
Práctica 3:
Implementación de un Sistema de
Recuperación de Información con Lucene.

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



**UNIVERSIDAD
DE GRANADA**

Departamento de Ciencias de la
Computación e Inteligencia Artificial

Recuperación de Información (2020-2021)

Daniel Bolaños Martínez
Fernando de la Hoz Moreno
Grupo 10 - Martes 11:30h

Índice

| | |
|--|-----------|
| 1. Introducción. | 3 |
| 2. Preprocesado. | 3 |
| 3. Indexación. | 4 |
| 3.1. Clase MiAnalizador. | 4 |
| 4. Búsqueda. | 5 |
| 4.1. Consulta genérica. | 6 |
| 4.2. Consulta booleana. | 7 |
| 4.3. Consulta por tamaño exacto y rango. | 8 |
| 5. Facetas. | 9 |
| 5.1. Indexación. | 9 |
| 5.1.1. Clase RangosTam. | 10 |
| 5.2. Búsqueda. | 10 |
| 5.2.1. Mostrar las facetas. | 10 |
| 5.2.2. Filtrar por facetas. | 12 |
| 6. Pruebas realizadas. | 14 |
| 6.1. Interfaz por terminal. | 14 |
| 6.2. Interfaz gráfica. | 16 |
| 6.2.1. Búsqueda simple. | 16 |
| 6.2.2. Búsqueda avanzada. | 19 |
| 7. ¿Cómo ejecutar el buscador? | 21 |
| 8. Trabajo en Grupo. | 22 |
| Referencias. | 22 |

1. Introducción.

El objetivo de la práctica es la creación de un Sistema de Recuperación de Información haciendo uso de la biblioteca Lucene. Extraeremos los archivos del conjunto de datos *COVID-19 Open Research Dataset Challenge (CORD-19)* que contiene más de 100.000 publicaciones científicas sobre COVID-19 con diferentes campos de interés para cada artículo (Autores, Países, Instituciones, etc). [2]

Indexaremos cada uno de estos campos en un índice que posteriormente nos será útil para realizar diferentes consultas sobre la base de datos obtenida. Finalmente, utilizaremos las herramientas que nos ofrecen las facetas para realizar una búsqueda específica sobre el Sistema de Recuperación de Información diseñado.

2. Preprocesado.

El conjunto de datos que utilizaremos para la creación de nuestro buscador vienen en formato JSON y contienen la información que utilizaremos para la construcción de nuestro SRI. Por ello, es necesario realizar un preprocesado de los archivos antes de indexarlos.

Para el preprocesado, hemos hecho uso del paquete **json-simple-1.1.1.jar** que nos permite extraer los campos de un archivo JSON como cadenas de caracteres e iterar sobre las listas contenidas en el propio formato.

El constructor de la clase **JsonReader** recibe como parámetro un archivo JSON y a partir de él, extrae en formato *String*, información relativa a:

- **Tamaño:** tamaño del archivo en KB.
- **Título:** título del artículo. Unknown si es vacío.
- **Abstract:** abstract del artículo. Unknown si es vacío.
- **Texto:** texto completo del artículo. Unknown si es vacío.
- **Autores:** lista de autores (nombre y apellidos) que firman el artículo.
- **Países:** lista de países a los cuales pertenecen los autores del artículo.
- **Instituciones:** lista de instituciones participantes en el artículo.

3. Indexación.

Para la indexación, hemos creado la clase **Index** que contiene las funcionalidades necesarias para crear un índice a partir de los archivos JSON una vez preprocesados. En esta clase, crearemos tanto el índice como las facetas, pero nos centraremos en la indexación ya que hablaremos de las facetas en su propio apartado.

Primero configuramos el índice con el método **configurarIndice**, el cual se encarga de inicializar los valores de *IndexWriterConfig*, el cual usamos para escribir el índice con los valores por defecto de similaridad *ClassicSimilarity* y analizador *WhitespaceAnalyzer*. Además, hemos creado un analizador propio con distintos flags que aplicaremos a campos específicos en lugar de usar el analizador por defecto. Después guardaremos el índice generado en el directorio *./index*, el cual crearemos si no existe previamente.

Finalmente, usaremos el método **indexarDocumentos** para indexar los documentos de uno en uno utilizando la clase **JsonReader** antes descrita, todo ello como variables temporales, evitando almacenar toda la colección de archivos y borrando la información de cada documento una vez indexado. Para cada artículo crearemos un objeto *Doc* donde guardaremos cada campo con la función *doc.add()* y especificaremos los campos que queramos almacenar, en nuestro caso almacenaremos todos menos el campo *Texto* por su gran extensión.

Una vez almacenados todos los campos del artículo, añadiremos el documento al índice utilizando la función *writer.addDocument(doc)*. Por último, una vez indexados todos los documentos, guardaremos los cambios y cerraremos el índice con los métodos *writer.commit* y *writer.close*.

3.1. Clase MiAnalizador.

Consiste en un analizador con diferentes flags para hacerlo multifunción dependiendo de si los parámetros toman valores true o false. El analizador implementa las siguientes características:

Posibilidad de elegir entre dos *tokenizer*:

- **StandardTokenizer**: Por defecto.
- **UAX29URLEmailTokenizer**: Para campos susceptibles a tener emails o urls, como el *Texto*.

Aplicaremos en todos los casos **LowerCaseFilter** y daremos la posibilidad de usar los siguientes analizadores:

- **AniosFilter:** Implementa un analizador propio contenido en la clase **AniosFilter**. Consiste en un filtro para eliminar los números decimales separados por puntos o comas y aquellos números que sean distintos de 3 y 4 cifras. De esta forma, nos quedaremos únicamente con los números que sean susceptibles a ser años que nos ofrezcan información relevante.
- **SynonymFilter:** Filtro de sinónimos que asigna las mismas referencias en el índice a las palabras que determinemos en el diccionario de sinónimos. El diccionario podría hacerse todo lo complejo que se quiera.
- **StopFilter:** Filtro de palabras vacías, en particular especificaremos el idioma y se aplicará a un conjunto de palabras para ese idioma. Por defecto usaremos el inglés.
- **SnowballFilter:** Filtro de *stemming* para el idioma que especifiquemos en el parámetro. Por defecto usaremos el inglés.

4. Búsqueda.

Una vez realizado el proceso de indexación, crearemos la clase **Busqueda** que nos dará las funcionalidades necesarias para realizar diferentes consultas sobre el índice creado.

El proceso de búsqueda puede ser muy complejo ya que podemos realizar consultas usando desde una única palabra, hasta búsquedas por campo o incluso aplicando lógica booleana sobre varios campos. La clase **Busqueda** lee desde el directorio que contiene nuestro índice y asigna una medida de similitud (por defecto, BM25) para establecer como se comparan documentos.

También hemos creado el método **mostrarDocs** que recibe como parámetro un objeto *TopDocs* que contiene los documentos obtenidos para una consulta específica e imprime por pantalla un resumen con la información más relevante de los documentos obtenidos.

Finalmente la función *indexSearch* crea una interfaz por terminal que implementa las siguientes funcionalidades:

4.1. Consulta genérica.

La consulta consiste en un campo y una palabra que será el origen de la búsqueda. Buscamos todos los documentos que contengan dicha palabra en el campo especificado.

Usaremos la función **QueryParser** para obtener la consulta que queremos realizar a la que le aplicamos el campo y el analizador, parseamos el *query* resultante y realizamos la búsqueda obteniendo el número de documentos que indiquemos.

Podemos realizar la búsqueda sobre:

- El texto completo del artículo.
- Sobre otro campo a especificar.

```
¿Cómo desea realizar la consulta genérica?
1: Sobre el texto.
2: Especificar campo/s.
3: Salir.

Opcion: 1

Introduzca la consulta: covid

Se han encontrado 98 documento/s en total. ¿Desea mostrar las facetas? y/n: n

¿Desea mostrar los 98 resultado/s obtenidos? y/n: y
-----
Nombre fichero: 0a11a62372ae38f658c542b42a727f44f85e150e.json
Tamaño: 21
Titulo: Incidence and mortality of pulmonary embolism in COVID-19: a systematic review and meta-analysis
Autores: Shu-Chen Liao, Shih-Chieh Shao, †, Yih-Ting Chen, Yung-Chang Chen, Ming-Jui Hung
Países: Taiwan, Taiwan
Instituciones: Chang Gung University, Chang Gung University
Abstract: Unknown
-----
Nombre fichero: 0a7f49331e69629979044fde3cf7e5b41a17c1c0.json
Tamaño: 39
Titulo: COVID-19 Induced Economic Uncertainty: A Comparison between the United Kingdom and the United States
Autores: Ugur Korkut, Pata
Países: Osmaniye, Osmaniye
Instituciones: Osmaniye Korkut Ata University, Osmaniye Korkut Ata University
Abstract: The purpose of this study is to investigate the effects of the COVID-19 pandemic on economic policy uncertainty in the US and the UK
. The impact of the increase in COVID- [...]
-----
Nombre fichero: 0a9ed84b2d6679a8348ec48b48d493b2931998d5.json
Tamaño: 46
Titulo: A benchmark of online COVID-19 symptom checkers
Autores: Nicolas Munsch, Alistair Martin, Stefanie Gruarin, Jama Nateqi, Isselmou Abdrahmane, Rafael Weingartner-Ortner, Bernhard Knapp
Países: Unknown
Instituciones: Unknown
Abstract: A large number of online COVID-19 symptom checkers and chatbots have been developed but anecdotal evidence suggests that their conclusions are highly variable. To our kno [...]
-----
```

Figura 1: Ejemplo de consulta genérica mostrando los resultados.

4.2. Consulta booleana.

Realizamos una consulta sobre varios campos de forma simultánea obteniendo unos resultados más precisos. Realizamos una lectura secuencial para cada campo donde indicaremos las palabras que queramos buscar y almacenamos la consulta como un **createPhraseQuery** sobre un **QueryBuilder** creado a partir del analizador específico para el campo.

Finalmente realizaremos una única consulta con **BooleanClause** a partir de todas las consultas anteriores sobre cada campo e indicaremos con la cláusula *MUST* que queremos una búsqueda booleana equivalente a *AND*.

[illegible]

Figura 2: Ejemplo de consulta booleana mostrando los resultados.

4.3. Consulta por tamaño exacto y rango.

Para este tipo de búsqueda, usaremos la clase *IntPoint* de Lucene. Realmente la función utilizada realiza una búsqueda tanto exacta como por rango del campo *Tamaño*. Dependiendo de si el parámetro de búsqueda es un único número o un intervalo de la forma “min-max”.

