Bogotá D. C., 31 de agosto de 2023

Señores:
CONSEJO CURRICULAR
TECNOLOGÍA EN SISTEMATIZACIÓN DE DATOS E INGENIERÍA EN TELÉMATICA
Universidad Distrital F.J.C.
Ciudad

Respetados Señores:

En cumplimiento del reglamento de trabajo de grado, atentamente me permito presentar el trabajo de grado titulado "APLICACIÓN DISTRIBUIDA PARA LA DETECCIÓN DE AMBIGÜEDADES EN LOS REQUERIMIENTOS DE SOFTWARE UTILIZANDO TÉCNICAS DE MACHINE LEARNING Y BASES DE CONOCIMIENTO", en la modalidad de Pasantía, desarrollado por los estudiantes **Julian David Niño Gonzalez, 20191678030, 1136885787 y Fabian Mauricio Hernández Capera, 20191678012, 1022364415** para optar al título de Ingenieros en Telemática , el cual cumple con los objetivos de la propuesta inicial.

Cordialmente,



Jairo Hernández Gutiérrez
Tutor
Docente Universidad Distrital
Facultad Tecnológica

Anexos: Carta de la empresa en caso de ser pasantía, cumplimiento de objetivos.

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS	**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS** **FACULTAD TECNOLÓGICA** **FORMATO DE EVALUACIÓN PASANTÍA EN MODALIDAD DE TRABAJO DE GRADO** Proceso: Extensión y Proyección Social	UNIDAD DE EXTENSIÓN FACULTAD TECNOLÓGICA

DÍA:	25	MES:	08	AÑO:	2023

Título del proyecto de pasantía: Aplicación distribuida para la detección de ambigüedades en los requerimientos de software utilizando técnicas de machine learning y bases de conocimiento

Entidad: Eprocess S.A.S. Dependencia: Departamento de Innovación

Docente Director: Jairo Hernández Gutiérrez

Profesional designado por la entidad: María Isabel Cuellar Izquierdo

Nombre Estudiante Fabián Mauricio Hernández Capera Código: 20191678012

Total, N° horas Pasantía: 400 Horas ejecutadas: 400

Características	Escala de Evaluación			
	Excelente	Bueno	Regular	No aplica
Formulación y desarrollo del proyecto	X			
Liderazgo en la toma de decisiones	X			
Capacidad para resolver problemas	X			
Desempeño frente a sus actividades	X			
Aportes de mejoramiento	X			
Desarrollo y aplicación de modelos y metodologías	X			
Participación Interdisciplinaria	X			
Cumplimiento de responsabilidades y compromisos	X			
Relaciones Interpersonales	X			

Observaciones: Ninguna.

Firma Docente Director:

Jairo Hernández Gutiérrez
Tutor
Docente Universidad Distrital
Facultad Tecnológica

Firma Profesional designado por la Entidad:

Maria Isabel Cuellar I.

Nombre: María Isabel Cuellar Izquierdo

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS	**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS** **FACULTAD TECNOLÓGICA** **FORMATO DE EVALUACIÓN PASANTÍA EN MODALIDAD DE TRABAJO DE GRADO**	UNIDAD DE EXTENSIÓN FACULTAD TECNOLÓGICA
	Proceso: Extensión y Proyección Social	

DÍA:	25	MES:	08	AÑO:	2023

Título del proyecto de pasantía: Aplicación distribuida para la detección de ambigüedades en los requerimientos de software utilizando técnicas de machine learning y bases de conocimiento

Entidad: Eprocess S.A.S. Dependencia: Departamento de Innovación

Docente Director: Jairo Hernández Gutiérrez

Profesional designado por la entidad: María Isabel Cuellar Izquierdo

Nombre Estudiante Julián David Niño Gonzalez Código: 20191678030

Total, N° horas Pasantía: 400 Horas ejecutadas: 400

Características	**Escala de Evaluación**			
	Excelente	**Bueno**	**Regular**	**No aplica**
Formulación y desarrollo del proyecto	X			
Liderazgo en la toma de decisiones	X			
Capacidad para resolver problemas	X			
Desempeño frente a sus actividades	X			
Aportes de mejoramiento	X			
Desarrollo y aplicación de modelos y metodologías	X			
Participación Interdisciplinaria	X			
Cumplimiento de responsabilidades y compromisos	X			
Relaciones Interpersonales	X			

Observaciones: Ninguna.

Firma Docente Director:

Jairo Hernández Gutiérrez
Tutor
Docente Universidad Distrital
Facultad Tecnológica

Firma Profesional designado por la Entidad:

Maria Isabel Cuellar I.

Nombre: María Isabel Cuellar Izquierdo

25 de Agosto de 2023
Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería en Telemática
Bogotá.
Coordinacion Tecnologia en Sistematizacion Datos e Ingenieria en Telematica
Unidad de Extensión Facultad Tecnológica.

Estimada.
Coordinación de Tecnologia en Sistematizacion Datos e Ingenieria en Telematica.
Unidad de Extensión Facultad Tecnológica.

Por medio de la presente, nos complace informar que **FABIÁN MAURICIO HERNANDEZ CAPERA** identificado con Cédula de Ciudadanía 1.022.364.415 y **JULIÁN DAVID NIÑO GONZALEZ** identificado con Cédula de Ciudadanía 1.136.885.787, estudiantes de la carrera de **INGENIERÍA EN TELEMÁTICA** en la Universidad Distrital Francisco José de Caldas. **CULMINARON SATISFACTORIAMENTE** la pasantía "Aplicación distribuida para la detección de ambigüedades en los requerimientos de software utilizando técnicas de Machine Learning y bases de conocimiento." en nuestra compañía **EPROCESS S.A.S.** identificada con NIT 9007434425. Dicha pasantía se ejecutó desde el 02 de junio de 2023 hasta el 25 de Agosto de 2023 en la modalidad y contrato de PASANTÍA UNIVERSITARIA dentro del área de innovación de la compañía.

Por otro lado se cumplieron todos los objetivos generales y objetivos específicos propuestos dentro de la pasantía universitaria:

Objetivo General	Desarrollar una aplicación Web distribuida que utilice técnicas de Machine Learning y bases de conocimiento para la detección de ambigüedades en la especificación de los requerimientos de software dentro del desarrollo de cualquier herramienta tecnológica.
Objetivo específico 1	1. Diseñar un método para la resolución de ambigüedades más comunes dentro de la especificación de requerimientos de software usando técnicas de Machine learning y bases de conocimiento.
Objetivo específico 2	2. Crear una aplicación distribuida, intuitiva y eficiente que aplique el método diseñado para resolver ambigüedades en las especificaciones de requerimientos de software, utilizando técnicas de aprendizaje automático y bases de conocimiento.

Objetivo específico 3	3. Implementar un modelo de infraestructura distribuida basado en servicios y alojado en la nube que implemente un sistema de balanceo de carga y aseguramiento con JWT - Json Web Token.

Para finalizar, agradecemos la oportunidad de colaborar con la Universidad Distrital Francisco José de Caldas en la formación de profesionales altamente capacitados y comprometidos con la excelencia. Si requiere información adicional o tiene alguna inquietud, no dude en ponerse en contacto con nosotros.

Cordialmente,



María Isabel Cuellar Izquierdo
Cargo: Coordinadora de proyectos de software
Eprocess S.A.S
Dirección: CALLE 68 SUR 70 D 71 TORRE 6 AP 2204 BONAVISTA 2
Teléfono:3005288642 - 3007910615
Correo electrónico: mici387@gmail.com - maria.cuellar@eprocess.com.co

APLICACIÓN DISTRIBUIDA PARA LA DETECCIÓN DE AMBIGÜEDADES EN LOS REQUERIMIENTOS DE SOFTWARE UTILIZANDO TÉCNICAS DE MACHINE LEARNING Y BASES DE CONOCIMIENTO.



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

FABIÁN MAURICIO HERNANDEZ CAPERA

20191678012

JULIAN DAVID NIÑO GONZALEZ

20191678030

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD TECNOLÓGICA

INGENIERÍA EN TELEMÁTICA

Bogotá D.C.

2023

APLICACIÓN DISTRIBUIDA PARA LA DETECCIÓN DE AMBIGÜEDADES EN LOS REQUERIMIENTOS DE SOFTWARE UTILIZANDO TÉCNICAS DE MACHINE LEARNING Y BASES DE CONOCIMIENTO

FABIÁN MAURICIO HERNANDEZ CAPERA
20191678012
JULIAN DAVID NIÑO GONZALEZ
20191678030

MODALIDAD
PASANTÍA

PRESENTADO PARA OPTAR AL TÍTULO DE: INGENIERO EN TELEMÁTICA.

TUTOR:
JAIRO HERNÁNDEZ GUTIÉRREZ

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD TECNOLÓGICA
INGENIERÍA EN TELEMÁTICA
BOGOTÁ D.C.
2023

AGRADECIMIENTOS

Antes que nada, queremos expresar nuestro más sincero agradecimiento a la Universidad Distrital Francisco José de Caldas, en particular a la facultad de ingeniería y tecnología de la sede Tecnológica ubicada en la sede de Candelaria la nueva, por ofrecernos las herramientas, conocimientos y oportunidades para crecer tanto académica como profesionalmente. ¡Que viva la universidad pública en el país!

Un agradecimiento especial a nuestro tutor de proyecto, Jairo Hernández Gutiérrez, por su inestimable apoyo, consejos, y dirección durante todo el proceso, así como su aporte significativo en nuestra formación académica dentro del ciclo de Ingeniería en Telemática, en materias como RAV - Redes de alta velocidad, Planificación y diseño de redes, entre otras. Su experiencia y conocimiento fueron vitales para la realización de este trabajo.

A nuestros mentores Ingenieros de la compañía Eprocess, Sergio Andrés Reyes y María Isabel Cuellar, quienes nos guiaron y nos encaminaron hacia una ejecución impecable de nuestra pasantía. Son un gran ejemplo para continuar nuestra ruta académica.

Finalmente, a todos aquellos que de alguna u otra manera contribuyeron a la realización de este proyecto, ya sea con palabras de aliento, críticas constructivas o simplemente su presencia en los momentos clave, mi más sincero agradecimiento.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS 8

ÍNDICE DE FIGURAS 9

RESUMEN 12

ABSTRACT 13

INTRODUCCIÓN 14

1.1 TITULO 15

1.2 PLANTEAMIENTO DEL PROBLEMA 15

1.3 OBJETIVOS 17

1.3.1 OBJETIVOS GENERAL 17

1.3.2 OBJETIVOS ESPECÍFICOS 17

1.4 JUSTIFICACIÓN 17

1.5 DELIMITACIÓN Y ALCANCE DEL PROYECTO 19

1.6 MARCO TEÓRICO Y CONCEPTUAL 21

1.6.1 ESTADO DEL ARTE 21

1.6.2 MARCO METODOLÓGICO 26

1.7 SOLUCIÓN TECNOLÓGICA 29

1.8 PRESUPUESTO Y FINANCIACIÓN 31

1.8.1 FACTIBILIDAD ECONÓMICA 31

1.9. CRONOGRAMA. 34

CAPÍTULO 2 - ANÁLISIS Y DESARROLLO DEL PROYECTO 35

2.1 INTRODUCCIÓN 35

2.2 INFORMACIÓN DE LA ORGANIZACIÓN: EPROCESS 35

2.2.1 DESCRIPCIÓN Y PORTAFOLIO 35

2.2.2 MISIÓN 36

2.2.3 VISIÓN 36

2.2.4 SITUACIÓN ACTUAL 36

2.3 HERRAMIENTAS EXISTENTES 37

2.3.1 ELECCIÓN DE LENGUAJE DE PROGRAMACIÓN 37

2.3.2 MÉTODOS PARA EL PROCESAMIENTO DEL LENGUAJE NATURAL 42

2.3.3 CONCLUSIONES CAPITULO 46

CAPÍTULO 3 - DISEÑO DE ARQUITECTURA – MODELO 48

3.1 INTRODUCCIÓN 48

3.2 PINECONE 49

3.3 ÍNDICES PINECONE 51

3.3.1 MÉTRICAS DE DISTANCIA 51

3.4 VECTOR EMBEDDINGS 53

3.4.1 CREACIÓN DE ÍNDICE 55

3.5 CHATGPT - OPENIA 57

3.6 SEGURIDAD - JSON WEB TOKEN 59

3.7 BALANCEO DE CARGA 61

3.8. SERVIDOR DE BASE DE DATOS VECTORIALES PINECONE: 62

3.9 COMPONENTES DEL MÉTODO 68

3.10 DIAGRAMA GENERAL DEL MÉTODO 73

CAPÍTULO 4 - IMPLEMENTACIÓN 77

4.1 INTRODUCCIÓN 77

4.2 MODELO DE INFRAESTRUCTURA DISTRIBUIDA 77

4.2.1 DESCRIPCIÓN RENDER 77

4.3 REPOSITORIOS DE SOFTWARE 78

4.4 IMPLEMENTACIÓN DE INTERFAZ GENERAL 80

4.5 INTERFAZ GRÁFICA RESPONSIVE WEB DESIGN – RWD: 82

4.6. PRUEBAS FUNCIONALES 83

4.6.1 ESCENARIO PARA PRUEBAS FUNCIONALES 83

4.6.2 PRUEBA DE SERVICIOS DE BACKEND DISTRIBUIDOS 85

4.6.3 PRUEBA DE AUTENTICACIÓN JWT – JSON WEB TOKEN 85

4.6.4 PRUEBA DE PROCESAMIENTO DE LENGUAJE NATURAL 86

4.6.5 PRUEBA DE SERVICIOS DE BACKEND DISTRIBUIDOS 87

4.6.6 PRUEBA DE AUTENTICACIÓN JWT – JSON WEB TOKEN 91

CONCLUSIONES 94

RECOMENDACIONES 95

INNOVACIÓN EN EXPERIENCIA DEL USUARIO (USER EXPERIENCE - UX). 95

ADOPCIÓN DE INTELIGENCIA ARTIFICIAL (IA): 95

BIBLIOGRAFÍA 97

ANEXO 1 - FACTIBILIDAD ECONÓMICA 99

ANEXO 3. MANUAL DE USUARIO 102

ANEXO 4. REPOSITORIOS GITHUB 106

ÍNDICE DE TABLAS

Tabla 1. Factibilidad económica recursos humanos 32

Tabla 2.Factibilidad económica recursos técnicos 32

Tabla 3. Gastos del proyecto 33

Tabla 4.Tokenización 44

Tabla 5. Métricas de distancia pinecone 53

Tabla 6. Descripción Render 77

ÍNDICE DE FIGURAS

Figura 1. Fases de Crips – DM 28

Figura 2. Cronograma de actividades 34

Figura 3. Diagrama de Gantt 34

Figura 4. Tokenización 39

Figura 5.WordNet 39

Figura 6. PyWS 40

Figura 7. OurSentence 41

Figura 8. Ambigüedad de la frase. 41

Figura 9. Ambigüedad de la palabra 42

Figura 10. Infraestructura Pinecone 49

Figura 11. Ejemplo Embedding ..... **¡Error! Marcador no definido.**

Figura 12.Ejemplo espacio vectorial 54

Figura 13. Ejemplo red neuronal convolucional 55

Figura 14. Creación índice en pinecone 56

Figura 15. Inserción datos pinecone 56

Figura 16. Visualización índice 57

Figura 17. Plataforma servidor RENDER 61

Figura 18. Activación de balanceo 62

Figura 19. Servidor de BD vectoriales 62

Figura 20. Plan estandar 63

Figura 21.Maquina virtual 66

Figura 22. Ubuntu Desktop 67

Figura 23. Pruebas HA Proxy 67

Figura 24. Servidor BD vectorial 68

Figura 25. Servidor aplicaciones Render 69

Figura 26. Servidor Render Nodo1 71

Figura 27. Servidor Render Nodo1 71

Figura 28. Modelo detención de ambiguedades 73

Figura 29. Repositorio GitHub nodo 1 79

Figura 30. Repositorio GitHub nodo 2 79

Figura 31. Repositorio GitHub frontEnd 80

Figura 32.Versión Móvil 82

Figura 33. Versión de PC 83

Figura 34. Generación token nodo 1 88

Figura 35. Chat conversación nodo 1 88

Figura 36. Chat conversación nodo 2 89

Figura 37. Generación token nodo 2 89

Figura 38. URL nodo 1.....90

Figura 39. URL Nodo 2.....90

Figura 40. Requerimiento el token JWT. 91

Figura 41. Respuesta petición .....**¡Error! Marcador no definido.**

Figura 42. Generación token nodo 1 .....**¡Error! Marcador no definido.**

Figura 43. Generación token nodo 2 92

RESUMEN

En este proyecto de grado, se abordó la creación de una aplicación web de enfoque distribuido, que utiliza técnicas avanzadas de Machine Learning y se apoya en bases de conocimiento, con el objetivo de identificar y resolver ambigüedades presentes en los requerimientos de software. Para lograrlo, se diseñó un método que combina algoritmos de aprendizaje automático con información almacenada en bases de datos vectoriales, permitiendo una detección precisa y eficiente de las ambigüedades más comunes. La aplicación final se destaca por su interfaz de fácil acceso y su rendimiento excepcional, respaldado adicionalmente por una infraestructura distribuida en la nube que asegura una distribución equitativa de la carga y una seguridad robusta a través de la aplicación de JSON Web Tokens. En conjunto, este proyecto culmina en una solución tecnológica innovadora que potencia la calidad y eficiencia en el proceso de desarrollo de software al abordar de manera efectiva las ambigüedades presentes en los requerimientos.

ABSTRACT

In this thesis project, the creation of a distributed approach web application was tackled. The application leverages advanced Machine Learning techniques and draws on knowledge bases to identify and resolve ambiguities within software requirements. To achieve this, a method was designed that blends machine learning algorithms with information stored in vectorized databases, enabling precise and efficient detection of the most common ambiguities. The final application stands out for its user-friendly interface and exceptional performance. Additionally, it is bolstered by a cloud-based distributed infrastructure, ensuring equitable load distribution and robust security through the implementation of JSON Web Tokens. In summary, this project culminates in an innovative technological solution that enhances the quality and efficiency of the software development process by effectively addressing ambiguities present in requirements.

INTRODUCCIÓN

La comprensión completa de los datos fue esencial en el desarrollo de aplicaciones distribuidas, especialmente en la detección de ambigüedades en los requerimientos de software. En este contexto, se presentó la metodología CRISP-DM como un enfoque eficaz. El proyecto tenía como objetivo crear una aplicación distribuida que combinara Machine Learning y bases de conocimiento para identificar y resolver ambigüedades en los requerimientos de software. A través del modelo CRISP-DM, se buscaba mejorar la detección y resolución de ambigüedades, contribuyendo de esta manera al avance en el desarrollo de software.

En un entorno donde coexistían múltiples lenguajes de programación como Python, Java, JavaScript y otros, cada uno ocupaba su espacio según sus respectivas fortalezas. Python se destacaba por su versatilidad y legibilidad, Java por su aplicabilidad empresarial, JavaScript por su papel en el desarrollo web, y otros lenguajes por sus habilidades en sistemas, videojuegos o aplicaciones móviles. El proyecto aspiraba a optimizar la detección y resolución de ambigüedades en el desarrollo de software, mediante la implementación de enfoques avanzados en el manejo de datos y la combinación de tecnologías innovadoras.

CAPÍTULO 1 - FASE DE ORGANIZACIÓN, DEFINICIÓN Y PLANEACIÓN

1.1 TITULO

APLICACIÓN DISTRIBUIDA PARA LA DETECCIÓN DE AMBIGÜEDADES EN LOS REQUERIMIENTOS DE SOFTWARE UTILIZANDO TÉCNICAS DE MACHINE LEARNING Y BASES DE CONOCIMIENTO.

1.2 PLANTEAMIENTO DEL PROBLEMA

El levantamiento de requerimientos de software es un proceso fundamental para el desarrollo de sistemas eficientes y exitosos. Por ello, el uso de herramientas de software asistidas por Machine Learning e Inteligencia Artificial es esencial para identificar de manera detallada las necesidades y requisitos de cualquier sistema informático. Estas herramientas pueden ayudar a los equipos de desarrollo a comprender de manera precisa lo que se espera del software y cómo debe hacerlo, lo que disminuye el riesgo de malentendidos y conflictos en el futuro. Además, el proceso de levantamiento de requerimientos debe ser realizado de manera rigurosa y sistemática para garantizar la calidad del software y la satisfacción de los usuarios.

A pesar de la importancia del levantamiento de requerimientos, este proceso enfrenta diversos desafíos que pueden afectar su éxito. Las ambigüedades lingüísticas y semánticas del lenguaje natural son uno de los principales problemas a los que se enfrentan los equipos de desarrollo. También existen otros desafíos, como la comunicación no efectiva, el acceso limitado de las partes interesadas, el tiempo limitado para el proyecto, entre otros. Por lo tanto, los equipos destinados al levantamiento de requerimientos deben estar preparados para enfrentar estos desafíos y utilizar

herramientas y estrategias efectivas para el levantamiento de requerimientos. Para superar estos desafíos, existen diversas herramientas y estrategias que pueden ayudar a los equipos de desarrollo a clasificar los requerimientos funcionales y no funcionales, así como extraer los requisitos de software de manera efectiva. Estas herramientas y estrategias incluyen enfoques basados en reglas, redes neuronales, aprendizaje, entre otros. Además, el uso de enfoques basados en el conocimiento y el procesamiento del lenguaje natural (PNL) pueden ser útiles para controlar las ambigüedades y mejorar la precisión del proceso de levantamiento de requerimientos pues sin ello se puede generar el fracaso del proyecto de software aumentando los costos y tiempos de respuesta de los diversos equipos de desarrollo.

En conclusión, el levantamiento de requerimientos de software es un proceso fundamental para el desarrollo de sistemas informáticos. Dentro de este proceso se deben utilizar herramientas y estrategias efectivas para enfrentar los diversos desafíos, entre ellos la ambigüedad en la especificación del alcance de un proyecto de software, con esto alcanzar el objetivo de garantizar la calidad del software y la satisfacción de los usuarios. Por último, el uso de herramientas de software asistidas por Machine Learning e Inteligencia Artificial, enfoques basados en el conocimiento y el procesamiento del lenguaje natural, los equipos de desarrollo van a poder realizar el proceso de levantamiento de requerimientos de manera rigurosa y precisa, lo que asegura el éxito del proyecto de software. Para finalizar y teniendo en cuenta todo lo anterior dentro de este proyecto se plantea ¿Cómo pueden la Inteligencia Artificial, el Aprendizaje Automático y el Procesamiento del Lenguaje Natural ayudar en el proceso de levantamiento de requerimientos de software para garantizar la precisión y la calidad del proyecto de software para aminorar las

fallas que generan una mala calidad en el software aumentando costos, tiempo y esfuerzos humanos?

