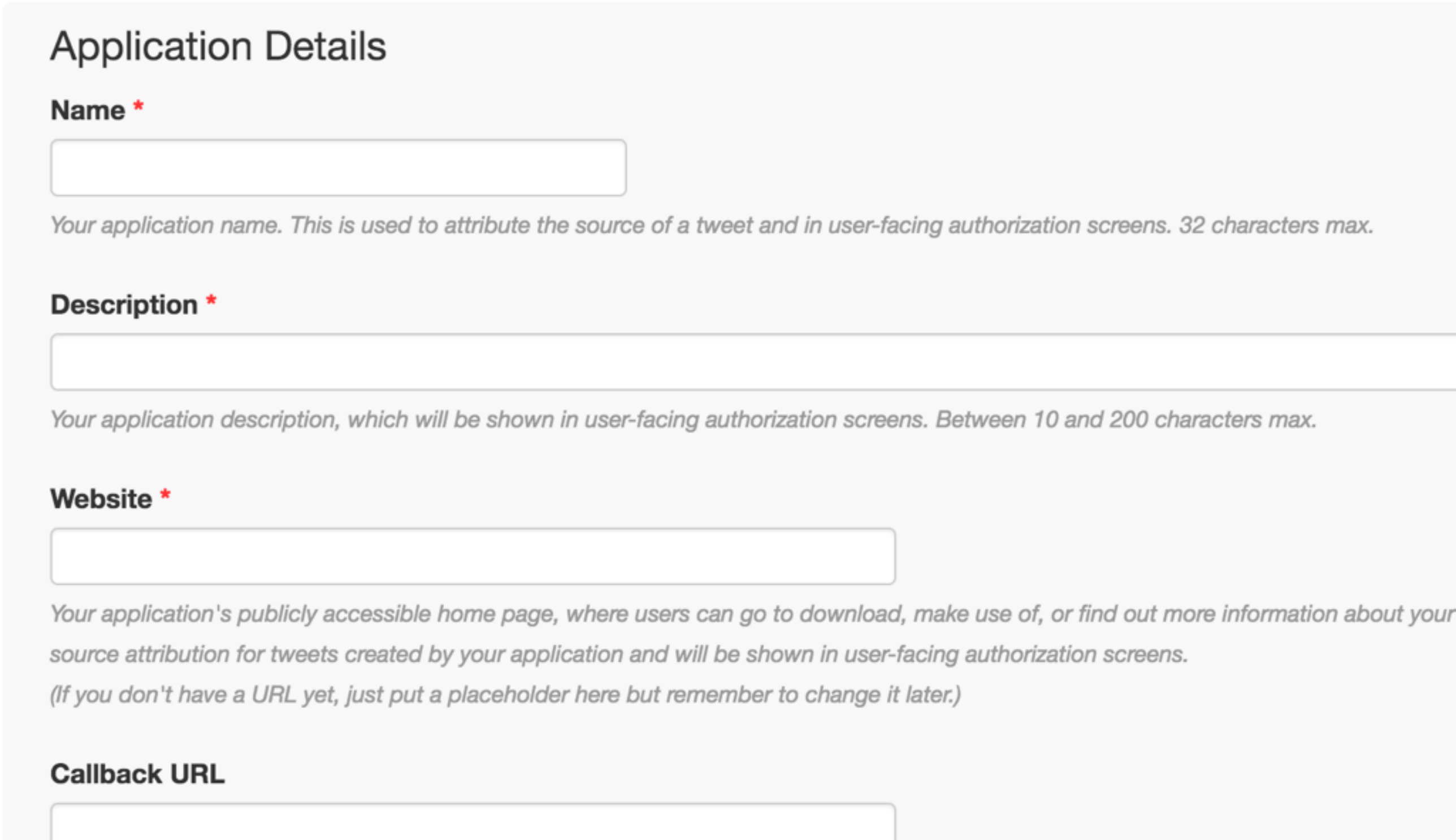
Twitter's basic REST and Streaming APIs enable free access to numerous endpoints. To get started, you must first create an app.

1. Create an app

To use an endpoint, you must create an app and use our OAuth-based authorization system. Visit apps.twitter.com to create one.

