UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE COMPUTACIÓN

BUSCADOR INTERNO WEB CON PROCESAMIENTO DEL LENGUAJE NATURAL Y MÉTRICAS DE SIMILITUD DE INTELIGENCIA ARTIFICIAL TOMANDO COMO CASO DE ESTUDIO UN E-COMMERCE

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Ciencias de la Computación

AUTORES: HERIG ALEXANDER RECALDE MORALES
LENIN SANTIAGO SORIA COLUMBA
TUTOR: DIEGO FERNANDO VALLEJO HUANGA

Quito - Ecuador
2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Herig Alexander Recalde Morales con documento de identificación N° 1720849346 y Lenin Santiago Soria Columba con documento de identificación N° 1726334616; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 11 de marzo del año 2022

Atentamente,

Herig Alexander Recalde Morales
1720849346

Lenin Santiago Soria Columba
1726334616

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Herig Alexander Recalde Morales con documento de identificación No. 1720849346 y Lenin Santiago Soria Columba con documento de identificación No. 1726334616, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Académico: "Buscador Interno Web con Procesamiento del Lenguaje Natural y Métricas de Similitud de Inteligencia Artificial tomando como Caso de Estudio un E-commerce", el cual ha sido desarrollado para optar por el título de: Ingeniero en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 11 de marzo del año 2022

Atentamente,

Herig Alexander Recalde Morales
1720849346

Lenin Santiago Soria Columba
1726334616

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Diego Fernando Vallejo Huanga con documento de identificación N° 1720162708, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: BUSCADOR INTERNO WEB CON PROCESAMIENTO DEL LENGUAJE NATURAL Y MÉTRICAS DE SIMILITUD DE INTELIGENCIA ARTIFICIAL TOMANDO COMO CASO DE ESTUDIO UN E-COMMERCE, realizado por Herig Alexander Recalde Morales con documento de identificación N° 1720849346 y por Lenin Santiago Soria Columba con documento de identificación N° 1726334616, obteniendo como resultado final el trabajo de titulación bajo la opción Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 11 de marzo del año 2022

Atentamente,

Ing. Diego Fernando Vallejo Huanga, MSc
1720162708

Buscador Interno Web con Procesamiento del Lenguaje Natural y Métricas de Similitud de Inteligencia Artificial tomando como Caso de Estudio un E-Commerce.

1st Herig Recalde-Morales
hrecalde@est.ups.edu.ec

2nd Lenin Soria-Columba
lsoriac@est.ups.edu.ec

3rd Diego Vallejo-Huanga
dvallejoh@ups.edu.ec

***Resumen*—En los últimos años, el Procesamiento Natural del Lenguaje ha contribuido al avance de diferentes áreas y aplicaciones como reconocimiento de voz, motores de búsqueda web, minería social, etc. Este artículo aplica técnicas de Procesamiento Natural del Lenguaje para el desarrollo de un buscador interno web y sistema recomendador en un *e-commerce*. El sistema está desplegado en una PaaS y cuenta con una base de datos *NoSQL* en la que se almacenó un *dataset* con 100 documentos con información sobre obras literarias. Cada documento está estructurado por 15 campos y para la implementación del sistema se consideraron solo seis campos de los documentos almacenados. El funcionamiento del buscador compara una consulta con el *corpus* de los documentos mediante el coeficiente de *Jaccard*, el coeficiente de *Sorensen-Dice*, y el coseno de *Salton*. Además, el sistema recomendador aplica métricas de similitud para encontrar libros similares a los últimos visualizados por el usuario. Las herramientas fueron validadas mediante pruebas funcionales y no funcionales. Las pruebas funcionales usaron validación humana, por medio de una encuesta de satisfacción a un grupo etario de 25 personas entre 23 y 25 años con educación universitaria. Los resultados mostraron que al menos un 65% de los usuarios calificaron a las herramientas con el nivel máximo de satisfacción. Para las pruebas no funcionales se realizaron pruebas de estrés y carga, obteniendo una latencia elevada en ciertos casos, debido a que se utilizaron servicios gratuitos.**

***Palabras Clave*—Aprendizaje de Máquina, Recuperación de la Información, Similitud Semántica, Motor de Búsqueda.**

***Abstract*—Nowadays, Natural Language Processing contributes to the advancement of different areas and applications such as voice recognition, web search engines, social mining, etc. This scientific article applies Natural Language Processing techniques for the development of an internal web search engine and recommender system in an *e-commerce*.**

The system is deployed in a PaaS with a *NoSQL* database, and a *dataset* with 100 documents about literary works. Each document is structured by 15 fields, and for the implementation of the system, only six document fields' were considered. The behavior of the search engine compares a query with the *corpus* of documents using the *Jaccard* coefficient, the *Sorensen-Dice* coefficient, and the *Salton* cosine. Also, the recommender system applies similarity metrics to find books similar to the last ones viewed by the user. The tools were validated through functional and non-functional tests. The functional tests used human validation through a satisfaction survey in a group of 25 people between 23 and 25 years old with a university education. The results showed that at least 65% of users rated the tools with the highest level of satisfaction. For the non-functional tests, stress and load tests were carried out, obtaining a high latency in some cases, due to the fact that free services were used.

***Keywords*—Machine Learning, Information Retrieval, Semantic Similarity, Search Engine.**

I. Introducción

El avance tecnológico en los últimos años ha permitido que más organizaciones, con diferentes servicios y productos, implementen sus plataformas en Internet, destacándose: *e-commerce*, redes sociales y servicios de *streaming* [1]. Al contar con grandes volúmenes de información, es necesario tener sistemas de búsqueda avanzados que permitan encontrar elementos con un grado de precisión alto. De esta manera, aparecen buscadores especializados, para facilitar la navegación y mostrar sólo los resultados que le interesen al usuario.

Por otro lado, el Procesamiento del Lenguaje Natural (NLP) comprende diferentes áreas de Inteligencia Artificial (IA) [2] y lingüística. Existen varias aproximaciones en este campo, con diferentes aplicaciones. Una de las más relevantes fue planteada en 1950 para la traducción automática entre un lenguaje natural y otro, encontrándose varias vicisitudes, por lo que su uso no se masificó durante varias décadas [3].

