



Optimización de la Gestión de Inventarios en una Empresa de Operaciones Logísticas:

Diseño e Implementación de un Nuevo Módulo y una Nueva ETL

Juan Pablo Yarce Escobar

Informe de Práctica Empresarial para optar al título de Ingeniero de sistemas

Asesor

Jeysson Pérez Gómez, Especialista en Gerencia integral

Universidad de Antioquia

Facultad de ingeniería

Ingeniería de sistemas

Medellín

2024

Cita	Yarce Escobar, J. P. (2024)
Referencia	Yarce Escobar, J. P. (2024) Optimización de la Gestión de Inventarios en una Empresa de Operaciones Logísticas Semestre de industria. Universidad de Antioquia, Medellín UdeA (
Estilo APA 7 (2020)	



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes

Decano/Director: Julio César Saldarriaga

Jefe departamento: Danny Alejandro Múnera Ramírez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

Dedico este trabajo a mis padres por toda la confianza depositada en mi.

Agradecimientos

Presento especiales agradecimientos a la empresa Extiblu SAS por darme la oportunidad de crecer en mi ámbito profesional al darme la confianza para desempeñar las funciones realizadas en este proyecto.

Tabla de contenido

Resumen	8
Abstract	9
Introducción	10
1 Objetivos	11
1.1 Objetivo general	11
1.2 Objetivos específicos	11
2 Marco teórico	12
3 Metodología	15
4 Resultados	16
5 Análisis	20
6 Conclusiones	21
7 Recomendaciones	22
Referencias	23
Anexos	24

Lista de figuras

Figura 1: Representación gráfica de un DAG	17
Figura 2: Diseño del proceso migración gastos de importación	22

Siglas, acrónimos y abreviaturas

DAG	Directed Acyclic Graph
ELT	Extracción, carga y transformación
ERP.	Enterprise Resource Planning
ETL	Extracción, transformación y carga
UdeA	Universidad de Antioquia

Resumen

En el contexto de una empresa de operaciones logísticas que busca mejorar su sistema de inventarios, se propone desarrollar un nuevo módulo de inventarios y una ETL para optimizar la gestión de mercancías. El objetivo principal es abordar las deficiencias del sistema actual, un monolito en PHP Laravel, mediante la migración a tecnologías más modernas como Apache Airflow mejorando la escalabilidad y eliminando inconsistencias en el proceso y dándole un enfoque en la automatización, separación de responsabilidades e integridad de la información. El enfoque metodológico seleccionado es “Dragonfly”, una metodología ágil que permitirá una ejecución iterativa y colaborativa del proyecto.

Esto contribuirá a una gestión de inventarios más eficiente y precisa, reduciendo las posibilidades de errores y garantizando una mayor adaptabilidad a futuras demandas del negocio.

Palabras clave: inventarios, ETL, apache airflow, automatización, pipeline, ingeniería de datos, operaciones logísticas, escalabilidad.

Abstract

In the context of a logistics operations company looking to improve its inventory system, it is proposed to develop a new inventory module and an ETL to optimize the management of goods. The main objective is to address the shortcomings of the current system, a monolith in PHP Laravel, by migrating to more modern technologies such as Apache Airflow improving scalability and eliminating inconsistencies in the process and giving it a focus on automation, separation of responsibilities and integrity of information. The selected methodological approach is “Dragonfly”, an agile methodology that will allow an iterative and collaborative execution of the project.

This will contribute to a more efficient and accurate inventory management, reducing the chances of errors and ensuring greater adaptability to future business demands.

Keywords: inventories, ETL, apache airflow, automation, pipeline, data engineering, logistics operations, scalability.

Introducción

El presente proyecto surge en el contexto de Extiblu, una empresa ubicada en la zona franca de Rionegro, la cual se dedica principalmente a las operaciones logísticas y que viene desde el 2005 aproximadamente ofreciendo servicios de compras, importación, nacionalización, alistamiento, acondicionamiento, despacho, entre otros, con el objetivo de permitir a sus clientes enfocarse exclusivamente en sus actividades de venta mientras que todo lo relacionado al contacto con el proveedor, importe del producto y acondicionamiento de la mercancía viene por parte de la empresa. Sin embargo, el sistema actual de gestión de inventarios presenta limitaciones significativas debido a su estructura monolítica desarrollada en PHP Laravel. Esta limitación impide una escalabilidad adecuada y genera inconsistencias en la gestión de inventarios. Por lo tanto, este proyecto tiene como objetivo principal abordar estas deficiencias mediante el desarrollo de un nuevo módulo de inventarios y una nueva ETL desde cero con un nuevo lenguaje de programación. Se realizará con C# en la tecnología .NET, junto a la implementación de la metodología ágil “dragonfly” de GML software. La ETL por su lado se realizará con la tecnología apache Airflow, la cual permitirá la automatización de distintas tareas repetitivas a la par que un manejo de la información mucho más íntegro, funcional y seguro.

