

# Procesamiento de corpus y visualización de características

## ✓ Instalación de librerías

En esta sección se detalla qué librerías se deben instalar para hacer el procesamiento de un corpus:

- **PANDAS:** Análisis de datos.
- **NUMPY:** Creación de vectores y matrices. Procesamiento matemático.
- **SEABORN:** Librería para visualización de datos.
- **MATPLOTLIB** Librería para visualización de datos.
- **WORDCLOUD** Librería para crear nubes de palabras.
- **PILLOW:** Librería para edición de imágenes.

**Nota:** en este entorno se pueden insertar comentarios, los cuales no afectan la ejecución del código. Para insertarlos, es necesario introducirlos con la tecla #.

```
!pip install LIBRARY
# este comando se usa (!pip) para instalar librerías de Python
```

```
!pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.1.4)
Requirement already satisfied: numpy<2, >=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1
```

```
!pip install numpy
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.26.4)
```

```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.13.1)
Requirement already satisfied: numpy!=1.24.0, >=1.20 in /usr/local/lib/python3.10/dist-packages (from seaborn)
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.10/dist-packages (from seaborn) (2.1.4)
Requirement already satisfied: matplotlib!=3.6.1, >=3.4 in /usr/local/lib/python3.10/dist-packages (from seaborn)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.2->seaborn)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->seaborn)
```

```
!pip install matplotlib
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.0.1)
```

```
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.26.)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (10.)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.7)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7) (1.16.0)
```

```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in /usr/local/lib/python3.10/dist-packages (1.9.3)
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.10/dist-packages (from wordcloud) (1.26.)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from wordcloud) (10.4.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from wordcloud) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (1.0.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (0.12)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (4.22.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (1.0.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (20.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud) (2.7.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7) (1.16.0)
```

```
!pip install pillow
```

```
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (10.4.0)
```

## ✓ Importar librerías

Las librerías que se acaban de instalar ya están disponibles para hacer uso de ellas, solo que antes de emplearlas, hay que indicar en el código que se utilizarán. Para ello, se usa el comando:

- **import** LIBRARY.

```
import pandas as pd #es común que se usen abreviaturas para hacer más ligero el código. En este caso, se especifica
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
```

El proceso para importar puede especificarse para no cargar toda la librería, sino solo una parte. Por ejemplo, asumimos que la librería es *Office*, pero de ella solo queremos usar un programa como *Word* y no el resto de programas y plugins de Office. La forma de hacerlo es la siguiente:

- **from** LIBRARY **import** PACKAGE
- **from** OFFICE **import** WORD

```
from wordcloud import WordCloud
from PIL import Image
```

## ✓ Cargar corpus

Es importante saber la ruta del archivo que tiene el corpus.

Lo recomendable es que se suba a una ruta del entorno virtual para acelerar el procesamiento.

Para este taller usaremos algunos de los corpus que están disponibles en mi página de [Github](#).

```
ls #comando para listar la información que está en una carpeta
```

```
data.csv  hateSpeech.csv  language.png  nube.jpg  sample_data/
```

## ✓ Indexar el corpus en un DataFrame



Para poder explorar las características del corpus, primero hay que indexarlo en una variable. Esta variable en PANDAS tiene la estructura de un dataframe, en el cual la información del corpus está organizada en filas y columnas.

```
corpus = pd.read_csv('hateSpeech.csv', encoding='utf-8')
```

## ✓ Exploración del corpus

Se usarán algunos comandos para tener información sobre el corpus. La mayoría de estos comandos son funciones que se activan al crear variables.

```
corpus #primeras y últimas líneas del corpus
```

	Text	Class	Language	
0	Obvio la policía es una pendeja, pero ya vi qu...	Violencia	E	
1	Quemenla en leña verde junto con la poli asesina.	Violencia	E	
2	A este hombre lo mataron por ser transexual.	Violencia	I	
3	Y después se quejan de que se muere tanta pers...	Violencia	I	
4	Ojala y quedé como vegetal el hdp para que ya ...	Violencia	I	
...	...	...	...	
772	Alch el Carlos Trejo si le hubiera partido su ...	Violencia	E	
773	A puerco cabron, ya que se entere el mamarrach...	Violencia	E	
774	Team Carlos Trejo, partele su puta madre a Alf...	Violencia	E	
775	¡Ya Carlos Trejo ya dale una putiza a Alfredo ...	Violencia	E	
776	Ojalá Carlos Trejo se ponga una chinga a Alfre...	Violencia	E	

