



TECNOLÓGICO
NACIONAL DE MÉXICO



Educación
Secretaría de Educación Pública



**Tecnológico Nacional de México
Instituto Tecnológico de Orizaba
División de Estudios de Posgrado e Investigación**

Maestría en Sistemas Computacionales

PROYECTO DE TESIS

TÍTULO DEL PROYECTO:

**Desarrollo de un *Data Lakehouse* para identificar los factores necesarios
para alcanzar la remisión en la diabetes tipo 2**

PRESENTADO POR:

I.S.C Morales Rosales Carlos Andrés M18011260

PARA OBTENER EL GRADO DE:

Maestro en Sistemas Computacionales

DIRECTOR DEL PROYECTO DE TESIS:

Dra. Lisbeth Rodríguez Mazahua

CODIRECTOR DEL PROYECTO DE TESIS:

Dra. Nidia Rodríguez Mazahua

Febrero 2026



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Orizaba
Departamento de División de Estudios de Posgrado e Investigación

Orizaba, Veracruz, **21/septiembre/2026**
Dependencia: **División de Estudios de
Posgrado e Investigación**
Asunto: **Autorización de Impresión**
OPCION: I

C. CARLOS ANDRÉS MORALES ROSALES
CANDIDATO A GRADO DE MAESTRO EN:
SISTEMAS COMPUTACIONALES
P R E S E N T E.-

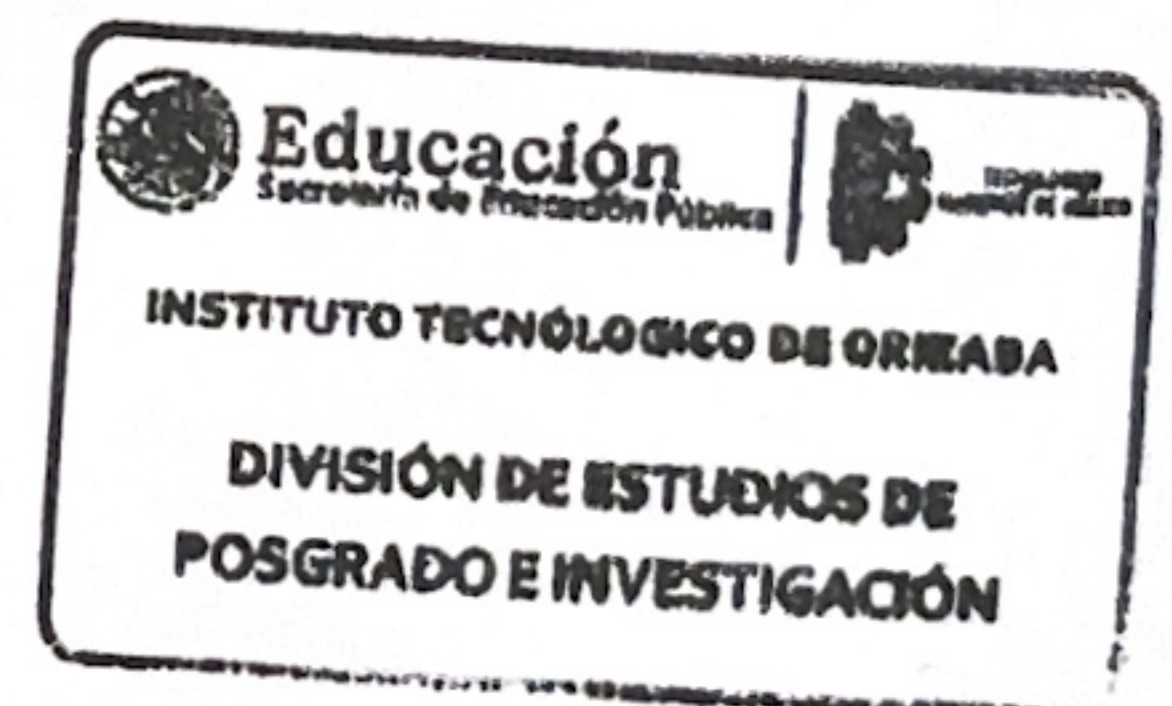
De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

" Desarrollo de un data lakehouse para identificar los factores necesarios para alcanzar la remisión en la diabetes tipo 2"

comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

ATENTAMENTE
Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®

DR. OSCAR BAEZ SENTIES
**JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN**



OG-13-F06



2026
año de
**Margarita
Maza**





Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Orizaba
Departamento de División de Estudios de Posgrado e Investigación

C. OSCAR BAEZ SENTIES
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
PRESENTE.-

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C.

CARLOS ANDRÉS MORALES ROSALES

La cual lleva el título de:

"Desarrollo de un data lakehouse para identificar los factores necesarios para alcanzar la remisión en la diabetes tipo 2"

Y concluyen que se acepta.

ATENTAMENTE

Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®

PRESIDENTE: Dra. Lisbeth Rodríguez Mazahua

SECRETARIO: Dr. José Luis Sánchez Cervantes

VOCAL: Dr. Giner Alor Hernández

VOCAL SUP: Dra. Nidia Rodríguez Mazahua

FIRMA

FIRMA

FIRMA

FIRMA

TA-09-18



2026
año de
Margarita
Maza

Av. Oriente 9 #852 Colonia Emiliano Zapata, C.P. 94320 Orizaba, Veracruz
Tel. 01 (272) 1105360 info@orizaba.tecnm.mx www.orizaba.tecnm.mx



CARTA DE ORIGINALIDAD

En la ciudad de Orizaba, Veracruz, el día 6 del mes de Octubre del año 2026, el (la) que suscribe Carlos Andrés Morales Rosales, alumno (a) del programa de Maestría en Sistemas Computacionales con número de control , manifiesta que es autor (a) del trabajo de tesis titulado “Desarrollo de un Data Lakehouse para identificar los factores necesarios para alcanzar la remisión en la diabetes tipo 2” y declara que el trabajo es original, ya que su contenido es producto de su directa contribución intelectual. Todos los datos y las referencias a materiales ya publicados están debidamente identificados con su respectivo crédito e incluidos en las notas bibliográficas y en las citas que se destacan como tal y, en los casos que así lo requieran, se tienen las debidas autorizaciones de quienes poseen los derechos patrimoniales. Por lo tanto, se hace responsable de cualquier litigio o reclamación relacionada con derechos de propiedad intelectual, exonerando de toda responsabilidad al Tecnológico Nacional de México / Instituto Tecnológico de Orizaba.

A handwritten signature in black ink, appearing to read 'Carlos Andrés Morales Rosales', written in a cursive style.

Carlos Andrés Morales Rosales

Nombre y firma autógrafa

CARTA DE CESIÓN DE DERECHOS

En la ciudad de Orizaba, Veracruz el día 6 del mes de Octubre del año 20, el (la) que suscribe Carlos Andrés Morales Rosales alumno (a) del programa de Maestría en Sistemas Computacionales con número de control M18011260, manifiesta que es autor (a) intelectual del trabajo de tesis bajo la dirección de la doctora Lisbeth Rodriguez Mazahua y ceden los derechos del trabajo intitulado “Desarrollo de un Data Lakehouse para identificar los factores necesarios para alcanzar la remisión en la diabetes tipo 2” al TecNM/Instituto Tecnológico de Orizaba para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y del director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: msc@orizaba.tecnm.mx . Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

A handwritten signature in black ink, appearing to read 'Carlos Morales Rosales', with a large, stylized 'R' at the end.

Carlos Andrés Morales Rosales

Nombre y firma

Contenido

Índice de tablas	5
Índice de figuras	6
Resumen	8
Introducción	10
Capítulo 1. Antecedentes	11
1.1 Marco teórico	11
1.1.1 Diabetes	11
1.1.2 Remisión de diabetes	13
1.1.3 Medición de glucosa	13
1.1.4 <i>Data Warehouse</i>	14
1.1.5 <i>Data Lake</i>	14
1.1.6 <i>Data Lakehouse</i>	15
1.1.7 <i>Big Data</i>	15
1.1.8 Repositorio de datos	17
1.1.9 Aprendizaje automático	17
1.2 Situación tecnológica, económica y operativa de la empresa	18
1.3 Planteamiento del problema	18
1.4 Objetivos general y específico	19
1.4.1 Objetivo general	19
1.4.2 Objetivos específicos	19
1.5 Justificación	20
Capítulo 2. Estado de la práctica	21
2.1 Trabajos relacionados	21
2.2 Análisis comparativo	32

2.3 Propuesta de solución.....	44
Capítulo 3. Aplicación de la metodología	46
3.1 Análisis.....	46
3.3 Desarrollo del Data Lakehouse	87
3.4 Diseño de la aplicación	87
3.4.1 Arquitectura	88
3.4.2 Modelado de la estructura	90
3.4.3 Despliegue.....	92
3.4.4 Ingesta	93
3.4.5 Procesamiento y Almacenamiento.....	94
3.4.6 Análisis	98
3.4.7 Visualización	100
3.4.8 Implementación de LLM's en ingesta	101
Capítulo 4. Resultados	103
4.1 Resultados de la investigación.....	103
4.2 Descripción final del esquema	107
4.3 Presentación de pipelines de integración y clasificación de datos	112
4.3.1 Pipelines de clasificación de información.....	112
4.3.2 Pipeline de clasificación de documentos PDF	112
4.3.3 Pipeline de clasificación de publicaciones de Mastodon.....	113
4.3.4 Integración de ambos pipelines en la arquitectura.....	115
4.4 Visualización de Resultados	116
4.4.1. Distribución de medicamentos asociados a la remisión.....	116
4.4.2 Intervenciones de actividad física y estilo de vida	117
4.4.3 Comorbilidades frecuentes en los pacientes analizados	118

4.4.4 Intervenciones quirúrgicas y su distribución	119
4.4.5 Datos de alimentación y dietas	120
4.4.6 Importancia de las visualizaciones para el análisis de correlaciones...	121
Capítulo 5. Conclusiones	123
5.1 Trabajo a futuro.....	124
Referencias	127

Índice de tablas

Tabla 2.1 Comparativa entre artículos relacionados	33
Tabla 3.1 Análisis de criterios de investigación	49
Tabla 3.2 Análisis comparativo de los trabajos analizados	85

Índice de figuras

Figura 3.1 PRISMA	47
Figura 3.2 Metodología de análisis.....	48
Figura 3.3 Arquitectura del Data Lakehouse	90
Figura 3.4 Diagrama relacional	91
Figura 3.5 Estructura de contenedores en Docker	93
Figura 3.6 Interfaz de Kafka leyendo un mensaje	94
Figura 3.7 Código Spark-Iceberg	96
Figura 3.8 Ejecución del código dentro de la consola	96
Figura 3.9 Interfaz de Minio S3	97
Figura 3.10 Interfaz de Trino	99
Figura 3.11 Consulta desde la consola de Trino	99
Figura 3.12 Interfaz gráfica de Apache Superset	101
Figura 3.13 Listado de modelos de Ollama en local.....	101
Figura 3.14 Kafka con el mensaje clasificado por Deepseek	102
Figura 4.1 Distribución de repositorios por dominio	103
Figura 4.2 Distribución de repositorios por dominio por año	104
Figura 4.3 Distribución de uso de tecnologías de integración de datos	105
Figura 4.4 Distribución de uso de tecnologías de procesamiento de datos	105
Figura 4.5 Distribución de uso de tecnologías de almacenamiento de datos.....	106
Figura 4.6 Distribución de productores de software de herramientas para gestionar datos	106
Figura 4.7 Distribución de uso de tecnologías de integración de datos	108
Figura 4.8 Interfaz de Apache Airflow para ejecución de DAG's	115
Figura 4.9 Muestreo de clasificación de PDF's mediante LLM local con ejecución en GPU	116
Figura 4.10 Consulta de medicamentos asociados positivamente a la remisión de diabetes tipo 2	117
Figura 4.11 Distribución de horas por semana por actividad física	118
Figura 4.12 Comorbilidades asociadas a pacientes con Diabetes tipo 2	119

Figura 4.13 Principales Intervenciones quirúrgicas en pacientes con diabetes tipo 2 asociadas positivamente con la remisión	120
Figura 4.14 Semanas por año de seguimiento de dietas específicas relacionadas positivamente con la remisión	121

Resumen

La diabetes tipo 2 es una enfermedad crónica de alta prevalencia en la población convirtiéndola en una de las mayores enfermedades a nivel mundial, requiere de cuidados y control, ya que sin ellos deriva en complicaciones de salud mayores. Por tal motivo se ha investigado y se encontró que la diabetes tipo 2 se lleva a un estado de remisión dependiendo de diversos factores los cuales son complejos de determinar correctamente, lo que sí se tiene claro es que el paciente llegará a determinados niveles de azúcar en sangre y lo ideal es mantenerlos para entrar en estado de remisión.

En consecuencia, el objetivo del presente proyecto fue desarrollar un *Data Lakehouse* que permita encontrar los factores para que un paciente alcance la remisión de la diabetes tipo 2 mediante el análisis de grandes cantidades de información multifuente y multiformato. Un *Data Lakehouse* es una tecnología reciente que combina las mejores características de los *Data Warehouse* con los *Data Lakes* permitiendo la ingesta y procesamiento de cantidades grandes de datos de las cuales se obtendrán los conocimientos deseados.

La construcción del *Data Lakehouse* se apoyó principalmente en la tecnología proporcionada por Apache como lo son NiFi, Kafka, Spark, Hudi, Iceberg, Airflow y Superset, adicionalmente estarán acompañadas de Trino y Postgres con el fin de tener variedad de herramientas en las capas que mejoran los procesos deseados, ya que es posible atender de diversas formas los requerimientos de información.

Abstract

Type 2 diabetes is a chronic disease with high prevalence in the population, which has made it one of the most significant diseases worldwide. It requires ongoing care and monitoring, since without them it can lead to major health complications. For this reason, research has been conducted and it has been found that type 2 diabetes can be brought into a state of remission depending on various factors, which are complex to determine accurately. What is clear is that the patient will reach certain blood sugar levels, and the ideal approach is to maintain them in order to enter a state of remission.

Consequently, the objective of this project was to develop a Data Lakehouse capable of identifying the factors that enable a patient to achieve remission of type 2 diabetes through the analysis of large volumes of multi-source and multi-format information. A Data Lakehouse is a recent technology that combines the best characteristics of Data Warehouses and Data Lakes, enabling the ingestion and processing of large amounts of data from which the desired knowledge can be obtained.

The development of the Data Lakehouse relied primarily on Apache technologies such as NiFi, Kafka, Spark, Hudi, Iceberg, Airflow, and Superset. In addition, Trino and Postgres were included in order to provide a variety of tools across layers that enhance the intended processes, since information requirements can be addressed in different ways.

Introducción

En la actualidad, la diabetes tipo 2 se ha convertido en una de las enfermedades crónicas más prevalentes a nivel mundial, requiriendo un manejo riguroso para prevenir complicaciones graves de salud. Sin embargo, investigaciones recientes han demostrado que la diabetes tipo 2 es considerada en estado de remisión si se alcanzan y mantienen ciertos niveles de glucosa en sangre, aunque los factores que influyen en este proceso son difíciles de determinar. Dado el impacto que la remisión tendría en la calidad de vida de los pacientes, es crucial identificar los factores y facilitar su control.

Este proyecto tuvo como objetivo desarrollar una plataforma que permita analizar grandes volúmenes de datos provenientes de diversas fuentes y formatos para identificar los factores clave que contribuyen a la remisión de la diabetes tipo 2. Para lograrlo, se desarrolló un *Data Lakehouse*, una tecnología moderna que combina las características de los *Data Warehouses* y los *Data Lakes*, optimizando la ingesta, almacenamiento y análisis de datos a gran escala.

El proyecto se estructura en tres capítulos, dentro del primer capítulo se encuentran los antecedentes del proyecto, incluyendo el marco teórico, el planteamiento del problema, la justificación y los objetivos. El segundo capítulo presenta el estado del arte y un análisis comparativo de trabajos relacionados con la remisión de la diabetes y el uso de tecnologías de procesamiento de datos. En el tercer capítulo, se detalla la solución empleando la metodología DataOps, se describe paso a paso la arquitectura y la forma en la que las capas funcionan demostrando como los datos pasan de un estado inicial a un estado procesado, la forma en la que el LLM es capaz de obtener la información relevante y cómo visualizar los resultados, adicionalmente el capítulo 4 es la culminación del procesamiento de toda la información depositada en el almacenamiento y como se obtienen los resultados. Por último, en el capítulo 5 se incluyen las conclusiones y el trabajo a futuro.

Capítulo 1. Antecedentes

Dentro del capítulo se presentan los principales términos relacionados con el trabajo de investigación, se especifica la situación tecnológica, económica y operativa de la empresa, se describe el planteamiento del problema, así como el objetivo general y objetivos específicos junto con la justificación.

1.1 Marco teórico

Dentro de esta sección se encuentran los principales términos relacionados con el tema de investigación que están definidos para la correcta comprensión del contexto.

1.1.1 Diabetes

La diabetes sacarina o diabetes *mellitus* (que aquí se denomina, para simplificar, «diabetes») es una enfermedad crónica que se presenta cuando el páncreas no secreta suficiente insulina o cuando el organismo no utiliza eficazmente la insulina que produce. La insulina es una hormona que regula la concentración de glucosa en la sangre, es decir, la glucemia. Un efecto común de la diabetes no controlada es la hiperglucemia (es decir, la glucemia elevada) que, con el tiempo daña gravemente muchos órganos y sistemas del organismo, sobre todo los nervios y los vasos sanguíneos [1].

En México, la Encuesta Nacional de Salud y Nutrición (Ensanut) 2022, señala que la prevalencia de prediabetes fue de 22.1% (equivalente a 17.6 millones de personas); la prevalencia de diabetes diagnosticada fue de 12.6% y la prevalencia de diabetes no diagnosticada fue de 5.8%, lo que resulta en una prevalencia de diabetes total de 18.3% (14.6 millones de personas), mientras que a nivel mundial 537 millones de personas padecían diabetes de acuerdo con las estadísticas de la Federación Internacional de Diabetes (IFD) publicadas en el año 2021 [2], [3].

Las complicaciones a largo plazo de la diabetes aparecen progresivamente, cuanto más tiempo se padece diabetes (y cuanto menos se controle la glucosa sanguínea), mayor será el riesgo de sufrir complicaciones. Con el tiempo, las complicaciones de la diabetes provocan discapacidad o incluso ponen en riesgo la vida [4], [5].

Algunas de estas complicaciones son:

- Enfermedades cardíacas y de los vasos sanguíneos (cardiovasculares)
- Daño a los nervios por diabetes (neuropatía diabética)
- Daño a los riñones por diabetes (nefropatía diabética)
- Daño a los ojos por diabetes (retinopatía diabética)
- Daños en el pie
- Afecciones de la piel y la boca
- Deterioro de la audición
- Alzheimer
- Depresión

A continuación, se enlistan las variantes de diabetes documentados por el Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales (NIDDK):

- Diabetes tipo 1

La diabetes tipo 1 (anteriormente conocida como insulín dependiente, juvenil o de inicio en la niñez) se caracteriza por una producción deficiente de insulina y requiere la administración diaria de insulina.

Los síntomas incluyen excreción excesiva de orina (poliuria), sed (polidipsia), hambre constante, pérdida de peso, cambios en la visión y fatiga [6].

- Diabetes tipo 2

La diabetes tipo 2 (antes llamada no insulín dependiente o de inicio en la edad adulta) es el resultado del uso ineficaz de la insulina por parte del cuerpo, es en gran parte el resultado del exceso de peso corporal y la inactividad física.

Los síntomas son similares a los de la diabetes tipo 1, pero a menudo son menos marcados, como resultado, es común que la enfermedad se diagnostique varios años después del inicio, después de que surjan complicaciones [6].

- Diabetes gestacional

La diabetes gestacional aparece durante el embarazo y se caracteriza por una hiperglucemia con valores que, pese a ser superiores a los normales, son inferiores a los establecidos para diagnosticar diabetes.

Quienes la presentan tienen más riesgo de sufrir complicaciones durante el embarazo y el parto. Además, tanto la madre como, posiblemente, sus hijos corren más riesgo de presentar diabetes de tipo 2 en el futuro [6].

1.1.2 Remisión de diabetes

La remisión se define como lograr una glucemia por debajo del rango diabético en ausencia de terapia farmacológica activa (medicamentos antihiper glucémicos, medicamentos inmunosupresores) o quirúrgica (procedimientos continuos como reemplazos repetidos de dispositivos endoluminales). Una remisión se caracteriza como parcial o completa [7], la remisión total implica niveles de azúcar en sangre en ayunas por debajo de 100 mg/dL y hemoglobina glicosilada (HbA1c) por debajo del 5.8% mientras que la remisión parcial se define por niveles de azúcar en sangre entre 100 y 125 mg/dL y/o HbA1c entre 5.8 y 6.4%. .

1.1.3 Medición de glucosa

Las pruebas de diabetes miden los niveles de glucosa en la sangre para controlar el nivel de glucosa en la sangre o ver si la persona tiene diabetes o está en riesgo de tenerla.

Hay varias formas de detectar, controlar y diagnosticar la diabetes. La mayoría de las pruebas miden los niveles de glucosa en la sangre.

Algunos de los diferentes tipos de pruebas de glucosa en sangre son:

- Prueba de glucosa en sangre
También conocida como prueba de glucemia en ayunas. Antes de la prueba, la persona ayunará (no comer ni beber nada) durante 8 horas. Esta prueba se realiza en la mañana y se usa a menudo como prueba de detección de la diabetes.
- Prueba de tolerancia a la glucosa oral
- Prueba de tolerancia de glucosa

- Prueba aleatoria de azúcar en sangre
- Hemoglobina glicosilada (HbA1c)

Mide la cantidad promedio de glucosa unida a hemoglobina en los últimos tres meses. La hemoglobina es la parte de los glóbulos rojos que transporta oxígeno de los pulmones al resto del cuerpo.

1.1.4 *Data Warehouse*

Los almacenes de datos, también llamados almacenes de datos empresariales (EDW), son una plataforma de datos empresariales que se utiliza para analizar y generar informes de datos estructurados y semiestructurados de varias fuentes de datos, como transacciones de puntos de venta, automatización del *marketing*, gestión de relaciones con clientes, entre otros.

Los almacenes de datos incluyen una base de datos analítica, así como componentes y procedimientos analíticos críticos. Estos admiten análisis *ad hoc* e informes personalizados, como flujos de procesamiento de datos, consultas y aplicaciones empresariales; permiten consolidar e integrar cantidades ingentes de datos actuales e historiales de datos en un solo lugar y están diseñados para ofrecer una visión de los datos a lo largo del tiempo. Gracias a estas funciones del almacén de datos, el almacenamiento de datos se ha convertido en un elemento básico de las analíticas empresariales para tomar decisiones empresariales fundamentadas [8].

1.1.5 *Data Lake*

Los *Data Lakes* constituyen una plataforma escalable y segura en la que las empresas hacen varias operaciones: ingerir toda clase de datos de cualquier sistema con independencia de su rapidez, incluso datos procedentes de sistemas *on-premise*, en la nube o de informática perimetral; almacenar cualquier tipo o volumen de datos con alta fidelidad; procesar datos en tiempo real o por lotes; y analizar datos usando cualquier tipo de lenguaje, como SQL, Python o R, u otras aplicaciones de datos o analíticas de terceros [9].

1.1.6 *Data Lakehouse*

Un *Data Lakehouse* es una arquitectura de datos moderna que crea una plataforma única mediante la combinación de los beneficios clave de los *Data Lakes* (grandes repositorios de datos sin procesar en su forma original) y los almacenes de datos (conjuntos organizados de datos estructurados). En particular, los *Data Lakehouses* permiten que las organizaciones usen almacenamiento de bajo costo para guardar grandes cantidades de datos sin procesar y proporcionar funciones de estructura y administración de datos.

Antes, los almacenes de datos en la nube y los *Data Lakes* tenían que implementarse como arquitecturas separadas y de silos a fin de evitar sobrecargar los sistemas subyacentes y crear contención para los mismos recursos. Sin embargo, el enfoque requería que los datos se cambiaran de forma periódica entre los dos sistemas separados cuando los datos de cualquiera de las arquitecturas se procesaban juntos, lo que generaba complejidad, mayores costos y problemas relacionados con la actualidad de los datos, la duplicación y la coherencia.

El objetivo de los *Data Lakehouses* es acabar con los silos de datos y ofrecer la flexibilidad, la escalabilidad y la agilidad necesarias para garantizar que los datos generen valor para la empresa, en lugar de ineficiencias [10].

1.1.7 *Big Data*

Se refiere a un conjunto de datos sumamente grande y complejo que se genera a partir de diversas fuentes, como dispositivos móviles, sensores, redes sociales, transacciones financieras y muchas más. El mayor desafío al trabajar con *Big Data* es encontrar maneras eficientes de recoger, almacenar, procesar y analizar esos datos para extraer información útil y conocimientos significativos.

El uso de tecnologías avanzadas, como la computación en la nube, la inteligencia artificial y el aprendizaje automático, son fundamentales para lidiar con el gran volumen y la complejidad de *Big Data* [11].

Características

Las definiciones de *Big Data* varían ligeramente, pero siempre se describirán en términos de volumen, velocidad y variedad. Estas características de *Big Data* suelen denominarse "las tres uves de *Big Data*" y Gartner las definió por primera vez en el 2001.

- Volumen

Como su nombre indica, la característica más común asociada a *Big Data* es su gran volumen, se describe como la enorme cantidad de datos que se recopilan y que se generan a partir de diversas fuentes y dispositivos de forma continua.

- Velocidad

La velocidad de *Big Data* hace referencia a la velocidad a la que se generan los datos. Hoy en día, los datos se suelen generar en tiempo real o casi en tiempo real y, por lo tanto, también se procesan, consultan y analizan al mismo ritmo para tener un impacto significativo.

- Variedad

Los datos son heterogéneos, es decir, proceden de muchas fuentes diferentes y están estructurados, no estructurados o semiestructurados. Los datos estructurados más tradicionales (como los datos de hojas de cálculo o bases de datos relacionales) se complementan ahora con texto, imágenes, audio y archivos de vídeo no estructurados, o con formatos semiestructurados como los datos de sensores que no se organizan en un esquema de datos fijo.

Además de las tres Vs originales, otras tres que a menudo se mencionan en relación con el aprovechamiento del potencial de *Big Data* son la veracidad, la variabilidad y el valor.

- Veracidad

Big Data resulta en algunas ocasiones desordenado, con mucho ruido y es propenso a errores, lo que dificulta el control de la calidad y la precisión de los datos. Los conjuntos de datos grandes resultan difíciles de manejar y confusos, mientras que los conjuntos más pequeños presentan una imagen incompleta. Cuanto mayor sea la veracidad de los datos, más fiable será.

- Variabilidad

El significado de los datos recogidos cambia constantemente, lo que provoca incoherencias a lo largo del tiempo, los cambios incluyen no solo los cambios de contexto y de interpretación, sino también en los métodos de recolección de datos basados en la información que las empresas quieren recoger y analizar.

- Valor

Es fundamental determinar el valor empresarial de los datos recolectados. Es necesario que los datos masivos sean datos adecuados y luego analizarlos de manera eficaz para extraer información valiosa que ayude a tomar decisiones fundamentadas.

1.1.8 Repositorio de datos

Un repositorio de datos es un lugar donde se almacena y accede conjuntos de datos. Normalmente, los conjuntos de datos en un repositorio están marcados con metadatos que proporcionan información de contexto esencial sobre los conjuntos de datos y permiten una búsqueda de datos eficiente. La arquitectura de un repositorio de datos es comparable a una biblioteca convencional y se encuentran varias comparaciones paralelas, como conjuntos de datos con copias impresas de publicaciones y metadatos con catálogos de publicaciones. Mientras que una biblioteca convencional necesita cierto espacio físico para almacenar las copias impresas de las publicaciones, un repositorio de datos requiere menos recursos físicos y organiza varios tipos de contenidos. En general, los repositorios de datos se clasifican en algunos tipos según los contenidos que organizan. Entre esos tipos, los más vistos son los datos heredados digitalizados, los datos digitales originales y los servicios de catálogo de datos, así como los híbridos de ellos [12].

1.1.9 Aprendizaje automático

El aprendizaje automático es un subconjunto de la inteligencia artificial que permite que un sistema aprenda y mejore de forma autónoma con técnicas como redes neuronales y aprendizaje profundo, sin necesidad de una programación explícita, a través del análisis de grandes cantidades de datos.

El aprendizaje automático permite que los sistemas informáticos se ajusten y mejoren continuamente a medida que acumulan más "experiencias". Por lo tanto, generalmente el rendimiento de estos sistemas puede mejorar si se proporcionan conjuntos de datos más grandes y variados para su procesamiento [13].

1.2 Situación tecnológica, económica y operativa de la empresa

El Instituto Tecnológico de Orizaba es una institución pública, que pertenece al Tecnológico Nacional de México. Se encuentra ubicado en Avenida Oriente 9 Núm. 852, Colonia Emiliano Zapata, C.P. 94320, Orizaba, Veracruz. Brinda educación superior, orientada al desarrollo de la ciencia y la tecnología. Actualmente, en el Instituto se imparten carreras a nivel licenciatura y estudios de posgrado. A nivel licenciatura, cuenta con once programas educativos. La División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Orizaba cuenta con una especialización, siete maestrías y dos doctorados, dentro de las siete maestrías se encuentra la Maestría en Sistemas Computacionales.

1.3 Planteamiento del problema

Desde 2011, la percepción de la diabetes tipo 2 cambió significativamente gracias a descubrimientos que demostraron que la reducción de la ingesta calórica y la consecuente pérdida de peso conducen a una disminución de la grasa intrahepática e intrapancreática, mejorando así la resistencia a la insulina [14]. Los nuevos hallazgos transformaron la visión de la diabetes tipo 2, pasando de considerarse una enfermedad crónica y progresiva a una condición potencialmente reversible. A partir de las nuevas consideraciones, se desarrollaron numerosos estudios y documentos de consenso para definir el concepto de remisión en la diabetes tipo 2. Aunque aún no se establecen criterios definitivos, se acepta comúnmente que mantener la hemoglobina glucosilada por debajo del 6.5% durante un período prolongado sin el uso de insulina o medicamentos hipoglucemiantes orales

(exceptuando posiblemente la metformina) se considera remisión de la diabetes [15].

Teniendo en cuenta la problemática descrita surge la necesidad de un análisis exhaustivo de los factores que permiten alcanzar la remisión en la diabetes tipo 2. Para lograrlo, se propuso el desarrollo de un *Data Lakehouse*, un repositorio de datos que combina las características de un *Data Warehouse* y un *Data Lake*. El *Data Lakehouse* no solo guarda datos en un sistema de almacenamiento de objetos, como lo hace un *Data Lake*, sino que también añade funcionalidades diseñadas para agilizar la gestión de datos, permitiendo una experiencia de ingeniería similar a la de un *Data Warehouse*. El *Data Lakehouse* incluye un soporte robusto de tablas y esquemas, así como características para gestionar actualizaciones y eliminaciones incrementales, además de mantener un historial de versiones de archivos y metadatos [16].

El objetivo del proyecto fue, por lo tanto, desarrollar un *Data Lakehouse* que permita analizar grandes volúmenes de datos relacionados con la remisión de la diabetes tipo 2. A través del análisis, se busca identificar los factores clave que contribuyen a alcanzar y mantener la remisión, proporcionando así una herramienta valiosa para avanzar en el tratamiento y la gestión de esta condición.

