

Infraestructura de Twitter



Universidad de Granada

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA
DE TELECOMUNICACIONES

Servidores Web de Altas Prestaciones

Aitor Jiménez Segura
Francisco Vázquez Escobar

Número de Wiki: 7
Número de horas: 22

Contents

1	Introducción	1
2	Preliminares	2
2.1	Historia	2
2.2	Distribución del Contenido	3
3	Análisis	5
3.1	Interfaz	5
3.1.1	Tuits	5
3.1.2	Hilos	6
3.2	Infraestructura	7
3.2.1	Tráfico de Red	7
3.2.2	Almacenamiento	8
3.2.3	Caché	10
3.3	Demostración	11
3.3.1	Twint	11
3.3.2	Instalación	11
3.3.3	Ejemplo	12
4	Conclusiones	16
4.1	Reflexión sobre la descentralización	16
4.2	Cómo evitar convertirnos en Eichmann	17
	Bibliografía	19
	Appendices	20
A	Herramientas utilizadas	20

List of Tables

3.1	Instalación de Twint	12
3.2	Código de Ejemplo	15

List of Figures

3.1	Distribución Hardware.	7
3.2	Topología de Red.	8
3.3	Scrapping en la Terminal	13
3.4	Interfaz Gráfica original	13
3.5	Interfaz Gráfica fallida	14

Chapter 1

Introducción

En este trabajo, nuestro objetivo es utilizar los contenidos vistos en esta asignatura para mostrar una imagen más o menos completa y rigurosa de la infraestructura bajo la que se soporta una de las redes sociales más hegemónicas en la actualidad. Nos referimos a Twitter. Argumentar la magnitud con la que esta aplicación se ha colado en nuestras vidas resulta casi ridículo. La propia RAE ha tenido que introducir en nuestro léxico de forma oficial no sólo su nombre corporativo, sino acciones vinculadas al uso reiterado de la misma como tuitear, retuitear o seguir (obviamente seguir existía ya en nuestro idioma, pero hoy en día hasta su uso mayoritario podríamos decir que se encuentra entorno al pájaro azul). Para ello nuestra idea es dividir el trabajo en dos grandes bloques.

- Por un lado, fundamentalmente en los puntos 2 y 4, trataremos de esbozar cuál es el inmenso calado socio-cultural de esta aplicación, mencionando sus consecuencias políticas y mediáticas o cuál es la distribución de su uso en el día a día.
- En el punto 3, el más extenso de este trabajo y de mayor interés para la asignatura, nos centraremos en tratar de dibujar de la forma más fidedigna posible (hay cosas que, por motivos obvios, no podremos conocer) la arquitectura, la anatomía de cables y servidores que se esconden detrás de este monstruo, además de los problemas y soluciones que sus creadores han ido afrontando por el camino.

Chapter 2

Preliminares

2.1 Historia

Los comienzos de Twitter son algo confusos. Como en muchas ramas de la ingeniería, es difícil situar cuál es el origen de un proyecto que, a la larga, ha sido tan conocido. Podemos pensar en el comienzo como la primera idea, la primera vez que esa idea se plasmó de una manera medianamente seria, la primera vez que esa idea obtuvo financiación o la primera vez que se pudo materializar toda su implementación y ver los resultados.

Plataformas de microblogueo que se quedaron en el camino pero abrieron el paso a Twitter hay muchísimas, de hecho, se cree que la que acabó desembocando en Twitter fue un blog interno de la empresa Odeo [1] para sus propios trabajadores.

Viendo el éxito del mismo, los propios creadores, algunos ex-trabajadores de Google, decidieron tratarlo como un proyecto propio. Como curiosidad, uno de los nombres candidatos fue Twitch, actualmente la plataforma de Streaming de video más famosa del mundo.

Nombres claves en el primer desarrollo del proyecto fueron Evan Williams y Biz Stone (aportaron su experiencia trabajando en Google), Hensaw-Plath (se dice que Twitter se ejecutó por primera vez en su máquina) o Ben Glass (el primero en bajarse del proyecto vendiendo su parte por un Wolskwagen de 7000 dólares).

Para quien quiera profundizar dejamos la referencia [1] en la bibliografía

2.2 Distribución del Contenido

La empresa estadounidense Pear Analytics realizó un estudio en 2009 con el objetivo de conocer la distribución temática de los tweets que se publicaban. Más que temas concretos, buscaban diferenciar utilidades comunicativas de la propia aplicación (ocio, seguimiento de ídolos, publicidad, seguimiento de la prensa, entre otros).

Para ello realizaron un muestreo simple de 200 tweets obteniendo los resultados que se muestran a continuación.

- Cháchara sin sentido – 40%
- Conversaciones – 38%
- Retuits (RT) o mensajes repetidos – 9%
- Autopromoción – 5%
- Mensajes basura (spam) – 4%
- Noticias – 4%

Cabe destacar que estos datos están considerablemente desactualizados. No solamente porque los datos tengan ya 13 años, sino que estos 13 años suponen más de un 80% de la vida de la aplicación. Podemos observar que tanto el crecimiento como el desarrollo cualitativo de la aplicación la ha llevado, por ejemplo, a tomar un papel mucho más político de lo que representan estos porcentajes.

