PR-Disaster-Tweets

April 3, 2025

1 PR-Disaster-Tweets: Analysis of Public Perception and Media Coverage During Natural Disasters in Puerto Rico

This notebook consolidates the entire repository code for the PR-Disaster-Tweets project. The repository includes analysis scripts for various datasets including HumAID, ISCRAM, a custom-scraped earthquake tweets dataset (January 2020), and advisory tweets (February 2025).

1.1 Repository Structure

```
PR-Disaster-Tweets/
  datasets/                  # All datasets used in the project
     HumAID_maria_tweets/   # HumAID dataset files for Hurricane Maria
     ISCRAM_maria_tweets/   # ISCRAM dataset files for Hurricane Maria
     PR_Earthquake_Tweets_Jan2020/ # Custom-scraped dataset for January 2020 earthquakes
     PR_Advisory_Tweets_Feb_2025/  # Custom-scraped dataset for February 2025 tsunami advisor
  .venv/                     # Virtual environment for dependencies
  CITATION.md                # Citation information
  LICENSE.md                 # License information
  README.md                  # Project documentation
  requirements.txt           # Python dependencies
```

1.2 README.md

2 PR-Disaster-Tweets: Analysis of Public Perception and Media Coverage During Natural Disasters in Puerto Rico

This project focuses on analyzing public perception and media coverage during natural disasters in Puerto Rico, with a particular emphasis on Hurricane Maria (2017), the 2020 earthquakes, and 2025 tsunami advisory events. The analysis combines multiple datasets, including HumAID, ISCRAM18, and custom-scraped datasets, to provide insights into disaster response patterns, public sentiment, and humanitarian needs.

2.0.1 Dataset Details

- **HumAID_maria_tweets:** Contains annotated tweets for Hurricane Maria, including thematic categories.
- **ISCRAM_maria_tweets:** Includes hydrated tweet IDs and image URLs from Hurricane Maria.

- **PR_Earthquake_Tweets_Jan2020:** Custom-scraped tweets for January 2020 earthquakes.
- **PR_Advisory_Tweets_Feb_2025:** Custom-scraped tweets for the February 2025 tsunami advisory.

2.0.2 Running the Analysis

1. Clone the repository and install dependencies (see instructions in the README).
2. Each analysis script (in the `datasets/<dataset>/analysis/` folder) can be run separately. This notebook combines them for a unified view.

2.1 CITATION.md

2.1.1 Citation Information

HumAID Dataset

```
@inproceedings{humaid2020,
    Author = {Firoj Alam, Umair Qazi, Muhammad Imran, Ferda Ofli},
    Booktitle = {15th International Conference on Web and Social Media (ICWSM)},
    Keywords = {Social Media, Crisis Computing, Tweet Text Classification, Disaster Response},
    Title = {HumAID: Human-Annotated Disaster Incidents Data from Twitter},
    Year = {2021}
}
```

ISCRAM Dataset

```
@article{firoj2018twitter,
    title={A Twitter Tale of Three Hurricanes: Harvey, Irma, and Maria},
    author={Alam, Firoj and Ofli, Ferda and Imran, Muhammad and Aupetit, Michael},
    journal={Proc. of ISCRAM, Rochester, USA},
    year={2018}
}
```

Project Citation

```
@misc{humaid_project,
    Author = {Your Name and Collaborators},
    Title = {HumAID: Analysis of Public Perception and Media Coverage During Natural Disasters
    Year = {2024},
    Publisher = {GitHub},
    Journal = {GitHub repository},
    Howpublished = {\url{https://github.com/yourusername/HumAID}}
}
```

Additional References

- HumAID Dataset: Link
- ISCRAM18 Dataset: Link

2.2 Analysis of HumAID Hurricane María Tweets

The following cell includes the code originally found in `datasets/HumAID_maria_tweets/analysis/analyze_huma`

[1]:
```
# File: datasets/HumAID_maria_tweets/analysis/analyze_humaid.py
import os
import re
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS

def load_data(filepath):
    try:
        df = pd.read_csv(filepath)
        print("Data loaded successfully.")
        return df
    except Exception as e:
        print(f"Error loading the file: {e}")
        return None

def preprocess_data(df):
    if 'tweet_text' in df.columns:
        df['tweet_length'] = df['tweet_text'].apply(lambda x: len(x) if␣
 ↪isinstance(x, str) else 0)
    else:
        print("Column 'tweet_text' not found for calculating tweet length.")
    return df

def plot_class_distribution(df):
    if 'class_label' in df.columns:
        plt.figure(figsize=(8, 5))
        sns.countplot(data=df, x='class_label', order=df['class_label'].
 ↪value_counts().index)
        plt.title("Distribution of Class Labels")
        plt.xlabel("Class Label")
        plt.ylabel("Count")
        plt.tight_layout()
        plt.show()
    else:
        print("Column 'class_label' not found for class distribution.")

def plot_split_distribution(df):
    if 'split' in df.columns:
        plt.figure(figsize=(8, 5))
        sns.countplot(data=df, x='split', order=df['split'].value_counts().
 ↪index)
        plt.title("Distribution of Dataset Splits")
```

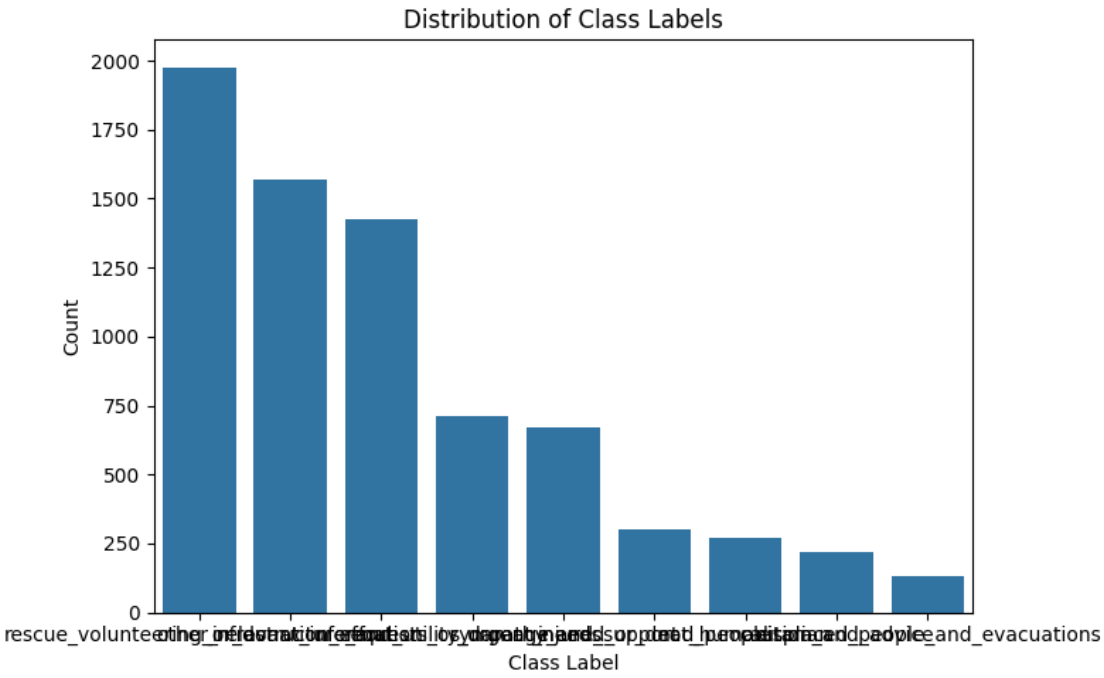
```
        plt.xlabel("Split")
        plt.ylabel("Count")
        plt.tight_layout()
        plt.show()
    else:
        print("Column 'split' not found for split distribution.")

def plot_tweet_length_distribution(df):
    if 'tweet_length' in df.columns:
        plt.figure(figsize=(12, 6))
        plt.hist(df['tweet_length'], bins=30, edgecolor='k', alpha=0.7)
        plt.title("Tweet Length Distribution")
        plt.xlabel("Length (number of characters)")
        plt.ylabel("Frequency")
        plt.tight_layout()
        plt.show()

        plt.figure(figsize=(8, 4))
        sns.boxplot(x=df['tweet_length'])
        plt.title("Tweet Length Boxplot")
        plt.xlabel("Length (number of characters)")
        plt.tight_layout()
        plt.show()
    else:
        print("Column 'tweet_length' is not available for length analysis.")

def generate_word_cloud(df):
    if 'tweet_text' not in df.columns:
        print("Column 'tweet_text' not found for generating word cloud.")
        return
    all_text = " ".join(df['tweet_text'].dropna().astype(str))
    cleaned_text = re.sub(r'https?://\S+', '', all_text)
    cleaned_text = re.sub(r'@\w+', '', cleaned_text)
    cleaned_text = re.sub(r'\bRT\b', '', cleaned_text)
    cleaned_text = re.sub(r'[^A-Za-z\s]', '', cleaned_text)
    cleaned_text = cleaned_text.lower()
    custom_stopwords = {"https", "http", "co", "amp", "rt"}
    all_stopwords = STOPWORDS.union(custom_stopwords)
    wordcloud = WordCloud(width=800, height=400, background_color='white',
                          stopwords=all_stopwords).generate(cleaned_text)
    plt.figure(figsize=(12, 6))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.title("Word Cloud (Cleaned Tweet Text)")
    plt.tight_layout()
    plt.show()
```

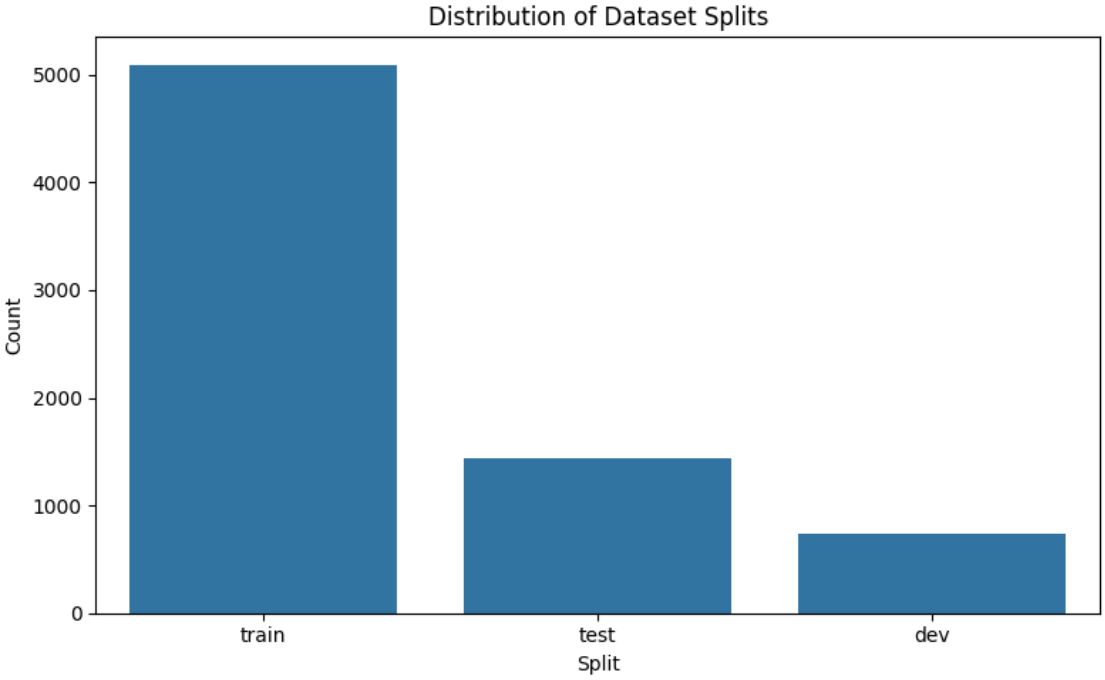
```python
def main():
    filepath = "datasets\HumAID_maria_tweets\HumAID_maria_tweets.csv"
    if not os.path.exists(filepath):
        print(f"The file '{filepath}' does not exist. Check the path.")
        return
    df = load_data(filepath)
    if df is None:
        return
    print("Dataset Information:")
    print(df.info())
    print(df.head())
    df = preprocess_data(df)
    plot_class_distribution(df)
    plot_split_distribution(df)
    plot_tweet_length_distribution(df)
    generate_word_cloud(df)

if __name__ == "__main__":
    main()
```

