

Creación de un almacén de datos para la empresa CITMATEL

**Trabajo de práctica profesional de 4er año de la carrera de ingeniería informática**

**Autor/es:**

Marcos Alejandro Fernández Piñero

maalfepi@gmail.com

**Tutores:**

Dra. C. Lisandra Bravo Ilisástigui

Ibravo@ceis.cujae.edu.cu

La Habana, Cuba

Diciembre, 2024

**Índice**

[**Resumen** 3](#_Toc188005501)

[**Abstract** 4](#_Toc188005502)

[**Introducción** 5](#_Toc188005503)

[**Desarrollo** 9](#_Toc188005504)

[**Capítulo 1: Análisis del marco teórico del problema** 9](#_Toc188005505)

[**1.1 Antecedentes del Estudio** 9](#_Toc188005506)

[**1.2 Análisis de la teoría asociada** 11](#_Toc188005507)

[**1.3 Opciones del mercado** 15](#_Toc188005508)

[**Capítulo 2: Análisis del negocio y Diseño de la solución** 18](#_Toc188005509)

[**2.1 Análisis y modelado del negocio** 18](#_Toc188005510)

[**2.2 Diseño de la solución** 25](#_Toc188005511)

[**Capítulo 3: Diseño y ejecución de pruebas** 41](#_Toc188005512)

[**Parámetros para la evaluación** 41](#_Toc188005513)

[**Diseño de las pruebas** 42](#_Toc188005514)

[**Resultados de las pruebas** 43](#_Toc188005515)

[**Conclusiones** 46](#_Toc188005516)

[**Referencias bibliográficas** 47](#_Toc188005517)

# **Resumen**

En esta tesis se abordará el desarrollo de un sistema integral para la gestión y manejo de los datos generados por la empresa cubana CITMATEL. El trabajo iniciará con un análisis exhaustivo del marco teórico, lo que permitirá comprender en profundidad el problema planteado y explorar posibles soluciones tecnológicas. Seguidamente, se realizará un estudio detallado del negocio para identificar y esclarecer todos los aspectos relevantes que puedan influir en la solución propuesta, asegurando que esta responda de manera efectiva a las necesidades específicas de la empresa.

A partir de los resultados obtenidos en estas etapas iniciales, se procederá al diseño del sistema, incorporando las funcionalidades necesarias para cubrir los requerimientos identificados en el análisis del negocio. Este diseño estará orientado a garantizar una solución robusta, eficiente y alineada con los objetivos estratégicos de CITMATEL.

Finalmente, se implementará un conjunto de pruebas rigurosas con el propósito de evaluar el desempeño del sistema desarrollado. Estas pruebas permitirán analizar su capacidad para resolver el problema inicial y detectar posibles áreas de mejora.

# **Abstract**

This thesis will address the development of a comprehensive system for the management and handling of data generated by the Cuban company CITMATEL. The work will begin with an exhaustive analysis of the theoretical framework, which will allow for a deep understanding of the problem at hand and the exploration of potential technological solutions. Subsequently, a detailed business study will be conducted to identify and clarify all relevant aspects that may influence the proposed solution, ensuring that it effectively meets the specific needs of the company.

Based on the results obtained in these initial stages, the system design will be carried out, incorporating the necessary functionalities to meet the requirements identified in the business analysis. This design will aim to ensure a robust, efficient solution aligned with CITMATEL's strategic objectives.

Finally, a set of rigorous tests will be implemented to evaluate the performance of the developed system. These tests will allow for an analysis of its ability to solve the initial problem and identify potential areas for improvement.

# **Introducción**

Según (Calle García, Alvarado Choez, Anzules Cruz, & Menéndez Parrales, 2024), en la era digital actual, las empresas enfrentan un crecimiento exponencial en el volumen de datos que provienen de diversas fuentes. Este fenómeno es impulsado por la proliferación de dispositivos conectados, plataformas digitales y tecnologías emergentes que generan grandes cantidades de información en tiempo real. Según un estudio, herramientas tecnológicas avanzadas han incrementado la eficiencia y precisión en la recopilación de datos en un 85%, lo que ha permitido a empresas como IBM y FedEx mejorar significativamente sus procesos. Sin embargo, este aumento en la cantidad de datos también presenta desafíos significativos para las organizaciones.

Este fenómeno del flujo constante y variado de información, como refiere (Biroğul & Gültekin, 2016), ha obligado a las empresas a adoptar nuevas estrategias para su gestión y análisis. La diversidad en los formatos y las fuentes de estos datos complica su integración dentro de los sistemas empresariales existentes y para la toma de decisiones asociada al análisis de estos datos. Seguidamente en (Kumar, 2023) se plantea que el manejo inadecuado o la falta de integración eficiente pueden llevar a la creación de silos de datos, donde la información queda aislada dentro de diferentes departamentos o sistemas.

Todo esto desembocado en que la capacidad para tomar decisiones a partir de estos datos se vea afectada. En primer lugar, la calidad y consistencia de los datos son esenciales para obtener resultados precisos. Sin embargo, muchas organizaciones enfrentan problemas con la calidad de los datos debido a su dispersión y falta de estandarización (Robles Angeles, Robles Angeles, Ángeles Hernández, & Acosta Miranda, 2023). Además, las herramientas tradicionales a menudo no están equipadas para manejar el volumen y velocidad actuales del flujo de datos, lo que limita su utilidad en la toma de decisiones estratégicas(Dmitriyev, Mahmoud, & Marín-Ortega, 2022).

Para superar estos desafíos, las empresas están adoptando soluciones que centralizan la gestión y análisis de datos. La inteligencia de negocios (BI) se presenta como una solución efectiva al integrar todos los datos distribuidos dentro de un único almacén centralizado. Esta centralización permite una visión más completa del desempeño empresarial y facilita la toma rápida y precisa de decisiones(Biroğul & Gültekin, 2016). Finalmente, según (Robles Angeles et al., 2023), el uso del proceso ETL (extracción, transformación y carga) es crucial para consolidar los datos provenientes de diversas fuentes en un repositorio centralizado, mejorando así su integridad y accesibilidad.

En el contexto cubano actual, empresas como CITMATEL se han visto inmersas en esta situación. CITMATEL, una empresa cubana con presencia en el ámbito digital, enfrenta al desafiante obstáculo de la dispersión de sus datos en múltiples bases de datos correspondientes a sus diversas tiendas digitales. Esta fragmentación dificulta la realización de análisis complejos necesarios para procesos críticos como el márquetin, las finanzas y las estadísticas, se hace necesario en acceso frecuente y variante a estos datos.

La falta de un sistema centralizado y los mecanismos adecuados para el acceso, gestión y visualización de los datos impide a la empresa aprovechar plenamente su potencial de datos, lo que limita su capacidad para responder ágilmente a las demandas del mercado y optimizar sus operaciones internas. Inicialmente la empresa intento acceder a tecnologías de análisis de datos como:

* Zoho Analitycs, disponible en <https://to.softdoit.com/syjawq2h>
* Qlik, disponible en <https://to.softdoit.com/w5ctu7>
* Tableau, disponible en https://to.softdoit.com/shxo15
* Power BI, disponible en <https://pxl.to/knnpzro>

Durante las pruebas no se pudieron acceder a estas herramientas porque las tres últimas están bloqueadas para cuba y la primera es de pago.

Actualmente, Los departamentos solicitan al equipo de desarrollo los datos a través de correo electrónico, los cuales deben hacer consultas SQL directamente en las bases de datos en producción para obtener la información requerida, exigiendo la inversión de tiempo y esfuerzo en tareas que no deberían ser su responsabilidad y limitando la capacidad de gestión y procesamiento de la empresa.

Con este proyecto se busca diseñar e implementar una solución que permita centralizar los datos en un único repositorio accesible, facilitando así el acceso rápido y seguro a información confiable en tiempo real por personal no relacionado a la informática. Esta centralización no solo mejorará la integridad y seguridad de los datos, sino que también fomentará una mayor colaboración entre los departamentos al proporcionar una única fuente de verdad.

Con esta iniciativa, se espera transformar la manera en que la Entidad gestiona sus datos, permitiendo una toma de decisiones más ágil y fundamentada. El proyecto incluirá tanto el diseño e implementación técnica como la documentación exhaustiva del proceso, asegurando que todos los aspectos del sistema sean comprensibles y replicables en el futuro.

De ahí que se establezca la siguiente **situación problemática**: las consultas a los datos se deben hacer de forma manual directamente a las bases de datos debido a la falta de un repositorio de datos que permita la recopilación y manipulación de los datos, consumiendo demasiado tiempo y esfuerzo de parte del personal de desarrollo de software que es el único capacitado para llevar a cabo esta tarea. Esto limita a los equipos consumidores de estos datos de poder hacer consultas más complejas o especializadas. Adicionalmente las soluciones disponibles en el mercado o son de pago o no están disponibles para Cuba.

Esto plantea una contradicción entre los consumidores de estos datos que requieren un sistema capaz de proporcionárselos de forma clara y directa para su análisis y las estructuras de almacenaje y gestión de los mismos. Planteando la siguiente **pregunta científica**: Cómo se pueden centralizar todas las fuentes de datos en un mismo lugar que permita el acceso a los datos de forma óptima en función de las necesidades de los consumidores y permitirles a estos acceder a los datos de forma clara y entendible para ellos.

Para poder resolver este problema se ha establecido como **objetivo de la investigación**: Implementar un sistema de almacenamiento y gestión de todos los datos de la empresa que satisfaga la demanda de estos datos por parte de los diferentes departamentos de la empresa.

Para la culminación de estas prácticas laborales se acoto la solución hasta la implementación de la Base de Datos centralizada y el sistema de ETL.

Para esto se plantearon como objetivos específicos:

1. Revisión y análisis del estado del arte y la teoría asociada al tema.
2. Comprensión y documentación del negocio.
3. Implementación de la solución al problema identificado
4. Revisión de la factibilidad solución para el problema.

Para el cumplimiento de estos objetivos se proponen las siguientes tareas:

1. Documentación de las bases de datos existentes para la comprensión de las fuentes de datos.
2. Identificación y acercamiento a los *StakeHolders* (interesados) para la obtención de requisitos del sistema
3. Revisión de estudio e investigaciones relacionadas con el tema para identificar conceptos y palabras claves.
4. Profundización de esos conceptos claves y la identificación de tecnologías y estrategias.
5. Identificación de las reglas del negocio y los requisitos del negocio y del sistema.
6. Modelado del negocio y del sistema.
7. Diseño de una solución general que se adapte a las necesidades específicas de la empresa.
8. Desnormalización de las bases de datos existentes y la renormalización de los datos en función de la demanda de los departamentos que requieren los datos.
9. Implementación de la solución propuesta.
10. Implementación de pruebas que validen la efectividad del sistema propuesto.
11. Validación de la capacidad del sistema para adaptase a las necesidades actuales y futuras de la empresa.
12. Despliegue final de la solución.

# **Desarrollo**

## **Capítulo 1: Análisis del marco teórico del problema**

El presente capítulo establece el marco teórico esencial para el desarrollo de un Sistema de Soporte a la Toma de decisiones (DSS en inglés) donde se unifiquen y pueden gestionar todos los datos de la entidad **CITMATEL** que gestiona cuatro tiendas en línea, cada una con una base de datos estructuralmente distinta. Este marco teórico proporciona la base conceptual y analítica necesaria para abordar los desafíos inherentes a la integración de múltiples sistemas de bases de datos.

Inicialmente, se presentarán los antecedentes del estudio, donde se revisarán investigaciones previas y estudios relacionados que han abordado problemáticas similares. Se identificarán las brechas existentes en el conocimiento actual que este proyecto busca llenar, destacando la necesidad de soluciones innovadoras para la integración de datos heterogéneos.

A continuación, se procederá con la definición de conceptos clave, donde se clarificarán los términos y conceptos fundamentales que serán recurrentes a lo largo del proyecto. Estas definiciones estarán respaldadas por fuentes académicas pertinentes, asegurando un entendimiento común y preciso.