1.4 Objetivos general y específicos

1.4.1 Objetivo general

Desarrollar un *Data Lakehouse* para el análisis de datos masivos sobre la remisión en la diabetes tipo 2 para la identificación de los factores necesarios para lograrla.

1.4.2 Objetivos específicos

1. Analizar trabajos relacionados con el desarrollo de repositorios de datos y la remisión en la diabetes tipo 2, identificando áreas de oportunidad.
2. Identificar bases de datos científicas, bases de datos genómicas, datos abiertos, entre otros, relacionados con la diabetes tipo 2 enfocados en *Big Data* para su incorporación en el *Data Lakehouse*.

3. Seleccionar las tecnologías que se utilizarán para el desarrollo del *Data Lakehouse*.
4. Diseñar el Data Lakehouse que determine los factores necesarios para la remisión en la diabetes tipo 2, basado en la metodología elegida.
5. Implementar el *Data Lakehouse* de acuerdo con el diseño establecido anteriormente.
6. Monitorizar el funcionamiento del *Data Lakehouse* para la identificación y solución de problemas.

1.5 Justificación

En México, la prevalencia de esta enfermedad es alarmante, afectando a millones de personas y siendo una de las principales causas de morbilidad y mortalidad. A pesar de los avances en el tratamiento, la remisión de la diabetes tipo 2 sigue siendo un objetivo difícil de alcanzar para la mayoría de los pacientes. La identificación de los factores que permiten lograr la remisión transformará la manera en que se trata la enfermedad, ofreciendo a los pacientes una esperanza tangible de recuperación. Por parte de las tecnologías, el uso de un *Data Lakehouse* es crucial, ya que combina la capacidad de almacenamiento masivo de datos de los *Data Lakes* con la estructura y facilidad de consulta de los *Data Warehouses* por lo que permite manejar grandes volúmenes de datos provenientes de diversas fuentes, como bases de datos científicas, datos genómicos y registros médicos, y analizarlos de manera eficiente. La flexibilidad y escalabilidad del *Data Lakehouse* lo convierten en la herramienta ideal para investigar la remisión en la diabetes tipo 2, permitiendo explorar patrones y correlaciones en conjuntos de datos que de otro modo serían difíciles de manejar.

Además, el proyecto no solo tiene implicaciones para la investigación científica, sino también para la toma de decisiones clínicas. Los resultados obtenidos ofrecerán una guía para los profesionales de la salud en la personalización de tratamientos y estrategias de intervención, optimizando los recursos y mejorando los resultados para los pacientes.

Capítulo 2. Estado de la práctica

En este capítulo se analizaron 15 artículos, los cuales se estudiaron detalladamente para extraer los aspectos más relevantes considerando el tema y lo que aportan para la presente investigación, posteriormente se comparan en una tabla y se ofrecen conclusiones al respecto.

2.1 Trabajos relacionados

Siddiqi et al. [17] abordaron en su artículo la necesidad de agilizar los ensayos clínicos para productos médicos, especialmente en situaciones críticas como pandemias, como lo ejemplifica el COVID-19. La principal contribución del artículo es un marco de trabajo integral diseñado para facilitar la investigación clínica y mejorar la eficiencia en la evaluación de productos médicos.

La demora en los procesos de investigación clínica resulta en la pérdida de vidas humanas durante emergencias sanitarias por ello la urgencia de acelerar los procesos se destaca como crucial para salvar vidas.

Por lo anterior expuesto, el marco de trabajo propuesto incorporó capacidades avanzadas de análisis de datos, seguimiento de la genealogía (*lineage tracking*) y gobernanza de datos. No solo permitió análisis basados en evidencia, sino que también facilitó la exploración visual de la semántica y los vínculos entre datos, mejorando la toma de decisiones informadas.

Consecuentemente se emplearon tecnologías como Apache Spark para el procesamiento distribuido de datos, Kafka y Akka Streams para el *streaming* de datos, y Hudi para el almacenamiento de datos en formato de serie temporal.

Por último, los resultados obtenidos incluyeron casos de uso que demostraron la aplicación exitosa del marco en diversos escenarios, tanto en organizaciones privadas como públicas y mixtas.

Continuando con el artículo de Wang et al. [18] el cual es la propuesta de un sistema de *Lakehouse* integrado basado en Hudi para mejorar la eficiencia del procesamiento de datos y satisfacer las crecientes demandas de datos analíticos en tiempo real en el escenario de redes eléctricas. El sistema combinó procesamiento por lotes y en *streaming*, mejorando la capacidad de respuesta y la estabilidad de las operaciones del sistema de producción.

En cuanto a la tecnología se incluyó Java, Spark, Flink, Apache Hudi para el almacenamiento, y sistemas de almacenamiento en objetos S3. Kubernetes se utilizó para el aislamiento de recursos y la gestión de permisos, con herramientas internas optimizadas para la ingesta y transformación de datos, y *frameworks* como Doris para computación en *Big Data*.

Respecto a los resultados obtenidos se destacaron mejoras significativas en la eficiencia de la ingesta de datos en tiempo real, reduciendo la latencia y los recursos necesarios. La implementación de una arquitectura que permite la escalabilidad independiente de los recursos de almacenamiento y computación mejoró el rendimiento y la eficiencia. Se desarrolló un modelo de datos que soporta la actualización y consulta en tiempo real. Además, se optimizó la lógica de escritura y fusión de datos para manejar grandes volúmenes de datos con operaciones de actualización frecuentes, utilizando modos de copia en tiempo real.

Por su parte, Begoli et al. [19] abordaron los desafíos en la gestión y análisis de grandes volúmenes de datos de salud protegidos, provenientes de múltiples fuentes estructuradas y no estructuradas. Su principal contribución fue la propuesta de una arquitectura de *Lakehouse* que combinó las ventajas de los *Data Lakes* y *Data Warehouses*, la arquitectura facilitó el almacenamiento, procesamiento y análisis eficiente de datos de salud protegidos, garantizando su integridad, accesibilidad y cumplimiento con los principios FAIR (*Findable, Accessible, Interoperable, Reusable*, Encontrable, Accesible, Interoperable, Reutilizable). Para lograr los objetivos, se utilizaron diversas tecnologías clave. Respecto al lenguaje de programación se utilizó Python, junto con PySpark para el procesamiento distribuido de datos, adicionalmente para el almacenamiento eficiente y de alto rendimiento, se

empleó Apache Parquet, se implementaron herramientas como Apache Airflow, Marquez y Great Expectations para la construcción de *pipelines* de datos automatizados y robustos.

Entre los resultados obtenidos destacaron mejoras significativas en el rendimiento y una reducción sustancial en los requisitos de almacenamiento, gracias al uso del formato Parquet en comparación con datos brutos como CSV (*Comma Separated Values*, Valores Separados por Comas). La arquitectura *Lakehouse* se adoptó exitosamente por varios proyectos, con un creciente número de usuarios beneficiándose de su implementación.

Adicionalmente, Li et al. [20] propusieron una solución de diagnóstico basada en inteligencia artificial para pequeñas y medianas instalaciones de salud, abordando la escasez mundial de recursos médicos exacerbada por la creciente población y la pandemia de COVID-19. Utilizando técnicas como *Big Data*, inteligencia artificial y computación en la nube, se buscó proporcionar una transformación digital que aumente la productividad. El estudio diseñó y evaluó una solución de *Lakehouse*, la arquitectura ofreció almacenamiento masivo de bajo costo y alta capacidad de consulta y análisis. Además, se destacó que la creación de recursos relevantes para la solución solo tomó alrededor de 2 minutos utilizando la técnica de IaC (*Infrastructure as Code*, Infraestructura como Código), en contraste con los días o semanas que requería el despliegue en un centro de datos tradicional.

Se utilizó el formato Parquet para mejorar el rendimiento y reducir costos, en comparación con el procesamiento y análisis de datos en archivos CSV, lo cual demostró una significativa reducción en el espacio de almacenamiento, con el formato Parquet ocupando solo el 30% del espacio que utilizaría el formato CSV. La precisión de los modelos de predicción se optimizó utilizando el método de *ensemble bagging*, reduciendo el riesgo de errores de juicio causados por un único modelo.

Como resultado la investigación verificó que, a pesar de la limitada diferencia en precisión entre diversos algoritmos a causa de conjuntos de datos pequeños, el método de *ensemble bagging* mejoró la precisión general.

Por otro lado, en un ámbito diferente al de la salud el artículo de Park et al. [21] presentó un enfoque integral para la gestión y análisis de datos relacionados con embarcaciones marítimas mediante la implementación de un *Data Lakehouse*. Se evaluó el rendimiento del *Data Lakehouse* comparando los tiempos de procesamiento de consultas entre PostgreSQL, una base de datos relacional, e Impala, el motor de consulta del *Data Lake*. Los datos utilizados provienen del *marine Data Lake* y se llevaron a cabo mediciones de tiempo para consultas simples de conteo de filas. En todos los casos evaluados, Impala demostró ser significativamente más rápido que PostgreSQL, con mejoras de rendimiento que oscilaron entre el 88.14% y el 97.38%, dependiendo del tamaño y la complejidad de los conjuntos de datos.

Además, se realizó una evaluación del rendimiento del análisis marino utilizando la capa de *Vessel AI* para predecir la actividad pesquera y clasificar el tipo de embarcación. Se emplearon modelos predictivos como árboles de decisión, *Random Forest*, LSTM (*Long Short-Term Memory*, Memoria a largo plazo) y HMM (*Hidden Markov Model*, Modelo Oculto de Markov), utilizando datos preprocesados para el entrenamiento y prueba.

Los resultados indicaron que los modelos de *Random Forest* y LSTM superaron en precisión a los modelos de árboles de decisión y HMM tanto en la clasificación del tipo de embarcación como en la predicción de la actividad pesquera.

Así mismo, Zhang et al. [22] describieron en su artículo la forma en que se aplicaron consultas de datos multimodales para asistir en la investigación científica sobre el tratamiento de la sepsis. Se utilizaron instrumentos personalizados para recolectar de manera automatizada imágenes de piel de pacientes ingresados en la UCI (Unidades de Cuidados Intensivos), sin intervención humana, las imágenes se emplearon para mejorar la efectividad del tratamiento de la sepsis en investigación clínica.

Durante el estudio, se recolectaron datos generados a partir del tratamiento de pacientes, incluyendo datos de series temporales posteriores al ingreso hospitalario. El conjunto de datos consistió en 63 GB de datos estructurados y 67.4 TB de datos

no estructurados. Se incluyeron signos vitales como frecuencia cardíaca, presión arterial media, temperatura; indicadores de pruebas como troponina, recuento de linfocitos, potasio en sangre; y comportamientos médicos como noradrenalina, fenilefrina, ingesta de líquidos, puntuaciones SOFA (*Sequential Organ Failure Assessment*, valoración de la disfunción orgánica) y volumen de orina. Además, se recolectaron automáticamente imágenes de las extremidades de los pacientes durante el proceso de tratamiento.

Se presentó un marco para datos médicos multimodales basado en *Data Lake*, implementando la fusión de datos heterogéneos, gestión de índices y exploración de datos híbridos. Basado en la capacidad de cálculo y el plan de ejecución optimizado de SparkSQL, se implementó la fusión de datos y la exploración utilizando modelos de aprendizaje profundo. Se propusieron estrategias de gestión de índices y almacenamiento dinámico para una gestión eficiente de características de alta dimensión que permitieron explorar datos multimodales con interfaces en SQL y GraphX.

Adicionalmente, en el artículo de Tripathi et al. [23] detallaron la implementación exitosa de la arquitectura MINDS (*Multimodal Integration of Oncology Data System*, Integración Multimodal del Sistema de Datos Oncológicos) para la gestión integrada de datos oncológicos multimodales. MINDS abordó el desafío fundamental de consolidar más de 41,000 perfiles de casos de cáncer de acceso abierto en un extracto estructurado de 25.85 MB, comprimido desde los 3.78 PB originales de los repositorios fuente, dicha reducción permitió realizar análisis complejos de SQL en un solo nodo, facilitando una gestión eficiente de los datos sin comprometer la capacidad analítica.

Además de la consolidación y el almacenamiento eficiente, MINDS proporcionó herramientas robustas para la construcción de cohortes personalizadas mediante consultas SQL. Los tiempos de respuesta rápidos, incluso para consultas multidimensionales complejas, aseguraron una iteración ágil durante el desarrollo de modelos.

Para mejorar la reproducibilidad y la trazabilidad, se implementaron identificadores de versión únicos para cada conjunto de datos de cohorte, asegurando que cualquier cambio se reflejara claramente en los modelos derivados, capacidad que fue fundamental para la confiabilidad de los resultados en entornos de aprendizaje automático y para apoyar la investigación reproducible.

Finalmente, MINDS ofreció capacidades avanzadas de análisis y visualización a través de paneles de control integrados como Amazon QuickSight, permitiendo la exploración detallada de características de cohortes, correlaciones entre datos clínicos y resultados de modelos.

De forma semejante en el aporte de Bellandi et al. [24] detallaron la implementación de flujos de trabajo automatizados en el proyecto SMART BEAR, centrados en la curación continua de datos y la aproximación del tamaño de muestra para aplicaciones de aprendizaje automático en EHR (*Electronic Health Record*, Registros Electrónicos de Salud), los flujos de trabajo se diseñaron para mejorar la calidad del conjunto de datos y facilitar la creación de modelos para pronósticos precisos en entornos de recopilación de datos médicos continuos.

Para alcanzar los objetivos, se utilizaron diversas tecnologías y herramientas avanzadas. Entre ellas se destacan Apache Airflow, empleado para la orquestación de flujos de trabajo; la API de BDA (*Big Data Analytics*, Análisis de *Big Data*) para análisis de *Big Data*; Delta Lake y Trino para almacenamiento y consulta eficiente de datos; Apache Zeppelin para exploración y visualización de datos; Python y Spark para manipulación y procesamiento de datos a gran escala; tsfresh para la extracción de características de series temporales; y XGBoost junto con H2O AutoML para la construcción de modelos de aprendizaje automático.

Los resultados obtenidos reflejaron una implementación exitosa de los flujos de trabajo automatizados. Se logró mejorar significativamente la calidad del conjunto de datos mediante técnicas avanzadas de imputación y homogeneización, lo cual fue crucial para garantizar la precisión de los modelos de pronóstico desarrollados. La integración de los flujos de trabajo permitió manejar eficazmente los desafíos

inherentes a la curación de datos en entornos médicos, asegurando que los modelos resultantes fueran robustos y reproducibles.

En contraste, Schneider et al. [25] llevaron a cabo un estudio de caso en un entorno industrial real donde se gestionan y explotan analíticamente datos de manufactura, un departamento de análisis dedicado probó y aprovechó un enfoque de *Lakehouse* durante varios meses en un entorno productivo con volúmenes de datos elevados y diversos tipos de cargas de trabajo analítico, el artículo presentó los resultados del análisis dentro del caso, centrándose en el entorno industrial y la arquitectura de *Lakehouse* utilizada.

En términos de infraestructura y almacenamiento de datos, el *Lakehouse* se basó en Azure Data Lake Storage Gen2 como sistema de almacenamiento y utilizó *Databricks* para el procesamiento de *batch* y *streaming*. El marco Delta Lake persiste los datos en formato columnar Apache Parquet y permite representar colecciones de archivos de datos como tablas autocontenidas con propiedades relacionales en el nivel lógico. Además, se utiliza Confluent Kafka como concentrador de eventos para el almacenamiento temporal de datos en *streaming* antes de la ingestión en el *Lakehouse*.

En cuanto a la modelación de datos, se llevó a cabo de manera informal, adaptando modelos según la estructura de los datos proporcionados y los requisitos de los casos de uso analíticos. La organización de datos sigue un modelo de zonas que separa los datos según su granularidad, calidad y especificidad de aplicación dentro del *Lakehouse*, demostrando ser adecuado para este entorno industrial específico.

Por otra parte, Vox et al. [26] investigaron la optimización de esquemas relacionales y estrategias de almacenamiento de metadatos para datos de sensores de vehículos en bases de datos, con el objetivo de determinar el esquema más eficaz, se desarrollaron tres esquemas diferentes y se exploraron dos estrategias de almacenamiento de metadatos.

Los resultados destacaron que Schema-2 proporcionó una mejora significativa en el rendimiento de consultas no agregadas y, en menor medida, en la recuperación

de datos agregados, en particular, la plataforma Druid, al utilizar Schema-2, demostró ser el sistema más rápido para consultas de agregación de baja selectividad. Además, se observó que la normalización generalmente mejoraba la velocidad de recuperación de datos, aunque algunas plataformas se beneficiaron de la estrategia de desnormalización dependiendo del tipo de consulta específica. El estudio concluyó que la optimización de esquemas relacionales juega un papel crucial en mejorar el rendimiento en la gestión de datos de sensores de vehículos. En contextos de consultas de alta selectividad y tareas de agregación, Schema-2 se destacó como la opción óptima., los hallazgos sugirieron que la partición basada en marcas de tiempo, característica de Schema-2, es útil para su uso en investigaciones futuras con otros conjuntos de datos similares.

Por su parte en el campo médico, el aumento rápido de dispositivos médicos propició un volumen sustancial de datos, Zhao et al. [27] propusieron una estrategia basada en la tecnología de *Data Lake*, que ofrece un enfoque robusto para almacenar, gestionar e integrar datos médicos multimodales. La plataforma de *Data Lake* permite la consolidación de datos estructurados, semi-estructurados y no estructurados desde diversas fuentes en un modelo de datos unificado facilitando una exploración de datos eficiente que permite a los profesionales médicos y a los investigadores localizar rápidamente conjuntos de datos específicos para análisis detallados.

Los resultados experimentales destacaron que la tecnología de *Data Lake* supera claramente a las bases de datos relacionales tradicionales en términos de requisitos de almacenamiento y tiempo de procesamiento de consultas, al aprovechar las capacidades de almacenamiento distribuido y el procesamiento paralelo de datos, la plataforma de *Data Lake* no solo redujo el espacio necesario para el almacenamiento, sino que también aceleró significativamente el tiempo necesario para realizar consultas complejas sobre grandes volúmenes de datos médicos.

Además, se presentó una optimización de consultas federadas para mejorar aún más el rendimiento de las consultas que involucran datos multimodales, utilizando *frameworks* de código abierto como Apache Hudi y Apache Spark, se demostró

cómo las tecnologías se implementan para manejar eficientemente consultas complejas y operaciones de fusión de datos en entornos médicos.

De modo similar, Harby et al. [28] detallaron la implementación de un sistema *Triage-Bot* médico que aprovecha MongoDB para almacenar datos en tiempo real, facilitando un acceso rápido y eficiente a información crítica como datos de video, audio y texto generados durante interacciones con pacientes, Delta Lakehouse complementa este sistema al gestionar cargas analíticas complejas y garantizar la integridad de datos estructurados, especialmente en entornos médicos donde la precisión y la velocidad en el acceso a la información son cruciales.

Para simular entornos clínicos, se utilizó el conjunto de datos MIMIC-III (*Medical Information Mart for Intensive Care III*, Base de Datos de Información Médica para Cuidados Intensivos III), que abarca más de 40,000 casos de pacientes críticos admitidos en el *Beth Israel Deaconess Medical Center*. Este conjunto de datos proporciona una amplia gama de datos clínicos detallados, incluyendo historiales médicos, signos vitales, medicamentos y notas clínicas, esenciales para la toma de decisiones médicas informadas.

Se evaluaron la escalabilidad y el rendimiento del sistema mediante la ejecución de 12 consultas predefinidas, divididas en consultas OLTP (*Online Transaction Processing*, Procesamiento de Transacciones en Línea) y OLAP (*Online Analytical Processing*, Proceso Analítico en Línea), prueba que demostró la capacidad del *Triage-Bot* para manejar eficientemente transacciones en tiempo real y consultas analíticas complejas, proporcionando conocimientos valiosos para la gestión hospitalaria y la planificación de recursos.

En términos de funcionalidad, el *Triage-Bot* se destacó por su capacidad para reducir la congestión en los departamentos de emergencia hospitalaria al clasificar la gravedad de condiciones médicas basadas en datos EMR (*Electronic Medical Records*, Historia Clínica Electrónica) y proporcionar recomendaciones basadas en datos para la atención médica, lo que representó un avance significativo hacia una atención médica más eficiente y personalizada que permitió a los profesionales de la salud tomar decisiones informadas de manera oportuna.

Adicionalmente, Ren et al. [29] presentaron una plataforma híbrida de inferencia causal médica basada en un *Data Lake*, diseñada para gestionar y fusionar datos médicos heterogéneos y multimodales, y realizar inferencias causales con la participación de médicos, la plataforma abordó la necesidad de un bucle cerrado unificado para la inferencia causal en el dominio médico, integrando tanto métodos basados en datos como en conocimientos previos. Se destacó la interacción con los médicos durante el análisis exploratorio de datos, lo que mejora la calidad y precisión de las inferencias.

La arquitectura de la plataforma se dividió en tres partes principales: 1) Fusión de datos heterogéneos, 2) Análisis exploratorio de datos con la intervención de médicos, y 3) Inferencia causal, utilizó el *Data Lake* para almacenar y unificar datos estructurados y no estructurados, transformándolos en un formato coherente para su análisis. Durante el análisis exploratorio, los médicos contribuyen con su conocimiento para refinar la selección de entidades y mejorar la calidad de la tabla fusionada. La inferencia causal se llevó a cabo en cuatro pasos: 1) Selección de entidades, 2) Determinación de causas y efectos, 3) Creación de restricciones y 4) Selección de métodos de inferencia y refutación.

La implementación de la plataforma se basó en Delta Lake, y se probó con datos de registros médicos electrónicos de pacientes con osteoartritis de rodilla. El conjunto de datos, compuesto por 6000 pacientes y 16000 registros médicos, se utilizó para diversas investigaciones estadísticas, demostrando la efectividad de la plataforma en la mejora de la precisión y fiabilidad de las inferencias causales en el contexto médico.

Además, Xiao et al. [30] presentaron una plataforma llamada MHDML (*Multi-source Heterogeneous Data of Medical Lakehouse*, Datos Heterogéneos de Múltiples Fuentes de un *Lakehouse* Médico), que se diseñó para integrar y gestionar datos médicos multimodales y heterogéneos de múltiples fuentes de manera eficiente. El MHDML utilizó software de código abierto para convertir datos heterogéneos en un

almacenamiento unificado y empleó APIs Restful personalizadas para facilitar la exploración de datos médicos por parte del personal de salud.

La plataforma se probó con datos de sepsis y osteoartritis de rodilla, demostrando su capacidad para integrar y gestionar datos médicos multimodales de manera efectiva. El MHDML mejoró la trazabilidad de los datos y facilitó la toma de decisiones médicas rápidas mediante la fusión de datos y la exploración visual.

Por último, en [31] se abordó un problema significativo en el manejo de datos médicos heterogéneos de múltiples fuentes, donde la falta de un marco eficiente para la extracción, limpieza e imputación de datos limita la precisión y utilidad de los análisis realizados sobre dichos datos, los conjuntos de datos médicos a menudo contienen valores faltantes y presentan una estructura compleja gracias a su naturaleza multimodal y la variabilidad en la calidad de los datos obtenidos de distintas fuentes, problemas que afectan negativamente la capacidad de los investigadores y profesionales médicos para realizar análisis precisos, como la predicción de resultados clínicos o el desarrollo de modelos de tratamiento efectivos.

Por lo anterior expuesto, Sheng et al. [31] propusieron un marco de mejora de datos heterogéneos de múltiples fuentes que incluye un proceso completo de extracción, limpieza e imputación de datos, el enfoque clave de su solución radica en la imputación, donde utilizaron el modelo SAITS (*Self-Attention-based Imputation for Time Series*, Imputación Basada en la Atención Propia para Series Temporales), basado en la estructura *Transformer*.

En los experimentos, demostraron que el entrenamiento con secuencias largas de valores enmascarados artificialmente mejoró notablemente la precisión de la imputación de datos faltantes. En la investigación se evaluó la efectividad del sistema en tareas de agrupamiento y clasificación. En el análisis de agrupamiento, aplicaron el algoritmo *TimeSeriesKMeans* en los conjuntos de datos eICU (*Electronic Intensive Care Unit*, Unidad de Cuidados Intensivos Electrónica) y MIMIC-IV, observando una mejora en el rendimiento del agrupamiento reflejada en menores valores de SSE (*Sum of Squared Errors*, Suma de Errores al Cuadrado).

En cuanto a la clasificación, emplearon técnicas como XGBoost (*Extreme Gradient Boosting*, Aumento de Gradiente Extremo), LDA (*Linear Discriminant Analysis*, Análisis Discriminante Lineal) y SVC (*Support Vector Classification*, Clasificación de Vectores de Soporte) para predecir la supervivencia de pacientes en la UCI, logrando una mejora en la puntuación F1, lo que sugiere una mejor calidad de datos y mayor precisión en las predicciones.

Finalmente, el estudio también exploró la predicción de estrategias utilizando aprendizaje por refuerzo fuera de política con el modelo BCQ (*Batch-Constrained deep Q-learning*, Q-learning Profundo Restringido por Lotes), el cual aprendió las políticas de tratamiento de los médicos para pacientes en la UCI. La mejora en la secuencia de estados permitió al modelo reflejar de manera más precisa la relación entre los valores Q y las tasas de mortalidad, indicando que los datos mejorados facilitaban la selección de acciones que maximizaron los valores Q y, por ende, minimizaron la mortalidad.

2.2 Análisis comparativo

La Tabla 2.1 presenta un análisis comparativo sobre los artículos relacionados con este proyecto, así como una breve descripción acerca de la problemática, contribución, tecnologías utilizadas, resultados obtenidos y el estado de la investigación.

Tabla 2.1 Comparativa entre artículos relacionados

Artículo	Problema	Contribución	Tecnologías	Resultados	Estado
Siddiqi et al. [17]	Demora en ensayos clínicos durante pandemias.	Marco de trabajo para agilizar ensayos clínicos y mejorar la gestión de datos médicos.	Apache Spark Kafka Hudi	Casos de uso exitosos, mejora en eficiencia y transparencia de datos.	Propuesta y arquitectura
Wang et al. [18]	Sincronización redundante, procesamiento no en tiempo real, lentitud en consultas, falta de integración de datos cuantitativos.	Propuesta de sistema integrado de <i>Lakehouse</i> para el escenario de redes eléctricas basado en Hudi, enfocado en mejorar la eficiencia y puntualidad de los datos analíticos.	Java Spark Flink Apache Hudi S3 Kubernetes Doris	Mejora en la eficiencia de procesamiento e implementación del modelo de datos en tiempo real.	Propuesta de arquitectura
Begoli et al. [19]	Problemas de acceso a datos redundantes y no integrados, lentitud en consultas, falta de	Propuesta de un sistema de <i>Lakehouse</i> para datos protegidos en salud, enfocado en seguridad, acceso controlado y	Parquet SQL W3C PROV-O Apache Airflow Marquez Great Expectations	Reducción de espacio de almacenamiento e incremento en la adopción de usuarios.	Finalizado

Artículo	Problema	Contribución	Tecnologías	Resultados	Estado
	sincronización en tiempo real.	adherencia a principios FAIR.			
Li et al. [20]	Escasez de recursos médicos, agudizada por la pandemia de COVID-19.	Propuesta de una solución de diagnóstico basada en IA para pequeñas y medianas instalaciones de salud, utilizando un sistema de <i>Lakehouse</i> de datos en la nube para mejorar la eficiencia y la precisión diagnóstica.	Big Data IA Computación en la nube Amazon S3 Apache Parquet Delta Lake laC	Creación de recursos en 2 minutos con laC, mejora en el rendimiento de procesamiento de datos, reducción significativa de costos de almacenamiento, optimización de precisión mediante <i>ensemble bagging</i> , implementación en un entorno de producción.	Propuesta y arquitectura

Artículo	Problema	Contribución	Tecnologías	Resultados	Estado
Park et al. [21]	Gestión y análisis eficiente de grandes volúmenes de datos marítimos heterogéneos.	Diseño e implementación de una arquitectura de <i>Data Lakehouse</i> para datos marítimos.	Hadoop Impala Kudu TensorFlow Keras Apache Spark Apache TEZ	Impala mostró un rendimiento significativamente superior a PostgreSQL en el procesamiento de consultas. Los modelos de inteligencia artificial como <i>Random Forest</i> y LSTM demostraron alta precisión en la predicción de actividades pesqueras y clasificación de	Finalizado

Artículo	Problema	Contribución	Tecnologías	Resultados	Estado
				tipos de embarcaciones.	
Zhang et al. [22]	Manejo eficiente de datos médicos multi-modales.	Desarrollo de un marco de trabajo basado en un lago de datos para fusión de datos multi-modales, gestión de índices dinámicos y exploración híbrida de datos.	SparkSQL Técnicas de aprendizaje profundo para la extracción de características GraphX	Mejora del 60% al 70% en eficiencia de exploración de datos comparado con Oracle basada en LOB (<i>Large Objects</i> , Objetos grandes).	Propuesta presentada
Tripathi et al. [23]	Integración y gestión de grandes volúmenes de datos oncológicos desde diversas fuentes.	Desarrollo de MINDS, una arquitectura innovadora para la gestión integrada de datos multimodales en oncología.	AWS Redshift Amazon QuickSight Python Pandas SQL APIs RESTful.	Optimización de almacenamiento, reducción del tamaño de datos. Construcción flexible de cohortes mediante consultas SQL, integrando datos estructurados y no	Finalizado

Artículo	Problema	Contribución	Tecnologías	Resultados	Estado
				estructurados para análisis multimodal.	
Bellandi et al. [24]	Automatización de la curación de datos y aproximación del tamaño de muestra en el proyecto SMART BEAR.	Desarrollo de flujos de trabajo automáticos para mejorar la calidad del conjunto de datos y aproximar el tamaño de muestra necesario para modelos de aprendizaje automático.	SMART BEAR BDA Engine Algoritmos de imputación de datos Herramientas de AutoML	Mejora en la precisión y robustez de los modelos de aprendizaje automático, validación continua de modelos y <i>datasets</i> , mejora en la calidad de datos para análisis de registros de salud electrónicos (EHR).	Finalizado

Artículo	Problema	Contribución	Tecnologías	Resultados	Estado
Schneider et al. [25]	Gestión y explotación analítica de datos de manufactura.	Implementación de un <i>Lakehouse</i> en un entorno industrial para gestionar y analizar datos de manufactura.	Azure Data Lake Storage Gen2 Databricks Delta Lake Confluent Kafka	Implementación exitosa en entorno productivo con volúmenes de datos elevados y diversos tipos de cargas de trabajo analítico.	Propuesta y arquitectura
Vox et al. [26]	Optimización del esquema relacional para almacenamiento de datos de sensores de vehículos.	Evaluación de tres esquemas y dos estrategias de almacenamiento de metadatos para mejorar el rendimiento de consultas de series temporales y agregaciones en bases de datos.	Druid Hive Impala InfluxDB Spark TsDB	Schema-2 mejoró notablemente la velocidad de consultas agregadas y no agregadas frente a Schema-1. TsDB con Schema-2 fue excelente en consultas no agregadas de alta	Finalizado

Artículo	Problema	Contribución	Tecnologías	Resultados	Estado
				selectividad, Spark con Schema-2 y Schema-3 con TsDB fueron efectivos en consultas de baja selectividad. Druid con Schema-2 se destacó en tareas de agregación.	
Zhao et al. [27]	Manejo ineficiente de grandes volúmenes de datos médicos multimodales.	Propuesta de estrategia de fusión de datos multimodales basada en tecnología de <i>Data Lake</i> .	Apache Hudi Apache Spark PostgreSQL	Mejora significativa en el rendimiento de consultas federadas y optimización del almacenamiento y procesamiento.	Finalizado

Artículo	Problema	Contribución	Tecnologías	Resultados	Estado
Harby et al. [28]	Implementación de un sistema <i>Triage-Bot</i> médico para mejorar la gestión de datos y capacidades analíticas en entornos de salud.	Desarrollo de una arquitectura funcional para soportar comunicaciones de voz, video y chat en servicios de triaje médico.	MongoDB Delta Lakehouse Next.js React.js Chakra UI Apache Spark MIMIC-III	Optimización de la gestión de datos, reducción de tiempos de espera en emergencias, clasificación de severidad basada en datos EMR.	Finalizado
Ren et al. [29]	La falta de una plataforma unificada de inferencia causal médica basada en datos.	Propuesta de una plataforma híbrida que integra datos y conocimiento previo, permitiendo la interacción con médicos durante el análisis exploratorio de datos.	Delta Lake	Mejora en la precisión y fiabilidad de la investigación médica, demostrada en la aplicación a la osteoartritis de rodilla.	Finalizado
Xiao et al. [30]	Ineficiencia en la integración y gestión de datos médicos	Propuesta del Medical <i>Lakehouse</i> para integrar datos médicos	Delta Lake HDFS Sqoop	Mejora en la gestión de datos médicos y en la	Finalizado

Artículo	Problema	Contribución	Tecnologías	Resultados	Estado
	heterogéneos y multimodales.	heterogéneos de múltiples fuentes y gestionar datos multimodales eficientemente.	Flume Kafka HBase Hive Druid Spark Zookeeper	exploración de datos por parte del personal de salud.	
Sheng et al. [31]	Los datos médicos heterogéneos de múltiples fuentes presentan valores faltantes y complejidad estructural, lo que dificulta el análisis preciso y la predicción clínica.	Marco de mejora de datos que utiliza el modelo SAITS para la imputación de datos faltantes, mejorando la calidad de los análisis clínicos y la predicción de resultados.	SAITS Transformadores TimeSeries KMeans XGBoost LDA SVC BCQ	Mejora en la imputación de datos, incrementó la precisión en predicción de estrategias, reflejado en una mejor calidad de datos.	Propuesta presentada

En los últimos años, ocurrió un incremento en el uso de *Data Lakehouses* como solución a los problemas que presentaban tanto los *Data Warehouses* como los *Data Lakes*, tecnologías base que sirvieron para establecer el concepto de *Data Lakehouse* y sus principios fundamentales, ya que combinan las mejores características de ambos sistemas por lo que permite un tratamiento efectivo de datos a gran escala sin sacrificar la rapidez en su uso.

