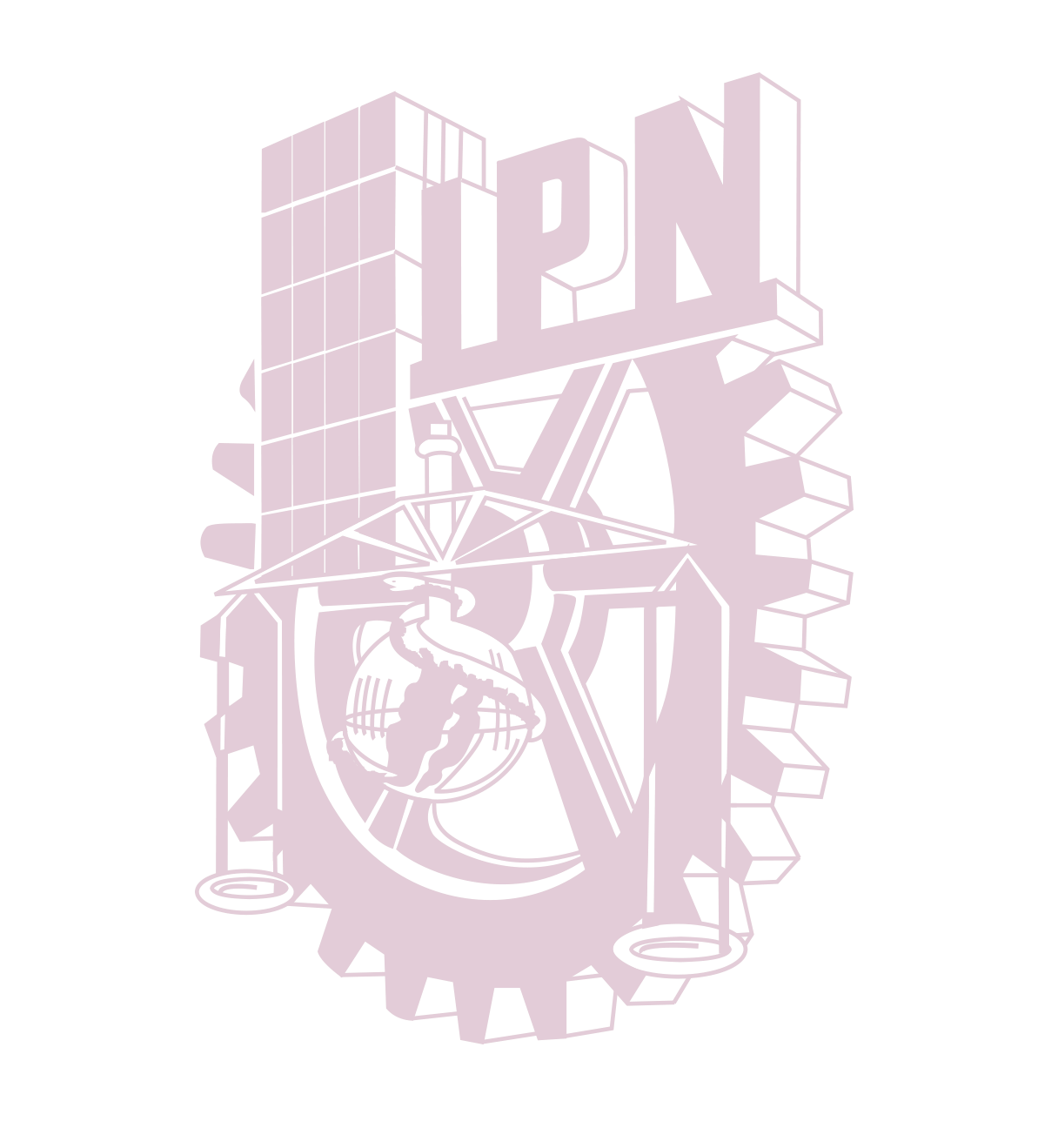


**Instituto Politécnico Nacional**

Escuela Superior de Cómputo

Practica 1

**Materia: TEORÍA DE LA COMPUTACIÓN**

**Grupo:** 5BV1

**Alumno:**

Calderon Sabbagh Juan Alberto

**Fecha:** 28/09/2023

**Nombre del profesor:** **CAMACHO VAZQUEZ VANESSA ALEJANDRA**

Contenido

[Procesamiento para el texto en español. 2](#_Toc146813328)

