UNIVERSIDAD DE GRANADA

Gestión de Información en la Web

Práctica 3. Desarrollo de un Sistema de Recuperación de Información con la biblioteca Lucene

Curso 2019-2020

Máster en Ingeniería Informática

Lidia Sánchez Mérida

lidiasm96@correo.ugr.es

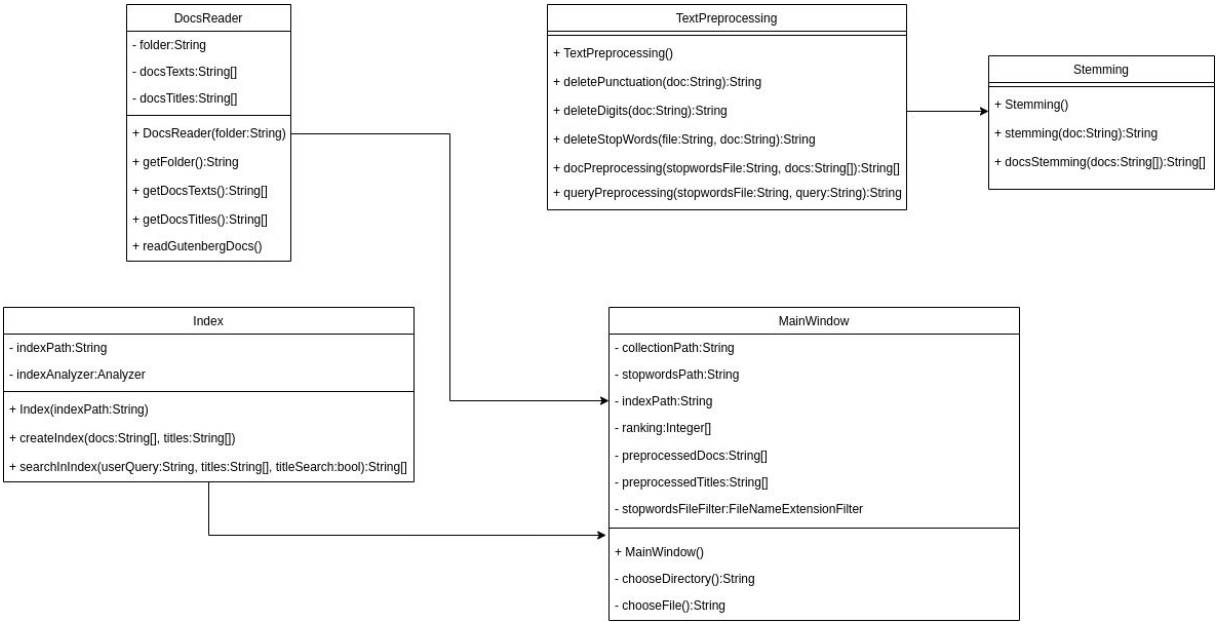
Índice

Descripción del problema

El objetivo de esta práctica consiste en implementar un Sistema de Recuperación de Información haciendo uso de **Lucene**, un conjunto de bibliotecas desarrolladas en Java orientadas a la indexación y la búsqueda de documentos a partir de una consulta particular. Por lo tanto, se debe desarrollar un sistema capaz de crear un **índice** a partir de la colección de documentos escogida, aplicando a sus respectivos textos un procesamiento previo a la formación del índice e indicando finalmente la ruta en la que se guardará el mismo. Por otra parte, se debe implementar un **motor de búsqueda** con el que poder recuperar documentos en función de la consulta realizada haciendo uso del índice previamente creado. Como resultado mostrará un *ranking* con los documentos más relevantes situados al comienzo de la lista y permitirá su visualización a través de una interfaz gráfica.

Desarrollo del sistema

En primer lugar he descargado la biblioteca Lucene desde la página oficial [1] y he añadido a mi proyecto Java los ficheros *jar* correspondientes que se mencionan en el guión de prácticas. A continuación se puede observar el diagrama de clases que representa el diseño del sistema de recuperación de información.