Un artículo que dejamos referenciado por aquí [2] nos deja datos muy interesantes sobre la distribución actual del uso de la plataforma. Nos habla de diferencias por sexo, edad, estatus social, ideología o intereses comunes.

Sabemos a ciencia cierta que esta aplicación ha sido usada como un agente activo para tratar de influir en eventos con una repercusión en nuestra sociedad del tamaño de las elecciones presidenciales estadounidenses, elecciones homólogas en Brasil o el mismo referéndum por el Brexit.

Aunque la plataforma líder en “mediación” política (por su largo historial de influencias sospechosas en el desarrollo socio-político de muchos territorios) es de lejos Facebook por, además, su hegemonía e interseccionalidad en todas las estratos sociales (edad, territorio, renta, ideología...). Aun así, si lo pensamos un poco, la que más se presta a ello o la que mayor potencialidad tiene para ello es Twitter, siendo no sólo periódico ya de varias generaciones, sino por su flexibilidad a la hora de presentar, argumentar y debatir todo tipo de propuestas políticas, sociales e identitarias

de formas muy diferentes, incluso llegando a superar en capacidad de influencia a cualquier medio tradicional.

No podemos olvidar que el gigantesco castillo tecnológico que estamos apunto de desgranar es la consecuencia, y no la causa, de su potencialidad como red social.

Twitter no es grande en nuestra vida porque tenga muchos servidores detrás, recordemos que en este mundo nadie da duros a cuatro pesetas. Twitter se ha vuelto grande en los servidores porque la capacidad comunicativa de la idea que se sustenta detrás es tremendamente potente.

Chapter 3

Análisis

3.1 Interfaz

La interfaz web de Twitter está escrita en un framework de aplicaciones web de código abierto llamado Ruby on Rails (también conocido como RoR o Rails) y los tuits se mantienen en un servidor que funciona con software programado en Scala (lenguaje multi-paradigma) y además dispone de una API abierta para todo tipo de desarrolladores, lo cual supone una gran ventaja para todos aquellos que quieran integrar Twitter como un servicio tanto en otras aplicaciones web como en aplicaciones de escritorio o móviles. Según Biz Stone, más del 50% de nuestro tráfico llega a través de dicha API.

3.1.1 Tuits

Los usuarios pueden agrupar mensajes sobre un mismo tema mediante el uso de etiquetas de almohadilla, palabras o frases iniciadas mediante el uso de una ”#” (almohadilla), conocidas como hashtag, su nombre original en inglés y el que se utiliza normalmente en Twitter. De forma similar, la ”@” (arroba) seguida de un nombre de usuario se usa para mencionar o contestar a otros usuarios. Para volver a publicar un mensaje de otro usuario, y compartirlo con los propios seguidores, la función de retuit se marca con un ”RT” en el mensaje.

A finales de 2009 se añadió la opción de listas, haciendo posible el seguir (así como mencionar y contestar) listas de usuarios en vez de usuarios individuales.

Los mensajes fueron fijados a 140 caracteres máximo para la compatibilidad con los mensajes SMS, introduciendo la notación de la taquigrafía y el argot de Internet comúnmente usado en los SMS. El límite de 140 caracteres también ha llevado a la

proliferación de servicios de reducción de URLs, como bit.ly, goo.gl, y tr.im, y web de alojamiento de material, como Twitpic, memozu.com y NotePub para subir material multimedia y textos superiores a 140 caracteres. El 11 de junio de 2015, Twitter anunciaba que esta restricción de caracteres se eliminaría en los mensajes privados a partir de julio del mismo año, quedando el límite establecido en 10 000 caracteres

En la actualidad, Twitter ha ampliado el límite de 140 caracteres en los tuits a 280, habiéndose realizado este proceso de manera escalonada con millones de usuarios en todo occidente.

Twitter utiliza su propio acortador de enlaces, t.co, lo que además es una medida de seguridad para evitar el spam.

En mayo de 2020, se incorporó una nueva funcionalidad que permite limitar quién puede contestar a los tuits de las personas. De este modo, antes de publicar un tuit el usuario puede escoger entre las siguientes opciones: “cualquier persona puede responder” (everyone can reply), “personas a las que sigues” (people you follow) o “solo las personas que mencionas” (only people you mention).

3.1.2 Hilos

Los hilos (threads en inglés) son una funcionalidad de Twitter que permiten agrupar visualmente diversos tuits. Cuando se crea un tuit, aparece el ícono “+” que permite añadir otro mensaje, y así sucesivamente. De esta manera, el límite de 280 caracteres se puede extender y el usuario puede expresar lo que quiere con mayor precisión. Por otra parte, su uso se ha extendido al ámbito periodístico, donde resulta en una manera más eficaz y concisa de comunicar noticias. Esto ha contribuido al alzamiento de Twitter no solo como una red social, sino también como un medio de comunicación con cierta fiabilidad. De hecho, es cada vez más común que las personas consulten primero Twitter para comprobar la veracidad de alguna noticia, pues suele ser uno de los primeros lugares donde estas son reportadas.

