

```
In [14]: !pip install pandas numpy matplotlib seaborn  
!pip install networkx  
!pip install nltk  
!pip install wordcloud  
!pip install textblob vaderSentiment  
!pip install plotly pyvis bokeh  
!pip install spacy transformers
```

```
Requirement already satisfied: pandas in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (2.2.3)  
Requirement already satisfied: numpy in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (1.26.3)  
Requirement already satisfied: matplotlib in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (3.10.0)  
Requirement already satisfied: seaborn in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (0.13.2)  
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2023.4)  
Requirement already satisfied: tzdata>=2022.7 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2023.4)  
Requirement already satisfied: contourpy>=1.0.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (1.2.0)  
Requirement already satisfied: cycler>=0.10 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (0.12.1)  
Requirement already satisfied: fonttools>=4.22.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (4.49.0)  
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (1.4.5)  
Requirement already satisfied: packaging>=20.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (23.2)  
Requirement already satisfied: pillow>=8 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (10.2.0)  
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from matplotlib) (3.1.1)  
Requirement already satisfied: six>=1.5 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)  
[notice] A new release of pip is available: 24.0 -> 25.2  
[notice] To update, run: python.exe -m pip install --upgrade pip  
Requirement already satisfied: networkx in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (3.3)
```

```
[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
Requirement already satisfied: nltk in c:\users\oscar escriba\appdata\local
\programs\python\python312\lib\site-packages (3.9.1)
[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
Requirement already satisfied: click in c:\users\oscar escriba\appdata\local
\programs\python\python312\lib\site-packages (from nltk) (8.2.1)
Requirement already satisfied: joblib in c:\users\oscar escriba\appdata\local
\programs\python\python312\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\oscar escriba\app
data\local\programs\python\python312\lib\site-packages (from nltk) (2025.7.3
4)
Requirement already satisfied: tqdm in c:\users\oscar escriba\appdata\local
\programs\python\python312\lib\site-packages (from nltk) (4.67.1)
Requirement already satisfied: colorama in c:\users\oscar escriba\appdata\ro
aming\python\python312\site-packages (from click->nltk) (0.4.6)
Requirement already satisfied: wordcloud in c:\users\oscar escriba\appdata\l
ocal\programs\python\python312\lib\site-packages (1.9.4)
Requirement already satisfied: numpy>=1.6.1 in c:\users\oscar escriba\appdat
a\local\programs\python\python312\lib\site-packages (from wordcloud) (1.26.
3)
Requirement already satisfied: pillow in c:\users\oscar escriba\appdata\loca
l\programs\python\python312\lib\site-packages (from wordcloud) (10.2.0)
Requirement already satisfied: matplotlib in c:\users\oscar escriba\appdata
\local\programs\python\python312\lib\site-packages (from wordcloud) (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\oscar escriba\ap
pdata\local\programs\python\python312\lib\site-packages (from matplotlib->wo
rdcloud) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\oscar escriba\appdat
a\local\programs\python\python312\lib\site-packages (from matplotlib->wordcl
oud) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\oscar escriba\ap
pdata\local\programs\python\python312\lib\site-packages (from matplotlib->w
ordcloud) (4.49.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\oscar escriba\ap
pdata\local\programs\python\python312\lib\site-packages (from matplotlib->w
ordcloud) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\oscar escriba\app
data\local\programs\python\python312\lib\site-packages (from matplotlib->wor
dcloud) (23.2)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\oscar escriba\ap
pdata\local\programs\python\python312\lib\site-packages (from matplotlib->wo
rdcloud) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\oscar escri
ba\appdata\local\programs\python\python312\lib\site-packages (from matplotlib->
wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\oscar escriba\appdata\lo
cal\programs\python\python312\lib\site-packages (from python-dateutil>=2.7->
matplotlib->wordcloud) (1.16.0)
[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
Collecting textblob
  Downloading textblob-0.19.0-py3-none-any.whl.metadata (4.4 kB)
Collecting vaderSentiment
  Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl.metadata (572 bytes)
Requirement already satisfied: nltk>=3.9 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from textblob) (3.9.1)
Requirement already satisfied: requests in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from vaderSentiment) (2.32.3)
Requirement already satisfied: click in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from nltk>=3.9->textblob) (8.2.1)
Requirement already satisfied: joblib in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from nltk>=3.9->textblob) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from nltk>=3.9->textblob) (2025.7.34)
Requirement already satisfied: tqdm in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from nltk>=3.9->textblob) (4.67.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from requests->vaderSentiment) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from requests->vaderSentiment) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from requests->vaderSentiment) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from requests->vaderSentiment) (2025.6.15)
Requirement already satisfied: colorama in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from click->nltk>=3.9->textblob) (0.4.6)
  Downloading textblob-0.19.0-py3-none-any.whl (624 kB)
    0.0/624.3 kB ? eta -----  
  10.2/624.3 kB ? eta -----  
  30.7/624.3 kB 330.3 kB/s eta 0:0  
0:02  
    61.4/624.3 kB 469.7 kB/s eta 0:0  
0:02  
    92.2/624.3 kB 581.0 kB/s eta 0:0  
0:01  
    122.9/624.3 kB 602.4 kB/s eta 0:0  
0:01  
    174.1/624.3 kB 655.4 kB/s eta 0:0  
0:01  
    225.3/624.3 kB 724.0 kB/s eta 0:0  
0:01  
    266.2/624.3 kB 744.2 kB/s eta 0:0  
0:01  
    337.9/624.3 kB 838.1 kB/s eta 0:0  
0:01  
    399.4/624.3 kB 890.4 kB/s eta 0:0
```

```
0:01          ----- 450.6/624.3 kB 972.0 kB/s eta 0:0
0:01          ----- 481.3/624.3 kB 887.1 kB/s eta 0:0
0:01          ----- 501.8/624.3 kB 898.6 kB/s eta 0:0
0:01          ----- 553.0/624.3 kB 868.4 kB/s eta 0:0
0:01          ----- 563.2/624.3 kB 862.7 kB/s eta 0:0
0:01          ----- 624.3/624.3 kB 893.8 kB/s eta 0:0
0:00
Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl (125 kB)
----- 0.0/126.0 kB ? eta -:-----
----- 10.2/126.0 kB ? eta -:-----
----- 41.0/126.0 kB 991.0 kB/s eta 0:0
0:01          ----- 122.9/126.0 kB 1.0 MB/s eta 0:0
0:01          ----- 126.0/126.0 kB 1.1 MB/s eta 0:0
0:00
Installing collected packages: vaderSentiment, textblob
Successfully installed textblob-0.19.0 vaderSentiment-3.3.2
[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
Collecting plotly
  Downloading plotly-6.3.0-py3-none-any.whl.metadata (8.5 kB)
Collecting pyvis
  Downloading pyvis-0.3.2-py3-none-any.whl.metadata (1.7 kB)
Collecting bokeh
  Downloading bokeh-3.8.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: narwhals>=1.15.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from plotly) (2.0.1)
Requirement already satisfied: packaging in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from plotly) (23.2)
Requirement already satisfied: ipython>=5.3.0 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from pyvis) (8.26.0)
Requirement already satisfied: jinja2>=2.9.6 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from pyvis) (3.1.6)
Collecting jsonpickle>=1.4.1 (from pyvis)
  Downloading jsonpickle-4.1.1-py3-none-any.whl.metadata (8.1 kB)
Requirement already satisfied: networkx>=1.11 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from pyvis) (3.3)
Requirement already satisfied: contourpy>=1.2 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from bokeh) (1.2.0)
Requirement already satisfied: numpy>=1.16 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from bokeh) (1.26.3)
Requirement already satisfied: pandas>=1.2 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from bokeh) (2.2.3)
Requirement already satisfied: pillow>=7.1.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from bokeh) (10.2.0)
Requirement already satisfied: PyYAML>=3.10 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from bokeh) (6.0.2)
Requirement already satisfied: tornado>=6.2 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from bokeh) (6.4.1)
Collecting xyzservices>=2021.09.1 (from bokeh)
  Downloading xyzservices-2025.4.0-py3-none-any.whl.metadata (4.3 kB)
Requirement already satisfied: decorator in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from ipython>=5.3.0->pyvis) (5.1.1)
Requirement already satisfied: jedi>=0.16 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from ipython>=5.3.0->pyvis) (0.19.1)
Requirement already satisfied: matplotlib-inline in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from ipython>=5.3.0->pyvis) (0.1.7)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from ipython>=5.3.0->pyvis) (3.0.47)
Requirement already satisfied: pygments>=2.4.0 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from ipython>=5.3.0->pyvis) (2.18.0)
Requirement already satisfied: stack-data in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from ipython>=5.3.0->pyvis) (0.6.3)
Requirement already satisfied: traitlets>=5.13.0 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from ipython>=5.3.0->pyvis) (5.14.3)
Requirement already satisfied: colorama in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from ipython>=5.3.0->pyvis) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from jinja2>=2.9.6->
```

```
pyvis) (3.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from pandas>=1.2->bokeh) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from pandas>=1.2->bokeh) (2023.4)
Requirement already satisfied: tzdata>=2022.7 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from pandas>=1.2->bokeh) (2023.4)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from jedi>=0.16->ipython>=5.3.0->pyvis) (0.8.4)
Requirement already satisfied: wcwidth in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from prompt-toolkit<3.1.0,>=3.0.41->ipython>=5.3.0->pyvis) (0.2.13)
Requirement already satisfied: six>=1.5 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from python-dateutil>=2.8.2->pandas>=1.2->bokeh) (1.16.0)
Requirement already satisfied: executing>=1.2.0 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from stack-data->ipython>=5.3.0->pyvis) (2.0.1)
Requirement already satisfied: asttokens>=2.1.0 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from stack-data->ipython>=5.3.0->pyvis) (2.4.1)
Requirement already satisfied: pure-eval in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from stack-data->ipython>=5.3.0->pyvis) (0.2.3)

Downloading plotly-6.3.0-py3-none-any.whl (9.8 MB)
-----
   0.0/9.8 MB ? eta ----
   0.0/9.8 MB 640.0 kB/s eta 0:00:1
6
   0.0/9.8 MB 487.6 kB/s eta 0:00:2
0
   0.1/9.8 MB 744.7 kB/s eta 0:00:1
4
   0.1/9.8 MB 654.9 kB/s eta 0:00:1
5
   0.2/9.8 MB 748.1 kB/s eta 0:00:1
3
   0.2/9.8 MB 784.3 kB/s eta 0:00:1
3
   0.3/9.8 MB 785.2 kB/s eta 0:00:1
3
   0.3/9.8 MB 922.8 kB/s eta 0:00:1
1
   0.4/9.8 MB 969.0 kB/s eta 0:00:1
0
   0.5/9.8 MB 1.1 MB/s eta 0:00:09
   0.6/9.8 MB 1.1 MB/s eta 0:00:09
   0.6/9.8 MB 1.1 MB/s eta 0:00:09
   0.6/9.8 MB 1.1 MB/s eta 0:00:09
   0.7/9.8 MB 1.1 MB/s eta 0:00:08
   0.8/9.8 MB 1.1 MB/s eta 0:00:08
   0.8/9.8 MB 1.1 MB/s eta 0:00:08
   0.9/9.8 MB 1.2 MB/s eta 0:00:08
```



```
----- 8.0/9.8 MB 1.3 MB/s eta 0:00:02
----- 8.0/9.8 MB 1.3 MB/s eta 0:00:02
----- 8.1/9.8 MB 1.3 MB/s eta 0:00:02
----- 8.2/9.8 MB 1.3 MB/s eta 0:00:02
----- 8.2/9.8 MB 1.3 MB/s eta 0:00:02
----- 8.3/9.8 MB 1.3 MB/s eta 0:00:02
----- 8.4/9.8 MB 1.3 MB/s eta 0:00:02
----- 8.4/9.8 MB 1.3 MB/s eta 0:00:02
----- 8.5/9.8 MB 1.3 MB/s eta 0:00:01
----- 8.6/9.8 MB 1.3 MB/s eta 0:00:01
----- 8.7/9.8 MB 1.3 MB/s eta 0:00:01
----- 8.7/9.8 MB 1.3 MB/s eta 0:00:01
----- 8.8/9.8 MB 1.3 MB/s eta 0:00:01
----- 8.9/9.8 MB 1.3 MB/s eta 0:00:01
----- 8.9/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.0/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.1/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.1/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.2/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.3/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.3/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.4/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.5/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.6/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.6/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.7/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.7/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.8/9.8 MB 1.3 MB/s eta 0:00:01
----- 9.8/9.8 MB 1.3 MB/s eta 0:00:00

Downloading pyvis-0.3.2-py3-none-any.whl (756 kB)
----- 0.0/756.0 kB ? eta -:-:-
----- 41.0/756.0 kB 1.9 MB/s eta 0:00:
01
----- 92.2/756.0 kB 1.3 MB/s eta 0:00:
01
----- 143.4/756.0 kB 1.4 MB/s eta 0:0
0:01
----- 204.8/756.0 kB 1.2 MB/s eta 0:0
0:01
----- 256.0/756.0 kB 1.3 MB/s eta 0:0
0:01
----- 337.9/756.0 kB 1.3 MB/s eta 0:0
0:01
----- 409.6/756.0 kB 1.3 MB/s eta 0:0
0:01
----- 491.5/756.0 kB 1.4 MB/s eta 0:0
0:01
----- 553.0/756.0 kB 1.4 MB/s eta 0:0
0:01
----- 665.6/756.0 kB 1.5 MB/s eta 0:0
0:01
----- 756.0/756.0 kB 1.5 MB/s eta 0:0
0:00

Downloading bokeh-3.8.0-py3-none-any.whl (7.2 MB)
----- 0.0/7.2 MB ? eta -:-:-
----- 0.0/7.2 MB ? eta -:-:-
```

```
----- 0.1/7.2 MB 2.0 MB/s eta 0:00:04  
----- 0.2/7.2 MB 2.0 MB/s eta 0:00:04  
----- 0.3/7.2 MB 1.6 MB/s eta 0:00:05  
----- 0.3/7.2 MB 1.8 MB/s eta 0:00:04  
----- 0.4/7.2 MB 1.7 MB/s eta 0:00:05  
----- 0.5/7.2 MB 1.6 MB/s eta 0:00:05  
----- 0.5/7.2 MB 1.5 MB/s eta 0:00:05  
----- 0.5/7.2 MB 1.4 MB/s eta 0:00:05  
----- 0.6/7.2 MB 1.4 MB/s eta 0:00:05  
----- 0.6/7.2 MB 1.2 MB/s eta 0:00:06  
----- 0.6/7.2 MB 1.2 MB/s eta 0:00:06  
----- 0.7/7.2 MB 1.2 MB/s eta 0:00:06  
----- 0.8/7.2 MB 1.2 MB/s eta 0:00:06  
----- 0.8/7.2 MB 1.2 MB/s eta 0:00:06  
----- 1.0/7.2 MB 1.3 MB/s eta 0:00:05  
----- 1.0/7.2 MB 1.3 MB/s eta 0:00:05  
----- 1.1/7.2 MB 1.3 MB/s eta 0:00:05  
----- 1.2/7.2 MB 1.4 MB/s eta 0:00:05  
----- 1.3/7.2 MB 1.4 MB/s eta 0:00:05  
----- 1.4/7.2 MB 1.4 MB/s eta 0:00:05  
----- 1.5/7.2 MB 1.5 MB/s eta 0:00:04  
----- 1.5/7.2 MB 1.5 MB/s eta 0:00:04  
----- 1.6/7.2 MB 1.5 MB/s eta 0:00:04  
----- 1.7/7.2 MB 1.5 MB/s eta 0:00:04  
----- 1.8/7.2 MB 1.5 MB/s eta 0:00:04  
----- 1.8/7.2 MB 1.5 MB/s eta 0:00:04  
----- 1.9/7.2 MB 1.5 MB/s eta 0:00:04  
----- 2.0/7.2 MB 1.5 MB/s eta 0:00:04  
----- 2.1/7.2 MB 1.5 MB/s eta 0:00:04  
----- 2.1/7.2 MB 1.5 MB/s eta 0:00:04  
----- 2.2/7.2 MB 1.5 MB/s eta 0:00:04  
----- 2.3/7.2 MB 1.5 MB/s eta 0:00:04  
----- 2.4/7.2 MB 1.5 MB/s eta 0:00:04  
----- 2.5/7.2 MB 1.5 MB/s eta 0:00:04  
----- 2.5/7.2 MB 1.5 MB/s eta 0:00:04  
----- 2.6/7.2 MB 1.5 MB/s eta 0:00:03  
----- 2.7/7.2 MB 1.5 MB/s eta 0:00:03  
----- 2.7/7.2 MB 1.5 MB/s eta 0:00:03  
----- 2.8/7.2 MB 1.5 MB/s eta 0:00:03  
----- 2.9/7.2 MB 1.5 MB/s eta 0:00:03  
----- 2.9/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.0/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.0/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.1/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.2/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.3/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.3/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.4/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.5/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.6/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.7/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.8/7.2 MB 1.5 MB/s eta 0:00:03  
----- 3.9/7.2 MB 1.5 MB/s eta 0:00:03  
----- 4.0/7.2 MB 1.6 MB/s eta 0:00:03  
----- 4.0/7.2 MB 1.5 MB/s eta 0:00:03  
----- 4.1/7.2 MB 1.5 MB/s eta 0:00:03
```

```
----- 4.1/7.2 MB 1.5 MB/s eta 0:00:03
----- 4.2/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.3/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.4/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.4/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.5/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.5/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.6/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.7/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.7/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.8/7.2 MB 1.5 MB/s eta 0:00:02
----- 4.9/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.0/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.1/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.1/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.2/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.3/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.4/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.4/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.5/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.5/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.6/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.7/7.2 MB 1.5 MB/s eta 0:00:02
----- 5.8/7.2 MB 1.5 MB/s eta 0:00:01
----- 5.8/7.2 MB 1.5 MB/s eta 0:00:01
----- 5.9/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.0/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.1/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.2/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.2/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.2/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.3/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.3/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.4/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.5/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.6/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.7/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.7/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.8/7.2 MB 1.5 MB/s eta 0:00:01
----- 6.9/7.2 MB 1.5 MB/s eta 0:00:01
----- 7.0/7.2 MB 1.5 MB/s eta 0:00:01
----- 7.0/7.2 MB 1.5 MB/s eta 0:00:01
----- 7.2/7.2 MB 1.5 MB/s eta 0:00:01
----- 7.2/7.2 MB 1.5 MB/s eta 0:00:00

Downloading jsonpickle-4.1.1-py3-none-any.whl (47 kB)
----- 0.0/47.1 kB ? eta ----- 47.1/47.1 kB ? eta 0:00:00

Downloading xyzservices-2025.4.0-py3-none-any.whl (90 kB)
----- 0.0/90.4 kB ? eta ----- 30.7/90.4 kB 1.4 MB/s eta 0:00:00

1 ----- 90.4/90.4 kB 1.7 MB/s eta 0:00:00
0

Installing collected packages: xyzservices, plotly, jsonpickle, bokeh, pyvis
Successfully installed bokeh-3.8.0 jsonpickle-4.1.1 plotly-6.3.0 pyvis-0.3.2
xyzservices-2025.4.0
```

```
[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
Collecting spacy
  Downloading spacy-3.8.7-cp312-cp312-win_amd64.whl.metadata (28 kB)
Collecting transformers
  Downloading transformers-4.56.1-py3-none-any.whl.metadata (42 kB)
    0.0/42.2 kB ? eta :---:
    30.7/42.2 kB 640.0 kB/s eta 0:0
0:01
    42.2/42.2 kB 679.4 kB/s eta 0:0
0:00
Collecting spacy-legacy<3.1.0,>=3.0.11 (from spacy)
  Downloading spacy_legacy-3.0.12-py2.py3-none-any.whl.metadata (2.8 kB)
Collecting spacy-loggers<2.0.0,>=1.0.0 (from spacy)
  Downloading spacy_loggers-1.0.5-py3-none-any.whl.metadata (23 kB)
Collecting murmurhash<1.1.0,>=0.28.0 (from spacy)
  Downloading murmurhash-1.0.13-cp312-cp312-win_amd64.whl.metadata (2.2 kB)
Collecting cymem<2.1.0,>=2.0.2 (from spacy)
  Downloading cymem-2.0.11-cp312-cp312-win_amd64.whl.metadata (8.8 kB)
Collecting preshed<3.1.0,>=3.0.2 (from spacy)
  Downloading preshed-3.0.10-cp312-cp312-win_amd64.whl.metadata (2.5 kB)
Collecting thinc<8.4.0,>=8.3.4 (from spacy)
  Downloading thinc-8.3.6-cp312-cp312-win_amd64.whl.metadata (15 kB)
Collecting wasabi<1.2.0,>=0.9.1 (from spacy)
  Downloading wasabi-1.1.3-py3-none-any.whl.metadata (28 kB)
Collecting srsly<3.0.0,>=2.4.3 (from spacy)
  Downloading srsly-2.5.1-cp312-cp312-win_amd64.whl.metadata (20 kB)
Collecting catalogue<2.1.0,>=2.0.6 (from spacy)
  Downloading catalogue-2.0.10-py3-none-any.whl.metadata (14 kB)
Collecting weasel<0.5.0,>=0.1.0 (from spacy)
  Downloading weasel-0.4.1-py3-none-any.whl.metadata (4.6 kB)
Collecting typer<1.0.0,>=0.3.0 (from spacy)
  Downloading typer-0.17.4-py3-none-any.whl.metadata (15 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from spacy) (4.67.1)
Requirement already satisfied: numpy>=1.19.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from spacy) (1.26.3)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from spacy) (2.32.3)
Collecting pydantic!=1.8,!>1.8.1,<3.0.0,>=1.7.4 (from spacy)
  Downloading pydantic-2.11.7-py3-none-any.whl.metadata (67 kB)
    0.0/68.0 kB ? eta :---:
    10.2/68.0 kB ? eta :---:
    30.7/68.0 kB 435.7 kB/s eta 0:0
0:01
    41.0/68.0 kB 108.9 kB/s eta 0:0
0:01
    41.0/68.0 kB 108.9 kB/s eta 0:0
```

```
0:01      ----- 41.0/68.0 kB 108.9 kB/s eta 0:0
0:01      ----- 41.0/68.0 kB 108.9 kB/s eta 0:0
0:01      ----- 68.0/68.0 kB 123.1 kB/s eta 0:0
0:00
Requirement already satisfied: jinja2 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from spacy) (78.1.0)
Requirement already satisfied: packaging>=20.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from spacy) (23.2)
Collecting langcodes<4.0.0,>=3.2.0 (from spacy)
    Downloading langcodes-3.5.0-py3-none-any.whl.metadata (29 kB)
Requirement already satisfied: filelock in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from transformers) (3.18.0)
Collecting huggingface-hub<1.0,>=0.34.0 (from transformers)
    Downloading huggingface_hub-0.34.4-py3-none-any.whl.metadata (14 kB)
Requirement already satisfied: pyyaml>=5.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from transformers) (2025.7.34)
Collecting tokenizers<=0.23.0,>=0.22.0 (from transformers)
    Downloading tokenizers-0.22.0-cp39-abi3-win_amd64.whl.metadata (6.9 kB)
Collecting safetensors>=0.4.3 (from transformers)
    Downloading safetensors-0.6.2-cp38-abi3-win_amd64.whl.metadata (4.1 kB)
Requirement already satisfied: fsspec>=2023.5.0 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from huggingface-hub<1.0,>=0.34.0->transformers) (2025.7.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from huggingface-hub<1.0,>=0.34.0->transformers) (4.14.1)
Collecting language-data>=1.2 (from langcodes<4.0.0,>=3.2.0->spacy)
    Downloading language_data-1.3.0-py3-none-any.whl.metadata (4.3 kB)
Collecting annotated-types>=0.6.0 (from pydantic!=1.8,!>1.8.1,<3.0.0,>=1.7.4->spacy)
    Downloading annotated_types-0.7.0-py3-none-any.whl.metadata (15 kB)
Collecting pydantic-core==2.33.2 (from pydantic!=1.8,!>1.8.1,<3.0.0,>=1.7.4->spacy)
    Downloading pydantic_core-2.33.2-cp312-cp312-win_amd64.whl.metadata (6.9 kB)
Collecting typing-inspection>=0.4.0 (from pydantic!=1.8,!>1.8.1,<3.0.0,>=1.7.4->spacy)
    Downloading typing_inspection-0.4.1-py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2.5.0)
```

```
Requirement already satisfied: certifi>=2017.4.17 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2025.6.15)
Collecting blis<1.4.0,>=1.3.0 (from thinc<8.4.0,>=8.3.4->spacy)
  Downloading blis-1.3.0-cp312-cp312-win_amd64.whl.metadata (7.6 kB)
Collecting confection<1.0.0,>=0.0.1 (from thinc<8.4.0,>=8.3.4->spacy)
  Downloading confection-0.1.5-py3-none-any.whl.metadata (19 kB)
Collecting numpy>=1.19.0 (from spacy)
  Downloading numpy-2.3.2-cp312-cp312-win_amd64.whl.metadata (60 kB)
----- 0.0/60.9 kB ? eta ------
----- 30.7/60.9 kB 1.3 MB/s eta 0:0
0:01 ----- 60.9/60.9 kB 1.6 MB/s eta 0:0
0:00
Requirement already satisfied: colorama in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from tqdm<5.0.0,>=4.38.0->spacy) (0.4.6)
Requirement already satisfied: click>=8.0.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from typer<1.0.0,>=0.3.0->spacy) (8.2.1)
Collecting shellingham>=1.3.0 (from typer<1.0.0,>=0.3.0->spacy)
  Downloading shellingham-1.5.4-py2.py3-none-any.whl.metadata (3.5 kB)
Requirement already satisfied: rich>=10.11.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from typer<1.0.0,>=0.3.0->spacy) (13.9.4)
Collecting cloudpathlib<1.0.0,>=0.7.0 (from weasel<0.5.0,>=0.1.0->spacy)
  Downloading cloudpathlib-0.22.0-py3-none-any.whl.metadata (16 kB)
Collecting smart-open<8.0.0,>=5.2.1 (from weasel<0.5.0,>=0.1.0->spacy)
  Downloading smart_open-7.3.0.post1-py3-none-any.whl.metadata (24 kB)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from jinja2->spacy) (3.0.2)
Collecting marisa-trie>=1.1.0 (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy)
  Downloading marisa_trie-1.3.1-cp312-cp312-win_amd64.whl.metadata (10 kB)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\oscar escriba\appdata\roaming\python\python312\site-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (2.18.0)
Requirement already satisfied: wrapt in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.1.0->spacy) (1.17.2)
Requirement already satisfied: mdurl~0.1 in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (0.1.2)
Downloading spacy-3.8.7-cp312-cp312-win_amd64.whl (13.9 MB)
----- 0.0/13.9 kB ? eta ------
----- 0.0/13.9 kB 1.3 MB/s eta 0:00:11
----- 0.1/13.9 kB 1.6 MB/s eta 0:00:09
----- 0.1/13.9 kB 744.7 kB/s eta 0:00:
19 ----- 0.2/13.9 MB 913.1 kB/s eta 0:00:
16 ----- 0.2/13.9 MB 1.1 MB/s eta 0:00:13
```

