Modelo de arquitectura para la ingestión, procesamiento y análisis de datos a gran escala

Cristian Eduardo Rojas Pedraza

Trabajo de Grado para Optar al Título de Ingeniero de Sistemas

Director

Jathinson Meneses Mendoza

Magister en Gestión, Aplicación y Desarrollo de Software

Codirector

Henry Andres Jimenez Herrera

Magister en Ingeniería de Sistemas e Informática

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas

Bucaramanga

2024

Tabla de contenido

Lista de Tablas

Lista de Figuras

Resumen

Título: Modelo de arquitectura para la ingestión, procesamiento y análisis de datos a gran escala[*]

Autor: Cristian Eduardo Rojas Pedraza[**]

Palabras Clave: Lago de datos, Arquitectura lagos de datos, Datos a gran escala, Procesamiento de datos

Descripción:

Esta tesis investiga el campo del big data, explorando soluciones para su gestión y aprovechamiento óptimo. En un mundo donde la acumulación de datos es imparable, las empresas encuentran oportunidades invaluables. Herramientas como Hadoop y Elasticsearch, diseñadas para superar los desafíos del procesamiento convencional, permiten la eficaz manipulación de vastos volúmenes de datos distribuidos. Esta investigación también ofrece un panorama completo de las herramientas disponibles para construir soluciones integrales.

El núcleo de esta tesis reside en la exploración profunda de los fundamentos de los Data Lakes y su integración en la gestión del big data. Se desglosan arquitecturas esenciales y se examinan tecnologías clave que posibilitan la construcción de entornos de almacenamiento y procesamiento de datos altamente adaptables.

Un caso de estudio concreto en la industria petrolera de Texas sirve de ejemplo ilustrativo. Se detalla la recolección y análisis de datos relevantes provenientes de diversas fuentes, como la Energy Information Administration y el Servicio Geológico de Estados Unidos. Este estudio demuestra cómo las herramientas de procesamiento de big data y la implementación de Data Lakes pueden generar valor y simplificar la toma de decisiones estratégicas en un sector industrial altamente competitivo.

En resumen, esta tesis aborda el universo del big data, enfocándose en cómo los Data Lakes se han convertido en una solución esencial para gestionar la avalancha de información en nuestros días. Con un enfoque particular en la industria petrolera de Texas, esta investigación explora cómo estas tecnologías innovadoras pueden revolucionar el tratamiento de datos, facilitando análisis profundos y decisiones fundamentadas.

---

[*] Trabajo de Grado

[**] Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Ingeniería de Sistemas. Director: Jathinson Meneses Mendoza. Magister en Gestión, Aplicación y Desarrollo de Software. Codirector: Henry Andres Jimenez Herrera. Magister en Ingeniería de Sistemas e Informática.

Abstract

Title: Architecture model for large-scale data ingestion, processing and analysis*

Author: Cristian Eduardo Rojas Pedraza**

Key Words: Data lake, Data Lake architecture, large scale data, Data processing

Description:

This thesis investigates the broad field of big data, exploring solutions for its management and optimal use. In a world where data accumulation is unstoppable, companies are finding invaluable opportunities. Tools such as Hadoop and Elasticsearch, designed to overcome the challenges of conventional processing, enable the efficient manipulation of vast volumes of distributed data. This research also provides a comprehensive overview of the tools available to build end-to-end solutions.

The core of this thesis lies in the in-depth exploration of the fundamentals of Data Lakes and their integration into big data management. It breaks down essential architectures and examines key technologies that make it possible to build highly adaptive data storage and processing environments.

A specific case study in the Texas oil industry serves as an illustrative example. The collection and analysis of relevant data from a variety of sources, such as the Energy Information Administration and the U.S. Geological Survey, is detailed. This study demonstrates how big data processing tools, and the implementation of Data Lakes can generate value and simplify strategic decision making in a highly competitive industry sector.

In summary, this thesis addresses the universe of big data, focusing on how Data Lakes have become an essential solution for managing today's flood of information. With a particular focus on the Texas oil industry, this research explores how these innovative technologies can revolutionize data processing, facilitating deep analysis and informed decisions.

---

* Degree Work

** Faculty of Physicomechanical Engineering. School of Systems Engineering and Computer Science. Systems engineer. Director: Jathinson Meneses Mendoza. Master’s in software management, Application, and Development. Co-director: Henry Andres Jimenez Herrera. Master’s in systems Engineering and Computer Science.

Introducción

En la era de la información actual, el ritmo acelerado con el que se generan datos ha dado lugar a grandes desafíos: cómo gestionar, analizar y extraer conocimientos significativos de cantidades masivas de información. Esta generación masiva de datos la conocemos con el término de big data (Liu, Isah, & Zulkernine, 2020), son conjuntos de datos complejos que superan las capacidades de las herramientas y técnicas tradicionales de procesamiento.

En este contexto, surge la necesidad de implementar una arquitectura innovadora y escalable que no solo aborde la ingesta y el procesamiento de datos a gran escala, sino que también permita su análisis en tiempo real para tomar decisiones informadas y estratégicas.

Una arquitectura efectiva para la ingestión, procesamiento y análisis de datos a gran escala es la columna vertebral que sustenta la transformación de datos brutos en información valiosa y aplicable. Los enfoques tradicionales ya no son suficientes; en su lugar, se requiere una estructura que pueda adaptarse a la evolución constante de las demandas de datos y que aproveche tecnologías de vanguardia para lograr resultados impactantes.

Dos de estas tecnologías merecen especial atención en este contexto. En primer lugar, Hadoop, un sistema de procesamiento distribuido de código abierto. Su enfoque escalable y con capacidad para gestionar eficientemente la distribución y el procesamiento permiten soluciones más robustas. Por otro lado, Elasticsearch, también de código abierto, ha redefinido la búsqueda y recuperación de datos, permitiendo un acceso rápido y preciso a información diversa en entornos donde la velocidad es esencial, soportando una amplia variedad de tipos de datos, desde datos textuales hasta números y datos geoespaciales, tanto estructurados como no estructurados.

Esta investigación se enfoca en la definición e implementación de una arquitectura que aborda los desafíos de la ingesta y el procesamiento de datos a gran escala. A través de la exploración de conceptos fundamentales, análisis de arquitecturas existentes y la identificación de tecnologías pertinentes, esta tesis busca construir un modelo arquitectónico integral y aplicarlo en un escenario concreto, sentando así las bases para una gestión efectiva y estratégica de los datos en la era del big data.

1. Justificación

En la actual era digital, el flujo constante de información ha transformado la manera en que las organizaciones operan y toman decisiones. La diversidad de fuentes, desde bases de datos internas hasta plataformas de redes sociales, ha dado lugar a una extensa variedad de datos que desafían los métodos tradicionales de gestión y análisis. Ante este panorama, surge una necesidad: ¿cómo podemos explorar y aprovechar esta información?

El concepto de "Data Lake" o lago de datos se ha convertido en un pilar fundamental en este contexto. Al permitir el almacenamiento escalable y eficiente de datos crudos en todas sus formas y tipos, el lago de datos se presenta como una solución crucial para administrar, analizar y extraer conocimientos significativos de la avalancha de información (Hlupić et al. 2022). Es el punto de encuentro entre la creciente necesidad de almacenar datos diversificados y la capacidad de acceder a ellos de manera ágil y efectiva.

Toda esta información es el combustible para la mayoría de las organizaciones, ya que, deben innovar con frecuencia en este mundo altamente competitivo, es aquí donde se hace necesario un análisis detallado de los datos con la finalidad de impulsar la toma de decisiones

comerciales inteligentes, permitiéndoles ser más eficaces y eficientes, en este punto es donde las herramientas de procesamiento de datos a gran escala toman importancia.

Esta investigación ayudará a comprender conceptos generales relacionados con lago de datos, almacenamiento, procesamiento y análisis de datos, haciendo énfasis en las arquitecturas, adicionalmente se cubren algunas de las tecnologías que potencialmente se pueden usar para su implementación, aspirando a contribuir una base de conocimientos en esta área en constante evolución.

2. Objetivos

2.1. Objetivo General

Definir una arquitectura que permita la integración de datos a gran escala para la visualización, monitoreo y explotación en tiempo real.

2.2. Objetivos Específicos

- Identificar los conceptos generales relacionados con lago de datos, almacenamiento, procesamiento y análisis de datos.
- Analizar las diferentes arquitecturas para la implementación de un lago de datos.
- Identificar las tecnologías que se pueden utilizar para la implementación de un lago de datos.
- Diseñar una arquitectura para la ingestión, procesamiento y análisis de datos a gran escala.
- Implementar el diseño de la arquitectura en un caso de estudio específico.

3. Marco Teórico

3.1. Lago de Datos (Data Lake)

Un lago de datos es un sistema de almacenamiento masivo que recopila y mantiene una gran cantidad de datos sin procesar en su formato original, permitiendo su acceso y análisis cuando es necesario.

Liu, R et al. (2020), mencionan que esto contrasta con el enfoque tradicional de almacenamiento de datos, también conocido como esquema de escritura, que requiere un diseño más inicial y suposiciones sobre cómo se utilizarán los datos.

A diferencia del almacenamiento tradicional, no requiere una estructura previa y esquema de los datos, lo que permite una mayor flexibilidad y eficiencia en el procesamiento y análisis de los datos. Convirtiéndolo en una herramienta valiosa para el almacenamiento de datos en su estado original, permitiendo a los usuarios realizar análisis ad-hoc y tomar decisiones basadas en los datos en tiempo real.

Liu, R et al. (2020), adicionalmente se comenta que un lago de datos permite integrar y procesar técnicas de búsqueda y análisis de datos que de otro modo no serían posibles.

Un lago de datos puede ser construido desde cero o utilizando una plataforma existente. Algunas plataformas de nube populares, como Amazon Web Services, Microsoft Azure, Google Cloud y el ecosistema Hadoop, ofrecen servicios que se pueden enlazar para lograr una escalabilidad adecuada.

Estas plataformas ofrecen una variedad de opciones de almacenamiento y procesamiento de datos, así como servicios de análisis y visualización, lo que permite a los usuarios implementar un lago de datos sin la necesidad de una gran inversión en infraestructura.

Según Fang et al. (2015), un lago de datos es capaz de:

- Capturar y almacenar grandes cantidades de datos sin procesar a bajo costo.
- Almacenar varios tipos de datos en un solo repositorio.
- Realizar transformaciones ETL (Extract, Transform, Load) en los datos.
- Ofrecer flexibilidad en cuanto a la estructura de los datos, permitiendo su lectura sin necesidad de un esquema previamente definido.

Cuzzocrea (2021) menciona que el objetivo final de un lago de datos es soportar procedimientos avanzados de análisis tales como la exploración de big data, el descubrimiento de conocimientos a través de big data, el análisis ad-hoc de big data, los análisis complejos de big data y el uso de herramientas complejas sobre big data, como OLAP, informes y paneles.

3.2 Almacén de Datos (Data Warehouse)

Un almacén de datos es una estructura de almacenamiento específicamente diseñada para almacenar y gestionar datos estructurados. La construcción de un almacén de datos implica un proceso de análisis y diseño detallado, que incluye el examen de las fuentes de datos, la comprensión de los procesos de negocio y el perfilamiento de los datos. Como resultado, se obtiene un modelo de datos altamente estructurado y normalizado, listo para su uso en la generación de informes y análisis de datos. La Tabla 1 del estudio de Liu, R et al. (2020) resume las diferencias entre Almacén de datos y Lago de datos.