En el ámbito médico, la implementación de *Data Lakehouses* ganó terreno gracias a la gran cantidad de información que generan los hospitales y las entidades que censan datos médicos, ya que la necesidad de gestionar grandes volúmenes de datos se identifica como la principal relación entre la mayoría de los artículos revisados. Aunque no todos los artículos analizados se enfocan exclusivamente en el sector médico, muchos de ellos [18], [21], [25], [26], [27] aportan conocimientos valiosos sobre la implementación de tecnologías avanzadas de almacenamiento y procesamiento de datos para optimizar la gestión de datos complejos y sus múltiples fuentes. Al aprovechar las capacidades de las arquitecturas de *Data Lakehouse*, las investigaciones demuestran cómo manejar grandes volúmenes de datos heterogéneos de manera eficiente, lo que permite realizar análisis más precisos y completos que mejoran y guían la toma de decisiones médicas.

La relación entre los artículos revela una tendencia significativa en la integración multimodal de datos médicos y su importancia en el campo de la gestión de enfermedades como la diabetes tipo 2. Por ejemplo, [23] y [29] destacaron cómo la integración de datos multimodales ofrece una visión más completa de los casos clínicos. Este enfoque es comparable a la forma en que un *Data Lakehouse* maneja datos de pacientes con diabetes tipo 2, identificando patrones que influyen en la remisión de la enfermedad. Por otro lado, Xiao et al. [30] y Zhao et al. [27] propusieron plataformas que unifican datos heterogéneos lo cual es crucial para consolidar información de diversas fuentes médicas y lograr un análisis más eficaz de la diabetes.

En términos de optimización del almacenamiento y procesamiento de datos, las investigaciones de Schneider et al. [25] y Vox et al. [26] examinaron cómo la arquitectura *Lakehouse* se emplea para mejorar el rendimiento de las consultas analíticas, lo que se traduce en tiempos de respuesta más rápidos y un análisis más eficiente, lo cual es esencial para manejar grandes volúmenes de datos relacionados con la diabetes tipo 2. De igual manera, Harby et al. [28] demostraron que el uso de

tecnologías como Delta Lakehouse y MongoDB facilita el manejo de datos críticos en tiempo real, lo cual es vital para el monitoreo continuo de pacientes diabéticos.

El enriquecimiento y la calidad de los datos también son aspectos críticos abordados por Sheng et al. [31], quienes exploraron técnicas avanzadas para la imputación de datos faltantes y la limpieza de datos, aspectos que son fundamentales para asegurar que los análisis sobre datos de diabetes tipo 2 sean precisos y fiables. Bellandi et al. [24] se centraron en la curación de datos y el uso de flujos de trabajo automatizados para mejorar la calidad de estos, algo esencial para obtener conclusiones válidas en estudios relacionados con la diabetes.

Por último, la interoperabilidad y los estándares son aspectos enfatizados por Tripathi et al. [23], quienes subrayaron la necesidad de utilizar estándares como FHIR y HL7 para asegurar que los datos de salud sean accesibles y consistentes, facilitando así un análisis integrado. La interoperabilidad es crucial para un *Data Lakehouse* enfocado en la diabetes tipo 2, ya que garantiza que los datos provenientes de múltiples sistemas de salud se integren y analicen de manera coherente.

La capacidad de integrar y analizar datos heterogéneos de diversas fuentes es crucial para identificar los factores que influyen en la remisión de la diabetes. Las estrategias descritas en los artículos para gestionar y consolidar datos multimodales se adaptaron para asegurar que se capturen todos los aspectos relevantes de los pacientes.

Otro punto importante para considerar es la optimización del almacenamiento y la mejora del rendimiento de las consultas que permiten que el *Lakehouse* procese eficientemente grandes volúmenes de datos. Las técnicas de limpieza y mejora de datos discutidas aseguran que los análisis se realicen sobre datos de alta calidad, reduciendo el sesgo y aumentando la precisión de los modelos predictivos relacionados con la remisión de la diabetes.

La capacidad de realizar inferencias causales y modelar datos complejos ayudó a identificar patrones y correlaciones que son cruciales para entender los mecanismos de remisión en la diabetes tipo 2, permitiendo el desarrollo de intervenciones más efectivas y personalizadas.

2.3 Propuesta de solución

De acuerdo con el análisis realizado, se consideraron los trabajos más relevantes y similares para establecer una propuesta de solución alineada con los objetivos establecidos. La arquitectura propuesta se diseñó utilizando herramientas de código abierto de Apache, seleccionadas por su disponibilidad sin restricciones de licencia y su robustez en entornos de procesamiento de datos a gran escala.

La solución se organizó en una arquitectura por capas orientada a los datos, estructurada en cinco componentes principales:

- Ingesta de Datos

Se emplean Apache Kafka [32] y Apache NiFi [33] para la ingesta y captura eficiente de datos. Kafka permite la transmisión en tiempo real de flujos de datos desde diversas fuentes, mientras que NiFi facilita la orquestación y transformación de datos antes de su almacenamiento.

- Procesamiento de Datos

Para el procesamiento y transformación de los datos, se integran Apache Spark [34] y Apache Flink [35]. Spark proporciona un marco de procesamiento en *batch* y en memoria altamente escalable, mientras que Flink se utiliza para el procesamiento de flujos en tiempo real con baja latencia.

- Almacenamiento de Datos

Se utilizó Apache Iceberg [36] como sistema de almacenamiento transaccional distribuido, el cual permite gestionar grandes volúmenes de datos con soporte para control de versiones, escrituras incrementales y optimización de consultas analíticas. Sus principales beneficios incluyen la consistencia transaccional, el versionado de datos, y la capacidad de integrarse de forma nativa con motores como Spark, Trino y Flink, facilitando el análisis confiable y eficiente en arquitecturas tipo *Lakehouse*.

- Análisis de Datos

Para la consulta y análisis de datos, se utilizan Apache Doris [37] y Trino. Doris ofrece capacidades de análisis OLAP en tiempo real, mientras que Trino permite realizar consultas distribuidas en diferentes formatos y fuentes de datos con alta eficiencia.

- Visualización de Datos

La interpretación y presentación de la información se realizan mediante Apache Superset [38] y Metabase [39], plataformas que permiten construir *dashboards*

interactivos, realizar análisis exploratorio y facilitar la toma de decisiones basada en datos.

Adicionalmente, Apache Airflow [40] se emplea como herramienta principal para la orquestación de flujos de trabajo y la automatización de procesos dentro del ecosistema del *Data Lakehouse*, permitiendo una integración fluida y controlada entre cada una de las capas de la arquitectura. Airflow permite definir *pipelines* de datos de manera declarativa mediante DAGs (*Directed Acyclic Graphs*), lo que facilita la visualización, monitoreo y gestión de tareas complejas de transformación, ingesta, análisis y carga. Esta herramienta se alinea con los principios de la metodología DataOps, al habilitar la automatización continua, la modularización de tareas, la trazabilidad de cada ejecución, y la capacidad de escalar y versionar procesos de manera eficiente.

Para llevar a cabo la construcción de este sistema se utilizó la metodología DataOps [41], ya que ofrece un enfoque centrado en los datos y su comportamiento a lo largo del ciclo de vida. Esta metodología combina principios de desarrollo ágil, integración y entrega continua (CI/CD), automatización y colaboración entre equipos de desarrollo, operación y análisis de datos. Su aplicación permite definir y orquestar flujos de datos robustos, repetibles y escalables, garantizando la calidad, trazabilidad y gobernanza de la información en cada etapa. Además, DataOps promueve la entrega continua de valor mediante la iteración rápida de componentes, el monitoreo automatizado de procesos, y la detección temprana de errores, lo que resulta especialmente valioso en entornos complejos como los *Data Lakehouse*, donde conviven datos estructurados, semiestructurados y no estructurados.

Esta propuesta permite gestionar de manera eficiente la captura, procesamiento, almacenamiento, análisis y visualización de datos en el contexto de un *Data Lakehouse*, asegurando escalabilidad, flexibilidad y trazabilidad de la información.

Capítulo 3. Aplicación de la metodología

Dentro de este capítulo se presenta una explicación detallada sobre la metodología empleada en dos vertientes complementarias. Por un lado, se describe el proceso de búsqueda, selección y análisis sistemático de trabajos relacionados con tecnologías de datos como *Data Lakehouse*, *Data Warehouse* y *Data Lake*, utilizando el enfoque PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) para asegurar la transparencia y exhaustividad en la revisión de la literatura. Por otro lado, se introduce la implementación de la metodología DataOps como eje central del desarrollo técnico del *Data Lakehouse*, permitiendo una integración continua, automatización de flujos de trabajo, y una gestión ágil y colaborativa de los datos a lo largo de todo su ciclo de vida. Esta distinción metodológica permite abordar tanto el sustento teórico como la ejecución práctica de la arquitectura propuesta.

3.1 Análisis

El estudio del estado del arte se realizó empleando la metodología PRISMA [42] en su versión del 2020, esto para obtener 50 artículos los cuales fueron filtrados de las principales bibliotecas digitales, esto se muestra en la Figura 3.1, se buscaron aquellos relacionados con *Data Lakehouse*, *Data Warehouse* y *Data Lake*, de los cuales se encontraron un total de 1652 artículos en las bibliotecas digitales IEEE Xplore (312), ACM Digital Library (41), ScienceDirect (26), SpringerLink (96), Wiley Online Library (11), MDPI (174) y Google académico (992).

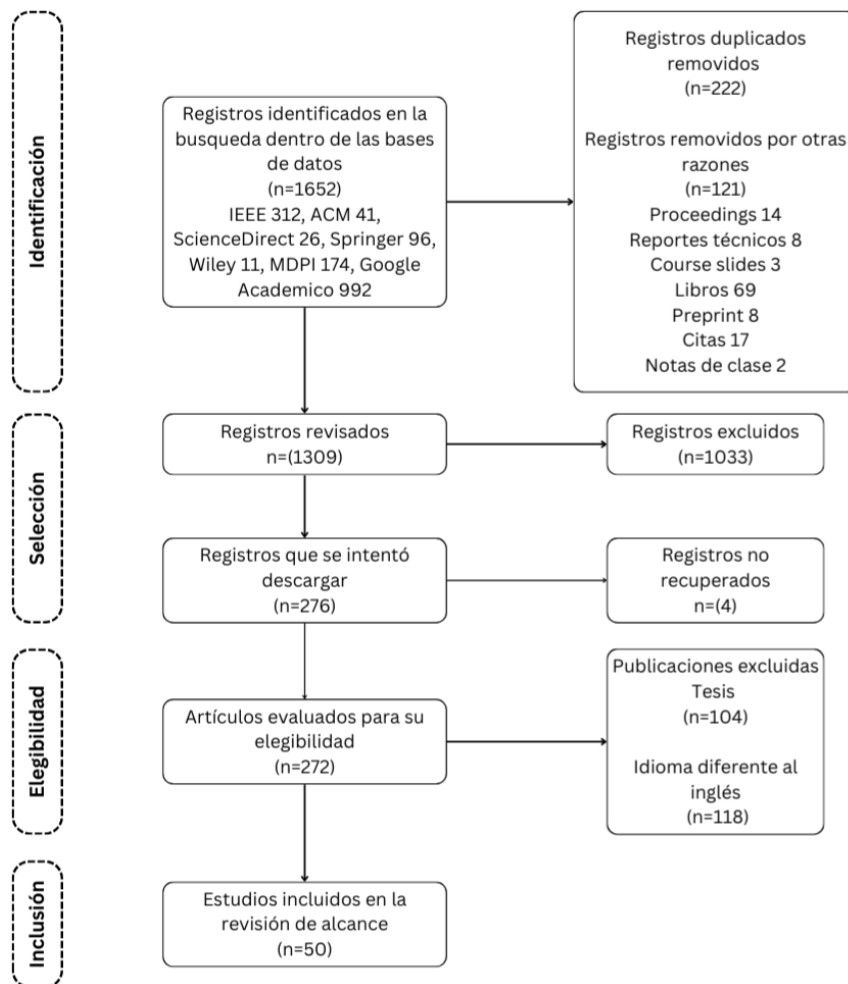


Figura 3.1 PRISMA

Estos artículos se filtraron adicionalmente de acuerdo con su antigüedad, que no tuvieran más de 4 años de haberse publicado, que estuvieran en inglés y que fueran accesibles, una vez que los 50 artículos quedaron seleccionados se aplicó una segunda fase de clasificación donde se separaron por su tipo de repositorio, por su dominio y por su enfoque en diabetes, por último, se clasificaron las tecnologías que emplearon para el desarrollo de repositorios de acuerdo con las capas esperadas. En la Figura 3.2 se representa la metodología de clasificación empleada.

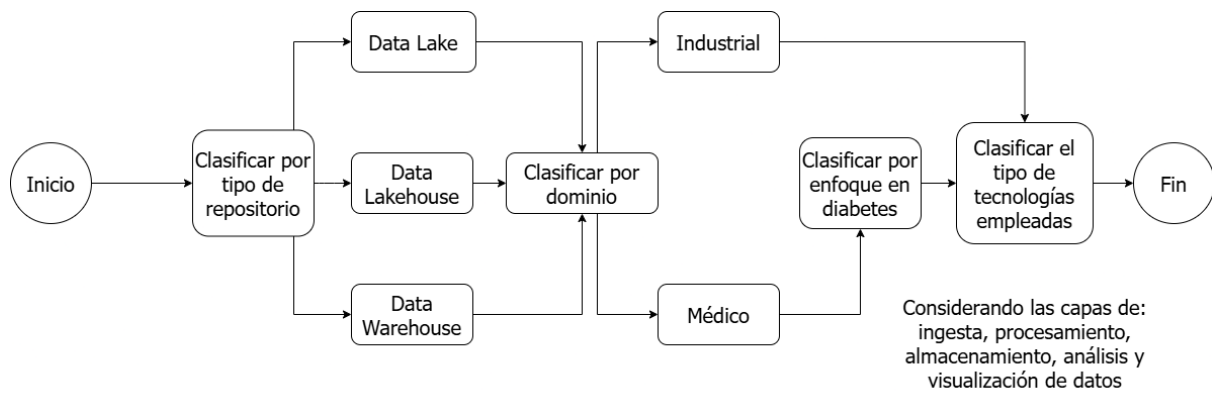


Figura 3.2 Metodología de análisis

La Tabla 3.1 ofrece una visualización más precisa de la representación de la metodología, se utilizaron los criterios de problema para definir qué es lo que se pretendía solucionar, la contribución se refiere al aporte que ofrece la investigación lo cual está directamente relacionado con las tecnologías y su aplicación.

En la parte de los resultados se consideran los productos arrojados por la investigación y si esto fue positivo o negativo acorde con el problema original.

Tabla 3.1 Análisis de criterios de investigación

Artículo	Problema	Contribución	Tecnologías	Resultado
Fleuren et al. [43]	La necesidad de datos confiables, multicéntricos y de admisión completa en cuidados intensivos para avanzar en la comprensión del COVID-19 y explorar estrategias de tratamiento.	Creación del <i>Dutch Data Warehouse</i> (DDW), la primera base de datos multicéntrica que incluye datos completos de admisión de pacientes críticos con COVID-19 en los Países Bajos.	• SQL • SHA-256 • Google Sheets • Python 3 • MetaVision • HiX • Epic • CSV • Cifrado de extremo a extremo.	Se estandarizaron los datos clínicos de 35 hospitales, creando un Data Warehouse con más de 200 millones de puntos de datos clínicos de 3463 pacientes, se integraron datos de 25 hospitales en una base consolidada con validación de calidad.
Pressat-Laffouilhère et al. [44]	Los datos no estructurados de los registros médicos electrónicos representan una riqueza de información difícil de procesar.	Presentación y evaluación formal de Doc'EDS, una herramienta de búsqueda avanzada basada en análisis textual y semántico.	• Almacén de datos clínicos • Base de datos orientada a documentos • Lucene	Doc'EDS mejoró la búsqueda avanzada de texto completo en documentos médicos en francés.

Artículo	Problema	Contribución	Tecnologías	Resultado
			• Procesamiento de lenguaje natural (NLP) • Tika Java • SQL	
Lamer et al. [45]	La necesidad de integrar, organizar y analizar datos anestésicos y clínicos en un único repositorio para facilitar estudios clínicos y mejorar la toma de decisiones.	Desarrollo de un almacén de datos clínicos para anestesia, integrando múltiples fuentes y adoptando el modelo de datos OMOP (<i>Observational Medical Outcomes Partnership</i> , Asociación de resultados médicos observacionales) para fomentar colaboraciones multicéntricas y aprendizaje federado.	• Oracle 11G • OpenText Integration Center • R • SAS • Modelo de datos de Inmon	Almacenamiento de más de 636,000 registros anestésicos de 353,000 pacientes, creación de <i>datamarts</i> para consultas optimizadas e incorporación de datos adicionales como biología, transfusiones y mortalidad extra hospitalaria.
Ansoborlo et al. [46]	Incremento en la demanda de artroplastias de cadera,	Implementación de un programa de procesamiento de lenguaje natural (NLP)	• NLP • R	<i>E-cohort</i> construida con 7,587 pacientes, promedio de edad 67.4

Artículo	Problema	Contribución	Tecnologías	Resultado
	rodilla y hombro (THKSA) y la necesidad de vigilancia automatizada para evitar complicaciones graves en dispositivos médicos implantables.	para la creación automatizada de una <i>e-cohort</i> de pacientes con artroplastias primarias a partir de dos almacenes de datos hospitalarios en Francia.	• Bases de datos de alta hospitalaria (CPT) • Farmacia (LPP) • Servidor RStudio en Ubuntu.	años y proporción 0.75 entre sexos, con análisis detallado de condiciones comórbidas como obesidad y enfermedades cardiovasculares.
Kaspar et al. [47]	Necesidad de combinar datos clínicos y de imágenes para investigación mediante consultas únicas en un almacén de datos hospitalarios.	Integración del sistema de almacenamiento de imágenes PACS con un almacén de datos clínicos (CDW) utilizando el modelo de datos EAV (<i>Entity-attribute-value</i> , Entidad-atributo-valor) extendido para consultas conjuntas multivelocidad.	• PACS • Modelo de datos EAV • Almacén de datos clínicos (CDW) • Metadatos DICOM (<i>Digital Imaging and Communications in Medicine</i>, Imágenes y comunicaciones digitales en medicina).	Consultas multivelocidad que aumentaron la representación cuantitativa de resultados hasta un 573% en comparación con consultas a nivel único, mejorando la integración y disponibilidad de datos combinados.

Artículo	Problema	Contribución	Tecnologías	Resultado
Baum et al. [48]	Necesidad de comprender y optimizar cómo los investigadores interactúan con almacenes de datos clínicos, específicamente con Informática para la Integración de la Biología y la Atención Médica (i2b2), para mejorar su uso en investigaciones médicas.	Desarrollo de métricas para describir las interacciones de los usuarios con i2b2 y creación de un panel interactivo con visualizaciones para identificar áreas de mejora en la plataforma.	• i2b2 • Python • Dash • Sqlite3	Panel interactivo que proporciona métricas clave como frecuencia de uso, duración de sesiones y complejidad de consultas, que ayuda a identificar fortalezas y limitaciones para su mejora iterativa.
Ndisha et al. [49]	Las bases de datos electrónicas agregadas y encuestas poblacionales no permiten un monitoreo en tiempo real ni un	Creación y desarrollo de un repositorio nacional centralizado de datos (NDW) en Kenia, basado en registros médicos electrónicos (EMRs), para el monitoreo y vigilancia	• DWAPI: Basado en Angular y .NET Core, soporta MSSQL (Microsoft SQL) y MySQL. Utiliza algoritmos de	Cobertura de datos longitudinales de 1,928,458 individuos en terapia antirretroviral (ART) hasta 2020. Proporciona monitoreo

Artículo	Problema	Contribución	Tecnologías	Resultado
	seguimiento longitudinal de pacientes, limitando la capacidad para evaluar resultados a largo plazo en programas de VIH.	de programas de VIH a nivel individual y poblacional.	deduplicación y generación de identificadores únicos (PKV) mediante métodos como Jaro-Winkler. • JSON • RabbitMQ • Hangfire • Highcharts	oportuno en tiempo real y fortalece la toma de decisiones basada en datos para el control de la epidemia de VIH.
Cheng-Zarate et al. [50]	La gestión de sangre en hospitales requiere evaluar la calidad y la documentación de las transfusiones de sangre según los estándares nacionales de Australia (NSQHS). Sin embargo, los registros médicos electrónicos (EMR) no	Se diseñó e implementó un almacén de datos clínicos para transformar datos del EMR en un formato que permita evaluar la conformidad hospitalaria con los estándares de gestión de sangre de los NSQHS. El sistema facilita el análisis de datos históricos, permite la	• Extracción de datos: Lenguaje de comandos Cerner (CCL) para consultar la base de datos Oracle Cerner. • Procesamiento y transformación: Python.	Se desarrolló un sistema funcional junto con Eastern Health, un importante servicio de salud australiano. Este almacén de datos permite realizar consultas eficientes sobre la documentación de transfusiones, generar

Artículo	Problema	Contribución	Tecnologías	Resultado
	están diseñados para realizar consultas analíticas complejas, lo que dificulta la generación de métricas necesarias para apoyar herramientas de decisión clínica.	generación de métricas y proporciona flexibilidad para recalcular estas métricas cuando se actualizan los datos.	- Almacenamiento y consulta: Almacén de datos diseñado con SQL, soportando reconstrucción de eventos de transfusión a través de triggers y procedimientos. - Visualización: Tableau para presentar métricas definidas por los NSQHS de forma comprensible para los clínicos.	métricas personalizadas, y facilitar su consumo por herramientas de soporte a decisiones clínicas. Además, se diseñó un <i>dashboard</i> para mostrar métricas clave relacionadas con la gestión de sangre.

Artículo	Problema	Contribución	Tecnologías	Resultado
Turcan et al. [51]	La pandemia de Covid-19 generó la necesidad de gestionar una cantidad masiva de datos pandémicos a nivel nacional, para facilitar decisiones rápidas y efectivas durante emergencias sanitarias.	Se propuso y desarrolló un almacén de datos utilizando la metodología <i>bottom-up</i> de Kimball y un modelo de esquema de constelación de hechos. Este almacén guarda datos multidimensionales sobre pruebas, vacunación, síntomas, cuarentenas y medicación, permitiendo análisis detallados y consultas ágiles.	Metodología <i>bottom-up</i> de Kimball, esquemas de constelación de hechos, tablas de hechos y tablas de dimensiones jerárquicas.	El diseño permitió consultas rápidas y efectivas para gestionar datos pandémicos, demostrando utilidad en el análisis multidimensional y apoyando iniciativas de salud pública locales, nacionales y globales.
Arnold et al. [52]	Aunque los registros electrónicos de salud (EHR) ofrecen un acceso sin precedentes a datos clínicos para investigación, su uso	El artículo detalla los procesos, consideraciones y herramientas necesarias para acceder y utilizar eficazmente un almacén de datos clínicos (<i>Health Data Warehouse</i>) en	Se describen almacenes de datos basados en la nube, como Health Data Compass, y procesos de vinculación con datos genómicos utilizando	El uso de almacenes de datos clínicos permite realizar análisis longitudinales de pacientes, integrar datos genómicos y clínicos, y

Artículo	Problema	Contribución	Tecnologías	Resultado
	secundario implica desafíos relacionados con la privacidad, interoperabilidad, validación y transformación de los datos en conjuntos útiles para análisis.	investigaciones. Además, aborda las etapas de solicitud, limpieza y análisis de datos, así como la integración con biobancos y otras fuentes, destacando el caso del Health Data Compass de la Universidad de Colorado.	biobancos y herramientas como el chip Illumina Infinium MEGA-EX para genotipado. También se mencionan modelos de datos comunes y plataformas para datos longitudinales.	aplicar descripciones analíticas avanzadas para mejorar la investigación traslacional, operativa y de calidad en salud.
Windsor et al. [53]	Las NSTIs (infecciones necrotizantes de tejidos blandos) son infecciones graves y de rápida progresión, caracterizadas por necrosis en tejidos subcutáneos, fascia e incluso músculos. Existe una comprensión limitada sobre la trayectoria de atención	Este estudio retrospectivo multicéntrico proporcionó datos a gran escala sobre la atención hospitalaria de pacientes con NSTIs en 13 hospitales del área de París. Se analizaron factores asociados con resultados complicados, mortalidad y reingreso en UCI, enfocándose en diferencias entre	- Base de datos hospitalaria francesa (PMSI) para identificación de casos. - Clasificación Internacional de Enfermedades (CIE-10) para diagnósticos. - Herramientas estadísticas: Python	□ Demografía: Predominio de hombres (71.8%) con una mediana de edad de 59 años. □ Complicaciones: 80.2% desarrollaron fallas orgánicas, requiriendo soporte vasopresor (77.2%) o ventilación mecánica (51.8%).

Artículo	Problema	Contribución	Tecnologías	Resultado
	de estos pacientes, lo que dificulta identificar a quienes están en riesgo de evolución complicada y requieren una atención más prolongada o intervenciones repetidas.	inmunocomprometidos y no inmunocomprometidos.	y R para análisis de datos.	☐ Mortalidad: En UCI fue de 21.8% y en hospital del 28.8%. ☐ Factores de riesgo: NSTIs polimicrobianas y bacteriemia aumentaron la probabilidad de una evolución complicada. ☐ Hospitalización: El 62.3% fueron readmitidos en un plazo de seis meses. ☐ Inmunosupresión: Pacientes inmunocomprometidos presentaron mortalidades significativamente más altas.

Artículo	Problema	Contribución	Tecnologías	Resultado
Shae et al. [54]	La propiedad de los datos, especialmente los médicos, presenta complejidades legales y éticas, sin una legislación específica en la UE (unión europea) que regule estos derechos, dificultando su compartición y uso.	Se propone una plataforma basada en blockchain para gestionar la propiedad de los datos médicos en un ecosistema federado, utilizando contratos inteligentes, IA y <i>federated learning</i> para asegurar la privacidad y protección de los datos.	• Blockchain • Contratos inteligentes • <i>Federated learning</i> • IA	La plataforma facilitaría el acceso y análisis de datos médicos de forma segura, con un sistema de incentivos y roles definidos para los participantes.
Melchor-Uceda et al. [55]	La creciente generación de grandes volúmenes de datos en el sector salud, provenientes de dispositivos biomédicos, usables y sistemas de información, plantea la necesidad de integrar y gestionar estos datos de manera eficiente y en	Se desarrolló un sistema de ingesta de datos hospitalarios basado en un lago de datos, que permite la integración y procesamiento de datos en tiempo real de dispositivos como electrocardiógrafos y pulseras inteligentes. Además, se creó una plataforma web interactiva	• MySQL • React • APIs de Google Fit, Fitbit y Huawei • Técnicas criptográficas • Microsoft Azure	Mejoró la interoperabilidad de los datos médicos, facilitando la toma de decisiones clínicas y optimizando la gestión de la información hospitalaria. Se logró una visualización en tiempo real de los datos a través

Artículo	Problema	Contribución	Tecnologías	Resultado
	tiempo real, con el fin de mejorar la interoperabilidad, visualización y procesamiento de la información médica.	para la visualización de estos datos, mejorando la interacción entre médicos, pacientes y la información obtenida. El sistema también incorpora técnicas de ciberseguridad para proteger los datos sensibles y la base de datos MySQL para almacenar registros médicos.		de una plataforma web, y la integración con dispositivos biomédicos y utilizables permite un monitoreo más eficiente de los pacientes. A futuro, se implementarán modelos predictivos para enfermedades como las cardíacas y la diabetes, y se ampliará la capacidad de integración con otros dispositivos mediante los modelos <i>push</i> y <i>pull</i> .
Babu et al. [56]	El almacenamiento eficiente y accesible de registros médicos electrónicos (EMR/EHR), especialmente en	Propuesta de un sistema basado en Microsoft Azure Data Lake que almacena datos médicos de cualquier tipo (texto, imágenes, videos, entre otros) y utiliza el	Microsoft Azure Data Lake para almacenamiento en la nube y KNN como algoritmo de clasificación de datos.	Acceso global y eficiente a datos médicos, mejor capacidad para el diagnóstico automatizado mediante el análisis de registros previos,

Artículo	Problema	Contribución	Tecnologías	Resultado
	emergencias, y la dificultad de manejar datos no estructurados en infraestructuras locales o con recursos limitados.	algoritmo KNN para clasificar y facilitar la recuperación y análisis de la información.		eliminación de problemas de almacenamiento local, gestión efectiva de información en emergencias, y posibilidad de eliminar datos duplicados o innecesarios.
Mabel Rani et al. [57]	La integración de registros médicos en repositorios de datos de salud enfrenta desafíos, especialmente en países subdesarrollados. Problemas como datos	Se propone una arquitectura de enlace de registros segura y preservadora de privacidad llamada <i>Secured Health Data Handler</i> (SHDH). Esta solución anonimiza datos privados de los pacientes mientras mantiene la	• Blockchain (Ethereum, Ganache, Web3.js, Solidity) • Sistema de archivos interplanetarios (IPFS)	Mejoras en la eficiencia del sistema: Incremento de hasta 5-6% en la frecuencia de datos (59.84% con SHDH frente a 52.65% con CDS para 1000 kbps).