Tal y como se puede comprobar existe una primera clase denominada ***DocsReader*** cuyo principal objetivo es leer la colección de documentos extrayendo tanto su texto como sus correspondientes títulos. En mi caso he optado por hacer uso de la biblioteca ***Project Gutenberg*** [2], de la cual me he descargado 100 ficheros en inglés de diferentes temáticas tales como animales, historia de diferentes países, entre otras. Para que cada una tuviese un número de ejemplos representativo y balanceado, he descargado manualmente 10 libros

de cada temática. La estructura que presenta cada uno de los documentos es similar a la de un libro, es decir, al comienzo aparece la licencia de uso libre de la biblioteca Gutenberg, a continuación el título, el índice y las correspondientes secciones propias de un libro normal y corriente. Cabe destacar que todos los **libros son bastante extensos** por lo que todos los archivos cuentan con una gran cantidad de texto. Asimismo contienen símbolos de todo tipo como comillas, guiones, números, entre otros.

La segunda clase desarrollada se denomina ***TextProcessing*** por la cual se aplica un procesamiento común tanto a los libros como a las consultas que posteriormente se realicen para recuperarlos. Es muy importante aplicar los mismos métodos de procesamiento a ambas entidades para que posteriormente se puedan realizar consultas correctamente. Si bien el analizador estándar de Lucene aplica un cierto procesamiento a los textos, como transformar todas las letras a minúscula [3], he observado en mi caso que tras aplicar el *StandardAnalyzer* los documentos seguían presentando ciertos signos de puntuación, como comillas o guiones. Por ello se aplica un primer método para **eliminar los signos de puntuación** de modo que tras este procedimiento los textos estén libres de cualquier símbolo que no sea un número o una letra.

El segundo método que se ha aplicado consiste en **eliminar los dígitos** puesto que en mi caso no aportan información relevante y por ende, no tiene sentido realizar consultas, por ejemplo, en torno a una página determinada.

A continuación se procede a **eliminar las palabras vacías** haciendo uso en primer lugar del fichero de *stop words* en inglés proporcionado en PRADO. Como método para aplicar este procedimiento he utilizado el analizador de Lucene [4] al que se le proporciona el texto a procesar y un conjunto con las palabras vacías del fichero. Este analizador (*StandardAnalyzer*) es capaz de realizar la tokenización del documento y solo incluir como resultado las palabras que no se clasifiquen como *stop words*.

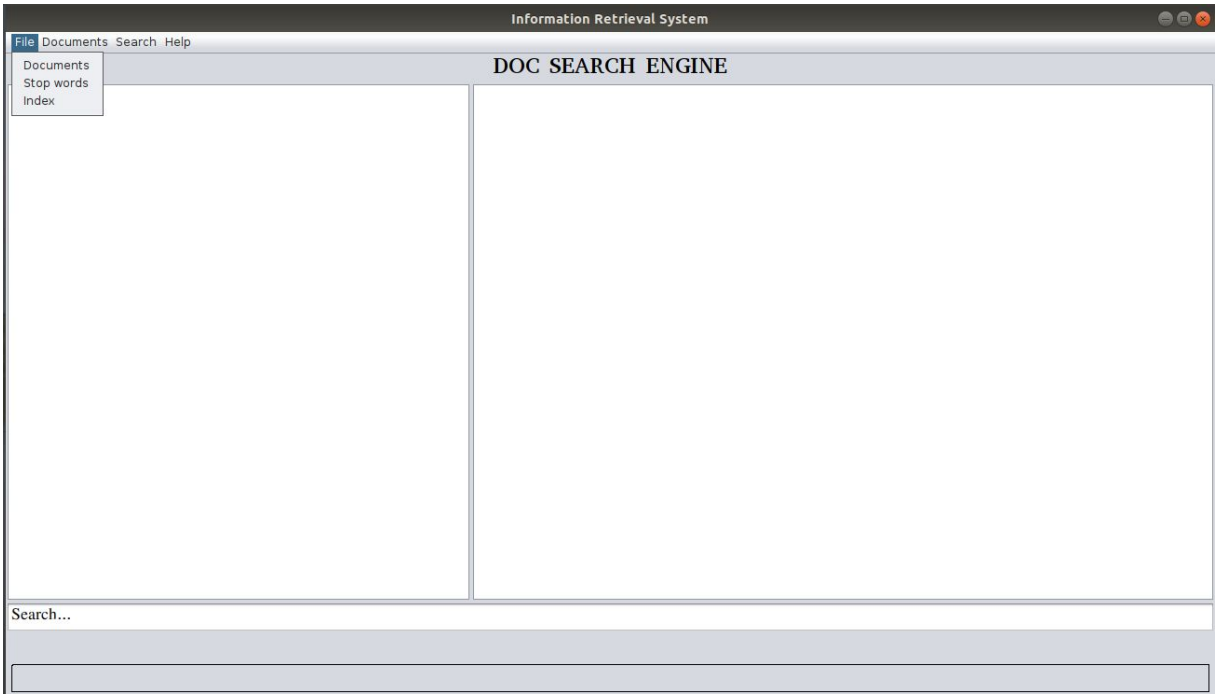
Por último he aplicado la técnica conocida como ***Stemming*** por la cual se extraen las raíces semánticas de las palabras con el objetivo de facilitar la indexación de los documentos almacenando solo las partes más relevantes de los términos. Para aplicar esta técnica he hecho uso, de nuevo, de la biblioteca Lucene [5] que realiza el mismo procedimiento de tokenización que en el caso anterior y además transforma cada término de cada documento en su raíz semántica.

