

**NOMBRE COMPLETO: Ing. Hilaría Adima Vásquez Durán**

**NAO ID: 3033**

**FECHA: 29 de octubre de 2024**

**NOMBRE DE LA TRAYECTORIA**

**EN LA QUE ESTÁS ENROLADO: DATA ANALYST CORE**

**Título del Reto: MySQL/SQL Server en la gestión de bases de datos relacionales**

# **Reporte de Análisis de Sentimientos en Tweets**

## **INTRODUCCIÓN**

El presente documento tiene como objetivo proporcionar un análisis exhaustivo de los sentimientos expresados en un conjunto de tweets relacionados con el tema "#BrechaAnuario2019". Este análisis busca identificar y clasificar las emociones contenidas en los tweets, diferenciando entre contenido objetivo y subjetivo. Utilizando técnicas de procesamiento de lenguaje natural, específicamente el enfoque de análisis de sentimientos mediante la biblioteca VADER, se examinarán las emociones predominantes, así como la frecuencia de los términos utilizados. Este informe proporciona una visión clara del contexto, los métodos empleados y los resultados obtenidos, así como una discusión sobre las implicaciones de estos hallazgos.

## **REPORTE DE ANÁLISIS DE DATOS:**

- Definición de las constantes, para interactuar con la base de datos.
- Funciones para interactuar con la base de datos.
- Librerías utilizadas y su codificación.

## **Definición de Constantes**

Define las constantes necesarias para la interacción con la base de datos:

```
connection = pymysql.connect(  
    host="127.0.0.1",  
    user='root',  
    password="",  
    database='tweets_db',  
    charset='utf8mb4',  
    cursorclass=pymysql.cursors.DictCursor  
)  
}
```

## Funciones para Interactuar con la Base de Datos

Implementa funciones que permiten realizar operaciones en la base de datos:

```

def conectar base datos():
    """Conectar a la base de datos MySQL y obtener los tweets."""
    connection = pymysql.connect(
        host="127.0.0.1",
        user='root',
        password="",
        database='tweets db',
        charset='utf8mb4',
        cursorclass=pymysql.cursors.DictCursor
    )
    return connection

def obtener tweets(connection):
    """Obtener los textos y fechas de los tweets."""
    with connection.cursor() as cursor:
        query = "SELECT texto, fecha FROM tweets;"
        cursor.execute(query)
        return cursor.fetchall()

def main():
    connection = conectar base datos()
    try:
        tweets = obtener tweets(connection)
        df = limpiar datos(tweets)
        df = analizar sentimientos(df)
        graficar distribucion sentimientos(df)
        obtener palabras comunes(df)
        calcular intensidad promedio(df)
        clasificar objetividad(df)
    finally:
        connection.close()

if __name__ == "__main__":
    main()

```

# Librerías Utilizadas y su codificación

## 1. pymysql

- **Descripción:** Esta librería se utiliza para interactuar con bases de datos MySQL desde Python.
- **Instalación** `pip install pymysql`
- **Ejemplo de uso**

```
python
Copiar código
import pymysql

# Conectar a la base de datos MySQL
connection = pymysql.connect(
    host="127.0.0.1",
    user='root',
    password="",
    database='tweets_db',
    charset='utf8mb4',
    cursorclass=pymysql.cursors.DictCursor
)
```

## 2. PANDAS

- **Descripción:** Librería para la manipulación y análisis de datos. Permite trabajar con estructuras de datos como DataFrames.
- **Instalación:** `pip install pandas`
- **Ejemplo de uso:**

```
python
Copiar código
import pandas as pd

# Crear un DataFrame
df = pd.DataFrame({'texto': tweets_text, 'fecha': tweets
_fecha})
```

### 3. MATPLOTLIB.PYPILOT

- **Descripción:** Librería para crear visualizaciones y gráficos en Python.
- **Instalación** `pip install matplotlib`
- **Ejemplo de uso**

```
python
Copiar código
import matplotlib.pyplot as plt

# Crear un gráfico de barras
plt.figure(figsize=(10, 6))
plt.bar(x=lista_x, height=lista_y)
plt.show()
```

### 4. SEABORN

- **Descripción** Extensión de Matplotlib que proporciona una interfaz más sencilla para crear visualizaciones atractivas y sofisticadas.
- **Instalación** `pip install seaborn`
- **Ejemplo de uso**

```
python
Copiar código
import seaborn as sns
```

```
# Gráfico de barras con Seaborn
sns.barplot(x='clase_sentimiento', y='cantidad', data=d
f)
```