Los avances en el campo del *Machine Learning* (ML) permitieron al NLP adoptar nuevos enfoques con modelos predictivos que aprenden directamente de un conjunto de textos. Esto aportó al mejoramiento de diferentes áreas y aplicaciones como: reconocimiento de voz, motores de búsqueda web, *machine translation* y minería social.

Los motores de búsqueda tienen diferentes aplicaciones, tanto en un entorno de escritorio como en uno de tipo *enterprise* [4]. A inicios de los años 70, del siglo pasado, se creó un sistema de búsqueda de literatura médica en línea donde en un principio, el *hardware* especializado para la búsqueda de texto se lo denominó "motor de búsqueda". En la década de 1980, esta terminología empezó a ser utilizada para referirse al *software* que compara consultas con documentos, entregando sus resultados en forma de listas indexadas [5].

En el entorno web, un motor de búsqueda debe indexar cientos de millones de páginas, que generan ingentes cantidades de datos. Esta herramienta debe atender a millones de consultas

diarias en el menor tiempo posible. Además, un motor de búsqueda debe gestionar el espacio de manera eficiente. La evolución los motores de búsqueda está directamente relacionada con el crecimiento tecnológico y la inmensa cantidad de información que necesita ser buscada por los usuarios de Internet [6].

En la actualidad el NLP, ha ganado bastante importancia en muchos ámbitos como traductores, aprendizaje automático, sistemas expertos, minería de datos, recuperación de información, etc [7]. La aplicación de NLP para el preprocesamiento de la consulta permite mejorar la eficiencia y precisión de la búsqueda, puesto que, se eliminan palabras y caracteres que resultan irrelevantes. Además, se reducen las palabras a sus raíces lo que permite que la búsqueda tenga una mayor cantidad de resultados en los que aparezcan términos derivados de la consulta [8].

En este sentido, la eficacia de un buscador es un factor decisivo para satisfacer las demandas de los usuarios, puesto que, del buscador dependerá que el usuario encuentre los resultados esperados o los más similares [9]. Los buscadores convencionales no implementan un preprocesamiento de la información, lo que causa, que la consulta del usuario aparezca íntegra en los resultados [10]. El desarrollo de un buscador es importante para garantizar que el usuario se sienta cómodo al interactuar con un sitio web. Un buscador también sirve para evaluar cómo se comporta el usuario, detectando sus intereses y promoviendo que las ventas se realicen de forma efectiva [11].

Entonces, dentro de los problemas que genéricamente se encuentran en un buscador está el gran volumen de información que contienen las plataformas. Esto puede afectar a las búsquedas, arrojando resultados no deseados o largos tiempos de espera. Otra causa es la falta de adaptabilidad al lenguaje humano [9]. Un buscador convencional no puede relacionar palabras morfológicamente similares. Los efectos que producen dichos problemas conllevan a la saturación del usuario, por el exceso de contenido ofrecido en la página web y desinterés debido a la dificultad de encontrar lo que se requiere.

Dentro de los buscadores podemos diferenciar dos tipos principales, los externos que se basan en la búsqueda e indexación de documentos de distintos sitios web y los internos se encargan de buscar elementos dentro del propio sitio web [12]. Dada la coyuntura y masificación de los servicios de Internet, los buscadores externos han crecido aceleradamente en los últimos años y han incorporado varias técnicas de IA para mejorar su rendimiento [13]. Gran parte de los buscadores internos consumen algún servicio o API de los buscadores externos. Ergo, el desarrollo de buscadores internos especializados con técnicas de NLP, ha sido escaso.

II. Trabajos Relacionados

Varios artículos científicos propugnan sistemas de recomendación (RS) y buscadores, basados en conjuntos de datos. Así, Nikishina et al. [14] aplicaron una metodología experimental, con un *dataset* de 400 artículos implementando tres modelos: *TF-IDF, LDA* y *Paragraph Vector*. El propósito de los experimentos fue conocer que algoritmo realizaba la búsqueda y recomendación más eficiente dentro del conjunto de datos. El resultado obtenido fue que *TF-IDF* superó a los demás algoritmos, en eficiencia. La mejora en el *performance* de *TF-IDF* estaba directamente relacionada al *corpus* pequeño que se usó para el entrenamiento.

Por otro lado, el estudio realizado por [15] se desarrolló bajo una metodología de investigación cuantitativa, donde se creó un almacén de datos eficiente (MolecularDB), adaptado a la genómica y los datos de sus variantes. La construcción del motor de búsqueda del genoma humano (VarSome) permitió a los médicos e investigadores realizar búsquedas por nombre de gen, símbolo de transcripción, ubicación genómica, ID de variante o nomenclatura HGVS. Además, la herramienta de búsqueda posibilita el ingreso de nuevas variantes. VarSome se constituye como una base de conocimiento, debido a que, cuenta con más de 56 000 usuarios de 120 países que aportan al sistema.

En [16] se propuso el desarrollo de un motor de búsqueda de noticias personalizado, con contenido recopilado mediante un proceso de extracción de una página web de noticias. Se manejó una metodología experimental, ya que para la evaluación del sistema se utilizaron dos métricas diferentes *precision* y *recall*. Como resultado se obtuvo que el método propuesto basado en semántica, supera al método convencional basado en la frecuencia. Entonces, la herramienta de minería de datos recopiló de forma exitosa la información relacionada de múltiples fuentes de noticias. En la metodología se realizó una limpieza de datos y se buscaron las relaciones semánticas para proporcionar solo información relevante al usuario.

En la investigación de Giles et al. [17] se presenta a *CiteSeer*, un sistema de indexación de citas autónomo. Esta herramienta tiene la capacidad de crear bases de datos más actualizadas, que no están limitadas a un conjunto preseleccionado de revistas. *CiteSeer* no proporciona un índice tan completo como en los sistemas tradicionales, debido a que muchas publicaciones no estaban disponibles en línea.

