CONTROL DE FUGA DE DATOS EN LA PLATAFORMA DE CONTENIDO TWITTER

Autor: Alfonso García Alonso

Tutor: Jordi Guijarro Olivares

Profesora: Helena Rifà Pous

Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones

Área de análisis de datos

15 de diciembre de 2019

Créditos/Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	Control de fuga de datos en la plataforma de contenido Twitter
Nombre del autor:	Alfonso García Alonso
Tutor:	Jordi Guijarro Olivares
Profesora:	Helena Rifà Pous
Fecha de entrega:	12/2019
Titulación o programa:	Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones
Área del Trabajo Final:	Análisis de datos
Idioma del trabajo:	Español
Palabras clave	*Detección-de-fugas, Twitter, Minería-de-datos, Análisis-de-sentimiento* *Leak-detection, Twitter, Data-Mining, Sentiment-Analysis*

Resumen del Trabajo / Abstract:

El presente trabajo aborda la creación de un módulo para la plataforma de control de filtraciones AIL Framework, que permita monitorizar las publicaciones realizadas durante un periodo concreto de tiempo en la red social Twitter, así como las que se produzcan en tiempo real, realizando además un análisis de sentimiento y un análisis estadístico de las publicaciones recopiladas. El módulo desarrollado consigue los objetivos de monitorización y análisis propuestos, integrándose completamente en la plataforma AIL, pudiendo ser de utilidad para equipos de respuesta ante incidentes de seguridad y, en general, para cualquier persona o entidad interesada en monitorizar posibles filtraciones o en realizar análisis de las publicaciones realizadas en la red social.

This paper addresses the creation of a module for AIL framework, a platform that analyses potential information leaks, that allows to monitor the content published in the social network Twitter either within a certain period of time or in real time, in addition to performing a sentiment and statistical analysis of the publications collected. The module developed achieves the objectives, integrating completely into the AIL platform and can be useful for security incident response teams and, in general, for any person or entity interested in monitoring possible leaks on Twitter or in performing analysis of the publications made in the social network.

Abstract

El presente trabajo aborda la creación de un módulo para la plataforma de control de filtraciones AIL Framework, que permita monitorizar las publicaciones realizadas durante un periodo concreto de tiempo en la red social Twitter, así como las que se produzcan en tiempo real, realizando además un análisis de sentimiento y un análisis estadístico de las publicaciones recopiladas. El módulo desarrollado consigue los objetivos de monitorización y análisis propuestos, integrándose completamente en la plataforma AIL, pudiendo ser de utilidad para equipos de respuesta ante incidentes de seguridad y, en general, para cualquier persona o entidad interesada en monitorizar posibles filtraciones o en realizar análisis de las publicaciones realizadas en la red social.

This paper addresses the creation of a module for AIL framework, a platform that analyses potential information leaks, that allows to monitor the content published in the social network Twitter either within a certain period of time or in real time, in addition to performing a sentiment and statistical analysis of the publications collected. The module developed achieves the objectives, integrating completely into the AIL platform and can be useful for security incident response teams and, in general, for any person or entity interested in monitoring possible leaks on Twitter or in performing analysis of the publications made in the social network.

Palabras clave

Detección-de-fugas, Twitter, Minería-de-datos, Análisis-de-sentimiento
Leak-detection, Twitter, Data-Mining, Sentiment-Analysis

Notaciones y Convenciones

A continuación, se detallan las notaciones y convenciones utilizadas en el presente TFM.

Comandos ejecutables en línea de comandos

Para resaltar los comandos de Shell, estos serán presentados de la siguiente manera:

```
#  ejemplo de comando
```

Nombres de archivos

Los nombres de archivos y carpetas se presentarán de la siguiente manera:

`Nombre_de_archivo`

Configuración

Los parámetros de configuración se presentarán de la siguiente manera:

```
pystemonpath = /home/pystemon/pystemon/
```

Acciones de menú:

Cuando se quiera mostrar una serie de pasos a seguir en la interacción con una interfaz de usuario, se empleará la siguiente nomenclatura:

Aplicaciones → **Accesorios** →

Índice

Figuras y tablas

Índice de figuras

Índice de tablas

1. Introducción

1.1. Análisis de situación

La privacidad de las personas y la protección de la información de las entidades (corporaciones, compañías, instituciones, etc.) se han convertido en aspectos capitales en la conocida como **sociedad de la información** en la que se da un <<uso intensivo de las tecnologías de la información y la comunicación (TIC), que facilitan la creación, distribución y manipulación de la información y desempeñan un papel esencial en las actividades sociales, culturales y económicas>>[1].

Desde el punto de vista de la información como **activo de gran valor para las entidades**, las filtraciones sobre el know-how, la información económica, la propiedad intelectual, la cartera de clientes, las estrategias o la propagación de bulos con fines maliciosos, puede hacer tambalear los cimientos más estables, con consecuencias económicas, legales o reputacionales que pueden llegar a ser catastróficas e irreversibles.

Por otro lado, la aparición de las redes sociales, el Big data[2] y los avances en los dispositivos tecnológicos de consumo, ha incrementado el riesgo de violaciones de la **privacidad y la intimidad personal**, ya sea por la utilización, por parte de empresas, de sistemas poco éticos de recolección y tratamiento de la información de los usuarios, ya sea por el uso delictivo o con voluntad maliciosa de redes sociales y tecnología para hacer daño a otras personas.

Si además tenemos en cuenta el carácter público y permanente de la información publicada en Internet, nos encontramos ante una situación en la que, una vez realizada la filtración de datos, es muy difícil contener su difusión y minimizar las consecuencias, siendo fundamental, por lo tanto, la detección y reacción temprana ante eventuales fugas de datos.

En las siguientes secciones de este capítulo, analizamos la plataforma de contenido Twitter[3], estudiando sus características y el uso que se le da por parte de los usuarios, con el fin de comenzar a esbozar, en el siguiente capítulo, un módulo que permita la monitorización en tiempo real, de la red de contenido, para la inyección de la información en la plataforma AIL, de tal manera que pueda seguir el flujo de tratamiento y análisis de fuga de datos establecido en dicha plataforma.

1 https://es.wikipedia.org/wiki/Sociedad_de_la_informaci%C3%B3n

2 Mediante el análisis de los usuarios, a través del procesamiento de grandes conjuntos de datos.

3 https://es.wikipedia.org/wiki/Twitter

1.1.1. Twitter

Twitter es una red social con más de 330 millones de usuarios activos[4] y un volumen promedio diario de 500 millones de mensajes[5]. La plataforma ha conseguido convertirse en una de las principales redes de información para el acceso y publicación de noticias, microblogging, interacción entre consumidores y empresas, búsqueda de información, interacción con figuras influyentes y para fines políticos y propagandísticos.

Para poder entender el tipo de filtraciones que pueden darse en esta plataforma, es necesario definir las **características básicas de la red social Twitter**:

- *Plataforma de comunicación abierta:* No es necesario el consentimiento del publicador de la información para acceder a esta, cualquier usuario puede ver la información que se publica.[6]
- *Sistema de etiquetado:* Twitter implementa una comunicación fuertemente basada en etiquetas, ya sea para citar a otros usuarios (mediante el empleo de las '@') o para referirse a temas específicos (mediante el empleo de las '#').
- *Limitación de caracteres:* Los mensajes (conocidos como Tweets) están limitados a 280 caracteres[7].
- *Emoticonos y emojis:* Twitter permite el uso de emoticonos[8] y emojis[9], siendo estos muy utilizados por los usuarios en las publicaciones.
- *Comunicación anónima:* En principio, la red permite la comunicación desde el anonimato, mediante el uso de un seudónimo escogido como perfil público.
- *Comunicación descentralizada:* La información, desde el punto de vista de la comunicación, viaja de manera descentralizada, multipunto.
- *Servicio gratuito:* El acceso y uso de la herramienta es gratuito, basándose el modelo de monetización de la empresa en la inclusión de publicidad en el servicio.

4 Fuente: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

5 Fuente: https://business.twitter.com/es.html

6 Con excepción de los mensajes directos entre usuarios, que son privados, y de los usuarios bloqueados, que no pueden ver ni seguir las publicaciones de quienes les hayan bloqueado.

7 Inicialmente la limitación era de 140 caracteres.

8 Secuencia de caracteres ASCII que representan una emoción, pej. :-).

9 Ideogramas, de origen japones, que han sido incorporados al estándar Unicode, y que representan emociones, objetos, animales, etc.

- *Servicio global:* El acceso a la red es global[10] y está disponible en más de 25 idiomas.
- *Simple y multiplataforma:* La red es de fácil uso, orientada o usuarios no expertos y permite el acceso desde una variedad de dispositivos.
- *Viral:* Su carácter abierto, global y social fomenta la rápida circulación y multiplicación de la información.
- *Comunicación sincrónica:* La fugacidad del *timeline*[11] hace coincidir los tiempos de publicación y lectura de la información[12].

Además, es importante entender los distintos **usos que los usuarios le dan a la red social**. Destacaremos[13] las siguientes áreas en las comunicaciones:

- Microblogging.
- Servidor de noticias.
- Marketing.
- Conversaciones entre usuarios.
- Cohesión social.
- Autopromoción.
- Política.

1.1.1.1. Filtraciones

Por último, analizaremos los **tipos de filtraciones, según sus causas y objetivos,** que se pueden dar en una red social como Twitter:

Publicación involuntaria de datos.

Tanto en el ámbito personal como en el empresarial se dan filtraciones de datos sensibles sin una intención deliberada. Las causas son muy variadas y van desde la publicación de

10 Twitter ha sido objeto de censura en Irán, Turquía, China, Egipto y Corea del Sur.

11 Cronología de tweets que se presentan al usuario.

12 Recientemente, Twitter ha implementado cambios para mostrar los Tweets en orden no cronológico, resaltando publicaciones realizadas en el pasado.

13 Fuentes: https://en.wikipedia.org/wiki/Twitter_usage, https://pearanalytics.com/

información sensible por el desconocimiento del carácter confidencial de dicha información[14], a fallos en el sistema, deficiencias en los procedimientos para el manejo de la información (o ausencia de estos), estados de alteración de la conducta del usuario o mal uso de los dispositivos electrónicos [16].

Filtraciones con dolo.

Las filtraciones también pueden realizarse con la voluntad deliberada de cometer un delito. Nos encontramos en una situación en la que se quiere dañar a una persona o entidad, ya sea atacando su imagen o reputación con la publicación de información privada o sensible, o mediante la propagación de bulos. Las causas pueden ser variadas: intereses económicos, motivos políticos, motivos personales, etc.

Filtraciones denuncia.

Se trata de filtraciones informativas que publican información secreta o privilegiada con el fin de que sea difundida y así denunciar una situación política, social, económica o de otra índole.

También pueden existir otras causas más particulares y residuales que lleven a la filtración de información, y que no estén recogidas en las categorías descritas, como son la autopromoción, la búsqueda de estatus o el mero desafío personal.

1.1.1.2. Análisis de sentimiento en redes sociales

El análisis de sentimiento, o minería de opinión, es el análisis de textos para, mediante el procesamiento del lenguaje natural, determinar la actitud, opinión y, en general, el sentimiento del interlocutor con respecto a un tema.

En el ámbito de las redes sociales, el análisis de sentimiento se convierte en una herramienta muy útil ya que permite determinar el tono emocional que hay detrás de una o varias palabras, ya sean estas una marca, tema de actualidad, compañía, etc. clasificándolas en base a su connotación positiva, negativa o neutra.

Hay que tener en cuenta que, más allá de la fiabilidad y precisión de las técnicas existentes en el análisis de sentimiento, la comunicación humana no siempre se corresponderá a una de

14 En ocasiones, además, la información puede ser aparentemente pública y no sensible, pero permitir, a través de ella o de su procesamiento junto con otros datos, inferir información privada o confidencial.

las tres categorías mencionadas y, por lo tanto, nos encontramos ante un sistema inmaduro y que necesita evolucionar.

Durante el desarrollo del presente TFM se estudiarán las técnicas de análisis de sentimiento existentes y se analizarán las herramientas disponibles hoy en día con el fin de integrar la que mejor se adapte a las necesidades del proyecto en la solución que propone este trabajo.

1.2. Descripción del TFM

Tras analizar la importancia del control de las fugas de datos relativos a las personas físicas y a entidades, en el contexto de una sociedad de la información (ver sección 1.1) y de estudiar las características de la red social Twitter (ver sección 1.1.1) para determinar los tipos de filtraciones que se pueden realizar en dicha plataforma (ver sección 1.1.1.1), podemos comenzar a plantear una solución que ayude a detectar y actuar ante estas filtraciones. Además, estudiaremos el estado del arte de las herramientas de análisis de sentimiento para determinar cuál puede ser más adecuada en Twitter, en base a las características de dicha red social.

Este TFM pretende abordar la detección de las fugas de datos que puedan darse en Twitter en tiempo real, o en un periodo de tiempo específico, desde un enfoque de monitorización y alerta, en base a palabras clave. Además, se estudiará el uso de técnicas de análisis de sentimiento con el fin de valorar la opinión de los usuarios en las publicaciones.

Para el procesamiento y análisis de los datos recopilados se ha escogido la plataforma de análisis de datos no estructurados, AIL Framework, que se estudia en profundidad en el *Anexo: AIL Framework*. Dicha plataforma carece de funcionalidad para la monitorización de la red social Twitter, y ese será el objetivo principal de este TFM.

1.3. Objetivos generales

Por lo tanto, este trabajo estará enfocado principalmente en el desarrollo de un módulo que se integre en la plataforma AIL y que permita recopilar la información publicada en Twitter, ya sea en tiempo real o en un periodo escogido por el usuario de la herramienta, para su posterior análisis en búsqueda de posibles filtraciones de datos y para el análisis de sentimiento de las publicaciones recopiladas.

1.3.1. Objetivos principales

El desarrollo de dicho módulo tiene los siguientes objetivos:

- Acceso en tiempo real a las publicaciones realizadas en Twitter.
- Acceso al histórico de publicaciones contenidas en Twitter.
- Integración con la plataforma AIL de manera que puedan utilizarse sus funciones de análisis y tratamiento de la información.
- Desarrollo de una interfaz para la visualización de datos agregados y estadísticos respecto a la información recopilada.
- Desarrollo de un módulo de análisis de sentimiento que determine la connotación positiva o negativa de las publicaciones de Twitter recopiladas.

1.4. Metodología y proceso de trabajo

Para la consecución de los objetivos, se ha planteado la siguiente línea de trabajo:

1. Estudio de la plataforma AIL.

   Se realizará el despliegue de la plataforma AIL para su estudio y posterior base de pruebas y desarrollo.

2. Investigación sobre la forma de monitorizar Twitter en tiempo real.

   Se estudiarán los posibles métodos para el acceso, en tiempo real, a las publicaciones realizadas en Twitter, analizando las ventajas e inconvenientes de cada uno de ellos, y escogiendo el que mejor se adapte a las siguientes características:

   - El acceso a la información debe de ser total, se descartarán métodos que proporcionen información parcial o filtrada.
   - El método debe de ser gratuito y sin limitaciones de uso.
   - El método debe de ser relativamente consistente en el tiempo, sin una gran dependencia de los cambios realizados en la plataforma social o en la plataforma de control de fuga de datos.

3. Investigación de las técnicas de análisis de sentimiento.

   Se estudiarán las técnicas existentes para el análisis de sentimiento de textos y se escogerá la que mejor se adecue a las características de las publicaciones en la red social. De nuevo, se descartarán métodos que no sean gratuitos o que impliquen limitaciones de uso.

4. Desarrollo del módulo de monitorización.

   Se desarrollará un módulo para el acceso en tiempo real a Twitter y la inyección de la información en la plataforma AIL, de tal manera que pueda seguir el flujo de tratamiento y análisis ya establecidos.

5. Desarrollo del módulo de análisis de sentimiento.

   Se desarrollará un módulo de análisis de sentimiento especializado en las publicaciones de Twitter y que se nutrirá de las publicaciones recopiladas por el módulo de monitorización.

6. Desarrollo del módulo de presentación.

   Se desarrollará un módulo de presentación que muestre las publicaciones recopiladas, así como el análisis de sentimiento realizado sobre dichos textos e información estadística y agregada de la información compilada.

1.5. Planificación

Una vez planteada la línea de trabajo, desglosaremos las fases que hemos establecido en subtareas, con el fin de organizar el desarrollo y asegurar así la consecución de los objetivos en los plazos existentes[15].

1. Estudio de la plataforma AIL.

   1.1. Despliegue y configuración de la plataforma AIL.

   1.2. Estudio en profundidad de la plataforma (arquitectura, interfaces, funcionalidad).

2. Investigación sobre la monitorización en tiempo real de Twitter y las técnicas de análisis de sentimiento.

   2.1. Investigación sobre los posibles métodos, escogiendo el que mejor se adapte a los requisitos marcados y a las características de las publicaciones que se realizan en Twitter.

3. Desarrollo de los módulos de monitorización, procesamiento y presentación.

   3.1. Desarrollo del código fuente, en Python.

   3.2. Integración y pruebas en la plataforma.

   3.3. Análisis de los resultados obtenidos.

   3.4. Documentación.

4. Producto final.

   Fase final de aceptación del producto, documentando el sistema y las conclusiones del trabajo. Se realizará, además, una memoria que será presentada y defendida ante el tribunal.

15 Los plazos están establecidos por las entregas prefijadas por la universidad para el TFM.

A continuación, se muestra un diagrama que muestra gráficamente las distintas fases del proyecto, incluyendo las subtareas, y el flujo de trabajo. Como se puede observar, la fase de desarrollo sigue un modelo de mejora continua y los resultados obtenidos en varias de las fases pueden implicar la revisión de decisiones tomadas en fases anteriores. Además, se reflejan las entregas que se realizarán al tutor del TFM.

