



**Centro Universitario de la Defensa
en la Escuela Naval Militar**

TRABAJO FIN DE GRADO

*Plataforma OSINT para el almacenamiento y
análisis inteligente de datos de la red social
Twitter*

Grado en Ingeniería Mecánica

ALUMNO: Javier Jiménez-Cervantes Pérez

DIRECTORES: Norberto Fernández García

Miguel Rodelgo Lacruz

CURSO ACADÉMICO: 2022-2023

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Plataforma OSINT para el almacenamiento y
análisis inteligente de datos de la red social
Twitter*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Infantería de Marina

Universida_deVigo

RESUMEN

Este trabajo describe el desarrollo de una plataforma de inteligencia de código abierto que permite el almacenamiento, procesamiento y análisis de grandes cantidades de datos de Twitter para obtener OSINT. La plataforma cuenta con diferentes componentes, desde la recopilación de datos hasta la visualización de la información procesada, así como herramientas de análisis avanzado tal que la segmentación de datos y la identificación de patrones y tendencias.

Se presentan ejemplos de aplicación de la plataforma en áreas como el análisis de la opinión pública y la identificación de "*influencers*". La plataforma utiliza el lenguaje de programación Python para su obtención y herramientas de procesamiento de datos para su representación. Además, es una herramienta de código abierto que permite a los usuarios adaptarla a sus necesidades específicas.

En conclusión, la plataforma desarrollada tiene como objetivo proporcionar una herramienta de inteligencia de código abierto para el acceso a información actualizada sobre temas de interés público. Con implicaciones significativas para la sociedad y la investigación, la plataforma en un futuro podría ser una contribución importante al campo de la inteligencia artificial y un recurso valioso para aquellos que buscan analizar grandes cantidades de datos de las redes sociales de manera eficiente y efectiva.

PALABRAS CLAVE

OSINT, Twitter, Elasticsearch, Kibana, Tweepy

CONTENIDO

Contenido	1
Índice de Figuras	3
1 Introducción y objetivos	7
1.1 Introducción y motivación	7
1.2 Objetivos y metodología	10
1.3 Estructura de la memoria	10
2 Estado del arte	13
2.1 La Red Social Twitter	13
2.1.1 Inicios	13
2.1.2 Los primeros pasos de Twitter.....	13
2.1.3 Primeros años de Twitter	14
2.1.4 Primeros problemas	14
2.1.5 Últimas mejoras y su influencia en las noticias.....	14
2.1.7 Elementos básicos de un tuit.....	16
2.2 OSINT	18
2.2.1 Historia de OSINT	18
2.2.2 Ciclo de OSINT	19
2.3 <i>Elasticsearch</i>	19
2.3.1 Qué es Elasticsearch	19
2.3.2 Cómo funciona Elasticsearch	20
2.3.3 Ventajas Elasticsearch	20
2.4 Kibana	20
2.4.1 Qué es Kibana.....	20
2.4.2 Visualización de datos	21
2.4.3 Posibilidades que ofrece	21
2.5 Tweepy.....	22
2.5.1 Qué es Tweepy	22
3 Desarrollo del TFG.....	23
3.1 Procedimiento para la creación de una cuenta de desarrollador en Twitter.....	23
3.1.1 Creación de una cuenta en Twitter	23
3.1.2 Creación de una cuenta de desarrollador	27
3.2 Instalación de Tweepy	30
3.2.1 Características del equipo	30
3.2.2 Instalación de Tweepy	31

3.3 Proceso de instalación de Elasticsearch y Kibana.....	31
3.3.1 Descarga de Elasticsearch y Kibana	31
3.3.2 Instalar Elasticsearch	34
3.3.3 Instalar Kibana.....	36
3.4 Conseguir la información de los tuits (.txt).....	38
3.5 Convertir .txt a .json.....	40
3.6 Seleccionar los diccionarios que nos interesan del archivo .json	41
3.7 Introducir .json a Elasticsearch creando sus índices y <i>mapping</i>	43
4 Resultados / Validación / Prueba.....	47
4.1 Comprobación de los índices	47
4.2 Creación de representaciones gráficas	48
5 Conclusiones y líneas futuras	57
5.1 Conclusiones	57
5.2 Líneas futuras	57
6 Bibliografía.....	59
Anexo I: Implicaciones Sociales, y/o Económicas, y/o Ambientales	61
Implicaciones sociales.....	61
Potencial para la manipulación.....	61
Implicaciones económicas	61
Costes de la plataforma.....	61
Beneficios económicos de la información	61
Implicaciones ambientales	61
Consumo de energía	61
Generación de residuos electrónicos	61
Anexo II: Reflexiones Éticas y Sociales	63

ÍNDICE DE FIGURAS

Figura 1-1 Estadística del uso de internet [1].....	7
Figura 1-2 Estadística del uso de plataformas para acceso a las noticias [1].....	8
Figura 1-3 Cuenta oficial del Ministerio de Defensa	9
Figura 1-4 Cuenta sobre la guerra de Ucrania.....	9
Figura 1-5 Cuenta de un periodista	10
Figura 2-1 Primer tuit de la historia [5].....	13
Figura 2-2 Primera etiqueta creada [5].....	14
Figura 2-3 Cambio a 280 caracteres por tuit [6].....	15
Figura 2-4 Cómo usar una mención [10].....	16
Figura 2-5 Cómo funciona una etiqueta [10]	16
Figura 2-6 Tuit con imagen [11]	16
Figura 2-7 Cómo retuitear [12]	17
Figura 2-8 Cómo dar “me gusta”	17
Figura 2-9 Cómo crear un “hilo” [14]	18
Figura 2-10 Cuenta verificada.....	18
Figura 2-11 Ciclo de trabajo de OSINT	19
Figura 2-12 Logo Elasticsearch [17]	20
Figura 2-13 Gráficos ofrecidos por Elasticsearch	21
Figura 2-14 Ejemplo de <i>canvas</i> sobre los porcentajes de compra en una tienda	21
Figura 2-15 Ejemplo de mapa sobre el origen de los vuelos.....	22
Figura 3-1 Enlace de acceso a Twitter	23
Figura 3-2 Crear una cuenta	24
Figura 3-3 Completar los datos personales	24
Figura 3-4 Personalización de la experiencia.....	25
Figura 3-5 Confirmación de los datos personales	25
Figura 3-6 Introducción del código de confirmación.....	26
Figura 3-7 Creación de una contraseña de seguridad.....	26
Figura 3-8 Inicio de sesión	27
Figura 3-9 Acceso a la configuración.....	28
Figura 3-10 Acceso a Desarrolladores	28
Figura 3-11 Acceso al portal de desarrolladores	29
Figura 3-12 Información a rellenar	29
Figura 3-13 Aceptar términos y condiciones.....	30
Figura 3-14 Instalación de Tweepy	31

Figura 3-15 Acceso a las aplicaciones de Elastic	32
Figura 3-16 Selección de las aplicaciones de Elastic que se van a usar.....	32
Figura 3-17 Búsqueda de la versión de Elasticsearch que se va a usar	33
Figura 3-18 Búsqueda de la versión de Kibana que se va a usar	33
Figura 3-19 Selección del formato de descarga de Elasticsearch	33
Figura 3-20 Selección del formato de descarga de Kibana	34
Figura 3-21 Inicialización de Elasticsearch.....	34
Figura 3-22 Página Elasticsearch	35
Figura 3-23 Carpeta de Elasticsearch	35
Figura 3-24 Carpeta " <i>config</i> "	36
Figura 3-25 Cambio de nombre del cluster	36
Figura 3-26 Acceso a la carpeta " <i>bin</i> "	37
Figura 3-27 Inicialización de kibana.bat	37
Figura 3-28 Página principal de Kibana.....	38
Figura 3-29 Código para la obtención de tuits	39
Figura 3-30 Información obtenida de Twitter	40
Figura 3-31 Código para limpiar el documento	41
Figura 3-32 Estructura de los tuits recogidos	41
Figura 3-33 Código para la depuración del archivo .json	43
Figura 3-34 Código para indexar un archivo .json a Elasticsearch y crear sus índices y <i>mapping</i> ..	44
Figura 4-1 Acceso a los índices existentes	47
Figura 4-2 Comprobación de los índices creados	48
Figura 4-3 Pestaña de creación de "Dashboard"	48
Figura 4-4 Creación de un nuevo Dashboard	49
Figura 4-5 Creación de un gráfico de barras	49
Figura 4-6 Selección de los valores de los ejes	49
Figura 4-7 Gráfico de barras obtenidos	50
Figura 4-8 Selección del gráfico de tipo Donut.....	50
Figura 4-9 Selección de los campos a usar.....	51
Figura 4-10 Porcentaje de uso de las 6 etiquetas más usadas.....	52
Figura 4-11 Creación de un gráfico de líneas.....	52
Figura 4-12 Establecimiento de los valores del eje X	53
Figura 4-13 Filtros para los ejes X e Y.....	53
Figura 4-14 Gráfico de comparación de etiquetas.....	54
Figura 4-15 Creación de la tabla	55
Figura 4-16 Establecimiento de los parámetros de la tabla.....	55

Figura 4-17 Tabla resultado de los usuarios más mencionados56

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción y motivación

En la actualidad, el acceso a internet se ha vuelto imprescindible para la sociedad. Tanto particulares, como empresas y organizaciones, dependen totalmente de este servicio. Según los datos del 2022, el 63.1% de la población mundial utiliza internet, de los cuales el 59% son usuarios activos de redes sociales. Esto lo podemos ver reflejado en la Figura 1-1.

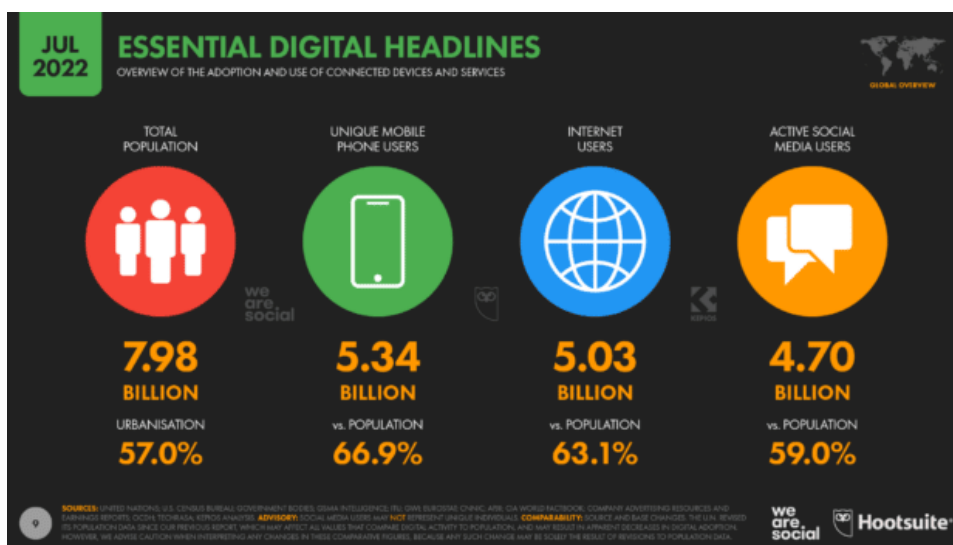


Figura 1-1 Estadística del uso de internet [1]

El número de personas que utilizan redes sociales aumentó en 227 millones durante el último año, alcanzando un total de 4,700 millones al comienzo de julio de 2022. La base global de usuarios de redes sociales ha crecido más del 5% en los últimos 12 meses, representando ahora el 59% de la población mundial.

A pesar de estos aumentos, las estadísticas muestran una desaceleración en el crecimiento digital en comparación con el aumento significativo observado durante la pandemia de COVID-19. Sin embargo, se espera que dos tercios de la población mundial estén conectados a internet en julio de 2023, y el número de usuarios de redes sociales alcance el 60% de la población mundial [1].

Como podemos ver en la Figura 1-2 el uso de las redes sociales es cada vez más importante, y se integra en nuestro día a día, llegando a convertirse en una de las principales fuentes de información de sus usuarios. Siendo según [1] Twitter una de las 5 redes sociales más consultadas.

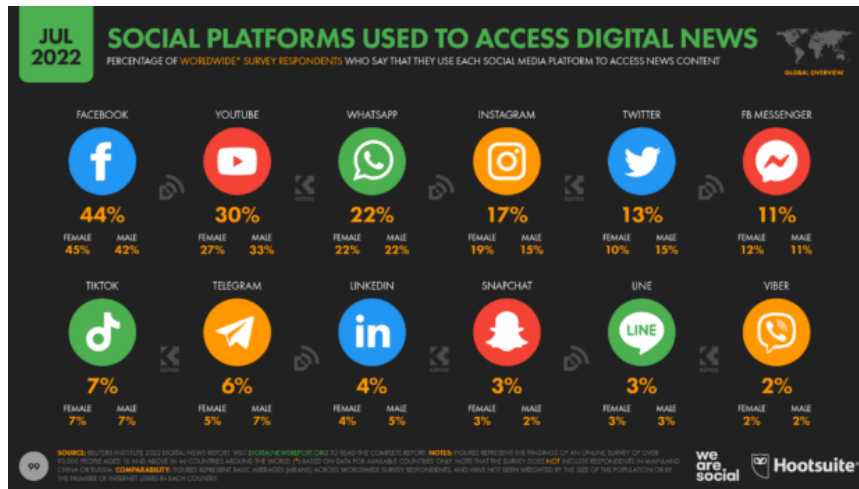


Figura 1-2 Estadística del uso de plataformas para acceso a las noticias [1]

Twitter es una plataforma de redes sociales en la que los usuarios pueden enviar y leer mensajes de texto cortos conocidos como "tuits" [2]. Cada tuit tiene una longitud máxima de 280 caracteres, lo que lo hace ideal para compartir pensamientos rápidos, noticias, enlaces y otros contenidos interesantes. Los usuarios pueden seguir a otros usuarios y ver sus tuits en su propia página de inicio, y también pueden "retuitear" o compartir tuits de otros usuarios. Twitter también tiene una función de "me gusta" para que los usuarios puedan mostrar su aprecio por un mensaje específico.

Además de los tuits, los usuarios de Twitter también pueden subir fotos, vídeos y contenido en vivo. También se pueden utilizar etiquetas ‘*hashtags*’ (palabras clave precedidas por el símbolo #) para clasificar los tuits por tema y permitir que los usuarios encuentren mensajes relacionados con un tema específico.

Twitter se ha vuelto muy popular entre los medios de comunicación, las celebridades y las empresas, ya que les permite interactuar directamente con sus seguidores y promocionar sus productos o servicios. También se ha convertido en una herramienta importante para la recopilación de noticias y la discusión de eventos en tiempo real.

OSINT ‘*Open-Source Intelligence*’ es un término utilizado para describir la recopilación, análisis y distribución de información que se encuentra en fuentes abiertas y disponibles para el público [3]. OSINT se refiere a cualquier tipo de información que pueda ser recolectada de fuentes públicas, como medios de comunicación, redes sociales o sitios web gubernamentales, entre otros.

La principal ventaja del OSINT es que la información que se recopila es generalmente gratuita y accesible para cualquier persona. También puede ser útil para obtener información precisa y actualizada sobre eventos, tendencias y personas.

OSINT se utiliza en una variedad de campos, incluyendo inteligencia militar, seguridad cibernética, investigaciones de fraude o investigaciones periodísticas, entre otros. Es una herramienta valiosa para obtener una comprensión más completa de una situación o evento, ya que permite recopilar información de varias fuentes diferentes.

Las redes sociales son una fuente común de información para OSINT debido a la cantidad de datos disponibles y la facilidad de acceso. Los usuarios de redes sociales comparten información personal y profesional en línea, lo que proporciona una gran cantidad de información valiosa para OSINT. Además, las redes sociales también son una fuente importante de noticias y actualizaciones en tiempo real, lo que permite a los investigadores estar al tanto de eventos actuales.

La información recopilada de las redes sociales también puede ser utilizada para la identificación de personas y organizaciones, la monitorización de eventos y tendencias, y para la recopilación de datos geográficos. Esto puede ser útil para la investigación, el análisis y la toma de decisiones.

En resumen, las redes sociales son una fuente valiosa para OSINT debido a la cantidad de datos disponibles, la facilidad de acceso y la posibilidad de recopilar información en tiempo real. Esta información puede ser utilizada para una gran variedad de propósitos.

Esto se ve reflejado en la importancia que se les da a los medios de comunicación de Defensa como puede ser la cuenta oficial del Ministerio de Defensa, donde se actualiza toda la información relacionada con nuestras Fuerzas y Cuerpos de Seguridad del Estado y su labor tanto dentro como fuera de nuestra nación (véase Figura 1-3).



Figura 1-3 Cuenta oficial del Ministerio de Defensa

También hay que recalcar la importancia de medios de información como puede ser las distintas cuentas que han ido surgiendo con el tiempo que actualizan los sucesos que van ocurriendo de manera diaria en los distintos conflictos, un ejemplo de esto puede ser la actual guerra de Ucrania, o la de periodistas que con sus labores de investigación son capaces de actualizarnos la información que tenemos de estos asuntos (véase Figura 1-4 y Figura 1-5).

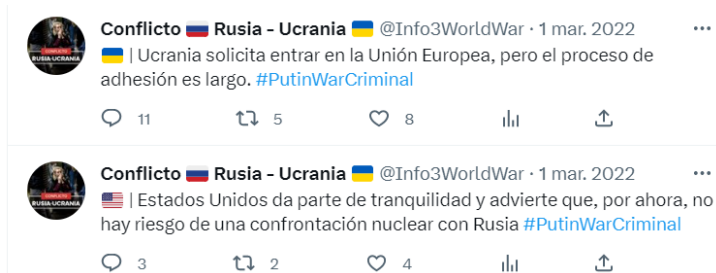


Figura 1-4 Cuenta sobre la guerra de Ucrania



Figura 1-5 Cuenta de un periodista

1.2 Objetivos y metodología

Los objetivos del trabajo son desarrollar una plataforma OSINT para el almacenamiento y análisis inteligente de datos de la red social Twitter, centrándose en tuits relacionados con la guerra de Ucrania. Para posteriormente poder indexarlos en Elasticsearch y trabajar sobre ellos con el programa Kibana.

En cuanto a la metodología utilizada en el trabajo incluirá los siguientes pasos:

- Recopilación de datos de Twitter utilizando Tweepy.
- Indexación de datos en Elasticsearch.
- Configuración y uso de Kibana para visualizar y analizar datos almacenados en Elasticsearch.

También se espera mantener reuniones periódicas con los tutores y un estrecho contacto para poder resolver cuestiones y tomar decisiones sobre la evolución a corto y medio plazo del trabajo a realizar.

1.3 Estructura de la memoria

La memoria de este TFG comienza con una introducción en la que se presenta el contexto y la motivación del proyecto, así como los objetivos y la metodología que se utilizará para alcanzarlos.

En el capítulo 2, se realiza un análisis del estado del arte en lo que respecta a las herramientas y técnicas utilizadas en la recolección y análisis de datos en redes sociales, en particular en Twitter. Se investiga sobre la minería de datos y las técnicas de visualización, así como el uso de herramientas de código abierto.

