

PTT: Python Twitter Tools by Mike Verdone

Desde hace tiempo venimos utilizando nuestro propio robot (o "bot" como es la moda ahora mentarlos) para promocionar esta nuestra y vuestra humilde página web por [Twitter](#) y la verdad es que no habíamos prestado mayor atención al trabajo hecho con la librería que permite hacerlo. Utilizamos Twitter por su simplicidad y el soporte que tenía por medio de mensajes de texto "[Short Message Service SMS](#)" a los móviles celulares: sencillo y elegante. Pero mejor estructuraremos esta entrada para explicaros (y compartir) nuestro conocimiento acerca del tema ¡Vengan con nosotros en este nuevo registro en la bitácora!

Breve historia del Twitter.

Que no os vamos a aburrir mucho con esto, pero por algo se empieza. Como preludeo consideramos justo reconocer que los creadores de Twitter *debe haberse inspirado en algún momento en el famoso protocolo de comunicaciones llamado "Internet Relay Chat IRC"* el cual fue creado en 1988 por Jarkko Oikarinen y que tuvo su apogeo en el año 2004 con el popular juego de acción QuakeNet (240 mil usuarios -solo esa red, a nivel mundial hay muchas otras redes de usuarios-). Desde ese año el **IRC** ha ido declinando en el número de usuarios y llegado el año 2006 [Jack Dorsey](#), [Noah Glass](#), [Biz Stone](#) y [Evan Williams](#) fundan Twitter y en seis años consiguen 100 millones de usuarios que "tuiteaban" 340 millones de mensajes diarios. Inicialmente *no fue pensada como una red social sino como un **blog** que utilizaría los mensajes de texto por teléfono móvil y debido a su limitación de 140 caracteres se convirtió de facto en un microblogging.*

Nosotros usamos Twitter desde el año 2011, justo un año después de haber comenzado a utilizar la tecnología **OAuth** -desarrollada en las entrañas de Twitter desde el mismo año de su nacimiento en el 2006- para ganar seguridad y libertad de programación a aplicaciones de terceros con una "Application Programming Interface **API**". Acá en Venezuela el operador Movistar (antes conocida como Telcel) daba soporte para enviar mensajes al Twitter por [medio de mensajes de texto al número 89338 \(TWEET\)](#) y funcionaba de maravillas pero actualmente no prestan el servicio, imaginamos que por falta de arreglo económico entre ambas empresas, recordad que nadie trabaja gratis (sin embargo se ganan una buena pasta vendiendonos megabytes de datos que usamos en Twitter). Otro aspecto del impacto de Twitter en nuestra vida diaria es la inclusión del verbo "tuitear" en el [Diccionario Castellano de la Real Academia Española](#) así que si en adelante *no lo incluimos entre comillas* pues ya veis que es "legal" usarlo así ya que es un anglicismo formalmente reconocido (¡qué fuertes son los *Poderes Fácticos* del siglo XXI!).

<https://twitter.com/ks7000/status/865149487999852544>

Para aquellos que queréis saber más de la historia y funcionamiento del Twitter -en inglés, publicado en el año 2016 por Amelia Norton y creado por Andreas Schreiber en 2009- [navegad hasta este sitio donde publica unas diapositivas digitales](#) donde la primera mitad explica todo esto

y más adelante entra en materia más profunda; está profusamente ilustrado y va paso a paso:

[Twittering with Python Python libraries for the Twitter API: Overview and applications](#)

¿Que cómo era el Twitter antes del **OAuth**? Podéis leer [este artículo publicado](#) en el año 2009 donde dan cuenta de como "tuitear" por medio del comando **curl**:

```
curl -u usuario:contraseña -d status="Tu mensaje" http://twitter.com/statuses/update.xml
```

¡Increíble! Fijaos que Twitter ni siquiera era una página segura (**https**) y el usuario y contraseña viajaban libres por la red *¡eran otros tiempos, éramos más confiados (y con menos preocupaciones)!* (Gracias [Nacho](#) por tu explicación, como veís 8 años después tus palabras aún resuenan en el ciberespacio). Acá [otro artículo comentan como](#), incluso, como "tuitear" con **PHP** y **curl** y analizar la respuesta en formato tanto en formato **XML** como **JSON**.

