



Universidad de Valladolid

E.T.S Ingeniería Informática
TRABAJO FIN DE GRADO

Grado en Ingeniería Informática - Mención Tecnologías de la
Información

**Estudio comparativo de motores de
búsqueda**

Autor:
David Robles Gallardo



Universidad de Valladolid

E.T.S Ingeniería Informática
TRABAJO FIN DE GRADO

Grado en Ingeniería Informática - Mención Tecnologías de la
Información

**Estudio comparativo de motores de
búsqueda**

Autor:
David Robles Gallardo

Tutores:
Pablo de la Fuente Redondo
Jose Carlos Jiménez Sánchez

Resumen

En la actualidad, el empleo de motores de búsqueda se ha extendido de forma generalizada. Esto se ha debido principalmente a la proliferación de sitios web, a la evolución de los sistemas y las necesidades de compartir documentos y realizar búsquedas en un mundo globalizado y digitalmente conectado.

Los motores de búsqueda solucionan los problemas que se presentan a la hora de realizar una búsqueda en una cantidad muy elevada de datos, permitiendo devolver resultados interesantes para el usuario mediante algoritmos de ordenación.

Cabe destacar que los motores de búsqueda se encuentran muy de moda en la actualidad, no solamente por su gran éxito en la red, indexando documentos, sino también por su utilización en el mundo de las redes sociales y las modernas técnicas de Big Data.

Sin embargo, esta expansión ha llevado a la creación de numerosos productos que cumplen las mismas funciones, aunque no de la misma forma. Por ello, resulta interesante observar que con este aumento de alternativas, también surge un gran número de comparativas entre ellas, a fin de encontrar la mejor bajo ciertas circunstancias.

A pesar de todo, las comparativas existentes se centran demasiado en los aspectos técnicos y las características numéricas de los motores de búsqueda, y no en otro tipo de propiedades, que sí suelen evaluarse en otros sistemas. Por ello, es necesario efectuar una comparativa desde un enfoque diferente, más próximo a las necesidades de las empresas que busquen analizar más características que las del mero rendimiento.

Este trabajo se ha realizado en colaboración con Everis Spain S.L. en el marco de prácticas ofertadas por la compañía en su acuerdo con la Universidad de Valladolid.

Agradecimientos

En primer lugar, agradecer a mis tutores, Pablo de la Fuente y Jose Carlos Jiménez su entusiasmo por el proyecto, su paciencia y su inestimable guía y conocimientos. También por ayudarme a adquirir nuevos conocimientos, y, sobre todo, por animarme a analizar los estudios del rendimiento desde un punto de vista diferente.

También quiero dar las gracias a mi familia, por haberme apoyado durante todos estos años, por animarme a seguir en los momentos difíciles y por enseñarme el valor del trabajo. En especial a mis padres, por no perder nunca la confianza en mis posibilidades.

Me gustaría asimismo reconocer a mis amigos y compañeros de grado el enorme esfuerzo que han realizado para mejorar mi experiencia universitaria, y el gran aporte que han supuesto para mi formación, ayudándome con su experiencia y consejos a llegar hasta aquí.

Por último, agradecer a todos los profesores de la Escuela de Ingeniería Informática de Valladolid su gran labor, orientándonos durante todos estos años para alcanzar un futuro mejor en esta gran profesión.

Índice general

Índice general	V
Índice de figuras	XI
Índice de tablas	XIII
1. Introducción	1
1.1. Problemática	1
1.2. Necesidad de la creación de una comparativa	2
1.3. Aportaciones del presente documento	3
1.4. Estructura del documento	3
1.5. Planificación del trabajo	4
1.5.1. Planificación inicial	5
1.5.2. Planificación final real	5
2. Estado del arte de los motores de búsqueda	7
2.1. Introducción	7
2.2. Funcionamiento de un motor de búsqueda	7
2.3. Descripción de los motores de búsqueda estudiados	9
2.3.1. Algolia	9
2.3.2. Amazon CloudSearch	10

2.3.3.	Apache Solr	10
2.3.4.	Crate	11
2.3.5.	ElasticSearch	11
2.3.6.	Google Search Appliance	12
2.3.7.	Indica ES	12
2.3.8.	Microsoft Azure Search	13
2.3.9.	SearchBlox	13
2.3.10.	Sphinx Search Server	14
3.	Descripción del benchmark construido	15
3.1.	Introducción	15
3.2.	Métricas a nivel ejecución	16
3.2.1.	Productividad	16
3.2.2.	Requisitos de ejecución	16
3.2.3.	Consumo de energía	17
3.2.4.	Multi-plataforma	17
3.2.5.	Seguridad	18
3.3.	Métricas a nivel operación	18
3.3.1.	Monitorización	18
3.3.2.	Mantenimiento	19
3.3.3.	Escalabilidad	19
3.3.4.	Arquitectura	20
3.4.	Métricas a nivel desarrollo	20
3.4.1.	Curva de aprendizaje	20
3.4.2.	Independencia del servidor	21
3.4.3.	Integración con otros sistemas	21

3.5. Métricas a nivel soporte	21
3.5.1. Soporte del proveedor	22
3.5.2. Soporte online y comunidad de usuarios	22
3.5.3. Documentación	22
3.5.4. Licencia del producto	23
3.5.5. Continuidad del producto	23
3.5.6. Multi-idioma	24
4. Estudio comparativo de los motores de búsqueda	25
4.1. Introducción	25
4.2. Comparativa a nivel de ejecución	26
4.2.1. Algolia	26
4.2.2. Amazon Cloud Search	27
4.2.3. Apache Solr	27
4.2.4. Elasticsearch	28
4.2.5. Google Search Appliance	29
4.2.6. Indica ES	29
4.2.7. Microsoft Azure Search	30
4.2.8. SearchBlox	31
4.2.9. Sphinx Search Server	31
4.2.10. Crate	32
4.2.11. Comentario sobre la comparativa a nivel de ejecución	32
4.3. Comparativa a nivel de operación	33
4.3.1. Algolia	33
4.3.2. Amazon Cloud Search	33
4.3.3. Apache Solr	34

4.3.4.	ElasticSearch	34
4.3.5.	Google Search Appliance	35
4.3.6.	Indica ES	35
4.3.7.	Microsoft Azure Search	36
4.3.8.	SearchBlox	36
4.3.9.	Sphinx Search Server	37
4.3.10.	Crate	38
4.3.11.	Comentario sobre la comparativa a nivel de operación	38
4.4.	Comparativa a nivel de desarrollo	38
4.4.1.	Algolia	38
4.4.2.	Amazon Cloud Search	39
4.4.3.	Apache Solr	39
4.4.4.	ElasticSearch	40
4.4.5.	Google Search Appliance	40
4.4.6.	Indica ES	41
4.4.7.	Microsoft Azure Search	41
4.4.8.	SearchBlox	42
4.4.9.	Sphinx Search Server	42
4.4.10.	Crate	43
4.4.11.	Comentario sobre la comparativa a nivel de desarrollo	43
4.5.	Comparativa a nivel de soporte	44
4.5.1.	Algolia	44
4.5.2.	Amazon Cloud Search	44
4.5.3.	Apache Solr	45
4.5.4.	ElasticSearch	46
4.5.5.	Google Search Appliance	46

<i>ÍNDICE GENERAL</i>	IX
4.5.6. Indica ES	47
4.5.7. Microsoft Azure Search	48
4.5.8. SearchBlox	48
4.5.9. Sphinx Search Server	49
4.5.10. Crate	50
4.5.11. Comentario sobre la comparativa a nivel de soporte	50
4.6. Comparativa final global	51
4.6.1. Comentario sobre la comparativa final global	52
5. Estudio práctico. Diseño de un repositorio web	55
5.1. Introducción	55
5.2. Descripción de las tecnologías a emplear	56
5.2.1. Java	56
5.2.2. Spring IO	56
5.2.3. Apache Tomcat	57
5.2.4. Maven	57
5.2.5. Bootstrap	57
5.3. Descripción del entorno de trabajo	57
5.4. Análisis y diseño de la aplicación web	58
5.4.1. Requisitos	58
5.4.2. Casos de uso	59
5.4.3. Diagramas de secuencia	61
5.4.4. Arquitectura escogida. Funcionamiento de Spring MVC	62
5.4.5. Diagrama de clases de análisis y diseño	64
5.5. Desarrollo de la aplicación	65
5.5.1. Introducción	65

5.5.2. Creación del proyecto base	66
5.5.3. Implementación del sistema de inicio de sesión	66
5.5.4. Configuración del motor de búsqueda	67
5.5.5. Integración del motor de búsqueda en la aplicación	67
5.5.6. Implementación del sistema de subida de documentos y registro en el índice	68
5.5.7. Implementación de búsqueda y descarga de documentos	69
5.6. Pruebas sobre el desarrollo	69
6. Conclusiones y trabajo futuro	71
6.1. Conclusiones	71
6.2. Trabajo futuro	71
Bibliografía	73
Palabras clave	75
Contenido del CD adjunto	76

Índice de figuras

4.1. Gráfica nota final sobre 10	52
5.1. Diagrama de casos de uso	59
5.2. Realizar búsqueda.	61
5.3. Diagrama de secuencia. Inicio de sesión.	62
5.4. Diagrama de secuencia. Subir archivo.	62
5.5. Diagrama funcionamiento SpringMVC.	64
5.6. Diagrama clases de diseño.	65

Índice de tablas

4.1. Ejecución - Algolia	26
4.2. Ejecución - ACS	27
4.3. Ejecución - Solr	27
4.4. Ejecución - ElasticSearch	28
4.5. Ejecución - GSA	29
4.6. Ejecución - Indica	29
4.7. Ejecución - MicrosoftAS	30
4.8. Ejecución - SearchBlox	31
4.9. Ejecución - Sphinx	31
4.10. Ejecución - Crate	32
4.11. Operación - Algolia	33
4.12. Operación - ACS	33
4.13. Operación - Solr	34
4.14. Operación - ElasticSearch	34
4.15. Operación - GSA	35
4.16. Operación - Indica	35
4.17. Operación - Microsoft AS	36
4.18. Operación - SearchBlox	36
4.19. Operación - Sphinx	37

4.20. Operación - Crate	38
4.21. Desarrollo - Algolia	38
4.22. Desarrollo - ACS	39
4.23. Desarrollo - Apache Solr	39
4.24. Desarrollo - ElasticSearch	40
4.25. Desarrollo - GSA	40
4.26. Desarrollo - Indica ES	41
4.27. Desarrollo - Microsoft AS	41
4.28. Desarrollo - SearchBlox	42
4.29. Desarrollo - Sphinx	42
4.30. Desarrollo - Crate	43
4.31. Soporte - Algolia	44
4.32. Soporte - ACS	44
4.33. Soporte - Solr	45
4.34. Soporte - ElasticSearch	46
4.35. Soporte - GSA	46
4.36. Soporte - Indica ES	47
4.37. Soporte - Microsoft AS	48
4.38. Soporte - SearchBlox	48
4.39. Soporte - Sphinx Search Server	49
4.40. Soporte - Crate	50
4.41. Tabla resumen áreas de evaluación	51
4.42. Comparativa final global	52
5.1. CU-01. Realizar búsqueda	60
5.2. CU-02. Descargar fichero	60
5.3. CU-03. Gestionar sesión	60

5.4. CU-04. Subir fichero al sistema 61

Capítulo 1

Introducción

Cuando se habla de motor de búsqueda actualmente, siempre se tiende a relacionarlo con la consulta web. En general, el impacto que ha supuesto la creación de Internet, junto con la necesidad del ser humano de compartir información, ha llevado a la inevitable búsqueda de herramientas que facilitasen el proceso de clasificación de documentos conforme a las necesidades del usuario. Además, la creciente cantidad de datos que coexisten en esta red de ordenadores precisa ser estructurada de alguna forma, clasificada de algún modo que permita su interpretación. En esencia, esta es la misión del motor de búsqueda: permitir a las personas acceder a través de un conjunto de palabras clave a la información que desean, pero intentando presentar en primer lugar los datos relevantes para el usuario.

Sin embargo, los motores de búsqueda no se emplean únicamente en Internet. Aunque ese fue el origen de su nacimiento, en la actualidad se han convertido en poderosas herramientas para aportar soluciones de negocio, productos para búsquedas internas en documentación de empresa, o situaciones que requieran localizar información dentro de un gran volumen de datos, como ocurre con el concepto de moda, el Big Data.

De esta manera, los motores de búsqueda se han convertido en algo intrínseco en nuestro día a día. Es por ello que realizar un estudio analizando las semejanzas y diferencias entre elementos existentes puede resultar muy interesante.

1.1. Problemática

Así pues, se ha examinado superficialmente la función y el objetivo de los motores de búsqueda. Sin embargo, es posible que una persona desee emplear los servicios de uno de ellos para resolver una tarea. Esta necesidad es muy común en el ámbito empresarial, y, en concreto, en aquellas compañías que manejan grandes volúmenes de información o documentación.

Para seleccionar el producto que mejor se adapta a sus necesidades, las empresas no se fijan en exclusividad en lo bueno que es el producto en cuanto a funcionalidad, sino que evalúan otros aspectos no numéricos necesarios para el buen funcionamiento e integración del servicio en el ámbito.

Estos otros elementos son muy relevantes a la hora de seleccionar un sistema para el uso empresarial, por razones muy concretas. El soporte, la documentación disponible o los extras que aporte ese motor de búsqueda no son elementos que puedan valorarse a través de fórmulas matemáticas, y, sin embargo, son criterios clave a la hora de escoger entre uno y otro.

He aquí la problemática existente. Aquellos estudios comparativos que pueden encontrarse examinan las diferencias entre los productos desde un punto de vista matemático, a través de fórmulas de rendimiento que pueden resultar útiles, pero que no terminan de dar una visión global del comportamiento del motor.

Existen además otras comparativas que intentan enfocar el estudio en elementos ajenos a lo numérico, aunque con un éxito parcial. Un ejemplo es el informe realizado por la consultora Gartner sobre los motores de búsqueda. Este informe explora las diferencias desde un punto de vista económico y de innovación tecnológica, que, aunque muy completo a su manera, deja fuera de la comparativa a productos muy relevantes en el ámbito empresarial debido a diversos criterios. Así, por ejemplo, este estudio ignora soluciones de código abierto alegando ciertas diferencias terminológicas subjetivas, que no terminan de convencer. Es por eso que estos estudios no consiguen explorar las diferencias relevantes que pueden existir entre estos productos.

1.2. Necesidad de la creación de una comparativa

Según lo expuesto anteriormente, en la actualidad los estudios de los motores de búsqueda se centran más en el producto, en evaluar lo bueno que puede ser desde un punto de vista del rendimiento o, cambiando el enfoque, desde la innovación tecnológica que aporta. Esto, en consecuencia, deja fuera de estas comparativas existentes a aquellos sistemas que no cumplen algún requisito imprescindible para el enfoque de estos análisis. Ejemplos de estas evaluaciones centradas en el producto comercial hechas por consultoras independientes existen y están disponibles para el que quiera acceder a ellas, tal y como hemos expuesto.

Es preciso pues buscar una comparativa que centre sus esfuerzos en los aspectos menos explorados de los motores de búsqueda, características relevantes que no precisan de un análisis numérico, sino de una evaluación objetiva y cualitativa.

1.3. Aportaciones del presente documento

Efectuar un estudio comparativo a gran escala de motores de búsqueda en general resulta muy costoso, en parte por la cantidad de elementos a evaluar, y en parte porque no siempre se dispone de toda la información en cuanto a productos comerciales se trata. En este documento se busca únicamente recoger, en la medida de lo posible, los aspectos más relevantes que pueden ser evaluados por cualquier persona que examine las especificaciones de esos productos, así como información de otras fuentes, y busca únicamente resaltar cuál de ellos se comporta mejor en ciertas situaciones.

La idea que subyace bajo este estudio es crear una comparativa poco convencional, en la que aparte de evaluar aspectos de los motores de búsqueda típicos, como el rendimiento, se busca examinar también otras características poco conocidas y tan importantes como las primeras. La disponibilidad de información, de documentación y de soporte determinan el valor de un producto casi tanto como su rendimiento. Aquellos programas que exigen una curva de aprendizaje muy pronunciada no suelen tener un gran éxito, principalmente si no vienen acompañados de un manual muy completo. En general, si a la hora de escoger entre varios sistemas, o entre varias soluciones, se evalúan más necesidades que la del mero rendimiento, a la hora de comparar cualquier conjunto de motores resulta necesario explorar todas esas características también.

Así pues, este documento busca aportar un punto de vista distinto a la comparativa de motores de búsqueda, de manera que puedan evaluarse los elementos fuertes y débiles de cada uno y, aparte de que en el documento se escoja un ganador, el lector pueda formar su propia opinión y ver cuál de ellos se adapta mejor a su situación. Asimismo, el benchmark de evaluación aportado puede adaptarse también, con lo que el usuario puede examinar aquellas características más relevantes para su caso.

La selección de métricas y motores del estudio se ha intentado generar de forma ejemplarizante, cubriendo el máximo número de criterios posible e intentando evitar análisis subjetivos, justificando cada decisión tomada.

1.4. Estructura del documento

El documento está formado por dos partes claramente diferenciadas. La primera recoge una comparativa entre 10 motores de búsqueda muy conocidos y empleados o bien en entornos empresariales o bien en la comunidad científica y académica. En general, no hay discriminación entre si son Open Source o no, y se ha intentado que ambos tipos de licencia tengan la misma presencia en el estudio, tal y como se ha dicho previamente. Para efectuar esta comparativa, se han escogido un conjunto de métricas fácilmente extraíbles por cualquier persona que acceda a las especificaciones de los motores de búsqueda, dividiéndolas en cuatro apartados: ejecución, operación, desarrollo y soporte. Dentro de aquellas relacionadas con la ejecución se exponen las que afectan al comportamiento del software durante su funcionamiento, su consumo y productividad. En las relacionadas

con la operación, se evalúan las características referidas a cualidades del comportamiento del producto, como el mantenimiento o la escalabilidad. Por su parte, las que engloba la categoría de desarrollo son aquellas que tienen que ver con las herramientas que provee el motor y su capacidad de integración en los sistemas, como la facilidad de aprendizaje o las herramientas disponibles. Por último, las métricas de soporte recogen la disponibilidad de recursos de ayuda a la solución de problemas o la continuidad de producto. El objetivo de este conjunto de medidas es elaborar un benchmark otorgando puntuaciones a cada uno de los motores en cada uno de estos aspectos, con el fin de extraer un ganador, o más bien, el producto más adecuado para un mayor número de situaciones.

En la segunda parte del documento, se empleará ese motor para construir una aplicación web que mostrará a través de un prototipo cómo integrar un motor de búsqueda con una aplicación de usuario. Para ello, desarrollaremos un repositorio web que incluya un sistema de búsqueda implementado con el producto anteriormente seleccionado.

Por último, a través de la comparativa y del caso práctico propuestos, extraeremos conclusiones sobre el desarrollo actual de los motores de búsqueda y qué trabajo futuro puede poseer un estudio como este.

1.5. Planificación del trabajo

A la hora de realizar la planificación del trabajo se dividió la funcionalidad a resolver en 4 partes con el fin de facilitar una separación entre ellas y establecer tiempos:

- Estudio de los motores de búsqueda y estado del arte
- Construcción del benchmark y las métricas que lo componen
- Recogida de datos y medidas
- Elaboración de una aplicación para evaluar la integración de los motores de búsqueda en aplicaciones de usuario

El trabajo se ha desarrollado a lo largo de tres meses, desde el día 1 del mes de abril de 2017, hasta el 12 de julio del mismo año. La planificación inicial estaba estimada para finalizar el día 1 de julio, con el fin de tener o bien un margen de tiempo relativo, por si había alguna dificultad o una planificación optimista, o bien para realizar una revisión más exhaustiva de los resultados y la memoria. Al disponer de tres meses como se ha dicho, se ha intentado repartir el número de horas a lo largo de la semanas de forma igualitaria, estimando realizar 25 horas de trabajo en cada una de ellas. Esto se ha debido principalmente a que las estimaciones de desarrollo de este tipo de proyectos están calculadas en unas 275 horas/hombre.