1.3 OBJETIVOS

1.3.1 OBJETIVOS GENERAL

Desarrollar una aplicación Web distribuida que utilice técnicas de Machine Learning y bases de conocimiento para la detección de ambigüedades en la especificación de los requerimientos de software dentro del desarrollo de cualquier herramienta tecnológica.

1.3.2 OBJETIVOS ESPECÍFICOS

1. Diseñar un método para la resolución de ambigüedades más comunes dentro de la especificación de requerimientos de software usando técnicas de Machine learning y bases de conocimiento.

2. Crear una aplicación distribuida, intuitiva y eficiente que aplique el método diseñado para resolver ambigüedades en las especificaciones de requerimientos de software, utilizando técnicas de aprendizaje automático y bases de conocimiento.

3. Implementar un modelo de infraestructura distribuida basado en servicios y alojado en la nube que implemente un sistema de balanceo de carga y aseguramiento con JWT - Json Web Token.

1.4 JUSTIFICACIÓN

El desarrollo de sistemas informáticos eficientes y exitosos depende en gran medida de la precisión y claridad en la captura de requerimientos de software. El proceso de levantamiento de requerimientos constituye un pilar fundamental en la ingeniería de software, ya que sienta las bases

para el diseño, desarrollo e implementación de soluciones tecnológicas que satisfagan las necesidades de los usuarios y las partes interesadas. Sin embargo, este proceso no está exento de desafíos, siendo la ambigüedad en la especificación de requerimientos uno de los obstáculos más críticos que enfrentan los equipos de desarrollo.

La ambigüedad lingüística y semántica presente en el lenguaje natural dificulta la interpretación y comprensión precisa de las necesidades del software, lo que puede derivar en malentendidos, errores en la implementación y, en última instancia, en la insatisfacción de los usuarios. A esto se suman otros obstáculos como la comunicación no efectiva entre las partes interesadas, limitaciones en el acceso a la información relevante y la presión del tiempo en los proyectos.

Para superar estos desafíos y asegurar la calidad de los sistemas desarrollados, es esencial explorar nuevas estrategias y herramientas que permitan abordar la ambigüedad y mejorar la precisión en el levantamiento de requerimientos. En este contexto, la aplicación de tecnologías como la Inteligencia Artificial (IA), el Aprendizaje Automático (AA) y el Procesamiento del Lenguaje Natural (PLN) emerge como una solución prometedora.

La IA, el AA y el PLN ofrecen la capacidad de analizar grandes cantidades de datos en lenguaje natural, identificar patrones y relaciones, y extraer información relevante. Estas tecnologías pueden ser empleadas para desarrollar modelos que detecten y resuelvan ambigüedades en las especificaciones de requerimientos, mejorando así la precisión en la captura de necesidades y facilitando la comunicación entre los equipos de desarrollo y las partes interesadas.

Este proyecto de grado busca explorar a fondo el potencial de la IA, el AA y el PLN en el proceso de levantamiento de requerimientos de software. A través del diseño y desarrollo de una aplicación

web distribuida, se pretende implementar un enfoque innovador que aproveche estas tecnologías para identificar y resolver las ambigüedades más comunes en las especificaciones de requerimientos. Al lograr este objetivo, se espera contribuir significativamente a la reducción de malentendidos y errores en el desarrollo de software, disminuyendo los costos asociados a retrabajos y aumentando la satisfacción de los usuarios.

La relevancia de este proyecto radica en su capacidad para abordar un problema crítico en el ámbito del desarrollo de software, al tiempo que explora el uso pragmático y efectivo de tecnologías emergentes. Además, esta investigación puede servir como punto de partida para futuros desarrollos en el campo de la ingeniería de software, fomentando la aplicación de la IA, el AA y el PLN en otras áreas cruciales del ciclo de vida de desarrollo de software.

En conclusión, este proyecto se justifica por la necesidad apremiante de resolver los desafíos inherentes a la ambigüedad en el proceso de levantamiento de requerimientos de software. La combinación estratégica de la IA, el AA y el PLN puede marcar una diferencia significativa en la calidad y éxito de los proyectos de software, y esta investigación pretende ser un paso importante hacia esa dirección.

1.5 DELIMITACIÓN Y ALCANCE DEL PROYECTO

La resolución de ambigüedades dentro de la especificación de requerimientos de software es un problema común que puede tener un gran impacto en el éxito o fracaso de un proyecto de software. Por esta razón, se ha decidido utilizar técnicas de Machine Learning y bases de conocimiento para desarrollar un método efectivo para la resolución de ambigüedades en los requerimientos de software. Para lograr este objetivo, se va a utilizar la metodología CRIPS-DM, que consta de seis

fases importantes: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue.

A través de la metodología CRIPS-DM, se espera desarrollar una aplicación web que implemente el método descubierto. Además, se va a implementar un modelo de infraestructura distribuida basado en servicios y alojado en la nube, que incluirá un sistema de balanceo de carga y seguridad de JWT (Json Web Token). El método descubierto se aplicará en este modelo posterior a la investigación planteada en las primeras fases de la solución tecnológica, se espera que este desarrollo sea una solución adecuada y apropiada para las dificultades actuales en los requerimientos de software. Los estudios y análisis realizados parecen indicar que el proyecto tendría un impacto positivo en la sociedad y se ha llevado a cabo una proyección financiera viable para respaldar esta afirmación. La implementación del proyecto no presenta ninguna limitante para los estudiantes y la Universidad Distrital Francisco José de Caldas, lo que hace que la puesta en funcionamiento de este proyecto sea completamente viable.

En conclusión, una vez completado este proyecto, se espera contar con u8na aplicación distribuida que utilice técnicas de Machine Learning y bases de conocimiento para detectar y disminuir o eliminar ambigüedades en los requerimientos de software. Esto permitirá la creación de documentos de software más precisos y detallados, lo que a su vez ayudará a asegurar el éxito del proyecto en general. En conclusión, se espera que al finalizar se cuente con los siguientes entregables:

Informe gerencial del método: Documento que describe el método propuesto para resolver ambigüedades usando Machine Learning y bases de conocimiento.

Infraestructura de nube: Se requiere entregar el sistema configurado en la infraestructura distribuida en la nube donde se encuentre implementado el método propuesto, junto con la configuración de balanceo de carga y JWT.

Código fuente: Repositorio en GitHub o similares donde repose el código completo de la aplicación distribuida que implementa el método diseñado.

Documentación: Manual de usuario donde se incluirá información de balanceo de carga y JWT (Json Web Token).

Pruebas y validaciones: Informe de pruebas y resultados obtenidos al evaluar el método y la aplicación.

Presentación final: Presentación en diapositivas con información que resuma el proyecto, sus objetivos, resultados y conclusiones.

1.6 MARCO TEÓRICO Y CONCEPTUAL

1.6.1 ESTADO DEL ARTE

En todo el proceso del Ciclo de Vida del Desarrollo de Software (SDLC), la fase de recopilación e ingeniería de requisitos juega un papel muy importante. Es la etapa de inicio y el éxito en esta etapa tiene un gran impacto en el éxito de las etapas posteriores y el proyecto en general.

En los últimos años, la ingeniería de requisitos de software (RE), que es una subdisciplina de la ingeniería de software (SE), ha ganado atención para las aplicaciones de IA. El campo de RE ha evolucionado en las últimas décadas a medida que aumenta la necesidad de comprender y analizar

las necesidades del usuario para producir software de calidad. En los primeros días de SE, el problema de la falta de análisis de requisitos se hizo cada vez más evidente. El software creado sin identificar las necesidades resulta en productos de diseño deficiente, retrasados o incluso cancelados (Liu et al., 2022)

La inteligencia artificial (IA) se ha aplicado a diversos campos, desde la medicina hasta la agricultura, la fabricación y especialmente, la ingeniería en particular, en informática, la IA se ha aplicado a varios campos como la minería de datos, el procesamiento del lenguaje natural, la robótica, el reconocimiento de imágenes, la analítica visual y la computación en la nube (Liu et al., 2022).

La ingeniería de requisitos de software ha ganado atención en las aplicaciones de inteligencia artificial (IA) en los últimos años. Esto se debe a que cada vez es más importante comprender y analizar las necesidades del usuario para producir software de calidad, esto conlleva a una problemática la cual es no analizar adecuadamente los requisitos del software, por esto se ha vuelto más evidente en las últimas décadas y puede resultar en un software mal diseñado, retrasado o incluso cancelado.

La validación de requisitos es un paso crucial para garantizar que el proceso de desarrollo de software se lleve a cabo de manera efectiva y que el producto final satisfaga las necesidades del cliente. Para lograr esto, es necesario asegurarse de que los requisitos establecidos en el documento de Especificación de Requisitos de Software (SRS) sean claros, concisos y no ambiguos. Osman y Zaharin (Liu et al., 2022) utilizaron el Procesamiento del Lenguaje Natural (NLP) para identificar palabras clave que contribuyen a la ambigüedad de los requisitos. Luego clasificaron los requisitos

como ambiguos o no ambiguos, siendo el clasificador Random Forest (RF) el que produjo los mejores resultados.

La dependencia del personal puede eliminarse mediante el uso de un Chatbot. Un chatbot se define en el Oxford English Dictionary como "un programa de computadora diseñado para simular la conversación con usuarios humanos, especialmente a través de Internet (Rajender Kumar Surana et al., 2019)".

El uso de la tecnología de chatbots ha ganado popularidad en los últimos años debido a su capacidad para simular conversaciones humanas a través de la tecnología de inteligencia artificial (IA). Un chatbot es un programa de computadora diseñado para interactuar con usuarios humanos y resolver problemas en diferentes áreas.

Para lograr esto, el chatbot utiliza una base de conocimientos que se compone de una representación de conocimiento y un motor de inferencia. La representación de conocimiento es la forma en que se organiza y almacena la información, mientras que el motor de inferencia es el proceso mediante el cual el chatbot utiliza esta información para procesar y responder a las preguntas de los usuarios. Además, el chatbot utiliza técnicas de comprensión del lenguaje natural, generación del lenguaje natural, preprocesamiento de texto y varios algoritmos de aprendizaje automático para comprender y responder adecuadamente a las preguntas de los usuarios.

Esta Base de conocimientos contiene dos partes, Representación de conocimiento y Motor de inferencia. El chatbot utiliza esta Base de conocimientos. El chatbot utiliza la comprensión del lenguaje natural, la generación del lenguaje natural, el preprocesamiento de texto y varios

algoritmos de aprendizaje automático para comprender la respuesta del usuario, tomar medidas adicionales y responder en consecuencia. (Rajender Kumar Surana et al., 2019)

El chatbot utiliza Natural Language Understanding (Comprensión del Lenguaje Natural), Natural Language Generation (Generación del Lenguaje Natural), Preprocesamiento de Texto y varios algoritmos de aprendizaje automático para comprender la respuesta del usuario, tomar acciones adicionales y responder en consecuencia. (Rajender Kumar Surana et al., 2019)

La Ingeniería de Requisitos se enfoca en la identificación, documentación, validación y gestión de los requisitos de un sistema, con el objetivo de garantizar que el software entregado cumpla con los requisitos del cliente y usuarios finales.

La construcción de glosarios es una práctica común en la Ingeniería de Requisitos, ya que ayuda a definir un conjunto común de términos y conceptos que se utilizan en la documentación de requisitos y especificaciones de software. Los glosarios permiten a los desarrolladores y clientes compartir un entendimiento común de los términos y conceptos que se utilizan en el proceso de desarrollo de software, lo que reduce la ambigüedad y facilita la comunicación.

Sin embargo, a pesar de la importancia de los glosarios, un estudio muestra que muchos términos suelen ser usados con significados cambiantes a lo largo del proceso de construcción del software, lo que pone en cuestión su eficacia. Por lo tanto, es importante establecer metodologías y buenas prácticas en la construcción de glosarios, para garantizar que los términos y conceptos se definan de manera clara y precisa y se mantengan consistentes a lo largo de todo el proceso de desarrollo.

Por último, en la actualidad existen sistemas que tienen como objetivo facilitar la detección de ambigüedades en los requerimientos de software. Estos sistemas buscan mejorar la precisión y

eficiencia del proceso de levantamiento de requerimientos, utilizando técnicas de Machine Learning y procesamiento del lenguaje natural para identificar y clasificar de manera más precisa los requisitos del software. Además, estos sistemas pueden ayudar a los equipos de desarrollo a reducir los riesgos de malentendidos y conflictos en el futuro, lo que a su vez puede mejorar la calidad del software y la satisfacción del usuario final, entre ellos podemos ver los siguientes:

ReQtest es una herramienta de gestión de requerimientos que se centra en optimizar la comunicación y colaboración durante el proceso de desarrollo de software. Su módulo de trazabilidad permite a los equipos identificar y resolver ambigüedades en los requerimientos, mientras que las herramientas de seguimiento de errores y pruebas garantizan que los requerimientos se cumplan de manera efectiva. Con ReQtest, los equipos pueden trabajar de manera más eficiente, mejorando la calidad y eficacia del software que desarrollan (ReQtest, 2020).

IBM Rational DOORS es una herramienta de gestión de requerimientos que permite a los equipos de desarrollo de software definir, rastrear y gestionar sus requerimientos de manera eficiente. Esta herramienta proporciona una estructura de trabajo sólida que ayuda a los equipos a identificar y resolver ambigüedades en los requerimientos, y también garantiza que los requerimientos se cumplan de manera efectiva. Con IBM Rational DOORS, los equipos pueden trabajar de manera más efectiva y mejorar la calidad del software que desarrollan (IBM, 2021).

Jama Connect es una herramienta de gestión de requerimientos que brinda a los equipos la capacidad de colaborar efectivamente en la definición de requerimientos. Su entorno de trabajo intuitivo y fácil de usar permite a los equipos identificar y resolver ambigüedades en los requerimientos de manera eficiente. Con Jama Connect, los equipos pueden trabajar de manera

más productiva y asegurarse de que los requerimientos se cumplan de manera efectiva, lo que contribuye a la calidad general del software que desarrollan (Jama Software, 2021).

CaliberRM es una herramienta de gestión de requerimientos que brinda a los equipos la capacidad de definir y gestionar los requerimientos de manera efectiva. Su marco de trabajo sólido ayuda a los equipos a identificar y resolver ambigüedades en los requerimientos de manera eficiente, y también asegura que los requerimientos se cumplan de manera efectiva. Con CaliberRM, los equipos pueden mejorar su productividad y la calidad del software que desarrollan (Micro Focus, 2021).

Enterprise Architect es una herramienta de modelado y diseño que ayuda a los equipos a definir y gestionar los requerimientos de manera efectiva. Su entorno de trabajo sólido permite a los equipos identificar y resolver ambigüedades en los requerimientos de manera eficiente y asegurar que los requerimientos se cumplan de manera efectiva. Con Enterprise Architect, los equipos pueden trabajar de manera más efectiva, lo que contribuye a la calidad general del software que desarrollan (Sparx Systems, 2021).

1.6.2 MARCO METODOLÓGICO

CRISP-DM (Cross-Industry Standard Process for Data Mining) es una metodología de minería de datos que fue desarrollada en los años 90 por un grupo de expertos en minería de datos de la industria y académicos de diferentes países. La metodología fue diseñada para proporcionar una estructura de trabajo sistemática y estándar para los proyectos de minería de datos, lo que permitió a los profesionales de la industria trabajar de manera más eficiente y efectiva. Desde entonces, CRISP-DM ha sido adoptada ampliamente y se ha convertido en una metodología de minería de

datos líder en la industria y la academia, utilizada en una variedad de campos, desde finanzas hasta marketing y ciencias de la salud. Consta de seis fases iterativas:

Comprensión del negocio: Dentro de esta fase se realiza la identificación de los objetivos y requerimientos del negocio para el proyecto de minería de datos, así como la identificación de las limitaciones y recursos disponibles (Hossain et al., 2019).

Comprensión de los datos: En esta fase, se recolecta y se explora la información disponible para el proyecto, con el fin de familiarizarse con los datos y determinar si son adecuados para cumplir con los objetivos del proyecto (Silalahi & Sinaga, 2021).

Preparación de los datos: Esta fase involucra la selección, limpieza y transformación de los datos para su uso en el análisis posterior (García et al., 2021).

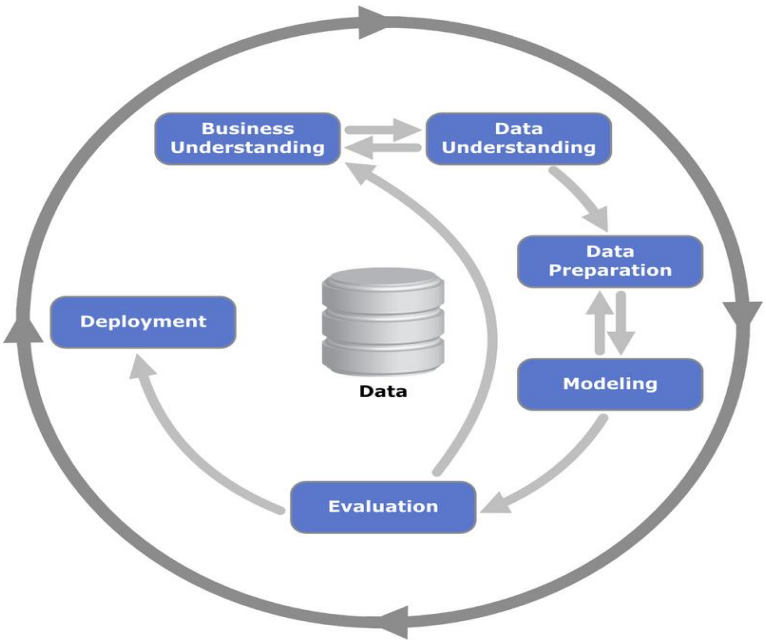
Modelado: Aquí se aplican técnicas de modelado para construir y evaluar modelos que puedan ayudar a alcanzar los objetivos del proyecto (Hossain et al., 2019).

Evaluación: En esta fase, se evalúan los modelos construidos para determinar si satisfacen los objetivos del negocio (Silalahi & Sinaga, 2021).

Despliegue: En la fase final, los modelos se implementan en un ambiente de producción y se monitorean para asegurarse de que sigan cumpliendo con los objetivos del negocio (García et al., 2021).

Figura 1

Fases de Crips – DM



Nota. Vallalta,J. (2023). CRISP-DM: una metodología para minería de datos en salud. IA Health Data Miner, portal web de https://healthdataminer.com/data-mining/crisp-dm-una-metodologia-para-mineria-de-datos-en-salud/#:~:text=Preparaci%C3%B3n%20de%20los%20datos.,Despliegue.

Ventajas

CRISP-DM proporciona una estructura completa y sistemática, lo que le permite abordarlo de manera organizada y efectiva.

La metodología se enfoca en el entendimiento del negocio y los objetivos del proyecto antes de pasar a la preparación y análisis de los datos, lo que le permitirá asegurarnos que la solución encontrada se ajuste al resultado esperado en la identificación de ambigüedades en los requerimientos de Software.

CRISP-DM es una metodología flexible y adaptable, lo que le permitirá ajustar el proyecto en caso de encontrar mejores formas de identificar y/o clasificar las ambigüedades en el proceso de levantamiento de requerimientos de software.

La metodología facilita la integración de técnicas de aprendizaje automático y análisis estadístico, lo que es esencial para la detección de ambigüedades en los requisitos de software utilizando técnicas de Machine Learning y bases de conocimiento.

CRISP-DM también proporciona una evaluación sistemática y objetiva del modelo de minería de datos, lo que nos permitirá determinar la eficacia y precisión de la aplicación distribuida que está desarrollando para la detección de ambigüedades en los requisitos de software.

En conclusión, esta metodología es adecuada para el proyecto, porque nos brinda la posibilidad de realizar fases iterativas centrando el objetivo de la metodología en trabajar sobre la "detección de ambigüedades en los requisitos de software", este procedimiento nos brinda un enfoque en el análisis de datos y la detección de patrones utilizando técnicas de aprendizaje automático y bases de conocimiento. CRISP-DM nos permite abordar el proyecto de manera sistemática, desde la comprensión de los requisitos del negocio hasta la implementación y evaluación de una aplicación distribuida para la detección de ambigüedades en los requisitos de software.

1.7 SOLUCIÓN TECNOLÓGICA

Para afrontar los desafíos que conllevaba el levantamiento de requerimientos de software y asegurar la precisión y calidad del proyecto, se planteó el desarrollo de una plataforma de asistencia basada en Inteligencia Artificial (IA) para el levantamiento de requerimientos. Dicha plataforma integraba técnicas de Aprendizaje Automático y Procesamiento del Lenguaje Natural (PLN). El

propósito de esta solución tecnológica era brindar apoyo a los equipos de desarrollo en la comprensión, análisis y documentación efectiva de los requisitos del software, con miras hacia el futuro.

1.7.1 ANÁLISIS Y EXTRACCIÓN DE REQUERIMIENTOS

Utiliza técnicas de PNL para identificar patrones en el lenguaje natural y determinar los posibles requerimientos funcionales y no funcionales.

1.7.2 DETECCIÓN DE AMBIGÜEDADES Y CONFLICTOS

Mediante la utilización de inteligencia artificial, la plataforma tiene la capacidad de identificar ambigüedades y conflictos potenciales que puedan surgir entre los requerimientos previamente identificados. Esta innovadora función permite una evaluación más exhaustiva y precisa de los elementos involucrados, lo que a su vez facilita la toma de decisiones informadas y la resolución anticipada de posibles problemas. La IA despliega su capacidad analítica para examinar detenidamente los datos y patrones, brindando a los usuarios una herramienta valiosa para optimizar la planificación y ejecución de proyectos de manera eficiente y efectiva.

1.7.3 ANÁLISIS DE AMBIGÜEDADES

La plataforma utilizará algoritmos de Procesamiento del Lenguaje Natural (NLP) y Machine Learning para identificar patrones de ambigüedad en los requerimientos.