Dado que la mayoría son muy extensos, existen herramientas como Thread Reader App, un bot que permite desplegarlos y leerlos fácilmente. Para aplicar sus servicios, el usuario puede ir a la página web e insertar la URL del hilo, o responder al hilo con la palabra clave “unroll”, siempre mencionando la cuenta del software.

3.2 Infraestructura

Desde el surgimiento de Twitter, los desarrolladores han actualizado su infraestructura de servicio web para tomar ventaja de los últimos estándares en tecnología y hardware de cara a aumentar la eficiencia y proporcionar la mejor experiencia posible. La distribución actual del hardware de Twitter es la siguiente:

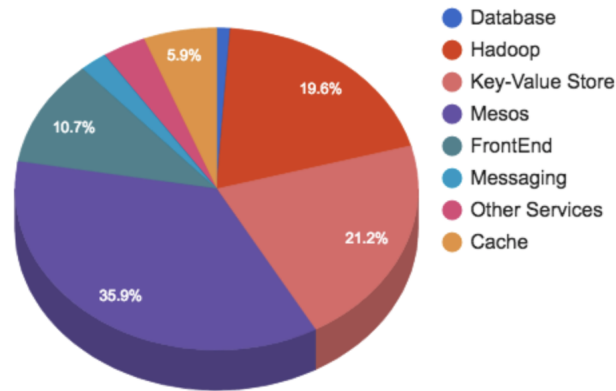


Figure 3.1: Distribución Hardware.

Ocupando más de la mitad de los recursos hardware Hadoop, que es un entorno de trabajo software de licencia libre usado para programar aplicaciones distribuidas que manejan grandes cantidades de datos, y un almacén de tipo clave-valor.

Respecto a la infraestructura, aunque Twitter ha evolucionado mucho y ya permite enlazar fotos y videos (más adelante veremos cómo salva estas cuestiones) la idea es que cada tweet, la unidad básica de información en nuestra aplicación, sea un texto, una cadena de caracteres, asociado a un usuario que publica, una fecha y más información que se va añadiendo a él como número de retweets o me gustas.

3.2.1 Tráfico de Red

En el año 2010, finalizaron su primera arquitectura de red tras migrar de third party hosting para abordar la escalabilidad y los problemas de servicio encontrados. Tenían un deep buffer ToRs para soportar el creciente tráfico de servicio.

Pasados los años, ya se ejecutaba una red con POPs (protocolo de oficina de correos) en cinco continentes y centros de datos con cientos de miles de servidores. En 2015, comenzaron a experimentar algunos errores que fueron incrementándose con el tiempo debido a ciertos cambios en la arquitectura de servicio y a la necesidad de incrementar la capacidad de los servidores. Además, los centros de datos IGP

empezaron a manifestar extraños comportamientos debido a la complejidad de la topología. Para abordar esto, los centros de datos se configuraron a una topología Clos + BGP (protocolo de puertas de enlace) que, a pesar del riesgo, se completó con el mínimo impacto.

Con todo este proceso evolutivo, la red luce del siguiente modo:

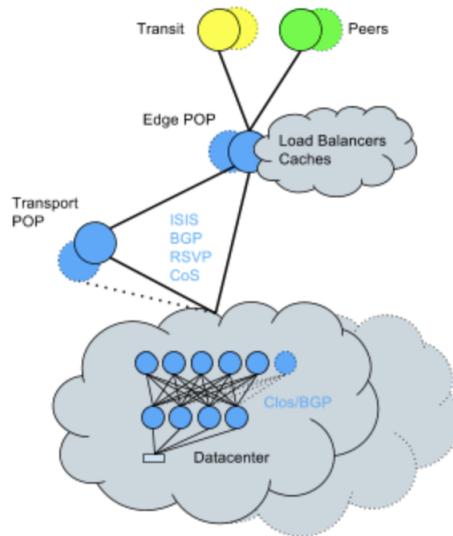


Figure 3.2: Topología de Red.

Este nuevo despliegue aporta los siguientes beneficios:

- Reducción del impacto ante el fallo de un servidor
- Ancho de banda escalable
- Menor sobrecarga de la CPU
- Mayor capacidad de encaminamiento
- Mejor control de la política de enrutamiento

3.2.2 Almacenamiento

Cientos de millones de twits son enviados cada día. Estos son procesados, almacenados, cacheados, servidos y analizados. Con tal masiva cantidad de contenido, se necesita un sistema de almacenamiento a la altura, de hecho, el almacenamiento y la mensajería representa un 45% de la infraestructura de Twitter.