Pero he aquí que ya el Sr. Nacho explicaba como utilizar el lenguaje Python con las librerías del sr. Mike Verdone creador de la "[Python Twitter Tools PTT](#)", un software de código abierto y apoyado plenamente por la [Fundación Python](#), y éste es el artículo principal de esta entrada.

"Python Twitter Tools PTT" por Mike Verdone.

Funcionamiento básico de la red Twitter.

Twitter siempre ha utilizado el software libre para lograr sus propósitos y en un principio utilizaba **MySql** como manejador de base de datos, pero pronto le quedó pequeña para sus propósitos.



Lo que aquí expresamos es a grandes rasgos, no esperéis una explicación al detal de la tecnología usada por Twitter; lo expresamos a nuestra forma y entendimiento y lo plasmamos de la manera más simple posible y si cometemos algún error hacednos saber por nuestra cuenta Twitter @ks7000.

Ahora Twitter utiliza **Ruby on Rails** en servidores **Java** con "Content Delivery Network **CDN**" distribuidos en todo el mundo y cada identificador de mensaje es único y tiene información acerca de su geolocalización para poder manejar la gran cantidad de información que se genera segundo a segundo. En el acortador de direcciones ubicado en <http://t.co> aparte de ahorrar espacio en los mensajes también revisa y detiene los enlaces maliciosos (páginas falsas, páginas con código dañino, etc) y en el servicio <https://twimg.com/> permite almacenar las imágenes que acompañan el mensaje. También han incluido soporte para **emoticons** que en realidad es soporte para todos los caracteres que soporta el **UNICODE**, así como soporte para "gifs animados" (que en realidad son convertidos a formato MP4), así como transmisión de vídeo en vivo por **Periscope** y a futuro vienen muchos cambios más para mantener con vida esta red. Uno de los más redituables son los "**Twitter Ads**" que es publicidad personalizada ¡**repetimos, nadie trabaja gratis!**

Twitter y el software desarrollado por terceros.

El equipo de Twitter creó y desarrolló de manera abierta la tecnología **OAuth** para que aplicaciones de terceros puedan acceder a ciertas características de nuestra cuenta en Twitter *sin necesidad de darle nuestra contraseña de cuenta*. Esto se logra de la siguiente manera:

- Un equipo de desarrolladores se unen y crean una cuenta en Twitter. Deben colocar, al menos, un número de teléfono celular para recibir por mensaje de texto la confirmación de apertura de la cuenta deseada, así es que al menos es una persona real a quien le pertenece el número telefónico y **es responsable de dicha cuenta**.
- Si es un proyecto muy grande el siguiente paso más probable es que contacten a Twitter -en San Francisco si viven y trabajan en los Estados Unidos de América o en Dublín, Irlanda, para el resto de los países del mundo- para solicitar una **cuenta verificada** (las cuentas que son identificadas con un símbolo azul y *tooltip* que las anuncia). Twitter recomienda que para proyectos pequeños -que por supuesto tienen su propio dominio web- publiquen en su página web la cuenta twitter oficial -tal como nosotros hacemos acá-. Esta norma rige también para personalidades, artistas, políticos, páginas de los gobiernos, etc.
- Una vez hayan configurado sus datos en Twitter hay un apartado que permite crear dos *token* llamados "Consumer Key" (Clave de Consumidor) y "Consumer Secret" (Secreto de Consumidor) que deben ser almacenados de forma segura por los desarrolladores de la futura aplicación (en PTT está almacenada en [donde alojan el código fuente](#), GitHub). Un *token* no es más que una larga secuencia alfanumérica de caracteres regidos por fórmulas matemáticas que permiten identificar de manera unívoca un objeto de software.
- Una vez obtenida la Clave del Consumidor la aplicación debe proveer un enlace hacia Twitter con la siguiente orden: «https://api.twitter.com/oauth/authorize?oauth_token=Consumer_Key». Si no hemos iniciado sesión en Twitter introduciremos nuestro usuario y contraseña **la cual se la estamos entregando a Twitter por medio de una página web segura https, no a la aplicación en sí**.

