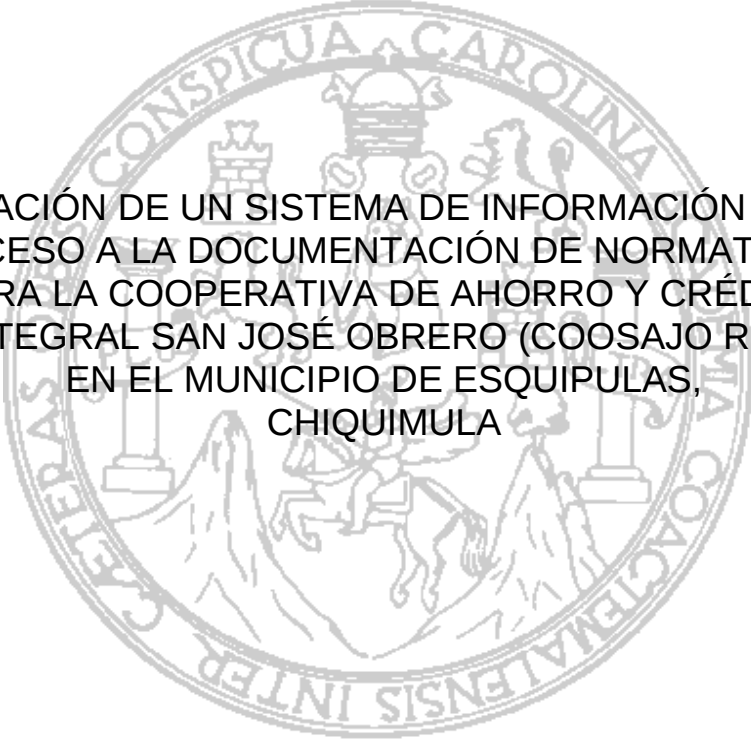


UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE ORIENTE
INGENIERÍA EN CIENCIAS Y SISTEMAS



CREACIÓN DE UN SISTEMA DE INFORMACIÓN PARA
ACCESO A LA DOCUMENTACIÓN DE NORMATIVAS
PARA LA COOPERATIVA DE AHORRO Y CRÉDITO
INTEGRAL SAN JOSÉ OBRERO (COOSAJO R.L.),
EN EL MUNICIPIO DE ESQUIPULAS,
CHIQUMULA

JOSÉ EDUARDO GUERRA SOSA

CHIQUMULA, GUATEMALA, SEPTIEMBRE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE ORIENTE
INGENIERÍA EN CIENCIAS Y SISTEMAS

CREACIÓN DE UN SISTEMA DE INFORMACIÓN PARA
ACCESO A LA DOCUMENTACIÓN DE NORMATIVAS
PARA LA COOPERATIVA DE AHORRO Y CRÉDITO
INTEGRAL SAN JOSÉ OBRERO (COOSAJÓ R.L.),
EN EL MUNICIPIO DE ESQUIPULAS,
CHIQUMULA

TRABAJO DE GRADUACIÓN

Sometido a consideración del Honorable Consejo Directivo

Por

JOSÉ EDUARDO GUERRA SOSA

Al conferírsele el título de

INGENIERO EN CIENCIAS Y SISTEMAS

En el grado académico de

LICENCIADO

CHIQUMULA, GUATEMALA, SEPTIEMBRE 2021

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE ORIENTE
INGENIERÍA EN CIENCIAS Y SISTEMAS**



**RECTOR EN FUNCIONES
M.A. PABLO ERNESTO OLIVA SOTO**

CONSEJO DIRECTIVO

Presidente:	Ing. Agr. Edwin Filiberto Coy Cordón
Representante de Profesores:	M.Sc. Mario Roberto Díaz Moscoso
Representante de Profesores:	M.Sc. Gildardo Guadalupe Arriola Mairén
Representante de Graduados:	Ing. Agr. Henry Estuardo Velásquez Guzmán
Representante de Estudiantes:	A.T. Zoila Lucrecia Argueta Ramos
Representante de Estudiantes:	Br. Juan Carlos Lemus López
Secretaria:	M.Sc. Marjorie Azucena González Cardona

AUTORIDADES ACADÉMICAS

Coordinador Académico:	M.A. Edwin Rolando Rivera Roque
Coordinador de Carrera:	M. Sc. Rolando Darío Chávez Valverth

ORGANISMO COORDINADOR DE TRABAJOS DE GRADUACIÓN

Presidente:	M. Sc. Rolando Darío Chávez Valverth
Secretario:	MBA. René Estuardo Alvarado González
Vocal:	Ing. Elder Avildo Rivera López

TERNA EVALUADORA

Presidente:	MBA. René Estuardo Alvarado González
Secretario:	MBA. Samy Eunice Pinto Castañeda
Vocal:	Ing. Carlos David Ardón Muñoz

Chiquimula, septiembre de 2021

Señores

Miembros del Consejo Directivo

Centro Universitario de Oriente

Universidad de San Carlos de Guatemala

Respetables Señores:

De conformidad con los estatutos establecidos por la ley de la Universidad de San Carlos de Guatemala y del Centro Universitario de Oriente, presento a su consideración mi trabajo de graduación titulado **“CREACIÓN DE UN SISTEMA DE INFORMACIÓN PARA ACCESO A LA DOCUMENTACIÓN DE NORMATIVAS PARA LA COOPERATIVA DE AHORRO Y CRÉDITO INTEGRAL SAN JOSÉ OBRERO (COOSAJO R.L.), EN EL MUNICIPIO DE ESQUIPULAS, CHIQUIMULA”**, como requisito previo a optar al título de Ingeniero en Ciencias y Sistemas, con el grado académico de licenciado.

Atentamente,

“ID Y ENSEÑAD A TODOS”

F. 

José Eduardo Guerra Sosa



**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE ORIENTE -CUNORI-
CARRERAS DE INGENIERÍA**



SEPC.01.20

Chiquimula, 16 de noviembre de 2020.

MBA. René Estuardo Alvarado González
Coordinador del Ejercicio Profesional Supervisado
De la Carrera de Ingeniería en Ciencias y Sistemas
Centro Universitario de Oriente
Universidad de San Carlos de Guatemala

Respetable Ingeniero:

Atentamente me dirijo a usted para hacer de su conocimiento que he revisado y efectuado las correcciones al trabajo de graduación titulado: **"CREACIÓN DE UN SISTEMA DE INFORMACIÓN PARA ACCESO A LA DOCUMENTACIÓN DE NORMATIVAS PARA LA COOPERATIVA DE AHORRO Y CRÉDITO INTEGRAL SAN JOSÉ OBRERO (COOSAJÓ R.L.), EN EL MUNICIPIO DE ESQUIPULAS, CHIQUIMULA"**, presentado por el estudiante José Eduardo Guerra Sosa.

Habiendo llenado dicho trabajo los requisitos, me permito aprobarlo en calidad de Asesor - Revisor del mismo.

"ID Y ENSEÑAR A TODOS"


MBA. Samy Eusebio Pinto Castañeda
Asesor - Revisor

c.c. archivo



CEPSICYS.02.2021

Chiquimula, 14 de enero de 2021.

Maestro en Ciencias

Rolando Darío Chávez Valverth

Coordinador Carreras de Ingenierías

CUNORI – USAC

Respetado M.Sc. Chávez:

Atentamente me dirijo a usted para informarle que he revisado el trabajo de graduación titulado: **“CREACIÓN DE UN SISTEMA DE INFORMACIÓN PARA ACCESO A LA DOCUMENTACIÓN DE NORMATIVAS PARA LA COOPERATIVA DE AHORRO Y CREDITO INTEGRAL SAN JOSÉ OBRERO (COOSAJÓ, R.L.) EN EL MUNICIPIO DE ESQUIPULAS, CHIQUIMULA”**, elaborado por el estudiante José Eduardo Guerra Sosa, quien contó con la asesoría de la MBA. Samy Eunice Pinto Castañeda.

Considero que el trabajo desarrollado por el estudiante José Eduardo Guerra Sosa,, satisface los requisitos exigidos, por lo cual recomiendo su aprobación.

Agradezco a usted la atención a la presente, atentamente,

“ID Y ENSEÑAD A TODOS”

MBA. René Estuardo Avarado González
Coordinador del Ejercicio Profesional Supervisado (E.P.S)
Carrera de Ingeniería en Ciencias y Sistemas

c.c. archivo



ALTG.3.21

Chiquimula, 6 de abril de 2021.

Maestro en Ciencias

Rolando Darío Chávez Valverth

Coordinador Carreras de Ingenierías

CUNORI-USAC

Respetado M.Sc. Chávez Valverth:

El propósito de la presente es para informarle que he procedido a revisar la parte lingüística, del trabajo de graduación titulado: **“CREACIÓN DE UN SISTEMA DE INFORMACIÓN PARA ACCESO A LA DOCUMENTACIÓN DE NORMATIVAS PARA LA COOPERATIVA DE AHORRO Y CRÉDITO INTEGRAL SAN JOSÉ OBRERO (COOSAJO R.L.), EN EL MUNICIPIO DE ESQUIPULAS, CHIQUIMULA”**; elaborado por el estudiante **José Eduardo Guerra Sosa**.

El informe cumple con los requisitos exigidos por la carrera de Ingeniería en Ciencias y Sistemas, por lo tanto, recomiendo su aprobación para seguir con los trámites correspondientes.

Agradeciendo su atención a la presente,

Deferentemente,

ID Y ENSEÑAD A TODOS

M.Sc. Julio César Hernández Ortiz
Revisor Área Lingüística

c.c. archivo



TGICYS.02-2021

Chiquimula, 6 de mayo de 2021.

Ingeniero Agrónomo

Edwin Filiberto Coy Cordón

Director

CUNORI - USAC

Respetable Ing. Agr. Coy Cordón:

El coordinador de las Carreras de Ingeniería del Centro Universitario de Oriente CUNORI de la Universidad de San Carlos de Guatemala, después de conocer el dictamen de la MBA. Samy Eunice Pinto Castañeda en su calidad de asesor- revisor y del coordinador del Ejercicio Profesional Supervisado de la carrera de Ingeniería en Ciencias y Sistemas el MBA. René Estuardo Alvarado González y luego de la revisión y aprobación del Maestro en Ciencias Julio César Hernández Ortiz, revisor del área de lingüística, al trabajo de graduación del estudiante **José Eduardo Guerra Sosa**, titulado: **“CREACIÓN DE UN SISTEMA DE INFORMACIÓN PARA ACCESO A LA DOCUMENTACIÓN DE NORMATIVAS PARA LA COOPERATIVA DE AHORRO Y CRÉDITO INTEGRAL SAN JOSÉ OBRERO (COOSAJÓ R.L.), EN EL MUNICIPIO DE ESQUIPULAS, CHIQUIMULA”**, da por este medio su aprobación a dicho trabajo de graduación y recomendando la autorización del mismo.

“ID Y ENSEÑAD A TODOS”



M.Sc. Rolando Darío Chávez Valverde
Coordinador Carreras de Ingeniería
CUNORI - USAC



c.c. archivo

D-TG-ICS-133/2021

EL INFRASCRITO DIRECTOR DEL CENTRO UNIVERSITARIO DE ORIENTE DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, POR ESTE MEDIO HACE CONSTAR QUE: Conoció el Trabajo de Graduación que efectuó el estudiante **JOSÉ EDUARDO GUERRA SOSA** titulado “**CREACIÓN DE UN SISTEMA DE INFORMACIÓN PARA ACCESO A LA DOCUMENTACIÓN DE NORMATIVAS PARA LA COOPERATIVA DE AHORRO Y CRÉDITO INTEGRAL SAN JOSÉ OBRERO (COOSAJO R.L.), EN EL MUNICIPIO DE ESQUIPULAS, CHIQUIMULA**”, trabajo que cuenta con el aval del Revisor y Coordinador de Trabajos de Graduación, de la carrera de Ingeniería en Ciencias y Sistemas. Por tanto, la Dirección del CUNORI con base a las facultades que le otorga las Normas y Reglamentos de Legislación Universitaria **AUTORIZA** que el documento sea publicado como **Trabajo de Graduación** a Nivel de Licenciatura, previo a obtener el título de **INGENIERO EN CIENCIAS Y SISTEMAS**.

Se extiende la presente en la ciudad de Chiquimula, el ocho de septiembre del dos mil veintiuno.

“ID Y ENSEÑAD A TODOS”

Ing. Agr. Edwin Filiberto Coy Cordón

**DIRECTOR
CUNORI – USAC**



ACTO QUE DEDICO

A Dios:

Quien con su bendición llena siempre mi vida, y la de mi familia. Por otorgarme la sabiduría en todo momento.

A mis padres:

Marvin Joaquín Guerra Mejía y Evelin Karina Sosa Morales, por brindarme amor verdadero, por el trabajo y sacrificio, por haber forjado la persona que soy actualmente, por sus consejos y guía, por inspirarme a ser una mejor persona.

A mis hermanas:

Jennifer Celeste Guerra Sosa y Jaretzi Karina Guerra Sosa, por siempre estar en los momentos más importantes de mi vida.

A mi familia:

Por su amor, ser la base de mi formación, por motivarme en cada proyecto que realizo.

AGRADECIMIENTOS

Al Centro Universitario de Oriente: Por ser la casa de estudios que me formó y educó profesionalmente.

A la carrera de Ingeniería en Ciencias y Sistemas: Por brindarme una gran educación a nivel profesional.

A mi asesora: Samy Eunice Pinto Castañeda, por su guía, sus consejos y apoyo en todo momento.

A la Cooperativa de Ahorro y Crédito Integral San José Obrero R.L.: Por abrirme las puertas y brindarme todo lo necesario para poder realizar mi EPS.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	I
LISTA DE SÍMBOLOS	V
GLOSARIO	VII
RESUMEN	XI
INTRODUCCIÓN	XIII
OBJETIVOS	XV
1. FASE DE INVESTIGACIÓN	1
1.1 Antecedentes de la organización	1
1.1.1 Descripción de la organización	1
1.1.2 Estructura organizativa	1
1.1.3 Misión	3
1.1.4 Visión	4
1.1.5 Descripción del trabajo de la organización	4
1.1.6 Productos o servicios que presta	4
1.1.7 Principales necesidades que plantea la organización	5
1.2 Análisis de los procesos actuales del negocio	5
1.3 Antecedentes del problema	6
1.4 Planteamiento y descripción del problema	6
1.5 Delimitación del problema	7
1.5.1 Delimitación geográfica	7
1.5.2 Delimitación institucional	8

1.5.3	Delimitación personal	9
1.5.4	Delimitación temporal	9
1.5.5	Delimitación teórica	9
1.6	Requerimientos de tecnología de información necesarios para la implementación del proyecto	9
1.7	Justificación	10
2.	FASE DE TRABAJO TÉCNICO PROFESIONAL	11
2.1	Descripción del proyecto	11
2.2	Beneficios del proyecto	13
2.3	Fundamentación teórica	13
2.3.1	Motor de búsqueda	14
2.3.2	Índice invertido	14
2.3.3	Analizadores de texto	15
2.3.4	Arquitectura Cliente Servidor	18
2.3.5	Servicios Web y APIs	19
2.3.6	Kanban	20
2.3.7	Proceso Unificado Racional (RUP)	21
2.3.8	Tecnologías de desarrollo	23
2.4	Alcance y productos esperados	25
2.5	Plan de trabajo	26
2.6	Modelado de negocio	26
2.6.1	Modelado de procesos de negocio	26
2.7	Requerimientos	29
2.7.1	Requerimientos funcionales	29

2.7.2	Requerimientos no funcionales	30
2.8	Análisis y diseño	31
2.8.1	Casos de uso	31
2.8.2	Narración de los casos de uso	32
2.8.3	Puntos de conexión del sistema	44
2.8.4	Diagrama de base de datos (ElasticSearch)	50
2.8.5	Procesos	55
2.9	Implementación	61
2.9.1	Prototipos	61
2.9.2	Diagrama de componentes	65
2.10	Pruebas	66
2.10.1	Pruebas del buscador	66
2.10.2	Pruebas del backend (API REST)	69
2.11	Entrega	73
2.12	Lista de riesgos	75
2.13	Gestión de cambios	78
2.14	Administración del proyecto	81
2.14.1	Cronograma de actividades inicial	81
2.14.2	Tablero Kanban	84
2.15	Ambiente	86
2.15.1	Tecnologías de desarrollo	86
2.15.2	Herramientas de apoyo	87
2.15.3	Ambiente de desarrollo	88
2.15.4	Ambiente de producción	88
2.16	Valoración económica del proyecto	89

3.	FASE DE ENSEÑANZA APRENDIZAJE	93
3.1	Capacitación para uso de software	93
3.1.1	Justificación	93
3.1.2	Objetivo	93
3.1.3	Personal objetivo	94
3.1.4	Documentación	94
3.1.5	Estructura del plan de capacitación	95
	CONCLUSIONES	97
	RECOMENDACIONES	99
	REFERENCIAS BIBLIOGRÁFICAS	101
	APÉNDICES	105
	ANEXOS	155

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Organigrama COOSAJO	3
2. Proceso para publicar normativo	5
3. Ubicación Oficinas de COOSAJO	8
4. Ejemplo del funcionamiento de un índice invertido	14
5. Ejemplo de la fase lowercase del analizador de texto	15
6. Ejemplo de la fase charfilter del analizador de texto	16
7. Ejemplo del tokenizador del analizador de texto	16
8. Ejemplo de la fase stopwords del analizador de texto	17
9. Ejemplo de la fase stemmer del analizador de texto	18
10. Kanban	20
11. Diagrama RUP	22
12. Modelado de proceso de negocio realizar búsqueda	27
13. Modelado de proceso de negocio eliminar normativo	28
14. Casos de uso de la aplicación	31
15. Modelo índice metadata	51
16. Modelo índice main	53
17. Modelo índice stats	54
18. Diagrama de secuencias agregar documento	55
19. Diagrama de actividades agregar documento (1)	56
20. Diagrama de actividades agregar documento (2)	57
21. Diagrama de secuencia eliminar documento	58
22. Diagrama de actividades eliminar documento	59
23. Diagrama de secuencias realizar búsqueda	60
24. Diagrama de actividades abrir documento	61
25. Mockup buscador	62

26. Mockup repositorio de normativas	63
27. Prototipo de buscador	64
28. Prototipo repositorio de normativas	64
29. Diagrama de componentes	65
30. Diagrama de despliegue de la aplicación	74
31. Cronograma de actividades inicial 1	81
32. Cronograma de actividades inicial 2	81
33. Cronograma de actividades inicial 4.	82
34. Cronograma de actividades inicial 3	82
35. Cronograma de actividades inicial 5	83
36. Cronograma de actividades inicial 6	83
37. Tablero Kanban Inicial	84
38. Tablero Kanban final	84

TABLAS

1. Narración caso de uso "ver estadísticas de uso"	32
2. Narración caso de uso "ver cantidad de documentos ingresados"	33
3. Narración caso de uso "ver cantidad de documentos vistos"	34
4. Narración caso de uso "ver documentos más vistos"	35
5. Narración caso de uso "gestionar normativos"	36
6. Narración caso de uso "agregar nuevo normativo"	37
7. Narración caso de uso "agregar nueva versión a un normativo"	38
8. Narración caso de uso "eliminar normativo"	39
9. Narración casos de uso "ver estadísticas del documento"	40
10. Narración caso de uso "realizar búsquedas sobre los normativos"	41
11. Narración caso de uso "ver documento del normativo"	42
12. Narración caso de uso "ver todos los normativos"	43

13. Punto de conexión obtener documentos	44
14. Punto de conexión agregar documento	45
15. Punto de conexión obtener documento	45
16. Punto de conexión eliminar documento	46
17. Punto de conexión obtener pdf	46
18. Punto de conexión documento puede ser eliminado	47
19. Punto de conexión obtener versiones del documento	47
20. Punto de conexión realizar búsqueda	48
21. Punto de conexión vistas totales	49
22. Punto de conexión documentos más vistos	49
23. Punto de conexión vistas totales	49
24. Punto de conexión estadísticas de un documento	50
25. Descripción índice metadata	52
26. Descripción índice main	53
27. Descripción índice stats	54
28. Prueba 1 del buscador	66
29. Prueba 2 del buscador	67
30. Prueba 3 del buscador	68
31. Prueba 1 para API REST	69
32. Prueba 2 API REST	70
33. Prueba 3 para API REST	71
34. Prueba 4 para API REST	72
35. Riesgo de acciones no autorizadas	75
36. Riesgo acceder a Elasticsearch sin autorización	76
37. Riesgo de guardar más de diez mil documentos	76
38. Riesgo de error a la mitad de una transacción de guardado	77
39. Gestión de cambios 01	78
40. Gestión de cambios 02	78
41. Gestión de cambios 03	79

42. Gestión de cambios 04	79
43. Gestión de cambios 05	80
44. Gestión de cambios 06	80
45. Tarjetas Kanban utilizadas	85
46. Roles involucrados en la creación del proyecto	89
47. Actividades realizadas arquitecto de software	90
48. Actividades realizadas desarrollador senior	90
49. Actividades realizadas desarrollador ElasticSearch	91
50. Actividades realizadas consultor de sistemas	91
51. Actividades realizadas project manager	91
52. Costo del recurso humano	92
53. Costo total del proyecto	92
54. Estructura del plan de capacitación técnica	95

LISTA DE SÍMBOLOS

Símbolo	Significado
API	<i>Application Programming Interface</i>
IDT	Investigación y Desarrollo Tecnológico
ms	Milisegundos
min	Minutos
Q	Quetzal
REST	<i>Representational State Transfer</i>
RUP	<i>Rational Unified Process</i>
s	Segundos

GLOSARIO

Asociado	Persona que puede utilizar los productos y servicios que brinda la cooperativa San José Obrero R.L.
Backend	Capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios. Además, contiene la lógica de la aplicación que maneja dichos datos.
Colaborador	Todo personal que trabaja en la cooperativa es un colaborador, todo colaborador también es un asociado.
Cooperativa	Se usa para referirse a la institución COOSAJÓ R.L.
Dashboard	Panel de control.
Documento	Archivo que contiene información que es útil para los usuarios.
Frontend	Es la parte de un programa o dispositivo a la que un usuario puede acceder directamente.

Full Text Search	Buscar texto sobre un documento de texto completo.
Indexación	Forma en la que se obtiene y se clasifica información, para luego ser almacenada en un índice.
Interfaz de programación de aplicaciones	Referida como <i>API</i> (del inglés <i>API: Application Programming Interface</i>). Es un conjunto de rutinas que provee acceso a funciones de un determinado software.
Investigación y desarrollo tecnológico	Departamento de trabajo dentro de la cooperativa, se utilizan las siglas IDT para referirse a este.
Metodología de desarrollo	Proceso que se suele seguir a la hora de diseñar una solución o un programa específico.
Mockup	Prototipo de la interfaz de usuario.
Modal	Ventana de la interfaz de usuario, que aparece sobre la ventana principal o programa.
Open Source	Software de código abierto, que está disponible para el público en general.

Parseo	Análisis de una secuencia de caracteres o símbolos.
Servidor	Un servidor es una aplicación en ejecución capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia.
Software	Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.
Transacciones	Conjunto de instrucciones a ejecutar en una base de datos, por ejemplo, ingresar o editar información para poder obtenerla posteriormente.
UX	<i>User Experience</i> es la experiencia que el usuario tiene al interactuar con el sistema, una interfaz con una buena UX debe permitir realizar tareas a los usuarios con una gran facilidad y rapidez.
Volumen de información	Cantidad de información.

RESUMEN

Se implementó un proyecto tecnológico en la Cooperativa de Ahorro y Crédito Integral San José Obrero, con el fin de brindar un servicio de mejor calidad y poder servir de apoyo para los colaboradores de la cooperativa. Este proyecto es un sistema de información que permite realizar búsquedas de texto completo sobre las normativas que rigen a COOSAJO R.L.

Esta es una búsqueda de texto avanzada que permite encontrar al instante un normativo específico dentro de una gran cantidad de normativos. Para almacenar este volumen de información se implementó un repositorio.

En este repositorio se almacenan todos los normativos, además de eso tiene soporte para versionado de estos. También se creó una sección de estadísticas, la cual permite ver cuáles son los documentos más visualizados y el uso de la plataforma. En conjunto, a estos sistemas se implementó un programa que permite la creación rápida de proyectos de búsqueda que utilicen las mismas tecnologías que fueron utilizadas en este proyecto.

Para poder realizar este proyecto se utilizaron tecnologías modernas como Elasticsearch, NodeJS, y PHP. Además, se utilizaron otras tecnologías auxiliares como JavaScript, HTML y CSS.

Como metodología de trabajo se utilizó el Proceso Unificado Racional auxiliado de la metodología Kanban.

INTRODUCCIÓN

Con el avance del hardware y el software en los últimos años se ha creado una revolución tecnológica que permite a las instituciones dar mejores servicios a sus clientes, y también facilitar herramientas que logren aumentar la productividad del trabajo que realiza su personal, haciendo que este sea más rápido y preciso.

Estas nuevas tecnologías permiten manejar grandes volúmenes de información, pudiendo agregar, modificar y encontrar información en cuestión de segundos. El acceso a todos estos datos permite realizar análisis que logran mejorar la toma de decisiones.

Todas las instituciones cuentan con una gran cantidad de normativos que permiten llevar un mejor orden, control y un flujo de trabajo organizado. Saber cuáles son estos documentos es esencial para cualquier colaborador, ya que toda decisión que deba de tomar estará regida por las reglas que se encuentran dentro de ellos. Muchas veces, para tomar acciones ante determinadas situaciones, es necesario tener este conocimiento de manera puntual y precisa. Para ayudar con esa necesidad se puede hacer uso de nuevas tecnologías que admiten guardar y obtener esta cantidad de información rápidamente.

La fase de investigación manifiesta información acerca de la institución en la cual se desarrolló el proyecto, mostrando las necesidades de esta y cuál es el problema planteado.

En la fase de trabajo técnico profesional se concreta la fundamentación teórica del proyecto y la planificación que apoyará el desarrollo de este.

La fase de enseñanza y aprendizaje muestra información sobre las capacitaciones y el material de apoyo para el uso de la aplicación.

OBJETIVOS

General

Crear un sistema de información que permita el filtrado y búsqueda de normativas para la Cooperativa de Ahorro y Crédito Integral San José Obrero (COOSAJÓ R.L.).

Específicos

1. Implementar una interfaz de programación de aplicaciones que permita la indexación y consulta de información.
2. Aplicar tecnologías de información para desarrollar un motor de búsqueda en los documentos que conforman las normativas que rigen los procesos de COOSAJÓ R.L.
3. Capacitar a los colaboradores de COOSAJÓ, R.L. para la consulta de normativas y para el uso de la interfaz de programación de aplicaciones generada.

1. FASE DE INVESTIGACIÓN

1.1 Antecedentes de la organización

1.1.1 Descripción de la organización

La Cooperativa de Ahorro y Crédito Integral San José Obrero COOSAJO R.L. es una institución trascendente que genera riqueza por medio del apoyo al acceso a bienes, servicios y valores a través de productos y buenas prácticas financieras y cooperativas, logrando que hombres y mujeres vivan digna y felizmente, dentro de una sociedad sostenible.

La Cooperativa de Ahorro y Crédito ve en los Asociados el potencial de desarrollo del país, concientizándose en sus necesidades y apoyando mediante el fomento de la salud y el desarrollo personal de cada uno de los habitantes de la región (Coosajo, 2018).

1.1.2 Estructura organizativa

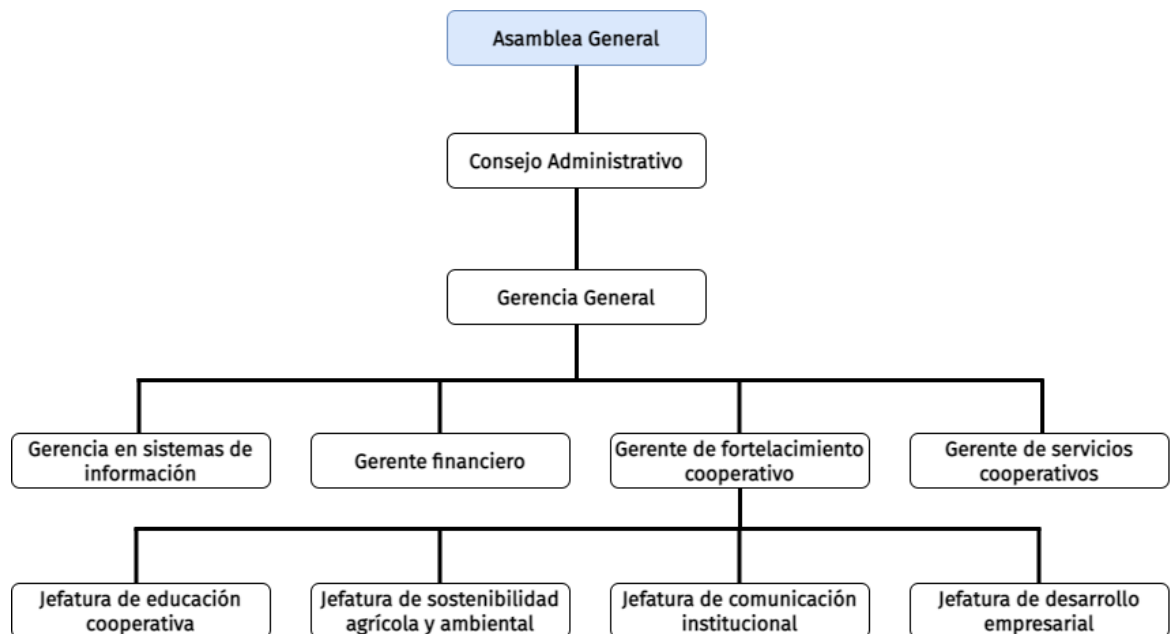
La máxima autoridad de la cooperativa es la Asamblea General conformada en su totalidad por todos los asociados del área de influencia de COOSAJO R.L.; es la encargada de tomar las decisiones para seleccionar que proyectos se realizarán. Así mismo, tiene la función de elegir a los funcionarios del Consejo Administrativo. En el mismo nivel se

encuentra la Comisión de Vigilancia, la que está integrada por tres miembros electos por el Consejo Administrativo, y su fin es realizar auditorías constantes en la toma de decisiones del consejo y de los créditos que se adjudican.