Una vez se preprocesa tanto el texto como los títulos de los libros se procede a crear el índice. Para ello se ha implementado la clase ***Index*** que contiene un método para crear el índice y otro para realizar una búsqueda a partir de los términos que el usuario introduce en el buscador gráfico. El desarrollo de esta clase se ha basado tanto en los ejemplos del guión de prácticas como en este otro más actualizado [6]. El primer método recorre todos los textos y títulos preprocesados de los libros pasados como argumento y crea el correspondiente documento de Lucene con dos campos: *title* para almacenar el título del libro y *doc* para guardar el texto completo del libro.

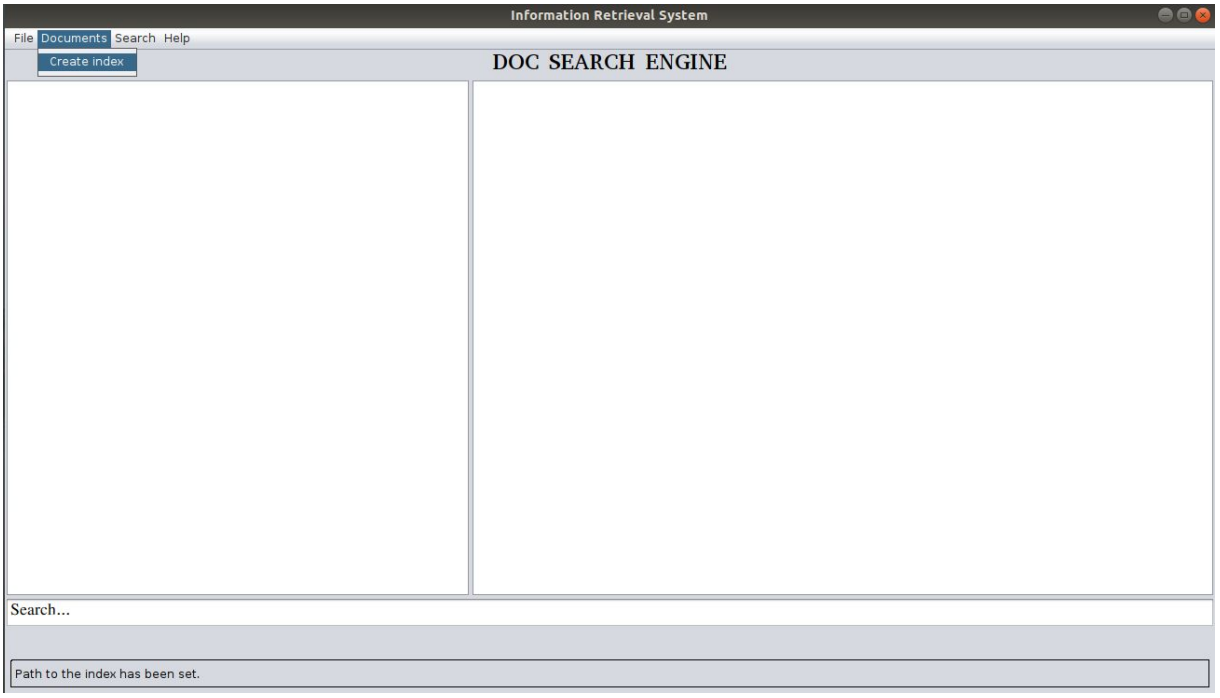
Para el método correspondiente a la consulta y recuperación de documentos se crea en primera instancia un buscador con la biblioteca *Lucene* y se lleva a cabo la consulta utilizando los términos de búsqueda que ha introducido el usuario ya preprocesados. Una vez obtenemos los resultados recuperamos los documentos en orden decreciente en relación a la relevancia con respecto a la consulta utilizando sus títulos y los mostramos por pantalla. La interfaz gráfica del sistema se encuentra en la clase ***MainWindow*** desde la que se pueden llevar a cabo sendas actividades, además de la especificación de los directorios correspondientes a la colección de documentos y al índice así como el fichero con las palabras vacías, en este caso, en inglés.

