ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Máster en Big Data: Tecnología y Analítica Avanzada

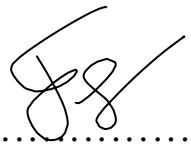
Diseño e Implantación de un Datalake y Capa de Consumo para Reporting Financiero

Autor
Fernando Carballeda Díez-Ponce

Dirigido por
Eduardo Ventas Maestre
Gabriel Iniesto Calabrés

Madrid
Mayo 2025

Fernando Carballeda Díez-Ponce, declara bajo su responsabilidad, que el Proyecto con título **Diseño e Implantación de un Datalake y Capa de Consumo para Reporting Financiero** presentado en la ETS de Ingeniería (ICAI) de la Universidad Pontificia Comillas en el curso académico 2023/24 es de su autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.



Fdo.: ..... Fecha: 12-05-2025

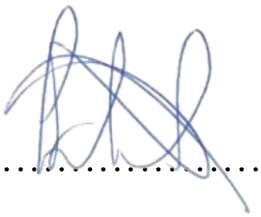
Autoriza la entrega:

LOS DIRECTORES DEL PROYECTO

Eduardo Ventas Maestre

Fdo.: ..... Fecha: 12-05-2025

Gabriel Iniesto Calabrés



Fdo.: ..... Fecha: 12-05-2025

V. B. DEL COORDINADOR DE PROYECTOS

Carlos Morrás Ruiz-Falcó

Fdo.: ..... Fecha: 12-05-2025

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D.Fernando Carballeda Díez-Ponce **DECLARA** ser el titular de los derechos de propiedad intelectual de la obra: Diseño e Implantación de un Datalake y Capa de Consumo para Reporting Financiero, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, los derechos de digitalización, de archivo, de reproducción, de distribución y de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

(a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.

(b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.

(c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.

(d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.

(e) Asignar por defecto a estos trabajos una licencia Creative Commons.

(f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

(a) Que la Universidad identifique claramente su nombre como autor de la misma

(b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.

(c) Solicitar la retirada de la obra del repositorio por causa justificada.

(d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

(a) El autor se compromete a:

(b) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.

(c) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.

(d) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

(e) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.

- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.

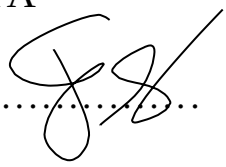
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.

- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 12 de mayo de 2025

ACEPTA

Fdo.: ……



Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Máster en Big Data: Tecnología y Analítica Avanzada

Diseño e Implantación de un Datalake y Capa de Consumo para Reporting Financiero

Autor
Fernando Carballeda Díez-Ponce

Dirigido por
Eduardo Ventas Maestre
Gabriel Iniesto Calabrés

Madrid
Mayo 2025

Resumen

Este proyecto describe la creación de un Datalake corporativo en favor de una entidad financiera y su posterior primer caso de uso: generación de informes personalizados para los clientes del propio banco.

Se escogió Snowflake como principal sistema de almacenamiento de datos y Talend como herramienta de integración y transformación de los datos.

Se diseñaron procesos ETL flexibles para garantizar la integridad y eficiencia de los datos, permitiendo manejar cambios en la estructura de los datos fuente sin rediseñar todo el proceso. El Datalake consta de varias capas, cada una con funciones específicas para organizar y procesar los datos de manera eficiente.

Los datos se transformaron, limpiaron y cargaron en la capa de consumo del Datalake. Posteriormente, se automatizó la generación un archivo JSON que sería enviado a una tercera entidad para la creación del informe final en formato PDF.

Abstract

This project describes the creation of a corporate Data Lake for a financial institution and its subsequent first use case: generating personalized reports for the bank's clients.

Snowflake was chosen as the main data storage system and Talend as the data integration and transformation tool.

Flexible ETL processes were designed to ensure data integrity and efficiency, allowing changes in the source data structure to be handled without redesigning the entire process. The Data Lake consists of several layers, each with specific functions to efficiently organize and process the data.

The data was transformed, cleaned, and loaded into the consumption layer of the Data Lake. Subsequently, the generation of a JSON file was automated and sent to a third party for the creation of the final report in PDF format.

Agradecimientos

Quiero expresar mi más sincero agradecimiento, en primer lugar, a Carlos Morrás Ruiz-Falcó, director del Máster en Big Data Tecnología y Analítica Avanzada de la Universidad Pontificia Comillas, ICAI, por su invaluable ayuda en la búsqueda de prácticas empresariales que me han permitido desarrollar este trabajo. Agradezco también a todos los docentes del programa por su labor de enseñanza y dedicación a los alumnos.

Asimismo, quiero agradecer a la familia de SDG Group, que durante estos meses de prácticas empresariales me ha acogido de manera impecable, incrementando mi entusiasmo por seguir trabajando junto a ellos. En especial, agradezco al equipo con el que trabajé durante casi cuatro meses, por sus enseñanzas y generosidad. Entre ellos, destaco a Eduardo Ventas Maestre y Gabriel Iniesto Calabrés, quienes me guiaron en este trabajo; y a Guillermo Cuenca Baciero, que se esforzó para que mi estancia fuera excelente.

Índice general

Índice de figuras

Índice de cuadros

Siglas y Acrónimos

API	Application Programming Interface
AWS	Amazon Web Services
BA	Business Analyst
BBDD	Bases de Datos
BI	Business Intelligence
DL	Data Lake
DM	Data Mart
DW	Data Warehouse
ELT	Extract, Load, Transform
ETL	Extract, Transform, Load
ETS	Escuela Técnica Superior
FK	Foreign Key
FW	Framework
HNWI	High-Net-Worth Individual
ICAI	Instituto Católico de Artes e Industrias
IoT	Internet of Things
JSON	JavaScript Object Notation
ML	Machine Learning
MSSQLS	Microsoft SQL Server
ORC	Optimized Row Columnar
PDF	Portable Document Format
PK	Primary Key
RDBMS	Relational DataBase Management System
S3	Simple Storage Service
SaaS	Software as a Service
SCD	Slowly Changing Dimension
SFTP	Secure File Transfer Protocol
SP	Stored Procedure
SQL	Structured Query Language
TI	Tecnología de la Información
URL	Uniform Resource Locator
XML	Extensible Markup Language

Capítulo 1

Introducción

1.1. Contextualización del problema

En la era de la digitalización y la transformación digital, la gestión eficiente de datos se ha convertido en un componente crítico para el éxito de las entidades financieras. Estas organizaciones se enfrentan al desafío de manejar grandes volúmenes de datos provenientes de diversas fuentes, tanto internas como externas, con el fin de mejorar sus procesos operativos, tomar decisiones informadas y ofrecer servicios personalizados a sus clientes. Sin una infraestructura de datos robusta y eficiente, las entidades financieras pueden enfrentar problemas de integridad de datos, retrasos en el procesamiento de información y dificultades para adaptarse a cambios en el entorno de datos.

1.2. Objetivos

El objetivo principal de este proyecto es diseñar e implementar un Datalake corporativo utilizando la plataforma Snowflake, apoyado en procesos ETL (Extract, Transform, Load) basados en metadatos y gestionados mediante la herramienta Talend. Este Datalake permitirá a la entidad financiera almacenar y procesar grandes volúmenes de datos de manera eficiente, facilitando la generación de reportes personalizados y mejorando la capacidad de análisis de datos. Los objetivos específicos incluyen:

1. Diseñar un Datalake corporativo que pueda manejar datos estructurados y no estructurados.
2. Implementar procesos ETL flexibles y dinámicos utilizando Talend.
3. Integrar datos de diversas fuentes para generar un reporte personalizado para los clientes.
4. Asegurar la integridad y la eficiencia en el manejo de los datos.

1.3. Estructura del trabajo

El trabajo está estructurado en dos secciones principales:

- La primera, dedicada al estado del arte, donde se analizarán en profundidad las diferentes tecnologías de gestión de datos y herramientas utilizadas en este proyecto.

- La segunda, dedicada al proyecto en sí, la cual se dividirá en dos subsecciones:
    1. La primera subsección hablará sobre el Data Lake (orígenes de los datos, diferentes capas, framework basado en metadatos, flujo de la información…).
    2. La segunda subsección estará enfocada en el primer caso de uso del Data Lake (necesidad del cliente, descripción del proceso, montaje del fichero JSON, recreación visual del informe…).

1.4. Motivación

La motivación detrás de este proyecto radica en la necesidad creciente de las entidades financieras de adaptarse a un entorno digital dinámico y altamente competitivo. La capacidad de gestionar grandes volúmenes de datos de manera eficiente no solo mejora la operatividad interna, sino que también permite ofrecer servicios de mayor calidad y personalizados a los clientes.

Implementar un Datalake corporativo en Snowflake, con procesos ETL basados en metadatos gestionados por Talend, representa un paso crucial para transformar la infraestructura de datos de la entidad financiera, potenciando su capacidad de análisis y mejorando su toma de decisiones estratégicas. Este proyecto busca demostrar cómo una solución de datos bien diseñada puede proporcionar beneficios tangibles y significativos, contribuyendo al éxito a largo plazo de la organización.

Destacar que la generación de informes personalizados para los clientes del banco no es el objetivo final de este proyecto, si no la primera puesta en práctica de la arquitectura de datos creada. Una vez finalizado este primer proyecto de caso de uso, se plantean muchas otras utilidades que se le pueden dar al Datalake, como análisis predictivos, desarrollo de dashboards, creación de alertas…

Capítulo 2

Estado del arte

2.1. Evolución DataWarehouse-DataMart-DataLake

¿Qué son?

Los DataWarehouses, DataMarts y DataLakes son diferentes soluciones de almacenamiento en la nube. Un Data Warehouse almacena datos en un formato estructurado. Se trata de un repositorio central de datos previamente procesados para llevar a cabo análisis y obtener inteligencia empresarial. Un DataMart es un almacenamiento de datos útil para las necesidades de un equipo o una unidad de negocios específico, como finanzas, marketing o ventas. Por último, un DataLake es un repositorio central que contiene datos sin procesar y no estructurados. Es posible almacenar datos y procesarlos posteriormente.

El término Data Warehouse fue el primero en escucharse, alrededor de la década de los 70. Posteriormente, en la década de los 90, surgió el concepto de DataMart, como una evolución del Data Warehouse. Ya a principios de la década de 2010, comenzó a ganar popularidad el término DataLake, con el aumento explosivo en cantidad y variedad de datos generados por las organizaciones.

Estas tres soluciones de almacenamiento ayudan a aumentar la disponibilidad, la fiabilidad y la seguridad de los datos.

DIFERENCIAS ENTRE DW Y DM

Orígenes de datos y tamaño

Un DataWarehouse tiene varios orígenes. Puede extraer datos de cualquier lugar, transformarlos en un formato estructurado y cargarlos en su almacenamiento.
Los DataMarts tienen menos orígenes de datos y su tamaño suele ser inferior.

Ámbito

Un DataWarehouse suele almacenar datos de varias unidades empresariales. Integran los datos de toda la organización de forma centralizada para que los análisis sean completos.
Los DataMarts se centran en un único asunto y tienen un carácter más descentraliza-

do. Suelen filtrar y resumir la información de otro almacenamiento de datos existente.

Uso

Un Data Warehouse suele tener una vida útil más larga y ser más complejos. Todos los miembros de la organización pueden ser usuarios del Data Warehouse.

Los DataMarts pueden centrarse en un proyecto y tener un uso limitado. Únicamente los miembros del departamento asociado al DataMart harán uso de este.

Diseño

Los científicos de datos usan un enfoque descendente al diseñar un DataWarehouse. Primero diseñan la arquitectura general y resuelven los desafíos a medida que surgen.

Con DataMarts, el ingeniero de datos ya conoce detalles como los valores, los tipos de datos y los orígenes de datos externos. Puede diseñar la implementación desde el principio y adoptar un enfoque de abajo a arriba para el diseño del DataMart.

	DW	**DM**
Ámbito	Centralizado, varias áreas de asuntos integradas juntas.	Descentralizado, área de asunto específica.
Usuarios	Toda la organización.	Un único departamento.
Origen de datos	Muchos orígenes.	Un único o pocos orígenes.
Tamaño	Grande, puede ser de cientos de gigabytes a petabytes.	Pequeño, generalmente de hasta decenas de gigabytes.
Diseño	De arriba hacia abajo.	De abajo hacia arriba.
Nivel de detalle de los datos	Datos completos y detallados.	Puede incluir datos resumidos.

Cuadro 2.1: Comparación entre Data Warehouse (DW) y Data Mart (DM)

DIFERENCIAS ENTRE DW Y DL

Los DataWarehouses almacenan datos estructurados, mientras que un DataLake es un repositorio centralizado que permite almacenar todos los datos a cualquier escala. Este último ofrece más opciones de almacenamiento, es más complejo y tiene diferentes casos de uso.

Orígenes de datos

Ambos pueden tener un número ilimitado de orígenes de datos.

Para el Data Warehouse es necesario diseñar un esquema antes de poder guardar los datos, ya que este solo puede cargar datos estructurados en el sistema.

Los DataLakes pueden almacenar datos no estructurados y semiestructurados.

Procesamiento previo

En los DataWarehouses, el procesamiento previo suele ser necesario antes del almacenamiento. Para ellos se utilizan herramientas de ETL.

Los DataLakes albergan cualquier tipo de datos, por lo que tienen la flexibilidad de decidir si se quiere llevar a cabo un procesamiento previo o no.

Calidad de los datos

Un DataWarehouse es más confiable, pues el procesamiento puede realizarse de antemano, aplicando funciones de deduplicación, clasificación, verificación o resumen para garantizar la precisión de los datos.

En un DataLake es posible encontrar datos erróneos, duplicados o sin verificar si no se comprueban previamente.

Rendimiento

El DataWarehouse se diseña para lograr un rendimiento de consultas más rápido.

La arquitectura de los DataLakes da prioridad al costo y al volumen de almacenamiento por encima del rendimiento. Obtiene mucho más volumen de almacenamiento a un costo reducido y sigue pudiendo acceder a los datos a velocidades razonables.