Cabe asimismo destacar que las jornadas de trabajo variaban según el día, en concreto por motivos laborales. Por eso y otras razones esa planificación del presente trabajo ha

sufrido modificaciones, por lo que en este apartado se aportará tanto la planificación estimada inicial, como la real que ha resultado tras esas variaciones.

En los apartados adjuntos se expone el total de horas hombre necesario para cubrir cada apartado, así como las fechas estimadas y reales que se han manejado.

1.5.1. Planificación inicial

La planificación inicial según los cuatro hitos descritos anteriormente era:

- Estudio de los motores de búsqueda y estado del arte: Del 3 de abril al 20 de abril, estimado en un total de 55 horas/hombre.
- Construcción del benchmark y las métricas que lo componen: 20 de abril al 7 de mayo, estimado en un total de 75 horas/hombre.
- Recogida de datos y medidas: 7 de mayo al 1 de junio, estimado en un total de 100 horas/hombre.
- Elaboración de una aplicación para evaluar la integración de los motores de búsqueda en aplicaciones de usuario: 1 junio al 1 de julio, estimado en un total de 45 horas/hombre.

1.5.2. Planificación final real

Esa planificación real ha sufrido un retraso de casi dos semanas, debido a dificultades encontradas durante el proceso, así como cuestiones ajenas al proyecto.

- Estudio de los motores de búsqueda y estado del arte: Del 3 de abril al 20 de abril, con 55 horas/hombre reales. En este apartado no hubo ningún retraso.
- Construcción del benchmark y las métricas que lo componen: 20 de abril al 18 de mayo, con 90 horas/hombre reales. Este hito se retrasó debido a que la estimación inicial se hizo planificando un estudio de 4 motores de búsqueda, cuando finalmente se ha realizado un estudio de 10.
- Recogida de datos y medidas: 18 de mayo al 15 de junio, con 120 horas/hombre reales. Este apartado sufrió un desajuste debido a la coincidencia existente entre el desarrollo de este trabajo y un examen final de asignatura.
- Elaboración de una aplicación para evaluar la integración de los motores de búsqueda en aplicaciones de usuario: 15 junio al 4 de julio, con 32 horas/hombre reales. Debido a que la presentación de los datos a los tutores sufrió un retraso en el hito anterior, se redujo el tiempo disponible para poder construir un prototipo con mayor cobertura y funcionalidad.

Como puede observarse, aunque las fechas sugieren una planificación poco ajustada, la estimación horas hombre ha sido muy similar en el cálculo inicial y en la planificación real, con la excepción del hito 4, que vió reducido su tiempo por proximidad con la fecha de entrega y la recogida de medidas, que requirió de más horas de lo esperado.

Capítulo 2

Estado del arte de los motores de búsqueda

2.1. Introducción

En la actualidad, a la hora de aportar soluciones de almacenamiento y recuperación de documentos tanto en aplicaciones web como en otro tipo de entornos se suele confiar en un motor de búsqueda. Tal y como se ha comentado en apartados anteriores, la elección de este producto puede depender de otros factores que no sean los meramente relacionados con el rendimiento. Sin embargo, para entender los sistemas a que se van a aplicar estos análisis se precisa primero de comprender el funcionamiento y los procesos que permiten recuperar la información al motor de búsqueda y presentársela al usuario.

2.2. Funcionamiento de un motor de búsqueda

Un motor de búsqueda, como se ha expuesto anteriormente, es un programa que permite localizar en un conjunto de documentos aquellos relacionados con una palabra o palabras clave seleccionadas. Para ofrecer esta funcionalidad, exploran los ficheros existentes en el sistema de forma muy eficiente, registrando información relevante sobre ellos en una base de datos. Esta es empleada por el motor para permitir a los usuarios realizar consultas y localizar esos archivos de forma más sencilla. Un ejemplo clásico pero efectivo sería el funcionamiento de un motor de búsqueda en Internet. Éste explora el conjunto de páginas web, viajando entre enlaces y construye un mapa que refleja la estructura de las conexiones entre los recursos. La información que se recoge en este proceso permite al motor al recibir una consulta mostrar los resultados que mejor se adapten a la misma, así como sus elementos relacionados.

Así pues, para conseguir esta funcionalidad, existen varios componentes principales que permiten al motor de búsqueda realizar estas operaciones:

- **Crawler:** Así se denomina al proceso que recorre el conjunto de documentos almacenados, ya sea en una máquina o en una red, recogiendo los metadatos y la ubicación de los mismos, para posteriormente guardar un registro de todos ellos y permitir su rápida localización y presentación al usuario.

En colecciones de documentos reducidas, esta operación no supone una gran carga-bilidad al proceso. Sin embargo, en entornos como la web, se precisa de un recorrido continuo de enlaces web, dado que la situación de los elementos es más dinámica y varía con el tiempo.

- **Índice:** El índice se puede describir como el conjunto de información recopilado por el crawler y que se almacena en un registro, denominándose, valga la redundancia, índice de búsqueda. Estos datos son los que emplea posteriormente el motor para, a través de las palabras clave que conforman la consulta, escoger los elementos que cumplen con la petición del usuario.

La importancia del índice es muy elevada. Las grandes soluciones de búsqueda en el mercado funcionan de forma prácticamente similar en el fondo, y algo que distingue a estos productos entre sí es la manera de manipular este índice o registro y el método de acceso al mismo, aunque no es la principal y única diferencia. Así pues, este elemento es como el corazón del motor de búsqueda: si el índice se actualiza frecuentemente, será necesario realizar ordenaciones de los registros de forma continuada, lo que repercutirá en el rendimiento del producto. Sin embargo, si este se actualiza de forma poco frecuente, es posible que haya circunstancias en las que no se ofrezcan al usuario datos actualizados, con la consiguiente penalización.

- **Algoritmo de ordenación:** Si el índice es el corazón del motor de búsqueda, el algoritmo de ordenación es el cerebro de la operación del sistema. Es el responsable de recoger los datos del índice de la forma más adecuada posible, devolviendo los resultados a las consultas del usuario, cosa relativamente compleja. Se le exige además que los datos más relevantes sean siempre los primeros en aparecer, siguiendo criterios de ordenación diversos. Esos criterios, en ocasiones, se permite que sean personalizables bajo ciertos requisitos, dotando al algoritmo de cierta flexibilidad y adaptación. Esto permite al motor adaptarse a las necesidades reales del entorno empresarial o del usuario.

Así pues, el algoritmo junto con el índice son los elementos diferenciadores que actualmente permiten la existencia de una gran diversidad de productos. Sin embargo, las verdaderas diferencias entre ellos no se encuentran en estos elementos, por lo que los distribuidores y desarrolladores actualmente enfocan los esfuerzos de la evolución del producto y el fomento de la competencia hacia elementos como la calidad del servicio o el soporte.

2.3. Descripción de los motores de búsqueda estudiados

A continuación se describen los datos generales de los motores de búsqueda considerados para la comparativa. Se han elegido de forma representativa siguiendo los siguientes criterios:

- **Uso:** los motores han sido seleccionados de entre aquellos más utilizados en el ámbito empresarial, personal y educativo.
- **Representatividad:** asimismo, se ha intentado que muestren características y elementos diferenciales entre si, para evitar semejanzas en ninguno de ellos.
- **Licencia:** por último, otro de los criterios ha sido el coste de la licencia, así como si es de código abierto o privativo, para poseer una visión completa de elementos de una y otra parte, con sus ventajas e inconvenientes.

Todas y cada una de estas cartas de presentación recogen exclusivamente las características más importantes de los productos. Para más información sobre cada uno de ellos, se adjunta el sitio web o un enlace a un archivo descriptivo.

2.3.1. Algolia

- **Nombre:** Algolia's Search.
- **Características:** Motor de búsqueda de tipo web.
 - **Requisitos hardware:** No precisa dado que es un servicio en la nube.
 - **Requisitos software:** Requiere de una cuenta en la plataforma web de Algolia.
 - **Licencia:** Privada.
 - **Información web:** <https://www.algolia.com/>

- **Descripción breve:**

Algolia's Search es un producto que se ofrece como un SaaS (Software as a Service), un servicio al que se accede a través de una plataforma web y que permite al usuario subir colecciones de documentos y realizar consultas sobre ellos.

Para interactuar con la plataforma, aparte del propio servicio web el servicio ofrece APIs (Interfaz de Programación de Aplicaciones) de acceso para poder acceder desde otras aplicaciones a los recursos de Algolia.

2.3.2. Amazon CloudSearch

- **Nombre:** Amazon Cloud Search.
- **Características:** Motor de búsqueda para soluciones software con servicio cloud.
 - **Requisitos hardware:** No precisa dado que es un servicio en la nube.
 - **Requisitos software:** Requiere de una cuenta en la plataforma Amazon Web Services.
 - **Licencia:** Privada.
 - **Información web:** <http://aws.amazon.com/>

- **Descripción breve:**

Amazon Cloud Search se oferta desde Amazon Web Services, el producto estrella de computación de servicios en la nube de la compañía Amazon. Este motor de búsqueda permite, tras registrarse en la plataforma, subir documentos y realizar búsquedas sobre los mismos.

Para construir aplicaciones que empleen los recursos de Amazon Cloud Search, la plataforma provee de una API que permite el desarrollo de aplicaciones diversas, aunque principalmente de tipo web.

2.3.3. Apache Solr

- **Nombre:** Apache Solr.
- **Características:** Motor de búsqueda de código abierto.
 - **Requisitos hardware:** Requiere de una máquina o servidor local para su ejecución.
 - **Requisitos software:** Tener instalada la JVM (Java Virtual Machine) en el sistema.
 - **Licencia:** Código abierto gratuita.
 - **Información web:** <http://lucene.apache.org/solr/>

- **Descripción breve:**

Solr es un motor de búsqueda veterano de código abierto, basado en la librería de búsqueda escrita en Java, Lucene. Al haber sido desarrollado durante varios años y con las colaboraciones de la comunidad de usuarios tan potente que tiene, ha conseguido pulir su funcionalidad casi al máximo.

Solr además provee de numerosas APIs para realizar desarrollos que precisen del uso del motor de búsqueda para recuperar información y es muy sencillo trabajar con él gracias a su funcionalidad transparente al usuario, pero sin perder el potencial del código abierto.

2.3.4. Crate

- **Nombre:** Crate DB.
- **Características:** Motor de búsqueda basado en solución de base de datos.
 - **Requisitos hardware:** Requiere de una máquina o servidor local para su ejecución.
 - **Requisitos software:** No precisa de software adicional para su instalación.
 - **Licencia:** Privada.
 - **Información web:** <https://crate.io/>

- **Descripción breve:**

Crate no es un motor de búsqueda puro. Su diseño está pensado para, a través de clásicas consultas SQL, almacenar y recuperar datos estructurados y no estructurados, entre los que podemos encontrar documentos. Es ahí donde Crate da un giro a su funcionalidad principal y permite que pueda usarse para almacenar archivos de diferente tipo, para realizar búsquedas sobre los mismos. De esta manera, lo que en principio no estaba pensado para usarse así pasó a emplearse como un motor de búsqueda más, hasta el punto de dar la cara mejor que muchos otros.

Crate se ejecuta en la máquina del usuario, y puede funcionar de forma distribuida transparente al usuario. Además, provee de APIs de acceso al servicio que permiten el desarrollo de aplicaciones que emplean este producto para cubrir su funcionalidad.

2.3.5. Elasticsearch

- **Nombre:** Elasticsearch.
- **Características:** Motor de búsqueda distribuido de código abierto.
 - **Requisitos hardware:** Requiere de una máquina o servidor local para su ejecución.
 - **Requisitos software:** Precisa de tener instalada la JVM (Java Virtual Machine).
 - **Licencia:** Código abierto con soluciones comerciales.
 - **Información web:** <https://www.elastic.co/>

- **Descripción breve:**

ElasticSearch es un clásico motor de búsqueda Open Source. Al igual que su hermano, Solr, está basado en la librería de búsqueda de Java, Lucene. Sin embargo, ElasticSearch ha centrado su desarrollo en optimizar su funcionamiento de forma distribuida, y ha intentado ofrecer siempre exclusivamente la funcionalidad justa para operar como motor de búsqueda de forma óptima. Esto quiere decir que, aunque provee de numerosas APIs de desarrollo, intenta dejar la funcionalidad ajena al proceso de búsqueda a elección del usuario. Esto lleva a que servicios de monitorización y seguridad tengan que ser implementados por el que adquiere este producto.

Aún así, Elasticsearch es un duro competidor de cualquier motor de búsqueda, debido a lo bien construido que está y a una gran comunidad de usuario que respalda el proyecto hoy en día.

2.3.6. Google Search Appliance

- **Nombre:** GSA (Google Search Appliance).
- **Características:** Motor de búsqueda que recopila datos de fuentes tanto internas como externas a través de Google.
 - **Requisitos hardware:** Precisa de adquirir hardware específico cuya potencia varía dependiendo del número de documentos a indexar.
 - **Requisitos software:** No precisa de requisitos software adicionales.
 - **Licencia:** Privada.
 - **Información web:** <https://enterprise.google.es/intl/es/search/>

- **Descripción breve:**

Google Search Appliance es un motor de búsqueda alojado en un sistema hardware diseñado por Dell. Esto quiere decir que el producto funciona con módulos hardware propios, cuyo software instalado diseña Google. Este motor permite búsquedas tanto a nivel interno de la biblioteca de documentos del usuario o empresa, como en toda la red, lo que da una visión de lo útil que puede llegar a ser.

A pesar de ello, la dependencia de módulos hardware del producto así como los costes de mantener un sistema de estas características permiten su adquisición a unos pocos. Además, existen rumores confirmados de que el producto va a desaparecer para sustituirse por una solución cloud. No obstante, es un motor que se hace respetar, con una gran potencia de búsqueda e indexación de documentos.

2.3.7. Indica ES

- **Nombre:** Indica Enterprise Search.
- **Características:**
 - **Requisitos hardware:** No precisa dado que es un servicio en la nube.
 - **Requisitos software:** Requiere un registro en la plataforma de Indica ES.
 - **Licencia:** Privada.
 - **Información web:** <http://www2.indica.nl/enterprisearch>

- **Descripción breve:**

Indica ES es un producto controvertido. En sí es relativamente nuevo (surgió en 2013) pero desde hace tiempo apunta maneras. En general los comentarios que recibe son

positivos y se va abriendo un hueco en el mercado. El producto en sí se ofrece como un SaaS (Software as a Service) al que se accede registrándose en la plataforma.

Sin embargo, dado que aún está empezando posee ciertos aspectos a mejorar, como la presencia en foros y sitios web, o la propia información aportada por la empresa. Aún así, resulta relevante en una comparativa evaluar a los nuevos motores, para analizar sus posibilidades actuales y futuras.

2.3.8. Microsoft Azure Search

- **Nombre:** Microsoft Azure Search
- **Características:**
 - **Requisitos hardware:** No precisa dado que es un servicio en la nube.
 - **Requisitos software:** Precisa de la creación de una cuenta en la plataforma web, además de datos bancarios.
 - **Licencia:** Privada.
 - **Información web:** <https://azure.microsoft.com/es-es/services/search/>

- **Descripción breve:**

Microsoft Azure Search es otra buena idea de Microsoft. No será famosa por inventar algo nuevo, dado que existen muchas otras plataformas que proveen de búsqueda de documentos como servicio web, pero sí por hacer las cosas bien. A pesar de su licencia privada bajo pago, se permite a los usuarios que no precisan de un servicio exigente operar sin ningún gasto. Además, la plataforma está accesible en una gran cantidad de idiomas, y tiene cierta presencia en foros y páginas web.

En cuanto al desarrollo de aplicaciones, Microsoft Azure Search provee de una API muy potente, que permite que los servicios que lo requieran accedan a la plataforma de forma rápida y sencilla.

2.3.9. SearchBlox

- **Nombre:** SearchBlox.
- **Características:**
 - **Requisitos hardware:** No precisa dado que es un servicio en la nube.
 - **Requisitos software:** Precisa de la creación de una cuenta en la plataforma web para poder usar el servicio.
 - **Licencia:** Privada.
 - **Información web:** <https://www.searchblox.com/>

- **Descripción breve:**

SearchBlox es un producto bien construido, que sus distribuidores definen como el mejor sustituto para Elasticsearch o Google Search Appliance, lo cual puede ser cierto o no. Lo que sí que es verdad es que el servicio se ofrece como búsqueda en la nube, al igual que entornos como Microsoft Azure Search o Algolia. Sin embargo, ofrecen también la instalación del servicio en local, aunque en cierto tipo de servidores y cuya versión no evaluaremos aquí.

El servicio provee así mismo de un potente sistema de búsqueda accesible a través de una API para los desarrollos, permitiendo a los usuarios que lo deseen emplear este servicio en caso de necesitarlo.

2.3.10. Sphinx Search Server

- **Nombre:** Sphinx Search.

- **Características:**

- **Requisitos hardware:** Requiere de una máquina o servidor local para su ejecución.
- **Requisitos software:** No requiere de software preinstalado para su funcionamiento.
- **Licencia:** Código abierto con soluciones comerciales.
- **Información web:** <http://sphinxsearch.com/>

- **Descripción breve:**

Sphinx es uno de los motores de búsqueda de código abierto que se exponen en el presente documento. A pesar de ser muy potente, es relativamente complejo de manejar, debido a que ha sido diseñado para ser altamente configurable y porque la mayor parte de la documentación y recursos que aporta son muy escasos. Sin embargo, es un motor de búsqueda muy completo y versátil, pudiendo actuar en ocasiones de gestor de bases de datos, y con una fuerte comunidad detrás, que lo mantiene actualizado.

En cuanto a su integración con aplicaciones web, provee de una API en PHP utilizada en desarrollos web, así como manuales y herramientas para el diseño de APIs propias por parte de los usuarios.

Capítulo 3

Descripción del benchmark construido

3.1. Introducción

El benchmark generado para ser aplicado en la evaluación de motores de búsqueda provee de un conjunto de métricas cualitativas. El motivo de este enfoque se basa en que a la hora de escoger un sistema, no siempre se dispone de la opción de aplicar al producto un conjunto de métricas cuantitativas y obtener así parámetros numéricos que lo caractericen. Además, y especialmente en el ámbito empresarial, no resulta interesante evaluar un sistema simplemente por su rendimiento, sino que otros elementos, como el costo de la licencia, pueden influir en la decisión que se tome. Esto obliga a que otros beneficios extra que pueda ofrecer el servicio no puedan ser evaluados de forma numérica, y por tanto, ser discriminados en una comparativa clásica.

Así pues, las métricas escogidas buscan evaluar los motores de búsqueda cualitativamente de forma objetiva, dentro de los límites que este enfoque pueda dar. Así mismo, buscan resaltar aquellos elementos más relevantes para una empresa que busque adquirir el producto.

Para categorizar y clasificar las métricas, estas se han dividido en cuatro áreas diferentes, cuatro enfoques que permiten examinar lo completo que resulta el motor de búsqueda en cada una de ellos. Esto provee a la comparativa de una ventaja importante: en caso de que el que desee aplicar el benchmark a su conjunto de posibles productos desee evaluar lo bien que cubre cada una de las áreas de forma independiente, podrá escoger qué áreas son más relevantes para ellos, y cuáles desea ignorar.

