UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

DISEÑO E IMPLEMENTACIÓN DE UN DATA LAKE EN LA VICERRECTORÍA DE TECNOLOGÍAS DE LA INFORMACIÓN DE LA UNIVERSIDAD DE CHILE QUE GARANTICE LA CONSISTENCIA DE DATOS EN EL PROCESO DE CALIFICACIÓN ACADÉMICA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

FLAVIO ESTEBAN POBLETE ARTEAGA

PROFESORA GUÍA:
CAROLINA SEGOVIA RIQUELME

MIEMBROS DE LA COMISIÓN:
MARÍA FERNANDA VARGAS COURBIS
LUCÍA MORENO CASTRO

SANTIAGO DE CHILE
2024

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL
POR: FLAVIO ESTEBAN POBLETE ARTEAGA
FECHA: 2024
PROF. GUÍA: CAROLINA SEGOVIA RIQUELME

DISEÑO E IMPLEMENTACIÓN DE UN DATA LAKE EN LA VICERRECTORÍA DE TECNOLOGÍAS DE LA INFORMACIÓN DE LA UNIVERSIDAD DE CHILE QUE GARANTICE LA CONSISTENCIA DE DATOS EN EL PROCESO DE CALIFICACIÓN ACADÉMICA

El presente informe detalla el diseño y la propuesta de implementación de un Data Lake en la Dirección de Datos de la Vicerrectoría de Tecnologías de la Información de la Universidad de Chile, con el objetivo de transformar la gestión de la información académica y garantizar la consistencia de los datos en el proceso de Calificación Académica, específicamente en el Formulario para la Calificación.

El principal objetivo del proyecto es resolver las inconsistencias detectadas entre las bases de datos Oracle y MongoDB, utilizadas actualmente para almacenar y gestionar la información académica, y establecer una base sólida para futuras iniciativas de análisis y toma de decisiones. La solución propuesta no solo aborda estos problemas, sino que sienta las bases para un sistema de datos confiable, escalable y alineado con las mejores prácticas de la industria.

La metodología aplicada se centró en el tratamiento de los archivos JSON almacenados en MongoDB, diseñando una arquitectura de Data Lake que puede ser fácilmente adaptable a otros tipos de datos y fuentes. Se logró completar el diseño de la arquitectura, que incluye la planificación detallada de las fases de ingesta, procesamiento y almacenamiento de datos. Aunque la fase de procesamiento no se completó debido a la falta de herramientas adecuadas, se estableció una estructura robusta y escalable que permite futuras integraciones y mejoras. Por otro lado, la arquitectura propuesta se destaca por su escalabilidad y flexibilidad, lo que permite adaptarse a futuros requerimientos. Esta capacidad de evolución asegura que el Data Lake seguirá siendo útil a lo largo del tiempo, proporcionando un soporte continuo a las necesidades de la universidad.

Por último, aunque la implementación completa del Data Lake es un proyecto a largo plazo, esta solución no solo abordará los desafíos actuales, sino que también posicionará a la Universidad de Chile como una institución líder en la gestión de datos académicos. Al prepararse para afrontar con éxito futuros retos tecnológicos y académicos, el Data Lake potenciará la capacidad institucional para tomar decisiones basadas en datos, contribuyendo así a una mayor eficiencia y efectividad en las operaciones académicas y administrativas.

Agradecimientos

Quiero expresar mi más profundo agradecimiento a mi profesora guía, Carolina Segovia, y a mi profesora Co-Guía M. Fernanda Vargas, por estar siempre ahí para guiar mi camino y resolver todas las dudas que tuve a lo largo de este proceso. A Lucía Moreno, le agradezco completamente por aceptarme para realizar mi práctica, por darme la confianza para llevar a cabo mis ideas y la libertad para expresarme plenamente.

Me siento afortunado de haber tenido una comisión como esta, compuesta por personas que ya son maestras en el campo y en la vida. Estoy profundamente agradecido por haber aprendido de su vasta experiencia, y sé que debo seguir trabajando para seguir logrando cosas, es un camino de nunca acabar.

Agradezco a la Vicerrectoría, en especial a la Dirección de Datos por brindarme el espacio necesario para desarrollar este trabajo. Al principio llegue tímido, pero con el tiempo gané confianza gracias al ambiente de tranquilidad y apoyo que ofrecen. Es un placer trabajar con todos ustedes.

Agradezco también a mis amigos, mis 'panas', les agradezco de corazón por estar siempre ahí, por esos momentos en los que nos reunimos, aunque ahora sea menos frecuente porque cada uno está siguiendo su propio camino. Incluso cuando solo nos comunicamos por chat, siempre logran sacarme una sonrisa.

A mi papá y mamá, les agradezco por su apoyo incondicional, por inculcarme valores y principios que me definen como persona y me han permitido ser el mejor profesional que puedo ser. Les agradezco por darme todo lo que tengo, por todo el esfuerzo que han hecho y por siempre querer lo mejor para mí. Sé que ser padre o madre es una tarea difícil, todos lo son por primera vez y nadie nace sabiendo cómo hacerlo, pero ustedes han hecho un trabajo admirable.

A mi hermana, le agradezco por ser el 'break' en mi vida, por sacarme del mundo en el que a veces me encierro y enseñarme que no todo es trabajo, que también hay que disfrutar los distintos matices que la vida nos ofrece. Sé que tal vez no he sido, ni soy, el mejor hermano mayor, pero hago todo lo posible por serlo y espero estar haciéndolo bien.

Durante todo este proceso, pude reencontrarme conmigo mismo y con lo que realmente me apasiona. Redescubrí esa inclinación que siempre había estado presente dentro de mí, pero que sentí que se perdió en algún momento. Ahora me siento libre, capaz de expresarme y de hacer lo que realmente me gusta. He encontrado eso que llaman "vocación".

Por último agradezco a la vida por darme la oportunidad de vivirla.

Tabla de Contenido

1. Introducción **1**

2. Desarrollo **2**

2.1. Antecedentes Generales . . . 2
2.1.1. Universidad de Chile . . . 2
2.1.2. Calificación Académica . . . 3
2.1.2.1. Instrumentos para la Evaluación . . . 4
2.2. Descripción del Problema u Oportunidad . . . 4
2.3. Descripción y Justificación del Proyecto . . . 6
2.4. Objetivo General . . . 7
2.5. Objetivos Específicos . . . 7
2.6. Alcances . . . 7
2.6.1. Datos . . . 7
2.6.2. Gestión de Datos . . . 7
2.6.3. Metodología . . . 8
2.7. Marco Conceptual . . . 8
2.7.1. Ingeniería de Datos . . . 8
2.7.2. Arquitectura de Datos . . . 8
2.7.2.1. Data Lake . . . 8
2.7.2.2. Data Zones . . . 9
2.7.3. Sistema Fuente . . . 9
2.7.3.1. Tipos de Datos . . . 9
2.7.3.1.1 Datos Estructurados . . . 9
2.7.3.1.2 Datos Semi-Estructurados . . . 9
2.7.3.2. Bases de Datos . . . 9
2.7.3.2.1 Tipos de Bases de Datos . . . 9
2.7.4. Integración de Datos . . . 10
2.7.5. Ingesta de Datos . . . 10
2.7.5.1. Pipelines . . . 10
2.7.5.2. ELT . . . 10
2.7.5.3. n8n . . . 10
2.7.5.4. Frecuencia de Ingesta . . . 10
2.7.5.4.1 Ingesta en Batch: . . . 10
2.7.6. Almacenamiento de Datos . . . 10
2.7.6.1. Almacenamiento de Objetos . . . 10
2.7.6.2. MinIO . . . 11
2.7.7. Sistemas de Datos . . . 11

2.7.7.1. Confiabilidad . . . 11
2.7.7.2. Escalabilidad . . . 11
2.7.7.3. Mantenibilidad . . . 11
2.7.8. ELK Stack . . . 11
2.8. Desarrollo Metodológico . . . 12
2.8.1. Metodología . . . 12
2.8.1.1. Fase 1: Entendimiento del Negocio . . . 12
2.8.1.2. Fase 2: Comprensión de los Datos . . . 13
2.8.1.3. Fase 3: Diseño de Arquitectura del Data Lake . . . 13
2.8.1.4. Fase 4: Ingesta y Almacenamiento . . . 16
2.8.1.5. Fase 5: Procesamiento . . . 16
2.8.1.6. Fase 6: Evaluación . . . 16
2.8.2. Resultados . . . 17
2.8.2.1. Fase 1: Entendimiento del Negocio . . . 17
2.8.2.1.1 Confiabilidad . . . 17
2.8.2.1.2 Escalabilidad . . . 17
2.8.2.1.3 Mantenibilidad . . . 17
2.8.2.2. Fase 2: Comprensión de los Datos . . . 18
2.8.2.3. Fase 3: Diseño de Arquitectura del Data Lake . . . 18
2.8.2.4. Fase 4: Ingesta y Almacenamiento . . . 20
2.8.2.4.1 MinIO . . . 20
2.8.2.4.2 Elasticsearch . . . 21
2.8.2.5. Fase 5: Procesamiento . . . 21
2.8.2.6. Fase 6: Evaluación . . . 22
2.9. Discusiones . . . 23

3. Conclusiones **26**

Bibliografía **27**

Anexos **30**

Índice de Ilustraciones

2.1. Proceso As-Is. 5
2.2. Proceso To-Be. 6
2.3. Ciclo de Vida de la Ingeniería De Datos. 8
2.4. Metodología para el diseño e implementación del Data Lake. 12
2.5. Proceso para escoger la adecuada organización de los datos. 14
.1. Anexo A Metamodelo del Data Lake 30
.2. Anexo B Flujos N8N y MinIO 31
.3. Anexo C Flujos N8N y Elasticsearch 32
.4. Anexo D Código de agrupación de JSON en Python 33
.5. Anexo E Configuración de nodo S3 34
.6. Anexo F JSON ingestados en capa Raw (ingestados) 35
.7. Anexo G Configuración de nodo de Elasticsearch 36
.8. Anexo H Creacion de Zona Raw 37
.9. Anexo I Código Input Logstash 37
.10. Anexo J Código Filter Logstash 38
.11. Anexo K Código Output Logstash 39

Capítulo 1

Introducción

Los datos juegan un papel fundamental en nuestra vida diaria, incluso en situaciones cotidianas, sin que nos demos cuenta. Al enviar un correo electrónico, buscar el restaurante más cercano o comprar ropa por Internet, todas estas actividades dejan rastros de nuestra rutina. Para las organizaciones funciona igual, en ocasiones los datos son algo que simplemente existe, y en otras puede ser un activo muy importante. Los datos almacenados pueden ser datos de clientes, de productos, inclusive datos de datos. Muchos de estos, que a simple vista pueden parecer innecesarios, se preservan y se analizan de diversas formas. Pero, ¿cómo se pueden ingerir, almacenar y analizar de manera efectiva una cantidad tan grande y diversa de datos?.

Nacido con el auge del Big Data hace aproximadamente 14 años, alrededor de 2010, el concepto de Data Lake ganó popularidad entre las organizaciones orientadas a los datos masivos. Aunque el concepto de Big Data ya se estaba desarrollando desde principios de la década de 2000, fue a partir de 2010 cuando cobró una importancia central. El Data Lake se introdujo para resolver los desafíos de los grandes datos, como el crecimiento constante de los datos y la integración de distintas fuentes de datos. Hoy en día, los Data Lakes han evolucionado significativamente, especialmente en términos de Gobernanza de Datos. Temas como la Inteligencia Artificial y Machine Learning, que están en boca de todos, requieren primero una base sólida con una buena gestión. Al igual que una pirámide, es esencial comenzar con una base firme antes de construir hacia arriba.

La Universidad de Chile se encuentra inmersa en un programa de Gobierno de Datos. Como institución educativa, la calidad y el desempeño de sus académicos son de suma importancia. Estos académicos son evaluados mediante un proceso de Calificación Académica, en el cual uno de los instrumentos clave es el Formulario para la Calificación. Sin embargo, se han identificado inconsistencias entre las bases de datos involucradas en este proceso, lo que puede generar diversos problemas. Para abordar estas inconsistencias y mejorar la precisión del proceso, se decidió crear un Data Lake. Esta solución permite integrar y gestionar grandes volúmenes de datos provenientes de diversas fuentes, proporcionando una base sólida para la correcta gestión de datos. Aunque la implementación del Data Lake en esta fase se centra principalmente en mejorar la consistencia de los datos, se espera que en el futuro permita a la Vicerrectoría gestionar y analizar sus datos de manera más eficaz, aprovechando al máximo el potencial de la información disponible.

Capítulo 2

Desarrollo

2.1. Antecedentes Generales

2.1.1. Universidad de Chile

La industria de la educación superior en Chile se encuentra en un periodo de profunda transformación, las nuevas tecnologías, la globalización y las cambiantes necesidades de los estudiantes configuran un escenario que exige a las universidades una constante adaptación a estos tiempos. Las universidades que se adapten a los requerimientos actuales estarán mejor posicionadas para enfrentar los desafíos del futuro y mantener su liderazgo en el mercado.

