UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

Migración de procesos ETL a la nube con Informatica IICS: caso práctico de integración de datos con Salesforce en una empresa del sector asegurador

TRABAJO FIN DE MÁSTER

Mayo - 2026

AUTOR: Joan Toral García

DIRECTOR: Prof. Pablo Corral González

RESUMEN

Este Trabajo Fin de Máster se centra en la migración de varios procesos ETL que utilizan la herramienta de Informatica PowerCenter a Informatica Intelligent Cloud Services (IICS) dentro de un entorno real de una empresa del sector asegurador. El proyecto surge de la necesidad de modernizar la integración de datos y mejorar la conexión con Salesforce mediante una plataforma en la nube más flexible y automatizada.

Se comparan ambas herramientas en términos de rendimiento, facilidad de mantenimiento, control de calidad y costes, incorporando prácticas de DataOps para garantizar la trazabilidad y coherencia de la información.

Tras la implementación y validación de las pruebas, se observa una mejora notable en la eficiencia del proceso y en la capacidad de supervisión de los flujos, demostrando que la migración a IICS es una alternativa fiable para modernizar sistemas ETL tradicionales.

Palabras clave: ETL, IICS, PowerCenter, Salesforce, DataOps, automatización, integración de datos, migración cloud.

ABSTRACT

This Master's Thesis describes the process of migrating ETL workflows from Informatica PowerCenter to Informatica Intelligent Cloud Services (IICS) in a real business environment within the insurance sector. The project was motivated by the need to modernize data integration and to improve connectivity with Salesforce through a more flexible cloud-based platform.

The study compares both tools regarding performance, maintainability, data quality control, and overall cost, applying DataOps principles to ensure traceability and consistency during the migration.

After the implementation and testing stages, a clear improvement in process efficiency and monitoring capability was achieved, confirming that moving from PowerCenter to IICS is a practical and effective step towards the modernization of traditional ETL systems.

Keywords: ETL, IICS, PowerCenter, Salesforce, DataOps, automation, data integration, cloud migration.

Índice de Figuras

Índice de Tablas

Índice

1. INTRODUCCIÓN

1.1 Contexto general de la gestión de datos

En el mundo actual, los datos son un recurso clave para las organizaciones. Vivimos en una era en la que se generan volúmenes masivos de datos provenientes de múltiples fuentes: sistemas internos de las empresas, aplicaciones en la nube, sensores IoT, redes sociales, entre otros. Este crecimiento ha llevado a las organizaciones a preguntarse cómo gestionar los datos, transformarlos y analizarlos para la toma decisiones. Debido a la abundancia de datos, las organizaciones deben hacer frente a diferentes retos:

- Fuentes de datos: Los datos proceden de múltiples sistemas, como bases de datos, hojas de cálculo o aplicaciones en la nube, y pueden presentarse en distintos formatos, ya sean estructurados, semiestructurados o no estructurados. Esta diversidad complica su integración y tratamiento.
- Calidad de los datos: En muchos casos, los datos no están completos, no se encuentran actualizados o no cumplen con el formato requerido para su análisis, lo que puede derivar en errores y afectar a la fiabilidad de los resultados.
- Velocidad de los procesos: En un entorno donde las decisiones deben tomarse en tiempo real, los procesos de integración y análisis de datos deben ejecutarse sin demoras, garantizando rapidez y eficiencia en el tratamiento de la información.

Para hacer frente a los desafíos, las organizaciones crean y usan herramientas para convertir los datos en información útil. Los procesos ETL (Extract, Transform, Load) son una de las soluciones y forman parte de la gestión de los datos desde su aparición [1].

Los sistemas ETL permiten:

- Centralizar los datos dispersos en un único repositorio (como un almacén de datos o un data warehouse).
- Asegurar la calidad y la consistencia de los datos con transformaciones y validaciones.
- Preparar los datos para análisis avanzado y generación de los reportes.

En la actualidad, los procesos ETL están evolucionando hacia soluciones en la nube y en tiempo real, donde tecnologías como ETL en la nube (IICS, Talend, etc.) están marcando el futuro [2]. Este TFM se enmarca en este contexto, mostrando cómo los procesos ETL han evolucionado y su impacto en la gestión moderna de datos.

1.2 Actividad de la empresa y entorno tecnológico

La empresa en la que se enmarca este trabajo pertenece al sector asegurador, y se dedica principalmente a la gestión de pólizas, clientes, campañas comerciales y servicios relacionados con la fidelización y retención de asegurados. Esta actividad implica trabajar con un volumen elevado de datos, especialmente cuando se lanzan campañas de marketing orientadas a ofrecer nuevos productos o condiciones personalizadas en función del perfil del cliente.