En el siguiente apartado se explican cada unas de estas cuatro áreas, así como las métricas que componen el benchmark. Además, en un apartado posterior se exponen los valores posibles que pueden asignarse a los productos por cada métrica, así como

los motivos que han llevado a esas conclusiones. Las puntuaciones van en una escala de porcentaje, para evidenciar la cobertura de lo que se desea medir.

3.2. Métricas a nivel ejecución

Las métricas a nivel de ejecución engloban todos aquellos datos relacionados con el funcionamiento del motor de búsqueda cuando está realizando sus funciones, qué requisitos son necesarios para que ese funcionamiento se lleve a cabo y qué elementos extra se proveen para mejorarlo. En este apartado se han escogido las siguientes métricas:

3.2.1. Productividad

Evalúa el grado de cumplimiento de la funcionalidad esperada por el motor de búsqueda, es decir, su utilidad, no el concepto de productividad clásico de la informática. En esencia, el motor de búsqueda debe permitir indexar gran cantidad de documentos y permitir cierta personalización en el algoritmo de indexación para adaptar las búsquedas a las necesidades de la empresa. Con este fin, se otorgan las siguientes puntuaciones dependiendo de la cobertura:

- **100 %:** El producto ofrece alta capacidad de indexación y personalización en la búsqueda de resultados.
- **75 %:** Alta capacidad de indexación y algún tipo de personalización en la búsqueda de resultados.
- **50 %:** Baja capacidad y alta personalización en la búsqueda de resultados o viceversa.
- **25 %:** Solamente alta capacidad o alta personalización en la búsqueda de resultados.
- **0 %:** Muy baja capacidad de indexación.

3.2.2. Requisitos de ejecución

Recoge los elementos o características necesarias en el funcionamiento del sistema. En general evalúa si funciona como un servicio externo, si requiere de una ejecución en máquina local o requiere hardware adicional para su ejecución. Para categorizar esta métrica se propone la siguiente escala:

- **100 %:** No requiere datos personales del usuario para operar con datos en plataformas cloud. El sistema se ofrece como servicio en la nube de acceso libre.

- **75 %:** Requiere cuentas y subidas de datos a plataformas cloud.
- **50 %:** Requiere de un sistema local en el que ejecutar el proceso.
- **25 %:** Requiere estar conectado a un sistema local y a un proceso de monitorización o identificación en red.
- **0 %:** Requiere hardware adicional para su ejecución.

3.2.3. Consumo de energía

Es la medida de la energía necesaria para mantener el entorno de ejecución del producto operativo. Evidentemente, cuando el servicio es externo o se ofrece como SaaS (Software as a Service) no se precisa de ningún gasto de energía adicional, mientras que si funciona localmente o requiere de hardware para su funcionamiento supondrá un consumo extra para el usuario o empresa. De esa manera, reflejamos esa idea según el siguiente planteamiento:

- **100 %:** El servicio es externo y no requiere de consumo extra.
- **75 %:** El servicio es externo pero requiere pago por albergar la máquina del cliente (hosting).
- **50 %:** El servicio se ejecuta en local.
- **25 %:** El servicio se ejecuta en local bajo condiciones especiales (hardware exigente).
- **0 %:** El servicio requiere de hardware adicional externo aparte de la propia infraestructura del cliente.

3.2.4. Multi-plataforma

Es la disponibilidad del motor de búsqueda para operar en diferentes plataformas o sistemas operativos. Idealmente, el producto debería estar disponible para cualquier sistema operativo o prácticamente cualquiera y, en el peor caso, exclusivamente para uno solo. Así pues, siguiendo este criterio se asignan las puntuaciones a la siguiente escala:

- **100 %:** El servicio puede ejecutarse en cualquier sistema operativo o funcionar con él.
- **75 %:** Funciona en Windows y Linux. También en otros sistemas UNIX puntuales.
- **50 %:** Funciona en cualquier sistema UNIX.
- **25 %:** Funciona en sistemas UNIX concretos.
- **0 %:** Funciona exclusivamente para un tipo de sistema operativo.

3.2.5. Seguridad

Evalúa si el producto aporta mecanismos de protección de datos y denegación de acceso al servicio a personas no autorizadas. También recoge el grado de integración de la seguridad del cliente dentro del servicio. Se busca que el servicio posea un gran nivel de defensa propio, penalizando el que no aporte seguridad propia ni integre la del cliente:

- **100 %:** El servicio aporta su propia seguridad además de integrar por completo la propia del cliente.
- **75 %:** El servicio aporta su propia seguridad sin integrar por completo la propia del cliente.
- **50 %:** El servicio aporta plug-in de seguridad dependientes de programas externos.
- **25 %:** El servicio acepta elementos de seguridad del cliente pero no aporta propios.
- **0 %:** El servicio no aporta ningún tipo de seguridad ni integra la del cliente.

3.3. Métricas a nivel operación

Las métricas a nivel de operación engloban todas aquellas medidas referentes a los procesos de monitorización, mantenimiento y calidad del servicio que ofrece el motor de búsqueda. Este enfoque se cubre con las siguientes métricas:

3.3.1. Monitorización

Mide la funcionalidad por parte del motor de búsqueda de analizar a tiempo real y posteriormente el estado y resultados de ejecución del proceso. Resulta interesante comprobar si el propio producto ofrece ese servicio de forma integrada, o por el contrario precisa de programas externos para llevar a cabo esa funcionalidad. Siguiendo este criterio, establecemos la siguiente escala:

- **100 %:** Monitorización del servicio, su estado, incidencias, accesos, usuarios y estadísticas de la información accedida.
- **75 %:** Monitorización del servicio, su estado, incidencias, accesos, usuarios y estadísticas puntuales de la información accedida.
- **50 %:** Monitorización genérica del estado del servicio e incidencias o bien de la información accedida.
- **25 %:** No existen procesos de monitorización pero pueden crearse e integrarse por el usuario.

- **0 %:** No existen procesos de monitorización ni pueden crearse e integrarse por el usuario.

3.3.2. Mantenimiento

Estudia la disponibilidad de administrar los sistemas internamente y disponer de parches software por parte del distribuidor del producto. Lo ideal en el entorno empresarial y en concreto de las TIC es disponer de la posibilidad de realizar mantenimiento interno, pero mantener el apoyo del externo por parte del proveedor. En este apartado se ha creído conveniente evaluar la métrica con tres posibles valores, dado que no existen más estados a analizar.

- **100 %:** Posibilidad de mantener y administrar los sistemas internamente y disponer asimismo de mantenimiento externo gratuito. Se valora positivamente la administración compartida, de forma que el propio usuario o empresa pueda administrar su propio servicio y datos y obtener soporte externo en caso de necesitarlo.
- **50 %:** Posibilidad de mantener y administrar los sistemas internamente, o bien interna y externamente bajo pago.
- **0 %:** Posibilidad de disponer de mantenimiento externo para los sistemas. Imposibilidad de administración del sistema por parte del usuario o empresa.

3.3.3. Escalabilidad

Analiza la facilidad del motor de búsqueda para adaptarse al aumento de la demanda de servicio sin perder calidad ni verse afectado en su operación. Es importante que esa escalabilidad esté diseñada de manera que no sea necesario realizar modificaciones en el servicio para que pueda seguir operando con normalidad. Dado que la escalabilidad es un concepto numérico, aquí se evalúan las características que permiten el funcionamiento de esa escalabilidad siguiendo los siguientes criterios:

- **100 %:** Solución altamente escalable sin añadido de módulos adicionales y reordenación de índices transparente al usuario. Es decir, que la configuración de la escalabilidad sea independiente de la actuación manual del usuario.
- **75 %:** Solución altamente escalable dependiente de prestaciones bajo pago. En esencia, esto significa que a mayor dinero pagado a la plataforma, mayor cantidad de recursos se aportarán al sistema.
- **50 %:** Solución altamente escalable dependiente de configuraciones manuales.
- **25 %:** Solución altamente escalable dependiente de adición de módulos hardware adicionales en situaciones extremas.
- **0 %:** Solución poco escalable.

3.3.4. Arquitectura

Analiza la capacidad del producto para adaptarse a una disposición distribuida o centralizada dependiendo de las necesidades del usuario. Esta configuración debe ser dependiente del usuario o, por el contrario, transparente para él, ahorrándole tiempo y configuraciones. Así pues, analizando esta propiedad asignaríamos las puntuaciones de la siguiente manera:

- **100 %:** Arquitectura distribuida o centralizada a elección del usuario con control de fallos, balanceo de carga y mecanismos de recuperación; o bien arquitectura transparente al usuario (servicio cloud).
- **75 %:** Arquitectura exclusivamente distribuida con balanceo de carga y control de fallos.
- **50 %:** Arquitectura exclusivamente distribuida.
- **25 %:** Arquitectura exclusivamente centralizada con control de fallos.
- **0 %:** Arquitectura exclusivamente centralizada sin control de fallos.

3.4. Métricas a nivel desarrollo

En el apartado de desarrollo se agrupan aquellas métricas relacionadas con la inclusión de herramientas por parte del motor de búsqueda que faciliten o hagan posible su integración en proyectos o sistemas. En este apartado pues, tendremos los siguientes elementos:

3.4.1. Curva de aprendizaje

Examina la complejidad existente a la hora de dominar las herramientas que provee el producto, así como la cantidad de las mismas. Cuantas menos herramientas y complejidad haya que dominar, y mayor cantidad de recursos se provean por parte del producto para ayudar a hacerlo, mejor se cubrirá esta métrica:

- **100 %:** Muy suave. Recursos de aprendizaje extensos y muy precisos. Poca complejidad que dominar.
- **75 %:** Suave. Recursos poco extensos y muy precisos. Poca complejidad que dominar.
- **50 %:** Media. Recursos extensos. Elevada cantidad de herramientas que dominar.
- **25 %:** Alta. Poca cantidad de recursos disponible.
- **0 %:** Muy alta. Recursos mínimos o inexistentes.

3.4.2. Independencia del servidor

Mide la capacidad del motor de búsqueda para ser transportado a servidores distintos sin afectar a su comportamiento o necesitar algún tipo de adaptación. Al igual que en alguna métrica anterior, se contemplan exclusivamente tres valores para analizar. Cuanto más transportable sea el producto, mejor puntuación se le dará en la siguiente escala:

- **100 %:** El servicio es totalmente transportable, sin modificar su configuración o adaptarla.
- **50 %:** Puede ser portado a sistemas concretos cambiando algunos parámetros de configuración.
- **0 %:** El servicio es dependiente del servidor. Los servicios en la nube se engloban asimismo en esta categoría.

3.4.3. Integración con otros sistemas

Analiza la facilidad del producto para interactuar y compartir datos con otros programas y sistemas. También, métodos que provee para poder emplearlo en aplicaciones y otros lugares en los que se le necesite. Ese grado de integración se mide siguiendo estos criterios:

- **100 %:** API genérica. Cobertura para gran cantidad de sistemas y procesos.
- **75 %:** API adaptada a gran cantidad de sistemas y procesos.
- **50 %:** Existe una API en desarrollo o en fase de pruebas.
- **25 %:** El sistema se puede integrar de alguna forma con procesos concretos, a través de desarrollos por parte del usuario.
- **0 %:** El sistema funciona de forma independiente y no puede integrarse con otros procesos, o bien no se hace mención a ningún método de hacerlo.

3.5. Métricas a nivel soporte

En el área de soporte se engloban aquellas métricas que, aunque no relacionadas directamente con el funcionamiento del producto, facilitan su uso en cuanto a resolución de problemas, disponibilidad del producto en el futuro y el coste que supone adquirirlo y cubrir la licencia. Para categorizar este enfoque se han escogido las siguientes observaciones:

3.5.1. Soporte del proveedor

Evalúa la disponibilidad del proveedor del producto para apoyar al usuario a la hora de resolver problemas o dificultades en cuanto al uso. La cobertura de esta utilidad se mide siguiendo los siguientes criterios:

- **100 %:** Numerosas vías de comunicación diferentes. Atención personalizada por parte del proveedor.
- **75 %:** Numerosas vías de comunicación, sin atención personalizada.
- **50 %:** Existe algún método de soporte al usuario.
- **25 %:** Existe soporte bajo ciertos problemas o circunstancias, como errores o situaciones excepcionales. También soporte dependiente de la cantidad de dinero pagado.
- **0 %:** No existe ningún tipo de soporte por parte del proveedor.

3.5.2. Soporte online y comunidad de usuarios

Mide la existencia de una comunidad online con experiencia en el trabajo con el motor de búsqueda y las necesidades que los usuarios han tenido a lo largo de la vida del producto. A mayor cantidad de entradas independientes en foros conocidos y publicaciones, mayor cobertura se hace de la métrica:

- **100 %:** Numerosas fuentes de información externa al distribuidor. Comunidad amplia y presente en numerosos foros ajenos a la plataforma o el distribuidor.
- **75 %:** Existen fuentes de información externa al distribuidor. Hay personas concretas que hacen aportes al servicio. También presencia en foro con enlaces patrocinados.
- **50 %:** Algunos foros se hacen eco de ciertos elementos puntuales relacionados.
- **25 %:** Apenas existen fuentes de información externas al distribuidor.
- **0 %:** No existe ningún tipo de soporte externo o comunidad.

3.5.3. Documentación

Examina el conjunto formado por la recopilación de textos y el material gráfico sobre el funcionamiento del producto, útiles para comprender su funcionamiento y uso. Su cobertura se mide tanto desde un enfoque interno como externo, dado que es relevante la existencia de documentación ajena al proveedor:

- **100 %:** Documentación interna y externa abundante.
- **75 %:** Documentación interna y externa disponible.
- **50 %:** Sólo documentación interna o bien interna y externa muy limitada.
- **25 %:** Sólo documentación interna y limitada.
- **0 %:** No existe ningún tipo de documentación disponible.

3.5.4. Licencia del producto

Mide el impacto del precio de la licencia, y diferencia entre aquellas que son de código abierto gratuitas o evalúa a través de un gradiente aquellas que son privativas.

- **100 %:** Código abierto y gratuita.
- **75 %:** Código abierto con soluciones comerciales en cuanto al soporte o al servicio.
- **50 %:** Privada gratuita bajo ciertas circunstancias, como por ejemplo, cuotas de uso.
- **25 %:** Privada bajo pago con demo temporal gratuita disponible para descargar.
- **0 %:** Privada bajo pago sin demo.

3.5.5. Continuidad del producto

Evalúa la continuidad de la versión actual del producto y cómo evolucionará en el futuro. Se penaliza que el proveedor del producto no busque la continuidad del mismo, así como que el mismo esté en fase de mantenimiento o fusión con otro motor. En este apartado se analizan exclusivamente tres valores, derivados de los posibles estados en los que puede encontrarse el producto.

- **100 %:** El producto está en fase de continuidad.
- **50 %:** El producto se encuentra en fase de integración con otros sistemas o con una versión en mantenimiento.
- **0 %:** Se tiene pensado discontinuar el producto.

3.5.6. Multi-idioma

Mide la cantidad de lenguas diferentes que pueden emplearse para efectuar búsquedas en el motor o presentar los resultados. Resulta relevante dar la puntuación más alta a aquellos productos que provean de un mayor número de lenguas diferentes.

- **100 %:** Se oferta con un amplio abanico de idiomas.
- **75 %:** Disponible en los idiomas más extendidos.
- **50 %:** Disponible en algún idioma aparte del inglés
- **25 %:** Sólo disponible en inglés.
- **0 %:** No posee versión en inglés.

Capítulo 4

Estudio comparativo de los motores de búsqueda

4.1. Introducción

Como se ha podido observar en el apartado anterior, las métricas recogidas se han dividido en cuatro áreas diferentes. En este apartado, se recoge una comparativa para cada una de esas áreas, ofreciendo al final una comparativa global. Asimismo, en cada enfoque propuesto se indica cuál es el motor ganador, así como el vencedor la comparativa total en un apartado final.

A cada motor se le asigna la puntuación o cobertura que ofrece para la característica a medir. Esta puntuación se muestra en porcentaje, para reflejar lo que el sistema aporta sobre el 100 % de lo esperado, que sería la puntuación máxima. De esta manera, si cumple los requisitos para recibir una puntuación, reflejados en los apartados anteriores, esa será la que se muestre aquí.

Además, se le asigna a cada métrica un peso dependiendo de la importancia que se le da a esa característica, para que en la nota final predominen aquellas puntuaciones de mayor importancia.

Estos pesos recogen valores de uno a tres, donde uno es que la característica es recomendable, dos que es necesaria y tres que se considera imprescindible.

De esta manera, realizando la multiplicación del peso con la puntuación, podemos extraer qué parte de la métrica cubre, y, sobre la puntuación final en cada apartado (suma de los pesos), qué nota se le otorga al motor en ese área de estudio.

Además, y para concluir, bajo cada cuadro resumen se explican los motivos que han llevado a otorgar al motor esa puntuación, para que se vea que se ha realizado de la forma

más objetiva posible. Toda la información proviene de las propias características de los productos, y se consideran de fiar, dado que el propio fabricante está obligado a dar fe de que esos valores son correctos y demostrables.

Aquellos que no pueden extraerse de las definiciones de los fabricantes, se han obtenido de fuentes con cierta reputación, o tras un análisis explorativo de documentos y páginas web.

Cabe destacar que las explicaciones son lo más sucintas posibles, y se ha creído adecuado exponer los conceptos utilizados y que son desconocidos de las explicaciones en el apartado Palabras Clave en el capítulo 8 del documento.

4.2. Comparativa a nivel de ejecución

4.2.1. Algolia

Métricas	Pesos	Puntuación	Resultado
Productividad	2	50 %	1
Requisitos de ejecución	1	75 %	0,75
Consumo de energía	1	100 %	1
Multi-plataforma	2	100 %	2
Seguridad	3	100 %	3
TOTAL	9		7,75

Tabla 4.1: Ejecución - Algolia

- **Productividad:** El motor provee un alto grado de indexación de documentos, aunque limitado por la licencia a pagar. La personalización y regulación de la jerarquía de búsqueda son limitadas, pues son bajo unos ajustes predeterminados, eso sí, a disposición del usuario.
- **Requisitos de ejecución:** El servicio se oferta en la nube. Requiere una cuenta en la web de Algolia bajo registro, con datos personales. Además, es necesario subir el set de datos a la plataforma para poder emplear el motor de búsqueda.
- **Consumo de energía:** El software reside en la plataforma de Algolia y se accede a través de la web, por lo que no precisa de una máquina de ejecución del proceso.
- **Multi-plataforma:** Se accede al servicio a través de la web, lo que le da cobertura global para todos los sistemas con conexión a Internet.
- **Seguridad:** El servicio opera a través del protocolo HTTPS. Además, dispone a elección del usuario de ACLs y autenticación a través de claves 2FA. Asimismo, como servicio web puede coexistir con la seguridad extra que aporte el usuario en su máquina local de acceso.

4.2.2. Amazon Cloud Search

Métricas	Pesos	Puntuación	Resultado
Productividad	2	50 %	1
Requisitos de ejecución	1	75 %	0,75
Consumo de energía	1	100 %	1
Multi-plataforma	2	100 %	2
Seguridad	3	100 %	3
TOTAL	9		7,75

Tabla 4.2: Ejecución - ACS

- **Productividad:** Amazon Cloud Search provee de un alto grado de indexación de documentos, restringido al coste de la licencia. La personalización del algoritmo de búsqueda es limitada, pues se realiza bajo unos ajustes predeterminados y no dispone de otros mecanismos de modificación.
- **Requisitos de ejecución:** Al igual que Algolia requiere una cuenta en una plataforma, en este caso, Amazon Web Services. Además, es necesario subir el set de datos a la plataforma para poder emplear el motor de búsqueda.
- **Consumo de energía:** El software reside en la plataforma de AWS y se accede a través de ella. Por lo tanto, no requiere de una máquina local que ejecute el proceso.
- **Multi-plataforma:** Funciona en todos los sistemas con acceso a Amazon Web Services.
- **Seguridad:** Se accede al servicio a través de HTTPS. Además, aporta el servicio IAM (Identity and Access Management), que es el sistema de identificación que posee la plataforma, y autenticación a través de firma. Todo mecanismo de seguridad que posea el cliente en su máquina de acceso funciona con normalidad.