La Universidad de Chile, como una institución pública con una larga trayectoria y un rol protagónico en el desarrollo del país enfrenta estos desafíos de manera pro activa. Se alza como una institución pública de educación superior desde el año 1842, siendo ésta la primera universidad fundada en el país [1]. Desde sus inicios, ha asumido un rol sumamente importante en la formación de profesionales y ciudadanos, impulsando la docencia con un marcado énfasis en la investigación. Esta ha entrado en un periodo de digitalización sumamente importante, siendo el año 2020 cuando surge la Vicerrectoría de Tecnologías de la Información, referida como 'VTI' en lo que sigue en este informe, cuyo objetivo es digitalizar la Universidad, convirtiéndola en una institución en línea con los avances tecnológicos de los tiempos actuales; fomentando la incorporación de herramientas digitales en sus procesos internos de modo que contribuyan a la eficiencia en la gestión administrativa y propiciando procesos de aprendizaje que aprovechen la riqueza de las plataformas actuales, permitiendo el intercambio de conocimientos y experiencias con el resto del mundo [2].

Para ello se centra en fomentar la incorporación de tecnologías que favorecen el funcionamiento coordinado, transversal e integrado de la Universidad; entregando la infraestructura tecnológica necesaria que apoye los procesos comunicacionales inherentes a la comunidad universitaria; y promoviendo el acceso y la utilización de los diferentes servicios de tecnologías de la información que permitan enriquecer el trabajo diario de los alumnos, académicos y funcionarios [2].

La VTI se compone de las siguientes unidades::

- **Dirección de Datos:** La Dirección de Datos es la encargada de recolectar y disponibilizar los datos que produce y recopila la Universidad generando herramientas que faciliten la información para la toma de decisiones al interior de la Institución, contribuyendo a la construcción de parámetros de alta calidad en los ámbitos de docencia, investigación,

creación artística y extensión, así como para la gestión universitaria [2].

- **Dirección de Tecnología:** La Dirección de Tecnología es la encargada de proporcionar soluciones y servicios de tecnología de información y comunicación a la Universidad manteniendo y operando la infraestructura, diseñando sistemas que sean eficaces y eficientes de conformidad a la misión de la Institución y proponiendo estrategias, políticas, normativas y estándares de funcionamiento [2].

- **Dirección de Innovación:** Su objetivo es ser una unidad pionera en la transformación digital de la educación superior en Chile. Además, acercar a la U. de Chile, mediante tecnologías digitales e inteligencia artificial, a los nuevos escenarios del mundo globalizado [3].

- **Oficina de Educación Online:** Es la encargada del desarrollo de capacidades técnicas, humanas y de nuevas tecnologías, a fin de dar respuesta a las necesidades y particularidades de los organismos universitarios y unidades académicas vinculadas con la docencia [3].

- **Oficina de Seguridad de la Información:** Es la responsable de planificar y desarrollar los sistemas de seguridad que permitan a la comunidad universitaria desenvolverse en un ambiente digital seguro, íntegro y confiable [3].

Este proyecto se realiza en la Dirección de Datos antes descrita dentro del marco del programa de Gobierno de Datos. El Programa de Gobierno de Datos en la Universidad de Chile propicia establecer un conjunto de prácticas y principios que ayudan a administrar y aprovechar los datos de manera efectiva. Consiste en establecer procesos y políticas claras para garantizar que los datos estén disponibles, sean confiables y se utilicen de forma adecuada [4].

2.1.2. Calificación Académica

El proceso de calificación académica en la Universidad de Chile es un sistema diseñado para evaluar el desempeño de sus académicos. Se basa en el compromiso de la Universidad de mantener los más altos estándares en docencia, investigación y servicio. A través de un marco de calificación estructurado, la Universidad garantiza que sus académicos brinden constantemente experiencias excepcionales a sus estudiantes y contribuyan de manera significativa al avance del conocimiento en sus respectivas disciplinas [5].

Este proceso abarca una variedad de criterios, que incluyen la eficacia pedagógica, la productividad en investigación y las contribuciones a la comunidad universitaria. Los profesores son calificados en su capacidad para diseñar y dictar cursos, realizar investigaciones y participar activamente en las actividades de gobierno y servicio de la Universidad [5]. El proceso de calificación se lleva a cabo de manera objetiva y transparente, asegurando que los profesores tengan amplias oportunidades para brindar aportes y recibir retroalimentación sobre su desempeño.

Los resultados del proceso de calificación tienen implicaciones significativas para el desarrollo profesional y el avance de los académicos. Las calificaciones positivas pueden conllevar a ascensos, titularidad y mayores oportunidades de financiamiento para la investigación. Por el contrario, las calificaciones negativas pueden conllevar la realización de planes de acción correctivos y, en casos mas extremos, pueden resultar en el abandono de la Universidad. [5]

El proceso de calificación académica de la Universidad es un pilar fundamental de su compromiso con la excelencia en docencia, investigación y servicio. Al evaluar rigurosamente a su cuerpo docente, la Universidad se asegura de mantener una comunidad académica altamente calificada y dedicada que fomente el crecimiento intelectual y personal.

2.1.2.1. Instrumentos para la Evaluación

El proceso calificatorio se fundamenta en la información asentada en los siguientes instrumentos.

Para uso del académico:

- Programa Anual de Actividades del Académico.
- Informe Anual del Académico (electrónico).
- Formulario para la calificación (electrónico).

De uso institucional:

- Pauta de Calificación.
- Informe del Director de Departamento, Director de Escuela o Jefe de Unidad de Instituto.
- Informe del Decano o Director de Instituto.

El proyecto se encuentra en el Formulario para la Calificación, este es el documento en el cual todos los académicos registran la información relacionada con el cumplimiento de los Programas Anuales, correspondientes al período de calificación [6].

2.2. Descripción del Problema u Oportunidad

Actualmente, existe una inconsistencia de datos entre las bases de datos utilizadas para el proceso de Calificación Académica. El Formulario para la Calificación abarca áreas como docencia, investigación y extensión, entre otras.

Por ejemplo, en el área de docencia, se podrían presentar discrepancias en la información sobre asignaturas impartidas, horas de clase, lugares y las instancias de evaluación, así como en el apoyo en tesis y memorias. Esto puede resultar en evaluaciones inexactas del desempeño docente y afectar la planificación de actividades futuras.

En cuanto a la investigación, las inconsistencias pueden afectar la información sobre proyectos realizados, sus características, autores, montos, lugares e instituciones, así como participaciones en comités. Estas discrepancias no solo afectan la valoración del trabajo de investigación, sino también la obtención de futuros fondos y colaboraciones.

Por el lado de la extensión, actividades extracurriculares como ponencias, cursos, proyectos de extensión, actividades de vinculación con la sociedad, charlas, conferencias, premios recibidos y prestaciones de servicio en mesas y comités pueden verse afectadas. La incorrecta o incompleta información puede desvirtuar el entendimiento del impacto y la efectividad de estas actividades.

Estas inconsistencias dificultan la gestión y el seguimiento del desempeño académico y pueden afectar negativamente la toma de decisiones y la planificación institucional, al basarse en datos incompletos o incorrectos. La integridad y coherencia de la información son fundamentales para garantizar un proceso de calificación preciso y confiable, por lo que es crucial abordar y resolver estos problemas de sincronización y almacenamiento de datos. El proceso actual se aprecia en la siguiente figura:

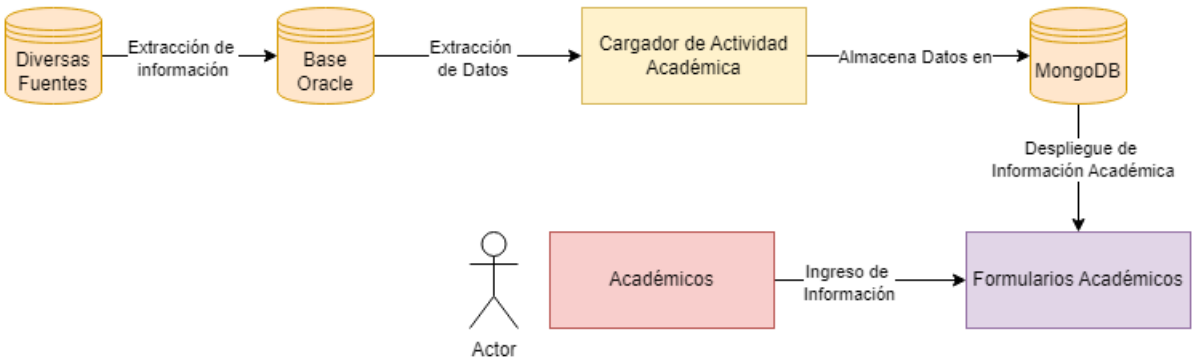


Figura 2.1: Proceso As-Is.

La base de datos principal que almacena estos datos está en Oracle, los datos de esta base se encuentran replicados en MongoDB. El académico rellena el Formulario de Calificación Académica, este formulario se encuentra pre-llenado con los datos que se encuentran en MongoDB. El académico puede agregar información y realizar cambios a la información entregada, estos nuevos datos se almacenan en MongoDB, pero no existe una devuelta de estos datos hacia Oracle, quedando solamente en MongoDB.

2.3. Descripción y Justificación del Proyecto

El proyecto consiste en el diseño de un Data Lake como sistema centralizado de información para los datos de Calificación Académica de la Universidad de Chile. En la figura siguiente se puede apreciar la participacion del Data Lake.

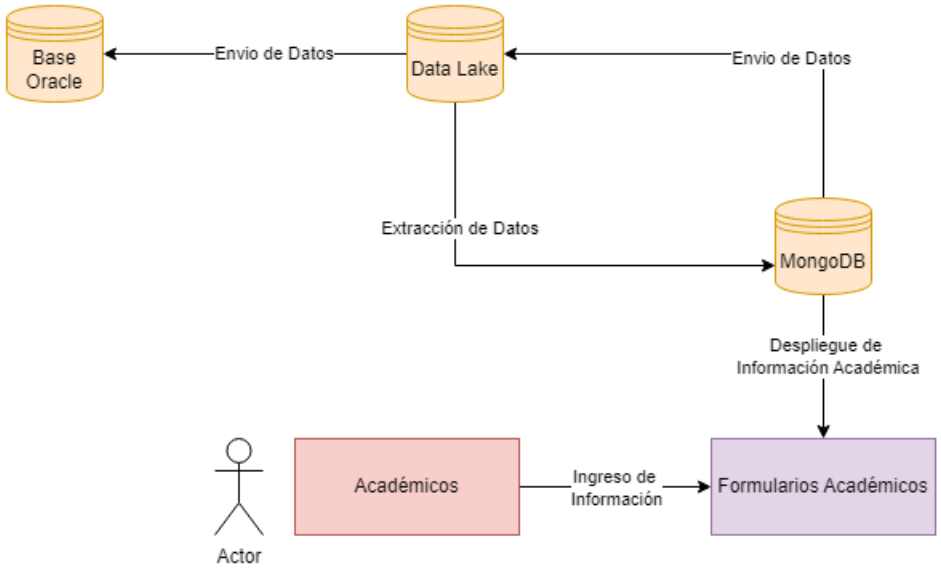


Figura 2.2: Proceso To-Be.

Este proyecto entrega una solución al problema antes planteado dada las siguientes características:

- **Consolidación de datos:** Un Data Lake puede actuar como un repositorio centralizado donde se consolidan todos los datos relevantes, incluidos los provenientes de Oracle y MongoDB. Al tener una única fuente de verdad para los datos, se reduce la posibilidad de inconsistencias causadas por la dispersión de la información en diferentes sistemas. Se pueden aplicar procesos de transformación y limpieza para garantizar la coherencia y la calidad de los datos. Esto puede incluir la estandarización de formatos, la detección y corrección de duplicados, y la normalización de valores. Al hacerlo, se minimizan las discrepancias entre las bases de datos.

- **Implementación de políticas de gobernanza de datos:** Mediante la implementación de políticas de gobernanza de datos en el Data Lake, se pueden establecer reglas y controles para garantizar la coherencia y la integridad de los datos. Esto incluye la definición de roles y permisos, la aplicación de estándares de calidad de datos y la auditoría de cambios. Al hacerlo, se reduce el riesgo de inconsistencias y se mejora la confiabilidad de los datos.

Un Data Lake ofrece una serie de herramientas y funcionalidades que ayudan a abordar la inconsistencia entre las bases de datos Oracle y MongoDB, proporcionando un entorno centralizado, controlado y confiable para la gestión de los datos de Calificación Académica.

2.4. Objetivo General

El objetivo general del proyecto es **'Diseñar e Implementar un Data Lake que funcione como repositorio centralizado de datos, garantizando la correcta consistencia entre los datos de las fuentes de información involucradas en el Formulario de Calificación Académica del proceso de Calificación Académica de la Universidad de Chile'**.

2.5. Objetivos Específicos

Los objetivos específicos del proyecto son los siguientes:

1. Diseñar una Arquitectura para el Data Lake que establezca una base sólida y estructurada que soporte el almacenamiento y procesamiento eficiente para los datos.
2. Lograr la correcta ingesta de los datos utilizados al sistema de almacenamiento, asegurando asi la accesibilidad de los datos desde su origen hasta su almacenamiento.
3. Procesar los datos de manera eficiente, realizando las transformaciones adecuadas, estableciendo las diferentes zonas del Data Lake para lograr una organización óptima y facilitar el análisis y uso de los datos según las necesidades específicas.
4. Elaborar un documento que describa en detalle la Arquitectura del Data Lake, incluyendo todos los procesos involucrados en su funcionamiento, facilitando así su mantenimiento y evolución futura.

