Alternativa Libre para la Minería de Opiniones en Twitter: Integración de Nitter, Puppeteer y VADER

Joshua Nathaniel Arrazola Elizondo[1], Ricardo Emmanuel Uriegas Ibarra[1], Hector Hugo Áviles Arriaga[1]

[1] Universidad Politécnica de Victoria

joshuanarrazola@gmail.com
risardo01@gmail.com
havilesa@upv.edu.mx

Resumen: Se presenta una herramienta para análisis de sentimientos en publicaciones de la red social **X** (antes Twitter), sin depender de su API oficial. La herramienta propuesta utiliza web scraping (extracción web) sobre Nitter para recolectar contenido, filtrar por idioma y aplicar clasificación sentimental utilizando el modelo VADER. Los resultados se visualizan mediante gráficos y agrupaciones automáticas. En pruebas con el término “Liquid Glass”, el sistema muestra precisión aceptable, con un F1-score general superior al 81 %. Esta propuesta representa una alternativa viable, sin costo y replicable para investigadores u organizaciones con barreras económicas al acceso a datos sociales.

Palabras clave: Análisis de sentimientos, Nitter, redes sociales, scraping, VADER

Abstract: A tool is presented for sentiment analysis on posts from the social network X (formerly Twitter), without relying on its official API. The proposed tool uses web scraping (data extraction) through Nitter to collect content, filter by language, and apply sentiment classification using the VADER model. The results are visualized through graphs and automatic groupings. In tests with the term “Liquid Glass,” the system showed acceptable accuracy, with an overall F1-score above 81%. This proposal represents a viable, cost-free, and replicable alternative for researchers or organizations facing economic barriers to accessing social data.

Keywords: Nitter, scraping, sentiment analysis, social media, VADER

INTRODUCCIÓN:

Desde su aparición, **X** (anteriormente **Twitter**) se ha convertido en una fuente clave para monitorear opiniones públicas y detectar tendencias en áreas como política, marketing y gestión de crisis empresariales. Sin embargo, el reciente cambio hacia una *API* pagada ha limitado significativamente el acceso gratuito y masivo a datos [1]. Ejemplo de esto puede contemplarse en el nivel *Enterprise*; el plan más económico brinda acceso a 50 millones de tweets por 42,000 dólares mensuales, siendo los siguientes niveles (100 millones y 200 millones tweets) aún más costosos, alcanzando los 125,000 dólares y 210,000 dólares al mes, respectivamente [2]. En los niveles de uso personal, se encuentran planes como *Basic* (200 USD/mes para 10,000 tweets) y *Pro* (5,000 USD/mes para 1 millón de tweets) [3].

Esta barrera económica, que exige inversiones fuertes únicamente para datos limitados, dificulta el acceso a investigadores, académicos, pequeñas empresas y colectivos de recursos limitados.

El panorama actual obliga a buscar alternativas que permitan el acceso a datos públicos, sin depender exclusivamente de APIs propietarias.

En respuesta, este trabajo propone una alternativa al uso de la API oficial de X. El *scraping* consiste en un proceso automatizado que simula la navegación de un usuario para extraer datos públicos visibles en páginas web; en este caso se utiliza **Nitter** [4], una interfaz de código abierto que presenta contenido público de X en HTML estático, el cual permite recolectar las mismas publicaciones que la API, pero sin necesidad de autenticación ni planes de pagos.

Esta estrategia no incurre en ninguna actividad ilegal. En Estados Unidos, por ejemplo, el fallo **hiQ Labs vs. LinkedIn** [5, 6] determinó que extraer datos accesibles públicamente no viola la ley federal de fraude informático (Computer Fraud and Abuse Act, CFAA). En México sucede algo similar: siempre que se limiten a datos disponibles públicamente (sin acceder mediante contraseñas ni omitir medidas técnicas), el scraping no implica ilegalidad [7].

