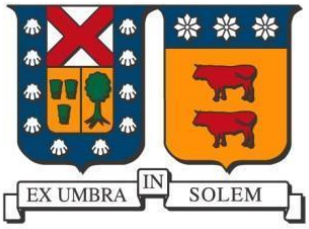
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



"EXTRACCIÓN DE DATOS PÚBLICOS EN REDES SOCIALES MEDIANTE TÉCNICAS DE WEB SCRAPING"

MATÍAS FAJARDO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Xavier Bonnaire
Profesor Correferente:

Octubre - 2022

DEDICATORIA

Para mi amada familia, amados amigos y todos que fueron parte de este proceso y camino. Nada de esto sería posible sin el apoyo incondicional de ustedes. Gracias totales <3.

AGRADECIMIENTOS

En primer lugar quiero dedicar este proyecto a mis padres Cecilia e Iván por el apoyo incondicional durante todo este tiempo que estuve en la universidad a mis hermanos Sebastián y Jacqueline también por sus consejos, mención especial a mi madrina Yeny que me apoya constantemente y como siempre digo, es mi segunda mamá y yo me siento como un hijo más.

Por otro lado, dedicar esto también a los “Malayas” que ya no son sólo compañeros de universidad, sino más bien amigos para toda la vida, Diego, Cristóbal (Teté), Jorge, Álvaro (Rusio), Roberto (Payo), Chris, Nico, Joan, Mauri y Bryan, fueron parte más que esencial en este proceso formamos una familia durante estos casi 6 años hubieron buenos y malos momentos y siempre supimos apoyarnos entre todos, independiente si a día de hoy siguen en la universidad. Agradecer también a los chicos que conocí en este camino que también considero amigos con los cuales seguimos compartiendo diversas actividades y momentos, Guille, Raúl, Bernal, Gianni, entre otras increíbles personas que conocí.

Imposible no dar mención honrosa a Diego y Teté que prácticamente estuvimos durante todos los ramos, proyectos y actividades juntos, nos apoyamos como hermanos, sólo nosotros sabemos lo que costó, imposible no acordarse de la “remontada épica” de la feria de software, desde ese momento supimos que nada nos detenía.

En fin, dar las gracias a todas las personas que influyeron de alguna manera en este largo proceso, a los profesores también por la calidad de enseñanza entregada.

RESUMEN

Resumen— La Policía de investigaciones constantemente se enfrenta a investigaciones relacionadas a las redes sociales, ya que estas, son muy utilizadas por todo tipo de personas, el problema es que a día de hoy esto se realiza de manera manual evidentemente esto implica un consumo de tiempo muy grande, ya que implica la obtención y manipulación de información, para esto se desarrolló una aplicación basada en la técnica scraping que automatiza este proceso significando una disminución notable en cuanto al proceso de extracción de información, con ello apoyar en las labores investigativas y éstas sólo sean enfocadas al análisis de la data obtenida.

Palabras Clave—investigación,*scraping,* extracción de información, redes sociales, consumo de tiempo

ABSTRACT

Abstract*— PDI constantly confronts investigations related to social media, because, they are widely used por every people, the problem today is the investigations is done manually evidently this implies a lot of resources and time, because implies the obtention and manipulation of information, this is why we made an application base on scraping technique that automate this process, this made a disminution of time extracting of information, and with that support the investigative work, and this resources just focus in the analysis of the obtained data.*

Keywords—*investigation, scraping, information extraction, social media, consume time*

GLOSARIO

API: Application Programming Interface, es una vía de comunicación entre dos aplicaciones.

CLI: Command Line Interface, Linea de comandos.

CP: *Children porn.* Pornografía Infantil

DNS: Sistema de nombres de dominio.

OSINT: Open Source Intelligent, Inteligencia de fuentes abiertas, se refiere a la obtención de información desde cualquier sitio de libre acceso.

PDI: Policía de Investigaciones.

Posts: Publicaciones realizadas en Facebook.

Tweets: Publicaciones realizadas en Twitter.

Url: *Uniform Resource Locator*, mecanismo usado por los navegadores para obtener cualquier recurso publicadon en la web

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Durante el último tiempo las redes sociales se han transformado en un aspecto fundamental de nuestras vidas, formando parte tanto en la cotidianidad como en el trabajo para algunas personas . Ahora bien, es sabido que también existen personas que utilizan las redes sociales para ámbitos negativos, ya sea para cometer delitos (Bigas & Jiménez, 2020) o para jactarse de estos (Canal 13, 2018). Es por ello, que estas se han transformado en un

punto de investigación bastante relevante hoy en día , pudiendo utilizarse como prueba irrefutable en contra de una persona (Kelly, 2012).

Junto con estos antecedentes cabe destacar que, en la actualidad, tanto Facebook como Twitter se han mantenido como las principales redes sociales que utilizan no sólo los chilenos, si no que, el resto del mundo (Fernández, 2022).

De esta forma, podemos observar a las redes sociales como un punto fuerte en las investigaciones realizadas por la policía de investigaciones de Chile (Kelly, 2012) , ahora bien, estas investigaciones son realizadas de manera manual perdiendo por consiguiente mucho tiempo y recursos tal como mencionan en la sección 3.1 de este documento, para solventar esto es que se busca con este proyecto automatizar y optimizar el proceso de la búsqueda de información, con el fin de agilizar los procesos investigativos y que se espera que la policía de investigaciones lo utilice para evitar el trabajo manual que a día de hoy se realiza.

Para llevar a cabo este proyecto, luego de una serie de reuniones con la PDI, se lograron recabar los suficientes antecedentes y requerimientos que necesitaban para lograr una aplicación que les sea realmente útil. Con todos los antecedentes a disposición se procede a llevar a cabo este proyecto con la finalidad de brindar apoyo a las labores investigativas.

Finalmente, se pudo contrastar y probar que la solución aportaría no sólo en tiempo, sino que además apoyaría en términos de métricas y de otra serie de informaciones que a día de hoy era muy difícil o costaba mucho tiempo obtener.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

En informática existen diversas áreas muy distintas unas de otras, en esta memoria trataremos de enfocar el proyecto hacia el área de ciberseguridad y datos, en este caso, del departamento de la Brigada de Cibercrimen de la Policía de Investigaciones de Valparaíso quienes han solicitado realizar este proyecto que, en resumen, busca alivianar el proceso

de extracción de datos en redes sociales, proceso el cual hoy en día la policía realiza manualmente. Es por esto que, usualmente, suele desperdiciarse demasiado tiempo y recursos de acuerdo a los antecedentes obtenidos, ya que, aún no han encontrado una herramienta que pueda satisfacer sus necesidades.

Asimismo, la principal dificultad radica en la cantidad de data que se puede extraer en un perfil público de una red social ya que, hoy en día, las redes sociales son accesibles para todos y todas. Sin embargo, las personas no siempre le dan un uso correcto a estas y dejan en evidencia múltiples sucesos que hacen en su día a día que traspasa la línea de lo legal, por ejemplo, delincuentes que se jactan de sus delitos (Canal 13, 2018), por ello, es que se busca apoyar en esta labor a la policía, para que, el día de mañana cuando este proyecto finalice, sean capaces de realizar esta labor de forma automatizada y así puedan tener más recursos en otras labores investigativas.

Esta memoria urge realizarla, ya que, como se ha expuesto en el punto anterior, la forma que hoy en día se realiza la extracción, podría ser comprendida como poco óptima, poco eficiente, entre otros. Lo que podría provocar que la recabación de evidencia sobre algún delito o sospechoso quede en libertad por falta de pruebas, algo que hoy en día, es muy común y ocurre precisamente por falta de herramientas (Venturini & Díaz, 2014). Es por esto, que el principal interés de hacer esta memoria yace precisamente para crear herramientas útiles para el desarrollo del país y ser un aporte en el área de la ciberseguridad y el análisis de datos.

Sin embargo, este proyecto no es el primero que utiliza la tecnología OSINT[1], ya que existen otras herramientas tales como, Maltego (Maltego for Professional Investigators and Small Teams, 2022), Foca (Alonso, 2017), web preverser (Pagefreezer, 2022), entre otras, que precisamente utilizan esta metodología en sus labores principales, en donde mediante una interfaz bastante intuitiva y sencilla, se pueden realizar labores investigativas, que van, desde una investigación forense, es decir, IP'S, DNS[2], análisis de enlaces, entre otras, hasta investigaciones en redes sociales, las cuales dejan en evidencia conexiones que tienen las personas. Es decir, con estas conexiones podríamos recolectar información con respecto a la vida social del sujeto, como por ejemplo, familia, ciudad en la que vive, su trabajo, entre otras. De forma que finalmente se pueda conocer la reputación online de una persona física o jurídica (Fernandéz, 2021).

Por otra parte, Foca es una herramienta complementaria a Maltego utilizada en el análisis de metadatos en Microsoft Office, Open Office o PDF. Con esta herramienta se podrán

[1] **OSINT** (**Open Source Intelligence**), traducido como Inteligencia de Fuentes Abiertas, hace referencia al conjunto de técnicas y herramientas para recopilar información pública, analizar los datos y correlacionarlos convirtiéndolos en conocimiento útil.

[2] **DNS :** Sistema de nombres de dominio.

analizar hasta el lugar más recóndito de la ofimática[3].

Ambas herramientas, si bien cumplen el objetivo de hacer un análisis mediante metodología OSINT[4], no tienen como principal enfoque los perfiles de redes sociales públicos. Además, para el caso de maltego su versión PRO (que es la que sigue de la versión de prueba), alcanza un valor de los 999 dólares (*Maltego for Professional Investigators and Small Teams*, 2022) lo que podría verse como una desventaja. A pesar de esto, podría ser una herramienta poderosa en la cual perfectamente se podría utilizar para estos fines. Por otro lado, en esta memoria, buscamos realizar un proyecto personalizado en conjunto con la PDI, lo que nos diferenciaría con el uso de Maltego, en donde, es importante destacar, la PDI nos indicaría qué datos extraer, y cómo mostrarlos, de forma que para ellos el análisis y la recaudación de evidencia sea mucho más efectiva y rápida que a día de hoy. Ya que en algunas ocasiones se podrían extraer datos irrelevantes que podrían retrasar la investigación.

Por otro lado, web preserver (WebPreserver, 2022), se utiliza en complemento con el navegador Chrome, y es utilizada para recabar información en tiempo real sobre algún perfil de Facebook, sin embargo, para realizar esto también se necesita del pago del servicio para poder utilizar todas las funcionalidades que esta provee, a su vez es importante destacar que para utilizar esta herramienta se debe ingresar de forma manual al perfil para ir rescatando la información. Lo que no es tan óptimo el uso de esta herramienta para labores investigativas, es por esta razón que la herramienta desarrollada surge como solución a la problemática encontrada.

1.1 Objetivo General:

Elaborar e implementar un sistema automatizado capaz de extraer información pública de diversas redes sociales para generar evidencias y encontrar datos clave para la posible investigación.

1.2 Objetivos Específicos:

1. Elaborar un sistema capaz de extraer información mediante técnica de web scraping.
2. Diseñar una plataforma que sea capaz que con la información que la PDI necesite buscar sobre un perfil en una red social, realizar búsqueda de palabras claves en publicaciones, imágenes, entre otras.

[3] **OFIMATICA,** Aplicación de la informática a las técnicas y trabajos de oficina.

[4] **OSINT** (**Open Source Intelligence**), traducido como Inteligencia de Fuentes Abiertas, hace referencia al conjunto de técnicas y herramientas para recopilar información pública, analizar los datos y correlacionarlos convirtiéndolos en conocimiento útil.

3. Automatizar la extracción de información en redes sociales.

El alcance de este proyecto radica en un sistema capaz de extraer información de forma automatizada, y también presentar esta información con una interfaz sencilla, en donde la policía sea capaz de buscar información clave para sus investigaciones, validando si la información extraída le es útil para la investigación, se buscará extraer información en redes sociales que más se utilizan a día de hoy, como Facebook, Twitter, y si es posible entre otras como Reddit, de tal forma tener un abanico de posibilidades que se puedan cubrir entregando fidelidad y extender el uso de esto para poder apoyar de mejor forma el proceso investigativo.

CAPÍTULO 2: MARCO CONCEPTUAL

En este capítulo se abordará la base conceptual del proyecto. Se abordarán conceptos técnicos, metodologías y herramientas, con las cuales se realizará esta memoria.

Se precisará y delimitará el problema de forma de establecer definiciones unificando conceptos.

2.1. OSINT

Traducido al español Inteligencia de Fuentes Abiertas, esta metodología hace referencia a un conjunto de técnicas y herramientas para recopilar información pública, analizar los datos correlacionarlos convirtiéndolos en conocimiento útil, en nuestro caso, en conocimiento para una investigación futura o en curso.

En simples palabras, esta metodología la utilizamos para recabar toda la información disponible en cualquier fuente pública, pudiendo ser de una empresa, persona física o sobre cualquier otra cosa que queramos realizar una investigación, provocando que todo el cúmulo de información rescatado se convierta en inteligencia y ser más eficaces a la hora de obtener resultados.

Esta metodología se utiliza en distintos ámbitos:

- Puede ser utilizada en la etapa de reconocimiento de un pentesting[5], reconociendo hosts, información de Whois[6] (whois, 2022), subdominios, información del DNS,

[5] Un pentesting es un ataque a un sistema informático con la intención de encontrar posibles debilidades en ámbitos de seguridad y todo lo que podría tener acceso a ella, ya sea, datos o funcionalidad.
[6] Herramienta para obtener información de dominios o IP's

archivos de configuración, contraseñas, etc…

- Se utiliza para ingeniería social[7], como por ejemplo, buscar toda la información sobre un usuario(en redes sociales, documentos, etc.) buscando ser consciente de los datos disponibles para evitar ataques phishing.
- Prevención de ciberataques, obtener información acerca de la organización de forma de estar alerta y/o prevenir cualquier potencial ataque de filtración de datos, etc.
- Cualquier tipo de investigación que se pueda resolver utilizando las herramientas disponibles y metodologías.

2.1.1. Fases de la metodología OSINT

En esta sección se abordarán las distintas fases para lograr realizar la metodología OSINT. (Martínez, 2014)

Figura 1: Proceso Osint. Fuente: Elaboración Propia.

Requisitos:

en esta fase se establecen los requerimientos que se deben cumplir, es decir, aquellas

[7] La ingeniería social se refiere a la manipulación psicológica de las personas para realizar acciones que impliquen la divulgación de contenido confidencial

condiciones que de alguna u otra manera deben satisfacerse para cumplir el objetivo.