[Investigación. 2](#_Toc146813329)

[NLTK 2](#_Toc146813330)

[Spacy 2](#_Toc146813331)

[TextBlob 2](#_Toc146813332)

[Analisis exploratorio de textos. 3](#_Toc146813333)

[Identificar el numero total de tokens dentro del texto 3](#_Toc146813334)

[Identificar el numero total de tokens únicos dentro del texto 3](#_Toc146813335)

[Histograma de tokens únicos dentro del texto 4](#_Toc146813336)

[10 tokens más comunes 5](#_Toc146813337)

[10 tokens menos comunes 5](#_Toc146813338)

# Procesamiento para el texto en español.

## Investigación.

### NLTK

* Idiomas: NLTK es una biblioteca de procesamiento de lenguaje natural ampliamente utilizada que admite varios idiomas, incluyendo inglés y español.
* Facilidad de uso: NLTK es conocida por ser fácil de usar y es una excelente opción para quienes están aprendiendo procesamiento de lenguaje natural.
* Recursos lingüísticos: NLTK proporciona una amplia variedad de recursos lingüísticos predefinidos, como listas de palabras en varios idiomas, lemas y etiquetadores POS (Part-of-Speech) para inglés y español.
* Limpieza de texto: NLTK incluye funciones para tokenizar, eliminar stopwords, lematizar y etiquetar el texto, lo que facilita la preparación de los datos antes del análisis.

### Spacy

* Idiomas: spaCy es otra biblioteca de procesamiento de lenguaje natural popular que también admite varios idiomas, incluyendo inglés y español.
* Eficiencia: spaCy está diseñada para ser altamente eficiente y rápida, lo que la hace ideal para procesar grandes cantidades de texto de manera rápida y precisa.
* Modelos pre-entrenados: spaCy ofrece modelos pre-entrenados de alta calidad para varios idiomas, incluyendo modelos específicos para el procesamiento de español y inglés. Estos modelos incluyen etiquetadores POS, análisis sintáctico, reconocimiento de entidades nombradas y más.
* Facilidad de extensión: spaCy es altamente modular y permite la extensión con componentes personalizados para adaptarse a tareas específicas de procesamiento de texto.

### TextBlob

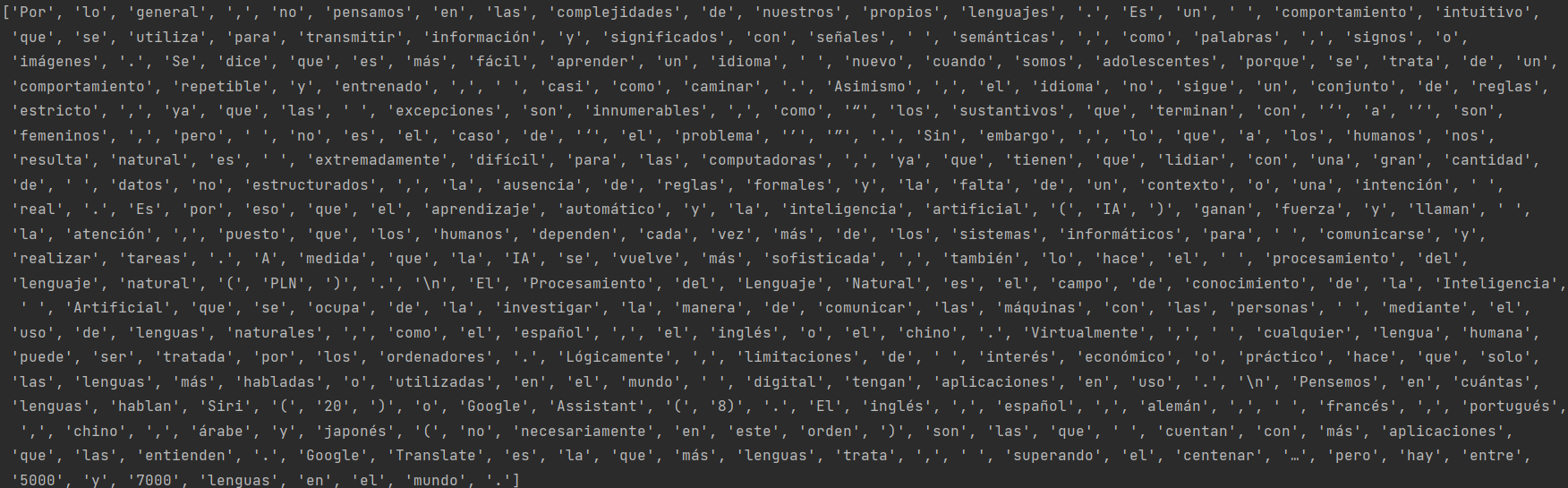
* Idiomas: TextBlob es una biblioteca que simplifica el procesamiento de texto en varios idiomas, incluyendo español e inglés.
* Facilidad de uso: Es conocida por su facilidad de uso y su simplicidad. Ofrece una API sencilla para tareas comunes de procesamiento de texto como tokenización, análisis de sentimientos, extracción de frases clave y corrección ortográfica.
* Análisis de sentimientos: TextBlob incluye un módulo para el análisis de sentimientos que permite determinar si un texto tiene una connotación positiva o negativa.
* Integración con WordNet: Ofrece integración con WordNet para la lematización y búsqueda de sinónimos en inglés.