Next steps:

[Generate code with corpus](#)
[View recommended plots](#)
[New interactive sheet](#)

```
corpus.head(10)
```

	Text	Class	Language	
0	Obvio la policía es una pendeja, pero ya vi qu...	Violencia	E	
1	Quemenla en leña verde junto con la poli asesina.	Violencia	E	
2	A este hombre lo mataron por ser trangénero.	Violencia	I	
3	Y después se quejan de que se muere tanta pers...	Violencia	I	
4	Ojala y quedé como vegetal el hdp para que ya ...	Violencia	I	
5	Ojalá le haga reacción y se lo cargue su chdm !!	Violencia	I	
6	A mí en lo personal me vale madre si se vacuna...	Violencia	E	
7	El que no debe existir es este Marrano violado...	Violencia	E	
8	Dónde te agarren HDTPM unos plomazos cabron pa...	Violencia	E	

Next steps:

[Generate code with corpus](#)

[View recommended plots](#)

[New interactive sheet](#)

corpus.tail(8)

	Text	Class	Language	
769	Cuánta razón tenía el caza fantasmas Carlos Tr...	Violencia	E	
770	Ni a uno ni a otro, par de impresentables.	Violencia	E	
771	No es por nada pero ahora sí quiero que Carlos...	Violencia	E	
772	Alch el Carlos Trejo si le hubiera partido su ...	Violencia	E	
773	A puerco cabron, ya que se entere el mamarrach...	Violencia	E	
774	Team Carlos Trejo, partele su puta madre a Alf...	Violencia	E	
775	¡Ya Carlos Trejo ya dale una putiza a Alfredo ...	Violencia	E	

corpus.Class.value\_counts()

	count
Class	
Violencia	520
Acoso	257

corpus.Language.value\_counts()

	count
Language	
E	552
I	225

corpus.info()


<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 777 entries, 0 to 776

```
Data columns (total 3 columns):
```


#	Column	Non-Null Count	Dtype
0	Text	777 non-null	object
1	Class	777 non-null	object
2	Language	777 non-null	object

```
dtypes: object(3)  
memory usage: 18.3+ KB
```

```
corpus.dtypes
```



	0
<b>Text</b>	object
<b>Class</b>	object
<b>Language</b>	object





```
corpus.Text.size
```

 777

```
#corpus.Text.unique()
```

```
corpus.sample(30)
```



	Text	Class	Language	
5	Ojalá le haga reacción y se lo cargue su chdm !!	Violencia	I	
228	Qué pateticas se ven quejándose de lo mierda q...	Violencia	E	
325	Comparto, si tienes estómago para bichar algún...	Violencia	E	
643	Jjaajjaa pinche vieja tripona	Violencia	E	
420	Pero con el detente y con no mentir, no robar ...	Acoso	E	
12	Menos mal gente con d4wn como tu mu3r3n a los ...	Violencia	I	
649	Si la tripona esa tiene pareja, cualquiera tie...	Violencia	E	
624	Jajajajajajajajajaja, hasta para ser pendejos ...	Acoso	E	
736	Así de cortita como su riatita	Violencia	I	
387	Y El covidiota mayor dando "el ejemplo"	Violencia	E	
327	Freddy, si no eres católico, has tu vida de pa...	Violencia	E	
339	Si a ti te inyectaron aire, a este Chairo ni s...	Violencia	E	
611	Que horror!!! Que gente tan corriente y vulgar...	Acoso	E	
234	odio a las hombres no quiero verles el pito pu...	Violencia	E	
165	Andaban con hambre los asquerosos policías mex...	Violencia	E	
394	Pero con el detente y con no mentir, no robar ...	Violencia	E	
20	Y si mejor te violan?	Violencia	E	
180	No, fue asesinada solo por un policía, que por...	Violencia	E	
335	Que odio y ganas de cagar a trompadas me provo...	Violencia	E	
658	Vuelta la que te dio el rey de Tepito aquella ...	Violencia	I	
456	Sos la Carrió de la izquierda.	Violencia	I	
287	Jajajaja i know , no ves el doblaje y ya. Bien...	Violencia	E	
397	Pinches mierdas q se van d vacaciones	Violencia	E	
205	No existe el término "poca mujer" por lo que s...	Acoso	E	
748	Tu no eres politiquillo eres actor y como acto...	Violencia	I	
178	Jamas entenderan. Que deben perseguir crimanal...	Violencia	E	
566	un individuo que no se ha leído dos libros com...	Violencia	I	
704	No eres más que un pobre e insignificante matr...	Acoso	E	
773	A puerco cabron, ya que se entere el mamarrach...	Violencia	E	