- **Fuentes de información relevante:** Consiste en especificar, a partir de la etapa anterior, las fuentes de interés para recopilar la información. Es sabido que en internet el volumen de información es demasiado, por lo que, se debe identificar y concretar las fuentes de información relevante con el fin de optimizar el proceso de adquisición.
- **Adquisición:** En esta etapa se obtiene la información a partir de las fuentes determinadas en el paso anterior.
- **Procesamiento:** Consiste en dar formato a toda la información recopilada para que posteriormente sea analizada.
- **Análisis:** En esta fase se genera la inteligencia a partir de los datos recopilados y procesados. El principal objetivo es relacionar la información obtenida desde distintos orígenes buscando patrones que sean relevantes para llegar a alguna conclusión.
- **Presentación de inteligencia:** Consiste en presentar la información de manera eficaz, potencialmente útil y comprensible, de manera que pueda ser utilizada de forma correcta.

2.1.2 Problemas de la metodología OSINT

Los principales problemas de esta metodología son los siguientes:

- **Demasiada Información:** Se sabe que internet está abarrotado de información pública por todas partes (*¿Cuánta Información Se Genera Y Almacena En El Mundo?*, 2020), por lo que, el proceso de identificar las fuentes se debe hacer de manera exhaustiva. El hecho de tener demasiadas fuentes de información podría provocar una ralentización del sistema.
- **Fiabilidad:** Es importante destacar que se debe realizar una exhaustiva valoración de las fuentes de información, ya que, cualquier error en cuanto a fiabilidad de estas, podría provocar que el trabajo de inteligencia final desemboque en resultados erróneos y desinformación.

2.1.3 Herramientas

En esta sección dividiremos las herramientas utilizadas por esta metodología en **Buscadores habituales y especializados.**

Buscadores habituales

Como se menciona en el nombre estas herramientas son habituales en nuestra navegación

por internet que si les damos un uso más avanzado estaremos utilizando esta metodología, por ejemplo tenemos google, bing, yahoo, etc. Todas estas herramientas permiten realizar búsquedas avanzadas con el propósito de conocer información más “escondida”.

Google Dork

Esta técnica se utiliza como parte de la etapa de reconocimiento en un pentesting o para encontrar información, básicamente el hacker y ciberdelincuente utilizará parámetros para realizar una búsqueda avanzada y especializada, por ejemplo, sentencias del tipo “site:cert.inteco.es” + ext: pdf”, estaremos buscando archivos pdf en el sitio que indicamos el uso de estas expresiones significan un uso avanzado en la herramienta, si bien, es algo que todas las personas tienen alcance de hacerlo, no muchas conocen estás prácticas.

Mediante estos parámetros se pueden conocer entre otras cosas, información sensible, páginas de registro, localización de servicios, etc.

Buscadores Especializados

Shodan: Esta herramienta permite entre otras cosas geolocalizar ordenadores, webcams,etc. Basándose en el software, la dirección IP, etc.

NameCHK: es una herramienta que permite comprobar si un nombre de usuario está disponible en más de 150 servicios online. De este modo, se puede saber los servicios que utiliza un usuario en concreto, ya que habitualmente la gente mantiene dicho nombre para todos los servicios que utiliza.

***Tineye*:** es un servicio que, partiendo de una imagen, indica en qué sitios web aparece. Es similar a la búsqueda por imagen que incorpora Google Imágenes. -

Buscadores de información de personas: permiten realizar búsquedas a través de diferentes parámetros como nombres, direcciones de correo o teléfonos. A partir de datos concretos localizan a usuarios en servicios como redes sociales, e incluyen posibles datos relacionados con ellos como números de teléfono o fotos. Algunos de los portales que incorporan este servicio son: Spokeo, Pipl, 123people o Wink.

Existen muchísimas otras herramientas para realizar esta metodología. En este proyecto, sin embargo, nos enfocamos en las redes sociales por lo que estas herramientas podrían ser un complemento al software que se busca desarrollar.

2.2 Técnica Scraping

El web scraping, es básicamente la técnica de rastreo utilizada por muchas herramientas a día de hoy, siendo la más conocida, google, por lo que es una técnica que se ha usado durante muchos años.

Durante el proceso del web scraping, se extrae información de páginas web para analizarlas o utilizar sus datos en otra parte, para realizar este proceso se simula una navegación a un sitio web y de esa forma acceder a distintos recursos de la propia web. En nuestro caso, y para el propósito de este proyecto se utilizarán las redes sociales como fuente de información pública, es decir, perfiles públicos, en los cuales se puedan extraer información para realizar investigaciones que necesite realizar la policía de investigaciones de Chile.

Hoy en día la policía, utiliza esta técnica de forma manual según las reuniones previas con la gente de la PDI, es decir, se utiliza demasiado tiempo y recursos en buscar la información, es por esta razón que nace el objetivo de esta memoria, que es utilizar herramientas automatizadas para rescatar la mayor cantidad de información útil en un corto lapso de tiempo, priorizando así el tiempo y los recursos limitados que posee la policía de investigaciones.

2.2.1 Scraping automático utilizando python

Como se mencionó anteriormente, y también se ha mencionado en la definición del problema, con este proyecto se busca realizar un algoritmo capaz de extraer la información de forma automatizada, buscando la optimización de este proceso y permitiendo así que las investigaciones se hagan de manera eficaz.

Python es un lenguaje que posee una gran cantidad de bibliotecas para realizar scraping en las redes sociales más importantes para obtener información OSINT (edureka, 2022).

Figura 2: Proceso de la aplicación. Fuente: Elaboración propia

Como se pudo observar en la anterior figura, se presenta un diagrama que permite visualizar todo el proceso que realizará el sistema antes de presentar la información.

Como se ha mencionado anteriormente, debemos seleccionar nuestras fuentes de información, una vez realizado este proceso, debemos extraer la información de forma rápida y automatizada, para posteriormente procesar estos datos y finalmente presentarlos en una interfaz para que la policía sea capaz de utilizarlos en sus investigaciones.

Por último mencionar las ventajas de realizar este tipo de técnica por sobre las manuales, es importante destacar que cuando una persona realiza este tipo de investigación sobre algún perfil de Facebook, Twitter, etc. Puede perder de vista cierta información que podría ser relevante, sin embargo, los scraper automáticos extraen toda la información que el usuario exige sin excepción, es por esto que claramente, el scraping es una técnica muchísimo más valiosa que una técnica manual. Pero no sólo eso, las personas que están detrás de este proyecto miembros de la brigada de cibercrimen comentaban que "*usualmente las personas eliminan publicaciones con el fin de no ser investigadas*", es por

ello que un scraper que en poco tiempo es capaz de obtener esta información de forma silenciosa toma aún más valor puesto que permitiría obtener y almacenar esta información de forma de poder comparar en algún tiempo posterior si efectivamente hubieron publicaciones eliminadas, lo que abre muchas puertas y opciones que se podrían realizar con este sistema, como por ejemplo, mediante tareas cron realizar el procedimiento cada cierto tiempo de forma de ir almacenando cambios que tuvo esta persona en cuanto a sus publicaciones y/o imágenes en el último tiempo.

A día de hoy se han realizado pruebas de concepto con respecto a Facebook, de forma poder demostrar el uso y el avance de este sistema, obteniendo la siguiente información:


```
{
   "post_id":"10220824975117315",
   "text":"",
   "post_text":"",
   "shared_text":"None",
   "time":datetime.datetime('2019','10','26','15','37'),
   "timestamp":"None",
   "image":"None",
   "image_lowquality":"None",
   "images":[

   ],
   "images_description":[

   ],
   "images_lowquality":[

   ],
   "images_lowquality_description":[

   ],
   "video":"None",
   "video_duration_seconds":"None",
   "video_height":"None",
   "video_id":"None",
   "video_quality":"None",
   "video_size_MB":"None",
   "video_thumbnail":"None",
   "video_watches":"None",
   "video_width":"None",
   "likes":0,
   "comments":0,
   "shares":0,
   "post_url":"https://facebook.com/matias.faaja/posts/10220824975117315",
   "link":"None",
   "links":[
```


Figura 3: Prueba de concepto. Fuente: Elaboración propia

Estos datos fueron obtenidos de un pequeño *script* realizado en Python el cual realiza un scraping hacía un perfil de Facebook, en este caso, se utilizó el propio.

Lo cual nos demuestra la factibilidad de este problema y la forma de poder resolverlo, cabe recalcar que esto es una prueba de concepto por lo que aún faltan muchas cosas que depurar para obtener una versión final.

2.2.2 Legalidad del Scraping

Como se ha mencionado anteriormente el scraping es una técnica utilizada por muchas empresas hace muchos años (Octoparse, 2020), por lo tanto es algo de conocimiento público.

Durante mucho tiempo el scraping se ha relacionado con la ciberdelincuencia, ya que es una técnica que muchas veces se ha utilizado con fines ilícitos, como por ejemplo, para la competencia desleal, violación de las condiciones de uso, vulneración de los derechos de propiedad intelectual, entre otros fines (Ecija, 2017). Sin embargo, esto no quiere decir que la técnica sea de por sí ilegal, si no que, depende el uso que se le de.

El propósito de este proyecto es utilizar esta técnica para la obtención pública de información, es decir, automatizar lo que la gente normalmente puede obtener visitando perfiles de forma manual, información por cierto que las personas dejan de forma voluntaria público en su red social o en Internet en general. Es importante destacar que además esta información será utilizada con fines investigativos, por lo cual se cumplen las normas de las condiciones de uso de cada red social que se utilizará para este proyecto.

2.3 Redes sociales

Hoy en día las redes sociales son una herramienta utilizada por la mayoría de las personas en todas partes del mundo (Martín & Fernández, 2020), existen diversas redes sociales que son utilizadas para distintos fines, es por ello que en este subcapítulo se buscará mencionar las más importantes y que serán relevantes para nuestra investigación.

Facebook

Esta red social es por lejos la más conocida y la más utilizada durante mucho tiempo (Fernández, 2022), llegó como una red social que permitía vincularse con otras personas del mundo, formar eventos, grupos, etc. Hoy en día, Facebook ha diversificado sus usos permitiendo hasta incluso vender productos o encontrar pareja.

Provocó que en poco tiempo, la mayoría de acciones cotidianas que hacíamos en nuestro

día a día, ahora se convertían a Facebook, tales como, publicar lo que hiciste en el día, publicar con quien estabas, etc. Todos estos cambios que se produjeron en tan poco tiempo, llevaron al ser humano a una era digital, “sin darse cuenta”.

Facebook es uno de los principales eslabones de este proyecto, ya que, es una de las fuentes de información que más se sigue utilizando para cometer delito o jactarse de alguno, no es de sorprenderse que a día de hoy, se siga atrapando a delincuentes que deciden jactarse de alguna fechoría en esta red social (Canal 13, 2018), sin embargo, hay ciberdelincuentes que han ido más lejos utilizando palabras en clave para comunicarse dificultando así la extracción de información o que sean atrapados (Bigas & Jiménez, 2020). Sin embargo, la policía está al tanto de ello, es por esto, que la información pública que nos permita la privacidad del usuario extraer podrá ser filtrada y analizada por las personas de la brigada de cibercrimen de la policía de investigaciones de Chile.

Twitter

Esta red social, si bien, no es tan universalmente utilizada como Facebook, sigue siendo una de las más utilizadas a nivel mundial (BOUISSIERE, 2021), a diferencia de Facebook, Twitter permite solo publicaciones de no más de 280 carácteres.

A día de hoy, Twitter permite reflejar lo que se está hablando o lo más popular en Chile mediante las tendencias populares, en las cuales se puede observar que es lo que más se está twitteando a día de hoy, esto provoca que en cuanto a carácter investigativo sea interesante realizar análisis de sentimientos cuando ocurren sucesos importantes en el país como por ejemplo, las elecciones.

2.3.1 Realidad nacional respecto a redes sociales

Chile no se queda lejos de este fenómeno, incrementando según un estudio hecho por Data Reportal (Kemp, 2021), en el cual plasma la realidad nacional y cuantifica el número de personas que a día de hoy utilizan las redes sociales. De acuerdo al estudio la cantidad de personas que utiliza redes sociales creció en un 6,71% respecto al año pasado (2020), eso significa alrededor de un millón de personas, también se indica que el 98,8% de esas personas las utiliza mediante algún dispositivo móvil.

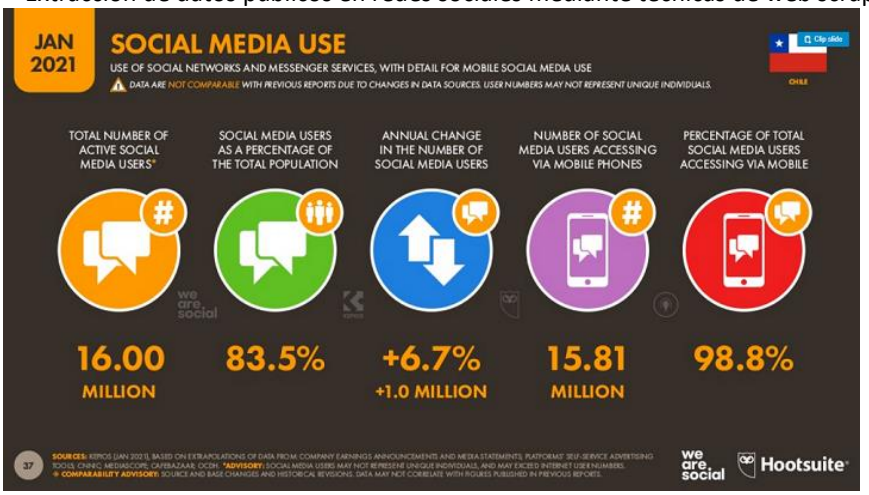


Figura 4: El uso de las redes sociales. Fuente: (Kemp,2021)

Es importante declarar que el informe indica que la **cantidad de usuarios no necesariamente indica la cantidad de personas**, ya que puede existir la posibilidad que una persona tenga múltiples usuarios, o por ejemplo, que usuarios representen a mascotas y no a personas, a colectivos y/o asociaciones. Se menciona también que el rango etario que en su mayoría las utiliza radica entre los 25 y 34 años.

¿Qué redes sociales prefieren los Chilenos?