TWITTER API

- Create your Twitter application:



TWITTER API

- Inside your newly created app you can access your keys:
- You need 4 keys:
  - `CONSUMER_KEY`
  - `CONSUMER_SECRET`
  - `ACCESS_TOKEN_KEY`
  - `ACCESS_TOKEN_SECRET`

- Done. This is what you needed from Twitter.

- *NOTES:*
  - *Keys might not be operative immediately*
  - *You'll probably need to associate your phone number to your Twitter account.*

INSTALLING MONGODB

- Download mongodb Community Edition that works for your platform:

  https://docs.mongodb.com/manual/installation/

- EASY INSTALLATION (recommended):
  1. Requires homebrew (go to https://brew.sh/ if you do not have it):
     sudo brew install mongodb
  2. Create the folder where your databases will be stored
     sudo mkdir -p /data/db
  3. Change the owner of the folder per tenir permisos per accedir al directori
     sudo chown yourUserName /data/db
  4. Start mongo once (it should and will fail)
     mongo
  5. Start mongod starts deamon behind the database. Always opened when using mongo. Open ports, etc.

     Obrim dues termonals i en una start deamon i a l'altra la DB.

     mongod

     It should say something like "waiting for connections on port 27017". This means it's working.

Ara tenim tot per poder collect data.