4.2.3. Apache Solr

Métricas	Pesos	Puntuación	Resultado
Productividad	2	100 %	2
Requisitos de ejecución	1	50 %	0,5
Consumo de energía	1	50 %	0,5
Multi-plataforma	2	100 %	2
Seguridad	3	50 %	1,5
TOTAL	9		6,5

Tabla 4.3: Ejecución - Solr

- **Productividad:** El motor ofrece la posibilidad de indexar miles de millones de documentos a través de cientos de nodos distribuidos. Permite asimismo al usuario

modificar el algoritmo de ordenación al proveerse como software de código abierto, o adaptarse con la modificación de variables de configuración.

- **Requisitos de ejecución:** Precisa de tener instalada la versión Java 8 o superior en una máquina local.
- **Consumo de energía:** La empleada por el propio sistema que esté ejecutando el servicio en local.
- **Multi-plataforma:** Es prácticamente multiplataforma, dado que funciona siempre que el sistema pueda ejecutar una JVM (Java Virtual Machine), cosa muy extendida en la actualidad para casi todos los sistemas operativos.
- **Seguridad:** Ofrece plug-ins para cubrir elementos de seguridad pero son dependientes de otros programas instalados. Los propios desarrolladores y distribuidores de Apache Solr recomiendan su uso con firewall.

4.2.4. ElasticSearch

Métricas	Pesos	Puntuación	Resultado
Productividad	2	75 %	1,5
Requisitos de ejecución	1	50 %	0,5
Consumo de energía	1	50 %	0,5
Multi-plataforma	2	100 %	2
Seguridad	3	25 %	0,75
TOTAL	9		5,25

Tabla 4.4: Ejecución - ElasticSearch

- **Productividad:** El motor es capaz de indexar gran cantidad de documentación, dependiente de las máquinas del usuario que alberguen el servicio. Sin embargo, ofrece poca personalización en el algoritmo de ordenación de resultados.
- **Requisitos de ejecución:** Requiere de un sistema que ejecute el proceso de forma local. Precisa de tener instalada la versión Java 8 o superior en la máquina local en la que va a operar el servicio.
- **Consumo de energía:** El proceso se ejecuta de forma local en la máquina del usuario.
- **Multi-plataforma:** Disponible para cualquier sistema UNIX o Windows que permita la ejecución de una JVM.
- **Seguridad:** El propio motor está planteado para realizar su funcionalidad de forma exclusiva y deja las tareas de seguridad en manos del desarrollador, que tiene la posibilidad de implementarlas como servicio externo.

4.2.5. Google Search Appliance

Métricas	Pesos	Puntuación	Resultado
Productividad	2	100 %	2
Requisitos de ejecución	1	0 %	0
Consumo de energía	1	0 %	0
Multi-plataforma	2	100 %	2
Seguridad	3	100 %	3
TOTAL	9		7

Tabla 4.5: Ejecución - GSA

- **Productividad:** Está planteado para ser capaz de indexar miles de millones de documentos, con restricciones hardware. La personalización de resultados es posible gracias a una configuración avanzada, que, aunque sin ser del todo libre, es muy completa.
- **Requisitos de ejecución:** Requiere hardware adicional para su funcionamiento, a través de un conjunto de módulos externos que provee el propio fabricante.
- **Consumo de energía:** Dado que precisa de hardware adicional para la ejecución del proceso, el consumo de energía es elevado.
- **Multi-plataforma:** El servicio puede integrarse con cualquier sistema del mercado, con excepciones mínimas. Su ejecución en un módulo aislado le otorga una independencia clave en esa integración.
- **Seguridad:** Ofrece compatibilidad con varios mecanismos de autenticación y de inicio de sesión único, entre ellos, LDAP, autenticación básica y NTLM, autenticación PKI con certificados x.509, Kerberos y autenticación integrada de Windows. Además, se integra con los sistemas de seguridad que posea ya el propio usuario. Posibilidad de inclusión listas de acceso y otros mecanismos de seguridad adicionales.

4.2.6. Indica ES

Métricas	Pesos	Puntuación	Resultado
Productividad	2	50 %	1
Requisitos de ejecución	1	75 %	0,75
Consumo de energía	1	100 %	1
Multi-plataforma	2	100 %	2
Seguridad	3	75 %	2,25
TOTAL	9		7

Tabla 4.6: Ejecución - Indica

- **Productividad:** Alta capacidad de indexación de documentos, aunque dependiente del precio pagado por el servicio. Sin embargo, el fabricante no menciona ningún

mecanismo de personalización del algoritmo de búsqueda, aunque es posible que se ofrezca algún tipo de configuración.

- **Requisitos de ejecución:** El servicio requiere cuentas y subidas de datos a la plataforma de Indica ES, tal y como otros servicios ofrecen.
- **Consumo de energía:** Dado que el servicio provee de una solución online no precisa de máquina de ejecución por parte del usuario.
- **Multi-plataforma:** El servicio se accede a través de la web, lo que le otorga una funcionalidad multi-plataforma en todos los sistemas que puedan conectarse a la web a través de un navegador.
- **Seguridad:** El servicio funciona sobre protocolo HTTPS, además de proporcionar otros mecanismos de seguridad en red. Sin embargo, no se hace mención a que el usuario pueda hacer funcionar servicios ajenos de seguridad en colaboración con la plataforma.

4.2.7. Microsoft Azure Search

Métricas	Pesos	Puntuación	Resultado
Productividad	2	100 %	2
Requisitos de ejecución	1	75 %	0,75
Consumo de energía	1	100 %	1
Multi-plataforma	2	100 %	2
Seguridad	3	75 %	2,25
TOTAL	9		8

Tabla 4.7: Ejecución - MicrosoftAS

- **Productividad:** Microsoft AS ofrece un alto grado de indexación, que varía con la licencia pagada, junto con una alta personalización de resultados accesible al usuario, a través de variables de configuración y búsquedas avanzadas.
- **Requisitos de ejecución:** Requiere datos del usuario para ejecutar el proceso en la nube, con un registro en la plataforma y los datos de la tarjeta de crédito. Además, precisa de subida de documentos a la plataforma.
- **Consumo de energía:** El servicio se oferta como SaaS(Software as a Service) y no requiere de un mantenimiento local.
- **Multi-plataforma:** El motor provee de acceso web, como otros expuestos anteriormente, lo que le otorga flexibilidad en el acceso desde otras plataformas.
- **Seguridad:** Aporta exclusivamente mecanismos propios de seguridad, como trabajo sobre protocolo HTTP, protección de datos en red y seguridad en los servidores de almacenamiento de los datos.

4.2.8. SearchBlox

Métricas	Pesos	Puntuación	Resultado
Productividad	2	75 %	1,5
Requisitos de ejecución	1	100 %	1
Consumo de energía	1	100 %	1
Multi-plataforma	2	100 %	2
Seguridad	3	75 %	2,25
TOTAL	9		7,75

Tabla 4.8: Ejecución - SearchBlox

- **Productividad:** Ofrece una alta capacidad de indexación dependiente de la licencia adquirida con una configuración personalizable predeterminada y poco flexible en la búsqueda de resultados.
- **Requisitos de ejecución:** Aporta una solución como SaaS accesible a través de la web, por lo que no requiere de sistemas de ejecución locales.
- **Consumo de energía:** Requiere crear una cuenta en la plataforma de SearchBlox. Además, es necesario subir el set de datos a la plataforma para poder emplear el motor de búsqueda.
- **Multi-plataforma:** Dado que el servicio se aprovisiona a través de una web, es accesible bajo cualquier plataforma que provea acceso a Internet a través de un navegador.
- **Seguridad:** Cobertura de la seguridad a nivel de documento a través de Active Directory. También ofrece opciones bajo pago de seguridad con Okta. Las conexiones con la plataforma se hacen a través del protocolo HTTPS.

4.2.9. Sphinx Search Server

Métricas	Pesos	Puntuación	Resultado
Productividad	2	100 %	2
Requisitos de ejecución	1	50 %	0,5
Consumo de energía	1	50 %	0,5
Multi-plataforma	2	75 %	1,5
Seguridad	3	25 %	0,75
TOTAL	9		5,25

Tabla 4.9: Ejecución - Sphinx

- **Productividad:** El sistema otorga una alta capacidad de indexación y personalización en el sistema de ordenación para los resultados en las búsquedas, además de que al ser Open Source todo es modificable.

- **Requisitos de ejecución:** Al igual que otros motores de búsqueda similares requiere una máquina propia del usuario en la que ejecutarse.
- **Consumo de energía:** Ejecución en local como servicio del sistema, por lo que precisa de una máquina propia.
- **Multi-plataforma:** El producto ofrece cobertura de instalación en sistemas Windows y parte de sistemas UNIX, como FreeBSD y MAC.
- **Seguridad:** El servicio acepta elementos de seguridad del cliente pero no aporta elementos de seguridad propios, lo que obliga a los usuarios a implementarlos o agregarlos a través de programas externos.

4.2.10. Crate

Métricas	Pesos	Puntuación	Resultado
Productividad	2	75 %	1,5
Requisitos de ejecución	1	50 %	0,5
Consumo de energía	1	50 %	0,5
Multi-plataforma	2	75 %	1,5
Seguridad	3	75 %	2,25
TOTAL	9		6,25

Tabla 4.10: Ejecución - Crate

- **Productividad:** Alto grado de indexación de elementos en el sistema de base de datos que conforma el corazón de Crate.io. No existen elementos de personalización más allá de los que provee el propio usuario con sus consultas, dado que Crate no es un motor de búsqueda puro.
- **Requisitos de ejecución:** Al igual que otros sistemas similares expuestos requiere una máquina propia del usuario en la que ejecutar el proceso.
- **Consumo de energía:** Contempla el consumo derivado de mantener el proceso en ejecución localmente
- **Multi-plataforma:** Ofrece instalación del servicio en sistemas Windows, Linux y Mac OS
- **Seguridad:** Aporta elementos de seguridad propios pero no muy abundantes, como listas de acceso. Puede llegar a integrar algunos mecanismos de seguridad del cliente.

4.2.11. Comentario sobre la comparativa a nivel de ejecución

Como puede recogerse de los cuadros, Microsoft Azure Search sale victorioso en el apartado de ejecución. Cubre con mucha soltura todas las métricas, aunque hay otros que

se mantienen muy próximos a él. En general, en este área han salido beneficiados aquellos productos ofertados como SaaS, sin máquinas locales que mantener y con acceso web a la plataforma.

4.3. Comparativa a nivel de operación

4.3.1. Algolia

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	50 %	1
Mantenimiento	2	0 %	0
Escalabilidad	3	75 %	2,25
Arquitectura	2	100 %	2
TOTAL	9		5,25

Tabla 4.11: Operación - Algolia

- **Monitorización:** Permite visualizar que búsquedas son las más populares, además de cuántas se hacen por país y cuáles no han devuelto resultados.
- **Mantenimiento:** El mantenimiento depende de la plataforma, y es transparente para el usuario.
- **Escalabilidad:** La escalabilidad es alta, pero depende del precio pagado. Sin embargo, se garantiza la escalabilidad en la mensualidad pagada, en caso de que el usuario supere su cuota asignada.
- **Arquitectura:** Transparente para el usuario al ser un servicio en la nube.

4.3.2. Amazon Cloud Search

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	50 %	1
Mantenimiento	2	0 %	0
Escalabilidad	3	75 %	2,25
Arquitectura	2	100 %	2
TOTAL	9		5,25

Tabla 4.12: Operación - ACS

- **Monitorización:** La monitorización se reduce a visualizar el estado y la calidad del servicio. No provee de métodos de recogida de estadísticas a la información accedida.

- **Mantenimiento:** Es dependiente de la plataforma e inaccesible para el usuario.
- **Escalabilidad:** Alta, pero dependiente del precio pagado. Sin embargo, se garantiza la escalabilidad en la mensualidad pagada, en caso de que el usuario supere la cuota de servicio esperado.
- **Arquitectura:** Se trata de una solución transparente al usuario al ser un servicio cloud

4.3.3. Apache Solr

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	100 %	2
Mantenimiento	2	50 %	1
Escalabilidad	3	100 %	3
Arquitectura	2	100 %	2
TOTAL	9		8

Tabla 4.13: Operación - Solr

- **Monitorización:** A través de su herramienta SPM permite crear métricas personalizables, evaluar el estado del sistema filtrando por servidor o estudiar las consultas realizadas. El estado de la JVM sobre la que funciona Solr puede evaluarse con las herramientas de Java.
- **Mantenimiento:** Permite mantenimiento interno por parte del usuario, aunque no existe posibilidad de mantenimiento externo.
- **Escalabilidad:** Alta escalabilidad con total transparencia para el usuario con reordenamiento del índice transparente al usuario.
- **Arquitectura:** Solución diseñada con finalidad distribuida pero que permite funcionamiento centralizado si se desea.

4.3.4. ElasticSearch

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	25 %	0,5
Mantenimiento	2	100 %	2
Escalabilidad	3	100 %	3
Arquitectura	2	100 %	2
TOTAL	9		7,5

Tabla 4.14: Operación - ElasticSearch

- **Monitorización:** Es posible crear procesos de usuario para monitorizar las actividades, aunque no provee de los mismos, dado que la filosofía del producto es la de otorgar funcionalidad mínima.
- **Mantenimiento:** Mantenimiento interno por parte del usuario con posibilidad de mantenimiento externo bajo pago.
- **Escalabilidad:** Solución altamente escalable y personalizable en algunos aspectos para ofrecer transparencia al usuario en las operaciones internas del servicio.
- **Arquitectura:** Funcionamiento distribuido con posibilidad de trabajo centralizado en un único nodo.

4.3.5. Google Search Appliance

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	25 %	0,5
Mantenimiento	2	25 %	0,5
Escalabilidad	3	25 %	0,75
Arquitectura	2	25 %	0,5
TOTAL	9		2,25

Tabla 4.15: Operación - GSA

- **Monitorización:** Existe un panel de control para la administración de los sistemas. Sin embargo, las soluciones de monitorización más completas pasan por aquellas hechas por los propios usuarios a través de la API del sistema.
- **Mantenimiento:** Dependiente de mantenimiento externo por parte de Google, que es el que conoce el funcionamiento de su hardware.
- **Escalabilidad:** La escalabilidad es elevada, pero dependiente de módulos hardware adicionales en casos de crecimiento extremo o aumento elevado de la demanda.
- **Arquitectura:** Solución centralizada en el hardware adquirido con control de fallos y mecanismos de recuperación de errores.

4.3.6. Indica ES

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	0 %	0
Mantenimiento	2	0 %	0
Escalabilidad	3	75 %	2,25
Arquitectura	2	100 %	2
TOTAL	9		4,25

Tabla 4.16: Operación - Indica

- **Monitorización:** El distribuidor no hace mención a ningún sistema de monitorización en ningún lugar, ni de que pueda implementarse de alguna manera o haya alguna facilidad para hacerlo.
- **Mantenimiento:** Dispone de mantenimiento exclusivamente externo.
- **Escalabilidad:** Dependiente del pago por parte del usuario, dado que no se le permite superar ciertos umbrales de uso no contratados.
- **Arquitectura:** Solución transparente para el usuario al proveerse como SaaS.

4.3.7. Microsoft Azure Search

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	75 %	1,5
Mantenimiento	2	0 %	0
Escalabilidad	3	75 %	2,25
Arquitectura	2	100 %	2
TOTAL	9		5,75

Tabla 4.17: Operación - Microsoft AS

- **Monitorización:** Monitorización del servicio y estadísticas breves de acceso a datos. La monitorización del servicio es bastante completa y ofrece estadísticas de acceso y métricas de rendimiento. Las estadísticas de acceso a datos son limitadas y no proveen de excesiva información relevante.
- **Mantenimiento:** El servicio se mantiene en exclusiva de forma externa por parte de la empresa distribuidora.
- **Escalabilidad:** En caso de superar ciertos umbrales de uso se requiere pagar por el servicio, aunque la solución aportada es altamente escalable.
- **Arquitectura:** Arquitectura transparente al usuario con solución en la nube accesible como SaaS.

4.3.8. SearchBlox

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	50 %	1
Mantenimiento	2	0 %	0
Escalabilidad	3	75 %	2,25
Arquitectura	2	100 %	2
TOTAL	9		5,25

Tabla 4.18: Operación - SearchBlox

- **Monitorización:** La monitorización es limitada. Permite visualizar que búsquedas son las más populares, además de cuántas se hacen por país y cuáles no han devuelto resultados.
- **Mantenimiento:** Totalmente externalizado y bajo las funciones de la plataforma.
- **Escalabilidad:** Solución altamente escalable pero bajo los umbrales que dicta la licencia pagada.
- **Arquitectura:** Sistema con arquitectura transparente al usuario al aportar una solución cloud.

4.3.9. Sphinx Search Server

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	25 %	0,5
Mantenimiento	2	100 %	2
Escalabilidad	3	75 %	2,25
Arquitectura	2	100 %	2
TOTAL	9		6,75

Tabla 4.19: Operación - Sphinx

- **Monitorización:** No existen procesos de monitorización pero pueden crearse algunos personalizados por parte de los usuarios. Existen tutoriales en la red de como hacerlos.
- **Mantenimiento:** Mantenimiento del servicio por parte del usuario con mantenimiento externo bajo pago disponible.
- **Escalabilidad:** Solución altamente escalable dependiente de prestaciones bajo pago. Si se precisa de unas prestaciones mayores el servicio requiere de una solución empresarial ofertada por la plataforma, aunque no es obligatorio adquirirla para operar con normalidad.
- **Arquitectura:** Sphinx aporta una solución distribuida con opciones de operar de forma centralizada.

4.3.10. Crate

Métricas	Pesos	Puntuación	Resultado
Monitorización	2	50 %	1
Mantenimiento	2	100 %	2
Escalabilidad	3	100 %	3
Arquitectura	2	75 %	1,5
TOTAL	9		7,5

Tabla 4.20: Operación - Crate

- **Monitorización:** Ofrece monitorización del estado del sistema, carga, capacidad y rendimiento, aunque no estadísticas de la información accedida.
- **Mantenimiento:** El mantenimiento del servicio depende del usuario, aunque la plataforma ofrece mantenimiento bajo pago.
- **Escalabilidad:** Sharding automático sin reordenación de índices. Si se añade cualquier nodo adicional a la red el proceso de inserción es transparente al usuario.
- **Arquitectura:** Solución totalmente distribuida con opciones de configuración elevadas y control de fallos.

4.3.11. Comentario sobre la comparativa a nivel de operación

En el apartado de operación el motor vencedor ha sido Solr, gracias a su planteamiento distribuido y sus herramientas de monitorización. En general los productos que han salido beneficiados en este apartado de evaluación han sido aquellos en los que la escalabilidad no es dependiente del precio pagado, y permiten su uso tanto a nivel distribuido como centralizado dependiendo de los requisitos del usuario.

4.4. Comparativa a nivel de desarrollo

4.4.1. Algolia

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	50 %	1
Independencia del servidor	1	0 %	0
Integración con otros sistemas	3	75 %	2,25
TOTAL	6		3,25

Tabla 4.21: Desarrollo - Algolia

- **Curva de aprendizaje:** No posee muchas herramientas que dominar, aunque los recursos de aprendizaje no abundan.
- **Independencia del servidor:** Depende totalmente de los servidores de Algolia, y no es transportable a otros.
- **Integración con otros sistemas:** Existen numerosas APIs para integrar el servicio de Algolia con diversos desarrollos. Sin embargo, la cobertura no es global y las APIs proporcionadas varían dependiendo de la implantación.