Según el estudio entre tantas posibilidades de redes sociales que existen, Facebook es quien lidera las preferencias con una audiencia cerca del 81,1% como se indica en la siguiente figura:

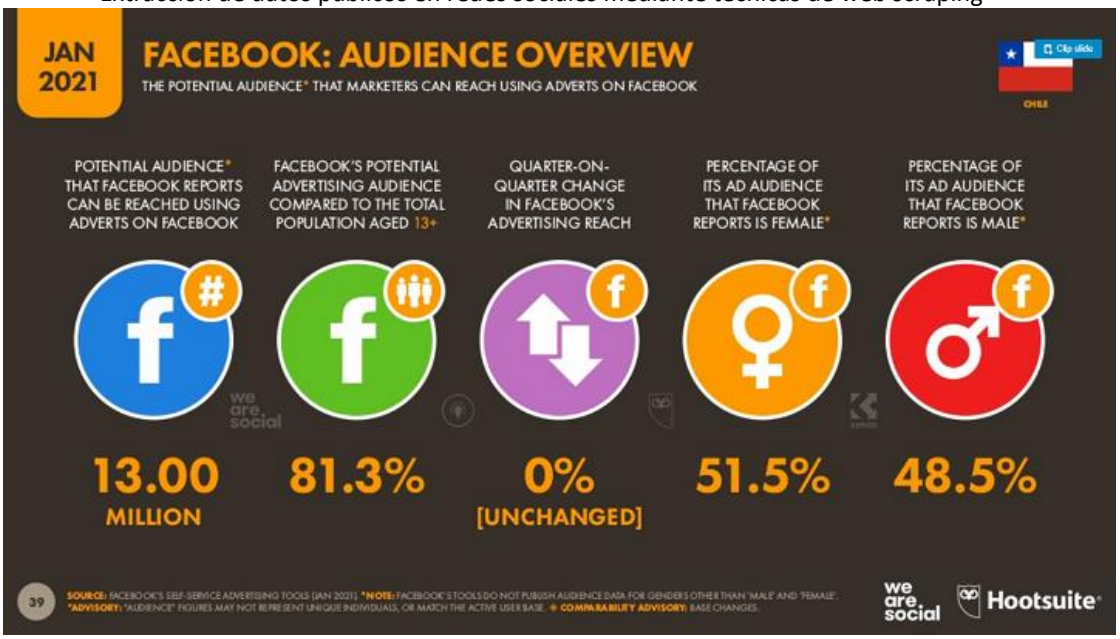


Figura 5: Cantidad de personas que utiliza Facebook. Fuente: (Kemp,2021)

Muy de cerca sigue Youtube que percibe alrededor del 80,6% de los usuarios activos.

El objetivo de nuestro estudio radica también en Twitter que si bien, en otros países es muy utilizada en Chile desde el año pasado ha perdido alrededor de 50.000 usuarios quedándose por debajo de otras redes sociales como Instagram.

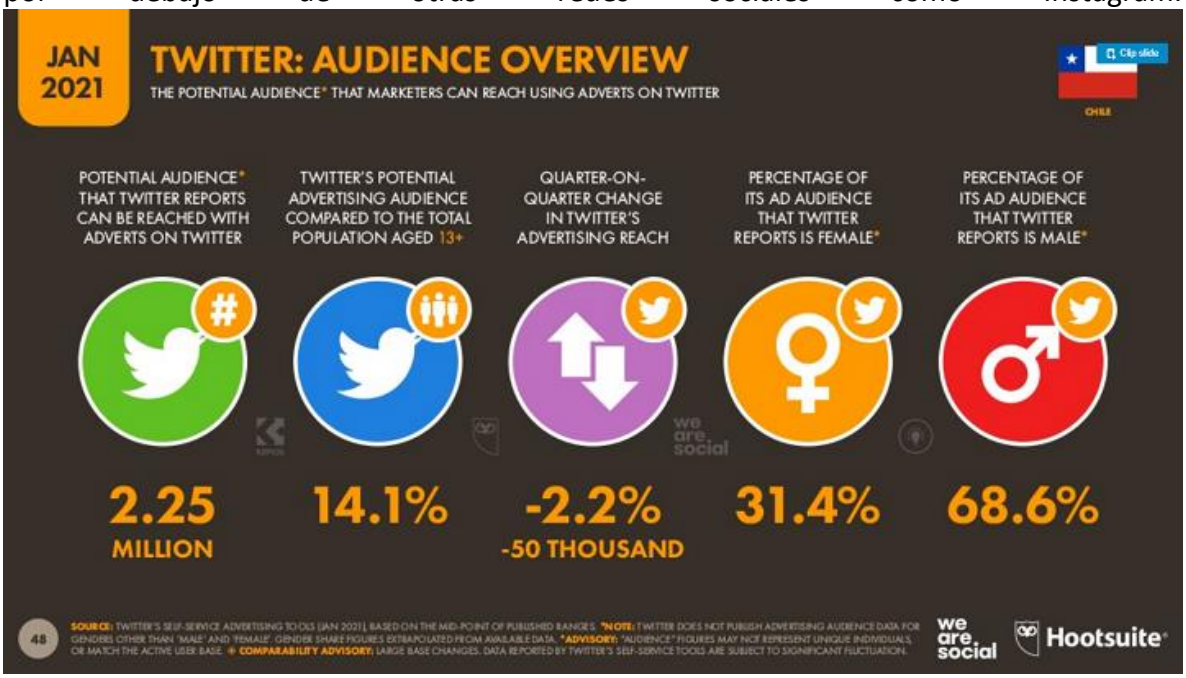


Figura 6: El uso de Twitter. Fuente: (Kemp,2021)

Sin embargo, Twitter es una buena fuente de información dado que como se mencionó anteriormente la gente suele expresar opiniones de forma más liberada. Además es sabido que en Twitter muy rara vez una cuenta suele ser privada por lo que obtener información pública no es tarea complicada, como sí lo sería en otras redes sociales como Instagram, dado que, usualmente las personas tienen su perfil en privado, ante lo cual obtener la información sería sumamente complicado a menos que se logre estar en sus seguidores.

Cabe destacar que también a día de hoy Twitter y otras redes sociales sufren de grupos de pederastas que comparten información y divulgan contenido sensible sobre pornografía Infantil.

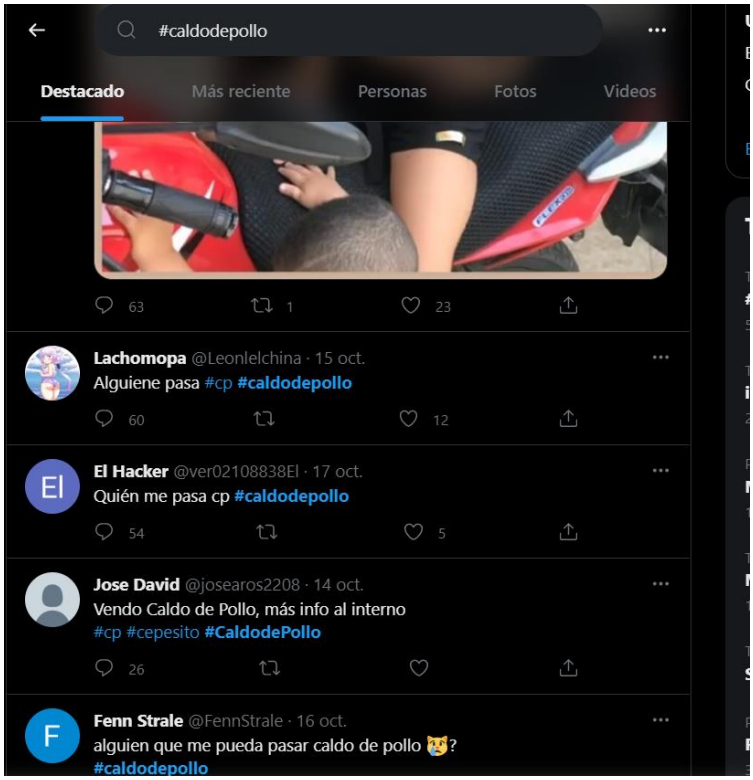


Figura 7: Divulgación de pornografía infantil. Fuente: Twitter

Como se puede observar en la imagen los ciberdelincuentes utilizan palabras en clave para evitar dar sospechas de sus delitos, en la imagen se puede observar la utilización de siglas como “Caldo de Pollo”, que hace referencias por sus acrónimos a “CP” que es *children porn*. (Bigas & Jiménez, Abuso sexual en internet: El lenguaje secreto de los pedófilos en la red: así funcionan, 2020)

Se puede notar debido a los antecedentes entregados la importancia de este proyecto y su realización, ya que, hoy en día es un problema vigente y que urge solucionar dentro de los parámetros posibles.

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

La siguiente propuesta de solución se desarrolló bajo la primicia de que las investigaciones por parte de la PDI era muy poco eficiente y requería de muchos esfuerzos para encontrar algún resultado interesante. Ante esto, se requiere de una solución que permita de forma eficaz obtener datos en perfiles de redes sociales obteniendo resultados en el corto plazo de forma rápida.

3.1 Antecedentes

De las reuniones que se tuvo con el cliente se pueden extraer varios puntos claves para el desarrollo de la solución, de los cuales se pueden desprender los siguientes:

- **Mucho tiempo para extraer la información.**

  Se menciona constantemente que los tiempos dedicados a este proceso exceden los recursos que posee la PDI.

- **Se hace de forma manual.**

  Como se ha mencionado anteriormente en esta memoria este proceso se hace de forma manual, claramente esto implica un proceso poco óptimo, este proyecto busca optimizar este proceso permitiendo que las personas que antes hacían este trabajo de manera manual, ocupen sus capacidades en investigar el resultado obtenido.

- **Las personas suelen borrar información de sus perfiles.**

  Si bien este es un aspecto que escapa de nuestro alcance, se pudo desarrollar una aplicación capaz de solventar este problema guardando cada proceso de extracción con una *timestamp* correspondiente. Ahora bien, esto implica que tanto como las imágenes como los posts son almacenados y si la imagen en un futuro o la publicación llega a ser eliminada esta ya se almacenó, por ende, se puede investigar de todas formas.

  También como forma adicional se puede automatizar aún más el proceso utilizando tareas cron, ya que, se implementó el uso de CLI[8] para poder mediante tiempos regulares de tiempo extraer la información.

[8] CLI: Command Line Interface, Linea de comandos.

- **Se necesita una interfaz sencilla que todos puedan usar.**

  Para ello desarrolló una interfaz que pide poca información para realizar el proceso, además que es sencilla de utilizar, por otro lado, la información que se extrae también posee una interfaz sencilla de utilizar y de visualizar para futuras investigaciones. Estas se pueden observar en la sección 4.2.

- **Eliminar datos irrelevantes para una investigación.**
  En este punto la PDI, se dio cuenta a partir de la prueba de concepto que existían datos irrelevantes para la investigación, en ese caso, se hicieron algunas correcciones y se desecharon algunos datos que obstruían el análisis de la información extraída.

En base a estos 4 criterios obtenidos, se puede desarrollar un prototipo que cumpla con los estándares mínimos exigidos por el cliente.

De acuerdo a lo investigado se concluyó que las dos redes sociales en donde más información se puede obtener tanto por las configuraciones de privacidad, así también por el uso masivo de estas y sus respectivas leyes, son Facebook y Twitter.

Además se hicieron consultas respecto a que información extraer de acuerdo a la prueba de concepto obtenida, ante lo cual se pueden diseñar las siguientes tablas:

Para el caso de Facebook se obtiene la siguiente información respecto a las publicaciones:

Nombre del campo	**Explicación**
Post_id	El identificador del post
text	El texto del post
Image	La url de la imagen
Comments_full	Comentarios del post
Reactors	Personas que reaccionaron
Fetched_time	El tiempo en que se realizó

Tabla 1: Información extraída de Facebook, Fuente: Elaboración Propia

Además de los perfiles de Facebook se seleccionaron los siguientes datos:

Nombre del campo	**Definición**
Friend_count	Número de amigos/amigos
Follower_count	Número de seguidores
Following_count	Número de personas siguiendo
Profile_picture	Url de la foto de perfil

Id	Identificador
Name	Nombre del perfil
Education	Universidad o colegio que pertenece
Placed_live	Lugar(es) de residencia

Tabla 2: Información del perfil de Facebook, Fuente: Elaboración propia

Para el caso de Twitter se definieron los siguientes campos de información del perfil:

Nombre del campo	**Definición**
Name	Nombre del perfil
Username	Nombre de fantasía
Bio	Biografía del perfil
Join_date	Fecha de ingreso a Twitter
Tweets	Cantidad de Tweets
Following	Cantidad de personas que sigue
Followers	Cantidad de seguidores
Profile_url	Url de la imagen de perfil
Friends	Información de seguidos y seguidores

Tabla 3: Información extraída del perfil de Twitter. Fuente: Elaboración Propia

Para el caso de los Tweets se definió la siguiente información:

Nombre del campo	**Definición**
Created_at	Fecha donde fue creado el Tweet
Username	Nombre de usuario
Name	Nombre del perfil
Tweet	Contenido del tweet
Urls	Si se compartió alguna url
Mentions	Menciones realizadas
Photos	Si se subió alguna imagen
Reply_to	Si hubieron respuestas

Tabla 4: Información obtenida de Twitter. Fuente Elaboración Propia

3.1.1 Facebook

Como se mencionó en el capítulo 2.3.1, Facebook ha sido una de las redes sociales que más se han utilizado en Chile durante muchos años, esto es un claro indicador que para obtener una buena data es necesario procesar los datos obtenidos de esta red social para realizar alguna investigación.

De acuerdo a su archivo de configuración Facebook nos menciona lo siguiente respecto al scraping automático en su página “robots.txt”.

```
# Notice: Collection of data on Facebook through automated means is
# prohibited unless you have express written permission from Facebook
# and may only be conducted for the limited purpose contained in said
# permission.
# See: http://www.facebook.com/apps/site_scraping_tos_terms.php
```

Figura 8: robots.txt Facebook. Fuente: Facebook

Se observa que se necesita un permiso especial por parte de Facebook para poder realizar el scraping, ante esto se dificulta el proceso de realizarlo, sin embargo, en este proyecto se desarrollará una herramienta capaz de obtener información básica, y con tiempos acotados de forma de poder no realizar acciones ilegales, recordando que se utiliza la metodología OSINT, es decir, obtener información de fuentes abiertas (públicas), por lo que en simples palabras busca reemplazar lo que haría una persona de manera manual.