El estudio realizado por [18] propuso un motor de búsqueda de nicho *eBizSearch* enfocado en el campo de *e-Business*, basado en la tecnología de *CiteSeer*. El objetivo principal del sistema fue la indexación de publicaciones académicas relacionadas al campo de *e-Business* y pretendía cumplir con la Iniciativa de Archivos Abiertos (OAI). Los resultados de la aplicación de un algoritmo de aprendizaje automático, *Support Vector Machine* (SVM), mejoraron la extracción automática y aumentó la precisión de texto no etiquetado.

Esta investigación propugna desarrollar un buscador web con técnicas de NLP y métricas de similitud de IA tomando como caso de estudio un *e-commerce*, dado que la literatura científica refleja la inexistencia de un buscador web interno con estas características. Además, a partir de las mismas técnicas, se implementará un sistema recomendador basado en el historial de visualizaciones del usuario. La solución propuesta facilitará la navegación del usuario y brindará una asesoría personalizada de manera automática en el *e-commerce*.

III. Metodología y Materiales

A. Metodologías de Desarrollo y Técnicas de NLP

El sistema propuesto utilizará la metodología de desarrollo *CRoss-Industry Standard Process for Data Mining* (CRISP-DM) ya que suele ser uno de los estándares de facto para el desarrollo de tareas de minería de datos y proyectos de descubrimiento de conocimiento [19]. Además, esta es una metodología holística que incluye los resultados obtenidos en el entorno de negocio. Este proyecto de investigación, dada la naturaleza del caso estudio en un *e-commerce*, requiere una tecnología estándar para la explotación de los datos en sistemas industriales [20]. Las etapas del proceso metodológico, para el caso de estudio, incluyen la comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue.

Por otro lado, para el desarrollo del software se usará la metodología ágil *Scrum* [21] sobre el modelo tradicional en cascada, para tener una retroalimentación continua por parte del usuario, mediante el *product owner*. La metodología maneja *sprints*, para delimitar y describir la finalidad de los ciclos de trabajo y plasmarlos como módulos funcionales del sistema, que permiten correcciones oportunas a-posteriori [22]. Los *sprints* que se desplegarán, incluyen la construcción del *e-commerce*, implementación de métricas de similitud, construcción del buscador interno web e implementación del sistema de recomendación.

El sistema recibe como entrada un conjunto de datos con textos que le permiten establecer parámetros de similitud y recomendaciones. Por lo tanto, es necesario realizar procesos de limpieza y curación de los datos, antes de que estos ingresen en los procesos de NLP. En el módulo de NLP el sistema realiza la normalización de los *corpus*, que consiste en la eliminación de signos de puntación, tildes y caracteres especiales, además de convertir toda la cadena a minúsculas. Luego, el sistema realiza un proceso de tokenización de las consultas ingresadas por el usuario y el *dataset* embebido en la plataforma, para segmentar los *corpus* en unidades lingüísticas llamadas *tokens*. Con la finalidad de disminuir el tamaño del diccionario de *tokens* extraídos, se realiza un proceso de eliminación de palabras vacías (*stopwords*). Las *stopwords* son *tokens* que no aportan con semántica al contenido lingüístico del texto y que generalmente pertenecen a las categorías gramaticales de artículos, preposiciones o pronombres. Finalmente, se realiza un proceso de *stemming* que reduce los *tokens* a sus raíces, eliminando los sufijos.

Una vez preprocesada la información, se utilizarán tres métricas de similitud coeficiente de *Jaccard*, coseno de *Salton* y coeficiente de *Sorensen-Dice* de acuerdo con el tipo texto, para encontrar resultados similares a la consulta. La similitud coseno es útil cuando el *corpus* es extenso, por lo tanto se usará en campos que cumplan esta característica. El coeficiente de *Sorensen-Dice* es tolerante a erratas, a diferencia del índice de *Jaccard*. Ergo, se usará *Sorensen-Dice* para los campos en los que resulte más fácil cometer algún error. Los resultados obtenidos se ordenarán en forma descendente según su índice de similitud. La Figura 1 es una representación gráfica del esquema metodológico que se propone en esta investigación.

B. Descripción del Conjunto de Datos

El *dataset* tiene por objetivo almacenar en el ambiente web del *e-commerce*, un conjunto de libros que permitan probar las recomendaciones y los algoritmos de similitud del buscador web interno. La construcción del *dataset* fue realizada de forma manual y toma en consideración títulos de libros con diferentes géneros y extensión.

Así, el conjunto de datos es una colección de documentos, donde cada instancia es un libro D_i $(i = 1, ..., n)$, siendo