0.3/13.9 MB 1.0 MB/s eta 0:00:14
0.4/13.9 MB 1.2 MB/s eta 0:00:12
0.4/13.9 MB 1.2 MB/s eta 0:00:12
0.5/13.9 MB 1.2 MB/s eta 0:00:12
0.5/13.9 MB 1.2 MB/s eta 0:00:12
0.6/13.9 MB 1.2 MB/s eta 0:00:12
0.7/13.9 MB 1.3 MB/s eta 0:00:11
0.7/13.9 MB 1.3 MB/s eta 0:00:11
0.8/13.9 MB 1.3 MB/s eta 0:00:11
0.9/13.9 MB 1.3 MB/s eta 0:00:11
0.9/13.9 MB 1.3 MB/s eta 0:00:11
1.0/13.9 MB 1.3 MB/s eta 0:00:11
1.0/13.9 MB 1.2 MB/s eta 0:00:11
1.1/13.9 MB 1.3 MB/s eta 0:00:11
1.2/13.9 MB 1.3 MB/s eta 0:00:11
1.2/13.9 MB 1.2 MB/s eta 0:00:11
1.2/13.9 MB 1.2 MB/s eta 0:00:11
1.3/13.9 MB 1.2 MB/s eta 0:00:11
1.3/13.9 MB 1.2 MB/s eta 0:00:11
1.4/13.9 MB 1.2 MB/s eta 0:00:11
1.4/13.9 MB 1.2 MB/s eta 0:00:11
1.5/13.9 MB 1.2 MB/s eta 0:00:11
1.5/13.9 MB 1.2 MB/s eta 0:00:11
1.6/13.9 MB 1.2 MB/s eta 0:00:11
1.6/13.9 MB 1.2 MB/s eta 0:00:11
1.7/13.9 MB 1.2 MB/s eta 0:00:11
1.8/13.9 MB 1.2 MB/s eta 0:00:11
1.9/13.9 MB 1.2 MB/s eta 0:00:10
1.9/13.9 MB 1.2 MB/s eta 0:00:10
2.0/13.9 MB 1.2 MB/s eta 0:00:10
2.0/13.9 MB 1.2 MB/s eta 0:00:10
2.0/13.9 MB 1.2 MB/s eta 0:00:10
2.1/13.9 MB 1.2 MB/s eta 0:00:10
2.1/13.9 MB 1.2 MB/s eta 0:00:10
2.2/13.9 MB 1.2 MB/s eta 0:00:10
2.2/13.9 MB 1.2 MB/s eta 0:00:10
2.3/13.9 MB 1.2 MB/s eta 0:00:10
2.3/13.9 MB 1.2 MB/s eta 0:00:11
2.3/13.9 MB 1.2 MB/s eta 0:00:11
2.4/13.9 MB 1.1 MB/s eta 0:00:11
2.4/13.9 MB 1.2 MB/s eta 0:00:10
2.5/13.9 MB 1.2 MB/s eta 0:00:10
2.5/13.9 MB 1.1 MB/s eta 0:00:10
2.6/13.9 MB 1.1 MB/s eta 0:00:10
2.6/13.9 MB 1.1 MB/s eta 0:00:11
2.7/13.9 MB 1.1 MB/s eta 0:00:10
2.7/13.9 MB 1.1 MB/s eta 0:00:10
2.8/13.9 MB 1.1 MB/s eta 0:00:10
2.8/13.9 MB 1.1 MB/s eta 0:00:10
2.8/13.9 MB 1.1 MB/s eta 0:00:10
2.9/13.9 MB 1.1 MB/s eta 0:00:11
2.9/13.9 MB 1.1 MB/s eta 0:00:11
2.9/13.9 MB 1.1 MB/s eta 0:00:11
3.0/13.9 MB 1.1 MB/s eta 0:00:11
3.0/13.9 MB 1.1 MB/s eta 0:00:11
3.0/13.9 MB 1.1 MB/s eta 0:00:11

----- 3.0/13.9 MB 1.1 MB/s eta 0:00:11
----- 3.0/13.9 MB 1.1 MB/s eta 0:00:11
----- 3.1/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.1/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.1/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.2/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.3/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.3/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.3/13.9 MB 998.3 kB/s eta 0:00:
11 ----- 3.4/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.5/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.5/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.6/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.6/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.7/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.7/13.9 MB 1.0 MB/s eta 0:00:11
----- 3.8/13.9 MB 1.0 MB/s eta 0:00:10
----- 3.8/13.9 MB 1.0 MB/s eta 0:00:10
----- 3.9/13.9 MB 1.0 MB/s eta 0:00:10
----- 4.0/13.9 MB 1.0 MB/s eta 0:00:10
----- 4.0/13.9 MB 1.0 MB/s eta 0:00:10
----- 4.0/13.9 MB 1.0 MB/s eta 0:00:10
----- 4.1/13.9 MB 1.0 MB/s eta 0:00:10
----- 4.1/13.9 MB 1.0 MB/s eta 0:00:10
----- 4.2/13.9 MB 986.5 kB/s eta 0:00:
10 ----- 4.2/13.9 MB 990.3 kB/s eta 0:00:
10 ----- 4.3/13.9 MB 984.0 kB/s eta 0:00:
10 ----- 4.5/13.9 MB 1.0 MB/s eta 0:00:10
----- 4.6/13.9 MB 1.0 MB/s eta 0:00:10
----- 4.6/13.9 MB 1.0 MB/s eta 0:00:10
----- 4.7/13.9 MB 1.0 MB/s eta 0:00:09
----- 4.7/13.9 MB 1.0 MB/s eta 0:00:09
----- 4.8/13.9 MB 1.0 MB/s eta 0:00:09
----- 4.9/13.9 MB 1.0 MB/s eta 0:00:09
----- 4.9/13.9 MB 1.0 MB/s eta 0:00:09
----- 5.0/13.9 MB 1.0 MB/s eta 0:00:09
----- 5.1/13.9 MB 1.1 MB/s eta 0:00:09
----- 5.1/13.9 MB 1.1 MB/s eta 0:00:09
----- 5.2/13.9 MB 1.1 MB/s eta 0:00:09
----- 5.3/13.9 MB 1.1 MB/s eta 0:00:09
----- 5.4/13.9 MB 1.1 MB/s eta 0:00:08
----- 5.4/13.9 MB 1.1 MB/s eta 0:00:08
----- 5.5/13.9 MB 1.1 MB/s eta 0:00:08
----- 5.6/13.9 MB 1.1 MB/s eta 0:00:08
----- 5.6/13.9 MB 1.1 MB/s eta 0:00:08
----- 5.7/13.9 MB 1.1 MB/s eta 0:00:08
----- 5.8/13.9 MB 1.1 MB/s eta 0:00:08

5.9/13.9 MB 1.1 MB/s eta 0:00:08
6.0/13.9 MB 1.1 MB/s eta 0:00:08
6.0/13.9 MB 1.1 MB/s eta 0:00:08
6.1/13.9 MB 1.1 MB/s eta 0:00:07
6.2/13.9 MB 1.1 MB/s eta 0:00:07
6.3/13.9 MB 1.1 MB/s eta 0:00:07
6.3/13.9 MB 1.1 MB/s eta 0:00:07
6.4/13.9 MB 1.1 MB/s eta 0:00:07
6.5/13.9 MB 1.1 MB/s eta 0:00:07
6.6/13.9 MB 1.1 MB/s eta 0:00:07
6.6/13.9 MB 1.1 MB/s eta 0:00:07
6.7/13.9 MB 1.1 MB/s eta 0:00:07
6.7/13.9 MB 1.2 MB/s eta 0:00:07
6.8/13.9 MB 1.2 MB/s eta 0:00:07
6.9/13.9 MB 1.2 MB/s eta 0:00:07
7.0/13.9 MB 1.2 MB/s eta 0:00:07
7.1/13.9 MB 1.2 MB/s eta 0:00:06
7.1/13.9 MB 1.2 MB/s eta 0:00:06
7.2/13.9 MB 1.2 MB/s eta 0:00:06
7.3/13.9 MB 1.2 MB/s eta 0:00:06
7.4/13.9 MB 1.2 MB/s eta 0:00:06
7.4/13.9 MB 1.2 MB/s eta 0:00:06
7.5/13.9 MB 1.2 MB/s eta 0:00:06
7.6/13.9 MB 1.2 MB/s eta 0:00:06
7.7/13.9 MB 1.2 MB/s eta 0:00:06
7.7/13.9 MB 1.2 MB/s eta 0:00:06
7.8/13.9 MB 1.2 MB/s eta 0:00:06
7.9/13.9 MB 1.2 MB/s eta 0:00:06
7.9/13.9 MB 1.2 MB/s eta 0:00:05
8.0/13.9 MB 1.2 MB/s eta 0:00:05
8.0/13.9 MB 1.2 MB/s eta 0:00:05
8.1/13.9 MB 1.2 MB/s eta 0:00:05
8.1/13.9 MB 1.2 MB/s eta 0:00:05
8.2/13.9 MB 1.2 MB/s eta 0:00:05
8.2/13.9 MB 1.2 MB/s eta 0:00:05
8.3/13.9 MB 1.2 MB/s eta 0:00:05
8.3/13.9 MB 1.2 MB/s eta 0:00:05
8.4/13.9 MB 1.2 MB/s eta 0:00:05
8.4/13.9 MB 1.2 MB/s eta 0:00:05
8.5/13.9 MB 1.2 MB/s eta 0:00:05
8.6/13.9 MB 1.2 MB/s eta 0:00:05
8.6/13.9 MB 1.2 MB/s eta 0:00:05
8.7/13.9 MB 1.2 MB/s eta 0:00:05
8.8/13.9 MB 1.2 MB/s eta 0:00:05
8.9/13.9 MB 1.2 MB/s eta 0:00:05
8.9/13.9 MB 1.2 MB/s eta 0:00:05
9.0/13.9 MB 1.2 MB/s eta 0:00:05
9.1/13.9 MB 1.2 MB/s eta 0:00:05
9.1/13.9 MB 1.2 MB/s eta 0:00:04
9.2/13.9 MB 1.2 MB/s eta 0:00:04
9.3/13.9 MB 1.2 MB/s eta 0:00:04
9.4/13.9 MB 1.2 MB/s eta 0:00:04
9.4/13.9 MB 1.2 MB/s eta 0:00:04
9.5/13.9 MB 1.2 MB/s eta 0:00:04
9.6/13.9 MB 1.2 MB/s eta 0:00:04
9.7/13.9 MB 1.2 MB/s eta 0:00:04

----- 9.7/13.9 MB 1.2 MB/s eta 0:00:04
----- 9.8/13.9 MB 1.2 MB/s eta 0:00:04
----- 9.9/13.9 MB 1.2 MB/s eta 0:00:04
----- 10.0/13.9 MB 1.2 MB/s eta 0:00:0
4 ----- 10.1/13.9 MB 1.2 MB/s eta 0:00:0
4 ----- 10.2/13.9 MB 1.2 MB/s eta 0:00:0
4 ----- 10.2/13.9 MB 1.2 MB/s eta 0:00:0
3 ----- 10.3/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 10.4/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 10.5/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 10.6/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 10.6/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 10.7/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 10.8/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 10.9/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 11.0/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 11.1/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 11.2/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 11.2/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 11.3/13.9 MB 1.3 MB/s eta 0:00:0
3 ----- 11.4/13.9 MB 1.3 MB/s eta 0:00:0
2 ----- 11.5/13.9 MB 1.3 MB/s eta 0:00:0
2 ----- 11.5/13.9 MB 1.3 MB/s eta 0:00:0
2 ----- 11.6/13.9 MB 1.3 MB/s eta 0:00:0
2 ----- 11.7/13.9 MB 1.3 MB/s eta 0:00:0
2 ----- 11.8/13.9 MB 1.3 MB/s eta 0:00:0
2 ----- 11.9/13.9 MB 1.3 MB/s eta 0:00:0
2 ----- 12.0/13.9 MB 1.3 MB/s eta 0:00:0
2 ----- 12.1/13.9 MB 1.3 MB/s eta 0:00:0
2 ----- 12.1/13.9 MB 1.3 MB/s eta 0:00:0

```
2          ----- 12.2/13.9 MB 1.3 MB/s eta 0:00:0
2          ----- 12.3/13.9 MB 1.3 MB/s eta 0:00:0
2          ----- 12.4/13.9 MB 1.3 MB/s eta 0:00:0
2          ----- 12.5/13.9 MB 1.4 MB/s eta 0:00:0
2          ----- 12.5/13.9 MB 1.4 MB/s eta 0:00:0
2          ----- 12.6/13.9 MB 1.4 MB/s eta 0:00:0
1          ----- 12.7/13.9 MB 1.4 MB/s eta 0:00:0
1          ----- 12.8/13.9 MB 1.4 MB/s eta 0:00:0
1          ----- 12.9/13.9 MB 1.4 MB/s eta 0:00:0
1          ----- 13.0/13.9 MB 1.4 MB/s eta 0:00:0
1          ----- 13.1/13.9 MB 1.4 MB/s eta 0:00:0
1          ----- 13.2/13.9 MB 1.4 MB/s eta 0:00:0
1          ----- 13.3/13.9 MB 1.5 MB/s eta 0:00:0
1          ----- 13.4/13.9 MB 1.5 MB/s eta 0:00:0
1          ----- 13.5/13.9 MB 1.5 MB/s eta 0:00:0
1          ----- 13.6/13.9 MB 1.5 MB/s eta 0:00:0
1          ----- 13.7/13.9 MB 1.6 MB/s eta 0:00:0
1          ----- 13.8/13.9 MB 1.6 MB/s eta 0:00:0
1          ----- 13.9/13.9 MB 1.6 MB/s eta 0:00:0
1          ----- 13.9/13.9 MB 1.6 MB/s eta 0:00:0
0
Downloading transformers-4.56.1-py3-none-any.whl (11.6 MB)
----- 0.0/11.6 MB ? eta ------
----- 0.1/11.6 MB 1.7 MB/s eta 0:00:07
----- 0.1/11.6 MB 1.8 MB/s eta 0:00:07
----- 0.2/11.6 MB 2.2 MB/s eta 0:00:06
----- 0.3/11.6 MB 2.0 MB/s eta 0:00:06
----- 0.3/11.6 MB 2.0 MB/s eta 0:00:06
----- 0.4/11.6 MB 1.6 MB/s eta 0:00:08
----- 0.5/11.6 MB 1.6 MB/s eta 0:00:07
----- 0.6/11.6 MB 1.9 MB/s eta 0:00:06
----- 0.8/11.6 MB 2.0 MB/s eta 0:00:06
----- 0.9/11.6 MB 2.1 MB/s eta 0:00:06
----- 1.1/11.6 MB 2.3 MB/s eta 0:00:05
----- 1.2/11.6 MB 2.4 MB/s eta 0:00:05
----- 1.4/11.6 MB 2.3 MB/s eta 0:00:05
```