En el capítulo 3, se describirá en detalle el proceso de creación de la plataforma OSINT para el almacenamiento y análisis inteligente de datos de la red social Twitter, enfocada en la recolección de datos relacionados con la guerra de Ucrania. Se explicará el diseño de la arquitectura de la plataforma, las tecnologías y herramientas utilizadas, así como el proceso de recolección de los datos y su almacenamiento en la base de datos. Además, se detallará cómo se ha llevado a cabo el análisis de los datos.

El siguiente capítulo, el 4, se enfoca en el proceso para el análisis de los datos obtenidos en Twitter relacionados con la guerra en Ucrania. Se describen las diferentes técnicas de análisis utilizadas para comprender mejor los patrones y las tendencias en los datos, así como el procedimiento que hay que

realizar para poder poner en funcionamiento el programa y las herramientas de visualización empleadas para presentar los resultados.

Por último, en el apartado 5, se expondrán las principales conclusiones y reflexiones que se han extraído del proyecto. Se destacarán las principales aportaciones y limitaciones del trabajo, y se discutirán posibles aplicaciones futuras y líneas de investigación que se podrían seguir para mejorar y ampliar el alcance de la plataforma OSINT.

2 ESTADO DEL ARTE

2.1 La Red Social Twitter

2.1.1 Inicios

La historia de Twitter comenzó en el año 2004 cuando Noah Glass, un programador estadounidense, ideó un proyecto llamado Odeo [5]

Su idea fundamental era permitir a los usuarios dejar mensajes grabados en formato MP3 al llamar por teléfono y almacenarlos en la nube.

Más tarde, Evan Williams, creador de Blogger.com, se unió al proyecto. Odeo se convirtió en una plataforma de podcasting con 14 empleados, pero debido a la competencia de Apple con su plataforma de podcasts en iTunes, Odeo comenzó a tener problemas.

Williams, entonces CEO de la compañía, decidió que la *start-up* necesitaba reinventarse o modernizarse para evitar el fracaso. Es entonces cuando Jack Dorsey, diseñador de la empresa, tuvo la idea de usar SMS para crear conversaciones entre pequeños grupos de personas, lo que daría origen a Twitter.

2.1.2 Los primeros pasos de Twitter

Twitter fue creado en 2006 como un servicio interno de Odeo, con varios nombres propuestos antes de elegir "Twtr" para imitar el trino de un pájaro. La tendencia en ese momento era dejar caer las vocales en el nombre de las *start-ups* (como Flickr, Tumblr, Scribd). El primer tuit en la plataforma fue escrito por Jack Dorsey el 21 de marzo de 2006 y decía "*Just setting up my twtr*" (véase Figura 2-1)



Figura 2-1 Primer tuit de la historia [5]

El límite de 140 caracteres en los tuits que caracterizaba a Twitter en sus inicios se debía a que la plataforma se basaba en el protocolo SMS de los operadores de telefonía móvil.

2.1.3 Primeros años de Twitter

En octubre de 2006, Twitter fue comprado por inversores y se creó Obvious Corporation. Con este cambio, Jack Dorsey asumió el cargo de primer director ejecutivo, Biz Stone se convirtió en director creativo y Evan Williams en presidente.

En 2007, Twitter se convirtió en una empresa independiente y Jack Dorsey fue nombrado CEO. En ese tiempo, el número de usuarios y el tráfico en la plataforma aumentaron significativamente, pasando de 20.000 a 70.000 tuits al día. Además, en agosto de ese año, un usuario llamado Chris Messina inventó la etiqueta ‘*hashtag*’ por casualidad, que más tarde se implementaría en Twitter (véase Figura 2-2).



Figura 2-2 Primera etiqueta creada [5]

Sin embargo, en la parte de atrás, las tensiones entre los fundadores de la compañía estaban creciendo. Finalmente, Williams propuso la expulsión de Jack Dorsey de la compañía, una decisión que fue apoyada por el resto de la compañía.

Con Williams como CEO, Twitter continuó creciendo, y el número de usuarios únicos aumentó un 1.300% en 2009, pasando de 5 a 71 millones de usuarios. Sin embargo, la plataforma comenzó a sufrir caídas continuas debido al alto tráfico.

2.1.4 Primeros problemas

Debido a que la plataforma no podía mantenerse estable y el problema no se resolvía, las críticas empeoraron. Williams había supervisado el rápido aumento de usuarios, pero en ese momento no era capaz de tomar las decisiones adecuadas. El y Biz Stone sabían que necesitaban contratar a un gran número de empleados para manejar el tráfico. A pesar de tener 145 millones de usuarios, Twitter no estaba generando muchos ingresos.

Finalmente, ante esta situación, los miembros de la junta de Twitter decidieron despedir a Evan Williams y recontrataron a Jack Dorsey como CEO. Dick Costolo, un emprendedor y antiguo comediante, sustituyó a Williams como CEO hasta que Jack Dorsey volvió a asumir el cargo en 2015. Biz Stone dejó la compañía en junio de 2011 debido a la falta de apoyo.

2.1.5 Últimas mejoras y su influencia en las noticias

Twitter ha estado tratando de mantener su posición como una red social de referencia a través de la incorporación de nuevas funcionalidades para atraer a usuarios nuevos y mantener a los usuarios veteranos. Esto incluye la opción de incluir múltiples fotos o gifs en los tuits, la posibilidad de hacer

encuestas, la eliminación del límite de caracteres para los mensajes privados, y la ampliación del límite de caracteres a 280 (véase Figura 2-3) [6].



Figura 2-3 Cambio a 280 caracteres por tuit [6]

Sin embargo, no todos los cambios han sido bien recibidos por todos, como el cambio en la línea de tiempo, donde los tuits aparecen en orden de relevancia en lugar de orden cronológico, lo que causó la protesta de los usuarios y la creación de la etiqueta *#RIPTwitter*.

A pesar de esto, Twitter sigue siendo una plataforma importante en la comunicación y el marketing, y muchas personalidades políticas, deportivas y culturales lo usan para comunicarse con su audiencia. Incluso una sola publicación mal gestionada puede afectar negativamente la reputación de una empresa, persona o *influencer*. Twitter ha sido una revolución en la comunicación en internet y sigue siendo una plataforma poderosa para hacer algo viral.

Twitter también ha sido utilizado para cubrir eventos de actualidad en tiempo real, desde protestas hasta desastres naturales. Los tuits de los testigos y los reporteros en el terreno pueden proporcionar una perspectiva única y actualizaciones instantáneas sobre lo que está sucediendo.

Sin embargo, la rapidez con la que se comparten las noticias en Twitter también ha llevado a una mayor probabilidad de difundir información falsa o engañosa. Por lo tanto, es importante ser crítico y verificar la veracidad de las noticias antes de compartirlas.

2.1.6 Aparición de Elon Musk y sus cambios

En abril de 2021, Musk anunció que quería comprar la plataforma por la suma de 44000 millones de dólares, aunque la junta decía que no estaba en venta [8]. Por ello, Elon compró gran cantidad de las acciones de la empresa y tras un tiempo de negociaciones, finalmente se aceptó la oferta por la empresa. Finalmente, la transacción se realizó con éxito el 28 de octubre de 2021.

Después de comprar Twitter, Elon Musk ha tomado varias medidas para cambiar la dirección de la compañía, incluyendo el despido de varios altos ejecutivos. También ha cambiado su postura sobre la publicidad en la red social, pasando de querer eliminar los anuncios a querer convertir a Twitter en la "plataforma de publicidad más respetable del mundo." Por último, Musk ha decidido cobrar por la verificación de perfiles, primero a 19.99\$, luego a 8\$ al mes.

2.1.7 Elementos básicos de un tuit

Dentro de un tuit, tenemos gran cantidad de elementos distintos, que si no se conocen pueden crear cierta confusión cuando se hablen de ellos en el trabajo, por lo que se repasarán aquellos elementos más importantes.

- Texto: es el principal contenido del mensaje, como ya se comentó con anterioridad el límite se encontraba en 140 caracteres, pero esto fue aumentado hasta los 280 caracteres.
- Menciones (@): consiste en un tuit que contiene un “@” seguido del nombre de usuario de otra cuenta. De esta manera notificará al usuario de la cuenta mencionada. [9] (véase Figura 2-4)



Figura 2-4 Cómo usar una mención [10]

- Etiqueta (#): Se puede colocar en cualquier lugar del tuit, de tal manera que Twitter te lo detectará e hipervinculará, de tal manera que si clicamos sobre la etiqueta se nos desplegará una lista con todos los tuits con la misma etiqueta. También se puede buscar las etiquetas escribiéndolas en el buscador. Es por ello por lo que usar las etiquetas hace que sea más visible tu tuit (véase Figura 2-5).

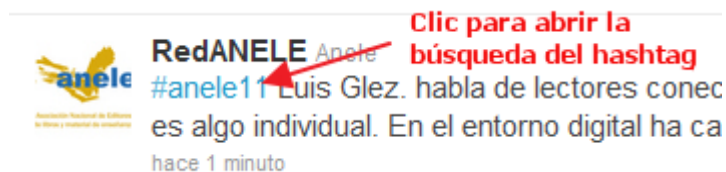


Figura 2-5 Cómo funciona una etiqueta [10]

- Imágenes/Videos: opcionalmente, se pueden añadir imágenes o vídeos al tuit (véase Figura 2-6).



Figura 2-6 Tuit con imagen [11]

- Retuits (*RT*): permite compartir un tuit publicado por otro usuario o por ti mismo y hacerlo visible a tus seguidores [12], dándote la posibilidad de añadir información extra o no a ese tuit.

Para retuitear únicamente has de pulsar en el símbolo de dos flechas que forman un cuadrado (véase Figura 2-7).



Figura 2-7 Cómo retuitear [12]

- Me gusta (*Like*): antiguamente esto se representaba con una estrella y se llamaba favorito, pero esto se cambió en 2015 y pasó a representarse con un corazón [13] (véase Figura 2-8); y no solo significa que te gusta lo que has leído, sino que también sirve para dar apoyo al autor de ese tuit, mostrar que estás de acuerdo con la información dada o simplemente para marcar ese tuit y poder releerlo en otro momento.



Figura 2-8 Cómo dar “me gusta”

- Hilo: Un hilo se realiza cuando con un solo tuit no eres capaz de expresar toda la información que quieres retransmitir [9]. Esto nos permite dar un mayor contexto al tuit original o actualizar la información. Esto se realiza mediante el botón (+) que sale al postear el tuit original (véase Figura 2-9).

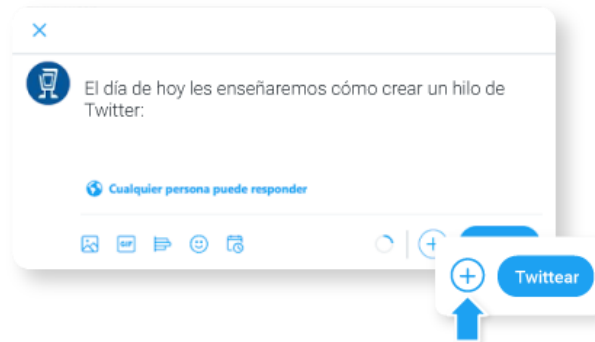


Figura 2-9 Cómo crear un “hilo” [14]

- Verificación de perfiles: sirve únicamente para mostrar al público que las personas más relevantes que se encuentran en la red social Twitter son quienes dicen ser [15]. Esto se representa con un *tick* azul a la derecha del nombre del usuario (véase Figura 2-10).



Figura 2-10 Cuenta verificada

2.2 OSINT

2.2.1 Historia de OSINT

La historia de la OSINT se remonta a la época de la Segunda Guerra Mundial [16], cuando los gobiernos de Estados Unidos y Reino Unido comenzaron a recopilar la información públicamente disponible potencialmente relevante para la seguridad nacional y la defensa. Estados Unidos fue pionero en la creación de una capacidad autónoma para vigilar y analizar medios de comunicación extranjeros con la creación del *Foreign Broadcast Monitoring Service (FBMS)* en 1941. En Reino Unido, la BBC lanzó un programa similar conocido como *Digest of Foreign Broadcasts*. Con el tiempo, estas iniciativas evolucionaron para convertirse en herramientas estratégicas y complejas para recopilar e interpretar información de fuentes públicas. La OSINT ha seguido evolucionando con el aumento de la tecnología y la información disponible en internet, y se ha convertido en un campo importante para la seguridad nacional y la investigación criminal. Con el aumento del uso de internet y las redes sociales, el OSINT se ha vuelto cada vez más importante en la recopilación de información. Ahora, los profesionales del

OSINT utilizan una variedad de herramientas y técnicas para recopilar información de fuentes en línea, como motores de búsqueda, redes sociales y sitios web de noticias.

2.2.2 Ciclo de OSINT

“El ciclo de inteligencia es lo que permite transformar esos datos en el conocimiento que deseamos obtener.” [17]

Este ciclo de inteligencia se puede dividir en 5 etapas (véase Figura 2-11).



Figura 2-11 Ciclo de trabajo de OSINT

1. **Dirección**: en esta fase se determinan cuáles son los objetivos que queremos alcanzar y que datos son de los que disponemos. Lo que nos permite valorar todo lo que nos falta por saber y como abarcarlo.
2. **Recolección**: aquí es donde empezamos a obtener la información necesaria para rellenar nuestros vacíos de información, bien sea por herramientas automatizadas o trabajo manual. El mayor riesgo de esta fase es no tener un método, lo que puede suponer que vayamos saltando de información a información, dejándonos por el camino lagunas de información.
3. **Procesamiento**: esta información recolectada en la etapa anterior es necesario analizarla para poder darle una estructura, lógica y orden
4. **Análisis**: En esta fase convertimos toda esta información recogida en conocimiento al analizar e interconectar las pequeñas partes de información obtenida. Lo común es obtener nuevas fuentes de información al descubrir carencias que tienes para poder enlazar la que ya tienes. De esta manera vemos que el proceso no tiene una dirección única y es flexible.
5. **Difusión**: esta etapa únicamente se alcanza cuando hemos completado el objetivo marcado en la parte de “dirección”. Y se entrega un informe de como se ha conseguido esta información para que el que la reciba pueda reconstruir el proceso y asegurar el origen de los datos.

2.3 Elasticsearch

2.3.1 Qué es Elasticsearch

Elasticsearch es un motor de búsqueda y análisis de información (textual, numérica, geoespacial, estructurado y no estructurados) gratuito, aunque también existe de pago, pero para este proyecto no será necesario el uso de los programas de pago. Su función es buscar, almacenar y analizar grandes volúmenes de información [17] (véase Figura 2-12).



Figura 2-12 Logo Elasticsearch [17]

Su principal uso es en el ámbito de búsqueda empresarial y analítica de datos y se usa de manera conjunta con gran cantidad de herramientas para procesar estos datos recogidos y visualizarlos como puede ser Kibana.

2.3.2 *Cómo funciona Elasticsearch*

Su funcionamiento se basa en recoger datos desde diferentes fuentes, sistemas y aplicaciones. Una vez registrada la información en Elasticsearch el usuario podrá realizar consultas sobre estos datos y recuperar resúmenes de ellos.

Durante el proceso de registro, Elasticsearch almacena estos documentos y crea un índice invertido para poder buscar datos en el documento en tiempo real

2.3.3 *Ventajas Elasticsearch*

- Es rápido: tiene una excelente capacidad de búsqueda de texto completo. También es una plataforma de búsqueda en casi tiempo real con baja latencia de indexación y búsqueda, lo que lo hace ideal para casos de uso con restricciones de tiempo como analítica de seguridad y monitoreo de infraestructura.
- Distribuidos por naturaleza: los documentos que se almacenan se distribuyen en distintos almacenes de información conocidos como “*Shards*”, de los cuales se realizan copias para crear duplicados redundantes por si falla el hardware, lo que permite gestionar gran cantidad de datos.
- Tiene un amplio conjunto de características: además de su rapidez, capacidad de escalar y la resistencia, Elasticsearch tiene otra gran cantidad de características integradas que ayudan a que la búsqueda de almacenamiento y almacenamiento de datos sea más eficaz.
- Simplifica la ingesta, visualización y distribución de datos: Kibana le ofrece la capacidad de representar en tiempo real de datos, así como UI para acceder rápidamente al monitoreo de rendimiento de aplicaciones.

2.4 Kibana

2.4.1 *Qué es Kibana*

Es una herramienta que permite la visualización y gestión de datos obtenidos con Elasticsearch, convirtiéndolos en histogramas, gráficas y mapas.

Esto significa que podemos dar forma a nuestros datos, crear índices para poder aplicarle operaciones como puede ser “*machine learning*”, analítica de logs o análisis semántico de los campos de textos; también es capaz de crear visualizaciones de nuestros datos, así como crear *dashboards* y reportes

2.4.2 Visualización de datos

La visualización de los datos busca representarlos de manera gráfica para que sea más sencillo entenderlos.

La visualización de datos se logra a través de un "frontend" abierto que consiste en una interfaz gráfica para los datos. De esta manera puedes crear gráficos como mapas, tablas, presentaciones y más.

2.4.3 Posibilidades que ofrece

- 'Dashboard': Nos ofrece un panel con distintas gráficas en las que se comparan distintos gráficos sobre distintos valores que le insertamos mediante archivos 'json' consiguiendo así unos valores como estos (véase Figura 2-13):

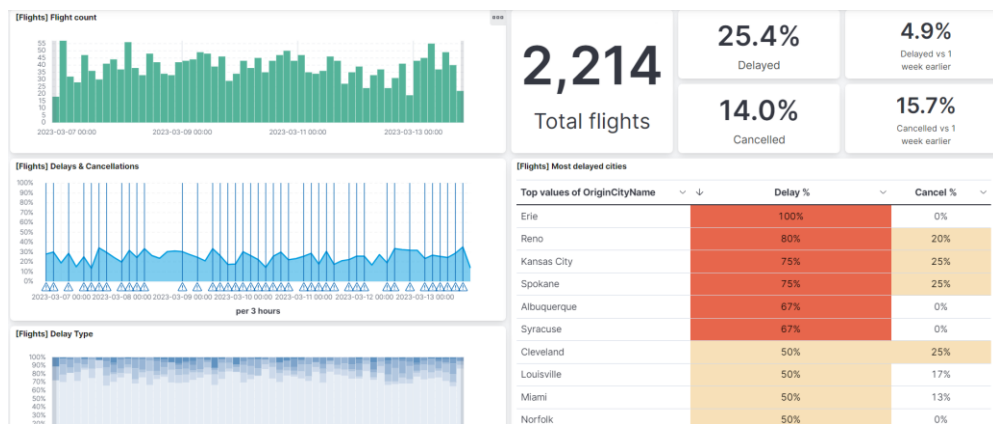


Figura 2-13 Gráficos ofrecidos por Elasticsearch

- 'Canvas': Crea tableros personalizados en los que se presentarán de manera gráfica los distintos valores con diversos logotipos, colores y diseños de manera automática (véase Figura 2-14).