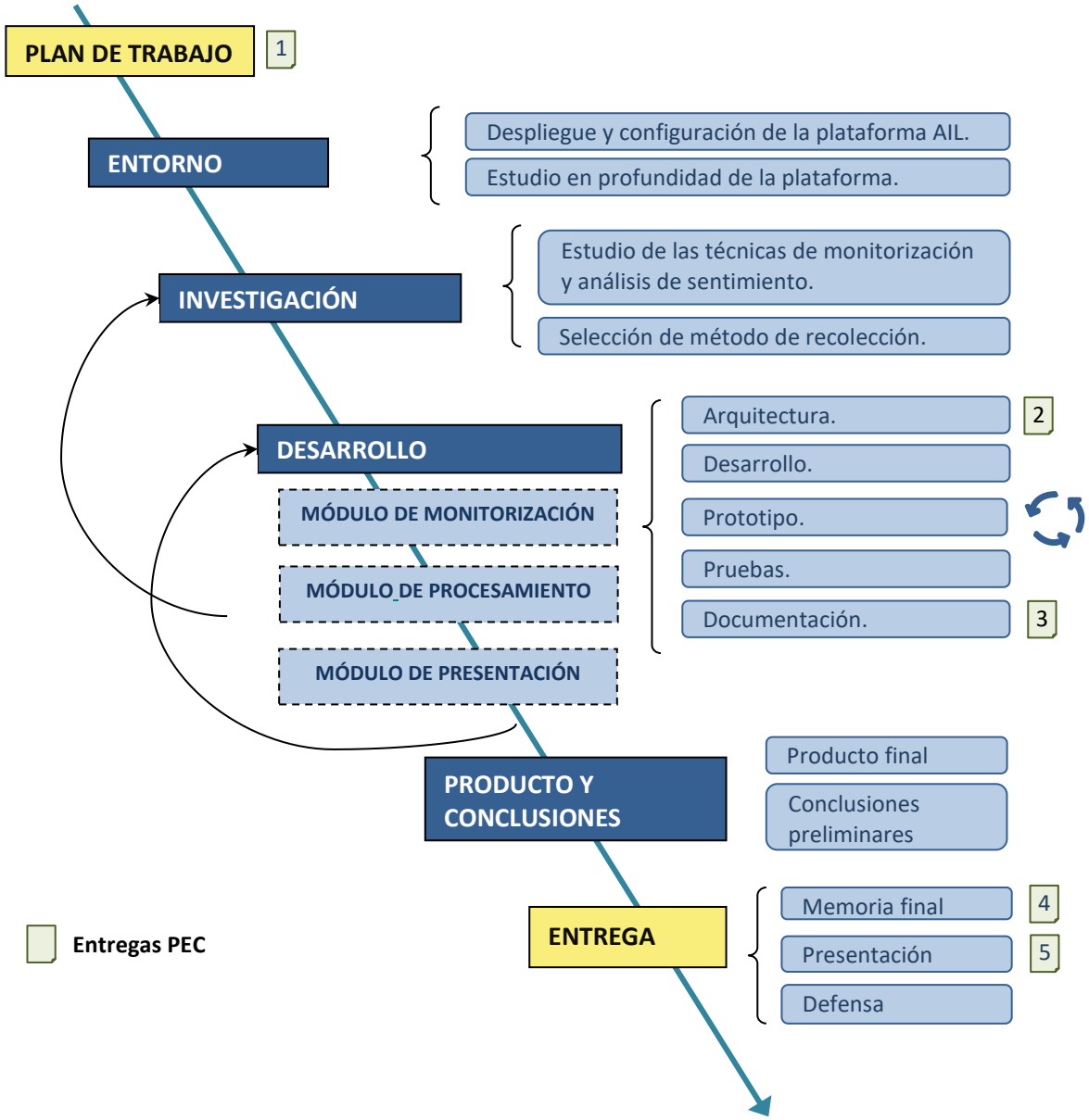


Figura 1: Plan de trabajo

Hitos y Fechas clave:

En base a la planificación realizada, se han establecido las siguientes entregas:

- *Entrega 1 – Plan de trabajo*: Planificación del TFM incluyendo análisis de situación, objetivos, metodología, hitos y fechas clave.
- *Entrega 2 – Arquitectura*: Entrega de la arquitectura definida para la solución.
- *Entrega 3 – Prototipo*: Entrega del prototipo desarrollado, junto con la documentación de la investigación e implementación.
- *Entrega 4 – Memoria final*: Síntesis del trabajo realizado, incluyendo el producto final y las conclusiones.
- *Entrega 5 – Presentación*: Video que sintetice el trabajo realizado sobre una presentación de diapositivas.
- *Defensa*: Respuesta a las preguntas planteadas por el tribunal en el foro virtual.

Se ha establecido una planificación teniendo en cuenta una disposición semanal de 2 horas de lunes a jueves y 4 horas de sábado a domingo. Ante cualquier eventualidad inesperada o desviación de la planificación se realizará un esfuerzo de dedicación para alcanzar los objetivos establecidos.

La siguiente tabla resume los distintos entregables que han sido planificados, indicando la fecha de entrega:

Entregable	Fecha	Descripción de la entrega
Entrega 1	01/10/2019	Plan de trabajo.
Entrega 2	29/10/2019	Arquitectura.
Entrega 3	26/11/2019	Prototipo.
Entrega 4	31/12/2019	Memoria final.
Entrega 5	07/01/2020	Presentación en video.
Defensa	17/01/2020	Defensa del TFM.

Tabla 1: Entregables

1.6. Público objetivo

El producto resultante de este TFM puede ser de utilidad para equipos de respuesta ante incidentes de seguridad (CSIRT o CERT), empresas u organizaciones y, en general, para cualquier persona o entidad interesada en monitorizar posibles filtraciones en Twitter o en realizar análisis de las publicaciones realizadas en la red social.

Los centros de respuesta ante incidencias de seguridad surgen de la necesidad de dar una respuesta organizada a las múltiples y diversas amenazas en el ámbito de la seguridad de las tecnologías de la información.

1.7. Antecedentes

Este TFM continua el marco de trabajo de CIRCL, el equipo de respuesta ante incidentes informáticos de Luxemburgo, que comenzó en 2014 el desarrollo de la plataforma AIL como un proyecto interno y que ha puesto el código fuente a disposición del público con licencia GNU Affero General Public License v3.0.

La idea de este TFM surge de la necesidad de añadir la monitorización de la red social Twitter en la plataforma AIL utilizada por el equipo de respuesta a incidentes de la anella científica (CSUC-CSIRT)[16].

16 https://www.csuc.cat/es/comunicaciones/seguridad

2. Análisis del estado del arte

2.1. Monitorización en tiempo real

En este capítulo abordamos la monitorización en tiempo real de las publicaciones de Twitter, estudiando las alternativas existentes para la conexión con Twitter y los datos a obtener.

2.1.1. Método de obtención de la información

La siguiente tabla muestra las distintas opciones que se han encontrado para la monitorización de Twitter, resumiendo las ventajas e inconvenientes de cada una de ellas.

Opción	Detalle	Ventajas y Desventajas
1. Twitter APIs	APIs oficiales de Twitter.	▪ Solución oficial de Twitter. ▪ Posibilidad de acceder a todo el archivo de publicaciones. --- ▪ Solución de pago o con resultados limitados, incompletos o con una tasa de consulta limitada. ▪ Dependencia de API, para la que se requiere autenticación.
2. Query Twitter search	Aprovechar el sistema de búsqueda oficial de Twitter.	▪ Independencia de API oficial. ▪ Gratuita. --- ▪ Posibles limitaciones. ▪ Sujeto a cambios.
3. Query Twitter timeline	Simulación de las solicitudes realizadas por el navegador web.	▪ Independencia de API oficial. ▪ Gratuita. --- ▪ Sujeto a cambios.
4. Proveedor externo.	Obtención de los datos a través de un proveedor externo.	▪ Solución de pago o con limitaciones en el acceso a los datos (ofrecen solo la interfaz).

Tabla 2: Opciones para la monitorización de Twitter

Opción 1 – Twitter APIs

Twitter ofrece tres APIs para el acceso de desarrolladores a la plataforma:

- *Standard Search API*:
    - Proporciona una ventana retrospectiva de 7 días.
    - Gratuita.
    - No se proporcionan todos los resultados (incomplete data fidelity).
    - Formato de respuesta: JSON
    - Requiere autenticación.
    - Tasa limitada.
    - Solicitudes / ventana de 15 minutos (user auth): 180
    - Solicitudes / ventana de 15 minutos (app auth): 450
- *Premiun Search API*: Existen dos opciones para esta API, ambas de pago:
    - API de 30 días: proporciona todos (data Fidelity) los Tweets publicados con los últimos 30 días.
    - API del archivo completo: proporciona acceso a todos (data Fidelity) los Tweets desde 2006.
- *Enterprise Search API*: Proporciona, al igual que sucede con la Premium Search API, las dos opciones según la ventana de tiempo que se desee, pero ofrece más opciones y soporte.
    - *PowerTrack API:* Esta API proporciona acceso en tiempo real a todas las publicaciones en Twitter.

En principio, el hecho de que las *APIs Premium* y *Enterprise* sean de pago nos reduce esta primera solución a únicamente la opción de la *Standard Search API*. Sin embargo, según se indica en la página oficial de Twitter[17], esta opción no ofrece una fidelidad absoluta en los datos devueltos con lo que tampoco parece la mejor opción para una monitorización de la totalidad de las publicaciones. Habrá que valorar también la posibilidad de combinar la *Standard Search API* con otra de las soluciones propuestas de tal manera que se tenga acceso a todo el archivo de publicaciones.

Hay que señalar que esta solución podría utilizarse, bien desarrollando desde cero un módulo que se conecte a Twitter a través de la API, o mediante alguna librería o plataforma preexistente que ya tenga implementado el módulo de autenticación. En este sentido, se han

[17] https://developer.twitter.com/en/docs/tweets/search/overview

encontrado varias soluciones que se podrían aprovechar parcialmente o utilizarse como referencia. Se listan a continuación algunos ejemplos:

- mmdemo-dockerized: Un conjunto de servicios para la monitorización de múltiples plataformas de redes sociales basadas en Docker. Ver https://github.com/MKLab-ITI/mmdemo-dockerized.
- twitterMonitor: Realiza una monitorización de los temas que son tendencia en Twitter. Ver https://github.com/dmarx/twitterMonitor
- TwitterStockMonitor: Monitorización de twitter enfocado al análisis de los movimientos bursátiles. Ver https://github.com/semaaJ/TwitterStockMonitor/.

Opción 2 – Query Twitter Search

Twitter ofrece un sistema de búsqueda de Tweets en https://twitter.com/search-home. Se podría explorar la opción de utilizar estas consultas para la recolección de los datos, aunque es bastante probable que existan limitaciones en las consultas que se realicen.

Opción 3 – Query Twitter timeline

Esta opción simula la búsqueda que se realiza en Twitter desde un navegador cuando se carga la página, solicitando bloques de tweets a un proveedor JSON. Está opción permitiría, a priori, bucear en todo el histórico de tweets. Si bien no se han detectado limitaciones en pruebas preliminares, es una posibilidad a tener en cuenta.

Las solicitudes JSON se realizan a la url **https://twitter.com/i/search/timeline?f=tweets&q=**

Opción 4 – Proveedor externo

Existen multitud de servicios profesionales para la monitorización y análisis de redes sociales que se han descartado por ser de pago o por no proporcionar una librería para el acceso a los datos en bruto, ofreciendo únicamente una interfaz web para la monitorización. A continuación, se listan algunos como ejemplo:

- Hootsuite: https://hootsuite.com/es/.
- Topsy: http://topsy.thisisthebrigade.com.
- Simplify360: https://simplify360.com.
- Sentione: https://sentione.com.
- MediaToolKit: https://www.mediatoolkit.com.

Como dato curioso, cabe señalar que existen varias cuentas de Twitter que publican sobre filtraciones detectadas en la propia red social. A continuación, se listan algunas de ellas:

- dumpmon
- BotChangePasswd
- DataLeakBot
- d2dedwad2

2.1.2. Características de la información a recolectar

Una vez hayamos escogido el método para obtener las publicaciones de Twitter, será necesario realizar un tratamiento de dicha información para su gestión y presentación. Para ello, presentamos en el Anexo: *Twitter - I.A* los parámetros principales de una publicación en Twitter, conocida como Tweet.

2.2. AIL framework

La solución escogida para la monitorización de Twitter será integrada en la plataforma AIL siguiendo el flujo de datos establecido en ella y adaptándose a la arquitectura existente. En el *Anexo: AIL Framework* se estudia en profundidad dicha plataforma.

2.3. Análisis de sentimiento

En la actualidad, las técnicas de análisis de sentimiento[18] se basan en alguno de los siguientes enfoques:

- Técnicas basadas en el conocimiento. Estas técnicas clasifican los textos basándose en la aparición de ciertas palabras prefijadas. Dentro de este enfoque, existen dos técnicas que pueden usarse por separado o en combinación:
    - Localización de palabras clave: se clasifica el texto basado en la aparición de palabras clave con una connotación inequívoca como triste, encantado, feliz, etc.
    - Afinidad léxica: se asigna a palabras arbitrarias una probabilidad de afinidad a una emoción.

18 https://en.wikipedia.org/wiki/Sentiment_analysis

- Métodos estadísticos: se hace uso de técnicas de aprendizaje de máquina (machine learning) como el análisis de semántica latente, máquinas de vectores de soporte, bolsas de palabras y orientación semántica - información mutua puntual. Algunas versiones más avanzadas de estos métodos tratan de detectar en los textos tanto al sujeto que expresa el estado afectivo como al objeto que causa la emoción.
- Métodos híbridos: hacen uso de las técnicas de aprendizaje de máquina, pero utilizando bases de conocimiento a través del análisis de textos mediante redes semánticas y el análisis ontológico, permitiendo detectar emociones no implícitas en la comunicación a través de las relaciones y enlaces entre conceptos.

El tipo de comunicación que se realiza en Twitter, compuesto por grandes volúmenes de contenido y que se caracteriza por textos muy breves, el uso de un lenguaje informal abreviado y la inclusión de emojis y emoticonos, presenta retos para el análisis de sentimiento de los textos y para las técnicas existentes. En una red social como Twitter es necesario tener en cuenta que el sentimiento de los textos puede cambiar al incluir este tipo de símbolos o cuando se usa una jerga que enfatiza las emociones mediante el uso de mayúsculas, signos de puntuación o un lenguaje informal.

Para la realización del análisis de sentimiento de los Tweets recopilados con la herramienta a desarrollar, se han estudiado varias librerías existentes[19] y se han analizado teniendo en cuenta las características de las publicaciones que se realizan en Twitter. Se listan a continuación:

- Stanford Sentiment Analysis Module [21].
- Textblob [19].
- LingPipe [20].
- CLIPS pattern [22].
- Vader [23].
- Librerías que permiten el desarrollo de módulos de análisis de sentimiento utilizando técnicas basadas en algoritmos de aprendizaje automático como Naive Bayes, Maximum Entropy y Support Vector Machine (SVM).

Podemos comprobar, por ejemplo, que la solución desarrollada por la Universidad de Stanford [21], basada en una red neuronal recursiva que se basa en estructuras gramaticales

[19] Se han obviado las librerías desarrolladas ad-hoc para plataformas fuera del ámbito de este TFM o que se basen en el uso de la API de Twitter, también descartada, como la plataforma GATE o Tweepy.

entrenada con un conjunto de datos[20], y de la cual ofrecen una demo web[21], no contempla el uso de emojis o emoticonos en los textos. La herramienta tampoco distingue el uso de palabras en mayúsculas o los signos de exclamación, factores que pueden incrementar la intensidad de la emoción expresada en una frase, como, por ejemplo: << La película es muy buena>> << La película es MUY BUENA>> << La película es MUY BUENA !!>>

La librería de TextBlob [19] está más enfocada al procesamiento de lenguaje natural, y aunque es destacable que realiza traducción de los textos a varios idiomas, tampoco contempla el uso de emojis o emoticonos. Se ha considerado que su uso puede tener un impacto negativo en la velocidad de procesamiento de las publicaciones ya que convierte las cadenas de texto en objetos Textblob para la realización de varios análisis de texto innecesarios para la solución que se plantea en este TFM.

LingPipe es un kit de herramientas para procesar texto usando lingüística computacional que permite realizar tareas como encontrar nombres de personas, organizaciones o ubicaciones en las noticias, clasificar automáticamente los resultados de búsquedas en Twitter o sugerir cambios ortográficos en consultas. Las desventajas que se han considerado de esta herramienta son que se trata de una librería con licencia de código pseudo-abierto[22], con limitaciones de uso, el hecho de estar desarrollada en Java (cuando la solución propuesta en este TFM estará basada en Python) y que no contempla el uso de emojis o emoticonos.

CLIPS proporciona módulos para el etiquetado de textos que permite identificar sustantivos, adjetivos, verbos, etc. en una oración, además de proporcionar análisis de sentimientos de dichos textos. Sin embargo, tampoco contempla el uso de emojis o emoticones y obliga a importar una librería distinta para cada idioma a analizar.

VADER es una herramienta de análisis de sentimientos, desarrollada en Python, basada en reglas y léxicos, que está especializada en el tipo de comunicación que se produce en redes sociales. Ha sido incorporada a NTLK [24] de manera que se puedan tokenizar[23] las publicaciones para su posterior análisis. Contempla el uso de emojis y emoticonos. Además, tiene en cuenta el uso de mayúsculas o símbolos a la hora de clasificar las emociones expresadas en los textos.

[20] https://nlp.stanford.edu/sentiment/treebank.html
[21] http://nlp.stanford.edu:8080/sentiment/rntnDemo.html
[22] http://alias-i.com/lingpipe/web/download.html
[23] Proceso de convertir una secuencia de caracteres en una secuencia de tokens (cadenas con un significado asignado e identificado).

Las librerías que proporcionan las bases para el desarrollo de módulos de análisis de sentimiento basados en algoritmos de aprendizaje automático han sido descartadas por el coste computacional, la necesidad de un conjunto de entrenamiento, no disponible, específico para una comunicación tan particular como la que se da en Twitter, y porque no se han encontrado evidencias de una fiabilidad suficiente que compense su complejidad en comparación con otras técnicas más simples basadas en modelos de reglas y léxicos.

3. Arquitectura

3.1. Arquitectura general

El siguiente diagrama muestra la arquitectura de la solución propuesta en este TFM, en su integración con la plataforma AIL:

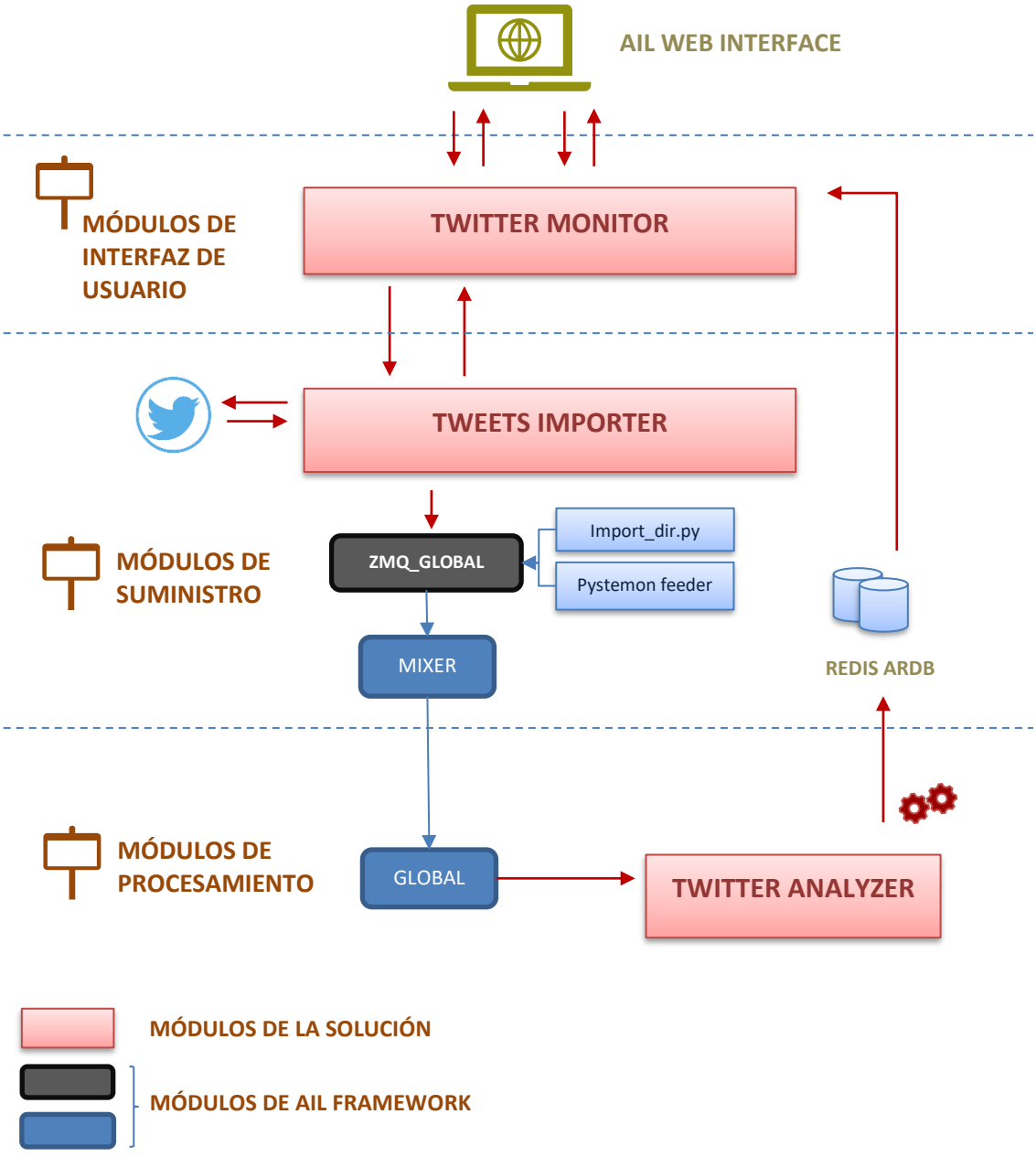


Figura 2: Arquitectura de la solución.

