GRADO EN CIENCIA DE DATOS

Sistema de Delivery Cross-Cloud Basado en Data Fabric

Presentado por:

Paula de Moya Romero

Dirigido por:

Dr. Ronal Muresano

CURSO ACADÉMICO 2023-2024

Resumen

Este proyecto explora el desarrollo e implementación de un sistema de delivery de datos cross-cloud basado en Data Fabric, diseñado para facilitar la gestión y transferencia de datos entre Azure y AWS. Utilizando una infraestructura de Data Fabric, el sistema ofrece una gestión coherente y unificada de los datos, permitiendo una transferencia eficiente y segura entre diversas plataformas cloud. Se destacan características clave como la automatización de procesos mediante el uso de pipelines y Data Flows en Azure Data Factory, así como la implementación de triggers para iniciar procesos automáticamente cuando se detectan cambios en los datos. La metodología en cascada guio el desarrollo del proyecto, asegurando una implementación sistemática y verificada que cumple con los estándares de seguridad y eficiencia operativa. Los resultados demostraron la capacidad del sistema para adaptarse a distintos entornos de datos y responder dinámicamente a las necesidades empresariales en constante cambio.

Palabras clave

Pipeline, Data Flow, Trigger, Data Mesh, Data Fabric, Data Sharing, ETL.

Abstract

This project explores the development and implementation of a cross-cloud data delivery system based on Data Fabric**,** designed to facilitate data management and transfer between Azure and AWS**.** Using a Data Fabric infrastructure, the system provides consistent and unified data management, enabling efficient and secure transfer across various cloud platforms. Key features include process automation using pipelines and Data Flows in Azure Data Factory**,** as well as trigger implementation to automatically initiate processes when data changes are detected. The cascade methodology guided the project's development, ensuring a systematic and verified implementation that adheres to security and operational efficiency standards. Results demonstrated the system's ability to adapt to different data environments and dynamically respond to ever-changing business needs.

Keywords

Pipeline, Data Flow, Trigger, Data Mesh, Data Fabric, Data Sharing, ETL.

CAPÍTULO 1: INTRODUCCIÓN

1.1 Introducción

En el contexto tecnológico actual, la migración de datos hacia la nube se ha convertido en un imperativo para las empresas que buscan optimizar sus operaciones y aprovechar el potencial de sus datos. La nube ofrece una serie de ventajas, desde una escalabilidad prácticamente ilimitada hasta una mayor flexibilidad operativa, lo que la convierte en un entorno ideal para el desarrollo de equipos de datos ágiles y eficientes.

Además de estos beneficios, la seguridad se ha vuelto crucial para las empresas que migran a la nube. Según Satya Nadella, CEO de Microsoft, "Las empresas de hoy y del mañana exigen una plataforma en la nube que sea confiable, escalable y flexible" (Satya Nadella: “Microsoft Ofrece La Nube Más Completa De La Industria” | Diario TI, s.f.). Las principales plataformas en la nube, como Azure y AWS, ofrecen robustas medidas de seguridad, proporcionando tranquilidad a las empresas respecto a la protección de sus datos.

Con la creciente adopción de la nube por parte de las empresas y el gran volumen de datos que gestionan las empresas hoy en día, surge la necesidad de gestionar eficientemente los datos dentro de las organizaciones. En este contexto, han surgido dos nuevas arquitecturas de gestión de datos: Data Mesh y Data Fabric. Mientras que el Data Mesh propone una gestión descentralizada de datos en las empresas, donde cada equipo es responsable de sus datos, fomentando asi la autonomía y la colaboración entre equipos, el Data Fabric ofrece una infraestructura unificada para gestionar datos sin importar su ubicación o formato, facilitando así el intercambio eficiente entre equipos y departamentos.

A medida que el contexto tecnológico actual junto con estas medidas que empujan a las empresas a avanzar hacia una mayor colaboración y asociación entre sí surge la necesidad de compartir datos de manera segura y eficiente y con ella el concepto de Data Sharing. Este proceso se vuelve aún más complejo con la aparición de conceptos como contratos inteligentes y el intercambio de datos entre distintos departamentos de una empresa.

Así surge la idea de desarrollar un sistema de entrega de datos entre plataformas de la nube que siga la arquitectura de Data Fabric. Este sistema se centrará en utilizar las principales plataformas de la nube, aprovechando sus beneficios en términos de flexibilidad y seguridad. El objetivo es garantizar un intercambio seguro y eficiente de datos entre diferentes plataformas en la nube, sin comprometer la coherencia ni la integridad de estos.

El proyecto estará dividido en cinco capítulos. En este capítulo inicial, se proporcionará una visión general del trabajo a realizar, incluyendo una descripción del problema actual, su justificación y los objetivos que se pretenden alcanzar. A continuación, se explorará el marco teórico, situando el proyecto dentro del contexto de la ciencia de datos y presentando los conceptos clave pertinentes. Luego, se detallará el método utilizado para el desarrollo del proyecto, seguido de la implementación de la metodología, donde se describirá paso a paso la ejecución de cada fase. Finalmente, se presentarán las conclusiones del estudio, acompañadas de recomendaciones para maximizar el rendimiento del sistema. Estas secciones proporcionarán una comprensión completa del problema abordado, el contexto teórico relevante, el enfoque metodológico empleado y las conclusiones derivadas del estudio.

1.2 Planteamiento del problema

En la actualidad digital, las pequeñas y medianas empresas (PYMEs) se enfrentan al desafío de gestionar eficientemente grandes volúmenes de datos en entornos de nube. La migración generalizada hacia la nube y la creciente adopción de flujos de trabajo en este entorno han generado una demanda sin precedentes de soluciones efectivas de gestión de datos. Sin embargo, las PYMEs a menudo carecen de los recursos y la experiencia técnica necesarios para implementar estos sistemas de manera efectiva. Además, la seguridad de los datos es una preocupación crítica, lo que requiere que cualquier sistema desarrollado garantice la seguridad y la integridad de los datos durante su transferencia.

Estos desafíos pueden tener un impacto significativo en las PYMEs en términos de costos, eficiencia y seguridad. La falta de recursos y experiencia puede resultar en costos elevados y en una implementación ineficiente de los sistemas de gestión de datos. Además, en un entorno empresarial cada vez más interconectado, surge la necesidad de facilitar el intercambio de información confiable dentro de la organización o incluso entre

empresas a través de asociaciones. Si bien las asociaciones entre empresas pueden fomentar una colaboración más efectiva y una toma de decisiones más informada, requieren un intercambio de datos seguro y eficiente.

El concepto de Data Sharing ha surgido como una práctica esencial para impulsar la colaboración y la innovación en el ámbito empresarial. Esto se potencia con el surgimiento de los contratos inteligentes, que automatizan la negociación o el cumplimiento de un contrato. Por lo tanto, las empresas necesitan compartir datos de manera segura y controlada entre diferentes entidades para obtener valiosos insights y tomar decisiones informadas.

La solución propuesta a este problema consiste en desarrollar un mecanismo basado en las principales plataformas de la nube, lo que garantiza de por sí la seguridad y flexibilidad necesarias para que las PYMEs puedan transferir datos entre plataformas en la nube de manera eficiente y segura, todo ello respaldado por la arquitectura Data Fabric, que proporciona una infraestructura de datos unificada, independientemente de la ubicación o el formato de los datos. Esto facilita el Data Sharing, ya que los datos pueden fluir de manera eficiente a través de la infraestructura

Este enfoque permite a las PYMEs enfocarse en su negocio principal en lugar de preocuparse por la gestión de datos, dado que el sistema será fácil de implementar, escalable y adaptable a las cambiantes necesidades del entorno empresarial, proporcionando una solución sólida y confiable para las necesidades actuales de las organizaciones.

1.3 Objetivo general

El objetivo general de este proyecto es desarrollar un sistema de delivery de datos entre plataformas en la nube basado en las principales plataformas de la nube (Azure y AWS) que permita a las PYMEs y a las asociaciones entre empresas transferir datos de manera eficiente y segura, promoviendo la colaboración efectiva y la toma de decisiones informada.

Además, busca hacer una contribución significativa a varios Objetivos de Desarrollo Sostenible (ODS). Por ejemplo, contribuye al ODS 9 (Industria, innovación e infraestructuras) al fomentar la innovación y el desarrollo de infraestructuras resilientes a través de la creación de un sistema innovador para la gestión y transferencia de datos. También apoya el ODS 12 (Producción y consumo responsables) al promover la efi-

ciencia en la producción y el consumo de datos, lo que puede resultar en prácticas más sostenibles. Finalmente, al facilitar el intercambio de información entre diferentes plataformas en la nube, el proyecto fomenta la colaboración y las alianzas estratégicas, que son esenciales para alcanzar el ODS 17 (Alianza para lograr los objetivos).

Ilustración 1: Representación gráfica de tres Objetivos de Desarrollo Sostenible.

1.4 Objetivos específicos

1. Un sistema automatizado y confiable para la entrega segura y eficiente de datos entre los servicios de almacenamiento de la nube desde Azure Blob Storage hasta AWS S3.
2. Diseñar un Data Flow (Flujo de trabajo) utilizando Azure Data Factory, que realice el proceso ETL (extracción, transformación y envió).
3. Configurar un sistema de notificaciones utilizando Azure Logic Apps que se active automáticamente al ejecutarse un pipeline para informar a los usuarios sobre el estado y eventos importantes del proceso de entrega de datos.
4. Diseñar un pipeline (canalización) en Azure Data Factory que incluya el Data Flow y el sistema de notificaciones.
5. Configurar un mecanismo de automatización mediante un trigger (disparador) para ejecutar el pipeline de entrega de datos de manera programada y sin intervención manual al insertar datos en la o las ubicaciones especificadas.
6. Aunque el sistema desarrollado en este trabajo sea para unas características determinadas, el sistema debe ser configurable para adaptarse a las necesidades específicas del negocio. Esto incluye la capacidad de personalizar los flujos de trabajo lo que incluye los formatos en los que se encuentran los datos, las

ubicaciones de almacenamiento y la ubicación de envió para seguir los principios de una arquitectura basada en Data Fabric.

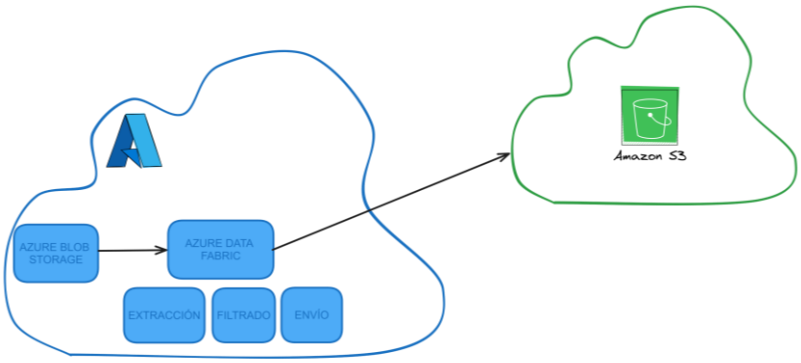


Ilustración 2: Esquema general del sistema de delivery cross-cloud basado en Data Fabric

1.5 Justificación

El proyecto se justifica en la creciente necesidad de las organizaciones, especialmente las pequeñas y medianas empresas (PYMEs), de gestionar y transferir datos de manera eficiente y segura entre diferentes plataformas en la nube. En la era digital actual, la gestión de datos se ha convertido en un desafío crítico, especialmente para las PYMEs que a menudo carecen de los recursos y la experiencia técnica necesarios para implementar sistemas efectivos de gestión de datos. Como señaló Peter Sondergaard “La información es el petróleo del siglo XXI, y la analítica es el motor de combustión” (Motor Adcreative ML/AI, s.f.) resaltando así el valor crucial de los datos en nuestra sociedad actual y la importancia de gestionarlos eficientemente.

El desarrollo de un sistema de entrega de datos entre plataformas en la nube, basado en un enfoque de Data Fabric, responde directamente a esta urgente necesidad. Al automatizar el proceso de extracción, transformación y envío de datos, este sistema permite a las organizaciones centrarse en sus operaciones principales, liberándolas de tareas tediosas relacionadas con la gestión de datos. Además, con la creciente incorporación de conceptos innovadores como Data Sharing y los contratos inteligentes, se refuerza aún más la relevancia de este proyecto al adaptarse y complementarse con las últimas tendencias tecnológicas y empresariales.

La interconexión de Azure con Amazon Web Services (AWS) demuestra la versatilidad y escalabilidad del sistema propuesto, dos aspectos fundamentales en un contexto donde las organizaciones buscan adoptar cada vez más estrategias híbridas y de múltiples nubes. Esta capacidad de interoperabilidad entre diferentes plataformas en la nube no solo amplía las opciones disponibles para las organizaciones, sino que también facilita la adopción de tecnologías avanzadas en la gestión de datos, asegurando su eficiencia y seguridad.

Además, cada vez más datos confidenciales se trasladan a la nube. Según un informe de IT Digital Media Group "tres cuartas partes (75%) de las empresas encuestadas señalaron que más del 40% de los datos almacenados en la nube se clasifican como confidenciales, en comparación con el 49% del año pasado" (IT Digital Media Group, 2023). Esta estadística resalta la importancia crítica de gestionar los datos de manera segura y eficiente para proteger la información sensible de la empresa.

En resumen, este proyecto se justifica por su potencial para revolucionar la forma en que las empresas gestionan grandes volúmenes de datos en la nube. Al proporcionar una solución práctica y efectiva, las empresas pueden liberar recursos para centrarse en otras tareas, aligerando así su carga de trabajo. Además, este proyecto garantiza un intercambio seguro de información, aspectos esenciales cuando las empresas establecen asociaciones. Esto mejora la toma de decisiones y los resultados a nivel organizacional.

Según Eric Schmidt "Desde el amanecer de la civilización hasta el 2003, la humanidad generó 5 exabytes de data. Ahora producimos esa cantidad de data cada dos días y el ritmo de crecimiento continúa acelerándose" (¿Qué Es En Verdad Big Data? Y Por Qué Está Cambiando El Mundo, 2014a). Esta declaración subraya la creciente cantidad de datos generados en la actualidad y la necesidad de sistemas eficientes para gestionarlos. Al proporcionar soluciones prácticas y efectivas para las necesidades actuales de gestión de datos en la nube, este proyecto tiene el potencial de contribuir significativamente al avance y la innovación en este campo.

CAPÍTULO 2: MARCO TEÓRICO

2.1 Entornos cloud

La computación en la nube ha cambiado la forma en que las empresas manejan sus datos. Además de ofrecer más potencia de cómputo y espacio de almacenamiento, permite usar servicios preconfigurados, sin necesidad de programar desde cero. Plataformas como AWS, Azure o Google Cloud ofrecen una variedad de estos servicios, como bases de datos y análisis de datos, lo que acelera el desarrollo y reduce la carga de trabajo.

Líderes del Mercado: Azure y AWS

Microsoft Azure y Amazon Web Services (AWS) son dos de las principales opciones en la nube, con una amplia gama de servicios adaptados a diferentes necesidades empresariales:

- **Infraestructura como Servicio (IaaS):** Permite a las empresas alquilar infraestructura física virtualizada, tales como servidores y almacenamiento de datos, proporcionando el control completo sobre el hardware virtualizado. Esto elimina la necesidad de invertir en y mantener hardware físico, mientras ofrece la flexibilidad de configurar el entorno según las necesidades específicas de la empresa.

- **Plataforma como Servicio (PaaS):** Ofrece un ambiente de desarrollo y despliegue que facilita a los desarrolladores construir aplicaciones y servicios sin tener que gestionar la infraestructura subyacente. Este modelo es ideal para desarrolladores que desean concentrarse en la creación de software sin preocuparse por el mantenimiento del sistema operativo, actualizaciones de software y configuraciones de seguridad, ya que la responsabilidad de todo tipo de pérdida de datos recae sobre la nube.

