



UNIVERSIDAD SIMÓN BOLÍVAR

Departamento de Computación y Tecnología de la Información

EP4793 - Mini Proyecto de Desarrollo de Software

Abril - Julio 2023

Mini Proyecto T-CREo

Tutores:

Yudith Cardinale
Irvin Dongo
Ana Aguilera

Autores:

Jonathan Bautista (16-10109)
Daniela Ramírez (16-10940)
Valeria Vera (16-11233)
Ka Man Fung (18-10492)

Sartenejas, Julio 2023



Índice general

1. Introducción	3
2. Marco Conceptual	4
2.1 Descripción	4
2.2 Trabajos previos	5
2.2.1 T-CREo: A Twitter Credibility Analysis Framework	5
2.2.2 Web Scraping versus Twitter API: A Comparison for a Credibility,	6
2.2.3 Credibility Analysis on Twitter Considering Topic Detection	6
2.2.4 CrediBot: Applying Bot Detection for Credibility Analysis on Twitter	7
3. Definición del Problema	8
3.1 Formulación del problema	8
3.2 Objetivos	8
3.2.1 General	8
3.2.2 Específicos	8
4. Diseño Arquitectónico	9
Anteriormente, la extensión de Chrome T-CREo v1.0, poseía la siguiente estructura:	9
5. Implementación	10
5.1 Extensión T-CREo	10
5.1.1 Herramientas utilizadas	10
5.1.2 Implementación	11
5.2 Twitter Clone	12
5.2.1 Herramientas utilizadas	12
5.2.2 Implementación	13
6. Pruebas	15
6.1 Pruebas de limpieza de texto	15
6.2 Pruebas de credibilidad de texto	16
6.3 Pruebas de manejo de errores	17
6.4 Prueba de validación de parámetros	17
6.5 Prueba de Estado del Back-end	19
6.6 Tiempos de Ejecución	19
7. Conclusiones y trabajos futuros	20
8. Referencias	21



1. Introducción

En el mundo actual, las redes sociales tienen un gran impacto en la sociedad. Contenido educativo, publicitario, de entretenimiento, de concientización e informativo es compartido día a día a través de ellas. En especial, en la red social Twitter que actualmente cuenta con unos 354 millones de cuentas creadas (Iqbal) y aproximadamente 500 publicaciones diarias de *tweets* (Internet Live Stats). Según estadísticas proveídas por el equipo de noticias de Twitter, el 55% de los usuarios obtienen sus noticias diarias en la plataforma, y a su vez, el 50% de los usuarios dicen que en el 2022 siguieron al menos a un periodista en redes sociales.

Sin embargo, ¿cómo se puede verificar que la información leída es verdadera? Twitter, al igual que todas las redes sociales, no está exenta de información falsa y rumores. De hecho, los *tweets* que contienen noticias falsas son 70% más propensos a ser compartidos que aquellos con información verificada (Langin). En un estudio realizado por Soroush Vosoughi et al. para la revista Science donde se investigó la difusión de noticias falsas y verificadas compartidas en Twitter entre 2016 y 2017, se observó que el top 1% de noticias falsas había sido compartido entre 1000 y 100.000 personas, mientras que la verdad apenas llegó a ser difundida entre 1000 personas.

Además de esto, con los cambios surgidos en Twitter en el último año, en especial la eliminación del *checkmark* para las cuentas verificadas y la creación de un sistema de mensuralidad para poder obtener uno siendo un usuario regular, se torna más difícil para los usuarios de la plataforma saber si la cuenta que está compartiendo información es certera o no.

Ante esta problemática, surge la necesidad de una herramienta que ayude al usuario a verificar la credibilidad de los *tweets*, y es por ello que se crea T-CREo (Twitter **CRE**ditability analysis framework), un *framework* (extensión de Google Chrome) capaz de analizar la credibilidad de los *tweets* de manera automática y en tiempo real.

2. Marco Conceptual

2.1 Descripción

La credibilidad se refiere al grado de confianza o veracidad que se percibe en un objeto, evento o persona. En las redes sociales, donde no hay una clasificación definida de información, por lo que determinar la credibilidad puede ser un desafío debido a la gran cantidad de información generada diariamente. Dado que esta información es creada por personas y a menudo no tiene una referencia para verificarla, puede generar incertidumbre sobre su confiabilidad y calidad, lo que aumenta la necesidad de desarrollar herramientas para validar y clasificar la información (Dongo et al., 2019). Por ello, T-CREo tiene como objetivo poder medir la credibilidad de un *tweet*, dado un modelo de credibilidad que dividido principalmente en cuatro módulos:

- **Credibilidad de texto:** mide el nivel de relevancia y precisión del texto, independientemente del tema referenciado o con respecto a un tema determinado.
 - Filtro de spam
 - Filtro de ortografía
 - Filtro de malas palabras
 - Filtro de semántica
- **Credibilidad de usuario:** calcula la credibilidad de la cuenta de usuario en función de los atributos que la describen.
 - Verificación de usuario a través del sistema de *checkmark* de Twitter
 - Año de creación de la cuenta
 - Detección de bots
- **Credibilidad social:** calcula la credibilidad de una publicación, relacionada o no con un tema, en base a los metadatos disponibles que describen el impacto social de la cuenta de usuario y la publicación en sí, con respecto a otros usuarios.
 - Impacto según el número de seguidores
 - Proporción entre cantidad de seguidores y cantidad de cuentas seguidas
- **Credibilidad de tópicos:** mide el nivel de similitud de tópicos entre el texto del tweet y los hashtags utilizados en él.
 - Filtro de hashtags