3.2. Módulos

3.2.1. Módulos de interfaz de usuario

La solución desarrollada contará con una interfaz de usuario para poder interactuar con ella y que dispondrá de tres áreas funcionales:

- Interfaz para configurar, activar y presentar el estado de la monitorización de Twitter.
- Interfaz para la presentación de datos relativos a la búsqueda realizada y los resultados del análisis de sentimiento de las publicaciones.
- Interfaz para la configuración del sistema.

3.2.1.1. Interfaz para la monitorización de Twitter

Mediante esta interfaz gráfica el usuario podrá activar y desactivar la monitorización de Twitter, así como establecer los parámetros de la monitorización. El sistema permitirá elegir entre monitorizar las publicaciones realizadas **en un periodo de tiempo** establecido por el usuario o activar la monitorización en **tiempo real** que realizará, hasta que el usuario lo cancele, una recopilación de las publicaciones que se realicen desde el momento de la activación.

El siguiente diagrama muestra la arquitectura de la interfaz de monitorización:

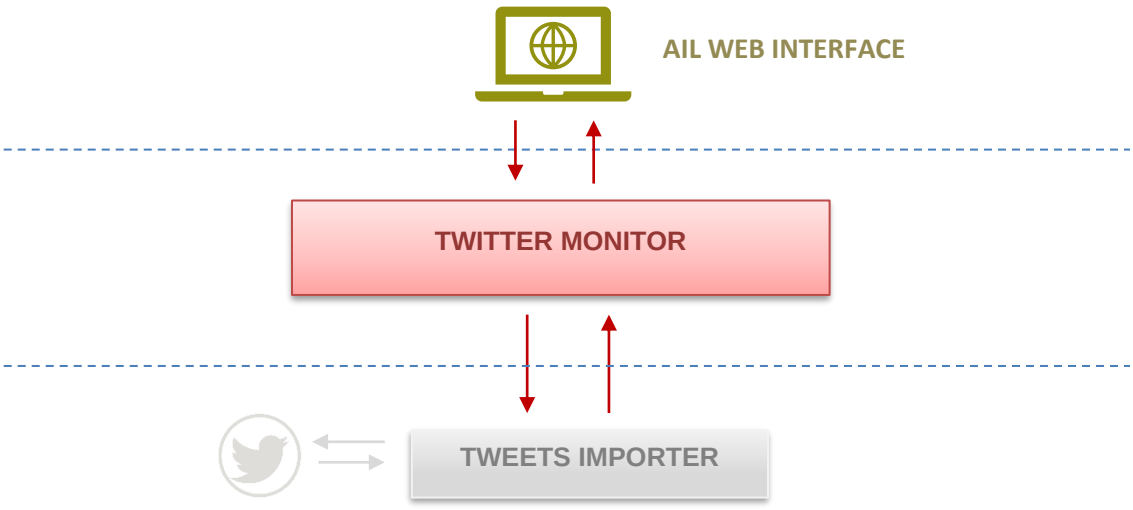


Figura 3: Arquitectura de la interfaz de usuario para la monitorización.

La interfaz está compuesta por archivos HTML, que harán uso de HTML, CSS y JavaScript y que se servirán del framework de Flask para interactuar con un archivo python, que será el encargado de recibir las ordenes desde la interfaz y devolver el estado de la operación para su presentación en la web. En el *ANEXO G* se pueden consultar los archivos específicos que conforman la solución propuesta.

3.2.1.2. Interfaz de resultados

Esta interfaz gráfica permitirá al usuario visualizar un resumen de las publicaciones recopiladas, además del resultado del análisis de sentimiento de cada uno de los bloques de datos o búsquedas realizadas por el usuario. El resultado del análisis consistirá en una medida en forma de valor compuesto, normalizado y ponderado, sobre el sentimiento positivo, neutral o negativo de las publicaciones de Twitter recopiladas.

El siguiente diagrama muestra la arquitectura de la interfaz de presentación de resultados:

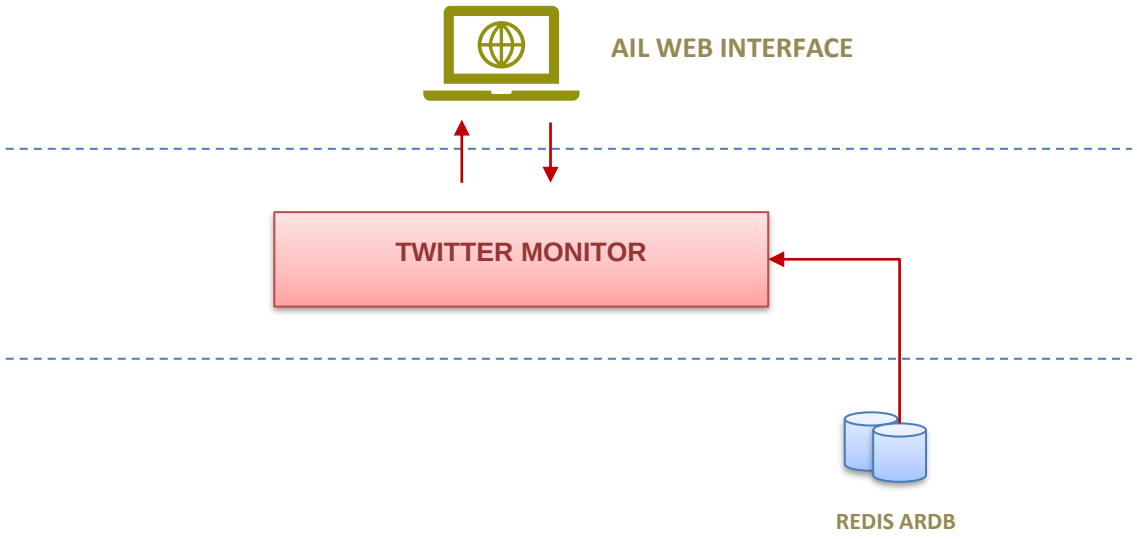


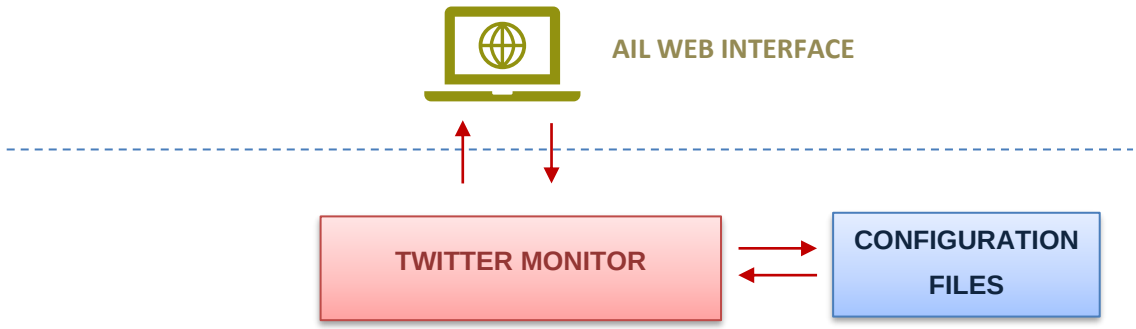
Figura 4: Arquitectura de la interfaz de usuario para la presentación de resultados

La interfaz está compuesta por archivos HTML, que harán uso de HTML, CSS y JavaScript y se servirán del framework de Flask para interactuar con un archivo python, que será el encargado de obtener la información almacenada en la base de datos Redis ARDB, que contiene el resultado de los análisis de sentimiento realizados y el resumen de las

publicaciones recolectadas. En el *ANEXO G* se pueden consultar los archivos específicos que conforman la solución propuesta.

3.2.1.3. Interfaz de configuración

Esta interfaz gráfica permitirá al usuario configurar el módulo según sus necesidades, almacenando en un fichero de configuración las preferencias del usuario. El siguiente diagrama muestra la arquitectura de la interfaz de configuración:



Creación de un módulo web en la plataforma AIL

A continuación, se explican los pasos para la creación de un módulo web en la plataforma AIL. El proceso consiste en ejecutar un script de python, proporcionado junto con la plataforma, que facilita su integración dentro de esta:

1. Ejecutamos el script `/var/www/créate_new_web_module.py` que nos solicita un nombre para el módulo:

```
#  /var/www/create_new_web_module.py
```

2. Introducimos el nombre para el módulo, en este caso se ha establecido TwitterMon.

Esto genera la siguiente estructura de archivos en AIL-framework/:

```
/var/www/modules/TwitterMon/
                    |__ Flask_TwitterMon.py
                    |__ Templates/
                      |__ Header_TwitterMon.html
                      |__ TwitterMon.html
```

3.2.2. Módulos de recopilación y suministro de la información

Estos módulos serán responsables de recopilar las publicaciones de Twitter, en base a los parámetros establecidos por el usuario en la interfaz web, y de suministrar la información a la plataforma AIL.

El siguiente diagrama muestra la arquitectura del módulo de recopilación y suministro de las publicaciones de Twitter:

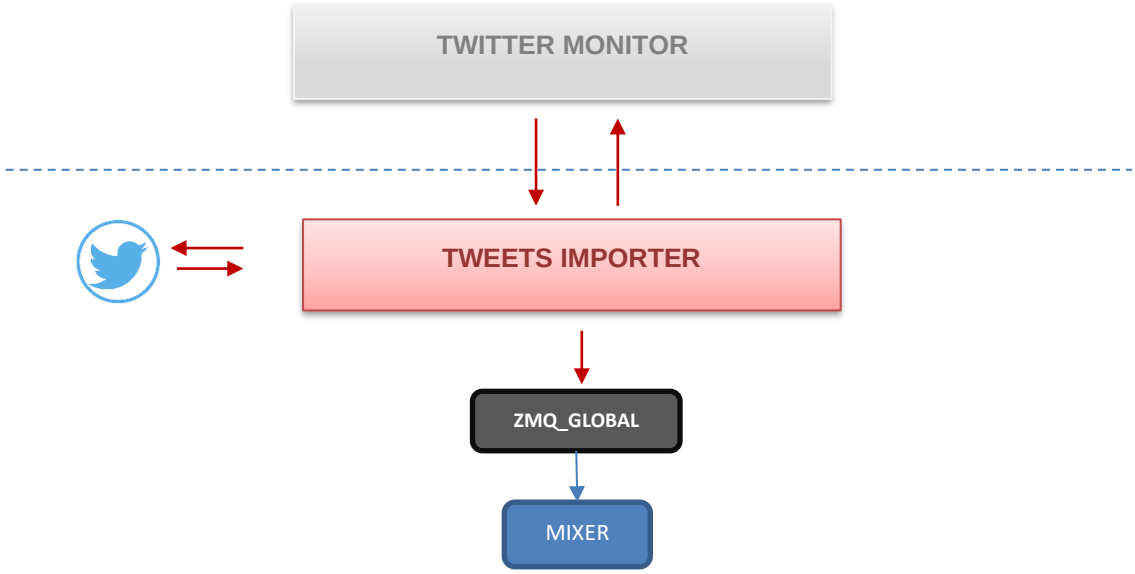


Figura 5: Arquitectura para la recopilación de publicaciones y suministro a la plataforma.

Recopilación de la información

Para la recolección de las publicaciones de Twitter, de entre los métodos de obtención que se plantearon en la sección 2.1, se ha escogido implementar una solución en Python basada en la simulación de las solicitudes realizadas por un navegador web cuando un usuario consulta las publicaciones de Twitter. Dicha solución se basa en una libraría desarrollada para la obtención de Tweets antiguos [18].

Suministro de la información

Según se vayan recibiendo y procesando los bloques de archivos JSON, la información será inyectada en la cola ZMQ que, a su vez, suministra a la cola Redis Global a la que están

subscritos los distintos módulos de procesamiento proporcionados por la plataforma AIL (Bitcoin, Credential, CVE, etc. Ver sección I.D). Esto permitirá que la información recopilada de Twitter sea clasificada y procesada por todos los módulos de la plataforma. En el *ANEXO G* se pueden consultar los archivos específicos que conforman la solución propuesta.

Colas de subscripción y publicación de los módulos

En el archivo `/AIL-framework/bin/packages/modules.cfg` se configura a que colas se subscriben y en que colas publican los distintos módulos de la plataforma. El archivo tiene la siguiente estructura:

```
[Mixer]
subscribe = ZMQ_Global
publish = Redis_Mixer,Redis_preProcess1

[Global]
subscribe = Redis_Mixer
publish = Redis_Global,Redis_ModuleStats
```

3.2.3. Módulos de procesamiento

El último paso en la solución propuesta en este TFM consistirá en un módulo de resumen estadístico de las publicaciones recopiladas y en un análisis de sentimiento, especializado en el tipo de contenido que se publica en la red social Twitter, que se nutrirán de los Tweets volcados en la cola Redis Global.

El siguiente diagrama muestra la arquitectura de los módulos de procesamiento:

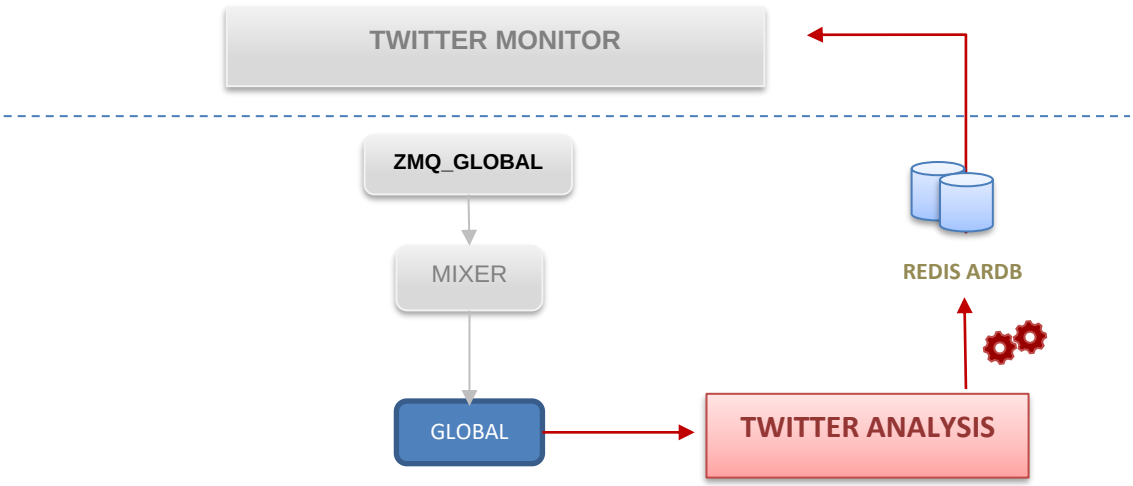


Figura 6: Arquitectura para el procesamiento de la información.

Análisis de sentimiento

Para el análisis de sentimiento de los tweets, de entre las soluciones estudiadas en la sección 2.3, se ha escogido la librería Vader Sentiment Analysis [23] por su simplicidad, basada en reglas y léxicos, por el hecho de estar desarrollada en Python y por estar adaptada a las conversaciones y publicaciones que se realizan en redes sociales como Twitter. En el ANEXO F se estudia la librería en más profundidad.

Si bien la plataforma AIL ya implementa un análisis de sentimiento utilizando la librería nltk.sentiment.vader, esta no soporta el análisis de emojis, mientras que la librería de Vader original [23] sí que lo hace. Además, la solución que se propone en este TFM pretende mejorar la presentación de los resultados obtenidos en el análisis y su adaptación al formato de publicaciones que se realizan en Twitter.

Análisis de las publicaciones

Además del análisis de sentimiento, se realizarán análisis estadísticos de las publicaciones recopiladas, incluyendo la generación de datos agregados de los bloques procesados.

Por último, los resultados de los análisis realizados se grabarán en la base de datos para su presentación en la interfaz gráfica de la solución. En el *ANEXO G* se pueden consultar los archivos específicos que conforman la solución propuesta.

Creación de un módulo de procesamiento en la plataforma AIL

A continuación, se explican los pasos para la creación de un módulo de procesamiento en la plataforma AIL. La plataforma proporciona una plantilla que facilita la creación e integración del módulo dentro de esta:

Plantilla: `AIL-framework/bin/template.py`

En este caso se ha establecido `TwitterAnalyzer.py` para el módulo de análisis.

Si se desea que el módulo se arranque junto con la plataforma, es necesario añadir la siguiente línea de código al script de lanzamiento `/bin/LAUNCH.sh` :

```
# screen -S "Script_AIL" -X screen -t "TwitterAnalysis" bash -c "cd
  ${AIL_BIN}; ${ENV_PY} ./TwitterAnalysis.py; read x"
```

4. Diseño

4.1. Estructura y archivos

A continuación, se listan los archivos y carpetas que componen la solución propuesta en este TFM dentro de la estructura de la plataforma AIL. Asimismo, se indica con la etiqueta [M] los ficheros pertenecientes a la plataforma AIL que son modificados para el correcto funcionamiento del sistema de monitorización.

La estructura de la solución, en su formato entregable previo a la instalación, puede encontrarse en el ANEXO G.

```
/bin/
   |__ LAUNCH.sh [M]
   |__ TwitterAnalyzer.py
   |__ packages/
         |__ modules.cfg [M]
         |__ Tweet.py

/var/www/
       /templates/
              |__ nav_bar.html [M]
              |__/twittermon/
                    |__ menu_sidebar.html
       /modules/
          |__ /TwitterMon/
                    |__ Flask_TwitterMon.py
                    |__ TweetsImporter.py
                    |__ TM_Status.py
                    |__ data/
                    |__ logs/
                    |__ config/
                          |__ TwitterMon.cfg
                    |__ templates/
                          |__ header_TwitterMon.html
                          |__ TwitterMon.html
                          |__ TwitterMon_results.html
                          |__ TwitterMon_settings.html
```

La siguiente tabla muestra una breve descripción funcional de los ficheros que componen la solución. En las siguientes secciones se describe en detalle su papel en las funciones del sistema.

Archivo	Descripción
`LAUNCH.sh`	Fichero de AIL para **arrancar la plataforma**. • **Tipo**: Linux shell script. Se modifica para añadir el lanzamiento del módulo de procesamiento de Tweets, `/bin/TwitterAnalyzer.py`.
`TwitterAnalyzer.py`	Módulo de **procesamiento y análisis** de Tweets. • **Tipo**: Python v3. Este módulo de procesamiento se encarga de procesar los Tweets recopilados, realizando un análisis de sentimiento y un análisis estadístico.
`modules.cfg`	Fichero de configuración de AIL Framework. • **Tipo**: Fichero de configuración. Se modifica para indicar a la plataforma la cola a la que se subscribe la solución para recibir los Tweets recopilados.
`Tweet.py`	Fichero auxiliar del módulo de **procesamiento y análisis** de Tweets. • **Tipo**: Python v3. Este fichero auxiliar Python es el encargado de crear objetos diseñados para contener los Tweets recopilados, ayudando así al análisis de la información recopilada.
`nav_bar.html`	Fichero web de AIL Framework. • **Tipo:** HTML. Este archivo HTML de AIL define la barra de navegación superior de la plataforma web. Se ha modificado para añadir un acceso directo al sistema de monitorización de Twitter, TwitterMon.
`menu_sidebar.html`	Fichero HTML auxiliar que define el menú de accesos directos que se presenta en la barra lateral de la página del módulo de Twitter. • **Tipo:** HTML.