- **Software como Servicio (SaaS):** Proporciona a los usuarios acceso a aplicaciones de software completas a través de Internet, a menudo mediante un modelo de suscripción. Este servicio elimina la necesidad de instalar y ejecutar aplicaciones en computadoras individuales, facilitando el mantenimiento y la actualización centralizada.

En el contexto del proyecto, tanto Azure Blob Storage como AWS S3 juegan roles cruciales por su capacidad de proporcionar un almacenamiento de datos escalable, duradero y accesible. Estos servicios están optimizados para manejar enormes volúmenes de datos con cualquier tipo de formato, ya sean datos estructurados (CSV, Excel, JSON…) o no estructurados, tales como imágenes, vídeos, o grandes bases de datos, lo que es esencial para la recolección y análisis de datos distribuidos en gran escala.

- **Azure Blob Storage:** Ofrece soluciones optimizadas para el almacenamiento de grandes cantidades de datos. Su capacidad para integrarse con Azure Data Factory, por ejemplo, permite la creación de flujos de trabajo de datos complejos y escalables que pueden automatizar la transformación y el análisis de grandes volúmenes de datos de manera personalizada según las necesidades del negocio y eliminando casi por completo la necesidad de código.

- **AWS S3:** Se destaca por su durabilidad, disponibilidad y escalabilidad. La integración con otras soluciones de AWS, como AWS Lambda y AWS Glue, permite que los datos almacenados en S3 se procesen y analicen en tiempo real o mediante procesos batch, facilitando así flujos de trabajo de análisis de datos altamente eficientes y escalables.

La elección de estos servicios cloud no solo refleja una preferencia por soluciones robustas y escalables, sino también una estrategia para maximizar la eficiencia operativa y la flexibilidad en el manejo de grandes conjuntos de datos. La capacidad de ambos entornos para integrarse con herramientas de análisis avanzadas y manejar picos de demanda de manera eficiente sin necesidad de intervención manual es esencial para proyectos que requieren una gran capacidad de análisis de datos y respuesta rápida a las necesidades cambiantes del mercado.

Además, estos servicios no solo son potentes por sí solos, sino que también pueden interconectarse entre sí y con otros servicios en la nube para crear soluciones más completas y personalizadas, lo que brinda flexibilidad y capacidad de escala a las empresas.

2.2 Tipos de ficheros

En la ciencia de datos, la distinción entre datos estructurados y no estructurados es fundamental para entender cómo manejar y analizar diferentes tipos de información

Datos estructurados

Los datos estructurados son aquellos que tienen un formato definido y organizado, lo que facilita su almacenamiento, búsqueda y análisis. Estos datos suelen estar contenidos en bases de datos relacionales o en formatos de archivo que siguen un esquema rígido (filas y columnas), como los archivos CSV o Excel.

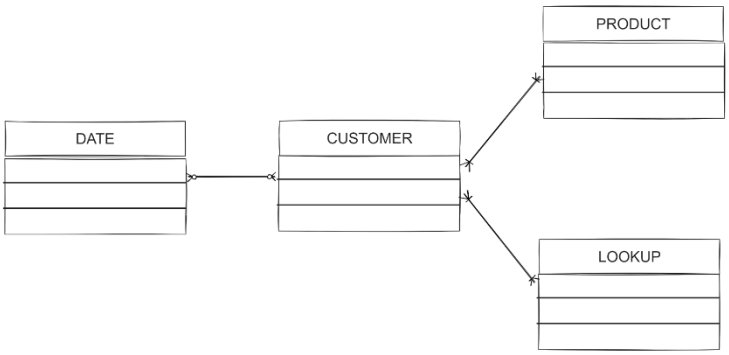


Ilustración 3: Esquema base de datos relacionales

- **CSV:** El formato CSV es perfecto para representar datos tabulares simples, donde la información se organiza en filas y columnas, con cada fila representando un registro y cada columna un campo específico. Al separar los valores con comas u otro delimitador, se crea una estructura que facilita la lectura y el procesamiento de los datos por parte de programas y herramientas de análisis.

  Este tipo de archivos son muy utilizados en la ciencia de datos debido a su estructura sencilla y fácil manipulación. Son compatibles con una amplia gama de herramientas de análisis, desde simples hojas de cálculo hasta sofisticadas plataformas de análisis de big data. La simplicidad del formato CSV permite a los analistas y científicos de datos realizar rápidamente importaciones y exportaciones de grandes volúmenes de datos sin necesidad de procesamiento complejo, lo que facilita la interoperabilidad entre diferentes sistemas y plataformas.

ID,Name,Color,Date
3,Paula,Red,03-12-2000
5,Antoni,Blue,11-10-1999
8,Fernando,Pink,15-12-1998
10,Victoria,Yellow,15-11-2001

Ilustración 4: Ejemplo visualización CSV

- **Excel:** Similar al CSV en términos de manejo de datos tabulares, pero con capacidades adicionales para la manipulación y visualización de datos a través de funciones y gráficos, permite a los usuarios manipular y visualizar datos rápidamente, ofreciendo una amplia gama de herramientas analíticas y estadísticas integradas.

ID	Product	Price
1532	Car	40.000
2457	Computer	1.000
1365	Phone	500
7239	Book	14

Ilustración 5: Ejemplo visualización Excel

Datos no estructurados

Los datos no estructurados no siguen un modelo o formato definido, lo que complica su procesamiento y análisis con herramientas convencionales. Estos datos incluyen texto, imágenes, vídeos y otros tipos de contenido que no encajan fácilmente en tablas o bases de datos relacionales. Formatos como JSON y XML, aunque estructurados en su formato, son flexibles en la organización de los datos y pueden adaptarse a estructuras de datos más complejas y menos rígidas.

- **JSON:** Ampliamente empleado en aplicaciones web, facilita el intercambio dinámico de datos. Su versatilidad permite representar estructuras complejas y anidadas, comunes en muchas aplicaciones modernas. Dada su compatibilidad intrínseca con JavaScript, es muy utilizado en APIs y servicios web. Además, su facilidad de consumo por aplicaciones cliente, incluyendo navegadores y aplicaciones móviles, lo convierte en una opción popular para el intercambio de datos en entornos web.

```
{
  "name": "John",
  "age": 30,
  "city": "New York",
  "interests": ["programming", "traveling", "reading"],
  "contact": {
    "email": "john@example.com",
    "phone": "555-123-4567"
  }
}
```

- **XML:** Similar a JSON en su capacidad para manejar datos complejos, ofrece una estructura que permite definiciones detalladas y validaciones estrictas. Este formato es especializado para Big Data, especialmente útil en entornos que demandan una validación rigurosa de datos y donde las especificaciones de los documentos deben seguirse estrictamente, como en transacciones financieras o en la comunicación entre diferentes sistemas empresariales.

<person>
<name>John</name>
<age>30</age>
<city>New York</city>
<interests>
<interest>programming</interest>
<interest>traveling</interest>
</interests>
<contact>
<email>john@example.com</email>
<phone>555-123-4567</phone>
</contact>
</person>

- **Parquet:** Es un formato de archivo columnar diseñado para almacenar y consultar grandes volúmenes de datos de manera eficiente. Su estructura columnar optimizada permite un acceso rápido a columnas específicas de datos, lo que resulta en una mejora significativa en el rendimiento, especialmente en consultas que no necesitan acceder a todas las columnas del conjunto de datos. Integrado con herramientas de procesamiento de big data como Apache Hadoop y Apache Spark, Parquet ofrece compresión y codificación eficientes, lo que lo convierte en una opción ideal para entornos de big data.

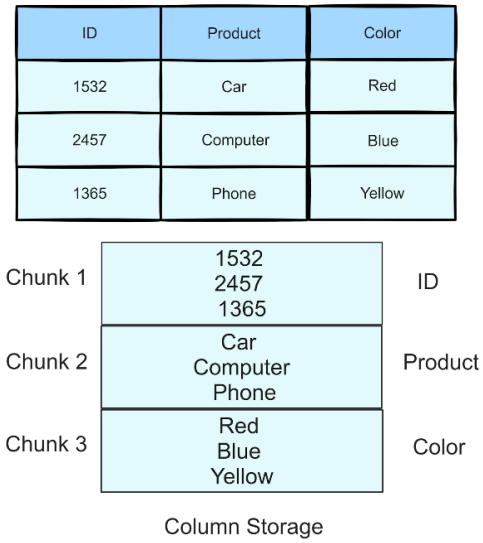


Ilustración 6: Ejemplo visualización Parquet

2.3 Entornos de procesamiento de datos

Los entornos de procesamiento de datos son esenciales en la ciencia de datos y en la mayoría de las industrias modernas. Estos entornos implican una combinación de herramientas, tecnologías y metodologías que permiten a los científicos de datos recopilar, almacenar, procesar y analizar grandes volúmenes de datos para transformarlos en información útil.

Un entorno de procesamiento de datos se refiere a la infraestructura y las herramientas utilizadas para recopilar, procesar y analizar datos. Esto puede incluir hardware, como servidores y dispositivos de almacenamiento, software, como sistemas operativos y aplicaciones de bases de datos, y servicios, como la nube y otras plataformas de procesamiento de datos. Estos componentes trabajan juntos para extraer de los datos en bruto información útil que puede ser utilizada para tomar decisiones informadas. Sin entornos de procesamiento de datos adecuados, sería extremadamente difícil, si no imposible, trabajar con grandes volúmenes de datos de manera efectiva.

2.3.1 Herramientas de Procesamiento de Datos

Las herramientas de procesamiento de datos son esenciales para manejar y analizar grandes volúmenes de datos. Estas herramientas permiten a los científicos de datos y a otros profesionales recopilar, almacenar, procesar y analizar datos de manera eficiente. Las herramientas de procesamiento de datos pueden variar desde software de bases de datos hasta lenguajes de programación y plataformas de análisis de datos.

Bases de Datos

Las bases de datos son una parte integral de cualquier entorno de procesamiento de datos. Proporcionan un medio para almacenar, recuperar y manipular datos. Hay dos tipos principales de sistemas de gestión de bases de datos (DBMS): SQL y NoSQL.

- **SQL (Structured Query Language):** Los DBMS SQL, como MySQL, Oracle y SQL Server, utilizan un lenguaje de consulta estructurado para interactuar con los datos. Estos sistemas son excelentes para manejar datos estructurados y proporcionan potentes capacidades de consulta.

- **NoSQL (Not Only SQL):** Los DBMS NoSQL, como MongoDB, Cassandra y CouchDB, son útiles para manejar datos no estructurados o semi-estructurados.

Estos sistemas son altamente escalables y ofrecen flexibilidad en términos de esquemas de datos.

La elección entre SQL y NoSQL dependerá de los requisitos específicos del proyecto. Por ejemplo, si se necesita realizar consultas complejas en datos estructurados, un DBMS SQL podría ser la mejor opción. Por otro lado, si se está trabajando con grandes volúmenes de datos no estructurados, un DBMS NoSQL podría ser más adecuado.

Tecnologías de procesamiento de datos

Entre las herramientas de procesamiento de datos se encuentran varias tecnologías que son esenciales para manejar grandes volúmenes de datos. Entre las más populares se encuentran Apache Hadoop y Apache Spark:

Apache Hadoop es una plataforma de software de código abierto que permite el procesamiento distribuido de grandes conjuntos de datos a través de clústeres de computadoras utilizando modelos de programación simples. Es fundamental en proyectos que requieren el manejo de grandes volúmenes de datos no estructurados o semi-estructurados, como registros de servidores, transacciones de comercio electrónico y datos de redes sociales.

Hadoop se basa en dos componentes principales:

- **Hadoop Distributed File System (HDFS)**: Un sistema de archivos que proporciona almacenamiento de datos distribuido y tolerante a fallos. HDFS descompone los datos en bloques y los distribuye en todo el clúster, lo que permite un procesamiento eficiente y altamente disponible.

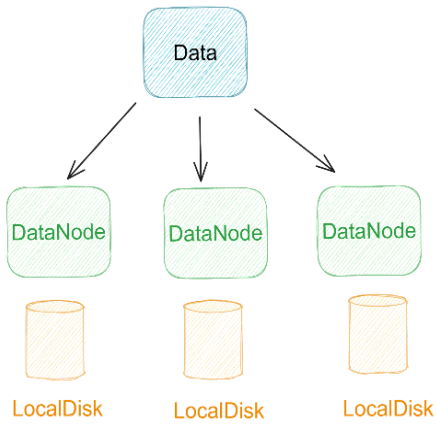


Ilustración 7: Esquema explicativo HDFS

- **MapReduce**: Un modelo de procesamiento que divide las tareas en pequeñas partes, cada una de las cuales puede ser ejecutada o reejecutada en cualquier nodo del clúster. Facilita el manejo de grandes volúmenes de datos, ya que cada nodo del clúster procesa una pequeña parte de los datos de manera paralela.

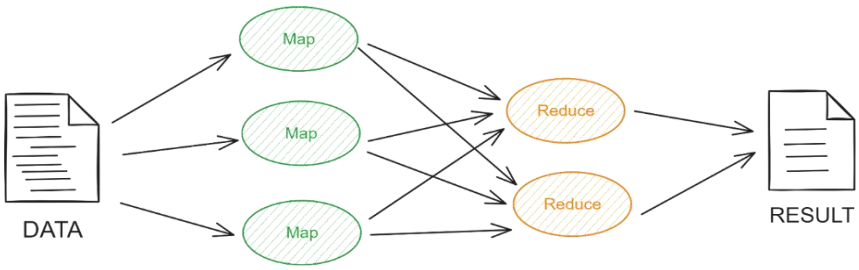


Ilustración 8: Esquema explicativo de MapReduce

Apache Spark es un motor de procesamiento de datos de código abierto que se destaca por su capacidad para realizar análisis avanzados y procesamiento en tiempo real. Con habilidades para manejar datos por lotes y en memoria, brinda una alta velocidad y eficiencia, especialmente para grandes volúmenes de datos. Además, funciona como un motor multi-lenguaje, lo que significa que puede ser utilizado con varios lenguajes de programación, incluyendo Python, SQL y R. Esta característica lo hace accesible para una amplia gama de profesionales, desde ingenieros de datos hasta científicos de datos y analistas. Spark también cuenta con capacidades avanzadas para el aprendizaje automático, lo que permite a los usuarios construir y entrenar modelos predictivos y de clasificación utilizando herramientas y bibliotecas populares.

En resumen, la capacidad de procesar y analizar grandes volúmenes de datos de manera eficiente en entornos distribuidos es fundamental para el éxito de los proyectos modernos de ciencia de datos y machine learning. Además, gracias a la integración de Apache Hadoop y Apache Spark en plataformas cloud como AWS y Azure permite a las organizaciones desplegar soluciones de big data altamente escalables, robustas y coste-eficientes que son esenciales para la obtención de insights oportunos y precisos a partir de sus datos.

Lenguajes de programación

Los lenguajes de programación desempeñan un papel fundamental en el procesamiento y análisis de datos, con Python, R y SQL destacándose como herramientas esenciales en la ciencia de datos:

- **Python:** Este lenguaje de programación de alto nivel es ampliamente preferido en la ciencia de datos debido a su simplicidad y su robusta comunidad de desarrolladores. Python ofrece una sintaxis clara y legible, lo que facilita el desarrollo y la depuración de código. Además, cuenta con una amplia gama de bibliotecas especializadas en análisis de datos, como Pandas, NumPy y SciPy, que ofrecen herramientas poderosas para la manipulación, visualización y modelado de datos.

