



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE MOTORES DE BÚSQUEDA DE PRODUCTOS EN
PLATAFORMA ECOMMERCE UTILIZANDO PROCESAMIENTO DE
LENGUAJE NATURAL

VALAREZO PEREIRA MARVIN RICARDO
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE MOTORES DE BÚSQUEDA DE PRODUCTOS
EN PLATAFORMA ECOMMERCE UTILIZANDO
PROCESAMIENTO DE LENGUAJE NATURAL

VALAREZO PEREIRA MARVIN RICARDO
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

DESARROLLO DE MOTORES DE BÚSQUEDA DE PRODUCTOS EN PLATAFORMA
ECOMMERCE UTILIZANDO PROCESAMIENTO DE LENGUAJE NATURAL

VALAREZO PEREIRA MARVIN RICARDO
INGENIERO DE SISTEMAS

RIVAS ASANZA WILMER BRAULIO

MACHALA, 24 DE SEPTIEMBRE DE 2021

MACHALA
2021

marvin valarezo

INFORME DE ORIGINALIDAD

5%

INDICE DE SIMILITUD

4%

FUENTES DE INTERNET

0%

PUBLICACIONES

1%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	dspace.utb.edu.ec Fuente de Internet	1%
2	repositorio.ug.edu.ec Fuente de Internet	1%
3	www.slideshare.net Fuente de Internet	1%
4	docplayer.es Fuente de Internet	<1%
5	Submitted to Universidad Continental Trabajo del estudiante	<1%
6	www.adictosaltrabajo.com Fuente de Internet	<1%
7	biblioteca.usac.edu.gt Fuente de Internet	<1%
8	core.ac.uk Fuente de Internet	<1%
9	Submitted to Universidad Ricardo Palma Trabajo del estudiante	<1%

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, VALAREZO PEREIRA MARVIN RICARDO, en calidad de autor del siguiente trabajo escrito titulado DESARROLLO DE MOTORES DE BÚSQUEDA DE PRODUCTOS EN PLATAFORMA ECOMMERCE UTILIZANDO PROCESAMIENTO DE LENGUAJE NATURAL, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 24 de septiembre de 2021



VALAREZO PEREIRA MARVIN RICARDO
0705740926

DEDICATORIA

Este trabajo realizado lo dedico con todo esfuerzo a mis papas, hermanos, por darme ese apoyo incondicional, que han estado junto a mí en todo momento, han hecho de mí una mejor persona, me reprendieron para seguir ahí dando lo mejor de mí, todo lo que he conseguido hasta el día de hoy es gracias a todos los que me apoyaron.

Marvin Ricardo Valarezo Pereira

AGRADECIMIENTO

Quiero agradecer a Dios por mantenernos con salud, por haberme dado una grandiosa familia, unos padres extraordinarios y una familia única.

Quiero agradecer a mis padres por su ayuda incondicional, por ayudarme a crecer, seguir superándome y a mis tíos por su ayuda para culminar esta etapa.

Y por último agradecer al Ing. Wilmer Rivas por ser mi tutor, por su apoyo no solo en este trabajo, sino que en el transcurso de toda mi etapa universitaria además gracias a todos los docentes que me impartieron sus conocimientos en sus clases, y me brindaron su amistad.

Marvin Ricardo Valarezo Pereira

RESUMEN

Desde los inicios de la inteligencia artificial hasta la actualidad ha existido grandes avances tecnológicos, lo que se diría que era una ciencia ficción hoy es realidad. con el lapso de tiempo el ser humano se ha convertido muy ambicioso por lo que ha ido adquiriendo diversos conocimientos y potenciándolos hasta crear sistemas que ayuden a facilitar su vida diaria, como es el caso de los robots que son creados con el fin de hacer acciones que realiza el ser humano, el desarrollo de teléfonos inteligentes, asistentes virtuales, entre otros los cuales facilitan la comunicación.

Durante diversas generaciones la humanidad ha tenido que adaptarse a diferentes desastres que amenazan su existencia, entre uno de esos inconvenientes es la pandemia causada por el covid19 que obligo a la población a permanecer en cuarentena, de esta manera las actividades se redujeron, uno de los sectores que tuvo un crecimiento notable es el comercio online debido a que los usuarios comenzaron a visitar estos sitios para realizar sus compras.

Las plataformas e-commerce se centran más en el diseño, olvidando un punto importante que son funciones básicas como el buscador de los productos, lo cual es de gran importancia ya que los usuarios ingresan a la tienda online, buscan productos, muchas veces obteniendo resultados ambiguos o gran cantidad de información no relevante. Por ello, para cumplir con la finalidad de este proyecto, se ha planteado el desarrollo de un prototipo referente a un motor de búsqueda en plataformas e-commerce utilizando procesamiento de lenguaje natural.

El procesamiento de lenguaje natural (NLP), es muy importante dentro de la inteligencia artificial a su vez la NLP es trascendental dentro de la lingüística computacional, esto permite convertir el lenguaje natural en un mecanismo que las computadoras puedan entender.

Para concluir con la finalidad del proyecto se desarrolló una arquitectura del prototipo, este permitirá el procesamiento de los datos fase por fase. Organizando los datos de entrada para aplicar procesamiento de lenguaje natural, y como resultado la obtención de una respuesta.

Para la codificación se realizó mediante la utilización de diferentes herramientas: Java, Netbeans, JSP, Web Services y la librería Apache Lucene; además se desarrolla diferentes métodos y clases los cuales combinados con los métodos de la librería Lucene, realizan una búsqueda sintáctica.

La etapa de la evaluación se realizó mediante métricas de clasificación: precisión y exhaustividad o Recall. En la primera evaluación se realizó una consulta de producto, donde los resultados fueron del valor máximo 100%, con lo cual la puntuación de precisión y sensibilidad se acercaron a 1, esto indica que los valores de la respuesta son relevantes con una exactitud perfecta. En la segunda prueba se realizó una consulta con varias palabras donde los resultados arrojados en cuanto a la precisión son del 33%, sensibilidad del 50% lo cual indica que los valores en la búsqueda se repiten en varios productos, lo que demuestra que la precisión no es exacta y algunas respuestas no son relevantes. En la tercera evaluación se ingresó un dato incompleto para conocer el nivel de proximidad del buscador, con lo cual se obtuvo que los resultados devueltos al realizar la consulta son correctos en algunos casos.

Palabras claves: procesamiento de lenguaje natural, inteligencia artificial, plataforma e-commerce, buscador, productos.

ABSTRACT

Since the beginnings of artificial intelligence until today there have been great technological advances, what was said to be science fiction is now reality. With the lapse of time the human being has become very ambitious so it has been acquiring various knowledge and enhancing them to create systems that help facilitate their daily lives, as is the case of robots that are created in order to do actions that humans perform, the development of smart phones, virtual assistants, among others which facilitate communication.

During several generations humanity has had to adapt to different disasters that threaten its existence, one of those inconveniences is the pandemic caused by the covid19 that forced the population to remain in quarantine, in this way the activities were reduced, one of the sectors that had a remarkable growth is the online commerce because users began to visit these sites to make their purchases.

E-commerce platforms focus more on design, forgetting an important point which are basic functions such as the product search engine, which is of great importance because users enter the online store, search for products, often obtaining ambiguous results or a lot of irrelevant information. Therefore, to fulfill the purpose of this project, we have proposed the development of a prototype of a search engine for e-commerce platforms using natural language processing.

Natural language processing (NLP) is very important within artificial intelligence and NLP is also transcendental within computational linguistics, this allows to convert natural language into a mechanism that computers can understand.

To conclude with the purpose of the project, a prototype architecture was developed, this will allow the processing of data phase by phase. Organizing the input data to apply natural language processing, and as a result obtaining an answer.

The coding was done by using different tools: Java, Netbeans, JSP, Web Services and the Apache Lucene library; in addition, different methods and

classes were developed, which combined with the methods of the Lucene library, perform a syntactic search.

The evaluation stage was performed using classification metrics: precision and completeness or Recall. In the first evaluation a product query was performed, where the results were of the maximum value 100%, whereby the precision and sensitivity score were close to 1, this indicates that the response values are relevant with perfect accuracy. In the second test, a query with several words was performed where the results obtained in terms of precision were 33%, sensitivity 50%, which indicates that the values in the search are repeated in several products, which shows that the precision is not exact and some answers are not relevant. In the third evaluation, an incomplete data was entered to know the level of proximity of the search engine, which showed that the results returned in the query are correct in some cases.

Keywords: natural language processing, artificial intelligence, e-commerce platform, search engine, products.

ÍNDICE DE CONTENIDO

DEDICATORIA	1
AGRADECIMIENTO	2
RESUMEN.....	3
ABSTRACT	5
INTRODUCCIÓN.....	11
1. CAPITULO I: DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS	12
1.1. Ámbito de aplicación: descripción del contexto y hechos de interés.....	12
1.2. Establecimiento de requerimientos.....	12
1.3. Justificación de requerimiento a satisfacer.....	13
2. CAPITULO II: DESARROLLO DEL PROTOTIPO.....	13
2.1. Definición del prototipo tecnológico	13
2.2. Fundamentación teórica del prototipo	14
2.2.1. Entorno de trabajo	14
2.2.1.1. Librería	14
2.2.1.1.1. Lucene-Core.....	14
2.2.1.2. Capacidad computacional.....	15
2.2.1.2.1. CPU	15
2.2.1.2.2. GPU	16
2.2.2. Procesamiento de lenguaje natural	16
2.2.2.1. Enfoques de procesamiento de lenguaje natural	17
2.2.2.2. Modelo del procesamiento de lenguaje natural	17
2.2.2.3. Niveles del procesamiento de lenguaje natural.....	18
2.2.3. Análisis sintáctico	19
2.2.3.1. Arquitectura de interpretación sintáctica.....	19
2.2.4. Lingüística computacional.....	20
2.2.5. Motor de búsqueda	20
2.2.6. Análisis semántico.....	21
2.2.6.1. Arquitectura de interpretación semántica	22
2.2.6.2. Buscador semántico.....	22
2.3. Metodología.....	23
2.4. Objetivos del prototipo	24
2.4.1. Objetivo general.....	24
2.4.2. Objetivos específicos	24
2.5. Diseño del prototipo.....	24