## Referencias

Si quieres explorar más **atributos** y **métodos**, puedes consultar la documentación de PANDAS en este [link](#).


## ✓ Graficación de los datos

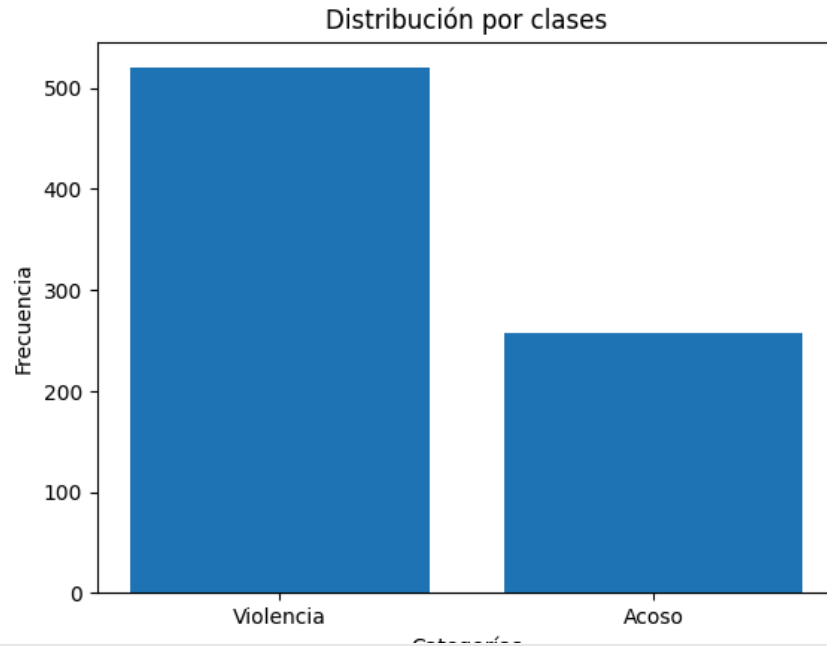
#Se graficará la información de Class con base en su frecuencia

```
labels = ['Violencia', 'Acoso'] #etiquetas en eje X
counts = [520, 257] #valores en eje Y
ypos = np.arange(len(labels)) #vectorización
```


ypos

```
plt.xticks(ypos, labels)
plt.xlabel("Categorías")
plt.ylabel("Frecuencia")
plt.title("Distribución por clases")
plt.bar(ypos, counts)
```