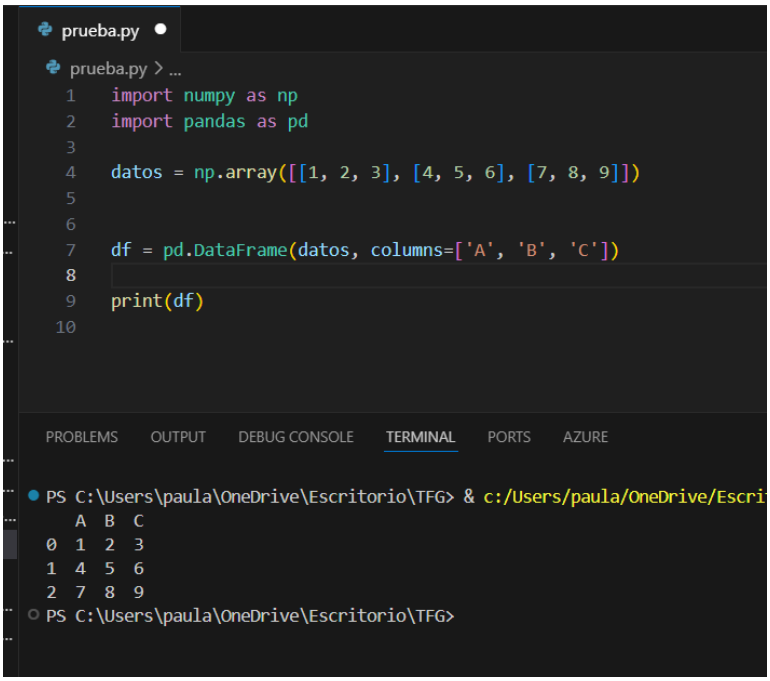


Ilustración 9: Ejemplo Python

- **R:** Es tanto un lenguaje de programación como un entorno de software diseñado específicamente para el análisis estadístico y la visualización de datos. R es muy valorado entre los estadísticos y los científicos de datos por su riqueza en funciones estadísticas y su extensa colección de paquetes especializados. Además, R ofrece herramientas avanzadas de visualización de datos que permiten crear

gráficos complejos y personalizados para explorar y comunicar patrones en los datos.

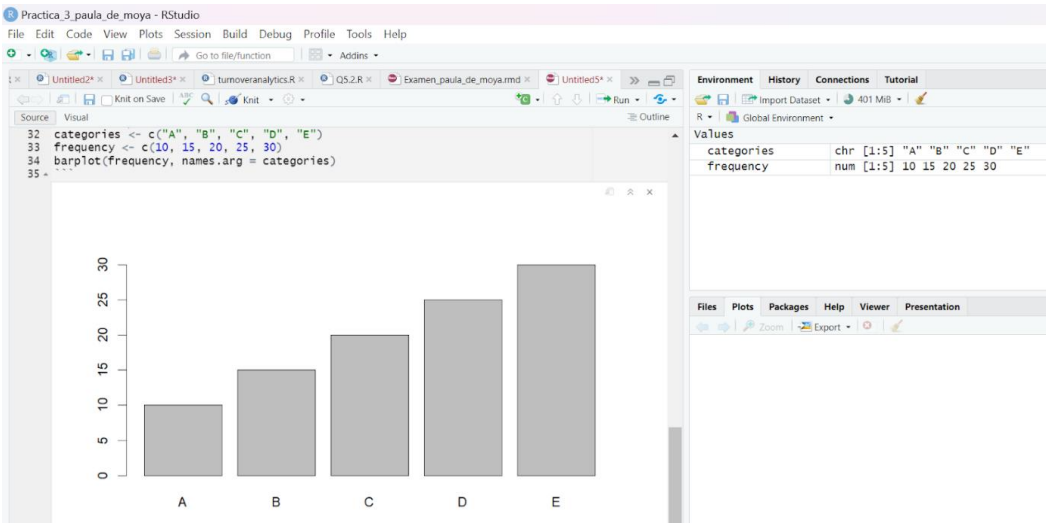


Ilustración 10: Ejemplo R

- **SQL:** El lenguaje estructurado de consulta (SQL) es esencial para la manipulación de bases de datos relacionales, lo que lo convierte en un componente crítico en el arsenal de herramientas de un científico de datos. SQL se utiliza para consultar, recuperar y manipular datos almacenados en bases de datos relacionales, lo que lo hace indispensable para tareas como la limpieza de datos, la exploración de datos y la extracción de características. Su sintaxis intuitiva y su capacidad para realizar consultas complejas lo convierten en una herramienta poderosa para la gestión de datos en entornos de ciencia de datos.

```
SELECT product, name
FROM Customer
WHERE name = 'Paula';
```

product	name
Car	Paula
Book	Paula
Computer	Paula

Ilustración 11: Ejemplo SQL

2.3.2 Metodologías

Las metodologías de procesamiento de datos se refieren a los enfoques y técnicas utilizadas para manejar y analizar datos. Estas metodologías proporcionan un marco estructurado para transformar los datos en bruto en información útil:

- **Recopilación de Datos:** En esta etapa, se identifican y recogen los datos necesarios para el análisis desde diversas fuentes, como bases de datos, archivos, APIs y servicios web.

- **Preprocesamiento de Datos:** Una vez recopilados, los datos pasan por un proceso de limpieza y transformación para prepararlos para su análisis. Esto puede incluir la eliminación de datos duplicados o irrelevantes, la corrección de errores, la gestión de datos faltantes y asegurarse de que cada dato se encuentre en su formato correspondiente.

- **Análisis de Datos:** Es el corazón del procesamiento de datos, donde se utilizan técnicas estadísticas y algoritmos de aprendizaje automático para descubrir patrones y tendencias en los datos. Incluye la exploración de datos, la construcción de modelos predictivos y la evaluación de estos.

- **Visualización de Datos:** Esta etapa implica la creación de gráficos y tablas para representar visualmente los resultados del análisis de datos. Ayuda a entender mejor los patrones y tendencias en los datos y a comunicar los resultados del análisis de manera efectiva.

- **Toma de Decisiones:** Utiliza la información obtenida del análisis de datos para tomar decisiones informadas. Puede implicar la implementación de cambios en políticas o estrategias, o la toma de decisiones sobre futuros análisis de datos.

Cada una de estas etapas es crucial para el procesamiento eficaz y eficiente de los datos. Sin embargo, no son necesariamente lineales y pueden variar según las necesidades específicas del proyecto de ciencia de datos.

2.4 Data Mesh Vs Data Fabric

2.4.1 Data Mesh

El Data Mesh es una forma diferente de manejar y compartir datos. En lugar de tener un solo equipo que controle todos los datos, el Data Mesh asigna a diferentes equipos la responsabilidad de sus propios conjuntos de datos. Los datos son considerados como productos independientes, cada uno con su propio propietario y equipo responsable. En lugar de depender de un único equipo de ingeniería de datos para gestionar y proporcionar acceso a los datos, diferentes equipos en la organización son designados como propietarios de sus propios conjuntos de datos.

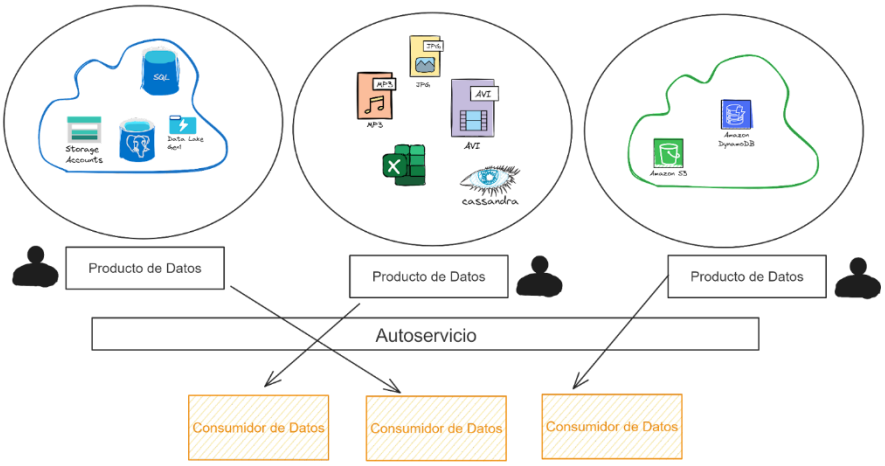


Ilustración 12: Arquitectura Data Mesh

Servicios del Data Mesh

Los principales servicios del Data Mesh son:

- **Creación de productos de datos:** Cada conjunto de datos se trata como un producto independiente, con su propio dueño y reglas para usarlo. Por ejemplo, el equipo de salud puede ofrecer un producto de datos sobre el historial de pacientes, mientras que el equipo de finanzas ofrece un producto de datos sobre transacciones financieras.
- **Cambio de roles:** Hay personas responsables de cuidar los datos (los dueños) y otras que los usan para hacer análisis o construir aplicaciones (los consumidores).
- **Acceso a datos:** Facilita que los equipos compartan datos entre sí de manera eficiente y segura. Por ejemplo, el equipo de salud puede acceder a los datos financieros relevantes para mejorar la precisión de sus aplicaciones.
- **Infraestructura de autoservicio:** Cada equipo tiene acceso a las herramientas y recursos necesarios para acceder y utilizar los datos por sí mismos. Esto significa que no tienen que depender de otros equipos para obtener los datos que necesitan para su trabajo.

Beneficios del Data Mesh

- **Mejora de la eficiencia:** Al dividir la responsabilidad de los datos entre varios equipos, se simplifica el proceso y se hace más rápido. Los equipos pueden trabajar de manera independiente y no tienen que esperar a que otros les proporcionen los datos que necesitan.
- **Mejora de la calidad de los datos:** Al tener cada equipo la responsabilidad de sus propios datos, están más familiarizados con ellos y pueden mantenerlos mejor. Esto significa que los datos tienden a ser más precisos y actualizados.
- **Promoción de la colaboración:** Facilita que los equipos trabajen juntos y compartan datos de manera más fácil. Al tener acceso a los datos de otros equipos, pueden colaborar en proyectos interdisciplinarios de manera más efectiva.
- **Mayor agilidad:** Al permitir que los equipos accedan y usen los datos por sí mismos, se pueden tomar decisiones más rápidas y adaptarse mejor a los cambios. Los equipos pueden iterar más rápidamente en el desarrollo de aplicaciones y mejorar la experiencia del usuario de manera más eficiente.

2.4.2 Data Fabric

Data Fabric es una arquitectura de datos que proporciona una forma unificada y coherente de acceder y gestionar los datos que están distribuidos en diferentes ubicaciones y formatos. Su objetivo es facilitar el acceso a los datos, independientemente de dónde se almacenen o cómo se generen.

La arquitectura de Data Fabric se basa en la idea de que los datos deben ser accesibles y útiles, independientemente de su ubicación, formato o aplicación. Esto significa que puede recoger datos de diversas fuentes, organizarlos de manera inteligente y hacerlos accesibles para los consumidores de datos de manera eficiente y segura.

Además, Data Fabric es flexible y escalable, lo que significa que puede adaptarse a diferentes entornos y crecer con las necesidades de la organización. Puede funcionar en diferentes lugares, como en la nube o en diferentes computadoras, lo que lo hace muy adaptable a diferentes necesidades y escenarios.

Lo que la hace una solución integral para la gestión de datos que permite a las organizaciones aprovechar al máximo sus datos, independientemente de dónde se encuentren o cómo se generen.

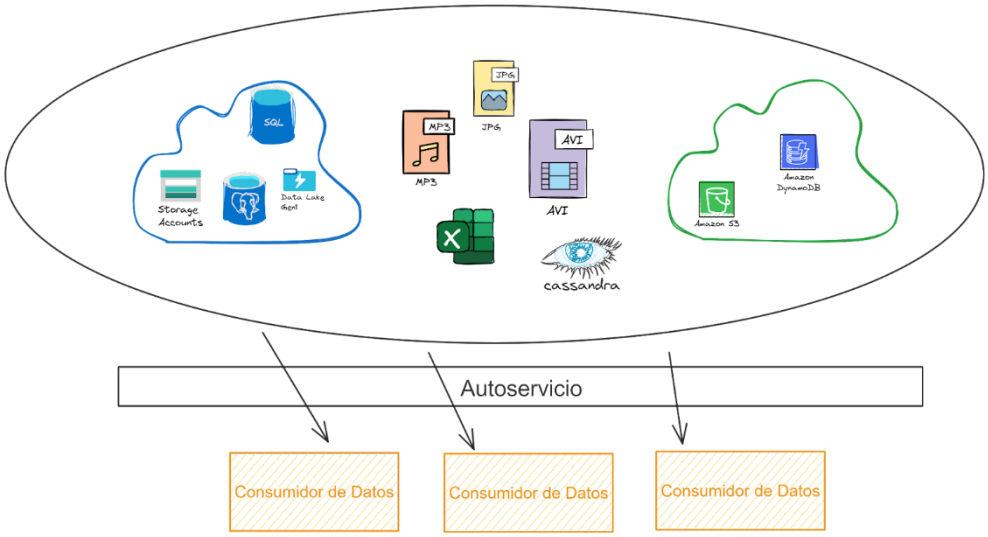


Ilustración 13: Arquitectura Data Fabric

Servicios del Data Fabric

- **Integración de datos:** La empresa puede unir cualquier fuente de datos mediante conectores y componentes pre-empaquetados, lo que elimina la necesidad de escribir código personalizado para cada integración.

- **Consumo de datos:** Proporciona capacidades de integración y consumo de datos, permitiendo que diferentes aplicaciones y sistemas accedan y utilicen los datos de manera eficiente.

- **Soporte de Big Data:** El Data Fabric es capaz de manejar grandes volúmenes de datos, tanto en tiempo real como por lotes, lo que permite a la empresa trabajar con conjuntos de datos de cualquier tamaño y complejidad.

Beneficios de Data Fabric

- **Visibilidad de los datos:** Al implementar Data Fabric, la empresa mejora la visibilidad de sus datos y la información asociada, lo que facilita la toma de decisiones informadas basadas en datos.

- **Acceso y control de los datos:** Data Fabric mejora el acceso y el control de los datos, permitiendo a la empresa definir políticas de acceso y compartir datos de manera segura entre diferentes departamentos y equipos.

- **Protección de datos:** La implementación de Data Fabric también mejora la protección de datos y la seguridad de la empresa, garantizando que los datos estén cifrados, respaldados y disponibles solo para aquellos autorizados a acceder a ellos.

2.3.3 Comparación

Cuando decidir entre aplicar procesos de Data Mesh o procesos de Data Fabric depende de varios factores, incluyendo la naturaleza de la organización, sus necesidades específicas y su capacidad para adaptarse al cambio.

El enfoque de Data Mesh se centra en un cambio organizacional, fomentando la participación de toda la plantilla en la producción y el uso efectivo de los datos, mientras que

Data Fabric se enfoca más en un cambio tecnológico, asegurando que los datos estén bien conectados y accesibles para todos los miembros de la organización.

No obstante, es importante tener en cuenta que estos enfoques no son mutuamente excluyentes. De hecho, integrar ambos procesos puede ser beneficioso para muchas organizaciones. Data Mesh y Data Fabric pueden complementarse entre sí, aprovechando las fortalezas de cada uno. Por ejemplo, se puede aprovechar la automatización y la infraestructura escalable proporcionada por Data Fabric en ciertas etapas del proceso de Data Mesh, mejorando así la velocidad y la precisión del análisis de datos.

Teniendo todo esto en cuenta, la decisión sobre qué enfoque adoptar dependerá de las necesidades y capacidades específicas de cada organización. Algunas organizaciones pueden optar por centrarse más en el aspecto organizativo y cultural de la gestión de datos, mientras que otras pueden priorizar la optimización de la infraestructura tecnológica. En cualquier caso, es importante entender que ambos enfoques tienen su lugar y pueden ser utilizados de manera efectiva para abordar los desafíos de la gestión de datos en la era digital.

Data Mesh	Data Fabric
Datos organizados a través de los propietarios de dominios	Datos organizados a través de la tecnología
Grupos distintos de equipos respecto a la administración de datos	Una única capa de administración virtual sobre los datos distribuidos
Datos tratados, controlados y manipulados por los equipos data como producto al servicio de la organización	Datos directamente controlados y presentados para el consumo de la organización
Reglas de propiedad marcadas por los consumidores de datos	Reglas de propiedad marcadas por los propietarios, desarrolladores y controladores del dato

Ilustración 14: Tabla comparación Data Fabric y Data Mesh

CAPITULO 3: METODOLOGÍA

En este capítulo, se presenta la metodología utilizada para el desarrollo e implementación del sistema de delivery cross-cloud plataforma, siguiendo el enfoque de la metodología en cascada. La metodología en cascada es un modelo de desarrollo de software secuencial que divide el proceso en fases distintas y bien definidas, donde cada fase debe completarse antes de pasar a la siguiente.