Posteriormente, se expondrán las teorías relevantes que sustentan el trabajo. Se explicará cómo estas teorías se aplican específicamente al contexto del proyecto, proporcionando un marco conceptual robusto para guiar el desarrollo e implementación del sistema.

El estado del arte ofrecerá una visión general de los desarrollos actuales en el ámbito de la Inteligencia de Negocios y los Repositorios de Datos. Se incluirán innovaciones recientes y tendencias futuras que podrían influir en el diseño y ejecución del proyecto, asegurando que las soluciones propuestas sean contemporáneas y efectivas.

### **1.1 Antecedentes del Estudio**

Para realizar un análisis del estado del arte se precisa primero realizar un levantamiento exhaustivo de investigaciones previas sobre problemas similares y las soluciones aplicadas, es esencial considerar diversos estudios y enfoques que han abordado cuestiones análogas en el campo de interés. Para ello, se realizó una búsqueda en bases de datos académicas sobre términos como “*Business intelligence*”, “Repositorios de datos”, “Almacenes de datos” y “Sistemas de Apoyo a la toma de Decisiones”. De esta investigación se obtuvo una serie de documentos que serán analizados en esta sección.

En el contexto actual, donde las empresas enfrentan un aumento exponencial en el volumen y variedad de datos, se han identificado varios enfoques y soluciones que abordan problemas similares al que enfrenta CITMATEL.

El uso de Data Warehouses (DW) o almacenes de datos se ha consolidado como una herramienta esencial para integrar y analizar grandes volúmenes de datos provenientes de diversas fuentes para sistemas de soporte a la toma de decisiones (DSS). Estos sistemas permiten transformar datos brutos en información útil y accesible, facilitando así el análisis por parte de los usuarios finales sin necesidad de conocimientos técnicos avanzados(Vite, Villegas, & Pizarro, 2018). La metodología Hefesto, por ejemplo, ha sido ampliamente utilizada para guiar el diseño e implementación de DW, destacándose por su enfoque en la recolección de requerimientos y procesos ETL (Extracción, Transformación y Carga) eficientes (Romero-Chuquital & Melendres-Velasco, 2023).

En investigaciones previas, se ha demostrado que la implementación de un DW puede mejorar significativamente la eficiencia operativa y la capacidad de respuesta empresarial. Por ejemplo, en el ámbito bibliotecario, se ha utilizado un sistema de soporte a las decisiones basado en DW para integrar datos heterogéneos y mejorar la gestión bibliotecaria (Romero-Chuquital & Melendres-Velasco, 2023). Este enfoque ha permitido a los gestores generar informes personalizados y optimizar procesos transaccionales diarios.

Asimismo, y según (Valle, Rocano, Siguenza-Guzman, Cattrysse, & Galarza, 2014), en sistemas de gestión de proyectos, los repositorios de datos han facilitado la investigación al proporcionar una base estructurada para el análisis y la toma de decisiones. Estos repositorios permiten compartir y reutilizar datos, lo cual es crucial para el descubrimiento de nuevo conocimiento dentro de las organizaciones. En particular, se han desarrollado sistemas que utilizan técnicas avanzadas como la minería de datos para asegurar ingresos y evaluar proyectos bajo condiciones inciertas.

Por otro lado, el concepto de Data Lakes(DL) o lago de datos ha sido mencionado como una solución prometedora para manejar grandes volúmenes de datos estructurados y no estructurados por otros estudios como una alternativa más flexible. Este enfoque permite a las empresas almacenar datos en su formato nativo, ofreciendo flexibilidad para el análisis y experimentación por parte de científicos de datos y analistas(Llave, 2018). La DL pueden servir como áreas de preparación para los DW, facilitando la integración de diversas fuentes de datos sin necesidad de transformaciones iniciales. Sin embargo, según (Sulova, 2019) presentan desafíos como la gobernanza de datos y la calidad de los mismos, aspectos que requieren atención cuidadosa durante su implementación.

De esta exhaustiva investigación se definen como conceptos de interés temas como “Data WareHouse”, “Data lake”, “Data Marks” (DM), “DSS”, “Business intelligence”, “ETL”, “Silos de datos” y “Data Hubs”.

### **1.2 Análisis de la teoría asociada**

La inteligencia de negocios (BI, por sus siglas en inglés) se define como un marco integral que abarca un conjunto de herramientas y técnicas destinadas a transformar datos brutos en información valiosa. Esta información es crucial para la toma de decisiones estratégicas dentro de una organización (Zhou, San, & Liu, 2023). A medida que las empresas generan cantidades masivas de datos, la inteligencia de negocios se ha convertido en una infraestructura crítica que permite a las organizaciones reaccionar de manera efectiva a las operaciones tanto operativas como estratégicas de sus clientes, mercados y otros grupos de interés (Arturo & Fernández, 2012).

Este proceso implica la extracción, análisis y presentación de datos mediante sistemas de información más entendibles por los clientes. Estos sistemas permiten a los gerentes obtener un conocimiento profundo sobre el comportamiento histórico del negocio y su entorno, lo que facilita decisiones informadas que mejoran el rendimiento organizacional y optimizan la utilidad económica mediante una reducción significativa de costos (Juárez & Francisco, 2019).

Como ya se mencionó, los sistemas de BI deben cumplir con la característica de permitir a los clientes entender estos datos y tomar decisiones en base a ellos, por lo que surge la necesidad de la creación de sistemas de apoyo a la toma de decisiones(DSS).

Los Sistemas de Apoyo a la Toma de Decisiones (DSS, por sus siglas en inglés) son aplicaciones prácticas de la inteligencia de negocios, diseñadas para asistir a los gestores en la toma de decisiones informadas basadas en datos analíticos. Estos sistemas integran diversas fuentes de información y aplican modelos analíticos avanzados para proporcionar recomendaciones y pronósticos que facilitan el proceso decisional en las organizaciones (Ordoñez Cuthbert & Sambola, 2023).

Los DSS se caracterizan por su capacidad para manejar grandes volúmenes de datos y transformarlos en información útil, permitiendo a los gestores evaluar diferentes escenarios y tomar decisiones estratégicas más acertadas. Al utilizar herramientas de análisis predictivo y modelos de simulación, los DSS ayudan a identificar tendencias, evaluar riesgos y oportunidades, y optimizar los recursos empresariales (Vásquez, Martinez, Aguilera, & Gil, 2018).

Según refiere (Torres & Velastegui, 2020) para logran una mayor eficiencia en el tratamiento de fuentes muy dispersas y heterogéneas, se hace necesario el empleo de herramientas ETL para la correcta recuperación y preparación de estos datos.

Los procesos de Extracción, Transformación y Carga (ETL) son fundamentales en la preparación de datos para su análisis en sistemas de inteligencia de negocios (BI) y sistemas de apoyo a la toma de decisiones (DSS). Estos procesos permiten integrar datos desde múltiples fuentes hacia un formato coherente y utilizable, lo que es esencial para convertir datos brutos en información valiosa.

Según las explicaciones de (Ratnaparkhi, 2019), estos sistemas, como su nombre lo indica, constan de 3 partes: Extracción, Transformación y Carga. La primera implica la recopilación de datos desde diversas fuentes, que pueden incluir bases de datos, archivos planos, servicios web, entre otros. Esta fase es crítica porque asegura que todos los datos relevantes sean capturados para el análisis posterior. Por otro lado, la Transformación es el proceso donde los datos extraídos se limpian y se convierten en un formato adecuado para el análisis. Esto puede incluir tareas como la depuración de datos duplicados, la normalización de formatos y la aplicación de reglas de negocio específicas. La transformación es vital para asegurar que los datos sean precisos y consistentes. Por último, la Carga es la fase final donde los datos transformados se almacenan en un sistema de destino, como una DW o un DM. La eficiencia en esta etapa garantiza que los datos estén disponibles para su análisis en tiempo real o casi real.

Para lograr la realización de un ETL es necesario entender la procedencia de los datos y su estructura en estos lugares. Usualmente, y según refiere la fuente (Hubspot, 2021), estos datos provienen de silos, o sea almacenes de datos que registran datos de un tema o departamento exclusivamente y que otros no necesitan, dejando la información inaccesible para otros grupos. En ocasiones la ocurrencia de silos es un fenómeno que se desea evitar, principalmente cuando los silos requieren ser consultados por varios departamentos.

Como vimos en los sistemas ETL, para poder llevarlo a cabo el proceso de Carga, es fundamental contar con un lugar bien estructurado para el almacenamiento de estos datos. Durante la anterior investigación inicial detectaron 4 “tipos de almacenes”: Data Hubs(DH), DW, DL y DM.

Un **Data Warehouse** es un sistema centralizado diseñado para almacenar grandes volúmenes de datos estructurados provenientes de diversas fuentes. Su principal objetivo es facilitar el análisis y la generación de informes. Este organiza los datos en esquemas predefinidos, como el esquema estrella o copo de nieve, lo que permite realizar consultas complejas de manera eficiente. Este tipo de almacenamiento es ideal para el análisis histórico y la toma de decisiones estratégicas, ya que ofrece una visión consolidada y limpia de los datos (Aissi et al., 2021).

Por su parte un **Data Mart** es una versión más pequeña y especializada de un DW. Está diseñado para atender las necesidades específicas de un departamento o área dentro de una organización, como ventas o finanzas. Los data marts suelen contener subconjuntos de datos relevantes para ese departamento, permitiendo un acceso más rápido y eficiente a la información necesaria para análisis específicos. Esto reduce la complejidad y el tiempo de respuesta en comparación con un data warehouse completo (Jaleel & Abbas, 2020).

Por su parte, Un **Data Lake** es un repositorio de almacenamiento que permite guardar grandes cantidades de datos en su formato nativo, ya sean estructurados, semiestructurados o no estructurados. A diferencia de un DW, los DL no requieren una estructura predefinida para los datos, lo que los hace más flexibles y escalables. Son ideales para el análisis exploratorio y el procesamiento de grandes volúmenes de datos, ya que permiten almacenar datos brutos que pueden ser transformados y analizados posteriormente según las necesidades del negocio (Aissi et al., 2021)

Finalmente, **un Data Hub** se define como una plataforma que facilita la integración de datos en tiempo real para aplicaciones colaborativas. Aunque no existen estándares fijos para su definición, generalmente se caracteriza por su capacidad para manejar datos estructurados, ofreciendo una baja flexibilidad, pero alta accesibilidad. Este tipo de sistema permite a las organizaciones integrar eficientemente información de diversas fuentes, mejorando la colaboración y el intercambio de datos entre diferentes sistemas y departamentos(Amid et al., 2019).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Característica | Data Hub | Data Warehouse | Data Mart | Data Lake |
| Formato de Datos | Estructurado | Estructurado | Estructurado | Estructurado, semiestructurado y no estructurado |
| Escalabilidad | Limitada | Limitada | Limitada | Alta |
| Flexibilidad | Baja | Baja | Media | Alta |
| Objetivo Principal | Integración de datos en tiempo real para aplicaciones colaborativas. | Análisis histórico y generación de informes | Análisis específico por departamento | Almacenamiento y análisis exploratorio de big data. |

Gracias a esta tabla que resume las características de cada uno, y teniendo en cuenta las características del negocio, resulta más conveniente la implementación de un Data Lake sobre los otros sistemas de almacenamiento, pues permite el manejo de datos más heterogéneos y permite mantener estos datos en su forma original sin necesidad de transformarlos.

### **1.3 Opciones del mercado**

Realizando una exhaustiva revisión en internet se identificaron varias tecnologías que son ampliamente empleadas en el mundo del BI como SAS Visual Analytics, Zoho Analytics y Google Data Studio. Adicionalmente, revisando estudios y trabajos publicados también se encontraron las tecnologías Power BI, Tableau, IBM Cognos Analytics, Qlik Sense (Guerrero Garcia & Rodas-Silva, 2022),(Ayala, Ortíz, Guevara, & Maya, 2018).