Artículo	Problema	Contribución	Tecnologías	Resultado
	faltantes, identificación ambigua de pacientes, y ruido en la información dificultan el enlace de registros, agravados por la falta de identificadores únicos. Además, existe una preocupación seria por la privacidad de la información personal en los registros electrónicos de salud, que son susceptibles a brechas de seguridad.	capacidad de enlazar registros en repositorios integrados de salud. También introduce medidas basadas en <i>blockchain</i> e IPFS para garantizar la seguridad y accesibilidad descentralizada de los registros médicos.	• Contratos inteligentes para control de acceso y privacidad • <i>Frameworks</i> para evaluación de seguridad, como SPAKE y bases de datos distribuidas.	Incremento de 6-8% en la frecuencia de clústeres (52.69% con SHDH frente a 46.26% con CDS para 125 kbps). Mayor precisión y seguridad: SHDH permite un acceso controlado a datos confidenciales basado en niveles de confianza, protegiendo la privacidad de los pacientes.
Amara et al. [58]	El proceso de manejo de datos DICOM es complejo, y el	Un almacén de datos clínicos que organiza imágenes y datos médicos, facilitando el	• Encabezados DICOM para organizar los datos.	La integración de datos redujo el tiempo de diagnóstico y mejoró la

Artículo	Problema	Contribución	Tecnologías	Resultado
	diagnóstico de cáncer de mama requiere integrar información de diversas fuentes, lo que puede retrasar la evaluación. Además, la segmentación incorrecta de imágenes y la dependencia de la experiencia del radiólogo pueden afectar la precisión del diagnóstico.	diagnóstico y la investigación. Se implementó un sistema de segmentación automatizada en imágenes (Imágenes de resonancia magnética) MRI, combinado con (Tomografía por Emisión de Positrones / Tomografía Computarizada) PET/CT, para mejorar la precisión en la localización de lesiones. También se creó una interfaz XML para análisis de datos, con consultas avanzadas y visualización interactiva.	• Bases de datos SQL y Sybase con controlador jTDS para la extracción. • MRI y PET/CT • SAS y SPSS • XML	precisión utilizando MRI y PET/CT. La segmentación automatizada evitó falsas detecciones, y la capacidad de realizar consultas complejas optimizó el análisis clínico y el seguimiento de pacientes.
Kumar et al. [59]	La capacidad de la infraestructura actual de Redshift para gestionar grandes volúmenes de datos históricos, debido	Se propuso implementar un sistema que desacoplara almacenamiento y cómputo, utilizando Databricks para aprovechar su flexibilidad de	La plataforma utiliza AWS S3 para almacenamiento, Delta Lake para la capa de datos ACID-compliant, y Spark para el	El uso de Databricks permitió procesar consultas complejas en menos de 2 horas en comparación con las 22

Artículo	Problema	Contribución	Tecnologías	Resultado
	a su combinación de almacenamiento y cómputo, lo que incrementaba los costos. La necesidad de escalar rápidamente el almacenamiento afectaba la eficiencia y la sostenibilidad económica de la plataforma.	almacenar datos en AWS S3 y procesarlos con <i>clusters</i> Spark, lo cual permitió reducir los costos operativos y mejorar la eficiencia en los casos de uso tanto para Ingeniería de Datos como para Ciencia de Datos.	procesamiento y análisis de datos. La plataforma también usa servicios como Unity Catalog para la gobernanza de datos y control de acceso.	horas que tomaban en Redshift, lo que resultó en una mejora en el tiempo de ejecución y en la reducción de costos. Al comparar costos, el uso de Databricks fue aproximadamente la mitad de lo que costaba en Redshift, a pesar de que las consultas eran 11 veces más rápidas.
Hernandez Cruz et al. [60]	El caso aplicado busca desarrollar un Data Warehouse para gestionar datos sobre la COVID-19, como contagios, muertes y vacunación, utilizando fuentes externas de la	Se implementó la metodología Hefesto v2.0 para diseñar un Data Warehouse con un esquema estrella, incluyendo la creación de Tablas de Dimensiones y la Tabla de Hechos, y la identificación de	• Microsoft SQL Server 2019 (gestión de base de datos). • Visual Studio 2019 (para ETL con Integration Service). • Excel (fuentes de datos).	Se construyó con éxito un Data Warehouse que integró los datos sobre COVID-19, automatizó la carga mediante ETL y facilitó el análisis de indicadores clave como

Artículo	Problema	Contribución	Tecnologías	Resultado
	OMS para ofrecer información actualizada y facilitar la toma de decisiones.	relaciones entre los datos. Además, se automatizó el proceso ETL con Visual Studio y SQL Server.	Hefesto v2.0	contagios, muertes y vacunación.
Spengler et al. [61]	El artículo abordó problemas de calidad de los datos en procesos ETL en proyectos de investigación médica, como valores faltantes, categorías inválidas y violaciones de integridad, especialmente cuando se integran diversas fuentes de datos (<i>dataset</i> de investigación, datos clínicos y hospitalarios).	Se implementó un sistema de monitoreo de calidad de datos que incluye: Generación de eventos asignados a pantallas predefinidas para análisis estructurado. Plantillas de auditoría para realizar verificaciones de calidad genéricas, específicas del sistema y específicas de la instancia. <i>Dashboard</i> en Grafana para visualizar parámetros del sistema y calidad de datos.	- API de eventos para la clasificación de problemas de calidad de datos. - Grafana para la visualización. - Docker Compose para la implementación del servicio de auditoría. - ETL <i>pipeline</i> automatizado integrado con monitoreo de calidad.	Permitió identificar problemas de calidad en tres proyectos de investigación con datos de hasta 6 millones de hechos: <i>Billing dataset</i>: 0.1% de problemas. <i>Research dataset</i>: 0.05%. <i>Clinical dataset</i>: 6%, principalmente valores faltantes y categorías inválidas.

Artículo	Problema	Contribución	Tecnologías	Resultado
		Implementación de circuitos de retroalimentación para mejorar la calidad de los datos iterativamente.		
Divya et al. [62]	Garantizar la calidad de datos sensibles en Enterprise Data Warehouses (EDW) y Enterprise Data Hubs (EDH) del sector salud para análisis confiables y soluciones efectivas.	Desarrollo de un marco bien definido que asegura la calidad de datos mediante estrategias como segregación de datos, monitoreo de desempeño, y reglas de calidad específicas. Incluye la propuesta del F-Status para gestionar la diversidad en conjuntos de datos heterogéneos.	• Data Governance. • Data Profiling. • Master Data Management (MDM). • Monitoreo y reporte de sostenibilidad de datos con KPI y reglas de calidad.	Mejora de la precisión, completitud, consistencia, relevancia y oportunidad de los datos. Reducción de costos operativos gracias a datos sin errores. Creación de métricas de calidad accesibles mediante <i>dashboards</i> y reportes.
Setiadi et al. [63]	El diseño existente de almacenes de datos para monitorear casos de COVID-19 en Indonesia carecía de	Se propuso un diseño mejorado de almacén de datos utilizando la metodología de cuatro pasos de Ralph Kimball, incluyendo	• Metodología de cuatro pasos de Ralph Kimball • Esquema copo de nieve	El diseño resultante en un esquema de copo de nieve permitió mejorar la calidad de los datos, optimizar el uso de disco

Artículo	Problema	Contribución	Tecnologías	Resultado
	información clave como la dimensión "variante", lo que limitaba el análisis preciso y regionalizado de casos. Además, las soluciones actuales enfrentaban problemas de calidad y granularidad de los datos.	la integración de "variante" como una nueva dimensión junto a fecha, pacientes y centros de salud. El diseño emplea un esquema de copo de nieve para ofrecer mayor granularidad y calidad de los datos.	ETL	y facilitar análisis regionales para decisiones informadas sobre el manejo del COVID-19 en Indonesia.
Neamah et al. [64]	La gestión de registros médicos electrónicos (EHR) enfrenta desafíos significativos relacionados con la disponibilidad de datos, la predicción de diagnósticos y la integración de información	Enfoque basado en almacenes de datos flexibles y módulos de registros médicos electrónicos que integren tecnologías como bases de datos NoSQL, modelos de aprendizaje automático y aplicaciones móviles. Este enfoque mejora el almacenamiento,	- Almacenes de Datos Flexibles: Implementación de bases de datos estructuradas y no estructuradas para la integración y manejo de EHR. - Bases de Datos NoSQL.	El enfoque mejoró significativamente la disponibilidad y precisión en el análisis de registros médicos. Este proporcionó beneficios como la minimización de redundancias, optimización de espacio de almacenamiento,

Artículo	Problema	Contribución	Tecnologías	Resultado
	estructurada y no estructurada. Además, los sistemas tradicionales presentan limitaciones en el manejo de grandes volúmenes de datos y la flexibilidad requerida para el análisis predictivo en el sector salud.	recuperación y análisis de datos para apoyar decisiones clínicas con mayor precisión y eficiencia.	- Modelos de Aprendizaje Automático. - Servicios Basados en la Nube: Para acceso y almacenamiento centralizado.	autenticación mediante firmas electrónicas y análisis profundo mediante minería de datos. Además, facilitó la integración de datos con plataformas móviles y la automatización de flujos de trabajo.
Yeng et al. [65]	Falta de seguridad y privacidad en las estrategias digitales de salud en naciones africanas, especialmente en países de ingresos bajos y medianos (LMICs), y la necesidad de cumplir	Desarrollo de un marco de gestión de incidentes de seguridad y un SLA (<i>Service Level Agreement</i> , Acuerdo de nivel de servicio o Garantía de nivel de servicio) alineado con las políticas nacionales de salud, utilizando tecnologías como Apache Atlas, NiFi,	- Apache Atlas: Gestión de metadatos y linaje de datos. - Apache NiFi: Ingesta y procesamiento de datos.	Un sistema seguro y eficiente para la gestión de datos de salud, alineado con políticas de seguridad, que facilita el análisis avanzado, la protección de datos y la generación de informes en tiempo real.

Artículo	Problema	Contribución	Tecnologías	Resultado
	con normativas como el Ghana Cybersecurity Act.	Spark, Cassandra y Superset para la gobernanza de datos, procesamiento, y visualización.	• Apache Spark: Procesamiento distribuido de datos. • Apache Cassandra: Almacenamiento de datos. • Apache Superset: Visualización de datos. • FME: Integración de datos GIS (<i>Geographical Information System</i>, Sistema de Información Geográfica).	
Ramesh et al. [66]	Las enfermedades cardiovasculares son una de las principales causas de muerte en el	El artículo propone un modelo basado en un Data Lake para integrar y analizar datos de salud de diversas fuentes,	• Data Lake: Para almacenar y gestionar grandes volúmenes de datos	El modelo propuesto mostró una alta precisión en la predicción de enfermedades cardíacas.

Artículo	Problema	Contribución	Tecnologías	Resultado
	mundo, y existe una creciente necesidad de métodos para predecir y detectar tempranamente estos riesgos en pacientes. Actualmente, los datos médicos están distribuidos entre múltiples fuentes (Registros de Salud Electrónicos, Registros de Salud Personales, entre otros), lo que dificulta el análisis integrado y la predicción precisa del riesgo cardiovascular.	utilizando algoritmos de clasificación (KNN, Naïve Bayes y Random Forest) para predecir el riesgo de enfermedades cardíacas. Este enfoque permite el procesamiento de datos de manera eficiente y la clasificación de pacientes con mayor riesgo, calculando puntajes de riesgo para una intervención temprana.	de salud no estructurados, provenientes de múltiples fuentes (EHR, PHR). - Algoritmos de clasificación: KNN (K=1), Naïve Bayes, Random Forest para predecir el riesgo de enfermedades cardíacas. - UCI Repository: Para experimentar con un conjunto de datos sobre enfermedades cardíacas.	El algoritmo KNN (K=1) alcanzó una precisión del 99.75%, seguido de Random Forest (96.02%) y Naïve Bayes (84.1%). Estas métricas de evaluación, como la curva ROC (Receiver Operating Characteristic) y los valores de Kappa, indican la efectividad del modelo para identificar pacientes en riesgo. Esto demostró el potencial de los Data Lakes para mejorar la precisión en la predicción y tratamiento preventivo de enfermedades cardíacas.

Artículo	Problema	Contribución	Tecnologías	Resultado
Kumar et al. [67]	El artículo abordó la necesidad de adaptar la Tecnología de Información en Salud (HIT) para manejar grandes volúmenes de datos en tiempo real. Con la acumulación masiva de datos de salud, el proceso tradicional de análisis por lotes ya no es suficiente, y se requiere la capacidad de realizar análisis de datos en <i>streaming</i> a gran escala.	Se propuso una plataforma avanzada de HIT para el análisis de datos en tiempo real y de múltiples granularidades, que integra tecnologías de big data como Apache Kafka, Spark Structured Streaming Engine y Delta Lake. Esta plataforma permite el procesamiento y análisis de datos provenientes de múltiples fuentes, y aplica un algoritmo de detección de peligros basado en el control estadístico de procesos para identificar anomalías en los datos de cancelación de órdenes del Departamento de Asuntos de Veteranos de EE. UU. (VA).	• Apache Kafka: Para la ingestión de datos en tiempo real. • Spark Structured Streaming Engine: Para el procesamiento paralelo de datos en tiempo real. • Delta Lake: Para el almacenamiento de datos con propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), asegurando la durabilidad y	Se implementó con éxito un sistema de detección de peligros en el entorno de salud, que mejora la capacidad de monitorear y detectar anomalías en los registros electrónicos de salud (EHR) del VA en tiempo real. Además, se permitió realizar análisis con diferentes granularidades de tiempo (horaria, diaria, mensual) y optimizar el proceso mediante procesamiento paralelo.

Artículo	Problema	Contribución	Tecnologías	Resultado
			consistencia de los datos.	
Patil et al. [68]	La falta de interoperabilidad y la complejidad de procesar datos FHIR (<i>Fast Healthcare Interoperability Resources</i>) para análisis.	Desarrollo de un enfoque genérico que convierte los datos FHIR en un formato tabular para análisis, sin necesidad de conocimientos profundos de FHIR.	• FHIR • Spark • Iceberg • MinIO • Synthea	Reducción de la curva de aprendizaje para procesar datos FHIR, mejora en la escalabilidad y el rendimiento del procesamiento distribuido. Los datos se almacenan en formato Iceberg, con una compresión eficiente.
Abdelhedi et al. [69]	Abordó la necesidad de proporcionar un mecanismo de recomendación para seleccionar esquemas de datos (DMs) más adecuados para las	La contribución principal del artículo es la implementación del proceso DWtoDM, que utiliza un sistema de recomendación basado en un análisis histórico de sesiones (matriz HS) y la ponderación	• OrientDB (NoSQL para almacenar DMs) • Java (Desarrollo del módulo) • XMI (Formato de metadatos)	El sistema de recomendación funcionó correctamente, variando las recomendaciones según las palabras clave ingresadas, lo que mostró que una selección

Artículo	Problema	Contribución	Tecnologías	Resultado
	sesiones de consulta dentro de un sistema de procesamiento de datos médicos. El desafío es garantizar que las sesiones de los usuarios puedan encontrar los esquemas de datos más relevantes a sus necesidades específicas de forma eficiente.	de atributos para identificar los esquemas de datos más adecuados (DMs). Este mecanismo permite automatizar la recomendación de esquemas de datos para sesiones de consulta en el contexto de un sistema de almacenamiento y análisis de datos médicos. La solución incluye el cálculo de puntajes para cada sesión, teniendo en cuenta la importancia y penalización de los atributos no utilizados, lo cual mejora la relevancia de los esquemas recomendados.	NoSQL Query Language (Consultas en OrientDB)	imprecisa de palabras clave podría aumentar el tiempo de personalización del esquema.
Babu et al. [70]	El manejo de grandes volúmenes de datos médicos no	Se desarrolló una arquitectura de Data Lake que optimiza la ingestión y organización de	Se utilizó el Sistema de Archivos Distribuidos de	La implementación de esta arquitectura mejoró la eficiencia en la

Artículo	Problema	Contribución	Tecnologías	Resultado
	estructurados presenta dificultades en su integración, análisis y almacenamiento. Los sistemas tradicionales de almacenamiento no son eficaces para manejar consultas complejas ni organizar datos de manera eficiente, lo que puede generar un "pantano de datos" difícil de analizar.	datos provenientes de diversas fuentes, evitando la creación de un "pantano de datos". Se emplearon algoritmos de aprendizaje automático, como K-means y SVM, para mejorar la precisión en el análisis de datos médicos y personalizar tratamientos.	Hadoop (HDFS) para almacenar datos de diferentes tipos, tanto estructurados como no estructurados. • La arquitectura de Data Lake se implementó utilizando Hadoop, permitiendo el almacenamiento eficiente de grandes volúmenes de datos. • Para la transferencia de datos, se emplearon herramientas como Spring XD de	ingestión de datos y permitió obtener resultados más precisos en la segmentación de pacientes y la personalización de tratamientos médicos. La adecuada organización de los metadatos evitó el "pantano de datos", facilitando un análisis más efectivo y una mejor toma de decisiones clínicas.

Artículo	Problema	Contribución	Tecnologías	Resultado
			Hadoop y Apache Flume. - El análisis de datos se realizó utilizando algoritmos como K-means para el agrupamiento y SVM para la personalización de recetas médicas. - Los metadatos jugaron un papel crucial en la gestión y organización de los datos en el Data Lake.	
González-Velázquez et al. [71]	La necesidad de monitorear señales en tiempo real para evaluar el estado y la durabilidad	Se desarrollaron productos de datos, incluyendo señales de pruebas de equipos experimentales, estimaciones	Se utilizaron un modelo físico experimental implementado en DLL de C# derivado de MATLAB en	La plataforma gestiona 32 activos industriales divididos en dominios de manufactura y pruebas,

Artículo	Problema	Contribución	Tecnologías	Resultado
	de equipos industriales, reemplazar inspecciones post-manufactura para detectar porosidad en procesos de deposición láser, y la falta de integración centralizada de datos almacenados de forma descentralizada en diferentes dominios dificultan la relación lógica y efectiva entre los datos.	de vida útil remanente (RUL) y métricas de daño para engranajes, así como la predicción de porosidad en procesos de deposición láser. Además, se implementó un catálogo de datos centralizado basado en tecnologías web semánticas para mapear y explotar datos, junto con la creación de ontologías para establecer relaciones entre productos de datos dentro y entre dominios, iniciando con "TribOnt" para tribómetros.	un servicio .NET Core, un modelo de clasificación basado en Random Forest para predecir porosidad, bases de datos MongoDB junto con GraphDB y tecnología OBDA (Acceso a Datos Basado en Ontologías) para mapeo semántico, además de tecnologías web semánticas y el lenguaje SPARQL para consultas avanzadas.	generando nuevos servicios basados en datos mediante técnicas avanzadas de análisis. Se creó un ciclo de aprendizaje para optimizar procesos industriales y se implementó la ontología "TribOnt" para tribómetros, con planes de expandir el catálogo con vocabularios para otros equipos.
Oliveira et al. [72]	Explora cómo superar las limitaciones inherentes a las herramientas	Se propuso un Catálogo de Datos (DC) que integra un diseño modular y escalable. Este catálogo separa	• Microsoft Power BI • Modelos de Grafo • Ontologías	Logra separar la capa conceptual de la implementación técnica, permitiendo flexibilidad

Artículo	Problema	Contribución	Tecnologías	Resultado
	propietarias y formatos específicos en la configuración y documentación de proyectos de inteligencia de negocios (BI). Estas herramientas suelen estar vinculadas a enfoques arquitectónicos predeterminados que no permiten una separación clara entre la capa conceptual y la capa de implementación. Este problema impacta directamente la interoperabilidad, el manejo de metadatos y	claramente la capa conceptual, donde se mapean los artefactos del negocio, de la capa de herramientas, asegurando flexibilidad y adaptabilidad. El DC incluye un modelo de metadatos basado en un grafo que soporta tres niveles principales: metadatos, datos y aplicaciones. Además, se apoya en una ontología para modelar conceptos relacionados con el diseño dimensional, como tablas de hechos, dimensiones y medidas.	• DAX (<i>Data Analysis Expressions</i>) • Motor de Inferencia • Modelo de Datos Dimensional • Modelo de Integración	en la gestión de datos y configuraciones. Utilizando un modelo de grafo para almacenar metadatos, se facilita la navegación semántica y la interoperabilidad entre sistemas, mientras que las ontologías garantizan la consistencia y validación automática de métricas. Los módulos de calidad, alertas y seguridad aseguran que los datos sean precisos, completos y protegidos. Además, la automatización de configuraciones y la inferencia automática de

Artículo	Problema	Contribución	Tecnologías	Resultado
	la validación de modelos analíticos.			relaciones mejoran la eficiencia y reducen errores.
Harby et al. [73]	La ingesta de datos de alta velocidad y volumen, proveniente de fuentes como dispositivos IoT, presenta dificultades en la priorización y adaptación automática a los cambios en la estructura de los datos. Además, la gestión de metadatos se complica por la creciente digitalización y la necesidad de gobernanza. Los <i>Data Lakes</i> pueden	El artículo propone el uso de tecnologías como Apache NiFi y Apache Flink para mejorar la ingesta y procesamiento de datos en tiempo real. Además, se compararon diferentes sistemas de gestión de metadatos para manejar la heterogeneidad de los datos y facilitar su descubrimiento. También se sugirieron estrategias para evitar que los <i>Data Lakes</i> se conviertan en <i>Data Swamps</i> mediante la correcta limpieza y preparación de los datos.	Se mencionan Apache NiFi y Apache Flink para la ingesta y procesamiento de datos en tiempo real. Amazon Redshift y Amazon S3 se emplean para almacenar y procesar datos estructurados y no estructurados en la arquitectura de AWS. Redshift Spectrum y Snowflake se utilizan	La implementación de Apache NiFi y Flink mejoró la ingesta de datos de manera adaptativa y de bajo costo. La transformación de datos en zonas específicas de un <i>Data Lake</i> facilita su calidad y accesibilidad para el análisis. Finalmente, la integración de Amazon S3 y Redshift optimiza las consultas y la toma de decisiones empresariales.

Artículo	Problema	Contribución	Tecnologías	Resultado
	convertirse en <i>Data Swamps</i> si no se gestionan adecuadamente, dificultando la limpieza y preparación de los datos para su uso.		para realizar consultas eficientes sobre grandes volúmenes de datos.	
Bindal et al. [74]	El desafío principal de Quantum Flow radica en la gestión eficiente de los datos a lo largo de todo el ciclo de vida del procesamiento, desde la recolección hasta el análisis. Esto incluye garantizar la integridad de los datos, la calidad del proceso y la	Quantum Flow ofrece una arquitectura modular y escalable que permite la recolección, ingestión, transformación, y análisis de datos de manera eficiente. Además, se implementaron algoritmos de validación de datos y detección de anomalías para asegurar la integridad y precisión de los	• ETL (<i>Extract, Transform, Load</i>): Usado para procesar y transformar los datos. • <i>Data Lake, Data Warehouse</i>, Bases de Datos: Usados como soluciones de almacenamiento	El sistema asegura la calidad y la accesibilidad de los datos a lo largo de todo el ciclo de vida, mejorando la fiabilidad y permitiendo la escalabilidad. Además, la implementación de pruebas rigurosas asegura que el sistema sea confiable, seguro y

Artículo	Problema	Contribución	Tecnologías	Resultado
	capacidad de adaptarse a requisitos cambiantes.	datos. La integración de herramientas de monitoreo facilita la supervisión del rendimiento del sistema y la resolución de problemas.	escalables y duraderas. - Herramientas de Monitoreo y Análisis: Para asegurar la calidad y el desempeño del sistema. <i>Web Technologies</i> (HTML, CSS, JavaScript): Para el diseño de interfaces accesibles. <i>Data Query Engine</i>: Para realizar análisis e interrogaciones sobre los datos transformados.	eficiente, brindando una experiencia fluida para los usuarios finales.

Artículo	Problema	Contribución	Tecnologías	Resultado
Edara et al. [75]	El sistema Vortex aborda la optimización del almacenamiento de datos en un entorno distribuido, garantizando la consistencia de los datos durante el proceso de escritura y la integración con herramientas de procesamiento de datos en tiempo real como BigQuery y Dataflow. Además, se enfoca en problemas de verificación de datos y	Vortex ofrece un enfoque innovador para la gestión y optimización del almacenamiento de datos, manteniendo una estructura distribuida mediante la integración de WOS (<i>Write-Optimized Storage</i>) y ROS (<i>Read-Optimized Storage</i>). El sistema garantizó la veracidad de los datos mediante un proceso continuo de verificación, y optimiza el almacenamiento mediante la compactación y la conversión entre generaciones. Además,	• Vortex: Motor de almacenamiento y procesamiento distribuido. • BigQuery: Herramienta de análisis de datos de Google. • Dataflow: Plataforma de procesamiento de datos basada en Apache Beam. • Colossus: Sistema de almacenamiento utilizado por Google.	Vortex demostró ser capaz de gestionar el tráfico de BigQuery en producción para conjuntos de datos a escala petabyte, con un rendimiento de múltiples gigabytes por segundo. Además, logró mantener la integridad de los datos mediante la verificación constante de las operaciones de escritura y lectura, y permitió la ejecución eficiente de consultas analíticas a

Artículo	Problema	Contribución	Tecnologías	Resultado
	optimización de tablas en entornos de alta escala.	proporcionó capacidades de eliminación de particiones y manejo de mutaciones de datos (como UPDATE, DELETE y MERGE), con un enfoque específico en la integración de flujos de datos en BigQuery y Dataflow.	• Apache Iceberg: Formato utilizado para BigLake Managed Tables. • Parquet: Formato de almacenamiento de datos.	gran escala, garantizando la correcta integración y optimización del almacenamiento de datos.
Batalla et al. [76]	El problema abordado en el artículo es la necesidad de estandarizar los datos no estructurados, cargarlos en el sistema SAP BW/4HANA, y permitir que los usuarios mantengan y actualicen los datos de manera estandarizada utilizando consultas listas para la	La contribución principal del artículo es la implementación de SAP Lumira Designer como herramienta para crear aplicaciones analíticas interactivas y visualizaciones en el contexto de un sistema BI basado en SAP BW/4HANA. Se detallan las configuraciones clave necesarias para implementar consultas listas para la	• SAP BW/4HANA • SAP Lumira Designer • OLAP • Consultas y Agregaciones: Usadas para manejar la entrada y salida de datos estructurados a través de la interfaz	Creación de un sistema de BI interactivo: Con paneles de control, KPIs, y consultas listas para la entrada que permiten a los usuarios explorar y analizar datos en tiempo real. Mejora en la visualización de datos: Se personalizaron más de 20 tipos diferentes de

Artículo	Problema	Contribución	Tecnologías	Resultado
	entrada, todo dentro de un sistema de <i>Business Intelligence</i> (BI) que facilite la visualización y análisis de datos. La necesidad de crear un flujo de trabajo eficiente para manejar grandes volúmenes de datos dentro de un sistema empresarial también se presenta como un desafío clave.	entrada, paneles de control, KPIs, y análisis OLAP (<i>Online Analytical Processing</i>), con el objetivo de facilitar la toma de decisiones basadas en datos estructurados y no estructurados. También se resalta cómo el uso de componentes visuales (como gráficos, tablas y filtros dinámicos) ayuda a los usuarios a interactuar con los datos de manera eficiente.	de SAP Lumira Designer. JavaScript	gráficos (como <i>Heat Maps</i> , Gráficos de Barras, Columnas, y Donuts) y se creó un <i>crosstab</i> para la entrada de datos transaccionales. Mejoras en la eficiencia de la gestión de datos: La integración entre SAP Lumira Designer y SAP BW/4HANA permitió automatizar y mejorar los procesos de carga y actualización de datos.
Hoseini et al. [77]	El manejo de grandes volúmenes de datos en entornos industriales requiere una gestión eficiente de metadatos y la integración de	SEDAR (<i>Semantic Data Reservoir</i> , Depósito de datos semánticos) proporciona una solución integral para gestionar y transformar datos	- PLASMA - MLFlow - HDFS - Postgres - MongoDB - SparkSQL	El sistema facilitó la ingestión y transformación de datos, asegurando la calidad y la semántica mediante la integración de

Artículo	Problema	Contribución	Tecnologías	Resultado
	diferentes fuentes de datos, a menudo con estructuras complejas y formatos variados. Además, se enfrentan desafíos relacionados con la calidad de los datos, la gobernanza, la seguridad y la integración con aplicaciones de <i>machine learning</i> (ML) para procesos de optimización, como la mejora de formulaciones químicas.	de diversas fuentes utilizando un catálogo de datos que soporta la gestión de metadatos, el versionado, la trazabilidad de los cambios y el enlace entre los datos, esta implementó una arquitectura de trabajo basada en espacios de trabajo, ofreciendo control de acceso, encriptación y visualización semántica de los datos, utiliza un enfoque semántico mediante el uso de ontologías, y una integración de procesos de MLOps, utilizando herramientas como MLFlow y Jupyter Notebooks para el seguimiento de	• SPARQL • Tab2KG • SeMi • RML • OBDA • Squerall.	ontologías. A través de un flujo de trabajo automatizado, SEDAR permitió la creación de modelos semánticos de datos, el seguimiento de experimentos y la integración de resultados de ML. Además, la plataforma habilitó la búsqueda eficiente de <i>datasets</i> y soportó la gobernanza de datos, garantizando su accesibilidad y la implementación de medidas de privacidad.

Artículo	Problema	Contribución	Tecnologías	Resultado
		experimentos y despliegue de modelos.		

Los 50 trabajos se distribuyeron mayormente hacia el dominio médico independientemente del tipo de repositorio, pero en ningún caso se encontró que el tema principal estuviera enfocado en diabetes, tal como se muestra en la tabla 3.2. Esta investigación es de gran relevancia, ya que aborda una brecha crítica en la infraestructura de datos aplicada al manejo de enfermedades crónicas. Aunque diversos estudios han explorado componentes tecnológicos de forma aislada, como el almacenamiento o el procesamiento de datos en contextos médicos generales, ninguno ha propuesto una arquitectura integral orientada específicamente al análisis de la remisión de la diabetes tipo 2. Al diseñar una arquitectura de Data Lakehouse enfocada en este propósito, el estudio permite la integración y el análisis de datos multifuente a gran escala, con el fin de identificar los factores clave que influyen en la remisión. Este enfoque especializado tiene el potencial de apoyar la toma de decisiones clínicas, mejorar la personalización de los tratamientos y, en última instancia, optimizar los resultados en la salud de los pacientes, lo que convierte a este trabajo en una contribución significativa tanto para la ingeniería de datos como para el ámbito médico.

