



# UNIVERSIDAD DE GRANADA

## Recuperación de Información Práctica 2 : Analyzer

Alejandro Ledesma Pascual  
Curso 2022/2023

# Índice

<b>1.- Introducción</b>	<b>3</b>
<b>2.- Programa</b>	<b>3</b>
2.1 Analyzer	4
2.2 Filtros	5
2.3 Custom Analyzer	5

# 1.- Introducción

Esta práctica consiste en probar distintos tipos de analizadores de contenidos y ver las diferencias entre los mismos. Para ello, usaremos “Lucene” que es un motor de búsqueda de texto.

Dentro de esta biblioteca existen distintos tipos de Analizadores, que dependiendo el tipo puede estudiar el texto de diferentes formas como por ejemplo borrando todo contenido que no sea una palabra o usando solamente la raíz de las palabras como contenido. A parte de incluir sus propios analizadores, se pueden crear un propio, como podemos ver en el apartado 3.3 de esta práctica.

## 2.- Programa

Este programa sigue la estructura de la práctica anterior, usando solamente dos ficheros, “Fichero.java” para almacenar los datos relativos a los documentos usados y “Main.java” para ejecutar las distintas opciones que nos pide la práctica.

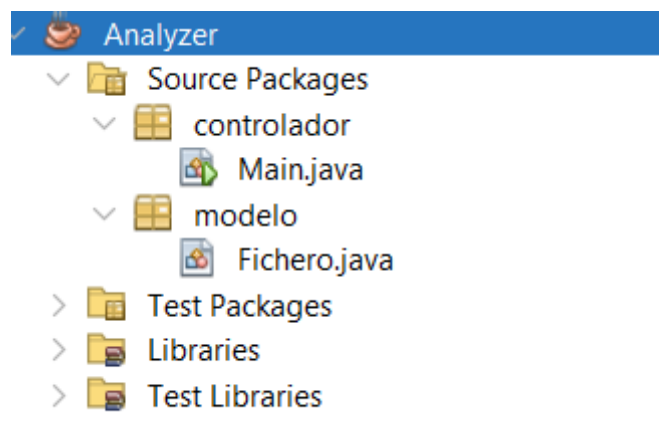


Figura 1. Disposición de los ficheros

Para poder ejecutar el programa se tiene que ir al CMD donde se encuentra el ejecutable del programa y añadir la ruta de los archivos que vamos a analizar. Antes de realizar sus funciones, el programa lee y almacena todos los datos que necesita de los documentos, tal y como, se hizo en la práctica 1.

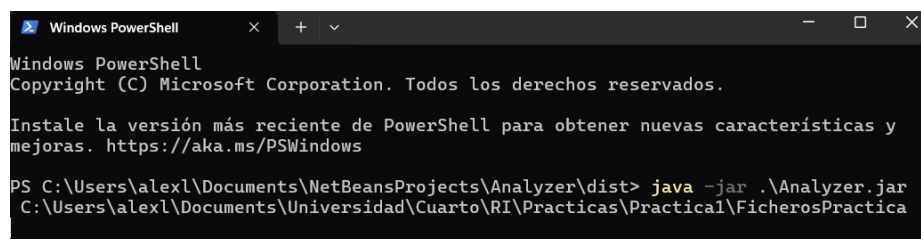


Figura 2. Ejecución programa

## 2.1 Analyzer

Para esta práctica se ha decidido usar 3 diferentes de analizadores : SimpleAnalyzer, WhitespaceAnalyzer y StandardAnalyzer. Para ello se ha usado una función que ejecuta los diferentes analyzers y almacena sus resultados para después hacer un estudio.

```
public ArrayList<List> analizadores() throws IOException {
    ArrayList<List> analizadores = new ArrayList<>();

    analizadores.add(simpleAnalyzer());
    analizadores.add(whiteSpaceAnalyzer());
    analizadores.add(standardAnalyzer());

    return analizadores;
}
```

**Figura 3.** Función analizadores.

El “SimpleAnalyzer” divide el texto separando todo aquello que no sean letras y números.