A nivel tecnológico, el entorno en el que opera la empresa ha estado tradicionalmente apoyado en soluciones locales, siendo Informatica PowerCenter una de las herramientas clave en los procesos de integración de datos. Durante años, esta plataforma ha centralizado la información de diferentes fuentes aplicando reglas de negocio y cargando el resultado en diferentes destinos ya sea ficheros, bases de datos internas o plataformas externas.

Sin embargo, la empresa Informatica, propietaria de Powercenter, está siguiendo una estrategia para modernizar y pasar a modelos más flexibles iniciando una migración a la nube. En este contexto, la herramienta Informatica IICS (Intelligent Cloud Services) ha tomado el relevo ofreciendo capacidades similares para transformar y cargar datos igual de Powercenter pero brindando mayor escalabilidad, facilidad de mantenimiento y mejor conectividad con sistemas en la nube. Esta transición no solo responde a un tema técnico, también responde a una necesidad del negocio reduciendo la dependencia de infraestructuras físicas y facilitando el despliegue de procesos en diferentes entornos. Migrar a un entorno cloud mejora los tiempos de respuesta en campañas que necesitan agilidad y datos actualizados todo el tiempo.

1.3 Justificación de la migración de PowerCenter a IICS

La transición de Informatica PowerCenter a Informatica Intelligent Cloud Services (IICS) no es solo una elección técnica, sino que está dentro de un proceso de transformación tecnológico y sostenible. La empresa aseguradora impulsó la decisión por los cambios en el ecosistema de las herramientas de integración de datos debido en gran parte al anuncio por parte del proveedor Informatica de que la versión 10.4 de PowerCenter, la utilizada en la compañía en el momento del proyecto, dejaría de contar con soporte oficial, lo que introducía un riesgo significativo en términos de seguridad, mantenimiento y continuidad de servicio [3].

La falta de soporte de la versión impide recibir más actualizaciones, parches de seguridad

y correcciones de vulnerabilidades convirtiendo cualquier sistema con la versión 10.4 en una infraestructura obsoleta en el medio plazo por lo que ya es un argumento fuerte para iniciar una migración y crucial en entornos con los datos como núcleo del negocio haciendo que la integridad del sistema sea prioritaria.

A nivel funcional, PowerCenter ha sido una solución fiable por años y permite diseñar procesos ETL complejos y estables pero, la arquitectura se basa en un modelo tradicional de infraestructura on-premise, obligando al equipo técnico a gestionar todo de forma constante, mantener los servidores dedicados, realizar las tareas de mantenimiento periódico y manejar las versiones de forma manual. Este modelo funciona bien en algunos contextos pero tiene limitaciones cuando la organización necesita escalar, colaborar desde distintas ubicaciones o cuando se quiere integrar los servicios en la nube de forma rápida. [4].

Por el contrario, IICS tiene un enfoque más moderno y está en línea con las tendencias actuales de la ingeniería de datos. Como es una solución nativa en la nube, la plataforma gestiona automáticamente muchas de las tareas administrativas que antes hacían los equipos internos, actualiza las versiones, escala los recursos, asegura el entorno y distribuye el acceso. Además, IICS ofrece una integración nativa con otras plataformas en la nube como Salesforce. Esta novedad supone un beneficio en proyectos donde intervienen múltiples sistemas que trabajan de forma sincronizada. [5].

Otra ventaja importante de IICS es el modelo de trabajo descentralizado eliminando así la dependencia de los servidores físicos en las ubicaciones concretas. Los equipos pueden trabajar con el modelo de trabajo descentralizado de una manera más flexible y respondiendo a las necesidades de una forma más rápida. Esta adaptabilidad se vuelve clave cuando hablamos de campañas comerciales dinámicas, donde los plazos son ajustados y los cambios de último momento son frecuentes [5].

Finalmente, con la migración a IICS, la empresa aseguradora se posiciona mejor para futuras integraciones, nuevos modelos de negocio basados en datos, y la incorporación de tecnologías emergentes como la automatización inteligente y la analítica avanzada.

En conjunto, la decisión de abandonar PowerCenter 10.4 y adoptar IICS se configura como una respuesta coherente, tanto a las limitaciones tecnológicas del entorno anterior como a la oportunidad de evolucionar hacia un modelo más sostenible, moderno y alineado con la realidad operativa del negocio actual.