2.6. Alcances

A continuación se definen los alcances del proyecto. Se establecen las limitaciones y especificaciones del proyecto para garantizar su viabilidad y éxito dentro del tiempo disponible.

2.6.1. Datos

Si bien el proyecto incluye la integración de datos de diferentes fuentes en el Data Lake, no aborda necesariamente una integración completa y exhaustiva con todos los sistemas existentes dentro de la universidad. Es posible que algunos sistemas específicos requieran integraciones más complejas que pueden ser tratadas como proyectos independientes en el futuro. Se consideran solo datos de Calificación Académica alojados en la base de datos correspondiente en MongoDB. Dado la naturaleza de los documentos en MongoDB es que solo se tratarán los archivos de tipo JSON. Se establece como indicador de éxito el porcentaje de datos del Formulario integrados y procesados exitosamente en el Data Lake, siendo la meta del 100 %.

2.6.2. Gestión de Datos

En términos de gestión de datos, solo se llevó a cabo la gestión de metadata, específicamente mediante la propuesta de un metamodelo. No se desarrollaron pipelines de ingesta de metadata al metamodelo.

No se aplicaron medidas de seguridad ni de privacidad de datos, ya que únicamente el equipo de trabajo tenía acceso a estos. Además, los equipos se conectan mediante Virtual Private Net (VPN) desde dispositivos personales.

2.6.3. Metodología

El proyecto se centra en una problemática específica en lugar de abordar un alcance más general para un Data Lake. Además, solo se realizó el diseño e implementación hasta la fase 4, a partir de la fase 5 se llevó a cabo únicamente el diseño.

2.7. Marco Conceptual

2.7.1. Ingeniería de Datos

La ingeniería de datos es el desarrollo, implementación y mantenimiento de sistemas y procesos que toman datos sin procesar y los transforman en información consistente y de alta calidad que respalda casos de uso posteriores, como análisis y aprendizaje automático [7].

El ingeniero de datos administra el ciclo de vida de la ingeniería de datos, comenzando con la obtención de datos de los sistemas fuente y finalizando con el aprovisionamiento de datos para casos de uso como análisis o aprendizaje automático [7], como se muestra en la siguiente figura.

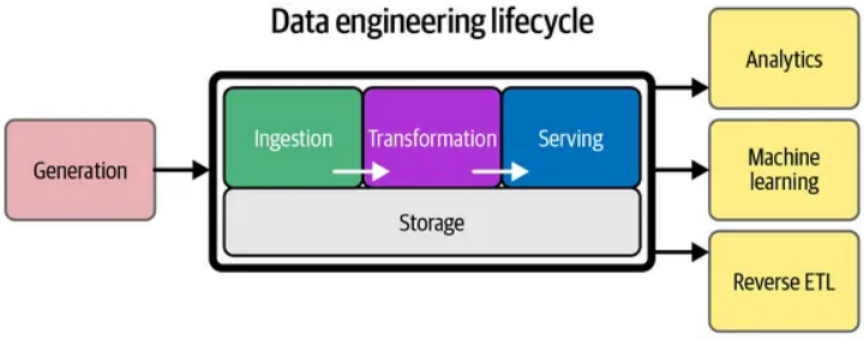


Figura 2.3: Ciclo de Vida de la Ingeniería De Datos.

2.7.2. Arquitectura de Datos

La arquitectura de datos implica diseñar sistemas que puedan adaptarse a las necesidades cambiantes de datos de una empresa. Esto se logra tomando decisiones flexibles y reversibles después de evaluar cuidadosamente las diferentes opciones [7].

Por otro lado, la arquitectura de ingeniería de datos es una parte específica de esta arquitectura más amplia. Se refiere a los sistemas y marcos que forman las etapas esenciales del ciclo de vida de la ingeniería de datos [7].

2.7.2.1. Data Lake

Un Data Lake es un repositorio de almacenamiento que contiene una gran cantidad de datos tanto brutos como procesados y se mantienen allí hasta que sea necesario. A diferencia

de un Data Warehouse jerárquico que almacena datos en ficheros o carpetas, un Data Lake utiliza una arquitectura plana para almacenar los datos. [8]. Sin embargo, es importante tener en cuenta que sin una gestión y gobernanza adecuadas, un Data Lake puede convertirse rápidamente en un 'Data Swamp', lo que dificulta su utilización eficaz y segura [10].

2.7.2.2. Data Zones

La arquitectura por zona de los Data Lakes son un tipo de arquitectura de datos diseñada para administrar datos en diversas etapas de procesamiento. Estas zonas actúan como contenedores para los datos, cada una con características específicas que los datos alojados deben cumplir [9].

2.7.3. Sistema Fuente

Un sistema fuente (source system), en el contexto de la ingeniería de datos y gestión de la información, se refiere a cualquier sistema o archivo que contiene los datos originales utilizados para alimentar otros sistemas o procesos.

2.7.3.1. Tipos de Datos

2.7.3.1.1. Datos Estructurados

Los datos estructurados se almacenan en un formato definido y organizado, generalmente en tablas con filas y columnas. Cada elemento de datos tiene un significado y relación claros dentro del conjunto de datos. Algunos ejemplos son las hojas de cálculo, Excel y CSV.

2.7.3.1.2. Datos Semi-Estructurados

Los datos semi-estructurados no tienen un formato completamente definido, pero siguen una organización lógica. A menudo contienen etiquetas, marcadores o elementos que permiten interpretarlos. Algunos ejemplos son los formatos JSON y YAML [14].

2.7.3.2. Bases de Datos

Las bases de datos son utilizadas para almacenar, mantener y acceder a cualquier tipo de dato. Recolectan información sobre personas, lugares o cosas. Esta información se recopila en un solo lugar para su observación y análisis. En esencia, las bases de datos pueden ser consideradas como una colección organizada de información [15].

2.7.3.2.1. Tipos de Bases de Datos

- **SQL:** Las bases de datos relacionales son un tipo de base de datos que almacena y organiza puntos de datos con relaciones definidas para un acceso rápido. En una base de datos relacional, los datos se organizan en tablas que contienen información sobre cada entidad y representan categorías predefinidas mediante filas y columnas [18]. Las bases de datos relacionales también se crean para comprender lenguaje de consulta estructurado (SQL), un lenguaje de programación estandarizado que se usa para almacenar, manipular y recuperar datos [18]. Algunos ejemplos son MySQL, PostgreSQL, **Oracle**.
- **NoSQL:** Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y almacenan los datos en esquemas flexibles que se escalan con facilidad para aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala [19]. Algunos ejemplos son **MongoDB**, CouchDB.

MongoDB es una base de datos NoSQL de código abierto que almacena datos en colecciones, en lugar de las tablas tradicionales que se utilizan en las bases de datos relacionales. Las colecciones son estructuras flexibles que permiten almacenar una gran variedad de datos sin necesidad de un esquema predefinido [22].

2.7.4. Integración de Datos

Integración de datos se refiere al proceso de combinar y armonizar datos de múltiples fuentes en un formato unificado y coherente que puede ser utilizado para diversos fines analíticos, operativos y de toma de decisiones.

2.7.5. Ingesta de Datos

La ingestión de datos es el proceso de mover datos de un lugar a otro [7].

2.7.5.1. Pipelines

Un pipeline de datos abarca arquitectura, sistemas y procesos que guían los datos a lo largo del ciclo de vida de la ingeniería de datos. Los pipelines de datos se originan en sistemas fuente, siendo la ingestión el inicio del diseño activo del pipeline de datos. Existen diversos patrones de movimiento y procesamiento de datos, incluyendo ETL, ELT, ETL inverso y compartición de datos [7].

2.7.5.2. ELT

Segun sus siglas Extract-Load-Transform, es un patrón en el cual se extraen los datos desde la fuente, se cargan en algún sistema de almacenamiento y posteriormente se transforman. ELT permite que las transformaciones ocurran después de la carga en el sistema objetivo, ademas permite que los datos fuente se instancien en el sistema objetivo como datos crudos, lo que puede ser útil para otros procesos. Esto es común en entornos de Big Data donde ELT carga el Data Lake [16].

2.7.5.3. n8n

N8N es una herramienta de automatización de flujos de trabajo [21], su principal ventaja es que es low-code, no se necesita realizar grandes scripts de códigos, solo seleccionar el conector asociado.

2.7.5.4. Frecuencia de Ingesta

2.7.5.4.1. Ingesta en Batch:

El proceso de recopilar, transformar y cargar grandes cantidades de datos en un sistema de destino a intervalos definidos [7]

2.7.6. Almacenamiento de Datos

2.7.6.1. Almacenamiento de Objetos

El almacenamiento de objetos trata los datos como objetos individuales, cada uno compuesto por los datos en sí, metadata descriptivos y un identificador único [7]. Esta solución altamente escalable y rentable destaca a la hora de manejar grandes volúmenes de datos no estructurados, convirtiéndola en la opción preferida para las necesidades modernas de almacenamiento de datos.

2.7.6.2. MinIO

MinIO es una solución de almacenamiento de objetos de alto rendimiento [24]. Diseñado para ser utilizado tanto en entornos empresariales como en la nube, MinIO ofrece una plataforma robusta y escalable para almacenar grandes volúmenes de datos no estructurados.

2.7.7. Sistemas de Datos

A continuación se presentan las tres preocupaciones mas importantes en la mayoría de los sistemas de datos.

2.7.7.1. Confiabilidad

El sistema debe continuar funcionando correctamente (realizando la función adecuada al nivel de rendimiento deseado) incluso frente a la adversidad (fallos de hardware o software, e incluso errores humanos). Esto implica implementar mecanismos que aseguren la integridad y disponibilidad de los datos, así como estrategias de recuperación ante fallos y redundancia [23].

2.7.7.2. Escalabilidad

A medida que el sistema crece (en volumen de datos, volumen de tráfico o complejidad), debe haber formas razonables de gestionar ese crecimiento. Esto incluye la capacidad de manejar aumentos en la carga sin comprometer el rendimiento, y la posibilidad de expandir la infraestructura de manera eficiente. La escalabilidad asegura que el sistema pueda adaptarse a las demandas crecientes sin una degradación significativa del rendimiento [23].

2.7.7.3. Mantenibilidad

Con el tiempo, muchas personas diferentes trabajarán en el sistema (ingeniería y operaciones, manteniendo el comportamiento actual y adaptando el sistema a nuevos casos de uso), y todos deberían poder trabajar en él de manera productiva. Esto requiere un código limpio y bien documentado, interfaces de usuario intuitivas y una estructura modular que facilite las actualizaciones y el mantenimiento. La creación de una documentación exhaustiva es clave para asegurar que cualquier miembro del equipo pueda realizar tareas de mantenimiento y actualización de manera eficiente [23].

2.7.8. ELK Stack

ELK Stack proporciona una solución completa para la gestión de registros, el análisis de datos y la visualización de información. Es ampliamente utilizado por organizaciones de diversos tamaños para recopilar, analizar y comprender datos de una amplia gama de fuentes, incluyendo registros de servidores, aplicaciones web, dispositivos IoT y redes sociales. Está compuesto por la siguientes tres soluciones:

- **ElasticSearch:** ElasticSearch es un motor de búsqueda y análisis distribuido que permite almacenar, buscar y analizar grandes volúmenes de datos de manera eficiente [20].
- **Logstash:** Logstash es una herramienta de ingesta de datos que recopila, transforma y carga datos de diversas fuentes en ElasticSearch [20].
- **Kibana:** Kibana es una interfaz de usuario web que permite visualizar, analizar e interactuar con los datos almacenados en ElasticSearch [20].

2.8. Desarrollo Metodológico

2.8.1. Metodología

La presente metodología para el diseño e implementación de un Data Lake se inspira en los trabajos realizados por el Instituto de Sistemas Paralelos y Distribuidos de la Universidad de Stuttgart, Alemania. Estos trabajos han sentado las bases para un enfoque sistemático y riguroso en la construcción de Data Lakes exitosos. A partir de estos conocimientos, se propone una metodología integral que abarca desde la comprensión de los requerimientos hasta la implementación, gobernanza y mantenimiento del Data Lake. Esta se aprecia en la siguiente figura:

Figura 2.4: Metodología para el diseño e implementación del Data Lake.

2.8.1.1. Fase 1: Entendimiento del Negocio

Para comprender adecuadamente los requisitos fundamentales de la organización, es primordial obtener una visión integral de la misma, lo cual implica realizar un análisis de su estructura y funcionamiento.

Inicialmente, se debe evaluar el grado de adopción de tecnologías digitales dentro de la organización, incluyendo su predisposición a la innovación y al cambio. Además, es esencial examinar la infraestructura tecnológica existente, incluyendo los diversos sistemas y aplicaciones que generan datos en la actualidad, así como la manera en que estos datos se integran

y comparten entre dichos sistemas, con especial atención a la posible existencia de silos de información. También se debe analizar la capacidad de la infraestructura actual para manejar eficientemente el almacenamiento, procesamiento y análisis de grandes volúmenes de datos.

Posteriormente, es crucial determinar si en la organización prevalece una cultura de apertura y colaboración en lo que respecta a la gestión de datos, evaluando la capacidad de las personas para comprender y utilizar dichos datos, así como la existencia de políticas y procesos establecidos para su correcta administración.

Es importante en esta fase identificar los principales desafíos que enfrenta la organización en la actualidad y oportunidades que ofrece el Data Lake para abordar estos mismos y generar valor para la organización.

