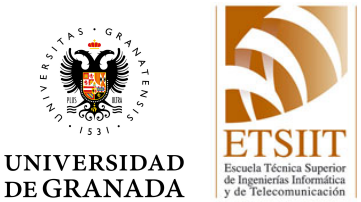
UNIVERSIDAD DE GRANADA
E.T.S.I INFORMÁTICA Y TELECOMUNICACIÓN



Recuperación de Información (RI)

Implementación de un Sistema de Recuperación de Información utilizando Lucene

Parte A. Indexación

Curso 2017-2018

Cuarto Curso del Grado en Ingeniería Informática

María Camarero Granados

Javier Gómez Luzón

Francisco Porcel Molina

Índice

Análisis previo de los requisitos

Se nos pide implementar un programa que permita añadir documentos a un índice creado utilizando la biblioteca Lucene, que posteriormente formara parte de un Sistema de Recuperación de Información que servirá para buscar artículos ciéntificos.

Además, se debe utilizar Luke para ver el índice y realizar distintas consultas sobre él para demostrar que se ha creado de forma adecuada.

Interfaz de búsqueda

Se debe saber qué y cómo se va a buscar en nuestra interfaz de búsqueda para así saber qué campos de los documentos con los que se trabaja y de qué forma van a ser tenidos en cuenta para elaborar el índice.

Se deben identificar, al menos, los siguientes tipos de campos sobre los documentos de entrada:

- StringField: Texto simple que no se tokeniza. Útil para la búsqueda por facetas y para el filtrado de consultas.
- TextField: Términos que son procesados para la indexación.
- Numérico.
- Facetas (Categorías)

Además, en la aplicación se deben poder realizar:

- Consultas booleanas que involucre a los operadores lógicos OR, AND o NOT.
- Consultas avanzadas (consultas por proximidad, por ejemplo)
- Presentaciones de la información utilizando distintos criterios de ordenación.

Analizador

Hay que identificar qué términos serán utilizados en la búsqueda y cuales no.

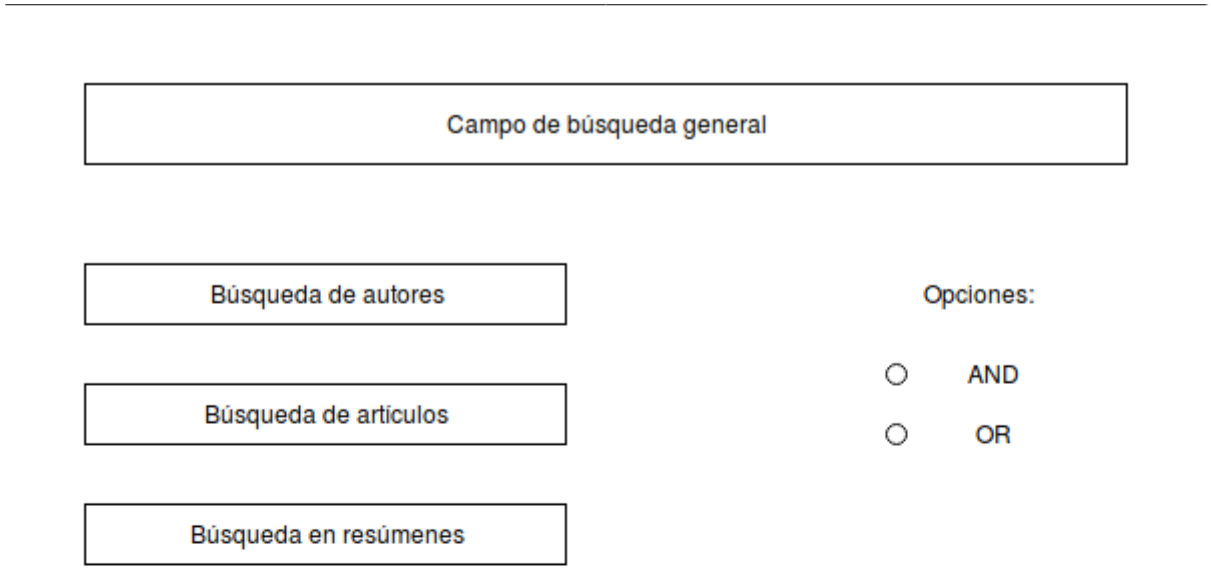
Para ello, se debe seleccionar entre los tipos de analizador que tiene implementados Lucene el más adecuado para la aplicación, justificando la decisión. Es posible que haya que utilizar un analizador distinto para cada uno de los campos a indexar.