En el siguiente escalón está la Gerencia General que se subdivide en la Gerencia de Negocios, Gerencia Financiera, Gerencia de Fortalecimiento Cooperativo y Gerencia de Riesgos.

La Gerencia de Fortalecimiento Cooperativo se subdivide en cuatro jefaturas, estas son: Jefatura de Educación Cooperativa, encargada de desarrollar capacitaciones relacionadas al cooperativismo y de otras índoles similares; Jefatura de Sostenibilidad Agrícola y Ambiental, la que está encargada de prestar servicios respecto a temas de mejoramiento en la producción agrícolas y protección del ambiente y está compuesta por doce técnicos, de los cuales ocho son en campo y cuatro en asistencia técnica. También se cuenta con un encargado administrativo de la jefatura. La Jefatura de Comunicación Institucional es el ente encargado de las relaciones públicas, la cual está integrada por cinco técnicos para desempeñar las labores. Jefatura de Desarrollo Empresarial que capacita a empresarios emprendedores y está compuesta por cuatro técnicos (Coosajo, 2018).

Figura 1. **Organigrama COOSAJO**



Fuente: elaboración propia.

1.1.3 Misión

“Brindar soluciones financieras innovadoras, a través de un servicio profesional, personalizado, que permite brindar a sus asociados beneficios integrales” (Coosajo, 2018).

1.1.4 Visión

Ser una institución trascendente que genera riqueza por medio del apoyo al acceso a bienes, servicios u valores a través de productos y buenas prácticas financieras y cooperativas, logrando que hombres y mujeres vivan digna y felizmente, dentro de una sociedad sostenible y exitosa (Coosajo, 2018.).

1.1.5 Descripción del trabajo de la organización

Los trabajadores del parque Chatún en conjunto con los trabajadores de COOSAJO R.L., llevan a cabo programas, talleres, charlas, conferencias y asambleas para crear valores de cooperativismo, apoyar y fortalecer las capacidades económicas, productivas y compartir los beneficios que obtienen al ser asociados. Para estas actividades se toman en cuenta principalmente a las personas que son asociadas, aunque también se hacen invitaciones para organizaciones comunitarias no asociadas o personas particulares que pueden participar y beneficiarse de los programas.

1.1.6 Productos o servicios que presta

COOSAJO R.L. ofrece servicios de ahorro y crédito, seguros, remesas, acompañamiento empresarial, agrícola, ambiental y de educación cooperativa.

La institución también se ve encargada de realizar un proyecto de excedentes cada año que va dirigido a todos los asociados de la misma como una muestra de agradecimiento por la confianza depositada en ellos.

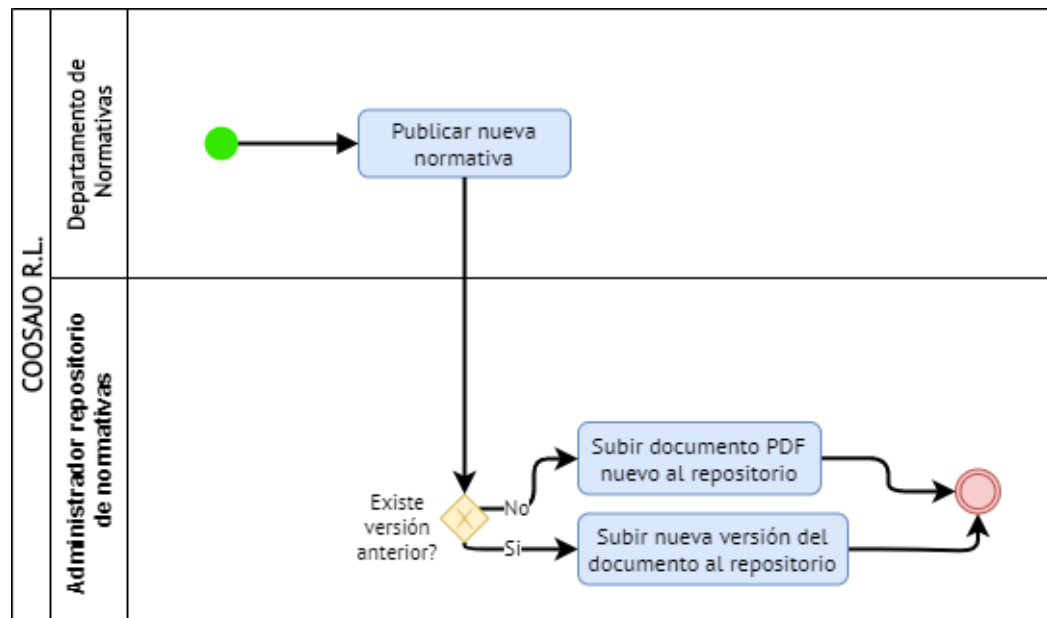
1.1.7 Principales necesidades que plantea la organización

Prestar un mejor servicio a los colaboradores y asociados mediante la implementación de nuevas tecnologías.

1.2 Análisis de los procesos actuales del negocio

Cuando un nuevo normativo es aprobado y publicado, este documento es subido a un repositorio para que todos los colaboradores tengan acceso a este.

Figura 2. Proceso para publicar normativo



Fuente: elaboración propia.

1.3 Antecedentes del problema

COOSAJO R.L. al principio no contaba con un servidor donde se almacenarían las normativas, así que decidieron crear una carpeta compartida en la cual se subían los documentos y podían ser visualizadas por los colaboradores.

Luego de la creación de esta carpeta compartida, en el año 2012, se empezó con la creación de un repositorio de normativas la cual las mostraba en una tabla, que no contaba con buscador o filtrado, y solo las separaba en categorías como normas internas y externas. Además de esto tenía un sistema versionado de documentos y a los colaboradores únicamente se les mostraba la versión más reciente. Este repositorio era un servidor de archivos simple, en el cual todos los documentos estaban guardados en archivos PDF.

El volumen de normativas ha crecido enormemente, llegando a ser más de cien documentos que contienen múltiples versiones cada uno. Actualmente se están clasificando por departamentos, lo cual reduce significativamente el número de estas en cada departamento; este proyecto todavía está en etapas iniciales y simplemente es una clasificación que se utilizará como sugerencia al momento de que los colaboradores tengan que leer los distintos documentos.

1.4 Planteamiento y descripción del problema

Al momento de tomar una decisión o realizar una acción, los colaboradores de COOSAJO R.L deben tener en cuenta las normativas que rigen a la Cooperativa, las cuales están clasificadas como: políticas, reglamentos, leyes y acuerdos.

Los documentos han aumentado en volumen rápidamente en los últimos años, lo cual, ahora es un problema para los colaboradores, ya que se les dificulta la lectura, el aprendizaje o la búsqueda de una normativa específica, esto genera otros problemas, como, grandes tiempos de espera al momento de tener que realizar una acción, siendo esto poco eficiente.

Al no contar con un motor de búsqueda ni un filtrado de las normativas se crean contratiempos para los colaboradores, ya que no pueden buscar automáticamente un documento para un caso específico, teniendo ellos que hacerlo entre todos los que existen, o preguntar a alguien más qué hacer en esa determinada situación, generando también problemas para ese colaborador.

1.5 Delimitación del problema

1.5.1 Delimitación geográfica

El EPS se llevó a cabo en el Edificio Oficinas Corporativas “Nuestra Casa” COOSAJÓ, R.L., ubicada en el km 224.5, Aldea Atulapa, Esquipulas, Chiquimula, Guatemala (Ver figura 3).

Figura 3. Ubicación Oficinas de COOSAJO



Fuente: Google Maps, s.f.

1.5.2 Delimitación institucional

El proyecto se realizó para la Cooperativa de Ahorro y Crédito Integral San José Obrero, COOSAJO R.L.

1.5.3 Delimitación personal

Se involucró al licenciado Jorge Flores como supervisor, al ingeniero Darwin Pinto del área de infraestructura y al ingeniero Julio Padilla jefe de IDT.

1.5.4 Delimitación temporal

Se realizó entre los meses de febrero a agosto del año 2020. En este período se realizaron las fases de análisis, desarrollo e implementación del proyecto.

1.5.5 Delimitación teórica

Esta aplicación se desarrolló utilizando tecnologías web y un servidor de búsqueda, donde se utilizó la metodología RUP en conjunto con Kanban para la administración de las tareas necesarias para la creación del software.

1.6 Requerimientos de tecnología de información necesarios para la implementación del proyecto

Es necesario tener un servidor en el cual desplegar las distintas aplicaciones desarrolladas. Este debe contar con un sistema operativo moderno, por ejemplo, Ubuntu 18.04 y un servidor web como NGINX.

1.7 Justificación

La implementación de un motor de búsqueda de normativas nace de la necesidad de querer mejorar la eficiencia dentro de COOSAJO. Debido a la gran cantidad de normas, más de 100 documentos con múltiples versiones de cada uno, puede ser difícil para los colaboradores tener conocimiento de todos ellos; para mejorar esto se implementó de un motor de búsqueda que muestra las normativas más relevantes según el criterio de búsqueda que haya ingresado el colaborador, logrando que los colaboradores sepan que acciones tomar para resolver cualquier problema que se les presente de manera rápida y sencilla.

2. FASE DE TRABAJO TÉCNICO PROFESIONAL

2.1 Descripción del proyecto

Aunque existía un repositorio anteriormente en COOSAJO R.L., este proyecto se creó desde cero, ya que de esta forma se logró implementar nuevas tecnologías que permiten búsquedas mucho más rápidas y precisas, que es uno de los principales objetivos que el proyecto debe de cumplir.

Este proyecto cuenta con una arquitectura cliente servidor, con unas peculiaridades que se explicarán a profundidad más adelante en el apartado de vista de despliegue, para resumir estas, existe un *backend* que funciona localmente en el servidor (NodeJS), este es consultado por otro *backend* (PHP) que sirve de intermediario entre el usuario (*frontend*) y el *backend* local.

Para realizar este *software* se crearon 3 aplicaciones, un *backend*, que es el encargado del ingreso y búsqueda de datos, se codificó en NodeJS y fue utilizado en conjunto con Elasticsearch siendo este el encargado del almacenado y búsqueda de la información.

Además, se creó un *frontend*, que es la interfaz que ven los usuarios, esta es una interfaz amigable siguiendo distintos criterios de *UX*, para hacer el proceso de ingreso y búsqueda de información lo más intuitivo posible. Esta interfaz fue desarrollada en PHP en conjunto con JavaScript.

Adicional a esto se desarrolló un instalador para el *backend* que crea los índices necesarios en Elasticsearch para que la aplicación funcione de manera correcta. Este instalador fue desarrollado en NodeJS, con la mentalidad de hacer un proceso de instalación flexible, esto quiere decir que, además de ser utilizado

para este proyecto, fácilmente puede ser modificado para la creación de índices de diferentes tipos de proyectos que hagan uso de ElasticSearch.

Como se mencionó anteriormente, el *backend* es el encargado de realizar la búsqueda de la información por lo que se crearon dos tipos de esta, una búsqueda por nombre de documento y otra por frases que se encuentran dentro del texto documento.

La búsqueda por nombre de documento funciona en el apartado de ingresar nuevos documentos, es decir, el *dashboard* de administrador, para poder así de forma rápida encontrar que documentos se encuentran disponibles en el repositorio.

La búsqueda por frase será usada por los colaboradores, en la cual colocarán su criterio de búsqueda y el *software* encontrará frases similares dentro de todos los documentos y mostrará los que coincidan con este criterio.

Originalmente se había planteado la clasificación de las normativas, pero esto se descartó principalmente por la falta de acceso a la información, sin embargo, esto no afectó el rendimiento del buscador y, al contrario, esto demuestra la eficiencia de este, dando resultados precisos con una mayor cantidad de documentos sobre los cuales realizar la búsqueda.

La única clasificación que quedo en el proyecto fue la original con la que el sistema contaba: documentos internos y externos, aunque al momento de realizar una búsqueda de frases no se utiliza esta clasificación como filtrado, porque como se mencionó anteriormente, no afecta el rendimiento de esta y además hace el proceso mucho más simple y rápido para los colaboradores ya que no se tienen que preocupar por filtros extras y solo deben de escribir la frase que desean encontrar.

2.2 Beneficios del proyecto

Los beneficiarios del proyecto fueron todos los colaboradores de COOSAJOR.L.

Los principales beneficios fueron:

- Una herramienta para que los colaboradores puedan aumentar la productividad al momento de tener que tomar una decisión.
- El conocimiento y manejo de información dentro de la cooperativa.
- Encontrar de manera rápida cualquier normativo que se necesite.
- Una API de búsqueda que además de utilizarse para las normativas, también podrá ser utilizada por la cooperativa para proyectos que se creen en un futuro.

2.3 Fundamentación teórica

Antes de empezar con la creación del proyecto fue necesario investigar conceptos básicos, acerca de lo que se va a realizar, en este caso el proyecto es un motor de búsqueda.

2.3.1 Motor de búsqueda

“Un motor de búsqueda es una plataforma que permite recuperar archivos almacenados en un servidor de Internet” (Castrillón-Estrada et al., 2008, p. 99). La mayoría de motores de búsqueda utilizan texto para buscar los archivos almacenados.

2.3.2 Índice invertido

Un índice invertido es una estructura de datos que contiene los términos del vocabulario de una colección de documentos y por cada término almacena una lista de registro con información referente al número de identificación (ID) de cada uno de los documentos que contienen al término y de ser necesario, las posiciones en las que aparece (Ariza, 2012, p. 11).

Un ejemplo del funcionamiento de un índice invertido se puede ver en la figura 4.

Figura 4. **Ejemplo del funcionamiento de un índice invertido**

Vocabulario	Documentos	
hola	D1. D2.	D1.
mundo	D1.	Hola mundo
como	D2.	D2.
estas	D2.	¿Hola como estas?

Fuente: elaboración propia.

2.3.3 Analizadores de texto

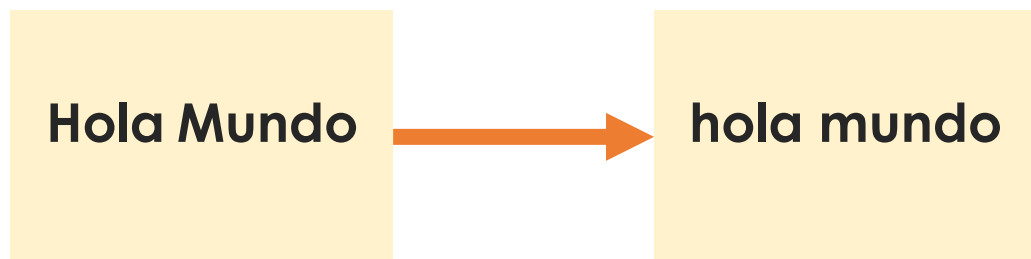
Transformaran el contenido de un texto para adaptarse a las búsquedas realizadas por las personas y dar mejores resultados, un ejemplo de esto sería transformar todo el texto a minúsculas, permitiendo así que la búsqueda de la palabra “HOla” sea lo mismo que buscar la palabra “hola”.

Los analizadores de texto pasan por distintas fases como:

a. *Lowercase*

Esta fase del analizador convierte todo el texto a minúsculas (Ver figura 5).

Figura 5. Ejemplo de la fase *lowercase* del analizador de texto



Fuente: elaboración propia.

b. Charfilter

Remueve caracteres especiales de otros idiomas como las tildes y las “ñ” (Ver figura 6).

Figura 6. Ejemplo de la fase *charfilter* del analizador de texto

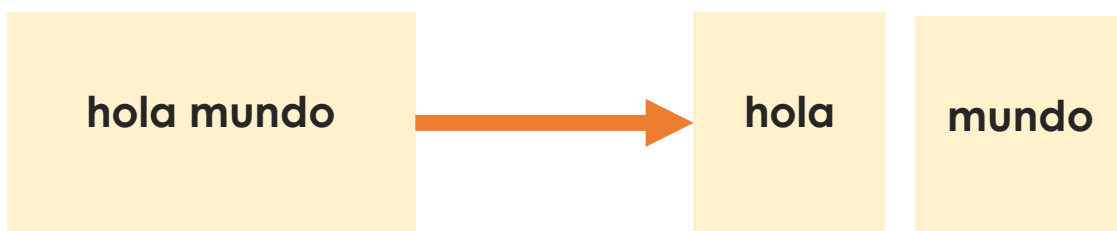


Fuente: elaboración propia.

c. Tokenizer

En esta fase del analizador de texto se separan las “frases” en “tokens” es decir se dividen todas las palabras del texto (Ver figura 7).

Figura 7. Ejemplo del tokenizador del analizador de texto



Fuente: elaboración propia.

d. *Stopwords*

También llamadas palabras vacías, estas son “las palabras sin significado como artículos, pronombres, preposiciones, etc. que son filtradas antes o después del procesamiento de datos en lenguaje natural (texto)” (Palabra vacía, 2019, párr. 1) (Ver figura 8).

Figura 8. Ejemplo de la fase *stopwords* del analizador de texto

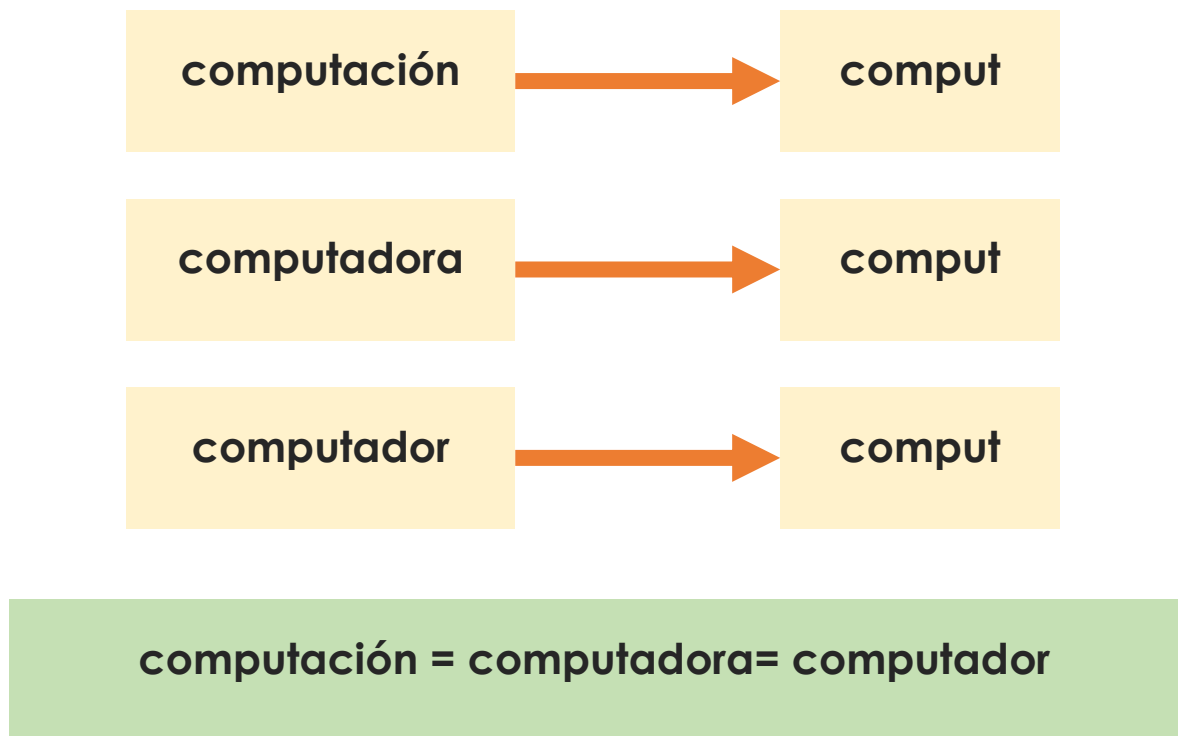


Fuente: elaboración propia.

e. *Stemmer*

Stemming “es un algoritmo que a través de un proceso computacional reduce todas las palabras con la misma raíz a una forma común” (Beth, 1968, p. 22) (Ver figura 9).

Figura 9. Ejemplo de la fase *stemmer* del analizador de texto



Fuente: elaboración propia.

2.3.4 Arquitectura Cliente Servidor

“Un sistema que sigue el patrón cliente-servidor se organiza como un conjunto de servicios y servidores asociados, y de clientes que acceden y usan los servicios” (Sommerville, 2011, p. 161).

2.3.5 Servicios Web y APIs

Massé (2011) indica como un servicio web trabaja en conjunto con una *API* de la siguiente manera:

Los servicios web son servidores web que atienden las necesidades de un sitio web o cualquier otra aplicación. Los programas clientes utilizan las interfaces de programación de aplicación (*APIs*) para comunicarse con los servicios web. En términos generales, una *API* expone un conjunto de datos y funciones para facilitar la interacción entre los programas y permitirles intercambiar información. Una *API Web* de un servicio web, directamente escucha y responde a solicitudes de los clientes (p. 5).

Una forma muy común de crear *APIs* web es la utilización de una arquitectura *REST* (Transferencia de Estado Representacional, en inglés *representational state transfer*), como indica Massé (2011):

Una arquitectura tipo *REST* es comúnmente aplicada en el diseño de *APIs* para servicios web modernos. Una *API* web conformada de una arquitectura *REST* es una *API REST*.

Tener una *API REST* hace que un servicio sea “*RESTful*”. Una *API REST* consiste de un ensamblaje o interconexión de recursos (p. 6).

2.3.6 Kanban

Kanban es un método para gestionar el trabajo que surgió en *Toyota Production System* (TPS). A finales de los años 40, Toyota implementó en su producción el sistema “*just in time*” (justo a tiempo) que en realidad representa un sistema de arrastre. Esto significa que la producción se basa en la demanda de los clientes y no en la práctica tradicional “pull” de fabricar productos e intentar venderlos en el mercado (Kanbanize, s.f., párr. 3).

Kanban funciona como un tablero, generalmente de tres columnas, en una de ellas se colocarán todas las tareas que se realizarán, en otra se deben de poner todas las que están en progreso es decir que se están realizando en el momento y por ultimo las tareas ya realizadas.

Figura 10. Kanban



Fuente: Arriaga Laura, 2018

2.3.7 Proceso Unificado Racional (RUP)

Es una metodología de desarrollo de software la cual se puede describir como un modelo híbrido que combina el desarrollo estático y dinámico como indica Sommerville (2011), RUP se describe en tres perspectivas:

1. Una perspectiva dinámica que muestra las fases del modelo a través del tiempo.
2. Una perspectiva estática que presenta las actividades del proceso que se establecen.
3. Una perspectiva práctica que sugiere buenas prácticas a usar durante el proceso (p. 50).

Sommerville (2011) también describe RUP como un modelo de fases:

El RUP es un modelo en fases que identifica cuatro fases discretas en el proceso de software. Sin embargo, a diferencia del modelo en cascada, donde las fases se igualan con actividades del proceso, las fases en el RUP están más estrechamente vinculadas con la empresa que con las preocupaciones técnicas. [...]

Éstas son:

1. *Concepción* La meta de la fase de concepción es establecer un caso empresarial para el sistema. Deben de identificarse todas las entidades externas (personas y sistemas) que interactuarán con el sistema y definirán dichas interacciones. [...]

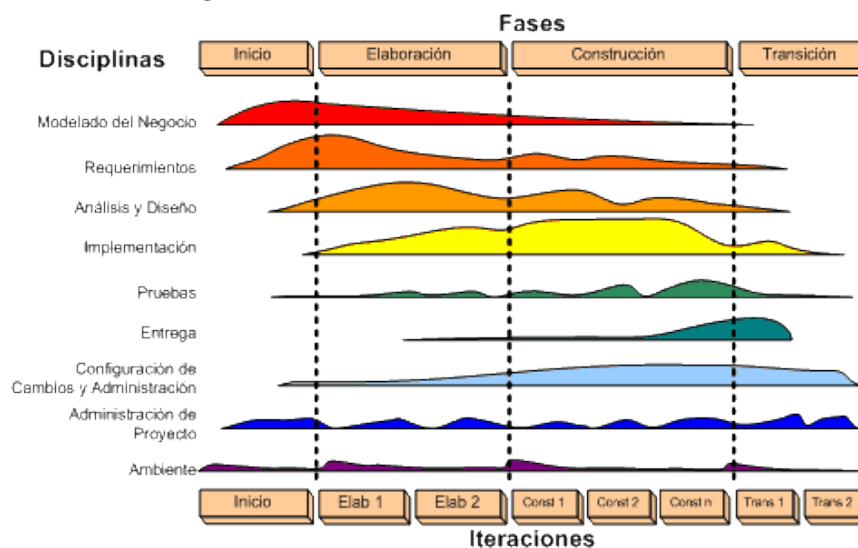
2. *Elaboración* Las metas de la fase de elaboración consisten en desarrollar la comprensión del sistema de dominio, establecer un marco conceptual arquitectónico para el sistema, diseñar el plan del proyecto e identificar los riesgos claves del proyecto. [...]

3. *Construcción* La fase de construcción incluye diseño, programación y pruebas del sistema. Partes del sistema se desarrollan en paralelo y se integran durante esta fase. [...]

4. *Transición* La fase final de RUP se interesa por el cambio del sistema desde la comunidad de desarrollo hacia la comunidad de usuarios, y por ponerlo a funcionar en un ambiente real (pp. 50-51).

Cada fase se trabaja en iteraciones, además de eso también la tarea se puede agrupar en diferentes disciplinas, como se indica en la figura 11:

Figura 11. Diagrama RUP



Fuente: rupnoobs, s.f.

2.3.8 Tecnologías de desarrollo

Ya teniendo claro que se va a desarrollar y los conceptos básicos sobre el funcionamiento del proyecto, fue necesario elegir las tecnologías correctas para la implementación de este proyecto.

Se buscaron tecnologías que además de ser modernas, se acoplaran perfectamente a las necesidades de este proyecto.

a. Apache Lucene

Es una biblioteca de Java que provee una poderosa indexación y características de búsqueda, como también revisión ortográfica, resalto de resultados y capacidades de análisis y tokenización (Apache Lucene, s.f., párr. 2).

b. Elasticsearch

“Elasticsearch es un motor de analítica y análisis distribuido, gratuito y abierto para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y desestructurados” (Elastic, s.f., párr. 1).

c. JavaScript

“Es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás” (MDN Web Docs, s.f., párr. 2).

d. HTML

“Es el lenguaje de marcado que usamos para estructurar y dar significado a nuestro contenido web, por ejemplo, definiendo párrafos, encabezados y tablas de datos, o insertando imágenes y videos en la página” (MDN Web Docs, s.f., párr. 3).

e. CSS

“Es un lenguaje de reglas de estilo que usamos para aplicar estilo a nuestro contenido HTML, por ejemplo, establecer colores de fondo y tipos de letra, y distribuir nuestro contenido en múltiples columnas” (MDN Web Docs, s.f., párr. 4).

f. NGINX

Higuerey (s.f.) indica que “NGINX es un servidor web open source, de alta performance. Cuenta con una arquitectura avanzada basada en eventos — EBA, o *Event-Based Architecture*. Esta característica permite numerosas conexiones simultáneas, lo que proporciona más velocidad y escalabilidad” (párr. 8).

g. Node JS

“Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome” (NodeJS s.f., párr. 1).

h. PHP

“PHP (acrónimo recursivo de PHP: *Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML” (PHP s.f., párr. 1).

2.4 Alcance y productos esperados

Una API de búsqueda que puede ser utilizada para otros proyectos que se desarrollen en el futuro.

Se creó un apartado en el portal de COOSAJO R.L., en el cual se colocó el *dashboard* que permite a los administradores poder agregar los normativos y ver estadísticas del uso del sistema.

En el portal se agregó una barra de búsqueda en la parte superior de la página. Al momento de ingresar la frase que se desea encontrar, se abrirá un *modal* que mostrará todos los resultados que coincidan.

Se entregó un manual técnico que detalla el proceso de instalación y las distintas partes que conforman el proyecto.

También se entregó un manual de usuario en el cual se detalla el proceso para ingresar documentos, ver estadísticas y realizar búsquedas.

2.5 Plan de trabajo

Como metodología de trabajo se decidió utilizar RUP en conjunto con Kanban. RUP divide el trabajo en fases, y cada una se divide en diferentes iteraciones, así que, para realizar un desarrollo aún más ágil, se decidió utilizar Kanban para la gestión de cada fase del proyecto.

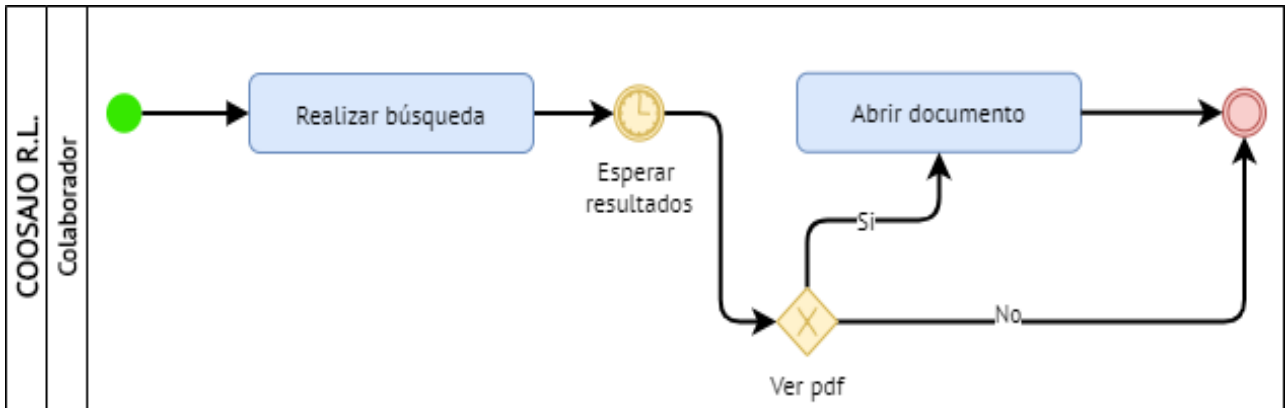
2.6 Modelado de negocio

2.6.1 Modelado de procesos de negocio

Los procesos que realizan los colaboradores son muy similares a los que existían anteriormente en COOSAJO R.L. Por ejemplo, al momento de ingresar un normativo al sistema, el proceso es exactamente el mismo para el colaborador, este diagrama se muestra en la Figura 2.