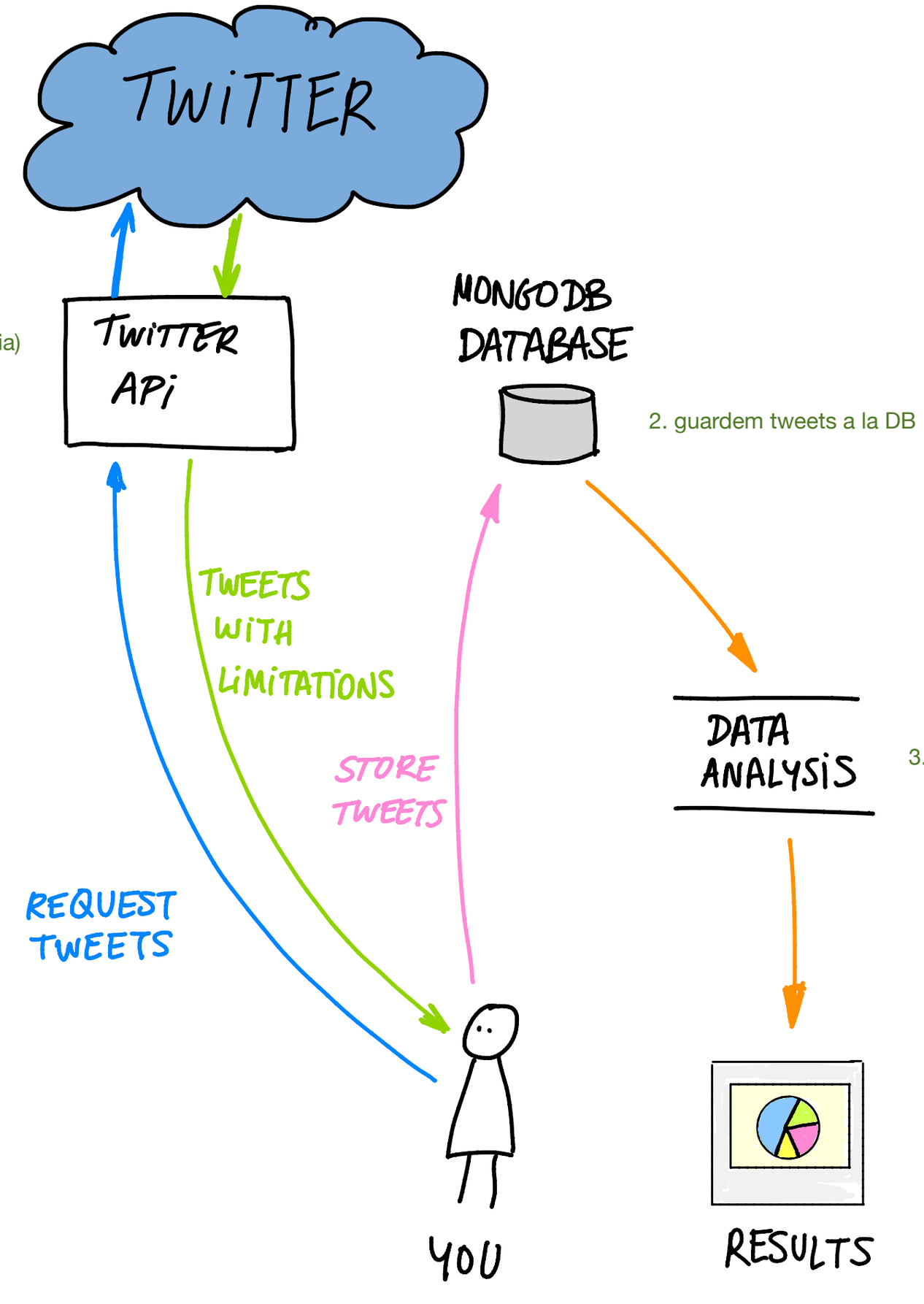
INSTALLING MONGODB

- MANUAL INSTALLATION: Following the instructions for your specific platform from:
  `https://docs.mongodb.com/manual/tutorial`

- To access the mongo console, you need to have 2 processes running (in two different terminals):