2.8.1.2. Fase 2: Comprensión de los Datos

Es necesario identificar los tipos de datos que se prevé almacenar en el Data Lake, así como su volumen, velocidad y variedad [11]. Asimismo, es importante conocer la frecuencia y el método de actualización de estos datos.

En esta fase también es esencial explorar los potenciales usos y aplicaciones de los datos almacenados en el Data Lake. Estos pueden incluir no solo análisis exploratorio y visualización de datos, sino también aplicaciones avanzadas de aprendizaje automático, generación de informes predictivos, y la alimentación de sistemas de toma de decisiones.

La organización no solo descubre la riqueza de sus datos, sino que también sienta las bases para su explotación efectiva. Al comprender a fondo las características y el potencial de los datos, se puede diseñar un Data Lake que responda a las necesidades específicas de la organización y genere valor tangible para el negocio.

2.8.1.3. Fase 3: Diseño de Arquitectura del Data Lake

El diseño de la arquitectura del Data Lake comprende los siguientes pasos fundamentales:

1. **Diseño del Flujo de Datos:** Al determinar el concepto adecuado de flujo de datos, es crucial considerar los requisitos previos, especialmente aquellos relacionados con el tiempo y el uso de los datos. Este aspecto abarca la arquitectura y la interacción para los dos modos de movimiento de datos que pueden tener lugar en un Data Lake: el procesamiento por lotes (batch) y el procesamiento en tiempo real.

2. **Diseño de la Organización de Datos:** La organización de datos define la estructura conceptual dentro del Data Lake, centrándose en la gestión eficiente de la información para diversos propósitos. Por lo tanto, al elegir el concepto apropiado para este aspecto, es crucial considerar cómo se utilizarán los datos. Para determinar el tipo de organización necesario, se sigue el siguiente proceso basado en [10]:

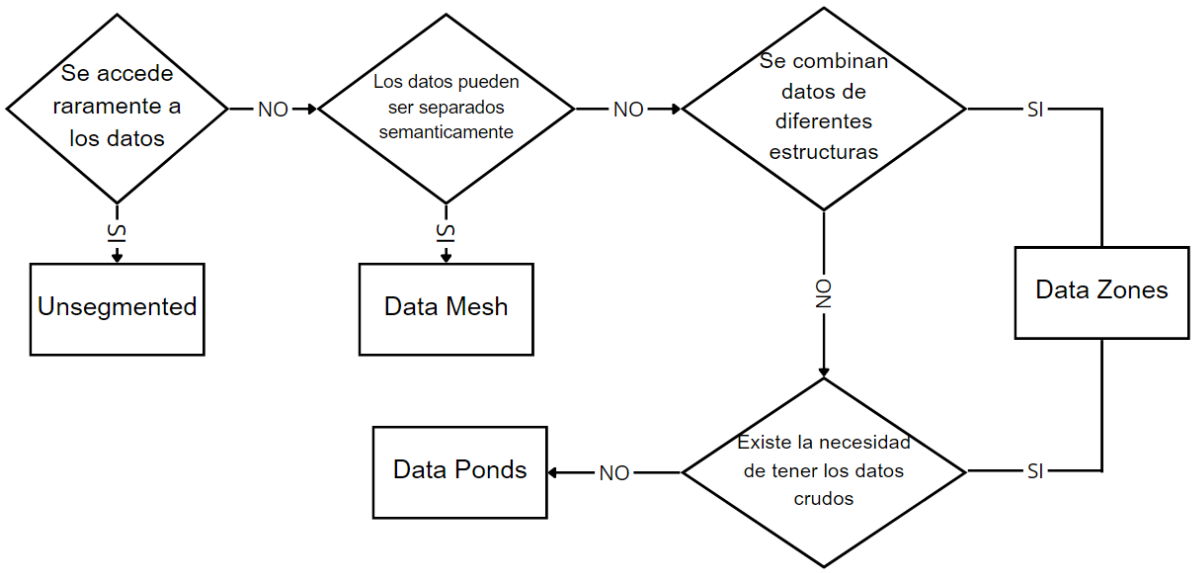


Figura 2.5: Proceso para escoger la adecuada organización de los datos.

El proceso inicia respondiendo la pregunta que se encuentra en el lado izquierdo de la figura en forma de diamante, avanzando gradualmente hasta llegar a la respuesta final.

3. **Diseño del Almacenamiento de Datos:** El almacenamiento de datos se enfoca en los sistemas empleados para guardar y procesar datos. La configuración de un concepto de almacenamiento de datos varía según el tipo de datos a manejar y su uso. Algunas dudas esenciales que se deben abordar incluyen si se requiere la implementación de diversos tipos de sistemas de almacenamiento o qué sistemas de almacenamiento son capaces de satisfacer las características de los datos en cuestión [10].

   Es necesario considerar la seguridad y privacidad de los datos, así como los requisitos de calidad de datos al definir el aspecto de almacenamiento de datos, ya que diferentes tipos de sistemas de almacenamiento de datos ofrecen diferentes niveles de consistencia, restricciones, etc.

4. **Diseño de la Infraestructura:** El aspecto de infraestructura abarca los conceptos para la implementación física del Data Lake. Se enfoca en sistemas y herramientas específicos de almacenamiento, en lugar de la arquitectura en general. Los conceptos establecidos de almacenamiento y flujo de datos se emplean para determinar una infraestructura adecuada para el Data Lake.

   Además de lo señalado anteriormente, es esencial tener en cuenta ciertos factores adicionales que pueden ser determinantes al decidir la infraestructura a utilizar. Estos factores incluyen, por ejemplo, las tasas de ingreso de datos esperadas, la posibilidad de requerir índices o claves foráneas, y los estándares de rendimiento deseados para las operaciones de lectura y escritura [10].

5. **Diseño de Modelo de Datos:** El aspecto de modelado de datos describe si los datos se modelan dentro del data lake y, de ser así, cómo se hace. Por lo general, la técnica

de modelado empleada variará en función de las características y el uso previsto de los datos.

La respuesta a qué datos se gestionan y cómo se utilizan resulta crucial para determinar los enfoques de modelado de datos adecuados, ya que influyen en la elección de los modelos de datos apropiados. También es necesario abordar los requisitos en materia de seguridad y privacidad de datos, así como los de calidad de datos [10].

6. **Diseñar Procesos de Datos:** El aspecto de procesamiento de datos aborda todos los aspectos relacionados con la manipulación de datos dentro del Data Lake. Se enfoca en cómo se lleva a cabo el movimiento de datos dentro del entorno del Data Lake, así como en los procesos de transformación aplicados a dichos datos. Además, se considera el ciclo de vida completo de los datos, desde su ingreso hasta su eventual eliminación o archivo, para garantizar una gestión eficiente y efectiva de los mismos [10].

   Los procesos de datos relacionados con la seguridad y privacidad de los datos, como los procesos para acceder a datos sensibles, y la calidad de los datos deben ser seleccionados de manera significativa para ajustarse a las necesidades del escenario de aplicación.

   No es necesario detallar la estructura de los pipelines en esta etapa, más bien, se busca obtener una visión general de los diferentes procesos que intervienen en el flujo de datos. El diseño de los procesos debe ser eficiente, escalable y capaz de gestionar fallos de manera resiliente.

7. **Gestión de Metadata:** La gestión de metadata abarca todas las actividades que implican gestionar el conocimiento de una organización sobre sus datos. Sin este conocimiento, los datos pueden no ser aplicables para el propósito previsto, por ejemplo, debido a la falta de calidad o confianza [12].

   La metadata juega un papel doble en la arquitectura del Data Lake. En primer lugar, actúa como habilitador, proporcionando el conocimiento y la información necesaria para que otros componentes del sistema funcionen correctamente. Esto tiene múltiples aplicaciones, incluida la facilitación de la búsqueda y exploración de datos dentro del Data Lake, así como la comprensión del significado, contexto y uso de los datos [10].

   Más allá de su papel como habilitador, la metadata también funciona como una función independiente que permite aprovechar al máximo los datos almacenados en el Data Lake, como un catálogo de datos. Dado que la implementación de estas características adicionales requiere un conocimiento detallado de la arquitectura del Data Lake, este paso se realiza generalmente al final. Esta parte del Data Lake puede diseñarse pensando qué beneficios adicionales puede ofrecer la metadata [10], como la mejora en la capacidad de búsqueda y descubrimiento de datos, junto con la facilitación del cumplimiento normativo con el apoyo a la gobernanza de datos.

   Dado que la metadata también es un dato, requiere de las mismas gestiones que un dato común y corriente. Aspectos como el almacenamiento y el modelado son cruciales para garantizar su correcto funcionamiento y accesibilidad.

   En esta fase, la organización no solo comprende el valor de la metadata, sino que también establece las bases para su gestión efectiva. Al definir estrategias claras, se garantiza que

la metadata sea un activo valioso que impulse el descubrimiento, la comprensión y el uso efectivo de los datos en el Data Lake.

2.8.1.4. Fase 4: Ingesta y Almacenamiento

Durante esta fase, se construyen los pipelines de ingesta de datos, y utilizando exclusivamente la tecnología seleccionada para este propósito, sumado al almacenamiento. Estos pipelines son fundamentales para el flujo eficiente de datos desde las diversas fuentes hacia el Data Lake, siguiendo el enfoque ELT.

Además, se integra metadata relevante durante la ingesta de datos, como la procedencia de los mismos, para facilitar su gestión y trazabilidad en el Data Lake. Es esencial garantizar que estos pipelines estén alineados con los objetivos y requisitos del proyecto, y que aprovechen al máximo las capacidades de la tecnología elegida.

2.8.1.5. Fase 5: Procesamiento

En esta fase se realizan los pipelines de procesamiento que transforman y limpian los datos según sea necesario, utilizando exclusivamente la tecnología seleccionada para este propósito. Estos pipelines son la columna vertebral del Data Lake, asegurando que los datos estén listos para su análisis y explotación.

Se utiliza metadata para orientar las transformaciones, el enriquecimiento y el análisis de datos. Integrar metadata en los pipelines nos permite realizar un seguimiento de las transformaciones de los datos a lo largo del tiempo, entender su linaje y evaluar su calidad en cada etapa del proceso. Esto es fundamental para garantizar la integridad y fiabilidad de los datos en todo el procesamiento.

2.8.1.6. Fase 6: Evaluación

Para garantizar el funcionamiento futuro del Data Lake, es crucial realizar una auditoría exhaustiva de todo el proceso. Esto implica una evaluación detallada del rendimiento del Data Lake en términos de velocidad, eficiencia y escalabilidad, identificando áreas de mejora. Se recomienda utilizar herramientas de monitoreo de rendimiento para seguir la utilización de recursos, los tiempos de ejecución y los patrones de los accesos a los datos.

Además, es fundamental implementar prácticas de gestión de metadata para garantizar la calidad, integridad y accesibilidad de metadata. Esto incluye establecer políticas y procedimientos de gobierno de metadata para mantener la consistencia y precisión de esta. En cuanto a la seguridad, se debe monitorear continuamente el Data Lake para detectar y prevenir problemas de seguridad. Se recomienda implementar medidas de seguridad como control de acceso, cifrado y sistemas de detección de ataques.

Para que se mantenga el Data Lake a lo largo del tiempo, es necesario documentarlo, ya sea, su arquitectura, procesos y políticas, y establecer procedimientos de mantenimiento continuo para abordar problemas de calidad de datos, cambios en la infraestructura, etc.

Consideraciones adicionales incluyen establecer políticas y procedimientos de gobierno de datos para garantizar un uso responsable y ético de los datos, proporcionar capacitación y educación a los usuarios sobre el uso del Data Lake, e incorporar conceptos de metadata en los materiales de capacitación.

2.8.2. Resultados

Siguiendo la metodología descrita, a continuación se detallan los resultados obtenidos.

2.8.2.1. Fase 1: Entendimiento del Negocio

Se profundizó en la comprensión del problema a resolver, lo que llevó a una mayor claridad sobre el proceso involucrado y una consideración más completa de las necesidades y desafíos del negocio. Hubo un intento de solución a esta problemática previamente, pero fue solo por un requerimiento puntual, los datos desde MongoDB se cargaron en formato BLOB [17] en una celda de una columna de la tabla principal en Oracle.

Además, se realizó un análisis de los sistemas actuales utilizados tomando en cuenta los tres pilares fundamentales para cualquier sistema de datos según Martin Kleppmann en [23]:

2.8.2.1.1. Confiabilidad

La confiabilidad es esencial para el Data Lake, garantizando que los datos sean precisos y estén disponibles cuando se necesiten. La cantidad de datos es relativamente pequeña y no están distribuidos por lo que la necesidad de implementar estrategias de confiabilidad puede no ser tan crítica en la fase inicial. La gestión de un volumen limitado de datos reduce la probabilidad de fallos. Así, se establece que el sistema cumple con el criterio de confiabilidad en su contexto actual.

2.8.2.1.2. Escalabilidad

Los datos de Calificación Académica suelen tener un volumen no mayor a 1GB. Por lo tanto, el sistema puede adaptarse sin dificultades al procesamiento y almacenamiento necesarios. La Dirección de Tecnología se encarga de gestionar las instalaciones y de satisfacer cualquier requisito adicional que pueda surgir, garantizando así un crecimiento fluido y eficiente. De esta manera, se asegura que el sistema sea escalable.