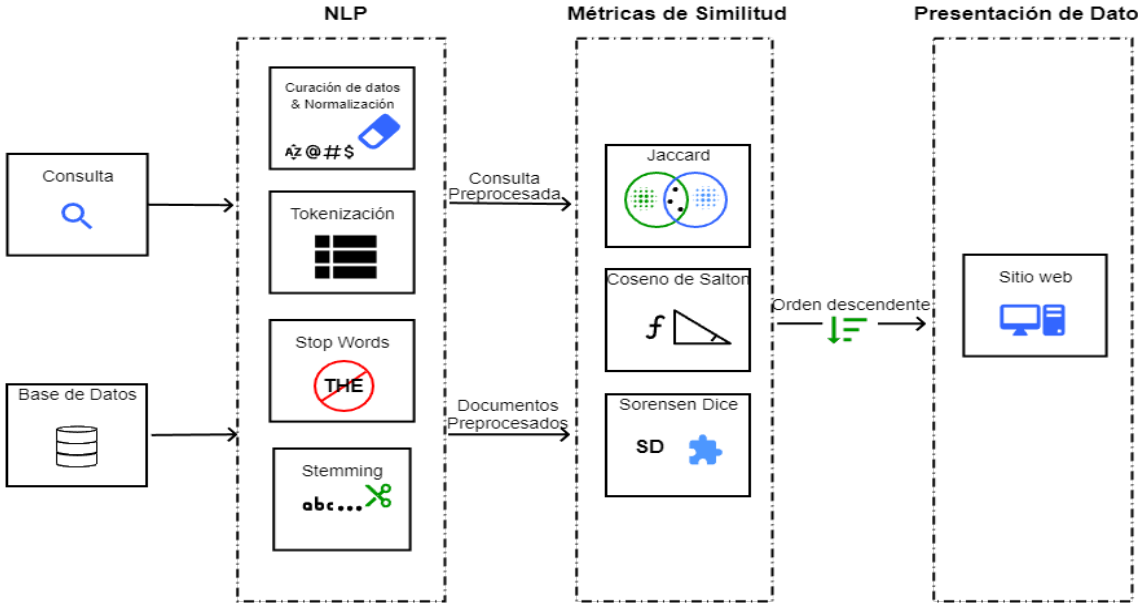

Fig. 1: Diagrama de bloques del sistema recomendador y del buscador interno web

$n = 100$ la cantidad total de libros. Cada D_i consta de 15 campos (variables), nueve de estos serán utilizados para tareas administrativas del *e-commerce*, tales como: número de existencias, unidades vendidas, identificador del libro, etc. Los seis campos restantes, se utilizan en el desarrollo del buscador y sistema recomendador, puesto que aportan información relevante del libro. Las seis variables seleccionadas son Título (T_i), Autor (A_i), Editorial (E_i), Categoría (C_i), Género (G_i) y Descripción (S_i).

El *dataset* consta de D_i en idioma español en su totalidad, pero algunos títulos, autores y/o editoriales son de origen extranjero, por lo que es posible que existan palabras que no sean propias de la lengua española. El campo C_i describe el tipo de obra al que pertenece cada libro y se definieron tres taxonomías literatura, novela y teatro. En el campo G_i se especifican los temas que aborda la obra y son 45 en total.

En la Tabla I se pueden observar los campos asociados a cada módulo del sistema web. Además, se detalla el rango que indica el número mínimo y máximo de *tokens* y los valores estadísticos de media μ, y desviación estándar σ, para cada campo. Estos valores son detallados para los D_i antes y después de aplicar las técnicas de NLP.

Tabla I: Campos del conjunto de datos y medidas estadísticas de los documentos con NLP y sin NLP

		Sin NLP		Con NLP	
Campo	**Módulo**	**Rango**	$\mu \pm \sigma$	**Rango**	$\mu \pm \sigma$
Autor	Buscador, RS	[1, 4]	2.20 ± 0.51	[1, 4]	2.15 ± 0.51
Categoría	Buscador, RS	[2, 2]	2.00 ± 0.00	[2, 2]	2.00 ± 0.00
Género	Buscador, RS	[1, 9]	3.36 ± 1.53	[1, 8]	3.27 ± 1.53
Título	Buscador	[1, 8]	3.81 ± 1.85	[1, 5]	2.37 ± 2.35
Editorial	Buscador	[1, 4]	1.59 ± 0.76	[1, 4]	1.57 ± 0.76
Descripción	Buscador	[76, 218]	144.59 ± 37.38	[50, 125]	87.48 ± 71.63

En el caso de A_i, C_i, G_i, T_i y E_i la media de *tokens* es baja con respecto al campo S_i, por lo tanto, se determina que el campo *Descripción* tiene un *corpus* extenso. Por otro lado, σ nos indica la dispersión de los datos con respecto a la media de *tokens* por campo. El único campo que tiene un valor de σ elevado es S_i, puesto que, varía dependiendo de la extensión del libro y el nivel de detalle que se muestre. Después de aplicar NLP se reduce el rango y la media de *tokens*, esto es más latente para el campo S_i, donde la reducción de la cantidad de *tokens* es mayor y permitirá disminuir el coste computacional con respecto al cálculo de las métricas de similitud.

C. Desarrollo del Back-end y Front-end de la Aplicación

Para el desarrollo de la aplicación se utilizará JavaScript como lenguaje de programación y NodeJS como entorno de ejecución. Además, el diseño del sistema está dividido en *Back-end* y *Front-end*, usando el modelo cliente-servidor. El *e-commerce* se desplegará en la plataforma PaaS [23] *Heroku* para permitir el acceso al sitio web y como herramienta de banco de pruebas. La Figura 2, muestra la arquitectura propuesta, mediante un diagrama de bloques.

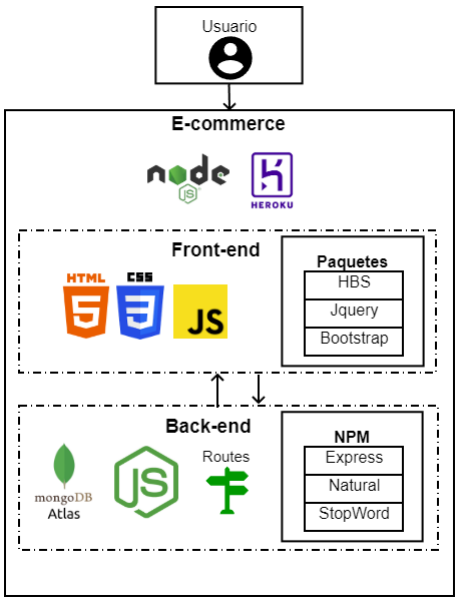

Fig. 2: Arquitectura de la plataforma web para el *e-commerce*

La interfaz de usuario del *Front-end* utiliza *HTML5* y *CSS3* para el diseño del sitio web. También se usarán librerías como *Bootstrap* para conseguir un diseño *responsive*, el sistema de plantillas *Handlebars* (HBS) para manejar elementos *HTML* de forma sencilla y rápida, y la librería *jQuery* para realizar peticiones *AJAX* con el objetivo de que el contenido sea dinámico.

Por su parte, el *Back-end* utilizará la librería *Express* como base, sobre la cual se manejarán rutas, creando así interacciones dinámicas entre el cliente y servidor. Para el almacenamiento de la información se empleará *MongoDB Atlas*, un gestor de base de datos *NoSQL* en la nube. *MongoDB* tiene un mejor rendimiento en las consultas comparado con un gestor *SQL* tradicional [24] y además facilita el manejo de información no estructurada, para el trabajo con colecciones de documentos.

El gestor de paquetes de NodeJS (NPM), ofrece varias librerías para ejecutar tareas de limpieza y normalización de NLP. Para el desarrollo del sistema se emplearán las librerías *StopWord* y *Natural*. Estos paquetes se usarán para la eliminación de palabras vacías, el *stemming* y el cálculo del coeficiente de *Sorensen-Dice*.