El “WhiteSpaceAnalyzer” divide el texto considerando como separadores de tokens los espacios en blanco.

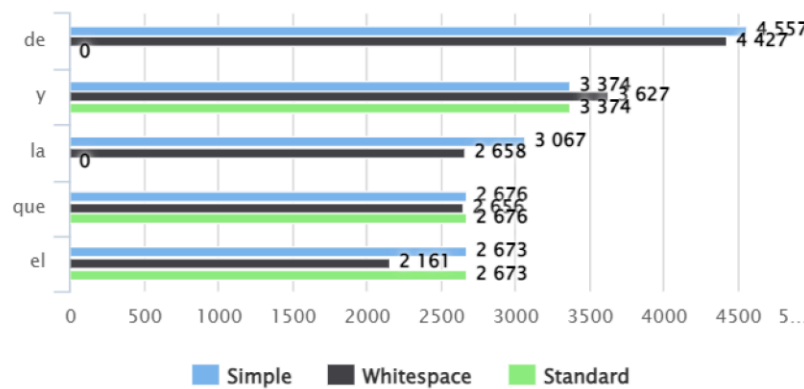
El “StandardAnalyzer” es el más elaborado, y es capaz de gestionar acrónimos, direcciones de correo, etc. Convierte a minúscula y elimina palabras vacías.

Para realizar los estudios se han escogido los diferentes “CSV” del libro “El Señor de los Anillos La Comunidad del Anillo”. Como se puede apreciar en la figura 4, la frecuencia de palabras cambia bastante de un analizador a otro, siendo el más diferencial el StandardAnalyzer. Se puede apreciar que de, al estar incluido en la lista de palabras vacías en el Standard Analyzer no la tiene en cuenta.

SimpleAnalyzer			WhiteSpaceAnalyzer			StandardAnalyzer		
1	Text	Size	1	Text	Size	1	Text	Size
2	de	4557	2	de	4427	2	y	3774
3	y	3774	3	y	3627	3	que	2676
4	la	3067	4	la	2658	4	el	2673
5	que	2676	5	que	2656	5	a	1955
6	el	2673	6	el	2161	6	se	1544
7	en	2164	7	en	1989	7	no	1505
8	los	2097	8	a	1894	8	un	1229
9	a	1955	9	los	1819	9	del	1003
10	se	1544	10	se	1382	10	una	946

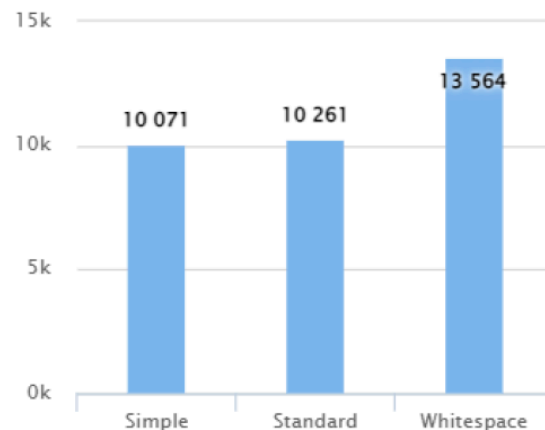
**Figura 4.** Top 10 frecuencias de los 3 analizadores

En el siguiente gráfico se puede ver la diferencia entre 5 de los términos con más apariciones en los 3 analizadores. Donde se puede apreciar que tanto el “StandardAnalyzer” como el “SimpleAnalyzer” tienen las mismas frecuencias en los términos excepto si el término aparece en la lista de palabras vacías.



**Figura 5.** Diferencias entre los diferentes términos

Para finalizar este estudio se puede apreciar la diferencia de términos totales donde el SimpleAnalyzer es el que menos términos aparecen y el WhitespaceAnalyzer en el que más.