2.2 Trabajos previos

2.2.1 T-CREo: A Twitter Credibility Analysis Framework

Robayo et al. (2020) proponen un modelo de credibilidad de Twitter basado en credibilidad de texto, de usuario y social, incorporado en una extensión de Chrome. Para estos cálculos de credibilidad se tomaron los siguientes parámetros y formulación:

1. Credibilidad de texto ($O(n)$ con n cantidad de palabras):

$$\begin{aligned} \text{TextCred}(p.\text{text}) = & w\text{SPAM} \times \text{isSpam}(p.\text{text}) + \\ & w\text{BadWords} \times \text{bad_words}(p.\text{text}) + \\ & w\text{MisspelledWords} \times \text{misspelling}(p.\text{text}) \end{aligned}$$

donde $w\text{SPAM}$, $w\text{BadWords}$, $w\text{MisspelledWords}$ determina su importancia relativa en el cálculo de la credibilidad del texto. Mientras que $\text{isSpam}(p.\text{text})$ devuelve un valor que indica si el texto es spam o no, $\text{bad_words}(p.\text{text})$ representa la cantidad de malas palabras presentes en el texto y $\text{misspelling}(p.\text{text})$ devuelve un valor que representa la cantidad de palabras mal escritas presentes en el texto.

2. Credibilidad de usuario ($O(1)$):

$$\text{UserCred}(p.\text{user}) = \text{Verif_Weight}(p.\text{user}) + \text{Creation_Weight}(p.\text{user})$$

donde $\text{Verif_Weight}(p.\text{user})$ representa el impacto de la verificación del usuario en su credibilidad y $\text{Creation_Weight}(p.\text{user})$ representa el impacto de la fecha de creación de la cuenta del usuario en su credibilidad.

3. Credibilidad social ($O(1)$):

$$\text{SocialCred}(p.\text{user}) = \text{FollowersImpact}(p.\text{user}) + \text{FFProportion}(p.\text{user})$$

donde $\text{FollowersImpact}(p.\text{user})$ representa el impacto del número de seguidores del usuario en su credibilidad social y $\text{FFProportion}(p.\text{user})$ representa el impacto de la proporción entre seguidores y seguidos del usuario en su credibilidad social.

Dicho módulo fue implementado en una extensión de Chrome compuesta por una vista principal *popup* que permite al usuario calcular la credibilidad con Twitter API v1.0 o web scraping, según la vista del sitio en donde el usuario se encuentre, y la vista de opciones que permite configurar los parámetros de análisis.

2.2.2 Web Scraping versus Twitter API: A Comparison for a Credibility,

Martínez et al. (2020) realiza una comparación del uso de *web scraping* y con Twitter API v1.0 para la recopilación de data de *tweets* para analizar su credibilidad. Así, se tiene:

Mayor rapidez en comparación con el Twitter API v1.0	Información detallada y directa
	No necesita buscar cada campo del tweet por separado sino simplemente realizar una llamada a la API
Se debe buscar cada campo del tweet por separado.	Requiere del ID del <i>tweet</i> y del nombre de usuario
Su programación es específica a cada sitio web	Hay un límite de volumen y tasa de solicitud
Requiere constante mantenimiento ya que es sensible a los cambios del formato de la página web.	

Tabla 1: Comparación entre WebScraping y Twitter API

2.2.3 Credibility Analysis on Twitter Considering Topic Detection

Hernández et al. (2021) proponen una extensión del modelo de credibilidad anterior al incluir la detección de tópicos basada en la evaluación de *hashtags*. Así, se pasa de tres ramas del modelo de credibilidad donde cada una tiene un peso de 33.33%, a cuatro ramas con 25% de peso para un total de 100%.

Para ello, se realiza un preprocesamiento de la data, consistiendo esta en la eliminación de menciones, caracteres especiales, emojis, enlaces, números, y palabras repetidas. Posteriormente, se dividen el *tweet* y los *hashtags* presentes en él para determinar el tópico del cual están hablando por medio del algoritmo de Factorización de Matriz no Negativa (NFM), lo cual se realiza a través de una librería de Python.

Luego, se hace uso de la distancia Hellinger (HDS) para determinar qué tan coherente es un *tweet* con respecto a los *hashtags* que posee. Finalmente, se propone una función donde se realiza este proceso que retorna un valor entre 0 y 100 el cual representa la medida de credibilidad del tópico:

$$TopicCred(p.text) = 100 \times \left(1 - \frac{1}{n} \sum_{i=1}^n HDS(NMF(p.text), NMF(p.text.hashtag_i))\right)$$

Donde n es el número de *hashtags* presente.



2.2.4 CrediBot: Applying Bot Detection for Credibility Analysis on Twitter

Quinteros et al. (2022) propone un modelo para la detección de bots de usuarios de Twitter. Para ello, dado un tweet de un usuario en particular, se extraen la siguiente información:

- Del usuario: cantidad de seguidores, de cuentas seguidas, de estados, de *tweets* favoritos y de listas creadas.
- Del *tweet*: texto, lenguaje, cantidad de personas que lo tienen en sus favoritos y cantidad de *retweets*.

Luego, se identifica si el usuario es un *bot* con un modelo de *Random Forest*, entrenado con un dataset de 500.000 cuentas. Si el usuario trata de un *bot*, se penaliza su puntuación obtenido en la credibilidad del usuario:

UserCredBot(p.user) =

$$\begin{cases} \text{UserCred(p.user)} & \text{Si p.user no es bot,} \\ 0.85 \times \text{UserCred(p.user)} & \text{Si p.user es un bot y } \text{UserCred(p.user)} > 50, \\ 0.75 \times \text{UserCred(p.user)} & \text{Si p.user es un bot y } 35 < \text{UserCred(p.user)} < 50 \\ 0 & \text{En cualquier otro caso } \end{cases}$$

Donde UserCred(p.user) es el valor original del modelo de credibilidad T-CREo.