1 Objetivos

1.1 Objetivo general

Diseñar, desarrollar e implementar un nuevo módulo de inventarios y una ETL para optimizar la gestión de mercancías en una empresa de operaciones logísticas, mejorando la escalabilidad y eliminando inconsistencias.

1.2 Objetivos específicos

- Diseñar y desarrollar un nuevo módulo de inventarios que permita una gestión eficiente y precisa de las mercancías, considerando los requisitos específicos de la empresa y garantizando su integración con el sistema existente.
- Diseñar, desarrollar y configurar una ETL para facilitar la extracción, transformación y carga de datos de manera automatizada, asegurando la integridad y consistencia de la información en el sistema de gestión de inventarios.
- Realizar pruebas exhaustivas de los nuevos módulos de inventarios y la ETL para validar su funcionamiento correcto, identificar posibles fallos y garantizar su compatibilidad con el sistema actual, asegurando una transición sin contratiempos.
- Implementar el módulo de inventarios y la ETL para corroborar el funcionamiento adecuado de cada una de las partes ya en conjunto.

2 Marco teórico

La gestión de inventarios es un aspecto fundamental de cualquier empresa que maneje productos físicos, ya que implica el control y supervisión del flujo de mercancías desde la recepción hasta la distribución (Chopra y Meindl, 2016). La eficiente gestión de inventarios permite a las empresas garantizar la disponibilidad de productos, minimizar los costos de almacenamiento y evitar pérdidas por obsolescencia o exceso de inventario.

En el contexto de la gestión de inventarios, la integración de datos es un aspecto crucial para obtener una visión completa y precisa del inventario (Han et al., 2011). El proceso de integración de datos implica la extracción, transformación y carga (ETL) de datos de diferentes fuentes, como sistemas de punto de venta, sistemas de gestión de almacenes y sistemas de planificación de recursos empresariales (ERP) (Elbashir et al., 2017). La ETL es un proceso de integración de datos que consiste en extraer datos de diferentes fuentes, transformarlos en un formato coherente y cargarlos en una base de datos o un sistema de información (Kimball y Ross, 2002) lo cual permite a las empresas integrar datos de diferentes sistemas y fuentes, mejorar la calidad de los datos y facilitar la toma de decisiones basadas en datos.

Una alternativa a la ETL es el proceso de extracción, carga y transformación (ELT). A diferencia de la ETL, en ELT los datos se cargan primero en el sistema de destino y luego se transforman. Esto puede ser beneficioso cuando se dispone de sistemas de almacenamiento de datos altamente escalables y potentes, ya que permite realizar transformaciones más complejas y rápidas directamente en el sistema de destino.

La automatización en la gestión de inventarios y en los procesos de integración de datos ha cobrado gran importancia en los últimos años. Herramientas como Apache Airflow permiten la automatización de flujos de trabajo de ETL, facilitando la orquestación y monitoreo de tareas de manera eficiente. La automatización no solo reduce la intervención manual, sino que también

mejora la precisión y la consistencia de los datos, asegurando que la información sea actual y precisa para la toma de decisiones.

3 Metodología

La metodología ágil "Dragonfly", desarrollada por GML Software, fue implementada de manera rigurosa y estructurada en cada fase del proyecto, asegurando que todos los principios de colaboración, transparencia y calidad fueran integrados profundamente en el proceso de desarrollo. A continuación, se describe cómo se aplicó esta metodología en cada una de las fases del proyecto.