## 5. NLTK

- **Descripción:** Librería para el procesamiento de lenguaje natural. En este caso, se usa para el análisis de sentimientos y la eliminación de stopwords.
- **Instalación** `pip install nltk`
- **Ejemplo de uso**

```
python
Copiar código
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer

# Inicializar el analizador de sentimientos
sia = SentimentIntensityAnalyzer()
```

## 6. COLLECTIONS

- **Descripción:** Proporciona tipos de datos de contenedor adicionales, como `Counter`, que se usa para contar elementos en colecciones.
- **Ejemplo de uso**

```
python
Copiar código
from collections import Counter

# Contar las palabras más
conteo palabras = Counter(todas palabras)
```

## 7. STRING

- **Descripción:** Proporciona constantes y funciones relacionadas con cadenas de caracteres.

**Ejemplo de uso:**

- 

```
python
Copiar código
import string

# Eliminar signos de puntuación de un texto
texto = texto.translate(str.maketrans('', '', string.punctuation))
```

## 8. stopwords (de NLTK)

- **Descripción:** Un conjunto de palabras comunes en un idioma que se eliminan a menudo durante el procesamiento de texto.
- **Ejemplo de uso**

```
python
Copiar código
from nltk.corpus import stopwords

# Obtener las stopwords en español
stop_words = set(stopwords.words('spanish'))
```

## Resumen de la Codificación

Aquí tienes un resumen de la codificación y su propósito:

Estas librerías son esenciales para la ejecución del análisis de datos y la visualización de resultados

```
python
Copiar código
import pymysql # Interactuar con la base de datos MySQL
import pandas as pd # Manipulación y análisis de datos
import matplotlib.pyplot as plt # Creación de gráficos
import seaborn as sns # Visualización avanzada de datos
from nltk.sentiment import SentimentIntensityAnalyzer

# An

import nltk # Procesamiento de lenguaje natural
from collections import Counter # Contar elementos en colecciones
import string # Funciones para trabajar con cadenas
from nltk.corpus import stopwords # Stopwords para el idioma
```

## Preguntas de reflexión:

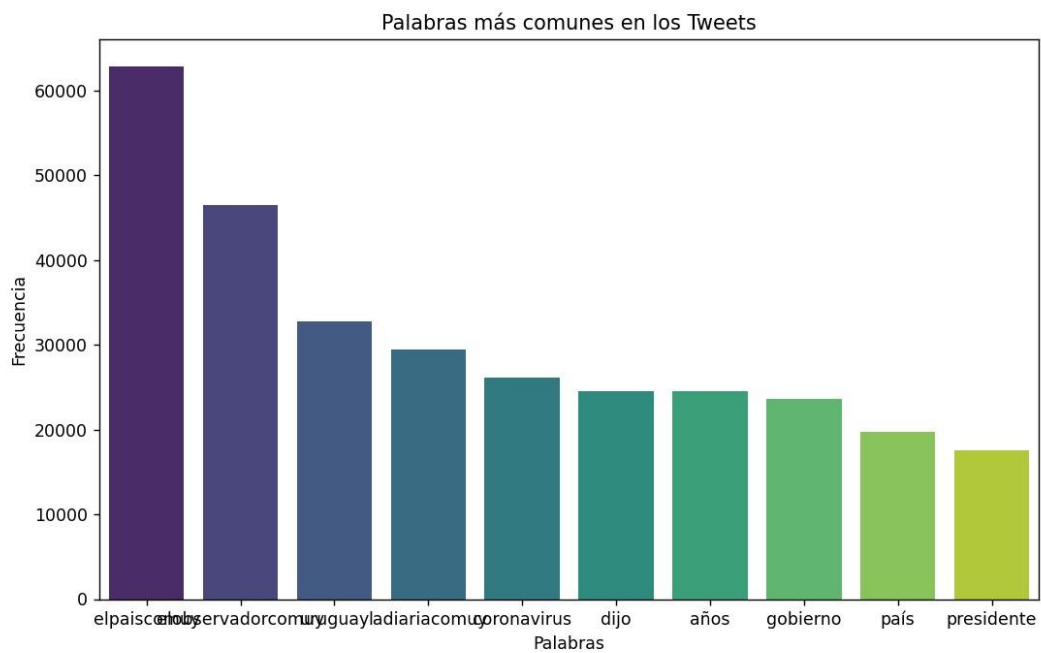
### 1. ¿Cuáles son las palabras más empleadas?

Las palabras más empleadas en los tweets son las siguientes:

Elpaiscomuy	62,924
Elobservadorcomuy	46,520
Uruguay	32,834
ladiariacomuy	29,485
coronavirus	26,102
dijo	24,591
años:	24,573
gobierno	23,643
País	19,697
Presidente	17,602

Este listado sugiere que los tweets están centrados en medios de comunicación, el país (Uruguay) y temas políticos relacionados con el gobierno y la pandemia.