Cabe mencionar, que Facebook en la década de 2010 estuvo en el punto de mira debido al escándalo de Cambridge Analytica (bbc, 2018), este caso fue sumamente relevante ya que deja de entrever la forma en que Facebook maneja los datos de casi 87 millones de usuarios. Por lo que hoy en día, si bien se puede obtener información de forma segura y rápida, si existen bastantes limitaciones para realizar el proceso, lo cual podría afectar de alguna manera los tiempos de ejecución de este programa.

3.1.2 Twitter

Para el caso de twitter, si bien no hay una nota expresa que nos prohíba realizar scraping, mediante nuestras investigaciones se obtuvo que fue muy complicado realizar las

investigaciones sin el uso de su API[9], por lo cual, se tuvo que utilizar su API para desarrollar la solución. Con esto, podemos obtener información como sus seguidores y seguidos, obteniendo una información mucho más fidedigna y que pueda reemplazar a la investigación manual. Ahora bien, twitter también posee el archivo "robots.txt", el cual posee la siguiente información:

```
# Every bot that might possibly read and respect this file
# ========================================================
User-agent: *
Allow: /*?lang=
Allow: /hashtag/*?src=
Allow: /search?q=%23
Allow: /i/api/
Disallow: /search/realtime
Disallow: /search/users
Disallow: /search/*/grid

Disallow: /*?
Disallow: /*/followers
Disallow: /*/following

Disallow: /account/deactivated
Disallow: /settings/deactivated

Disallow: /oauth
Disallow: /1/oauth

Disallow: /i/streams
Disallow: /i/hello
```

Figura 9: Robots.txt Twitter. Fuente: Twitter

Como se puede observar, en la sección de *followers y following* están limitados los rastreadores, por lo que, esta sería la razón de limitante a la hora de extraer este tipo de información sin el uso de la api propia de Twitter.

3.2 Precedentes

Se hizo una investigación de acuerdo a los antecedentes mencionados anteriormente y también a las redes sociales escogidas, y se llegó a la conclusión que python era uno de los mejores lenguajes para realizar este proyecto, así también el que más librerías de scraping tenia permitiéndonos realizar un proyecto que cumpla con las expectativas y así también que logre cumplir el objetivo.

Hoy en día, el scraping es una técnica muy utilizada pero también que la redes sociales se han encargado de regularizar de buena manera, en su mayoría restringiendo el uso de

[9] API: Application Programming Interface, son un conjunto de definiciones y protocolos que permiten la comunicación entre dos aplicaciones.

esta técnica y limitandola, provocando dificultades que se mencionan posteriormente en la sección 4.4

De acuerdo con todo los antecedentes previamente analizados, se comenzó con la elaboración de la solución.

En primera instancia al tratarse de un proyecto de desarrollo se planificó un plan de trabajo el cual durante los primeros meses se evaluó de acuerdo con los requerimientos del cliente las funcionalidades que esta debiese tener junto con las evaluaciones pertinentes.

Junto con ello, se realizó un estudio acerca de las librerías que se podían utilizar con python para poder realizar el proceso de extracción de información. Se probaron distintos tiempos y configuraciones para poder evitar las limitaciones que se encontraron durante el desarrollo.

Para ello se desarrolló el siguiente flujo de trabajo el cual muestra como el programa funcionará en su totalidad.

3.3 Flujo de la solución

A partir de los requerimientos observados en el capítulo anterior podemos definir un prototipo de solución el cual tendrá el siguiente flujo, que corresponderá al flujo total del programa.

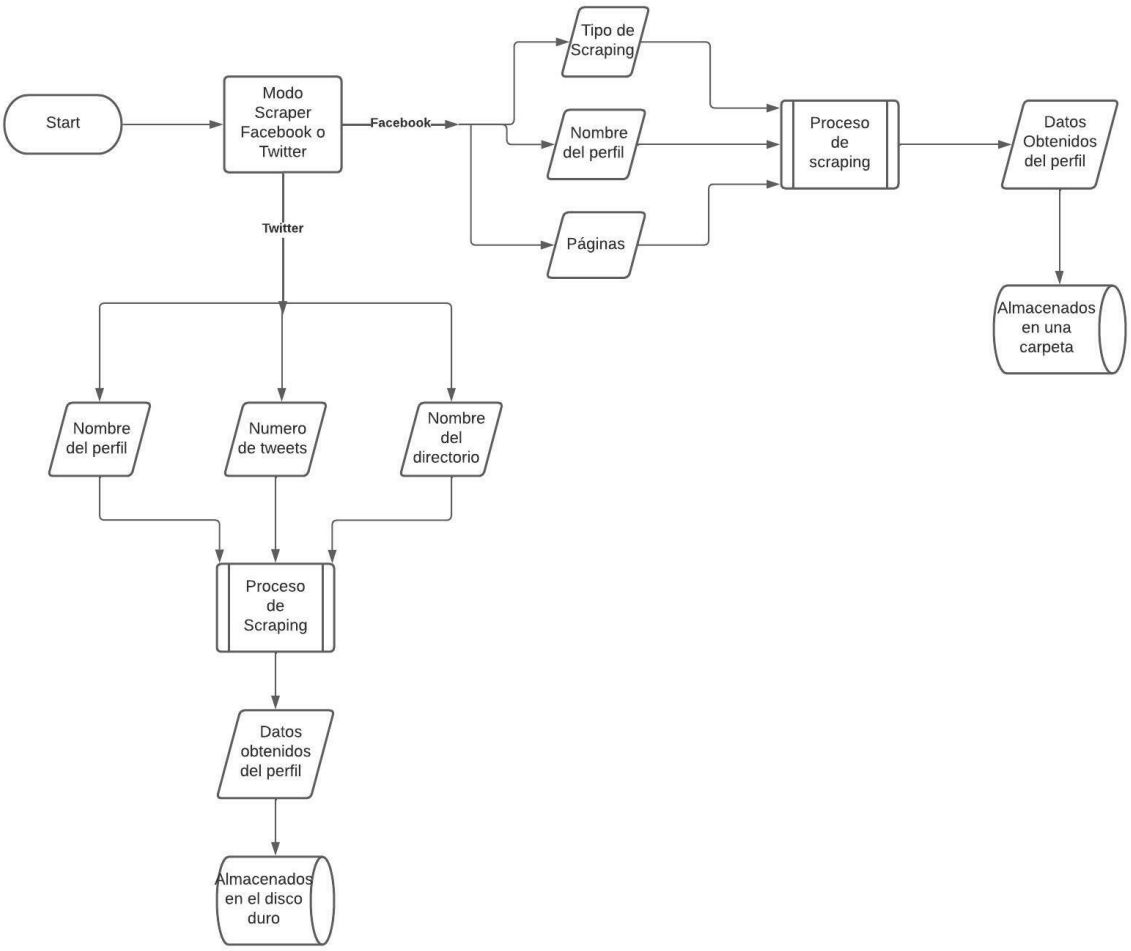


Figura 10: Estructura de la aplicación. Fuente: Elaboración propia

Como se mencionó en el capítulo anterior existen varios alcances a tener en cuenta sobre todo en ambas redes sociales, estas consideraciones son básicamente el tiempo, dado que, uno de los objetivos es ser más eficaces que cualquier investigación manual. Pero este limitante es precisamente lo más problemático de este proyecto, puesto que ambas redes sociales han puesto bastantes restricciones para extraer su información de forma automática.

3.4 Interfaz Gráfica

Durante el desarrollo de la aplicación algo que nuestro cliente hizo mucho hincapié fue el uso de una interfaz gráfica de la cual se pudiera continuar el flujo de forma sencilla, ante esto se desarrolló este primer prototipo. Que presenta la siguiente interfaz:

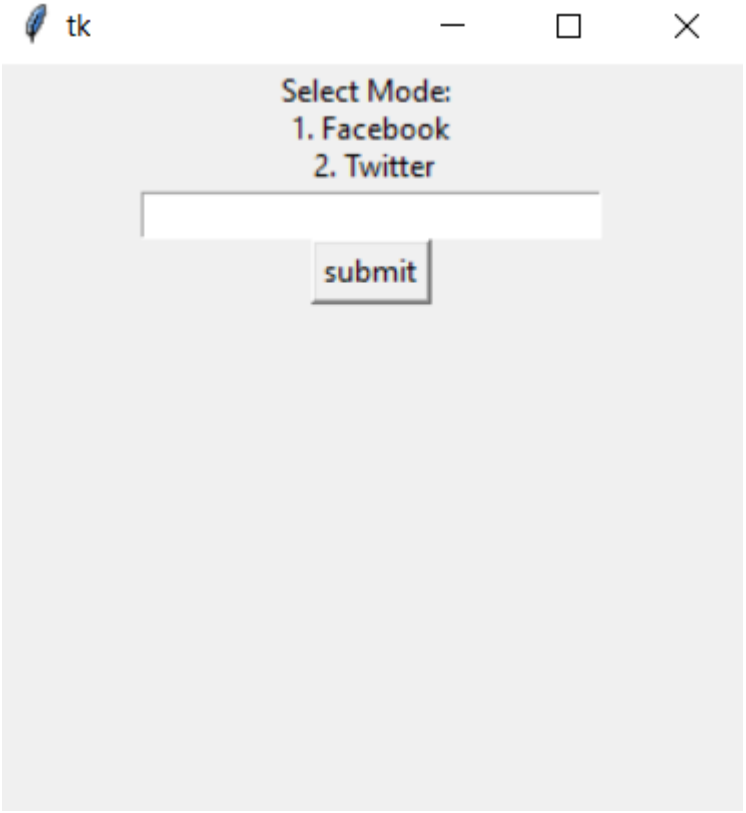


Figura 11: Selección de modo. Fuente: Elaboración propia

En esta primera etapa, se debe seleccionar a qué tipo de red social realizar la obtención de datos. Para ello, se debe ingresar el número que corresponde.

3.4.1 Flujo de la aplicación para Facebook

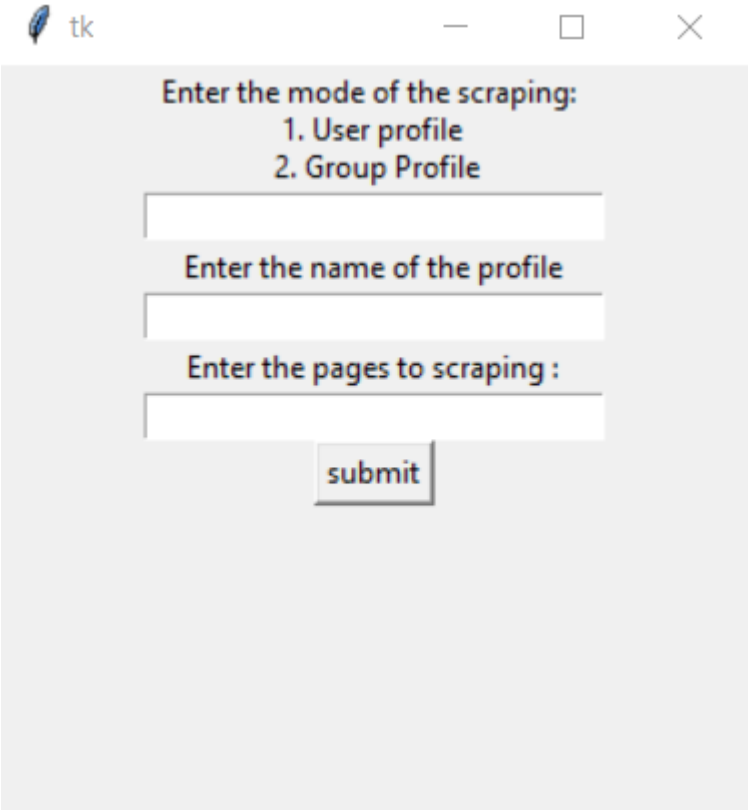


Figura 12: Flujo de facebook. Fuente: Elaboración propia

En este paso, se deberá entregar la información que el programa exige, en este caso, son 3 puntos:

- A qué tipo de perfil está dirigido la investigación, si para usuarios o para grupos de facebook.
- El nombre de este perfil.
- Y las páginas que se desean extraer, es importante destacar que entre más páginas se deseen extraer más tiempo demorara la extracción, esto es un punto importante a destacar, puesto que, como mencionamos en el subcapítulo anterior es relevante el tiempo en este trabajo.

Una vez entregados los inputs, el programa empezará a extraer la información del perfil de la persona o grupo dependiendo de lo que se haya escogido.

3.4.2 Flujo para Twitter

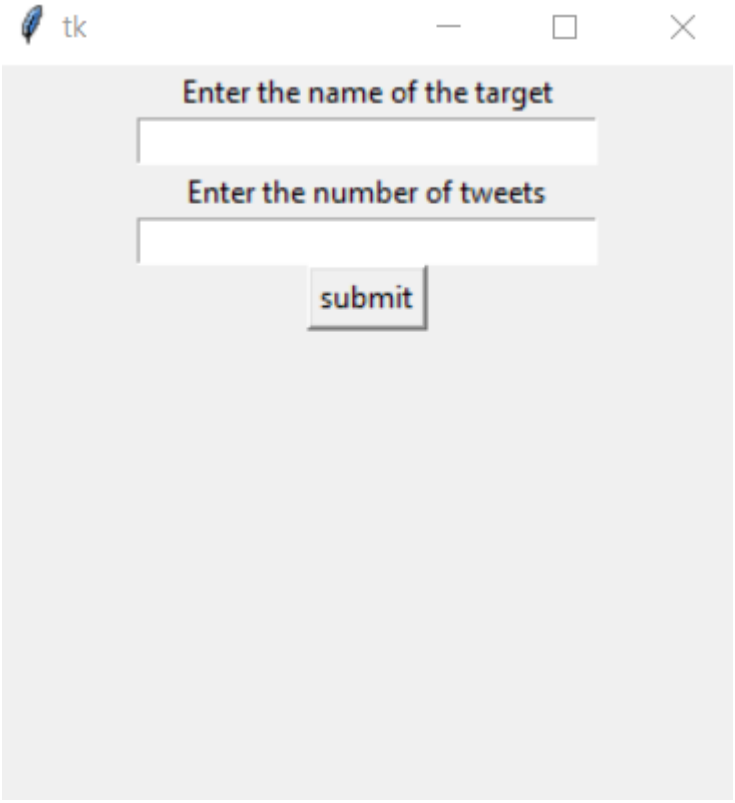


Figura 13: Flujo de Twitter. Fuente: Elaboración Propia.

En este paso, tal como con Facebook se deberá entregar la información que la aplicación pide, estos son:

- Nombre del objetivo, es decir, el nombre del perfil de Twitter de la persona a la cual queremos extraer la información.
- Número de tweets, en este caso deberemos entregar la cantidad de tweets que deseamos extraer del objetivo.

