

Bajando datos de Twitter

Usando T-hoarder_kit

Mariluz Congosto @congosto

Índice

1. Captura de datos
2. Taller de uso de la API de Twitter

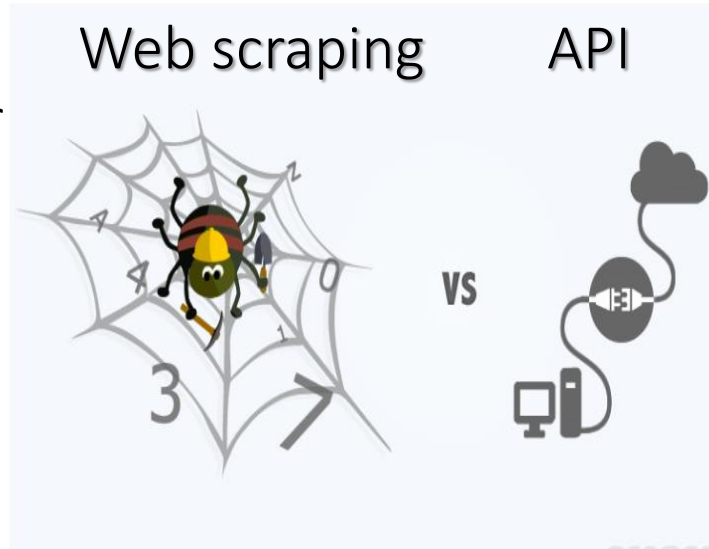
Objetivos del taller

- Comprender el proceso de extracción de datos y su problemática
- Disponer de un kit para aprender y experimentar en la extracción de datos

1. CAPTURA DE DATOS

Captura de datos

- Se puede obtener todo lo publicado en a Web
- No hay rate limit
- Estructura de datos:
 - [HTML, XHTML validated](#)
 - [JSON, JSONP, XML, Microdata](#)



- Se puede obtener lo que ofrezca la API, a veces más, otras veces menos que en la Web
- Hay rate limit
- Menos volumen de información
- Estructura de datos:
 - [JSON](#)

Captura de datos: Autenticación

API

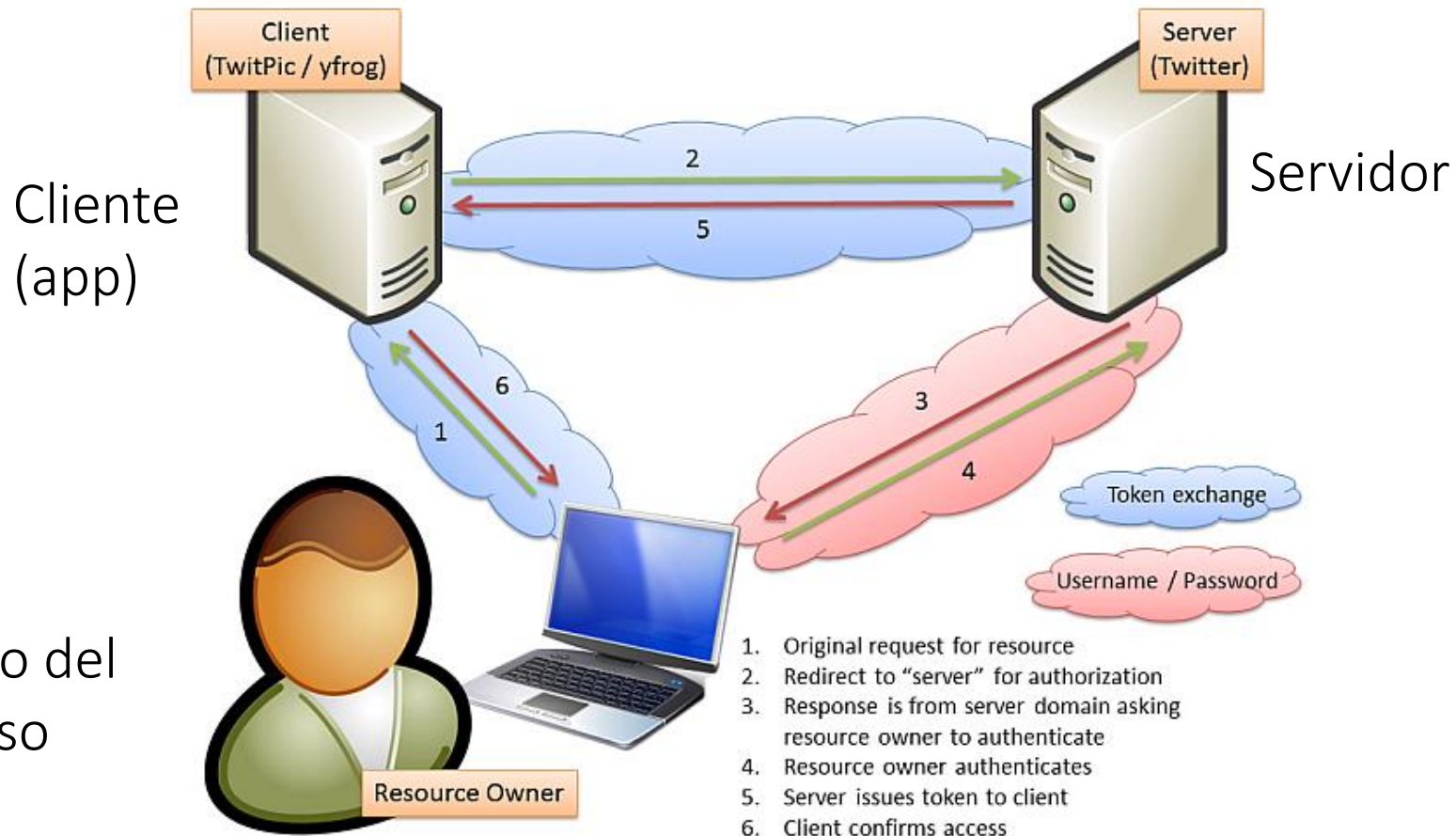


- Autenticación
- Protocolo
- Formatos
- API REST
- API streaming
- Limitaciones
- Consola

Captura de datos: Autenticación



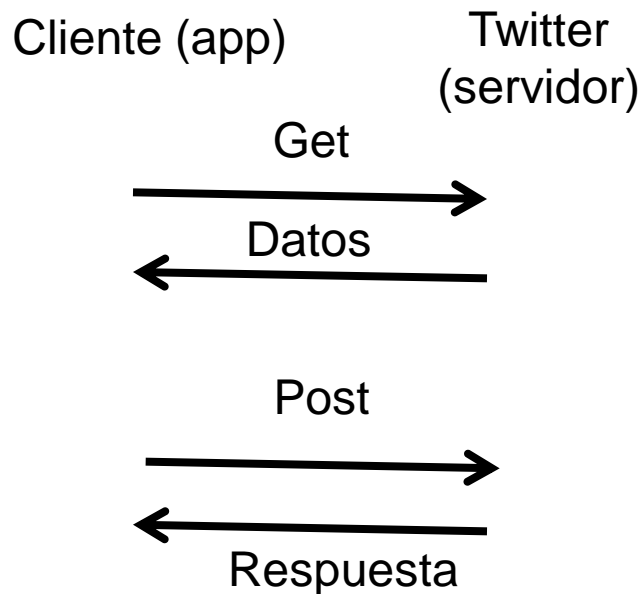
Captura de datos: Autenticación



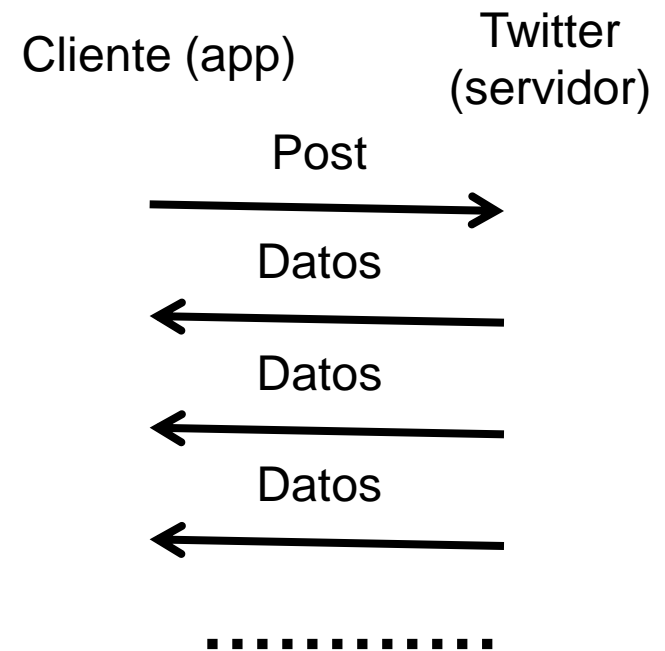
Dueño del recurso

Captura de datos: Protocolos

REST



Streaming



Captura de datos: Formatos

What is JSON?