La metodología en cascada se adapta bien a proyectos con requisitos y objetivos claros desde el principio, por lo que la fase de análisis y definición de requisitos es esencial en esta metodología.

A lo largo de este capítulo, se detallan las tres fases principales del proyecto: análisis, diseño e implementación. En la fase de análisis, se define claramente el alcance del proyecto, los objetivos del sistema y los requisitos clave. En la fase de diseño, se elaboran los planes detallados para la implementación del sistema. Finalmente, en la fase de implementación, se lleva a cabo la traducción de los planes y especificaciones en acciones concretas, configuraciones y ajustes en los entornos de Azure y AWS.

Además, de una descripción detallada de la fase de verificación, donde se despliega el sistema y se realizan comprobaciones para asegurar que todas las funciones operan como se espera. Por último, se presenta la fase de mantenimiento, que es crucial en el ciclo de vida del sistema, donde se realizan actividades de monitoreo, resolución de problemas, actualizaciones y mejoras continuas.

A través de esta metodología en cascada, se garantiza una gestión efectiva del proyecto y se logra el desarrollo e implementación exitosos del sistema de delivery cross-cloud plataforma.

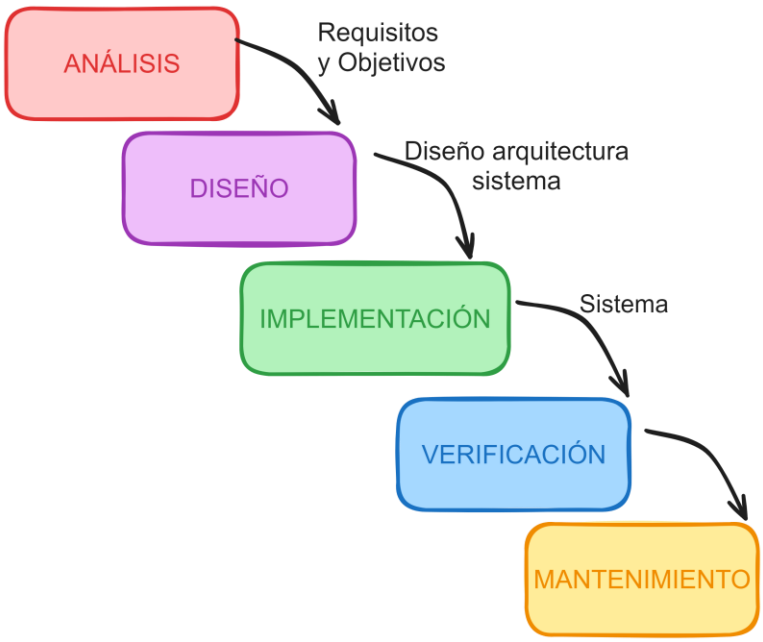


Ilustración 15: Esquema de la metodología en cascada

A continuación, se describen más detalladamente las fases principales del proyecto:

3.1 Fase de Análisis

En la fase de análisis, se define claramente el alcance del proyecto, los objetivos del sistema y los requisitos clave. Las actividades principales incluyen:

- **Identificación del Alcance:** Se establece qué funcionalidades y características incluirá el sistema de entrega cross-cloud plataforma.
- **Definición de Objetivos:** Se establecen los objetivos específicos que el sistema debe lograr, como eficiencia en la transferencia de datos y seguridad.
- **Determinación de Requisitos:** Se identifican y documentan los requisitos funcionales y no funcionales del sistema, como la capacidad de transferir datos de manera segura entre Azure y AWS.

3.2 Fase de Diseño

En la fase de diseño, se elaboran los planes detallados para la implementación del sistema basándose en los requisitos y objetivos definidos en la fase de análisis. Las actividades principales incluyen:

- **Diseño de Recursos en la Nube:** Se diseñan y planifican los recursos necesarios en las plataformas de nube AWS y Azure, y la configuración de almacenamiento necesarias.
- **Diseño del Data Flow en Azure Data Factory:** Se planeará la lógica del Data Flow y las transformaciones que se van a realizar en el
- **Diseño del pipeline en Azure Data Factory**
- **Diseño de como automatizar el sistema con un trigger**
- **Diseño del Sistema de Notificaciones:** Planear como será el diseño del sistema de notificaciones utilizando Azure Logic Apps.

3.3 Fase de Implementación

En la fase de implementación, se traducen los planes y especificaciones en acciones concretas. Por lo que se comenzara la construcción de la planificación definida en la fase anterior.

3.4 Fase de Verificación

En la fase de verificación, se despliega el sistema y se realizan comprobaciones para asegurar que todas las funciones operan como se espera. Las actividades principales incluyen:

- **Verificación de la Carga de Datos:** Se verifica que los datos se carguen correctamente en el sistema y se muevan según lo especificado.
- **Verificación de las Transformaciones de Datos:** Se asegura de que las transformaciones de datos se realicen correctamente según los requisitos del negocio.
- **Verificación del Sistema de Notificaciones:** Se comprueba que el sistema de notificaciones funcione correctamente al informar a los usuarios sobre el estado del proceso.

3.5 Fase de Mantenimiento

En esta fase se realizan actividades continuas para garantizar el funcionamiento óptimo del sistema a lo largo del tiempo. Las actividades principales incluyen:

- **Monitoreo del Sistema:** Se realiza un seguimiento constante del sistema para detectar y resolver cualquier problema que pueda surgir.

- **Resolución de Problemas:** Se toman medidas para resolver cualquier problema identificado durante el monitoreo del sistema.
- **Actualizaciones y Mejoras:** Se implementan actualizaciones y mejoras en el sistema para adaptarse a cambios en los requisitos o tecnologías, será crucial que todos los cambios y mejoras queden documentados.

CAPÍTULO 4: IMPLEMENTACIÓN DE LA METODOLOGÍA

Una vez definida y planteada la metodología en el capítulo anterior en este capítulo se procederá a detallar el paso a paso del desarrollo de cada una de sus fases incluyendo la configuración del sistema delivery de datos cross-cloud plataforma.

4.1 Fase de análisis

En la fase inicial del proyecto, se establece una comprensión clara de los objetivos y requisitos del sistema de delivery cross-cloud plataforma. Se identifican las plataformas de la nube que se utilizarán y se definen los requisitos clave del sistema. Esto permite establecer una base sólida para el diseño y la implementación del proyecto. Además, se deben crear las cuentas necesarias en AWS y Azure.

4.1.1 Objetivos del Sistema

- Establecer un sistema eficiente y confiable para la entrega de datos entre distintas plataformas de la nube.
- Implementar un sistema de notificaciones utilizando Azure Logic Apps, que informe a los usuarios sobre el estado y los eventos importantes del proceso de entrega de datos.
- Debe ser un sistema automatizado.
- Garantizar la integridad, seguridad y disponibilidad de los datos transferidos entre las plataformas.

4.1.2 Plataformas de la Nube

- Se utilizará Azure como plataforma principal para el desarrollo y la implementación del sistema. Se usará Azure Data Factory para la creación del pipeline que ejecutará el Data Flow donde se realizará el proceso ETL (Extracción, transformación y carga) y se activará el sistema de notificaciones.
- Se integrará con AWS para la entrega de datos, aprovechando las capacidades y servicios disponibles en ambas plataformas.

4.1.3 Requisitos Clave del Sistema

1. **Transferencia de Datos:**
    - El sistema debe ser capaz de transferir datos de manera eficiente y segura entre Azure y AWS.
    - Se deben implementar transformaciones de datos (join, filter, select, aggregate) según los requisitos del negocio.
2. **Notificaciones**:
    - Se debe establecer un sistema de notificación que informe a los usuarios tanto si la entrega ha sido realizada con éxito o no.
3. **Integración y Automatización:**
    - El sistema debe integrarse con los servicios de almacenamiento en la nube, como Azure Blob Storage y AWS S3, para la transferencia de datos.
    - Se debe configurar un mecanismo de automatización utilizando un trigger para que el pipeline se ejecute de manera programada y sin intervención manual una vez que se inserten nuevos datos en el Blob de Azure.
4. **Creación de Cuentas y Recursos:**
    - Se debe crear una cuenta en AWS y un bucket en S3 para recibir los datos.
    - En Azure, se debe crear un grupo de recursos que contendrá Azure Data Factory, Azure Blob Storage y Azure Logic Apps.

4.2 Fase de diseño

En esta fase, se realizó el diseño detallado de los requisitos y expectativas del sistema para trabajar con la base de datos AdventureWorks sobre compras online. El enfoque está en la obtención de información sobre las devoluciones según unas características determinadas.

4.2.1 Creación de Recursos en la Nube

Se creará un grupo de recursos en Azure que contendrá una cuenta de almacenamiento. Dentro de esta cuenta de almacenamiento, se creará un blob llamado AdventureWorks donde se cargarán todos los CSV de la base de datos. Dentro del blob, se creará un di-

rectorio llamado Lookup donde se cargarán todos los CSV de devoluciones (ReturnsData, ProductLookup, ProductSubCategoryLookUp y TerritoryLookup) y otro directorio llamado Basura para mover los datos una vez sean procesados para evitar generar datos duplicados. Además, se creará una cuenta en AWS y un bucket en S3 que será el destino donde serán enviados nuestros datos.

Ilustración 16: Recursos en las plataformas de la nube

4.2.2 Diseño de la integración de Azure Data Factory

Se diseñará la integración de Azure Data Factory con los servicios de Azure Blob Storage y AWS S3 para la creación de nuestro pipeline.

4.2.3 Diseño del Sistema de Notificaciones

Se diseñará un sistema de notificaciones utilizando Azure Logic Apps. Este sistema informará a los usuarios sobre el estado y los eventos importantes del proceso de entrega de datos. Se configurará para que cuando se ejecute un pipeline se envíe un correo.

4.2.4 Diseño del Data Flow

Se diseñará un Data Flow en Azure Data Factory donde se configurarán y ejecutarán las transformaciones de datos requeridas. Este Data Flow incluirá la configuración de los conjuntos de datos para que, una vez extraídos de su directorio, se muevan a nuestro directorio de basura, la unión de archivos CSV por sus columnas en común, el filtrado de filas por territorio y categoría, la selección de filas de interés y la configuración del envío del CSV final al bucket de AWS S3.

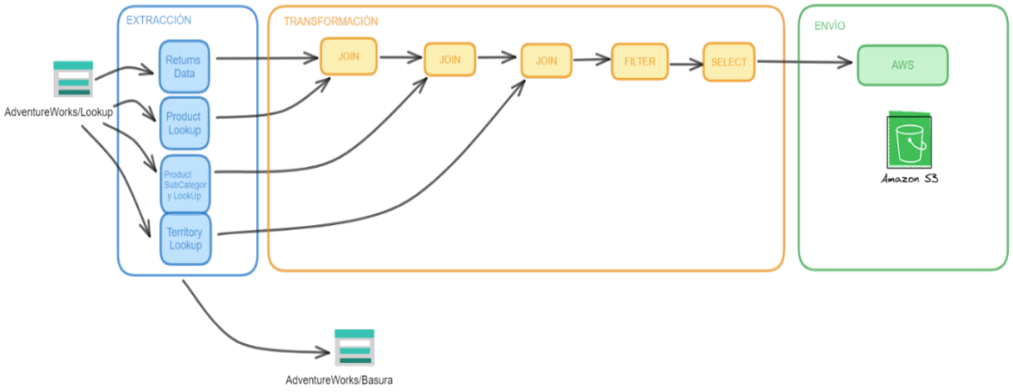


Ilustración 17: Esquema del Data Flow

4.2.5 Diseño del Pipeline

Una vez creado el Data Flow, se diseñará un pipeline en Azure Data Factory que ejecute el Data Flow. Este pipeline incluirá actividades que ejecutarán la función creada en Azure Logic Apps para enviar el correo tanto si la ejecución es correcta como si da problemas.

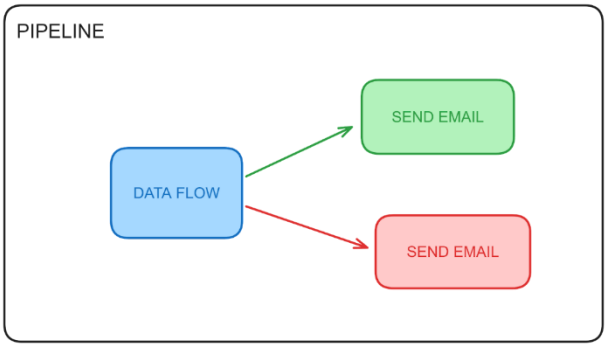


Ilustración 18: Esquema diseño del pipeline

4.2.6 Diseño del Trigger

Una vez finalizado el pipeline, se diseñará un trigger que se activará cuando se añadan nuevos archivos CSV a nuestro directorio Lookup dentro del blob AdventureWorks. Este trigger iniciará automáticamente el proceso de entrega de datos.

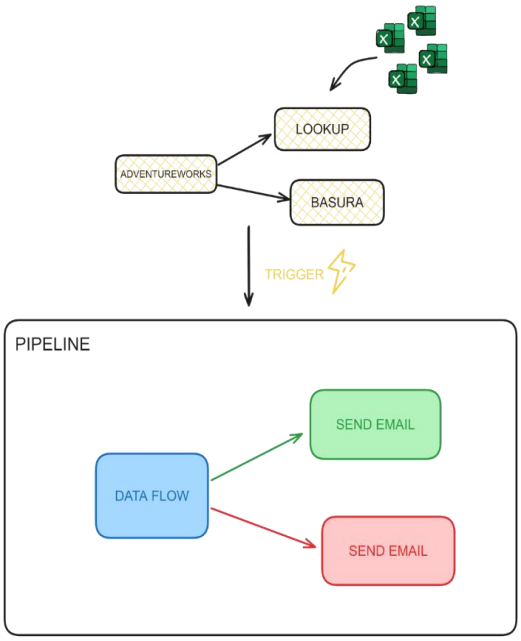


Ilustración 19: Esquema diseño trigger

4.3 Fase de implementación

En esta fase, se llevará a cabo la implementación del diseño detallado en la fase de diseño. Se traducirán los planes y especificaciones en acciones concretas, configuraciones y ajustes en los entornos de Azure y AWS. A continuación, se detallan las actividades clave de esta fase:

4.3.1 Implementación de Recursos en la Nube

Se inicio sesión en la cuenta de Azure y se creó el grupo de recursos, seguido de la cuenta de almacenamiento y el Azure Data Factory que estarán asociados a este grupo de recursos.

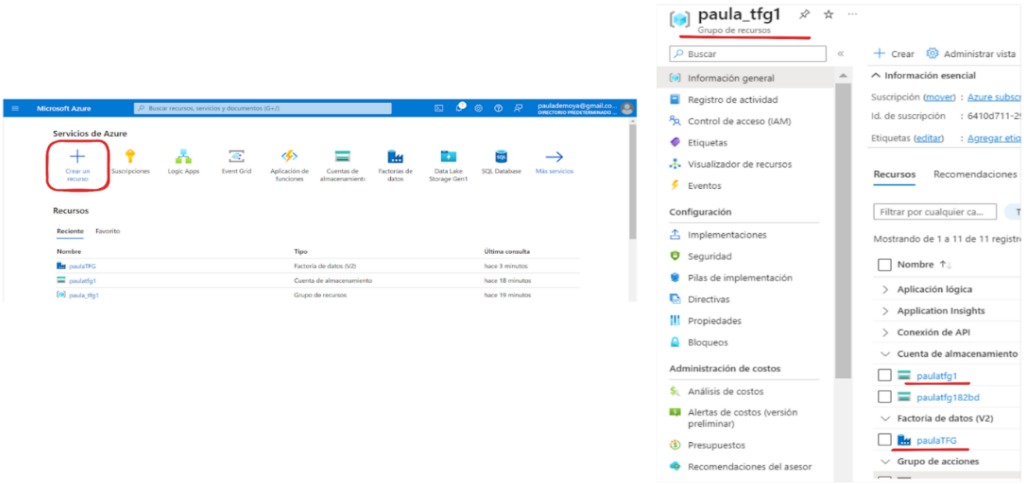


Ilustración 20: Imagen crear recursos en Azure

Dentro de la cuenta de almacenamiento, se creó un blob para cargar los CSV de la base de datos AdventureWorks. En este blob, se crearon dos directorios: Lookup, donde se cargarán los CSV sobre devoluciones que se emplearán en el sistema, y Basura, donde se moverán los CSV utilizados una vez se ejecute el Data Flow.