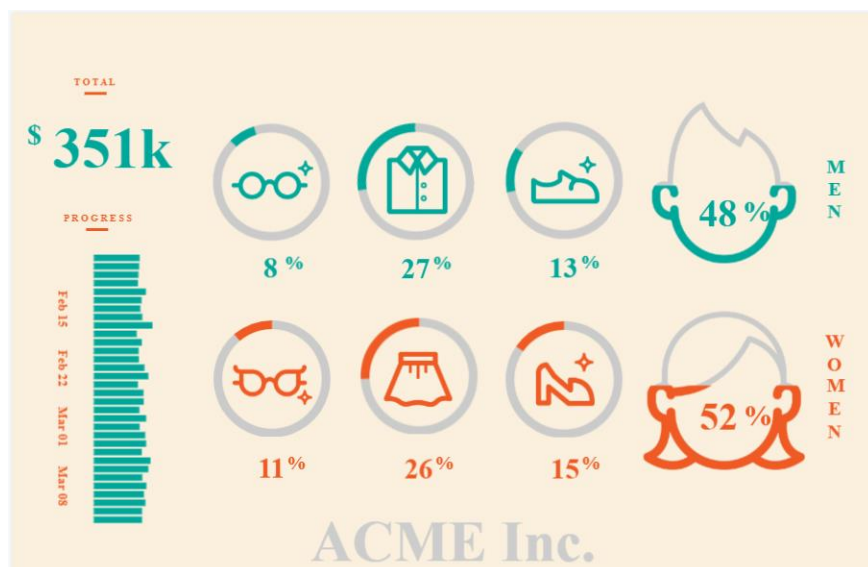


Figura 2-14 Ejemplo de canvas sobre los porcentajes de compra en una tienda

- 'Maps': Sirve para representar de manera geográfica de qué lugar provienen los distintos valores insertados en el programa (véase Figura 2-15).

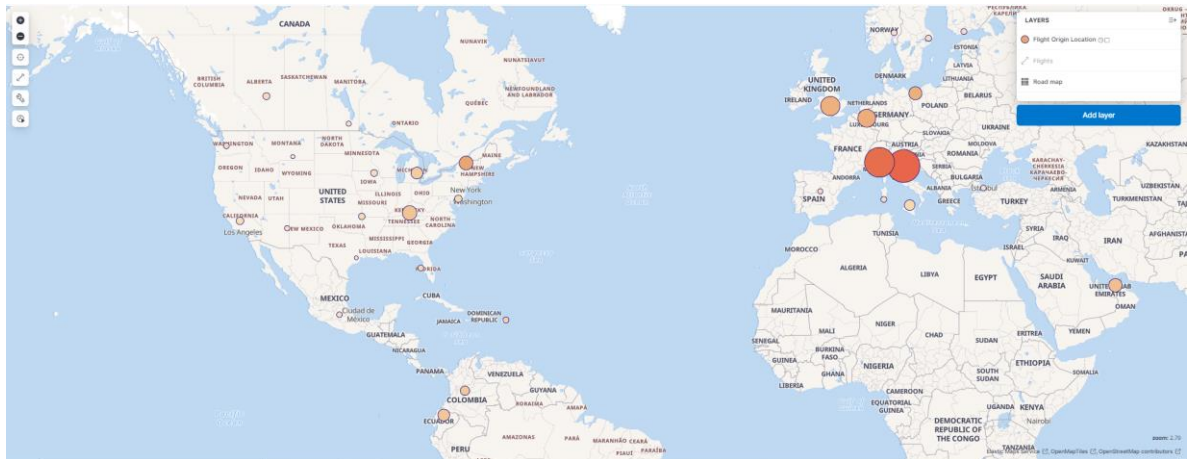


Figura 2-15 Ejemplo de mapa sobre el origen de los vuelos

2.5 Tweepy

2.5.1 Qué es Tweepy

Tweepy es una biblioteca de Python que permite interactuar con la API de Twitter para acceder a datos y realizar acciones en la plataforma de Twitter. Tweepy simplifica la tarea de trabajar con la API de Twitter al proporcionar una interfaz fácil de usar y abstraer los detalles complejos de la autenticación y la interacción con los “endpoints” de la API de Twitter.

Tweepy se utiliza comúnmente para realizar tareas como buscar tweets, enviar tweets, seguir y dejar de seguir usuarios, obtener información de usuario y mucho más. Es una herramienta muy útil para cualquier persona que desee integrar la funcionalidad de Twitter en sus aplicaciones de Python.

3 DESARROLLO DEL TFG

En este capítulo vamos a explicar cómo instalar Elasticsearch, Kibana y la creación de una cuenta de desarrollador en Twitter

3.1 Procedimiento para la creación de una cuenta de desarrollador en Twitter

3.1.1 Creación de una cuenta en Twitter

Lo primero sería crear una cuenta en la aplicación de Twitter, para ello se tendrá que entrar en la página web de Twitter introduciendo la siguiente URL: “<https://twitter.com/home?lang=es>” (véase Figura 3-1).

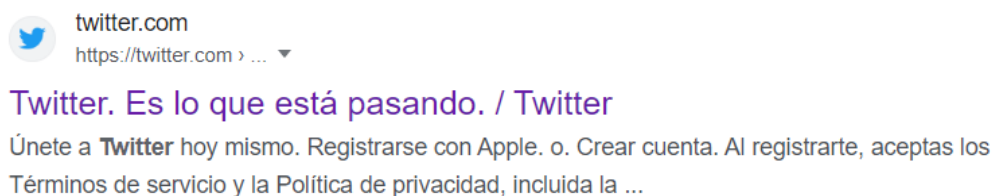


Figura 3-1 Enlace de acceso a Twitter

A continuación, en la página principal en el lado superior derecho de la ventana se ofrecerá la posibilidad de iniciar sesión o de crear una cuenta. En este caso supuesto se creará una cuenta nueva al pinchar en “crear cuenta” (véase Figura 3-2).

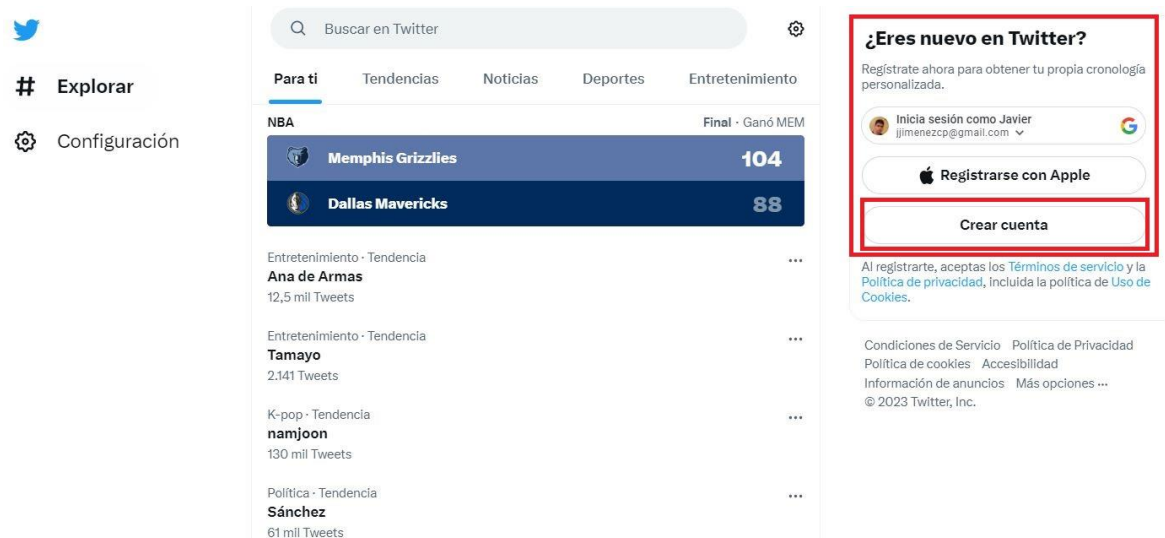


Figura 3-2 Crear una cuenta

Para ello se tendrán que completar 5 pasos, siendo el primero el de rellenar el nombre, número de teléfono y la fecha de nacimiento (véase Figura 3-3).

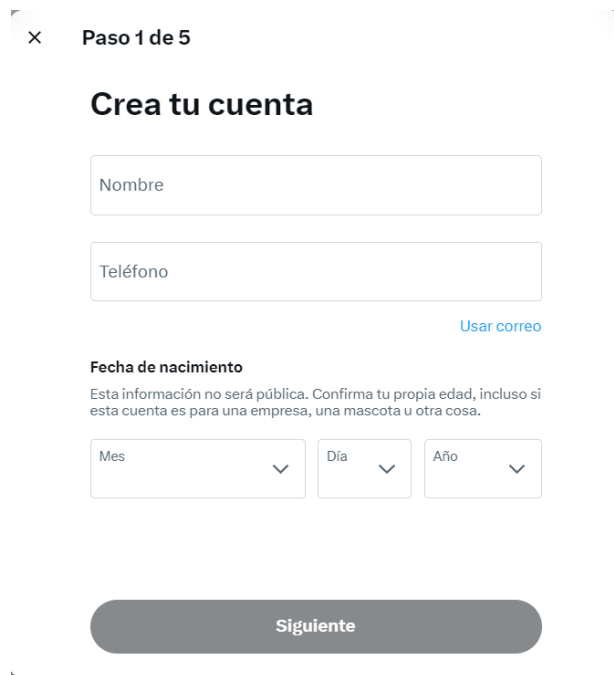


Figura 3-3 Completar los datos personales

Lo siguiente sería personalizar la experiencia del usuario con distintas opciones (véase Figura 3-4). Ya que esta cuenta se está creando con la intención de realizar un estudio, no será necesario personalizar la experiencia, pudiendo saltar este paso al hacer clic en siguiente sin haber seleccionado ninguna casilla.

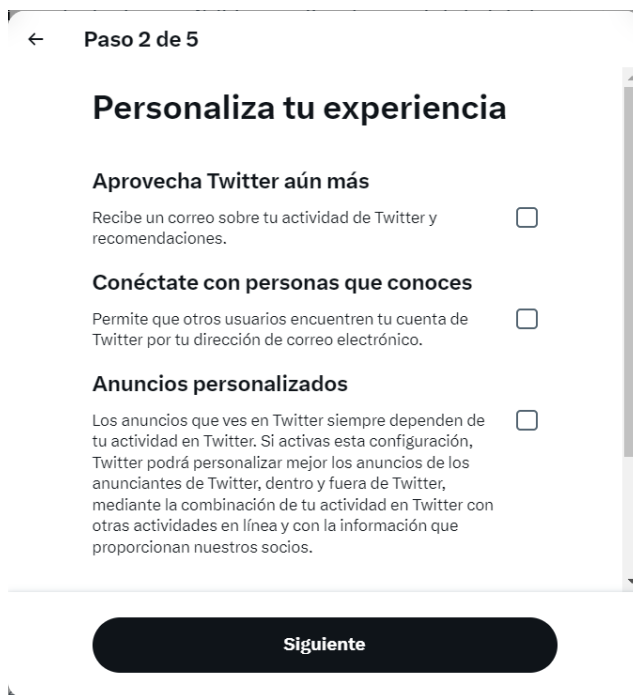


Figura 3-4 Personalización de la experiencia

De esta manera aparecerá en pantalla el nombre, correo electrónico y fecha de nacimiento que se introdujo en el paso 1 (véase Figura 3-5), para que así se confirme que los datos registrados son correctos y están disponibles.

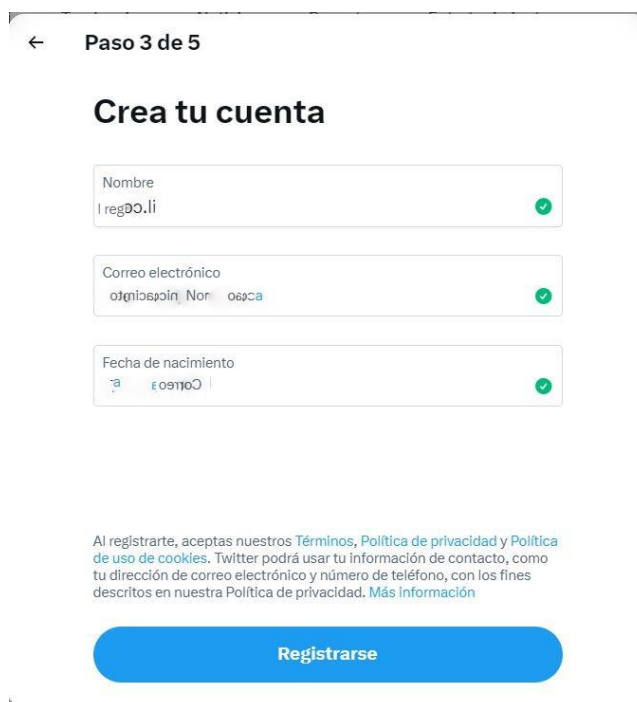


Figura 3-5 Confirmación de los datos personales

Es en este momento en el que se mandará al correo que se había indicado con anterioridad un código de autenticación que confirmará a Twitter que el usuario tiene verdaderamente acceso a la cuenta de correo electrónico que se ha introducido (véase Figura 3-6).

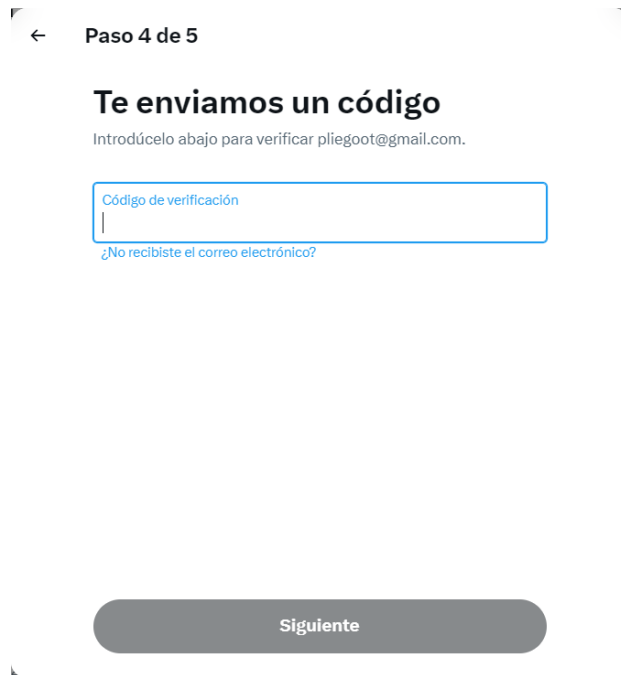


Figura 3-6 Introducción del código de confirmación

Dando pie así al último paso para la creación de la cuenta, la selección de la contraseña que se usará, donde se exigirá que esta misma tenga como mínimo un total de 8 caracteres (véase Figura 3-7).

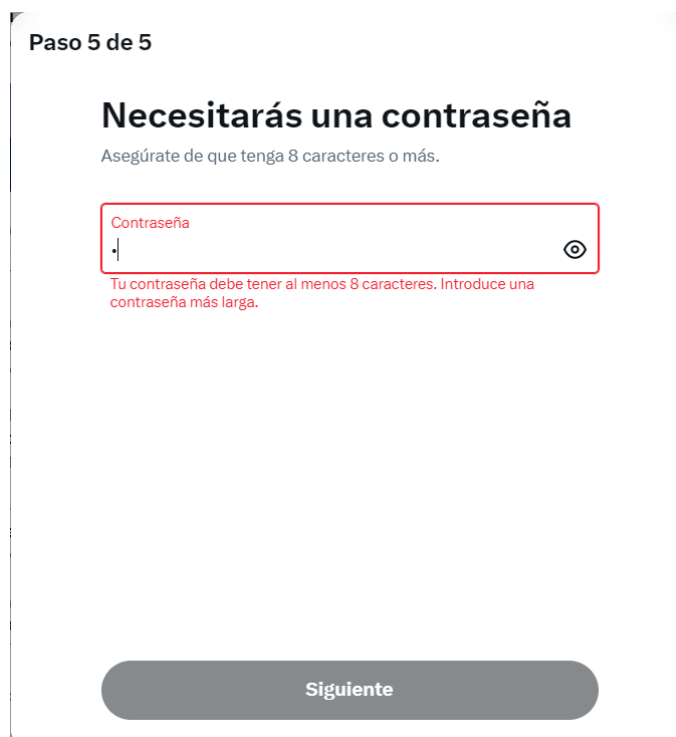


Figura 3-7 Creación de una contraseña de seguridad

3.1.2 Creación de una cuenta de desarrollador

Una vez creada la cuenta de usuario en Twitter será necesario crear una cuenta de desarrollador. Para ello se deberá iniciar sesión en la página oficial de Twitter (véase Figura 3-8).

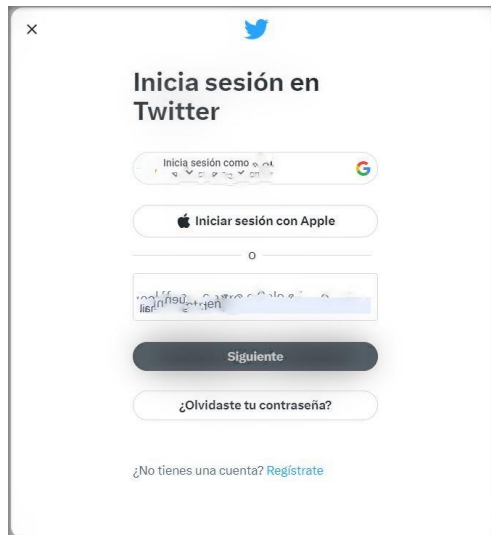


Figura 3-8 Inicio de sesión

Una vez iniciada la sesión en la esquina superior izquierda de Twitter se ofrecerá una serie de opciones tales como inicio (que llevará al *feed* principal), explorar (que llevará a las tendencias de Twitter del día) o mensajes (donde se almacenan todas las conversaciones privadas que se han mantenido con otros usuarios). Pero el apartado que nos interesa en nuestro caso será el de más opciones, y dentro de él, se entrará a el apartado de configuración y privacidad (véase Figura 3-9)

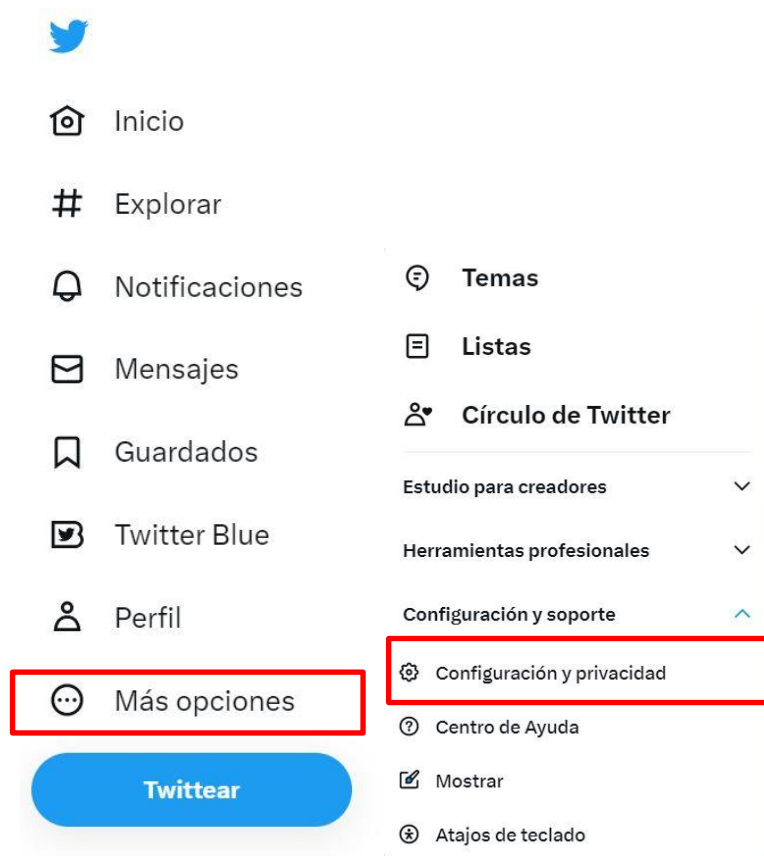


Figura 3-9 Acceso a la configuración

Una vez dentro de la configuración, de entre todas las opciones que se ofrecen, habrá que dirigirse a la opción de “recursos adicionales”, lo cual desplegará una ventana con diversas alternativas como son “notas de la versión”, “legal” o “varios”. Será esta última en la que se tendrá que hacer clic en el ítem denominado “desarrolladores” (véase Figura 3-10).

