



Universidad Nacional Autónoma de México

Facultad de Ingeniería

División de Ingeniería Mecánica e Industrial

Servicio Social



*Gestión y administración de proyectos en el laboratorio  
de inteligencia artificial*

---

Nombre del alumno:

Altamirano González David

Número de cuenta:

316295158

Periodo: del 10 de octubre del 2023 al 10 de abril del 2024

Supervisor: Dr. Ismael Everardo Barcenas Patiño

Responsable: Dra. Rocío Aldeco Pérez

## **Reporte Final de Servicio Social en el Laboratorio de Inteligencia Artificial**

### **OBJETIVO**

- Obtener información geográfica por medio de las bases de datos de redes sociales que esté relacionada con el COVID-19 para deducir la perseverancia o contagios de dicha enfermedad.

### **INTRODUCCIÓN**

En los últimos años han habido sucesos que han cambiado drásticamente el orden mundial, tanto en la forma de vivir como en la creación de nuevas tecnologías. El COVID-19 ha sido un suceso sin precedentes que dejó secuelas en la humanidad y por el nivel de su magnitud, es importante estudiarlo desde diferentes ámbitos.

Las nuevas tecnologías como la inteligencia artificial, que también ha implicado un cambio trascendental, nos proporcionan herramientas que nos permiten abordar todo tipo de problemas de una manera más rápida, eficiente y que nos permite tener nuevos alcances. En este proyecto implementamos dichas herramientas para estudiar el COVID-19 por medio de información obtenida de redes sociales, la cual nos permite obtener, directa o indirectamente la ubicación de las personas con una posibilidad de que se siga esparciendo dicha enfermedad. Esto nos permite comprender parcialmente el nivel de contagio y la vigencia que tiene después de 4 años desde la declaración de pandemia por parte de la ONU.

### **DESARROLLO**

Para obtener los datos necesarios para realizar el estudio propuesto en el proyecto, se aplicaron diferentes alternativas para la extracción de información, siendo la primera alternativa, la API de Twitter.

- API Twitter

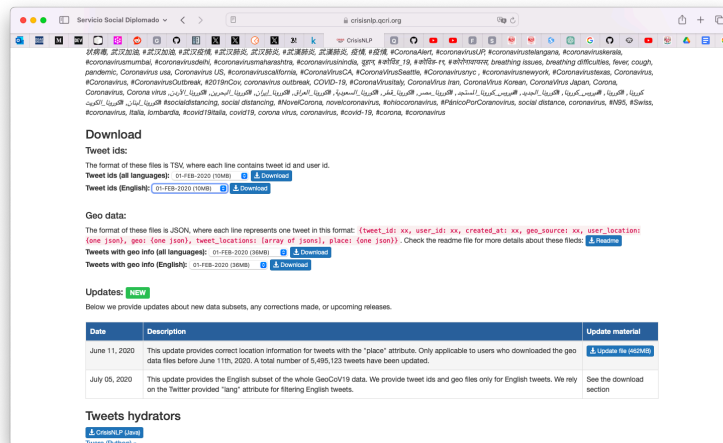
Es importante comentar que en 2022 Elon Musk compró Twitter y lo renombró como "X". Esto es un suceso que cambió directamente la dinámica de conexión a la base de datos de Twitter por medio de su API, lo cual, limitó drásticamente la obtención de información, pues para tener acceso, se ha impuesto un pago mensual para poder hacer uso de la API y aún con dicho pago, se tienen restricciones como la cantidad de tweets por mes, el tipo de información, etc. Es por eso que este método no fue útil para la obtención de información, pero al menos fue un primer acercamiento a la extracción de datos. Esto nos llevó a buscar alternativas que nos permitieran obtener tweets que tuvieran al menos información del usuario, el texto y la ubicación de ser posible.

- Dataset de Hamad Bin Khalifa University

Como alternativa, utilizamos un dataset de la Universidad Hamad Bin Khalifa que realizó un estudio similar en 2020, recolectando aproximadamente 524 millones de tweets con múltiples lenguajes. Dichos tweets están contenidos en documentos JSON que se pueden descargar directamente de la página del dataset, y con dicha información, se realizó un programa que extrajera el id del usuario, la información de la ubicación tanto del usuario como del tweet y se generó un link que nos dirija al tweet en cualquier buscador para checar su texto. A continuación se presenta paso a paso la extracción de esta información.

1. Descarga de documento JSON de la página <https://crisisnlp.qcri.org/covid19> :

En la página, nos dirigimos a Download y damos click en "Tweet ID (All languages)", ya que en estos documentos extraemos información preferentemente en español.