1. Exploración y Concepción de la Iteración

La fase de exploración y concepción fue el punto de partida crítico en la implementación de "Dragonfly". Este proceso comenzó con la identificación y definición de los requerimientos iniciales del proyecto. Se realizaron sesiones de descubrimiento con los stakeholders clave, donde se exploraron las necesidades del negocio, los objetivos del proyecto, y las expectativas de los usuarios finales. A través de estas actividades, se definieron las iteraciones clave, que fueron concebidas como bloques de trabajo manejables que permitirían avances progresivos hacia la meta final del proyecto.

2. Diseño Colaborativo y Planeación

Una vez definidos los requerimientos, se avanzó a la fase de diseño colaborativo y planeación. Aquí, la metodología "Dragonfly" puso en práctica su enfoque único de involucrar a todo el equipo, no solo en la planificación técnica, sino también en la alineación estratégica con los objetivos del negocio. Se utilizaron herramientas de diseño colaborativo, como pizarras digitales y software de prototipado, para crear maquetas visuales y diagramas de flujo que ilustran claramente cómo se desarrollaría cada iteración.

Se llevaron a cabo reuniones diarias de planificación, inspiradas en los principios de SCRUM, donde se revisaron las tareas pendientes, se ajustaron prioridades, y se resolvieron obstáculos. Este enfoque permitió una planificación dinámica y ajustada a las necesidades cambiantes del proyecto.

3. Construcción e Ingeniería de Calidad

La fase de construcción fue donde las ideas y planes previamente concebidos comenzaron a tomar forma tangible. Un aspecto fundamental de "Dragonfly" fue su fuerte enfoque en la calidad desde el inicio. Se aplicaron prácticas de ingeniería de calidad como la integración continua, donde cada pieza de código fue automáticamente probada e integrada al sistema principal para detectar y corregir errores de manera temprana. Esto evitó la acumulación de defectos y permitió mantener un alto estándar de calidad a lo largo del desarrollo.

4. Integración y Certificación

Tras la construcción, el equipo avanzó a la fase de integración, donde los diferentes módulos y componentes desarrollados se unieron en un sistema cohesivo.

El proceso de integración incluyó pruebas exhaustivas para asegurar que todas las partes del sistema funcionaran correctamente en conjunto. Asimismo, se llevó a cabo un proceso de certificación donde se verificó que el sistema cumpliera con los estándares y requisitos establecidos tanto a nivel funcional como no funcional.

5. Despliegue y Monitoreo

Finalmente, una vez que el sistema estuvo completamente integrado y certificado, se procedió a su despliegue en el entorno de producción. Se realizó un despliegue incremental, donde las nuevas funcionalidades fueron liberadas de manera gradual, permitiendo al equipo monitorear su desempeño y realizar ajustes según fuera necesario.

Durante el monitoreo post-despliegue, se implementaron herramientas de seguimiento en tiempo real que permitieron al equipo identificar y resolver rápidamente cualquier problema que pudiera surgir en el entorno de producción

A lo largo de todas estas fases, un aspecto distintivo de la metodología fue su enfoque colaborativo y participativo. Desde el inicio hasta el final del proyecto, se promovió un ambiente donde todos los miembros del equipo, independientemente de su rol, tuvieron la oportunidad de contribuir activamente en la toma de decisiones y en la ejecución de tareas clave.

4 Resultados

El primer paso involucró la familiarización con Apache Airflow y la configuración inicial de la plataforma. Este proceso incluyó la preparación de las conexiones necesarias a las bases de datos y otros sistemas involucrados en el proceso ETL, la empresa consiste en múltiples fuentes de datos cómo pueden ser una variedad de archivos (EDI, XML, excel) y bases de datos de distintos esquemas, relacionales cómo mariaDB, postgres y MySQL a la par que no relacionales cómo MongoDB, siguiendo está línea, todas estas fuentes de datos debían de unificarse y tratarse con la confidencialidad requerida en un entorno productivo. Entre ellos, específicamente, se establecieron conexiones con PostgreSQL para la base de datos de etapa (stage), MongoDB para asegurar la consistencia de la información como documento, y RabbitMQ como broker de mensajes. Esta preparación fue crucial para garantizar que el entorno de trabajo estuviera correctamente configurado y listo para soportar el flujo de trabajo automatizado con la carga y rendimiento esperado

Diseño y Desarrollo del Pipeline ETL para subida de pedidos

El diseño del pipeline ETL se basó en la creación de varios DAGs (Directed Acyclic Graphs), que son estructuras clave en Apache Airflow para definir y organizar tareas. Un DAG en Airflow es un conjunto de tareas organizadas en un orden específico, que define cómo deben ejecutarse de manera secuencial o paralela. Cada DAG en este proyecto fue responsable de una etapa específica del proceso, permitiendo una gestión modular y escalable del flujo de trabajo. En otras palabras, sería una traducción de cómo serían los pasos de un flujo de trabajo convencional en una empresa, una serie de pasos a cumplir y las reglas de ejecución de los mismos pero traducido a software en un entorno controlado y con posibilidades de automatización.

