|  |
| --- |
|  |
| TALLER 2 |
| Minería de Datos & Big Data: “Web Scraping” |

|  |
| --- |
| Integrantes: Cristian Álvarez – Jorge González  Profesor Mauricio Sepúlveda |

****

# 

[**Introducción**](#_wn6n9bdxzsp2) **3**

[**Desarrollo**](#_e8jfkxst3e26) **3**

[Parte 1. Implementación de Código](#_8fq6l9fwsskz) 3

[Parte 2. Análisis LDA](#_x6xjl2d439jk) 4

[Parte 3. Modificación Web Scraping](#_isasafxakm2o) 5

[Parte 4. Serie de Tiempo y red LSTM](#_5t6kfa152tc6) 5

[Parte 5. Clasificaciones y Agrupaciones.](#_w2scbllcafum) 5

[**Conclusión**](#_qj5uvxgjxdzu) **5**

[**Anexos**](#_9usu94t8za7g) **5**

[**Referencias**](#_jj0qta3sx0wd) **5**

# 

# 

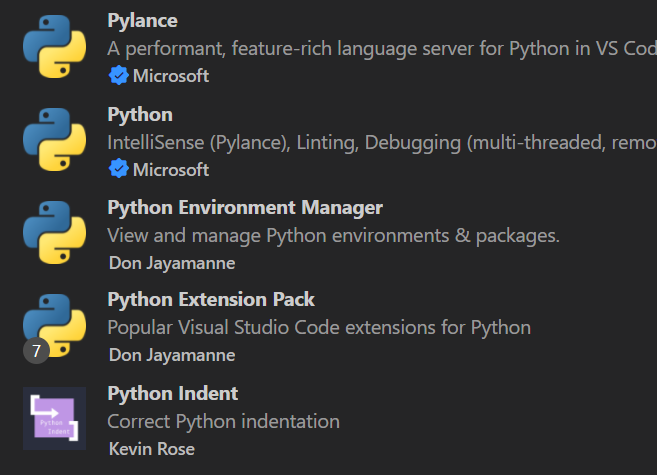
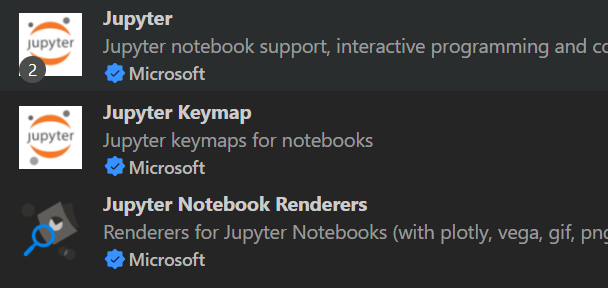
Como bien sabemos dentro existe una cantidad inimaginable de información y datos alrededor de toda la red de internet, estos datos nos sirven tanto para realizar trabajos y análisis. Para poder facilitar la extracción de los datos que nos resultan importantes para nuestros trabajos es que existe el Web Scraping.

El Web Scraping, como se comentaba, es una técnica utilizada para extraer grandes cantidades de información desde sitios web, como datos de contacto, correos electrónicos o como en nuestro caso términos de búsqueda. Para luego poder guardar nuestros datos en bases de datos locales con el fin de procesarlos.

En este informe utilizaremos la técnica de Web Scraping y recopilaremos información desde sitios de publicaciones científicas relacionados a Inteligencia Artificial.

# **Desarrollo**

Para empezar, nosotros utilizamos Visual Studio Code para trabajar, a través de la herramienta que ofrece Anaconda Navigator ya que es donde mas cómodos nos sentíamos para trabajar. Además, mencionar que adaptamos el VSC para utilizar Jupyter, como también algunas extensiones para trabajar con Python. Le adjunto imagen de las extensiones que utilizamos(No todas requeridas ni obligatorias para realizar el taller):

Otro punto a considerar es que en todos los archivos copiamos las mismas importaciones, pero no todas fueron utilizadas, las colocamos en caso de que fuera necesario, por ello algunas no fueron requeridas, pero están en los documentos igualmente. Agregar que en las imágenes que colocaremos se puede apreciar un color mas oscuro en las importaciones, el cual indica que no están siendo utilizadas en ese documento, por lo que no hace falta llamarlas. Ejemplo:



## **Parte 1. Implementación de Código**

Para esta primera etapa se nos pide implementar un código para extraer la información desde un sitio de publicaciones científicas, en específico “Web Of Science” y buscar artículos sobre el tema en el que en nuestro caso quisimos seleccionar el tema “Security TI” entre los años 2010 y 2021.

Por ello se debe a través de web Scraping lograr que se pueda llegar a la pagina destinatario con los filtros especificados para posteriormente extraer la información

En cuanto a la recopilación de datos estos se almacenaron en un archivo Excel con las siguientes columnas: “ID”, “TÍTULO”, “MES”, “AÑO”, “AUTORES”, “REVISTA”, “CITACIONES” y “REFERENCIAS”. Para el tópico de “Security TI”, el resultado de la extracción de datos fue de un total de 750 registros.