Después descargamos cualquier documento de la lista.

## Download

### Tweet ids:

The format of these files is TSV, where each line contains tweet id and user id.

**Tweet ids (all languages)** ✓ 01-FEB-2020 (10MB) [Download](#)

**Tweet ids (English):** 01-FEB-2020 (10MB) [Download](#)

**Geo data:**

02-FEB-2020 (21MB) [Download](#)

03-FEB-2020 (18MB) [Download](#)

04-FEB-2020 (16MB) [Download](#)

05-FEB-2020 (14MB) [Download](#)

06-FEB-2020 (15MB) [Download](#)

Para poder limpiar toda la información del JSON, ejecutamos el programa “Lectura\_de\_datos\_7\_prioridades.py”, el cual extrae el documento JSON y selecciona únicamente los apartados de ID, Geosource, País, Estado, Condado, Ciudad y genera un Tweet Link. Todo esto ordenado por prioridades dependiendo del Geosource ya que existen diferentes tipos de ubicación dependiendo de cómo se haya obtenido la ubicación, ya que puede ser por coordinates, place, user\_location, o tweet\_text; siendo la de mayor fidelidad la de coordinates y siguiendo el orden de manera descendente. Al correr el programa, arroja como documento final un csv que nos permite manipular la información de manera más sencilla en programas como Excel.

Programa:

```

Lectura_datos_7_prioridades.py > ...
1  import json
2  import csv
3
4  archivo_principal = 'locations-01Febrero.json'
5  limite_archivos = 666575
6  archivos_procesados = 0
7  archivo_csv = 'output4.csv'
8
9  def generate_tweet_link(user_id, tweet_id):
10     # Verificar si hay un user_id antes de generar el enlace
11     if user_id:
12         return f'https://twitter.com/{user_id}/status/{tweet_id}'
13     return f'https://twitter.com/status/{tweet_id}'
14
15  def geo_source_priority(geo_source):
16     # Asignar prioridades a las fuentes de geo
17     priorities = {'coordinates': 0, 'place': 1, 'user_location': 2, 'tweet_text': 3}
18     return priorities.get(geo_source, float('inf'))
19
20  with open(archivo_csv, 'w', newline='', encoding='utf-8') as archivo_csv:
21     csv_writer = csv.writer(archivo_csv)
22
23     # Encabezados para el archivo CSV
24     encabezados = ['Tweet ID', 'Geo Source', 'Pais', 'Estado', 'Condado', 'Ciudad', 'Tweet Link']
25     csv_writer.writerow(encabezados)
26
27     filas = []
28
29     with open(archivo_principal, 'r') as archivo_principal:
30         for linea in archivo_principal:
31             try:
32                 datos = json.loads(linea)
33                 user_id = datos.get('user_id', '')
34                 tweet_id = datos.get('tweet_id')
35                 geo_source = datos.get('geo_source', '')
36
37                 fila = [tweet_id, geo_source]
38
39                 if geo_source == 'coordinates':
40                     # Tomar la información de las coordenadas
41                     geo = datos.get('geo', {})
42                     fila.extend([
43                         geo.get('country_code', ''),
44                         geo.get('state', ''),
45                         geo.get('county', ''),
46                         geo.get('city', ''),
47                     ])
48                 elif geo_source == 'place':
49                     # Tomar la información de place
50                     place = datos.get('place', {})
51                     fila.extend([
52                         place.get('country_code', ''),
53                         place.get('state', ''),
54                         place.get('county', ''),
55                         place.get('city', ''),
56                     ])
57                 elif geo_source == 'user_location':
58                     user_location = datos.get('user_location', {})
59                     fila.extend([
60                         user_location.get('country_code', ''),
61                         user_location.get('state', ''),
62                         user_location.get('county', ''),
63                         user_location.get('city', ''),
64                     ])
65                 elif geo_source == 'tweet_text':
66                     # Tomar la información de tweet_locations
67                     tweet_locations = datos.get('tweet_locations', [{}])[0]
68                     fila.extend([
69                         tweet_locations.get('country_code', ''),
70                         tweet_locations.get('state', ''),
71                         tweet_locations.get('county', ''),
72                         tweet_locations.get('city', ''),
73                     ])
74                 else:
75                     # Manejar otros casos, si es necesario
76                     fila.extend([''] * 4)
77
78                 tweet_link = generate_tweet_link(user_id, tweet_id)
79                 fila.append(tweet_link)
80
81                 filas.append(fila)
82                 archivos_procesados += 1
83
84                 if archivos_procesados >= limite_archivos:
85                     break
86
87             except json.decoder.JSONDecodeError as e:
88                 print(f"Error al decodificar JSON en la línea {archivos_procesados + 1}: {e}")
89
90     # Ordenar las filas por prioridad de geo_source
91     filas_ordenadas = sorted(filas, key=lambda x: geo_source_priority(x[1]))
92
93     # Escribir las filas ordenadas al archivo CSV
94     csv_writer.writerows(filas_ordenadas)
95
96     print(f"La información ha sido guardada y ordenada por prioridades en {archivo_csv}")
97