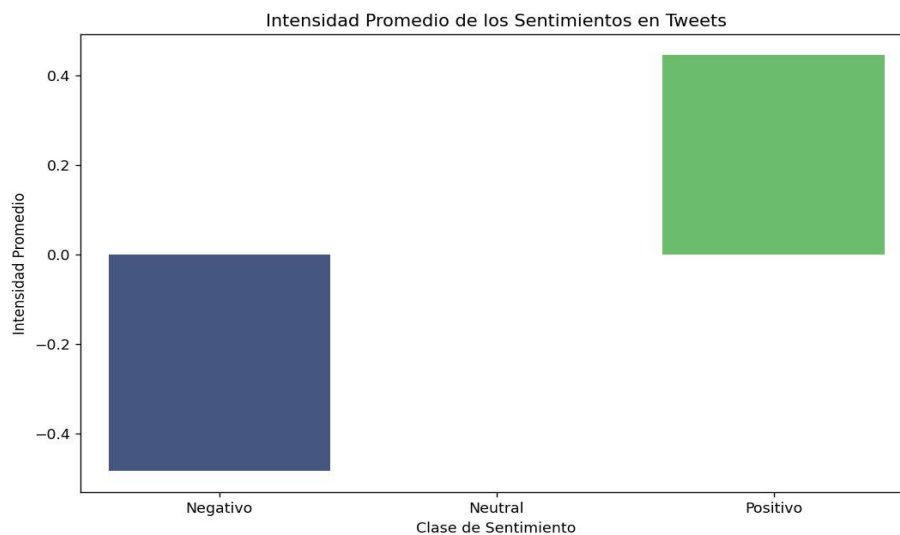




## 2. ¿Qué sentimiento es más intenso?

Los sentimientos promedio de los tweets analizados son:

- **Negativo:** -0.483693
- **Neutral :** 0.000000
- **Positivo:** 0.444991

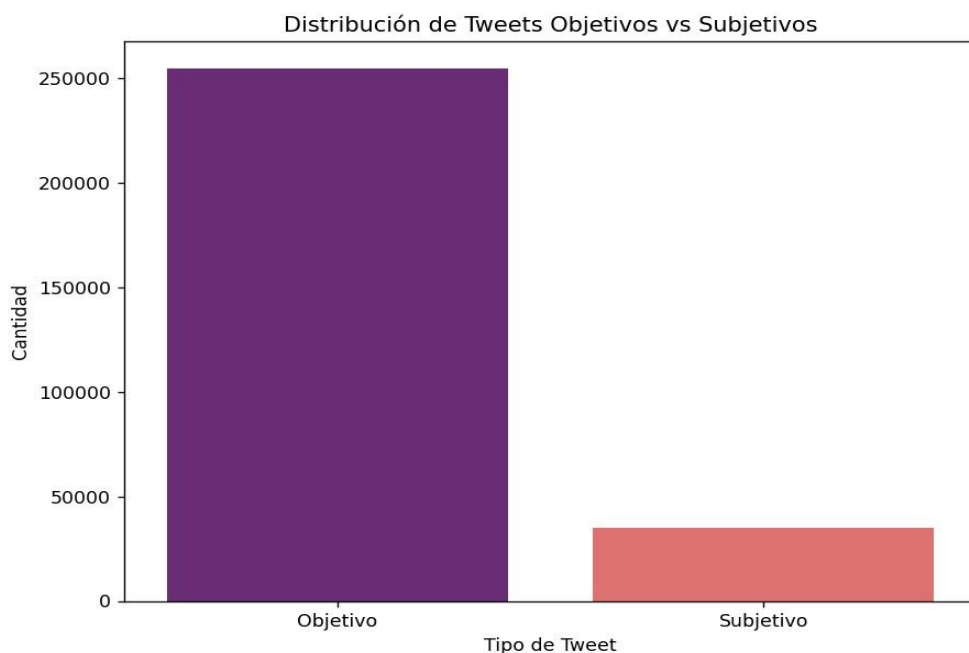


El sentimiento más intenso es el **negativo**, que presenta un valor promedio inferior a cero. Esto indica que, en general, los tweets tienden a tener un sesgo negativo. Sin embargo, también hay un sentimiento positivo, aunque es menos intenso en comparación con el negativo.