Una vez entregados estos inputs el programa comenzará a extraer la información y la guardará en una carpeta con el nombre del perfil seleccionado.

3.5 Limitaciones

Durante el desarrollo de la solución se encontraron diversas limitaciones, tales como:

- Tiempo de scraping, esto quiere decir, que debemos limitar la cantidad y el tiempo con el que la extracción de información se hace, dado que, Facebook posee una muy fuerte restricción sobre esto, recordemos que es una red social muy popular y poderosa.
- Uso de cookies[10], para poder extraer de forma eficaz la información es necesario poseer una cuenta de Facebook, y por esto, es necesario tener los cookies de estas sesiones para realizar este procedimiento.
- Bloqueos temporales, durante el desarrollo se debieron probar distintos tiempos para poder evitar estos bloqueos, por lo que, como se comentará más adelante se diseñó una solución para este problema evitando este error.
- Limitaciones para la extracción en Twitter, en esta red social también hubieron complicaciones en el tiempo de extracción para los seguidores y seguidos, esta información es manejada por la propia API de Twitter y para la investigación se utilizó una cuenta gratuita, por lo que, por razones obvias contiene bastantes limitaciones a la hora de extraer esta información (Twitter, 2022), por lo que, la cantidad de seguidores y seguidos podría ser menor a la que es originalmente, de todas formas, lo que se pudo obtener queda almacenado en un archivo llamado “Profile”.

3.6 Resultados preliminares

De acuerdo a las limitaciones, y antecedentes obtenidos se pudo realizar una extracción de los datos públicos en ambas redes sociales, quedando almacenados en carpetas divididas por cada usuario, también se logró disminuir el tiempo considerablemente, teniendo en cuenta distintos factores, tales como, las reacciones del perfil, los comentarios, entre otras características.

Todos los resultados son creados en archivos de extensión “html” y almacenados en carpetas dependiendo de qué información contienen, estos estarán divididos en las siguientes categorías:

- People, esta carpeta contiene la información de todas las personas que han reaccionado en algún post del perfil, junto con el número de interacciones y su propio perfil, de forma que sea más sencillo realizar un segundo procedimiento únicamente copiando ese nombre, y también para saber qué tan frecuente es que la persona interactúe con el objetivo.

[10] Cookies: pequeña información enviada por un sitio web y almacenada en el navegador del usuario

- Images, esta carpeta contiene todas las imágenes que se han podido extraer del perfil y cada nombre corresponde al post desde donde se obtuvo, para poder evaluar toda la información relacionada.
- Information, esta carpeta contiene la información de la duración del procedimiento de forma de poder saber los tiempos en que la aplicación demoró en obtener esa información, contiene también, el número de amigos del perfil, los posts extraídos y una estampilla de tiempo para saber cuando se hizo.
- Profile, esta carpeta contiene la información del perfil en sí, es decir, contiene la biografía de Facebook o de Twitter de la persona, también se almacena ahí la foto de perfil de forma de poder reconstruir el perfil.
- Posts, esta carpeta contiene la información de los posts del perfil.

El diagrama de de nuestra aplicación nos quedaría de la siguiente manera:

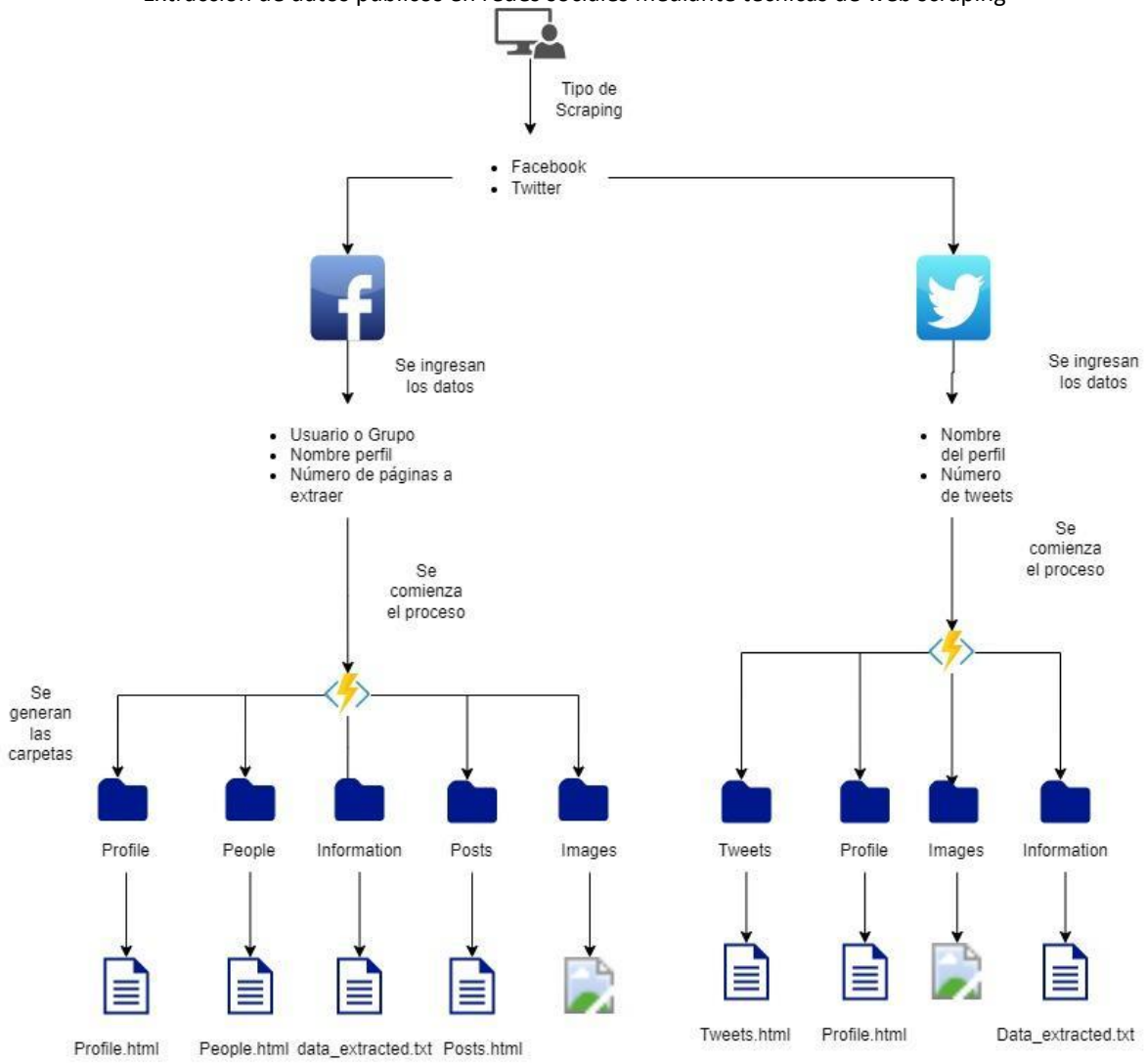


Figura 14: Estructura prototipo. Fuente: Elaboración propia

En este punto, somos capaces de extraer información y poder almacenarla, de esta forma logramos mantener un orden y eficacia a la hora de analizar perfiles de Facebook o de Twitter.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se abarcarán los análisis de los resultados, tiempos de extracción, que información se pudo extraer y algunas notas sobre las pruebas realizadas.

En primer lugar, destacar que para realizar estas pruebas se utilizó la siguiente plataforma:

Procesador: Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz

Ram: 16,0 GB

Sistema Operativo: Windows 10.

Y las cuentas utilizadas para realizar este proceso fueron cuentas creadas este año. Con el propósito de observar alguna diferencia entre cuentas de amigos o cuenta desconocidas y poder corroborar que la extracción de información sea posible.

4.1 Objetivos de las pruebas.

Los principales objetivos de las pruebas son los siguientes:

- Tiempos de extracción, observar que tiempos se emplean para extraer información, de esta forma comparar el proceso manual.
- Cantidad de data recuperada.

4.2 Tiempos de extracción

Se hizo un diagrama en el cual podemos visualizar los tiempos dependiendo de la cantidad de posts extraídos, como se puede ver en la siguiente tabla:

Facebook:

Tipo de cuenta	Cantidad de posts	Cantidad de pages	Tiempo empleado [min]
Usuario	103	10	6.1
Usuario	19	10	1
Grupo	200	10	15.2
Usuario	100	10	6.5
Usuario	101	10	6.5
Grupo	210	10	14.2
Usuario	252	25	14.79

Grupo	351	25	25.6
Grupo	545	60	45.7
Usuario	250	25	25.4

Tabla 5: Resultados preliminares. Fuente: Elaboración Propia.

Anotaciones: Las pages en el caso de Facebook, como se mencionó en el inciso 2.2, el scraping es una técnica que simula la navegación web, en este caso de Facebook, por lo que este parámetro indica el máximo de pages que el rastreador navegará.

Como se puede observar los tiempos varían muy poco de acuerdo a la cantidad de posts que se extraen, lo cual se podría relacionar al objetivo de la eficacia, que se presenta en el inicio de este documento, esto porque dada la cantidad de información capaz de extraer en los tiempos expuestos es claro notar la diferencia a que si esto se hiciera de manera manual tal como se sigue haciendo.

Es relevante destacar que mediante los timeouts, se evitaron los bloqueos de forma temporal, sin embargo, aún está la opción de aplicar distintas cuentas de Facebook para poder evitar este problema.

Es importante destacar también que los tiempos pueden variar por muchos factores, entre ellos:

- Cantidad de Imágenes.
- Cantidad de comentarios y/o reactores.
- Timeout.
- Cantidad de publicaciones.

Sin embargo, tal como se puede observar, esto hace variar los tiempos de extracción pero no de forma relevante para el caso.

Twitter:

Tipo de usuario	**Cantidad de Tweets**	**Tiempo empleado [min]**

Usuario	40	0.11
Usuario	60	0.18
Página	220	0.79
Página	100	0.17
Página	260	0.59

Tabla 6: Resultados preliminares Twitter. Fuente: Elaboración Propia.

Tal como se observa en la tabla los tiempos son muy cortos en relación a la cantidad de información extraída, lo cual se podría relacionar al objetivo de la eficacia, que se presentó al inicio del documento, esto porque como pasó con Facebook los tiempos son claramente menores a que si este proceso se hiciera de forma manual.

Entre los factores que podrían diferenciar la cantidad de tiempos que se demore la extracción, se pueden destacar:

- Cantidad de Imágenes.
- Cantidad de Tweets.
- Políticas de Twitter, en este punto, se requiere destacar que esta red social tiene políticas estrictas de acuerdo a las herramientas automatizadas de extracción de información, como se menciona en la sección 3.1.7 de este documento.

Este último factor en particular explica la razón de porque hay una diferencia abrumante entre ambas redes sociales, ya que, si bien se extrae información de forma mucho más rápida si está mucho más limitado la información que se extrae de esta. Ahora bien, esto no significa que la información obtenida no sea relevante para una investigación, al ser tiempos tan cortos y además se extraigan alrededor de 100 o más Tweets, de los cuales se puedan obtener un patrón o realizar análisis a las palabras obtenidas los hace relevante para elaborar pruebas.

Limitaciones:

A partir de las pruebas realizadas existen algunos parámetros que se deben modificar para evitar bloqueos temporales en ambas redes sociales, para ello se elaboró la siguiente tabla en la cual se indica que parámetros se deben variar en cada red social.

Facebook:

Timeout	60 segundos
Post_per_page	25(máximo)

Tabla 7: Limitaciones Facebook. Fuente: Elaboración Propia

A partir de la tabla se puede notar que el Timeout se refiere a la cantidad de tiempo de espera entre que el rastreador navegará por página, es decir, entre cada página habrá un tiempo de espera.

Por otro lado, el *post_per_page* se definió de forma de que el rastreador como máximo obtendrá 25 posts de una página.

Por otro lado en Twitter:

Seguidos	20
Seguidores	20

Tabla 8: Limitaciones Twitter. Fuente: Elaboración propia

Para el caso de Twitter en particular, como se ha mencionado a lo largo de este documento restringe el acceso de rastreadores web a la sección de seguidos y seguidores, por ende, para evitar el consumo limitado de la api se restringe el número de información.

4.6 *Dashboard*

Como se mencionó en la sección 3.1 de antecedentes se pidió un dashboard en donde se muestra la información que se extrajo para ello se desarrolló la siguiente interfaz web en la cual se puede observar la información extraída:

4.6.1 Facebook

4.6.1.1 *Information*

En este documento de texto se puede observar los tiempos, lo que se extrajo, el número de amigos, entre otra información. Es clave para generar métricas de extracción observar los tiempos y la cantidad de información que se extrae, esto se ve de la siguiente manera:

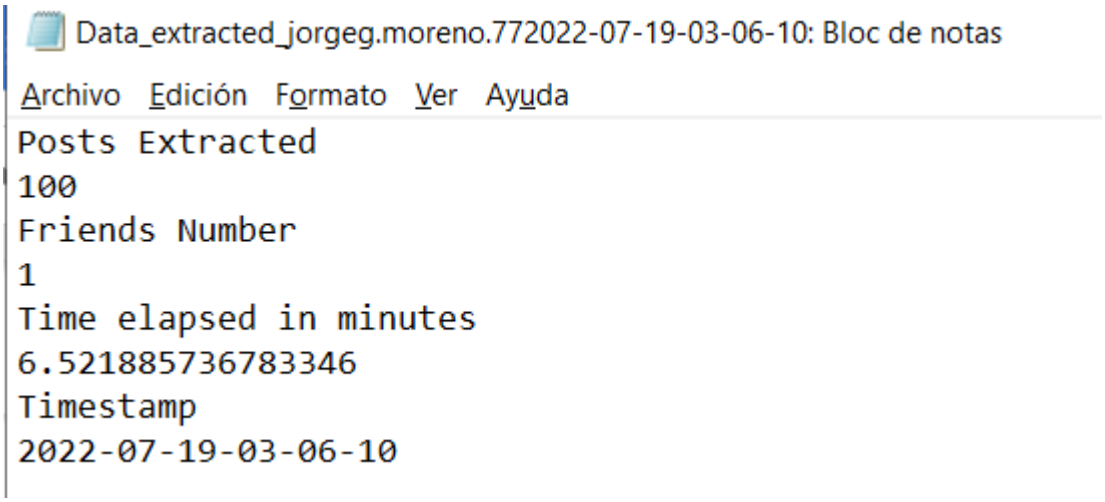


Figura 15: Data_extracted.txt. Fuente: Elaboración propia

4.6.1.2 People

En este documento se puede observar todas las personas que interactúan con el perfil en cuestión, de esta forma se puede realizar nuevamente el proceso considerando las personas que más interactúan con el perfil y así poder obtener la mayor información posible. Esto se puede ver de la siguiente forma:

Jorge Moreno	• jorgeg.moreno. / / • Numero de interacciones : 49
Kimberly Alejandra Seida Arevalo	• kimberlyalejandra.seidaarevalo.1 • Numero de interacciones : 3
Batmian Espinosa Charpentier	• ianex.espinosacharpentier • Numero de interacciones : 1
Nicolas Oporto	• Wazzzzzzzuuup • Numero de interacciones : 6
Telvy Abril Godoy Vargas	• telvyabril.godoyvargas • Numero de interacciones : 3
Christopher Arancibia Cardenas	• christopher.elpadrinocardenas • Numero de interacciones : 3
Phillipa Paredes	• pipasfritas • Numero de interacciones : 6
Camila Lagos Kurten	• camilagoos • Numero de interacciones : 3

Figura 16: Interacción de las personas. Fuente: Elaboración propia

Se puede observar que cada una de las personas contiene el nombre del perfil de Facebook, de esta forma se puede realizar nuevamente el proceso de scraping solo tomando en cuenta este nombre, luego también se observa el número de interacciones que es la cantidad de comentarios, likes, reacciones, entre otras interacciones que tuvo esta persona con el perfil en cuestión.