Manual de usuario

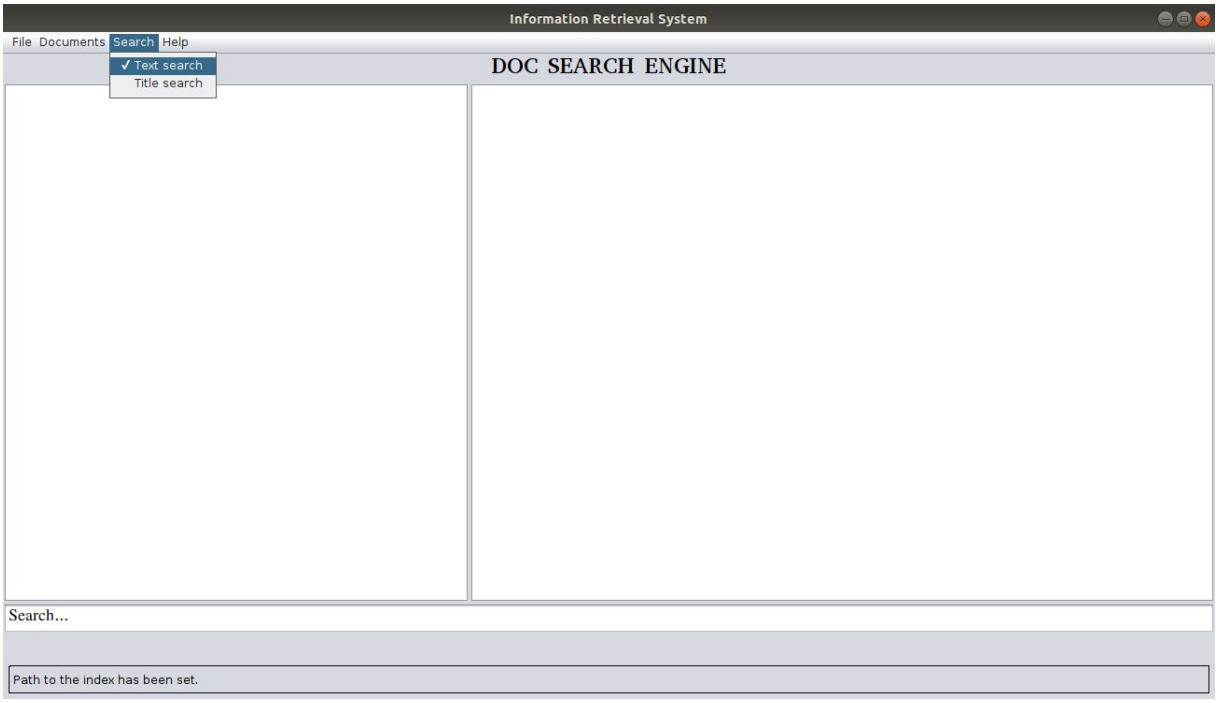
Todas las funcionalidades del sistema son explorables a través de la interfaz gráfica. En primer lugar, tras ejecutar el programa, se deben especificar el directorio donde se encuentran la colección de documentos, el fichero con las palabras vacías y la carpeta donde se almacena el índice mediante las tres correspondientes opciones dentro del elemento *File* situado en el menú superior.