Archivo	Descripción
`Flask_TwitterMon.py`	Fichero Flask Python que sirve de interfaz entre la interfaz web, los módulos de operación Python y la base de datos. • **Tipo**: Python v3. Este fichero permite la comunicación entre los módulos de interfaz de usuario y los módulos de recopilación, procesamiento y análisis.
`TweetsImporter.py`	Módulo de **recopilación de Tweets.** • **Tipo**: Python v3. Su función es realizar solicitudes a Twitter, en base a los parámetros de monitorización establecidos por el usuario, y suministrar las publicaciones recopiladas a la cola de procesamiento de la plataforma.
`TM_Status.py`	Fichero auxiliar del módulo de **recopilación de Tweets**, `TweetsImporter.py`. • **Tipo**: Python v3. Se encarga de operaciones auxiliares a la importación de Tweets, como almacenar el estado de la monitorización en curso o la conexión.
`TwitterMon.cfg`	**Fichero de configuración del sistema** de monitorización. • **Tipo**: Fichero de configuración. Este archivo de configuración guarda las preferencias establecidas por el usuario para la monitorización y análisis de Tweets.
`header_TwitterMon.html`	Fichero HTML auxiliar que define el acceso directo a la aplicación, en el submenú superior de la plataforma. • **Tipo**: HTML.

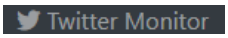
Archivo	Descripción
`TwitterMon.html`	Página web para **configurar, operar y mostrar el estado de la monitorización** de Twitter. • **Tipo**: HTML, JavaScript. Ofrece al usuario un panel para configurar la monitorización de la red social, así como la posibilidad de arrancar y parar la monitorización.
`TwitterMon_results.html`	Página web que ofrece al usuario los **resultados** de la recopilación de Tweets, así como datos del **análisis realizado**. • **Tipo**: HTML, JavaScript. Ofrece al usuario un listado de todas las búsquedas realizadas, así como el detalle del análisis de cada una de ellas.
`TwitterMon_settings.html`	Página web que permite al usuario **configurar** ciertos parámetros de la monitorización y análisis de Tweets. • **Tipo**: HTML, JavaScript. Este archivo es el responsable de la lectura y escriturar en el fichero de configuración `TwitterMon.cfg`.

Tabla 3: Descripción funcional de los archivos de la solución

4.2. Módulo de interfaz de usuario

El módulo de interfaz de usuario comprende la interfaz para configurar y operar la **monitorización,** la interfaz que muestra los **resultados de los análisis** realizados y la interfaz para la **configuración del módulo**. La interfaz de usuario está compuesta por los siguientes ficheros:

```
/var/www/
      /templates/
            |__ nav_bar.html [M]
            |__/twittermon/
               |__ menu_sidebar.html
      /modules/
         |__ /TwitterMon/
                  |__ Flask_TwitterMon.py
                  |__ templates/
                        |__ header_TwitterMon.html
                        |__ TwitterMon.html
                        |__ TwitterMon_results.html
                        |__ TwitterMon_settings.html
                  |__ config/
                        |__ TwitterMon.cfg
```

El acceso a la interfaz de usuario, una vez arrancada la plataforma AIL, se puede realizar haciendo clic en el enlace 

de la barra superior de la plataforma AIL:

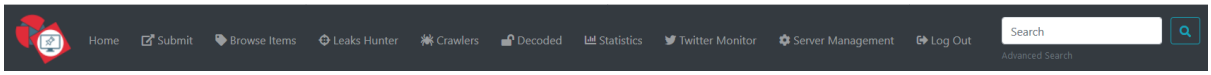


Figura 7: Menú superior de la plataforma AIL.

Una vez el usuario se encuentre en la interfaz del módulo de monitorización, podrá navegar entre las tres páginas del sistema, a través del menú disponible en la barra lateral izquierda:

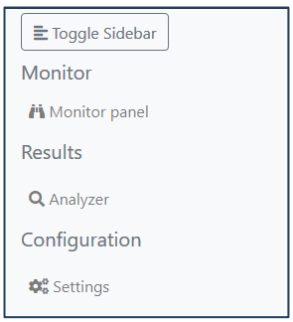


Figura 8: Menú lateral del módulo de monitorización.

4.2.1. Interfaz para la monitorización de Twitter

La interfaz para la monitorización de Twitter, accesible en la dirección `https://<AIL-SERVER>:7000/TwitterMon/`, permite al usuario activar o desactivar la monitorización, así como seguir el estado de la monitorización en curso y establecer los parámetros de dicha monitorización.

La siguiente imagen muestra el aspecto de la interfaz web para la monitorización de Twitter:

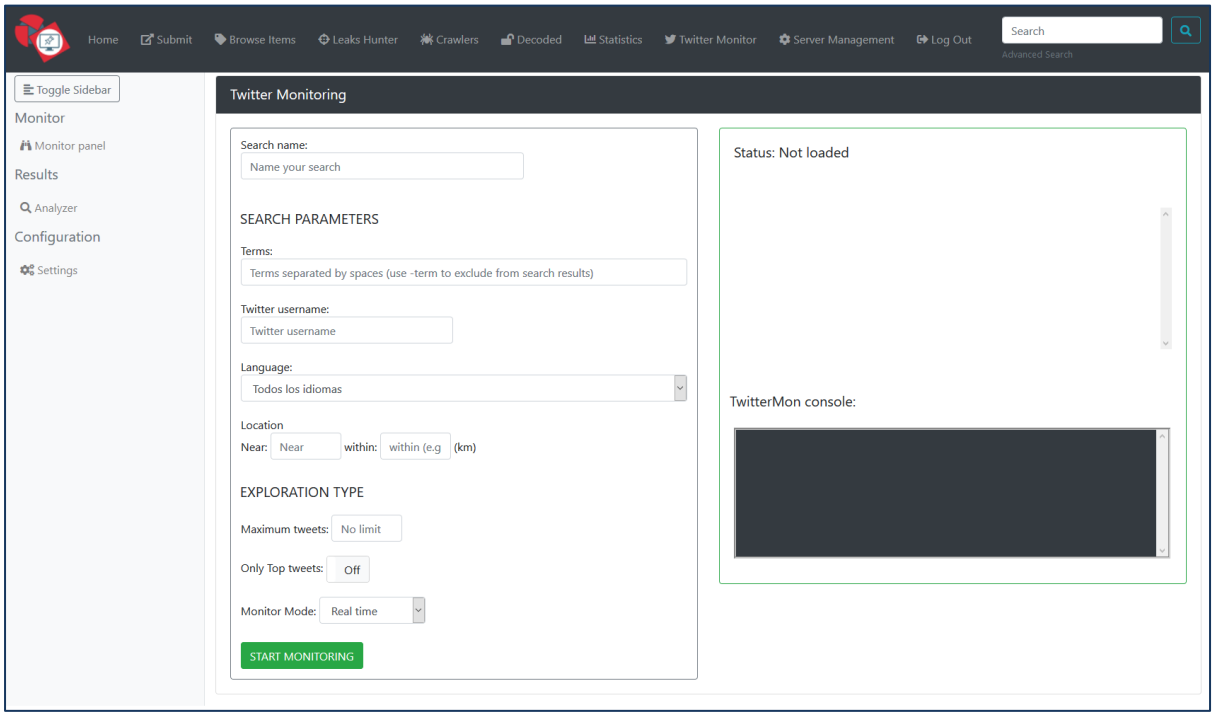


Figura 9: Interfaz para la monitorización de Twitter.

4.2.1.1. Funcionalidades

A. Parametrización de la monitorización

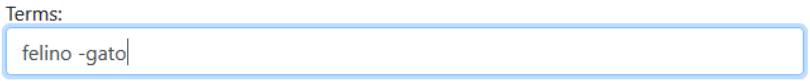
La solución permite realizar una monitorización enfocada, permitiendo establecer parámetros de filtrado y acotación de los tweets a monitorizar. La configuración de la búsqueda y los parámetros que el usuario podrá establecer para la búsqueda de publicaciones, divididos en tres áreas, son los siguientes:

IDENTIFICACIÓN DE LA BÚSQUEDA

- Nombre de la búsqueda: Nombre que el usuario da a la búsqueda y que se usará para identificarla. Si el nombre ya existe en la base de datos[24], la información y los análisis realizados se añadirán a los existentes, de esta manera se puede detener una monitorización y retomarla en cualquier momento.

PARÁMETROS DE LA BÚSQUEDA

- Términos: Se recopilarán las publicaciones que contengan algunas de las palabras clave introducidas por el usuario. El módulo permite excluir términos de los resultados de búsqueda introduciendo el signo – delante del término, por ejemplo, la siguiente búsqueda recogerá las publicaciones que contengan la palabra "felino" pero descartará las que contengan la palabra "gato":



- Usuario: Se buscarán las publicaciones de un usuario específico.
- Localización geográfica: Se seleccionarán las publicaciones realizadas en un lugar y radio de acción concretos.
- Idioma: El sistema permite especificar que sólo se recopilen las publicaciones que estén en un idioma en concreto.

TIPO DE MONITORIZACIÓN

- Número máximo: Se permitirá seleccionar el número máximo de Tweets que serán recopilados por el módulo. Una vez el módulo haya procesado el número de publicaciones establecidas por este parámetro, la monitorización se detendrá.
- Tweets destacados: Se dará la opción de recoger sólo de entre las publicaciones destacadas, conocidas como Top Tweets.
- Modo de monitorización: Elección entre monitorización en tiempo real o monitorización de un periodo de tiempo concreto.

[24] La página avisará al usuario de este hecho.

B. Modos de monitorización

La interfaz permite escoger entre una **monitorización en tiempo real**, que se ejecutará de manera continua, analizando las nuevas publicaciones o la **monitorización de un periodo de tiempo en concreto**. El detalle sobre la implementación de ambos tipos de monitorización se describe en el módulo de recopilación y suministro de la información, sección 4.3.

El tipo de monitorización se escoge mediante el campo desplegable [Real time] que se encuentra dentro de la sección denominada **Exploration Type**.

Monitorización en tiempo real

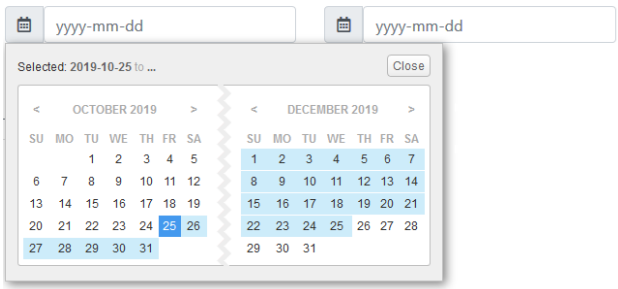
La monitorización en tiempo real analiza las publicaciones que se realicen en Twitter, en base a los parámetros de búsqueda, de manera continua e ininterrumpida, hasta que el usuario cancele la operación o hasta que se recopile el número máximo de Tweets establecidos por el usuario en el campo [Maximum Tweets], si fuera el caso.

Los detalles sobre los periodos de refresco y la horquilla de tiempo que el monitor utiliza se detallan en la sección 4.3.

Monitorización de un periodo de tiempo

En el caso de elegir la monitorización de un periodo de tiempo específico, se mostrarán dos campos para poder seleccionar la fecha de inicio y la fecha de fin del periodo a monitorizar.

Hay que tener en cuenta que, si el usuario opta por establecer un número máximo de Tweets a recolectar, solo se recogerán ese número máximo de publicaciones en el periodo seleccionado.



C. Estado de la monitorización

La interfaz de monitorización ofrece al usuario un panel con información sobre el estado actual de la monitorización. La siguiente imagen muestra un ejemplo del panel de información mostrando información sobre una monitorización en curso:

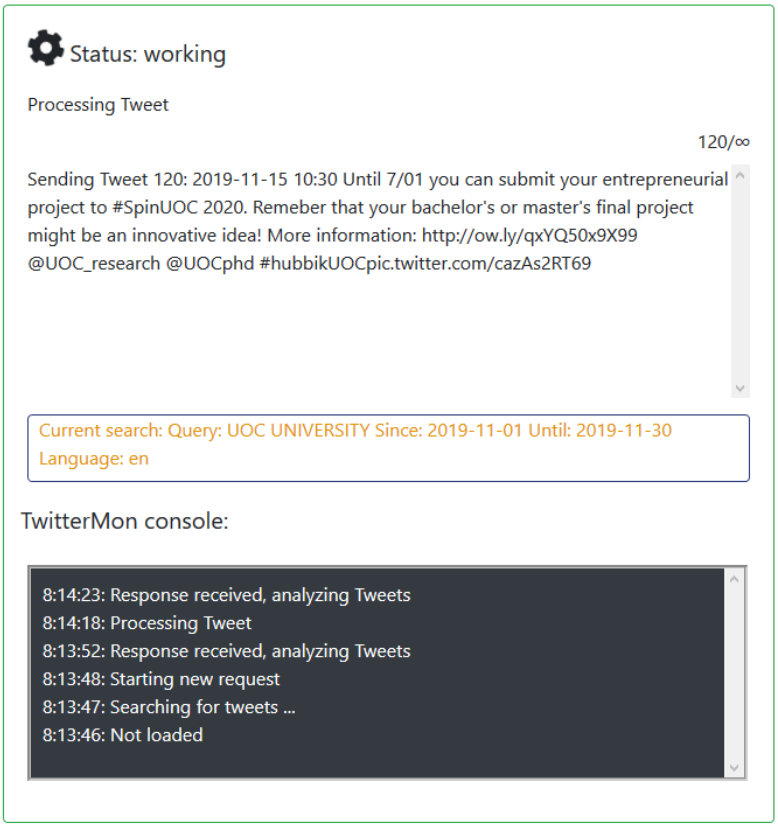


Figura 10: Panel de información de la interfaz de monitorización.

Los datos que se presentan son los siguientes.

- Estado de la monitorización (status): La monitorización podrá estar en alguno de los siguientes estados:
  - Working: indica que la monitorización está en curso.
  - Cancelled: indica que la monitorización ha sido cancelada por el usuario.
  - Finished: indica que la monitorización ha terminado ya que se ha alcanzado el número máximo de tweets o no existen más tweets en base a los parámetros de búsqueda (monitorización de un periodo de tiempo).
  - Error: indica que ha ocurrido un error.

- Descripción del estado (status description): Añade detalle al estado definido por "status". Puede mostrar indicaciones como "Cancelled by user" o "Searching for tweets".
- Procesando (processing): Este estado recoge la fecha y texto del tweet que se está tratando y que va a ser enviado a la cola de procesamiento de la plataforma AIL.
- Progreso (progress): Se trata de un contador que indica el número de tweets procesados hasta el momento. Un tweet procesado implica que el tweet ha sido recibido y enviado a la cola de procesamiento.

En cuanto el módulo inyecte Tweets en la cola de procesamiento de AIL, se podrán empezar a visualizar los resultados del análisis en la página de resultados.

4.2.1.2. Diseño

El diseño de la interfaz de monitorización obedece al siguiente diagrama:

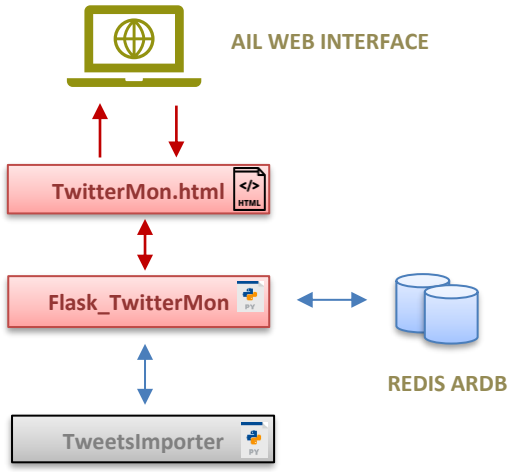


Figura 11: Diseño de la interfaz de monitorización.

TwitterMon.html

El archivo `/var/www/modules/TwitterMon/templates/TwitterMon.html` es el responsable de la representación visual de la página de monitorización, de enviar las solicitudes de

monitorización al fichero **Flask_TwitterMon** y de recibir el estado de la monitorización de **Flask_TwitterMon**. De esta página web, hay que destacar que:

- La página realiza una validación de datos en cliente, comprobando que el valor introducido por el usuario para los parámetros **Máximum Tweets** y **Within** es un número entero y que el valor introducido para **Search name** no contiene caracteres especiales.
- Comprueba si el nombre de la búsqueda escogido por el usuario ya ha sido utilizado, en cuyo caso informará al usuario de que los resultados se combinarán.
- Hace uso de las siguientes librerías:
    - jQuery.
    - Bootstrap.
    - Font-awesome.

Flask_TwitterMon.py

El archivo PYTHON, ubicado en `var/www/modules/TwitterMon/`**Flask_Twittermon.py,** actúa como intermediario entre la página HTML, que recoge las ordenes de usuario, y los módulos de monitorización y bases de datos utilizadas por la solución. En el caso de la interfaz de monitorización, **Flask_Twittermon.py** recoge los parámetros de la consulta introducidos por el usuario y envía la orden al fichero python **TweetsImporter.py**, que se detalla en la sección 4.3. De este fichero, se destacan las siguientes funciones:

- Funciones de consulta:

  Las funciones `get_Status()`, `get_StatusDescription()`, `get_Processing()` y `get_Progress()` son las encargadas de consultar aspectos distintos del estado de la monitorización para su presentación en la página de monitorización. Por otro lado, la función `check_if_search_exists()` se encarga de consultar si el nombre de búsqueda recibido ya existe en la base de datos.

- Funciones de monitorización:

  La función `start_monitoring()` es la encargada de recoger los parámetros de la monitorización enviados desde **TwitterMon.html** e invocar al script de monitorización **TweetsImporter.py**. Por otro lado, `stop_monitoring()` se encarga de enviar la orden de cancelación de la monitorización realizada por el usuario.

4.2.2. Interfaz de resultados

La interfaz de presentación de los resultados, accesible en la dirección `https://<AIL-SERVER>:7000/TwitterMon/results_page`, muestra el listado de búsquedas realizadas, así como el detalle y análisis realizado sobre cada una de las muestras. La interfaz tiene **tres áreas** diferenciadas, una dedicada al **listado y manejo de las búsquedas realizadas,** otra con **información de interés general** y finalmente, el área de **detalle de la búsqueda seleccionada**.

La siguiente imagen muestra el aspecto de la interfaz web de resultados:

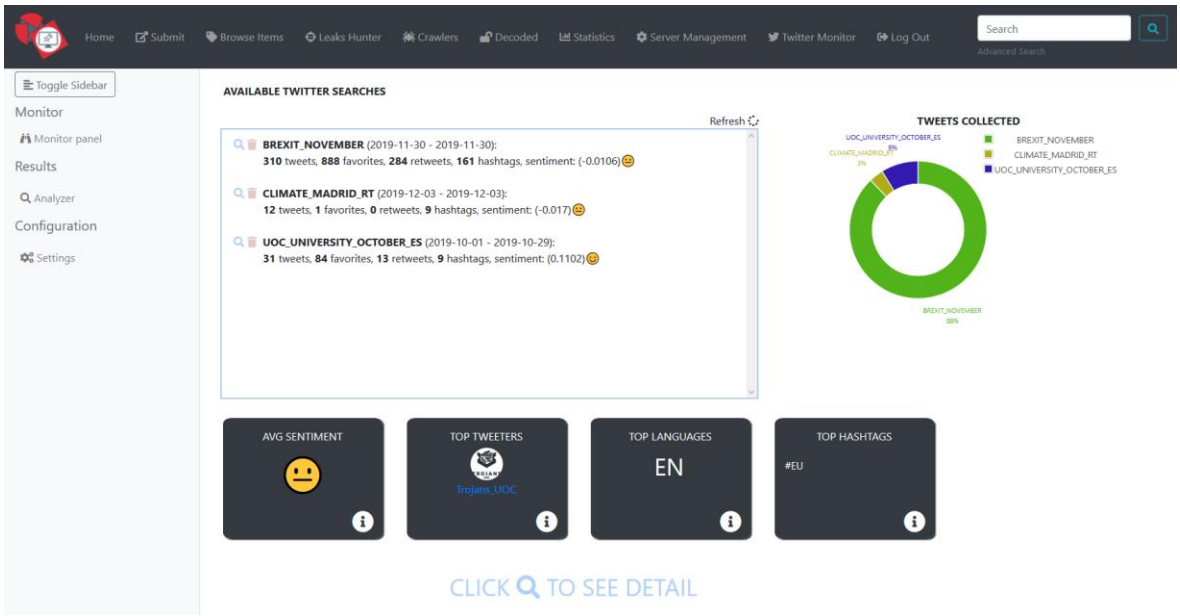
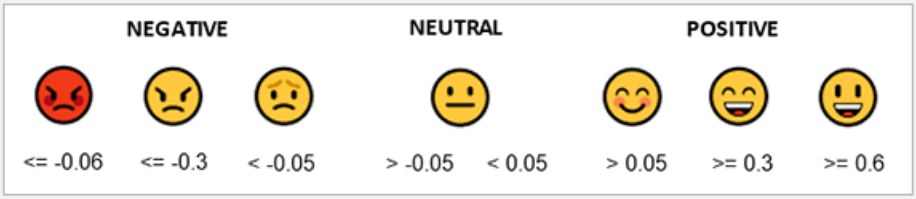


Figura 12: Interfaz de resultados.