4.6.1.3 Posts

En esta interfaz se pueden observar todos los posts que pudieron ser extraídos, y toda la

información obtenida de cada uno de ellos, en donde como consideración adicional, nuevamente se puede observar las personas que reaccionaron o que interactuaron con los posts que publicó el perfil en cuestión.

post_id	10226832471699623
text	
post_text	
shared_text	
original_text	None
time	2021-11-02 21:13:00
timestamp	None
image	https://scontent.fscl13-1.fna.fbcdn.net/v/t39.30808-6/249340850_10226832471659622_6325368511959526711_n.jpg?stp=cp0_dst-jpg_e15_fr_q65&_nc_cat=110&ccb=1-7&_n
images	• https://scontent.fscl13-1.fna.fbcdn.net/v/t39.30808-6/249340850_10226832471659622_6325368511959526711_n.jpg?stp=cp0_dst-jpg_e15_fr_q65&_nc_cat=110&ccb=1-
images_description	• May be an image of 1 person, guitar and indoor
video	None
video_id	None
video_size_MB	None
video_watches	None
likes	23
comments	5
shares	0
post_url	https://facebook.com/diego.moragaa/posts/10226832471699623
link	None
links	
user_id	1286245816
username	Diego Moraga Arellano
user_url	https://facebook.com/Diego.Moraga.A?lst=100079425745638%3A1286245816%3A1658213216&refid=17&__tn__=qid.-5952291955393937471%3Amf_story_key.1022683247
shared_post_id	None
shared_time	None
shared_user_id	None
shared_username	None
shared_post_url	None

comments_full

comment_id	comment_url	commenter_id	commenter_url	commenter_name	commenter_meta	com
10226832875589720	https://facebook.com/10226832875589720	100026933055286	https://facebook.com/camila.riveraerazo?fref=nf&refid=13&__tn__=R	Camila Rivera Erazo	None	Cam Era
10226834439108807	https://facebook.com/10226834439108807	1316626870	https://facebook.com/paulita.vergaragonzalez?fref=nf&refid=13&__tn__=R	Paulita Vergara Gonzalez	None	Pau Ver Gon
10226837600427838	https://facebook.com/10226837600427838	1421170772	https://facebook.com/roxana.a.rojas.31?fref=nf&refid=13&__tn__=R	Roxana Azocar Rojas	None	Aw que
10226837600587842	https://facebook.com/10226837600587842	1421170772	https://facebook.com/roxana.a.rojas.31?fref=nf&refid=13&__tn__=R	Roxana Azocar Rojas	None	Rox Azo

Figura 17: Ejemplo de post. Fuente: Elaboración propia

Este es un ejemplo de cómo se vería un post extraído, como se puede observar existen distintos puntos de información para poder realizar una investigación.

4.6.1.4 Profile

En esta interfaz se puede observar toda la información del perfil disponible.

latest_post_id	5422395267810637
Friend_count	1
Follower_count	None
Following_count	None
cover_photo_text	Cover Photo: 'Y entonces me dije a mi mismo: Este es el final de un mal día pero nunca el final de mis sueños.😔'
cover_photo	https://scontent.fscl13-2.fna.fbcdn.net/v/t31.18172-8/20451727_1596068870443315_9205418134993309867_o.jpg?stp=cp0_dst-jpg_e15_fr_q65&_nc_cat=105&ccb=1-7&_nc_sid=dd9801&efg=eyJpIjoidCJ9&_nc_eui2=AeGN_z_DMICWpMKsN5LVfot0PG4IUkuk4Fo8bghSS6TgWo0fCWfZcnE3yZImIpvxiCObV-ZxpZ6qyNRd4r0M6JU6&_nc_ohc=4g7Vm51ZNF8AX9aPwcs&tn=nIWaygWAPwqNSVla&_nc_ht=scontent.fscl13-2.fna&oh=00_AT9cc8fzWTyXoKXO2RLCbvg3VG8N-c6aObdBgnl-56DWPQ&oe=62FA3845&manual_redirect=1
profile_picture	https://scontent.fscl13-1.fna.fbcdn.net/v/t1.18169-9/16508862_1411066748943529_6047587087765786048_n.jpg?stp=cp0_dst-jpg_e15_fr_q65&_nc_cat=110&ccb=1-7&_nc_sid=85a577&efg=eyJpIjoidCJ9&_nc_eui2=AeFOii769gZv8am98jrMKzF-NSF_z6GP2d41IX_PoY_Z3hCtbRAgfHowBEi14BR4BiuF_IXFxIJWJxqNPD5hRuR8&_nc_ohc=rmHMGX-zjVMAX9HOAko&_nc_ht=scontent.fscl13-1.fna&oh=00_AT8wqEDGiIR0zBboC5msyuUObTDhMJ8Ppi_1eb36z1U7wA&oe=62FC12D7&manual_redirect=1
id	100001206535293
Name	Jorge Moreno
Education	Universidad Técnica Federico Santa María College 2017 - Present Colegio Champagnat - Colegio Marista, Villa Alemana - Chile High school
Places lived	**link** / **text** / **type** /profile.php?id=106039289436408 / Quilpué / Current City /profile.php?id=106647439372422 / Viña del Mar / Hometown
Contact info	/jorgeg.moreno.77 Facebook ssjorge Instagram
Basic info	March 18 Birthday Hombre Gender Español chileno Languages
About	Tus sentimientos van directo a la deriba, y yo puedo salvarlos, mantenerte viva�?
Life events	
Friends	**id** / **link** / **name** / **profile_picture** 1326808133 / /matias.faaja / Matias Ignacio Fajardo / https://scontent.fscl13-1.fna.fbcdn.net/v/t1.18169-1/26804789_10215344897758806_7738593894836294701_n.jpg?stp=cp0_dst-jpg_e15_p100x100_q65&_nc_cat=110&ccb=1-7&_nc_sid=dbb9e7&efg=eyJpIjoidCJ9&_nc_eui2=AeFFungKutbwgoIYKZr4ypsyxA7NExvz3_vEDs0TG_Pf-3gbQ9qyqAXlEkvGXRlx4idMpGtdH109Yo2wE4OT3NHL&_nc_ohc=mjH6FjGS-AwAX8liV7D&_nc_ht=scontent.fscl13-1.fna&oh=00_AT8T5Y_Hpsxn58lCE55bQGkHBWDV1oE7_COMbEcHiAWeMg&oe=62FA45B6

Figura 18: Interfaz profile. Fuente: Elaboración propia

4.6.1.5 Images

En la sección 3.1 antecedentes se pidió observar el caso en que algunas personas eliminen publicaciones o fotografías de sus perfiles para su conveniencia, en ese contexto se soluciona descargando todas las imágenes de todos los posts y con esto se puede rescatar imágenes que pudieran ser eliminadas en un futuro. Con esto se pueden realizar

investigaciones incluso sobre imágenes que ya no están.

4.6.2 Twitter

4.6.2.1 Information

Tal como se hizo en Facebook, en este documento de texto se observa la información acerca de los tiempos de extracción y la información que se obtuvo de forma de poder generar métricas que sirvan para las investigaciones futuras.

4.6.2.2 Profile

En Twitter existe una limitante respecto a la extracción de información sobre los seguidores y seguidos, es por ello que se utilizó la api oficial de Twitter para realizar este proceso, ahora bien, esto implica ciertas limitaciones propias de Twitter y su api, ante esto, se hicieron pruebas de concepto de forma de poder visualizar cómo sería si no existiesen estas limitantes.

FRIENDS

Username	Location
deportes_13	
mau_carcenac	Montréal
yennyperales	coyhaique
jorgesaid10	Los Angeles, california
Aquisebaila13	
5mandamientos13	
mikelzulueta	chile
mujer13cl	
vamoschilenostv	
MinDesarrollo	Catedral 1575, Santiago

FOLLOWING

Username	Location
Genesisyio	
LissetteMatama1	
VelzAndres	
ColitasChuecas	
ochoa_suyin	
Catheri27572387	
Luciano78634494	
Valenti66411791	
sotofernando30	
jorgeba08169688	

Figura 19: Seguidos y seguidores. Fuente: Elaboración propia

Como se mencionó anteriormente Twitter posee restricciones para extraer este tipo de información por lo que se utilizó la api oficial la cual posee limitantes en cuanto a la cantidad y al tiempo de extracción, de esta manera, para no influir en el flujo de la aplicación se realizó esta prueba de concepto en la cual se puede observar como se distribuye esta información, lo relevante de esto es que con los nombres de usuario se puede realizar un nuevo proceso de extracción de forma de poder realizar una investigación más exhaustiva.

4.6.2.3 Tweets

En esta sección se observan los tweets correspondientes a cada perfil junto con información relevante para su posterior investigación.

id	1584330721912492034
conversation_id	1584330721912492034
created_at	2022-10-23 20:47:34 Hora verano Sudamérica Pacífico
date	2022-10-23
time	20:47:34
user_id	123955962
username	joseantoniokast
name	José Antonio Kast Rist CL
place	
tweet	El 56% de los chilenos cree que hay que mantener la Constitución o reformarla en el Congreso. Solo un 42% quiere hacer una nueva. Luego de 45 dias, ¿seguirán algun y promoviendo una nueva Convención en contra de sus intereses?
mentions	
urls	
photos	
replies_count	645
retweets_count	2773
likes_count	5708
hashtags	
cashtags	
link	https://twitter.com/joseantoniokast/status/1584330721912492034
retweet	False
video	0
thumbnail	
user_rt_id	
user_rt	
retweet_id	
reply_to	
retweet_date	

Figura 20: Ejemplo de Tweet. Fuente: Elaboración Propia

Como se puede observar, en Twitter la extracción automática está bastante limitada y por ende, no se puede observar tanta información como en Facebook, es por esto que se realiza de forma mucho más rápida pero con menos detalle.

4.6.2.4 Images

Tal como se hizo con Facebook, las imágenes son rescatadas en su totalidad dando la opción de observar imágenes que pudieran ser eliminadas, almacenándose en carpetas con su respectivo nombre.

CAPÍTULO 5: CONCLUSIONES

5.1 Conclusiones generales

En esta memoria se desarrolló un sistema capaz de extraer la información pública en perfiles

de redes sociales utilizando técnicas de web scraping en conjunto con python. Con esto se logró apoyar y optimizar el proceso que a día de hoy la PDI realizaba a mano, considerando los tiempos y los recursos que se gastan día a día realizando las investigaciones se puede concluir que la aplicación logró el principal objetivo.

Por otro lado, es importante concluir que las redes sociales hoy en día contienen información que como usuarios no nos damos cuenta que estamos entregando, pudiendo obtener una serie de información relacionada a cada persona, y además con la personas que más interactúan con uno. Esta información claramente es de suma importancia cuando se realizan investigaciones.

Es importante destacar las redes sociales seleccionadas para realizar este proyecto, tanto Facebook como Twitter, a día de hoy son muy utilizadas por millones de personas a diario, y por consiguiente es un flujo inmenso de información que se consigue todos los días. Es más, según (Fernández, 2022) Facebook encabeza el ranking de las redes sociales más utilizadas con más de 2900 millones de usuarios activos.

Finalmente es importante notar que como usuarios de una red social debemos tener cuidado con las publicaciones que hacemos o las imágenes que subimos, ya que, no todos tienen una configuración pertinente en sus perfiles los cuales podrían asegurarse sobre quien ve su perfil y quien accede a cierta información, de esta forma podemos regular las personas que son capaces de acceder a nuestro perfil.

5.2 Cumplimiento de objetivos

A modo de concluir, se puede destacar que los objetivos propuestos para este proyecto se lograron, los cuales se pueden nombrar de la siguiente manera:

1. **Elaborar un sistema capaz de extraer información mediante técnica de web scraping.**

En primer lugar, se elaboró un sistema capaz de extraer información en tiempo notoriamente menor a que si se hubiese realizado de manera manual, como se mencionó anteriormente existen limitaciones que pudiesen dificultar la libre extracción de información, estas son producidas por las propias redes sociales y en el contexto de cualquier proceso de scraping pudiesen dificultar a su realización, ahora bien, se logró mediante el desarrolló lograr evitar estas limitaciones de forma que no afecten al desarrollo de la solución. Esto significa que el sistema mediante técnicas de web scraping es capaz de extraer mucha información de varios perfiles de personas, las cuales son almacenadas en carpetas dependiendo de la información que se obtenga.

1. **Diseñar una plataforma que sea capaz que con la información que la PDI necesite**

buscar sobre un perfil en una red social, generar patrones de búsqueda en publicaciones, imágenes, entre otras.

Por otro lado, la plataforma web que se diseñó es capaz de ordenar la información que se extrae permitiendo así la búsqueda de información, también es cierto, que a modo de trabajo futuro se pudiese generar otro tipo de interfaz web con diseño de frontend a modo de agilizar este proceso de búsqueda de información.

2. Automatizar la extracción de información en redes sociales.

Por último, se logró automatizar la extracción de información en redes sociales creando un sistema capaz de no sólo realizar la extracción, si no que también organizando la información en carpetas y de esta forma poder observar de forma organizada la información que se necesita.