Los equipos de almacenamiento y mensajería proporcionan los siguientes servicios:

- Clusters Hadoop ejecutando HDFS (sistema de ficheros distribuidos de Hadoop)
- Clusters Manhattan para todo el almacenamiento clave-valor de baja latencia
- Clusters MySQL para gráficos
- Blobstore clusters para videos, fotos, archivos...
- Clusters para caché
- Clusters para mensajería
- Bases de datos relacionales (MySQL, PostgreSQL y Vertica)

Los clusters de Hadoop almacenan más de 500 PB divididos en cuatro grupos (tiempo real, procesamiento, backup y cold storage). El mayor de ellos dispone de 10 mil nodos. Se ejecutan 150 mil aplicaciones y se despliegan 130 millones de contenedores al día.

Manhattan es el backend para los twits, mensajes directos, cuentas de Twitter, entre otros. Se ejecutan numerosos clusters para diferentes casos de usos como solo lectura, lectura/escritura, etcétera. Los cluster de solo lectura manejan decenas de millones de peticiones por segundo mientras que un cluster de lectura/escritura manipula millones de peticiones por segundo.

El Gizzard/MySQL legacy basado en cluster fragmentado para el almacenamiento de los gráficos, como por ejemplo Flock (gráfico social), puede manejar picos de hasta decenas de millones de peticiones por segundo, siendo la media de los servidores MySQL entre 35 y 45 mil peticiones por segundo.

Los clusters de Redis y Memcache cachean usuarios, twits, cronologías y más.

MySQL/PostgreSQL son usados donde se necesita una mayor consistencia, manipulando campañas publicitarias y herramientas internas. Vertica es usado como backend para Tableau (visualización de datos interactivos).

En un inicio, todos los datos estaban almacenados en MySQL. Fueron progresando desde una pequeña instancia de base de datos a una de mayor tamaño, siendo finalmente un conjunto de clusters de base de datos. Clonar la información entre las instancias MySQL requiere mucho tiempo, además de trabajo manual. En consecuencia, se introdujo Gizzard (un framework para construir bases de datos distribuidas) en 2010 seguido de FlockDB por encima de los dos anteriores para los gráficos y Snowflake como único servicio de identificación. Ese mismo año, invirtieron en Hadoop, que inicialmente almacenaba los backups de MySQL.

Mientras el tráfico aumentaba exponencialmente necesitaban potenciar el cluster. Para ello, en 2014, lanzaron Manhattan para satisfacer estas nuevas necesidades.

3.2.3 Caché

A pesar de que la caché supone un 3% de la infraestructura, es fundamental para Twitter. Ofrece procción a los discos al guardar datos para que las solicitudes futuras de dichos datos se puedan atender con mayor rapidez y sin necesidad de acceder a él. Twitter usa algunas tecnologías caché como Redis y Twemcache (memcached dedicada multiusuario) a gran escala. Sin embargo, todo el cacheo principal migró a Mesos.

La escalabilidad y el rendimiento son los principales desafíos de la caché. Operan cientos de clusters con un total de 320 millones de paquetes por segundo, enviando más de 120GB/s a los clientes, con objeto de enviar cada respuesta con restricciones de latencia entre el 99.9^o y el 99.99^o percentil incluso en momentos de máximo rendimiento.

Para alcanzar los objetivos en niveles de alta productividad y baja latencia, necesitan medir continuamente el rendimiento de los sistemas y buscar optimizaciones eficientes. Con esto en mente, escribieron `rpc-perf` (una herramienta para benchmarking) para tener un mejor conocimiento de cómo rinde el sistema caché. Como resultado de estos esfuerzos, consiguieron doblar la productividad por servidor sin cambios en la latencia.

Para profundizar sobre esto, se pueden visitar los foros donde los mismos trabajadores comparten estas cuestiones [3]

3.3 Demostración

Como siempre, una de las partes que más tiempo se lleva en un trabajo relacionado con la informática es la de instalación de herramientas que vayamos a utilizar, además de obviamente la decisión previa o a posteriori de cuál elegir.

La primera idea que se nos ocurriría a todos es buscar alguna biblioteca o API (entendamos una forma de conectar con los servicios que Twitter ofrece en la red de forma más automatizada que la interfaz de usuario) oficial o cerrada que tuviera la propia empresa y en un lenguaje conocido por nosotros.

Efectivamente, Twitter tiene una potentísima API en Python, pero que se encuentra restringida al uso de sus propios trabajadores, así que buscamos alternativas.

Fundamentalmente, tratamos dos vías, una biblioteca desarrollada enteramente por un amigo personal, Víctor Castro Serrano, también estudiante de la UGR en el Doble Grado con matemáticas, que dejamos su enlace en GitHub para quien esté interesado. Está muy bien documentada y no usa demasiadas herramientas, por lo que es un proyecto sencillo de comprender. [4]

Pero lamentablemente solemos ejecutar nuestras pruebas de programación en máquinas virtuales y no conseguimos por ningún medio un entorno de Python que nos permitiera trabajar sin errores, algo que ya venía avisado por el compañero.

Tras esto, descubrimos Twint y nos dimos cuenta desde el primer momento que teníamos que conseguir hacerlo funcionar ya que era exactamente lo que buscábamos.