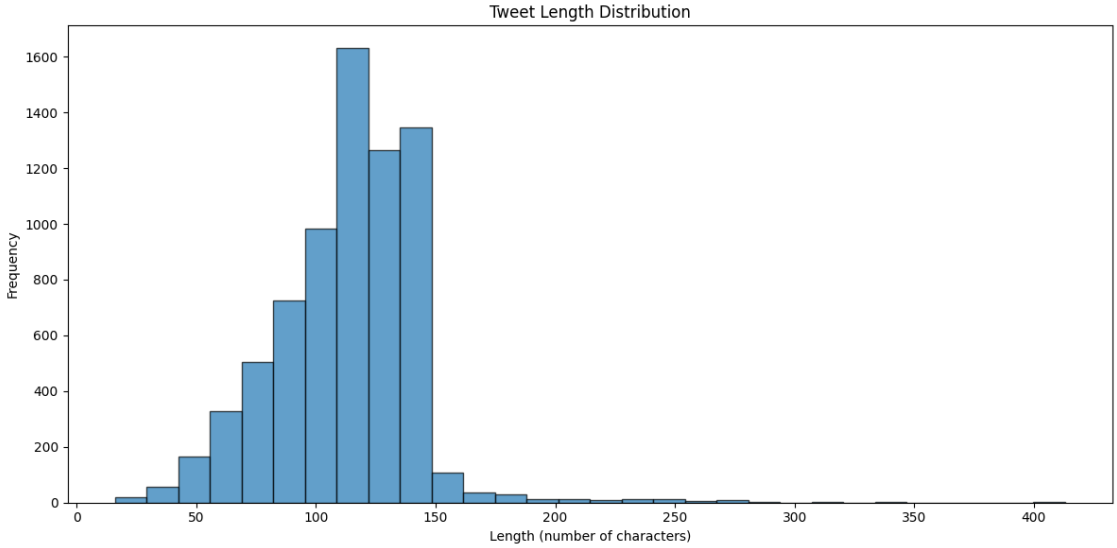
```
<>:90: SyntaxWarning: invalid escape sequence '\H'
<>:90: SyntaxWarning: invalid escape sequence '\H'
C:\Users\Marco\AppData\Local\Temp\ipykernel_28036\906284674.py:90:
SyntaxWarning: invalid escape sequence '\H'
  filepath = "datasets\HumAID_maria_tweets\HumAID_maria_tweets.csv"
```

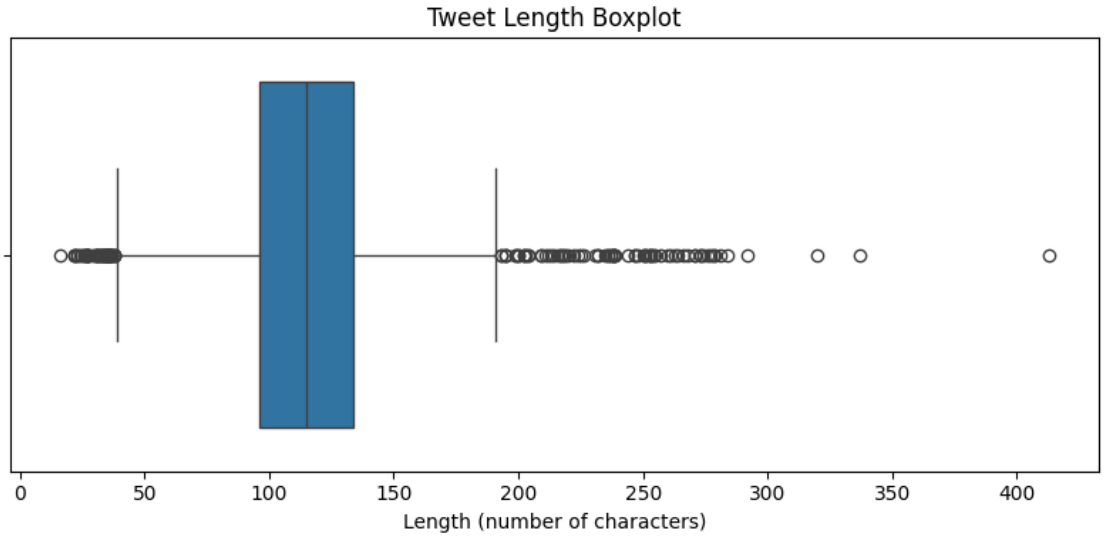
```
Data loaded successfully.
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7278 entries, 0 to 7277
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   tweet_id     7278 non-null   int64
 1   tweet_text   7278 non-null   object
 2   class_label  7278 non-null   object
 3   split        7278 non-null   object
dtypes: int64(1), object(3)
memory usage: 227.6+ KB
None
             tweet_id                                         tweet_text  \
0  914134332226330625  San Juan: Trump lashes out with good reason. #…
1  910783670134476800  Hurricane Maria Live Updates: Catastrophic Flo…
2  912134938727780355  Getting food to the island is, obviously, crit…
3  910669838842056704  My heart breaks for the families in Puerto Ric…
4  912287091026997248  #B-FAST sending medical, reconstruction &amp; …

                          class_label  split
```

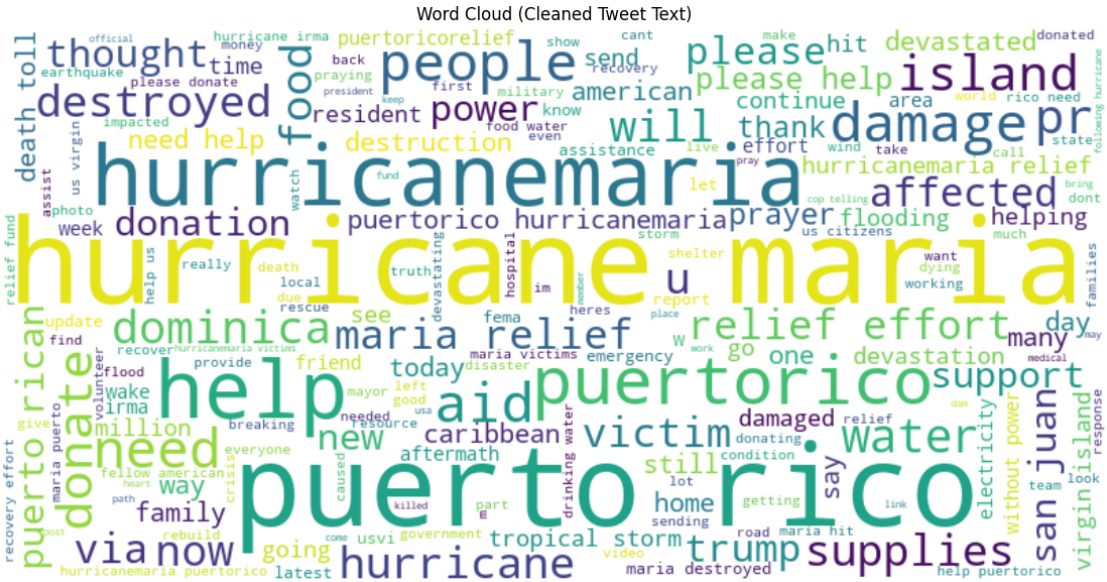
```
0              other_relevant_information  train
1                     caution_and_advice  train
2  rescue_volunteering_or_donation_effort  train
3                   sympathy_and_support  train
4  rescue_volunteering_or_donation_effort  train
```






Tweet Length Distribution
Frequency
Length (number of characters)
0
200
400
600
800
1000
1200
1400
1600
0
50
100
150
200
250
300
350
400


Tweet Length Boxplot
Length (number of characters)
0
50
100
150
200
250
300
350
400



2.3 Maria Analysis Report

Below is the content from the `maria_analysis.md` file summarizing the Hurricane Maria tweet analysis.

```
# Hurricane Maria Tweet Analysis

## Overview
Total tweets analyzed: 7278

## Data Split Distribution
| Split | Number of Tweets | Percentage |
|-------|------------------|------------|
| train | 5094 | 70.0% |
| dev | 742 | 10.2% |
| test | 1442 | 19.8% |

## Label Distribution
| Label | Count | Percentage |
|-------|-------|------------|
| rescue_volunteering_or_donation_effort | 1977 | 27.2% |
| other_relevant_information | 1568 | 21.5% |
| infrastructure_and_utility_damage | 1427 | 19.6% |
| requests_or_urgent_needs | 711 | 9.8% |
| sympathy_and_support | 672 | 9.2% |
| injured_or_dead_people | 302 | 4.1% |
| not_humanitarian | 270 | 3.7% |
| caution_and_advice | 220 | 3.0% |
```

```
| displaced_people_and_evacuations | 131 | 1.8% |

## Visualizations
### Split Distribution
![Split Distribution](maria_splits_distribution.png)

### Label Distribution
![Label Distribution](maria_label_distribution.png)

## Key Findings
### Most Common Tweet Categories:
- rescue_volunteering_or_donation_effort: 1977 tweets (27.2%)
- other_relevant_information: 1568 tweets (21.5%)
- infrastructure_and_utility_damage: 1427 tweets (19.6%)

### Infrastructure and Urgent Needs:
- Infrastructure damage related tweets: 1427
- Urgent needs related tweets: 711
- Combined: 2138 tweets (29.4% of total)
```

2.4 Analysis of ISCRAM Hurricane Maria Tweets

The next section contains the code from datasets/ISCRAM_maria_tweets/analysis/analyze_ISCRAM_tweets.py

[2]:
```python
# File: datasets/ISCRAM_maria_tweets/analysis/analyze_ISCRAM_tweets.py
import os
import re
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS

def load_data(filepath):
    try:
        df = pd.read_csv(filepath)
        print("Datos cargados exitosamente.")
        return df
    except Exception as e:
        print(f"Error al cargar el archivo: {e}")
        return None

def preprocess_data(df):
    if 'created_at' in df.columns:
        try:
            df['created_at'] = pd.to_datetime(df['created_at'], format='%a %b␣
 ↪%d %H:%M:%S %z %Y', errors='coerce')
            print("Columna 'created_at' convertida a datetime.")
        except Exception as e:
```

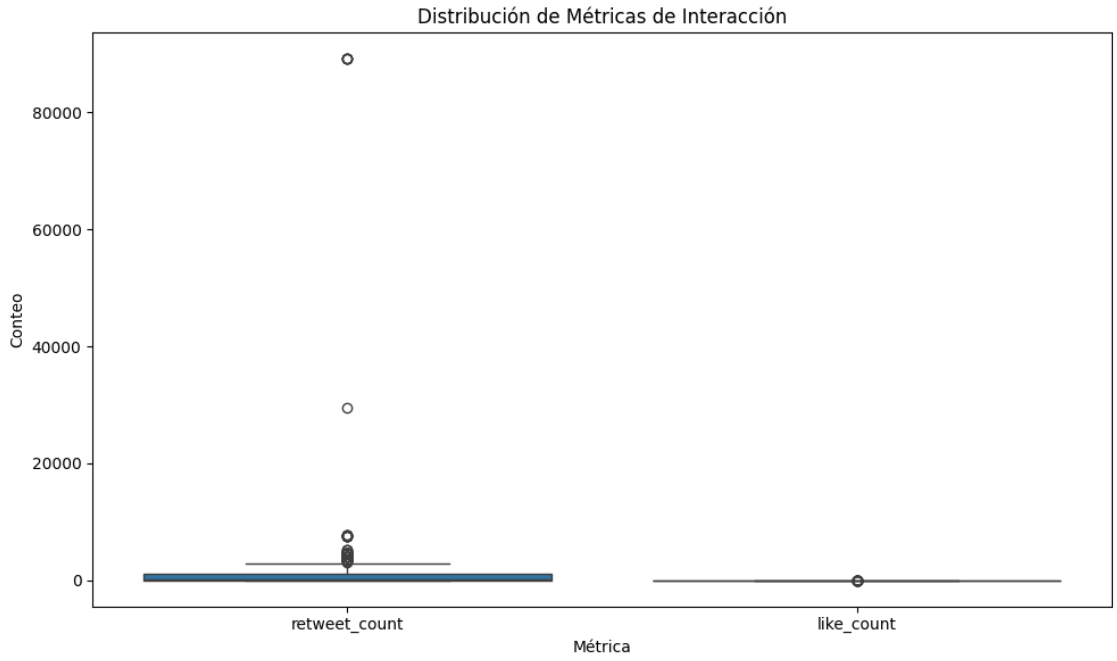
```
            print(f"Error al convertir 'created_at': {e}")
    else:
        print("No se encontró la columna 'created_at'.")
    if 'text' in df.columns:
        df['tweet_length'] = df['text'].apply(lambda x: len(x) if isinstance(x,␣
 ↪str) else 0)
    else:
        print("No se encontró la columna 'text' para calcular la longitud de␣
 ↪los tuits.")
    return df

def plot_engagement_metrics(df):
    engagement_cols = []
    for col in ['retweet_count', 'like_count']:
        if col in df.columns:
            engagement_cols.append(col)
    if engagement_cols:
        df_engagement = df[engagement_cols].melt(var_name="Métrica",␣
 ↪value_name="Conteo")
        plt.figure(figsize=(10, 6))
        sns.boxplot(x="Métrica", y="Conteo", data=df_engagement)
        plt.title("Distribución de Métricas de Interacción")
        plt.tight_layout()
        plt.show()
    else:
        print("No se encontraron columnas de métricas de interacción␣
 ↪(retweet_count, like_count).")

def plot_tweet_length_distribution(df):
    if 'tweet_length' in df.columns:
        plt.figure(figsize=(12, 6))
        plt.hist(df['tweet_length'], bins=30, edgecolor='k', alpha=0.7)
        plt.title("Distribución de la Longitud de los Tuits")
        plt.xlabel("Longitud (número de caracteres)")
        plt.ylabel("Frecuencia")
        plt.tight_layout()
        plt.show()

        plt.figure(figsize=(8, 4))
        sns.boxplot(x=df['tweet_length'])
        plt.title("Diagrama de Caja de la Longitud de los Tuits")
        plt.xlabel("Longitud (número de caracteres)")
        plt.tight_layout()
        plt.show()
    else:
        print("La columna 'tweet_length' no está disponible para el análisis de␣
 ↪longitud.")
```