  - **`mongod`** (this is a daemon and will continue its execution until the process is killed)
  - **`mongo`** (our access to the database)

LET'S START !

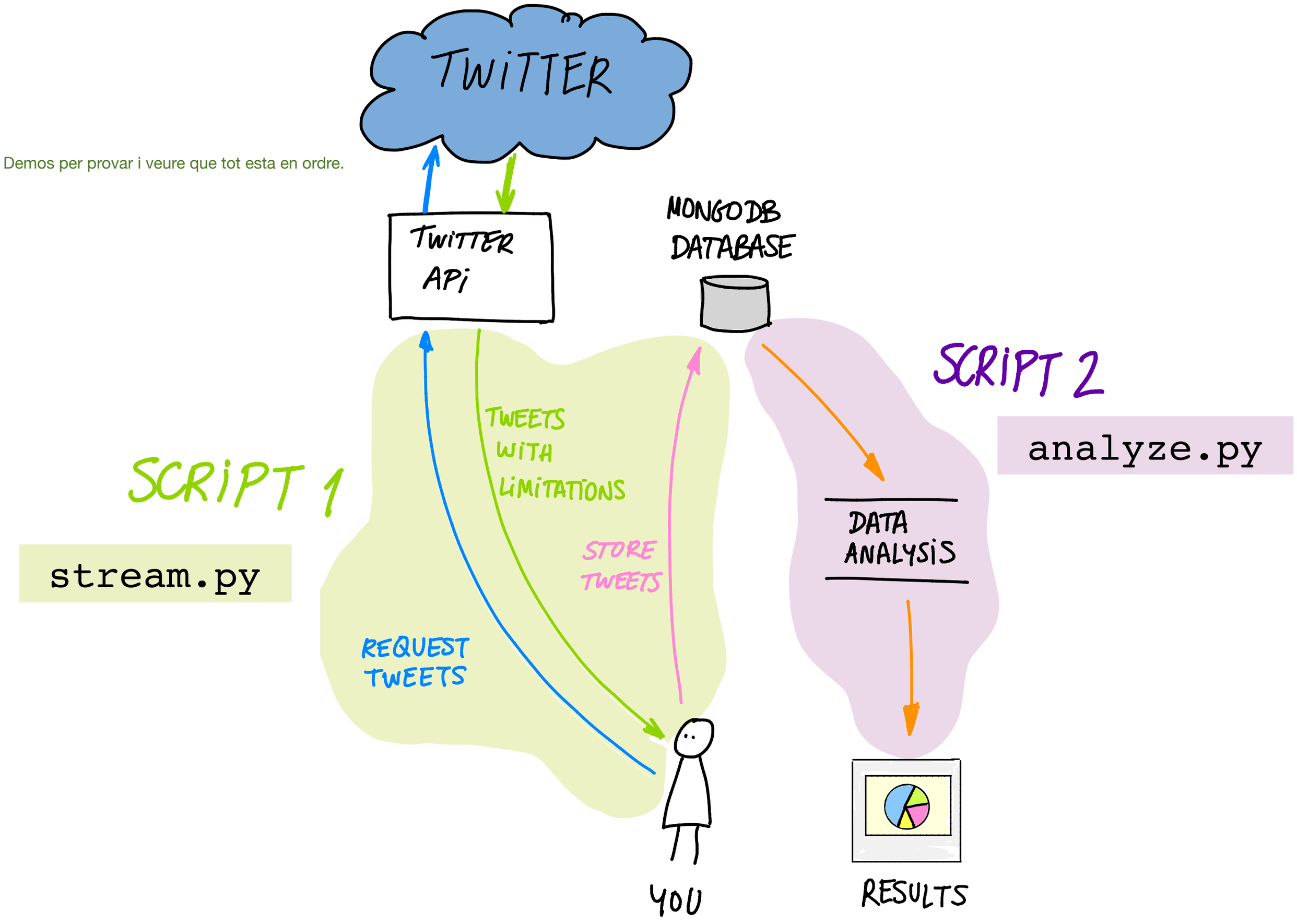
1. API comunica amb twitter
API torna tweets amb some limitations (teoria)



