

**Benemérita Universidad
Autónoma de Puebla**

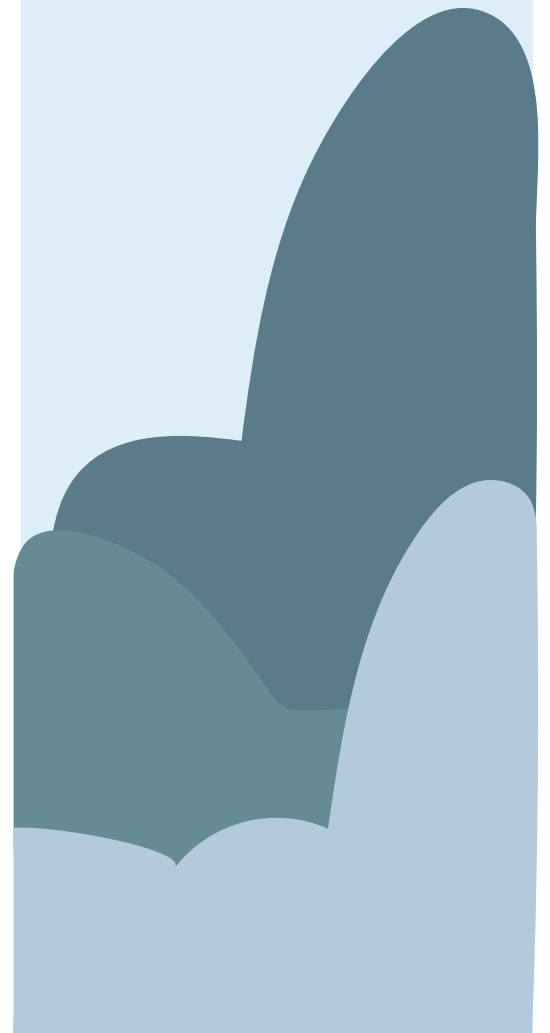
Facultad de Ciencias de la Computación
Ingeniería en Ciencias de la Computación

MANUAL COMPLETO SOBRE EL USO DE:

*"AL Twitter Sentiment Analysis for Data
Science."*

TÉCNICAS DE INTELIGENCIA ARTIFICIAL

Joshua Mayorga Maca
201930699
Otoño 2022



AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

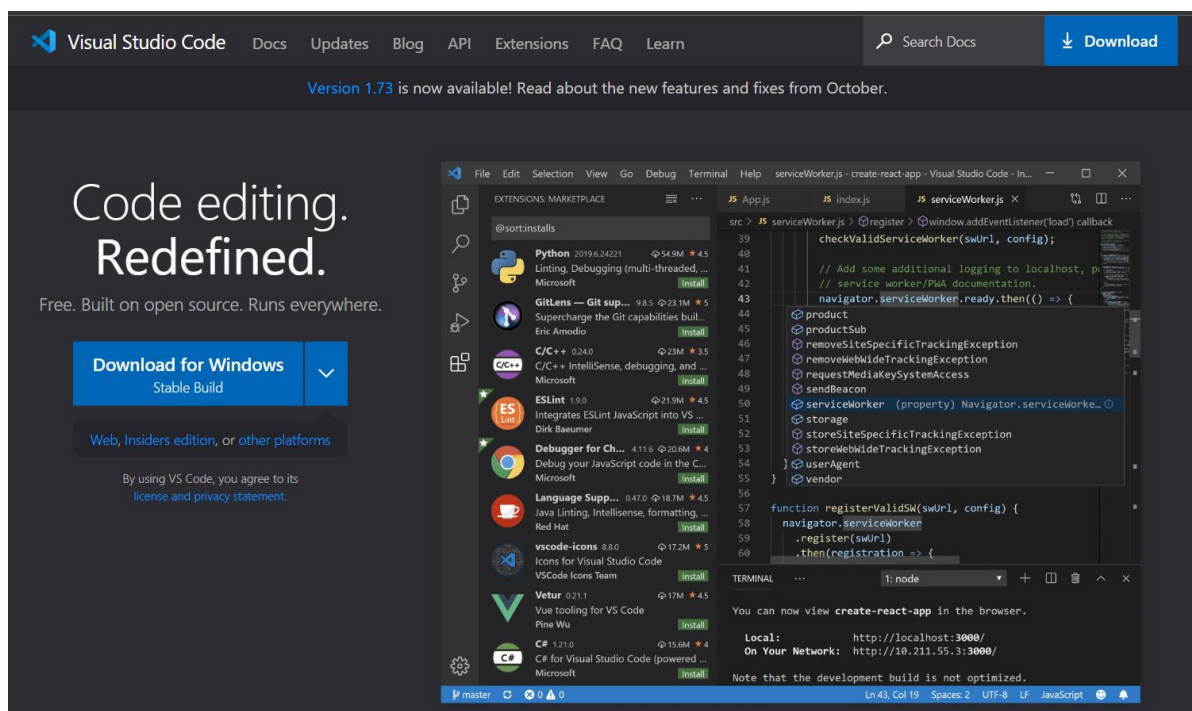
CREACIÓN DE LA IDEA:

El programa en cuestión se crea en base a un proyecto dado en la materia de “Técnicas de Inteligencia Artificial” de la Benemérita Universidad Autónoma de Puebla. Proyecto el cual amplió mi conocimiento acerca de las herramientas que se tienen hoy en día para realizar el análisis de sentimientos en Python, es por esto que se decide crear una herramienta práctica como método de uso para demostrar una mejor aplicación que practique todo lo necesario para así imprimir datos concretos y sólidos fácilmente.

COMO HACER FUNCINAR EL PROGRAMA:

Lo primero que se necesita es tener un programa que pueda usar Python. En lo personal se recomienda que sea **Visual Studio Code** el cual puedes descargar desde este enlace:

<https://code.visualstudio.com/>

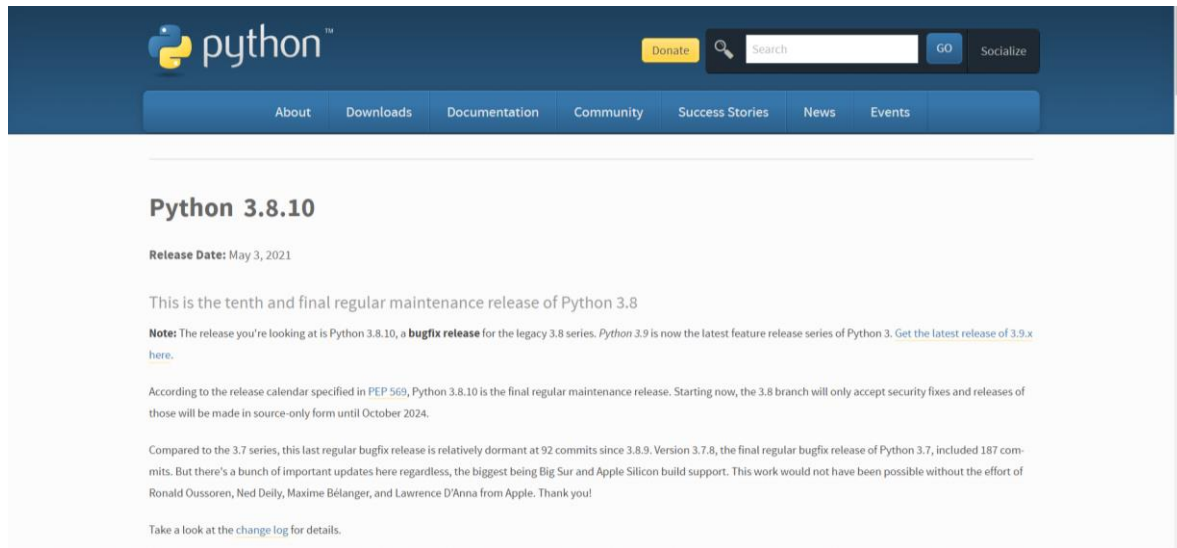


Aclarar que este programa es totalmente funcional para Windows, si se requiere ejecutar en Linux verificar los comandos sustitutos necesarios.

Una vez instalado el software encargado de la ejecución del lenguaje de programación lo siguiente que se hace para ejecutar el programa es tener instalada la versión de **Python 3.8.10**, el cual se puede encontrar en el siguiente enlace:

AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

<https://www.python.org/downloads/release/python-3810/>



Nota: Al momento de instalar Python es necesario revisar que se encuentre habilitada la opción de **instalar pip**.