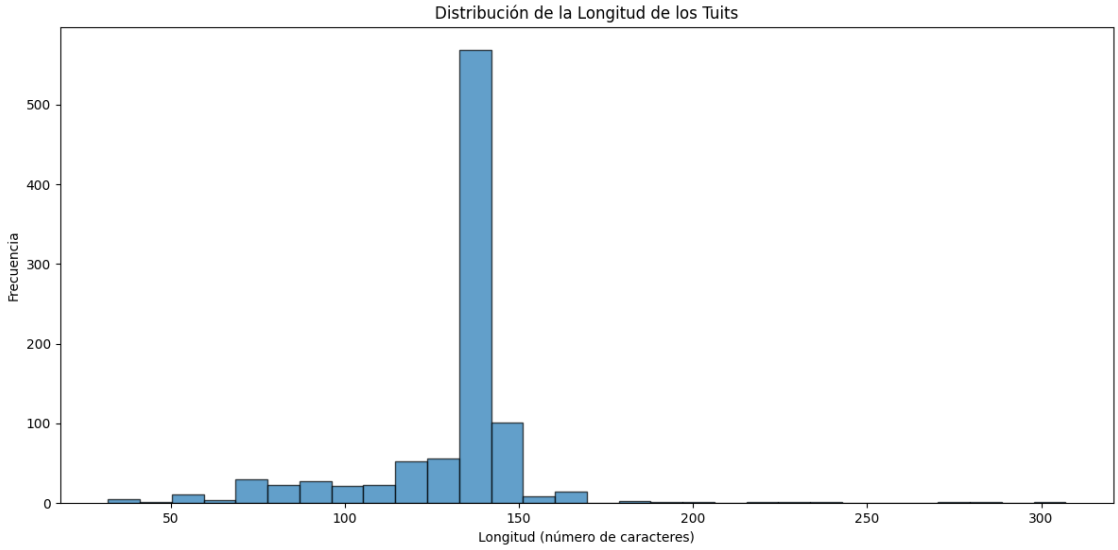
```python
def plot_likes_distribution(df):
    if 'like_count' in df.columns:
        plt.figure(figsize=(12, 6))
        plt.hist(df['like_count'], bins=30, edgecolor='k', alpha=0.7)
        plt.title("Distribución de Likes en los Tuits")
        plt.xlabel("Número de Likes")
        plt.ylabel("Frecuencia")
        plt.tight_layout()
        plt.show()
    else:
        print("La columna 'like_count' no está disponible para analizar la␣
 ↪distribución de likes.")

def plot_length_vs_likes(df):
    if 'tweet_length' in df.columns and 'like_count' in df.columns:
        plt.figure(figsize=(10, 6))
        sns.scatterplot(x='tweet_length', y='like_count', data=df, alpha=0.7)
        plt.title("Relación entre Longitud de Tuits y Número de Likes")
        plt.xlabel("Longitud del Tuit (caracteres)")
        plt.ylabel("Número de Likes")
        plt.tight_layout()
        plt.show()
    else:
        print("No se encontraron las columnas necesarias ('tweet_length',␣
 ↪'like_count') para este análisis.")

def generate_word_cloud(df):
    if 'text' not in df.columns:
        print("La columna 'text' no se encontró para generar la nube de␣
 ↪palabras.")
        return
    all_text = " ".join(df['text'].dropna().astype(str))
    cleaned_text = re.sub(r'https?://\S+', '', all_text)
    cleaned_text = re.sub(r'@\w+', '', cleaned_text)
    cleaned_text = re.sub(r'\bRT\b', '', cleaned_text)
    cleaned_text = re.sub(r'[^A-Za-záéíóúñüÁÉÍÓÚÑÜ\s]', '', cleaned_text)
    cleaned_text = cleaned_text.lower()
    custom_stopwords = {"https", "http", "co", "amp"}
    stopwords = STOPWORDS.union(custom_stopwords)
    wordcloud = WordCloud(width=800, height=400, background_color='white',
                          stopwords=stopwords).generate(cleaned_text)
    plt.figure(figsize=(12, 6))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.title("Nube de Palabras (Texto Limpio)")
    plt.tight_layout()
```

```
    plt.show()

def main():
    filepath = "datasets\ISCRAM_maria_tweets\ISCRAM_maria_tweets.csv"
    if not os.path.exists(filepath):
        print(f"El archivo '{filepath}' no existe. Verifica la ruta.")
        return
    df = load_data(filepath)
    if df is None:
        return
    print("Información del dataset:")
    print(df.info())
    print(df.head())
    df = preprocess_data(df)
    plot_engagement_metrics(df)
    plot_tweet_length_distribution(df)
    plot_likes_distribution(df)
    plot_length_vs_likes(df)
    generate_word_cloud(df)

if __name__ == "__main__":
    main()
```

```
Datos cargados exitosamente.
Información del dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 959 entries, 0 to 958
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   id             959 non-null    int64
 1   text           959 non-null    object
 2   created_at     959 non-null    object
 3   like_count     959 non-null    int64
 4   retweet_count  959 non-null    int64
 5   lang           959 non-null    object
 6   username       959 non-null    object
dtypes: int64(3), object(4)
memory usage: 52.6+ KB
None
                   id                                                text  \
0  914278688144883713  RT @joebereta: Hey @realDonaldTrump you're a r…
1  914278695657000960  RT @Newsweek: Meet Carmen Yulín Cruz, the woma…
2  914278688144883713  RT @joebereta: Hey @realDonaldTrump you're a r…
3  914278695657000960  RT @Newsweek: Meet Carmen Yulín Cruz, the woma…
4  914278698769174528  RT @CBPFlorida: U.S. Customs and Border Protec…

                      created_at  like_count  retweet_count lang  \
```

```
0  Sun Oct 01 00:00:00 +0000 2017           0            169   en
1  Sun Oct 01 00:00:02 +0000 2017           0             32   en
2  Sun Oct 01 00:00:00 +0000 2017           0            169   en
3  Sun Oct 01 00:00:02 +0000 2017           0             32   en
4  Sun Oct 01 00:00:02 +0000 2017           0           1583   en

       username
0      grizzy_m
1      harva352
2      grizzy_m
3      harva352
4  orlisara0927
Columna 'created_at' convertida a datetime.

<>:113: SyntaxWarning: invalid escape sequence '\I'
<>:113: SyntaxWarning: invalid escape sequence '\I'
C:\Users\Marco\AppData\Local\Temp\ipykernel_28036\4144205790.py:113:
SyntaxWarning: invalid escape sequence '\I'
  filepath = "datasets\ISCRAM_maria_tweets\ISCRAM_maria_tweets.csv"
```




Distribución de la Longitud de los Tuits
Frecuencia
0
100
200
300
400
500
50
100
150
200
250
300
Longitud (número de caracteres)

