**ÍNDICE GENERAL**

[1. Introducción](#_Toc454867310) 2

[1.1. Antecedentes](#_Toc454867311) 2

[1.2. Definición del problema](#_Toc454867312) 2

[1.3. Objetivos](#_Toc454867313) 3

[1.3.1. Objetivo general](#_Toc454867314) 3

[1.3.2. Objetivos específicos](#_Toc454867315) 3

[1.4. Innovación tecnológica](#_Toc454867316) 3

[1.5. Justificación 4](#_Toc454867317)

[1.6. Metodología de desarrollo](#_Toc454867318) 4

[1.6.1. Matriz de objetivos](#_Toc454867314) 5

[1.6.2. Diagrama de Gantt](#_Toc454867315) 7

[1.7. Alcance …8](#_Toc454867319)

1.8. Bibliografía……………………………………………………………………………………………………………………………….8

1. Introducción

La integración de las tecnologías de información y las comunicaciones (TICs), ha contribuido a un mayor acceso a la información.

Hoy en día el acceso a la información se realiza por medio de la web. La naturaleza de la web es distribuida en relación a la información, la cantidad de la información va creciendo de manera progresiva día tras día sobre todo en la gestión de los documentos más específicamente en la publicación de edictos digitalizados. Esta situación ha generado la necesidad de crear un motor de búsqueda enfocado a la búsqueda de edictos digitalizados.

Los edictos son documentos que se publican en el periódico de manera cotidiana, ellos mismos representan una fuente de información importante que es manejada de una manera muy ineficiente dado que contienen datos importantes que no pueden ser encontrados directamente por el interesado dado su cantidad de contexto y el número de periódicos que hay en el país.

Dada esta problemática, este trabajo propone el desarrollo de un motor de búsqueda basada en bases de datos no relacionales aplicando Elasticsearch para la búsqueda de los edictos digitalizados proveyendo la información de manera más rápida y eficiente.

* 1. Antecedentes

La gestión de los documentos es un área muy importante dada la extensa cantidad de información que se maneja a través de ellos, información se va acumulando documentos desde que el hombre comenzó a escribirlos y a la necesidad de almacenarlos en bases de datos se vuelve imperiosa, esto teniendo en cuenta que los tipos y áreas de los documentos son variados una de esas áreas se encuentra en el aspecto legal más específicamente con la publicación de los edictos de prensa.

Edicto en el derecho moderno, es el mandato o decreto publicado con autoridad magistrado, juez o autoridad administrativa que dispone la observancia de ciertas reglas en algún asunto. También se determina así a los anuncios fijados en lugares públicos de las ciudades, villas o edificios gubernamentales sobre algún asunto para que sea notorio y de conocimiento general. ​En la práctica forense del derecho es un tipo de comunicación procesal. En nuestro caso nosotros trabajaremos con edictos de prensa digitalizados.

El crear un motor de búsqueda para realizar búsquedas en esa gran cantidad de información que contienen los edictos y las cuales son almacenadas en bases de datos y encontrar cualquier información de una manera más directa aprovechando la eficiencia de Elasticsearch combinado con Spring data para la implementación del motor de búsqueda.

* 1. Definición del problema

Los edictos son documentos publicados por un medio impreso un periódico de maneras distintas y con gran cantidad de información haciendo difícil encontrar la información de interés como ser los nombres, carnet de identidad, direcciones y otros, el interesado debe realizar una búsqueda manual y exhaustiva leyendo todo el contexto del edicto.

Se debe de leer una gran cantidad de información de manera exhaustiva dado la cantidad de edictos en un periódico y sin mencionar la calidad pésima del texto impreso que se tiene que se tiene que tratar de encontrar que varía de empresa a empresa además de tener que conseguir los periódicos de otros departamentos diferentes al que uno se encuentra haciendo bastante extensa teniendo que perder mucho tiempo. En el peor de los casos no logrando encontrar la información y teniendo que sufrir las consecuencias del mismo.