----- 1.6/11.6 MB 2.5 MB/s eta 0:00:04
----- 1.7/11.6 MB 2.6 MB/s eta 0:00:04
----- 1.8/11.6 MB 2.6 MB/s eta 0:00:04
----- 1.9/11.6 MB 2.6 MB/s eta 0:00:04
----- 2.1/11.6 MB 2.6 MB/s eta 0:00:04
----- 2.3/11.6 MB 2.6 MB/s eta 0:00:04
----- 2.4/11.6 MB 2.6 MB/s eta 0:00:04
----- 2.5/11.6 MB 2.6 MB/s eta 0:00:04
----- 2.7/11.6 MB 2.7 MB/s eta 0:00:04
----- 2.8/11.6 MB 2.7 MB/s eta 0:00:04
----- 2.9/11.6 MB 2.6 MB/s eta 0:00:04
----- 3.0/11.6 MB 2.6 MB/s eta 0:00:04
----- 3.2/11.6 MB 2.7 MB/s eta 0:00:04
----- 3.3/11.6 MB 2.7 MB/s eta 0:00:04
----- 3.4/11.6 MB 2.7 MB/s eta 0:00:04
----- 3.5/11.6 MB 2.7 MB/s eta 0:00:04
----- 3.7/11.6 MB 2.7 MB/s eta 0:00:03
----- 3.9/11.6 MB 2.7 MB/s eta 0:00:03
----- 3.9/11.6 MB 2.7 MB/s eta 0:00:03
----- 4.1/11.6 MB 2.7 MB/s eta 0:00:03
----- 4.2/11.6 MB 2.7 MB/s eta 0:00:03
----- 4.3/11.6 MB 2.7 MB/s eta 0:00:03
----- 4.5/11.6 MB 2.7 MB/s eta 0:00:03
----- 4.6/11.6 MB 2.7 MB/s eta 0:00:03
----- 4.7/11.6 MB 2.7 MB/s eta 0:00:03
----- 4.8/11.6 MB 2.7 MB/s eta 0:00:03
----- 5.0/11.6 MB 2.7 MB/s eta 0:00:03
----- 5.1/11.6 MB 2.7 MB/s eta 0:00:03
----- 5.3/11.6 MB 2.8 MB/s eta 0:00:03
----- 5.4/11.6 MB 2.7 MB/s eta 0:00:03
----- 5.4/11.6 MB 2.7 MB/s eta 0:00:03
----- 5.5/11.6 MB 2.7 MB/s eta 0:00:03
----- 5.6/11.6 MB 2.7 MB/s eta 0:00:03
----- 5.8/11.6 MB 2.7 MB/s eta 0:00:03
----- 5.9/11.6 MB 2.7 MB/s eta 0:00:03
----- 6.1/11.6 MB 2.7 MB/s eta 0:00:03
----- 6.1/11.6 MB 2.7 MB/s eta 0:00:03
----- 6.3/11.6 MB 2.7 MB/s eta 0:00:02
----- 6.4/11.6 MB 2.7 MB/s eta 0:00:02
----- 6.6/11.6 MB 2.7 MB/s eta 0:00:02
----- 6.7/11.6 MB 2.7 MB/s eta 0:00:02
----- 6.8/11.6 MB 2.7 MB/s eta 0:00:02
----- 7.0/11.6 MB 2.7 MB/s eta 0:00:02
----- 7.2/11.6 MB 2.7 MB/s eta 0:00:02
----- 7.4/11.6 MB 2.8 MB/s eta 0:00:02
----- 7.5/11.6 MB 2.7 MB/s eta 0:00:02
----- 7.6/11.6 MB 2.8 MB/s eta 0:00:02
----- 7.8/11.6 MB 2.8 MB/s eta 0:00:02
----- 7.9/11.6 MB 2.7 MB/s eta 0:00:02
----- 7.9/11.6 MB 2.7 MB/s eta 0:00:02
----- 8.1/11.6 MB 2.7 MB/s eta 0:00:02
----- 8.2/11.6 MB 2.8 MB/s eta 0:00:02
----- 8.4/11.6 MB 2.7 MB/s eta 0:00:02
----- 8.5/11.6 MB 2.7 MB/s eta 0:00:02
----- 8.7/11.6 MB 2.8 MB/s eta 0:00:02
----- 8.8/11.6 MB 2.8 MB/s eta 0:00:02

```
----- 8.9/11.6 MB 2.8 MB/s eta 0:00:01
----- 9.1/11.6 MB 2.8 MB/s eta 0:00:01
----- 9.3/11.6 MB 2.8 MB/s eta 0:00:01
----- 9.4/11.6 MB 2.8 MB/s eta 0:00:01
----- 9.6/11.6 MB 2.8 MB/s eta 0:00:01
----- 9.8/11.6 MB 2.8 MB/s eta 0:00:01
----- 9.9/11.6 MB 2.8 MB/s eta 0:00:01
----- 10.1/11.6 MB 2.9 MB/s eta 0:00:0
1 ----- 10.2/11.6 MB 2.8 MB/s eta 0:00:0
1 ----- 10.4/11.6 MB 2.9 MB/s eta 0:00:0
1 ----- 10.5/11.6 MB 2.9 MB/s eta 0:00:0
1 ----- 10.6/11.6 MB 2.9 MB/s eta 0:00:0
1 ----- 10.7/11.6 MB 2.9 MB/s eta 0:00:0
1 ----- 10.7/11.6 MB 2.9 MB/s eta 0:00:0
1 ----- 10.8/11.6 MB 2.9 MB/s eta 0:00:0
1 ----- 10.9/11.6 MB 2.9 MB/s eta 0:00:0
1 ----- 10.9/11.6 MB 2.9 MB/s eta 0:00:0
1 ----- 10.9/11.6 MB 2.8 MB/s eta 0:00:0
1 ----- 11.0/11.6 MB 2.8 MB/s eta 0:00:0
1 ----- 11.1/11.6 MB 2.8 MB/s eta 0:00:0
1 ----- 11.2/11.6 MB 2.8 MB/s eta 0:00:0
1 ----- 11.3/11.6 MB 2.7 MB/s eta 0:00:0
1 ----- 11.4/11.6 MB 2.6 MB/s eta 0:00:0
1 ----- 11.5/11.6 MB 2.6 MB/s eta 0:00:0
1 ----- 11.6/11.6 MB 2.6 MB/s eta 0:00:0
0
Downloading catalogue-2.0.10-py3-none-any.whl (17 kB)
Downloading cymem-2.0.11-cp312-cp312-win_amd64.whl (39 kB)
Downloading huggingface_hub-0.34.4-py3-none-any.whl (561 kB)
----- 0.0/561.5 kB ? eta -:--:-
----- 30.7/561.5 kB 1.3 MB/s eta 0:00:
```

```
01          ----- 71.7/561.5 kB 1.3 MB/s eta 0:00:  
01          ----- 122.9/561.5 kB 1.0 MB/s eta 0:0  
0:01          ----- 174.1/561.5 kB 1.2 MB/s eta 0:0  
0:01          ----- 235.5/561.5 kB 1.1 MB/s eta 0:0  
0:01          ----- 286.7/561.5 kB 1.1 MB/s eta 0:0  
0:01          ----- 389.1/561.5 kB 1.3 MB/s eta 0:0  
0:01          ----- 430.1/561.5 kB 1.2 MB/s eta 0:0  
0:01          ----- 460.8/561.5 kB 1.3 MB/s eta 0:0  
0:01          ----- 561.5/561.5 kB 1.3 MB/s eta 0:0  
0:00  
Downloading langcodes-3.5.0-py3-none-any.whl (182 kB)  
----- 0.0/183.0 kB ? eta -:---  
----- 30.7/183.0 kB 1.4 MB/s eta 0:00:  
01          ----- 71.7/183.0 kB 1.3 MB/s eta 0:00:  
01          ----- 122.9/183.0 kB 1.2 MB/s eta 0:0  
0:01          ----- 183.0/183.0 kB 1.2 MB/s eta 0:0  
0:00  
Downloading murmurhash-1.0.13-cp312-cp312-win_amd64.whl (24 kB)  
Downloading preshed-3.0.10-cp312-cp312-win_amd64.whl (116 kB)  
----- 0.0/116.7 kB ? eta -:---  
----- 10.2/116.7 kB ? eta -:---  
----- 41.0/116.7 kB 653.6 kB/s eta 0:0  
0:01          ----- 112.6/116.7 kB 939.4 kB/s eta 0:0  
0:01          ----- 116.7/116.7 kB 976.7 kB/s eta 0:0  
0:00  
Downloading pydantic-2.11.7-py3-none-any.whl (444 kB)  
----- 0.0/444.8 kB ? eta -:---  
----- 30.7/444.8 kB ? eta -:---  
----- 102.4/444.8 kB 1.5 MB/s eta 0:0  
0:01          ----- 174.1/444.8 kB 1.5 MB/s eta 0:0  
0:01          ----- 256.0/444.8 kB 1.8 MB/s eta 0:0  
0:01          ----- 317.4/444.8 kB 1.6 MB/s eta 0:0  
0:01          ----- 399.4/444.8 kB 1.7 MB/s eta 0:0  
0:01          ----- 444.8/444.8 kB 1.6 MB/s eta 0:0  
0:00  
Downloading pydantic_core-2.33.2-cp312-cp312-win_amd64.whl (2.0 MB)  
----- 0.0/2.0 MB ? eta -:---
```

```
----- 0.0/2.0 MB 1.3 MB/s eta 0:00:02
----- 0.1/2.0 MB 1.3 MB/s eta 0:00:02
----- 0.2/2.0 MB 1.5 MB/s eta 0:00:02
----- 0.3/2.0 MB 1.6 MB/s eta 0:00:02
----- 0.3/2.0 MB 1.6 MB/s eta 0:00:02
----- 0.4/2.0 MB 1.6 MB/s eta 0:00:02
----- 0.4/2.0 MB 1.4 MB/s eta 0:00:02
----- 0.4/2.0 MB 1.2 MB/s eta 0:00:02
----- 0.5/2.0 MB 1.2 MB/s eta 0:00:02
----- 0.5/2.0 MB 1.1 MB/s eta 0:00:02
----- 0.5/2.0 MB 1.1 MB/s eta 0:00:02
----- 0.6/2.0 MB 1.1 MB/s eta 0:00:02
----- 0.6/2.0 MB 1.1 MB/s eta 0:00:02
----- 0.7/2.0 MB 1.1 MB/s eta 0:00:02
----- 0.7/2.0 MB 1.0 MB/s eta 0:00:02
----- 0.8/2.0 MB 1.1 MB/s eta 0:00:02
----- 0.8/2.0 MB 1.1 MB/s eta 0:00:02
----- 0.9/2.0 MB 1.1 MB/s eta 0:00:01
----- 0.9/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.0/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.1/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.1/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.2/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.3/2.0 MB 1.2 MB/s eta 0:00:01
----- 1.3/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.3/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.4/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.4/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.4/2.0 MB 1.0 MB/s eta 0:00:01
----- 1.5/2.0 MB 1.0 MB/s eta 0:00:01
----- 1.5/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.6/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.7/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.7/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.8/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.8/2.0 MB 1.1 MB/s eta 0:00:01
----- 1.9/2.0 MB 1.1 MB/s eta 0:00:01
----- 2.0/2.0 MB 1.1 MB/s eta 0:00:00

Downloading safetensors-0.6.2-cp38-abi3-win_amd64.whl (320 kB)
----- 0.0/320.2 kB ? eta -:-:-
----- 61.4/320.2 kB 3.4 MB/s eta 0:00:
01
----- 122.9/320.2 kB 1.8 MB/s eta 0:0
0:01
----- 184.3/320.2 kB 1.4 MB/s eta 0:0
0:01
----- 256.0/320.2 kB 1.6 MB/s eta 0:0
0:01
----- 320.2/320.2 kB 1.5 MB/s eta 0:0
0:00

Downloading spacy_legacy-3.0.12-py2.py3-none-any.whl (29 kB)
Downloading spacy_loggers-1.0.5-py3-none-any.whl (22 kB)
Downloading srsly-2.5.1-cp312-cp312-win_amd64.whl (632 kB)
----- 0.0/632.6 kB ? eta -:-:-
----- 41.0/632.6 kB 991.0 kB/s eta 0:0
```

```
0:01          ----- 92.2/632.6 kB 1.1 MB/s eta 0:00:  
01          ----- 174.1/632.6 kB 1.3 MB/s eta 0:0  
0:01          ----- 235.5/632.6 kB 1.3 MB/s eta 0:0  
0:01          ----- 297.0/632.6 kB 1.3 MB/s eta 0:0  
0:01          ----- 348.2/632.6 kB 1.3 MB/s eta 0:0  
0:01          ----- 399.4/632.6 kB 1.3 MB/s eta 0:0  
0:01          ----- 460.8/632.6 kB 1.3 MB/s eta 0:0  
0:01          ----- 512.0/632.6 kB 1.3 MB/s eta 0:0  
0:01          ----- 563.2/632.6 kB 1.3 MB/s eta 0:0  
0:01          ----- 632.6/632.6 kB 1.3 MB/s eta 0:0  
0:00  
Downloading thinc-8.3.6-cp312-cp312-win_amd64.whl (1.7 MB)  
----- 0.0/1.7 MB ? eta -:-:-  
----- 0.1/1.7 MB 1.7 MB/s eta 0:00:02  
----- 0.1/1.7 MB 1.3 MB/s eta 0:00:02  
----- 0.1/1.7 MB 1.2 MB/s eta 0:00:02  
----- 0.2/1.7 MB 1.4 MB/s eta 0:00:02  
----- 0.3/1.7 MB 1.5 MB/s eta 0:00:01  
----- 0.5/1.7 MB 1.7 MB/s eta 0:00:01  
----- 0.5/1.7 MB 1.7 MB/s eta 0:00:01  
----- 0.6/1.7 MB 1.7 MB/s eta 0:00:01  
----- 0.7/1.7 MB 1.8 MB/s eta 0:00:01  
----- 0.8/1.7 MB 1.7 MB/s eta 0:00:01  
----- 0.9/1.7 MB 1.9 MB/s eta 0:00:01  
----- 1.0/1.7 MB 1.9 MB/s eta 0:00:01  
----- 1.1/1.7 MB 1.9 MB/s eta 0:00:01  
----- 1.2/1.7 MB 1.9 MB/s eta 0:00:01  
----- 1.3/1.7 MB 1.9 MB/s eta 0:00:01  
----- 1.4/1.7 MB 1.9 MB/s eta 0:00:01  
----- 1.5/1.7 MB 1.9 MB/s eta 0:00:01  
----- 1.5/1.7 MB 1.8 MB/s eta 0:00:01  
----- 1.6/1.7 MB 1.9 MB/s eta 0:00:01  
----- 1.7/1.7 MB 1.9 MB/s eta 0:00:01  
----- 1.7/1.7 MB 1.9 MB/s eta 0:00:00  
Downloading numpy-2.3.2-cp312-cp312-win_amd64.whl (12.8 MB)  
----- 0.0/12.8 MB ? eta -:-:-  
----- 0.1/12.8 MB 1.7 MB/s eta 0:00:08  
----- 0.1/12.8 MB 1.1 MB/s eta 0:00:13  
----- 0.2/12.8 MB 1.1 MB/s eta 0:00:12  
----- 0.3/12.8 MB 1.3 MB/s eta 0:00:10  
----- 0.3/12.8 MB 1.4 MB/s eta 0:00:10  
----- 0.4/12.8 MB 1.6 MB/s eta 0:00:08  
----- 0.5/12.8 MB 1.7 MB/s eta 0:00:08  
----- 0.6/12.8 MB 1.7 MB/s eta 0:00:08  
----- 0.7/12.8 MB 1.7 MB/s eta 0:00:07  
----- 0.8/12.8 MB 1.7 MB/s eta 0:00:08
```


----- 5.5/12.8 MB 1.8 MB/s eta 0:00:05
----- 5.6/12.8 MB 1.8 MB/s eta 0:00:05
----- 5.6/12.8 MB 1.8 MB/s eta 0:00:05
----- 5.7/12.8 MB 1.8 MB/s eta 0:00:05
----- 5.8/12.8 MB 1.8 MB/s eta 0:00:04
----- 5.9/12.8 MB 1.7 MB/s eta 0:00:04
----- 5.9/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.0/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.1/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.2/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.2/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.3/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.3/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.4/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.5/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.5/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.6/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.7/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.8/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.8/12.8 MB 1.7 MB/s eta 0:00:04
----- 6.9/12.8 MB 1.7 MB/s eta 0:00:04
----- 7.0/12.8 MB 1.7 MB/s eta 0:00:04
----- 7.1/12.8 MB 1.7 MB/s eta 0:00:04
----- 7.2/12.8 MB 1.7 MB/s eta 0:00:04
----- 7.3/12.8 MB 1.7 MB/s eta 0:00:04
----- 7.4/12.8 MB 1.7 MB/s eta 0:00:04
----- 7.5/12.8 MB 1.7 MB/s eta 0:00:04
----- 7.7/12.8 MB 1.8 MB/s eta 0:00:03
----- 7.9/12.8 MB 1.7 MB/s eta 0:00:03
----- 8.1/12.8 MB 1.8 MB/s eta 0:00:03
----- 8.2/12.8 MB 1.8 MB/s eta 0:00:03
----- 8.4/12.8 MB 1.8 MB/s eta 0:00:03
----- 8.5/12.8 MB 1.8 MB/s eta 0:00:03
----- 8.9/12.8 MB 1.9 MB/s eta 0:00:03
----- 9.1/12.8 MB 1.9 MB/s eta 0:00:02
----- 9.2/12.8 MB 1.9 MB/s eta 0:00:02
----- 9.2/12.8 MB 1.9 MB/s eta 0:00:02
----- 9.3/12.8 MB 1.9 MB/s eta 0:00:02
----- 9.5/12.8 MB 1.9 MB/s eta 0:00:02
----- 9.6/12.8 MB 1.9 MB/s eta 0:00:02
----- 9.7/12.8 MB 1.9 MB/s eta 0:00:02
----- 9.7/12.8 MB 1.9 MB/s eta 0:00:02
----- 9.8/12.8 MB 1.9 MB/s eta 0:00:02
----- 9.9/12.8 MB 1.9 MB/s eta 0:00:02
----- 10.0/12.8 MB 1.9 MB/s eta 0:00:0