- Una vez entramos en Twitter nos redirigirá a una página donde explica qué privilegios solicita la aplicación y y botón para aprobarla. **Muy importante que leamos bien que derechos de acceso otorgamos, si estamos de acuerdo le damos aceptar.**
- Una vez que aprobamos Twitter genera a su vez otros dos token que son enviados a la aplicación que desea acceder a ciertos elementos de nuestra cuenta. Si la aplicación web está instalada en nuestro ordenador o dispositivo móvil por medio de un PIN que nos muestra Twitter por pantalla lo copiaremos y suministraremos a la aplicación para que ella obtenga por sus propios medios los dos token *personalizados para nosotros* y procederá a guardar en nuestra máquina. Si es una aplicación web se redirigirá la página y obtendrá ella misma los dos token necesarios.
- El punto anterior funciona de dos maneras diferentes: en la aplicación instalada en nuestro ordenador o celular los token se almacenarán en nuestros equipos. **Si es una aplicación web los token se almacenarán en el o los servidores de la aplicación.** En el primer caso nosotros usaremos los 2 token personalizados -público y privado-, en el segundo caso será la aplicación de manera autónoma quien los usará -y nosotros nunca sabremos los valores de los token generados, a diferencia del primer método-.
- **En cualquier momento podemos eliminar el acceso a la aplicación si nos dirigimos a «<https://twitter.com/settings/applications>» y pulsamos el botón de revocar: esto simplemente hace que Twitter inactive los dos token que generamos y así negará el acceso a las partes de nuestra cuenta que habíamos autorizado.**
- Las partes de nuestras cuentas que podemos otorgar acceso son las siguientes (caso **PTT**):
 - Acceder a nuestra línea de tiempo (los tuits de las cuentas que seguimos).
 - Enviar mensajes públicos y/o privados con nuestra cuenta.
 - Permitir leer nuestros datos: avatar, nombre e incluso nuestro correo electrónico.
 - Seguir y/o dejar de seguir otras cuentas en Twitter.
- Se ha hecho ya costumbre el crear estas aplicaciones simplemente para permitirnos emitir comentarios en las páginas web o por medio de varias páginas web que se afilien a esa aplicación: **Disqus** es un buen ejemplo de ello, son aplicaciones con una API basadas a su vez en las herramienta API de Twitter con el esquema de autorización **OAuth**.
- Aunque logren acceder a nuestro ordenador o si nos roban o se pierde nuestro teléfono celular **siempre podemos iniciar sesión en Twitter en cualquier otro equipo confiable y revocar el acceso a dichas aplicaciones: eso inutiliza los token personalizados que fueron generados.**
- De igual manera si el sitio que aloja la aplicación web que autorizamos les roban dichos token personalizados hacemos lo mismo del punto anterior. [Un ejemplo de pérdida de datos fue lo sucedido con TweetGif.](#)

Instalando PTT.

Ya hemos publicado [como instalar el lenguaje Python](#) en nuestras máquinas, así que aquí solo escribiremos como instalar las utilerías desarrolladas por el sr. Mike Verdone. Debemos entonces

referiros a otro de nuestros posts donde explicamos la [clasificación de las funciones en Python](#): las que provienen de terceros deben instalarse con PIP. Hacemos notar por experiencia propia que GNU/Linux Debian Jessie 8.8 *no trae instalado por defecto el **pip***. No hay problema, lo que debemos hacer es escribir -con derechos de administrador **root** ganados con **su**-:

```
apt-get install python-pip
```

Tras lo cual no solicitaré que introduzcamos el DVD N° 1 (el 2° y el 3° contienen aún mucho más software pero menos utilizados) por lo que nos ahorramos el conectarnos a internet (por eso decimos que viene "instalado por defecto" en las distribuciones GNU/Linux). Una vez tengamos instalado el **pip** debemos lanzar la siguiente instrucción:

```
sudo -H pip install twitter
```