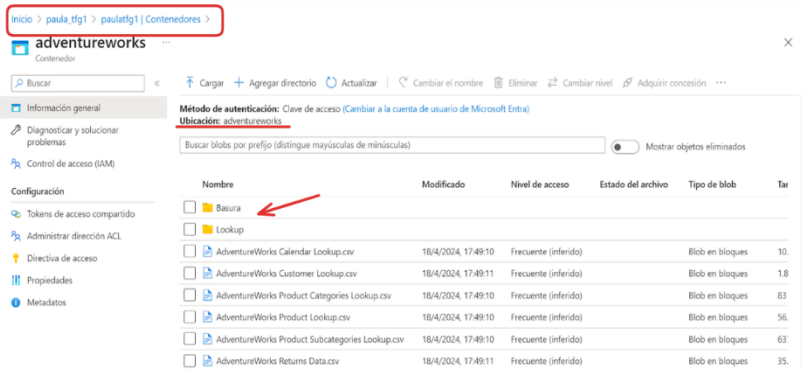


Ilustración 21: Creación del Blob y los directorios en Azure

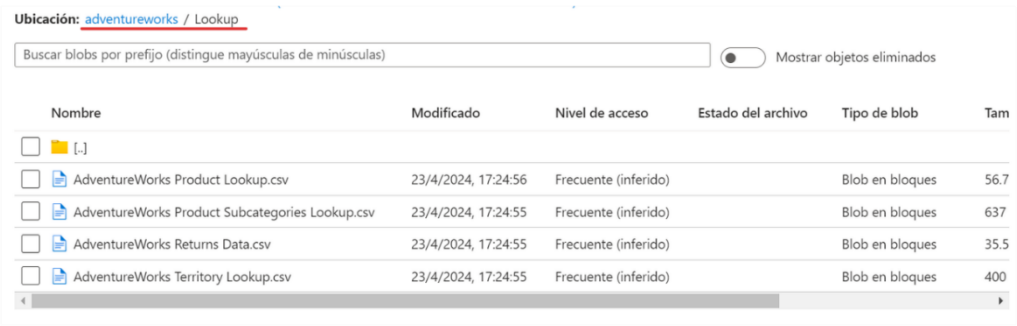


Ilustración 22: Carga de datos en el directorio

Además, se inicia sesión en AWS y se crea un servicio en Amazon S3 donde se creará el bucket que será el destino final de la entrega de datos del sistema.

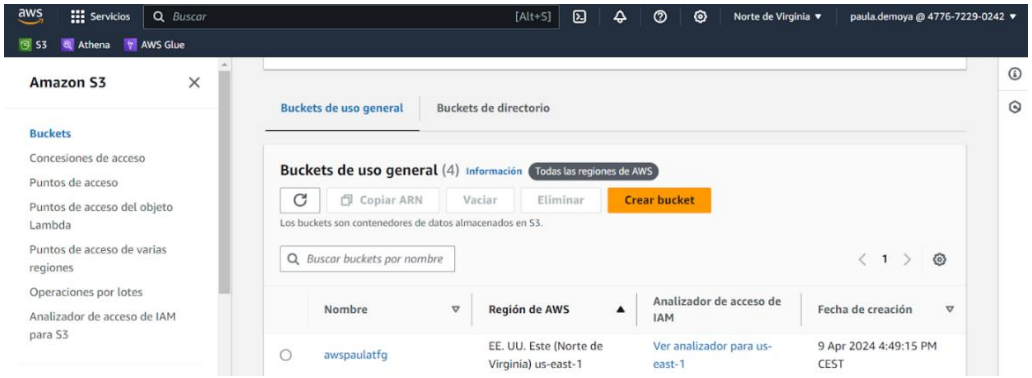


Ilustración 23: Creación del servicio AWS S3

4.3.2 Implementación de la Integración de Azure Data Factory

Una vez creado el Azure Data Factory, se accede al portal y se configura para integrarse con Azure Blob Storage y AWS S3. Para vincular el servicio con AWS S3, se utilizan las credenciales creadas al crear la cuenta en AWS.

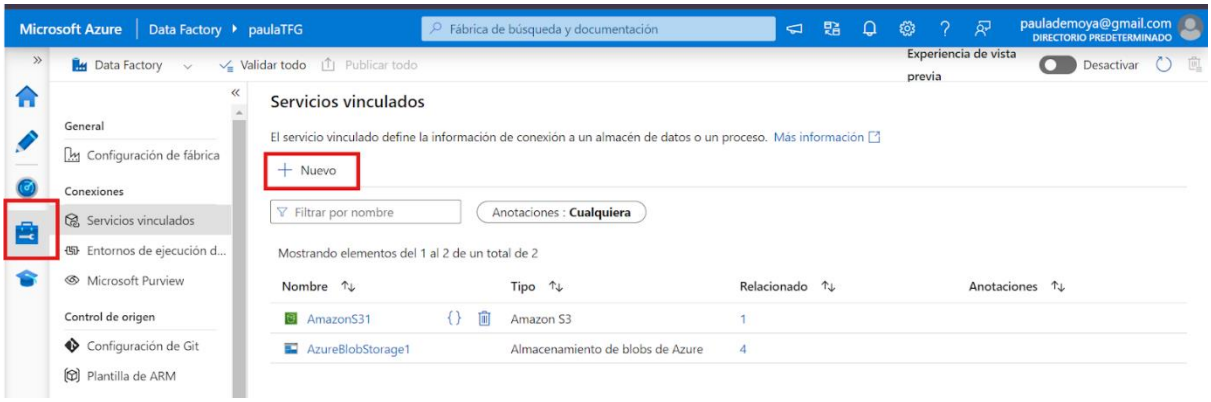


Ilustración 24: Vincular servicios en Azure

Ilustración 25: Vinculación con AWS S3

4.3.3 Implementación del Data Flow

Para la implementación del Data Flow, se cargaron todos los conjuntos de datos especificando la ruta de la cuenta de almacenamiento y se creó un Data Flow para comenzar a configurarlo.

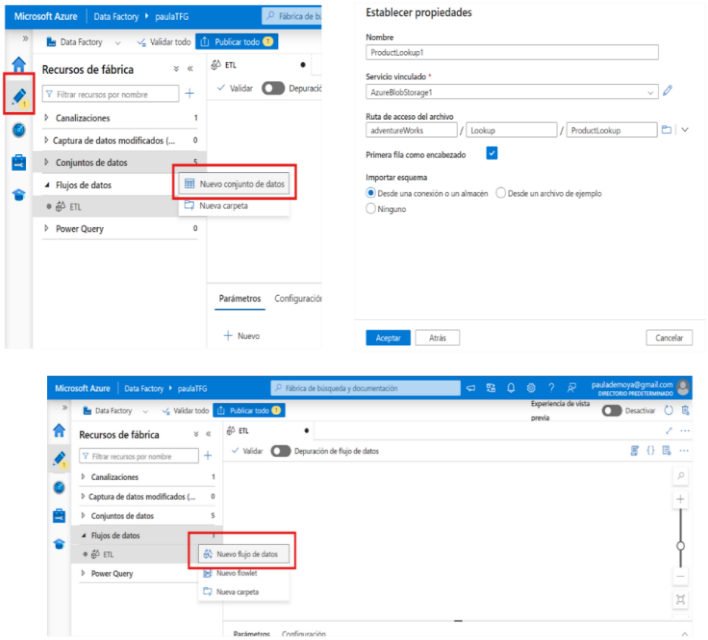


Ilustración 26: Carga de conjuntos de datos y creación del Data Flow en Azure

Se agregan al Data Flow los CSV que se han cargado previamente y se configuran para que, una vez cargados, se muevan al directorio Basura creado anteriormente.

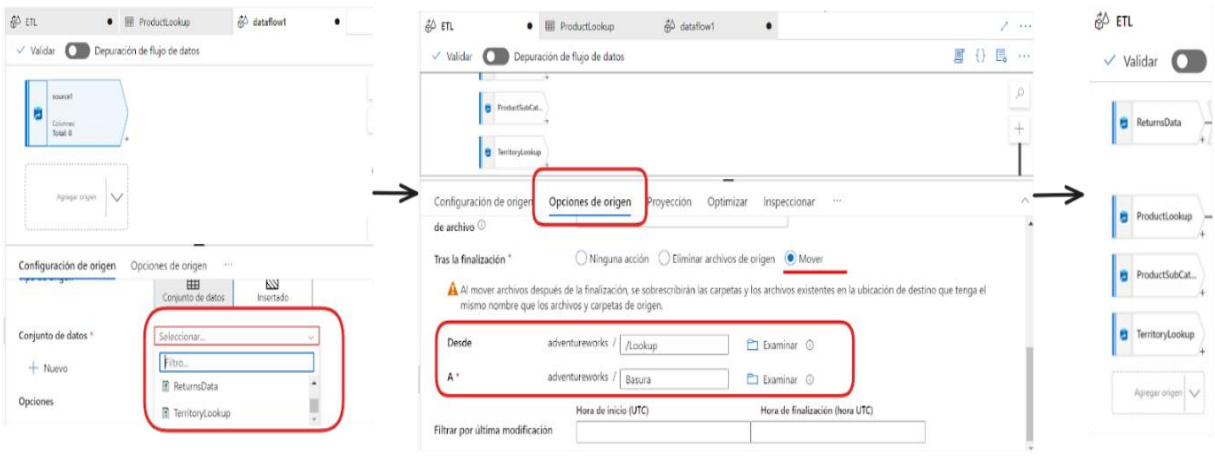


Ilustración 27: Carga de ficheros en el Data Flow

Una vez configurados todos los orígenes de datos, se agregaron las funciones necesarias para implementar las transformaciones definidas en la fase de análisis.

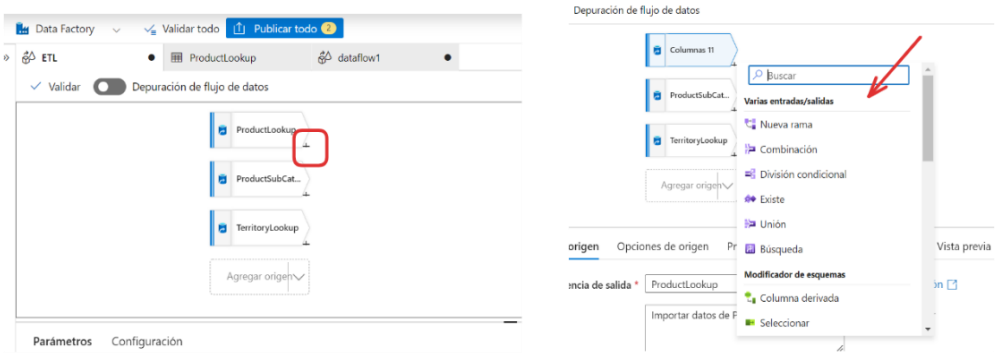


Ilustración 28: Agregar funciones al Data Flow

- **Unión de CSVs**: En este proyecto específico, se realizó inicialmente funciones de unión (join) de los cuatro archivos CSV de devoluciones, es crucial realizar la unión a través de las columnas en común. La primera unión se realizó entre ReturnsData y ProductLookup cuando los valores del atributo ProductKey son iguales. A partir del resultado de esta unión, se realizó una segunda unión con el CSV ProductSubCategoryLookup cuando los atributos ProductSubCategoryKey son iguales. Finalmente, se une el CSV resultante con el último CSV, TerritoryLookup, cuando TerritoryKey es igual a SalesTerritoryKey.

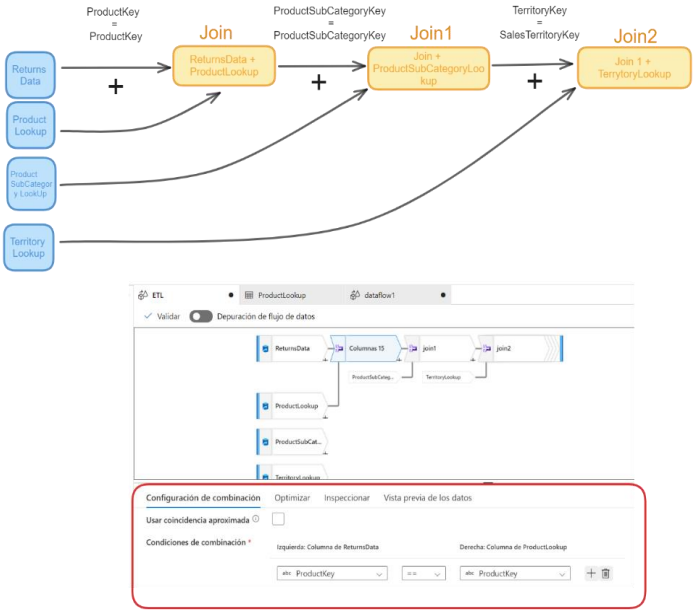


Ilustración 29: Función join en Azure Data Factory

- **Función de Columna Derivada:** Con el CSV combinado final, se procede a configurar una función de columna derivada. Se crearon dos columnas que devolverán True o False según se cumpla la condición o no. En este proyecto, se filtró por la categoría de producto ‘Accesories’ que corresponde al código 4 y por ‘Francia’ que corresponde al código 7. Las columnas pueden configurarse según las necesidades de negocio específicas en cualquier momento.

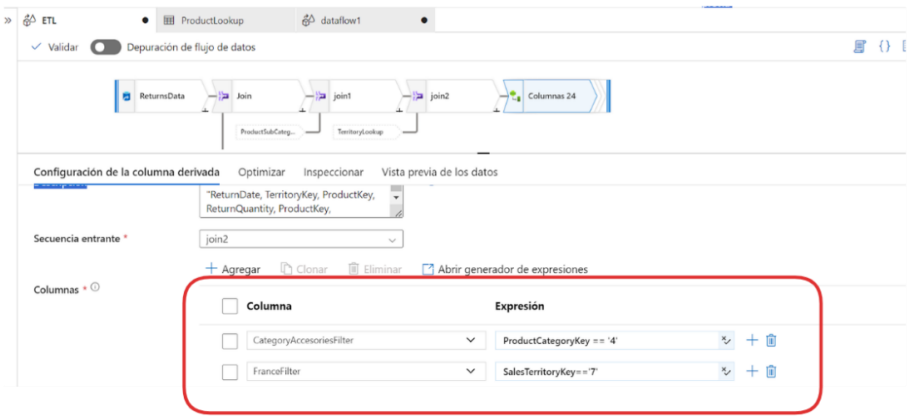


Ilustración 30: Configuración de una columna derivada en Azure Data Factory

- **Función de Filtrado:** A continuación, se configuro una función de filtrado que filtrará los registros en los que ambas columnas derivadas sean True, es decir,

que se cumplan las condiciones de ambas columnas derivadas. El filtro puede modificarse para añadir, eliminar columnas derivadas o cambiar las condiciones y en vez de hacer que se cumplan ambas (& ‘and’) que se cumpla una u otra (|| ‘or’) según las variaciones de las necesidades de negocio.

