

```
In [1]: import pandas as pd
import numpy as np
import re

import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
stopwords = stopwords.words('spanish')

from wordcloud import WordCloud, ImageColorGenerator
import matplotlib.pyplot as plt
import PIL.Image
from PIL import Image
from IPython.display import display
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
In [2]: # Ruta de archivo
post = pd.read_excel("proposiciones.xlsx")

# Muestra las primeras filas del DataFrame
post.head()
```

Out[2]:

	tema
0	Citacion a control politico - Aumento tarifari...
1	Proposición Relacionada con presupuesto.
2	Debate de control Político - problemas estruct...
3	Debate de control político relacionado con las...
4	Debate de control policitco relacionado con al...

```
In [73]: post.tema
palabra = post.tema.str.cat(sep=" ")
stopwords.extend(["solicitud", "petición", "información", "cuáles", "cuánto", "informe", "indique", "fecha",
"ministerio",

"entidad", "traslado", "formato", "excel", "existen", "existe", "siguiente", "vigente", "pregunta", "preguntas",
",

"vigencia", "cargo", "indicar", "informar", "sírvas", "cómo", "realizado", "remitir", "colombia", "ministro",

"respuesta", "respecto", "solicito", "incluyendo", "frente", "número", "remitido", "indicando", "presidente",

"indicando", "copia", "dentro", "solicita", "detallar", "presente", "cuales", "actualmente", "congreso",

"preguntas", "colombiano", "actual", "documento", "cuántos", "cuestionario", "informacion", "comisión",

"último", "correspondiente", "últimos", "manera", "encuentran", "derecho", "petición", "radicado", "favor",

"peticion", "urgente", "proposición", "citación", "debate", "control", "político", "citación", "citar",

"citacion", "medida", "politico", "explique", "parte", "gobierno", "nacional", "proposicion", "primera", "segunda",

"tercera", "cuarta", "quinta", "sexta", "senado", "cámara", "representantes", "relacionada", "aditiva",
    "legislatura", "sesión", "aplazamiento", "realizará", "audiencia pública",
"realizarse" "ubicada",
    "dirección", "miércoles", "formal", "aplazamiento", "abril", "agosto", "diciembre"
])
```

In [74]:

```
def pre_procesado(texto):
    if isinstance(texto, float): # Handle potential NaN values
        return ""
    texto = texto.lower()
    texto = re.sub(r"[\W\']+", " ", texto)
    texto = [palabra for palabra in texto.split() if palabra not in stopwords and len(palabra) >= 5]
    texto = " ".join(texto)
    return texto

post['pp'] = post['tema'].apply(lambda texto: pre_procesado(texto))
post
```

Out[74]:

	tema	pp	cluster	nombres_clusters
0	Citacion a control politico - Aumento tarifari...	aumento tarifario rgion caribe	3	Ejecución presupuestal
1	Proposición Relacionada con presupuesto.	presupuesto	5	Matriz energética
2	Debate de control Político - problemas estruct...	problemas estructurales funcionales servicio e...	3	Energía
3	Debate de control político relacionado con las...	relacionado irregularidades presentadas median...	5	Ejecución presupuestal
4	Debate de control policitco relacionado con al...	policitco relacionado alsas tarifas energia el...	2	Subsidios y tarifas
...	...	...	...	...
273	Proposición No. 036 Legislatura 2022 – 2023		5	Ejecución presupuestal
274	Traslado por Competencia proposición 21 numera...	competencia numeral aeronáutica civil	5	Ejecución presupuestal
275	Traslado de Min Hacienda al DNP y este al MME ...	hacienda declaratoria conmoción interior febre...	1	Ejecución presupuestal
276	El H.R Diego Caicedo, representante por el dep...	diego caicedo representante departamento cundi...	2	Energía
277	Traslado de DNP a traslado de Min Hacienda sol...	hacienda solicitando camara impacto generado r...	5	Ejecución presupuestal

278 rows × 4 columns

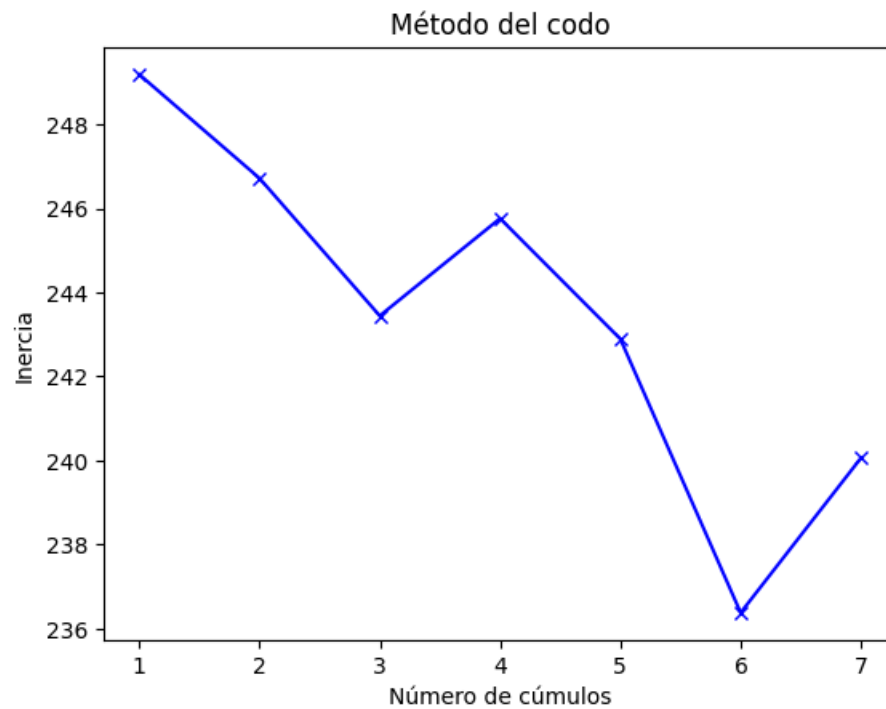
[illegible]

```
from sklearn.neighbors import NearestNeighbors
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
```

```
Out[75]: [(1, 249.18799528880268),
(2, 246.71774963951415),
(3, 243.44307183817355),
(4, 245.7592849762495),
(5, 242.89164045250595),
(6, 236.3783461519135),
(7, 240.06056061148516)]
```

```
In [76]: # Método del codo
plt.xlabel("Número de cúmulos")
plt.ylabel("Inercia")
plt.title("Método del codo")
plt.plot(ks, k_inertias, 'bx-')
```