Por último, se realiza un análisis semántico para detectar qué tan relacionadas están las palabras utilizadas en un tweet. Para ello, se realiza un proceso de tokenización, revisión léxica y semántica, para posteriormente calcular el promedio de similitud.



3. Definición del Problema

3.1 Formulación del problema

La extensión de Google Chrome T-CREo, inicialmente desarrollada como una prueba de concepto de un proyecto de cálculo de credibilidad para fuentes en internet instanciado para Twitter, realiza un análisis automático de credibilidad utilizando técnicas de web scraping y los componentes API REST y API Twitter para lograr realizar este análisis en tiempo real. Durante los últimos años, se han implementado nuevas funcionalidades que mejorarían el modelo existente, como la detección de bots y la credibilidad de tópicos.

Asimismo, el empleo de scraping no resulta mantenible dado que varía según las actualizaciones de la aplicación Twitter. Por otro lado, desde Enero 2023, Chrome Extension ha modificado el empleo y creación de extensiones, al cambiar de Manifest V2 a Manifest V3. Como también, desde Abril 2023, Twitter API ha cambiado de versión de v1.1 a v2.0, como también su uso gratuito a una suscripción mensual.

En este trabajo se pretende integrar los distintos modelos existentes para el cálculo de la credibilidad de Twitter, como también la refactorización y actualización según los requerimientos necesarios en cada módulo de credibilidad.

3.2 Objetivos

3.2.1 General

El proyecto tiene como objetivo general refactorizar y brindar modularidad a la aplicación T-CREo. Además, las funcionalidades de detección de tópicos, detección de bots y el uso de herramientas de almacenamiento (mongoDB) deben ser integradas a la aplicación.

3.2.2 Específicos

- I. Eliminar métodos de acceso de información de Twitter secundarios y utilizar solamente la API.
- II. Refactorizar el código y brindar modularidad a la aplicación a través de archivos de configuración donde los módulos puedan ser activados a voluntad del usuario.
- III. Incorporar a T-CREo el módulo de detección de tópicos.
- IV. Incorporar a T-CREo el módulo de detección de bots.
- V. Incorporar a T-CREo el módulo de almacenamiento de información periódica.
- VI. Actualizar el Plugin de Google Chrome con las funcionalidades de T-CREo.

4. Diseño Arquitectónico

Anteriormente, la extensión de Chrome T-CREo v1.0, poseía la siguiente estructura:

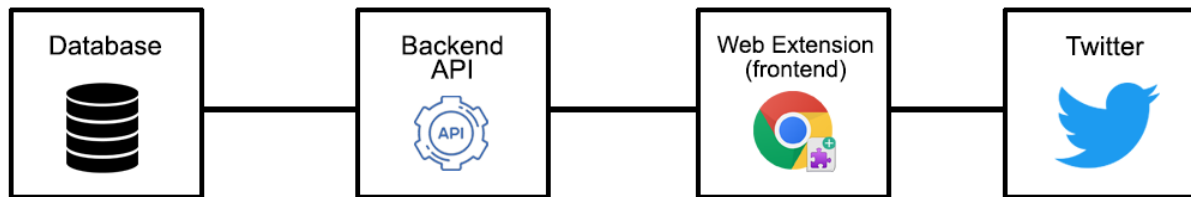


Figura 1: Arquitectura de T-CREo v1.0

Sin embargo, dado las limitaciones que han surgido para el uso de la Twitter API, se decidió la implementación de una interfaz que simula Twitter, con datos recolectados anteriormente de la Twitter API:

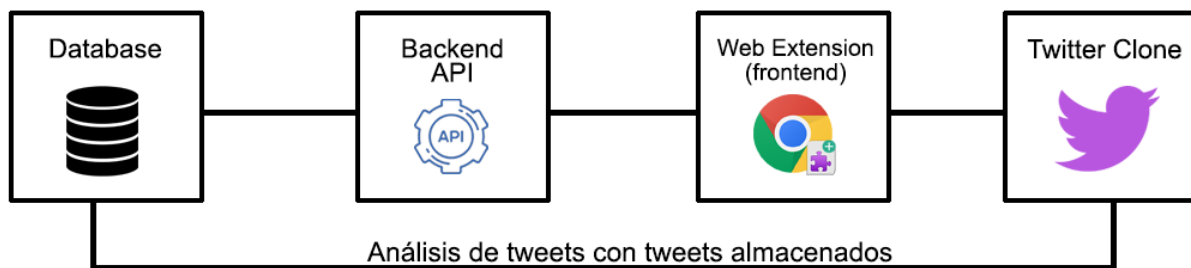


Figura 2: Arquitectura propuesta para T-CREo v2.0

De esta manera, se realiza un análisis de tweets almacenados en una base de datos. La implementación de este procedimiento se explicará a profundidad en la siguiente sección.

A su vez, se hizo una modificación del esquema de credibilidad siguiendo la propuesta realizada por Hernández et al. de dividir el modelo de credibilidad en cuatro ramas, siendo esta cuarta la credibilidad según el análisis de tópicos para la cual se utilizó la función TopicCred descrita previamente. También se implementó la función UserCrediBot descrita por Quinteros et al para determinar la probabilidad de que un usuario sea un humano o un *bot*. Dado que ambos trabajos mencionados en el párrafo anterior utilizaron Python, se hace uso de un *virtual environment* para instalar las dependencias necesarias para hacer uso de las funciones y las librerías que ambas necesitan.

Es importante también señalar que, dado que la base de datos con la que se trabaja maneja información de tweets con el sistema de *checkmarks* previo a la actualización de este año, se siguen tomando en cuenta para la credibilidad de usuario.

