
Práctica final

Recuperación de Información

Amin Kasrou Aouam



**UNIVERSIDAD
DE GRANADA**

2021-01-11

Índice

Práctica final	3
<i>Parsing</i>	3
Indexación	4
Buscador	4
Ejecución	4

Práctica final

En esta práctica, vamos a implementar un buscar de información en una serie de documentos *JSON* usando *Apache Lucene*.

Parsing

Parseamos los documentos *JSON* mediante la librería *GSON*, dado que ésta nos ofrece la posibilidad de serializar el documento en una clase Java.

Esta funcionalidad nos facilita la extracción de información en *JSON* complejos. Para ellos, creamos la clase **Paper**:

```
1 package org.RI.P2;
2
3 import java.util.List;
4
5 import com.google.gson.annotations.SerializedName;
6
7 class Affiliation {
8     String laboratory;
9     String institution;
10 }
11
12 class Author {
13     String first;
14     List<String> middle;
15     String last;
16     String suffix;
17     Affiliation affiliation;
18     String email;
19 }
20
21 class Metadata {
22     String title;
23     List<Author> authors;
24 }
25
26 class Abstract {
27     String text;
28 }
29
30 public class Paper {
31     String paper_id;
32     Metadata metadata;
33     @SerializedName("abstract") List<Abstract> abstr;
34 }
```

Es esencial utilizar el mismo nombre de atributo en nuestra clase, dado que esto permitirá un *mapping* correcto. En caso de que no fuera posible, podemos hacer uso del decorador **@SerializedName**.

El único paso que nos falta es indicarle a la librería la entrada (*JSON*) y la clase.

```
1 Gson gson = new Gson();
2 Paper data = gson.fromJson(readerJson, Paper.class);
```

Indexación

Para la indexación, hemos elegido los siguientes atributos:

- `paper_id`
- `title`
- `authors`
- `institution`
- `emails`
- `abstract`

Optamos por crear un índice en cada ejecución, para evitar obtener valores repetidos.

Buscador

Implementamos un buscador con interfaz gráfica (GUI), basada en el proyecto de ejemplo disponible en la plataforma Prado.

Podemos optar por buscar en diferentes campos, mediante un argumento pasado por línea de comandos. Especificamos las distintas opciones en la sección siguiente.

Ejecución

En el caso que deseemos utilizar *Maven*, debemos ejecutar los siguientes comandos:

1. Compilar el proyecto

```
1 mvn compile
```

1. Ejecutar el proyecto

```
1 mvn exec:java -Dexec.mainClass="org.RI.P2.Searcher" -Dexec.args="data title"
```

Debemos modificar el argumento **title** según la salida que deseemos:

- **title**
- **authors**
- **abstract**
- **institutions**
- **emails**