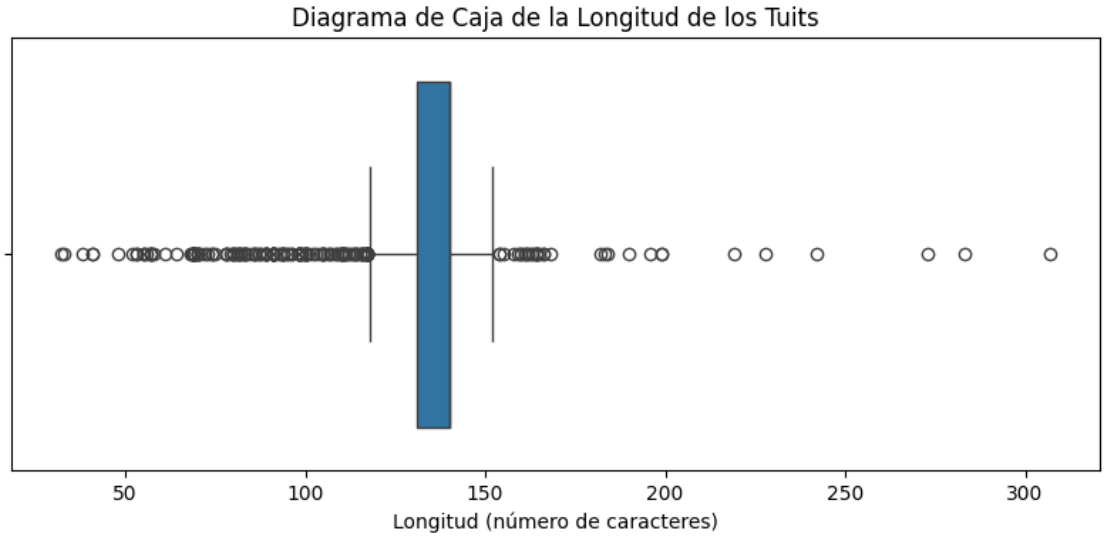
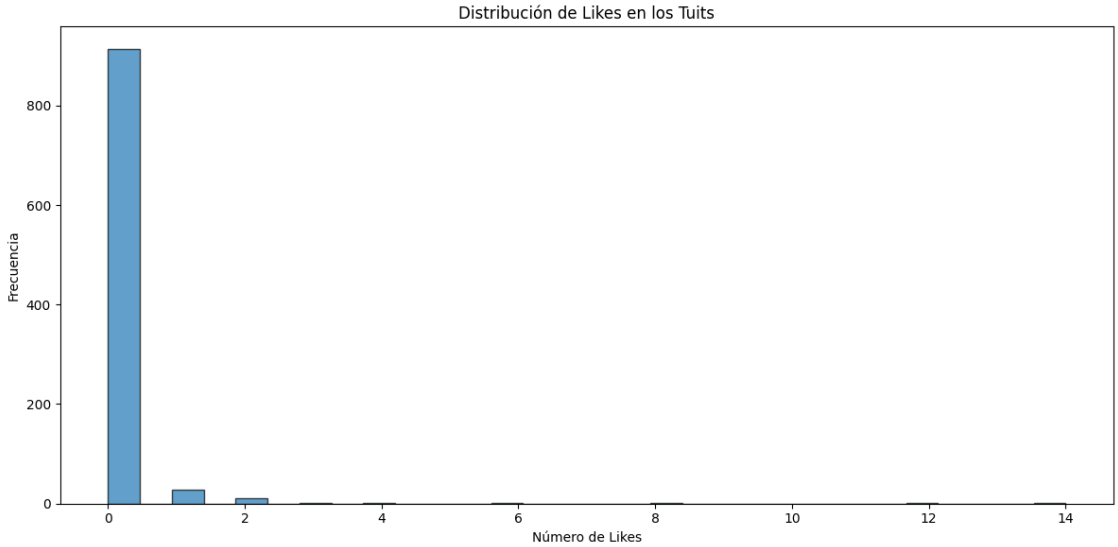
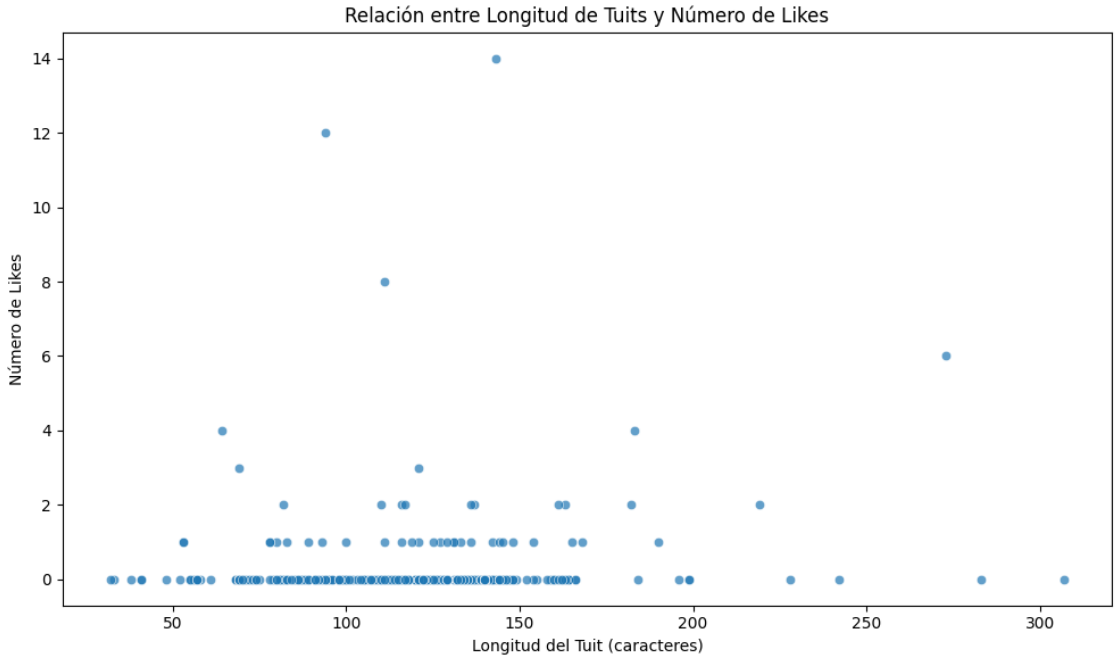
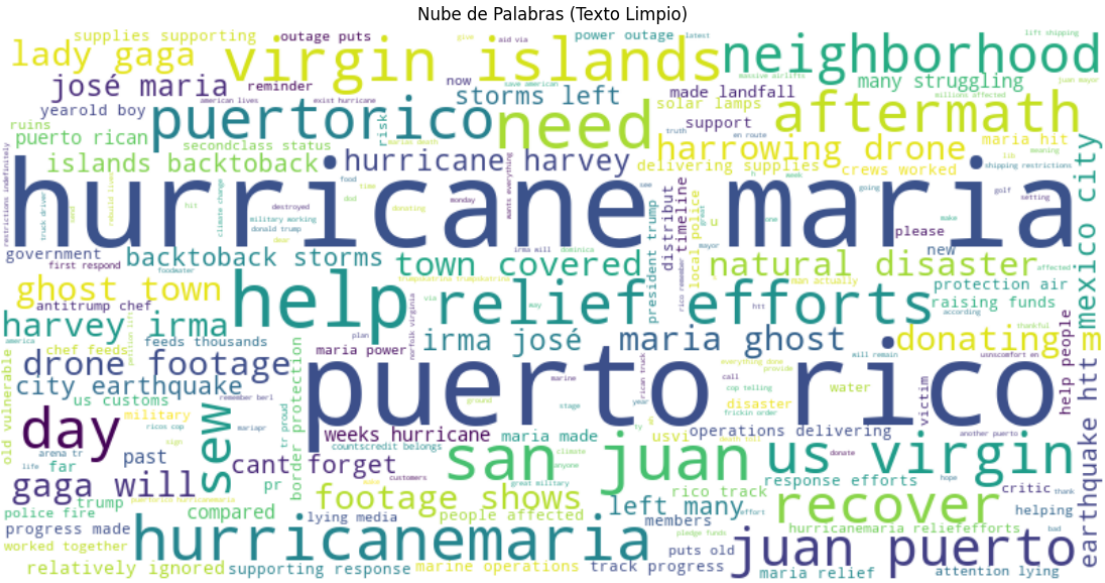


Diagrama de Caja de la Longitud de los Tuits
50
100
150
200
250
300
Longitud (número de caracteres)


Distribución de Likes en los Tuits
Frecuencia
Número de Likes
0
200
400
600
800
0
2
4
6
8
10
12
14


Relación entre Longitud de Tuits y Número de Likes
Número de Likes
Longitud del Tuit (caracteres)
0
2
4
6
8
10
12
14
50
100
150
200
250
300



2.5 Analysis of Advisory Tweets (Feb 2025)

The following cell contains the code from `datasets/PR_Advisory_Tweets_Feb_2025/analysis/analyze_Feb2025`

[3]:
```python
# File: datasets/PR_Advisory_Tweets_Feb_2025/analysis/analyze_Feb2025_tweets.py
import os
import re
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS

def load_data(filepath):
    try:
        df = pd.read_csv(filepath)
        print("Datos cargados exitosamente.")
        return df
    except Exception as e:
        print(f"Error al cargar el archivo: {e}")
        return None

def preprocess_data(df):
    if 'UTC_Time' in df.columns:
        try:
            df['UTC_Time'] = pd.to_datetime(df['UTC_Time'], errors='coerce')
            print("Columna 'UTC_Time' convertida a datetime.")
        except Exception as e:
            print(f"Error al convertir 'UTC_Time': {e}")
```

```
    else:
        print("La columna 'UTC_Time' no se encontró.")
    if 'Tweet_Content' in df.columns:
        df['tweet_length'] = df['Tweet_Content'].apply(lambda x: len(x) if␣
 ↪isinstance(x, str) else 0)
    else:
        print("La columna 'Tweet_Content' no se encontró para calcular la␣
 ↪longitud de los tuits.")
    return df

def plot_engagement_metrics(df):
    interaction_cols = ['Reply_Count', 'Repost_Count', 'Like_Count',␣
 ↪'Bookmark_Count']
    existing_cols = [col for col in interaction_cols if col in df.columns]
    if existing_cols:
        df_interactions = df[existing_cols].melt(var_name="Métrica",␣
 ↪value_name="Conteo")
        plt.figure(figsize=(10, 6))
        sns.boxplot(x="Métrica", y="Conteo", data=df_interactions)
        plt.title("Distribución de Métricas de Interacción")
        plt.tight_layout()
        plt.show()
    else:
        print("No se encontraron columnas de interacción para visualizar.")

def plot_tweet_length_distribution(df):
    if 'tweet_length' in df.columns:
        plt.figure(figsize=(12, 6))
        plt.hist(df['tweet_length'], bins=30, edgecolor='k', alpha=0.7)
        plt.title("Distribución de la Longitud de los Tuits")
        plt.xlabel("Longitud (número de caracteres)")
        plt.ylabel("Frecuencia")
        plt.tight_layout()
        plt.show()

        plt.figure(figsize=(8, 4))
        sns.boxplot(x=df['tweet_length'])
        plt.title("Diagrama de Caja de la Longitud de los Tuits")
        plt.xlabel("Longitud (número de caracteres)")
        plt.tight_layout()
        plt.show()
    else:
        print("La columna 'tweet_length' no está disponible para el análisis de␣
 ↪longitud.")

def plot_likes_distribution(df):
    if 'Like_Count' in df.columns:
```

```
        plt.figure(figsize=(12, 6))
        plt.hist(df['Like_Count'], bins=30, edgecolor='k', alpha=0.7)
        plt.title("Distribución de Likes en los Tuits")
        plt.xlabel("Número de Likes")
        plt.ylabel("Frecuencia")
        plt.tight_layout()
        plt.show()
    else:
        print("La columna 'Like_Count' no está disponible para analizar la␣
 ↪distribución de likes.")

def plot_length_vs_likes(df):
    if 'tweet_length' in df.columns and 'Like_Count' in df.columns:
        plt.figure(figsize=(10, 6))
        sns.scatterplot(x='tweet_length', y='Like_Count', data=df, alpha=0.7)
        plt.title("Relación entre Longitud de Tuits y Número de Likes")
        plt.xlabel("Longitud del Tuit (caracteres)")
        plt.ylabel("Número de Likes")
        plt.tight_layout()
        plt.show()
    else:
        print("No se encontraron las columnas necesarias ('tweet_length',␣
 ↪'Like_Count') para este análisis.")

def plot_language_distribution(df):
    if 'Language' in df.columns:
        lang_counts = df['Language'].value_counts()
        plt.figure(figsize=(8, 5))
        sns.barplot(x=lang_counts.index, y=lang_counts.values)
        plt.title("Distribución de Idiomas de los Tuits")
        plt.xlabel("Idioma")
        plt.ylabel("Cantidad de Tuits")
        plt.tight_layout()
        plt.show()
    else:
        print("La columna 'Language' no se encontró para visualizar la␣
 ↪distribución de idiomas.")

def generate_word_cloud(df):
    if 'Tweet_Content' not in df.columns:
        print("La columna 'Tweet_Content' no se encontró para generar la nube␣
 ↪de palabras.")
        return
    all_text = " ".join(df['Tweet_Content'].dropna().astype(str))
    cleaned_text = re.sub(r'https?://\S+', '', all_text)
    cleaned_text = re.sub(r'@\w+', '', cleaned_text)
    cleaned_text = re.sub(r'\bRT\b', '', cleaned_text)
```

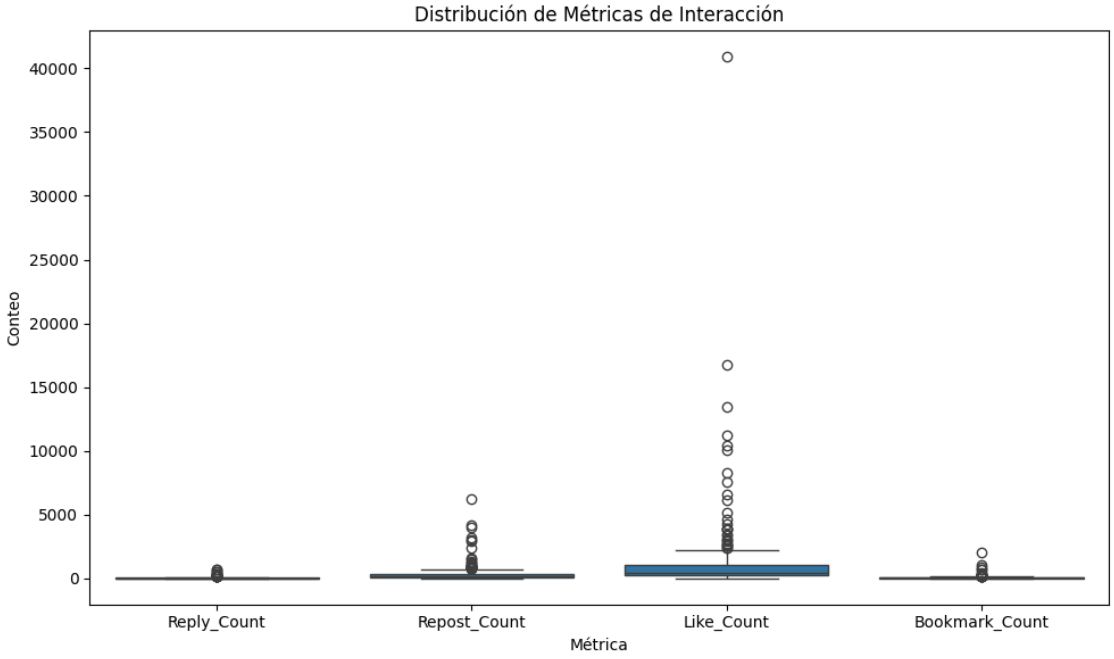
```
    cleaned_text = re.sub(r'[^A-Za-záéíóúñüÁÉÍÓÚÑÜ\s]', '', cleaned_text)
    cleaned_text = cleaned_text.lower()
    spanish_stopwords = {
        "de", "el", "que", "se", "la", "en", "por", "los", "las", "del", "al",
        "un", "una", "con", "para", "este", "esta", "estos", "estas", "ese",
        "esa", "esos", "esas", "y", "o", "u", "pero", "su", "sus", "porque",
        "son", "un", "una", "ser", "sido", "ha", "han", "hay", "qué", "etc"
    }
    custom_stopwords = {"https", "http", "co", "amp"}
    all_stopwords = STOPWORDS.union(spanish_stopwords).union(custom_stopwords)
    wordcloud = WordCloud(
        width=800,
        height=400,
        background_color='white',
        stopwords=all_stopwords
    ).generate(cleaned_text)
    plt.figure(figsize=(12, 6))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.title("Nube de Palabras (Texto Limpio)")
    plt.tight_layout()
    plt.show()

def main():
    filepath =␣
 ↪"datasets\PR_Advisory_Tweets_Feb_2025\PR_Advisory_Tweets_Feb_2025.csv"
    if not os.path.exists(filepath):
        print(f"El archivo '{filepath}' no existe. Verifica la ruta.")
        return
    df = load_data(filepath)
    if df is None:
        return
    print("Información del dataset:")
    print(df.info())
    print(df.head())
    df = preprocess_data(df)
    plot_engagement_metrics(df)
    plot_tweet_length_distribution(df)
    plot_likes_distribution(df)
    plot_length_vs_likes(df)
    plot_language_distribution(df)
    generate_word_cloud(df)

if __name__ == "__main__":
    main()
```