D. Descripción de la Herramienta y Algoritmos

El *e-commerce* tomado como caso de estudio, está orientado a la venta de libros y se puede consumir en el siguiente enlace: https://bookbrary-beta.herokuapp.com, además el código fuente está disponible en https://github.com/Herig14/bookbrary. La arquitectura de la plataforma contempla módulos que determinan las diferentes funcionalidades a las que el usuario puede acceder. Un usuario no registrado que ingrese al sitio web puede interactuar con las páginas que muestran los artículos y el carrito de compras. El usuario tiene la opción de registrarse al sitio web, con la finalidad de

tener un perfil que servirá para realizar compras y visualizar el estado de sus pedidos, además podrá modificar sus datos personales. El *e-commerce* cuenta con roles de administrador y usuario. El administrador tiene disponibles las funcionalidades para gestionar usuarios, productos y puede monitorear las ventas, mediante el apartado de pedidos.

Los usuarios pueden interactuar con la plataforma de *e-commerce*, mediante páginas que les permiten realizar búsquedas o encontrar recomendaciones. La página principal, contiene dos módulos, el buscador en la parte superior y el sistema recomendador en la sección llamada *Relacionado con libros que has mirado*. Además, se encuentra la sección *Populares*, dónde se podrá visualizar los diez libros con mayor número de ventas, como se muestra en la Figura 3.

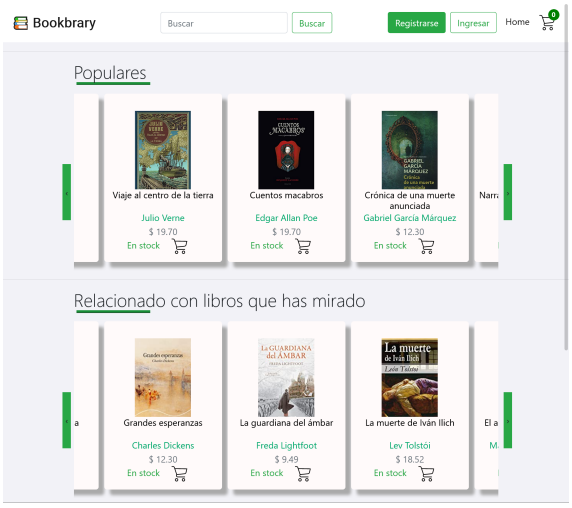

Fig. 3: Vista de la página principal de la plataforma web

Otra página con la que puede interactuar el usuario es la que se genera para cada libro D_i, dentro de la colección de documentos. En esta página se pueden visualizar los detalles del libro y al lado derecho la sección de *Relacionados*, que desplegará una lista con libros similares al observado. El módulo del buscador también se encuentra disponible en la parte superior de esta página y puede ser utilizado por el usuario para explorar los diferentes artículos del *e-commerce*.

Una vez que el usuario ingrese a la página principal, tendrá la opción de usar el buscador, donde debe ingresar una consulta Q_k para que inicie el proceso de búsqueda. Después de enviar la consulta se aplican técnicas de NLP a Q_k y D_i para ser comparadas mediante métricas de similitud, donde cada D_i contiene los seis campos detallados en la Tabla I.

En los campos T_i, A_i y E_i se utilizará el coeficiente de *Sorensen-Dice*, SDC, como se describe en la Ec. 1. Este algoritmo compara dos cadenas de texto divididas en bigramas, lo cual otorga una mayor precisión y tolerancia a posibles fallos ortográficos y erratas.

$$\left\{ SDC = \frac{2|Q_k \cap D|}{|Q_k| + |D|} \,\middle|\, D \subset \{T_i, A_i, E_i\} \right\} \tag{1}$$

Para C_i y G_i se usará el coeficiente de *Jaccard* (ver Ec.2), donde Q_k y D_i son dos conjuntos de *tokens*.

$$\left\{ J(Q_k, D) = \frac{|Q_k \cap D|}{|Q_k \cup D|} \,\middle|\, D \subset \{C_i, G_i\} \right\} \tag{2}$$

Finalmente, para medir la similitud entre la S_i del libro y la consulta Q_k se utilizará la métrica de similitud de coseno de *Salton* de la Ec. 3, sobre una bolsa de palabras en la que se aplicó TF-IDF. Se usa esta técnica para ponderar con mayor peso a los *tokens* poco frecuentes y que se repiten con mayor frecuencia, considerando que S_i contiene una mayor extensión de *corpus* en su diccionario.

$$\left\{ \cos(\vec{Q_k}, \vec{D}) = \vec{Q_k} \cdot \vec{D} = \sum_{l=1}^{|V|} Q_k D \,\middle|\, D \subset \{S_i\} \right\} \tag{3}$$

Los resultados obtenidos al aplicar las métricas de similitud se almacenan para cada una de las seis variables consideradas, con sus respectivas s puntuaciones en T_{i_s}, A_{i_s}, E_{i_s}, C_{i_s}, G_{i_s} y S_{i_s} y se suman para obtener una puntación final. Estos valores serán ordenados de forma descendente y almacenados en un arreglo B_j $(j = 1, 2, 3, ..., m)$, donde m es el número de libros relacionados con la consulta. Luego, se desplegará la lista de resultados, donde el usuario podrá ver algunos metadatos del libro de interés. En el Algoritmo 1 se puede observar los pasos para el funcionamiento del buscador web.

