

Desarrollo de un SRI Con Lucene

Francisco José González
fran98@correo.ugr.es



@Neo-Stark

Tecnologías usadas

Scala



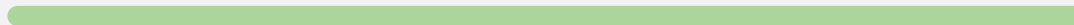
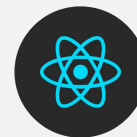
Lucene



Electron + React



+



Indexador

```
val dir: Directory = FSDirectory.open(Paths.get(indexPath))  
val analyzer: Analyzer = new EnglishAnalyzer()  
val iwc: IndexWriterConfig = new IndexWriterConfig(analyzer)  
iwc.setOpenMode(OpenMode.CREATE)  
val writer: IndexWriter = new IndexWriter(dir, iwc)
```

Indexador - EnglishAnalyzer

- Eliminación de los posesivos en inglés: 's
- Normalizado de tokens a minúsculas
- StopFilter: eliminado de palabras vacías del inglés
- Marcado de palabras clave para que posteriormente no sean procesadas en el proceso de stemming
- PorterStemFilter: proceso de stemming utilizando el algoritmo de Porter

Indexador - indexDoc

- Función para añadir los documentos al índice
- Primer campo: **path** -> almacena la ruta hasta el documento
- Segundo campo: **contents**-> contenido del documento que es indexado
- Se preprocesa el texto con la expresión regular "<[>]*>" para quitar las etiquetas html

Buscador

```
val reader = DirectoryReader.open(FSDirectory.open(Paths.get(indexDir)))
val searcher = new IndexSearcher(reader)
val analyzer = new EnglishAnalyzer()
val parser = new QueryParser(field, analyzer)
val query = parser.parse(queryString)
println(s"Buscando = ${query.toString(field)}")
val search = searcher.search(query, 100)
val results = search.scoreDocs
println(s"results.size = ${results.size}")
results.foreach(hit => {
    val hitDoc = searcher.doc(hit.doc)
    println(s"Doc: ${hitDoc.get("path")}")
})
println(s"Total hits: ${search.totalHits.value}")
reader.close()
```