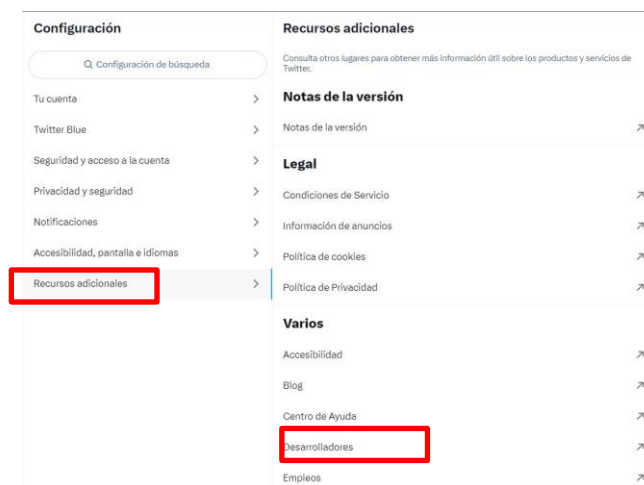


Figura 3-10 Acceso a Desarrolladores

Esto dirigirá al usuario a la Plataforma de Desarrolladores, una vez ahí se entrará en el enlace de la parte superior derecha “Developer Portal” (véase Figura 3-11).



Figura 3-11 Acceso al portal de desarrolladores

Esto abrirá una nueva ventana en la que se pedirá confirmar el nombre de usuario y el correo electrónico que usa el individuo en Twitter y también se tendrá que indicar cuál es el país de origen.

Dentro del apartado de cuál será el uso que se le dará a la cuenta de desarrollador, habrá que seleccionar la opción de “Academic research” (véase Figura 3-12).

En cuanto a la última pregunta que se realiza es recomendable indicar que no se usará la información que se obtenga para distribuirla a alguna entidad gubernamental, lo cual además es el caso en el trabajo que nos ocupa.

Figura 3-12 Información a rellenar

Lo siguiente será aceptar los términos y condiciones que hablan sobre las restricciones en el uso de las APIs de Twitter, respetar los derechos de propiedad intelectual, la prohibición de ciertos tipos de

contenido, la responsabilidad del desarrollador, la posibilidad de suspensión de acceso, modificaciones de los términos, prohibición de uso comercial y cumplimiento de las políticas de privacidad (véase Figura 3-13).

Developer agreement & policy

Developer Agreement

Effective: January 19, 2023

This Twitter Developer Agreement (“**Agreement**”) is made between you (either an individual or an entity, referred to herein as “**you**”) and Twitter (as defined below) and governs your access to and use of the Licensed Material (as defined below). Your use of Twitter’s websites, SMS, APIs, email notifications, applications, buttons, embeds, ads, and our other covered services is governed by our general Terms of Service and Privacy Policy.

PLEASE READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY, INCLUDING ANY LINKED TERMS REFERENCED BELOW, WHICH ARE PART OF THIS LICENSE AGREEMENT. BY USING THE LICENSED MATERIAL, YOU ARE AGREEING THAT YOU

Accept Terms & Conditions

By clicking on the box, you indicate that you have read and agree to this [Developer Agreement](#) and the [Twitter Developer Policy](#), additionally as it relates to your display of any of the Content, the [Display Requirements](#); as it relates to your use and display of the Twitter Marks, the [Twitter Brand Assets and Guidelines](#); and as it relates to taking automated actions on your account, the [Automation Rules](#). These documents are available in hardcopy upon request to Twitter.

Back

Submit

Figura 3-13 Aceptar términos y condiciones

Por último, habrá que rellenar una serie de datos en los que se tendrá que explicar de manera concienzuda para qué se va a usar la cuenta de desarrollador, cómo se va a analizar los datos que se obtendrán de Twitter, qué se planea hacer con esos datos, cómo serán obtenidos desde fuera de Twitter y en el caso de que se tenga algún tipo de enlace con algún gobierno, cuáles son esos gobiernos.

Será entonces cuando después de recibir la confirmación de Twitter se tendrá acceso a la cuenta de desarrollador y se proporcionarán la “consumer_key”, “consumer_secret”, “access_token”, “access_token_secret” y “bearer_token”, que serán usadas más adelante en el código de “Tweepy” para poder acceder a la información de los tuits que se deseen.

3.2 Instalación de Tweepy

3.2.1 Características del equipo

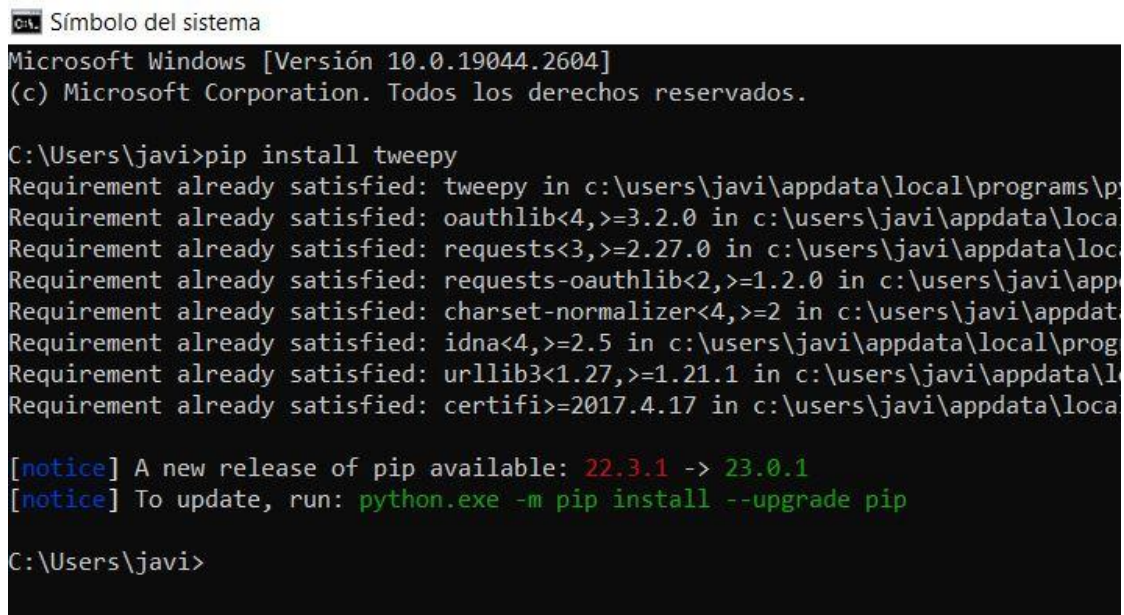
Antes de empezar con la descarga, se mostrarán las especificaciones técnicas del equipo en el que se llevará a cabo la descarga e instalación de estos programas. El dispositivo es un MSI GE66 Raider (2020), con las siguientes características:

- Sistema operativo: Windows 10 Pro.
- Procesador: Procesador Intel Core i7-10750H CPU 2.60GHz.
- Memoria: 64GB DDR4-3200.

- Gráficos: NVIDIA GeForce RTX 2060 con 6GB GDDR6

3.2.2 Instalación de Tweepy

En este paso se instalará Tweepy [18] en la consola de comandos de Windows, para ello se usará el comando `pip install + lo que se quiere instalar` (en este caso Tweepy) lo que responderá con el proceso de instalación y avisará en el caso de que haya disponible una nueva versión pip para actualizar (véase Figura 3-14)



```
ca. Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.2604]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\javi>pip install tweepy
Requirement already satisfied: tweepy in c:\users\javi\appdata\local\programs\p
Requirement already satisfied: oauthlib<4,>=3.2.0 in c:\users\javi\appdata\loca
Requirement already satisfied: requests<3,>=2.27.0 in c:\users\javi\appdata\loc
Requirement already satisfied: requests-oauthlib<2,>=1.2.0 in c:\users\javi\app
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\javi\appdat
Requirement already satisfied: idna<4,>=2.5 in c:\users\javi\appdata\local\prog
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\javi\appdata\l
Requirement already satisfied: certifi>=2017.4.17 in c:\users\javi\appdata\loca

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\javi>
```

Figura 3-14 Instalación de Tweepy

3.3 Proceso de instalación de Elasticsearch y Kibana

3.3.1 Descarga de Elasticsearch y Kibana

Lo primero será descargarse el archivo de instalación desde el sitio web oficial de Elasticsearch (<https://www.elastic.co>).

Ya en la página principal, se entrará en “Platform” y se hará clic en la pestaña de “Prueba gratuitas y descargas” (véase Figura 3-15).



Figura 3-15 Acceso a las aplicaciones de Elastic

Dentro de todas las opciones que se ofrecen (“Elastic Enterprise Search”, “Elastic Observability”, “Elastic Security”, “Elasticsearch”, “Kibana”, entre otras). Se realizará clic en los enlaces de “Elasticsearch” y “Kibana” (véase Figura 3-16).

El Elastic (ELK) Stack gratuito y abierto

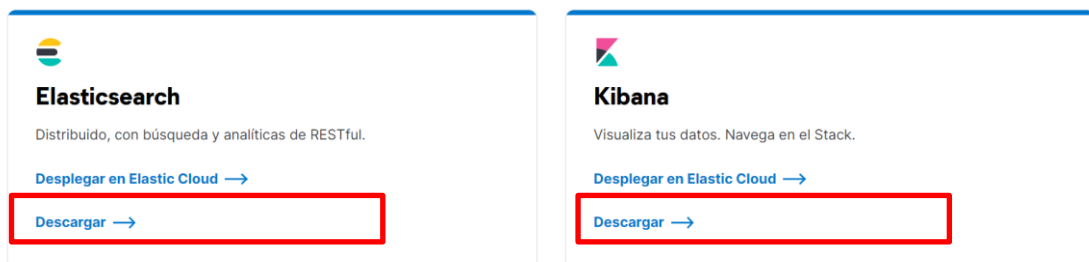


Figura 3-16 Selección de las aplicaciones de Elastic que se van a usar

Lo primero que se ofrece para elegir es la plataforma para la cual se descargarán las distintas aplicaciones. En este caso será para la plataforma Windows, pero ofrece una amplia variedad de posibilidades para descargas como puede ser macOS o Linux. A continuación, se da a elegir entre distintos formatos de descarga, para descargar la última versión disponible. En este caso se descargará la versión 7.16.3, siendo esta una versión anterior, la cual fue publicada el 13 de enero de 2022.

Para ello se tendrá que usar el enlace “View past releases” que se encuentra en la parte superior derecha de la ventana (véase Figura 3-17 y Figura 3-18).

Download Elasticsearch

1 Download and unzip Elasticsearch

Choose platform:

Windows

[Windows](#) [sha](#) [asc](#)

Package managers:

[yum.dnf.or.zypper](#) [apt-get](#)

Containers:

[Docker](#)

Summary

Version: 8.6.2

[View past releases](#)

Release date: February 16, 2023

[Detailed release notes](#)

License: Elastic License 2.0

[Elastic License 2.0](#)

Supported OS/JVM/Browser

[Support Matrix](#)

Notes:

Running on Kubernetes? Try [Elastic Cloud on Kubernetes](#).

Figura 3-17 Búsqueda de la versión de Elasticsearch que se va a usar

Download Kibana

1 Download and unzip Kibana

Choose platform:

Windows

[Windows](#) [sha](#) [asc](#)

Package managers:

[yum.dnf.or.zypper](#) [apt-get](#)

Containers:

[Docker](#)

Summary

Version: 8.6.2

[View past releases](#)

[Upgrade guidance](#)

Release date: February 16, 2023

[Detailed release notes](#)

License: Elastic License 2.0

[Elastic License 2.0](#)

Supported OS/JVM/Browser

[Support Matrix](#)

Notes:

Running on Kubernetes? Try [Elastic Cloud](#)

Figura 3-18 Búsqueda de la versión de Kibana que se va a usar

El formato que se necesita descargar para este proyecto es para el sistema operativo Windows (véase Figura 3-19 y Figura 3-20), que descargará un fichero comprimido “zip” que contendrá dentro de una carpeta todos los documentos necesarios para poder instalar estos programas.

Elasticsearch 7.16.3

[WINDOWS sha](#)

[MACOS X86_64 sha](#)

[MACOS AARCH64 sha](#)

[LINUX X86_64 sha](#)

[LINUX AARCH64 sha](#)

[DEB X86_64 sha](#)

[DEB AARCH64 sha](#)

[RPM X86_64 sha](#)

[RPM AARCH64 sha](#)

[MSI \(BETA\) sha](#)

Figura 3-19 Selección del formato de descarga de Elasticsearch



Figura 3-20 Selección del formato de descarga de Kibana

De esta manera se tendrá ya en el terminal las carpetas que contendrán los documentos necesarios para poder instalar los programas.

3.3.2 Instalar Elasticsearch

Una vez que se han descargado los ficheros comprimidos tanto de Elasticsearch como de Kibana, se accederá a ellos desde la ventana de comandos de Windows, en la que se usará el comando “*cd + el directorio de la carpeta a la que se quiere acceder*” (véase Figura 3-21).

Una vez allí se ejecutara Elasticsearch escribiendo en la ventana de comandos “*elasticsearch*”, dando pie así a que se empiecen a realizar una serie de comprobaciones y a activar una serie de “*plugins*” que dará como resultado la inicialización de Elasticsearch.

```

C:\> Símbolo del sistema - elasticsearch
Microsoft Windows [Versión 10.0.19044.2728]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\javi> cd C:\Users\javi\Desktop\python\elasticsearch-7.16.3\bin

C:\Users\javi\Desktop\python\elasticsearch-7.16.3\bin> elasticsearch
[2023-03-17T16:34:56,790][INFO ][o.e.n.Node ] [DESKTOP-LS1P6H6] version[7.16.3], pid[18776], build[default/10.0/amd64], JVM[Eclipse Adoptium/OpenJDK 64-Bit Server VM/17.0.1/17.0.1+12]
[2023-03-17T16:34:56,795][INFO ][o.e.n.Node ] [DESKTOP-LS1P6H6] JVM home [C:\Users\javi\Desktop\python\elasticsearch-7.16.3\bin]
[2023-03-17T16:34:56,796][INFO ][o.e.n.Node ] [DESKTOP-LS1P6H6] JVM arguments [-Des.networkaddress.cache.adultless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -XX:+ShowCodeDetailsInException, -Djna.nosys=true, -Ddio.netty.allocator.numDirectArenas=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx.enabled=true, -Djava.io.tmpdir=C:\Users\javi\AppData\Local\Temp\elasticsearch, -XX:+HeapDumpOnOutOfMemoryError, -XX:G1ReservePercent=25, -Delasticsearch, -Des.path.home=C:\Users\javi\Desktop\python\elasticsearch-7.16.3, -Des.path.conf=C:\Users\javi\Desktop\python\elasticsearch-7.16.3\config, -Des.distribution.type=zip, -Des.bundled_jdk=true]
[2023-03-17T16:34:59,258][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [aggs-matrix-stats]
[2023-03-17T16:34:59,259][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [analysis-common]
[2023-03-17T16:34:59,259][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [constant-keyword]
[2023-03-17T16:34:59,259][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [frozen-indices]
[2023-03-17T16:34:59,260][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [ingest-common]
[2023-03-17T16:34:59,260][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [ingest-geoip]
[2023-03-17T16:34:59,260][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [ingest-user-agent]
[2023-03-17T16:34:59,260][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [kibana]
[2023-03-17T16:34:59,261][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [lang-expression]
[2023-03-17T16:34:59,263][INFO ][o.e.p.PluginsService ] [DESKTOP-LS1P6H6] loaded module [lang-mustache]
    
```

Figura 3-21 Inicialización de Elasticsearch

Lo siguiente que se deberá hacer será la comprobación de que Elasticsearch está funcionando correctamente, y para ello se pondrá en el buscador “localhost:9200”, donde “localhost” hace referencia al equipo local en el que se está realizando la llamada a Elasticsearch, y “9200” es el puerto de escucha de Elasticsearch.

Una vez allí se devolverá el nombre del equipo que está ejecutando Elasticsearch, el nombre del clúster de Elasticsearch al que pertenece el nodo, la versión de Elasticsearch que se está usando y el lema de Elasticsearch (“*You Know, for Search*”) entre otros (véase Figura 3-22).