2.5.1.	Requisitos	24
2.5.2.	Herramientas de desarrollo.....	25
2.5.2.1.	Java	25
2.5.2.2.	JavaScript	25
2.5.2.3.	Web Services	25
2.5.2.4.	GlassFish Server	25
2.5.2.5.	Netbeans.....	25
2.5.2.6.	JavaServer Pages	26
2.5.3.	Arquitectura del prototipo	26
2.5.3.1.	Diseño de arquitectura del buscador a desarrollar	26
2.5.3.2.	Aplicación de la librería Apache Lucene.....	27
2.6.	Desarrollo de prototipo.....	30
2.6.1.	Métodos de búsqueda	30
2.6.2.	Clase producto.....	30
2.6.3.	Clase Base producto.....	31
2.6.4.	Interfaz del buscador.....	32
2.7.	Ejecución y/o ensamblaje del prototipo.....	34
3.	CAPITULO III: EVALUACIÓN DEL PROTOTIPO	35
3.1.	Plan de evaluación	35
3.1.1.	Precisión	35
3.1.2.	Exhaustividad, Recall.....	36
3.2.	Resultados de evaluación	36
3.2.1.	Resultados Prueba 1.....	36
3.2.2.	Resultados Prueba 2.....	37
3.2.3.	Resultados prueba 3.....	37
3.3.	Conclusiones	39
3.4.	Recomendaciones	40
4.	BIBLIOGRAFIA	41

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Mapa mental de la fundamentación teórica del prototipo.....	14
Ilustración 2: Arquitectura de Apache Lucene	15
Ilustración 3: Estructura CPU	16
Ilustración 4:Estructura GPU	16
Ilustración 5: Modelo de procesamiento de lenguaje natural	18
Ilustración 6: Arquitectura de interpretación sintáctica	19
Ilustración 7: motor de búsqueda	21
Ilustración 8: Arquitectura de interpretación semántica	22
Ilustración 9: Metodología	23
Ilustración 10:Arquitectura del prototipo a desarrollar	26
Ilustración 11:Estructura de indexación.....	27
Ilustración 12: Tokenización de texto	28
Ilustración 13: Métodos de búsqueda.....	30
Ilustración 14: clase producto.....	31
Ilustración 15: Clase base producto	32
Ilustración 16: código de la interfaz	32
Ilustración 17: codificación del buscador	33
Ilustración 18: código aproximación	33
Ilustración 19: ejecución del buscador	34
Ilustración 20: interfaz gráfica del buscador	34
Ilustración 21: búsqueda de productos.....	35
Ilustración 22: formula de precisión.....	35
Ilustración 23: formula de exhaustividad	36
Ilustración 24: prueba 3 ingreso consulta	37
Ilustración 25: prueba 3 resultados	38

ÍNDICE DE TABLAS

Tabla 1: Niveles de procesamiento de lenguaje natural.....	19
Tabla 2: Requisitos de hardware.....	24
Tabla 3: requisitos software.....	24
Tabla 4: Valores de los atributos del token	28
Tabla 5: Atributos de la tokenización.....	29
Tabla 6: resultados prueba 1	36
Tabla 7: resultados prueba 2.....	37

INTRODUCCIÓN

Hoy en día con el uso de la tecnología y la internet de la mano realizar una compra en línea en plataformas e-commerce es muy sencillo, por ello muchas personas prefieren utilizar plataformas para hacer sus compras. Al momento de realizar una búsqueda se ha encontrado un déficit, lo que ha ocasionado investigar maneras más eficientes y precisas de buscar información.

El procesamiento de lenguaje natural es la base para la comunicación entre humano y ordenadores, su principal tarea es la lingüística computacional según [1], apareció por los años 1950 en EEUU, surgió por la necesidad de traducir de manera automática documentos en otros idiomas, aunque sus traducción no eran exactas se tenía cierto nivel de perfección, en los años noventa tuvo sus puntos altos que con ayuda se expandió trayendo grandes avances como lo es en el análisis morfológico y sintáctico, así como también técnicas de reconocimiento de voz, texto, traducción automática, análisis de sentimientos.

Lograr que un buscador muestre datos relevantes de la consulta, además entienda lo que un usuario está buscando y de un resultado eficiente es muy ansiado para los consumidores de hoy en día. Razón por la cual este proyecto se enfoca en desarrollar un buscador que aplique procesamiento de lenguaje natural, cabe mencionar que el desarrollo del mismo se limitará solo a la búsqueda de productos para plataformas e-commerce.

En el presente documento se expondrán conceptos claves sobre NLP, lingüística computacional, machine learning, buscador semántico. Además, consta de los siguientes capítulos:

Capítulo 1: en este capítulo se describen los hechos de intereses, se establecen los requerimientos y su justificación.

Capítulo 2: se establece la fundamentación teórica la cual sirve de ayuda a la investigación para la creación del prototipo, definición de objetivos, diseño del prototipo donde encontramos la arquitectura del buscador a desarrollar, el desarrollo que esta compuesto por el algoritmo, clases y por último ejecución del prototipo.

Capítulo 3: tenemos la evaluación del resultado del prototipo, se especifican las conclusiones y recomendaciones de los objetivos propuestos.

1. CAPITULO I: DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1. Ámbito de aplicación: descripción del contexto y hechos de interés

En Ecuador el comercio electrónico ha ido aumentando gracias a que hoy en día las compras en línea ya no son ninguna novedad y los usuarios confían cada vez más en estas plataformas. Según [2], menciona que el método de compra y venta de productos en la actualidad tiene grandes cambios de generaciones, pasando de modos habituales de visitar tiendas de compras, supermercados a emplear alternativas tecnológicas que se adapten al mercado y esa principal alternativa es el comercio electrónico, haciendo uso de la expansión de la internet.

Cuando un usuario ingresa a una plataforma e-commerce siempre espera encontrar de una manera rápida, este motor de búsqueda facilitara obtener una respuesta exacta de lo que el usuario necesita, optimizando las consultas.

El procesamiento de lenguaje natural es un ambiente en donde intervienen dos conceptos; la informática y la lingüística que, con ayuda de distintas técnicas entre una de ellas que son las redes de transición permiten hacer uso de ella.

Las redes de transición son parte del procesamiento de lenguaje natural para el análisis de la estructura sintáctica de una oración o palabra, estas redes están compuestas por nodos y arcos, donde los nodos muestran el análisis del texto, los arcos las restricción del texto, estos arcos se señalan con símbolos de puntuación [3].

El alcance de este proyecto es desarrollar un buscador de productos utilizando el procesamiento de lenguaje natural, mediante el uso de herramientas tecnológicas y técnicas que permiten utilizar NLP.

1.2. Establecimiento de requerimientos

Para el desarrollo del buscador se hizo uso de librerías en java que permiten aplicar el procesamiento de lenguaje natural. Una de las principales librerías utilizadas es Lucene-Core que permite realizar búsqueda de texto, una de sus importantes funciones es la indexación y normalización.

Las fases que se emplearon para el desarrollo de este proyecto se puntúan a continuación:

- En la primera fase se realizó una recolección de información, para obtener un conocimiento sobre el desarrollo de un motor de búsqueda de productos aplicando el procesamiento de lenguaje natural.
- Se estableció un conjunto de datos de los productos en lenguaje natural para que, por medio de los métodos, clases, algoritmo permita realizar una búsqueda sintáctica mediante la librería y posterior dar una respuesta
- Por último, en la fase de pruebas se empleó métricas de clasificación que ayudan a evaluar la precisión y sensibilidad del buscador, además se ingresaron consultas con errores para comprobar el análisis semántico.

1.3. Justificación de requerimiento a satisfacer

El procesamiento de lenguaje natural en el comercio electrónico ha crecido mucho por la constante demanda de comprar productos en línea, la opción de búsqueda es uno de los usos más evidentes en la actualidad, los usuarios han encontrado un gran avance en estas plataformas, a como era anteriormente que era muy difícil de encontrar lo que el usuario necesita por la gran cantidad de resultados que obtenía, el beneficio y el impacto que proporciona NLP es muy notorio.

La propuesta del proyecto a desarrollar es el de un buscador de productos de plataformas e-commerce utilizando procesamiento de lenguaje natural que permitirá al usuario tener una mejor experiencia al realizar una búsqueda, no solo en el tiempo de respuesta sino también al momento de escribir en el buscador.

2. CAPITULO II: DESARROLLO DEL PROTOTIPO

2.1. Definición del prototipo tecnológico

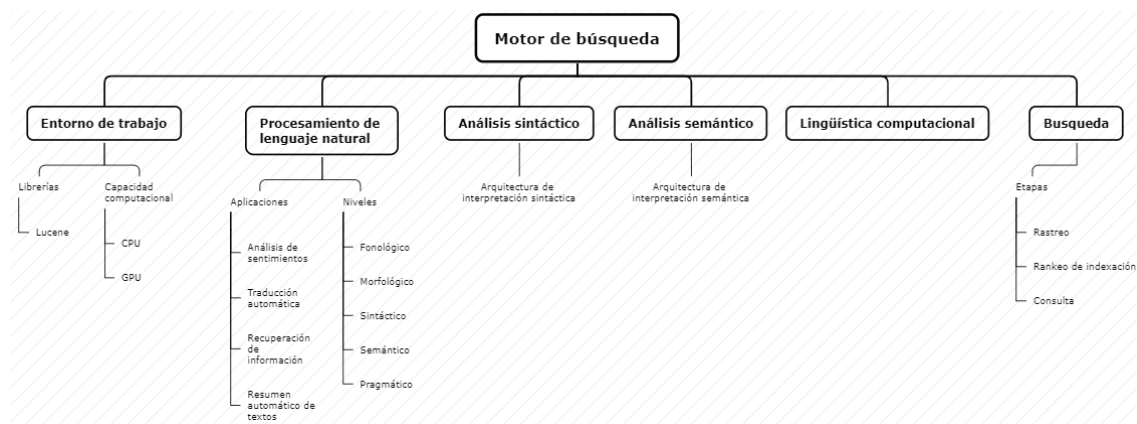
Procesamiento de lenguaje natural o NLP es el cruce entre dos aspectos de la computación, una la inteligencia artificial y segundo la lingüística computacional, con el avanzar de la tecnología NLP crece cada vez, a efecto del interés de establecer una conexión entre usuario y ordenadores, gracias a este efecto se

ha desarrollado sistemas como el asistente de voz Alexa, el buscador de Google, entre otros que ayudan a satisfacer nuestras necesidades.