 <BarContainer object of 2 artists>



```
corpus.Language.value_counts()
```

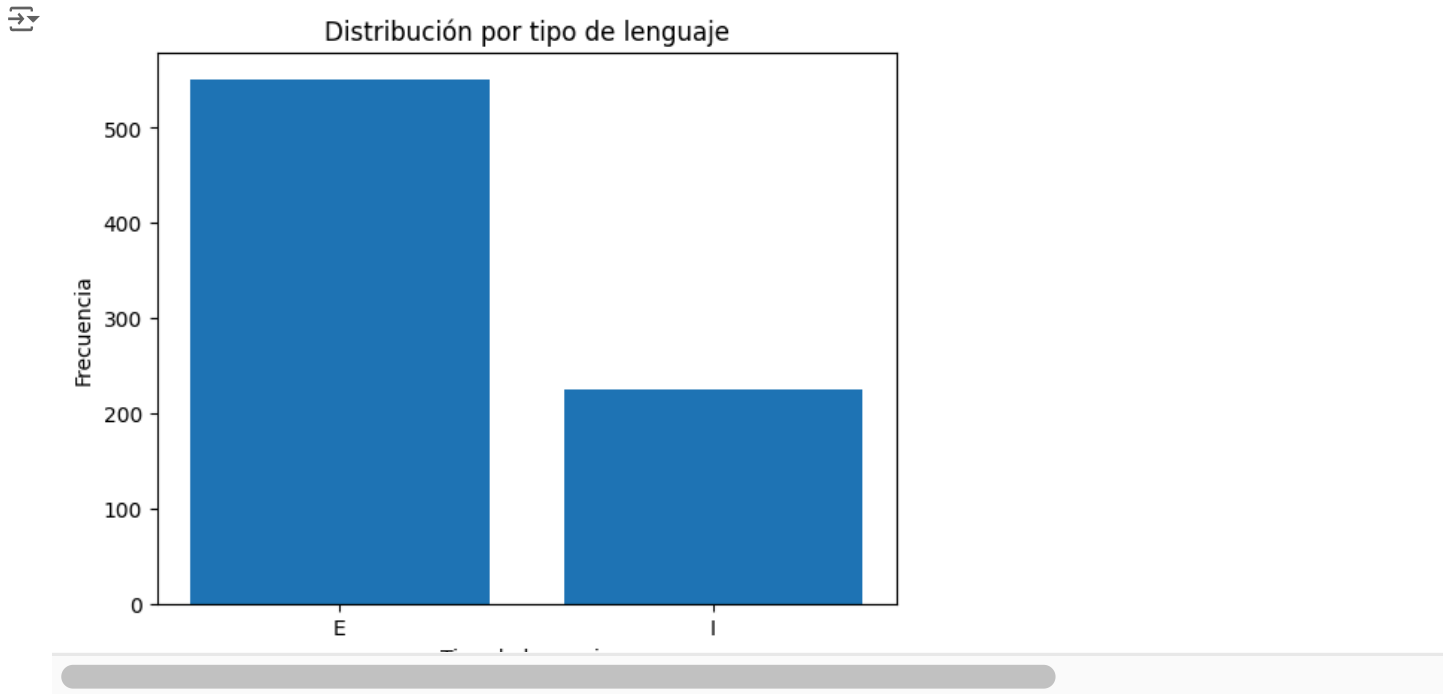
 **count**

Language	
E	552
I	225

#Se graficará la información de Language con base en su frecuencia

```
labels = ['E', 'I']
counts = [552, 225]
ypos = np.arange(len(labels))
ypos
```

```
plt.xticks(ypos, labels)
plt.xlabel("Tipo de lenguaje")
plt.ylabel("Frecuencia")
plt.title("Distribución por tipo de lenguaje")
plt.bar(ypos, counts)
plt.savefig('language.png')          #con este comando guardamos el gráfico en el path
```



## ✓ Visualización por nube de palabras

#Preparación de los datos por categoría. Se dividen con base en la categoría.

```
cloud1 = corpus[corpus.Class=='Violencia']
cloud2 = corpus[corpus.Class=='Acoso']
```

#Generación de listas por categoría. Se toman los datos de la columna Text.

```
lists_cloud1 = cloud1['Text'].tolist()
lists_cloud2 = cloud2['Text'].tolist()
```

#Generación de elementos en listas

```
words_cloud1 = ("").join(lists_cloud1)
```

```
words_cloud2 = ("").join(lists_cloud2)
#filtered_career = filtered_career.lower()
```

#Se crea la plantilla de la nube de palabras para las palabras de la categoría Violencia.

```
template = np.array(Image.open("nube.jpg"))
```

