

Propiedad Intelectual

Universidad de Granada

Breve descripción

Desarrollo de un Sistema de Recuperación deInformación con Lucene, Solr o ElasticSearch

Autor

Álvaro de la Flor Bonilla (alvdebon@correo.ugr.es) 15408846-L

Práctica 1

Gestión de Información en la Web

**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**

**Máster Profesional en Ingeniería Informática**

**Curso 2020/2021**

# Resumen

En algún momento nos puede surgir la necesidad de desarrollar un sistema de recuperación de información. Aunque podemos partir de cero, existe un gran número de bibliotecas en diferentes lenguajes de programación que nos pueden permitir montar el software de búsqueda de una forma rápida y sencilla.

Este es el caso de Lucene, el cual es un conjunto de bibliotecas, escritas en Java(nativo), C++, C#, PHP, Python, Ruby o Perl, que nos facilitará la confección de aplicaciones para realizar la indexación de grandes volúmenes de documentos y recuperación a partir de consultas suministradas por los usuarios del sistema.

En esta práctica se construirá un sistema de recuperación de información empleando la biblioteca Lucene (en su última versión publicada), compuesto de dos programas:

1. Un indexador, el cual recibirá como argumentos la ruta de la colección documental a indexar, el fichero de palabras vacías a emplear y la ruta donde alojar los índices, y llevará a cabo la indexación, creando los índices oportunos y ficheros auxiliares necesarios para la recuperación. Esta aplicación se ejecutará en la línea de mandatos y no tendrá ningún componente gráfico. Este software realizará las tareas de tokenización, eliminación de palabras vacías y extracción de raíces antes de crear el índice.
2. Un motor de búsqueda, que al ejecutarse recibirá como argumento la ruta donde está alojado el índice de la colección y permitirá que un usuario realice una consulta de texto y obtenga el conjunto de documentos relevantes a dicha consulta. En este caso, el programa sí será gráfico. Sobre la consulta se realizarán los mismos procesos que sobre los documentos en el indexador.

ÍNDICE DEL PROYECTO

[Resumen 1](#_Toc67522431)

[No se encuentran elementos de tabla de ilustraciones. 4](#_Toc67522432)

[1 Introducción 5](#_Toc67522433)

[1.1 Colecciones 5](#_Toc67522434)

[1.1.1 NSF Research Awards Abstracts 1990-2003 5](#_Toc67522435)

[1.1.2 Wikipedia 5](#_Toc67522436)

[1.1.3 Project Gutenberg eBook 5](#_Toc67522437)

[1.2 Bibliotecas y librerías 5](#_Toc67522438)

[1.2.1 IDE de desarrollo 5](#_Toc67522439)

[2 Descripción del desarrollo 6](#_Toc67522440)

[2.1 Indexador 6](#_Toc67522441)

[2.2 Buscador 6](#_Toc67522442)

[2.3 Configurador 6](#_Toc67522443)

[2.4 Scraping 6](#_Toc67522444)

[3 Manual de usuario 7](#_Toc67522445)

[4 Pruebas 8](#_Toc67522446)

[5 Conclusiones 9](#_Toc67522447)

[6 Bibliografía 10](#_Toc67522448)

íNDICE DE Ilustraciones

[Ilustración 1 - Creación del índice principal 6](#_Toc67521576)

íNDICE DE tablas

# No se encuentran elementos de tabla de ilustraciones.

# Introducción

## Colecciones

Para la realización de este trabajo se han utilizado tres colecciones de datos diferentes.

### NSF Research Awards Abstracts 1990-2003

En concreto se ha utilizado el paquete *“*[*Abstracts\_Part1.zip*](https://kdd.ics.uci.edu/databases/nsfabs/Part1.zip)*”*  y dentro de este el directorio *“./Part1\awards\_1994\awd\_1994\_96*”.

Esta colección tiene 288 elementos.

### Wikipedia

Colección de artículos, disponibles en el directorio *“*[*http://www.search-engines-book.com/collections/*](http://www.search-engines-book.com/collections/)*”*, en concreto se ha utilizado el archivo *“Wiki Large (tar.gz)”*.

Esta colección tiene 2000 elementos.