El parámetro **-H** en **sudo** (o, si gustan, **--set-home**) especifica que la política de seguridad sea fijada en el usuario actual que ejecuta el comando a efectos de registrar una entrada en la base de datos de contraseñas del usuario, aunque es la opción por defecto hay que recalcarla cuando [instalemos cualquier otro paquete](#) con **PIP**. Acá podéis ver el resultado del comando, nos instala la última versión disponible a la fecha, la 1.17.1:

Comentarios de Mike Verdone sobre la versión 1.9.1

En enero de 2013 anunció la liberación de la versión 1.9.1 con las nuevas características para el momento pero lo que nos llamó la atención fue el siguiente comentario [en el último párrafo en su blog](#):

At this point I rarely do dev work on Python Twitter Tools. I merely evaluate and merge the pull requests submitted by other talented developers. A great big thank you to all of them! Their names reside safely in the Git project history, for all eternity.

Que traducido al castellano sería más o menos lo siguiente:

En este momento raramente desarrollo trabajo en "Python Twitter Tools". Yo simplemente evaluo y uno las propuestas enviadas por otros talentosos programadores. ¡Un grandioso agradecimiento a todos y cada uno de ustedes! Sus nombres reposarán de forma segura en la historia del proyecto (en formato Git), por toda la eternidad.

Esto, para nosotros, es poesía para nuestros oídos y una muy buena apología al Software Libre, ¡Libertad!

Configurando PTT por primera vez.

El **Python Twitter Tools** viene conformado por dos componentes: las librerías en si mismas -que podremos usar si tenemos nuestras propios **token** otorgados por Twitter a través de nuestra cuenta allí creada- y un archivo ejecutable llamado **twitter** que podremos llamar por medio de una ventana terminal.

Primero explicaremos como usar el ejecutable compilado con ayuda los **token** pertenecientes a Mike Verdone, es decir, vamos a autorizar *única y exclusivamente al programa **twitter** que instalamos por medio de **PIP** para que interactue con nuestra cuenta en Twitter.*

Lanzamos una ventana terminal y escribimos simplemente "**twitter authorize**" lo cual hará que dicho software lance un navegador por el siguiente enlace:

Notad que la página web es segura, "candado verde" (si no hemos iniciado sesión e Twitter, pues ingresad y la página se redirige) y que nos explica brevemente que accederá a leer los tuits que publican a quienes seguimos -esto es conocido como *línea de tiempo* o "*timeline*" en inglés-, ver la lista de quienes seguimos -y el permiso de seguir nuevas cuentas-, **actualizar nuestro perfil cosa que nos sorprende ya que no vemos opciones documentadas para hacer esto (cambiar nuestro nombre, avatar, etc) lo cual investigaremos en detalle para qué quiere tener acceso a este renglón**, y por último la habilidad de tuitear por nosotros de una manera automatizada (vamos a comenzar por esto último, enviar un mensaje al ciberespacio).

Una vez hallamos autorizado a la aplicación en la página del [Twitter nos mostrará](#) un **PIN** el cual fue pensado para aplicaciones que no tienen una interfaz web -léase aplicaciones de líneas de comandos- al cual el **PTT** llama con un argumento distinto al comando **oauth_callback** (toma el valor **oob** "out-of-band"). Dicho PIN debemos introducirlo en la entrada que requiere PTT tras lo

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

cual se comunicará -vía **curl**, [imaginamos](#)- con Twitter para obtener nuestros **token personalizados** que serán almacenados en la siguiente ubicación: `~/.twitter_oauth` (un archivo oculto en nuestra carpeta personal "home").