Algoritmo 1 Buscador Interno Web

Entrada: Q_k, D
Salida: R
Paso 1: Inicialización
$Q_k \leftarrow$ "$Viaje$";
$D \leftarrow [1, 2, ..., n]$;
$B \leftarrow [1, 2, ..., m]$;
Paso 2: NLP consulta
$Q_k \leftarrow NLP(Q_k)$;
for $i \leftarrow 1, n$ **do**
Paso 3: NLP documento
    $D_i \leftarrow NLP(i)$;
Paso 4: Métricas de similitud
    $T_{i_s} \leftarrow Dice(Q_k, T_i)$;
    $A_{i_s} \leftarrow Dice(Q_k, A_i)$;
    $E_{i_s} \leftarrow Jaccard(Q_k, E_i)$;
    $C_{i_s} \leftarrow Jaccard(Q_k, C_i)$;
    $G_{i_s} \leftarrow Jaccard(Q_k, G_i)$;
    $S_{i_s} \leftarrow Coseno(Q_k, S_i)$;
    $Total \leftarrow TS + AS + ES + CS + GS + DS$;
    if $Total > 0.1$ **then**
        $B.push(Total)$;
    end if
end for
Paso 5: Ordenar Resultados
$B \leftarrow OrdenDescendente(B)$;

El RS está compuesto por dos módulos, el primero se ejecuta al ingresar a la descripción del libro D_i, donde se desplegará una lista de libros relacionados R. El segundo módulo se encuentra en la página principal y muestra, en la sección de *Relacionado con libros que has mirado*, los artículos relacionados a los últimos tres libros visualizados $D_{i_{z1}}$, $D_{i_{z2}}$ y $D_{i_{z3}}$ por el usuario módulo.

El funcionamiento de ambos módulos del sistema recomendador es similar. La única diferencia es que Q_k en el primer caso es el libro observado y en el segundo caso Q_k contiene los tres últimos libros visualizados. Antes de la comparación con métricas de similitud, se realiza un filtrado en la consulta a la base de datos para descartar aquellos libros que no tengan coincidencias en ninguno de los campos A_i, C_i y G_i, evitando procesamiento innecesario. De los datos obtenidos se mide la similitud con el coeficiente de *Jaccard* y se multiplica por las ponderaciones $WA = 0.3$, $WC = 0.2$ y $WG = 0.5$, correspondientes a cada campo. Los valores de las ponderaciones obedecen a una heurística donde se considera que el género es el campo más importante, ya que detalla los temas que aborda el libro. En segundo lugar se encuentra el campo autor, ya que generalmente los autores tienen una línea literaria predefinida y sus libros tienen temáticas similares. Por último, el campo categoría se considera el menos importante porque únicamente indica el tipo de obra y por lo tanto los géneros que aborde pueden ser muy variados. El producto obtenido se almacenará en las variables AS_p, CS_p, GS_p y su suma se asignará a la variable S_p. Una vez realizada la comparación con los tres libros que contiene la Q, se calcula el promedio con los resultados y se almacena en $R_j(j = 1, 2, 3, ..., m)$, para posteriormente, ordenarlos y presentarlos en el respectivo módulo. En el Algoritmo 2 se puede visualizar el proceso para que el sistema ofrezca una recomendación al usuario.

Algoritmo 2 Sistema Recomendador

Entrada: Q_k, D
Salida: R
Paso 1: Inicialización
$Q_k \leftarrow [D_{iz1}, D_{iz2}, D_{iz3}]$;
$Q_k \leftarrow [D_{iz1}]$;
$R \leftarrow [1, 2, ..., m]$;
Paso 2: Consulta a la base de datos
$D \leftarrow BDD.query(D_{iz1}, D_{iz2}, D_{iz3})$;
Paso 3: NLP consulta
$Q_k \leftarrow NLP(Q_k)$;
for $i \leftarrow 1, m$ **do**
Paso 4: NLP documento
    $D_i \leftarrow NLP(D_i)$;
Paso 5: Ponderaciones
    $WA \leftarrow 0.3$;
    $WG \leftarrow 0.5$;
    $WC \leftarrow 0.2$;
Paso 6: Métricas de similitud
    if $A_{iz1} == A_i$ **then** $AS_1 = WS$ **else** $AS_1 = 0$
    $CS1 \leftarrow Jaccard(C_{iz1}, C_i) * WG$;
    $GS1 \leftarrow Jaccard(G_{iz1}, G_i) * WC$;
    $S1 \leftarrow AS1 + CS1 + GS1$;
    if $A_{iz2} == A_i$ **then** $AS_2 = WS$ **else** $AS_2 = 0$
    $CS2 \leftarrow Jaccard(C_{iz2}, C_i) * WG$;
    $GS2 \leftarrow Jaccard(G_{iz2}, G_i) * WC$;
    $S2 \leftarrow AS2 + CS2 + GS2$;
    if $A_{iz3} == A_i$ **then** $AS_3 = WS$ **else** $AS_3 = 0$
    $CS3 \leftarrow Jaccard(C_{iz3}, C_i) * WG$;
    $GS3 \leftarrow Jaccard(G_{iz3}, G_i) * WC$;
    $S3 \leftarrow AS3 + CS3 + GS3$;
Paso 7: Calcular promedio de puntuaciones
    $R[j] \leftarrow (S1 + S2 + S3)/3$;
end for
Paso 8: Ordenar Resultados
$R \leftarrow OrdenDescendente(R)$;

E. Sistema de Validación de la Plataforma Web

El sitio web que contiene los módulos de buscador y recomendador será evaluado mediante un sistema de validación humana, i.e., utilizando el criterio de satisfacción de varios usuarios de la plataforma. Para este propósito se utilizó una encuesta que permitió recolectar las opiniones de usuarios tanto del sistema recomendador como del buscador. El grupo etario esta conformado por 25 estudiantes de una Universidad privada del Ecuador, que se encuentran cursando los últimos niveles de una Ingeniería en Ciencias de la Computación, con un rango de edad entre 23 y 25 años. La muestra es estratificada para usuarios que conocen acerca de temas relacionados con el objetivo de este artículo científico, por lo que se espera al menos un nivel medio de experiencia en el manejo de sistemas web y Tecnologías de la Información (TI), además de conocer de manera tácita como se utiliza el buscador de un sitio web. La muestra tiene una prevalencia del género masculino con un 72% sobre el femenino.

El diseño de la encuesta evaluó dos aspectos mediante una escala de *likert* [25], graduada de uno a cinco, donde el valor de uno indica un grado nulo de satisfacción y cinco el máximo valor de satisfacción posible.