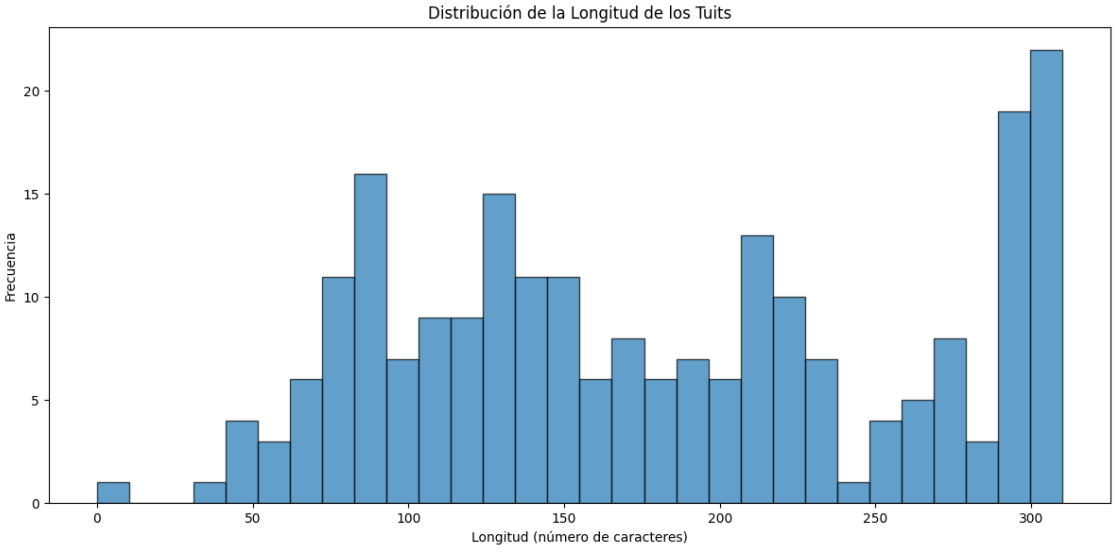
```
Datos cargados exitosamente.
Información del dataset:
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 229 entries, 0 to 228
Data columns (total 23 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Query_Str               229 non-null    object
 1   Post_URL                229 non-null    object
 2   Author_Name             228 non-null    object
 3   Author_Web_Page_URL     229 non-null    object
 4   Author_Handle           228 non-null    object
 5   Verified_Status         228 non-null    object
 6   UTC_Time                228 non-null    object
 7   Ads                     229 non-null    bool
 8   Tweet_Content           228 non-null    object
 9   Post_ID                 228 non-null    float64
 10  Tweet_URL               228 non-null    object
 11  Reply_Count             228 non-null    float64
 12  Repost_Count            229 non-null    int64
 13  Like_Count              229 non-null    int64
 14  View_Count              229 non-null    int64
 15  Bookmark_Count          229 non-null    int64
 16  Tweet_Image_URL         155 non-null    object
 17  Replying_to             229 non-null    bool
 18  Reply_to_Whom           25 non-null     object
 19  Reply_to_Whom_URL       25 non-null     object
 20  Reply_to_Whom_Username  17 non-null     object
 21  Reply_to_Whom_Handle    25 non-null     object
 22  Language                228 non-null    object
dtypes: bool(2), float64(2), int64(4), object(15)
memory usage: 38.1+ KB
None
                                          Query_Str  \
0  Puerto Rico (tsunami OR sismo OR terremoto OR …
1  Puerto Rico (tsunami OR sismo OR terremoto OR …
2  Puerto Rico (tsunami OR sismo OR terremoto OR …
3  Puerto Rico (tsunami OR sismo OR terremoto OR …
4  Puerto Rico (tsunami OR sismo OR terremoto OR …

                                           Post_URL  \
0  https://x.com/search?q=Puerto Rico (tsunami OR…
1  https://x.com/search?q=Puerto Rico (tsunami OR…
2  https://x.com/search?q=Puerto Rico (tsunami OR…
3  https://x.com/search?q=Puerto Rico (tsunami OR…
4  https://x.com/search?q=Puerto Rico (tsunami OR…

                      Author_Name            Author_Web_Page_URL  \
0                             ASB          https://x.com/ASB2509
1  Liga ARCO Mexicana del Pacífico        https://x.com/Liga_Arco
```

```
2                    Julio Rangel       https://x.com/julioranr_
3                  Emergencias Ec    https://x.com/EmergenciasEc
4                          https://x.com/Lucia1041903411

     Author_Handle Verified_Status                  UTC_Time    Ads  \
0          ASB2509            True  2025-02-07 16:01:38+00:00  False
1        Liga_Arco            True  2025-02-07 06:27:21+00:00  False
2       julioranr_           False  2025-02-07 14:28:45+00:00  False
3    EmergenciasEc            True  2025-02-07 17:10:06+00:00  False
4  Lucia1041903411           False  2025-02-07 18:49:01+00:00  False

                                      Tweet_Content       Post_ID  …  \
0  JAJAJAJAJAJAJAJAJAJAJAJAJAJAJAJAJAJAJAJAJAJA…  1.887894e+18  …
1  Puerto Rico se lleva el juego por el 3er lugar…  1.887750e+18  …
2  El refuerzo de Algodoneros Isan Díaz tuvo una …  1.887871e+18  …
3  Urgente!\nSe reporta sicariato en estos moment…  1.887912e+18  …
4    \n\n   Parte del Todo   \n\n          …  1.887937e+18  …

  Like_Count  View_Count  Bookmark_Count  \
0       1471       19137              28
1         52        2848               0
2         76        3104               3
3        890      121923              36
4        234        2497               4

                                    Tweet_Image_URL  Replying_to  \
0    https://pbs.twimg.com/media/GjMmCneWwAAuwul.jpg        False
1    https://pbs.twimg.com/media/GjKimNpWkAATfRf.jpg        False
2  https://pbs.twimg.com/ext_tw_video_thumb/18877…        False
3  https://pbs.twimg.com/ext_tw_video_thumb/18879…        False
4  https://pbs.twimg.com/ext_tw_video_thumb/18879…        False

   Reply_to_Whom Reply_to_Whom_URL  Reply_to_Whom_Username  \
0            NaN               NaN                     NaN
1            NaN               NaN                     NaN
2            NaN               NaN                     NaN
3            NaN               NaN                     NaN
4            NaN               NaN                     NaN

  Reply_to_Whom_Handle Language
0                  NaN       es
1                  NaN       es
2                  NaN       es
3                  NaN       es
4                  NaN      und

[5 rows x 23 columns]
Columna 'UTC_Time' convertida a datetime.
```

```
<>:134: SyntaxWarning: invalid escape sequence '\P'
<>:134: SyntaxWarning: invalid escape sequence '\P'
C:\Users\Marco\AppData\Local\Temp\ipykernel_28036\3666516517.py:134:
SyntaxWarning: invalid escape sequence '\P'
  filepath =
"datasets\PR_Advisory_Tweets_Feb_2025\PR_Advisory_Tweets_Feb_2025.csv"
```