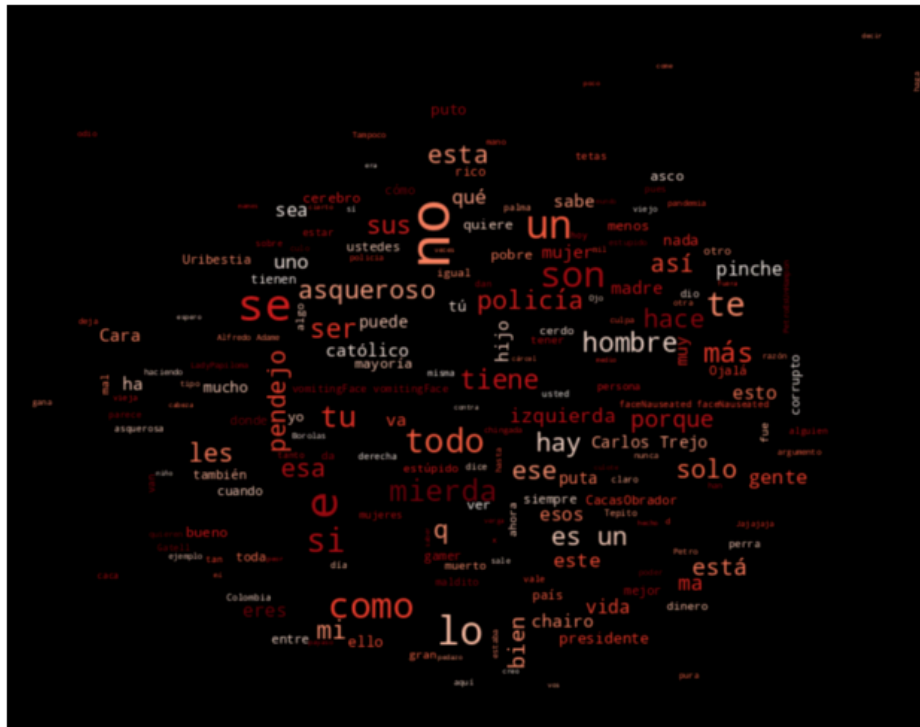
```
wordcloud_1 = WordCloud(max_font_size = 50,
                        margin=3,
                        stopwords=['que', 'la', 'el', 'los', 'las', 'con', 'y', 'pero', 'de', 'en', 'del',
                                   'con', 'a', 'sin', 'le', 'por', 'para', 'su', 'me', 'le', 'al', 'porque',
                                   'te', 'o', 'una', 'eso', 'ni', 'ya'],
                        background_color = "Black", mask = template,
                        colormap="Reds").generate(words_cloud1)
```

#Graficación de la nube de palabras para categoría Violencia

```
plt.figure(figsize=[8,8])
plt.imshow(wordcloud_1,interpolation='bicubic')
plt.axis("off")
plt.margins(x=1, y=1)
```

```
plt.savefig("wordcloud_Violencia.png", bbox_inches='tight') #comando para guardar la nube.
plt.show()
```