Estos tres tipos de almacenamiento en la nube son muy utilizados por la mayoría de las organizaciones grandes en su infraestructura de almacenamiento. La manera habitual de uso es que todos los datos lleguen a un DataLake para su posterior carga en distintos DataWarehouses o DataMarts.

Dependiendo de las necesidades de la compañía, el tipo y volumen de datos que tenga y la finalidad, esta tendrá la opción de elegir los tipos de almacenamiento que mejor se ajusten a sus proyectos y forma de trabajar.

En el caso particular de este proyecto, la opción escogida fue el DataLake, debido a la posibilidad de almacenar datos tanto estructurados como semiestructurados. [3]

	DW	DL
Datos	Datos relacionales provenientes de sistemas transaccionales, bases de datos operativas y aplicaciones de línea de negocio.	Todos los datos, incluidos los estructurados, los semiestructurados y los no estructurados.
Esquema	Se suele diseñar antes de la implementación del almacenamiento de los datos, aunque también se puede escribir a la hora del análisis.	Escrito al momento del análisis (esquema de lectura).
Precio/rendimiento	Resultados de búsqueda más rápidos con almacenamiento local.	Resultados de búsqueda más rápidos con almacenamiento de bajo costo y desacoplamiento de la informática y el almacenamiento.
Calidad de los datos	Datos muy mantenidos que funcionan como versión central de la verdad.	Cualquier dato que pueda estar mantenido o no (datos no procesados).
Usuarios	Analistas empresariales, científicos de datos y desarrolladores de datos.	Analistas empresariales, científicos de datos, desarrolladores de datos, ingenieros de datos y arquitectos de datos.
Análisis	Generación de informes en lotes, BI y visualizaciones.	Machine learning, análisis de exploración, descubrimiento de datos, streaming, análisis de operaciones, macrodatos y generación de perfiles.

Cuadro 2.2: Comparación entre Data Warehouse (DW) y Data Lake (DL)

2.2. Metadata-Driven

Un framework de integración de datos basado en metadatos (metadata driven) no es solo una combinación de tecnologías, sino un concepto de diseño que se basa en metadatos para gestionar e integrar datos de diversas fuentes. Los metadatos son datos que proporcionan información sobre otros datos. Incluyen información sobre la estructura de los datos, el tipo de los datos, las reglas de transformación. . . Estos metadatos se utilizan para generar mapeos o código dinámico que puede ser usado más tarde en el proceso de integración de datos.

Este repositorio de metadatos almacenará y gestionará todo el conocimiento empresarial.

Una arquitectura de metadatos es vital para una implementación de BI eficaz. Proporciona contexto para los elementos del almacén de datos, incluida información estructural sobre las relaciones entre entidades y los formatos de datos que componen la arquitectura del Data Warehouse. [4]

Ventajas sobre la integración de datos tradicional

- **Estandarización:** Proporciona un enfoque estandarizado para integrar datos de múltiples fuentes. Garantiza que todas las fuentes de datos se integren mediante un conjunto de reglas y estándares administrados, lo que mejora la calidad de los datos y reduce el riesgo de errores. La integración de datos tradicional, por otro lado, se basa en la codificación manual y puede generar inconsistencias y errores.

- **Reutilizabilidad:** Promueve un enfoque de crear una vez y reutilizar. Las asignaciones entre las estructuras de datos de origen y de destino se pueden reutilizar para futuros proyectos de integración de datos, lo que reduce aún más el tiempo y el coste de desarrollo. En un enfoque tradicional, los desarrolladores crean canales punto a punto que se adaptan a casos de uso específicos y, a menudo, no son reutilizables.

- **Automatización:** Automatiza varios pasos del proceso de integración de datos, eliminando la necesidad de realizar un mapeo manual. Utilizando administradores de metadatos avanzados, los desarrolladores pueden automatizar el mapeo de origen a destino. En este enfoque aumentado, el administrador de metadatos puede utilizar un clasificador/mapeador de datos automático impulsado por aprendizaje automático (ML) para analizar y comparar metadatos, datos, semántica, contextos, relaciones entre conjuntos de datos y predecir el mapeo de origen a destino junto con las reglas necesarias para transformar los datos de origen en los datos de destino deseados. Por el contrario, la integración de datos tradicional requiere que un analista de negocios (BA) perfile manualmente los datos, prepare el mapeo de origen a destino y luego los desarrolladores escriban código personalizado para cada mapeo de datos, lo cual es ineficiente y requiere mucho tiempo.

- **Flexibilidad:** Proporciona una solución flexible y escalable. Dado que se basa en la configuración, no es necesario codificar cada vez una nueva fuente o requisitos. Las empresas pueden agregar nuevas fuentes de datos y estructuras de datos a medida que cambian sus necesidades, y pueden escalar sus procesos de integración de datos para satisfacer sus crecientes necesidades de integración de datos. La integración de datos tradicional está limitada por las habilidades y la disponibilidad de los desarrolladores, lo que la hace menos flexible y escalable.

- **Mantenimiento más sencillo:** Es más fácil de mantener que la integración de datos tradicional. Se pueden realizar cambios en el proceso de integración de datos actualizando los metadatos, en lugar de modificar el código. Esto facilita la actualización y el mantenimiento del proceso de integración de datos a lo largo del tiempo.

- **Colaboración mejorada:** Promueve la colaboración entre desarrolladores y usuarios comerciales. Los usuarios empresariales pueden crear y gestionar metadatos sin necesidad de conocimientos de programación, lo que mejora la comunicación y la colaboración entre el departamento de TI y los usuarios empresariales.

Tres componentes de un Framework de Integración de Datos Basado en Metadatos

1. **Repositorio de metadatos.** Una base de datos centralizada que almacena metadatos sobre las fuentes de datos, las estructuras de los datos y las reglas de negocio. Este repositorio de metadatos almacena también mapeos entre las estructuras de datos de origen y de destino, reglas de orquestación, información de ejecución de Jobs... En nuestro caso, SDG Group cuenta con un framework de metadatos que contiene información sobre los datos necesarios para llevar a cabo este proyecto.

2. **Herramienta de gestión de metadatos.** Utilizada para crear, actualizar y gestionar los metadatos almacenados en el repositorio de metadatos. Debe proporcionar una interfaz de usuario intuitiva que permita a los usuarios empresariales o no técnicos crear y editar metadatos para el mapeo de origen a destino junto con transformaciones, orquestación, manejo de excepciones y reglas de validación de datos sin requerir ninguna habilidad de programación. Para este proyecto, el framework integrado se encuentra en el gestor de bases de datos PostgresSQL.

3. **Motor de integración.** Responsable de leer los metadatos del repositorio de metadatos y usarlos para realizar diversas acciones para integrar datos de diversas fuentes. El motor de integración utiliza las asignaciones almacenadas en el repositorio de metadatos para transformar los datos del formato de origen al formato de destino. Se escogió Talend Data Management como herramienta de integración para este proyecto. [6]

2.3. Cloud DWH - Snowflake

Los almacenes de datos en la nube representan una innovación fundamental en la gestión y análisis de datos para las organizaciones modernas. Estos sistemas ofrecen una solución integral y altamente eficiente para almacenar, procesar y analizar grandes volúmenes de datos en entornos de nube pública. Entre los diversos proveedores de almacenes de datos en la nube, Snowflake se destaca como una opción líder debido a su arquitectura única y su conjunto de características avanzadas.

2.3.1. Arquitectura de Snowflake

Snowflake se basa en una arquitectura de tres capas que consta de almacenamiento, computación y servicios de gestión. Esta arquitectura proporciona una separación completa entre el almacenamiento de datos y el procesamiento, lo que permite escalar horizontalmente la capacidad de computación según sea necesario, sin afectar el almacenamiento subyacente. Además, Snowflake utiliza un modelo de almacenamiento basado en columnas que optimiza el rendimiento y la eficiencia en la recuperación y procesamiento de datos.

2.3.2. Características clave de Snowflake

1. **Escalabilidad y rendimiento:** Snowflake ofrece una escalabilidad masiva que permite manejar cargas de trabajo de cualquier tamaño con facilidad. Su capacidad de escalar dinámicamente recursos de computación garantiza un rendimiento óptimo incluso durante los picos de demanda.
2. **Seguridad y cumplimiento:** Snowflake prioriza la seguridad y el cumplimiento normativo. Utiliza técnicas avanzadas de encriptación para proteger los datos en reposo y en tránsito, y ofrece funciones de control de acceso basado en roles para garantizar que solo los usuarios autorizados puedan acceder a los datos.
3. **Flexibilidad de precios:** Snowflake opera en un modelo de precios basado en el consumo, lo que significa que las organizaciones solo pagan por los recursos que utilizan. Esto elimina la necesidad de realizar inversiones costosas por adelantado y proporciona una mayor flexibilidad para ajustar los costos según sea necesario.
4. **Integración con herramientas de terceros:** Snowflake se integra fácilmente con una amplia gama de herramientas de análisis, visualización y desarrollo de terceros. Esto permite a las organizaciones aprovechar al máximo su inversión en tecnología existente y simplificar el proceso de migración a Snowflake.
5. **Soporte para cargas de trabajo múltiples:** Snowflake es capaz de soportar una variedad de cargas de trabajo, incluyendo análisis ad hoc, procesamiento de datos en tiempo real, aplicaciones de inteligencia artificial y aprendizaje automático, entre otros. Esto lo hace adecuado para organizaciones de todos los tamaños y sectores.
6. **Almacenamiento de datos estructurados y semiestructurados:** Snowflake permite almacenar tanto datos estructurados como semiestructurados en una única plataforma. Utiliza tipos de datos como VARIANT, OBJECT, y ARRAY para manejar formatos como JSON, Avro, Parquet, ORC y XML, lo que facilita la consolidación de diferentes tipos de datos y su análisis en conjunto. [5]

2.4. Contexto Banca Privada

La banca privada es una división especializada de los bancos que se dedica a brindar servicios financieros y de gestión de patrimonio a clientes de alto valor neto (high-net-worth individual (HNWI)) y a familias adineradas. Estos clientes generalmente tienen un nivel de riqueza significativo y requieren servicios personalizados y soluciones financieras adaptadas a sus necesidades específicas. Los servicios ofrecidos por la banca privada incluyen:

- **Gestión de inversiones.** Proporciona servicios de gestión de inversiones personalizados, teniendo en cuenta los objetivos financieros y la tolerancia al riesgo de cada cliente. Esto implica la creación y gestión de una cartera de inversiones diversificada que puede incluir acciones, bonos, fondos mutuos, fondos de cobertura, bienes raíces y otros activos.

- **Planificación patrimonial.** Ayuda a los clientes a administrar y preservar su patrimonio a largo plazo. Esto incluye asesoramiento sobre impuestos, sucesión, planificación de herencias, estructuración de fideicomisos y planificación financiera integral para las futuras generaciones.

- **Servicios bancarios personalizados.** Los clientes de la banca privada tienen acceso a una amplia gama de servicios bancarios personalizados, que incluyen cuentas corrientes y de ahorro, tarjetas de crédito y débito, préstamos y líneas de crédito, servicios de banca en línea y servicios de banca internacional.

- **Asesoramiento financiero.** Proporciona asesoramiento financiero especializado a sus clientes en áreas como la diversificación de inversiones, la gestión del riesgo, la planificación de jubilación, la educación de los hijos, la filantropía y otros aspectos relacionados con la gestión del patrimonio.

- **Servicios de gestión fiduciaria.** También ofrecen servicios de gestión fiduciaria, actuando como fiduciarios o administradores de fondos para proteger y gestionar los activos de sus clientes, a menudo a través de fideicomisos y estructuras legales similares.

La principal diferencia con la banca minorista tradicional es que brinda servicios altamente personalizados y exclusivos a clientes con un patrimonio significativo. Estos clientes esperan una atención individualizada, soluciones a medida y un alto nivel de confidencialidad en sus transacciones financieras. Generalmente tienen una cantidad considerable de activos y una capacidad de inversión significativa. No existe un umbral específico para ser considerado cliente de banca privada, ya que varía según la institución financiera, pero suele requerirse un nivel mínimo de riqueza o ingresos para acceder a estos servicios.

Banco Cliente

El banco cliente para el que va dirigido este proyecto es reconocido como una de las mejores entidades de banca privada en 2024. Con cerca de 13.500 clientes y 13 oficinas en España, su patrimonio está cerca de los 13.500 millones de euros.

Esta banca privada ofrece a sus clientes tres grados de asesoramiento diferentes en función de los conocimientos que se tengan. Así, permite el acceso a toda su gama de productos para que sea el cliente el que construya su cartera a su gusto; también ofrece una opción mixta en la que la cartera se crea entre el banquero privado y el propio cliente y, por último, también ofrece la construcción de carteras por parte de un banquero privado en función al perfil de riesgo del cliente.

El patrimonio mínimo necesario para ser cliente de esta entidad de banca privada es de un millón de euros. [8]

Capítulo 3

Proyecto

3.1. Data Lake

3.1.1. Orígenes de datos

Se trabaja con dos tipos de orígenes de datos:

- **BBDD (Bases de Datos):**

  La entidad financiera cuenta con bases de datos donde almacenan información sobre diferentes ámbitos. Entre ellas encontramos:

  - **MARKETING:** Tiene un gran número de tablas, principalmente relacionadas con datos de empleados y clientes.
  - **BACKOFFICE:** Tablas con información generada en la operativa diaria del banco como oficinas y departamentos del mismo.
  - **DATALYTICS:** Tablas con información maestro para el análisis de documentos financieros.
  - **VISIONCREDIT:** Varias tablas relacionadas con hipotecas y clientes hipotecarios.
  - **INFOCIS:** Información sobre el inventario de productos
  - **ANALYTICS:** Tablas relacionadas con operaciones y rentabilidad.

- **Ficheros:**

  Además, cuenta con ficheros provenientes de diferentes proveedores externos. Algunos son:

  - **CADIT**
  - **FINAMETRIX**
  - **SALESFORCE**
  - **DXC**
  - **AXA**
  - **CASER**
  - **INVERSIS**

  Cada una de estas empresas proporciona a la entidad financiera ficheros con información sobre seguros, gestión de activos...