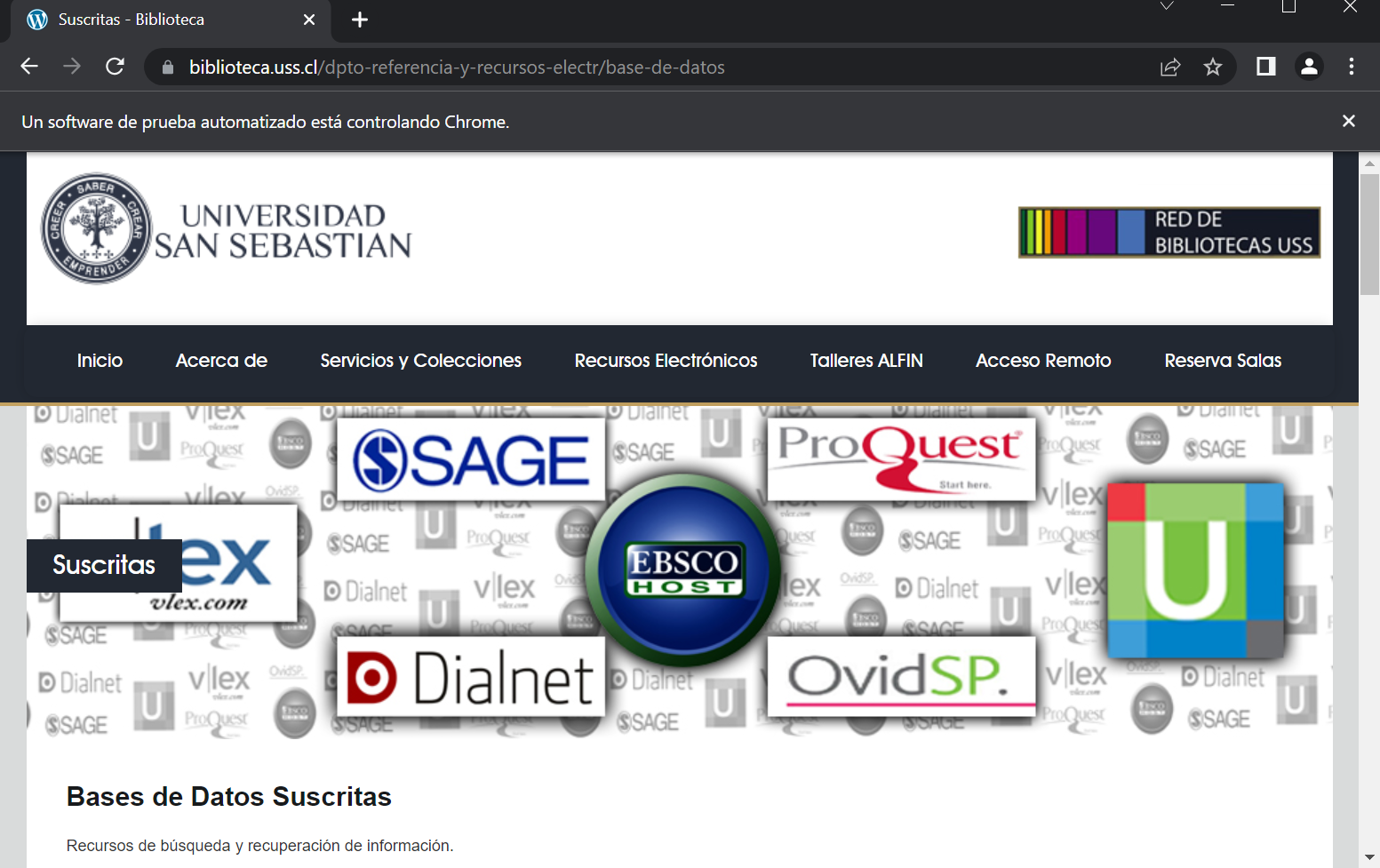
### Código

A continuación, le dejamos el link del GitHub para que puedan ver todo el código que se realizo para cumplir con lo requerido

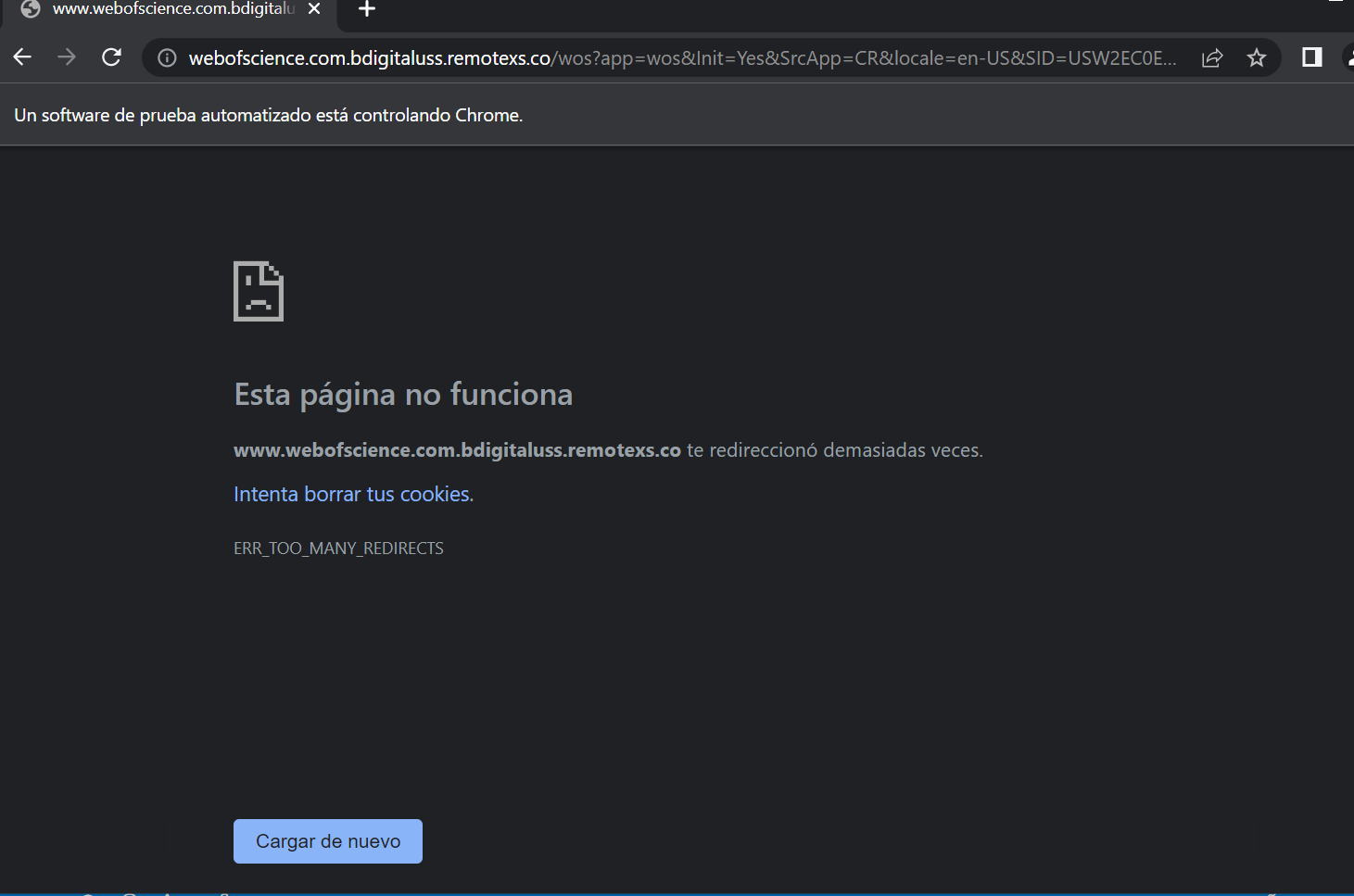
<https://github.com/jorgeigna/USS-DMDB-Taller2/blob/main/Parte1.ipynb>