Base de Conocimiento Integrada: Se incorporará una base de conocimiento contextual para enriquecer la comprensión del contexto y la terminología de la industria, mejorando así la precisión de las sugerencias.

1.7.4 APLICACIÓN WEB INTERACTIVA

La solución se presentará como una aplicación web interactiva que permitirá a los usuarios cargar sus requerimientos y recibir recomendaciones precisas.

Se implementará una infraestructura distribuida en la nube con un sistema de balanceo de carga y seguridad JWT (Json Web Token) para garantizar el rendimiento y la seguridad de los datos.

1.8 PRESUPUESTO Y FINANCIACIÓN

1.8.1 FACTIBILIDAD ECONÓMICA

El proyecto de grado para la Ingeniería en Telemática "Aplicación distribuida para la detección de ambigüedades en los requerimientos de software utilizando técnicas de Machine Learning y bases de conocimiento." es completamente viable, la factibilidad económica del proyecto es alta, ya que lo que se requiere en términos financieros son:

- Dos equipos de cómputo.
- Asesorías de los tutores del proyecto
- Acceso a Internet

En las tablas que se presentarán a continuación se describe la factibilidad económica, identificando los costos de hardware, software y recursos humanos necesarios para la realización del proyecto de pasantía que se propone.

Tabla 1

Factibilidad económica recursos humanos.

Tipo	Descripción	Valor-Hora	Cantidad	Total
Tutor 1	Asesorías para la realización del proyecto, referente a la metodología.	$ 100.000	200	$ 20.000.000
Desarrolladores	Dos (2) programadores que realicen la implementación de la solución.	$ 60.000	720 horas. 360 horas por estudiante 15 horas semanales x seis meses	$ 43.200.000
Asesor Eprocess	Dos (2) asesores de Eprocess para la realización del proyecto, referente a la metodología.	$100.000	300	$ 30.000.000
Total Recursos Humanos				$ 93.200.000

Nota. La tabla muestra el presupuesto en pesos colombianos que implicaría la ejecución de este proyecto, considerando la asignación de recursos humanos necesarios. Elaboración propia.

A continuación, en la Tabla 2 se presentarán los gastos de los recursos que se ostentan en el desarrollo del proyecto.

Tabla 2

Factibilidad económica recursos técnicos

Recurso	Descripción	Valor Unitario	Cantidad	Total
Computadores (Servidor - clientes)	Equipos portátiles para el desarrollo y las pruebas del sistema.	$ 6.000.000	2	$ 12.000.000

Total Recursos Técnicos	$ 12.000.000

Nota. Esta tabla proporciona una evaluación detallada de los aspectos financieros relacionados con la viabilidad de los recursos técnicos necesarios para el proyecto. Elaboración propia.

A continuación, en la tabla 3 se relacionan los gastos totales del proyecto

Tabla 3.

Gastos del proyecto

Recurso	**Valor**
Total Recursos Humanos	$ 93.200.000
Total Recursos Técnicos	$ 12.100.000
Total Otros recursos	$ 0
Costos imprevistos (10%)	$ 10.530.000
TOTAL COSTO	$ 115.830.000

Nota. Esta tabla se incluye el costo total de la asignación de recursos humanos, así como los costos asociados con los equipos. Elaboración propia.

1.9. CRONOGRAMA.

Figura 2

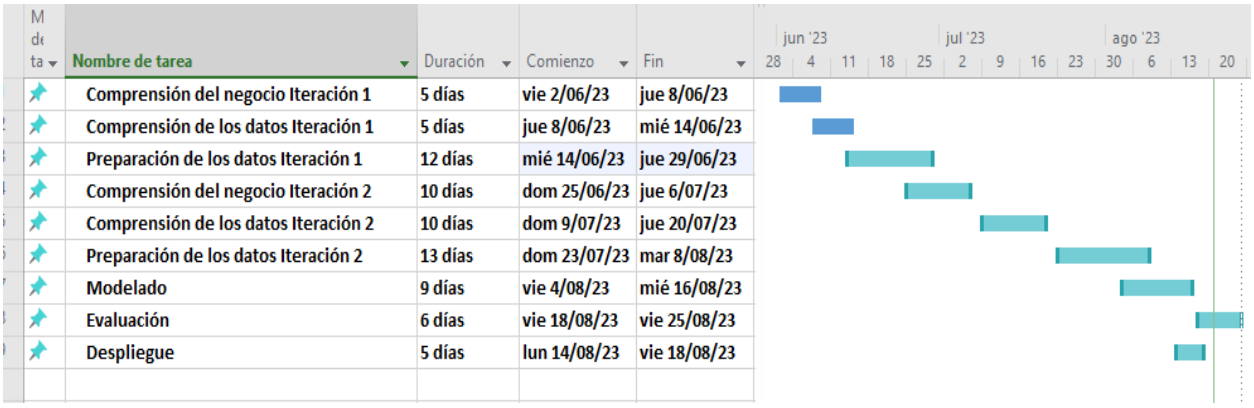
Cronograma de actividades

M de ta	Nombre de tarea	Duración	Comienzo	Fin
	Comprensión del negocio Iteración 1	5 días	vie 2/06/23	jue 8/06/23
	Comprensión de los datos Iteración 1	5 días	jue 8/06/23	mié 14/06/23
	Preparación de los datos Iteración 1	12 días	mié 14/06/23	jue 29/06/23
	Comprensión del negocio Iteración 2	10 días	dom 25/06/23	jue 6/07/23
	Comprensión de los datos Iteración 2	10 días	dom 9/07/23	jue 20/07/23
	Preparación de los datos Iteración 2	13 días	dom 23/07/23	mar 8/08/23
	Modelado	9 días	vie 4/08/23	mié 16/08/23
	Evaluación	6 días	vie 18/08/23	vie 25/08/23
	Despliegue	5 días	lun 14/08/23	vie 18/08/23

Nota. En el gráfico se evidencia el cronograma planificado para el proyecto de investigación. Elaboración propia

Figura 3

Diagrama de Gantt



Nota. Diagrama de Gantt donde en el eje horizontal exhibe las fechas correspondientes a las distintas etapas del proyecto, mientras que en el eje vertical se indica la duración estimada en semanas para cada fase. Elaboración propia.

CAPÍTULO 2 - ANÁLISIS Y DESARROLLO DEL PROYECTO

2.1 INTRODUCCIÓN

En el mundo de la ingeniería de software, se destaca el crucial proceso de levantamiento de requerimientos. En este escenario de constante evolución e innovación, emerge con determinación Eprocess S.A.S, una empresa con la ambiciosa aspiración de incursionar en este dinámico ámbito de la IA.

En este contexto, la elección del lenguaje de programación apropiado adquiere un rol esencial para lograr el éxito en las tareas relacionadas con el procesamiento del lenguaje natural (PLN). En este desafiante inicio, se ha optado por Python como su lenguaje de programación principal, y se apoya en la base de datos vectorial Pinecone.

2.2 INFORMACIÓN DE LA ORGANIZACIÓN: EPROCESS

2.2.1 DESCRIPCIÓN Y PORTAFOLIO

Eprocess S.A.S llega al mercado colombiano con una propuesta innovadora y de calidad, a sus necesidades tecnológicas en desarrollo de software a la medida (aplicaciones de escritorio y móviles), soporte técnico, suministro de personal especializado y tercerización de fábrica de Software bajo el modelo de outsourcing total.

Gracias a la experiencia de más de 10 años de nuestro equipo profesional, nuestros productos y servicios se caracterizan por su versatilidad y adaptabilidad a sus necesidades particulares, enfocando nuestra atención en la satisfacción del cliente, siempre apoyados de nuestro talento humano.

Nuestros servicios son:

- Desarrollo de Software a la medida
- Soporte Técnico
- Suministro de Personal Especializado
- Consultoría en Servicios Administrativos}

2.2.2 MISIÓN

Garantizar a nuestros clientes productos y servicios tecnológicos de la mejor calidad que superen sus expectativas, apoyados de un recurso humano y tecnológico que nos permita ser reconocidos como una compañía innovadora y creativa generando valor para nuestros funcionarios y clientes, contribuyendo así al desarrollo del país.

2.2.3 VISIÓN

Para el año 2020, Eprocess se propuso firmemente sobresalir como una compañía innovadora y altamente competitiva en el ámbito del diseño de software y soluciones tecnológicas de primera categoría. Esta visión llevó consigo el compromiso de generar un valor excepcional tanto para sus valiosos clientes como para los empleados que conforman su talentoso equipo. Más allá de su enfoque en el mundo empresarial,

2.2.4 SITUACIÓN ACTUAL

Actualmente Eprocess, se encuentra en una posición estratégica abordando desafíos y oportunidades que presenta el entorno empresarial en Colombia. La organización se enfoca en la implementación de soluciones de software personalizadas y en la transformación digital de los

procesos comerciales, liderando el camino hacia la automatización de procesos y la toma de decisiones basada en datos. Aunque se enfrenta a un entorno altamente competitivo en el sector tecnológico en constante evolución, se destaca por su enfoque en la calidad, la innovación y la colaboración cercana con los clientes. Además de las oportunidades que brinda el auge de la transformación digital, la empresa también aborda los desafíos críticos de seguridad de datos, demostrando su compromiso con la responsabilidad tecnológica.

2.3 HERRAMIENTAS EXISTENTES

2.3.1 ELECCIÓN DE LENGUAJE DE PROGRAMACIÓN

La elección de un lenguaje de programación y una base de datos vectorial adecuados desempeña un papel crucial en la implementación y eficacia de las tareas de procesamiento del lenguaje natural (PLN). Se ha decidido utilizar el lenguaje de programación Python en combinación con Pinecone como base de datos vectorial.

Al contar con Python como lenguaje de programación, se encuentran diversas razones que respaldan esta elección. Entre ellas se destacan la versatilidad y facilidad de uso inherentes a Python, especialmente notables debido a su idoneidad para el procesamiento del lenguaje natural. Proporcionando un entorno flexible, Python se convierte en una plataforma propicia para la implementación de algoritmos complejos y soluciones específicas destinadas a una amplia variedad de tareas. Desde análisis sintácticos hasta generación de lenguaje, sus capacidades abarcan un espectro diverso.

Otro motivo relevante para optar por estos lenguajes de programación es la vibrante comunidad de desarrollo de Python, cuyo resultado se refleja en la creación de una extensa gama de bibliotecas

y marcos de trabajo especializados en Procesamiento del Lenguaje Natural (PLN). Ejemplares como NLTK, spaCy y Transformers proveen herramientas y modelos preconstruidos que agilizan tanto la implementación como el análisis de labores en el ámbito del PLN. La existencia de estas bibliotecas disminuye la necesidad de construir todo desde cero, incentivando una implementación más eficaz y eficiente.

La elección de Python también descansa en su capacidad innata para integrarse sin inconvenientes con diversas tecnologías. En esta ocasión, se erige como la opción óptima para interactuar con Pinecone, la base de datos vectorial que ha sido seleccionada. La fluidez en la comunicación entre ambas tecnologías posibilitará el almacenamiento y la consulta de vectores pertinentes para tareas de PLN, tales como la evaluación de similitud semántica y la recuperación de información.2.3.2 Python

NLTK: Es la biblioteca con la que muchas personas comienzan su aventura en el procesamiento de lenguaje natural. Ofrece herramientas útiles para tareas como la preparación previa de textos, la creación de unidades de palabras (tokens), la reducción de palabras a su forma base (stemming) y la asignación de etiquetas a palabras según su función gramatical (POS tagging).

TextBlob: Se alza como una capa adicional sobre NLTK y se distingue por su facilidad de uso. Además de las funciones que proporciona NLTK, incorpora características adicionales, como el análisis de sentimientos en un texto y la corrección ortográfica.

Gensim: Diseñada específicamente para el modelado de temas, esta biblioteca ofrece varias técnicas, como LDA y LSI. También tiene la capacidad de calcular la similitud entre documentos.

SpaCy: Similar a NLTK en términos de funcionalidad, pero se destaca por su rapidez y eficiencia en el procesamiento.

WebScraping: Consiste en la obtención de contenido textual desde diferentes páginas web, permitiendo recopilar información valiosa de la internet.

Figura 4

Tokenización

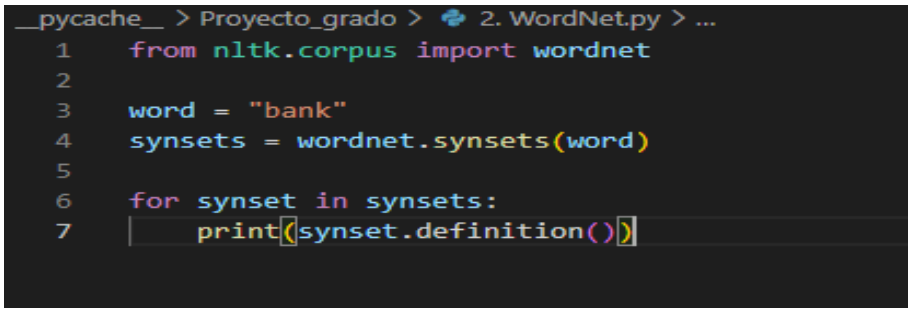


Nota. La imagen muestra un ejemplo de Tokenización en el contexto de procesamiento de lenguaje natural. Utilizando la biblioteca NLTK en Python, se realiza el proceso de descomponer un fragmento de texto en unidades más pequeñas conocidas como tokens. Elaboración propia

Por otro lado, el código realiza el etiquetado de partes del discurso en las palabras de la oración proporcionada y muestra las etiquetas POS resultantes en la consola.

Figura 5

WordNet

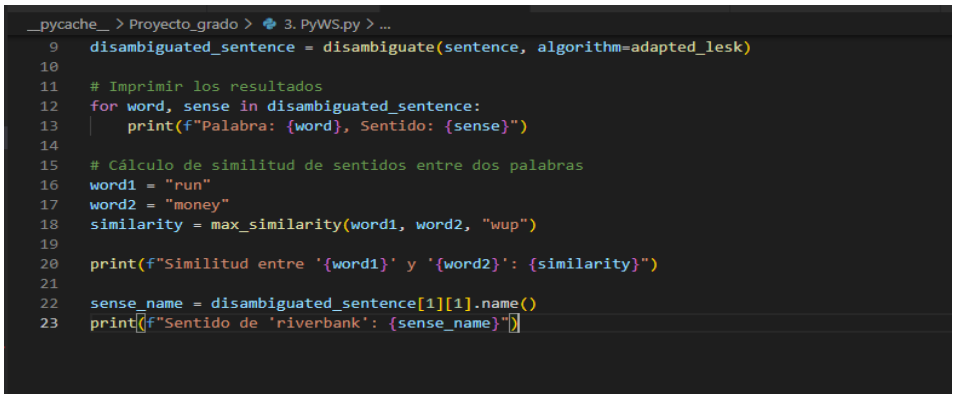


Nota. La imagen ilustra el uso de WordNet a través de la biblioteca NLTK en un contexto de procesamiento de lenguaje natural. Donde WordNet se emplea como una base de datos léxica. Elaboración propia.

El código busca los synsets asociados con la palabra "bank" utilizando wordnet de NLTK y muestra las definiciones de cada synset. Esto puede ser útil para explorar los diferentes significados y contextos en los que la palabra "bank" puede ser utilizada.

Figura 6

PyWS

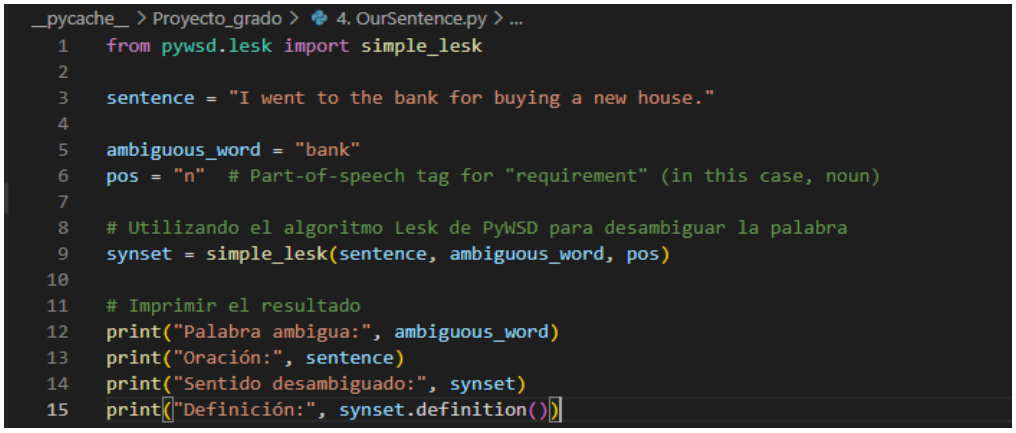


Nota. La imagen exhibe el empleo de PyWS, una herramienta que permite acceder y buscar en WordNet mediante Python. Elaboración propia

A continuación, se muestra cómo utilizar la biblioteca pywsd para llevar a cabo la desambiguación del sentido de palabras en una frase, calcular la similitud de sentidos entre dos palabras y obtener el nombre del sentido desambiguado.

Figura 7

OurSentence

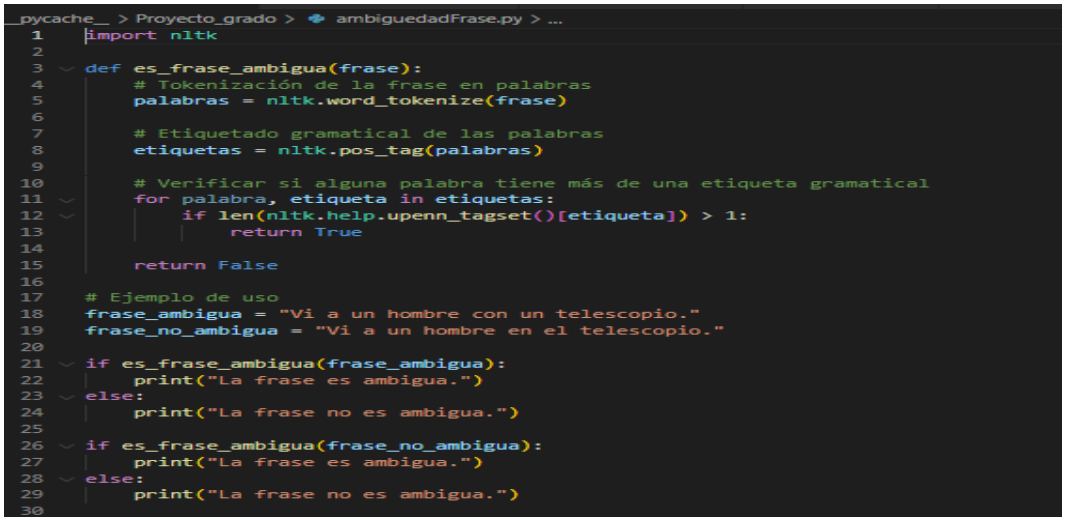


Nota. La imagen ilustra la aplicación de "OurSentence", una herramienta especializada en el análisis de estructuras de oraciones en el contexto del procesamiento de lenguaje natural. Elaboración propia

El código ilustra cómo utilizar la función simple_lesk de la biblioteca pywsd para desambiguar el sentido de una palabra en una oración y cómo obtener detalles sobre el sentido seleccionado, como su definición

Figura 8

Ambigüedad de la frase.

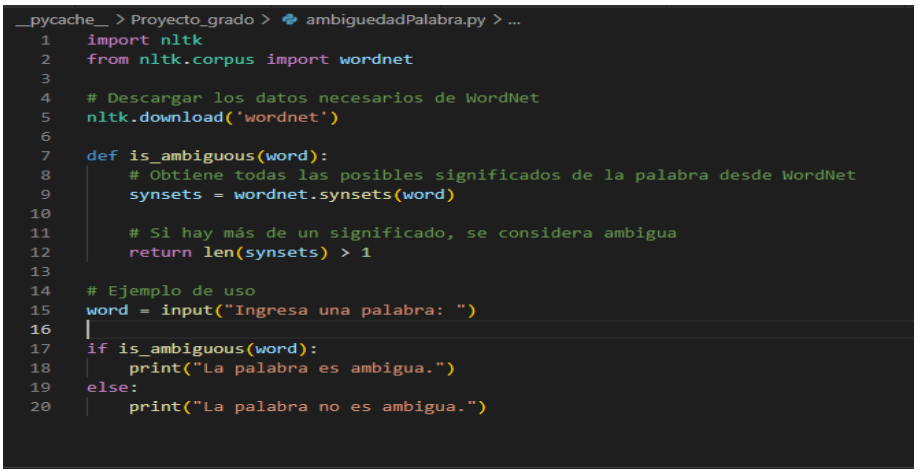


Nota. La figura muestra un ejemplo ilustrativo de la ambigüedad presente en una frase, analizada desde una perspectiva de procesamiento de lenguaje natural Elaboración propia

Define una función que utiliza el etiquetado gramatical de NLTK para verificar si una frase contiene palabras que pueden tener múltiples etiquetas gramaticales. Esta función podría ser útil para identificar frases en las que una palabra puede tener varios roles gramaticales posibles.

Figura 9

Ambigüedad de la palabra



Nota. La figura presenta un ejemplo visual que aborda la ambigüedad inherente a una palabra en el contexto del procesamiento de lenguaje natural. Elaboración propia

Muestra cómo utilizar la biblioteca NLTK y WordNet para determinar si una palabra tiene múltiples significados. Se implementa para identificar palabras en las que es importante considerar su contexto para comprender su significado exacto.

2.3.2 MÉTODOS PARA EL PROCESAMIENTO DEL LENGUAJE NATURAL

En el ámbito del procesamiento de lenguaje natural (NLP), se emplean diversas técnicas para desentrañar y comprender la riqueza del lenguaje humano contenido en textos. Una de las etapas

iniciales y fundamentales en este proceso es la tokenización. La tokenización implica segmentar un texto en unidades más pequeñas llamadas tokens, que pueden ser palabras individuales o incluso signos de puntuación, dependiendo de la estrategia adoptada. Este proceso nos proporciona las entidades básicas con las que posteriormente trabajaremos.

