

Maestría en ingeniería del software

Trabajo Fin de Maestría

Desarrollo e implementación de un prototipo basado en inteligencia artificial para la generación de consultas en lenguaje natural sobre bases de datos relacionales

AUTOR: Pedro Alejandro Alomía Flórez

PROFESOR: Rodrigo Cadena Martínez

Meta, Villavicencio, Colombia

Diciembre de 2025

Universidad Americana de Europa (ROR: https://ror.org/03698ym41)

Resumen

El presente trabajo académico tuvo como finalidad evaluar y comparar el rendimiento de una base de datos inteligente frente al de una base de datos tradicional mediante pruebas técnicas desarrolladas en escenarios controlados. Dichas pruebas analizaron resultados obtenidos en tareas específicas relacionadas con la gestión y administración de grandes volúmenes de información. El estudio surgió como una alternativa de mejora para los sistemas de almacenamiento tradicionales, debido a las dificultades observadas durante el manejo eficiente de datos generados continuamente por procesos y actividades cotidianas. Por esta razón, se investigó con mayor profundidad el diseño y la implementación de una base de datos inteligente utilizando una metodología de investigación con enfoque mixto. El propósito consistió en documentar y comparar el nivel de precisión alcanzado por ambos modelos mediante métricas, pruebas de rendimiento y análisis de eficiencia aplicados sobre conjuntos masivos de datos. La metodología contempló una revisión bibliográfica relacionada con sistemas gestores de bases de datos relacionales e inteligencia artificial aplicada. Finalmente, el trabajo verificó el grado de exactitud, eficiencia y capacidad de respuesta presentado por una base de datos inteligente frente a un modelo tradicional en procesos de gestión y administración de información. Además, los resultados obtenidos permitieron identificar ventajas significativas en tiempos de procesamiento, automatización de consultas y optimización de recursos computacionales, evidenciando que la incorporación de inteligencia artificial favoreció la toma de decisiones y mejoró la adaptabilidad del sistema frente a cambios constantes en la cantidad y complejidad de los datos administrados durante las pruebas experimentales realizadas satisfactoriamente.

Palabras clave: bases de datos inteligentes, inteligencia artificial, gestión de datos, bases de datos relacionales, rendimiento computacional

Abstract

This academic work aimed to evaluate and compare the performance of an intelligent database with that of a traditional database through technical tests carried out in controlled environments. The study analyzed the results obtained from specific tasks related to the management and administration of large volumes of information. The research emerged as an improvement alternative for traditional storage systems due to the limitations identified in handling continuously generated data. Therefore, the design and implementation of an intelligent database were examined using a mixed research methodology. The study documented and compared the precision, efficiency, and response capacity of both models through performance metrics and large-scale data testing. The methodology included a bibliographic review focused on relational database management systems and applied artificial intelligence. The results demonstrated significant advantages in processing time, query automation, resource optimization, and decision-making support, highlighting the adaptability of intelligent databases when managing complex and constantly changing data environments.

Keywords: intelligent database, artificial intelligence, data management, relational databases, computational performance.

Dedicatoria

A mis padres, los cuales, con mucho esfuerzo, me dieron las herramientas necesarias para seguir adelante, demostrándome que la disciplina y una buena actitud son las claves del éxito.

Agradecimientos

Quiero agradecer, en primer lugar, a mis docentes, los cuales me han acompañado a lo largo de todo este proceso de formación.

A mis amigos y familiares, los cuales me han dado el ánimo para seguir adelante con mi trabajo.

Pero en especial, quiero agradecer a mi tutor, el doctor Rodrigo Cadena Martínez, el cual, con mucha paciencia y dedicación, ha logrado orientarme, para la culminación de esta etapa.

Contenidos

Resumen....ii

Abstract.... iii

Dedicatoria....iv

Agradecimientos v

Capítulo 1. Marco introductorio. 1

1.1. Antecedentes. 1

1.2. Problema de investigación.....6

1.3. Justificación.....9

Capítulo 2. Marco Teórico.....11

2.1. Estado del arte. 11

2.1.1. Agente IA. 11

2.1.2. Aprendizaje automático.....12

2.1.3. Base de datos. 13

2.1.4. Bases de datos inteligentes. 14

2.1.5. Consulta en lenguaje natural (NQL).....15

2.1.6. Entornos inteligentes. 16

2.1.7. Model Context Protocol (MCP). 17

2.1.8. Procesamiento de lenguaje natural (PNL).....18

2.1.9. Sistema gestor de bases de datos (SGBD).....19

2.1.10. Structured Query Language (SQL). 20

2.1.11. Sistemas híbridos inteligentes basados en la lógica difusa.....21

2.1.12. Análisis descriptivo del impacto de la IA en Contac Center de Bogotá. ...23

2.1.13. Aplicación de la IA en la administración de la justicia colombiana. 25

2.1.14. La implementación de la IA en el sector público de Colombia.....27

2.1.15. Inteligencia artificial orientada a la función pública. 28

2.1.16. Modelo inteligente para bases de datos distribuidas. 30

2.1.17. Bases de datos inteligentes por Brena. R,1991. 33

2.1.18. Neo: Alearned Query Optimizer. 33

2.1.19. Balsa: Learning a Query Optimizer Without Expert Demonstrations. 35

2.1.20. Survey on Learnable Databases: A Machine Learning Perspective. 37

2.1.21. InferDB: In-Database Machine Learning Inference Using Indexes. 38

Capítulo 3. MARCO EMPÍRICO. 40

3.1. Hipótesis general. 40

3.2. Objetivos General. 43

3.3. Objetivos específicos. 43

3.4. Diseño y tipo de investigación. 44

3.5. Población y muestra. 45

3.6. Instrumentos de evaluación. 47

3.7. Descripción general de la solución propuesta. 49

3.7.1. Descripción detallada de cada fase. 50

3.8. Resultados. 51

3.8.1. Resultados experimentales. 51

3.8.2. Resultados perceptuales y selección de la muestra. 84

Capítulo 4. CONCLUSIONES Y RECOMENDACIONES 88

REFERENCIAS BIBLIOGRÁFICAS 93

APÉNDICES ..... **¡Error! Marcador no definido.**

Apéndice A. Diagrama de Gantt. 98

Apéndice B. Diagramas UML y documentación de casos de uso. 99

Apéndice C. Captura de pantalla de los datos en bruto del dataset utilizado. 105

Apéndice D. Diccionario de datos de la base de datos inteligente. 107

Apéndice E. Script de inserción de datos. 111

Apéndice F. Mockups de la interfaz gráfica de usuario. 112

Apéndice G. Técnica de Role Prompting. 114

Apéndice H. Instrumentos de medición. 115

Apéndice I. Resultados preliminares. 116

Apéndice J. Consultas NQL usadas para la generación de métricas de precisión. 119

Apéndice K. Tabla de consolidado de métricas. 121

Apéndice L. Encuesta aplicada para la toma de datos perceptuales. 122

Lista de tablas y figuras

Figuras

Figura 1 Diagrama de caso de uso 01 Realizar petición. 55

Figura 2 Diagrama de arquitectura de red. 56

Figura 3 Especificación de la arquitectura software. 57

Figura 4 Sitio web de datos abiertos del gobierno de Colombia. 59

Figura 5 Diagrama entidad relación de la base de datos. 60

Figura 6 Modelo relacional de la base de datos. 61

Figura 7 Inserción de los datos del dataset en MySQL. 62

Figura 8 GUI del prototipo de bases de datos inteligente. 63

Figura 9 Product backlog del proyecto. 64

Figura 10 Sprint backlog del proyecto visto por medio de un tablero Kanban. 64

Figura 11 Configuración de credenciales de acceso y API KEY. 66

Figura 12 Resultados comparativos entre el prototipo y el SGBD. 67

Figura 13 Resultado 1 de consulta 3. 71

Figura 14 Resultado 2 de consulta 3. 71

Figura 15 Resultado de medición de tiempo consulta 3. 72

Figura 16 Logs del SGBD de MySQL. 73

Figura 17 Resultado final 1 cantidad de mujeres por programa en el 2024-1 78

Figura 18 Resultado final 2 cantidad de hombres en Ing. electrónica 2022. 79

Figura 19 Resultado final 3 la facultad con más programas en la universidad. 79

Figura 20 Diagrama de Gantt del proyecto. 98

Figura 21 Diagrama de caso de uso 02 ejecutar consulta SQL. 99

Figura 22 Diagrama de caso de uso 03 ver resultados con medición de tiempo. 100

Figura 23 Diagrama de secuencia. 101

Figura 24 Diagrama de clases. 102

Figura 25 Diagrama de despliegue. 102

Figura 26 Documentación de caso de uso 01. 103

Figura 27 Documentación de caso de uso 02. 104

Figura 28 Documentación de caso de uso 03. 105

Figura 29 Datos del dataset "Admitidos Unillanos". 106

Figura 30 Diccionario de datos de la tabla colegio. 107

Figura 31 Diccionario de datos de la tabla estudiante. 108

Figura 32 Diccionario de datos de la tabla facultad. 108

Figura 33Diccionario de datos de la tabla programa. 109

Figura 34 Diccionario de datos de la tabla sede. 109

Figura 35 Diccionario de datos de la tabla admisión. 110

Figura 36 Fragmento de código del script de inserción de datos. 111

Figura 37 Mockup 01 petición de usuario. 112

Figura 38 Mockup 02 respuesta del prototipo. 113

Figura 39 Fragmento de código aplicando la técnica de role promting a un agente IA.114

Figura 40 Registros de tiempos de ejecución de peticiones NQL. 115

Figura 41 Resultado 1 de consulta 1. 116

Figura 42 Resultado 2 de consulta 1. 116

Figura 43 Resultado de medición de tiempo consulta 1. 117

Figura 44 Resultado 1 de consulta 2. 117

Figura 45 Resultado 2 de consulta 2. 118

Figura 46 Resultado de medición de tiempo consulta 2. 118

Figura 47 Consulta NQL nro. 1 usada para la generación de métricas de precisión. ... 119

Figura 48 Consulta NQL nro. 2 usada para la generación de métricas de precisión. ... 119

Figura 49 Consulta NQL nro. 3 usada para la generación de métricas de precisión. ... 120

Tablas

Tabla 1-Tabla de variables de investigación. 42

Tabla 2-Cronograma de actividades. 50

Tabla 3-Tabla de tiempos de ejecución. 74

Tabla 4-Tiempos de escritura de consultas. 75

Tabla 5-Consolidado de tasa de éxito, precisión semántica y tasa de corrección. 76

Tabla 6- Resumen estadístico de los valores consolidados de las métricas.. 81

Tabla 7- Resumen de pruebas estadísticas de Shapiro Wilk y t de Student. 82

Tabla 8-Métricas aplicadas a consultas NQL y SQL. 121

1. Marco introductorio

1.1. Antecedentes

En la actualidad, nos encontramos en una era donde la tecnología no para de avanzar, y en donde cada vez más se hace necesario integrar la inteligencia artificial (IA) a nuestra vida cotidiana. Podemos ver en estos momentos que la IA ya se encuentra en asistentes virtuales, en diferentes navegadores web y en aplicaciones de todo tipo, demostrándonos que el futuro está orientado hacia este campo.

Lo anterior se debe al éxito que ha tenido la IA para automatizar y realizar tareas que antes se consideraban totalmente desgastantes y monótonas, haciendo que los distintos procesos cotidianos que se llevan a cabo pasen a ser tareas sumamente sencillas y rápidas de ejecutar.

Estamos cada vez más inmersos en un mundo donde la agilidad lo es todo, y es por ello que cada vez más se hace necesario delegar tareas extensas y complejas a la IA.

Esto debido a que la sociedad moderna ha mostrado una profunda transformación a lo largo de toda su historia, en donde se puede apreciar la inmersión de dicha sociedad en un entorno cada vez más competitivo y ágil.

Siendo la empresa un elemento más de la sociedad, también se ve inmersa en dicho entorno, en donde día a día debe luchar con los desafíos de un mercado cambiante, contando únicamente con los recursos que tenga disponibles, dando como resultado una generación masiva de datos, como producto de labores cotidianas que la empresa desarrolla.

Lo que al final desencadena una problemática, relacionada directamente con la generación de dichos datos, en donde cada vez se hace más difícil el analizar y procesar información, debido al tamaño de los volúmenes de datos que se manejan, ocasionado la aparición de una serie de tareas sumamente extenuantes y complejas para la mente humana.

Es por ello, que la empresa de los tiempos modernos se ve en la obligación de delegar dichas tareas a la inteligencia artificial, para mantenerse competitiva y ágil. Haskel y Westlake (2017) nos dicen que mientras más pasa el tiempo, se hace necesario adaptarse y hacer uso de la inteligencia artificial para poder seguir siendo competitivos.

Un gran volumen de datos se define como un conjunto de información, el cual, debido a su naturaleza, se hace sumamente difícil de analizar y procesar de forma adecuada por medio de una serie de herramientas que permitan dar un resultado óptimo dentro de periodo de tiempos razonables (Hasperué et al., 2025).

Por ello, grandes compañías de la industria de la tecnología han visto en la IA un mercado muy rentable para aumentar sus ganancias, teniendo como punto de partida el consumo incremental de la misma a lo largo de estos ultimo años.

Según el portal web de noticias Webline India (2025), nos encontramos en un punto de no retorno, en donde la toma de decisiones informadas, así como la capacitación de personal y la experiencia de usuario, se han visto altamente afectadas, de forma positiva por parte de la integración de la IA, en los procesos internos de las compañías.

Otros aspectos que se han visto afectado por el uso de la IA han sido algunos hábitos comunes por parte de nosotros como usuarios, entre los cuales podemos destacar compras online y el uso de los motores de búsqueda tradicionales a la hora de hallar información.

Fuentes como Capgemini (2025, enero 21) nos indica que estamos alterando por completo la percepción de nuestro entorno, reemplazando en más de un 71 % el uso de motores de búsqueda como Google o Bing, por sitios web de los principales modelos generativos de IA como lo son ChatGPT, Gemini o Perplexity, viendo así pruebas concretas acerca del incremento del uso de la IA en todo tipo de tareas.

Es entonces, que a partir del éxito de ChatGPT, modelo inteligencia artificial generativa de Open AI, marca un modelo de negocio sumamente rentable, para ser adoptado por otras compañías, haciendo que se lancen al ruedo por competir en este nuevo mercado.

Desde producción de texto, mensajería, imágenes y traductores, hasta llegar a aspectos más avanzados como clonación de voz, generación de videos musicales o asistentes virtuales.

Es por ello que vale la pena conocer un poco más acerca de los antecedes más relevantes de la IA en el contexto del desarrollo e integración aplicada a las bases de datos, con la finalidad de entender los orígenes de las bases de datos inteligentes.

Antes de ChatGPT de OpenAI ya existían antecedentes de inteligencia artificial aplicada al ámbito empresarial, pero no eran tan populares en aquel entonces.

Gigantes tecnológicos como la empresa Oracle, pioneros en la implementación de bases de datos tradicionales para las empresas, vieron la necesidad de comenzar con el desarrollo e implementación de bases de datos inteligentes, integrando pequeños componentes de inteligencia artificial a sus productos, con la finalidad de agilizar los procesos de gestión de los datos dentro de una base de datos de manera autónoma (Oracle, s.f.).

Oracle se dio cuenta de que, mientras más volúmenes de datos maneja un base de datos, más difícil es de administrar. Y por ello, decidido sacar al mercado un producto llamado “Oracle Autonomous Databases” en el año 2020, un producto software orientado en la autogestión de los datos, con un intérprete integrado basado en inteligencia artificial, esto con la finalidad de facilitar la gestión de los datos, por medio del uso del lenguaje natural.

Su funcionalidad se basaba en darle órdenes al intérprete para que este traduzca el lenguaje humano en consultas SQL, y así extraer los datos solicitados dentro de una base de datos, comunicando de forma directa al usuario con esta.

Tiempo después, otro gigante tecnológico como lo es IBM, en el año 2023, decidió sacar al mercado un nuevo software denominado IBM Watson, el cual surgió como un proyecto de investigación que ya se venía desarrollando años atrás. IBM Watson es un modelo de inteligencia artificial orientado a la analítica de datos, en donde por medio de la gestión de grandes volúmenes de datos, permitía una serie de decisiones informadas, en base al análisis que realizaba. Al igual que las bases de datos autónomas de Oracle, IBM Watson podía interpretar el leguaje natural, para dar respuestas específicas, en base al conjunto de datos con el que se estuviera trabajando. Sin embargo, su principal diferencia con Oracle Autonomous Databases es que no tenía la capacidad de retroalimentarse, debido a que su principal uso se basaba en los datos que podía manejar en el momento, ya que carecía del componente de autogestión de datos, generando una dependencia casi completa entre usuario y máquina. Pese a ello, IBM sienta las bases de lo que a futuro se define como un sistema inteligente, y en consecuencia también de una base de datos inteligente, por medio de su manifiesto de la computación autónoma (IBM, 2001).

Otro antecedente que vale la pena mencionar es el de una empresa que no es tan conocida, pero aun así ha comenzado poco a poco con el desarrollo de bases de datos inteligentes en el ámbito empresarial, PowerData (2021) es una empresa desarrolladora de software en donde ya han optado por la implementación de bases de datos inteligentes dentro de su portafolio de servicios empresariales. Debido a que han entendido el potencial que tiene los datos cuando estos se interpretan de forma correcta, puede ser mucho más provechoso cuando se hace una analítica de datos usando IA, esta empresa no solo programa bases de datos inteligentes, sino que también monta todo un sistema automatizado que va desde la etapa de recolección de datos hasta la etapa de toma de decisiones informada por medio de IA.

Visto desde otro punto de vista, autores como Aguado-Labrador (2024) nos enseñan que los modelos de inteligencia artificial aplicados al desarrollo de bases de datos son tan versátiles que fácilmente se pueden adaptar para gestionar y reducir el consumo energético de un edificio por medio de la gran cantidad de datos generado por el mismo sistema. Así mismo, también nos recalca la importancia de entrenar estos modelos por medio de la integración aplicada a sistemas gestores de bases de datos tradicionales.

Finalmente, Hasperué et al. (2024), en su ponencia titulada: “Procesamiento inteligente de grandes volúmenes de datos”, nos señalan la gran importancia que tiene la inteligencia artificial en la gestión de grandes volúmenes de datos, siendo esta misma el eje central que permite desarrollar sistemas de información inteligentes más precisos, que permiten predecir y mitigar cierto tipo de problemáticas antes de su aparición. En esta ponencia, podemos ver cómo se pueden implementar un sistema inteligente, que por medio del análisis de grandes cantidades de datos originados por el sistema GPS de un vehículo se pueden predecir las rutas más óptimas, para evitar embotellamientos, congestión vehicular o accidentes de tránsito.

En resumen, se puede apreciar el gran impacto que ha tenido la IA en los diferentes ámbitos cotidianos, y más allá de ello, ver el gran avance que se está generando en el campo de las bases de datos.

1.2. Problema de investigación

En un mundo cada vez más globalizado, se puede observar como las tecnologías emergentes han pasado a ser un elemento crucial para el desarrollo de diferentes negocios, ocasionando así el surgimiento de nuevos mercados con las mismas oportunidades para todos.

Bien sea que se trate de una pyme o una gran empresa, cada organización cuenta con nuevos escenarios sumamente competitivos en donde la innovación y la optimización de recursos son clave para mantenerse a flote.

Asimismo, el nivel de exigencia por parte de los clientes ha aumentado, logrando que sea necesario generar productos y servidos con un mayor grado de calidad. Obligando a las empresas a realizar un mayor esfuerzo por mejorar sus procesos internos e infraestructura, con el fin de garantizar una productividad y rentabilidad a largo plazo.

Al final, todo este esfuerzo se traduce en datos e información que le permiten a la empresa de hoy en día estar al margen de las últimas tendencias del mercado para ser competitivas. Es por ello, que las bases de datos cada vez más juegan un papel sumamente relevante en las organizaciones actuales, siendo la información el activo más valioso de todos.

Sin embargo, existe una problemática intrínseca en este panorama, y es la cantidad de datos que una empresa puede generar al día, según Duarte (2025) este es un valor casi aproximado a 402.74 millones de Terabytes al día. Esto se debe, en gran parte, a que toda acción realizada en las labores cotidianas genera datos, entre los cuales se encuentran, principalmente, datos de compras en línea y datos generados por diferentes redes sociales.

Como consecuencia de lo anterior, las organizaciones se han visto en la necesidad de implementar centros de datos cada vez más complejos y con un mayor nivel de almacenamiento, para así garantizar la gestión, el análisis y el procesamiento de grandes

volúmenes de datos. Representando todo un desafío en términos de infraestructura tecnológica, consumo energético y eficiencia operativa.

Adicionalmente, la gestión y operación de bases de datos demanda la contratación de personal especializado en las áreas de la gestión de los datos, seguridad de la información y administración de sistema. Incrementando el nivel de dificultad que se tiene para garantizar la sostenibilidad de un sistema dentro de una organización.

Asimismo, se debe tener en cuenta que la administración y operación de una base de datos puede llegar a ser una tarea compleja, sobre todo cuando no se tiene un buen nivel de conocimiento en el lenguaje SQL, debido a que se pueden presentar fallos en la ejecución de las consultas por una mala sintaxis en su programación, modificaciones no previstas en la integridad de los datos y pérdida de información por un mal uso en comandos de borrado. De igual forma, durante los procesos de inserción y actualización de los datos, se podrían presentar problemas por información inexacta o duplicada, generando así una redundancia de datos.

En base a lo anterior, si se lograra una mejor gestión de los datos por medio de la integración de la IA, se podrían optimizar los procesos de creación y administración de una base de datos, para que estas sean más eficientes y menos complejas, así como también se obtendría otra medida de seguridad para proteger los datos con los que se está trabajando, y tal vez lo más importante de todo esto sería el hecho de colocar al alcance de todos la información que reside dentro de una base de datos, sin necesidad de tener conocimientos explícitos en SQL, lo que implicaría un menor consumo de diferentes recursos, al momento de llevar a cabo esta tarea de gestión de datos.

Si se logra que una base de datos cuente con un mayor rendimiento, se podrían disminuir aspectos como: el tamaño, haciendo que sean más fáciles de manipular; el reclutamiento de personal con conocimiento técnico en SQL, la mitigación y/o disminución

de la cantidad de errores causados por el factor humano; así como también, en cierta forma, el tiempo dedicado a tareas complejas y monótonas, las cuales son bastantes comunes en procesos de gestión de bases de datos.

Aunque ya hay varios estudios acerca de las bases de datos inteligentes implementadas en entorno reales con resultados positivos, llegamos a un punto en donde seguimos en la lucha por determinar el grado de precisión de una base de datos inteligente contra una base de datos tradicional, evaluando su grado de exactitud para determinar qué tan confiables son y así poder preguntarnos si estamos listo para una migración hacia estas.

Según Brena (1991), las bases de datos inteligentes incorporan tecnologías avanzadas como la inteligencia artificial y el aprendizaje automático para optimizar su gestión.

Es por ello, que la posición de este autor debería ser nuestro punto de partida para comenzar a profundizar más en el tema de las bases de datos inteligentes y poder resolver esta interrogante.

1.3. Justificación

Hoy en día, la tecnología es uno de los factores más importantes dentro de una organización, ya que define si una empresa permanece en el mercado. Bien sea por medio de una buena gestión de clientes, una calidad inigualable en los productos que ofrece que ofrece o por una buena optimización de sus recursos.

Es por ello que autores como Alvarado (2023) definen un panorama en donde una organización debe optar por el uso de IA y de nuevas tecnologías, para mejorar sus procesos tanto internos como externos y así mantenerse a flote en el mercado.

La optimización y la automatización de tareas complejas y monótonas debería ser una prioridad para cualquier organización en estos momentos, con la finalidad de preservar y administrar ese recurso tan valioso, como lo es el tiempo.

Donde el activo más valioso de una empresa pasa a ser la información, dando como resultado una era dentro de las organizaciones denominada la era de los intangibles. Según Muñoz (s. f.), la era de los intangibles es como una nueva economía, en donde podemos ver como la forma de hacer negocios ha cambiado, siendo la tecnología y los datos los nuevos factores determinantes para la supervivencia de una empresa.

Esto debido a que la mayoría de los procesos empresariales giran en torno a la administración de sus propios datos, y como a partir de allí se puede obtener una gran ventaja estratégica, por encima de la competencia, mediante una toma de decisiones informada, por medio de la integración de nuevas tecnologías aplicadas a los modelos de negocios de cada organización.

Antes simplemente se sacaba un producto o servicio al mercado sin tener en cuenta la experiencia de usuario y el grado de satisfacción de este, lo que ocasionaba que muchas empresas se fueran a la bancarrota por sacar al mercado productos que nadie necesitaba.

Con el tiempo, se entendió que si se conocen exactamente cuáles son las necesidades del cliente, se podrían sacar al mercado productos y servicios con una probabilidad más alta de ser aceptados y consumidos por el cliente.

Por ello, en la actualidad, la manera más viable para conocer las necesidades de un cliente es mediante los datos que él mismos genera, recolectados por medio de encuestas de satisfacción y/o estudios estadísticos basados en historiales de consumo o de ventas.

A partir de allí, vemos como las empresas deben invertir parte de sus recursos en actualizar su infraestructura tecnológica, no solo para mantener la calidad de sus productos y servicios, sino también para salvaguardar ese activo tan valioso como lo son los datos almacenados dentro de una base de datos y la información que se extrae de esta.

En ese orden de ideas, al resolver la problemática que estamos planteando, se estaría un paso más cerca de una optimización de recursos para todas las organizaciones, en base al apalancamiento tecnológico ofrecido por las bases de datos inteligentes. Dando como resultado final, una toma de decisiones informada en base a análisis hechos con inteligencia artificial, en procesos de gestión y administración de grandes volúmenes de datos.

2. Marco Teórico

En el presente capítulo de este trabajo se presentan los conceptos teóricos más fundamentales de las bases de datos inteligentes, así como también el conjunto de las teorías y estudios previos más relevantes sobre el tema. Comenzado por un contexto local, siendo Colombia el lugar donde se origina esta investigación, y posterior a ello a un contexto internacional, en donde se busca relacionar y delimitar las áreas de conocimiento más relevantes para el objeto de estudio, el cual es el diseño y desarrollo de una base de datos inteligente.

La implementación de estas bases de datos, aplicadas a la automatización de tareas y resolución de problemas en el ámbito de la administración de grandes volúmenes de datos, trae consigo grandes beneficios los cuales están directamente relacionados con rendimiento, escalabilidad y tiempos de ejecución óptimos, a comparación de una base de datos tradicional.

Es por ello, que se ha escogido de forma muy específica las siguientes investigaciones, entre las cuales se encuentran principalmente trabajados de grado y artículos científicos, los cuales están orientados o relacionados al objeto de estudio, en donde podemos ver antecedentes concretos que involucran diferentes visiones acerca de los beneficios y desafíos que trae consigo el desarrollo de las bases de datos inteligentes en entornos reales.

2.1. Estado del arte

2.1.1. Agente IA

Un agente IA es una entidad autónoma que percibe su entorno y actúa para lograr objetivos definidos (Russell & Norvig, 2020). Otra definición puntual que se podría dar sobre este concepto sería el definirlo como un software especializado que actúa por sí solo, para llevar a cabo una toma de decisiones más acertada, en base a los datos que recolecta y analiza

de su entorno, con la finalidad de diseñar y planificar la ejecución de soluciones a tareas en concreto, de una forma rápida y segura, sin la necesidad de la intervención humana.

El objetivo por el cual se crearon estos agentes es para agilizar y garantizar el cumplimiento de tareas sumamente repetitivas y triviales, para alcanzar las metas establecidas dentro de una organización. Como ya se mencionaba antes, los agentes IA son totalmente autónomos y capaces de tomar decisiones por sí solos y, así mismo, también tiene la capacidad de aprender de sus errores y ser sumamente adaptables con otras tecnologías.

Estas características convirtieron a los agentes de IA en herramientas altamente precisas y eficaces para diferentes ámbitos, como la gestión de proyectos, atención al cliente, ventas y marketing digital, entre otros.

Así mismo, estos agentes también aportan una solución moderna y efectiva para automatizar procesos y tareas, aportando una mejor eficiencia operativa.

2.1.2. Aprendizaje automático

El aprendizaje automático es una subdivisión existente en el campo de la IA, y se enfoca en la programación de algoritmos capaces de reconocer patrones, según Mitchell (1997) esta clase de algoritmos le permiten a la IA aprender en base a una serie de experiencias iniciales, las cuales no son otra cosa que el mismo conjunto de datos que la misma IA procesa.

Se define como conjunto de datos a toda colección organizada de datos estructurados, los cuales por su naturaleza pueden ser utilizados para diversas tareas, como lo pueden ser, cálculos predictivos basados en analítica de datos o la mejora de modelos de IA por medio de procesos de aprendizaje supervisado y no super visado, permitiendo entonces la construcción de las bases del aprendizaje automático (López & Pérez, 2024).

En base a lo anterior, se puede apreciar como el aprendizaje automático se considera uno de los pilares que le permiten a una IA desarrollar predicciones muy precisas en base al

gran volumen de los datos que tiene disponibles. Esto, debido a que el proceso de analizar grandes volúmenes de datos suele ser una tarea casi imposible para una mente humana.

El aprendizaje automático también permite que un sistema siempre esté en constante mejora, debido a la acumulación de experiencias generadas en base al análisis continuo de los conjuntos de datos que recibe.

2.1.3. Base de datos

Una base de datos es una estructura electrónica que cumple con la función de recolectar, procesar y almacenar de forma ordenada uno o más datos.

Su propósito es en el de convertirse en un repositorio para que los datos dentro de una organización siempre estén protegidos y disponibles, con la finalidad de preservar su integridad y validez (Elmasri & Navathe, 2016).

Dentro del contexto de las bases de datos, existen dos grandes clasificaciones de estas, las cuales son: las bases de datos relaciones y las bases de datos no relacionales, existiendo diferencias muy notorias entre ambas.

Por su parte, las bases de datos relaciones se basan en el modelo propuesto por Edgar F. Codd, donde los datos reposan en tablas, las cuales se integran con otras, mediante el intercambio de columnas, dando origen a lo que se conoce como el modelo relacional, en donde por medio de llaves primarias y llaves foráneas exista una relación entre tablas y así obtener acceso a todos datos que se encuentren dentro de cada una.

La llave primaria es una columna dentro de una tabla, la cual se encarga de identificar de manera inequívoca cada registro dentro de una tabla albergada dentro de un base de datos.

Mientas que la llave foránea es la representación de una columna considerada llave primaria dentro de otra tabla, haciendo que se relacionen entre sí por medio de esta conexión.

Aunque el modelo relacional de Edgar F. Codd es ampliamente aceptado en los tiempos modernos, con el pasar el tiempo se está quedando atrás, debido a que los problemas

en el ámbito de las bases de datos se volvieron más complejos, debido a la enorme cantidad de datos que se manejan en estos momentos, dando como resultado varios problemas a la hora de procesarlos.

Es entonces cuando surgen un nuevo modelo conocido como las bases de datos no relacionales, siendo este tipo de bases de datos una respuesta para poder gestionar de forma efectiva los grandes volúmenes de datos que se presentan en la actualidad.

Este tipo de bases de datos se caracterizan por manejar problemas descentralizados y horizontales, haciendo que sean totalmente flexibles y adaptables al conjunto de datos que manejan, ocasionando que sean más rápidas a la hora de buscar datos dentro de un gran volumen de datos.

Aunque las bases de datos no relacionales son bastante flexibles y soportan una gran cantidad de datos, su principal desventaja radica en mantener una consistencia e integridad de los datos a la hora de utilizarlas para realizar consultas.

Adicional a ello, si se necesita extraer datos desde diferentes lugares dentro de una base de datos no relacional, se vuelve una tarea un poco más compleja, ya que no están diseñadas para esa tarea.

Por eso, en la actualidad, ambos modelos de bases de datos coexisten y son aplicados en base a las necesidades que se tengan.

2.1.4. Bases de datos inteligentes

Las bases de datos inteligentes constituyen el siguiente paso en la escala evolutiva de las bases de datos, siendo estas la unión entre inteligencia artificial y una base de datos tradicional. Estas bases de datos son bastantes particulares, ya que contienen mecanismos de autogestión, modelos predictivos y optimización de recursos, en base a los datos que están administrando, logrando un rendimiento superior al de una base de datos tradicional o normal, así como también un mayor grado de usabilidad por parte de un usuario.

Su principal ventaja con respecto a una base de datos tradicional es que cuentan con la capacidad de hallar y entender patrones en base a los datos que tiene, lo que significa que al final las bases de datos inteligentes se pueden transformar en sistemas que aprenden de sí mismos, para alcanzar un estado de mejora continua a nivel de administración y gestión de datos.

2.1.5. Consulta en lenguaje natural (NQL)

Antes de definir este término, se hace necesario entender qué es en si el lenguaje natural, siendo este, precisamente, el lenguaje que nosotros como seres humanos utilizamos para comunicarnos entre nosotros mismos.

Dicho esto, una consulta en lenguaje natural, traducido al inglés como Natural Query Language (NQL), es sencillamente la traducción del lenguaje natural en consultas SQL.

Es decir, que se podría definir como un paradigma que busca flexibilizar la usabilidad de las bases de datos, con la finalidad de garantizar una democratización de los datos, haciendo que estos estén disponibles para todos los usuarios que los necesiten.

En base al contexto anterior, se podría garantizar una gobernanza de datos dentro de cualquier organización, facilitando el acceso a la información que se encuentre dentro de una base de datos a cualquier tipo de usuario, sin necesidad de que este último cuente con un conocimiento técnico avanzado en SQL, para poder manipular una base de datos relacional.

Con NQL se logra romper esa barrera, y hacer que cualquier usuario dentro de una organización sea capaz de acceder a los datos, sin necesidad de tener ese conocimiento técnico en SQL.

Según Yang et al. (2022), NQL es uno de los componentes más importantes dentro de las bases de datos inteligentes, ya que es el vehículo que permite agilizar la interacción hombre-máquina, aplicada a la gestión de datos para una toma de decisiones informada, de forma precisa.

Por ejemplo, si se quisiera conocer todos los datos de los colaboradores cuyo nombre sea Juan dentro de una base de datos, su consulta SQL seria la siguiente:

```
SELECT * FROM EMPLOYEES WHERE NOMBRE =’JUAN’;
```

En base a la consulta SQL anterior, si se quisiera hacer uso de una base de datos inteligente, por medio del uso de NQL, obtendríamos que la instrucción pasaría a ser la siguiente:

“Dame todos los datos de los empleados, que se llamen Juan”.

Haciendo que la segunda opción sea ejecutada por cualquier usuario, sin necesidad de tener conocimientos específicos en SQL.

2.1.6. Entornos inteligentes

Según Cook y Das (2005), un entorno inteligente es un espacio físico el cual ha sido automatizado mediante toda clase de sensores, para ser administrados por componentes de IA. En donde por medio de campos de estudio como el internet de las cosas (IoT) y la analítica de datos, se llega a la implementación de espacios autogestionados en base a las preferencias de usuario.

Y en donde, al igual que en una base de datos inteligente, lo que se busca es que un agente de IA tome el control sobre ciertos procesos de manera autónoma.

Su relevancia para esta investigación se basa en la relación que tiene con las bases de datos inteligentes, ya que son estas bases de datos un componente integrado que hace parte de los entornos inteligentes, haciendo que estos se conviertan en un caso aplicado de lo que es una base de datos inteligente implementada en un escenario real.

A partir de este punto, aparecen otros conceptos importantes que se relacionan dentro del contexto de los entornos inteligentes y que también presentan pequeñas integraciones de bases de datos inteligentes en sus componentes, dichos conceptos son:

- **Inteligencia ambiental**, este término hace referencia a la evolución de los entornos inteligentes, en donde pasamos a un control total del entorno por parte de un agente inteligente, creando un ambiente sumamente reconfortante y adaptable al usuario, en base a sus necesidades, sin que este tenga que hacer algún tipo de intervención.
- **Computación oblicua**, son las bases del internet de las cosas, siendo la computación oblicua el hecho de poder integrar toda clase de procesadores y microprocesadores a objetos de uso cotidiano.
- **IoT**, el cual es la tecnología que permite la conexión a internet de objetos de uso cotidiano, como relojes, ropa, accesorios y/o electrodomésticos, entre mucho otros.
- **Domótica**, es una rama de la robótica que se encarga de automatizar espacios en base a acciones concretas de un usuario, como, por ejemplo: aplaudir para encender una habitación o abrir una cochera, cuando se detecta la placa de un vehículo por medio de una cámara.

2.1.7. Model Context Protocol (MCP)

MCP es un protocolo de comunicación emergente para la interacción segura entre modelos de IA y sistemas externos, ofreciendo un marco estandarizado para la integración y ejecución de tareas.

Dicho de otra forma, se podría decir que también es un componente sumamente relevante para todos los agentes IA utilizados actualmente, en donde se establece todo un protocolo de comunicación entre un agente IA y una herramienta externa para que sea dicho agente el encargado de manipular esa herramienta.

Cuando se habla de herramientas externas, se indica que los agentes de IA pueden obtener la capacidad de manipular de forma segura una serie de elementos como programas, servicios, API´s y ciertos softwares sin la necesidad de la intervención humana.

El objetivo de este protocolo de comunicación es el de delegarle a un agente IA la manipulación de una herramienta en concreto, con la finalidad de que tome las decisiones más acertadas en base a su percepción y su entorno, para que así se logre una automatización segura en los procesos que ejecute.

Por lo general, los agentes IA deben esperar una instrucción en específico para poder actuar. Esto quiere decir que, pese a poder contar con todos los datos necesarios para poder llevar a cabo una tarea en concreto, se debe contar con la intervención de una persona para poder iniciarla.

Lo que permite el protocolo es que se suprima el factor humano, para que así de forma automática se lleve a cabo el desarrollo de tareas de forma automatizada.

En ese orden de ideas, es importante conocer más a fondo las partes del MCP, las cuales son:

MCP Host, siendo este propiamente el conjunto de agentes IA, o herramientas de IA, que pueden integrase para la conexión y manipulación de componentes específicos.

MCP Client, los cuales representan protocolos específicos de comunicación que se encargan de dar acceso a los MCP Host y conectarlos con los MCP Server.

MCP Server, son las herramientas que van a ser integradas para que los MCP Host tomen el control y comiencen a manipular dichas herramientas.

2.1.8. Procesamiento de lenguaje natural (PNL)

El procesamiento del lenguaje natural, mejor conocido como Natural Language Proccesing por su traducción en inglés, es una rama de la inteligencia artificial la cual se encarga del estudio e implementación del lenguaje natural, para que sea estudiando e interpretado por una máquina.

Según Jurafsky y Martin (2021), el PLN combina la lingüística computacional, el aprendizaje automático y otras ramas como la estadística, para hacer posible que las máquinas entiendan el lenguaje humano.

De esta manera, se construyen las bases para el funcionamiento de los agentes inteligentes enfocados en interacciones de usuario para el desarrollo de tareas básicas. Logrando así, una mejor comprensión de instrucciones para ejecutar tareas de forma autónoma.

Siendo uno de los mayores exponentes de este campo el desarrollo de la inteligencia artificial generativa, la cual se encuentra presente en modelos de inteligencia artificial como ChatGPT de OpenAI o Gemini de Google.

En base a lo anterior, podemos ver como el PNL les permite a los agentes IA especializarse en otras tareas como la generación de texto, imágenes, videos, traducción de idiomas en tiempo real, y más importante aún, el desarrollo de habilidades para hallazgos de patrones y tendencias, por medio de la creación de modelos predictivos.

2.1.9. Sistema gestor de bases de datos (SGBD)

Un SGBD es un software especializado en la creación y administración de bases de datos, siendo este una herramienta indispensable para trabajar en el ámbito de la gestión y administración de los datos dentro de una organización.

Su función principal consiste en garantizar la seguridad, integridad y disponibilidad de los datos, haciendo que efectivamente la información que se genera mediante el uso de una base de datos sea relevante, verídica y consistente.

En base a lo anterior, se puede apreciar cómo han evolucionado las bases de datos. Desde sus orígenes a manos de Edgar F. Codd y su modelo relacional propuesto en 1970, hasta la actualidad con implementación y desarrollo de los diferentes sistemas gestores de

bases de datos que inundan el mercado del desarrollo del software, entre los cuales podemos mencionar algunos como Oracle, MySQL, MariaDB, PostgreSQL y SQL Server, entre otros.

Así mismo, con la implementación de los modelos no relacionales de bases de datos, también se han generado nuevos SGBD NoSQL para poder administrar bases de datos no relacionales, entre los que se pueden encontrar herramientas como MongoDB, CassandraDB o CouchDB.

2.1.10. Structured Query Language (SQL)

Structured Query Language o lenguaje de consulta estructurado, traducido al español, es un lenguaje de programación declarativo, por medio del cual se puede realizar el proceso de gestión y administración de los datos, dentro de un sistema gestor de bases de datos relacional.

Con lo que respecta a la estructura de SQL, se compone de cinco partes, las cuales permiten tener un control total, tanto de la estructura de una base de datos como de los datos que residen dentro de esta

Dicho esto, se procede a especificar cada componente de SQL:

- **DDL**: los Data Definition Language o lenguaje de definición de datos, es un componente dentro de SQL que permite crear, alterar y suprimir objetos dentro de una base de datos. Las instrucciones por excelencia en este componente son ALTER, DROP y CREATE.
- **DML**: Data Manipulation Language o lenguaje de manipulación de datos, es un componente de SQL que permite realizar operaciones de inserción, actualización, consulta y eliminado de datos. Las instrucciones por excelencia en este componente son INSERT, UPDATE y DELETE.
- **DCL**: Data Control Language o lenguaje de control de datos, es un componente SQL, que se encarga de administrar los permisos de usuario dentro de una base de datos,

con la finalidad de restringir las operaciones CRUD que un usuario puede hacer a nivel de datos y de objetos de bases de datos. Las instrucciones por excelencia en este componente son GRANT y REVOKE.

- **TCL**: Transaction Control Language o lenguaje de control de transacciones, es un componente SQL que permite administrar todas las transacciones de datos que se generan dentro de una base de datos. Siendo COMMIT y ROLLBACK las operaciones más usadas dentro de este grupo, con la finalidad de aprobar o rechazar un movimiento de transacción de datos dentro de una base de datos.
- **DQL**: Data Query Language o lenguaje de consulta de datos, se caracteriza por buscar y retornar un conjunto de valores dentro de un base de datos. Su función principal es la de visualizar los datos que extrae en una base de datos para que puedan ser representados en pantalla. Siendo el SELECT el comando principal para hacer esta tarea.

2.1.11. Sistemas híbridos inteligentes basados en la lógica difusa

Este es un trabajo de grado en donde la implementación de la lógica difusa y su relación con el aprendizaje profundo ha sido estudiada un poco más a fondo en base a la aplicación teórica de estos conceptos, orientado al desarrollo de nuevas tecnologías para implementar el proceso de monitoreo de diferentes tipos de cultivos por medio de captura de imágenes satelitales, con el propósito de poderlos identificar y clasificar de una forma más precisa, por medio del aprendizaje profundo y las bases de datos difusas (Zapata, 2022).

La relevancia de este trabajo académico, para el objeto de estudio que se está trabajando, parte del uso e implementación de dicha lógica difusa aplicada a las bases de datos tradicionales.

En donde por medio de modelos de inteligencia artificial, se desarrolla un sistema inteligente, capaz de realizar consultas al componente de bases de datos donde se encuentra

toda la información de los cultivos, dando como resultado un primer bosquejo de lo que sería un sistema de información con un componente integrado de bases de datos inteligente.

Vale la pena recalcar la importancia del aprendizaje profundo para este trabajo, debido a que es el elemento más relevante para explicar cómo se unen la IA con las bases de datos, el aprendizaje profundo es una subclasificación del aprendizaje automático, que, a su vez, es una técnica aplicada al entrenamiento de la IA. Lo que se busca con este aprendizaje profundo es imitar la forma de cómo piensa un ser humano, simulando su cerebro por medio de redes neuronales (LeCun et al., 2015). Siendo estas redes un sistema computacional avanzado que le permite a la máquina tener la capacidad de reconocer imágenes con bastante precisión, que para este caso en particular son imágenes de cultivos.

A partir de este punto, pasamos a otro componente sumamente importante de este trabajo, y es por ello que debemos dar una definición un poco más concreta acerca de lo que es la lógica difusa, con la finalidad de entender el funcionamiento de una base de datos difusa, siendo la lógica difusa un tipo de razonamiento en específico a partir de un argumento vago o abstracto.

Por ejemplo, podríamos decir que la siguiente expresión: “hace calor”, es una expresión bastante vaga e imprecisa que no nos dice mucho.

Sin embargo, haciendo uso de la lógica difusa, podríamos hacer uso de una escala numérica en grados centígrados, dando un aproximado más específico y concreto acerca de la expresión anterior de forma precisa haciendo uso de dicha escala.

El objetivo de la lógica difusa es interpretar estados difusos basados en el lenguaje natural, para así mismo dar un resultado muy específico.

Es decir, que la lógica difusa puede interpretar un estado de temperatura como: “hace calor”, en toda una escala de términos como muy frío, frío, tibio, caliente y muy caliente.

Para que, así mismo, la premisa anterior “hace calor” se convierta en un valor medible y calculable y nos dé información más exacta acerca del estado de temperatura del entorno, por medio de una escala relacional entre una categoría y valores numéricos para que sean unidades de medida en grados centígrados.

Dicha escala se conoce como “conjunto difuso”, un concepto que hace parte de la lógica difusa en donde, por medio del grado de pertinencia, se puede llegar a dar un valor en específico, para la expresión anterior, indicando, por ejemplo, que la expresión “hace calor” se podría relacionar con una escala donde se marca una categoría denominada “caliente”, marcando una temperatura de 26C°.

Para el caso específico de este trabajo de grado, se implementó el concepto de la lógica difusa, con respecto al aprendizaje profundo, siendo la misma lógica difusa la herramienta que permite clasificar las imágenes satelitales de diferentes cultivos, y así determinar su tipo.

Respecto a la relación de este trabajo con el objeto de estudio, podemos ver que su relevancia se basa en el uso de las bases de datos difusas, como una herramienta para la implementación de agentes IA, totalmente autónomos en el análisis de un gran conjunto de datos no estructurado, aplicado a la clasificación de cultivos.

Dando como resultado indicios muy puntuales de la interacción que tienen los agentes de IA, con diferentes componentes de administración de bases de datos.

2.1.12. Análisis descriptivo del impacto de la IA en Contac Center de Bogotá

Este estudio parte de la hipótesis de que la IA, en un futuro cercano, desplazará de forma paulatina las funciones de diferentes empleados en establecimientos de Contac Center, siendo estas empresas especializadas en llamar, escribir y/o establecer canales de comunicación con potenciales cliente (Sánchez et al., 2023).

El estudio comienza analizando los sistemas de información de los Contact Center en la ciudad de Bogotá (ciudad capital de Colombia), en donde por medio del uso de agentes de IA se logra automatizar la atención al cliente.

Vale la pena aclarar que un Contac Center se diferencia con respecto a un Call Center en cómo se comunica con un cliente. Siendo que los Call Center solo atienden a sus clientes por medio de llamadas telefónicas, mientras que un Contact Center puede gestionar a sus clientes por cualquier otro medio, como videollamadas, correos, canales de mensajería instantánea como WhatsApp o Telegram, así como también una página web o red social por medio de chats.

En donde se puede apreciar un conjunto de resultados favorables debido a la capacidad operativa de los agentes de inteligencia artificial, basados en una reducción considerable en los tiempos de respuesta y la cantidad de usuarios atendidos.

Sin embargo, al ver los resultados desfavorables de la implementación de este sistema de información, se pudo apreciar una serie de anotaciones pertinentes en los comentarios de los usuarios que fueron atendidos por estos agentes. En donde se señala una serie de quejas basadas en inconformidades específicas como falta de empatía y la escaza interacción para resolver problemas o situaciones complejas que requerían un trato más humano para poder solucionarlas.

Lo que lleva a este estudio a indicar que los procesos automatizados al 100 % libres de intervención humana no garantizan que se dé una experiencia de usuario totalmente favorable. Incluso el estudio también menciona que en casos muy específicos los agentes de IA no estaban en la capacidad de dar una respuesta satisfactoria, porque requerían de un entrenamiento más riguroso en sus modelos de aprendizaje.

Así mismo, el estudio indica que los Contac Center tuvieron que realizar grandes cambios para poder seguir trabajando con esta clase de sistemas de información.

La solución documentada en este estudio nos señala que se cambiaron los roles de los agentes de IA, para que actúen como asistentes o coach de los mismos empleados de cada Contac Center, con la finalidad de que estos mejoraran sus habilidades comunicativas, principalmente en la gestión de resolución de conflictos y el dominio de una segunda lengua, como el idioma inglés, así como también en sus tareas diarias de atención al cliente, por medio de ejercicios de simulación de ventas, cerrando el estudio con un balance positivo ante este cambio.

El estudio termina dándonos una breve conclusión acerca de cómo los agentes de IA se están volviendo más indispensables en este tipo de negocios y de cómo a futuro se espera que sean capaces de sonar más humanos para realizar tareas de gestión al cliente. Sin embargo, los autores también señalan que el factor humano sigue siendo indispensable de una u otra forma.

En base a casos de estudio como este, también se han encontrado otros autores como Arias-Naranjo y López-Martínez (2023), los cuales muestran usos distintos pero bastantes complementarios a este modelo de negocio, en donde exponen como la IA también puede ser aplicada a tareas de control y gestión de mensajes y llamadas de spam, dándonos una visión más amplia acerca de cómo la IA puede aplicare a tareas complejas de gestión de clientes.

La relevancia de este trabajo para el objeto de estudio es el aporte que se hace, documentando el desarrollo de sistemas basados en agentes IA, para que estos sean capaces de manipular herramientas por medio de protocolos de contexto de modelos (MCP) en entornos empresariales.

2.1.13. Aplicación de la IA en la administración de la justicia colombiana

Este artículo científico tuvo como objeto de estudio el impacto de la inteligencia artificial en el ámbito del derecho colombiano (Bernal, 2024). En el estudio se recopilan distintos puntos de vista de diferentes autores y se evalúa el uso de los diferentes agentes de

inteligencia artificial aplicados al desarrollo de varias tareas específicas en procesos judiciales, con el objetivo de evaluar su nivel de precisión, razonamiento, fiabilidad y grado de transparencia.

Entre los elementos más destacados dentro de este artículo encontramos el trabajo de López-Vega et al. (2023), en donde confirman la posición que toma este autor, indicando la favorabilidad del uso de la IA aplicada a procesos judiciales, mediante un análisis detallado a herramientas ya existentes en este ámbito como Pretoria y Prisma.

Si bien los resultados son bastantes positivos, el autor nos demuestra que todo depende del contexto inicial que se le dé a la inteligencia artificial para dar una respuesta, por lo que se llega a la conclusión de que se deben tomar medidas más estrictas e implementar de forma rigurosa leyes que permitan garantizar el grado de transparencia de sistemas inteligentes en Colombia.

Autores como Chaudhuri y Narasayya (2007) definen un sistema inteligente como aquel que utiliza un componente de IA capaz de percibir su entorno y procesar datos con la finalidad de que sea este mismo el encargado de tomar las decisiones más acertadas de manera autónoma.

La relevancia de este articulo no es solamente por los datos que recopila sobre la IA o las nuevas leyes que están emergiendo en base a este tema, su verdadera relevancia para el objeto de estudio radica en el hecho de que este tipo de trabajos abre un serio debate acerca de cómo se pueden solucionar problemas complejos como lo son: una sentencia, un proceso de restitución de tierras o un fallo judicial en segunda instancia, por medio del uso de IA. Dando como resultado un antecedente que muestra una disminución en los tiempos de los procesos judiciales y con un grado mucho mayor de imparcialidad.

2.1.14. La implementación de la IA en el sector público de Colombia

En este trabajo de grado Figueroa (2024) nos da una visión mucho más amplia de la transformación digital que se está viviendo en Colombia en el ámbito del derecho a manos de la IA.

Podemos ver como el autor hace un análisis general acerca de los aspectos más relevantes del sector público colombiano, en donde se recopila a grandes rasgos toda la información pertinente acerca de procesos gubernamentales, hasta llegar a casos más específicos del derecho.

Así mismo, se presenta una visión más concreta acerca de la creación de futuros prototipos de herramientas inteligentes aplicadas al ámbito de prevención de la evasión de impuestos y fraude fiscal.

Si bien este trabajo de grado muestra resultados bastante similares al resto de los estudios ya antes mencionados, lo realmente interesante es ver cómo se documentan casos más puntuales en donde la IA pasa a ser un elemento de gestión indispensable para la gobernanza de datos dentro de una organización. Según DAMA International (2017), la gobernanza de datos representa la información, normas, roles y estándares dentro de una organización como tal, siendo esta un factor clave para una buena gestión de procesos organizacionales.

En base a lo anterior, se puede apreciar como la IA se está transformando en ese factor clave para fortalecer los procesos internos dentro del ámbito de diferentes elementos que conforman al sector público colombiano.

Este estudio también nos muestra las dificultades de todo el proceso de implementación de estas herramientas inteligentes ya que, al requerir acceso a las distintas bases de datos del gobierno, hay muchos obstáculos de tipo legal.

El autor también llega a la misma conclusión de otros, en donde señala que la IA puede pasar a ser una herramienta de soporte muy efectiva para agilizar distintas clases de proceso.

A partir de este punto, debemos definir dos componentes clave en los que se basa este trabajo, los cuales son la arquitectura de datos y como esta se relaciona con la gobernanza de datos.

La arquitectura de datos se considera un marco de trabajo que busca especificar el cómo se deben gestionar los datos dentro de una organización, teniendo como punto de partida la definición de los mecanismos de recolección, transformación, distribución y consumo de estos (DAMA International, 2017).

Podemos ver entonces como la arquitectura de datos se alinea con los estándares definidos dentro la gobernanza de datos, por medio de la cultura y el conocimiento organizacional, dando como resultado una aplicación real de lo que un agente de IA puede llegar a realizar en base a un gran conjunto de datos, dentro de los alcances de este trabajo.

Aunque no se había mencionado antes, los resultados de este trabajo se basaron en conjuntos de datos bastantes grandes, como sistemas de PQR, formularios de registro y/o declaraciones de renta a nivel nacional, entre muchos otros procesos del Estado colombiano.

Si bien este trabajo tiene diferentes componentes vistos desde distintos puntos de vista, el verdadero interés y grado de relevancia para el objeto de estudio se basa en el concepto de la arquitectura y gestión de los datos por medio de una gobernanza de datos mediante el uso de IA.

2.1.15. Inteligencia artificial orientada a la función pública

Este trabajo de grado tiene una visión bastante similar a los otros estudios ya antes mencionados, en donde se explora los beneficios de la IA aplicada a la gestión pública y

gobernanza de datos dentro de la normativa colombiana para los procesos organizacionales internos y externos del Estado colombiano.

Pero su diferencia y relevancia radica en dos puntos clave:

El primer punto se basa en la definición del concepto de “transparencia algorítmica”, que, según su autor Guevara (2025), es la disponibilidad de la información que permite conocer la operación y rendimiento de una herramienta IA, con la finalidad de garantizar una ejecución limpia y, valga la redundancia, transparente de esta ante un posible proceso de auditoría pública. El concepto surge como una iniciativa gubernamental, para que toda la ciudadanía en general conozca el funcionamiento de estas herramientas, con la finalidad de garantizar el mayor nivel de imparcialidad posible.

El segundo punto, y tal vez el más importante, se basa en la recolección de datos y evidencias de sistemas inteligentes y agentes IA en Colombia, lo que nos permite tener una evidencia más sólida acerca del desarrollo de esta tecnología en este país.

En base al segundo punto, vale la pena profundizar un poco más en este tema y dar a conocer dichos sistemas inteligentes, con el objetivo de ver un panorama más real y acorde al objeto de estudio.

PretorIA es una apuesta del Gobierno colombiano para la implementación de sistemas inteligentes en el ámbito judicial, su principal enfoque se basa en la automatización de todo el proceso de radicación y gestión de tutelas. Esta herramienta inteligente ayuda a los usuarios a radicar de forma correcta su tutela. Adicional a ello, sirve como asistente para que los jueces de los distintos despachos organicen su trabajo y encuentren de forma más rápida información acerca de los casos más críticos que tengan asignados.

Plataforma Océano, esta plataforma se considera un sistema inteligente con un enfoque más orientado a la analítica de datos por medio de la minería de datos. Su principal objetivo es que sirva como un sistema de detección ante el fraude fiscal y el enriquecimiento

ilícito de los ciudadanos colombianos. Aunque su iniciativa es bastante innovadora, este sistema fue abandonado en el 2020 ya que, por falta de recursos, no fue posible que fuera implementado y puesto en marcha. Este sistema fue específicamente diseñado para apoyar los procesos internos del órgano judicial de la fiscalía general de la Nación y aunque este sistema nunca se vio en funcionamiento, queda como un antecedente importante acerca de cómo ya se ha intentado implementar sistemas inteligentes con bases de datos en Colombia.

Sistema PRISMA, este sistema inteligente proponía un análisis basado en la literatura judicial del caso de un individuo, y así justificar ante un juez si este individuo era apto para recibir una medida de aseguramiento, esta acción corresponde a privar de la libertad a una persona de forma preventiva, para evitar que se escape mientras se resuelve su situación judicial. Aunque faltaba poco para su puesta en marcha, no se tuvieron las evidencias suficientes para garantizar que el sistema era lo suficientemente transparente y así dar una respuesta totalmente imparcial y libre de todo tipo de sesgos, por lo que nunca se pudo poner en marcha.

2.1.16. Modelo inteligente para bases de datos distribuidas

Este artículo científico muestra como es el desarrollo e implementación de un modelo inteligente para la optimización y análisis de bases de datos distribuidas y bases de datos federadas, por medio del uso de modelos canónicos.

Según Muñoz et al. (2005), lo que se busca es crear desde sus inicios un modelo ontológico de la información para representar una base de datos a partir de sus componentes más importantes.

Para entender un poco más el contexto de este trabajo se hace necesario abarcar los conceptos más fundamentales que definen la esencia de toda esta investigación.

Modelo inteligente: es aquel modelo que se basa en una estructura lógica computacional y matemática. Diseñada para el procesamiento de datos. Con el objetivo de

aprender y tomar decisiones, que permitan predecir acciones, sin que estas estén intrínsecamente programadas o definidas en su naturaleza.

Sin embargo, para este caso en particular, reciben ese nombre porque son creados desde cero y evidentemente no son tan avanzados como los modelos de IA que se conocen actualmente.

Dicho de una forma más específica, los modelos inteligentes son aquellos programas informáticos que han sido creados y entrenados con una gran cantidad de datos, para así realizar tareas en concreto, sin necesidad de la intervención humana.

Base de datos distribuida: son aquellas bases de datos que están representadas por el enlace y/o conexión lógica entre un conjunto de distintas bases de datos, las cuales, por lo general, no se encuentran dentro de una misma ubicación. Con la finalidad de soportar un volumen de datos mucho más grande, a comparación de un modelo centralizado, el cual se limita por la capacidad de hardware de la maquina donde se encuentra alojada (Mell y Grance, 2011).

Base de datos federada: es la unión de varias bases de datos. Su principal diferencia con una base de datos distribuida es que en las bases de datos federadas sí podemos apreciar una división clara entre cada uno de sus componentes. Es decir, que se pueden apreciar todas y cada una de las bases de datos que conforman a una base de datos federadas como si fuesen una sola, por su parte esta visibilidad dentro de una base de datos distribuida es totalmente inexistente ya que efectivamente se trata de una sola base de datos.

Ontología: en informática es la definición y/o especificación de los conceptos o componentes más relevantes dentro de un área de conocimiento claramente definida.

Por lo que, al aplicar este concepto en un ámbito real, permite que diferentes sistemas puedan comunicarse, al manejar una especie de estándar o protocolo dentro de sus compontes internos, cuando estos se conectan.

Taxonomía: en el ámbito de la informática, es un modelo de clasificación de los componentes de un sistema de forma jerárquica, lo que nos permite tener acceso a ellos de una forma más fácil y rápida.

Y su relación con la ontología radica en la forma de cómo podemos desglosar y organizar todos los componentes que conforman una base de datos por medio de un modelo inteligente, con la finalidad de estudiar a fondo su composición y estructura interna.

Modelos canónicos: se consideran patrones de diseño que permiten definir la arquitectura de software que se va a implementar, para permitir la comunicación entre los dos componentes ya antes mencionados. Se podría decir que es un lenguaje estándar o neutral encargado de traducir las instrucciones de un lado a otro, para garantizar la conexión y comunicación entre componentes de un sistema.

En resumen, los autores plasmaron en su investigación el análisis del comportamiento de un modelo de inteligencia artificial aplicada a la manipulación de una base de datos distribuida, dando como resultado una documentación rigurosa acerca del proceso de desarrollo de software de este modelo, sin el uso de API´s, frameworks o herramientas de desarrollo de código asociadas al Vipe Coding. Evidenciando así un proceso de desarrollo de software limpio y robusto aplicado a la inteligencia artificial, por medio de las metodologías de desarrollo tradicionales, las cuales están presentes en el ámbito de la ingeniería del software.

La relevancia de este artículo para el objeto de estudio se basa en la documentación e implementación de un modelo IA, aplicado a un tipo de base de datos en específico, dando como resultado un antecede que demuestra la viabilidad del desarrollo y programación del objeto de estudio en un escenario investigativo.

2.1.17. Bases de datos inteligentes por Brena. R, 1991

Pese a la antigüedad de este articulo científico y de todos los cambios que ha tenido el campo de la inteligencia artificial desde sus orígenes hasta los tiempos modernos, se hace necesario mencionar este artículo porque sienta las bases de la teoría aplicada al desarrollo de una base de datos inteligente, pasando por todo un estudio de modelos matemáticos sobre el tema hasta la programación e implementación de dichos modelos a una base de datos relacional, dando como resultado un antecede sólido y detallado sobre el proceso de desarrollo e implementación de una base de datos inteligentes.

Su análisis es bastante completo, ya que abarca desde los conceptos más elementales del objeto de estudio hasta llegar a la explicación por medio de modelos matemáticos y estadísticos acerca de cómo el lenguaje natural de consultas (NQL) es interpretado y traducido en un modelo inteligente para su posterior ejecución dentro de una base de datos.

Aunque su alcance es teórico, no dejar de ser bastante impresionante el cómo se detalla el funcionamiento de un base de datos inteligente por medio del lenguaje natural.

Se puede apreciar como desde aquella época ya se estaba visualizando el desarrollo de una base de datos inteligentes, y de cómo también esta misma base de datos podría autogestionar sus propios datos de entrada. También, se presentan parte de los cálculos y modelos necesarios para poder realizar esa interacción hombre-máquina.

Volviendo a recalcar la relevancia de esta investigación, se indica que efectivamente se convierte en el antecedente más relevante de todos, pese a su antigüedad, debido a que el autor demuestra por medio de un arduo trabajo el origen y funcionamiento de una base de datos inteligente.

2.1.18. Neo: Alearned Query Optimizer

En los estudios de Marcus et al. (2018) se nos muestran como el proceso de optimización de una consulta dentro de un sistema gestor de bases de datos (SGBD) puede

utilizarse como un mecanismo para la implementación de algoritmos o procedimientos aplicados al aprendizaje automático. En esencia, los autores identificaron que, por lo general, cuando un agente IA se conecta con un SGBD, toma una estructura predefinida por el usuario para poder controlarlo, generando un resultado predecible en donde hay un correcto funcionamiento de la automatización, pero hace que el proceso sea sumamente lento cuando el volumen de los datos es robusto.

Esto, debido a que repite dicha estructura o rutina predefinida para analizar cada registro que forma parte del conjunto de datos con el que está trabajando.

Por ello, los autores de este trabajo desarrollaron esta herramienta denominada “Neo” una IA que autogestiona los procesos de cargue, gestión y administración de datos, simulando un aprendizaje no supervisado, en donde el mismo agente de IA sea quien tome las decisiones más acertadas, para analizar y procesar datos, en base al mismo contexto definido por estos.

Es decir, que el mismo agente IA era el encargado de implementar rutinas de análisis de datos, de una formas más rápida y eficiente, dando como resultado un modelo predictivo de datos totalmente autónomo.

Así mismo, el sistema inteligente que se formaba con Neo era capaz de gestionar de forma exitosa grandes volúmenes de datos para que fueran procesados por un SGBD.

Como todo modelo de IA, Neo requería ser entrenado para que fuera útil. Pero sus autores se dieron cuenta que el proceso de entrenamiento era bastante extenso y, por ello, en los prototipos finales de Neo, optaron por hacer uso de un aprendizaje por refuerzo profundo, para que la misma IA fuera capaz de determinar qué consultas SQL eran las más óptimas ante un conjunto de datos en específico.

Dicho de otra forma, se podría decir que los autores de esta investigación hicieron que una máquina aprendiera a usar consultas SQL, mediante un modelo de aprendizaje humano,

basado en el método de ensayo-error, para que al final pudiera ser tan diestra en realizar consultas SQL únicamente leyendo e interpretando el contexto de un conjunto de datos.

Es entonces, donde se puede apreciar la relevancia de este estudio para el objeto de investigación, en donde la autogestión de consultas SQL por medio de IA se usa para aumentar el rendimiento y la escalabilidad de una base de datos frente a grandes volúmenes de datos.

Como dato técnico, el estudio especifica la utilización del SGBD relacional de PostgreSQL, para la visualización de los datos se utilizaron tecnologías tradicionales de una aplicación web, como lo son PYTHON con HTML y CSS3. Así mismo, el conjunto de datos utilizado era información de tipo demográfico, como lo son el género, edad y nacionalidad, de un grupo de personas en específico.

2.1.19. Balsa: Learning a Query Optimizer Without Expert Demonstrations

Este artículo es bastante similar al anterior, ya que también propone un modelo de aprendizaje por refuerzo para que sea el mismo modelo de IA el encargado de tomar las decisiones óptimas de administración de una base de datos.

No obstante, al comparar esta investigación con el trabajo de otros autores, se puede apreciar como Yang et al. (2022) plantean un modelo de aprendizaje autónomo dentro de los mismos agentes de inteligencia artificial, eliminando el factor humano en proceso de edición o cargue de información. Lo anterior permitió la implementación del aprendizaje profundo haciendo que el proceso de Machine Learning disminuyera sus sesgos cognitivos, como resultado de la autogestión de datos a manos de los modelos de inteligencia artificial.

Otros autores como Gandomi y Haider (2015) muestran en sus investigaciones todo el proceso que implica el proceso de recolección y limpieza de datos, hasta convertirlos en un conjunto de datos que permita extraer información valiosa, por medio de la Analítica de

Datos, enseñándonos la relevancia de este proceso de limpieza de datos aplicado a proceso de Machine Learning orientado al aprendizaje profundo.

En base a lo anterior, se puede apreciar como el desarrollo de Balsa se orienta que su principal objetivo sea la interacción directa entre IA y los datos alojados en una base de datos, para que se genere el aprendizaje necesario, en donde la IA sea capaza de manipular el SGBD sin problemas.

La finalidad de esta investigación busca sobrepasar los límites del rendimiento de una base de datos tradicional, convirtiéndola en un base de datos inteligente.

En donde se busca una verdadera optimización al 100 %, y en donde además no exista la intervención humana para los procesos de entrenamiento y aprendizaje máquina o Machine Learning, dando como resultado una aproximación bastante cercana a lo que se podría denominar un sistema autoentrenado, por medio de técnicas de aprendizaje autosupervisado.

Otros autores, como Kipf et al. (2019), muestran por medio de sus estudios alternativas más diferentes para llegar a ese estado de optimización total, en donde por medio del Deep Learning se obtiene un resultado similar al esperado. Pero con ciertas limitantes, llegando a la conclusión de que hace falta poco para llegar a ese punto en donde ya no sea necesaria la intervención humana en estos sistemas.

Esta postura también se ve reflejada en autores como Goodfellow et al. (2016), en donde también se propone un panorama similar al anterior, indicando que los sistemas autogestionados son el futuro de la IA. En donde, según el autor, la clave para poder llegar a ese punto es la implementación de nuevos algoritmos basados en el Deep Learning orientado a la implementación de procesos autogestionados, algo que por el momento sigue siendo un tema de estudio entre expertos en el área.

Por ello, la relevancia de esta investigación radica en un resultado concreto del desarrollo de una base de datos inteligente, orientado a sistemas basados en aprendizaje autosupervisado basado en diferentes componentes del Machine Learning.

2.1.20. Survey on Learnable Databases: A Machine Learning Perspective

Este estudio en concreto muestra un enfoque holístico acerca de las bases de datos inteligentes, en donde Zou et al. (2022), mediante la recopilación de diferentes fuentes académicas, permiten analizar y comprender la evolución del objeto de estudio, visto desde el punto de vista de diferentes autores. El estudio se conforma de la clasificación y/o categorización de varios artículos científicos, estudios técnicos y documentos especializados que permiten tener una mejor comprensión acerca de la aplicabilidad de las bases de datos inteligentes en el ámbito de la gestión de los datos en distintos escenarios.

En base a lo anterior, se puede apreciar que el objetivo principal de este estudio fue analizar, clasificar y documentar el impacto de esta tecnología en diferentes contextos o ámbitos de trabajo.

Si bien este trabajo no contiene ningún tipo de desarrollo práctico, nos da una perspectiva más amplia de cómo será la evolución de las bases de datos en un futuro cercano.

Indicando, al igual que otros autores, que los sistemas inteligentes llegarán a tal punto en donde ya no se requiera de la intervención humana para el desarrollo de tareas específicas.

Entre las principales conclusiones de esta investigación se define que efectivamente toda la información gestionada por una base de datos inteligentes da como resultado un aumento de rendimiento, a comparación de una base de datos tradicional.

Asimismo, esta investigación también tuvo muy presente otros trabajos académicos orientados a la documentación y desarrollo de prototipos funcionales del objeto de estudio, enfocados en la integración de inteligencia artificial aplicada a motores de sistemas gestores de bases de datos tradicionales.

En base a lo anterior, la revisión de la literatura propuesta en este estudio, se enfocó en analizar la posible correlación entre el nivel de automatización que pueden alcanzar las bases de datos inteligentes y el grado de intervención humana requerida para el funcionamiento de estas. Indicando que mientras más alto sea el nivel de automatización en este tipo de bases de datos, se genera una menor dependencia de intervención humana, dando como resultado la generación de sistemas autónomos totalmente funcionales y con márgenes de errores muy inferiores a comparación de los modelos tradicionales.

Los sistemas autónomos, son un desafío que ya se ha logrado resolver, pero la mayor dificultad que existe al momento de implementarlos, es el predecir su comportamiento cuando interactúan con varios conjuntos de datos heterogéneos. Debido a que esto forma sesgos de diferentes tipos en el componente de inteligencia artificial que poseen, dando como resultado, un grado de inexactitud en sus respuestas.

Teniendo en cuenta lo anterior, se indica que el gran aporte que hace este trabajo al objeto de estudio está en la recopilación de artículos, evidencias, historias y registros acerca de cómo las bases de datos inteligentes se aplican en entornos reales.

2.1.21. InferDB: In-Database Machine Learning Inference Using Indexes

Este artículo propone una perspectiva totalmente distinta a los estudios anteriores, en donde se especifica que los sistemas inteligentes administrados por agentes de IA pueden llegar a ser más útiles si se integran directamente dentro del mismo SGBD, es por ello que, según sus autores, Salazar et al. (2024), indican que InferDB es un modelo hibrido totalmente integrado en un SGBD y un modelo de IA.

Los autores también describieron a InferDB como un modelo de desarrollo nativo y totalmente libre de API´s, librerías o dependencias externas utilizadas normalmente para la comunicación de un agente inteligente con una base de datos. Dando como resultado una

disminución en los tiempos de ejecución de las transacciones generadas en las tareas de procesamiento de datos.

En base a lo anterior, lo que se propuso fue el desarrollo de un SGBD con inteligencia artificial integrada, dando como resultado una disminución en los tiempos de ejecución en un proceso de transacción de datos.

Esta investigación demuestra que el consumo de elementos externos para administrar una base de datos puede llegar a causar cierto grado de latencia entre componentes, debido al tiempo que toma un proceso comunicación entre dos o más elementos del sistema, cuando estos se encuentran en un entono real. Así mismo, el estudio también evalúa otros factores, como la incompatibilidad entre diferentes tecnologías o limitaciones de hardware y software en entornos de trabajo complejos, indicando que estos factores pueden ser bastante nocivos para un proceso de comunicación entre un agente IA y el SGBD.

InferDB nace como una propuesta para una implementación totalmente alternativa de una base de datos inteligente, en donde se propone un concepto de indexación inteligente, el cual combina la teoría de grafos por medio de un modelo de árboles binarios, presente en los modelos de IA y la indexación tradicional que es visible en cualquier SGBD. Dicho de otra forma, lo que se busca es combinar estos dos conceptos para dar como resultado un SGBD con un componente interno de inteligencia artificial totalmente integrado.

Aunque InferDB llegó a ser únicamente un prototipo funcional, abre la posibilidad de la creación de bases de datos totalmente autónomas y autooptimizables, lo que permitiría que este tipo de bases de datos si tengan la capacidad de analizar y entender datos totalmente heterogéneos sin necesidad de la intervención humana o de datos preetiquetados.

3. Marco empírico

En el presente capítulo se definen la hipótesis y los objetivos de este trabajo académico, con la finalidad de dar a entender el rumbo elegido para el desarrollo de este. Luego de ello, pasamos a definir el diseño y tipo de investigación, en donde se especifica en detalle la metodología de investigación seleccionada.

Posteriormente, se procede a hablar acerca de la población y muestra escogida, para definir y aclarar el origen de los datos recolectados a lo largo de todo el proceso de desarrollo, así mismo, también se especifican los instrumentos de evaluación y los resultados obtenidos.

Y, por último, se pasa a hablar acerca de la propuesta y diseño de la solución implementada, como un resultado concreto de todo lo desarrollado en base al objeto de estudio.

3.1. Hipótesis general

Con base en la observación de las nuevas tecnologías emergentes, se ha podido apreciar que las tendencias actuales en el sector TI buscan a toda costa disminuir costos operativos por medio de procesos de automatización mediante el uso de IA, en donde se pueden ver avances tecnológicos de todo tipo gracias a la IA.

Dentro de dichos avances tecnológicos podemos encontrar nuestro objeto de estudio, el cual trata sobre el desarrollo e implementación de bases de datos inteligentes, para el desarrollo autónomo de procesos de gestión y administración de grandes volúmenes de datos.

Y es allí cuando surge una interrogante, en donde nos cuestionamos acerca del impacto que han tenido estas bases de datos en el sector empresarial.

Por ello, y con la finalidad de darnos una idea o aproximación sobre la solución a esta interrogante, se hace necesario comparar el rendimiento que estas bases de datos tienen con respecto a una base de datos tradicional.

En base a lo anterior, pasamos a recordar la hipótesis de investigación propuesta en este trabajo:

"La utilización de un prototipo basado en inteligencia artificial para transformar consultas formuladas en lenguaje natural en consultas SQL produce una reducción en el tiempo de respuesta frente al método tradicional de consulta SQL sobre una base de datos relacional".

Así mismo, nuestra hipótesis nula (H_0), sería la siguiente:

"La utilización del prototipo basado en inteligencia artificial no produce una diferencia estadísticamente significativa en el tiempo de respuesta respecto al método tradicional de consulta SQL".

En base a lo anterior, también se plantea una hipótesis alternativa (H1), siendo esta:

"La utilización del prototipo basado en inteligencia artificial produce una diferencia estadísticamente significativa en el tiempo de respuesta respecto al método tradicional de consulta SQL".

Teniendo en cuenta el planteamiento de la hipótesis de investigación, se procede a representar las variables de investigación dentro de la **Tabla 1**, en donde se evalúan aspectos generales como la integración del modelo con la base de datos relacional, el rendimiento del Sistema gestor de bases de datos, el nivel de eficiencia de la base de datos inteligente al momento de gestionar grandes volúmenes de datos, el nivel de predicción de las consultas SQL por parte del agente de IA y la percepción de la población objetivo acerca de las bases de datos inteligentes.

Tabla 1.

Tabla de variables de investigación.

Variable	Tipo de variable	Definición conceptual	Indicadores	Escala de medición
Integración de IA con Bases de datos.	Independiente	Es el nivel de acoplamiento y/o funcionamiento que tiene un agente de IA, al momento de manipular un sistema gestor de bases de datos.	Presencia del agente de IA. Capacidad de autoajuste de consultas.	Nominal
Rendimiento del SGBD.	Dependiente	Es el tiempo que le toma al SGBD realizar una operación típica de la gestión y administración de los datos.	Tiempo medio de ejecución por consulta.	De razón
Eficiencia de procesamiento de grandes volúmenes de datos.	Dependiente	Representa el grado de capacidad del sistema para analizar, almacenar y procesar grandes volúmenes de datos.	Comportamiento de la escalabilidad del sistema.	De razón
Nivel de predicción de consultas SQL.	Dependiente	Es el grado de precisión, que tiene el sistema para determinar y predecir una consulta SQL, en base al contexto de los datos que está recibiendo.	Porcentaje de error al crear y desarrollar las consultas SQL.	De intervalo
Perspectiva de personal de TI sobre bases de datos inteligentes.	Dependiente	Visión u opinión de las personas que trabajan en el sector de TI, acerca de las bases de datos inteligentes.	Escala liker, para medir el nivel de conocimiento de un usuario.	Ordinal

Nota. Elaboración propia.

3.2. Objetivos General

"Desarrollar y evaluar un prototipo funcional basado en inteligencia artificial que permita transformar consultas formuladas en lenguaje natural a consultas SQL sobre una base de datos relacional, comparando su desempeño con el método tradicional mediante indicadores de tiempo de respuesta y exactitud.".

Lo que se busca principalmente con este objetivo general es el desarrollo de una base de datos inteligente, para comenzar a desarrollar cierto tipo de pruebas comparativas con una base de datos tradicional. Así mismo, esto se hace con la finalidad de ir recolectando datos muy concretos acerca de comportamientos, tiempos de ejecución y nivel de precisión de una base de datos inteligente cuando trabaja con una gran cantidad de datos, para así darnos una idea más aproximada acerca de las principales diferencias que pueden existir entre un modelo de bases de datos inteligente y un modelo de bases de datos tradicional. Dicho esto, pasamos a hablar acerca de los objetivos específicos de este trabajo.

3.3. Objetivos específicos

- Diseñar la arquitectura de un prototipo funcional que integre un modelo de inteligencia artificial con un sistema gestor de bases de datos relacionales para la generación de consultas SQL a partir de lenguaje natural.

- Desarrollar e implementar el prototipo propuesto, integrando el procesamiento de consultas en lenguaje natural, la generación de instrucciones SQL y su ejecución sobre una base de datos relacional.

- Evaluar el comportamiento del prototipo mediante un conjunto de consultas experimentales, considerando como indicadores el tiempo de respuesta y la exactitud de las consultas generadas.

- Comparar los resultados obtenidos mediante el prototipo basado en inteligencia artificial con los obtenidos mediante el método tradicional de consulta SQL, determinando las diferencias en tiempo de respuesta y porcentaje de mejora.

Los cuatro objetivos específicos anteriormente definidos constituyen una guía clara y estructurada para alcanzar el objetivo general del presente trabajo de investigación, en donde se busca aplicar los fundamentos propuestos en la ingeniería del software al desarrollo de dichos objetivos.

Adicional a ello, aplicando la teoría propuesta en el ciclo de vida del desarrollo de software, se logra estructurar los objetivos específicos de tal forma, que estén acordes al proceso de documentación, desarrollo e implementación de un prototipo funcional del objeto de estudio, con la finalidad de generar una serie de elementos que permitan validar la hipótesis de investigación de una forma más sencilla.

3.4. Diseño y tipo de investigación

En este apartado se define que la investigación realizada, con respecto a nuestro objeto de estudio, tiene un enfoque empírico de carácter experimental, con una metodología de investigación cuantitativa de carácter comparativo, delimitado y aplicado únicamente a bases de datos relacionales.

Se indica que esta investigación tuvo un enfoque empírico, porque efectivamente se realizaron pruebas exhaustivas orientadas a verificar la hipótesis general, en donde por medio de la observación y la experimentación con las consultas SQL hechas en diferentes escenarios se llega a resultados concretos.

Así mismo, cuando señalamos que la metodología de investigación es cuantitativa, es porque se utilizaron los tiempos de respuesta obtenidos durante la evaluación experimental del prototipo, un total de diez consultas ejecutadas mediante dos escenarios: el prototipo basado en inteligencia artificial y el método tradicional mediante consultas SQL.

Debido a que las mismas consultas fueron evaluadas en ambos escenarios, las observaciones fueron consideradas como datos pareados.

Inicialmente se realizó un análisis descriptivo mediante el cálculo de la media, mediana, desviación estándar, valores mínimo y máximo. Asimismo, se calcularon las diferencias de tiempo entre ambos escenarios y el porcentaje de mejora para cada consulta.

Posteriormente, se verificó la normalidad de las diferencias mediante la prueba de Shapiro-Wilk. En consecuencia, también se aplicó una prueba t de Student para muestras relacionadas, con el propósito de determinar si existía una diferencia estadísticamente significativa entre los tiempos de respuesta de ambos escenarios.

3.5. Población y muestra

La población objetivo está representada por todos los profesionales del sector de TI que específicamente han trabajado con bases de datos en su entorno laboral, así como también de estudiantes de pregrado de carreras tecnológicas, en la ciudad de Villavicencio (V/cio), ciudad capital del departamento del Meta, ubicado en Colombia, siendo esta la región donde se llevó a cabo este trabajo académico.

V/cio es una zona aleñada a Bogotá (ciudad capital de Colombia), y se caracteriza por ser una zona que basa su economía en el turismo y en el desarrollo agroindustrial, por lo que la demanda de profesionales del sector de TI suele ser un poco baja. Ocasionando que las oportunidades de crecimiento tecnológico, sean cada vez más escasas.

En base a lo anterior se pretende conocer un poco más a fondo la opinión de las personas del sector de TI de la región, en base al objeto de estudio, con la finalidad de entender un poco más acerca del impacto que podría llegar a tener las bases de datos inteligentes en regiones como esta.

En base a lo anterior, la muestra elegida se basó en grupos más pequeños de individuos, siendo dos empresas del sector tecnológico de la región y dos universidades con carreras tecnológicas.

La razón de no solo delimitar la población de estudio a únicamente profesionales del sector de TI se basa en ampliar un poco más la perspectiva de nuestro trabajo y comenzar a ver diferencias entre la visión que tiene un profesional y la de un estudiante de pregrado, con relación al objeto de estudio.

Es por ello que al final la muestra incluye estos dos elementos: profesionales del sector de TI y estudiantes de pregrado de carreras enfocadas en el área de TI, de únicamente dos universidades.

Así mismo, la muestra elegida cuenta con una serie de características importantes que permiten integrarla perfectamente con el objeto de estudio, en donde:

1. Esta muestra es la única que tiene el conocimiento técnico lo suficientemente avanzado para entender y manipular una base datos, teniendo en cuenta el lugar donde se llevó a cabo esta investigación, a diferencia de un usuario estándar o una persona normal. Es decir, que los individuos que conforman la muestra están en la capacidad de realizar una consulta SQL o de manipular una base de datos, ya que se consideran un usuario experto en el área, o que por lo menos tiene nociones básicas para entender el tema.

2. La muestra elegida en sí representa el mismo entorno donde interactúan las bases de datos, ya que es en las empresas y en las universidades donde están presentes, además de ser el lugar idóneo para aplicar las pruebas técnicas del objeto de estudio.
3. Esta muestra permite aplicar de forma conjunta teoría y práctica acerca del tema, dando como resultado un feedback que nos permita ver nuestro objeto de estudio desde otras perspectivas.

3.6. Instrumentos de evaluación

Para este apartado se tuvieron en cuenta las variables de investigación previamente definidas en la tabla 1- tabla de variables de investigación, en donde por cada una de las variables se diseñó un instrumento de evaluación correspondiente, partiendo principalmente de los siguientes elementos complementarios:

- Un script automatizado para la medición de tiempos.
- Configuración de logs para llevar un registro de los resultados de las pruebas.
- Una encuesta para medir la percepción de la muestra estudiada con respecto al objeto de estudio.

Siendo los tres elementos anteriores resultados que se generaron durante la documentación del proceso de desarrollo del prototipo, dando origen a los instrumentos de evaluación que se presentan a continuación:

Variable 1-Integración de IA con Bases de datos.

Para esta variable se midió el nivel de acoplamiento de un agente IA, por medio de la ejecución de consultas SQL, en donde se determinaba el grado de exactitud alcanzado por este.

Para ello se comenzaba con la toma de tiempos al momento de realizar una consulta SQL haciendo uso del agente IA por medio de NQL, luego se realizaba la misma consulta

SQL dentro del SGBD, para determinar el grado de exactitud de los resultados generados por el agente AI, esperando que efectivamente, si todo era correcto, ambos resultados debían coincidir.

Variable 2-Rendimiento del SGBD.

En el caso concreto de esta variable, se diseñaron una serie de consultas SQL muy específicas para poner a prueba el rendimiento del prototipo funcional de base de datos inteligente. Así mismo, también fue necesario el uso e intercambio de diferentes conjuntos de datos, los cuales representan la carga de trabajo a la que debe ser sometido el objeto de estudio, con el fin de medir y comparar los tiempos de ejecución.

Variable 3-Eficiencia de procesamiento de grandes volúmenes de datos.

Para esta variable, se usaron herramientas y script automatizados que permitieron simular el comportamiento típico de las transacciones que maneja una base de datos, y con ello evaluar la escalabilidad del sistema.

Variable 4-Nivel de predicción de consultas SQL.

En relación con esta variable, se generaron archivos de salida o logs del sistema cada vez que se le hacia una petición NQL al prototipo implementado, los cuales posteriormente eran almacenados dentro del mismo SGBD utilizado, para poder determinar si los resultados entre una petición NQL y una consulta SQL eran los esperados.

Variable 5-Perspectiva de personal de TI sobre bases de datos inteligentes.

Para esta variable se optó por diseñar una encuesta con 26 preguntas conformada con preguntas abiertas y cerradas, así como también una escala Likert para determinar la categorización de conocimiento de un usuario con respecto a las bases de datos inteligentes.

La encuesta conformada por las 26 preguntas divididas en 6 secciones principales aborda datos demográficos, datos de conocimiento técnico y datos de perspectivas u opiniones orientado a un leve componente cualitativo acerca del del objeto de estudio.

3.7. Descripción general de la solución propuesta

En esta sección se especificó el diseño de la solución planteada, mediante el desarrollo de una base de datos inteligente conformada por un agente de IA externo y una base de datos relacional hecha con el SGBD de MySQL, mediante el consumo de una API. El objetivo por el cual se optó por el desarrollo de esta solución fue para facilitar el desarrollo de pruebas que permitieran evaluar las ventajas y desventajas las bases de datos inteligentes frente a modelos tradicionales.

Así mismo, el desarrollo de la solución propuesta se divide en dos partes:

Parte 1-Configuración del entorno tradicional de bases de datos.

En donde por medio del uso del lenguaje SQL en consola de un SGBD, se propone realizar una serie de pruebas, basadas en consultas básicas-intermedias, para hallar datos específicos dentro de un gran conjunto de información almacenada en una base de datos.

Parte 2-Configuración del entorno inteligente de bases de datos.

Para poder realizar esta tarea se utilizaron dos entornos de prueba distintos, uno de ellos utilizando una base de datos relacional y el otro haciendo uso del prototipo de base de datos inteligente. Con el propósito de registrar y comparar los datos obtenidos en ambos escenarios de prueba y así obtener información precisa mediante el uso de indicadores y métricas que nos permitan rechazar o refutar la hipótesis de investigación.

Entre las principales fuentes registradas para obtener los datos de cada escenario de prueba, se encuentran los *test* de software realizados como lo fueron: pruebas de caja negra, caja blanca, caja gris, de estrés y de humo. Así como también registro del sistema y los archivos de salida del SGBD de MySQL, los cuales fueron analizados y procesados en base al tiempo de ejecución que registraban en cada consulta SQL de prueba.

3.7.1. Descripción detallada de cada fase

En la **Tabla 2** se describe la división y agrupación de todas las actividades del proyecto en diferentes actividades. En donde se inicia por el planteamiento del problema, para delimitar el alcance de la investigación, luego se avanza con la revisión bibliográfica y así construir los fundamentos del marco teórico, después de ello se procede con el diseño metodológico y el diseño de los instrumentos de investigación para así definir el enfoque y tipo de investigación que se va a usar, después de esta parte se procede con la construcción del prototipo funcional de bases de datos inteligente, y una vez alcanzado el punto anterior se prosigue con el desarrollo de un conjunto de pruebas para dar inicio a la etapa de recolección de datos, posteriormente se culmina con un breve análisis estadístico que permita formular una serie de indicadores y de métricas que permitan arrojar resultados concretos para validar la hipótesis de investigación, por ultimo se cierra todo este ejercicio redactando los hallazgos, conclusiones y recomendaciones finales.

Tabla 2.

Cronograma de actividades

Actividad	Mes 1	Mes 2	Mes 3	Mes 4
Planteamiento del problema	✓			
Revisión bibliográfica	✓	✓		
Diseño metodológico		✓		
Diseño de instrumentos		✓		
Validación de instrumentos		✓	✓	
Desarrollo del prototipo		✓	✓	
Recolección de datos			✓	
Análisis estadístico				✓
Redacción de resultados				✓
Conclusiones				✓

Nota. Elaboración propia.

En base al cronograma de actividades propuesto, también se genera un diagrama de Gantt como complemento al cronograma propuesto y el cual se ha detallado en el Apéndice A de este trabajo.

3.8. Resultados

En este apartado se procede a explicar en detalle todo el proceso de desarrollo realizado a lo largo de toda la investigación, y en donde por medio de la presentación de las pruebas pertinentes del proceso llegamos a confirmar o rechazar la hipótesis de investigación planteada desde un inicio.

Así mismo, los resultados están divididos en resultados experimentales y resultados perceptuales.

Los resultados experimentales corresponden a todo el desarrollo de pruebas técnicas de las consultas NQL y SQL ejecutadas por medio del prototipo de base de datos inteligente.

Por su parte, los resultados perceptuales corresponden al análisis estadístico de la percepción o visión de la muestra elegida para el desarrollo de las pruebas técnicas llevadas a cabo en trabajo académico, en donde, por medio de un formulario utilizado como herramienta de recolección de datos, pasamos a tabular las respuestas de dicho formulario con la finalidad de hallar tendencias u opiniones con respecto al objeto de estudio.

Por último, con la evidencia mostrada a partir de los resultados, pasamos a confirmar o refutar nuestra hipótesis.

3.8.1. Resultados experimentales

A partir de este punto, definimos las herramientas necesarias para el desarrollo de nuestro objetivo general, así como también las especificaciones de diseño de la aplicación, teniendo como punto de partida algunas de las etapas establecidas en el ciclo del desarrollo de software, esto debido a que el objetivo propuesto solo abarca un desarrollo a nivel de

prototipo funcional, por lo que las etapas de despliegue y mantenimiento no se tuvieron en cuenta.

Dicho esto, se categorizan las etapas restantes del ciclo en tres fases, las cuales se han definido de la siguiente manera:

- Fase 1 (planeación, análisis y diseño).
- Fase 2 (desarrollo e implementación).
- Fase 3 (pruebas).

Mientras que la fase 1 abarca todo el proceso de levantamiento de requerimientos, la fase 2 se enfoca en mostrar el proceso de codificación y funcionalidades del sistema, mientras que la fase 3 es donde se definen las pruebas específicas que se llevaron a cabo.

Fase 1 (planeación, análisis y diseño).

Se comienza con todo lo concerniente a la documentación del proceso de desarrollo de software del prototipo funcional de bases de datos inteligente, en donde por medio del uso de técnicas de modelamiento de software se pasa a documentar todo el proceso realizado.

Analizando más a fondo el objeto de estudio, se inicia con la documentación de los requerimientos funcionales y no funcionales para el desarrollo del prototipo funcional.

Esto, con el objetivo de definir el alcance del proceso de desarrollo del prototipo, para así mismo definir el diseño y las funcionalidades de este.

En base a ello, se delimita de forma clara y concisa el alcance que tiene el desarrollo de dicho prototipo funcional, resumiendo todo el proceso de desarrollo software en los siguientes tres puntos:

Alcance del desarrollo.

1. Ejecución de consultas SQL, por medio de lenguaje natural.
2. Implementación y optimización de consultas SQL por medio de un agente de IA.

3. Realizar comparativos entre tiempos de ejecución entre consultas SQL hechas a un base de datos inteligente vs una base de datos tradicional.

Teniendo en cuenta los tres puntos anteriores, y siguiendo un proceso tradicional de desarrollo de software, se procede con la iniciación de una etapa de levantamiento de requerimientos, por medio de la construcción de tres narrativas que representan las historias de usuario.

Historias de usuario.

1. Yo, como usuario, quiero preguntarle algo específico a la aplicación, para que me dé una respuesta, en base al conjunto de datos que estamos trabajando.
2. Yo, como desarrollador, quiero que la aplicación tome y registre los tiempos de ejecución de cada consulta hecha por un usuario, para posteriormente realizar comparaciones entre estos.
3. Yo, como desarrollador, quiero que la aplicación me muestre en pantalla la consulta SQL correspondiente a la consulta NQL hecha por el usuario, para así validar que la interpretación del agente IA y sus resultados sea correctos.

Las tres historias de usuario anteriores delimitan todo el conjunto de actividades a desarrollar para la implementación del prototipo de base de datos inteligente, en donde se da continuidad al proceso por medio de la definición de los requerimientos funcionales y no funcionales, dando como resultado el siguiente resumen de requerimientos:

Requerimientos funcionales (RF).

RF-01: el prototipo debe manejar un componente de IA para realizar consultas SQL, por medio de peticiones hechas en lenguaje natural.

RF-02: el prototipo debe estar en la capacidad de mostrar la consulta SQL usada en la base de datos, con la finalidad de poder comparar los resultados de una consulta SQL, hecha de forma manual o directamente desde la base de datos.

RF-03: el prototipo debe contar con un script o un componente que muestre los tiempos de ejecución de cada consulta hecha a la base de datos, con la finalidad de tomar registros y así comparar tiempos de ejecución.

Requerimientos no funcionales (RNF).

RNF-01: el prototipo debería tener un tiempo promedio de ejecución de entre 5 a 10 segundos, y mostrar una alerta en caso de que algo salga mal.

RNF-02: el prototipo debe soportar grandes volúmenes de datos, para demostrar qué tan escalable es cuando se le realizan peticiones complejas, como operación JOIN entre más de dos tablas.

RNF-03: el prototipo debe contar con una interfaz gráfica intuitiva que le permita a cualquier usuario entender la usabilidad de este.

Una vez definidos los requerimientos funcionales y no funcionales involucrados en el proceso de desarrollo de software, se procede con el modelado del sistema, haciendo uso del lenguaje de notación gráfica, conocido como Unified Modeling Language (UML), con la finalidad de documentar el comportamiento y estructura del prototipo funcional de base de datos inteligente, en donde se define la construcción de dos diagramas estructurales y dos diagramas comportamentales, especificados de la siguiente manera:

Diagramas comportamentales:

- Diagrama de casos de uso.
- Diagrama de secuencias.

Diagramas estructurales:

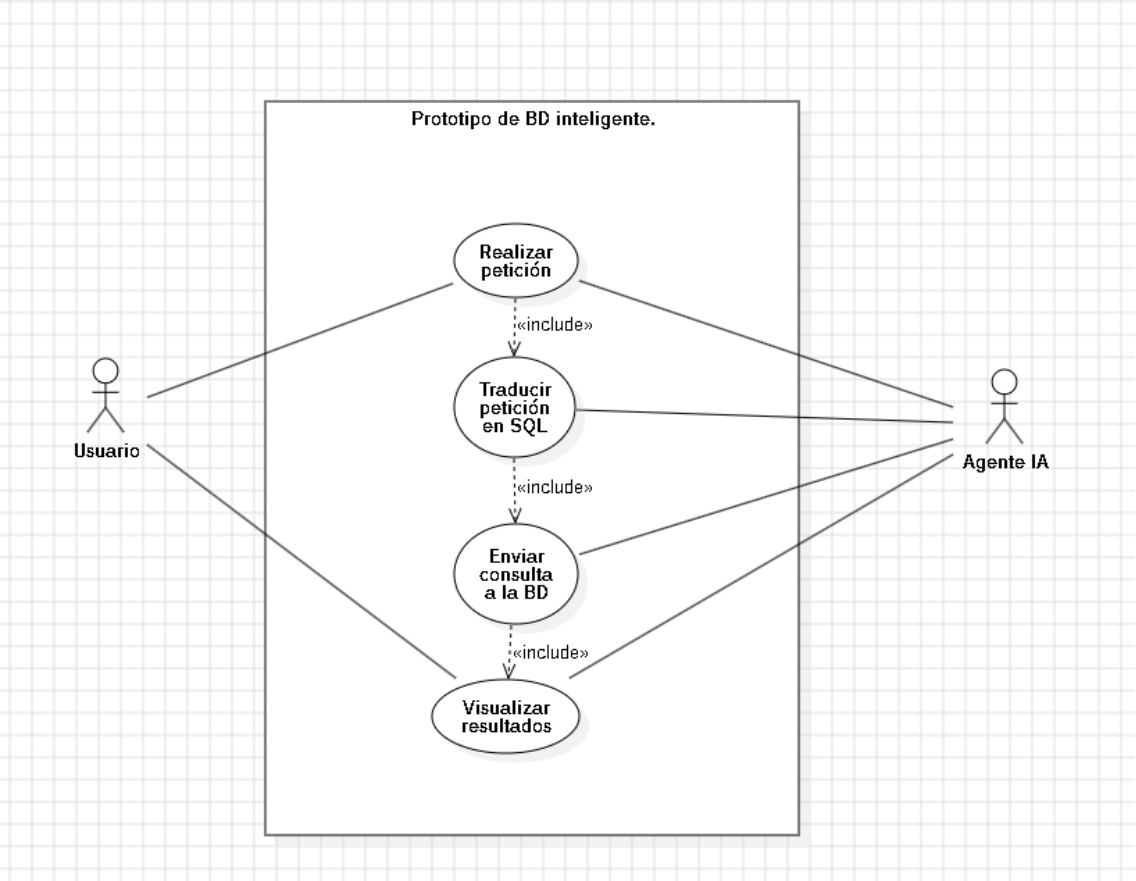
- Diagrama de clases.
- Diagrama de despliegue.

Adicional a ello, se han documentado los diagramas de casos en uso en un formato estándar, para complementar la documentación de estos.

En la **Figura 1** se muestra el caso de uso general, en donde se explica el comportamiento que debe tener el prototipo cuando recibe una petición NQL. Así mismo, también se representa la función que cumple el agente IA al momento de interactuar con el sistema.

Figura 1.

Diagrama de caso de uso 01 Realizar petición



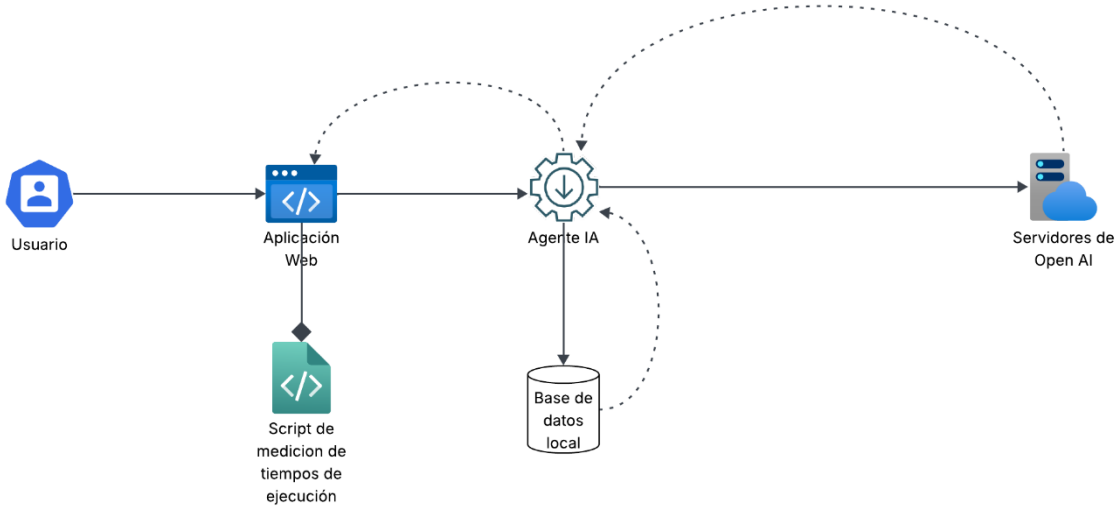
Nota. Elaboración propia.

Con lo que respecta al resto de los diagramas y el resto de formatos de documentación, se han colocado en la sección de anexos, debido a su nivel de detalle y tamaño. Una vez terminada la documentación de la estructura y comportamiento de la aplicación por medio de UML, se procede a elaborar un diagrama de red.

En la **Figura 2** se pueden apreciar los componentes de dicho diagrama, haciendo énfasis en su arquitectura y en la interacción que existe entre cada elemento, así como también el rol que cumple el usuario y el agente de IA dentro de la solución planteada.

Figura 2.

Diagrama de arquitectura de red



Nota. Elaboración propia.

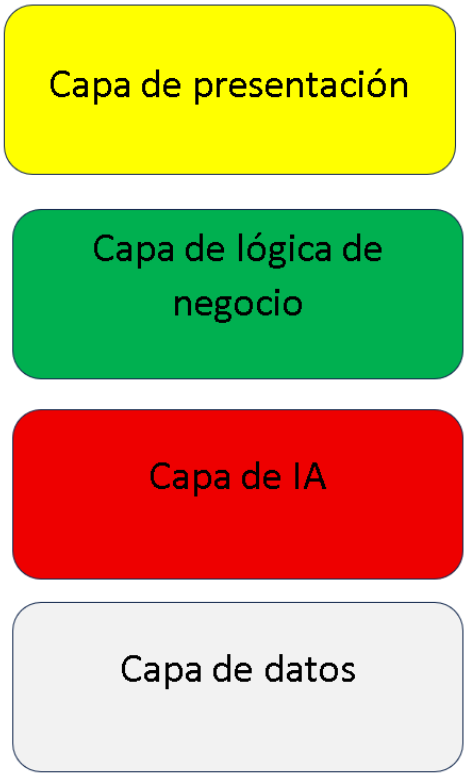
Como se puede apreciar en el diagrama anterior, la solución propuesta para el desarrollo de este prototipo funcional de bases de datos inteligente abarca la implementación de una aplicación web, la cual se comunica con un agente de IA por medio de una API, permitiéndole al agente IA la manipulación de una base de datos relacional hecha en MySQL.

En donde, por medio de la escritura de una consulta NQL por parte de un usuario, se obtienen el conjunto de datos específicos que se encuentren almacenados dentro de la base de datos utilizada. Una vez definida la arquitectura de red para la solución planteada, se procede con la definición de la arquitectura de software.

En la **Figura 3** se muestra que se escogió una arquitectura de software por capas, en donde se integra una capa extra, la cual corresponde al agente de IA que interviene en el proceso.

Figura 3.

Especificación de la arquitectura software.



Nota. Elaboración propia.

En este Capa de IA, es donde se definen los componentes de conexión entre la aplicación web y el agente de IA, encargado de acceder a la capa de datos, mediante el consumo de una API KEY.

Así mismo, en la capa de presentación están todas las tecnologías que le permiten ver al usuario la aplicación que se comunica con el agente de IA.

Por su parte, la capa lógica contiene todos los elementos necesarios para garantizar la comunicación entre la interfaz gráfica de la aplicación y el agente IA, así como también la estructura lógica de las reglas de negocio que representan las funcionalidades software en base a los requerimientos funcionales y no funcionales previamente establecidos.

Por último, en la capa de acceso a datos es donde están definidas todas las instrucciones necesarias para obtener el acceso a los datos de prueba que se están manejando, mediante una conexión estándar por medio de credenciales de acceso a una base de datos (host, usuario y contraseña).

Una vez definido el diseño, la estructura y las tecnologías usadas para el desarrollo del prototipo, así como también la arquitectura de software a implementar, se procede a modelar la base de datos y a definir el conjunto de datos con los que se va a trabajar.

Es por ello, que se optó por hacer uso de datos abiertos, los cuales son grandes conjuntos de datos que están disponibles de forma pública y sin costo alguno. En el contexto de Colombia, al igual que en muchos otros países, los datos abiertos se encuentran en plataformas gubernamentales donde se pueden descargar sin ningún costo, restricción legal o de derechos de autor, y cuya finalidad es dar cumplimiento a políticas de transparencia garantizando el acceso público a la información.

En base a lo anterior, se ha seleccionado un conjunto de datos denominado “Admitidos Estudiantes Unillanos”, el cual trata sobre la lista de aspirantes periodos 2021-2025.

Este conjunto de datos abarca información sumamente importante acerca de las personas que han sido admitidas en la universidad de los Llanos, aclarando que principalmente se trata de datos socio demográficos y de selección de carreras, sin comprometer ningún tipo de información personal. Con el objetivo de visualizar la estructura

de los datos del *dataset*, se ha colocado una representación gráfica de estos, en el apéndice C de este trabajo.

En la **Figura 4** se procede a mostrar la plataforma gubernamental de datos abiertos, con el objetivo de dar un contexto visual acerca de la fuente escogida para la obtención del conjunto de datos utilizado en este trabajo.

Figura 4.

Sitio web de datos abiertos del gobierno de Colombia



Nota. Captura de pantalla del sitio web www.datos.gov.co. Elaboración propia.

Una vez elegido el *dataset* correspondiente, para la realización de pruebas, se procede con el diseño e implementación de la base de datos con la que se va a trabajar.

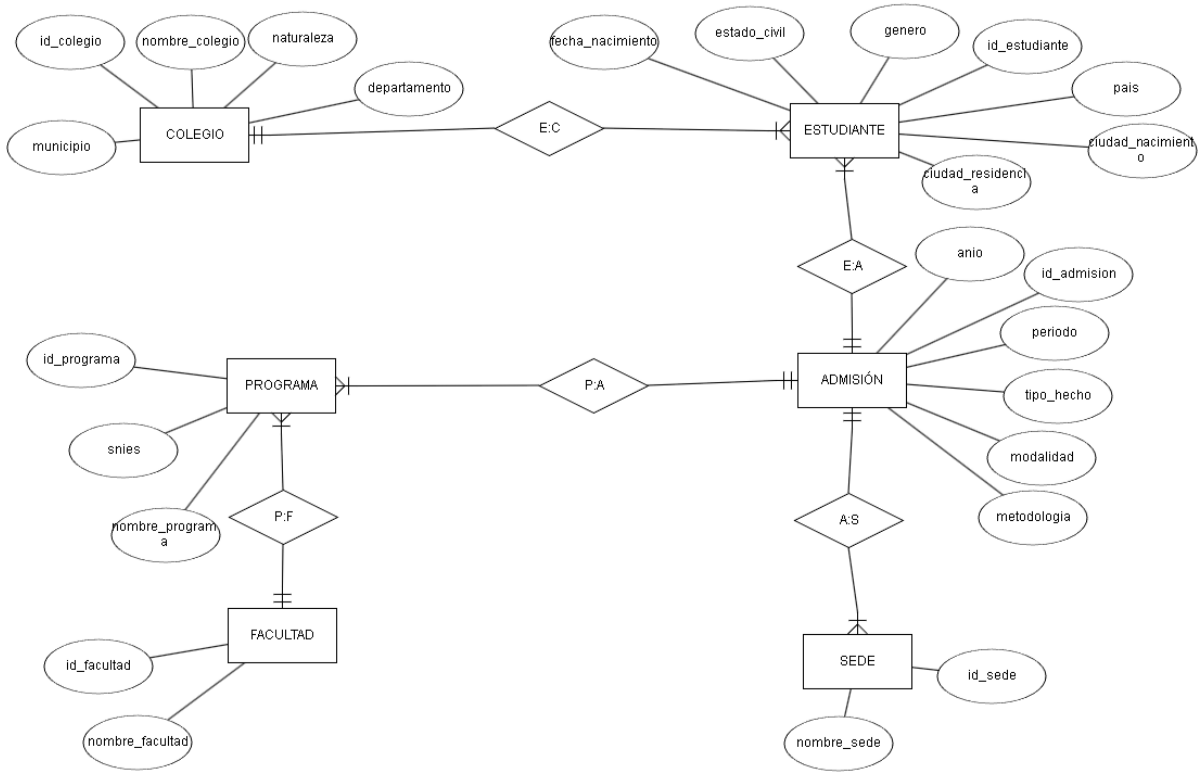
Iniciando con el desarrollo del diagrama entidad relación con notación de Peter Cheng, en donde se han definido seis entidades relevantes junto con sus respectivos atributos

y relaciones, donde también se puede apreciar los valores de cardinalidad y nivel mandatorio por cada una.

En la **Figura 5** se muestra la elaboración de dicho diagrama, en donde se han definido seis entidades con un promedio de entre 3 a 7 atributos, identificando plenamente componentes de cardinalidad en cada relación entre entidades.

Figura 5.

Diagrama entidad relación de la base de datos



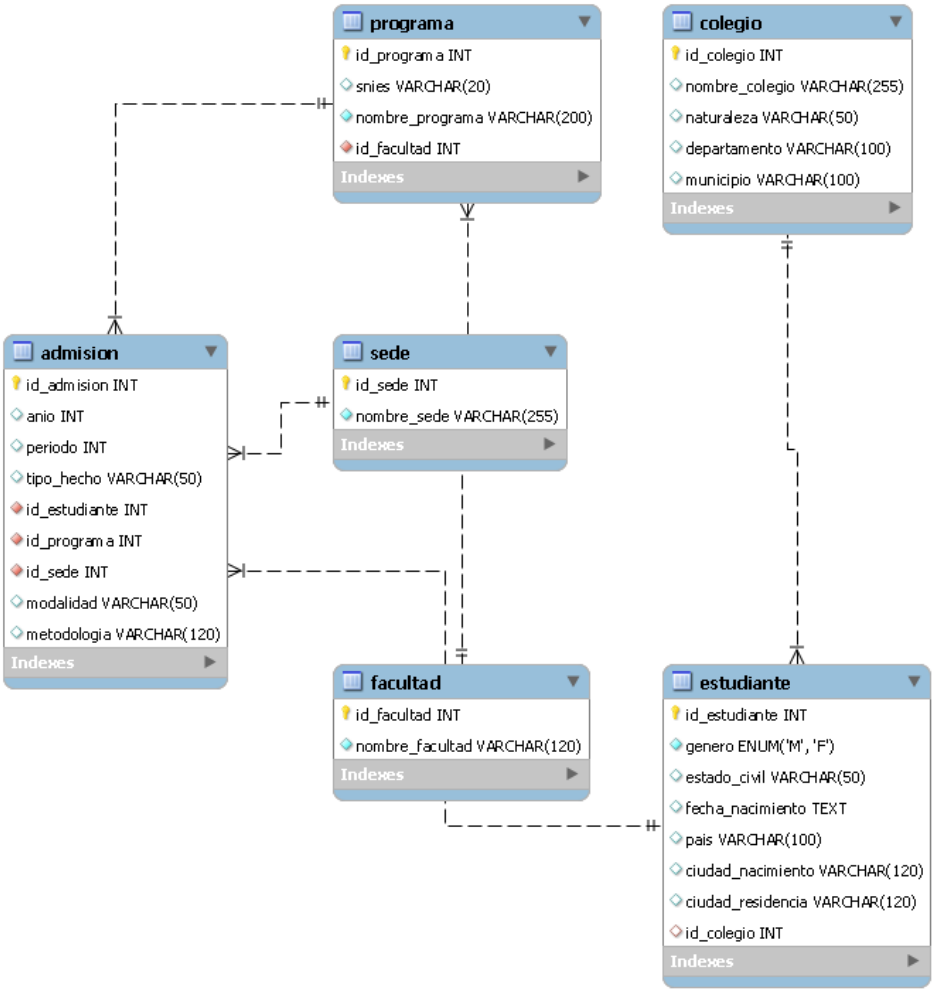
Nota. Elaboración propia.

Una vez definidas las entidades, los atributos de cada entidad y las relaciones entre entidades de la base de datos, se continua con el proceso de modelado de base de datos, por medio del diseño del modelo relacional, en donde se puede apreciar de manera más formal la estructura del diagrama anterior mediante un nivel de detalle más alto.

En la **Figura 6** se muestran los mismos elementos construidos en el diagrama entidad relación, dando a entender que deben estar acorde para conservar la coherencia en el proceso de desarrollo de la base de datos. Adicional a ello, los atributos ahora pasan a tener un tipo de dato definido, con el propósito de facilitar la programación de la base de datos y, así mismo, las relaciones entre entidades han pasado a convertirse en llaves foráneas, indicando que se está haciendo uso del modelo relacional para comunicar las diferentes tablas que componen el esquema de base de datos.

Figura 6.

Modelo relacional de la base de datos.



Nota. Elaboración propia.

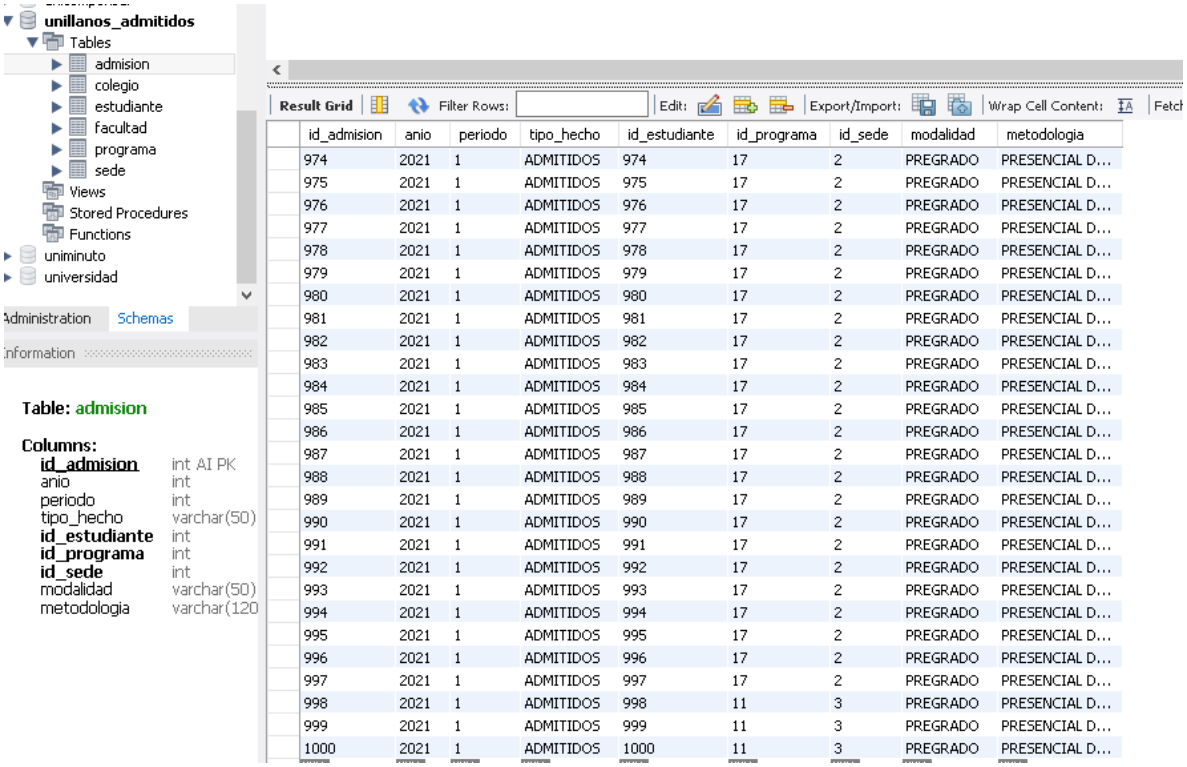
Culminando con el desarrollo del modelo relacional de base de datos, se procede con la generación del diccionario de datos, siendo este uno de los entregables más importantes en este ámbito de desarrollo, debido a que contiene toda la estructura de los datos que residen dentro de una base de datos. Sin embargo, su nivel de detalle es bastante extenso, por lo que el diccionario de datos se ha colocado dentro del apéndice D de este trabajo.

Continuando con el proceso de desarrollo de la base de datos, se procede con la construcción de la base de datos en MySQL, la cual se nombró como “unillanos_admitidos”.

En la **Figura 7** se muestra una captura de pantalla del sistema gestor de bases de datos utilizado, en donde también se puede apreciar la inserción de datos correspondientes al *dataset* seleccionado para realizar las pruebas.

Figura 7.

Inserción de los datos del dataset en MySQL



Nota. Captura de pantalla del SGBD de MySQL. Elaboración propia.

Para el proceso de inserción de los datos del dataset en la base de datos, se procede a desarrollar un script en Python que de forma automática realice esta tarea. En el apéndice E de este trabajo se ha colocado parte del script de inserción de datos implementado.

Una vez poblada la base de datos de prueba, se procede con el diseño conceptual de la interfaz gráfica de usuario y posteriormente se continua con el desarrollo del prototipo.

Para fines prácticos, se ha colocado el diseño de los mockups en el apéndice F, mostrando directamente el resultado del desarrollo de la interfaz gráfica.

En la **Figura 8** se muestra la interfaz gráfica de usuario de la aplicación que permite la comunicación con la base de datos inteligente, la cual se compone de una caja de texto para recibir las consultas NQL por parte del usuario, una pequeña guía de instrucciones que se visualizan al utilizar la aplicación, para que cualquier usuario pueda utilizar este sistema y un cuadro de resultados para poder apreciar la conversión de la consulta NQL en SQL, incluyendo el conjunto de datos relacionados con la consulta hecha por el usuario.

Figura 8.

GUI del prototipo de bases de datos inteligente



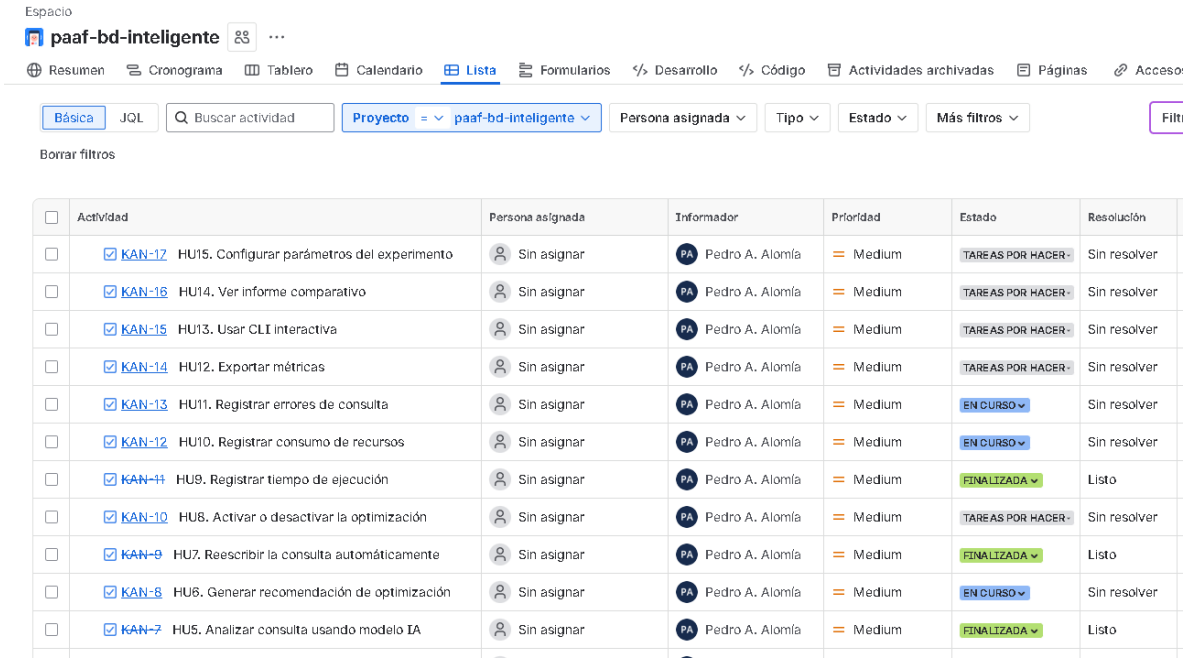
Nota. Captura de pantalla de la aplicación de bases de datos inteligente. Elaboración propia.

Una vez terminado el desarrollo de la interfaz gráfica de usuario, se procede a aplicar las pautas establecidas en el marco de trabajo Scrum, con la finalidad de desarrollar la parte funcional del proyecto y continuar con el proceso de implementación de la base de datos inteligente.

En la **Figura 9** se muestran los elementos utilizados de la metodología Scrum para la realización de dicha implementación, dando como resultado la creación de un *product backlog* el cual busca especificar un listado de tareas que son clave para la implementación del prototipo.

Figura 9.

Product backlog del proyecto



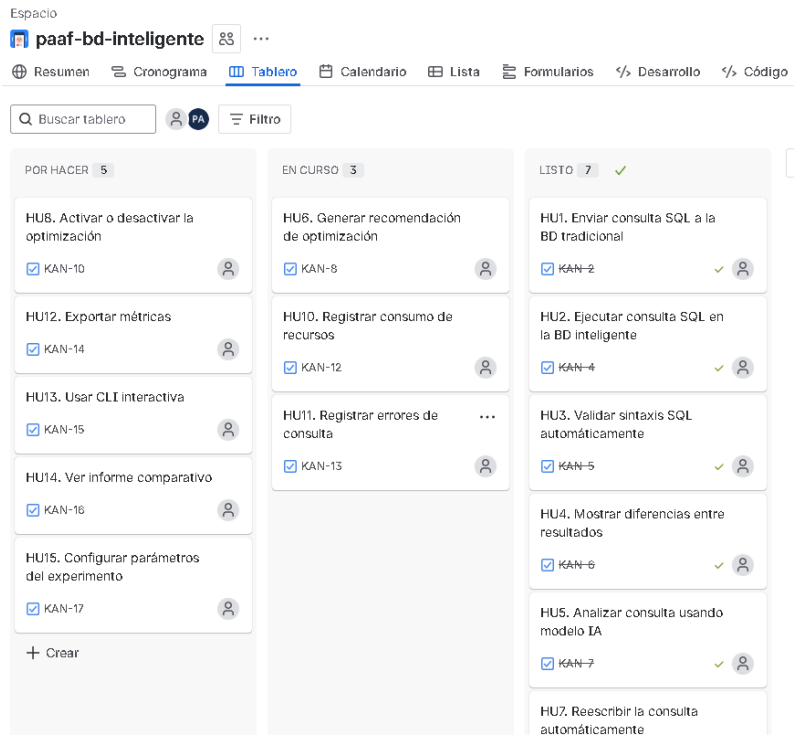
Nota. Captura de pantalla del producto backlog hecho en Jira. Elaboración propia.

Por otra parte, en la **Figura 10** se muestra otro componente importante de la metodología Scrum, siendo este el *Sprint backlog* del proyecto. Dicho elemento representa el conjunto de tareas necesarias para el desarrollo del prototipo funcional de base de datos

inteligente, dando como resultado un table KANBAN, el cual representa la clasificación de un grupo de tareas en base a su nivel de relevancia.

Figura 10.

Sprint backlog del proyecto visto por medio de un tablero Kanban



Nota. Captura del Sprint backlog hecho en Jira. Elaboración propia.

Con la culminación de las Fase 1 (planeación, análisis y diseño), se procede con la siguiente fase.

Fase 2 (desarrollo e implementación).

Se comienza con la configuración del entorno inicial de desarrollo, en donde se configura la *API KEY* de Open AI, para que tenga acceso a la base de datos en MySQL, con los datos del *dataset* de admitidos de la universidad de los Llanos, periodo 2021-2025.

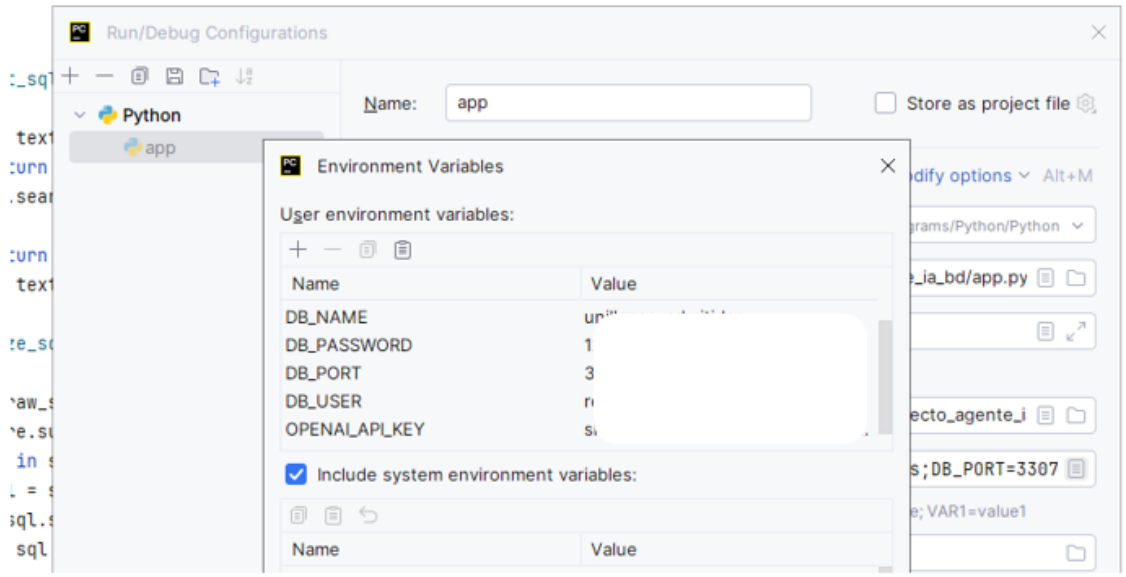
Luego de ello, se procede a colocar las credenciales de acceso a la base de datos, para dar el acceso correspondiente al agente IA.

En la **Figura 11** se muestra la configuración de las credenciales de acceso a la base de datos y los datos de la API KEY de Open AI, para permitir la manipulación de la base de

datos a manos del agente de IA. Dicho proceso se realizó en base a buenas prácticas, por lo que todas las credenciales han sido configuradas como variables de entorno, para encapsular los datos y así restringir la visibilidad de estos para que sean secretos.

Figura 11.

Configuración de credenciales de acceso y API KEY



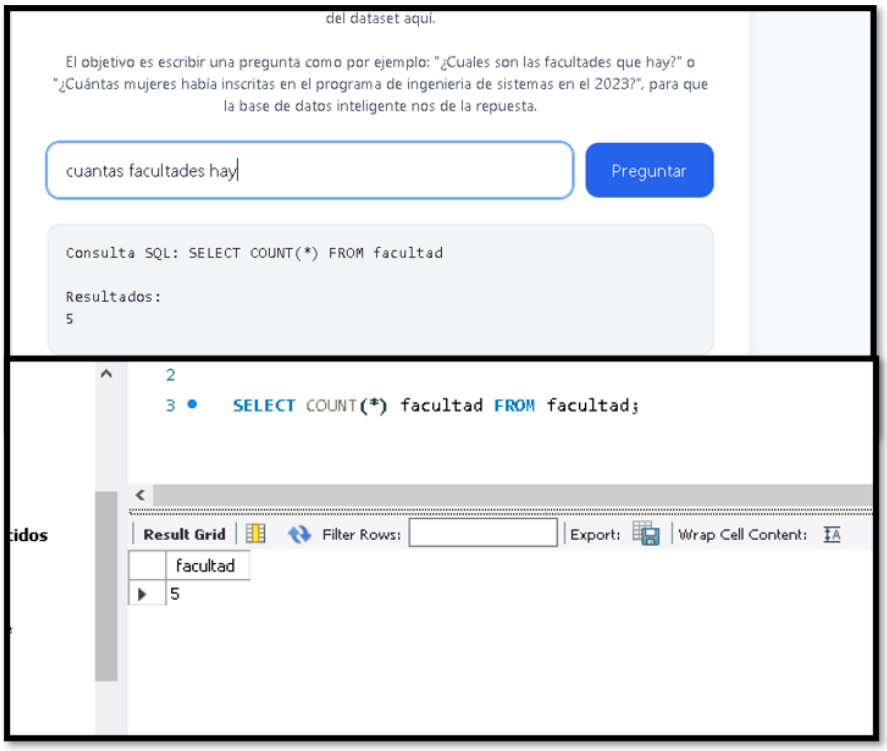
Nota. Captura de pantalla de credenciales de acceso en PyCharm. Elaboración propia.

Entre las tecnologías usadas para el desarrollo de la aplicación, se encuentran los lenguajes web HTML5 y CSS3 para el desarrollo de la aplicación web, y Python como lenguaje de programación el cual fue usado mediante el entorno de desarrollo de Pycharm. Después de esto, se termina con la construcción de la aplicación y se procede con el desarrollo de pruebas técnicas, para determinar el grado de exactitud de la aplicación.

En la **Figura 12** se muestra una captura de pantalla donde se puede apreciar la comparación de resultados obtenidos del entorno de prueba del prototipo de base de datos inteligente y la base de datos tradicional mediante el uso del SGBD de MySQL. Llegando a la conclusión de que sí existe coherencia entre ambas bases de datos, lo que permite garantizar que la base de datos inteligente funciona de manera correcta.

Figura 12.

Resultados comparativos entre el prototipo y el SGBD



Nota. Captura de pantalla de resultados. Elaboración propia.

Como se observa en la figura anterior, los resultados son coherentes entre una consulta NQL, en donde se preguntaba cuántas facultades hay, y el resultado obtenido de una consulta SQL, directamente hecha desde el SGBD de MySQL. Por fines prácticos se le ha especificado al agente IA que en su respuesta coloque, de forma adicional, la consulta SQL que entiende en base a la instrucción NQL recibida.

Esto se hizo con el objetivo de observar el nivel de precisión del agente de IA y ajustarlo de ser necesario, ya que al necesitar de un contexto tan específico para que sus respuestas fueran coherentes, era necesario ver la traducción SQL que arrojaba en base a la consulta NQL realizada.

Para lograr la precisión de las respuestas de la base de datos inteligente, se utilizó la técnica de Role prompting, en donde se asignó un rol predefinido al agente IA con la

finalidad de hacer que sus respuestas fueran más acordes al contexto de la información contenida dentro de la base de datos, y así disminuir su margen de inexactitud.

Así mismo, se delimito el alcance del agente IA para que solo pudiera dar respuesta en base a elementos propios del Data Query Language (DQL), mejor conocido como el componente SELECT dentro de SQL. Esto, debido a que es la única operación SQL que permite la visualización de los datos contenidos dentro de una base de datos.

Con los ajustes anteriores realizados, se termina la fase 2 conformando el prototipo de base de datos inteligente para ponerlo a prueba en un entorno que permita recolectar los datos necesarios y así comprobar o refutar la hipótesis general de este trabajo. Dentro del apéndice G de este trabajo se ha colocado parte del código utilizado para aplicar la técnica del Role Prompting en un agente IA.

Fase 3 (Pruebas).

En esta fase se dio inicio a una serie de consultas básicas, con la finalidad de medir y procesar los datos obtenidos en los tiempos de ejecución de cada consulta SQL y NQL. En donde se midió el tiempo de respuesta mostrado por el SGBD de MySQL y el tiempo arrojado por los logs de la aplicación del prototipo funcional de bases de datos inteligente. En esta última parte, para poder lograr la medición de los resultados del prototipo, se agregó un módulo que permitiera registrar el tiempo que pasa desde que se envía la consulta NQL, hasta la visualización de la información en pantalla.

Adicional a ello, se ajusta la estructura de la base de datos, para que dichos registros de tiempo se almacenen en una tabla llamada “consulta_logs”. Esto último se hizo con la finalidad de hacer un análisis más detallado en cuanto a la medición y comparación de los tiempos de ejecución en cada entorno de prueba, en el apéndice H se han colocado los resultados de los cambios realizados a la estructura de la base de datos para el registro de los

tiempos de medición. Una vez terminada la fase de pruebas, se procede con la presentación de los resultados preliminares y los resultados finales.

Resultados preliminares.

Una vez finalizados estos ajustes se reemplazó el conjunto de datos inicial por otro denominado Sakila, siendo este último un estándar de prueba muy popular en el ámbito de las bases de datos relacionales. La razón de este cambio se debió a la necesidad de comprobar el nivel de exactitud y veracidad de los resultados generados por el prototipo funcional de la base de datos inteligente, con fuentes externas como lo son manuales, repositorios, foros y la documentación oficial de Sakila, permitiendo validar de forma rápida y certera los resultados arrojados por la base de datos inteligente. De esta forma, al realizar una comparación entre ambas partes, se pudo confirmar que los resultados eran similares.

Sakila cuenta con 16 tablas y con más de mil registros entre todas ellas, lo que convierte a este conjunto de datos de ejemplo en un candidato idóneo para realizar pruebas de comparación y medición.

En base a lo anterior, se procede nuevamente a realizar un ajuste por medio de la técnica de role prompting, con la finalidad de elevar el nivel de precisión de las respuestas de la base de datos inteligente.

Una vez realizado este ajuste se comienza con la ejecución de tres (3) consultas NQL, y así evaluar el desempeño de la base de datos inteligente.

Dichas consultas fueron las siguientes:

1. Qué películas hay.
2. Cuántos actores son.
3. Dame la cantidad de películas por categoría.

El objetivo en este punto era medir el tiempo de ejecución de una consulta SQL hecha directamente desde la base de datos y compararlo con las tres consultas NQL hecha a la base de datos inteligente.

A medida que se fueron ejecutando las consultas, se registraron hallazgos bastante sorprendentes en cuanto al objeto de estudio. Y esto se debe al hecho de que las consultas NQL propuestas tienen ciertas características que se explican a continuación:

La consulta número 1 tiene como objetivo evaluar el nivel más básico de complejidad de la instrucción, siendo esta primera consulta una orden para traer todos los datos de la tabla películas.

Por su parte, la consulta número 2 se torna un poco más compleja, ya que para su desarrollo se requiere de hacer uso de la función de agregación COUNT (), la cual nos permite contar de manera muy específica la cantidad de actores presentes en la base de datos.

Y la última consulta, la número 3, es la más compleja de todas, ya que a nivel de tiempos de ejecución y dificultad de escritura representa un nivel superior, esto debido a que se necesita de un JOIN para poder cruzar los datos de tres tablas: la tabla films, la tabla category y la tabla film_category.

Lo anterior se debe a la relación foránea que existen entre las tres tablas planteadas de esa forma en su estructura, en donde se necesita de la tabla film_category para poder enlazar las tablas films y category.

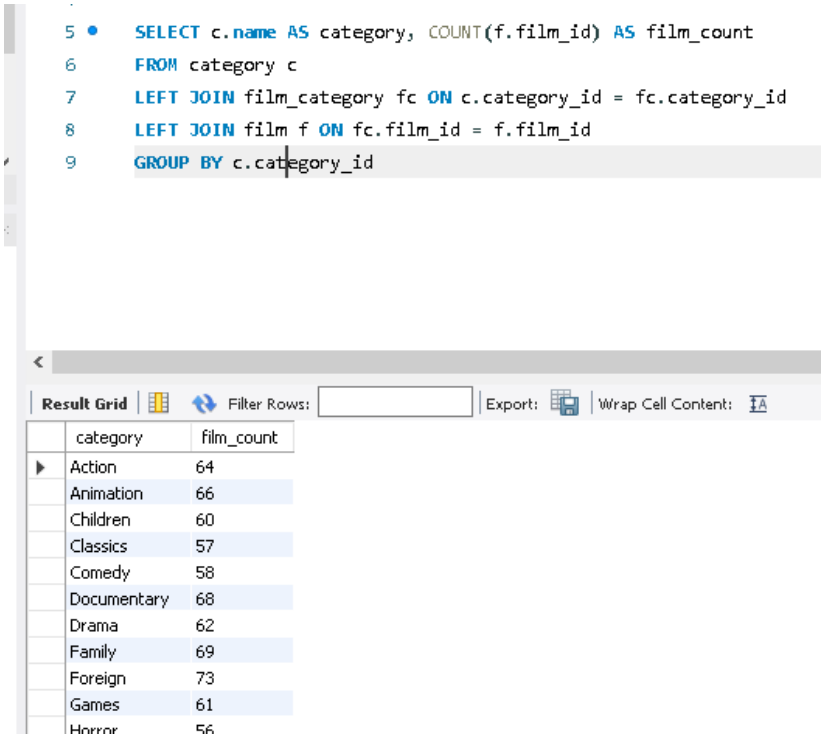
Una vez finalizadas las pruebas, se procede a presentar los resultados obtenidos de estas, en donde primero se muestra la consulta SQL hecha por un usuario, luego se muestra el resultado generado por la base de datos inteligente y por último se muestran los logs donde se puede apreciar los tiempos de ejecución de la consulta NQL.

Por fines prácticos, solo se muestra el resultado de la consulta número 3, la cual se considera como la más compleja, los resultados de las consultas 1 y 2 se han colocado en el apéndice I de este trabajo.

En la **Figura 13** se muestra el resultado 1 de la consulta 3, el cual trata sobre la implementacion de un LEFT JOIN anidado a tres tablas, dentro del SGBD de MySQL, dando como resultado el numero total de peliculas por cada categoría.

Figura 13.

Resultado 1 de consulta 3

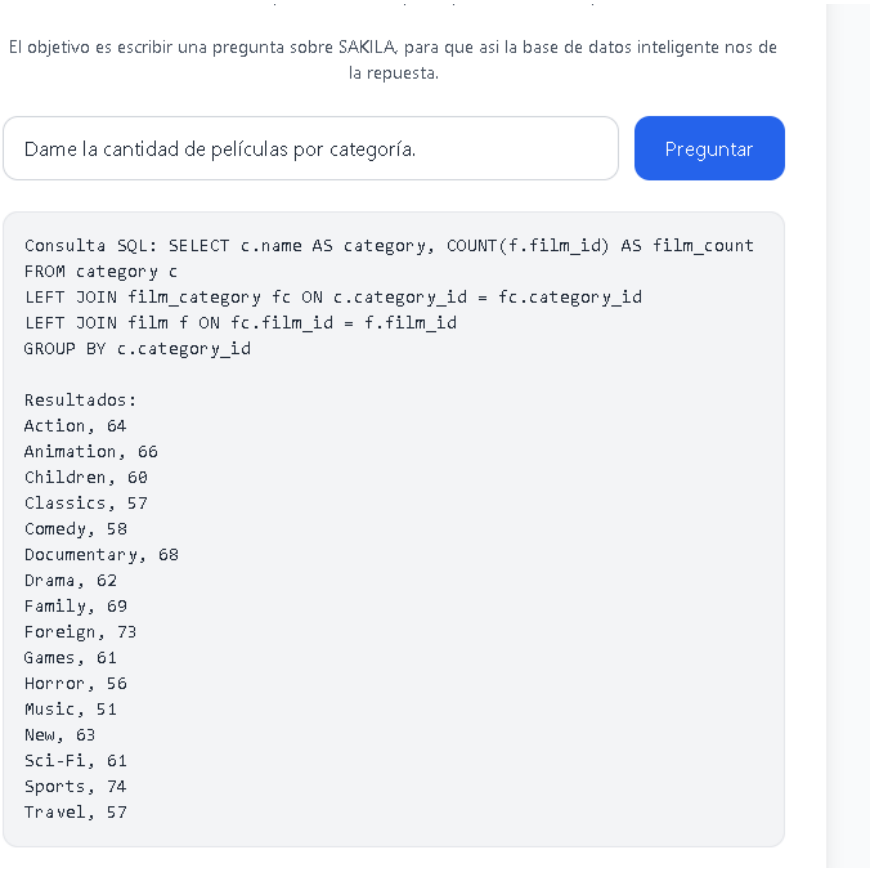


Nota. Captura de pantalla del SGBD de MySQL. Elaboración propia.

En la **Figura 14** se muestra el resultado 2 de la consulta 3, y para obtener los datos de este escenario de prueba se ha utilizado la misma consulta que en el apartado anterior, solo que esta vez la información solicitada se pasó directamente al prototipo de bases de datos inteligente.

Figura 14.

Resultado 2 de consulta 3

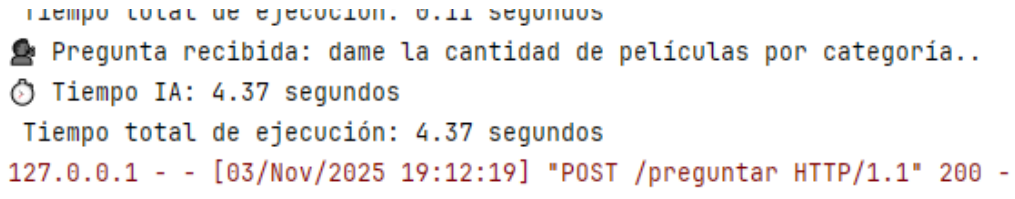


Nota. Captura de pantalla del prototipo de base de datos inteligente. Elaboración propia.

En la **figura 15** se muestra una captura de pantalla de la salida de los logs del sistema, en donde se puede apreciar el tiempo que le toma a la base de datos inteligente procesar una consulta NQL.

Figura 15.

Resultado de medición de tiempo consulta 3

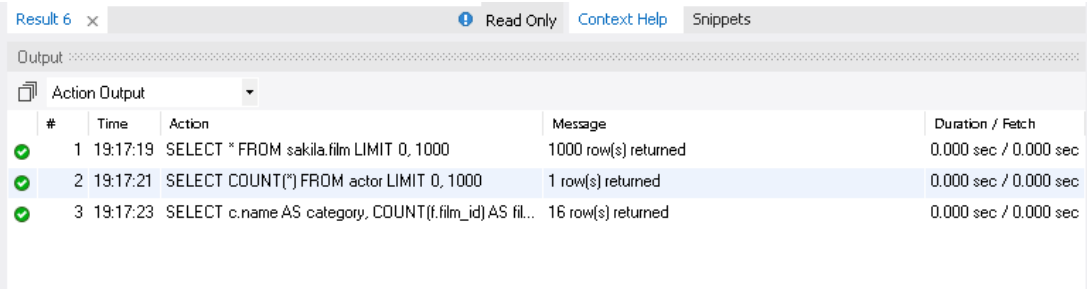


Nota. Captura de pantalla de los logs del sistema. Elaboración propia.

En la **Figura 16** se muestra una captura de pantalla de los *logs* o registros de SGBD de MySQL, los cuales quedan organizados por medio de un historial de operaciones, en donde específicamente para este escenario de prueba se ejecutaron las consultas 1,2 y 3 de forma consecutiva.

Figura 16.

Logs del SGBD de MySQL



Nota. Captura de pantalla de los logs de MySQL. Elaboración propia.

Una vez presentados los resultados anteriores, se procede a colocar los logs registrados por el prototipo y el SGBD en una tabla, con la finalidad de realizar un comparativo entre los tiempos de ejecución de las consultas SQL y las consultas NQL.

En la **Tabla 3** se muestran la organización de los datos obtenidos en ambos escenarios de prueba, teniendo en cuenta la cantidad de registros procesados y el tiempo de ejecución en segundos y milisegundos, así como también el contenido de las consultas NQL y SQL propuestas.

Tabla 3.

Tabla de tiempos de ejecución

Nro Consulta	Consulta NQL	Consulta SQL	N° registros	Tiempo promedio tradicional (s)	Tiempo promedio inteligente (s)
consulta 1	Que películas hay.	SELECT * FROM sakila.film;	1.000	0	3,88
consulta 2	Cuantos actores son.	SELECT COUNT(*) FROM actor;	1	0	0,11
consulta 3	Dame la cantidad de películas por categoría.	SELECT c.name AS category, COUNT(f.film_id) AS film_count FROM category c LEFT JOIN film_category fc ON c.category_id = fc.category_id LEFT JOIN film f ON fc.film_id = f.film_id GROUP BY c.category_id	16	0	4,37

Nota. Elaboración propia.

Como conclusión de la tabla anterior, podemos decir con toda certeza que al dar una instrucción en lenguaje natural a un agente IA se puede observar que los tiempos de ejecución aumentan, ya que el proceso de analizar e interpretar una consulta en NQL, transformarlo en una consulta SQL y posteriormente enviar dicha consulta a la base de datos, para obtener las respuestas por medio de un SELECT, hace que el proceso incremente sus tiempos de ejecución.

También, con el objetivo de verificar otro ámbito en el proceso de las consultas SQL, se realizó el mismo ejercicio, pero esta vez tomando el tiempo que se demora un usuario en digitar una consulta SQL, directamente dentro de la consola del SGBD, para ello se le pidió ayuda a personas que conforman la muestra de la población objetivo para que hicieran este ejercicio.

En la **Tabla 4** se muestran los resultados de la medición de los tiempos obtenidos por cada consulta digitada, en donde se categorizaron los resultados por intervalos de tiempo y por el nivel de formación de cada usuario, siendo esto último definido por su respectivo rol.

Tabla 4.

Tiempos de escritura de consultas

Perfil	Consulta 1	Consulta 2	Consulta 3
Persona Normal	10 - 15 s	5 - 8 s	45 - 60 s
Estudiante de TI	5 - 8 s	3 - 5 s	20 - 35 s
Ing. Sistemas / DBA	3 - 5 s	2 - 4 s	15 - 25 s

Nota. Elaboración propia.

El tamaño de la muestra estuvo conformado por cinco a siete individuos pertenecientes a cada perfil evaluado. Para el desarrollo y ejecución de las pruebas se utilizó un computador que tenía MySQL, y dentro de este una copia del dataset de Sakila, logrando así que se pudieran registrar los datos obtenidos para sacar un valor promedio en los tiempos de digitación de cada consulta.

Aunque los resultados presentaron cierto grado de variabilidad entre las personas que conformaron la muestra, debido a factores específicos como el nivel de conocimiento en SQL y la habilidad para teclear una instrucción, se pudo confirmar que sí existe una diferencia significativa al momento de digitar una consulta SQL y NQL.

Al final, se llega a la conclusión de que los tiempos que le toma a una persona el digitar una consulta SQL, comparado con los tiempos que le toma al agente IA para completar cada consulta, son bastante diferentes, indicándonos de que, efectivamente, sí se reducen tiempos cuando se automatizan este tipo de tareas.

Con lo que respecta al nivel de exactitud de la base de datos inteligente, se puede apreciar que en la salida de cada una de las consultas es coherente con los resultados arrojados por el SGBD de la base de datos tradicional.

Mas allá de ello, se generaron tres métricas de precisión en donde se evaluaron aspectos como precisión semántica, tasa de corrección y tasa de éxito, por medio de una consulta NQL, escrita de tres formas distintas (dos de ellas escritas con mala ortografía y una de ellas en otro idioma), con la finalidad de evaluar el nivel de exactitud del modelo IA utilizado en el prototipo de base de datos inteligente. En base a lo anterior se generaron datos concretos, representados en la siguiente tabla:

En la **Tabla 5** se muestra el consolidado de los resultados obtenidos en las pruebas realizadas, para determinar el nivel de exactitud de la base de datos inteligente, plasmando en dicha tabla los casos de prueba correspondientes a la mala ortografía de una misma consulta, y el porcentaje de acierto por parte del agente de IA.

Tabla 5.

Consolidado de tasa de éxito, precisión semántica y tasa de corrección

Consulta / Caso de Prueba	Pregunta en Lenguaje Natural (NQL)	Acción Generada por I.A. (SQL/Comando)	Cantidad de Registros Obtenido	Tasa de Éxito (%)	Precisión Semántica (Sí/No)	Tasa de Corrección (Nº de Intentos)
1	Dime cuantas ciudades que comiencen por la letra A, hay.	Consulta SQL: SELECT COUNT(*) FROM city WHERE city LIKE 'A%'	43	100%	Si	0
2	Dims cuantasciudades por la letra A, ha	Consulta SQL: SELECT COUNT(DISTINCT city) FROM city WHERE city LIKE 'A%'	43	100%	Si	0
3	How many cities exits in th edata base with a inthe nam	Error: 1054 (42S22): Unknown column 'name' in 'where clause'	0	0%	No	0

Nota. Elaboración propia.

Cuando hablamos acerca de la precisión semántica nos referimos a la capacidad que tiene el modelo para interpretar la consulta NQL que está recibiendo y, así mismo, evaluar qué entiende por esa consulta y qué tan coherente son sus resultados.

Por otra parte, la tasa de corrección es el valor correspondiente a las veces que un usuario debe estar modificando su consulta, para que el agente de IA interprete de forma correcta la consulta NQL.

Con lo que respecta a la tasa de éxito, es el valor porcentual que se ve reflejado cuando el agente hace la consulta SQL a la base de datos de forma positiva.

Al final de estas pruebas, se llega a la conclusión de que se necesita un nivel mínimo de escritura para poder dar una instrucción que sea entendible por la IA para así obtener los resultados que buscamos.

Resultados finales.

En base a todos los datos recolectados y al corroborar de manera rigurosa los resultados dados por las consultas NQL y las consultas SQL, finalmente se procede con las pruebas finales en donde nuevamente pasamos a trabajar con la base de datos de "unillanos_admitidos", la cual, como ya se había mencionado antes, contiene todo el conjunto de datos correspondientes a las personas que fueron admitidas por la universidad de los Llanos en el periodo comprendido entre 2021-2025. A partir de este punto, se procede a realizar pruebas más exhaustivas por medio de la escritura de consultas SQL y NQL más complejas, con el objetivo de consolidar los resultados finales de esta investigación.

En la **Figura 17** se muestran los resultados de una consulta intermedio-avanzado, la cual requiere hacer uso de un JOIN anidado a tres tablas y las cláusulas GROUP BY, ORDER BY y LIMIT.

Figura 17.

Resultado final 1 cantidad de mujeres por programa en el 2024-1



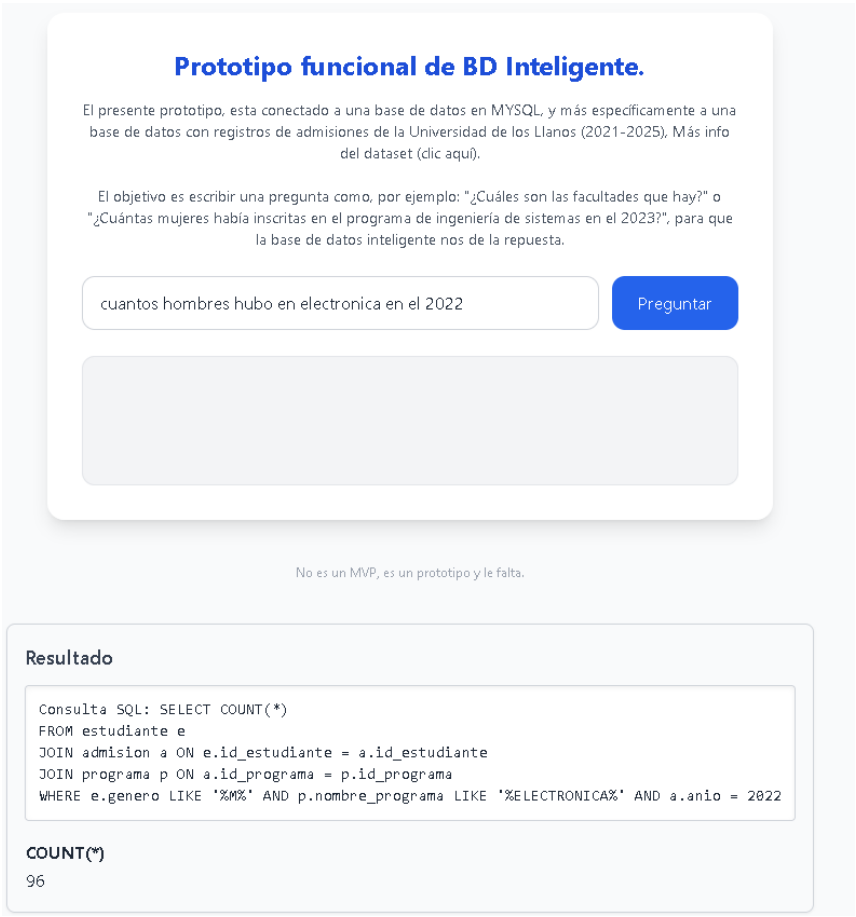
Nota. Captura de pantalla de resultados del prototipo. Elaboración propia.

En la **Figura 18** se presenta una prueba más difícil, la cual involucra el uso de operaciones JOIN y clausulas utilizadas en el tratamiento de cadenas de caracteres, mediante la aplicación de la condicional LIKE. Esta prueba permite evaluar la capacidad del prototipo funcional de bases de datos inteligente, para analizar e interpretar consultas SQL con un grado de complejidad más alto.

Asimismo, también se observa como las instrucciones NQL proporcionadas al sistema son traducidas y convertidas de forma exitosa en código SQL. Este tipo de respuestas por parte del sistema ayuda a evidenciar la capacidad del prototipo para gestionar las consultas de usuario, manteniendo la coherencia y nivel de exactitud semántica del lenguaje SQL.

Figura 18.

Resultado final 2 cantidad de hombres en Ing. electrónica 2022

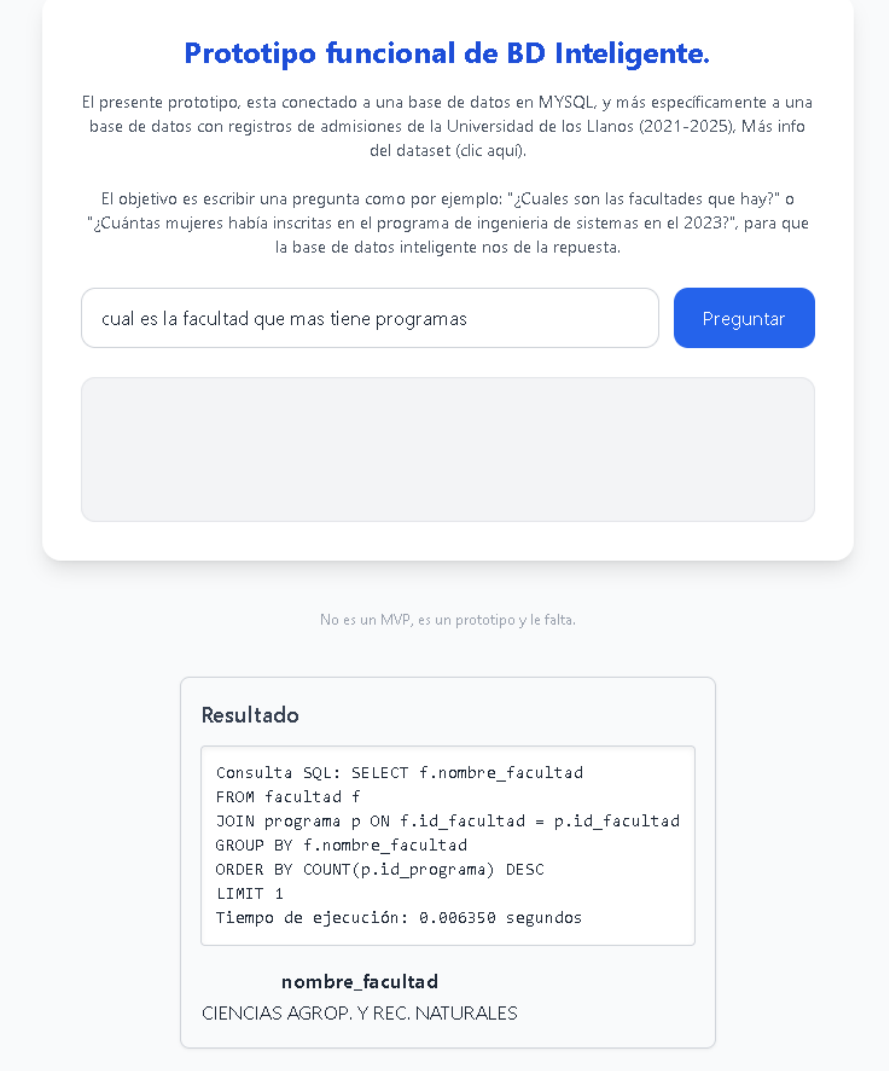


Nota. Captura de pantalla del prototipo. Elaboración propia.

En la **Figura 19** se muestra la última prueba realizada, la cual consistía en realizar una consulta que requería cierto grado de análisis, aunque este último ejercicio no es tan complejo como los anteriores, sí representaba un grado mayor de conocimiento en el lenguaje, ya que en esta última parte era necesario construir una consulta que fuera precisa y muy óptima, para no desperdiciar los recursos de la base de datos.

Figura 19.

Resultado final 3 la facultad con más programas en la universidad



Nota. Captura de pantalla del prototipo. Elaboración propia.

Una vez obtenidos los resultados finales, se procede a formular cuatro métricas, que permitieron cuantificar ciertos aspectos claves del rendimiento de la base de datos inteligente.

Las métricas ya antes mencionadas, son las siguientes:

1. Tiempo de respuesta de la aplicación.
2. Porcentaje de mejora, entre una base de datos inteligente vs tradicional.
3. Porcentaje de exactitud de predicciones correctas.
4. Diferencia entre tiempos de respuesta.

1. Tiempo de respuesta de la aplicación.

El tiempo de respuesta representa la medición del tiempo que le toma al sistema el recibir, procesar y mostrar un resultado correspondiente a una consulta NQL hecha por un usuario.

Fórmula:

$$RT = t_(fin) - t_(inicio)$$

Donde tiempo de respuesta (RT) es el tiempo que se toma el sistema en dar una respuesta, el cual es igual a tiempo fin (tfin) – tiempo inicio (tinicio), los cuales representan el proceso de medición de tiempo tomado por el script de medición de tiempos de ejecución, desde el inicio del envío de la petición hasta que se muestran los resultados.

2. Porcentaje de mejora, entre una base de datos inteligente vs tradicional.

El porcentaje de mejora representa la cifra comparativa del rendimiento de una base de datos inteligente con el de una base de datos tradicional.

Fórmula:

$$Mejora\ (\%) = (RT_(tradicional) - RT_(inteligente)) \,/\, RT_(tradicional) * 100$$

En la formula anterior, el porcentaje de mejora (Mejora%) se calculó haciendo uso de la métrica de tiempo de respuesta (RT), aplicada a los tiempos de ejecución obtenidos de las pruebas realizadas a la base de datos inteligente (RTinteligente) y a la base de datos tradicional (RTtradicional).

3. Porcentaje de exactitud de predicciones correctas.

El porcentaje de exactitud de predicciones correctas muestra el nivel de precisión de la base de datos inteligente, teniendo en cuenta la cantidad de las consultas NQL hechas por los usuarios.

Fórmula:

$$Accuracy = (Predicciones\ correctas) / (Total\ de\ predicciones) * 100$$

Donde el porcentaje de exactitud está representado por la variable (Accuracy), nombrado así por su traducción al inglés, la cual es igual al número de predicciones correctas sobre el total de predicciones hechas y multiplicado por 100, para poder sacar dicho porcentaje.

4. Diferencia entre tiempos de respuesta

La diferencia entre tiempos de respuesta, depende la métrica Nro. 1 - Tiempo de respuesta (RT), la cual debe ser aplicada a los datos obtenidos de ambas bases de datos (inteligente y relacional), aclarando que a diferencia de la métrica Nro. 2 - Porcentaje de mejora, se busca visualizar el grado de diferencia entre los tiempos de repuesta, cuando se miden los tiempos de ejecución en ambas bases de datos.

Fórmula:

$$Tiempo = RT_(tradicional) - RT_(inteligente)$$

Donde se aplica la métrica número uno a los valores obtenidos en ambas bases de datos. Dando como resultado las variables (RTtradicional) y (RTinteligente), aclarando que no se aplica a un proceso de sumatoria, sino de forma individual para cada registro tomado.

En base a las métricas anteriores, se procedió con a la aplicación de estas a los datos tomados en las pruebas hechas con el prototipo de bases de datos inteligente, y siendo más específicos a los registros de las consultas NQL almacenadas dentro de la base de datos. El resultado final de este ejercicio se puede apreciar en el apéndice K-Tabla de consolidado de métricas.

Se seleccionaron solo 10 consultas básicas porque el objetivo del experimento no era evaluar todas las consultas posibles sobre la base de datos de prueba, sino establecer un

conjunto representativo de casos de éxito que permitiera evaluar el funcionamiento del prototipo bajo diferentes tipos de consultas.

Teniendo como punto de partida la aclaración ya antes mencionada, se presenta a continuación un resumen de los datos contenidos en el apéndice K, aplicando la media, mediana, desviación estándar, máximo y mínimo:

Tabla 6.

Resumen estadístico de los valores consolidados de las métricas.

Estadístico	Inteligente	Tradicional
Media	2,784 s	6,232 s
Mediana	2,800 s	5,100 s
Desviación estándar	0,528 s	4,176 s
Mínimo	1,940 s	1,700 s
Máximo	3,600 s	13,800 s

Nota. Elaboración propia.

En base a los resultados anteriores, se determinó que el prototipo funcional de inteligencia artificial presentó un tiempo promedio de respuesta de 2,784 segundos, frente a 6,232 segundos del método tradicional. Dando como resultado una diferencia promedio entre ambos escenarios, la cual fue de 3,448 segundos.

Así mismo, continuando con el análisis de los resultados anteriores, se aplicaron las pruebas de Shapiro Wilk y la prueba t de Student con la finalidad de hallar la medida de p-valor y así determinar el nivel de significancia, así poder respaldar la decisión de verificar o rechazar la hipótesis de investigación. En base a ello, se obtuvieron los siguientes resultados:

Tabla 7.

Resumen de pruebas estadísticas de Shapiro Wilk y t de Student.

Prueba	Resultado
Shapiro-Wilk	p = 0,0883
t de muestras relacionadas	t(9) = 2,846
p-valor	0,0192
Nivel de significancia	0,05

Nota. Elaboración propia.

Partiendo de la lectura de la tabla 7, la interpretación que se puede hacer es la siguiente:

Debido a que p = 0,0192 < 0,05, se rechaza la hipótesis nula y se concluye que existe una diferencia estadísticamente significativa entre los tiempos de respuesta registrados en ambos escenarios experimentales.

3.8.2. Resultados perceptuales y selección de la muestra

En este apartado se elaboró una encuesta con la finalidad de recopilar datos de tipo cuantitativo y cualitativo, con el objetivo de conocer la percepción que tenían los profesionales y estudiantes del sector de T.I pertenecientes a la muestra de la población objetivo, con respecto al objeto de estudio, para así obtener una visión más holística sobre el impacto y el potencial que podría tener una base de datos inteligente en la región donde se llevó a cabo la investigación.

La encuesta antes mencionada fue elabora en base a 26 preguntas divididas en 6 módulos especificados de la siguiente manera:

- Modulo 1: consentimiento informado, se le pregunta al encuestado si acepta participar en el estudio y en donde se le indica que sus respuestas harán parte de un estudio.

- Modulo 2: perfil del encuestado, en donde se pide el género, el nivel educativo, cargo actual (si lo posee), años de experiencia y tres preguntas técnicas para determinar si se ha trabajado con bases de datos inteligentes.
- Modulo 3: este módulo se compone de tres preguntas clave en donde se mide el nivel de conocimiento del encuestado con respecto al uso de bases de datos inteligentes.
- Modulo 4: en esta sección buscamos medir la percepción que tiene el encuestado acerca del objeto de estudio por medio de preguntas abiertas relacionadas con el tema.
- Modulo 5: en el módulo 5 hacemos uso de una escala Likert para determinar la percepción que tiene el encuestado, si usará un base de datos inteligente en su entorno cotidiano.
- Modulo 6: y para finaliza la encuesta, se da un mensaje de agradecimiento por su participación.

De los 73 individuos con los que se tuvo contacto, entre estudiantes y profesionales del sector de TI de la región, solo 48 decidieron participar en la encuesta, dando como resultado una participación aproximada del 65 %.

Los resultados detallados de la encuesta aplicada, así como también el detalle especifico de cada pregunta, se especifican en el apéndice L de este trabajo.

En base a los datos de la encuesta podemos sacar las siguientes conclusiones:

Solo el 60 % de los encuestados tiene conocimientos acerca de lo que son las bases de datos, indicándonos que existe una brecha en el mercado de TI de la región en donde hay un porcentaje de la muestra que no conoce esta tecnología.

Aunque más del 60 % de los encuestados conoce y maneja una base de datos, se puede apreciar que poco o nada se sabe acerca de lo que son las bases de datos relacionales.

Cuando se miran los resultados del ítem 7, en donde se ponía aprueba el nivel de conocimiento de los encuestados acerca de la tecnología OLAP y OLTP, temas clave para la

gestión de los datos dentro de una organización, se llega a la conclusión de que la mayoría de los encuestados no conoce mucho sobre lo que es la arquitectura de datos.

Con lo que respecta a los resultados de las preguntas del módulo 4, las cuales eran pregunta abiertas, se concluye que sí hay una percepción positiva acerca de la automatización de procesos por medio de tecnologías emergentes con inteligencia artificial.

Así mismo, en el análisis de la escala Likert, se puede apreciar que la mayoría de los encuestados tiene una visión de confianza hacia la IA, pese a no estar familiarizado con las bases de datos inteligentes.

Lo que en términos generales nos da a entender que, si bien no se conoce mucho sobre el objetivo de estudio, hay una percepción bastante positiva acerca de la adopción de esta tecnología.

En base a lo anterior, se hace necesario especificar el origen de la muestra seleccionada para esta investigación, siendo esto un complemento en lo mencionado en la sección 3.8 (<<Población y muestra>>) de este documento.

Donde se aclara que tanto la aplicación de la encuesta, como las pruebas diseñadas para el uso de las 10 consultas SQL en el prototipo de base de datos inteligente estuvo conformada por 48 personas entre los 18 a 42 años en la ciudad de V/cio en el departamento del Meta en el país de Colombia, siendo este un rango de edad representativo para estudiantes y docentes de carreras de T.I. Se empleó un muestreo no probabilístico de tipo intencional, seleccionando participantes que cumplían con los criterios establecidos para la evaluación del prototipo. Siendo el criterio más importante, el tener conocimiento básico sobre el lenguaje SQL, debido a que era necesario comparar el uso de dicho lenguaje con el lenguaje natural de consultas NQL.

Validación de hipótesis.

En el último apartado de este capítulo se debe recordar la hipótesis general de este trabajo, la cual tenía la siguiente premisa:

"La utilización de un prototipo basado en inteligencia artificial para transformar consultas formuladas en lenguaje natural en consultas SQL produce una reducción en el tiempo de respuesta frente al método tradicional de consulta SQL sobre una base de datos relacional".

Y la respuesta corte es si, pero solo en ciertos aspectos, esto se debe al hecho de que los resultados obtenidos en base a las pruebas experimentales nos han demostrado que los tiempos de ejecución son más cortos cuando las consultas SQL se ejecutan directamente desde la consola de comando de un SGBD.

Pese a lo anterior, sí se confirma la hipótesis de investigación, al analizar otros aspectos relevantes del objeto de estudio, como la experiencia de usuario (UX), los tiempos de escritura e interpretación de las consultas, así como también el nivel de precisión de las consultas SQL, se pudo evidenciar una mejora significativa que no se tenía prevista en términos de eficiencia de recursos por medio de la automatización y optimización de las consultas al utilizar consultas NQL.

4. Conclusiones y recomendaciones

En el presente capitulo se muestran los elementos más relevantes de este trabajo, en donde se generan una serie de conclusiones en base a los resultados obtenidos, así como también la verificación del cumplimiento del objetivo general y, por supuesto, de los objetivos específicos propuestos inicialmente.

También se exaltan los hallazgos más pertinentes obtenidos durante el diseño, desarrollo e implementación del prototipo funcional de bases de datos inteligente, incluyendo también elementos sumamente valiosos generados en la fase de pruebas, en donde por medio del método comparativo se evaluó el desempeño y/o rendimiento de una base de datos inteligente contra una base de datos tradicional.

Las conclusiones condensan los aspectos más relevantes de la metodología empleada, la cual fue una metodología de investigación mixta con un enfoque experimental de carácter comparativo, partiendo de una recolección de datos y especificaciones del sistema, representados en una fase de levantamiento de requerimientos, hasta llegar a la implementación final de un prototipo funcional, lo que nos permitió la recolección y procesamiento de datos cuantitativos enfocados a evaluar aspectos como la administración y gestión de grandes volúmenes de datos, desde la perspectiva del rendimiento y escalabilidad de un sistema.

Dicho esto, pasamos a presentar las siguientes conclusiones obtenidas a lo largo de todo este proceso.

La integración e implantación de una gente IA, dentro de una base de datos inteligente, mejoró de forma significativa los procesos de administración y gestión de grandes volúmenes de datos.

Si, pero solo en cierto aspecto, esto se debe al hecho de que se delimitó el alcance de la aplicación solo a las consultas SELECT, haciendo que sí exista una mejora significativa

cuando se trabajan consultas SQL complejas o muy extensas. Pero solo cuando se toma una medición rigurosa en los tiempos de ejecución entre una consulta hecha por medio de la base de datos inteligente y otra hecha directamente desde la consola del SGBD. Los resultados de las pruebas preliminares demuestran que los tiempos de ejecución son más extensos cuando se usa la base de datos inteligente, debido a todo el proceso que conlleva el traducir una consulta NQL a una consulta SQL por medio del agente IA, para poder obtener los resultados correspondientes. Sin embargo, cuando una consulta SELECT se vuelve compleja, la medición de los tiempos es menor cuando se usa una base de datos inteligente, debido a la automatización del proceso de consulta.

La metodología de investigación propuesta, así como también las herramientas y técnicas de modelamiento definidas dentro de un proceso de ingeniería del software, permitieron documentar el proceso de diseño e implementación de una base de datos inteligente.

Efectivamente, al hacer uso de un enfoque de investigación mixto se pudo evaluar el impacto que tiene el objeto de estudio en un ámbito con datos reales como lo fueron los datos de estudiantes admitidos de la universidad de los Llanos periodo 2021-2025. Así mismo, al aplicar las técnicas de modelamiento y demás herramientas en la rama de la ingeniería del software, nos permitió el diseño e implementación de un prototipo funcional de bases de datos inteligente, lo que permitió conocer más a fondo el funcionamiento de esta tecnología y los beneficios que trae consigo. Adicional a esto, cuando se recolectaron los datos cualitativos de la investigación por medio de una encuesta, se logró apreciar que esta tecnología está apenas dándose a conocer en la población de estudio, lo que hace que sea bastante llamativa en el entorno donde se desarrolló la investigación.

El desarrollo de todo el proyecto permitió demostrar que sí es posible integrar un modelo de inteligencia artificial existente a los procesos de administración y gestión de grandes volúmenes de datos almacenados dentro de una base de datos.

Al desarrollar y documentar todo el proceso de desarrollo del prototipo funcional de base de datos inteligente, se logra evidenciar la viabilidad de este tipo de implementaciones, para tareas típicas de administración y gestión de los datos dentro de una base de datos, incluso cuando estos tienen un alto volumen. Lo que trae consigo beneficios como una gran experiencia de usuario debido a su gran usabilidad y un cierre de brechas entre un usuario sin conocimientos en SQL comparado con un usuario experto en el tema, debido a que la capa de IA de la aplicación actúa como interprete para comunicar a los usuarios con la base de datos.

La base de datos inteligente demostró un nivel de precisión más alto de lo esperado, superando en ciertos aspectos el desempeño de una base de datos tradicional.

Si bien ya se había mencionado que hay una mejora significativa al momento de implementar una base de datos inteligente en procesos típicos de administración y gestión de los datos dentro de una base de datos, lo realmente sorprendente vino cuando se obtuvieron los resultados finales de las pruebas preliminares, en donde por medio de instrucciones básicas incluyendo algunas mal escritas o redactadas, se logra poner a prueba el grado de exactitud de la base de datos inteligente, dando como resultado una tasa del 100 % de exactitud cuando las consultas NQL están escritas de forma correcta. Esto debido a que al usar un modelo preentrenado tiene un mayor conjunto de datos para predecir y corregir las peticiones de usuario hechas en lenguaje natural.

Aportaciones principales

En base a la recolección de los datos perceptuales obtenidos en la encuesta sobre las bases de datos inteligentes, así como también de la recolección de antecedes definidos en el marco teórico, se llega a la conclusión de que las bases de datos inteligente son muy poco

conocidas en nuestro entorno cotidiano, y en base a ello se han podido identificar los siguientes puntos definidos como aportaciones principales de este trabajo al objeto de estudio.

Se ha desarrollado una experiencia de usuario más innovadora en el ámbito de gestión y administración de grandes volúmenes de datos, en el contexto de administración y gestión de una base de datos.

Se ha hecho un análisis más profundo acerca del impacto que tiene el uso del Natural Query Language o NQL en el entorno del desarrollo de software, aplicado al desarrollo de aplicaciones con inteligencia artificial.

Se ha visto la mitigación o disminución de la brecha existente entre un usuario sin conocimientos del lenguaje SQL y un usuario experto en el tema.

Se ha comprobado el nivel de eficiencia que tiene una base de datos inteligente, con respecto a una base de datos tradicional, dando como resultado un impacto positivo en el entorno de gestión y administración de un gran volumen de datos.

Se ha implementado el desarrollo de una base de datos inteligente a un problema real, en donde se pudo evidenciar por medio de los resultados finales una toma de decisiones informada de manera óptima y eficiente, basados en los datos de registros de estudiantes admitidos de la universidad de los Llanos.

Recomendaciones.

Ampliar muchísimo más los conjuntos de datos trabajados, esto debido a que las pruebas tuvieron un numero de 1000 a 11000 registros procesados, pero sería bastante pertinente conocer cuáles son los límites que puede alcanzar el sistema.

Utilizar el resto de las operaciones que conforman SQL, y más específicamente los DDL y DML. Aunque el trabajo demuestra que la base de datos inteligente soporta perfectamente estas operaciones, se hace necesario conocer más a fondo el comportamiento

del sistema cuando trabaja con operaciones DML como lo son DELETE, UPDATE, INSERT y operaciones DDL como ALTER, DROP y CREATE.

Probar la integración del agente IA con otros SGBD SQL, ya que puede existir la posibilidad de que el sistema sea más óptimo trabajando con otros softwares de SGBD como Oracle, PostgreSQL o SQL Server.

Realizar integraciones con modelos de bases de datos NoSQL y/o modelos híbridos, esto con la finalidad de verificar si hay algún cambio en el comportamiento del sistema y también para evaluar a futuro, si el sistema es más óptimo trabajando con este tipo de modelos de bases de datos, a comparación de un modelo únicamente relacional.

Evaluar el comportamiento del sistema cuando se somete a integraciones de bases de datos federadas y/o bases de datos distribuidas, con la finalidad de conocer más a fondo el funcionamiento del sistema, cuando este se somete a bases de datos que se encuentran en la nube de forma descentralizada y con escalabilidad horizontal.

5. Referencias

Aguado-Labrador, P. (2024). *Estudio de rendimiento de bases de datos para series temporales aplicado a la construcción de modelos basados en datos para la mejora de eficiencia energética en edificios inteligentes* [Trabajo de fin de grado, Universidad de Valladolid]. UVaDOC. https://uvadoc.uva.es/handle/10324/71318

Alvarado-Artunduaga, J. Á. (2023). *La inteligencia artificial en administración de empresas*. [Trabajo de pregrado, Universidad del Rosario]. Repositorio institucional EdocUR. https://repository.urosario.edu.co/items/2fc2be86-a7d6-4436-8184-8ec2b2c6cd7a

Arias-Naranjo, H. F., & López-Martínez, C. F. (2023). *Diseño de un sistema de supervisión y control de mensajes de spam por medio de inteligencia artificial (Machine Learning), cumpliendo con la norma ISO 27000 implementando la metodología de trabajo basado en ITIL 4*. [Trabajo de pregrado, Universidad Cooperativa de Colombia]. Repositorio Institucional UCC. https://repository.ucc.edu.co/entities/publication/bba81ce7-f3d7-4a26-9c70-92a6e735458e

Bernal-Montenegro, D. A. (2024). *Aplicación de la inteligencia artificial en la administración de justicia colombiana*. [Trabajo de pregrado, Universidad Católica de Colombia]. Repositorio institucional RIUCaC. https://repository.ucatolica.edu.co/handle/10983/31722

Brena, R. (1991). Bases de datos inteligentes. *Revista Universidad EAFIT, 27*(83), 47–65. https://publicaciones.eafit.edu.co/index.php/revista-universidad-eafit/article/view/1504/1376

Capgemini. (2025, 21 de enero). *El 71 % de los consumidores desea que la IA generativa se integre a sus experiencias de compra* [Comunicado de prensa]. Capgemini España.

https://www.capgemini.com/co-es/news/press-releases/el-71-de-los-consumidores-desea-que-la-ia-generativa-se-integre-a-sus-experiencias-de-compra/

Chaudhuri, S., & Narasayya, V. R. (2007). Self-tuning database systems: A decade of progress. *Proceedings of the VLDB*, 3(2), 3–14. https://www.microsoft.com/en-us/research/publication/self-tuning-database-systems-a-decade-of-progress/

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, *13*(6), 377–387. https://doi.org/10.1145/362384.362685

Cook, D. J., & Das, S. K. (2005). *Smart environments: Technology, protocols, and applications*. John Wiley & Sons. https://doi.org/10.1002/047168659X

DAMA International. (2017). *The DAMA guide to the data management body of knowledge (DAMA-DMBOK)* (2nd ed.). Technics Publications. https://technicspub.com/dmbok/

Duarte, F. (2025, 24 de abril). Amount of data created daily (2025). *Exploding Topics*. https://explodingtopics.com/blog/data-generated-per-day

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7.ª ed.). Pearson.

Figueroa-Mosquera, D. (2024). *La implementación de la inteligencia artificial en el sector público de Colombia: Una revisión de literatura*. [Trabajo de grado, Universidad EAFIT]. Repositorio Institucional Universidad EAFIT. https://hdl.handle.net/10784/35385

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137–144. https://doi.org/10.1016/j.ijinfomgt.2014.10.007

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. http://www.deeplearningbook.org

Guevara, F. C. (2025). *Inteligencia artificial orientada hacia la función pública*. [Trabajo de grado de maestría, Universidad del Rosario]. Repositorio institucional EdocUR. https://repository.urosario.edu.co/items/660fe01c-7fb5-4be2-8c49-0a2aa63d7f2f

Haskel, J., & Westlake, S. (2017). *Capitalism without capital: The rise of the intangible economy*. Princeton University Press.

Hasperué, W., Estrebou, C., Camele, G., Rucci, E., Ronchetti, F., Eguren, S., Castillo, D., Reyes Zambrano, G., Lanzarini, L., & Fernández Bariviera, A. (2025). *Procesamiento inteligente de grandes volúmenes de datos* [Ponencia]. XXVII Congreso Argentino de Ciencias de la Computación, Sedici UNLP. https://sedici.unlp.edu.ar/bitstream/handle/10915/176262/Ponencia.pdf?sequence=3&isAllowed=y

IBM. (2001). *Autonomic computing manifesto*. IBM. https://www.scirp.org/reference/referencespapers?referenceid=1675401

Jurafsky, D., & Martin, J. H. (2021). *Speech and language processing* (3rd ed.). Stanford University. https://web.stanford.edu/~jurafsky/slp3/

Kipf, A., Vorona, D., Müller, J., Leis, V., Boncz, P., Neumann, T., & Kemper, A. (2019). Estimating cardinalities with deep sketches. *Proceedings of ACM SIGMOD*, 1937–1940. https://doi.org/10.1145/3299869.3320218

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

López B., A., & Pérez, C. A. (2024). *Creación de un conjunto de datos para el entrenamiento de modelos de inteligencia artificial aplicados al sector judicial.* [Trabajo de grado, Universidad Distrital Francisco José de Caldas]. Repositorio Institucional UD. https://repository.udistrital.edu.co/items/086e165b-016d-4806-b900-f737958cb56e

López-Vega, J. E., Becerra, V., Guzmán, M. de los Á., & Landázuri-Sandoval, J. K. (2023). Inteligencia artificial en la justicia colombiana: ¿La solución a la congestión judicial? *Revista Lecciones Vitales*, *2023*(1), lv0101. https://doi.org/10.18046/RLV.2023.5655

Marcus, R., Negi, P., Mao, H., Zhang, C., Alizadeh, M., Kraska, T., Papaemmanouil, O., & Tatbul, N. (2018). Neo: A learned query optimizer. *Proceedings of VLDB Endowment*, *12*(11), 1705–1718. https://doi.org/10.14778/3342263.3342644

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing* (Special Publication 800-145). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.

Muñoz G., A. C., Aguilar, J., & Martínez, R. (2005*). Modelo inteligente para bases de datos distribuidas. Revista Gerencia Tecnológica Informática*, *4*(10), 91-116. https://revistas.uis.edu.co/index.php/revistagti/article/view/1562

Muñoz, E. C. (s. f.). *Los activos intangibles en la era del conocimiento*. ProQuest. https://www.proquest.com/openview/019a8d556da3388c1715c5a7defee5dd

Oracle Latinoamérica. (s. f.). *Autonomous Database*. Oracle. https://www.oracle.com/latam/autonomous-database/

PowerData. (2021, 6 de agosto). Bases de datos inteligentes en un mundo inteligente. *PowerData Blog*. https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bases-de-datos-inteligentes-en-un-mundo-inteligente/

Russell, S. J., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson. https://aima.cs.berkeley.edu/

Salazar-Díaz, R., Glavic, B., & Thomas, T. (2024). InferDB: In-database machine learning inference using indexes. *Proceedings of the VLDB Endowment*, *17*(2), 296–309.

https://www.researchgate.net/publication/381055111_InferDB_In-Database_Machine_Learning_Inference_Using_Indexes

Sánchez-Ando, J., Santana-Camargo, H. A., Restrepo-Mejía, D. A., & Malagón-Sánchez, D. A. (2023). *Análisis descriptivo del impacto de la inteligencia artificial en el sector de Contact Center en Bogotá*. [Trabajo de grado, Universidad del Rosario]. Repositorio Institucional EdocUR. http://hdl.handle.net/10882/13462

Webline India. (2025, 26 de agosto). Principales estadísticas de IA que debe conocer en 2024 y más allá: tendencias, cifras y pronósticos. *Webline India*. https://www.weblineindia.com/es/blog/artificial-intelligence-statistics/

Yang, Z., Chiang, W. L., Luan, S., Mittal, G., Luo, M., & Stoica, I. (2022). Balsa: Learning a query optimizer without expert demonstrations. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 931–944. http://dx.doi.org/10.1145/3514221.3517885

Zapata, L. (2022). *Sistemas híbridos inteligentes basados en lógica difusa y aprendizaje profundo*. [Tesis de doctorado, Universidad de los Andes]. Repositorio Institucional Séneca. https://repositorio.uniandes.edu.co/entities/publication/2887befe-f953-4e14-9e2c-aa9ba2ddb302

Zou, B., You, J., Wang, Q., Wen, X., & Jia, L. (2022). Survey on learnable databases: A machine learning perspective. *Big Data Research*, *27*. https://www.researchgate.net/publication/357548809_Survey_on_Learnable_Databases_A_Machine_Learning_Perspective

6. Apéndices

Los apéndices presentados a continuación, representan toda la información adicional de este trabajo, entre los cuales se encuentran diagramas, capturas de pantalla y entregables generados en base a las técnicas de modelamiento utilizadas, así como también elementos propios del marco de trabajo Scrum.

Evidentemente, estos apéndices no se colocaron dentro del cuerpo de este trabajo debido a su nivel de detalle y extensión. Sin embargo, eso no implica que su aporte no sea sumamente valioso al trabajo realizado.

Apéndice A. Diagrama de Gantt.

Este diagrama, representa la forma visual del cronograma de actividades del proceso de desarrollo del prototipo, en donde se aprecian las 6 etapas que conforman todo el proyecto.

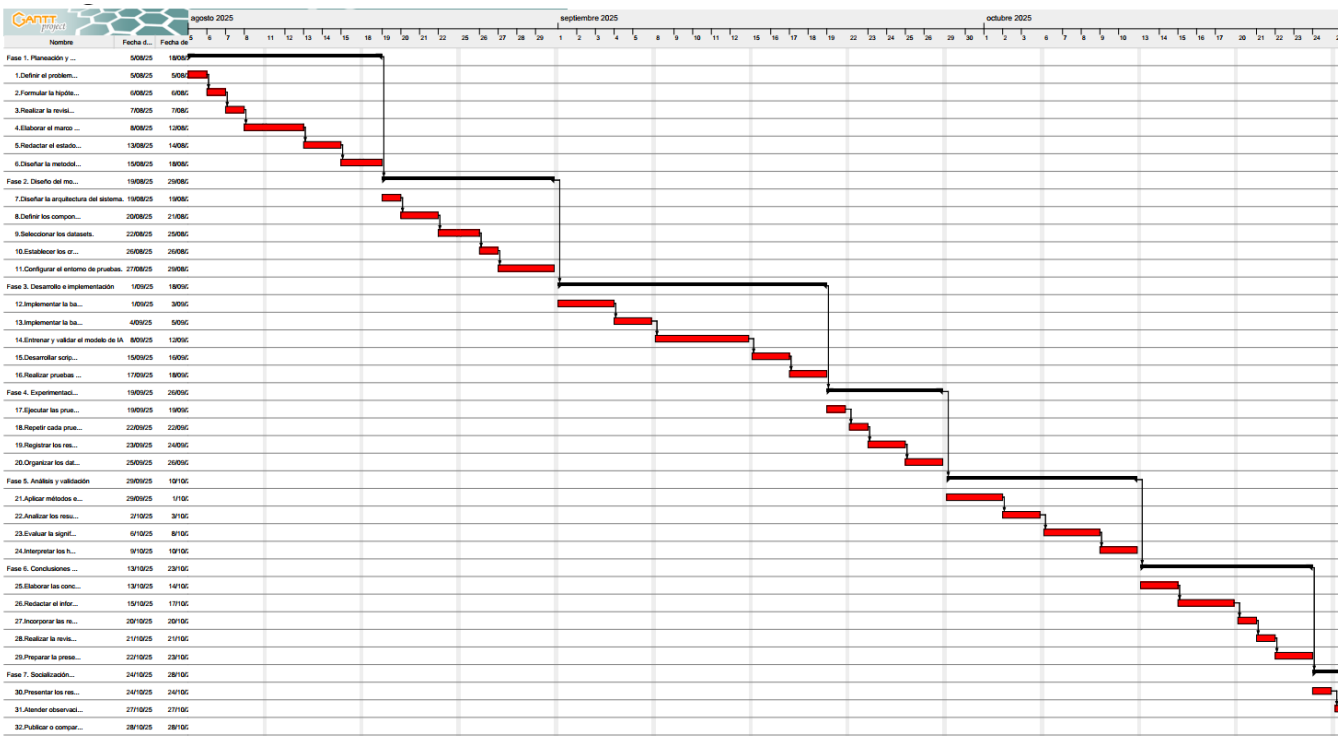


Figura 20 Diagrama de Gantt del proyecto.

Apéndice B. Diagramas UML y documentación de casos de uso.

Los siguientes diagramas UML, representan el análisis realizado en cuanto al comportamiento y estructura del prototipo.

Es por ello, se muestran a continuación:

- Dos diagramas de casos de uso.
- Un diagrama de secuencias.
- Un diagrama de clases.
- Un diagrama de despliegue.

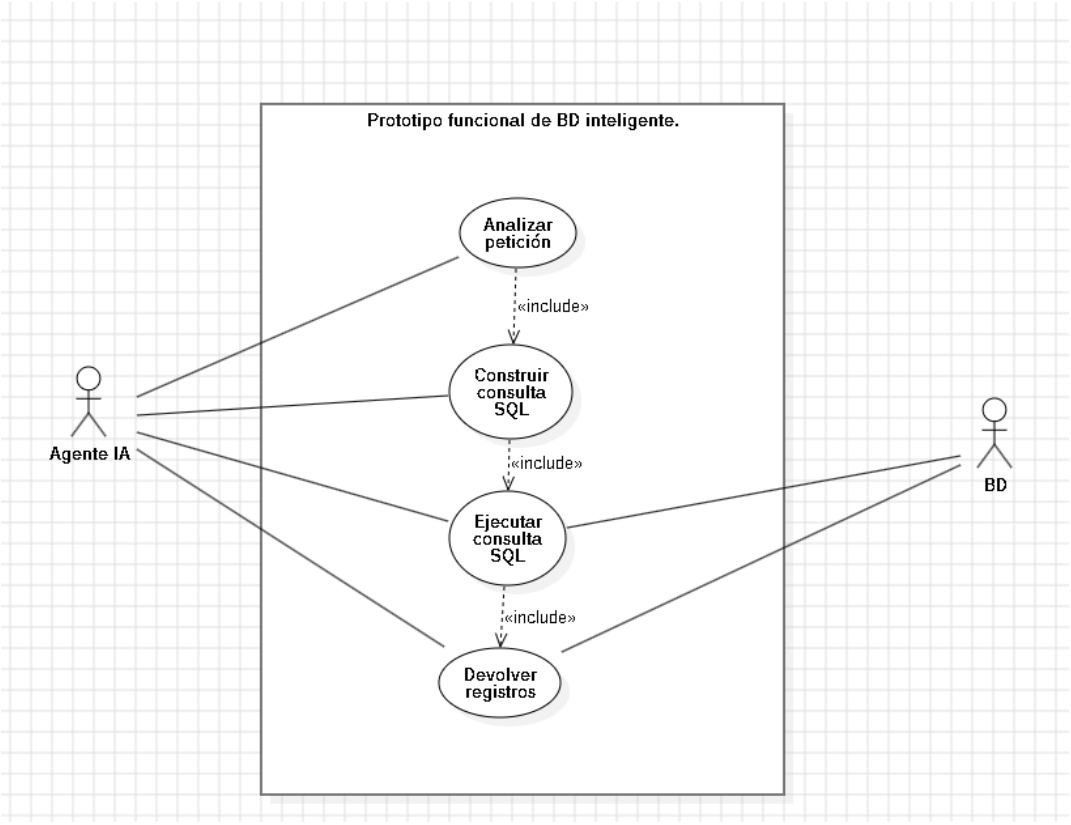


Figura 21 Diagrama de caso de uso 02 ejecutar consulta SQL.

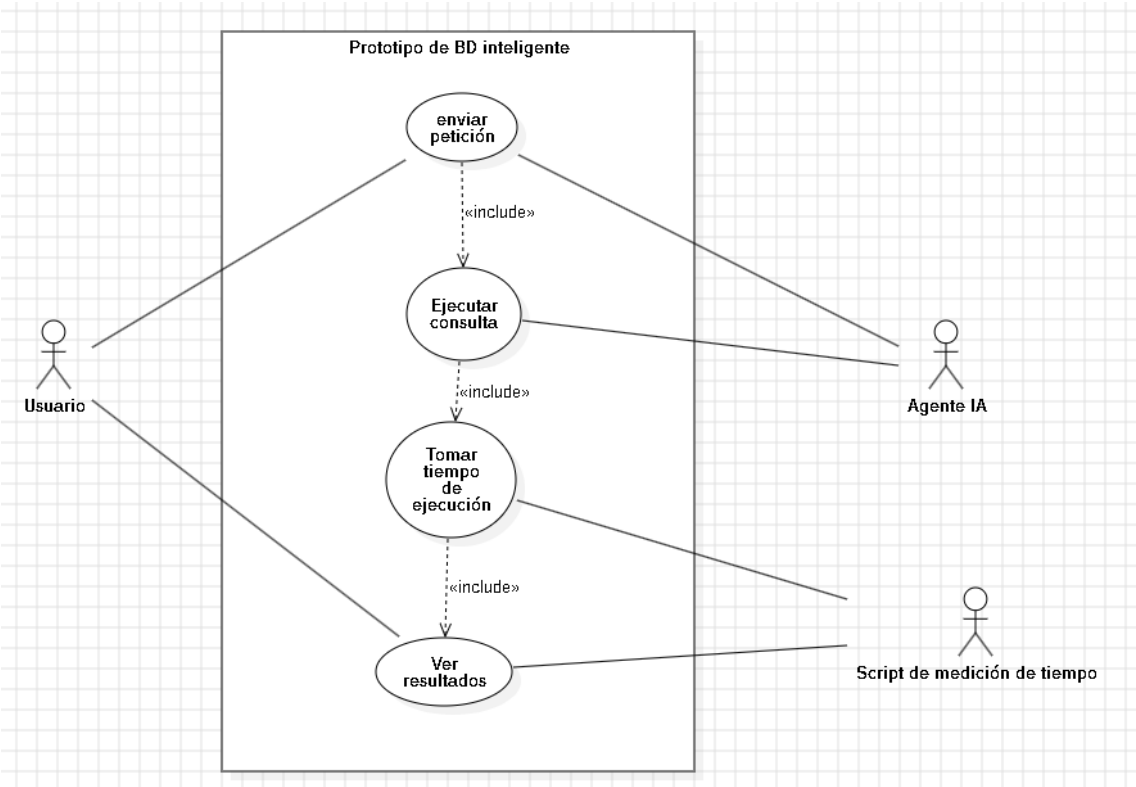


Figura 22 Diagrama de caso de uso 03 ver resultados con medición de tiempo.

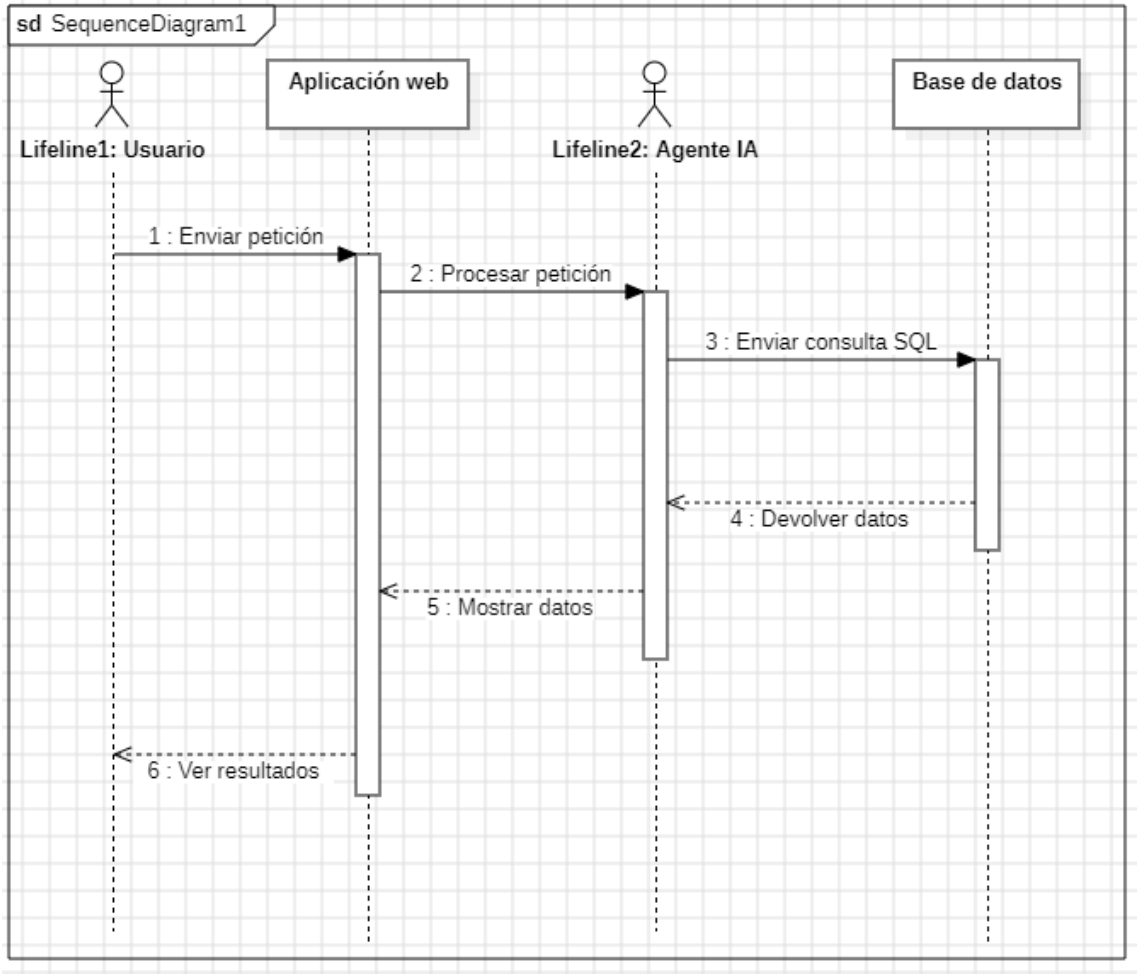


Figura 23 Diagrama de secuencia.

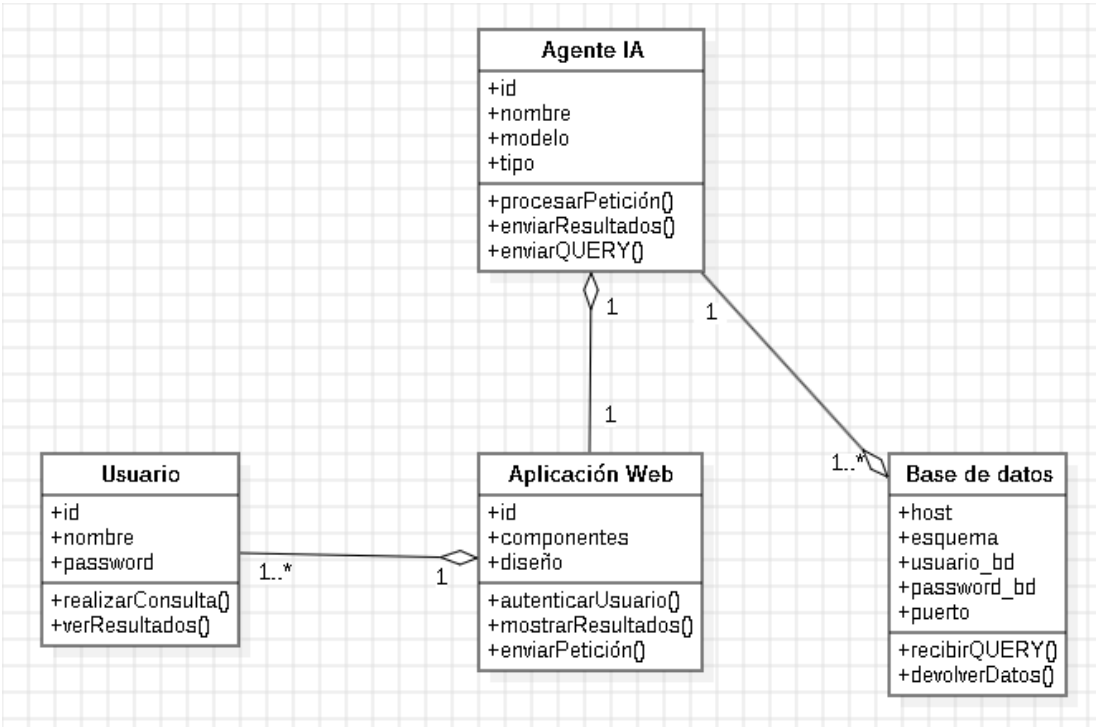


Figura 24 Diagrama de clases.

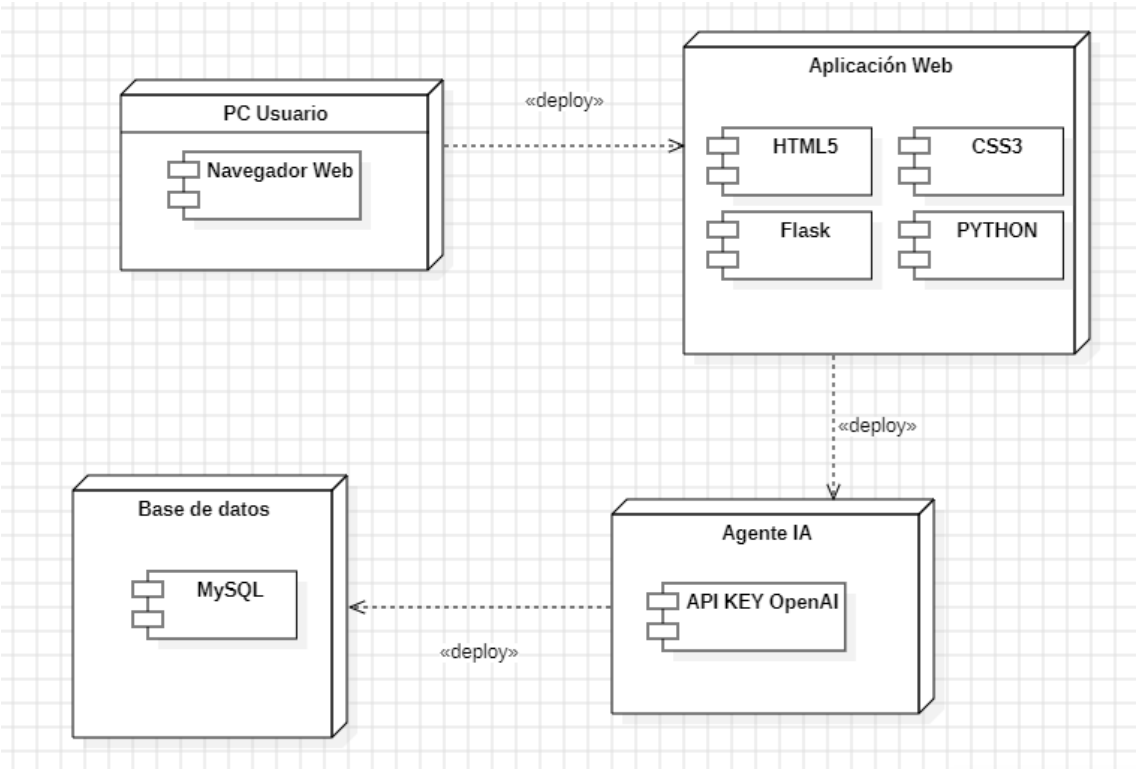


Figura 25 Diagrama de despliegue.

CU-001	**Realizar Petición**		
Descripción	El **Usuario** desea realizar una consulta en lenguaje natural para obtener información de la BD. El sistema y el **Agente IA** procesarán la petición, la traducirán a SQL y mostrarán los registros relevantes al usuario.		
Actores	Usuario, Agente IA		
Pre condiciones	El sistema de Base de Datos Inteligente está operativo.		
Post condiciones	Los resultados de la consulta se muestran al Usuario.		
Secuencia Normal	**#**	**Acción (actor)**	**Reacción (sistema)**
	1	El Usuario inicia el proceso e ingresa su consulta en lenguaje natural.	El sistema recibe la petición y llama al Agente IA.
	2	El Agente IA procede a realizar el caso de uso **Traducir Petición en SQL**.	El sistema ejecuta el caso de uso **Traducir Petición en SQL**.
	3	El Agente IA procede a realizar el caso de uso **Enviar Consulta a la BD**.	El sistema ejecuta el caso de uso **Enviar Consulta a la BD**.
	4	El sistema recibe los registros de la BD y ejecuta el caso de uso **Visualizar Resultados**.	El sistema muestra los resultados al Usuario.
Excepciones	#	**Acción (actor)**	**Reacción (sistema)**
	1	En el caso de que **la petición no pueda ser traducida a SQL** el sistema deberá mostrar un mensaje de error y solicitar reformular la petición.	En el caso de que **la petición no pueda ser traducida a SQL** el sistema deberá mostrar un mensaje de error y finalizar el proceso.
Rendimiento	El sistema deberá realizar la/s acción/es descrita/s en los pasos (1) al (4) en un máximo de **5 segundos**.		
Frecuencia	Este caso de uso se espera que se lleve a cabo unas 4 o 5 veces al día, esto debido a que es un prototipo que usa una API gratuita y es posible que se comience a facturar después de mucho uso.		
Importancia	Vital.		
Urgencia	Inmediatamente.		
Comentarios	Se debe asegurar que el Agente IA pueda manejar diversas estructuras de lenguaje natural.		

Figura 26 Documentación de caso de uso 01.

CU-002	**Ejecutar Consulta SQL**		
Descripción	El **Agente IA** toma una petición en lenguaje natural ya recibida, la analiza, construye la consulta SQL correspondiente y la ejecuta en la **BD** para obtener los registros solicitados.		
Actores	Agente IA, BD		
Pre condiciones	El Agente IA ha recibido una petición válida. La BD está disponible y accesible.		
Post condiciones	La BD ha devuelto un conjunto de registros relevantes al Agente IA.		
Secuencia Normal	**#**	**Acción (actor)**	**Reacción (sistema)**
	1	El Agente IA inicia el caso de uso **Analizar Petición**.	El sistema ejecuta el caso de uso **Analizar Petición**.
	2	El Agente IA procede a realizar el caso de uso **Construir Consulta SQL** basándose en el análisis.	El sistema genera la cadena SQL a partir de la petición.
	3	El Agente IA procede a realizar el caso de uso **Ejecutar Consulta SQL** enviando la cadena SQL a la BD.	El sistema envía la consulta SQL a la BD y espera la respuesta.
	4	La BD ejecuta la consulta y procede a realizar el caso de uso **Devolver Registros**.	El sistema recibe los registros de la BD y los transfiere al Agente IA.
Excepciones	#	**Acción (actor)**	**Reacción (sistema)**
	1	En el caso de que **el análisis de la petición no encuentre términos válidos en el esquema de la BD** el sistema deberá detener la construcción de la consulta y notificar al Agente IA.	En el caso de que **el análisis de la petición no encuentre términos válidos en el esquema de la BD** el sistema deberá devolver un código de error al Agente IA.
	2	En el caso de que **la consulta SQL generada contenga errores de sintaxis** el sistema deberá {notificar el error al Agente IA}.	En el caso de que **la consulta SQL generada contenga errores de sintaxis** el sistema deberá detener la ejecución y devolver un mensaje de error de sintaxis al Agente IA.
Rendimiento	El sistema deberá realizar la/s acción/es descrita/s en los pasos (2) al (4) en un máximo de **3 segundos**.		
Frecuencia	Este caso de uso se espera que se lleve a cabo unas 4 o 5 veces al día, esto debido a que es un prototipo que usa una API gratuita y es posible que se comience a facturar después de mucho uso.		
Importancia	Vital		
Urgencia	Inmediatamente		

Figura 27 Documentación de caso de uso 02.

CU-003	Ver Resultados con Medición de Tiempo		
Descripción	El **Usuario** envía una petición y el sistema registra el tiempo que le toma al **Agente IA** ejecutar la consulta y devolver los resultados. El **Script de Medición de Tiempo** es crucial para este prototipo.		
Actores	Usuario, Agente IA, Script de Medición de Tiempo		
Pre condiciones	El Agente IA está activo. El Script de Medición de Tiempo está listo para iniciar.		
Post condiciones	El Usuario visualiza los resultados junto con el tiempo exacto de ejecución.		
Secuencia Normal	**#**	**Acción (actor)**	**Reacción (sistema)**
	1	El Usuario inicia el proceso enviando la petición.	El sistema ejecuta el caso de uso **Enviar Petición** (al Agente IA).
	2	El sistema inicia el Script de Medición de Tiempo y ejecuta el caso de uso **Ejecutar Consulta** (por el Agente IA).	El sistema y el Agente IA procesan la petición en segundo plano.
	3	Una vez ejecutada la consulta, el Script de Medición de Tiempo realiza el caso de uso **Tomar Tiempo de Ejecución**.	El sistema registra el tiempo transcurrido desde el paso 2.
	4	El Usuario procede a realizar el caso de uso **Ver Resultados**.	El sistema muestra los datos obtenidos y el tiempo de ejecución registrado.
Excepciones	#	**Acción (actor)**	**Reacción (sistema)**
	1	En el caso de que **el Script de Medición de Tiempo falle al iniciar** el sistema deberá proceder sin la medición.	En el caso de que **el Script de Medición de Tiempo falle al iniciar** el sistema deberá continuar el proceso, pero el tiempo de ejecución será reportado como nulo.
Rendimiento	El sistema deberá realizar la/s acción/es descrita/s en los pasos [1] al [4] en un máximo de **4 segundos**.		
Frecuencia	Este caso de uso se espera que se lleve a cabo unas 4 o 5 veces al día, esto debido a que es un prototipo que usa una API gratuita y es posible que se comience a facturar después de mucho uso.		
Importancia	Vital		
Urgencia	Hay presión para el desarrollo del prototipo.		
Comentarios	El tiempo registrado debe ser preciso y representar únicamente el tiempo de procesamiento y ejecución de la consulta, excluyendo el tiempo de renderizado de la interfaz.		

Figura 28 Documentación de caso de uso 03.

Apéndice C. Captura de pantalla de los datos en bruto del dataset utilizado.

En este apéndice, se puede apreciar cómo se ven los datos dentro del dataset de “Admitidos Unillanos” del sitio web de datos abiertos del gobierno de Colombia, periodo 2021-2025 con un total de 11928 registros de aspirantes.

11917	2,025,"2","ADMITIDOS","F","SOLTERO(S)","08/04/2005 12:00:00 AM","COLOMI
11918	2,025,"2","ADMITIDOS","M","SOLTERO(S)","02/08/2008 12:00:00 AM","COLOM
11919	2,025,"2","ADMITIDOS","F","SOLTERO(S)","12/03/2006 12:00:00 AM","ITALIA",
11920	2,025,"2","ADMITIDOS","M","SOLTERO(S)","10/05/2006 12:00:00 AM","COLOM
11921	2,025,"2","ADMITIDOS","F","SOLTERO(S)","06/24/2006 12:00:00 AM","COLOMI
11922	2,025,"2","ADMITIDOS","F","SOLTERO(S)","07/14/2005 12:00:00 AM","COLOMI
11923	2,025,"2","ADMITIDOS","M","SOLTERO(S)","04/04/2007 12:00:00 AM","COLOM
11924	2,025,"2","ADMITIDOS","F","SOLTERO(S)","03/27/2006 12:00:00 AM","COLOMI
11925	2,025,"2","ADMITIDOS","M","SOLTERO(S)","01/28/2008 12:00:00 AM","COLOM
11926	2,025,"2","ADMITIDOS","M","SOLTERO(S)","03/14/2007 12:00:00 AM","COLOM
11927	2,025,"2","ADMITIDOS","M","SOLTERO(S)","09/13/2004 12:00:00 AM","COLOM
11928	2,025,"2","ADMITIDOS","F","SOLTERO(S)","08/24/2007 12:00:00 AM","COLOMI
11929	

Figura 29 Datos del dataset "Admitidos Unillanos".

Apéndice D. Diccionario de datos de la base de datos inteligente.

En este apéndice, se muestra el diccionario de datos generado en base a la estructura de la base de datos inteligente implementada, en donde se busca visualizar de forma clara y concisa todos y cada uno de los campos disponibles dentro de la base de datos, teniendo en cuenta su nombre, tipo de dato, descripción y su naturaleza en donde se determina si se trata de un campo primario, foráneo u ordinario.

Tabla: colegio

Columna	Tipo de Dato	PK/FK	Nulo (NULL)	Descripción
id_colegio	INT	PK	NO	Identificador único del colegio (Autoincremental).
nombre_colegio	VARCHAR(255)		SÍ	Nombre completo del colegio.
naturaleza	VARCHAR(50)		SÍ	Tipo de naturaleza del colegio (e.g., Público, Privado).
departamento	VARCHAR(100)		SÍ	Departamento donde se encuentra ubicado el colegio.
municipio	VARCHAR(100)		SÍ	Municipio donde se encuentra ubicado el colegio.

Figura 30 Diccionario de datos de la tabla colegio.

Tabla: estudiante

Columna	Tipo de Dato	PK/FK	Nulo (NULL)	Descripción
id_estudiante	INT	PK	NO	Identificador único del estudiante (Autoincremental).
genero	ENUM('M', 'F')		NO	Género del estudiante (Masculino o Femenino).
estado_civil	VARCHAR(50)		SÍ	Estado civil del estudiante.
fecha_nacimiento	TEXT		SÍ	Fecha de nacimiento del estudiante (almacenado como texto).
pais	VARCHAR(100)		SÍ	País de nacimiento del estudiante.
ciudad_nacimiento	VARCHAR(120)		SÍ	Ciudad de nacimiento del estudiante.
ciudad_residencia	VARCHAR(120)		SÍ	Ciudad de residencia actual del estudiante.
id_colegio	INT	FK	SÍ	Clave foránea referenciando a la tabla colegio.

Figura 31 Diccionario de datos de la tabla estudiante.

Tabla: facultad

Columna	Tipo de Dato	PK/FK	Nulo (NULL)	Descripción
id_facultad	INT	PK	NO	Identificador único de la facultad (Autoincremental).
nombre_facultad	VARCHAR(120)		NO	Nombre de la facultad (debe ser único).

Figura 32 Diccionario de datos de la tabla facultad.

Tabla: programa

Columna	Tipo de Dato	PK/FK	Nulo (NULL)	Descripción
id_programa	INT	PK	NO	Identificador único del programa académico (Autoincremental).
snies	VARCHAR(20)		SÍ	Código SNIES del programa académico.
nombre_programa	VARCHAR(200)		NO	Nombre completo del programa académico.
id_facultad	INT	FK	NO	Clave foránea referenciando a la tabla facultad.

Figura 33Diccionario de datos de la tabla programa.

Tabla: sede

Columna	Tipo de Dato	PK/FK	Nulo (NULL)	Descripción
id_sede	INT	PK	NO	Identificador único de la sede (Autoincremental).
nombre_sede	VARCHAR(255)		NO	Nombre de la sede de la universidad (debe ser único).

Figura 34 Diccionario de datos de la tabla sede.

Tabla: admision

Columna	Tipo de Dato	PK/FK	Nulo (NULL)	Descripción
id_admision	INT	PK	NO	Identificador único del registro de admisión (Autoincremental).
anio	INT		SÍ	Año en que se realiza la admisión.
periodo	INT		SÍ	Periodo académico de la admisión (e.g., 1 o 2).
tipo_hecho	VARCHAR(50)		SÍ	Tipo de hecho de la admisión (e.g., Admitido, Matriculado, etc.).
id_estudiante	INT	FK	NO	Clave foránea referenciando a la tabla estudiante.
id_programa	INT	FK	NO	Clave foránea referenciando a la tabla programa.
id_sede	INT	FK	NO	Clave foránea referenciando a la tabla sede.
modalidad	VARCHAR(50)		SÍ	Modalidad de estudio (e.g., Presencial, Distancia).
metodologia	VARCHAR(120)		SÍ	Metodología del programa (e.g., Diurna, Nocturna).

Figura 35 Diccionario de datos de la tabla admisión.

Apéndice E. Script de inserción de datos.

Este apéndice representa parte del código fuente del script utilizado para el proceso de inserción automática de los datos del dataset dentro de la base de datos inteligente.



```
def ensure_table():
    conn = get_connection()
    try:
        cur = conn.cursor()
        cur.execute(DDL)
        conn.commit()
    finally:
        cur.close()
        conn.close()


def insert_many(rows):
    """Inserta múltiples filas. Cada fila debe ser una tupla con el siguiente orden:
    (anio, periodo, tipo_hecho, genero, estado_civil, fecha_nacimiento, pais,
     ciudad_nacimiento, ciudad_residencia, departamento_colegio, municipio_colegio,
     colegio, naturaleza_colegio, snies, programa, facultad, modalidad, metodologia, sede)
    """
    conn = get_connection()
    try:
        cur = conn.cursor()
        sql = f"""
            INSERT INTO {TABLE_NAME} (
                anio, periodo, tipo_hecho, genero, estado_civil, fecha_nacimiento, pais,
                ciudad_nacimiento, ciudad_residencia, departamento_colegio, municipio_colegio,
                colegio, naturaleza_colegio, snies, programa, facultad, modalidad, metodologia, sede
            ) VALUES (
                %s, %s, %s, %s, %s, %s, %s,
                %s, %s, %s, %s,
                %s, %s, %s, %s, %s, %s, %s, %s
            )
        """
        cur.executemany(sql, rows)
        conn.commit()
```

Figura 36 Fragmento de código del script de inserción de datos.

Apéndice F. Mockups de la interfaz gráfica de usuario.

Este apéndice muestra el diseño conceptual de la interfaz gráfica de usuario, dando un primer acercamiento de cómo se vería la aplicación una vez terminada, en los diseños de los mockups se puede apreciar como seria la interacción entre el usuario y el prototipo de bases de datos inteligente. En donde una vez el usuario digita su pregunta en forma de sentencia NQL, se proceden a visualizar los datos traídos por una sentencia SQL.

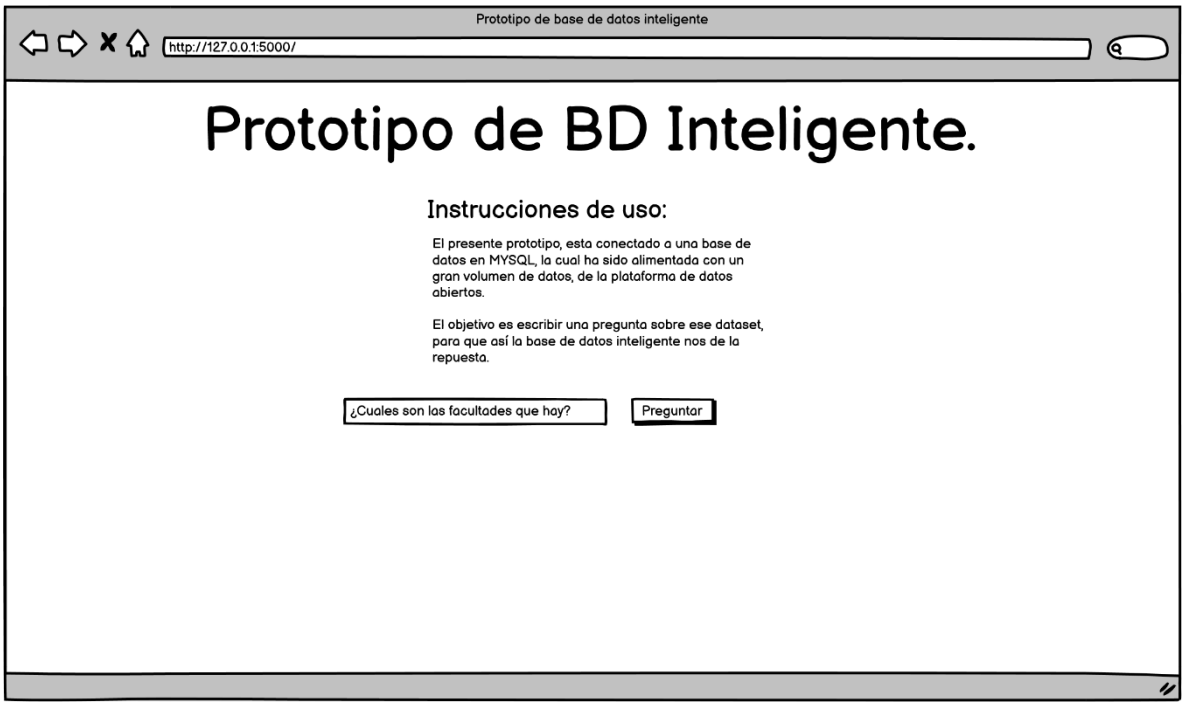


Figura 37 Mockup 01 petición de usuario.



Figura 38 Mockup 02 respuesta del prototipo.

Apéndice G. Técnica de Role Prompting.

Este apéndice, muestra la aplicación de la técnica de Role Prompting, una estrategia que puede ser usada por un usuario, para que una IA de una serie de respuestas más acertadas y puntuales. Lo que se busca es que por medio del prompt usado por el usuario, se le indique a la IA que debe asumir un rol en específico, con la finalidad de que la IA entienda que sus respuestas se deben limitar únicamente al campo de acción del rol que está asumiendo.

```
    pregunta = data.get('pregunta', '').strip()

    if not pregunta:
        return jsonify({"respuesta": "Por favor, ingresa una pregunta válida."}), 400

    try:

        prompt = f"""
Eres un asistente experto en SQL.
Trabajas con la base de datos unillanos_admitidos, que contiene las siguientes tablas y columnas:

TABLA: estudiante
- id_estudiante
- genero
- estado_civil
```

Figura 39 Fragmento de código aplicando la técnica de role promting a un agente IA.

Apéndice H. Instrumentos de medición.

Este apéndice representa la implementación de los instrumentos de medición implementados para el registro de los tiempos de ejecución de las peticiones hechas por los usuarios en NQL al prototipo funcional de bases de datos inteligente.

En donde por medio de la integración de una nueva tabla dentro de la base de datos y un script para medir y almacenar los resultados obtenido, se logra crear un histórico de registros de consultas NQL, con la finalidad de que sea más fácil comparar los resultados obtenidos con los tiempos de ejecución de una base de datos tradicional.

```
    cursor.execute("""
        SELECT id, pregunta, consulta_sql, tiempo_ejecucion, fecha
        FROM consulta_log
        ORDER BY fecha DESC
    """)

    logs = cursor.fetchall()
    cursor.close()
    conn.close()

    return render_template( template_name_or_list: "logs.html", logs=logs)


if __name__ == "__main__":
    app.run(debug=True)
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	pregunta	consulta_sql	tiempo_ejecucion	
1	cuantas facultades hay?	SELECT COUNT(*) FROM facultad	0.001947	2
2	que programas hay?	SELECT * FROM programa	0.007436	2
3	¿Cuántas mujeres había inscritas en el program...	SELECT COUNT(*) FROM estudiante e JOIN ad...	0.060284	2
4	cuantas mujeres hay en sistemas en el 2025	SELECT COUNT(*) FROM estudiante e JOIN ad...	0.002417	2

Figura 40 Registros de tiempos de ejecución de peticiones NQL.

Apéndice I. Resultados preliminares.

En este apéndice, se muestran los resultados obtenidos de las pruebas realizadas a la base de datos inteligente, por medio de las consultas 1 y 2mencionadas en la sección de pruebas preliminares de la fase 3 del proceso de desarrollo del prototipo.

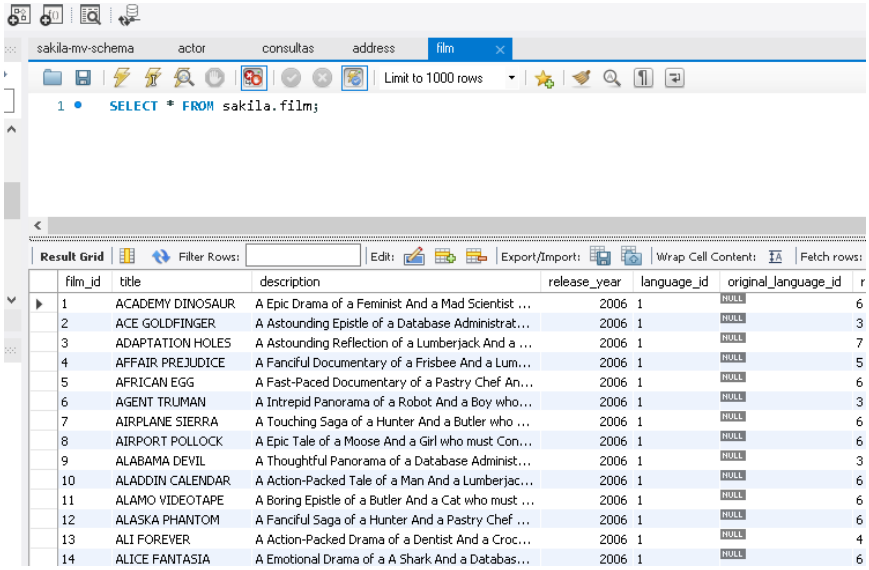


Figura 41 Resultado 1 de consulta 1.

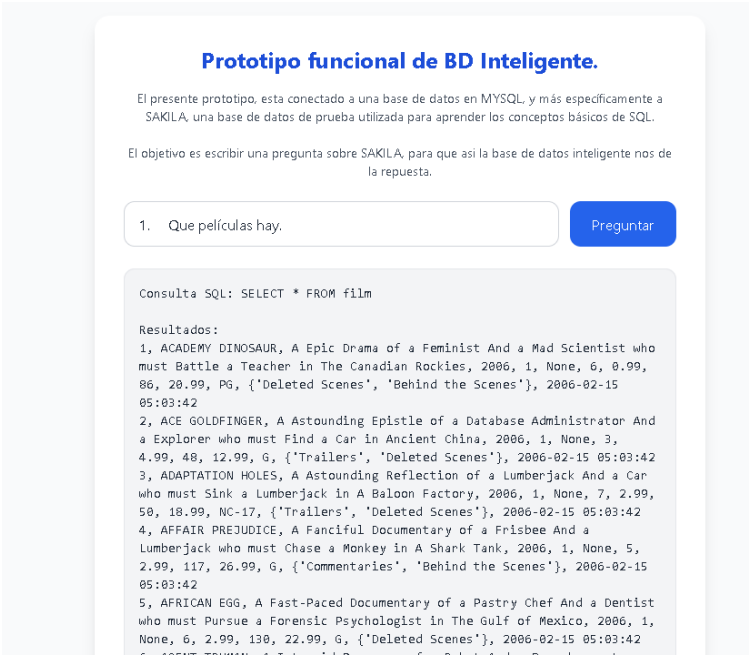


Figura 42 Resultado 2 de consulta 1.

```
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 128-667-829
⏱ Tiempo total de ejecución: 0.01 segundos
127.0.0.1 - - [03/Nov/2025 18:45:49] "GET / HTTP/1.1" 200 -
🎥 Pregunta recibida: qeu peliculas hay
127.0.0.1 - - [03/Nov/2025 18:46:36] "POST /preguntar HTTP/1.1" 200 -
⏱ Tiempo IA: 3.88 segundos
⏱ Tiempo total de ejecución: 3.88 segundos
```

Figura 43 Resultado de medición de tiempo consulta 1.

```
3 •  SELECT COUNT(*) FROM actor;
4
5 •  SELECT c.name AS category,
6    FROM category c
7    LEFT JOIN film_category fc
8    LEFT JOIN film f ON fc.filn
9    GROUP BY c.category_id
```

esult Grid | Filter Rows:

COUNT(*)
200

Figura 44 Resultado 1 de consulta 2.



Figura 45 Resultado 2 de consulta 2.

```
127.0.0.1 - - [03/Nov/2025 19:10:53] "GET / HTTP/1.1" 200 -
 Tiempo total de ejecución: 0.00 segundos
127.0.0.1 - - [03/Nov/2025 19:10:55] "POST /preguntar HTTP/1.1" 200 -
🕵️ Pregunta recibida: cuantos actores son.
⏱️ Tiempo consulta MySQL: 0.11 segundos
 Tiempo total de ejecución: 0.11 segundos
```

Figura 46 Resultado de medición de tiempo consulta 2.

Apéndice J. Consultas NQL usadas para la generación de métricas de precisión.

En este apéndice, se muestran las consultas NQL hechas al prototipo funcional de bases de datos inteligente, en donde a partir de su uso, se logra determinar tres métricas de precisión al modelo de IA que se está usando.

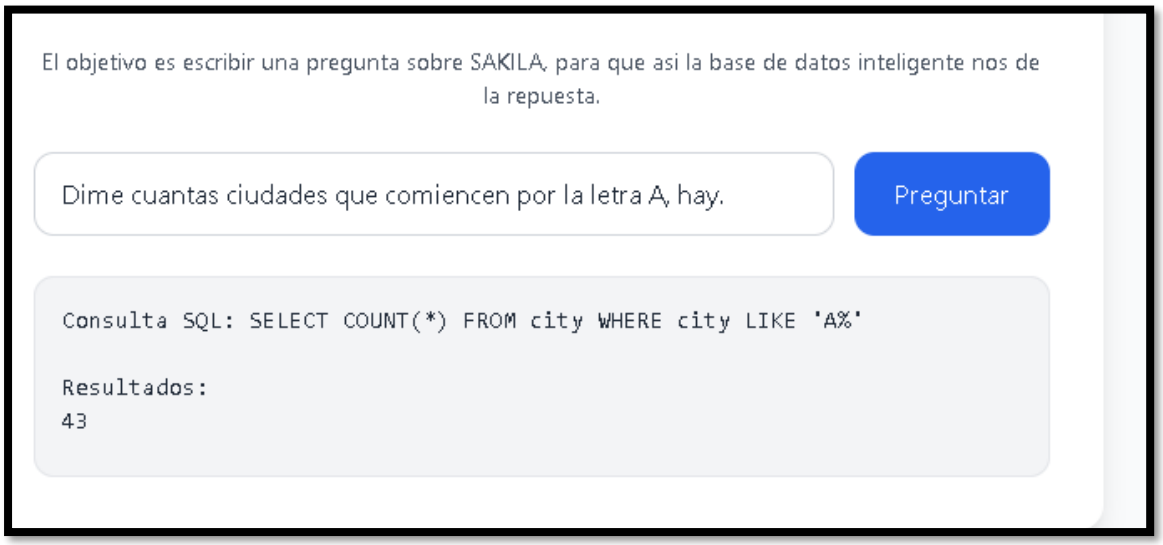


Figura 47 Consulta NQL nro. 1 usada para la generación de métricas de precisión.

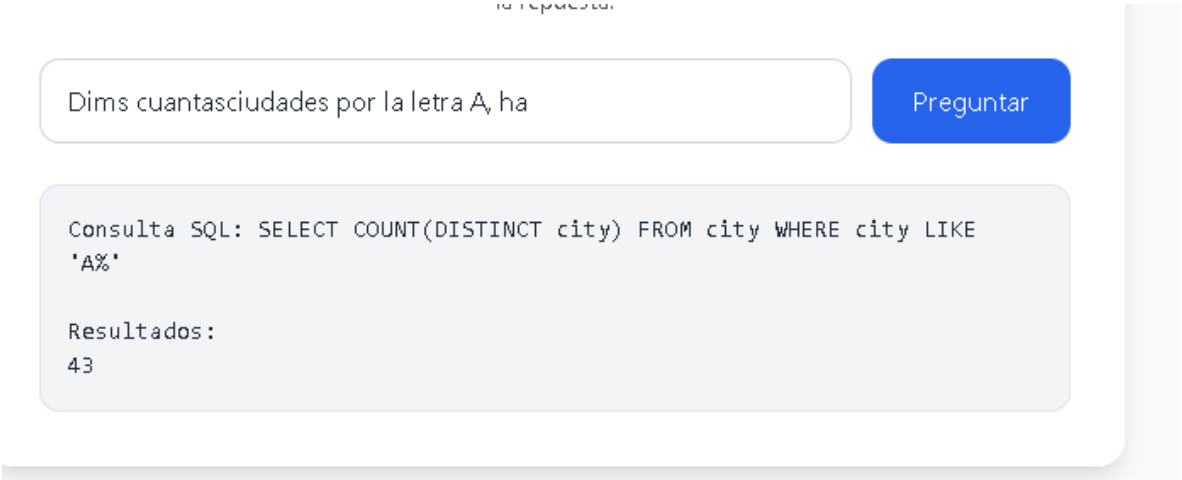


Figura 48 Consulta NQL nro. 2 usada para la generación de métricas de precisión.

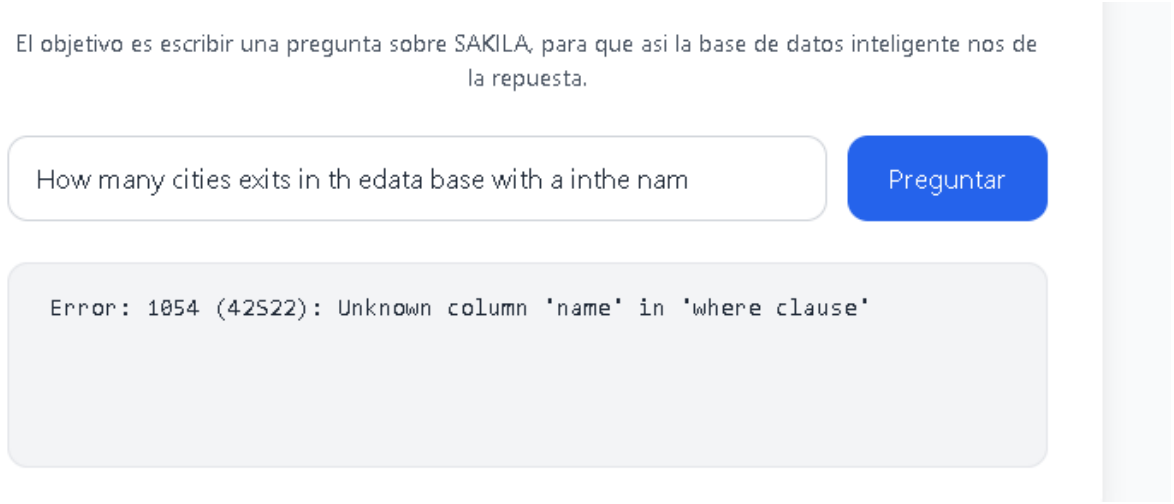


Figura 49 Consulta NQL nro. 3 usada para la generación de métricas de precisión.

Apéndice K. Tabla de consolidado de métricas.

En este apéndice, se consolida todos los resultados obtenidos en base a la aplicación de las métricas formuladas en los resultados finales de este trabajo, por medio de una tabla la cual nos permite generar una serie de indicadores, que nos permiten determinar con exactitud la

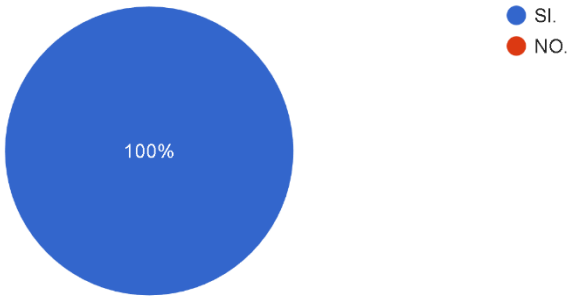
Tabla 8 métricas aplicadas a consultas NQL y SQL.

diferencia entre los niveles de rendimiento de una base de datos inteligente con el de una base de datos tradicional.

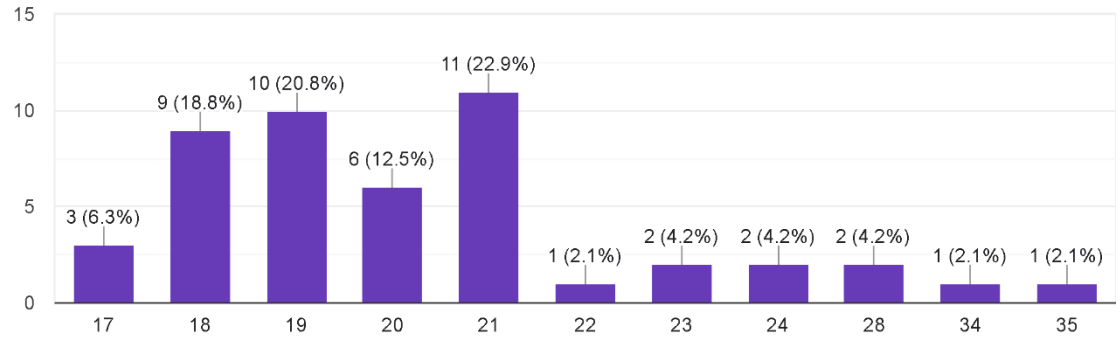
ID CONSULTA	CONSULTA NQL	CONSULTA SQL	T. RESPUESTA X CONSULTA DIGITA		DIFERENCIA EN TIEMPOS DE	PORCENTAJE DE MEJORA (%)	RESULTADO DE LA CONSULTA	
			BD. INTELIGENTE (sg)	BD. TRADICIONAL (sg)			CORRECTO	INCORRECTO
1	cuantas facultades hay?	SELECT COUNT(*) FROM facultad	2,20	2,40	0,20	8,33	1	
2	que programas hay?	SELECT * FROM programa	1,94	1,70	0,24	-14,12	1	
3	¿Cuántas mujeres había inscritas en el programa de ingenieria de sistemas en el 2023?	SELECT COUNT(*) FROM estudiante e JOIN admision a ON e.id_estudiante = a.id_estudiante JOIN programa p ON a.id_programa = p.id_programa WHERE e.genero = 'F' AND p.nombre_programa = 'ingenieria de sistemas' AND a.anio = 2023	3,20	6,30	3,10	49,21	1	
4	cuantas mujeres hay en sistemas en el 2025	SELECT COUNT(*) FROM estudiante e JOIN admision a ON e.id_estudiante = a.id_estudiante JOIN programa p ON a.id_programa = p.id_programa WHERE e.genero = 'Femenino' AND p.nombre_programa = 'Sistemas' AND a.anio = 2025	2,50	7,20	4,70	65,28	1	
5	dame el top 3 de programas con mas mujeres en el 2024-1	SELECT p.nombre_programa, COUNT(*) AS cantidad_mujeres FROM estudiante e JOIN admision a ON e.id_estudiante = a.id_estudiante JOIN programa p ON a.id_programa = p.id_programa WHERE e.genero LIKE '%F%' AND a.anio = 2024 AND a.periodo = 1 GROUP BY p.nombre_programa ORDER BY cantidad_mujeres DESC LIMIT 3	3,30	8,12	4,82	59,36	1	
6	dame un listado de los cinco colegios con mas aspirantes	SELECT c.nombre_colegio, COUNT(a.id_estudiante) AS total_aspirantes FROM colegio c JOIN estudiante e ON c.id_colegio = e.id_colegio JOIN admision a ON e.id_estudiante = a.id_estudiante GROUP BY c.nombre_colegio ORDER BY total_aspirantes DESC LIMIT 5	2,90	12,30	9,40	76,42	1	
7	dame los tres programas con mas aspirantes en el 2023-2	SELECT p.nombre_programa, COUNT(a.id_admision) AS aspirantes FROM programa p JOIN admision a ON p.id_programa = a.id_programa WHERE a.anio = 2023 AND a.periodo = 2 GROUP BY p.nombre_programa ORDER BY aspirantes DESC LIMIT 3	3,60	13,80	10,20	73,91	1	
8	dame un listado de las primeras cinco ciudades de donde vienen los aspirantes	SELECT DISTINCT ciudad_nacimiento FROM estudiante LIMIT 5	3,10	2,80	0,30	-10,71	1	
9	dame un listado de todas las tablas en la bd	SELECT TABLE_NAME FROM information_schema.tables WHERE table_schema = 'unillanos_admitidos'	2,40	3,90	1,50	38,46	1	
10	cual es la facultad que mas tiene programas	SELECT f.nombre_facultad FROM facultad f JOIN programa p ON f.id_facultad = p.id_facultad GROUP BY f.nombre_facultad ORDER BY COUNT(p.id_programa) DESC LIMIT 1	2,70	3,80	1,10	28,95	1	
						TOTAL R. CONSU	10	0
						% EXACTITUD:	100	

Apéndice L. Encuesta aplicada para la toma de datos perceptuales.

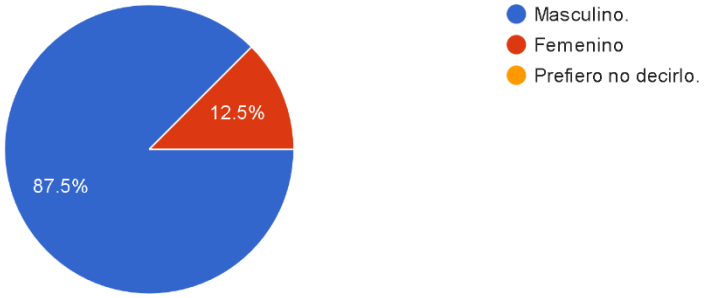
¿Aceptas participar en este estudio y permitir el uso de tus respuestas con fines académicos?
48 respuestas



1. Edad.
48 respuestas

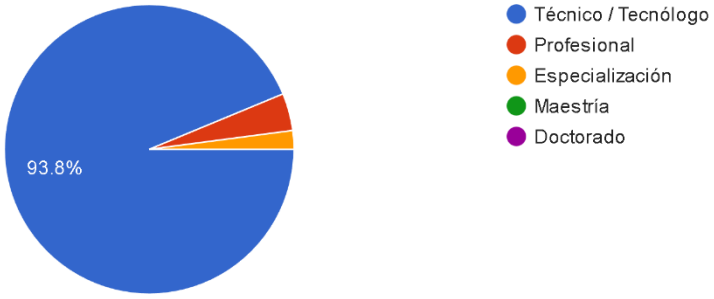


2. Género.
48 respuestas



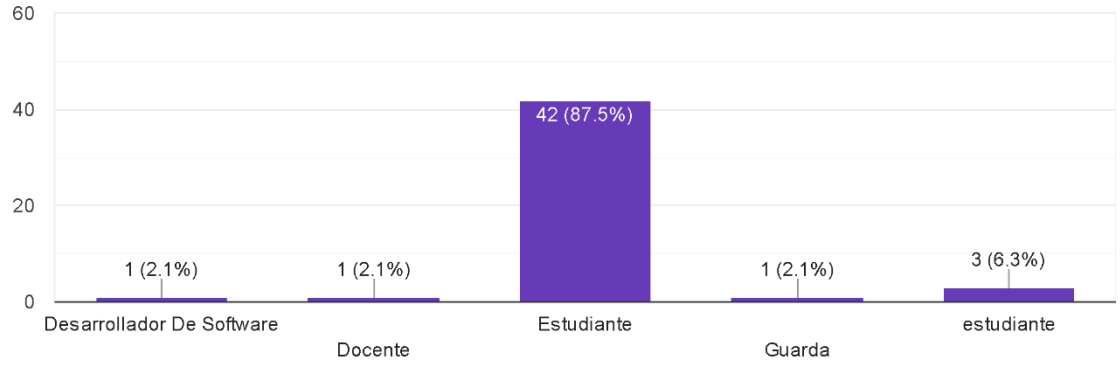
3. Nivel Educativo.

48 respuestas



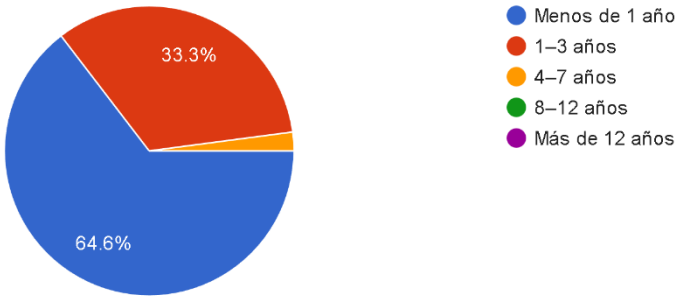
4. Cargo o rol actual:

48 respuestas



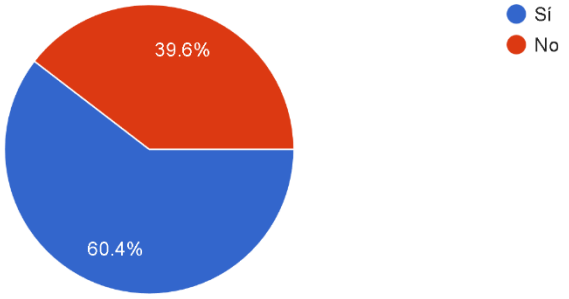
5. Años de experiencia en bases de datos o TI:

48 respuestas



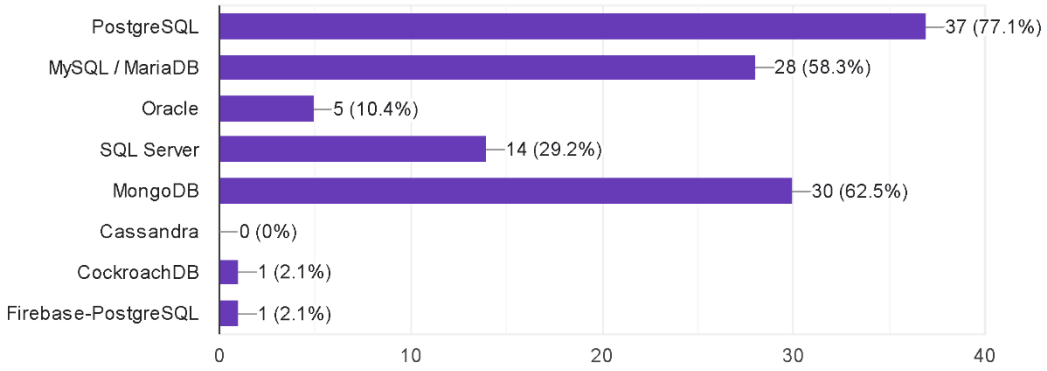
6. ¿Trabajas actualmente con sistemas gestores de bases de datos (SGBD)?

48 respuestas



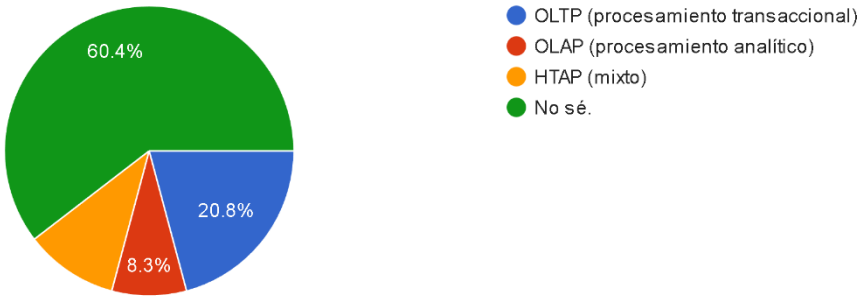
7. ¿Qué sistema gestor de bases de datos utilizas con mayor frecuencia?

48 respuestas



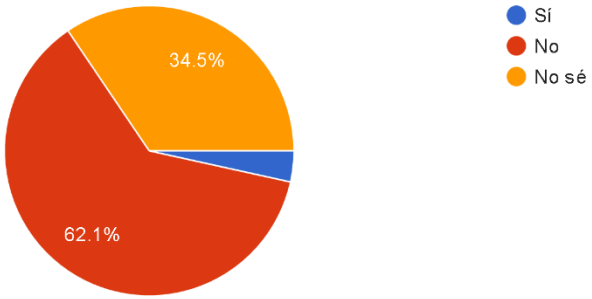
8. Tipo de carga dominante en tu entorno:

48 respuestas



9. ¿Has utilizado funciones automáticas o autónomas (autotuning, recomendaciones, ML integrado) en algún SGBD?

29 respuestas



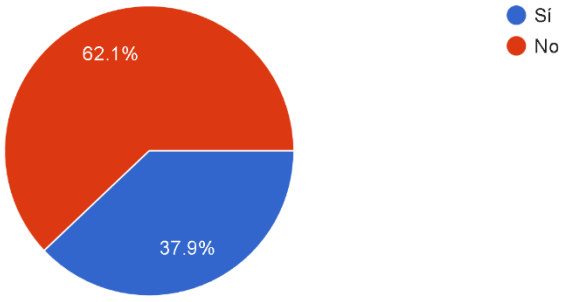
10. Si respondiste "Sí", indica cuáles y en qué entorno

7 respuestas

Si
No
N.A
no se
NA
-
Linux - Arch

11. ¿Has usado consultas en lenguaje natural (NQL) o interfaces conversacionales para acceder a datos?

29 respuestas



Sección 4: Preguntas abiertas

12. ¿Cuáles consideras que son las principales ventajas de las bases de datos inteligentes?

11 respuestas

Las bases de datos inteligentes permiten automatizar el análisis de información, optimizar el rendimiento, ofrecer resultados personalizados y mejorar la toma de decisiones en tiempo real.

Automatización de tareas: Reducen la necesidad de intervención manual en la gestión y el análisis de datos.
* Análisis avanzado: Permiten descubrir patrones y tendencias complejas que serían difíciles de identificar con métodos tradicionales.
* Personalización: Facilitan la adaptación de la información a las necesidades específicas de cada usuario.
* Mejora en la toma de decisiones: Proporcionan información más precisa y relevante para la toma de decisiones estratégicas.
* Optimización de recursos: Permiten identificar áreas de mejora y optimizar el uso de los recursos disponibles.

La automatización

Tiene muchas ventajas ya que ahorra trabajo y son mas practicas

Las principales ventajas de las bases de datos inteligentes son la toma de decisiones más rápida y eficiente, la personalización de servicios y la automatización de procesos.

13. ¿Cuáles son las principales limitaciones o riesgos que observas?

11 respuestas

Sus principales riesgos son el alto costo, la dependencia tecnológica, los posibles sesgos algorítmicos y los problemas de privacidad o seguridad de los datos.

Sesgos en los datos: Si los datos de entrenamiento están sesgados, la base de datos inteligente reproducirá y amplificará esos sesgos.
* Falta de transparencia: Puede ser difícil entender cómo la base de datos inteligente llega a sus conclusiones, lo que dificulta la identificación y corrección de errores.
* Riesgos de privacidad: La recopilación y el análisis de grandes cantidades de datos pueden plantear problemas de privacidad si no se gestionan adecuadamente.
* Dependencia de la tecnología: Una excesiva dependencia de la tecnología puede reducir la capacidad de las personas para tomar decisiones informadas.
* Costos de implementación: La implementación y el mantenimiento de bases de datos inteligentes pueden ser costosos.

La dependencia o rigidez que se puede generar

Enn estos momentos ninguna

14. Describe una experiencia en la que una recomendación automática haya resultado beneficiosa o problemática.

10 respuestas

N/A

Una recomendación útil fue en Spotify, que me ayudó a descubrir música nueva acorde a mis gustos; una problemática fue recibir anuncios de productos que busqué una sola vez.

Beneficiosa: Utilizar un servicio de streaming de música que sugiere canciones basadas en mi historial de escucha me ha permitido descubrir nuevos artistas y géneros que me gustan.

Una experiencia en la que una recomendación automática haya resultado problemática:
* Problemática: Recibir publicidad dirigida basada en mi historial de navegación me ha hecho sentir vigilado y ha generado preocupaciones sobre mi privacidad.

las bases de datos en la nube que las tablas se crean automaticamente

Un ejemplo común beneficioso ocurre con plataformas como Netflix, donde las recomendaciones automáticas ayudan a descubrir nuevas películas relacionadas con las preferencias del usuario, aumentando la satisfacción y facilitando el acceso a contenidos interesantes.

15. En tu opinión, ¿Qué condiciones técnicas o éticas son necesarias para adoptar sistemas de bases de datos inteligentes en entornos reales?

10 respuestas

Se necesitan sistemas seguros, transparentes y éticos, con protección de datos, supervisión humana y tecnología confiable para garantizar un uso responsable.

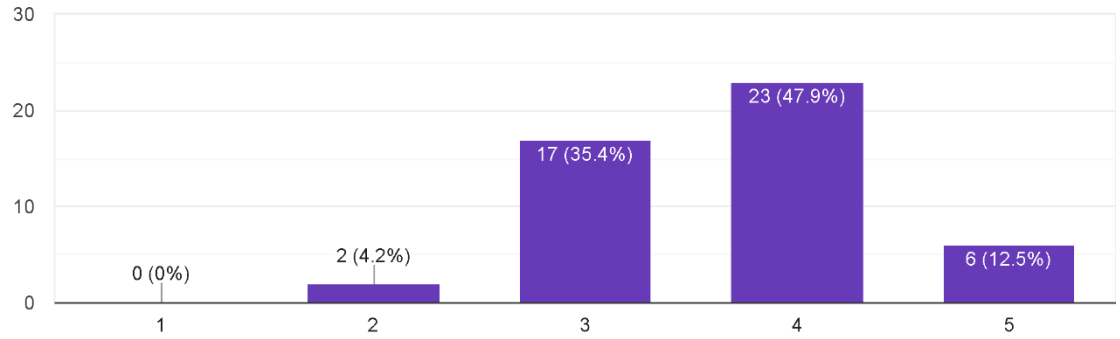
Calidad de los datos: Asegurar que los datos sean precisos, completos y relevantes.
* Transparencia y explicabilidad: Hacer que el funcionamiento de la base de datos inteligente sea comprensible para los usuarios.
* Privacidad y seguridad: Proteger la privacidad de los datos y garantizar su seguridad.
* Responsabilidad y rendición de cuentas: Establecer quién es responsable de las decisiones tomadas por la base de datos inteligente.
* Consideraciones éticas: Evaluar y mitigar los posibles impactos negativos de la base de datos inteligente en la sociedad.

con solo saber lo básico uno se puede desenvolver bien en todo

Para adoptar sistemas de bases de datos inteligentes en entornos reales, son necesarias condiciones técnicas como una gobernanza sólida de datos, infraestructura tecnológica confiable y mecanismos de seguridad avanzados. Éticamente, se requiere transparencia, protección de la privacidad, supervisión humana y normativas que fomenten la equidad y prevengan discriminaciones.

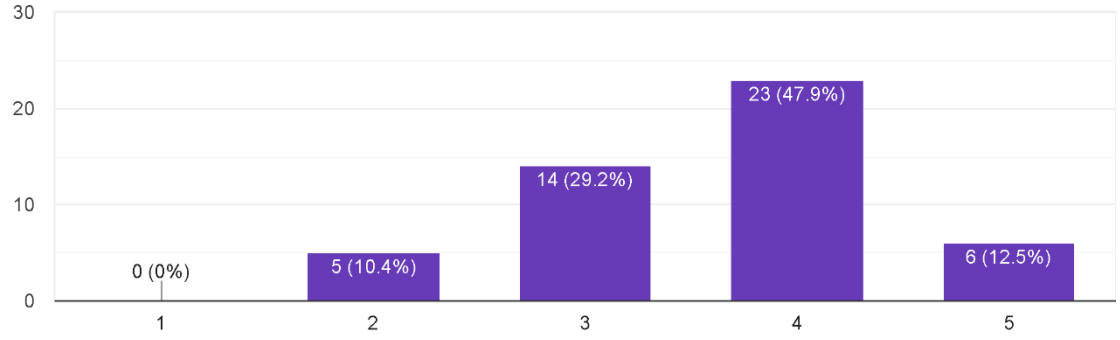
16. Los sistemas con IA producen información más precisa que los sistemas tradicionales.

48 respuestas



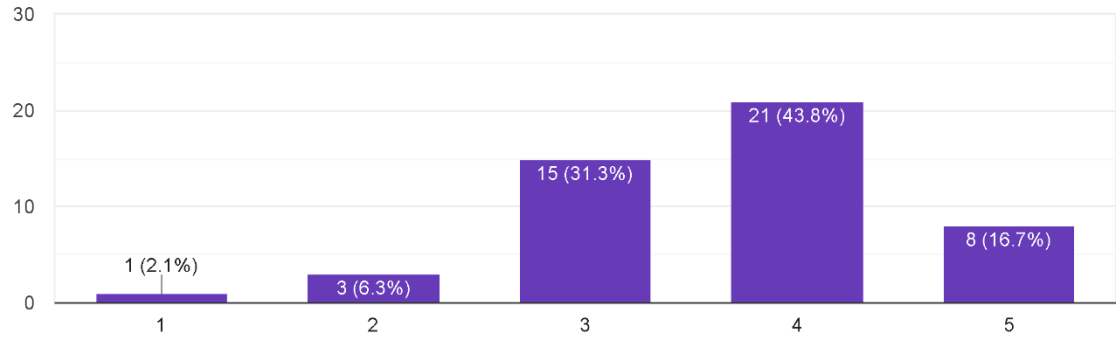
17. Confío en las recomendaciones automáticas (índices, parámetros) de un SGBD inteligente.

48 respuestas



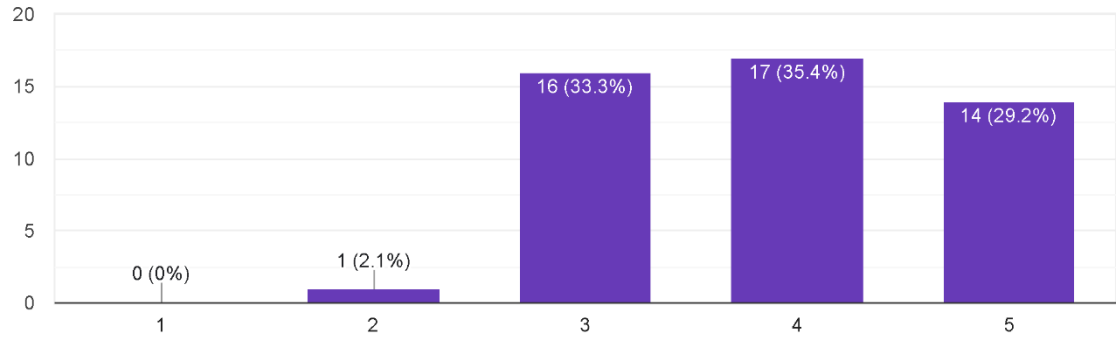
18. He observado mejoras de rendimiento al activar funciones automáticas.

48 respuestas



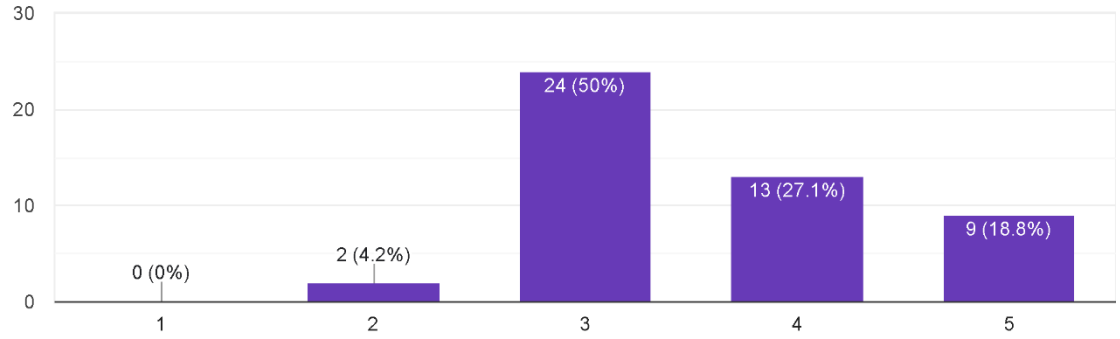
19. Un SGBD inteligente debería reducir la latencia en consultas críticas.

48 respuestas



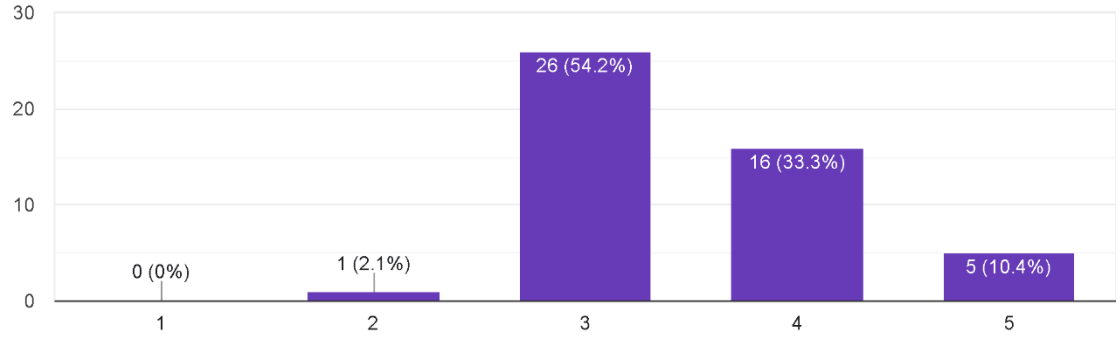
20. El uso de NQL facilita el acceso a datos por usuarios no técnicos.

48 respuestas



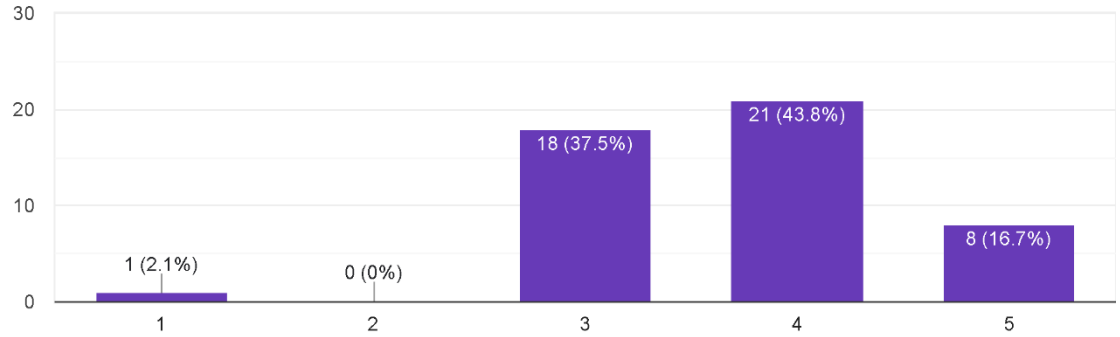
21. In-database ML (aprendizaje automático dentro del SGBD) es más eficiente al reducir procesos ETL.

48 respuestas



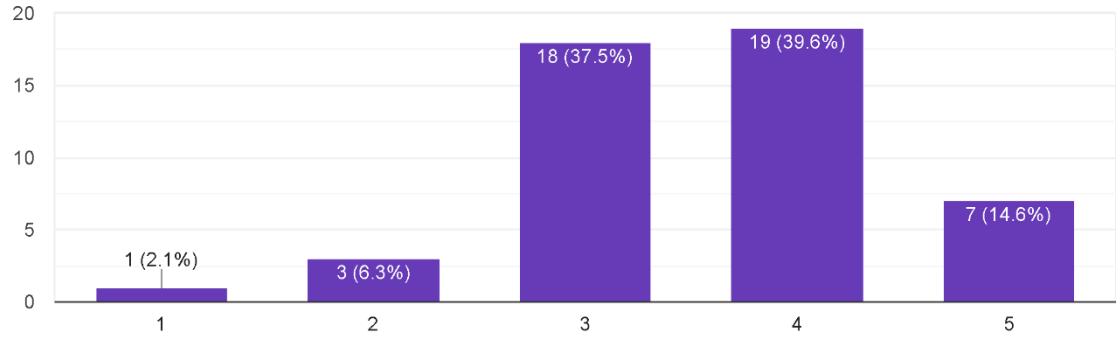
22. Me preocupa la falta de explicabilidad en las decisiones automáticas del sistema.

48 respuestas



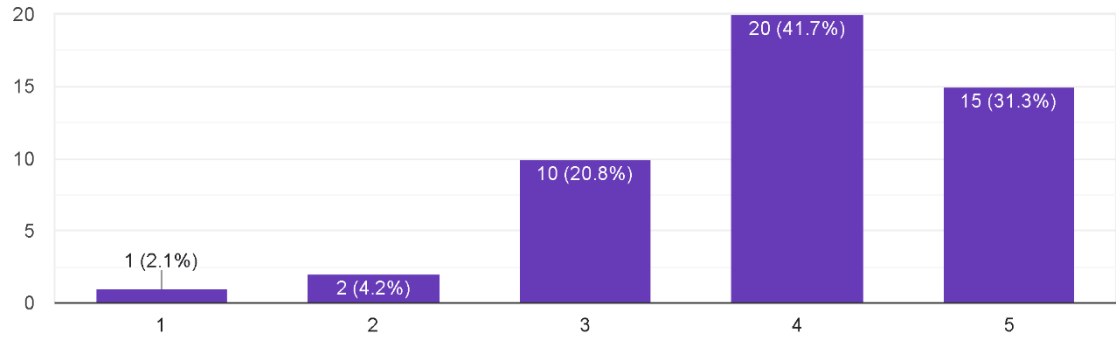
23. La automatización podría reducir la necesidad de intervención humana (DBA).

48 respuestas



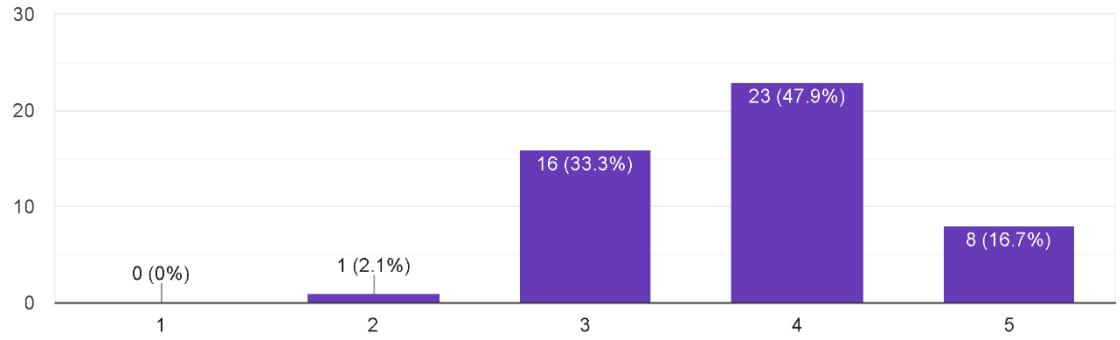
24. Las soluciones con IA deben incluir mecanismos de trazabilidad y auditoría de decisiones.

48 respuestas



25. Recomendaría la adopción de bases de datos inteligentes en mi organización.

48 respuestas



26. Considero que la relación costo-beneficio de estas tecnologías es favorable.

48 respuestas