Una vez hallamos configurado adecuadamente la aplicación, no tenemos mas que escribir nuestro primer mensaje por medio de PTT: "**twitter set**" tras lo cual pedirá que ingresemos nuestro mensaje, presionamos intro y listo, se fue al universo del *microblogging* nuestra prueba:

<https://twitter.com/ks7000/status/866280372014854145>

Usando PTT (agradecimiento incluido):

<https://twitter.com/ks7000/status/866024018247331840>

Este mensaje lo enviamos el día 20 de mayo de 2017, cuando publicamos esta entrada (nosotros publicamos cuando consideramos tenemos algo suficientemente redactado, luego lo refinamos y con el paso del tiempo lo mantenemos actualizadas cada una de nuestras entradas en nuestro blog). En ese mensaje recibimos un "me gusta" por parte del usuario [@sixohsix](#) (ya comentamos que toda aplicación web de Twitter debe tener, por supuesto, una cuenta en Twitter).

Revisando cada una de las acciones de PTT.

Seguimos revisando la aplicación compilada **twitter** por la línea de comando, luego veremos el uso de las utilerías y programaremos. Para PTT es necesario tener instalado **argparse** [-que ya os hemos explicado como funciona-](#) y es la que permite introducir "**twitter -h**" para saber sus acciones y preferencias. De número uno la opción por defecto es la que estudiaremos.

"twitter" ó "twitter friends".

Al escribir "**twitter**" estamos invocando en realidad "**twitter friends**": esto devuelve por pantalla los mensajes o tuits -incluyendo los retuits- de las cuenta a las cuales seguimos. Recordad que vuestra velocidad de conexión al internet influirá en la respuesta, paciencia que puede tomar cierto tiempo.

Como nosotros vamos de programación, y por ende de tareas automatizadas, el primer uso que le vamos a dar es filtrar resultados con **grep** mediante el comando tubería. Lo que vamos a buscar son los tuits que contengan la palabra -valga la publicidad- "Samsung" utilizando la salida **stdout** al comando de filtrado **grep** de la siguiente manera:

Primero notad que debemos acostumbrarnos a entrecomillar siempre la clave a buscar ya que Twitter utiliza etiquetas cuyo prefijo es el caracter numeral "#" el cual significa comentario en el

shell bash.**"twitter help" ó "twitter -h".**

Por supuesto esta opción es insoslayable: retribuye las opciones documentadas o declaradas (como tenemos acceso al código fuente podremos estudiar y buscar opciones *no documentadas debidamente*).

"twitter follow" y "twitter leave".

Recordemos que le dimos permiso a la aplicación de seguir o dejar de seguir otra cuenta en Twitter, así que para seguir una cuenta escribimos **"twitter follow @nombre_de_la_cuenta"** y para dejar de seguirla **"twitter leave @nombre_de_la_cuenta"**. Nosotros la probamos con **@cementerio** una cuenta argentina que se dedica a emitir día a día el fallecimiento tanto de personajes famosos a nivel mundial como a nivel local, mirad la imagen:

"twitter list" y "twitter mylist".

Una lista en Twitter simplemente es un conjunto de cuentas que agrupamos según un criterio definido, en nuestro caso clasificamos a los desarrolladores de software por cada país del mundo. Las listas pueden ser públicas como privadas así que si vemos alguna cuenta que NO sigue a nadie no necesariamente quiere decir que sea una cuenta inactiva, tal vez tenga una lista privada para recibir información de manera discreta. Con **"twitter list @nombre_de_usuario"** podremos obtener los nombres de las listas **públicas** de un usuario de Twitter, *lo estemos siguiendo o no (por supuesto, si no lo seguimos y su perfil es público)*. De lo contrario, si la cuenta está "protegida" (perfil privado) devolverá el error 34 con el mensaje **"Sorry, that page does not exist."**; esto es así por la máxima privacidad al usuario: cuando le preguntamos al Twitter éste lo niega, incluso su existencia (es decir, no anuncia que la cuenta es privada). De esta opción hicimos pruebas pero no publicamos las cuentas protegidas que consultamos para conservar la privacidad *pero hemos de decirles que funciona a la perfección (si acaso podríamos hacer un "fork" para el tratamiento de mensajes de error, -vamos a ver hasta dónde llegamos-)*. Con la opción **mylist** y como son nuestras propias listas, obtendremos tanto listas públicas como privadas.