Sin embargo, la tokenización es solo el punto de partida. Luego entra en juego el etiquetado de las partes del discurso (PoS), que implica clasificar los tokens en categorías gramaticales como verbos, sustantivos, adjetivos, preposiciones, entre otros. Esta etiquetación es esencial para entender la estructura gramatical y el significado de las oraciones, para profundizar aún más en el análisis lingüístico, se emplea el análisis superficial o "shallow parsing", también conocido como "chunks". Aquí, se realiza un análisis más profundo de los tokens y, basándonos en sus etiquetas de partes del discurso, se construye una representación jerárquica de la estructura gramatical a través de árboles sintácticos.

Pero el procesamiento del lenguaje natural no se detiene en la gramática. También tiene en cuenta los aspectos pragmáticos del lenguaje, como la detección de ironía, sarcasmo e intencionalidad en el discurso. Esta comprensión pragmática, conocida como "Pragmatic Analysis", agrega un nivel más profundo de interpretación a las palabras y oraciones.Una vez que hemos desglosado y comprendido la estructura lingüística y pragmática, surge la necesidad de representar el lenguaje de una manera que las máquinas puedan manejar. Aquí es donde entran en juego conceptos como "Bag of Words". Esta técnica consiste en construir una matriz en la que cada token es una columna, y se registra cuántas veces aparece cada token en cada oración. Aunque pierde información de orden y contexto, esta representación es útil para ciertas tareas.

Finalmente, una técnica avanzada en NLP es "word2vec", que aborda el significado semántico de las palabras. A través del entrenamiento en grandes volúmenes de texto, esta técnica asigna vectores numéricos a las palabras de manera que palabras similares se representan en el espacio vectorial como puntos cercanos.

Esto brinda una riqueza de información semántica y contextual que puede ser utilizada para una comprensión más profunda del significado detrás de las palabras. En conjunto, estas técnicas permiten a las máquinas no solo manipular el lenguaje humano, sino también capturar su riqueza estructural y significado, allanando el camino para una amplia gama de aplicaciones en el procesamiento de lenguaje natural.

Tabla 4

Tokenización

Tokenización	**Descripción**	**Ejemplo**

Tokenización de palabras	Divide un texto en unidades individuales llamadas tokens, donde cada token representa una palabra.	import nltk
		from nltk.tokenize import word_tokenize, sent_tokenize
		# Tokenización de palabras
		texto = "¡Hola! Esto es un ejemplo de tokenización."
		tokens_palabras = word_tokenize(texto)
		print("Tokenización de palabras:")
		print(tokens_palabras)
		print()

Tokenización de oraciones	Divide un texto en oraciones individuales.	texto = "¡Hola! Esto es un ejemplo de tokenización. ¡Espero que te sea útil!" tokens_oraciones = sent_tokenize(texto) print("Tokenización de oraciones:") print(tokens_oraciones) print()
Tokenización de expresiones regulares	Utiliza patrones personalizados para dividir el texto en tokens.	from nltk.tokenize import RegexpTokenizer texto = "Hola a todos. ¡Bienvenidos a NLTK!" patron = r'\w+' tokenizer = RegexpTokenizer(patron) tokens_expresion_regular= tokenizer.tokenize(texto) print("Tokenización de expresiones regulares:") print(tokens_expresion_regular)

Nota. Esta tabla resume los diferentes tipos de tokenización y proporciona ejemplos para cada uno de ellos. Elaboración propia.

Tokenizar: "El proceso de Tokenización implica la separación de palabras en el texto en entidades conocidas como tokens. En esta etapa, es importante considerar si los signos de puntuación también se tratarán como tokens, si las mayúsculas tendrán relevancia y si se agruparán palabras similares en un solo token." (León, 2020)

Tagging Part of Speech (PoS): Para parametrizar el PoS se deben "Clasificar las oraciones en verbo, sustantivo, adjetivo preposición, etc." (Aprende Machine Learning, 2023).

Shallow parsing / Chunks: "Sirve para entender la gramática en las oraciones. Se hace un parseo de los tokens y a partir de su PoS se arma un árbol de la estructura." (León, 2020)

Pragmatic Analysis: "Detectar cómo se dicen las cosas: ironía, sarcasmo, intencionalidad, etc" (León, 2020)

Bag of Word: Es una manera de representar el vocabulario que utilizamos en nuestro modelo y consiste en crear una matriz en la que cada columna es un token y se contabilizará la cantidad de veces que aparece ese token en cada oración (representadas en cada fila). (León, 2020)

word2vec: Es una técnica que aprende de leer enormes cantidades de textos y memorizar qué palabras parecen ser similares en diversos contextos. Luego de entrenar suficientes datos, se generan vectores de 300 dimensiones para cada palabra conformando un nuevo vocabulario en donde las palabras "similares" se ubican cercanas unas de otras. Utilizando vectores pre-entrenados, logramos tener muchísima riqueza de información para comprender el significado semántico de los textos. (León, 2020)

2.3.3 CONCLUSIONES CAPITULO

La selección adecuada de herramientas y tecnologías desempeñan un papel fundamental en el éxito y eficacia de las tareas de procesamiento del lenguaje natural (PLN), como se aborda en este proyecto de grado. La elección del lenguaje de programación Python y la base de datos vectorial Pinecone ha sido respaldada por su versatilidad y facilidad de uso, especialmente en el contexto del PLN. Python ofrece una plataforma flexible para implementar algoritmos complejos y soluciones específicas para diversas tareas de PLN, desde análisis sintácticos hasta generación de lenguaje.

La rica comunidad de desarrollo en Python ha dado lugar a una amplia gama de bibliotecas y marcos de trabajo especializados en PLN, como NLTK, spaCy y Transformers. Estas bibliotecas preconstruidas aceleran la implementación y el análisis de tareas de PLN al proporcionar herramientas y modelos listos para usar. La capacidad de Python para integrarse con otras

tecnologías, como Pinecone en este caso, permite un almacenamiento eficiente y recuperación de información relevante para el PLN.

CAPÍTULO 3 - DISEÑO DE ARQUITECTURA – MODELO

3.1 INTRODUCCIÓN

En el paisaje actual de la tecnología, donde la inteligencia artificial y el procesamiento de lenguaje natural han alcanzado nuevas alturas, se presenta un proyecto de grado que tiene como objetivo transformar la manera en que accedemos a la información. Este proyecto se centra en la optimización de las búsquedas en bases de datos vectoriales a través de la innovadora plataforma Pinecone. Al aprovechar la combinación de modelos de embebimiento de texto de dimensiones compactas, como el "Modelo text-embedding-ada-002," junto con índices especializados, se aspira a lograr una eficiencia sin precedentes en la recuperación de datos similares.

El "Modelo text-embedding-ada-002" presenta un emocionante avance al reducir drásticamente las dimensiones de las incrustaciones, lo que se traduce en un uso más eficiente de recursos y una optimización del almacenamiento. Al integrar esta innovación con las capacidades de los índices en Pinecone, se busca crear un sistema que acelere las búsquedas y permita una recuperación de información más rápida y precisa en bases de datos de gran envergadura. Este proyecto se posiciona en la vanguardia de la búsqueda de soluciones que desafían los límites tradicionales en la recuperación de datos, promoviendo una mayor fluidez en la interacción entre humanos y máquinas.

A medida que los avances en el procesamiento de lenguaje natural continúan transformando diversos sectores, desde la atención al cliente hasta la educación y más allá, este proyecto tiene el potencial de impactar significativamente la forma en que las máquinas comprenden el lenguaje humano y se comunican en tiempo real. La combinación de modelos eficientes y técnicas de

búsqueda avanzadas abre nuevas puertas hacia una experiencia más fluida y natural en la interacción con sistemas basados en inteligencia artificial. En última instancia, este proyecto representa un paso audaz hacia la creación de sistemas más inteligentes y receptivos que se integren aún más en nuestras vidas cotidianas.

3.2 PINECONE

En Pinecone, el proceso de indexación implica tomar vectores de datos, asignarles identificadores únicos y almacenarlos de manera que se pueda acceder a ellos rápidamente durante las búsquedas.

Figura 10

Infraestructura Pinecone



Nota. La imagen representa la arquitectura de la base de datos vectorial. Pinecone. (2023). *Arquitectura de datos*. Recuperado de https://www.pinecone.io/

Los índices juegan un papel crucial en acelerar las búsquedas, ya que permiten una búsqueda rápida y eficiente de vectores similares en una gran base de datos.

El proceso general de cómo funcionan los índex en Pinecone es el siguiente:

Ingesta de datos: Los vectores de alta dimensión, que representan los datos que se desea indexar, se envían al servicio de Pinecone para su almacenamiento y procesamiento.

Vectores embebidos: Los vectores de datos se embeben en un espacio vectorial de alta dimensión utilizando modelos de aprendizaje automático previamente entrenados, como modelos de lenguaje o redes neuronales. Estos modelos transforman los datos en representaciones vectoriales densas, donde la similitud entre vectores se puede medir mediante métricas como la distancia euclidiana o el coseno del ángulo entre ellos.

Creación del índice: Los vectores embebidos se indexan en una estructura de datos especializada, como un índice de búsqueda en el vecino más cercano (k-NN) o un índice de hash distribuido. Estos índices se diseñan para permitir búsquedas rápidas y eficientes de vecinos cercanos, lo que facilita la recuperación de vectores similares a una consulta dada.

Búsqueda de vectores: Cuando se realiza una consulta de búsqueda, el vector de consulta se embebe de la misma manera que los vectores indexados y se utiliza para buscar en el índice los vectores más cercanos. El índice devuelve los vectores que son los vecinos más cercanos del vector de consulta, lo que representa los resultados relevantes de la búsqueda.

Recuperación de resultados: Los resultados de la búsqueda, es decir, los vectores similares más cercanos a la consulta se devuelven al usuario para su procesamiento o presentación.

3.3 ÍNDICES PINECONE

Dentro del contexto de Pinecone, cada índice desempeña un papel fundamental al organizar datos en forma de vectores. Los índices no solo almacenan estos vectores, sino que también están diseñados para manejar consultas y operaciones relacionadas con ellos. Esencialmente, un índice sirve como la unidad primaria de organización de datos vectoriales en Pinecone. Cada uno de estos índices se ejecuta en uno o más "pods", lo que contribuye a la eficiencia y velocidad en la manipulación de los datos dentro de la plataforma.

Pods, tipos de pods y tamaños de pods

Los pods son hardware listo para usar en Pinecone. Los índices operan en uno o varios. Más pods suelen significar mayor almacenamiento, menos latencia y mejor rendimiento. Diferentes tamaños de pods son posibles.

Después de hacer un índice en un tipo de casillero, no se puede cambiar. Pero se puede hacer uno nuevo con otro tipo si se requiere.

3.3.1 MÉTRICAS DE DISTANCIA

Puede elegir entre diferentes métricas al crear un índice vectorial:

Euclidean

- Esto se utiliza para calcular la distancia entre dos puntos de datos en un plano. Es una de las métricas de distancia más utilizadas.

- Cuando se usa, los resultados más similares son aquellos con la puntuación más baja. metric='euclidean'

- La distancia Euclidiana es ampliamente utilizada en aplicaciones donde se trabaja con espacios euclidianos y se necesita medir la similitud basada en la distancia geométrica entre puntos. Se usa comúnmente en aplicaciones de búsqueda de vecinos cercanos en espacios de alta dimensión, como en sistemas de recomendación, análisis de imágenes y más.

Cosine

- Esto se usa a menudo para encontrar similitudes entre diferentes documentos. La ventaja es que las puntuaciones están normalizadas en el rango [-1,1].

- La distancia coseno es especialmente útil cuando se trabaja con vectores que representan datos dispersos o en aplicaciones de procesamiento de lenguaje natural. Es frecuentemente utilizada en tareas de búsqueda de texto y recuperación de información, como la recuperación de documentos similares o la clasificación de texto.

Dotproduct

- Esto se usa para multiplicar dos vectores. Puedes usarlo para decirnos qué tan similares son los dos vectores. Cuanto más positiva sea la respuesta, más cerca están los dos vectores en términos de sus direcciones.

Tabla 5

Métricas de distancia pinecone

Métrica de Distancia	Rango de Valores	Descripción
Distancia Euclidiana	Cualquier número real no negativo	Mide la distancia geométrica entre dos puntos en un espacio euclidiano.
Distancia Coseno	Rango de -1 a 1	Mide la similitud entre dos vectores en función del ángulo que forman entre sí. Un valor cercano a 1 indica similitud, mientras que -1 indica similitud inversa.
Distancia de Jaccard	Rango de 0 a 1	Mide la similitud entre conjuntos. Se utiliza para comparar la similitud entre conjuntos de elementos. Un valor de 1 indica que los conjuntos son idénticos, mientras que 0 indica que no tienen elementos en común.
Distancia de Hamming	Rango de 0 a la dimensión del vector	Mide el número de posiciones en las que dos vectores difieren. Se utiliza para vectores binarios o discretos.

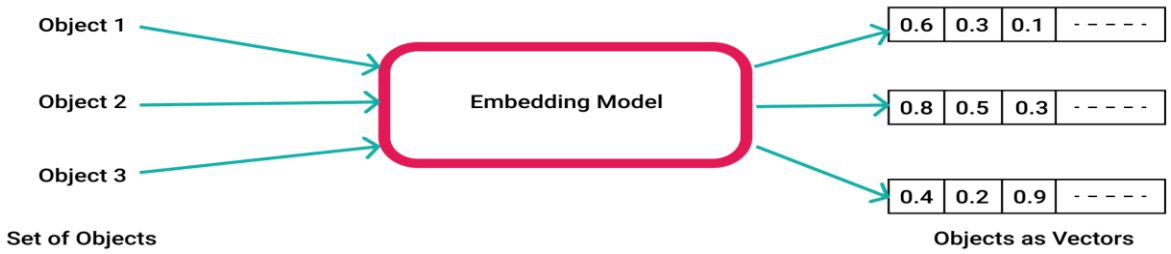
Nota. La tabla proporciona una visión panorámica de las diversas métricas utilizadas para cuantificar la distancia o similitud entre elementos en diferentes contextos. Elaboración propia

3.4 VECTOR EMBEDDINGS

"Los vectores embeddings son esenciales en el aprendizaje automático, usados en búsqueda, recomendación y PNL. Son clave en sistemas como recomendadores, asistentes de voz y traductores de idiomas." (Tripathi, 2023)

Figura 11

Ejemplo Embedding

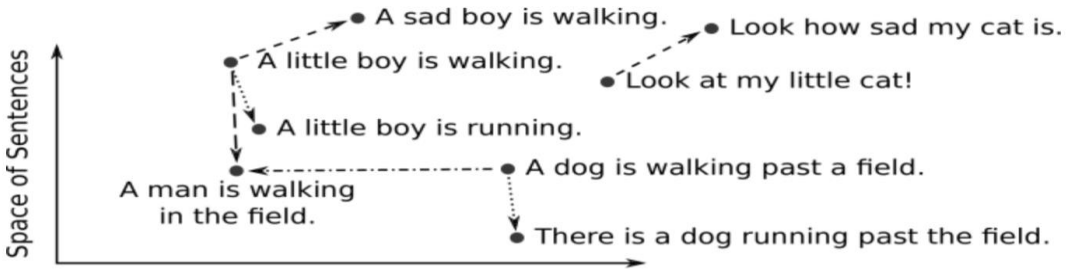


Nota. En esta imagen podemos observar cómo es el funcionamiento del modelo Embedding. Pinecoce. (2023). *Modelo Embeeding* Recuperado de pinecone.io/learn/vector-embeddings/

Los vectores poseen características singulares que los convierten en herramientas excepcionales. Facilitan la conversión de la similitud semántica que los seres humanos perciben en la proximidad dentro de un espacio vectorial, lo cual es un logro significativo.

Figura 12

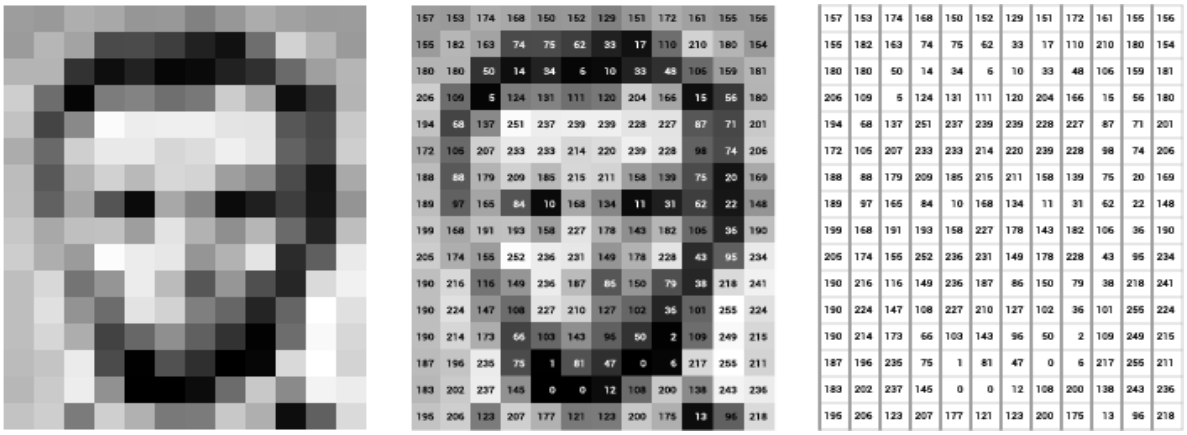
Ejemplo espacio vectorial



Nota. Podemos observar en la siguiente grafica como podemos recuperar frases en un espacio vectorial según su similitud. Pinecone. (2023). E*spacio Vectorial*. Recuperado de pinecone.io/learn/vector-embeddings/.

Figura 13

Ejemplo red neuronal convolucional



Nota. Aquí se puede observar que las imágenes sin procesar se representan como píxeles en escala de grises. Pinecone.(2023). *Pixeles grises.* Recuperado de pinecone.io/learn/vector-embeddings/.

Modelo text-embedding-ada-002: Dimensiones Compactas para Eficiencia Óptima

En este modelo, hemos reducido el tamaño de las incrustaciones al mínimo. Con solo 1536 dimensiones, estas nuevas incrustaciones representan una octava parte del tamaño de las incrustaciones de davinci-001. Este ajuste resulta en incrustaciones más rentables y eficientes, especialmente al trabajar con bases de datos vectoriales."

3.4.1 CREACIÓN DE ÍNDICE

A continuación, se presentará la manera de crear un índice (Base de Datos vectorial) en Pinecone utilizando Python, además de explorar la inserción de datos en el mismo.

Figura 14

Creación índice en pinecone



```
import pinecone
import random
pinecone.init(api_key="da2c5269-8077-4bac-b3af-719aef7dab3e", environment="us-west4-gcp-free")

#pinecone.create_index("quickstart", dimension=8, metric="euclidean")

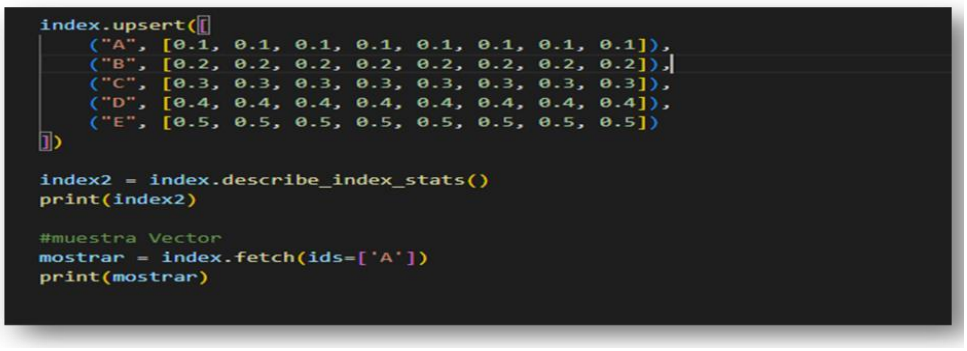
#Accede al indice
index = pinecone.Index("quickstart")
print(index)
```

Nota. Se puede observar una forma desde Python de crear nuestros índices en Pinecone. Elaboración propia.

En la siguiente imagen podemos visualizar una manera de insertar y adicional visualizar un vector del Index (Base de Datos)

Figura 15

Inserción datos pinecone



```
index.upsert([
    ("A", [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]),
    ("B", [0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2]),
    ("C", [0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3]),
    ("D", [0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4]),
    ("E", [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5])
])

index2 = index.describe_index_stats()
print(index2)

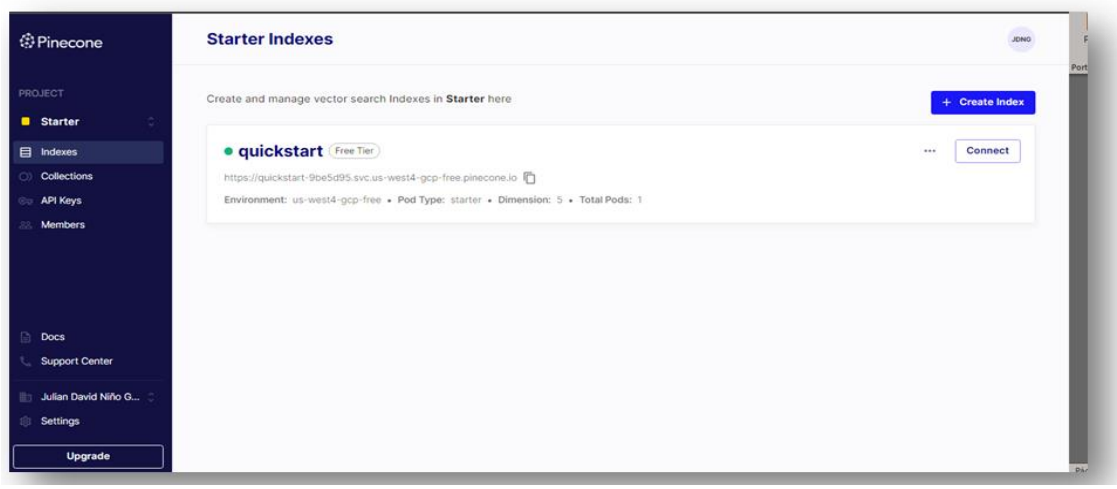
#muestra Vector
mostrar = index.fetch(ids=['A'])
print(mostrar)
```

Nota. Figura del código el cual se encarga de insertar datos a nuestra base de datos vectorial. Elaboración propia.

Obtenemos como resultado la siguiente base de datos en Pinecone, mejorando así la eficiencia y la capacidad de búsqueda.