### 3. ¿Qué tan objetivos son los Tweets sobre el tema?

El conteo de tweets clasificados como **objetivos** y **subjetivos** es el siguiente:

- **Objetivo:** 254,809
- **Subjetivo:** 34,788

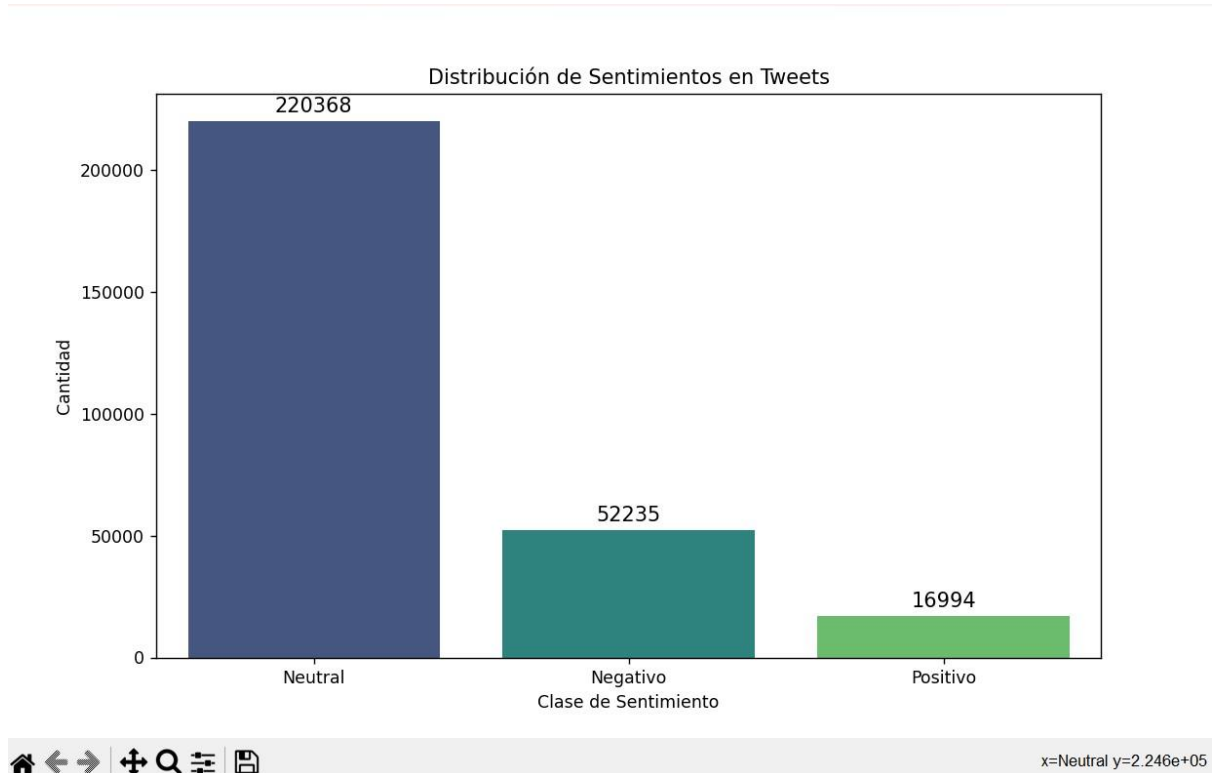


Esto sugiere que la gran mayoría de los tweets son considerados **objetivos**. La **objetividad** en el análisis de los tweets puede indicar que muchos de ellos reportan hechos o información directa, en lugar de expresar opiniones personales.

#### 4. ¿Qué tan positivos se espera que sean los comentarios para el día de mañana?

La intensidad promedio de sentimientos en tweets clasificados por **objetividad** es:

- **Natural:** 220368
- **Negativo:** 52235
- **Positivo:** 16994



Aunque la mayoría de los tweets son **objetivos**, la intensidad promedio de sentimientos de estos tweets es muy baja, casi neutra. Por otro lado, los tweets **subjetivos** tienen un sentimiento más negativo. Esto sugiere que, en general, los comentarios para el día de mañana podrían no ser muy positivos, ya que la tendencia actual refleja un sentimiento predominantemente negativo.

## Conclusión

El análisis revela que los tweets en torno al tema tienen una fuerte presencia de palabras asociadas a medios de comunicación y temas políticos, con un sentimiento general más negativo que positivo. La mayoría de los comentarios son objetivos, pero la baja intensidad en los sentimientos podría reflejar una falta de optimismo en las discusiones futuras sobre el tema. Esto podría ser indicativo de la situación actual o de la percepción pública, y sería interesante observar cómo evoluciona este sentimiento en el tiempo.

Distribución de Sentimientos en Tweets