METODOLOGÍA:

Se diseñó un sistema modular para extraer publicaciones y analizar su contenido sentimental polarizado a 3 clases (positivo, negativo y neutral), sin uso ni dependencia de autenticaciones o claves privadas.

La Figura **1** muestra el flujo lógico del sistema.

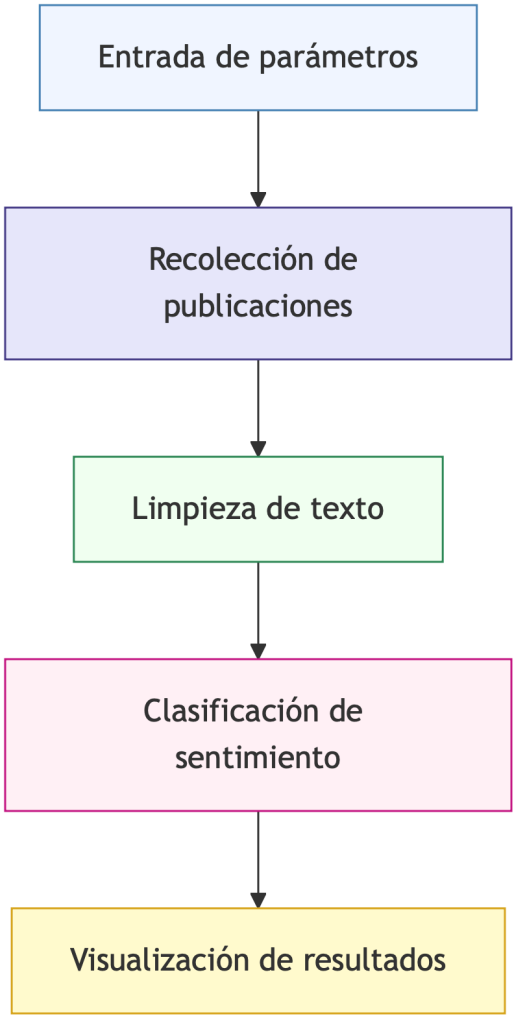


Figura 1: Flujo principal del sistema de análisis de sentimientos.

El sistema propuesto facilita el análisis automatizado de *tweets* públicos en X. El proceso inicia en una interfaz gráfica desarrollada en **PyQt5** [8], que permite al usuario introducir el término de búsqueda, idioma deseado, rango de fechas y la profundidad. Este último parámetro indica cuántas páginas de resultados serán recorridas durante la recolección; es decir, representa el nivel de desplazamiento vertical dentro del entorno de **Nitter**, donde cada página agrupa entre 20 y 35 publicaciones. Por ejemplo, una profundidad de diez implica que el sistema explorará las diez primeras páginas, lo que suele arrojar una muestra cercana a los 300 tweets. Además de ofrecer flexibilidad y escalabilidad al sistema, este parámetro incide directamente en la representatividad temporal de los datos obtenidos: una profundidad baja tiende a capturar publicaciones más recientes, lo que puede ser adecuado en ciertos contextos, pero también podría introducir un sesgo si se busca una visión más amplia o retrospectiva del fenómeno analizado. Por otro lado, una profundidad alta captura una mayor cantidad de datos, pero exige mayor consumo de recursos computacionales de manera lineal, sin embargo, es importante mencionar que no existe un límite teórico de profundidad de búsqueda.

La recolección de publicaciones se lleva a cabo mediante un *scraper* automatizado basado en **Puppeteer** [9], una biblioteca de **Node.js** [10] diseñada por Google para controlar navegadores de forma programática, simular *clics*, *scroll* (deslizar) y navegación como si fuesen realizados por un usuario real. Esta herramienta permite acceder al contenido de Nitter. A través de Puppeteer, el sistema ejecuta búsquedas automatizadas con parámetros definidos (término de búsqueda, idioma, rango temporal y

profundidad), el cual carga dinámicamente las páginas de resultados y extrae el contenido textual sin utilizar la API oficial ni autenticaciones externas.