El análisis de sentimiento realizado sobre los diferentes valores que se presentan en la página de resultados muestra una serie de emoticonos en base a los resultados obtenidos (valor compound del análisis de sentimiento). Las relaciones entre el icono mostrado y el valor obtenido son las siguientes:



4.2.2.1. Funcionalidades

A. Gestión de las búsquedas realizadas

El área que muestra el listado de búsquedas ofrece un listado de todas las búsquedas realizadas, acompañadas de información general interesante sobre cada una de ellas, como el número de tweets recopilados, el número de tweets marcados como favoritos, el sentimiento general, etc.

Además, la interfaz permite la eliminación de una búsqueda y el acceso al detalle del análisis realizado sobre ella:

- Si se pulsa sobre el icono , se muestra en la parte inferior de la página el detalle del análisis realizado sobre esa búsqueda.
- Mediante el icono , el usuario puede eliminar, tras una confirmación de seguridad, la búsqueda seleccionada.
- El botón de **Refresh** permite actualizar la página en caso de que exista una monitorización en curso.

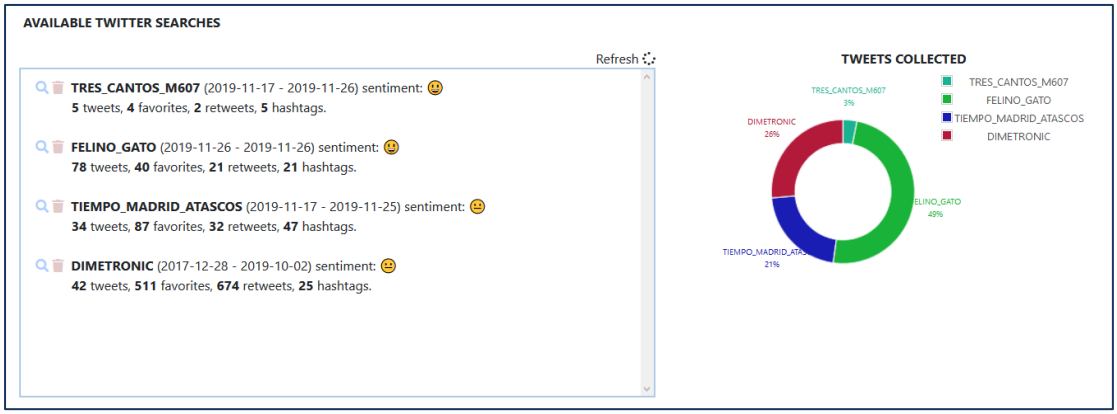


Figura 13: Interfaz para la monitorización de Twitter: Área de búsquedas.

B. Información general de interés

En la parte central de la página se muestran varias tarjetas con información relativa a todas las búsquedas realizadas: la media del análisis de sentimiento de todas las búsquedas, los usuarios con más Tweets, los idiomas más utilizados y los hashtags que aparecen más veces.

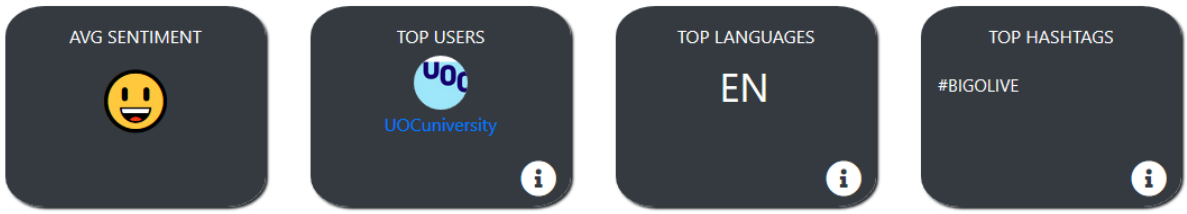


Figura 14: Interfaz para la monitorización de Twitter: Área de tarjetas resumen.

Además, si se pulsa sobre el icono se desplegará el listado de los 5 elementos con más apariciones en cada uno de los resultados (excepto para el análisis de sentimiento) con un enlace a la página de Twitter del usuario o del hashtag.

C. Resultado del análisis de los Tweets

En la parte inferior de la página, una vez el usuario pulse sobre el icono de una de las búsquedas realizadas, se muestra la información del análisis realizado sobre cada una de las búsquedas, además de permitir filtrar y analizar los resultados presentados.

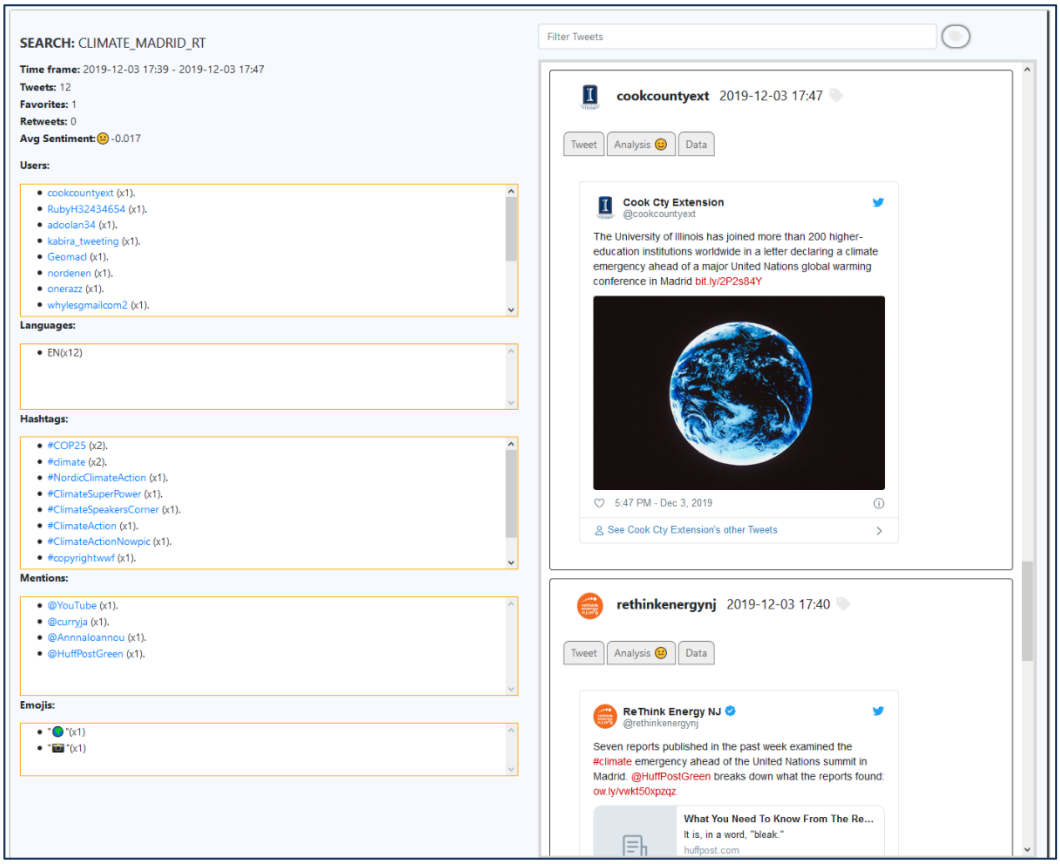


Figura 15: Interfaz para la monitorización de Twitter: Área de Tweets.

Esta área se divide en dos zonas: la columna izquierda muestra la información del análisis de la búsqueda mientras que en la columna derecha se muestran los tweets recopilados junto con información de su tratamiento.

SECCIÓN GENERAL

Esta sección muestra información de la búsqueda como el periodo de tiempo en el que se encuentran los Tweets, así como el número de Tweets, favoritos, hashtags, etc. Además, se listan en áreas diferenciadas, y ordenados por el número de apariciones, todos los usuarios, idiomas, hashtags, menciones y emojis de los Tweets recopilados.

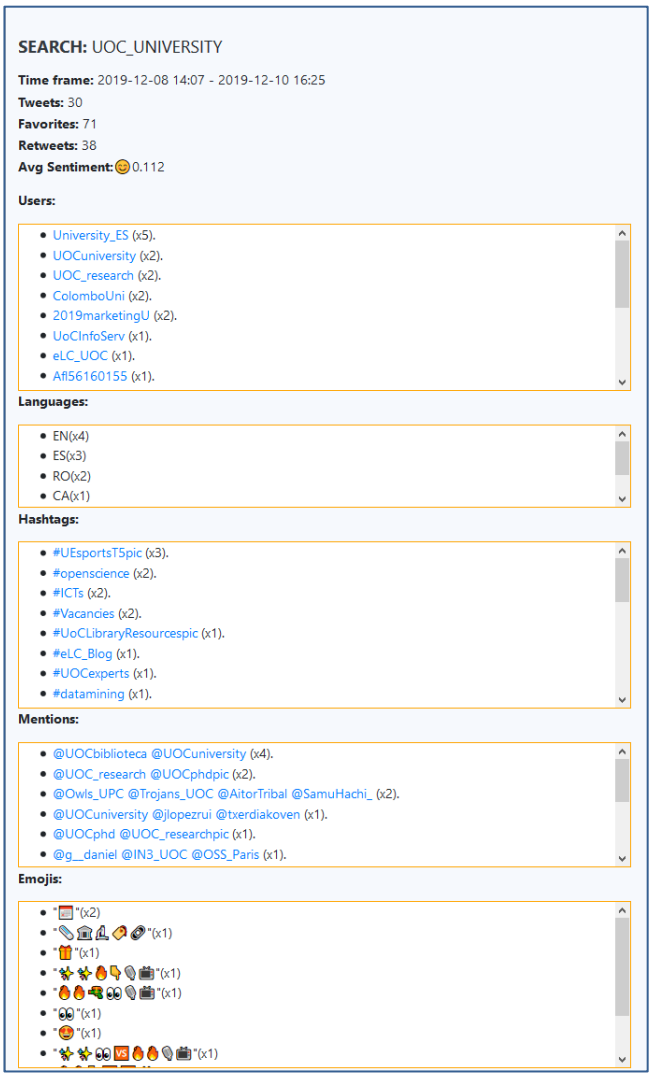


Figura 16: Interfaz para la monitorización de Twitter: Área de Tweets, sección general.

SECCIÓN DE TWEETS

Este panel con scroll vertical muestra todos los tweets que se han recopilado en la búsqueda, así como la siguiente información para cada uno de ellos:

- El tweet en su maquetación oficial de Twitter.
- El texto del tweet.
- El texto del tweet traducido al inglés.
- El resultado del análisis de sentimiento realizado.

Además, se permite al usuario marcar el Tweet, a través del icono , para su posterior seguimiento.

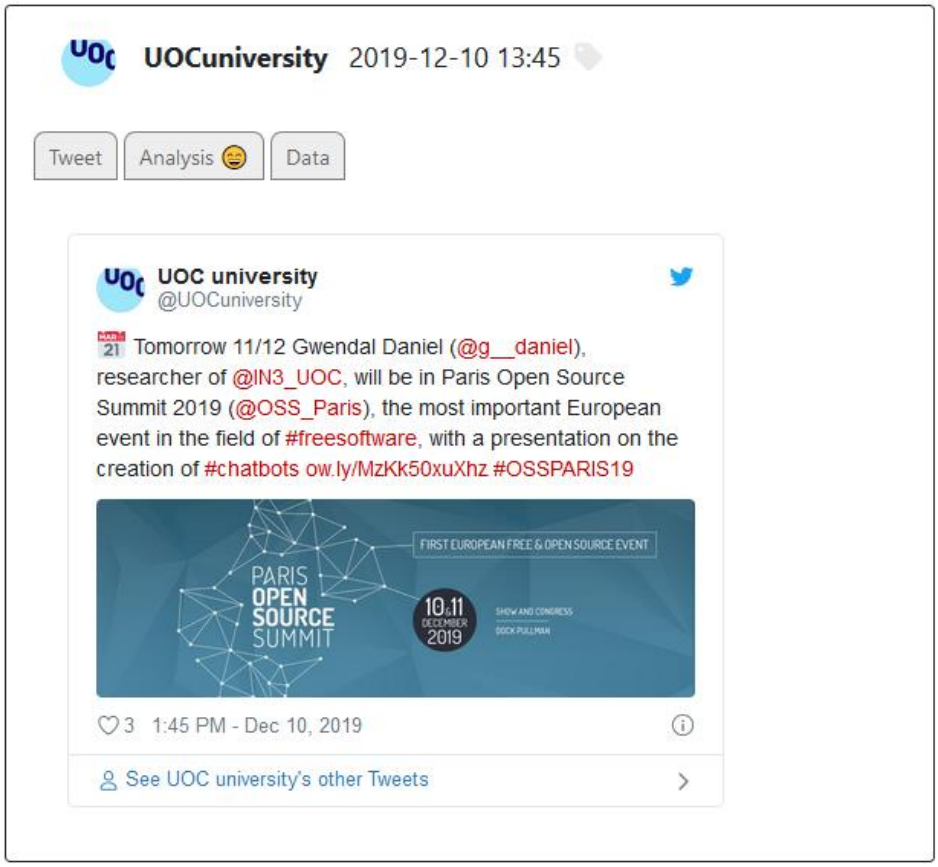


Figura 17: Presentación del Tweet.

El módulo permite filtrar el listado de Tweet mediante la introducción de términos en la barra de búsqueda. Además, presionando el icono , se pueden filtrar los Tweets etiquetados como importantes por el usuario.

4.2.2.2. Diseño

La interfaz de presentación de resultados obedece al siguiente diagrama:

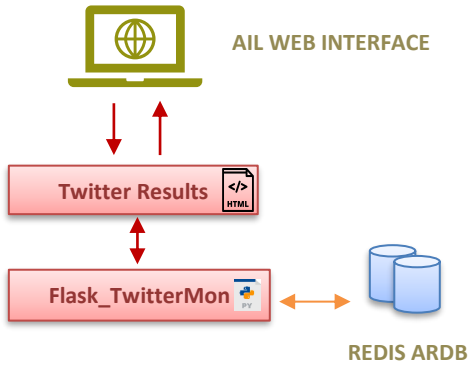


Figura 18: Diseño de la interfaz de resultados.

TwitterMon_results.html

El archivo `/var/www/modules/TwitterMon/templates/TwitterMon_results.html` es el responsable de la representación visual de la página de presentación de los resultados. Destacamos:

- Recalcula los resultados en caso de eliminar una búsqueda.
- Obtiene los enlaces a Twitter de los usuarios y hashtags mostrados.
- Permite etiquetar Tweets considerados como importantes.
- Hace uso de las siguientes librerías:
  - jQuery.
  - Bootstrap.
  - Font-awesome.
  - Platform.twitter.com/widget.

Flask_TwitterMon.py

El archivo `/var/www/modules/TwitterMon/Flask_Twittermon.py`, actúa como intermediario entre la página HTML, que presenta los resultados de los análisis realizados, y las bases de datos que contienen los datos recopilados. En concreto, el fichero Python transmite la orden de listado de búsquedas, Tweets y resultados y las órdenes de eliminación de las búsquedas seleccionadas por el usuario. De este fichero, se destacan las siguientes funciones:

- Funciones de consulta:

  Las funciones `get_twitter_search_general_data()`, `get_tweets_from_twitter_search()` y `get_list_of_searches()` realizan las consultas a las bases de datos.

- Funciones de borrado:

  Las función `delete_twitter_search_data()` elimina una búsqueda de las bases de datos.

4.2.3. Interfaz de configuración

La interfaz de configuración, accesible en la dirección `https://<AIL-SERVER>:7000/TwitterMon/settings_page`, permite modificar ciertos parámetros de configuración del módulo, almacenados en el fichero `/var/www/modules/TwitterMon/config/TwitterMon.cfg`.

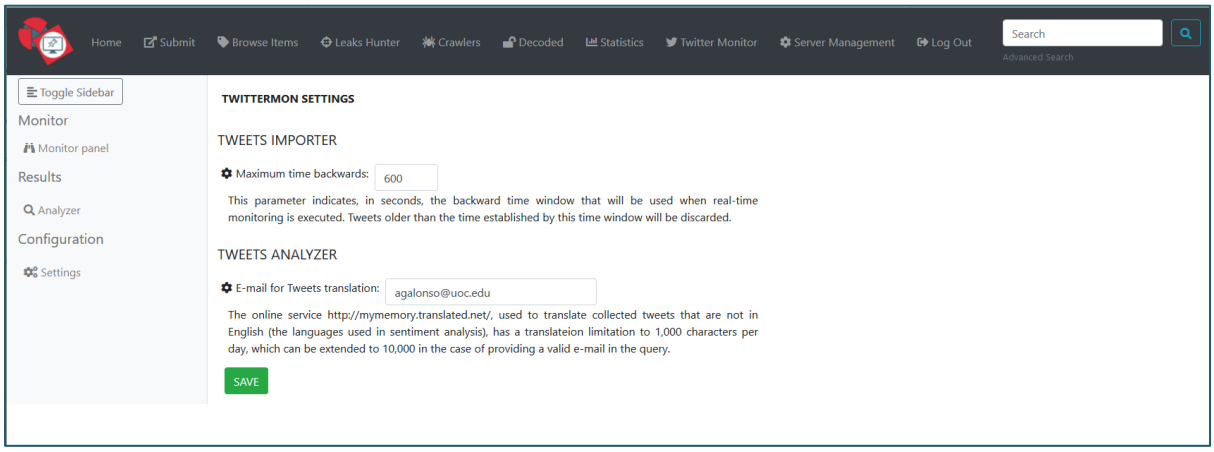


Figura 19: Interfaz para la configuración del sistema.

4.2.3.1. Funcionalidades

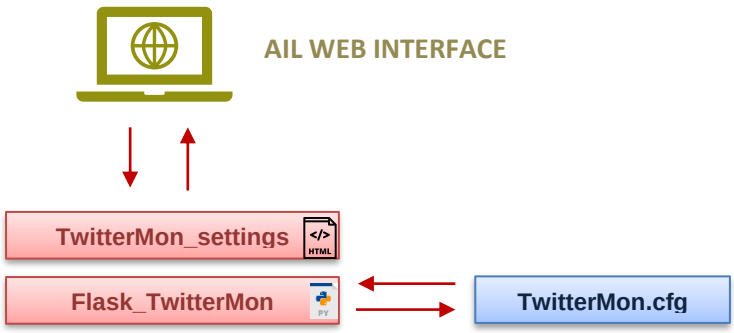
Los parámetros que se pueden configurar son los siguientes:

- **Maximum Time Backwards**: Este parámetro, almacenado como `real_time_max_time_backwards` en el fichero de configuración, indica, en segundos, la ventana de tiempo hacia atrás que se utilizará cuando se esté haciendo una monitorización en tiempo real. Los tweets que tengan una fecha más antigua a la que marque la ventana de tiempo serán descartados.