3.1.2. Capas del DataLake

La arquitectura propuesta para la creación del DataLake está compuesta por varias capas o zonas que se encuentran en diferentes entornos y que permiten el movimiento y su consecuente transformación de los datos, todo con la finalidad de disponer de una última capa en la que los datos estén procesados para su utilización.

Capa Bronze

La primera capa será la denominada Capa Bronze. Estará en un bucket de S3 y a ella llegarán datos provenientes de los sistemas operacionales y proveedores externos del banco. S3 almacena "objetos", que son unidades básicas de datos compuestas por una clave y datos. Los ficheros de origen provenientes de proveedores externos se almacenan en el mismo formato de origen en este bucket de S3, pero las tablas pertenecientes a las bases de datos de la entidad financiera, son convertidas a ficheros para poder almacenarse en S3, pues este no es un RDBMS (Relational DataBase Management System). Se ha diseñado una estructura de directorios para poder identificar los sistemas origen o proveedores de cada tipo de información. En la Figura 3.1 podemos ver esta estructura.

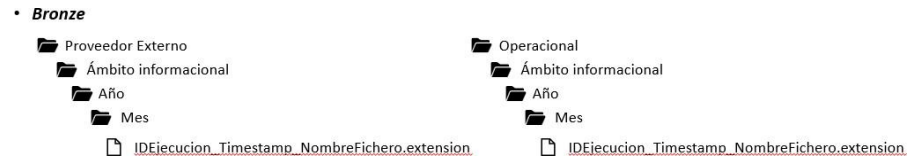


Figura 3.1: Estructura de directorios

Capa Silver

Posteriormente, los datos pasarán de la Capa Bronze a la Capa Silver, que estará contenida en Snowflake. En esta Capa Silver habrá tres subcapas: **Staging**, **Temporal** y **Persistent**. En la subcapa Staging llegarán los datos directamente desde el bucket de S3 (Capa Bronze) mediante la ejecución de maestros que se reajustan en función de unos datos de entrada. Después se generarán tablas temporales (Temporal) donde se aplicarán reglas de calidad a los datos. Estas reglas de calidad, diseñadas como plantillas SQL, están almacenadas en una base de datos aparte de la Capa Silver, denominada **Control Process**, en Snowflake. Los datos que cumplan esas reglas de calidad pasarán a la tercera subcapa de la Capa Silver, denominada Persistent. Los datos que no cumplan con estas reglas de calidad, irán a parar a la denominada Error Zone, que se encuentra en un bucket de S3.

Capa Gold

Los datos que lleguen a la subcapa Persistent de la Capa Silver, y que por lo tanto hayan cumplido con las reglas de calidad, pasarán finalmente a la Capa Gold, contenida también en Snowflake y donde la información se almacena en diferentes tablas y vistas pertenecientes a varios modelos de datos.

Podemos ver en la Figura 3.2 un esquema de esta arquitectura.

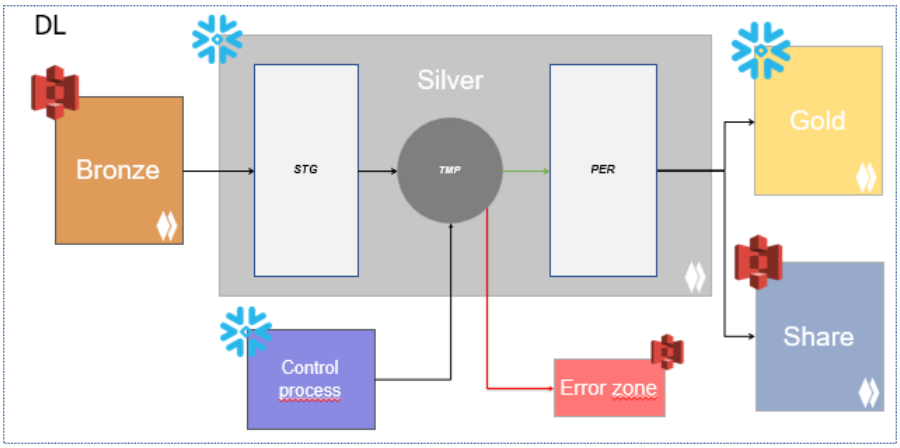


Figura 3.2: Diferentes capas del DataLake

3.1.3. ELT vs ETL

Definición

El proceso de extracción, transformación y carga (ETL) y el de extracción, carga y transformación (ELT) son dos enfoques de procesamiento de datos para el análisis (secuencias de procesos que preparan los datos para su posterior análisis). En el contexto del BigData, las empresas u otras organizaciones tienen que filtrar, clasificar y limpiar un gran volumen de datos para que sea útil para el análisis y la inteligencia empresarial. El enfoque de ETL utiliza un conjunto de reglas para procesar datos de varios orígenes antes de la integración centralizada. El enfoque de ELT carga los datos tal como están y los transforma en una etapa posterior, según el caso de uso y los requisitos de análisis. El proceso ETL requiere más definición al principio. Se debe incluir el análisis desde el principio para definir los tipos de datos de destino, las estructuras y las relaciones. Los científicos de datos utilizan principalmente ETL para cargar bases de datos heredadas en el almacenamiento de datos, y el proceso de ELT se ha convertido en la norma en la actualidad.

- **Extracción.** Recopilación de datos sin procesar de diferentes orígenes (BBDD, archivos, SaaS, sensores de IoT. . .). Puede recopilar datos semiestructurados, estructurados o no estructurados.
- **Transformación.** Cambio de los datos sin procesar de su estructura original a un formato que cumpla con los requisitos del sistema de destino en el que planea almacenar los datos para su análisis. Algunos ejemplos de transformación:

  1. Cambio de tipos o formatos de datos
  2. Eliminación de datos incoherentes o inexactos
  3. Eliminación de la duplicación de datos

  Se aplican reglas y funciones para limpiar y preparar los datos para su análisis en el sistema de destino.

- **Carga.** Almacenamiento de los datos en la base de datos de destino. El proceso de ETL procesa los datos de carga como paso final, de modo que las herramientas

de informes puedan utilizarlos directamente para generar informes e información procesables. El proceso ELT sigue siendo necesario transformar los datos extraídos después de cargarlos.

Diferencias entre ETL y ELT

Tres pasos en el proceso ETL:

1. Extracción de datos sin procesar de varios orígenes.
2. Uso de un servidor de procesamiento secundario para transformar esos datos.
3. Carga de datos en una base de datos de destino.

La etapa de transformación garantiza el cumplimiento de los requisitos estructurales de la base de datos de destino. Los datos solo se mueven una vez que estén transformados y listos.
Tres pasos en el proceso ELT:

1. Extracción de datos sin procesar de varios orígenes.
2. Carga de los datos en su estado natural en un almacenamiento o lago de datos.
3. Transformación de los datos según sea necesario mientras estén en el sistema de destino.

Con ELT, toda la limpieza, la transformación y el enriquecimiento de los datos se producen dentro del almacenamiento de datos. Puede interactuar con los datos sin procesar y transformarlos tantas veces como sea necesario.

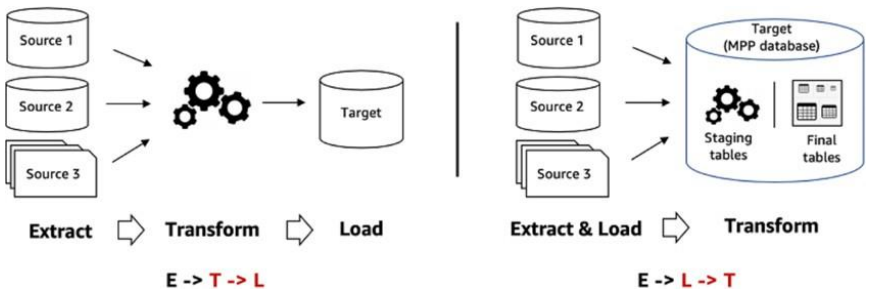


Figura 3.3: Diferencias entre ETL y ELT

La evolución de las tecnologías en la nube permitió que las empresas pudieran almacenar datos sin procesar ilimitados a escala y analizarlos posteriormente según fuera necesario. Así surgió el proceso ELT, que se convirtió en el método moderno de integración de datos para un análisis eficiente. Podemos ver en la Figura 3.3 un esquema con las diferencias entre estos dos procesos.

Ventajas de ELT sobre ETL

- **Más rápido.** El proceso ELT carga los datos directamente en el sistema de destino. Utiliza la potencia de procesamiento que ofrecen los almacenamientos de datos en la nube para ofrecer una transformación de datos en tiempo real (o casi).

- **Menos costoso.** El proceso ELT tiene menos sistemas que el de ETL, ya que todas las transformaciones se producen dentro del almacenamiento de datos de destino. Además, no es necesario que los analistas planifiquen con antelación la estructura y formato de los datos, lo cual disminuye los costos.

- **Mayor compatibilidad de datos.** El proceso ELT gestiona todo tipo de datos, incluidos los datos no estructurados, como imágenes o documentos, que no se pueden almacenar en formato tabular. Permite cargar estos datos en el almacenamiento de datos de destino y allí transformarlos en el formato que necesite. [2]

¿Por qué ELT al usar Snowflake?

En el contexto de Snowflake, realizar una ETL podría requerir el uso de herramientas de transformación de datos fuera de la plataforma, lo que podría aumentar la complejidad y el tiempo de procesamiento.

Por el contrario, si el proceso que elegimos en un proceso ELT, la transformación de los datos se realiza dentro de la plataforma de Snowflake, utilizando sus capacidades de procesamiento masivo y paralelo y su posibilidad de escalado horizontal. Snowflake está optimizado para realizar transformaciones complejas de datos a gran escala de manera eficiente, lo que puede hacer que el proceso de transformación sea más rápido y menos complejo en comparación con realizar transformaciones fuera de la plataforma. [7]

Pushdown

Pushdown es una de las alternativas recomendadas para la carga de base de datos de un Data Warehouse. Consiste en trasladar todo el procesamiento posible a la base de datos. Toda la lógica queda "pintada" en la ETL, no obstante, el procesamiento se deriva al servidor de base de datos. En nuestro caso, la herramienta ETL utilizada es Talend. En Talend estará toda esa lógica del proceso ETL, mientras que el procesamiento tendrá lugar en Snowflake. (Talend organiza el plan, Snowflake lo lleva a cabo). Una de las ventajas que tiene realizar pushdown es que los datos siempre están alojados en el mismo sitio durante su transformación. La privacidad de los datos no abandona el perímetro de red del cliente y los datos sensibles son procesados por un único servidor (Snowflake). [9]

3.1.4. Talend

La integración de datos es el proceso de fusionar datos de varias fuentes. Este proceso comienza por el mapeo, ingesta, limpieza y transformación de los datos, para el posterior uso de la persona que accede a ellos.

Talend ofrece herramientas potentes de integración de datos para realizar procesos ETL.

¿Qué es Talend Data Management?

Es una de las herramientas de integración de datos más potentes disponibles en el mercado. Talend Data Management ayuda a llevar a cabo los diferentes pasos en un proceso ETL de manera sencilla, comenzando por la definición básica de proceso hasta la

ejecución de este.

Cuenta con una interfaz gráfica de usuario que permite crear flujos de datos desde la ubicación de origen hasta la ubicación de destino. La forma principal de crear "Jobs" o flujos de datos es arrastrar diferentes componentes de un amplio catálogo a un lienzo y conectarlas de manera adecuada para que los datos viajen y se transformen como consideremos necesario (Figura 3.4). Las componentes de Talend son numerosas y están explicadas en su documentación. En este trabajo describiremos sólo las utilizadas. [11]

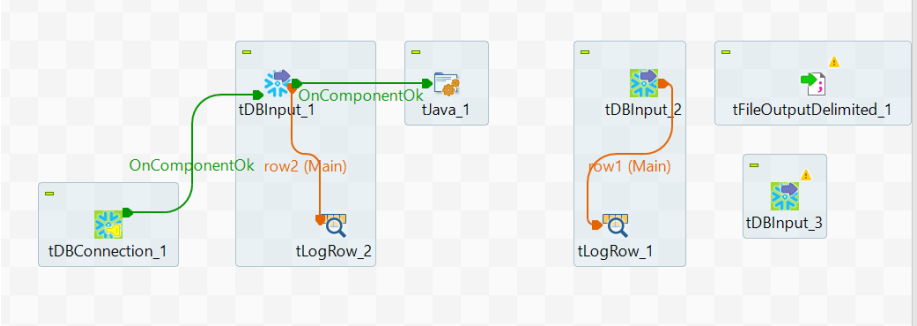


Figura 3.4: Talend Data Management

Estas componentes nos permiten realizar acciones como:

- Extracción de datos
  - Conexión a diferentes fuentes
- Transformación de datos
  - Limpieza de los datos
  - Agregación de datos
  - Filtrado de los datos
  - Conversión de formato
  - Validación
- Carga de datos
  - Conexión a bases de datos para añadir o modificar datos
  - Creación de ficheros
- Programación de tareas
  - Ejecución de Jobs en momentos específicos y con una repetición específica
  - Envío de correos electrónicos de confirmación o error

¿Por qué Talend Data Management?

- **Interfaz Gráfica Intuitiva.** Esto permite a los desarrolladores diseñar, configurar y ejecutar flujos de trabajo de ETL o ELT de manera visual, lo cual facilita la creación de procesos complejos de integración de datos sin necesidad de escribir código manualmente. A la hora de debugear un proceso ayuda a la comprensión funcional visualmente.