2 ----- 10.1/12.8 MB 1.9 MB/s eta 0:00:0
2 ----- 10.2/12.8 MB 1.9 MB/s eta 0:00:0
2 ----- 10.3/12.8 MB 1.9 MB/s eta 0:00:0
2 ----- 10.4/12.8 MB 1.9 MB/s eta 0:00:0

```
2          ----- 10.6/12.8 MB 1.9 MB/s eta 0:00:0
2          ----- 10.7/12.8 MB 1.9 MB/s eta 0:00:0
2          ----- 10.9/12.8 MB 1.9 MB/s eta 0:00:0
1          ----- 10.9/12.8 MB 1.9 MB/s eta 0:00:0
1          ----- 11.1/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 11.2/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 11.3/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 11.4/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 11.6/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 11.6/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 11.7/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 11.8/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 12.0/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 12.1/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 12.2/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 12.3/12.8 MB 2.0 MB/s eta 0:00:0
1          ----- 12.4/12.8 MB 2.1 MB/s eta 0:00:0
1          ----- 12.6/12.8 MB 2.1 MB/s eta 0:00:0
1          ----- 12.7/12.8 MB 2.1 MB/s eta 0:00:0
1          ----- 12.8/12.8 MB 2.1 MB/s eta 0:00:0
1          ----- 12.8/12.8 MB 2.1 MB/s eta 0:00:0
0
Downloading tokenizers-0.22.0-cp39-abi3-win_amd64.whl (2.7 MB)
----- 0.0/2.7 MB ? eta -:-----
----- 0.1/2.7 MB 3.0 MB/s eta 0:00:01
----- 0.1/2.7 MB 1.4 MB/s eta 0:00:02
----- 0.2/2.7 MB 1.4 MB/s eta 0:00:02
----- 0.2/2.7 MB 1.4 MB/s eta 0:00:02
----- 0.3/2.7 MB 1.5 MB/s eta 0:00:02
----- 0.4/2.7 MB 1.4 MB/s eta 0:00:02
----- 0.5/2.7 MB 1.5 MB/s eta 0:00:02
----- 0.6/2.7 MB 1.6 MB/s eta 0:00:02
----- 0.7/2.7 MB 1.7 MB/s eta 0:00:02
----- 0.8/2.7 MB 1.8 MB/s eta 0:00:02
----- 0.9/2.7 MB 1.8 MB/s eta 0:00:02
```

```
----- 1.0/2.7 MB 1.8 MB/s eta 0:00:01  
----- 1.1/2.7 MB 1.9 MB/s eta 0:00:01  
----- 1.2/2.7 MB 2.0 MB/s eta 0:00:01  
----- 1.4/2.7 MB 2.0 MB/s eta 0:00:01  
----- 1.5/2.7 MB 2.0 MB/s eta 0:00:01  
----- 1.5/2.7 MB 2.0 MB/s eta 0:00:01  
----- 1.7/2.7 MB 2.0 MB/s eta 0:00:01  
----- 1.8/2.7 MB 2.0 MB/s eta 0:00:01  
----- 1.9/2.7 MB 2.1 MB/s eta 0:00:01  
----- 2.0/2.7 MB 2.1 MB/s eta 0:00:01  
----- 2.2/2.7 MB 2.1 MB/s eta 0:00:01  
----- 2.3/2.7 MB 2.2 MB/s eta 0:00:01  
----- 2.4/2.7 MB 2.2 MB/s eta 0:00:01  
----- 2.6/2.7 MB 2.2 MB/s eta 0:00:01  
----- 2.7/2.7 MB 2.2 MB/s eta 0:00:00  
Downloading typer-0.17.4-py3-none-any.whl (46 kB)  
----- 0.0/46.6 kB ? eta -:---  
----- 46.6/46.6 kB 2.3 MB/s eta 0:00:00  
0  
Downloading wasabi-1.1.3-py3-none-any.whl (27 kB)  
Downloading weasel-0.4.1-py3-none-any.whl (50 kB)  
----- 0.0/50.3 kB ? eta -:---  
----- 50.3/50.3 kB 2.5 MB/s eta 0:00:00  
0  
Downloading annotated_types-0.7.0-py3-none-any.whl (13 kB)  
Downloading blis-1.3.0-cp312-cp312-win_amd64.whl (6.3 MB)  
----- 0.0/6.3 MB ? eta -:---  
----- 0.1/6.3 MB 2.0 MB/s eta 0:00:04  
----- 0.2/6.3 MB 2.8 MB/s eta 0:00:03  
----- 0.3/6.3 MB 2.0 MB/s eta 0:00:04  
----- 0.3/6.3 MB 2.0 MB/s eta 0:00:04  
----- 0.4/6.3 MB 1.7 MB/s eta 0:00:04  
----- 0.5/6.3 MB 1.9 MB/s eta 0:00:04  
----- 0.6/6.3 MB 1.9 MB/s eta 0:00:03  
----- 0.7/6.3 MB 2.0 MB/s eta 0:00:03  
----- 0.9/6.3 MB 2.1 MB/s eta 0:00:03  
----- 1.0/6.3 MB 2.2 MB/s eta 0:00:03  
----- 1.1/6.3 MB 2.2 MB/s eta 0:00:03  
----- 1.2/6.3 MB 2.2 MB/s eta 0:00:03  
----- 1.3/6.3 MB 2.2 MB/s eta 0:00:03  
----- 1.5/6.3 MB 2.3 MB/s eta 0:00:03  
----- 1.6/6.3 MB 2.3 MB/s eta 0:00:03  
----- 1.7/6.3 MB 2.3 MB/s eta 0:00:02  
----- 1.8/6.3 MB 1.7 MB/s eta 0:00:03  
----- 1.9/6.3 MB 1.7 MB/s eta 0:00:03  
----- 1.9/6.3 MB 1.7 MB/s eta 0:00:03  
----- 1.9/6.3 MB 1.6 MB/s eta 0:00:03  
----- 2.0/6.3 MB 1.6 MB/s eta 0:00:03  
----- 2.0/6.3 MB 1.5 MB/s eta 0:00:03  
----- 2.1/6.3 MB 1.5 MB/s eta 0:00:03
```



```
----- 5.4/6.3 MB 1.4 MB/s eta 0:00:01
----- 5.6/6.3 MB 1.4 MB/s eta 0:00:01
----- 5.6/6.3 MB 1.4 MB/s eta 0:00:01
----- 5.7/6.3 MB 1.4 MB/s eta 0:00:01
----- 5.8/6.3 MB 1.4 MB/s eta 0:00:01
----- 5.9/6.3 MB 1.4 MB/s eta 0:00:01
----- 5.9/6.3 MB 1.4 MB/s eta 0:00:01
----- 5.9/6.3 MB 1.4 MB/s eta 0:00:01
----- 6.0/6.3 MB 1.3 MB/s eta 0:00:01
----- 6.1/6.3 MB 1.3 MB/s eta 0:00:01
----- 6.1/6.3 MB 1.3 MB/s eta 0:00:01
----- 6.3/6.3 MB 1.3 MB/s eta 0:00:00
Downloading cloudpathlib-0.22.0-py3-none-any.whl (61 kB)
----- 0.0/61.5 kB ? eta -:--:-
----- 30.7/61.5 kB 1.3 MB/s eta 0:00:0
1
----- 61.5/61.5 kB 1.7 MB/s eta 0:00:0
0
Downloading confection-0.1.5-py3-none-any.whl (35 kB)
Downloading language_data-1.3.0-py3-none-any.whl (5.4 MB)
----- 0.0/5.4 MB ? eta -:--:-
----- 0.0/5.4 MB 1.9 MB/s eta 0:00:03
----- 0.2/5.4 MB 1.8 MB/s eta 0:00:03
----- 0.2/5.4 MB 1.8 MB/s eta 0:00:03
----- 0.3/5.4 MB 1.7 MB/s eta 0:00:03
----- 0.4/5.4 MB 1.8 MB/s eta 0:00:03
----- 0.5/5.4 MB 1.7 MB/s eta 0:00:03
----- 0.5/5.4 MB 1.7 MB/s eta 0:00:03
----- 0.6/5.4 MB 1.7 MB/s eta 0:00:03
----- 0.7/5.4 MB 1.8 MB/s eta 0:00:03
----- 0.8/5.4 MB 1.8 MB/s eta 0:00:03
----- 0.9/5.4 MB 1.9 MB/s eta 0:00:03
----- 1.0/5.4 MB 1.9 MB/s eta 0:00:03
----- 1.1/5.4 MB 1.9 MB/s eta 0:00:03
----- 1.2/5.4 MB 1.9 MB/s eta 0:00:03
----- 1.3/5.4 MB 1.9 MB/s eta 0:00:03
----- 1.4/5.4 MB 1.9 MB/s eta 0:00:03
----- 1.5/5.4 MB 1.9 MB/s eta 0:00:03
----- 1.6/5.4 MB 2.0 MB/s eta 0:00:02
----- 1.7/5.4 MB 1.9 MB/s eta 0:00:02
----- 1.8/5.4 MB 2.0 MB/s eta 0:00:02
----- 1.9/5.4 MB 2.0 MB/s eta 0:00:02
----- 2.0/5.4 MB 2.0 MB/s eta 0:00:02
----- 2.1/5.4 MB 2.0 MB/s eta 0:00:02
----- 2.2/5.4 MB 1.9 MB/s eta 0:00:02
----- 2.3/5.4 MB 1.9 MB/s eta 0:00:02
----- 2.3/5.4 MB 1.9 MB/s eta 0:00:02
----- 2.4/5.4 MB 1.9 MB/s eta 0:00:02
----- 2.5/5.4 MB 1.9 MB/s eta 0:00:02
----- 2.6/5.4 MB 1.9 MB/s eta 0:00:02
----- 2.7/5.4 MB 1.9 MB/s eta 0:00:02
----- 2.8/5.4 MB 1.9 MB/s eta 0:00:02
----- 2.9/5.4 MB 1.9 MB/s eta 0:00:02
```

```
----- 3.0/5.4 MB 1.9 MB/s eta 0:00:02
----- 3.1/5.4 MB 2.0 MB/s eta 0:00:02
----- 3.2/5.4 MB 1.9 MB/s eta 0:00:02
----- 3.3/5.4 MB 1.9 MB/s eta 0:00:02
----- 3.3/5.4 MB 1.9 MB/s eta 0:00:02
----- 3.4/5.4 MB 1.9 MB/s eta 0:00:02
----- 3.5/5.4 MB 1.9 MB/s eta 0:00:01
----- 3.5/5.4 MB 1.9 MB/s eta 0:00:01
----- 3.6/5.4 MB 1.9 MB/s eta 0:00:01
----- 3.7/5.4 MB 1.9 MB/s eta 0:00:01
----- 3.8/5.4 MB 1.9 MB/s eta 0:00:01
----- 3.9/5.4 MB 1.9 MB/s eta 0:00:01
----- 3.9/5.4 MB 1.9 MB/s eta 0:00:01
----- 4.0/5.4 MB 1.9 MB/s eta 0:00:01
----- 4.1/5.4 MB 1.9 MB/s eta 0:00:01
----- 4.2/5.4 MB 1.9 MB/s eta 0:00:01
----- 4.2/5.4 MB 1.9 MB/s eta 0:00:01
----- 4.3/5.4 MB 1.8 MB/s eta 0:00:01
----- 4.4/5.4 MB 1.8 MB/s eta 0:00:01
----- 4.5/5.4 MB 1.8 MB/s eta 0:00:01
----- 4.6/5.4 MB 1.8 MB/s eta 0:00:01
----- 4.6/5.4 MB 1.8 MB/s eta 0:00:01
----- 4.7/5.4 MB 1.9 MB/s eta 0:00:01
----- 4.8/5.4 MB 1.8 MB/s eta 0:00:01
----- 4.9/5.4 MB 1.8 MB/s eta 0:00:01
----- 5.0/5.4 MB 1.8 MB/s eta 0:00:01
----- 5.1/5.4 MB 1.8 MB/s eta 0:00:01
----- 5.2/5.4 MB 1.8 MB/s eta 0:00:01
----- 5.2/5.4 MB 1.8 MB/s eta 0:00:01
----- 5.3/5.4 MB 1.8 MB/s eta 0:00:01
----- 5.4/5.4 MB 1.8 MB/s eta 0:00:01
----- 5.4/5.4 MB 1.8 MB/s eta 0:00:00

Downloading shellingham-1.5.4-py2.py3-none-any.whl (9.8 kB)
Downloading smart_open-7.3.0.post1-py3-none-any.whl (61 kB)
----- 0.0/61.9 kB ? eta -:-----
----- 61.9/61.9 kB 1.7 MB/s eta 0:00:00
0
Downloading typing_inspection-0.4.1-py3-none-any.whl (14 kB)
Downloading marisa_trie-1.3.1-cp312-cp312-win_amd64.whl (138 kB)
----- 0.0/138.6 kB ? eta -:-----
----- 41.0/138.6 kB 1.9 MB/s eta 0:00:
01
----- 71.7/138.6 kB 991.0 kB/s eta 0:0
0:01
----- 138.6/138.6 kB 1.0 MB/s eta 0:0
0:00
Installing collected packages: cymem, wasabi, typing-inspection, spacy-logge
rs, spacy-legacy, smart-open, shellingham, safetensors, pydantic-core, numpy,
murmurhash, marisa-trie, cloudpathlib, catalogue, annotated-types, srsly,
pydantic, preshed, language-data, huggingface-hub, blis, typer, tokenizers,
langcodes, confection, weasel, transformers, thinc, spacy
Attempting uninstall: numpy
Found existing installation: numpy 1.26.3
Uninstalling numpy-1.26.3:
Successfully uninstalled numpy-1.26.3
Successfully installed annotated-types-0.7.0 blis-1.3.0 catalogue-2.0.10 clo
```

```
udpathlib-0.22.0 confection-0.1.5 cymem-2.0.11 huggingface-hub-0.34.4 langco  
des-3.5.0 language-data-1.3.0 marisa-trie-1.3.1 murmurhash-1.0.13 numpy-2.3.  
2 preshed-3.0.10 pydantic-2.11.7 pydantic-core-2.33.2 safetensors-0.6.2 shel  
lingham-1.5.4 smart-open-7.3.0.post1 spacy-3.8.7 spacy-legacy-3.0.12 spacy-l  
oggers-1.0.5 srsly-2.5.1 thinc-8.3.6 tokenizers-0.22.0 transformers-4.56.1 t  
yper-0.17.4 typing-inspection-0.4.1 wasabi-1.1.3 weasel-0.4.1
```

```
WARNING: Failed to remove contents in a temporary directory 'C:\Users\Osca  
r Escriba\AppData\Local\Programs\Python\Python312\Lib\site-packages\~umpy.li  
bs'.
```

```
You can safely remove it manually.
```

```
WARNING: Failed to remove contents in a temporary directory 'C:\Users\Osca  
r Escriba\AppData\Local\Programs\Python\Python312\Lib\site-packages\~umpy'.
```

```
You can safely remove it manually.
```

```
ERROR: pip's dependency resolver does not currently take into account all th  
e packages that are installed. This behaviour is the source of the following  
dependency conflicts.
```

```
contourpy 1.2.0 requires numpy<2.0,>=1.20, but you have numpy 2.3.2 which is  
incompatible.
```

```
numba 0.60.0 requires numpy<2.1,>=1.22, but you have numpy 2.3.2 which is in  
compatible.
```

```
scipy 1.13.0 requires numpy<2.3,>=1.22.4, but you have numpy 2.3.2 which is  
incompatible.
```

```
tensorflow 2.19.0 requires numpy<2.2.0,>=1.26.0, but you have numpy 2.3.2 wh  
ich is incompatible.
```

```
[notice] A new release of pip is available: 24.0 -> 25.2
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
In [1]: import pandas as pd  
import numpy as np  
import json  
import re  
import networkx as nx  
import matplotlib.pyplot as plt  
import seaborn as sns  
from collections import Counter  
from wordcloud import WordCloud  
import nltk  
from nltk.corpus import stopwords  
from nltk.tokenize import word_tokenize  
import warnings  
warnings.filterwarnings('ignore')  
  
nltk.download('stopwords', quiet=True)  
nltk.download('punkt', quiet=True)
```

```
Out[1]: True
```

```
In [2]: !pip install chardet
```

```
Collecting chardet
  Downloading chardet-5.2.0-py3-none-any.whl.metadata (3.4 kB)
  Downloading chardet-5.2.0-py3-none-any.whl (199 kB)
    -----
      0.0/199.4 kB ? eta -:-:--
    -----
      0.0/199.4 kB ? eta -:-:--
    -----
      10.2/199.4 kB ? eta -:-:--
    -----
      30.7/199.4 kB 325.1 kB/s eta 0:0
0:01
    -----
      81.9/199.4 kB 508.4 kB/s eta 0:0
0:01
    -----
      112.6/199.4 kB 595.3 kB/s eta 0:0
0:01
    -----
      143.4/199.4 kB 607.9 kB/s eta 0:0
0:01
    -----
      194.6/199.4 kB 653.6 kB/s eta 0:0
0:01
    -----
      199.4/199.4 kB 636.9 kB/s eta 0:0
0:00
Installing collected packages: chardet
Successfully installed chardet-5.2.0
[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
In [3]: DATASET_CHOICE = "traficogt"

if DATASET_CHOICE == "traficogt":
    filename = "traficogt.txt"
    dataset_name = "@TráficoGT"
elif DATASET_CHOICE == "tioberny":
    filename = "tioberny.txt"
    dataset_name = "@BArevalodeLeon"
```

```
In [4]: import chardet

tweets = []
failed_lines = 0
successful_lines = 0
with open(filename, 'rb') as file:
    raw_data = file.read()
    encoding_result = chardet.detect(raw_data)
    detected_encoding = encoding_result['encoding']

print(f"Encoding detectado: {detected_encoding}")
try:
    with open(filename, 'r', encoding=detected_encoding) as file:
        for line_num, line in enumerate(file, 1):
            line = line.strip()
            if line:
                try:
                    tweet = json.loads(line)
                    tweets.append(tweet)
                    successful_lines += 1
                except json.JSONDecodeError as e:
                    failed_lines += 1
                    if failed_lines <= 5:
```

```

        print(f"Error en línea {line_num}: {str(e)[:100]}...")
except Exception as e:
    print(f"Error al abrir archivo: {e}")
    encodings_to_try = ['utf-8-sig', 'utf-8', 'latin-1', 'cp1252']

    for encoding in encodings_to_try:
        try:
            tweets = []
            with open(filename, 'r', encoding=encoding) as file:
                for line in file:
                    line = line.strip()
                    if line:
                        try:
                            tweet = json.loads(line)
                            tweets.append(tweet)
                        except json.JSONDecodeError:
                            continue
            if tweets:
                print(f"Cargado con encoding: {encoding}")
                break
        except:
            continue

print(f"Total de tweets cargados: {len(tweets)}")
print(f"Líneas procesadas exitosamente: {successful_lines}")
print(f"Líneas con errores: {failed_lines}")

if tweets:
    print("Archivo cargado correctamente")
else:
    print("No se pudieron cargar los tweets")

df_tweets = pd.DataFrame(tweets)

```

Encoding detectado: UTF-16
 Error en línea 5605: Expecting ',' delimiter: line 1 column 4314 (char 431
 3)...
 Total de tweets cargados: 5604
 Líneas procesadas exitosamente: 5604
 Líneas con errores: 1
 Archivo cargado correctamente

In [5]:

```

def clean_text(text):
    if pd.isna(text) or not isinstance(text, str):
        return ""

    text = text.lower()
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)
    text = re.sub(r'@\w+', '', text)
    text = re.sub(r'#\w+', '', text)
    text = re.sub(r'[^a-zA-ZáéíóúñÁÉÍÓÚÑ\s]', '', text)
    text = re.sub(r'\s+', ' ', text).strip()

    return text

def extract_mentions(text):

```

```

if pd.isna(text) or not isinstance(text, str):
    return []
return re.findall(r'@(\w+)', text.lower())

def extract_hashtags(text):
    if pd.isna(text) or not isinstance(text, str):
        return []
    return re.findall(r'#(\w+)', text.lower())

def extract_urls(text):
    if pd.isna(text) or not isinstance(text, str):
        return []
    return re.findall(r'http\S+|www\S+|https\S+', text)

```

```

In [6]: print("Columnas disponibles en el DataFrame:")
print(df_tweets.columns.tolist())
print(f"\nShape del DataFrame: {df_tweets.shape}")

if len(df_tweets) > 0:
    print("\nPrimeras columnas del primer tweet:")
    for col in df_tweets.columns[:10]:
        sample_value = df_tweets[col].iloc[0]
        print(f"  {col}: {type(sample_value)} - {str(sample_value)[:100]}...")
text_column = None
possible_text_columns = ['rawContent', 'text', 'full_text', 'content', 'twee

for col in possible_text_columns:
    if col in df_tweets.columns:
        text_column = col
        print(f"\nColumna de texto encontrada: '{col}'")
        break

if text_column is None:
    print("\nNo se encontró columna de texto. Columnas disponibles:")
    for col in df_tweets.columns:
        print(f"  - {col}")
    print("\nRevisar manualmente y definir text_column")

```

Columnas disponibles en el DataFrame:

```
['id', 'id_str', 'url', 'date', 'user', 'lang', 'rawContent', 'replyCount', 'retweetCount', 'likeCount', 'quoteCount', 'bookmarkedCount', 'conversationId', 'conversationIdStr', 'hashtags', 'cashtags', 'mentionedUsers', 'links', 'viewCount', 'retweetedTweet', 'quotedTweet', 'place', 'coordinates', 'inReplyToTweetId', 'inReplyToTweetIdStr', 'inReplyToUser', 'source', 'sourceUrl', 'sourceLabel', 'media', 'card', '_type']
```

Shape del DataFrame: (5604, 32)

Primeras columnas del primer tweet:

```
id: <class 'numpy.int64'> - 1834236045598056867...
id_str: <class 'str'> - 1834236045598056867...
url: <class 'str'> - https://x.com/traficogt/status/1834236045598056867...
date: <class 'str'> - 2024-09-12 14:22:06+00:00...
user: <class 'dict'> - {'id': 93938886, 'id_str': '93938886', 'url': 'https://x.com/traficogt', 'username': 'traficogt', 'd...
lang: <class 'str'> - es...
rawContent: <class 'str'> - Es comprensible la resolución... El ruso sabe de engrasar maquinaria....
replyCount: <class 'numpy.int64'> - 0...
retweetCount: <class 'numpy.int64'> - 0...
likeCount: <class 'numpy.int64'> - 1...
```

Columna de texto encontrada: 'rawContent'

```
In [7]: df_tweets['text_original'] = df_tweets['rawContent']
df_tweets['text_clean'] = df_tweets['rawContent'].apply(clean_text)
df_tweets['mentions'] = df_tweets['rawContent'].apply(extract_mentions)
df_tweets['hashtags'] = df_tweets['rawContent'].apply(extract_hashtags)
df_tweets['urls'] = df_tweets['rawContent'].apply(extract_urls)
df_tweets['mention_count'] = df_tweets['mentions'].apply(len)
df_tweets['hashtag_count'] = df_tweets['hashtags'].apply(len)
df_tweets['url_count'] = df_tweets['urls'].apply(len)

print(f"Procesamiento completado:")
print(f" - Tweets totales: {len(df_tweets)}")
print(f" - Tweets con menciones: {len(df_tweets[df_tweets['mention_count'] > 0])}")
print(f" - Tweets con hashtags: {len(df_tweets[df_tweets['hashtag_count'] > 0])}")
print(f" - Tweets con URLs: {len(df_tweets[df_tweets['url_count'] > 0])}")

if len(df_tweets) > 0:
    sample_idx = 0
    print(f"\nEjemplo de procesamiento (tweet {sample_idx + 1}):")
    print(f"Original: {df_tweets['text_original'].iloc[sample_idx]}")
    print(f"Limpio: {df_tweets['text_clean'].iloc[sample_idx]}")
    print(f"Menciones: {df_tweets['mentions'].iloc[sample_idx]}")
    print(f"Hashtags: {df_tweets['hashtags'].iloc[sample_idx]}")
    print(f"URLs: {df_tweets['urls'].iloc[sample_idx]}")
```

Procesamiento completado:

- Tweets totales: 5604
- Tweets con menciones: 4502
- Tweets con hashtags: 409
- Tweets con URLs: 1007

Ejemplo de procesamiento (tweet 1):

Original: Es comprensible la resolución... El ruso sabe de engrasar maquinaria.