Tabla 1

Comparación Data Lake y Data Warehouse

Características	Almacén de datos	Lago de datos
Estructura de datos	Los datos son procesados, sólo la información estructurada es capturada y organizada en esquemas	Los datos son sin procesar, todos los tipos de datos (estructurados, semi estructurados, no estructurados) se capturan en su forma original
Costo de almacenamiento	El almacenamiento de datos lleva mucho tiempo y es costoso	El almacenamiento de datos es relativamente económico.
Accesibilidad	Costoso hacer cambios, por lo tanto, bastante complicado	Las actualizaciones se pueden realizar rápidamente, lo que lo hace muy accesible.
Esquema	El esquema se define antes de que se almacenen los datos, lo que ofrece rendimiento y seguridad	El esquema se define después de almacenar los datos, lo que lo hace muy ágil y escalable
Procesamiento de datos	Utiliza el proceso de extracción, transformación y carga (ETL).	Utiliza el proceso de extracción, transformación y carga (ETL).
Usuarios	Ideal para usuarios operativos, como analistas de negocios, ya que los datos están estructurados y son fáciles de usar.	Ideal para usuarios avanzados como científicos de datos que realizan análisis profundos con herramientas analíticas avanzadas

Los almacenes de datos siguen siendo herramientas empresariales valiosas para el análisis de datos. Sin embargo, con la evolución de las necesidades analíticas, se ha desarrollado el concepto de lago de datos como una solución complementaria, permitiendo una gestión más versátil y robusta de la información.

3.3 Arquitectura

La arquitectura de Data Lake ha evolucionado significativamente desde su introducción en los primeros años de la década de 2010. Según Hlupić et al. (2022), inicialmente, los lagos de datos estaban destinados a contener grandes volúmenes de diversos datos en su forma original y sin cambios. Han surgido varias arquitecturas, cada una de las cuales brinda ciertos beneficios para el almacenamiento de datos, el análisis de datos y el consumo del usuario final.

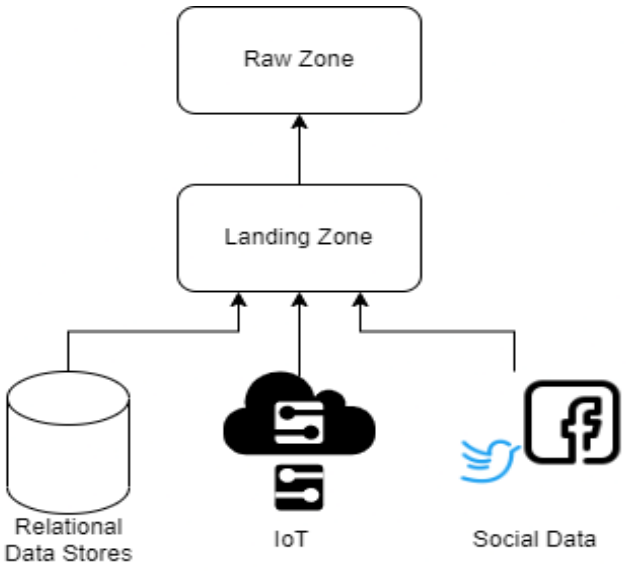
Uno de los bloques comunes en todas las arquitecturas es la zona de aterrizaje, donde los datos sin procesar se colocan para su ingestión en el lago de datos (Hlupić et al., 2022). La ingestión de datos se ha convertido en el enfoque principal para cargar los datos en el lago de datos, ya que es más adecuado para procesos de extracción, carga y transformación que para la integración de datos. Este es especialmente significativo cuando las fuentes de datos proporcionan datos de alta velocidad y no estructurados, como redes sociales y dispositivos móviles. Estas descripciones, identifican dos arquitecturas principales: el estanque de datos y la zona de datos.

3.3.1. Arquitectura de Dos Capas

Esta arquitectura consta de la zona de aterrizaje para datos temporales y transitorios y la capa de almacenamiento permanente para datos sin procesar. Con el tiempo, esta evolucionó a la arquitectura Lambda, enfocada en el procesamiento y consumo de datos. La arquitectura Lambda consta de dos capas de procesamiento: una capa por lotes para datos almacenados en la memoria persistente y una capa de velocidad para datos incrementales no almacenados (Hlupić et al., 2022). Una vez que los datos se almacenan en la memoria persistente, dejan de estar disponibles en la capa de velocidad.

Figura 1

Arquitectura de Dos Capas



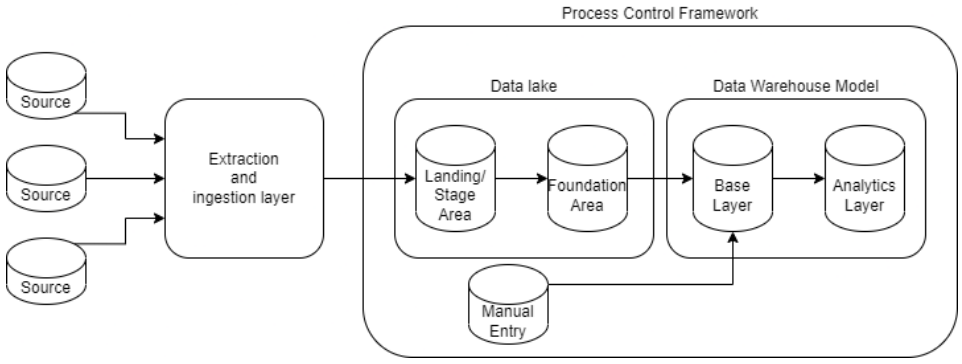
Nota. Adaptada de Two layered Data Lake architecture de Hlupić, T., Oreščanin, D., Ružak, D., & Baranović, M. (2022).

3.3.2. Arquitectura Data Lakehouse

La arquitectura de Data Lakehouse combina los beneficios de Data Warehouse y Data Lake, permitiendo acceso a datos sin procesar y con un modelo bien definido a través de una capa virtualizada. La capa de extracción e ingesta recopila los datos para su procesamiento y almacenamiento temporal en el área de aterrizaje. Los datos son luego almacenados de forma persistente en el área de base, que sirve como fuente para el acceso y análisis de datos (Hlupić et al., 2022).

Figura 2

Arquitectura Data Lakehouse Alto Nivel



Nota. Adaptada de High-level Data Lakehouse architecture de Hlupić, T., Oreščanin, D., Ružak, D., & Baranović, M. (2022).

3.3.3. Arquitectura Del Estanque de Datos (Data Pond)

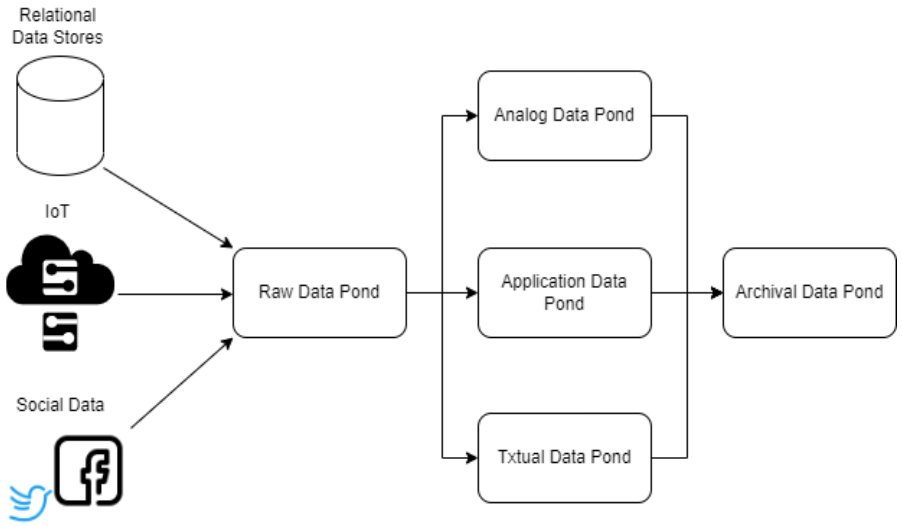
Arquitectura propuesta por Bill Inmon, y está compuesta por:

1. El estanque de datos sin procesar es el primer lugar donde se almacenan los datos recopilados, sin ningún tipo de procesamiento ni metadatos.
2. El estanque de datos analógicos contiene principalmente datos semiestructurados de alta velocidad, como IoT y datos provenientes de APIs, con algunos metadatos aplicados.
3. El estanque de datos de la aplicación es un almacén de datos donde se integran los datos sin procesar a través de procesos ETL, utilizado para brindar soporte al negocio y aplicaciones existentes.
4. El estanque de datos textuales almacena datos no estructurados para su análisis de texto y contextualización, mediante un proceso ETL textual.

5. El estanque de datos de archivo se utiliza para descargar y almacenar datos inactivos menos utilizados de los otros estanques para su uso solo cuando son necesarios para el análisis.

Figura 3

Arquitectura Data Pond



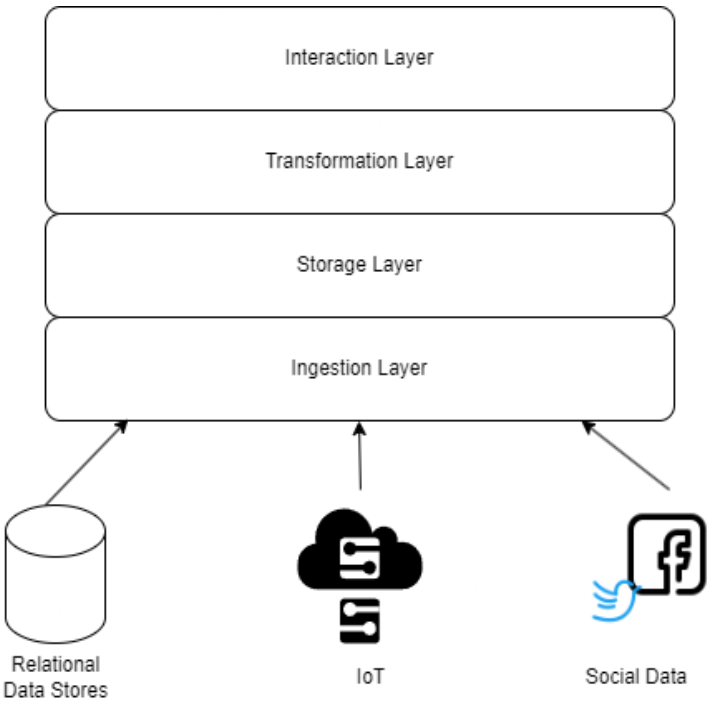
Nota. Adaptada de Data Pond architecture de Hlupić, T., Oreščanin, D., Ružak, D., & Baranović, M. (2022).

3.3.4. Arquitectura Multicapa

Este enfoque se basa en incorporar capas con separación de preocupaciones. Cada capa se comunica con las contiguas y los datos deben canalizarse a través de las cuatro capas (Hlupić et al., 2022). Y describe las capas como se muestra a continuación.

Figura 4

Arquitectura Multicapa



Nota. Adaptada de Layered Data Lake architecture de Hlupić, T., Oreščanin, D., Ružak, D., & Baranović, M. (2022).

Capa de Ingesta se encarga de recopilar datos diversos de diversas fuentes y extraer metadatos iniciales de forma automatizada, almacenándolos en un repositorio específico.

Capa de almacenamiento, alberga tanto los metadatos como los datos sin procesar, brindando soporte para distintas formas y estructuras de datos, y proporciona una interfaz que simplifica la consulta de información.

Capa de Transformación, permite la ejecución escalable de operaciones como limpieza, transformación e integración de datos para obtener una forma final definida, además de crear modelos de acceso a datos según las necesidades de los usuarios.