El primer aspecto tiene por objetivo evaluar el grado de satisfacción del buscador web, para esto se proponen tres tipologías de búsqueda, en las que se aumenta el número de *tokens* y la complejidad de la consulta. La primera búsqueda propuesta contiene un solo *token*, las segunda dos y la última es una consulta que utiliza procesos de NLP. Luego, como segundo aspecto el sistema recomendador será evaluado en función de los libros que el usuario haya seleccionado en las tres búsquedas anteriores y también se evalúa el nivel de satisfacción. Las cadenas de *tokens* para la evaluación del sistema están predefinidas y no son seleccionadas arbitrariamente por el usuario, con el objetivo de evitar posibles errores ortográficos y controlar que el usuario no ingrese una consulta que no esté relacionada con los documentos en la base de datos.

IV. Experimentos

A. Pruebas Funcionales del Sistema

Luego de ejecutar la encuesta al grupo etario, los resultados para evaluar el módulo del buscador y sistema recomendador se dividieron en dos apartados. El primer apartado del buscador contiene tres preguntas y en cada una de ellas se varía la consulta propuesta al usuario. En la primera consulta se evalúa al sistema de búsqueda con el *token* predefinido "viaje". Los resultados muestran que un 84% de usuarios calificaron la respuesta de la búsqueda con el máximo nivel de satisfacción.

En la siguiente consulta se definieron dos *tokens* como entrada de la cadena de búsqueda y se obtuvo un 64% de personas con el máximo nivel de satisfacción. La cadena de búsqueda fue "tragedia romántica" y se observa que el nivel de satisfacción disminuye con respecto a la búsqueda de un solo *token*, dado que algunos documentos solo coinciden parcialmente con la consulta. Aún así, el 88% de usuarios

calificó el resultado con un grado de satisfacción en la escala de cuatro o cinco.

La última consulta tiene tres *tokens* y dos palabras vacías, "obras de misterio y suspenso". Los resultados muestran que el 72% de los usuarios calificaron al sistema con una puntuación de cinco. Es menester clarificar que para esta consulta se ejecutan tareas de NLP, tanto en la cadena búsqueda como en los documentos.

En la Figura 4 se puede visualizar un diagrama de caja y bigotes con los resultados obtenidos en las encuestas.

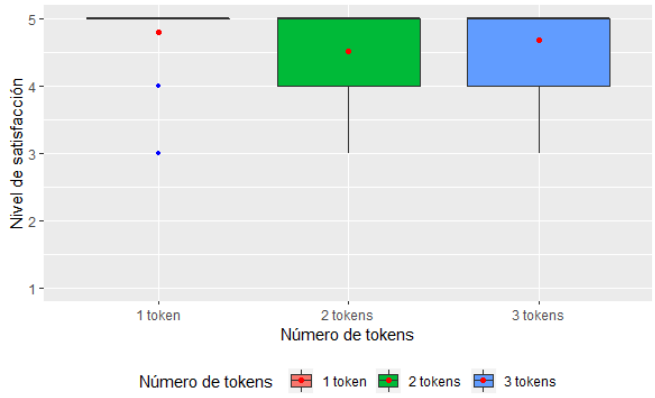

Fig. 4: Resumen de los niveles de satisfacción del buscador

Por otro lado, para el segundo apartado donde se evalúa al RS, es necesario que el usuario haya visualizado al menos tres libros, seleccionados en las consultas previas. Los resultados obtenidos indican un 68% de nivel máximo de satisfacción por parte de los usuarios. Esto denota que los libros recomendados son pertinentes en la mayoría de los casos.

B. Métricas de rendimiento de la Plataforma Web

Se realizaron pruebas no funcionales para evaluar el rendimiento de la plataforma web que almacena el *e-commerce*, con el objetivo de medir la respuesta del sitio web ante variaciones de concurrencia de usuarios en un intervalo de tiempo determinado. Para este propósito, se utilizó la herramienta k6 Cloud, que permite crear pruebas de carga y estrés de manera automatizada [26].

Fig. 5: Prueba de carga al utilizar el buscador

Las pruebas fueron ejecutadas sobre el módulo del buscador y en la página de un documento donde se muestran libros relacionados por el sistema recomendador. Las pruebas de carga se realizaron con 25 Usuarios Virtuales (VUs) en un período de tiempo de cuatro minutos. Por otro lado, para las pruebas de estrés se cargaron 10 VUs iniciales con incrementos de 10 VUs por minuto, durante cuatro minutos, alcanzando un total de 40 VUs.

Los resultados obtenidos de las pruebas de carga y estrés para el buscador se pueden visualizar en las Figuras 5 y 6, respectivamente. Las medidas presentadas en estos gráficos son la cantidad de solicitudes realizadas (reqs), solicitudes fallidas (reqs), el valor máxi mo de solicitudes por segundo (reqs/s) y el tiempo de respuesta promedio (ms).

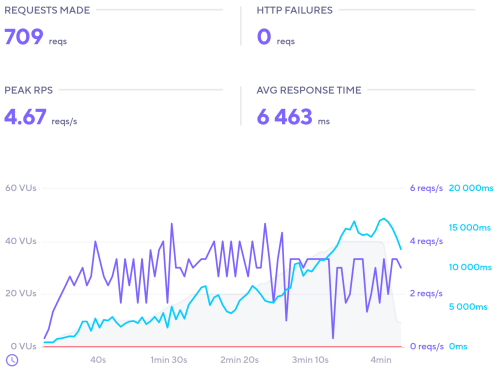

Fig. 6: Prueba de estrés al utilizar el buscador

En la Tabla II se resumen las métricas obtenidas de las pruebas de carga y estrés en los módulos implementados y además se calcula la desviación estándar, σ, para los dos casos. El estadístico σ indica la dispersión del tiempo entre las solicitudes realizadas. El valor de σ en las pruebas de estrés es mayor al obtenido en las pruebas de carga, porque la cantidad de VUs incrementa con el tiempo, a diferencia de las pruebas de carga donde los VUs son concurrentes la mayor parte del tiempo. Entonces, esto causa un mayor tiempo de respuesta y por ende una mayor dispersión en las pruebas no funcionales de estrés.