Limpio: es comprensible la resolución el ruso sabe de engrasar maquinaria

Menciones: []

Hashtags: []

URLs: []

```
In [8]: def extract_user_info(tweet_data):  
    user_info = {}  
    if 'user' in tweet_data:  
        user_data = tweet_data['user']  
        user_info['user_id'] = user_data.get('id_str', user_data.get('id', 'unknown'))  
        user_info['username'] = user_data.get('screen_name', 'unknown')  
        user_info['display_name'] = user_data.get('name', 'unknown')  
        user_info['followers_count'] = user_data.get('followers_count', 0)  
        user_info['friends_count'] = user_data.get('friends_count', 0)  
    else:  
        user_info['user_id'] = tweet_data.get('user_id', 'unknown')  
        user_info['username'] = tweet_data.get('username', 'unknown')  
        user_info['display_name'] = tweet_data.get('display_name', 'unknown')  
        user_info['followers_count'] = tweet_data.get('followers_count', 0)  
        user_info['friends_count'] = tweet_data.get('friends_count', 0)  
    return user_info
```

```
In [9]: def extract_engagement_metrics(tweet_data):  
    metrics = {}  
    metrics['retweet_count'] = tweet_data.get('retweet_count', 0)  
    metrics['favorite_count'] = tweet_data.get('favorite_count', 0)  
    metrics['reply_count'] = tweet_data.get('reply_count', 0)  
    metrics['is_retweet'] = 'retweeted_status' in tweet_data  
    return metrics
```

```
In [10]: user_info_list = [extract_user_info(tweet) for tweet in tweets]  
user_df = pd.DataFrame(user_info_list)  
  
engagement_list = [extract_engagement_metrics(tweet) for tweet in tweets]  
engagement_df = pd.DataFrame(engagement_list)  
  
df_tweets = pd.concat([df_tweets, user_df, engagement_df], axis=1)
```

```
In [11]: original_count = len(df_tweets)  
df_tweets = df_tweets[df_tweets['text_clean'].str.len() > 3]  
df_tweets = df_tweets.drop_duplicates(subset=['text_clean'])  
final_count = len(df_tweets)
```

```
In [12]: df_tweets['username_normalized'] = df_tweets['username'].str.lower().str.str  
df_tweets['mentions_normalized'] = df_tweets['mentions'].apply(  
    lambda x: x.replace('&', '&').replace('<', '<').replace('>', '>').replace('&lt;', '<').replace('&gt;', '>'))
```

```
lambda x: [mention.lower().strip() for mention in x] if x else []
)
```

```
In [16]: import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to C:\Users\Oscar
[nltk_data]     Escriba\AppData\Roaming\nltk_data...
[nltk_data]     Package punkt is already up-to-date!
```

```
Out[16]: True
```

```
In [20]: import nltk
nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('stopwords')

stop_words_spanish = set(stopwords.words('spanish'))
stop_words_english = set(stopwords.words('english'))
stop_words = stop_words_spanish.union(stop_words_english)

def remove_stopwords(text):
    if not text:
        return ""
    tokens = word_tokenize(text)
    filtered_tokens = [word for word in tokens if word not in stop_words and
                       word.isalpha()]
    return ' '.join(filtered_tokens)

df_tweets['text_processed'] = df_tweets['text_clean'].apply(remove_stopwords)
```

```
[nltk_data] Downloading package punkt to C:\Users\Oscar
[nltk_data]     Escriba\AppData\Roaming\nltk_data...
[nltk_data]     Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to C:\Users\Oscar
[nltk_data]     Escriba\AppData\Roaming\nltk_data...
[nltk_data]     Unzipping tokenizers\punkt_tab.zip.
[nltk_data] Downloading package stopwords to C:\Users\Oscar
[nltk_data]     Escriba\AppData\Roaming\nltk_data...
[nltk_data]     Package stopwords is already up-to-date!
```

```
In [21]: interactions_list = []

for idx, row in df_tweets.iterrows():
    source_user = row['username_normalized']

    if row['is_retweet']:
        interactions_list.append({
            'source': source_user,
            'target': 'retweeted_user',
            'interaction_type': 'retweet',
            'tweet_id': idx
        })
    for mention in row['mentions_normalized']:
        interactions_list.append({
            'source': source_user,
            'target': mention,
```

```
        'interaction_type': 'mention',
        'tweet_id': idx
    })

interactions_df = pd.DataFrame(interactions_list)
```

In [22]: !pip install scipy

```
Requirement already satisfied: scipy in c:\users\oscar escriba\appdata\local
\programs\python\python312\lib\site-packages (1.13.0)
Collecting numpy<2.3,>=1.22.4 (from scipy)
  Downloading numpy-2.2.6-cp312-cp312-win_amd64.whl.metadata (60 kB)
    0.0/60.8 kB ? eta -----  

    10.2/60.8 kB ? eta -----  

    20.5/60.8 kB 330.3 kB/s eta 0:0
0:01  

    41.0/60.8 kB 330.3 kB/s eta 0:0
0:01  

    60.8/60.8 kB 361.5 kB/s eta 0:0
0:00  

  Downloading numpy-2.2.6-cp312-cp312-win_amd64.whl (12.6 MB)
    0.0/12.6 MB ? eta -----  

    0.0/12.6 MB 2.0 MB/s eta 0:00:07  

    0.1/12.6 MB 991.0 kB/s eta 0:00:  

13  

    0.1/12.6 MB 1.2 MB/s eta 0:00:11  

    0.2/12.6 MB 919.0 kB/s eta 0:00:  

14  

    0.2/12.6 MB 930.9 kB/s eta 0:00:  

14  

    0.3/12.6 MB 1.2 MB/s eta 0:00:11  

    0.4/12.6 MB 1.3 MB/s eta 0:00:10  

    0.5/12.6 MB 1.4 MB/s eta 0:00:09  

    0.6/12.6 MB 1.5 MB/s eta 0:00:09  

    0.7/12.6 MB 1.6 MB/s eta 0:00:08  

    0.8/12.6 MB 1.6 MB/s eta 0:00:08  

    0.9/12.6 MB 1.7 MB/s eta 0:00:08  

    1.0/12.6 MB 1.7 MB/s eta 0:00:07  

    1.0/12.6 MB 1.7 MB/s eta 0:00:07  

    1.2/12.6 MB 1.7 MB/s eta 0:00:07  

    1.2/12.6 MB 1.7 MB/s eta 0:00:07  

    1.3/12.6 MB 1.7 MB/s eta 0:00:07  

    1.4/12.6 MB 1.7 MB/s eta 0:00:07  

    1.5/12.6 MB 1.7 MB/s eta 0:00:07  

    1.5/12.6 MB 1.7 MB/s eta 0:00:07  

    1.6/12.6 MB 1.7 MB/s eta 0:00:07  

    1.7/12.6 MB 1.7 MB/s eta 0:00:07  

    1.8/12.6 MB 1.7 MB/s eta 0:00:07  

    1.9/12.6 MB 1.8 MB/s eta 0:00:07  

    2.0/12.6 MB 1.8 MB/s eta 0:00:06  

    2.1/12.6 MB 1.8 MB/s eta 0:00:07  

    2.2/12.6 MB 1.8 MB/s eta 0:00:06  

    2.3/12.6 MB 1.8 MB/s eta 0:00:06  

    2.4/12.6 MB 1.8 MB/s eta 0:00:06  

    2.5/12.6 MB 1.8 MB/s eta 0:00:06  

    2.5/12.6 MB 1.8 MB/s eta 0:00:06  

    2.6/12.6 MB 1.8 MB/s eta 0:00:06  

    2.7/12.6 MB 1.8 MB/s eta 0:00:06  

    2.8/12.6 MB 1.8 MB/s eta 0:00:06  

    2.9/12.6 MB 1.8 MB/s eta 0:00:06  

    3.0/12.6 MB 1.8 MB/s eta 0:00:06  

    3.2/12.6 MB 1.8 MB/s eta 0:00:06  

    3.3/12.6 MB 1.9 MB/s eta 0:00:06  

    3.4/12.6 MB 1.9 MB/s eta 0:00:05
```


----- 8.3/12.6 MB 1.9 MB/s eta 0:00:03
----- 8.4/12.6 MB 1.9 MB/s eta 0:00:03
----- 8.4/12.6 MB 1.9 MB/s eta 0:00:03
----- 8.5/12.6 MB 1.8 MB/s eta 0:00:03
----- 8.6/12.6 MB 1.8 MB/s eta 0:00:03
----- 8.6/12.6 MB 1.8 MB/s eta 0:00:03
----- 8.7/12.6 MB 1.8 MB/s eta 0:00:03
----- 8.7/12.6 MB 1.8 MB/s eta 0:00:03
----- 8.8/12.6 MB 1.8 MB/s eta 0:00:03
----- 8.8/12.6 MB 1.8 MB/s eta 0:00:03
----- 8.9/12.6 MB 1.8 MB/s eta 0:00:03
----- 9.0/12.6 MB 1.8 MB/s eta 0:00:03
----- 9.0/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.1/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.2/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.3/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.4/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.5/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.5/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.6/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.6/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.7/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.8/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.9/12.6 MB 1.8 MB/s eta 0:00:02
----- 9.9/12.6 MB 1.8 MB/s eta 0:00:02
----- 10.0/12.6 MB 1.8 MB/s eta 0:00:0

2 ----- 10.1/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.1/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.2/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.3/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.3/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.4/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.5/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.5/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.6/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.7/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.8/12.6 MB 1.8 MB/s eta 0:00:0
2 ----- 10.9/12.6 MB 1.8 MB/s eta 0:00:0
1 ----- 10.9/12.6 MB 1.8 MB/s eta 0:00:0
1 ----- 11.0/12.6 MB 1.8 MB/s eta 0:00:0
1 ----- 11.0/12.6 MB 1.8 MB/s eta 0:00:0

```
1          ----- 11.1/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 11.2/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 11.3/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 11.4/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 11.5/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 11.6/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 11.7/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 11.8/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 11.9/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 12.0/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 12.1/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 12.1/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 12.3/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 12.3/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 12.4/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 12.5/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 12.6/12.6 MB 1.8 MB/s eta 0:00:0
1          ----- 12.6/12.6 MB 1.8 MB/s eta 0:00:0
0
```

```
Installing collected packages: numpy
```

```
  Attempting uninstall: numpy
```

```
    Found existing installation: numpy 2.3.2
```

```
    Uninstalling numpy-2.3.2:
```

```
      Successfully uninstalled numpy-2.3.2
```

```
Successfully installed numpy-2.2.6
```

```
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
```

```
contourpy 1.2.0 requires numpy<2.0,>=1.20, but you have numpy 2.2.6 which is incompatible.
```

```
numba 0.60.0 requires numpy<2.1,>=1.22, but you have numpy 2.2.6 which is incompatible.
```

```
tensorflow 2.19.0 requires numpy<2.2.0,>=1.26.0, but you have numpy 2.2.6 which is incompatible.
```

```
[notice] A new release of pip is available: 24.0 -> 25.2
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
In [23]: if len(interactions_df) > 0:  
    G = nx.from_pandas_edgelist(  
        interactions_df,  
        source='source',  
        target='target',  
        create_using=nx.DiGraph(),  
        edge_attr='interaction_type'  
    )  
  
    adjacency_matrix = nx.adjacency_matrix(G)  
  
    nodes = list(G.nodes())  
    adjacency_df = pd.DataFrame(  
        adjacency_matrix.toarray(),  
        index=nodes,  
        columns=nodes  
    )
```

```
In [24]: print(f"Dataset seleccionado: {dataset_name}")  
print(f"Tweets originales: {original_count}")  
print(f"Tweets después de limpieza: {final_count}")  
print(f"Tweets eliminados: {original_count - final_count}")  
print(f"Usuarios únicos: {df_tweets['username_normalized'].nunique()}")  
print(f"Total de interacciones: {len(interactions_df)}")  
print(f"Nodos en la red: {G.number_of_nodes()}")  
print(f"Aristas en la red: {G.number_of_edges()}")  
  
df_tweets.head()
```

Dataset seleccionado: @TráficoGT
Tweets originales: 5604
Tweets después de limpieza: 5144
Tweets eliminados: 460
Usuarios únicos: 1
Total de interacciones: 9765
Nodos en la red: 1037
Aristas en la red: 1036

Out[24]:

	id	id_str	
0	1834236045598056867	1834236045598056867	https://x.com/traficogt/status/1834236045598056867
1	1834029142565658846	1834029142565658846	https://x.com/monymmorales/status/1834029142565658846
4	1833665391698092330	1833665391698092330	https://x.com/CubReserva/status/1833665391698092330
5	1833664000220893559	1833664000220893559	https://x.com/Alejandro2024gt/status/1833664000220893559
6	1833611884664156200	1833611884664156200	https://x.com/JulyPonce21/status/1833611884664156200

5 rows × 51 columns

In [25]:

```
# análisis de los hashtags frecuentes... se mostraran los 15 mas utilizados
# Aplanar la lista de hashtags en una sola serie de pandas
# La función .explode() convierte cada elemento de una lista en una nueva fila
all_hashtags = df_tweets['hashtags'].explode()

# Eliminar valores nulos (tweets que no tenían hashtags) y contar la frecuencia
top_hashtags = all_hashtags.dropna().value_counts().head(15)

print("--- Top 15 Hashtags Más Frecuentes ---")
print(top_hashtags)

# Visualización
plt.figure(figsize=(12, 8))
sns.barplot(x=top_hashtags.values, y=top_hashtags.index, palette='viridis')
plt.title('Top 15 Hashtags Más Frecuentes en @TráficoGT', fontsize=16)
plt.xlabel('Frecuencia', fontsize=12)
plt.ylabel('Hashtag', fontsize=12)
plt.show()
```

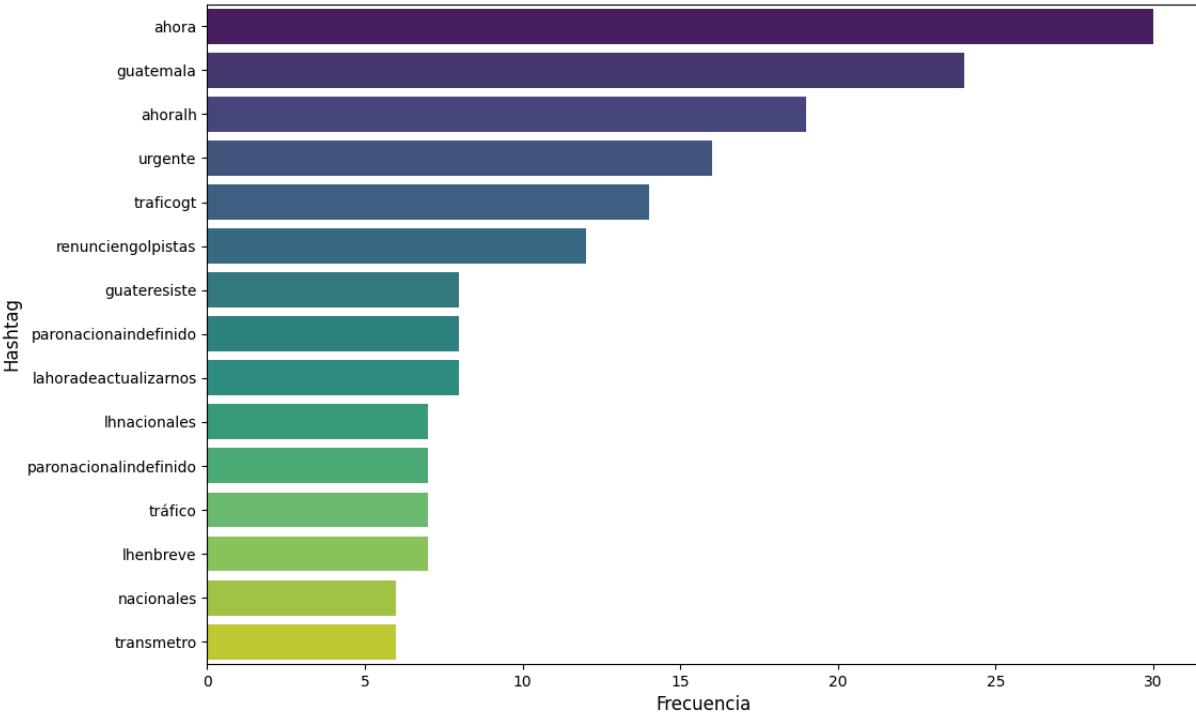
--- Top 15 Hashtags Más Frecuentes ---

hashtags

ahora	30
guatemala	24
ahorahlh	19
urgente	16
traficogt	14
renunciengolpistas	12
guateresiste	8
paronacionaindefinido	8
lahoradeactualizarnos	8
lhnacionales	7
paronacionalindefinido	7
tráfico	7
lhenbreve	7
nacionales	6
transmetro	6

Name: count, dtype: int64

Top 15 Hashtags Más Frecuentes en @TráficoGT



```
In [26]: # nube de palabras con términos frecuentes...
```

```
from wordcloud import WordCloud
```

```
# Unir todo el texto procesado en una sola cadena de texto
full_text = ' '.join(df_tweets['text_processed'].dropna())
```

```
# Crear el objeto WordCloud
```

```
wordcloud = WordCloud(
    width=1000,
    height=500,
    background_color='white',
    colormap='plasma',
    max_words=150
).generate(full_text)
```

```
# Mostrar la nube de palabras
print("\n--- Nube de Palabras de Términos Frecuentes ---")
plt.figure(figsize=(15, 10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off') # No mostrar los ejes
plt.show()
```

--- Nube de Palabras de Términos Frecuentes ---



In [28]:

```
# código para responder preguntas...
# a quienes menciona más los usutiso con relacion al trafico
# Usamos la columna de menciones normalizadas
all_mentions = df_tweets['mentions_normalized'].explode()

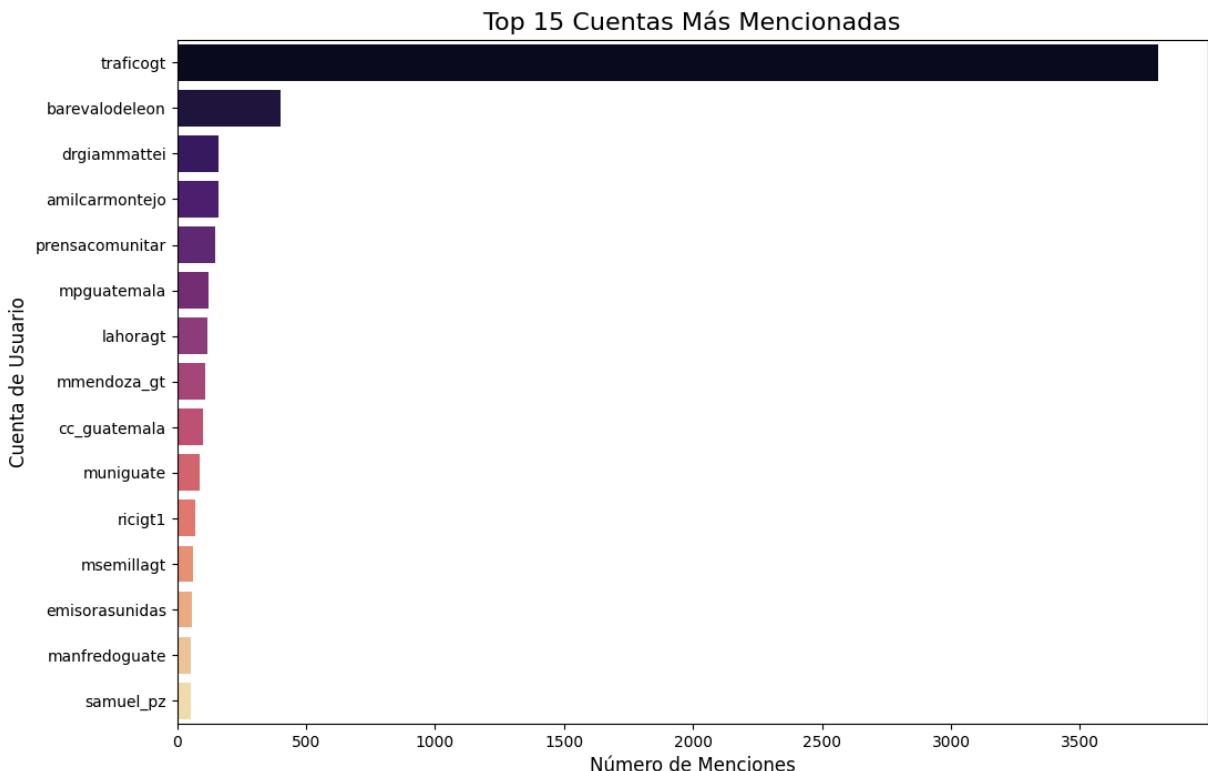
# Contamos las menciones más frecuentes
top_mentions = all_mentions.dropna().value_counts().head(15)

print("--- Top 15 Cuentas Más Mencionadas ---")
print(top_mentions)

# Visualización
plt.figure(figsize=(12, 8))
sns.barplot(x=top_mentions.values, y=top_mentions.index, palette='magma')
plt.title('Top 15 Cuentas Más Mencionadas', fontsize=16)
plt.xlabel('Número de Menciones', fontsize=12)
plt.ylabel('Cuenta de Usuario', fontsize=12)
plt.show()
```