Mostramos todos los artículos cuyo tamaño de fichero está en el intervalo especificado. Si sólo introducimos un número usaremos, para evitar replicar código, el mismo número para ambas cotas obteniendo la búsqueda exacta. Para ello usaremos la clase *IntPoint* con la función **newRangeQuery**.

```
¿Cómo desea realizar la consulta?
1: Tamaño exacto.
2: Rango de tamaños.
3: Salir.

Opcion: 2

Introduzca el min del rango: 140
Introduzca el máx del rango: 400

Se han encontrado 33 documento/s en total. ¿Desea mostrar las facetas? y/n: n

¿Desea mostrar los 33 resultado/s obtenidos? y/n: y
-----
Nombre fichero: 0a24f22d8d49dc503c52d43dbc51459fc06fbb14.json
Tamaño: 160
Titulo: Antigenic Subversion: A Novel Mechanism of Host Immune Evasion by Ebola Virus
Autores: G S Mohan, W Li, L Ye, R W Compans, C Yang
Países: Unknown
Instituciones: Unknown
Abstract: In addition to its surface glycoprotein (GP 1,2 ), Ebola virus (EBOV) directs the production of large quantities of a truncated glycoprotein isoform (sGP) that is secrete [...]
-----
Nombre fichero: 0a7f8c6431eb8f8957e882db5d5429abd3b5dd81.json
Tamaño: 196
Titulo: The Influence of Essential Oils on Gut Microbial Profiles in Pigs
Autores: Modestas Ruzauskas, Elena Bartkiene, Arunas Stankevicius, Jurga Bernatoniene, Daiva Zadeike, Vita Lele, Vytaute Starkute, Paulina Zavanaviciute, Juozas Grigas, Egle Zokaityte, Arnoldas Pautienius, Grazina Juodelkiene, Valdas Jakstas
Países: Lithuania, Lithuania, Lithuania, Lithuania, Lithuania, Lithuania, Lithuania, Lithuania, Lithuania, Lithuania, Lithuania, Lithuania, Lithuania
Instituciones: Lithuanian University of Health Sciences, Lithuanian University of Health Sciences, Lithuanian University of Health Sciences, Lithuanian University of Health Sciences, Kaunas University of Technology, Lithuanian University of Health Sciences, Lithuanian University of Health Sciences, Lithuanian University of Health Sciences, Lithuanian University of Health Sciences, Lithuanian University of Health Sciences, Kaunas University of Technology, Lithuanian University of Health Sciences
Abstract: Simple Summary: In recent years, the intake of ultra-processed foods has increased dramatically worldwide. Missing natural foods in the diet raise the need of biologicall [...]
-----
Nombre fichero: 0a1ca73dc5f51aaecd0a451d8031316252603664.json
Tamaño: 175
Titulo: RESPIRATORY VIRAL INFECTION AND ASTHMA
Autores: Kecia N Carroll, Tina V Hartert
Países: USA
```

Figura 3: Ejemplo de consulta por rango de tamaño mostrando los resultados.

5. Facetas.

Una vez realizada la consulta sobre el índice, el Sistema de Recuperación de Información debe permitir un filtrado de los resultados obtenidos. En este momento, la herramienta que nos proporciona el uso de las facetas es la más útil y la que utilizaremos en nuestro proyecto.

La búsqueda por facetas requiere que se añadan a un directorio en tiempo de indexación. Por lo que explicaremos cómo se ha implementado el sistema de facetas tanto en la clase **Index** como en la clase **Busqueda**.

5.1. Indexación.

Aprovecharemos la clase **Index** para crear las facetas durante el proceso de creación del índice. En el método **configurarIndice** declaramos un objeto *FacetsConfig* y otro *DirectoryTaxonomyWriter* que recibirá como parámetro el directorio donde se crearán las facetas, en nuestro caso *./facet* el cual crearemos si no existe previamente.

Declararemos como *multiValued* las facetas asociadas a los campos *Autores*, *Países* e *Instituciones* ya que un artículo puede contar con más de un atributo de estos tipos a la vez.

```
fconfig.setMultiValued("autores", true);  
fconfig.setMultiValued("paises", true);  
fconfig.setMultiValued("instituciones", true);
```

Luego en el método **indexarDocumentos** añadiremos las facetas extrayendo cada campo del artículo haciendo uso de la clase **JsonReader** de la siguiente forma:

```
for (String autor : j.getListaAutores())  
    doc.add(new FacetField("autores", autor));  
for (String insti : j.getListaInstituciones())  
    doc.add(new FacetField("instituciones", insti));  
for (String pais : j.getListaPaises())  
    doc.add(new FacetField("paises", pais));  
  
doc.add(new FacetField("tamanio", rangos.getIndexIntervalos(j.getTamanio()  
    ))));
```

Como el preprocesado nos devuelve los *Autores*, *Países* e *Instituciones* como listas de *String*, simplemente crearemos una faceta para cada elemento en la lista iterando

sobre las mismas. Para el caso del tamaño, hemos creado una clase **RangosTam** que devuelve un *String* con el intervalo al que pertenece el tamaño de nuestro fichero.

Por último, escribimos en nuestro documento haciendo uso del método *writer.addDocument*, al que le pasamos *fconfig.build(taxoWriter,doc)* donde *taxoWriter* había sido declarada previamente.