4.4.2. Amazon Cloud Search

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	75 %	1,5
Independencia del servidor	1	0 %	0
Integración con otros sistemas	3	75 %	2,25
TOTAL	6		3,75

Tabla 4.22: Desarrollo - ACS

- **Curva de aprendizaje:** Muy poco pronunciada, dado que no existe demasiada complejidad que dominar y los recursos de aprendizaje son muy extensos y detallados.
- **Independencia del servidor:** Depende totalmente del servidor de Amazon, y no puede portarse a otros sistemas.
- **Integración con otros sistemas:** Provee de una API para la interacción con aplicaciones diseñadas por el usuario a nivel local y web bastante completas y sencillas de usar.

4.4.3. Apache Solr

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	50 %	1
Independencia del servidor	1	100 %	1
Integración con otros sistemas	3	100 %	3
TOTAL	6		5

Tabla 4.23: Desarrollo - Apache Solr

- **Curva de aprendizaje:** Existen muchas herramientas y configuraciones que dominar, aunque hay mucha información a disposición de los usuarios.

- **Independencia del servidor:** Funciona de forma totalmente independiente del servidor, dado que se ejecuta sobre la JVM y sólo precisa que el otro servidor ejecute asimismo JVM.
- **Integración con otros sistemas:** Provee una API que se integra prácticamente con cualquier sistema que sea capaz de interpretar formatos XML, JSON y otros.

4.4.4. ElasticSearch

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	50 %	1
Independencia del servidor	1	100 %	1
Integración con otros sistemas	3	100 %	3
TOTAL	6		5

Tabla 4.24: Desarrollo - ElasticSearch

- **Curva de aprendizaje:** Recursos extensos y muy precisos aunque existen muchas herramientas y configuración que dominar.
- **Independencia del servidor:** Totalmente independiente siempre y cuando los sistemas objetivos sean capaces de operar con JVM.
- **Integración con otros sistemas:** Numerosas APIs sencillas de utilizar. Disponibilidad de trabajo en varios lenguajes de programación y soporte y recursos para emplearlos.

4.4.5. Google Search Appliance

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	25 %	0,5
Independencia del servidor	1	0 %	0
Integración con otros sistemas	3	25 %	0,75
TOTAL	6		1,25

Tabla 4.25: Desarrollo - GSA

- **Curva de aprendizaje:** Precisa de una instalación de hardware, además de necesitar de una amplia configuración inicial, y compresión de recursos hardware y software.
- **Independencia del servidor:** Precisa de hardware adicional para su funcionamiento, y no puede operar fuera del hardware proporcionado por el distribuidor.

- **Integración con otros sistemas:** Dispone de una API para integrar sus servicios con otros sistemas, a través de objetos JSON y solicitudes de otros procesos con las credenciales adecuadas y web.

4.4.6. Indica ES

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	25 %	0,5
Independencia del servidor	1	0 %	0
Integración con otros sistemas	3	0 %	0
TOTAL	6		0,5

Tabla 4.26: Desarrollo - Indica ES

- **Curva de aprendizaje:** No existen muchos recursos que faciliten el aprender a usar la herramienta. Además, son difíciles de comprender y localizar.
- **Independencia del servidor:** Dependiente de los servidores de Indica ES, por lo que le es imposible operar fuera de los mismos.
- **Integración con otros sistemas:** No ofrece integración con otros sistemas o servicios o al menos los distribuidores no hacen mención de ello. Puede que sea posible realizar solicitudes desde otros procesos al servicio web

4.4.7. Microsoft Azure Search

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	50 %	1
Independencia del servidor	1	0 %	0
Integración con otros sistemas	3	100 %	3
TOTAL	6		4

Tabla 4.27: Desarrollo - Microsoft AS

- **Curva de aprendizaje:** Gran cantidad de herramientas y configuraciones que dominar. Sin embargo, hay asimismo una buena cantidad de recursos explicativos y tutoriales de uso disponibles.
- **Independencia del servidor:** El servicio se ejecuta bajo los servidores de Microsoft Azure Search y no opera fuera de ese entorno.
- **Integración con otros sistemas:** Dispone de una API REST integrable con gran cantidad de servicios o sistemas, además de ser sencilla de manejar.

4.4.8. SearchBlox

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	75 %	1,5
Independencia del servidor	1	0 %	0
Integración con otros sistemas	3	100 %	3
TOTAL	6		4,5

Tabla 4.28: Desarrollo - SearchBlox

- **Curva de aprendizaje:** Recursos bien presentados y fáciles de localizar. Posee ejemplos de configuraciones disponibles y no posee gran cantidad de herramientas que emplear.
- **Independencia del servidor:** El servicio es dependiente de los servidores de SearchBlox al proveerse como un SaaS, por lo que no es operable desde otros servidores.
- **Integración con otros sistemas:** Dispone de una API REST integrable con otros sistemas y servicios.

4.4.9. Sphinx Search Server

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	25 %	0,5
Independencia del servidor	1	50 %	0,5
Integración con otros sistemas	3	75 %	2,25
TOTAL	6		3,25

Tabla 4.29: Desarrollo - Sphinx

- **Curva de aprendizaje:** Bastante elevada. Posee recursos poco extensos, aunque bien presentados. Aporta asimismo muchas herramientas y funcionalidad compleja a dominar. además de opciones de personalización y disponibilidad de creación de procesos adjuntos a la funcionalidad existente.
- **Independencia del servidor:** Precisa de adaptar el servicio al servidor de destino, aunque es relativamente transportable.
- **Integración con otros sistemas:** Existe una API en PHP preparada para crear servicios web, y también permite al usuario crear APIs propias.

4.4.10. Crate

Métricas	Pesos	Puntuación	Resultado
Curva de aprendizaje	2	100 %	2
Independencia del servidor	1	50 %	0,5
Integración con otros sistemas	3	100 %	3
TOTAL	6		5,5

Tabla 4.30: Desarrollo - Crate

- **Curva de aprendizaje:** Gran cantidad de recursos de aprendizaje. Herramienta con poca complejidad que dominar, muy similar a MySQL y otras bases de datos relacionales.

- **Independencia del servidor:** Es dependiente del sistema de instalación, aunque puede transportarse a otros sistemas con una instalación nueva en los mismos y una migración de configuración y datos.

- **Integración con otros sistemas:** Provee de una API REST para la integración del servicio con aplicaciones y sistemas web muy sencilla de usar.

4.4.11. Comentario sobre la comparativa a nivel de desarrollo

En el área de desarrollo el producto vencedor ha sido Crate. Gracias a la gran cantidad de recursos bien presentados que posee el motor, junto con la sencillez de su herramienta al emplearse con lenguaje SQL, familiar a gran parte de usuarios, el motor mantiene una curva de aprendizaje muy suave y sencilla. En este apartado, los productos con mejor nota han sido aquellos que funcionan de forma independiente al servidor, como los que operan sobre la JVM, y aquellos que poseen gran cantidad de recursos y funcionalidad no muy compleja.

4.5. Comparativa a nivel de soporte

4.5.1. Algolia

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	100 %	2
Soporte online y comunidad de usuarios	2	25 %	0,5
Documentación	3	50 %	1,5
Licencia del producto	3	25 %	0,75
Continuidad del producto	3	100 %	3
Multi-idioma	1	100 %	1
TOTAL	14		8,75

Tabla 4.31: Soporte - Algolia

- **Soporte del proveedor:** Completo a través del correo electrónico. Mucho feedback con presencia en las redes sociales.
- **Soporte online y comunidad de usuarios:** Existen algunos foros con información, pero el que posee más contenido lo ofrecen ellos mismos en su página web.
- **Documentación:** Muy completa pero poco extensa. Muy bien presentada.
- **Licencia del producto:** La licencia más cara es de 899 dólares mensuales. Posee una demo gratuita, pero sólo disponible durante 14 días.
- **Continuidad del producto:** La compañía mantiene en la actualidad versiones del producto y diariamente muestra el estado de las APIs.
- **Multi-idioma:** La solución aportada por Algolia admite gran cantidad de idiomas, aunque no especifican cuantos.

4.5.2. Amazon Cloud Search

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	100 %	2
Soporte online y comunidad de usuarios	2	100 %	2
Documentación	3	75 %	2,25
Licencia del producto	3	50 %	1,5
Continuidad del producto	3	100 %	3
Multi-idioma	1	100 %	1
TOTAL	14		11,75

Tabla 4.32: Soporte - ACS

- **Soporte del proveedor:** Existen numerosos canales de soporte personalizado bajo pago.
- **Soporte online y comunidad de usuarios:** Gran cantidad de foros con información sobre ACS. Presencia de comunidades de usuarios con aportes frecuentes.
- **Documentación:** Existen libros y documentación tanto interna como externa en webs y foros.
- **Licencia del producto:** Posee una versión gratuita de uso en caso de no superar ciertos umbrales de uso. El resto se cobra por recursos consumidos.
- **Continuidad del producto:** El producto se encuentra actualmente en proceso de evolución y mantenimiento.
- **Multi-idioma:** Disponible en más de 30 idiomas para su uso.

4.5.3. Apache Solr

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	75 %	1,5
Soporte online y comunidad de usuarios	2	100 %	2
Documentación	3	100 %	3
Licencia del producto	3	100 %	3
Continuidad del producto	3	100 %	3
Multi-idioma	1	50 %	0,5
TOTAL	14		13

Tabla 4.33: Soporte - Solr

- **Soporte del proveedor:** Provee de soporte a través de un sistema de listas de correo electrónico y otros medios.
- **Soporte online y comunidad de usuarios:** Existen numerosas entradas en foros propios de Solr, además de otros en plataformas diferentes, como el foro StackOverflow, libros online y otros.
- **Documentación:** Posee una amplia gama de documentos y tutoriales hechos por la comunidad, así como algunos propios de la plataforma.
- **Licencia del producto:** Licencia de código abierto gratuita.
- **Continuidad del producto:** En la actualidad se llegan por la versión 6.6.0. Se esperan en el futuro desarrollos de nuevas versiones.
- **Multi-idioma:** Provee usos con otros idiomas aparte del inglés. La comunidad de usuarios que tiene está implementando otras traducciones.

4.5.4. ElasticSearch

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	75 %	1,5
Soporte online y comunidad de usuarios	2	100 %	2
Documentación	3	100 %	3
Licencia del producto	3	75 %	2,25
Continuidad del producto	3	100 %	3
Multi-idioma	1	50 %	0,5
TOTAL	14		12,25

Tabla 4.34: Soporte - ElasticSearch

- **Soporte del proveedor:** Disponibilidad de métodos de contacto diferentes como correo o foros. Atención personalizada por parte de la plataforma bajo precio.
- **Soporte online y comunidad de usuarios:** Gran presencia en foros y una comunidad extensa. Presencia en GitHub y otras plataformas.
- **Documentación:** Abundante documentación interna y externa.
- **Licencia del producto:** Open Source con soluciones comerciales en cuanto al soporte.
- **Continuidad del producto:** Se llega por la versión 5.4.1. El producto se espera continuar en el futuro.
- **Multi-idioma:** Ofrece soporte en algunos idiomas diferentes al inglés, aunque de manera reducida.

4.5.5. Google Search Appliance

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	100 %	2
Soporte online y comunidad de usuarios	2	50 %	1
Documentación	3	75 %	2,25
Licencia del producto	3	0 %	0
Continuidad del producto	3	0 %	0
Multi-idioma	1	100 %	1
TOTAL	14		6,25

Tabla 4.35: Soporte - GSA

- **Soporte del proveedor:** Posee una web de soporte, así como contactos y métodos de acceso remotos para dar cobertura a los usuarios.

- **Soporte online y comunidad de usuarios:** Existen algunos foros con menciones a cosas concretas, así como algunos libros concretos con algún estudio.
- **Documentación:** Existen libros y documentación externa aparte de la propia interna del proveedor.
- **Licencia del producto:** El precio de la licencia varía según el uso de recursos que se precise. El presupuesto es personalizado a cada solución. Carece de demo para probar.
- **Continuidad del producto:** Aunque de manera confidencial, se ha avisado a los clientes que se piensa cesar la producción de GSA hacia 2018 para apostar por una solución cloud.
- **Multi-idioma:** Disponible en más de 20 idiomas.

4.5.6. Indica ES

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	100 %	2
Soporte online y comunidad de usuarios	2	25 %	0,5
Documentación	3	25 %	0,75
Licencia del producto	3	25 %	0,75
Continuidad del producto	3	100 %	3
Multi-idioma	1	25 %	0,25
TOTAL	14		7,25

Tabla 4.36: Soporte - Indica ES

- **Soporte del proveedor:** Numerosos canales de comunicación: correo, teléfono...
- **Soporte online y comunidad de usuarios:** Existen menciones puntuales al producto en alguna página externa.
- **Documentación:** Solamente documentación interna y breve, además de difícil de interpretar.
- **Licencia del producto:** El servicio es bajo pago, pero tiene demo gratuita si se solicita por correo. Sin embargo, no se especifica qué aporta la demo ni cuánto tiempo está disponible.
- **Continuidad del producto:** Se esperan en el futuro desarrollos de nuevas versiones, reflejado por ejemplo en los post de noticias que suben los desarrolladores.
- **Multi-idioma:** No se hace referencia alguna a una versión del producto en un idioma distinto del inglés.

4.5.7. Microsoft Azure Search

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	25 %	0,5
Soporte online y comunidad de usuarios	2	50 %	1
Documentación	3	75 %	2,25
Licencia del producto	3	75 %	2,25
Continuidad del producto	3	100 %	3
Multi-idioma	1	100 %	1
TOTAL	14		10

Tabla 4.37: Soporte - Microsoft AS

- **Soporte del proveedor:** Soporte que varía en calidad dependiendo del precio pagado por parte del usuario.
- **Soporte online y comunidad de usuarios:** Existen foros sobre temas puntuales, aunque la mayoría de entradas las aporta Microsoft. Cabe resaltar que los foros aportan soluciones a problemas y documentación muy completa.
- **Documentación:** Documentación interna y externa disponible, aunque la externa no es excesivamente abundante, dado que la mayoría la acapara la propia plataforma de Microsoft.
- **Licencia del producto:** Privada. Demo gratuita de 30 días y continuidad gratuita revisable bajo ciertos parámetros de uso.
- **Continuidad del producto:** Se espera que el producto se continúe actualizando en el futuro por parte de los desarrolladores.
- **Multi-idioma:** Ofrece más de 50 idiomas entre los que se incluyen algunos muy poco comunes.

4.5.8. SearchBlox

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	100 %	2
Soporte online y comunidad de usuarios	2	50 %	1
Documentación	3	75 %	2,25
Licencia del producto	3	25 %	0,75
Continuidad del producto	3	100 %	3
Multi-idioma	1	100 %	1
TOTAL	14		10

Tabla 4.38: Soporte - SearchBlox

- **Soporte del proveedor:** Disponen de varios canales de comunicación y foros de debate en redes sociales.

- **Soporte online y comunidad de usuarios:** Existen menciones y tutoriales de la API externos, aunque no en exceso.
- **Documentación:** Existe documentación interna y externa a disposición del usuario, muy bien presentada.
- **Licencia del producto:** Privada. Ofrece una demo gratuita de 30 días.
- **Continuidad del producto:** Se espera que el producto siga evolucionando en el futuro con nuevas versiones del mismo.
- **Multi-idioma:** El producto se ofrece con soporte en 40 idiomas diferentes.

4.5.9. Sphinx Search Server

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	25 %	0,5
Soporte online y comunidad de usuarios	2	75 %	1,5
Documentación	3	50 %	1,5
Licencia del producto	3	75 %	2,25
Continuidad del producto	3	100 %	3
Multi-idioma	1	50 %	0,5
TOTAL	14		9,25

Tabla 4.39: Soporte - Sphinx Search Server

- **Soporte del proveedor:** Dependiente del precio pagado. Ofrece soporte a empresas bajo ciertas tarifas.
- **Soporte online y comunidad de usuarios:** Existen foros internos. Algunas personas hacen aportes en foros externos, aunque en general son para resolver cuestiones ajenas al funcionamiento del proceso en sí, más relacionadas con procesos de monitorización u otras soluciones.
- **Documentación:** Documentación interna de presentación pobre y externa limitada. La documentación sólo se ofrece en ruso e inglés.
- **Licencia del producto:** Open source con soluciones comerciales para empresas o usuarios con necesidades grandes.
- **Continuidad del producto:** Se espera que el producto se mantenga actualizado en el futuro con nuevas versiones del mismo.
- **Multi-idioma:** Disponible en inglés y en algún idioma extra, como el ruso.

4.5.10. Crate

Métricas	Pesos	Puntuación	Resultado
Soporte del proveedor	2	100 %	2
Soporte online y comunidad de usuarios	2	75 %	1,5
Documentación	3	50 %	1,5
Licencia del producto	3	25 %	0,75
Continuidad del producto	3	100 %	3
Multi-idioma	1	25 %	0,25
TOTAL	14		9

Tabla 4.40: Soporte - Crate

- **Soporte del proveedor:** Posee Slack Channel, Github y contacto por correo. Para los clientes grandes además ofrece contacto por teléfono.
- **Soporte online y comunidad de usuarios:** Existen algunos foros con información en StackOverflow, GitHub y otras plataformas. Sin embargo, muchas entradas son patrocinadas por ellos mismos.
- **Documentación:** Muy completa y extensa. Muy bien presentada. La externa es escasa.
- **Licencia del producto:** Privada. La plataforma cobra por el servicio y el soporte que ofrece al usuario. Ofrece una demo de prueba aunque no especifica el tiempo máximo de duración de la misma.
- **Continuidad del producto:** La compañía mantiene en la actualidad versiones del producto y se esperan nuevas versiones en el futuro.
- **Multi-idioma:** En ningún lugar se hace referencia a la disponibilidad del producto en idiomas diferentes del inglés.

4.5.11. Comentario sobre la comparativa a nivel de soporte

En la sección de soporte, el motor ganador ha sido Solr. Su nota se ha beneficiado especialmente de su licencia de código abierto gratuita, su gran comunidad de usuarios y la gran cantidad de documentación interna y externa que ofrecen tanto la plataforma como investigadores y usuarios.

En general, han salido beneficiados en este área aquellos motores con licencia de código abierto, documentación abundante y soporte por parte de la comunidad elevado. Sorprende la nota de Google Search Appliance, que sufre debido a su licencia de pago y su discontinuidad futura.

4.6. Comparativa final global

En esta sección se exponen las notas finales obtenidas por todos los motores en cada uno de los apartados.

Para ello, en primer lugar se muestra una tabla resumen de los valores en cada una de las áreas de los motores de búsqueda, para ofrecer visualmente en un único lugar todos los valores finales reflejados anteriormente. Para resaltar los valores vencedores, se han marcado en negrita junto con las puntuaciones máximas.

Motores	Ejecución	Operación	Desarrollo	Soporte
Algolia	7,75	5,25	3,25	8,75
Amazon Cloud Search	7,75	5,25	3,75	11,75
Apache Solr	6,5	8	5	13
ElasticSearch	5,25	7,5	5	12,25
Google Search Appliance	7	2,25	1,25	6,25
Indica ES	7	4,25	0,5	7,25
Microsoft Azure Search	8	5,75	4	10
SearchBlox	7,75	5,25	4,5	10
Sphinx	5,25	6,75	3,25	9,25
Crate	6,25	7,5	5,5	9
NOTA MÁXIMA FINAL	9	9	6	14

Tabla 4.41: Tabla resumen áreas de evaluación

Seguidamente se aporta la comparativa final, con las notas sobre el peso total final y normalizadas a 10. Para obtener esa nota final sobre 10, se suman las puntuaciones obtenidas por el motor en cada una de las cuatro áreas y se divide por el sumatorio total de los pesos, multiplicando por 10 el resultado. Esto es, para cada motor, se toma la puntuación que ha sacado en cada una de las áreas, se suman todas ellas, y se divide entre el sumatorio total de los pesos, multiplicando el número por 10.