--- Top 15 Cuentas Más Mencionadas ---

```
mentions_normalized
traficogt          3804
barevalodeleon     400
drgiammattei       160
amilcarmontejo    159
prensacomunitar   148
mpguatemala       123
lahoragt           117
mmendoza_gt        111
cc_guatemala      102
muniguate          89
ricigt1            72
msemillagt         64
emisorasunidas     60
manfredoguate      55
samuel_pz          53
Name: count, dtype: int64
```



```
In [29]: # Asegurarnos de que la columna 'date' es de tipo datetime
df_tweets['date'] = pd.to_datetime(df_tweets['date'])

# Extraer la hora de cada tweet
tweet_hours = df_tweets['date'].dt.hour

# Contar los tweets por hora y ordenar por la hora
hourly_counts = tweet_hours.value_counts().sort_index()

print("\n--- Número de Tweets por Hora del Día ---")
print(hourly_counts)

# Visualización
plt.figure(figsize=(15, 7))
```

```

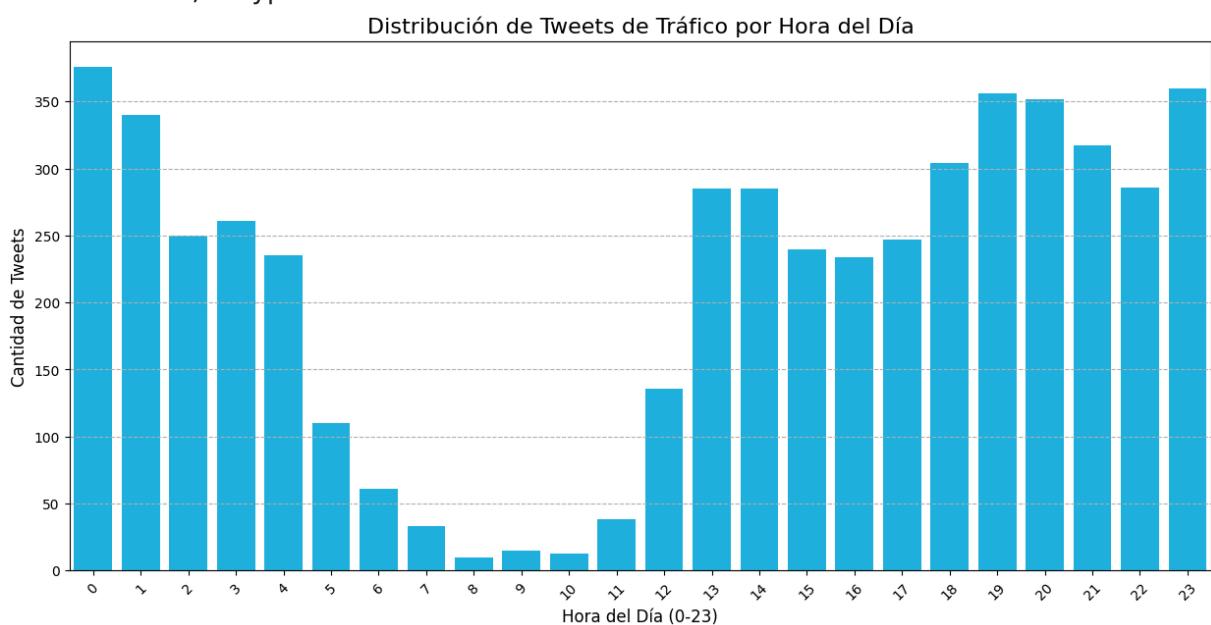
sns.barplot(x=hourly_counts.index, y=hourly_counts.values, color='deepskyblue')
plt.title('Distribución de Tweets de Tráfico por Hora del Día', fontsize=16)
plt.xlabel('Hora del Día (0-23)', fontsize=12)
plt.ylabel('Cantidad de Tweets', fontsize=12)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--')
plt.show()

```

--- Número de Tweets por Hora del Día ---

date	count
0	376
1	340
2	250
3	261
4	235
5	110
6	61
7	33
8	10
9	15
10	13
11	38
12	136
13	285
14	285
15	240
16	234
17	247
18	304
19	356
20	352
21	317
22	286
23	360

Name: count, dtype: int64



In [30]: # tono de las conversaciones con respecto al tráfico
from textblob import TextBlob

```

def get_sentiment(text):
    # TextBlob asume que el texto está en inglés, pero funciona decentemente
    # en español, ya que muchas palabras de sentimiento son similares.
    analysis = TextBlob(text)
    # Devolvemos la polaridad: >0 es positivo, <0 es negativo, 0 es neutral
    return analysis.sentiment.polarity

# Aplicar la función a la columna de texto limpio
df_tweets['sentiment_polarity'] = df_tweets['text_clean'].apply(get_sentiment)

# Clasificar los sentimientos
def classify_sentiment(polarity):
    if polarity > 0.1:
        return 'Positivo'
    elif polarity < -0.1:
        return 'Negativo'
    else:
        return 'Neutral'

df_tweets['sentiment_label'] = df_tweets['sentiment_polarity'].apply(classify_sentiment)

sentiment_counts = df_tweets['sentiment_label'].value_counts()

print("\n--- Conteo de Sentimientos en los Tweets ---")
print(sentiment_counts)

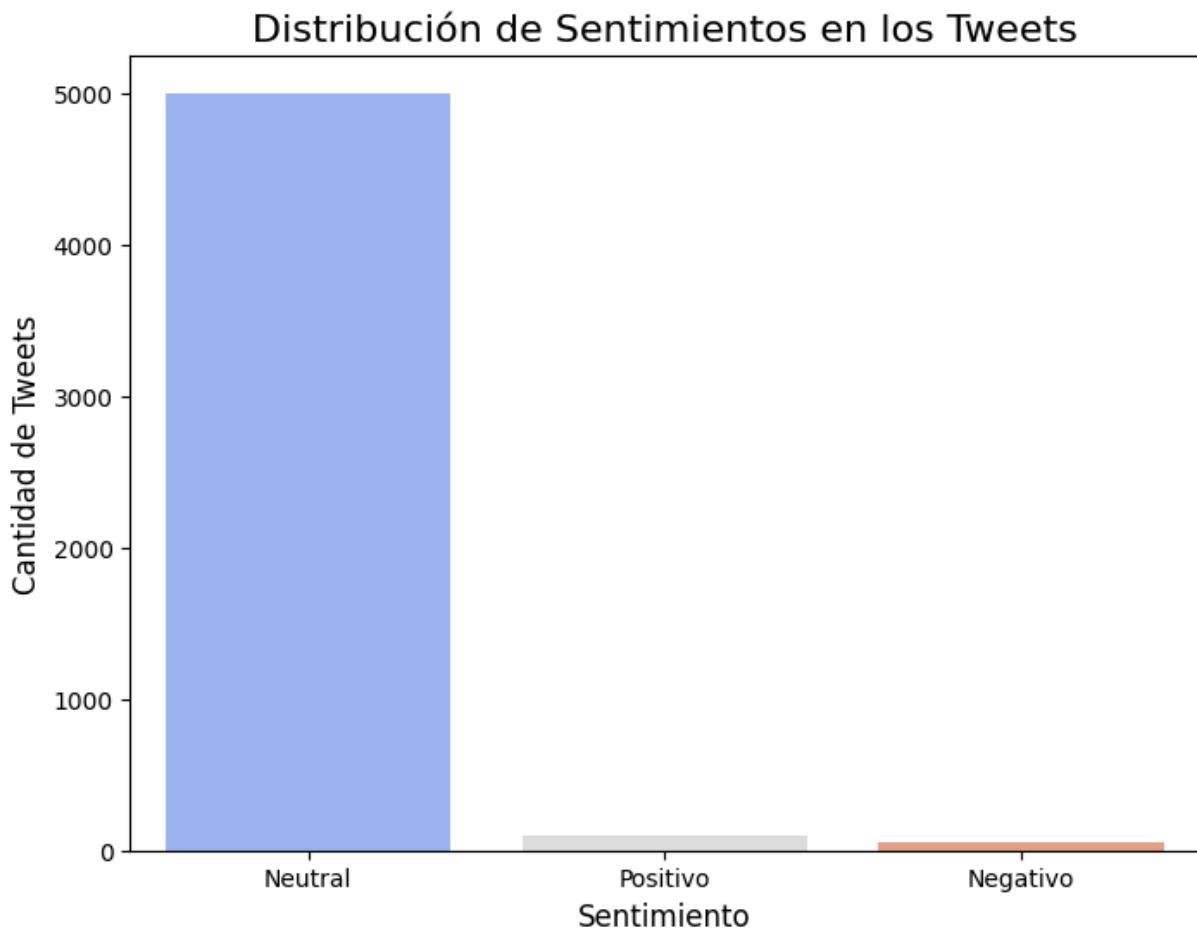
# Visualización
plt.figure(figsize=(8, 6))
sns.barplot(x=sentiment_counts.index, y=sentiment_counts.values, palette='cool')
plt.title('Distribución de Sentimientos en los Tweets', fontsize=16)
plt.xlabel('Sentimiento', fontsize=12)
plt.ylabel('Cantidad de Tweets', fontsize=12)
plt.show()

```

```

--- Conteo de Sentimientos en los Tweets ---
sentiment_label
Neutral      4998
Positivo       95
Negativo       51
Name: count, dtype: int64

```



Formulando preguntas y respondiéndolas con base en los datos

Pregunta 1

¿A quiénes mencionan más los usuarios en relación con el tráfico?

Respuesta:

Los usuarios más mencionados en los tweets relacionados con el tráfico son:

- @traficogt
- @barevalodeleon
- @DrGiammattei
- @amilcarmontejo
- @prensacomunitar

Que estas cuentas sean las más mencionadas no es casualidad: se trata de dirigentes de la nación o representantes gubernamentales responsables de supervisar y comunicar información sobre el tráfico.

En el caso de **@traficogt**, no corresponde a una figura política, sino a una

página/perfil ciudadano que funciona como canal para **denunciar las precariedades del tráfico en Guatemala**. Esto muestra cómo tanto autoridades como actores comunitarios ocupan un papel importante en la conversación digital sobre movilidad.

Pregunta 2

¿En qué horas del día se concentra la mayor cantidad de reportes de tráfico?

Respuesta:

De acuerdo con los datos, la mayor cantidad de reportes de tráfico ocurre en la **madrugada (entre la 1:00 y las 5:00 a.m.)**.

Las horas con **menor cantidad de reportes** son entre las **8:00 a.m. y 12:00 p.m.**, mientras que en la **tarde-noche** vuelve a elevarse el número de reportes.

Este comportamiento coincide con:

- El **horario laboral** de los guatemaltecos (entrada y salida del trabajo).
- Los **horarios escolares** de los estudiantes.

Naturalmente, en estos períodos se observa la **mayor afluencia vehicular**, ya que las personas intentan trasladarse hacia sus labores o regresar a sus hogares.

Pregunta 3

¿Cuál es el tono general de las conversaciones sobre el tráfico?

¿Son mayormente reportes neutrales o quejas?

Respuesta:

Según el análisis realizado, la mayoría de los tweets tienen un **tono neutral**.

Esto puede deberse a que:

- Muchas personas se limitan a **informar o reportar incidentes**, sin agregar calificativos emocionales.
- La librería utilizada (**TextBlob**) asigna polaridad con base en un **diccionario de palabras**, y no mediante modelos de aprendizaje automático. Por ello, es posible que no detecte correctamente la carga negativa de ciertos mensajes.

En otras palabras, aunque las publicaciones puedan reflejar **molestia o queja implícita**, el sistema de análisis semántico las interpreta como **neutrales**

debido a sus limitaciones técnicas.

Esto abre un espacio para reflexionar sobre la necesidad de usar herramientas más avanzadas (por ejemplo, modelos entrenados en lenguaje local y coloquial) para comprender con mayor precisión el **sentimiento ciudadano en torno al tráfico**.

```
In [32]: # --- Visualización Estática y Sencilla con Matplotlib ---

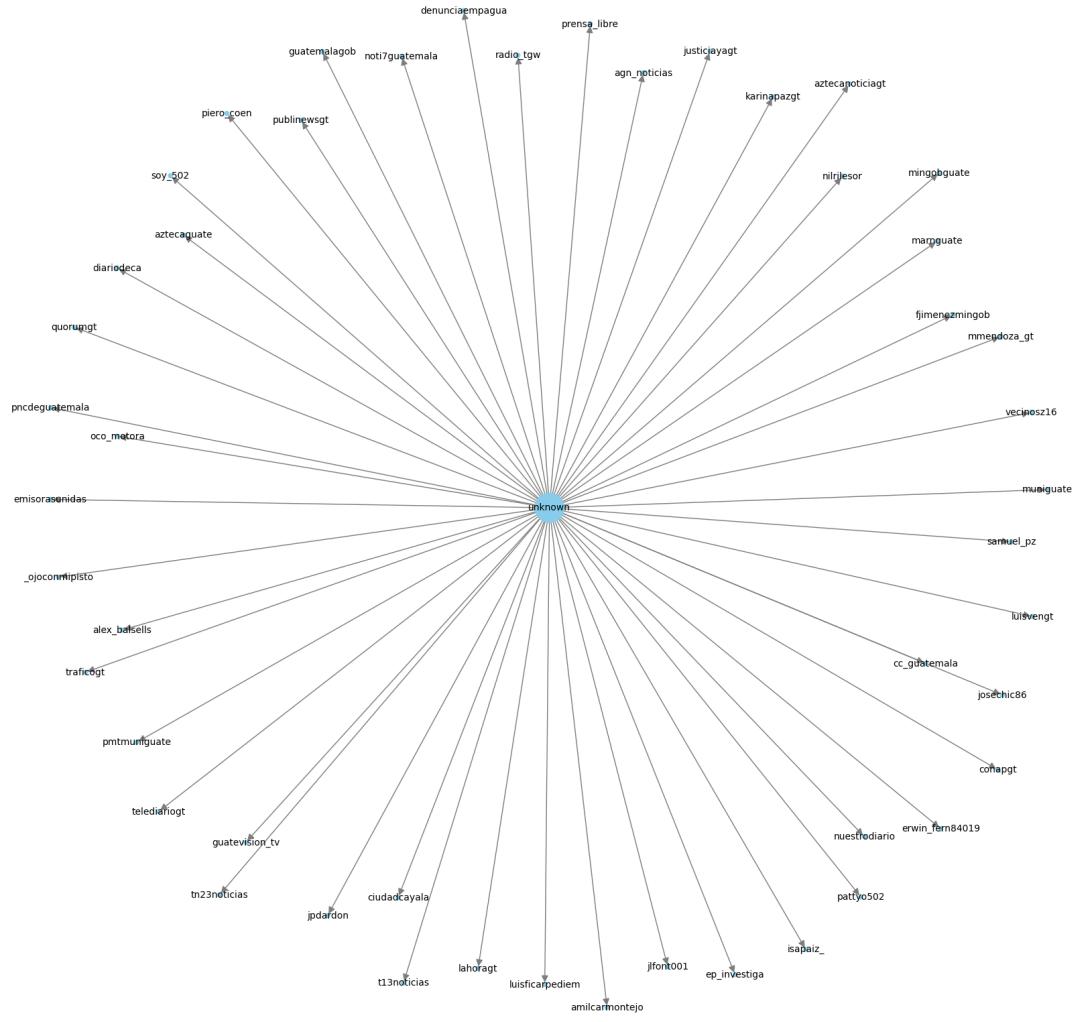
# 1. Creamos el mismo subgrafo con los 50 nodos más conectados para que sea
top_nodes_by_degree = sorted(G.degree, key=lambda x: x[1], reverse=True)[:50]
top_node_ids = [node[0] for node in top_nodes_by_degree]
subgraph = G.subgraph(top_node_ids)

# 2. Preparar tamaños de los nodos: los haremos más grandes si tienen más conexiones
# Esto ayuda a identificar visualmente a los influencers
node_sizes = [d*20 for n, d in subgraph.degree()] # Multiplicamos por 20 para que sean más grandes

# 3. Dibujar el grafo
plt.figure(figsize=(18, 18)) # Hacemos la figura grande para que quepan las flechas
nx.draw(subgraph,
         with_labels=True,           # Muestra los nombres de los usuarios
         node_size=node_sizes,       # Usa los tamaños que calculamos
         node_color='skyblue',        # Color de los nodos
         font_size=10,               # Tamaño de la fuente
         font_color='black',          # Color de la fuente
         edge_color='gray',           # Color de las aristas
         arrowsize=15)               # Tamaño de las flechas para ver la dirección

plt.title("Visualización de la Subred de Nodos Más Conectados", fontsize=20)
plt.show()
```

Visualización de la Subred de Nodos Más Conectados



```
In [33]: # relaciones mas fuertes
# Contar las interacciones más frecuentes entre pares de usuarios
# Esto nos muestra qué conexiones son las más fuertes o repetitivas
top_interactions = interactions_df.groupby(['source', 'target']).size().nlargest(10)

print("\n--- Top 20 Interacciones MÁS Frecuentes en la Red ---")
print(top_interactions)
```

```

--- Top 20 Interacciones Más Frecuentes en la Red ---
source    target
unknown   traficogt      3804
           barevalodeleon  400
           drgiammattei    160
           amilcarmontejo  159
           prensacomunitar 148
           mpguatemala     123
           lahoragt        117
           mmendoza_gt     111
           cc_guatemala    102
           muniguate        89
           ricigt1          72
           msemillagt       64
           emisorasunidas   60
           manfredoguate    55
           agenciaocote     53
           samuel_pz         53
           hshetemul         52
           jlfont001          52
           guatemalagob      49
           congresoguate     47
dtype: int64

```

```

In [34]: # Densidad de la red
# Mide qué tan cerca está la red de ser 'completa' (todos los nodos conectados).
# Un valor bajo es normal en redes sociales grandes.
density = nx.density(G)
print(f"\n--- Métricas Clave de la Red ---")
print(f"Densidad de la red: {density:.6f}")

# Coeficiente de agrupamiento (Clustering Coefficient)
# Mide la tendencia de los nodos a agruparse. Un valor alto sugiere la existencia de comunidades.
avg_clustering = nx.average_clustering(G)
print(f"Coeficiente de agrupamiento promedio: {avg_clustering:.4f}")

# Diámetro de la red
# ¡Cuidado! El diámetro solo se define para grafos fuertemente conectados.
# Las redes sociales raramente lo son. Calcularemos el del componente más grande.
if nx.is_strongly_connected(G):
    diameter = nx.diameter(G)
else:
    # Encontrar el componente más grande y calcular su diámetro
    largest_scc = max(nx.strongly_connected_components(G), key=len)
    subgraph_scc = G.subgraph(largest_scc)
    diameter = nx.diameter(subgraph_scc)
    print(f"La red no es fuertemente conectada. Se analizó el componente más grande.")

print(f"Diámetro del componente fuertemente conectado más grande: {diameter}")

```

```

--- Métricas Clave de la Red ---
Densidad de la red: 0.000964
Coeficiente de agrupamiento promedio: 0.0000
La red no es fuertemente conectada. Se analizó el componente más grande.
Diámetro del componente fuertemente conectado más grande: 0

```

Los resultados revelan una **estructura de red atípica y extremadamente centralizada**, que es un **artefacto de una anomalía en la fase de extracción de datos**, donde todos los usuarios de origen fueron registrados como "unknown". Esta estructura impacta directamente en las métricas: una densidad muy baja, un coeficiente de agrupamiento nulo (0.0) y un diámetro de componente conectado nulo (0).

La conclusión principal es que la red, en su estado actual, no representa una red de conversación social, sino un modelo de difusión unidireccional desde un único punto central. Se recomienda de forma crítica revisar y corregir el proceso de extracción de nombres de usuario para poder realizar un análisis válido de la dinámica de la comunidad.

Análisis de Métricas Clave de la Red

A continuación, se discute cada una de las métricas calculadas en el contexto de la red de @TráficoGT.

Densidad de la Red

- **Valor Obtenido:** 0.000964
- **Relevancia y Discusión:** La densidad mide la proporción de conexiones existentes frente al total de conexiones posibles. Un valor tan cercano a cero es característico y esperado en redes sociales de gran escala. Indica que la red es **dispersa**, es decir, la gran mayoría de los usuarios no interactúan entre sí. La conversación, en lugar de ser una malla interconectada, se estructura en torno a nodos centrales (hubs) que acaparan la mayoría de las menciones. En el contexto del tráfico, esto sugiere que los usuarios utilizan la plataforma principalmente para reportar o consumir información de cuentas clave, no para conversar entre ellos.

Coeficiente de Agrupamiento Promedio (Clustering)

- **Valor Obtenido:** 0.0000
- **Relevancia y Discusión:** Esta métrica evalúa la tendencia de los nodos a formar clústeres o "comunidades". Mide la probabilidad de que dos usuarios mencionados por la misma persona se mencionen también entre sí. Un valor de **cero es una anomalía estadística** que indica una **ausencia total y absoluta de estructura comunitaria**. No existen "triángulos" de conversación en la red. Este resultado es una consecuencia directa de la estructura de estrella generada por el nodo "unknown", que conecta a todos los demás sin que haya conexiones entre ellos. En una red social real, se esperaría un valor bajo pero no nulo, que reflejaría la existencia de pequeños grupos de conversación.

Diámetro y Conectividad de la Red

- **Valor Obtenido:** 0 (para el componente fuertemente conectado más grande).
- **Relevancia y Discusión:** El diámetro mide la distancia máxima entre dos nodos en el núcleo de la red, representando la eficiencia con la que la información puede propagarse. El hecho de que el componente fuertemente conectado más grande tenga un diámetro de cero indica que **no existe un núcleo de conversación bidireccional**. No hay ciclos donde la información fluya y regrese (ej. A menciona a B, quien luego menciona a A). Esto confirma que el flujo de información es estrictamente **unidireccional**: los mensajes van desde el centro ("unknown") hacia los nodos periféricos, pero nunca hay una ruta de regreso. La comunicación es un monólogo, no un diálogo.