2. guardem tweets a la DB

3. proceed with data analysis

Demos per provar i veure que tot esta en ordre.



STREAM.PY

```
from __future__ import print_function
import tweepy
import json
from pymongo import MongoClient

# assuming you have mongoDB installed locally
# and a database called 'test'
MONGO_HOST= 'mongodb://localhost/test'
```

definim client per a mongoDB

```
WORDS = [‘#HASHTAG1','#HASHTAG2']  #This is an OR relation
```

Les nostres keys per accedir a tweeter des de la nostra plataforma.

```
CONSUMER_KEY = "0eMK1UflvWVq0D4he2V6h"
CONSUMER_SECRET = "CmUkXyF07nWs8UQnpCAHJ3DRoJg3CFfhYWrILjypw1Mv"
ACCESS_TOKEN = "71464516127415243 7-hmUXfCV7EYA4OC9G8rlOC9GPx"
ACCESS_TOKEN_SECRET = “DlikwFPddGIjmgD27iJAzRUd1AsrKWjdYiiZedH4"
```

STREAM.PY

```
class StreamListener(tweepy.StreamListener):
    #This is a class provided by tweepy to access the Twitter Streaming API.

    def on_connect(self):
        # Called initially to connect to the Streaming API
        print("You are now connected to the streaming API.")

    def on_error(self, status_code):
        # On error - if an error occurs, display the error / status code
        print('An Error has occured: ' + repr(status_code))
        return False

def on_data(self, data):
      #This is the meat of the script...it connects to your mongoDB and
      stores the tweet
    try:
        client = MongoClient(MONGO_HOST)

        # Use test database. If it doesn't exist, it will be created.
        db = client.test

        # Decode the JSON from Twitter
        datajson = json.loads(data)
```

you listen whatever is going i n tweeter in real time.

El que fa les coses.

data: data read from the streaming.

STREAM.PY

```
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)

auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)

#Set up the listener. The 'wait_on_rate_limit=True' is needed
to help with Twitter API rate limiting.

listener = StreamListener(api=tweepy.API(wait_on_rate_limit=True))

streamer = tweepy.Stream(auth=auth, listener=listener)

print("Tracking: " + str(WORDS))

streamer.filter(track=WORDS)
```

STREAM.PY

- Let's put it to work:

1. Grab **`stream.py`** from the Lab Exercises Folder
2. Change the keys to your own Twitter API Keys.
3. Change the hashtags to **`#ICTPSAIFRBIGDATA`**
4. Make sure *mongod* process is running!

posem el que volem
rebrem tot el que tingui aquest hashtag
un que creem nosaltres o que sapiguem que està a internet

```
In a terminal:
$ mongod
```

```
If you have an error like: Resource temporarily unavailable. Is a mongod instance already running?,
                terminating. It's because mongod is already running. Skip this step then.
```

5. Run the script: **`python stream.py`**

Tot el que parli del hashtag es guardarà a la MongoDB, in real time.

Start sending tweets with that hashtag!
(From your cellphone, browser...)
You can also retweet other participants tweets and reply to them.

Your script should print a message every time a new tweet is captured

Only tweets with the hashtag will be captured!