Capa de Interacción, los usuarios finales tienen acceso a los metadatos y a los datos transformados, permitiéndoles explorar, consultar y visualizar la información almacenada mediante diversas herramientas de visualización.

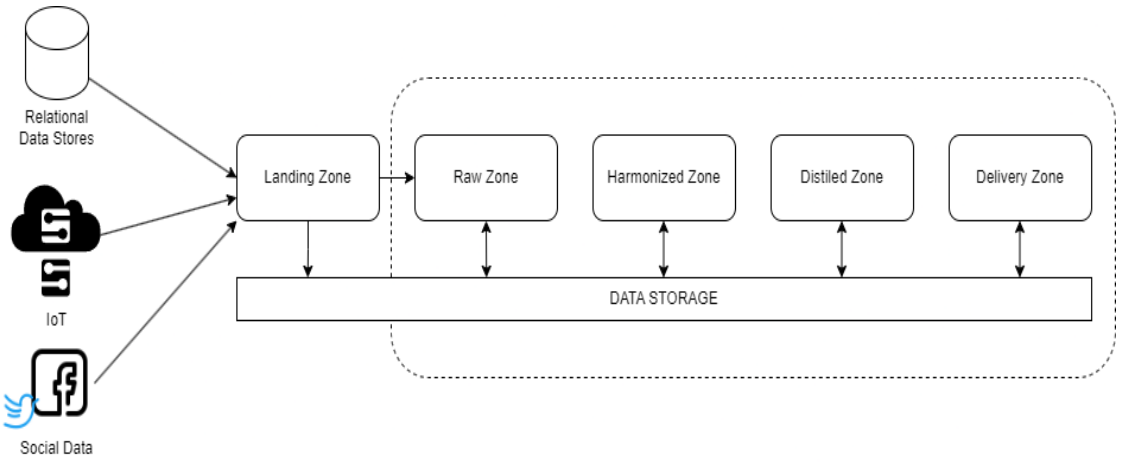
3.3.5. Arquitectura de Zona Basada en Data Vault

El modelo descrito se basa en la arquitectura zonificada según Hlupić et al., 2022. Según esta propuesta, se establecen múltiples zonas con distintos propósitos. La ***zona de aterrizaje*** temporalmente almacena datos en su estado original hasta que son procesados y almacenados de manera permanente. Paralelamente, la ***zona sin procesar*** mantiene los datos en su formato original para asegurar la preservación de la información histórica y ofrecer una fuente para futuros análisis.

Las zonas ***armonizada*** y ***destilada*** siguen el enfoque Data Vault y contienen datos estructurados. La primera incluye datos estructurados sin procesar, mientras que la segunda aplica lógica empresarial a los datos. En contraste, la ***zona de entrega*** proporciona datos procesados para aplicaciones comerciales, mientras que la "zona exploratoria" ofrece un modelo definido para consultas de usuarios.

Figura 5

Arquitectura de zona basada en Data Vault



Nota. Adaptada de Data Vault based zone architecture de Hlupić, T., Oreščanin, D., Ružak, D., & Baranović, M. (2022).

3.4. Gobernanza de Datos en Lagos de Datos

Se entiende por gobernanza de datos al conjunto de procesos, políticas, estándares y controles que aseguran la calidad, integridad, disponibilidad y seguridad de los datos en una organización. Esto implica establecer reglas y procedimientos para la recolección, almacenamiento, procesamiento, uso y distribución de la información.

La gobernanza de datos es esencial en el contexto de Big Data y Data Lakes, ya que estos almacenan grandes cantidades de datos sin procesar. La gestión adecuada de estos datos es fundamental para el éxito de un sistema analítico, ya que la falta de gestión puede convertir el lago de datos en un depósito de datos inútiles. Los autores también compararon los Data Lakes con ecosistemas naturales, destacando la necesidad de gobernanza para garantizar la disponibilidad precisa y oportuna de los datos heterogéneos almacenados. Esta necesidad de gobernanza de datos

se identificó previamente en Data Warehouses, y se espera que siga siendo igualmente importante para los Data Lakes. Un ejemplo concreto de gobernanza de datos se refiere al modelado arquitectónico, donde es crucial rastrear y almacenar las transformaciones entre capas y zonas en los modelos arquitectónicos a través del linaje de datos y el repositorio de metadatos. Esto proporciona la capacidad de almacenar modelos de datos para los mismos datos en todo el lago de datos (Hlupić et al., 2022).

3.5. Sistema de Ingestión

Los sistemas de ingestión de datos están diseñados para trasladar datos desde diferentes fuentes a almacenes de datos centralizados.

Isah y Zulkerine (2018) mencionan que un sistema de ingesta de datos debería poder soportar un alto rendimiento y baja latencia y que hay varias características importantes al considerar las herramientas de ingesta incluyendo, facilidad de instalación, publicar, transportar, consumir y archivar flujos en disco.

Adicionalmente exponen seis requerimientos que deberían cumplir los sistemas de ingestión de datos.

- Integración de fuentes y preprocesamiento: Los flujos de datos se obtienen de fuentes como API HTTP/Web Sockets, API REST, IoT Hubs y colas de mensajes. Integrar datos de múltiples fuentes es desafiante, especialmente al transformarlos a un formato común.

- La adquisición efectiva de flujos implica priorizar fuentes, validar archivos y dirigir datos. Los sistemas deben verificar y filtrar fuentes, idiomas y formatos para integrar sin ruido.

- Tolerancia a fallos y garantías de entrega de mensajes: Los sistemas de ingesta deben ser altamente disponibles para operar de manera continua a pesar de fallas, siendo robustos en el manejo de fallas y minimizando pérdida de datos.

- Procedencia y seguridad: Las tecnologías de mensajería eficientes recopilan datos en tiempo real de muchas fuentes y los distribuyen a múltiples consumidores, incluyendo aplicaciones en tiempo real. La transmisión de mensajes eficaz es esencial para la infraestructura de transmisión de datos.

- Escalabilidad: Un sistema de ingesta debe ser escalable para poder incorporar volúmenes cada vez mayores de datos de múltiples fuentes.

- Contrapresión y enrutamiento: La contrapresión en el procesamiento de flujos de datos ocurre cuando la velocidad de recepción de datos supera la capacidad de ingestión. Para manejar esto, el sistema debe almacenar temporalmente datos en un búfer y permitir su reproducción posterior.

- Extensibilidad: Los sistemas de ingesta deben ser flexibles para trabajar con diversas fuentes y aplicaciones

3.6. Tecnologías

3.6.1. Tecnologías de Ingesta de Datos

Apache Kafka

Es un sistema de cola de mensajes distribuido que permite la ingesta de datos en tiempo real de un sistema a otro. Isah y Zulkerine (2018) implementaron Kafka en la capa de distribución de flujo de datos. Por otro lado, Rooney et al. (2019) de IBM Research, Zurich Laboratory, en el

desarrollo de un lago de datos corporativo eligieron Kafka para la transmisión de datos no estructurados.

Apache NiFi

Plataforma de código abierto diseñada para automatizar y gestionar el flujo de datos entre diferentes sistemas y aplicaciones en una arquitectura de datos distribuida. Isah y Zulkerine (2018) utilizaron esta herramienta para la adquisición, integración y extracción de flujos de datos en su marco de flujo de datos.

Apache Sqoop

Herramienta de código abierto que permite transferir datos entre sistemas relacionales y Hadoop. Sqoop es capaz de importar datos de bases de datos relacionales como MySQL, Oracle, Postgres y SQL Server a HDFS o Hive, o exportar datos desde HDFS o Hive a bases de datos relacionales.

En el estudio de Tunjić, A. (2019), describe todo el proceso de configuración de Sqoop para lograr la ingesta de datos de MSSQL, MySQL y Postgres a una base de datos Hive.

Apache Flume

Es una herramienta de ingesta de datos de código abierto diseñada para recolectar, agregar y mover grandes cantidades de datos de forma escalable y fiable.

Logstash

Es una herramienta de pipeline de procesamiento de datos de código abierto que permite recolectar, limpiar, analizar y enviar datos desde diferentes fuentes a diferentes destinos.

3.6.2. Tecnologías de Almacenamiento de Datos

Existen dos formas principales de almacenamiento para manejar los datos, tales como bases de datos relacionales tradicionales y sistemas de archivos distribuidos como Hadoop HDFS. Mientras que las bases de datos relacionales son adecuadas para almacenar datos estructurados, los sistemas de archivos distribuidos como HDFS son más eficientes para manejar datos no estructurados o semiestructurados, especialmente para grandes cantidades de datos. Sin embargo, es importante tener en cuenta que HDFS debe ser utilizado en conjunto con bases de datos relacionales para manejar datos estructurados de manera eficiente.

Hadoop

Hadoop es una plataforma de procesamiento y almacenamiento de datos distribuido que utiliza HDFS (Hadoop Distributed File System) como su sistema de almacenamiento central. HDFS permite almacenar grandes cantidades de datos de manera redundante y económica, a un costo significativamente menor que los sistemas de almacenamiento tradicionales. El enfoque del lago de datos de Hadoop se basa en almacenar todos los datos en su formato original y realizar el procesamiento ETL (Extract, Transform, Load) mediante las aplicaciones de Hadoop. El ecosistema de Hadoop ofrece varias herramientas para importar y exportar datos a HDFS y procesarlos una vez almacenados.

Apache HBase

Es un sistema de base de datos NoSQL para Hadoop que permite el acceso a los datos en tiempo real y el procesamiento de operaciones de lectura y escritura en grandes volúmenes de datos.

Apache AsterixDB

Es un sistema de gestión de big data, paralelo de código abierto que proporciona una gestión de datos distribuida completa para datos semiestructurados a gran escala. Utiliza el modelo datos NoSQL basado en la extensión JSON. Además, con un lenguaje de consulta expresivo y declarativo.

Amazon Simple Storage Service (S3)

Servicio de almacenamiento en la nube ofrecido por Amazon permite el almacenamiento de grandes volúmenes de datos. S3 es escalable, fiable y de bajo costo, lo que lo convierte en una opción popular para el almacenamiento de datos en un lago de datos.

3.6.3. Tecnologías de Procesamiento de Datos

Apache Spark

Motor de procesamiento de datos distribuido que proporciona una interfaz de programación unificada para el procesamiento de datos en batch y en tiempo real. Spark es compatible con Hadoop y se utiliza para procesar grandes volúmenes de datos en un cluster de nodos.

Apache Hadoop MapReduce

Es un framework de procesamiento de datos distribuido que permite procesar grandes volúmenes de datos en paralelo. MapReduce divide los datos en pequeños fragmentos y los procesa en varios nodos, lo que permite un procesamiento rápido y escalable.

Apache Flink

Es un motor de procesamiento de datos distribuido que proporciona una interfaz de programación unificada para el procesamiento de datos en tiempo real y batch.

Elasticsearch

Herramienta que puede ser utilizada para procesar datos en un lago de datos al proporcionar una interfaz de búsqueda y análisis de datos sobre los datos almacenados en HDFS o cualquier otro sistema de almacenamiento.