* 1. Objetivo
     1. Objetivo general

Implementar un motor de búsqueda de edictos digitalizados basado Elasticsearch para facilitar la búsqueda de la información.

* + 1. Objetivos específicos
* Diseñar el sistema la arquitectura del motor de búsqueda tanto en el lado del servidor y en el lado del cliente.
* Implementar un algoritmo de indexación de tablas relacionadas de base de datos para pasarla a colección de documentos en Elasticsearch.
* Implementar el algoritmo de búsqueda para búsqueda de texto completo y búsqueda avanzada.
* Implementar una interfaz de búsqueda por texto completo y búsqueda avanzada.
* Probar el rendimiento del buscador por texto y al buscador avanzado.
  1. Innovación tecnológica

Se desarrolló un motor de búsqueda de edictos almacenados en una base de datos no relacional en Elasticsearch el cual estará implementado enteramente en la parte del back end en lenguaje orientado a objetos Java haciendo uso de framework Spring Boot añadiendo la dependencia Spring Data para poder manejar los Documentos persistidos en Elasticsearch y generar un algoritmo de búsqueda tanto por texto como para búsquedas avanzadas, generando un api Restful, para que sea consumido por una aplicación cliente basado en un framework JavaScript llamado Angular.

Elasticsearch es un motor de búsqueda orientado a documentos que nos permite indexar un gran volumen de datos para poder hacer consultas sobre ellos posteriormente.

Entre otras cosas nos permite hacer búsquedas de texto completo, búsquedas aproximadas, facetas y de resultado.

Las ventajas tecnológicas que nos provee Elasticsearch son:

* **Acceso en tiempo real**: Elasticsearch nos permite acceder sobre los datos que se están modificando en tiempo real.
* **Escalabilidad**: Gracias a su diseño nos permite escalar de forma horizontal e ir escalando nuestros servidores según nuestras necesidades.
* **Alta disponibilidad**: Los clúster de Elasticsearch son capaces de detectar qué nodos están fallando y reorganizarse para hacer que los datos sean siempre accesibles.
* **Multi-Tenant:** Nos permite operar sobre distintos índices al mismo tiempo y así potenciar nuestras búsquedas.
* **No utiliza esquemas:**Permite trabajar sin una estructura fija de base de datos.
* **Orientado a documentos:** Las entidades de Elasticsearch se almacenan como archivos JSON estructurados donde todos los campos son indexados y podemos incluir todos los índices en una misma consulta.
* **API:** Elasticsearch nos proporciona API Restful en JSON junto con API para diferentes lenguajes.
* **Búsquedas basadas en texto:**Elasticsearch está basado en Lucene, lo que incrementa las capacidades de búsqueda de texto, soportando geo localización, autocompletado.
* **Gestión de conflictos:** Previene la pérdida de datos al editar simultáneamente los registros.

Spring Data es un proyecto de SpringSource cuyo propósito es unificar y facilitar el acceso a distintos tipos de tecnologías de persistencia, tanto a bases de datos relacionales como a las del tipo NoSQL.

Spring ya proporcionaba soporte para JDBC, Hibernate, JPA, JDO o MyIbatis, simplificando la implementación de la capa de acceso a datos, unificando la configuración y creando una jerarquía de excepciones común para todas ellas.

Y ahora, Spring Data viene a cubrir el soporte necesario para distintas tecnologías de bases de datos NoSQL y, además, integra las tecnologías de acceso a datos tradicionales, simplificando el trabajo a la hora de crear las implementaciones concretas.

Con cada tipo de tecnología de persistencia los DAOs (Data Access Objects) ofrecen las funcionalidades típicas de un CRUD (Create-Read-Update-Delete ) para objetos de dominio propios, métodos de búsqueda, ordenación y paginación. Spring Data proporciona interfaces genéricas para estos aspectos (CrudRepository, PagingAndSortingRepository) e implementaciones específicas para cada tipo de tecnología de persistencia.