Pero además de estos procesos, también se crearon otros nuevos que no existían en el sistema anterior, como la búsqueda de documentos por frase o texto que se encuentra dentro de estos. En la figura 12 se muestra un diagrama que describe este proceso.

Figura 12. **Modelado de proceso de negocio realizar búsqueda**



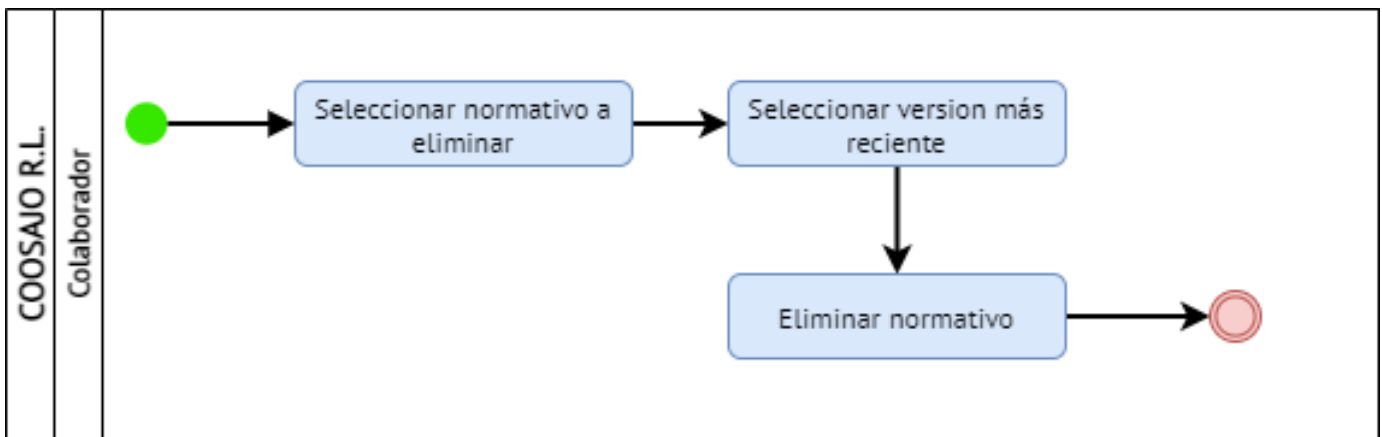
Fuente: elaboración propia.

Y también existe un proceso de eliminación de normativos, pero este es únicamente para remover instantáneamente un documento, por ejemplo, en caso de subir un documento erróneo.

Esto se realizó de esta forma, ya que, si un colaborador realiza una acción para resolver un determinado problema basándose en un normativo, pero luego se introduce una nueva versión de este documento que cambia el proceso que se debe realizar para abordar la situación, el colaborador debe poder respaldarse con la versión anterior del reglamento, por lo que estos no se pueden eliminar.

Así que, para eliminar un normativo se deben cumplir los siguientes criterios: solo puede realizarse en la versión más reciente del documento, es decir, la versión que se acaba de subir al sistema y además no debe haber sido visualizado por ningún usuario, en caso contrario, la aplicación no permitirá la eliminación de este. El proceso de eliminación se muestra en el la figura 13.

Figura 13. **Modelado de proceso de negocio eliminar normativo**



Fuente: elaboración propia.

2.7 Requerimientos

2.7.1 Requerimientos funcionales

- El sistema tiene una interfaz amigable para el usuario.
- Se muestra una barra de búsqueda en la parte superior del portal interno de la cooperativa.
- Todos los colaboradores con acceso al portal podrán realizar búsquedas de normativas a través de este.
- Al momento que se realice una búsqueda, el sistema mostrará los posibles resultados en una nueva página.
- Cuando se encuentre el resultado deseado, se podrá abrir el documento que contiene la normativa de manera rápida y sencilla.
- El sistema permite el ingreso de normativas ya existentes a los usuarios autorizados.
- Se aprueba el ingreso de nuevas normativas únicamente a los usuarios autorizados.
- Se cuenta con versionados de documentos.

2.7.2 Requerimientos no funcionales

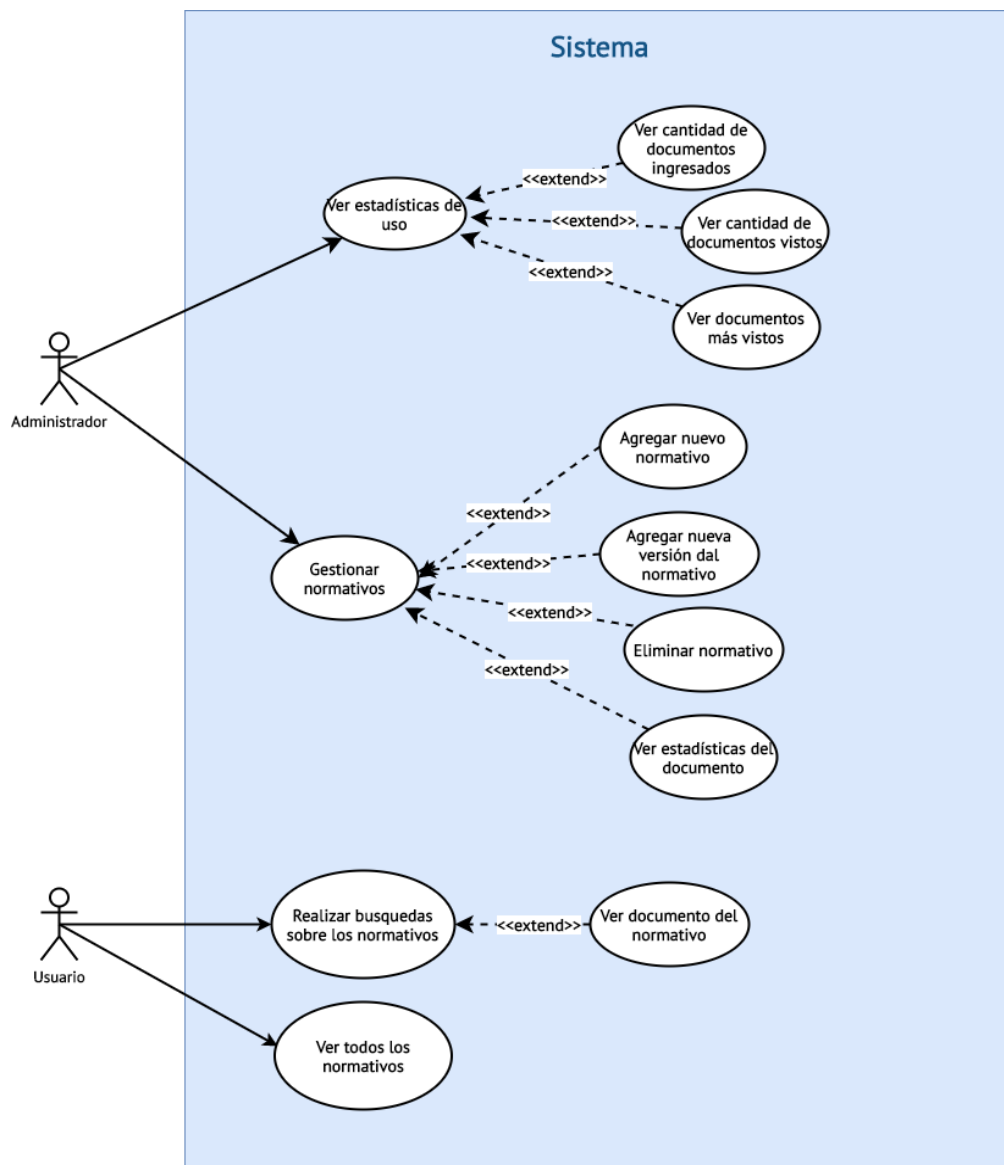
- El sistema debe dar una respuesta en un tiempo menor a 10 segundos.
- Únicamente se permite el ingreso de documentos PDF que contengan texto, no se buscan imágenes ni texto escaneado.
- Solo es posible el ingreso de PDF con una estructura bien definida, es decir, todos los metadatos dentro de este deben estar correctos, por ejemplo, los saltos de página, o saltos de línea.
- El tiempo de aprendizaje que le toma a un nuevo usuario aprender a usar el sistema debe ser menor a 1h.
- El software debe mostrar resultados que sean útiles y de relevancia para el usuario que está buscándolos.
- Los datos presentados deben ser los más actualizados.
- Se debe contar con una alta disponibilidad.

2.8 Análisis y diseño

2.8.1 Casos de uso

La figura 14 presenta los casos de uso de la aplicación.

Figura 14. Casos de uso de la aplicación



Fuente: elaboración propia.

2.8.2 Narración de los casos de uso

En las tablas de la 1 a la 12 se presentan la narración de los casos de uso.

Tabla 1. Narración caso de uso "ver estadísticas de uso"

Caso de uso	Ver estadísticas de uso
Actores	Administrador
Propósito	Permite al usuario administrador ver las estadísticas del uso de la plataforma.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder ver la sección de <i>dashboard</i> .
Flujo básico	1. Entrar al <i>dashboard</i>. 2. Explorar las diferentes gráficas que muestran las estadísticas de uso de la aplicación.
Flujos alternos	No aplica.
Postcondición	Visualización de las estadísticas.
Puntos de inclusión	No aplica.
Puntos de extensión	Ver cantidad de documentos ingresados. Ver cantidad de documentos vistos. Ver documentos más vistos.

Fuente: elaboración propia.

Tabla 2. Narración caso de uso "ver cantidad de documentos ingresados"

Caso de uso	Ver cantidad de documentos ingresados
Actores	Administrador
Propósito	Muestra la cantidad de documentos ingresados que existen en el sistema, sin contar las versiones.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder ver la sección de <i>dashboard</i> .
Flujo básico	1. Visualización de la cantidad de documentos ingresados.
Flujos alternos	No aplica.
Postcondición	No aplica.
Puntos de inclusión	No aplica.
Puntos de extensión	No aplica.

Fuente: elaboración propia.

Tabla 3. Narración caso de uso “ver cantidad de documentos vistos”

Caso de uso	Ver cantidad de documentos vistos
Actores	Administrador
Propósito	Muestra la información de cuantos documentos se han visto en total, y también la información de cuantos documentos han sido vistos en determinado intervalo de tiempo.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder ver la sección de <i>dashboard</i> .
Flujo básico	1. Encontrarse en el <i>dashboard</i>. 2. Seleccionar intervalo de tiempo deseado.
Flujos alternos	No aplica.
Postcondición	No aplica.
Puntos de inclusión	No aplica.
Puntos de extensión	No aplica.

Fuente: elaboración propia.

Tabla 4. Narración caso de uso "ver documentos más vistos"

Caso de uso	Ver documentos más vistos
Actores	Administrador
Propósito	Muestra los 6 documentos que han sido más vistos en el sistema.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder ver la sección de <i>dashboard</i> .
Flujo básico	1. Encontrarse en el <i>dashboard</i>. 2. Visualización de estadística de los 6 documentos más vistos.
Flujos alternos	No aplica.
Postcondición	No aplica.
Puntos de inclusión	No aplica.
Puntos de extensión	No aplica.

Fuente: elaboración propia.

Tabla 5. Narración caso de uso "gestionar normativos"

Caso de uso	Gestionar normativos
Actores	Administrador
Propósito	Permite al usuario ver los normativos, y gestionar cada normativo individualmente.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder ver la sección de <i>dashboard</i> .
Flujo básico	1. Encontrarse en el <i>dashboard</i>. 2. Ver tabla de normativos. 3. Seleccionar normativo a gestionar.
Flujos alternos	No aplica.
Postcondición	No aplica.
Puntos de inclusión	No aplica.
Puntos de extensión	Agregar nuevo normativo. Agregar nueva versión del normativo. Eliminar normativo. Ver estadísticas de uso.

Fuente: elaboración propia.

Tabla 6. Narración caso de uso "agregar nuevo normativo"

Caso de uso	Agregar nuevo normativo
Actores	Administrador
Propósito	Permite agregar un normativo completamente nuevo al sistema.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder ver la sección de dashboard.
Flujo básico	1. Encontrarse en el dashboard. 2. Dar clic en el botón agregar nuevo documento. 3. Subir documento pdf. 4. Agregar nombre. 5. Agregar fecha de aprobación. 6. Seleccionar tipo de normativo. 7. Dar clic en subir documento.
Flujos alternos	Error al momento de agregar el normativo.
Postcondición	<i>Backend</i> guardará el nuevo documento en el sistema, al usuario se le mostrará un mensaje de confirmación. En caso que ocurra un error se mostrará un mensaje de error.
Puntos de inclusión	No aplica.
Puntos de extensión	No aplica.

Fuente: elaboración propia.

Tabla 7. Narración caso de uso "agregar nueva versión a un normativo"

Caso de uso	Agregar nueva versión a un normativo
Actores	Administrador
Propósito	Permite agregar una nueva versión a un normativo ya existente.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder ver la sección de <i>dashboard</i> .
Flujo básico	1. Encontrarse en el <i>dashboard</i>. 2. Dar clic en el normativo al cual se le quiere agregar una nueva versión. 3. Ir a la sección de agregar versión. 4. Subir nuevo documento PDF. 5. Agregar fecha de aprobación. 6. Dar clic en subir documento.
Flujos alternos	Error al momento de agregar el normativo.
Postcondición	<i>Backend</i> guardará el nuevo documento en el sistema, al usuario se le mostrará un mensaje de confirmación. En caso que ocurra un error se mostrará un mensaje de error.
Puntos de inclusión	No aplica.
Puntos de extensión	No aplica.

Fuente: elaboración propia.

Tabla 8. Narración caso de uso "eliminar normativo"

Caso de uso	Eliminar normativo
Actores	Administrador
Propósito	Permite eliminar un normativo si se cumplen ciertos requisitos que son: ser la versión más reciente del normativo y que no haya sido visto por ningún usuario.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder ver la sección de <i>dashboard</i> . El normativo no ha sido visto por ningún colaborador (exceptuando administradores). Es la versión más reciente del normativo.
Flujo básico	1. Encontrarse en el <i>dashboard</i>. 2. Dar clic en el normativo que se quiere eliminar. 3. Ir a la sección de eliminar. 4. Escribir la palabra eliminar en el campo de texto. 5. Dar clic al botón de eliminar.
Flujos alternos	Error al momento de eliminar el documento.
Postcondición	<i>Backend</i> eliminará el documento en el sistema, al usuario se le mostrará un mensaje de confirmación. En caso que ocurra un error se mostrará un mensaje de error.
Puntos de inclusión	No aplica.
Puntos de extensión	No aplica.

Fuente: elaboración propia.

Tabla 9. Narración casos de uso "ver estadísticas del documento"

Caso de uso	Ver estadísticas del documento
Actores	Administrador
Propósito	Muestra estadísticas del documento, la información de estas estadísticas son el número de vistas totales que contiene el documento agrupadas por departamento.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder ver la sección de <i>dashboard</i> .
Flujo básico	1. Encontrarse en el <i>dashboard</i>. 2. Dar clic en el normativo del cual se quiere visualizar las estadísticas. 3. Ir a la sección de estadísticas.
Flujos alternos	No aplica.
Postcondición	Se mostrará gráfica con las estadísticas de visualización de documento.
Puntos de inclusión	No aplica.
Puntos de extensión	No aplica.

Fuente: elaboración propia.

Tabla 10. Narración caso de uso "realizar búsquedas sobre los normativos"

Caso de uso	Realizar búsquedas sobre los normativos
Actores	Usuario
Propósito	Permite al usuario buscar una frase en el texto del documento y el sistema dará resultados que crea correspondientes al criterio de búsqueda del usuario.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder entrar al portal de COOSAJÓ R.L.
Flujo básico	1. Encontrarse en el portal de COOSAJÓ R.L. 2. Ingresar frase o texto a buscar en el campo de texto que se encuentra en la parte superior de la interfaz.
Flujos alternos	No se encuentra texto. No hay resultados.
Postcondición	Se mostrarán los resultados en un modal, en caso de no haber texto no realizará la búsqueda, y en caso de no haber resultados, mostrará esa información.
Puntos de inclusión	No aplica.
Puntos de extensión	Ver documento del normativo

Fuente: elaboración propia.

Tabla 11. Narración caso de uso "ver documento del normativo"

Caso de uso	Ver documento del normativo
Actores	Usuario
Propósito	Permite abrir el documento pdf del normativo.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder entrar al portal de COOSAJÓ R.L. Haber realizado una búsqueda con resultados satisfactorios.
Flujo básico	1. Realizar búsqueda. 2. Dar clic sobre resultado de búsqueda.
Flujos alternos	No aplica.
Postcondición	Se abrirá el documento pdf en la página en la que se encuentra el resultado de la búsqueda.
Puntos de inclusión	No aplica.
Puntos de extensión	No aplica.

Fuente: elaboración propia.

Tabla 12. **Narración caso de uso "ver todos los normativos"**

Caso de uso	Ver todos los normativos
Actores	Usuario
Propósito	Permite al usuario ver todos los normativos existentes.
Precondición	El usuario cuenta con credenciales de acceso autorizadas para poder entrar al portal de COOSAJÓ R.L. Encontrarse en la sección de documentos públicos.
Flujo básico	1. Entrar a la sección de documentos públicos.
Flujos alternos	No aplica.
Postcondición	Tabla con todos los documentos ingresados, además de un buscador para estos.
Puntos de inclusión	No aplica.
Puntos de extensión	No aplica.

Fuente: elaboración propia.

2.8.3 Puntos de conexión del sistema

El *backend* funciona como un servidor *API RESTful*, el cual está organizado en 3 grupos, que son: repositorio, buscador y estadísticas.

- Repository

En este grupo se resuelven peticiones acerca del repositorio de documentos, es decir la gestión de estos, como por ejemplo obtener los documentos o subir un documento nuevo. El servidor atenderá las peticiones a las direcciones presentadas en las tablas 13-19.

Tabla 13. Punto de conexión obtener documentos

GET		/api/owl/repository/documents	
Obtiene todos los documentos que están guardados en el repositorio (última versión de cada documento).			
Parámetros			
NOMBRE	VARIABLE	TIPO	DESCRIPCIÓN
page	Integer	Query	Número de página que se quiere obtener.
size	Integer	Query	Cantidad de elementos por página.
name	String	Query	Nombre del documento para utilizar como filtro.
Responses			
200			application/json
500			application/json

Fuente: elaboración propia.

Tabla 14. **Punto de conexión agregar documento**

POST		/api/owl/repository/documents	
Agrega un nuevo documento al repositorio. Si se pasa el parámetro previousId será una nueva versión del documento anterior			
Parámetros			
NOMBRE	VARIABLE	TIPO	DESCRIPCIÓN
name	String	FormData	Nombre del documento.
date	Long	FormData	Fecha de aprobación (en ms desde el epoch).
pdf	File	FormData	Documento PDF.
previousId	String	FormData	Id del documento al que se le quiere agregar una nueva versión.
Responses			
201			application/json
500			application/json

Fuente: elaboración propia.

Tabla 15. **Punto de conexión obtener documento**

GET		/api/owl/repository/documents/{id}	
Obtiene el documento con el id solicitado			
Parámetros			
NOMBRE	VARIABLE	TIPO	DESCRIPCIÓN
id	String	Path	Id del documento
Responses			
200			application/json
500			application/json

Fuente: elaboración propia.

Tabla 16. **Punto de conexión eliminar documento**

DELETE		/api/owl/repository/documents/{id}	
Eliminará el documento en caso de poder realizarse esa acción			
Parámetros			
NOMBRE	VARIABLE	TIPO	DESCRIPCIÓN
id	String	Path	Id del documento
Responses			
200		application/json	
500		application/json	

Fuente: elaboración propia.

Tabla 17. **Punto de conexión obtener pdf**

GET		/api/owl/repository/files/{id}	
Obtiene el URL en el que se encuentra el archivo pdf de normativas solicitado, además guarda información en estadísticas que ese documento ha sido visto			
Parámetros			
NOMBRE	VARIABLE	TIPO	DESCRIPCIÓN
id	String	Path	Id del documento
Responses			
200			application/json
500			application/json

Fuente: elaboración propia.

Tabla 18. **Punto de conexión documento puede ser eliminado**

GET	/api/owl/repository/documents/{id}/can-be-deleted		
Obtiene información sobre si el documento puede ser eliminado o no, teniendo en cuenta criterios como si es la última versión y no ha sido visto por ningún usuario.			
Parámetros			
NOMBRE	VARIABLE	TIPO	DESCRIPCIÓN
id	String	Path	Id del documento
Responses			
200			application/json
500			application/json

Fuente: elaboración propia.

Tabla 19. **Punto de conexión obtener versiones del documento**

GET	/api/owl/repository/documents/{id}/versions		
Obtiene las versiones existentes del documento solicitado.			
Parámetros			
NOMBRE	VARIABLE	TIPO	DESCRIPCIÓN
id	String	Path	Id del documento
Responses			
200	application/json		
500	application/json		

Fuente: elaboración propia.

- Search

En este grupo se resuelven peticiones del buscador de normativas, el cual atiende solicitudes en la dirección presentada en la tabla 20.

Tabla 20. **Punto de conexión realizar búsqueda**

GET		/api/owl/search	
Obtiene los documentos que coincidan con la búsqueda.			
Parámetros			
NOMBRE	VARIABLE	TIPO	DESCRIPCIÓN
text	String	Query	Frase que se desea buscar.
page	Integer	Query	Número de página que se quiere obtener.
size	Integer	Query	Cantidad de elementos por página.
Responses			
200		application/json	
500		application/json	

Fuente: elaboración propia.

- Stats

En este grupo se resuelven peticiones acerca de las estadísticas del sistema, por ejemplo, los documentos más vistos. Se atienden las direcciones que se muestran de la tablas 21 a la 24.

Tabla 21. **Punto de conexión vistas totales**

GET	/api/owl/stats/total-views
Obtiene la cantidad total vistas que se han realizado	
Responses	
200	application/json
500	application/json

Fuente: elaboración propia.

Tabla 22. **Punto de conexión documentos más vistos**

GET	/api/owl/stats/documents
Devolverá los 6 documentos más vistos en general.	
Responses	
200	application/json
500	application/json

Fuente: elaboración propia.

Tabla 23. **Punto de conexión vistas totales**

GET	/api/owl/stats/views-per-date
Obtiene la cantidad de vistas se han realizado en un intervalo de tiempo dado. Estos resultados describen el uso del sistema.	
Parámetros	
NOMBRE	VARIABLE TIPO DESCRIPCIÓN
interval	String Query (month, week, day, hour, minute)
Responses	
200	application/json
500	application/json

Fuente: elaboración propia.

Tabla 24. **Punto de conexión estadísticas de un documento**

GET	/api/owl/stats/documents/{id}		
Obtiene las estadísticas del documento dado, esas estadísticas son el número de veces que ha sido visto por área.			
Parámetros			
NOMBRE	VARIABLE	TIPO	DESCRIPCIÓN
id	String	Path	Id del document
Responses			
200			application/json
500			application/json

Fuente: elaboración propia.

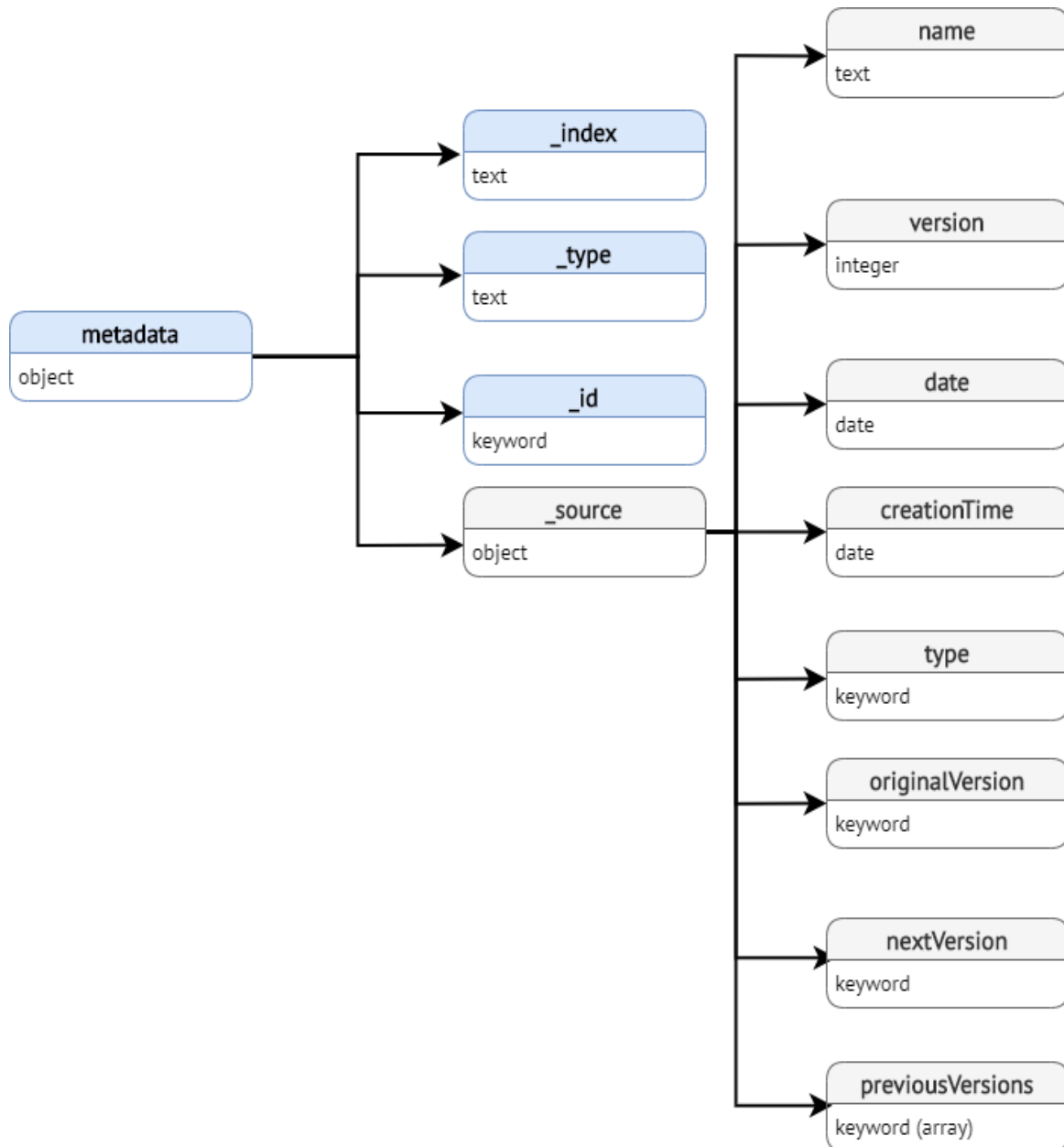
2.8.4 Diagrama de base de datos (ElasticSearch)

ElasticSearch es una base de datos no relacional, es decir que no existe una relación entre un índice y otro. Se crearon 3 índices para el funcionamiento de este proyecto:

- Metadatos

Este índice almacena la información necesaria para el repositorio, es decir que gracias a este índice se puede saber todos los documentos ingresados, además de llevar el control de las versiones de estos documentos (figura 15 y tabla 25).

Figura 15. **Modelo índice *metadata***



Fuente: elaboración propia.

Tabla 25. **Descripción índice *metadata***

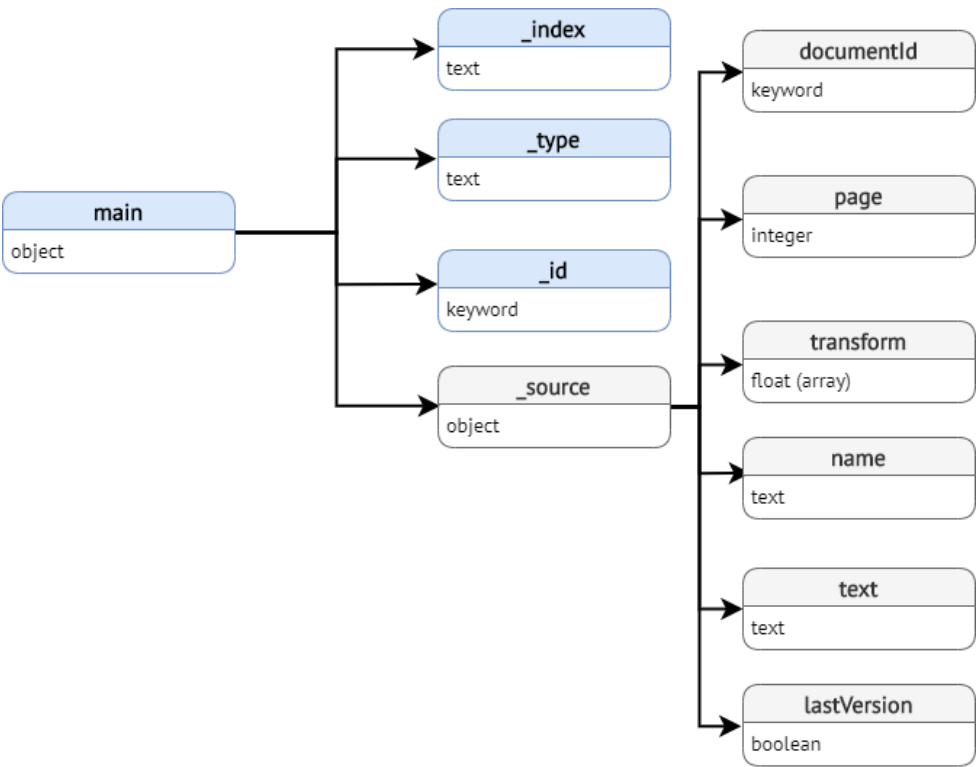
Campo	Tipo	Descripción
name	Text	Nombre del documento
version	Integer	Versión del documento
date	Date	Fecha de aprobación
creationTime	Date	Fecha y tiempo de creación
type	Keyword	Tipo de documento (interno, externo)
originalVersion	Keyword	Id de la versión original del documento
nextVersion	Keyword	Id de la versión siguiente del documento
previousVersions	Keyword []	Id de todos los documentos anteriores

Fuente: elaboración propia.