3.3.1 Twint

Twint es una biblioteca sin restricción alguna de uso que nos permite hacer demostraciones simples de scrapping relacionadas con Twitter (mostrar tweets de alguna cuenta, buscar entre los mismos, filtrar por fecha. . .) y posee varias interfaces, lo cual la hace muy flexible y cómoda de usar. La manera más rápida de empezar a interactuar con ella es con una interfaz para la línea de comandos, el comando `twint`. Por ejemplo, tras instalarlo, la orden devuelve los últimos tweets de una determinada cuenta. Además, desde un programa en Python donde añadamos , tenemos muchas opciones para programar o configurar búsquedas y utilizarlas en nuestra aplicación.

3.3.2 Instalación

Podemos instalarlo tanto desde `pip3` como desde su propio repositorio.

Una opción sería lo mostrado en la tabla 3.1, donde lo instalamos desde el repositorio.

Con esto ya tenemos instalado todo lo mencionado previamente y podemos hacer uso de sus herramientas.

Si ejecutamos una prueba con un breve código y una cuenta de ejemplo podemos ver si funciona correctamente. Dejamos esta prueba en la Tabla 3.2

Interfaz Gráfica

Ahora que ya podíamos realizar scrapping desde la terminal, nos dimos cuenta que desde esa misma API habían desarrollado una interfaz gráfica llamada Tweet OSINT Explorer. Desde la wiki de Twint se puede leer información de ella.

3.3.3 Ejemplo

Nuestra idea inicial fue tratar de mostrar a la clase algún tipo de monitorización a tiempo real del tráfico de Twitter, por ejemplo, en alguna zona concreta como Granada. No sabemos si por falta de herramientas o habilidad, pero al final nos hemos quedado mucho más cortos.

Aunque conseguimos ejecutar la interfaz gráfica de Twint, no fuimos capaces de ponerla a funcionar por fallos, creemos, en el almacenamiento en forma de Base de Datos de los datos extraídos por la aplicación, pero sí hemos sido capaces de hacer algo de Scrapping desde la terminal.

Al ejecutar el código de ejemplo de la foto 3.2, lo que hacemos es configurar una búsqueda para luego llevarla a cabo, y los tweets que se encuentre Twint que contengan la palabra (también se pueden modificar los rangos de fecha donde buscar, por ejemplo si dejamos la palabra vacía lo hemos configurado para filtrar los últimos) y los muestra por la terminal con su ID, fecha de publicación y texto.

Table 3.1: Instalación de Twint

```
git clone https://github.com/twintproject/twint.git
cd twint
pip3 install . -r requirements.txt
```



```
paco@DESKTOP-7CVMK3I:/mnt/c/Users/Paco/MEGA/segundo_cuatri/SWAP/trabajo$ /bin/python3 /mnt/c/Users/Paco/MEGA/segundo_cuatri/SWAP/trabajo/twint/prueba.py
/usr/lib/python3/dist-packages/requests/_init_.py:89: RequestsDependencyWarning: urllib3 (1.26.9) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported version"
1534227879411728386 2022-06-07 19:36:45 +0200 <GitteTitter> Ever wonder what the recommendations are to mitigate OWASP API Security Top 10 threats using API Management? Check out t
his doc the FastTrack for Azure team contributed to: https://t.co/xd1OqEpB1d
1463398189407645700 2021-11-24 07:44:51 +0200 <GitteTitter> My colleague @xstof put some awesome work into this: https://t.co/zDB1l9Jntd Ever wonder what's going on in your app u
w m gateway waf? This workbook is an excellent help! #Azure
1019588409910673408 2018-07-18 16:23:14 +0200 <GitteTitter> @proximus and to top that, appointment next week, which got already booked as a backup will AGAIN send the wrong team
to do the connection. That's 3 MONTHS guys of you not being able to do your job. And then you wonder why we get frustrated when calling you?
999618145299070978 2018-05-24 13:48:32 +0200 <GitteTitter> No wonder @TechMike2kx is on top of the @techoramaBE tweet leaderboard, did you see the battery he is carrying for his ph
one??? My phone is almost dead halfway through the day 😊 I ll never beat him
597644759407148864 2015-05-11 08:09:49 +0200 <GitteTitter> @techoramaBE wow, that looks aaawwwweeeessssomme!!! No wonder you guys needed 3 trucks...
400693175603699712 2013-11-13 19:34:32 +0200 <GitteTitter> No wonder they are being hacked RT @davybrion: #belgacom security fail http://t.co/905fewbJ8c
203158191935209472 2012-05-17 18:21:02 +0200 <GitteTitter> I wonder why #dlvrit set all my feeds as inactive? I didn't change that now, didn't I?
[!] No more datal Scrapping will stop now.
found 0 deleted tweets in this search.
```