A día de hoy, Spring Data proporciona soporte para las siguientes tecnologías de persistencia:

* JPA y JDBC
* Apache Hadoop
* GemFire
* Redis
* MongoDB
* Neo4j
* HBase
  1. Justificación

Justificación técnica porque no hay un motor de búsqueda que este enfocados a la búsqueda de edictos, además del aprovechamiento de Elasticsearch que es una de las mejores tecnologías para realizar búsquedas.

Justificación económica porque al usuario que haga uso de la aplicación tendrá beneficios en el proceso de búsqueda de edictos de su interés minimizando el tiempo y logrando responder con los edictos que coinciden con la búsqueda.

Justificación social el proyecto puede tener una orientación de apoyo en la sociedad Boliviana, mejorando la calidad de las personas dando más tiempo a las personas que tengan la necesidad de buscar edictos en algún periódico a nivel nacional.

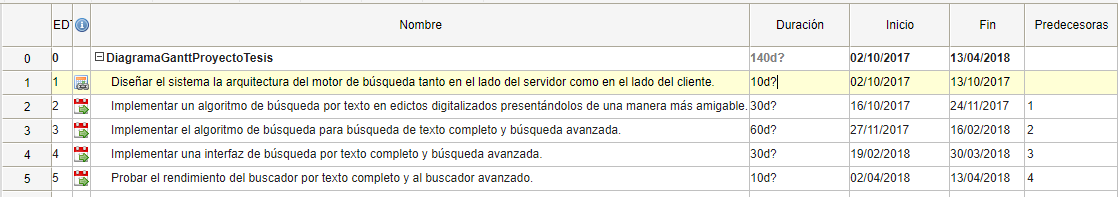
* 1. Metodología de desarrollo

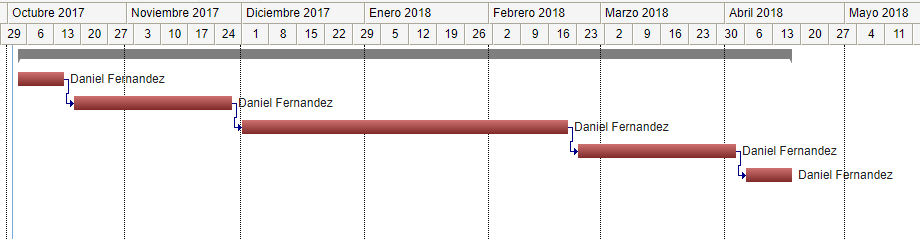
El proyecto se desarrollará de manera incremental, en la cual cada módulo o subsistema se tomará como un incremento de la aplicación. Cada incremento seguirá este proceso, la primera es realizar la implementación de un algoritmo de indexación para poder llevar la información de la base de datos relacional a la base de datos de Elasticsearch, la segunda etapa la implementación de un algoritmo de búsqueda por texto, y después pasaría a la etapa de implementación de un algoritmo de búsqueda avanzada para realizar búsquedas más específicas, por último en la quinta etapa se realizara la aplicación cliente para realizar las búsquedas por el usuario y mostrar la información de los resultados de las búsquedas.

