

PRACTICA 5:

Indexación y Facetas con Lucene



Alumno: Pablo Valenzuela Álvarez

DNI: 76652136-J

Correo: pvalenzuela@correo.ugr.es

Alumno: Francisco Javier García Maldonado

DNI: 76654015-Y

Correo: franelas@correo.ugr.es

Índice:

1. Análisis del problema
2. Diseño de la solución
3. Manual de usuario

1. Análisis del problema

El objetivo de esta práctica es realizar una búsqueda sobre el índice creado en la práctica anterior.

Para realizar esta búsqueda necesitamos una query y un indexsearcher que se encargue de sacar el resultado de ella.

Los índices sobre los que vamos a hacer la búsqueda ya los tenemos de la práctica anterior.

Por último, necesitamos el archivo jar queryparser de lucene para poder trabajar.

2. Diseño de la solución

Antes de todo declaramos las siguiente variables que tendrá la clase:

- **IndexReader reader:** nos da acceso al índice.
- **IndexSearcher searcher:** nos permite realizar búsqueda sobre el IndexReader.
- **TaxonomyReader taxoReader:** lista las facetas durante la búsqueda.
- **FacetsConfig fconfig:** configuración de facetas
- **FacetsCollector fcollector:** recupera el número de aciertos en a búsqueda por facetas.
- **BooleanQuery bq:** el tipo de query que se usará en la práctica.

Para **inicializar** el índice usamos la función `inicializar(index_dir, facet_dir)` en la que ambos parametros son strings que indican la ruta de los índices de búsqueda y de facetas. Dentro de la función inicializamos los elementos siguientes:

```
public static void inicializar(String index, String facet) throws IOException{

    Directory indexDir = FSDirectory.open(Paths.get(index));
    Directory taxoDir = FSDirectory.open(Paths.get(facet));
    reader = DirectoryReader.open(indexDir);
    searcher = new IndexSearcher(reader);
    taxoReader = new DirectoryTaxonomyReader(taxoDir);
    fconfig = new FacetsConfig();
    fcollector = new FacetsCollector();
    //fcollector = new FacetsCollector(true); //asi almacenamos los scores
}
```

Para **realizar la búsqueda** usamos la siguiente función:

```
77 public static TopDocs busqueda(String field, String query, int tam) throws ParseException, IOException{
78
79     QueryParser parser = new QueryParser(field, new Analizador()); //pongo mi super-analizador
80     StringTokenizer str = new StringTokenizer(query);
81     List<Query> lq = new ArrayList<>();
82
83     while (str.hasMoreElements()) {
84         String nextElement = (String)str.nextElement();
85         Query q = parser.parse(nextElement);
86         if(q.toString().length() != 0){
87             //System.out.println(q.toString());
88             lq.add(q);
89         }
90     }
91
92     BooleanQuery.Builder constructor = new BooleanQuery.Builder();
93     for (Query queryl : lq) {
94         constructor.add(queryl, BooleanClause.Occur.MUST);
95     }
96     bq = constructor.build();
97
98     fcollector = new FacetsCollector();
99     return FacetsCollector.search(searcher, bq, tam, fcollector);
100 }
```

Donde field es el campo sobre el cual buscamos, query es la consulta que hacemos y tam es el límite de documentos recuperados.

Como se puede observar de la línea 78 a 89, se hace el proceso de crear la query. Primero creamos un QueryParser donde le pasamos el Analizados que se usó en la práctica anterior; seguido dividimos la consulta en tokens, los parseamos y los añadimos a una lista.

En la línea 91 creamos el BooleanQuery y la vamos añadiendo los elementos de la lista.

Se retorna la colección con todos los documentos recuperados que satisfacen la consulta.

Para hacer Drill-Down usamos esta función:

```
139 public static TopDocs hacerDrillDown(String faceta, String query, int tam) throws ParseException, IOException{
140     //System.out.println(bq.toString());
141     FacetsCollector fcollector = new FacetsCollector();
142     DrillDownQuery ddq = new DrillDownQuery(fconfig, bq);
143     ddq.add(faceta, query);
144     //System.out.println(ddq.toString());
145     return FacetsCollector.search(searcher, ddq, tam, fcollector);
146 }
```