Figura 16

Visualización índice



Nota. En la imagen se puede observar las características con las que cuenta el índice después de su creación. Elaboración propia.

3.5 CHATGPT - OPENIA

Con el auge de la inteligencia artificial (IA) y el procesamiento de lenguaje natural (PLN), surgieron varias herramientas y modelos que prometían cambiar la forma en que las máquinas comprenden y generan lenguaje humano. Una de estas herramientas que ha demostrado ser un cambio de juego en el campo es ChatGPT.

A diferencia de otros modelos de lenguaje, GPT y por extensión ChatGPT, son entrenados en dos fases: preentrenamiento y afinamiento. En la fase de preentrenamiento, el modelo aprende a

predecir la siguiente palabra en una oración utilizando grandes cantidades de texto. Luego, durante el afinamiento, se ajusta a tareas específicas utilizando conjuntos de datos más pequeños y orientados. (FERNÁNDEZ, 2023)

A diferencia de otros modelos de lenguaje, GPT y por extensión ChatGPT, son entrenados en dos fases: preentrenamiento y afinamiento. En la fase de preentrenamiento, el modelo aprende a predecir la siguiente palabra en una oración utilizando grandes cantidades de texto. Luego, durante el afinamiento, se ajusta a tareas específicas utilizando conjuntos de datos más pequeños y orientados.

Interacción Humano-Máquina: ChatGPT ha llevado la interacción entre humanos y máquinas a un nivel completamente nuevo, permitiendo conversaciones fluidas y naturales. Esto ha abierto la puerta a aplicaciones en soporte al cliente, asistentes virtuales, y tutoriales educativos, por nombrar algunos.

Flexibilidad: A diferencia de los chatbots basados en reglas, ChatGPT puede manejar una amplia gama de temas y adaptarse a las desviaciones en las conversaciones sin requerir una programación específica para cada escenario.

Avance en PLN: ChatGPT representa un hito en el procesamiento de lenguaje natural, demostrando que las máquinas pueden generar lenguaje que es indistinguible del humano en muchas circunstancias. Este nivel de competencia era inimaginable hace solo unos años.

Democratización de la IA: Con herramientas como ChatGPT, el poder de la IA avanzada es ahora accesible para desarrolladores y empresas sin la necesidad de contar con recursos masivos o equipos de investigación especializados.

ChatGPT, y la serie GPT en general, simbolizan una revolución en la forma en que las máquinas comprenden y generan lenguaje humano. A medida que estas tecnologías continúan avanzando, es probable que veamos una integración aún más profunda de máquinas inteligentes en nuestra vida diaria, desdibujando las líneas entre la comunicación humana y la máquina.

3.6 SEGURIDAD - JSON WEB TOKEN

En el mundo de la ciberseguridad y el desarrollo de aplicaciones web, garantizar la seguridad de la información y mantener la integridad de las comunicaciones entre un cliente y un servidor es fundamental. Una herramienta que ha emergido como una solución robusta.

JSON Web Token (JWT).

"JSON Web Token es un estándar para transmitir información de forma segura en internet, por medio de archivos en formato JSON, que es un tipo de archivo de texto plano con el cual se pueden crear parámetros y asignarles un valor." (keepcoding, 2023) Estos tokens están diseñados para ser compactos, URL-seguros y pueden ser utilizados para identificar a los usuarios entre aplicaciones, para transferir información entre partes y para asegurar contenido.

Un JWT tiene tres partes:

- Encabezado (Header)
- Carga útil (Payload)
- Firma (Signature)

El encabezado suele consistir en dos partes: el tipo de token, que es JWT, y el algoritmo de firma, como HMAC SHA256 o RSA. La carga útil contiene las afirmaciones, que son declaraciones

acerca del sujeto (por ejemplo, el usuario). Por último, la firma es lo que valida el token y garantiza que el contenido no haya sido alterado de ninguna manera durante su transmisión.

Autenticación y autorización: Uno de los usos más comunes de JWT es para autenticación. Una vez que el usuario inicia sesión, se genera un JWT y se devuelve al usuario, quien puede enviarlo con su solicitud subsiguiente para acceder a rutas, servicios y recursos permitidos. Este token puede decirle al servidor todo lo que necesita saber sobre si un usuario está autorizado o no, eliminando la necesidad de buscar en la base de datos en cada solicitud.

Simplicidad y eficiencia: Debido a su diseño compacto, los JWT son ligeros y fáciles de transmitir en URL, encabezados de HTTP y entre diferentes sistemas.

Descentralización: Los JWT no requieren de un almacenamiento centralizado, lo que los hace ideales para aplicaciones escalables. No necesitas almacenar tokens JWT en una base de datos, lo que reduce la sobrecarga y los riesgos asociados.

Seguridad: La firma digital en JWT asegura que el token no haya sido alterado durante la transmisión. Los JWT también pueden ser cifrados, brindando confidencialidad a la información.

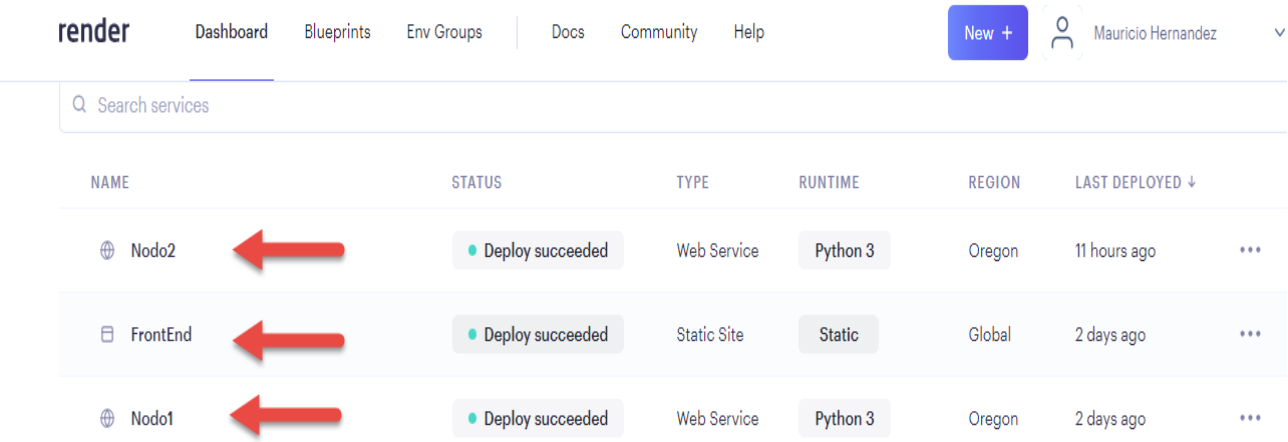
En conclusión, el JWT ha transformado la manera en que las aplicaciones web manejan la autenticación y la autorización. Su diseño compacto y seguro lo hace ideal para su uso en aplicaciones modernas, tanto en dispositivos de escritorio como móviles. Sin embargo, es esencial comprender y seguir las mejores prácticas para garantizar su seguridad, como no incluir información sensible en la carga útil, utilizar algoritmos de firma fuerte y gestionar adecuadamente la expiración de tokens.

3.7 BALANCEO DE CARGA

Dentro de la implementación de balanceo de carga, se tuvo en cuenta que los componentes están divididos en dos plataformas así:

Figura 17

Plataforma servidor RENDER

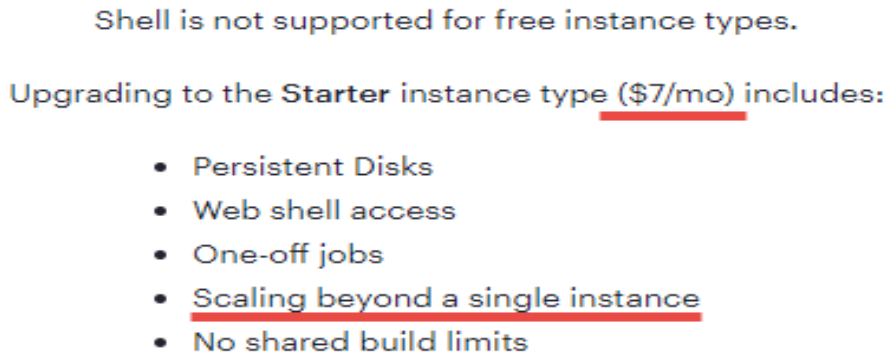


Nota. Se puede observar los nodos montados en la plataforma Render. Elaboración propia

Para activar el balanceo de carga la plataforma está lista, simplemente se requiere activar el plan premium por un valor mensual de $ 7 USD mensuales:

Figura 18

Activación de balanceo



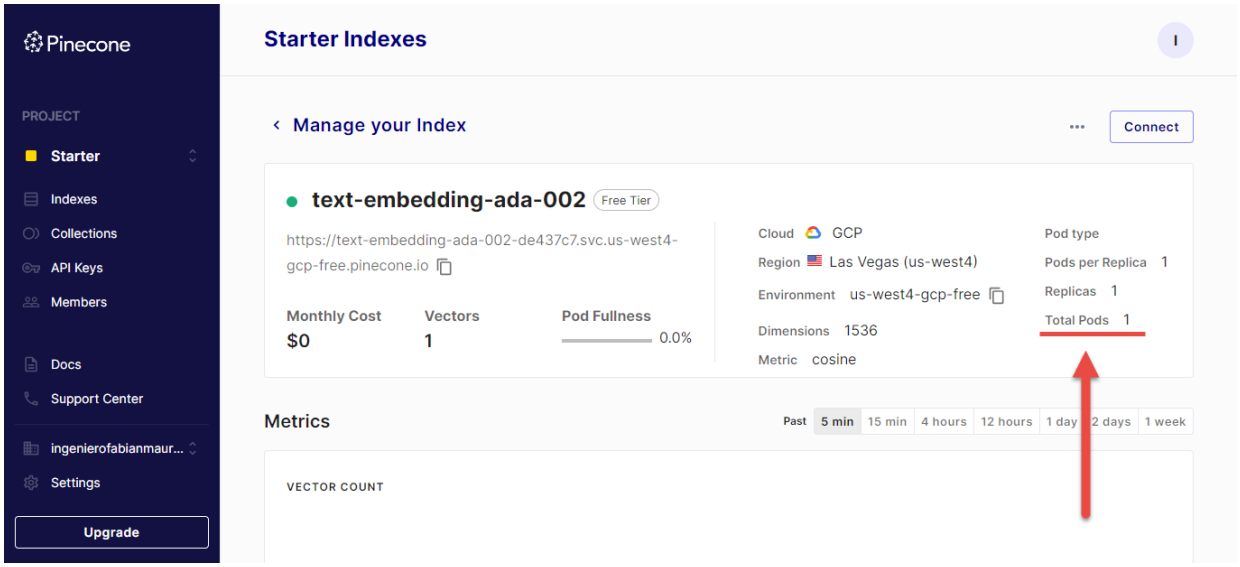
Nota. Para la activar el balanceo de cargas se debe realizar un pago por el monto indicado, en la figura se muestra el valor y lo que incluye. Elaboración propia

3.8. SERVIDOR DE BASE DE DATOS VECTORIALES PINECONE:

A continuación, te presento una representación gráfica que ilustra la estructura de la base de datos vectorial junto con todos sus componentes. Esta visualización está diseñada para proporcionar una comprensión más completa y clara del conjunto de datos en cuestión.

Figura 19

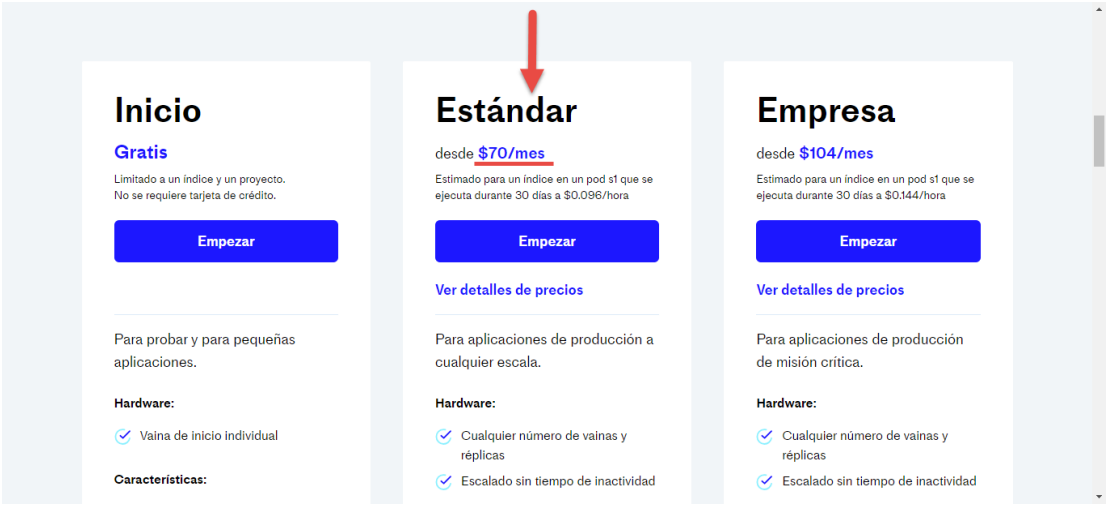
Servidor de BD vectoriales



Nota. Imagen que muestra los vectores empleados y sus respectivas dimensiones en la base de datos vectorial Pinecone. Elaboración propia.

Figura 19

Plan estándar



Nota. Para implementar la base de datos Pinecone está también cuenta con distintos planes los cuales se muestran en la imagen. Elaboración propia.

En total para las dos plataformas mensualmente se requiere una inversión mensual adicional de $ 77 USD. Adicionalmente se realizaron pruebas y se contempló la posibilidad de implementación balanceo de carga local con HA – Proxy. HAProxy, que significa "High Availability Proxy", es un balanceador de carga y un servidor proxy para aplicaciones TCP y HTTP. Se utiliza en muchos entornos donde la disponibilidad y el rendimiento son cruciales. Pero ¿por qué es tan relevante la implementación de HAProxy en infraestructuras modernas? En este ensayo, abordaremos las principales razones por las que HAProxy se ha vuelto esencial.

Mejora de la Disponibilidad:

Una de las principales ventajas de HAProxy es su habilidad para mejorar la disponibilidad de un servicio. A través de su función de balanceo de carga, puede distribuir tráfico entrante entre múltiples servidores, asegurando que, si uno falla, los demás puedan continuar manejando las solicitudes. Esto reduce el tiempo de inactividad y asegura que los usuarios finales no experimenten interrupciones en el servicio.

Escalabilidad:

Con el aumento de usuarios y tráfico, las aplicaciones y servicios requieren más recursos. HAProxy permite a las organizaciones escalar horizontalmente, añadiendo más servidores al grupo según la necesidad. De este modo, en lugar de invertir en un servidor más grande (escalabilidad vertical), se pueden añadir más servidores pequeños, lo que suele ser más coste-eficiente.

Distribución Eficiente del Tráfico:

HAProxy es capaz de distribuir el tráfico basándose en diversos algoritmos como round robin, menor cantidad de conexiones, o incluso basado en la IP del cliente. Estos algoritmos aseguran que el tráfico se distribuya de la forma óptima posible, maximizando el uso de recursos y evitando cuellos de botella.

Flexibilidad y Personalización:

HAProxy permite una gran flexibilidad en su configuración. Se pueden definir reglas específicas para dirigir el tráfico basado en diversos criterios, como la URL, el tipo de solicitud o las cookies. Esta personalización facilita la implementación de soluciones específicas para cada escenario.

Mejora en la Seguridad:

HAProxy puede actuar como un punto de terminación SSL, descargando la carga de la gestión de cifrado de los servidores de backend. También ofrece protecciones contra ciertos ataques, como el DDoS, y puede integrarse con soluciones adicionales de seguridad para ofrecer una protección más robusta.

Monitorización y Análisis:

Con HAProxy, los administradores pueden tener una visión detallada del tráfico y el comportamiento de los servidores. Proporciona métricas en tiempo real, lo que facilita la detección de problemas y su posterior análisis.

Para finalizar, en un mundo donde la disponibilidad, el rendimiento y la seguridad son esenciales para el éxito de los servicios en línea, herramientas como HAProxy se vuelven imprescindibles. Su capacidad para balancear carga, escalar según demanda, personalizar el tráfico y proteger los

recursos, lo convierten en un componente vital en cualquier arquitectura moderna que busque brindar un servicio excepcional a sus usuarios. Por tanto, la implementación de HAProxy no es solo una opción, sino una inversión estratégica en la infraestructura tecnológica de una organización.

Adicionalmente se generaron pruebas exhaustivas instalando una máquina virtual en Virtual Box:

Figura 20

Máquina virtual

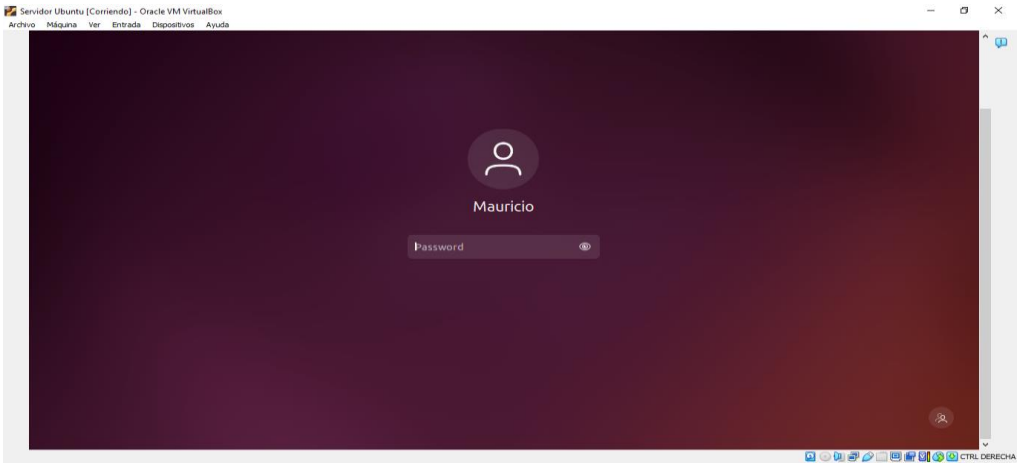


Nota. Imagen de la maquina y virtual utilizada y su respectiva versión. Elaboración propia.

Dentro de Virtual Box se instaló un Ubuntu Desktop donde se instaló HAProxy:

Figura 21

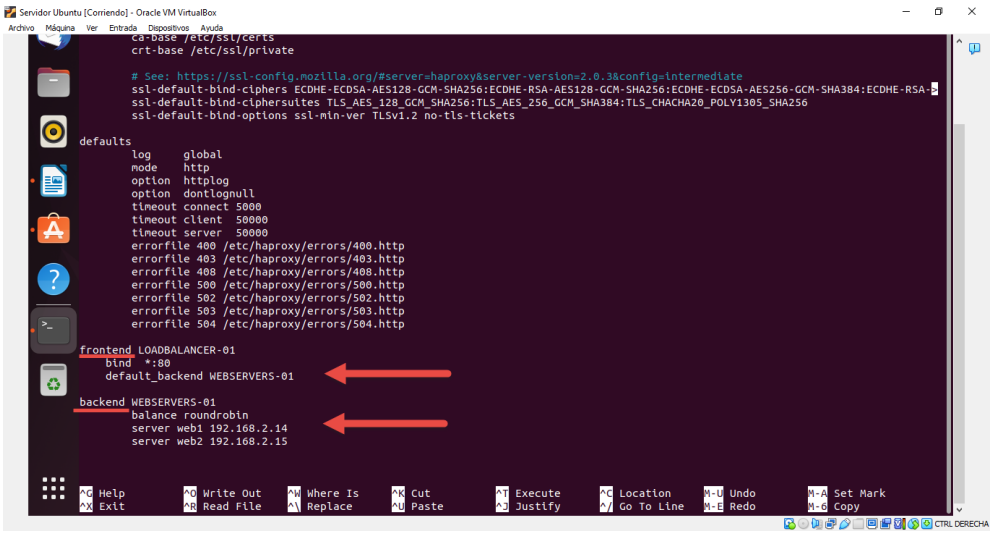
Ubuntu Desktop



Nota. Figura del sistema operativo utilizado en su inicio. Elaboración propia.

Figura 22

Pruebas HA Proxy



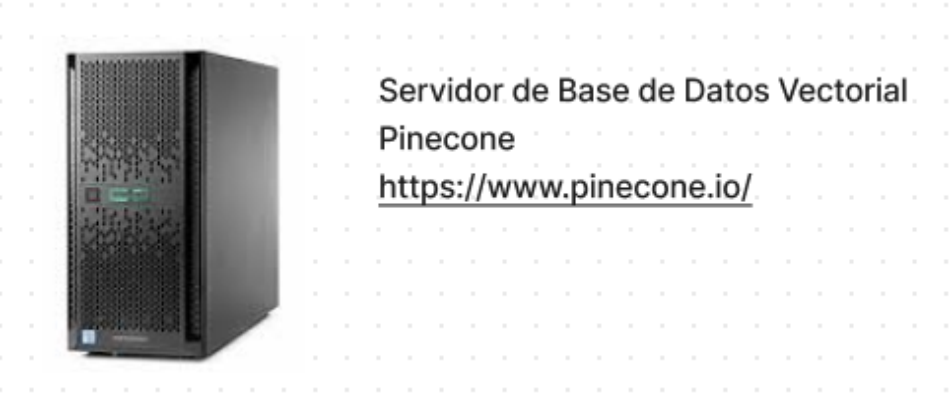
Nota. HAProxy es un servidor proxy de nivel de aplicación y un equilibrador de carga. Elaboración propia.