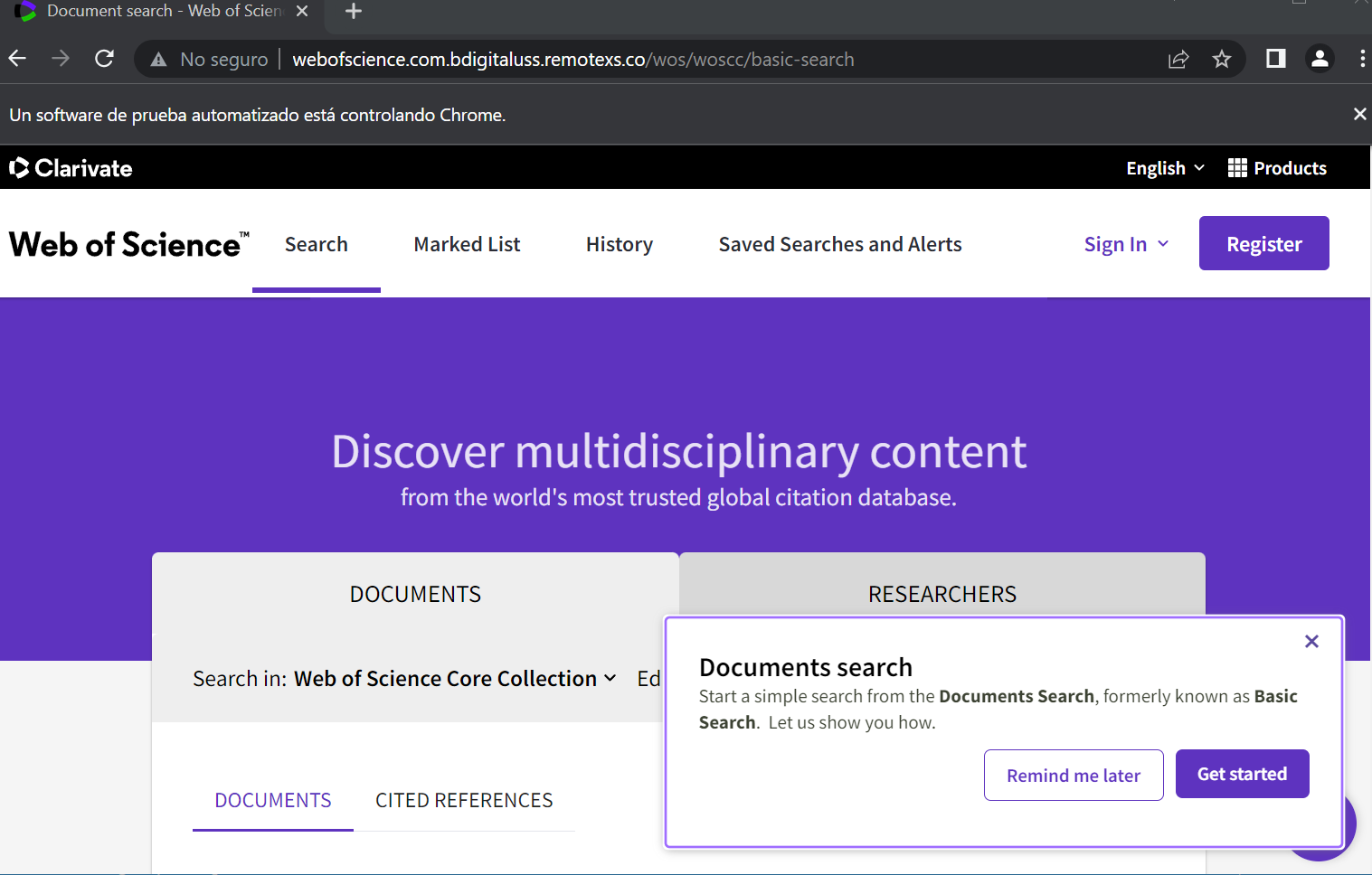
### Resultados

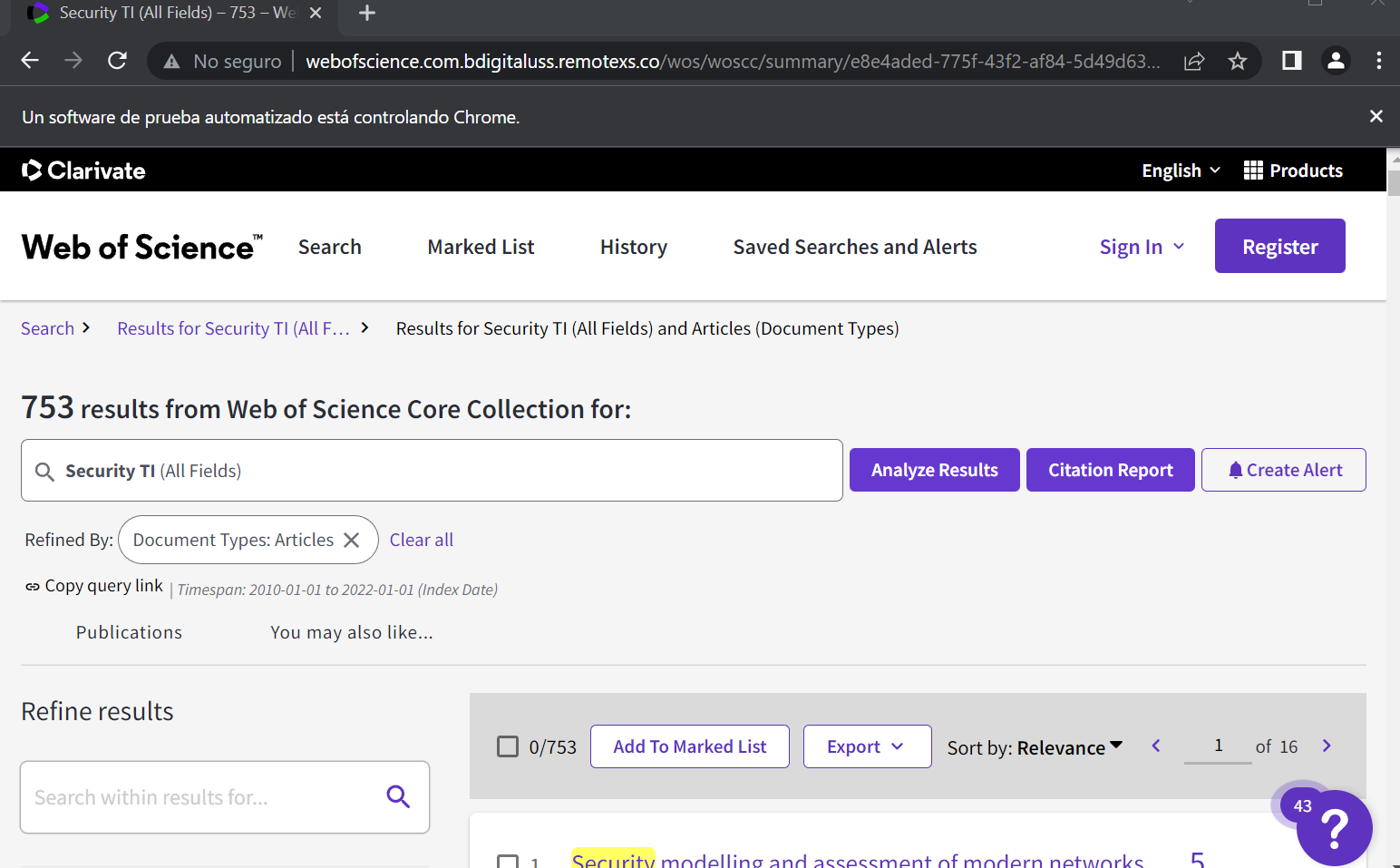
Se parte en el link entregado el cual se selecciona el apartado “Web Of Science”, luego se inicia sesión en la universidad ingresando usuario y contraseña