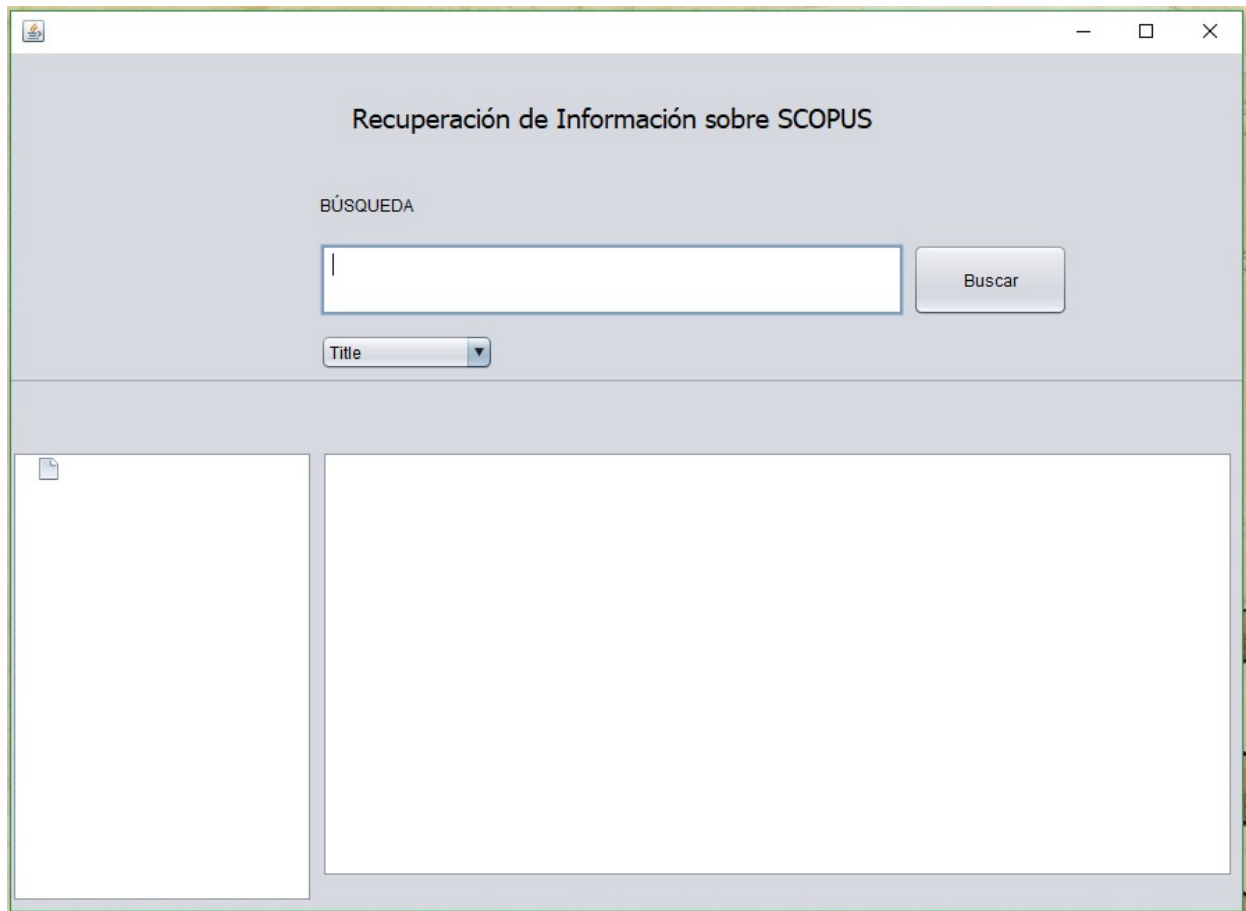
Donde *faceta* es la faceta por al que clasificamos y *query* es el valor de esa faceta.

Se usa DrillDownQuery con los parámetros fconfig y bq, y añadimos los dos campos (faceta y query) para hacer la clasificación.

Se retorna un colección de documentos que se satisfacen la consulta, pero ahora haciendo la clasificación drill-down.

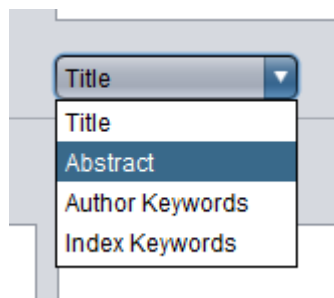
3. Manual de usuario

Iniciamos la aplicación y se muestra esta pantalla:



The screenshot shows a window titled "Recuperación de Información sobre SCOPUS". Inside the window, there is a section labeled "BÚSQUEDA" (Search). Below this label, there is a text input field with a cursor inside, and a button labeled "Buscar" (Search) to its right. Below the input field, there is a dropdown menu currently showing "Title". The main area of the window is divided into two large empty rectangular boxes, one on the left and one on the right, intended for displaying search results.

Podemos observar tres grandes espacios en blanco, el de arriba lo usaremos para escribir la consulta, y los de abajo nos mostrarán los resultados, uno las facetas o categorías y otro nos mostrará el score del documento seguido de su título. También observamos un campo desplegable que contiene los campos por los se realizará la búsqueda.



Vamos a mostrar su funcionamiento con un ejemplo, escribiremos "population of pets" en el area de busqueda, seleccionaremos "Abstract" (es un pequeño resumen) y pulsaremos el botón buscar.