2.8.2.1.3. Mantenibilidad

La mantenibilidad es fundamental para la operación continua del Data Lake. La organización cuenta con un equipo con sólidas habilidades técnicas y una gestión eficaz de los datos, lo que asegura que la mantenibilidad no sea un problema. Además, la creación de documentación reduce el esfuerzo necesario para mantener el sistema, asegurando que cualquier miembro del equipo pueda realizar tareas dentro del sistema de manera eficiente. De esta manera, se establece que el sistema cumple con el criterio de mantenibilidad.

Se identificó una cierta resistencia a la adopción de nuevas tecnologías a nivel organizacional, lo que también influyó en la selección de herramientas para la arquitectura del Data Lake. Como se detalló anteriormente la instalación de aplicaciones requiere permisos de la Dirección de Tecnología, lo que puede demorar el proceso. Las solicitudes pueden ser rechazadas debido al desconocimiento de la aplicación, lo que obliga a proporcionar documentación adicional o a realizar instalaciones locales. Esto retrasa el tiempo estimado, además, hay periodos en los que la Dirección está atendiendo otros pedidos de otras áreas, lo que puede causar más demoras.

Es importante destacar que, dado el programa de Gobierno de Datos [4], se denota un sólido paradigma en torno a la correcta administración de los datos. La Dirección de Datos entiende al dato como un activo importante, preocupándose de entregar datos de mayor calidad posible. Dado el alcance definido para este proyecto, se ha decidido dar por finalizada

esta fase.

2.8.2.2. Fase 2: Comprensión de los Datos

Se analizó la base de datos MongoDB, la cual contiene 268.316 documentos JSON con un tamaño de almacenamiento de 68.86 MB en su totalidad (en el sistema MongoDB). La frecuencia de acceso mínima a estos es cada 2 años, pero de vez en cuando se puede ingresar a revisar los datos. La base de datos posee un esquema de 13 campos, aunque algunos campos contienen JSON anidados, sumando un total de aproximadamente 150 campos entre todos los esquemas. Se estableció el deseo de poder realizar analítica de negocios, dado que existió el requerimiento de un reporte en algún momento, para una situación en particular, pero no se aplica hoy en día.

Además se establecieron los requerimientos para la arquitectura del Data Lake, estos se detallarán en el apartado siguiente.

2.8.2.3. Fase 3: Diseño de Arquitectura del Data Lake

1. **Flujos de datos:** Se opta por el enfoque batch, considerando que los datos se actualizan en intervalos regulares, no siendo necesario un proceso de ingestión en tiempo real.

2. **Organización:** Siguiendo el procedimiento descrito en la Metodología es que se adopta una arquitectura por zonas. Se puede acceder a los datos constantemente y no pueden ser separados semánticamente, dado que los datos no están definidos estructuralmente, además de combinar datos de diferentes estructuras, por lo que se utiliza una organización por zonas con tres capas:

   - **Zona Raw:** Aquí se almacenan los datos crudos. En esta zona se encuentran los datos tal cual se extraen desde el sistema fuente.
   - **Zona Clean:** Actúa como la única fuente de verdad. En esta zona se realiza la limpieza y validación de los datos.
   - **Zona Service:** Es la última capa donde se gestionan los datos para su uso final. En esta zona se modelan y se preparan los datos para su consumo posterior.

   Esta estructura permite manejar los datos de manera eficiente y organizada, cumpliendo con las necesidades específicas del sistema.

3. **Almacenamiento:** Se eligió utilizar almacenamiento en objetos debido a su alto rendimiento para Big Data, lo cual lo hace ideal para Data Lakes. Como alternativa, se consideraron los recursos existentes seleccionando un Search Index. Esta opción es especialmente adecuada para realizar búsquedas, esta combinación puede acelerar procesos en los cuales se requiera búsqueda de información, como las apelaciones.

4. **Infraestructura:** Para el almacenamiento, se eligió MinIO debido a sus características de almacenamiento de objetos. Además, ya existe una instancia instalada en la Dirección de Tecnologías, lo que agilizó su implementación. Las especificaciones del sistema de MinIO son: 250 GB de almacenamiento, 2 CPU y 8 GB de RAM. También se utilizó el stack ELK. Ya había una instancia de Elasticsearch instalada en la Dirección de Datos, la cual se aprovechó. Las especificaciones de esta instancia son 3 máquinas Ubuntu 20.04 LTS, cada una con CPU de 4 núcleos, 8 GB de RAM, 50 GB de disco para el sistema operativo (SSD), y 250 GB de almacenamiento. Para la ingesta de datos, se

seleccionó la herramienta n8n, que también cuenta con una instancia instalada. Las especificaciones de esta instancia son 8 GB de RAM, CPU de 2 nucleos y 50GB de SSD. Para el procesamiento de datos, se utiliza Logstash, para el cual se realizó el requerimiento correspondiente.

5. **Modelo de datos:** No se estableció un modelado de datos específico dentro del Data Lake. Sin embargo, para la zona Clean, se decidió utilizar una sola tabla como estructura, desjerarquizando los JSON para dejarlos en un formato plano.

6. **Procesos de datos:** No se realizaran procesos de limpieza, ya que la fuente de datos ya se realizan chequeos de calidad y esta bajo políticas de gobernanza.

7. **Gestión de Metadata:** Se desarrolló el siguiente modelo basado en los trabajos HANDLE [12] y el propuesto en [9]. El metamodelo está diseñado para organizar y gestionar metadata en un Data Lake, permitiendo una categorización y análisis eficiente de grandes volúmenes de datos. Este modelo incluye varias tablas principales, cada una con un propósito específico. Se aprecia en la figura .1 del anexo.

   La tabla de **Categorización** (categorización) tiene como objetivo clasificar la metadata en diferentes tipos. Cada tipo tiene un identificador único, un nombre y una descripción. Se establecieron los tipos descritos en [16], los cuales son Descriptivo, Administrativo y Estructural.

   La tabla de **Granularidad** (granularidad) define el nivel de detalle o granularidad de las entidades de datos. Cada nivel de granularidad tiene un identificador único, un nombre y una descripción. Esto permite especificar niveles como "Documento", "Tabla", etc.

   La tabla de la **Entidad de Datos** (dataentidad) representa las entidades de datos almacenadas en el Data Lake. Cada entidad de datos tiene un identificador único, una ubicación de almacenamiento y un nivel de granularidad asociado. Esta relación se establece a través de una clave foránea que conecta con la tabla de granularidad. Un ejemplo de entidad de datos podría ser la colección de MongoDB.

   Para gestionar las diferentes zonas dentro del Data Lake, se utiliza la tabla de **Indicador de Zona** (indicadorzona). Esta tabla define las zonas del Data Lake, cada una con un identificador único, un nombre y una descripción. Las zonas son las tres descritas anteriormente en la organización de los datos.

   La tabla de **Zona de Datos** (datazone) relaciona las entidades de datos con las zonas del Data Lake. Cada entrada en esta tabla incluye el identificador de la entidad de datos, el identificador de la zona, el origen de la importación del dato y un timestamp que indica cuándo se importó el dato. Esto permite rastrear en qué zona del Data Lake se encuentra una entidad de datos específica y cuándo se importó.

   La tabla de **Entidad de Metadata**(metadataentidad) almacena la metadata asociados a las entidades de datos. Cada metadata tiene un identificador único, un contexto que describe su propósito (como üsuario que accede.º "fuente del dato"), un identificador de la entidad de datos asociada y un identificador de la categoría asociada. Estas relaciones se establecen a través de claves foráneas que conectan con las tablas de entidades de

datos y categorización. Un ejemplo de metadata podría describir quién accedió a un dato específico y a qué categoría pertenece este metadata.

Finalmente, la tabla de **Propiedades de Metadata** (propiedadmetadata) representa propiedades específicas de la metadata en forma de pares clave-valor. Cada propiedad de metadata tiene un identificador único, un nombre y una descripción, y está asociada a una entidad de metadata a través de una clave foránea. Esto permite almacenar detalles adicionales de los metadata de manera flexible.

Se desarrolló el metamodelo utilizando un modelo de Entidad-Relación, ya que es el modelo más comúnmente utilizado en la Dirección de Datos. Esto facilitará su implementación y mantenimiento. Este modelo de datos organiza y gestiona metadata en un Data Lake de manera estructurada y eficiente, facilitando la categorización, granularización y contextualización de la metadata. Las relaciones entre las tablas permiten una búsqueda y análisis eficiente, proporcionando una base sólida para la gestión de metadata en un entorno de Data Lake.

2.8.2.4. Fase 4: Ingesta y Almacenamiento

Se llevó a cabo el proceso de extracción y carga a dos sistemas de datos: MinIO y Elasticsearch.

2.8.2.4.1. MinIO

Se crearon tres buckets, uno para cada zona. Se desarrolló el flujo en n8n. Debido a la cantidad de datos y las especificaciones, se seleccionaron los JSON por año mediante una consulta "find". Se configuraron flujos separados para los años menores a 2020 y uno para cada año posterior hasta 2023, con el objetivo de mantener constante el número de JSON, aproximadamente 70,000 por flujo, ya que con las capacidades actuales procesar mas de 100000 items conlleva a bajadas de performance en n8n.

En este punto, se descubrió que había 10 documentos que no se podían extraer con la consulta 'find'. De estos, 9 tenían años mayores a 2023 (aleatorios) y 1 documento no tenía el campo 'anio'(año). Por lo tanto, fue necesario realizar una ingesta utilizando una consulta que abarcara los años mayores a 2023 y aquellos con el campo 'anio' nulo.

Para evitar un exceso de archivos en MinIO, se decidió crear lotes y agruparlos en archivos de 5,000 JSON cada uno, resultando en archivos de aproximadamente 5 MB. El código en Python para esta tarea se muestra en la figura .4. Cada flujo tomó alrededor de 3 minutos, siendo la búsqueda en MongoDB la parte más lenta del proceso.

Los JSON se transformaron a datos binarios, ya que esta es la forma en la que se ingresan los datos a MinIO según el nodo. Los archivos se nombraron según la colección, el año y el número de lote (iteración), tal como se ilustra en la figura .5 del anexo, resultando como se ilustra en la figura .6.

Todo este proceso se automatizó para optimizar la gestión y el almacenamiento de datos en el Data Lake. Sin embargo, aunque el proceso en sí es automatizado, su ejecución no se ha automatizado debido al período entre procesos, se puede comenzar su inicio de manera manual.

2.8.2.4.2. Elasticsearch

Al igual que en el caso anterior se crearon tres índices, uno para cada zona. Para Elasticsearch, el proceso de ingesta de datos fue mas directo en comparación con la otra solución, debido a que Elasticsearch almacena los datos en formato JSON, los archivos pueden ser insertados de manera individual y secuencial. El flujo de trabajo consiste en realizar una búsqueda en MongoDB basada en los años y luego insertar los registros en Elasticsearch uno a uno, como se muestra en la figura .3.

Para la Zona Raw (rawdata) se creó un campo llamado 'MongoDB-itemContenedor' en Elasticsearch, en el cual se ingresa el JSON completo. El nombre del campo incluye la fuente de datos y la colección, lo que permite identificar el origen y rastrear el linaje del dato. Este campo utiliza el tipo 'flattened' en Elasticsearch. Además, se estableció un mapeo dinámico de campos ('dynamic') para manejar cualquier campo adicional que pudiera surgir, asegurando así que el sistema permanezca flexible y capaz de adaptarse a cambios en la estructura de los datos.

El uso del tipo de campo 'flattened' es crucial debido a que Elasticsearch, al detectar numerosos campos, los indexa separadamente, lo que puede llevar a alcanzar el límite de 1000 campos por índice. Este tipo de campo es ideal para manejar documentos JSON anidados o aquellos en los que un mapeo previo no es eficiente [20]. El código utilizado para esta configuración se muestra en la figura .8 del anexo.

Al igual que antes todo este proceso es automatizado y su inicio es manual.

2.8.2.5. Fase 5: Procesamiento

Se diseñó la transformación de un documento JSON con estructura anidada a un formato plano utilizando Logstash. Esta transformación es uniforme para cualquier esquema presente, de las otras secciones de la colección. A continuación, se detalla el proceso utilizado.

1. **Entrada (Input)**: La entrada de datos se configura para leer desde el índice rawdata en Elasticsearch. Aquí se especifica la ubicación del servidor de Elasticsearch, el índice de origen, y las credenciales de acceso. El código se aprecia en la figura .9 del anexo.

    - **hosts:** Dirección del servidor de Elasticsearch.
    - **index:** Nombre del índice de donde se leerán los datos (rawdata).
    - **user y password:** Credenciales para acceder a Elasticsearch.
    - **docinfo:** Incluye información del documento, como _id, en el evento de Logstash.

2. **Filtro (Filter)**:Aqui se transforman los datos de su estructura anidada a un formato plano. Se utilizan varios filtros para desanidar el JSON, extraer campos relevantes, y formatear fechas. El codigo se aprecia en la figura .10 del anexo, es lo mismo para cualquier campo.

    - **json:** Desanida el JSON en el campo especificado.
    - **source:** Campo del cual se extrae el JSON anidado.
    - **mutate:** Extrae y aplanar los campos relevantes del JSON.
    - **add_field:** Añade nuevos campos a la estructura plana.
    - **remove_field:** Elimina campos no deseados del evento.

- **date:** Convierte los campos de fecha al formato ISO8601.
- **match:** Define el formato de fecha de origen.
- **target:** Campo de destino para la fecha formateada.