Debido a un mal rendimiento general de la pagina esta no carga, por lo que volvemos a cargar la pagina y continuamos con ingresar la búsqueda “Security TI”, abrimos la opción de filtrar por fecha “2010-01-01” hasta “2022-01-01”, aceptamos los parámetros ingresados y en la siguiente pagina hacemos filtro para ver solo los artículos. Asi acaba la primera parte de este punto.

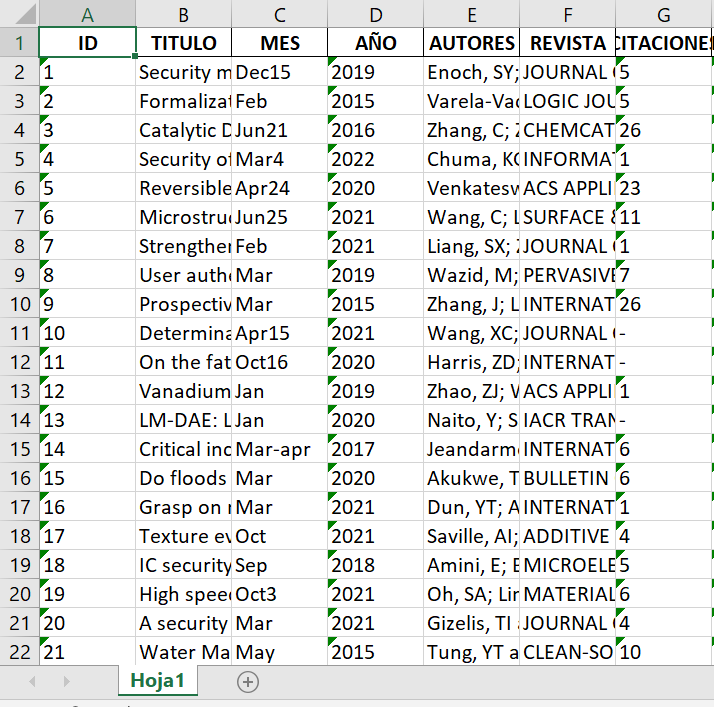






Ahora es cuando comienza la extracción de datos el cual extraemos “ID”, “TÍTULO”, “MES”, “AÑO”, “AUTORES”, “REVISTA”, “CITACIONES” y “REFERENCIAS” de cada uno de los artículos y recorriendo uno a uno de ellos, como también todas las páginas de la búsqueda que estamos haciendo.

Esta información la almacenamos un archivo Excel llamado “DatosInvestigacion.xlsx”. donde podrán encontrar no solo ese archivo sino la copia llamada “**DatosInvestigacion1-CargaCompleta.xlsx**” que contiene todos los datos almacenados para ver el resultado.



## **Parte 2. Análisis LDA**

Para esta etapa, se nos pide implementar un análisis LDA, el cual nos permite clasificar un texto y así generar Topics para las cada revista y tener un análisis de ello.

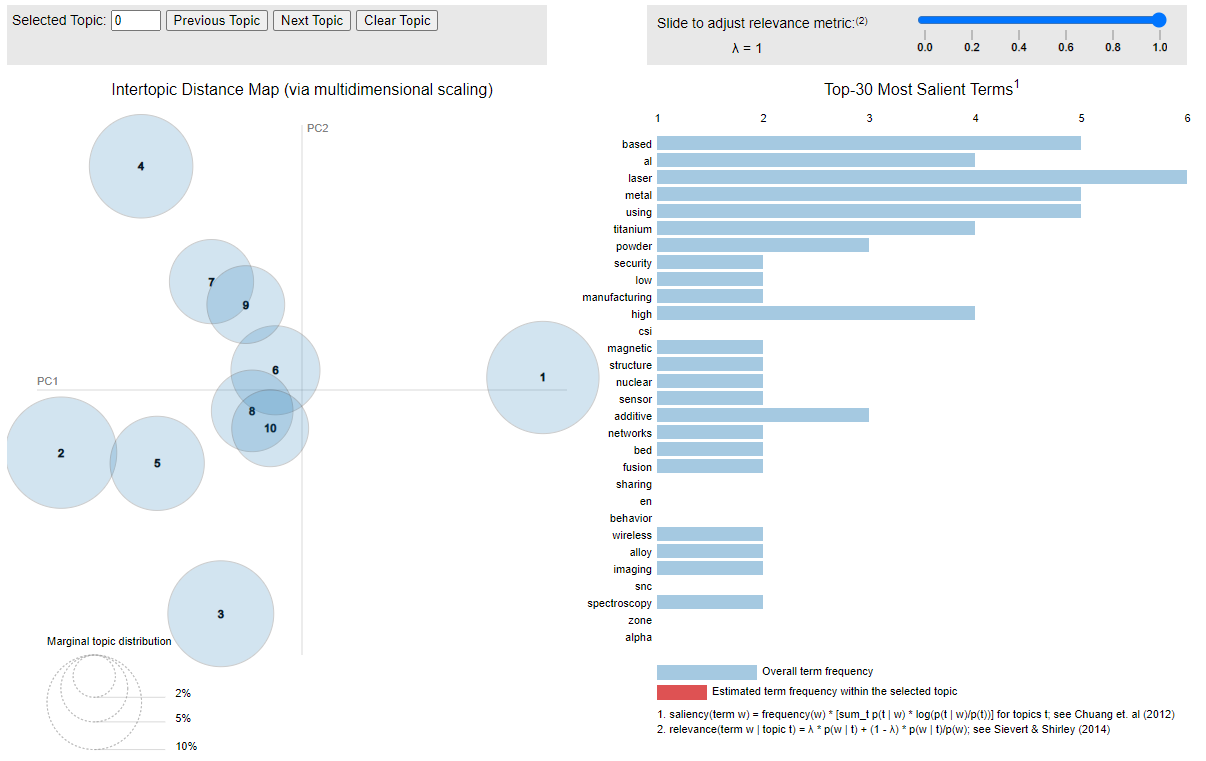
En este caso utilizamos el archivo Excel donde se obtuvieron los datos previos, eliminamos las columnas, quedándonos solo con las de “TÍTULO” y “REVISTA” y mediante un procedimiento llamado Wordcloud, realizamos esta llamada nube de palabras para determinar la relevancia de estas.