1.4 Objetivo del TFM y estructura del documento

El objetivo principal de este trabajo es analizar cómo se lleva a cabo la integración de datos en un entorno real, partiendo de un caso concreto que surge dentro de una empresa del sector asegurador. Para ello, se ha elegido un proceso representativo que involucra el tratamiento de información relacionada con campañas comerciales y sus clientes, y que ha sido desarrollado en dos plataformas diferentes: una tradicional y una moderna basada en la nube.

A través de este análisis se pretende no solo comparar herramientas como PowerCenter e IICS, sino también entender cómo ha evolucionado la forma de trabajar con datos, qué ventajas aporta cada enfoque y cuáles son los aspectos clave que deben tenerse en cuenta a la hora de afrontar una migración tecnológica de este tipo. Además, se plantea una pequeña simulación previa que reproduce manualmente el proceso, con el fin de mostrar las diferencias entre resolver este tipo de problemas con código y hacerlo mediante soluciones especializadas.

Para que la lectura sea clara y progresiva, el documento se ha dividido en distintos bloques. En primer lugar, se presenta un repaso general sobre el tratamiento de datos y la evolución de las herramientas disponibles para ello, a continuación, se describe en detalle el proceso seleccionado, tanto desde su versión manual como en su implementación en PowerCenter e IICS y posteriormente se expone el bloque de análisis de resultados y discusiones, donde se comparan los enfoques, se presentan observaciones derivadas de la práctica y se valoran los resultados obtenidos. Finalmente, se incluyen unas conclusiones, anexos complementarios y la bibliografía utilizada como soporte.

2. MATERIAL Y MÉTODOS

Para mantener una lectura más sencilla y dar continuidad al desarrollo del trabajo, en este apartado se agrupan tanto las referencias teóricas como la explicación de los pasos que se han seguido durante el proyecto. De este modo, se presenta en un mismo lugar el contexto que sostiene la propuesta y la forma en la que se ha llevado a cabo, facilitando comprender el proceso completo sin separar la parte conceptual de la práctica.

2.1 Estado del arte

En los últimos años se han publicado numerosos estudios que analizan la evolución de los procesos ETL y su adaptación a entornos en la nube, los cuales, permiten comparar enfoques, herramientas y metodologías orientadas a la mejora de la calidad del dato. Se muestran a continuación varios trabajos especialmente relevantes para este proyecto por su similitud con las tecnologías usadas como Powercenter o IICS o por la aplicabilidad de los resultados.

Evolution of ETL Tools: Trends and Insights from On-Premises to Cloud Solutions (2025) [6]

Este estudio analiza la evolución de las herramientas ETL, desde soluciones locales hasta plataformas en la nube. Destaca que la escalabilidad, la automatización y la gobernanza del dato son pilares de las nuevas arquitecturas en la nube.

Las conclusiones del trabajo apoyan la migración del proyecto demostrando que pasar de entornos tradicionales como PowerCenter a soluciones en la nube como IICS aumenta la eficiencia operativa y reduce los costos de mantenimiento y de infraestructura física.

Automating ETL Process with Scripting Technology (2012) [7]

El estudio presenta una forma de automatizar el proceso ETL con scripts y mediante la ejecución por lotes, antes del auge de los servicios cloud gestionados. Aunque los scripts redujeron la intervención manual, también hicieron la administración y el mantenimiento del sistema más complejos. En este TFM la comparación es importante porque muestra cómo IICS simplifica la gestión y trazabilidad de los procesos al incluir de forma nativa la automatización que antes se tenía que crear fuera del sistema.

ETL-Based Billing System for Azure Services with Cost Estimation (2023) [8]

En esta investigación se desarrolla un sistema de facturación basado en procesos ETL en la nube de Azure, capaz de estimar los costes y analizar el consumo de recursos. El enfoque propuesto resulta especialmente útil para evaluar el coste total de propiedad (TCO) en escenarios de migración a IICS, ya que proporciona una base comparativa sólida para analizar el impacto económico del cambio tecnológico y valorar la rentabilidad de las operaciones en la nube.

From Ingestion to Action: Architecting Intelligent DataOps Pipelines for Real-Time Consumer Systems (2024) [9]

El artículo presenta una arquitectura DataOps basada en microservicios, en la que se integran mecanismos de validación en tiempo real, retroalimentación posterior y control de calidad durante el flujo de datos. Estos principios pueden extrapolarse a la metodología empleada en este trabajo, especialmente en aspectos como el tratamiento de registros duplicados y la validación automática previa a la carga en Salesforce, reforzando así el control de calidad en las fases críticas del proceso de integración.