"twitter replies".

Con este comando nos retribuye las últimas respuestas a nuestros tuits -primero los más antiguos-, valga decir que son mensajes que se hilan en una nueva conversación o en una existente. En Twitter esto es como un árbol con la raíz invertida: si solo son dos usuarios es una línea directa de conversación pero si varios usuarios intervienen entonces se convierte en un árbol. Es de notar, por supuesto, que si uno opta por silenciar "mute" una conversación, ya no recibiremos aviso alguno de la misma.

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

"twitter search".

Al nosotros utilizar esta opción de inmediato nos devolvió error (y no, no es por incluir el "octothorpe" o símbolo numeral que tiene un significado especial en Python). El [error está denunciado en este enlace](#) y en la medida de lo posible estamos revisando el código para ver si podemos ayudar en algo.

"twitter shell".

Todos los argumentos que acabamos de ver los ejecutamos directamente de la consola de nuestra distribución GNU/Linux pero si usamos "**twitter.shell**" entraremos en el *prompt* del propio PTT: así podremos escribir "**set 'mensaje'**", presionar intro y enviamos nuestro mensaje. Es decir, nos ahorramos la palabra "twitter" en cada llamado.

<https://twitter.com/ks7000/status/866353298969853952>

Para salir del *shell* del **PTT** solo debemos pulsar la tecla CTRL+D tras lo cual preguntará la aplicación "**You really want to Leave (Y/n)?** " y pulsamos la tecla "Y", presionamos intro y "salimos" al *shell* de nuestro ordenador.

"twitter rate".

Así como hay un límite máximo de 100 tuits por hora para cada cuenta en Twitter (si emitimos 100 tuits en 100 segundos -un minuto y cuarenta segundos- estaremos bloqueados por una hora más o menos) también hay unos límites en las tareas automatizadas. Cada límite está establecido para cada uno de los métodos establecidos en la **API** y sería largo de enumerar aquí; algo relevante es que dichos valores se "reinician" cada 900 segundos, osea cada 30 minutos. Como veís, es muy similar al límite que comentamos al principio, solo que el bloqueo dura media hora en vez de una hora completa.

"twitter repl".

Con este comando entraremos en una ventana normal de Python pero con un mensaje de bienvenida que es más que elocuente:

En el anuncio nos indica que por medio del objeto "**twitter**" (declarado y conectado por un *script* o guion previo) podremos interactuar con la REST API lo cual consideramos algo avanzado y creemos con propósitos de depuración a bajo nivel. Para salir usamos CTRL+D, recuerden que estamos en Python terminal. Por ahora no le encontramos mayor utilidad y lo dejamos tranquilo para pasar al otro comando que nos parece tremendamente útil.

"twitter pyprompt"

Esto si lo consideramos muy útil y sirve como paso previo antes de comenzar nuestros ejemplos y pruebas con las librerías en sí mismas: si con **twitter shell** podemos escribir solo los comandos (set, rate, etc.) con este comando **tendremos un objeto "twitter" declarado y conectado con el cual podremos programar aspectos más avanzados, como por ejemplo tuitear incluyendo imágenes. La diferencia principal es que estaremos conectados con los token que autorizamos (¿recuerda "twitter authorize"?) y podemos programar a nuestras anchas, en cambio al lanzar Pythom e importar la librería debemos nosotros mismos suministrar los 4 elementos completos:**

- **token** (del lado de la aplicación registrada en Twitter).
- **token_secret** (del lado de la aplicación registrada en Twitter **¡estricto cuidado NO difundir!**).
- **consumer_key** (del lado nuestro, los token que solicitamos y autorizamos).
- **consumer_secret** (del lado nuestro, los token que solicitamos y autorizamos **¡estricto cuidado NO difundir!**).