**Power BI** es una plataforma de análisis de datos desarrollada por Microsoft que permite a los usuarios conectar, modelar y visualizar datos desde diversas fuentes. Ofrece una versión en la nube llamada Power BI Service, que facilita la creación de dashboards interactivos y la colaboración en tiempo real. Su integración con otros productos de Microsoft, como Excel y Azure, lo convierte en una opción atractiva para empresas que ya utilizan el ecosistema de Microsoft. Además, Power BI incluye capacidades avanzadas como el uso de inteligencia artificial para identificar patrones en los datos. Es ampliamente utilizado por su facilidad de uso y su capacidad para manejar grandes volúmenes de información (Microsoft, 2015).

**Tableau** es una herramienta líder en visualización de datos que permite a los usuarios crear gráficos y dashboards interactivos sin necesidad de conocimientos avanzados en programación. Tableau se destaca por su capacidad para manejar grandes conjuntos de datos y su facilidad para integrarse con múltiples fuentes, incluidas bases de datos SQL y servicios en la nube. La versión Tableau Online permite a los usuarios acceder a sus informes desde cualquier lugar con conexión a internet. Tableau es conocido por su enfoque en el diseño visual, lo que permite crear informes altamente personalizables y estéticamente atractivos. Además, ofrece funciones avanzadas como análisis predictivo y soporte para big data (Software, 2019).

**IBM Cognos Analytics** es una solución integral de Inteligencia de Negocio que combina análisis avanzado con capacidades robustas de generación de informes. La plataforma permite a los usuarios crear tableros de mando o dashboards interactivos y realizar análisis predictivos utilizando técnicas como el Aprendizaje Automático (ML por sus siglas en ingles). Cognos Analytics ofrece una versión en la nube que facilita la colaboración entre equipos distribuidos geográficamente. Su arquitectura escalable lo convierte en una opción ideal para grandes organizaciones que requieren un alto grado de personalización y seguridad. Además, se integra fácilmente con otras soluciones empresariales, lo que permite un flujo continuo de datos entre diferentes sistemas (Ibm, 2024).

**Qlik Sense** es una herramienta diseñada para permitir a los usuarios explorar sus datos sin restricciones mediante un motor asociativo único. Este motor permite descubrir relaciones ocultas entre los datos que no son evidentes con las herramientas tradicionales basadas en consultas SQL. Qlik Sense ofrece una versión en la nube llamada Qlik Sense Cloud, que facilita la colaboración entre equipos al permitir el acceso remoto a los tableros de mando. La plataforma también incluye capacidades avanzadas como análisis predictivo e integración con grandes volúmenes de datos. Su enfoque centrado en el usuario final lo hace adecuado tanto para analistas técnicos como para usuarios empresariales (Qlik, 2022).

**SAS Visual Analytics** es una plataforma avanzada que combina visualización de datos con capacidades analíticas robustas, incluyendo análisis predictivo y minería de datos. Esta herramienta está diseñada para manejar grandes volúmenes de información y realizar análisis complejos utilizando técnicas estadísticas avanzadas. SAS Visual Analytics ofrece opciones tanto locales como basadas en la nube, permitiendo a las empresas elegir la infraestructura que mejor se adapte a sus necesidades. La plataforma también incluye herramientas colaborativas que facilitan el intercambio de datos e información valiosa entre equipos multidisciplinarios. Es ampliamente utilizada en sectores como finanzas, salud y manufactura debido a su capacidad para procesar datos complejos (Statistical Analysis, 2023).

**Zoho Analytics** es una solución asequible y fácil de usar diseñada principalmente para pequeñas y medianas empresas (PYMEs). Ofrece tanto versiones locales como basadas en la nube, permitiendo a las empresas elegir cómo desean gestionar sus datos. Zoho Analytics destaca por su interfaz intuitiva que facilita la creación rápida de informes interactivos sin necesidad de conocimientos técnicos avanzados. La plataforma también incluye capacidades básicas de análisis predictivo y soporte para grandes volúmenes de datos, lo que permite a las PYMEs obtener datos e información valiosa sin incurrir en altos costos (Zoho, 2019).

**Google Data Studio** es una herramienta gratuita basada completamente en la web que permite crear informes interactivos conectando diversas fuentes de datos, como Google Analytics, Google Sheets o bases de datos SQL. Data Studio se destaca por su simplicidad y facilidad para compartir tableros de mando interactivos con otros usuarios mediante enlaces públicos o privados. Aunque no ofrece algunas funcionalidades avanzadas presentes en otras plataformas comerciales, su integración nativa con productos Google lo convierte en una opción atractiva para empresas pequeñas o equipos internos que ya utilizan el ecosistema Google Workspace (Nedelcu, 2024).

Estas herramientas al ser probadas presentaron en general 3 problemas críticos. El primero es que eran software propietario, lo cual limita la capacidad de escalar en funcionalidades especificas del cliente, además son de pago o tienen versiones gratuitas muy limitadas y algunas estaban bloqueadas para Cuba. Esto llevo a la necesidad del desarrollo de un nuevo sistema capaz de adaptarse a las necesidades actuales y futuras que sea propio de la entidad y de pueda escalar y personalizar.

## **Capítulo 2: Análisis del negocio y Diseño de la solución**

### **2.1 Análisis y modelado del negocio**

En el proceso de captura y modelado de requerimientos para un sistema de ingeniería de software, se distinguen varias fases indispensables. La primera es la captura de requisitos y reglas del negocio, donde se recopilan las necesidades funcionales y no funcionales del sistema, así como las reglas que rigen los procesos del negocio. Este proceso implica identificar las expectativas y restricciones que se tienen sobre el sistema, asegurando que los requisitos sean claros, completos y verificables. Además, se documentan las reglas del negocio, que son fundamentales para garantizar que el sistema se alinee con los objetivos estratégicos de la organización. Estas reglas definen cómo se deben ejecutar los procesos dentro del sistema y establecen las condiciones bajo las cuales se toman decisiones automatizadas o manuales (Sommerville, 2011) .

Posteriormente, se procede a la fase de modelado del negocio y modelado del sistema, donde se construyen representaciones abstractas tanto del entorno empresarial como del sistema en desarrollo. El modelado del negocio permite visualizar y analizar los procesos actuales, identificando oportunidades de mejora y asegurando que el diseño del sistema refleje fielmente la realidad operativa. En paralelo, el modelado del sistema traduce los requisitos capturados en una representación técnica mediante diagramas permitiendo a los desarrolladores entender cómo deben implementarse las funcionalidades dentro del software (Sommerville, 2011).

#### **2.1.1 Requisitos y reglas del negocio**

Para una correcta captura de los requisitos lo primero es la correcta identificación de los interesados. Los interesados son todas aquellas personas o entidades que tienen un interés directo o indirecto en el desarrollo del sistema, ya sea porque lo utilizarán, lo afectarán o se verán impactados por su implementación. En este caso, se identificaron cuatro grupos principales de interesados:

* Elena (Líder del Equipo de Desarrollo)
* Equipo de Desarrollo de software
* Equipo de Marketing y Publicidad
* Osvaldo (Analista de datos)

Luego es una correcta identificación y aplicación de técnicas de captura de la información. Estas son métodos estructurados que permiten recolectar datos relevantes sobre los requisitos y las reglas del negocio de un sistema. Estas técnicas facilitan la interacción entre los analistas y los interesados, asegurando que se obtenga una visión clara y precisa de las necesidades del proyecto. En esta investigación, se emplearon tres técnicas principales: entrevistas, sesiones de trabajo conjunto y cuestionarios. Las entrevistas consistieron en reuniones individuales con los interesados clave, donde se profundizó en sus expectativas y requerimientos específicos. Las sesiones de trabajo conjunto permitieron a varios actores involucrados colaborar simultáneamente para identificar problemas y soluciones en tiempo real, fomentando el consenso sobre los requisitos. Finalmente, los cuestionarios se utilizaron para recopilar información estructurada de un grupo más amplio de participantes, permitiendo obtener datos cuantitativos y cualitativos que complementaron las otras técnicas (Roger & Bruce, 2015).

De estas técnicas se obtuvieron las siguientes reglas de negocio:

* + El sistema maneja información de pedidos, comprador, receptor, pagos, transportistas y proveedores.
  + Los departamentos de Marketing, Ventas y Estadísticas requieren para sus operaciones el consumo de datos alojados en las bases de datos.
  + Los datos requeridos por estos departamentos son heterogéneos, lo que implica que la combinación, tiempo o rango de los datos en un pedido no se repite con frecuencia.
  + El acceso a los datos debe estar restringido al personal autorizado

Una vez identificadas las reglas de negocios se procede a la identificación de los requisitos funcionales y no funcionales de la misma.

**Requerimientos funcionales**

* El sistema debe permitir la recuperación de datos procedentes de las bases de datos de las plataformas de ventas.
* El sistema debe permitir la actualización diaria de los datos.
* Si un dato que ya se encontraba en el repositorio fue modificado en la fuente, el sistema debe ser capaz de reflejar esa actualización
* El sistema debe ser capaz de eliminar información que sobrepaso los 5 años de antigüedad
* Si un dato que ya fue eliminado del repositorio es solicitado, el sistema debe ser capaz de recuperarlo
* El sistema debe ser capaz de obtener toda la información referente a: pedidos, comprador, receptor, pagos, transportistas y proveedores.
* El sistema debe implementar lógica de autentificación para evitar el acceso a los datos de personal no autorizado.
* El sistema debe implementar mecanismos de seguridad que impidan el acceso externo a la empresa a los datos.
* El sistema debe permitir al usuario la creación de consultas dinámicas de los datos.
* El sistema debe permitir la visualización de los datos en formatos entendibles para los consumidores.
* El sistema debe permitir la creación de diferentes tipos de gráficos y tablas.
* El sistema debe ser capaz de realizar resúmenes estadísticos sobre los datos.
* El sistema debe de permitir la creación de reportes de los datos obtenidos en las consultas.
* El sistema debe permitir la exportación de estos reportes a formatos entendibles por herramientas de hojas de cálculos
* El sistema debe comunicarse en solo lectura con las bases de datos de las plataformas.

En cuanto a los **requerimientos no funcionales** solamente se solicitó el empleo de tecnología de código abierto y que se realizara una aplicación web para dar solución al problema.

#### **2.1.2 Casos de uso y actores**

Luego de haber identificado las requisitos y reglas del negocio podemos empezar a analizar cómo debería comportarse el negocio. Según la fuente (Sommerville, 2011) una forma de lograr este objetivo es a través de la identificación de actores y casos de uso y la creación de un diagrama de casos de uso.

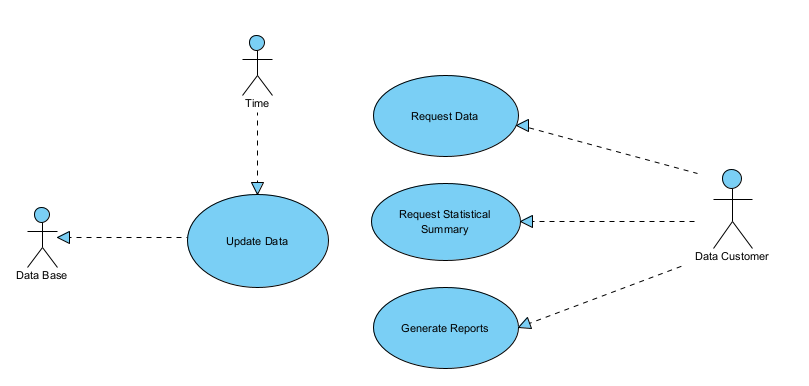
En este caso de identificaron dos actores principales: El tiempo y los consumidores de estos datos y como actor secundario se identificó a las fuentes de datos (bases de datos de las plataformas). En cuanto a los casos de uso se identificaron fundamentalmente 4: Solicitar datos, Solicitar resumen estadístico, Generar reportes y actualizar datos.