3.6.4. Elección tecnología

Una vez finalizada la revisión y análisis de diversas herramientas disponibles para la ingesta, procesamiento y visualización de datos, se logró determinar que Elastic (Elasticsearch y Kibana) proporciona un conjunto integral de características que satisfacen los requisitos del caso de estudio planteado. Este estudio se centra en la industria petrolera en Texas, donde la necesidad de procesar diversas fuentes de información con tipos de datos heterogéneos, que abarcan desde datos estructurados hasta datos semiestructurados y no estructurados, presenta un desafío. Además, considerando la magnitud de los conjuntos de datos involucrados, el caso de estudio se detalla en la Sección 8, Elastic demuestra ser una solución integral que cumple con los criterios fundamentales necesarios para llevar a cabo este análisis de datos. Las razones específicas que respaldan esta elección se discuten a continuación.

- Facilidad de Despliegue y Uso: Una de sus ventajas de Elastic es la sencillez con la cual se logró desplegar todo el stack desde un contenedor en Docker. Esta facilidad de configuración y gestión reduce la complejidad operativa y ayuda a enfocarse en el desarrollo del sistema.

- Soporte de Diversidad de Datos: Elastic también se destaca por su capacidad de manejar una amplia gama de fuentes de datos, incluyendo datos estructurados, no estructurados y semiestructurados. Esta flexibilidad es esencial para abordar la diversidad de datos presentes en el caso de estudio de la industria petrolera en el estado de Texas.

- Motor de Búsqueda y Análisis: Elastic cuenta con un potente motor de búsqueda que proporciona la capacidad necesaria para realizar búsquedas rápidas y análisis avanzados, piezas fundamentales para extraer información significativa del conjunto de datos.

- Facilidad de Uso de Lenguaje de Programación: Además las herramientas de Elastic, como lo es Elasticsearch permiten una fácil interacción a través de lenguajes de programación como Python. Permitiendo hacer tareas de indexación de datos, crear índices y otras operaciones necesarias para el desarrollo del proyecto.

- Elasticidad y Rendimiento: Elastic ofrece una arquitectura escalable y robusta, diseñada para gestionar grandes volúmenes de datos, garantizando un rendimiento adecuado de acuerdo con las necesidades cambiantes de las cargas de trabajo.

4. Estado del Arte

En esta sección, se presenta una descripción general de las propuestas de investigación más relevantes.

En su artículo Zagan & Danubianu, 2021, hablan sobre un caso de estudio en el que demuestran cómo utilizaron CoreDB para almacenar más de 15 millones de datos de redes sociales

relacionados con el presupuesto de salud del gobierno de Australia. Posteriormente, crearon una base de datos relacional para almacenar información detallada sobre el programa presupuestario de atención médica.

Esta solución no solo posibilita el procesamiento de datos estructurados, sino también de datos no estructurados almacenados en su formato original en el lago de datos. Además, simplifica el proceso de transformación de datos mediante el uso de expresiones U-SQL SELECT, a diferencia del procesamiento basado en Hadoop/Spark, que requiere la adaptación a patrones de procesamiento específicos como MapReduce y la implementación de funciones de procesamiento dedicadas (Zagan & Danubianu, 2021).

En otro estudio relevante, se aborda la modernización del tráfico aéreo en EE. UU. (NextGen) y Europa (SESAR), centrándose en el intercambio de datos como clave para mejorar la eficiencia y seguridad (Raju, Mital y Finkelsztein, 2018). Para capitalizar el potencial de grandes volúmenes de datos de vuelo, meteorológicos y aeronáuticos, SGT y el Centro Nacional de Sistemas de Transporte Volpe desarrollaron un prototipo de Data Lake basado en la nube. Este sistema almacena y procesa información proveniente de diversas fuentes de la FAA, lo que permite llevar a cabo análisis avanzados de datos, tanto estructurados como no estructurados. Además, se emplean herramientas tanto de código abierto como comerciales, tales como PostgreSQL, ElasticLogstash-Kibana y Tableau, para procesar y visualizar la información.

En su trabajo, Cravero, Lefiguala, Tralma y González (2020) proponen una arquitectura de Data Lake para la Dirección General Impositiva (DGI) de una Universidad. La propuesta se fundamenta en la estructura diseñada por Ravat y Zhao, implementando tres zonas de almacenamiento en Amazon Web Service (AWS) y siguiendo pautas de Sawadogo para la gestión de metadatos. El proceso involucra la carga de documentos en la zona "Raw Data", seguida por un

proceso ETL para almacenar datos procesados en la zona "Data Processing". Finalmente, los indicadores de gestión se almacenan en la zona "Data Querying". La implementación aprovecha herramientas como AWS Glue, Redshift y Athena para un manejo eficaz de datos y metadatos (Cravero et al., 2020).

En otro contexto, se analiza una aplicación práctica de los grandes lagos de datos en el ámbito de la atención médica personalizada. Este estudio destaca que la personalización de los servicios de atención médica se basa en el uso de datos relacionales de pacientes y análisis de big data para adaptar las recomendaciones de medicamentos. Sin embargo, se destaca que la mayoría de los datos sanitarios están en formato no estructurado, lo que requiere un esfuerzo considerable para convertirlos en una forma relacional. El artículo propone una nueva arquitectura de lago de datos que tiene como objetivo reducir el tiempo de procesamiento de datos y mejorar la precisión de los análisis en este contexto (Cuzzocrea, 2021).

Otros trabajos abordan casos de estudio adicionales, como el desarrollo de redes inteligentes emergentes. Estos avances permiten agregar y analizar grandes cantidades de datos para diversas aplicaciones de redes inteligentes. Sin embargo, se destaca la necesidad de abordar los desafíos de escalabilidad y capacidad de almacenamiento y procesamiento en los sistemas tradicionales de gestión de datos de redes inteligentes. Para ello, se presenta un ecosistema de big data de red inteligente basado en la arquitectura Lambda de última generación (Cuzzocrea, 2021).

En su artículo, Hai, Geisler y Quix (2016) presentan Constance como una solución práctica y flexible para los desafíos en la gestión de Data Lakes. Este sistema destaca por su enfoque integral en la gestión de metadatos estructurales y semánticos, ofreciendo una interfaz unificada para el procesamiento de consultas. A diferencia de otras propuestas, Constance se centra en la ingesta de datos, la gestión de metadatos y la respuesta a consultas, proporcionando también

mecanismos básicos de seguridad y procedencia. Se destaca que, según los autores, un Data Lake va más allá de ser simplemente un repositorio de almacenamiento en un sistema de archivos Hadoop, ya que este último no abarca todas las funcionalidades de metadatos necesarias para un Data Lake completo.

En su informe, Schoenenwald et al. (2021) proponen una solución completa que utiliza planes de ejecución de PySpark y se basa en el componente de código abierto Spline. Esto permite adquirir de manera confiable metadatos de linaje e identificar interdependencias. Además, transforman los datos procesados en un modelo de datos expansible, facilitando la extracción de estructuras gráficas para el linaje de datos a niveles de granularidad tanto gruesos como finos. En la etapa final, la solución no solo visualiza el linaje de datos extraído mediante una aplicación web moderna, sino que también se integra con el Cloud Data Hub de código abierto

Otro enfoque se centra en Azure Data Lake Store (ADLS), un sistema de archivos completamente administrado y escalable diseñado para respaldar una amplia gama de análisis de big data en Azure. El artículo proporciona una descripción detallada de la arquitectura, puntos de diseño y rendimiento de ADLS (Cuzzocrea, 2021).

Finalmente, se abordan los desafíos de mantenimiento de los grandes lagos de datos, reconociendo que estos almacenan grandes cantidades de datos sin procesar de diversas fuentes. A medida que se incorporan más conjuntos de datos en un lago de datos, surge la necesidad de técnicas eficientes para perfilarlos y detectar relaciones entre sus esquemas, lo que se conoce como coincidencia de esquemas holísticos. Los autores proponen un nuevo enfoque de poda temprana para mejorar la eficiencia en este proceso, utilizando diferentes tipos de metadatos para filtrar comparaciones de coincidencia de esquemas y así detectar conjuntos de datos similares con mayor efectividad (Cuzzocrea, 2021).

5. Metodología

La metodología de prototipado evolutivo es un enfoque iterativo para el desarrollo de un sistema que se basa en la creación continua de prototipos simplificados.

La idea principal es recibir retroalimentación temprana y mejorar continuamente el diseño a través de la iteración con las siguientes etapas:

1. Investigación:

Búsqueda en literatura: identificación de términos clave y revisión de artículos relevantes.

Análisis de los artículos encontrados.

2. Definición del entorno de trabajo y requerimientos de herramientas:

Establecimiento de requisitos y evaluación de herramientas.

Tabla comparativa de herramientas seleccionadas.

3. Desarrollo del caso de estudio:

Especificación de requisitos del sistema.

Definición de objetivos y alcance del sistema.

Especificación de casos de uso.

4. Implementación del sistema:

Desarrollo e implementación de una solución independiente de la plataforma.

Generación del modelo de caso de estudio.

5. Verificación y validación:

Identificación de posibles errores y plan de pruebas.

6. Evaluación de resultados:

Verificación de que el sistema cumple los requisitos establecidos.

6. Desarrollo e Implementación de Arquitectura

La revisión de diversas arquitecturas proporcionó una visión de sus características y enfoques. Como resultado, se ha elaborado una matriz de comparación, presentada en la Tabla 2, que aborda específicamente tres arquitecturas seleccionadas. Estas arquitecturas se eligieron por presentar una complejidad de implementación considerada media, lo que implica que ofrecen un equilibrio entre funcionalidad y practicidad.

Tabla 2

Comparación de Arquitecturas: Data Pond, Multicapa y Dos Capas (Lambda)

Característica	Arquitectura Dos Capas (Lambda)	Arquitectura Multicapa	Estanque de Datos (Data Pond)
Enfoque Principal	Dos capas para procesamiento por lotes y en tiempo real	Separación de responsabilidades en capas de ingestión, almacenamiento, transformación e interacción	Modular y especializado en varios estanques de datos.
Gestión de Datos Crudos	Utiliza dos capas para procesar datos en tiempo real y por lotes	Capa de Ingesta recopila datos diversos	Almacena datos recopilados sin procesamiento
Almacenamiento	No cuenta con capa de almacenamiento de datos crudos, solo persiste los procesados	Capa de Almacenamiento alberga los diversos tipos datos sin procesar y procesados	Múltiples estanques para clasificar los diferentes tipos de datos

Procesamiento de Datos	Procesamiento en dos capas para datos en tiempo real y por lotes	Capa de Transformación para limpieza, transformación e integración	No es su enfoque, pero se usan procesos ETL específicos para cada estanque
Acceso a Datos	Acceso a datos procesados a través de capas específicas	Interfaz en la Capa de Interacción para usuarios finales	A través de interfaces específicas para cada estanque
Visualización de Datos	No es su enfoque, se usan herramientas de consulta mas no de visualización	Herramientas de visualización en Capa de Interacción	No se especifica, depende del estanque específico
Tipo de Datos Soportados	Soporte diverso tipos de datos crudos en ambas capas	Diversas formas y estructuras de datos en Capa de Almacenamiento y Transformación	Diversos tipos de datos en estanques especializados
Enfoque en Tiempo Real	Procesamiento en tiempo real en capa específica	Capa de Ingesta y Capa de Interacción pueden admitir tiempo real	No se especifica