```

Al correr el programa obtenemos el siguiente documento “output.csv”:

```
selenium_extraccion.py  limpieza_csv_tweet_text.py  output4.csv x  Lectura_datos_7_prioridades.py  datos_combinados.csv  ...
output4.csv
1 Tweet ID,Geo Source,País,Estado,Condado,Ciudad,Tweet Link
2 1223740277430591488,coordinates,qa,Doha,,Doha,https://twitter.com/144906244/status/1223740277430591488
3 1223525783471370243,coordinates,au,Western Australia,,https://twitter.com/14928639/status/1223525783471370243
4 1223474266169102337,coordinates,pl,Masovian Voivodeship,Warsaw,Warsaw,https://twitter.com/1316862031/status/1223474266169102337
5 1223487344545869824,coordinates,th,Bangkok,,Bangkok,https://twitter.com/15692330/status/1223487344545869824
6 1223502124799016960,coordinates,th,Bangkok,,Bangkok,https://twitter.com/208056970/status/1223502124799016960
7 1223412331729817601,coordinates,us,Florida,Broward County,https://twitter.com/20415334/status/1223412331729817601
8 1223422078403207170,coordinates,bd,Dhaka Division,,Dhaka,https://twitter.com/369981979/status/1223422078403207170
9 122366265359183744,coordinates,de,North Rhine-Westphalia,Rhein-Kreis Neuss,Neuss,https://twitter.com/15797413/status/122366265359183744
10 1223663962513760257,coordinates,us,Pennsylvania,Philadelphia County,Philadelphia,https://twitter.com/2811634514/status/1223663962513760257
11 1223674932766433286,coordinates,ng,Oyo,Ibadan North,https://twitter.com/803994563249664002/status/1223674932766433286
12 1223458629535371264,coordinates,us,Georgia,Fulton County,Johns Creek,https://twitter.com/38550012/status/1223458629535371264
13 1223450749595803651,coordinates,th,Bangkok,,Bangkok,https://twitter.com/83784822/status/1223450749595803651
14 1223587669168611329,coordinates,cn,Jiangsu,Hanjiang District,Yangzhou City,https://twitter.com/1288410030/status/1223587669168611329
15 1223592182579372032,coordinates,pl,Masovian Voivodeship,Warsaw,Warsaw,https://twitter.com/1316862031/status/1223592182579372032
16 122366640933421056,coordinates,us,California,Riverside County,https://twitter.com/27352034/status/122366640933421056
17 1223607938603978753,coordinates,us,Florida,Lee County,Fort Myers,https://twitter.com/45474869/status/1223607938603978753
18 1223464308975263747,coordinates,kr,Gyeongbuk,,Gimcheon-si,https://twitter.com/1114286820/status/1223464308975263747
19 1223620870196948000,coordinates,pl,Masovian Voivodeship,Warsaw,Warsaw,https://twitter.com/1316862031/status/1223620870196948000
20 1223631654297784320,coordinates,us,New York,New York County,New York,https://twitter.com/766533596051968000/status/1223631654297784320
21 1223650236180722176,coordinates,gb,England,,London,https://twitter.com/138418838/status/1223650236180722176
22 1223410659720933376,coordinates,cn,Shanghai,Huangpu District,Huangpu,https://twitter.com/356484644/status/1223410659720933376
23 1223509678577147904,coordinates,pl,Masovian Voivodeship,Warsaw,Warsaw,https://twitter.com/1316862031/status/1223509678577147904
24 1223548411946758146,coordinates,us,Florida,Duval County,Jacksonville,https://twitter.com/1164502039600271360/status/1223548411946758146
25 1223703744887607296,coordinates,us,Louisiana,East Baton Rouge Parish,Baton Rouge,https://twitter.com/738994410/status/1223703744887607296
26 1223523708591058752,coordinates,cn,Hubei,Jiangnan District,万松街道,https://twitter.com/1094921571750182912/status/1223523708591058752
27 1223556789066784768,coordinates,lk,Central Province,,Dambulla,https://twitter.com/871744007193239552/status/1223556789066784768
28 1223459266981396480,coordinates,cn,Hubei,Hongshan District,路南街道,https://twitter.com/997050657856143360/status/1223459266981396480
29 1223603571784142848,coordinates,kr,Gyeonggi,Bundang-gu,Seongnam-si,https://twitter.com/16624930/status/1223603571784142848
```