Para poder procesar el lenguaje natural a través de reconocimiento de texto, voz, se establecen reglas de reconocimiento de patrones y con información almacenada en diccionarios computacionales se concretan los patrones que se deben identificar en una oración o palabra.

2.2. Fundamentación teórica del prototipo

Ilustración 1: Mapa mental de la fundamentación teórica del prototipo



Fuente: elaboración propia

2.2.1. Entorno de trabajo

2.2.1.1. Librería

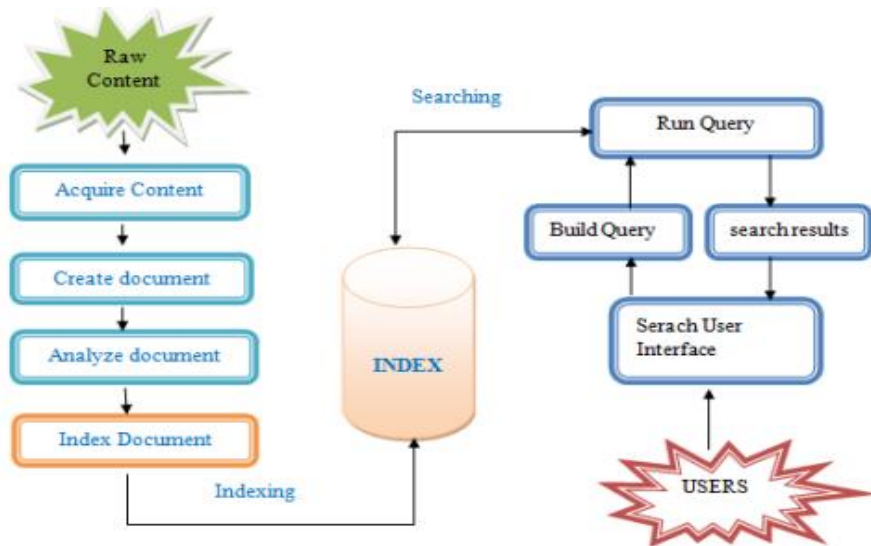
2.2.1.1.1. Lucene-Core

Es una librería implementada para Java según [4], indica que Lucene utiliza una estructura de índice invertido esta permite realizar búsquedas de texto rápidamente. Un índice invertido es un diccionario de las palabras únicas de un documento para crear un índice se divide el campo de contenido en palabras separadas para después crear un diccionario con todas las palabras únicas y por último se enumeran en donde aparece cada término.

Apache Lucene presenta funciones como gestión de procesos, selección de archivos y el análisis, además brinda una búsqueda más efectiva. El proceso por el cual Lucene realiza la indexación del texto, es por medio de un disco del

sistema en donde escribe los índices y este índice se une en la memoria mediante algoritmos específicos [5].

Ilustración 2: Arquitectura de Apache Lucene



Fuente: tomado de [5]

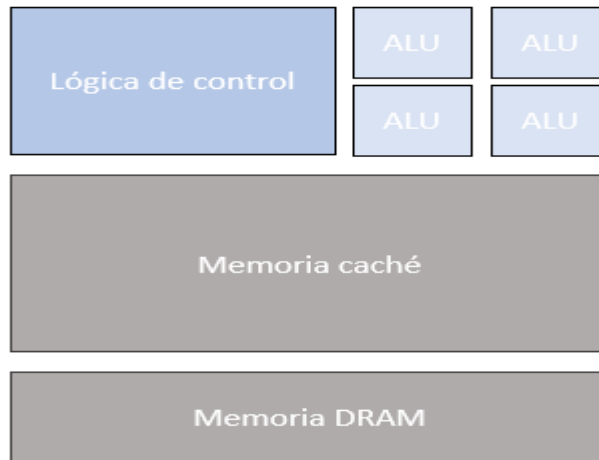
2.2.1.2. Capacidad computacional

Hoy en día con el avance tecnológico y utilizar a toda plenitud un sistema requiere de un gran consumo del hardware, por ello la CPU tanto como la GPU son importantes en un computador.

2.2.1.2.1. CPU

Unidad de procesamiento central o también conocido como CPU es la parte más importante dentro del computador, es el cerebro de toda la operación en donde se realizan todos los cálculos y sus funciones [6].

Ilustración 3: Estructura CPU

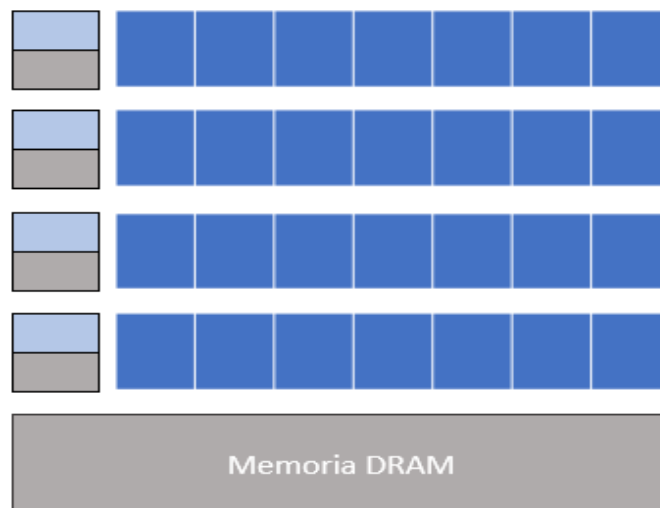


Fuente: elaboración propia

2.2.1.2.2. GPU

Las GPUs son dispositivos conformados por una gran cantidad de unidades de proceso, originalmente la GPU fue creada con la finalidad de ayudar a la CPU en los gráficos, pero si se los programa adecuadamente estas se pueden manejar como procesadores de propósito general [7].

Ilustración 4: Estructura GPU



Fuente: elaboración propia

2.2.2. Procesamiento de lenguaje natural

Procesamiento de lenguaje natural (NLP) es una ciencia de la inteligencia artificial que se basa en la lingüística, en donde su principal función es procesar

un texto en lenguaje humano y que estos sean procesados por una función o algoritmo obteniendo un resultado en base a los datos de entrada [8].

Además [9], menciona que NLP posee varias aplicaciones como:

- **Análisis de sentimientos:**

Es la capacidad q tiene un sistema para analizar una conversación, comentarios y determinar si esa información es buena o mala.

- **Recuperación de información:**

Es la búsqueda de información mediante la extracción de palabras claves, como ocurre en los buscadores.

- **Resumen automático de textos:**

Cuando una persona obtiene varios resultados de una consulta, el resumen automático es una selección de información relevante de la búsqueda realizada.

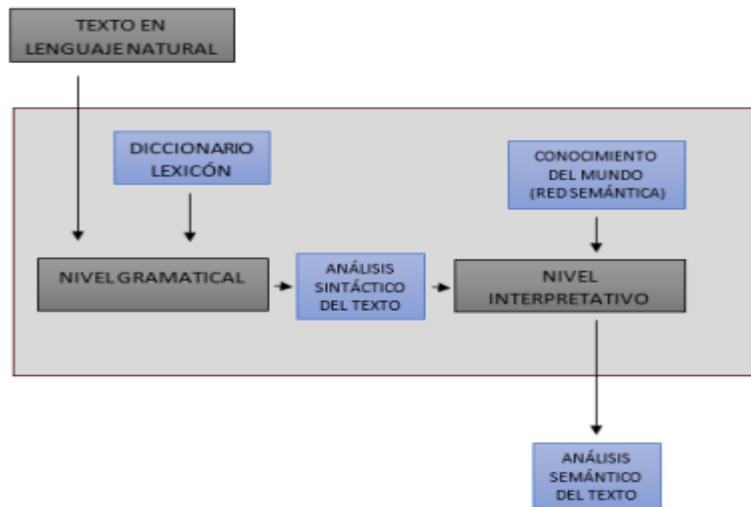
2.2.2.1. Enfoques de procesamiento de lenguaje natural

[10], Menciona que NLP tiene dos enfoques el primero enfoque simbólico y segundo enfoque estadístico, el primero se basa en la creación de sistemas que ayudan a almacenar hechos lingüísticos, como representación tenemos en el caso de un buscador. El segundo enfoque se describe en la creación de sistemas que no ayudan a almacenar hechos lingüísticos, utilizando técnicas de procesamiento de información como por ejemplo máquinas de estado, modelos probabilísticos.

2.2.2.2. Modelo del procesamiento de lenguaje natural

La representación de este modelo comienza con el ingreso de datos en lenguaje natural, su proceso contiene dos módulos que son: módulo gramatical, donde se encuentra el análisis sintáctico y el segundo módulo que es el interpretativo en donde está el análisis semántico [10].

Ilustración 5: Modelo de procesamiento de lenguaje natural



Fuente: tomado de [10].

2.2.2.3. Niveles del procesamiento de lenguaje natural

De acuerdo con [10] y [11] mencionan que el procesamiento de lenguaje natural tiene los siguientes niveles:

- **Nivel fonológico**
Este nivel se basa en el sonido del lenguaje humano es decir la resonancia que contienen las palabras.
- **Nivel morfológico**
Estudia cómo están construidas las palabras examinando su significado y sus contribuyentes, además se pueden clasificar en verbos, sustantivos, adjetivos, etc.
- **Nivel sintáctico o sintaxis**
Se basa en el orden de las palabras dentro para la construcción de una oración además de las conexiones para esa oración, una de las técnicas utilizadas para este fin es las redes de transición.
- **Nivel semántico**
Se enfoca en todo lo que concierne con el significado de cada palabra dentro de la oración, aquí se utilizan técnicas de desambiguación semántica la misma que nos ayuda a entender palabras que tienen varias interpretaciones.
- **Nivel pragmático**

En el último nivel se menciona cómo se utiliza y qué enfoque tiene cada oración en distintas situaciones, aquí se emplean herramientas que ayuden a la interpretación de las intenciones y función de un texto.