Out[76]: [<matplotlib.lines.Line2D at 0x7c77efc1b650>]



```
In [77]: #Entrenamiento del modelo
k = 6
model = KMeans(n_clusters=k)
model.fit(tfidf_matrix)
```

Out[77]:

KMeans ⓘ ?

KMeans(n_clusters=6)

```
In [78]: post['cluster'] = model.labels_
```

```

In [79]: #los vecinos más cercanos al centroide
#entrenamiento con matriz tfidf
nbrs = NearestNeighbors(n_neighbors=3, metric='euclidean').fit(tfidf_matrix.values)

#porcentaje de los datos de cada clase
clust_cnt = post['cluster'].value_counts()
clust_cnt_pct = post['cluster'].value_counts(normalize=True)

centroids = model.cluster_centers_

terms = tfidf_vect.get_feature_names_out()
order_centroids = centroids.argsort()[::-1]

for i in range(k):
    print(f"CLUSTER #{i}")
    print(f"Tiene {clust_cnt[i]} registros ({clust_cnt_pct[i]:.2%} de los datos)")

    print("Palabras más importantes: ")
    for ind in order_centroids[i][5]:
        print(terms[ind], end=" ")
    print()

    print(f"Comentarios más Representativos: ")
    for vecino in post.iloc[nbrs.kneighbors([centroids[i]])[1][0]].pp.values:
        print(f" {vecino}")

    print("*****")
    print()

```

```

CLUSTER #0
Tiene 24 registros (8.63% de los datos)
Palabras más importantes:
ecopetrol febrero irregularidades perdida administracion
Comentarios más Representativos:
*
*
*
*****

```

```

CLUSTER #1
Tiene 97 registros (34.89% de los datos)
Palabras más importantes:
energia electrica competencia energetica gasolina
Comentarios más Representativos:
*
*
*
*****

```

```

CLUSTER #2
Tiene 31 registros (11.15% de los datos)
Palabras más importantes:
energía tarifas comercialización servicio industria
Comentarios más Representativos:
*
*
*
*****

```

```

CLUSTER #3
Tiene 33 registros (11.87% de los datos)
Palabras más importantes:
miércoles ejecución presupuestal formal aplazamiento
Comentarios más Representativos:
*
*

```

```

*
*****

CLUSTER #4
Tiene 19 registros (6.83% de los datos)
Palabras más importantes:
agosto combustibles aumento martes energéticas
Comentarios más Representativos:
*
*
*
*****

```

```

CLUSTER #5
Tiene 74 registros (26.62% de los datos)
Palabras más importantes:
energía energética proyectos transición adelantado
Comentarios más Representativos:
*
*
*
*****

```

```

In [80]: clusters = {0: 'Ecopetrol',
                    1: 'Energía',
                    2: 'Subsidios y tarifas',
                    3: 'Ejecución presupuestal',
                    4: 'Combustibles',
                    5: 'Seguimiento a planeación'
                    }

post['nombres_clusters'] = post['cluster'].apply(lambda val: clusters[val])
post.sample(5)

```

Out[80]:

	tema	pp	cluster	nombres_clusters
151	Debate de Control Político, cuestionario de 16...	responda situación pagos subsidios energía cam...	2	Subsidios y tarifas
16	Citación a debate de control político, posició...	posición minero cauca nordeste antioqueño	1	Energía
108	Debate de Control Político: Cuestionario de 16...	minería artesanal manual mecanizada implementa...	0	Ecopetrol
11	Proposición No. 031 aditiva a la Proposición N...	proyecto hidroeléctrico quimbo	3	Ejecución presupuestal
256	Proposición 05		1	Energía

```

In [81]: from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.cluster import DBSCAN
        from sklearn.preprocessing import StandardScaler

```

```

In [82]: from sklearn.model_selection import train_test_split

X = post['pp']
y = post['cluster'] # Se cambió 'label' a 'cluster'

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```
In [83]: import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.countplot(data=post, x='nombres_clusters', hue='nombres_clusters', palette='viridis',
legend=False)
plt.xlabel('Temas')
plt.ylabel('Solicitudes')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