Lo siguiente es instalar todas las bibliotecas siguientes, para hacer esto podemos hacerlo ya sea dentro de la línea de comandos del **command prompt de Windows** o desde el visual studio en el apartado de **terminal**:

Para hacer la instalación por command prompt ejecutamos toda la línea completa, en su defecto para ejecutar dentro de la terminal de python únicamente la parte en **rojo**.

PRIMERO: Tenemos que tener en cuenta la necesidad de actualizar el pip que viene por defecto en la versión de python.

Esto con la siguiente línea de comandos:

```
C:\Users\joshu>python -m pip install --upgrade pip
Requirement already satisfied: pip in d:\python\lib\site-packages (22.3)
```

Como se puede ver se actualizó a la versión 22.3.

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show pip
Name: pip
Version: 22.3
```

Instalar la versión 0.17.1 de textblob

`python -m pip install textblob==0.17.1`

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip install textblob
```

AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

Modulo agregado correctamente:

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show textblob
Name: textblob
Version: 0.17.1
```

Instalar la versión 3.6.2 de matplotlib

python -m pip install matplotlib==3.6.2

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip install matplotlib
```

Modulo agregado correctamente:

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show matplotlib
Name: matplotlib
Version: 3.6.2
```

Instalar la versión 1.8.2 de wordcloud

python -m pip install wordcloud==1.8.2

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip install wordcloud
```

Modulo agregado correctamente:

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show wordcloud
Name: wordcloud
Version: 1.8.2.2
```

Instalar la versión 0.7.0 de better_profanity

python -m pip install better_profanity==0.7.0

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip install better_profanity
```

Modulo agregado correctamente:

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show better_profanity
Name: better-profanity
Version: 0.7.0
```

Instalar mínimo la versión del 2022.10.31 de regex

python -m pip install regex

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip install regex
```

Modulo agregado correctamente:

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show regex
Name: regex
Version: 2022.10.31
```

AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

NOTA: si la versión de regex no llegara a funcionar probar con la siguiente línea de comandos:

```
python -m pip install black --no-binary=regex
```

```
PS D:\PROGRAMAS PY\UK\UK\PLN> pip install black --no-binary=regex
```

Instalar la versión 9.3.0 de Pillow

```
python -m pip install pillow==9.3.0
```

```
PS D:\PROGRAMAS PY\UK\UK\PLN> pip install Pillow
```

Modulo agregado correctamente:

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show pillow
Name: Pillow
Version: 9.3.0
```

NOTA: Actualmente Pillow no se encuentra en un funcionamiento correcto, si usted se encuentra con un problema de estos para poder arreglarlo en su defecto se procederá a hacer lo siguiente.

Probar las ejecuciones siguientes:

```
python -m pip install --no-index -f https://dist.plone.org/thirdparty/ -U PIL
```

```
python -m pip install pillow
```

```
python -m pip uninstall PIL
```

```
python -m pip uninstall pillow
```

```
python -m pip install PIL
```

Instalar la versión 1.23.4 de numpy

```
python -m pip install numpy==1.23.4
```

```
PS D:\PROGRAMAS PY\UK\UK\PLN> pip3 install numpy
```

Modulo agregado correctamente:

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show numpy
Name: numpy
Version: 1.23.4
```

AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

Instalar la versión 5.2.0 de configparser

python -m pip install configparser==5.2.0

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip install configparser==5.2.0
```

Modulo agregado correctamente:

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show configparser
Name: configparser
Version: 5.2.0
```

Instalar la versión 4.6.0 de tweepy

python -m pip install tweepy==4.6.0

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip install tweepy==4.6.0
```

Modulo agregado correctamente:

```
PS D:\PROGRAMAS PY\UK\UK\PLN\PLN> pip show tweepy
Name: tweepy
Version: 4.6.0
```

Teniendo todo esto anteriormente instalado no tendrías problemas al momento de intentar ejecutar el programa.

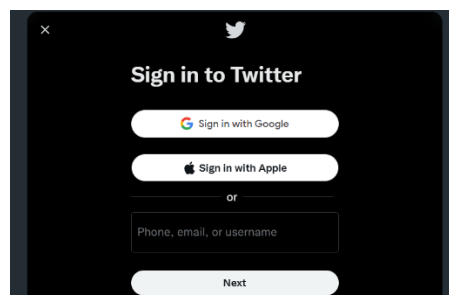
El siguiente elemento es imprescindible para la ejecución de la librería principal para la extracción de tweets con la herramienta de Tweepy.