3.9 COMPONENTES DEL MÉTODO

A continuación, se enumeran todos y cada uno de los elementos que forman parte del método identificado para la detección de ambigüedades en los requerimientos de software mediante la aplicación de técnicas de aprendizaje automático y bases de conocimiento

Figura 23

Servidor BD vectorial



Nota. Imagen descriptiva del servidor BD Vectorial. Elaboración propia

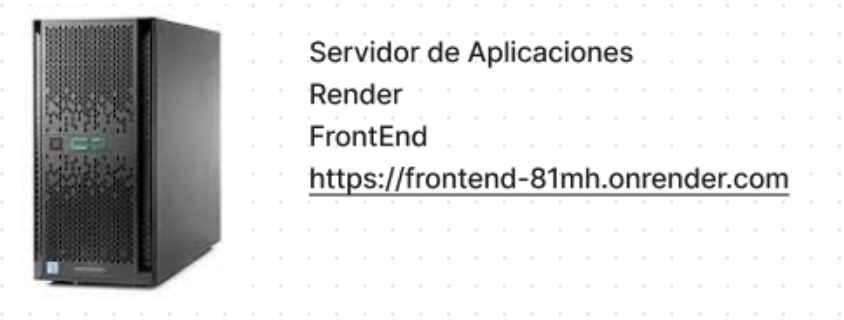
Pinecone es una base de datos vectorial gestionada que proporciona búsqueda de vectores (o "búsqueda de similitud") para desarrolladores con una API sencilla y precios basados en el uso. Una base de datos vectorial es un tipo de base de datos que está optimizada para almacenar y buscar datos vectoriales. Los datos vectoriales son datos que pueden representarse como una matriz de números, por ejemplo, los datos que se utilizan en el aprendizaje automático. La búsqueda de vectores es una técnica para buscar datos vectoriales que son similares a un vector

dado. Esto se puede utilizar para una variedad de tareas, como la búsqueda de imágenes similares, la búsqueda de texto similar y la búsqueda de productos similares.

Pinecone utiliza una combinación de búsqueda aproximada y exacta para recuperar información de manera eficiente. La búsqueda aproximada utiliza algoritmos como ANN (Nearest Neighbor) para encontrar los datos vectoriales que son más similares a un vector dado. La búsqueda exacta utiliza algoritmos como LSH (Locality Sensitive Hashing) para encontrar los datos vectoriales que son exactamente iguales a un vector dado.

Figura 24

Servidor aplicaciones Render



Nota. Imagen descriptiva del servidor Aplicaciones Render. Elaboración propia.

Render.com es una plataforma de alojamiento en la nube unificada que permite a los desarrolladores crear y ejecutar aplicaciones y sitios web de forma rápida y sencilla. Ofrece una amplia gama de características, incluyendo:

Instancias de servidor escalables: Render.com permite a los desarrolladores escalar sus aplicaciones y sitios web vertical y horizontalmente según sea necesario.

Almacenamiento y bases de datos: Render.com ofrece almacenamiento y bases de datos en la nube para que los desarrolladores puedan centrarse en el desarrollo de sus aplicaciones y no en la gestión de la infraestructura.

CDN global: Render.com ofrece una red de distribución de contenido global para garantizar que las aplicaciones y los sitios web sean accesibles rápidamente desde cualquier parte del mundo.

Protección DDoS: Render.com ofrece protección DDoS para proteger las aplicaciones y los sitios web de los ataques cibernéticos.

Redes privadas: Render.com ofrece redes privadas para permitir a los desarrolladores conectar sus aplicaciones y sitios web de forma segura.

Despliegue automático: Render.com permite a los desarrolladores desplegar sus aplicaciones y sitios web de forma automática desde Git.

Render.com está dirigido a desarrolladores de todos los niveles de experiencia. Es una buena opción para desarrolladores que quieren crear y ejecutar aplicaciones y sitios web de forma rápida y sencilla, sin tener que preocuparse por la gestión de la infraestructura.

En una instancia de Render se colocó la interfaz desarrollada en HTML, allí se encuentra ubicada la interfaz que interactúa entre el usuario final y el método construido para la identificación de ambigüedades.

Figura 25

Servidor Render Nodo1



Nota. Imagen descriptiva de servidor (Nodo1) Elaboración propia.

Figura 26

Servidor Render Nodo2



Nota. Imagen Descriptiva del servidor Render (Nodo2) Elaboración propia

Nodo1 es un backend en Flask Python que se puede utilizar para crear aplicaciones web escalables y eficientes. Está diseñado para ser fácil de usar y de configurar, y nos ofrece una amplia gama de funciones para satisfacer las necesidades de una variedad de aplicaciones.

Nodo1 utiliza una arquitectura modular que facilita la adición de nuevas características y la extensión de la funcionalidad existente. Está construido sobre una base sólida de código probado y probado, y se mantiene activamente por un equipo de desarrolladores experimentados. Es la mejor opción para crear aplicaciones web de alta calidad sin tener que empezar desde cero. Está diseñado para ser escalable y eficiente, y ofrece una amplia gama de funciones para satisfacer las necesidades de una variedad de aplicaciones.

Características principales:

Fácil de usar y de configurar: Nodo1 es fácil de instalar y configurar. Se puede instalar en cualquier servidor que ejecute Python, y no requiere ninguna configuración especial.

Amplitud de funciones: Nodo1 ofrece una amplia gama de funciones para satisfacer las necesidades de una variedad de aplicaciones. Estas funciones incluyen:

- Gestión de usuarios
- Gestión de contenido
- Gestión de API
- Caching
- Seguridad

Escalabilidad y eficiencia: Nodo1 está diseñado para ser escalable y eficiente. Se puede escalar vertical y horizontalmente según sea necesario, y utiliza técnicas de optimización para mejorar el rendimiento.

Mantenimiento activo: Nodo1 está mantenido activamente por un equipo de desarrolladores experimentados. Se actualiza regularmente con nuevas funciones y correcciones de errores.

3.10 DIAGRAMA GENERAL DEL MÉTODO

Figura 27

Modelo detención de ambigüedades



Nota. Ilustración del modelo para la detención de ambigüedades, detalladamente mostrando cada una de sus partes. Elaboración propia.

Para el método destinado a la detección de ambigüedades se utilizaron varios componentes. A continuación:

- Dos nodos del BackEnd en Python que fue construido en lenguaje Python haciendo uso del framework Flask:

  El sistema del BackEnd desarrollado en Python se estructura en dos nodos distintos, ambos implementados utilizando el marco de trabajo Flask. Cada uno de estos nodos constituye la infraestructura subyacente para el despliegue del componente BackEnd en el sistema. En el interior de estos nodos, se alojan dos servicios fundamentales que desempeñan roles específicos. El primero de ellos se encarga de la gestión de tokens JWT (JSON Web Tokens), proporcionando un mecanismo seguro para la autenticación y autorización de usuarios. Por otro lado, el segundo servicio cumple una función esencial al abordar la resolución de ambigüedades presentes en el lenguaje natural, lo que demuestra una capacidad avanzada de procesamiento de lenguaje natural (NLP). En conjunto, estos dos nodos y sus respectivos servicios forman una base robusta para el funcionamiento eficiente y seguro del BackEnd, permitiendo una comunicación fluida y segura entre el sistema y los usuarios, así como una comprensión precisa de las instrucciones expresadas en lenguaje natural.

  La base de conocimiento se encuentra alojada en Pinecone, una base de datos vectorial gestionada que destaca por su capacidad para realizar búsquedas de similitud en vectores. En este contexto, la implementación de la base de datos involucra un vector específico

denominado "Los aportes únicamente son para pagar de SALUD, PENSION, EPS". Pinecone simplifica la búsqueda de vectores al proporcionar una API amigable y estructurar sus precios basados en el uso.

En esencia, una base de datos vectorial es optimizada para almacenar y recuperar datos vectoriales, que son representaciones numéricas, como las empleadas en el aprendizaje automático. La búsqueda de vectores se refiere a la técnica que localiza datos vectoriales similares a un vector dado, abarcando aplicaciones como la búsqueda de imágenes, texto y productos similares. Pinecone optimiza la búsqueda al combinar métodos aproximados y exactos: emplea algoritmos como ANN (Nearest Neighbor) para similitudes aproximadas y LSH (Locality Sensitive Hashing) para coincidencias exactas.

Pinecone ofrece múltiples ventajas respecto a otras bases de datos vectoriales al ser accesible, escalable y económica. Facilita la recuperación, agrupación y análisis de datos. Entre los casos de uso habituales de Pinecone se encuentran la búsqueda de imágenes similares, texto y productos, permitiendo su aplicación en diversas tareas. En resumen, Pinecone se presenta como una herramienta poderosa y versátil para operar con datos vectoriales, con aplicaciones que abarcan desde la búsqueda de información hasta la comparación de productos electrónicos.

Para el procesamiento del lenguaje natural se implentó CHATGP del Proyecto de Interacción Entre Agentes Orgánicos y Artificiales (OPEIA, por sus siglas en inglés) que se enmarca en la sinergia entre la inteligencia humana y la artificial, explorando la colaboración entre agentes biológicos y sistemas de procesamiento del lenguaje natural. En este emocionante proyecto, se emplea la versión 3.5 de ChatGPT como intérprete de lenguaje natural, habilitando conversaciones coherentes y contextualmente relevantes entre

humanos y sistemas. Un aspecto destacado de esta iniciativa es la utilización de la técnica de aprendizaje por transferencia, en la que el modelo es previamente entrenado en una amplia variedad de datos lingüísticos. Además, en este contexto, la elección de la arquitectura de vectores de embeddings Ada se ha revelado como un enfoque efectivo para representar y entender la semántica y la estructura de las interacciones lingüísticas, mejorando la capacidad de OPEIA para comprender y generar respuestas coherentes y contextualmente precisas.

- Para el balanceo de carga en el sistema distribuido en cuatro componentes permite que en cada uno de ellos haya un sistema de balanceo de carga:
    - Base de datos vectorial
    - Nodos Render
    - Nodo FrontEnd

CAPÍTULO 4 - IMPLEMENTACIÓN

4.1 INTRODUCCIÓN

Un modelo de infraestructura distribuida es una forma de diseñar sistemas informáticos que se enfoca en tener diferentes partes del sistema en lugares diferentes en lugar de tener todo en un solo lugar. Esto ayuda a que los sistemas sean más estables, ya que, si algo falla en un lugar, no afecta todo el sistema. Además, permite que los sistemas sean más rápidos y puedan manejar más cosas al mismo tiempo. Aunque puede ser un poco más complicado de manejar, es útil para aplicaciones en línea y servicios que necesitan estar siempre disponibles y responder rápidamente.

4.2 MODELO DE INFRAESTRUCTURA DISTRIBUIDA

4.2.1 DESCRIPCIÓN RENDER

A continuación, se presenta una concisa tabla que describe las características del render, considerando su funcionalidad.

Tabla 6.

Descripción Render

Característica	Descripción
Creación y Ejecución de Aplicaciones	Plataforma para desarrollar y alojar aplicaciones web, sitios estáticos, API y más.
Certificados TLS Gratuitos	Ofrece certificados SSL/TLS gratuitos para asegurar conexiones seguras y encriptadas.

CDN Global	Red de entrega de contenido global para acelerar la carga de contenido y mejorar la experiencia.
Protección DDoS	Protege contra ataques de denegación de servicio distribuido para garantizar la disponibilidad.
Redes Privadas	Permite crear redes privadas virtuales (VPC) para aislar y proteger recursos de manera efectiva.
Implementaciones Automáticas desde Git	Se integra con repositorios de Git para implementar automáticamente cambios en el código.
Compatibilidad con Diversos Lenguajes	Admite múltiples lenguajes de programación y marcos, como Python, Node.js, Ruby, Java, etc.

Nota. La tabla presenta una síntesis de las características clave de la plataforma Render, que ofrece una gama diversificada de herramientas para el desarrollo y alojamiento de aplicaciones web y recursos digitales. Elaboración propia.

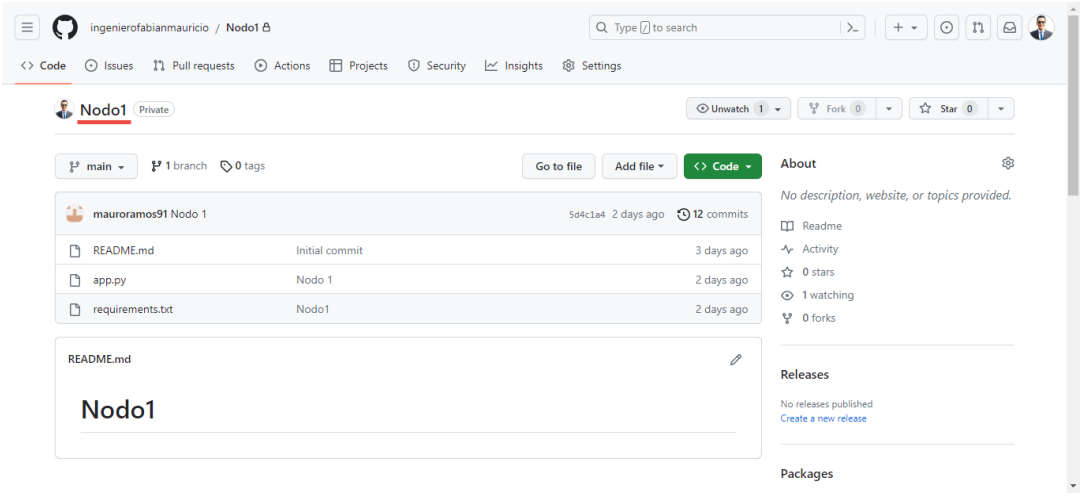
4.3 REPOSITORIOS DE SOFTWARE

A continuación, se encuentran las imágenes de cada uno de los repositorios de software. Estas imágenes ofrecen una representación visual de los contenidos y estructuras de los respectivos repositorios. Es esencial para los desarrolladores y usuarios tener una visión clara de lo que cada repositorio contiene, para facilitar la elección del software adecuado para sus necesidades. Estas imágenes también sirven como una rápida referencia visual para aquellos que buscan entender la organización y jerarquía de los diferentes proyectos alojados en dichos repositorios:

Repositorio de software en lenguaje de programación Python donde reside el Nodo 1 de la malla de servidores distribuidos para BackEnd:

Figura 29

Repositorio GitHub nodo 1



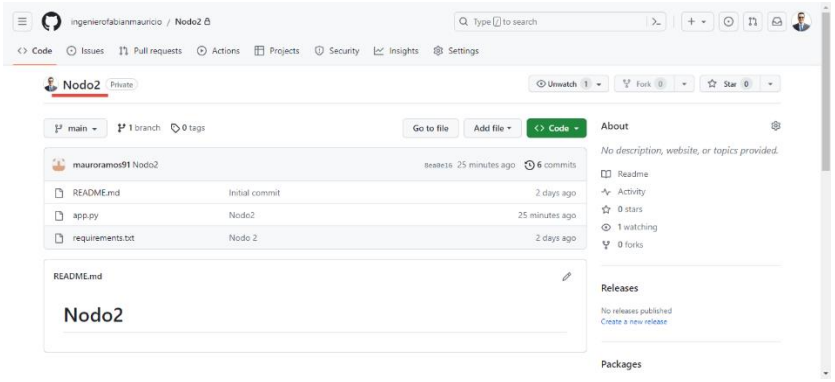
Nota. Imagen del repositorio en GitHub, donde se encuentra alojado la información del Nodo1. Elaboración propia

Repositorio de software en lenguaje de programación Python donde reside el Nodo 2 de la malla

de servidores distribuidos para BackEnd:

Figura 28

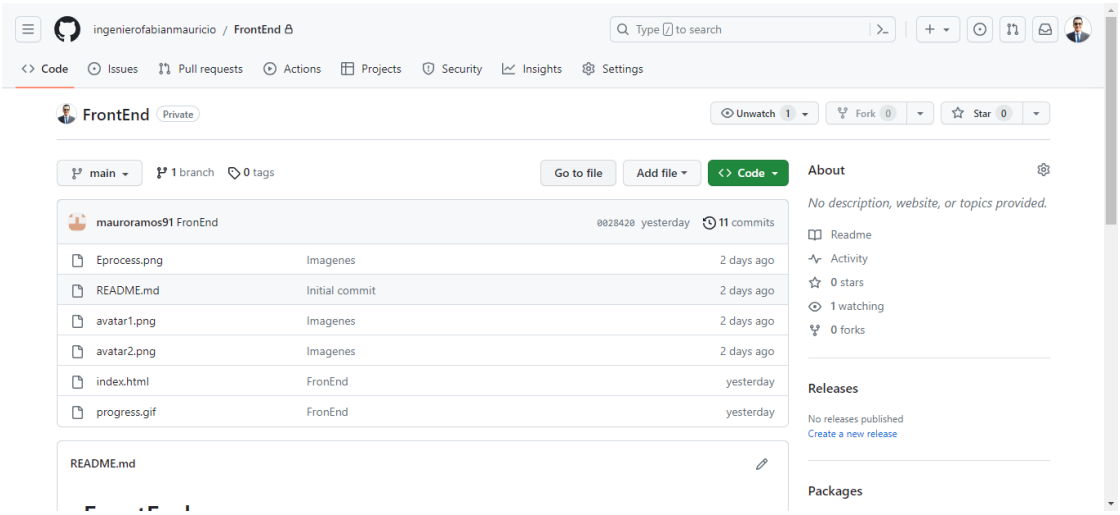
Repositorio GitHub nodo 2



Nota. Imagen del repositorio en GitHub, donde se encuentra alojado la información del Nodo2. Elaboración propia.

Repositorio que contiene la herramienta tecnológica correspondiente a la interfaz de usuario utilizada para la interacción entre el usuario funcional y él método encontrado para la resolución de ambigüedades: **Figura 29**

Repositorio GitHub frontEnd



Nota. Imagen del repositorio en GitHub, donde se encuentra alojado la información del FrontEnd. Elaboración propia.

4.4 IMPLEMENTACIÓN DE INTERFAZ GENERAL

Para la implementación de interfaces dentro de la solución actual, se contemplaron dos prototipos de interfaz general:

- Interfaz de interacción entre usuario funcional y método desarrollado.
- Interfaz para la visualización de servicio BackEnd API Rest

La interfaz de interacción propuesta para la interacción entre usuario funcional y el método desarrollado está basada en un diseño web adaptable (RWD) que cuenta con un enfoque de diseño y desarrollo que hace que las páginas web se adapten a diferentes tamaños de pantalla y dispositivos. El RWD se basa en el uso de rejillas flexibles, imágenes escalables y consultas de medios CSS. Estas técnicas permiten que el contenido de una página web se redimensione, se reorganice o se oculte según el tamaño de la pantalla en la que se esté visualizando.

Por otro lado, para la interacción con los servicios Rest se implementó Insomnia Rest que cuenta con una plataforma de desarrollo de API colaborativa y de código abierto que facilita la construcción de API de alta calidad. Es una herramienta fácil de usar que ofrece una interfaz intuitiva, automatización integrada y una amplia gama de funciones para diseñar, depurar y probar API.

Insomnia Rest ofrece una serie de ventajas sobre otras herramientas de desarrollo de API, entre ellas:

Facilidad de uso: Insomnia Rest es una herramienta muy intuitiva y fácil de aprender a usar. Incluso los desarrolladores principiantes pueden empezar a usarla rápidamente.

Automatización: Insomnia Rest ofrece una serie de funciones de automatización que pueden ahorrar tiempo a los desarrolladores. Por ejemplo, puede automatizar pruebas de API, generar código y exportar datos.

Funcionalidad: Insomnia Rest ofrece una amplia gama de funciones para diseñar, depurar y probar API. Estas funciones incluyen la posibilidad de crear solicitudes y respuestas personalizadas, definir parámetros, trabajar con archivos, y más.

Colaboración: Insomnia Rest es una herramienta colaborativa que permite a los desarrolladores trabajar juntos en los mismos proyectos de API. Esto puede ayudar a mejorar la productividad y la calidad de las API.

En un mundo cada vez más digitalizado, donde el acceso a internet a través de dispositivos móviles es predominante, el RWD es esencial para garantizar una experiencia de usuario coherente con alta calidad en cualquier dispositivo e Insomnia Rest es una herramienta potente y versátil que puede ayudar a los desarrolladores a crear API de alta calidad de forma rápida y sencilla.

4.5 INTERFAZ GRÁFICA RESPONSIVE WEB DESIGN – RWD:

A continuación, se puede ver la implementación de cada una de las pantallas:

Versión Móvil:

Figura 30

Versión Móvil

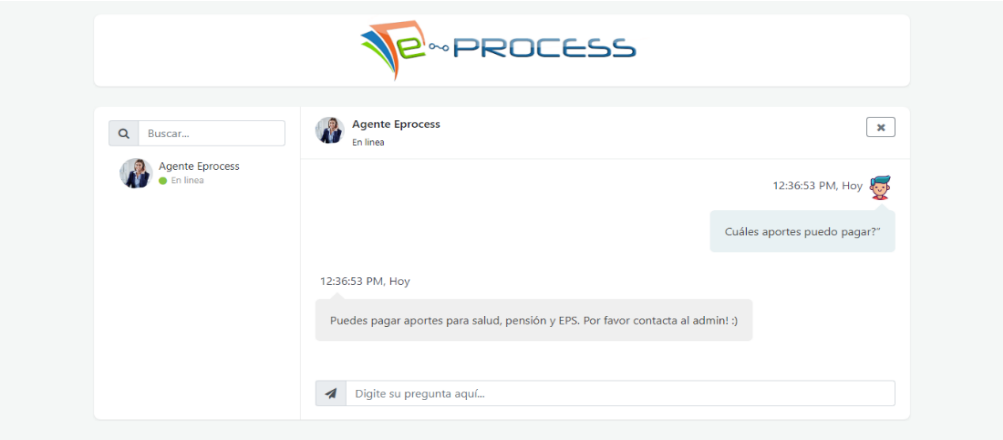


Nota. Imagen donde se evidencia la interfaz de la aplicación desde un móvil celular. Elaboración propia.

Versión de PC:

Figura 31

Versión de PC



Nota. Imagen donde se evidencia la interfaz de la aplicación desde un computador de escritorio. Elaboración propia.

4.6. PRUEBAS FUNCIONALES

4.6.1 ESCENARIO PARA PRUEBAS FUNCIONALES

A continuación, se describen las pruebas funcionales, dichas pruebas fueron agrupadas en tres secciones:

- prueba procesamiento de lenguaje natural.
- prueba de servicios de backend distribuidos.
- prueba de autenticación jwt – json web token.
- prueba procesamiento de lenguaje natural.

Para el proceso de pruebas funcionales desde el equipo funcional de la compañía Eprocess S.A.S se propuso el siguiente requerimiento:

La base de conocimiento debe tener la siguiente información:

"Los aportes únicamente son para pagar de SALUD, PENSION, EPS"

A partir de la información existente en la base de conocimiento, la herramienta tecnológica debe permitir desde un dispositivo móvil y desde un equipo de cómputo PC, realizar las siguientes preguntas en lenguaje natural:

- ***"Cuáles aportes puedo pagar?"***
- ***"Puedo pagar mi celular a través de los aportes?"***

Y por último se espera que la herramienta tecnológica proporcione el siguiente resultado a cada una de las pruebas.