5. Implementación

Desde que se realizó T-CREo v1.0, han surgido distintos cambios en las herramientas que utiliza; el más importante de ellos ha sido el Twitter API v1.0, que al momento de iniciar este proyecto pasó a ser reemplazado por el Twitter API v2.0, el cual posee una versión gratuita que sólo permite analizar 1.500 tweets mensualmente de un único ID y una versión con suscripción paga con mayor cantidad de opciones pero que por los momentos no se puede costear para esta investigación. Asimismo, el empleo de *web scraping* no resulta mantenible ya que en el último año Twitter ha tenido múltiples actualizaciones, lo cual hace que la extensión resulte obsoleta en muy poco tiempo. Por último, Chrome Extension modificó el empleo y creación de extensiones, ya que pasó de usar Manifest V2 a Manifest V3.

Por todas estas razones mencionadas, así como el uso de *frameworks* y librerías obsoletas, se decidió iniciar el proyecto desde cero, dando paso a T-CREo V2.0. Asimismo, se realizó un estudio de las herramientas empleadas anteriormente y cuáles emplear para los distintos segmentos de esta versión.

5.1 Extensión T-CREo

5.1.1 Herramientas utilizadas

Para el front-end de la extensión, se realizó un cambio en el lenguaje de programación la cual fue implementada la extensión, al emplear TypeScript como sustituto de JavaScript. Dado que trata de un lenguaje tipado, permite una mayor escalabilidad del código, al reducir el número de errores y mantener la sencillez en el código. Asimismo, se decidió utilizar la framework React JS, para estructurar de manera organizada cada módulo en componentes.

	Anterior	Actual	Motivo
Lenguaje	JavaScript	TypeScript	Lenguaje Tipado, más escalable
Frameworks	-	React JS	Implementación con mayor rapidez y organizado en componentes.
Chrome	V2	V3	V2 se encuentra obsoleto

Tabla 2: Herramientas empleadas para el Front-end de T-CREo

Para el back-end de la extensión, se refactorizó el código con el mismo lenguaje de programación y framework empleados en la primera versión de T-CREo. Se integró con MongoDB para simular las consultas a la Twitter API.



	Anterior	Actual	Motivo
Lenguaje	TypeScript	TypeScript	Se mantuvo el uso del lenguaje de programación.
Frameworks	Express	Express	Se mantuvo el uso de Express por su flexibilidad con MongoDB.
Base de Datos	-	MongoDB	Complemento adecuado para ser usado con Express

Tabla 3: Herramientas empleadas para el Back-end de T-CREo

5.1.2 Implementación

- Se reestructuró el front-end del trabajo previo para que sea adecuado con el framework de React.
- Se crearon las vistas de:
 - *Popup*: Vista principal de la extensión.
 - *Options*: Vista para la configuración de parámetros.
- Se conectó con Chrome Extension con Manifest v3, lo cual requirió actualizar la configuración y reestructurar el código.
- Se conectó el *front-end* con el *back-end*.