Tabla 1: Niveles de procesamiento de lenguaje natural

Niveles de PLN	Ámbito de actuación	Herramientas
fonológico	Sonidos	Corpus de aprendizaje Modelos acústicos
morfológico	Formas	Etiquetado POS Léxico computacional
sintáctico o sintaxis	Estructuras	Bases de datos sintácticas Arboles de decisión
semántico	Significados	Bases de datos semánticas
pragmático	Comunicación	Bases de datos semánticas Ontologías

Fuente: elaboración propia

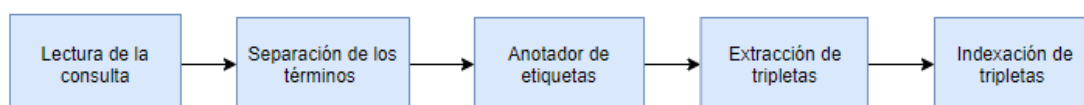
2.2.3. Análisis sintáctico

Este término se encuentra dentro de la lingüística computacional, se encarga de estudiar la composición de la oración, crear arboles de relación basándose en las palabras para posteriormente realizar tokenización y establecer un orden de jerarquía [12].

Según [13], menciona que existen diversas técnicas en el análisis sintáctico, hay buscadores que emplean técnicas de análisis superficiales como el etiquetado POS. Y otros que utilizan técnicas más robustas como análisis de dependencias.

2.2.3.1. Arquitectura de interpretación sintáctica

Ilustración 6: Arquitectura de interpretación sintáctica



Fuente: elaboración propia.

Como primer paso se tiene el ingreso de una consulta, luego realiza la separación de los términos. Estos términos se separan en tripletas, que son etiquetados para ser ingresados a un índice. Posterior se extrae esas tripletas del índice para por último ser indexadas en orden de prioridad.

2.2.4. Lingüística computacional

La lingüística computacional (LC) está compuesta por dos bases, la primera lingüística teórica y la segunda ciencia cognitiva, la LC trata de ajustar la lengua natural a base de la vista computacional, esto quiere decir que resuelve problemas cuando oraciones tiene distintas interpretaciones, además permite clasificar, identificar y extraer palabras [14].

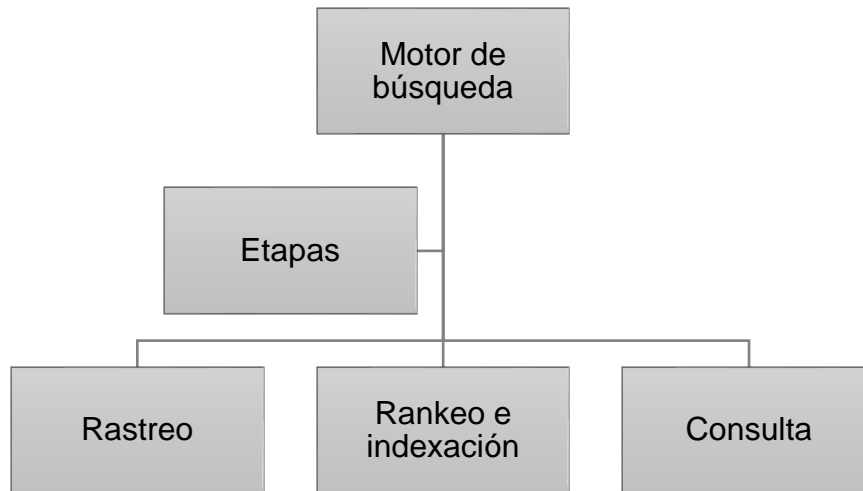
Según [15], considera que en la interacción de la informática con la lingüística existen tres niveles los cuales son:

- Como en primer nivel tenemos que las computadoras son elementales en la vida de las personas ya que se las utiliza muy a menudo para agilizar nuestras tareas en el trabajo lingüístico, que anteriormente se las realizaba manualmente o el empleo de máquinas menos eficientes.
- En el segundo nivel los computadores ayudan a manejar grandes cantidades de información y así evitar pérdidas de tiempo de los lingüistas en el almacenamiento de información de textos, hoy se cuenta con corpus electrónicos en donde se tiene acceso a la información con rapidez.
- En el tercer nivel y como lo señala [15], el más interesante es la comunicación entre usuario y ordenador por medio del lenguaje natural como intervención para esta comunicación.

2.2.5. Motor de búsqueda

Son muy usados en la red para realizar consultas de un tema en específico, un motor de búsqueda está organizado por programas que ayudan a automatizar el rastreo e indexación de información que se busca en las páginas web [16].

Ilustración 7: motor de búsqueda



Fuente: elaboración propia

Para ampliar los conceptos de las etapas de un motor de búsqueda [16], menciona lo siguiente:

- **Rastreo**
Es un sistema llamado robot que realiza el rastreo por medio de la recolección de los datos obtenidos en la web a través de los enlaces.
- **Ranqueo e indexación**
Un algoritmo almacena la información recopilada en el rastreo, para posterior ranquear esa información y por último indexar la información en bases de datos.
- **Consulta**
El programa realiza una búsqueda en la base de datos mostrando los enlaces y ordenados según el ranqueo dado.

2.2.6. Análisis semántico

La semántica da un importante sentido al contenido de los sitios web, al punto de que softwares procesen tareas complicadas, para leer los sitios y obtener datos más eficientes, para ayudar en las tareas de indexar y búsqueda [17].

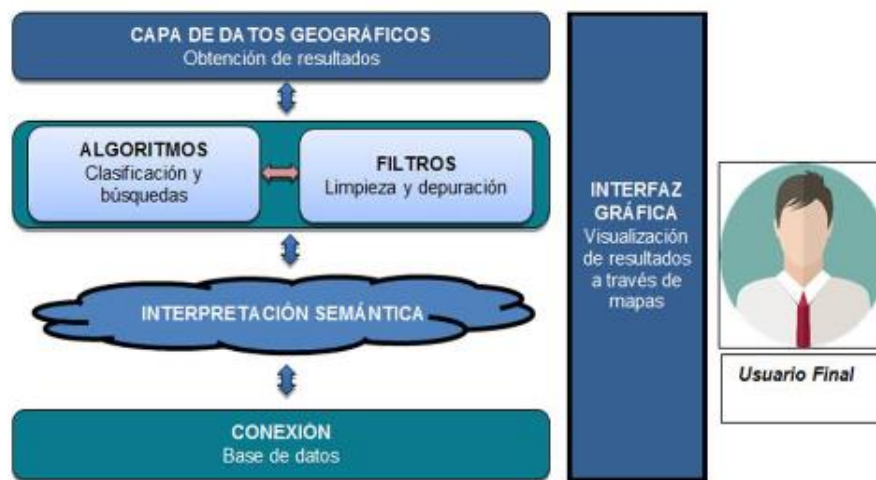
Por medio de la web semántica han aparecido nuevas tecnologías como lo son XML, XML Schema, RDF/RDFS, OWL, etc. La semántica ofrece a la web

almacenar información y estos a su vez le dan significado bien definidos a esos datos para que el software los pueda utilizar [18].

2.2.6.1. Arquitectura de interpretación semántica

Esta arquitectura se fundamentó en el modelo vista controlador, en vista encontramos los resultados de la búsqueda, mientras que en el controlador realiza todos los procesos de búsqueda y por último tenemos como modelo esta la administración de BD [19].

Ilustración 8: Arquitectura de interpretación semántica



Fuente: tomada de [19].

En la ilustración se puede apreciar que el usuario realiza la interacción con el visualizador geográfico por medio de su interfaz. En primera instancia el usuario ingresa la consulta, esta busca en la Base de datos, luego realiza la interpretación semántica la mismo que aplica algoritmos de clasificación, búsqueda para posterior realizar filtros de limpieza y depuración de esos datos, para mostrar al usuario final los resultados obtenidos.

2.2.6.2. Buscador semántico

El término semántico no es desconocido en los buscadores, se entiende por esta palabra cuando una consulta ingresada tiene errores gramaticales y como respuesta da resultados positivos, en otras palabras el algoritmo de búsqueda trata de adivinar lo que el usuario ingresa al buscador y entregar como resultado lo que considera que el usuario busca [20].

También [21], indica que un buscador semántico en la actualidad cara a cara con un buscador convencional el aumento del uso de estos buscadores ha sido muy alto por la alta demanda de recuperación de información y porque ayuda en el entendimiento del significado de las palabras.

2.3. Metodología

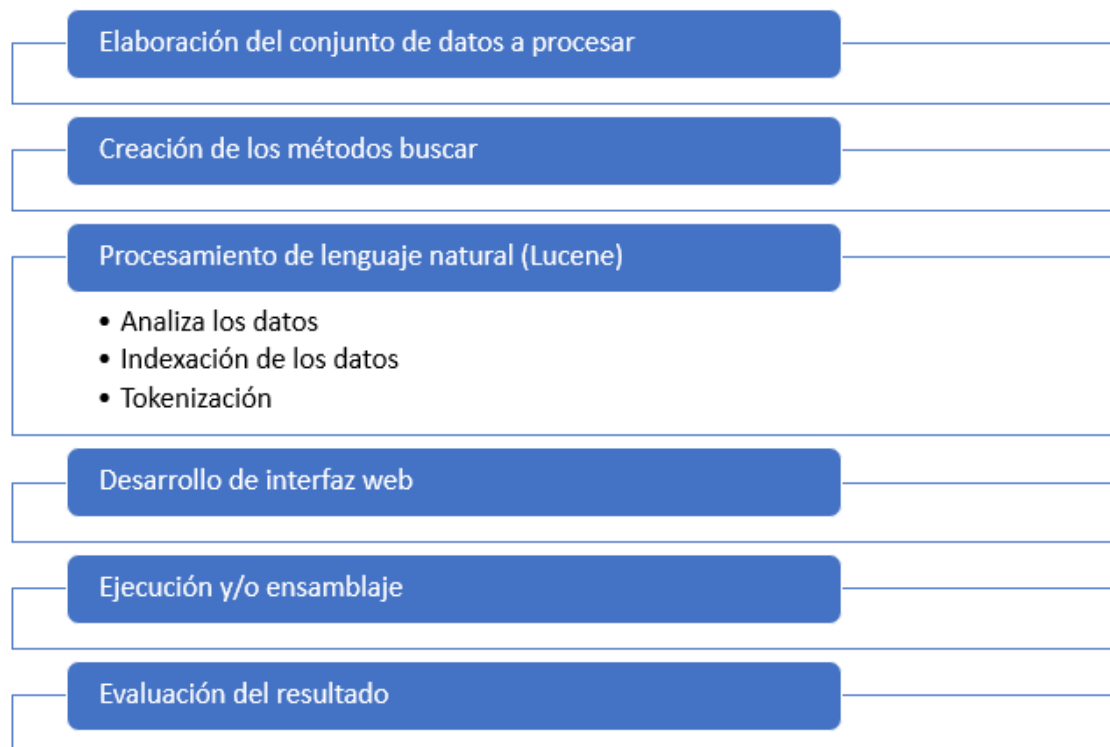
Después de un análisis exploratorio de la información recopilada se identificó varios niveles de procesamiento de lenguaje natural, que por medio del uso de librerías se pueden aplicar a un conjunto de datos.