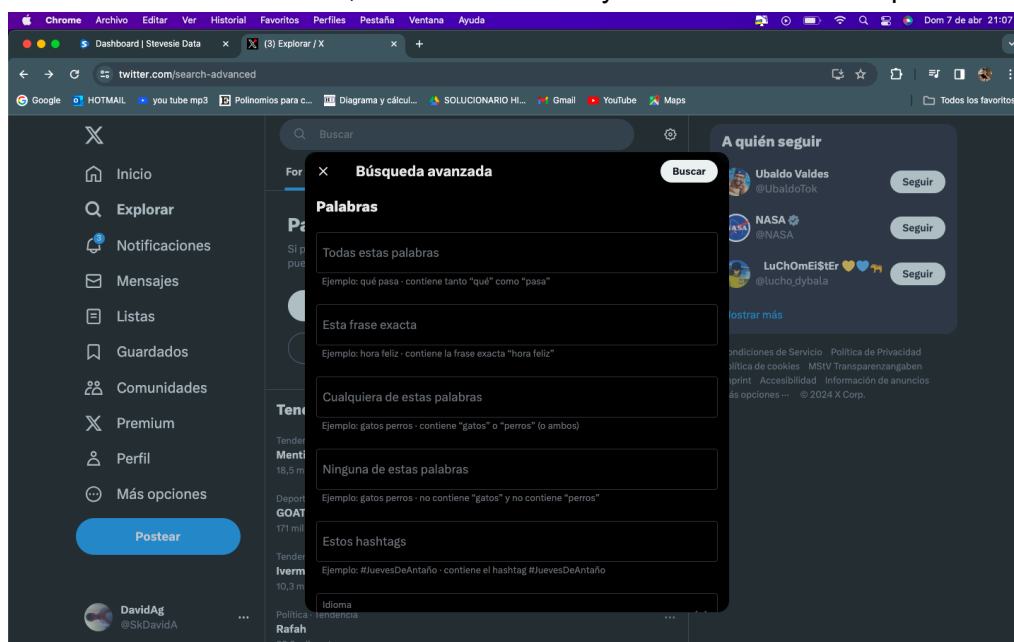
Con esta información podemos buscar el tweet en internet y realizar un estudio de las ubicaciones obtenidas.

- Scrapping de Tweeter (X) en Chrome

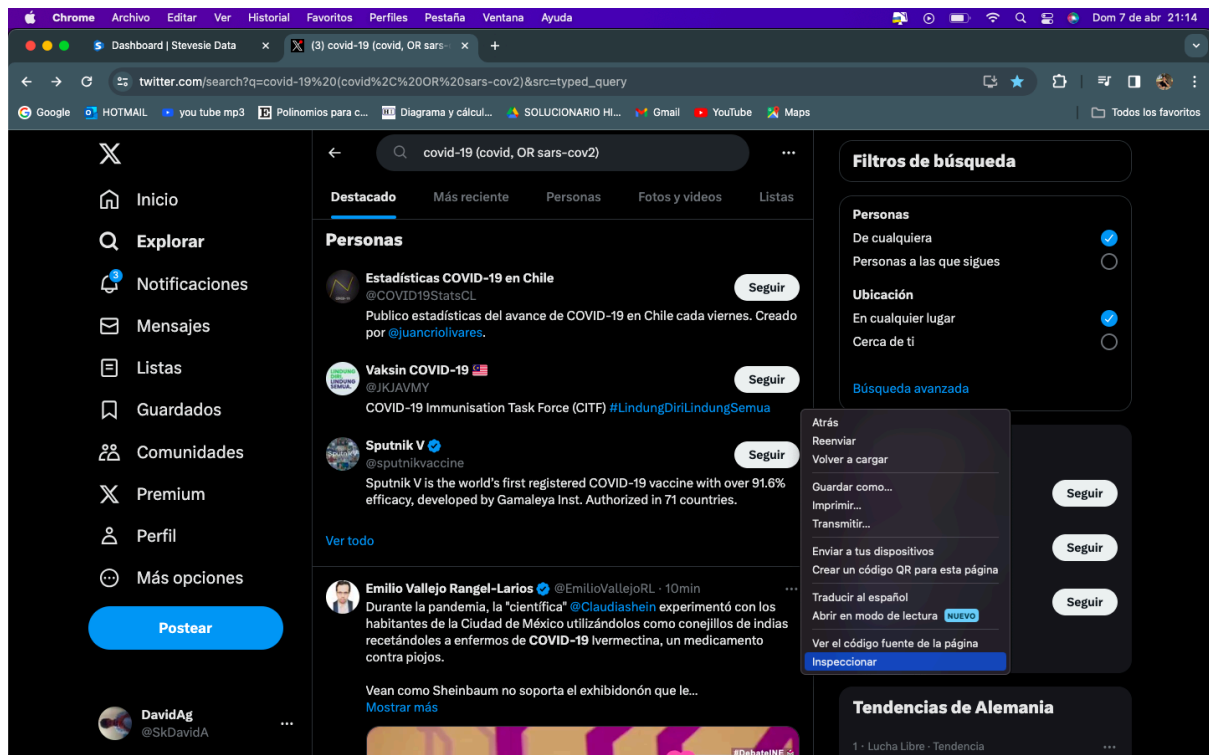
Este último método de extracción de información es un tipo de scrapping manual. Se optó por buscar alternativas que nos permitieran recolectar la información necesaria para alcanzar nuestros objetivos sin la necesidad de utilizar la API de Twitter debido a las limitantes y el poco alcance que nos permitía.