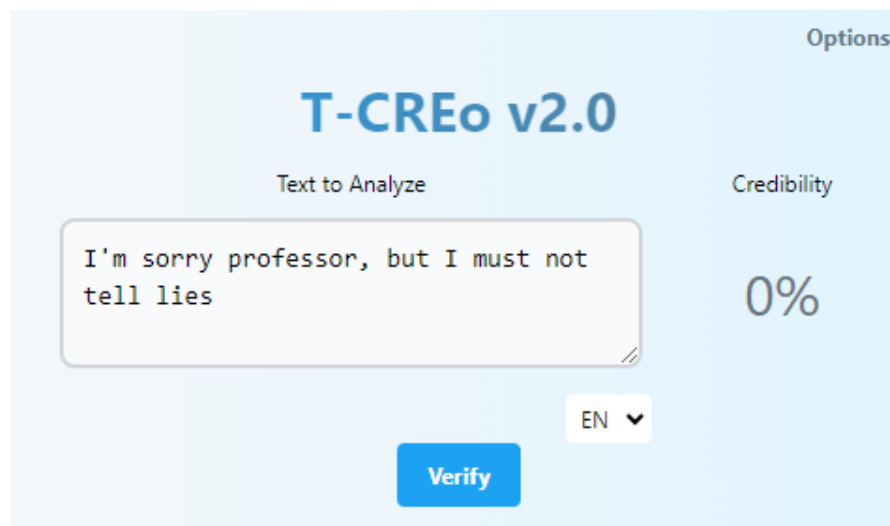


Figura 3: *Popup* de T-CREo al estar en una página web distinta a Twitter

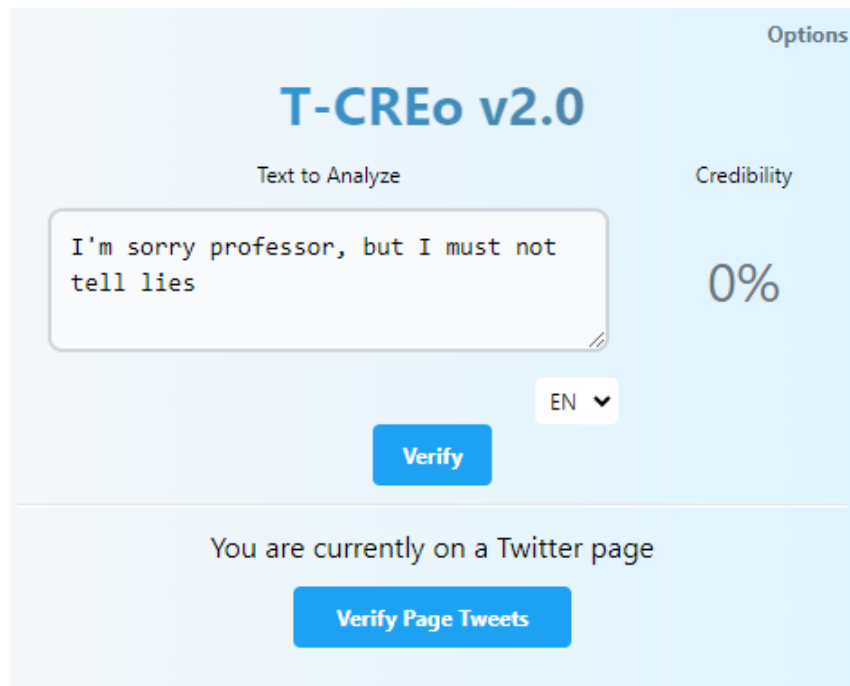


Figura 4: *Popup* de T-CREo al estar en Twitter.com o en Twitter Clone

T-CREo - Credibility parameters

How to use

Text parameters

Spam detection	0.44
Bad words proportion to text	0.33
Misspelling detection	0.23
Semantic analysis	0

Tweet parameters

Text credibility	0.25
User credibility	0.25
Social credibility	0.25
Topic credibility	0.25

Followers configuration

Maximum followers	2000000
-------------------	---------

Save Weights

Figura 5: Página de configuración de T-CREo

5.2 Twitter Clone

5.2.1 Herramientas utilizadas

Dado que la interfaz del Twitter Clone tiene como funcionalidad mostrar los tweets de la base de datos para simular la página web de Twitter, se utilizó TypeScript en conjunto de React JS para mantener una estructura sencilla del proyecto.



	Anterior	Actual	Motivo
Lenguaje	-	TypeScript	Lenguaje Tipado, más escalable
Frameworks	-	React JS	Implementación con mayor rapidez y organizado en componentes.
Database	-	MongoDB	Complemento adecuado para TypeScript

Tabla 4: Herramientas empleadas para el Twitter Clone

5.2.2 Implementación

Dada la imposibilidad de obtener información de *tweets* por medio de la Twitter API, se utilizó una base de datos de 11.991 *tweets* provista por el profesor Irvin Dongo sobre la cual la extensión trabaja para realizar el análisis de credibilidad.

La información proveniente de esta base de datos pasó a ser compartida en lo que los creadores de este proyecto denominaron *Twitter Clone*, puesto que cuenta con una vista de *timeline* y de perfil de usuario como en la página web Twitter; además de compartir el nombre de las etiquetas con Twitter en caso de querer migrar a esta última posteriormente.