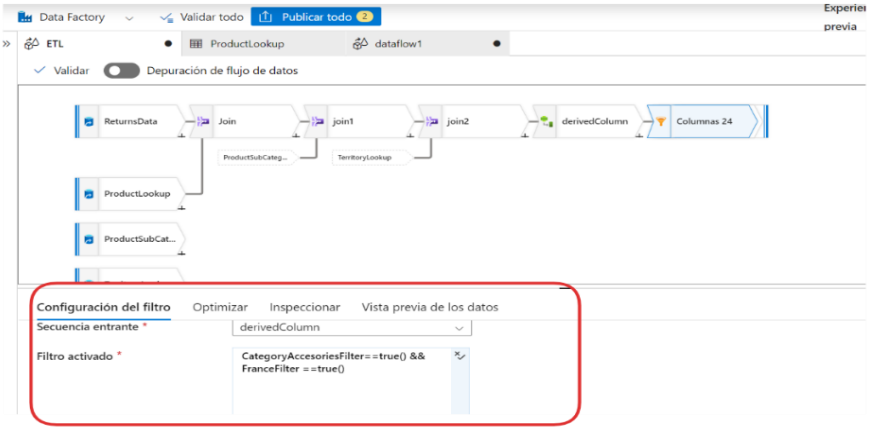


Ilustración 31: Función de filtrado en Azure Data Factory

- **Función Select y Configuración del Envío:** Antes de realizar el envío al bucket de AWS S3, se realizó una función select para especificar qué atributos se desean en el CSV final. Se pueden eliminar columnas que no sean de interés, como la descripción o las medidas del producto, y cambiar el nombre de las columnas por otro más descriptivo.

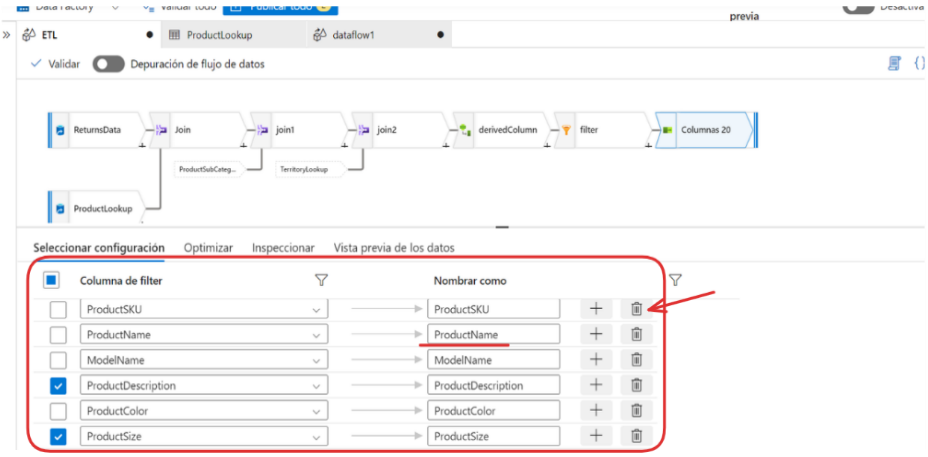


Ilustración 32: Función select en Azure Data Factory

Para configurar el envío, se creó un nuevo CSV donde se guardó el contenido del dataframe final y se especificó la ruta del bucket de AWS S3 como destino.

Asi el Data Flow con la ETL (Extracción, Transformación y Envió) quedaría configurado.

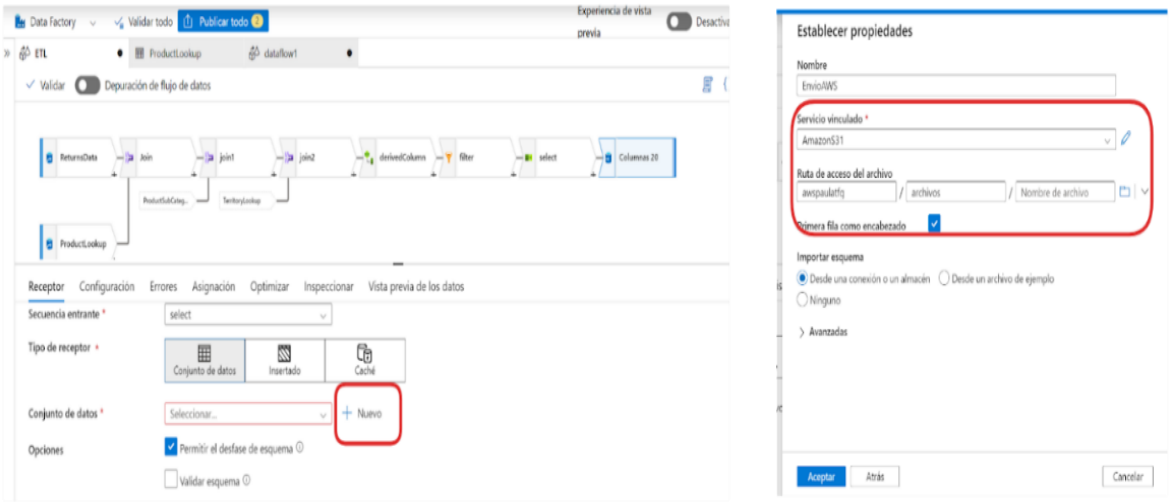


Ilustración 33: Configuración del envió en Azure Data Factory

4.3.4 Implementación del Sistema de Notificaciones

En primer lugar, se creó Azure Logic Apps y se asoció al grupo de recursos existente. Una vez desplegada la Logic App, se inició la creación del flujo de datos desde el "Diseñador de Aplicación Lógica".

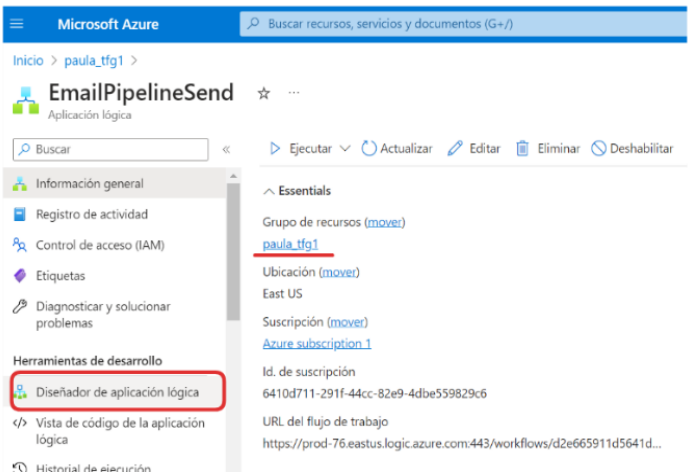


Ilustración 34: Creación Azure Logic Apps

Para iniciar el flujo de trabajo, se utilizan disparadores o desencadenadores. En este caso, se utilizó el disparador "When a HTTP request is received" para enviar un correo

cuando se ejecute un pipeline. Este disparador se configura con un esquema JSON específico como aparece en la imagen.

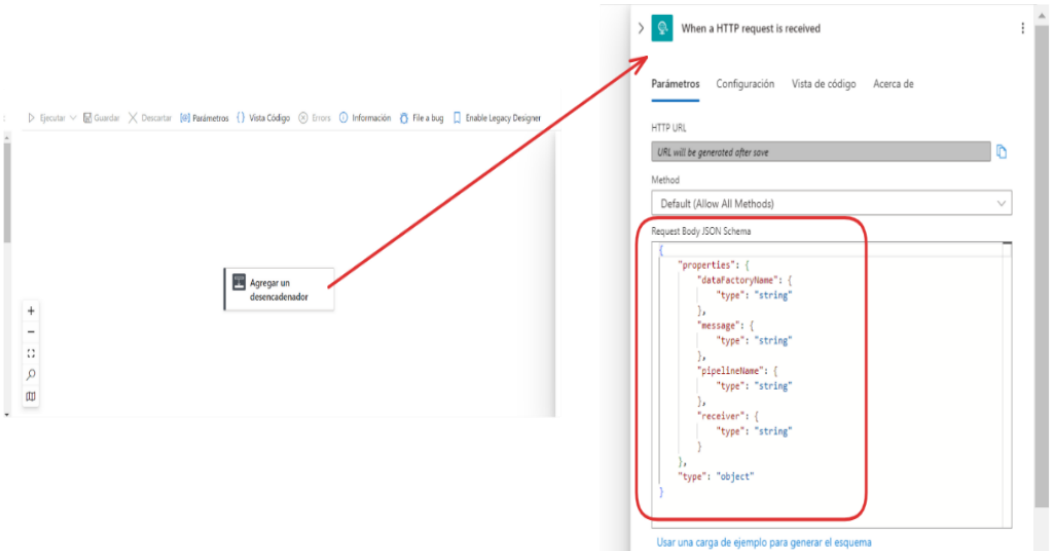


Ilustración 35: Crear un disparador en Azure Logic Apps


```
{
"properties": {
"dataFactoryName": {
"type": "string"
},
"message": {
"type": "string"
},
"pipelineName": {
"type": "string"
},
"receiver": {
"type": "string"
}
},
"type": "object"
```

Tras crear el disparador que se activará cuando se ejecute un pipeline, se procedió a configurar la acción encargada de enviar el correo electrónico. Aquí se pueden configurar parámetros como el mensaje y el asunto del correo, permitiendo personalizar el contenido del correo según las necesidades.

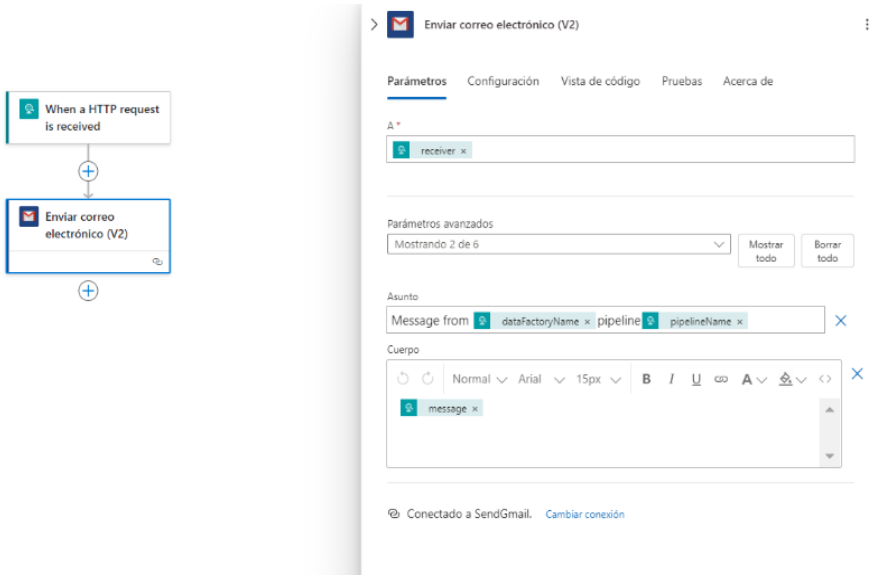


Ilustración 36: Configurar una acción en Azure Logic Apps

Una vez configurado el flujo de trabajo, se debe de copiar su URL para poder activarlo con el pipeline que se creara a continuación.

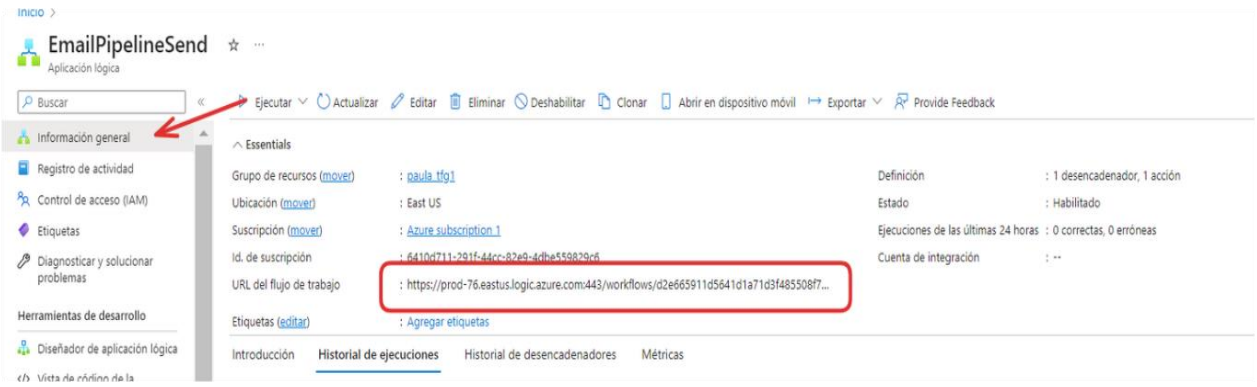


Ilustración 37: URL de la aplicación lógica

4.3.5 Implementación del pipeline

Después de crear el flujo de trabajo de la aplicación lógica para enviar correo electrónico, se activó desde un pipeline en Data Factory mediante una actividad web donde también se añadirá el Data Flow.

Primero se creó el pipeline o canalización y se añadió el Data Flow con el proceso de ETL creado anteriormente.

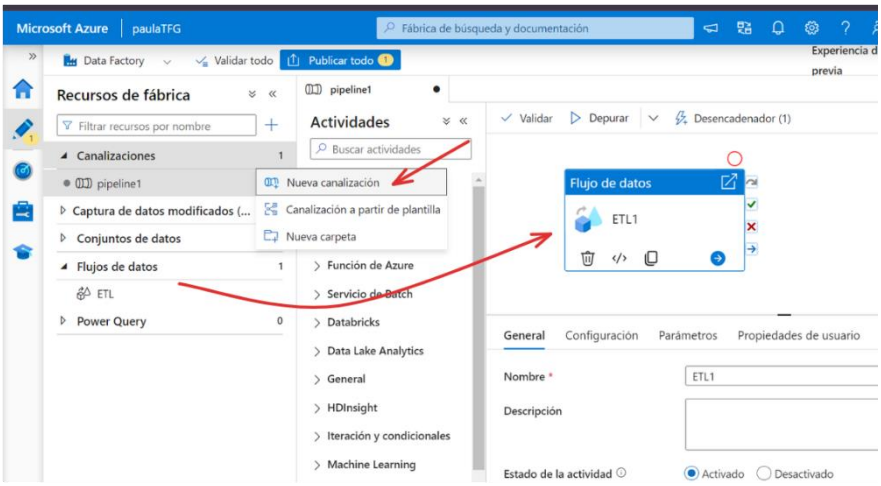


Ilustración 38: Creación de un pipeline en Azure Data Factory

Para activar el flujo de trabajo de la aplicación lógica, se añadieron dos actividades web, tanto para si la ejecución del Dataflow ha sido exitosa como si no. Para configurar la actividad web, se proporcionó la URL del flujo de trabajo de la aplicación lógica que se creó anteriormente en el campo URL. Se debe proporcionar el siguiente JSON para el cuerpo según la actividad.

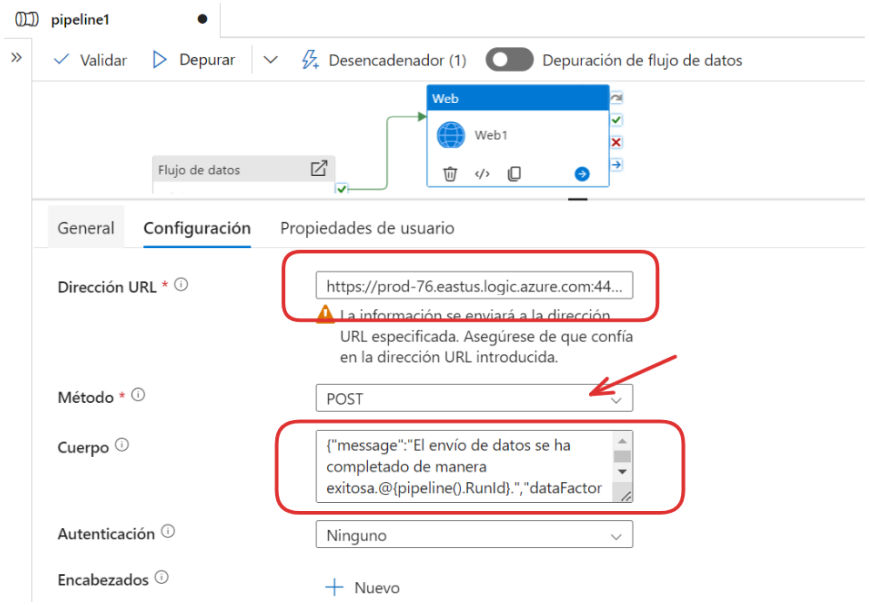


Ilustración 39: Configuración de una acción web en Azure Data Factory

```
{
"message": "Ha ocurrido un error en la entrega de datos, por favor revise que ha podido ocurrir. @{pipeline().RunId}.",
"dataFactoryName" :"@{pipeline().DataFactory}",
"pipelineName" : "@{pipeline().Pipeline}",
"receiver" : "@{pipeline().parameters.receiver}"
}
```

```
{
"message": "Ha ocurrido un error en la entrega de datos, por favor revise que ha podido ocurrir. @{pipeline().RunId}.",
"dataFactoryName": "@{pipeline().DataFactory}",
"pipelineName": "@{pipeline().Pipeline}",
"receiver": "@{pipeline().parameters.receiver}"
}
```

Se selecciona el área de fondo del diseñador del pipeline para asi seleccionar la página de propiedades del pipeline y se agregó un nuevo parámetro llamado receptor, proporcionando una dirección de correo electrónico como valor predeterminado.