Para ilustrar los datos se aporta una pequeña tabla comparativa, así como un diagrama final en la que se expone en un gráfico de barras las puntuaciones finales de los motores, para mostrar representativamente al producto ganador.

Motores	Nota máxima (suma pesos en cada área)	Nota final sobre 10
Algolia	25	6,58
Amazon Cloud Search	28,5	7,5
Apache Solr	32,5	8,55
ElasticSearch	30	7,89
Google Search Appliance	16,75	4,41
Indica ES	19	5
Microsoft Azure Search	27,75	7,3
SearchBlox	27,5	7,24
Sphinx	24,5	6,45
Crate	28,25	7,43
NOTA MÁXIMA FINAL	38	10

Tabla 4.42: Comparativa final global

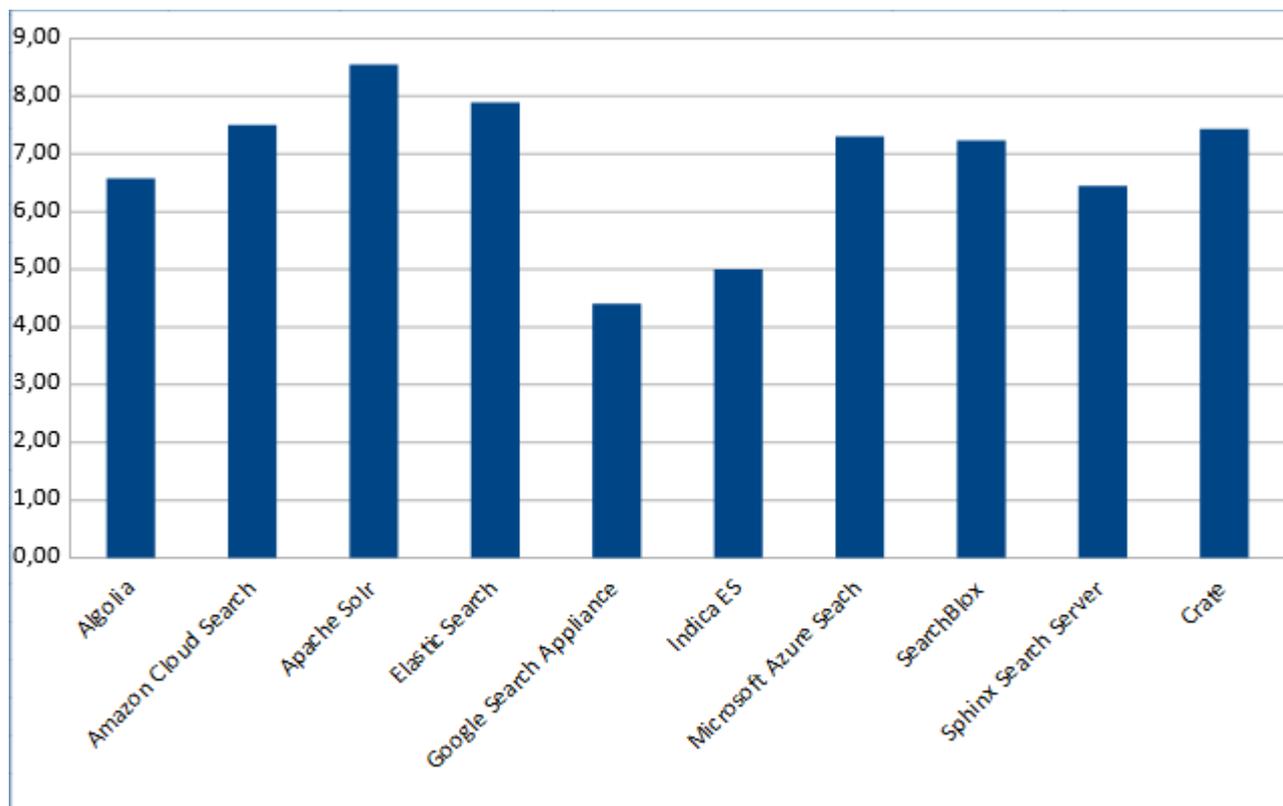


Figura 4.1: Gráfica nota final sobre 10

4.6.1. Comentario sobre la comparativa final global

Como puede observarse, Apache Solr sale vencedor de la comparativa global, analizando las cuatro áreas en su conjunto. En general, las notas de todos son muy próximas entre

sí, y son exclusivamente algunos apartados dentro de cada sección las que marcan la diferencia. Destacar que la victoria de Solr depende de su elevada puntuación en las áreas de soporte y operación, que han sido suficientes para elevarlo hasta la primera posición.

En cuanto al resto de motores, Elasticsearch se lleva la segunda posición de forma relativamente holgada, gracias principalmente también a su cobertura en el área de soporte.

Las siguientes posiciones se las reparten los motores de las grandes plataformas cloud, Microsoft Azure Search y Amazon Cloud Search, gracias a su servicio en la nube y su nota positiva en ejecución.

Sorprendentemente, Crate sin ser un motor de búsqueda puro se mantiene en la parte superior de las notas más altas, beneficiado por su gran documentación, y sobre todo, su gran cobertura en operación.

Por último destacar las notas más bajas del cuadro. Indica ES es un producto relativamente nuevo, y carece de ciertos elementos que van generándose con el tiempo y la experiencia. En concreto, su cobertura de las métricas a nivel de operaciones muy baja, debido a que no posee recursos de aprendizaje aún, ni hacen mención a APIs de integración.

Por su parte Google Search Appliance tiene la característica de funcionar de forma diferente a todos los demás. En general, le ha perjudicado su planteamiento hardware, que conlleva una gran penalización en el gasto de recursos y energía. Además, y como ya hemos mencionado anteriormente, obtiene una nota muy baja en soporte originada sobre todo por los rumores más o menos confirmados de su falta de continuidad en un tiempo relativamente corto.

Capítulo 5

Estudio práctico. Diseño de un repositorio web

5.1. Introducción

En este apartado se va a proceder a la creación de una sencilla aplicación web, que ilustre de cierta manera unas pautas a seguir para emplear estos productos en aplicaciones de usuario.

Cabe destacar que se trata de un pequeño prototipo con poca funcionalidad y que busca exclusivamente reflejar con un ejemplo práctico la facilidad de integración de los motores de búsqueda en este tipo de proyectos.

Para ello, tomando al motor ganador de la comparativa global, construiremos un repositorio con dos funcionalidades o requisitos bien diferenciados:

- Realizar búsquedas: el sistema debe proveer de un método de búsqueda integrado con el motor que permita al usuario realizar consultas sobre la colección de documentos existente en la plataforma.
- Subir documentos: a través de un proceso de inicio de sesión previo, se permitirá a los usuarios registrados subir documentos, aportando ellos mismos los metadatos.

Cabe destacar que para el caso de realizar búsquedas no será necesario ingresar en la plataforma, para que todo aquel que desee realizar una consulta pueda hacerlo. Sin embargo, para mantener un control de los documentos en el servidor, será necesario tener una cuenta en la plataforma para subir ficheros.

Asimismo, la búsqueda realizada permitirá al usuario descargar los ficheros del listado

resultado para su consulta. Esta funcionalidad también estará disponible sin registro en la plataforma.

5.2. Descripción de las tecnologías a emplear

Para construir esta aplicación ejemplo, emplearemos tecnologías ampliamente utilizadas actualmente en el entorno empresarial. Todas ellas poseen documentación extensa, y se han escogido entre otros motivos por su frecuencia de combinación, dado que se emplean en gran cantidad de proyectos académicos comerciales; y su sencilla programación, dentro de sus respectivas curvas de aprendizaje claro está.

5.2.1. Java

Java es un lenguaje de programación orientado a objetos empleado por más de 9 millones de desarrolladores en todo el mundo, y que busca portabilidad de código mediante un sistema semi-interpretado. Esto quiere decir que el código Java solamente se compila una vez, y que a partir de esa compilación se obtiene un programa interpretable por la conocida como JVM (Java Virtual Machine). Ésta funciona como un intérprete que permite a aquel que la tenga instalada emplear programas Java ya compilados. De esta forma, se favorece el código transportable y la universalidad en los desarrollos. En la actualidad, el lenguaje es ampliamente utilizado en desarrollo de aplicaciones web, programación móvil y diseño de juegos.

5.2.2. Spring IO

Spring IO es una plataforma que ofrece un conjunto de frameworks pensados para operar sobre la JVM, de manera que faciliten las tareas más pesadas a los desarrolladores de aplicaciones en Java. También ponen a disposición de los programadores entornos de desarrollo y otras herramientas pensadas para facilitarles las tareas. Todo el software que ofrecen es de código abierto y modular, además de que deja libertad al usuario para emplear sus funcionalidades hasta el punto que él elija.

En este desarrollo mostrado en el capítulo, se emplea uno de los frameworks más conocidos de Spring IO, Spring MVC. Es un framework pensado para el desarrollo de aplicaciones web a través de un patrón de diseño Modelo-Vista-Controlador altamente configurable y sencillo de manejar, dentro de su complejidad. Además, permite su integración total con las clases de Java, lo que le otorga flexibilidad y libertad al programador para usar sus recursos y combinarlos con otros a su gusto.

5.2.3. Apache Tomcat

Tomcat es un contenedor web que permite a los usuarios desplegar aplicaciones en red. Puede funcionar como servidor independiente y es empleado sobre todo junto con tecnología web de Java, las páginas JSP y los servlets. Funciona sobre la Máquina Virtual de Java (JVM), lo que le otorga características multiplataforma, y es ampliamente utilizado en el entorno empresarial para ofrecer servicios y aplicaciones en red. Además, es un sistema de código abierto de la Apache Software Foundation, y, como tal, está soportado por su personal y la comunidad que tiene detrás.

5.2.4. Maven

Maven es una herramienta que facilita la gestión de proyectos Java a través de la aportación de elementos de configuración automáticos para la importación de librerías de software. Esto le otorga un gran valor, dado que gran parte del trabajo que requiere el desarrollo de una aplicación pasa por importar elementos externos útiles para el sistema a crear.

Además de configurar y aportar el código de las librerías de software, facilita la comprensión del código al agrupar todos los elementos importados en un fichero de configuración xml altamente manejable. Así pues, y, especialmente, para aplicaciones de gran tamaño, es un recurso altamente recomendable.

5.2.5. Bootstrap

Bootstrap es un framework que engloba una colección de recursos HTML, CSS y JavaScript para integrar directamente en sitios y aplicaciones web. Es una herramienta ampliamente utilizada, incluso por organizaciones de gran importancia, debido principalmente a que simplifica los diseños de páginas web, al evitar al usuario el trabajo de escribir y declarar sus propios elementos de estilo.

Esta funcionalidad es extensible a la práctica totalidad de navegadores, e incluso es capaz de trabajar en entornos tan exigentes como móviles y tabletas. Además, el proyecto es de código abierto, algo muy positivo a la hora de consultar dudas y de que el producto mantenga su popularidad y evolucione.

5.3. Descripción del entorno de trabajo

Para construir la aplicación se ha empleado el entorno Spring Tool Suite, recomendado por Spring IO para los desarrollos con su framework. La instalación del entorno es muy

sencilla, basta con descargar un zip y descomprimirlo. La zona de trabajo recuerda a la herramienta Eclipse, dado que STS está basado en ella.

Además, el sistema operativo escogido para realizar el proyecto ha sido Windows, en su versión 8.1.

5.4. Análisis y diseño de la aplicación web

Al ser un prototipo sencillo, se ha integrado el análisis de la aplicación con su diseño, debido a que la funcionalidad cubierta no es tan extensa como para ofrecerlas por separado.

En los siguientes apartados se detalla la funcionalidad del sistema, partiendo de los requisitos y los escenarios de uso identificados. Para ello, se detallan los casos de uso que cubren los requisitos descritos, con sus diagramas de secuencia correspondientes, así como la arquitectura de la aplicación y su diseño empleando el framework Spring MVC.

5.4.1. Requisitos

En esta sección se recogen aquellos elementos o características que nuestro sistema ejemplo debe poseer para cumplir su funcionalidad. Aquellos requisitos que sean funcionales será imprescindible cumplirlos tal y como refleja su descripción, mientras que los que son no funcionales quedan a elección del programador.

Requisitos funcionales

- El sistema permitirá la identificación del usuario de forma única para funcionalidad restringida.
- El sistema permitirá añadir nuevos documentos al servidor a usuarios registrados.
- El sistema deberá permitir realizar búsquedas a través del motor de búsqueda.
- El sistema deberá permitir el inicio de sesión a usuarios registrados con sus credenciales.
- El sistema proveerá de interfaces de programación para poder sustituir tanto el motor de búsqueda como el gestor de base de datos por otros compatibles.
- La aplicación estará construida sobre el framework de Spring 4.

Requisitos no funcionales

- La base de datos para el registro de usuarios estará construida con el gestor MySQL.

- El sistema utilizará Apache Tomcat como servidor web.
- El sistema empleará Solr como motor de búsqueda para la aplicación.
- El sistema se construirá a través de tecnología HTML y Bootstrap.

5.4.2. Casos de uso

Los casos de uso resumen el comportamiento que ha de tener el sistema dependiendo de la funcionalidad que se le requiera en cada momento. Es decir, que si el usuario desea realizar la actividad A, el caso de uso englobará aquellos procesos que deba seguir el sistema para que esa actividad A cumpla sus objetivos.

En nuestro caso, tenemos dos casos de uso principales que engloban la funcionalidad de la aplicación. En la siguiente imagen puede observarse el diagrama correspondiente que resumen ese comportamiento.

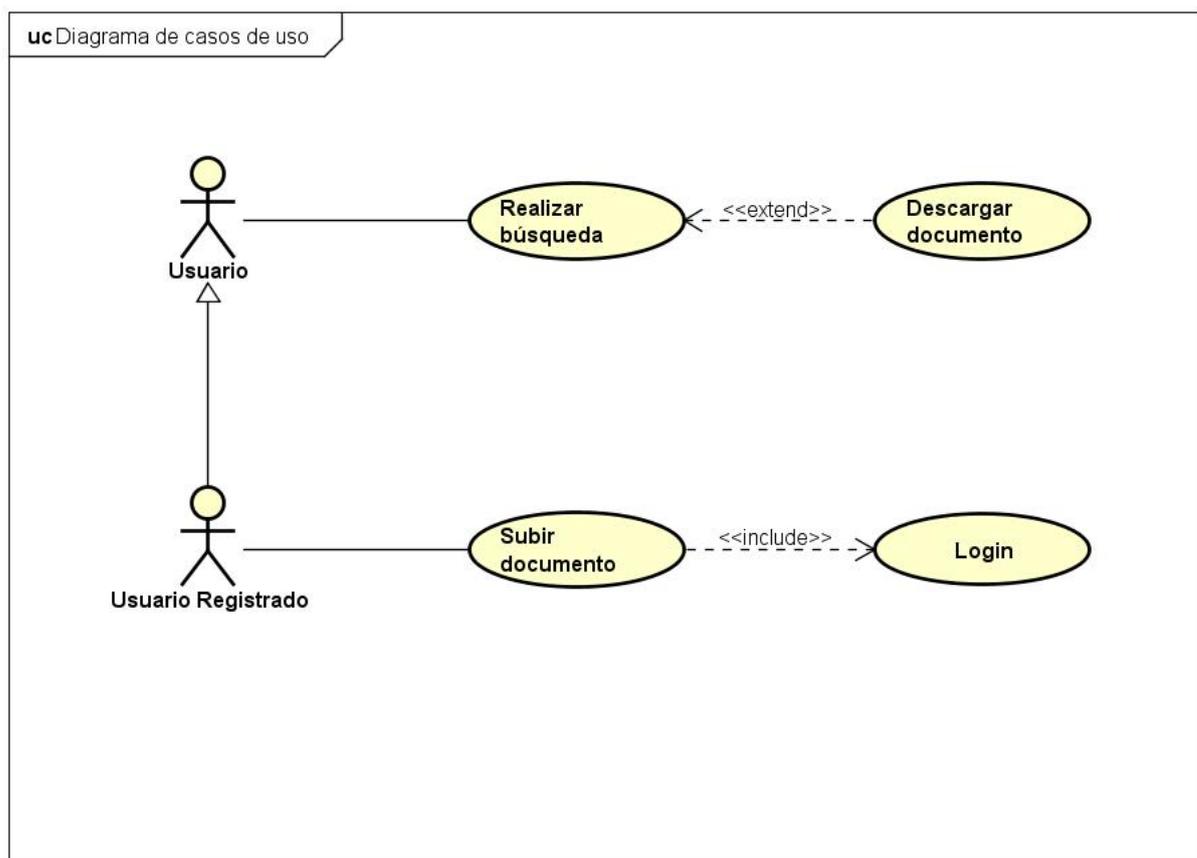


Figura 5.1: Diagrama de casos de uso

Como puede observarse en el diagrama, existen dos tipos de usuario (actores) para la aplicación: registrado y no registrado. El no registrado puede operar exclusivamente

con el caso de uso principal Realizar búsqueda, que tiene asociado la opción de realizar el de Descargar documento. El usuario registrado puede ejecutar los casos de uso relacionados con el usuario normal, además de desarrollar el suyo propio: Subir documento. Evidentemente, para cumplirlo, necesita realizar la operación Login o Inicio de sesión.

A continuación se expone el flujo de operaciones que siguen los casos de uso del diagrama, a modo de tabla para facilitar su lectura.

Identificador	CU-01	
Nombre	Realizar búsqueda	
Descripción	El sistema devuelve un listado con los documentos que posean la cadena de búsqueda	
Precondición	Ninguna	
Secuencia Normal	Paso	Acción
	1.	El usuario introduce el texto a buscar
	2.	El sistema procesa la solicitud del cliente
	3.	El motor de búsqueda devuelve los resultados correspondientes
	4.	El sistema muestra los resultados al cliente
Postcondición	Ninguna	
Secuencia Alternativa	Paso	Acción
	4.1.	Si ningún documento cumple con los criterios, la búsqueda se muestra como infructuosa

Tabla 5.1: CU-01. Realizar búsqueda

Identificador	CU-02	
Nombre	Descargar documento	
Descripción	El usuario descarga un documento a través de un enlace	
Precondición	El usuario ha realizado una búsqueda	
	El fichero está presente en el listado de búsqueda	
Secuencia Normal	Paso	Acción
	1.	El usuario pulsa sobre el enlace con el nombre del fichero para descargarlo.
	2.	El sistema envía el fichero al usuario y se descarga.
Postcondición	El fichero está en la máquina del usuario	

Tabla 5.2: CU-02. Descargar fichero

Identificador	CU-03	
Nombre	Gestionar sesión	
Descripción	Gestionar el inicio y cierre de sesión de un usuario del sistema.	
Precondición	El usuario posee una cuenta en el sistema.	
Secuencia Normal	Paso	Acción
	1.	Un usuario del sistema introduce su nombre y contraseña para iniciar sesión.
	2.	El sistema comprueba que exista esta cuenta.
	3.	El usuario entra en el sistema.
	4.	El usuario pulsa la opción de cierre de sesión.
	5.	El sistema cierra la sesión del usuario.
Postcondición	Se valida e invalida la sesión del usuario	
Secuencia Alternativa	Paso	Acción
	2.1.	Si los datos introducidos no son correctos el caso de uso termina y queda sin efecto.