- **Translation e-mail**: El servicio online `http://mymemory.translated.net/`, utilizado para traducir los tweets recopilados que no estén en inglés (el idioma utilizado en el análisis de sentimiento), tiene una limitación de traducción a 1.000 caracteres diarios, que puede ser ampliada a 10.000 caracteres en el caso de proporcionar un e-mail válido en la consulta. Este parámetro, almacenado como `email_for_translation` permite establecer dicho e-mail.

4.2.3.2. Diseño

El diseño de la interfaz de configuración obedece al siguiente diagrama:



TwitterMon_settings.html

El archivo `/var/www/modules/TwitterMon/templates/TwitterMon_settings.html` es el responsable de la representación visual de la página de configuración y de enviar los cambios introducidos por el usuario al fichero `Flask_TwitterMon`. De esta página web, hay que destacar que:

- La página realiza una validación de datos en cliente, comprobando que el valor introducido por el usuario para el parámetro `real_time_max_time_backwards` es un número y que el valor introducido para `email_for_translation` tiene el formato de un correo electrónico.
- Muestra en pantalla el resultado de la operación de grabar los datos en el archivo de configuración.
- Hace uso de las siguientes librerías:
  - jQuery.
  - Bootstrap.
  - Font-awesome.

Flask_TwitterMon.py

El archivo `/var/www/modules/TwitterMon/Flask_TwitterMon.py` es el interfaz entre la página web de configuración y el archivo de configuración `TwitterMon.cfg`. De este archivo Python, hay que destacar que:

- La función `get_configuration()` se encarga de leer la configuración del fichero **`TwitterMon.cfg`** y de enviarla a la página web para ser mostrada.
- La función `save_configuration()` se encarga de escribir la configuración establecida por el usuario en el archivo **`TwitterMon.cfg`**.

TwitterMon.cfg

Este archivo de texto plano almacena los datos de configuración del módulo de monitorización. La estructura del archivo es la siguiente:

```
[TwitterAnalyzer]
email_for_translation = my-emailo@uoc.edu

[TweetsImporter]
real_time_max_time_backwards = 600
```

4.3. Módulo de recopilación y suministro de la información

El módulo de recopilación y suministro de la información se encarga de realizar las consultas a Twitter, para la obtención de los Tweets que cumplan con los criterios de búsqueda, y su posterior inyección en la cola de procesamiento de la plataforma AIL. El módulo está compuesto por los siguientes ficheros:

```
/var/www/
      /modules/
        |__ /TwitterMon/
                |__ TweetsImporter.py
                |__ TM_Status.py
                |__ data/
                |__ logs/
```

4.3.1. Recopilación de Tweets

El módulo de recopilación de Tweets realiza consultas a Twitter, como si de un navegador web se tratase, y recibe bloques de archivos JSON con los tweets que cumplan con los criterios de la búsqueda.

4.3.1.1. Funcionalidades

A. Tipos de monitorización

Como se ha explicado anteriormente, la solución propuesta en este TFM permite realizar tanto una monitorización en tiempo real, como una monitorización de un periodo de tiempo predefinido. En el caso concreto de la **monitorización en tiempo real**, dado que el sistema recibe Tweets secuencialmente y estos se reciben siguiendo un orden temporal que posiciona en primer lugar los más recientes, surge la necesidad de establecer una ventana de tiempo para la recolección. Esta ventana de tiempo abarcará un intervalo que irá desde el momento en que se envíe la petición hasta un periodo máximo de tiempo hacia el pasado (que será por defecto de 600 segundos). En el momento en que el sistema comience a recibir Tweets que hayan sido publicados en un momento anterior al límite inferior del intervalo que hemos definido, el sistema actualizará la petición.

B. Solicitud a Twitter

Este sistema para la recopilación de Tweets emula las consultas realizadas por un navegador web cuando interactúa con Twitter. Cuando se realiza una consulta HTTP GET a la dirección que se muestra a continuación, junto con una cabecera HTTP, se recibe una respuesta en formato JSON con un bloque de Tweets.

Solicitud:

```
https://twitter.com/i/search/timeline?q=
```

Cabecera de la solicitud:

```
Host: twitter.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate,
Connection: keep-alive
```

El siguiente listado detalla los campos, y los parámetros que los recogen, que se extraen del archivo JSON y que se utilizan para crear el objeto Tweet a tratar:

- **Identificador del Tweet** -> "data-tweet-id"
- **Usuario del Tweet** -> "span:first.username.u-dir b"
- **Texto del Tweet** -> "p.js-tweet-text"
- **Número de retweets** -> "span.ProfileTweet-action--retweet span.ProfileTweet-actionCount"
- **Número de veces marcado commo favorito** -> "span.ProfileTweet-action--favorite span.ProfileTweet-actionCount"
- **Timestamp** -> "small.time span.js-short-timestamp"
- **Idioma** -> "lang"
- **Enlace al Tweet** -> tweetPQ.attr("data-permalink-path")
- **Geolocalización** -> geoSpan = tweetPQ('span.Tweet-geo')

Una vez realizada la consulta, los archivos JSON son procesados para la extracción de la información relevante y la generación y posterior envío a la cola de procesamiento del objeto Tweet.

4.3.1.2. Diseño

El diseño del módulo de recopilación obedece al siguiente diagrama:

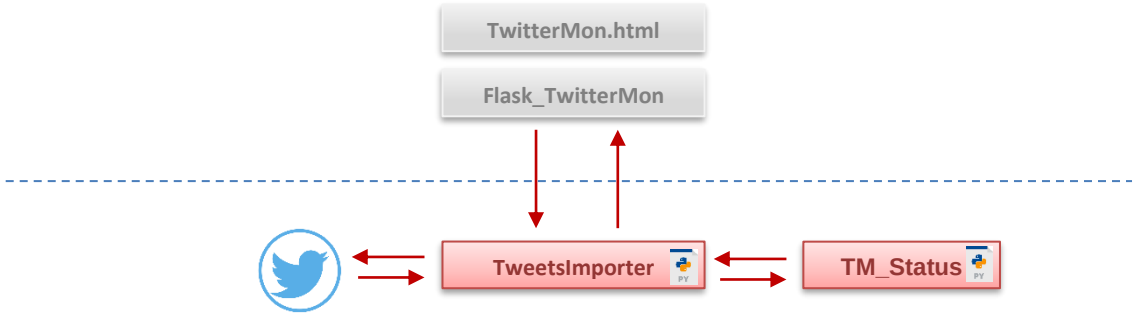


Figura 20: Diseño del módulo de recopilación de Tweets.

TweetsImporter.py

El archivo `/var/www/modules/TwitterMon/TweetsImporter.py` es el responsable de hacer las solicitudes a Twitter, en base a los parámetros recibidos desde `Flask_TwitterMon.py`. De este archivo, hay que destacar que:

- El archivo tiene un log, gestionado por la clase **TMLogger**, que se almacena en `/var/www/modules/TwitterMon/logs/TwitterMon.log`.
- El módulo permite su ejecución a través de la **línea de comandos**, útil para la realización de pruebas o para inyectar Tweets sin necesidad de utilizar la interfaz web.
- La **clase auxiliar Tweet** sirve para almacenar los Tweets recopilados en una estructura que permita el manejo de las publicaciones según sus características. La clase contiene los siguientes atributos:
    - Id: Identificador único del Tweet establecido por Twitter.
    - Permalink: Enlace al Tweet oficial.
    - Username: Usuario que publica el Tweet.
    - Text: Texto del Tweet.
    - Emojis: Emojis usados en el texto del Tweet.
    - Date: Fecha del Tweet.
    - Retweets: Número de retweets que se han realizado del Tweet.
    - Favourites: Número de veces que el Tweet fue marcado como favorito.

  - Mentions: Menciones realizadas en el Tweet.
  - Hashtags: Hashtags existentes en el texto del Tweet.
  - Geo: Geolocalización del Tweet.
  - Lang: Idioma del Tweet.

- La **clase auxiliar TweetCriteria** se utiliza para almacenar los parámetros de monitorización, establecidos por el usuario, para las consultas a Twitter. La clase contiene los siguientes atributos:

  - Username: Usuario del que se buscarán Tweets, en caso de haberse indicado.
  - Since: Fecha desde la que se buscarán Tweets, en caso de haberse indicado.
  - Until: Fecha hasta la que se buscarán Tweets, en caso de haberse indicado.
  - QuerySearch: Términos que se buscarán, en caso de haberse indicado.
  - MaxTweets: Número máximo de Tweets a recopilar, en caso de haberse indicado.
  - TopTweets: Indica que solo se deben seleccionar de entre los Top Tweets de Twitter.
  - Near: Localización en la que buscar Tweets, en caso de haberse indicado.
  - Within: Radio de búsqueda, respecto al campo "Near", en caso de haberse indicado.
  - Lang: Idioma de los Tweets, en caso de haberse indicado.

- Las **funciones getTweets()** y **getJsonResponse()** son las encargadas de la comunicación con Twitter y de la recepción de las respuestas en formato JSON.

- La **función manageMonitor()** recibe y gestiona las ordenes de arranque y parada de la monitorización.

- La **función manageStatus()** gestiona el estado de la monitorización para informar a la interfaz web y al log del sistema.

- Debido a la manera en que se implementa la monitorización en tiempo real es posible que se llegue a recibir el mismo Tweet de manera repetida en el tiempo. Por ello, el sistema comprueba si el fichero generado para el Tweet ya existe y, en ese caso, lo descarta y no lo envía a la cola de AIL.[25]

[25] Existe una segunda comprobación, que se explica en la sección 4.4.1, para la situación que se pueden dar al retomar una monitorización tras un periodo largo de tiempo.

TM_Status.py

Este archivo PYTHON sirve de apoyo al módulo de recopilación, guardando el estado de varias operaciones.

4.3.2. Inyección de Tweets

El módulo de inyección de Tweets procesa las publicaciones recibidas y las inyecta en la cola de procesamiento de AIL Framework.

4.3.2.1. Funcionalidades

La inyección a la cola de procesamiento de AIL se realiza a través de un socket con ZMQ a través del puerto 5556. La información inyectada es etiquetada con el formato "`TwitterMon >> [TM] searchname`" para su posterior identificación, siendo *searchname* el nombre introducido por el usuario para identificar la búsqueda y que será el nombré identificativo, dentro de la plataforma AIL, de dicha búsqueda.

4.3.2.2. Diseño

El diseño del módulo de inyección de publicaciones obedece al siguiente diagrama:

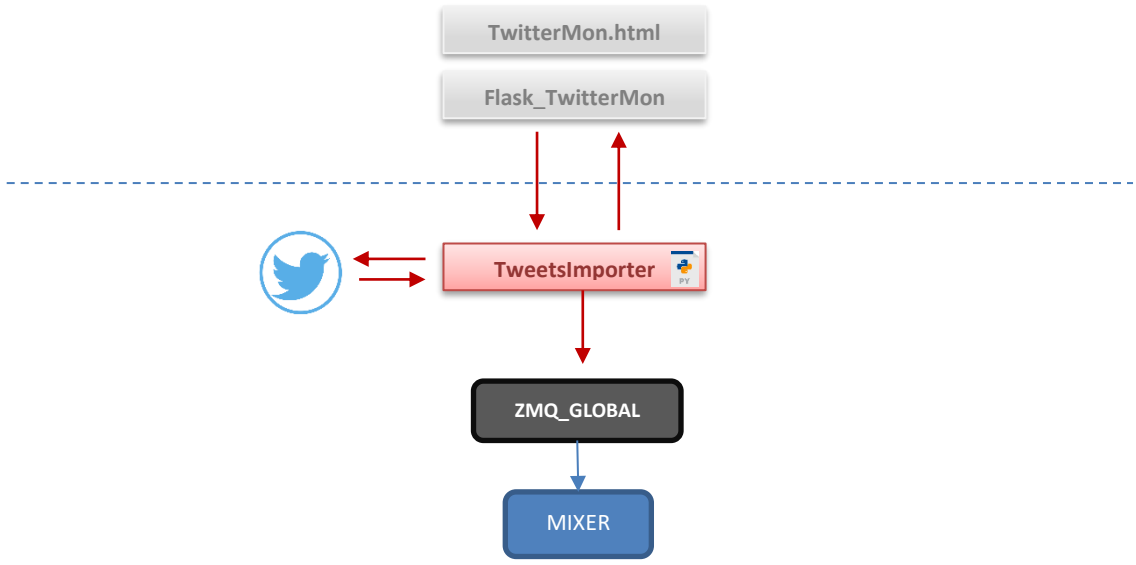


Figura 21: Diseño del módulo de inyección de Tweets.

TweetsImporter.py

El archivo `/var/www/modules/TwitterMon/TweetsImporter.py` es el responsable de inyectar los Tweets recopilados y publicados por el sistema. De este archivo, hay que destacar que:

- El socket de conexión a la cola ZMQ se establece en el puerto 5556.
- La información es etiquetada con el formato "`TwitterMon >> [TM] searchname`".
- El módulo permite su ejecución a través de la **línea de comandos**, útil para la realización de pruebas o para inyectar Tweets sin necesidad de utilizar la interfaz web.

4.4. Módulos de procesamiento

El módulo de procesamiento es el encargado de leer las publicaciones de Twitter recibidas, analizar los Tweets, realizar un análisis de sentimiento de los textoss y, finalmente, almacenar los resultados en la base de datos. El módulo está compuesto por los siguientes ficheros:

```
/bin/
   |__ LAUNCH.sh [M]
   |__ TwitterAnalyzer.py
   |__ packages/
         |__ modules.cfg [M]
         |__ Tweet.py
/logs/
```

4.4.1. Módulo de procesamiento

El módulo de procesamiento, lanzado al arrancar la plataforma AIL, sigue el modelo del resto de módulos de procesamiento de la plataforma AIL y el seguimiento del estado del análisis de la cola puede seguirse, como para el resto de los módulos de AIL, en la página principal del sistema:

Queue Name.PID	Amount
Indexer.23814	0
Mixer.23722	1
Keys.23823	0
Bitcoin.23839	0
Global.23731	0
Release.23857	8
Tools.23844	0
Phone.23850	4
TwitterAnalyzer.23931	9
SentimentAnalysis.23901	1
Decoder.23834	0
RegexTracker.23806	4
DomClassifier.23747	4
TermTrackerMod.23799	4

Figura 22: Colas de procesamiento de AIL.

4.4.1.1. Funcionalidades

A. Subscripción y lectura de la cola de mensajes

El módulo de procesamiento lee continuamente de la cola de Redis_Global, a la espera de recibir Tweets recopilados. Es importante señalar que el módulo ignorará cualquier publicación que no sea un Tweet recopilado por el módulo de monitorización de Twitter.

Además, en este módulo se hace una segunda comprobación (ver 4.3.1) para chequear si el Tweet ya ha sido recopilado y analizado ya que existe la posibilidad de que una monitorización haya sido parada y retomada en el tiempo, con lo que el archivo generado en la descarga del Tweet ya no existiría y la primera comprobación no lo detectaría.

B. Análisis estadístico de los Tweets

El módulo de procesamiento extrae la información relevante de los Tweets para calcular el número de hashtags, usuarios, retweets y favoritos totales, almacenar y procesar los emojis, analizar los usuarios más activos, calcular las menciones y traducir los textos al inglés.

C. Análisis de sentimiento de los Tweets

Finalmente, el módulo utilizar la librería vaderSentiment para realizar el análisis de sentimiento de los textos. De dicho análisis se extraen lo siguientes valores:

- Puntuación “pos”, “neu” y ”neg” que indican, con una suma total de 1, el balance de sentimiento del texto.
- Puntuación “compound” que se calcula sumando las puntuaciones de valencia de cada palabra del Tweet, ajustado en base al léxico de la librería de vadersentiment y normalizado para estar entre -1 (extremo negativo) y +1 (extremo positivo).

4.4.1.2. Diseño

El diseño del módulo de procesamiento de Tweets obedece al siguiente diagrama:

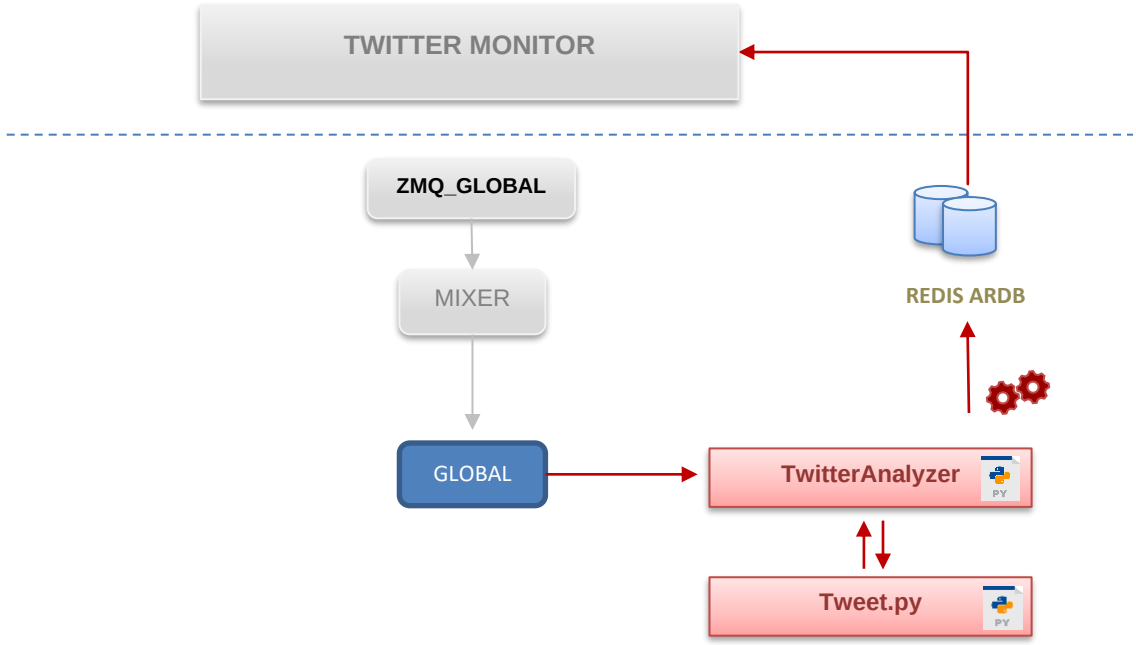


Figura 23: Diseño del módulo de procesamiento.

TwitterAnalyzer.py

El archivo `/bin/TwitterAnalyzer.py` es el responsable del análisis de los Tweets recopilados por el sistema. De este fichero, hay que destacar que:

- El módulo de procesamiento es lanzado al arrancar la plataforma a través del fichero `LAUNCH.sh`.
- Se conecta a la cola de mensajes de AIL a través de una subscripción a Redis_Global.
- Hace uso de la librería **ntlk.tokenize** para la descomposición de los Tweets y su posterior análisis de sentimiento.
- Utiliza la librería **vaderSentiment.vaderSentiment** para el análisis de sentimiento de los textos.
- Almacena en base de datos **información general de interés relativa a la búsqueda** de Tweets. La información que se almacena es: el número de hashtags de todos los Tweets contenidos en la búsqueda, el número de menciones de todos los Tweets, los idiomas de los Tweets, todos los emojis que aparecen en los Tweets, los usuarios que han publicado los Tweets, la fecha más antigua y la más reciente de los Tweets recopilados y el análisis de sentimiento medio de todos los Tweets recopilados.

- Almacena los **Tweets recibidos**, asociados con la búsqueda en la que fueron recopilados, junto con información de interés del análisis del Tweet. La información que se almacena para cada Tweet es: el identificador del Tweet, su fecha de publicación, el texto del Tweet, la traducción del Tweet al inglés (si fuera necesario), el usuario del Tweet, el idioma original, el análisis de sentimiento realizado sobre el texto del Tweet (incluyendo emoticonos y emojis) y un valor que indica si el Tweet ha sido marcado como importante en la interfaz de resultados (ver sección 4.2.2).