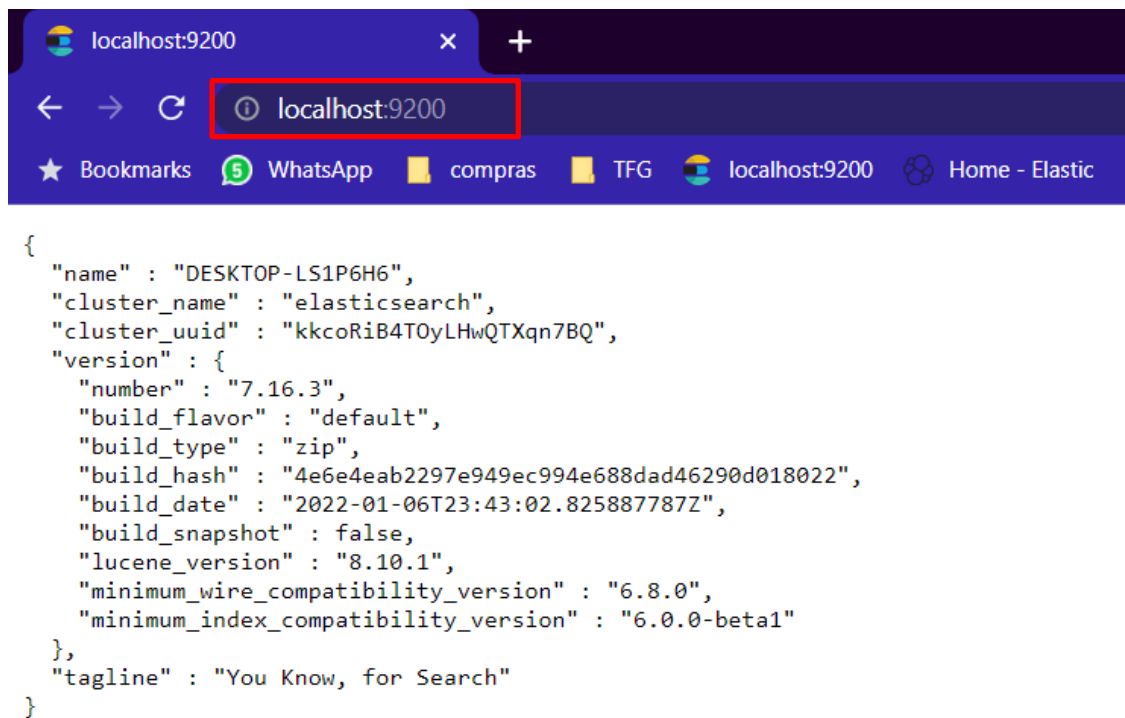


Figura 3-22 Página Elasticsearch

Cuando se haya comprobado que Elasticsearch está correctamente en funcionamiento, es recomendable cambiar el nombre que viene de fábrica en el clúster, el cual por defecto es *my-application*.

Para ello dentro de la carpeta donde se contienen todos los archivos de Elasticsearch se accederá a la carpeta “*config*” (véase Figura 3-23)

bin	06/01/2022 23:49	Carpeta de archivos
config	17/03/2023 12:07	Carpeta de archivos
data	17/03/2023 12:07	Carpeta de archivos
jdk	06/01/2022 23:49	Carpeta de archivos
lib	06/01/2022 23:49	Carpeta de archivos
logs	17/03/2023 16:34	Carpeta de archivos
modules	06/01/2022 23:50	Carpeta de archivos
plugins	06/01/2022 23:45	Carpeta de archivos
LICENSE.txt	06/01/2022 23:40	Documento de tex...
NOTICE.txt	06/01/2022 23:45	Documento de tex...
README.asciidoc	06/01/2022 23:40	Archivo ASCIIIDOC

Figura 3-23 Carpeta de Elasticsearch

Lo siguiente que se deberá hacer será acceder al fichero de configuración de Elasticsearch “*elasticsearch.yml*” (véase Figura 3-24)

📁 jvm.options.d	06/01/2022 23:45	Carpeta de archivos
📄 elasticsearch.keystore	17/03/2023 12:07	Archivo KEYSTORE
📄 ! elasticsearch.yml	17/03/2023 12:33	Archivo de origen ...
📄 ! elasticsearch-plugins.example.yml	06/01/2022 23:40	Archivo de origen ...
📄 jvm.options	06/01/2022 23:40	Archivo OPTIONS
📄 log4j2.properties	06/01/2022 23:44	Archivo de origen ...
📄 ! role_mapping.yml	06/01/2022 23:44	Archivo de origen ...
📄 ! roles.yml	06/01/2022 23:44	Archivo de origen ...
📄 users	06/01/2022 23:44	Archivo
📄 users_roles	06/01/2022 23:44	Archivo

Figura 3-24 Carpeta "config"

Esto abrirá el editor de código en el que en la línea 17 se cambiará el nombre del cluster de *my-application* a *twitter* (véase Figura 3-25). Cabe remarcar que tras hacer este paso será necesario que se reinicie Elasticsearch, para ello mientras que está en funcionamiento basta con que se pulse en la ventana de comandos del terminal, donde habíamos ejecutado Elasticsearch, *Ctrl+C* para que una vez detenido, volver a ejecutarlo una vez más.

```

1 # ===== Elasticsearch Configuration =====
2 #
3 # NOTE: Elasticsearch comes with reasonable defaults for most settings.
4 # Before you set out to tweak and tune the configuration, make sure you
5 # understand what are you trying to accomplish and the consequences.
6 #
7 # The primary way of configuring a node is via this file. This template lists
8 # the most important settings you may want to configure for a production cluster.
9 #
10 # Please consult the documentation for further information on configuration options:
11 # https://www.elastic.co/guide/en/elasticsearch/reference/index.html
12 #
13 # ----- Cluster -----
14 #
15 # Use a descriptive name for your cluster:
16 #
17 #cluster.name: my-application
18 #
19 # ----- Node -----
20 #
21 # Use a descriptive name for the node:
22 #
23 #node.name: node-1
24 #
25 # Add custom attributes to the node:
26 #
27 #node.attr.rack: r1
28 #
  
```

Figura 3-25 Cambio de nombre del cluster

3.3.3 Instalar Kibana

Para instalar Kibana se podría realizar de la misma manera que se ha utilizado para instalar Elasticsearch, pero existen otras maneras en las cuales no será necesario ejecutar todos los comandos desde la ventana de comandos.

Para ello se accederá a la carpeta “*bin*” que se encuentra dentro de la carpeta que se ha descargado de Kibana (véase Figura 3-26).

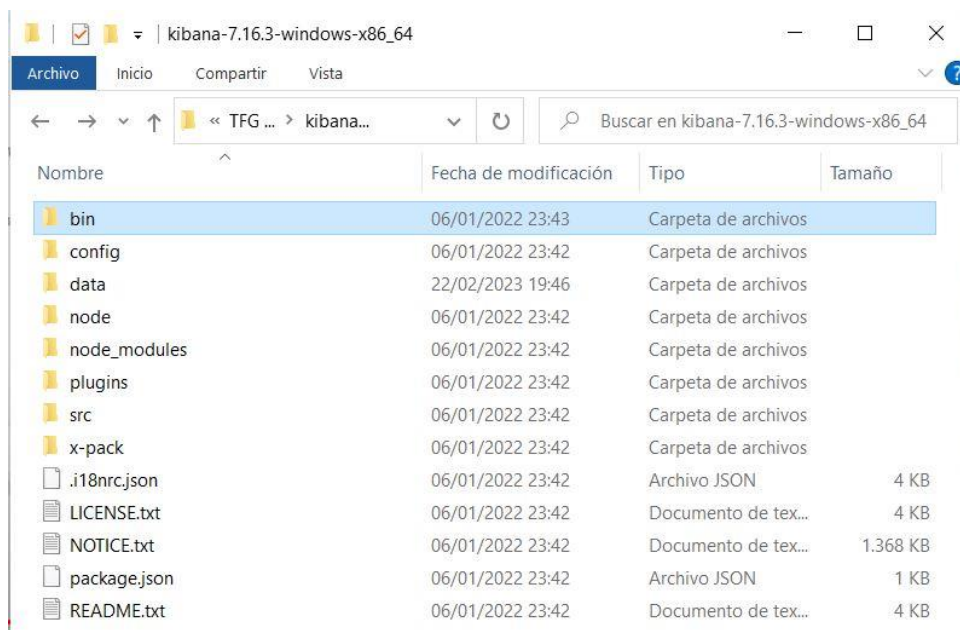


Figura 3-26 Acceso a la carpeta “bin”

Una vez que se encuentre dentro, seleccionará el archivo “kibana.bat” (véase Figura 3-27), el cual automáticamente abrirá la ventana de comandos del terminal y empezará a ejecutarlo.

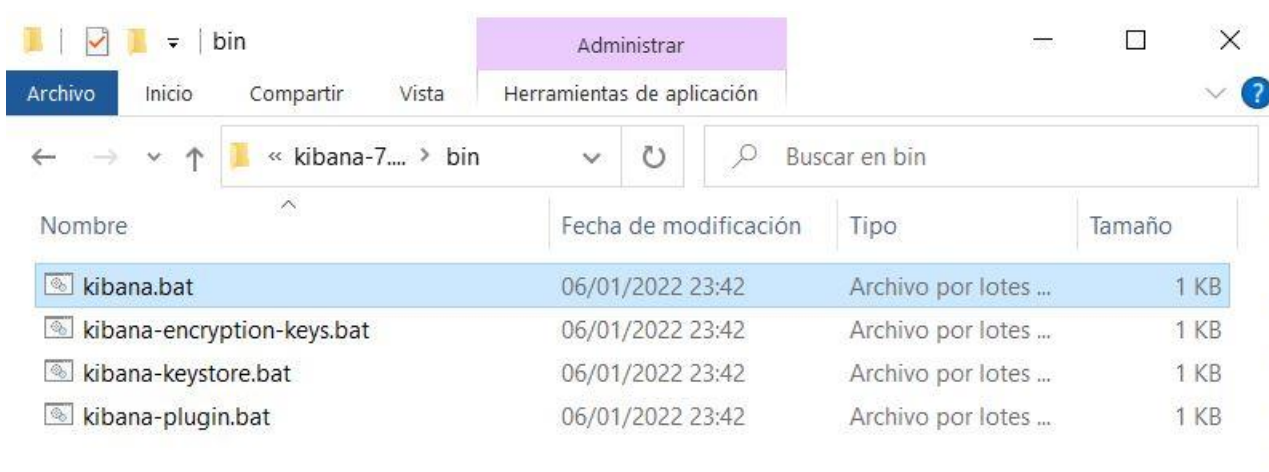


Figura 3-27 Inicialización de kibana.bat

Una vez el proceso de ejecución haya finalizado, al igual que se hizo con la instalación de Elasticsearch, se comprobará que Kibana se está ejecutando correctamente. Es importante remarcar que, si Elasticsearch no está en funcionamiento, no se podrá acceder a la página inicial de Kibana.

Para comprobar que Kibana está funcionando correctamente se usará el navegador de internet donde se escribirá “localhost:5601” (véase Figura 3-28), lo cual si se han realizado todos los pasos correctamente permitirá acceder a la página principal de Kibana.

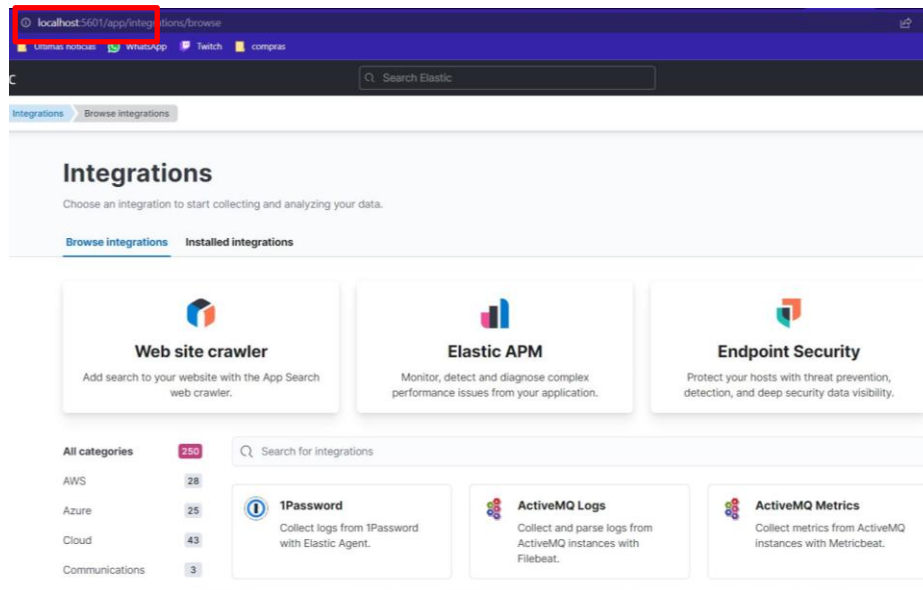


Figura 3-28 Página principal de Kibana

3.4 Conseguir la información de los tuits (.txt)

Para poder conseguir la información de los tuits que se usarán más adelante en Elasticsearch y Kibana se tendrá que haber instalado Tweepy en el ordenador y haber obtenido con anterioridad la cuenta de desarrollador de Twitter, en donde se habrá entregado una serie de claves de acceso, las cuales se tendrán que guardar, siendo la más relevante el “*Bearer_token*”.

Con este código de Python (véase Figura 3-29) se utiliza la biblioteca Tweepy y se busca conectarse a la API de Twitter, filtrar tuits en función de palabras clave específicas (etiquetas) y mostrarlos en la consola.

En las primeras líneas del código, se importa la biblioteca Tweepy y el módulo time. A continuación, hay que importar la cola y el submódulo de subprocesos desde la biblioteca estándar de Python.

En la línea 8, se definirá una variable “*bearer_token*” que contiene las credenciales de acceso a la API de Twitter. Estas credenciales se utilizan para autenticar nuestra conexión con la API.

En la línea 11, habría que crear una subclase llamada “*TweetPrinter*”. Esta clase hereda de la clase “*tweepy.StreamingClient*”, que es una clase que proporciona una interfaz para conectarse y recibir datos desde la API de Twitter.

En la línea 13, se definirá el método “*__init__*”. Este método inicializa la cola de subprocesos y los hilos que se utilizarán para procesar los tuits. También llama al constructor de la superclase “*tweepy.StreamingClient*” con el *token* de portador que definimos anteriormente.

En las líneas 23 a 32, se definirán tres métodos que se llamarán cuando se reciban tuits, errores o excepciones. Estos métodos simplemente imprimen información en la consola para ayudarnos a depurar y solucionar problemas.

En la línea 33, se tendrá que definir el método “*process_tweet*”, que es el que realmente procesará los tuits que se agregan a la cola. Este método está diseñado para ejecutarse en un subproceso separado, lo que significa que puede procesar varios tuits a la vez sin bloquear la recepción.

En la línea 42, se creará una instancia de la subclase “*TweetPrinter*” que se había definido con anterioridad, pasándole el *token* de portador que fue definido posteriormente.

```

1 import tweepy
2 import time
3
4 from queue import Queue
5 from threading import Thread
6
7 # NOTA: Introducir aquí las credenciales de acceso a la API
8 bearer_token = "A..."
9
10 # Hacer una subclase e implementar los métodos que se llaman cuando ocurre un evento: on_data, on_status
11 class TweetPrinter(tweepy.StreamingClient):
12
13     def __init__(self, bearer_token, wait_on_rate_limit = True, num_threads = 10, queue = Queue()):
14         super(TweetPrinter, self).__init__(bearer_token)
15         self.queue = queue
16         for i in range(num_threads):
17             t = Thread(target=self.process_tweet)
18             t.daemon = True
19             t.start()
20
21     # Versión simplificada para evitar problemas de procesamiento
22     # Referencias:
23     def on_tweet(self, tweet):
24         # Añadimos un campo al sistema con el instante de recepción de tweet
25         tweet.data['local_time_ns'] = time.time_ns()
26         self.queue.put(tweet)
27     def on_errors(self, errors):
28         print(errors)
29         self.disconnect()
30     def on_exception(self, exception):
31         print(exception)
32         self.disconnect()
33     def process_tweet(self):
34         while True:
35             tweet = self.queue.get()
36             # print(tweet.user.screen_name)
37             print(tweet.data)
38             print("===")
39             self.queue.task_done()
40
41     # Autenticación
42     stream = TweetPrinter(bearer_token=bearer_token)
43
44     # Obtener las reglas actualmente en activo y borrarlas
45     rules = stream.get_rules()[0]
46     to_delete = []
47     if rules is not None:
48         for rule in rules:
49             print(rule)
50             to_delete.append(rule[2])
51     stream.delete_rules(to_delete)
52
53     # Definir las reglas que se usaran para seleccionar tuits
54     stream.add_rules(tweepy.StreamRule('#UkraineRussiaWar -is:retweet lang:en', 'UkraineRussiaWar'))
55     stream.add_rules(tweepy.StreamRule('#UkraineWar -is:retweet lang:en', 'UkraineWar'))
56     stream.add_rules(tweepy.StreamRule('#Ukraine -is:retweet lang:en', 'Ukraine'))
57     stream.add_rules(tweepy.StreamRule('#Vuhledar -is:retweet lang:en', 'Vuhledar'))
58     stream.add_rules(tweepy.StreamRule('#Donbass -is:retweet lang:en', 'Donbass'))
59     stream.add_rules(tweepy.StreamRule('#WarinUkraine -is:retweet lang:en', 'WarinUkraine'))
60     stream.add_rules(tweepy.StreamRule('#Bakhmut -is:retweet lang:en', 'Bakhmut'))
61     stream.add_rules(tweepy.StreamRule('#RussianArmy -is:retweet lang:en', 'RussianArmy'))
62
63     stream.filter(tweet_fields=['context_annotations', 'created_at', 'author_id', 'lang', 'organic_metrics', 'entities'])

```

Figura 3-29 Código para la obtención de tuits

En las líneas 45 a 51, se obtendrán las reglas de filtrado de tuits que ya están activas en la API de Twitter y se borrarán. Esto asegurará que no se están duplicando reglas de filtrado y que se está empezando con una consulta limpia.

En las líneas 54 a 61, se definirán las reglas de filtrado de tuits que se seguirán. Estas reglas especifican ciertas palabras clave que resultan de interés. Si se encuentra un tuit que contenga una de estas palabras clave, se agregará a la cola para su posterior procesamiento.

Por último, en la línea 63, se utilizará la función “*stream.filter*” para iniciar la recepción de tuits y comenzar a filtrar y procesar los tuits en función de las reglas que se habían definido con anterioridad. Los tuits que coinciden con las reglas se mostrarán en la consola a medida que se procesen.