5.3 Futuras mejoras

5.3.1 Bloqueos temporales

Se podría diseñar una solución basada en un pool de cuentas de Facebook, esto con el propósito de que la ejecución de la aplicación no terminará de forma abrupta, esto genera ciertas ventajas y desventajas, como las que se mencionan en el siguiente subcapítulo:

Ventajas de esta solución:

- La principal ventaja es poder continuar con el flujo de la aplicación sin complicaciones y poder evitar los bloqueos temporales que se pudiesen provocar por la cantidad de requests emitidas por segundo.

Desventajas de esta solución:

- La desventaja de esta solución es que, al tener varias cuentas de Facebook para poder realizar el proceso, no se tiene control sobre que accesos tiene esta cuenta con respecto a objetivos de una investigación, por ende, la información podría estar bastante más limitada.

5.3.2 Interfaz web

En el contexto de la visualización de información se podría mejorar la interfaz web utilizando mejores patrones de búsqueda que sólo los que hay por defecto. Con esto mejorar el alcance de la aplicación y de su usabilidad.

Se podrían implementar patrones de búsqueda mediante expresiones regulares permitiendo la búsqueda veloz de la información que se necesita.

Referencias

Alonso, C. (2017). *El lado del mal*. Obtenido de https://www.elladodelmal.com/2017/10/foca-open-source.html

bbc. (2018). *bbc*. Obtenido de https://www.bbc.com/mundo/noticias-49093124

Bigas, N., & Jiménez, A. (20 de December de 2020). *Abuso sexual en internet: El lenguaje secreto de los pedófilos en la red: así funcionan.* Recuperado el 7 de September de 2022, de La Vanguardia: https://www.lavanguardia.com/vivo/lifestyle/20201220/49533241030/senales-pederastia-pedofilia-redes.html

Bigas, N., & Jiménez, A. (20 de December de 2020). *Abuso sexual en internet: El lenguaje secreto de los pedófilos en la red: así funcionan.* Recuperado el 7 de September de 2022, de La Vanguardia: https://www.lavanguardia.com/vivo/lifestyle/20201220/49533241030/senales-pederastia-pedofilia-redes.html

BOUISSIERE, Y. (2021). *Número de usuarios de Twitter ™ y cifras clave de Twitter ™ 2021 (2020) •.* Recuperado el 7 de September de 2022, de Proinfluent: https://www.proinfluent.com/es/numero-de-usuarios-de-twitter/

Canal 13. (3 de January de 2018). *Jóvenes delincuentes se jactan en redes sociales de sus delitos.* Recuperado el 7 de September de 2022, de Canal 13: https://www.13.cl/programas/bienvenidos/noticias/jovenes-delincuentes-se-jactan-en-redes-sociales-de-sus-delitos

Ecija. (13 de September de 2017). *Web Scraping: ¿legal o ilegal? - ECIJA.* Recuperado el 7 de September de 2022, de Ecija Abogados: https://ecija.com/web-scraping-legal-ilegal/

edureka. (2022). Obtenido de https://www.edureka.co/blog/web-scraping-with-python/

Fernández, R. (28 de July de 2022). • *Usuarios mundiales de las redes sociales líderes en 2022.* Recuperado el 6 de September de 2022, de Statista: https://es.statista.com/estadisticas/600712/ranking-mundial-de-redes-sociales-por-numero-de-usuarios/

Kelly, H. (4 de October de 2012). *Recopilacion de informacion en redes sociales.* Recuperado el 7 de September de 2022, de La policía usa las redes sociales para recopilar evidencias sobre crímenes: https://cnnespanol.cnn.com/2012/10/04/la-policia-usa-las-redes-sociales-para-recopilar-evidencias-sobre-crimenes/

Kemp, S. (27 de January de 2021). *Digital 2021: Global Overview Report — DataReportal – Global Digital Insights.* Recuperado el 7 de September de 2022, de DataReportal: https://datareportal.com/reports/digital-2021-global-overview-report

Maltego for Professional Investigators and Small Teams. (2022). Recuperado el 7 de September de 2022, de Maltego: https://www.maltego.com/maltego-for-pro/

Mapfre. (2020). *¿Cuánta información se genera y almacena en el mundo?* Recuperado el 7 de September de 2022, de Fundación MAPFRE: https://www.fundacionmapfre.org/blog/cuanta-informacion-se-genera-y-almacena-en-el-mundo/

Martín, A., & Fernández, C. (9 de December de 2020). *Las Redes Sociales más utilizadas: cifras y estadísticas.* Recuperado el 7 de September de 2022, de IEBS: https://www.iebschool.com/blog/medios-sociales-mas-utilizadas-redes-sociales/

Martínez, A. (28 de May de 2014). *OSINT - La información es poder | INCIBE-CERT.* Recuperado el 7 de September de 2022, de INCIBE-CERT |: https://www.incibe-cert.es/blog/osint-la-informacion-es-poder

Octoparse. (14 de October de 2020). *Los 3 Usos Más Prácticos de Herramienta de Web Scraping de Datos de Comercio Electrónico.* Recuperado el 7 de September de 2022, de Octoparse: https://www.octoparse.es/blog/3-usos-pr%C3%A1cticos-de-herramientas-de-web-scraping-de-datos-de-comercio-electr%C3%B3nico

Pagefreezer. (2022). *webpreserver*. Obtenido de https://www.pagefreezer.com/webpreserver/

Python. (2020). Recuperado el 7 de September de 2022, de Wikipedia: https://es.wikipedia.org/wiki/Python

Twitter. (2022). *About Twitter's APIs.* Recuperado el 7 de September de 2022, de Twitter Help Center: https://help.twitter.com/en/rules-and-policies/twitter-api

Venturini, J., & Díaz, M. (2014). *La PDI está revisando tu Facebook, parte II.* Recuperado el 7 de September de 2022, de Derechos Digitales: https://derechosdigitales.tumblr.com/post/89806043126/la-pdi-est%C3%A1-revisando-tu-facebook-parte-ii

WebPreserver. (Agosto de 2022). *Extensión Chrome*. Obtenido de https://chrome.google.com/webstore/detail/webpreserver/ebofmienemijnilnonphmmmahgmnpflh?hl=en

whois. (2022). Obtenido de https://who.is/

ANEXOS

Anexo Main

```
from statistics import mode
from facebook_scraper import *
import nest_asyncio
import json
from soupsieve import match
import twint
from json2html import *
import time
import os
import wget
from os import path
from pathlib import Path
from instascrape import *
from tkinter import *
import twitter_module as twt
import facebook_module as fb
import json
import time
path_handler = 'C:\\Users\\matia\\proyectos\\' # Parámetro que
modifica la ruta en donde se almacenará la información obtenida
def get_input():
    print("Saved!")
if(len(sys.argv)) == 5: # Función que permite utilizar la cron tab
    mode = sys.argv[1]
    if int(mode) == 1:
        mode_fb= sys.argv[2]
        profile = sys.argv[3]
        pages = sys.argv[4]
        fb.fb_scraping(mode_fb,profile,pages)
    elif int(mode) == 2:
        target = sys.argv[2]
        tweets = sys.argv[3]
        twt.Twitter_scraper(target,tweets)
    else:
        print("Help :\n To use Facebook mode, '1 nameprofile type
pages'\n To use Twitter mode, '2 nameprofile tweets'")
else: #Permite utilizar la interfaz gráfica
    window = Tk("Selecting Social Media to scrap")
    window.geometry("300x300")
    label1 = Label(window,text='Select Mode: \n 1. Facebook \n 2.
Twitter')
    label1.pack()
    label1.config(justify = CENTER)
    var4 = StringVar()
```

```
    entry1 = Entry(window,width=30,textvariable=var4)
    entry1.pack()
    button1 = Button(window, text = 'submit')
    button1.pack()
    button1.config(command = get_input)
    window.mainloop()
    select_mode = int(var4.get())
    if(select_mode == 1):
        if(os.path.exists(path_handler+'\\Facebook\\') == False):
            os.mkdir(path_handler + '\\Facebook\\')
        window = Tk("Facebook Scraping")
        window.geometry("300x300")
        label1 = Label(window,text='Enter the mode of the
scraping: \n 1. User profile \n 2. Group Profile')
        label1.pack()
        label1.config(justify = CENTER)
        var1=StringVar()
        var2=StringVar()
        var3=StringVar()
        entry1 = Entry(window,width=30,textvariable=var1)
        entry1.pack()
        label2 = Label(window,text='Enter the name of the
profile')
        label2.pack()
        label2.config(justify = CENTER)
        entry2 = Entry(window,width=30,textvariable=var2)
        entry2.pack()
        label3 = Label(window,text='Enter the pages to scraping
:')
        label3.pack()
        label3.config(justify = CENTER)
        entry3 = Entry(window,width=30,textvariable=var3)
        entry3.pack()
        button1 = Button(window, text = 'submit')
        button1.pack()
        button1.config(command = get_input)
        window.mainloop()
        scraper = var1.get()
        profile = var2.get()
        pages = int(var3.get())
        fb.fb_scraping(scraper,profile,pages) #Comienza el proceso
en Facebook
    elif(select_mode == 2):
        if(os.path.exists(path_handler+'\\Twitter\\') == False):
            os.mkdir(path_handler + '\\Twitter\\')
        window = Tk("Twitter Scraping")
        window.geometry("300x300")
        label1 = Label(window,text='Enter the name of the target')
        label1.pack()
```

```
        label1.config(justify = CENTER)
        var1=StringVar()
        var2=StringVar()
        var3=StringVar()
        var4 = StringVar()
        entry1 = Entry(window,width=30,textvariable=var1)
        entry1.pack()
        label2 = Label(window,text='Enter the number of tweets')
        label2.pack()
        label2.config(justify = CENTER)
        entry2 = Entry(window,width=30,textvariable=var2)
        entry2.pack()
        button1 = Button(window, text = 'submit')
        button1.pack()
        button1.config(command = get_input)
        window.mainloop()
        username = var1.get()
        n_tweets = int(var2.get())
        output = var4.get()
        twt.Twitter_scraper(username,n_tweets,output) #Comienza el
proceso en Twitter
```