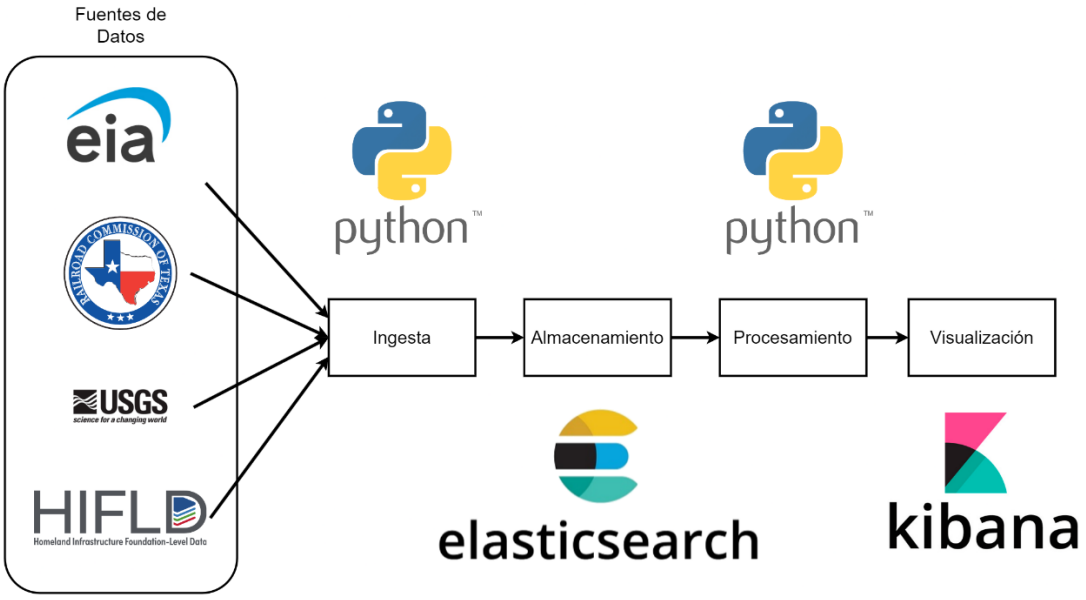
Tras revisar las principales características de las arquitecturas presentadas en la matriz comparativa, se ha tomado la decisión de optar por la arquitectura multicapa. Esta elección se base en su adaptabilidad, que permite gestionar eficientemente la diversidad de datos. La arquitectura multicapa ofrece ventajas, como el almacenamiento persistente tanto de datos procesados como sin procesar. Además, destaca por contar con capas específicas diseñadas para el procesamiento y la visualización de información. En comparación con alternativas como la arquitectura de dos capas o lambda, que se enfocan principalmente en el procesamiento de datos, y la arquitectura de estanque, centrada en el almacenamiento según tipos de datos, la arquitectura multicapa se presenta como la opción más completa.

6.1. Implementación de Arquitectura

En esta sección, se detalla la implementación y desarrollo de la arquitectura propuesta, que implica la utilización de Elasticsearch y Kibana como componentes fundamentales. Se proporcionan detalles sobre la configuración y el uso de estos sistemas, así como el lenguaje de programación Python para interactuar con Elasticsearch.

Figura 6

Implementación de Arquitectura de Procesamiento de Datos



Nota. Ilustra el flujo general de los datos desde fuente hasta la visualización.

6.2. Configuración del Entorno

Se implementó un entorno de desarrollo utilizando contenedores Docker para Elasticsearch y Kibana. Esto permitió un despliegue local eficiente y una configuración simplificada para el desarrollo y prueba del modelo de arquitectura propuesto.

Elasticsearch en Docker: Se desplegó Elasticsearch en un contenedor Docker, lo que facilitó su configuración y puesta en marcha sin la necesidad de un servidor dedicado.

Kibana en Docker: De manera similar, Kibana se ejecutó en un contenedor Docker, lo que proporciona una interfaz de usuario accesible para la visualización y exploración de los datos.

La decisión de trabajar de forma local en lugar de utilizar un servidor dedicado se basó en la necesidad de una configuración y despliegue ágil.

6.3. Interfaz Python para Elasticsearch

Es importante destacar que Elasticsearch ofrece la posibilidad de integración con varios lenguajes de programación, lo que facilita su implementación en una amplia gama de proyectos. Sin embargo, se optó por Python debido a sus ventajas notables en manipulación de datos y su diversa colección de bibliotecas, lo que facilita el procesamiento y análisis de datos a gran escala de manera efectiva y eficiente.

Para la interacción con Elasticsearch, se utilizaron bibliotecas específicas de Python, como Elasticsearch. Esto posibilitó establecer conexiones y realizar operaciones con los datos de Elasticsearch de manera sencilla.

6.4. Ingestión de Datos

Se diseñó un sistema de ingestión para recopilar y almacenar datos a gran escala en Elasticsearch. Esto incluyó la definición de índices, mapeos y configuraciones específicas para optimizar el almacenamiento y búsqueda de datos. Se implementaron técnicas de ingestión en tiempo real y por lotes para mejorar los tiempos de carga de datos al sistema.

Fuentes de Datos: Administración de Información Energética (EIA), Comisión de Ferrocarriles de Texas (RRC), Servicio Geológico de Estados Unidos (USGS) y Homeland Infrastructure Foundation Data (HIFLD).

Ingestión con Python: Para la extracción de datos de diversas fuentes, se emplearon scripts en Python. Se utilizan bibliotecas como ‘requests’ para obtener datos a través de una API, que es el caso de la información proveniente de la EIA, adicionalmente técnicas para el procesamiento de datos de archivos planos, ya sean estructurados como no estructurados.

6.5. Almacenamiento

Almacenamiento en Elasticsearch: Los datos se ingresan y almacenan en Elasticsearch, un motor de búsqueda y análisis. Elasticsearch es eficiente para la búsqueda y análisis de grandes volúmenes de datos, ya sean numéricos, de texto, geográficos, estructurados, no estructurados.

6.6. Procesamiento de Datos

Se implementaron diferentes técnicas de procesamiento, transformación y enriquecimiento de datos. En este contexto se utiliza Python y bibliotecas como Pandas para llevar a cabo operaciones de filtrado, transformación y cruce de datos. Pandas proporciona herramientas flexibles y eficientes para el análisis de datos. Además, se implementa procesamiento en paralelo

mediante la biblioteca multiprocessing en Python, permitiendo aprovechar la capacidad de procesamiento de múltiples núcleos de la máquina para acelerar el procesamiento de datos.

6.7. Visualización y Análisis

Kibana se utilizó como herramienta central para visualizar y analizar los datos almacenados en Elasticsearch ya que se integra naturalmente con Elasticsearch. Kibana permite la creación de tableros interactivos y visualizaciones, facilitando la exploración y comprensión de los datos procesados.

Esta metodología proporciona una base sólida para la implementación y evaluación de la arquitectura propuesta, centrándose en la utilización efectiva de Elasticsearch, Kibana y Python para la ingestión, procesamiento y análisis de datos a gran escala.

6.8. Razones de la Arquitectura

En esta sección, se habla de las razones y consideraciones que apoyan la elección de la arquitectura implementada.

6.8.1. Escalabilidad y Rendimiento

Esta arquitectura permite manejar grandes volúmenes de datos y escalar eficientemente a medida que aumenta la carga de trabajo. Elasticsearch es conocido por su escalabilidad horizontal y rendimiento en búsquedas y análisis.

6.8.2. Flexibilidad en el Procesamiento

La combinación de Python y bibliotecas como Pandas proporciona flexibilidad en el procesamiento de datos. Pandas es potente y versátil para realizar diversas operaciones en datos tabulares, y Python permite una programación flexible y fácil integración con otras herramientas.

En situaciones en las que se requiere una herramienta con mayor potencia en el procesamiento de datos, Elasticsearch admite la integración con Apache Spark.

6.8.3. Flexibilidad Integración de Datos de Múltiples Fuentes

La diversidad de fuentes de datos, a su vez proporcionan diferentes tipos de datos, Elasticsearch provee una amplia variedad de tipos de datos (Texto, Formas, Números, Vectores, Histogramas, Series temporales o de fechas, Puntos geográficos/formas geométricas, Datos no estructurados (JSON), Datos estructurados).

Esta capacidad de Elasticsearch para adaptarse a diversos tipos de datos se traduce en una integración sin problemas de información heterogénea, facilitando un análisis unificado y eficiente en el sistema.

6.8.4. Análisis en Tiempo Real

La implementación de técnicas de ingestión en tiempo real refleja la capacidad de la arquitectura de analizar datos en tiempo real o con latencias mínimas. Elasticsearch, junto con Kibana, es conocido por su capacidad para admitir consultas y visualizaciones en tiempo real.

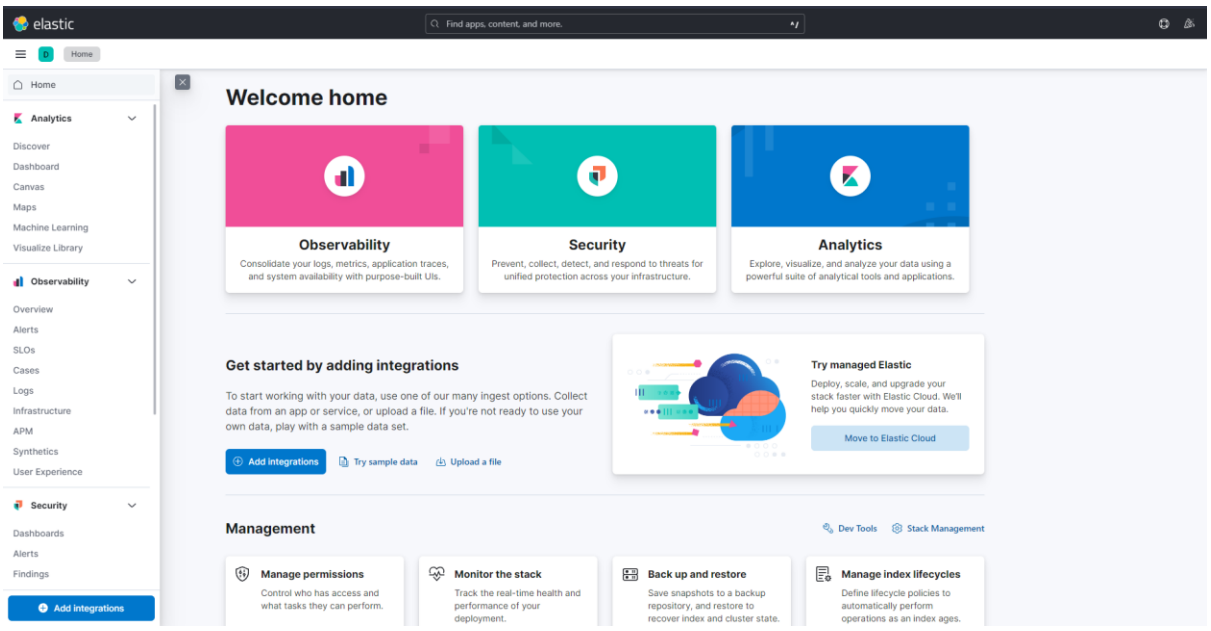
6.8.5. Facilidad de Integración con Herramienta de Visualización

Kibana se selecciona como herramienta de visualización debido a su integración natural con Elasticsearch, también tiene la capacidad para crear tableros interactivos y visualizaciones de datos de manera intuitiva.

Adicionalmente Kibana cuenta con variedad de plugins que permiten simplificar la visualización de datos, un ejemplo de esta versatilidad es la sencilles con la que permite la visualización de datos geográficos.

Figura 7

Panel General de Visualización Elastic



7. Caso de Estudio Aplicado

En esta sección, no solo se detalla el caso de estudio específico en el cual se aplicó el modelo de arquitectura propuesto, sino que también se explora varios escenarios y el proceso de selección que llevó a este escenario particular.