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is a lightweight data-interchange format
- JSON is language independent *
- JSON is "self-describing" and easy to understand

¡¡Menos mal que existen conversores a csv!!



JSON Example

```
{
  "employees": [
    {
      "firstName": "John",
      "lastName": "Doe"
    },
    {
      "firstName": "Anna",
      "lastName": "Smith"
    },
    {
      "firstName": "Peter",
      "lastName": "Jones"
    }
  ]
}
```

XML Example

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

<http://www.w3schools.com/json/>

Captura de datos: Formatos

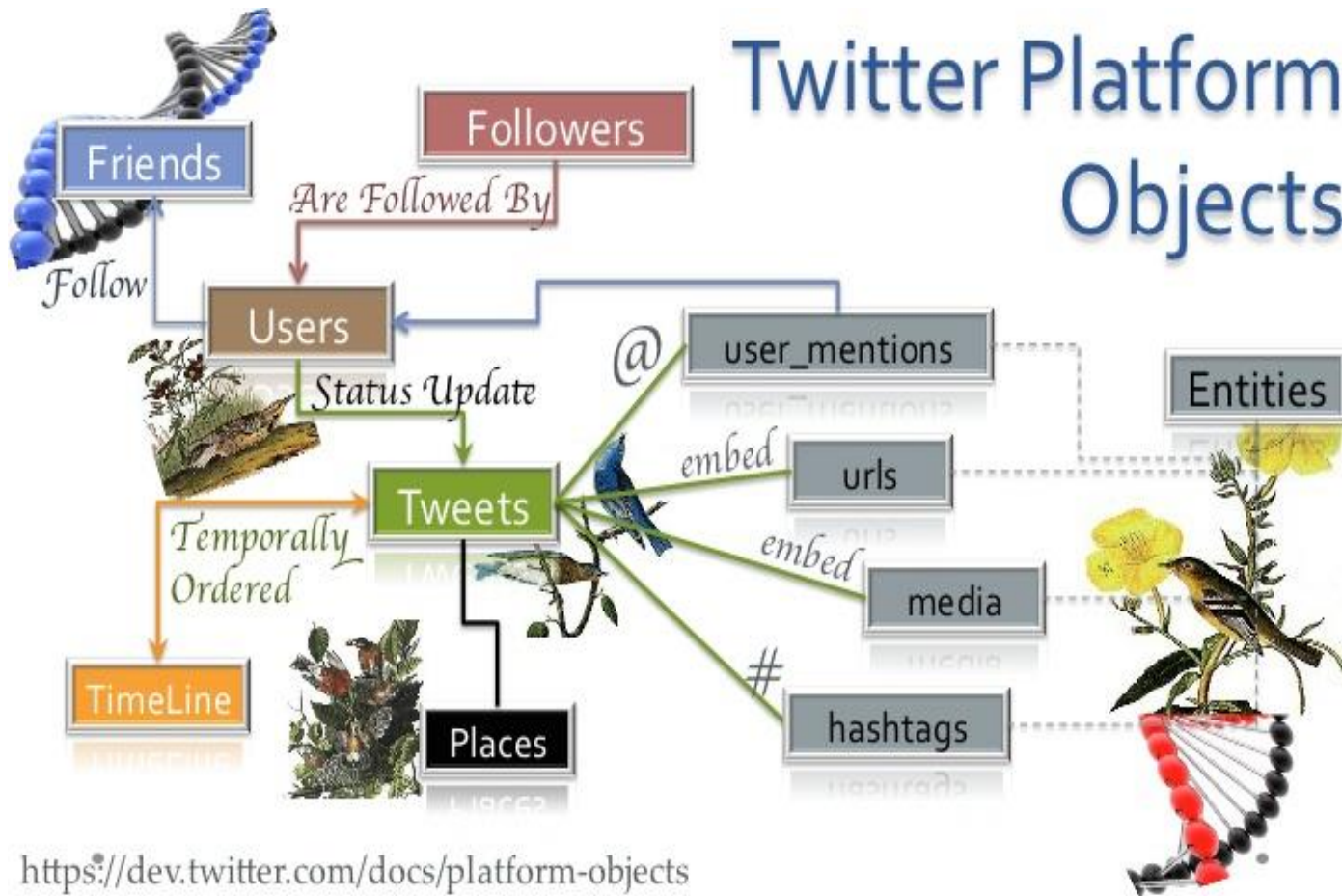


```
{
  "tweet": {
    "created_at": "Thu Apr 06 15:24:15 +0000 2017",
    "id_str": "850006245121695744",
    "text": "1/\ Today we\u2019re sharing our vision for the future of the Twitter API platform!\nht",
    "user": {
      "id": 2244994945,
      "name": "Twitter Dev",
      "screen_name": "TwitterDev",
      "location": "Internet",
      "url": "https://dev.twitter.com/",
      "description": "Your official source for Twitter Platform news, updates & events. Need technic",
    },
    "place": {
    },
    "entities": {
      "hashtags": [
    ],
      "urls": [

```

<https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json>

Captura de datos: Formatos

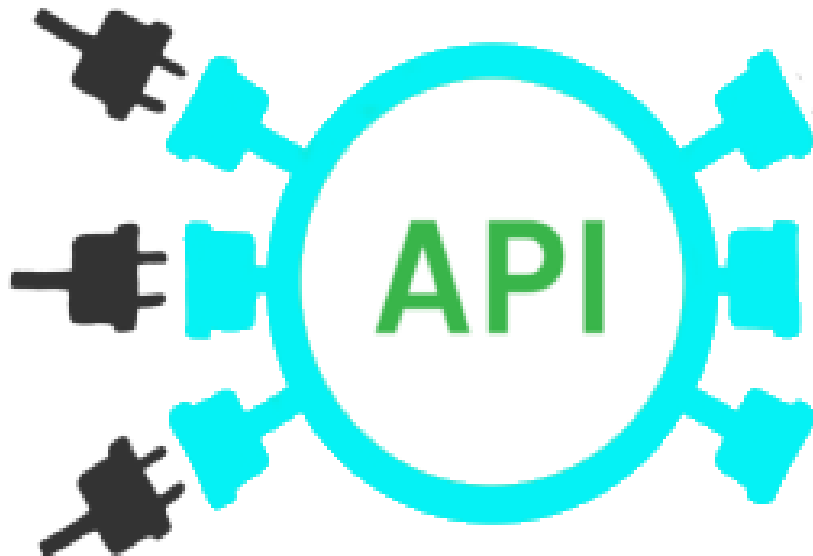


<https://dev.twitter.com/docs/platform-objects>

Captura de datos: APIs

Twitter APIs

Rate limit



REST API

Permite el acceso al core de los datos de Twitter

15-900

15 min.

Search API

Permite realizar búsquedas

180

15 min.