El caso de uso Solicitar datos inicia cuando un actor Consumidor de datos hace una consulta predefinida o diseñada al sistema solicitando un conjunto de datos. Si la consulta no está predefinida, el usuario tiene que crearla. Una vez lista la consulta el sistema recopila los datos solicitados. Luego el sistema prepara los datos para la visualización. El caso de uso concluye con la entrega de los datos solicitados.

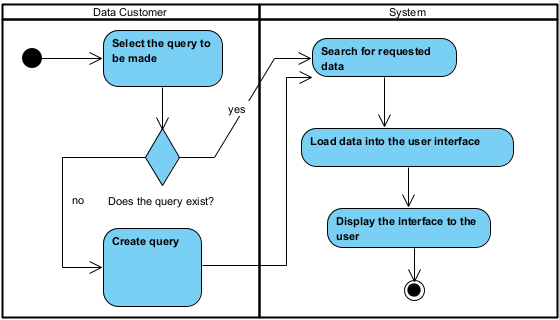
El segundo caso de uso, Solicitar Resumen Estadístico, inicia cuando el actor Consumidor de datos Solicita un resumen de diferentes métricas estadísticas predefinidas. El sistema realiza las operaciones estadísticas y las prepara para ser mostradas. El caso de uso culmina con la entrega del resumen al Consumidor de Datos.

El tercer caso de uso, Generar Reportes, inicia cuando el actor data Customer solicita la creación de un reporte al sistema. Este genera un archivo de hoja de cálculo con la extensión xlxs e introduce los datos en la misma. El caso de uso concluye con el aviso de la conclusión de la generación del reporte.

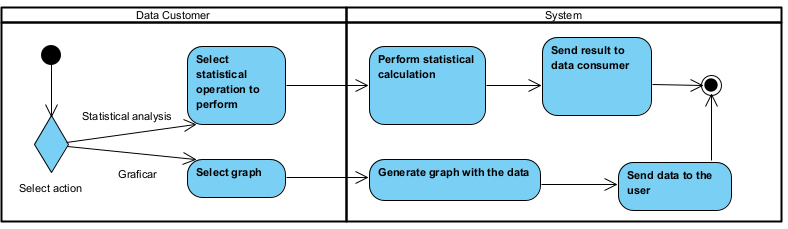
El último caso de uso, Actualizar Datos, inicia cuando ha pasado un día desde su última actualización. El sistema solicita a las bases de datos de las plataformas los datos alojados en ella, luego el sistema supervisa si se ha modificado algún campo de las filas viejas o se ha agregado alguna fila. Si algún dato fue modificado, se actualiza en el sistema. El sistema finaliza con la revisión de todas las filas de las bases de datos.

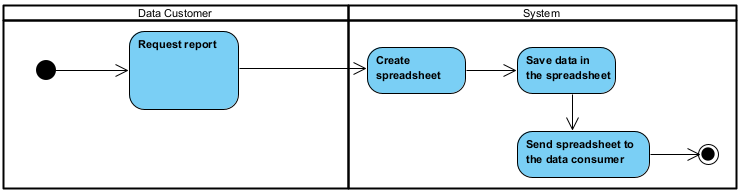
****

**Figura número 1: Diagrama de Casos de Uso**

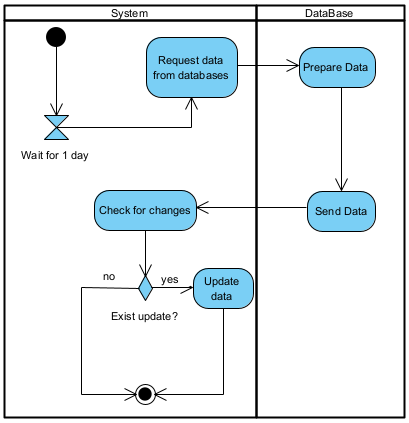
****

**Figura número 2: Diagrama de actividades del caso de uso Solicitar datos**

**Figura número 3: Diagrama de actividades del caso de uso Solicitar Resumen Estadístico**

****

**Figura número 4: Diagrama de actividades del caso de uso Generar Reportes**

****

### **2.2 Diseño de la solución**

Una de las etapas más relevantes en la resolución de un problema es el diseño de la solución. En esta fase, se representa la estructura general del sistema, la organización de sus componentes y cómo interactúan estos componentes entre ellos y con el entorno. El objetivo principal es comprender su dinámica interna y externa, asegurando que se satisfagan, en la mayor medida posible, las demandas y necesidades de la organización durante el proceso de resolución del problema. Este análisis permite obtener una visión clara y detallada de lo que se desea construir antes de proceder con la implementación (Rosado Castellanos, Pacheco Farfán, Fuentes Chab, & Cantun Páez, 2023).

El primer paso a efectuar para el diseño de la solución es **selección de las tecnologías**, pues una vez teniendo claras las necesidades del proyecto a través del análisis del negocio y del marco teórico de este campo, se está en condiciones de seleccionar que tecnologías pueden ser más útiles. Luego se procede en los **patrones de diseño** y los **patrones arquitectónicos**. Estos últimos proporcionan soluciones comprobadas para problemas recurrentes en el desarrollo de software a nivel global, promoviendo consistencia, escalabilidad y calidad en el producto final. Por otro lado, los **patrones de diseño** buscan solucionar problemas comunes, agilizando el proceso de desarrollo, garantizar la legibilidad y eficiencia del código. Asimismo, se analizarán los **principios de diseño**, estos juegan un papel crucial al garantizar que las decisiones tomadas respeten valores esenciales como el modularidad, la cohesión y la baja dependencia entre componentes (Roger & Bruce, 2015).

Además, se describirá el enfoque para el **despliegue del sistema**, priorizando una implementación eficiente y adaptable a los sistemas del cliente. Finalmente, se detallan los **mecanismos técnicos** que habilitan las funcionalidades clave del repositorio de datos, asegurando que estos componentes sean robustos, confiables y alineados con los objetivos definidos.

#### **2.2.1 Selección de las tecnologías**

La selección de las tecnologías constituye un paso crucial en el diseño de la solución, ya que estas determinarán las herramientas y frameworks o marcos de trabajo que permitirán implementar las funcionalidades requeridas, cumplir con los requisitos establecidos y garantizar la calidad del sistema. En este proyecto, las tecnologías seleccionadas han sido Java, Spring Boot, MINIO, Couch DB y Shyiko. Durante esta sección, se justificará la elección de cada una de estas tecnologías en función de su adecuación a los requisitos funcionales y no funcionales del sistema.

La primera herramienta seleccionada fue el lenguaje de programación Java debido a su robustez, versatilidad, amplia adopción en el desarrollo de aplicaciones empresariales y por la previa experiencia en el uso de la tecnología que reduce el tiempo de adopción y por consiguiente el tiempo de desarrollo. Su naturaleza multiplataforma garantiza que el sistema pueda ejecutarse en diversos entornos sin necesidad de modificaciones significativas. Además, Java cuenta con una rica colección de bibliotecas y marcos de trabajo que aceleran el desarrollo y facilitan la implementación de características complejas en el ámbito de la Ingeniería de datos. La elección de Java también responde al requisito no funcional de emplear tecnologías de código abierto, ya que el lenguaje y su ecosistema son ampliamente accesibles bajo licencias abiertas (Oracle).

Luego se necesitó la selección de un marco de trabajo que permitiera el desarrollo de una aplicación web, que es uno de los requisitos solicitados. Para ello se seleccionó Spring Boot, un marco de trabajo basado en Java diseñado para simplificar el desarrollo de aplicaciones empresariales mediante una configuración mínima y un enfoque modular. Fue seleccionado debido a la previa experiencia en el uso de la tecnología que reduce el tiempo de adopción y por consiguiente el tiempo de desarrollo y por su capacidad para construir aplicaciones escalables y seguras, lo cual es esencial para cumplir con los requisitos relacionados con la autenticación, la seguridad del acceso a los datos y la creación de consultas dinámicas. Spring Boot permite integrar fácilmente componentes adicionales como bases de datos, sistemas de almacenamiento y servicios RESTful, lo que lo convierte en una opción ideal para desarrollar un sistema que interactúe con múltiples fuentes de datos y proporcione funcionalidades avanzadas (SpringBoot).

Previendo futuras necesidades del sistema se determinó agregar además un servicio de almacenamiento de objetos por si aumenta la diversidad de fuentes de datos a ficheros Excel o csv. Para este caso se escogió MINIO. MINIO es una solución código abierto capaz de gestionar grandes volúmenes de datos no estructurados o semi-estructurados, como Excel, tablas Delta o txt. MINIO garantiza alta disponibilidad y escalabilidad, lo cual es crucial para manejar las demandas crecientes del sistema a medida que se acumulan datos históricos o se generan reportes complejos. Además, su compatibilidad con estándares abiertos asegura una integración fluida con otros componentes del sistema (MINIO, 2022).

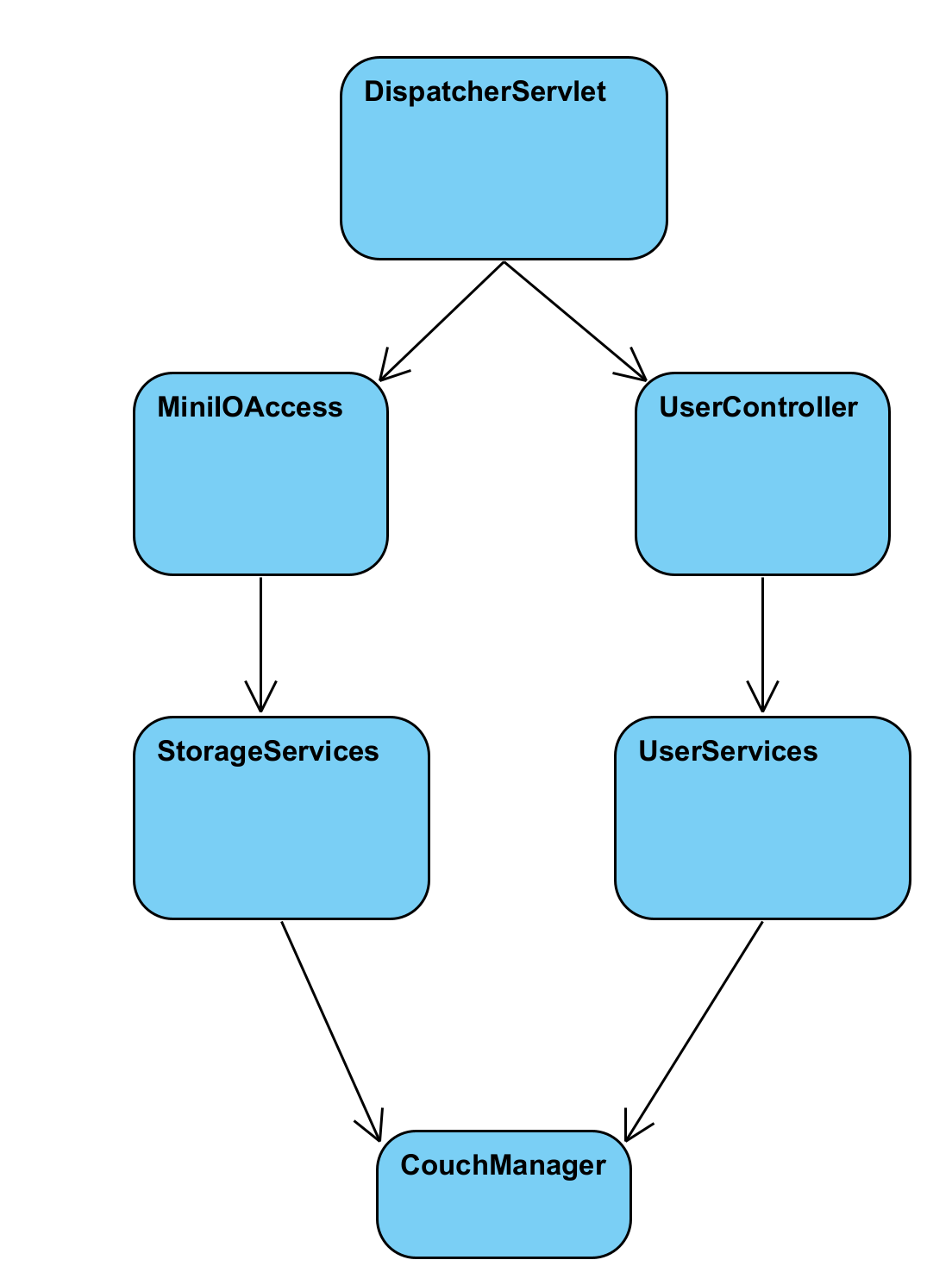
Además del sistema de almacenamiento de objetos se hizo necesario un lugar donde almacenar los datos ya estandarizados para su consumo. Para ello se escogió una base de datos documental llamada Couch BD. Fue seleccionada debido a su capacidad para manejar datos heterogéneos y dinámicos, tal como lo requieren los departamentos involucrados en este proyecto (Marketing, Ventas y Estadísticas). Su modelo flexible permite almacenar información estructurada y no estructurada sin necesidad de esquemas rígidos, lo cual facilita la combinación y consulta de datos diversos. Además, CouchDB soporta replicación eficiente entre nodos, lo cual es útil para garantizar disponibilidad y consistencia en entornos distribuidos (Apache).