Como primer punto es crear los métodos de búsqueda utilizando parámetros de la librería Lucene. Luego se creó la clase de producto donde se encuentra los parámetros de descripción de cada uno. La clase base producto donde se registran los n productos y por último la creación de la interfaz.

Posterior se realizó las pruebas ejecución y/o ensamblaje del prototipo.

Una vez concluido se continuo con la evaluación del del buscador.

Ilustración 9: Metodología



Fuente: elaboración propia

2.4. Objetivos del prototipo

2.4.1. Objetivo general

Desarrollar un motor de búsqueda de productos para plataformas e-commerce mediante la utilización de procesamiento de lenguaje natural.

2.4.2. Objetivos específicos

- Consultar en artículos, revistas científicas temas afines al procesamiento de lenguaje natural.
- Analizar los niveles semántico y sintáctico de NLP.
- Diseñar una arquitectura para el desarrollo del buscador.
- Realizar pruebas del buscador para conocer su efectividad con el uso de procesamiento de lenguaje natural.

2.5. Diseño del prototipo

2.5.1. Requisitos

Los requisitos a utilizar son:

Tabla 2: Requisitos de hardware

HARDWARE	
Marca	Lenovo
Procesador	Intel(R) Core (TM) i7-6500U CPU @ 2.50GHz 2.60 GHz
Memoria	8 GB
Disco	1 TB
Sistema Operativo	Windows 10

Fuente: elaboración propia

Tabla 3: requisitos software

SOFTWARE
Java
Lucene Core

JavaScript
GlassFish

Fuente: elaboración propia

2.5.2. Herramientas de desarrollo

2.5.2.1. Java

Java utilizado por los programadores, esta herramienta se basa en la programación orientada a objetos, además es muy fácil de utilizar, una de sus ventajas es que se encuentra muy documentado por lo que no es difícil de entender este lenguaje [22]. Los programas creados en Java son portables.

2.5.2.2. JavaScript

JavaScript es una tecnología que permite realizar acciones complejas dentro de una página web, por medio de este lenguaje se puede implementar acciones interactivas como por ejemplo el de completar una palabra [23].

2.5.2.3. Web Services

Servicios web es una tecnología que implementa protocolos, estándares para el canje de información en las aplicaciones [24]. Este servicio permite la comunicación de aplicaciones desarrolladas en distintos lenguajes de programación, a través del lenguaje de marcado extensible (XML) [25].

2.5.2.4. GlassFish Server

Es un servidor gratuito muy utilizado por los programadores debido a que permite crear aplicaciones en Java escalables y portables. GlassFish es una comunidad donde todos los usuarios pueden hacer uso del servicio, y agregar funciones importantes [26].

2.5.2.5. Netbeans

Netbeans es un entorno de desarrollo, principalmente para Java, es un software gratuito y libre. Es utilizado para el desarrollo de aplicaciones de escritorio, móviles, web que emplean plataformas como Java, HTML5, etc. [27].

2.5.2.6. JavaServer Pages

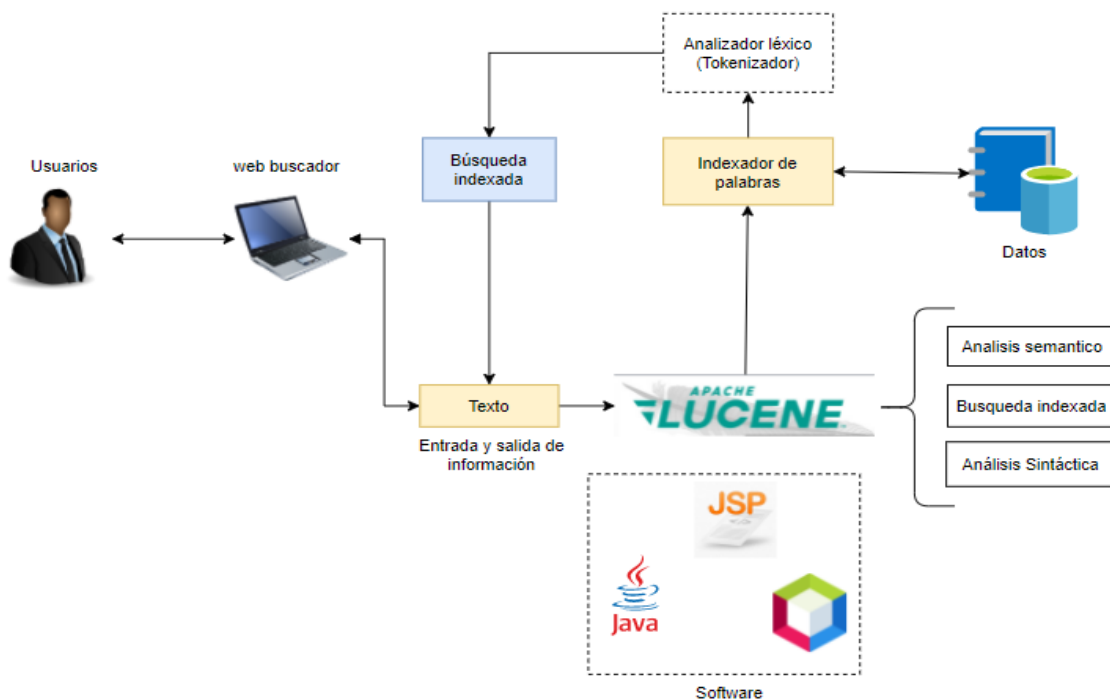
Es una tecnología originalmente de java que es utilizado por los programadores para la creación de páginas web JavaServer pages permite que los programas creados se puedan ejecutar en servidores web [28].

2.5.3. Arquitectura del prototipo

2.5.3.1. Diseño de arquitectura del buscador a desarrollar

Se crea la arquitectura para el desarrollo del prototipo

Ilustración 10:Arquitectura del prototipo a desarrollar



Fuente: elaboración propia

En la arquitectura de la ilustración 10 se muestra el ingreso de un dato por parte del usuario, el mismo que es procesado por los componentes de la librería Apache Lucene, de esta manera la palabra ingresada es indexada y tokenizada lo que facilita la búsqueda.

Se utiliza el lenguaje programación java para realizar la programación de los métodos pertinentes de la librería Lucene, además de la creación de la interfaz gráfica de usuario, entre los componentes que utiliza la arquitectura esta la siguiente:

Usuario: es la persona interactuar con el buscador.

Comunicación: se establece la conexión con el sitio web mediante una computadora, de esta manera se accede al buscador.

Software: se encuentra java el cual es un lenguaje de programación de alto nivel utilizado para desarrollar los métodos del buscador mediante el entorno de programación Netbeans, y JavaScript para crear el autocompletado de texto.

Librería Apache Lucene: Permite la tokenización para el análisis sintáctico y el significado del texto para el análisis semántico.

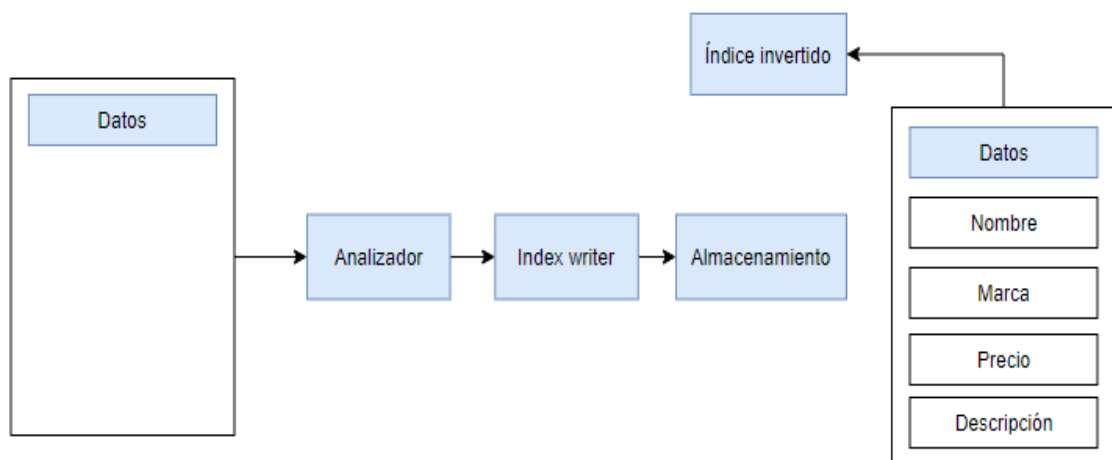
Respuesta: Se muestran los resultados aproximados dependiendo de la búsqueda del usuario.

2.5.3.2. Aplicación de la librería Apache Lucene

El buscador tiene como base a la librería Apache Lucene tal como muestra la Ilustración 10: El proceso inicia con el ingreso de una palabra a buscar por parte del usuario. Lucene aplica el proceso de indexación el cual consiste en extraer y analizar lo más relevante, el índice es una estructura de datos que facilita la búsqueda de información, algo similar a el índice de un libro.

Durante la búsqueda se consulta el índice, con lo cual retornan los datos solicitados. El índice es una tabla donde se registra la posición de cada término.

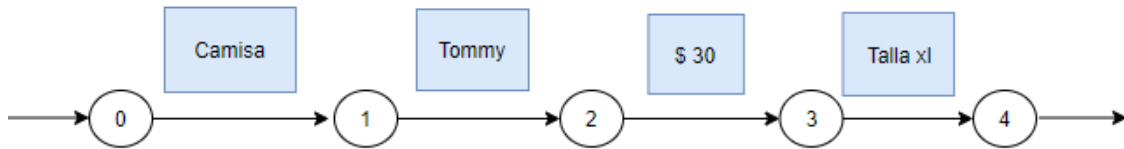
Ilustración 11: Estructura de indexación



Fuente: elaboración propia

El analizador viene dado por la tokenización la cual hace referencia al dividir en fragmentos las palabras de una oración. Ejemplo: **Camisa tommy 30\$ talla:xl**

Ilustración 12: Tokenización de texto



Fuente: elaboración propia

En este ejemplo se asigna un valor a cada palabra de la oración(nodo), esto indica la secuencia de la semántica de la oración inicial.