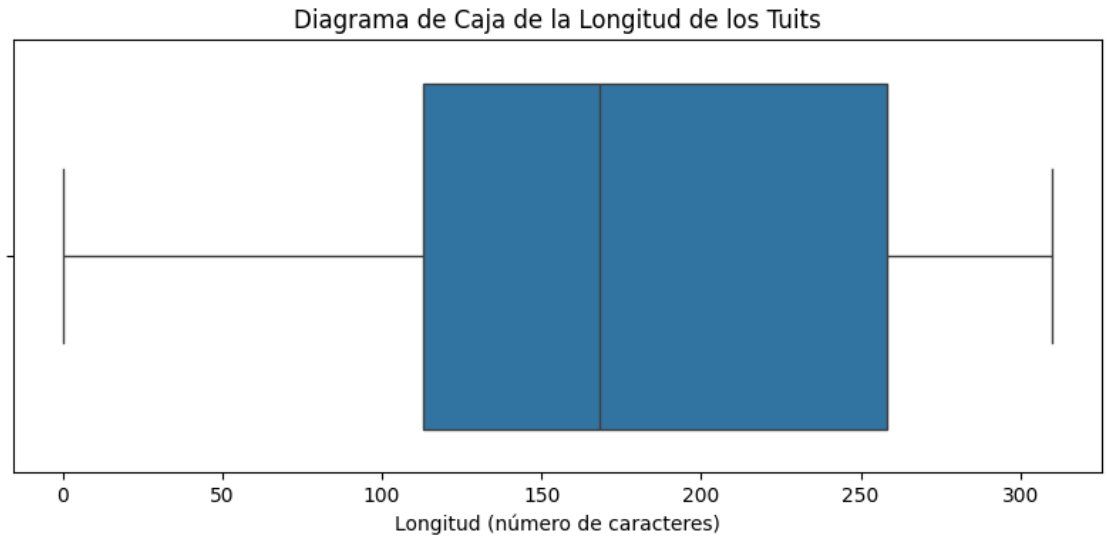
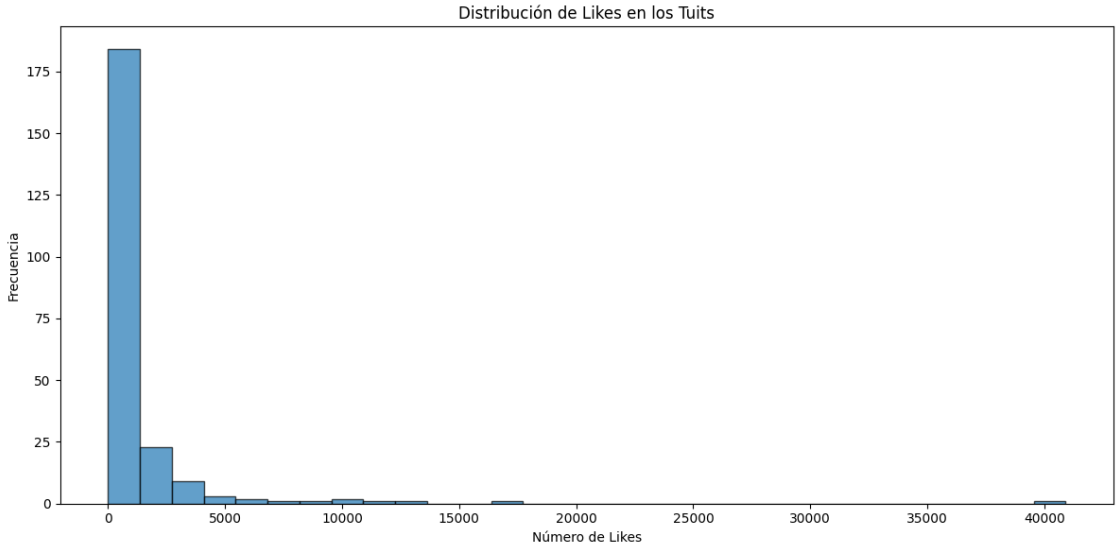
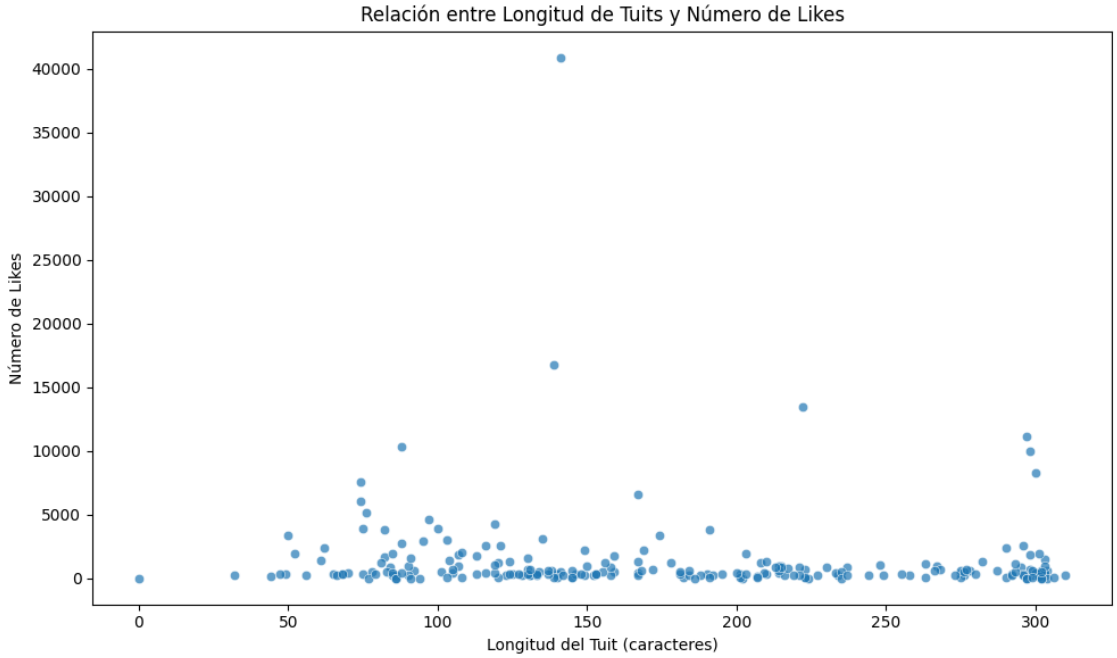
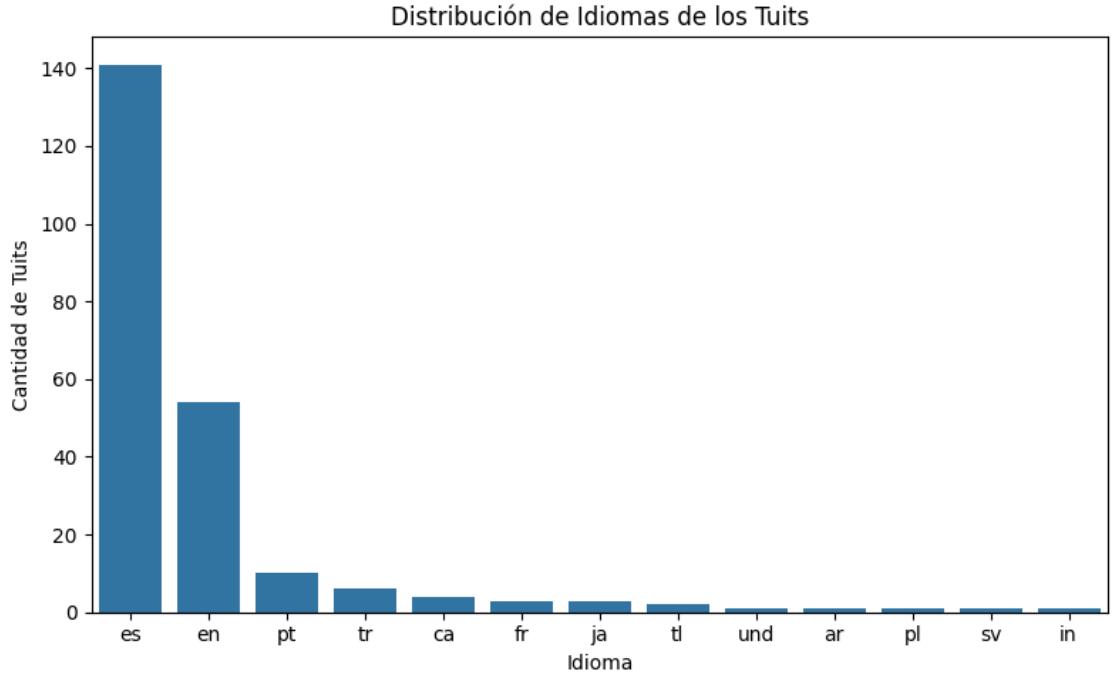
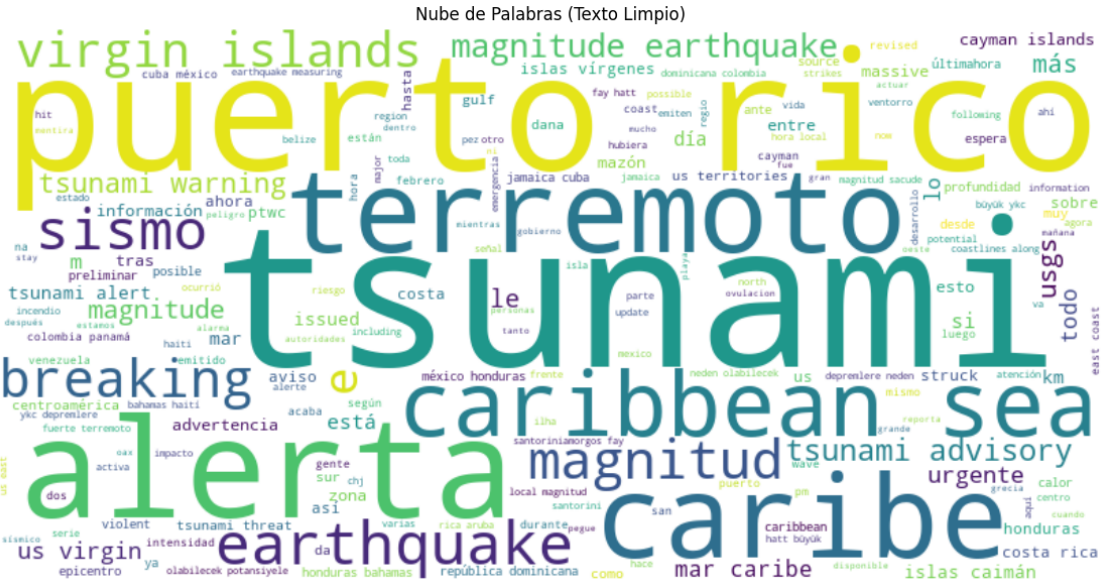


Diagrama de Caja de la Longitud de los Tuits
0
50
100
150
200
250
300
Longitud (número de caracteres)


Distribución de Likes en los Tuits
Frecuencia
0
25
50
75
100
125
150
175
0
5000
10000
15000
20000
25000
30000
35000
40000
Número de Likes


Relación entre Longitud de Tuits y Número de Likes
Número de Likes
Longitud del Tuit (caracteres)
0
5000
10000
15000
20000
25000
30000
35000
40000
0
50
100
150
200
250
300


Distribución de Idiomas de los Tuits
Cantidad de Tuits
Idioma
0
20
40
60
80
100
120
140
es
en
pt
tr
ca
fr
ja
tl
und
ar
pl
sv
in



2.6 Analysis of Earthquake Tweets (Jan 2020)

The following cell contains the code from `datasets/PR_Earthquake_Tweets_Jan2020/analysis/analyze_Jan202`

[4]:
```python
# File: datasets/PR_Earthquake_Tweets_Jan2020/analysis/analyze_Jan2020_tweets.py
import os
import re
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS

def load_data(filepath):
    try:
        df = pd.read_csv(filepath)
        print("Datos cargados exitosamente.")
        return df
    except Exception as e:
        print(f"Error al cargar el archivo: {e}")
        return None

def preprocess_data(df):
    if 'UTC_Time' in df.columns:
        try:
            df['UTC_Time'] = pd.to_datetime(df['UTC_Time'], errors='coerce')
            print("Columna 'UTC_Time' convertida a datetime.")
        except Exception as e:
            print(f"Error al convertir 'UTC_Time': {e}")
```

```
    else:
        print("La columna 'UTC_Time' no se encontró.")
    if 'Tweet_Content' in df.columns:
        df['tweet_length'] = df['Tweet_Content'].apply(lambda x: len(x) if␣
 ↪isinstance(x, str) else 0)
    else:
        print("La columna 'Tweet_Content' no se encontró para calcular la␣
 ↪longitud del tuit.")
    return df

def plot_interaction_metrics(df):
    interaction_cols = ['Reply_Count', 'Repost_Count', 'Like_Count',␣
 ↪'Bookmark_Count']
    existing_cols = [col for col in interaction_cols if col in df.columns]
    if existing_cols:
        df_interactions = df[existing_cols].melt(var_name="Métrica",␣
 ↪value_name="Conteo")
        plt.figure(figsize=(10, 6))
        sns.boxplot(x="Métrica", y="Conteo", data=df_interactions)
        plt.title("Distribución de Métricas de Interacción")
        plt.tight_layout()
        plt.show()
    else:
        print("No se encontraron columnas de interacción para visualizar.")

def plot_tweet_length_distribution(df):
    if 'tweet_length' in df.columns:
        plt.figure(figsize=(12, 6))
        plt.hist(df['tweet_length'], bins=30, edgecolor='k', alpha=0.7)
        plt.title("Distribución de la Longitud de los Tuits")
        plt.xlabel("Longitud (número de caracteres)")
        plt.ylabel("Frecuencia")
        plt.tight_layout()
        plt.show()

        plt.figure(figsize=(8, 4))
        sns.boxplot(x=df['tweet_length'])
        plt.title("Diagrama de Caja de la Longitud de los Tuits")
        plt.xlabel("Longitud (número de caracteres)")
        plt.tight_layout()
        plt.show()
    else:
        print("La columna 'tweet_length' no está disponible para analizar la␣
 ↪longitud de los tuits.")

def plot_likes_distribution(df):
    if 'Like_Count' in df.columns:
```

```
        plt.figure(figsize=(12, 6))
        plt.hist(df['Like_Count'], bins=30, edgecolor='k', alpha=0.7)
        plt.title("Distribución de Likes en los Tuits")
        plt.xlabel("Número de Likes")
        plt.ylabel("Frecuencia")
        plt.tight_layout()
        plt.show()
    else:
        print("La columna 'Like_Count' no está disponible para analizar la␣
 ↪distribución de likes.")

def plot_length_vs_likes(df):
    if 'tweet_length' in df.columns and 'Like_Count' in df.columns:
        plt.figure(figsize=(10, 6))
        sns.scatterplot(x='tweet_length', y='Like_Count', data=df, alpha=0.7)
        plt.title("Relación entre Longitud del Tuit y Número de Likes")
        plt.xlabel("Longitud del Tuit (caracteres)")
        plt.ylabel("Número de Likes")
        plt.tight_layout()
        plt.show()
    else:
        print("No se encontraron las columnas necesarias ('tweet_length',␣
 ↪'Like_Count') para este análisis.")

def plot_language_distribution(df):
    if 'Language' in df.columns:
        lang_counts = df['Language'].value_counts()
        plt.figure(figsize=(8, 5))
        sns.barplot(x=lang_counts.index, y=lang_counts.values)
        plt.title("Distribución de Idiomas de los Tuits")
        plt.xlabel("Idioma")
        plt.ylabel("Cantidad de Tuits")
        plt.tight_layout()
        plt.show()
    else:
        print("La columna 'Language' no se encontró para visualizar la␣
 ↪distribución de idiomas.")

def generate_word_cloud(df):
    if 'Tweet_Content' not in df.columns:
        print("La columna 'Tweet_Content' no se encontró para generar la nube␣
 ↪de palabras.")
        return
    all_text = " ".join(df['Tweet_Content'].dropna().astype(str))
    cleaned_text = re.sub(r'https?://\S+', '', all_text)
    cleaned_text = re.sub(r'@\w+', '', cleaned_text)
    cleaned_text = re.sub(r'\bRT\b', '', cleaned_text)
```