Para finalizar con la ayuda de la librería “pyLDAvis”, visualizamos de una manera interactiva los clústeres que se crearon con el análisis LDA.

### Código

El código lo puedes encontrar en el siguiente link

<https://github.com/jorgeigna/USS-DMDB-Taller2/blob/main/Parte2.ipynb>

Resultados

## **Parte 3. Modificación Web Scraping**

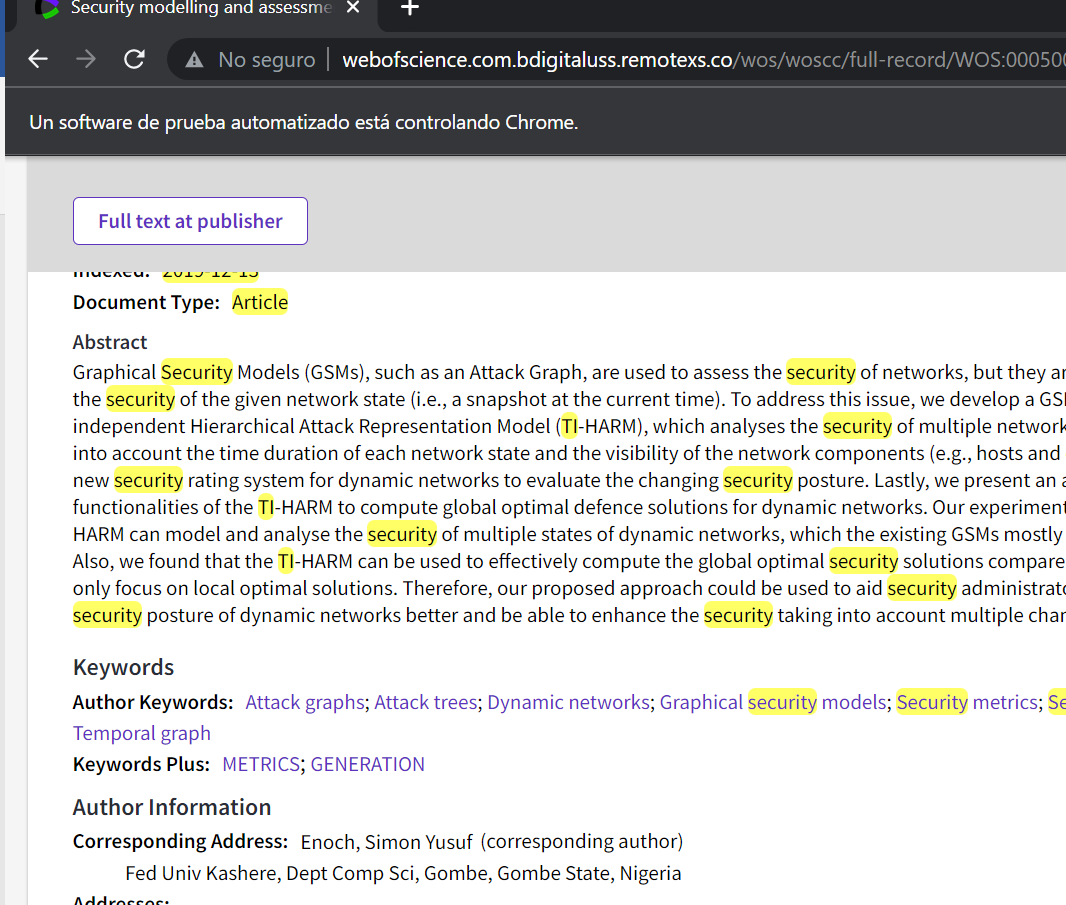
Como su título lo dice, en esta etapa realizamos una modificación del Web Scraping, para obtener la información adicional de “Author Keywords”, “Keyword Plus”, “Categorie” y “Research Areas”, estas columnas las añadimos a nuestro archivo previo de Excel. Y para ver los resultados dejamos un archivo Excel con la carga completa.