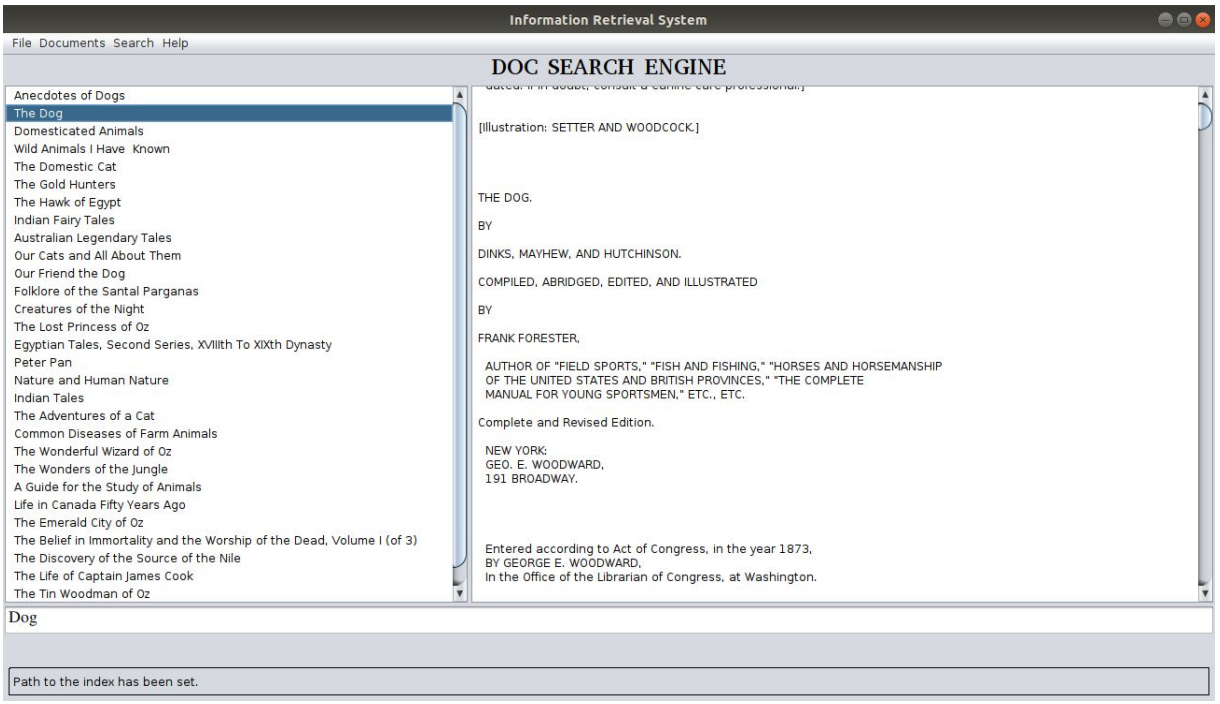
Si la carpeta especificada ya contiene un índice elaborado con *Lucene* entonces se puede comenzar a realizar consultas directamente. Si por el contrario el índice no está creado se deberá pulsar sobre *Documents/Create Index* situado también en el menú superior y esperar a que haya terminado el proceso.



Se pueden realizar dos tipos de consultas. En el menú superior existe una opción denominada *Search* por la cual podemos especificar que las búsquedas se realicen en torno a los textos de los libros o a sus títulos. Por defecto el sistema realiza las consultas en base al texto completo del libro. No obstante se puede cambiar esta preferencia clicando sobre *Title Search,* para una búsqueda por títulos, o *Text Search* para consultas exploratorias considerando todo el documento.



Para enviar una consulta basta con escribir los términos de búsqueda deseados y pulsar la tecla *Enter* como se realizaría con cualquier motor de búsqueda convencional. Una vez se hayan obtenido los resultados de la consulta se visualizarán en el lateral izquierdo la lista de documentos resultante. Como máximo se mostrarán un total de hasta 30 documentos. Si deseamos visualizar alguno de ellos basta con seleccionarlo y su contenido aparecerá en la sección lateral derecha.



Bibliografía

1. Apache, *Lucene Downloads*, https://lucene.apache.org/core/downloads.html
2. *Project Gutenberg,* https://www.gutenberg.org/wiki/Main_Page
3. Baeldung, *Guide to Lucene Analyzers*, 2018, https://www.baeldung.com/lucene-analyzers
4. Stackoverflow, *Tokenize, remove stop words using Lucene with Java*, https://stackoverflow.com/questions/17625385/tokenize-remove-stop-words-using-lucene-with-java
5. Stackoverflow, *Stemming english words using Lucene 6,* https://stackoverflow.com/questions/43621672/stemming-english-words-using-lucene-6
6. Lokesh Gupta, HowToDoInJava, *Lucene Tutorial - Index and Search Examples*, https://howtodoinjava.com/lucene/lucene-index-search-examples/