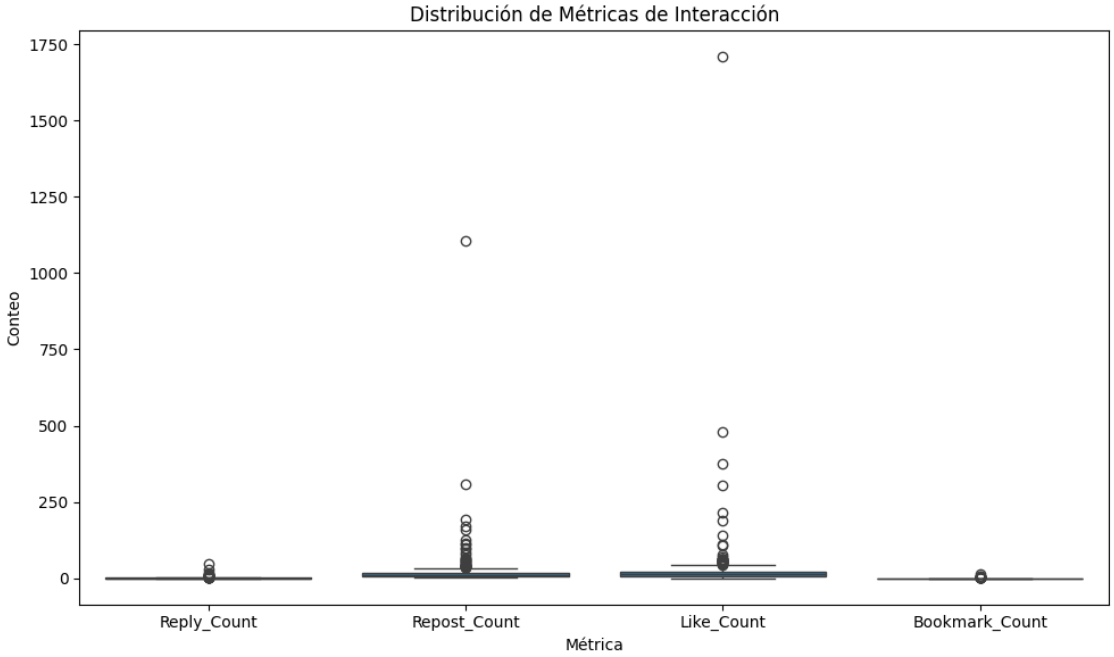
```
    cleaned_text = re.sub(r'[^A-Za-záéíóúñüÁÉÍÓÚÑÜ\s]', '', cleaned_text)
    cleaned_text = cleaned_text.lower()
    spanish_stopwords = {
        "de", "el", "que", "se", "la", "en", "por", "los", "las", "del", "al",
        "un", "una", "con", "para", "este", "esta", "estos", "estas", "ese",
        "esa", "esos", "esas", "y", "o", "u", "pero", "su", "sus", "porque",
        "son", "un", "una", "ser", "sido", "ha", "han", "hay", "qué", "etc"
    }
    custom_stopwords = {"https", "http", "co", "amp"}
    all_stopwords = STOPWORDS.union(spanish_stopwords).union(custom_stopwords)
    wordcloud = WordCloud(
        width=800,
        height=400,
        background_color='white',
        stopwords=all_stopwords
    ).generate(cleaned_text)
    plt.figure(figsize=(12, 6))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.title("Nube de Palabras (Texto Limpio)")
    plt.tight_layout()
    plt.show()

def main():
    filepath =␣
 ↪"datasets\PR_Earthquake_Tweets_Jan2020\PR_Earthquake_Tweets_Jan2020.csv"
    if not os.path.exists(filepath):
        print(f"El archivo '{filepath}' no existe. Verifica la ruta.")
        return
    df = load_data(filepath)
    if df is None:
        return
    print("Información del dataset:")
    print(df.info())
    print(df.head())
    df = preprocess_data(df)
    plot_interaction_metrics(df)
    plot_tweet_length_distribution(df)
    plot_likes_distribution(df)
    plot_length_vs_likes(df)
    plot_language_distribution(df)
    generate_word_cloud(df)

if __name__ == "__main__":
    main()
```

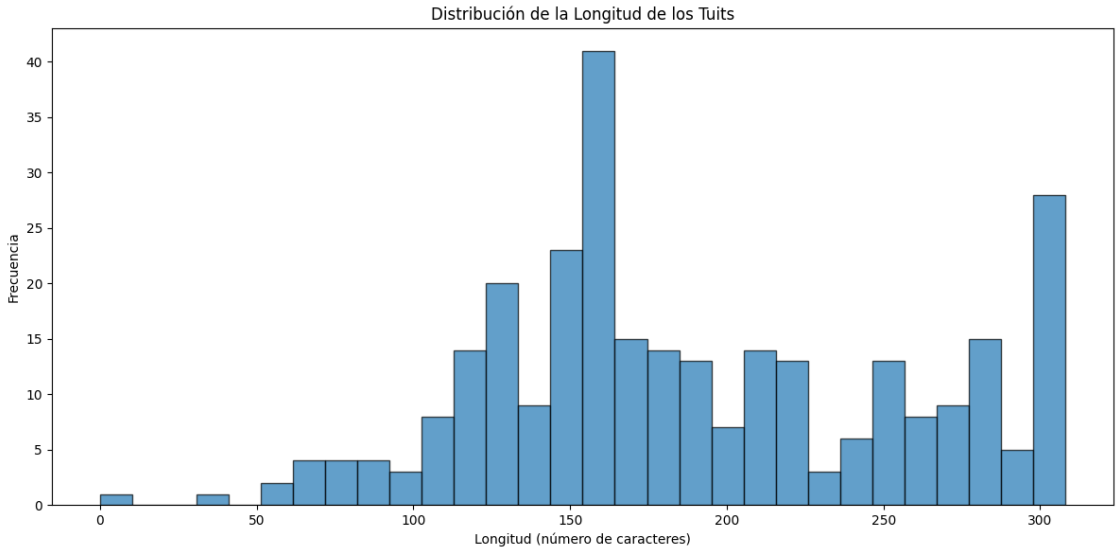
```
Datos cargados exitosamente.
Información del dataset:
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 297 entries, 0 to 296
Data columns (total 50 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   UTC_Time                296 non-null    object
 1   Tweet_Content           296 non-null    object
 2   Post_ID                 296 non-null    float64
 3   Tweet_URL               296 non-null    object
 4   Reply_Count             296 non-null    float64
 5   Repost_Count            296 non-null    float64
 6   Like_Count              296 non-null    float64
 7   View_Count              296 non-null    float64
 8   Bookmark_Count          296 non-null    float64
 9   Tweet_Image_URL         178 non-null    object
 10  Language                296 non-null    object
 11  id                      1 non-null      object
 12  object                  1 non-null      object
 13  result_position         1 non-null      float64
 14  task_id                 1 non-null      object
 15  internal_unique_id      1 non-null      float64
 16  tweet_url               1 non-null      object
 17  original_tweet_url      0 non-null      float64
 18  name                    1 non-null      object
 19  user_id                 1 non-null      float64
 20  username                1 non-null      object
 21  published_at            1 non-null      object
 22  content                 1 non-null      object
 23  views_count             0 non-null      float64
 24  retweet_count           1 non-null      float64
 25  likes                   1 non-null      float64
 26  quote_count             1 non-null      float64
 27  reply_count             1 non-null      float64
 28  bookmarks_count         1 non-null      float64
 29  media_0_thumbnail       1 non-null      object
 30  media_0_type            1 non-null      object
 31  media_0_url             1 non-null      object
 32  media_1_thumbnail       0 non-null      float64
 33  media_1_type            0 non-null      float64
 34  media_1_url             0 non-null      float64
 35  media_2_thumbnail       0 non-null      float64
 36  media_2_type            0 non-null      float64
 37  media_2_url             0 non-null      float64
 38  media_3_thumbnail       0 non-null      float64
 39  media_3_type            0 non-null      float64
 40  media_3_url             0 non-null      float64
 41  binded_media_url        0 non-null      float64
 42  binded_media_domain     0 non-null      float64
```

```
 43  binded_media_thumbnail_url  0 non-null      float64
 44  binded_media_title          0 non-null      float64
 45  binded_media_description    0 non-null      float64
 46  is_retweeted                1 non-null      object
 47  is_quoted                   1 non-null      object
 48  collected_at                1 non-null      object
 49  input_url                   1 non-null      object
dtypes: float64(30), object(20)
memory usage: 116.1+ KB
None
                   UTC_Time  \
0  2019-12-31 23:17:22+00:00
1  2019-12-31 23:14:47+00:00
2  2020-01-02 21:05:59+00:00
3  2020-01-02 20:55:07+00:00
4  2020-01-02 23:58:39+00:00

                                      Tweet_Content       Post_ID  \
0  #TemblorPR En efecto, volvió a temblar en el s…  1.212151e+18
1  #TemblorPR A 4.50 magnitude earthquake has occ…  1.212150e+18
2  23 min.ago #earthquake 4.9 has hit Guayanilla,…  1.212842e+18
3  12 min.ago #earthquake 4.9 has hit Guayanilla,…  1.212840e+18
4  4.5 quake hits Puerto Rico amid rare seismic a…  1.212886e+18

                                          Tweet_URL  Reply_Count  \
0  https://x.com/Motinsitepegas/status/1212150785…          1.0
1  https://x.com/TemblorPR/status/121215013370756…          0.0
2  https://x.com/TemblorPR/status/121284249827353…          0.0
3  https://x.com/TemblorPR/status/121283976054221…          0.0
4  https://x.com/TemblorPR/status/121288595014728…          0.0

   Repost_Count  Like_Count  View_Count  Bookmark_Count  \
0          22.0        13.0         0.0             0.0
1          15.0        15.0         0.0             0.0
2           9.0         7.0         0.0             0.0
3           6.0         4.0         0.0             0.0
4           8.0        11.0         0.0             1.0

                                  Tweet_Image_URL  … media_3_url  \
0  https://pbs.twimg.com/media/ENJstqRXYAMzChh.jpg  …         NaN
1  https://pbs.twimg.com/media/ENJsHTdWoAMtzjq.jpg  …         NaN
2  https://pbs.twimg.com/media/ENThOCIWsAAvDlW.jpg  …         NaN
3  https://pbs.twimg.com/media/ENTfVQmXUAI4Lah.jpg  …         NaN
4                                              NaN  …         NaN

  binded_media_url binded_media_domain  binded_media_thumbnail_url  \
0              NaN                 NaN                         NaN
1              NaN                 NaN                         NaN
```

```
2              NaN                  NaN                          NaN
3              NaN                  NaN                          NaN
4              NaN                  NaN                          NaN

  binded_media_title  binded_media_description is_retweeted  is_quoted  \
0                NaN                       NaN          NaN        NaN
1                NaN                       NaN          NaN        NaN
2                NaN                       NaN          NaN        NaN
3                NaN                       NaN          NaN        NaN
4                NaN                       NaN          NaN        NaN

  collected_at  input_url
0          NaN        NaN
1          NaN        NaN
2          NaN        NaN
3          NaN        NaN
4          NaN        NaN

[5 rows x 50 columns]
Columna 'UTC_Time' convertida a datetime.

<>:134: SyntaxWarning: invalid escape sequence '\P'
<>:134: SyntaxWarning: invalid escape sequence '\P'
C:\Users\Marco\AppData\Local\Temp\ipykernel_28036\3518864080.py:134:
SyntaxWarning: invalid escape sequence '\P'
  filepath =
"datasets\PR_Earthquake_Tweets_Jan2020\PR_Earthquake_Tweets_Jan2020.csv"
```




Distribución de la Longitud de los Tuits
Frecuencia
Longitud (número de caracteres)
0
5
10
15
20
25
30
35
40
0
50
100
150
200
250
300