- *"Cuáles aportes puedo pagar?"*

Respuesta: Si, se pueden pagar los aportes de SALUD, PENSION Y EPS.

Ó alguna respuesta similar.

- *"Puedo pagar mi celular a través de los aportes?"*

Respuesta: **No.**

Ó alguna respuesta similar.

Por otro lado, las pruebas funcionales se consideran satisfactorias si el resultado a las dos preguntas planteadas es el esperado ó similar. Lo cual indica que el método desarrollado cumple a cabalidad

con la resolución de ambigüedades ya que por si sola la palabra aportes es polisémica: o que significa que tiene múltiples significados o interpretaciones dependiendo del contexto en el que se utilice. A continuación, algunos de sus posibles usos y significados:

Contribuciones económicas: En el contexto financiero o laboral, "aportes" se refiere a las contribuciones monetarias que se realizan a un fondo, como por ejemplo los aportes a la seguridad social o a un plan de pensiones.

Contribuciones en conocimiento o información: En un contexto académico o de investigación, "aportes" puede referirse a las contribuciones intelectuales, ideas o hallazgos que alguien presenta a una discusión o proyecto.

Incorporación de elementos: En un contexto más general, puede referirse a la adición o incorporación de algo a un conjunto, como cuando se dice "los aportes de vitaminas y minerales de un alimento".

4.6.2 PRUEBA DE SERVICIOS DE BACKEND DISTRIBUIDOS

Dentro de las pruebas funcionales, se requiere que los servicios de BackEnd estén distribuidos en dos nodos basados en la nube cloud

4.6.3 PRUEBA DE AUTENTICACIÓN JWT – JSON WEB TOKEN

Por último, para finalizar las pruebas funcionales se requiere que los servicios de BackEnd estén distribuidos en dos nodos basados en la nube cloud y que cada servicio solicite autenticación a través de JWT – Json Web Token

4.6.4 PRUEBA DE PROCESAMIENTO DE LENGUAJE NATURAL

Teniendo en cuenta lo anterior, se insertó un registro en la base de datos Pinecone, ese registro tiene como Metadata la siguiente frase:

"Los aportes únicamente son para pagar de SALUD, PENSION, EPS"

Posteriormente se hizo uso de la interfaz desarrollada para la interacción entre el usuario funcional y el método desarrollado, como se evidencia en la siguiente imagen:

El promt o la pregunta enviada a través del chat fue:

"Cuáles aportes puedo pagar?"

A lo que la herramienta tecnológica desarrollada respondió:

"Puedes pagar aportes pasa salud, pensión y EPS, Por favor contacta al admin :)"

A continuación, se presentan los pantallazos / imágenes que corresponden a los resultados de las pruebas funcionales en un dispositivo móvil y en un equipo de cómputo PC:

Por último y para finalizar las pruebas funcionales, se procedió a enviar el promt o la pregunta:

"*Puedo pagar mi celular a través de los aportes****"***

A lo que la herramienta tecnológica desarrollada respondió:

"No, los aportes solo pueden ser utilizados para pagar gastos relacionados con salud, pensión y EPS. ¡Por favor contacta al admin! :) "

A continuación, se relacionan los pantallazos / imágenes correspondientes al resultado de las pruebas funcionales en un dispositivo móvil y en un equipo de cómputo PC:

4.6.5 PRUEBA DE SERVICIOS DE BACKEND DISTRIBUIDOS

Para la ejecución de la prueba se hizo uso de la herramienta Insomia - https://insomnia.rest/, Insomnia es una aplicación de escritorio multiplataforma gratuita que simplifica la interacción y el diseño de API basadas en HTTP. Insomnia combina una interfaz fácil de usar con funciones avanzadas como asistentes de autenticación, generación de código y variables de entorno.

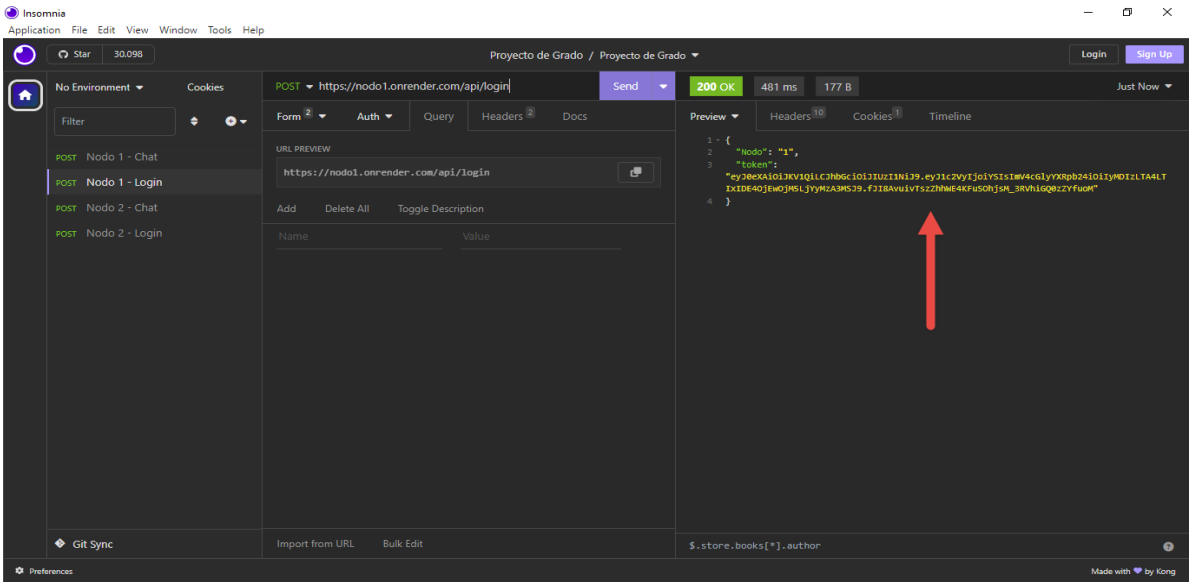
A continuación, se relacionan los pantallazos / imágenes correspondientes al resultado de las pruebas funcionales del Nodo 1:

Url Generación de Token Nodo 1: https://nodo1.onrender.com/api/login

Url Chat Conversacional Nodo 1: https://nodo1.onrender.com/api/pregunta

Figura 32

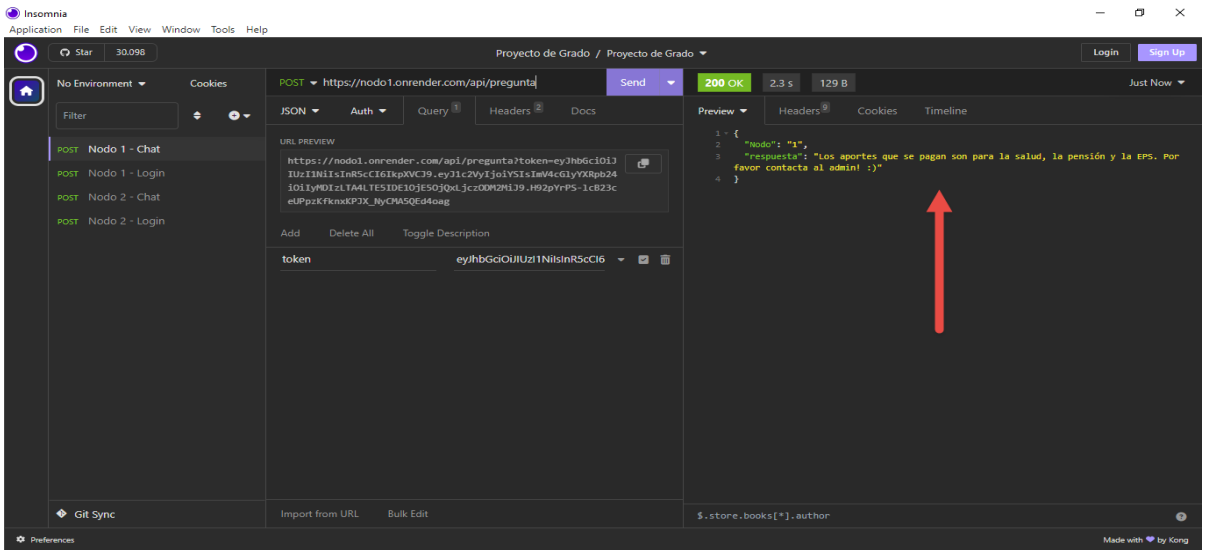
Generación token nodo 1



Nota. En esta imagen se puede evidenciar como se genera el token de acceso para el nodo 1. Elaboración propia.

Figura 33

Chat conversación nodo 1



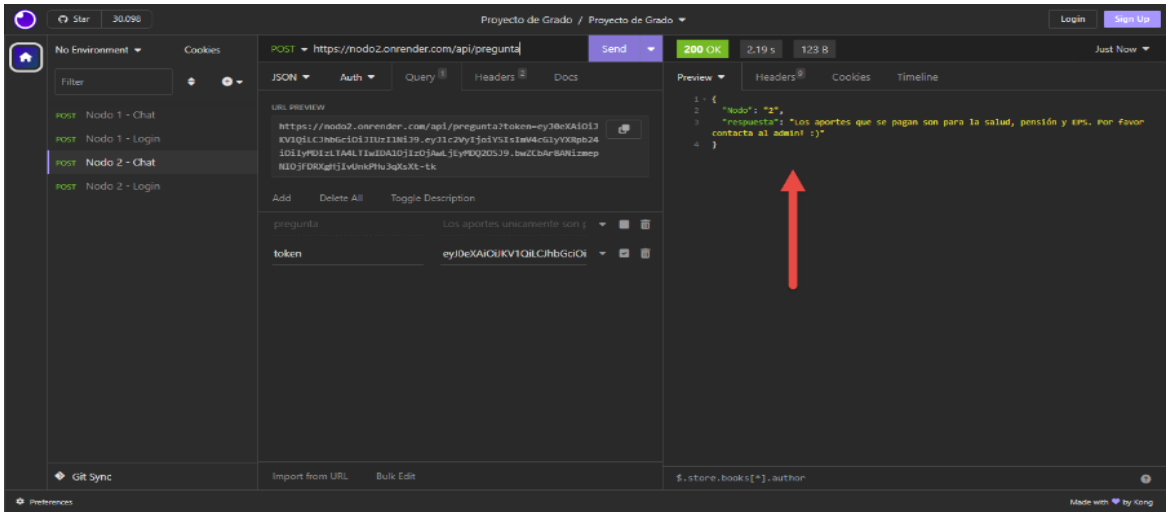
Nota. Podemos ver en esta imagen como se realiza la conversación con él chat teniendo en cuenta el Nodo1. Elaboración propia.

Url Generación de Token Nodo 2: https://nodo2.onrender.com/api/login

Url Chat Conversacional Nodo 2: https://nodo2.onrender.com/api/pregunta

Figura 34

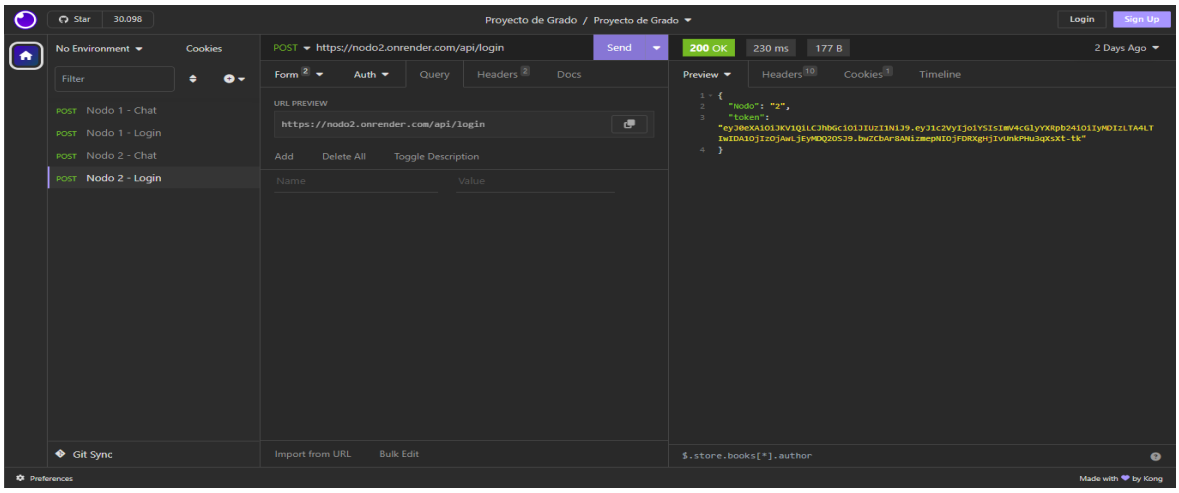
Chat conversación nodo 2



Nota. Podemos ver en esta imagen como se realiza la conversación con él chat teniendo en cuenta el Nodo1. Elaboración propia.

Figura 35

Generación token nodo 2

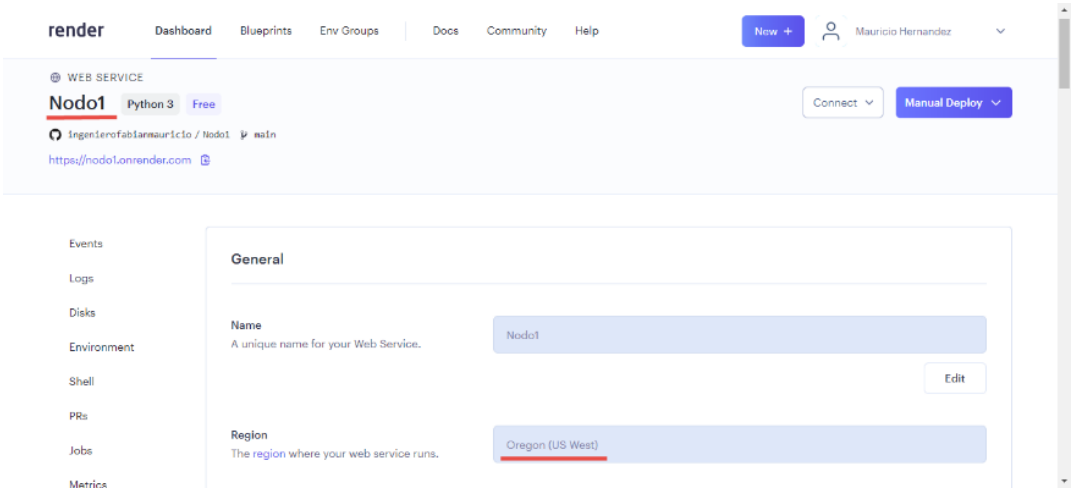


Nota. En esta imagen se puede evidenciar como se genera el token de acceso para el nodo 2. Elaboración propia

A continuación, en las siguientes imágenes se puede observar la ubicación y url general de cada uno de los nodos:

Figura 36

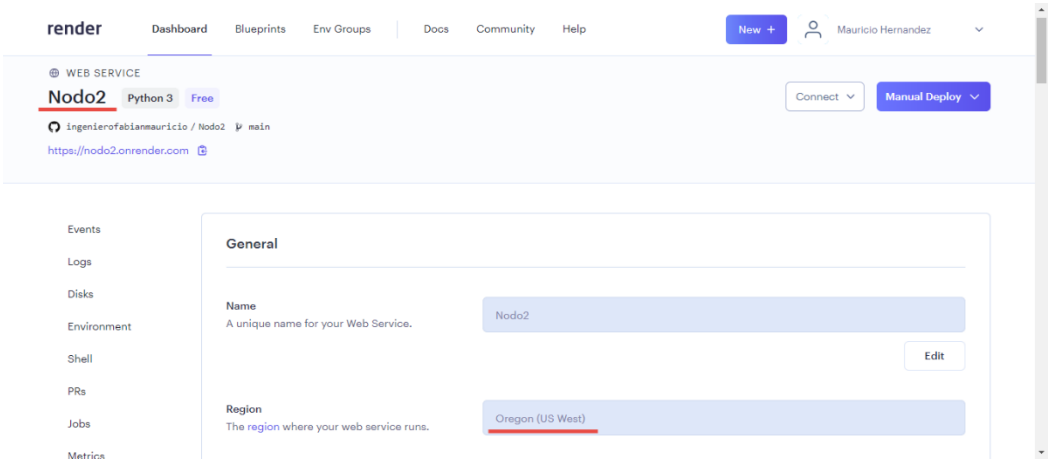
URL nodo 1



Nota. En la figura se presenta la disposición y la URL general del nodo1, la cual facilita la identificación y navegación eficiente. Elaboración propia.

Figura 37

URL Nodo 2



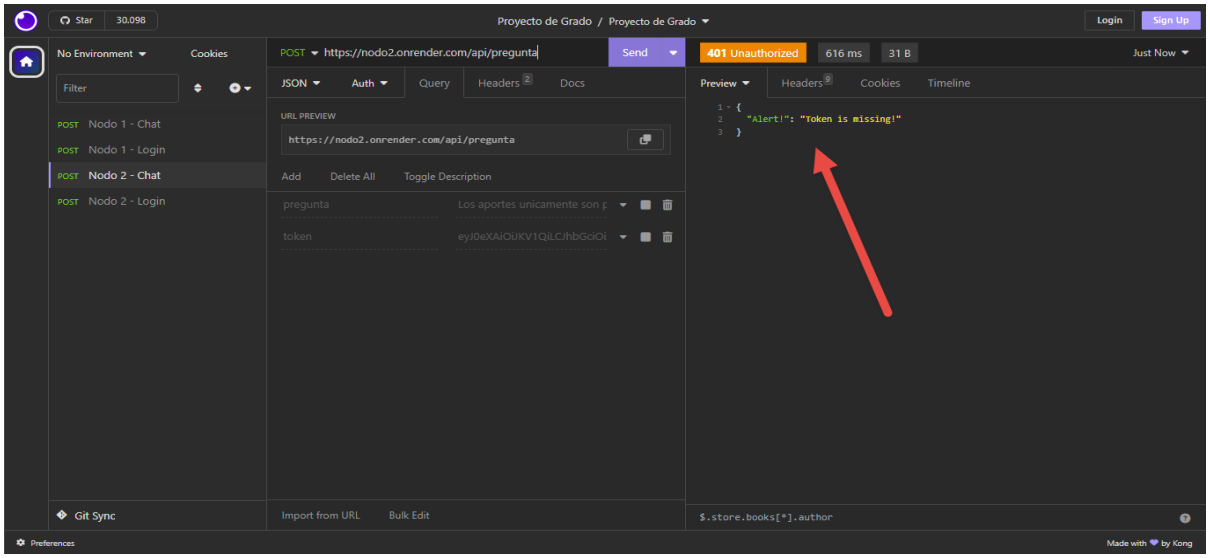
Nota. En la figura se presenta la disposición y la URL general del nodo2, la cual facilita la identificación y navegación eficiente. Elaboración propia. Elaboración propia

4.6.6 PRUEBA DE AUTENTICACIÓN JWT – JSON WEB TOKEN

En la siguiente imagen es visible que la herramienta desarrollada requiere el token JWT – Json Web Token como:

Figura 38

Requerimiento el token JWT.

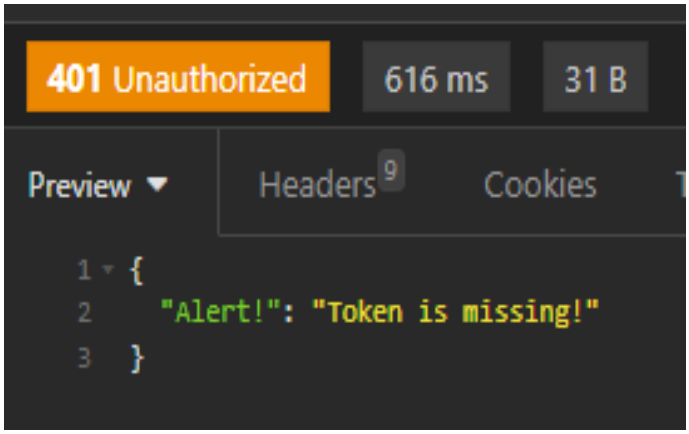


Nota. En la imagen se ilustra el requerimiento del token JWT (JSON Web Token) como parte del proceso de autenticación en una aplicación web. Elaboración propia

El sistema dará la siguiente respuesta para cada petición que no tenga el token dentro de su cabecera Http:

Figura 39

Respuesta petición



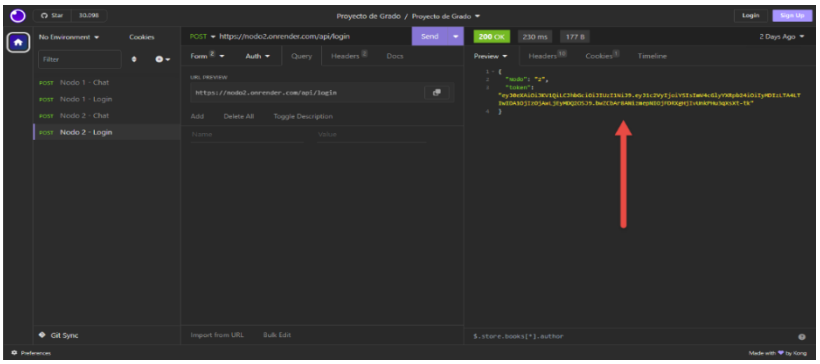
Nota. La imagen visualiza la "Respuesta a la Petición" como un elemento clave en el proceso de interacción en línea. Elaboración propia

A continuación, en las siguientes imágenes se puede observar la generación del token JWT – Json

Web Token en cada uno de los nodos existentes: Nodo1 y Nodo2:

Figura 40

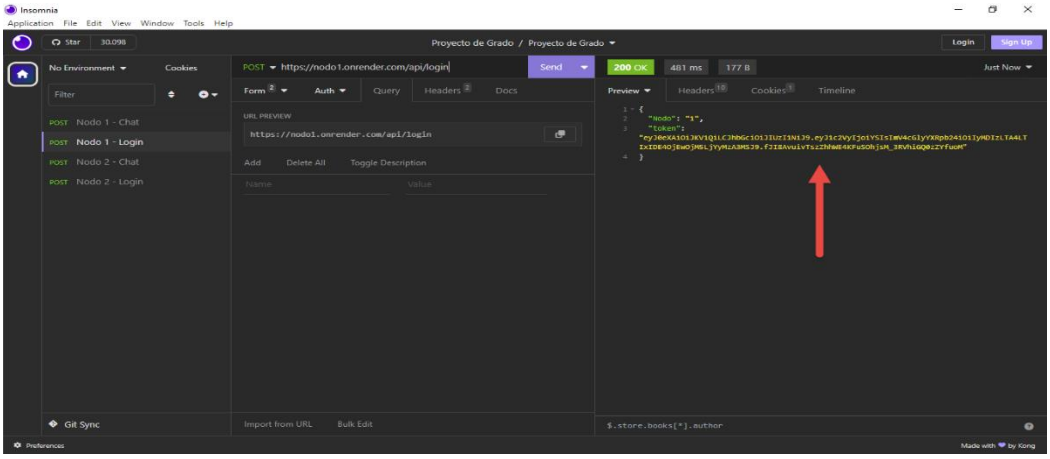
Generación token nodo 1



Nota. En la ilustración se visualiza cómo crear un token único y seguro en el primer nodo del flujo de autenticación. Elaboración propia

Figura 41

Generación token nodo 2



Nota. En la ilustración se visualiza cómo crear un token único y seguro en el segundo nodo del flujo de autenticación. Elaboración propia Elaboración propia

Para finalizar, el resultado de las pruebas funcionales fue satisfactorio ya que las respuestas indicadas por la herramienta tecnológica desarrollada fueron las esperadas y cumplen con el requerimiento y tienen en cuenta el insumo de información suministrado a la base de conocimiento *"Los aportes únicamente son para pagar de SALUD, PENSION, EPS"* existente en la base de datos vectorial Pinecone. Así mismo se evidenció que los nodos para el alanceamiento de carga funcionan correctamente y cada uno tiene la implementación de seguridad JWT – Json Web Token para servicios WEB ó EndPoints. Por lo tanto, se concluye que las pruebas funcionales fueron satisfactorias.