- **Amplia Variedad de Conectores.** Estos están predefinidos para interactuar con diversas fuentes y destinos de datos, lo que simplifica la integración con diferentes sistemas y aplicaciones.
- **Capacidades de Transformación de Datos.** Ofrece una gran variedad de componentes que ayudan al cocinado de la información.
- **Gestión y Monitoreo de Procesos.** Dispone de funcionalidades que facilitan la supervisión del rendimiento y la ejecución de los flujos de trabajo de ETL.
- **Integración con Snowflake.** Ofrece conectores específicos para interactuar con Snowflake, lo que simplifica la carga de datos hacia y desde esta plataforma de almacenamiento en la nube. Estos conectores están diseñados para aprovechar las capacidades de Snowflake y garantizar un rendimiento óptimo en el proceso de ETL. [10]

3.1.5. Metadata-Driven: Framework

El Framework de metadatos desarrollado por SDG Group es un **modelo de metadatos** que permite la gestión, monitorización y gobierno centralizado de los flujos de datos involucrados en los procesos de integración y transformación en una ETL.

Jerarquía en la ETL

Al utilizar Talend, un job puede llamar a otro job, este segundo job puede llamar a más jobs y así de forma indefinida. Sin embargo, al usar Talend con el Framework, este nos impone una jerarquía en la ETL, lo que significa que asociaremos los Jobs de Talend a distintos niveles de jerarquía impuestos por el Framework. Estos niveles son (ordenados de mayor a menor jerarquía):

- **Proyecto.** Nivel de jerarquía más alto. Toda la ETL se orquesta desde el job que representa esta jerarquía en Talend. Un proyecto está formado por fases (ingestar datos en Silver, pasar datos a Gold. . .). Habrá un único job que ejecute todo el proyecto. Este job llamará a otros Jobs que representarán distintas fases. La parametrización de los proyectos se realiza en la tabla de definición ETL llamada fw_dim_proyectos.
- **Fase.** Está formada por procesos. Normalmente se asocia a una zona de persistencia (Extracción, Silver Staging...). La parametrización de las fases se realiza en la tabla de definición ETL llamada fw_dim_fases. Habrá un job de Talend para cada fase, que llamará a distintos Jobs que representarán procesos.
- **Proceso.** Conjunto de subprocesos que se asocian a un ámbito informacional (Marketing, Backoffice, Saldos. . .). La parametrización de los procesos se realiza en la tabla de definición ETL llamada fw_dim_procesos. Para cada proceso existe un job en Talend que llamará a los diferentes subprocesos.
- **Subproceso.** Operaciones sobre un conjunto de datos. Suelen tener la lógica de cargas y transformaciones llevadas a cabo en la ETL. La parametrización de los subprocesos se realiza en la tabla de definición ETL llamada fw_dim_subprocesos. En ocasiones un subproceso se encarga de ejecutar otro job que suele ser un maestro.

Esto ocurre cuando la lógica del subproceso es generalizable y se encapsula en este job maestro. También se añaden Jobs maestros a los subprocesos para tener mayor control sobre la captura de errores y sobre las transformaciones llevadas a cabo. Estos Jobs maestros no se parametrizan en ninguna tabla del framework.

El modelo está formado por distintas tablas relacionadas entre sí en un esquema de PostgreSQL. En base a su utilidad funcional, estas tablas pueden clasificarse en:

- **Tablas de Definición ETL.** Contienen los metadatos que definirán qué scripts (jobs) se pueden ejecutar en una ETL y con qué orden y qué dependencias unos de otros. En estas tablas el usuario deberá realizar operaciones de insertar o modificar datos para controlar cómo funciona la ETL. El usuario modificará estas tablas cuando sea necesario modificar o evolucionar la ETL (incluir nuevos orígenes, realizar transformaciones...). Entre estas tablas se encuentran:

  1. **fw_dim_proyectos:** Metadatos sobre el estado de los proyectos.
  2. **fw_dim_fases:** Metadatos sobre el estado de las fases.
  3. **fw_dim_procesos:** Metadatos sobre el estado de los procesos.
  4. **fw_dim_subprocesos:** Metadatos sobre el estado de los subprocesos.
  5. **fw_dim_ingestas:** Metadatos sobre las ingestas desde los orígenes de datos a S3.
  6. **fw_dim_alertas_email:** Parametrización de las alertas por email.
  7. **fw_dim_ambitos_ejecucion:** Metadatos sobre los ámbitos de ejecución.

- **Tablas de definición de estados.** Especifican y definen los estados a los que hacen referencia las tablas de definición de ETL. Actúan a modo de tablas de referencia y estarán relacionadas con las tablas de definición de ETL y las tablas de registro de ejecuciones mediante foreign key. Estas tablas también serán rellenadas por el usuario, pero en este caso la ingesta de datos será un one-shot, es decir, se rellenarán solo una vez y no será necesario volver a modificarlas. Entre estas tablas se encuentran:

  1. **fw_dim_estados_job:** Metadatos sobre el estado de los jobs.
  2. **fw_dim_estados_ejecucion:** Metadatos sobre los estados de la ejecución.

- **Tablas de registro de ejecuciones.** Se encargan de monitorizar la ejecución de la ETL. Estas tablas no son modificadas por el usuario, sino que será la propia ETL la que insertará datos asociados a las ejecuciones en estas tablas, es decir, una vez se ejecute un proyecto, estas tablas serán modificadas de manera automática por la ETL. Entre estas tablas se encuentran:

  1. **fw_ejecuciones:** Información sobre las ejecuciones a nivel general. Un registro por ejecución.
  2. **fw_ejecuciones_activas:** Información sobre las ejecuciones activas donde solo se indica si están activas y si han tenido algún error.
  3. **fw_ejecuciones_mensajes:** Información sobre los mensajes generados por las ejecuciones a través de sentencias catch.

4. **fw_ejecuciones_trazabilidad:** Información sobre la información movida en cada proyecto ETL (ficheros, BBDD, número de registros…).
5. **fw_ejecuciones_detalle:** Información detallada sobre el estado de cada ejecución. El detalle se encuentra a nivel de subproceso. Un registro por subproceso de cada proyecto.
6. **fw_ejecuciones_validaciones:** Resultados de las validaciones/reglas de calidad.

En la Figura 3.5 podemos ver un esquema con las tablas del Framework y sus relaciones.

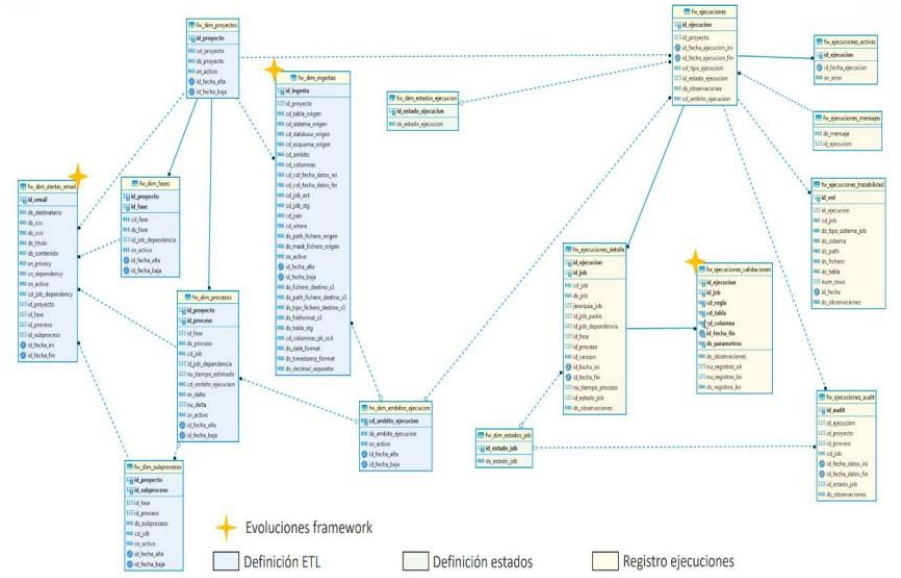


Figura 3.5: Modelo de metadatos del Framework

Integración en la ETL

La integración del Framework en la ETL se realiza a través de Talend, que es utilizado como herramienta de orquestación de la ETL. Es necesario entender por tanto que el Framework está desarrollado para interactuar con Talend y el modelo de metadatos desarrollado está condicionado en parte por las particularidades de Talend. La forma de la ETL de interactuar con el modelo de metadatos es a través de una serie de jobs maestros que se encargan de:

1. Comprobar que el job a ejecutar está registrado en la estructura de proyecto, fase, proceso y subproceso. En caso negativo generar un error.
2. Registrar al principio y al final de la ejecución de cada job su progreso.
3. Registrar cualquier tipo de movimiento de datos o mensaje de error.

Nomenclatura Framework

En general los nombres de las columnas del framework se escriben en español. Cada columna puede contener un prefijo que aporta un significado:

- **id_:** Código numérico que actúa como identificador. En muchos casos va a ser un incremental, cuyo incremento dependerá de la jerarquía (proyecto, fase, proceso, subproceso).
- **cd_:** Código no número que identifica algo.
- **ds_:** Campo descriptivo normal.
- **num_:** Número o cantidad.
- **sn_:** Campo booleano que se guarda como S o N en caso de que sea verdadero o falso respectivamente.

3.1.6. Maestros

Antes de explicar los jobs maestros de Talend, es importante entender cómo funciona una de las tablas más importantes del Framework. Esta tabla es **fw_dim_ingestas**. Se encuentra dentro del grupo de Tablas de Definición ETL y contiene la información sobre las ingestas desde los orígenes de datos hasta S3 (Capa Bronze) y bajada hasta Silver Staging. En esta tabla se parametrizan tanto los orígenes de datos del tipo BBDD como del tipo fichero. La extracción se realiza en cada sistema de origen correspondiente, y se generan ficheros que se suben a S3. Las columnas de esta tabla son:

- **id_ingesta**. Primary Key identificador de la Ingesta. Autonumérico.
- **id_proyecto**. Foreign Key identificador del proyecto.
- **cd_tabla_origen**. Campo usado para orígenes procedentes de BBDD. Designa el nombre de la tabla de origen.
- **cd_sistema_origen**. Tipo de sistema origen. Toma valores como MSSQLS, SFTP, SHARE...
- **cd_database_origen**. Base de datos para orígenes BBDD.
- **cd_esquema_origen**. Esquema para orígenes BBDD.
- **cd_ambito**. Foreign key de la tabla fw_dim_ambitos_ejecucion. Indica el ámbito funcional de la ejecución.
- **cd_columnas**. Columnas a extraer de los orígenes. Para ficheros delimitados se especifican las columnas separadas por ",". Para ficheros de ancho fijo toma el valor SINGLECOLUMN.
- **cd_col_fecha_datos_ini**. Fecha inicio de validez de los datos.
- **cd_col_fecha_datos_fin**. Fecha fin de validez de los datos.
- **cd_job_ext**. Nombre del job de extracción.
- **cd_job_stg**. Nombre del job que realiza el paso a staging.
- **cd_join**. Columna tipo texto donde se encuentra la sentencia para hacer joins a las tablas.

- **cd_where**. Columna tipo texto donde se encuentra la sentencia para hacer wheres a las tablas. Por defecto se toma todo, por tanto, el valor es 1=1.
- **ds_path_fichero_origen**. Columna para especificar el path a los orígenes del tipo fichero.
- **ds_mask_fichero_origen**. Máscara a usar para seleccionar ficheros procedentes del path de origen.
- **sn_activo**. Indiciador de si un origen está activo o no. Toma valores S/N.
- **id_fecha_alta**. Fecha en la que se da de alta un origen en la tabla fw_dim_ingestas.
- **id_fecha_baja**. Fecha en la que se da de baja un origen en la tabla fw_dim_ingestas.
- **ds_fichero_destino_s3**. Contiene el formato de la nomenclatura que debe tener el fichero en S3.
- **ds_path_fichero_destino_s3**. Columna que compone la ruta absoluta que forma parte del path del fichero en S3.
- **ds_tipo_fichero_destino_s3**. Tipo de fichero en S3. En función de este tipo el tratamiento en el paso a stg es distinto.
- **ds_fileformat_s3**. Parametrización de los parámetros usados por los jobs de staging para ejecutar la sentencia COPY INTO que se encarga de pasar la información del fichero en S3 a formato tabla en Snowflake.
- **ds_tabla_stg**. Nombre de la tabla a la que bajará la información en staging.
- **cd_columnas_pk_scd**. Conjunto de columnas que componen la PK para hacer uso de los mecanismos de control de vigencia.
- **ds_date_format**. Formato de las fechas para ejecutar el casteo de string a date correctamente.
- **ds_timestamp_format**. Formato del timestamp para ejecutar el casteo de string a timestamp correctamente.
- **ds_decimal_separator**. Caracter que se usa como separador decimal.

Para que las tablas o ficheros viajen desde el origen en el que se encuentran a la capa de consumo final, se cuenta con una serie de Jobs de Talend a los que se denominan Maestros. Siguiendo la jerarquía en la ETL descrita anteriormente, se hará un barrido de cómo se llega a ejecutar uno de estos maestros.

En la Figura 3.6 podemos ver un ejemplo de un job Proyecto. En la parte de abajo se observan siete SubJobs, que serían las diferentes fases del proyecto. Estas fases son: Inicio, Extracción, Staging, Temporal, Error, Persistent y Fin. Cada una de estas fases se encarga del movimiento de las tablas o ficheros de una zona a otra. Por ejemplo, la fase EXT se encarga de llevar los datos de los diferentes orígenes a la Capa Bronze, es decir, al bucket de S3. Veamos como es una de estas fases.