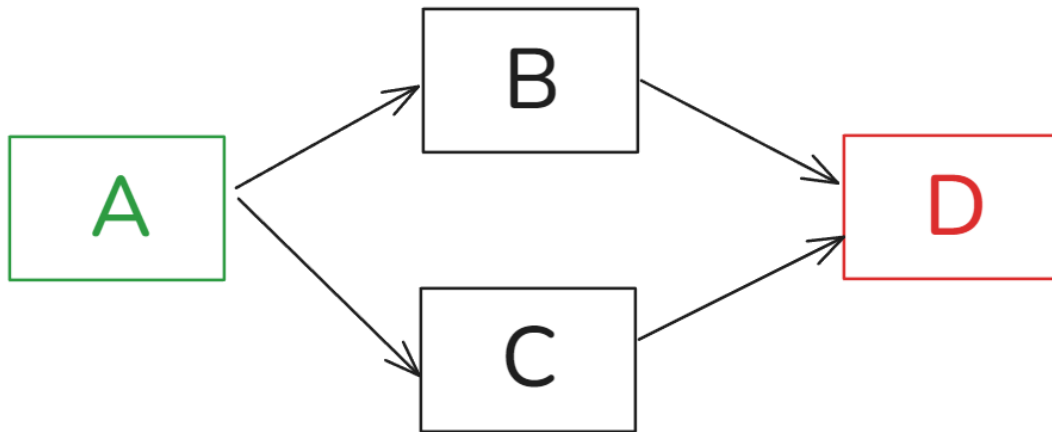


Figura 1: Representación gráfica de un dag

Inicialmente, se empezó por la realización de todo lo relacionado a subir pedidos, actualmente los pedidos llegan en distintos formatos y esta información debe normalizarse y llevarse a un estándar para poder subirla al sistema de manera óptima, para este fin, se desarrolló un DAG que monitorea una carpeta específica donde se suben los archivos EDI, Excel y TXT. Este DAG detecta la llegada de nuevos archivos y los mueve a una carpeta denominada "en proceso". Este paso es fundamental para mantener un control organizado de los archivos que están siendo procesados y evitar inconsistencias en caso de que un mismo archivo pueda correr el riesgo de procesarse varias veces. Este control es necesario debido a que la naturaleza de este paso es que se encuentre en ejecución constante, todos los días a todas horas ya que los pedidos pueden llegar a cualquier hora, por lo que el manejo debe de ser cuidadoso.

Una vez que los archivos se encuentran en la carpeta "en proceso", otro DAG se encarga de la extracción de la información contenida en los archivos. Durante esta fase, los datos son transformados y almacenados en una base de datos transitoria (stage) en PostgreSQL. Este almacenamiento temporal permite realizar una limpieza y transformación inicial de los datos, asegurando que estén en un formato adecuado para su posterior procesamiento, una vez la información ha sido transmitida de un archivo a la base de datos en cuestión, el archivo se mueve a otra carpeta llamada "procesado", de esta forma se crea un ciclo de vida claro y conciso sobre

cómo sería el desplazamiento del archivo entre distintas carpetas manteniendo siempre cómo prioridad la integridad de la información..

Posteriormente, otro DAG extrae la información de la base de datos stage, la convierte a formato JSON y la almacena en MongoDB. La elección de MongoDB se basó en su capacidad para manejar datos no estructurados y semi-estructurados, facilitando así la consistencia de la información como documento. Esta fase es crucial para garantizar que los datos están estructurados de manera coherente y puedan ser fácilmente accesibles para los siguientes pasos del proceso. El caso actual se adapta especialmente bien a la naturaleza no relacional de esta base de datos al tratarse de documentos, un pedido tiene detalles del pedido así como información general, pero ambas son parte de un todo, es absolutamente fundamental que la información se transmita junta, no sirve tener un encabezado sin un detalle y viceversa.