Tabla 5.3: CU-03. Gestionar sesión

Identificador	CU-04	
Nombre	Subir fichero a la plataforma.	
Descripción	El sistema sube un fichero a la plataforma y lo registra en el índice del motor	
Precondición	El usuario ha iniciado sesión en el sistema.	
	El documento del usuario está disponible para subir	
Secuencia Normal	Paso	Acción
	1.	El cliente entrega los metadatos del fichero, así como el fichero mismo.
	2.	El sistema almacena el fichero y registra su información en el índice del motor
Postcondición	El documento se encuentra subido y su información en el índice	

Tabla 5.4: CU-04. Subir fichero al sistema

5.4.3. Diagramas de secuencia

En los diagramas de secuencia siguientes se desarrollan las interacciones entre los componentes del sistema a fin de cubrir los casos de uso previamente descritos. Para cada acción, de forma más o menos acertada, el usuario realiza una solicitud a la vista, la cual es procesada por un controlador, accediendo al modelo de datos para actualizar o mostrar una nueva vista.

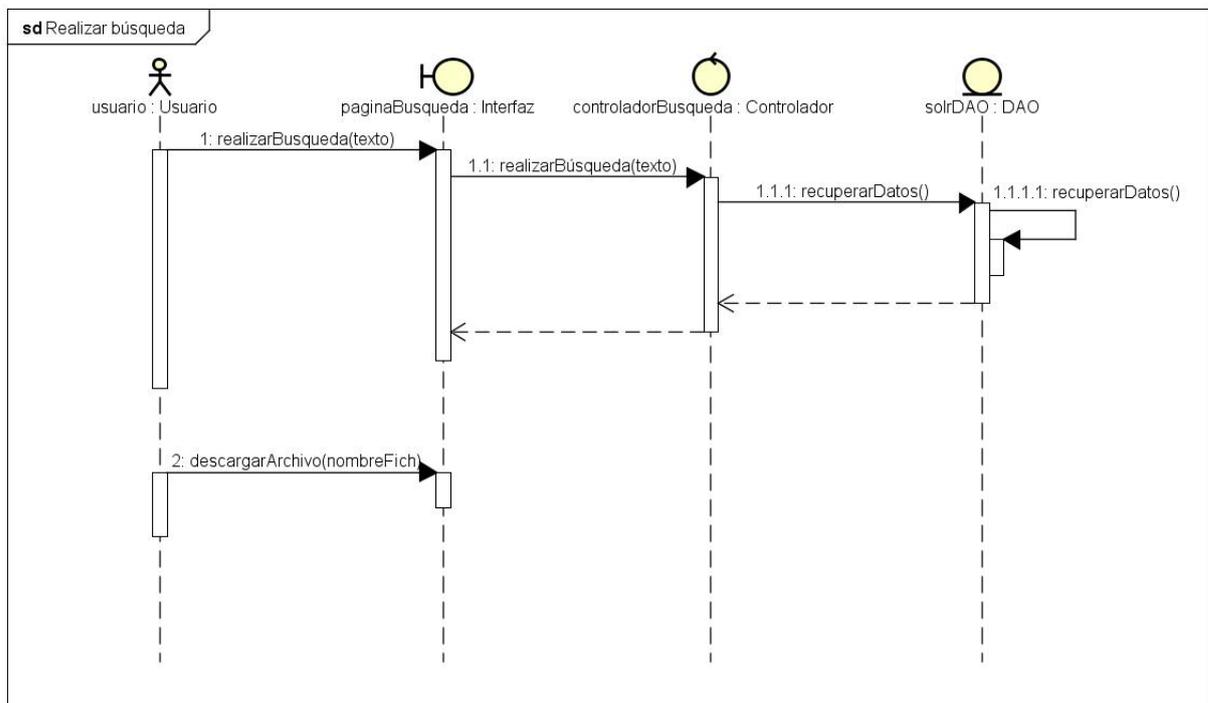


Figura 5.2: Realizar búsqueda.

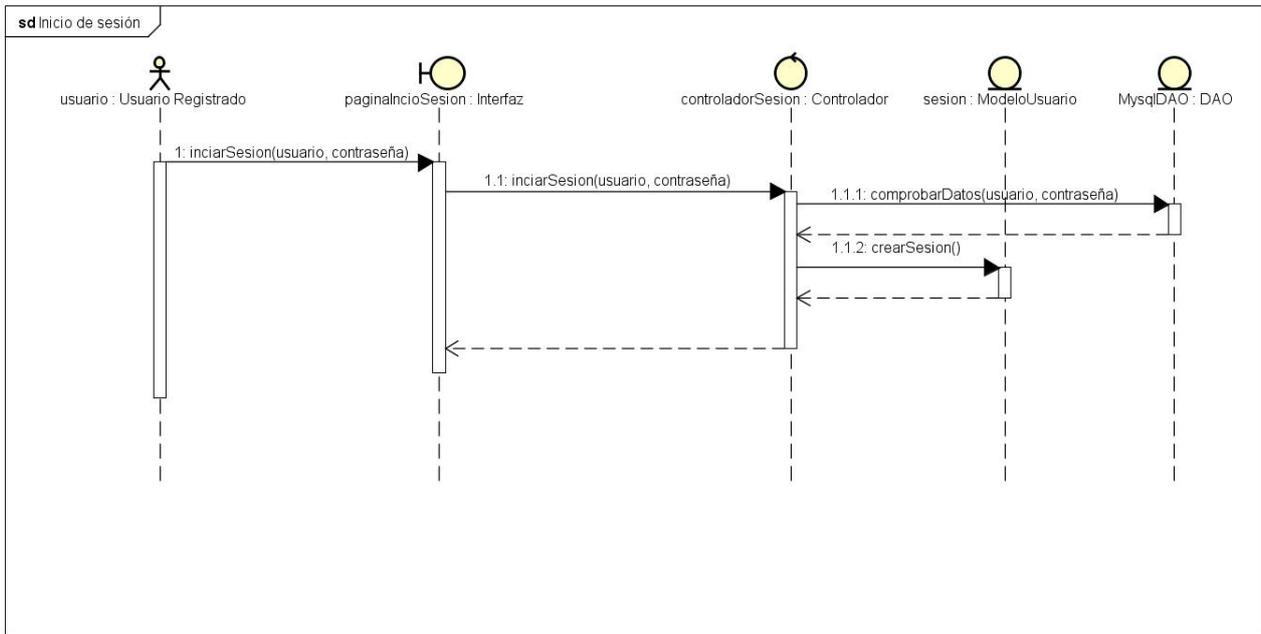


Figura 5.3: Diagrama de secuencia. Inicio de sesión.

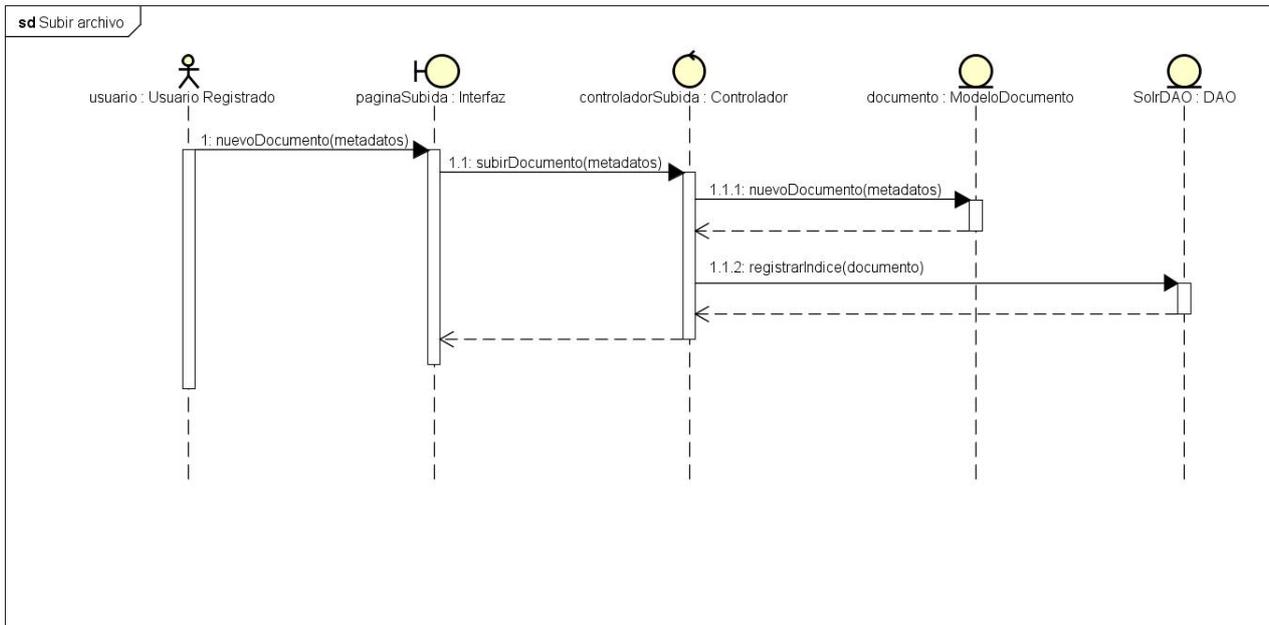


Figura 5.4: Diagrama de secuencia. Subir archivo.

5.4.4. Arquitectura escogida. Funcionamiento de Spring MVC

Se emplea a la hora de repartir la funcionalidad desde el punto de vista físico el modelo cliente-servidor para construir la aplicación web. Es un modelo en el que las tareas del

sistema se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Es decir, consiste básicamente en que un usuario realiza peticiones de servicio a otro programa, residente en el servidor, que de forma remota le da una respuesta.

Es el modelo óptimo para un sistema de las características buscadas, ya que se desea que el sistema contenga los archivos por los cuales el usuario navegará y efectuará sus operaciones. Además, el cliente empleará exclusivamente su navegador web para operar con el sistema de forma remota, sin precisar de nada más.

Entre las ventajas de este modelo se encuentran el control sobre los datos por parte del proveedor del servicio, así como la opción de la escalabilidad, que le permite modificar las prestaciones del servicio para evitar que el aumento de la demanda acabe con su calidad.

El patrón de diseño funcional del que se va a hacer uso es el Modelo-Vista-Controlador. Es un patrón arquitectónico muy difundido en la creación de aplicaciones Web. Pese a que hay distintas metodologías a la hora de aplicarlo e implementarlo, en esencia los elementos principales son comunes a todas ellas y se mantienen en sus diferentes versiones. Esos tres elementos principales son los que se reparten la funcionalidad, y los que orientan el flujo de operación de la aplicación:

- **Modelo:** Es el responsable de toda la lógica y estado del dominio de negocio. Dependiendo del tipo de arquitectura sobre el que se está construyendo la aplicación, puede seguir distintos patrones en su diseño.
- **Vista:** Se encarga de la presentación del dominio de negocio, de mostrar los datos del modelo de forma agradable e interactiva al usuario. Está compuesta por aquellos elementos que aportan funcionalidad a la presentación, como jsps, páginas html, etc.
- **Controlador:** Es responsable del flujo de control de la aplicación, la navegabilidad y las configuraciones. Además, se encarga de la lógica del negocio y permite que las peticiones del usuario modifique el flujo y el comportamiento de la aplicación.

De esta manera, la parte referente a la Vista en MVC será presentada en la parte del cliente, mientras que la lógica de negocio (Controlador) y los datos (Modelo) residirán en el servidor.

Tal y como hemos expuesto pues, existen numerosas variantes del patrón MVC a la hora de repartir la funcionalidad. De esta manera, en este caso se aplica la versión de Spring MVC para el desarrollo de esta pequeña aplicación. El funcionamiento de Spring MVC es muy sencillo: mantiene la funcionalidad de las vistas JSP clásicas de Java, así como el modelo, que no difiere de la forma clásica de hacerse. La diferencia se focaliza en la parte del controlador. Existen dos elementos diferenciados en esta parte:

- **Dispatcher servlet:** El dispatcher servlet es el controlador principal. Por él pasan todas las solicitudes, y también todas las respuestas. Es el encargado de entregar las

solicitudes al resto de controladores para que realicen sus funciones, le respondan y el devuelva esa respuesta.

- **Controladores:** Los controladores son los elementos clásicos encargados de procesar peticiones recibidas desde el dispatcher servlet. Cada controlador resuelve su propia funcionalidad separada, y es responsable de la interacción con el modelo y los datos.

En la siguiente imagen se expone el funcionamiento de Spring de forma visual:

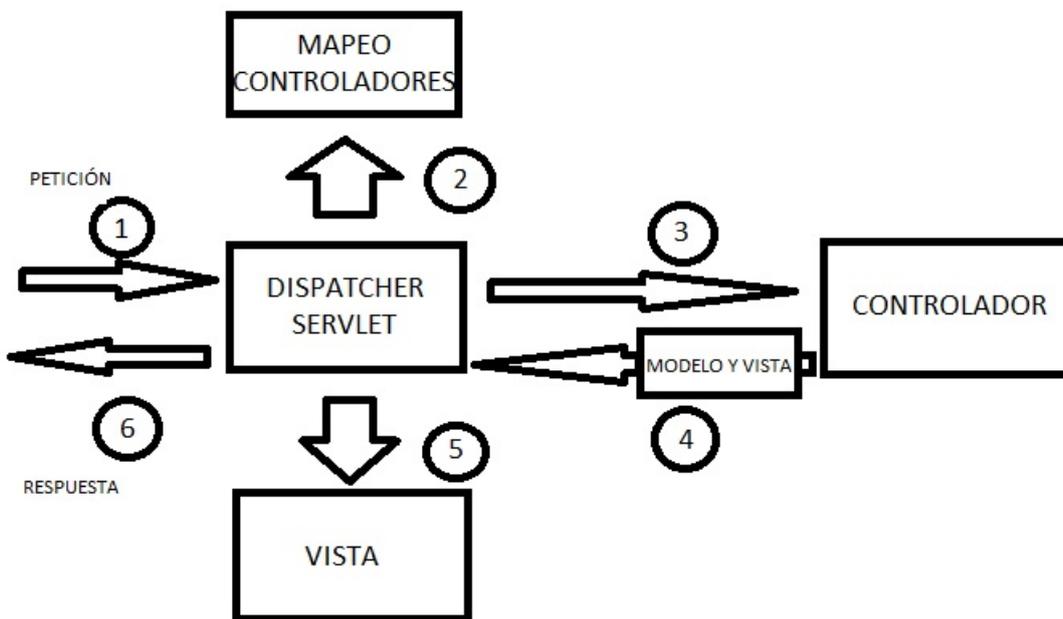


Figura 5.5: Diagrama funcionamiento SpringMVC.

5.4.5. Diagrama de clases de análisis y diseño

Al poseer un modelo de datos tan sencillo con solamente dos clases, Usuario y Documento, se ha decidido integrar el diagrama de clases de diseño con el de análisis, exponiendo el funcionamiento final de la plataforma.

Como puede observarse, el flujo de funcionalidad pasa en su totalidad por el dispatcher servlet, que reenvía las solicitudes a los respectivos controladores responsables, encargados de manejarlas. Para ello, acceden a los objetos de acceso a datos, implementados a partir de interfaces para que puedan sustituirse los componentes por otros deseados por el usuario.

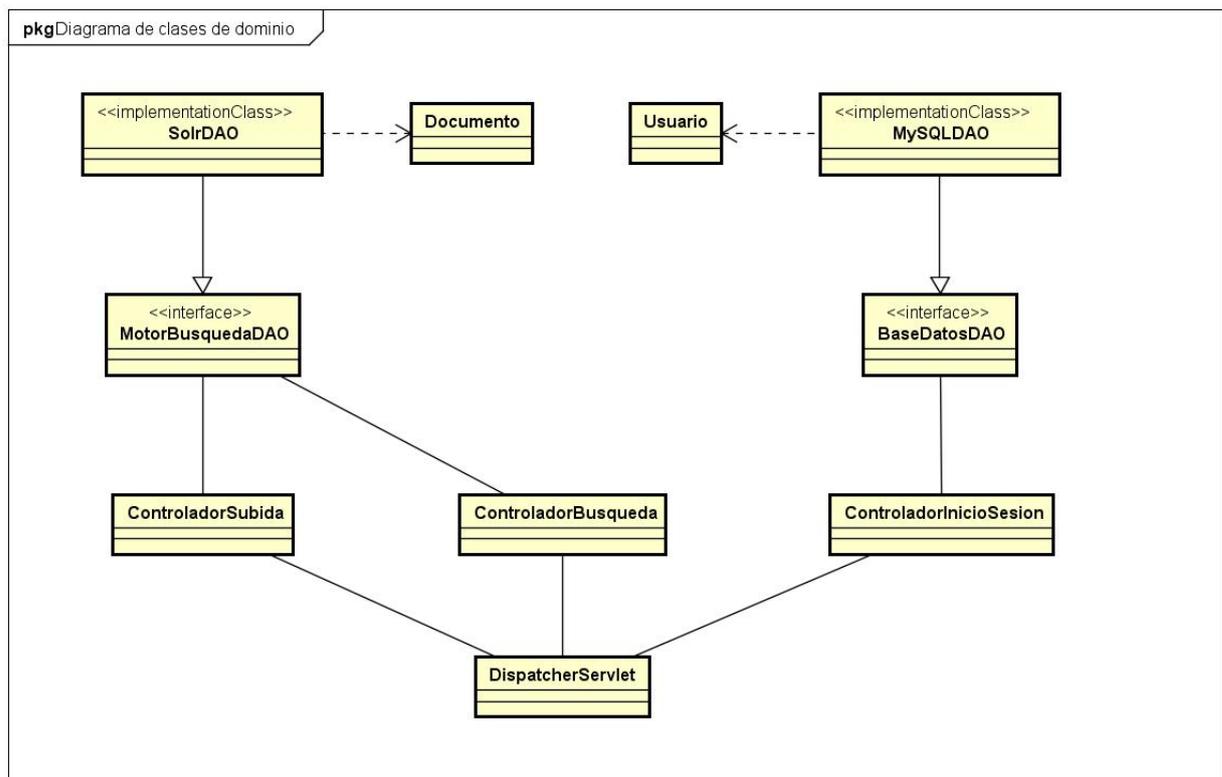


Figura 5.6: Diagrama clases de diseño.

5.5. Desarrollo de la aplicación

5.5.1. Introducción

La aplicación tal y como se ha expuesto se realiza a través del entorno STS con el framework de Spring. Sin embargo, el proceso de desarrollo con este sistema requiere de cierta preparación inicial, la creación de una base que permita el funcionamiento del sistema según las características de Spring.

Además, para implementar el proyecto se han seguido las recomendaciones de estructura de directorios para un proyecto web, depositando el código según dictan los estándares.

A fin de aplicar esa configuración inicial, se ha resumido ese proceso inicial en el apartado de creación del proyecto base.

5.5.2. Creación del proyecto base

A fin de crear el sistema de carpetas, se ha generado un proyecto con Maven en la plataforma de STS. A través del fichero pom.xml de Maven, añadimos las librerías correspondientes a Spring, el conector de MySQL, que vamos a emplear en el proceso de inicio de sesión, y el conector de Solr, para poder trabajar con el motor de búsqueda.

Asimismo implementamos el dispatcher_servlet, tal y como hemos expuesto en la descripción del modelo MVC de Spring. Este controlador frontal servirá para redireccionar las solicitudes al resto de controladores en respuesta a acciones del usuario desde las vistas. Para desarrollarlo, creamos un fichero en formato .xml con el nombre de la aplicación, en el que insertamos la configuración recomendada por el framework. Además, activamos la opción de escaneo de controladores, mediante la cual el dispatcher_servlet explora el sistema de carpetas del proyecto, mapeando los controladores etiquetados con el tag @Controller de forma automática.