En este método únicamente se necesita tener una cuenta de twitter para poder tener acceso al buscador del mismo, el buscador Chrome y la página Stevesie Data; a continuación se describe el proceso a seguir para extraer la información:

1. Primeramente en Chrome, buscamos Twitter y realizamos una Búsqueda Avanzada.

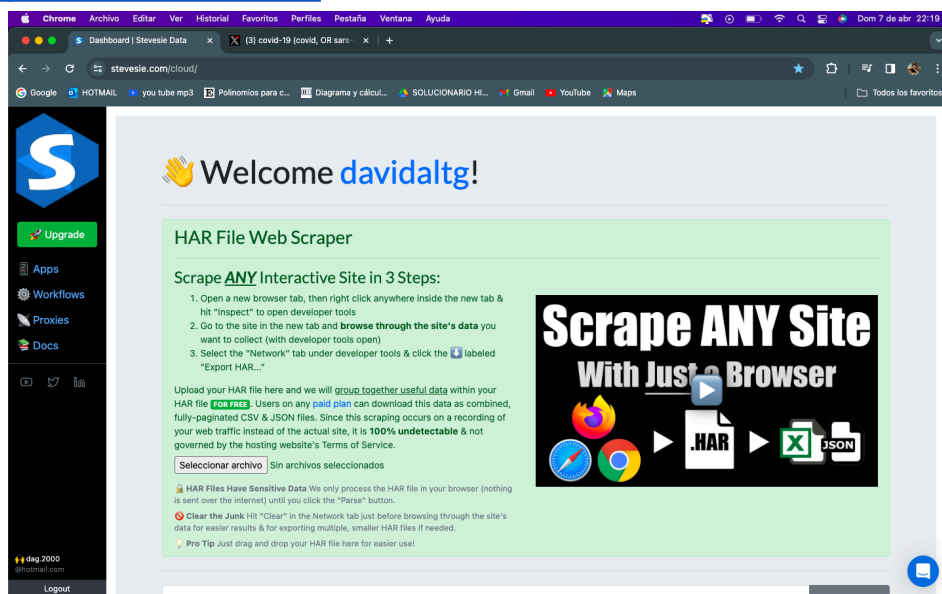


En el buscador escribimos las palabras clave a buscar, en este caso: COVID-19, COVID, SARS COV 2,etc. Dentro del buscador de Twitter, damos click derecho en cualquier espacio de la página y en las opciones, damos click en la sección de “Inspección”.



Al dar click, se desplegará una ventana en la parte inferior que registrará toda la información que se presentó en la página. Hecho esto, procederemos a scrollear la página para que se registre toda la información; posteriormente en el apartado de inspección, daremos click en el apartado de “Network” y daremos click en el botón rojo para detener la grabación de la información. Finalmente, daremos click en el botón de “Export HAR”, que nos descargará un documento en formato HAR con toda la información grabada.