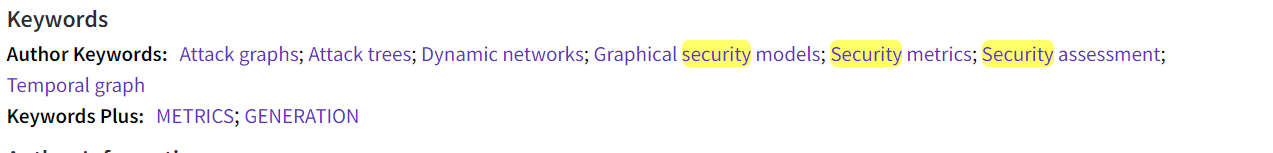
### Código

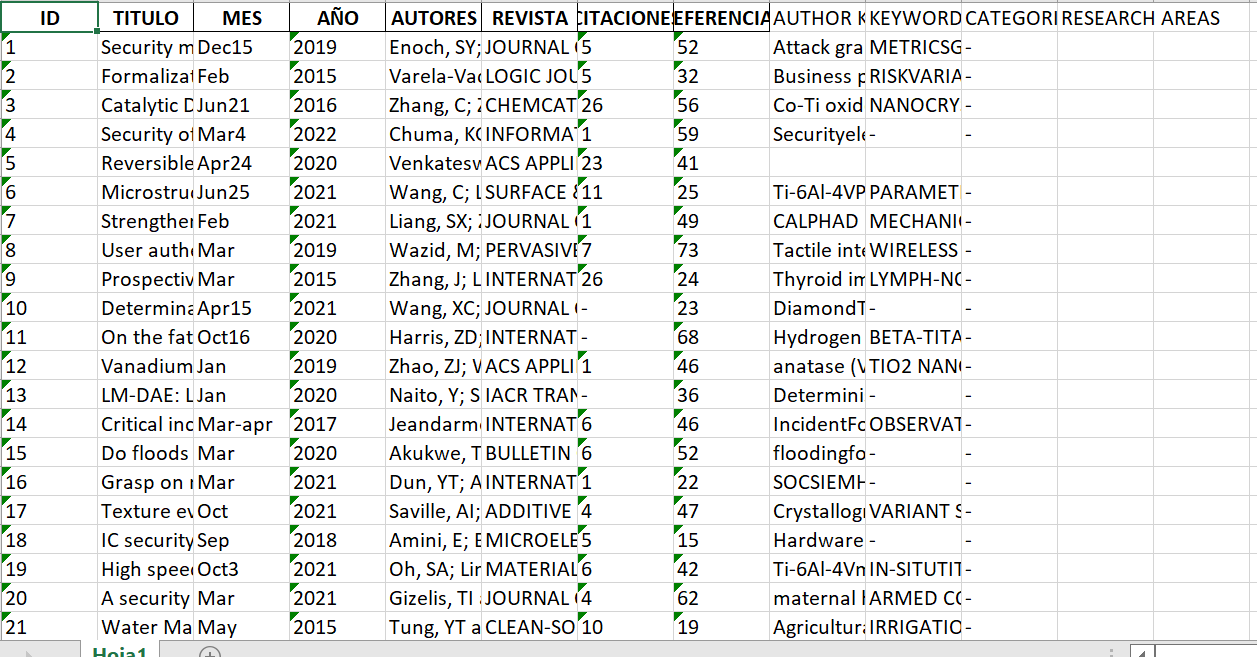
<https://github.com/jorgeigna/USS-DMDB-Taller2/blob/main/Parte3.ipynb>

### Resultados

Como resultado obtenemos la información de la pagina especifica de cada artículo por lo que debemos ir de articulo a articulo capturando los datos y almacenándolos en el archivo mencionado





## **Parte 4. Serie de Tiempo y red LSTM**

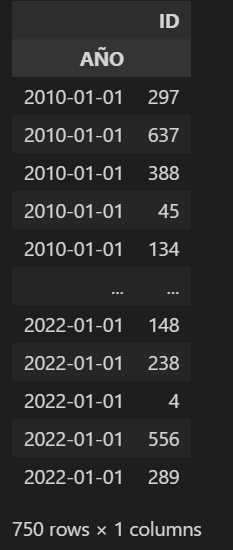
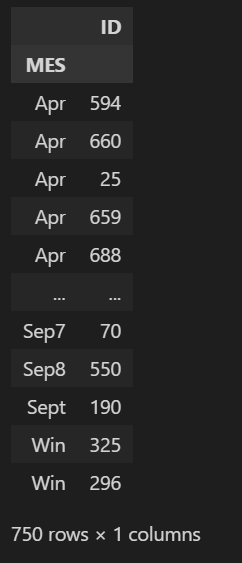
En este caso, se nos pide entrenar una red LSMT, generando series de tiempo para así, predecir o pronosticar el número de publicaciones que se van a registrar en el año seleccionado.

### Código

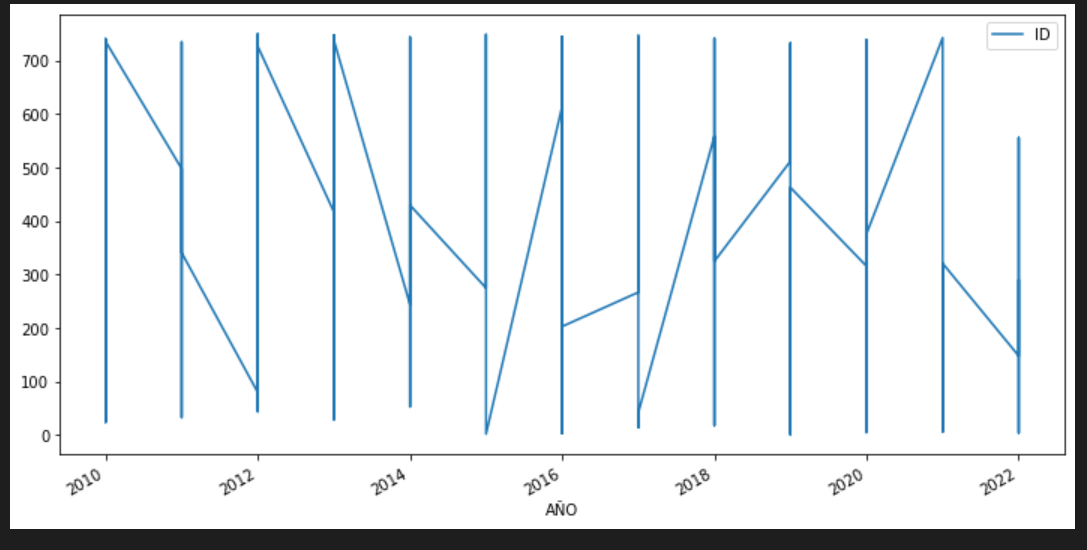
<https://github.com/jorgeigna/USS-DMDB-Taller2/blob/main/Parte%204.ipynb>