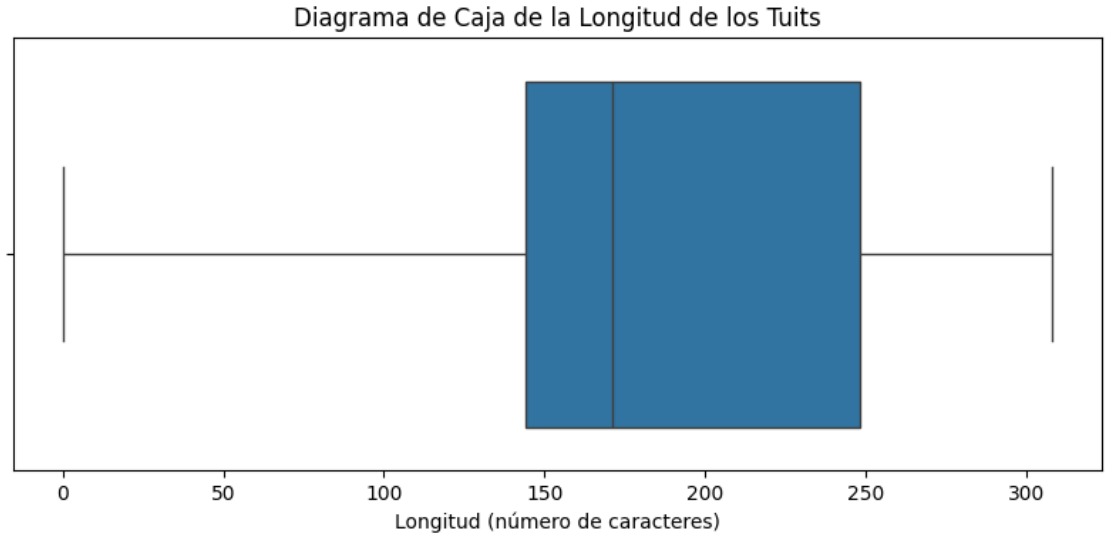
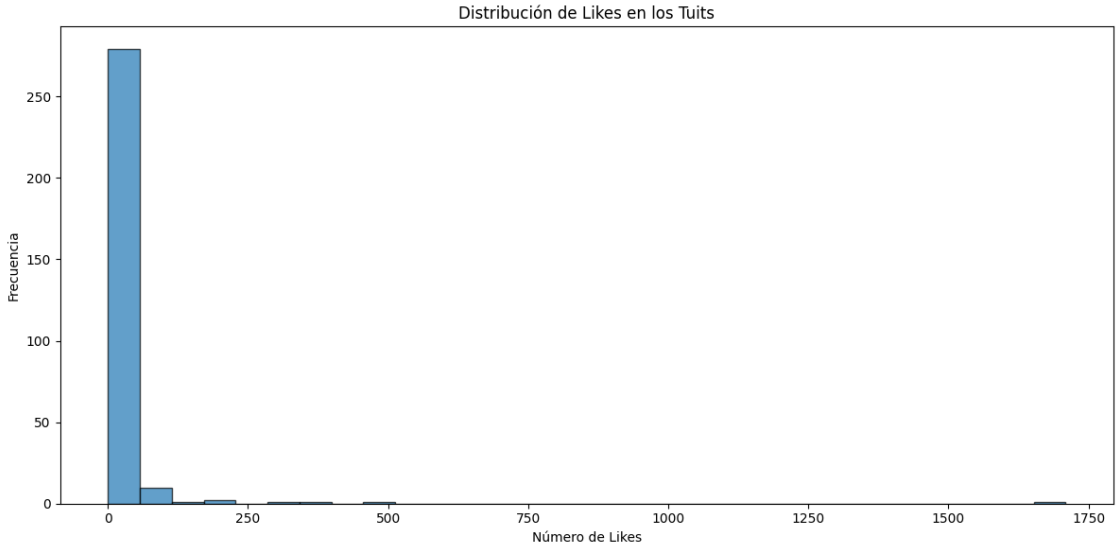
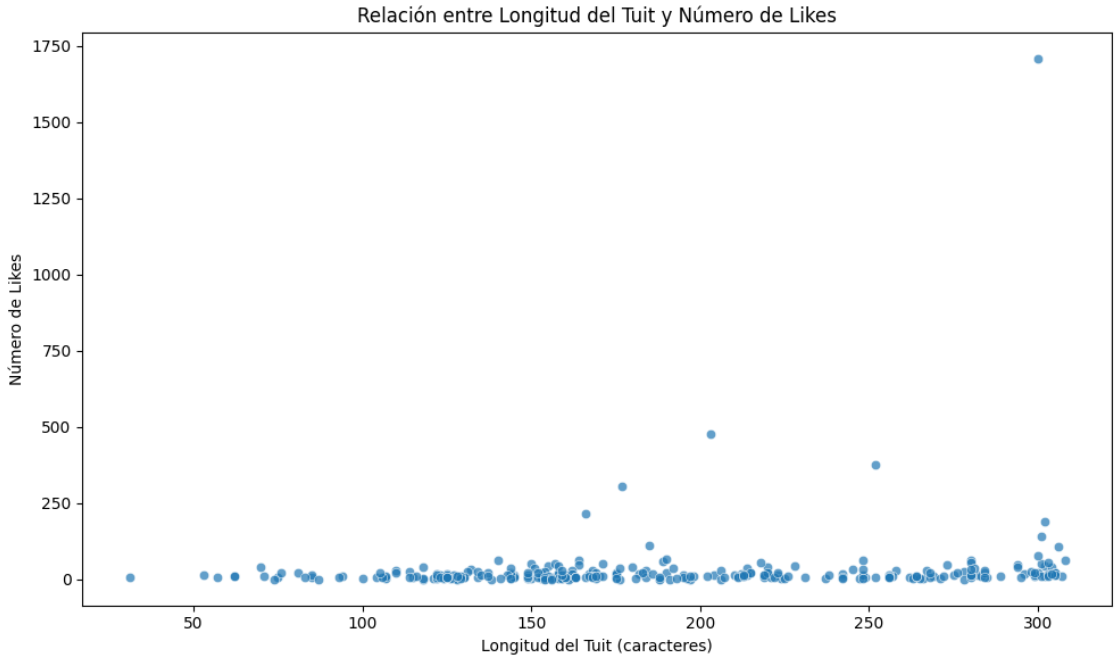
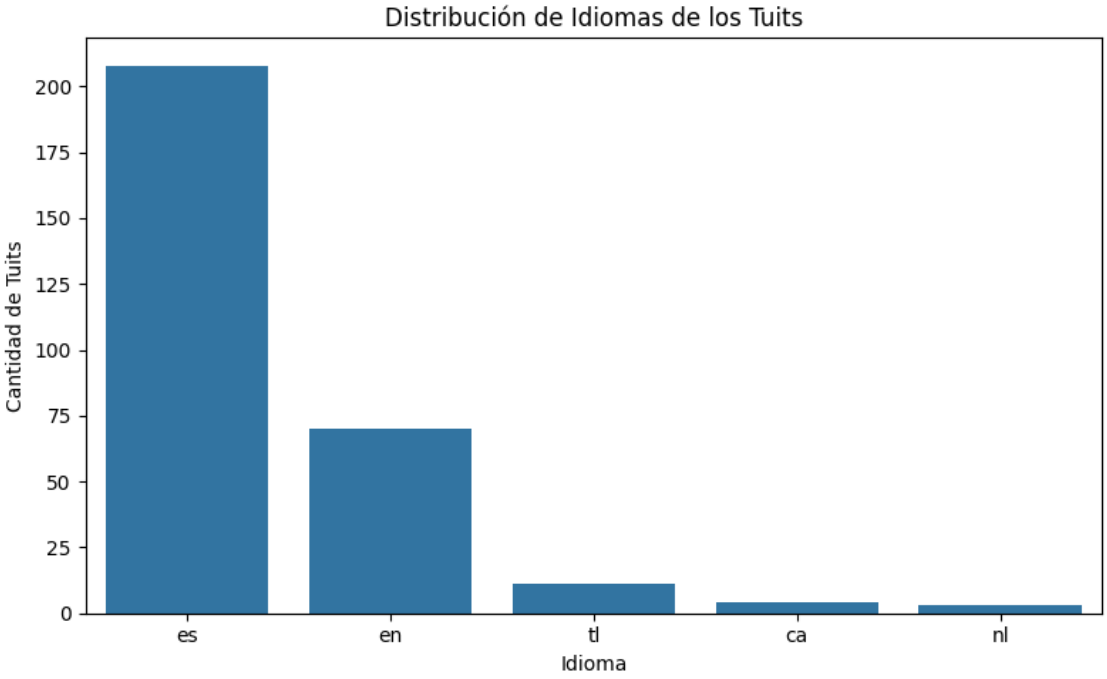
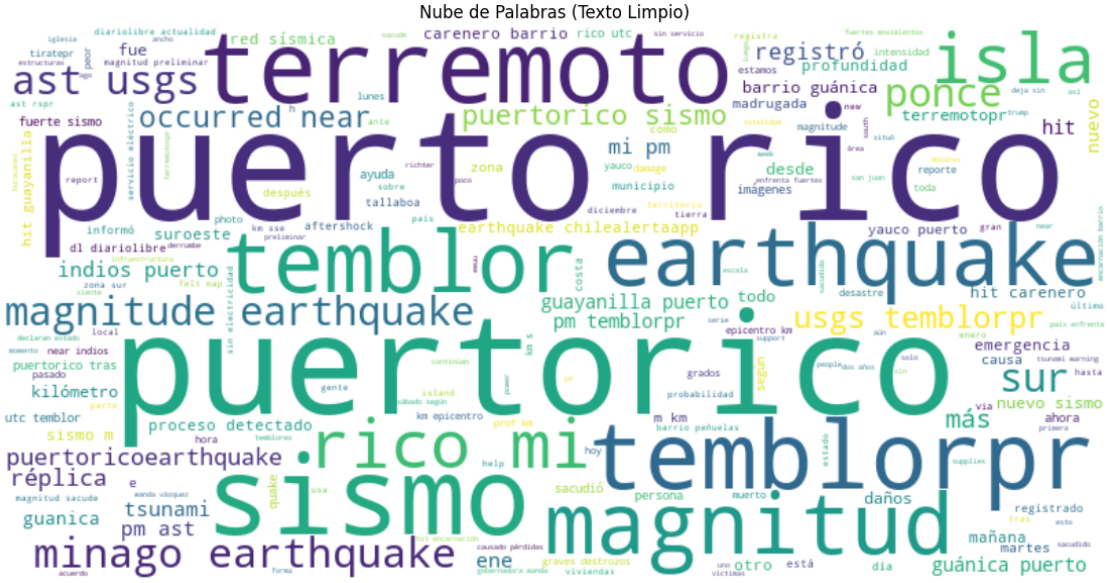


Diagrama de Caja de la Longitud de los Tuits
0
50
100
150
200
250
300
Longitud (número de caracteres)


Distribución de Likes en los Tuits
Frecuencia
Número de Likes
0
50
100
150
200
250
0
250
500
750
1000
1250
1500
1750


Relación entre Longitud del Tuit y Número de Likes
Número de Likes
Longitud del Tuit (caracteres)
0
250
500
750
1000
1250
1500
1750
50
100
150
200
250
300


Distribución de Idiomas de los Tuits
Cantidad de Tuits
Idioma
0
25
50
75
100
125
150
175
200
es
en
tl
ca
nl


Nube de Palabras (Texto Limpio)
terremoto
puerto rico
temblor
earthquake
puertorico
sismo
magnitud
temblorpr
isla
ponce
usgs
ast
occurred near
puertorico sismo
registró
profundidad
barrio guánica
madrugada
terremotopr
hit
nuevo
mi pm
desde
imágenes
municipio
zona
ayuda
después
tallaboa
aftershock
earthquake chilealertaapp
yauco puerto
suroeste
informó
dl diariolibre
indios puerto
magnitude earthquake
guayanilla puerto todo
pm temblorpr
usgs temblorpr
hit carenero
emergencia
causa
sur
más
nuevo sismo
ahora
rico mi
proceso detectado
puertoricoearthquake
réplica
tsunami
pm ast
guanica
minago earthquake
ene
sacudió
persona
muerto
daños
otro
está
día
guánica puerto
mañana
martes
registrado
carenero barrio
red sísmica
fuerte sismo
hit guayanilla
kilómetro
puertorico tras
utc temblor
sismo m
near indios
grados
probabilidad
km epicentro
barrio peñuelas
hoy
persona
graves destrozos
viviendas