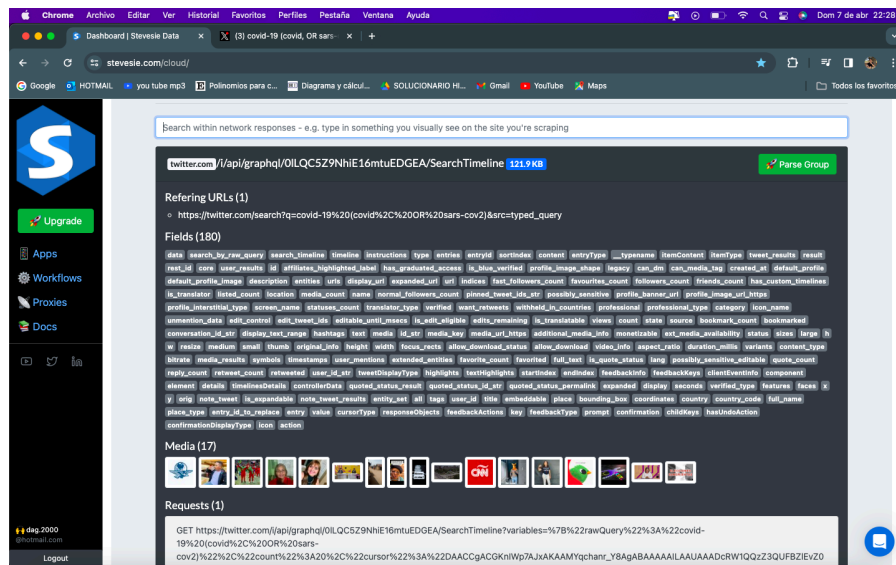
2. Para poder hacer uso de este documento, necesitamos convertir el documento HAR a un formato más amigable que nos permita hacer una selección óptima de la información. Para lograr esto, usaremos la página Stevesie Data <https://stevesie.com/cloud/>.



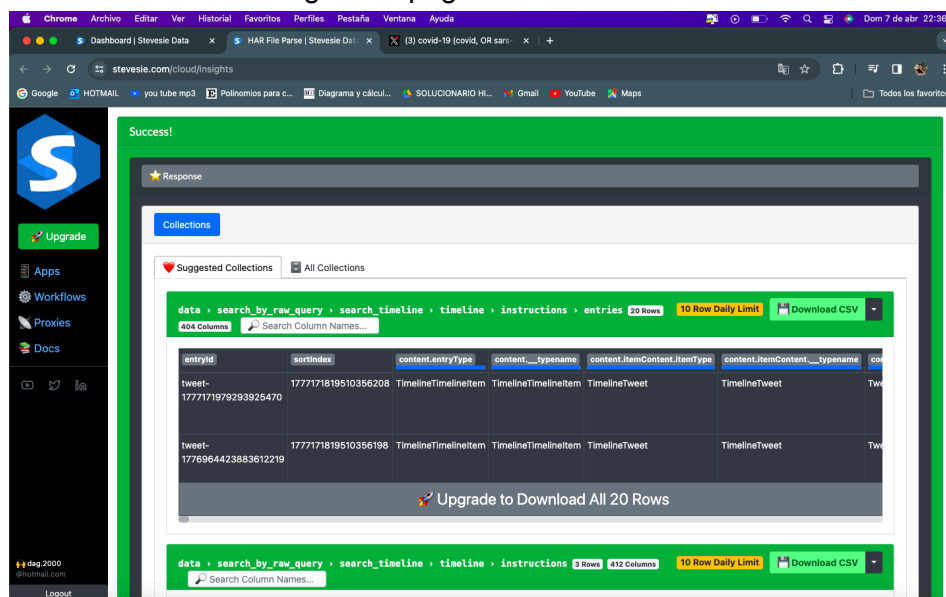
Esta página nos permitirá convertir el documento HAR en un documento JSON, el cuál, es mucho más amigable para la lectura de datos y su debida extracción.



Lo siguiente es arrastrar el documento HAR descargado a la página y dejar que se procese la información.



Debido a que el documento tiene demasiada información y no tiene un patrón fácil de interpretar, necesitamos una herramienta que proporcione la misma página, que convierta en automático el JSON en un documento CSV con toda la información ordenada por columnas, lo cual, simplifica el trabajo. Para esto, damos click en el botón verde “Parse Group” y nos direcciona a la siguiente página :



Ahora, damos en “Download CSV”. Automáticamente nos descargará el documento. El único punto de mejora respecto a esta metodología es que, esta página solo nos permite descargar el documento con 10 filas de información cada 24 hrs, si deseamos descargar más tenemos que pagar una membresía de \$89.99 USD. Ahora, con este documento descargado lo añadimos a una carpeta donde estén todos los archivos de la colecta de información que vayamos haciendo con el transcurso de los días. Esto nos servirá para que, en el siguiente programa de Python podamos juntar toda esa información en un solo documento CSV con la información limpia y clara. El siguiente programa, selecciona la siguiente información:

- Id de usuario
- Id del tweet
- Texto del tweet
- Generación del link del tweet original
- Ubicación geográfica de tweet

```

1  limpieza_csv_tweet_text.py > ...
2  1 import os
3  2 import pandas as pd
4  3 from decimal import Decimal
5  4
6  5 def generate_tweet_link(user_id, tweet_id):
7  6     return f"https://twitter.com/{user_id}/status/{tweet_id}"
8  7
9  8 def to_str(num):
10  9     return str(Decimal(num).to_integral_value())
11  10
12  11 def merge_csv_files(folder_path, output_file):
13  12     # Lista de nombres de archivos CSV en la carpeta
14  13     csv_files = [file for file in os.listdir(folder_path) if file.endswith('.csv')]
15  14
16  15     # Inicializar una lista para almacenar los datos combinados
17  16     combined_data = []
18  17
19  18     # Iterar sobre cada archivo CSV
20  19     for file in csv_files:
21  20         # Leer el archivo CSV
22  21         file_path = os.path.join(folder_path, file)
23  22         df = pd.read_csv(file_path)
24  23
25  24         # Extraer las columnas requeridas
26  25         user_ids = df['data.search_by_raw_query.search_timeline.timeline.instructions.entries.content.itemContent.tweet_results.re
27  26         tweet_ids = df['data.search_by_raw_query.search_timeline.timeline.instructions.entries.content.itemContent.tweet_results.re
28  27         texts = df['data.search_by_raw_query.search_timeline.timeline.instructions.entries.content.itemContent.tweet_results.resu
29  28         locations = df['data.search_by_raw_query.search_timeline.timeline.instructions.entries.content.itemContent.tweet_results.r
30  29
31  30         # Convertir los números a cadenas de texto
32  31         user_ids_str = user_ids.apply(to_str)
33  32         tweet_ids_str = tweet_ids.apply(to_str)
34  33
35  34         # Generar los enlaces para cada tweet
36  35         tweet_links = [generate_tweet_link(user_id, tweet_id) for user_id, tweet_id in zip(user_ids_str, tweet_ids_str)]
37  36
38  37         # Combinar los datos en una lista de tuplas
39  38         combined_data.extend(zip(user_ids_str, tweet_ids_str, texts, locations, tweet_links))
40  39
41  40     # Crear un DataFrame con los datos combinados
42  41     combined_df = pd.DataFrame(combined_data, columns=['User_ID', 'Tweet_ID', 'Texto', 'Ubicación', 'Enlace del Tweet Original'])
43  42
44  43     # Guardar el DataFrame combinado en un nuevo archivo CSV
45  44     combined_df.to_csv(output_file, index=False)
46  45     print(f"Datos combinados guardados en {output_file}")
47  46
48  47     # Carpeta que contiene los archivos CSV
49  48     folder_path = "Textos tweets Dashboard"
50  49
51  50     # Nombre del archivo de salida
52  51     output_file = "datos_combinados.csv"
53  52
54  53     # Llamar a la función para combinar los archivos CSV
55  54     merge_csv_files(folder_path, output_file)

```

Al correr el programa tendremos la información organizada en columnas y en el documento final “datos combinados.csv”:

	A	B	C	D	E
1	User ID	Tweet ID	Texto	Ubicación	Enlace del Tweet Original
2	2970779998	72077682035487800	Un día los graves errores del gobierno D. Bogotá, D. C., Colombia		<a href="https://twitter.com/2970779998/status/72077682035487800">https://twitter.com/2970779998/status/72077682035487800</a>
3	150158259991734272	76873095385127036	Gracias!!! con el Doble del Proyecto Virella "La Esfera, Espo"		<a href="https://twitter.com/150158259991734272/status/76873095385127036">https://twitter.com/150158259991734272/status/76873095385127036</a>
4	15047377466378462	76891702903466608	TOODS A FORTALECER Mientras la gente muere y ya estaba ausente, los listos se		<a href="https://twitter.com/15047377466378462/status/76891702903466608">https://twitter.com/15047377466378462/status/76891702903466608</a>
5	150477175725312032	76891702903466608	Yo los tengo. Seguramente tiene la gente a poder por ejemplo, se en cada		<a href="https://twitter.com/150477175725312032/status/76891702903466608">https://twitter.com/150477175725312032/status/76891702903466608</a>
6	16335263067863268	76891702903466608	Yo los tengo. Seguramente tiene la gente a poder por ejemplo, se en cada		<a href="https://twitter.com/16335263067863268/status/76891702903466608">https://twitter.com/16335263067863268/status/76891702903466608</a>
7	848407419	76885603539032264	¡Hola! Convidando a todos los ciudadanos de Bogotá, D. C., Colombia		<a href="https://twitter.com/848407419/status/76885603539032264">https://twitter.com/848407419/status/76885603539032264</a>
8	151198500015631872	76891702903466608	El presidente de la División San Fernando, el COVID		<a href="https://twitter.com/151198500015631872/status/76891702903466608">https://twitter.com/151198500015631872/status/76891702903466608</a>
9	150158259991734272	76891702903466608	Suplementos ingresos por desastres de COVID		<a href="https://twitter.com/150158259991734272/status/76891702903466608">https://twitter.com/150158259991734272/status/76891702903466608</a>
10	1490190903	76891702903466608	Con el Estado, con el Gbo y el Shania, Egipto		<a href="https://twitter.com/1490190903/status/76891702903466608">https://twitter.com/1490190903/status/76891702903466608</a>
11	259800403	76880511811035780	Una persona con función en el país ha habido ordenado por COVID		<a href="https://twitter.com/259800403/status/76880511811035780">https://twitter.com/259800403/status/76880511811035780</a>
12	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
13	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
14	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
15	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
16	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
17	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
18	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
19	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
20	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
21	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
22	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
23	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
24	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
25	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
26	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
27	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
28	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
29	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
30	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
31	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
32	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
33	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
34	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
35	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
36	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
37	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
38	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
39	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
40	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
41	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
42	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
43	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
44	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
45	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
46	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
47	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
48	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
49	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
50	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
51	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
52	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
53	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
54	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
55	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
56	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
57	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
58	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
59	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
60	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
61	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
62	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
63	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
64	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
65	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
66	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
67	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
68	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
69	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
70	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
71	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
72	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
73	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
74	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
75	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
76	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
77	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
78	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
79	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
80	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
81	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
82	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
83	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
84	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
85	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
86	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
87	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
88	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
89	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
90	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
91	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
92	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
93	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
94	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
95	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
96	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
97	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
98	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
99	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>
100	NaN	NaN	NaN	NaN	<a href="https://twitter.com/NaN/NaN/NaN">https://twitter.com/NaN/NaN/NaN</a>




Finalmente, como resultado obtenemos un único documento CSV con toda la información necesaria para mapear la ubicación de los tweets que tienen relación con la enfermedad de COVID 19.

## CONCLUSIÓN

El COVID-19 trajo consigo muchos cambios en todos los ámbitos, los cuales son importantes seguir estudiando, y más con las nuevas herramientas tecnológicas que tenemos al alcance el día de hoy como la inteligencia artificial, las API's, los lenguajes de programación, etc. En este caso de estudio, gracias a la masiva información que se genera día con día en las redes sociales, la implementación de dichas herramientas tecnológicas para poder encontrar las ubicaciones de los tweets, nos hace ver que es posible detectar la actividad de las personas respecto a esta enfermedad.

Pese a las limitaciones que tuvimos debido a los cambios internos en Twitter y su API para extraer información, pudimos tener un acercamiento muy bueno con la información de otros estudios muy similares y con el método del scrapping manual. De tal forma que obtuvimos la información geográfica necesaria tanto del texto como del tweet mismo, lo cual varía dependiendo de cada usuario. La cantidad de información que se vaya agregando a la última metodología de extracción, permitirá tener a largo plazo un estudio más completo y con un mapeo geográfico con más objetividad.

## REFERENCIAS:

- Akbarzadeh, N. (2023). Scraping tweets without Twitter API for FREE. Recuperado de: <https://dev.to/iw4p/scraping-tweets-without-twitter-api-and-free-5g9c> .
- Bettenbuk, Z. (2022). How to Scrape Twitter Data Using Python Without Using Twitter's API. Recuperado de: <https://www.scraperaapi.com/blog/scrape-twitter-data/>.
- Ritesh, V. (2021). COVID-19  Vaccine Tweets. Recuperado de: <https://www.kaggle.com/datasets/ritesh2000/covid19-vaccine-tweets> .
- -. (2023). Tweepy Documentation. Recuperado de: <https://docs.tweepy.org/en/stable/index.html> .
- -. (2020). GeoCoV19: A Dataset of Hundreds of Millions of Multilingual COVID-19 Tweets with Location Information. Recuperado de: <https://crisisnlp.qcri.org/covid19> .
- -. (2024). Web Scraping X (Formerly Twitter) Data with Python & Selenium (Updated Article). Recuperado de: <https://www.scrapingdog.com/blog/scrape-twitter/> .