Con los datos almacenados en MongoDB, se da la señal a otro DAG para que extraiga los datos y los inserte en una cola de mensajes de RabbitMQ. RabbitMQ actúa como intermediario, permitiendo que los datos sean gestionados y distribuidos de manera eficiente entre los diferentes componentes del sistema, en este caso el uso que se le dará es para la organización de la información antes de enviarse a un endpoint del ERP de la compañía llamado SAMS.

Finalmente, un daemon desarrollado en Node.js lee los mensajes de RabbitMQ y realiza la subida de la información a un endpoint existente del sistema ERP. Este daemon es responsable de interactuar con la API del ERP, enviando los datos de manera segura y confiable. Dependiendo de la respuesta de la AP.

Manejo de Resultados y Consistencia de Información

Para asegurar la correcta gestión de los pedidos, se implementó un sistema de manejo de colecciones en MongoDB que permite diferenciar los estados de los pedidos. Inicialmente, los datos se almacenan en una colección llamada "EDI", que contiene la información original extraída y almacenada antes de su envío a RabbitMQ. Una vez que los datos son enviados a RabbitMQ, se trasladan a la colección "edi_rabbit", que actúa como un registro de los datos que han sido procesados hasta este punto lo cual servirá como un histórico de los registros que pasaron por la ETL.

Dependiendo del resultado del procesamiento en el sistema ERP, los datos se mueven a diferentes colecciones en MongoDB: Los pedidos que han sido procesados exitosamente se almacenan en la colección "edi_exitoso", mientras que los pedidos que han fallado en su procesamiento, debido a errores del tipo 409 o 500, se almacenan en la colección "edi_fallido". Esta organización permite un seguimiento detallado del estado de cada pedido y facilita la gestión de errores y excepciones al igual que se le trazabilidad a todo el movimiento de los datos.

El proceso descrito se adaptó para manejar tres tipos de formatos de archivos de entrada: EDIFACT, Excel y TXT. Cada tipo de archivo sigue el flujo descrito, con las particularidades necesarias para la correcta extracción y transformación de la información según su formato específico. Esta flexibilidad fue esencial para asegurar que el sistema pudiera integrar de manera efectiva los diferentes tipos de datos recibidos por la empresa.

Además de la implementación del sistema ETL para la gestión de pedidos en formatos EDIFACT, Excel y TXT, se realizó un proceso similar para la extracción y transformación de archivos XML. Este proceso también fue diseñado y orquestado utilizando Apache Airflow, asegurando la consistencia y eficiencia en el manejo de los datos de facturación.

Extracción y Transformación de Archivos XML

El proceso de manejo de archivos XML sigue un flujo de trabajo similar al descrito anteriormente para los archivos EDI, Excel y TXT, con algunas particularidades específicas para el formato XML.

Inicialmente, se desarrolló un DAG en Apache Airflow encargado de monitorear una carpeta específica donde se suben los archivos XML. Este DAG detecta la llegada de nuevos archivos y los mueve a una carpeta denominada "en proceso". Este paso es esencial para mantener un control organizado de los archivos que están siendo procesados y evitar la mezcla de archivos procesados y no procesados.

Una vez que los archivos se encuentran en la carpeta "en proceso", el DAG procede a la extracción de la información contenida en los archivos XML. Durante esta fase, los datos son transformados directamente a un formato JSON. Esta transformación es crítica, ya que el formato JSON es ampliamente utilizado y compatible con diversos sistemas y aplicaciones, facilitando su integración y uso posterior.

Los datos transformados en formato JSON son luego almacenados en MongoDB. En este contexto, los datos almacenados en MongoDB pueden ser utilizados para diversas integraciones contables o cualquier otro uso que se le quiera dar en la empresa.

Este proceso de extracción y transformación de archivos XML es particularmente importante para manejar datos de facturación. Los archivos XML pueden contener información detallada sobre facturas, incluyendo notas de crédito y facturas como tal. La transformación a JSON y el almacenamiento en MongoDB aseguran que estos datos estén disponibles de manera estructurada y accesible para cualquier análisis o integración necesaria. En este caso específico no se realiza la última parte de ETL (Extract, Transform, Load), ya que no se requiere la carga de dichos archivos por el momento, pero se realizó una preparación y adecuación minuciosa de los datos contenidos en dicho formato de archivo, cosa importante a tener en cuenta y en

funcionamiento debido a que la DIAN por ejemplo usa este tipo de archivos para todo lo relacionado a facturación.