El módulo que yo voy a utilizar es Spacy ya que me ofrece bastantes funciones y tiene un buen soporte tanto para inglés como para español, además el uso de esta librería me servirá para a la larga poder desarrollar proyectos de mayor complejidad. Igual se usará NLTK para la tarea de la de stemming ya que Spacy no cuenta con dicha función.

## Analisis exploratorio de textos.

En esta seccion se explorara el texto en cuestión y nos familiarizaremos con el texto mediante la obtención de sus tokens, identificación de los tokens mas y menos comunes dentro del texto y además vamos a ver cuales son los tokens menos comunes dentro del texto

### Identificar el numero total de tokens dentro del texto

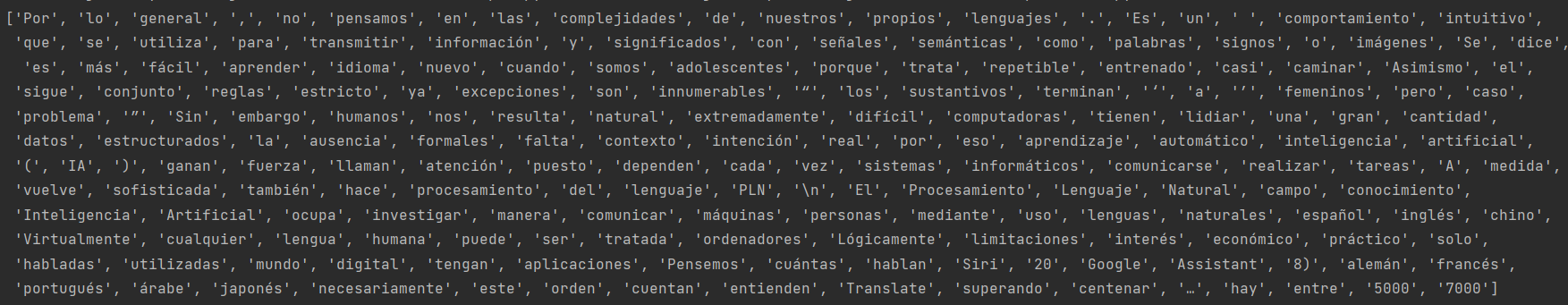
Al realizar la tokenizacion se obtienen 396 tokens, entre estos están caracteres especiales y signos de puntuación.

*Lista de tokens encontrados dentro del texto*

*Numero de tokens obtenido mediante len(nombre\_Array)*

### Identificar el numero total de tokens únicos dentro del texto

Para identificar los tokens únicos, se va recorriendo la lista de todos los tokens y en cada iteración se meten a un array y se va comprobando que estos tokens no existan dentro de este array para ser metidos. Finalmente se obtienen 197 tokens únicos.