Si observamos la Figura 3.7 podemos ver que dentro de la fase de extracción se encuentran diferentes procesos (Maestros Backoffice, Maestros Marketing, Maestros Datalytics,

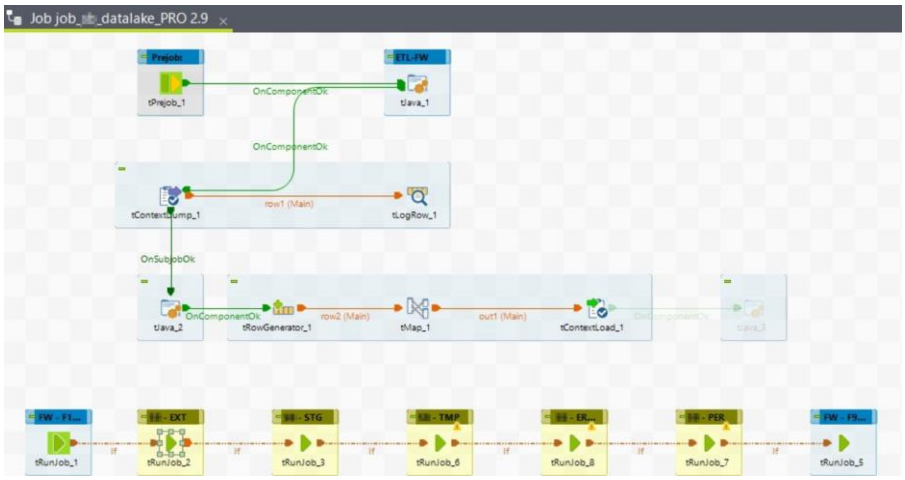


Figura 3.6: Job Proyecto

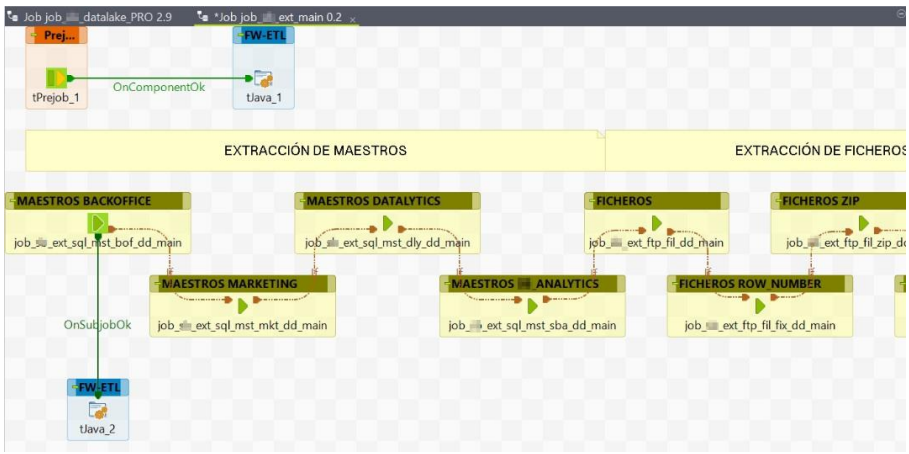


Figura 3.7: Job Fase Extracción

Ficheros...). Cada uno de estos procesos estará enfocado en la extracción y carga de datos en S3 de una base de datos o sistema de ficheros.

En la Figura 3.8 podemos ver qué contiene el proceso de Maestros Marketing. Esa componente de color amarillo que vemos en el centro del lienzo es el subproceso. Ese subproceso llamará al job maestro necesario para extraer desde origen hacia la capa de bronce los archivos (en este caso tablas) de la base de datos Marketing. En la Figura 3.9 podemos ver cómo es el job asociado al subproceso y cómo hace una llamada al job maestro que se encarga de ingestar los datos de las bases de datos de origen a la capa de bronce.

Podemos acceder a este job maestro que sigue los siguientes pasos. Primero, realiza una conexión a PostgreSQL para acceder a la tabla del Framework fw_dim_ingestas (Figura 3.10). Una vez conectado, hace un SELECT de las columnas de la tabla fw_dim_ingestas con información necesaria para realizar la extracción (base de datos, esquema, tabla, path de fichero de destino, job de extracción...) filtrando por el tipo de origen del que se quieren extraer los datos (en este caso MARKETING). Esta query dará como resultado un filtrado de la tabla fw_dim_ingestas donde cada registro hará referencia a los parámetros necesarios para la ingesta de cada una de las tablas de la base de datos de Marketing. Mediante una iteración sobre los registros obtenidos, se crearán las querys de creación de

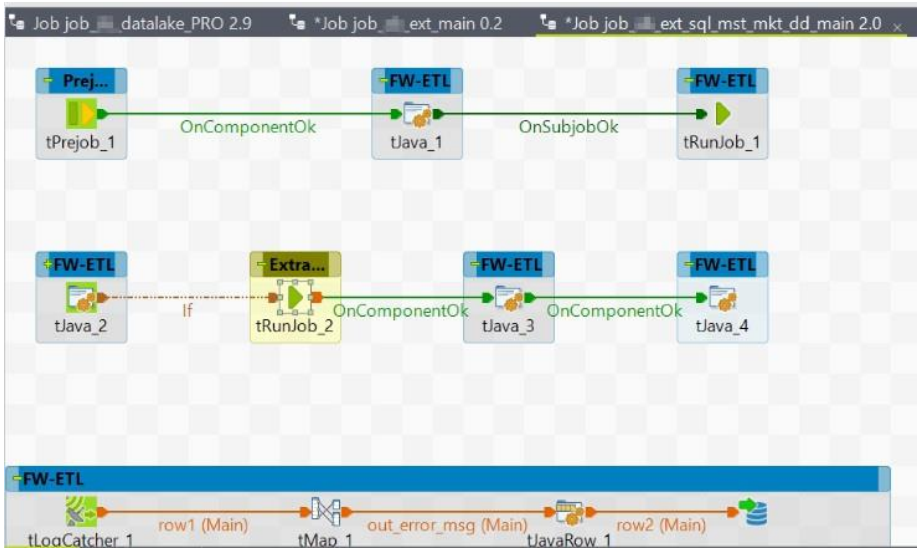


Figura 3.8: Job Proceso Maestros Marketing

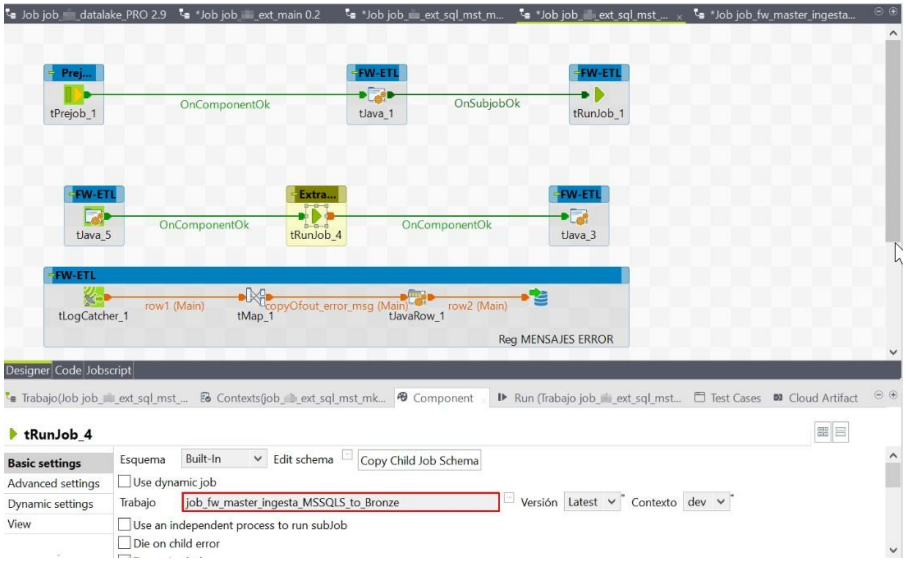


Figura 3.9: Job Subproceso

ficheros en S3 que contengan la información de cada tabla. Después, ejecuta esas consultas y crea esos ficheros (uno para cada tabla) como se puede ver en la Figura 3.11. Por último, se conecta al bucket de S3, donde copia esos ficheros, para finalmente eliminarlos del servidor que ejecuta la ETL a modo de política de housekeeping. Una vez finalizado este proceso, los datos de los orígenes han sido copiados al bucket de S3 (Capa Bronze). En caso de realizar la ingesta de un fichero que proviene de almacenamiento de ficheros (SFTP), una vez procesado se mueve el fichero a una carpeta llamada OLD, para evitar que al hacer la siguiente ingesta estos datos vuelvan a entrar en el proceso, lo que significaría datos repetidos. La ingesta de datos provenientes de ficheros es similar al caso descrito para bases de datos. En ese caso no habrá base de datos o esquema en el origen, si no que habrá un file_path donde se encuentre el fichero. Además, se pueden dar casuísticas donde un fichero necesite ser descomprimido, o del que haya que seleccionar únicamente algunas columnas específicas. El utilizar jobs maestros junto con el Framework nos ayuda a que el proceso de ingesta de datos sea parametrizable y mucho más sencillo.

Como se puede ver, en todos los jobs existe un prejob que ejecuta un componente llamado tRunJob_1 encargado de registrar el inicio del proceso y comprobar que está definido en el Framework para poder ejecutarse. A su vez, existe un apartado de captura de errores (tLogCatcher_1) que captura y registra los diferentes errores que se puedan dar a lo largo del proceso. Pese a no verse en la imagen hay un componente de postjob análogo al prejob que registra la finalización de la ejecución del job.

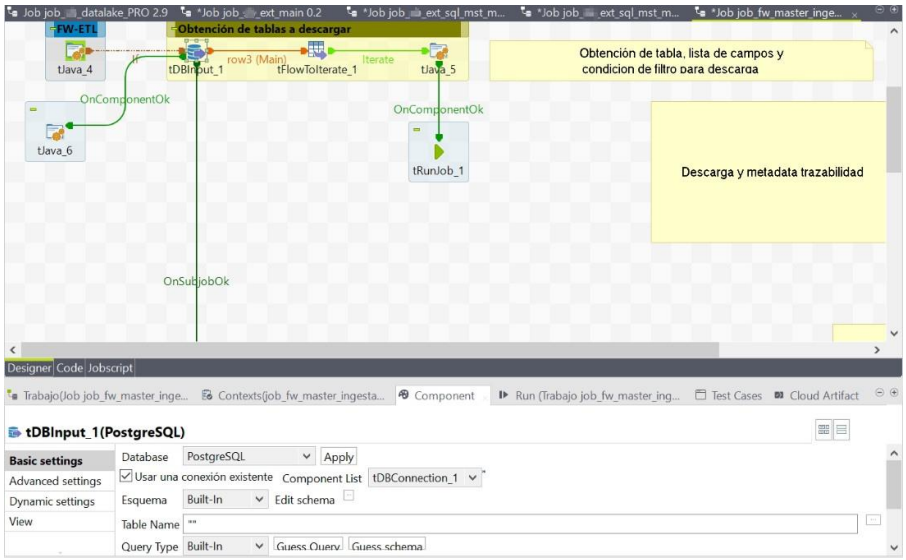


Figura 3.10: Conexión al servidor de PostgreSQL

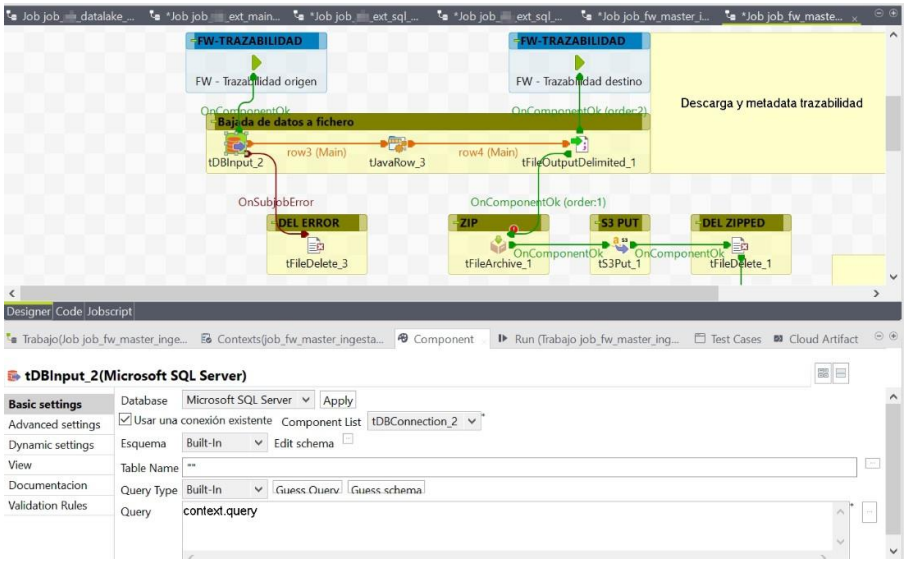


Figura 3.11: Creación de los ficheros

3.1.7. Flujo de la información

En esta sección describiremos cómo se mueve la información a través de las diferentes capas, desde los orígenes hasta la capa de consumo, con la ayuda del Framework y los jobs maestros. Podemos ver en la Figuras 3.12, 3.13 y 3.14 un ejemplo de este flujo de los datos. Cuando se produce una nueva ejecución de un proceso ETL, se pasa por las siguientes fases:

1. Aparición de la ejecución en las tablas de hechos sobre ejecuciones del Framework.
2. El job de ingesta del nuevo fichero hace uso de los jobs maestros y la tabla de ingestas para trasladar el fichero de los orígenes de datos a la Capa Bronze y posteriormente de la Capa Bronze a la Capa Silver en Snowflake. Todo este proceso es el explicado en la sección Maestros.
3. Se aplican las reglas de calidad funcionales a los datos de Staging y, la información correcta pasa a la Capa Persistente de Silver. Los datos que no cumplan estas reglas de calidad viajarán hasta la Error Zone.
4. La información que debe ir a la Capa Gold es cargada por los procesos ETL.