Finalmente se buscó una solución que permitiera detectar de una forma lo más eficiente posibles que cambios se habían hecho sobre las bases de datos originales. Para este cometido se seleccionó mysql-binlog-connector-java (Shyiko), una herramienta especializada en la lectura y procesamiento de binlogs (archivos binarios) generados por MySQL. Su selección responde al requisito funcional relacionado con la sincronización diaria entre las bases de datos fuente y el repositorio del sistema. Shyiko permite capturar cambios realizados en las bases existentes (inserciones, actualizaciones o eliminaciones) sin afectar el rendimiento operativo del sistema fuente. Esto asegura que el repositorio refleje siempre los datos más recientes sin necesidad de realizar consultas completas a las bases originales ni tener que modificar los sistemas ya existentes.

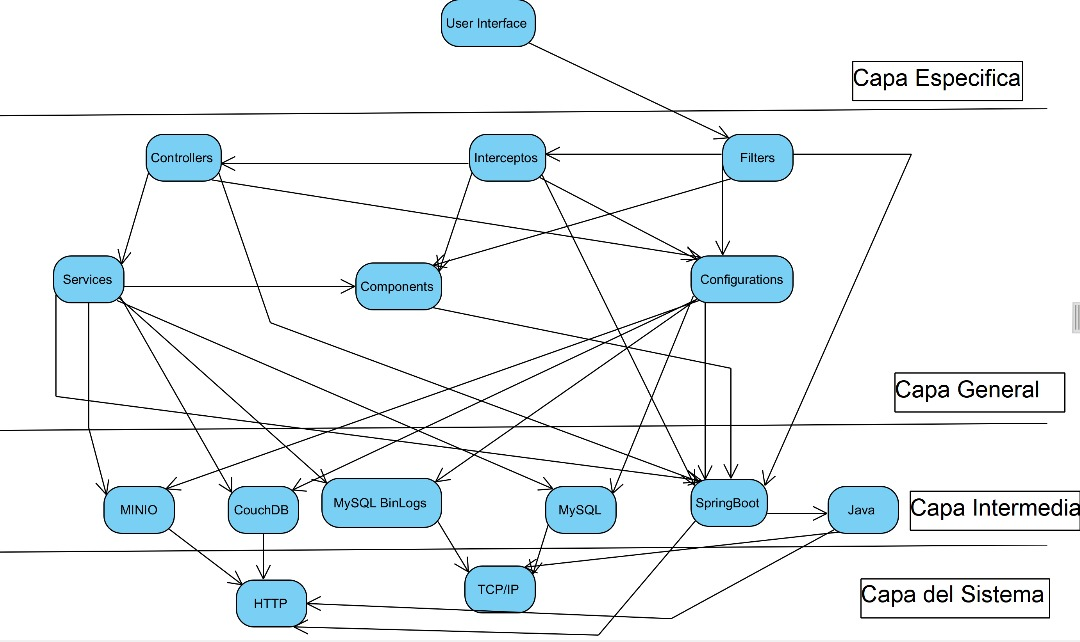
#### **2.2.2 Patrones Arquitectónicos**

Luego de tener claro con que herramientas vamos a desarrollar se hace necesario ver que patrones arquitectónicos nos podrían ayudar a desarrollar una aplicación lo más eficiente y funcional posible. En este caso los patrones identificados fueron: **llamada-Retorno, N-Capas, Cliente – Srvidor, Repositorio, Repositorio y Filtros y Tuberías.** A continuación, se describirá como se implementó cada uno en el proyecto.

El primer patrón empleado es el patrón **llamada-Retorno**, representado en la figura 5. En este se definió una jerarquía de componentes donde los componentes superiores son los que llaman a los componentes inferiores para realizar una tarea en concreta, pero nunca puede suceder al revés. O sea, cada controlador tiene un servicio asociado que es el responsable de ejecutar la lógica de negocio pertinente. A su vez, estos servicios llaman a al sistema de acceso a la base de datos Couch DB para ejecutar la consulta a los datos.

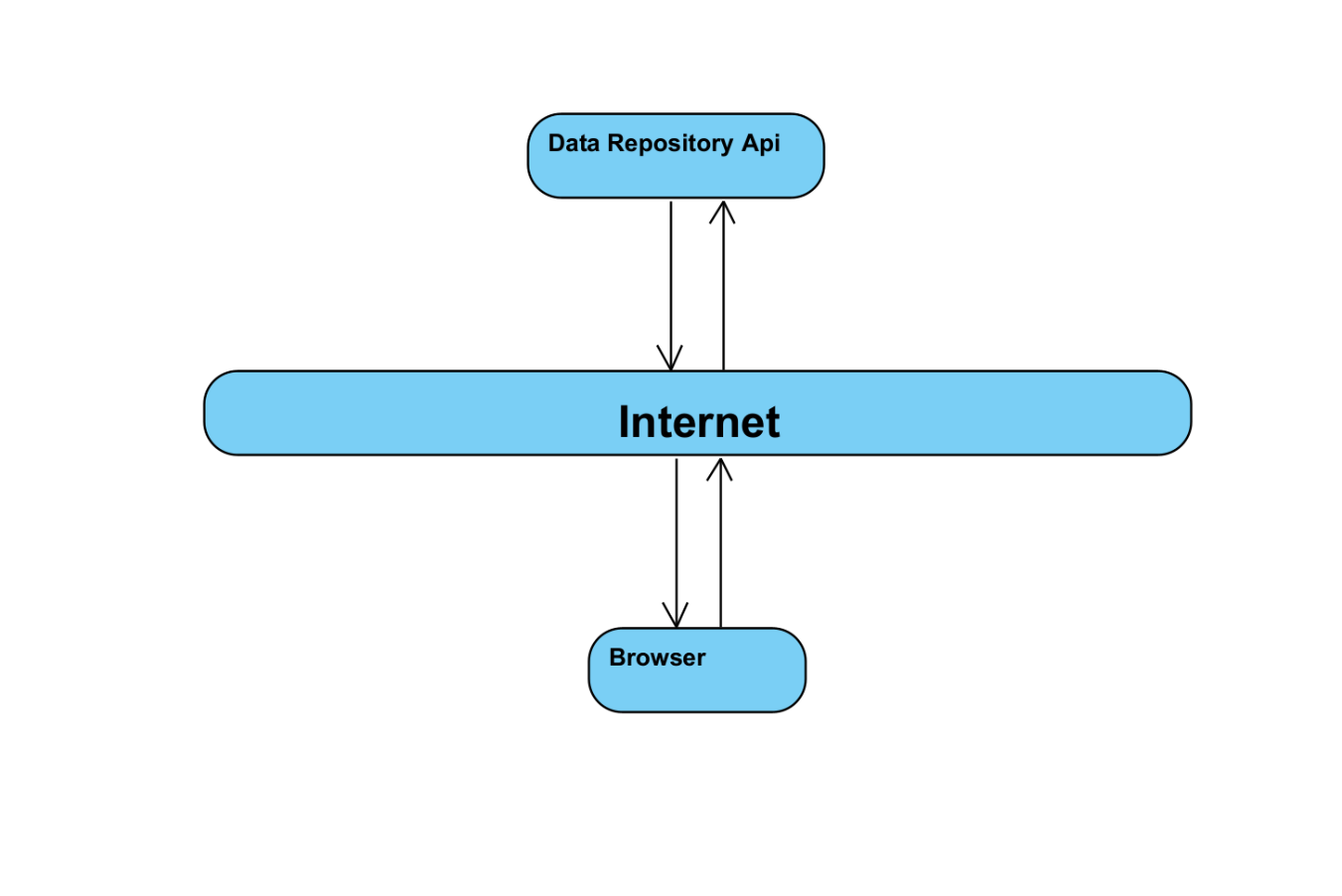
**Figura número 5: Diagrama del patrón Llamada-Retorno**

También se empleó el patrón **N-Capas**, representado en la figura número 6. En este se dividió el sistema en diferentes capas. Estas capas agrupan los diferentes componentes del sistema. En el caso de este sistema se empleó el enfoque de reutilización, que contiene las capas Especificas, donde se encuentras los componentes que se reutilizan poco o no se reutilizan. Aquí se encuentra la interfaz gráfica con la que se comunicara el usuario a través del navegador. En la segunda capa, la capa general se encuentran los componentes que son más reutilizado. Aquí encontramos los controladores que es donde se definen los puntos de acceso(endpoints) para acceder a la aplicación, los servicios donde se define la lógica del negocio, los filtros y los interceptores para realizar validaciones a los datos de entrada, las configuraciones para definir características generales de un conjunto de componentes o de toda la aplicación y los componentes donde se agregan bloques de código altamente reutilizables. En la tercera capa, Capa intermedia encontramos los componentes de terceros, en este caso tenemos SpringBoot, MINIO, MySQL, BinLogs, Java y CouchDB. En la última capa, Capa de software del sistema tenemos los protocolos HTTP y TCP/IP.