5.1.1. Clase RangosTam.

El funcionamiento de la clase **RangosTam** es el siguiente:

- El constructor recibe dos parámetros, el primero *top* especifica el número de subintervalos y el segundo *max* el tamaño máximo.
- Divide en *top* subintervalos el intervalo 0-*max*.
- El método **getIndexIntervalos** recibe como parámetro un número y devolverá como *String* el subintervalo al que pertenece ese número.

5.2. Búsqueda.

En la clase **Busqueda** y al igual que hicimos con el índice, deberemos indicar el directorio donde están almacenadas las facetas que hemos creado en tiempo de indexación. Además crearemos tres variables que contienen los objetos *TaxonomyReader* al cual asignamos el directorio donde se encuentran las facetas y otros dos *FacetsConfig* y *FacetsCollector* que usaremos más adelante.

5.2.1. Mostrar las facetas.

Para mostrar las facetas, hemos creado un método llamado **mostrarFacetas** que recibe como parámetro una consulta y a partir de ella obtiene las facetas de la colección de artículos que se adecuen a dicha consulta. Las facetas contienen información sobre los campos *Autores*, *Países*, *Instituciones* y *Tamaño*.

Guardaremos en *allDims* la lista de todas las facetas obtenidas para la consulta aplicada a los documentos y cada faceta será un objeto *FaceResult* que contiene la información específica de cada faceta. La variable *FastTaxonomyFacetCounts* almacenará el total de ocurrencias para cada faceta para esa consulta determinada.

Finalmente, se mostrará por pantalla una lista con las top 5 facetas para la consulta realizada sobre la colección de documentos indexados. Además se mostrará la categoría a la que pertenecen y el número de ocurrencias de la búsqueda.

```

Introduzca la consulta: mental

Se han encontrado 19 documento/s en total. ¿Desea mostrar las facetas? y/n: y

Categorías totales 4
Mostrando las 5 (máx) más relevantes de cada una...

Categoría: autores
Emma E Sypes (#n)-> 1
Liam Whalen-Browneid (#n)-> 1
Sofia B Ahmed (#n)-> 1
Karen E A Burns (#n)-> 1
Alison Fox-Robichaudid (#n)-> 1

Categoría: instituciones
Princess Margaret Hospital (#n)-> 1
University of Toronto (#n)-> 1
Rishikesh District Dehradun Uttarakhand (#n)-> 1
Michigan Medicine (#n)-> 1
Johns Hopkins School of Medicine (#n)-> 1

Categoría: países
India (#n)-> 3
Canada (#n)-> 3
Italy (#n)-> 2
Belgium (#n)-> 2
Japan (#n)-> 1

Categoría: tamaño
0-800 (#n)-> 18
800-1600 (#n)-> 1

```

Figura 4: (Máx) 5 facetas obtenidas para la consulta genérica para la palabra *mental*

Podemos ver que a partir de los 19 resultados obtenidos, se muestran de forma dinámica el top 5 (como máximo) por categoría de las facetas más relevantes de cada documento.

En el caso de tamaño, como todavía no estamos trabajando con la totalidad de documentos, no obtenemos 5 subintervalos, aunque vemos que la creación de rangos tal y como la hemos programado, funciona correctamente.

5.2.2. Filtrar por facetas.

Una vez recopiladas las facetas, podremos realizar un filtrado de la búsqueda haciendo uso de ellas. Hemos creado un método llamado **FiltrarPorFacetas** que recibe por parámetro la consulta, los documentos resultantes de la búsqueda previa, un vector de *String* que contiene las opciones de las facetas para elegir y un *Map* que contiene la categoría a la que pertenece cada faceta.

Nuestro objetivo es realizar un filtrado por *DrillDown* sobre los documentos resultantes de una consulta para las categorías seleccionadas. En el ejemplo anterior, hemos obtenido 19 documentos como resultado de la consulta genérica buscando por la palabra *mental* en el *Texto*. Observando la categoría *Países*, vemos que 3 de los documentos son de *India*, si filtramos por esa faceta, obtendremos los 3 documentos de los 19 que contienen entre sus países a la *India*.

Este proceso se puede implementar haciendo uso del objeto *DrillDownQuery* al que le pasaremos la consulta inicial y todas las facetas y luego haremos el filtrado sobre las facetas seleccionadas por el usuario.

Este filtro puede implementarse como un AND (por defecto) o incluso como un OR añadiendo todas las facetas como la misma etiqueta al objetivo *DrillDownQuery*. Nosotros usaremos la versión con AND, pues es la más común en los Sistemas de Recuperación de Información de este tipo.

La variable *vector_facetas* contiene un *String* con cada faceta obtenida dinámicamente de la búsqueda anterior. El usuario podrá seleccionar las facetas para las que desee realizar el filtrado y se almacenarán como un *String* que servirá como parámetro al objeto *DrillDownQuery*.

Finalmente para especificar la información necesaria para el *DrillDownQuery* sobre a qué categoría pertenece cada faceta, hacemos una consulta a un *Map* (creado cuando mostramos las facetas) con *ddq.add(map_facetas.get(faceta), faceta)*.