CUENTA DEVELOPER DE TWITTER Y ARCHIVO CONFIG.INI:

Para poder hacer la ejecución se necesita si o si una cuenta developer de Twitter, esto lo puedes conseguir con el siguiente enlace:

<https://developer.twitter.com/en/apply-for-access>

Te pedirá hacer login con tu cuenta de Twitter normal existente o la creación de una nueva.



AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

Una vez creada la cuenta de Twitter developer se recomienda estrictamente aplicar para la cuenta elevada ya que con esta podrás gestionar arriba de 5 mil tweets por palabra sin tanto problema.

V2 Access Levels

Essential

With Essential access, you can now get access to Twitter API v2 quickly and for free!

- Retrieve 500,000 Tweets per month
- 1 Project per account
- 1 App environment per Project
- Limited access to standard v1.1 (**only media endpoints**)
- No access to premium v1.1, or enterprise

Elevated

With Elevated access, you can get free, additional access to endpoints and data, as well as additional App environments.

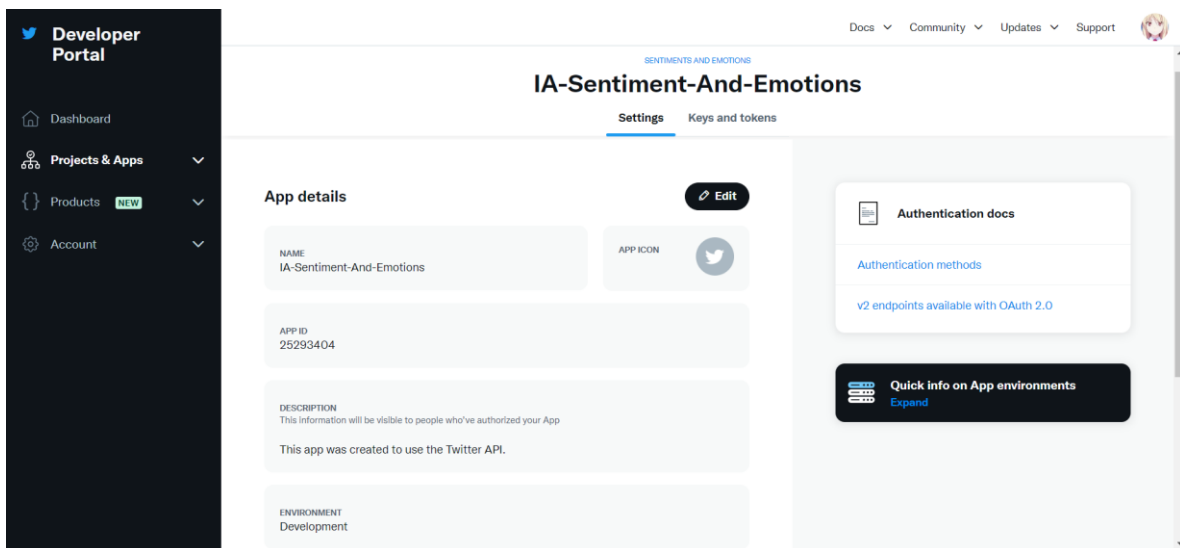
- Retrieve 2 million Tweets per month
- 1 Project per account
- 3 App environments per Project
- Access to standard v1.1, premium v1.1, and enterprise

Academic Research

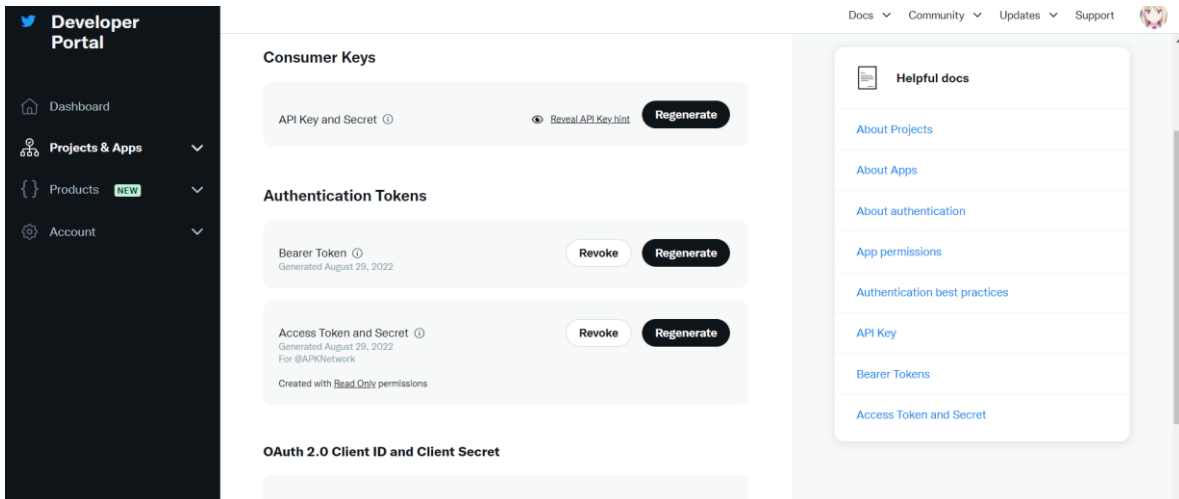
If you qualify for our Academic Research access level, you can get access to even more data and advanced search endpoints.

- Retrieve 10 million Tweets per month
- Access to full-archive search and full-archive Tweet counts
- Access to advanced search operators

Después de obtener el acceso al nivel 2 se procede a crear un proyecto, para eso es necesario consultar la guía de creación de proyectos de la cuenta de Twitter, teniendo ya creado el proyecto se verá de esta manera.



En la opción de **Keys and Tokens** podremos encontrar las claves que vamos a utilizar en el archivo **config.ini** dentro del proyecto.



Al abrir el archivo config.ini veremos lo siguiente:

```

Al_TwitterSAD.py • config.ini
config.ini
1 [twitter]
2
3 api_key =
4 api_key_secret =
5
6 access_token =
7 access_token_secret =
    
```

Las claves que se ocuparan se encuentran en los correspondientes apartados del proyecto de twitter developer, **api_key** y **api_key_token** en el apartado de **Consumer Keys** y en el apartado de **Authentication Tokens** podras encontrar las claves de **access_token** y **ccess_token_Secret**. Un ejemplo sería el siguiente:

```

config.ini
1 [twitter]
2
3 api_key = 1d22gRD455ed0UbIB02BCiLyOPko0Jo
4 api_key_secret = 8hYF665DMDdYSQyykMZVWmsezL0lIPoKVonNAPpHnziAPoanOJ
5
6 access_token = 38797389676476272526-Qsjiepkj0JoJcRSe5PbQgTA5rCCAlPKjiJ
7 access_token_secret = Ay6r66rRF6F6f6BbsasBIjpeAHYpaUOp1Kk1RhdkKjAkWahQ
    
```

NOTA: Los datos de ejemplo no se encuentran disponibles ni son auténticos, su único fin es el enseñar el proceso de colocación de los mismos.

ARCHIVO AI_TwitterSAD:

En este archivo lo modificable para poder hacer una correcta ejecución es alrededor de la **línea 56**, en la cual se puede encontrar lo siguiente:

```

54 # Aquí va la consulta del usuario
55 #palabra = ['Ricardo Anaya']
56 palabra=['MORENA','AMLO','Cabeza de Algodón','Elecciones 2024','Marcelo Ebrard', 'Claudia Sheinbaum',
57 for x in palabra:
    
```


AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

Dentro de este arreglo podrás ya sea agregar *keywords*, *hashtags*, *términos completos*, *búsquedas espejo*, *búsquedas complejas con parámetros de perfil*, etc. O en su defecto únicamente trabajar con palabras sueltas.

En la **línea 66** se puede encontrar el parámetro modificable para el total de tweets a minar, en este caso definido a 500 para evitar trabas por el límite de la cuenta del developer.

```

63 # Se genera los últimos tweets sobre la consulta dada
64 tweets = tweepy.Cursor(api.search_tweets,
65                          q=filtered,
66                          lang="en").items(500)

```

La forma de trabajar de programa es que le pasas un arreglo de problemas y el solo hace todo el trabajo de **conectarse a la cuenta de developer**.

```

#Funcion para guardar los archivos de salida
def write_csv_type_of_param(nombre_archivo_salida,outtweets, type):
    with io.open(nombre_archivo_salida, type, encoding="utf-8", newline='') as f:
        writer = csv.writer(f, quoting=csv.QUOTE_ALL)
        if type == "w":
            writer.writerow(['id', 'created_at', 'text', 'user', 'retweet_count', 'favorite_count'])
            writer.writerows(outtweets)
        pass

# Instrucciones para leer en algunas versiones de Py archivos tipo .ini
def get_app_file_path(file):
    """Return the absolute path of the app's files. They should be in the same folder as this py file."""
    folder,_ = os.path.split(__file__)
    file_path = os.path.join(folder,file)
    return file_path

config = configparser.ConfigParser()
config.read(get_app_file_path('config.ini'))

# Asignación de las claves almacenadas del proyecto (TWITTER DEVELOPER)
#Podrás agregar tus claves en el archivo config.ini
consumer_key = config['twitter']['api_key']
consumer_secret = config['twitter']['api_key_secret']
access_token = config['twitter']['access_token']
access_token_secret = config['twitter']['access_token_secret']

# Acceso a la data de twitter
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth, wait_on_rate_limit=True) #wait_on_rate_limit hace la pausar al alcanzar el límite de tweepy

```

AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

Minar y guardar los tweets en un marco de datos.

```
# Aquí va la consulta del usuario
#palabra = ['Ricardo Anaya']
palabra=['MORENA','AMLO','Cabeza de Algodón','Elecciones 2024','Marcelo Ebrard', 'Claudia Sheinbaum', 'Adán Augusto lopez', 'Ricardo Anaya']
for x in palabra:
    # En este caso, el proyecto que nos fue asignado requería el arreglo de palabras sobre AMLO
    # Se filtra la consulta para eliminar retweets
    query = input(x)
    filtered = query + "-filter:retweets"

    # Se genera los últimos tweets sobre la consulta dada
    tweets = tweepy.Cursor(api.search_tweets,
                            q=filtered,
                            lang="en").items(500)

    # Se crea una lista de los tweets, los usuarios y su ubicación
    list1 = [[tweet.text, tweet.user.screen_name, tweet.user.location] for tweet in tweets]

    # Se convierte la lista en un marco de datos
    df = pd.DataFrame(data=list1,
                      columns=['tweets', 'user', "location"])

    # Convierte solo los tweets en una lista
    tweet_list = df.tweets.to_list()
```

Hacer una limpieza de caracteres, stopwords y palabras no utilizadas.

```
# Crea una función para limpiar los tweets.
# Eliminar las blasfemias o insultos, los caracteres innecesarios, los espacios y las palabras vacías.
def clean_tweet(tweet):
    if type(tweet) == np.float:
        return ""
    r = tweet.lower()
    r = profanity.censor(r)
    r = re.sub("'", "", r) # Esto es para evitar quitar las contracciones en ingles.
    r = re.sub("@[A-Za-z0-9_]+", "", r)
    r = re.sub("#[A-Za-z0-9_]+", "", r)
    r = re.sub(r'http\S+', '', r)
    r = re.sub('([()!@])', ' ', r)
    r = re.sub('\\[.*?\\]', ' ', r)
    r = re.sub("[^a-z0-9]", " ", r)
    r = r.split()

    # Los stopwords son palabras que normalmente no se usan al momento del analisis de sentimientos
    stopwords = ["a", "actualmente", "adelante", "además", "afirmó", "agregó", "ahora", "ahí", "al", "algo", "alguna", "algunas", "alguno",
    "stopwords = ["s", "may", "for", "on", "an", "a", "of", "and", "in", "the", "to", "from"] # Stopwords mas usadas en ingles
    r = [w for w in r if not w in stopwords]
    r = " ".join(word for word in r)

    return r

# Se ejecute la lista de tweets a través de la función.
cleaned = [clean_tweet(tw) for tw in tweet_list]
```

Hacer el guardado de los datos en archivos de tipo CSV con la fecha y la búsqueda en el nombre del archivo.

```
now = datetime.now()
nombre_archivo_salida = x + ' ' + now.strftime("%d-%m-%Y %H-%M-%S") + '.csv' #Se le da el nombre al archivo

if os.path.isfile(nombre_archivo_salida):
    write_csv_type_of_param(nombre_archivo_salida,cleaned, "a");
else:
    write_csv_type_of_param(nombre_archivo_salida,cleaned, "w");
```

AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

Define los objetos de sentimientos.

```
# Define los objetos de sentimiento usando TextBlob
sentiment_objects = [TextBlob(tweet) for tweet in cleaned]
sentiment_objects[0].polarity, sentiment_objects[0]

# Crea una lista de valores de polaridad y texto de tweet
sentiment_values = [[tweet.sentiment.polarity, str(tweet)] for tweet in sentiment_objects]

# Imprime el valor de la fila 0.
sentiment_values[0]

# Imprime todos los valores de sentimiento
sentiment_values[0:99]
```

Obtiene la polaridad de cada una.

```
# Crea un marco de datos de cada tweet contra su polaridad
sentiment_df = pd.DataFrame(sentiment_values, columns=["polarity", "tweet"])
print(sentiment_df)
print()
sentiment_df

# Guarda la columna de polaridad como 'n'.
n=sentiment_df["polarity"]

# Convierte esta columna en una serie, 'm'.
m=pd.Series(n)
```

Clasifica en Positivos y Negativos.

```
# Inicializa variables, 'pos', 'neg', 'neu'.
pos=0
neg=0
neu=0

# Crea un ciclo para clasificar los tweets como Positivos, Negativos o Neutrales.
# Cuenta el número de cada uno.
for items in m:
    if items>0:
        print("Positive")
        pos=pos+1
    elif items<0:
        print("Negative")
        neg=neg+1
    else:
        print("Neutral")
        neu=neu+1

print(pos, neg, neu)
```

AL Twitter Sentiment Analysis for Data Science		
INTELIGENCIA ARTIFICIAL	PYTHON	Joshua Mayorga Maca

E imprime los datos obtenidos en forma de Piechart y Wordcloud.

```

pieLabels=["Positive","Negative","Neutral"]
populationShare=[pos,neg,neu]

figureObject, axesObject = plt.subplots()

axesObject.pie(populationShare,labels=pieLabels,autopct='%1.2f',startangle=90)

axesObject.axis('equal')

plt.show()

# Muestra la cantidad de usuarios de Twitter que se sienten de cierta manera sobre el tema en cuestión.
print("%f percent of twitter users feel positive about %s"%(pos,query))

print("%f percent of twitter users feel negative about %s"%(neg,query))

print("%f percent of twitter users feel neutral about %s"%(neu,query))

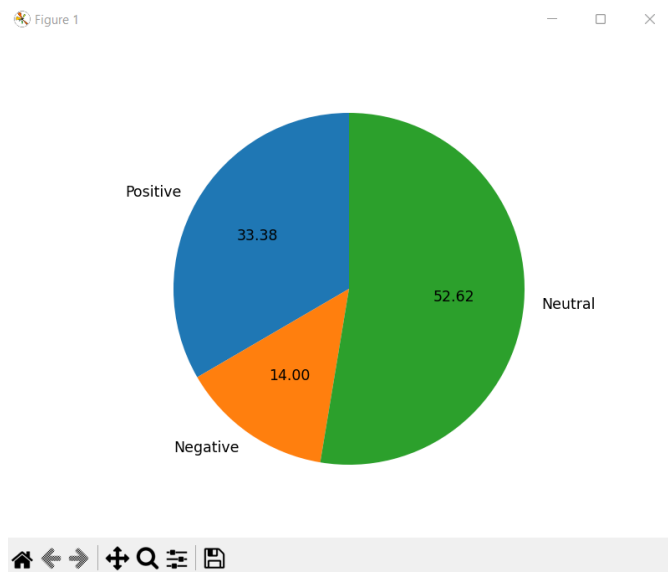
# Crear una Wordcloud a partir de los tweets
all_words = ' '.join([text for text in cleaned])
wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(all_words)

plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()

```

No se ocupa nada mas que ejecutar el programa de la manera mas adecuada para el usuario, el programa por sí solo irá avanzando en las diferentes etapas hasta llegar a la traficación.

IMPRESIÓN DE RESULTADOS:



Ejemplo 1. PieChart del tema AMLO donde se muestra un análisis de 5000 tweets tomando como variables positivos, negativos y neutrales.