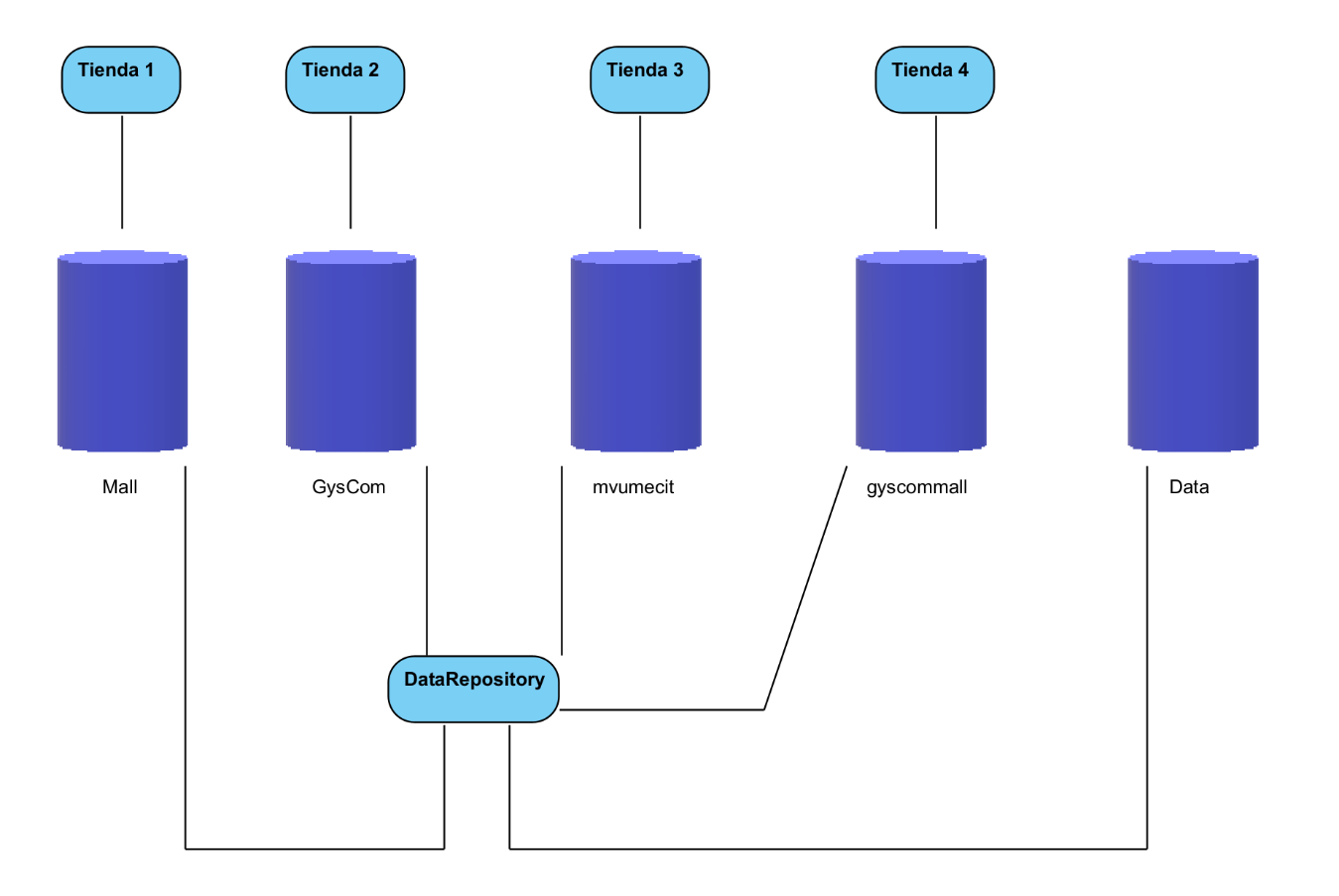


**Figura número 6: Diagrama del patrón N-Capas**

Otro patrón empleado es el **Cliente – Srvidor**, representado en la figura 7, el cual plantea dividio el sistema en una aplicación que se ejecute en un servidor remoto y sea utilizado por algún cliente como un navegador web a través de una red como internet. Concretamente se creó una aplicación Backend que será consultada desde los navegadores de los trabajadores de la empresa a través de la red local.

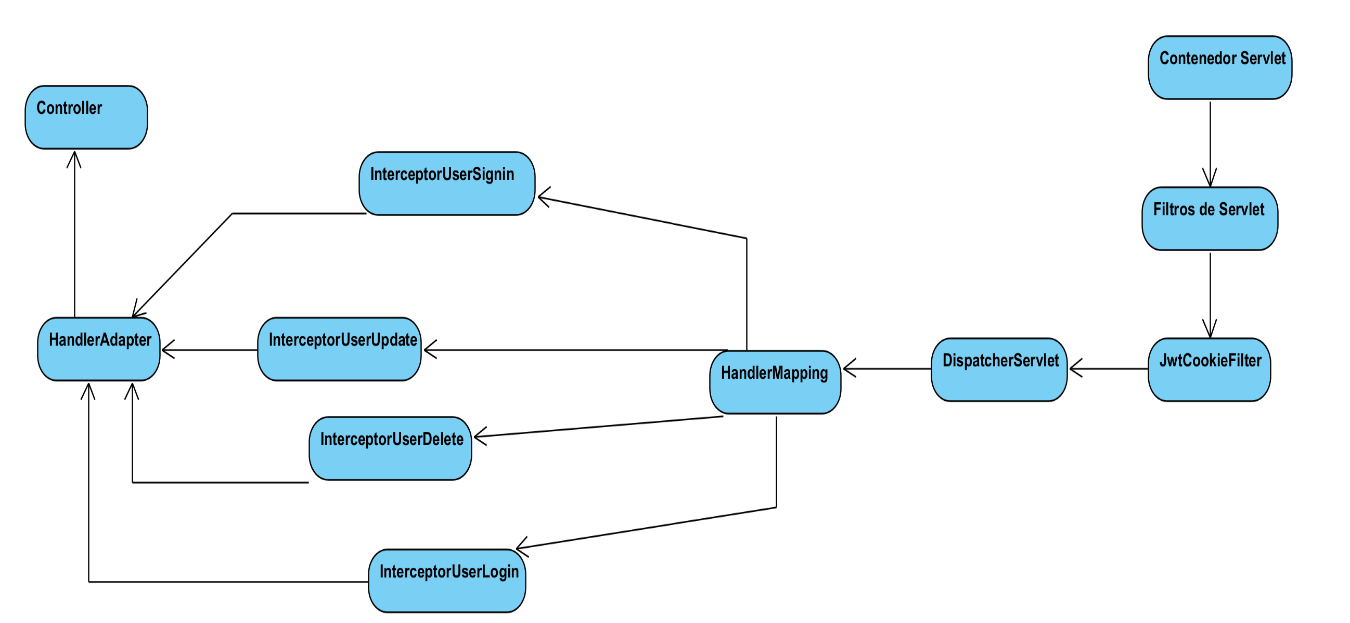
**Figura número 7: Diagrama del patrón Cliente-Servidor**

Además, se implementó el patrón **Repositorio**, representado en la figura 8. Este sistema consta de 4 bases de datos. Cada una asociada a una tienda y un sistema diferente. Pero también son utilizadas por el software creado para obtener los datos, así como también se creó otra base de datos en Couch DB para almacenar los datos en un formato nuevo, que ayude a la consulta de los mismo.

****

**Figura número 8: Diagrama del patrón Repositorio**

El último patrón implementado fue el de **Filtros y Tuberías**, representado en la figura 9. Spring Boot implementa este patrón de forma interna, y puede ser mejorado a través de las interfaces Filter e Interceptor. Este patrón se utiliza para definir un conjunto de clases que se encargan de procesar y validar los datos de manera secuencial. Cada clase actúa como un filtro que realiza una tarea específica, y los datos fluyen a través de estas etapas, permitiendo una separación clara de responsabilidades y un diseño más modular y reutilizable.

**Figura número 8: Diagrama del patrón Repositorio**

#### **2.2.3 Patrones de diseño**

Luego de haber identificado que patrones de arquitectura podrían ayudar a estructurar el proyecto, se hace necesario identificar aquellos patrones de diseño que podrían resolver ciertas situaciones problemáticas en la implementación. En concreto se detectaron dos. La primera fue que la clase ObjectMapper de la librería Jackson para el manejo de ficheros Json no admite los objetos de fecha y tiempo nativos de java y se necesitaba una infraestructura que permitiera, de forma eficiente, identificar en función de cada log que había que hacer y cómo, pues como cada base de datos tiene una estructura diferente las validaciones y consultas también son diferentes.

Para resolver el primer problema no se encontró directamente ningún patrón, pero haciendo una variación del patrón **Decorador** se descubrió que se podrían obtener los resultados deseados. El patrón decorador requiere de una interfaz que permita la creación de una clase donde se extiendan las funcionalidades de la clase inicial, en su lugar se extendió de una clase ya existente con el objetivo de adaptar una funcionalidad a las circunstancias del proyecto. En concreto, la librería clase ObjectMapper no es capaz de manejar los tipos de datos de fecha que trae java por defecto. Para resolver este problema se hizo uso de una clase anónima. Esta es una característica de java que permite instanciar y definir una clase al mismo tiempo

que deben heredar de una interfaz o clase padre. e



**Imagen número 1: Código de la implementación del patrón decorador**

Para resolver el segundo problema se empleó el patrón **Estrategia**. Para ello primero se lee el log y se determina si la acción es de creación, actualización o eliminación, luego se identifica si a que base de datos pertenece y a que tabla, en función de eso se ejecuta un algoritmo que cumpla con las condiciones para realizar la acción sobre la base de datos especificada.



**Imagen número 1: Código de la implementación del patrón Estrategia**

#### **2.2.3 Principios de diseño**

Los principios de diseño son directrices fundamentales que guían a los desarrolladores a crear un producto de alta calidad, robusto, eficiente, mantenerle, escalable y a disminuir la probabilidad de cometer un error (Richards & Ford, 2020).

En este proyecto se aplicaron cinco principios fundamentales de diseño. El **Principio de Responsabilidad Única** se implementó para lograr una mejor abstracción de cada funcionalidad, asignándola a una clase específica. Por ejemplo, la clase CouchManager se encarga exclusivamente de la comunicación con CouchDB, conteniendo todas las funcionalidades relacionadas con la conexión a esta base de datos. Este enfoque también incorpora la regla "**Once and Only Once**", ya que, al agrupar las funcionalidades referentes a una entidad en una misma clase, esa funcionalidad se encuentra únicamente en ese lugar.

El **Principio de Hollywood** permitió establecer un flujo de comunicación secuencial entre diferentes funcionalidades. Por ejemplo, los controladores encargados de recibir las solicitudes llaman a los servicios, y nunca al revés, manteniendo una jerarquía clara en el flujo de control. Esto se logró en parte gracias al principio de **favorecer la composición sobre la herencia**, que se aplicó para permitir que las funcionalidades se comuniquen a través de una composición de los servicios hacia los controladores, en lugar de depender de la herencia.

La **Ley de Demeter** se relaciona con los principios anteriores, ya que establece que las clases solo deben comunicarse con sus atributos, parámetros de métodos o con el propio método. Esto fomenta un bajo acoplamiento y favorece la composición o agregación.

Por último, se empleó el **Principio de Abstracción**, ya que se buscó abstraer las funcionalidades en clases y métodos específicos, mejorando la organización y mantenibilidad del código.

#### **2.2.4 Mecanismos**

Durante la implementación de un sistema informático es común encontrarse con bloques de código altamente reutilizables que necesitan ser utilizados varias veces. Para evitar la repetición de código se crean mecanismos para implementar estos bloques de código. Para este proyecto se empleó un mecanismo para el acceso a CouchBD y un mecanismo para la autenticación de los usuarios.

El mecanismo de acceso a datos para CouchDB implementa operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para interactuar con la base de datos NoSQL CouchDB. Este mecanismo encapsula la lógica de conexión, autenticación y manejo de errores, proporcionando una interfaz simplificada para realizar operaciones en la base de datos.

* Crear documento
  + Recibe el nombre de la base de datos y un mapa con los datos del documento.
  + Convierte el mapa a formato JSON.
  + Establece una conexión HTTP POST con CouchDB.
  + Envía el JSON como cuerpo de la solicitud.
  + Verifica la respuesta del servidor (espera código 201 Created).
* Leer documento
  + Recibe el nombre de la base de datos y el ID del documento.
  + Establece una conexión HTTP GET con CouchDB.
  + Deserializa la respuesta JSON a un mapa de datos.
* Actualizar documento
  + Recibe el nombre de la base de datos y un mapa con los datos actualizados (incluyendo \_id y \_rev).
  + Convierte el mapa a formato JSON.
  + Establece una conexión HTTP PUT con CouchDB.
  + Envía el JSON como cuerpo de la solicitud.
  + Verifica la respuesta del servidor (espera código 200 OK o 201 Created).
* Eliminar documento
  + Recibe el nombre de la base de datos, el ID del documento y la revisión actual.
  + Establece una conexión HTTP DELETE con CouchDB.
  + Verifica la respuesta del servidor (espera código 200 OK).

Por otro lado, el mecanismo de autenticación implementado, utiliza JSON Web Tokens (JWT) para gestionar las sesiones de usuario de forma segura y eficiente. Este mecanismo consta de tres componentes principales: el JwtManager, UserService y el proceso de autenticación en el UserController.