En la revisión de la literatura, se exploraron varios casos de estudio que se consideraron como opciones viables para su implementación. Por ejemplo, Melchor-Uceda et al. (2021) propusieron un sistema de interoperabilidad de datos hospitalarios; sin embargo, este caso de estudio presentaba desafíos significativos en cuanto a la protección de datos, dado que involucraba información sensible.

Otro caso de estudio, desarrollado por Liu, R et al. (2020), se centró en la creación de un lago de datos para almacenar grandes volúmenes de datos sin procesar. Aunque esta propuesta fue

interesante, la adquisición de grandes volúmenes de datos resultó ser poco práctico para el escenario de una empresa comercializadora de automóviles.

Durante el proceso de selección, se centró la atención en el sector de la Industria Petrolera. A pesar de los esfuerzos por buscar información dentro de Colombia, no se logró encontrar datos adecuados. Por lo tanto, se amplió la búsqueda a Estados Unidos, donde se identificaron entidades que proporcionaban información relevante del sector con grandes volúmenes de datos. Estas entidades, junto con sus amplias colecciones de datos, se presentaban como la elección indicada para poner a prueba las capacidades de la arquitectura implementada, fundamentando así la elección de este caso de estudio.

El caso de estudio específico en el que se aplicó el modelo de arquitectura propuesto fue seleccionado dentro de la industria petrolera en el estado de Texas. Esta elección se fundamenta en la necesidad de abordar diversidad de fuentes de datos y tipos de datos no estructurados y semiestructurados.

7.1. Requerimientos del Caso de Estudio

Para llevar a cabo el caso de estudio en la industria petrolera en Texas, es esencial definir los requisitos y criterios específicos que guiarán la implementación y análisis. Se consideran algunos requisitos:

Disponibilidad de Datos:

Verificar la disponibilidad de datos históricos y actuales sobre la producción de petróleo en Texas, incluyendo información a nivel de campo, condado y distrito. Además, asegurar la disponibilidad de datos históricos relacionados con el precio del petróleo para un análisis.

Formato de Datos:

Asegurarse de que los datos estén disponibles en formatos compatibles con las herramientas utilizadas, tales como JSON, CSV u otros, para facilitar la integración y el procesamiento.

Datos Geoespaciales:

La herramienta debe ser capaz de procesar de manera eficiente datos geoespaciales, incluyendo las coordenadas de los pozos petroleros en Texas, como los proporcionados por Homeland Infrastructure Foundation (HIFLD), así como datos geológicos provenientes del Servicio Geológico de Estados Unidos (USGS). Además, asegurarse de que la visualización de esta información sea sencilla y accesible para facilitar el análisis geográfico.

Escalabilidad:

Evaluar la capacidad de la arquitectura para escalar, enfocada en gestionar de manera eficaz tanto el volumen de datos actual como las demandas que puedan surgir debido a un potencial crecimiento futuro de datos o cambios en los requisitos.

7.2. Fuentes de Datos

Se aprovecharon diversas fuentes de datos, cada una proporcionando un tipo de información esencial para el análisis integral de la industria petrolera en Texas:

7.2.1. Energy Information Administration (EIA)

Se utilizó EIA para extraer información sobre los precios del petróleo a lo largo del tiempo. Estos datos proporcionaron una perspectiva histórica y actualizada sobre la fluctuación de precios en la industria petrolera.

La extracción de información se realizó mediante la API que dispone la EIA, el formato de la respuesta de la API son tipo JSON. La API está disponible en (https://www.eia.gov/opendata/browser/petroleum/pri/spt).

Figura 8

API Web de EIA

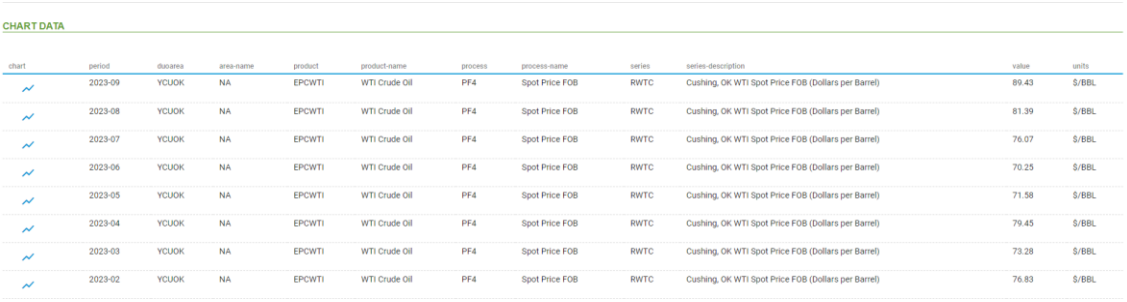


CHART DATA

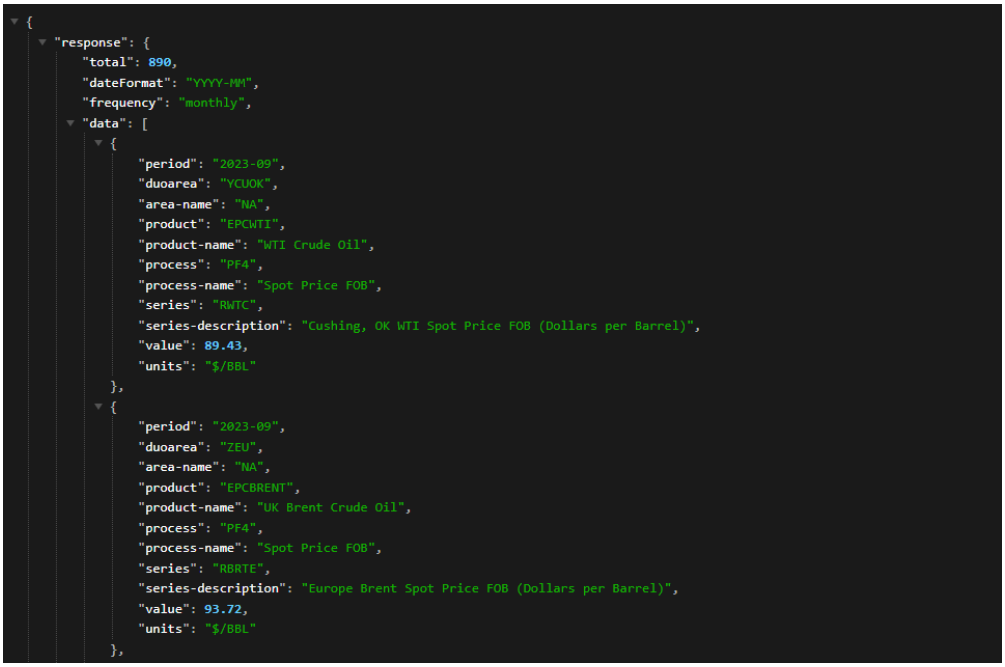
chart	period	duoarea	area-name	product	product-name	process	process-name	series	series-description	value	units
	2023-09	YCUOK	NA	EPCWTI	WTI Crude Oil	PF4	Spot Price FOB	RWTC	Cushing, OK WTI Spot Price FOB (Dollars per Barrel)	89.43	$/BBL
	2023-08	YCUOK	NA	EPCWTI	WTI Crude Oil	PF4	Spot Price FOB	RWTC	Cushing, OK WTI Spot Price FOB (Dollars per Barrel)	81.39	$/BBL
	2023-07	YCUOK	NA	EPCWTI	WTI Crude Oil	PF4	Spot Price FOB	RWTC	Cushing, OK WTI Spot Price FOB (Dollars per Barrel)	76.07	$/BBL
	2023-06	YCUOK	NA	EPCWTI	WTI Crude Oil	PF4	Spot Price FOB	RWTC	Cushing, OK WTI Spot Price FOB (Dollars per Barrel)	70.25	$/BBL
	2023-05	YCUOK	NA	EPCWTI	WTI Crude Oil	PF4	Spot Price FOB	RWTC	Cushing, OK WTI Spot Price FOB (Dollars per Barrel)	71.58	$/BBL
	2023-04	YCUOK	NA	EPCWTI	WTI Crude Oil	PF4	Spot Price FOB	RWTC	Cushing, OK WTI Spot Price FOB (Dollars per Barrel)	79.45	$/BBL
	2023-03	YCUOK	NA	EPCWTI	WTI Crude Oil	PF4	Spot Price FOB	RWTC	Cushing, OK WTI Spot Price FOB (Dollars per Barrel)	73.28	$/BBL
	2023-02	YCUOK	NA	EPCWTI	WTI Crude Oil	PF4	Spot Price FOB	RWTC	Cushing, OK WTI Spot Price FOB (Dollars per Barrel)	76.83	$/BBL

Nota. Visualización de la información en el API Dashboard de la página oficial de la EIA. Tomada de Energy Information Administration (https://www.eia.gov/opendata/)

Para acceder a esta fuente de datos, se debe registrar y obtener una API key a través del sitio oficial de la EIA. Una vez obtenida la clave, se utilizó para consultar la API y adquirir la información requerida en formato JSON.

Figura 9

API Precios de Petróleo



```
{
  "response": {
    "total": 890,
    "dateFormat": "YYYY-MM",
    "frequency": "monthly",
    "data": [
      {
        "period": "2023-09",
        "duoarea": "YCUOK",
        "area-name": "NA",
        "product": "EPCWTI",
        "product-name": "WTI Crude Oil",
        "process": "PF4",
        "process-name": "Spot Price FOB",
        "series": "RWTC",
        "series-description": "Cushing, OK WTI Spot Price FOB (Dollars per Barrel)",
        "value": 89.43,
        "units": "$/BBL"
      },
      {
        "period": "2023-09",
        "duoarea": "ZEU",
        "area-name": "NA",
        "product": "EPCBRENT",
        "product-name": "UK Brent Crude Oil",
        "process": "PF4",
        "process-name": "Spot Price FOB",
        "series": "RBRTE",
        "series-description": "Europe Brent Spot Price FOB (Dollars per Barrel)",
        "value": 93.72,
        "units": "$/BBL"
      },
```


Para lograr indexar esta información en Elasticsearch, primero se debe definir la estructura del índice que se quiere cargar.

```
price_sett = {
"mappings": {
    "properties": {
        "period": {
            "type": "date",
            "format": "yyyyMM",
        },
        "product": {
```

```
                "type": "text",
            },
            "product-name": {
                "type": "text",
            },
            "units": {
                "type": "text",
            },
            "value": {"type": "float"},
        }
    }
}
```

Una vez que la información ha sido indexada en Elasticsearch, el siguiente paso es crear una visualización en Kibana. Para ello, se siguen los siguientes pasos:

Acceder a la sección de Management: En el panel de navegación de Kibana, selecciona la opción "Management". Esta sección es fundamental para la configuración y gestión de datos.

Seleccionar Kibana Data Views: Dentro de la sección de Management, se busca y selecciona "Kibana Data Views". Esta es la herramienta que te permitirá crear vistas personalizadas de tus datos.