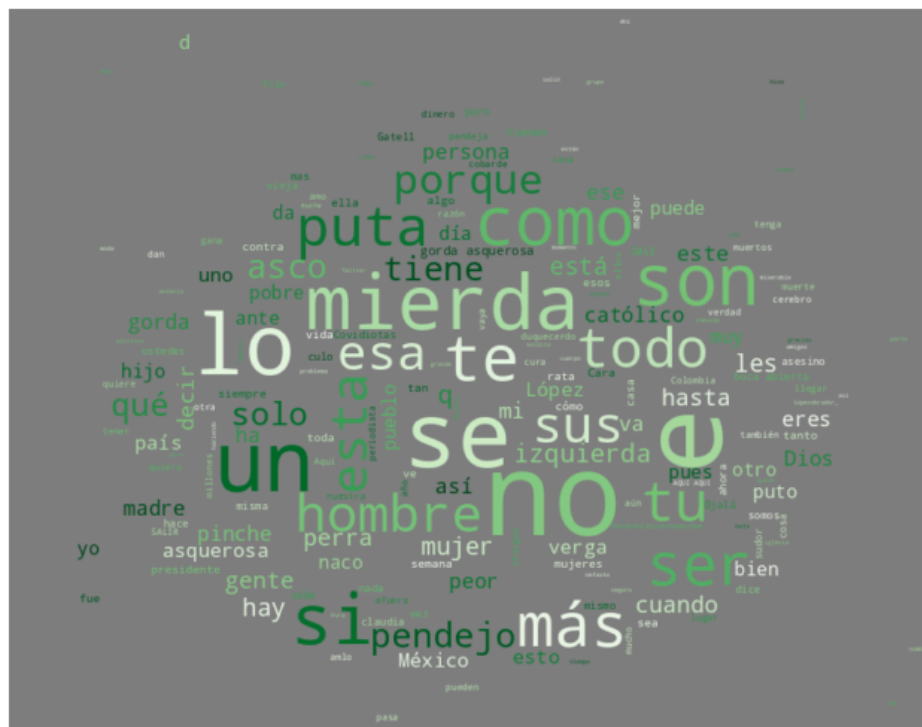


```
template = np.array(Image.open("nube.jpg"))

wordcloud_2 = WordCloud(max_font_size = 100,
                        margin=3,
                        stopwords=['que', 'la', 'el', 'los', 'las', 'con', 'y', 'pero', 'de', 'en', 'del',
                                  'con', 'a', 'sin', 'le', 'por', 'para', 'su', 'me', 'le', 'al', 'porque',
                                  'te', 'o', 'una', 'eso', 'ni', 'ya'],
                        background_color = "Grey",
                        mask = template,
                        colormap="Greens").generate(words_cloud2)

plt.figure(figsize=[8,8])
plt.imshow(wordcloud_2,interpolation='bilinear')
plt.axis("off")
plt.margins(x=1, y=1)

plt.savefig("wordcloud_Acoso.png", bbox_inches='tight')
plt.show()
```





10 rows x 97 columns

```
datos.info()
```




```

76 sev_liver_group // non-null object
77 sev_liver 77 non-null float64
78 sev_head_group 77 non-null object
79 sev_head 77 non-null float64
80 sev_blood_group 77 non-null object
81 sev_blood 77 non-null float64
82 sev_stomach_group 77 non-null object
83 sev_stomach 77 non-null float64
84 sev_cov_group 77 non-null object
85 sev_cov 77 non-null float64
86 association_covid 76 non-null object
87 preferred_description 76 non-null object
88 worries_covid_group 76 non-null object
89 worries_general 76 non-null float64
90 worries_covid 69 non-null object
91 leaflet_literacy 74 non-null object
92 age 74 non-null object
93 gender 74 non-null object
94 education 74 non-null object
95 employment 74 non-null object
96 household 74 non-null object
dtypes: bool(1), float64(75), int64(2), object(19)
memory usage: 106.8+ KB

```

```
datos.UserLanguage.value_counts()
```




	count
<b>UserLanguage</b>	
<b>EN</b>	142

```
datos.worries_covid_group.value_counts()
```



	count
<b>worries_covid_group</b>	
<b>Detractor</b>	60
<b>Passive</b>	13
<b>Promoter</b>	3

```
datos.age.value_counts()
```



	count
<b>age</b>	
<b>18-29</b>	37
<b>30-39</b>	17
<b>50-59</b>	11
<b>40-49</b>	7
<b>60-69</b>	2

```

labels = ['18-29', '30-39', '50-59', '40-49', '60-69'] #Etiquetas en eje X
counts = [37, 17, 11, 7, 2] #Valores en eje Y
ypos = np.arange(len(labels)) #converting text labels to numeric value, 0 and
ypos

```

```
plt.xticks(ypos, labels)
plt.xlabel("Rango etario")
plt.ylabel("Frecuencia")
plt.title("Frecuencia por grupo etario")
plt.bar(ypos, counts)
plt.savefig('etario.png')
```



Frecuencia por grupo etario