Streaming API

Permite descargar tuits en tiempo real

50

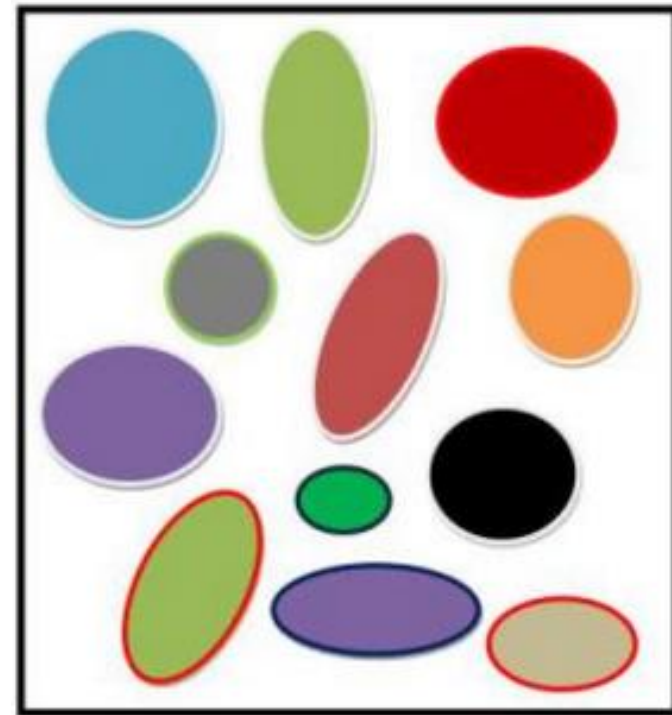
1 seg.

Captura de datos: API REST

Familias

- **Search API**
- **Favoritos**
- **Statuses**
- **Users**
- **Followers**
- **Friends**
- **Friendships**
- **Lists**
- **Trends**
- **Help**

Métodos



Captura de datos: API Streaming

Rate limit

➤ [POST statuses / filter](#)

50 TWEETS/SEGUNDO

➤ [GET statuses / sample](#)

➤ [GET statuses / firehose](#)

SIN LIMITE (DE PAGO)

Captura de datos: API Streaming

Basic Streaming API request parameters

1. delimited
2. stall_warnings
3. filter_level
4. language
5. follow → Lista ID Usuarios
6. track → Lista de palabras
7. locations → Lista de localizaciones
8. count
9. with
10. replies
11. stringify_friend_id

<https://developer.twitter.com/en/docs/tweets/filter-realtime/guides/basic-stream-parameters>

Captura de datos: limitaciones

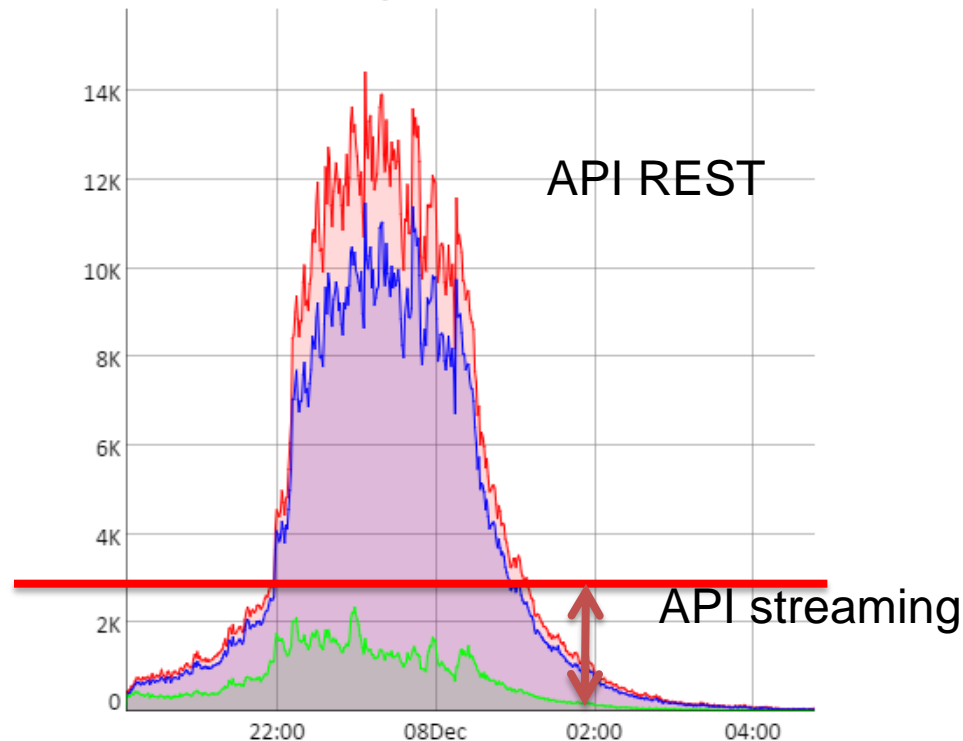


1. Cantidad de información
 - REST: número máximo de consultas cada 15 minutos
 - Streaming: máximo de 50 tuits por segundo
2. Historial
 - REST: una semana de antigüedad
 - Streaming: ninguna antigüedad

Captura de datos: limitaciones

La frecuencia importa: el efecto meseta

Participación de usuarios



<http://t-hoarder.com/7DElDebateDecisivo/>

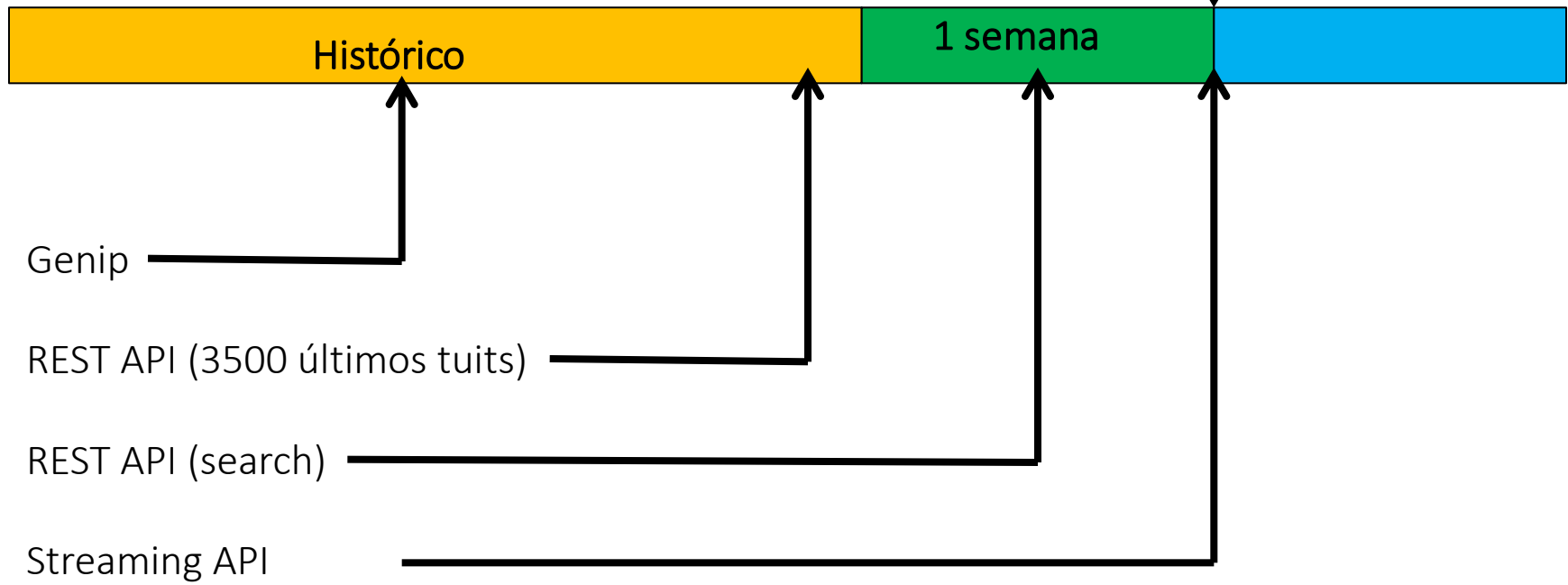
— Nº Tweets — Nº Usuarios únicos — Nº Usuarios nuevos