Tweet.py

El archivo `/bin/packages/Tweet.py` es una clase auxiliar que se encarga de crear el objeto Tweet a analizar y de la traducción de los textos al inglés, si fuera necesario. De este fichero, hay que destacar que:

- Procesa el mensaje recibido de la cola de AIL, creando un objeto Tweet.
- Hace uso de la herramienta online http://mymemory.translated.net para traducir los textos que no están en ingles ya que el análisis de sentimiento está preparado para dicho idioma.
- Hace uso de la librería **LanguageIdentifier** para identificar un texto del que se desconoce su idioma.
- Hace uso de las librerias **nltk.tokenize** y **textblob** para el tratamiento de los textos.

4.4.2. Bases de datos

La solución presentada en este TFM hace uso de la base de datos Redis ARDB, integrada en la plataforma AIL, para almacenar información. En concreto, se utilizan dos bases de datos:

Base de datos ARDB_TwitterAnalizer

Esta base de datos almacena un listado de las búsquedas realizadas, así como un resumen de los resultados de análisis.

La base de datos tiene la siguiente configuración:

```
[ARDB_TwitterAnalyzer]
host = localhost
port = 6382
db = 10
```

La base de datos tiene la siguiente estructura:

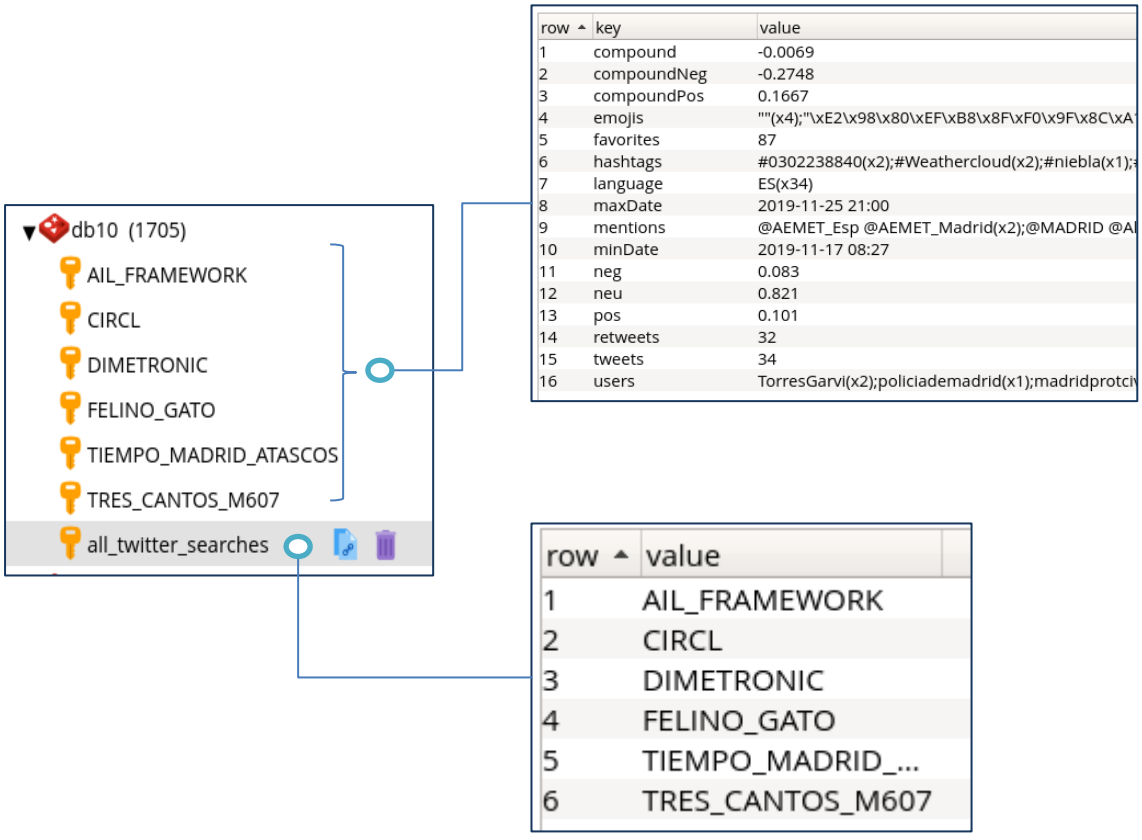


Figura 24: Base de datos ARDB_TwitterAnalizer.

La base de datos tiene una entrada principal, **all_twitter_searches**, que almacena un listado con todas las búsquedas realizadas por el usuario.

Además, el sistema almacena una **entrada por cada búsqueda** con información general relativa a dicha búsqueda.

Base de datos ARDB_TwitterTweets

Esta base de datos almacena, para cada búsqueda, **todos los tweets recopilados** junto con el resultado de varias operaciones de análisis realizadas sobre las publicaciones.

La base de datos tiene la siguiente configuración:

```
[ARDB_TwitterTweets]
host = localhost
port = 6382
db = 11
```

La base de datos tiene la siguiente estructura:

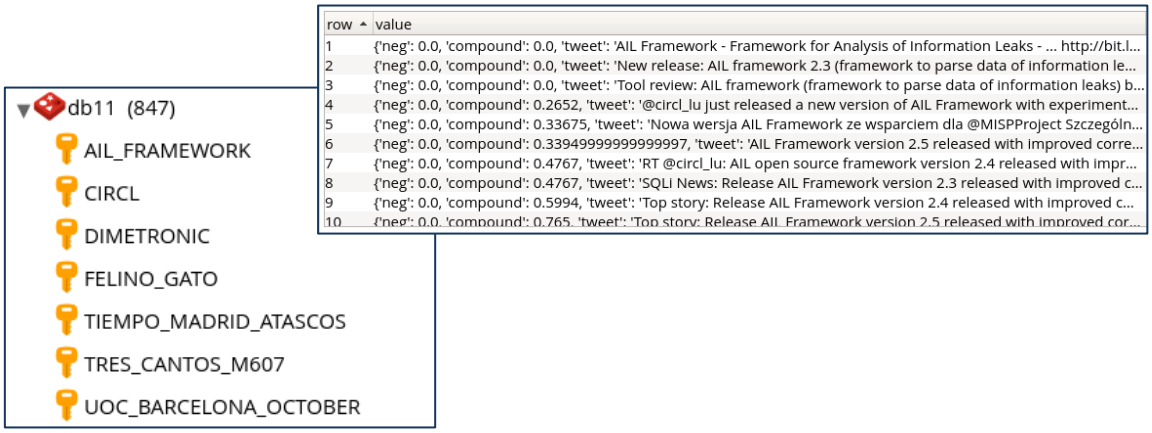


Figura 25: Base de datos ARDB_TwitterTweets.

Existe una entrada para cada búsqueda realizada y, en cada una de ellas, se almacenan los tweets recopilados e información del resultado del análisis de dichos Tweets.

4.5. Ficheros auxiliares

El sistema consta también de varios archivos auxiliares cuyo cometido es proporcionar accesos directos, dentro de la plataforma AIL, al módulo de monitorización de Twitter. Son los siguientes:

El archivo `/var/www/templates/twitterMon/menu_sidebar.html` es un archivo HTML muy simple que define la barra lateral que se presenta en la página de la solución y que permite cambiar entre la página web de monitorización, la página web de resultados y la página de configuración. Se escogió definir la barra lateral en este archivo para mantener la estructura utilizada por la plataforma AIL.

El archivo `/var/www/modules/TwitterMon/header_TwitterMon.html` es otro archivo auxiliar, que define el nombre y enlace a la página de monitorización de Twitter, desde la cabecera de la plataforma. Al igual que sucede con `menu_sidebar.html`, la decisión de diseño se basa en mantener la estructura existente en la plataforma AIL.

5. Instalación

A continuación, se describen los requisitos necesarios y el proceso para el despliegue y puesta en marcha de la solución.

Se presentan dos modos de instalación, uno automático que realiza el despliegue sobre una instalación existente de la plataforma AIL y otra manual que describe los pasos a seguir para realizar la instalación.

 En el repositorio de GitHub que aloja el proyecto se pueden encontrar las instrucciones de instalación en inglés.

5.1. Requisitos

- AIL Framework v2.2, v2.3, 2.4 o v2.5[26]
- Python >3 (AIL configura un entorno virtual en AILENV).
- Librería PyQuery.
- Librería VaderSentiment.

Si se desea que AIL no se actualice automáticamente (por si existieran problemas con versiones no probadas e integradas con la solución presentada en este TFM), se pueden desactivar las actualizaciones automáticas mediante el campo **auto_update** que se puede encontrar en el fichero AIL -framework/configs/**update.cfg**.

5.2. Instalación automática

Para la instalación automática, basta con ejecutar el archivo **installTM.sh** que se proporciona con el paquete de instalación de la solución. Este se encargará de copiar los archivos a las carpetas de la plataforma AIL, modificar los archivos necesarios de la plataforma (mediante el fichero python **modifyAIL.py**) y de instalar las dependencias que el sistema necesita.

[26] El sistema ha sido probado con las versiones 2.2, 2.3, 2.4 y 2.5 de la plataforma AIL.

5.3. Instalación manual

El siguiente método, además de ser de utilidad en el caso de que exista algún problema con la instalación automática, permite ver las dependencias necesarias y los cambios que hay que realizar sobre la plataforma AIL para desplegar el módulo de monitorización de Twitter.

Pasos:

1. Copia de ficheros

Es necesario copiar los archivos y carpetas que se indican en la columna de la izquierda a las carpetas de AIL que se indican en la columna de la derecha.

Paquete de instalación	Estructura de AIL Framework
AIL_bin/packages/**Tweet.py**	$AIL-framework/bin/packages
AIL_bin/**TwitterAnalyzer.py**	$AIL-framework/bin
AIL_var/www/modules/**TwitterMon**/	$AIL-framework/var/www/modules/
AIL_var/www/templates/**twittermon**/	$AIL-framework/var/www/templates/
fichero carpeta	

2. Instalación de librerías

El sistema necesita que se instalen las librerías PyQuery y VaderSentiment en el entorno virtual Python3 de la plataforma AIL:

```
# $AIL-framework/AILENV/bin/pip3 install pyquery
# $AIL-framework/AILENV/bin/pip3 install vaderSentiment
```

Nota: EL entorno virtual Python de la plataforma AIL se activa mediante el siguiente comando: source ~/ AIL-framework/AILENV/bin/activate

3. Modificación de archivos de AIL

Finalmente, es necesario modificar los siguientes ficheros de la plataforma AIL como se indica:

$AIL-framework/bin/packages/modules.cfg

Añadir la siguiente entrada:

```
[TwitterAnalyzer]
subscribe = Redis_Global
```

$AIL-framework/bin/LAUNCH.sh

Incluir la ejecución del script de procesamiento de Tweets:

```
sleep 0.1
screen -S "Script_AIL" -X screen -t "TwitterAnalyzer" bash -c "cd ${AIL_BIN};
${ENV_PY} ./TwitterAnalyzer.py; read x"
```

$AIL-framework/var/www/templates/nav_bar.html

Incluir el enlace a la aplicación:

```
<li class="nav-item mr-3">
<a          class="nav-link"          id="page-options"          href="{{
url_for('TwitterMon.TwitterMon_page') }}" aria-disabled="true"><i class="fab
fa-twitter"></i> Twitter Monitor</a>
 </li>
```

Una vez realizados estos pasos, ya se podría lanzar la plataforma AIL mediante su script de ejecución:

```
#  $AIL-framework/bin/LAUNCH.sh -l
```

6. Guía de usuario

En el repositorio de GitHub que aloja el proyecto se puede encontrar la guía de uso que se ha elaborado en inglés para el manejo del módulo desarrollado.

La guía de usuario no ha sido incluida en este documento para no hacerlo demasiado extenso y porque se estaría repitiendo información ya plasmada en la memoria.

7. Tests

Para la validación de la solución desarrollada se han llevado a cabo dos tipos de pruebas. Por un lado, se han realizado pruebas técnicas, para comprobar la solidez de la herramienta y detectar errores y, por otro lado, se han realizado pruebas de uso para verificar la utilidad de la solución en el control y análisis de fugas de datos y pulir así su funcionalidad.

Las pruebas que se han realizado se pueden englobar en los siguientes tipos:

- Pruebas funcionales en base a los requisitos y a la arquitectura del módulo.
- Pruebas de estrés de la monitorización.
- Pruebas de formato de los textos recopilados, incluyendo caracteres especiales.
- Pruebas de análisis de sentimiento, incluyendo idiomas diferentes, expresiones informales, emojis y emoticonos y analizando la coherencia de los resultados obtenidos.

8. Conclusiones y líneas de futuro

8.1. Conclusiones

Este trabajo ha conseguido el objetivo inicial marcado: desarrollar un módulo para la plataforma AIL que permita la monitorización de Twitter en tiempo real o en un periodo de tiempo establecido por el usuario, además de realizar un tratamiento estadístico y un análisis de sentimiento de los Tweets recopilados. De las características del módulo desarrollado se destacan las siguientes:

- Se trata de un módulo de monitorización independiente de la API de Twitter, que emula las consultas de un navegador web, y que por tanto no necesita ningún tipo de registro y autenticación ni está limitado por las funciones de la API oficial, alcanzándose, por lo tanto, los objetivos que se establecieron al comienzo de este trabajo:
    - Acceso total a la información publicada.
    - Método gratuito y sin limitaciones de uso.
    - Método relativamente consistente en el tiempo, sin una gran dependencia de los cambios realizados en la plataforma social o en la plataforma de control de fuga de datos.
- Realiza un análisis de sentimiento diseñado específicamente para las publicaciones que se realizan en Twitter, incluyendo emojis y emoticonos y el tipo de lenguaje que se utiliza en dicha plataforma social.

El desarrollo de este trabajo ha supuesto una inmersión en el aprendizaje de la plataforma AIL, en lenguajes de programación y sistemas no conocidos para el autor y en el tipo de comunicación que se realiza en la plataforma social Twitter.

El autor considera que el producto final puede ser de utilidad para equipos de respuesta ante incidentes de seguridad y, en general, para cualquier persona o entidad interesada en monitorizar posibles filtraciones en Twitter o en realizar análisis de las publicaciones realizadas en la red social.

8.2. Líneas de futuro

El módulo desarrollado en este TFM es susceptible de mejoras y de ampliaciones futuras, como las que se listan a continuación:

- El código es susceptible de ser mejorado y optimizado.
- Ni la plataforma AIL ni el módulo desarrollado son “responsive”, es decir, no están adaptados a dispositivos móviles. Esta adaptación de la interfaz de usuario sería una mejora interesante para facilitar el manejo de la herramienta en diferentes entornos.
- La traducción de los Tweets que no están en inglés se realiza con una librería externa que tiene limitaciones, algo que se ha intentado evitar en este TFM. El uso de un método alternativo, que no tenga limitaciones, sería una buena mejora para el módulo.

Anexos

ANEXO A Glosario

Término	Definición
API	Application Programming Interface
CERT	Computer Emergency Response Team (Equipo de Respuesta ante Emergencias Informáticas).
CSIRT	Computer Security Incident Response Team (Equipo de Respuesta ante Incidencias de Seguridad Informáticas).
CVE	Common Vulnerabilities and Exposures.
DLP	Data Loss Prevention.
TIC	Tecnologías de la Información y Comunicación.
TFM	Trabajo fin de Máster.

Tabla 4: Glosario de términos

ANEXO B Referencias y bibliografía

Ref.	Título
[1]	https://es.wikipedia.org/wiki/Sociedad_de_la_informaci%C3%B3n
[2]	https://es.wikipedia.org/wiki/Twitter
[3]	https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/
[4]	https://en.wikipedia.org/wiki/Twitter_usage, https://pearanalytics.com/
[5]	https://www.csuc.cat/es/comunicaciones/seguridad
[6]	https://developer.twitter.com/en/docs/tweets/search/overview
[7]	https://en.wikipedia.org/wiki/Sentiment_analysis
[8]	https://nlp.stanford.edu/sentiment/treebank.html
[9]	http://nlp.stanford.edu:8080/sentiment/rntnDemo.html
[10]	http://alias-i.com/lingpipe/web/download.html

Ref.	Título
[11]	https://pastebin.com/
[12]	https://www.python.org/
[13]	https://es.wikipedia.org/wiki/Flask
[14]	https://redis.io
[15]	https://zeromq.org/
[16]	Huina Mao, Xin Shuai, Apu Kapadia. Loose Tweets: An Analysis of Privacy Leaks on Twitter.
[17]	C.J.Hutto, Eric Gilbert. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text

Tabla 5: Referencias y bibliografía

ANEXO C Librerías y herramientas

Ref.	Título	
[18]	**GetOldTweets-Python**	Proyecto desarrollado en Python (v2) para la obtención de Tweets antiguos saltándose la limitación de la API oficial de Twitter. **Autor**: Jefferson Henrique **Código**: https://github.com/Jefferson-Henrique/GetOldTweets-python
[19]	**TextBlob**	Librería desarrollada en Python (v2 y v3) para el análisis de sentimiento en textos. **Web**: https://textblob.readthedocs.io/en/dev/
[20]	**LingPipe**	Librería desarrollada en Java para el análisis de sentimiento en textos. **Web**: http://alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html
[21]	**Stanford Sentiment Analysis Module**	Librería desarrollada en Java para el análisis de sentimiento en textos basada en el modelo de aprendizaje profundo. **Live demo:** http://nlp.stanford.edu:8080/sentiment/rntnDemo.html **Web:** https://nlp.stanford.edu/sentiment/ **Paper:** https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf
[22]	**CLIPS**	CLIPS (Computational Linguistics & Psycholinguistics Research Center) proporciona módulos para el etiquetado de textos y análisis de sentimientos entre otras funciones. **Web:** https://www.clips.uantwerpen.be/pages/pattern-en#sentiment
[23]	**Vader Sentiment**	Vader (Valence Aware Dictionary and sEntiment Reasoner) es una librería desarrollada en Python (v2 y v3), basada en reglas y léxicos y que está especializada en el análisis de publicaciones en redes sociales. **Autor**: cjhutto **Web:** https://github.com/cjhutto/vaderSentiment
[24]	**NTLK**	NLTK (Natural Language Toolkit) es una plataforma para crear programas de Python para trabajar con datos de lenguaje humano. **Web**: https://www.nltk.org/

Tabla 6: Librerías y herramientas

ANEXO D AIL Framework

A. Datos rápidos

Tecnologías	Python, Flask, Redis ARDB.
Versión	2.5
Código fuente	https://github.com/CIRCL/AIL-framework
Licencia	AGPL-3.0.
Desarrollador	CIRCL (https://www.circl.lu/)

B. Introducción

AIL es una plataforma de código abierto modular, desarrollada en python, para el análisis de datos con el fin de detectar filtraciones en fuentes de datos no estructurados.

La solución DLP nació como un proyecto interno de CIRCL en 2014 y está orientada principalmente al tratamiento de información publicada en sitios web para el almacenamiento y compartición de textos (fragmentos de código fuente, salidas de programas, etc.) como Pastebin[27], por el uso que se ha dado de estos sitios web para actos cibercriminales.

C. Características

Estas son las características principales de la plataforma AIL:

- Soporta multiprocesamiento por defecto, permitiendo que los módulos de análisis se inicien en tantas instancias como se necesiten.
- Permite la entrada de datos de múltiples orígenes de manera concurrente.
- Realiza un seguimiento de las entradas duplicadas.
- Proporciona un indexador de texto para indexar información no estructurada.

[27] https://pastebin.com/

- Permite el etiquetado de la información para su clasificación y búsqueda.
- Permite el envío de información a sitos de intercambio y publicación de amenazas y a plataformas de respuesta a incidentes (MISP y TheHive).
- Permite el seguimiento y cuantificación de las ocurrencias de términos, colecciones y expresiones regulares.