* + 1. Matriz de objetivos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Objetivos | Actividades a Realizar | Resultados | Nro. Días Provistos | Fecha de Inicio | Fecha de Conclusión |
| Diseñar el sistema la arquitectura del motor de búsqueda tanto en el lado del servidor y en el lado del cliente. | * Creación de nuevo proyecto en Spring Boot. * Instalación de dependencias (Spring Data, Elasticsearch). * Configuración de propiedades para conexión con Elasticsearch * Creación de modulo cliente en Angular | * Proyecto creado en Spring Boot, con dependencias instaladas y configuración para la conexión a Elasticsearch | 10 | 02/10/2017 | 13/10/2017 |
| Implementar un algoritmo de indexación de tablas relacionadas de base de datos para pasarla a colección de documentos en Elasticsearch.  . | * Creación de algoritmo de migración de datos desde una base de datos relacional * Creación de la base de datos a través de los objetos persistentes Spring Data y JPA | * Base de datos no relacional en Elasticsearch * Datos migrados a la base de datos | 30 | 16/10/2017 | 24/11/2017 |
| Implementar el algoritmo de búsqueda para búsqueda de texto completo y búsqueda avanzada. | * Diseño de algoritmo de búsqueda. * Implementación del algoritmo en Spring Boot y Spring Data para búsquedas en Elasticsearch. * Diseñar el algoritmo de búsqueda con campos establecidos. * Implementación de algoritmo en Java Spring Boot tomando como criterios obligatorios los parámetros. | * Implementación de algoritmo para búsqueda avanzada con criterios | 60 | 25/11/2017 | 16/02/2018 |
| Implementar una interfaz de búsqueda por texto completo y búsqueda avanzada. | * Implementar la interfaz de búsqueda por texto completo. * Implementar la interfaz de búsqueda avanzada | * Interfaz para realizar las búsquedas texto completo y búsquedas avanzadas | 30 | 19/02/2018 | 30/02/2018 |
| Probar el rendimiento del buscador por texto completo y al buscador avanzado. | * Implementación de pruebas de rendimiento para medir la mejora en las búsquedas |  | 10 | 02/04/2018 | 13/04/2018 |

* + 1. Diagrama de Gantt





* 1. Alcance
* Se contempla únicamente el diseño la implementación del algoritmo de búsqueda.
* Para para la implementación se hará uso de Elasticsearch para realizar almacenar los datos.
* El sistema no contempla la gestión de los usuarios ni todo lo que respecta a los usuarios.
* El sistema no contempla la digitalización ni el registro de los edictos.
  1. Bibliografía
* Phillip Webb, Dave Syer, Josh Long, Stéphane Nicoll, Rob Winch, Andy Wilkinson, Marcel Overdijk, Christian Dupuis, Sébastien Deleuze, Michael Simons. (2012-2017). Documentacion spring boot. 25-09-2017, de Documentacion spring boot Sitio web: <https://docs.spring.io/spring-boot/docs/current/reference/html/>
* Pivotal Software. (15-10-2015). spring boot. 25-09-2017, de Spring boot Sitio web: <https://projects.spring.io/spring-boot/>
* Gerard Auladell. (28-11-2016). ¿Qué es Elasticsearch?. 25-09-2017, de Elasticsearch Sitio web: <https://www.drauta.com/que-es-elasticsearch>
* Elasticsearch. (11-09-2017). Elasticsearch documentacion. 25-09-2017, de Elasticsearch Documentacion Sitio web: <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>
* Elasticsearch. (2011). ElasticSearch documentacion. Julio 2011, de ElasticSearch Sitio web: [https://www.elastic.co](https://www.elastic.co/)
* Elasticsearch. (2016). A Distributed RESTful Search Engine. Julio 2016, de ElasticSearch Sitio web: <https://github.com/elastic/elasticsearch>
* ElasticSearch. (2015). Spring Data Elasticsearch. Junio 2015, de Elasticsearch Sitio web: <https://docs.spring.io/spring-data/elasticsearch/docs/current/reference/html/>
* ElasticSearch Spring Data. (2015). Spring Data Elasticsearch. Junio 2015, de ElasticSearch Sitio web: <https://github.com/spring-projects/spring-data-elasticsearch>
* Hector Estrada. (2015). Clases de Notificacion. Octubre, 2015, de Clases de Notificacion Sitio web: <http://tareasjuridicas.com/2015/10/05/clases-notificacion/>
* Vinneth Mohan. (2015). Building Your Own E-Commerce Solution. En Elasticsearch Blueprints(40). Livery Place: Pack Publishing.