**Figura 6.** Gráfica total de frecuencia de términos

## 2.2 Filtros

Para hacer el estudio de los diferentes tipos de TokenFilters se ha usado el siguiente texto :

“Sobre los documentos utilizados en la práctica anterior, hacer un estudio estadístico sobre los distintos tokens que se obtienen al realizar distintos tipos de análisis ya predefinidos. Por tanto, será necesario contar el número de términos de indexación así como frecuencias de los mismos en cada documento. Realizar un análisis comparativo entre los distintos resultados obtenidos. “

## LowerCaseFilter

Este filtro convierte todas las palabras en minúscula

LowerCaseFilter.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

sobre los documentos utilizados en la práctica anterior hacer un estudio estadístico sobre los distintos tokens que se obtienen al realizar distintos tipos de análisis ya predefinidos por tanto será necesario contar el número de términos de indexación así como frecuencias de los mismos en cada documento realizar un análisis comparativo entre los distintos resultados obtenidos

Figura 7. Texto usando LowerCaseFilter

## StopFilter

Usa una lista de palabras vacías que elimina del análisis. Las palabras son las, los, de, en y la.

StopFilter.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Sobre documentos utilizados práctica anterior hacer un estudio estadístico sobre distintos tokens que se obtienen al realizar distintos tipos análisis ya predefinidos Por tanto será necesario contar el número términos indexación así como frecuencias mismos cada documento Realizar un análisis comparativo entre distintos resultados obtenidos

Figura 8. Texto usando StopFilter

## SnowballFilter

Realiza un steaming con palabras en español

SnowballFilter.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Sobr los document utiliz en la practic anterior hac un estudi estadist sobr los distint tokens que se obtien al realiz distint tip de analisis ya predefin Por tant ser necesari cont el numer de termin de index asi com frecuenci de los mism en cad document Realiz un analisis compar entre los distint result obten

Figura 9. Texto usando SnowballFilter

## ShingleFilter

Selecciona un término y según el rango que le ponga selecciona más o menos, en este caso tres (Sobre | Sobre los | Sobre los documentos)

ShingleFilter.txt: Bloc de notas

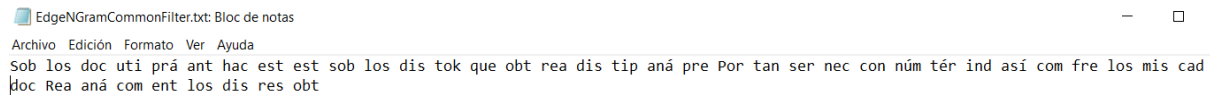
Archivo Edición Formato Ver Ayuda

Sobre Sobre los Sobre los documentos los los documentos los documentos utilizados documentos documentos utilizados documentos utilizados en utilizados utilizados en utilizados en la en la en la en la práctica la la práctica la práctica anterior práctica práctica anterior práctica anterior hacer anterior anterior hacer anterior hacer un hacer hacer un hacer un estudio un un estudio un estudio estadístico estudio estudio estadístico estudio estadístico sobre estadístico estadístico sobre estadístico sobre los sobre sobre los sobre los distintos los los distintos los distintos tokens distintos tokens distintos tokens que tokens tokens que tokens que se que se que se se obtienen se se obtienen se obtienen al obtienen obtienen al obtienen al realizar al al realizar al realizar distintos realizar realizar distintos realizar distintos tipos distintos tipos distintos tipos de tipos tipos de tipos de análisis de de análisis de análisis ya análisis análisis ya predefinidos ya ya predefinidos ya predefinidos Por predefinidos predefinidos Por predefinidos Por tanto Por Por tanto Por tanto será tanto tanto será tanto será necesario será necesario será necesario contar necesario necesario contar necesario contar el contar contar el contar el número el el número el número de número número de número de términos de de términos de términos de términos términos de términos de términos de indexación de de indexación de indexación así indexación indexación así como así así como así como frecuencias como como frecuencias como frecuencias de frecuencias frecuencias de frecuencias de los de de los de los mismos los los mismos los mismos en mismos mismos en mismos en cada en en cada en cada documento cada documento cada documento Realizar documento documento Realizar documento Realizar un Realizar Realizar un Realizar un análisis un un análisis un análisis comparativo análisis análisis comparativo análisis comparativo entre comparativo comparativo entre comparativo entre los entre entre los entre los distintos los los distintos los distintos resultados resultados distintos distintos resultados distintos resultados obtenidos resultados obtenidos obtenidos