- *Main*

Este índice almacena el contenido de cada normativo, es decir todo el texto que se encuentra dentro del PDF, el texto del documento es separado por párrafos. Además, se incluye información acerca de la posición en la que se encuentra el párrafo en el documento (figura 16 y tabla 26).

Figura 16. **Modelo índice *main***



Fuente: elaboración propia.

Tabla 26. **Descripción índice *main***

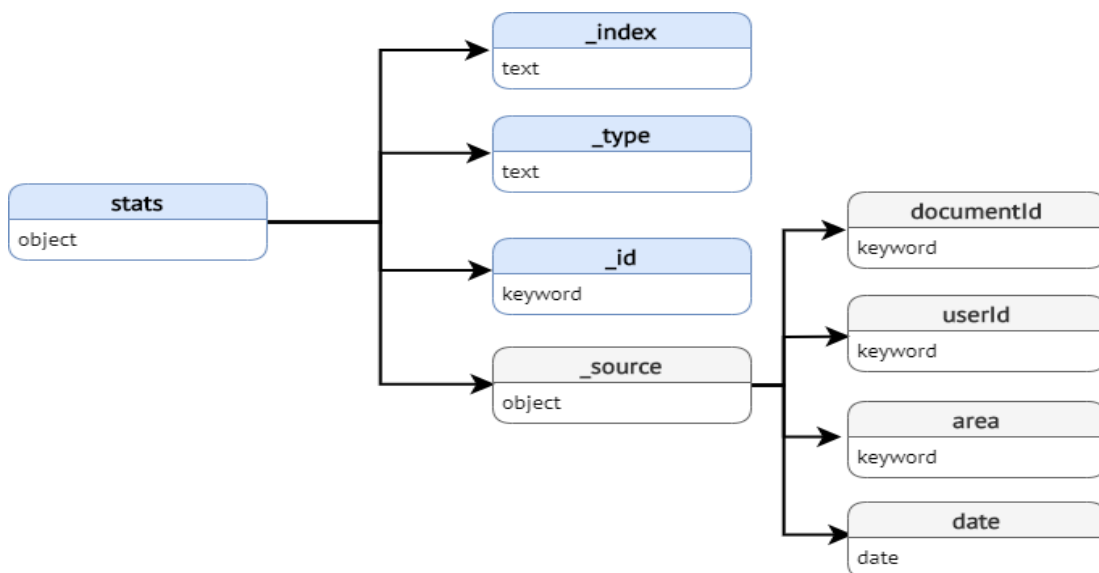
Campo	Tipo	Descripción
documentId	Keyword	Id del documento
page	Integer	Página del documento en la que se encuentra este párrafo.
transform	Float []	Posición del párrafo en la página.
name	Text	Nombre del documento
text	Text	Texto del párrafo
lastVersion	Boolean	Verdadero en caso que este documento se encuentre en su versión más reciente.

Fuente: elaboración propia.

- *Stats*

Este índice almacena estadísticas del uso de la aplicación, conteniendo información sobre quien ha visto las normativas, cuando se vieron y el departamento al que pertenece (figura 17 y tabla 27).

Figura 17. **Modelo índice stats**



Fuente: elaboración propia.

Tabla 27. **Descripción índice stats**

Campo	Tipo	Descripción
documentId	Keyword	Id del documento
userId	Keyword	Id del usuario que vio el documento.
area	Keyword	Departamento al que pertenece el usuario.
date	Text	Fecha en la que se vio el documento.

Fuente: elaboración propia.

2.8.5 Procesos

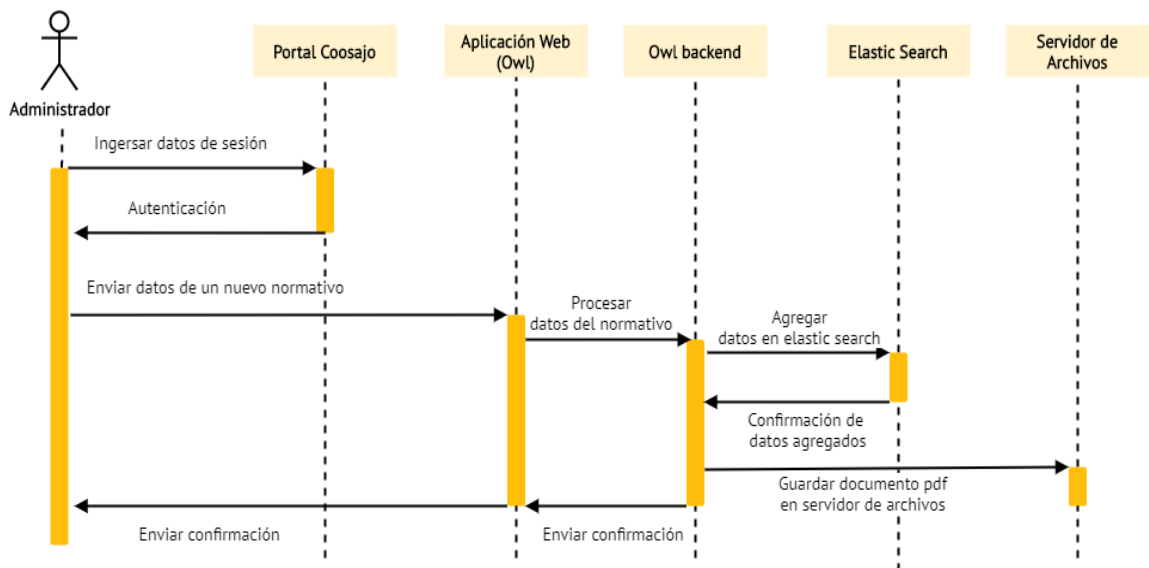
Existen diferentes procesos que se realizan en el sistema. Algunos de estos no cambiaron drásticamente desde la versión anterior del repositorio con el que contaba COOSAJO R.L.

Un ejemplo de esto sería que el ingreso de documentos sigue siendo básicamente lo mismo para el colaborador, exceptuando los cambios de interfaz comparados con la versión anterior. La diferencia en estos procesos viene directamente en la forma en la que el sistema maneja la información, ya que todo se diseñó desde cero, utilizando nuevas tecnologías.

- Agregar Nuevo Normativo

Este proceso es utilizado tanto para agregar nuevos documentos, como para nuevas versiones (figura 18).

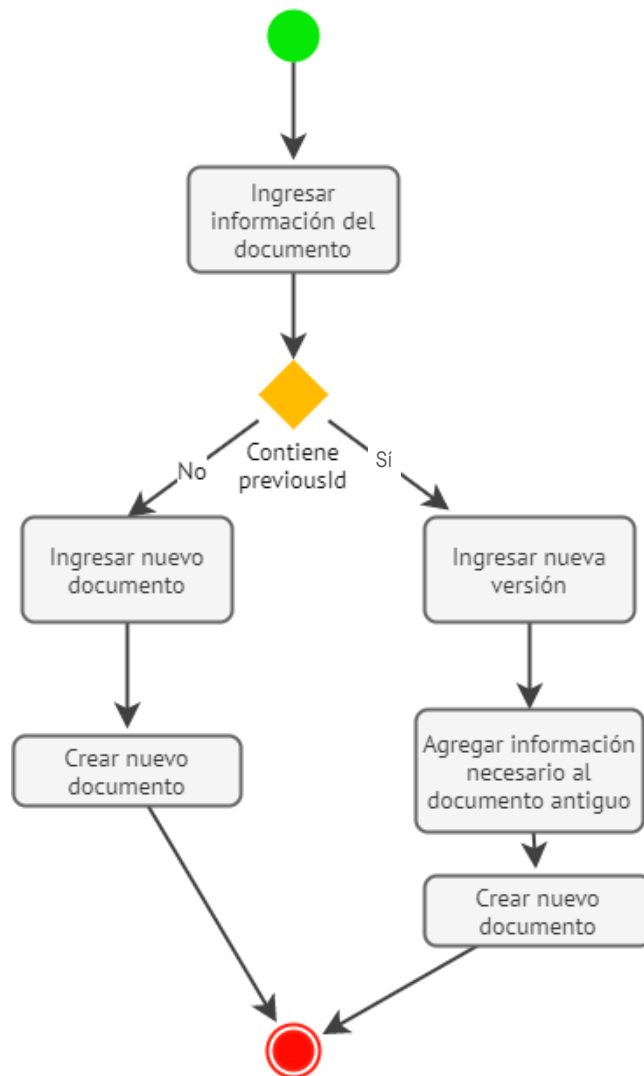
Figura 18. Diagrama de secuencias agregar documento



Fuente: elaboración propia.

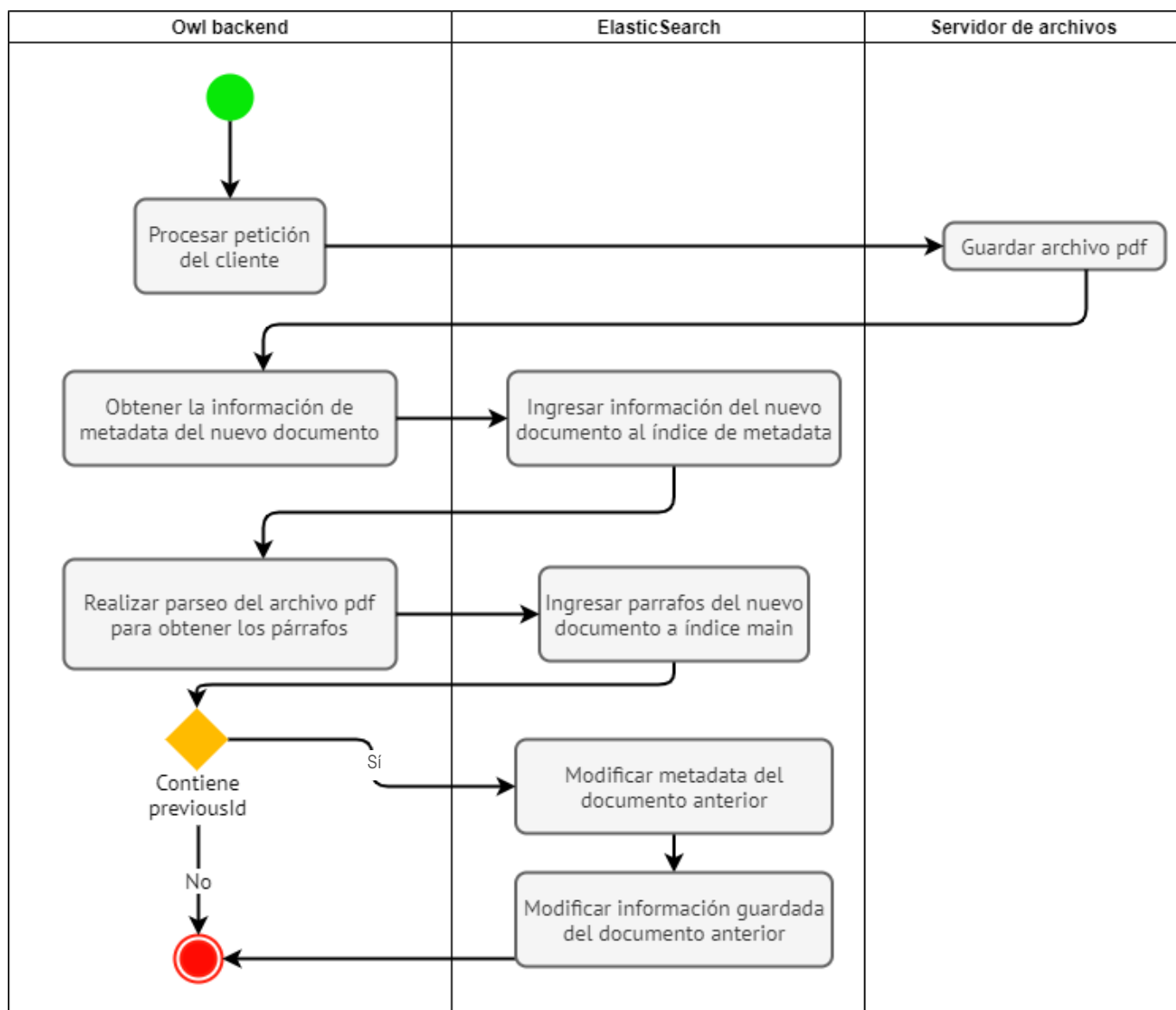
Al momento de llegar al *backend* de NodeJS este decidirá si debe agregar un nuevo documento o una nueva versión de un documento ya existente (Ver figuras 19 y 20).

Figura 19. **Diagrama de actividades agregar documento (1)**



Fuente: elaboración propia.

Figura 20. Diagrama de actividades agregar documento (2)

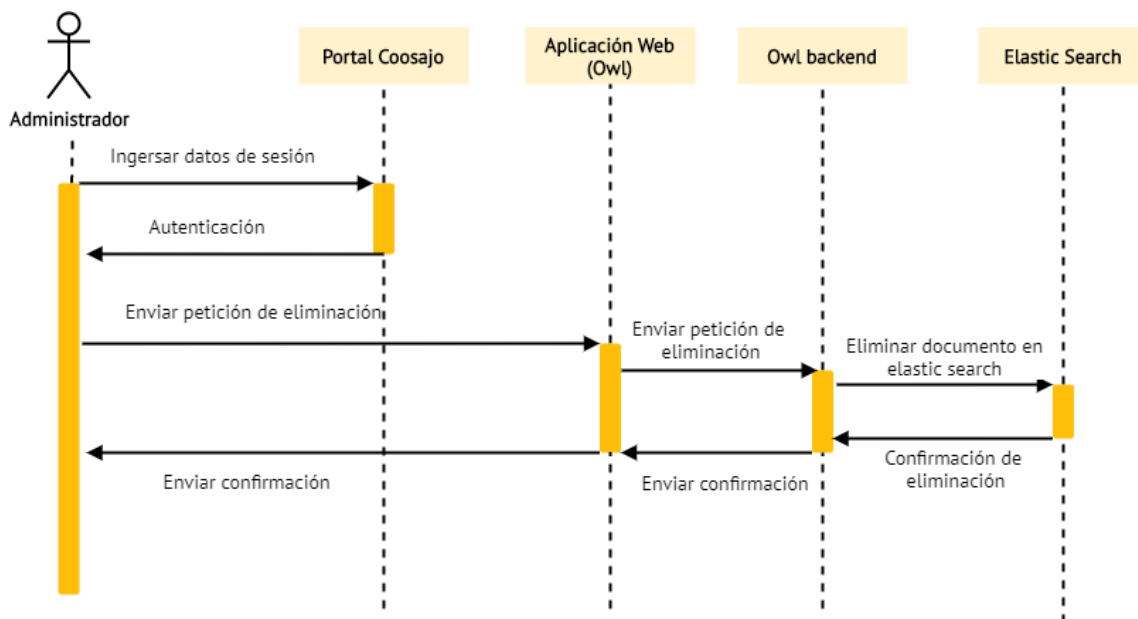


Fuente: elaboración propia.

- Eliminar Normativo

La figura 21 muestra el diagrama de secuencia para eliminar el documento.

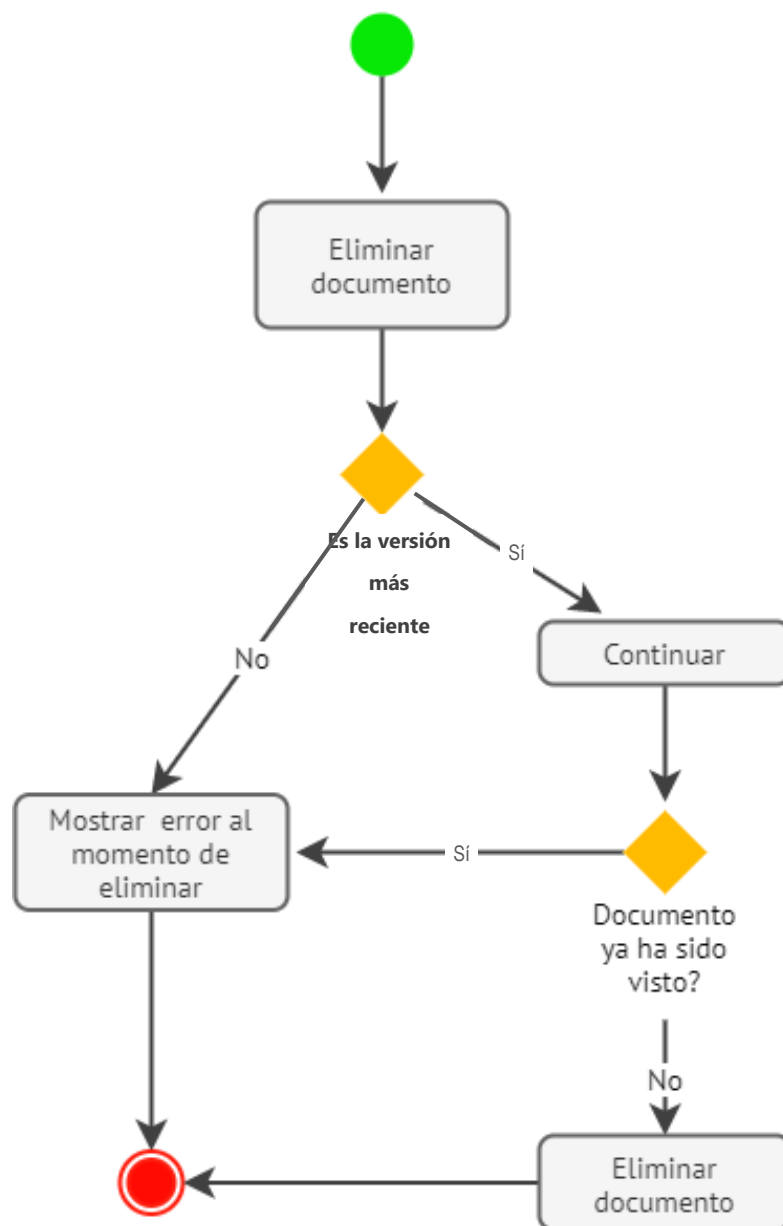
Figura 21. **Diagrama de secuencia eliminar documento**



Fuente: elaboración propia.

Antes de eliminar un documento el sistema deberá comprobar que este pueda ser eliminado, es decir comprobará que sea la versión más reciente del normativo y que este no haya sido visualizado por un colaborador (exceptuando al administrador) (Ver figura 22).

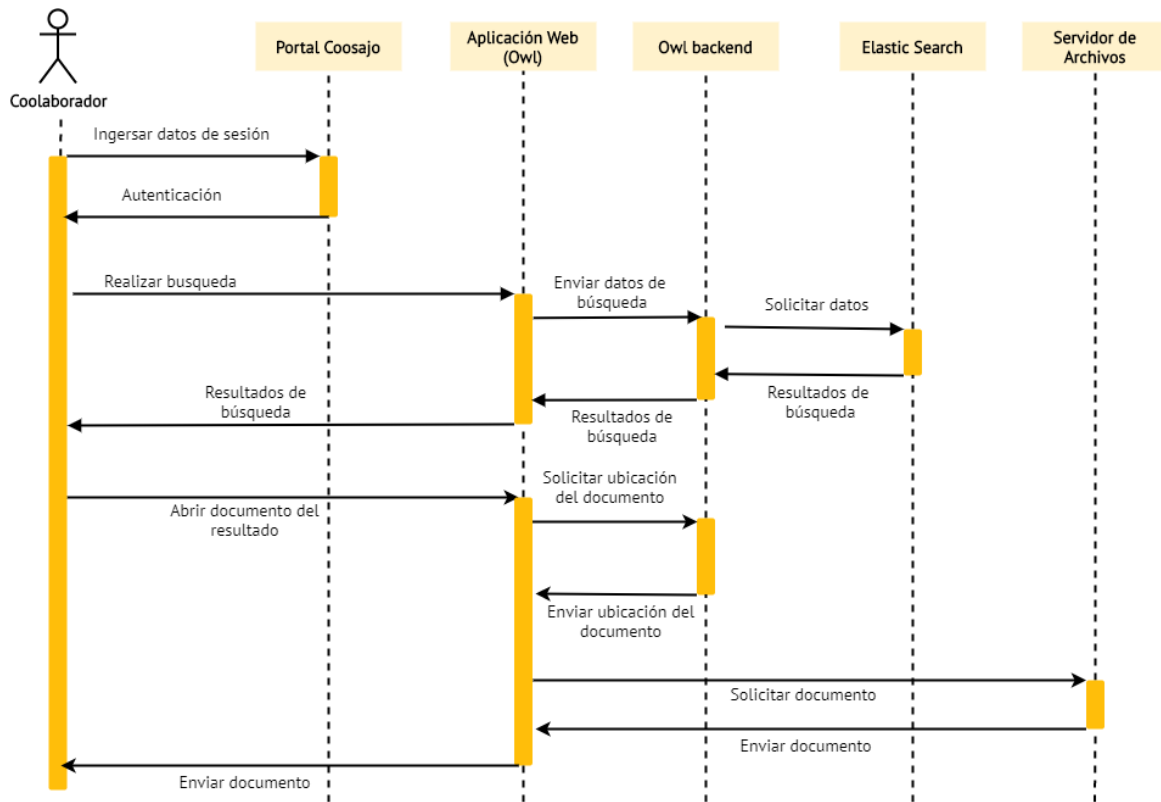
Figura 22. Diagrama de actividades eliminar documento



Fuente: elaboración propia.

- Realizar búsqueda

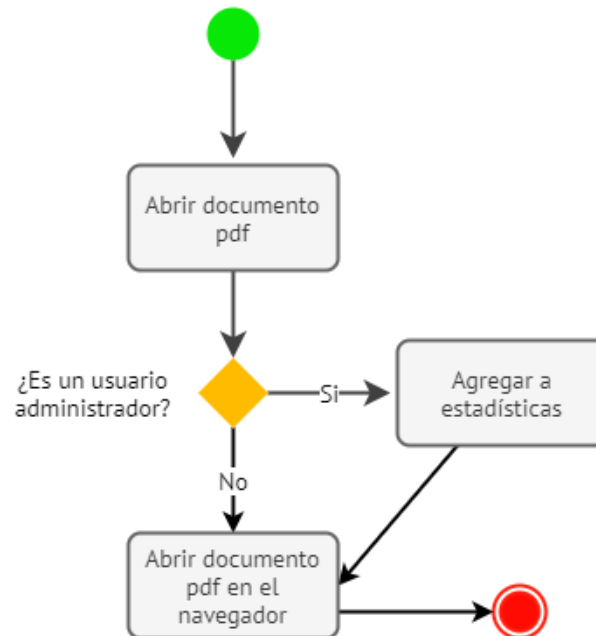
Figura 23. Diagrama de secuencias realizar búsqueda



Fuente: elaboración propia.

Cuando el usuario realiza una búsqueda, esta retornará los diferentes resultados que coincidan con esta, al darle clic a uno se abrirá el documento PDF en la página en la cual se encuentra este (Ver figura 23). Antes de abrir el PDF, el sistema decidirá si agregar o no la búsqueda a las estadísticas de la aplicación, esto lo hará dependiendo de la autorización del usuario que esté conectado, es decir que, si el usuario es un administrador, la búsqueda no se agregará a las estadísticas, de caso contrario, esta sí se hará (Ver figura 24).

Figura 24. Diagrama de actividades abrir documento



Fuente: elaboración propia.

2.9 Implementación

2.9.1 Prototipos

Antes de empezar la creación de la aplicación se hicieron *mockups* de la interfaz, las dos interfaces principales que se crearon fueron la del buscador y la del repositorio.

- Buscador

El *mockup* es un diseño simple, contando únicamente con una barra de búsqueda en la que el usuario ingresa el texto que quiere encontrar, y al momento de presionar la tecla *ENTER*, se muestran los resultados en la parte inferior de la página (Ver figura 25).

Figura 25. **Mockup buscador**

The mockup shows a search interface with the following elements:

- Header:** The word "Normativas" is displayed at the top left.
- Search Bar:** A rectangular input field with the placeholder text "Escribe tu búsqueda".
- Results Section:** Four identical blocks are stacked vertically. Each block contains:
 - The word "Resultado" in bold.
 - A paragraph of Lorem Ipsum text.
 - A button labeled "Ver documento".
- Footer:** A button labeled "Ver más resultados" is positioned at the bottom right of the results area.

Fuente: elaboración propia.

- Repositorio

El *mockup* del repositorio es una tabla en la que se muestran todos los normativos que están ingresados, en la parte superior hay una barra de búsqueda para encontrar estos utilizando su nombre. Además, también cuenta con un botón para agregar nuevos documentos (Ver figura 26).

Figura 26. **Mockup repositorio de normativas**

Normativas

Escribe tu búsqueda

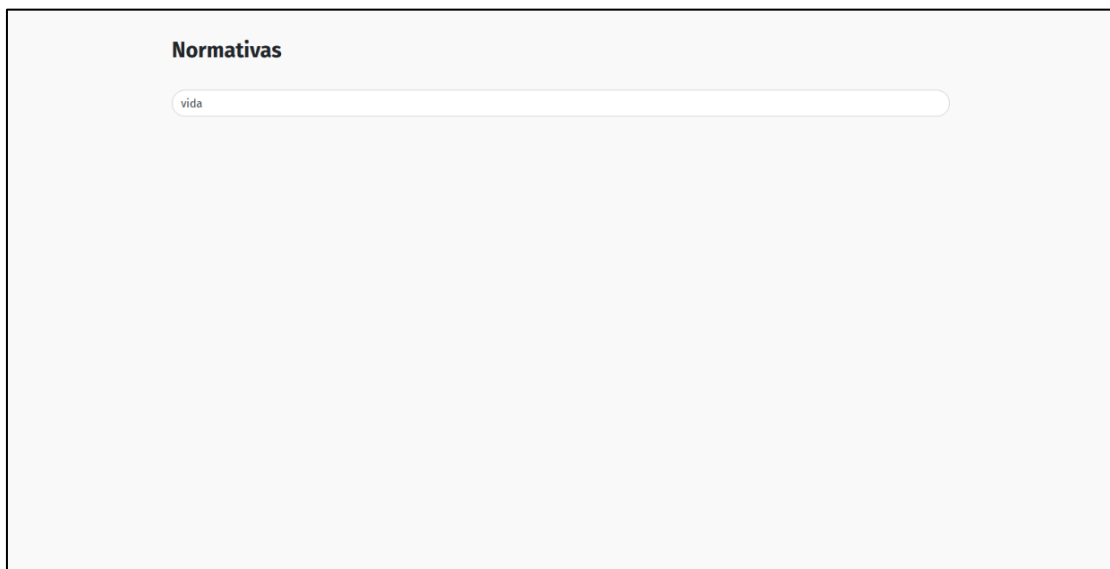
NUEVO DOCUMENTO

Nombre del documento	Fecha	Versión
Listado de normativos		

Fuente: elaboración propia.

Ya teniendo los mockups, se crearon los prototipos de la aplicación utilizando HTML, CSS y JavaScript (Ver figura 27 y 28).

Figura 28. **Prototipo de buscador**



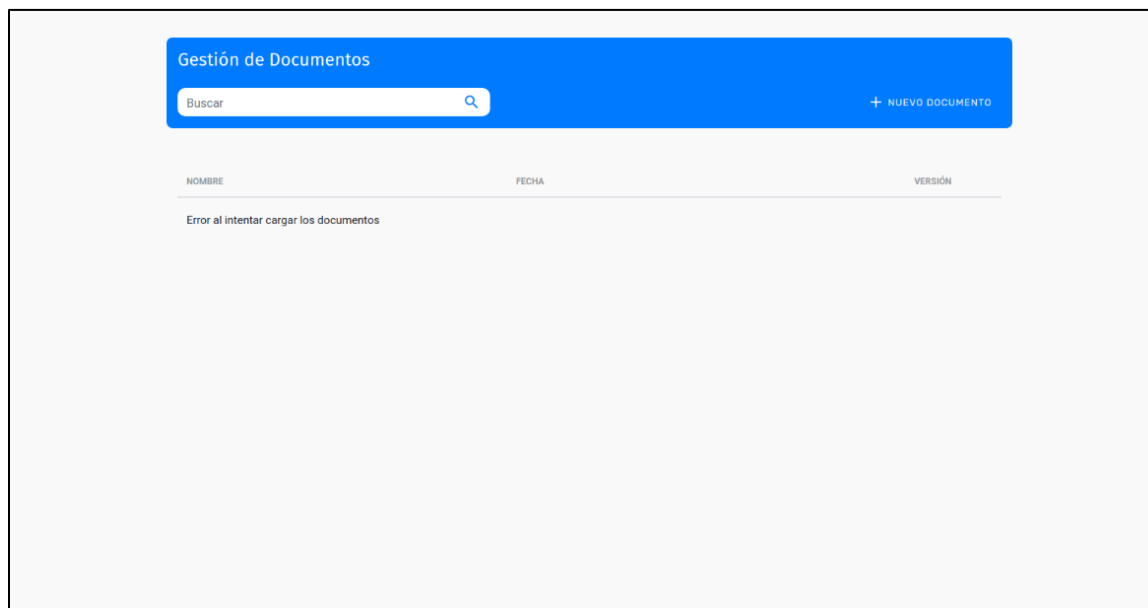
Normativas

vida

This is a search interface for 'Normativas'. It features a title 'Normativas' and a search input field containing the text 'vida'.