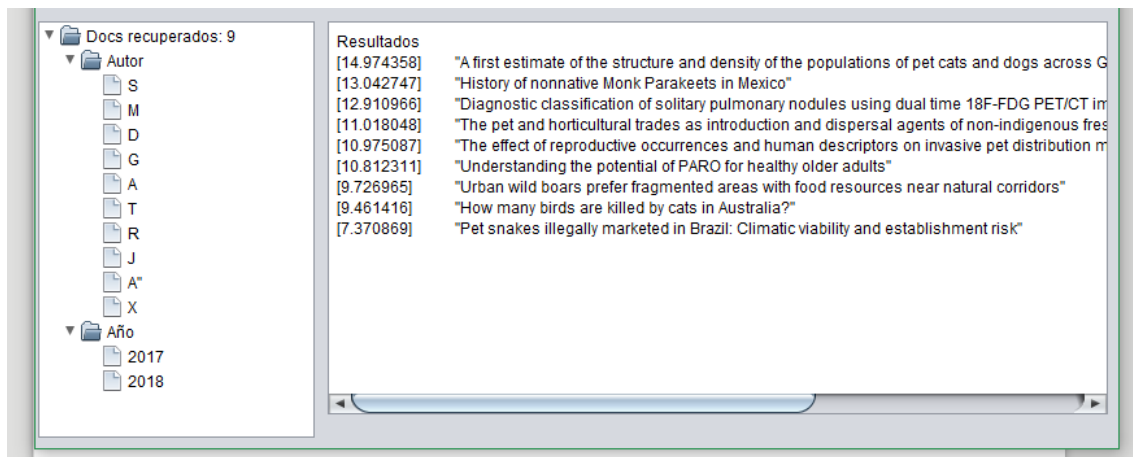
BÚSQUEDA

population of pets

Buscar

Abstract ▼

Estos son los resultados obtenidos:



Docs recuperados: 9

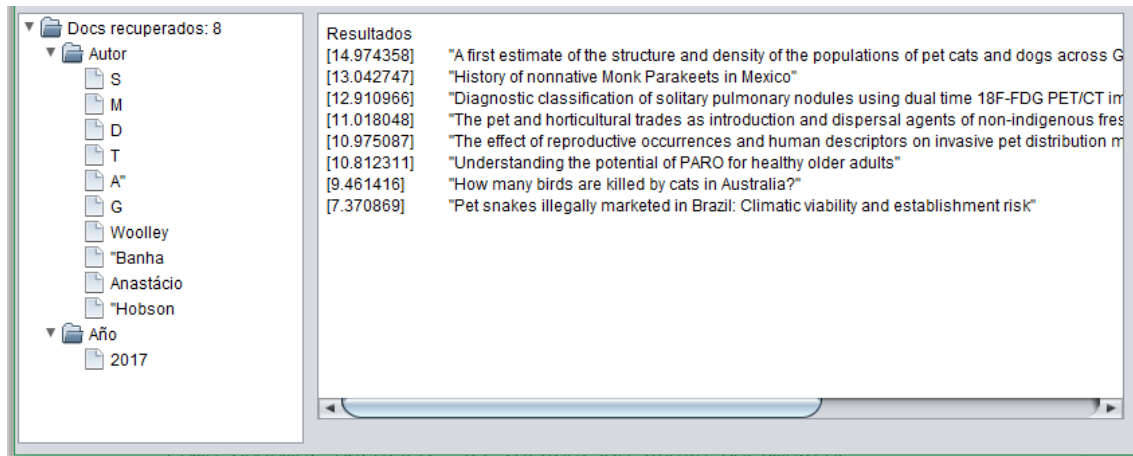
- Autor
 - S
 - M
 - D
 - G
 - A
 - T
 - R
 - J
 - A*
 - X
- Año
 - 2017
 - 2018

Resultados

Score	Title
[14.974358]	"A first estimate of the structure and density of the populations of pet cats and dogs across G
[13.042747]	"History of nonnative Monk Parakeets in Mexico"
[12.910966]	"Diagnostic classification of solitary pulmonary nodules using dual time 18F-FDG PET/CT im
[11.018048]	"The pet and horticultural trades as introduction and dispersal agents of non-indigenous fres
[10.975087]	"The effect of reproductive occurrences and human descriptors on invasive pet distribution m
[10.812311]	"Understanding the potential of PARO for healthy older adults"
[9.726965]	"Urban wild boars prefer fragmented areas with food resources near natural corridors"
[9.461416]	"How many birds are killed by cats in Australia?"
[7.370869]	"Pet snakes illegally marketed in Brazil: Climatic viability and establishment risk"

Como podemos observar, ha recuperado nueve documentos. Nos muestra las facetas que van asociadas a esos documentos, tanto los autores como los años, y al lado, los documentos ordenados de mayor a menor por su score.

Ahora vamos a pulsar de la lista de facetas el año 2017.



Al hacer esto hemos hecho drill-down en la búsqueda, y como podemos observar hay menos documentos obtenidos, pero los que estan son los que tienen la categoría de año 2017.

Si ahora seleccionamos al autor "Woolley":



Nos recupera un solo documento ya que ahora filtra por ese autor, y no debe haber muchos más "Woolley" en nuestra colección.

Para resetear la búsqueda formulamos una consulta nueva y volvemos a empezar.