Una vez obtenidas las publicaciones, el sistema las almacena en formato **JSON** para facilitar su procesamiento. En esta etapa, se aplican dos transformaciones esenciales:

1. **Filtrado por idioma**, por medio de la librería **langdetect** [11], que emplea modelos de probabilidad basados en n-gramas para detectar si el contenido está en inglés o español.
2. **Limpieza de texto**, donde se eliminan menciones, *URLs*, *emojis*, *hashtags* y otros elementos no lingüísticos. Este paso asegura que los textos sean representativos del lenguaje natural y comprensibles para el modelo de análisis.

La clasificación de sentimientos se realiza mediante **VADER** [12, 13] (*Valence Aware Dictionary and sEntiment Reasoner*), un modelo diseñado específicamente para redes sociales, que no requiere entrenamiento previo y es capaz de capturar matices sentimentales complejas. VADER asigna a cada texto un *compound score* (puntaje compuesto) entre -1 y +1, que se traduce en una etiqueta de sentimientos con base en umbrales estándar: positivo si es mayor a 0.05, negativo si es menor a -0.05, y neutral en los casos intermedios [14].

El sistema culmina con una etapa de visualización. A través de gráficos de barras generados con **matplotlib** [15], se presenta la distribución general de los sentimientos identificados. Además, los tweets son organizados en pestañas según su polaridad, lo que permite una revisión más eficiente tanto a nivel exploratorio como para validación manual.

Se analizaron un total de 989 publicaciones con el término "*Liquid Glass*" [16] (el cual fue seleccionado como objeto de análisis debido a su vínculo reciente con el ecosistema *Apple*, lo que garantiza una variedad significativa de opiniones), cada publicación se clasificó automáticamente mediante el modelo VADER y se validó de manera individual por un humano. Este análisis permite comparar el desempeño del modelo frente a una clasificación considerada como "verdadera".

Para evaluar el rendimiento del sistema frente a cada clase de sentimiento, se recurrió al cálculo del *F1-score*, una métrica que combina *precision* y *recall* en una única medida. Esta elección se debe a que el conjunto de datos presenta un desequilibrio entre clases, lo cual puede sesgar otras métricas como el *accuracy* global. En contextos como

este, donde ciertas clases predominan sobre otras, el *F1-score* permite valorar el desempeño del modelo sin favorecer a la clase mayoritaria.

Las componentes del *F1-score*, *precision* y *recall*, se calcularon de forma individual para cada clase. La *precision* indica la proporción de verdaderos positivos entre todas las predicciones realizadas para una clase, mientras que el *recall* representa la proporción de verdaderos positivos recuperados entre todos los ejemplos reales de esa clase. A partir de estos valores, se derivó el *F1-score*, que penaliza tanto los falsos positivos como los falsos negativos, proporcionando así una evaluación más equilibrada.

Todos los conteos necesarios para este análisis se extrajeron directamente de la matriz de confusión, construida al comparar las predicciones del sistema con las etiquetas asignadas manualmente por un humano.

RESULTADOS:

Se obtuvo la matriz de confusión que se presenta en la Figura **2**, la cual muestra el desempeño del sistema en la detección de sentimientos positivos, negativos y neutros.

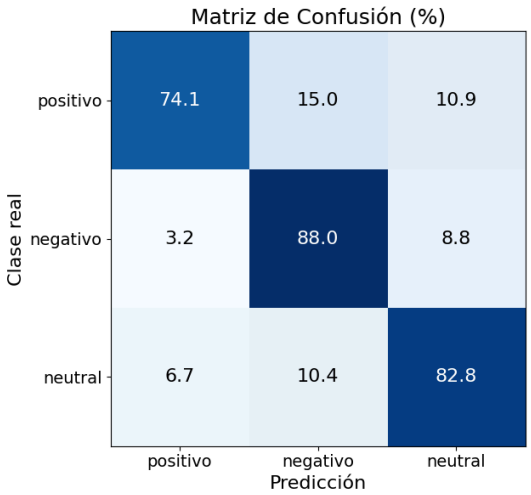


Figura 2. Matriz de confusión porcentual para las tres clases de sentimientos (positivo, negativo y neutral).