Una vez añadidas al *DrillDownQuery* todas las facetas, llamaremos al método *FacetsCollector.search* al que le pasaremos el nuevo objeto *DrillDownQuery* y nos devolverá un nuevo objeto *TopDocs* con el filtro aplicado sobre los documentos que habíamos obtenido a partir de la primera consulta.

Mostraremos un ejemplo del filtrado de facetas sobre el caso anterior de la consulta genérica buscando por la palabra *mental* sobre el texto completo:

```

Filtramos query( +texto:mental ) a la que aplicaremos DrillDownQuery
Total hits = 19 hits

Filtrar por:

(0) autores (#n)-> Emma E Sypes
(1) autores (#n)-> Liam Whalen-Browneid
(2) autores (#n)-> Sofia B Ahmed
(3) autores (#n)-> Karen E A Burns
(4) autores (#n)-> Alison Fox-Robichaudid
(5) instituciones (#n)-> Princess Margaret Hospital
(6) instituciones (#n)-> University of Toronto
(7) instituciones (#n)-> Rishikesh District Dehradun Uttarakhand
(8) instituciones (#n)-> Michigan Medicine
(9) instituciones (#n)-> Johns Hopkins School of Medicine
(10) paises (#n)-> India
(11) paises (#n)-> Canada
(12) paises (#n)-> Italy
(13) paises (#n)-> Belgium
(14) paises (#n)-> Japan
(15) tamaño (#n)-> 0-800
(16) tamaño (#n)-> 800-1600

```

Figura 5: Filtrar por facetas nos muestra una lista con todas las opciones.

```

Introduzca los filtros: 0 6 15

Nueva búsqueda ( +texto:mental #($facets:autoresEmma E Sypes) #($facets:institucionesUniversity of Toronto) #($facets:tamaño0-800) )

Coincidencias totales = 1 hits
-----
Nombre fichero: 00a1a921b93d9773f46d21ac22b1363371c7d535.json
Tamaño: 122
Título: A national cross-sectional survey of public perceptions of the COVID-19 pandemic: Self-reported beliefs, knowledge, and behaviors
Autores: Jeanna Parsons Leigh, Kirsten Fiest, Rebecca Brundin-Mather Id, Kara Plotnikoff, Andrea Soo, Emma E Sypes, Liam Whalen-Browneid, Sofia B Ahmed, Karen E A Burns, Alison Fox-Robichaudid, Shelly Kupsch, Shelly Longmore, Srinivas Murthy, Daniel J Niven, Bram Rochweg, Henry T Stelfox
Países: Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada
Instituciones: Dalhousie University, University of Calgary, University of Calgary, University of Calgary, University of Calgary, University of Calgary, University of Calgary, University of Calgary, University of Toronto, McMaster University, University of Calgary, University of Calgary, Vancouver British Columbia, University of Calgary, McMaster University, University of Calgary
Abstract: Efforts to mitigate the global spread of the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) causing Corona Virus Disease-19 (COVID-19) have largely relied o [...]
-----

```

Figura 6: Aplicamos los filtros deseados a la consulta.

6. Pruebas realizadas.

Se han creado dos interfaces para realizar las búsquedas en nuestro Sistema de Recuperación de información cuyo uso es bastante intuitivo. Ambas necesitan de la creación previa del índice y facetas, acción que se puede realizar ejecutando el archivo *script.sh* con la opción *-i* en el directorio *./terminal/*

Los documentos en formato JSON que vayan a ser indexados deberemos guardarlos en el directorio *./terminal/pdf_json/*.

6.1. Interfaz por terminal.

```
¿Cómo desea realizar la consulta genérica?
1: Sobre el texto.
2: Especificar campo/s.
3: Salir.

Opción: 2
Campos disponibles: ( titulo / autores / paises / instituciones / tamaño )

Introduzca el campo: paises

Introduzca la consulta: USA

Se han encontrado 17 documento/s en total. ¿Desea mostrar las facetas? y/n: y

Categorías totales 4
Mostrando las 5 (máx) más relevantes de cada una...

Categoría: autores
Jennifer S Herrick (#n)-> 1
Heather T Keenan (#n)-> 1
David G Kirsch (#n)-> 1
Maximilian Diehn (#n)-> 1
Francis A Cucinoata (#n)-> 1

Categoría: instituciones
University of California (#n)-> 2
Louisiana State University (#n)-> 2
Hong Kong Special Administrative Region (#n)-> 1
University of Utah Health (#n)-> 1
Duke University Medical Center (#n)-> 1

Categoría: paises
USA (#n)-> 17
China (#n)-> 2
India (#n)-> 1
United States (#n)-> 1
UK (#n)-> 1

Categoría: tamaño
0-800 (#n)-> 17
```

Figura 7: Búsqueda genérica sobre el campo países.

Realizamos una búsqueda genérica sobre el campo países y seleccionaremos los documentos de *USA*, obtenemos (del subconjunto de 196 archivos cargado actualmente) 17 documentos que cumplen esta restricción.

```
¿Quieres filtrar por facetas? y/n: y

Filtramos query( +Synonym(paises:eeuu paises:usa) ) a la que aplicaremos DrillDownQuery
Total hits = 17 hits

Filtrar por:

(0) autores (#n)-> Jennifer S Herrick
(1) autores (#n)-> Heather T Keenan
(2) autores (#n)-> David G Kirsch
(3) autores (#n)-> Maximilian Diehn
(4) autores (#n)-> Francis A Cucinoata
(5) instituciones (#n)-> University of California
(6) instituciones (#n)-> Louisiana State University
(7) instituciones (#n)-> Hong Kong Special Administrative Region
(8) instituciones (#n)-> University of Utah Health
(9) instituciones (#n)-> Duke University Medical Center
(10) paises (#n)-> USA
(11) paises (#n)-> China
(12) paises (#n)-> India
(13) paises (#n)-> United States
(14) paises (#n)-> UK
(15) tamaño (#n)-> 0-800

Introduzca los filtros: 7 11 15
```

Figura 8: Facetas por las cuales podemos filtrar la búsqueda anterior.