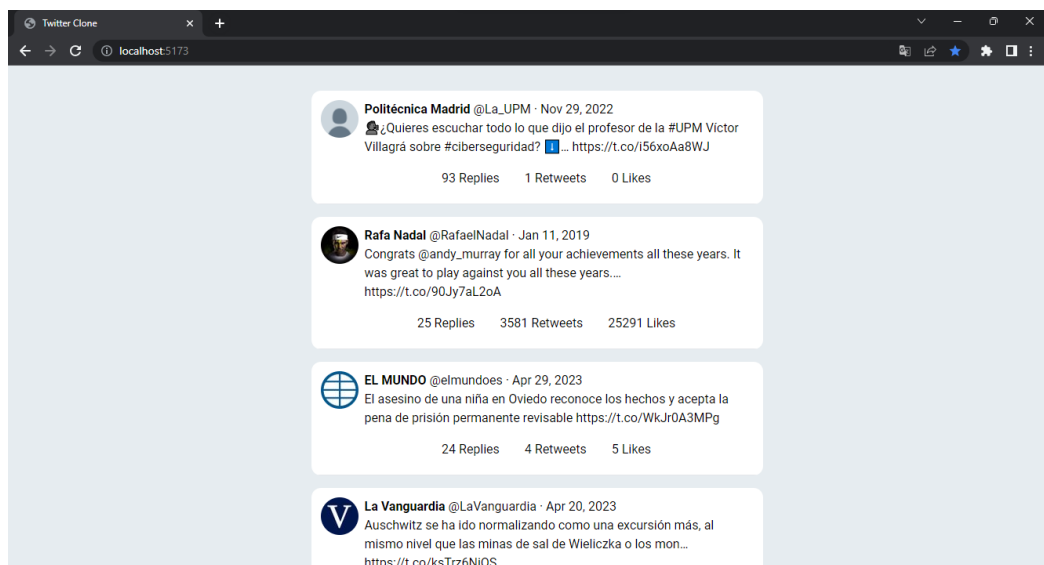


Figura 6: *Timeline* de Twitter Clone

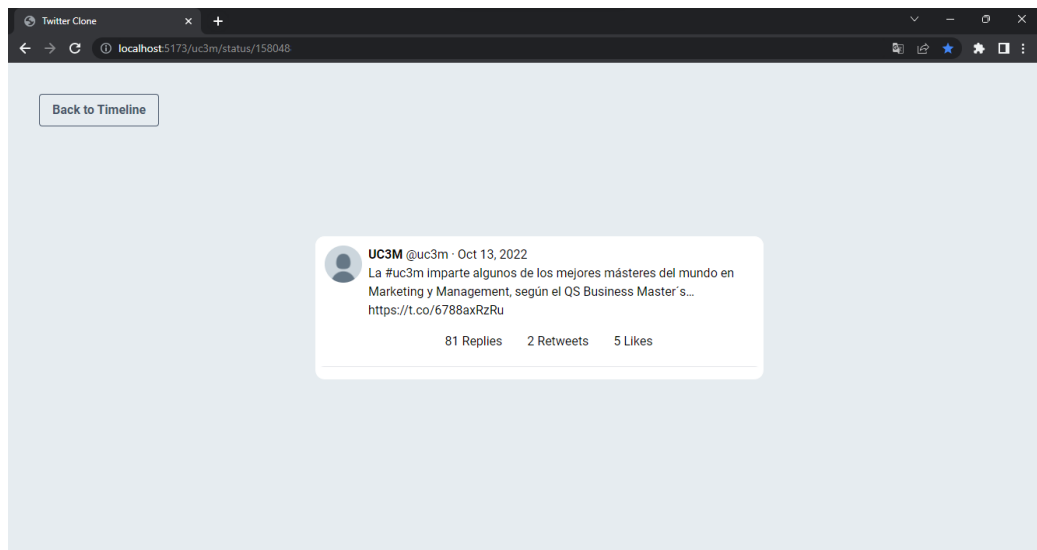


Figura 7: Vista de un *tweet* único

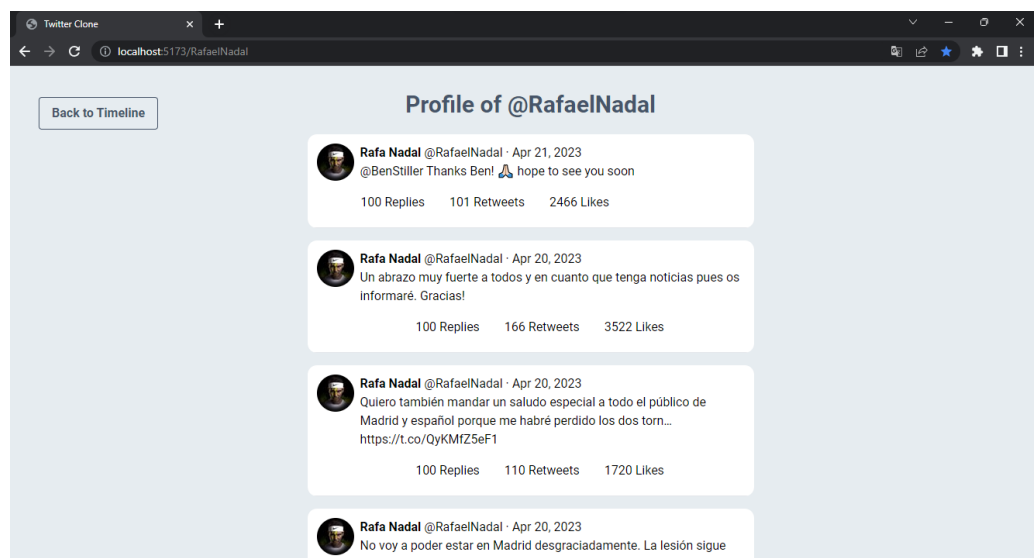


Figura 8: Vista del perfil de un usuario



6. Pruebas

Para realizar las pruebas, se utilizó tanto la base de datos de MongoDB en la nube como la base de datos almacenada localmente para comparar el tiempo de ejecución entre ambas. Los equipos empleados para realizar las pruebas fueron una laptop Asus VivoBook con un procesador Intel i3 10ma generación, 8 Gb de RAM y WSL Ubuntu 22.04.2 en la cual se realizaron las pruebas con la base de datos en la nube; y una laptop Dell Latitude 5510 con un procesador Intel Core i3 de primera generación, 4gb RAM y S.O Windows 10 en la cual se realizaron las pruebas con la base de datos local.

Se crearon cinco suites de pruebas por medio de la librería Jest, así como pruebas del análisis de credibilidad siguiendo el modelo de credibilidad con texto

6.1 Pruebas de limpieza de texto

Se crearon distintos tests para verificar que la limpieza del texto de un tweet se realizara satisfactoriamente con la función CleanTweet(). Para esto, cada uno de los textos con tokens a ser removidos tiene que cumplir con las siguientes pruebas:

- Limpieza del texto cuando sólo posee un token a remover
- Limpieza del texto cuando posee un token a remover al principio
- Limpieza del texto cuando posee múltiples tokens del mismo tipo
- Limpieza del texto cuando está constituido únicamente por tokens del mismo tipo
- Limpieza del texto sin tokens

En todos los casos, la función debe retornar el texto sin la presencia de tokens en él.

Pruebas de remoción de hashtags	Estado	Exec. Nube	Exec. Local	Pruebas de remoción de puntuación	Estado	Exec. Nube	Exec. Local
Texto con sólo un hashtag	Pasó	39 ms	73 ms	Texto con sólo un punto	Pasó	5 ms	4 ms
Texto con sólo un hashtag al inicio	Pasó	3 ms	5 ms	Texto con sólo un punto al inicio	Pasó	17 ms	3 ms
Texto con múltiples hashtags	Pasó	2 ms	3 ms	Texto con punto antes de palabra	Pasó	4 ms	4 ms
Texto únicamente con hashtags	Pasó	23 ms	4 ms	Texto con punto después de palabra	Pasó	4 ms	7 ms
Texto sin hashtags	Pasó	6 ms	19 ms	Texto con punto separado de palabra	Pasó	4 ms	3 ms
Pruebas de remoción de menciones	Estado			Texto únicamente con puntos	Pasó	3 ms	2 ms
Texto con sólo una mención	Pasó	3 ms	3 ms	Texto sin puntos	Pasó	31 ms	2 ms
Texto con sólo una mención al inicio	Pasó	4 ms	3 ms	Pruebas de remoción de espaciado. extra	Estado		
Texto con múltiples	Pasó	3 ms	4 ms	Texto con sólo un espacio	Pasó	21 ms	3 ms