Figure 3.3: Scrapping en la Terminal

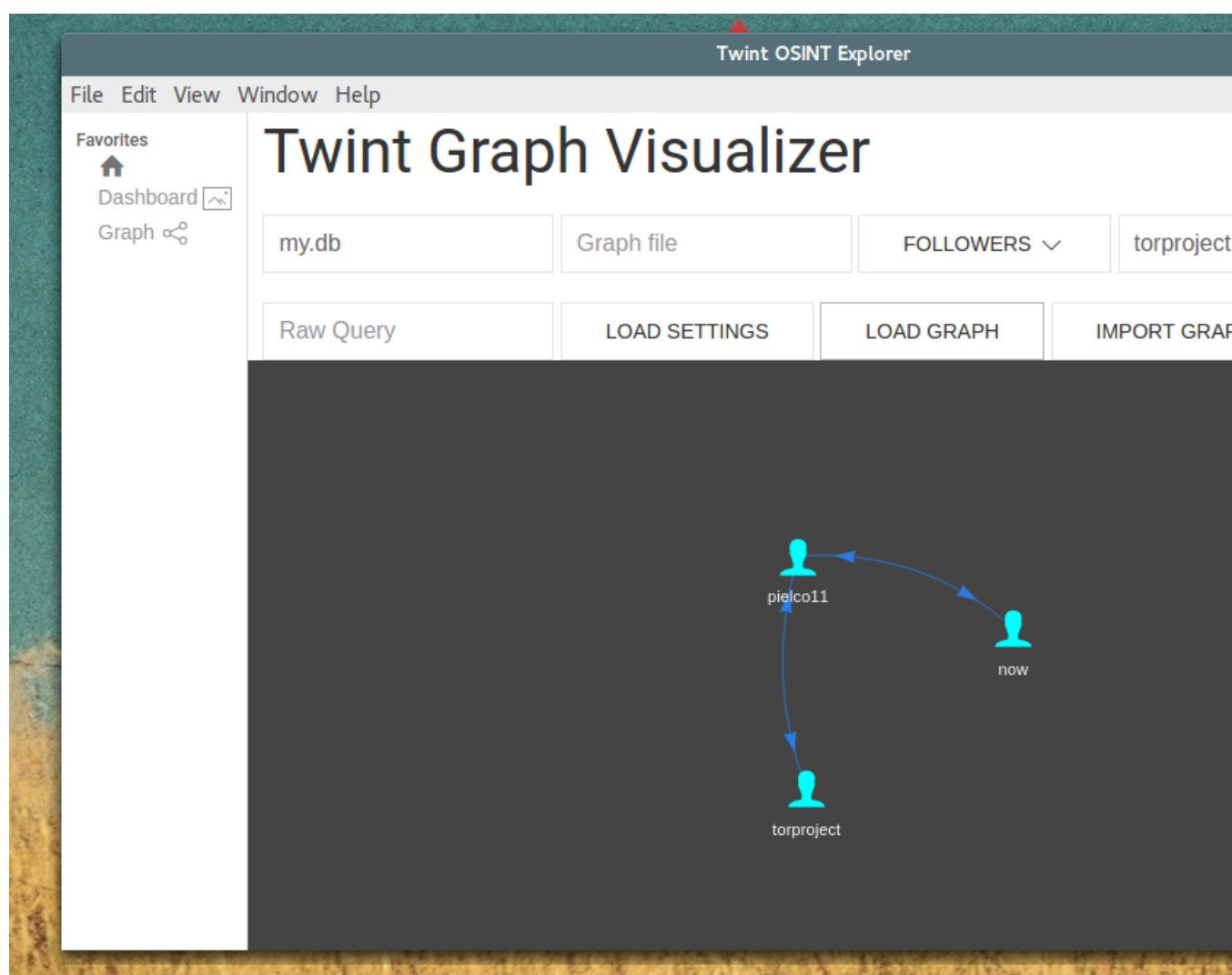


Figure 3.4: Interfaz Gráfica original

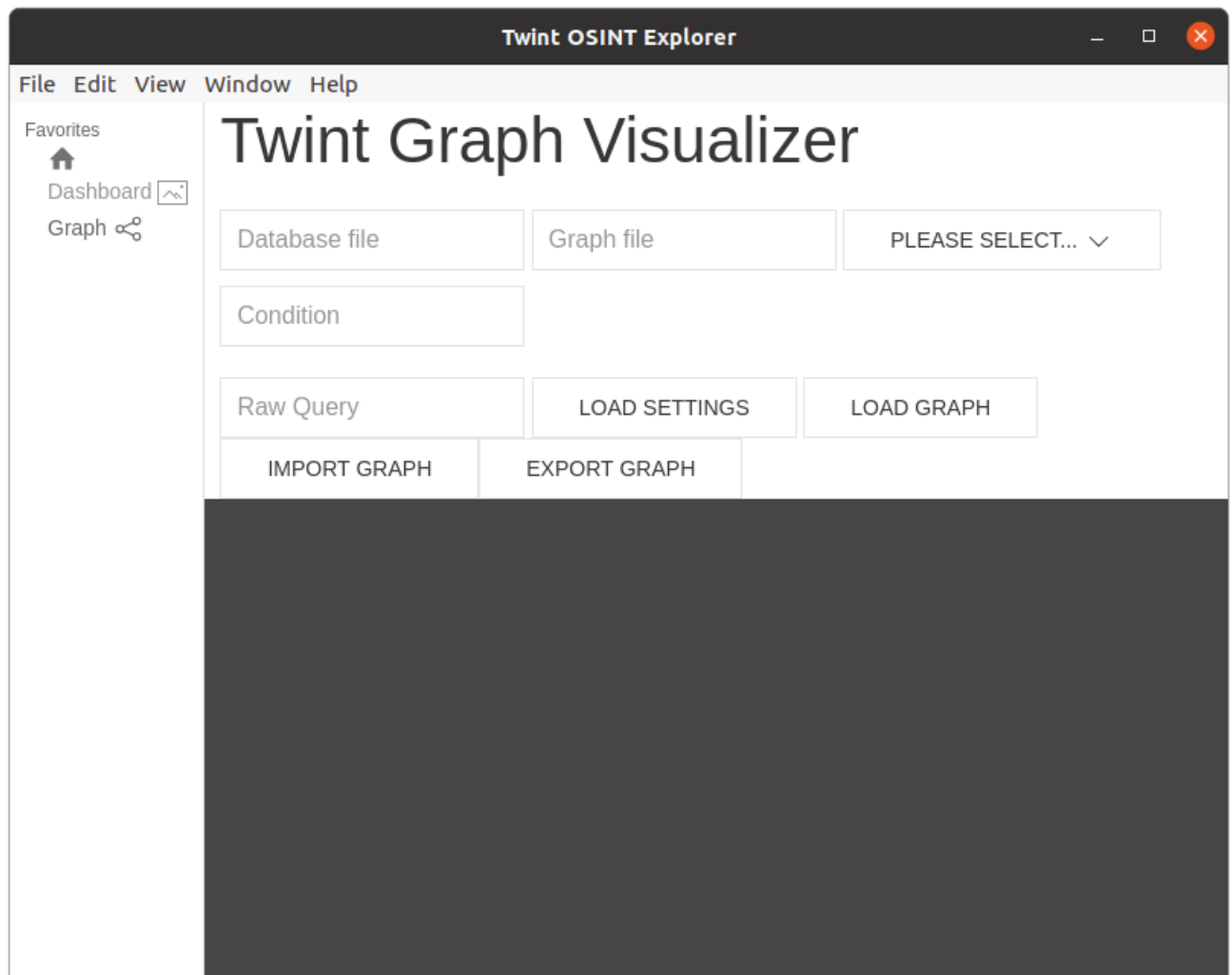


Figure 3.5: Interfaz Gráfica fallida

Table 3.2: Código de Ejemplo

```
import twint

c = twint.Config()
print("Introduce porfa una cuenta de Twitter: \t")
c.Username = input()
print("Quieres buscar una palabra?: Si no es asi mostraremos los ultimos tweets.
[y/n]\t")
n = input()
if n == 'y':
    c.Search = input()

# Run

twint.run.Search(c)
```

Chapter 4

Conclusiones

4.1 Reflexión sobre la descentralización

Como hemos visto a lo largo de esta presentación, Twitter es un complejo entramado, pero a la vez éste muy concreto y controlado, de servidores, trabajo humano, y organización meticulosa de su propia producción y mantenimiento, a cargo en este caso de una empresa privada, pero cuyo objetivo es presentarse al mundo como una plataforma de comunicación completamente horizontal donde la máxima a cumplir es. "Escribes lo que quieres, lees lo que quieres, te lee quien tú quieres y sólo te llega de quien tú quieras".

Toda la discusión política, sociológica y también ingenieril entorno a esta cuestión es lo que se conoce como descentralización o democratización de la comunicación, algo que desde el primer día y de forma automática por su propia naturaleza, se nos prometió que traerían internet y las redes sociales, frente a la mucha menor flexibilidad, contenidos mascados y coste de mantenimiento de la televisión, la radio, y no digamos la prensa escrita.

La gran pregunta es. ¿Es, en su despliegue y uso diario, esto efectivamente así?

Es evidente que la extensión de esta discusión es algo que se escapa a las capacidades de este trabajo. Hay una infinidad escrita sobre este tema, de la cual dejamos una mínima bibliografía al respecto para quien quiera profundizar, pero una cosa innegable es que la voluntad (el mismo verbo querer) es algo que está sujeto a muchas más variables que las de nuestro propio ser y actuar, y lo que es más peligroso, todo esto las máquinas lo saben calcular y manipular muy bien.