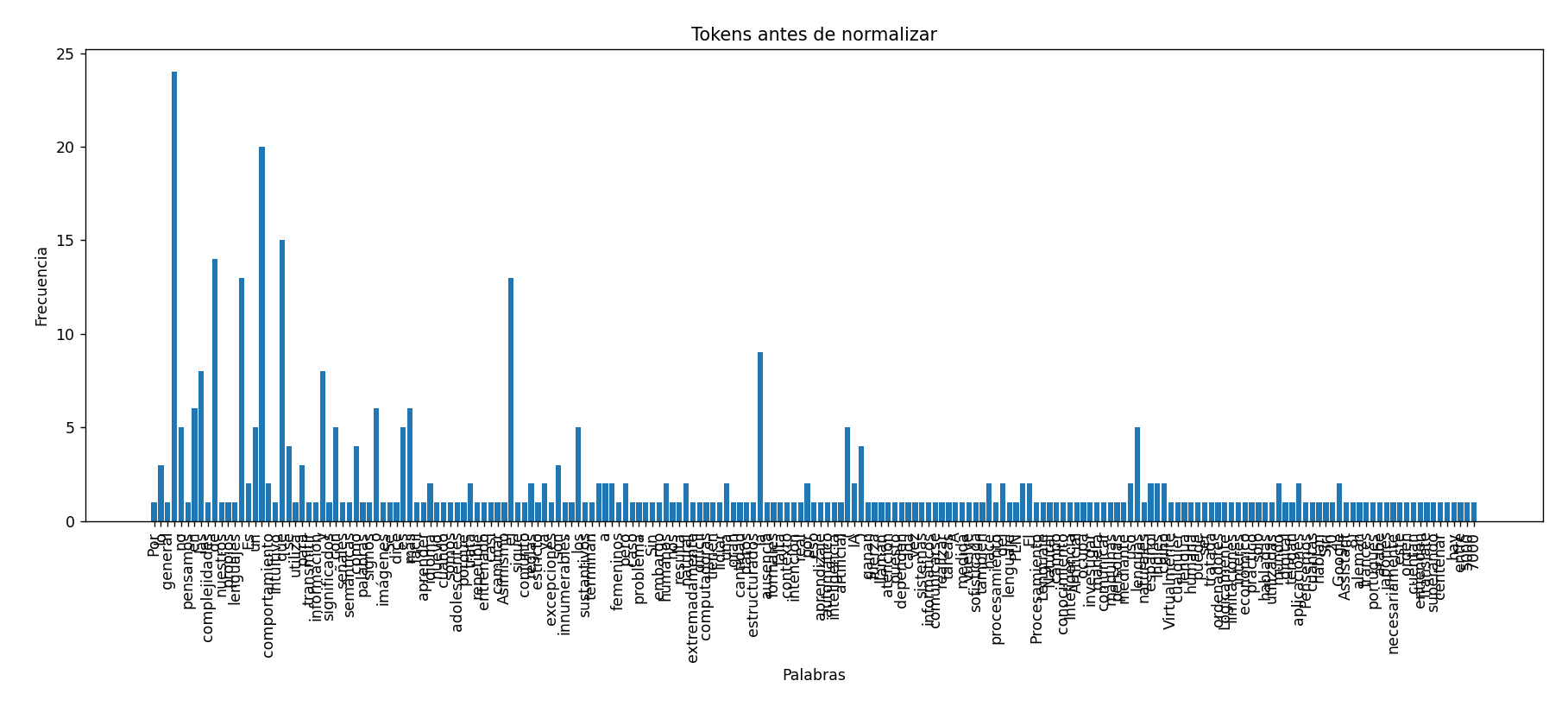
*Lista de tokens únicos*

**

*Numero de tokens obtenido mediante len(nombre\_Array)*

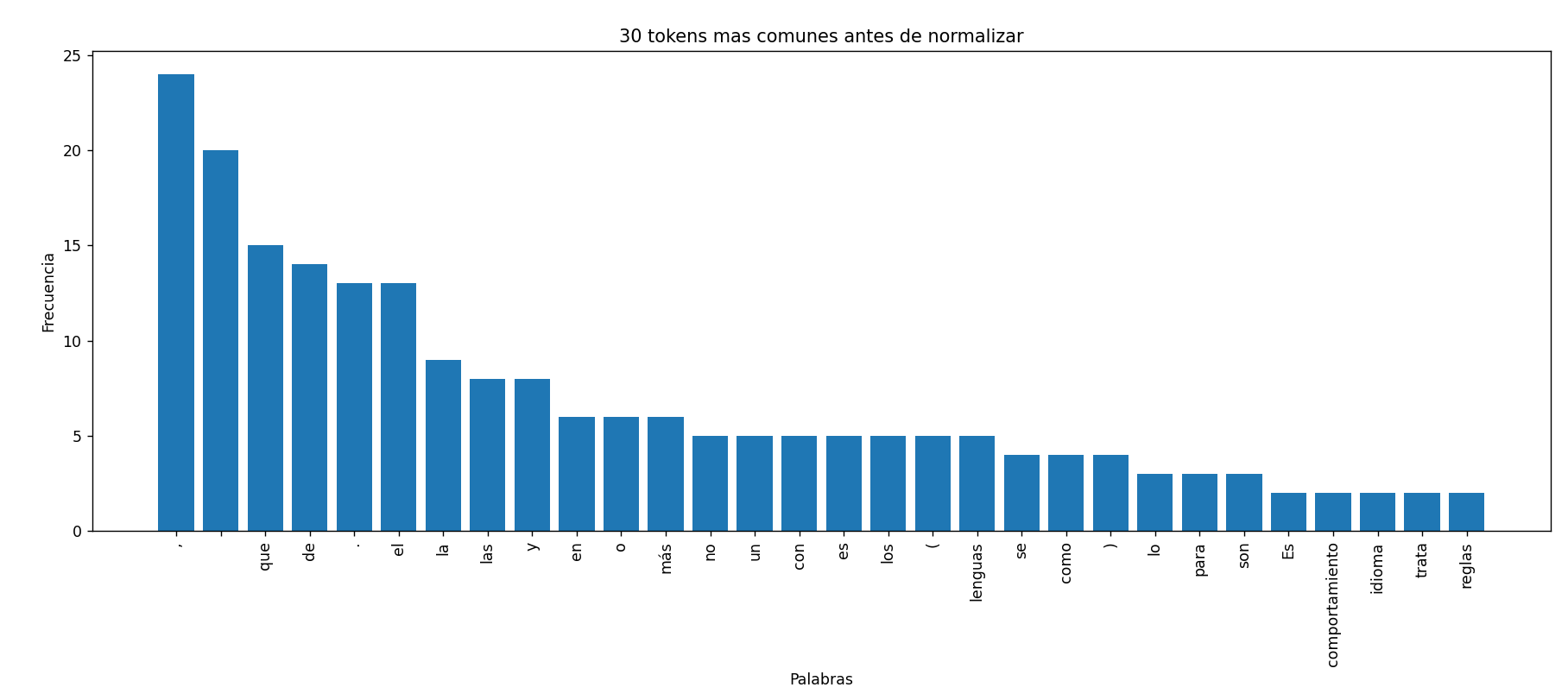
### Histograma de tokens únicos dentro del texto

Por medio de la la librería matplotlib se grafica el histograma de cada token, igual se obtiene la frecuencia de cada uno.



*Histograma de tokens únicos dentro del texto*

Como se puede observar en la imagen, el histograma obtenido esta en un completo caos, así que solo se graficaran los 30 tokens mas comunes.

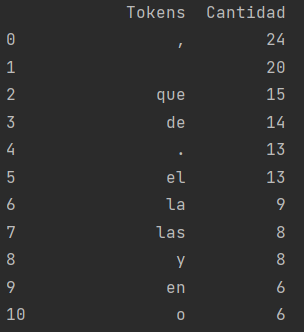


*Histograma de los 30 tokens mas comunes*

De esta manera se puede apreciar de mejor manera cuales son los tokens mas comunes. En la siguiente sección se hablara en mas detalle sobre los resultados obtenidos en este punto.

### 10 tokens más comunes

A partir de la lista de los 30 tokens mas comunes podemos obtener los 10 mas comunes.

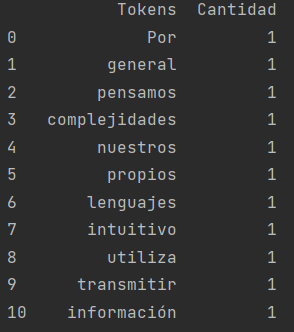


*Lista de los 10 tokens mas comunes*

Con esta lista podemos observar mas claramente los resultados obtenidos en los histogramas de la sección anterior. El token con mayor frecuencia es una coma, el segundo un carácter que no es visible y los demás son tokens que no nos ofrecen ningún tipo de información relevante.

### 10 tokens menos comunes

Similar al proceso anterior para obtener los tokens mas comunes se realizo algo parecido pero para los menos comunes.



*Lista de los 10 tokens menos comunes*

Para el caso de los tokens menos comunes no hay mucho que agregar ya que hay demasiados que aparecen una sola vez y la lista se hizo conforme fueron apareciendo los tokens que tenían solo una aparición.