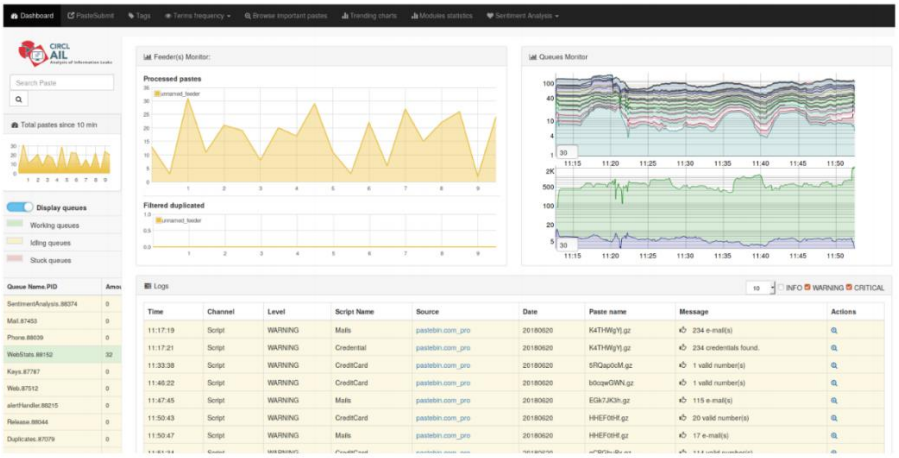


Figura 26: AIL Framework.

D. Tratamiento de la información

AIL ofrece **módulos de tratamiento** de los datos obtenidos (ver I.F) para los siguientes análisis de datos:

- Detección de:
  - Credenciales.
  - Datos de tarjetas de crédito.
  - CVE[28].
  - Direcciones de correo electrónico.
  - Teléfonos.
- Análisis de sentimiento.
- Extracción y validación de nombres de host.

[28] Sistema de clasificación para vulnerabilidades de seguridad conocidas.

Además, la plataforma puede ser ampliada, añadiendo módulos con nuevas funcionalidades, ya sea para la obtención de datos, su tratamiento o presentación.

E. Arquitectura

La plataforma AIL está basada en Python3[29] y consta de:

- Interfaz web: *Flask web interface* es la interfaz web, basada en Flask[30], que sirve de frontend para la plataforma.
- Bases de datos: la plataforma implementa Redis ARDB[31] como backend para la base de datos, compatible con Redis[32].
- Interfaces de comunicación internas: Consta de un sistema de publicación/subscripción de mensajes basado en Redis para la comunicación entre módulos y una cola ZMQ [33] para la inyección de datos en los módulos de procesamiento.

El siguiente diagrama oficial de CIRCL, ofrece una visión global de la arquitectura de suministro de la plataforma AIL:

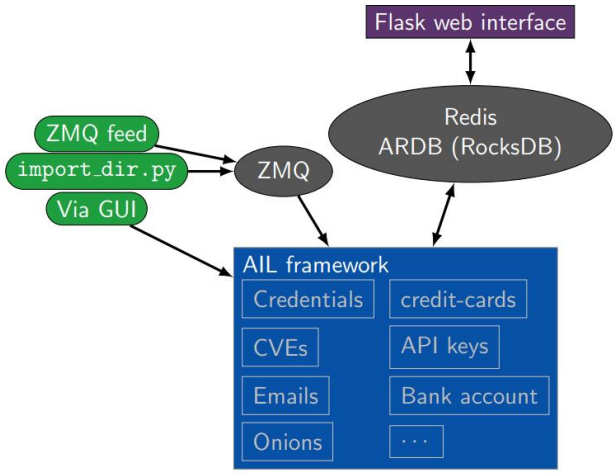


Figura 27: Arquitectura de suministro de AIL Framework (https://www.circl.lu)

29 https://www.python.org/

30 Framework minimalista, escrito en Python, para crear aplicaciones web: https://es.wikipedia.org/wiki/Flask.

31 Base de datos no sql.

32 https://redis.io

33 Librería de mensajería asíncrona: https://zeromq.org/

La siguiente figura muestra la arquitectura global de la plataforma AIL:

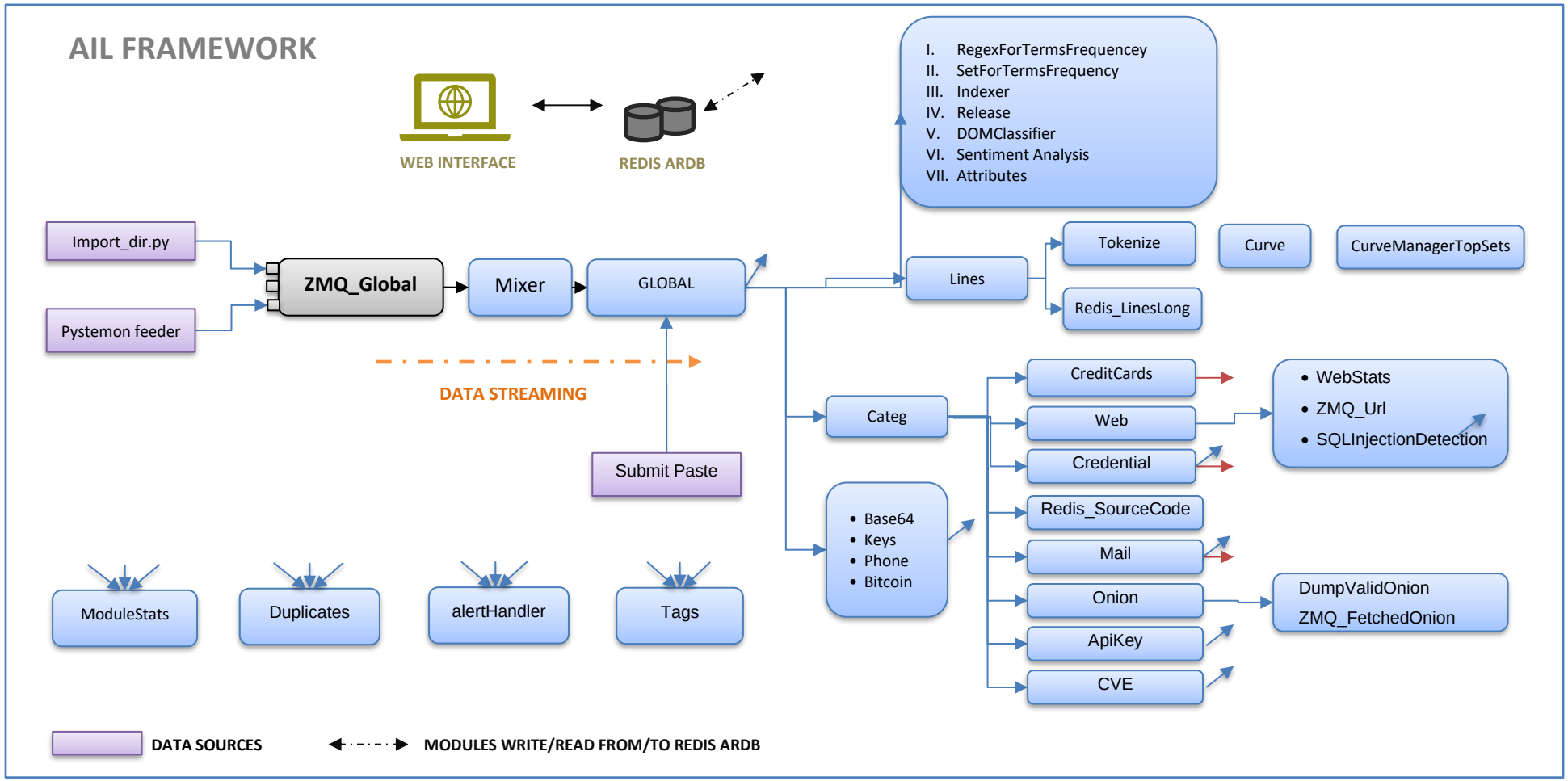


Figura 28: Arquitectura global de AIL Framework.

F. Obtención de datos

AIL ofrece varias formas de alimentar a la plataforma con datos para su análisis:

1. Mediante el feed oficial de CIRCL que envía a la IP que se configure, la información.
2. Importando un archivo comprimido (con una estructura específica), con los datos a procesar, a través de la pestaña que ofrece la interfaz web.
3. Mediante el script de importación, denominado como `import_dir.py` en la carpeta `/bin` de la plataforma.
4. A través de una instancia de **pystemon**, mediante el script denominado como `pystemon_feeder.py`, situado en la carpeta `bin/feeder/` de la plataforma.

Pystemon[34] es una herramienta de monitorización escrita en Python, independiente de la plataforma AIL, que nació para la recolección de pastes en sitios del estilo de PasteBin, y que puede ser utilizada para alimentar automáticamente la plataforma a través de la cola ZMQ.

El siguiente diagrama muestra el flujo para la importación de datos a través de Pystemon y del script de importación:

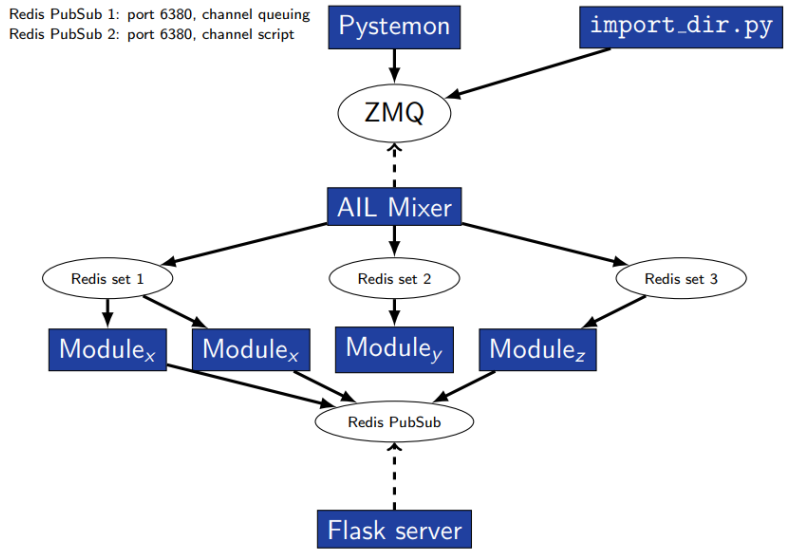


Figura 29: Alimentación de datos a través de Pystemon y del script de importación (https://www.circl.lu).

[34] https://github.com/cvandeplas/pystemon

Además, entre las funcionalidades de Pystemon, se destacan las siguientes:

- Búsqueda de expresiones regulares en pastes.
- Permite agregar fácilmente nuevos sitios de pastes, permitiendo funciones de descarga personalizada.
- Permite múltiples hilos por sitio para descargar los pastes.
- Para la conexión, es capaz de conectar con agentes de usuario y/o proxies aleatorios, descartando los proxies no confiables.
- Permite configurar una ventana de tiempo entre descargas, personalizable por cada sitio.
- Es capaz de descartar pastes en base a expresiones regulares.
- Permite destinatarios de correo distintos según el patrón de búsqueda.
- Es capaz de comprimir en Gzip.
- Puede ejecutarse como un demonio.

G. Instalación

En la página del repositorio de la plataforma, https://github.com/CIRCL/AIL-framework, se pueden consultar los pasos a realizar para la instalación de la herramienta. Además, CIRCL proporciona una máquina virtual para su descarga con la plataforma ya instalada.

H. Operaciones

En este apartado se presentan las operaciones básicas para operar con AIL Framework:

Arrancar la plataforma

Para arrancar la plataforma, hay que ejecutar el siguiente script:

```
# /bin/LAUNCH.sh -l
```

Una vez arrancada la plataforma, se puede acceder a la interfaz, mediante un navegador web:

```
http://<AIL-SERVER>:7000/
```

Las credenciales para el acceso web se encuentran en el archivo `DEFAULT_PASSWORD`. Una vez configurada la contraseña, este archivo se elimina.

El script de arranque de la plataforma permite además las siguientes acciones:

```
#  LAUNCH.sh -m
   [-l  | --launchAuto]        LAUNCH DB + Scripts
   [-k  | --killAll]           Kill DB + Scripts
   [-ks | --killscript]        Scripts
   [-u  | --update]            Update AIL
   [-c  | --crawler]           LAUNCH Crawlers
   [-f  | --launchFeeder]      LAUNCH Pystemon feeder
   [-t  | --thirdpartyUpdate]  Update Web
   [-m  | --menu]              Display Advanced Menu
   [-h  | --help]              Help
```

ANEXO E Twitter

A. Parámetros de un Tweet

La siguiente tabla lista los atributos que Twitter asigna a un Tweet:

Parámetro	Descripción
Tweet ID	Identificador único del Tweet
Conversation ID	Identificador de la conversación incluyendo respuestas, comentarios, etc. Si no existe conversación, será igual a Tweet ID.
Author Id	Identificador del autor creador del Tweet.
Author Name	Nombre del autor.
isVerified	Indica si la cuenta de Twitter está verificada.
DateTime	Fecha y hora de la creación del Tweet (Eastern time).
Language	Abreviatura del idioma del Tweet.
Tweet Text	Texto del Tweet, respuestas y rewteets.
Replies	Número de respuestas al Tweet dado.
Retweets	Número de retweets.
Favorites	Veces que se ha marcado como favorito por los usuarios.
Mentions	Todos los identificadores de usuario que aparecen en el texto de Tweet.
Hashtags	Todos los hashtags mencionados en el texto del Tweet.
Permalink	Enlace al Tweet.
URLs	Todas las URL que aparecen en el texto de Tweet dado.
isPartOfConversation	Indica si el tweet es parte de una conversación.
isReply	Indica si el tweet es una respuesta a otro tweet.
isRetweet	Indica si el tweet es un retweet.
Reply To User ID	ID de usuario del Tweet original al que se está respondiendo.
Reply To User Name	Nombre de usuario del Tweet original al que se está respondiendo.
Quoted Tweet ID	ID del Tweet que se está citando.
Quoted Tweet User Name	Nombre de usuario del tweet que se cita / retuitea
Quoted Tweet User ID	ID de usuario del tweet que se cita / retuitea

ANEXO F VADER Sentiment Analysis

A. Datos rápidos

Tecnologías	Python.
Versión	Commit 5a6ccd9 (Original version) NTLK v3.4.5 (NTLK version)
Código fuente	https://github.com/cjhutto/vaderSentiment https://www.nltk.org/_modules/nltk/sentiment/vader.html
Licencia	MIT License (MIT)
Desarrollador	C.J. Hutto
Artículo	VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text

B. Introducción

VADER es una librería desarrollada en Python (y portada a otras plataformas) para el análisis de sentimiento en textos. Está basada en reglas y léxicos específicamente diseñados para contextos de microblogs o redes sociales, de manera que contempla el uso de emojis y emoticonos y tiene en cuenta el uso de signos en los textos para el análisis sentimiento. VADER puede utilizarse junto con NLTK para realizar el análisis de textos largos mediante su descomposición, conocida como “tokenización”.

Cabe resaltar que VADER fue incorporado a NTLK [24], sin embargo, la versión existente hoy en día en su versión de NTLK, no incorpora el procesamiento de emojis mientras que la versión original sí lo hace.

C. Características

El motor de análisis de sentimientos basado en reglas implementa reglas gramaticales y sintácticas, incorporando cuantificaciones derivadas del impacto de cada regla en la intensidad percibida del sentimiento en el texto, en base a pruebas empíricas. Además, incorpora relaciones sensibles al orden de las palabras entre los términos, modificadores de grado, palabras de refuerzo o adverbios de grado que afectan a la intensidad del sentimiento.

VADER proporciona un léxico con las siguientes características:

- El léxico de VADER es sensible tanto a la polaridad como a la intensidad de los sentimientos expresados en los textos.
- El léxico incluye:
    - Emoticonos de estilo occidental, por ejemplo, :-).
    - Traducción de emojis codificados en utf-8 como 💋 y 😬.
    - Acrónimos e iniciales relacionados con sentimientos, como LOL o WTF.
    - Expresiones típicas de jerga como nah, meh y risita.

Sistema de puntuación

VADER produce 4 métricas: positivo, negativo, neutral y compuesto. La puntuación compuesta se calcula sumando la puntuación de valencia de cada palabra en el léxico, ajustado de acuerdo con las reglas y luego normalizado para estar entre -1 (más extremo negativo) y +1 (más extremo positivo).

ANEXO G Entregables del proyecto

A continuación, se muestra la estructura de carpetas y el listado de archivos que componen el paquete instalable de la solución propuesta en este TFM. Una vez realizada la instalación, siguiendo los pasos descritos en la sección 5, la solución se integrará en la estructura de la plataforma AIL, como se puede comprobar en la sección 4.1.

El código se encuentra disponible en el siguiente repositorio de GitHub: https://github.com/Sisifo-Stone/TwitterMon

```
LICENSE
LIBRARIES
README.md
installTM.sh
modifyAIL.py
/AIL_bin/
   |__ TwitterAnalyzer.py
   |__ packages/
         |__ Tweet.py

/AIL_var/www/
          |__ templates/
                |__ twittermon/
                            |__ menu_sidebar.html
          |__ modules/
                   |__ TwitterMon/
                            |__ Flask_TwitterMon.py
                            |__ TweetsImporter.py
                            |__ TM_Status.py
                            |__ data/
                            |__ logs/
                            |__ config/
                                  |__ TwitterMon.cfg
                            |__ templates/
                                  |__ header_TwitterMon.html
                                  |__ TwitterMon.html
                                  |__ TwitterMon_results.html
                                  |__ TwitterMon_settings.html

Archivo: Archivo exclusivo de la instalación.
```

ANEXO H Currículum Vitae

En la siguiente página de LinkedIn se puede consultar un resumen de mi Currículum Vitae:

https://www.linkedin.com/in/alfonso-garcía-9a709357

Citas y licencias

Vader Sentiment Analysis

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

The MIT License (MIT)
Copyright (c) 2016 C.J. Hutto
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Ail Framework

Copyright (C) 2014 Jules Debra
Copyright (C) 2014-2019 CIRCL - Computer Incident Response Center Luxembourg (c/o smile, security made in Lëtzebuerg, Groupement d'Intérêt Economique)
Copyright (c) 2014-2019 Raphaël Vinot
Copyright (c) 2014-2019 Alexandre Dulaunoy
Copyright (c) 2016-2019 Sami Mokaddem
Copyright (c) 2018-2019 Thirion Aurélien

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

GetOldTweets-python

The MIT License (MIT)

Copyright (c) 2016 Jefferson Henrique

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Historia del documento

Versión	Fecha	Autor	Descripción
0.1	30/09/2019	Alfonso García Alonso	Primera versión del documento. • Creación de la estructura del documento. • Desarrollo de las siguientes secciones: ○ 1. Introducción ○ 2. Análisis de mercado y estado del arte. ○ 3. Propuesta ○ Anexo A: Glosario. ○ Anexo B: Referencias. ○ Anexo C: Herramientas
0.2	28/10/2019	Alfonso García Alonso	Segunda entrega con las siguientes incorporaciones y modificaciones: • Se detalla la arquitectura de la solución propuesta en la sección 3. Arquitectura y diseño. • Se añade una nueva funcionalidad al proyecto, el análisis de sentimiento de los tweets, viéndose afectadas las secciones 1, 2 y 3, así como el objetivo y abstract del TFM. • Se añaden los anexos F y H. • Se añade la sección de Citas y licencias. • Se añade más detalle al anexo D. • Corrección de errores.
0.3	26/11/2019	Alfonso García Alonso	Tercera entrega con las siguientes incorporaciones y modificaciones: • Se detalla el diseño de la solución en la sección 4. • Se detalla el proceso de instalación en la sección 5. • Se añade el ANEXO G con los entregables del proyecto. • Corrección de errores.

Versión	Fecha	Autor	Descripción
1.0	15/12/2019	Alfonso García Alonso	Última entrega con las siguientes incorporaciones y modificaciones: • Se completa el resumen del trabajo. • Actualización de varias imágenes de la interfaz web por cambios en el código. • Se modifican los apartados de recopilación de Tweets (4.3) y suministro (4.4) debido a modificaciones en el código. • Se modifica la sección 5 de instalación del módulo. • Se añade la sección 7. Tests. • Se añade el capítulo 8 de Conclusiones. • Corrección de errores.