Una vez creados los elementos clave de Spring, generamos carpetas para almacenar el resto de ficheros según el sistema de carpetas expuesto en la sección Contenido del CD, al final del presente documento. Una vez llegados a este punto, ya estamos preparados para definir las vistas y controladores con los que implementar toda la funcionalidad.

5.5.3. Implementación del sistema de inicio de sesión

Para implementar el sistema de inicio de sesión, se ha decidido que sea el administrador del sistema quien registre a los usuarios previamente en la base de datos del servicio. De esta manera, se evitan los problemas de registros masivos e incontrolados en la plataforma, y se garantiza que exclusivamente aquellos usuarios con permisos accedan a los servicios restringidos.

Una vez dicho esto, se procede a explicar el planteamiento seguido para desarrollar este proceso:

- **Creación de la vista de inicio de sesión:** De esta manera se ofrece una interfaz más o menos sencilla de identificación al usuario, solicitándole su usuario y contraseña de manera clásica, tal y como hacen otras plataformas.
- **Creación de un controlador para acceder a la vista:** Para comunicar la vista principal con la de inicio de sesión se define un controlador que es llamado cuando se solicita el enlace de iniciar sesión. Este al recibir la llamada, la procesa y en la respuesta envía la página de login.
- **Creación de un controlador para recibir los datos:** Una vez que el usuario ha introducido los datos pertinentes, se define un nuevo controlador que procesa los datos recibidos. Para ello accede a un DAO (Data Access Object) definido a través

de una interfaz. Esto permite que el gestor de base de datos y sus procesos asociados puedan sustituirse por los que el usuario desee, facilitando la modularidad.

En nuestro caso, hemos hecho una implementación para ese DAO en su versión para MySQL, de forma que el controlador consulta al DAO si el usuario existe en la base de datos, y en caso de ser afirmativo, crea una sesión para el usuario y devuelve la vista principal modificada para la nueva funcionalidad del usuario registrado.

5.5.4. Configuración del motor de búsqueda

A fin de emplear a Solr como motor de búsqueda, debe realizarse previamente la instalación del mismo. En Windows el proceso es sencillo: se accede a la página de descargas del producto y se descarga la versión comprimida en formato .zip del software. Ese fichero puede descomprimirse en un directorio cualquiera del usuario, quedando listo para su uso. Para arrancar el servicio, se accede a la subcarpeta bin y mediante la consola de comandos se inicia el servicio con la sentencia *solr start*.

Para operar con Solr, creamos un nuevo nodo con un índice asociado, al que le asignamos un esquema para determinar que campos va a almacenar el motor en su base de datos y a través de cuales va a permitir las búsquedas.

Por último implementamos un DAO (Objeto de Acceso a Datos) a partir de una interfaz para poder realizar las operaciones de inserción, rellenando los campos obligatorios requeridos por el índice, y de búsqueda. La operación de búsqueda, al ser un prototipo sencillo se hace exclusivamente realizando búsquedas de texto en los metadatos de los documentos, sin acceder al contenido de ninguno de ellos, aunque cabe resaltar que es común en los motores de búsqueda hacer consultas sobre los metadatos y los propios datos.

5.5.5. Integración del motor de búsqueda en la aplicación

La integración del motor merece una mención aparte. Al igual que para el acceso a los datos de la base de datos, se ha implementado una interfaz que permita al usuario sustituir el motor por el que él mismo desee.

En cuanto a la clase DAO que se ha implementado empleando esa interfaz para Solr, posee dos métodos que son útiles para completar los casos de uso de subida de archivos y las búsquedas. A través de dos funciones distintas, Solr procesa bien los metadatos para agregar el fichero a la colección, bien la solicitud del usuario devolviendo el conjunto de ficheros que cumplen los criterios de búsqueda.

5.5.6. Implementación del sistema de subida de documentos y registro en el índice

Para desarrollar el sistema de subida de documentos se ha buscado que el propio usuario suba los metadatos del fichero, así como el documento mismo. Con esa información, el sistema recoge la ruta del fichero y las almacena en el índice del motor de búsqueda con sus metadatos para que en el futuro los usuarios puedan hacer consultas sobre él.

Cabe destacar que es imprescindible que el usuario esté registrado en la plataforma y con una sesión iniciada para poder acceder a este servicio.

Debido al poco tiempo de desarrollo disponible y a intentar emplear el framework de la forma más sencilla posible, se ha dividido la funcionalidad en pedir inicialmente los metadatos, y recoger a continuación el fichero, según el siguiente proceso:

- **Creación de la vista de subida de metadatos:** El usuario puede emplear esta vista para rellenar los metadatos básicos del documento: autor, título, tipo de documento y descripción. A fin de dar algo más de cobertura, se permite al usuario subir ficheros tanto de texto, como gráficos.
- **Creación de un controlador para acceder a la vista:** Para comunicar la vista principal con la de subir documento se define un controlador que es llamado cuando se solicita el enlace correspondiente. Este al recibir la llamada, la procesa y responde enviando la vista de inserción de metadatos.
- **Creación de un controlador para recibir los metadatos:** Una vez que el usuario ha introducido los datos pertinentes, se define un nuevo controlador que procesa los datos recibidos, almacenándolos provisionalmente y devolviendo la vista de subida del fichero.
- **Creación de la vista de subida de metadatos:** El usuario puede emplear esta vista para subir el fichero que desea añadir a Solr asociado a los metadatos que ha introducido previamente. Para facilitar la programación, se deja libertad al usuario para asignar los metadatos libremente al documento.
- **Registro del fichero en el índice del motor de búsqueda:** Tras haber introducido el fichero correspondiente, se define un nuevo controlador que procesa la solicitud. Para ello accede al DAO de Solr que se ha descrito previamente, definido a través de una interfaz. Juntando los metadatos obtenidos en la primera solicitud con la ruta del fichero, se almacena toda esa información en el índice de Solr.

Para que el prototipo sea más visual, se ha configurado el DAO para que realice un commit, es decir, una actualización del índice, cada vez que se sube un fichero, aunque no es lo recomendable. Esto simplemente busca demostrar que tras subir un fichero resulta inmediato realizar la búsqueda asociada a él y recuperarlo.

5.5.7. Implementación de búsqueda y descarga de documentos

Por último, el sistema de búsqueda de Solr tiene un funcionamiento similar al de los otros procesos, manteniendo la estructura lógica de Spring. El usuario desde el cuadro de búsqueda de la página principal realiza su consulta introduciendo la cadena que desea buscar. Esta se procesa por el sistema, que consulta el índice de Solr y según la configuración que se ha descrito previamente, devuelve el listado de documentos que cumplen con la consulta. De ese listado de documentos, el usuario escoge el que le interesa, y pulsando sobre el enlace del título, procede a su descarga.

Este proceso tal y como se ha expuesto en apartados anteriores no requiere de un inicio de sesión, y por tanto, cualquier usuario de la plataforma puede acceder al mismo:

- **Creación de la vista de resultados:** Dado que la página de inicio se creó con el desarrollo del proyecto base, se debe crear una vista nueva que recoja los posibles resultados de la solicitud del usuario, además de proveer de un nuevo cuadro de búsqueda para que el usuario no tenga que regresar a la vista anterior.
- **Creación de un controlador para realizar la búsqueda y acceder a la vista:** Una vez el usuario ha introducido su solicitud en el sistema, el controlador la recoge y la procesa. Para ello accede al DAO de Solr que ya se ha empleado en la subida, pero por su funcionalidad de búsqueda. El motor comprueba que documentos satisfacen la solicitud, y, hecho esto, devuelve el conjunto de documentos correspondiente. El controlador entonces devuelve la vista con los resultados en caso de que los hubiera, o por el contrario con un mensaje de que la búsqueda no ha devuelto resultados.
- **Proceso de descarga:** Una vez que el usuario ha recibido el listado de ficheros, puede pulsar sobre los enlaces, y mediante el sistema de descargas de HTML, descargar el fichero a su máquina.

5.6. Pruebas sobre el desarrollo

En este apartado se recogen algunos escenarios de uso para el caso en el que los usuarios empleen la plataforma, comprobando que los flujos de la aplicación se aplican correctamente y cómo el sistema reacciona dependiendo de las interacciones del actor:

- **Intento de sesión fallido:** En el caso de que ocurra esto, el controlador devuelve la vista de inicio de sesión con un mensaje de error que indica que alguno de los datos era incorrecto. En caso de que el usuario no haya rellenado alguno de los datos, es el propio control del formulario en la página el que impide al usuario ingresar.
- **Intento de inicio de sesión exitoso:** Cuando el inicio de sesión tiene éxito, el sistema crea una sesión para el usuario y muestra la página inicial, con nueva funcionalidad para subir documentos y cerrar la sesión.

- **Datos requeridos del formulario de metadatos no introducido:** En caso de que el usuario no haya rellenado alguno de los datos, es el propio control del formulario en la página el que impide al usuario ingresar. El sistema en sí no posee control de metadatos, ya que en el diseño del prototipo se permite al usuario añadir los metadatos que desee.
- **Campo para el fichero que intenta subir el usuario inexistente:** En caso de que el usuario intente subir un fichero sin rellenar el campo del formulario, el sistema responde con un mensaje que solicita el fichero.
- **Proceso de subida de fichero exitoso:** Si el fichero se sube exitosamente, el sistema no tiene prevista una respuesta positiva. Simplemente devuelve la página de inicio para que el usuario pueda continuar con sus operaciones.
- **Consulta sin ningún resultado:** Cuando el usuario realiza una consulta, el sistema comprueba en el índice de Solr si la petición tiene asociada algún archivo. De no ser así, el sistema devuelve la página de resultados, indicando que ningún elemento residente en el sistema coincide con el parámetro de búsqueda.
- **Consulta exitosa:** Por el contrario, si la búsqueda realizada por el usuario posee datos asociados, el sistema muestra el listado con los metadatos de los ficheros resultado.
- **Fichero ya subido a la plataforma:** En el caso de que el usuario suba un fichero ya subido a la plataforma, el sistema impide que el fichero suba al servidor, dado que no pueden coexistir ambos en el directorio de subida.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

A lo largo del presente documento se ha intentado exponer de una forma lo más clara y concisa posible la realización de una evaluación de productos de búsqueda desde un enfoque diferente. Sin embargo, cabe destacar unas cuantas ideas extraídas del estudio.

En la actualidad, resulta interesante evaluar los productos de búsqueda debido a su extenso uso en numerosos ámbitos académicos y empresariales. También se emplean en ámbitos de redes sociales Big Data. Sin embargo, las comparativas clásicas evalúan exclusivamente algunas características numéricas de los motores de búsqueda, llegando incluso a dejar fuera de las mismas a algunos productos bien por intereses comerciales, bien por otras dificultades. A pesar de ello, existe la posibilidad de realizar una comparativa cualitativa, que aunque menos precisa y más compleja de hacer en ciertos aspectos que la cuantitativa, permite evaluar todas las características que no suelen analizarse normalmente.

Además, este estudio permite extraer conclusiones valiosas que de otro modo no podrían ser contempladas, con resultados interesantes que pueden asimismo emplearse para discriminar los productos que mejor se adaptan a las circunstancias. Por último, destacar que este tipo de análisis puede emplearse también como complemento a un análisis cuantitativo, obteniendo una mejor visión de todos estos productos.

6.2. Trabajo futuro

A modo de conclusión, se proponen algunas posibles mejoras y campos de investigación posibles para ampliar el alcance de este estudio:

- Incrementar el número de parámetros de estudio en el benchmark.
- Explorar un mayor número de motores de búsqueda y soluciones.
- Colaborar con los propios proveedores para poder acceder a una mayor cantidad de información y recursos evaluables.
- Hacer cuadros comparativos más visuales y útiles a las empresas.
- Mejorar el ejemplo de aplicación para que englobe una mayor funcionalidad.
- Mejorar asimismo el ejemplo de aplicación para que cubra aspectos como la seguridad o la cobertura de búsqueda de una forma más profunda.

Bibliografía

- [Wal14] Craig Walls. *Spring*. Anaya, 2014.
- [Alg] Algolia. *Algolia Main Page*. URL: <https://www.algolia.com/> (visitado 11-07-2017).
- [BBC] BBC. *How do search engines work?* URL: <http://www.bbc.co.uk/guides/ztbjq6f> (visitado 11-07-2017).
- [Booa] Bootstrap. *Bootstrap components*. URL: <http://getbootstrap.com/2.3.2/components.html> (visitado 11-07-2017).
- [Boob] Bootstrap. *Bootstrap main page*. URL: <http://getbootstrap.com/2.3.2/> (visitado 11-07-2017).
- [BV] Elasticsearch BV. *Document APIs*. URL: <https://www.elastic.co/guide/en/elasticsearch/reference/0.90/docs.html> (visitado 11-07-2017).
- [Cra] Crate. *Crate.IO Main Page*. URL: <https://crate.io/> (visitado 11-07-2017).
- [Ela] Elasticsearch. *ElasticSearch Main Page*. URL: <https://www.elastic.co/> (visitado 11-07-2017).
- [DBE] DB-Engines. *Elasticsearch vs Solr vs Sphinx*. URL: <https://db-engines.com/en/system/Elasticsearch;Solr;Sphinx> (visitado 11-07-2017).
- [Fou] The Apache Software Foundation. *Solr Tutorial*. URL: http://lucene.apache.org/solr/4_9_0/tutorial.html (visitado 11-07-2017).
- [Gara] Gartner. *Magic Quadrant for Enterprise Search - 2013*. URL: <https://www.extended-content.com/wp-content/uploads/2013/05/Gartner-Magic-Quadrant-For-Enterprise-Search.pdf> (visitado 11-07-2017).
- [Garb] Gartner. *Magic Quadrant for Enterprise Search - 2015*. URL: <http://www.project-consult.de/files/Garter%5C%20Magic%5C%20Quadrant%5C%20for%5C%20Enterprise%5C%20Search%5C%20Q3%5C%202015.pdf> (visitado 11-07-2017).
- [Gooa] Google. *Google Search Appliance 7.6*. URL: <https://enterprise.google.es/intl/es/search/products/gsa.html> (visitado 11-07-2017).
- [Goob] Google. *Google Search Appliance Main Page*. URL: <https://enterprise.google.es/intl/es/search> (visitado 11-07-2017).
- [Gooc] Google. *GSA - JSON Communication*. URL: <https://developers.google.com/custom-search/json-api/v1/overview> (visitado 11-07-2017).

- [Ind] Indica. *Indica ES Main Page*. URL: <http://www2.indica.nl/enterprisesearch> (visitado 11-07-2017).
- [Keh] Miles Kehoe. *Why Elasticsearch is not in Gartner chart?* URL: <https://www.quora.com/Why-elastic-search-is-not-in-gartner-chart> (visitado 11-07-2017).
- [Kel] Toby Bussa Kelly M. Kavanagh Oliver Rochford. *2016 Magic Quadrant for SIEM*. URL: <https://www.securelink.be/wp-content/uploads/sites/2/2016-Magic-Quadrant-for-SIEM.pdf> (visitado 11-07-2017).
- [Mic] Microsoft. *Microsoft Azure Search Main Page*. URL: <https://azure.microsoft.com/es-es/services/search/> (visitado 11-07-2017).
- [Mor] Francisco Grimaldo Moreno. *Spring paso a paso*. URL: <http://www.uv.es/grimo/teaching/SpringMVCv4PasoAPaso/overview.html> (visitado 11-07-2017).
- [ORi] Adrian O’Riordan. *Open Search Environments: The Free Alternative to Commercial Search Services*. URL: <https://ejournals.bc.edu/ojs/index.php/ital/article/view/4520/pdf> (visitado 11-07-2017).
- [Péra] Alberto Pérez. *Consultas en Solr*. URL: <http://www.brujuleo.es/consultas-en-solr/> (visitado 11-07-2017).
- [Pérb] Alberto Pérez. *Esquemas en Solr*. URL: <http://www.brujuleo.es/esquemas-en-solr/> (visitado 11-07-2017).
- [Piv] Inc. Pivotal Software. *Getting started with Spring Tool Suite*. URL: <https://spring.io/guides/gs/sts/> (visitado 11-07-2017).
- [Seaa] Amazon Cloud Search. *ACS Main Page*. URL: <http://aws.amazon.com/> (visitado 11-07-2017).
- [Seab] Solr Search. *Solr Main Page*. URL: <http://lucene.apache.org/solr/> (visitado 11-07-2017).
- [Seac] SearchBlox. *SearchBlox Search Main Page*. URL: <https://www.searchblox.com/> (visitado 11-07-2017).
- [Sha] ShawnHeisey. *FAQ Solr*. URL: <https://wiki.apache.org/solr/FAQ> (visitado 11-07-2017).
- [Sph] Sphinx. *SphinxSearchServer Main Page*. URL: <https://www.searchblox.com/> (visitado 11-07-2017).
- [Tan] Kelvin Tan. *SolrJ Tutorial*. URL: <http://www.solrtutorial.com/solrj-tutorial.html> (visitado 11-07-2017).
- [Tho] Portia Tung Thomas Risberg Rick Evans. *Spring step by step*. URL: <https://docs.spring.io/docs/Spring-MVC-step-by-step/index.html> (visitado 11-07-2017).
- [Yig] Asaf Yigal. *Solr vs Elasticsearch*. URL: <https://logz.io/blog/solr-vs-elasticsearch/> (visitado 11-07-2017).

Palabras clave

En este apartado se recogen las palabras más importantes y necesarias para comprender conceptos básicos expuestos en la memoria.

- **Motor de búsqueda:** programa informático que realiza búsquedas en un conjunto de documentos, especialmente en Internet, para una palabra o palabras dadas, devolviendo un listado ordenado de los resultados.
- **Crawling:** técnica explorativa de documentos que emplean los motores de búsqueda para, mediante enlaces, encontrar elementos a indexar.
- **Indexar:** proceso que se encarga de contruir listas y ordenar la información sobre los documentos explorados, a fin de facilitar su búsqueda posterior.
- **Consulta:** palabra o conjunto de palabras que emplea el usuario para que el motor realice la búsqueda de elementos relacionados.
- **Búsqueda:** proceso mediante el cual el motor accede al índice construido para localizar documentos relacionados con la consulta.
- **Métrica:** medida o conjunto de medidas que resumen el comportamiento de un sistema o programa
- **SaaS:** método de distribución de software en el que el distribuidor alberga todo el servicio y los datos en sus servidores y el cliente accede a ellos desde la web.
- **Cloud:** propuesta tecnológica que busca ofrecer servicios a través de la red tanto de computación como de almacenamiento.
- **Open Source:** también conocido como código abierto, hace referencia al software que publica el contenido del software a fin de que este sea de libre distribución para poder modificar la fuente de los programas sin restricciones de licencia.
- **Metadatos:** se define como un conjunto de datos que permite describir otros datos. En el ámbito de la informática son los datos que permiten identificar otros elementos de información de forma unívoca.
- **API:** conjunto de elementos software preparados para funcionar como una caja negra, que el usuario emplea para acceder a métodos y funciones que le permiten comunicarse con el servicio y operar con él.

Contenido del CD adjunto

A fin de facilitar la lectura, se indican los ficheros en cursiva.

- *memoria.pdf*
- ConfiguracionSolr
 - *schema.xml* (Fichero de configuración del índice de Solr)
- Prototipo
 - Aero (nombre dado a la aplicación)
 - *.settings*
 - *src* (ficheros fuente)
 - *target*
 - *.classpath*
 - *.project*
 - *pom.xml* (fichero de configuración de Maven)
- ScriptsMySQL
 - *CREAR_BBDD.sql*
 - *CREAR_TABLA_USUARIO.sql*
 - *INSERTAR_USUARIOS.sql*