ANALYZE.PY

Ara he d'analitzar el que he guardat a la DB.

You can query mongo to see the number of captured tweets by doing:

```
In a new terminal:
$ mongo
$ show dbs          show DataBase
$ use test          Name of DataBase
>> switched to db test
$ show collections
$ db.twitterBrazil.count()     quants tweets tinc a la collection twitterBrazil
>> 12
$ db.twitterBrazil.findOne()  #Print one tweet
```

Stop the stream.py process when you've reached enough tweets

mongodb queries cheatsheet

Semblant a SQL queries

`db.people.find()`	**`SELECT`** `* ` **`FROM`** `people`
`db.people.find({ },` `{user_id: 1, status: 1})`	**`SELECT`** `user_id, status` **`FROM`** `people`
`db.people.find({status:'A'},{})`	**`SELECT`** `* ` **`FROM`** `people` **`WHERE`** `status = 'A'`
`db.people.find({status:'A'},` `{user_id: 1})`	**`SELECT`** `user_id` **`FROM`** `people` **`WHERE`** `status = 'A'`

ANALYZE.PY

```
# -*- coding: utf-8 -*-
from __future__ import division
from pymongo import MongoClient
import matplotlib.pyplot as plt
from collections import Counter
import numpy as np
import operator

# Establish connection with database

client = MongoClient()
db = client.test
col = db.twitterBrazil

##########################################################
# Retrieve data from the mongodb database, choosing
# the fields you'll need afterwards
##########################################################

my_tweets = db.twitterBrazil.find({},{'lang':1, '_id':0, 'text':1,
'entities.hashtags':1,'in_reply_to_status_id':1, 'is_quote_status':1,
'retweeted_status':1, 'user.screen_name':1})

                                    agafar data q necessito d'aquest tweet de la DB per analitzar-lo
numTweets = db.twitterBrazil.count()
```

ANALYZE.PY

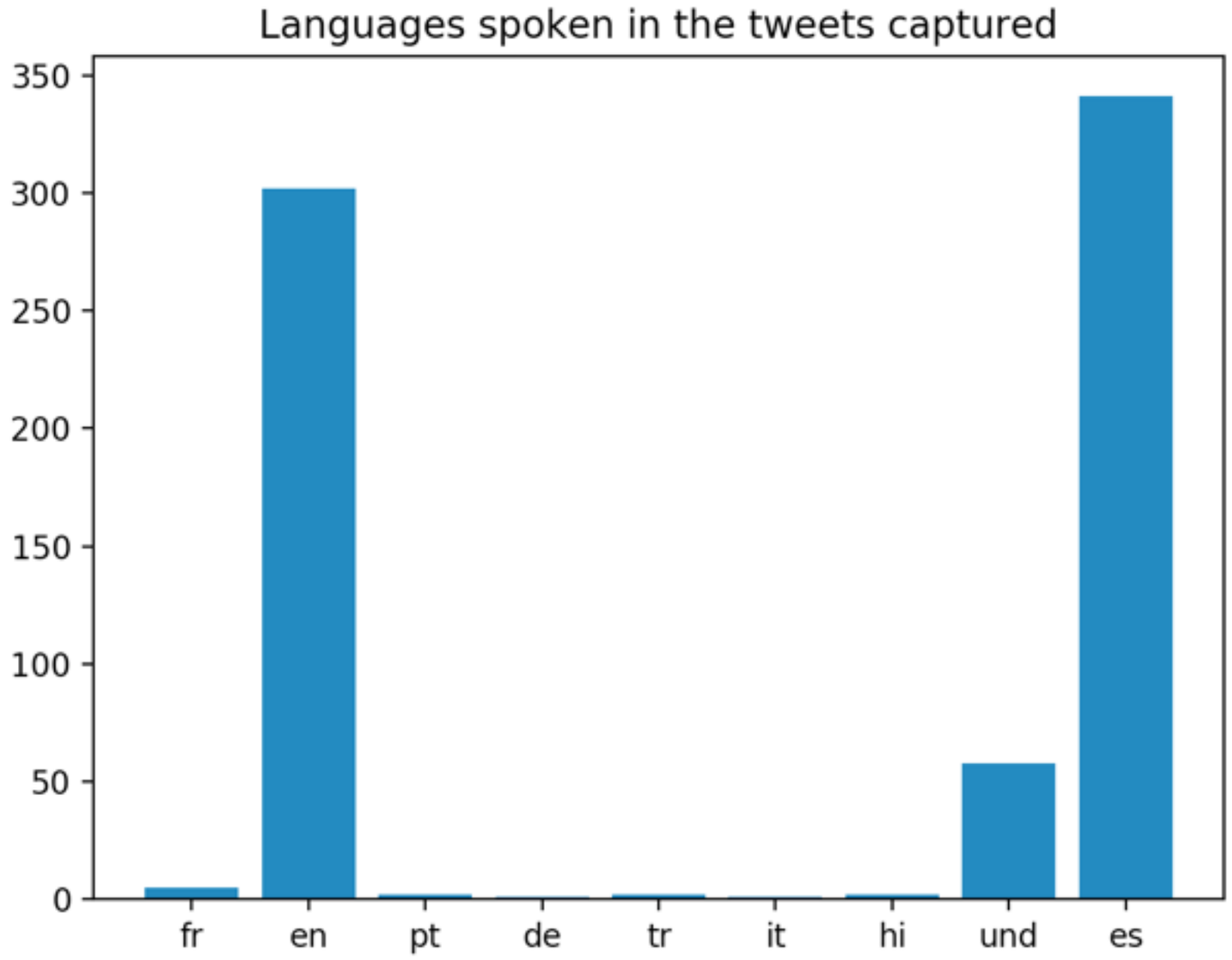
Part #1: code

```
##########################################################
# Plot of Languages (autodetected by Twitter)
##########################################################

langsList = []
for t in my_tweets:
    langsList.append(t['lang'])
D = Counter(langsList)

# ----------- Bar Plot -----------------------
plt.bar(range(len(D)), D.values(), align='center')
plt.xticks(range(len(D)), D.keys())
plt.title('Languages spoken in the tweets captured')
plt.show()
```