Los nodos de la ilustración 12 representan posiciones, donde cada token posee los atributos: posición inicial y final, los cuales son suficientes para describir la posición del token.

Tabla 4: Valores de los atributos del token

N°	Token	P. inicial	P. Final
1	Camisa	0	1
2	Tommy	1	2
3	\$30	2	3
4	Tall xl	3	4

Fuente: elaboración propia

Lucene utiliza dos atributos a parte de los mencionados anteriormente estos son: posición incremental y tamaño:

Posición incremental: muestra el aumento de la posición del token tomando como referencia el token anterior se utiliza la siguiente formula.

$$PINC = PI - PIA$$

PI = Posición inicial.

PIA = Posición inicial actual.

PINC = posición incremental

Tamaño de la posición: Muestra el rango de la posición actual del token. Usa la siguiente formula:

$$TP = FP - IPA$$

TP = Tamaño de la posición.

FP = Posición final.

IPA = posición inicial actual

Tabla 5: Atributos de la tokenización

N°	Token	P. inicial	P. Final	P. incre	P. tam
1	Camisa	0	1	1	1
2	Tommy	1	2	1	1
3	\$30	2	3	1	1
4	Tall xl	3	4	1	1

Fuente: elaboración propia

El proceso de tokenización es uno de los elementos de más relevancia dentro del procesamiento de lenguaje natural, debido que permite dividir texto en oraciones y estas a su vez en palabras, permitiendo reconocer una variedad de términos al realizar la búsqueda.

Otro punto importante en el analizador es la normalización de texto, con lo cual las letras en mayúsculas las transforma en minúsculas mediante la clase StandardAnalyzer.

Para la interpretación del texto apache Lucene: proporciona un analizador de consulta para búsqueda difusas mediante el algoritmo de distancia Levenshtein. Ejemplo: Para realizar una consulta se interpone una virgulilla (~) después del texto: **panlon~**. Este algoritmo para encontrar la coincidencia de las palabras, internamente realiza un cálculo mínimo de operaciones de inserción, ediciones como, por ejemplo, la distancia entre “panlon” y “pantalón”:

1. Panlon → Pantlon (inserción 1: agrega la **t**)
2. Pantlon → pantalón (inserción 2: agrega la **a**)

2.6. Desarrollo de prototipo

Para el desarrollo del prototipo se realiza lo siguiente:

2.6.1. Métodos de búsqueda

En el siguiente código se muestra el método “MotorBusqueda”, el cual se encarga de realizar la tokenización, al dividir la cadena de texto en una lista lo cual facilita las consultas. Seguido se presenta el método “funcionBuscar”, este devuelve el resultado de la búsqueda por parte del usuario. Dentro de este método se observa una variable tipo Query, propia de la librería Lucene la cual se encarga de realizar la consulta, a su vez esta variable se ejecuta con la función interna search, la misma que se almacena en la variable tipo hits la cual retorna al usuario.

Ilustración 13: Métodos de búsqueda

```
package buscador.buscar;

import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.queryParser.ParseException;
import org.apache.lucene.queryParser.QueryParser;
import org.apache.lucene.search.Hits;
import org.apache.lucene.search.IndexSearcher;
import org.apache.lucene.search.Query;

public class MotorBusqueda {
    private IndexSearcher buscador = null;
    private QueryParser analizador = null;

    public MotorBusqueda() throws IOException {
        try {
            buscador = new IndexSearcher("index-directory");
            analizador = new QueryParser("Contenido", new StandardAnalyzer()); //Toquenza palabras
        } catch (IOException ex) {
            Logger.getLogger(MotorBusqueda.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public Hits funcionBuscar(String textoConsulta)
    throws IOException, ParseException {
        Query query = analizador.parse(textoConsulta);
        Hits hits = buscador.search(query);
        return hits;
    }
}
```

Fuente: elaboración propia

2.6.2. Clase producto

Se crea una clase Producto, la misma que posee parámetros necesarios para describir productos, los mismo que se utilizan para realizar las búsquedas.

Ilustración 14: clase producto

```
1 2
public class Producto implements Serializable {
    private String id;
    private String nombre;
    private String marca;
    private String descripcion;
    private String precio;
    private String imagen;

    public Producto() {
    }

    public Producto(String id, String nombre, String marca, String descripcion, String precio, String imagen) {
        this.id = id;
        this.nombre = nombre;
        this.marca = marca;
        this.descripcion = descripcion;
        this.precio = precio;
        this.imagen = imagen;
    }

    public String getPrecio() {
        return precio;
    }

    public void setPrecio(String precio) {
        this.precio = precio;
    }

    public String getImagen() {
        return imagen;
    }
}
```

Fuente: elaboración propia

2.6.3. Clase Base producto

En esta clase se crea un array donde se instancia la clase producto, con lo cual se registran n productos, los cuales son utilizados para realizar las búsquedas.

Ilustración 15: Clase base producto

```
package buscador.reglas;

public class BaseProducto {

    private static final Producto[] Productos = {
        new Producto("1", "Reloj", "Tommy", "15cm", "150", "https://static2.abc.es/media/sumum/2019/02/27/apertura-relojes-"),
        new Producto("2", "Pantalon", "HUNGARO", "Talla 30", "52", "https://media.vogue.es/photos/5cc7367ffdc82261481fda65/m"),
        new Producto("3", "Pantalon", "Extreme", "Talla xl.", "85", "https://totboelsalvador.vteximg.com.br/arquivos/ids/174"),
        new Producto("4", "Camiseta", "Charvet", "Talla xl.", "45", "https://unitystores.vteximg.com.br/arquivos/ids/159546-"),
        new Producto("5", "Camiseta", "Tommy", "XL", "50", "https://http2.mistatic.com/D NQ NP 678303-MEC44030459387 112020-"),
        new Producto("6", "Camiseta", "Tommy", "s", "85", "https://unitystores.vteximg.com.br/arquivos/ids/159550-650-650/pe"),
        new Producto("7", "Reloj", "Tommy", "15cm", "150", "https://media.revistagq.com/photos/5f1039e496c6999a6368606c/16:9"),
        new Producto("8", "Telefono", "Samsung", "S03", "52", "https://www.tventas.com/8321049-large-default/telefono-celula"),
        new Producto("9", "Computadora", "Php", "CORE i7 8th gen", "85", "https://img.global.news.samsung.com/mx/wp-content/"),
        new Producto("10", "Mochila", "Vans", "Talla xl.", "45", "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRq96"),
        new Producto("11", "Reloj", "Fossil", "Analogico", "300", "https://media.revistagq.com/photos/6017eae096421468deb60a"),
        new Producto("12", "Camiseta", "Prada", "s", "85", "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSj2leUFYR"),
        new Producto("13", "Telefono", "iPhone", "13", "1200", "https://store.storeimages.cdn-apple.com/4982/as-images.apple")
    };

    public static Producto[] getProductos() {
        return Productos;
    }

    public static Producto getProductos(String id) {
        for (Producto producto : Productos) {
            if (id.equals(producto.getId())) {
                return producto;
            }
        }
        return null;
    }
}
```

Fuente: elaboración propia

2.6.4. Interfaz del buscador

En este caso se ha utilizado HTML y Bootstrap para el diseño de la interfaz del buscador.

Ilustración 16: código de la interfaz

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
    <script src="script.js"></script>
    <title >Buscador</title>
  </head>
  <body>
    <div style="border: 0px solid black;background: #146aff;color:#fff;padding: 3px">
      <h3 class="text-center" class="span12" >Buscar productos</h3>
    </div> <br>
    <div class="container-fluid">
      <form action="#" method="GET" class="form-control">
        <div class="row">
          <div class="col-md-10">
            <input type="text" id="txtbuscar" name="txtbuscar" class="form-control" placeholder="Buscar" onclick="">
          </div>
          <div class="col-md-2">
            <input type="submit" id="" name="" class="form-control btn-success" value="Buscar">
          </div>
        </div>
      </form>
      <div id="mensaje"></div>
    </div>
```

Fuente: elaboración propia

Al presionar el botón, se enviará el valor a buscar, de esta forma se ejecutará el método que se encarga de realizar la búsqueda sintáctica.

Ilustración 17: codificación del buscador