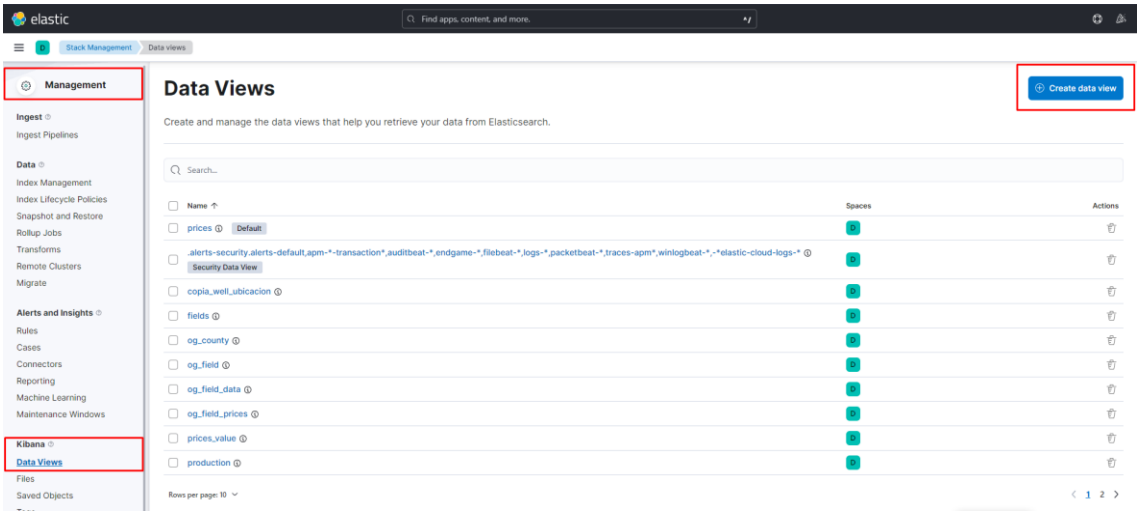
Crear una nueva Data View: Se hace clic en "Create Data View" para iniciar el proceso de creación de una nueva vista de datos.

Asignar un nombre a la vista: Se debe proporcionar un nombre descriptivo para la vista que se está creando. Este nombre ayudará a identificar y organizar las diferentes visualizaciones dentro de Kibana.

Seleccionar el índice correspondiente: A continuación, elige el índice de Elasticsearch que contiene los datos que deseas visualizar en esta vista.

Figura 10

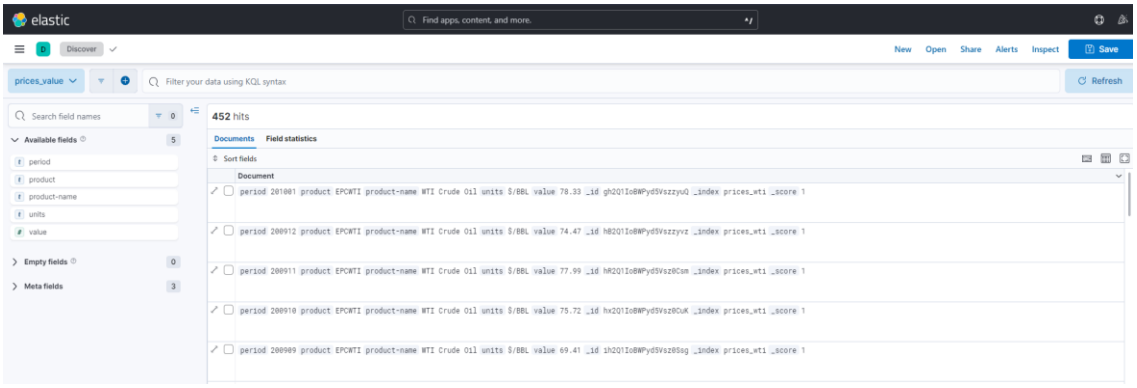
Panel de Data Views Elastic



El resultado de la vista se presenta en la siguiente imagen.

Figura 11

Visualización de Index Price



7.2.2. Comisión de Ferrocarriles de Texas (Railroad Commission of Texas)

La Comisión de Ferrocarriles de Texas fue una fuente crucial para obtener datos sobre la producción de petróleo a nivel de campo, condado y distrito en el estado de Texas. Esta información proporcionó la base esencial para llevar a cabo un análisis detallado y cronológico de la producción petrolera en la región. Es importante resaltar que los conjuntos de datos obtenidos abarcan no sólo datos de producción, sino también información general de los campos de petróleo.

Adicionalmente hay que mencionar que los datos se presentaron en diversos formatos, incluyendo ASCII Format, dBase Format y EBCDIC Format. Los conjuntos de datos están disponibles para descarga en el siguiente enlace (https://www.rrc.texas.gov/resource-center/research/data-sets-available-for-download/).

7.2.3. Servicio Geológico de Estados Unidos (USGS)

Se accedió al Servicio Geológico de Estados Unidos para obtener datos geológicos de todo el estado de Texas. Esta información fue esencial para comprender la geología subyacente que

influye en la producción y exploración de petróleo. La información geológica fue extraída de (https://mrdata.usgs.gov/geology/state/).

Este conjunto de datos está compuesto por cuatro formatos distintos que se complementan mutuamente:

DBF (Database File): Este formato almacena datos en una estructura de tabla. Contiene información detallada sobre atributos específicos de las características geológicas, lo que permite una categorización y clasificación precisa.

PRJ (Projection File): Proporciona información sobre el sistema de coordenadas y proyección utilizados en la representación cartográfica de los datos geológicos. Es esencial para la correcta ubicación y visualización en un sistema de coordenadas específico.

SHP (Shapefile): Es un formato ampliamente utilizado para representar datos geoespaciales. Contiene información geométrica sobre puntos, líneas y polígonos que representan elementos geológicos y su ubicación en el espacio.

SHX (Shape Index File): Este archivo de índice complementa el archivo SHP y acelera la recuperación de información espacial.

La geología de Texas se logró cargar a la herramienta mediante una integración disponible llamada GeoJSON. Esta integración requiere los cuatro archivos mencionados previamente para llevar a cabo el proceso de carga.

Figura 12

Panel Integraciones Elastic

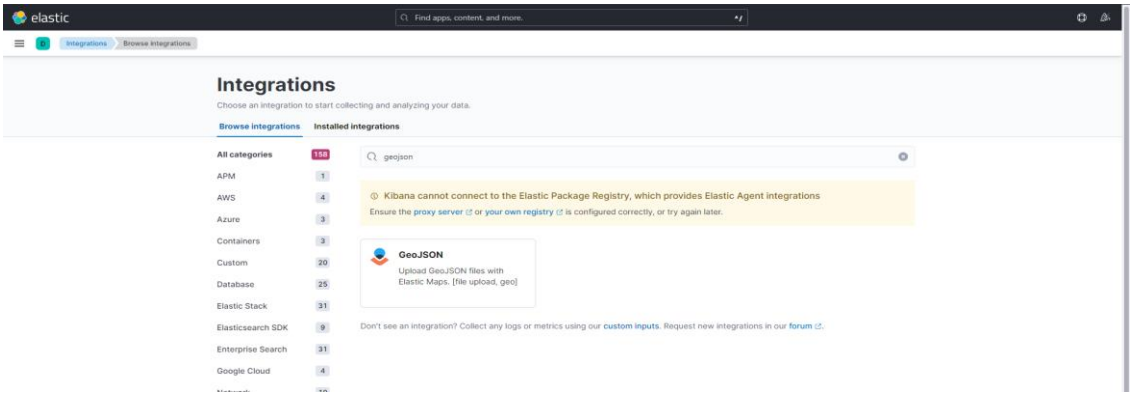
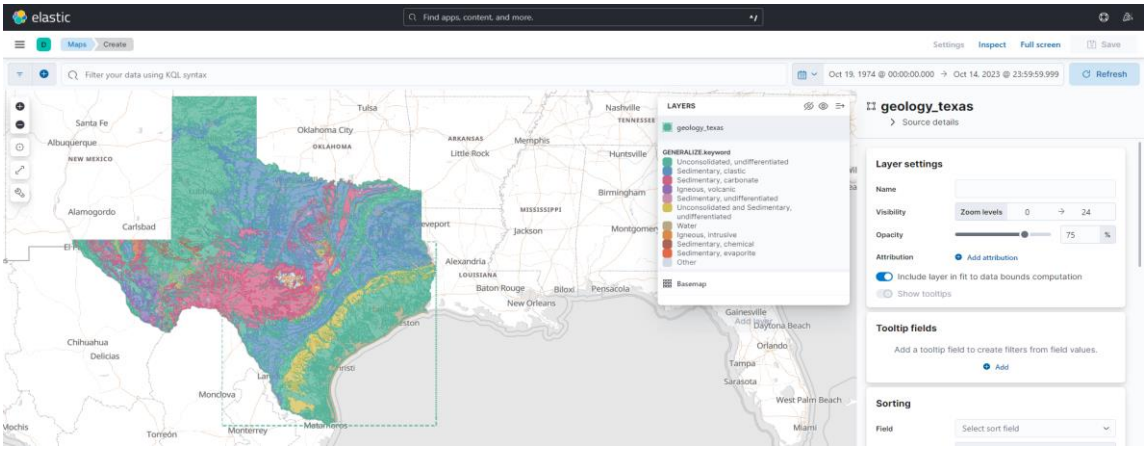


Figura 13

Visualización Datos Geológicos



7.2.4. Homeland Infrastructure Foundation (HIFLD)

Se utilizaron los datos proporcionados por HIFLD para extraer las coordenadas de los pozos petroleros en Texas. Estos datos geoespaciales fueron cruciales para ubicar y analizar la distribución de los pozos en relación con la geología subyacente en cada campo petrolero.

Para llevar a cabo la carga de estos datos, se aprovechó la misma integración que se utilizó para cargar la información geológica.

Los datos geoespaciales fueron extraídos de la categoría de energía están disponibles en (https://hifld-geoplatform.opendata.arcgis.com/).

7.3. Procesamiento y Análisis de Datos

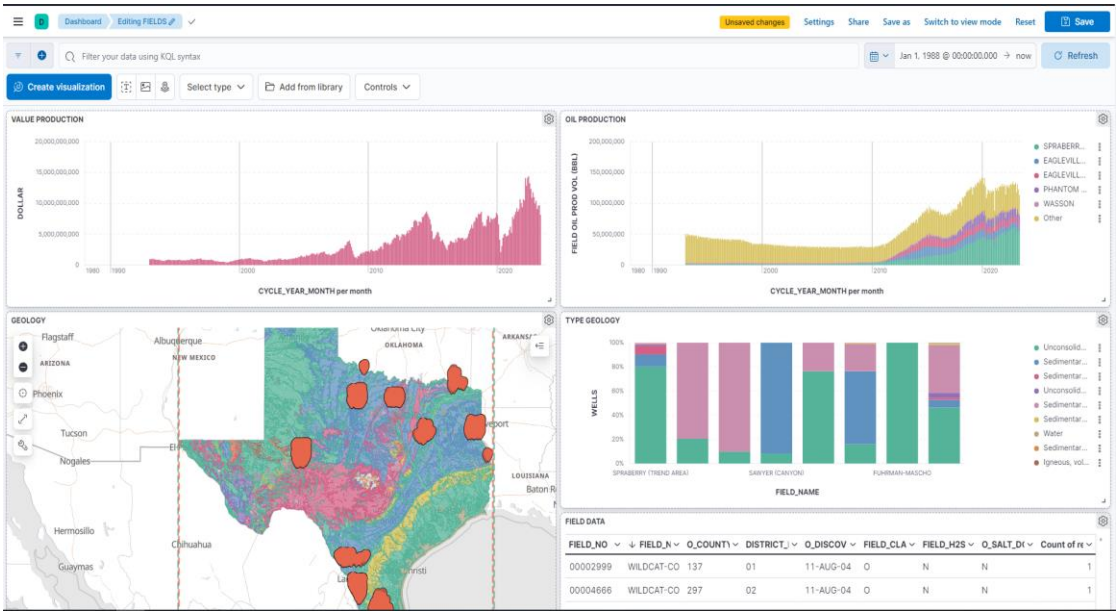
Utilizando el modelo de arquitectura propuesto, se llevó a cabo un proceso de integración de datos procedentes de estas diversas fuentes. Se realizaron operaciones de limpieza, transformación y enriquecimiento para preparar los datos para su análisis.