Seleccionamos las facetas para la institución *Hong Kong Special Administrative Region*, país *China* y tamaño entre *0-800*. De los 17 documentos que tienen como país a *USA* filtramos los que cumplan las tres propiedades elegidas.

Obtenemos 1 sólo resultado que es el único que cumple todas las condiciones especificadas como consulta AND.

```

Nueva búsqueda ( +Synonym(paises:eeuu paises:usa) #($facets:institucionesHong Kong Special Administrative Region) #($facets:paisesChina) #($facets:tamaño0-800) )

Coincidencias totales = 1 hits
-----
Nombre fichero: 00a4b80a7e0f35c3f8ba74f5a9eaea73b77bdc3b.json
Tamaño: 35
Titulo: China: Estimation of Super-spreading Events, Serial Interval, and Hazard of Infection
Autores: Xiao-Ke Xu, Xiao-Fan Liu, Ye Wu, Taslim Ali, Zhanwei Du, Paolo Bosetti, Eric H Y Lau, Benjamin J Cowling, Lin Wang
Países: China, China, China, China, USA, France, China, China, UK
Instituciones: Dalian Minzu University, Hong Kong Special Administrative Region, Beijing Normal University, Hong Kong Special Administrative Region, University of Texas at Austin, CNRS, Hong Kong Special Administrative Region, Hong Kong Special Administrative Region, University of Cambridge
Abstract: A c c e p t e d M a n u s c r i p t 2 Summary): A unique COVID-19 line-list database comprising 1,407 transmission pairs that formed 643 clusters in mainland China outsid [...]
-----

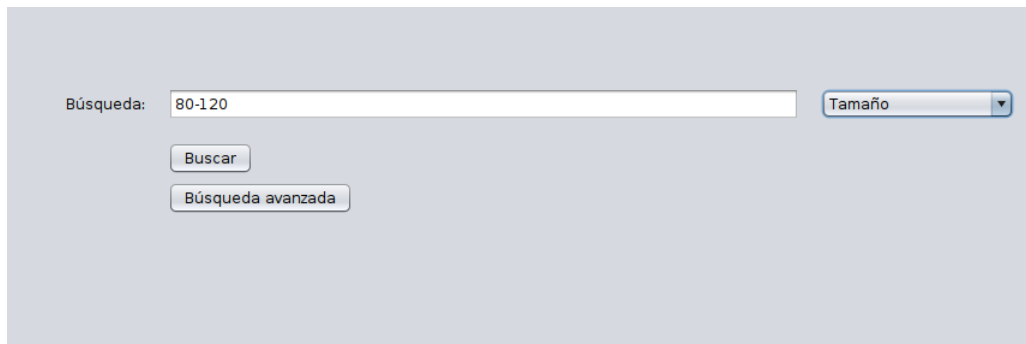
```

Figura 9: Resultados después de filtrar.

6.2. Interfaz gráfica.

6.2.1. Búsqueda simple.

Ahora realizaremos una consulta desde la interfaz gráfica para demostrar que su funcionamiento es muy similar al de la interfaz por terminal.



The image shows a web-based search interface. At the top, there is a search bar containing the text "80-120". To the right of the search bar is a dropdown menu labeled "Tamaño". Below the search bar, there are two buttons: "Buscar" and "Búsqueda avanzada".

Figura 10: Realizamos una búsqueda por rango de tamaño.

Buscamos los documentos cuyo tamaño de fichero se encuentra entre 80 y 120 KB. Obteniendo 35 resultados, que como se puede observar, cumplen la restricción inicial especificada.

A continuación, podemos realizar una búsqueda por facetas sobre los resultados que hemos encontrado de igual manera que lo hacíamos en terminal, esta vez con un desplegable donde podremos seleccionar la opción para cada categoría y mostrará el resultado que interseque con la búsqueda especificada.

Ahora, elegiremos algunas facetas para modificar los resultados de la búsqueda inicial. Podemos ir seleccionando unas facetas u otras y los resultados irán cambiando de forma dinámica dependiendo de nuestra especificación.