A partir de esta distribución, se calculan las métricas de **recall**, **precision** y **F1-score**, definidas a partir de los conteos de verdaderos positivos (**TP**), falsos positivos (**FP**) y falsos negativos (**FN**). Las expresiones generales para cada métrica se presentan en las ecuaciones **1**, **2** y **3**:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3)$$

Las Ecuaciones **4, 5, 6** corresponden al cálculo de *precision* para cada clase:

$$Precision_{positivo} = \frac{143}{143 + 16 + 20} = 0.798 \quad (4)$$

$$Precision_{negativa} = \frac{439}{439 + 29 + 31} = 0.879 \quad (5)$$

$$Precision_{neutral} = \frac{246}{246 + 21 + 44} = 0.790 \quad (6)$$

Las Ecuaciones **7**, **8**, **9** definen el *recall* por clase:

$$Recall_{positivo} = \frac{143}{(143 + 50)} = 0.740 \quad (7)$$

$$Recall_{negativo} = \frac{439}{(439 + 60)} = 0.879 \quad (8)$$

$$Recall_{neutral} = \frac{246}{(246 + 51)} = 0.828 \quad (9)$$

A partir de los valores obtenidos anteriormente, se procede al cálculo del *F1-score*, como se puede apreciar en las ecuaciones **10**, **11** y **12**:

$$F1_{positivo} = 2 \cdot \frac{(0.798)(0.740)}{0.798 + 0.740} = 0.768 \quad (10)$$

$$F1_{negativo} = 2 \cdot \frac{(0.879)(0.879)}{0.879 + 0.879} = 0.879 \quad (11)$$

$$F1_{neutral} = 2 \cdot \frac{(0.790)(0.828)}{0.790 + 0.828} = 0.808 \quad (12)$$

En la clase positiva, se registra un *F1-score* de **0.7688**, resultado de una *precision* de 0.7989 y un *recall* de 0.7409. Este valor indica que el sistema logra identificar correctamente una cantidad considerable de publicaciones con sentimiento positivo, aunque aún existe un margen de error, principalmente por omisión de casos reales.

Para la clase negativa, el *F1-score* alcanzó **0.8798** (el más alto entre las tres categorías). Tanto la *precision* como el *recall* en esta clase son simétricos (0.8798), resultados que evidencian cómo el sistema identifica la mayoría de los casos negativos y rara vez los confunde con otras clases.

En cuanto a la clase neutral, se obtuvo un *F1-score* de **0.8092**, con una *precision* de 0.7910 y un *recall* de 0.8283. Esto sugiere un equilibrio sólido, donde se es capaz de recuperar adecuadamente las publicaciones neutras sin incurrir en un volumen elevado de falsos positivos.

La Figura **3** presenta una comparación visual de los valores de *F1-score* por clase.

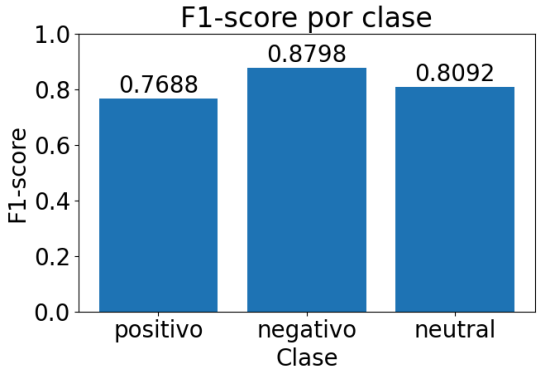
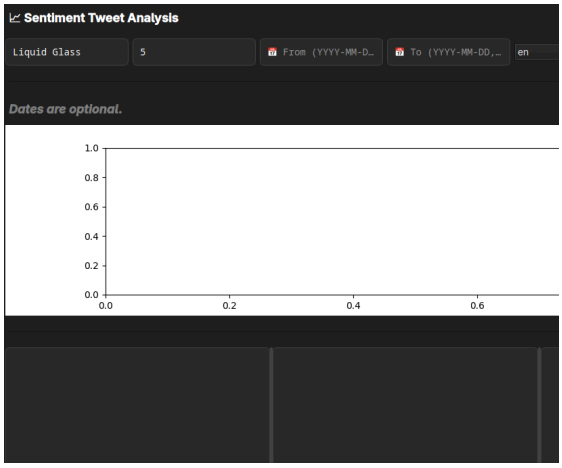