El análisis se enfocó concretamente, en el análisis de la producción para los diferentes campos petroleros de Texas, donde se puede cuantificar monetariamente el valor de la producción de petróleo, adicionalmente se puede hacer una identificación del porcentaje de pozos que contienen determinada geología y también se incluyeron datos generales del campo tales como si contenía o no un domo de sal, si contiene ácido sulfhídrico H2S entre otros.

A continuación, se presenta una vista general del panel

Figura 14

Panel de Visualización Final



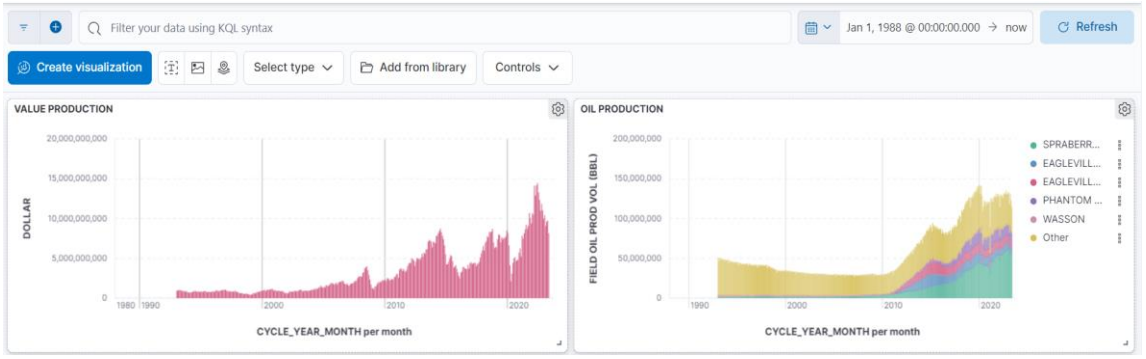
Nota. Visualización de precios de petróleo, valor cuantificable en dólares de producción, geología, distribución de geología y ubicación de pozos petroleros.

En la Figura 14 las barras de la parte izquierda representan el precio en dólares de la producción y en la derecha está la producción de petróleo en BBL, con una categorización de los cinco campos petroleros con mayor producción a través del tiempo.

El diagrama de barras de la parte izquierda representa el precio en dólares de la producción y en la derecha está la producción de petróleo en BBL, con una categorización de los cinco campos petroleros con mayor producción a través del tiempo.

Figura 15

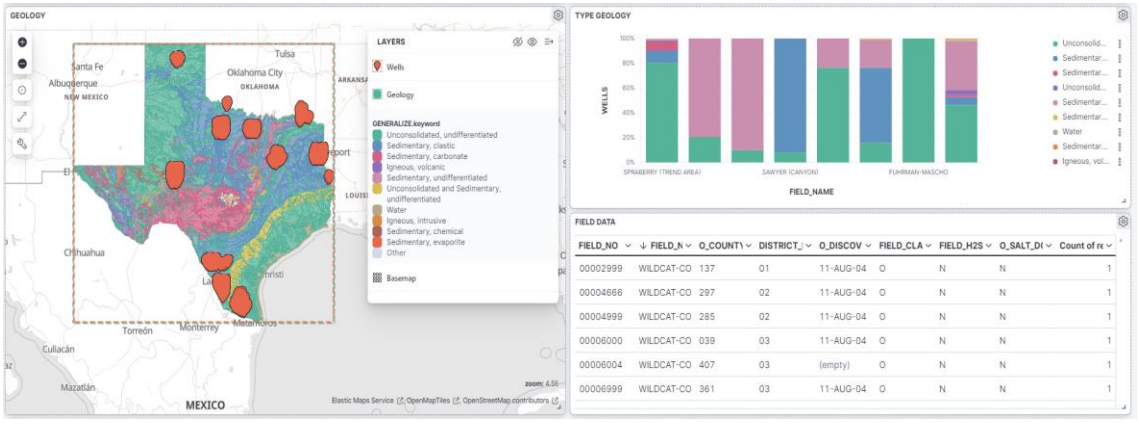
Precios y Producción de Campos Petroleros Texas



En la Figura 15 en el diagrama izquierdo se encuentra la geología con sus diferentes tipos y adicionalmente se encuentra la ubicación de los pozos, en la derecha parte superior está el porcentaje de pozos petroleros con un determinado tipo de geología y en la parte inferior está algunos datos adicionales del campo, como el número del campo, el nombre del campo, el condado y distrito al que pertenece, fecha en que se descubre el primer pozo de petróleo, banderas para saber si el campo cuenta con H2S o tiene un domo de sal.

Figura 16

Geología y Distribución



La representación integral de los diferentes diagramas en el panel general proporciona una visualización condensada y significativa de todo el conjunto de datos. Esta síntesis visual se convierte en un recurso importante para la toma de decisiones estratégicas al simplificar la comprensión de datos complejos.

Conclusiones

Con la exploración de los conceptos clave, se logró la comprensión del concepto de lago de datos, sus implicaciones y su relevancia en la gestión de información. Este conocimiento permitió tener una proyección del modelo de arquitectura deseado.

La investigación de diversas arquitecturas y tecnologías disponibles para la implementación del sistema de procesamiento de datos proporciono una visión detallada de las opciones disponibles. Esta revisión permitió la elección de las herramientas proporcionadas por Elastic, concretamente Elasticsearch y Kibana, como la infraestructura óptima para cumplir los objetivos propuestos.

La implementación del sistema de procesamiento de datos se llevó a cabo de manera favorable. Permitiendo la integración de diversas fuentes de información. Además, el uso del lenguaje Python fue fundamental en el procesamiento de datos y la interacción con Elasticsearch mostrando una gran flexibilidad para manejar flujos de información.

La aplicación del caso de estudio en el área Petrolera del estado de Texas arrojó resultados satisfactorios, permitiendo aprovechar los datos de las diversas fuentes de información. La capacidad de Elasticsearch para indexar grandes volúmenes de datos permitió la correlación de variables de fuentes diversas, como el precio del petróleo y la producción, proporcionando una visión detallada y precisa del valor asociado. Además, Kibana, al facilitar la visualización de estos datos, permitió la identificación de patrones en la distribución de la producción en los distintos campos petroleros y la ubicación geoespacial de pozos.

Trabajo Futuro

Aunque los resultados obtenidos fueron satisfactorios, se identificaron áreas de mejoras, tales como definición de estrategias de ingesta de datos e implementar un sistema para la gestión de metadatos. Además, se percibe la oportunidad para extender a otros sectores industriales. Esto no se limita únicamente a la ingesta de datos en el lago, sino también tiene el potencial para el monitoreo de aplicaciones con arquitectura de microservicios, como la centralización de logs, métricas y eventos permitiendo visualizar la salud y rendimiento de los microservicios. Integración con sistemas de inteligencia artificial para ampliar las capacidades analíticas del sistema, proporcionando conocimientos más profundos de los datos.

Referencias

Hlupić, T., Oreščanin, D., Ružak, D., & Baranović, M. (2022). An Overview of Current Data Lake Architecture Models. IEEE.

Munshi, A. A., & Mohamed, Y. A. R. I. (2018). Data Lake Lambda Architecture for Smart Grids Big Data Analytics. IEEE Access, 6, 40463-40471.

Liu, R., Isah, H., & Zulkernine, F. (2020). A Big Data Lake for Multilevel Streaming Analytics. IEEE

Ramchand, S., & Mahmood, T. (2022). Big Data Architectures for Data Lakes: A Systematic Literature Review. IEEE

Fang, H. (2015, June). Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem. In 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER) (pp. 820-824). IEEE.

Giebler, C., Gröger, C., Hoos, E., Schwarz, H., & Mitschang, B. (2019). Leveraging the Data Lake: Current State and Challenges (pp. 179–188)

Amazon Web Services (AWS) (s.f.). Calculator [en cursiva]. https://calculator.aws/. Accedido el 30 de enero de 2023.

Cuzzocrea, A. (2021). Big Data Lakes: Models, Frameworks, and Techniques. IEEE

Zagan, E., & Danubianu, M. (2021). Cloud Data Lake: The new trend of data storage. IEEE.

Raju, R., Mital, R., & Finkelsztein, D. (2018). Data Lake Architecture for Air Traffic Management. IEEE.

Cravero, A., Lefiguala, I., Tralma, R., & González, S. (2020). Data Lake architecture proposal for the Analysis Directorate of a Regional University. IEEE.

Mehmood, H., Gilman, E., Cortes, M., Kostakos, P., Byrne, A., Valta, K., Tekes, S., & Riekki, J. (2019). Implementing big data lake for heterogeneous data sources. IEEE.

Wrembel, R. (2021). Still Open Problems in Data Warehouse and Data Lake Research. IEEE.

Zhao, Y., Megdiche, I., & Ravat, F. (2021). Data Lake Ingestion Management.

Tunjić, A. (2019). The Automation of the Data Lake Ingestion Process from Various Sources. Opatija Croatia

Rooney, S., Bauer, D., Garces-Erice, L., Urbanetz, P., Froese, F., & Tomić, S. (2019). Experiences with Managing Data Ingestion into a Corporate Datalake. In IEEE. IBM Research, Zurich Laboratory.

Sinthong, P., & Carey, M. J. (2019). AFrame: Extending DataFrames for Large-Scale Modern Data Analysis. En IEEE International Conference on Big Data

Isah, H., & Zulkernine, F. (2018). A Scalable and Robust Framework for Data Stream Ingestion. En International Conference on Big Data (IEEE).

Melchor-Uceda, I. A., Olivares-Rojas, J. C., Gutiérrez-Gnecchi, J. A., García-Ramírez, M. C., Reyes-Archundia, E., & Téllez-Anguiano, A. C. (2021). Data Ingestion System for

Interoperability and Integration of Hospital Data Online and in Real Time. En 2021 Mexican International Conference on Computer Science (ENC). Morelia, Mexico.

Sawadogo, P., & Darmont, J. (2021). On Data Lake Architectures and Metadata Management. Journal of Intelligent Information Systems, 56(1), 97-120. Springer Verlag.

Alrehamy, H., & Walker, C. (2018). SemLinker: Automating Big Data Integration for Casual Users. Journal of Big Data, 5(14).

R. Hai, S. Geisler, and C. Quix. (2016), "Constance: An intelligent data lake system," in Proceedings of the 2016 International Conference on Management of Data.

Yu, H., Cai, H., Liu, Z., Xu, B., & Jiang, L. (2022). An Automated Metadata Generation Method for Data Lake of Industrial WoT Applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(8).

Schoenenwald, A., Kern, S., Viehhauser, J., & Schildgen, J. (2021). Collecting and visualizing data lineage of Spark jobs.

Halevy, A., Korn, F., Noy, N. F., Olston, C., Polyzotis, N., Roy, S., & Whang, S. E. (2016). Managing Google’s data lake: an overview of the GOODS system. IEEE Engineering Bulletin, 39(3), 5.

Sawadogo, P. N., Kibata, T., & Darmont, J. (May 2019). Metadata Management for Textual Documents in Data Lakes. En 21st International Conference on Enterprise Information Systems (ICEIS 2019), Heraklion, Greece (pp. 72-83).

Nogueira, I. D., Romdhane, M., & Darmont, J. (June 2018). Modeling Data Lake Metadata with a Data Vault. Proceedings of the 22nd International Database Engineering & Applications Symposium, 253–261.