Tanto para profundizar en el capítulo 2, es decir, la magnitud de Twitter Y Facebook no como inocentes plataformas de interacción personal, sino como motores, barqueros, auténticas toneladoras masivas de opinión mediática (Aunque el término más

extendido y usado es opinión pública, remito aquí a la reflexión del filósofo español Jose Luis Sampedro, quien diferenciaba entre opinión pública, lo que para él era resultado mediado, democratizado y reflexionado de una comunidad concreta como consciente colectivo, y la opinión mediática, como todos los significantes y dogmas vacíos que desde una élite permiten tener a una masa controlada), como en las dos reflexiones que dejamos por aquí, no podemos dejar de recomendar un libro que explora tanto ejemplos concretos de manipulación mediática como todos estos debates más en el aire entorno a la voluntad y la capacidad de las redes sociales de hacernos sentir libres. [4]

4.2 Cómo evitar convertirnos en Eichmann

“Únicamente la pura y simple irreflexión (...) fue lo que le predispuso a convertirse en el mayor criminal de su tiempo. No era estupidez, sino una curiosa, y verdaderamente auténtica, incapacidad para pensar”.

Hannah Arendt, sobre Adolf Eichmann, una de las mayores cabezas de la jerarquía nazi que pasó por los juicios de Nüremberg. [5]

Un ejemplo bastante grotesco pero muy ilustrativo de cómo la irreflexión política y el dejarnos llevar por la corriente del, por ejemplo, avance tecnológico. Hannah Arendt fue una famosa periodista y escritora judía, víctima, no de muerte pero sí de reclusión, de los campos de exterminio nazis, la cual cubrió muy de cerca los juicios de Nüremberg, y la cual sostuvo firmemente que la imagen maligna, sádica y esencialmente malvada de estas desgracias para la humanidad son caricaturas, no se corresponden con la realidad. Una imagen, parece ser por la gente que lo estudió a fondo, mucho mas fiel a la realidad de estos generales y perfiles técnicos era la de fríos funcionarios obsesionados con cumplir de forma eficiente su labor para con la patria.

Como dijo Malcolm X, más peligrosos son los que miramos para otro lado que los que hacen, porque somos la mayoría. Por resumirlo en una frase, “Yo hacía mi trabajo” fue sin duda alguna la frase más escuchada en todo este proceso judicial.

Esto nos lleva a obligarnos a hacer un mínimo llamado a navegantes. Tanto como usuarios como posibles futuros ingenieros que sostengan todo esto, tenemos la responsabilidad de cuidar la honestidad, sinceridad y principios que creemos rigen a la ciencia, rigen a nuestra disciplina, y que muchas veces se ven distorsionados en nombre del cortoplacismo, las ganancias y nuestros miedos, lo que nos lleva a utilizar armas de este calibre (sí, armas, con todas las letras) de forma mucho más devastadora que constructiva.

Todo esto que estamos viviendo sólo es posible gracias a nuestros teclados, somos nosotros quienes creamos estos monstruos que nunca fueron meras máquinas, meros códigos. Tenemos un poder, por tanto tenemos una responsabilidad.

No queremos dejar este mensaje en clave, como unos cuantos de tiros vacíos al aire. Las redes sociales son una herramienta fundamental para quienes viven de esparcir odio, miedo y hacer negocio con ellos.

Twitter, que no son sus servidores, es la voz de millones de personas, ya está en proceso de ser comprada por uno de los mayores acaparadores de trabajo humano a nivel mundial, el cual no sólo hizo su fortuna a costa de la vida y sufrimiento de millones de personas originalmente en sus minas de diamantes y posteriormente a todo lo ancho y largo del globo, sino que ha mostrado de forma completamente descarada su intención de convertir esta red social en un altavoz mucho más grosero de su fantasía neoliberal.

El caso concreto de este señor al final es sólo un ejemplo más de lo que tratamos por ejemplo de desarrollar en el punto 2, el peligro que puede tener una plataforma masiva de opinión en malas manos es algo que la historia nos tiene más que enseñado, pero tampoco hemos tenido nunca las capacidades comunicativas, de vigilancia y de control masivo de la opinión que tenemos hoy en día.

Esto es un doble mensaje. Seamos conscientes de lo que podemos hacer con la informática. Entendido este mensaje como motivación para lo positivo y posible, y como advertencia para lo dañino y, probablemente, lo que está por venir.

Bibliography

- [1] Wikipedia. <https://es.m.wikipedia.org/wiki/Twitter>.
- [2] 33 Twitter Stats That Matter to Marketers in 2022.
<https://blog.hootsuite.com/twitter-statistics/>.
- [3] Twitter's Infrastructure. https://blog.twitter.com/engineering/en_us/topics/infrastructure/2017/infrastructure-behind-twitter-scale.
- [4] En el Acuario de Facebook. El Resistible Ascenso del Anarco-Capitalismo.
<http://imillahacker.sdf.org/en-el-acuario-de-facebook.pdf>.
- [5] Hannah Arendt y la banalización del mal.
https://verne.elpais.com/verne/2017/03/23/articulo/1490255737_690085.html.

Appendix A

Herramientas utilizadas

Demostracion práctica de Scrapping a Twitter

1. Twint. <https://github.com/twintproject/twint>
2. Twint OSINT Explorer. <https://github.com/twintproject/twint/wiki/Graph>

Elaboración de la Memoria

1. Overleaf. <https://www.overleaf.com/>

Trackeo del Tiempo Empleado

1. ClickUp. <https://clickup.com/>