Tabla II: Resultados de pruebas de rendimiento de carga y estrés

Ruta	Prueba	Muestras	$\mu \pm \sigma$ (ms)	Rendimiento (req/seg.)	Error
/search	Carga	25	4067 ± 2055	6.67	0
/search	Estrés	40	6463 ± 4323	4.67	0
/book?isbn=8427213735	Carga	25	108 ± 80	37.33	0
/book?isbn=8427213735	Estrés	40	210 ± 107	37.00	0

Es menester aclarar que el clúster gratuito de *MongoDB Atlas* tiene un límite de conexiones y ancho de banda, que provoca una mayor latencia en las solicitudes.

V. Conclusiones y Limitaciones

En este trabajo se desarrolló un buscador y sistema recomendador con el uso de técnicas de NLP y métricas de similitud en un *e-commerce*, orientado a la venta de libros. En el desarrollo del buscador se observó que la aplicación de NLP mejora los resultados de una consulta, particularmente cuando presenta una estructura compuesta por múltiples *tokens* a manera de frase. Además, los resultados obtenidos en las encuestas reflejan que más del 64% de los usuarios califican las búsquedas y recomendaciones con el máximo nivel satisfacción, esto indica que tanto el sistema recomendador como el buscador interno web pueden ser utilizados en entornos de producción.

Dado que el sistema se despliega en una PaaS y una base de datos gratuita, las pruebas no funcionales presentaron limitaciones en torno a la latencia. La evaluación del sistema está limitado a un conjuntos de datos de 100 instancias, ergo, en trabajos futuros se propone la expansión del *dataset* para incluir más géneros y categorías literarias, con el objeto de mejorar las recomendaciones.

References

[1] S. Sivapalan, A. Sadeghian, H. Rahnama, and A. M. Madni, "Recommender systems in e-commerce," in *2014 World Automation Congress (WAC)*. IEEE, 2014, pp. 179–184.

[2] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[3] M. Bates, "Models of natural language understanding," *Proceedings of the National Academy of Sciences*, vol. 92, no. 22, pp. 9977–9982, 1995.

[4] B. Croft, D. Metzler, and T. Stronhman, "Search engines: Information retrieval in practice," Boston, 2010, p. 7.

[5] D. Sharma, R. Shukla, A. K. Giri, and S. Kumar, "A brief review on search engine optimization," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2019, pp. 687–692.

[6] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[7] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[8] Y. Kang, Z. Cai, C.-W. Tan, Q. Huang, and H. Liu, "Natural language processing (nlp) in management research: A literature review," *Journal of Management Analytics*, vol. 7, no. 2, pp. 139–172, 2020.

[9] X. Yue, G. Di, Y. Yu, W. Wang, and H. Shi, "Analysis of the combination of natural language processing and search engine technology," *Procedia Engineering*, vol. 29, pp. 1636–1639, 2012.

[10] U. Kruschwitz, "Exploiting structure for intelligent web search," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*. IEEE, 2001, pp. 9–pp.

[11] H.-p. Chan, L. Xu, H.-h. Liu, R.-t. Zhang, and A. K. Sangaiah, "System design of cloud search engine based on rich text content," *Mobile Networks and Applications*, vol. 26, no. 1, pp. 459–472, 2021.

[12] A. Ortiz-Cordova, Y. Yang, and B. J. Jansen, "External to internal search: Associating searching on search engines with searching on sites," *Information Processing & Management*, vol. 51, no. 5, pp. 718–736, 2015.

[13] D. Sánchez, L. Martínez-Sanahuja, and M. Batet, "Survey and evaluation of web search engine hit counts as research tools in computational linguistics," *Information Systems*, vol. 73, pp. 50–60, 2018.

[14] I. Nikishina, A. Bakarov, and A. Kutuzov, "Rusnlp: semantic search engine for russian nlp conference papers," in *International Conference on Analysis of Images, Social Networks and Texts*. Springer, 2018, pp. 111–120.

[15] C. Kopanos, V. Tsiolkas, A. Kouris, C. E. Chapple, M. A. Aguilera, R. Meyer, and A. Massouras, "Varsome: the human genomic variant search engine," *Bioinformatics*, vol. 35, no. 11, p. 1978, 2019.

[16] M. Kanakaraj and S. S. Kamath, "Nlp based intelligent news search engine using information extraction from e-newspapers," in *2014 IEEE International Conference on Computational Intelligence and Computing Research*. IEEE, 2014, pp. 1–5.

[17] C. L. Giles, K. D. Bollacker, and S. Lawrence, "Citeseer: An automatic citation indexing system," in *Proceedings of the third ACM conference on Digital libraries*, 1998, pp. 89–98.

[18] C. L. Giles, Y. Petinot, P. B. Teregowda, H. Han, S. Lawrence, A. Rangaswamy, and N. Pal, "ebizsearch: A niche search engine for e-business," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 413–414.

[19] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "The crisp-dm user guide," in *4th CRISP-DM SIG Workshop in Brussels in March*, vol. 1999. sn, 1999.

[20] F. Martínez-Plumed, L. Contreras-Ochando, C. Ferri, J. H. Orallo, M. Kull, N. Lachiche, M. J. R. Quintana, and P. A. Flach, "Crisp-dm twenty years later: From data mining processes to data science trajectories," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[21] M. Arora, S. Verma, S. Chopra *et al.*, "A systematic literature review of machine learning estimation approaches in scrum projects," *Cognitive Informatics and Soft Computing*, pp. 573–586, 2020.

[22] A. Srivastava, S. Bhardwaj, and S. Saraswat, "Scrum model for agile methodology," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2017, pp. 864–869.

[23] D. Beimborn, T. Miletzki, and S. Wenzel, "Platform as a service (paas)," *Business & Information Systems Engineering*, vol. 3, no. 6, pp. 381–384, 2011.

[24] B. Jose and S. Abraham, "Performance analysis of nosql and relational databases with mongodb and mysql," *Materials today: PROCEEDINGS*, vol. 24, pp. 2036–2043, 2020.

[25] A. N. Ghazi, K. Petersen, S. S. V. R. Reddy, and H. Nekkanti, "Survey research in software engineering: Problems and mitigation strategies," *IEEE Access*, vol. 7, pp. 24 703–24 718, 2018.

[26] Grafana Labs. K6 cloud. Accessed: 2022-18-01. [Online]. Available: https://k6.io/cloud/