CONCLUSIONES

Se desarrolló una aplicación Web distribuida donde se implementó una base de conocimiento para la detección de ambigüedades en la especificación de los requerimientos de software dentro del desarrollo de cualquier herramienta tecnológica.

Se construyó satisfactoriamente un método que incluye algoritmos y herramientas de Inteligencia Artificial para la resolución de ambigüedades en la especificación de requerimientos de software.

Se desarrolló una interfaz distribuida, intuitiva y eficiente para la interacción entre el usuario final y la herramienta tecnológica construida donde se implementó el método para la resolución de ambigüedades en la especificación de requerimientos de software.

Se implementó el método para la resolución de ambigüedades en la especificación de requerimientos de software dentro de tecnologías que cuentan con aseguramiento JWT - Json Web Token y servidores redundantes para el balanceo de carga.

RECOMENDACIONES

El desarrollo de herramientas tecnológicas destinadas al mejoramiento y optimización del proceso de identificación de requerimientos de software ha sido siempre crucial ya que la tecnología tiene un avance exponencial o inclusive cuántico que sobrepasa las herramientas existentes para tal fin.

La correcta identificación y especificación de estos requerimientos garantiza que el producto final cumpla con las expectativas del cliente, que no hay sobrecostos, se reduzcan los errores y retrasos en el desarrollo del software. Con base en las conclusiones obtenidas dentro de este proyecto, se proponen las siguientes recomendaciones:

INNOVACIÓN EN EXPERIENCIA DEL USUARIO (USER EXPERIENCE - UX).

Teniendo en cuenta que dentro del proyecto se realizó la creación de una interfaz para la interacción centre el usuario y el método encontrado para la detección de ambigüedades en los requerimientos de software, dicha interfaz se destaca por ser intuitiva y eficiente, sin embargo, es recomendable que se realice un proceso de mejora continúa haciendo uso de los conceptos de Experiencia de usuario - UX e interfaz de Usuario - UI. El objetivo es hacer interfaces más amigables para el usuario. Una interfaz intuitiva facilita el proceso de aprendizaje y adaptación por parte del usuario, permitiéndole centrarse en la tarea principal: la revisión y especificación de requerimientos.

ADOPCIÓN DE INTELIGENCIA ARTIFICIAL (IA):

El empleo de algoritmos y herramientas de IA, como se evidencia en el proyecto, ofrece un enfoque proactivo para la identificación y resolución de los diferentes tipos de ambigüedades. Es

recomendable estar actualizando constantemente la herramienta construida para estar al día y a la par con las nuevas innovaciones del segmento de mercado de la Inteligencia artificial, así lograr implementar las últimas mejoras de cada herramienta con el objetivo de mejorar la precisión y eficiencia en la detección de ambigüedades.

ALIMENTACIÓN DE BASES DE CONOCIMIENTO:

La implementación de una base de conocimiento, que actúa como un repositorio de información sobre ambigüedades y resoluciones previas, puede actuar como un recurso invaluable para desarrolladores y analistas. Es esencial mantener, actualizar y ampliar regularmente estas bases de conocimiento para asegurar su relevancia y efectividad en la resolución de ambigüedades.

En conclusión, la correcta detección y resolución de ambigüedades en los requerimientos de software es fundamental para el éxito de cualquier proyecto tecnológico. Adoptando técnicas avanzadas, como la Inteligencia Artificial, y garantizando la seguridad y eficiencia de la herramienta, se asegura una base sólida para el desarrollo de software que cumple con las expectativas y demandas del mundo actual.

BIBLIOGRAFÍA

ReQtest. (2020). ReQtest – A Complete Guide. Recuperado de https://reqtest.com/what-is-reqtest/.

IBM. (2021). *IBM Rational DOORS*. Recuperado de https://www.ibm.com/products/rational-doors.

Jama Software. (2021). *Jama Connect*. Recuperado de https://www.jamasoftware.com/.

Micro Focus. (2021). *CaliberRM*. Recuperado de https://www.opentext.com/. Sparx Systems. (2021). *Enterprise Architect*. Recuperado de https://sparxsystems.com/products/ea/index.html

Pauzi, Z., & Capiluppi, A. (2023). *Applications of natural language processing in software traceability: A systematic mapping study*. Recuperado de https://www.sciencedirect.com/science/article/pii/S0164121223000110?via%3Dihub

Siahaan, D., Raharjana, I. K., & Fatichah, C. (2023). *User Story Extraction from Natural Language for Requirements Elicitation: Identify Software-Related Information from Online News*. Recuperado de https://www.sciencedirect.com/science/article/abs/pii/S0950584923000496?via%3Dihub.

Riaz, M. Q., Butt, W. H., & Rehman, S. (2019). *Automatic Detection of Ambiguous Software Requirements: An Insight*. Recuperado de https://ieeexplore.ieee.org/document/8714682

Rajender Kumar Surana, C. S., Shriya, Gupta, D. B., & Shankar, S. P. (2019). *Intelligent Chatbot for Requirements Elicitation and Classification*. Recuperado de https://ieeexplore.ieee.org/document/9016907.

Liu, K., Reddivari, S., & Reddivari, K. (2022). *Artificial Intelligence in Software Requirements Engineering: State-of-the-Art*. Recuperado de https://ieeexplore.ieee.org/document/9874229

Elsevier. (2023). *A review on risk assessment techniques for building services*. Recuperado de https://linkinghub.elsevier.com/retrieve/pii/S0950584923000496.

ScienceDirect. (s.f.). (2023) *Explore scientific, technical, and medical research on ScienceDirect. Recuperado de* https://www.sciencedirect.com/

Health Data Miner. (2019). CRISP-DM: *Una metodología para minería de datos en salud. Recuperado el 19 de marzo de 2023*, Recuperado de https://healthdataminer.com/data-mining/crisp-dm-una-metodologia-para-mineria-de-datos-en-salud/

Biblioteca Hugging Face Transformers. (2023) *Huggingface* Recuperado de https://github.com/huggingface/transformers

Pinecone (2023) *Búsqueda de Similitud Vectorial a Escala*. Recuperado de https://www.pinecone.io/

Gensim. (2023). *Documentación vía GitHub* Recuperado de https://radimrehurek.com/gensim/auto_examples/index.html#core-tutorials.

Pinecone. (2023). *Choosing index type and size* Recuperado de https://docs.pinecone.io/docs/choosing-index-type-and-size

Hugging Face (2023). *Transformers* Recuperado de https://huggingface.co/docs/transformers/index.

Abstracta Chile. (2023). *Herramienta para pruebas de API: Review de Insomnia*. Recuperado de https://cl.abstracta.us/blog/insomnia-herramienta-pruebas-api/.

Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT). RFC 7519*. Recuperado de https://tools.ietf.org/html/rfc7519

Auth0. (n.d.). *JSON Web Token Introduction*. Recuperado de https://auth0.com/learn/json-web-tokens/

Leon,E. (2020). *Procesamiento del lenguaje natural con Python*. Recuperado de https://www.baoss.es/procesamiento-del-lenguaje-natural-pln-con-python/

Fernandez. (2023) *ChatGPT: Que es, cómo usarlo y qué puedes hacer con este chat de Inteligencia Artificial* Recuperado de: https://www.xataka.com/basics/chatgpt-que-como-usarlo-que-puedes-hacer-este-chat-inteligencia-artificial.

Tripathi, R. (2023). *What are vector embeddings.* Recuperado de www.pinecone.io/learn/vector-embeddings/

Pinecone. *(2023). Arquitectura de datos.* Recuperado de https://www.pinecone.io/

Pinecoce. (2023). *Modelo Embeeding*. Recuperado de www.pinecone.io/learn/vector-embeddings/

Rivas, A. (2023) *Normas APA: La guía definitiva para presentar trabajos escritos. Guía Normas APA. Recuperado de https://normasapa.in/*

ANEXOS

ANEXO 1 - FACTIBILIDAD ECONÓMICA

El proyecto de grado para la Ingeniería en Telemática "Aplicación distribuida para la detección de ambigüedades en los requerimientos de software utilizando técnicas de Machine Learning y bases de conocimiento." es completamente viable, la factibilidad económica del proyecto es alta, ya que lo que se requiere en términos financieros son:

- Dos equipos de cómputo.
- Asesorías de los tutores del proyecto
- Acceso a Internet

En las tablas que se presentarán a continuación se describe la factibilidad económica, identificando los costos de hardware, software y recursos humanos necesarios para la realización del proyecto de pasantía que se propone.

ANEXO 2 - FACTIBILIDAD ECONÓMICA RECURSOS HUMANOS

Tipo	Descripción	Valor-Hora	Cantidad	Total
Tutor 1	Asesorías para la realización del proyecto, referente a la metodología.	$ 100.000	200	$ 20.000.000

Desarrolladores	Dos (2) programadores que realicen la implementación de la solución.	$ 60.000	720 Horas. 360 Horas por estudiante 15 horas semanales x seis meses	$ 43.200.000
Asesor Eprocess	Dos (2) asesores de Eprocess para la realización del proyecto, referente a la metodología.	$100.000	300	$ 30.000.000
Total Recursos Humanos				$ 93.200.000

Tabla 1- Factibilidad económica recursos humanos

A continuación, en la Tabla 2 se presentarán los gastos de los recursos que se ostentan en el desarrollo del proyecto.

Recurso	Descripción	Valor Unitario	Cantidad	Total

Computadores (Servidor - clientes)	Equipos portátiles para el desarrollo y las pruebas del sistema.	$ 6.000.000	2	$ 12.000.000
Total Recursos Técnicos				$ 12.000.000

Tabla 2. Factibilidad económica recursos técnicos

A continuación, en la tabla 3 se relacionan los gastos totales del proyecto

Recurso	Valor
Total Recursos Humanos	$ 93.200.000
Total Recursos Técnicos	$ 12.100.000
Total Otros recursos	$ 0
Costos imprevistos (10%)	$ 10.530.000
TOTAL COSTO	$ 115.830.000

Tabla 3. Factibilidad económica costo total.

ANEXO 3. MANUAL DE USUARIO

Bienvenido a la aplicación distribuida para la detección de ambigüedades en los requerimientos de software utilizando técnicas de machine learning y bases de conocimiento. Esta aplicación está diseñada para ayudar a los desarrolladores de software a identificar y mitigar las ambigüedades en los requerimientos de sus proyectos.

Las ambigüedades en los requerimientos pueden causar problemas graves en los proyectos de software, como retrasos, sobrecostos y fallas en el producto final. Por eso, es importante detectarlas y resolverlas lo antes posible. Esta aplicación utiliza técnicas de machine learning y bases de conocimiento para identificar las ambigüedades en los requerimientos. El machine learning se utiliza para aprender patrones en los requerimientos y detectar aquellos que son más propensos a ser ambiguos. Las bases de conocimiento se utilizan para almacenar información sobre las diferentes formas de expresar los requerimientos y ayudar a la aplicación a identificar las ambigüedades.

La aplicación está dividida en dos partes:

La interfaz de usuario: Esta parte permite a los usuarios ingresar los requerimientos de sus proyectos.

El motor de detección de ambigüedades: Esta parte utiliza las técnicas de machine learning y bases de conocimiento para identificar las ambigüedades en los requerimientos.

Este manual le enseñará cómo usar la aplicación para detectar ambigüedades en los requerimientos de sus proyectos.

Objetivos de este manual

Los objetivos de este manual son los siguientes:

Proporcionar una descripción general de la aplicación.

Explicar cómo usar la interfaz de usuario.

Describir el funcionamiento del motor de detección de ambigüedades.

Alcance de este manual

Este manual cubre los siguientes temas:

Cómo iniciar la aplicación.

Cómo ingresar los requerimientos de software.

Cómo ver los resultados de la detección de ambigüedades.

Este manual no cubre los siguientes temas:

La implementación técnica de la aplicación.

Los detalles del funcionamiento del motor de detección de ambigüedades.

Requisitos previos

Para usar esta aplicación, necesitará los siguientes requisitos previos:

Un ordenador con acceso a Internet.

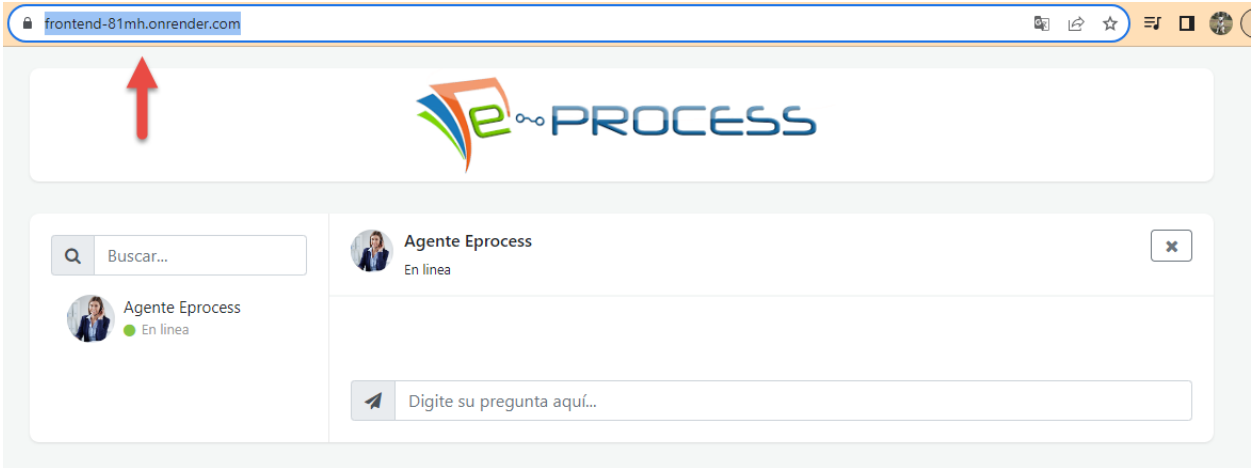
Un navegador web compatible con HTML5.

Conocimientos básicos de los requerimientos de software.

Si no cumple con estos requisitos previos, es posible que no pueda usar la aplicación correctamente.

4.2.1. Ingresar al aplicativo

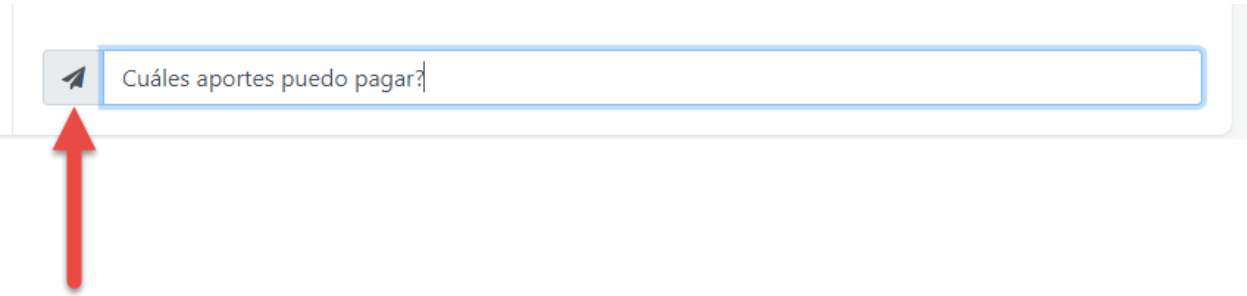
https://frontend-81mh.onrender.com/



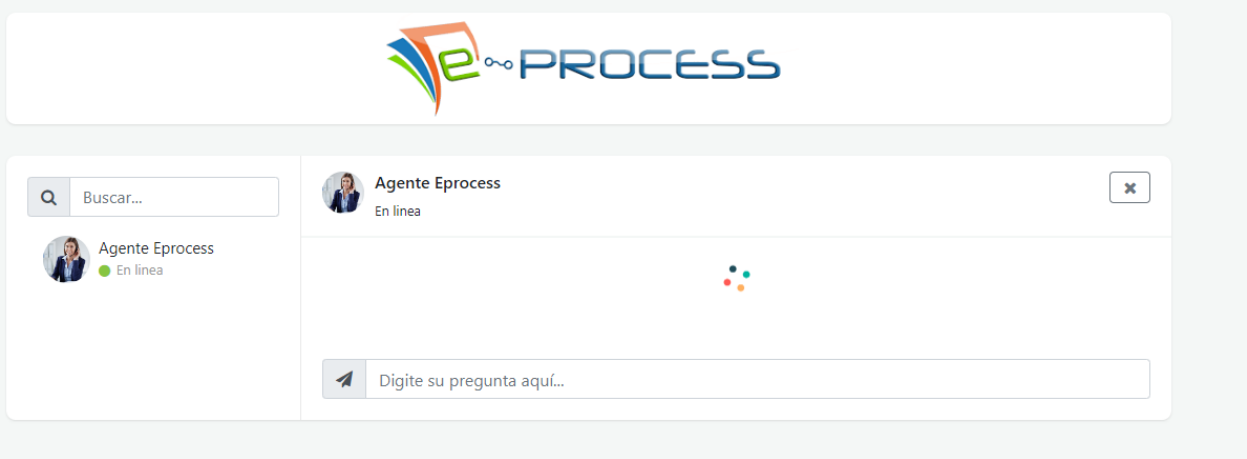
4.2.2. En la parte inferior hay un cuadro de texto para ingresar la pregunta:



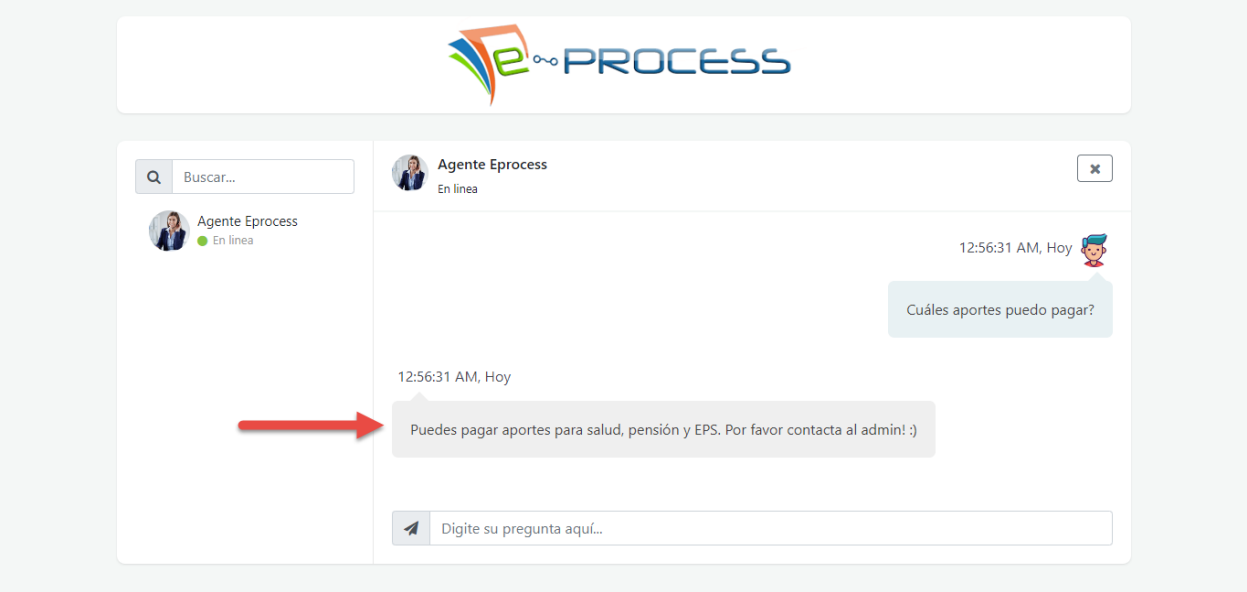
Digite la pregunta y dé clic en el icono para enviar el mensaje:



4.2.3. Por último, por favor espere que el sistema le genera la respuesta en Lenguaje Natural:



Respuesta generada:



ANEXO 4. REPOSITORIOS GITHUB

En los siguientes enlaces se encuentran los repositorios de software donde reside el código fuente de cada uno de los nodos y la implementación de método encontrado para la resolución de ambigüedades:

https://github.com/ingenierofabianmauricio/Nodo2

https://github.com/ingenierofabianmauricio/FrontEnd

https://github.com/ingenierofabianmauricio/Nodo1