### Resultados

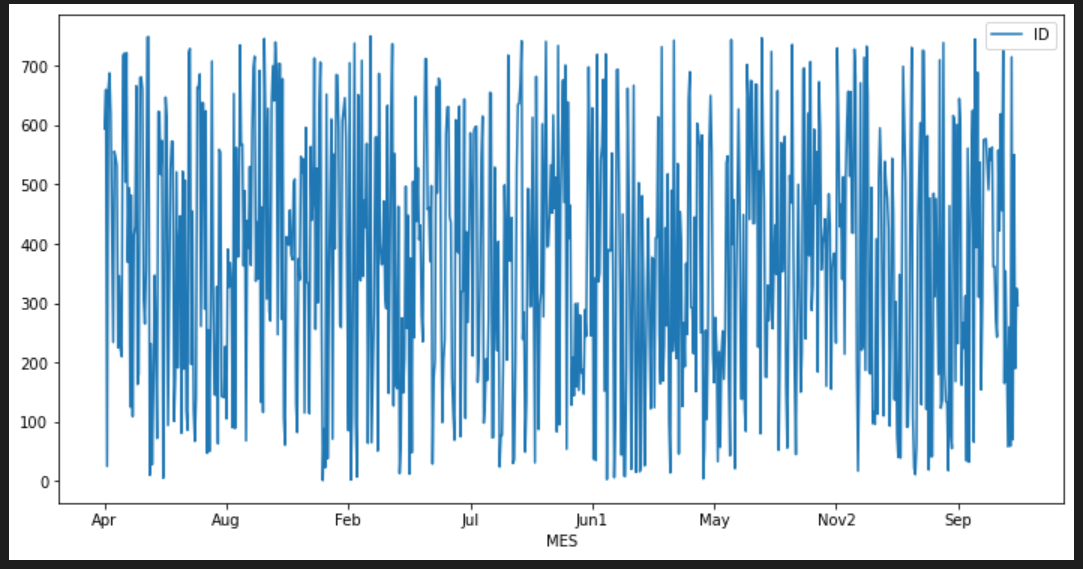
Como resultados obtenidos es la clasificación de los artículos ordenados tanto por año como por mes, además de ello ver como se manifiestan a lo largo del tiempo en relación con la cantidad de artículos existentes por fecha.

Años vs Artículos



Meses vs Artículos



Debido a un problema con la librería Tensorflow, no se nos fue posible testear con la LSMT para ver el pronóstico esperado, pero en teoría si ejecuta el código teniendo la libraría podría ver resultados

## **Parte 5. Clasificaciones y Agrupaciones.**

En esta última etapa nos piden realizar clasificaciones y agrupaciones con nuevos datos extraídos, por ellos realizamos los mismos procedimientos que en la parte1 con la diferencia que filtramos solo para un año en específico.

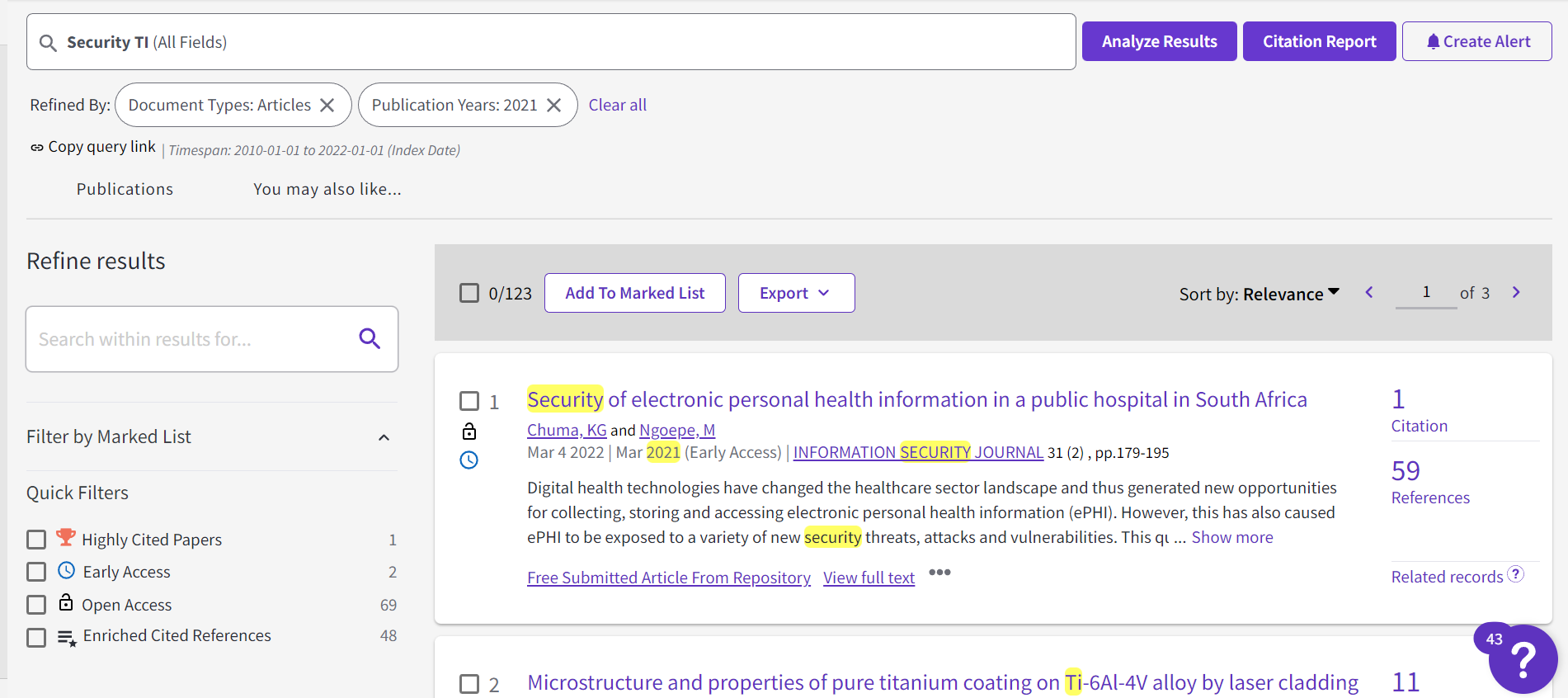
### Código

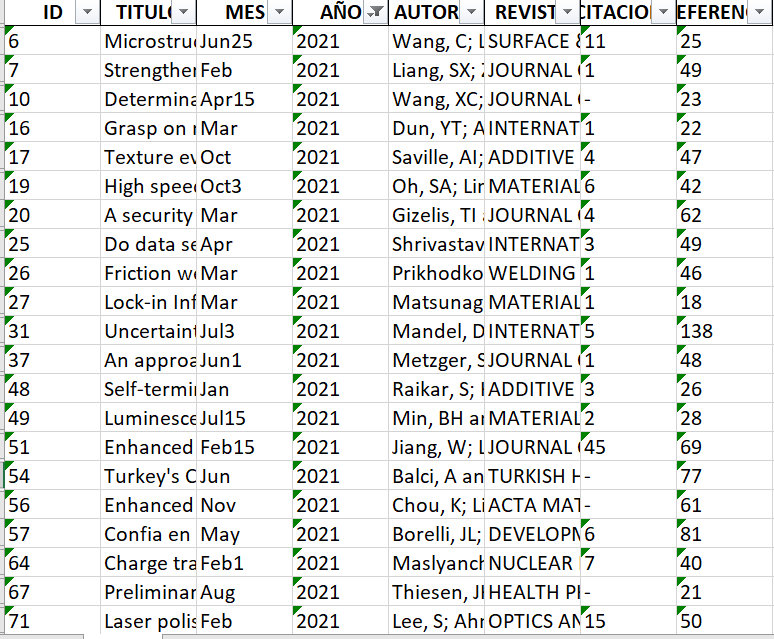
<https://github.com/jorgeigna/USS-DMDB-Taller2/blob/main/Parte5.ipynb>