Busco l'idioma en el que està escrit el tweet per mirar la distribution of languages.

ANALYZE.PY

Part #1: output



Tweets captured with hashtag #MondayMotivation and #FelizLunes
on Monday, Feb 26 2018

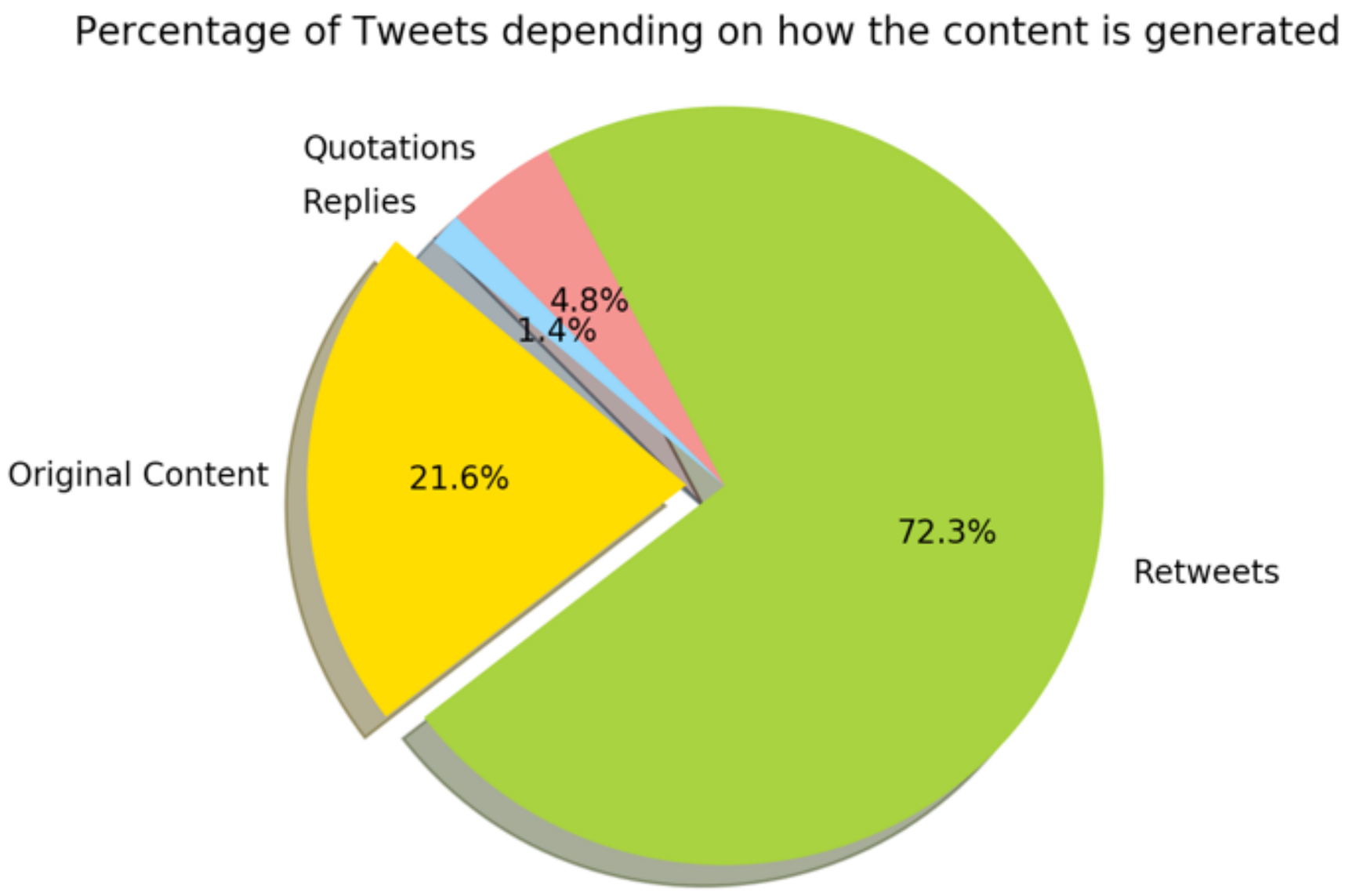
ANALYZE.PY

Part #2: output

```
##################################################################
# Plot how many of them are retweets, replies, quotations or original tweets
##################################################################
my_tweets.rewind() #Reset cursor
retweets = replies = quotations = originals = 0
for t in my_tweets:
    if t.get('retweeted_status') is not None:
        retweets=retweets+1
    elif t['is_quote_status'] is not False:
        quotations = quotations+1
    elif t.get('in_reply_to_status_id') is not None:
        replies = replies+1
    else:
        originals = originals+1
# ---------- Pie Chart -----------------------
labels = 'Original Content', 'Retweets', 'Quotations', 'Replies'
sizes = [originals, retweets, quotations, replies]
frequencies = [x/numTweets for x in sizes]
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue']
explode = (0.1, 0, 0, 0)  # explode 1st slice
# Plot
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=140)
plt.axis('equal')
plt.title('Percentage of Tweets depending on how the content is generated')
plt.show()
```

Proposta: fer servir pychart però podem fer servir el que volguem.