menciones				extra			
Texto únicamente con menciones	Pasó	4 ms	7 ms	Texto con sólo un espacio extra al inicio	Pasó	4 ms	3 ms
Texto sin menciones	Pasó	4 ms	4 ms	Texto con múltiples espacios extra	Pasó	4 ms	3 ms
Pruebas de remoción de emojis	Estado			Texto únicamente con espacios extra	Pasó	4 ms	3 ms
Texto con sólo un emoji	Pasó	23 ms	20 ms	Texto sin espacios extra	Pasó	6 ms	3 ms
Texto con sólo un emoji al inicio	Pasó	4 ms	13 ms	Tiempo total pruebas con BD local			
Texto con múltiples emojis	Pasó	3 ms	4 ms		34,997 s	42,552 s	
Texto únicamente con emojis	Pasó	4 ms	3 ms				
Texto sin emojis	Pasó	4 ms	3 ms				
Pruebas de remoción de enlaces	Estado						
Texto con sólo un enlace	Pasó	15 ms	3 ms				
Texto con sólo un enlace al inicio	Pasó	4 ms	3 ms				
Texto con múltiples enlaces	Pasó	3 ms	3 ms				
Texto únicamente con enlaces	Pasó	4 ms	3 ms				
Texto sin enlaces	Pasó	4 ms	3 ms				

Tabla 5: Tests de limpieza de texto

6.2 Pruebas de credibilidad de texto

Se verifica el buen funcionamiento de los parámetros de análisis de texto, ingresando tokens que disminuyan el nivel de credibilidad:

- Malas palabras
- Texto spam
- Palabras con errores ortográficos

Además, se prueba que el programa sea capaz de calcular correctamente la credibilidad de textos en otros idiomas.

Pruebas de texto con malas palabras	Estado	Exec. Nube	Exec. Local
Texto sin malas palabras	Pasó	6240 ms	186 ms
Texto de 2 palabras con una mala p.	Pasó	55ms	24 ms



Texto de 2 palabras con 2 malas p.	Pasó	33ms	18 ms
Texto con una mala palabra en cualquier idioma	Pasó	43 ms	53 ms
Pruebas de texto con spam	Estado		
Texto sin spam	Pasó	94ms	101 ms
Texto son palabras de spam	Pasó	38ms	36 ms
P. de texto con errores ortográficos	Estado		
Texto sin errores ortográficos	Pasó	46 ms	64 ms
Texto con cada palabra con error ort.	Pasó	42 ms	33 ms
T. con una palabra buena y otra con error ortográfico	Pasó	52 ms	72 ms
Texto con una palabra con error y un núm.	Pasó	105 ms	28 ms
Pruebas de texto variadas	Estado		
Texto con pruebas variadas	Pasó	49 ms	40 ms
Texto con pruebas variadas en Español	Pasó	63 ms	24 ms
Texto con pruebas variadas en Francés	Pasó	82 ms	58 ms
Tiempo Total Pruebas		366,698 s	21,39 s

Tabla 6: Tests de limpieza de texto

6.3 Pruebas de manejo de errores

Retornar código de error 400	Pasó	20 ms	4 ms
Retornar código de error 401	Pasó	31 ms	4 ms
Retornar código de error 403	Pasó	31 ms	6 ms
Retornar código de error 404	Pasó	31 ms	4 ms
Retornar código de error 500	Pasó	33 ms	4 ms
Tiempo Total Pruebas		11,481	0.022 s

Tabla 7: Tests de manejo de errores



6.4 Prueba de validación de parámetros

Prueba de validación en texto simple	Estado	Exec. Nube	Exec. Local
text.REQUIRED	Pasó	6241 ms	130 ms
text.NOTEMPTY	Pasó	150 ms	14 ms
weightSpam.REQUIRED	Pasó	30 ms	12 ms
weightSpam.NUMBER	Pasó	32 ms	20 ms
weightSpam.NOT_IN_RANGE	Pasó	43 ms	11 ms
<i>weightBadWords.REQUIRED</i>	Pasó	46 ms	10 ms
weightBadWords.NUMBER	Pasó	47 ms	7 ms
weightBadWords.NOT_IN_RANGE	Pasó	48 ms	9 ms
weightMisspelling.REQUIRED	Pasó	36 ms	13 ms
weightMisspelling.NUMBER	Pasó	149 ms	11 ms
weightMisspelling.NOT_IN_RANGE	Pasó	34 ms	8 ms
customValidation.WEIGHT_TEXT_CRED_SUM_NOT_1	Pasó	42 ms	16 ms
Prueba de validación en tweets			
Muestra error si no se provee tweetID	Pasó	14883 ms	10107 ms
weightBadWords.NOT_IN_RANGE	Pasó	73 ms	31 ms
weightMisspelling.NOT_IN_RANGE	Pasó	122 ms	8 ms
weightSpam.NOT_IN_RANGE	Pasó	49 ms	8 ms
weightSemantic.NOT_IN_RANGE	Pasó	74 ms	10 ms
maxFollowers.POSITIVE	Pasó	98 ms	8 ms
weightUser.NOT_IN_RANGE	Pasó	40 ms	7 ms
weightTopic.NOT_IN_RANGE (14 ms)	Pasó	98 ms	7 ms



customValidation.WEIGHT_TEXT_CRED_NOT_EQUALS_TO_1	Pasó	76 ms	9 ms
customValidation.WEIGHT_TWEET_CRED_SUM_NOT_1	Pasó	23 ms	7 ms
Tiempo Total Pruebas		382,08 s	32,817 s

Tabla 8: Tests de limpieza de texto

6.5 Prueba de Estado del Back-end

Prueba de status del endpoint	Estado	Exec. Nube	Exec. Local
/health endpoint retorna https status 200	Pasó	11,720 ms	97 ms
Tiempo Total Pruebas		371,68 s	23,825 s

Tabla 9: Prueba de estado

- Tiempo total de ejecución de las suites de pruebas con la base de datos local: 60.195 s
- Tiempo total de ejecución de las suites de pruebas con la base de datos en la nube: 419.659 s