### Resultados

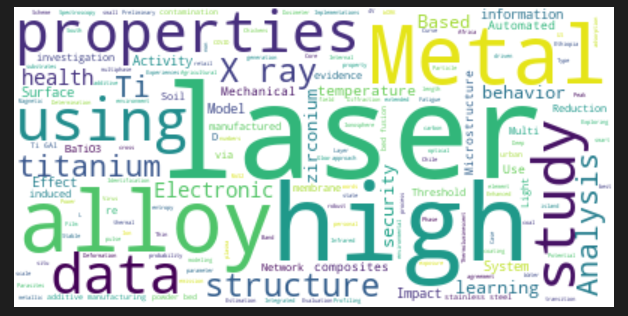
Como resultado obtuvimos un nuevo archivo con los datos solo de un año anteriormente filtrado, además de ello recopilamos las palabras mas utilizadas en cada articulo

Extracción de datos:

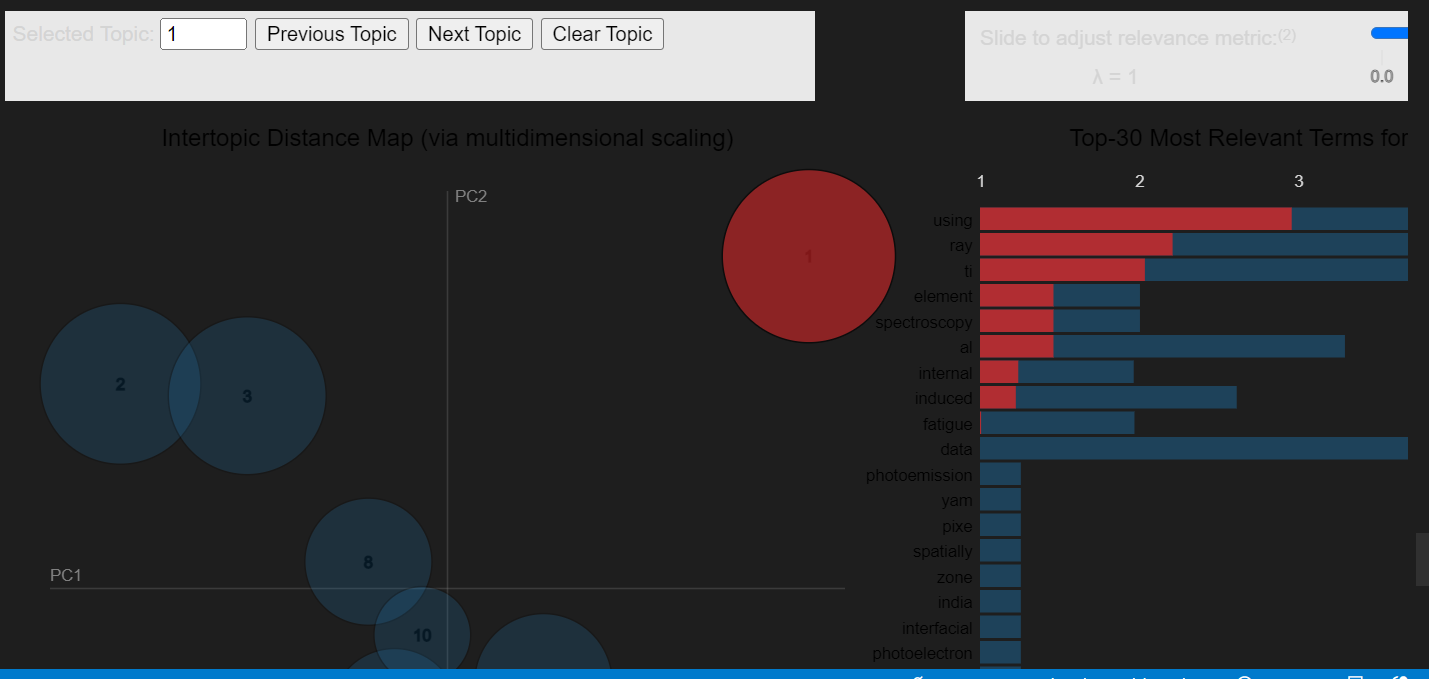




Palabras más utilizadas ejemplo:



Resultados de top de palabras



# **Conclusión**

Como se pudo apreciar en el trabajo que hicimos y en este informe, el Web Scraping nos permite obtener una gran cantidad de información para poder analizarla de una forma más eficaz.

Este método de extracción de datos, puede llegar a ser muy complejo si es que no cuentas con las herramientas necesarias o información clara de lo que necesitas o tienes que hacer.

El Web Scraping, se puede utilizar para examinar los distintos productos o alguna comparación de precios, y así analizar los datos dentro de una empresa que está relacionada con estos productos, por ejemplo y procesar estos datos.

# 

# **Anexos**

# **Referencias**

*How to Get Month Name from Month Number in Python - Studytonight*. (s. f.). Study Tonight. Recuperado 27 de junio de 2022, de https://www.studytonight.com/python-howtos/how-to-get-month-name-from-month-number-in-python

*Biswal, A. (2022, 22 junio). Recurrent Neural Network (RNN) Tutorial: Types, Examples, LSTM and More. Simplilearn.Com. Recuperado 27 de junio de 2022, de https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn*

# 

## Bibliografía / Links

<https://elmundodelosdatos.com/topic-modeling-gensim-asignacion-topicos/>

<https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0>

<https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0>

<https://datascienceplus.com/topic-modeling-and-latent-dirichlet-allocation-lda/>

Para el punto 3

<https://guru99.es/rnn-tutorial/#5>

[How to Get Month Name from Month Number in Python - Studytonight](https://www.studytonight.com/python-howtos/how-to-get-month-name-from-month-number-in-python)

Códigos Pt4

<https://drive.google.com/drive/folders/16BstwgmzTAGLXpFG7_nWNtOb7-InaFBd?usp=sharing>