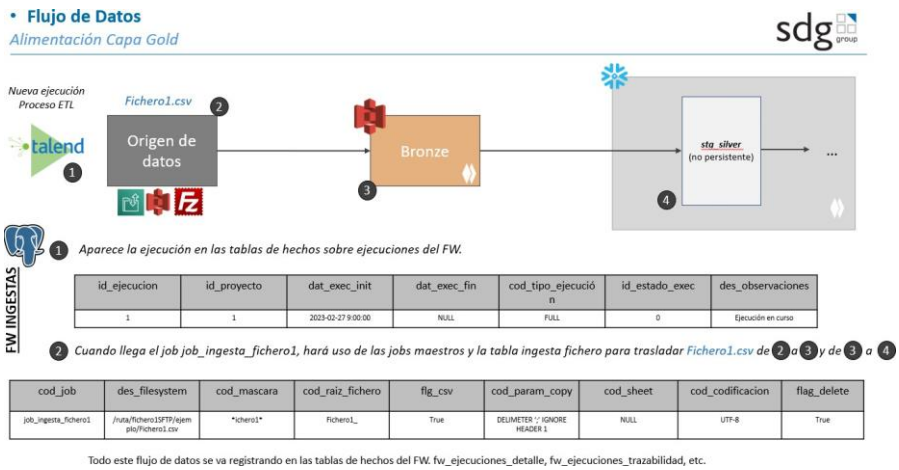


Figura 3.12: Flujo de Datos I

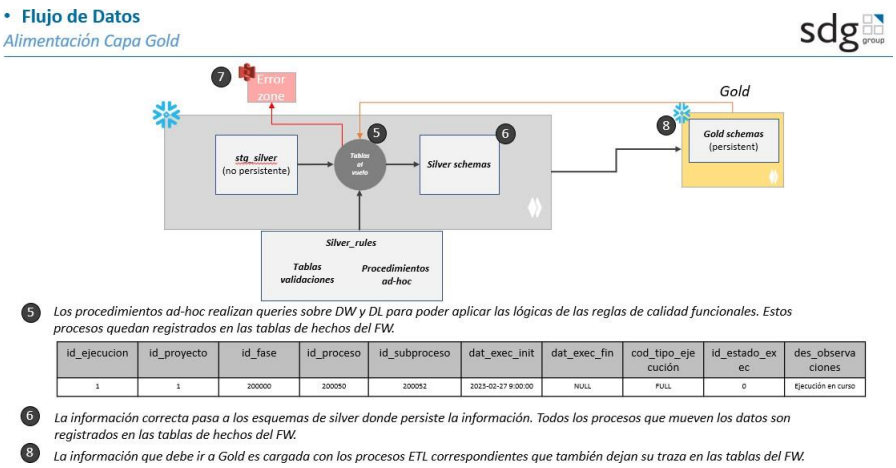


Figura 3.13: Flujo de Datos II

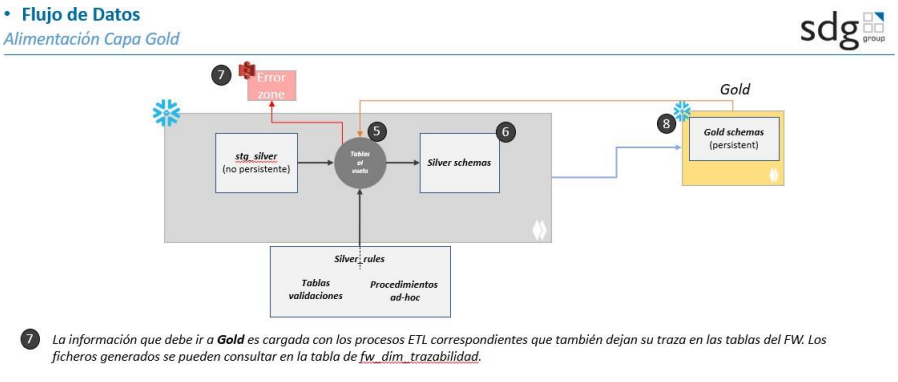


Figura 3.14: Flujo de Datos III

3.1.8. Capa de consumo - GOLD

Una vez finalizado todo el proceso de transformación de los datos a través de las diferentes capas mediante procesos ETL, tendremos finalmente una capa de consumo (GOLD) con los datos necesarios y apropiadamente modificados para su uso.

La base de datos final se llama **PRO_DL_GOLD** y contiene un modelo de datos que da servicio a los distintos departamentos que consumen el DataLake. Este modelo es similar a un modelo de estrella con tablas maestras y de hechos, y cubre diferentes ámbitos informacionales como ingresos, saldos, rentabilidades, etc.

Mecanismo de Control de Vigencia

Antes de explicar qué tablas tiene cada esquema debemos entender qué es el SCD (Slow Changing Dimension) o Mecanismo de Control de Vigencia.

Un Slowly Changing Dimension (SCD) en la gestión de datos y almacenamiento de datos se refiere a una dimensión que contiene datos relativamente estáticos que pueden cambiar lentamente pero de manera impredecible, en lugar de seguir un cronograma regular. Ejemplos típicos de dimensiones de cambio lento incluyen entidades como nombres de ubicaciones geográficas, clientes o productos.

Los escenarios que involucran SCD pueden causar problemas de integridad referencial. Por ejemplo, en una base de datos que contiene una tabla de hechos que almacena registros de ventas, esta tabla de hechos estaría vinculada a dimensiones mediante claves foráneas. Una de estas dimensiones puede contener datos sobre los vendedores de la empresa, como las oficinas regionales en las que trabajan. Sin embargo, los vendedores a veces son transferidos de una oficina regional a otra. Para los propósitos de informes históricos de ventas, puede ser necesario mantener un registro del hecho de que un vendedor particular había sido asignado a una oficina regional particular en una fecha anterior, mientras que ahora ese vendedor está asignado a una oficina regional diferente. Usar SCD puede ayudar a resolver este problema. Además, si se toman decisiones en base a los datos y más tarde estos se corrigen, es necesario poder justificar qué información era vigente cuando se tomaron esas decisiones.

Veamos con un ejemplo cómo funciona el **SCD2**:

Datos Silver (primera ejecución):

Campo 1 PK	Campo 2 PK	Campo 3 ATR	Campo 4 ATR	ID_EJECUCIÓN	IND_VIG_ACTIVO
A	A	Amarillo	10	1	TRUE
B	B	Azul	20	1	TRUE
C	C	Rojo	30	1	TRUE

Figura 3.15: Primera ejecución SCD2

En la Figura 3.15 vemos como sería la tabla de la Capa Silver en la primera ejecución. En la Figura 3.16 vemos la forma en la que se cargarían los datos en la Capa Gold en la que sería denominada la segunda ejecución. Se han añadido cuatro nuevas columnas con

Datos Gold (segunda ejecución):

Campo 1 PK	Campo 2 PK	Campo 3 ATR	Campo 4 ATR	ID_EJECUCIÓN	DT_VIG_FEC_INI	DT_VIG_FEC_FIN	TS_VIG_FEC_INI	TS_VIG_FEC_FIN	IND_VIG_ACTIVO
A	A	Amarillo	10	2	CURRENT_DATE	9999-12-31	CURRENT TIMESTAMP	9999-12-31 23:59:59:000	TRUE
B	B	Azul	20	2	CURRENT_DATE	9999-12-31	CURRENT TIMESTAMP	9999-12-31 23:59:59:000	TRUE
C	C	Rojo	30	2	CURRENT_DATE	9999-12-31	CURRENT TIMESTAMP	9999-12-31 23:59:59:000	TRUE

Figura 3.16: Segunda ejecución SDC2

fechas de vigencia. Vamos a ver ahora qué ocurre cuando se modifican, añaden o eliminan registros en la tabla de la Capa Silver.

Datos Silver (tercera ejecución):

Campo 1 PK	Campo 2 PK	Campo 3 ATR	Campo 4 ATR	ID_EJECUCIÓN	IND_VIG_ACTIVO
A	A	Amarillo	10	1	TRUE
B	B	Azul	20	1	TRUE
C	C	Rojo	30	1	TRUE
A	A	Amarillo	10	3	TRUE
C	C	Violeta	20	3	TRUE
D	D	Verde	40	3	TRUE

Figura 3.17: Tercera ejecución SDC2

Vemos en la Figura 3.17 que ha habido una serie de cambios. En la Figura 3.18 podemos ver qué cambios se han realizado.

Datos Silver (tercera ejecución):

Campo 1 PK	Campo 2 PK	Campo 3 ATR	Campo 4 ATR	ID_EJECUCIÓN	IND_VIG_ACTIVO
A	A	Amarillo	10	1	TRUE
B	B	Azul	20	1	TRUE
C	C	Rojo	30	1	TRUE
A	A	Amarillo	10	3	TRUE
C	C	Violeta	30	3	TRUE
D	D	Verde	40	3	TRUE

En la ejecución número 3, ha desaparecido el registro con PK: B B
En la ejecución número 3, ha cambiado el atributo del campo 3 el registro con PK: C C
En la ejecución número 3, se ha mantenido estático el registro con PK: A A
En la ejecución número 3, ha aparecido un nuevo registro con PK: D D

Figura 3.18: Cambios tercera ejecución SCD2

¿Cómo se refleja esto en la cuarta ejecución (GOLD) si estamos utilizando un SCD2? En la Figura 3.19 podemos ver cómo se muestran estos cambios en la capa de consumo.

Comparando las tablas de la Capa Gold en las ejecuciones segunda y cuarta se puede observar que:

- El registro con PK: A A ha cambiado su fecha de inicio de vigencia de CURRENT TIMESTAMP a 2024-04-09.
- El registro con PK: B B ha cambiado su ID_EJECUCION de 2 a 4 (pues es el último en el que ha sufrido cambios). Además, sus fechas de inicio y fin de vigencia han cambiado a 2024-04-09 y CURRENT TIMESTAMP respectivamente, pues el registro ya no está activo y, por lo tanto, el valor en la columna IND_VIG_ACTIVO pasa a ser FALSE.

Datos Gold (cuarta ejecución):

Campo 1 PK	Campo 2 PK	Campo 3 ATR	Campo 4 ATR	ID_EJECUCIÓN	DT_VIG_FEC_INI	DT_VIG_FEC_FIN	TS_VIG_FEC_INI	TS_VIG_FEC_FIN	IND_VIG_ACTIVO
A	A	Amarillo	10	2	2024-04-09	9999-12-31	2024-04-09 09:00:00:000	9999-12-31 23:59:59:000	TRUE
B	B	Azul	20	4	2024-04-09	CURRENT_DATE	2024-04-09 09:00:00:000	CURRENT_TIME STAMP – 1s	FALSE
C	C	Rojo	30	4	2024-04-09	CURRENT_DATE	2024-04-09 09:00:00:000	CURRENT_TIME STAMP – 1s	FALSE
C	C	Violeta	30	4	CURRENT_DATE	9999-12-31	CURRENT TIMESTAMP	9999-12-31 23:59:59:000	TRUE
D	D	Verde	40	4	CURRENT_DATE	9999-12-31	CURRENT TIMESTAMP	9999-12-31 23:59:59:000	TRUE

Figura 3.19: Cuarta ejecución SCD2

- El registro con PK: C C ha sufrido cambios en sus atributos, por lo tanto se da de baja el registro con los atributos anteriores (cambios en las fechas de vigencia y en el índice de vigencia activo) y se añade un nuevo registro con los nuevos atributos y las fechas correspondientes.
- Aparece un nuevo registro con PK: D D, por lo tanto se añade un nuevo registro con las fechas correspondientes.

Veamos ahora de la misma manera cómo funciona el **SCD4**:

Datos Silver (primera ejecución):

Campo 1 PK	Campo 2 PK	Campo 3 FEC FACT	Campo 4 ATR	ID_EJECUCIÓN	IND_VIG_ACTIVO
A	A	Lunes	2%	1	TRUE
B	B	Lunes	2.5%	1	TRUE
C	C	Lunes	3%	1	TRUE

Figura 3.20: Primera ejecución SCD4

En la Figura 3.20 podemos ver cómo sería la primera tabla que llega a la Capa Silver. Ahora, en Gold se crearán dos tablas diferentes. Una de ellas tendrá la foto actual y la otra tendrá una foto histórica de los datos, como se puede ver en la Figura 3.21.

Datos Gold (segunda ejecución):

Tabla foto actual

Campo 1 PK	Campo 2 PK	Campo 3 FEC FACT	Campo 4 ATR	ID_EJECUCIÓN	DT_VIG_FEC_INI	TS_VIG_FEC_INI
A	A	Lunes	2%	2	CURRENT_DATE	CURRENT TIMESTAMP
B	B	Lunes	2.5%	2	CURRENT_DATE	CURRENT TIMESTAMP
C	C	Lunes	3%	2	CURRENT_DATE	CURRENT TIMESTAMP

Tabla foto histórica

Campo 1 PK	Campo 2 PK	Campo 3 FEC FACT	Campo 4 ATR	ID_EJECUCIÓN	DT_VIG_FEC_INI	DT_VIG_FEC_FIN	TS_VIG_FEC_INI	TS_VIG_FEC_FIN	IND_VIG_ACTIVO

Figura 3.21: Segunda ejecución SCD4

Ahora se aplican cambios sobre la tabla de Silver en la que se llamará tercera ejecución. La nueva tabla de Silver será la de la Figura 3.22.

En la Figura 3.23 podemos ver cuáles han sido los cambios en la tercera ejecución.