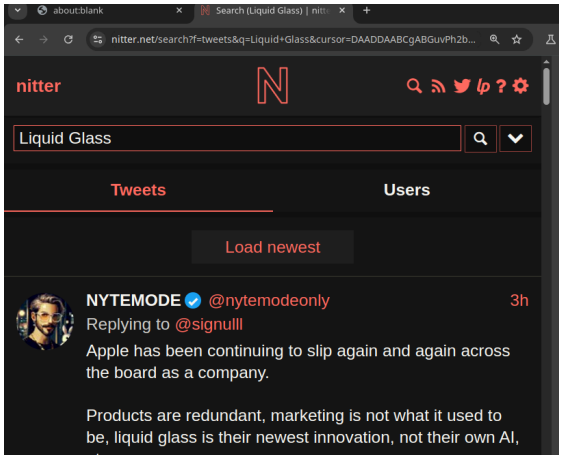
Figura 3: *F1-Score* obtenido por el modelo en cada una de las tres clases: positiva, negativa y neutral.

Durante la ejecución del sistema, la interfaz responde de manera fluida y sin errores notables. En las Figuras **4a** y **4b** se muestra el momento en que el usuario define los parámetros de búsqueda y comienza el proceso de extracción sobre el parámetro de búsqueda *Liquid Glass*.

Una vez finaliza el análisis, como se aprecia en la Figura **5**, la interfaz presenta los resultados mediante un gráfico de barras; que resume la distribución de sentimientos identificados en las publicaciones recolectadas. Los tweets se organizan automáticamente en pestañas según su polaridad.



(a) Parámetros de búsqueda en la interfaz



(b) Proceso de extracción sobre el parámetro de búsqueda Liquid Glass

Figura 4a y 4b: Interfaz del sistema durante la ejecución del *scraping* sobre el tema *Liquid Glass*.

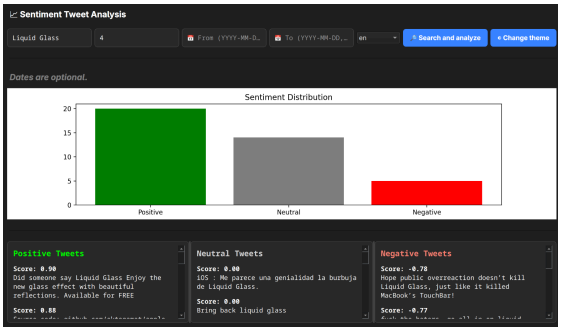


Figura 5: Vista de resultados con gráficas de distribución y tweets clasificados por sentimiento.

Se observa que el sistema logra una velocidad promedio de scraping de aproximadamente 30 publicaciones por minuto, con un comportamiento estable incluso en profundidades de búsqueda superiores a treinta páginas.

CONCLUSIÓN:

El desarrollo de esta herramienta representa un avance significativo en el acceso libre y automatizado al análisis de sentimientos en redes sociales, particularmente considerando las restricciones impuestas por plataformas como **X**. Al evitar la dependencia de su API oficial, se presenta una alternativa accesible y funcional para investigadores, periodistas o instituciones que requieren analizar el discurso público digital sin costos asociados ni limitaciones técnicas estrictas.