Si a la final metemos la pata y hacemos pública nuestro token privado de cliente, pues iniciamos sesión en Twitter y revocamos el acceso a la aplicación. Si somos desarrolladores y hacemos pública nuestro token privado de aplicación (tal como hizo **PTT**) pues vamos con la cuenta twitter con que creamos la aplicación y le revocamos el acceso. Como veís la seguridad está en parte garantizada y Twitter feliz de la vida delegando esas responsabilidades en nosotros: *ellos ponen la maquinaria, nosotros [la mano de obra](#).*

Preferencias u *opciones* de PTT.

De las acciones que estudiamos anteriormente, estas preferencias son compatibles con todas ellas, comencemos por "--refresh".

Opción "-r" ó "--refresh".

Cuando utilizamos el comando **grep** para filtrar resultados -de una manera "rudimentaria" bien pensamos agregar el comando al **crontab** de nuestro sistema GNU/Linux pero no es necesario

llegar a tanto. Para ello contamos con la opción **"-r"** ó **"--refresh"** para que el comando que corriendo indefinidamente hasta que presiones **CTRL+C** para interrumpir el proceso. *De manera predeterminada el intervalo es de 5 minutos (300 segundos)* pero con la opción **"-R"** ó **"--refresh-rate"** podemos especificarle un valor en segundos que querramos -e incluso programar para que sea un valor aleatorio-. De las pruebas que hicimos nos dimos cuenta que valores inferiores a 60 segundos devuelve error el **Twitter**: recordad que no debemos sobrecargar a los servidores de dicha empresa con solicitudes tan frecuentes.

Opción **"-l"** ó **"--length"**.

Otro valor por defecto es el número de mensajes retribuidos, el valor normal es 20 pero podemos pedir uno solo o el que queramos pero con un máximo de 200 mensajes o tuits. Como por consola es difícil ver dónde termina y dónde comienza cada mensaje la siguiente opción nos permitirá contar de lo lindo el número de tuits retribuidos a ver si es verda que se ejecuta tal como lo pedimos.

Opciones **"-t"** ó **"--timestamp"** y **"-d"** ó **"--datestamp"**.

Con estas opciones podremos colocar tanto la hora y minuto como la fecha de cada mensaje, algo útil si queremos guardar los datos en una tabla de base dedatos y ordenarlos luego de forma cronológica. *Atención: la hora y fecha de cada mensaje es, por supuesto, cuando lo emitieron originalmente pero cuando alguien a quien nosotros seguimos lo retuitea -o reenvía- nos mostrará fecha y hora del retuiteo, no de la hora y fecha original.*

Opción **"--no-ssl"**.

Esta opción *de facto* está eliminada: toda transacción hoy en día con los servidores de la empresa Twitter solamente están disponibles en formato seguro **https**: toda nuestra información viaja cifrada entre nosotros y Twitter.

Opción **"c"** ó **"--config"**.

Si decidimos almacenar nuestro usuario y contraseña en un archivo con esta información la podemos cargar con esta opción. No se nos ocurre un uso práctico, sin contar que los datos, de manera implícita, estarán almacenadas en texto plano, un riesgo de seguridad para nosotros. Por esa razón es que dudamos -como dice la ayuda- que es para guardar usuario y contraseña. En más detalle indica que dicho archivo de configuración permite guardar consultas personalizadas siempre y cuando dentro del archivo tenga el encabezado **"[twitter]"** y a continuación en una línea que comienza con la etiqueta **"format:"** le colocamos las opciones que ya revisamos y en otra línea algo nuevo: un *prompt* personalizado en texto y colores. Esta opción es la más adecuada "para despertar al nerd que hay en tí" ja, ja, ja ;-)

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

Opción "--oauth".

De igual manera que la anterior pero con nuestras propias credenciales como es nuestro caso. os explicamos mejor, nosotros registramos una aplicación en Twitter que generó los cuatro tokens que explicamos párrafos atrás. Si los colocamos en un archivo con esta opción nos conectaremos por nuestra aplicación y no por la de **PTT**.

Opción "--format".