Podemos ver en la Figura 3.24 cómo quedarían las tablas de Gold tras la cuarta ejecución. Si las comparamos con las tablas que teníamos tras la segunda ejecución podemos observar que:

Datos Silver (tercera ejecución):

Campo 1 PK	Campo 2 PK	Campo 3 FEC FACT	Campo 4 ATR	ID_EJECUCIÓN	IND_VIG_ACTIVO
A	A	Lunes	2%	1	TRUE
B	B	Lunes	2.5%	1	TRUE
C	C	Lunes	3%	1	TRUE
B	B	Lunes	2.4%	3	TRUE
A	A	Martes	2.1%	3	TRUE
B	B	Martes	2.3%	3	TRUE
C	C	Martes	3.2%	3	TRUE

Figura 3.22: Tercera ejecución SCD4

Datos Silver (tercera ejecución):

Campo 1 PK	Campo 2 PK	Campo 3 FEC FACT	Campo 4 ATR	ID_EJECUCIÓN	IND_VIG_ACTIVO
A	A	Lunes	2%	1	TRUE
B	B	Lunes	2.5%	1	TRUE
C	C	Lunes	3%	1	TRUE
B	B	Lunes	2.4%	3	TRUE
A	A	Martes	2.1%	3	TRUE
B	B	Martes	2.3%	3	TRUE
C	C	Martes	3.2%	3	TRUE

En la ejecución número 3, ha cambiado el atributo del campo 4 el registro con PK: B B

En la ejecución número 3, han entrado nuevos hechos

¡Ahora la PK real son los campos de PK + la fecha del hecho! ⚠

Figura 3.23: Cambios tercera ejecución SCD4

Datos Gold (cuarta ejecución):

Tabla foto actual

Campo 1 PK	Campo 2 PK	Campo 3 FEC FACT	Campo 4 ATR	ID_EJECUCIÓN	DT_VIG_FEC_INI	TS_VIG_FEC_INI
A	A	Lunes	2%	2	2024-04-09	2024-04-09 09:00:00:000
B	B	Lunes	2.4%	2	CURRENT_DATE	CURRENT TIMESTAMP
C	C	Lunes	3%	2	2024-04-09	2024-04-09 09:00:00:000
A	A	Martes	2.1%	4	CURRENT_DATE	CURRENT TIMESTAMP
B	B	Martes	2.3%	4	CURRENT_DATE	CURRENT TIMESTAMP
C	C	Martes	3.2%	4	CURRENT_DATE	CURRENT TIMESTAMP

Tabla foto histórica

Campo 1 PK	Campo 2 PK	Campo 3 FEC FACT	Campo 4 ATR	ID_EJECUCIÓN	DT_VIG_FEC_INI	DT_VIG_FEC_FIN	TS_VIG_FEC_INI	TS_VIG_FEC_FIN
B	B	Lunes	2.5%	4	2024-04-09	2024-04-10	2024-04-09 09:00:00:000	Current Timestamp – 1s

Figura 3.24: Cuarta ejecución SCD4

- La tabla de foto histórica ahora tiene un registro con PK: B B Lunes, que ya no está vigente.

- Los tres nuevos registros se han añadido a la tabla foto actual con las fechas e ID_EJECUCION correspondientes.

Una vez explicado esto, podemos ver qué tablas contiene cada esquema, qué información contienen y qué tipo de SCD siguen.

Figura 3.25: División de esquemas de la BBDD final

Haciendo referencia a la Figura 3.25, tenemos:

- Los esquemas INFORMATION_SCHEMA y PUBLIC se crean por defecto al crear una base de datos y por lo tanto no contienen información del proyecto.

- Las tablas pertenecientes a los esquemas CAPTACION y RENTABILIDADES no son aprovisionadas por el proyecto de Gold.

- El esquema AUX_TMP contiene tablas auxiliares para el movimiento de los datos.

- Las tablas del esquema MASTERDATA son tablas de dimensión, atributos que cambian poco, y por lo tanto siguen un SCD2. En la Figura 3.26 podemos ver qué tablas hay en este esquema, donde podemos ver tablas con información sobre clientes, banqueros, instrumentos...

- Los esquemas INGRESOS y SALDOS contienen tablas de hechos con datos diarios y por lo tanto siguen un SCD4. Podemos ver en la Figura 3.27 las tablas pertenecientes a estos dos esquemas. Observamos que, al seguir un SCD4, cada tabla tiene su foto actual y su foto histórica.

- El esquema OPERATIVA guarda información de la operativa diaria del banco. En este caso, las tablas siguen un SCD4, pero solo se guarda la tabla de última foto, descartando la tabla con datos históricos. Podemos ver estas tablas en la Figura 3.28.

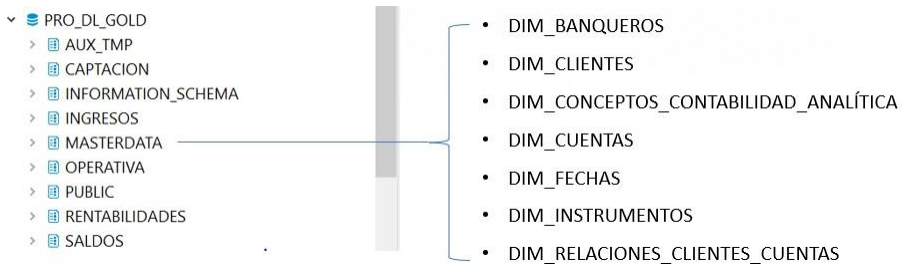


Figura 3.26: Tablas de MASTERDATA

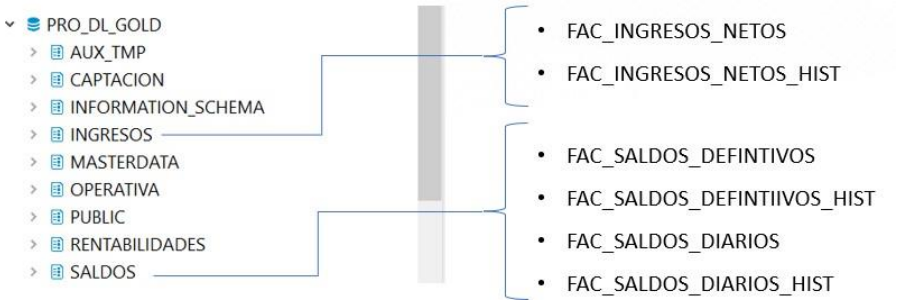


Figura 3.27: Tablas de INGRESOS y SALDOS

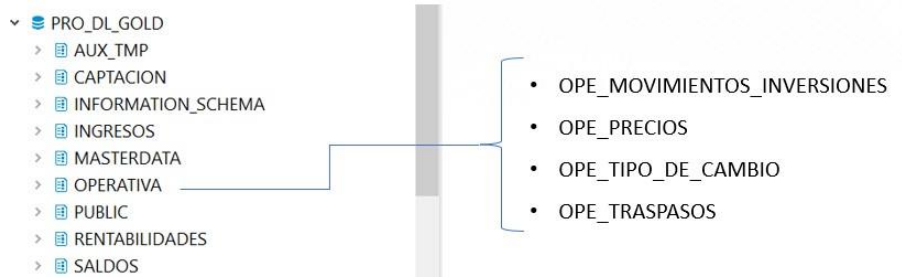


Figura 3.28: Tablas de OPERATIVA

Las ingestas de datos de la Capa Silver a la Capa Gold se realizan mediante SPs (Store Procedures). Los SPs son scripts ejecutables en Snowfkale que permiten utilizar parámetros de entrada y anidarse con otros. Cada tabla de Gold tiene asociado un SP al que se hace una llamada cuando se quieran actualizar los datos de la capa de consumo. Esto se paraleliza con la actualización de la tabla de ingestas del Framework, para tener un registro de qué actualizaciones se han hecho sobre los datos.

3.2. Caso de uso (PDF)

3.2.1. Necesidad del cliente

Como se mencionó anteriormente, los servicios de banca privada ofrecen a sus clientes servicios personalizados. En esta línea, la empresa contratante decidió que el primer uso que se le quería dar al Data Lake creado en la primera parte del proyecto era la automatización en la generación de reportes personalizados para cada cliente cuando este los solicitase, donde pudiese observar datos sobre sus cuentas y carteras durante un intervalo de tiempo determinado.

3.2.2. Descripción del circuito

Hay tres agentes involucrados en la generación de estos informes:

1. Banqueros que lo solicitan.
2. El servidor de Snowflake propiedad del banco y los procesos ETL que se ejecutan sobre él.
3. Una entidad que se encarga de utilizar la información extraída de Snowflake para crear un PDF estético que se envía de vuelta al banquero para que se lo entregue al cliente.

El circuito para la generación de estos informes sigue los siguientes pasos:

1. **Recibo de la solicitud.** Un banquero desde una oficina tiene la capacidad de generar informes a través de una solicitud mediante un formulario. Esta solicitud se traduce en una petición a una API de Snowflake que generará un registro en una de nuestras tablas donde se indica el cliente para el que se hará el informe, el banquero de ese cliente, las fechas entre las que se quiere el informe, la divisa y el tipo de informe. También se especifican los contratos que se quiere que aparezcan en el informe (carteras asesoradas, planes de pensiones...).
2. **Creación del JSON.** Una vez se recibe la solicitud, utilizando un proceso de almacenado y varias tablas auxiliares que se describen más adelante, se genera un fichero de tipo JSON que contiene toda la información necesaria para la generación del informe deseado.
3. **Envío del JSON.** A través de una lambda de AWS [1] se envía el JSON generado al tercer agente del circuito que se encarga del montaje del PDF usando la información generada en Snowflake y lo entrega via mail al banquero.

3.2.3. Descripción del proceso

El proceso se realizará mediante un SP (Stored Procedure) que realizará funciones desde la llegada de la solicitud hasta la generación del JSON final.

Todo este proceso comienza con la llegada de un JSON de entrada, que tendrá los datos del cliente para el que será el informe, su banquero, las fechas entre las que se quiere el informe, la divisa, los contratos de los que se quiere información... Se puede ver en la Figura 3.29 cómo es este fichero.

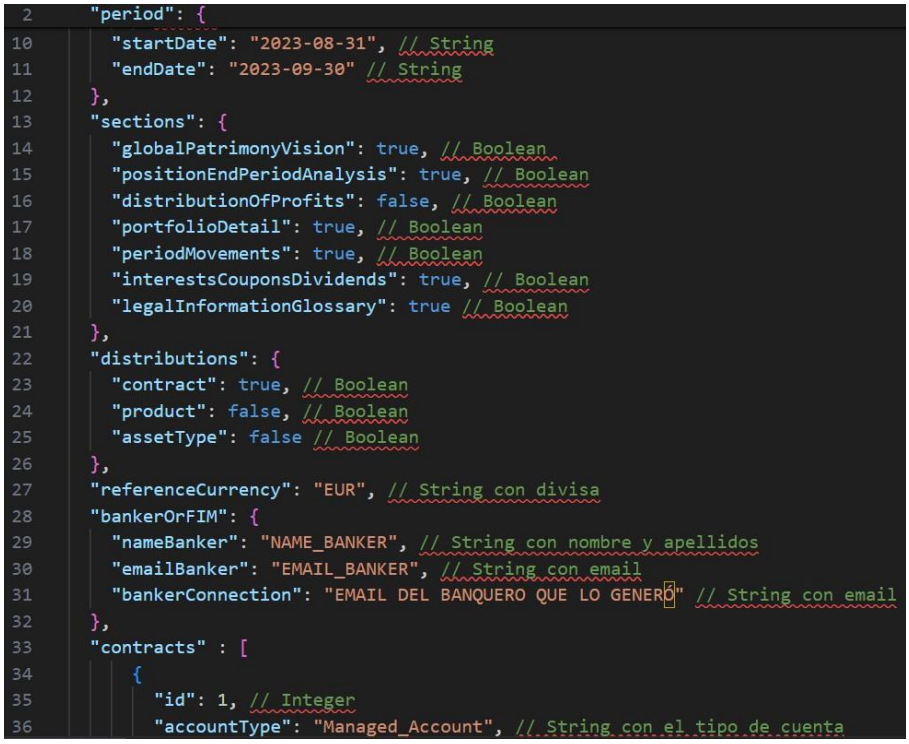


Figura 3.29: JSON de entrada

Habrá una tabla llamada **json_path** (Figura 3.30) que será común a todas las solicitudes y donde se encuentran una serie de etiquetas asociadas con un json_path. Un json_path es la localización de un campo en un JSON. Esta tabla sirve para que, una vez se extraiga la información de las tablas de la Capa de Consumo de Snowflake, no se asocie cada dato con el json_path correspondiente, sino que se asocie con un nombre más legible, en este caso la etiqueta.

La primera parte del SP consiste en la creación de cuatro tablas auxiliares:

1. **tabla_datos_solicitud:** En esta tabla se extraerá del JSON de entrada un registro con la información sobre el cliente, el banquero, las fechas, la divisa y el tipo de informe. Podemos ver un ejemplo en la Figura 3.31.

2. **tabla_datos_contratos:** En esta tabla se extraerán del JSON de entrada datos sobre los diferentes contratos o activos del cliente que se incluirán en el informe. La Figura 3.32 es un ejemplo de esta tabla. Tanto esta tabla como la anterior serán cruzadas con las tablas de Snowflake a lo largo del SP para filtrar solo los datos que nos interesan.

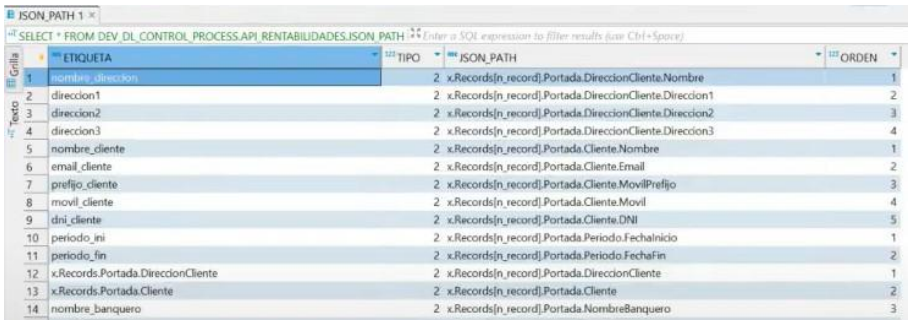


Figura 3.30: Tabla jsonpath

3. **tabla_indices:** Esta tabla se va llenando conforme avanza el SP. La función de esta tabla es dar un orden definido por negocio a los distintos datos que entran en una misma categoría. En otras palabras, la finalidad es asegurar que los registros siempre se pintan en el mismo orden sin importar la solicitud. Por ejemplo, si se muestra un listado de contratos y negocio quiere que se muestren primero las cuentas corrientes, después los planes de ahorro y por último los planes de pensiones, esta tabla ordena los distintos datos según el criterio generado por negocio. Podemos ver un ejemplo en la Figura 3.33.

4. **tabla_profundidades:** En esta tabla, que será la que almacena los datos finales para la creación del JSON, tendremos los valores que sacaremos de la base de datos asociados con el json_path correspondiente a partir de la etiqueta. Habrá además una columna llamada "profundidad" que será básicamente la profundidad del valor en el JSON. Esta columna será útil después para la creación del JSON completo de manera recursiva. Al igual que la tabla de índices, esta tabla también se va llenando a medida que avanza el SP. Podemos ver un ejemplo de esta tabla en la Figura 3.34.