3. **Salida (Output):** La salida se configura para escribir los datos transformados en el índice cleandata, el cual es el índice de la zona Clean, de Elasticsearch. El codigo se aprecia en la figura .11 del anexo.
   - **hosts:** Dirección del servidor de Elasticsearch.
   - **index:** Nombre del índice de donde se leerán los datos (rawdata).
   - **user y password:** Credenciales para acceder a Elasticsearch.
   - **document_id:** Incluye información del documento, como _id, en el evento de Logstash.

Con esto se establece el formato tabla, solo queda propuesto establecer en formato tabular y enviar los datos modificados a la fuente de destino.

2.8.2.6. Fase 6: Evaluación

Este informe sirve como evidencia documental y está dirigido a un público técnico. Se requiere un conocimiento especializado para entender su contenido. Los destinatarios del documento serán los responsables de su mantenimiento, y deben contar con el dominio técnico necesario para su correcta gestión y actualización.

Para personas sin conocimientos previos, se espera diseñar un documento con la siguiente estructura:

- **Introducción**

  Este apartado contendría una definición simple de un Data Lake, junto con los objetivos y propósito de este en la organización.

- **Visión general de la arquitectura**

  Este apartado contendría un diagrama de alto nivel de los componentes principales del Data Lake junto a una breve descripción de cada uno.

- **Fuentes de datos**

  Este apartado contendría la lista de las fuentes de datos que alimentan el Data Lake y los tipos de datos (estructurados, semi-estructurados, no estructurados).

- **Ingesta de datos**

  Este apartado contendría los métodos y herramientas utilizadas para la ingesta de los datos, junto con la frecuencia de actualización de estos.

- **Almacenamiento**

  Este apartado contendría las tecnologías utilizadas y la organización de los datos (estructura de carpetas, buckets, etc).

- **Procesamiento de datos**

  Este apartado contendría las herramientas y tipo de procesamiento, además de las transformaciones realizadas.

- **Gobierno de datos**

  Este apartado contendría todo lo que tiene que ver con gestión de datos, en este caso seria gestión de metadata.

- **Escalabilidad y rendimiento**

  En este apartado se detallaría la capacidad actual y proyecciones futuras, junto con estrategias de optimización.

- **Glosario de términos**

  Este apartado contendría definiciones simples de términos técnicos utilizados, con tal de facilitar el entendimiento del documento.

- **Preguntas frecuentes (FAQ)**

  Por ultimo se detallarían respuestas a preguntas comunes sobre el Data Lake.

2.9. Discusiones

El Data Lake soluciona directamente la problemática al proporcionar un camino para el retorno de los datos a Oracle. Sin embargo, la implementación aún está pendiente porque es necesario definir el modelo de datos en Oracle y determinar en qué tablas se insertarán los datos. Además, se deben resolver cuestiones relacionadas con las diferencias entre las estructuras de datos, como la adaptación a nuevas columnas y el tratamiento de datos que no se insertan correctamente.

Se logró diseñar una arquitectura sólida para el Data Lake, teniendo en cuenta que se trabajó solo con un tipo de datos, estableciendo una base estructurada para el almacenamiento y procesamiento de datos. Queda pendiente explorar diferentes arquitecturas para manejar datos no estructurados, como documentos, lo que permitirá ampliar las capacidades y la flexibilidad del Data Lake para satisfacer diversas necesidades de almacenamiento y análisis de datos en el futuro.

Se consideraron dos opciones de almacenamiento: Elasticsearch y MinIO.

Elasticsearch es ideal para casos donde se necesita realizar búsquedas rápidas y complejas sobre datos estructurados y semiestructurados, aprovechando su capacidad para indexar y consultar datos basados en JSON. Por otro lado, MinIO proporciona una solución eficaz para almacenar datos estructurados y no estructurados, como archivos multimedia, documentos y backups, utilizando un enfoque de almacenamiento de objetos que ofrece escalabilidad y bajos costos de almacenamiento. Ademas, otros formatos de serializacion no están disponibles en este sistema, por lo que requiere recursos externos.

Cada opción tiene sus propias ventajas según el tipo de datos y el uso previsto. Elasticsearch es más adecuado para aplicaciones que requieren búsquedas y consultas detalladas, mientras que MinIO es preferido para almacenar grandes cantidades de datos de cualquier tipo de manera eficiente y económica. Por lo que la opción de MinIO es la preferida debido a su versatilidad.

n8n es una herramienta adecuada para automatizar flujos simples, pero podría enfrentar desafíos al manejar grandes volúmenes de datos o flujos más complejos en el futuro. Para manejar eficazmente grandes flujos de datos, se requiere una capacidad de escalabilidad horizontal robusta, la cual n8n puede no proporcionar en la medida necesaria comparada con herramientas diseñadas específicamente para este propósito.

Conforme aumenta la complejidad de los flujos de datos, es probable que surjan necesidades adicionales de integración con diferentes sistemas y formatos de datos. Herramientas más especializadas suelen ofrecer una amplia variedad de conectores y adaptadores preconfigurados que facilitan estas integraciones complejas, además de permitir la serialización y transformación de datos durante la ingestión. Además, la comunidad activa de la herramienta puede ofrecer soluciones y mejoras constantes, así como resolver errores. Los foros de internet permiten encontrar soluciones a problemas específicos, ya que es probable que otros usuarios hayan enfrentado problemas similares y compartido sus soluciones.

Es recomendable monitorear constantemente el estado del flujo de trabajo y evaluar la capacidad de escalado de n8n. Si se encuentran dificultades significativas para escalar debido a limitaciones de la aplicación, sería prudente considerar otras opciones más adecuadas para manejar volúmenes y complejidades crecientes de datos.

Aunque los requerimientos se gestionan a través de un sistema de tickets, se identificó que esto no siempre facilita una comunicación clara y fluida. La dependencia exclusiva de los tickets puede llevar a malentendidos, demoras en las respuestas y una falta de contexto que es crucial para la correcta ejecución de tareas. Una alternativa propuesta es establecer reuniones regulares de sincronización o checkpoints donde se puedan discutir los tickets en detalle y resolver dudas. Además, considerar el uso de herramientas de colaboración como Slack o Discord para facilitar la comunicación en tiempo real entre la Dirección de Datos y la Dirección de Tecnología.

Actualmente, el versionamiento de las aplicaciones se realiza de manera manual, lo que no solo es propenso a errores humanos, sino que también dificulta el seguimiento preciso de los cambios y la gestión de versiones. Se propone implementar un sistema de versionamiento automatizado utilizando herramientas de CI/CD (Integración Continua / Entrega Continua) con las herramientas utilizadas en la Dirección de Datos, aunque en este momento no se tiene conocimiento sobre practicas CI/CD se tiene en consideración su importancia.

Una opción viable y flexible para mejorar nuestro sistema es adoptar una estrategia híbrida, combinando almacenamiento y análisis en la nube. El almacenamiento en la nube ofrece una escalabilidad flexible, permitiendo ampliar la capacidad según las necesidades sin necesidad de inversiones iniciales en infraestructura física [25], al pagar solo por el almacenamiento utilizado, se optimizan los costos, especialmente útil en las fases iniciales del proyecto. Además, las plataformas en la nube gestionan las actualizaciones y el mantenimiento de las herramientas analíticas, reduciendo la carga operativa sobre la Dirección de Tecnología.

Esta alternativa es ideal para comenzar con recursos limitados y expandirse conforme crecen las necesidades. Permite aprovechar lo mejor de ambos mundos: la infraestructura local existente y los recursos en la nube, las aplicaciones on-premises generan datos de eventos que pueden enviarse a la nube esencialmente de forma gratuita. La mayor parte de los datos permanece en la nube, donde se analizan, mientras que cantidades menores de datos se envían de vuelta a las instalaciones locales para desplegar modelos en las aplicaciones, realizar ETL inverso, etc [7]. Cabe destacar que en la nube, los mayores costos están asociados con el egreso

de datos al almacenamiento. Para reducir estos costos, es recomendable aplicar técnicas de compresión y partición, lo que disminuiría el tamaño de los datos y, por ende, los costos relacionados con su transferencia.

Implementar alternativas y extensiones al sistema actual podría introducir una mayor complejidad técnica. Esto se debe a que nuevas herramientas y tecnologías pueden requerir configuraciones más avanzadas, integración con sistemas existentes y la adaptación de procesos operativos. A medida que el sistema se vuelve más complejo, es probable que surjan nuevos desafíos que necesitarán una planificación cuidadosa. Esto va de la mano con la capacidad técnica, la capacidad técnica humana actual del equipo podría resultar insuficiente para manejar el volumen de trabajo asociado con la implementación de nuevas herramientas. Si no se amplía el equipo, es probable que se enfrenten dificultades para mantener el nivel de eficiencia y calidad esperado. Aumentar el tamaño del equipo no solo proporcionará más manos para distribuir la carga de trabajo, sino que también aportará una diversidad de habilidades y conocimientos técnicos como no técnicos que pueden ser cruciales para abordar problemas complejos y desarrollar soluciones innovadoras. Invertir en la ampliación del equipo es una estrategia que no está de más, ya que ayudará a garantizar que se cuente con los recursos humanos necesarios para soportar el crecimiento y la evolución del sistema, así como para afrontar cualquier desafío que pueda surgir en el futuro.

Capítulo 3

Conclusiones

En primer lugar, el proyecto ha demostrado que es posible asegurar la consistencia de los datos en entornos complejos y heterogéneos mediante el uso de un Data Lake. La creación de una arquitectura unificada para el almacenamiento de grandes volúmenes de datos permite no sólo solucionar las discrepancias actuales, sino también establecer una base sólida para el manejo eficiente de datos en el futuro.

Se logró la ingesta y el almacenamiento de los datos, un proceso que considero la parte más difícil del proyecto debido a la necesidad de cumplir con formatos específicos. Esta fase requirió una considerable inversión de tiempo, consultas con profesionales expertos en el tema y una extensa búsqueda de información. Finalmente, se pudo llegar a una solución efectiva que permitió avanzar con éxito en el proyecto. Aunque se diseñaron las transformaciones adecuadas y se establecieron zonas en el Data Lake para facilitar el análisis de los datos, la fase de procesamiento de datos no se pudo implementar ya que aún no se dispone del servidor con la herramienta, lo cual llevó a una demora de más de dos meses. Esto limitó la capacidad del proyecto para demostrar la efectividad completa del diseño propuesto.

La alternativa de implementar el Data Lake en la nube surgió como una solución integral, capaz de manejar todos los aspectos operativos detrás de escena. Esta opción habría asegurado el cumplimiento de los objetivos planteados inicialmente. No obstante, el principal punto de incertidumbre sigue siendo el costo asociado a esta alternativa, el cual no ha sido evaluado completamente y representa un factor crítico para la toma de decisiones futuras.

Una lección importante aprendida durante el proyecto es la necesidad de una planificación cuidadosa y una comprensión clara de los requisitos y desafíos específicos de la institución. La colaboración entre diferentes departamentos y la alineación de objetivos son factores cruciales para el éxito de cualquier proyecto. Sin embargo, en este caso, la comunicación no fue tan transversal debido al alcance específico del proyecto. Aun así, se reconoce la importancia de la colaboración interdisciplinaria para futuros proyectos.

Además, aunque el propósito del proyecto se centró en resolver problemas de consistencia de datos, se han identificado múltiples oportunidades para el futuro. En particular, el Data Lake tiene el potencial de integrar diversas fuentes de datos y mejorar la accesibilidad y calidad de la información disponible. Esto abre la puerta a la implementación de herramientas avanzadas de análisis y Machine Learning, que pueden transformar los datos en conocimientos valiosos y apoyar tanto la investigación académica como la toma de decisiones estratégicas. permitiendo a la Universidad de Chile adelantarse en innovación y eficiencia. Para maximizar el impacto y los beneficios del Data Lake en el futuro, se recomienda invertir en capacidades

analíticas avanzadas y en la formación continua del personal.

Como es bien sabido, existen diversas maneras de abordar un problema, y siempre se busca encontrar la mejor solución posible. La solución que se presenta aquí es una de las opciones consideradas. Aunque el objetivo es mejorar y resolver todos los problemas identificados, la aplicación de estas ideas en el mundo real puede ser compleja. La vida está llena de incertidumbres, y a menudo, implementar una solución efectiva no es un proceso simple ni inmediato. Llevar una idea a la práctica implica enfrentar desafíos y adaptarse a circunstancias cambiantes, lo que puede requerir tiempo y paciencia.

En conclusión, el diseño e implementación del Data Lake ha establecido una base sólida para una gestión de datos más coherente y eficiente dentro de la Dirección de Datos creando una infraestructura que permitirá futuras mejoras e innovaciones. La experiencia y el conocimiento adquirido permiten seguir avanzando y mejorando el proyecto, contribuyendo así al avance de la gestión de datos en el ámbito educativo y organizacional.

Bibliografía

[1] Wikipedia. (s. f.). Universidad de Chile. En Wikipedia. Recuperado el 29 de marzo de 2024, de [https://es.wikipedia.org/wiki/Universidad_de_Chile].

[2] Vicerrectoría de Tecnologías de la Información de la Universidad de Chile. (s. f.). Recuperado el 29 de marzo de 2024, de [https://vti.uchile.cl].

[3] Universidad de Chile. (s. f.). Vicerrectoría de Tecnologías de la Información. Recuperado el 29 de marzo de 2024, de [https://uchile.cl/vti].