The screenshot shows a search interface with the following components:

- Filters (Left):**
 - autores:** Dropdown menu with "Todos" selected.
 - instituciones:** Dropdown menu with "Todos" selected.
 - tamaño:** Dropdown menu with "Todos" selected.
 - países:** Dropdown menu with "Todos" selected.
- Results (Right):**
 - Header: "Resultados: 35 hits"
 - Item 1:
 - Instituciones: Unioersity of Iowa College of Medicine
 - Abstract: Unknown
 -
 - Item 2:
 - Nombre fichero: 0a2cdaf20965864cdc7b66ef39f9418c6328ce94.json
 - Tamaño: 86
 - Título: Unknown
 - Autores: Unknown
 - Países: Unknown
 - Instituciones: Unknown
 - Abstract: Unknown
 -
 - Item 3:
 - Nombre fichero: 0a5c11c9f6c76efe534c05013e4029acc8433af1.json
 - Tamaño: 97
 - Título: SARS-CoV-2 infection, disease and transmission in domestic cats
 - Autores: Natasha N Gaudreault, Jessie D Trujillo, Mariano Carossino, David A Meekins, Igor Morozov, Daniel W Madden, Sab
 - Países: USA, USA, USA, USA, USA, USA, USA, USA, USA, USA, USA, USA, United States, USA, USA, USA, USA
 - Instituciones: Kansas State University, Kansas State University, Louisiana State University, Kansas State University, Kansas
 - Abstract: Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) is the cause of Coronavirus Disease 2019 and r
 -
 - Item 4:
 - Nombre fichero: 0a5e7650d8ff6d24c2ee42d666a3c46aeef596ae.json
 - Tamaño: 107
 - Título: Distribution equality as an optimal epidemic mitigation strategy
 - Autores: Adar Hacohen, Reuven Cohen, Sol Efroni, Ido Bachelet, Baruch Barzel
 - Países: Israel, Israel
 - Instituciones: Bar-Ilan University, Bar-Ilan University
 - Abstract: Upon the development of a drug or vaccine, a successful response to a global pandemic, such as COVID-19, req
 -
- Buttons (Bottom):**
 - "Nueva búsqueda"
 - "Finalizar"

Figura 11: Resultados obtenidos de la consulta por rango de tamaño.

Finalmente hemos pasado de 35 resultados a 1 restringiendo la búsqueda por los campos *Princeton University* para la institución y *USA* para el país. Es evidente que nuestro resultado también cumple la búsqueda inicial por rango de tamaño que habíamos especificado en primera instancia.

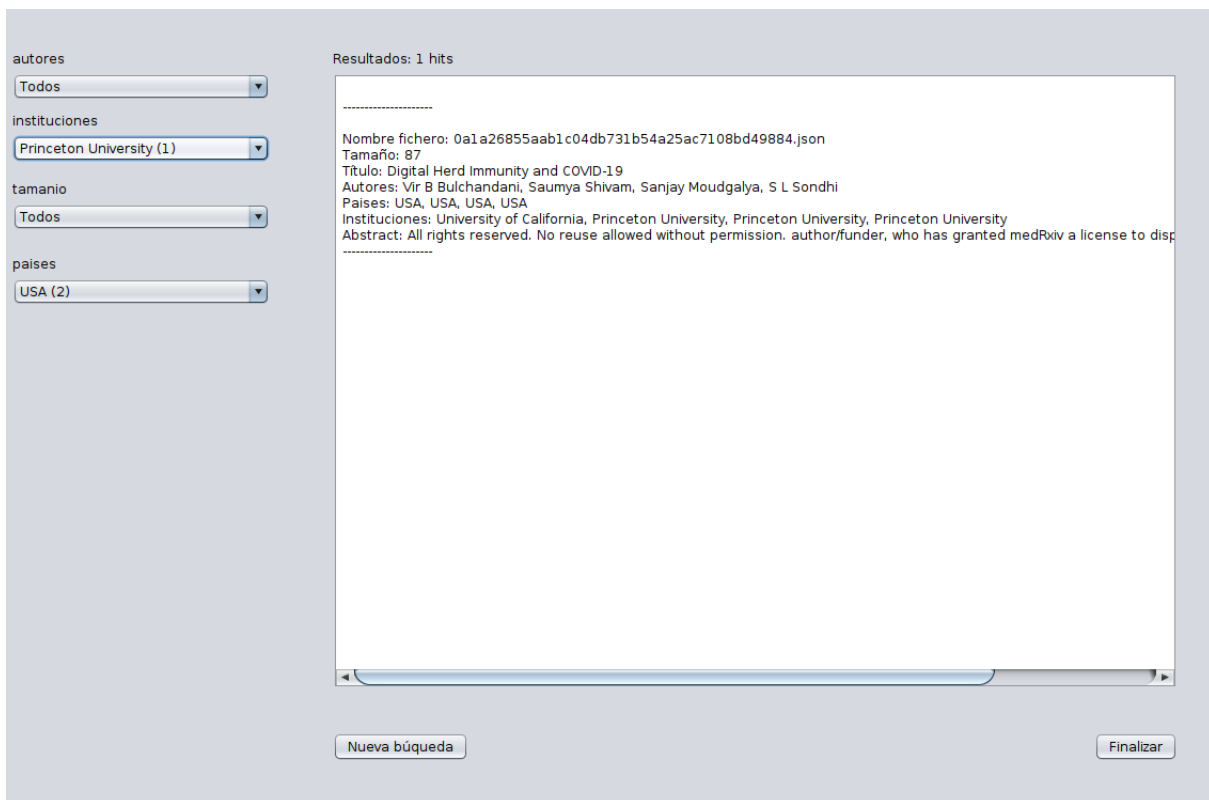


Figura 12: Resultados después de filtrar.

6.2.2. Búsqueda avanzada.

Realizaremos desde interfaz gráfica una búsqueda avanzada que equivale a una consulta booleana sobre varios campos. Además podemos especificar si queremos hacer una consulta AND o OR en el desplegable que aparece en la parte superior derecha.

