

# Memoria Final RI

---

## Introducción

En este caso hemos decidido hacer un sistema de recuperación web utilizando la lista de scopus.

La arquitectura de nuestro sistema estará compuesto de estos 3 módulos:

- Indexador encargado de mantener actualizado el índice de lucene a partir de los documentos que vayan entrando en la carpeta /Datasets. El indexador también se encarga de construir el índice para las facetas.
- Backend REST encargado de leer el índice de lucene y responder a las consultas de búsquedas
- Frontend web que ofrecerá una interfaz sencilla donde poder realizar las consultas, las cuales se enviarán al backend donde se devolverá el resultado y se mostrará en pantalla

## Indexador

Para el indexador hemos realizado un programa que está constantemente en funcionamiento buscando nuevas listas de documentos en formato csv.

Cuando un documento entra en la carpeta, el indexador lo detecta y hace un append al índice para actualizarlo.

El índice está compuesto por dos carpetas:

- index: Donde está el índice
  - Title
  - Abstract
  - Author
  - EID: Aunque no se use para indexar, lo almacenaremos para poder ir a la página del documento de Scopus
- facets: Donde se indexan las facetas
  - Año
  - Keywords Todo esto se consigue hacer dentro del método IndexarDocumentos de la clase IndiceSimple.

## Backend

Para el backend hemos usado la librería javalin, que nos permitirá tener un servidor REST con java de una forma sencilla de implementar.

El servidor solamente escuchará una sola petición :

`http://localhost:7030/?title=&author=&year=&keyword=`

De esta forma, con una única petición podemos enviar todos los parámetros de búsqueda que queremos realizar.

## Frontend

Para el frontend hemos utilizado la librería de frontend Vue.js con la herramienta de diseño Vuetify para poder hacer de una forma rápida y visualmente atractiva una interfaz encargada de poder introducir comodamente los parámetros de búsqueda y poder visualizar los resultados, y así poder mostrar que nuestro motor de búsqueda puede aplicarse a un entorno web real.

El frontend está constituido por diferentes componentes:

- Barra de búsqueda principal: Busca directamente en el abstract de los documentos
- Navegación Lateral
  - Filtro por título: Permite refinar la búsqueda introduciendo palabras que deben estar en el título
  - Filtro por autor: Permite buscar documentos de un autor en concreto, como es complicado que el usuario sepa introducir exactamente el nombre del autor, la idea es que el usuario pueda hacer click en el nombre del autor de un documento en concreto y automáticamente filtrar por ese usuario
  - Filtro por año: Permite buscar los documentos filtrados por un año en concreto
  - Filtro por keywords: Muestra las 5 keywords más comunes de los documentos filtrados y permite filtrar los resultados por uno de esos keywords.

Cada vez que se aplican los diferentes filtros que tenemos, la lista de keywords se actualiza para mostrarse siempre las 5 más populares de los documentos **filtrados**.

Si queremos filtrar por otro tipo de keyword, podemos seleccionar en la lista de resultados una de las keywords que tenga un documento y filtrar por esa keyword.