6.6 Tiempos de Ejecución

A su vez, se calculó el tiempo de ejecución del *back-end* con la base de datos local y la base de datos en la nube:

Cantidad de tweets analizados	Tiempo ejec. nube	Tiempo ejec. local	Cantidad de tweets analizados	Tiempo ejec. nube	Tiempo ejec. local
5	15679 ms	24911 ms	60	82833 ms	276853 ms
10	18724 ms	50785 ms	70	56231 ms	328984 ms
20	94255 ms	100755 ms	80	72855 ms	371733 ms
30	48222 ms	152975 ms	90	81209 ms	446825 ms
40	47787 ms	188062 ms	100	106087 ms	480542 ms
50	70222 ms	242138 ms	AVG	63059,45	242233 ms

Tabla 9: Prueba de estado

A pesar de esperar tiempos de ejecución más bajos al tener la base de datos de manera local, se puede observar un aumento de casi el 400% con respecto al tiempo de ejecución con la base de datos en nube. Esto puede deberse a factores externos como el hardware del computador utilizado para medir el tiempo de ejecución, así como el S.O y los posibles otros procesos que éste estuviese ejecutando al momento de estar corriendo las pruebas.

7. Conclusiones y trabajos futuros

Con el progreso tecnológico, compartir información y noticias con el mundo se ha vuelto mucho más sencillo, siendo las redes sociales el principal medio para hacerlo. Esto permite que las personas estén al tanto de los eventos que suceden en todo el mundo. No obstante, también permite que individuos o grupos utilicen estos medios para difundir información falsa o aprovecharse de usuarios que no pueden identificar fácilmente los riesgos potenciales, como fraudes.

En este trabajo se presentó una nueva versión del *framework* T-CREo, al extender el modelo del cálculo de credibilidad, con nuevos módulos como la detección de bots y el análisis de tópicos. Dicha extensión permite mejorar el análisis de credibilidad, al evaluar distintos parámetros, desde el texto del tweet, hasta la información del usuario en Twitter. Asimismo, se refactorizó la anterior versión de T-CREo para brindar mayor modularidad, en caso de extender el modelo. Como también, se desarrolló una interfaz que imita la página web de Twitter, en caso de que se quiera extender el framework con la Twitter API.

Sin embargo, dado los constantes cambios que ha presentado Twitter en los últimos años, desde las políticas de uso en su API hasta la información que se difunde en la red social, pensamos que en un futuro existirán otros parámetros que se puedan llegar a considerar para seguir haciendo a T-CREo una iniciativa exitosa. Entre ellos, sería hacer uso de los emojis para extraer mayor información del tweet, realizar un análisis de sentimientos de un usuario, registrar los resultados obtenidos del cálculo de credibilidad para mantener un historial y evaluar el rendimiento del cálculo para posibles optimizaciones. También, se podría mejorar el dataset de entrenamiento para la detección de bots y el módulo de tópicos, con datos mucho más actualizados. Asimismo, proponemos que T-CREo continúe más allá de Twitter y expanda sus fronteras a otras redes sociales similares tales como Threads, recientemente creada por Facebook, o Mastodon: las cuales siguen en sus etapas iniciales.

8. Referencias

- Dongo, I., Cardinale, Y., and Aguilera, A., “Credibility analysis for available information sources on the web: A review and a contribution,” in 2019 4th International Conference on System Reliability and Safety (ICSRS). IEEE, Nov. 2019. [Online]. Available: <https://doi.org/10.1109/icsrs48664.2019.8987623>
- Dongo, I.; Cardinale, Y.; Aguilera, A.; Martínez, F.; Quintero, Y.; Barrios, S. Web Scraping versus Twitter API: A Comparison for a Credibility Analysis. In Proceedings of the 22nd International Conference on Information Integration and Web-Based Applications & Services, iiWAS '20, Chiang Mai, Thailand, 30 November–2 December 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 263–273.+ Available: <https://dl.acm.org/doi/10.1145/3428757.3429104>
- Hernandez-Mendoza, M.; Aguilera, A.; Dongo, I.; Cornejo-Lupa, J.; Cardinale, Y. (2022) Credibility Analysis on Twitter Considering Topic Detection, Caracas, Miranda: Appl. Sci [Online] Available: <https://doi.org/10.3390/app1218908>
- Internet Live Stats. (2023). *Twitter Usage Statistics*. Internet Live Stats. Retrieved July 21, 2023, from <https://bit.ly/3rBXnef>
- Iqbal, M. (2023, julio 18). *Twitter Revenue and Usage Statistics (2023)*. Business of Apps. Retrieved July 21, 2023, from <https://bit.ly/3K6Sss1>
- Langin, K. (2018, marzo 8). *Fake news spreads faster than true news on Twitter—thanks to people, not bots*. Science. Retrieved July 20, 2023, from <https://bit.ly/3QdRxd8>
- Vosoughi, S., You, D., & Aral, S. (2023, Junio 22). *The spread of true and false news online*. Science. Retrieved July 20, 2023, from <https://bit.ly/3Q5WKE7>
- Quinteros, P; Aguilera, a; Dongo, I; Cardinale, Y. (2023) CrediBot: Applying Bot Detection for Credibility Analysis on Twitter: IEEE



Wu, K. (2022, septiembre 12). *How many people come to Twitter for news? As it turns out, a*

LOT. Twitter Blog. Retrieved July 20, 2023, from <https://bit.ly/3pPgotn>

Y. Cardinale, I. Dongo, G. Robayo, D. Cabeza, A. Aguilera and S. Medina, (2021) T-CREo:

A Twitter Credibility Analysis Framework, Caracas, Miranda: IEEE [Online]

Available: doi: 10.1109/ACCESS.2021.3060623.