Fuente: elaboración propia

Figura 27. **Prototipo repositorio de normativas**



Gestión de Documentos

Buscar + NUEVO DOCUMENTO

NOMBRE	FECHA	VERSIÓN
Error al intentar cargar los documentos		

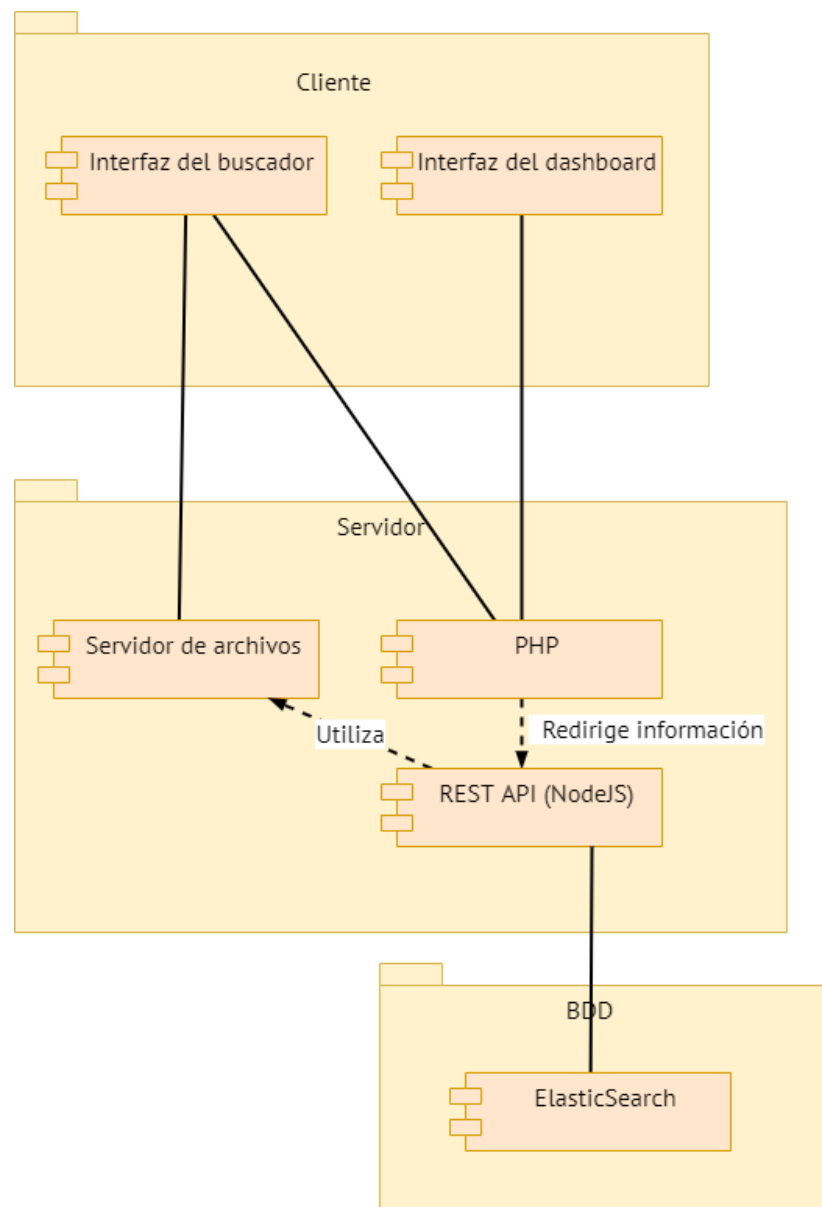
This is a document repository interface titled 'Gestión de Documentos'. It includes a search bar with the placeholder 'Buscar' and a magnifying glass icon, and a button labeled '+ NUEVO DOCUMENTO'. Below the header is a table with columns 'NOMBRE', 'FECHA', and 'VERSIÓN'. The table body contains a single row with the message 'Error al intentar cargar los documentos'.

Fuente: elaboración propia.

2.9.2 Diagrama de componentes

En la figura 29 se presenta el diagrama de componentes de la aplicación.

Figura 29. Diagrama de componentes



Fuente: elaboración propia.

2.10 Pruebas

Durante el transcurso del desarrollo del proyecto se fueron realizando pruebas de rendimiento del software, esta con tal de comprobar la capacidad y calidad del sistema.

2.10.1 Pruebas del buscador

Se enfocaron en comprobar el rendimiento de Elasticsearch. Se realizaron pruebas como la capacidad de almacenamiento o el tiempo de respuesta del software. Las mismas se presentan en las tablas 28-30.

Tabla 28. **Prueba 1 del buscador**

Id de la prueba	TA-01
Descripción	Tiempo de respuesta de la búsqueda por nombre de documento.
Objetivos	Comprobar si el tiempo de búsqueda es rápido y preciso.
Fecha	04/05/2020
Resultados obtenidos	El tiempo promedio de respuesta fue de 7.8 ms al realizar una búsqueda por nombre, que es bastante rápido para una consulta que hace una <i>Full Text Search (stemming, n-grams, etc.)</i>
Acciones realizadas	No aplica.

Fuente: elaboración propia.

Tabla 29. **Prueba 2 del buscador**

Id de la prueba	TA-02
Descripción	Tiempo de respuesta de la búsqueda dentro del texto del documento.
Objetivos	Comprobar si el tiempo de búsqueda es rápido y preciso.
Fecha	11/05/2020
Resultados obtenidos	El tiempo promedio de respuesta fue de 137 ms, este tiempo es bastante mayor al de la prueba anterior, pero hay que tener en cuenta que la cantidad de texto es mucho más vasta. Anteriormente la búsqueda era sobre una frase (nombre del documento), pero en este caso se realiza sobre todo el texto que se encuentra dentro de todos los normativos. La velocidad de búsqueda también dependía bastante de la frase que se quería encontrar, por ejemplo, buscar una palabra es mucho más rápido (60 ms) que hacerlo con una oración (192 ms). Los tiempos de respuesta obtenidos dieron un resultado muy satisfactorio y estos son reproducibles al existir más documentos (ver prueba siguiente).
Acciones realizadas	No aplica.

Fuente: elaboración propia.

Tabla 30. **Prueba 3 del buscador**

Id de la prueba	TA-03
Descripción	Ingresar una gran cantidad de documentos.
Objetivos	Comprobar si el sistema soporta una gran cantidad de documentos sin que existan repercusiones.
Fecha	11/05/2020
Resultados obtenidos	Se ingresaron más de 12000 documentos a Elasticsearch, el tiempo de búsqueda resultaba igual a cuando existen pocos documentos, pero la búsqueda solo es capaz de devolver 10,000 resultados.
Acciones realizadas	Se investigo por qué existía un límite de 10,000 resultados y se descubrió que, aunque Elasticsearch guardara más de 10,000 documentos, la búsqueda solo puede devolver 10,000 resultados. Existen varias formas de corregir esto, por ejemplo, haciendo búsquedas más específicas con filtros (la paginación no se toma en cuenta así que se sigue teniendo un límite de 10,000). Otra forma de resolver esto es cambiando la configuración de ElasticSearch para permitir una mayor cantidad.

Fuente: elaboración propia.

2.10.2 Pruebas del backend (API REST)

Estas pruebas se realizaron para comprobar la lógica del sistema, con el objetivo de encontrar errores en el software para poder resolverlos, además de eso también se hicieron algunas pruebas de rendimiento. Estas se presentan en las tablas 31-34.

Tabla 31. **Prueba 1 para API REST**

Id de la prueba	TB-01
Descripción	Parsear documento pdf por párrafos.
Objetivos	Obtener el texto del pdf separado por párrafos y comprobar que este sea suficientemente bueno para realizar búsquedas sobre él.
Fecha	20/04/2020
Resultados obtenidos	El texto extraído del PDF es bastante similar al que se encuentra en el original. Existen algunos casos en los que el algoritmo utilizado devuelve resultados no deseados, por ejemplo, las listas numeradas (cada elemento de ella será un párrafo separado), la numeración de página y en algunas ocasiones párrafos cortados, pero el resultado es suficientemente bueno para que el buscador encuentre frases utilizando el texto extraído.
Acciones realizadas	Aunque se mencionó que el texto era suficientemente bueno para realizar la búsqueda, se mejoró un poco más el algoritmo para extraer el

	texto (más que todo en párrafos cortados). Las listas enumeradas o la numeración de página no se solucionaron, ya que requerían un algoritmo muy avanzado, y es algo que no influye en la búsqueda del texto.
--	---

Fuente: elaboración propia.

Tabla 32. **Prueba 2 API REST**

Id de la prueba	TB-02
Descripción	Tiempo de ingreso del documento
Objetivos	Obtener cuanto tiempo se tarda en ingresar un nuevo documento.
Fecha	11/05/2020
Resultados obtenidos	Como se esperaba el tiempo de ingresar un documento es mucho más elevado que el de búsqueda, llegando a ser hasta 2 s y esto es porque además de ingresar una gran cantidad de datos a Elasticsearch también se debe extraer el texto del PDF y separarlo por párrafos. El tiempo de inserción difiere dependiendo del tamaño del documento que se ingrese, es decir, es mucho más rápido ingresar un documento de 10 páginas que uno de 500.
Acciones realizadas	Se optimizo el algoritmo de parseo para disminuir el tiempo de ingreso de documentos.

Fuente: elaboración propia.

Tabla 33. Prueba 3 para API REST

Id de la prueba	TB-03
Descripción	Ingreso y eliminación de documentos.
Objetivos	Comprobar que el ingreso y la eliminación de los documentos funcione de manera correcta.
Fecha	No aplica.
Resultados obtenidos	Esta prueba se realizó periódicamente para comprobar que el código siguiera funcionando correctamente. En múltiples ocasiones ayudo a encontrar errores en el código que fueron arreglados en versiones posteriores.
Acciones realizadas	Se arreglaron los errores que se encontraban durante las pruebas.

Fuente: elaboración propia.

Tabla 34. **Prueba 4 para API REST**

Id de la prueba	TB-04
Descripción	Búsqueda de información
Objetivos	Comprobar que la búsqueda siga funcionando correctamente.
Fecha	No aplica.
Resultados obtenidos	Esta prueba se realizó periódicamente para comprobar que el código siguiera funcionando correctamente. En múltiples ocasiones ayudo a encontrar errores en el código que fueron arreglados en versiones posteriores.
Acciones realizadas	Se arreglaron los errores que se encontraban durante las pruebas.

Fuente: elaboración propia.

2.11 Entrega

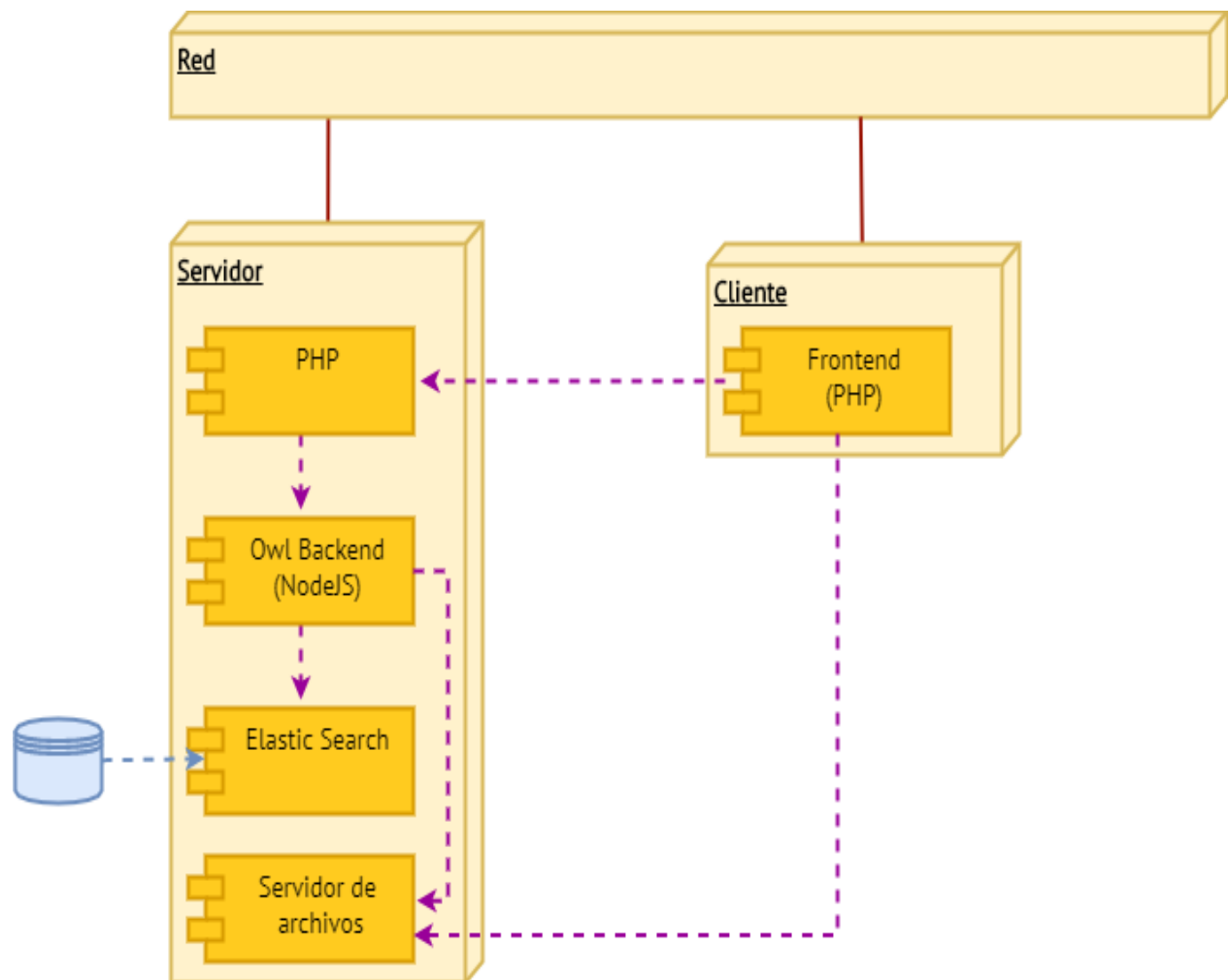
Como se mencionó anteriormente, existe algo peculiar sobre la arquitectura cliente-servidor que se utilizó y es que el servicio *RESTful* funciona únicamente en un ambiente local y se utiliza PHP para realizar la comunicación entre el cliente y el *backend* de NodeJS. Este archivo de PHP no es exactamente el mismo que el *frontend*, aunque funciona en la misma aplicación, es decir que un archivo de PHP (*frontend*), mediante el uso de JavaScript, se comunicará con otro archivo PHP, este archivo PHP funcionará como una API Web cuyo objetivo es comunicarse con el API REST de NodeJS que se encuentra localmente.

Esto se creó así ya que, aunque NodeJS fue una tecnología muy importante para el desarrollo del proyecto, el portal de COOSAJÓ R.L. está desarrollado sobre PHP, por lo que la autenticación de usuarios, servicios de autorización y rutas protegidas se encuentran en PHP, así que por seguridad de la aplicación se decidió que únicamente se pueda acceder localmente al servicio RESTful de NodeJS, lo cual PHP puede hacer porque se encuentran en el mismo servidor. De esta manera PHP (portal de COOSAJÓ R.L.) le puede decir al servicio RESTful de NodeJS quien es el usuario que está realizando la petición y si este tiene la autorización necesaria para realizar la acción solicitada.

La aplicación se alojó en el servidor de desarrollo de COOSAJÓ R.L., que es un servidor NGINX con sistema operativo Ubuntu 18.04, a esta se le otorgó el nombre de *OWL*. Esta se integró con el portal de COOSAJÓ R.L., colocando una barra de búsqueda en la parte superior del portal, la cual es visible desde cualquier ubicación ya que de esta forma los colaboradores pueden realizar sus consultas en cualquier momento.

Todo esto está descrito en el diagrama de la figura 30.

Figura 30. Diagrama de despliegue de la aplicación



Fuente: elaboración propia.

2.12 Lista de riesgos

Siempre existe la posibilidad que un software ejecute acciones o tareas de manera errónea, afectando el funcionamiento del sistema. Aunque esto sea una ocurrencia muy poco probable siempre existe el riesgo que esto suceda.

En el sistema se identificaron los riesgos presentados en las tablas 32-38.

Tabla 35. **Riesgo de acciones no autorizadas**

Acciones no autorizadas	
Descripción	
Usuario realiza acciones para las que no tiene autorización, por ejemplo, ingresar o eliminar normativos.	
Probabilidad	Nivel de riesgo
Muy Baja	Alto
Plan de contingencia	
El API REST se encuentra localmente en el servidor por lo que nadie puede acceder a esta, excepto un <i>backend</i> de PHP, el cual utiliza los métodos de autorización proporcionados por el portal de COOSAJO R.L.	

Fuente: elaboración propia.

Tabla 36. **Riesgo acceder a Elasticsearch sin autorización**

Acceder a Elasticsearch sin autorización	
Descripción	
Acceder a Elasticsearch sin autorización.	
Probabilidad	Nivel de riesgo
Muy Baja	Alto
Plan de contingencia	
ElasticSearch también funciona localmente, por lo que la única forma de ingresar datos a este es mediante el API REST.	

Fuente: elaboración propia.

Tabla 37. **Riesgo de guardar más de diez mil documentos**

Guardar más de diez mil documentos	
Descripción	
ElasticSearch tiene un límite sobre la cantidad de documentos por búsqueda.	
Probabilidad	Nivel de riesgo
Muy Baja	Muy bajo
Plan de contingencia	
ElasticSearch permite cambiar el límite de la cantidad de documentos máximos por búsqueda o petición (a cambio de perder un poco de rendimiento).	

Fuente: elaboración propia.

Tabla 38. **Riesgo de error a la mitad de una transacción de guardado**

Error a la mitad de una transacción de guardado	
Descripción	
ElasticSearch no tiene soporte para transacciones ACID, por lo que cualquier error que ocurra durante una transacción hará perder la integridad de la base de datos.	
Probabilidad	Nivel de riesgo
Baja	Alto
Plan de contingencia	
Ya que ElasticSearch no soporta este tipo de transacciones se intenta mitigar este problema guardando los elementos por prioridad, es decir, primero se guarda el PDF en el servidor de archivos, luego se extrae el texto de este, a continuación, se ingresan los datos en el índice de <i>metadata</i> y por último en el índice <i>main</i> .	

Fuente: elaboración propia.

2.13 Gestión de cambios

En las tablas 39-44 se muestran los cambios realizados al proyecto.

Tabla 39. **Gestión de cambios 01**

Solicitante	Darwin Pinto
Fecha	15/05/2020
Prioridad	Media
Descripción	Poder seleccionar si el documento es interno o externo al momento de ingresarlo al sistema.
Acción realizada	Se agregó esta funcionalidad al sistema.

Fuente: elaboración propia.

Tabla 40. **Gestión de cambios 02**

Solicitante	Julio Padilla
Fecha	25/05/2020
Prioridad	Media
Descripción	Mostrar estadísticas del uso de la aplicación en una sección de <i>dashboard</i> .
Acción realizada	Se modificó la sección de <i>dashboard</i> para que muestre estadísticas del uso del sistema y cuales documentos son los más visualizados. Además, se agregó una sección al menú de cada documento para ver las estadísticas de ese documento específico.

Fuente: elaboración propia.

Tabla 41. **Gestión de cambios 03**

Solicitante	Julio Padilla
Fecha	25/05/2020
Prioridad	Alta
Descripción	Al mostrar todos los normativos ingresados, únicamente presentar la versión más reciente de cada uno.
Acción realizada	Se implementó este cambio al sistema.

Fuente: elaboración propia.

Tabla 42. **Gestión de cambios 04**

Solicitante	Darwin Pinto
Fecha	03/06/2020
Prioridad	Media
Descripción	Agregar barra de búsqueda en la parte superior del portal de COOSAJÓ R.L.
Acción realizada	Se agregó esta funcionalidad al sistema.

Fuente: elaboración propia.

Tabla 43. **Gestión de cambios 05**

Solicitante	Darwin Pinto
Fecha	03/06/2020
Prioridad	Media
Descripción	Abrir resultados de búsqueda en un <i>modal</i> .
Acción realizada	Se agregó esta funcionalidad al sistema.

Fuente: elaboración propia.

.

Tabla 44. **Gestión de cambios 06**

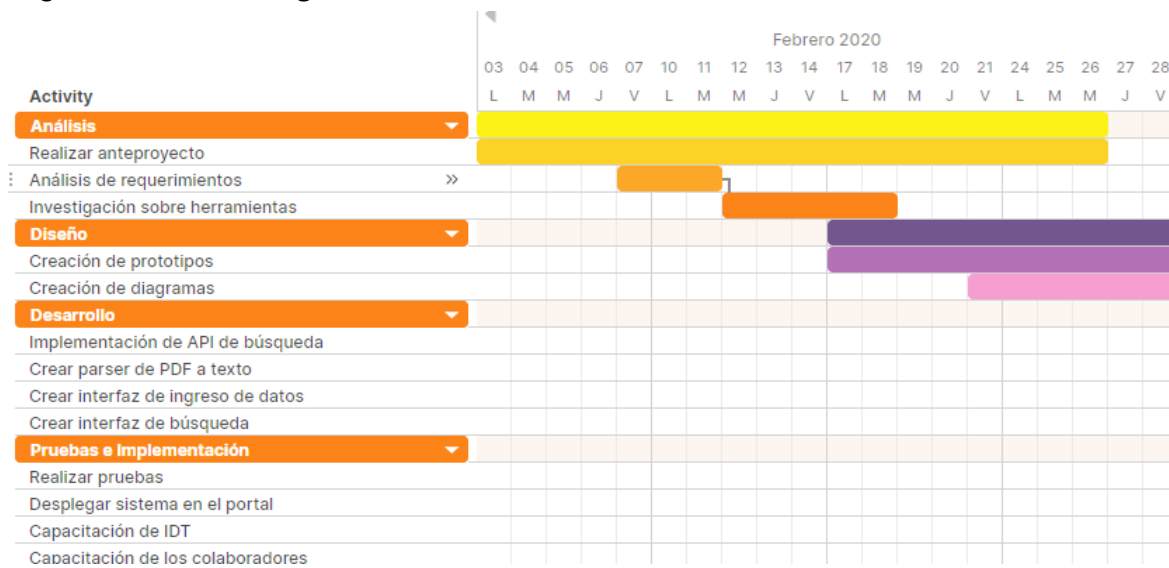
Solicitante	Darwin Pinto
Fecha	03/06/2020
Prioridad	Alta
Descripción	Mostrar todos los normativos existentes en una sección pública para que los colaboradores puedan visualizarlos.
Acción realizada	Se utilizó la primera versión del <i>dashboard</i> , para crear una sección de documentos públicos, en el que únicamente se pueden visualizar todos los documentos (solo se muestra la última versión de estos).

Fuente: elaboración propia.

2.14 Administración del proyecto

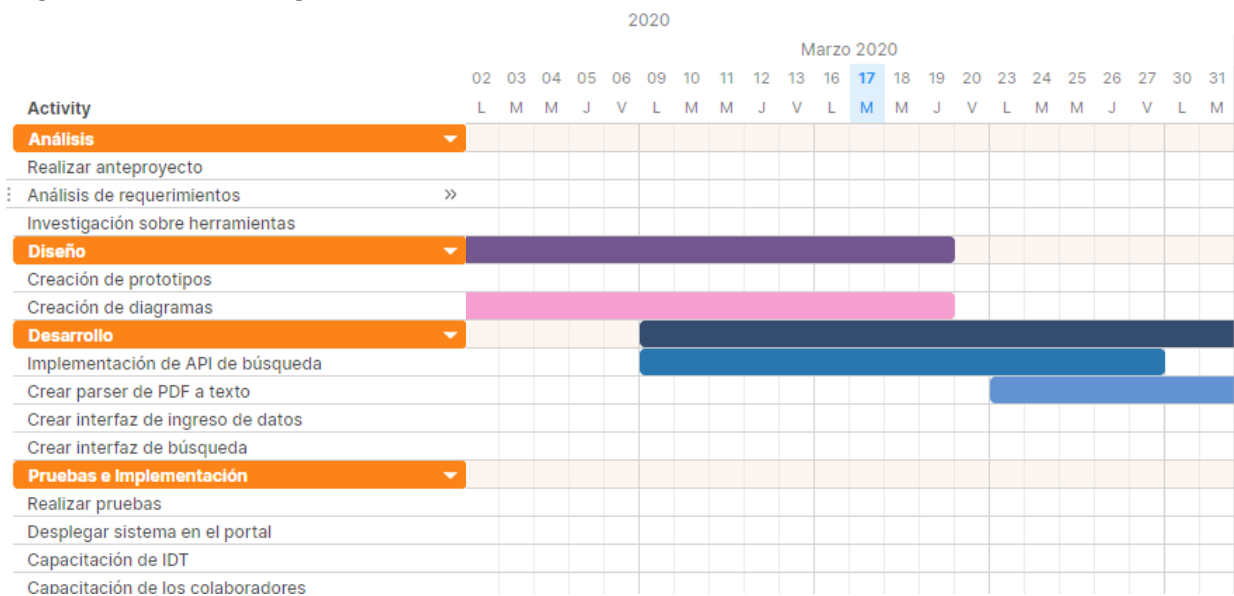
2.14.1 Cronograma de actividades inicial

Figura 31. Cronograma de actividades inicial 1



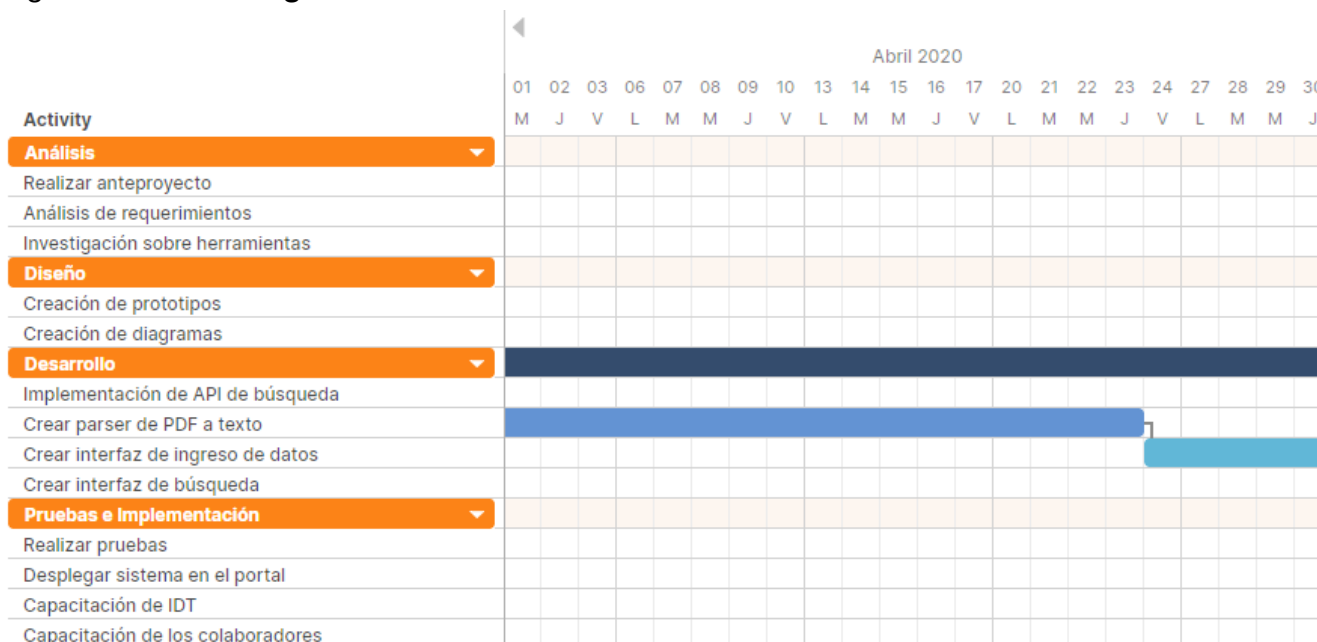
Fuente: elaboración propia.

Figura 32. Cronograma de actividades inicial 2



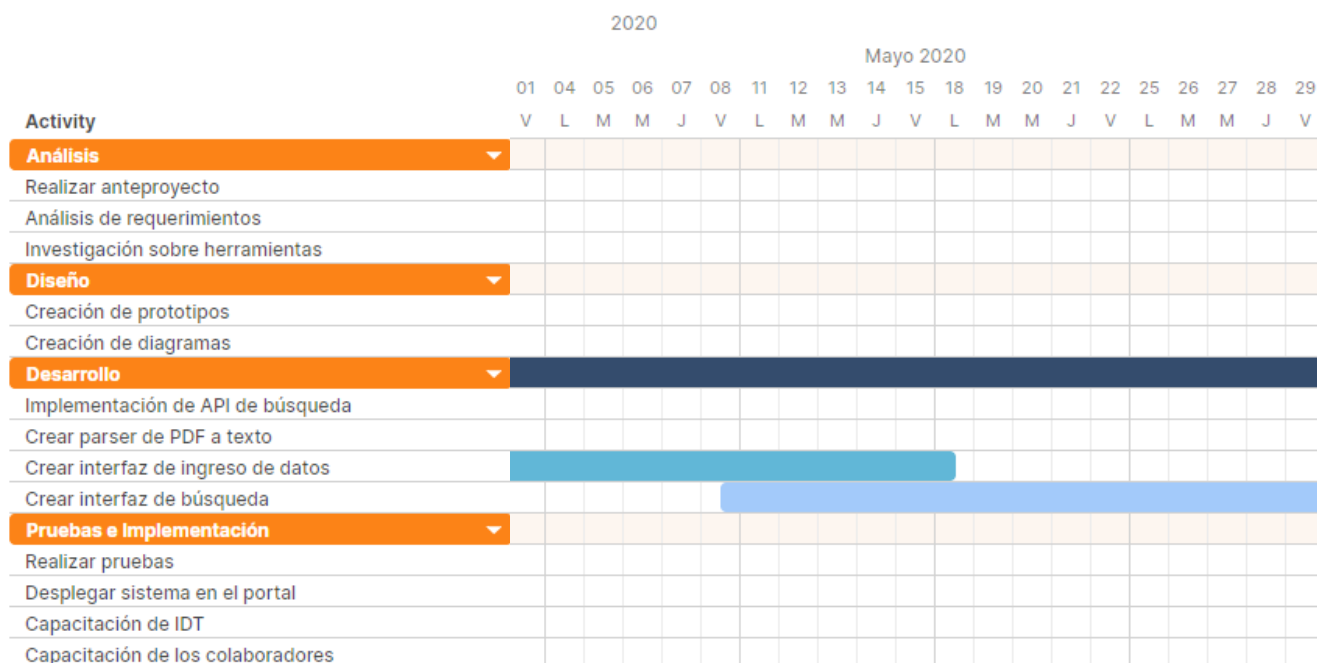
Fuente: elaboración propia.