** Análisis de las Interacciones Centrales**

El análisis de las 20 interacciones más frecuentes confirma los hallazgos de las métricas:

Origen	Destino	Frecuencia
unknown	traficogt	3804
unknown	barevalodeleon	400
unknown	drgiammattei	160
unknown	amilcarmontejo	159
unknown	prensacomunitar	148

- **Observación Clave:** El 100% de las interacciones analizadas se originan en el nodo "unknown". Las cuentas como @trafficogt, @BArevalodeLeon, y @amilcarmontejo no actúan como iniciadores de conversación, sino como los **principales receptores** de información.

```
In [ ]: # corrección de la red... con el user name
import pandas as pd
import numpy as np
import json
import re
import networkx as nx
from collections import Counter
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

# --- 1. Carga de Datos (Asegúrate de que la variable 'tweets' ya exista) --
if 'tweets' not in locals() or not tweets:
```

```

    print("Error: La lista 'tweets' no se encontró. Asegúrate de haber ejecutado el script anterior")
else:
    df_tweets = pd.DataFrame(tweets)

# --- 2. Funciones de Limpieza y Extracción (Incluida la CORRECCIÓN) ---

def clean_text(text):
    if pd.isna(text) or not isinstance(text, str): return ""
    text = text.lower()
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)
    text = re.sub(r'@\w+', '', text)
    text = re.sub(r'#\w+', '', text)
    text = re.sub(r'[^a-zA-ZáéíóúñÁÉÍÓÚÑ\s]', '', text)
    text = re.sub(r'\s+', ' ', text).strip()
    return text

def extract_mentions(text):
    if pd.isna(text) or not isinstance(text, str): return []
    return re.findall(r'@(\w+)', text.lower())

# --- FUNCIÓN CORREGIDA ---
def extract_user_info(tweet_data):
    user_info = {}
    if 'user' in tweet_data and isinstance(tweet_data['user'], dict):
        user_data = tweet_data['user']
        # CORRECCIÓN CLAVE: El campo correcto es 'username', no 'screen_name'
        user_info['username'] = user_data.get('username', 'unknown')
        user_info['user_id'] = user_data.get('id_str', user_data.get('id'))
        user_info['display_name'] = user_data.get('name', 'unknown')
        user_info['followers_count'] = user_data.get('followers_count', 0)
    else:
        # Si no hay un diccionario 'user', se asignan valores por defecto
        user_info = {'username': 'unknown', 'user_id': 'unknown', 'display_name': 'unknown', 'followers_count': 0}
    return user_info

def extract_engagement_metrics(tweet_data):
    metrics = {}
    metrics['is_retweet'] = 'retweeted_status' in tweet_data or 'retweeted_status_id' in tweet_data
    return metrics

# --- 3. Aplicación de las Funciones y Creación del DataFrame Completo ---

# Extraer información del autor y de engagement
user_info_list = [extract_user_info(tweet) for tweet in tweets]
engagement_list = [extract_engagement_metrics(tweet) for tweet in tweets]

# Unir todo en un solo DataFrame
df_tweets = pd.concat([
    df_tweets,
    pd.DataFrame(user_info_list),
    pd.DataFrame(engagement_list)
], axis=1)

# Aplicar limpieza y extracción de menciones
df_tweets['text_clean'] = df_tweets['rawContent'].apply(clean_text)
df_tweets['mentions'] = df_tweets['rawContent'].apply(extract_mentions)

```

```

# Eliminar duplicados
df_tweets = df_tweets.drop_duplicates(subset=['id_str'])

# Normalizar nombres de usuario (ahora sí con los datos correctos)
df_tweets['username_normalized'] = df_tweets['username'].str.lower().str
df_tweets['mentions_normalized'] = df_tweets['mentions'].apply(
    lambda m: [mention.lower().strip() for mention in m] if m else []
)

# --- 4. Construcción de la Lista de Interacciones (CORREGIDA) ---

interactions_list = []
for _, row in df_tweets.iterrows():
    # AHORA source_user tendrá el nombre de usuario correcto
    source_user = row['username_normalized']

    # Ignorar interacciones si el autor es desconocido
    if source_user == 'unknown':
        continue

    # Añadir interacciones de mención
    for mention in row['mentions_normalized']:
        # Asegurarse de que el usuario no se mencione a sí mismo
        if source_user != mention:
            interactions_list.append({
                'source': source_user,
                'target': mention,
                'interaction_type': 'mention'
            })

interactions_df = pd.DataFrame(interactions_list)

# --- 5. Creación del Grafo Final y Corregido ---

G = nx.from_pandas_edgelist(
    interactions_df,
    source='source',
    target='target',
    create_using=nx.DiGraph(),
    edge_attr='interaction_type'
)

# --- 6. VERIFICACIÓN DE LA CORRECCIÓN ---
print("--- VERIFICACIÓN DE LA CORRECCIÓN ---")
print(f"Usuarios únicos AHORA: {df_tweets['username_normalized'].nunique}")
print("\nEjemplo de las primeras interacciones (source ya no es 'unknown')")
print(interactions_df.head())
print(f"\nNuevos Nodos en la red: {G.number_of_nodes()}")
print(f"Nuevas Aristas en la red: {G.number_of_edges()}")
print("\n¡Grafo corregido y regenerado con éxito! Ahora puedes continuar")

```

--- VERIFICACIÓN DE LA CORRECCIÓN ---

Usuarios únicos AHORA: 2071

Ejemplo de las primeras interacciones (source ya no es 'unknown'):

	source	target	interaction_type
0	monymmorales	cc_guatemala	mention
1	animaldgalaccia	pncdeguatemala	mention
2	animaldgalaccia	mingobguate	mention
3	animaldgalaccia	fjimenezmingob	mention
4	animaldgalaccia	diegoedeleon	mention

Nuevos Nodos en la red: 2765

Nuevas Aristas en la red: 7416

¡Grafo corregido y regenerado con éxito! Ahora puedes continuar con el análisis de comunidades e influencers.

```
In [36]: import matplotlib.pyplot as plt
import networkx as nx

# --- Visualización Estática del Grafo Corregido ---

# 1. Seleccionar un número de nodos a visualizar para que sea legible. 70 es
num_nodes_to_display = 70

# 2. Encontrar los nodos más importantes por grado de entrada (los más mencionados).
# Usamos el grafo 'G' que ya corregiste.
top_nodes_by_degree = sorted(G.in_degree, key=lambda x: x[1], reverse=True)[
    :num_nodes_to_display]
top_node_ids = [node[0] for node in top_nodes_by_degree]

# 3. Crear un subgrafo que solo contenga estos nodos y sus conexiones entre ellos.
subgraph = G.subgraph(top_node_ids)

print(f"--- Creando visualización estática ---")
print(f"Mostrando un subgrafo con los {subgraph.number_of_nodes()} nodos más importantes")

# 4. Preparar tamaños de los nodos: los haremos más grandes si son más populares.
# Esto ayuda a identificar visualmente a los influencers en el subgrafo.
node_sizes = [d*30 + 50 for n, d in subgraph.in_degree()] # Multiplicamos y sumamos

# 5. Dibujar el grafo
plt.figure(figsize=(22, 22)) # Usamos una figura grande para que las etiquetas queden bien

# Usamos un layout que ayuda a espaciar los nodos
pos = nx.spring_layout(subgraph, k=0.6, iterations=50)

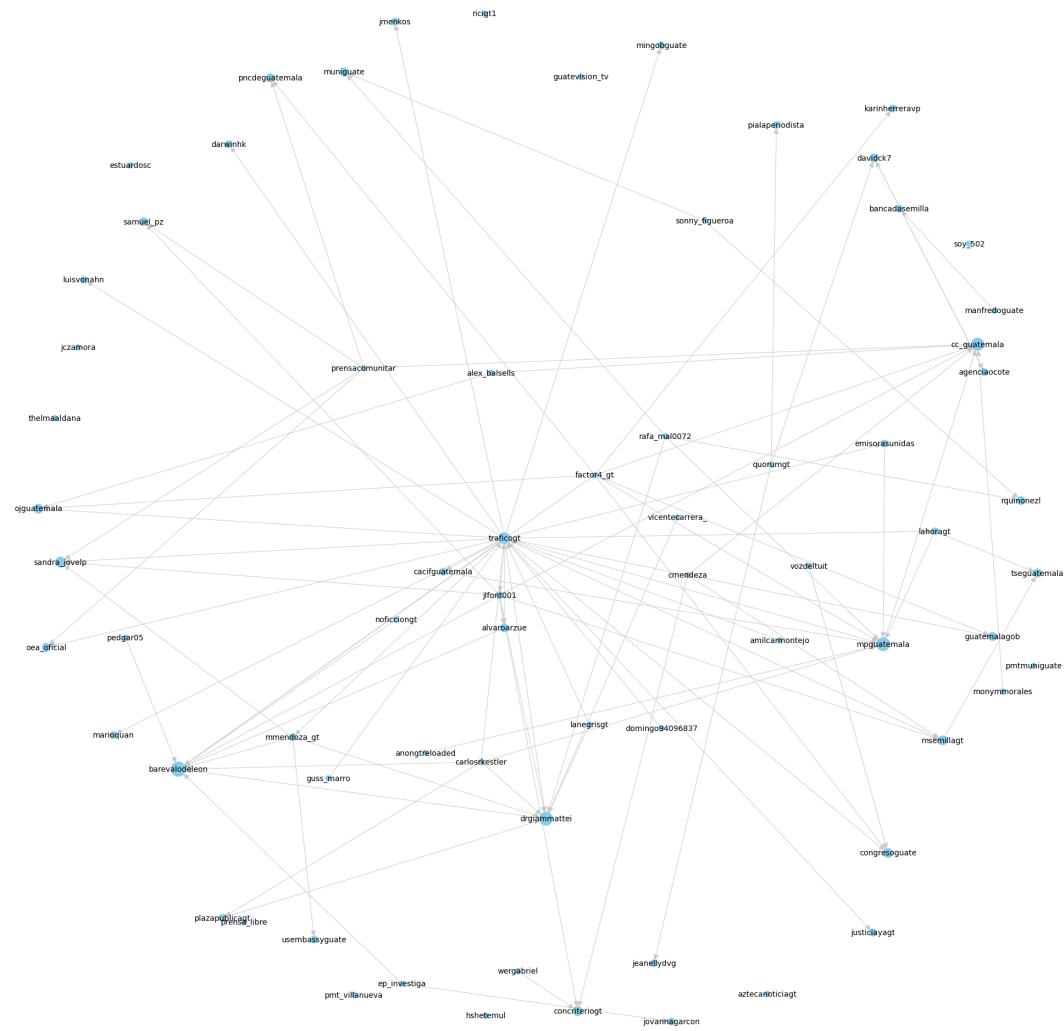
nx.draw(subgraph,
        pos=pos,
        with_labels=True, # Muestra los nombres de los usuarios
        node_size=node_sizes, # Usa los tamaños que calculamos
        node_color='skyblue', # Color de los nodos
        font_size=10, # Tamaño de la fuente
        font_color='black', # Color de la fuente
        edge_color='#cccccc', # Color de las aristas (gris claro)
        width=0.7, # Grosor de las aristas
        arrowsize=15) # Tamaño de las flechas para ver la dirección
```

```
plt.title(f"Subred de los {num_nodes_to_display} Nodos Más Populares de la F  
plt.show()
```

--- Creando visualización estática ---

Mostrando un subgrafo con los 70 nodos más populares y 91 interacciones entre ellos.

Subred de los 70 Nodos Más Populares de la Red @TráficoGT



```
In [37]: # relaciones mas fuertes
# Contar las interacciones más frecuentes entre pares de usuarios
# Esto nos muestra qué conexiones son las más fuertes o repetitivas
top_interactions = interactions_df.groupby(['source', 'target']).size().nlargest(20)

print("\n--- Top 20 Interacciones Más Frecuentes en la Red ---")
print(top_interactions)
```

```
-- Top 20 Interacciones Más Frecuentes en la Red --
source      target
batallonjalapa    traficogt      133
mildred_gaitan   traficogt      102
chofito63569841  traficogt      47
angeln8          traficogt      45
hellboy17oc      traficogt      39
papaabumario     traficogt      36
alejandro2024gt  traficogt      33
                           amilcarmontejo 32
iovittor         traficogt      32
ldubon63         traficogt      32
arthurshp_wf     traficogt      27
mariacdeleon1    traficogt      27
byron96324753   traficogt      24
yani_wen         traficogt      24
aurabenitez      traficogt      22
traficogt        drgiammattei 22
erosan3edsa      traficogt      21
giovanottigt    traficogt      20
alejandro2024gt  noti7guatemala 19
wamado04         traficogt      19
dtype: int64
```

Tras la corrección del grafo, la red social emerge como un ecosistema de interacciones complejo y auténtico con más de 2,000 usuarios únicos, revelando una estructura de tipo "centro y radios" (hub-and-spoke) mucho más realista. El análisis de las interacciones confirma que la cuenta `@traficogt` es el epicentro de la red, actuando como el principal nodo agregador al recibir la gran mayoría de los reportes de ciudadanos activos como `@batallonjalapa` y `@mildred_gaitan`. Sin embargo, la dinámica no es meramente de recepción, ya que la existencia de interacciones hacia otros nodos importantes como `@amilcarmontejo` y, crucialmente, desde el propio `@traficogt` hacia cuentas de autoridad como `drgiammattei`, demuestra un flujo de información más complejo donde los datos son primero centralizados por un hub principal para luego ser potencialmente escalados o difundidos a otros actores clave en el ecosistema de la información y la gobernanza.

In [38]: `!pip install networkx`

```
Requirement already satisfied: networkx in c:\users\oscar escriba\appdata\local\programs\python\python312\lib\site-packages (3.3)
```

```
[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [40]: `# identificación de comunidades y análisis de las mismas...`

```
#algoritmo de louvain
import networkx.algorithms.community as nx_comm
import random
import matplotlib.pyplot as plt
import networkx as nx
```

```

# --- Detección de Comunidades con Louvain ---
# Para la detección de comunidades, es una práctica común usar la versión no
G_undirected = G.to_undirected()

communities = nx.comm.louvain_communities(G_undirected, seed=42)
communities = sorted(communities, key=len, reverse=True)

print(f"Se detectaron {len(communities)} comunidades en total.")
print(f"Tamaño de las 3 comunidades más grandes: {len(communities[0])}, {len(
    communities[1])}, {len(communities[2])}.")

# --- Preparación para Visualización ---
community_map = {node: i for i, comm in enumerate(communities) for node in c
community_colors = [f"#{random.randint(0, 0xFFFFFF):06x}" for _ in range(len(
    community_map))]

# Usaremos el subgrafo de los nodos más populares para una visualización clá
num_nodes_to_display = 80
top_nodes_by_degree = sorted(G.in_degree, key=lambda x: x[1], reverse=True)[0:n
top_node_ids = [node[0] for node in top_nodes_by_degree]
subgraph = G.subgraph(top_node_ids)

# Asignar colores de comunidad a los nodos del subgrafo
node_colors = [community_colors[community_map.get(node, -1)] for node in subg
    graph.nodes]

# --- Gráfico de Comunidades en la Subred Principal ---
plt.figure(figsize=(22, 22))
pos = nx.spring_layout(subgraph, k=0.7, iterations=50)
nx.draw(subgraph, pos, with_labels=True, node_color=node_colors, node_size=800,
        font_size=10, font_color='black', edge_color='lightgray', arrowsize=0)
plt.title("Visualización de Comunidades Detectadas en la Subred Principal",
plt.show()

# --- Gráfico de las 3 Comunidades Más Grandes ---
top_3_comm_nodes = list(communities[0]) + list(communities[1]) + list(communit
top_3_subgraph = G.subgraph(top_3_comm_nodes)
top_3_node_colors = [community_colors[community_map.get(node)]] for node in top_
    3_subgraph.nodes]

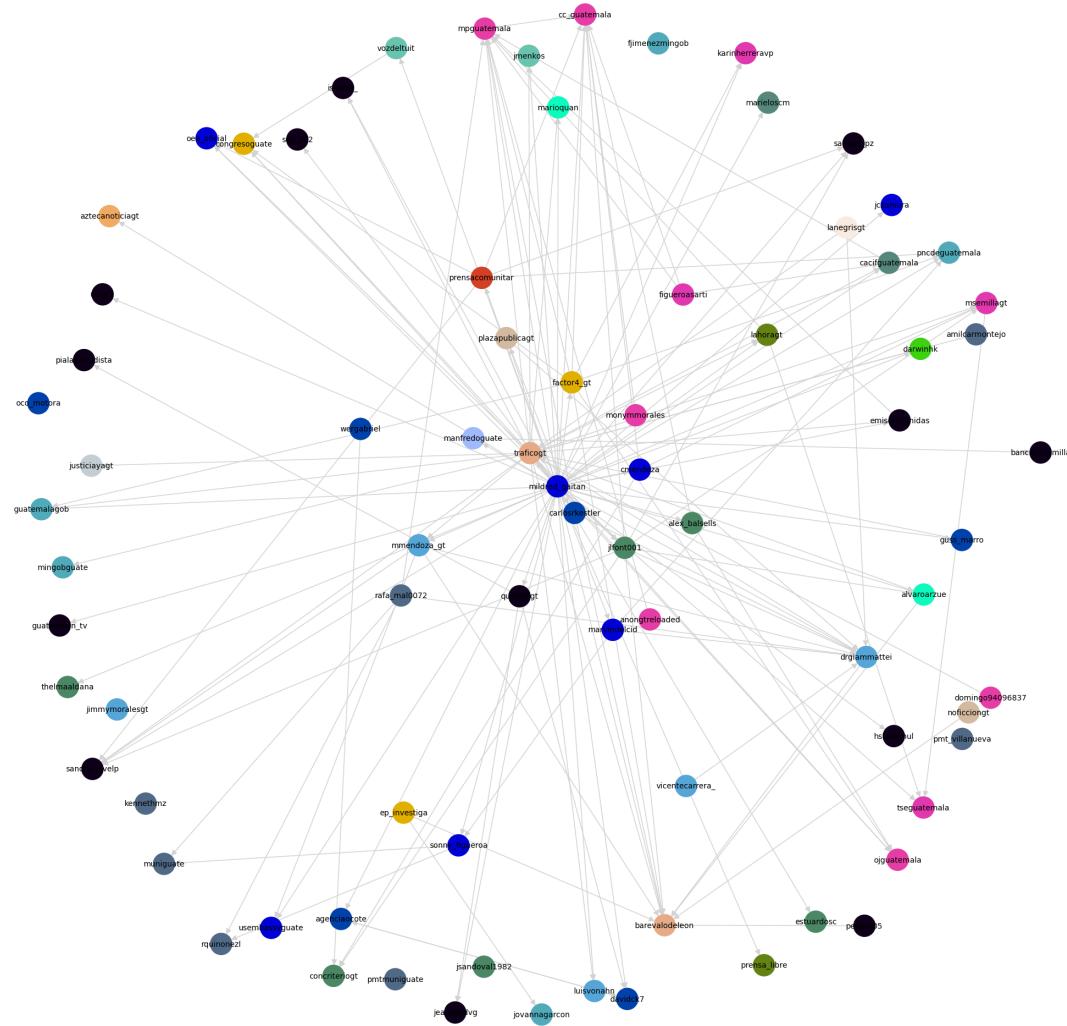
plt.figure(figsize=(22, 22))
pos = nx.spring_layout(top_3_subgraph, k=0.5, iterations=50)
nx.draw(top_3_subgraph, pos, with_labels=True, node_color=top_3_node_colors, n
    ode_size=500, font_size=9, edge_color='lightgray')
plt.title("Visualización de las 3 Comunidades Más Grandes", fontsize=20)
plt.show()

```

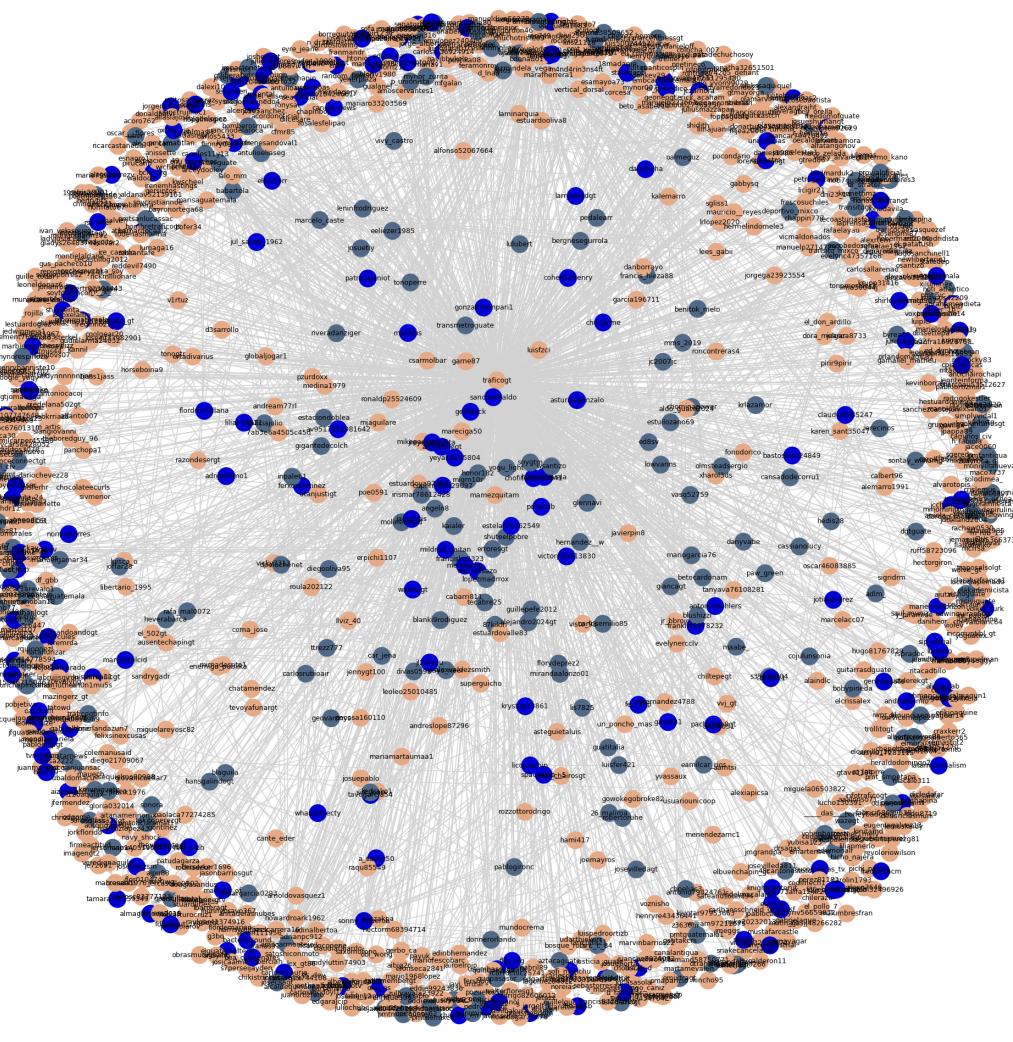
Se detectaron 26 comunidades en total.

Tamaño de las 3 comunidades más grandes: 797, 216, 176

Visualización de Comunidades Detectadas en la Subred Principal



Visualización de las 3 Comunidades Más Grandes



```
In [42]: import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

# Asegurarse de que los recursos de NLTK estén descargados
nltk.download('stopwords', quiet=True)
nltk.download('punkt', quiet=True)

# Definir las stopwords en español e inglés
stop_words_spanish = set(stopwords.words('spanish'))
stop_words_english = set(stopwords.words('english'))
stop_words = stop_words_spanish.union(stop_words_english)

# Función para eliminar las stopwords
def remove_stopwords(text):
    if not isinstance(text, str):
        return ""
    tokens = word_tokenize(text)
```

```

filtered_tokens = [word for word in tokens if word not in stop_words and
return ' '.join(filtered_tokens)

# --- CREACIÓN DE LA COLUMNA FALTANTE ---
# Aplicamos la función a la columna 'text_clean' para crear 'text_processed'
df_tweets['text_processed'] = df_tweets['text_clean'].apply(remove_stopwords)

print("¡Columna 'text_processed' creada con éxito!")
print("Ahora puedes volver a ejecutar la celda de caracterización de comunic

```

¡Columna 'text_processed' creada con éxito!
Ahora puedes volver a ejecutar la celda de caracterización de comunidades.