```
<% Indexador indexer = new Indexador();
try {
    indexer.reconstruyeIndexado();
} catch (IOException ex) {
    System.out.println(ex);
}
MotorBusqueda instancia = null;
instancia = new MotorBusqueda();
Hits hits = null;
String datos = request.getParameter("txtbuscar");
if (datos == null) {
    datos = "productos";
}
try {
    hits = instancia.funcionBuscar(datos);
} catch (ParseException ex) {
    System.out.println(ex);
}
System.out.print(hits.length());
Iterator<Hit> iter = hits.iterator();
%>
```

Fuente: elaboración propia

Si existe el producto buscado o un aproximado retornan estos valores, los mismos que fueron creados en la clase producto tal como se aprecia a continuación.

Ilustración 18: código aproximación

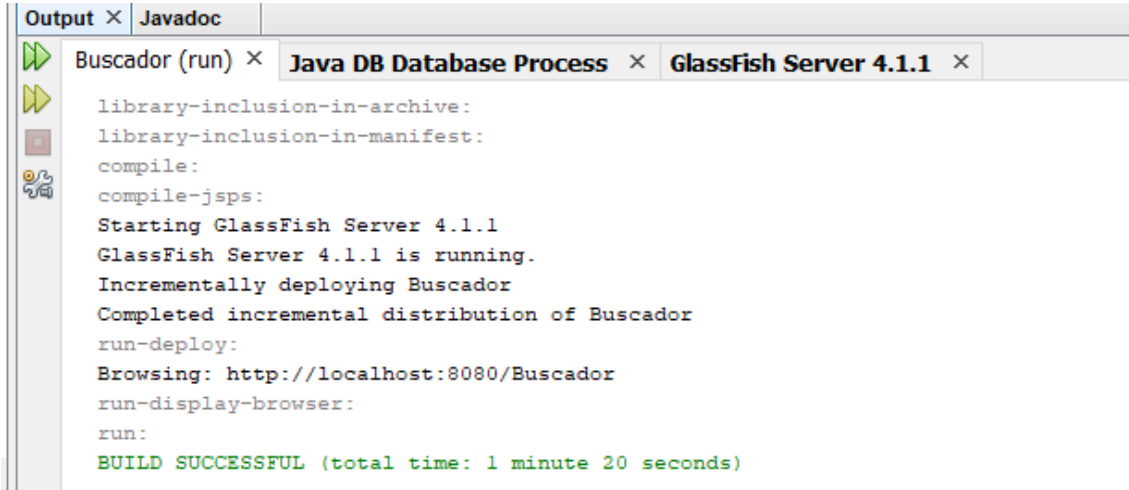
```
<div class="card">
  <div class="card-body">
    <div class="row">
      <div class="col-md-6">
        " width="230" >
      </div>
      <div class="col-md-6">
        <h4>Producto:<%=doc.get("Nombre") %></h4>
        <p>Marca:<%=doc.get("Marca") %></p>
        <p>Talla:<%=doc.get("Descripcion") %></p>
        <h4>Precio $ <%=doc.get("precio") %></h4>
        <% if (doc.get("imagen") != null) { %>
        <p><%=hit.getScore() %></p>
      </div>
    </div>
  <% } %>
</div>
<%
```

Fuente: elaboración propia

2.7. Ejecución y/o ensamblaje del prototipo

Para poner en marcha el buscador se ejecuta el servidor glassfish, el cual lanza la aplicación en el puerto 8080.

Ilustración 19: ejecución del buscador

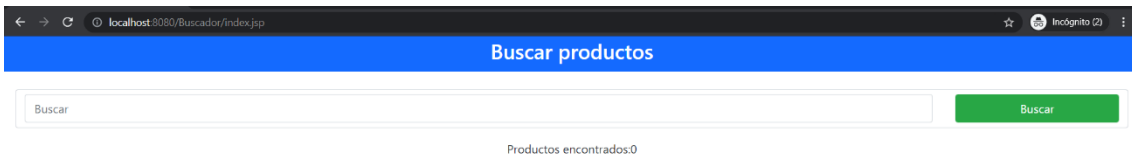