El JwtManager es un componente Spring que se encarga de la generación y validación de tokens JWT.

* Inicio de sesión:
  + El usuario proporciona su nombre y contraseña.
  + El sistema verifica las credenciales (esta parte no está visible en el código proporcionado, pero se asume que existe).
* Verificación
  + El UserService se encarga de revisar si la contraseña y el nombre de usuario son correctos.
  + De no serlo notifican del error al cliente.
* Generación y gestión del token:
  + Si las credenciales son válidas, se genera un token JWT usando el JwtManager.
  + El token se almacena en una cookie HTTP segura.
  + La cookie se configura como HttpOnly y Secure para mejorar la seguridad.
  + Se establece un tiempo de expiración para la cookie.
* Respuesta al cliente:
  + Se notifica al cliente del éxito o fracaso de la operación.

**2.2.5Modelo de despliegue**

El diagrama de despliegue ilustra la arquitectura de la API del proyecto. Esta cuenta con un módulo para el despliegue de CouchDB, otro para el despliegue de las 4 Bases de datos de MySQL y una para el servidor de la aplicación Backend y una para el servidor de la aplicación Frontend.

Características de cada nodo:

**Nodos de MySQL**

Requisitos de Hardware:

Procesador: Intel Core i3 2.0 GHz o superior

RAM: 4 GB mínimo

Almacenamiento: 1.5 GB mínimo por base de datos

Requisitos de Software:

Sistema Operativo: Linux

MySQL Server versión compatible

**Nodo CouchDB**

Requisitos de Hardware:

Procesador: 2 vCPUs mínimo

RAM: 4 GB mínimo

Almacenamiento: 4.5GB espacio libre en disco

Requisitos de Software:

Sistema Operativo: Linux

Apache CouchDB 3.0 o superior

**Nodo DataRepository (Spring Boot)**

Requisitos de Hardware:

Procesador: Intel Core i5 o superior

RAM: 8 GB recomendado

Almacenamiento: 10 GB

Requisitos de Software:

JDK 21 o superior

Spring Boot 3.4.1

Apache Tomcat 10.1 o superior

Maven 3.6.3 o superior

**Nodos MinIO**

Requisitos de Hardware:

Procesador: 8 vCPUs por nodo

RAM: 32 GB por nodo

Almacenamiento: 40 GB mínimo

Requisitos de Software:

Sistema Operativo: Linux

MinIO Server

MINIO Client

**Nodo Cliente**

Requisitos de Hardware:

Procesador: Intel Core i5 o superior

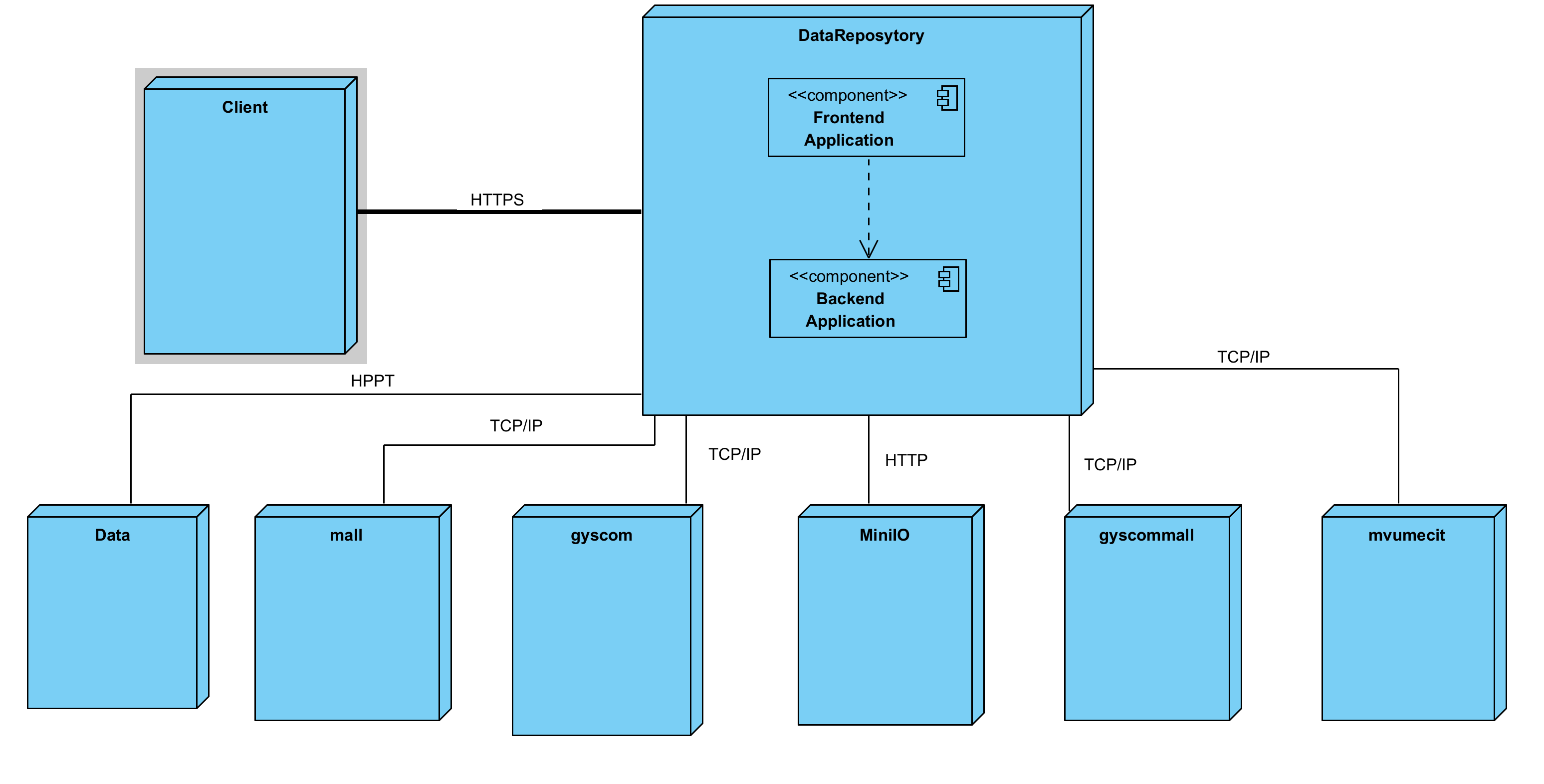
RAM: 4 GB

Almacenamiento: -

Requisitos de Software:

Sistema Operativo: Windows 10/11, Linux o MacOS

Navegador web moderno (Chrome, Firefox, Safari)

****

**Figura numero 10: Diagrama de despliegue.**

## **Capítulo 3: Diseño y ejecución de pruebas**

El presente capítulo tiene como objetivo documentar el proceso de evaluación del proyecto mediante la realización de pruebas diseñadas para validar su correcto funcionamiento, desempeño y cumplimiento de los objetivos planteados. Estas pruebas son fundamentales para garantizar que la solución implementada cumpla con los requisitos del negocio y las especificaciones técnicas definidas en las fases previas del proyecto. Además, permiten identificar posibles áreas de mejora y asegurar la calidad del sistema desarrollado.

En primer lugar, se definen los parámetros clave que serán utilizados como criterios de evaluación. Estos parámetros incluyen métricas relacionadas con el rendimiento, la integridad de los datos, la escalabilidad y los tiempos de ejecución. La selección de estos parámetros responde a los objetivos específicos del proyecto y a las mejores prácticas en el diseño y evaluación de sistemas de gestión de datos empresariales.

Posteriormente, se describe el diseño de las pruebas realizadas, detallando los escenarios planteados, las herramientas utilizadas y los métodos empleados para medir el desempeño del sistema. Este diseño busca cubrir tanto aspectos funcionales como no funcionales, asegurando una evaluación integral que considere tanto el comportamiento esperado como las posibles limitaciones del sistema en situaciones reales o simuladas.

Finalmente, se presentan los resultados obtenidos durante la ejecución de las pruebas, acompañados de un análisis crítico que permite interpretar los datos recopilados. Este análisis no solo evalúa el cumplimiento de los objetivos iniciales, sino que también proporciona recomendaciones para futuras iteraciones o mejoras en la solución implementada. De esta manera, este capítulo contribuye a cerrar el ciclo de desarrollo del proyecto al proporcionar evidencia objetiva sobre su efectividad y alineación con las necesidades del negocio.

### **Parámetros para la evaluación**

En esta sección se describen los parámetros seleccionados para evaluar el desempeño y la calidad del repositorio de datos basado en un Data Lake. Estos parámetros han sido definidos considerando las características técnicas del sistema, los objetivos del proyecto y las necesidades del negocio, permitiendo medir aspectos clave que aseguran la efectividad y eficiencia de la solución implementada.

El primer parámetro considerado es **la integridad de los datos**, que se refiere a la capacidad del sistema para garantizar que los datos almacenados sean consistentes, completos y libres de errores durante todo su ciclo de vida. Este parámetro evalúa si los datos transferidos al Data Lake coinciden con las fuentes originales y si se mantienen intactos tras procesos como transformaciones o consultas. Su correcta implementación asegura la confiabilidad de los datos para análisis y toma de decisiones.

La **tasa de ingestión de datos** también es un aspecto crítico a evaluar, ya que mide la velocidad a la que el sistema puede incorporar nuevos datos desde diversas fuentes. Este parámetro se evalúa en términos de volumen, como gigabytes por hora, y permite determinar si el Data Lake es capaz de manejar grandes cantidades de información sin afectar su desempeño. Asimismo, se considera la **optimización del almacenamiento**, que mide la eficiencia con la que el sistema utiliza los recursos disponibles. Esto incluye el uso de formatos comprimidos, estrategias jerárquicas como almacenamiento caliente o frío, y la relación entre el espacio utilizado y el volumen total de datos almacenados. Finalmente, se evalúa el **costo por operación o consulta**, un parámetro que analiza la relación entre los costos operativos asociados al uso del Data Lake, como almacenamiento y procesamiento, y el valor generado por las operaciones realizadas. Esta métrica permite identificar oportunidades para optimizar recursos y mejorar la eficiencia económica del sistema. En conjunto, estos parámetros ofrecen una evaluación integral tanto técnica como funcional del proyecto, asegurando su alineación con los objetivos planteados y las necesidades del negocio.

### **Diseño de las pruebas**

Una vez identificados que parámetros necesitamos medir para identificar las potencialidades y deficiencias del sistema, procedemos al diseño de las pruebas que nos permitirán hacer la evaluación.

Primeramente, se evaluará la integridad de los datos transmitidos. Para ello, en primer lugar, se seleccionaron los datos a transmitir desde las cuatro bases de datos originales. Estos conjuntos de datos consisten en registros simbólico de datos ficticios. Posteriormente, los datos seran transmitidos hacia el Data Lake utilizando el canal de comunicación establecido entre las bases de datos fuente y CouchDB. Una vez que los datos llegaron al Data Lake se evaluaran si estos datos coinciden con los datos originales. Para ellos se utilizarán varias veces los mismos datos.

Para evaluar la taza de ingesta de datos se empleará una metodología basada en simular la transferencia de datos con diferentes volúmenes y registrar el tiempo necesario para completar cada operación, calculando así la tasa efectiva de ingestión en megabytes por segundo (MB/s). Para realizar esta prueba, se definieron cinco escenarios con volúmenes representativos: 20,50,100 y 200MB.

El procedimiento comenzó con la transmisión del primer conjunto de datos (20 MB) desde una base de datos fuente hacia el Data Lake. Una vez iniciado el proceso se registra la hora en milisegundos. Una vez finalizado el primer bloque de datos se resta la hora de fin con la registrada al inicio.

Para la medición del parámetro de optimización del almacenamiento se reutilizo este mismo caso de prueba. Para ello se registró el tamaño de estos datos en la base de datos en CouchDB y la diferencia entre estos y los pesos originales.

Finalmente, para el ultimo parámetro se registró los beneficios estas tecnologías contra el costo computacional en memoria, almacenamiento y tiempo de demora.