Migración de Gastos de Importación

Además de los procesos ETL para la gestión de pedidos y la extracción de archivos XML, se llevó a cabo un proyecto de migración de gastos de importación desde un ERP antiguo a uno nuevo. Este proyecto abordó la necesidad de simplificar y automatizar un proceso que anteriormente implicaba una gran cantidad de inserciones manuales en la base de datos. La implementación de este sistema automatizado no solo redujo significativamente el tiempo y el esfuerzo requeridos, sino que también mejoró la precisión y la consistencia de los datos migrados.

El flujo de trabajo para la migración de gastos de importación se diseñó para que, al recibir un año y mes específicos de gastos a migrar al ERP actual, el sistema realice los siguientes pasos:

1. Consulta y Extracción de Datos:

- Se inicia una consulta a la base de datos del ERP antiguo, que es una base de datos MySQL, para obtener los datos de gastos de importación correspondientes al año y mes especificados.
- Estos datos se extraen y se trasladan a una base de datos PostgreSQL, utilizada específicamente para fines de ETL. En esta base de datos, los datos se almacenan inicialmente con un estado denominado "STAGED".

2. Transformación y Procesamiento:

- Una vez que los datos están en la base de datos PostgreSQL, se les aplica el tratamiento necesario para asegurar que sean compatibles con el nuevo ERP. Esto incluye transformaciones y limpiezas de datos para garantizar su integridad y consistencia.
- Los datos transformados se insertan en la base de datos del nuevo ERP, la cual es MariaDB. Este paso se realiza con un manejo preciso de la transaccionalidad para evitar

inconsistencias, falta de integridad o duplicados. Además, se guarda el “id” del registro insertado en la base de datos destino.

3. Actualización de Estados:

- Después de la inserción en el nuevo ERP, los registros en PostgreSQL se actualizan para reflejar su nuevo estado. Si la inserción fue exitosa, el estado se cambia a "SAVED"; si ocurre algún fallo, el estado se actualiza a "FAILED".

4. Proceso de Aceptación:

- Los gastos migrados, aunque ya están insertados en el nuevo ERP, requieren un proceso de aceptación para ser completamente integrados y activar sus efectos contables y operacionales.
- Este proceso de aceptación se realiza mediante la API del ERP actual (SAMS). Para cada registro migrado, se envían solicitudes al endpoint existente para cambiar su estado, asegurando que los datos sean aprobados y correctamente incorporados en el sistema. Si el proceso de aceptación fue exitoso, el registro se le cambia el estado en la base de datos transitoria a “APPROVED”, en caso de que falle, tendrá en su lugar “REFUSED”.

Al finalizar todo el proceso, se emite un correo electrónico con el resumen de la migración. El proceso se puede ver resumido y reflejado en el siguiente diagrama:

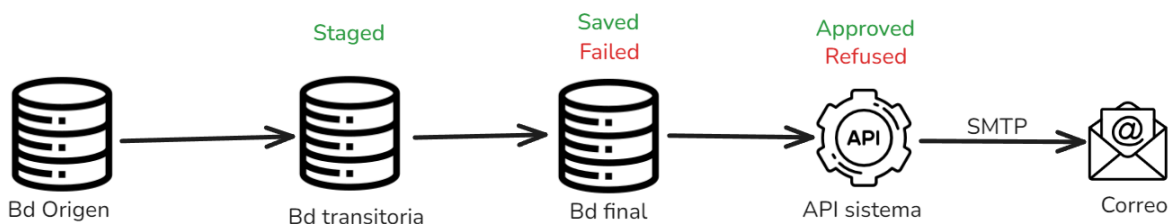


Figura 2: Diseño del proceso migración gastos de importación

5 Análisis

Además de la implementación del sistema ETL y la gestión de pedidos, se lograron otros resultados significativos en el proyecto, entre los cuales se destacan:

Optimización de los Procesos de Inventario:

Se mejoró la eficiencia en la gestión de inventarios al automatizar el flujo de datos desde la recepción de pedidos hasta su inserción en el sistema ERP. Esto redujo significativamente el tiempo requerido para procesar los pedidos y minimizó los errores asociados a la manipulación manual de datos.