Figura 10. Texto usando ShingleFilter

## EdgeNGramCommonFilter

Acorta cada término según la precisión indicada (3) y si tiene menos de 3 se eliminan.



EdgeNGramCommonFilter.txt: Bloc de notas

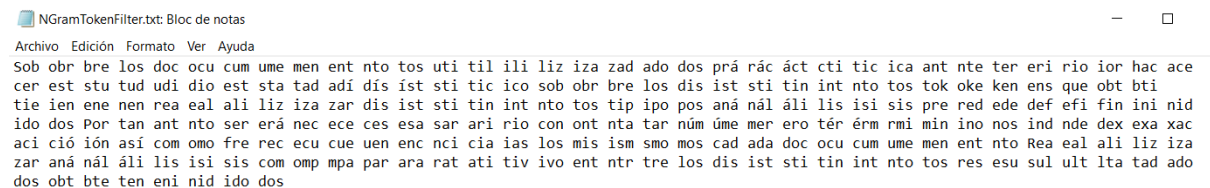
Archivo Edición Formato Ver Ayuda

Sob los doc uti prá ant hac est est sob los dis tok que obt rea dis tip aná pre Por tan ser nec con núm tér ind así com fre los mis cad  
doc Rea aná com ent los dis res obt

Figura 11. Texto usando EdgeNGramCommonFilter

## NgramTokenFilter

Corta la palabra según la precisión indicada (3) y si tiene menos de 3 se eliminan. Cada término es cortado n veces siendo N = número de de caracteres – precisión + 1



NgramTokenFilter.txt: Bloc de notas

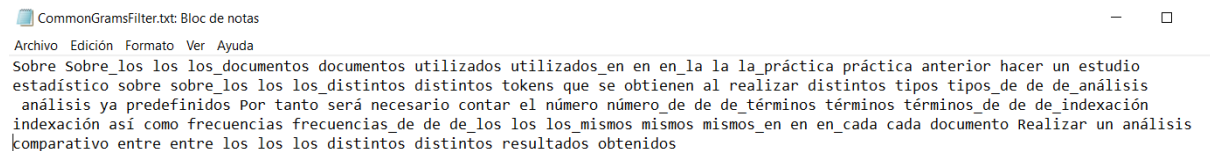
Archivo Edición Formato Ver Ayuda

Sob obr bre los doc ocu cum ume men ent nto tos uti til ili liz iza zad ado dos prá rác áct cti tic ica ant nte ter eri rio ior hac ace  
cer est stu tud udi dio est sta tad adí dís ist sti tic ico sob obr bre los dis ist sti tin int nto tos tok oke ken ens que obt bti  
tie ien ene nen rea eal ali liz iza zar dis ist sti tin int nto tos tip ipo pos aná nál áli lis isi sis pre red ede def efi fin ini nid  
ido dos Por tan ant nto ser erá nec ece ces esa sar ari rio con ont nta tar núm úme mer ero tér érm rmi min ino nos ind nde dex exa xac  
aci ció ión así com omo fre rec ecu cue uen enc nci cia ias los mis ism smo mos cad ada doc ocu cum ume men ent nto Rea eal ali liz iza  
zar aná nál áli lis isi sis com omp mpa par ara rat ati tiv ivo ent ntr tre los dis ist sti tin int nto tos res esu sul ult lta tad ado  
dos obt bte ten eni nid ido dos

Figura 12. Texto usando NgramTokenFilter.