### **Resultados de las pruebas**

Una vez diseñadas las pruebas, se procedió a su ejecución utilizando cuatro conjuntos de datos seleccionados de la base de datos mall, específicamente de la tabla comprador. Estos conjuntos fueron empleados para evaluar diferentes escenarios y tenían tamaños de 20 MB, 50 MB, 100 MB y 200 MB, equivalentes a aproximadamente 18 mil, 45 mil, 90 mil y 180 mil filas respectivamente. Todos los conjuntos contenían los mismos atributos y estructuras de datos, lo que permitió garantizar la consistencia en las condiciones de prueba.

Las pruebas consistieron en evaluar el rendimiento del sistema al transferir los datos hacia el Data Lake implementado en CouchDB, registrando tanto los tiempos de procesamiento como el volumen final almacenado. En el primer caso, con un conjunto de datos de 20 MB, el sistema completó la operación en 5 minutos, y el volumen almacenado en CouchDB aumentó en 15 MB. Para el segundo caso, con un conjunto de 50 MB, la transferencia tomó 13 minutos y medio, registrándose un incremento en el almacenamiento de 35.8 MB. En el tercer escenario, correspondiente a un conjunto de datos de 100 MB, la operación se completó en 28 minutos y medio, con un aumento del volumen almacenado de 71.7 MB. Finalmente, para el último caso, con un conjunto de datos de 200 MB, el proceso tomó 48 minutos y resultó en un incremento del almacenamiento en CouchDB de 140.3 MB.

En todos los casos evaluados, no se detectaron datos corruptos ni inconsistencias entre los registros almacenados en CouchDB y los conjuntos originales. Esto confirma que el sistema fue capaz de mantener la integridad de los datos durante todo el proceso de transferencia. En la tabla número 2 se hace un resumen de estos dato.

Adicionalmente, se evaluó el costo computacional del sistema mediante el monitoreo del uso de recursos como RAM y CPU durante la ejecución de las pruebas. En las dos últimas pruebas (100 MB y 200 MB), se observó un consumo elevado de recursos computacionales debido al tamaño creciente de los datos procesados. Este comportamiento sugiere la necesidad de optimizar ciertos aspectos del sistema para mejorar su eficiencia bajo cargas altas. Sin embargo, considerando que las pruebas se realizaron con otras aplicaciones ejecutándose simultáneamente en el equipo, los resultados obtenidos en términos de tiempos y almacenamiento son satisfactorios y compensan esta vulnerabilidad.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Volumen de datos en MB | 20 | 50 | 100 | 200 |
| cANTIDAD DE FILAS(MILES) | 18 | 45 | 90 | 180 |
| Demora en minutos | 5 | 13:30 | 28:30 | 48 |
| Volumen en el data Lake | 15 | 35.8 | 100 | 140.3 |
| Datos corruptos | 0 | 0 | 0 | 0 |

**Tabla número 2: Resumen de resultados**

En conclusión, los resultados indican que el sistema es capaz de cumplir con las necesidades identificadas por la empresa para almacenar y consultar grandes volúmenes de datos provenientes de diversas fuentes. Esto valida que la solución implementada satisface los requisitos funcionales definidos durante el diseño presentado en el capítulo dos.

# **Conclusiones**

Este trabajo demuestra que es posible garantizar la consistencia de los datos en entornos complejos y heterogéneos mediante la implementación de un Data Lake. La creación de una arquitectura unificada para el almacenamiento y gestión de grandes volúmenes de datos no solo permitió resolver discrepancias existentes, sino que también estableció una base sólida para el manejo eficiente de datos a largo plazo. Este enfoque asegura que la organización esté preparada para adaptarse a las crecientes demandas de procesamiento y análisis de información en el futuro.

Uno de los principales logros del proyecto fue la ingesta y almacenamiento de datos, un desafío significativo debido al estado actual de las bases de datos de CITMATEL y a la diversidad de las fuentes de datos involucradas. A pesar de estas dificultades, se alcanzó una solución funcional que habilita a los equipos de marketing, ventas y estadísticas de CITMATEL a realizar estudios y análisis más efectivos sobre los datos generados durante sus operaciones diarias. Esto representa un avance importante en la toma de decisiones basada en datos.

Adicionalmente, se garantizó la escalabilidad del sistema mediante la integración de un sistema avanzado de almacenamiento de ficheros. Esta mejora no solo amplía las capacidades para incorporar nuevos tipos de fuentes de datos, sino que también facilita la implementación futura de funciones automatizadas para el procesamiento y análisis avanzado. De esta manera, se asegura que el sistema pueda evolucionar junto con las necesidades organizacionales, consolidándose como una herramienta estratégica para el manejo integral de información.

# **Referencias bibliográficas**

Aissi, M. E. M. E., Benjelloun, S., Loukili, Y., Lakhrissi, Y., Boushaki, A. E., Chougrad, H., & Ali, S. E. B. (2021). Data Lake Versus Data Warehouse Architecture: A Comparative Study. *Lecture Notes in Electrical Engineering*.

Amid, C., Pakseresht, N., Silvester, N., Jayathilaka, S., Lund, O., Dynovski, L. D., . . . Alako, B. T. (2019). The COMPARE data hubs. *Database, 2019*, baz136.

Apache. Overview — Apache CouchDB® 3.1 Documentation. *docs.couchdb.org*. Retrieved from <https://docs.couchdb.org/en/stable/>

Arturo, H., & Fernández, F. (2012). *Inteligencia de negocios como apoyo a la toma de decisiones en la gerencia Business Intelligence as Support of Decisions Making in Management*.

Ayala, J. L., Ortíz, J., Guevara, C. P., & Maya, E. (2018). *Herramientas de Business Intelligence (BI) modernas, basadas en memoria y con lógica asociativa*.

Biroğul, S., & Gültekin, H. (2016). Importance of Business Intelligence Solution on Decision-Making Process of Companies. *International Journal of Applied Mathematics, Electronics and Computers, 4*, 86-89.

Calle García, A. J., Alvarado Choez, K. J., Anzules Cruz, J. J., & Menéndez Parrales, A. G. (2024). NUEVAS HERRAMIENTAS TECNOLÓGICAS EN LA RECOPILACIÓN DE DATOS DE MERCADO. *Ciencia y Desarrollo*.

Dmitriyev, V., Mahmoud, T., & Marín-Ortega, P. M. (2022). SOA enabled ELTA: approach in designing business intelligence solutions in Era of Big Data. *International Journal of Information Systems and Project Management*.

Guerrero Garcia, M. J., & Rodas-Silva, J. (2022). Análisis comparativo de metodologías y herramientas tecnológicas para procesos de Business Intelligence orientado a la toma de decisiones. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones*.

Hubspot. (2021). Silos de datos: qué son, qué los causa y cómo solucionarlos. *Hubspot.es*. Retrieved from <https://blog.hubspot.es/marketing/silos-de-datos>

Ibm. (2024). IBM Cognos Analytics. [*www.ibm.com*](www.ibm.com). Retrieved from <https://www.ibm.com/es-es/products/cognos-analytics>

Jaleel, R. A., & Abbas, T. M. J. (2020). Design and Implementation of Efficient Decision Support System Using Data Mart Architecture. *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 1-6.

Juárez, A., & Francisco, J. A. (2019). *La inteligencia de negocios como una herramienta estratégica y de apoyo a la toma de decisiones en el área de ventas de la empresa Benites SRL*.

Kumar, S. (2023). Data Silos A Roadblock for AIOps. *ArXiv, abs/2312.10039*.

Llave, M. R. (2018). Data lakes in business intelligence: reporting from the trenches. *Procedia computer science, 138*, 516-524.

Microsoft. (2015). Power BI: visualización de datos | Microsoft Power Platform. [*www.microsoft.com*](www.microsoft.com). Retrieved from <https://www.microsoft.com/es-es/power-platform/products/power-bi>

MINIO. (2022). MinIO Object Storage for Kubernetes — MinIO Object Storage for Kubernetes. *Min.io*. Retrieved from <https://min.io/docs/minio/kubernetes/upstream/index.html>

Nedelcu, L. (2024). Exploring Google Analytics 4: Unveiling Insights through Data Analysis.

Oracle. Java Documentation. *Oracle Help Center*. Retrieved from <https://docs.oracle.com/en/java/>

Ordoñez Cuthbert, D. K., & Sambola, D.-M. (2023). Herramienta basada en Inteligencia de Negocios y Analíticas para la toma de decisiones académicas. Caso de Bluefields Indian & Caribbean University. *Revista Científica de FAREM-Estelí*.

Qlik. (2022). Qlik | Analytics & Data Integration Platform. *Qlik*. Retrieved from <https://www.qlik.com>

Ratnaparkhi, R. S. (2019). The Extract Transform Load (ETL) Process and Optimization using Ab Initio. *International Journal for Research in Applied Science and Engineering Technology*.

Richards, M., & Ford, N. (2020). *Fundamentals of software architecture: an engineering approach*: O'Reilly Media.

Robles Angeles, L. J., Robles Angeles, M. B., Ángeles Hernández, L., & Acosta Miranda, M. L. (2023). El poder de los datos: Cómo ETL reformuló la gestión de datos. *Programación Matemática y Software*.

Roger, S. P., & Bruce, R. M. (2015). Software engineering: a practitioner’s approach. In: McGraw-Hill Education.

Romero-Chuquital, A., & Melendres-Velasco, J. J. (2023). Uso de data Warehouse para la toma de decisiones empresariales: una revisión literaria. *Revista Científica de Sistemas e Informática, 3*(2), e543-e543.

Rosado Castellanos, D. U., Pacheco Farfán, I. S., Fuentes Chab, I. H., & Cantun Páez, J. C. (2023). Arquitectura de software para el desarrollo de aplicaciones web orientada a micro-servicios en TecNM campus Escárcega. *Programación Matemática y Software*.

Software, T. (2019). Tableau Public. *Tableau Public*. Retrieved from <https://public.tableau.com>

Sommerville, I. (2011). Software engineering (ed.). *America: Pearson Education Inc*.

SpringBoot. Spring Boot Documentation. *docs.spring.io*. Retrieved from <https://docs.spring.io/spring-boot/index.html>

Statistical Analysis, S. (2023). SAS: Analítica, Inteligencia Artificial y Gestión de Datos. [*www.sas.com*](www.sas.com). Retrieved from <https://www.sas.com/es_mx/home.html>

Sulova, S. (2019). *The Usage of Data Lake for Business Intelligence Data Analysis.* Paper presented at the International Conference Information and communication technologies in business and education.

Torres, L., & Velastegui, E. A. S. (2020). *Creación de una solución de inteligencia de negocios para la generación de reportes con información relevante del proceso de titulación de los estudiantes de la Carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Sede Quito*.

Valle, V. A. H., Rocano, W. R. P., Siguenza-Guzman, L., Cattrysse, D., & Galarza, V. H. S. (2014). *Diseño e Implementación de un Sistema de Soporte de Decisiones para el Centro de Documentación Regional “Juan Bautista Vázquez”*.

Vásquez, A. C., Martinez, G., Aguilera, F. J. E., & Gil, M. V. L. (2018). *EVOLUCIÓN DE LOS SISTEMAS DE SOPORTE A LA TOMA DE DECISIONES*.

Vite, V. J., Villegas, S. C., & Pizarro, G. (2018). Aplicación de un Spatial Data Warehouse en la gestión de proyectos de vinculación: Caso de Estudio. *Journal of Science and Research: Revista Ciencia e Investigación, 3*(9), 19-24.

Zhou, J., San, O. T., & Liu, Y. (2023). Design and Implementation of Enterprise Financial Decision Support System Based on Business Intelligence. *International Journal of Professional Business Review*.

Zoho. (2019). Data Analytics Platform for Organizations - Zoho Analytics. *Zoho*. Retrieved from <https://www.zoho.com/analytics/data-analytics-organization.html>