```
Output x Javadoc
Buscador (run) x Java DB Database Process x GlassFish Server 4.1.1 x
library-inclusion-in-archive:
library-inclusion-in-manifest:
compile:
compile-jsp:
Starting GlassFish Server 4.1.1
GlassFish Server 4.1.1 is running.
Incrementally deploying Buscador
Completed incremental distribution of Buscador
run-deploy:
Browsing: http://localhost:8080/Buscador
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 minute 20 seconds)
```

Fuente: elaboración propia

Luego de la ejecución, se muestra el archivo principal index.jsp, el cual contiene la interfaz gráfica del buscador

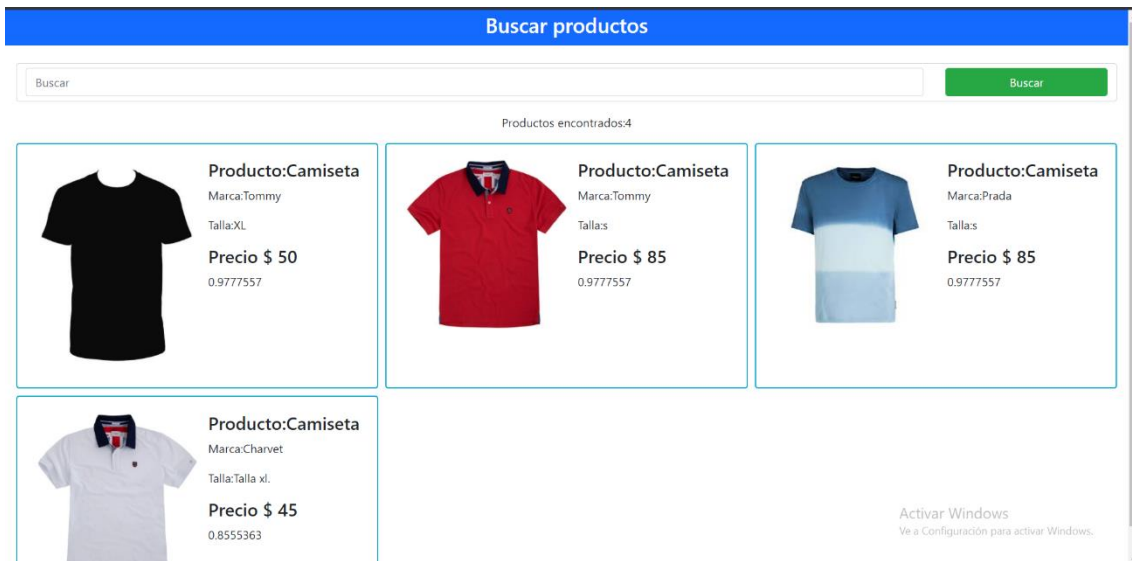
Ilustración 20: interfaz gráfica del buscador



Fuente: elaboración propia

EL buscador es muy simple, ingresas un producto y si existe devuelve un resultado, y si se antepone una Virgulilla (~), devolverá valores aproximados.

Ilustración 21: búsqueda de productos



Fuente: elaboración propia

3. CAPITULO III: EVALUACIÓN DEL PROTOTIPO

3.1. Plan de evaluación

Para la evaluación de los resultados del buscador se divide en dos niveles el primero a nivel sintáctico donde se utiliza métricas de clasificación que ayudan a evaluar la precisión y la relevancia de lo buscado. Por último, a nivel semántico donde se ingresó búsquedas incorrectas para conocer la escala de entendimiento del significado de la palabra.

3.1.1. Precisión

Según [29], indica que la precisión es la cantidad de valores exactamente relevantes recuperados, se puede conocer que tiene mayor precisión cuando el valor se acerca a un uno, mientras que una menor precisión es cuando el valor se acerca al cero.

Ilustración 22: formula de precisión

$$\text{Precisión} = \frac{\text{Documentos relvantes recuperados}}{\text{Documentos recuperados}}$$

Fuente: basada en [30].

3.1.2. Exhaustividad, Recall

Esta métrica ayuda a valorar la sensibilidad, con este término se puede encontrar la cantidad de información relevante de la búsqueda, del total de datos mostrados al usuario final [30].

Ilustración 23: formula de exhaustividad

$$\text{Exhaustividad} = \frac{\text{Documentos relevantes recuperados}}{\text{Documentos relevantes}}$$

Fuente: basada en [30].

3.2. Resultados de evaluación

Para realizar las pruebas se utilizó el recurso nombrado en la **tabla 2**.

3.2.1. Resultados Prueba 1

Esta prueba tiene como objetivo calificar la precisión y la sensibilidad del buscador cuando se ingresa una consulta donde tiene varios productos con el mismo término ingresado, como en este caso **pantalón** se repite en varios productos.

Tabla 6: resultados prueba 1

Consulta 1	Pantalón	
		Buscador
Prod. Relevantes de la muestra		6
Recuperados	Relevantes encontrados	6
	No relevantes	0
	Total	6
No recuperados		0
Precisión		1
Exhaustividad, Recall		1

Fuente: elaboración propia

De los resultados obtenidos muestran que el buscador tiene una precisión del 100% y una sensibilidad del 100%, estos porcentajes indica que, si el valor de la

precisión y sensibilidad se acerca al valor máximo, indicara que los productos obtenidos son relevantes.

3.2.2. Resultados Prueba 2

En esta prueba se realizó una consulta donde se ingresan varios términos como en este caso **camiseta xl** en la cual contiene como respuesta distintos resultados.

Tabla 7: resultados prueba 2

Consulta 1	Camiseta xl	
		Buscador
	Prod. Relevantes de la muestra	6
Recuperados	Relevantes encontrados	3
	No relevantes	6
	Total	9
	No recuperados	0
	Precisión	0.33
	Exhaustividad, Recall	0.50

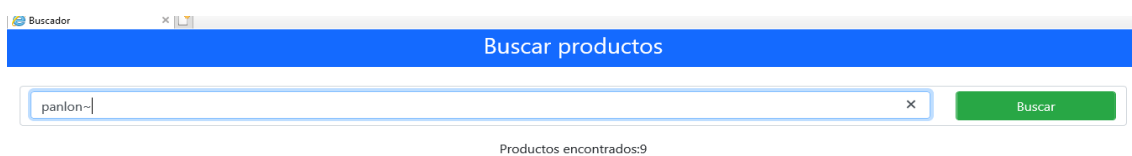
Fuente: elaboración propia

De los resultados obtenidos se puede analizar que se obtuvo una precisión del 33%, debido a que solo hay 3 productos relevantes del total y una sensibilidad del 50% porque determina que de la mitad del total no son relevantes.

3.2.3. Resultados prueba 3

Para esta prueba se ingresó una consulta incorrecta en este caso se escribió **camiseta** incorrectamente acompañado de una virgulilla (~) para mostrar los resultados aproximados al término mencionado

Ilustración 24: prueba 3 ingreso consulta







Fuente: elaboración propia

Ilustración 25: prueba 3 resultados

Buscar productos

Buscar

Productos encontrados:6

	<p>Producto:Pantalon Marca:HUGABO Talla:Talla 30 Precio \$ 52 0.8743564</p>
	<p>Producto:Pantalon Marca:Extreme Talla:Talla xl Precio \$ 85 0.8743564</p>
	<p>Producto:Pantalon Marca:Tommy Talla:Talla m Precio \$ 50 0.8743564</p>
	<p>Producto:Pantalon Marca:Givenchy Talla:Talla m Precio \$ 60 0.8743564</p>

Fuente: elaboración propia

Como resultados se obtuvo que el buscador entendió lo que el usuario trataba de escribir y como respuesta mostró los productos aproximados a esa consulta.

3.3. Conclusiones

Como resultado del buscador aplicando procesamiento de lenguaje natural se puede concluir que:

- A través de las consultas en artículos y revistas científicas se obtuvo el conocimiento fiable, seguro sobre procesamiento de lenguaje natural, para el desarrollo del documento y el prototipo.
- Se adquirió un conocimiento sobre los niveles semántico y sintáctico para establecer diferencias entre los ámbitos de actuación y herramientas con las que trabajar.
- Se pudo establecer una arquitectura en donde se detalla el proceso de la interacción entre el usuario con el buscador.
- Se realizaron distintas pruebas para conocer la precisión, la exhaustividad y el proceso de semántica del buscador.

3.4. Recomendaciones

Para la implementación del buscador aplicando procesamiento de lenguaje natural se recomienda que:

- Instalar las librerías necesarias, para el correcto funcionamiento del proyecto caso contrario se tendría errores al momento de ejecutar el programa.
- Realizar un análisis más profundo de las fuentes bibliográficas para tener mayor conocimiento de técnicas y herramientas tecnológicas del procesamiento de lenguaje natural.
- Tratar de no ingresar una consulta con errores para que los resultados presentados sean más exactos.

4. BIBLIOGRAFIA

- [1] A. C. da S. Oliveira y A. C. Freschi, «LINGÜÍSTICA COMPUTACIONAL: um mapeamento bibliográfico de 2000 a 2020», *Revista Interface Tecnológica*, vol. 17, n.º 2, Art. n.º 2, dic. 2020, doi: 10.31510/infa.v17i2.1056.
- [2] A. E. Pesántez-Calva, J. A. Romero-Correa, y M. L. González-Illescas, «Comercio electrónico B2B como estrategia competitiva en el comercio internacional: Desafíos para Ecuador», *INNOVA Research Journal*, vol. 5, n.º 1, Art. n.º 1, ene. 2020, doi: 10.33890/innova.v5.n1.2020.1166.
- [3] R. Romero-Conde y M. Murguía-Romero, «Red de transición aumentada y lenguaje formal para la danza Bhāratānāṭyam», *RCS*, vol. 114, n.º 1, pp. 83-96, dic. 2016, doi: 10.13053/rcs-114-1-7.
- [4] L. Stanescu y B. Savu, «Automatic Assessment of Narrative Answers Using Information Retrieval Techniques», sep. 2019, pp. 355-358. doi: 10.15439/2019F96.
- [5] E. L. Lydia, S. Satyanarayan, K. V. Kumar, y D. Ramya, «Indexing documents with reliable indexing techniques using Apache Lucene in Hadoop», *Int. J. Intelligent Enterprise*, vol. 7, n.º 1/2/3, pp. 203-214, 2020, doi: 10.1504/IJIE.2020.104656.
- [6] J. C. Phillips *et al.*, «Scalable molecular dynamics on CPU and GPU architectures with NAMD», *The Journal of Chemical Physics*, vol. 153, n.º 4, p. 044130, jul. 2020, doi: 10.1063/5.0014475.
- [7] G. Cano *et al.*, «Predicción de solubilidad de fármacos usando máquinas de soporte vectorial sobre unidades de procesamiento gráfico», *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, vol. 33, n.º 1, pp. 97-102, ene. 2017, doi: 10.1016/j.rimni.2015.12.001.

- [8] D. C. Moreno y M. V. Lombardo, «Ontología y Procesamiento de Lenguaje Natural», *KnE Engineering*, pp. 492-501, feb. 2018, doi: 10.18502/keg.v3i1.1453.
- [9] L. Molina y J. Maldonado, «Implementación de un Método para la Clasificación Automática de Documentos Usando Tareas de Procesamiento de Lenguaje Natural y un Algoritmo de Máxima Entropía», *Revista Venezolana de Computación*, vol. 4, n.º 1, Art. n.º 1, abr. 2018.
- [10] D. Moreira *et al.*, «Análisis del Estado Actual de Procesamiento de Lenguaje Natural», *risti*, pp. 126-136, 2021.
- [11] A. D. Escobar Macías, «Análisis del uso del procesamiento del lenguaje natural y su aplicación en sistemas conversacionales», Thesis, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Networking y Telecomunicaciones, 2019. Accedido: sep. 03, 2021. [En línea]. Disponible en: <http://repositorio.ug.edu.ec/handle/redug/44837>
- [12] M. Cárdenas, «Predicación secundaria en español: representación y análisis sintáctico-semántico», *Arboles y Rizomas Revista de estudios lingüísticos y literarios*, vol. 1, n.º 1, Art. n.º 1, mar. 2019, doi: 10.35588/ayr.v1i1.3771.
- [13] A. C. Cardoso, M. A. Pérez Abelleira, y E. Notario, «Búsqueda de respuestas como aplicación del problema de extracción de relaciones», *RTyC*, n.º 33, pp. 45-64, oct. 2018, doi: 10.33414/rtyc.33.45-64.2018.
- [14] H. G. R. Pescio y V. A. R. Petrasic, «Reconocimiento de patrones lingüísticos para la generación de ontologías aplicadas en la medición del conocimiento», *GPT*, vol. 12, n.º 35, pp. 74-82, 2019.

- [15] A. Fernández, «Aproximación a la Lingüística Computacional y sus herramientas para el trabajo con corpus», *Futhark. Revista de Investigación y Cultura*, pp. 37-52, ene. 2018, doi: 10.12795/futhark.2018.i13.02.
- [16] M. A. A. Ramírez, «Evaluación de herramientas de recuperación de información electrónica», *revista Venezolana de Información, Tecnología y Conocimiento*, vol. 14, n.º 1, pp. 29-50, 2017.
- [17] D. Llanes-Padrón y M. Moro-Cabero, «Records in contexts: un nuevo modelo para la representación de la información archivística en el entorno de la web semántica», *El profesional de la información (EPI)*, vol. 26, n.º 3, pp. 524-532, 2017, doi: 10.3145/epi.2017.may.19.
- [18] A. Perez-Soltero, M. E. Lopez-Muñoz, y M. B. Valenzuela, «Tecnologías de la web semántica para el desarrollo de una memoria organizacional en la atención de usuarios en un departamento de tecnología», *Latin American Journal of Development*, vol. 3, n.º 2, Art. n.º 2, may 2021, doi: 10.46814/lajdv3n2-016.
- [19] D. R. E.- Gallo, W. A. Escobar-Botina, Y. E. Estrada-Bastidas, y S. R. Timarán-Pereira, «Visualizador cartográfico semántico de datos geográficos del municipio de Pasto», *Ventana Informática*, n.º 38, Art. n.º 38, 2018, doi: 10.30554/ventanainform.38.2863.2018.
- [20] C. Lopezosa, L. Codina, y J. Caldera-Serrano, «SEO semántico: Framework ISS para la optimización de sitios intensivos en contenidos», *Cuad. doc. multimed.*, vol. 29, pp. 97-123, jun. 2018, doi: 10.5209/CDMU.60607.
- [21] R. López Carreño y F. J. Martínez Méndez, «Sistemas de recuperación de información implementados a partir de CORD-19: herramientas clave en la

- gestión de la información sobre COVID-19», *revespdoccient*, vol. 43, n.º 4, p. e275, dic. 2020, doi: 10.3989/redc.2020.4.1794.
- [22] M. R. Valarezo Pardo, J. A. Honores Tapia, A. S. Gómez Moreno, y L. F. Vines Sánchez, «Comparación de tendencias tecnológicas en aplicaciones web», *3C Tecnología*, vol. 7, n.º 3, pp. 28-49, sep. 2018, doi: 10.17993/3ctecno.2018.v7n3e27.28-49/.
- [23] H. S. Krohn, «Programación de buscadores en JavaScript para diccionarios digitales», *Cuadernos de Lingüística Hispánica*, n.º 34, Art. n.º 34, jul. 2019, doi: 10.19053/0121053X.n34.2019.9410.
- [24] L. Jiménez-Iglesias, M. Pérez-Montoro, y L. Sánchez-Gómez, «Diseño de información digital: revisión y clasificación de indicadores heurísticos para contenidos web», *EPI*, vol. 26, n.º 6, p. 1029, nov. 2017, doi: 10.3145/epi.2017.nov.03.
- [25] D. Torres-Salinas, P.-Á. Castillo-Valdivieso, Á. Pérez-Luque, y E. Romero-Frías, «Altmétricas a nivel institucional: visibilidad en la Web de la producción científica de las universidades españolas a partir de Altmetric.com», *EPI*, vol. 27, n.º 3, p. 483, jun. 2018, doi: 10.3145/epi.2018.may.03.
- [26] J. da S. A. Neto, D. V. Canne, L. A. C. Ybarra, A. A. dos Santos, J. R. B. Garay, y S. T. Kofuji, «PROPOSTA DE MODELO DE BANCO DE DADOS DISTRIBUÍDO PARA O PRONTUÁRIO ELETRÔNICO ÚNICO DO PACIENTE», *South American Development Society Journal*, vol. 4, n.º 11, Art. n.º 11, ago. 2018, doi: 10.24325/issn.2446-5763.v4i11p266-280.
- [27] D. L. Uribazó, S. V. Marcos, P. F. Oliva, y M. D. D. Dapena, «Ejecución automática de pruebas en entornos empresariales de producción de software», *Informática y Sistemas: Revista de Tecnologías de la Informática*

- y *las Comunicaciones*, vol. 3, n.º 1, Art. n.º 1, ene. 2019, doi: 10.33936/isrtic.v3i1.1585.
- [28] D. Palacios, J. Guamán, y S. Contenido, «Análisis del rendimiento de librerías de componentes Java Server Faces en el desarrollo de aplicaciones web», *Novasinerгия*, ISSN 2631-2654, vol. 1, n.º 2, Art. n.º 2, dic. 2018, doi: 10.37135/unach.ns.001.02.06.
- [29] P. Sánchez-Holgado, M. Martín-Merino Acera, y D. Blanco Herrero, «Del data-driven al data-feeling: análisis de sentimiento en tiempo real de mensajes en español sobre divulgación científica usando técnicas de aprendizaje automático», *Disertaciones*, vol. 13, n.º 1, ene. 2020, doi: 10.12804/revistas.uosario.edu.co/disertaciones/a.7691.
- [30] J. Medrano, «Enfoque Combinado de Word2Vec y 2-grams para la Recuperación de Avisos Clasificados Inmobiliarios Semánticamente Relacionados», *Revista Tecnología y Ciencia*, vol. 39, pp. 195-206, dic. 2020, doi: 10.33414/rtyc.39.195-206.2020.