[4] DataGob UCHILE. (s. f.). Gobierno de Datos de la Universidad de Chile. Recuperado el 29 de marzo de 2024, de [https://datagob.uchile.cl].

[5] Universidad de Chile. (2023). Reglamento General de Calificación Académica. Recuperado el 1 de Mayo de 2024, de [https://uchile.cl/presentacion/normativa-y-reglamentos/reglamento-general-de-calificacion-academica].

[6] Proceso de Calificación Académica. (s. f.). Instrumentos de Calificación Académica. Recuperado el 16 de Mayo de 2024, de [https://uchile.cl/presentacion/normativa-y-reglamentos/titulo-iii-de-los-instrumentos-de-calificacion].

[7] Reis, J., & Housley, M. (2021). *Fundamentals of Data Engineering*. O'Reilly Media.

[8] PowerData. (s. f.). Data Lake: definición, conceptos clave y mejores prácticas. Recuperado el 1 de Mayo de 2024, de [https://www.powerdata.es/data-lake].

[9] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, and B. Mitschang, *A Zone Reference Model for Enterprise-Grade Data Lake Management*, en *Proceedings of the 24th IEEE Enterprise Computing Conference (EDOC 2020)*, 2020. DOI: [https://doi.org/10.1109/EDOC49727.2020.00017].

[10] C. Giebler, C. Gröger, E. Hoos, R. Eichler, H. Schwarz, y B. Mitschang, *The Data Lake Architecture Framework: A Foundation for Building a Comprehensive Data Lake Architecture*, en *Proceedings der 19. Fachtagung für Datenbanksysteme für Business, Technologie und Web (BTW 2021)*, 2021.

[11] Marz, N., & Warren, J. (2015). *Big Data: Principles and Best Practices of Scalable Real-Time Data Systems*. Manning Publications.

[12] Eichler, R., Giebler, C., Gröger, C., Schwarz, H., & Mitschang, B. (2021). Modeling metadata in data lakes—A generic model. *Data & Knowledge Engineering*, Elsevier.

[13] Wikipedia. (s. f.). Unstructured Data. En Wikipedia. Recuperado el 14 de Mayo de 2024, de [https://en.wikipedia.org/wiki/Unstructured_data].

[14] Wikipedia. (s. f.). Semi-Unstructured Data. En Wikipedia. Recuperado el 14 de Mayo de 2024, de [https://en.wikipedia.org/wiki/Semi-structured_data].

[15] TechTarget. (s. f.). Database. En TechTarget. Recuperado el 14 de Mayo de 2024, de [https://www.techtarget.com/searchdatamanagement/definition/database].

[16] DAMA International. (2020). *Data Management Body of Knowledge (DMBOK) 2nd Edition.*. Technics Publications.

[17] Oracle. (s. f.). *Java DB Reference Guide: BLOB Data Type*. Recuperado de [https://docs.oracle.com/javadb/10.8.3.0/ref/rrefblob.html].

[18] Microsoft Azure. (s.f.). *What is a Relational Database?*. Recuperado el 16 de Mayo de [https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-a-relational-database].

[19] Amazon Web Services. (s.f.). *NoSQL*. Recuperado el 16 de Mayo de 2024, de [https://aws.amazon.com/es/nosql/].

[20] Elastic. (s.f.). *Elastic*. Recuperado el 16 de Mayo de 2024, de [https://www.elastic.co/]

[21] n8n. (s. f.). Recuperado el 16 de Mayo de 2024, de [https://n8n.io].

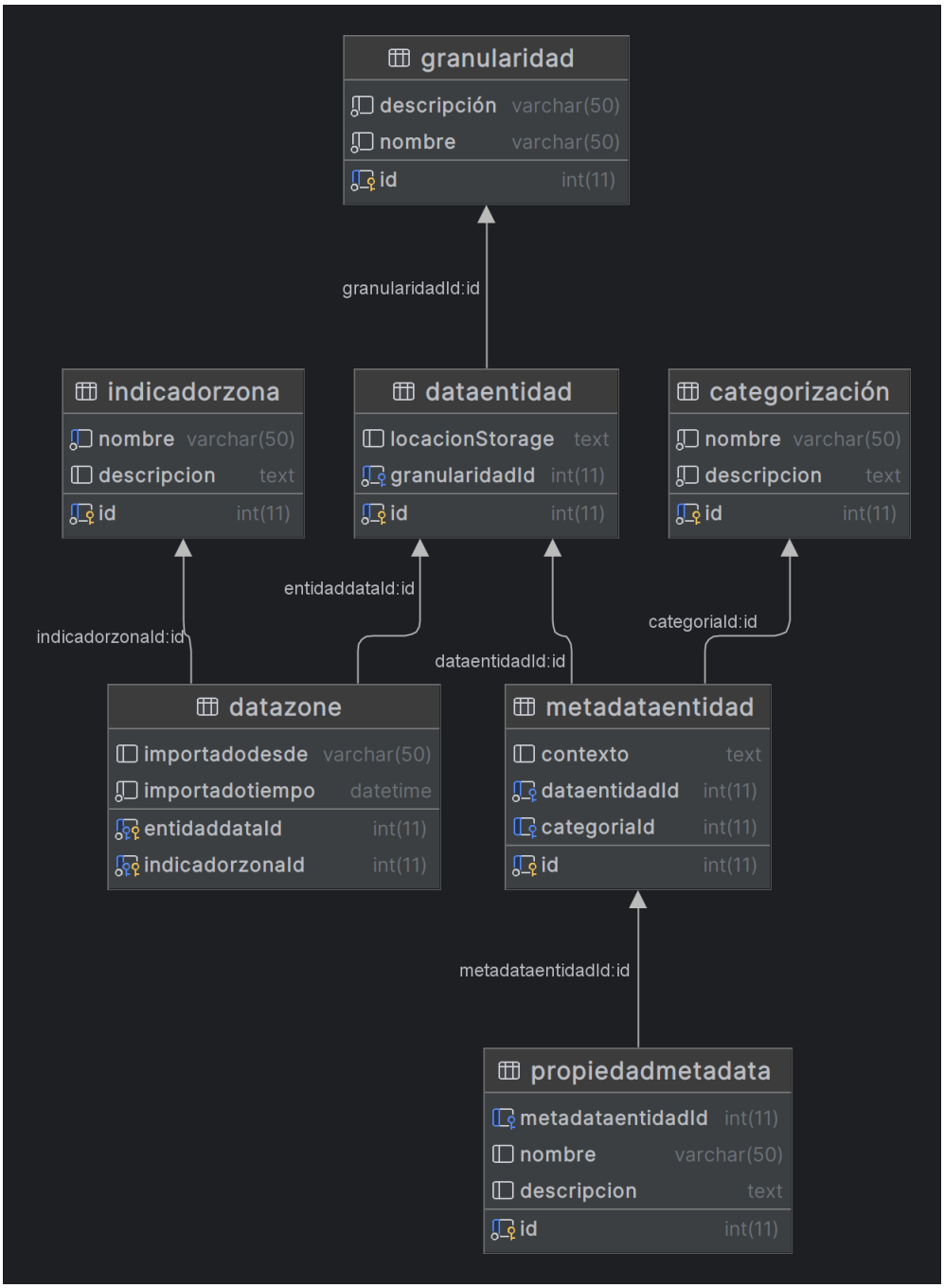
[22] MongoDB. (s. f.). Recuperado el 16 de Mayo de 2024, de [https://www.mongodb.com]

[23] Kleppmann, M. (2017). *Designing Data-Intensive Applications*. O'Reilly Media.

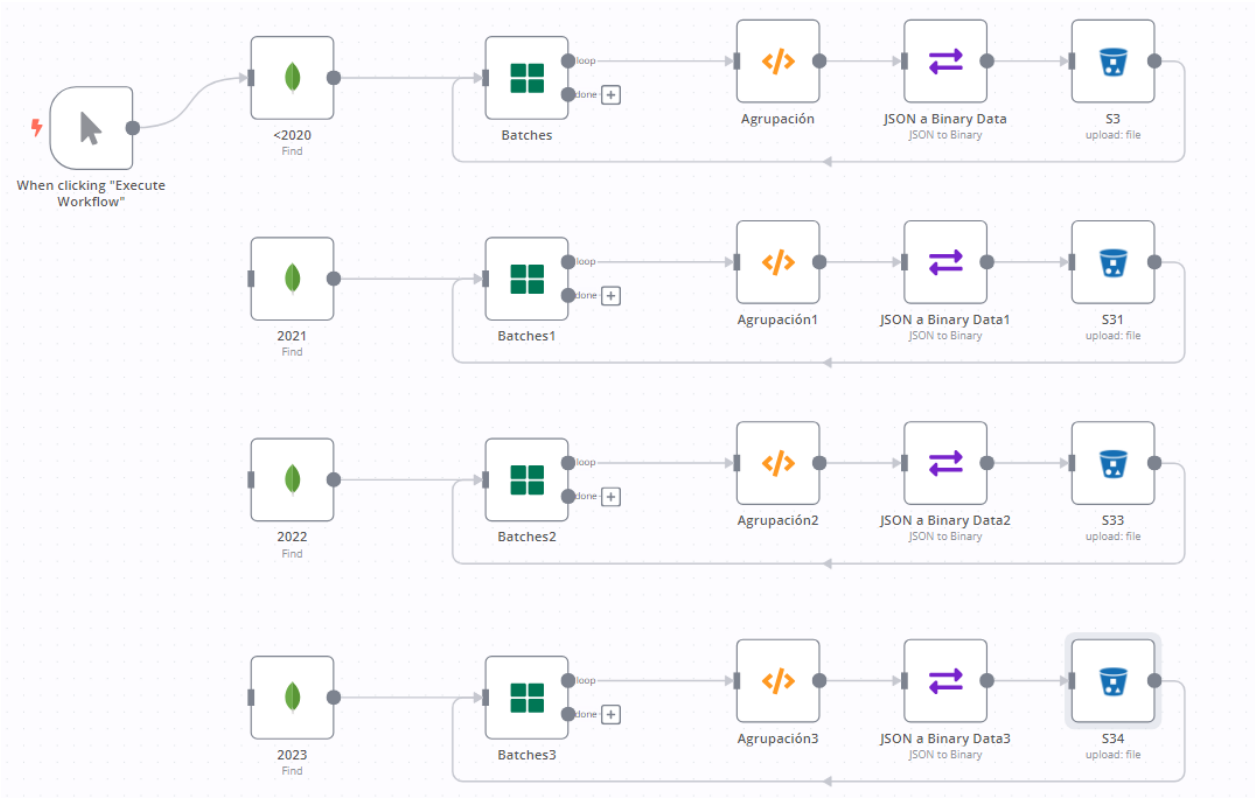
[24] MinIO, Inc. (s. f.). Recuperado el 12 de Junio de 2024, de [https://min.io].

[25] Cloud Computing. (s. f.). Recuperado el 25 de Junio de 2024 de [https://www.ibm.com/es-es/topics/cloud-computing].