Tabla 3.2 Análisis comparativo de los trabajos analizados

Artículo	Repositorio	Dominio médico	Enfoque diabetes en
[43]	<i>Data Warehouse</i>	Sí	No
[44]	<i>Data Warehouse</i>	Sí	No
[45]	<i>Data Warehouse</i>	Sí	No
[46]	<i>Data Warehouse</i>	Sí	No
[47]	<i>Data Warehouse</i>	Sí	No
[48]	<i>Data Warehouse</i>	Sí	No
[49]	<i>Data Warehouse</i>	Sí	No
[50]	<i>Data Warehouse</i>	Sí	No
[51]	<i>Data Warehouse</i>	Sí	No
[52]	<i>Data Warehouse</i>	Sí	No
[53]	<i>Data Warehouse</i>	Sí	No
[54]	<i>Data Lake</i>	Sí	No
[55]	<i>Data Lake</i>	Sí	No
[56]	<i>Data Lake</i>	Sí	No

Artículo	Repositorio	Dominio médico	Enfoque diabetes en
[57]	<i>Data Warehouse</i>	Sí	No
[58]	<i>Data Warehouse</i>	Sí	No
[59]	<i>Data Warehouse</i>	No	No
[60]	<i>Data Warehouse</i>	Sí	No
[61]	<i>Data Warehouse</i>	Sí	No
[62]	<i>Data Warehouse</i>	Sí	No
[63]	<i>Data Warehouse</i>	Sí	No
[64]	<i>Data Warehouse</i>	Sí	No
[65]	<i>Data Lake</i>	Sí	No
[66]	<i>Data Lake</i>	Sí	No
[67]	<i>Big Data</i>	Sí	No
[68]	<i>Data Lakehouse</i>	Sí	No
[69]	<i>Data Lake</i>	Sí	No
[70]	<i>Data Lake</i>	Sí	No
[71]	<i>Big Data</i>	No	No
[72]	<i>Data Catalog</i>	No	No
[73]	<i>Data Lakehouse</i>	No	No
[74]	<i>Big Data</i>	No	No
[75]	<i>Data Warehouse</i>	No	No
[76]	<i>Data Warehouse</i>	No	No
[77]	<i>Data Lake</i>	No	No
[17]	<i>Data Analysis</i>	Sí	No
[18]	<i>Data Lakehouse</i>	No	No
[19]	<i>Data Lakehouse</i>	Sí	No
[20]	<i>Data Lakehouse</i>	Sí	No
[21]	<i>Data Lakehouse</i>	No	No
[22]	<i>Data Lake</i>	Sí	No
[23]	<i>Datasets</i>	Sí	No
[24]	<i>Big Data</i>	Sí	No
[25]	<i>Data Lakehouse</i>	No	No

Artículo	Repositorio	Dominio médico	Enfoque en diabetes
[26]	<i>Data Lakehouse</i>	No	No
[27]	<i>Data Lake</i>	Sí	No
[28]	<i>Data Lakehouse</i>	Sí	No
[29]	<i>Data Lake</i>	Sí	No
[30]	<i>Data Lakehouse</i>	Sí	No
[31]	<i>Data Lakehouse</i>	Sí	No
Este trabajo	<i>Data Lakehouse</i>	Sí	Sí

3.2 Desarrollo del Data Lakehouse

Para el desarrollo del *Data Lakehouse* se optó por utilizar la metodología de desarrollo DataOps la cual promueve la automatización de procesos, la colaboración entre equipos multidisciplinarios y la mejora continua en la gestión del ciclo de vida de los datos. Esta metodología toma principios de DevOps, agilidad e integración continua, y los aplica al mundo de los datos, permitiendo una entrega ágil y de calidad de productos analíticos.

3.2.1 Diseño de la aplicación

El diseño del *Data Lakehouse* propuesto se basa en una arquitectura en capas que integra tecnologías especializadas para garantizar la ingesta, el procesamiento, el almacenamiento y la visualización eficiente de datos estructurados, semi-estructurados y no estructurados. Gracias al uso combinado de Apache NiFi y Kafka, se logra capturar información desde fuentes heterogéneas como bases de datos relacionales, archivos JSON, imágenes o contenidos multimedia en redes sociales. Estos datos se procesan en tiempo real o por lotes mediante motores como Apache Spark y Flink, y se almacenan en Apache Iceberg, que proporciona un modelo transaccional escalable y orientado a consultas analíticas. Esta flexibilidad permite integrar flujos de datos complejos en un entorno unificado, eliminando la fragmentación entre data lakes y data warehouses tradicionales. Además, el diseño potencia la trazabilidad, la eficiencia en el análisis y la gobernanza de datos,

ofreciendo una plataforma robusta para proyectos de ciencia de datos e inteligencia organizacional.

3.2.1.1 Arquitectura

Asimismo, la arquitectura del *Data Lakehouse*, que se observa en la Figura 3.3, se organiza en una estructura por capas que permite aislar responsabilidades, facilitar el mantenimiento y asegurar la escalabilidad del sistema. Cada capa cumple un rol específico dentro del ciclo de vida del dato, desde su origen hasta su consumo, integrando herramientas especializadas para cada función.

La arquitectura integra fuentes heterogéneas y habilita la ingestión, procesamiento, almacenamiento y explotación analítica de información relacionada con la remisión de la diabetes tipo 2. El diseño contempla datos estructurados, semi-estructurados y no estructurados, con el propósito de soportar análisis multifuente y multiformato a gran escala.

Apache NiFi se utiliza como componente de ingesta e integración, orientado a la construcción de flujos de datos. Permite conectar múltiples fuentes, ejecutar transformaciones iniciales, aplicar reglas de ruteo y controlar el transporte de información bajo condiciones de carga variable. Asimismo, facilita la trazabilidad de los datos durante su tránsito hacia las capas de procesamiento.

Apache Kafka opera como plataforma de mensajería distribuida basada en eventos. Se emplea para desacoplar productores y consumidores, habilitar la transmisión de mensajes en tiempo real y garantizar escalabilidad y tolerancia a fallos. Su función principal consiste en centralizar la distribución de eventos y cargas de datos hacia los motores que ejecutan transformaciones y almacenamiento posterior.

En la capa de procesamiento se integran motores de cómputo distribuido. Apache Spark se utiliza como motor de procesamiento para tareas de transformación, limpieza, agregación y preparación de conjuntos de datos analíticos a gran escala. Su adopción permite ejecutar procesos batch y, cuando es requerido, flujos de procesamiento continuo, consolidando información en formatos adecuados para el Lakehouse.

De manera complementaria, Apache Flink se considera para escenarios que demandan procesamiento continuo de baja latencia y manejo de estado. Su capacidad resulta adecuada cuando el análisis requiere operaciones sobre flujos de

eventos con ventanas temporales, actualizaciones incrementales o procesamiento cercano al tiempo real.

La capa central del almacenamiento se soporta en Apache Iceberg, empleado como formato de tablas para el Lakehouse. Este componente proporciona administración de metadatos, soporte de evolución de esquemas, particionamiento eficiente y consistencia de lectura mediante snapshots. En consecuencia, Iceberg permite organizar el almacenamiento en un esquema tabular apto para cargas analíticas, manteniendo gobernanza y control sobre versiones y estructura.

En la capa de análisis y consulta se incorpora Trino como motor SQL distribuido. Su objetivo consiste en habilitar consultas interactivas y de alto desempeño sobre las tablas del Lakehouse, facilitando el acceso analítico a los datos almacenados en Iceberg y soportando integración con herramientas de visualización.

Adicionalmente, Apache Doris se incorpora como motor analítico orientado a cargas OLAP. Se emplea para mejorar el rendimiento en consultas agregadas y tableros que demandan tiempos de respuesta reducidos, complementando los mecanismos de consulta basados en SQL distribuido según los requerimientos del caso de uso.

En la capa de visualización se incluyen herramientas de inteligencia de negocios. Apache Superset se utiliza para la construcción de dashboards y visualizaciones interactivas basadas en consultas SQL. Su integración permite representar métricas, tendencias y relaciones relevantes sobre los datos consolidados, apoyando la exploración y análisis de factores asociados a la remisión.

Como alternativa de visualización, Metabase se incorpora para facilitar la exploración rápida y la elaboración de tableros mediante interfaces orientadas al análisis. Su inclusión permite diversificar los mecanismos de consumo de información, mejorando la accesibilidad de los resultados para distintos perfiles de usuario.

Finalmente, Apache Airflow se emplea como componente de orquestación. Su función consiste en programar y coordinar flujos de trabajo (DAGs) que integran las etapas de ingestión, procesamiento, carga a Iceberg y disponibilidad para consulta y visualización. Asimismo, Airflow aporta control de dependencias, reintentos, monitoreo de ejecución y trazabilidad operativa de los pipelines.

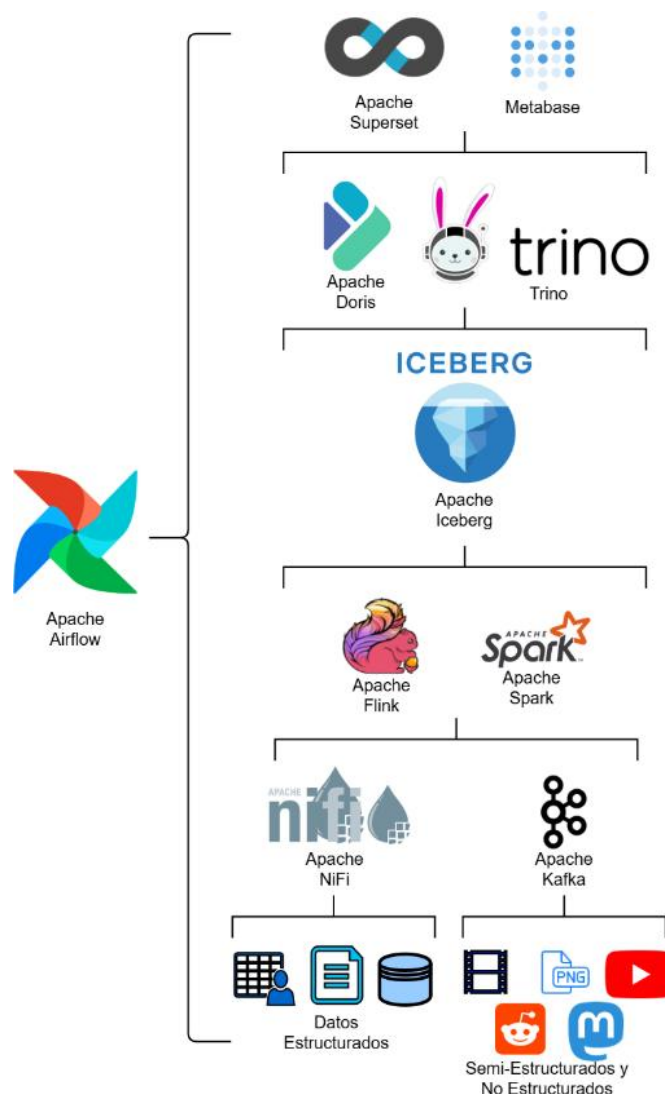


Figura 3.3 Arquitectura del Data Lakehouse

3.2.1.2 Modelado de la estructura

Como parte fundamental en la consolidación del *Data Lakehouse*, se diseñó el esquema lógico de almacenamiento, que se muestra en la Figura 3.4, que permite organizar, consultar y mantener la información extraída desde diversas fuentes heterogéneas. Dado que el sistema debe manejar datos estructurados, semiestructurados y no estructurados provenientes de archivos CSV, PNG, PDF, JPG, así como también textos, audios y videos, fue necesario establecer una estructura flexible, extensible y optimizada para análisis a gran escala. En este contexto, Apache Iceberg se utilizó como motor de almacenamiento transaccional distribuido, actuando como la columna vertebral de la arquitectura propuesta.

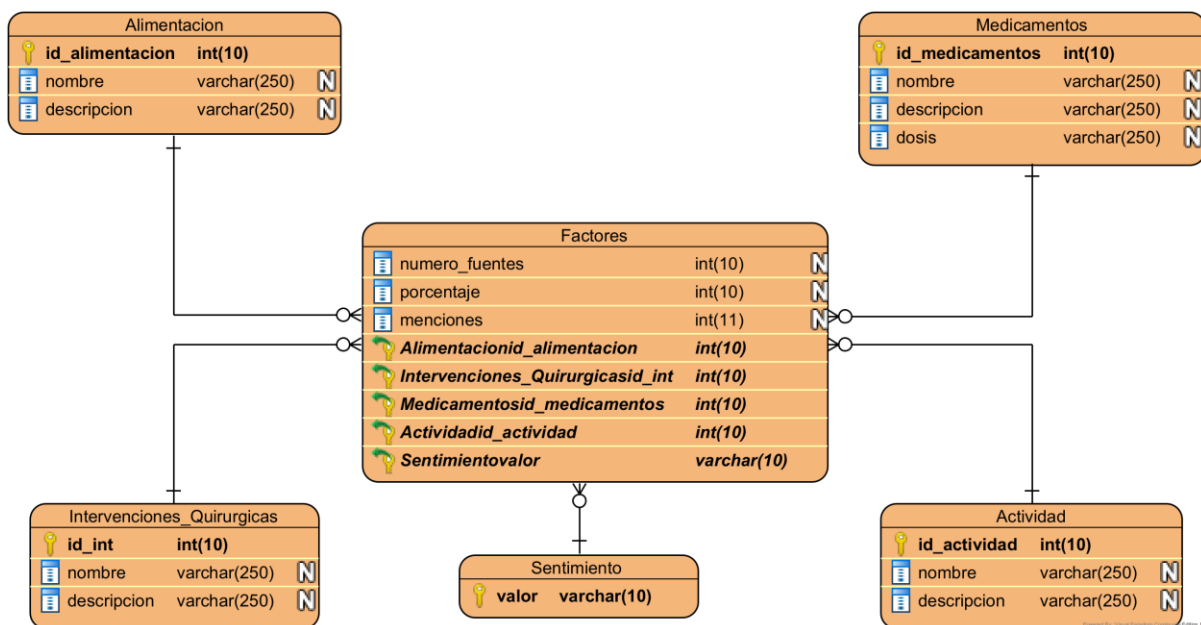


Figura 3.4 Diagrama relacional

El diseño del modelo se centra en la tabla principal denominada Factores, que funciona como tabla de hechos y que contiene métricas agregadas como el número de fuentes consultadas, el porcentaje de ocurrencia y el número de menciones identificadas por tipo de contenido. Esta tabla se relaciona con cinco dimensiones clave que permiten clasificar el contenido extraído según su naturaleza semántica:

- **Actividad:** agrupa menciones relacionadas con el ejercicio físico, movilidad y rutinas activas.
- **Alimentación:** recoge información vinculada a la dieta, hábitos alimenticios y consumo de nutrientes.
- **Intervenciones quirúrgicas:** identifica eventos quirúrgicos o procedimientos médicos relevantes.
- **Medicamentos:** agrupa términos referidos a fármacos, dosis y pautas de administración.
- **Sentimiento:** representa la polaridad emocional del contenido, categorizado como positivo, negativo o neutro.

Este diseño relacional permite mantener la normalización de los datos y habilita un análisis multidimensional de la información extraída, conservando la trazabilidad y facilitando el cruce de dimensiones para estudios más complejos. La elección de Iceberg como sistema de almacenamiento aporta ventajas como el control de versiones, las actualizaciones parciales (merge-on-read), la compatibilidad con motores analíticos como Trino y Spark, y el manejo eficiente de grandes volúmenes

de datos en formato columnar. Esta base estructural, adaptable y robusta, es esencial para la evolución futura de la plataforma, incluyendo mecanismos de enriquecimiento semántico, análisis longitudinal y personalización de resultados.

3.2.2 Despliegue

Para garantizar un entorno controlado, reproducible y escalable, el despliegue de todos los componentes del *Data Lakehouse* se realizó mediante contenedores Docker. Esta estrategia permitió encapsular las dependencias de cada herramienta, estandarizar su configuración y facilitar la orquestación del sistema completo en diferentes entornos de desarrollo o producción. A través de un archivo `docker-compose.yml`, se definieron los servicios necesarios, su red interna, los volúmenes persistentes y las variables de entorno requeridas para su correcto funcionamiento.

Cada una de las herramientas involucradas se desplegó como un contenedor independiente, pero conectado mediante una red virtual común que permite la comunicación eficiente entre servicios. Por ejemplo, Apache NiFi y Apache Kafka actúan como puerta de entrada para la ingesta de datos, transmitiendo información estructurada, semiestructurada y no estructurada que posteriormente es procesada por motores como Apache Spark y Apache Flink, también desplegados en contenedores. Estos motores transforman los datos y los escriben en tablas gestionadas por Apache Iceberg, cuyo almacenamiento subyacente se encuentra en MinIO, un sistema compatible con S3 también desplegado en contenedor.

Para el análisis, los motores Trino y Apache Doris acceden directamente a las tablas de Iceberg a través de catálogos configurados en tiempo de arranque, lo que permite ejecutar consultas SQL de forma distribuida y de alto rendimiento. Finalmente, los resultados se visualizan mediante *dashboards* interactivos en Apache Superset y Metabase, ambos conectados al motor de consultas mediante la red interna de Docker.

La orquestación de todos los procesos se encuentra bajo el control de Apache Airflow, el cual ejecuta flujos definidos para automatizar la ingesta, transformación y análisis de datos. Airflow se comunica con los servicios restantes mediante conectores y operadores personalizados, programando y monitoreando cada tarea de forma centralizada.

Este enfoque basado en contenedores no solo mejora la portabilidad y el aislamiento de servicios, sino que también facilita la escalabilidad horizontal del sistema y permite realizar pruebas, mantenimiento o actualizaciones de forma controlada sin comprometer la estabilidad del entorno completo, como se observa en la Figura 3.5 donde los contenedores están albergando las herramientas y comunicándolas entre sí.

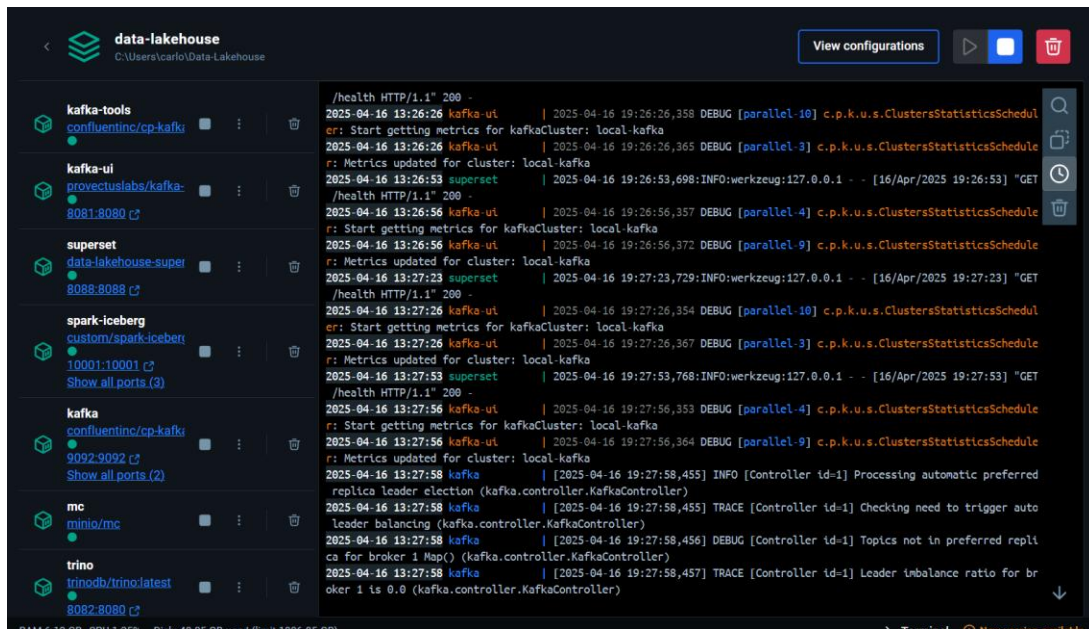


Figura 3.5 Estructura de contenedores en Docker

3.2.3 Ingesta

Para cubrir la necesidad de ingestión de datos en tiempo real, se desplegó el sistema de mensajería Apache Kafka mediante contenedores Docker, complementado por Zookeeper, requerido como coordinador de los brokers Kafka. Ambos servicios fueron definidos dentro de la red interna *iceberg_net* para asegurar una comunicación fluida y segura con el resto de los componentes del *Data Lakehouse*.

El servicio kafka expone los puertos 9092 y 9093, permitiendo tanto la comunicación interna entre servicios como la conexión desde el entorno anfitrión. La configuración define dos listeners: `PLAINTEXT://kafka:9092` para comunicación interna entre contenedores y `PLAINTEXT_EXT://localhost:9093` para interacción externa, lo que facilita el envío de datos desde *scripts* o clientes externos. Además, se establece el mapeo explícito de protocolos mediante `KAFKA_LISTENER_SECURITY_PROTOCOL_MAP`, habilitando así la correcta interpretación de los canales definidos.

Para la observación y monitoreo de los tópicos y mensajes enviados, se integró la herramienta kafka-ui, accesible desde el puerto 8081. Esta interfaz gráfica proporciona una vista completa del clúster Kafka, permitiendo visualizar y administrar tópicos, particiones, consumidores y mensajes individuales en tiempo real, en la Figura 3.6 se ilustra cómo los mensajes se despliegan dentro de la interfaz. Esta funcionalidad ha sido clave para validar visualmente la llegada de mensajes provenientes de flujos de ingesta (como PyMuPDF o DeepSeek) y confirmar la correcta estructuración de los datos antes de ser procesados por Spark.

El contenedor kafka-tools fue añadido para facilitar la ejecución de comandos administrativos y pruebas directas sobre el broker Kafka, permaneciendo activo en segundo plano mediante el comando sleep infinity. Esta configuración brinda flexibilidad para depuración y administración avanzada del sistema sin necesidad de salir del entorno Docker.

En conjunto, esta infraestructura permite establecer flujos de datos continuos y fiables que conectan orígenes semiestructurados y no estructurados con el resto de la arquitectura analítica, posicionando a Kafka como el eje central de la ingesta distribuida y asincrónica del sistema.

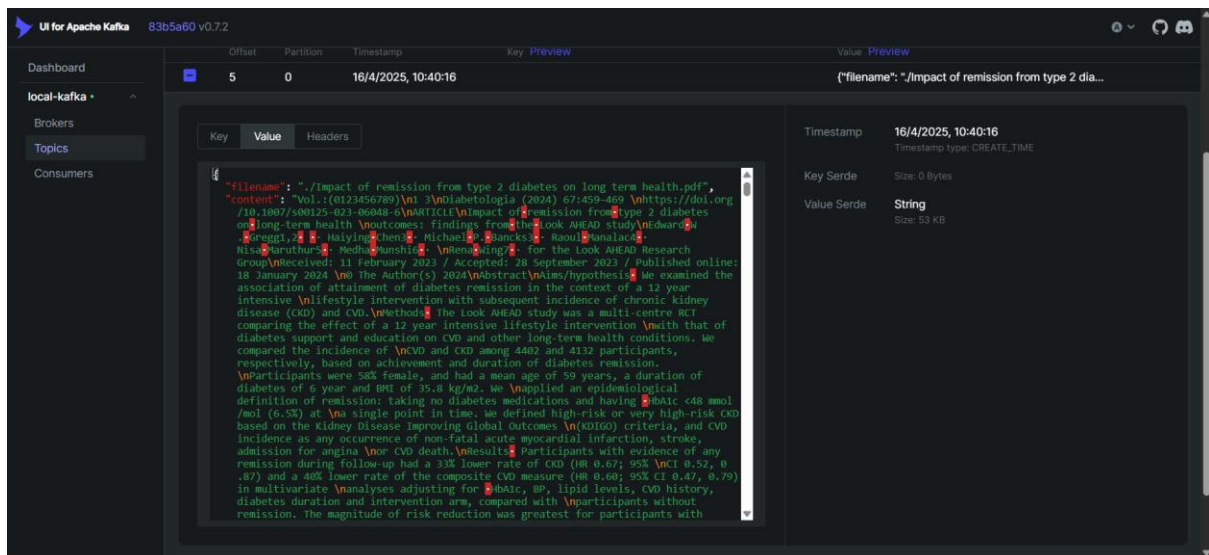


Figura 3.6 Interfaz de Kafka leyendo un mensaje

3.2.4 Procesamiento y Almacenamiento

El despliegue del componente de procesamiento y almacenamiento se realizó mediante un contenedor personalizado denominado spark-iceberg, construido a partir de la imagen base de Apache Spark e incluyendo configuraciones específicas para el uso de Apache Iceberg como sistema de almacenamiento transaccional. Este

contenedor se ejecuta en el puerto 8080 para la interfaz de Spark y en el puerto 8888 para acceso a notebooks en Jupyter, lo cual facilita la experimentación directa con consultas SQL sobre tablas Iceberg.

Durante la inicialización, se monta el *script* `spark_iceberg.sql` el cual se ilustra en la Figura 3.7 y su aplicación dentro del contenedor se muestra en la figura 3.8, ubicado en el directorio compartido `./warehouse`, el cual contiene las instrucciones SQL necesarias para la creación de las tablas Iceberg a partir del modelo relacional representado en el diagrama conceptual. Dicho *script*, orientado a la sintaxis específica de Iceberg y ejecutado desde consola, traduce el diseño lógico del sistema en estructuras físicas optimizadas para análisis distribuido. Esta operación puede realizarse directamente desde un notebook Jupyter integrado en el contenedor o a través de la consola interactiva de PySpark, aprovechando la inclusión de los paquetes necesarios para operar con Iceberg y Kafka mediante la variable `PYSPARK_SUBMIT_ARGS`.

```
-- Elimina las tablas si ya existen
DROP TABLE IF EXISTS iceberg.default.factorres;
DROP TABLE IF EXISTS
iceberg.default.medicamentos;
DROP TABLE IF EXISTS
iceberg.default.intervenciones_quirurgicas;
DROP TABLE IF EXISTS
iceberg.default.alimentacion;
DROP TABLE IF EXISTS iceberg.default.actividad;
DROP TABLE IF EXISTS
iceberg.default.sentimiento;
-- Dimensiones
CREATE TABLE iceberg.default.actividad (
  id_actividad INT,
  nombre STRING,
  descripcion STRING
) USING iceberg;

CREATE TABLE iceberg.default.alimentacion (
  id_alimentacion INT,
  nombre STRING,
  descripcion STRING
) USING iceberg;

CREATE TABLE
iceberg.default.intervenciones_quirurgicas (
  id_int INT,
  nombre STRING,
  descripcion STRING
) USING iceberg;

CREATE TABLE iceberg.default.medicamentos (
  id_medicamentos INT,
  nombre STRING,
  descripcion STRING,
  dosis STRING
) USING iceberg;

CREATE TABLE iceberg.default.sentimiento (
  valor STRING
) USING iceberg;

-- Hechos
CREATE TABLE iceberg.default.factorres (
  numero_fuentes INT,
  porcentaje INT,
  menciones INT,
  id_alimentacion INT,
  id_int INT,
  id_medicamentos INT,
  id_actividad INT,
  sentimiento_valor STRING
) USING iceberg;
```