Figura 3.31: Tabla auxiliar datos solicitud

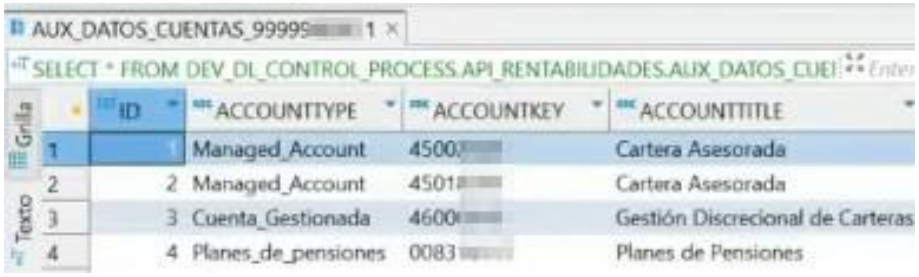


Figura 3.32: Tabla auxiliar datos contratos

AUX_JSON_INDICES_999999690 1 ×

SELECT * FROM DEV_DL_CONTROL_PROCESS

Grilla | Texto | Record

	NOMBRE_INDICE	VALOR	PK
1	n_periodo	0	0
2	n_periodo	1	1
3	n_periodo	2	2
4	n_periodo	3	3
5	n_periodo	4	4
6	n_periodo	5	5
7	n_periodo	6	6
8	n_vencimiento	2	3-5 años
9	n_vencimiento	3	5-10 años
10	n_vencimiento	4	10-30 años
11	n_vencimiento	5	Más de 30 años
12	n_vencimiento	6	Otros
13	n_riesgo	5	B
14	n_riesgo	6	<B
15	n_riesgo	7	NC
16	n_riesgo	8	Otros
17	n_vencimiento	0	Menos de un año
18	n_vencimiento	1	1-3 años
19	n_categoria	6	Materias primas
20	n_categoria	7	Inversión alterna
21	n_categoria	8	Inmobiliarios
22	n_categoria	9	Otros
23	n_contrato	0	Cliente

Figura 3.33: Tabla auxiliar índices

Figura 3.34: Tabla auxiliar profundidades

3.2.4. Montaje del JSON

La última parte del SP es la parte que monta el JSON. Haciendo uso de la tabla_profundidades, buscaremos la máxima profundidad presente en la tabla y crearemos el JSON de dentro hacia afuera. En esa misma tabla de profundidades, iremos creando nuevos valores que almacenarán toda la información que esté más profunda.

Por ejemplo, si la máxima profundidad es 11, buscaremos los json_path con profundidad 11 que tengan el mismo parent_path, es decir, que solo difieran en esa última profundidad. Se creará entonces un registro nuevo con el parent_path y profundidad 10 cuyo valor será un JSON conteniendo los valores de los registros de profundidad 11 con ese parent_path. De esta forma, se van creando pequeños JSON que, según vayamos ascendiendo en profundidad, serán cada vez más grandes, hasta llegar a un JSON con profundidad 0 que será el que contendrá toda la información necesaria.

Este JSON se envía a la tercera entidad a través de una API de AWS y mediante un procedimiento de almacenado.

3.2.5. Recreación visual del informe

Una vez el JSON llega a esta empresa, ellos crean el informe personalizado utilizando los datos que nosotros enviamos. En las Figuras 3.35, 3.36, 3.37 y 3.38 podemos ver algunos ejemplos de páginas de estos informes personalizados.

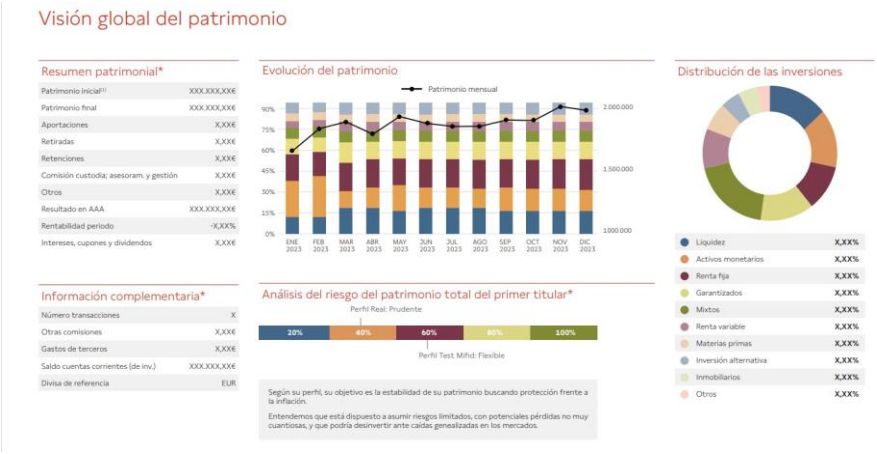


Figura 3.35: Visión Global del Patrimonio

Se puede ver que es un documento estético y bien organizado en el que el cliente podrá ver el estado de sus activos financieros de manera clara y sencilla.

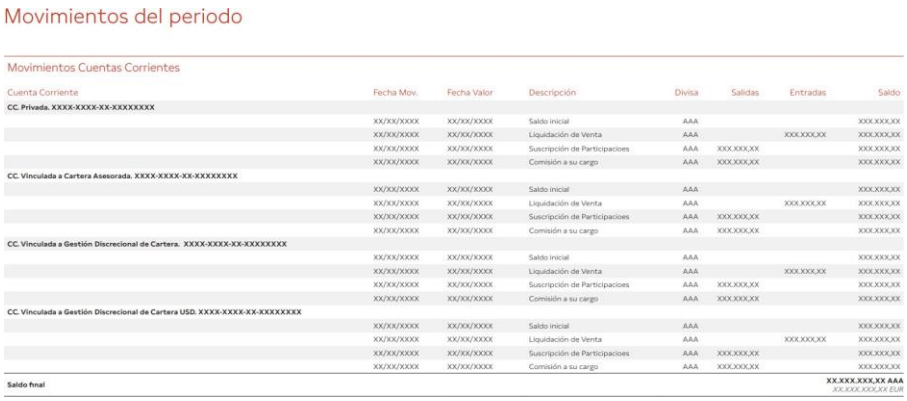


Figura 3.36: Movimientos del periodo

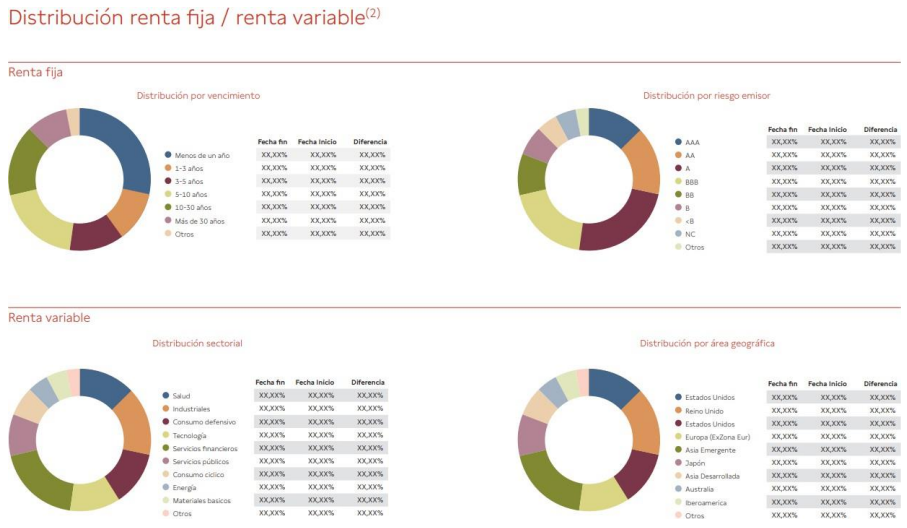


Figura 3.37: Distribución Renta Fija Renta Variable

Figura 3.38: Análisis de Productos con Saldo a Fin de Periodo

Capítulo 4

Conclusiones

La implementación de un Datalake corporativo utilizando Snowflake y procesos ETL basados en metadatos gestionados con Talend ha demostrado ser una solución robusta, eficiente y unificada para la gestión de datos en esta entidad financiera. Este proyecto ha logrado abordar varios desafíos críticos relacionados con la integración, almacenamiento y procesamiento de grandes volúmenes de datos provenientes de diversas fuentes, dando servicio a departamentos que antes tenían silos informacionales.

Resultados Alcanzados:

1. **Almacenamiento y Procesamiento Eficiente:**
   - Snowflake ha proporcionado una plataforma escalable y de alto rendimiento, permitiendo el almacenamiento eficiente de datos estructurados y no estructurados.
   - La separación de almacenamiento y computación ha optimizado los costos y mejorado la capacidad de respuesta del sistema.
2. **Flexibilidad y Adaptabilidad en Procesos ETL:**
   - Los procesos ETL diseñados con Talend han demostrado ser flexibles y dinámicos, facilitando la adaptación a cambios en las estructuras de datos fuente sin necesidad de rediseñar los flujos de trabajo completos.
   - La utilización de un enfoque basado en metadatos ha mejorado la gestión y y el time to market de los datos, siendo muchos más fácil añadir nueva información al DataLake.
3. **Generación de Informes Personalizados:**
   - La integración de datos de diversas fuentes, su transformación y limpieza, y la generación automatizada de archivos JSON han permitido la creación de informes detallados y útiles para los clientes, entregados en formato PDF por un tercero colaborador.

Impacto en la Organización:

- Mejora en la Toma de Decisiones: La disponibilidad de datos precisos y actualizados ha potenciado la capacidad de análisis y la toma de decisiones informadas en la entidad financiera.
- Eficiencia Operativa: La automatización de procesos y la optimización en la gestión de datos han reducido tiempos y costos operativos.
- Valor Agregado para los Clientes: La capacidad de generar informes personalizados se espera que mejore la experiencia del cliente con un trato más individualizado a un cliente de tipo banca privada.

Antes	**Después**
Cada departamento obtenía sus propios datos de fuentes distintas, provocando desalineamiento con otras áreas.	Se usa un repositorio de información centralizado, evitando discrepancias entre datos de distintos departamentos.
Diariamente se inviertía tiempo en generar ficheros excel y realizar tareas manuales.	La generación de ficheros y realización de tareas manuales está automatizada generando tiempo para otras tareas.
La toma de decisiones estaba exclusivamente basada en el know-how y el expertise de los trabajadores de la empresa.	La toma de decisiones combina el expertise presente en la empresa con datos que apoyan las decisiones.
A menudo se repetían tareas entre departamentos. No había un conocimiento profundo de toda la información disponible.	Facilita la compartición de información y la colaboración entre áreas, mejorando la eficiencia con información ya incluida por proyectos de otros departamentos.

Cuadro 4.1: Antes y después del proyecto

Futuras Expansiones y Mejoras:

Este proyecto sienta las bases para futuras expansiones y mejoras en la infraestructura de datos de la organización. Algunas de las áreas clave para el desarrollo continuo incluyen:

- Ampliación del Datalake: Incorporar nuevas fuentes de datos y expandir las capacidades del Datalake para cubrir más áreas de negocio.
- Mejora de la Calidad de los Datos: Implementar herramientas y procesos adicionales para el enriquecimiento y la validación de datos.
- Integración de Analítica Avanzada: Utilizar técnicas de análisis avanzado y machine learning para obtener insights más profundos y predictivos.
- Optimización Continua de ETL: Seguir mejorando los procesos ETL para aumentar la eficiencia y reducir aún más los tiempos de procesamiento.

En conclusión, la implementación de un Datalake corporativo con Snowflake y Talend ha marcado un avance significativo en la gestión de datos de la entidad financiera, proporcionando una base sólida para futuras innovaciones y mejoras operativas. Este proyecto no solo ha cumplido con los objetivos iniciales, sino que también ha demostrado el valor tangible de una infraestructura de datos bien diseñada en el contexto de la transformación digital de la organización.

Bibliografía

[1] Amazon Web Services, Inc. Aws lambda. https://aws.amazon.com/es/pm/lambda/, 2023. Retrieved May 3, 2024.

[2] Amazon Web Services, Inc. The difference between etl and elt. https://aws.amazon.com/es/compare/the-difference-between-etl-and-elt/#:~:text=The%20ETL%20process%20transforms%20data,data%20whenever%20you%20need%20it, 2023. Retrieved March 15, 2024.

[3] Amazon Web Services, Inc. ¿cuál es la diferencia entre un almacenamiento de datos, un lago de datos y un data mart? https://aws.amazon.com/es/compare/the-difference-between-a-data-warehouse-data-lake-and-data-mart/, 2023. Retrieved March 13, 2024.

[4] Astera. Introduction to metadata architecture. https://www.astera.com/es/type/blog/introduction-to-metadata-architecture/, 2023. Retrieved March 20, 2024.

[5] David Baum. *Cloud Data Warehousing for Dummies, 3rd Snowflake Special Edition*. John Wiley Sons, Inc., 2024.

[6] Soumen Chakraborty. Why you need a metadata-driven data integration framework. https://www.freshgravity.com/metadata-driven-data-integration-framework/, 2023. Retrieved March 20, 2024.

[7] Data Ideology. Snowflake etl vs elt. https://www.dataideology.com/snowflake-etl-vs-elt/#:~:text=For%20use%20with%20the%20cloud,the%20fast%2Dpaced%20company%20culture., n.d. Retrieved March 15, 2024.

[8] Estrategias de Inversión. Ranking de las mejores entidades de banca privada. https://www.estrategiasdeinversion.com/actualidad/noticias/empresas/ranking-de-las-mejores-entidades-de-banca-privada-n-627289, 2024. Retrieved March 11, 2024.

[9] KeepCoding. Pushdown in data warehousing: What it is and how it works. https://keepcoding.io/blog/pushdown-data-warehouse/, 2024. Retrieved March 15, 2024.

[10] Mindmajix. Talend etl tutorial. https://mindmajix.com/talend-etl, 2023. Retrieved April 21, 2024.

[11] Talend. Talend documentation. https://help.talend.com/en-US/components/7.3/Content/Components/Home.htm, n.d. Retrieved April 21, 2024.