Figura 34. Cronograma de actividades inicial 3



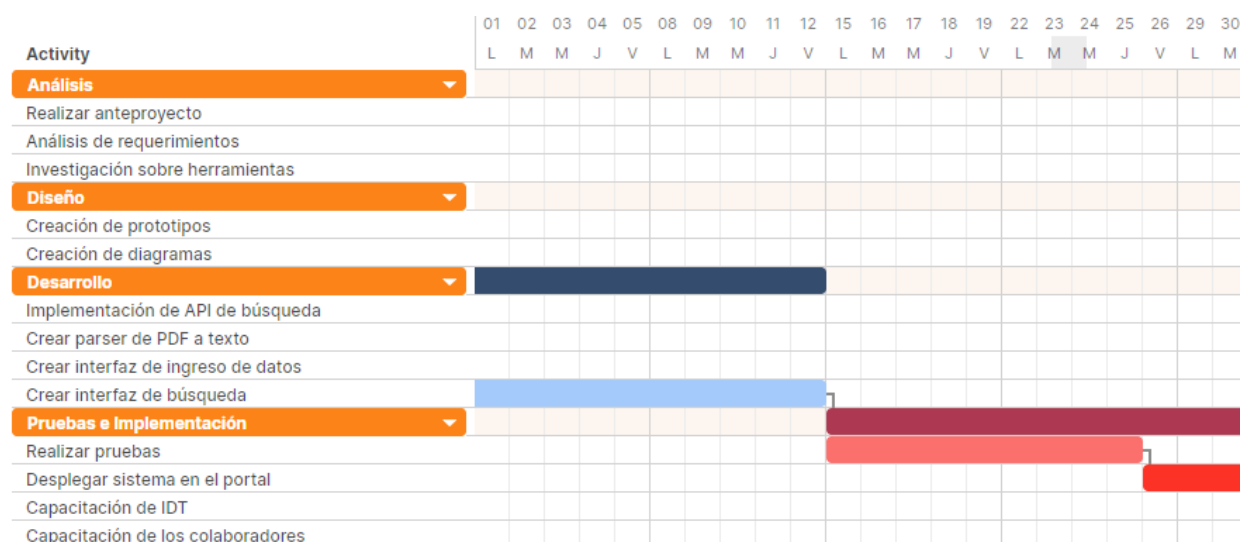
Fuente: elaboración propia.

Figura 33. Cronograma de actividades inicial 4.



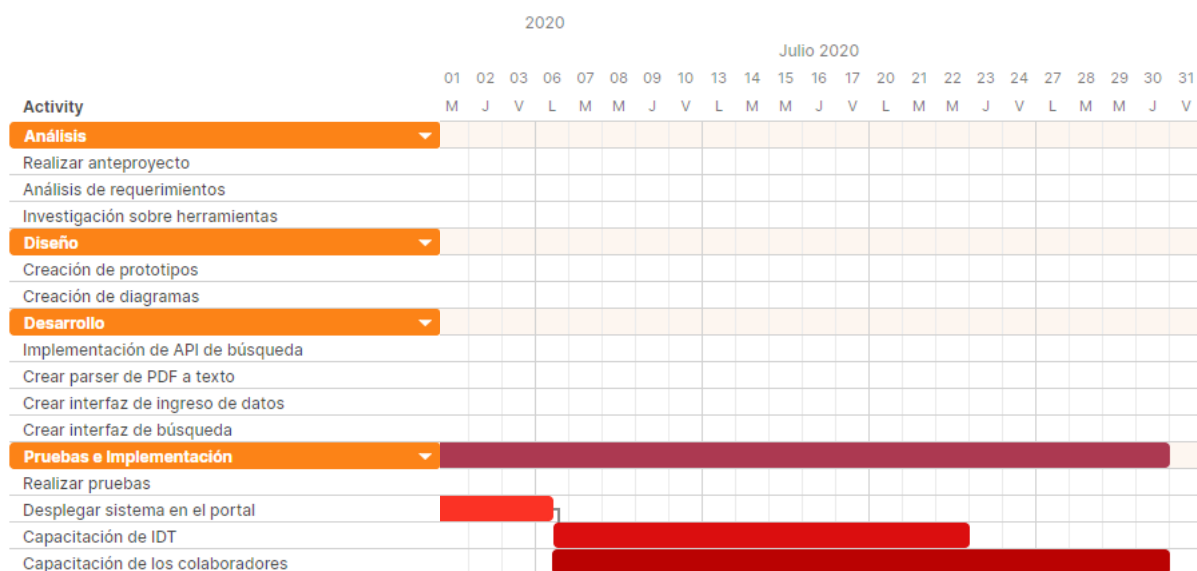
Fuente: elaboración propia.

Figura 35. Cronograma de actividades inicial 5



Fuente: elaboración propia.

Figura 36. Cronograma de actividades inicial 6

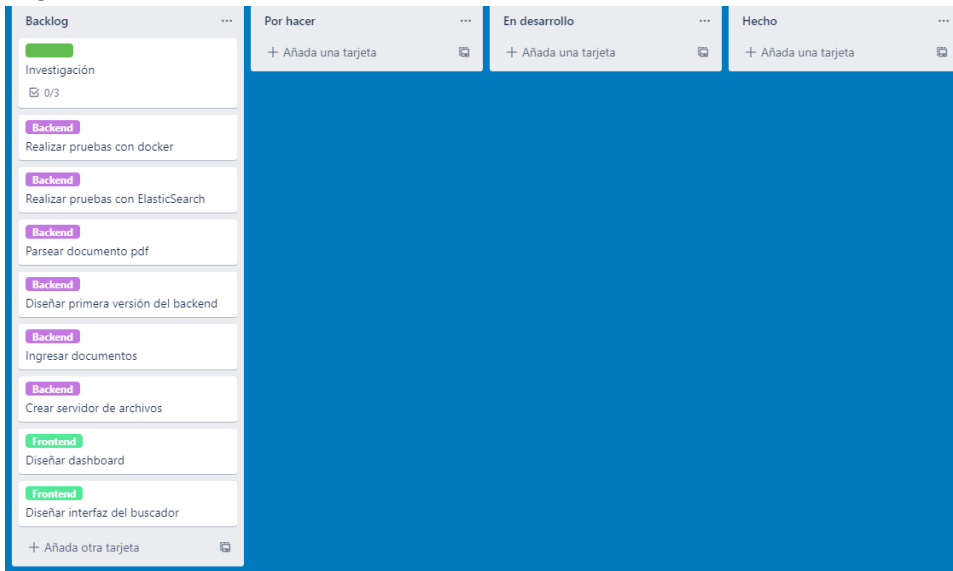


Fuente: elaboración propia.

2.14.2 Tablero Kanban

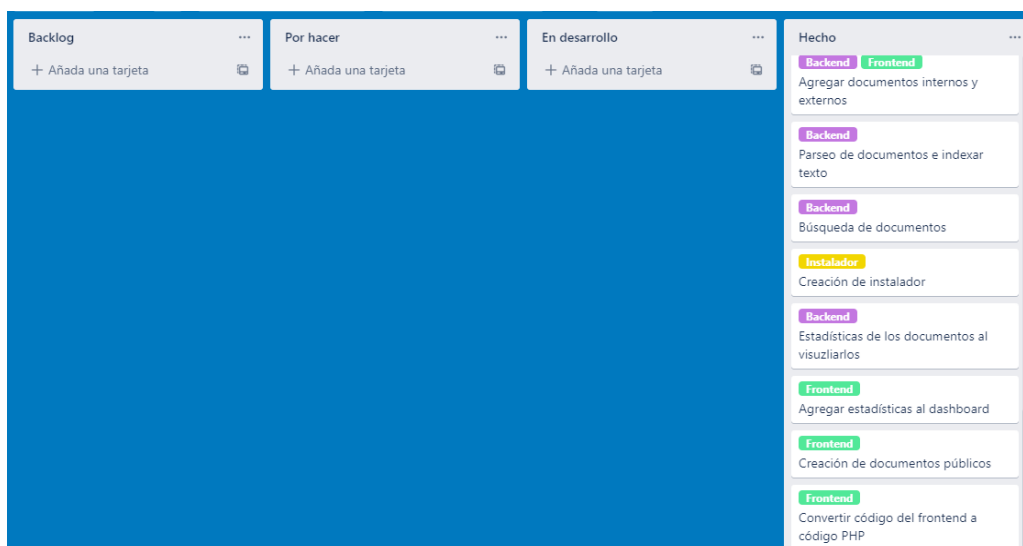
Para llevar el control de desarrollo de los diferentes artefactos que se desarrollaron se utilizó Kanban (Ver figura 37 y 38 y la tabla 45).

Figura 37. Tablero Kanban Inicial



Fuente: elaboración propia.

Figura 38. Tablero Kanban final



Fuente: elaboración propia.

Tabla 45. **Tarjetas Kanban utilizadas**

Nombre	Tipo
Investigación	Otros
Realizar pruebas con Docker	<i>Backend, Pruebas</i>
Realizar pruebas con Elasticsearch	<i>Backend, Pruebas</i>
Parsear documento PDF	<i>Backend, Pruebas</i>
Diseñar primera versión del backend	<i>Backend</i>
Crear servidor de archivos	<i>Backend</i>
Ingresar documentos	<i>Backend</i>
Diseñar interfaz de dashboard	<i>Frontend</i>
Interfaz menú de documento	<i>Frontend</i>
Crear tabla de documentos	<i>Frontend</i>
Agregar versionado de documentos	<i>Frontend, backend</i>
Parseo de documentos	<i>Backend</i>
Indexar texto parseado	<i>Backend</i>
Búsqueda de documentos	<i>Backend</i>
Interfaz del buscador	<i>Frontend</i>
Agregar documentos internos y externos	<i>Frontend, backend</i>
Creación del instalador	Instalador
Agregar estadísticas al dashboard	<i>Frontend</i>
Creación de documentos públicos	<i>Frontend</i>
Estadísticas de los documentos al visualizarlos	<i>Backend</i>
Convertir código del frontend a código PHP	<i>Frontend</i>
Refactorización del backend	<i>Backend</i>
Refactorización del instalador	Instalador
Creación de backend PHP que funcione como Proxy	<i>Backend</i>

Fuente: elaboración propia.

2.15 Ambiente

2.15.1 Tecnologías de desarrollo

El sistema se desarrolló haciendo el uso de las siguientes tecnologías:

- NodeJS 12.16.2
- Elasticsearch 7.6.2
- PHP 7.2
- JavaScript
- HTML 5
- CSS3

a. Módulos de NodeJS

- Express, para construir el servidor http.
- Axios, para realizar peticiones.
- Pdf-parser, para extraer el texto del pdf.
- Elastic client, para poder utilizar Elasticsearch con NodeJS.
- Morgan, para la visualización de las peticiones al servidor.
- Mocha, para la creación de pruebas.
- Dot env, para el manejo de las variables de entorno.

b. Módulos de PHP

- CURL, para realizar peticiones.

c. Módulos de JavaScript

- Axios, para realizar peticiones.
- Chart.js, para la creación de gráficas.
- Bootstrap 4, para el *frontend*.

2.15.2 Herramientas de apoyo

- Trello, para llevar control de Kanban.
- Draw.io, para la creación de diagramas.
- VSCode, como editor de código.
- GitLab, como repositorio y control de versiones.

2.15.3 Ambiente de desarrollo

- Sistema Operativo Ubuntu 18.04
- Sistema Operativo Windows 10 Pro
- 16GB de Memoria RAM
- Procesador Intel Core i5 4460
- 120 GB SSD
- 2TB HDD
- Tarjeta Gráfica GTX 970
- NGINX cómo servidor de archivos

2.15.4 Ambiente de producción

- Sistema Operativo Ubuntu 18.04
- NGINX cómo servidor de archivos

2.16 Valoración económica del proyecto

En la tabla 46 se presentan los roles involucrados en la creación del proyecto.

Tabla 46. Roles involucrados en la creación del proyecto

Rol	Descripción
Arquitecto de software (AS)	Se encarga de definir los requerimientos del proyecto. Además de seleccionar las tecnologías más adecuadas para el desarrollo del proyecto.
Desarrollador senior (DeS)	Se encarga de desarrollar el software, logrando que se cumplan todos los requerimientos dados.
<i>Project Manager</i> (PM)	Se encarga de gestionar el proyecto realizando la planificación de este y utilizando la metodología más adecuada para el desarrollo del proyecto.
Desarrollador ElasticSearch (DeES)	Desarrollo de software basado en la tecnología que proporciona Elasticsearch.
Consultor de Sistemas (CS)	Se encarga de la comunicación entre los interesados del proyecto y el equipo de desarrollo.

Fuente: elaboración propia.

En las tablas 47-51 se presentan las actividades que se realizaron en el proyecto y las tablas 52 y 53 describen los costos del proyecto.

Tabla 47. **Actividades realizadas arquitecto de software**

Descripción	Horas	Costo por hora	Rol
Definición de los requerimientos	32	Q100. ⁰⁰	Arquitecto de software
Investigación de las tecnologías a utilizar	40		
Análisis y Diseño	40		
Configuración del entorno de desarrollo	8		
Establecer riesgos	8		
Definición de las pruebas	16		
Despliegue de la aplicación	32		
Total	176		

Fuente: elaboración propia.

Tabla 48. **Actividades realizadas desarrollador senior**

Descripción	Horas	Costo por hora	Rol
Codificación de los prototipos	40	Q75. ⁰⁰	Desarrollador Senior
Desarrollo del <i>backend</i>	250		
Desarrollo del <i>frontend</i>	150		
Desarrollo del instalador	150		
Implementación de pruebas	40		
Total	630		

Fuente: elaboración propia.

Tabla 49. **Actividades realizadas desarrollador ElasticSearch**

Descripción	Horas	Costo por hora	Rol
Desarrollo ElasticSearch	100	Q350. ⁰⁰	Desarrollador ElasticSearch
Total	100		

Fuente: elaboración propia.

Tabla 50. **Actividades realizadas consultor de sistemas**

Descripción	Horas	Costo por hora	Rol
Elaboración de casos de Uso	24	Q110. ⁰⁰	Consultor de sistemas
Capacitación de usuarios	8		
Total	32		

Fuente: elaboración propia.

Tabla 51. **Actividades realizadas *project manager***

Descripción	Horas	Costo por hora	Rol
Definición de la metodología de trabajo	24	Q125. ⁰⁰	<i>Project Manager</i>
Planificación semanal	192		
Total	216		

Fuente: elaboración propia.

Tabla 52. Costo del recurso humano

Rol	Salario por hora	Horas realizadas	Subtotal
Arquitecto de software	Q 100. ⁰⁰	176	Q 17,600. ⁰⁰
Desarrollador senior	Q 75. ⁰⁰	630	Q 47,250. ⁰⁰
Desarrollador ElasticSearch	Q 350. ⁰⁰	100	Q 35,000. ⁰⁰
Consultor de software	Q 110. ⁰⁰	32	Q 3,520. ⁰⁰
<i>Project Manager</i>	Q 125. ⁰⁰	216	Q 27,000. ⁰⁰
	Total	1154	Q 130,370. ⁰⁰

Fuente: elaboración propia.

Tabla 53. Costo total del proyecto

	Costos fijos	Costos variables		Total
		Costo Unitario	Cantidad	
Costos directos				
Mano de obra	Q 130,370. ⁰⁰			Q 130,370. ⁰⁰
Costos indirectos				
Energía Eléctrica		Q300. ⁰⁰ /mes	7 meses	Q 2,100. ⁰⁰
Internet		Q350. ⁰⁰ /mes	7 meses	Q 2,450. ⁰⁰
Viáticos		Q500. ⁰⁰ /mes	3 meses	Q 1500. ⁰⁰
			Total	Q 136,420. ⁰⁰

Fuente: elaboración propia.

3. FASE DE ENSEÑANZA APRENDIZAJE

3.1 Capacitación para uso de software

3.1.1 Justificación

A haber utilizado tecnologías modernas, nació la necesidad de una capacitación para la parte técnica de la aplicación. Esta se impartió al área de IDT y abarcó el uso de la API REST, el instalador de aplicaciones y que pasos seguir para poder extenderlo.

También fue necesario enseñar el uso del sistema desde el punto de vista no técnico, es decir, la forma en la que los usuarios finales utilizarán el sistema.

3.1.2 Objetivo

- Facilitar a los colaboradores el uso del buscador de normativas, para que sea utilizado de manera efectiva.
- Lograr que se desarrollen futuras herramientas utilizando la misma API de búsqueda.

3.1.3 Personal objetivo

Los colaboradores del área de Investigación y Desarrollo, capacitando un total de 3 técnicos.

3.1.4 Documentación

- Manual de usuario: Este indica el uso de la plataforma, mostrando cómo gestionar los documentos y como realizar las búsquedas de estos.
- Manual de instalación: Explica la forma en la que está organizado el sistema, los distintos componentes y la forma en la que este se debe instalar.
- Manual técnico: Muestra la forma de extender la API de instalación para poder crear fácilmente más proyectos utilizando Elasticsearch.
- Video de instalación: Da a conocer todo el proceso de instalación del sistema.
- Video de uso del sistema: Expresa las distintas funcionalidades del sistema y como utilizar cada una de ellas.
- Documentación digital: Servicio web que muestra todos los puntos de salida de la API REST.

3.1.5 Estructura del plan de capacitación

Se realizó dos sesiones de capacitación para explicar el aspecto técnico de la aplicación. La tabla 54 muestra los aspectos que se cubrieron en cada presentación.

Para capacitar al usuario final sobre el uso de la aplicación se entregaron los manuales y un vídeo de capacitación sobre el uso del sistema, el cual pueden consultar en cualquier momento. Estos cubren el gestionamiento de las normativas, uso del *dashboard* y buscador.

Tabla 54. Estructura del plan de capacitación técnica

Tema	Duración
Introducción	15 min
Conceptos de un motor de búsqueda	20 min
Primeros pasos	15 min
Uso del instalador y como modificarlo	45 min
Como funciona una API REST	10 min
Conectar Elasticsearch y NodeJS	30 min

Fuente: elaboración propia.

CONCLUSIONES

1. Se desarrolló una API REST que permite la gestión de un repositorio de normativos, la búsqueda de estos y obtener estadísticas del uso del software, esta API cuenta con un instalador que puede ser extendido para crear nuevas aplicaciones.
2. Se desarrolló un motor de búsqueda, con ayuda de ElasticSearch, que permite una búsqueda de texto completa (*Full Text Search*) sobre el contenido que se encuentra dentro de los normativos, generando resultados satisfactorios para el usuario.
3. Se capacitó a 3 colaboradores de IDT sobre los aspectos técnicos del sistema, como el proceso de instalación del software en un servidor, el funcionamiento del sistema y los puntos de partida sobre los cuales pueden iniciar para poder extender el software.

RECOMENDACIONES

1. Realizar copias de respaldo de manera periódica, tanto de Elasticsearch, como de la carpeta que contiene los documentos PDF alojados en el servidor.
2. Se sugiere implementar la herramienta Kibana, para aplicarla como un *dashboard* de visualización de datos, lo cual sería de utilidad para aplicar acciones de recuperación de errores de transacción sobre la base de datos Elasticsearch.
3. Utilizar el API implementada para la creación de proyectos que utilicen Elasticsearch en nuevas tecnologías que utilicen Big Data.
4. Tomar en cuenta las gráficas del *dashboard* como herramienta para toma de decisiones para mejorar el conocimiento de los normativos por parte de los colaboradores de la institución.
5. Mantener Elasticsearch y NodeJS en un ambiente local, es decir, en el mismo servidor, en caso de necesitar separarlas, tomar en cuenta mecanismos de seguridad para la comunicación de datos.



REFERENCIAS BIBLIOGRÁFICAS

Acerca de Node.js. (s.f.). *En NodeJS [Pág. Web]*. Recuperado de <https://nodejs.org/es/about/>

Ariza Ladino, C.F. (2012). *Sistema de recuperación de información utilizando índices combinados de términos y extracción de información (Tesis de maestría)*. Universidad Nacional de Colombia, Facultad de Ingeniería, Bogotá, Colombia.

Beth Lovins, J. (1968). Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics*, 11(1-2), 22-31. Recuperado de <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.231.1109&rep=rep1&type=pdf>

Castrillón Estrada, J. A., García Domínguez, J. C., Anaya Taboada, M., Rodríguez Berdugo, D., De la Rosa Barranco, D., & Caballero Uribe, C. V. (2008). *Bases de datos, motores de búsqueda e índices temáticos: herramientas fundamentales para el ejercicio médico*. Salud Uninorte, 24(1), 95-119

Cooperativa de Ahorro y Crédito Integral San José Obrero R.L. (2018). *Quienes somos*. *En COOSAJO R.L. [Pág. Web]*. Recuperado de https://coosajo.com/website/?page_id=8163



Google. (s.f.) [*Edificio Oficinas Corporativas "Nuestra Casa" Coosajo R.L.*]. Recuperado de [https://www.google.com/maps/place/Edificio+ Oficinas+ Corporativas+%22Nuestra+Casa%22+Coosajo,+R.+L./@14.5603682,-89.3285568,15.75z/data=!4m9!1m2!2m1!1soficinas+corporativas+esquipulas!3m5!1s0x0:0xd9345fab2b6e7dc2!8m2!3d14.5617102!4d-89.3187994!15sCiBvZmljaW5hcyBjb3Jwb3JhdGl2YXMgZXNxdWlwdWxhc1o5ChVvZmljaW5hcyBjb3Jwb3JhdGl2YXMiIG9maWNpbmFzIGNvcnBvcnF0aXZhcyBlc3F1aXB1bGFzkgERZ292ZXJubWVudF9vZmZpY2U](https://www.google.com/maps/place/Edificio+Oficinas+Corporativas+%22Nuestra+Casa%22+Coosajo,+R.+L./@14.5603682,-89.3285568,15.75z/data=!4m9!1m2!2m1!1soficinas+corporativas+esquipulas!3m5!1s0x0:0xd9345fab2b6e7dc2!8m2!3d14.5617102!4d-89.3187994!15sCiBvZmljaW5hcyBjb3Jwb3JhdGl2YXMgZXNxdWlwdWxhc1o5ChVvZmljaW5hcyBjb3Jwb3JhdGl2YXMiIG9maWNpbmFzIGNvcnBvcnF0aXZhcyBlc3F1aXB1bGFzkgERZ292ZXJubWVudF9vZmZpY2U)

Higuerey, E. (17 de febrero de 2020). NGINX: entiende cómo funciona este servidor web y cómo se diferencia a Apache. *En Rockcontent [Blog]*. Recuperado de <https://rockcontent.com/es/blog/nginx/>

Massé, M. (2011). *REST API: Design Rulebook*. Estados Unidos de América: O'Reilly Media, Inc.

Palabra vacía. (2019). *En Wikipedia [Enciclopedia en línea]*. Recuperado de https://es.wikipedia.org/wiki/Palabra_vac%C3%ADa

¿Qué es JavaScript? (2021). *En MDN Web Docs [Blog de Mozilla]*. Recuperado de https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript

Qué es Kanban: definición, características y ventajas. (s.f.). *En Kanbanize [Pág. Web]*. Recuperado de <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban/>



¿Qué es PHP? (s.f.). *En PHP [Pág. Web]*. Recuperado de <https://www.php.net/manual/es/intro-what-is.php>

RUP Guía para Novatos. (s.f.). *En Rupnoobs [Blog de Wordpress.com]*. Recuperado de <https://rupnoobs.wordpress.com/>

Sommerville, I. (2011). *Ingeniería del Software* (9 ed.). España: Pearson.

Welcome to Apache Lucene. (s.f.). *En Apache Lucene [Pág. Web]*. Recuperado de <https://lucene.apache.org/>

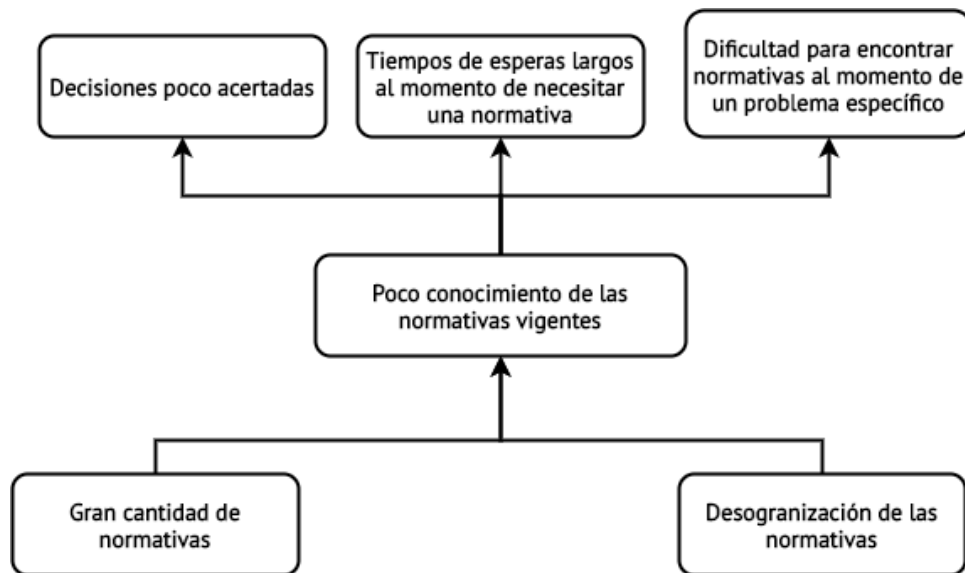
What is Elasticsearch? (s.f.). *En Elastic [Pág. web]*. Recuperado de <https://www.elastic.co/es/what-is/elasticsearch>



APÉNDICES

- 1. Árbol de problema**
- 2. Manual de usuario**
- 3. Manual de instalación**
- 4. Manual técnico API**
- 5. Contenido del CD**

Apéndice 1. **Árbol de problema: Creación de un sistema de información para acceso a la documentación de normativas para la Cooperativa de Ahorro y Crédito**



Fuente: elaboración propia.

Apéndice 2. **Manual de usuario**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE ORIENTE
CARRERAS DE INGENIERÍA

**“MANUAL DE USUARIO PARA MOTOR DE BÚSQUEDA DE LAS
NORMATIVAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO
INTEGRAL SAN JOSÉ OBRERO (COOSAJÓ, R.L.) DEL MUNICIPIO
DE ESQUIPULAS, CHIQUIMULA”**

JOSÉ EDUARDO GUERRA SOSA

OCTUBRE, 2020

Empezando

Owl cuenta con tres interfaces o páginas web:

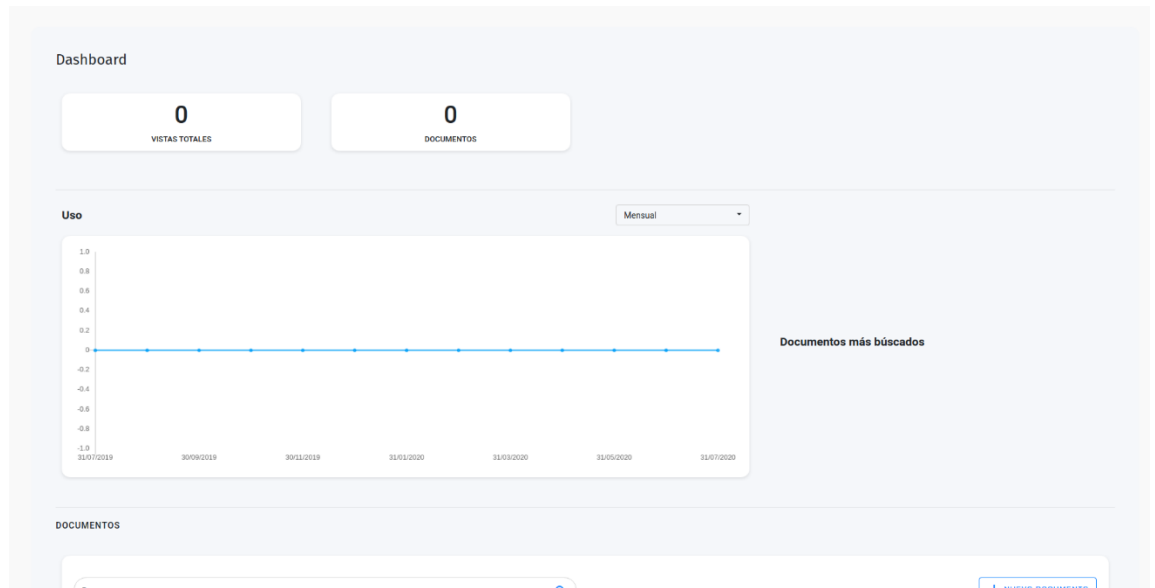
- **Dashboard** En esta interfaz los administradores podrán subir los documentos, agregar versiones y ver estadísticas del uso de la aplicación con las cuales podrán tomar decisiones.
- **Buscador** En esta interfaz los usuarios podrán realizar búsquedas sobre el texto que se encuentran en los documentos.
- **Documentos Públicos** En esta interfaz se mostrarán todos los documentos que se encuentran ingresados, para el usuario final (que no es administrador)

Nota: El manual asume que se está ejecutando el software en un ambiente localhost, de no ser así se debe cambiar en las URL, el localhost por la IP o nombre de dominio correspondiente.