Vemos como para la misma búsqueda, cambiando el operador lógico, obtenemos los resultados esperados en cada caso.

| | | |
|-------------------|--------------------------------------|----------------|
| Título | <input type="text" value="covid"/> | <div>AND</div> |
| Autores | <input type="text"/> | |
| Países | <input type="text" value="Canada"/> | |
| Instituciones | <input type="text"/> | |
| Abstract | <input type="text"/> | |
| Texto | <input type="text" value="vaccine"/> | |
| <div>Buscar</div> | | |

Figura 13: Búsqueda avanzada booleana con AND.

[illegible]

Figura 14: Resultados de la búsqueda.

Titulo: covid
 Autores:
 Países: Canada
 Instituciones:
 Abstract:
 Texto: vaccine

OR

Buscar

Figura 15: Búsqueda avanzada booleana con OR.

Resultados: 93 hits

autores: Todos
 instituciones: Todos
 tamaño: Todos
 países: Todos

Nombre fichero: 0a0af20f552c58dd3d3b0f9d937d7648ba59f7fc.json
 Tamaño: 37
 Título: Participation in TREC 2020 COVID Track Using Continuous Active Learning
 Autores: Jean Xue, Jun Wang, Maura R Grossman, Kevin, Seung Gyu, Hyun
 Países: Canada, Canada, Canada, Canada, Canada, Canada, Canada
 Instituciones: University of Waterloo Waterloo, University of Waterloo Waterloo, University of Waterloo Waterloo, University
 Abstract: We describe our participation in all five rounds of the TREC 2020 COVID Track (TREC-COVID). The goal of TREC-CO

Nombre fichero: 00a1a921b93d9773f46d21ac22b1363371c7d535.json
 Tamaño: 122
 Título: A national cross-sectional survey of public perceptions of the COVID-19 pandemic: Self-reported beliefs, knowledge
 Autores: Jeanna Parsons Leigh, Kirsten Fiest, Rebecca Brundin-Mather Id, Kara Plotnikoff, Andrea Soo, Emma E Sypes, Lian
 Países: Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada, Canada
 Instituciones: Dalhousie University, University of Calgary, University of Calgary, University of Calgary, University of Calgary, U
 Abstract: Efforts to mitigate the global spread of the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) caus

Nombre fichero: 0a6e9aa35d1320355bf071879e98aabcacfe2b85.json
 Tamaño: 66
 Título: Twelve Essentials of Science-based Policy • Centers for Disease Control and Prevention 7
 Autores: Bernard C K Choi
 Países: Canada
 Instituciones: University of Ottawa
 Abstract: This article presents a systematic framework of 12 essentials, or basic elements, of science-based policy. The 1

Nombre fichero: 0a01f5cf1c5cdc2711bce74315dc54a6e143df0.json

Figura 16: Resultados de la búsqueda.

7. ¿Cómo ejecutar el buscador?

Encontraremos el *script* dentro del directorio *terminal*. Para ejecutar el buscador en interfaz por terminal, haremos uso de la orden siguiente, donde *-option* se refiere a *[-i, -b, - -- help]*.

```
> ./script.sh -option
```

- **-i**: Crea el índice. Esta opción es necesaria tanto para ejecutar el buscador por terminal como el de interfaz gráfica.
- **-b**: Inicia el buscador en interfaz por terminal.
- **--help**: Da una descripción sobre las opciones e indica el directorio donde se añaden los archivos previa indexación (./pdf_json/).

Para ejecutar el buscador en interfaz gráfica, abriremos el proyecto en Netbeans y correremos el programa.

Para ejecutar la práctica se necesita añadir los siguientes directorios y paquetes:

Para la interfaz por terminal, añadir en el directorio ./terminal/library/:

- Directorio **lucene-8.6.2** [4]
- **json-simple-1.1.1.jar** [5]
- **hppc-0.8.2.jar** [6]

Para la interfaz gráfica, añadir en el directorio ./src/main/java/Librerias/:

- **lucene-queryparser-8.6.2.jar**
- **lucene-facet-8.6.2.jar**
- **lucene-core-8.6.2.jar**
- **lucene-analyzers-common-8.6.2.jar**

*Los 4 archivos *.jar* han sido extraídos del directorio *lucene-8.6.2*

- **hppc-0.8.2.jar** [6]

8. Trabajo en Grupo.

El trabajo lo hemos repartido, en primera instancia, de la siguiente manera:

- **Daniel Bolaños Martínez:** Lector de archivos JSON, búsqueda por facetas, búsqueda por rangos de tamaño y memoria.
- **Fernando de la Hoz Moreno:** Creación del índice, búsquedas booleanas y búsqueda por términos e interfaz gráfica.

No obstante, hemos mantenido el contacto durante el desarrollo de la práctica y hemos colaborado conjuntamente en la elaboración del proyecto, haciendo un seguimiento continuo del trabajo realizado.

Referencias

- [1] Guiones para las prácticas 3-6 de la asignatura.
- [2] <https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge>
- [3] https://lucene.apache.org/core/8_6_2/core/index.html
- [4] <https://lucene.apache.org/core/downloads.html>
- [5] <https://code.google.com/archive/p/json-simple/downloads>
- [6] <https://mvnrepository.com/artifact/com.carrotsearch/hppc/0.8.2>