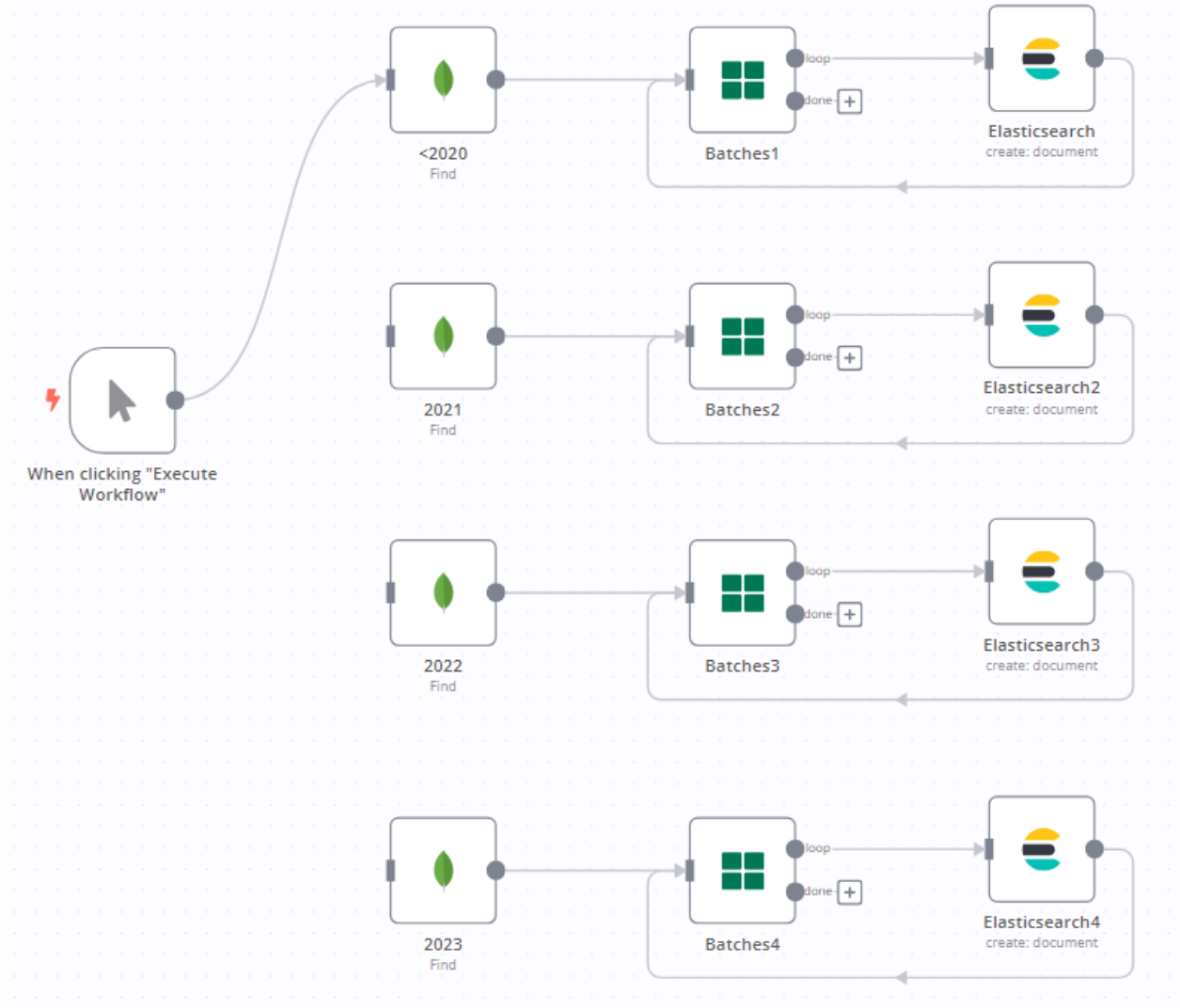
Anexos



Anexo A Metamodelo del Data Lake



Anexo B Flujos N8N y MinIO

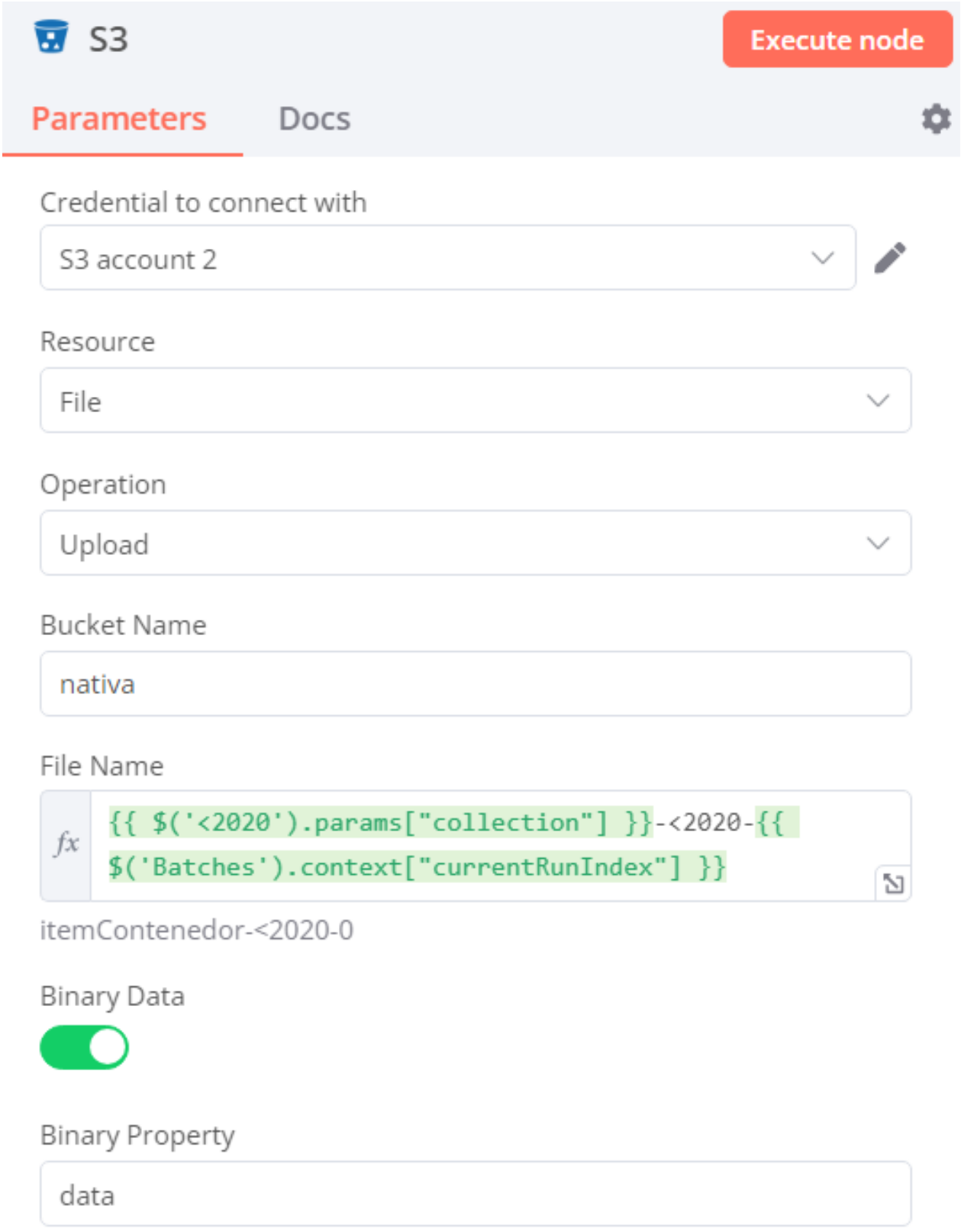


Anexo C Flujos N8N y Elasticsearch

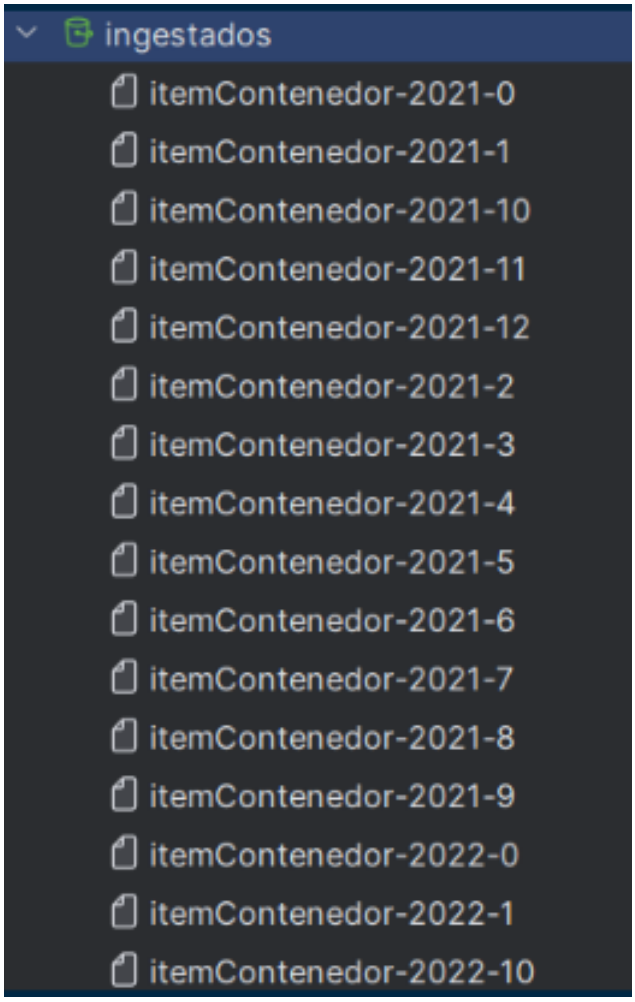
Python

```python
# Crear un diccionario vacío para almacenar los elementos
diccionario = {}

# Crear una lista vacía para almacenar los elementos JSON
lista = []

# Iterar sobre todos los elementos JSON
for item in _input.all():
    # Agregar el elemento a la lista
    lista.append(item)

# Agregar la lista de elementos al diccionario
diccionario["JSON Agrupado"] = lista

# Devolver el diccionario con los elementos JSON agrupados
return diccionario
```

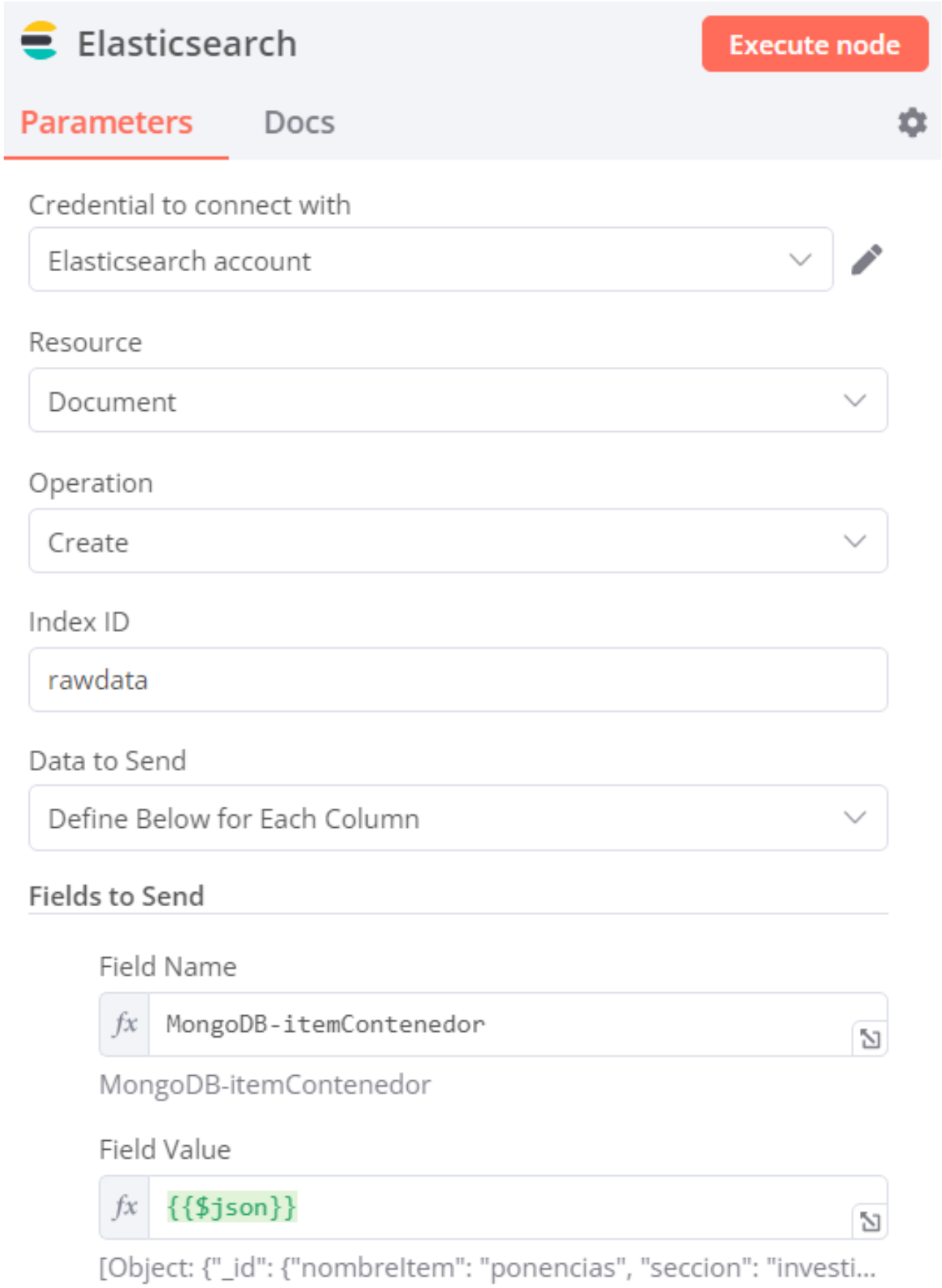
Anexo D Código de agrupación de JSON en Python



Anexo E Configuración de nodo S3



Anexo F JSON ingestados en capa Raw (ingestados)



Anexo G Configuración de nodo de Elasticsearch

```
# Creación Capa Raw
PUT /rawdata
{
  "mappings": {
    "properties": {
      "MongoDB-itemContenedor": {
        "type": "flattened"
      }
    },
    "dynamic": true
  }
}
```

Anexo H Creacion de Zona Raw

```
input {
  elasticsearch {
    hosts => ["Directorio_del_tipo:_http://localhost:9200"]
    index => "rawdata"
    user => "usuario"
    password => "contraseña"
    docinfo => true
  }
}
```

Anexo I Código Input Logstash

```
filter {
  json {
    source => "[_source][MongoDB-itemContenedor]"
  }

  mutate {
    add_field => {
      "nombreItem" => "%{[_source][MongoDB-itemContenedor][_id][nombreItem]}"
      "seccion" => "%{[_source][MongoDB-itemContenedor][_id][seccion]}
       Demas columnas....."
    }
    remove_field => ["_source", "_id", "_score"]
  }

  date {
    match => ["fechaInicio", "ISO8601"]
    target => "fechaInicio"
  }

  date {
    match => ["fechaTermino", "ISO8601"]
    target => "fechaTermino"
  }

  date {
    match => ["fechaCreacion", "ISO8601"]
    target => "fechaCreacion"
  }
}
```

Anexo J Código Filter Logstash

```
output {
  elasticsearch {
    hosts => ["Directorio_del_tipo:http://localhost:9200"]
    index => "cleandata"
    user => "usuario"
    password => "contraseña"
    document_id => "%{itemId}"
  }
  stdout { codec => rubydebug }
}
```

Anexo K Código Output Logstash