Más allá de los buenos resultados alcanzados en pruebas controladas, el sistema demuestra ser replicable, adaptable a otros términos de búsqueda, y útil para explorar tendencias o estados de opinión. Su diseño modular permite futuras mejoras; como la integración de modelos más sofisticados o expansión a soporte de distintos idiomas. En conjunto, esta propuesta evidencia el potencial del *scraping* y la inteligencia artificial ligera como herramientas para el estudio social contemporáneo.

REFERENCIAS BIBLIOGRÁFICAS:

[1] **Stokel-Walker, C**. (2023, 10 de marzo). Twitter's $42,000-per-Month API Prices Out Nearly Everyone. WIRED. Recuperado de https://www.wired.com/story/twitter-data-api-prices-out-nearly-everyone/

[2] **Safe Software.** (2024, 28 de marzo). X (Twitter) Ends Free Access to API: FME Implications. FME Knowledge Center. Recuperado de https://support.safe.com/hc/en-us/articles/27530052447885-X-Twitter-Ends-Free-Access-to-API-FME-Implications

[3] **Data365**. (2023). Twitter API limitations and pricing. Recuperado de https://data365.co/guides/twitter-api-limitations-and-pricing

[4] **Zedeus**. (s.f.). Nitter search. Recuperado de https://nitter.net/search

[5] **International Association of Privacy Professionals**. (2022, April 6). Data scraping and the implications

of the latest LinkedIn-HiQ court ruling. IAPP. https://iapp.org/news/a/data-scraping-and-the-implications-of-the-latest-linkedin-hiq-court-ruling/

[6] **PromptCloud**. (2023, October 20). Is web scraping legal in the US? A complete guide. https://www.promptcloud.com/blog/is-web-scraping-legal-in-us-a-complete-guide/

[7] **Guzmán González, J. G.** (2023, 31 de marzo). "Web Scraping" como método de obtención de contenido para la práctica legal. LegalTec Lab. Recuperado el 30 de junio de 2025, de https://legalteclab.mx/%C2%A8web-scraping%C2%A8-como-metodo-de-obtencion-de-contenido-para-la-practica-legal/

[8] **Riverbank Computing Limited. (s. f.)**. PyQt5 official documentation. Recuperado el 27 de junio de 2025, de https://www.riverbankcomputing.com/static/Docs/PyQt5/

[9] **Google, Inc. (s. f.).** *Puppeteer* [Documentación oficial]. Recuperado el 27 de junio de 2025, de https://pptr.dev/

[10] **Node.js.** (2025). *Node.js: JavaScript runtime built on V8 engine*. OpenJS Foundation. Recuperado de https://nodejs.org/es

[11] I**BM Corporation**. (25 de julio de 2024). Lang-detect – Watson Natural Language Processing Library [Documentación en línea]. Recuperado el 27 de junio de 2025, de https://www.ibm.com/docs/en/watson-libraries?topic=catalog-lang-detect

[12] **Hutto, C. J.** (2021). VaderSentiment documentation. Recuperado el 27 de junio de 2025, de https://vadersentiment.readthedocs.io/en/latest/

[13] **Hutto, C. J**. (s.f.). vaderSentiment (Valence Aware Dictionary and sEntiment Reasoner). GitHub. Recuperado el 30 de junio de 2025, de https://github.com/cjhutto/vaderSentiment

[14] **BobLd.** (2022, febrero 2). VaderSharp2 (versión 3.3.2.1). NuGet. Recuperado de https://www.nuget.org/packages/VaderSharp2

[15] **The Matplotlib development team**. (2025). Matplotlib — Visualization with Python [Documentación en línea]. Recuperado el 27 de junio de 2025, de https://matplotlib.org/stable/index.html

[16] **Apple Inc.** (2025, 9 de junio). Apple introduces a delightful and elegant new software design [Nota de prensa]. Recuperado el 27 de junio de 2025, de https://www.apple.com/mx/newsroom/

2025/06/apple-introduces-a-delightful-and-elegant-new-software-design/