### Project Gutenberg eBook

Se ha realizado un pequeño script para descargar los últimos libros disponibles en “[*https://www.gutenberg.org/browse/scores/top*](https://www.gutenberg.org/browse/scores/top)*”.* En concreto, en la respectiva sección será explicado este pequeño programa.

Esta colección tiene 100.

## Bibliotecas y librerías

Para la realización de este proyecto ha sido utilizado *“Lucene”* en su versión 8.8.1 siendo las bibliotecas utilizadas:

1. lucene-analyzers-\*-8.8.1.jar
2. lucene-core-8.8.1.jar
3. lucene-queryparser-8.8.1.jar

### IDE de desarrollo

Para el desarrollo de la práctica hemos utilizado IntelliJ IDEA 2020.2.

# Descripción del desarrollo

Podemos destacar 3 secciones principales en el desarrollo. En concreto destacamos la construcción del módulo **indexador**, encargado de generar el índice a partir de la colección de datos que seleccione el usuario. Este indexador es utilizado por el módulo **buscador**, que será quien reciba las consultas del usuario y las ejecute sobre el índice creado anteriormente.

Ambos módulos vienen controlados por un modulo **auxiliar** cuya función será dirigir al usuario entre las distintas funcionalidades que permite realizar este proyecto.

Por último, también debemos destacar el pequeño script que fue desarrollado para destacar la colección de libros en formato HTML de la web *“Project Gutenberg”*.

## Indexador

En primer lugar, para crear el indexador en sí, lo más importante que hemos tenido en cuenta ha sido crear un analizador que cuente con un filtro de palabras vacías.

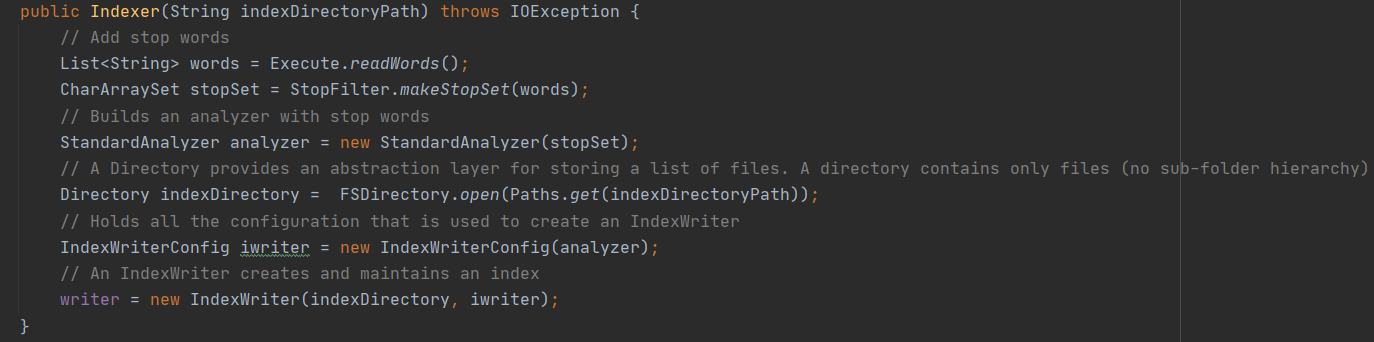


Ilustración 1 - Creación del índice principal

Como dato a tener en cuenta es que existen dos conjuntos de palabras vacías utilizables, una para español y otra para inglés. Este parámetro es configurable por el usuario (después será explicado más en detalle en el módulo de configuración).

Una vez creado el indexador (que a su vez contiene el analizador) simplemente hemos usado el *“writer”* generado para ir añadiendo cada documento de la colección a nuestro índice.

En lo que queda de código que no ha sido mostrado *(“Indexer.java”)* se realiza lo que se comenta el en párrafo anterior, el análisis y adición en el índice en un bucle de cada elemento de la colección.

Por otro lado, se utiliza una clase auxiliar para ejecutar el indexador *(“ExecuteIndex.java”)*, en la que se configura como parámetros el directorio tanto del índice como de la colección. También se realiza un cálculo del tiempo empleado en la realización del indexado.

## Buscador

#TODO

### Umbral

## Configurador

#TODO

## Scraping

#TODO

# Manual de usuario

#TODO

# Pruebas

#TODO

# Conclusiones

#TODO

# Bibliografía

#TODO