Figura 3.7 Código Spark-Iceberg

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\carlo\Data-Lakehouse> docker exec -it spark-iceberg /opt/spark/bin/spark-sql -f /home/iceberg/notebooks/note
books/EstructuraIceberg.sql
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/04/24 00:38:51 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
25/04/24 00:38:51 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark Web UI available at http://9669b1a8dde3:4041
Spark master: local[*], Application Id: local-1745455131923
Time taken: 1.718 seconds
Time taken: 0.023 seconds
Time taken: 0.024 seconds
Time taken: 0.025 seconds
Time taken: 0.02 seconds
Time taken: 0.018 seconds
Time taken: 1.176 seconds
Time taken: 0.065 seconds
Time taken: 0.066 seconds
Time taken: 0.065 seconds
Time taken: 0.063 seconds
Time taken: 0.068 seconds
PS C:\Users\carlo\Data-Lakehouse> |
```

Figura 3.8 Ejecución del código dentro de la consola

El almacenamiento subyacente se realiza a través de MinIO, configurado como un servicio S3-compatible dentro del ecosistema Docker, y accesible desde el contenedor spark-iceberg mediante las credenciales definidas como variables de entorno. La visualización del contenido almacenado es posible desde la consola web de MinIO expuesta en el puerto 9001, donde se observa la creación automática del *bucket warehouse*, gestionado desde el contenedor mc con políticas de acceso público preestablecidas.

Estos *buckets* se muestran como una organización de carpetas dentro de la interfaz tal como se ve en la Figura 3.9.

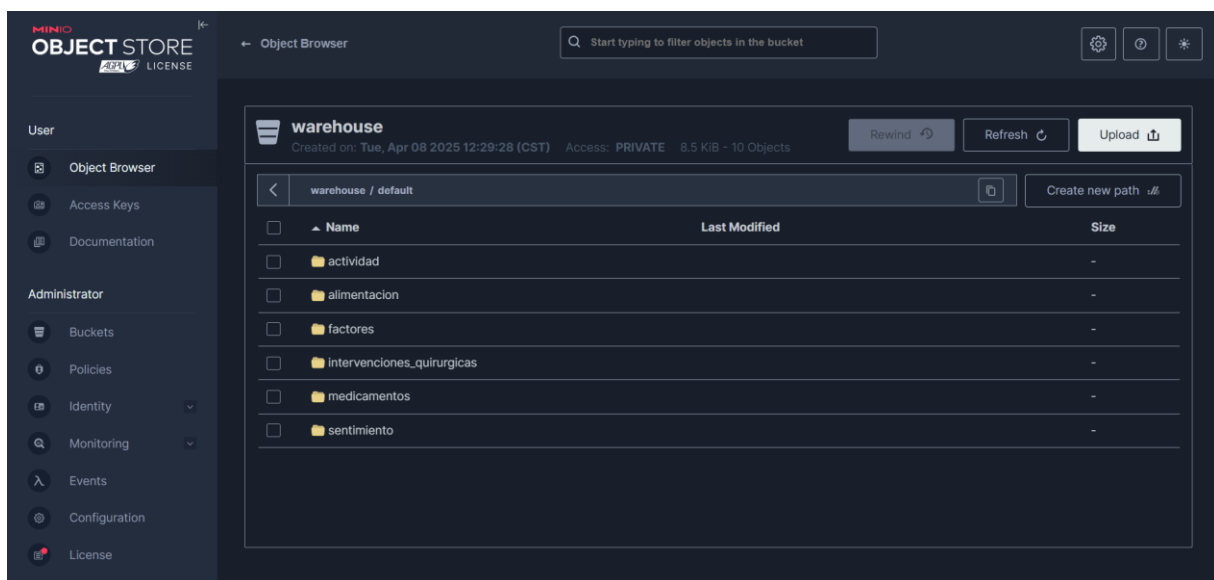


Figura 3.9 Interfaz de Minio S3

La persistencia de los datos queda garantizada gracias al volumen montado `./minio-data:/data`, que asegura la conservación del contenido del *bucket* incluso tras reinicios del sistema. Esta integración entre Spark, Iceberg y MinIO permite ejecutar flujos de procesamiento ETL robustos y consultar las tablas resultantes mediante motores analíticos como Trino o visualizarlas desde herramientas como Apache Superset. En conjunto, esta configuración constituye la espina dorsal del *Data Lakehouse*, habilitando operaciones de lectura, escritura y actualización incremental de forma transaccional sobre grandes volúmenes de datos heterogéneos.

3.2.5 Análisis

Con el objetivo de habilitar consultas analíticas de alto rendimiento sobre datos distribuidos, se integró Trino (anteriormente conocido como PrestoSQL) como motor de consulta principal dentro del ecosistema del *Data Lakehouse*. Su despliegue se realizó mediante contenedores Docker, exponiendo el servicio en el puerto 8082 y conectándolo a través de la red `iceberg_net`, lo que permite una comunicación directa con los servicios de almacenamiento y visualización.

Trino se configuró con múltiples archivos de propiedades montados desde el entorno anfitrión:

- `config.properties`, que define los parámetros generales del servidor Trino.
- `node.properties`, que identifica el nodo del clúster.

La carpeta `catalog`, que contiene los archivos de conexión a catálogos específicos como Iceberg, definidos con acceso a MinIO como backend S3.

Dentro de esta configuración, se establecieron las credenciales de acceso (`AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`) y el endpoint de MinIO (`CATALOG_S3_ENDPOINT=http://minio:9000`), asegurando la correcta integración con el almacén de datos `s3://warehouse/`. Esta configuración permite que Trino acceda y ejecute consultas SQL directamente sobre las tablas administradas por Apache Iceberg, sin necesidad de mover los datos ni realizar transformaciones intermedias, estas operaciones se muestran en el *dashboard* de trino descrito por la Figura 3.10.

La arquitectura de Trino favorece la separación entre procesamiento y almacenamiento, ejecutando consultas federadas incluso sobre múltiples catálogos si fuera necesario, muestra de ello es la Figura 3.11 la cual ejecuta una consulta sobre un catálogo. Esto lo convierte en una herramienta clave para explorar datos estructurados y semiestructurados de forma eficiente y en paralelo, aprovechando su compatibilidad con JDBC para integrarse con herramientas externas como Apache Superset y Metabase.

En resumen, el despliegue de Trino proporciona una capa de acceso analítico flexible y de alto rendimiento, optimizada para entornos distribuidos y transaccionales como el que se construye con Apache Iceberg, fortaleciendo así la capacidad exploratoria del *Data Lakehouse*.

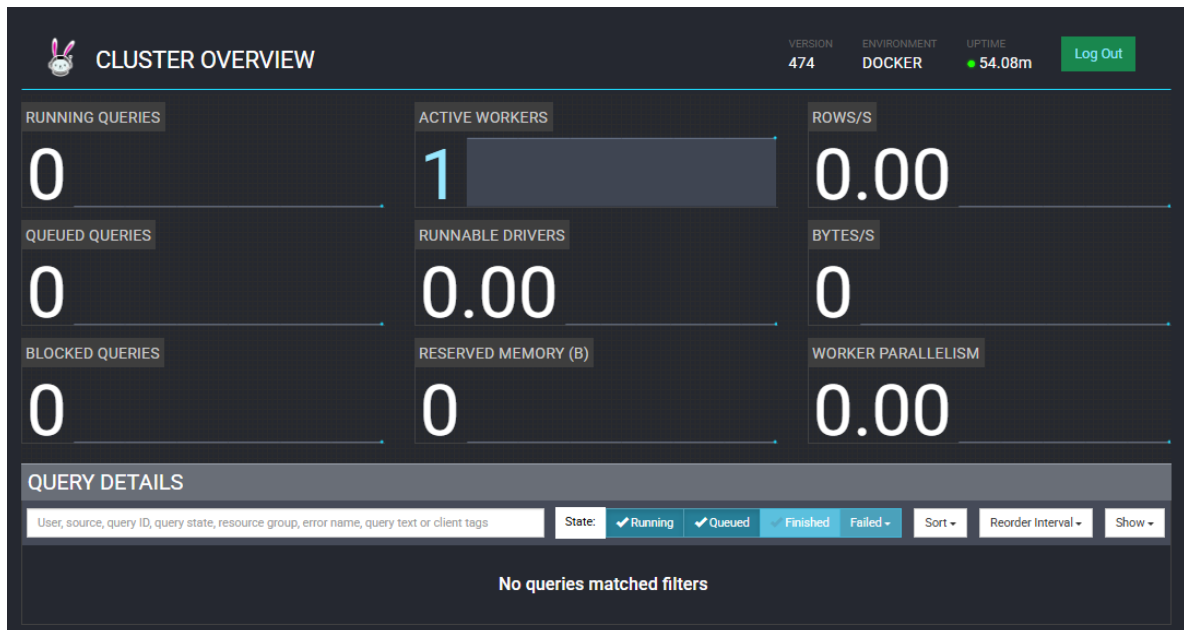


Figura 3.10 Interfaz de Trino

```
trino> SHOW SCHEMAS FROM iceberg;
Schema
-----
default
information_schema
(2 rows)

Query 20250424_011443_00002_2zc4g, FINISHED, 1 node
Splits: 11 total, 11 done (100.00%)
0.42 [2 rows, 35B] [4 rows/s, 84B/s]

trino> SHOW TABLES FROM iceberg.default;
Table
-----
actividad
alimentacion
factores
intervenciones_quirurgicas
medicamentos
sentimiento
(6 rows)

Query 20250424_011501_00003_2zc4g, FINISHED, 1 node
Splits: 11 total, 11 done (100.00%)
0.26 [6 rows, 180B] [22 rows/s, 684B/s]
```

Figura 3.11 Consulta desde la consola de Trino

3.2.6 Visualización

Con el propósito de proporcionar una interfaz gráfica robusta para la exploración y visualización de datos, se integró Apache Superset como plataforma de *business intelligence* dentro de la arquitectura del *Data Lakehouse*. Superset se desplegó mediante un contenedor Docker personalizado definido a través del archivo `Dockerfile-superset`, permitiendo incluir configuraciones adicionales y *scripts* de inicialización en su entorno de ejecución.

El servicio se expone a través del puerto 8088, habilitando el acceso a su interfaz web desde el navegador como se aprecia en la Figura 3.12. Durante su arranque, el contenedor ejecuta una serie de comandos secuenciales que incluyen la actualización del esquema de la base de datos interna (`superset db upgrade`), la creación del usuario administrador (`superset fab create-admin`), la carga de ejemplos (opcional), y la inicialización del entorno (`superset init`). Estos comandos aseguran que Superset esté listo para su uso inmediatamente después del despliegue.

Para garantizar la persistencia de configuraciones y *dashboards*, se montaron volúmenes externos en los directorios `superset_home` y `pythonpath`, permitiendo que las configuraciones personalizadas sobrevivan a reinicios o reconstrucciones del contenedor.

La conexión con los datos almacenados se establece a través de Trino, configurado como fuente de datos mediante el driver SQLAlchemy compatible. Esto permite consultar de forma directa las tablas gestionadas por Apache Iceberg, accediendo al almacén distribuido respaldado por MinIO. Gracias a esta integración, Superset es capaz de generar *dashboards* interactivos, gráficos de series temporales, mapas de calor, diagramas de dispersión, entre otros, con capacidad de filtrado en tiempo real y consultas *ad-hoc*.

La incorporación de Superset al ecosistema permite a los usuarios técnicos y no técnicos explorar los datos de manera intuitiva, identificar patrones, y comunicar hallazgos mediante visualizaciones dinámicas. Esto completa el ciclo de análisis del *Data Lakehouse*, facilitando la toma de decisiones basada en datos desde una interfaz accesible y altamente configurable.

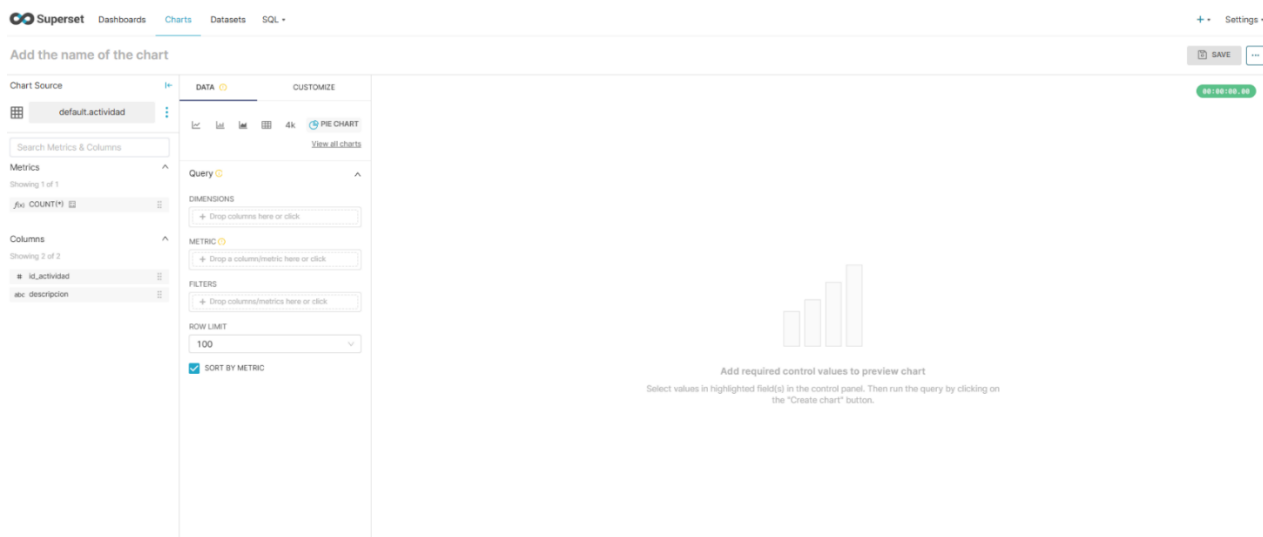


Figura 3.12 Interfaz gráfica de Apache Superset

3.2.7 Implementación de LLM's en ingesta

Con el objetivo de enriquecer el proceso de ingesta de datos no estructurados y dotarlo de capacidades semánticas, se integró un modelo de lenguaje de gran escala (LLM) denominado Deepseek R1, versión 7B (7 billones de parámetros), desplegado de forma local mediante el entorno de ejecución Ollama como se observa en la Figura 3.13 que consulta los modelos disponibles localmente. Esta solución permite realizar inferencias sin depender de servicios externos, garantizando así la privacidad de los datos y el control total del flujo de procesamiento.

```
Microsoft Windows [Versión 10.0.26100.3775]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\carlo>ollama list
NAME                ID                SIZE      MODIFIED
deepseek-llm:7b     9aab369a853b     4.0 GB    4 weeks ago
deepseek-r1:1.5b    a42b25d8c10a     1.1 GB    4 weeks ago