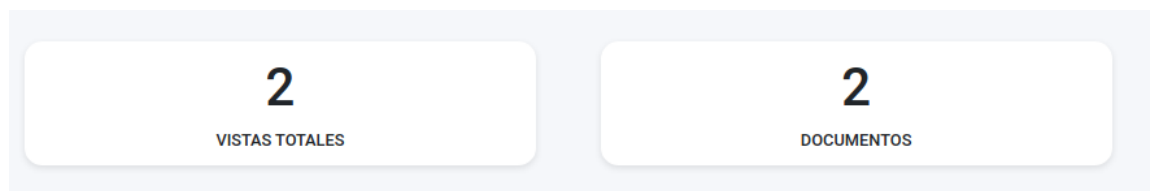
Dashboard

Para ingresar a esta sección se debe ir a la siguiente URL en el navegador

<http://localhost/dashboard.php>



1. Información Básica



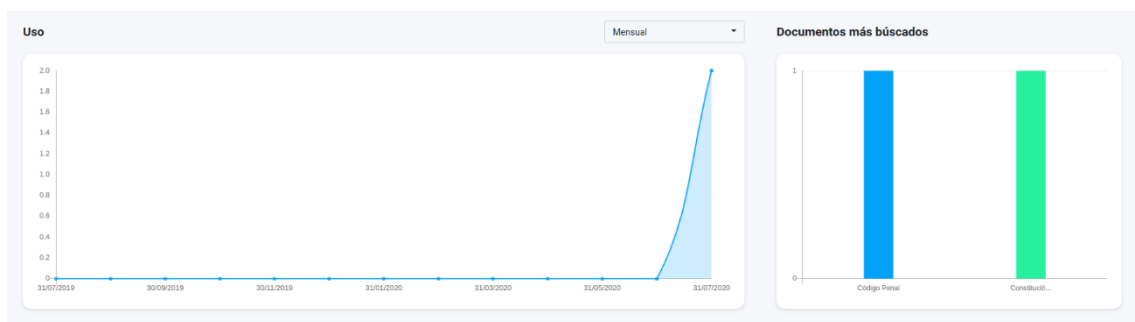
En este apartado se verá información básica como la cantidad de visitas que se han hecho y cuantos documentos que se encuentran ingresados.

- **Vistas Totales** Cuantas veces los usuarios han visualizado cualquier normativo, estas búsquedas solo incrementan al abrir uno desde la

sección de búsqueda o la de documentos públicos y no lo harán al hacerlo desde el dashboard.

- **Documentos** Cantidad de documentos ingresados.

2. Estadísticas

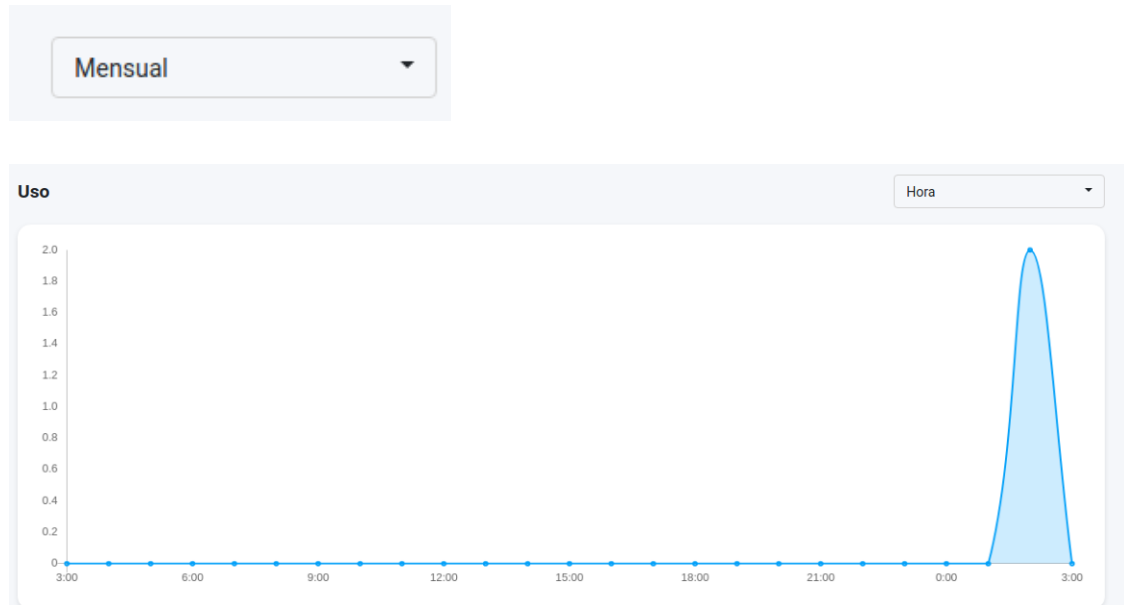


En esta sección se encuentran estadísticas de uso del programa. Mostrará dos gráficas: documentos más buscados y el uso que se le está dando al programa.

a. Uso

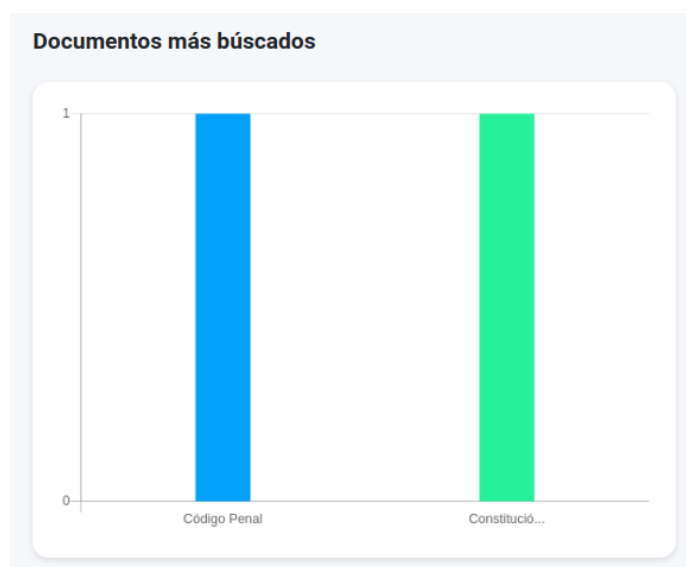
Esta gráfica indica cuántas búsquedas se han realizado agrupándolas por fecha.

Se puede cambiar el rango de tiempo en la esquina superior derecha:



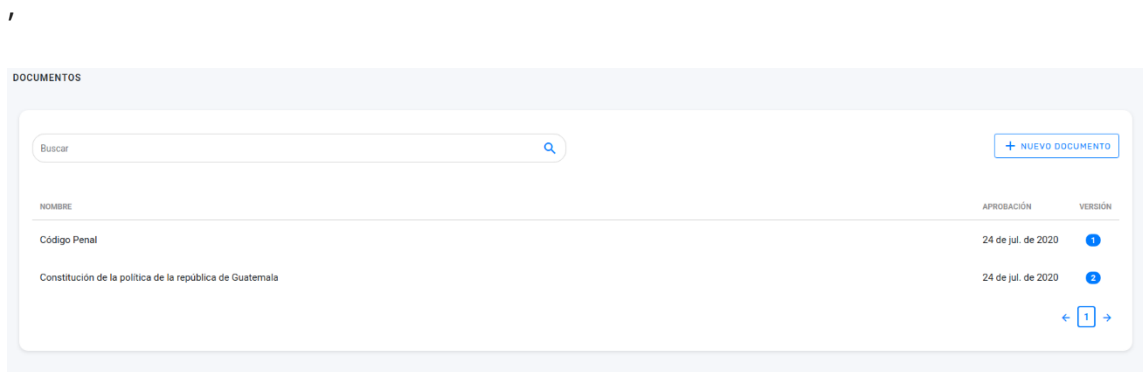
b. Documentos más buscados

En esta sección se encontrará el top de los documentos que más búsquedas han recibido. Al darle clic al gráfico en este caso la barra, se abrirá el menú del documento, con la versión que es la que ha sido más buscada, este se verá a fondo en la sección respectiva.



3. Tabla de Documentos

En esta sección se realizará el gestionamiento de los documentos, es decir agregar documentos, agregar versiones, abrir documentos, eliminar documentos y ver las estadísticas específicas de cada documento.



La vista principal de la tabla cuenta con las siguientes partes:

El buscador: Servirá para encontrar fácilmente cualquier documento.

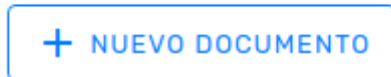
Agregar Nuevo Documento: Con este botón se podrá agregar documentos nuevos (la primera versión de estos).

Tabla: Mostrará todos los documentos, si no hay búsqueda los ordenará por tiempo de modificación, es decir que el documento que se modificó más recientemente se mostrará primero. Se puede dar clic a las filas de la tabla para abrir el menú del documento que se encuentra en esta.

Paginación: En caso de existir muchos documentos se cuenta con paginación en la tabla.

a. Agregar Nuevo Documento

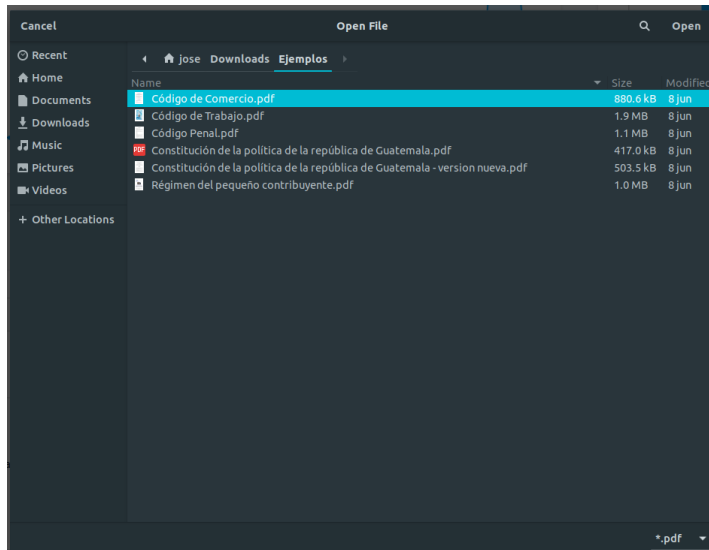
Para agregar nuevos documentos se debe dar clic al botón que se encuentra en la esquina superior derecha de la tabla de documentos.



Al dar clic en Nuevo Documento, se abrirá el siguiente menú con un formulario que se debe llenar con la información del documento a ingresar:

Una ventana modal titulada 'Nuevo Documento' con un botón de cerrar (X) en la esquina superior derecha. El formulario contiene los siguientes campos: 'ESCOGE UN ARCHIVO' con un campo de texto que dice 'Buscar' y un botón 'Browse'; 'NOMBRE DEL NORMATIVO' con un campo de texto vacío; 'TIPO' con un menú desplegable que muestra 'Interno'; 'FECHA DE APROBACIÓN' con un campo de texto que muestra '14/08/2020'. En la parte inferior derecha hay dos botones: 'CERRAR' (gris) y 'GUARDAR' (azul).

Lo primero que se debe hacer es dar clic en la opción de escoger archivo, esto abrirá una ventana, allí se tiene que escoger el archivo que se ha de agregar.



Al haber seleccionado el archivo se autocompleta el nombre del documento, aunque puede cambiarse si se desea.

Nuevo Documento

ESCOGE UN ARCHIVO

Código de Comercio.pdf

Browse

NOMBRE DEL NORMATIVO

Código de Comercio

TIPO

Interno

FECHA DE APROBACIÓN

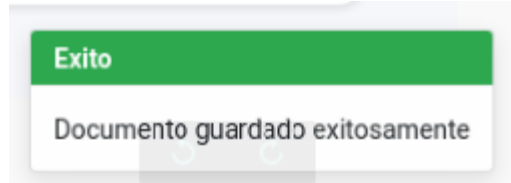
14/08/2020

CERRAR

GUARDAR

Luego de eso se debe indicar si es un documento interno o externo, y la fecha en la que se aprobó el documento. Con los datos ya ingresados se debe dar clic en guardar.

En caso de éxito se mostrará la siguiente notificación



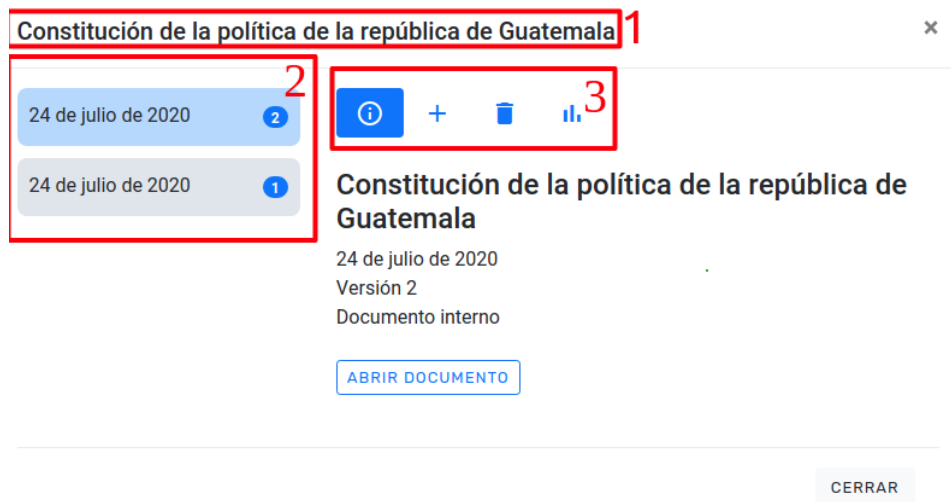
b. Menú de Documento

Para abrir las opciones de cada documento, se debe dar clic a la fila de la tabla con el documento que se quiere ingresar:

NOMBRE	APROBACIÓN	VERSIÓN
Código de Comercio	14 de ago. de 20...	1
Código Penal	24 de jul. de 2020	1
Constitución de la política de la república de Guatemala	24 de jul. de 2020	2

[←](#) 1 [→](#)

Esto abrirá el siguiente menú



Donde

1. **Nombre del Documento**
2. **Versiones del Documento** Acá es posible cambiar entre las diferentes versiones del documento, ya sea para ver su información o estadísticas.
3. **Opciones del Documento** son pestañas que mostrarán diferentes opciones o menús que se pueden realizar sobre cada documento. Estos menús son:
 - a. Información
 - b. Agregar Versión
 - c. Borrar Versión
 - d. Estadísticas
 - i. **Información del Documento**

En este menú se muestra información detallada acerca del documento, por ejemplo, la fecha de aprobación, versión y tipo de documento, además de eso, es posible abrir el archivo PDF del documento.

Al intercambiar entre versiones del documento, se mostrará la información de la versión que se encuentre seleccionada.



Constitución de la política de la república de Guatemala

24 de julio de 2020

Versión 2

Documento interno

ABRIR DOCUMENTO

Para ver el documento se debe dar clic en el botón abrir documento

ABRIR DOCUMENTO

Constitución Política de la República de Guatemala.

(Reformada por Acuerdo legislativo No. 18-93 del 17 de Noviembre de 1993)

TÍTULO I

LA PERSONA HUMANA, FINES Y DEBERES DEL ESTADO

CAPÍTULO ÚNICO

Artículo 1.- Protección a la Persona. El Estado de Guatemala se organiza para proteger a la persona y a la familia; su fin supremo es la realización del bien común.

Artículo 2.- Deberes del Estado. Es deber del Estado garantizarle a los habitantes de la República la vida, la libertad, la justicia, la seguridad, la paz y el desarrollo integral de la persona.

TÍTULO II

DERECHOS HUMANOS

CAPÍTULO I

DERECHOS INDIVIDUALES

ii. Agregar Versión



ESCOGE UN ARCHIVO

Buscar

Browse

FECHA DE APROBACIÓN

14/08/2020

CERRAR

GUARDAR

Al momento que se quiera agregar una nueva versión a un documento, se presentará un formulario similar al de “agregar nuevo documento”, pero con menos opciones ya que lo único que pedirá ingresar es el archivo y la fecha de aprobación.

iii. Eliminar Documento

INGRESA LA PALABRA ELIMINAR

eliminar

El documento ya tiene vistas de los usuarios finales

CERRAR

ELIMINAR

Antes de eliminar un documento se debe tener en cuenta lo siguiente: Solo se puede realizar esta acción en la última versión de este y no debe tener ninguna visualización por los usuarios que no sean administradores, es decir vistas que son desde el buscador o documentos públicos. En la imagen anterior no se cumplen esos requisitos ya que ese documento ya tiene una visualización. En caso de cumplir los requisitos se mostrará lo siguiente:



INGRESA LA PALABRA ELIMINAR

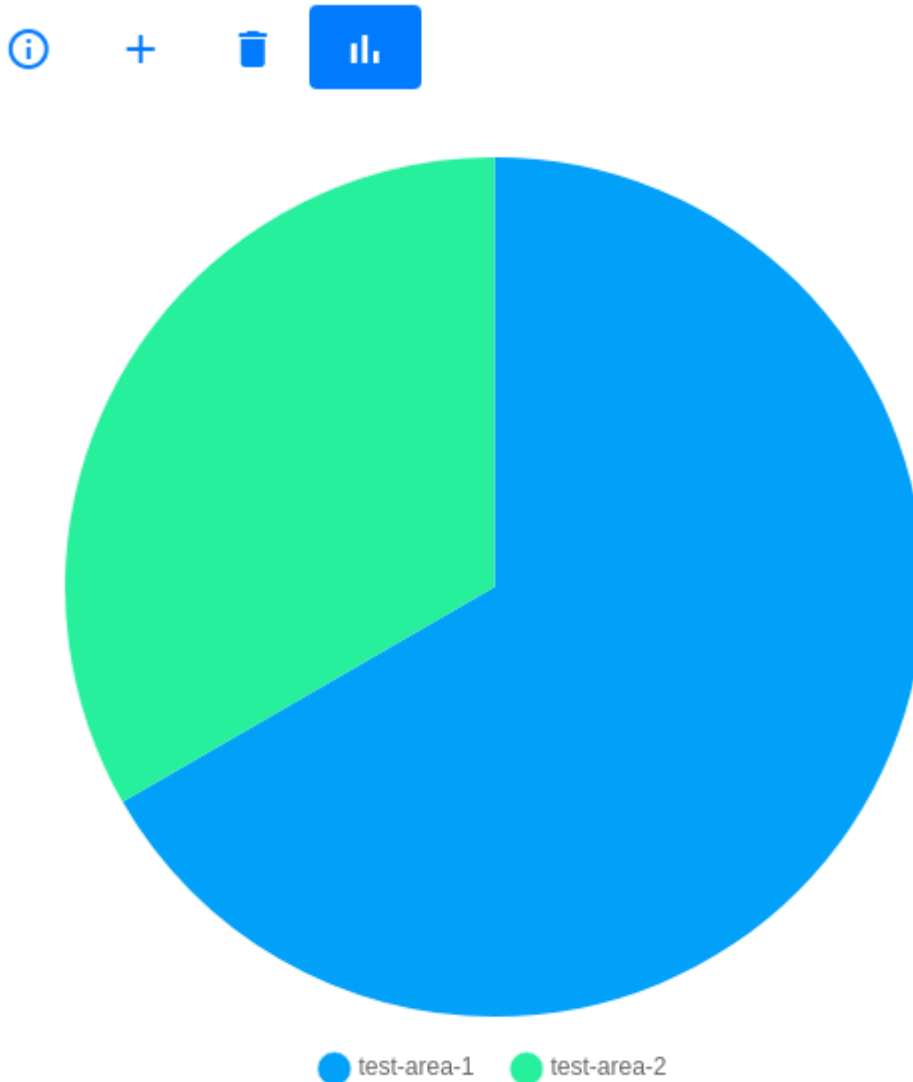
eliminar

Estas apunto de eliminar un documento esta accion no es reversible

Para borrar el documento se debe escribir la palabra eliminar en el campo de texto, lo cual habilitará el botón, al darle clic a este el documento será eliminado permanentemente (la última versión).

iv. Estadísticas del Documento

En la última sección se mostrarán las estadísticas de búsqueda del documento, estas se encuentran agrupadas por departamento.



Al pasar el cursor sobre cada área del gráfico mostrará cuántas búsquedas totales existen en ese departamento. (En el caso de la imagen son 2 búsquedas)

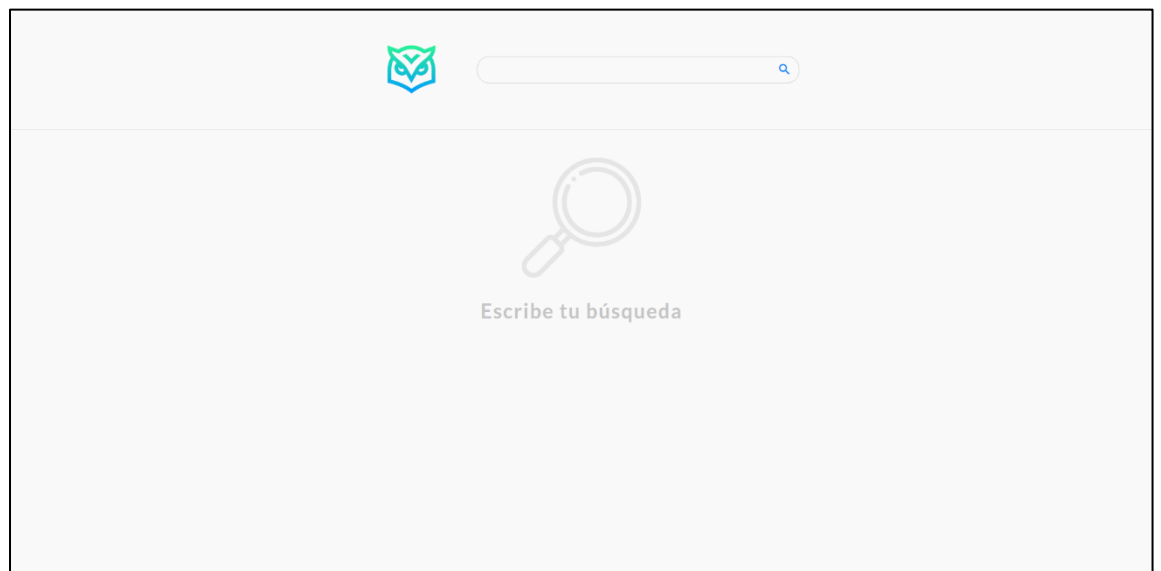


Buscador

Para ingresar a esta sección ir a la siguiente URL en el navegador

<http://localhost/search.php>

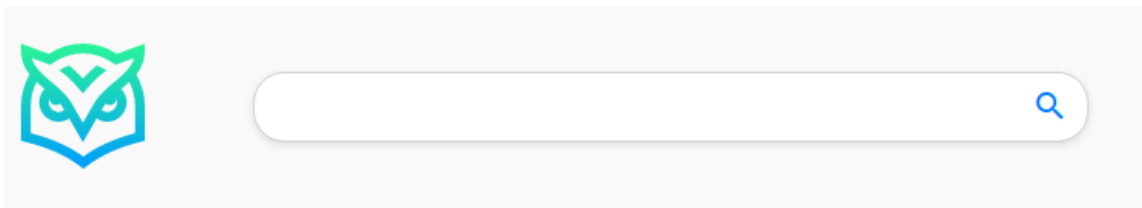
Al ingresar al buscador se presentará la siguiente interfaz:



Esta cuenta con dos secciones:

1. Búsqueda
2. Resultados

1. Sección de búsqueda

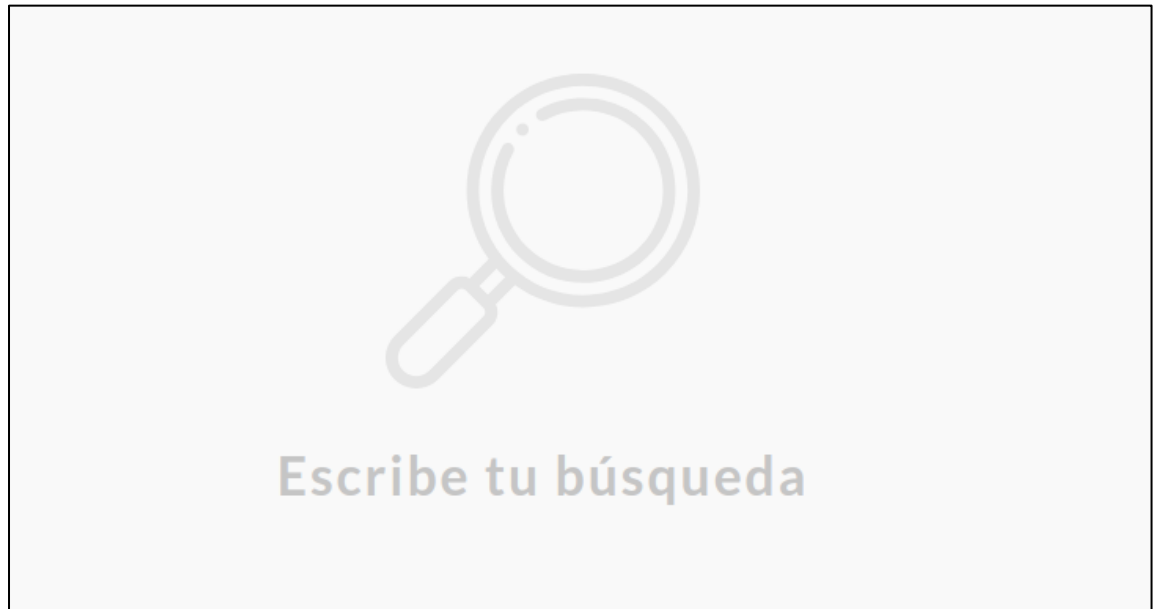


Acá se ingresa la búsqueda que se quiera realizar sobre el documento, por ejemplo, se puede buscar la frase “Como ser magistrado en la corte suprema de justicia”, esto se encuentra en la Constitución Política de la República de Guatemala, que es uno de los documentos ingresados (para el ejemplo).

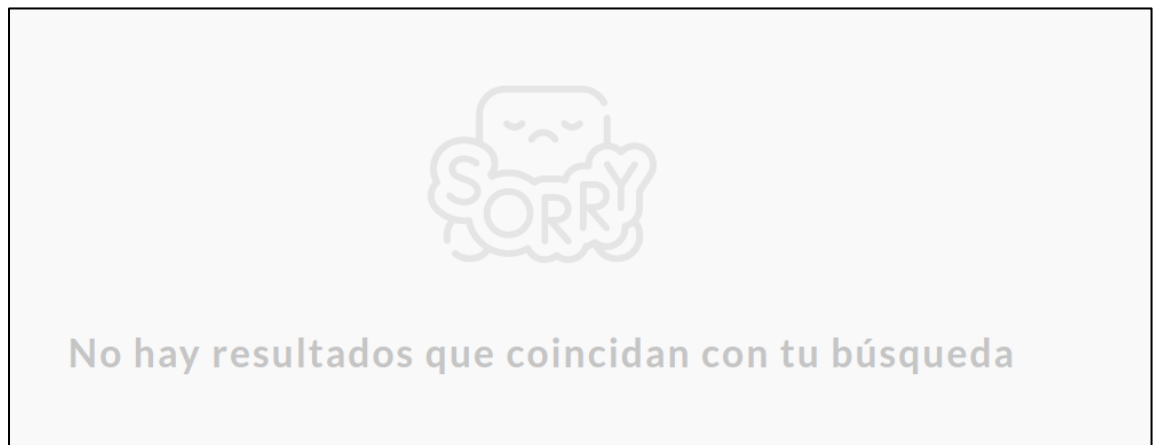
Para realizar esta búsqueda solo se ingresa en el campo de texto y se da clic en el botón con forma de lupa, o se presiona la tecla Entrar estando en el campo de texto.

2. Sección de resultados

Esta sección puede tener varios estados como, por ejemplo: solicitar que se ingrese la búsqueda



O que la búsqueda no se encontró



Además de ser el encargado de mostrar los resultados, probando con la búsqueda mencionada anteriormente:



Como ser magistrado en la corte suprema de justicia

Constitución de la política de la república de Guatemala

.- Requisitos para **ser magistrado** de la **Corte Suprema de Justicia**....

Constitución de la política de la república de Guatemala

.- Requisitos de los **magistrados** de la **Corte** de Constitucionalidad. Para **ser magistrado** de la **Corte** de Constitucionalidad, se requiere llenar los siguientes requisitos: a. **Ser** guatemalteco de origen; b. **Ser** abogado colegiado; c. **Ser** de reconocida honorabilidad; y d....

Constitución de la política de la república de Guatemala

.- Requisitos para **ser magistrado** o juez....

Como se puede ver muestra los documentos en los que encontré la búsqueda. Si se da clic al nombre del normativo, es decir al título en color azul llevará a la página del PDF en la que encontré ese resultado.

Constitución de la política de la república de Guatemala

.- Requisitos para *ser magistrado* de la *Corte Suprema de Justicia*....

En caso de falta temporal del Presidente del Organismo Judicial o cuando conforme a la ley no pueda actuar o conocer, en determinados casos, lo sustituirán los demás magistrados de la Corte Suprema de Justicia en el orden de su designación.

Artículo 215.- (Reformado) Elección de la Corte Suprema de Justicia. Los magistrados de la Corte Suprema de Justicia serán electos por el congreso de la República para un período de cinco años, de una nómina de veintiseis candidatos propuestos por una comisión de postulación integrada por un representante de los rectores de las universidades del país, que la preside, los decanos de las facultades de Derecho o Ciencias Jurídicas y Sociales de cada universidad del país, un número equivalente de representantes electos por la Asamblea General del Colegio de Abogados y Notarios de Guatemala y por igual número de representantes electos por los magistrados titulares de la Corte de Apelaciones y demás tribunales a que se refiere el artículo 217 de esta Constitución.

La elección de candidatos requiere del voto de por lo menos las dos terceras partes de los miembros de la comisión.

En las votaciones tanto para integrar la Comisión de Postulación para la integración de la nómina de candidatos, no se aceptará ninguna representación.