#7DElDebateDecisivo

Captura de datos: limitaciones

El tiempo importa

Ahora



2. TALLER DE USO DE LA API DE TWITTER

Taller de uso de la API de Twitter



Entorno de trabajo



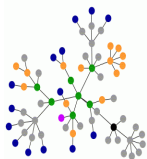
Autenticar



Obtener información de usuarios



Obtener tuits

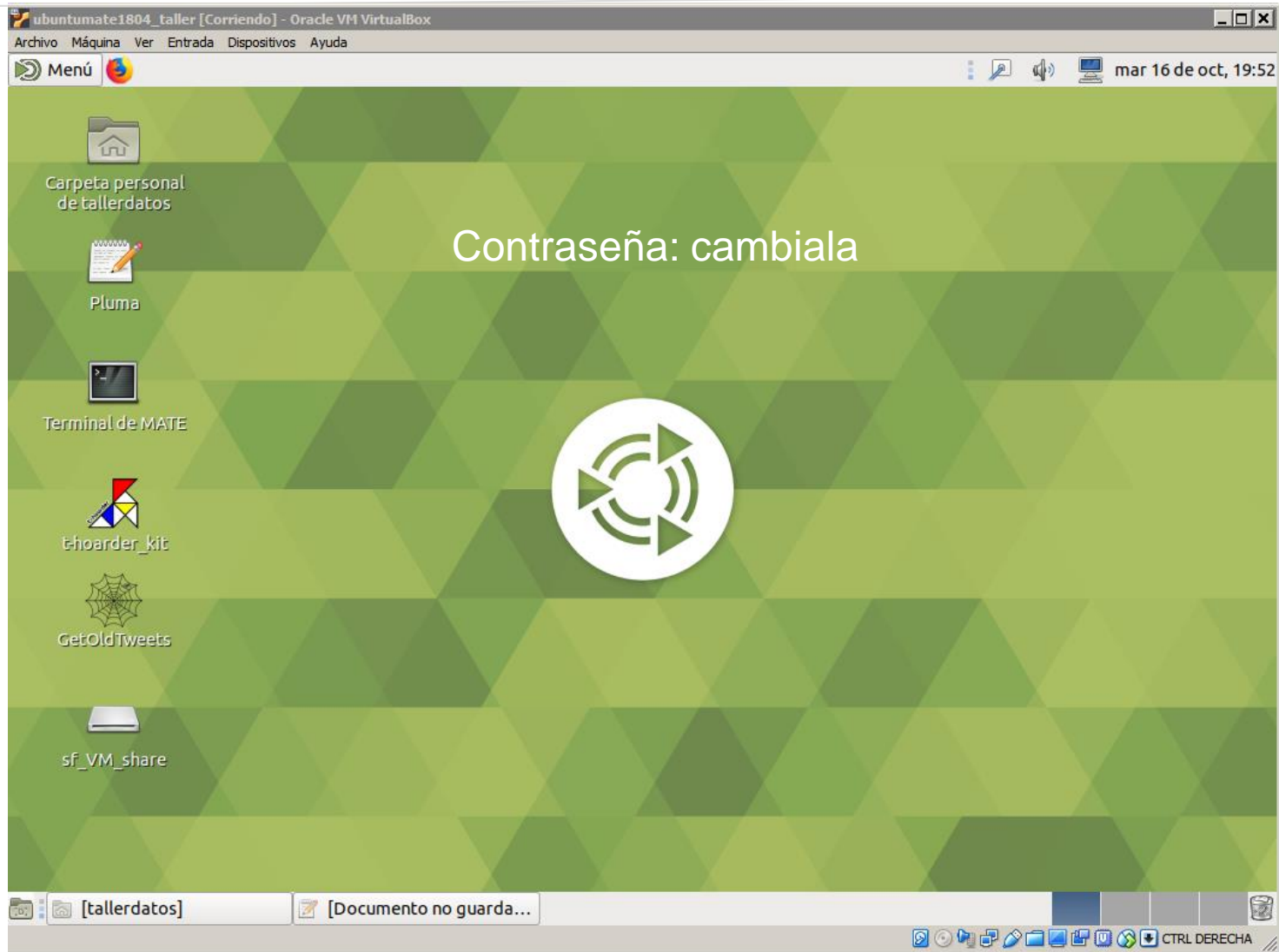


Obtener relaciones

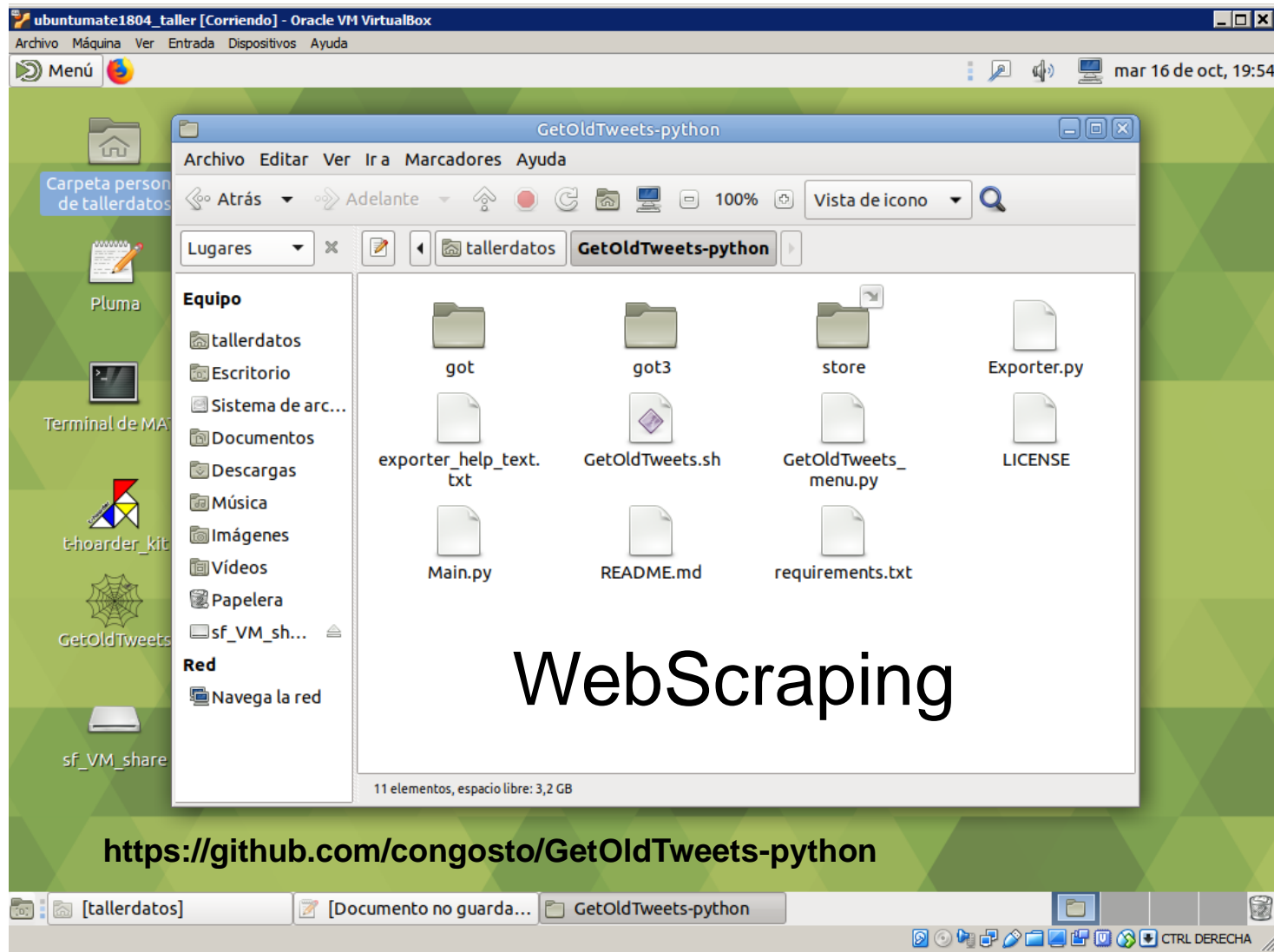


ENTORNO DE TRABAJO

Entorno de trabajo

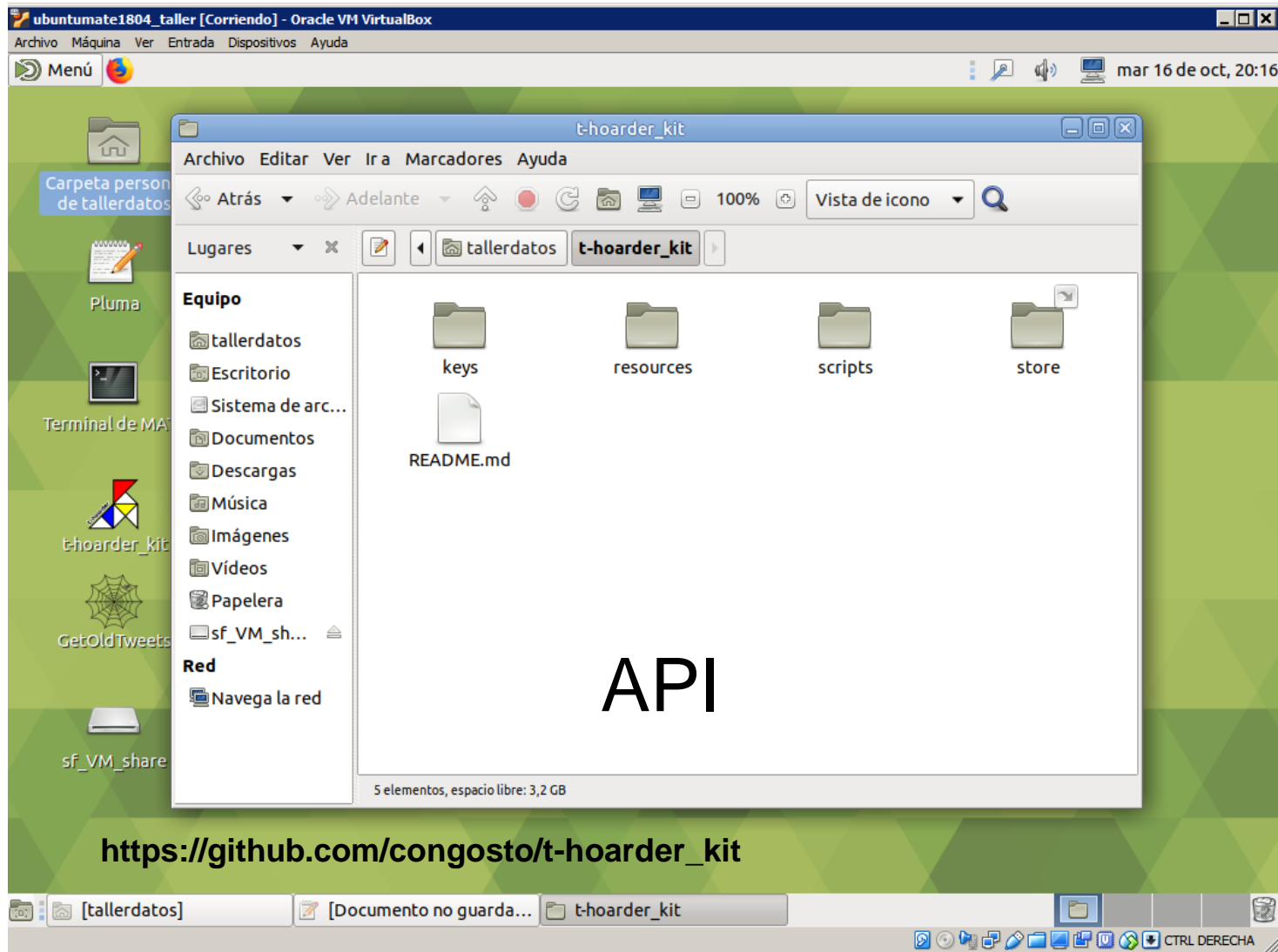


Entorno de trabajo



<https://github.com/congosto/GetOldTweets-python>

Entorno de trabajo



https://github.com/congosto/t-hoarder_kit

Entorno de trabajo

ubuntumate1804_taller [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Menú mar 16 de oct, 20:19

Carpeta personal de tallerdatos

Pluma

Terminal de MATE

t-hoarder_kit

GetOldTweets

GetOldTweets

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
-----> Welcome to GetOldTweets <-----
-----> Environment data <-----
Enter experiment name: prueba
----->
Working in:
  experiment: /home/tallerdatos/GetOldTweets-python/store/prueba/
----->
What function do you want to run?
----->
1. Get tweets by username
2. Get tweets by username and bound dates
3. Get tweets by query search
4. Get tweets by query search and bound dates
5. Exit
-----> Enter option: █
    