Este código hará que se genere la información necesaria para que se pueda escribir en un archivo .txt con la información sobre los tuits que Twitter nos envía en base a las reglas de filtrado indicadas (véase Figura 3-30).

```

1 StreamRule(value='UkraineRussiaWar -is:retweet lang:en', tag='UkraineRussiaWar', id='1626105655630200834')
2 StreamRule(value='UkraineWar -is:retweet lang:en', tag='UkraineWar', id='1626105661850361860')
3 StreamRule(value='Ukraine -is:retweet lang:en', tag='Ukraine', id='1626105667382566913')
4 StreamRule(value='Vuhledar -is:retweet lang:en', tag='Vuhledar', id='1626105672579354624')
5 StreamRule(value='Donbass -is:retweet lang:en', tag='Donbass', id='1626105678338162688')
6 StreamRule(value='WarinUkraine -is:retweet lang:en', tag='WarinUkraine', id='1626105683648081920')
7 StreamRule(value='Bakhmut -is:retweet lang:en', tag='Bakhmut', id='1626105689067208704')
8 StreamRule(value='RussianArmy -is:retweet lang:en', tag='RussianArmy', id='1626105694385577984')
9 {'author_id': '710814717250502656', 'created_at': '2023-02-16T06:27:08.000Z', 'edit_history_tweet_ids': ['1626105880646131712'], 'entities': {'annotations': [{'start': 40, 'end': 53,
10 ==
11 {'author_id': '15160748', 'context_annotations': [{'domain': {'id': '10', 'name': 'Person', 'description': 'Named people in the world like Nelson Mandela'}, 'entity': {'id': '11178189,
12 ==
13 {'author_id': '1565657060766896131', 'context_annotations': [{'domain': {'id': '131', 'name': 'Unified Twitter Taxonomy', 'description': 'A taxonomy of user interests. '}, 'entity': {
14 ==
15 {'author_id': '1466987376861974528', 'created_at': '2023-02-16T06:27:15.000Z', 'edit_history_tweet_ids': ['1626105912543830016'], 'entities': {'annotations': [{'start': 18, 'end': 24,
16 ==
17 {'author_id': '1623450865376829440', 'created_at': '2023-02-16T06:27:23.000Z', 'edit_history_tweet_ids': ['1626105943187501059'], 'entities': {'annotations': [{'start': 55, 'end': 60,
18 ==
19 {'author_id': '2428906852', 'created_at': '2023-02-16T06:27:26.000Z', 'edit_history_tweet_ids': ['1626105957271961601'], 'entities': {'annotations': [{'start': 14, 'end': 18, 'probabi
20 ==
21 {'author_id': '921385352299814912', 'context_annotations': [{'domain': {'id': '46', 'name': 'Business Taxonomy', 'description': 'Categories within Brand Verticals that narrow down the
22 ==
23 {'author_id': '1216078105', 'context_annotations': [{'domain': {'id': '123', 'name': 'Ongoing News Story', 'description': 'Ongoing News Stories like 'Brexit'"}, 'entity': {'id': '1484

```

Figura 3-30 Información obtenida de Twitter

3.5 Convertir .txt a .json

Es importante darse cuenta de que cada tuit obtenido está separado del siguiente tuit por “====”, es por ello por lo que se usará el siguiente código (véase Figura 3-31), en el cual se aprovechará para pasar el documento de .txt a formato JSON.

En la primera línea se importará la librería “json” para poder trabajar con este tipo de archivos, lo siguiente será abrir el archivo .txt en formato lectura (importante indicar el esquema de codificación usado, en este caso utf-8) y se indicará que este documento será el documento de entrada.

A continuación, se usará el comando “open” para que se abra el documento .json en modo escritura (este modo permitirá que se escriba en el documento seleccionado y en el caso de que no exista se generará uno nuevo) y codificación “utf-8” como documento de salida.

Será entonces cuando el código irá leyendo línea por línea el documento .txt y lo escribirá en el documento .json y en el caso de que la línea que esté leyendo sea distinta a “====” escribirá su contenido en la salida, y en el caso de que no sea así, continua con la siguiente línea del bucle sin hacer nada.

Esto generará un problema que habrá que solucionar más adelante dado que el documento .json no estará formado únicamente por un json si no que estará compuesto por varios, uno por tuit, esto se llama “json array”.

```

json_profe > pasar_de_txt_a_json.py > ...
1 import json
2
3 with open("fileExample.txt", "r",encoding="utf-8") as file_input:
4     with open("tuits.json", "w",encoding="utf-8") as output:
5         for line in file_input:
6             if line != "===\n":
7                 output.write(line)
    
```

Figura 3-31 Código para limpiar el documento

3.6 Seleccionar los diccionarios que nos interesan del archivo .json

Lo primero que hay que hacer es ver y entender la estructura que tendrán los distintos tuits recogidos con Tweepy, es por ello por lo que se ha realizado un análisis de diversos tuits del documento .json, del cual podemos deducir la estructura de la información proporcionada por Twitter (véase Figura 3-32).

No todo lo que ha devuelto el código de Tweepy será de interés para nuestros propósitos, es por ello que habrá que analizar qué información es útil para el proyecto y cuál no.

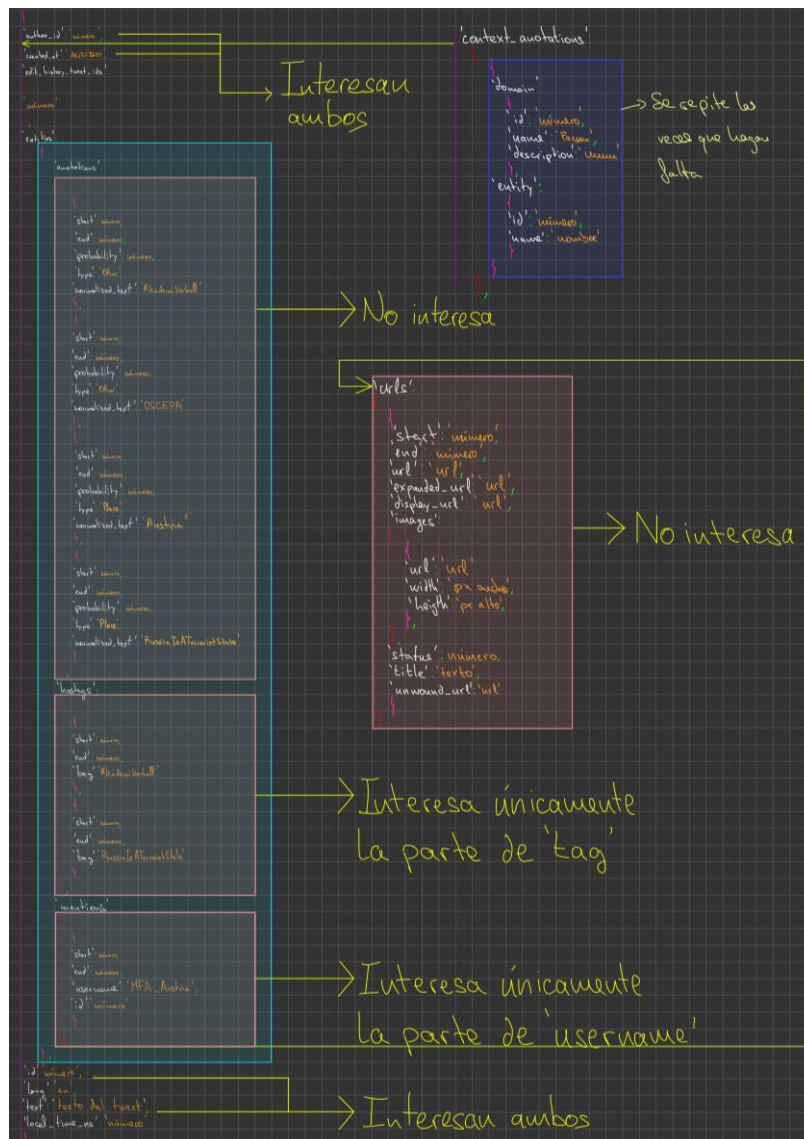


Figura 3-32 Estructura de los tuits recogidos

Para limpiar el documento .json de la información que no es necesaria y convertirlo en un documento .json compuesto por un json únicamente por los datos relevantes se usará el siguiente código Python (véase Figura 3-33), el cual lee un archivo json, procesa los datos y luego escribe un nuevo archivo json con los datos procesados.

El programa recorrerá cada línea del archivo y la procesará. Primero eliminará los caracteres de salto de línea y espacios en blanco innecesarios utilizando los métodos "*rstrip*" y "*replace*".

Luego, utiliza la función "*eval*" para convertir la línea de texto en un objeto Python. Si el objeto Python contiene la clave "*entities*", el programa extrae las etiquetas "*hashtags*", menciones y anotaciones del tuit y los almacena en listas separadas. Estos datos se utilizan para crear un nuevo objeto Python que contiene solo la información relevante del tuit, como el identificador del autor, la fecha de creación, el texto del tuit y las listas de "*hashtags*", menciones y anotaciones.

El objeto de tuit procesado se agrega a la lista "*output_data*", que contiene todos los tuits procesados.

Finalmente, el programa escribe el contenido de la lista "*output_data*" en un archivo json utilizando la función "*json.dump*". El argumento "*indent=2*" se utiliza para formatear el archivo json generado de manera legible para humanos con una sangría de 2 espacios.


```
1 import sys
2 import json
3
4 output_data = []
5
6 with open("tuits.json", encoding="utf-8") as file:
7     for line in file:
8         clean_line = line.rstrip().replace("\n", " ")
9         try:
10            data = eval(clean_line)
11            if "entities" in data:
12                hashtags = []
13                if "hashtags" in data["entities"]:
14                    etiquetas = data["entities"]["hashtags"]
15                    for etiqueta in etiquetas:
16                        hashtags.append(etiqueta['tag'])
17                mentions = []
18                if "mentions" in data["entities"]:
19                    menciones = data["entities"]["mentions"]
20                    for mencion in menciones:
21                        mentions.append(mencion['username'])
22                annotations = []
23                if "annotations" in data["entities"]:
24                    anotaciones = data["entities"]["annotations"]
25                    for anotacion in anotaciones:
26                        annotations.append(anotacion['normalized_text'])
27
28                tweet_data = {
29                    "author_id": data['author_id'],
30                    "created_at": data['created_at'],
31                    "hashtags": hashtags,
32                    "mentions": mentions,
33                    "annotations": annotations,
34                    "text": data['text']
35                }
36
37                output_data.append(tweet_data)
38
39            except SyntaxError:
40                continue
41
42 with open("tweets_output.json", "w", encoding="utf-8") as output:
43     json.dump(output_data, output, indent=2)
```

Figura 3-33 Código para la depuración del archivo .json

3.7 Introducir .json a Elasticsearch creando sus índices y *mapping*

Una vez que se ha generado el archivo .json ya depurado y conteniendo únicamente la información interesante para el trabajo, se tendrá que indexar en Elasticsearch para poder trabajar desde Kibana sobre la información del archivo.

Para ello se desarrollará un código en Python que genere estos índices y *mapping* necesarios para poder indexar el archivo a Elasticsearch (véase Figura 3-34).

```

3  import sys
4  import json
5
6  from elasticsearch import Elasticsearch
7
8  # Indicamos el nombre del índice que queremos utilizar.
9  _index_name="indices_de_los_tuits"
10
11 # Inicializamos la conexión con Elasticsearch, indicando host y puerto del servidor
12 elastic = Elasticsearch("http://localhost:9200")
13
14 # Primero se empieza borrando el índice, por si ya existiese alguna prueba anterior. Se ignoran los errores.
15 elastic.indices.delete(index=_index_name, ignore=404)
16
17 # Configuramos el índice que vamos a crear
18 _settings = {
19     'number_of_shards': 2,
20     'number_of_replicas': 1
21 }
22
23 _mappings = {
24     'properties': {
25         "author_id": { "type": "keyword" },
26         "created_at": { "type": "date" },
27         "hashtags": { "type": "text" },
28         "mentions": { "type": "text" },
29         "annotations": { "type": "text" },
30         "text": { "type": "text" }
31     }
32 }
33
34 # Creamos el índice
35 elastic.indices.create(index=_index_name, settings = _settings, mappings = _mappings)
36
37 # Contamos el número de documentos procesados
38 counter = 0
39
40 # Procesamos el fichero de datos cargando los tuits
41 with open("tweets_output.json", "r", encoding="utf-8") as jsonfile:
42     tuits = json.load(jsonfile)
43     for tuit in tuits:
44         response = elastic.index(index=_index_name, document=tuit)
45         print(tuit['created_at'])
46         counter = counter+1
47
48 # Finalmente, imprimimos el número de documentos procesados.
49 print(counter)

```

Figura 3-34 Código para indexar un archivo .json a Elasticsearch y crear sus índices y *mapping*

Este código de Python utiliza la biblioteca Elasticsearch para crear un índice y almacenar los datos de un archivo .json. A continuación, se describe el código línea por línea:

1. Se importan las librerías que se van a utilizar, incluyendo "sys" y "json" para trabajar con el archivo json, y Elasticsearch para conectarse a Elasticsearch.
2. Se establece el nombre del índice que se creará y utilizará en Elasticsearch.
3. Se establece la conexión con Elasticsearch, especificando el host y el puerto del servidor.
4. Se utiliza la función "delete" de Elasticsearch para eliminar cualquier índice previamente creado con el mismo nombre. Se utiliza el parámetro "ignore=404" para ignorar cualquier error en caso de que el índice no exista.
5. Se define la configuración del índice a crear utilizando un diccionario. En este caso, se establecen el número de fragmentos y réplicas.

6. Se define el mapeo de campos para el índice a crear. Se especifican los campos que se utilizarán para almacenar la información de los tuits, incluyendo el identificador del autor, la fecha de creación, las listas de etiquetas “*hashtags*”, menciones y anotaciones, y el texto del tuit.
7. Se crea el índice utilizando la función “*create*” de Elasticsearch. Se utilizan los valores del nombre del índice, la configuración y el mapeo de campos definidos anteriormente.
8. Se inicializa el contador de documentos procesados.
9. Se abre el archivo .json que contiene los tuits procesados utilizando la función “*open*” de Python. Se utiliza “*json.load*” para cargar los datos del archivo .json en una variable.
10. Se recorre la lista creada para cada tuit. Se utiliza la función “*index*” de Elasticsearch para agregar cada tuit al índice. El parámetro “*index*” se establece con el nombre del índice creado anteriormente y el parámetro “*document*” se establece con el tuit actual.
11. Se imprime la fecha de creación del tuit procesado.
12. Se incrementa el contador de documentos procesados.
13. Finalmente, se imprime el número total de documentos procesados.

4 RESULTADOS / VALIDACIÓN / PRUEBA

4.1 Comprobación de los índices

Después de haber iniciado Elasticsearch y Kibana como se ha visto anteriormente, lo primero que habría que hacer es comprobar de la existencia de los índices. Para ello habría que introducir la URL “localhost:5601” en el navegador. Una vez en la página principal de Kibana, si se desliza la pestaña de la ventana desplegable superior izquierda se podrá ver la opción de “Dev Tools”, la cual desplegará una nueva pestaña en donde habrá que entrar a “Index Management” (véase Figura 4-1).

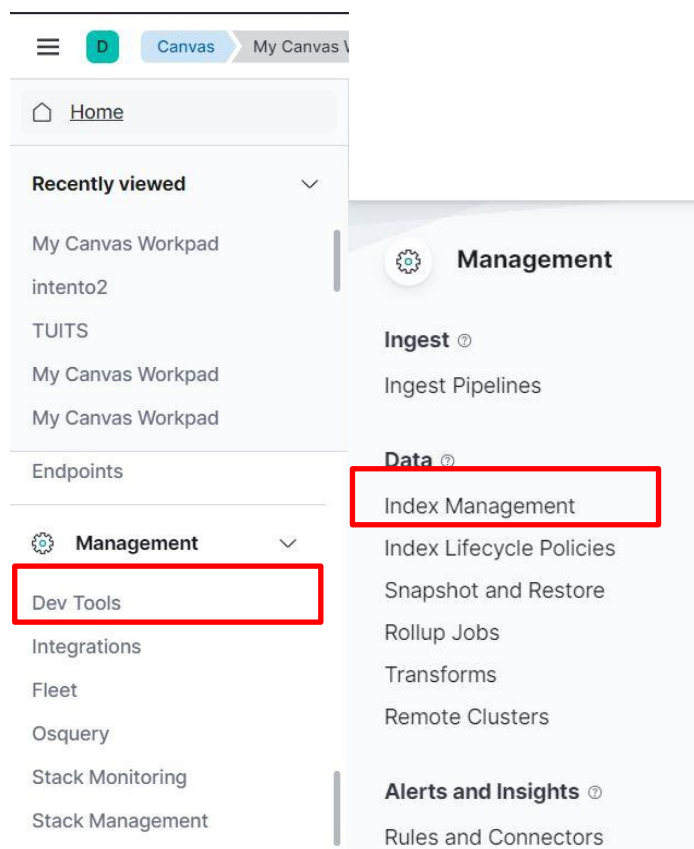


Figura 4-1 Acceso a los índices existentes

De esta manera se desplegará una ventana en donde se podrá ver todos los índices que se han creado y comprobar su estado, bien sea el número de “Shards”, el número de “Réplicas” o que el número de documentos que se han indexado sean los que deberían (véase Figura 4-2).

Indices Data Streams Index Templates Component Templates

Update your Elasticsearch indices individually or in bulk. [Learn more.](#) Include rollup indices Include hidden indices

Search Reload indices

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
indices_de_los_tuits	● yellow	open	2	1	2722	1.7mb	

Figura 4-2 Comprobación de los índices creados

4.2 Creación de representaciones gráficas

Una vez que se han comprobado los índices existentes, lo siguiente será empezar a crear las representaciones gráficas que se crean necesarias para reflejar la información obtenida de los tuits.

El proceso que se seguirá será entrar en la misma ventana que se abrió en el principio y pulsar en la pestaña “Dashboard” (véase Figura 4-3)

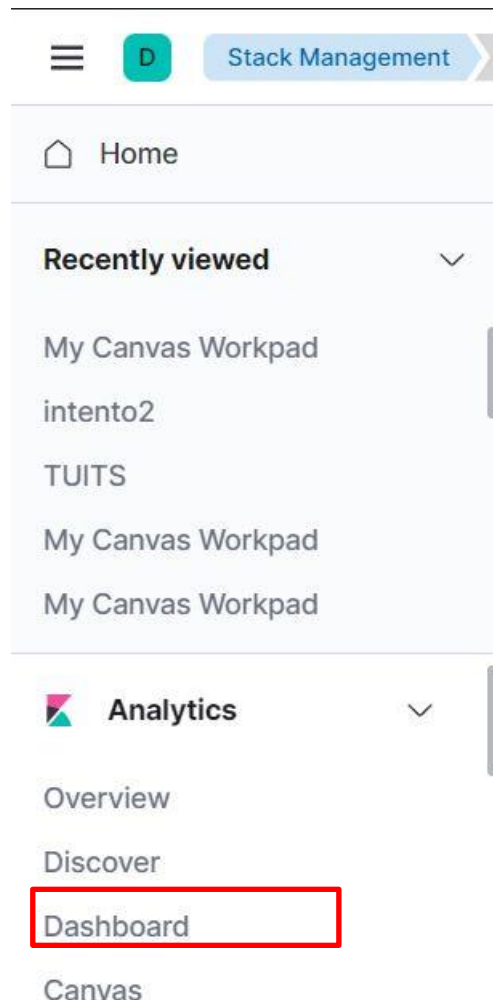


Figura 4-3 Pestaña de creación de "Dashboard"

Esto permitirá entrar a la ventana de creación de “Dashboard”, en donde habría que ajustar la fecha de la creación de los distintos tuits que hemos recogido. Lo siguiente sería ser crear el Dashboard en el que trabajaremos, pulsando en “Create visualization” (véase Figura 4-4).