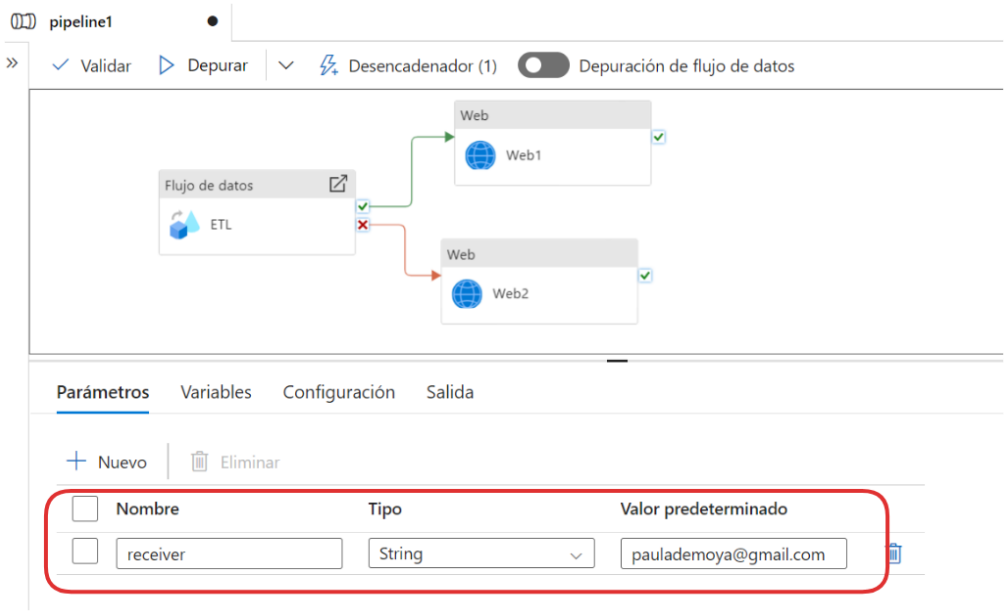


Ilustración 40: Configuración de parámetros en el pipeline

Se público el pipeline y luego se activó manualmente para confirmar que el correo electrónico se envió como se esperaba.

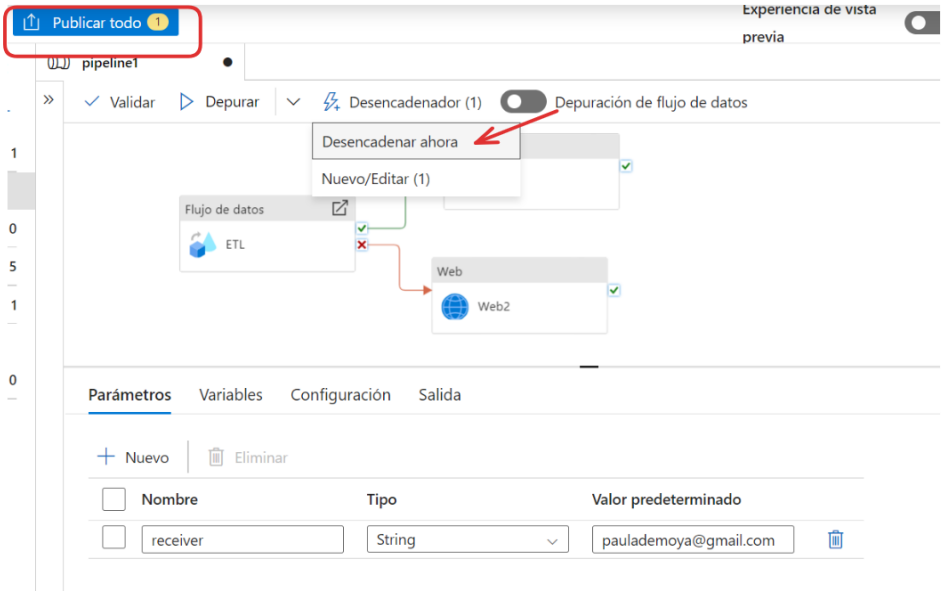


Ilustración 41: Como ejecutar el pipeline manualmente

4.3.6 Implementación del trigger

Para finalizar el sistema de delivery cross-cloud, se automatizo configurando un trigger que ejecute el pipeline cuando se carguen los CSV en el directorio Lookup de la siguiente manera.

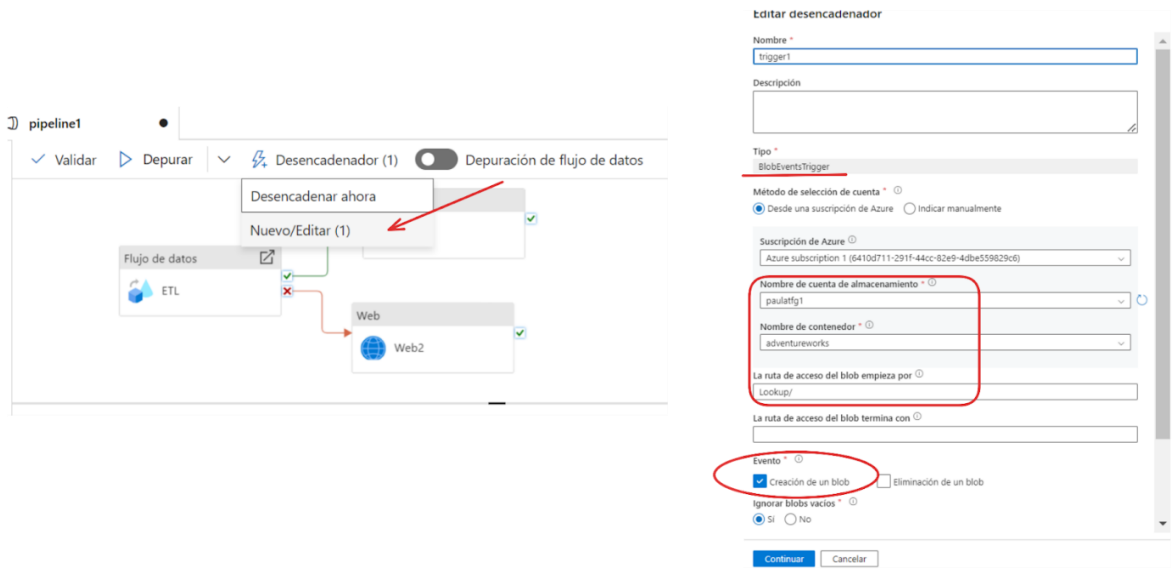


Ilustración 42: Creación de un Trigger en Azure Data Factory

4.4 Fase de Verificación

Se despliega el sistema y se realizan comprobaciones para asegurar que todas las funciones operan como se esperaba. A continuación, se detallan las actividades clave de esta fase:

4.4.1Verificación de la Carga de Datos

Se comprobó que los datos se cargan correctamente en el sistema. Esto implico verificar que los archivos CSV se han cargado correctamente en el blob de Azure y que, una vez procesados, se han movido al directorio Basura.

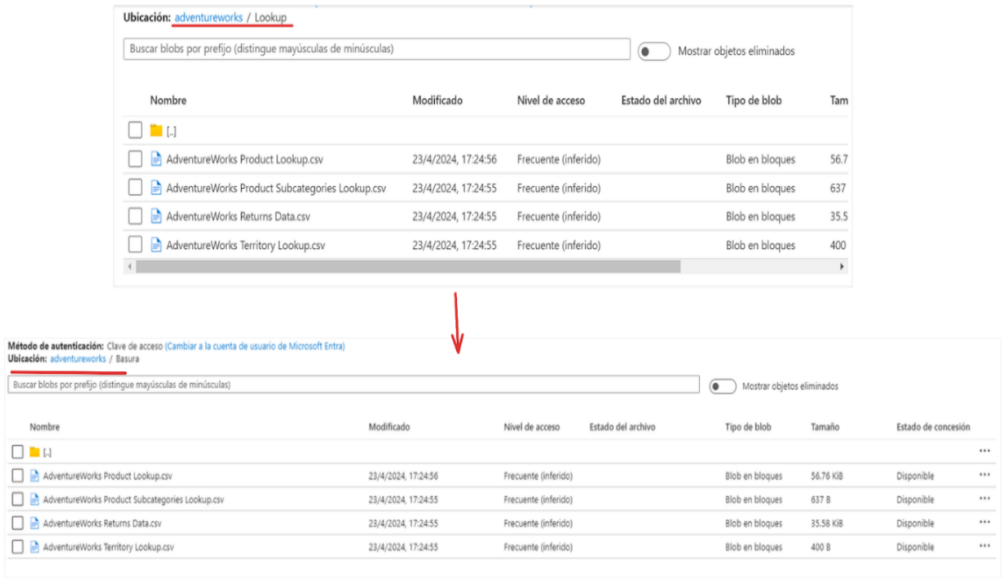


Ilustración 43: Comprobación de carga de datos correcta en Azure Blob Storage

4.4.2 Verificación de las Transformaciones de Datos

Se verifico que las transformaciones de datos se realizaron correctamente. Esto implico comprobar que las funciones de unión, filtrado y selección de columnas se ejecutaron como se esperaba. Para realizar esta verificación, se puede utilizar la opción de depuración de flujo de datos en Azure Data Factory, que permite inspeccionar los datos en cada paso del flujo de datos.

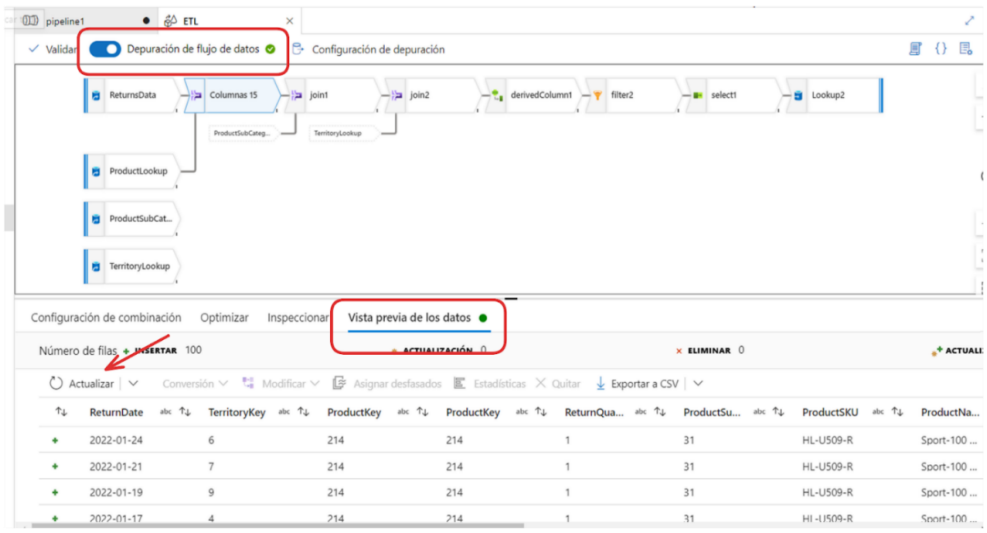


Ilustración 44: Depuración de Data Flow en Azure Data Factory

Se debe ir actualizando la vista previa de cada una de las funciones e ir visualizando si se están ejecutando de manera correcta.

4.4.3 Verificación del Sistema de Notificaciones

Se verifico que el sistema de notificaciones funciona correctamente. Esto implico comprobar si se envía un correo electrónico cuando se ejecuta un pipeline, tanto si la ejecución es exitosa como si ocurren problemas.

Al cargar los archivos CSV en el directorio Lookup, el sistema se activa automáticamente, lo que pone a prueba tanto el disparador del pipeline como el sistema de notificaciones. En caso de una ejecución exitosa, se espera recibir una notificación de éxito.

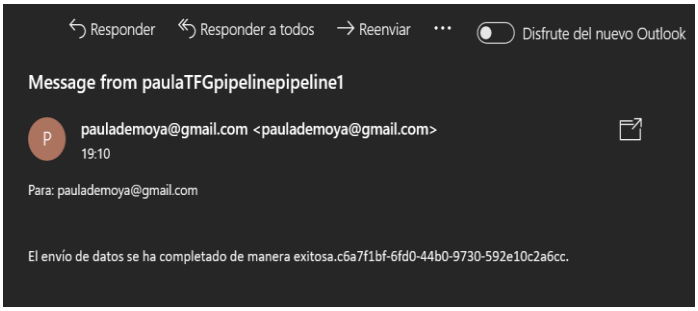


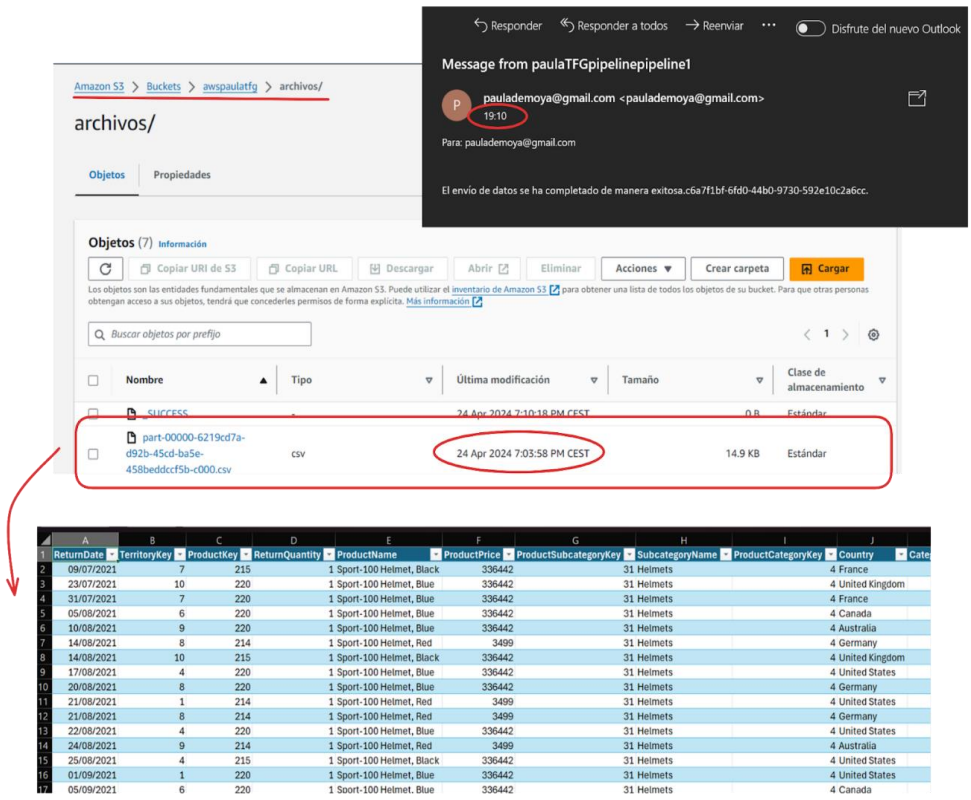
Ilustración 45: Email envió de datos exitoso

Para validar que el sistema de notificaciones opera correctamente, en especial en situaciones donde la ejecución del pipeline no se completa con éxito, se ha realizado una prueba específica. Esta prueba consiste en no cargar todos los archivos CSV necesarios en el directorio correspondiente, lo que intencionalmente provoca un error en la ejecución del pipeline.