Si lo que queremos es ver nosotros mismos de una manera agradable por pantalla podemos optar entre los siguientes:

- **default**: bueno, es como si no hubieramos escrito nada, una línea por mensaje.
- **verbose**: múltiples líneas por mensaje.
- **json**: devuelve los datos en este popular formato.
- **urls**: si solamente queremos ver los enlaces (recordad que Twitter usa acortador de direcciones).
- **ansi**: la mejor opción, colores a diestra y siniestra.

Licencia de uso de PTT.

Python Twitter Tools se rige por la Licencia del Instituto Tecnológico de Massachusetts (Massachusetts Institute of Technologies **MIT**) del año 2008 y donde aparece el sr. Mike Verdone como único propietario:

```
Copyright (c) 2008 Mike Verdone      Permission is hereby granted, free of
charge, to any person obtaining a copy of this software and associated d
ocumentation files (the "Software"), to deal in the Software without re
striction, including without limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense, and/or sell copies of the Softw
are, and to permit persons to whom the Software is furnished to do so, s
ubject to the following conditions:    The above copyright notice and th
is permission notice shall be included in all copies or substantial port
ions of the Software.                THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANT
Y OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WAR
RANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINF
RINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE
FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTR
ACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE S
OFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

La Licencia del **MIT** permite uso privado, modificación, distribución y uso comercial *con la condición de que la licencia sea distribuida junto con los demás ficheros de la aplicación y la advertencia de que no se hacen responsables del mismo (en el software libre los únicos responsables somos nosotros mismos, esta parte de la licencia es muy dura de entender por los usuarios acostumbrados al software privativo).*

Equipo de desarrollo de PTT.

He aquí una lista de las personas que participan -y mantienen al día- el proyecto "Python Twitter Tools" -una [lista actualizada la podeis encontrar en este enlace](#) en GitHub, al momento presente son los siguientes:

Developers:

Mike Verdone

Hatem Nassrat

Wes Devauld

Contributors: Horacio Duran (utf-8 patch for IRC bot) Rainer Michael Schmid (bugfix: crash when redirecting output to a file in 1.1) Anders Sandvig (cmdline -l, -d, and -t flags) Mark Hammond (OAuth support in API) Prashant Pawar (IRC bot multi-channel support) David Bittencourt (python 2.3 support) Bryan Clark (HTTP encoding bugfix, improved exception logging) Irfan Ahmad (Fixed #91 rate limit headers and #99 twitter API timeouts) StalkR (archiver, follow) Matthew Cengia (DM support, ISO time zone support, more API 1.1 support) Andrew (Fixed streams support for HTTP1.1 chunked answers) Benjamin Ooghe-Tabanou (Helped fix streams support for HTTP1.1 chunked answers, added image support and more API 1.1 support)

Fuentes consultadas:

En castellano:

- "[twitter desde línea de comandos](#)" por Nacho, año 2009.
- "[Entendiendo OAuth: Qué ocurre cuando te loggeas en un sitio con Google, Twitter o Facebook](#)" por Lucila Armentano.

En idioma inglés:

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

- ["What is IRC?"](#).
- ["Twittering with Python Python libraries for the Twitter API: Overview and applications"](#).
- ["Update Your Twitter Status Remotely Using PHP"](#) by [David Walsh](#).
- ["Python Twitter Tools PTT"](#) by Mike Verdone.
- ["twitter 1.17.1 An API and command-line toolset for Twitter \(twitter.com\)"](#)
- ["Custom Twitter Tools"](#).
- ["Python Twitter Tools"](#) by Michael Martinez.
- ["python-twitter-examples"](#) by Ideofoms at GitHub.
- ["weetweet"](#) by DarkDefender at GitHub.
- ["twitter"](#) by sixsohsix at GitHub.
- ["Twitter Service Tweetgif Hacked, Time to Prune Your Twitter Apps"](#) by Whitson Gordon.
- ["PIN-based authorization"](#) at Twitter.
- ["Twitter alerts: how to integrate Pandora FMS alerts with Twitter"](#) por Antonio Sánchez Canales.
- ["Collecting Twitter Data: Introduction"](#) by Sean Dolinar.