Part #2: output



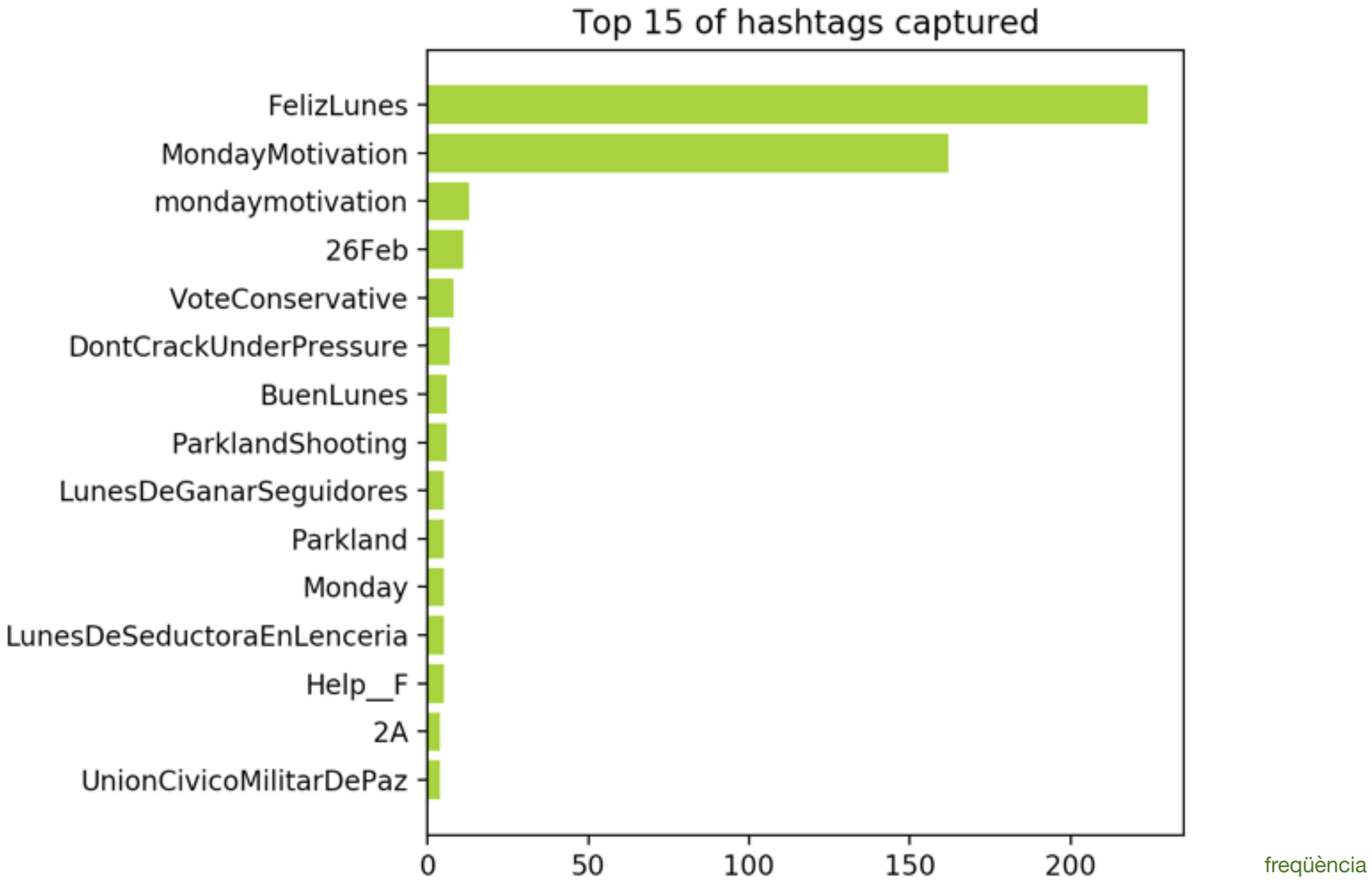
Tweets captured with hashtag #MondayMotivation and #FelizLunes
on Monday, Feb 26 2018

Part #3: code

```
##########################################################################
# Plot secondary hashtags
##########################################################################
my_tweets.rewind()
hashList = []
for t in my_tweets:
    for e in t['entities']['hashtags']:
        h = e['text']
        hashList.append(h)
D = Counter(hashList)
subset = dict(D.most_common(15))
sorted_subset = sorted(subset.iteritems(), key=operator.itemgetter(1))

# ----------- Horizontal Bar Plot ----------------------
pos = range(len(sorted_subset))
plt.barh(pos, [val[1] for val in sorted_subset], align = 'center', color =
'yellowgreen')
plt.yticks(pos, [val[0] for val in sorted_subset])
plt.tight_layout()
plt.title('Top 15 of hashtags captured')
plt.show()
```

ANALYZE.PY

Part #3: output



Tweets captured with hashtag #MondayMotivation and #FelizLunes
on Monday, Feb 26 2018

SOME WARNINGS AND RECOMMENDATIONS

1. You might want to clean your data in the **`stream.py`** script before inserting it into the database
2. Don't trust Twitter! (Fake "Places", incorrectly detected languages...). Recheck everything!
3. If you need to capture data for a long time, consider external hosting.

potser hi ha camps que desapareixen o els canvien.

4. Twitter changes the data structure from time to time.
5. Only <1% of the tweets are geolocalized.

HAVE FUN!