PRACTICA 4:
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 previo de los requisitos

El objetivo de esta práctica es realizar una aplicación que cree un índice que contendrá documentos científicos, estos documentos tendrán campos y facetas que facilitarán la posterior búsqueda dada una consulta.

Están almacenados en ficheros csv, en los cuales cada línea es un documento, por lo que tenemos que usar diversas funciones para obtenerlas y separar los campos dentro de ella.

Estas colecciones de documentos las hemos obtenido a través de la web SCOPUS. Estos documentos tienen los siguientes campos:

- Autores
- Título
- Año
- Título de la fuente
- Citas
- Links
- Resumen
- Palabras clave del autor
- Palabras clave del índice
- EID

Por otra parte, los índices de facetas contendrán solamente:

- Autores
- Año

2. Diseño de la solución

De acuerdo con el análisis previo, se ha diseñado la siguiente solución para satisfacer todos los requerimientos:

Indexación:

El primer paso que hemos realizado ha sido la creación del índice. Este lo hacemos con la función *obtenerDoc*. Al pasarle varios datos, entre ellos la ruta de la colección a indexar, esta recorrerá el directorio proporcionado de forma recursiva y tratará los ficheros a indexar uno a uno, hasta que no quede ninguno.

Estos datos que tratamos son cada una de las filas de nuestro documento CSV por separado. Para conseguir una buena indexación hemos tratado cada uno de los datos de forma diversa, ya que no es lo mismo un campo numérico que un campo de tipo textual.

Estos analizadores utilizados en cada uno de los datos son los siguientes (por defecto se usa un analizador creado por nosotros que usa el algoritmo de porter para segmentar, filtra letras sueltas, elimina palabras vacías y quita signos de puntuación entre otras cosas):

- **Autores**: tratado como *TextField*, puesto que queremos tokenizar los distintos términos que aparezcan en este campo, ya que habrá varios autores. Así conseguiremos un mayor número de coincidencias. Este campo lo almacenaremos en el índice para mostrarlo al usuario.

- **Título**: se indexará como también como tipo TextField ya que queremos realizar el mismo proceso que en el apartado anterior y tokenizar los distintos términos de los títulos y así encontrar nuevamente mayor número de coincidencias posibles. Este campo también lo almacenaremos en el índice para mostrarlo posteriormente al realizar las búsquedas.

- **Año**: el año de la publicación se indexará como tipo IntPoint. Para este campo se ha elegido este tipo ya que es de tipo numérico y como tal deben ser tratados de forma diversa. Este campo también lo almacenaremos en el índice ya que su información es bastante relevante a la hora de hacer las consultas

- **Título de la fuente**: este campo también lo indexaremos como tipo TextField, ya que también es de tipo textual. Y lo almacenaremos en el índice ya que también es importante para realizar búsquedas por esta categoría.

- **Citas**: nuevamente trataremos este campo como IntPoint ya que vuelve a ser un campo de tipo numérico y como tal debe ser tratado de forma diversa a los campos textuales. Finalmente tendremos que hacerle otra pasada como StoredField para poder terminar de procesar los datos que no sean numéricos y no dejarlos de lado. En este usaremos el analizador Standard para evitar que nos elimine los que tienen de 0 a 9 citas (el analizador por defecto elimina los tokens de 1 carácter).

- **Links**: los enlaces que nos encontremos en estos documentos son importantes, ya que normalmente suelen ser referencias a otros artículos u a citas. Por lo que estos deben tratarse como campos de tipo TextField. En este caso usaremos el analizador UAX29 por que nos interesa que nos guarde el link entero.

- **Resumen**: un resumen como tal, al ser el pequeño fragmento de la publicación debe contener palabras clave para poder realizar las búsquedas por lo que para su indexación también será tratado como tipo TextField.

- **Palabras clave del autor**: estas serán las palabras clave por las que el autor quiere que se encuentre su artículo al realizar las búsquedas, por lo que estas al ser de tipo textual también deberán ser tratadas como tipo TextField.

- **Palabras clave del índice:**

- **EID:** son códigos que se le asigna al documentos y son únicos. Para la indexación serán tratados como textfield. Aqui usaremos el analizador de keywords para que no haga nada y nos lo alamcene la keyword entera.

Una vez indexados todos los documentos junto a sus campos, cerramos el índice y finalizamos el proceso de indexación.
El índice, lo almacenamos en una carpeta llamada /resultados/índice. Esto nos permitirá no crear el índice cada vez que queramos realizar una búsqueda.

En segundo lugar, tendremos que realizar la creación de Facetas, para poder realizar búsquedas con respecto a ellas. Las facetas que vamos a utilizar en nuestra solución serán solamente dos, pero no por ello menos importantes. Estas serán autores y año.

Para indexar estas facetas tenemos que hacer los siguiente:
Mientras se está creando el documento, le pasamos a un método una lista con todas las facetas, en ese método debemos tokenizarlas y "arreglarlas" un poco (quitar signos de puntuación, …), y por último insertarlas en el mismo documento como si fuera un campo más.

3. **Manual de usuario**

Para realizar la indexación ejecutaremos nuestro programa. Y con eso será suficiente (de momento, hasta que creemos la interfaz de usuario en las siguientes prácticas).

Una vez que hemos indexado ejecutando nuestro programa, el siguiente paso será realizar las búsquedas. Estas búsquedas las realizaremos mediante la aplicación Luke. Esta aplicación nos permite seleccionar a través de un botón la ruta del índice que acabamos de indexar. Esto nos permitirá realizar consultas mediante los campos que hemos tratado. Una vez que hemos seleccionado esta ruta tan solo tendremos que seleccionar el valor por el que queremos realizar la búsqueda. Y la cantidad de documentos relevantes que queremos que nos muestre la aplicación. Al escribir esta misma en el cuadro de texto, seleccionar todo lo anterior y darle al botón de buscar nos recuperará toda la información que encuentre relevante a la susodicha.