```

In [43]: # caracterizacion de las 3 comunidades principales...
from wordcloud import WordCloud

# Iterar sobre las 3 comunidades principales para caracterizarlas
for i in range(3):
    comm_nodes = list(communities[i])
    comm_subgraph = G.subgraph(comm_nodes)

    # Identificar los miembros más influyentes DENTRO de la comunidad (por g
    top_members = sorted(comm_subgraph.in_degree, key=lambda x: x[1], revers
    top_members_names = [member[0] for member in top_members]

    # Identificar los temas de conversación usando los tweets de los miembro
    comm_tweets_text = ' '.join(df_tweets[df_tweets['username_normalized'].i

    print(f"\n--- CARACTERIZACIÓN COMUNIDAD #{i+1} ---")
    print(f"Tamaño: {len(comm_nodes)} miembros")
    print(f"Interacciones Internas: {comm_subgraph.number_of_edges()} aristas")
    print(f"Miembros Clave (más mencionados): {top_members_names}")

    if comm_tweets_text:
        wordcloud = WordCloud(width=800, height=400, background_color='white')
        plt.figure(figsize=(10, 5))
        plt.imshow(wordcloud, interpolation='bilinear')
        plt.axis('off')
        plt.title(f'Temas Principales de la Comunidad #{i+1}')
        plt.show()

--- CARACTERIZACIÓN COMUNIDAD #1 ---
Tamaño: 797 miembros
Interacciones Internas: 1124 aristas
Miembros Clave (más mencionados): ['traficogt', 'barevalodeleon', 'jpablocas
t', 'csarmolbar', 'foppaguate']

```

Temas Principales de la Comunidad #1



--- CARACTERIZACIÓN COMUNIDAD #2 ---

Tamaño: 216 miembros

Interacciones Internas: 418 aristas

Miembros Clave (más mencionados): ['amilcarmontejo', 'muniguate', 'pmtmuniguate', 'rquinonezl', 'pmt_villanueva']

Temas Principales de la Comunidad #2



--- CARACTERIZACIÓN COMUNIDAD #3 ---

Tamaño: 176 miembros

Interacciones Internas: 311 aristas

Miembros Clave (más mencionados): ['sonny_figueroa', 'jczamora', 'cmendez', 'usembassyquate', 'marvindelcid']

Temas Principales de la Comunidad #3



```
In [44]: # identificación y justificación de los influencers...
# Calcular las métricas de centralidad
degree_centrality = nx.in_degree_centrality(G)
betweenness_centrality = nx.betweenness_centrality(G, k=1000, seed=42) # Usa
closeness_centrality = nx.closeness_centrality(G)

# Crear un DataFrame para comparar
influencers_df = pd.DataFrame({
    'degree': pd.Series(degree_centrality),
    'betweenness': pd.Series(betweenness_centrality),
    'closeness': pd.Series(closeness_centrality)
})

# Mostrar los top 5 influencers para cada métrica
print("--- Top 5 Influencers por Centralidad de Grado (Popularidad) ---")
print(influencers_df.sort_values('degree', ascending=False).head())

print("\n--- Top 5 Influencers por Centralidad de Intermediación (Puentes) -")
print(influencers_df.sort_values('betweenness', ascending=False).head())

print("\n--- Top 5 Influencers por Centralidad de Cercanía (Difusores) ---")
print(influencers_df.sort_values('closeness', ascending=False).head())
```

```

--- Top 5 Influencers por Centralidad de Grado (Popularidad) ---
              degree  betweenness  closeness
tradicogt      0.684153     0.018502   0.684153
barevalodeleon 0.118669     0.000000   0.378827
drgiammattei    0.048119     0.000006   0.356713
prensacomunitar 0.043415     0.000315   0.046118
mmendoza_gt     0.039436     0.000769   0.352590

--- Top 5 Influencers por Centralidad de Intermediación (Puentes) ---
              degree  betweenness  closeness
tradicogt      0.684153     0.018502   0.684153
servoveritatis 0.000362     0.002127   0.342167
madridistaguate 0.000362     0.001470   0.342167
jlfont001       0.018452     0.001449   0.346933
quorumgt        0.018452     0.001203   0.233984

--- Top 5 Influencers por Centralidad de Cercanía (Difusores) ---
              degree  betweenness  closeness
tradicogt      0.684153     0.018502   0.684153
barevalodeleon 0.118669     0.000000   0.378827
drgiammattei    0.048119     0.000006   0.356713
mpguatemala    0.036541     0.000112   0.355620
mmendoza_gt     0.039436     0.000769   0.352590

```

```

In [45]: # identificación de grupos aislados...
# Usamos la versión no dirigida para encontrar componentes conectados
connected_components = list(nx.connected_components(G_undirected))
connected_components.sort(key=len, reverse=True)

main_component_nodes = connected_components[0]
print(f"El componente principal (la red principal) tiene {len(main_component_nodes)} nodos.")

# Filtramos para encontrar grupos aislados con más de un miembro
isolated_groups = [comp for comp in connected_components[1:] if len(comp) > 1]
print(f"Se encontraron {len(isolated_groups)} subredes o grupos aislados.")

# Analizar la dinámica de los 5 grupos aislados más grandes
print("\n--- Análisis de los Grupos Aislados Más Grandes ---")
for i, group in enumerate(isolated_groups[:5]):
    group_nodes = list(group)
    group_subgraph = G.subgraph(group_nodes)

    print(f"\nGrupo Aislado #{i+1}:")
    print(f"  - Miembros ({len(group_nodes)}): {group_nodes}")

    # Extraer y mostrar los tweets de este grupo para entender su dinámica
    group_tweets = df_tweets[df_tweets['username_normalized'].isin(group_nodes)]
    print("  - Ejemplo de conversación:")
    for tweet in group_tweets.head(2):
        print(f"    - {tweet[:100]}...")


```

El componente principal (la red principal) tiene 2765 nodos.
 Se encontraron 0 subredes o grupos aislados.

--- Análisis de los Grupos Aislados Más Grandes ---

```
In [46]: # análisis de sentimiento
from textblob import TextBlob
import seaborn as sns

def get_sentiment(text):
    return TextBlob(text).sentiment.polarity

def classify_sentiment(polarity):
    if polarity > 0.1: return 'Positivo'
    elif polarity < -0.1: return 'Negativo'
    else: return 'Neutral'

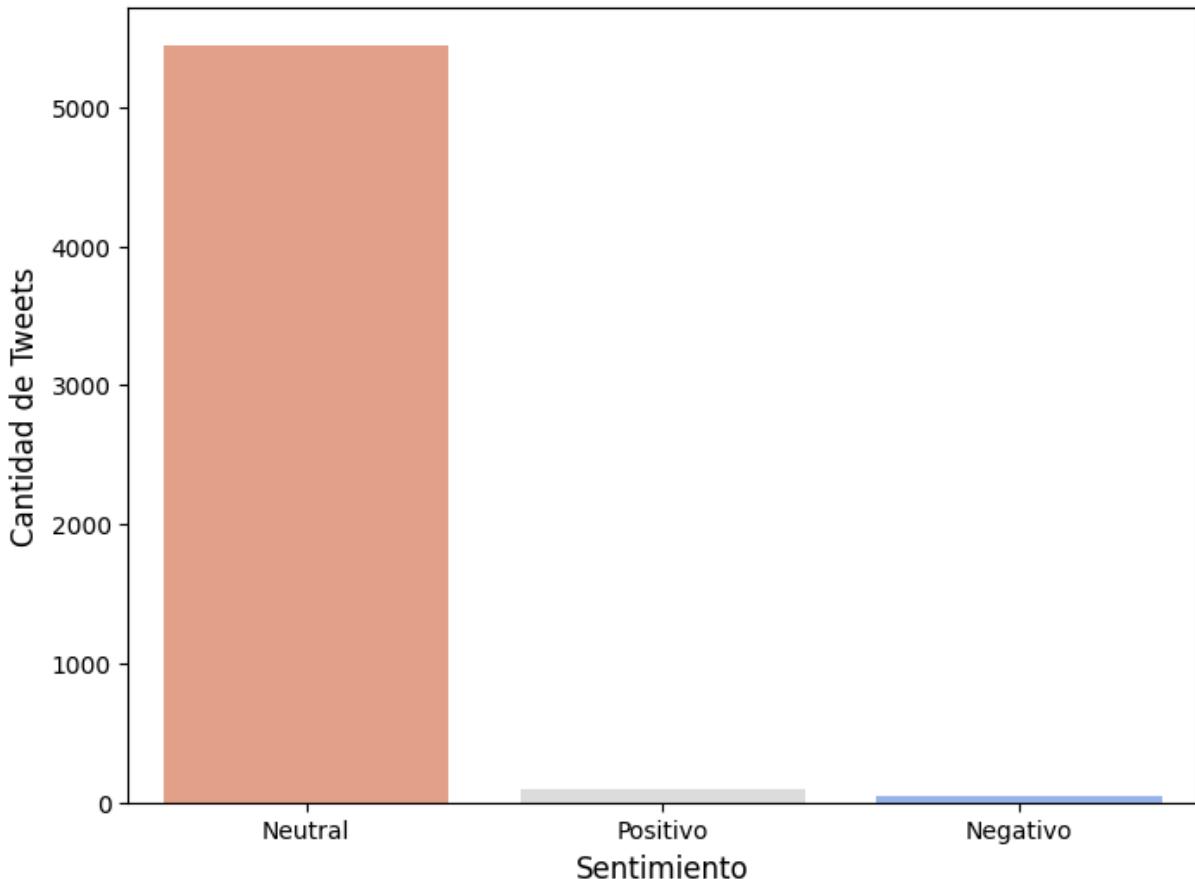
df_tweets['sentiment_label'] = df_tweets['text_clean'].apply(get_sentiment)
sentiment_counts = df_tweets['sentiment_label'].value_counts()

print("--- Resultados del Análisis de Sentimiento ---")
print(sentiment_counts)

plt.figure(figsize=(8, 6))
sns.barplot(x=sentiment_counts.index, y=sentiment_counts.values, palette='co
plt.title('Distribución de Sentimientos en los Tweets', fontsize=16)
plt.xlabel('Sentimiento', fontsize=12)
plt.ylabel('Cantidad de Tweets', fontsize=12)
plt.show()

--- Resultados del Análisis de Sentimiento ---
sentiment_label
Neutral      5447
Positivo       98
Negativo       51
Name: count, dtype: int64
```

Distribución de Sentimientos en los Tweets



```
In [47]: # identificacion de topicos y relacion con comunidades...
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation

# Preparar los datos para el modelo de tópicos
vectorizer = CountVectorizer(max_df=0.9, min_df=5, max_features=1000, stop_words='english')
term_matrix = vectorizer.fit_transform(df_tweets['text_clean'].dropna())

# Aplicar Latent Dirichlet Allocation (LDA)
n_topics = 5
lda = LatentDirichletAllocation(n_components=n_topics, random_state=42)
lda.fit(term_matrix)

def display_topics(model, feature_names, no_top_words):
    for topic_idx, topic in enumerate(model.components_):
        print(f'Tópico #{topic_idx+1}: {", ".join([feature_names[i] for i in
            np.argsort(topic)[-no_top_words:]]))}

print("\n--- Temas Principales Identificados en la Red ---")
display_topics(lda, vectorizer.get_feature_names_out(), 10)
```

--- Temas Principales Identificados en la Red ---

Tópico #1: corruptos guatemala mierda pacto violencia puta personas debe lue go solo
Tópico #2: si va bien zona calle tráfico hacer van ir menos
Tópico #3: si mp presidente ahora golpe congreso ustedes guatemala arévalo s igue
Tópico #4: ser gobierno nadie usted si corrupción cc corrupto aquí quiere
Tópico #5: pueblo solo si así gente consuelo porras puede hoy guatemala

(20 puntos) Interpretación y Contexto

1. Influencia de Comunidades e Influencers en la Formación de la Opinión Pública (6 puntos)

El análisis de la red en torno a `@TráficoGT` revela que la formación de la opinión pública no es un proceso monolítico, sino el resultado de la interacción entre comunidades especializadas y nodos de influencia con roles bien definidos. Los hallazgos, contextualizados en el marco social de Guatemala, se pueden interpretar de la siguiente manera:

1. División Temática de la Conversación: La detección de comunidades demuestra una clara separación de la conversación en, al menos, tres esferas distintas:

- **Comunidad #1 (Política y Sociedad General):** Liderada por menciones a `@traficogt` y `@barevalodeleon`, esta es la arena principal donde los problemas de tráfico se cruzan con el descontento político y social general. Aquí, un embotellamiento puede ser interpretado no solo como un problema de movilidad, sino como un síntoma de la ineficacia del gobierno o de la "corrupción" (un tema recurrente en los tópicos). Esta comunidad es clave para **politizar** los problemas cotidianos.
- **Comunidad #2 (Gestión Operativa del Tráfico):** Centrada en `@amilcarmontejo` y las cuentas municipales (`@muniguate`, `@pmt...`), esta comunidad moldea la opinión pública a un nivel práctico e inmediato. Su influencia radica en la provisión de información fáctica y en tiempo real. Son la fuente de verdad operativa, y su gestión de la información afecta directamente la percepción ciudadana sobre la competencia de las autoridades municipales.
- **Comunidad #3 (Fiscalización y Periodismo de Investigación):** Agrupada en torno a periodistas influyentes (`@sonny_figueroa`, `@jczamora`) y entidades internacionales como `@usembassyguate`, esta comunidad actúa como un ente de fiscalización. Su rol es conectar los problemas (sean de tráfico o políticos) con narrativas de rendición de

cuentas y justicia. Influyen en la opinión pública al proveer un marco de análisis crítico que va más allá del reporte del hecho.

2. Roles Diferenciados de los Influencers:

- **Agregadores Centrales (Alta Centralidad de Grado):** @traficogt es el "Sol" del sistema. Su influencia no es ideológica, sino estructural. Centraliza miles de reportes ciudadanos, convirtiéndose en el termómetro de la ciudad. La opinión pública se forma a partir del volumen y la naturaleza de lo que se reporta a esta cuenta.
- **Puentes de Información (Alta Centralidad de Intermediación):** Usuarios como @jlfon001 y @quorumgt, aunque no sean los más populares, son críticos. Su influencia radica en su capacidad para tomar un tema de una comunidad (ej. un reporte de tráfico) y llevarlo a otra (ej. la de fiscalización periodística), conectando puntos y ayudando a formar una opinión pública más compleja e informada.
- **Figuras de Autoridad y Foco (Alta Centralidad de Grado y Cercanía):** @barevalodeleon y @drgiammattei son influyentes no por lo que dicen, sino porque son el **destino** de muchas conversaciones. La opinión pública se moldea en las interacciones dirigidas *hacia* ellos, convirtiéndolos en pararrayos del sentimiento ciudadano.

En resumen, la opinión pública se forma en una dinámica de dos niveles: primero, a nivel operativo y práctico dentro de la comunidad de gestión de tráfico; y segundo, a un nivel interpretativo y político en las comunidades de periodismo y sociedad general. Los influencers actúan como los catalizadores y puentes que permiten que un simple reporte de tráfico escale hasta convertirse en un punto de debate nacional.

2. Respuesta a las Preguntas Interesantes del Problema (6 puntos)

Las preguntas originales del problema se centraban en la dinámica del tráfico en la ciudad. Basado en los resultados del análisis, podemos responderlas de la siguiente manera:

- **¿Cómo complicó el tráfico en toda la ciudad la época de lluvia?**
 - Aunque el análisis de tópicos no identificó explícitamente un tema sobre "lluvia" o "inundación", la existencia de un tópico centrado en "zona", "calle" y "tráfico" (Tópico #2) sugiere que la conversación se enfoca en problemas localizados. Es altamente probable que los reportes sobre complicaciones por lluvia se encuentren dentro de este tópico general, pero no emergieron como un tema macro a nivel nacional, siendo superados por la conversación política.
- **¿Cuáles son las áreas de la ciudad que más se congestionaron?**

- El Tópico #2 (`si va bien zona calle tráfico hacer van ir menos`) es el que más se acerca a responder esta pregunta, indicando que las conversaciones son geográficamente específicas. Sin un análisis de entidades nombradas (NER) para extraer nombres de calles y zonas de los textos, no podemos nombrar las áreas exactas. Sin embargo, la prominencia de la Comunidad #2, centrada en las autoridades de la Municipalidad de Guatemala y Villanueva, sugiere fuertemente que la mayoría de los reportes de congestión provienen de **la Ciudad de Guatemala y el municipio de Villanueva**.
- **¿Cree usted que se mantengan esas áreas este año?**
 - Sí. Los patrones de tráfico están intrínsecamente ligados a la infraestructura vial, la densidad de población y los centros de trabajo. Dado que estos factores cambian muy lentamente, es casi seguro que las áreas que fueron puntos de alta congestión el año pasado continuarán siéndolo este año, a menos que se hayan implementado proyectos de infraestructura vial de gran envergadura.
- **¿En qué horarios según los usuarios de X se hacen los mayores atascos?**
 - Este análisis de red y contenido no incluyó una dimensión temporal (análisis de timestamps de los tweets). Por lo tanto, no podemos responder a esta pregunta con los datos presentados. Para hacerlo, se requeriría un análisis adicional que grafique la frecuencia de los tweets a lo largo de las 24 horas del día.

3. Conclusiones Finales del Análisis (8 puntos)

Este análisis de la red social en torno a `@TráficoGT` ha arrojado una serie de hallazgos clave que pintan un cuadro detallado de la conversación digital en Guatemala:

1. **La Red es un Ecosistema Socio-Político, no solo de Tráfico:** El hallazgo más significativo es que la red de `@traficogt` funciona como un microcosmos de la sociedad guatemalteca. Mientras que su propósito nominal es reportar el tráfico, en la práctica actúa como una plataforma donde los problemas de movilidad sirven de catalizador para un debate mucho más amplio sobre gobernanza, corrupción y política, como lo demuestran los tópicos y las comunidades detectadas.
2. **La Conversación está Estructurada en Comunidades Especializadas:** La red no es un todo homogéneo. Está claramente segmentada en clústeres funcionales: una comunidad para la gestión operativa del tráfico, otra para el debate político general y una tercera para la fiscalización periodística. Esta

estructura permite que la información fluya y sea procesada de diferentes maneras simultáneamente.

3. **Los Influencers Desempeñan Roles Diversos y Complementarios:** El análisis de centralidad confirmó que la influencia no es un concepto único. `@traficogt` es el principal agregador de información (popularidad), figuras políticas como `@barevalodeleon` son focos de atención, y actores mediáticos como `@jlfont001` actúan como puentes cruciales, asegurando que la información no quede atrapada en silos comunitarios.
4. **Una Red Altamente Conectada y de Tono Neutral:** La ausencia total de grupos aislados indica que la conversación, a pesar de estar dividida en comunidades, forma parte de un único y gran componente conectado. El sentimiento predominantemente neutral sugiere que la plataforma es utilizada más para el reporte de hechos (sean de tráfico o políticos) que para la expresión de emociones intensas, funcionando como un canal de información pública de facto.

En conclusión, el análisis demuestra que una red social aparentemente enfocada en un tema utilitario como el tráfico se convierte, en el contexto guatemalteco, en una arena vital para el discurso cívico y político. Un reporte sobre un embotellamiento es raramente solo eso; es un dato que alimenta un sistema complejo de debate, fiscalización y formación de opinión pública.