```

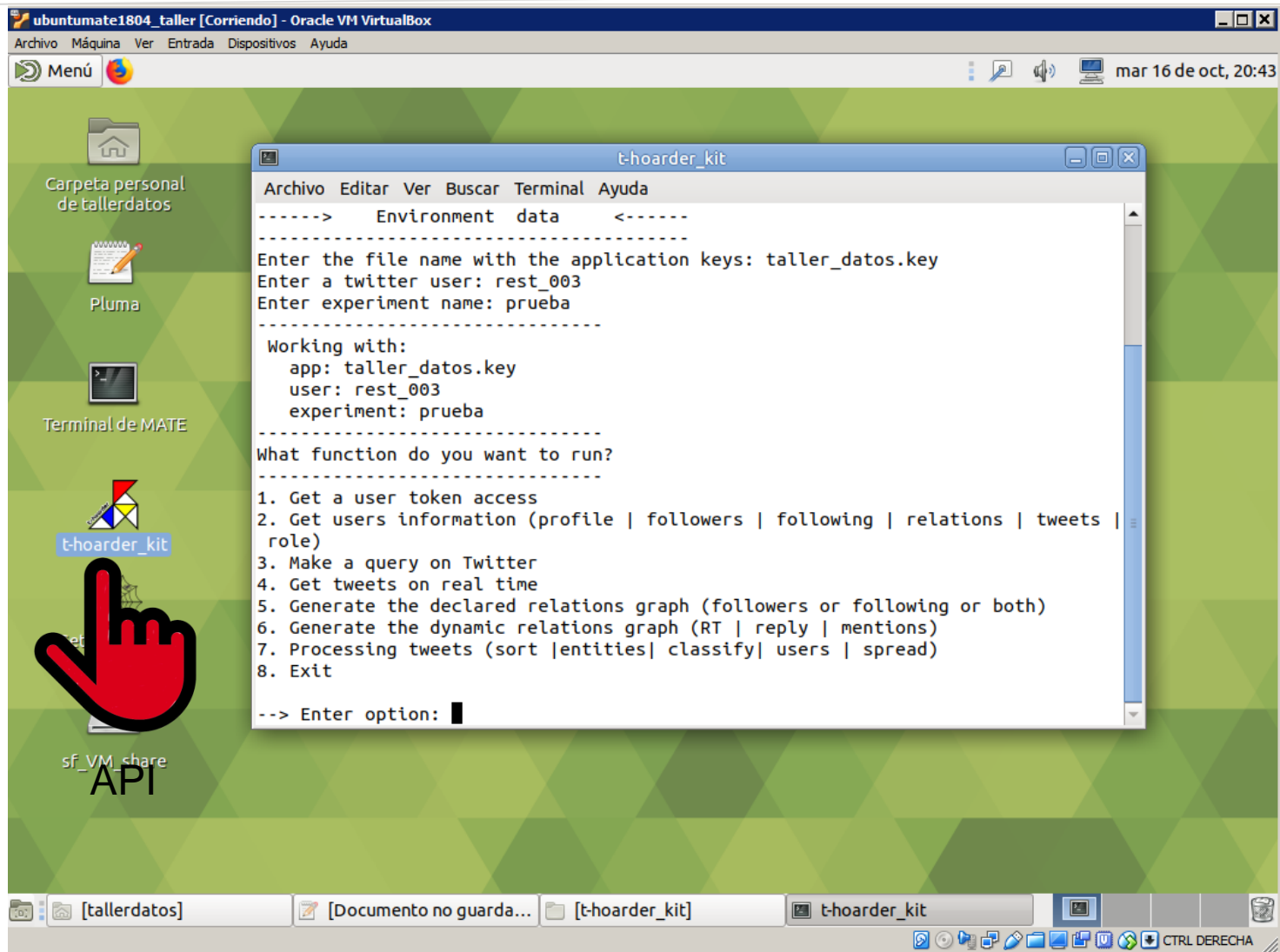
Web scraping

[tallerdatos] [Documento no guarda... [t-hoarder_kit] GetOldTweets



Almudena Garcia
Jurado-Centurion
AlmuHS

Entorno de trabajo



Descarga de datos



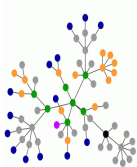
Autenticar



Obtener información de usuarios

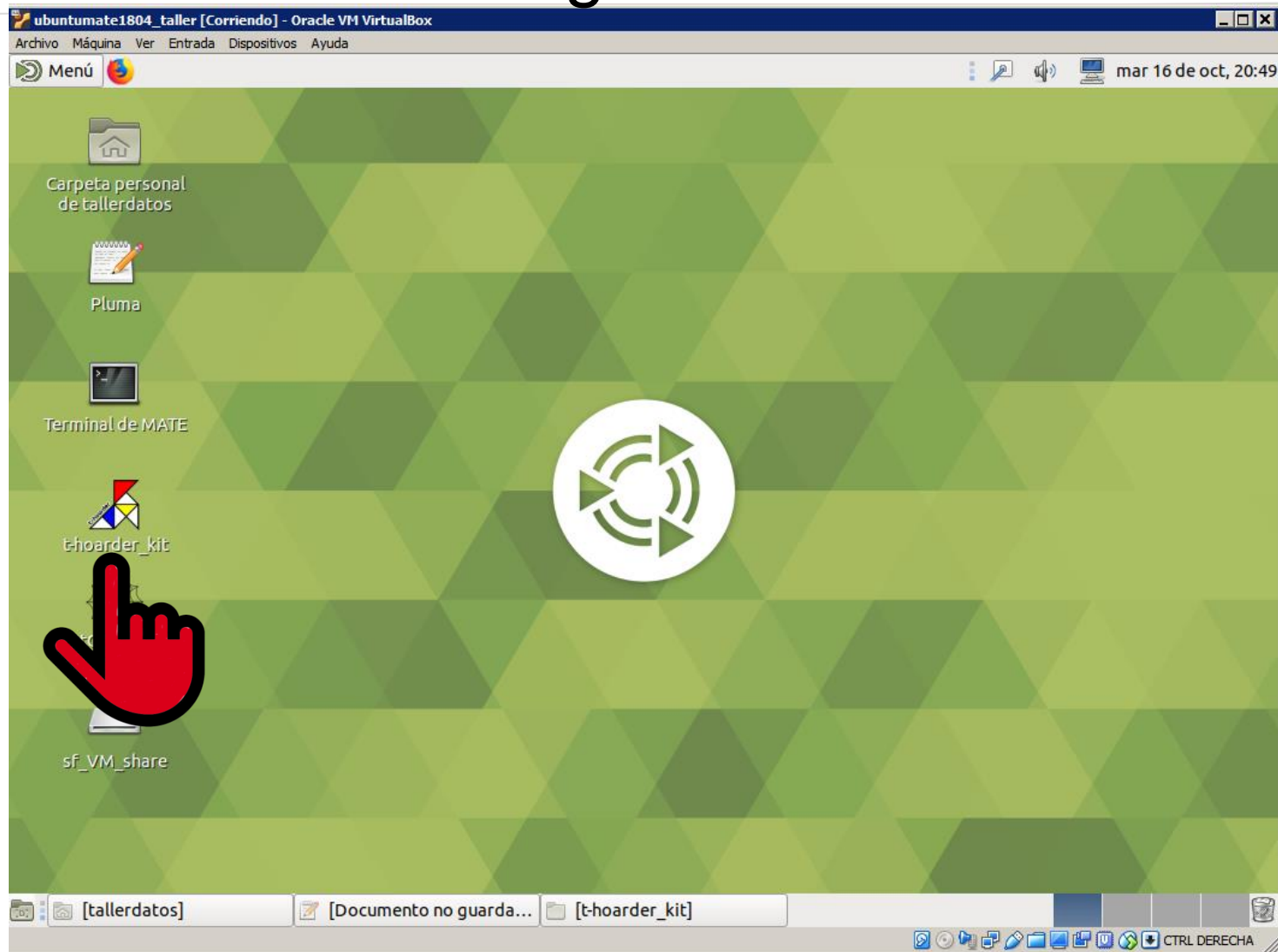


Obtener tuits



Obtener relaciones

Descarga de datos



Datos de contexto

App para acceder -> taller_datos.key

Usuario: cada uno el suyo

Experimento: directorio donde se dejarán los datos

Se introduce la principio y todos las peticiones utilizarán ese contexto

Descarga de datos

The screenshot shows a virtual machine environment with the following components:

- Terminal Window:**

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
-----
Enter the file name with the appl
Enter a twitter user: rest_003
Enter experiment name: prueba
-----
Working with:
  app: taller_datos.key
  user: rest_003
  experiment: prueba
-----
What function do you want to run?
-----
1. Get a user token access
2. Get users information (profile role)
3. Make a query on Twitter
4. Get tweets on real time
5. Generate the declared relation
6. Generate the dynamic relations
7. Processing tweets (sort [entit
8. Exit

--> Enter option: 1
Verification pin number from twitter.com: █

```
- Browser Window 1 (Twitter / Authorize an application - Mozilla Firefox):**

Shows the authorization page for the app 'taller_datos'. The 'Authorize app' button is highlighted with a red arrow pointing to the terminal.
- Browser Window 2 (Twitter / Authorize an application - Mozilla Firefox):**

Shows the confirmation screen: 'You've granted access to taller_datos!'. A blue box contains a verification pin. A red arrow points from this box to the terminal, and another red arrow points from the terminal back to this box. The word 'copiar' is written below the pin box.

At the bottom of the terminal window, the word 'Pegar' is written, indicating the paste action.



OBTENER INFORMACIÓN DE USUARIOS

Información de usuarios

The image shows a screenshot of Donald Trump's Twitter profile. The profile picture is a circular portrait of him. The background of the header is a large crowd of people. The profile name is "Donald J. Trump" with a verified badge and the handle "@realDonaldTrump". The statistics shown are: Tweets: 35,1 K; Siguiendo: 45; Seguidores: 32,8 M; Me gusta: 23; Momentos: 1. A green box labeled "Perfil" points to the profile picture. A green box labeled "Tweets publicados" points to the "Tweets" count. A green box labeled "Conexiones" contains three items: "Perfiles de sus seguidores", "Perfiles de sus seguidos", and "Perfiles de sus contactos". A green box labeled "Role" points to the name and handle. Arrows also point from the "Conexiones" box to the "Siguiendo" and "Seguidores" counts.