Ilustración 46: Email fallo envió de datos

4.4.4 Verificación de la Entrega de Datos

Se verifico que los datos se entregan correctamente al destino final. Esto implico comprobar que los datos finales se enviaron correctamente al bucket de AWS S3 y que llegaron a Azure.



	A	B	C	D	E	F	G	H	I	J
1	ReturnDate	TerritoryKey	ProductKey	ReturnQuantity	ProductName	ProductPrice	ProductSubcategoryKey	SubcategoryName	ProductCategoryKey	Country
2	09/07/2021	7	215	1	Sport-100 Helmet, Black	336442	31	Helmets	4	France
3	23/07/2021	10	220	1	Sport-100 Helmet, Blue	336442	31	Helmets	4	United Kingdom
4	31/07/2021	7	220	1	Sport-100 Helmet, Blue	336442	31	Helmets	4	France
5	05/08/2021	6	220	1	Sport-100 Helmet, Blue	336442	31	Helmets	4	Canada
6	10/08/2021	9	220	1	Sport-100 Helmet, Blue	336442	31	Helmets	4	Australia
7	14/08/2021	8	214	1	Sport-100 Helmet, Red	3499	31	Helmets	4	Germany
8	14/08/2021	10	215	1	Sport-100 Helmet, Black	336442	31	Helmets	4	United Kingdom
9	17/08/2021	4	220	1	Sport-100 Helmet, Blue	336442	31	Helmets	4	United States
10	20/08/2021	8	220	1	Sport-100 Helmet, Blue	336442	31	Helmets	4	Germany
11	21/08/2021	1	214	1	Sport-100 Helmet, Red	3499	31	Helmets	4	United States
12	21/08/2021	8	214	1	Sport-100 Helmet, Red	3499	31	Helmets	4	Germany
13	22/08/2021	4	220	1	Sport-100 Helmet, Blue	336442	31	Helmets	4	United States
14	24/08/2021	9	214	1	Sport-100 Helmet, Red	3499	31	Helmets	4	Australia
15	25/08/2021	4	215	1	Sport-100 Helmet, Black	336442	31	Helmets	4	United States
16	01/09/2021	1	220	1	Sport-100 Helmet, Blue	336442	31	Helmets	4	United States
17	05/09/2021	6	220	1	Sport-100 Helmet, Blue	336442	31	Helmets	4	Canada

Ilustración 47: Verificación en AWS S3 de entrega de datos

4.5 Fase de mantenimiento

Esta es una etapa crucial en el ciclo de vida de cualquier sistema. A continuación, se detallan las actividades clave de esta fase:

- **Monitoreo del Sistema**: Se debe realizar un seguimiento constante del sistema para asegurar que está funcionando correctamente. Esto implica la revisión continua de las notificaciones y la comprobación de la entrega de datos en S3.
- **Resolución de Problemas**: Si se identifica algún problema durante el monitoreo del sistema, se toman medidas para resolverlo. Esto puede implicar la corrección de errores en el código, la optimización de las transformaciones de datos o la actualización de las configuraciones del sistema.

- **Actualizaciones y Mejoras**: Con el tiempo, pueden surgir nuevas necesidades o mejoras potenciales para el sistema. En la fase de mantenimiento, se implementan estas actualizaciones y mejoras. Esto puede implicar la adición de nuevas funciones, la mejora de la eficiencia del sistema o la actualización del sistema para trabajar con nuevos formatos de datos o tecnologías.
- **Documentación**: La documentación del sistema se de actualizar durante la fase de mantenimiento. Esto incluye la documentación de cualquier cambio realizado en el sistema, así como la actualización de la documentación existente para reflejar las actualizaciones y mejoras.

4.6 Adaptabilidad y Flexibilidad del Sistema

Una de las características más destacadas del sistema de delivery cross-cloud plataforma es su capacidad para adaptarse a una amplia gama de necesidades y entornos de datos en línea con los principios fundamentales del Data Fabric. Aunque en el desarrollo específico se ha detallado un escenario donde se transfieren archivos CSV desde Azure a Amazon S3, es importante destacar que el sistema está diseñado para manejar una variedad de formatos de datos y ubicaciones, así como para integrarse con diversas plataformas en la nube.

4.6.1 Integración con Diversas Plataformas

Además de la transferencia a AWS S3, el sistema puede ser configurado para enviar datos a otras plataformas tanto en la nube, como Google Cloud Storage, IBM Cloud Object Storage, o cualquier otro servicio de almacenamiento gracias a su flexibilidad de poder vincularse con una gran variedad de servicios.

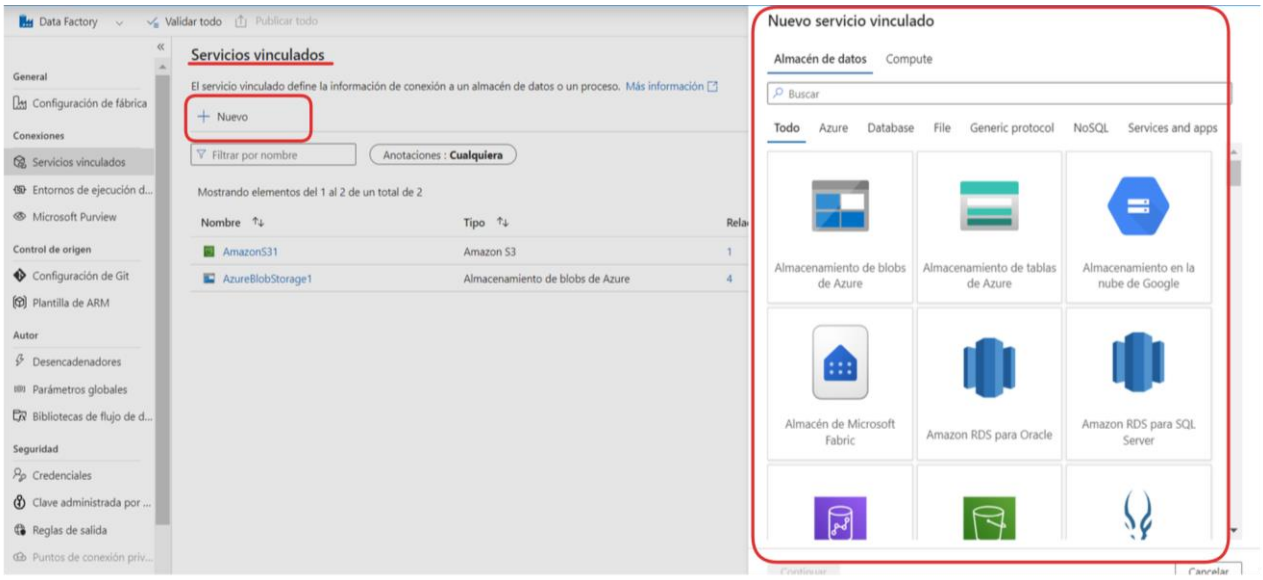


Ilustración 48: Vinculación de diversos servicios en Azure Data Factory

4.6.2 Configuración Versátil para Diferentes Formatos y Ubicaciones de Datos

El sistema puede ser configurado para trabajar con diferentes formatos de datos, ya sea CSV, JSON, XML u otros, y puede extraerlos de ubicaciones diversas tanto dentro de Azure, como Azure Blob Storage, Azure Data Lake Storage, o incluso bases de datos en la nube como Azure SQL Database, como de otros servicios de almacenamiento fuera de la nube, ya que, Azure Data Factory puede integrase con una variedad muy amplia de servicios.

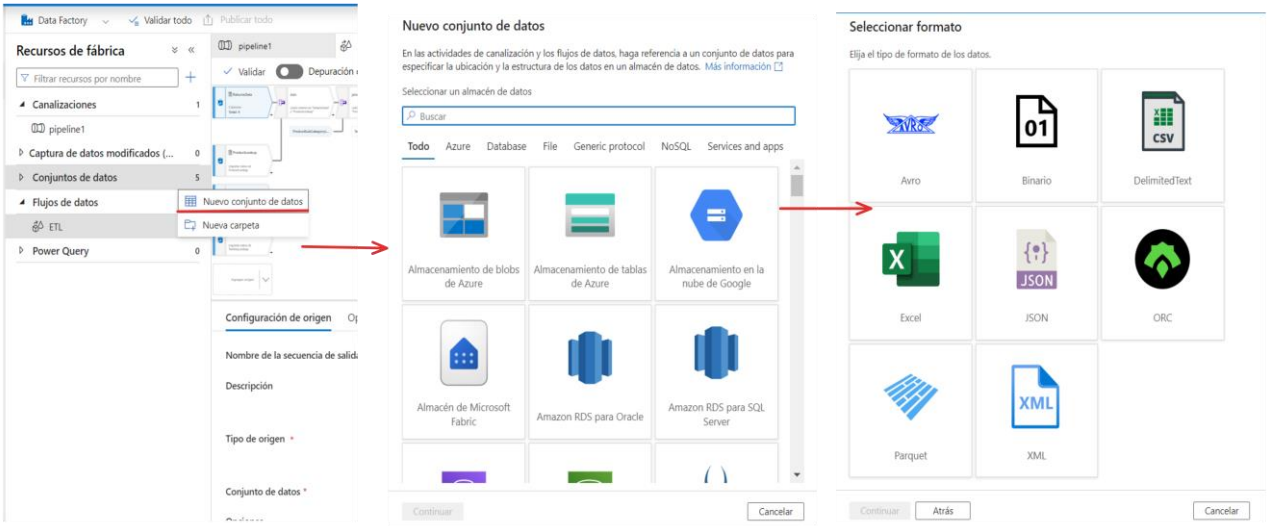


Ilustración 49: Configuración de conjuntos de datos en Azure Data Factory

La adaptabilidad y flexibilidad del sistema permiten que se configure de manera óptima y personalizada para cada proyecto y organización independientemente de los formatos de datos, ubicaciones o plataformas involucradas incluyendo las transformaciones a realizar. Esta capacidad de personalización lo convierte en una solución altamente adaptable y escalable para los desafíos en evolución del manejo de datos en entornos cloud.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Este proyecto ha logrado exitosamente la creación e implementación de un sistema de entrega de datos cross-cloud basado en Data Fabric, integrando con éxito las plataformas de Azure y AWS. Esta solución es especialmente valiosa para las pequeñas y medianas empresas (PYMEs), ya que simplifica la transferencia de datos de manera automática, reduciendo errores y aumentando la confiabilidad del proceso, además de facilitar la gestión de datos en entornos con múltiples servicios en la nube.

Al usar tecnologías avanzadas como Azure Data Factory, el sistema se vuelve más eficiente y escalable al adaptarse a las necesidades cambiantes de las empresas con una configuración personalizada. Debido a esta capacidad de adaptación y simplificación en la transferencia de datos, las PYMEs pueden sacar el máximo provecho de sus recursos tecnológicos.

En síntesis, este proyecto ha ofrecido una solución novedosa y eficaz para el intercambio de datos entre diferentes servicios en la nube, lo que marca un importante avance en la manera en que las pequeñas y medianas empresas abordan y administran los desafíos complejos relacionados con el manejo de datos en un entorno cada vez. El éxito de este sistema no solo establece un precedente significativo para futuros desarrollos en la ciencia de datos y la ingeniería de sistemas en la nube, sino que también confirma la viabilidad de soluciones avanzadas en entornos con recursos limitados.

También es crucial destacar que este trabajo ha contribuido a lograr diversos Objetivos de Desarrollo Sostenible (ODS). Primero, promovió la creación de infraestructuras resilientes respaldando el ODS 9. Utilizar tecnologías avanzadas como Azure Data Factory contribuye a establecer infraestructuras digitales sólidas capaces de ajustarse a los cambios en las necesidades empresariales.

Además, al promover la eficiencia en la producción y consumo de datos, este proyecto ha apoyado el ODS 12. La automatización y optimización del proceso de entrega de datos entre plataformas en la nube disminuye el desperdicio de recursos y mejora su utilización, lo que puede llevar a prácticas más sostenibles en la gestión de datos y a una reducción del impacto ambiental. Finalmente, el logro del ODS 17 promueve la colabo-

ración y las alianzas estratégicas entre empresas al facilitar el intercambio de información en varias plataformas en la nube, fomentando así el trabajo en equipo para alcanzar objetivos compartidos relacionados con la gestión y transferencia de datos. Promueve la colaboración entre diferentes sectores empresariales y tecnológicos, lo que puede resultar en soluciones más efectivas y sostenibles a largo plazo.

Por último, los objetivos planteados por este proyecto han sido satisfactoriamente alcanzados, demostrando la viabilidad y beneficio de implementar un sistema de entrega de datos cross-cloud basado en Data Fabric entre Azure y AWS. Se ha logrado superar las barreras tradicionales de interoperabilidad en la nube gracias a un enfoque sistemático y estructurado, lo que ha resultado en una mejora significativa de la eficiencia operativa y la seguridad de los datos, contribuyendo al desarrollo sostenible y el avance tecnológico.

5.2 Recomendaciones

5.2.1 Recomendaciones para una Gestión Eficiente del Almacenamiento

Aunque se ha implementado la funcionalidad de mover archivos al directorio de Basura para mantener ordenado el almacenamiento, para una aplicación más eficiente y escalable del sistema en una organización con un mayor flujo de datos, se sugiere crear una capa de retención con una limpieza automática programada. Esta capa de retención permitirá que los archivos se guarden en un directorio durante un período de tiempo limitado, que puede variar dependiendo del volumen de datos que gestione la organización.

La implementación de esta capa de retención con limpieza automática programada garantizará una gestión eficiente del almacenamiento a largo plazo, eliminando automáticamente los archivos obsoletos según la política de retención definida. Esto no solo ayudará a optimizar el uso del espacio de almacenamiento, sino que también contribuirá a mantener el sistema ordenado y a reducir la posibilidad de acumulación de datos innecesarios.

BIBLIOGRAFIA

Satya Nadella: "Microsoft ofrece la nube más completa de la industria" | Diario TI. (s.f.). https://diarioti.com/satya-nadella-microsoft-ofrece-la-nube-mas-completa-de-la-industria/83857

Motor Adcreative ML/AI. (s.f.). https://es.adcreative.ai/post/all-you-need-to-know-about-adcreative-ai-ml-powered-advertisement-engine

IT Digital Media Group. (2023, July 19). *Este es el estado de la seguridad en cloud 2023*. Cloud | IT Digital Security. https://www.itdigitalsecurity.es/cloud/2023/07/este-es-el-estado-de-la-seguridad-en-cloud-2023

¿Qué es en verdad Big Data? Y por qué está cambiando el mundo. (2014a, May 12). ELEVA TU LÍMITE. https://elevatulimite.wordpress.com/2013/11/25/que-es-en-verdad-big-data-y-por-que-esta-cambiando-el-mundo/

Yang, M. (2021a, July 18). *Snowflake Data Sharing Architecture Part 1 of 4 - Introduction*. https://www.linkedin.com/pulse/snowflake-data-sharing-architecture-part-1-4-minzhen-yang/

elternativa. (2022, December 7). *Data Mesh vs Data Fabric: buscando la mejor arquitectura de datos*. www.elternativa.com. https://www.elternativa.com/data-mesh-data-fabric/

Laoyan, S. (2024, February 6). Qué es la metodología waterfall y cuándo utilizarla [2024] • Asana. *Asana*. https://asana.com/es/resources/waterfall-project-management-methodology

Ssabat. (2023, July 20). *How to send email - Azure Data Factory & Azure Synapse*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/data-factory/how-to-send-email