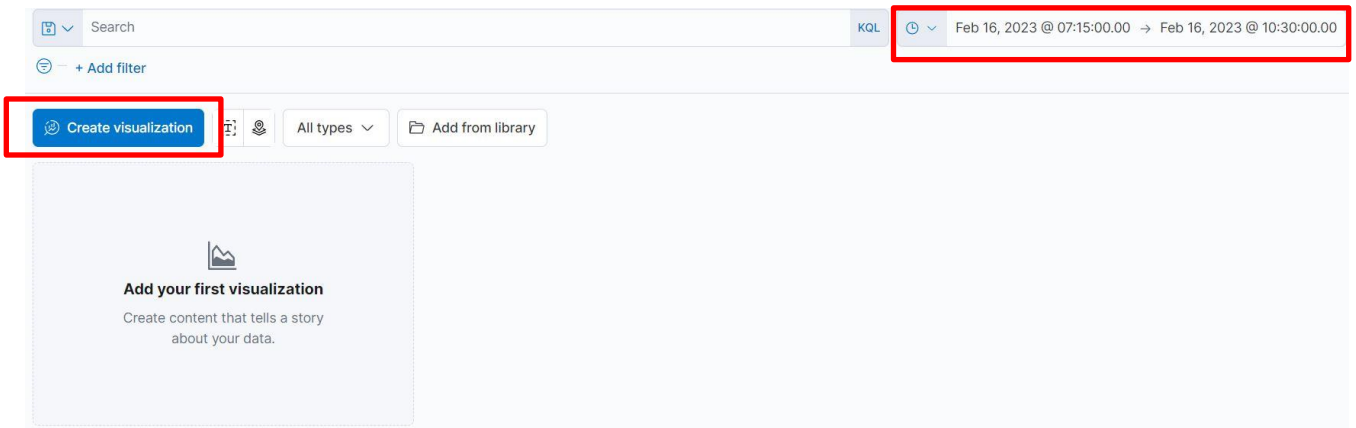


Figura 4-4 Creación de un nuevo Dashboard

Una vez que se haya seleccionado la opción para crear un Dashboard se podrá elegir la manera de representar la información, en este caso, un gráfico de barras (véase Figura 4-5).

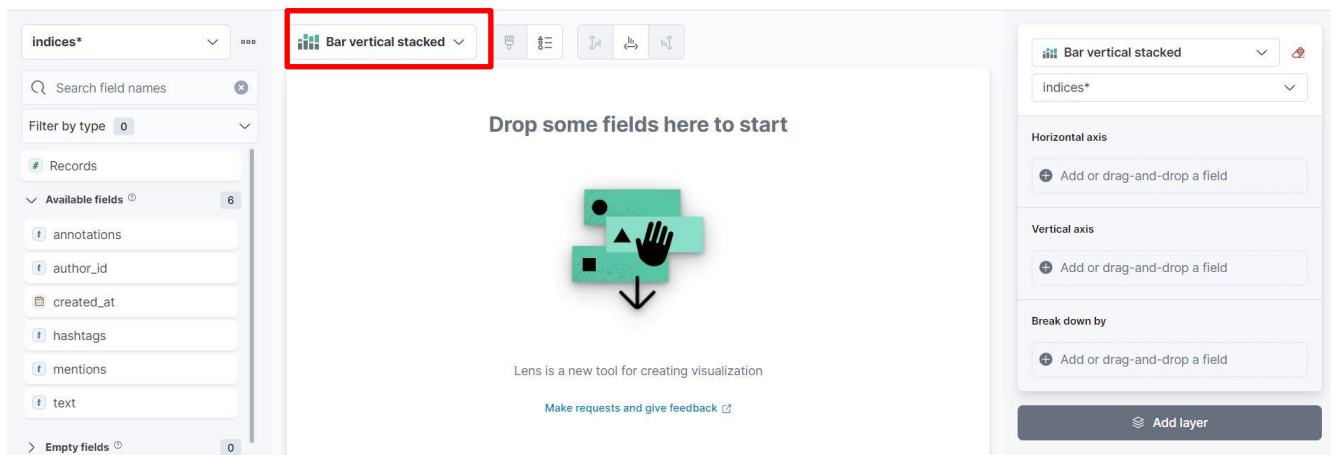


Figura 4-5 Creación de un gráfico de barras

Lo siguiente sería elegir los valores a usar en el eje X e Y, para representar el número de tuits que se han publicado en función del tiempo. Para ello se pulsará en “Vertical axis” y “Horizontal axis” y el campo que se usará en ambos ha de ser “created_at”, siendo “Unique count” la función a seleccionar para el eje Y (véase Figura 4-6).

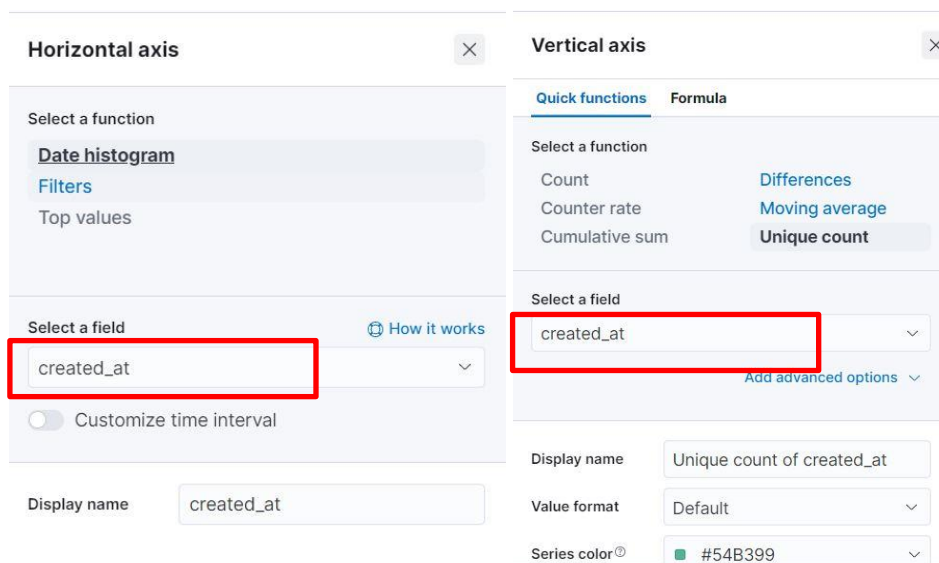


Figura 4-6 Selección de los valores de los ejes

De esta manera se obtendrá un gráfico de barras (véase Figura 4-7) que representa el número de tuits publicados cada 15 minutos. Se puede ver, por ejemplo, que en el periodo que comprende de las 9:15 a las 9:30, se han publicado 250 tuits.

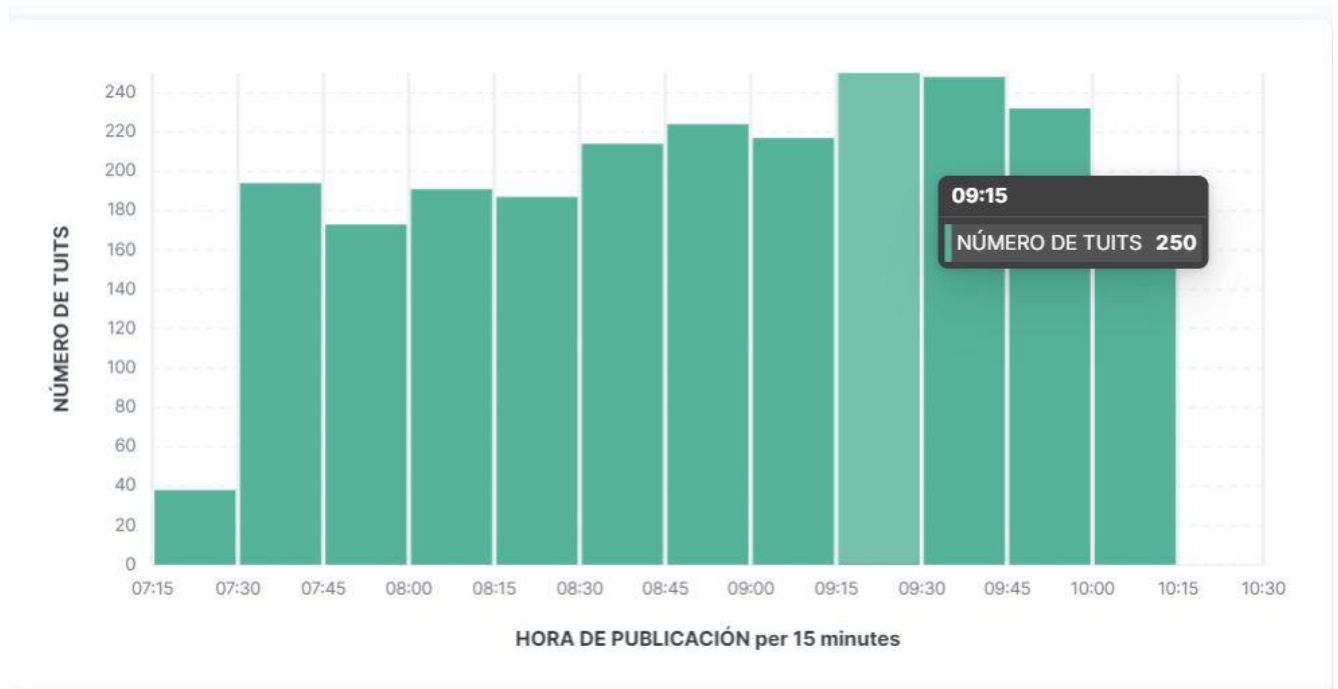


Figura 4-7 Gráfico de barras obtenidos

Ahora se creará un gráfico de tipo donut, para ello habrá que dirigirse nuevamente a la pestaña desplegable, sección “Proportion” y más específicamente a la opción de “Donut” (véase Figura 4-8).

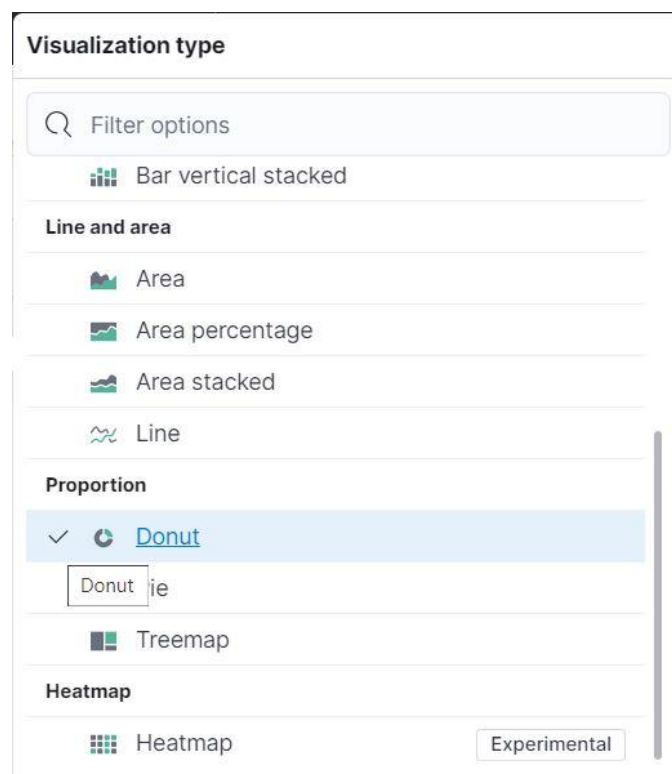


Figura 4-8 Selección del gráfico de tipo Donut

Una vez que se haya seleccionado el gráfico de tipo donut habrá que elegir el campo que se va a usar. En este caso serán los “hashstags” incluidos en los tuits, y el número de valores que se quieren representar (véase Figura 4-9).

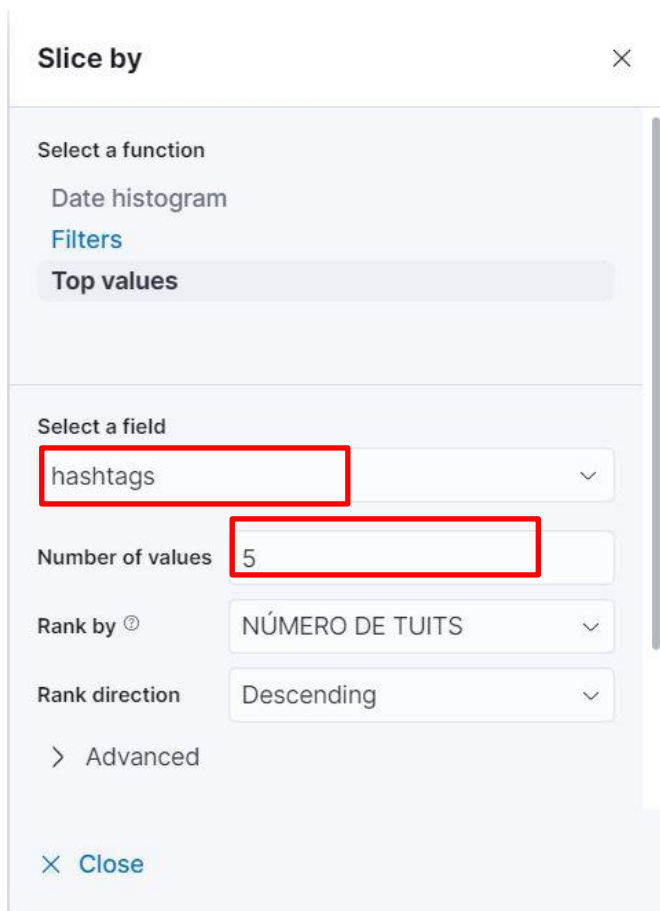


Figura 4-9 Selección de los campos a usar

De esta manera se generará un gráfico que contiene las 6 etiquetas que más se han usado en los tuits indexados, mostrando los valores en porcentaje respecto al total. Se puede comprobar que la gente hace muchas más menciones a Ucrania que a cualquier otro país, lo cual podría hacer pensar que la gente de Twitter apoya en mayor medida a este país (véase Figura 4-10).

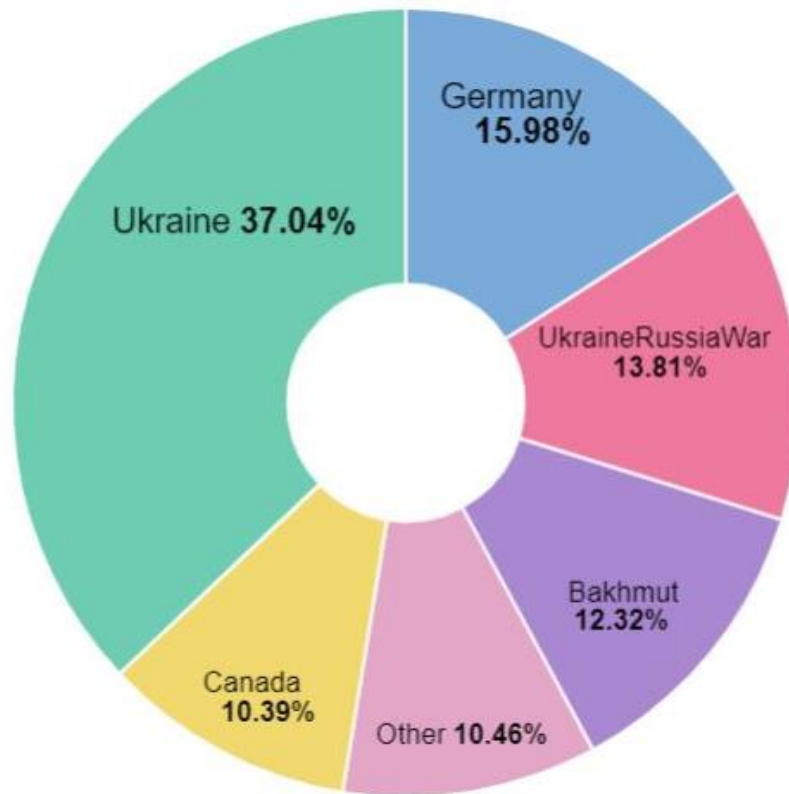


Figura 4-10 Porcentaje de uso de las 6 etiquetas más usadas

La siguiente representación que se realizará será una comparativa entre las etiquetas de Ucrania y Rusia. Para ello, habrá que seleccionar el tipo de gráfico “Line” (véase Figura 4-11), que permitirá seleccionar valores para el eje X y el eje Y.

Figura 4-11 Creación de un gráfico de líneas

Para lograr el objetivo de esta comparativa se seleccionará en el eje X el campo “created_at” (véase Figura 4-12) y estableceremos el nombre del eje para que sea fácil de identificar los datos que se representan para su estudio.

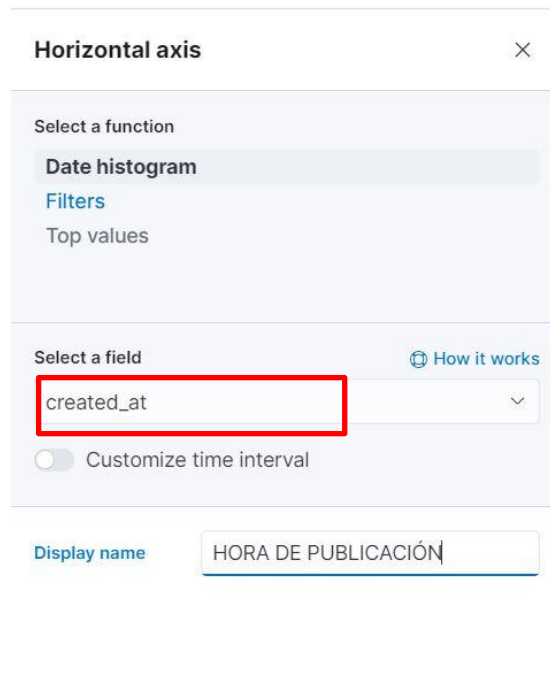


Figura 4-12 Establecimiento de los valores del eje X

Para el eje Y se establecerán 2 valores. En ambos se establecerá el campo “hashtags”, pero filtrando distintas etiquetas a través de la opción de “Filter by” y escribiendo “hashtags:”*lo que queremos filtrar*”. En uno de ellos se filtrarán las etiquetas que contengan la palabra “Russia” y en el otro las etiquetas que contenga la palabra “Ukraine” (véase Figura 4-13).

También es importante indicar el nombre por el cual se identificará cada filtro creado y seleccionar la función “Unique count”.