Anexo Modulo de Facebook

```
from itertools import count
from socket import timeout
from unittest.util import _count_diff_hashable
from facebook_scraper import *
import nest_asyncio
import json
import time
import os
import wget
from os import path
from pathlib import Path
from tkinter import *
from json2xml import json2xml
from json2xml.utils import readfromurl, readfromstring,
readfromjson
import dicttoxml
from json2html import json2html
from collections import Counter
from itertools import chain
from collections import OrderedDict
```

```
path_handler = 'C:\\Users\\matia\\proyectos\\Facebook\\'
#Guardar la información relacionada con Facebook
def profiles_user(name,pages): #Función para rescatar la
información del perfil
    time_start = time.time()
    data_profile = get_profile(name,cookies =
"cookies.txt", friends = True,timeout=90)
    data_profile_friends = data_profile['Friends']
    friends = []
    for friend in data_profile_friends:
        friends.append(friend['name'])
    posts = []
    for post in get_posts(name,pages = pages,timeout =
60,options={"comments":True,"reactors":True,"posts_per_p
age":25}):
          posts.append(post)
    image_download(posts,name)
    output = []
    output,xmloutput = parse(posts,data_profile,name)
    soup_parse(output,name)
    xml_parse(xmloutput,name,False)
    information_profile(data_profile,name)
    total_posts = len(posts)
    total_friends = len(friends)
    et = time.time()
    total_time = et - time_start
    total_time_minute = total_time / 60.0
    timestr = time.strftime("%Y-%m-%d-%H-%M-%S")
    file_posted = 'Data_extracted_' + name + timestr
+'.txt'
    if (os.path.exists(path_handler + name) == False):
            os.mkdir(path_handler + name)
    if(os.path.exists(path_handler + name +
'\\Information') == False):
        os.mkdir(path_handler + name + '\\Information')
    file = open(path_handler + name +'\\Information\\' +
file_posted,'w',encoding="utf-8")
    message = 'Posts Extracted\n' + str(total_posts) +
'\n' + 'Friends Number\n' + str(total_friends) +'\n'
+'Time elapsed in minutes\n' + str(total_time_minute)
+'\n' +'Timestamp\n' + timestr+'\n'
```

```
    file.write(message)
    file.close()
    window = Tk("Information Obtained")
    window.geometry("300x300")
    label1 = Label(window,text='Posts extracted :')
    label1.pack()
    label1.config(justify = CENTER)
    label2 = Label(window,text=total_posts)
    label2.pack()
    label2.config(justify = CENTER)

    friends_reactor = []
    friends_interacction = {}
    for i in posts:
        if(i['reactors'] == None):
            continue
        friends_reactor.append(i['reactors'])
    print(type(friends_reactor))
    friends_name = []
    friends_total = {}
    for j in friends_reactor:
        for k in j:
            friends_name.append(k.get('name'))
            string2 = k.get('link').split('?')[0]
            string3 = string2.split('/')[3]
            friends_interacction[k.get('name')] =
(string3)
    countabilizer = dict((i,friends_name.count(i)) for i
in friends_name)
    for key in friends_interacction:
        for key2 in countabilizer:
            if(key == key2):
                friends_total[key] =
(friends_interacction[key],'Numero de interacciones : '
+ str(countabilizer[key2]))
    json_object = json.dumps(friends_total)
    jsonparse = json2html.convert(json_object)
    friendsparse(jsonparse,name)
    label2 = Label(window,text='El usuario tiene la
lista de amigos privada')
    label2.pack()
```

```
    label2.config(justify = CENTER)
    label6 = Label(window,text='Friends number :')
    label6.pack()
    label6.config(justify = CENTER)
    label3 = Label(window,text=total_friends)
    label3.pack()
    label3.config(justify = CENTER)
    label4 = Label(window,text='Time elapsed')
    label4.pack()
    label4.config(justify = CENTER)
    label5 = Label(window,text=total_time)
    label5.pack()
    label5.config(justify = CENTER)
    window.after(10000,lambda:window.destroy())
    window.mainloop()
def friendsparse(output,name): #Rescatar los amigos
    timestr = time.strftime("%Y-%m-%d-%H-%M-%S")
    if (os.path.exists(path_handler + name) == False):
            os.mkdir(path_handler + name)
    if(os.path.exists(path_handler + name + '\\People')
== False):
        os.mkdir(path_handler + name + '\\People')
    filename = "People_Interaction - " + name +"--"+
timestr + '.html'
    savepath = name
    path_name = path_handler + savepath + '\\People'
+'\\' + filename
    file = open(path_name,'w',encoding="utf-8")
    inicio = """<html>
   <head></head>
   <body>"""
    fin= """
   </body>
   </html>"""
    message = inicio + output + fin
    file.write(message)
    file.close()
def soup_parse(output_lists,profile): #Creación del html
con los posts
    timestr = time.strftime("%Y-%m-%d-%H-%M-%S")
    string_entero = ''
```

```
    if (os.path.exists(path_handler + profile) ==
False):
        os.mkdir(path_handler + profile)
    if(os.path.exists(path_handler + profile +
'\\Posts\\') == False):
        os.mkdir(path_handler + profile + '\\Posts\\')
    for i in output_lists:
        string_entero = string_entero + "<div> ---Nuevo
Post--- </div>" + i
    filename = timestr + '.html'
    savepath = '\\' + profile + '\\Posts\\'
    path_name =  path_handler + savepath + '\\' +
filename
    file = open(path_name,'w',encoding="utf-8")
    inicio = """<html>
   <head></head>
   <body>"""
    fin= """
   </body>
   </html>"""
    message = inicio + string_entero + fin
    file.write(message)
    file.close()
def xml_parse(xml,profile,status): #Creación en formato
xml de los posts
    timestr = time.strftime("%Y-%m-%d-%H-%M-%S")
    if (os.path.exists(path_handler + profile) ==
False):
        os.mkdir(path_handler + profile)
    if (os.path.exists(path_handler + profile) ==
False):
        os.mkdir(path_handler + profile)
    if(os.path.exists(path_handler + profile +
'\\Posts\\') == False):
        os.mkdir(path_handler + profile + '\\Posts\\')
    if status == True:
        filename = profile + '.xml'
    else:
        filename = timestr + '.xml'
    savepath = '\\' + profile + '\\Posts\\'
    path_name = path_handler + savepath + '\\' +
```

```
filename
    file = open(path_name,'wb')
    message = xml
    file.write(message)
    file.close()
def image_download(posts,profile): #Funcion que descarga
las imagenes
    for i in posts:
        url = i['image']
        print(url)
        if (os.path.exists(path_handler + profile) ==
False):
            os.mkdir(path_handler + profile)
        if(os.path.exists(path_handler + profile +
'\\Images') == False):
            os.mkdir(path_handler + profile +
'\\Images')
        out = path_handler + profile + '\\Images\\' +
i['post_id'] + '.jpg'
        try:
            wget.download(url,out)
        except:
            print("No hay url")
def parse(posts,data_profile,profile): #Función que
retorna la listas de los posts
    outputs_lists = []
    output_xmllists=[]
    xml = dicttoxml.dicttoxml(posts)
    for i in posts:
        del i['images_lowquality']
        del i['image_lowquality']
        del i['video_quality']
        del i['is_live']
        del i['was_live']
        del i['image_ids']
        del i['video_thumbnail']
        del i['video_width']
        del i['factcheck']
        del i['image_id']
        del i['video_height']
        del i['images_lowquality_description']
```

```
            del i['video_duration_seconds']
            del i['w3_fb_url']
            del i['available']
            output = json2html.convert(json = i)
            outputs_lists.append(output)
        return outputs_lists,xml
def information_profile(data_profile,name): #Parsea la
información del perfil en formato html
        output = json2html.convert(data_profile)
        xml = dicttoxml.dicttoxml(data_profile)
        status = True
        xml_parse(xml,name,status)
        timestr = time.strftime("%Y-%m-%d-%H-%M-%S")
        if (os.path.exists(path_handler + name) == False):
            os.mkdir(path_handler + name)
        if(os.path.exists(path_handler + name + '\\Profile')
== False):
            os.mkdir(path_handler + name + '\\Profile')
        filename = "Profile_" + name +"--"+ timestr +
'.html'
        savepath = name
        path_name = path_handler + savepath + '\\Profile'
+'\\' + filename
        file = open(path_name,'w',encoding="utf-8")
        inicio = """<html>
       <head></head>
       <body>"""
        fin= """
       </body>
       </html>"""
        message = inicio + output + fin
        file.write(message)
        if (os.path.exists(path_handler + name) == False):
                os.mkdir(path_handler + name)
        if(os.path.exists(path_handler + name + '\\Profile')
== False):
            os.mkdir(path_handler + name +'\\Profile')
        out = path_handler + name + '\\Profile'
        try:
            url = data_profile['profile_picture']
        except:
```

```
            pass
        try:
            wget.download(url,out)
        except:
            print("No hay url")
        file.close()
def group_profile(name,pages): #Obtiene la información
de los grupos
    time_start = time.time()
    data_profile = get_group_info(name, cookies =
"cookies.txt",timeout=60)
    posts = []
    for post in
get_posts(group=name,options={"comments":True,"reactors"
:True,"posts_per_page":25},timeout = 60):
        posts.append(post)
    if (len(posts) == 0):
        print("Grupo Privado")
    total_posts = len(posts)
    friends_reactor = []
    friends_interacction = {}
    for i in posts:
        if(i['reactors'] == None):
            continue
        friends_reactor.append(i['reactors'])
    friends_name = []
    friends_total = {}
    for j in friends_reactor:
        for k in j:
            friends_name.append(k.get('name'))
            string2 = k.get('link').split('?')[0]
            string3 = string2.split('/')[3]
            friends_interacction[k.get('name')] =
(string3)
        countabilizer = dict((i,friends_name.count(i))
for i in friends_name)
    for key in friends_interacction:
        for key2 in countabilizer:
            if(key == key2):
                friends_total[key] =
(friends_interacction[key],'Numero de interacciones : '
```

```
+ str(countabilizer[key2]))
    json_object = json.dumps(friends_total)
    jsonparse = json2html.convert(json_object)
    friendsparse(jsonparse,name)
    image_download(posts,name)
    output = []
    output,_ = parse(posts,data_profile,name)
    soup_parse(output,name)
    information_profile(data_profile,name)
    et = time.time()
    total_time = et - time_start
    total_time_minute = total_time / 60.0
    timestr = time.strftime("%Y-%m-%d-%H-%M-%S")
    file_posted = 'Data_extracted_' + name + timestr
+'.txt'
    if (os.path.exists(path_handler + name) == False):
            os.mkdir(path_handler + name)
    if(os.path.exists(path_handler + name +
'\\Information') == False):
        os.mkdir(path_handler + name + '\\Information')
    file = open(path_handler + name +'\\Information\\' +
file_posted,'w',encoding="utf-8")
    message = 'Posts Extracted\n' + str(total_posts) +
'\n'  +'\n' +'Time elapsed in minutes\n' +
str(total_time_minute) +'\n' +'Timestamp\n' +
timestr+'\n'
    file.write(message)
    file.close()
    #for i in posts:
     #   print(i)
def get_input():
    print("Saved!")
def fb_scraping(scraper,profile,pages):
    if(scraper == "1"):
        print("Doing the Scrap ...")
        profiles_user(profile,pages)
    elif(scraper == "2"):
        print("Doing the Scrap ...")
        group_profile(profile,pages)
```

Anexo Modulo Twitter

```
from statistics import mode
import nest_asyncio
import json
import twint
from json2html import *
import time
import os
import wget
from os import path
from pathlib import Path
from tkinter import *
import twitter_module as twt
import tweepy
from json2html import json2html
from tabulate import tabulate
from json2xml import json2xml
from json2xml.utils import readfromurl, readfromstring,
readfromjson
path_handler = 'C:\\Users\\matia\\proyectos\\Twitter\\'
#Directorio en donde se almacena la información relacionada con
twitter
#Tokens correspondientes a la api oficial de Twitter
consumer_key = "xRFVXe6YJXVscB9vALcfkpLno"
consumer_secret =
"hTlt79nU7nOxyf8SP17z3COSyJ6o5N8i9B6kBXZwoIEytjh3yk"
access_token = "4389712335-
mpl678GHGP72FJBGamR9JR7ZH1PTzw8fsopCM5j"
access_token_secret =
"nWl6SnmIAVkSwHIPMPaL9ho9Gv4zI33JphlqjFDGyDqBq"
def information_profile(output,profile,friends,followers):
##Funcion para obtener la información del perfil
    timestr = time.strftime("%Y-%m-%d-%H-%M-%S")
    if (os.path.exists(path_handler + profile) == False):
        os.mkdir(path_handler + profile)
    if(os.path.exists(path_handler + profile + '\\Profile') ==
False):
        os.mkdir(path_handler + profile +'\\Profile') #Revisa si
ya existe la carpeta y si no, la crea
    filename = "Profile_" + profile +"--"+ timestr + '.html'
    savepath = path_handler + '\\' + profile
    path_name =  savepath + '\\Profile' + '\\' + filename
    file = open(path_name,'w',encoding="utf-8")
    friends_html =
tabulate(friends,tablefmt='html',headers=["Username","Location"])
    print(friends_html) #Genera tablas de html para los amigos
    followers_html =
tabulate(followers,tablefmt='html',headers=["Username","Location"]
)
    print(followers_html) # Genera tabla de html para los
```

```
seguidores
    inicio = """<html>
   <head></head>
   <body>"""
    intermedio = """<div> <b>FRIENDS</b> </div>"""
    intermedio_2 = """<div> <b>FOLLOWING </b> </div>"""
    fin= """
   </body>
   </html>"""
    message = inicio + output + intermedio + friends_html
+intermedio_2 +followers_html + fin #Se escribe todo en el archivo
    print(message)
    file.write(message)
    file.close()
    os.remove(path_handler + profile + '\\users.json')
def soup_parse(output_lists,profile): #Genera el archivo con los
tweets
    timestr = time.strftime("%Y-%m-%d-%H-%M-%S")
    string_entero = ''
    if (os.path.exists(path_handler + profile) == False):
        os.mkdir(path_handler + profile)
    if (os.path.exists(path_handler + profile + '\\Tweets\\') ==
False):
        os.mkdir(path_handler + profile + '\\Tweets\\')
    for i in output_lists:
        string_entero = string_entero + "---New Post---" + i
    filename = timestr + '.html'
    savepath = '\\' + profile + '\\Tweets\\'
    path_name = path_handler + savepath + '\\' + filename
    file = open(path_name,'w',encoding="utf-8")
    inicio = """<html>
   <head></head>
   <body>"""
    fin= """
   </body>
   </html>"""
    message = inicio + string_entero + fin
    file.write(message)
    file.close()
def get_input():
    print("Saved!")
def image_download(data,profile,mode): #Descarga las imagenes
obtenidas de los posts
    if mode == 1:
        if (os.path.exists(path_handler + profile) == False):
            os.mkdir(path_handler + profile)
        if(os.path.exists(path_handler + profile + '\\Images') ==
False):
            os.mkdir(path_handler + profile +'\\Images')
```

```
        out = path_handler + profile +'\\Images'
        try:
            wget.download(data,out)
        except:
            print("No hay url")
    else:
        url = data['profile_image_url']
        print(url)
        if (os.path.exists(path_handler + profile) == False):
            os.mkdir(path_handler + profile)
        if(os.path.exists(path_handler + profile +'\\Profile') ==
False):
            os.mkdir(path_handler + profile +'\\Profile')
        out = path_handler + profile +'\\Profile'
        try:
            wget.download(url,out)
        except:
            print("No hay url")
def Lookup_twt(username): #Busca la información del perfil
    nest_asyncio.apply()
    c = twint.Config()
    c.Username = username
    c.Store_json = True
    c.User_full = True
    c.Output = path_handler + username
    twint.run.Lookup(c)
    path = c.Output + '\\' + 'users.json'
    file = open(path,'r',errors = 'ignore',encoding="utf-8")
    for i in file:
        json_parse = json.loads(i)
    image_download(json_parse,username,2)
    output = json2html.convert(json = json_parse)
    file.close()
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    api = tweepy.API(auth)
    friends = []
    followers = []
    user_list = []
    for user in tweepy.Cursor(api.get_friends,screen_name =
username,count = 20).items():
        if (len(friends) >= 10):
            break
        friends.append([user.screen_name,user.location])
        print(friends)
        print('friend: ' + str(len(friends)))
    for user in tweepy.Cursor(api.get_followers, screen_name =
username,count = 20).items():
        if (len(followers) >= 10):
```

```
            break
        followers.append([user.screen_name,user.location])
        print(followers)
        print('follower: ' + user.screen_name)
    api.home_timeline()
    information_profile(output,username,friends,followers)
def Twitter_scraper(username,n_tweets,output):
    start_time = time.time()
    nest_asyncio.apply()
    c = twint.Config()
    c.Username = username
    c.Limit = n_tweets
    c.Store_json = True
    c.Output = path_handler + username
    c.User_full = True
    twint.run.Search(c)
    Lookup_twt(username)
    path = c.Output + '\\' + 'tweets.json'
    file = open(path,'r',errors = 'ignore',encoding="utf-8")
    tweets_lists = []
    for i in file:
        json_parse = json.loads(i)
        del json_parse['timezone']
        del json_parse['language']
        del json_parse['quote_url']
        del json_parse['near']
        del json_parse['geo']
        del json_parse['source']
        del json_parse['translate']
        del json_parse['trans_src']
        del json_parse['trans_dest']
        for j in json_parse['photos']:
            mode = 1
            image_download(j,username,mode)
        output = json2html.convert(json = json_parse)
        tweets_lists.append(output)
    file.close()
    os.remove(path_handler + username + '\\tweets.json')
    soup_parse(tweets_lists,username)
    elapsed_time = time.time()
    total_time = elapsed_time - start_time
    total_time_minute = total_time / 60.0
    timestr = time.strftime("%Y-%m-%d-%H-%M-%S")
    file_posted = 'Data_extracted_' + username + timestr +'.txt'
    if (os.path.exists(path_handler + username) == False):
            os.mkdir(path_handler + username)
    if(os.path.exists(path_handler + username + '\\Information')
== False):
        os.mkdir(path_handler + username + '\\Information')
```

```
    file = open(path_handler + username +'\\Information\\' +
file_posted,'w',encoding="utf-8")
    message = 'Posts Extracted\n' + str(len(tweets_lists)) + '\n'
+ '\n' +'Time elapsed in minutes\n' + str(total_time_minute) +'\n'
+'Timestamp\n' + timestr+'\n'
    file.write(message)
    file.close()
```

Tiempos SCT

Planificación	Búsqueda de información	Análisis	Desarrollo	Edición	**Total**
5hr	11hr	9hr	26hr	4 hr	55 hr