## CommonGramsFilter

Según una lista de palabras crea dos términos nuevos cada vez que salga cada palabra en el texto, un término nuevo con la palabra anterior y otro con la posterior.



CommonGramsFilter.txt: Bloc de notas

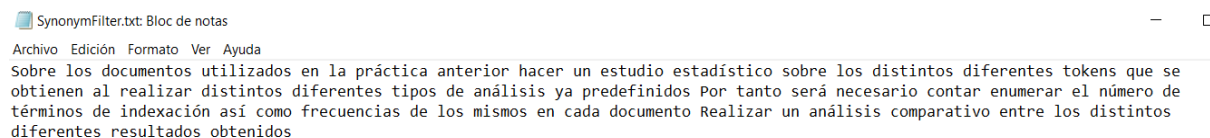
Archivo Edición Formato Ver Ayuda

Sobre Sobre los los los documentos documentos utilizados utilizados en en en la la la la práctica práctica anterior hacer un estudio  
estadístico sobre sobre los los los distintos distintos tokens que se obtienen al realizar distintos tipos tipos de de de análisis  
análisis ya predefinidos Por tanto será necesario contar el número número de de de términos términos términos de de de indexación  
indexación así como frecuencias frecuencias de de de los los los mismos mismos mismos en en en cada cada documento Realizar un análisis  
comparativo entre entre los los los distintos distintos resultados obtenidos

Figura 13. Texto usando CommonGramsFilter.

## SynonymFilter

Usando un mapa de sinónimos aparecerá junto a la palabra del texto su sinónimo como nuevo término nuevo, aunque este no aparezca en el texto.



SynonymFilter.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Sobre los documentos utilizados en la práctica anterior hacer un estudio estadístico sobre los distintos diferentes tokens que se  
obtienen al realizar distintos diferentes tipos de análisis ya predefinidos Por tanto será necesario contar enumerar el número de  
términos de indexación así como frecuencias de los mismos en cada documento Realizar un análisis comparativo entre los distintos  
diferentes resultados obtenidos

Figura 14. Texto usando SynonymFilter.

## 2.3 Custom Analyzer

Se ha creado un analyzer el cuál usa un filtro LowerCase para pasar todo el texto en minúscula, un filtro de StopWords con las palabras las, los, la, en y de en la lista de palabras vacías y por último un filtro SnowballFilter que usa un Stemmer de un conjunto de palabras en castellano.

```
String customAnalyzer() throws IOException {
    String aux = "";

    Analyzer ana = CustomAnalyzer.builder( configDir: Paths.get( first: "D:\\Universidad\\Cuarto\\RI\\Practicas\\Practica2" ))
        .withTokenizer( name: StandardTokenizerFactory.NAME )
        .addTokenFilter( name: LowerCaseFilterFactory.NAME )
        .addTokenFilter( name: StopFilterFactory.NAME, params: "ignoreCase", params: "false", params: "words", params: "stopwords.txt", params: "format", params: "wordset" )
        .addTokenFilter( name: SnowballPorterFilterFactory.NAME, params: "language", params: "Spanish" )
        .build();

    TokenStream stream = ana.tokenStream( fieldName: null, new StringReader( s: contenido ));

    stream.reset();
    while (stream.incrementToken())
        aux += stream.getAttribute( attrClass: CharTermAttribute.class ) + " ";

    stream.end();
    stream.close();

    return aux;
}
```

Figura 15. Función del custom Analyzer.

Por último se puede ver la salida del texto usando este analizador.

customAnalyzer.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

sobr document utiliz practic anterior hac un estudi estadist sobr distint tokens que se obtien al realiz distint tip analisis ya predefin por tant ser necesari cont el numer termin  
|index asi com frecuenci mism cad document realiz un analisis compar entre distint result obten

Figura 16. Texto con el custom Analyzer