Perfil

Tweets 35,1 K Siguiendo 45 Seguidores 32,8 M Me gusta 23 Momentos 1

Donald J. Trump ✓
@realDonaldTrump

Tweets publicados

Perfiles de sus seguidores
Perfiles de sus seguidos
Perfiles de sus contactos

Role

Conexiones

Información de usuarios

Operación	Método	Limitaciones
--profiles	GET users/show	3.600 perfiles /hora
--followers	GET followers/list	12.000 perfiles/hora
--following	GET friends/list	12.000 perfiles/hora
--relations	GET followers/list GET friends/list	12.000 perfiles/hora
--tweets	GET statuses/user_timeline	720.000 tweets/hora

Información de usuarios

The screenshot shows a Linux desktop environment with the following elements:

- Terminal Window (t-hoarder_kit):**

```

-----> Environment data <-----
----->
Enter the file name with the application keys: taller_datos.key
Enter a twitter user: rest_003
Enter experiment name: prueba
----->
Working with:
app: taller_datos.key
user: rest_003
experiment: prueba
----->
What function do you want to do:
----->
1. Get a user token access
2. Get users information (profile)
3. Make a query on Twitter
4. Get tweets on real time
5. Generate the declared relationships
6. Generate the dynamic relationships
7. Processing tweets (sort by)
8. Exit
--> Enter option: [ ]

```
- Text Editor (Pluma):**

users.txt (sf_VM_share ~/t-hoarder_kit/store/prueba) - Pluma

```

1 @vox_es

```
- Desktop Environment:**
 - Desktop icons: Carpeta personal de tallerdatos, Pluma, Terminal, t-hoarder_kit, GetOldTweets, sf_VM_share.
 - Taskbar: [tallerdatos], users.txt (sf_VM_share...), [t-hoarder_kit], t-hoarder_kit.
 - System tray: CTRL DERECHA.

- Abrir el editor pluma
- Escribir la lista de usuarios, uno por línea
- Guardarlo en el directorio del experimento (en este caso prueba con el nombre de usuarios.txt)

Información de usuarios

```

experiment: prueba
-----
What function do you want to run?
-----
1. Get a user token access
2. Get users information (profile | followers | following | relations | tweets |
role)
3. Make a query on Twitter
4. Get tweets on real time
5. Generate the declared relations graph (followers or following or both)
6. Generate the dynamic relations graph (RT | reply | mentions)
7. Processing tweets (sort |entities| classify| users | spread)
8. Exit

--> Enter option: 2
Enter input file name with the list of users or list of profiles (each user in a
line): users.txt
Enter an option (profile | followers | following |relations | tweets| role) : fo
llowing
-->Results in users_following_profiles.txt

Getting user following @vox_es
user: @vox_es --> getting 877 following profiles
    
```



OBTENER TUIITS

Obtener tuits

Operación	Método	Limitación
Buscar tweets	GET search/tweets	72.000 tweets /hora
Bajar tweets en tiempo real	POST statuses_filter	Máximo de 180.000 tweets/hora

<https://twitter.com/search-advanced?lang=es>

Obtener tuits

```

ubuntu1804_taller [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Menú
Carpeta personal de tallerdatos
Pluma
Terminal de MATE
t-hoarder_kit
GetOldTweets
sf_VM_share

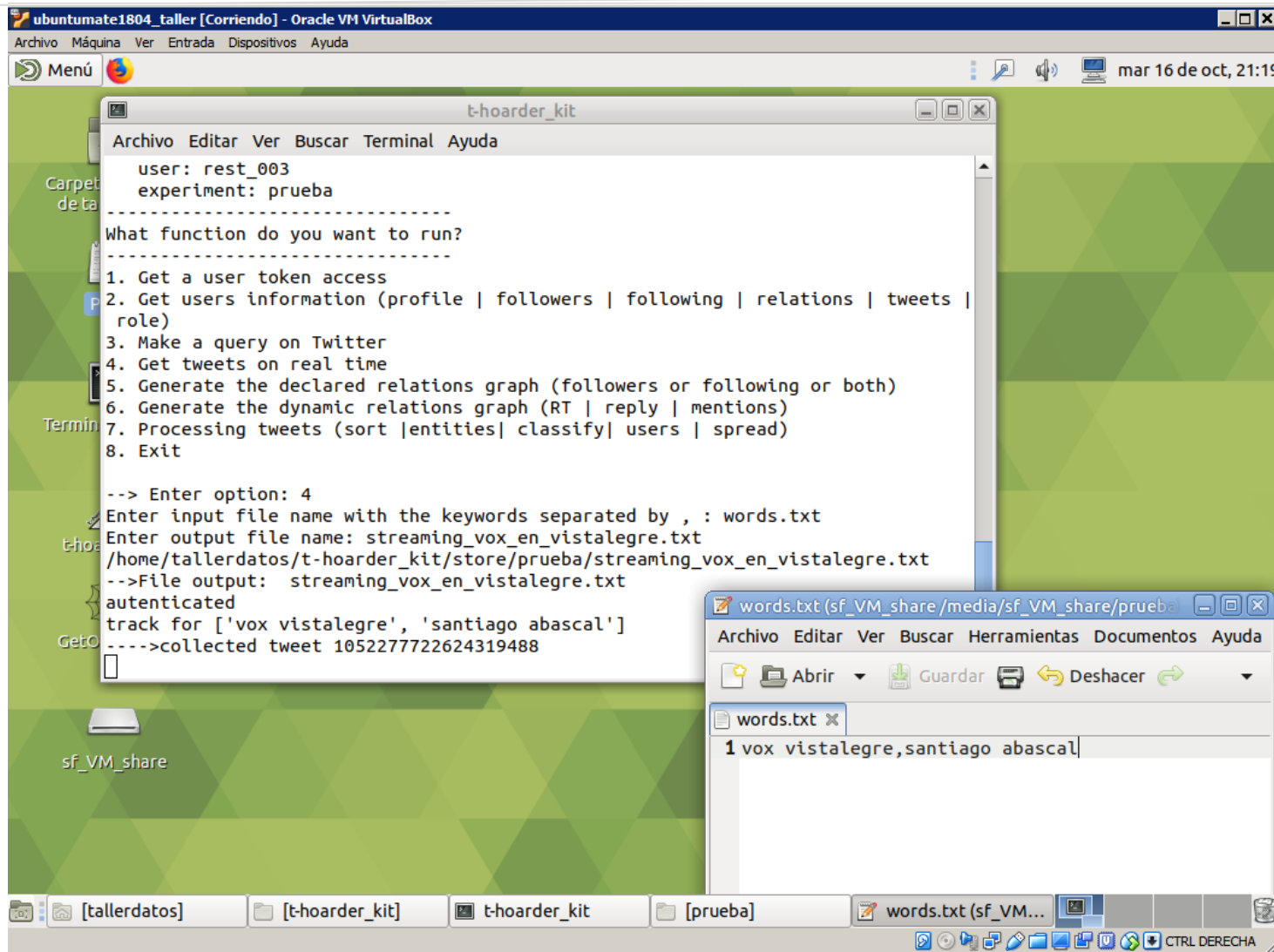
t-hoarder_kit
Archivo Editar Ver Buscar Terminal Ayuda
user: rest_003
experiment: prueba
-----
What function do you want to run?
-----
1. Get a user token access
2. Get users information (profile | followers | following | relations | tweets |
role)
3. Make a query on Twitter
4. Get tweets on real time
5. Generate the declared relations graph (followers or following or both)
6. Generate the dynamic relations graph (RT | reply | mentions)
7. Processing tweets (sort |entities| classify| users | spread)
8. Exit

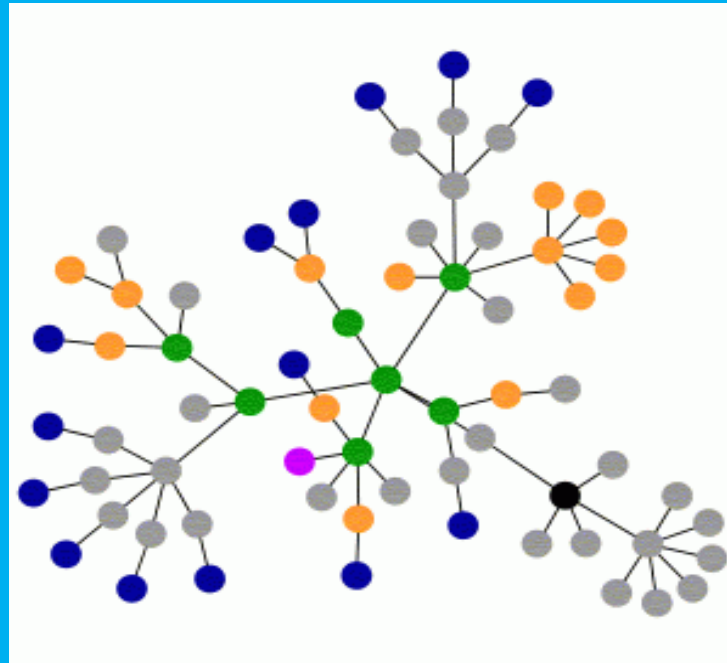
--> Enter option: 3
Enter a query (allows AND / OR connectors): VOX OR vistalegre
Enter output file name: vox_en_vistalegre.txt
/home/tallerdatos/t-hoarder_kit/store/prueba/vox_en_vistalegre.txt
results in vox_en_vistalegre.txt

collected 0
remaining hits 179