Integración de Múltiples Sistemas:

Se logró una integración efectiva entre diferentes sistemas y plataformas, incluyendo PostgreSQL, MongoDB y RabbitMQ, asegurando un flujo de datos coherente y confiable. Esta integración permitió una mejor coordinación entre los distintos departamentos de la empresa y mejoró la calidad de la información disponible para la toma de decisiones.

Mejora de la Trazabilidad:

La implementación de colecciones en MongoDB para diferentes estados de los pedidos permitió una mejor trazabilidad y seguimiento de cada pedido desde su recepción hasta su procesamiento final. Esto facilitó la identificación y resolución de problemas de manera más rápida y eficiente.

Escalabilidad y Flexibilidad:

El uso de Apache Airflow permitió diseñar un sistema escalable y flexible que puede adaptarse a cambios futuros en los requisitos de procesamiento de datos y en el volumen de pedidos. La modularidad de los DAGs facilita la incorporación de nuevas funcionalidades y la modificación de las existentes sin afectar al resto del sistema al igual que una escalabilidad sólida al poder modificar la cantidad de procesos simultáneos que se permiten.

6 Conclusiones

El proyecto centrado en la optimización de la gestión de inventarios a través de la implementación de una nueva ETL ha logrado cumplir de manera exitosa los objetivos planteados, permitiendo una mejora sustancial en la automatización y consistencia del manejo de datos en la empresa. A continuación, se presentan las conclusiones específicas relacionadas directamente con los objetivos establecidos:

- Se diseñó, desarrolló y configuró una ETL que automatiza de manera efectiva la extracción, transformación y carga de datos. Este logro está alineado con el objetivo específico de asegurar la integridad y consistencia de la información en el sistema de gestión de inventarios. La implementación de Apache Airflow fue clave en la creación de flujos de trabajo automatizados que no sólo redujeron el tiempo de procesamiento, sino que también minimizaron la intervención manual, lo que a su vez disminuyó significativamente el margen de error.
- La ETL desarrollada garantizó que los datos provenientes de diversas fuentes (como archivos EDI, Excel, XML, y bases de datos) fueran integrados de manera coherente y precisa en el sistema.
- El uso de Apache Airflow permitió diseñar un sistema ETL escalable y flexible, capaz de adaptarse a futuros cambios en los volúmenes de datos y en los requisitos de procesamiento.
- Se llevaron a cabo pruebas exhaustivas de la ETL para validar su funcionamiento correcto, identificar posibles fallos y garantizar su compatibilidad con los sistemas actuales.
- La implementación de la ETL fue realizada con éxito, corroborando su funcionamiento en un entorno de producción. La capacidad de la ETL para gestionar datos de diversas fuentes y formatos, y su integración con sistemas como MongoDB y RabbitMQ, demostró su eficacia en la automatización y optimización de procesos clave en la gestión de inventarios. Esta implementación ha sentado las bases para futuras expansiones y

mejoras, asegurando que el sistema sea capaz de adaptarse a las necesidades cambiantes de la empresa.

En resumen, el enfoque en la ETL permitió no sólo alcanzar, sino también superar las expectativas en términos de automatización, consistencia, escalabilidad y eficiencia en el manejo de datos. Esto ha resultado en un sistema robusto y adaptable, preparado para afrontar las demandas futuras del entorno logístico de la empresa.

Referencias

Agile Alliance. (2001). Agile Manifesto. Recuperado de <https://agilemanifesto.org/>

Berzal, F. (2013). Desarrollo Profesional de Aplicaciones Web con ASP.NET. Recuperado de <https://elvex.ugr.es/decsai/csharp/pdf/web/web-book-a4.pdf> • Chopra, S., & Meindl, P. (2016). Gestión de operaciones y cadenas de suministro. Pearson.

Elbashir, S. M., Nassar, A. M., & Mana, A. M. (2017). Data warehousing and data mining: concepts, methodologies, tools, and applications. Springer. • Han, Y., Kambhampati, C., & Ram, S. (2011). Data integration: a theoretical perspective. Synthesis Lectures on Data Management.

Kimball, R., & Ross, M. (2002). The data warehouse toolkit: the definitive guide to dimensional modeling. Wiley.

Stauffer, M. (2016). Laravel: Up and running. O'Reilly Media.