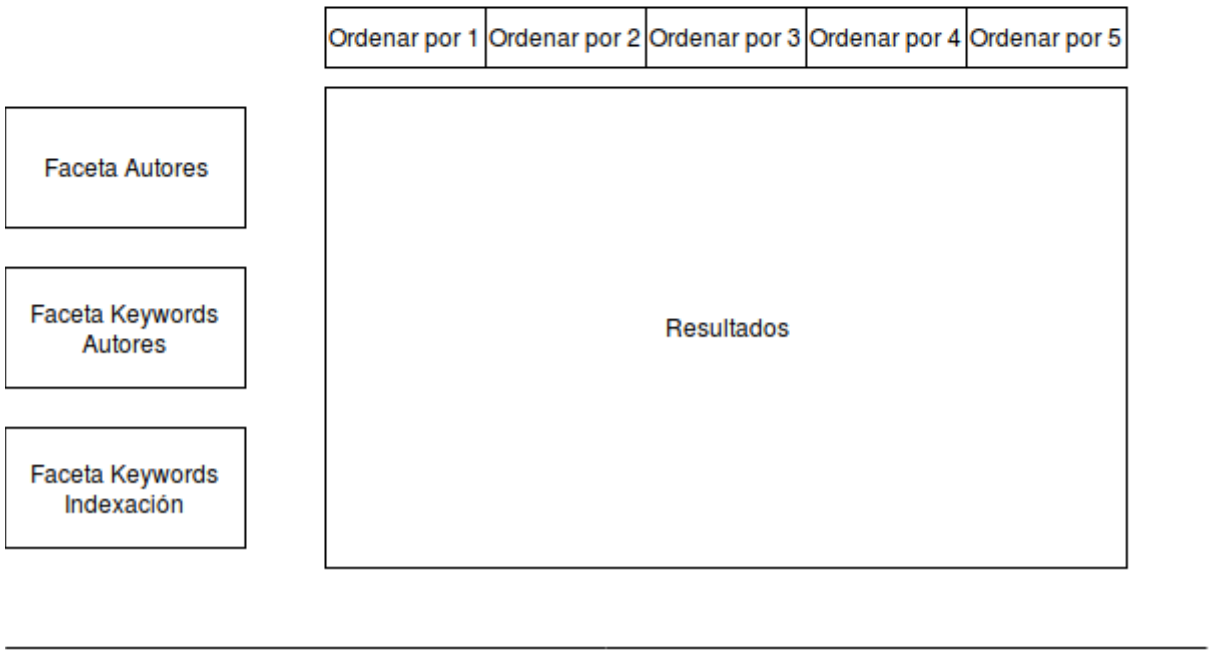
Facetas

Se deberá realizar la búsqueda por facetas. Para conseguirlo, se deben identificar los campos por los que se podrá clasificar los documentos.

Además, como resultado de la búsqueda podremos tener los resultados agrupados por categorías.

Diseño de la solución





Nuestra interfaz tendrá un aspecto similar al de la imagen, y, en consecuencia, deberemos implementar la creación del índice de una determinada manera, que describimos a continuación.

En nuestra solución, incluimos todos los campos de los que constan los registros sobre artículos científicos de la base de datos Scopus que hemos utilizado.

De esta forma, podemos devolver toda la información y no nos vemos limitados a la hora de trabajar más adelante con los distintos tipos de consultas.

Para los campos en los que tenemos previsto realizar una búsqueda general o realizar una búsqueda por campo específico utilizamos el analizador EnglishAnalyzer, debido a que tanto el título, como el resumen o los distintos keyboards se encuentran en inglés y deben ser tokenizados para facilitar que nuestras búsquedas obtengan buenos resultados (EnglishAnalyzer se ocupa de eliminar palabras vacías, hacer stemming, etc.).

Para el campo EID utilizamos el KeywordAnalyzer para que se indexe la cadena de texto sin dividirla de ningún modo. Esto se hace porque normalmente este campo sólo va a ser usado cuando se conoce el EID de un libro concreto y, por lo tanto, se introduce tal cual en el buscador.

Por último, utilizamos el StandardAnalyzer para el resto de campos, por si es necesario buscar artículos a partir de ellos.

Esta asignación de analizadores se realiza mediante PerFieldanalyzerWrapper en la función cargaAnalyzer.

Hemos seleccionado como facetas los autores de los artículos, las keywords de los autores y las keywords para indexar. Estos valores nos serán muy útiles para clasificar la información que nos devuelvan las búsquedas, ya que tienen un gran poder de agrupación y, además, nos permitirán visualizar la información de forma interesante.

Manual de usuario

Para la implementación del programa que se ocupa de crear el índice se ha utilizado el IDE NetBeans. El mismo se puede utilizar para su ejecución de forma sencilla, para lo cual hace falta especificar como argumento la ruta de la carpeta que contiene los archivos (o la ruta del archivo en el caso de que haya sólo uno).

Para el uso de Luke para la realización de consultas, basta con ejecutarlo y, a continuación, especificar la ruta del índice. Una vez hecho esto, accedemos a la pestaña Search y, una vez ahí, escribimos la expresión deseada además de especificar sobre qué campo queremos realizar la búsqueda y el analizador a utilizar.

En la siguiente imagen podemos observar como ejemplo la búsqueda de la palabra “twitter” en el campo Abstract utilizando el analizador EnglishAnalyzer: