

Construcción de un S.G.B.D.R fuertemente acoplado con Redes Neuronales Artificiales para la medición de la eficiencia en operaciones D.M.L con imágenes.

Santiago Alexander Oliva Chingal, santixddd222@gmail.com
Mayer Sebastian Malpud Noguera,
nogueramayer1000@gmail.com Sebastian Cifuentes Oliva,
sebitaz200114@gmail.com

Trabajo de grado para optar al título de ingeniero de sistemas

Asesor:
MSc. Héctor Andrés Mora Paz

Universidad CESMAG
Facultad de Ingeniería
Ingeniería de sistemas
Pasto – Nariño
2025

NOTA DE ACEPTACIÓN

NOMBRE JURADO 1

NOMBRE JURADO 2

San Juan de Pasto, 2025

NOTA DE EXCLUSIÓN

los autores de esta obra son los únicos responsables de las ideas expresadas en ella, y esta no refleja o no compromete la ideología de la Universidad CESMAG.

Dedicatoria (Santiago Oliva)

A mi madre, mi tía y mis abuelos, pilares fundamentales en mi vida, cuyo amor y guía han sido esenciales en mi formación personal. En especial, a mi madre, por su apoyo incondicional en cada etapa de mi vida y por hacer posible la educación que siempre anhelé.

A mi primo, más que un hermano, por su compañía y respaldo en los momentos más difíciles de mi camino académico y personal.

Y a todos los amigos que he encontrado a lo largo de mi formación universitaria, tanto dentro como fuera de la institución, por regalarme momentos inolvidables que han sido una fuente constante de inspiración para seguir adelante.

Dedicatoria (Sebastian Cifuentes)

A mis padres, por su amor incondicional, por ser mi guía y mi fortaleza en cada paso de este camino. Su apoyo, paciencia y sacrificio han sido la base sobre la cual he construido mis sueños.

A mi hermano, por ser mi compañero de vida, por sus palabras de aliento y por demostrarme que juntos podemos superar cualquier desafío.

A mis abuelos, por su sabiduría, enseñanzas y por ser un ejemplo de esfuerzo y perseverancia. Por amor, optimismo, logré avanzar.

Dedicatoria (Mayer Malpud)

A mi familia, por ser mi mayor fuente de amor, apoyo y motivación. A mis padres, por enseñarme con su ejemplo el valor del esfuerzo, la perseverancia y la dedicación. A mis hermanos, por ser inspiración constante y compañeros incondicionales en este camino, mi agradecimiento reside en todo su apoyo.

A mis profesores y mentores, quienes con su guía, paciencia y conocimiento han sido pilares fundamentales en mi formación. Su enseñanza ha dejado una huella imborrable en mi crecimiento académico y personal.

A mis amigos, por su compañía incondicional y por ser un refugio en los momentos de estrés y cansancio. Gracias por cada palabra de aliento, por cada risa y por estar siempre a mi lado en esta travesía.

Agradecimientos

En primer lugar, agradecemos a la Universidad CESMAG por brindarnos el espacio académico que hizo posible nuestra formación profesional. También, a los profesores que marcaron nuestro camino hacia el conocimiento con su sabiduría y dedicación.

En especial, expresamos nuestra gratitud a nuestro asesor, Héctor Mora Paz, un profesional excepcional cuya orientación y visión inspiraron la idea de este trabajo. Sus constantes recomendaciones y correcciones fueron invaluable en nuestro proceso.

Asimismo, extendemos nuestro reconocimiento a los jurados, Jorge Rivera y Carlos Viteri, cuyas observaciones y sugerencias aportaron la rigurosidad y claridad necesarias para el desarrollo de esta investigación.

Por último, queremos agradecer a nuestros compañeros de carrera, quienes, sin esperar nada a cambio, nos ofrecieron valiosas sugerencias y mejoras. Muchas de sus aportaciones quedaron reflejadas en este documento, recordándonos que la unión hace la fuerza. En un ámbito tan importante como el conocimiento científico, la colaboración se convierte en una herramienta poderosa que nos impulsa a alcanzar nuevos descubrimientos.

RESUMEN ANALÍTICO DE ESTUDIO RAE

Facultad: Ingeniería.

Programa: Ingeniería de Sistemas.

Fecha de elaboración: 28 de Octubre del 2024.

Autores de la investigación:

Santiago Alexander Oliva Chingal,
Mayer Sebastian Malpud Noguera,
Sebastian Cifuentes Oliva

Asesor: Héctor Andrés Mora Paz.

Título de la investigación: Construcción de un S.G.B.D.R fuertemente acoplado con Redes Neuronales Artificiales para la medición de la eficiencia en operaciones D.M.L con imágenes.

Descripción: La gestión de datos no estructurados, como las imágenes, en bases de datos relacionales convencionales representa un desafío creciente en el campo de la gestión de la información. Este proyecto se enfoca en abordar dicha problemática mediante la implementación de un sistema de gestión de bases de datos relacionales (SGBDR) que integra redes neuronales artificiales (ANN) para optimizar las operaciones de manipulación de datos con imágenes. Se adopta un enfoque de alto acoplamiento para medir la eficiencia de esta integración y evaluar su impacto en el rendimiento del sistema.

Objetivos

El objetivo general de este proyecto es evaluar y desarrollar un sistema SGBDR-ANN para optimizar la eficiencia en operaciones de manipulación de datos no estructurados, específicamente imágenes. Los objetivos específicos incluyen:

1. Construir un sistema simplificado que facilite el acoplamiento de modelos ANN en un entorno relacional.
2. Implementar rutinas de entrenamiento y evaluación en Python para ANN.
3. Integrar los modelos ANN en el SGBDR para medir su eficiencia en operaciones de manipulación de datos con imágenes.
4. Comparar el rendimiento del sistema propuesto con otros enfoques, como sistemas medianamente acoplados y PostgreSQL, mediante pruebas de rendimiento y escalabilidad.

El marco teórico de este proyecto abarca conceptos fundamentales y antecedentes en las áreas de bases de datos, inteligencia artificial y procesamiento de imágenes. Se han revisado estudios previos sobre sistemas de bases de datos y redes neuronales, destacando la necesidad de herramientas con

alto acoplamiento para mejorar la eficiencia. También se han incluido definiciones y teorías esenciales sobre SGBDR, ANN y técnicas de procesamiento de imágenes.

La metodología de este proyecto se basa en un enfoque experimental para desarrollar y evaluar el sistema SGBDR-ANN. Se han utilizado pruebas de rendimiento y análisis comparativo para recolectar datos y evaluar el sistema. Se han empleado técnicas de entrenamiento y despliegue de ANN en el contexto del sistema desarrollado.

El desarrollo del sistema y modelo ANN se ha enfocado en la creación de un SGBDR que incorpore ANN para el procesamiento de imágenes. Se han seleccionado herramientas y plataformas adecuadas para la implementación del sistema, y se ha diseñado una arquitectura que permita el acoplamiento de ANN en el SGBDR.

Los resultados de las pruebas mostraron que el sistema SGBDR-ANN ofrece una mejora significativa en eficiencia y tiempos de respuesta en comparación con sistemas tradicionales. La integración de ANN en SGBDR permitió una mejor gestión de datos no estructurados y una mayor eficiencia en el procesamiento de imágenes.

En conclusión este proyecto demostró que la integración de ANN en SGBDR puede mejorar significativamente la eficiencia en el procesamiento de imágenes. Los resultados de este estudio pueden aplicarse a otros tipos de datos no estructurados y ofrecen una base para futuras investigaciones en este campo. Se recomiendan futuras investigaciones para mejorar y aplicar este enfoque en diferentes contextos y tipos de datos.

TABLA DE CONTENIDO

Introducción	14
I. PROBLEMA DE INVESTIGACIÓN	16
A. Objeto o tema de estudio	16
B. Línea de investigación	16
C. Sub línea de investigación	16
D. Planteamiento del problema	16
E. Formulación del problema	18
F. Objetivos	19
1). Objetivo general	19
2). Objetivos específicos	19
G. Justificación	19
H. Delimitación	21
II. TÓPICOS DEL MARCO TEÓRICO	20
A. Antecedentes	20
1). Internacionales	20
2). Nacionales	21
3). Regional	24
B. Enunciado de supuestos teóricos de la investigación	24
1). Conceptos clave de base de datos	24
2). Conceptos clave de programación y desarrollo de software	28
3). Conceptos clave de inteligencia artificial (ia/ai)	32
4). Conceptos clave de evaluación y comparación	36
C. Variables del estudio	38
1). Definición nominal de variables	38
2). Definición operativa de variables	39
3). Formulación de la hipótesis	39

4). Hipótesis de investigación	39
5). Hipótesis nula	39
6). Hipótesis alternativa	40
III. METODOLOGÍA	41
A. Paradigma	41
B. Enfoque	41
C. Método	41
D. Tipo de investigación	42
E. Diseño de la investigación	42
F. Población	43
G. Muestra	43
H. Técnicas de recolección de la información	43
1). Validez de la técnica	44
2). Confiabilidad de la técnica	44
I. Instrumento de recolección de datos	44
IV. RESULTADOS DE LA INVESTIGACIÓN	46
A. Desarrollo del sgbdr simplificado	46
1). Inicio del proyecto	46
2). El motor de base de datos (toysql)	47
3). Api e interfaz de toydms	52
4). Detalles de la arquitectura y recolección de requisitos	53
5). Planificación del desarrollo	54
6). Implementación y retrospectiva	57
7). Finalización de toydms	57
B. Desarrollo del modelo de ann	59
1). Entendimiento del negocio	59
2). Entendimiento y preparación de los datos	60

3). Modelado	61
4). Evaluación del modelo	61
5). Despliegue del modelo	61
C. Acoplamiento del modelo “class_model” y toydms	61
D. Pruebas de eficiencia con alto, mediano acoplamiento y postgresql	64
V. ANÁLISIS DE RESULTADOS	70
1). Análisis de eficiencia (tiempos de respuesta)	70
2). Análisis de escalabilidad (cantidad de datos)	73
Conclusiones	74
Recomendaciones	75
Referencias bibliográficas	76
Anexos	83
Manual de usuario	84
Programa ejecutable	85

LISTA DE TABLAS

TABLA I: PRODUCT BACKLOG DEL SISTEMA	52
TABLA II: HISTORIAS DE USUARIO DEL SISTEMA TOYDMS	53
TABLA III: MÉTRICAS DE ENTRENAMIENTO Y PRUEBA	60
TABLA IV: RESULTADOS DE PRUEBAS CON ALTO ACOPLAMIENTO EN TOYDMS	65
TABLA V: RESULTADOS DE PRUEBAS CON MEDIANO ACOPLAMIENTO EN TOYDMS	65
TABLA VI: RESULTADOS DE PRUEBAS CON POSTGRESQL	68
TABLA VII: COMPARATIVA PROMEDIOS TIEMPOS DE RESPUESTA CADA ENFOQUE PROPUESTO	71

LISTA DE FIGURAS

Fig. 1. Ilustra los niveles de abstracción de un SGBD	25
Fig. 2. La estructura de una relación en un SGBDR	26
Fig. 3. La manera gráfica la estructura de un SGBDOR	27
Fig. 4. Representa un perceptrón	33
Fig. 5. El funcionamiento de una CNN	34
Fig. 6. Ecuación exactitud	39
Fig. 7. Arquitectura del procesamiento de consultas	47
Fig. 8. Arquitectura del motor ToySQL	49
Fig. 9. Arquitectura cliente/servidor de ToyDBMS.	51
Fig. 10. Cliente/interfaz de consola de ToyDBMS	57
Fig. 11. Cliente/interfaz web de ToyDBMS	57
Fig. 12. Interfaz principal de ToyDBMS.	58
Fig. 13. conjunto de datos utilizado, donde a, b y c corresponden a cada una de las clases.	59
Fig. 14. Arquitectura de “class_model”.	60
Fig. 15. Diagrama de actividades del funcionamiento del módulo para la inserción.	62
Fig. 16. Diagrama de actividades del funcionamiento del módulo para selección de imágenes.	63
Fig. 17. Script de inserción y consulta para pruebas de alto y mediano acoplamiento.	66
Fig. 18. Implementación de la función y consulta en PostgreSQL	69
Fig. 19. Grafica de rendimiento de los tres enfoques propuestos	71

ANEXOS

Anexo 1: Aval Asesor del proyecto	82
Anexo 2: Manual de usuario del sistema	84

INTRODUCCIÓN

Actualmente, el procesamiento y análisis de datos desempeñan un papel vital en la toma de decisiones en diversos campos, desde la industria hasta la ciencia y la medicina. Aunque los datos actuales se caracterizan por su creciente diversidad en estructura y formato. Se pueden dividir en dos categorías: estructurados y no estructurados.

Los datos estructurados, organizados en tablas y relaciones, facilitan su uso y análisis. Por el contrario, los datos no estructurados, como imágenes, audio, vídeo y texto sin formato carecen de este orden, lo que dificulta el análisis automático. Las herramientas disponibles actualmente se basan en enfoques con distintos niveles de acoplamiento, un concepto que hace referencia al grado de integración entre los módulos del sistema. Existen varios niveles de acoplamiento; en esta investigación se consideran tres de ellos. El primero es el bajo acoplamiento, se trata de soluciones donde los módulos funcionan de manera independiente, es decir, la funcionalidad de uno no afecta al otro. Esto puede darse, por ejemplo, en soluciones externas que no están incluidas en el código del sistema, como módulos que operan en un tercer sistema. El acoplamiento medio, o acoplamiento de estampilla, se caracteriza por el uso de una misma estructura de datos, como funciones definidas o construidas por el usuario dentro del sistema, este es el caso de las User Defined Function (UDF) [1], aunque pueden variar en su implementación. Finalmente, el alto acoplamiento implica una fuerte dependencia entre los módulos, donde en este caso la modificación de uno afecta directamente módulos del núcleo del sistema. Esto es similar a lo que ocurre con las UDF, pero de manera más profunda, ya que pueden ser importadas o construidas directamente en el núcleo. Este concepto es ampliamente reconocido como clave en el diseño de software, ya que se utiliza como una métrica para evaluar la mantenibilidad de un sistema, siendo el bajo acoplamiento lo más deseable. Sin embargo, se propone un enfoque de alto acoplamiento con redes neuronales, ya que la dependencia entre estos dos módulos podría aportar beneficios en una funcionalidad nativa de comparación de imágenes, gracias al mayor control que el modelo podría ejercer dentro del núcleo o motor del SGBD. Actualmente ninguna de las soluciones presenta una función nativa ni de alto acoplamiento a modelos de inteligencia artificial (IA) para su funcionamiento, este es un obstáculo, debido a que las herramientas como las Redes Neuronales Convolucionales (CNN) pueden contribuir significativamente a la medición de la eficiencia mediante un alto acoplamiento en la administración de datos no estructurados, específicamente imágenes, a través de la extracción de características para su comparación.

Se puede destacar el uso de herramientas actuales, las cuales se basan principalmente en el bajo o mediano acoplamiento, como es el caso de "pg_similarity" [3] un UDF que funciona como importación en el sistema PostgreSQL [4], o directamente no se aplican soluciones de este estilo y se opta por sistemas no relacionales.

Para abordar este desafío, se propone un Sistema de Gestión de Bases de Datos Relacional (SGBDR) que integra fuertemente modelos de Redes Neuronales Artificiales (ANN). Esta investigación se enfoca en la arquitectura de este SGBDR y su capacidad para realizar de manera eficiente consultas en datos no estructurados, en particular imágenes. El objetivo es evaluar su eficacia y su impacto en la optimización del procesamiento y análisis de datos no estructurados

I. Problema de investigación

A. Objeto o tema de estudio

La investigación se centra en la arquitectura de un SGBDR con ANN, altamente acopladas, su eficiencia y comparación con enfoques actuales.

B. Línea de investigación

La línea de investigación en la que se desempeña este trabajo es el área de inteligencia artificial (IA) [5], específicamente. Gestión de Datos No Estructurados y Análisis Inteligente. Este enfoque implica la administración de datos mediante herramientas de software, como los SGBDR, y el análisis de estos datos mediante la aplicación de técnicas de IA.

C. Sub línea de investigación

La sublínea de investigación que es abordada en este trabajo es visión artificial. Esta sublínea es tomada centrándose en el área de reconocimiento de patrones e integración de datos no estructurados [5], específicamente imágenes, en un sistema relacional. Este enfoque implica la búsqueda de patrones en imágenes mediante el uso de técnicas de IA, como se mencionó previamente, y su integración en un sistema relacional, como el SGBDR a desarrollar.

D. Planteamiento del problema

El incremento notable de datos multimedia, como imágenes, audio y video, clasificados como datos no estructurados, se ha evidenciado con los avances tecnológicos. Aunque estas tecnologías son adoptadas diariamente por las empresas, estos datos no pueden analizarse de las formas tradicionales, como el uso de archivos de bases de datos o Excel, que aún utilizan muchas empresas. Al carecer de una estructura predefinida, especialmente en el caso de las imágenes, resulta imposible organizarlas en una tabla de Excel o estructuras similares, a diferencia de los registros de productos o ventas. Se estima que entre el 70% y 90% de los datos

actuales son de tipo no estructurado [6, 7, 8], representando la mayoría de los datos en movimiento hoy en día. Esto impulsa a las empresas a buscar métodos eficaces de gestión y análisis de estos datos para mantenerse competitivas [7]. Sin embargo, los sistemas actuales que analizan estos datos utilizan enfoques de bajo o mediano acoplamiento con ANN, siendo este bajo acoplamiento lo que puede ser la causa de complicación en el proceso y de esta forma no aportar en la eficiencia de las consultas. De la misma manera, no existe un SGBDR que pueda manejar este tipo de datos con ANN de manera nativa con un enfoque altamente acoplado para medir su rendimiento y determinar si este alto acoplamiento podría mejorarlo significativamente.

Esta problemática impide analizar eficientemente los datos no estructurados, siendo la eficiencia crucial, especialmente considerando la rapidez con que la mayoría de los datos llegan a los usuarios [6]. La ineficiencia afecta áreas que dependen de consultas o análisis de estos datos, como las secciones que confían en gran medida en la información visual, ralentizando la extracción de conocimiento. Además, la experiencia del usuario es esencial, sobre todo en organizaciones que buscan destacar frente a otras. Cabe destacar también que los enfoques actuales son más complejos de manejar. Esta limitación no solo obstaculiza a los desarrolladores en la creación de sistemas más eficientes y complejos a nivel empresarial, sino también en áreas críticas como la educación e investigación.

En los enfoques y técnicas para abordar este desafío, se observa que ninguno ha integrado un enfoque a un nivel altamente acoplado con las técnicas de ANN, lo que provoca también un alargamiento del proceso mencionado anteriormente. Un ejemplo notable son los Sistemas Gestores de Bases de Datos Objeto-Relacionales (SGBDOR) como ORACLE, que permiten el uso de User Defined Types (UDT) para el tratamiento de imágenes [9]. Sin embargo, esta solución es compleja y requiere una secuencia extensa de pasos, como la declaración manual de características o descriptores de las imágenes para su posterior comparación y consulta. Esta tarea resulta tediosa, especialmente con grandes volúmenes de datos, aumentando la carga de trabajo. Otros enfoques basados en descriptores, como la

metodología MAM [10], utilizan diversas técnicas de extracción de características, incluyendo modelos de Convolutional Neural Networks (CNN) [2]. Sin embargo, muchas de estas etapas no están disponibles en los SGBDR. Un ejemplo común es la extracción de características o descriptores, fundamental para medir la eficiencia y calidad de las consultas. Además de las CNN, otro modelo de IA interesante son las Relational Networks (RN)[11] para realizar esta misma extracción. Aunque estas herramientas son potentes, actualmente no se aprovechan al máximo en los SGBDR existentes, recurriendo a extensiones en lugar de integrarlas directamente en el núcleo del sistema, como se observa en el caso de pg_similarity [3], que ofrece una variedad de algoritmos para la comparación, sin embargo, este enfoque no contempla tecnologías más eficientes como las CNN o RN mencionadas.

La persistencia de esta problemática está generando expectativas de posibles limitaciones en los sistemas actualmente en uso. La ineficiencia y la carga de trabajo asociadas están creando obstáculos y desventajas competitivas tanto para empresas como para investigaciones centradas en la administración de este tipo de datos. La ausencia de un sistema altamente acoplado con técnicas de ANN para evaluar cómo puede mejorar la gestión de estos datos está retrasando y podría seguir retrasando el desarrollo de nuevas tecnologías e investigaciones que requieran una mayor eficiencia en las consultas a bases de datos mediante el lenguaje DML para su implementación.

Es crucial evaluar la eficacia de estas soluciones para determinar si realmente pueden mejorar significativamente las consultas a las bases de datos y así ofrecer una solución a la ineficiencia actual. Esta evaluación es necesaria para medir si estas tecnologías pueden abordar adecuadamente el desafío existente.

E. Formulación del problema

Teniendo en cuenta el planteamiento realizado anteriormente podemos llegar a la siguiente pregunta.

¿En qué medida podría la integración de modelos de ANN altamente acoplados en un SGBDR mejorar el rendimiento del proceso de consulta de imágenes en comparación con los enfoques actuales?

F. Objetivos

1. Objetivo general

Construir un SGBDR fuertemente acoplado con ANN para la medición en la eficiencia en la gestión de operaciones de manipulación de datos (DML) con imágenes.

2. Objetivos específicos

- Desarrollar un SGBDR simplificado para facilitar el acoplamiento de modelos experimentales fuertemente acoplados, mediante un lenguaje de programación de alto nivel.
- Implementar rutinas de ANN para el entrenamiento, evaluación y almacenamiento de modelos inteligentes, a través del lenguaje de programación Python.
- Integrar las rutinas de ANN desarrolladas previamente con el SGBDR construido (SGBDR- AI) para la medición de la eficiencia en la ejecución de operaciones DML con imágenes mediante la comparación de su ejecución en un modelo medianamente acoplado y un modelo altamente acoplado.
- Comparar los tiempos de respuesta entre SGBDR (Postgres) y el SGBDR-AI construido en la ejecución de consultas con imágenes para la medición del alcance de la propuesta con un motor de base de datos industrial, mediante un experimento con proyecciones con resultados de crecimiento exponencial.

G. Justificación

El proyecto tiene como objetivo abordar la falta de un SGBDR que ofrezca un fuerte acoplamiento con ANN para la comparación de imágenes, lo cual constituye un problema con pocas investigaciones previas conocidas en este ámbito. Esta carencia representa un obstáculo significativo, ya que no se puede determinar si este enfoque mejora en términos de eficiencia y escalabilidad en comparación con los SGBDR actuales.

La resolución de esta problemática es esencial para fomentar la innovación en diversas aplicaciones. Un SGBDR con fuerte acoplamiento de las herramientas mencionadas podría permitir un procesamiento de imágenes más eficiente, preciso y escalable, lo que podría

conllevar mejoras sustanciales en la velocidad y simplicidad de las consultas.

La importancia práctica del proyecto radica en su capacidad para mejorar el procesamiento de datos no estructurados, específicamente el procesamiento de imágenes. Este avance podría tener un impacto significativo en sectores como la seguridad, la medicina, el comercio electrónico y la investigación científica, no solo con la realización de consultas, también en el área de minería de este tipo de datos no estructurados [12].

El proyecto propuesto contribuirá al conocimiento en el campo de la ingeniería de sistemas al desarrollar un nuevo SGBDR con ANN altamente acopladas para la comparación de imágenes. Este enfoque innovador aborda una necesidad clave en el procesamiento de imágenes. También tiene el potencial de aplicarse a otros tipos de datos no estructurados.

La característica más destacada del proyecto es la integración nativa de herramientas con ANN en un SGBDR, lo que permite a los usuarios aprovechar las capacidades de la inteligencia artificial sin necesidad de conocimientos especializados, esto se traduce en un sistema más simplificado en cuanto a su forma de uso.

Se espera que el nuevo SGBDR tenga un impacto positivo en el procesamiento de datos no estructurados debido a que puede ser más eficiente y escalable en comparación con los SGBDR existentes. Se puede anticipar un aumento en la productividad e innovación en campos que dependen del procesamiento de datos no estructurados, como es bien sabido, nos encontramos en la época de los “datos 4.0”, donde el tratar estos datos se vuelve fundamental [13]. Los resultados del proyecto podrán influir en la toma de decisiones en la ingeniería de sistemas de varias maneras, como ayudar a diseñar sistemas más efectivos y respaldar decisiones estratégicas relacionadas con la inversión en tecnologías emergentes.

El proyecto de desarrollo de SGBDR con ANN orientado a la clasificación de imágenes tiene el potencial de hacer una contribución significativa al campo de la ingeniería de sistemas al abordar el desafío del procesamiento de datos no estructurados. Este enfoque innovador podría ser la mayor ventaja debido a que puede mejorar la eficiencia de los SGBDR y su implementación en otros sistemas en el futuro.

H. Delimitación

La delimitación se centrará en la medición de la eficiencia de un SGBDR con ANN orientado a la consulta con imágenes.

En el ámbito geográfico no hay una delimitación exacta debido que la investigación puede ser leída y aplicada en todo el mundo en el ámbito tecnológico.

El tiempo estimado para el desarrollo del proyecto es de 12 meses (diciembre de 2023 hasta noviembre de 2024). En el contexto institucional. El proyecto se lleva a cabo en el contexto de la Universidad CESMAG, con acceso a sus recursos y colaboración con el personal académico de la institución.

II. Tópicos del marco teórico

A. Antecedentes

1. Internacionales

A pesar de que se ha llevado a cabo una búsqueda exhaustiva de proyectos relacionados, se ha observado que hay una escasez de iniciativas que propongan una arquitectura similar a la de esta investigación. No obstante, se encontraron otros enfoques basados en UDF de mediano acoplamiento en SGBDR como Oracle, o directamente técnicas y procesos que pueden ser útiles para nuestra investigación.

Para empezar, el artículo Experiencias en la gestión de imágenes por contenido en bases de datos objeto-relacionales, se enfoca en la gestión de imágenes en bases de datos objeto-relacionales (BDOR) y compara dos enfoques para el almacenamiento y recuperación de imágenes médicas y digitales. El estudio se basa en Oracle 10g, un Sistema de Gestión de Bases de Datos Objeto- Relacionales (SGBDOR), que examina las ventajas y desventajas de dos tipos de objetos definidos por el usuario (UDT) para el manejo de imágenes: ORDImage y SI_StillImage.

El artículo destaca la importancia de la gestión de datos multimedia, incluyendo imágenes, en las organizaciones y cómo las BDOR pueden ofrecer capacidades avanzadas para el almacenamiento y recuperación de información basada en contenido. Se discuten las características de los objetos ORDImage y SI_StillImage, que permiten almacenar imágenes junto con sus metadatos.

Además, se menciona la importancia de los metadatos en la recuperación por contenido, dado que permiten buscar imágenes por sus características físicas y semánticas. El artículo concluye que Oracle 10g y sus extensiones InterMedia brindan una infraestructura sólida para la gestión de imágenes en BDOR [9].

Uno de los proyectos cercanos a los objetivos de esta investigación es el trabajo de Sergio Sánchez Santiesteban, titulado "Recuperación de imágenes por contenido usando

descriptores generados por Redes Neuronales Convolucionales." Este proyecto propone un enfoque basado en descriptores generados por CNN para la recuperación de imágenes.

Aunque no incluye un componente de lenguaje DML, su metodología y resultados proporcionan una base sólida para la comparación [21]. La metodología propuesta contiene información relevante que puede ser usada e implementada en el sistema que se pretende desarrollar.

Por otro parte, se encontró un estudio titulado Comparación de Técnicas para Medir la Dispersión de Datos entre Gráficas de Datos que analiza técnicas de comparación de gráficas de datos, específicamente para medir la dispersión de datos entre gráficas. Este estudio evaluó cuatro métodos diferentes, incluyendo el Error Cuadrático Medio (MSE), la Relación Señal a Ruido de Pico (PSNR), la Información de Fisher y la Cota de Cramer-Rao. Cada método arrojó resultados distintos, y el último, que mide la varianza, se seleccionó para la comparación de imágenes en esta investigación [22].

Es importante considerar la extensión de PostgreSQL con la que se comparará el SGBDR- AI que se está desarrollando. "pg_similarity," creada por Euler Taveira como un UDF, representa una de las formas más sencillas para el usuario en la comparación de imágenes en PostgreSQL [3]. Su funcionamiento se basa en la comparación de vectores previamente extraídos de una imagen. Esta extensión recibe estos vectores y se encarga de realizar las comparaciones imagen por imagen. La información técnica obtenida de la documentación de esta extensión brinda las bases para el desarrollo del SGBDR medianamente acoplado y altamente acoplado.

Continuando con extensiones, al ser PostgreSQL un sistema de gestión de bases de datos relacional (SGBDR) altamente extensible y de código abierto, ofrece la posibilidad de utilizar funciones escritas en varios lenguajes de programación a través de extensiones. En esta investigación, la extensión plpython3u juega un papel importante en la integración de redes neuronales artificiales (ANN) dentro de PostgreSQL, ya que permite la ejecución de funciones escritas en Python 3 directamente en el entorno de la base de datos. Esto facilita de manera eficiente la interacción con las ANN, optimizando la integración y procesamiento de modelos de machine learning.

2. Nacionales

Se encontró también otro estudio relacionado: "Minería Multimedia: Hacia la construcción de una metodología y herramienta analítica de datos no estructurados," desarrollado por Efraín Oviedo, Alberto Carrascal y su equipo, presenta un software dedicado a la minería de datos no estructurados, incluyendo imágenes. Utiliza la metodología "MAM" en su software "PAM," que es similar a la metodología de descriptores mencionada anteriormente. Esta metodología se enfoca en la extracción de vectores característicos para su posterior comparación y análisis de similitudes [17]. Su relación con la investigación es alta, debido a que nos muestra las bases de cómo puede trabajar un sistema de estas magnitudes, las metodologías, alcance y restricciones específicas en las cuales los SGBDR-AI a desarrollar y el motor clásico PostgreSQL trabajan mejor

3. Regional

En el marco de la presente investigación, se realizó un exhaustivo análisis para identificar antecedentes regionales sobre el tema de interés. Aunque no se encontraron estudios que abordan este tema desde una perspectiva regional.

A partir de los antecedentes, se puede inferir una metodología o conjunto de herramientas a implementar en el sistema, aprovechando las lecciones aprendidas de los proyectos previos.

B. Enunciado de supuestos teóricos de la investigación

Los supuestos teóricos de esta investigación se derivan de la revisión de la literatura y el estado del arte, estos establecen las bases conceptuales de esta investigación.

1. Conceptos clave de bases de datos

Sistema Gestor de Base de datos (DBMS /SGBD)

Un Sistema Gestor de Bases de Datos, abreviado como SGBD, es una herramienta cuya función principal es el almacenamiento, administración y mantenimiento de la durabilidad de grandes conjuntos de datos por largos periodos de tiempo. Son esenciales para la gestión eficiente de datos en aplicaciones y sistemas. Permiten el almacenamiento, recuperación y

manipulación de información de manera organizada, además de mantener la integridad de los datos por largos periodos de tiempo. Ofrecen características como integridad de datos, seguridad, capacidad de consulta y transacciones, soporte para múltiples usuarios, estructuras de datos definidas por el usuario y distintos tipos como lo son: Los SGBDR y los SGBD no relacionales o no SQL [15].

En 1975, el comité ANTI-SPARC presentó una propuesta de arquitectura de tres niveles para los SGBD. El propósito de esta propuesta era la separación de la parte física de los programas en los que se fundamentaría el SGBD. El gráfico siguiente ilustra los tres niveles de esta arquitectura de un SGBD [23].

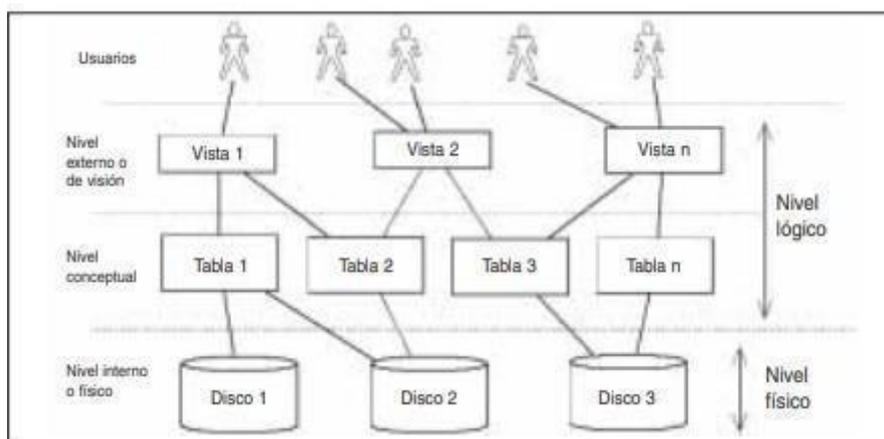


Fig. 1. Ilustra los niveles de abstracción de un SGBD

La Figura 1. ilustra los niveles de abstracción de un SGBD. Se destaca la incorporación de la capa conceptual, que actúa como un puente entre las solicitudes a nivel externo y el nivel interno del sistema.

Sistema Gestor de Base de datos Relacional (RDBMS /SGBDR)

Un sistema Gestor de Bases de Datos relacional, abreviado como SGBDR, es un tipo de SGBD que se caracteriza por tres grandes tópicos:

Estructura de los datos: Es el alto nivel de las estructuras de datos, y se refiere a la forma en la que estos están organizados, distinguiendo en el SGBDR características como los atributos, las tablas y las tuplas presentando una relación de orden entre atributos y tuplas, lo que hace organizado al SGBDR o modelo relacional de SGBD.

Operaciones en los datos: Son operaciones que devuelven o modifican los datos dependiendo de la necesidad. Estas operaciones pueden ser dadas a través de lenguaje DML.

Restricciones en los datos: Se refiere a las restricciones de rango en los datos, como lo puede ser su tipo, longitud [15].

Este modelo fue propuesto por Edgar Frank Codd a finales de la década de los 60. Como experto matemático, Codd utilizó la teoría de conjuntos para desarrollar este modelo, empleando la misma terminología. Su objetivo principal era mantener la independencia de esta estructura lógica con respecto al modo de almacenamiento.

En cuanto a su estructura, cada relación o tabla se compone de las siguientes partes:

1. **Cabecera:** Aquí se definen los atributos, siguiendo la regla de que ninguno de ellos puede tener el mismo nombre. Cada atributo tiene un rango de datos en el cual puede declararse.
2. **Cuerpo:** Este consiste en el conjunto de tuplas contenidas en la tabla, siguiendo la estructura atributo: valor.
3. **Cardinalidad:** Se refiere al número de tuplas en la relación o tabla.
4. **Grado:** Es el número de atributos o columnas de la tabla [24, 25].

La siguiente imagen representa la estructura de una relación en un SGBDR.

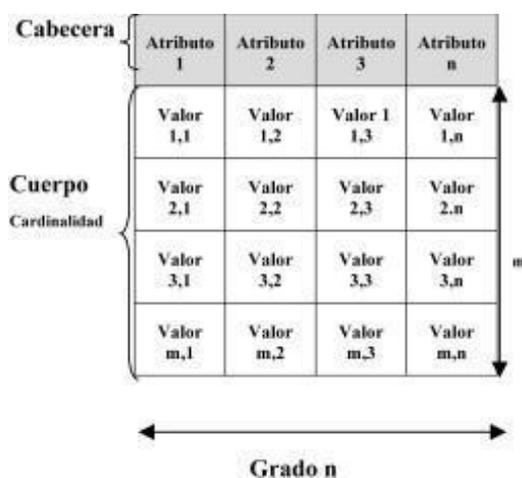


Fig. 2. La estructura de una relación en un SGBDR

En la figura 2 es posible observar las partes de la estructura relacional en un SGBDR.

Sistema Gestor de Base de datos Objeto - Relacional (ORDBMS /SGBDOR)

Un Sistema de Gestión de Bases de Datos Objeto-Relacional, conocido como SGBDOR, representa una fusión entre la tecnología relacional y la orientada a objetos, un paradigma que define objetos mediante sus características y métodos o funciones asociadas. Sus atributos distintivos incluyen:

Declaración de Atributos: Refiere a la especificación de los atributos o columnas de un objeto o relación en el contexto del SGBDOR.

Declaración de Métodos: Se refiere a las funciones definidas para el objeto o relación, esenciales para la manipulación eficiente de los datos almacenados.

Es esencial destacar que este tipo de SGBD se divide en dos partes fundamentales:

Especificación: Representa la parte pública y visible para el usuario, proporcionando información sobre la relación u objeto.

Cuerpo: Correspondiente a la parte privada, accesible sólo para el desarrollador. Aquí se declara el código y la funcionalidad de los métodos, lo que constituye la esencia del componente técnico del sistema [26].

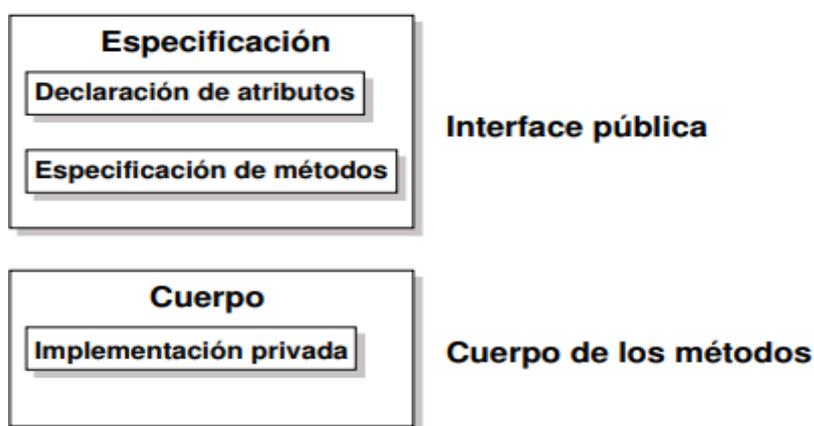


Fig. 3. La manera gráfica la estructura de un SGBDOR

La Figura 3. Proporciona una representación visual que permite observar de manera gráfica la estructura de un SGBDOR. En esta representación, se destaca claramente la distinción entre lo que se encuentra en la especificación y en el cuerpo del SGBDOR. La especificación

abarca la parte pública y visible para el usuario, ofreciendo detalles sobre la relación u objeto en cuestión. Por otro lado, el cuerpo del SGBDOR corresponde a la sección privada, accesible únicamente para el desarrollador, donde se declara el código y la funcionalidad de los métodos, constituyendo la esencia técnica del sistema.

PostgreSQL.

Es un SGBDR desarrollado por la Universidad de California en Berkeley por el departamento de ciencias. Fue pionero en los conceptos estándar de las bases de datos desarrolladas actualmente. Permite un desarrollo de consultas complejas y de definición de distintos tipos de datos, que contiene soporte para distintas funcionalidades como transacciones, triggers, llaves foráneas, control de versiones, entre otros. Es un sistema robusto que ha tenido una larga trayectoria desde su creación [28].

Data Definition Language commands (DDL) y Data Modification Language commands (DML).

Son los tipos de lenguaje de comandos que existen actualmente para la definición y modificación de los datos. Cada uno tiene un campo funcional en cuál debe ser usado.

DDL: Es el lenguaje de comandos para la definición de estructuras de datos, al cual solo el administrador de la base de datos tiene acceso.

DML: Es el lenguaje de comandos para la modificación y consulta de datos. Está relacionado con el usuario y diseñado para el uso por parte de este [15].

UDF

En bases de datos, son funciones declaradas por el usuario para extender la funcionalidad del SGBD, las cuales pueden ser declaradas en lenguaje DML (Especialmente SQL), que ejecutan operaciones por columnas en las tablas de las bases de datos. Estas funciones deben seguir un proceso de declaración dependiendo del lenguaje DML y el SGBD utilizado [1].

2. Conceptos clave de programación y desarrollo de software

Software/aplicación

El software se refiere al conjunto de componentes lógicos que conforman el

equipamiento lógico de un sistema informático. Estos componentes permiten la realización de tareas específicas dentro de un sistema. Su influencia se centra en la ejecución de funciones y procesos a nivel lógico, ya que está construido mediante lenguajes de programación.

En contraposición al hardware, que constituye la parte física de un sistema informático, el software representa la parte intangible encargada de la ejecución de tareas utilizando el hardware de manera lógica y funcional [29].

Software libre

El software de código abierto es aquel que permite a los usuarios utilizarlo, estudiarlo y modificarlo según sus necesidades, para posteriormente distribuirlo a toda la comunidad. Algunas de sus características principales son:

- Su código es accesible y transparente.
- Puede ser modificado según los requerimientos del usuario.
- Ofrece libertad para estudiar y adaptar el código.
- Proporciona libertad para distribuir copias del software [30]

Lenguajes de programación

Un lenguaje de programación se define como un conjunto de reglas que un usuario escribe para que sean ejecutadas por una máquina. Estas instrucciones son traducidas por la máquina y ejecutadas para realizar una acción específica. Para ser considerado como un lenguaje de programación, debe cumplir con las siguientes características:

No requiere conocimiento en lenguaje de máquina: Un lenguaje de programación está diseñado para que los usuarios puedan escribir instrucciones sin la necesidad de entender el lenguaje de máquina o código binario. Permite a los programadores trabajar en un nivel más abstracto y comprensible.

Independencia de la máquina: Un lenguaje de programación debe ser independiente de la arquitectura o características específicas de la máquina en la que se ejecuta. Esto significa que el mismo código debe ser capaz de ejecutarse en diferentes tipos de

hardware o sistemas.

Traducción a lenguaje de máquina: Las instrucciones escritas por el usuario en el lenguaje de programación deben ser traducidas a lenguaje de máquina o código ejecutable por la computadora para realizar las acciones deseadas.

Notación orientada al usuario: Los lenguajes de programación están diseñados para tener una notación que sea más comprensible y cercana al usuario que al lenguaje de máquina. Utilizan símbolos, palabras clave y estructuras que facilitan la comprensión y escritura de los programas [31].

El lenguaje de programación que fueron usados para el desarrollo del SGBDR-AI son los siguientes:

Python: Python es un lenguaje de programación de propósito general y alto nivel que utiliza la indentación para delimitar bloques de código, una característica no muy común en otros lenguajes. Permite la aplicación de distintos paradigmas de programación y es interpretado, lo que significa que el código se va compilando y ejecutando parte por parte en tiempo real, en lugar de ser compilado por completo antes de su ejecución [32].

Framework

Un framework es un conjunto de archivos o herramientas que agilizan el desarrollo de aplicaciones al incorporar funcionalidades previamente implementadas y probadas en diversos lenguajes de programación. Su objetivo principal radica en facilitar el desarrollo de aplicaciones, permitiendo que los desarrolladores se enfoquen en la problemática en lugar de las funcionalidades comunes. Esto conlleva a un mejor mantenimiento del código de la aplicación.

Los frameworks presentan distintas características, entre las que se incluyen:

Uso de patrones de diseño: Estos frameworks emplean paradigmas predefinidos para crear aplicaciones más robustas y funcionales, siguiendo pautas de diseño probadas y eficaces. **Estructura predefinida de la aplicación:** Se refiere a la organización de archivos dentro de una aplicación. Los frameworks ofrecen una estructura organizativa prediseñada, lo que ayuda a los programadores a encontrar y organizar los archivos de manera más eficiente, permitiendo un desarrollo más ordenado y comprensible.

Código altamente testado: El código integrado en un framework se somete a rigurosas pruebas para garantizar su funcionalidad y fiabilidad, lo que proporciona a los desarrolladores un ambiente confiable para construir sus aplicaciones.

Comunidad de usuarios: Los frameworks suelen contar con una comunidad activa de usuarios que contribuyen al desarrollo y mejora del framework. Esta comunidad a menudo crea nuevas extensiones, herramientas y documentación útil para el beneficio de otros desarrolladores, facilitando la expansión y el uso del framework [34].

Conjuntos de datos

Los conjuntos de datos representan recopilaciones de información que pueden adoptar diversos formatos y dividirse en distintas categorías y subcategorías, cada una con características específicas y en contextos variados.

Según el tipo de datos: En función del tipo de datos, se pueden clasificar en las siguientes categorías:

- Conjunto de datos numéricos. Se refiere a los datos usados para análisis cuantitativos.
- Conjunto de datos de texto. Se refiere a datos como mensajes de texto libre o documentos.
- Conjunto de datos multimedia. Se refiere a datos como imágenes, audio o video.
- Conjunto de datos de series temporales. Se refiere a datos recopilados a lo largo del tiempo.
- Conjunto de datos espaciales. Se refiere a datos con información geográfica.

Segundo su estructura: A partir de su estructura u organización pueden dividirse en las siguientes categorías:

- Conjunto de datos estructurados. Se refiere a datos organizados en una estructura específica como en un SGBDR.
- Conjunto de datos no estructurados. Se refiere a datos que no tienen una estructura definida.
- Conjunto de datos híbridos. Se refiere a un conjunto de datos compuesto de datos estructurados y no estructurados.

En modelos de aprendizaje automático: Dentro del ámbito de los modelos de aprendizaje automático, los conjuntos de datos se clasifican de la siguiente manera:

- Conjunto de datos de entrenamiento. Se refiere al conjunto de datos para el

entrenamiento de un modelo de aprendizaje automático.

- Conjunto de datos de validación. Se refiere a un conjunto de datos para reducir el sobreajuste de un modelo y mejorar la precisión.
- Conjunto de datos de prueba. Se refiere a un conjunto de datos destinado para realizar pruebas a un modelo [35].

Entorno de desarrollo

Un entorno de desarrollo se define como un conjunto de herramientas, procedimientos y recursos utilizados en el proceso de creación, depuración y construcción de software a través del desarrollo de código fuente o programas. El uso de un entorno de desarrollo implica una serie de pasos, desde la redacción inicial del código hasta las pruebas, depuración y, finalmente, la compilación del programa resultante [36].

Dentro del marco de esta investigación, se contempla el empleo de Google Colab como el entorno de desarrollo preferido.

Google Colab

Representa un entorno de desarrollo en línea, proporcionando una plataforma que permite a los usuarios desarrollar y ejecutar código de manera remota. Su estructura se organiza en bloques, donde se pueden intercalar secciones de código ejecutable junto con bloques de texto descriptivo. Esto permite a los desarrolladores no solo escribir y probar código, sino también documentar y marcar pautas a través de bloques de texto explicativos [37]. Este formato de bloques facilita la organización del flujo de trabajo, permitiendo una interacción más clara y estructurada durante el desarrollo de programas o modelos de Deep Learning en el contexto de esta investigación.

Google drive

Es un servicio de almacenamiento de datos en línea que ofrece 15 GB de almacenamiento para cada cuenta de Gmail registrada. Permite la copia de archivos desde el ordenador personal hacia este servicio en línea para asegurar una mayor persistencia y acceso a los datos desde cualquier lugar del mundo y en cualquier dispositivo. Algunas de sus características son:

- Edición en línea de los archivos.
- Capacidad para compartir datos con otros usuarios en tiempo real [38].

3. Conceptos clave de inteligencia artificial (IA/AI)

Inteligencia artificial (IA)

La Inteligencia Artificial (IA) se posiciona como una disciplina dentro de las ciencias computacionales que busca replicar la inteligencia humana para resolver problemas mediante el uso de símbolos y métodos que no siguen un enfoque estrictamente algorítmico o una secuencia de pasos específica [39].

En esencia, la IA se enfoca en desarrollar sistemas que puedan realizar tareas que, normalmente, requerirían inteligencia humana para llevarlas a cabo. Esto implica la capacidad de comprender, razonar, aprender de la experiencia y adaptarse a diferentes situaciones. Los métodos utilizados en IA incluyen enfoques como, el procesamiento del lenguaje natural, la visión por computadora o procesamiento digital de imágenes, entre otros, con el objetivo de crear sistemas capaces de tomar decisiones o realizar tareas complejas de manera autónoma y eficiente.

Redes Neuronales Artificiales (ANN/RNA)

Las Redes Neuronales Artificiales, conocidas como RNA o ANN, son un tipo de estructura diseñada para la síntesis de algoritmos y la solución de problemas lineales y no lineales. Son estructuras diseñadas para encontrar generalidades en conjuntos de datos experimentales, y de esta forma obtener una predicción a partir del afinamiento de parámetros. Una de sus características principales es que son robustas y persisten ante las fallas de procesamiento, lo cual es una ventaja para el análisis de sistemas complejos.

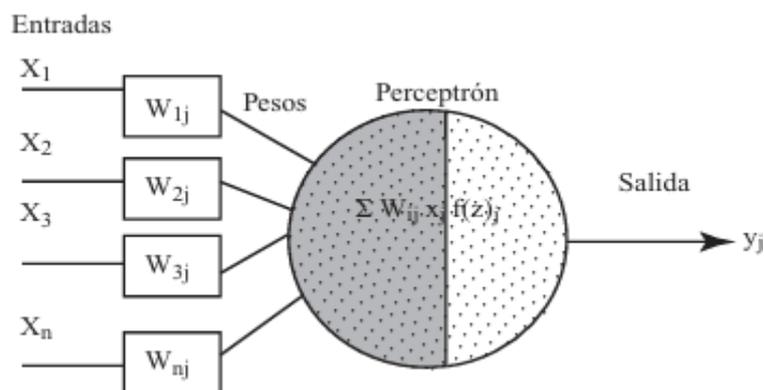


Fig. 4. Representa un perceptrón

La Figura 4. representa un perceptrón, que es la forma más simple de una ANN (Red Neuronal Artificial). Esta consta de entradas y pesos, que pueden ser positivos (excitación) o

negativos (inhibición). Estos, en combinación con las funciones, determinan el funcionamiento de la red neuronal. El proceso consiste en la suma de la multiplicación de las entradas por los pesos de la red. El valor resultante se evalúa en una función denominada función de activación para activar la neurona [39].

Redes Neuronales Convolucionales (CNN)

La CNN es un tipo de Red Neuronal Artificial (RNA/ANN) que procesa sus capas imitando al córtex visual del cerebro humano para identificar distintas características en las entradas. Para ello, la CNN contiene varias capas ocultas especializadas y con una jerarquía: esto significa que las primeras capas detectan propiedades o formas básicas y se van especializando hasta llegar a capas más profundas capaces de reconocer formas complejas como un rostro o una silueta.

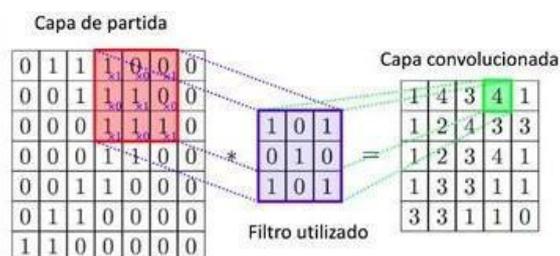


Fig. 5. El funcionamiento de una CNN

En la Figura 5.0, se puede apreciar el funcionamiento de una CNN. El proceso implica el uso de un filtro también llamado kernel que se aplica a los píxeles de la imagen, desplazándose horizontal y verticalmente, lo que resulta en una nueva matriz. Es importante destacar que se pueden aplicar múltiples filtros a una imagen para obtener un feature mapping. Este procedimiento se conoce como convolución y al final de este proceso se aplica una función de activación, similar a lo que se hace en las ANN [40].

Procesamiento - tratamiento de imágenes digital (PDI)

El procesamiento de imágenes comprende un conjunto de técnicas empleadas para la extracción de características o descriptores en una imagen, siendo un proceso que persigue diversos objetivos, tales como:

Mejoramiento de Información: Implica elevar la calidad de una imagen con el propósito de facilitar la obtención de información más precisa y clara.

Procesamiento de Datos Contenidos: Se refiere a la capacidad de la percepción autónoma de información dentro de una imagen.

A lo largo de los años, el uso del procesamiento de imágenes ha sido diverso y significativo. Un ejemplo de su aplicación se encuentra en el ámbito de la fotografía, donde contribuye al mejoramiento de la calidad de las imágenes. Además, despliega un papel crucial en la detección de enfermedades en el campo de la medicina, permitiendo a los profesionales mejorar sus diagnósticos mediante la extracción de los datos contenidos en imágenes médicas. Estos ejemplos representan sólo una fracción de los diversos usos del Procesamiento de Imágenes Digital (PDI), que actualmente ha adquirido relevancia en numerosas áreas operativas que hacen uso de imágenes en sus procesos [41].

Machine Learning

El Aprendizaje Automático es una rama crucial de la Inteligencia Artificial que emplea datos y algoritmos para emular el aprendizaje humano, perfeccionando gradualmente su precisión. Se basa en métodos estadísticos y operaciones matemáticas para entrenar algoritmos en tareas de predicción o clasificación. Por lo general, se construyen utilizando marcos especializados que agilizan su implementación. [42]

Deep Learning

Es una ANN que se concibe como un modelo que busca emular la capacidad del cerebro humano, permitiendo el aprendizaje a través de grandes volúmenes de datos y con múltiples capas en la red. Estos modelos, al contar con varias capas, tienen la capacidad de mejorar la precisión en predicciones en comparación con redes más simples o menos profundas.

La utilidad de estos modelos radica en su capacidad para mejorar la precisión en tareas analíticas y físicas, realizando acciones de manera automatizada sin requerir intervención humana, dependiendo en gran medida de su aplicación y del entrenamiento recibido. Durante el proceso de entrenamiento, se busca afinar los parámetros, particularmente los pesos de la red, haciendo uso de diversos algoritmos matemáticos y técnicas de optimización [43]. Este entrenamiento es fundamental para lograr que la red neuronal aprenda y se ajuste de manera óptima a los datos de entrada, permitiendo mejorar su capacidad predictiva y su rendimiento en diversas tareas.

Frameworks de Deep Learning

Un framework de Deep Learning es un conjunto de herramientas que facilitan el desarrollo y programación de modelos de ANN. Estos frameworks ofrecen un conjunto de funciones y operaciones matemáticas que simplifican el proceso de desarrollo.

En el contexto de esta investigación, se mencionan algunos frameworks que pueden ser utilizados:

TensorFlow: Es una herramienta de Deep Learning para Python diseñada para desarrollar Redes Neuronales Artificiales, empleando diversos cálculos matemáticos. TensorFlow es capaz de entrenar modelos, descubrir correlaciones y patrones en los datos suministrados [44].

Pytorch: PyTorch es una herramienta especializada en Deep Learning para Python que se centra en el cálculo tensorial. Se destaca por su naturaleza pionera y su facilidad para ser ampliada por la comunidad de desarrolladores. PyTorch es capaz de ser útil en la creación de modelos que pueden identificar patrones en los datos [45].

Flask:lask es un microframework para el desarrollo de aplicaciones web en Python, diseñado para ser ligero y modular, lo que permite a los desarrolladores construir aplicaciones de manera rápida y sencilla. A diferencia de otros frameworks más pesados, Flask proporciona las herramientas básicas necesarias para el desarrollo web, permitiendo a los programadores seleccionar y añadir solo las extensiones que requieren para sus proyectos.

Sus características clave incluyen un enrutador flexible, soporte para plantillas a través de Jinja2, y un sistema de manejo de solicitudes y respuestas que facilita la creación de aplicaciones RESTful. Flask promueve la separación de responsabilidades y la escalabilidad, lo que lo convierte en una opción ideal tanto para proyectos pequeños como para aplicaciones de mayor envergadura.

Además, la comunidad activa que respalda Flask proporciona una amplia gama de documentación y recursos, lo que facilita el aprendizaje y la resolución de problemas. Su simplicidad y flexibilidad hacen de Flask una herramienta valiosa en el ámbito del desarrollo web moderno

4. Conceptos clave de evaluación y comparación

Niveles de acoplamiento (Fuerte, mediano y bajo)

El **acoplamiento** se refiere a la relación y nivel de dependencia entre los módulos de un mismo sistema, siendo el bajo acoplamiento el estado más deseado en el desarrollo de software. Se identifican varios niveles de acoplamiento, los cuales se pueden categorizar de la siguiente manera:

1. **Fuerte acoplamiento:** Se caracteriza por una alta interdependencia entre los módulos, donde el funcionamiento de uno depende significativamente del otro. En este caso, una modificación en un módulo afecta el comportamiento del otro, generando una mayor complejidad en el mantenimiento y evolución del sistema.
2. **Mediano acoplamiento:** Implica una dependencia menos rígida, donde los módulos pueden compartir datos o estructuras, pero sin afectar directamente sus comportamientos centrales. Esta forma de acoplamiento es más flexible que el fuerte, pero aún puede generar problemas si los datos compartidos cambian de forma imprevista.
3. **Bajo o débil acoplamiento:** Se refiere a una relación mínima entre los módulos, donde una modificación en uno de ellos no repercute en el funcionamiento del otro. Este tipo de acoplamiento permite una mayor independencia entre los módulos y facilita el mantenimiento, ya que los cambios pueden ser realizados de manera aislada.

Estos niveles de acoplamiento influyen en la escalabilidad, mantenimiento y flexibilidad del sistema, siendo el bajo acoplamiento lo más recomendable para un diseño eficiente [58].

Algoritmos/métodos de comparación

Son los métodos mediante los que se obtiene la diferencia entre las características de las imágenes extraídas. Algunos de los métodos de comparación que podemos resaltar son:

Distancia euclidiana: Un enfoque de recuperación de imágenes basado en la transformación de distancia euclidiana. Se centra en la comparación de imágenes utilizando la distancia euclidiana y cómo esta métrica puede aplicarse en tareas de recuperación de imágenes [46].

Coefficiente de correlación: Se enfoca en la recuperación de imágenes utilizando información de color y forma. Explora cómo se pueden utilizar coeficientes de correlación y otras métricas para medir la similitud entre imágenes en función de sus características de color y forma [47]. **Similaridad del coseno:** Se centra en un enfoque de cálculo de ángulos en los vectores característicos de documentos [48]. Es una métrica que se utiliza en documentos de texto, pero también puede ser útil en imágenes.

Los supuestos teóricos identificados y planteados son de naturaleza general para la investigación. Los métodos de comparación, como los mencionados en los antecedentes, junto

con otros algoritmos para la extracción de características, serán incluidos más adelante en la investigación cuando se seleccionen las herramientas específicas que se utilizarán en el SGBDR- AI.

C. Variables del estudio

Las variables del estudio que se identifican a partir de los supuestos teóricos y la literatura son las siguientes:

Variables independientes

- Cantidad de datos

Variables dependientes

- SGBDR-AI
- Exactitud
- Eficiencia

1. Definición nominal de variables

Cantidad de Datos: Número entero que representa la cantidad de datos utilizados en cada operación o consulta realizada en el SGBDR-AI. Este valor variará durante las pruebas experimentales.

SGBDR-AI: Es el SGBD con ANN integradas a nivel de alto acoplamiento. Su función principal es la realización de consultas a través de lenguaje DML usando procesamiento de imágenes a través de los modelos ANN previamente mencionados

Exactitud: Proporción que indica el número de operaciones exitosas respecto al total de intentos. En el contexto del proyecto, se refiere a la cantidad de consultas exitosas realizadas por el SGBDR-AI al comparar los descriptores de imágenes.

Eficiencia: Medida del tiempo de respuesta en las consultas del SGBDR-AI. La eficiencia será mayor cuanto menor sea el tiempo de respuesta y menor cuando el tiempo de respuesta sea mayor.

2. Definición operativa de variables

Cantidad de Datos: Es la cantidad de datos expresada en números enteros. Su medición se realizará haciendo inventario de los datos contenidos en cada conjunto de datos.

SGBDR-AI: Para realizar su evaluación se tendrá en cuenta la exactitud y eficiencia de las consultas realizadas con grandes cantidades de datos.

Exactitud: Para medir esta variable, se calculó dividiendo el número total de imágenes clasificadas satisfactoriamente entre la cantidad total de datos. Podemos distinguir la siguiente fórmula:

$$\text{Exactitud} = (\#Intentos Exitosos / \#Total de datos) * 100$$

Fig. 6. Ecuación exactitud

#Total de datos: Se refiere al total de datos usados en la consulta.

El resultado obtenido se multiplica por 100 para expresar la exactitud en porcentaje

Eficiencia: Esta variable se refiere al tiempo de respuesta de las operaciones o consultas realizadas en el SGBDR-AI. La medición se realizó desde el momento en que se envía la consulta hasta que se recibe la respuesta, además esta variable será expresada en milisegundos.

3. Formulación de la hipótesis

4. Hipótesis de investigación

El sistema gestor de base de datos relacional tiene un tiempo de respuesta en la ejecución de operaciones DML significativamente menores, utilizando operaciones ANN con imágenes altamente acopladas que en operaciones de ANN medianamente acopladas tanto en el mismo motor como en el motor PostgreSQL.

5. Hipótesis nula

El sistema gestor de base de datos relacional tiene un tiempo de respuesta en la ejecución de operaciones DML significativamente mayores, utilizando operaciones ANN con imágenes altamente acopladas que en operaciones de ANN medianamente acopladas tanto en el mismo motor como en el motor PostgreSQL.

6. Hipótesis alternativa

El sistema de clasificación de imágenes con ANN tiene un tiempo de ejecución de operaciones DML similar al sistema medianamente acoplado y PostgreSQL.

III. Metodología

A. Paradigma

La investigación es de tipo positivista, un paradigma que es racional, objetivo y se basa en comprobar hechos y particularidades propias del conocimiento científico [49], característico de la esencia disciplinar ingenieril en la que se desarrolla la presente investigación, donde se comparará el rendimiento de dos sistemas de gestión de datos con imágenes. Los datos serán analizados utilizando métodos estadísticos para determinar si existe una diferencia significativa en el rendimiento de los dos sistemas.

B. Enfoque

La presente investigación está basada en el enfoque cuantitativo, el cual permite recopilar datos numéricos que se pueden analizar utilizando estadísticas [50], que implica comparar el tiempo de respuesta entre diferentes sistemas de clasificación de imágenes utilizando un SGBDR con ANN y SGBDR tradicional, como PostgreSQL con la extensión escogida, está recopilando datos cuantitativos (tiempo de respuesta) y utilizando un enfoque cuantitativo para analizar y comparar los resultados a través de mediciones numéricas y técnicas estadísticas.

C. Método

El método investigativo científico es el método más riguroso para probar una hipótesis. En este caso, el SGBDR-AI tendrá un tiempo de respuesta en la ejecución de operaciones DML significativamente menores, utilizando ANN altamente acopladas que en operaciones medianamente acopladas tanto en el mismo motor como en el motor PostgreSQL. Para evaluar esta hipótesis, se recopilarán datos, ejecutarán experimentos y analizarán los resultados. A través de este proceso, se buscará proporcionar evidencia que respalde o refute la hipótesis.

Este método científico tiene el objetivo de proporcionar una evaluación objetiva que arroje información sobre la eficacia del SGBDR-AI en el contexto de la comparación y consulta de imágenes, de esta manera contribuyendo así al conocimiento existente en este campo y ofreciendo orientación para futuros desarrollos.

D. Tipo de investigación

Es una investigación comparativa en el desarrollo de un SGBDR simplificado para facilitar el acoplamiento de modelos fuertemente acoplados implica la comparación y análisis de diferentes enfoques o soluciones disponibles para abordar este desafío

E. diseño de la investigación

Para la investigación se utilizó un diseño experimental es una técnica estadística que permite identificar y cuantificar las causas de un efecto dentro de un estudio experimental. En un diseño experimental se manipulan deliberadamente una o más variables, vinculadas a las causas, para medir el efecto que tienen en otra variable de interés [51]. La investigación se centra en el estudio de la eficiencia que alcanza un SGBD con ANN altamente acoplada en comparación con acoplamientos medianos y débiles, el diseño de investigación se presenta a continuación

G: Determina la cantidad de datos (Conjunto de datos de imágenes representativas) X: SGBDR-AI

O: Pruebas de desempeño mediante consultas

DML P: SGBD PostgreSQL (Extensión pertinente)

$G \rightarrow X \rightarrow O \rightarrow P$

En la investigación se busca aplicar un conjunto de datos **G** a un SGBDR-AI **X**, donde se realizarán pruebas mediante consultas DML **O** para para la medición de la eficiencia en la ejecución de operaciones DML con imágenes mediante la comparación de su ejecución, Para continuar la comparación con el SGBDR (PostgreSQL) **P**. Estos datos recopilados servirán para validar las hipótesis planteadas en la investigación.

En concreto, estamos comparando el rendimiento del SGBDR-AI con operaciones ANN

altamente acopladas con el rendimiento del SGBDR-AI con operaciones ANN medianamente acopladas. Si el rendimiento del SGBDR-AI con operaciones ANN altamente acopladas es significativamente mejor que el rendimiento del SGBDR-AI con operaciones ANN medianamente acopladas, entonces podremos concluir que el tipo de acoplamiento afecta significativamente el rendimiento de las operaciones DML con imágenes.

Por lo que la manipulación del tipo de acoplamiento en este estudio es necesaria para verificar si el rendimiento de las operaciones DML con imágenes. Si los resultados del estudio muestran que el rendimiento mejora significativamente con un acoplamiento alto, entonces esto proporcionará evidencia de que el tipo de acoplamiento es un factor importante que debe tenerse en cuenta al diseñar sistemas para gestionar datos no estructurados.

F. Población

Para la presente investigación se utilizan conjuntos de datos de entrenamiento como: Zoo, Mnist o Fashion Mnist, además de imágenes artificiales en las consultas de la base de datos.

G. Muestra

La muestra utilizada consistió en un conjunto de datos de imágenes representativas, adecuadas para modelos ANN de clasificación. Este conjunto de datos, creado por los investigadores, se basa en la estructura de otros conjuntos de entrenamiento disponibles en la red.

H. Técnicas de recolección de la información

Se utilizaron datos secundarios para comprender el estado de la investigación en el campo de los SGBDR fuertemente acoplados con ANN. Además, se emplearon los experimentos como técnica de recolección de la información, técnica diseñada específicamente para examinar relaciones de causa y efecto [52]. Estos datos se recopilaron de revistas científicas y técnicas, libros, conferencias y talleres, informes técnicos y sitios web de organizaciones profesionales. El análisis de los datos secundarios permite identificar tendencias y mejores prácticas en el diseño y desarrollo de SGBDR fuertemente acoplados con ANN.

Los experimentos se realizaron seleccionando un conjunto de datos de imágenes

representativas de las imágenes que se utilizarán en las operaciones DML, entrenando una red neuronal artificial para realizar una tarea de DML utilizando el SGBDR acoplado con la red neuronal artificial para realizar la tarea de DML en el conjunto de datos seleccionado y midiendo los siguientes indicadores de rendimiento: tiempo de ejecución, recursos utilizados y precisión de la clasificación o detección.

1. Validez de la técnica

La validez de las técnicas de recolección de información utilizadas en esta investigación es alta. Los experimentos permiten al investigador controlar las variables que afectan al rendimiento del SGBDR. acoplado con las Redes Neuronales Artificiales, como el conjunto de datos de imágenes, la arquitectura de la red neuronal artificial y los parámetros de configuración, la selección de variables adecuadas para controlar es fundamental para garantizar la validez de los experimentos.

En cuanto a los datos secundarios está influenciada por la calidad de las fuentes de las que proceden. Las revistas científicas y técnicas son las fuentes más confiables, ya que suelen someterse a un proceso de revisión por pares. La calidad de las fuentes secundarias también depende de otros factores, como el rigor del proceso de recopilación de datos, la transparencia de la presentación de los datos y la coherencia de los datos con otros estudios.

2. Confiabilidad de la técnica

Los experimentos se realizaron bajo diversas condiciones, utilizando el mismo conjunto de datos en diferentes cantidades y en un mismo equipo con un hardware específico. Esto asegura que los resultados sean consistentes y reproducibles. Además, los resultados fueron revisados por el experto y magíster en inteligencia artificial Héctor Mora, lo que añade una capa adicional de confiabilidad a los experimentos.

Los datos secundarios se recopilarán de fuentes confiables, como artículos científicos publicados en revistas revisadas. Estas fuentes proporcionan datos que han sido cuidadosamente revisados por expertos, lo que aumenta su confiabilidad.

1. Instrumento de recolección de datos

Los experimentos son una técnica de recolección de información cuantitativa. En el caso de la investigación los experimentos se realizaron utilizando un conjunto de datos de imágenes representativo en el mundo real. Los experimentos medirán los siguientes indicadores de

rendimiento: tiempo de ejecución, recursos utilizados y precisión de la clasificación o detección.

Además de los experimentos, se recopilaron datos secundarios de una variedad de fuentes, como artículos científicos, informes técnicos y sitios web. Los datos secundarios se utilizaron para proporcionar contexto a los resultados de los experimentos y para identificar áreas de investigación sobre el rendimiento de redes neuronales artificiales para la clasificación y detección de imágenes

En este proceso, contamos con la asesoría del profesor Héctor, quien brindó orientación durante el desarrollo del proyecto.

IV. RESULTADOS DE LA INVESTIGACIÓN

A. Desarrollo del SGBDR simplificado (ToyDBMS)

Para la investigación, fue necesario desarrollar un Sistema de Gestión de Bases de Datos Relacional (SGBDR) simplificado (bautizado como ToyDBMS), dado que no existe en el mercado uno que ofrezca de manera nativa la capacidad de realizar consultas inteligentes basadas en imágenes. La creación de un SGBDR completo implica un alto grado de complejidad, tanto en su motor (kernel) como en la creación de su interfaz. Por esta razón, se optó por reducir las funciones del SGBDR a las esenciales para este tipo de software especializado, reconociendo que el desarrollo de un sistema completo sería un reto considerable que requeriría un equipo de desarrollo más amplio.

Para cumplir con este objetivo, se optó por el uso de una metodología ágil, específicamente SCRUM, ya que es la que mejor se adapta al desarrollo del software deseado, debido a su flexibilidad y enfoque en la comunicación constante entre equipos de desarrollo pequeños.

El desarrollo se ejecutó en cinco fases principales: inicio, planificación, implementación, retrospectiva y cierre del proyecto. Exceptuando el inicio y el cierre, las fases de planificación, implementación y retrospectiva se repiten a lo largo de los sprints definidos al inicio del proyecto, siendo estos eventos clave dentro de la metodología SCRUM.

1. Inicio del proyecto

En la fase inicial del proyecto, fue fundamental recolectar los requisitos que un SGBD necesita. Para ello, se realizó un estudio detallado de las arquitecturas comunes en este tipo de sistemas, llegando a la conclusión de que la arquitectura ANSI-SPARC, con sus tres niveles de abstracción, era la más adecuada. Su principal propósito es garantizar una separación clara entre la lógica del sistema y el almacenamiento físico de los datos [59].

Otro aspecto clave considerado en esta fase fue la elección del lenguaje DML y DDL a utilizar. Se adoptó el estándar ISO/IEC 9075:2023 para el uso del lenguaje SQL en ToyDBMS, debido a la

simplificación que ofrece al unificar ambos tipos de lenguajes [60]. Sin embargo, fue necesario desarrollar una extensión específica de SQL para gestionar la definición y consulta de datos no estructurados, adaptando así el sistema a las particularidades del proyecto.

Como resultado de la investigación realizada, se optó por dividir el sistema en dos partes. La primera corresponde al motor de base de datos, que abarca las capas conceptual y física de la arquitectura ANSI-SPARC. La segunda parte corresponde a la API y las interfaces del sistema, con las cuales interactuaría el usuario, asociándose así a la capa externa del modelo ANSI-SPARC.

2. El motor de base de datos (ToySQL)

Teniendo en cuenta la división derivada de la arquitectura adoptada por el sistema, se inició la investigación de una arquitectura específica para el motor de base de datos. Este motor es el software encargado de operar en los niveles conceptual y físico de la arquitectura ANSI-SPARC. Realiza operaciones lógicas utilizando entidades y una estructura de datos que organiza la información a nivel lógico, además de gestionar la manipulación de los archivos en el disco duro que contienen los datos

Para el desarrollo del motor de **ToyDBMS**, que se denominó posteriormente **ToySQL**, se fijó como función principal llevar a cabo las operaciones lógicas del sistema. Esto se logró a través del procesamiento de consultas, donde este sistema tomaría como entrada las consultas en lenguaje SQL del usuario y las transformaría en instancias de clases que corresponden al nivel conceptual de la arquitectura ANSI-SPARC. Este proceso sería manejado por el parser, que sería capaz de convertir las consultas SQL en objetos conceptuales dentro de **ToyDBMS**.

Una vez que se obtienen estos objetos, el sistema pondría en marcha un análisis de los mismos y se determinaría el orden de ejecución de sus métodos a nivel lógico, lo cual corresponde al planeador del sistema y la estructura de datos seleccionada. Finalmente, el sistema realizaría los cambios en los archivos de base de datos, operando a nivel físico, lo que corresponde al ejecutor del sistema, también conocido como administrador de almacenamiento.

En la **Figura 7** se presenta la arquitectura del procesamiento de consultas en un motor de base de datos, que sirvió en este punto de inspiración para la arquitectura de **ToySQL**. Sin embargo, se omitieron bloques y funcionalidades de sistemas gestores más avanzados, dando lugar a una arquitectura significativamente más simple, como se observa en la **Figura 8**.

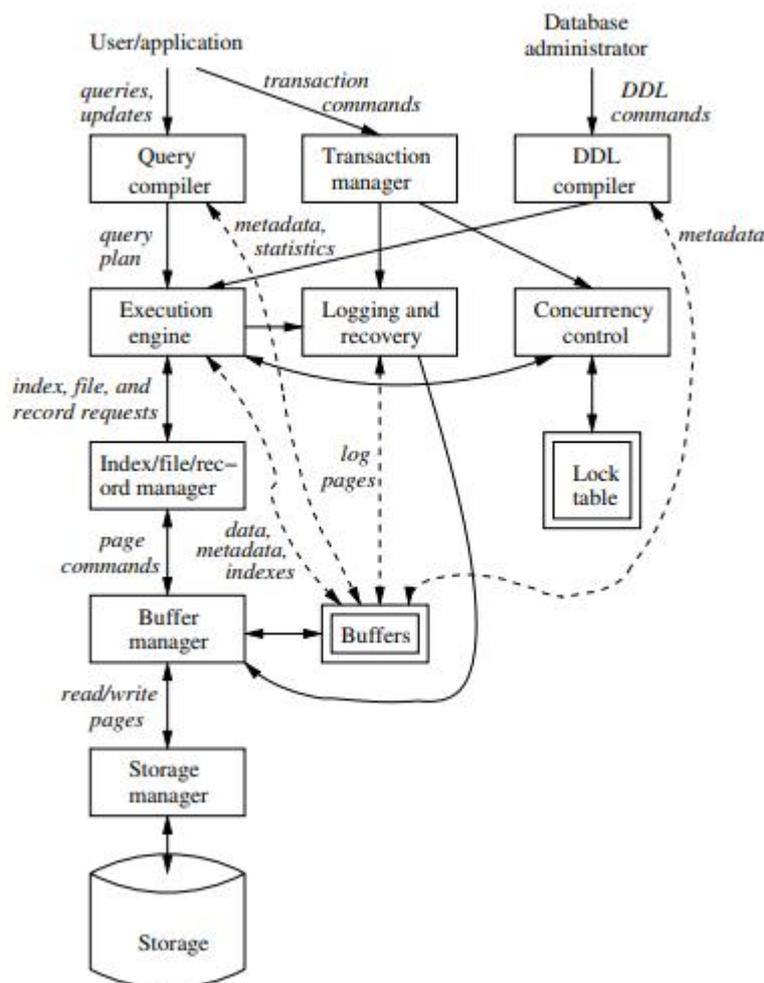


Fig. 7. Arquitectura del procesamiento de consultas

. Imagen obtenida de “Database Management Systems The complete Book”

La Figura 7 muestra un Sistema de Gestión de Bases de Datos (SGBD) de producción más complejo, que ofrece al usuario un conjunto amplio de funciones para interactuar y automatizar las operaciones sobre los datos. Este sistema se puede clasificar en tres tipos de "entradas" o interacciones principales:

Consultas de Usuarios

Los usuarios son responsables de realizar consultas a la base de datos mediante el lenguaje SQL. Estas consultas son gestionadas por el **query compiler**, el cual se encarga de analizar la sintaxis de las consultas y generar un **plan de acción** para su ejecución. Dicho plan de acción es luego procesado por el **execution engine**. Este motor de ejecución se encarga de obtener los datos solicitados, generalmente dividiéndolos en pequeñas partes, que luego son gestionadas e indexadas por el **index/file/record manager**.

Los datos indexados pasan al **buffer manager**, que les asigna un espacio temporal en la memoria antes de ser enviados al **almacenamiento secundario** (generalmente en un disco duro). Posteriormente, el **storage manager** se encarga de escribir o almacenar definitivamente estos datos en el dispositivo de almacenamiento físico.

Transacciones

Las transacciones son operaciones automáticas del tipo DML (Data Manipulation Language) que se ejecutan sobre los datos en el sistema. Estas transacciones aseguran la persistencia de la información a través del subsistema de **logging and recovery**. Los cambios realizados por las transacciones se almacenan inicialmente en el buffer y permanecen allí hasta que el **buffer manager** les asigna un lugar dentro del almacenamiento secundario.

Para asegurar la consistencia y evitar conflictos durante la ejecución de múltiples transacciones, el sistema implementa un mecanismo de **control de concurrencia**. Este mecanismo es crucial para mantener el orden de ejecución de las transacciones y prevenir la pérdida de información en situaciones donde varios usuarios acceden o modifican los mismos datos simultáneamente.

Operaciones de Manipulación de Datos (DDL)

El **compilador DDL** (Data Definition Language) es responsable de gestionar las operaciones que modifican la estructura o los datos almacenados en la base de datos. A diferencia del **query compiler**, que solo se utiliza para procesar consultas, el compilador DDL se encarga de interpretar y ejecutar comandos que definen o alteran las tablas, índices, y otros objetos dentro de la base de datos.

Estas operaciones DDL modifican directamente la estructura de la base de datos, y en muchos casos implican cambios en los datos almacenados. Una vez procesadas, estas operaciones siguen un flujo similar al de las consultas de los usuarios: los datos se pasan al **execution engine**, donde son manipulados antes de ser gestionados por el sistema de almacenamiento.

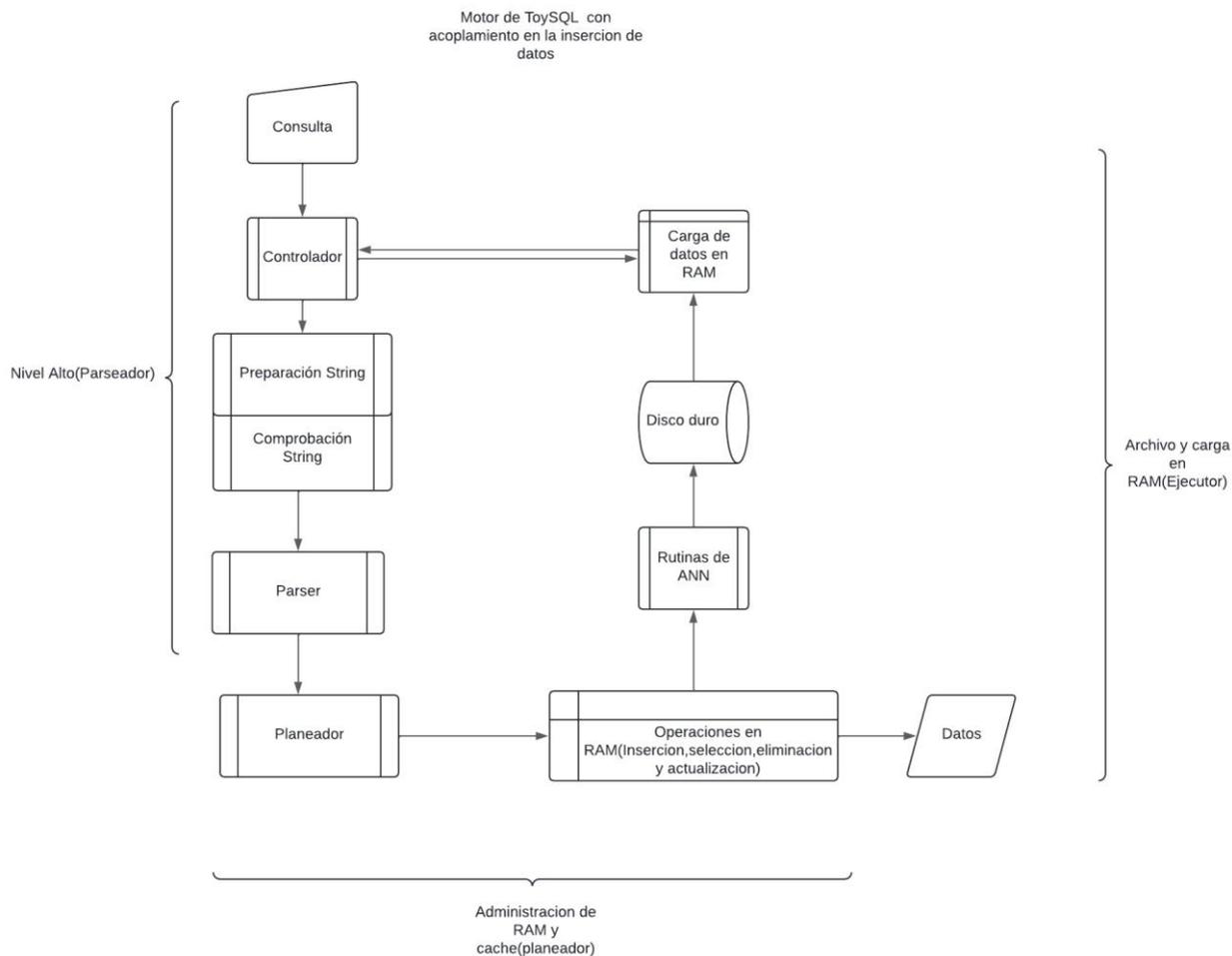


Fig. 8. Arquitectura del motor ToySQL.

La Figura 8 ilustra los tres niveles de la arquitectura de ToyDBMS, la cual sigue un enfoque similar al modelo presentado en la Figura 7, pero con varias simplificaciones para ajustarse a los objetivos de la investigación.

En este sistema, las operaciones DML y DDL, que en el modelo tradicional están separadas, se han unificado en una única entrada que gestiona tanto las acciones del usuario como las del administrador. Esta consolidación permite simplificar el flujo de operaciones y optimizar el procesamiento, manteniendo la funcionalidad esencial, pero eliminando la complejidad innecesaria.

Además, las transacciones han sido suprimidas en este diseño, ya que no son relevantes para los propósitos de la investigación. De esta forma, ToyDBMS se centra exclusivamente en la gestión de consultas y operaciones de manipulación de datos, reduciendo su complejidad y ajustándose a las necesidades específicas del estudio.

Nivel Alto o Parseador

En este nivel se encuentra la entrada del sistema: una consulta en lenguaje SQL. Esta consulta es recibida y procesada por el motor. La entrada inicial, denominada **consulta**, pasa a través del **controlador**, cuya función principal es gestionar tareas clave del motor, como abrir o cerrar la conexión con la base de datos y recibir las consultas de los usuarios.

El siguiente paso es el proceso de **preparación de string**, donde las consultas son transformadas a minúsculas para facilitar la interpretación por parte del motor. Luego, en el proceso de **comprobación de string**, se verifica la sintaxis de la consulta, asegurándose de que cumpla con los requisitos del sistema.

Una vez validada, la consulta es enviada al **parser**, el cual traduce las sentencias SQL en objetos que serán utilizados en el siguiente nivel del sistema. Estos objetos representan las operaciones lógicas que se ejecutarán posteriormente.

Administrador de RAM y Caché/Planeador

Este nivel es responsable de las operaciones a nivel lógico, manipulando la estructura de datos del sistema. Está compuesto por dos procesos principales:

- **Planeador:** El planeador contiene un conjunto de funciones que verifican las operaciones que serán ejecutadas en el árbol de datos. Además, determina el orden en que dichas operaciones deben realizarse, basado en un plan de ejecución predefinido.
- **Operaciones en RAM:** Aquí es donde se ejecutan las operaciones previamente planificadas y verificadas por el planeador. Este proceso aplica los cambios en la estructura de datos a nivel lógico. Si la consulta es de tipo **selección**, el sistema proporciona una salida con los datos solicitados. Si no es de este tipo, el sistema avanza al siguiente nivel para continuar con la ejecución.

Archivo y Carga en RAM (Nivel Físico)

En este último nivel, los cambios se realizan a nivel físico, es decir, en los archivos que almacenan los datos. Cada vez que se lleva a cabo una modificación, esta es procesada por el módulo de **inteligencia artificial** acoplado, el cual clasifica inteligentemente las imágenes que se insertan.

Es importante destacar que este proceso se repite cada vez que el sistema se enciende, ya que todos los datos almacenados son cargados en la memoria RAM. Antes de este paso, las rutinas de ANN (Redes Neuronales Artificiales) clasifican los datos, lo que dota a ToyDBMS de una capacidad nativa para la clasificación automática de imágenes.

Tras la clasificación de las imágenes mediante las rutinas de ANN, los datos son finalmente cargados en la RAM, y el ciclo continúa según las operaciones solicitadas por el usuario. Este enfoque optimiza el rendimiento del sistema y es un factor clave en la esencia de ToyDBMS, especialmente en su capacidad de gestión eficiente de imágenes.

3. API E INTERFAZ DE TOYDBMS

La interfaz de un sistema de gestión de bases de datos relacional (SGBDR) corresponde al nivel externo en el modelo ANSI-SPARC. En este nivel, los usuarios interactúan con el sistema mediante la introducción de consultas en lenguaje SQL y visualizan los resultados devueltos por el mismo.

En ToyDBMS, se decidió desarrollar una API denominada ToyServer, que facilitaría la comunicación entre el motor de base de datos ToySQL y el usuario a través de una interfaz conocida como ToyClient. Este enfoque permitiría ofrecer una funcionalidad más sencilla para el usuario, lo que resulta altamente adecuado para el despliegue del motor.

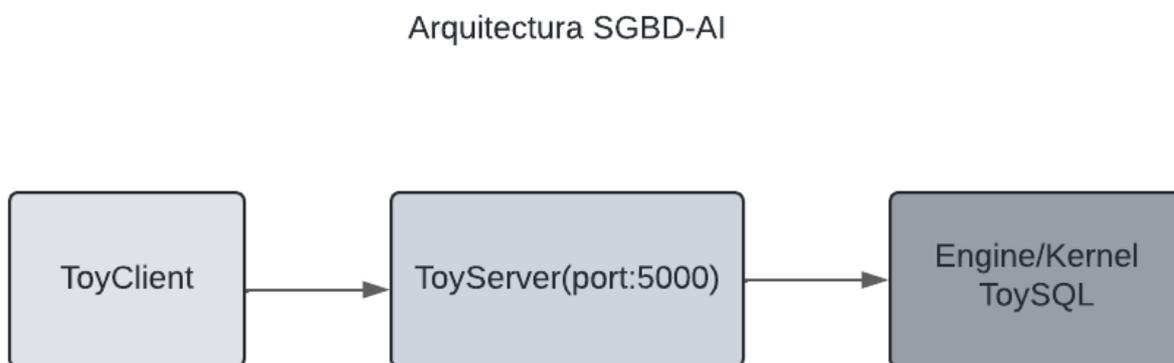


Fig. 9. Arquitectura cliente-servidor de ToyDBMS

4. Detalles de la arquitectura y recolección requisitos

Después de establecer la arquitectura del sistema ToyDBMS, se decidió implementar ToySQL como una biblioteca dentro de la API ToyServer. Esta elección permitió simplificar tanto el desarrollo como la usabilidad del sistema.

Con la arquitectura definida, se procedió a recopilar los respectivos requisitos de información, los cuales se organizaron en el siguiente product backlog.

TABLA I: PRODUCT BACKLOG DEL SISTEMA.

ID	Elemento	Prioridad	Estimación	Estado	Descripción	Criterio de aceptación
1	El sistema debe permitir convertir las consultas SQL en objetos o instancias de clase mediante un proceso de análisis (parsing).	Alta	1 semanas	Completo	Las consultas SQL serán convertidas en objetos compatibles con el lenguaje de programación para su interpretación y manipulación directa.	El sistema debe transformar consultas SQL en objetos de forma precisa y eficiente, manteniendo la estructura y semántica original.
2	El sistema debe soportar la ejecución de operaciones SQL utilizando una estructura de datos.	Alta	3 semanas	Completo	Los objetos generados ejecutarán operaciones básicas como inserción, selección, eliminación y creación de tablas dentro de la estructura de datos.	El sistema debe realizar operaciones de inserción, selección, eliminación y creación de tablas en la estructura de datos de manera precisa y eficiente.
3	El sistema debe permitir almacenar grandes volúmenes de datos en archivos de forma eficiente.	Alta	2 semanas	Completo	Los datos se almacenarán en archivos para asegurar su persistencia y manipulación.	Los datos deben ser almacenados de forma correcta y eficiente en archivos, garantizando su persistencia y fácil acceso para futuras consultas.
4	El sistema debe permitir la carga de datos almacenados en archivos directamente en la estructura de datos	Alta	2 semanas	Completo	Los datos almacenados en archivos serán cargados en la estructura de datos del sistema para asegurar su operatividad durante las consultas.	El sistema debe cargar los datos de archivos en la estructura de datos de forma correcta para permitir su posterior manipulación.

5	El sistema debe permitir la inserción y ejecución de modelos de redes neuronales artificiales (ANN) como parte de su clase principal.	Alta	4 semanas	Completo	El sistema permitirá la integración de modelos de redes neuronales artificiales (ANN) en cualquier fase de ejecución.	El sistema debe ser flexible para integrar modelos ANN en módulos críticos de funcionamiento.
6	El sistema debe estar diseñado para ser utilizado como una biblioteca desplegada a su vez dentro de una API .	Media	1 semana	Completo	El sistema ofrecerá una interfaz fácil de usar para mejorar la interacción entre el motor de base de datos y el usuario.	El sistema debe ser intuitivo y accesible para usuarios con conocimientos básicos de la consola de Windows y SQL.

5. Planificación del desarrollo

Luego de haber completado la investigación preliminar y la definición tanto de los requisitos como del backlog del sistema, se inició el proceso de creación de historias de usuario y la estimación del tiempo para los sprints. En base al product backlog y los requisitos de información definidos, se desarrollaron las siguientes historias de usuario:

TABLA II: HISTORIAS DE USUARIO DEL SISTEMA TOYDMS

HISTORIAS DE USUARIO		
# HUD	Descripción	Criterios de aceptación:
HUD 1	Como desarrollador del sistema, quiero que las consultas SQL se conviertan en objetos o instancias de clase mediante un proceso de análisis (parsing), para poder manipular y analizar las consultas SQL directamente en el lenguaje de programación de una forma eficiente.	El sistema debe transformar las consultas SQL en objetos de forma precisa y eficiente, manteniendo la estructura
HUD 2	Como desarrollador del sistema, quiero que el sistema soporte la ejecución de operaciones SQL utilizando una estructura de datos, para realizar operaciones como inserción, selección, eliminación y creación de tablas de manera precisa y eficiente.	<ul style="list-style-type: none"> - El sistema debe soportar operaciones de SELECT, INSERT, UPDATE y DELETE. - Los resultados de las consultas deben ser retornados en un

		<p>formato comprensible.</p> <ul style="list-style-type: none"> - Las operaciones deben ser eficientes y no afectar la integridad de los datos.
HUD 3	<p>Como desarrollador del sistema, quiero que el sistema almacene grandes volúmenes de datos en archivos de forma eficiente, para asegurar su persistencia y facilitar su manipulación en el futuro.</p>	<p>Los datos deben ser almacenados de forma correcta y eficiente en archivos, garantizando su persistencia y fácil acceso para futuras consultas.</p>
HUD 4	<p>Como desarrollador del sistema, quiero que el sistema permita cargar los datos almacenados en archivos directamente en la estructura de datos, para que el sistema pueda acceder y manipular estos datos de manera eficiente durante las consultas.</p>	<p>El sistema debe cargar los datos desde archivos a la estructura de datos de manera correcta para su posterior manipulación</p>
HUD 5	<p>Como desarrollador del sistema, quiero que el sistema permita la inserción y ejecución de modelos de redes neuronales artificiales (ANN), para realizar análisis predictivos.</p>	<p>El sistema debe ser flexible para permitir la integración de modelos.</p>
HUD 6	<p>Como usuario con conocimientos básicos de SQL y sistemas de bases de datos, quiero que el sistema sea una biblioteca fácil de usar desplegada en una API, para mejorar la interacción entre el motor de base de datos y el usuario.</p>	<p>El sistema debe ser intuitivo y accesible, ofreciendo una interfaz sencilla para usuarios con conocimientos básicos de SQL y consola de Windows.</p>

Se estimaron tres sprints, cada uno con una duración de un mes, en los que se abordarían los requisitos del backlog. Cada sprint incluirá una sesión de Sprint Planning para definir las tareas a realizar y asegurar el cumplimiento de los objetivos, es decir, los requisitos de información establecidos.

En esta fase se determinó la tecnología que sería usada para el desarrollo. Un aspecto clave a destacar es la estructura de datos utilizada a nivel lógico en el desarrollo del motor MySQL, así como el tipo de archivos seleccionados para el almacenamiento de la información.

Para la estructura de datos, se optó por un **árbol rojo-negro** debido a la facilidad que ofrece la biblioteca **bintrees**, la cual proporciona dos opciones: árbol binario y árbol rojo-negro. La elección del árbol rojo-negro se basó en su **eficiencia** y su capacidad de **auto-balanceo constante**, lo que permite mantener tiempos de ejecución **O(log n)** de forma consistente, pudiendo así obtener una

mejoría en el rendimiento de las operaciones del motor de base de datos al almacenamiento[70], se decidió utilizar el formato de archivo **HDF5**, ampliamente reconocido por su **eficiencia en la gestión de grandes volúmenes de datos** y su **facilidad de administración**. Estas características son bastante importantes en un motor de base de datos, ya que optimizan tanto el rendimiento como la escalabilidad en entornos convencionales.

Es importante destacar que esta no fue la primera opción considerada. Inicialmente, se evaluó el uso de archivos planos en formato **.dat**, pero esta alternativa fue descartada debido a sus limitaciones en términos de escalabilidad y rendimiento. Durante las pruebas con archivos planos, se observó que el rendimiento se deterioraba a medida que aumentaba el volumen de datos. Esto se debía a que, para insertar nuevos datos, era necesario reescribir todo el archivo, lo que incrementaba el tiempo de inserción conforme el tamaño del archivo crecía.

El formato **HDF5** proporciona un enfoque dinámico para el almacenamiento de datos. Su estructura jerárquica, junto con la biblioteca asociada, permite agregar nuevos datos sin la necesidad de reescribir completamente el archivo [71], lo que mejora la eficiencia en las operaciones de inserción. Esta estructura jerárquica facilitó la organización de las características de los datos que deben ser integrados en la estructura del motor de base de datos durante su inicialización.

En vista de estas limitaciones con los archivos planos, se llevó a cabo una búsqueda de formatos más adecuados para el proyecto. Finalmente, se eligió **HDF5** debido a su alta disponibilidad y flexibilidad, lo que permitió gestionar grandes volúmenes de datos de manera más eficiente, tanto en la inserción como en la consulta.

En cuanto a la tecnología que sería usada para el desarrollo se optó por el lenguaje de programación Python con un enfoque de programación orientada a objetos (POO). Lo que facilitó el desarrollo del motor en el nivel conceptual y físico de la arquitectura ANSI-SPARC.

Para el desarrollo de este motor fue necesario el uso de bibliotecas y así facilitar la funcionalidad del mismo con desarrollo simplificado. Las bibliotecas escogidas fueron:

sqlparse: Es una biblioteca que permite el análisis, división y formateo de consultas SQL sin realizar validación de sintaxis [63].

Bintrees: Un paquete que proporciona implementaciones eficientes de estructuras de datos, como los árboles rojo-negro y los árboles AVL [64].

NumPy: Es una biblioteca de computación científica que ofrece arreglos multidimensionales y métodos matemáticos optimizados para operar sobre ellos [65].

uuid: Es una biblioteca diseñada para generar identificadores únicos universales (UUIDs)[66].

h5py: Es una biblioteca que facilita la manipulación de archivos en formato HDF5, tanto a nivel alto como bajo [67].

PIL (Python Imaging Library): Esta librería añade capacidades de procesamiento de imágenes al intérprete de Python, proporcionando acceso rápido y soporte para una amplia variedad de formatos. [68]

6. Implementación y retrospectiva

Durante cada sprint, se llevaron a cabo las tareas definidas en el Sprint Planning correspondiente. Al finalizar cada sprint, se realizó una reunión de retroalimentación del proyecto, en la cual se implementaron algunos ajustes técnicos y modificaciones en las tecnologías utilizadas.

El primer cambio importante fue la transición del uso de archivos planos al formato HDF5, lo que permitió un manejo más eficiente de los datos y una mejor escalabilidad para grandes volúmenes de información. El segundo ajuste relevante fue la combinación y posterior división de las clases, siguiendo buenas prácticas de desarrollo para optimizar la estructura del sistema.

Esta fase ha sido crucial para mejorar la calidad del proyecto y adaptarlo a las necesidades del usuario, lo que resalta la importancia de la retroalimentación continua en el proceso de desarrollo con la metodología SCRUM.

7.Finalización de ToyDBMS

Al finalizar el desarrollo del sistema gestor de bases de datos relacional (SGBDR), se llevaron a cabo pruebas de funcionamiento, así como una sesión de retroalimentación para evaluar su desempeño. A partir de estos análisis, se tomaron decisiones estratégicas destinadas a mejorar tanto la experiencia del usuario como la funcionalidad del sistema.

El SGBDR ofrece dos tipos de clientes: uno basado en consola y otro con una interfaz web. Ambos permiten a los usuarios realizar consultas al servidor y recibir respuestas que se visualizan de

manera efectiva, ya sea en la consola o en la interfaz gráfica, brindando así opciones versátiles para interactuar con el sistema.

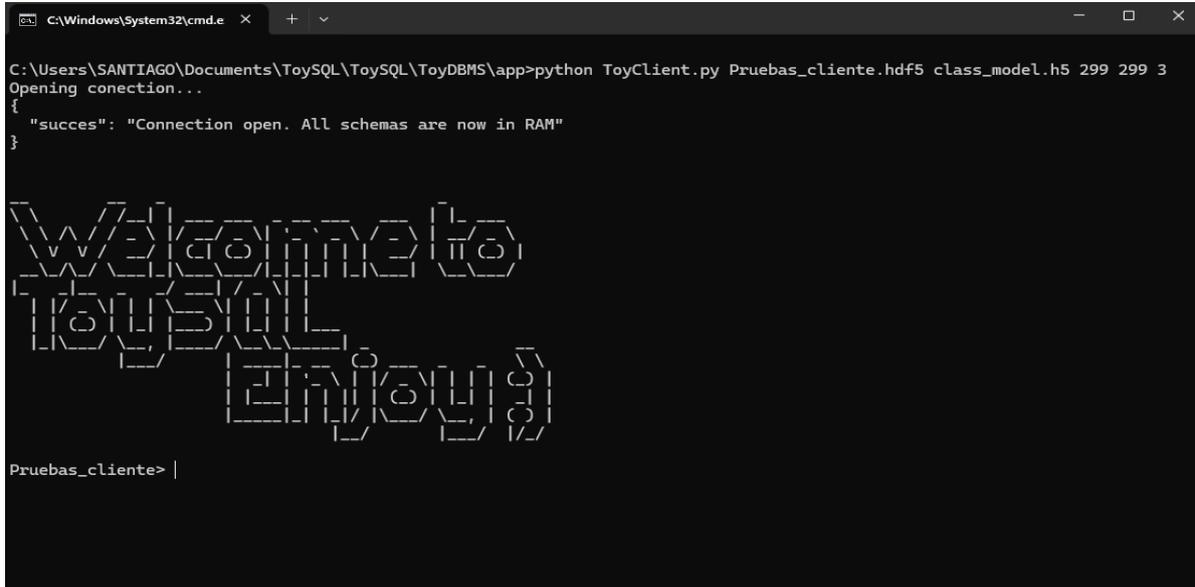


Fig.10. Cliente/interfaz de consola de ToyDBMS.

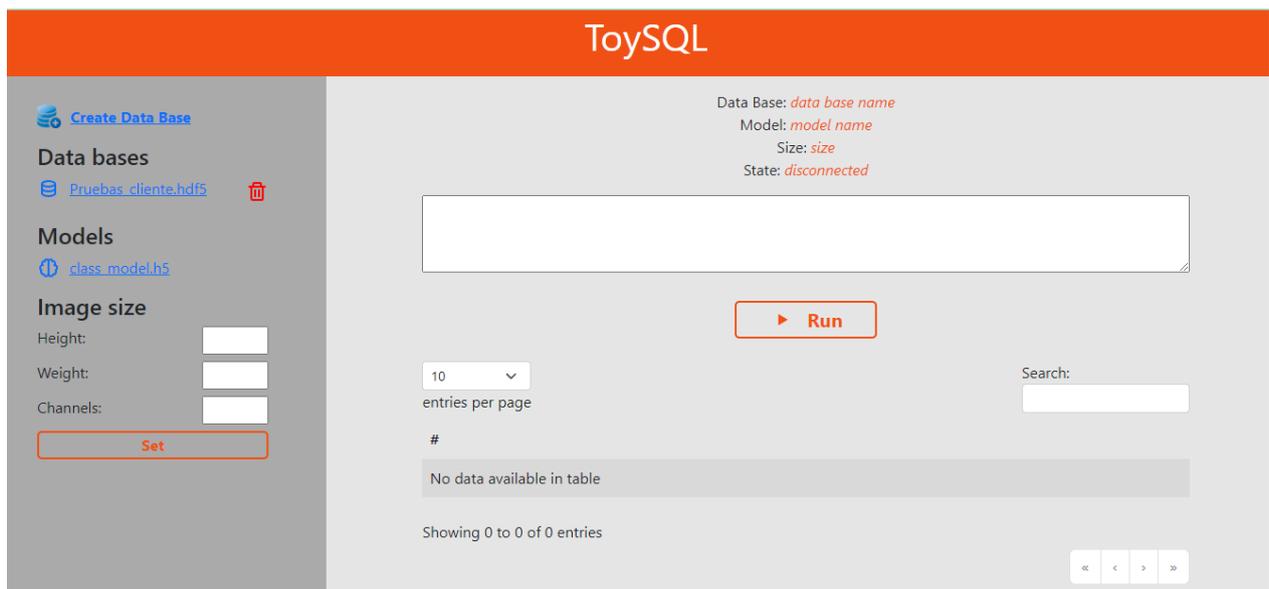


Fig. 11. Cliente/interfaz web de ToyDBMS

Además, el servidor y los clientes pueden ser accedidos a través de otra interfaz que se encarga de iniciar el servidor en el puerto **5000** y redirigir a los usuarios hacia el cliente que deseen utilizar.

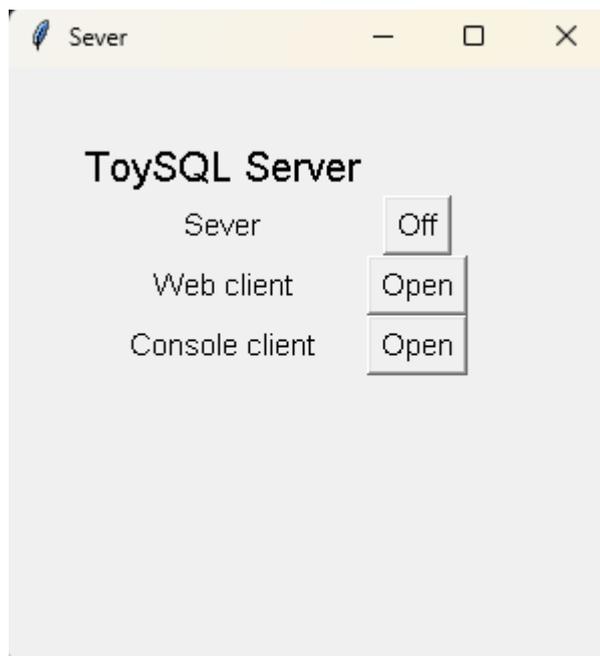


Fig. 12. Interfaz principal de ToyDBMS.

Así concluyo el desarrollo del SGBDR simplificado, con interfaces claramente definidas y un funcionamiento óptimo.

B. Desarrollo del modelo de ANN

Para el desarrollo del modelo, se optó por la metodología CRISP-DM, reconocida en proyectos de este tipo por sus fases analíticas y su flexibilidad en la extracción de información. Esta metodología se compone de seis fases: entendimiento del negocio, entendimiento de los datos, preparación de los datos, modelado, evaluación y despliegue. Estas fases aseguran el correcto desarrollo del modelo requerido y la extracción de la información necesaria para su adecuado funcionamiento.

1. Entendimiento del negocio

Para esta investigación, fue necesario desarrollar un modelo de inteligencia artificial orientado al procesamiento digital de imágenes (PDI). El modelo implementado es de tipo CNN (Red Neuronal Convolutiva), elegido por dos razones principales: en primer lugar, por la alta eficiencia de estas redes en tareas de procesamiento de imágenes, y en segundo lugar, porque se optó por aplicar el enfoque de **Sergio Sánchez** para la recuperación de imágenes mediante descriptores dentro del motor ToySQL. Estos descriptores se obtienen a partir del análisis de imágenes utilizando CNNs, lo que permite generar representaciones útiles para la tarea de recuperación de imágenes.

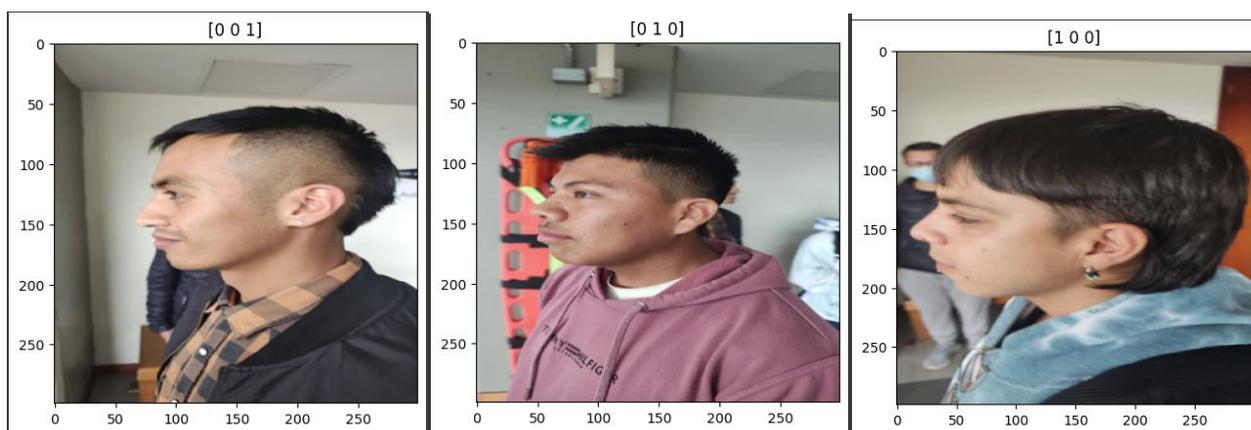
En un punto de la investigación, se contempló la inclusión de otros tipos de modelos, como las **Redes Neuronales Convolucionales Siamesas**, debido a su versatilidad en entradas y salidas. Estas redes comparan dos entradas y devuelven un valor que mide el grado de similitud entre ellas, produciendo una salida que indica el porcentaje de similitud en un rango de cero a uno. Sin embargo, este enfoque no se alineaba con la estrategia de recuperación basada en descriptores, que fue el enfoque elegido para la investigación.

2. Entendimiento y preparación de los datos

El conjunto de datos utilizado para el entrenamiento y las pruebas del modelo consiste en un dataset propio de clasificación de rostros, con un total de 630 imágenes correspondientes a tres personas distintas.

Para aumentar la variabilidad del conjunto de datos, se realizó un **data augmentation** en Google Colab, aplicando transformaciones tanto verticales como horizontales a las imágenes originales mediante un generador de datos.

Cada rostro en el conjunto de datos pertenece a una de las tres clases. Estas clases están etiquetadas utilizando un vector unitario, donde una de las componentes en los ejes **x**, **y** o **z** indica la clase correspondiente. Así, las tres clases se definen de la siguiente manera: **[0,0,1]** representa la clase 1, denominada "Sebastián"; **[0,1,0]** representa la clase 2, denominada "Mayer"; y **[1,0,0]** representa la clase 3, denominada "Santi". Este esquema de etiquetado fue diseñado en función de la arquitectura del modelo.



a.

b.

c.

Figura 13: Imágenes del conjunto de datos utilizado, donde **a**, **b** y **c** corresponden a cada una de las clases del conjunto.

3. Modelado

El modelo desarrollado es una red neuronal convolucional (CNN) que denominamos “**class_model**”. Su arquitectura se basa en **transfer learning** a partir del reconocido modelo **InceptionV3**, al cual se le añadieron capas adicionales para adaptarlo al conjunto de datos de entrenamiento y al tamaño de entrada correspondiente.

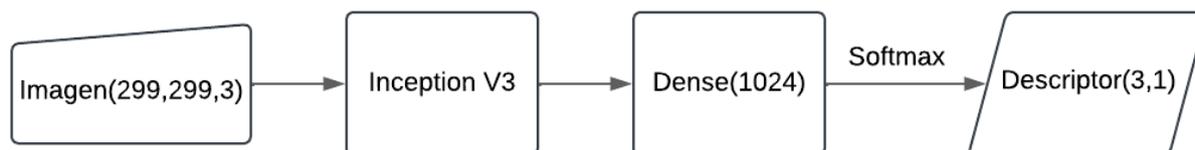


Fig.14. Arquitectura de “class_model”.

Se incorporó una capa densa con 1024 neuronas para mejorar el rendimiento con el conjunto de datos seleccionado. Además, la capa de salida se modificó para incluir 3 neuronas, acompañadas de una función de activación **softmax**, que se encarga de calcular la probabilidad de que una entrada pertenezca a una de las tres clases del conjunto de datos. La entrada del modelo también fue ajustada para aceptar imágenes de tamaño **299x299x3**.

4. Evaluación del modelo

Debido a la arquitectura propuesta, no fue necesario realizar un entrenamiento extenso del modelo. Este fue entrenado durante 10 épocas utilizando el conjunto de datos establecido y los cambios incorporados. Posteriormente, se llevaron a cabo las pruebas correspondientes con el conjunto de pruebas para evaluar su eficiencia. Al finalizar el proceso, se obtuvieron las siguientes métricas:

TABLA III: MÉTRICAS DE ENTRENAMIENTO Y PRUEBA

Conjunto de datos	Accuracy	Loss
Entrenamiento	1.0000	0.0018
Prueba	0.9944	0.0082

A partir de estas métricas, se concluyó que el modelo es efectivo y apto para ser integrado en ToyDBMS.

5. Despliegue del modelo

En esta fase, se llevó a cabo la exportación del modelo en un archivo de tipo h5 utilizando la biblioteca TensorFlow. Posteriormente, esta misma biblioteca se utilizó para realizar las predicciones del modelo dentro de ToyDBMS(ver la figura 8). El modelo es compatible con TensorFlow versión 2.15, y se recomienda evaluar su uso en el futuro o actualizar a versiones más recientes.

C. Acoplamiento del modelo "class_model" y ToyDBMS

La esencia del proyecto radica en esta implementación. Gracias a la flexibilidad desarrollada en ToyDBMS, fue posible crear un nuevo módulo para la comparación de imágenes en un nivel crítico del sistema, manteniendo un acoplamiento estrecho. Esta implementación se realizó a nivel conceptual, es decir, en la lógica del sistema. Como se muestra en la figura 8, la arquitectura final muestra claramente el punto en el que se integra el modelo de redes neuronales artificiales (ANN) dentro del motor.

Este enfoque permitió la carga e inserción inteligente de imágenes dentro de la estructura de datos basada en árboles. El proceso se fundamenta en la extracción de características mediante una red neuronal convolucional (CNN) utilizando el modelo desarrollado llamado "class_model". Este enfoque fue adoptado del trabajo de Sergio Sánchez.

Se creó un atributo llamado "matriz_de_referencia", que se encarga de indexar las imágenes almacenadas en el motor. Este atributo es en realidad un tensor de cuatro dimensiones que contiene las imágenes, las cuales se utilizan para comparar las nuevas entradas. Cada imagen tiene un descriptor generado por la salida del modelo, y dichas salidas se suman y promedian para crear una llave única para cada clase, que sirve como descriptor adicional.

Cuando el modelo recibe una nueva imagen, consulta la matriz de referencia y predice si la clase de esa imagen ya existe. Si la clase ya está presente, la matriz se mantiene sin cambios, y la ruta de la nueva imagen se asocia a la llave de esa clase en el árbol. Si no existe, se crea una nueva llave, se inserta la ruta de la imagen en el árbol, y la imagen aplanada se agrega a la matriz de referencia.

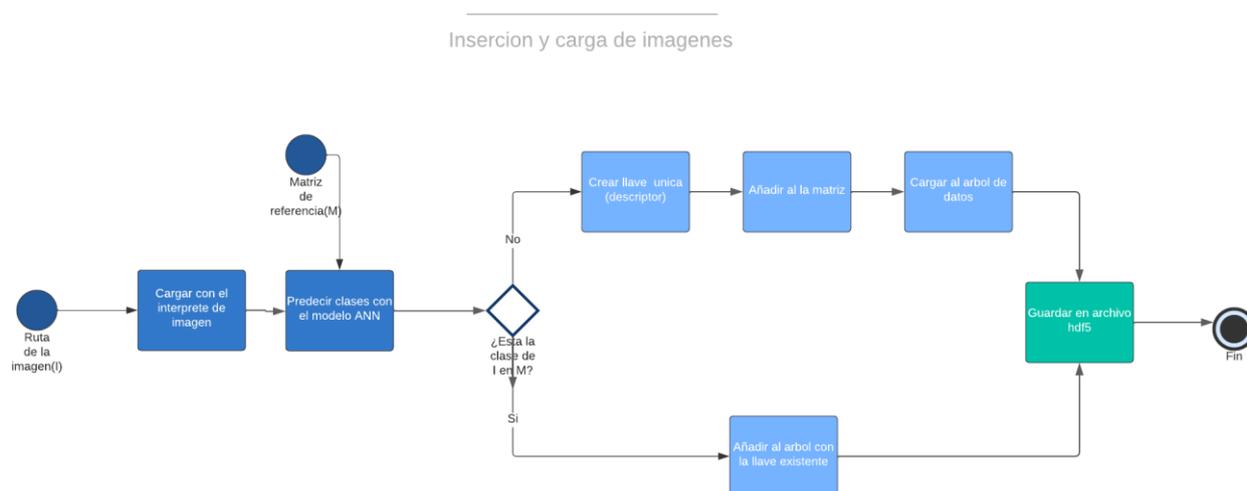


Fig. 15. Diagrama de actividades del funcionamiento del módulo para la inserción y carga de imágenes

Para implementar esta funcionalidad en el módulo, fue necesario introducir un nuevo tipo de dato y una extensión al lenguaje SQL, ambos denominados Unstructured. Este tipo de dato se maneja como una clase independiente, con sus propios métodos para el procesamiento y transformación de imágenes.

Además, el módulo incluye una funcionalidad para realizar consultas mediante la extensión SQL Route('Ruta de la imagen'), la cual permite especificar la ruta de la imagen que se desea analizar o consultar. El parser del sistema fue modificado para interpretar correctamente este tipo de consultas, tanto para la inserción como para la consulta de imágenes.

Al momento de crear una tabla, se utiliza el nuevo tipo de dato Unstructured a través de la cláusula SQL UNSTRUCTURED, la cual indica que la columna recibirá datos de tipo imagen o "no estructurados". El parser se encarga de procesar esta definición y construir las columnas de manera compatible para manejar imágenes.

Por ejemplo, para crear una base de datos llamada base_de_datos con una tabla denominada personas y tres campos: nombre, foto y cédula, la consulta SQL para crear la tabla sería la siguiente:

```
CREATE TABLE base_de_datos (nombre VARCHAR(20),foto UNSTRUCTURED,cedula INT);
```

En cuanto a la inserción de imágenes, se especifica únicamente la ruta de la imagen. El parser transformará la consulta en los objetos y operaciones necesarias para invocar los métodos de la clase Unstructured, que gestionará la inserción de la imagen en la base de datos. La consulta de inserción se vería de la siguiente manera:

```
INSERT INTO personas VALUES ('Pepe', '/ruta/de/foto.jpg', 1005266596);
```

Para realizar consultas de imágenes similares, se utiliza la cláusula Route. Por ejemplo, para obtener todos los registros cuya imagen sea similar a una imagen llamada imagen.jpg, la consulta sería la siguiente:

```
SELECT * FROM personas WHERE foto = Route('/ruta/de/imagen.jpg');
```

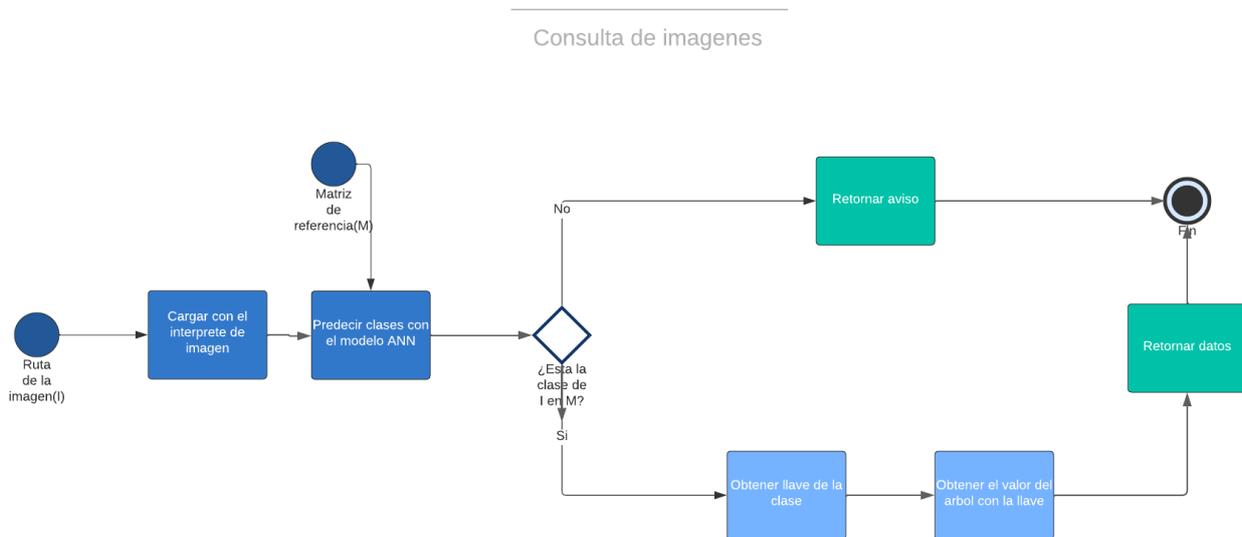


Fig. 16. Diagrama de actividades del funcionamiento del módulo para la selección de imágenes

Estas dos funcionalidades forman el módulo diseñado para integrar redes neuronales artificiales (ANN) con ToyDBMS. El sistema presenta un alto acoplamiento debido a que este módulo se utiliza tanto para la inserción como para la consulta de imágenes, ambas operaciones dependientes directamente de la calidad del modelo ANN. Si el modelo tiene un bajo rendimiento, las imágenes se insertarán incorrectamente en la matriz de referencia, lo que afectará la precisión de las consultas y aumentará los tiempos de respuesta. Esto sucede porque las inserciones incorrectas podrían generar tensores de gran tamaño, dificultando la eficiencia en las consultas.

Dado que múltiples módulos dependen del módulo diseñado para el uso de las ANN, el acoplamiento se vuelve patológico o extremadamente alto. Si el módulo o el modelo se cambian o se dañan, tanto las operaciones de inserción como las de selección se verán afectadas, causando que todo el sistema dependa de la calidad y funcionamiento del módulo y modelo en varias funcionalidades del motor de base de datos.

En un enfoque de acoplamiento moderado, el modelo ANN solo se emplea para las operaciones de selección, pero no para las de inserción. Este enfoque reduce significativamente el acoplamiento, ya que ToyDBMS solo dependería del módulo y modelo para las consultas. Esto crea un acoplamiento más bajo, conocido como acoplamiento de estampilla o de control, donde solo una parte específica del sistema (las consultas) se ve influenciada por la calidad de estos, mientras que otras funcionalidades, como la inserción, permanecen independientes del mismo.

D. Pruebas de eficiencia con alto, mediano acoplamiento y PostgreSQL

El objetivo se cumplió utilizando dos enfoques de integración de ToyDBMS con el módulo para la consulta de imágenes y el sistema PostgreSQL. Se ejecutaron pruebas con distintos volúmenes de datos para medir los tiempos de respuesta en consultas que involucran procesamiento de imágenes. Los datos se insertaron en bloques de 300, 3,000 y 30,000 imágenes, y se midió el tiempo que tomó realizar consultas específicas relacionadas con la comparación de imágenes, estos distribuidos equitativamente entre las tres clases, es decir, el 33.33% de los datos en cada prueba correspondía a cada clase.

Las pruebas se llevaron a cabo en un equipo con las siguientes características:

- **Procesador:** AMD Ryzen 7 4700U con gráficos Radeon, 2.00 GHz.
- **Memoria RAM:** 8 GB (7.36 GB utilizable).
- **Tipo de sistema:** Sistema operativo de 64 bits, procesador x64.
- **Sistema operativo:** Windows 11.
- **Tarjeta gráfica:** No se utilizó una tarjeta gráfica dedicada.

Las pruebas se realizaron utilizando tres enfoques para la integración de las Redes Neuronales Artificiales (ANN) con el Sistema de Gestión de Bases de Datos Relacional (SGBDR):

- Alto Acoplamiento
- Mediano Acoplamiento

- PostgreSQL con plpython3u

Los primeros enfoques en ser evaluados fueron el de alto acoplamiento y mediano acoplamiento, cuyos resultados son presentados en las tablas IV y V.

TABLA IV: RESULTADOS DE PRUEBAS CON ALTO ACOPLAMIENTO EN TOYDMS

300 datos	3000 datos	30000 datos
2.1973395347595215 sec	2.271679162979126 sec	3.467968225479126 sec
2.1126697063446045 sec	2.2187530994415283 sec	3.541433095932007 sec
2.144136667251587 sec	2.168632745742798 sec	5.785330533981323 sec
1.9531021118164062 sec	2.3990602493286133 sec	5.8155293464660645 sec
2.250380039215088 sec	2.2980705904618385 sec	3.1518305965227074 sec
Promedio: 2.132 sec	2.2712 sec	4.35228 sec
Desviación: 0.101 sec	0,0952 sec	1.18723 sec

El alto acoplamiento fue consistentemente el enfoque más rápido, debido a la integración nativa de las ANN con el núcleo del SGBDR. Este enfoque aprovechó al máximo la cercanía de los módulos, reduciendo la necesidad de transferencias de datos entre sistemas y permitiendo que las consultas se ejecutaran directamente sobre las imágenes almacenadas en la base de datos.

TABLA V: RESULTADOS DE PRUEBAS CON MEDIANO ACOPLAMIENTO EN TOYDMS

300 datos	3000 datos	30000 datos
30.28593111038208 sec	288.10817766189575 sec	22121.571089029312 sec
23.29633903503418 sec	253.74544429779053 sec	23210.8110138320565 sec
25.45926782500794 sec	262.26828241348267 sec	22234.1202124782677 sec
24.37780343002105 sec	259.23222374916077 sec	21518.6865838806883 sec
23.83707123252761 sec	259.8378412723541 sec	21356.4955868870407 sec
Promedio: 25.45128 sec	264.63839 sec	22028.33690 sec
Desviación: 2.46513 sec	12.0835 sec	69.1984 sec

El mediano acoplamiento mostró un rendimiento significativamente más lento en comparación con el alto acoplamiento. Esto se debe a la transferencia constante de datos entre el SGBDR y los

módulos de ANN externos, que introduce latencia adicional. Además, la duplicación de algunas operaciones como la preparación de datos antes del procesamiento fue un factor limitante. La Figura 16 muestra el script empleado para realizar las pruebas de rendimiento. Este script está diseñado específicamente para insertar todos los datos y, posteriormente, consultarlos con el fin de medir su desempeño.

```

1 CREATE OR REPLACE FUNCTION public.script_python()
2 RETURNS text
3 LANGUAGE plpython3u
4 AS $function$
5
6     import tensorflow as tf
7     import cv2
8     import psycpg2
9     from keras.models import load_model
10    from PIL import Image
11    import numpy as np
12
13    path_model = '/var/lib/postgresql/14/main/model/class_model.h5'
14    model = load_model(path_model)
15
16    imagen_referencia = '/var/lib/postgresql/14/main/images/Mayer.png'
17
18    connection = psycpg2.connect(
19        host="localhost",
20        database="nombre_base_de_datos",
21        user="postgres",
22        password="contraseña"
23    )
24
25    cursor = connection.cursor()
26
27    sql = "SELECT * FROM users_100;"
28    cursor.execute(sql)
29    datos_postgres = cursor.fetchall()
30
31    cursor.close()
32    connection.close()
33
34    conjunto_similitudes = { 'id': [], 'name': [], 'link': [] }
35    imagenes_procesadas = []
36
37    for dato in datos_postgres:
38        link = dato[2]
39        # Lee la imagen y conviértela a un arreglo NumPy
40        imagen = Image.open(link)
41        imagen = imagen.resize((299, 299), Image.BILINEAR)
42        imagen_array = np.array(imagen)
43        imagen_array = np.expand_dims(imagen_array, axis=0)
44        imagen_array = imagen_array / 255.0 # Normaliza la imagen
45        imagenes_procesadas.append(imagen_array)
46
47        imagen_comparacion = Image.open(imagen_referencia)
48        imagen_comparacion = imagen_comparacion.resize((299, 299), Image.BILINEAR)
49        array_procesado = np.array(imagen_comparacion)
50        array_procesado = np.expand_dims(array_procesado, axis=0)
51        array_procesado = array_procesado / 255.0
52
53        prediccion_1 = model.predict(array_procesado)
54
55        # Procesar 3000 datos
56        for dato in range(10):
57            for indice, valor in enumerate(imagenes_procesadas):
58                prediccion_2 = model.predict(valor)
59                if np.argmax(prediccion_1) == np.argmax(prediccion_2):
60                    conjunto_similitudes["id"].append(datos_postgres[indice][0])
61                    conjunto_similitudes["name"].append(datos_postgres[indice][1])
62                    conjunto_similitudes["link"].append(datos_postgres[indice][2])
63
64        # Contador de imagenes similares
65        return str(len(conjunto_similitudes['id']))
66
67 $function$;
68
69 SELECT * FROM public.script_python()

```

Fig. 17. Script de inserción y consulta para pruebas de alto y mediano acoplamiento.

Finalmente, para las pruebas con postgres se hizo uso de la Extensión plpython3u en PostgreSQL. La extensión plpython3u permite ejecutar scripts escritos en Python dentro de PostgreSQL. Esta funcionalidad es fundamental ya que permite integrar TensorFlow, una biblioteca de aprendizaje

profundo, directamente en el entorno de PostgreSQL, y así realizar tareas de procesamiento de imágenes dentro de la base de datos.

Aunque la extensión plpython3u permite ejecutar modelos de ANN, PostgreSQL no está diseñado específicamente para manejar grandes volúmenes de datos no estructurados (como imágenes) de manera eficiente. Esto resulta en tiempos de procesamiento más largos, ya que no es un sistema nativo de procesamiento de imágenes. Asimismo, el proceso de invocación de funciones externas introduce sobrecarga en el rendimiento

En cuanto a un escenario de uso de PostgreSQL con plpython3u sería ideal cuando se requiere un sistema de base de datos relacional con extensibilidad para funciones de inteligencia artificial, pero el volumen de datos no es extremadamente alto o requerimientos muy estrictos de tiempo de respuesta, no está diseñado para aplicaciones que requieran respuestas en milisegundos o tiempos de respuesta extremadamente bajos, como lo haría un sistema con alto acoplamiento entre IA y el SGBDR. Si el sistema puede tolerar respuestas que tardan unos segundos o incluso minutos, este enfoque es completamente adecuado.

Inicialmente, se identificaron problemas de compatibilidad con el sistema operativo utilizado para las pruebas de rendimiento, lo que llevó a la necesidad de migrar a un nuevo entorno de trabajo. Se eligió Ubuntu, un sistema operativo basado en Linux, ampliamente reconocido por su estabilidad y capacidad de configuración, especialmente en entornos de desarrollo y servidores. Cabe resaltar que todo se hizo en el mismo equipo, cambiando únicamente el sistema operativo en el cual PostgreSQL sería ejecutado.

TABLA VI: RESULTADOS DE PRUEBAS CON POSTGRESQL

300 datos	3000 datos	30000 datos
90.40130138397217 sec	1248.394596338272 sec	5712.597047567368 sec
87.25995540618896 sec	1267.018672466278 sec	6365.359642028809 sec
85.32007956504822 sec	1090.633315563202 sec	5635.178754806519 sec
87.39808869361877 sec	964.468633890152 sec	5314.760447554978 sec
86.70306992530823 sec	969.2264788150787 sec	6286.8945928275407 sec
Promedio: 87.21649 sec	1127.94836 sec	6382.97816 sec
Desviación: 1.8473 sec	132.3931 sec	691.4366 sec

Aunque PostgreSQL con ppython3u mejoró en comparación con el mediano acoplamiento en volúmenes de datos menores, fue mucho más lento en volúmenes mayores. La causa de esto es que PostgreSQL, aunque es extensible, no está optimizado para procesar grandes volúmenes de datos no estructurados (como imágenes) de manera nativa. La extensión ppython3u permitió ejecutar ANN en el mismo entorno de la base de datos, pero el proceso de invocar funciones externas generó sobrecarga significativa.

Los resultados mostraron que el tiempo de procesamiento de las consultas varía considerablemente según la complejidad del modelo de red neuronal empleado y el tamaño de las imágenes procesadas. No obstante, se evidenció una mejora en la eficiencia general al integrar la lógica de procesamiento de imágenes directamente en la base de datos, lo que eliminó la necesidad de transferir grandes volúmenes de datos entre sistemas distintos.

Un aspecto clave a destacar es la estructura de las consultas, ya que tanto la implementación de la función utilizando la biblioteca como la propia consulta pueden afectar de manera significativa el rendimiento y la escalabilidad del enfoque basado en PostgreSQL. La figura 16 ilustran la implementación de la función con ppython3u y la estructura de la consulta propuesta.

Consulta realizada con PostgreSQL.

```

1 CREATE OR REPLACE FUNCTION public.script_python()
2 RETURNS text
3 LANGUAGE plpython3u
4 AS $function$
5
6     import tensorflow as tf
7     import cv2
8     import psychopg2
9     from keras.models import load_model
10    from PIL import Image
11    import numpy as np
12
13    path_model = '/var/lib/postgresql/14/main/model/class_model.h5'
14    model = load_model(path_model)
15
16    imagen_referencia = '/var/lib/postgresql/14/main/images/Mayer.png'
17
18    connection = psychopg2.connect(
19        host="localhost",
20        database="nombre_base_de_datos",
21        user="postgres",
22        password="contraseña"
23    )
24
25    cursor = connection.cursor()
26
27    sql = "SELECT * FROM users_100;"
28    cursor.execute(sql)
29    datos_postgres = cursor.fetchall()
30
31    cursor.close()
32    connection.close()
33
34    conjunto_similitudes = { 'id': [], 'name': [], 'link': [] }
35    imagenes_procesadas = []
36
37    for dato in datos_postgres:
38        link = dato[2]
39        # Lee la imagen y conviértela a un arreglo NumPy
40        imagen = Image.open(link)
41        imagen = imagen.resize((299, 299), Image.BILINEAR)
42        imagen_array = np.array(imagen)
43        imagen_array = np.expand_dims(imagen_array, axis=0)
44        imagen_array = imagen_array / 255.0 # Normaliza la imagen
45        imagenes_procesadas.append(imagen_array)
46
47        imagen_comparacion = Image.open(imagen_referencia)
48        imagen_comparacion = imagen_comparacion.resize((299, 299), Image.BILINEAR)
49        array_procesado = np.array(imagen_comparacion)
50        array_procesado = np.expand_dims(array_procesado, axis=0)
51        array_procesado = array_procesado / 255.0
52
53        prediccion_1 = model.predict(array_procesado)
54
55        # Procesar 3000 datos
56        for dato in range(10):
57            for indice, valor in enumerate(imagenes_procesadas):
58                prediccion_2 = model.predict(valor)
59                if np.argmax(prediccion_1) == np.argmax(prediccion_2):
60                    conjunto_similitudes["id"].append(datos_postgres[indice][0])
61                    conjunto_similitudes["name"].append(datos_postgres[indice][1])
62                    conjunto_similitudes["link"].append(datos_postgres[indice][2])
63
64        # Contador de imagenes similares
65        return str(len(conjunto_similitudes['id']))
66
67 $function$;
68
69 SELECT * FROM public.script_python()

```

Fig. 18. Implementación de la función y consulta en PostgreSQL.

V. ANÁLISIS DE RESULTADOS

El análisis confirma el cumplimiento de la hipótesis de investigación, la cual plantea que la integración de modelos de Redes Neuronales Artificiales (ANN) fuertemente acoplados en un SGBDR mejora el rendimiento en consultas de imágenes en comparación con enfoques tradicionales de acoplamiento medio o bajo. Los resultados obtenidos evidencian esta mejora, mostrando tiempos de respuesta optimizados y un incremento en la eficiencia del procesamiento de datos no estructurados, en particular imágenes. Esta mejora es observable en pruebas de rendimiento comparativo, donde el sistema ToyDBMS supera en rapidez a soluciones como las vistas con la biblioteca ppython3u.

A su vez, se cumple el objetivo general del proyecto, que busca desarrollar un SGBDR que, a través de un alto acoplamiento con ANN, permita medir y optimizar la eficiencia en operaciones DML con imágenes. Este objetivo se alcanza al proporcionar un sistema que facilita la manipulación de datos visuales, y también lo hace de manera eficiente a gran escala. En cuanto a los objetivos específicos, cada uno ha sido logrado: primero, la construcción de un sistema simplificado para acoplar modelos ANN en un entorno relacional se ha completado con éxito, facilitando su integración. Segundo, se han implementado rutinas de entrenamiento, evaluación y almacenamiento en Python, lo que respalda el sistema con modelos ANN funcionales. Tercero, estas rutinas se integraron exitosamente en el SGBDR, formando así un SGBDR-ANN capaz de medir la eficiencia en operaciones DML con imágenes. Y por último la comparación del rendimiento de ToyDBMS con sistemas medianamente acoplados y con PostgreSQL respalda la propuesta experimental, demostrando las mejoras en la escalabilidad de los datos y el tiempo de respuesta del sistema

1. Análisis de eficiencia (tiempos de respuesta)

El objetivo de este capítulo es analizar los resultados obtenidos de las pruebas de eficiencia realizadas con diferentes enfoques de acoplamiento: alto y mediano, así como el enfoque de postgres con la extensión de ppython3u, esto a través de un análisis comparativo

Medir la eficiencia de estos sistemas fue esencial para comprender su rendimiento y efectividad en la ejecución de operaciones de manipulación de datos (DML), y así comprobar la hipótesis propuesta. Ya sabemos que la eficiencia no solo se traduce en tiempos de respuesta más cortos, sino que también impacta en la experiencia del usuario y en la capacidad de las organizaciones para tomar decisiones informadas basadas en datos. Por lo mismo, la evaluación de diferentes

enfoques es crucial en el contexto tecnológico actual. En la tabla VLL se pueden identificar los promedios de cada enfoque evaluado en formato de tabla comparativa.

TABLA VII: COMPARATIVA PROMEDIOS TIEMPOS DE RESPUESTA CADA ENFOQUE PROPUESTO

Cantidad de datos	Alto acoplamiento	Bajo acoplamiento	plpython3u
300	2.132 sec	25.45128 sec	87.21649 sec
3000	2.2712 sec	264.63839 sec	1127.94836 sec
30000	4.35228 sec	2028.33690 sec	6382.97816 sec

A partir de la tabla podemos inferir que el alto acoplamiento presentó el mejor rendimiento entre los enfoques evaluados, mostrando tiempos de respuesta consistentemente bajos y escalabilidad superior, incluso con volúmenes de datos elevados. La integración directa de las ANN dentro del núcleo del sistema de base de datos permitió la optimización del procesamiento de imágenes sin transferencias intermedias de datos.

El enfoque de mediano acoplamiento mostró un rendimiento significativamente inferior, con un incremento en los tiempos de respuesta conforme aumentaba el volumen de datos. La transferencia de datos entre módulos desacoplados fue la principal fuente de latencia, y la duplicación de operaciones internas contribuyó a una degradación considerable del rendimiento-

PostgreSQL con la extensión plpython3u, aunque más eficiente que el mediano acoplamiento en volúmenes pequeños, mostró limitaciones significativas en cuanto a escalabilidad. Los tiempos de respuesta se incrementaron considerablemente a medida que el volumen de datos aumentaba, y el uso de funciones externas para el procesamiento de imágenes resultó en una sobrecarga considerable.

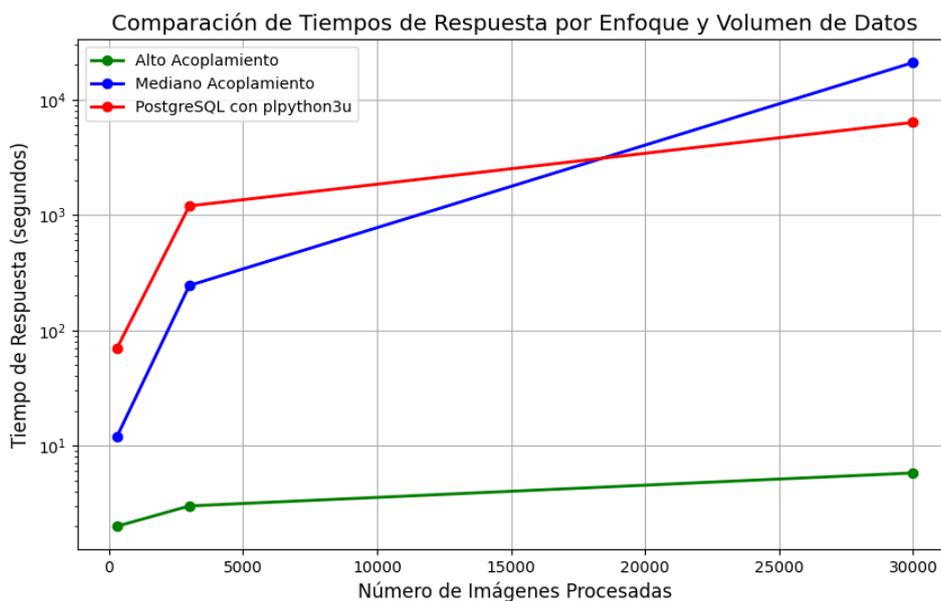


Fig. 19. Grafica de rendimiento de los tres enfoques propuestos (Alto acoplamiento, Mediano acoplamiento, PostgreSQL con plpython3u)

En la figura 19 se puede observar cómo varía el tiempo de enfoque respecto a la cantidad de datos.

- Alto Acoplamiento (línea verde): El tiempo de respuesta es significativamente menor y se mantiene relativamente constante, incluso con 30,000 imágenes.
- Mediano Acoplamiento (línea azul): Muestra un aumento drástico en los tiempos a medida que se incrementa el volumen de datos, sobre todo con 30,000 imágenes, alcanzando un tiempo muy alto.
- PostgreSQL con plpython3u (línea roja): Aunque es más eficiente que el mediano acoplamiento con volúmenes pequeños también aumenta notablemente en tiempos de respuesta a medida que crecen los datos.

Los resultados obtenidos en las pruebas de eficiencia reflejan el rendimiento del sistema propuesto. Se observa que la implementación de alto acoplamiento mostró tiempos de respuesta consistentemente bajos, lo cual valida la hipótesis de que la integración directa de redes neuronales puede mejorar significativamente la eficiencia operativa en comparación con otros enfoques. Aunque estos presentan una tendencia lineal en su rendimiento, se aprecia un “quiebre” específico en las gráficas que corresponde a un punto particular. Esta característica no fue abordada en el análisis de resultados, pero representa una posible área de estudio en futuros trabajos.

2. *Análisis de escalabilidad (cantidad de datos)*

Al analizar los tres enfoques (alto acoplamiento, mediano acoplamiento y PostgreSQL con plpython3u) bajo el parámetro de escalabilidad es posible identificar cómo cada enfoque maneja el aumento en el volumen de datos y cómo este crecimiento impacta los tiempos de procesamiento. Asimismo, se evalúan los factores que contribuyen a la degradación del rendimiento, como las transferencias de datos, la integración de los módulos, y la capacidad de optimización del sistema en diferentes escenarios de carga.

- **Alto Acoplamiento:** El alto acoplamiento mostró una excelente capacidad de escalabilidad, con tiempos de respuesta que aumentaron de manera casi lineal en función del número de datos procesados. Esto se debe a la optimización interna del sistema, que aprovecha al máximo la cercanía entre el SGBDR y las ANN, evitando la transferencia innecesaria de datos.
- **Mediano Acoplamiento:** El mediano acoplamiento, aunque funcional para volúmenes pequeños, mostró una degradación exponencial en el rendimiento cuando se procesaron grandes volúmenes de datos. Los tiempos de respuesta se incrementaron drásticamente, lo que indica que este enfoque no es adecuado para entornos donde se requieren tiempos de respuesta rápidos y grandes volúmenes de datos no estructurados.
- **PostgreSQL con plpython3u:** Aunque PostgreSQL con plpython3u es una solución flexible y extensible, su rendimiento fue inferior en comparación con el alto acoplamiento. El uso de funciones externas como las ANN integradas en Python presentó una sobrecarga significativa en tiempos de consulta, lo que lo hace menos efectivo para grandes volúmenes de datos, pero aún útil para prototipos o aplicaciones con volúmenes de datos moderados.

Al analizar la escalabilidad de los tres enfoques, es evidente que el alto acoplamiento no solo supera al mediano acoplamiento, sino que también se alinea con las conclusiones del estudio realizado por Efraín Oviedo. Su metodología para datos no estructurados resalta la necesidad de una extracción eficiente de características, algo que ToyDBMS logra al mantener una cercanía entre el SGBDR y las ANN. Esto permite que el rendimiento se mantenga estable incluso con un aumento en el volumen de datos, contrastando con el enfoque mediano, donde la latencia se incrementa restrictivamente.

CONCLUSIONES

El proceso de construcción de ToyDBMS representó un desafío significativo para los investigadores, no solo por la complejidad intrínseca de este tipo de sistemas, sino también por la necesidad de hacer el sistema lo suficientemente versátil para que fuera compatible con modelos de redes neuronales artificiales (ANN).

La integración de las ANN con ToyDBMS, gracias a la flexibilidad en el diseño de ambos componentes, se alcanzó de manera efectiva, lo que permitió concluir que un sistema diseñado con módulos flexibles puede aceptar un alto acoplamiento sin comprometer su funcionamiento.

Un enfoque de alto acoplamiento, a pesar de su carácter controvertido en el desarrollo de software, puede ser beneficioso en problemas específicos, como la consulta de imágenes en esta investigación. Estos resultados podrían abrir la puerta a nuevas investigaciones sobre el desarrollo de software y al diseño de arquitecturas innovadoras que aprovechen al máximo la capacidad tecnológica actual.

La integración de SGBDR con redes neuronales artificiales representa un avance crucial para adaptarse a las crecientes demandas del mundo actual. Con el aumento de datos no estructurados, es esencial modernizar estos sistemas para incorporar inteligencia artificial, lo que permite un análisis más eficiente y profundo, al integrar capacidades de IA en los SGBDR, se superan las limitaciones de los enfoques tradicionales, mejorando significativamente la velocidad y precisión en el manejo de datos. Esto abre la puerta a sistemas más inteligentes y adaptativos, vitales para enfrentar los desafíos de un futuro impulsado por los datos.

RECOMENDACIONES

Si bien las pruebas se realizaron en un entorno controlado, el rendimiento de cada enfoque puede estar condicionado por el hardware disponible. Las ANN son intensivas en el uso de CPU y memoria, por lo que el uso de hardware especializado, como GPU (procesadores gráficos), podría mejorar los tiempos de respuesta, especialmente en los enfoques de mediano acoplamiento y PostgreSQL con plpython3u.

En cuanto al diseño del software, la forma en que este se encuentra acoplado a las redes neuronales artificiales (ANN) podría presentar problemas de escalabilidad, no tanto en el volumen de datos, sino en el número de clases, especialmente si se trata de un modelo de clasificación como el utilizado en el acoplamiento del motor, por lo que se recomienda llevar a cabo pruebas de escalabilidad centradas en el número de clases manejadas por los modelos. Además, sería conveniente que futuras investigaciones exploren soluciones a este posible problema, con el fin de optimizar el rendimiento y asegurar una mejor adaptabilidad del sistema en escenarios más complejos.

Se recomienda fomentar la investigación en la aplicación del enfoque de SGBDR fuertemente acoplado con redes neuronales a otros tipos de datos no estructurados, como texto, audio y video. La diversificación de las aplicaciones no solo ampliará el alcance y la utilidad de estos sistemas, sino que también permitirá abordar desafíos específicos en diferentes campos, como la minería de texto en el análisis de opiniones, la clasificación automática de contenido multimedia y el reconocimiento de patrones en videos.

Se propone implementar mecanismos para recoger feedback continuo de los usuarios finales que interactúan con el sistema. Este feedback puede ayudar a identificar áreas de mejora, hacer el sistema más intuitivo, y adaptar las funcionalidades a las necesidades reales de quienes lo utilizan en su día a día. Un enfoque centrado en el usuario garantizará que el SGBDR evolucione de manera alineada con las expectativas y requerimientos del mercado.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Contributeurs aux projets Wikimedia, “Microsoft SQL Server,” May 23, 2023.
https://fr.wikipedia.org/wiki/Microsoft_SQL_Server
- [2] I. Goodfellow, Y. Bengio y A. Courville, Deep Learning (Adaptive Computation and Machine Learning series). MIT Press, 2016.
- [3] “pg_similarity: set of functions and operators for executing similarity queries.”
Accedido el 11 de noviembre de 2023. [En línea]. Disponible:
https://github.com/eulerto/pg_similarity
- [4] M. Gilbert Ginesta y O. Pérez Mora, “PDF de programación - Bases de datos en PostgreSQL”, mayo de 2017.
- [5] Formato de líneas de investigación - Universidad Cesmag formato lineas de investigación A-2020.xlsx (unicesmag.edu.co)
- [6] A. Ali, “Análisis de datos no estructurados: una guía completa,” Astera, Jun. 08, 2023.
<https://www.astera.com/es/type/blog/unstructured-data-analytics-a-complete-guide/>
- [7] Digital Medial Groupm S.L.,” Analítica de datos no estructurados: por qué es tan importante”, [En Línea] .Disponible en: <https://discoverthenew.ituser.es/predictive-analytics/2019/01/analitica-de-datos-no-estructurados-por-que-es-tan-importante,2019>
- [8] R. Gálvez, "Extracción y Análisis de Datos No Estructurados:Aplicaciones usando texto, audio, imágenes y video",Departamento de Computación, FCEyN, UBA
- [9] C. Alvez y A. Vecchietti, “Experiences in Image Management by Content in Object-Relational Databases”, Concordia, noviembre de 2009.
- [10] E. A. Oviedo Carrascal, A. I. Oviedo Carrascal y G. L. Velez Saldarriaga, “Minería multimedia hacia la construcción de una metodología y una herramienta de analítica de datos no estructurados”, Dialnet, vol. 16, n.º 31, pp. 125–142, 2017.

- [11] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. Torr, T. Hospedales “Learning to compare: Relation Network for Few-Shot learning,” arXiv.org, Nov. 16, 2017. <https://arxiv.org/abs/1711.06025>
- [12] “Bases de datos inteligentes en un mundo inteligente”. PowerData. Accedido el 3 de noviembre de 2023. [En línea]. Disponible: <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bases-de-datos-inteligentes-en-un-mundo-inteligente>
- [13] J. Díaz Lazo, “BASE DE DATOS INTELIGENTE”, Revista Digit. Sociedad de la Información, n.º 33, 2012.
- [14] T. Sercu, C. Puhersch, B. Kingsbury e Y. LeCun, "Redes neuronales convolucionales multilingües muy profundas para LVCSR", Conferencia internacional IEEE de 2016 sobre acústica, habla y procesamiento de señales (ICASSP) , Shanghai, China, 2016, págs. 4955-4959, doi: 10.1109/ICASSP.2016.7472620.
- [15] J. Ullman, J. Windom y H. Garcia Molina, Database Systems: The Complete Book. US: Prentice Hall, 2001.
- [16] “¿Qué es Java y por qué lo necesito?” https://www.java.com/es/download/help/whatis_java.html Accedido el 3 de diciembre de 2023
- [17] “PostgreSQL,” PostgreSQL, Dec. 05, 2023. <https://www.postgresql.org/> Accedido el 3 de diciembre de 2023
- [18] “Welcome to Python.org,” Python.org. <https://www.python.org/about/> Accedido el 3 de diciembre de 2023
- [19] Colaboradores de Wikipedia, “PyTorch,” Wikipedia, La Enciclopedia Libre, Dec. 02, 2023. <https://es.wikipedia.org/wiki/PyTorch> Accedido el 3 de diciembre de 2023 TensorFlow, “TensorFlow,” TensorFlow. <https://www.tensorflow.org/>
- [20] S. Sánchez Santiesteban, “Recuperación de imágenes basada en contenido utilizando descriptores generados por redes neuronales convolucionales”, Rev. Cuba. Cienc Inform., vol. 12, n.º 4, pp. 78–90, 2018.

- [21] B. Rivera Covarrubias, L. M. Ledesma Carrillo y C. Rodríguez Doñate, “ANÁLISIS DE SIMILITU PARA SEÑALES E IMÁGENES”, *Jovenes En Cienc.*, vol. 4, n.º 1, 2018.
- [22] “Sistemas Gestores de Bases de Datos - Apuntes - Base de datos I | Apuntes de Sistemas de Gestión de Bases de Datos | Docsity.” <https://www.docsity.com/es/sistemas-gestores-de-bases-de-datos-apuntes-base-de-datos-i/325862/>
- [23] V. Nevado Cabello, *INTRODUCCIÓN A LAS BASES DE DATOS RELACIONALES*. Vis. Libros, 2010
- [24] M. Kleppmann, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media, 2017.
- [25] Nguyendung, “Bases de datos objeto-relacionales - Portal · PDF file Universitat de València Departament d’Informàtica,” *dokumen.tips*, Oct. 2022, [Online]. Available: <https://dokumen.tips/documents/bases-de-datos-objeto-relacionales-portal-de-valencia-departament-dinformatica.html?page=1>
- [26] A. J. QUIJANO VODNIZA, “Mecanismos e Instrumentos para la Planificación, Seguimiento y Evaluación de los Proyectos de Investigación”, trabajo de grado, CESMAG, Pasto, 2004.
- [27] “What is PostgreSQL?,” *PostgreSQL Documentation*, Nov. 09, 2023. <https://www.postgresql.org/docs/current/intro-what-is.html>, Accedido el 28 de noviembre de 2023
- [28] P. Dávila Silva. “SOFTWARE Y HARDWARE”. unam.mx. Accedido el 9 de noviembre de 2023. [En línea]. Disponible: http://www.paginaspersonales.unam.mx/files/490/SOFTWARE_Y_HARDWARE.pdf

- [29] O. G. R. Angel, "Software libre Concepto y definición de software libre, historia y evolución, características de los software libre, software libre y la educación, aplicaciones," Dec. 19, 2018.
<https://repositorio.une.edu.pe/handle/20.500.14039/4083?show=full>
- [30] Encyclopedia of Computer Science (Encyclopedia of Computer Science, 4th Edition, Anthony Ralston (Editor), Edwin D. Reilly (Editor), David Hemmendinger (Editor), Wiley, 2000.
- [31] J. A. Alonso, "Programación declarativa (2008-09)." <https://www.cs.us.es/~jalonso/cursos/pd-08/> Accedido el 3 de diciembre de 2023
- [32] J. Martínez, Fundamentos de programación en Java. Independently, 2020. Accedido el 3 de diciembre de 2023
- [33] AcensTechnologies, "Framework para el desarrollo ágil de aplicaciones," Whitepaper, pp. 1-10.
- [34] E. Siman. "¿Qué es un conjunto de datos? Guía definitiva". Bright data. Accedido el 22 de noviembre de 2023. [En línea]. Disponible: <https://brightdata.es/blog/datos-web/what-is-a-dataset>
- [35] D. Urrutia, "Qué es Entorno de desarrollo - Definición y ejemplos," Arimetrics, Oct. 17, 2023. <https://www.arimetrics.com/glosario-digital/entorno-de-desarrollo> Accedido el 4 de diciembre de 2023
- [36] A. Garros, "Introducción a Google Colab," 2020. [Online]. Available: https://materias.df.uba.ar/f1qa2020c2/files/2020/09/guia_colab_01_intro_pdf.pdf#:~:text=Breve%20introducci%C3%B3n%20a%20Colab%20Lo%20que%20est%C3%A1s%20leyendo,de%20programaci%C3%B3n%20elegido%20para%20este%20entorno%20es%20Python.
- [37] Equipo editorial, Etecé, "Google Drive: qué es, cómo funciona y características," Enciclopedia Humanidades, Sep. 21, 2023. <https://humanidades.com/google-drive/>
- [38] P. Ponce Cruz, Inteligencia artificial con aplicaciones a la ingeniería - 1. ed. Santiago de Chile: Alfaomega, 2010.

- [39] Á. Artola Moreno, “Clasificación de imágenes usando redes neuronales convolucionales en Python”, resumen extendido de Trabajo de grado, Univ. Sevilla, Sevilla, 2019.
- [40] A. Dominguez Torres, “Procesamiento digital de imágenes”, Red Rev. Cient. Acceso Abierto Diam., 1996.
- [41] “¿Qué es machine learning?” IBM. Accedido el 4 de diciembre de 2023. [En línea]. Disponible: <https://www.ibm.com/es-es/topics/machine-learning>
- [42] “¿Qué es Deep Learning? | IBM.” Accedido el 4 de diciembre de 2023. [En línea]. <https://www.ibm.com/es-es/topics/deep-learning>
- [43] Sistemas Expertos e Inteligencia Artificial / Guía XII. (n.d.). Introducción a TensorFlow.
- [44] Ciberseg, “¿Qué es PyTorch? Todo lo que debes saber,” Ciberseguridad, Feb. 02, 2022. Accedido el 1 de diciembre de 2023 <https://ciberseguridad.com/guias/nuevas-tecnologias/machine-learning/pytorch/>
- [45] Y. Sun, H. Lu, L. Zhang, J. Yang y H. Huang, “Ciencia de la Inteligencia e Ingeniería de Big Data”, en Septima Conf. Int., Dalian, China, 22–23 de septiembre de 2017. Beijing: Peking Univ.
- [46] A. Simran, P. Kumar, and S. Bangalore, “Content based image retrieval using deep learning convolutional neural network,” IOP Conference Series, vol. 1084, no. 1, p. 012026, Mar. 2021.
- [47] G. GONZÁLEZ SÁNCHEZ. “Similitud de coseno: comprensión de las matemáticas y cómo funcionan (con códigos de Python)”. ML-Studio. Accedido el 14 de noviembre de 2023. [En línea]. Disponible: <https://mlstudio.jaol.net/similitud-del-coseno/>.
- [48] L. Cuenya y E. Ruetti, “Epistemological and Methodological Controversies Between the Qualitative and Quantitative Paradigm in Psychology”, Rev. Colomb. Psicol, julio de 2010.

- [49] C. L. José, “Apuntes sobre métodos de investigación.” http://scielo.sld.cu/scielo.php?pid=S1727-897X2016000100016&script=sci_arttext&tlng=pt
- [50] M. H. Badii, M. Castillo-Rodríguez, A. C. W. Wong, and P. Villalpando, “Diseños experimentales e investigación científica,” *Innovaciones De Negocios*, vol. 4, no. 8, Dec. 2017, doi: 10.29105/rinn4.8-5.
- [51] E. Carrero, “Experimentos como instrumento de recolección de datos,” *TodoSobreTesis*, Sep. 21, 2023., [En Línea]. Disponible en: <https://todosobretesis.com/experimentos-como-instrumento-de-recoleccion-de-datos/>
- [52] Colaboradores de Wikipedia, “Hardware,” *Wikipedia, La Enciclopedia Libre*, Dec. 05, 2023. <https://es.wikipedia.org/wiki/Hardware>, Accedido el 4 de diciembre de 2023
- [53] Salario para Ingeniero De Sistemas en Colombia - Salario Medio,” *Talent.com*. <https://co.talent.com/es/salary?job=ingeniero+de+sistemas>
- [54] Palacio, C. M. L. (2023, January 16). En 2.023 la hora cátedra en IES privadas debe pagarse, como mínimo, en \$41.667, más prestaciones. *El Observatorio De La Universidad Colombiana*. <https://www.universidad.edu.co/en-2-023-la-hora-catedra-en-ies-privadas-debe-pagarse-como-minimo-en-48-333-mas-prestaciones/>
- [55] Applications",2017“Almacenamiento de Google One,” *Google One*.https://one.google.com/storage?i=m&utm_source=drive&utm_medium=web&utm_campaign=g1_widget_normal#upgrade
- [56] Co, M. (n.d.). *Internet Hogar - Planes de internet - Movistar Colombia*. *Movistar*. <https://www.movistar.com.co/hogar/planes-internet-hogar>
- [57] Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.
- [58] *Sistemas gestores de bases de datos*
- [59] Kelechava, B. (2018, octubre 5). *The SQL standard - ISO/IEC 9075:2023 (ANSI X3.135) - ANSI blog*. *The ANSI Blog; American National Standards Institute*. <https://blog.ansi.org/sql-standard-iso-iec-9075-2023-ansi-x3-135/>

- [60] (S/f). Recuperado el 13 de octubre de 2024, de http://chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.infor.uva.es/~fdiaz/sd/2005_06/doc/SD_TE02_20060305.pdf
- [61] (S/f-b). Recuperado el 13 de octubre de 2024, de <http://chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- [62] Sqlparse. PyPI, de <https://pypi.org/project/sqlparse/>
- [63] Bintrees. PyPI, de <https://pypi.org/project/bintrees/>
- [64] NumPy documentation — NumPy v2.1 manual. (s/f). Numpy.org, de <https://numpy.org/doc/stable/>
- [65] uuid — UUID objects according to RFC 4122. (s/f). Python Documentation, de <https://docs.python.org/3/library/uuid.html>
- [66] H5py. (s/f). PyPI, de <https://pypi.org/project/h5py/>
- [67] Pillow. (s/f). PyPI, de <https://pypi.org/project/pillow/>
- [68] Haya, P. (2021, noviembre 29). La metodología CRISP-DM en ciencia de datos - IIC. Instituto de Ingeniería del Conocimiento.
- [69] Instituto de Ingeniería del Conocimiento (IIC), "Metodología CRISP-DM en Ciencia de Datos," [En línea]. Disponible en: <https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/>.
- [70] D. Viajes, "Árboles: Clasificación y Características," [En línea]. Disponible en: https://dviajes-images-dev.s3.amazonaws.com/07_arboles_55962513c3.pdf.
- [71] L. Mertz, "Design and Implementation of Neural Networks," en *Advanced Techniques in Neural Networks*, Springer, 2011, pp. 153-169. Disponible en: https://link.springer.com/chapter/10.1007/978-1-4419-7719-9_10.

ANEXOS

San Juan de Pasto, 30 de Octubre del 2023

Docente:

Mg. Carlos Fernando González Guzmán
PROGRAMA DE INGENIERÍA DE SISTEMAS

Asunto: Revisión investigación en la Construcción de un S.G.B.D.R fuertemente acoplado con Redes Neuronales Artificiales para la medición de la eficiencia en operaciones D.M.L con imágenes.

Saludo de Paz y Bien.

Yo, **Héctor Andrés Mora Paz** identificado con cédula de ciudadanía Nro. **1085251119**, expreso que he revisado el documento de investigación "Construcción de un S.G.B.D.R fuertemente acoplado con Redes Neuronales Artificiales para la medición de la eficiencia en operaciones D.M.L con imágenes", realizado por los estudiantes: **Oliva Chingal Santiago Alexander, Malpud Noguera Mayer Sebastian y Cifuentes Oliva Sebastian**, de igual manera, expresé que han asistido a los procesos de asesoramiento, por lo tanto, doy aval para revisión por parte docente del espacio académico de Investigación I.

Atte.



Mg. Héctor Andrés Mora Paz

Anexo 1: Aval Asesor del proyecto

Anexo 2: Manual de usuario del sistema.

El Manual de Usuario describe detalladamente los pasos para la correcta utilización del sistema ToyDBMS, incluyendo la configuración inicial, instrucciones para el procesamiento de imágenes, y los procedimientos de consulta. Este manual se encuentra en un documento aparte, disponible para los usuarios como referencia de apoyo.

Para una guía detallada de uso, consulte el Manual de Usuario en:

Anexo manual de usuario de ToySQL

Construcción de un S.G.B.D.R fuertemente acoplado con Redes Neuronales. pág. 85

Anexo 3: Programa Ejecutable

El repositorio [ToyDBMS](https://github.com/Santixddd2/ToyDBMS) contiene la implementación de un Sistema de Gestión de Bases de Datos Relacional (SGBDR) con integración de Redes Neuronales Artificiales para optimizar operaciones DML con imágenes.

Enlace repositorio:

<https://github.com/Santixddd2/ToyDBMS>

 <p>UNIVERSIDAD CESMAG NIT: 800.109.387-7 VIGILADA MREEDUCACIÓN</p>	CARTA DE ENTREGA TRABAJO DE GRADO O TRABAJO DE APLICACIÓN – ASESOR(A)	CÓDIGO: AAC-BL-FR-032
		VERSIÓN: 1
		FECHA: 09/JUN/2022

San Juan de Pasto, 10 de marzo del 2025

Biblioteca
REMIGIO FIORE FORTEZZA OFM. CAP.
Universidad
CESMAG Pasto

Saludo de paz y bien.

Por medio de la presente se hace entrega del Trabajo de Grado / Trabajo de Aplicación denominado Construcción de un S.G.B.D.R fuertemente acoplado con Redes Neuronales Artificiales para la medición de la eficiencia en operaciones D.M.L con imágenes, presentado por el (los) autor(es) Mayer Sebastian Malpud Noguera, Santiago Alexander Oliva Chingal y Sebastian Cifuentes Oliva del Programa Académico ingeniería de sistemas al correo electrónico biblioteca.trabajosdegrado@unicesmag.edu.co. Manifiesto como asesor(a), que su contenido, resumen, anexos y formato PDF cumple con las especificaciones de calidad, guía de presentación de Trabajos de Grado o de Aplicación, establecidos por la Universidad CESMAG, por lo tanto, se solicita el paz y salvo respectivo.

Atentamente,



(Firma del Asesor)

Héctor Andres Mora Paz
108525119
Ingeniería de Sistemas
3172537641
hamora@unicesmag.edu.co

 UNIVERSIDAD CESMAG <small>NIT: 800.109.387-7 VIGILADA MREDCACIÓN</small>	AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE GRADO O TRABAJOS DE APLICACIÓN EN REPOSITORIO INSTITUCIONAL	CÓDIGO: AAC-BL-FR-031
		VERSIÓN: 1
		FECHA: 09/JUN/2022

INFORMACIÓN DEL (LOS) AUTOR(ES)	
Nombres y apellidos del autor: Mayer Sebastian Malpud Noguera	Documento de identidad: 1004542900
Correo electrónico: nogueramayer1000@gmail.com	Número de contacto: 3114034437
Nombres y apellidos del autor: Santiago Alexander Oliva Chingal	Documento de identidad: 1004233250
Correo electrónico: santixddd222@gmail.com	Número de contacto: 3187847528
Nombres y apellidos del autor: Sebastian Cifuentes Oliva	Documento de identidad: 1193029978
Correo electrónico: sebitaz200114@gmail.com	Número de contacto: 3158015460
Nombres y apellidos del asesor: Héctor Andres Mora Paz	Documento de identidad: 108525119
Correo electrónico: hamora@unicesmag.edu.co	Número de contacto: 3172537641
Título del trabajo de grado: Construcción de un S.G.B.D.R fuertemente acoplado con Redes Neuronales Artificiales para la medición de la eficiencia en operaciones D.M.L con imágenes.	
Facultad y Programa Académico: Facultad de ingeniería - programa ingeniería de sistemas	

En mi (nuestra) calidad de autor(es) y/o titular (es) del derecho de autor del Trabajo de Grado o de Aplicación señalado en el encabezado, confiero (conferimos) a la Universidad CESMAG una licencia no exclusiva, limitada y gratuita, para la inclusión del trabajo de grado en el repositorio institucional. Por consiguiente, el alcance de la licencia que se otorga a través del presente documento, abarca las siguientes características:

- a) La autorización se otorga desde la fecha de suscripción del presente documento y durante todo el termino en el que el (los) firmante(s) del presente documento conserve(mos) la titularidad de los derechos patrimoniales de autor. En el evento en el que deje(mos) de tener la titularidad de los derechos patrimoniales sobre el Trabajo de Grado o de Aplicación, me (nos) comprometo (comprometemos) a informar de manera inmediata sobre dicha situación a la Universidad CESMAG. Por consiguiente, hasta que no exista comunicación escrita de mi(nuestra) parte informando sobre dicha situación, la Universidad CESMAG se encontrará debidamente habilitada para continuar con la publicación del Trabajo de Grado o de Aplicación dentro del repositorio institucional. Conozco(conocemos) que esta autorización podrá revocarse en cualquier momento, siempre y cuando se eleve la solicitud por escrito para dicho fin ante la Universidad CESMAG. En estos eventos, la Universidad CESMAG cuenta con el plazo de un mes después de recibida la petición, para desmarcar la visualización del Trabajo de Grado o de Aplicación del repositorio institucional.

 <p>UNIVERSIDAD CESMAG NIT: 800.109.387-7 VIGILADA MIREDCACIÓN</p>	AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE GRADO O TRABAJOS DE APLICACIÓN EN REPOSITORIO INSTITUCIONAL	CÓDIGO: AAC-BL-FR-031
		VERSIÓN: 1
		FECHA: 09/JUN/2022

- b) Se autoriza a la Universidad CESMAG para publicar el Trabajo de Grado o de Aplicación en formato digital y teniendo en cuenta que uno de los medios de publicación del repositorio institucional es el internet, acepto(amos) que el Trabajo de Grado o de Aplicación circulará con un alcance mundial.
- c) Acepto (aceptamos) que la autorización que se otorga a través del presente documento se realiza a título gratuito, por lo tanto, renuncio(amos) a recibir emolumento alguno por la publicación, distribución, comunicación pública y/o cualquier otro uso que se haga en los términos de la presente autorización y de la licencia o programa a través del cual sea publicado el Trabajo de grado o de Aplicación.
- d) Manifiesto (manifestamos) que el Trabajo de Grado o de Aplicación es original realizado sin violar o usurpar derechos de autor de terceros y que ostento(amos) los derechos patrimoniales de autor sobre la misma. Por consiguiente, asumo(asumimos) toda la responsabilidad sobre su contenido ante la Universidad CESMAG y frente a terceros, manteniéndola indemne de cualquier reclamación que surja en virtud de la misma. En todo caso, la Universidad CESMAG se compromete a indicar siempre la autoría del escrito incluyendo nombre de(los) autor(es) y la fecha de publicación.
- e) Autorizo(autorizamos) a la Universidad CESMAG para incluir el Trabajo de Grado o de Aplicación en los índices y buscadores que se estimen necesarios para promover su difusión. Así mismo autorizo (autorizamos) a la Universidad CESMAG para que pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

NOTA: En los eventos en los que el trabajo de grado o de aplicación haya sido trabajado con el apoyo o patrocinio de una agencia, organización o cualquier otra entidad diferente a la Universidad CESMAG. Como autor(es) garantizo(amos) que he(hemos) cumplido con los derechos y obligaciones asumidos con dicha entidad y como consecuencia de ello dejo(dejamos) constancia que la autorización que se concede a través del presente escrito no interfiere ni transgrede derechos de terceros.

Como consecuencia de lo anterior, autorizo(autorizamos) la publicación, difusión, consulta y uso del Trabajo de Grado o de Aplicación por parte de la Universidad CESMAG y sus usuarios así:

- Permiso(permitimos) que mi(nuestro) Trabajo de Grado o de Aplicación haga parte del catálogo de colección del repositorio digital de la Universidad CESMAG por lo tanto, su contenido será de acceso abierto donde podrá ser consultado, descargado y compartido con otras personas, siempre que se reconozca su autoría o reconocimiento con fines no comerciales.

En señal de conformidad, se suscribe este documento en San Juan de Pasto a los 10 días del mes de marzo del año 2025.

<i>Mayer Malpud</i>	<i>Firma del autor</i> <i>Santiago Oliva</i>
Firma del autor	Firma del autor
Mayer Sebastian Malpud Noguera	Santiago Alexander Oliva Chingal
<i>Sebastian Cifuentes O.</i>	<i>Firma del autor</i>
Firma del autor	Firma del autor
Sebastian Cifuentes Oliva	Nombre del autor:



UNIVERSIDAD
CESMAG

NIT: 800.109.387-7
VIGILADA MINEDUCACIÓN

**AUTORIZACIÓN PARA PUBLICACIÓN DE TRABAJOS DE
GRADO O TRABAJOS DE APLICACIÓN EN
REPOSITORIO INSTITUCIONAL**

CÓDIGO: AAC-BL-FR-031

VERSIÓN: 1

FECHA: 09/JUN/2022

Firma del asesor

Nombre del asesor:

Héctor Andres

Mora Paz