Los magistrados de la Corte Suprema de Justicia elegirán, entre sus miembros, con el voto favorable de las dos terceras partes, al presidente de la misma, el que durará en sus funciones un año y no podrá ser reelecto durante ese período de la Corte.

Artículo 216.- Requisitos para ser magistrado de la Corte Suprema de Justicia. Para ser electo magistrado de la Corte Suprema de Justicia, se requiere, además de los requisitos previstos en el artículo 207 de esta Constitución, ser mayor de cuarenta años, y haber desempeñado un período completo como magistrado de la Corte de apelaciones o de los tribunales colegiados que tengan la misma calidad, o haber ejercido la profesión de abogado por más de diez años.

SECCIÓN TERCERA

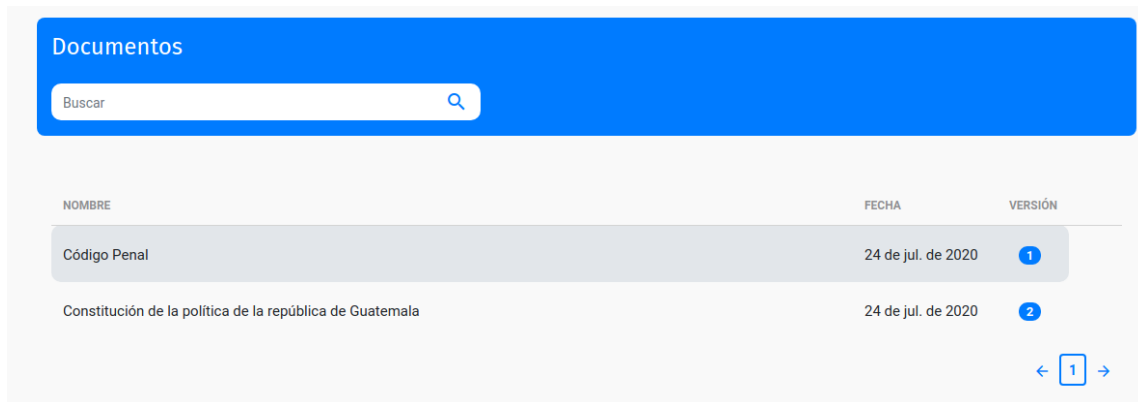
CORTE DE APELACIONES Y OTROS TRIBUNALES

Documentos Públicos

Para ingresar a esta sección se debe ir a la siguiente URL en el navegador

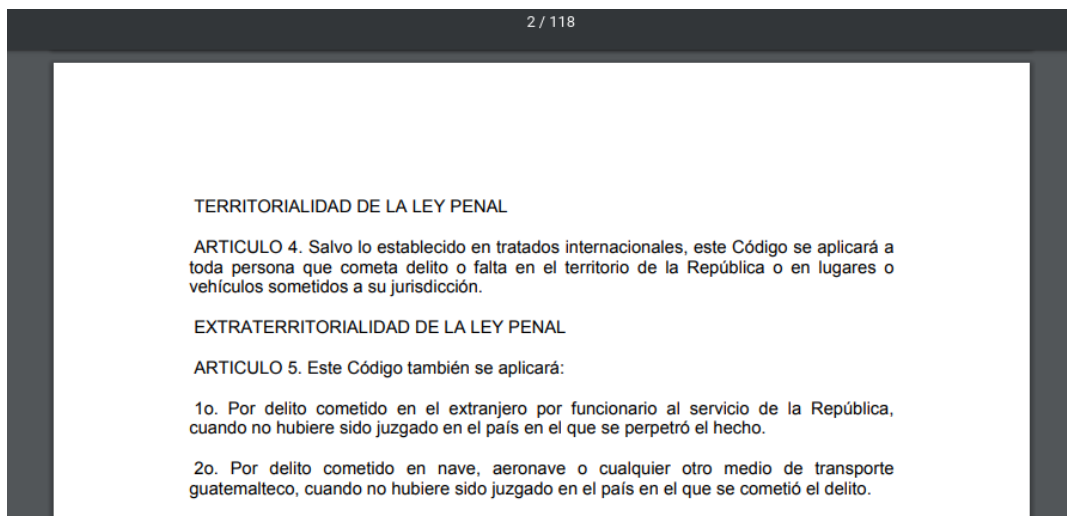
<http://localhost/documents-public.php>

Al ingresar al buscador se presentará la siguiente interfaz:



NOMBRE	FECHA	VERSIÓN
Código Penal	24 de jul. de 2020	1
Constitución de la política de la república de Guatemala	24 de jul. de 2020	2

Esta parte tiene una función similar a la tabla de documentos, pero únicamente se pueden visualizar los normativos.



2 / 118
TERRITORIALIDAD DE LA LEY PENAL ARTICULO 4. Salvo lo establecido en tratados internacionales, este Código se aplicará a toda persona que cometa delito o falta en el territorio de la República o en lugares o vehículos sometidos a su jurisdicción. EXTRATERRITORIALIDAD DE LA LEY PENAL ARTICULO 5. Este Código también se aplicará: 1o. Por delito cometido en el extranjero por funcionario al servicio de la República, cuando no hubiere sido juzgado en el país en el que se perpetró el hecho. 2o. Por delito cometido en nave, aeronave o cualquier otro medio de transporte guatemalteco, cuando no hubiere sido juzgado en el país en el que se cometió el delito.

Apéndice 3. Manual de instalación

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE ORIENTE
CARRERAS DE INGENIERÍA

“MANUAL DE INSTALACIÓN MOTOR DE BÚSQUEDA DE LAS
NORMATIVAS EN LA COOPERATIVA DE AHORRO Y CRÉDITO
INTEGRAL SAN JOSÉ OBRERO (COOSAJO, R.L.) EN EL
MUNICIPIO DE ESQUIPULAS, CHIQUIMULA”

JOSÉ EDUARDO GUERRA SOSA

OCTUBRE, 2020

Glosario

Backend es la parte del desarrollo web que se encarga que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no se ven como, por ejemplo, la comunicación con el servidor.

Documentos Archivos pdfs que se guardarán

ElasticSearch es un servidor de búsqueda basado en Lucene.

Índices es una colección de documentos relacionados entre sí. Elasticsearch almacena datos como documentos JSON.

Owl software que guarda documentos, realiza búsquedas sobre estos y obtiene estadísticas de estas búsquedas.

Servicio Programa (proceso) que correrá en la computadora, incluso luego de reiniciarla.

Servidor Un servidor es una aplicación en ejecución capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia.

Empezando

Este manual fue creado para la instalación de owl en un ambiente Linux, la distribución que se utilizó fue Ubuntu 18.04.

Owl necesita de tres componentes para su funcionamiento:

- **ElasticSearch**, que será el servidor de búsqueda, este se encarga de guardar los datos de los documentos y realizar búsquedas sobre estos datos.
- **El backend**, que está creado en *NodeJS*, este contiene la lógica de la aplicación, por ejemplo, el ingreso de documentos, o la búsqueda de texto en los documentos ingresados.
- **El frontend**, que es la parte visual con la que el usuario final interactuará, está creada en HTML y JS para ser implementada fácilmente junto a PHP.

Además de esos componentes se creó un instalador, que se encargará de crear las carpetas necesarias para el funcionamiento de owl; también es el encargado de crear la estructura de los índices en ElasticSearch.

Nota: El manual asume que la instalación se está realizando en un ambiente local de ser necesario se debe utilizar la IP del servidor en lugar de localhost.

Instalación de prerequisites

1. NodeJS

Es necesario tener instalado NodeJS y npm para hacer esto en Ubuntu 18.04 se tienen que ejecutar los siguientes comandos:

```
sudo apt install nodejs
```

```
sudo apt install npm
```

2. Elastic Search

También se debe instalar ElasticSearch, la versión que utiliza el proyecto es la **7.6.2** para descargarla hay que correr los siguientes comandos:

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.2-amd64.deb
```

Antes de instalar ElasticSearch se puede comprobar el checksum:

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.2-amd64.deb.sha512  
shasum -a 512 -c elasticsearch-7.6.2-amd64.deb.sha512
```

Luego se extrae el archivo:

```
sudo dpkg -i elasticsearch-7.6.2-amd64.deb
```

Con estos pasos ya se ha instalado ElasticSearch, lo siguiente que hay que hacer es habilitar ElasticSearch como un servicio:

```
sudo /bin/systemctl daemon-reload  
sudo /bin/systemctl enable elasticsearch.service
```

Para iniciar y parar los servicios se ejecuta lo siguiente:

```
sudo systemctl start elasticsearch.service  
sudo systemctl stop elasticsearch.service
```

Más información acerca de la instalación y otras formas de instalación:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/install-elasticsearch.html>

3. NGINX

NGINX servirá como servidor para el *frontend* para despachar páginas php. Para instalar NGINX ingresar en la terminal:

```
sudo apt update  
sudo apt install nginx
```

NGINX necesita un servicio para procesar php para eso se tiene que tener instalado php y php-fpm con el siguiente comando:

```
sudo apt-get install php libapache2-mod-php  
sudo apt-get install php-fpm
```

También hay que instalar php-curl ya que es un plugin necesario para la aplicación

```
sudo apt install php-curl
```

Ahora hay que configurar el archivo de ajustes de php, para esto se debe modificar el archivo de php.ini con cualquier editor de texto. Para este ejemplo se utilizó nano:

```
sudo nano /etc/php/7.4/fpm/php.ini
```

Y en el archivo realizar los siguientes cambios:

Agregar o descomentar la siguiente línea

```
extension=curl
```

Y luego descomentar y modificar estos valores

```
upload_max_filesize          =          100M
post_max_size                 =          100M
cgi.fix_pathinfo=0
```

Ya configurado esto, para reiniciar el servicio de php-fpm se ejecuta lo siguiente:

```
sudo systemctl restart php7.4-fpm
```

Ahora hay que configurar nginx, para eso se edita el siguiente archivo

```
sudo nano /etc/nginx/nginx.conf
```

En el archivo, ir a la sección de http y dentro de ella se agrega o modifica la siguiente línea

```
client_max_body_size 100M;
```

Y se reiniciar el servicio de NGINX

```
sudo service nginx restart
```

Instalación de Owl - Backend

1. Configuración del instalador

Primero descargar el archivo owl-installer y descomprimirlo.

Antes de instalar owl, modificar el archivo *settings.json* (que se encuentra en la carpeta raíz de owl-installer), este contendrá información necesaria al momento de la instalación y luego ejecución de owl.

El archivo contiene lo siguiente:

```

{
  "installationPath": "/opt/cunori/owl/",
    "fileServerURL": "http://localhost/documents",
  "serverPort": 3000,
  "metadataIndexName": "normativas-metadata",
  "filePath": "/home/user/Documents/Normativas/",
  "mainIndexName": "normativas",
  "statsIndexName": "normativas-stats",
  "backendRepository":
    "https://gitlab.com/josguerra99/buscador.git",
  "elasticHost": "http://localhost:9200"
}

```

Donde:

- installationPath es la ubicación en la que se descargará el programa.
- filePath es la ubicación en la que se guardarán los documentos PDFs en el sistema de archivos.
- fileServerURL dirección hacia el servidor que servirá los archivos (siempre debe terminar en /documents ya que el instalador creará esa carpeta automáticamente).
- serverPort es el puerto en el que funcionará el backend.
- metadataIndexName, mainIndexName y statsIndexName son los nombres de los índices que se crearán, no es necesario cambiar estos nombres.
- backendRepository el repositorio git del cual descargará los archivos del backend.
- elasticHost Dirección url al servidor de elasticsearch.

2. Instalación

Antes de empezar la instalación tiene que estar ejecutándose elasticsearch.

Además, es importante que el usuario tenga permisos para poder leer y escribir en las carpetas de instalación y en la que se guardaran los archivos, sin recurrir a sudo, para darle permisos a esas carpetas:

```
chmod 777 installationPath  
chmod 777 filePath
```

Donde installationPath y filePath son las ubicaciones que se asignaron en el archivo de settings.json.

Para iniciar la instalación ir a la carpeta raíz de owl-installer y ejecutar el siguiente comando:

```
./install.sh
```

Si es la primera vez que se ejecuta el instalador, este descargará unas dependencias para su funcionamiento. Luego de esto, el instalador presentará tres opciones:

```
> Instalar programa  
> Crear índices  
> Salir
```

Al ser la primera vez en el instalador, ir a la opción de Instalar programa y confirmar que es lo se quiere instalar.

Tener cuenta que la primera vez será una instalación en limpio, si se vuelve a instalar el programa este *eliminará todo* lo anterior.

Luego de esto, hay que crear los índices así que se elige la segunda opción, si es la primera vez que se está instalando, estos se crearán sin ningún problema pero si ya existieran se presentará la opción de volverlos a crear o no hacer nada, en esta parte se tiene que tener en cuenta que si se elige volverlo a crear se *eliminarán todos los datos anteriores*.

Iniciar Owl como servicio

Primero descargar PM2, que es una herramienta que maneja procesos de NodeJS y permite seguir corriendo la aplicación incluso luego de un reinicio o fallo.

Para instalar pm2 ejecutar el siguiente comando

```
npm install pm2 -g
```

Luego se debe ejecutar el siguiente comando para configurar pm2

```
pm2 startup
```

Esto devolverá un comando que se debe ejecutar, imprimirá algo similar a esto (este resultado puede variar).

```
[PM2] You have to run this command as root. Execute the following
command:
  sudo su -c "env
PATH=$PATH:/home/unitech/.nvm/versions/node/v4.3/bin pm2 startup
<distribution> -u <user> --hp <home-path>
```

Se ejecuta el comando que haya devuelto (como ejemplo se usará el comando que se encuentra en la documentación PM2)

```
sudo su -c "env
PATH=$PATH:/home/unitech/.nvm/versions/node/v4.3/bin pm2 startup
<distribution> -u <user> --hp <home-path>
```

Ahora se debe ir al directorio en que se instaló owl

```
cd /ubicación que hayas instalado/buscador
```

Para utilizar pm2 se realiza lo siguiente:

```
pm2 ecosystem
```

```
pm2 start server.js
```

Con esto terminado ya se instaló el backend de Owl en el servidor.

Instalación de Owl - FrontEnd

Primero descargar el archivo frontend y descomprimirlo. Luego se debe mover los archivos a la carpeta en la que se quiera guardar el servidor típicamente es /var/www/html. Estando en la carpeta contenedora del frontend ejecutar:

```
sudo mv frontend /var/www/html
```

Luego entrar a la carpeta frontend/js

```
sudo cd /var/www/html/frontend/js
```

Y editar el archivo helper.js

```
sudo nano helper.js
```

Y editar la variable apihost para que tenga la ip del servidor en este caso localhost

```
let apihost= "http://localhost/";
```

Ahora hay que configurar NGINX para que responda a estas peticiones, para

esto, se edita el siguiente archivo:

```
sudo nano /etc/nginx/sites-available/default
```

Y dejar el archivo de configuración de la siguiente forma

```
server {
listen 80 default_server;
listen [::]:80 default_server;
server_name localhost;

root /var/www/html/frontend;
index index.php index.html index.htm index.nginx-debian.html;

location ~ \.php {
    add_header "Access-Control-Allow-Origin" *;
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
}

location ~ \.pdf {
    root /home/jose/Documents/Normativas;
}

}
```

Donde root /var/www/html/frontend es la ubicación en la que se encuentra ubicada el frontend y root /home/jose/Documents/Normativas es filePath que se colocó en el archivo settings.json.

Ahora solo hay que reiniciar el servicio de NGINX

```
sudo service nginx restart
```

Y con esto ya se tiene instalado y funcionando Owl. Para probarlo se debe entrar a los siguientes links:

<http://localhost/dashboard.php>

<http://localhost/search.php>

<http://localhost/documents-public.php>

En el manual de usuario se encontrará más información acerca del uso de la aplicación.

Apéndice 4. **Manual técnico API**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE ORIENTE
CARRERAS DE INGENIERÍA

“MANUAL TÉCNICO API MOTOR DE BÚSQUEDA PARA LAS
NORMATIVAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO INTEGRAL
SAN JOSÉ OBRERO (COOSAJO, R.L.) EN EL MUNICIPIO DE ESQUIPULAS,
CHIMULULA”

JOSÉ EDUARDO GUERRA SOSA

OCTUBRE, 2020

Empezando

Para extender la aplicación y crear más índices de Elasticsearch se utilizará la aplicación de owl-installer.

Nota: *El manual asume que se está trabajando en un ambiente local. De ser necesario se debe de utilizar la ip del servidor en lugar de localhost.*

Instalación de prerequisites

1. NodeJS

Es necesario tener instalado NodeJS y npm, para hacer esto en Ubuntu 18.04 ejecutar los siguientes comandos:

```
sudo apt install nodejs
```

```
sudo apt install npm
```

2. Elastic Search

También instalar ElasticSearch, la versión que utiliza el proyecto es la **7.6.2** y para descargarla ejecutar lo siguiente:

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.2-amd64.deb
```

Antes de instalar elasticsearch se puede comprobar el checksum y para hacer esto ejecutar los siguientes comandos:

```
wget  
https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.2-amd64.deb.sha512  
shasum -a 512 -c elasticsearch-7.6.2-amd64.deb.sha512
```

Luego se debe extraer el archivo:

```
sudo dpkg -i elasticsearch-7.6.2-amd64.deb
```

Si ya se ha instalado ElasticSearch, lo siguiente es habilitar ElasticSearch como un servicio:

```
sudo /bin/systemctl daemon-reload  
sudo /bin/systemctl enable elasticsearch.service
```

Para iniciar y parar los servicios ejecutar los siguientes comandos:

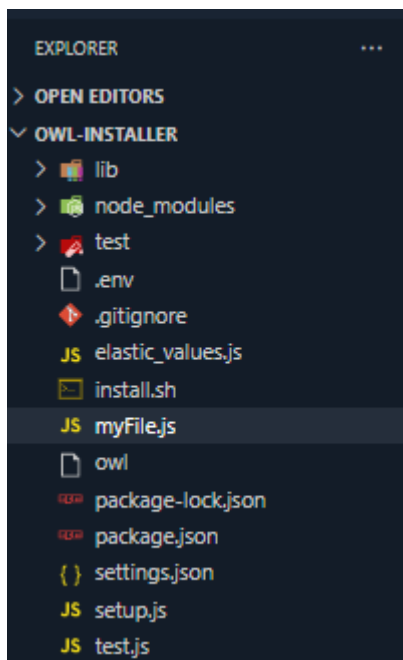
```
sudo systemctl start elasticsearch.service  
sudo systemctl stop elasticsearch.service
```

Para más información acerca de la instalación y otras formas de instalación dirigirse a: <https://www.elastic.co/guide/en/elasticsearch/reference/current/install-elasticsearch.html>

Extendiendo el instalador

Primero se debe descargar el archivo owl-installer y descomprimirlo. Y abrir la carpeta con cualquier editor de código, en este caso se utilizó VSCode.

Luego se debe crear un archivo de JavaScript nuevo dentro de la carpeta principal, en este caso se llamó myFile.js pero puede tener cualquier nombre.



Ahora abrir el archivo e importar unos módulos necesarios, estos son `elastic_values` e `indices_setup`.

`elastic_values` contiene plantillas que servirán para la creación de los índices de ElasticSearch.

`indices_setup` se comunica con ElasticSearch para crear el índice.

Para importar estos módulos se tiene que agregar lo siguiente:

```
const indices = require("../lib/indices_setup");
const values = require("../elastic_values");
```

Luego de eso hay que crear las propiedades, es decir los campos que tendrá el índice, para hacer eso solo se crea un objeto con el siguiente formato:

```
const properties = {
  name: { type: "text" },
  address: { type: "text" },
  date: { type: "date" },
};
```

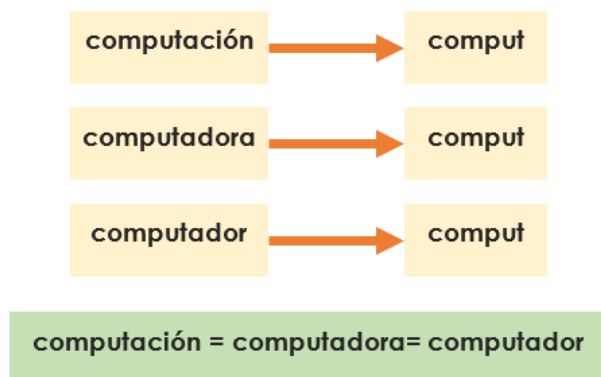
En este caso se agregaron 3 campos: nombre, dirección y fecha.

Además de eso, a los campos de tipo texto se les puede agregar un analizador, este es el que buscará el texto dentro del índice, si se utilizarán los que se encuentran dentro del módulo de elastic_values se pueden agregar 3 analizadores:

spanish_default que es el analizador de español por defecto.

spanish_no_stemmer el cual no pasará las palabras por un proceso de stemming, es decir no llevará las palabras a su raíz.

Ejemplo de stemming (este proceso no se realiza con este analizador)



spanish_autocomplete este analizador autocompleta las frases, es decir que encontrará coincidencias incluso cuando la palabra o frase que se está buscando no está completa.

Para agregar estos analizadores solo se debe modificar el objeto de propiedades que se acaba de crear de la siguiente manera:

```
const properties = {  
  name: {  
    type: "text",  
    analyzer: "spanish_autocomplete",  
  },  
  address: {type: "text",  
    analyzer: "spanish_default",  
  },  
  date: { type: "date" },  
};
```

Ahora solo hay que agregar una función la cual creará el índice, esta debe ser asíncrona o debe utilizar callbacks, en este caso se utilizará una asíncrona.

```
async function createIndex() {  
  try {  
  } catch (err) {  
    console.log(`Error en el proceso de creación ${err}`);  
  }  
}
```

Actualmente esta función no realiza ninguna acción (no existe nada dentro del try) así que allí es donde se debe de agregar la lógica para crear el índice en Elasticsearch y acá es donde se hace uso del otro módulo de indices_setup, así que dentro del try solo hay que hacer uso de la función createIndex que se encuentra dentro de este módulo. Esta función pide el nombre del índice, las propiedades o campos (que es lo que se creó anteriormente) y settings que en este caso se usarán los que se encuentran dentro del módulo de elastic_values.

```
async function createIndex() {  
  try {  
    indices.createIndex("prueba", properties, values.settings);  
  } catch (err) {  
    console.log(`Error en el proceso de creación ${err}`);  
  }  
}
```

Por último, solo es necesario llamar a esa función para iniciar el proceso de creación. En este caso se llamará directamente.

```
createIndex();
```

Para iniciar el proceso de creación solo se debe ejecutar el programa desde una terminal, con el comando `node myFile.js` desde la carpeta raíz. (De ser necesario antes de esto ejecutar el comando `npm install` para las dependencias).

```
PS D:\Documentos\Projects\Owl-20201113T110300Z-001\Owl\owl-installer\owl-installer> node .\myFile.js
El índice prueba ya existe, si reemplazas el índice este eliminará los datos que contenga
¿ Deseas reemplazarlo? ... no / si

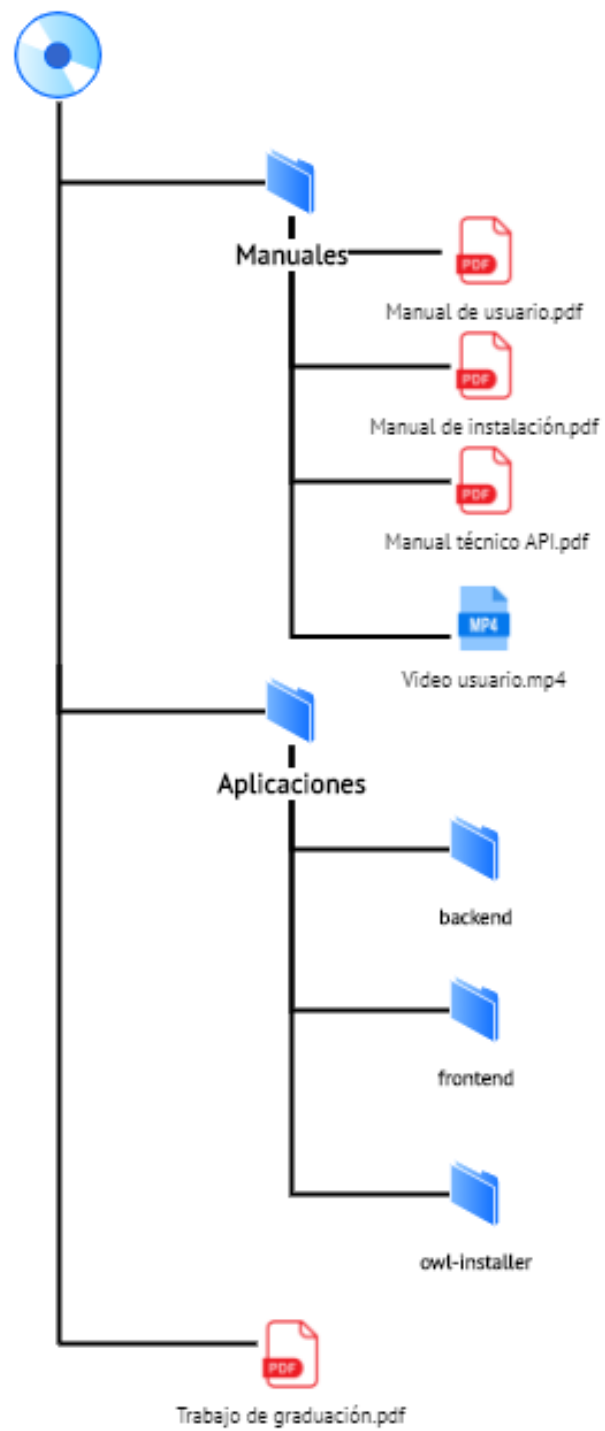
Índice [prueba] eliminado exitosamente:

Índice creado exitosamente [prueba]

Analizador agregado exitosamente a [prueba]:
{"analysis":{"filter":{"spanish_stop":{"type":"stop","stopwords":"_spanish_"},"spanish_keywords":{"type":"keyword_marker","keywords":[]},"spanish_keywords_filter":{"type":"keyword_filter","filters":{"spanish_default":{"tokenizer":"standard","filter":["lowercase","spanish_stop","spanish_keywords","asciifolding","spanish_stemmer"]},"spanish_keywords_filter":{"type":"keyword_filter","filters":{"spanish_default":{"tokenizer":"standard","filter":["lowercase","spanish_keywords","asciifolding","autocomplete_filter"]}}}}}}}}

Mapeado creado exitosamente para [prueba]:
{"name":{"type":"text","analyzer":"spanish_autocomplete"},"address":{"type":"text","analyzer":"spanish_default"},"date":{"type":"date"}}
```

Apéndice 5. Contenido del CD



ANEXOS

1. Certificación de entrega y aceptación del proyecto

Anexo 1. Certificación de entrega y aceptación del proyecto

COOPERATIVA DE AHORRO Y CRÉDITO INTEGRAL
"SAN JOSÉ OBRERO" R.L.

4ta. Ave. 9-01, Zona 1. Esquipulas, Chiquimula.
Guatemala, C.A. 20007 - PBX.: (502) 7873-0808
www.coosajo.com

Esquipulas, 9 de octubre de 2020.

Ing. René Estuardo Alvarado González
Coordinador de EPS y Trabajos de Graduación
Carrera de Ingeniería y Ciencias y Sistemas
Centro Universitario de Oriente
Universidad San Carlos de Guatemala

Respetable Ingeniero:

Por medio de la presente se hace constar la entrega-recepción del software desarrollado derivado de Ejercicio Profesional Supervisado del estudiante José Eduardo Guerra Sosa, que se identifica con el registro académico No. 201544097 y código único de identificación 3332 98284 1804, de la carrera de Ingeniería en Ciencias y Sistemas del Centro Universitario de Oriente. Se recibió a entera satisfacción por la Coordinación del área de Educación Cooperativa de COOSAJO R.L. la entrega de lo siguiente:

- Sistema de Información para el acceso a la documentación de normativas que rigen a la Cooperativa San José Obrero (COOSAJO R.L.), en el municipio de Esquipulas, Chiquimula.
- Manuales de usuario y técnico para el uso de la aplicación.

No habiendo otro asunto que tratar, se concluye la presente. Firmo a continuación como responsable de la verificación de la solución por parte del área de Educación Cooperativa de COOSAJO R.L.

Jorge Alberto Flores Sagastume
Gerente de Ingeniería de Sistemas de Información

COOSAJO R.L.

MICOPE

Las Minas Esquipulas
Km. 222.5 Carretera a Honduras
Tels: 7873-0944 / 7873-0945 / 7873-0946

Chanmagua
Aldea Chanmagua, Esquipulas
Tel: 7873-0948

Quezaltepeque
3ª. Ave. 1-26, Zona 1.
Teléfax: 7944-0004 / 7944-0172 / 7944-0229

Americana 2000 Quezaltepeque
3ª. Ave. 2-60, Zona 2.
Tel: 7944-0508

Concepción Las Minas
Calle Principal
Tels: 7943-5284 / 7943-5286 Fax: 7943-5127

Ipala
3ª. Ave. 1-75, Zona 1
Tels: 7942-8023 / 7942-8351

La Unión Zacapa
Calle Principal 1-78, Zona 1
Barrio El Centro
Tel: 7941-8200

Olopa
Calle Principal, Barrio El Centro
Tel: 7873-0820 / 7823-5367
7723-5869 / 7723-5867

Chiquimula
3ª. Calle 10-80, Zona 1
PBX: 7790-0707

Pradera Chiquimula
Centro Comercial Pradera Chiquimula
Tels: 7942-1709

San Luis Jilotepeque
Frente al Parque Central
Tels: 7923-7482 / 7923-7494 / 7820-9830

Ciudad Capital
Edificio Atlantis Local 36 y 37
2º. Nivel 13 Calle 3-40, Zona 10
Teléfax: 2363-2117 / 2363-2120 / 2363-2137

Santo Tomás de Castilla
Distrito Portuario Casa N°. 7
Tel: 7960-0514

Agencia Mountkisco USA
(914) 2421-877 / (914) 2421-878