C:\Users\carlo>
```

Figura 3.13 Listado de modelos de Ollama en local

El modelo Deepseek R1 fue configurado con base en un *distilled model* derivado de Qwen, que preserva una alta capacidad de comprensión contextual, pero con una menor carga computacional, permitiendo su ejecución en entornos locales con recursos limitados. La implementación se realizó integrando un servidor Ollama en

escucha en el puerto 11434, desde donde se reciben solicitudes HTTP con el contenido extraído desde archivos PDF, texto libre, o mensajes provenientes de Kafka, como se observa en la Figura 3.14.

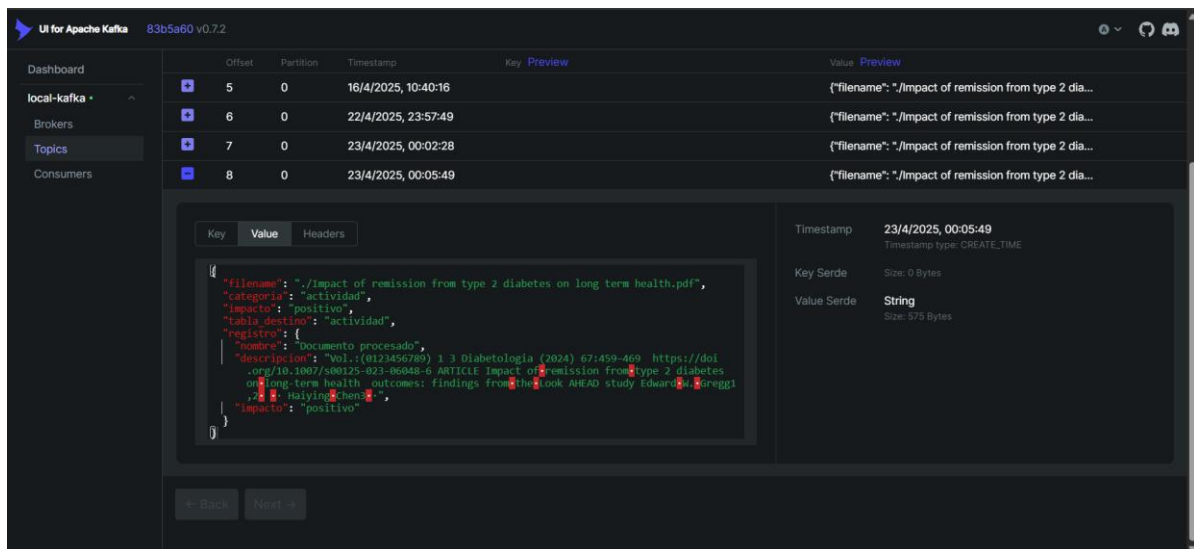


Figura 3.14 Kafka con el mensaje clasificado por Deepseek

Durante la ejecución del flujo de ingesta, los mensajes procesados (conteniendo texto plano) se envían al *endpoint* local de Ollama, donde el modelo Deepseek clasifica cada entrada según dos dimensiones:

- Categoría semántica: identificando si el contenido pertenece a una de las siguientes clases: Alimentación, Medicamento, Actividad, Intervenciones quirúrgicas.
- Sentimiento: analizando el impacto emocional del contenido y clasificándolo como positivo o negativo respecto al tratamiento de la condición médica analizada.

El *prompt* utilizado para esta clasificación se diseñó en español, orientado a maximizar la claridad de instrucciones para el modelo. Sin embargo, se consideró también la posibilidad de traducir las instrucciones al inglés en caso de observar mejoras de rendimiento debido al preentrenamiento del modelo. La respuesta generada por Deepseek se convierte en un objeto estructurado (JSON), que se almacena en tablas de dimensiones en Iceberg, permitiendo su análisis posterior mediante Trino o su visualización en Superset.

Esta integración representa un componente clave dentro del *Data Lakehouse*, al transformar datos textuales sin estructura en conocimiento explícito clasificable, lo

que potencia el análisis semántico de gran escala, especialmente en estudios relacionados con salud, emociones y efectos de intervenciones médicas.

Capítulo 4. Resultados

En este capítulo se presentan los resultados obtenidos a partir del diseño, implementación y explotación del Data Lakehouse propuesto a lo largo de esta tesis. De manera retrospectiva se retoma el problema original, la necesidad de analizar grandes volúmenes de datos heterogéneos relacionados con la remisión de la diabetes tipo 2 y otros factores clínicos. Se muestra cómo la arquitectura de *Data Lakehouse* permitió integrar, procesar y analizar dichas fuentes de información de forma unificada y reproducible.

4.1 Resultados de la investigación

Esta sección presenta los principales hallazgos de la revisión de la literatura e introduce la arquitectura propuesta. El análisis pone de manifiesto tendencias en el uso de repositorios y tecnologías de soporte, a la vez que identifica vacíos en los enfoques actuales. A partir de estos resultados, se propuso una arquitectura del *Data Lakehouse* diseñada para abordar dichas limitaciones y ofrecer un marco escalable y flexible para el estudio de la remisión de la diabetes tipo 2.

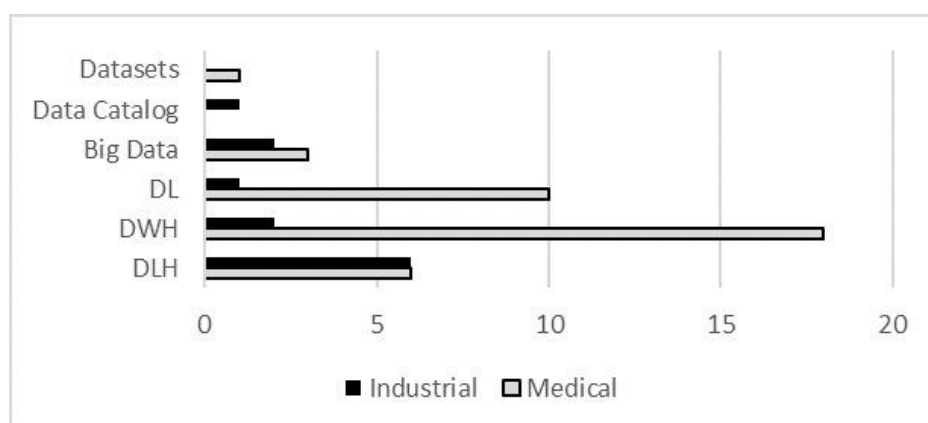


Figura 4.1 Distribución de repositorios por dominio

La Figura 4.1 ilustra cómo se utilizan distintos tipos de repositorios en cada dominio; destaca que el *Data Lakehouse* encabeza la gráfica, con un uso casi equivalente tanto en entornos médicos como industriales, lo que pone de relieve su amplio rango de

aplicaciones. En contraste, el *Data Warehouse* y el *Data Lake* se aplican predominantemente dentro del dominio médico.

En relación con la información presentada en la Figura 4.2, el eje horizontal muestra los años, mientras que la gráfica clasifica los distintos tipos de repositorios —DLH (*Data Lakehouse*), DWH (*Data Warehouse*) y DL (*Data Lake*)— por dominio. Los datos evidencian una tendencia al alza en el uso de los tres tipos de repositorios a partir de 2022. Los *Data Warehouse* enfocados en el dominio médico fueron los más utilizados durante 2022 [44], [47], [51], [53], [60], [63] y 2023 [45], [49], [52], [62]. Para 2024, el uso de *Data Lakehouse* [18], [25], [31], [73] y *Data Warehouse* [46], [48], [50], [57], [75], [76] se equilibró en mayor medida, mientras que el *Data Lake* dejó de emplearse por completo.

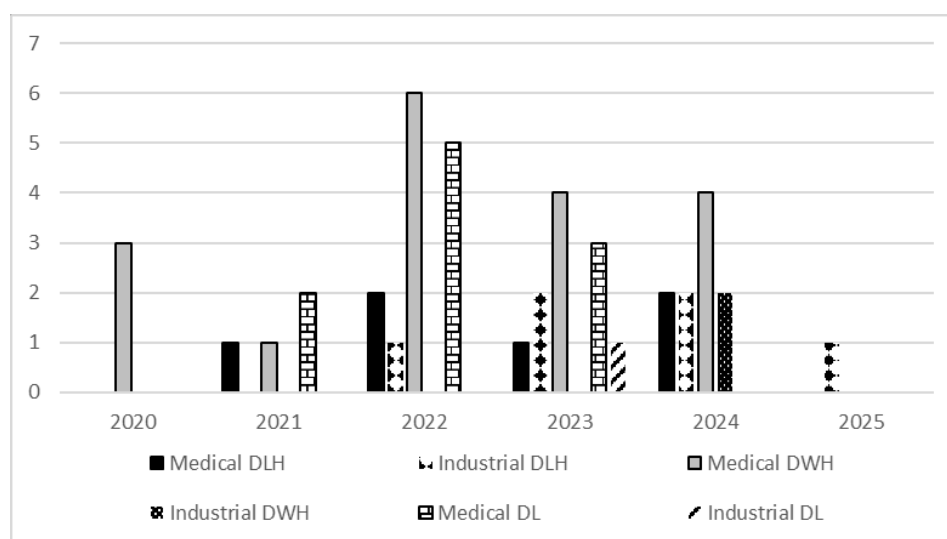


Figura 4.2 Distribución de repositorios por dominio por año

La Figura 4.3 muestra las tecnologías utilizadas para la integración de datos. Apache Kafka [17], [25], [30], [67] es una de las alternativas más empleadas para esta tarea, a menudo en combinación con Apache NiFi [65], [73]. Más allá de estas dos, aparece una variedad de otras herramientas, cuya selección depende del origen y la naturaleza de los datos considerados en cada estudio [25], [30], [55], [59], [75], [76].

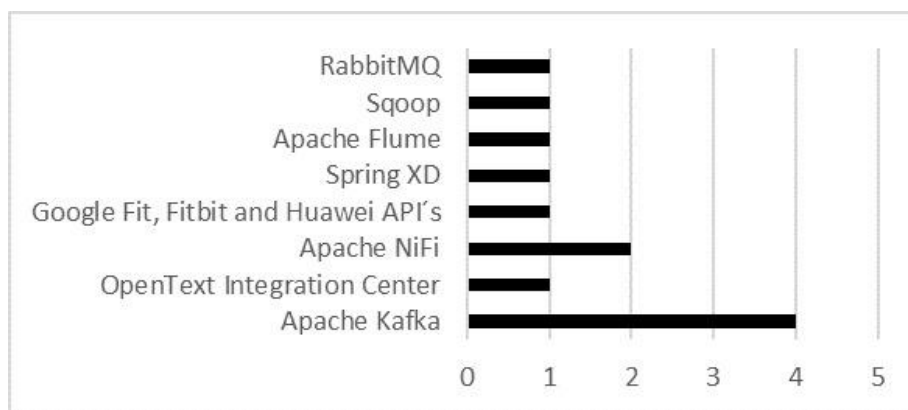


Figura 4.3 Distribución de uso de tecnologías de integración de datos

La Figura 4.4 ilustra la distribución de las tecnologías de procesamiento de datos utilizadas en los estudios revisados, donde Apache Spark sobresale de forma notable frente a sus competidores [17], [18], [21], [22], [26], [27], [28], [30], [59], [65], [67], [68], [71], [77]. Esta tecnología actúa como núcleo de procesamiento en el 56 % de los artículos que mencionan herramientas específicas de procesamiento, es decir, en 25 de los 50 estudios analizados. En algunos casos, como [17], [18] y [27], Spark se empleó en combinación con otras tecnologías, como Apache Flink o Apache Hudi, que también se encuentran entre las más utilizadas.

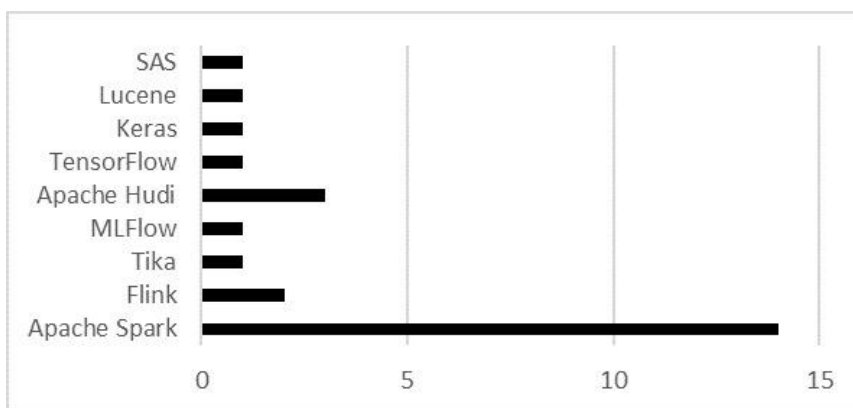


Figura 4.4 Distribución de uso de tecnologías de procesamiento de datos

La Figura 4.5 presenta las tecnologías de almacenamiento identificadas en los estudios analizados. Delta Lake [20], [25], [28], [29], [30], [59], [67] emergió como la solución más utilizada, valorada por su versatilidad y compatibilidad con diversas arquitecturas. HDFS (*Hadoop Distributed File System*) [21], [30], [70], [77] y Amazon S3 [18], [20], [59], [73] también se adoptaron ampliamente debido a su capacidad de escalabilidad y su integración con entornos de procesamiento de big data. Adicionalmente, se agrupó un conjunto de tecnologías menos frecuentes bajo la categoría "Otros". Esta categoría incluye una gama diversa de soluciones de

almacenamiento, como Google Sheets, Sybase, HBase, InfluxDB, Kudu, IPFS (*InterPlanetary File System*), Microsoft Azure Data Lake, SQLite3, i2b2, Oracle 11G, MIMIC-III (*Medical Information Mart for Intensive Care*), Apache Cassandra y Colossus, que se utilizaron en contextos específicos según la naturaleza de los datos y los objetivos de cada estudio.

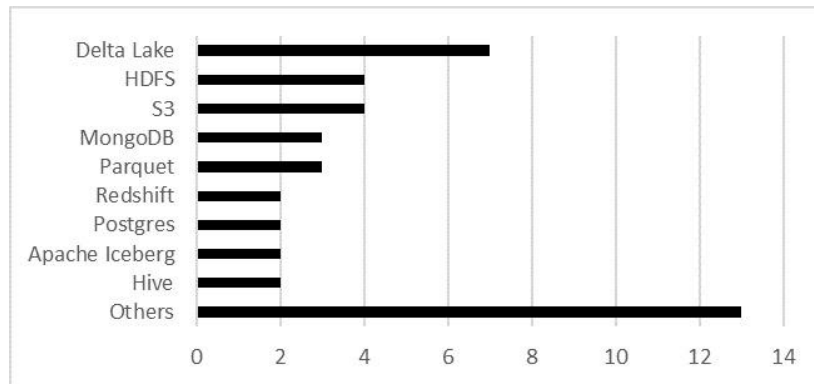


Figura 4.5 Distribución de uso de tecnologías de almacenamiento de datos

Asimismo, se observó un claro dominio de determinados proveedores de software a lo largo de las distintas etapas del flujo de datos. La Figura 4.6 muestra la distribución de tecnologías empleadas en los procesos de ingesta, procesamiento y almacenamiento de datos, destacando aquellas desarrolladas por la Apache Software Foundation frente a las ofrecidas por otros fabricantes. Entre los competidores más relevantes se encuentran Amazon Web Services (AWS), que provee servicios como AWS Glue y Amazon S3; Microsoft Azure, con herramientas como Azure Data Factory y Azure Synapse; y Databricks, conocida por su plataforma unificada basada en Apache Spark y Delta Lake.



Figura 4.6 Distribución de productores de software de herramientas para gestionar datos

En las etapas de análisis y visualización se identificó una amplia variedad de tecnologías a lo largo de los estudios revisados. Debido a la heterogeneidad de los trabajos analizados, el uso de estas herramientas se distribuyó de forma no uniforme,

adaptándose a las necesidades específicas de cada estudio. Esto pone de manifiesto la ausencia de un enfoque único y estandarizado para el análisis y la visualización; en cambio, las soluciones seleccionadas varían en función de los requisitos de cada dominio y de la infraestructura tecnológica disponible.

Tras una revisión exhaustiva de 50 artículos, no se identificaron trabajos que integraran explícitamente el enfoque propuesto en este estudio. Si bien algunos trabajos abordan componentes individuales como el almacenamiento, el procesamiento o la visualización en entornos de *Data Lakehouse*, ninguno propone una solución integral que combine estos elementos con el objetivo de descubrir factores que permitan la remisión de la diabetes tipo 2 dentro de un marco unificado.

4.2 Descripción final del esquema

A partir de los requerimientos funcionales y analíticos descritos en capítulos anteriores, se llevó a cabo un proceso iterativo de ajuste del esquema relacional con el objetivo de representar de forma integrada los factores asociados a la remisión de la diabetes tipo 2. La Figura 4.7 presenta el modelo resultante, en el que se organizan las entidades en torno a una tabla central denominada Factores, que actúa como tabla de hechos y concentra las relaciones con las distintas dimensiones clínicas y de estilo de vida.

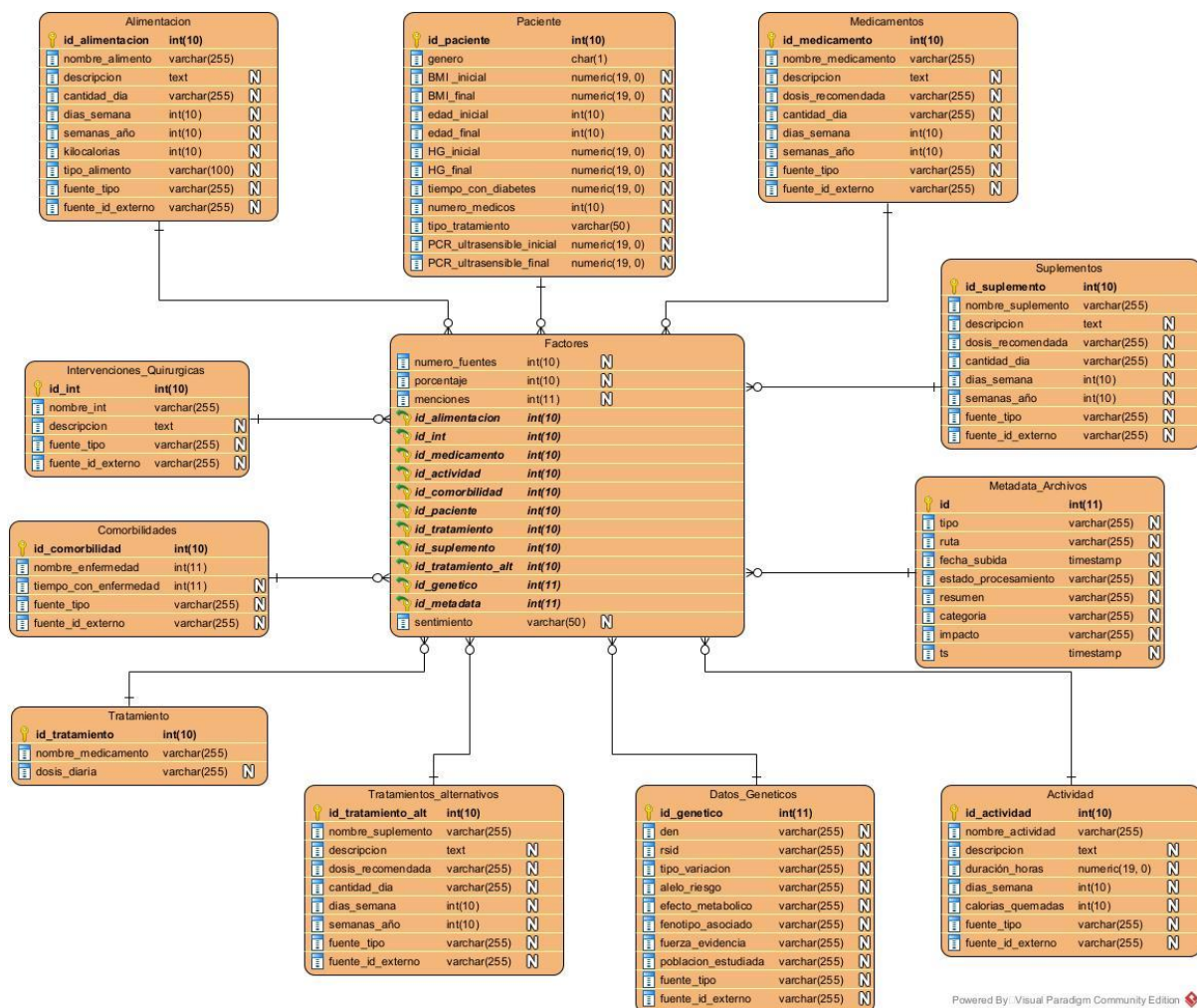


Figura 4.7 Distribución de uso de tecnologías de integración de datos

4.2.1 Tabla central de factores

La tabla Factores constituye el núcleo del modelo, ya que almacena la información agregada derivada del proceso de clasificación y extracción de conocimiento ejecutado sobre las fuentes textuales. Esta tabla integra, para cada combinación de paciente y factor identificado, los campos:

- *numero_fuentes*: cantidad de documentos en los que se detectó el factor.
- *porcentaje*: peso relativo del factor respecto del total de menciones asociadas al paciente.
- *menciones*: número de menciones explícitas del factor en los textos procesados.
- Una serie de claves foráneas hacia las tablas de contexto: *id_alimentacion*, *id_int*, *id_med medicamento*, *id_actividad*, *id_comorbilidad*, *id_paciente*, *id_tratamiento*, *id_suplemento* e *id_tratamiento_alt*.

- Campos analíticos adicionales como sentimiento, que captura la polaridad (positiva, negativa o neutra) asociada al factor, e *id_metadata*, que permite vincular cada registro con la tabla de metadatos de documentos en el *Data Lakehouse*.

Con este diseño, la tabla Factores se comporta como un punto de convergencia que relaciona el paciente con los distintos tipos de intervenciones, condiciones y hábitos identificados, manteniendo al mismo tiempo la trazabilidad hacia las fuentes originales.

4.2.1.1 Dimensión paciente

La entidad Paciente almacena las variables clínicas y demográficas que sirven como contexto para el análisis de remisión, tales como sexo (genero), índice de masa corporal inicial y final (BMI_inicial, BMI_final), edad, valores de hemoglobina glucosilada (HGI_inicial, HGI_final), tiempo de evolución de la diabetes, número de médicos tratantes y marcadores bioquímicos como la proteína C reactiva ultrasensible (PCR_ultrasensible_inicial, PCR_ultrasensible_final). La clave primaria id_paciente se utiliza como referencia en Factores, lo que permite relacionar cada registro de factor con el paciente correspondiente.

4.2.1.2 Alimentación, actividad física e intervenciones quirúrgicas

Las entidades Alimentación y Actividad representan, respectivamente, los patrones de consumo alimentario y las intervenciones relacionadas con actividad física. Cada tabla incluye un identificador propio (id_alimentacion, id_actividad), el nombre del alimento o actividad, una descripción detallada y atributos cuantitativos como cantidad_dia, dias_semana, semanas_año, duracion_horas o calorías_quemadas.

La tabla Intervenciones_Quirurgicas modela los procedimientos de tipo quirúrgico, con campos como nombre_int, descripcion, fuente_tipo y fuente_id_externo. Estas tres entidades se vinculan con Factores mediante claves foráneas, de modo que cada factor registrado puede asociarse a un alimento, una actividad o una intervención quirúrgica concretos, cuando aplique.

4.2.1.3 Medicamentos, suplementos y tratamientos

Uno de los principales ajustes del esquema consistió en separar explícitamente los medicamentos de los suplementos, manteniendo estructuras paralelas para ambos

tipos de intervención. La tabla Medicamentos y la tabla Suplementos comparten campos como:

- Identificador (id_medicamento / id_suplemento).
- nombre_medicamento / nombre_suplemento.
- descripcion.
- dosis_recomendada.
- cantidad_dia, dias_semana y semanas_año.
- Metadatos de origen (fuente_tipo, fuente_id_externo).

Esta separación permite distinguir entre fármacos y productos no farmacológicos, sin perder homogeneidad en el tratamiento analítico de las dosis y frecuencias.

Por otra parte, la tabla Tratamiento captura el esquema terapéutico principal de cada paciente, con campos como id_tratamiento, nombre_medicamento y dosis_diaria. De forma complementaria, Tratamientos_alternativos almacena intervenciones no convencionales o adicionales (por ejemplo, suplementos específicos o terapias complementarias), con una estructura similar a la de Suplementos. Ambos tipos de tratamiento pueden relacionarse con los registros de la tabla Factores mediante las claves foráneas id_tratamiento e id_tratamiento_alt.

4.2.1.3 Comorbilidades

La entidad Comorbilidades modela la presencia de enfermedades asociadas que pueden influir en la evolución de la diabetes y en su posible remisión. Incluye un identificador (id_comorbididad), el nombre de la enfermedad, el tiempo de evolución (tiempo_con_enfermedad) y campos de trazabilidad hacia las fuentes (fuente_tipo, fuente_id_externo). A través de la clave foránea correspondiente, las comorbilidades participan también en la combinación de factores registrada en la tabla Factores.

4.2.1.4 Trazabilidad y vinculación con el Data Lakehouse

De forma transversal, la mayoría de las tablas incluyen los campos fuente_tipo y fuente_id_externo, que permiten registrar el origen de la información (por ejemplo, artículo científico, publicación en redes sociales, registro clínico estructurado) y el identificador externo en el Data Lakehouse. Esta decisión de diseño se alinea con la arquitectura propuesta: los datos brutos se almacenan en MinIO, se catalogan mediante Iceberg y se enriquecen a través de los *pipelines* de procesamiento.

El campo *id_metadata* en la tabla Factores refuerza esta trazabilidad, al enlazar cada factor con el documento específico del que se extrajo la información, lo que facilita auditorías, reprocesamientos y análisis longitudinales según la fuente.

4.2.1.5 Justificación de los ajustes

En conjunto, los ajustes realizados sobre el esquema persiguen tres objetivos principales:

- Normalización y flexibilidad analítica
La separación en tablas especializadas (alimentación, actividad, medicamentos, suplementos, comorbilidades, tratamientos y procedimientos quirúrgicos) evita redundancias y permite incorporar nuevos tipos de intervenciones o factores sin reestructurar el modelo completo.
- Integración de resultados de minería de texto y LLM
La inclusión de campos como *numero_fuentes*, *menciones*, *porcentaje* y *sentimiento* en Factores permite almacenar de forma estructurada los resultados del proceso de clasificación automática y análisis semántico, habilitando consultas complejas sobre la contribución de cada factor a la remisión de la diabetes tipo 2.
- Trazabilidad de extremo a extremo
Los campos de metadatos y las claves hacia la tabla de metadatos del *Data Lakehouse* garantizan que cada registro del esquema relacional pueda rastrearse hasta el documento original, lo que resulta esencial en contextos clínicos y de investigación, donde la transparencia y la reproducibilidad son requisitos fundamentales.

Este esquema relacional constituye, por tanto, la base estructurada sobre la cual se apoyan las consultas analíticas y los experimentos posteriores orientados a identificar patrones y factores asociados a la remisión de la diabetes tipo 2.

4.3 Presentación de *pipelines* de integración y clasificación de datos

4.3.1 *Pipelines* de clasificación de información

Como parte de la implementación de la arquitectura de *Data Lakehouse*, se diseñaron y desplegaron dos *pipelines* principales de clasificación automática de información, orquestados mediante Apache Airflow:

- *Pipeline* de clasificación de documentos PDF
- *Pipeline* de clasificación de publicaciones de Mastodon

Ambos procesos comparten la misma filosofía de diseño: desacoplar la ingesta de datos de su procesamiento semántico, aprovechar servicios de mensajería basados en Kafka y persistir los resultados en tablas Iceberg que alimentan posteriormente el modelo relacional descrito en la sección anterior.

4.3.1.1 *Pipeline* de clasificación de documentos PDF

El primer *pipeline* corresponde al DAG ``clasification_pdf_dag``, encargado de procesar los documentos en formato PDF almacenados en el *Data Lakehouse*.

El funcionamiento general del DAG se resume en las siguientes etapas:

1. Selección de archivos pendientes.

En cada ejecución, el DAG consulta la tabla ``metadata_archivos`` (almacenada en Apache Iceberg) y recupera únicamente aquellos registros cuyo estado indica que el archivo está pendiente de clasificación. Esta tabla actúa como catálogo operativo y contiene, entre otros campos, el identificador del archivo, la ruta física en MinIO, el tipo de documento (por ejemplo, ``pdf``) y el estado de procesamiento.

2. Descarga y preprocesamiento del PDF.

Para cada archivo seleccionado, se descarga el documento desde MinIO utilizando el identificador y la ruta almacenados en ``metadata_archivos``. Posteriormente, se aplica un proceso de extracción de texto y normalización, que incluye la conversión de páginas a imágenes cuando es necesario, OCR mediante Tesseract y limpieza de caracteres especiales. El objetivo es obtener una representación textual unificada, independiente del formato original del PDF.

3. Clasificación mediante modelo de lenguaje.

El texto resultante se envía a un modelo de lenguaje grande (LLM) ejecutado localmente a través de Ollama. El modelo aplica un *prompt* diseñado para identificar factores relevantes para la remisión de la diabetes tipo 2, tales como medicamentos, suplementos, patrones de alimentación, actividad física, comorbilidades o intervenciones quirúrgicas. Adicionalmente, se calcula un sentimiento asociado a cada factor (positivo, negativo o neutro) y, cuando es posible, se extraen atributos cuantitativos (por ejemplo, dosis recomendadas, frecuencia semanal o duración de las actividades).

La ejecución de este componente se apoya en la GPU disponible en el servidor, tal como se aprecia en la Figura 4.9, donde se muestra el consumo de memoria de la tarjeta gráfica durante el procesamiento de los documentos.

4. Persistencia en el *Lakehouse*.

Los resultados estructurados generados por el LLM se insertan en las tablas de factores y de dimensiones definidas en Iceberg, particularmente en la tabla Factores y en las tablas asociadas de Medicamentos, Suplementos, Alimentación, Actividad, Intervenciones_Quirurgicas y Comorbilidades, según el tipo de información detectada. De este modo, cada documento procesado contribuye a enriquecer el modelo relacional con nuevos registros vinculados a su correspondiente 'id_metadata'.

5. Actualización del estado de procesamiento.

Finalmente, el DAG actualiza el registro correspondiente en 'metadata_archivos', marcando el archivo como procesado (y almacenando, en su caso, mensajes de error o indicadores de reintento). Este mecanismo garantiza la idempotencia del *pipeline* y evita reprocesar documentos ya clasificados.

La combinación de Airflow, MinIO, Iceberg y el modelo de lenguaje local permite implementar un flujo completamente automatizado, reproducible y escalable para la clasificación semántica de literatura científica y otros documentos en PDF relacionados con la diabetes tipo 2.

4.3.1.2 *Pipeline* de clasificación de publicaciones de Mastodon

El segundo *pipeline* desarrollado se centra en datos provenientes de redes sociales, específicamente publicaciones de la plataforma Mastodon. Este flujo se estructura en torno a dos tópicos de Kafka: 'mastodon_raw', que almacena los mensajes en bruto

recuperados desde la API de Mastodon, y ``mastodon_classified``, donde se publican los resultados ya estructurados tras el proceso de clasificación.

El DAG correspondiente se compone, de forma simplificada, de las siguientes etapas:

1. Consumo de publicaciones desde Kafka.

La tarea principal del DAG inicializa un consumidor de Kafka asociado al tópico ``mastodon_raw``. Cada mensaje contiene la información de una publicación (texto, usuario, fecha, etiquetas, entre otros) previamente recopilada por un proceso de ingesta independiente. El DAG está configurado para procesar lotes de mensajes respetando un límite máximo configurable por ejecución, lo que permite controlar la carga de trabajo y el uso de recursos.

2. Limpieza y normalización del texto.

Antes de la clasificación, se aplica un preprocesamiento ligero sobre el contenido textual, eliminando URLs, menciones y caracteres irrelevantes, y normalizando el idioma cuando es necesario. Esta etapa busca reducir ruido y facilitar la tarea del modelo de lenguaje.

3. Clasificación semántica con LLM.

De manera similar al *pipeline* de PDF, el texto de cada publicación se envía al LLM ejecutado localmente. El *prompt* está adaptado al contexto de redes sociales y orientado a identificar si la publicación se refiere a dieta, medicación, actividad física, síntomas, comorbilidades u otros factores relacionados con la diabetes. El modelo devuelve, entre otros atributos, el tipo de factor, el sentimiento asociado y, cuando procede, entidades como nombres de medicamentos, suplementos o tipos específicos de intervención.

4. Publicación en Kafka y persistencia en Iceberg.

Los resultados estructurados se envían al tópico ``mastodon_classified`` para su consumo por procesos posteriores. En paralelo, el DAG persiste la información en las tablas de Iceberg correspondientes, utilizando la misma estructura de Factores y dimensiones que en el *pipeline* de PDF. De esta forma, las publicaciones de Mastodon y los documentos científicos terminan integrados en un modelo común, lo que habilita análisis comparativos entre evidencia científica y discurso en redes sociales.

5. Manejo de errores y trazabilidad.

Cada mensaje procesado incluye un identificador de origen que permite rastrear la publicación original en Mastodon. En caso de error durante la clasificación, el DAG registra la incidencia en los *logs* de Airflow (como se aprecia en la Figura 4.8 para el

caso del DAG de PDF) y puede reenviar el mensaje a una cola de reintentos o marcarlo como fallido según la política de tolerancia a fallos configurada.

4.3.2 Integración de ambos *pipelines* en la arquitectura

Los dos *pipelines* descritos se integran de manera natural en la arquitectura de *Data Lakehouse*. Ambos se apoyan en la misma infraestructura de orquestación (Airflow), mensajería (Kafka), almacenamiento (MinIO + Iceberg) y análisis (tablas relacionales sobre Iceberg, consultadas desde herramientas como DBeaver y Apache Superset). Esta integración permite combinar información procedente de literatura científica en PDF con señales extraídas de redes sociales, convergiendo en el mismo esquema relacional centrado en la tabla Factores. De este modo, el sistema no solo almacena datos heterogéneos, sino que los transforma en conocimiento estructurado que puede explotarse para identificar patrones y factores asociados a la remisión de la diabetes tipo 2.

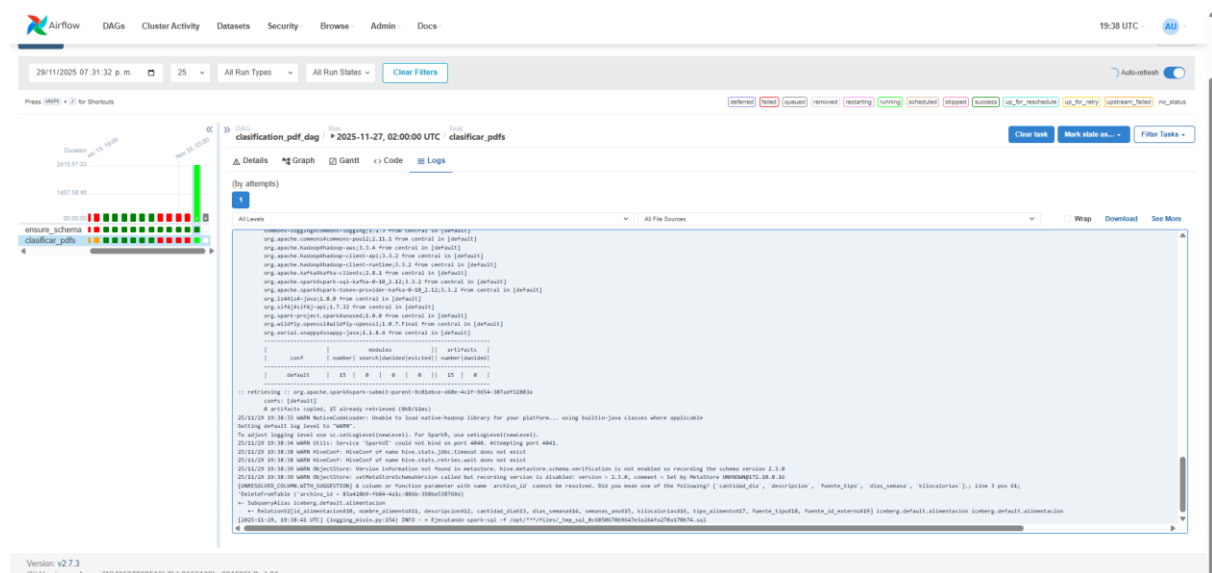


Figura 4.8 Interfaz de Apache Airflow para ejecución de DAG's

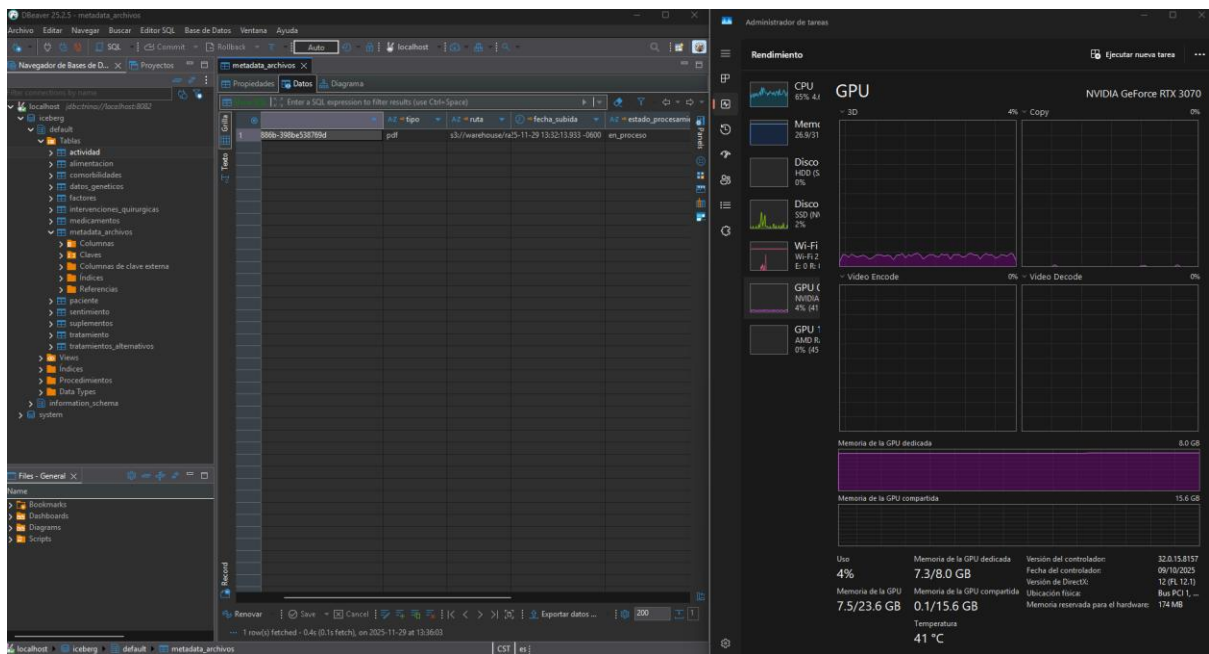


Figura 4.9 Clasificación de PDF's mediante LLM local con ejecución en GPU

4.4 Visualización de Resultados

Una vez consolidados los datos en el esquema relacional y expuestos a través del *Data Lakehouse*, se construyeron diversos paneles en Apache Superset para explorar de forma interactiva los factores asociados a la remisión de la diabetes tipo 2. Estas visualizaciones permiten resumir la información contenida en la tabla Factores y sus dimensiones, y constituyen el punto de partida para identificar correlaciones y patrones entre medicamentos, actividad física, comorbilidades e intervenciones quirúrgicas.

4.4.1. Distribución de medicamentos asociados a la remisión

La primera visualización corresponde a la Figura 4.10, donde una tabla muestra los nombres de los medicamentos o combinaciones farmacológicas identificadas (por ejemplo, metformina, dapagliflozin, combinaciones de análogos de GLP-1 con insulina, entre otros).

Este organizador grafico sintetiza qué fármacos aparecen con mayor frecuencia en los documentos y casos analizados como potencialmente relacionados con la remisión.

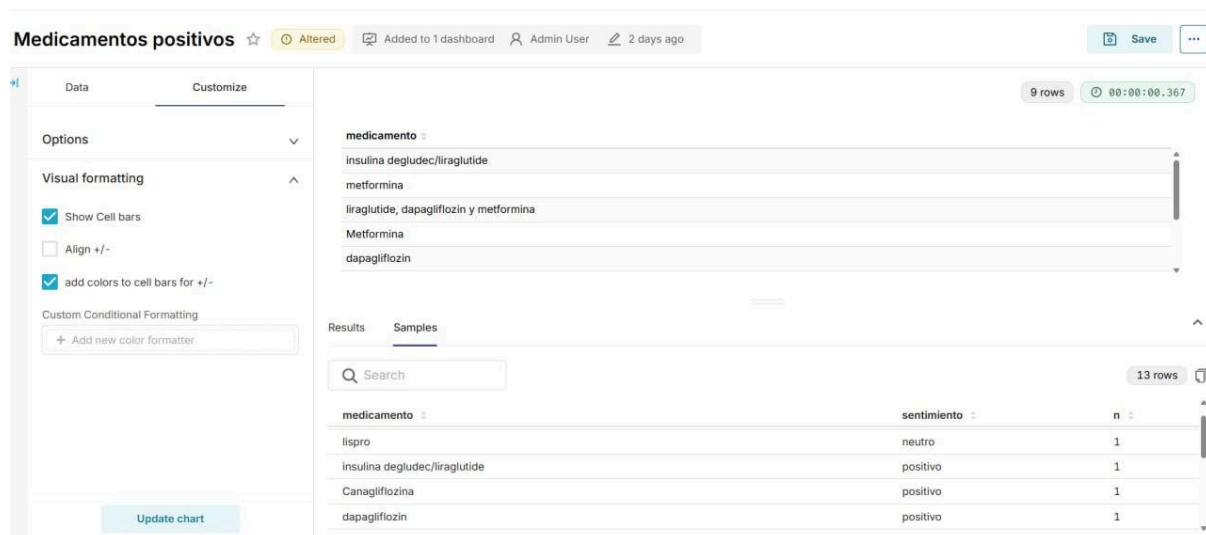


Figura 4.10 Consulta de medicamentos asociados positivamente a la remisión de diabetes tipo 2

Su importancia radica en que:

- Permite priorizar los medicamentos que concentran la mayor evidencia dentro del conjunto de estudios.
- Facilita la comparación entre monoterapias y tratamientos combinados, lo que puede sugerir sinergias o estrategias farmacológicas más efectivas.
- Sirve como base para análisis posteriores, donde estos recuentos pueden cruzarse con variables clínicas (por ejemplo, cambios en BMI o HbA1c) con el fin de estimar correlaciones entre determinados esquemas terapéuticos y la probabilidad de remisión.

4.4.2 Intervenciones de actividad física y estilo de vida

La segunda Figura, 4.11 es un gráfico de pastel que resume la distribución de las diferentes intervenciones de actividad física y estilo de vida registradas: programas estructurados de pérdida de peso, ejercicio combinado de resistencia y *endurance*, ejercicio moderado, incremento en la actividad física diaria, entrenamientos aeróbicos y de fuerza, así como intervenciones intensivas de cambio de estilo de vida.

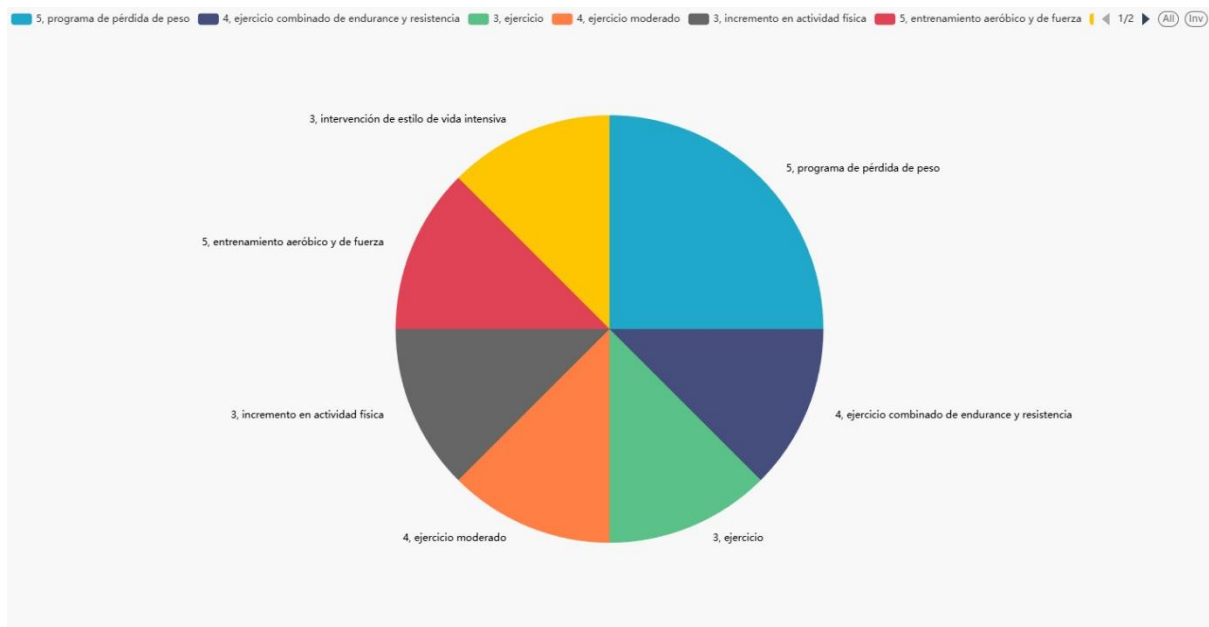


Figura 4.11 Distribución de horas por semana por actividad física

Cada segmento del pastel representa la proporción relativa de cada tipo de intervención respecto del total. Esta gráfica es relevante porque:

- Proporciona una visión global de qué tipo de intervención no farmacológica predomina en los estudios.
- Facilita la identificación de estrategias de estilo de vida que se reportan con mayor frecuencia en asociación con remisión o mejoría del control glucémico.
- En combinación con la información de medicamentos, se habilita análisis de correlaciones entre intervenciones farmacológicas y no farmacológicas, permitiendo explorar si ciertos tratamientos suelen implementarse junto con programas específicos de ejercicio o pérdida de peso.

4.4.3 Comorbilidades frecuentes en los pacientes analizados