```

Obtener tuits



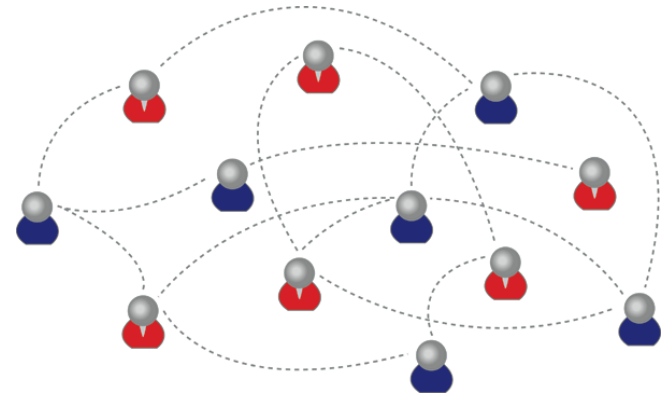


OBTENER RELACIONES

Obtener relaciones

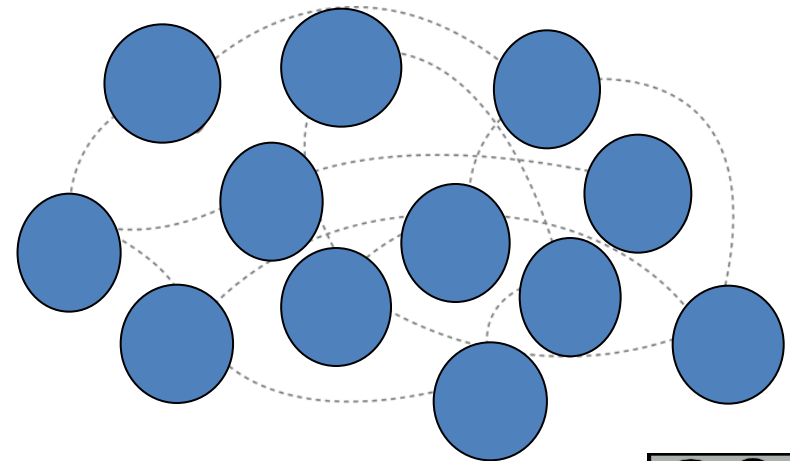
Con teoría de **grafos**, que modela:

- Individuos como **nodos**
- Relaciones como **aristas**



Un **grafo** es una abstracción que representa una red, donde:

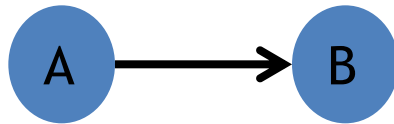
- Un conjunto de **nodos** o **vértices** está conectado mediante **aristas** o **enlaces**



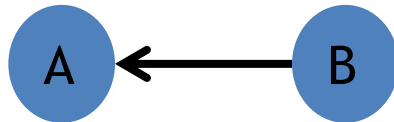
Obtener relaciones

Relaciones declaradas

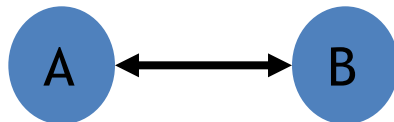
A sigue a B



A es seguido por B

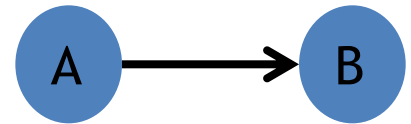


A y B se siguen mutuamente



Relaciones dinámicas

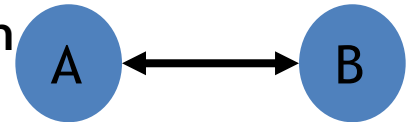
A retuitea a B



A es retuiteado por B



A y B se retuitean mutuamente



Obtener relaciones

Operación	Método	Limitación
Relaciones declaradas	GET followers/ids GET friends/ids	60 peticiones hora Máximo 300.000 ids /hora (5.000 lds por petición) Con -fast 60 conexiones de usuarios /hora
Relaciones dinámicas	No necesita la API-	-

Obtener relaciones

```

ubuntumate1804_taller [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Menú mar 16 de oct, 21:24

t-hoarder_kit
Archivo Editar Ver Buscar Terminal Ayuda
-----
1. Get a user token access
2. Get users information (profile | followers | following | relations | tweets |
  role)
3. Make a query on Twitter
4. Get tweets on real time
5. Generate the declared relations graph (followers or following or both)
6. Generate the dynamic relations graph (RT | reply | mentions)
7. Processing tweets (sort |entities| classify| users | spread)
8. Exit

--> Enter option: --> Enter option: 5
Enter input file name with the users profiles (It is necessary to get before the
users profiles): users_following_profiles.txt
opción --fast? (y/n:) n
-----

there are 0 commons users of 878 nodes
without the --fast option all following per user will be used
time estimated 25.90 hours
there are 76 users with more than 5.000 following
with --fast option only a maximum of 5000 folowing per user will be used
with --fast option, time estimated 14.63 hours

Continue? (y/n)
  
```

Obtener relaciones

```

ubuntumate1804_taller [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Menú
t-hoarder_kit
Archivo Editar Ver Buscar Terminal Ayuda
working with:
  app: taller_datos.key
  user: rest_003
  experiment: prueba
-----
What function do you want to run?
-----
1. Get a user token access
2. Get users information (profile | followers | following | relations | tweets | role)
3. Make a query on Twitter
4. Get tweets on real time
5. Generate the declared relations graph (followers or following or both)
6. Generate the dynamic relations graph (RT | reply | mentions)
7. Processing tweets (sort |entities| classify| users | spread)
8. Exit

--> Enter option: 6
Enter input file name with the tweets (got from a query or in real time): vox_en_vistalegre
.txt
Enter the relationship type (RT | reply | mention): RT
Introduce top size (100-50000): 10000
file name vox_en_vistalegre.txt
-----> Extracting relation RT

-----> second pass
format gdf
type: top nodes: 10000
generating gdf file
type: all nodes: 120
generating gdf file
  
```

Resumen

- Comprender el proceso de extracción de datos y su problemática
- Asumir que obtener los datos (gratis), tiene limitaciones de volumen y temporales
- Disponer de un kit para experimentar y profundizar en estos conceptos
- Posibilidad de mejorar o ampliar los scripts de t-hoarder_kit