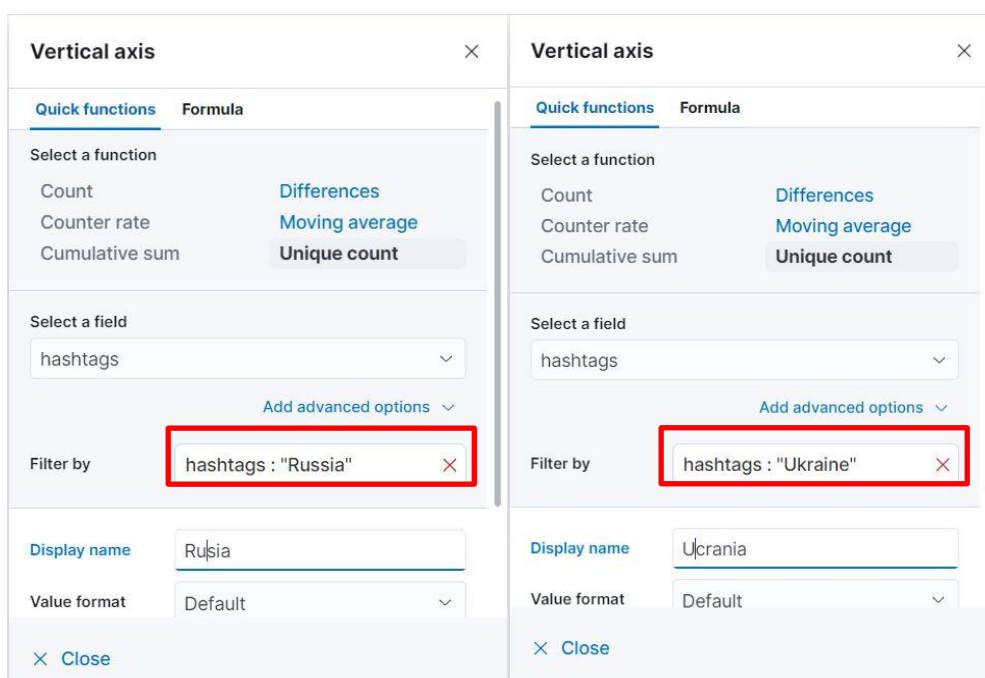


Figura 4-13 Filtros para los ejes X e Y

De esta manera se obtendrá un gráfico que reflejará el número de publicaciones por hora en la que se etiqueta a Rusia y a Ucrania. Se puede observar en el ejemplo que Ucrania estaba siendo más etiquetada en las publicaciones que Rusia, siendo a las 08:50 la hora en la que más se mencionó a Ucrania, con un total de 110 menciones frente a las 52 veces que se etiquetó a Rusia en el mismo periodo (véase Figura 4-14).

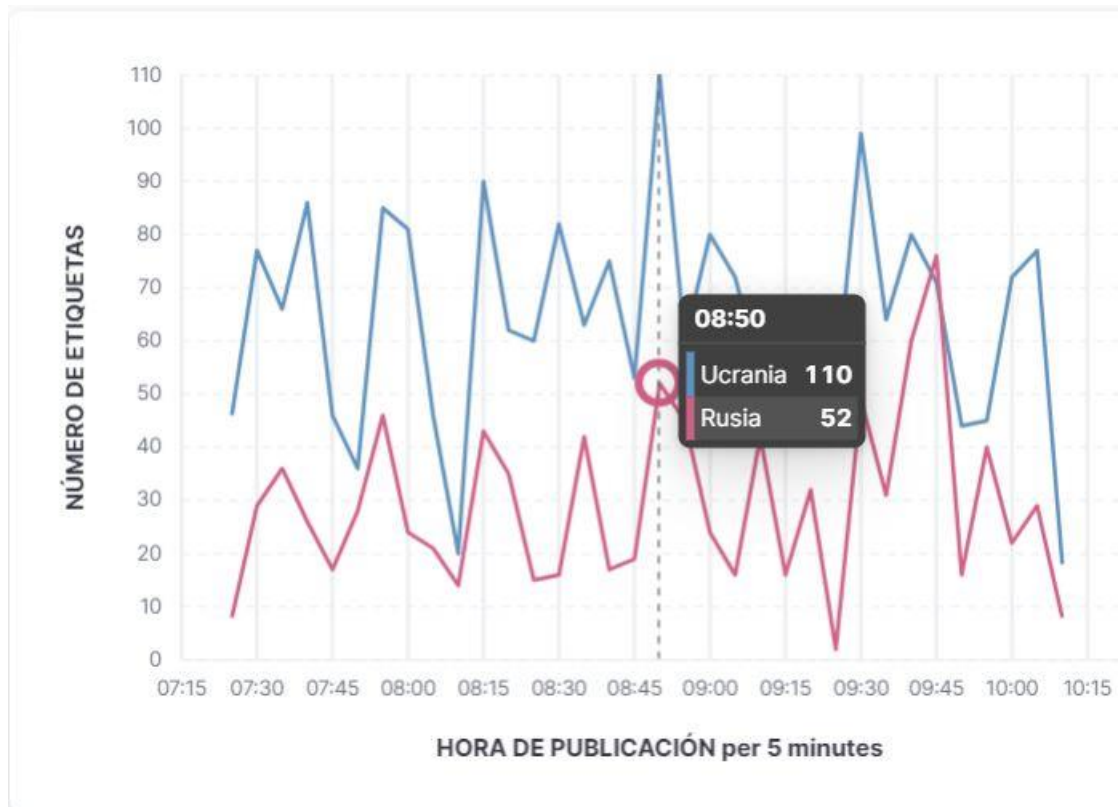


Figura 4-14 Gráfico de comparación de etiquetas

Por último, se procederá a crear una tabla que represente cuáles son los usuarios más mencionados del día. Para ello se elegirá la opción de “Table” (véase Figura 4-15), la cual contiene los parámetros “Rows”, “Columns” y “Metrics”.

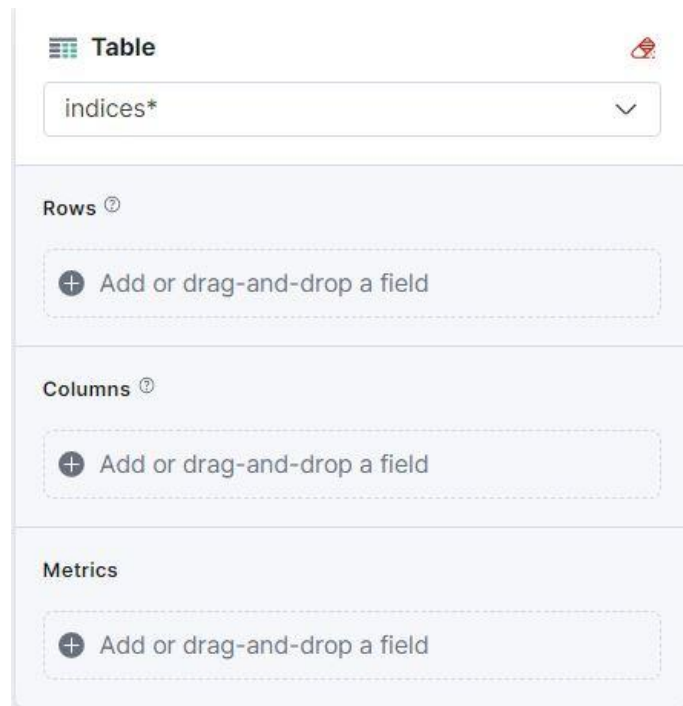


Figura 4-15 Creación de la tabla

Para este caso únicamente habrá que rellenar los parámetros de “Rows” (filas) y “Metrics” (parámetros). En las filas se establecerá que el campo que interesa para esta tabla serán las menciones y se indicará que muestre únicamente 10 menciones. En los parámetros de la tabla se seleccionará la función de conteo de los registros existentes (véase Figura 4-16).

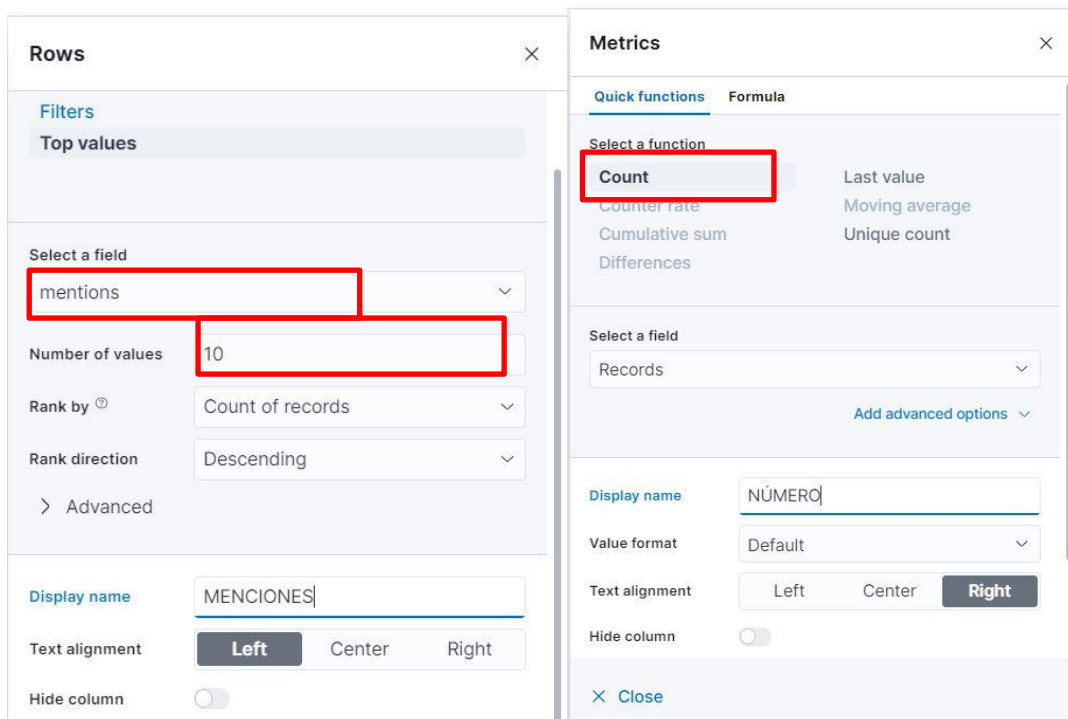


Figura 4-16 Establecimiento de los parámetros de la tabla

De esta manera se obtendrá una tabla que mostrará los 10 usuarios más mencionados y el número de veces que han sido mencionados. Cabe destacar que, en el ejemplo, los 3 más mencionados son el presidente ucraniano, Zelensky, el representante de la Unión Europea para Asuntos Exteriores y Política

de Asuntos Exteriores, Josep Borrell, y el Secretario general de la OTAN, Jens Stoltenberg (véase Figura 4-17).

MENCIONES	NÚMERO
jensstoltenberg	56
JosepBorrellF	35
ZelenskyyUa	32
AndrzejDuda	29
nexta_tv	29
vonderleyen	27
GlasnostGone	25
SlavaUk30722777	23
POTUS	21
HannaLiubakova	20

Figura 4-17 Tabla resultado de los usuarios más mencionados

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

En este trabajo se ha presentado el desarrollo de una plataforma OSINT para el almacenamiento y análisis inteligente de datos de la red social Twitter, enfocada en la recolección de datos relacionados con la guerra en Ucrania. A través de la implementación de herramientas de minería y visualización de datos, se ha logrado extraer y representar información valiosa de los datos recolectados, permitiendo obtener una mejor comprensión del conflicto y sus implicaciones.

La plataforma desarrollada ha demostrado ser eficaz en la recolección y análisis de datos de Twitter, permitiendo obtener información relevante y actualizada sobre el conflicto en Ucrania. La visualización de datos en Kibana ha sido una herramienta clave para representar de manera clara y concisa los hallazgos obtenidos a través de la plataforma, lo que facilita su interpretación por parte de los usuarios finales.

Aun así, hay destacar que, debido a la naturaleza dinámica de las redes sociales, la información obtenida a través de esta plataforma debe ser considerada como una instantánea del momento en que fue recolectada y puede cambiar rápidamente.

5.2 Líneas futuras

Para futuras mejoras de la plataforma OSINT para el almacenamiento y análisis inteligente de datos de Twitter enfocada en la guerra de Ucrania, se podrían explorar diferentes líneas de investigación. Una de ellas sería mejorar la actualización constante de los datos, mediante la incorporación de nuevas fuentes de información y la implementación de técnicas de recolección de datos en tiempo real. Además, se podría investigar la posibilidad de incluir nuevas técnicas de procesamiento de lenguaje natural para mejorar la precisión de los datos y la eficiencia de la plataforma. También se podrían considerar nuevas formas y extender sus capacidades de análisis. Con estas mejoras, se podría obtener una visión más completa y actualizada del conflicto en Ucrania, lo que permitiría a los usuarios finales obtener una mejor comprensión de las implicaciones políticas, sociales y económicas del mismo.

6 BIBLIOGRAFÍA

- [1] S.Kemp, «Hootsuite,» 15 agosto 2022. [En línea]. Available: <https://blog.hootsuite.com/es/informe-digital-estadisticas-de-redes-sociales/>. [Último acceso: 17 enero 2023].
- [2] Hotmart, «Hootmart,» 27 septiembre 2022. [En línea]. Available: <https://hotmart.com/es/blog/que-es-twitter>. [Último acceso: 17 enero 2023].
- [3] S.Pritchard, «The Daily Swing,» 14 septiembre 2021. [En línea]. Available: <https://www.sentinelone.com/cybersecurity-101/open-source-intelligence-osint/>. [Último acceso: 17 enero 2023].
- [4] A. NAVEIRA, 26 abril 2022. [En línea]. Available: <https://marketing4ecommerce.net/historia-de-twitter/#:~:text=La%20historia%20de%20Twitter%20m%C3%A1s,quedaba%20alojado%20en%20la%20Nube..> [Último acceso: 24 enero 2023].
- [5] M. Bleger, «crehana,» 26 abril 2022. [En línea]. Available: <https://www.crehana.com/blog/transformacion-digital/historia-twitter/>. [Último acceso: 24 enero 2023].
- [6] L. Wamsley, «Capradio,» 27 septiembre 2017. [En línea]. Available: <https://www.capradio.org/news/npr/story?storyid=553982675>. [Último acceso: 24 enero 2023].
- [7] R. Chicharro, «ThinkBig,» 6 febrero 2023. [En línea]. Available: <https://blogthinkbig.com/la-historia-de-twitter-dia-en-el-que-todo-cambio-en-la-red-social/>. [Último acceso: 13 febrero 2023].
- [8] «centro de ayuda Twitter,» [En línea]. Available: <https://help.twitter.com/es/using-twitter/types-of-tweets>. [Último acceso: 13 febrero 2023].
- [9] A. Garrido, «Twitter. Redes sociales en educación,» [En línea]. Available: http://twitter.antoniogarrido.es/menciones_y_retweets.html. [Último acceso: 07 febrero 2023].
- [10] «Atribus,» [En línea]. Available: <https://atribus.com/top-tweets-compartidos-2021/>. [Último acceso: 07 febrero 2023].

- [11] J. Bautista, «CCM,» 19 julio 2022. [En línea]. Available: <https://es.ccm.net/aplicaciones-e-internet/redes-sociales-y-mensajeria/3976-que-significa-retwittear-y-para-que-sirve/>. [Último acceso: 13 febrero 2023].
- [12] L. M. Rodríguez, «LMR,» 3 noviembre 2016. [En línea]. Available: <https://luciamonterorodriguez.com/el-uso-de-me-gusta-en-twitter/>. [Último acceso: 13 febrero 2023].
- [13] A. Tosiello, «SocialGest,» 2020. [En línea]. Available: <https://blog.socialgest.net/es/2021/04/18/como-crear-hilo-twitter/>. [Último acceso: 13 febrero 2023].
- [14] Y. Fernández, «Xataka,» 17 junio 2017. [En línea]. Available: <https://www.xataka.com/basics/que-es-una-cuenta-verificada-de-twitter>. [Último acceso: 13 febrero 2023].
- [15] F. Schaurer y J. Störger, «The Evolution of Open Source Intelligence,» OSINT Report 2010, Zurich, 2010.
- [16] «ODIN,» 21 febrero 2022. [En línea]. Available: <https://odint.net/osint-y-el-ciclo-de-an%C3%A1lisis-de-inteligencia/>. [Último acceso: 13 febrero 2023].
- [17] «Elasticsearch,» [En línea]. Available: <https://www.elastic.co/es/what-is/elasticsearch>. [Último acceso: 14 febrero 2023].
- [18] bliti, «Tweepy,» [En línea]. Available: <https://www.tweepy.org/>. [Último acceso: 30 marzo 2023].
- [19] J. Rodríguez y V. Fernández, *Cómo redactar el estado del arte de un trabajo*, Editorial Genios, 2010.
- [20] R. ARTURO, «Cision,» 20 marzo 2008. [En línea]. Available: <https://www.prweb.com/releases/2008/03/prweb786794.htm>. [Último acceso: 24 enero 2023].

ANEXO I: IMPLICACIONES SOCIALES, Y/O ECONÓMICAS, Y/O AMBIENTALES

Implicaciones sociales

Potencial para la manipulación

El mal uso de las herramientas de análisis y comunicación social podría desembocar en implicaciones negativas, como el potencial para la manipulación de la opinión pública. Si las entidades, gobiernos u organizaciones utilizan tácticas de comunicación engañosas o manipuladoras (las tan conocidas *fake news*), pueden influir en la percepción pública de manera negativa, lo que puede tener un impacto perjudicial en la sociedad. En este sentido, basta recordar el conocido escándalo de *Cambridge Analytica*.

Implicaciones económicas

Costes de la plataforma

La implementación de una plataforma OSINT para recopilar información requiere recursos financieros y técnicos significativos, lo que puede generar costes adicionales para quienes la desarrollan y utilizan. Estos costes pueden incluir gastos de hardware y software, así como costes de personal y capacitación.

Beneficios económicos de la información

La plataforma desarrollada tendría potencial de adaptación a otros dominios distintos a los de la guerra en Ucrania. La información recopilada a través de una plataforma OSINT puede tener valor económico para las empresas y organizaciones. Por ejemplo, esta información puede utilizarse para realizar análisis de mercado, monitorear tendencias y tomar decisiones de inversión.

Impacto en la industria de la tecnología

El uso de plataformas OSINT para recopilar información se está volviendo cada vez más popular, lo que podría impulsar la demanda de productos y servicios relacionados con la tecnología de recopilación y análisis de datos. Esto podría tener un impacto positivo en la industria de la tecnología y en la economía en general.

Implicaciones ambientales

Consumo de energía

La implementación de una plataforma OSINT para recopilar información requiere el uso de hardware y software, lo que puede generar un mayor consumo de energía y, por lo tanto, una mayor emisión de gases de efecto invernadero. Esto puede tener un impacto negativo en el medio ambiente y contribuir al cambio climático.

Generación de residuos electrónicos

El uso de una plataforma OSINT implica la adquisición y actualización constante de hardware y software, lo que puede generar una gran cantidad de residuos electrónicos. Estos residuos pueden ser tóxicos y difíciles de reciclar, lo que puede tener un impacto negativo en el medio ambiente.

ANEXO II: REFLEXIONES ÉTICAS Y SOCIALES

En primer lugar, es importante destacar que cualquier uso de una plataforma OSINT para recopilar información debe ser legal y ético. En el contexto de la guerra de Ucrania, esto significa que la recopilación de información debe realizarse de acuerdo con las leyes nacionales e internacionales y no debe violar los derechos humanos de ninguna persona.

Además, el uso de una plataforma OSINT para recopilar información sobre la guerra de Ucrania debe ser transparente y respetar la privacidad de las personas. Es importante tener en cuenta que los datos recopilados pueden contener información personal o privada que no se debe divulgar sin el consentimiento de las personas involucradas. La plataforma desarrollada tendría potencial de adaptación a otros dominios distintos a los de la guerra en Ucrania.

Por otro lado, la recopilación de información a través de una plataforma OSINT puede ser de gran valor para entender mejor la situación en Ucrania y sus implicaciones sociales y políticas. La información recopilada puede ayudar a identificar patrones y tendencias en la guerra, así como a informar la toma de decisiones políticas y humanitarias.