La tercera gráfica, figura 4.12 presenta un gráfico de barras horizontales que muestra la frecuencia de las principales comorbilidades observadas: obesidad, hipertensión, enfermedad hepática grasa no alcohólica (NAFLD), enfermedad coronaria y dislipidemia. El eje vertical contiene el nombre de la comorbilidad y el eje horizontal la cantidad de casos (COUNT(*)).

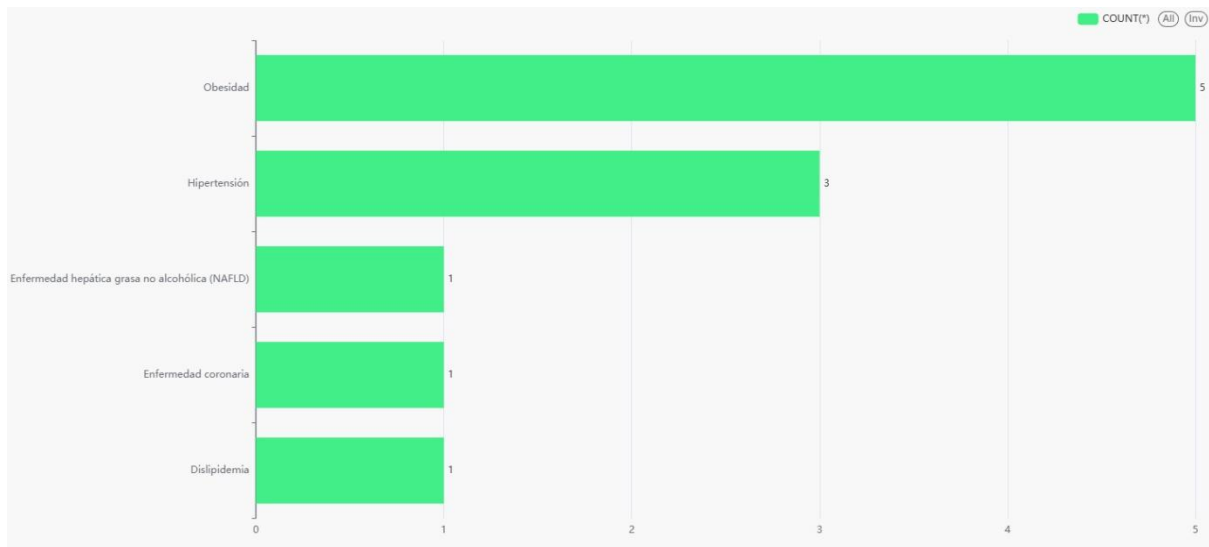


Figura 4.12 Comorbilidades asociadas a pacientes con Diabetes tipo 2

Esta visualización aporta información clave en dos sentidos:

- Permite caracterizar el perfil clínico de la población estudiada, evidenciando qué comorbilidades son más prevalentes en los pacientes candidatos a remisión.
- Sirve como insumo para análisis de correlación negativa o positiva entre la presencia de ciertas comorbilidades y la probabilidad de remisión. Por ejemplo, se puede contrastar la frecuencia de obesidad o NAFLD con los desenlaces de remisión, o explorar si determinados fármacos son más frecuentes en pacientes con ciertas comorbilidades.

4.4.4 Intervenciones quirúrgicas y su distribución

La última visualización, Figura 4.13, resume las intervenciones quirúrgicas asociadas a los casos analizados (por ejemplo, *banded sleeve gastrectomy*, *Roux-en-Y gastric bypass*, variantes laparoscópicas y técnicas combinadas). En este gráfico se representa, para cada técnica, la agregación de una métrica asociada al resultado (por ejemplo, número de casos, proporción de remisión o valor derivado de la literatura), distribuida a lo largo del eje horizontal.

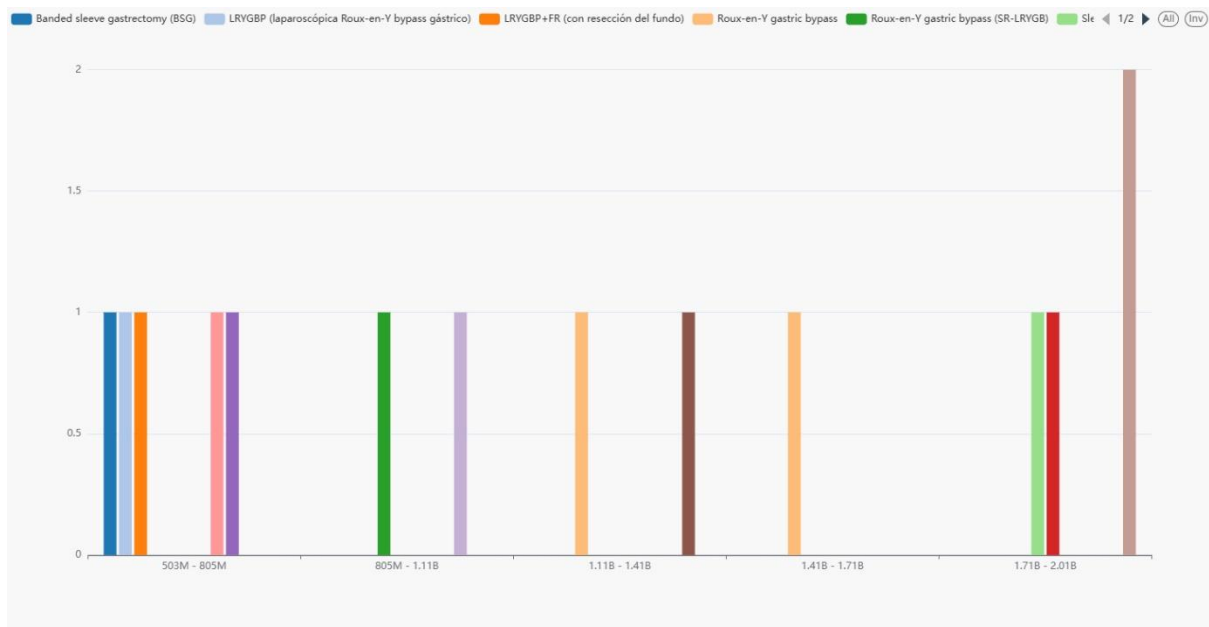


Figura 4.13 Principales Intervenciones quirúrgicas en pacientes con diabetes tipo 2 asociadas positivamente con la remisión

La relevancia de esta gráfica se centra en que:

- Permite comparar rápidamente la presencia de cada tipo de cirugía bariátrica o metabólica dentro del conjunto de estudios.
- Facilita la identificación de intervenciones que, aunque menos frecuentes, podrían vincularse con tasas de remisión especialmente altas, lo que motiva análisis más detallados.
- En combinación con las gráficas de comorbilidades y medicamentos, hace posible explorar correlaciones entre tipo de cirugía, perfil clínico previo y esquema farmacológico, enriqueciendo la comprensión multidimensional de la remisión.

4.4.5 Datos de alimentación y dietas

La Figura 4.14 presenta un *treemap* de intervenciones dietéticas donde se representa el promedio de semanas al año necesarias para que cada tipo de dieta resulte efectiva en términos de remisión o mejoría del control glucémico. Los recuadros menores detallan, de manera desagregada, las principales estrategias identificadas: dieta mediterránea (≈53.6 semanas), dieta baja en grasas (≈53.6 semanas), restricción calórica (52 semanas), incremento específico en consumo de frutas (52 semanas), frutas con una porción adicional (52 semanas) y dietas personalizadas con reemplazo de comidas (32 semanas).

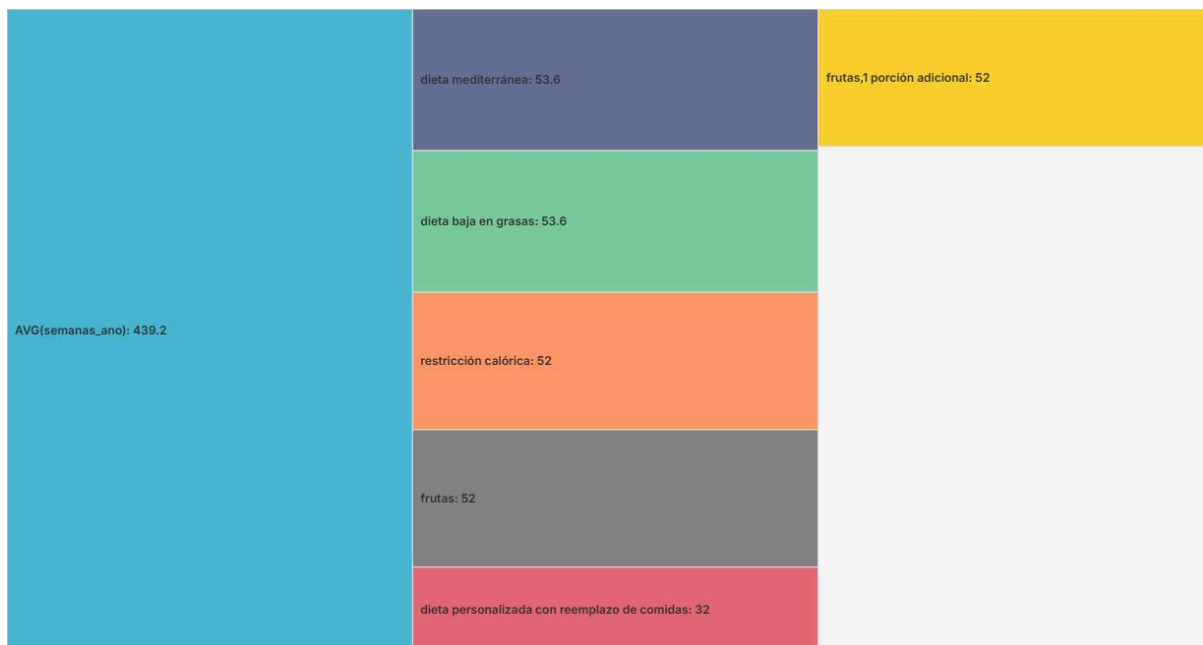


Figura 4.14 Semanas por año de seguimiento de dietas específicas relacionadas positivamente con la remisión

La distribución de tamaños pone de manifiesto que la mayoría de las intervenciones dietéticas requieren alrededor de un año de adherencia continua para mostrar resultados clínicamente relevantes, con la excepción de las dietas personalizadas con reemplazo de comidas, cuyo efecto se observa en un periodo menor. Esta visualización es relevante porque permite comparar de forma intuitiva la “intensidad temporal” de cada estrategia dietética y sugiere que la remisión de la diabetes tipo 2 no depende solo del tipo de dieta, sino también de la duración sostenida de la intervención, información clave para diseñar planes terapéuticos y para futuros análisis de correlación entre adherencia dietética y resultados clínicos.

4.4.6 Importancia de las visualizaciones para el análisis de correlaciones

En conjunto, estas visualizaciones generadas con Apache Superset no solo sirven como resúmenes descriptivos, sino que se integran en un entorno interactivo de análisis. Desde el mismo panel es posible aplicar filtros por paciente, tipo de factor, fuente de datos o periodo de tiempo; esto permite:

- Explorar correlaciones entre factores (por ejemplo, entre determinados medicamentos y programas de ejercicio, o entre comorbilidades y elección de cirugía).

- Detectar patrones emergentes que no son evidentes a partir de tablas aisladas, como combinaciones recurrentes de tratamientos y cambios de estilo de vida en pacientes que lograron remisión.
- Formular hipótesis clínicas fundamentadas en evidencia cuantitativa extraída tanto de la literatura científica como de otras fuentes integradas en el *Data Lakehouse*.

De este modo, la capa de visualización se convierte en un componente esencial de la arquitectura propuesta, al proporcionar un puente entre los datos estructurados almacenados en el *Lakehouse* y la interpretación clínica y epidemiológica necesaria para avanzar en el conocimiento sobre los factores que favorecen la remisión de la diabetes tipo 2.

Capítulo 5. Conclusiones y recomendaciones

5.1 Conclusiones

En este trabajo se presentó el desarrollo de *Data Lakehouse* orientado al estudio de la remisión de la diabetes tipo 2, integrando información procedente de literatura científica en formato PDF y de publicaciones en redes sociales. A partir del diseño de la arquitectura del *Data Lakehouse* se construyó un esquema relacional centrado en la tabla Factores, complementado por dimensiones clínicas y de estilo de vida, y se habilitaron *pipelines* automáticos de clasificación basados en modelos de lenguaje de gran tamaño (LLM).

En primer lugar, se demostró que contar con una infraestructura capaz de gestionar grandes volúmenes de datos heterogéneos —estructurados, semiestructurados y no estructurados— resulta fundamental para abordar problemas complejos como la identificación de factores asociados a la remisión de la diabetes tipo 2. La combinación de tecnologías de código abierto (Apache Kafka, Spark, Iceberg, MinIO, Airflow y Superset) permitió construir un entorno unificado donde la ingesta, el procesamiento, el almacenamiento y la visualización se integran de forma coherente y reproducible.

En segundo lugar, la arquitectura propuesta destacó por su modularidad, lo que facilita la sustitución o incorporación de nuevos componentes sin necesidad de rediseñar el sistema completo. Esta característica es especialmente relevante en un contexto de rápida evolución tecnológica, ya que posibilita la adopción futura de LLM más potentes o de motores de procesamiento y almacenamiento más avanzados, preservando la inversión realizada en el diseño del modelo de datos y en los *pipelines* existentes.

En tercer lugar, el esquema relacional y las visualizaciones desarrolladas demostraron ser útiles para organizar y explorar el conocimiento extraído de las fuentes analizadas. La tabla Factores permitió consolidar información sobre medicamentos, intervenciones de actividad física, comorbilidades e intervenciones quirúrgicas, manteniendo al mismo tiempo la trazabilidad hacia los documentos originales. Las gráficas construidas en Apache Superset facilitaron la identificación de patrones descriptivos y abren la puerta a análisis correlacionales más sofisticados entre tratamientos, características clínicas y remisión.

Finalmente, aunque el enfoque se centró en un conjunto acotado de factores y fuentes de información, los resultados obtenidos muestran que la arquitectura de *Data Lakehouse* aquí propuesta constituye una base sólida y extensible para apoyar investigaciones posteriores en remisión de diabetes tipo 2, así como en otros problemas clínicos que requieran integrar grandes volúmenes de datos multimodales.

5.2 Recomendaciones

A partir de los resultados alcanzados, se identifican diversas líneas de trabajo que pueden ampliar y fortalecer la propuesta realizada:

- Ampliación del esquema con nuevas dimensiones.

Resulta pertinente extender el modelo de datos incorporando dimensiones adicionales que capturen aspectos psicológicos, socioeconómicos y conductuales (por ejemplo, nivel de estrés, calidad del sueño, adherencia al tratamiento o apoyo social). La integración de registros electrónicos de salud, datos de dispositivos portátiles (*wearables*) y, eventualmente, información genómica permitiría enriquecer el análisis de factores asociados a la remisión y avanzar hacia modelos más personalizados.

- Mejora de los *pipelines* de procesamiento y de los modelos de lenguaje.

Si bien los LLM utilizados demostraron ser funcionales para extraer factores relevantes, la rápida evolución de estos modelos sugiere la conveniencia de evaluar futuras versiones con mayor capacidad de razonamiento clínico y mejor soporte multilingüe. Gracias a la naturaleza modular de la arquitectura, la sustitución del modelo actual por LLM más potentes, o la incorporación de etapas adicionales de validación y normalización terminológica, puede realizarse con un impacto mínimo en el resto de los componentes.

- Integración con aplicaciones web y móviles.

Una línea de trabajo especialmente prometedora consiste en exponer la información almacenada en el *Lakehouse* a través de aplicaciones web o móviles orientadas a profesionales de la salud y pacientes. Estas aplicaciones podrían presentar paneles personalizados sobre factores de remisión, alertas tempranas, recomendaciones basadas en evidencia y herramientas de seguimiento del tratamiento, apoyándose en las tablas de hechos y dimensiones ya definidas. De este modo, la arquitectura dejaría

de ser únicamente una plataforma analítica para convertirse en el *backend* de soluciones clínicas interactivas centradas en la remisión de la diabetes.

En conjunto, estas líneas de trabajo apuntan a consolidar la arquitectura de *Data Lakehouse* como una plataforma robusta, escalable y clínicamente relevante para el estudio y la promoción de la remisión de la diabetes tipo 2.

Productos académicos

- Artículo “Data Lakehouse for Type 2 Diabetes Remission: Scoping Review and Architectural Proposal”



Referencias

- [1] OMS, “Diabetes.” Accessed: Jul. 30, 2024. [Online]. Available: <https://www.who.int/es/news-room/fact-sheets/detail/diabetes>
- [2] Federación Internacional de Diabetes, “Datos y cifras sobre la diabetes.”
- [3] A. Basto-Abreu *et al.*, “Prevalencia de prediabetes y diabetes en México: Ensanut 2022,” *Salud Publica Mex*, vol. 65, pp. s163–s168, Jun. 2023, doi: 10.21149/14832.
- [4] S. Putta *et al.*, “Diabetes Mellitus and Male Aging: Pharmacotherapeutics and Clinical Implications,” *Curr Pharm Des*, vol. 23, no. 30, Nov. 2017, doi: 10.2174/1381612823666170823103830.
- [5] I.-I. Witzel, H. F. Jelinek, K. Khalaf, S. Lee, A. H. Khandoker, and H. Alsafar, “Identifying Common Genetic Risk Factors of Diabetic Neuropathies,” *Front Endocrinol (Lausanne)*, vol. 6, May 2015, doi: 10.3389/fendo.2015.00088.
- [6] Organización Panamericana de la Salud, “Diabetes.” Accessed: Jul. 30, 2024. [Online]. Available: <https://www.paho.org/es/temas/diabetes#:~:text=La%20diabetes%20es%20un a%20enfermedad,los%20ri%C3%B1ones%20y%20los%20nervios>.
- [7] J. B. Buse *et al.*, “How Do We Define Cure of Diabetes?,” *Diabetes Care*, vol. 32, no. 11, pp. 2133–2135, Nov. 2009, doi: 10.2337/dc09-9036.
- [8] Google Cloud, “What is a data warehouse?” Accessed: Jul. 30, 2024. [Online]. Available: <https://cloud.google.com/learn/what-is-a-data-warehouse?hl=es>
- [9] Google Cloud, “What is a data lake?” Accessed: Jul. 30, 2024. [Online]. Available: <https://cloud.google.com/learn/what-is-a-data-lake?hl=es-419#what-is-a-data-lake>
- [10] Google Cloud, “What is a data lakehouse?” Accessed: Jul. 30, 2024. [Online]. Available: <https://cloud.google.com/discover/what-is-a-data-lakehouse?hl=es-419>
- [11] Salesforce, “Big Data: ¿Qué es y cómo ayuda a mi negocio?” Accessed: Aug. 01, 2024. [Online]. Available: <https://www.salesforce.com/mx/blog/big-data/>
- [12] X. Ma, “Data Repository,” in *Encyclopedia of Big Data*, Cham: Springer International Publishing, 2017, pp. 1–4. doi: 10.1007/978-3-319-32001-4_59-1.

- [13] Google cloud, “¿Qué es el aprendizaje automático (AA)?” Accessed: Oct. 01, 2024. [Online]. Available: <https://cloud.google.com/learn/what-is-machine-learning?hl=es-419>
- [14] E. L. Lim, K. G. Hollingsworth, B. S. Aribisala, M. J. Chen, J. C. Mathers, and R. Taylor, “Reversal of type 2 diabetes: normalisation of beta cell function in association with decreased pancreas and liver triacylglycerol,” *Diabetologia*, vol. 54, no. 10, pp. 2506–2514, Oct. 2011, doi: 10.1007/s00125-011-2204-7.
- [15] P. Shamanna *et al.*, “Reducing HbA1c in Type 2 Diabetes Using Digital Twin Technology-Enabled Precision Nutrition: A Retrospective Analysis,” *Diabetes Therapy*, vol. 11, no. 11, pp. 2703–2714, Nov. 2020, doi: 10.1007/s13300-020-00931-w.
- [16] H. M. Reis Joe, *Fundamentals of Data Engineering*. O’Reilly Media, Inc., 2022.
- [17] M. H. Siddiqi, M. Idris, and M. Alruwaili, “FAIR Health Informatics: A Health Informatics Framework for Verifiable and Explainable Data Analysis,” *Healthcare*, vol. 11, no. 12, p. 1713, Jun. 2023, doi: 10.3390/healthcare11121713.
- [18] P. Wang *et al.*, “A construction System of Lake-Warehouse Integration in the Electricity Industry Based on Hudi,” in *Proceedings of the 3rd International Conference on Public Management and Big Data Analysis, PMBDA 2023, December 15–17, 2023, Nanjing, China*, EAI, 2024. doi: 10.4108/eai.15-12-2023.2345365.
- [19] E. Begoli, I. Goethert, and K. Knight, “A Lakehouse Architecture for the Management and Analysis of Heterogeneous Data for Biomedical Research and Mega-biobanks,” in *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2021, pp. 4643–4651. doi: 10.1109/BigData52589.2021.9671534.
- [20] R. Li, M. Choy, and M. N. Laik, “A Cloud Data Lakehouse-Based AI Diagnostic Solution for Small and Medium-Sized Health Facilities,” in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, Michigan, USA: IEOM Society International, 2022, pp. 945–956. doi: 10.46254/AP03.20220190.
- [21] S. Park, C.-S. Yang, and J. Kim, “Design of Vessel Data Lakehouse with Big Data and AI Analysis Technology for Vessel Monitoring System,” *Electronics (Basel)*, vol. 12, no. 8, p. 1943, Apr. 2023, doi: 10.3390/electronics12081943.

- [22] Y. Zhang *et al.*, “A heterogeneous multi-modal medical data fusion framework supporting hybrid data exploration,” *Health Inf Sci Syst*, vol. 10, no. 1, p. 22, Aug. 2022, doi: 10.1007/s13755-022-00183-x.
- [23] A. Tripathi, A. Waqas, K. Venkatesan, Y. Yilmaz, and G. Rasool, “Building Flexible, Scalable, and Machine Learning-Ready Multimodal Oncology Datasets,” *Sensors*, vol. 24, no. 5, p. 1634, Mar. 2024, doi: 10.3390/s24051634.
- [24] V. Bellandi, P. Ceravolo, J. Maggesi, and S. Maghool, “Data management for continuous learning in EHR systems,” *ACM Trans Internet Technol*, May 2024, doi: 10.1145/3660634.
- [25] J. Schneider, A. Lutsch, C. Gröger, H. Schwarz, and B. Mitschang, “First Experiences on the Application of Lakehouses in Industrial Practice,” in *35th GI-Workshop on Foundations of Databases*, May 2024. Accessed: Jan. 13, 2025. [Online]. Available: <https://ceur-ws.org/Vol-3710/paper1.pdf>
- [26] C. Vox, D. Broneske, J. Piewek, J. Feigel, and G. Saake, “Investigating Lakehouse-Backbones for Vehicle Sensor Data,” 2023, pp. 243–258. doi: 10.1007/978-3-031-39847-6_17.
- [27] T. Zhao *et al.*, “Multi-modal Medical Data Exploration Based on Data Lake,” in *Health Information Science: 12th International Conference*, Melbourne, Oct. 2023, pp. 213–222. doi: 10.1007/978-981-99-7108-4_18.
- [28] A. Harby, N. Al-Yateem, and F. H. Zulkernine, “Revolutionizing Healthcare Management: Architecture of a Web-based Medical Triage Service,” Osaka: COMPSAC, Jul. 2024. doi: 10.1109/COMPSAC61105.2024.00299.
- [29] P. Ren *et al.*, “A Hybrid Medical Causal Inference Platform Based on Data Lake,” in *Health Information Science: 11th International Conference*, 2022, pp. 136–144. doi: 10.1007/978-3-031-20627-6_13.
- [30] Q. Xiao *et al.*, “MHDML: Construction of a Medical Lakehouse for Multi-source Heterogeneous Data,” in *Health Information Science Lecture Notes in Computer Science*, 2022, pp. 127–135. doi: 10.1007/978-3-031-20627-6_12.
- [31] M. Sheng *et al.*, “A multi-source heterogeneous medical data enhancement framework based on lakehouse,” *Health Inf Sci Syst*, vol. 12, no. 1, p. 37, Jul. 2024, doi: 10.1007/s13755-024-00295-6.
- [32] The Apache Software Foundation, “Apache Kafka Documentation.” Accessed: Aug. 26, 2024. [Online]. Available: <https://kafka.apache.org/documentation/>

- [33] Apache NiFi Team, “Apache NiFi Documentation.” Accessed: Aug. 26, 2024. [Online]. Available: <https://nifi.apache.org/documentation/v2/>
- [34] The Apache Software Foundation, “Apache Spark.” Accessed: Aug. 26, 2024. [Online]. Available: <https://spark.apache.org/>
- [35] The Apache Software Foundation, “Apache Flink.” Accessed: Aug. 26, 2024. [Online]. Available: <https://flink.apache.org/>
- [36] The Apache Software Foundation, “Apache Iceberg.” Accessed: May 12, 2025. [Online]. Available: <https://iceberg.apache.org/>
- [37] The Apache Software Foundation, “Apache Doris.” Accessed: Sep. 08, 2024. [Online]. Available: <https://doris.apache.org/>
- [38] The Apache Software Foundation, “Apache Superset.” Accessed: Aug. 26, 2024. [Online]. Available: <https://superset.apache.org/>
- [39] Metabase, “A tour of Metabase.” Accessed: Sep. 08, 2024. [Online]. Available: <https://www.metabase.com/learn/metabase-basics/getting-started/tour-of-metabase>
- [40] The Apache Software Foundation, “Apache Airflow.” Accessed: May 12, 2025. [Online]. Available: <https://airflow.apache.org/>
- [41] T. Mucci, M. Scapicchio, and C. Stryker, “¿Que es DataOps?”
- [42] M. J. Page *et al.*, “Declaración PRISMA 2020: una guía actualizada para la publicación de revisiones sistemáticas,” *Rev Esp Cardiol*, vol. 74, no. 9, pp. 790–799, Sep. 2021, doi: 10.1016/j.recesp.2021.06.016.
- [43] L. M. Fleuren *et al.*, “The Dutch Data Warehouse, a multicenter and full-admission electronic health records database for critically ill COVID-19 patients,” *Crit Care*, vol. 25, no. 1, p. 304, Dec. 2021, doi: 10.1186/s13054-021-03733-z.
- [44] T. Pressat-Laffouilhère *et al.*, “Evaluation of Doc’EDS: a French semantic search tool to query health documents from a clinical data warehouse,” *BMC Med Inform Decis Mak*, vol. 22, no. 1, p. 34, Dec. 2022, doi: 10.1186/s12911-022-01762-4.
- [45] A. Lamer, M. D. Moussa, R. Marcilly, R. Logier, B. Vallet, and B. Tavernier, “Development and usage of an anesthesia data warehouse: lessons learnt from a 10-year project,” *J Clin Monit Comput*, vol. 37, no. 2, pp. 461–472, Apr. 2023, doi: 10.1007/s10877-022-00898-y.
- [46] M. Ansoborlo *et al.*, “Feasibility of automated surveillance of implantable devices in orthopaedics via clinical data warehouse: the Studio study,” *BMC Med Inform*

- Decis Mak*, vol. 24, no. 1, p. 324, Nov. 2024, doi: 10.1186/s12911-024-02697-8.
- [47] M. Kaspar *et al.*, “Querying a Clinical Data Warehouse for Combinations of Clinical and Imaging Data,” *J Digit Imaging*, vol. 36, no. 2, pp. 715–724, Nov. 2022, doi: 10.1007/s10278-022-00727-3.
 - [48] L. Baum *et al.*, “An interactive dashboard for analyzing user interaction patterns in the i2b2 clinical data warehouse,” *BMC Med Inform Decis Mak*, vol. 24, no. 1, p. 333, Nov. 2024, doi: 10.1186/s12911-024-02748-0.
 - [49] M. Ndisha *et al.*, “Leveraging electronic medical records for HIV testing, care, and treatment programming in Kenya—the national data warehouse project,” *BMC Med Inform Decis Mak*, vol. 23, no. 1, p. 183, Sep. 2023, doi: 10.1186/s12911-023-02265-6.
 - [50] D. Cheng-Zarate *et al.*, “Creating a data warehouse to support monitoring of NSQHS blood management standard from EMR data,” *BMC Med Inform Decis Mak*, vol. 24, no. 1, p. 353, Nov. 2024, doi: 10.1186/s12911-024-02732-8.
 - [51] G. Turcan and S. Peker, “A multidimensional data warehouse design to combat the health pandemics,” *Journal of Data, Information and Management*, vol. 4, no. 3–4, pp. 371–386, Dec. 2022, doi: 10.1007/s42488-022-00082-6.
 - [52] C. G. Arnold *et al.*, “Accessing and utilizing clinical and genomic data from an electronic health record data warehouse,” *Transl Med Commun*, vol. 8, no. 1, p. 7, Mar. 2023, doi: 10.1186/s41231-023-00140-0.
 - [53] C. Windsor *et al.*, “Healthcare trajectory of critically ill patients with necrotizing soft tissue infections: a multicenter retrospective cohort study using the clinical data warehouse of Greater Paris University Hospitals,” *Ann Intensive Care*, vol. 12, no. 1, p. 115, Dec. 2022, doi: 10.1186/s13613-022-01087-5.
 - [54] Z.-Y. Shae and J. J. P. Tsai, “On the Design of Medical Data Ecosystem for Improving Healthcare Research and Commercial Incentive,” in *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*, IEEE, Dec. 2021, pp. 124–131. doi: 10.1109/CogMI52975.2021.00024.
 - [55] I. A. Melchor-Uceda, J. C. Olivares-Rojas, J. A. Gutierrez-Gnecchi, M. C. Garcia-Ramirez, E. Reyes-Archundia, and A. C. Tellez-Anguiano, “Data Ingestion System for Interoperability and Integration of Hospital Data Online and in Real Time,” in *2021 Mexican International Conference on Computer Science (ENC)*, IEEE, Aug. 2021, pp. 1–5. doi: 10.1109/ENC53357.2021.9534795.

- [56] D. Babu, G. S, K. K, M. PT, and P. S, "Retrieval of Large Data From Medical Lake Repository-Heart Note," in *2022 1st International Conference on Computational Science and Technology (ICCST)*, IEEE, Nov. 2022, pp. 518–522. doi: 10.1109/ICCST55948.2022.10040351.
- [57] G. S, A. J. Mabel Rani, A. P. S, C. N. Ravi, M. Jeyalaxmi, and V. S. Pandi, "A Novel Approach to Design Secured and Privacy Enabled Health Data Linkage System based on Cyber Security Principles," in *2024 IEEE International Conference on Big Data & Machine Learning (ICBDML)*, IEEE, Feb. 2024, pp. 14–20. doi: 10.1109/ICBDML60909.2024.10577373.
- [58] N. Amara, O. Lamouchi, and S. Gattoufi, "Design of a Breast Image Data Warehouse Framework," in *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, IEEE, Feb. 2020, pp. 1–13. doi: 10.1109/OCTA49274.2020.9151855.
- [59] D. Kumar and S. Li, "Separating Storage and Compute with the Databricks Lakehouse Platform," in *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, Oct. 2022, pp. 1–2. doi: 10.1109/DSAA54385.2022.10032386.
- [60] L. Maria Hernandez Cruz *et al.*, "Use of the Hefesto v2.0 methodology to implement a Data warehouse: Case applied COVID-19," in *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, Jun. 2022, pp. 1–6. doi: 10.23919/CISTI54924.2022.9820132.
- [61] H. Spengler, I. Gatz, F. Kohlmayer, K. A. Kuhn, and F. Prasser, "Improving Data Quality in Medical Research: A Monitoring Architecture for Clinical and Translational Data Warehouses," in *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE, Jul. 2020, pp. 415–420. doi: 10.1109/CBMS49503.2020.00085.
- [62] S. D, P. Divya, S. K, A. K. Dubey, S. Vijay, and M. G, "A Novel AI Framework for Assuring Data Sustainability in Health Care Dataset," in *2023 Third International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, IEEE, Sep. 2023, pp. 161–166. doi: 10.1109/ICUIS60567.2023.00035.
- [63] T. F. Setiadi and R. Rahutomo, "A Data Warehouse Schema for Monitoring Regional COVID-19 Case Registration," in *2022 International Conference on*

- Information Management and Technology (ICIMTech)*, IEEE, Aug. 2022, pp. 144–148. doi: 10.1109/ICIMTech55957.2022.9915222.
- [64] A. F. Neamah, “Flexible Data Warehouse: Towards Building an Integrated Electronic Health Record Architecture,” in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, IEEE, Sep. 2020, pp. 1038–1042. doi: 10.1109/ICOSEC49089.2020.9215433.
- [65] P. K. Yeng *et al.*, “HEALER2: A Framework for Secure Data Lake Towards Healthcare Digital Transformation Efforts in Low and Middle-Income Countries,” in *2023 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, IEEE, Aug. 2023, pp. 1–9. doi: 10.1109/ETNCC59188.2023.10284959.
- [66] D. B. M, K. Ramesh, P. N. Renjith, and B. Prabha, “A Risk Model Based Heart Disease Prediction Using Data Lake Architecture,” in *2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, IEEE, Feb. 2022, pp. 1–6. doi: 10.1109/ICEEICT53079.2022.9768452.
- [67] M. Kumar, S. Lee, B. Hoon Park, J. Blum, M. Ward, and J. R. Nebeker, “Advanced Health Information Technology Analytic Framework and Application to Hazard Detection,” in *2020 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2020, pp. 1–4. doi: 10.1109/BigData50022.2020.9442793.
- [68] S. Patil and A. S. Z. A. Belloum, “Processing FHIR in modern Data Lakehouse,” in *2024 IEEE 20th International Conference on e-Science (e-Science)*, IEEE, Sep. 2024, pp. 1–7. doi: 10.1109/e-Science62913.2024.10678734.
- [69] F. Abdelhedi, R. Jemmali, and G. Zurfluh, “Medical data lake query assistance,” in *2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, IEEE, Dec. 2023, pp. 1–8. doi: 10.1109/AICCSA59173.2023.10479336.
- [70] D. Babu M., K. Ramesh, P. N. Renjith, and B. Prabha, “An Efficient Healthcare Medication System with Clustering Algorithm Using Euclidean Distance Adjoining Data Lake,” in *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, IEEE, Jan. 2022, pp. 1–5. doi: 10.1109/ACCAI53970.2022.9752473.

- [71] R. González-Velázquez *et al.*, “Smart Factory Hub – Towards a Data Mesh in Smart Manufacturing,” *Procedia Comput Sci*, vol. 232, pp. 2709–2719, 2024, doi: 10.1016/j.procs.2024.02.088.
- [72] B. Oliveira, A. Duarte, and Ó. Oliveira, “Towards a Data Catalog for Data Analytics,” *Procedia Comput Sci*, vol. 237, pp. 691–700, 2024, doi: 10.1016/j.procs.2024.05.155.
- [73] A. A. Harby and F. Zulkernine, “Data Lakehouse: A survey and experimental study,” *Inf Syst*, vol. 127, p. 102460, Jan. 2025, doi: 10.1016/j.is.2024.102460.
- [74] P. Bindal and K. Madan, “Quantum Flow: Enterprise Data Orchestration and Processing Suite,” in *Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing*, New York, NY, USA: ACM, Aug. 2024, pp. 577–584. doi: 10.1145/3675888.3676116.
- [75] P. Edara, J. Forbesj, and B. Li, “Vortex: A Stream-oriented Storage Engine For Big Data Analytics,” in *Companion of the 2024 International Conference on Management of Data*, New York, NY, USA: ACM, Jun. 2024, pp. 175–187. doi: 10.1145/3626246.3653396.
- [76] S. Batalla, “From data to value: turn unstructured data into a dimensional data model using Data Warehouse (SAP BW/4HANA) and Business Intelligence (SAP Lumira designer) visualizations.,” in *Proceedings of the 2024 10th International Conference on Computer Technology Applications*, New York, NY, USA: ACM, May 2024, pp. 103–108. doi: 10.1145/3674558.3674572.
- [77] S. Hoseini, A. Ali, H. Shaker, and C. Quix, “SEDAR: A Semantic Data Reservoir for Heterogeneous Datasets,” in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, New York, NY, USA: ACM, Oct. 2023, pp. 5056–5060. doi: 10.1145/3583780.3614753.