The Agent Approach to the ETL Task (2024) [10]

Los autores presentan el modelo que usa sistemas multiagente para mejorar la flexibilidad y la limpieza de datos en los procesos ETL en el que cada agente actúa sin depender de otro, corrige errores y verifica formatos. Este modelo alinea la propuesta con las estrategias de control de calidad que se usan en la migración de PowerCenter a IICS donde la detección y corrección sin intervención automática de inconsistencias, garantiza la fiabilidad de las cargas hacia Salesforce.

Dimensions of Automated ETL Management: A Contemporary Literature Review (2024) [11]

El trabajo revisa el tema de la automatización de procesos ETL mediante el uso de técnicas de aprendizaje automático para mejorar la eficiencia y calidad del dato abriendo una línea de investigación futura, incorporando modelos de inteligencia artificial. Dichos modelos detectarían errores y duplicados con anticipación implementándose en IICS.

A Project on Statistical Analysis, ETL Automation, and Intelligence Reporting (2022) [12]

El artículo expone un proyecto de automatización para el análisis de estadísticas y la generación de informes de inteligencia para la organización Teach for India. El estudio combina procesos ETL, técnicas de visualización y herramientas de business intelligence como Salesforce y Google Data Studio optimizando la toma de decisiones a partir de volúmenes de información. A través de la automatización de la limpieza, la transformación y la representación de los datos, los autores reducen el trabajo a mano mejorando la precisión en el análisis estando así, muy relacionado con la propuesta de este TFM ya que ambos buscan integrar la automatización y el control de calidad en entornos de análisis en la nube, y con eso se promueve una gestión más eficiente y fiable de los procesos de integración de datos.

Cloud-Native LLMOps Meets DataOps: A Unified Framework for High-Volume Analytical Systems (2025) [13]

El artículo muestra un marco que integra DataOps y LLMOps en entornos nativos en la nube con el fin de mejorar la automatización, escalabilidad y rendimiento de los sistemas analíticos que usan inteligencia artificial. Los autores usan una arquitectura modular basada en microservicios y en la orquestación por eventos unificando el manejo de los datos y el uso de los modelos de lenguaje de gran tamaño. Así, los autores mejoran la carga y la transformación de la información y también la inferencia en tiempo real. Los resultados muestran una reducción del 40 % en la latencia y un aumento del 89 % en el rendimiento. El planteamiento conecta con la filosofía del TFM con la integración y la automatización de los procesos en la nube buscan la eficiencia y la gestión de los datos en los entornos de empresa.

Comparison of Commercial and Open Source ETL Tools (2024) [14]

El estudio compara herramientas ETL comerciales y herramientas ETL de código abierto evaluando la funcionalidad de las herramientas ETL, rendimiento, escalabilidad y facilidad de configuración de las herramientas. Los resultados indican que las soluciones abiertas como Talend o Pentaho pueden igualar a los productos comerciales como SSIS y también, indican que las soluciones abiertas pueden superar a los productos comerciales en algunos casos como en entornos con recursos limitados o en configuraciones no

nativas. Este análisis confirma las decisiones del TFM, mostrando que la elección de herramientas depende más de la arquitectura, integración y optimización del proceso que del tipo de licencia validando así la migración a soluciones cloud como Informatica IICS.

Cross-Domain DataOps with Federated Learning: Unlocking AI in Regulated and Consumer-Facing Systems (2025) [15]

El estudio propone un marco que combina DataOps y Federated Learning para desarrollar sistemas de inteligencia artificial escalables que cumplan las normas de privacidad en sectores regulados. La arquitectura incluye automatización, trazabilidad y auditoría de los datos garantizando así la seguridad y el cumplimiento legal en entornos distribuidos. El enfoque del estudio coincide con los objetivos del TFM destacando la gobernanza del dato y automatización de procesos en ecosistemas cloud, siendo clave para la migración hacia IICS.

Data Extraction, Transformation, and Loading (ETL) Process Automation and Data Warehouse Implementation for Algorithmic Trading Machine Learning Modelling (2025) [16]

El estudio crea un sistema que, automatizado con procesos ETL y almacenamiento de datos, optimiza modelos de aprendizaje aplicados en el trading algorítmico. La solución conecta fuentes de datos de finanzas en tiempo real, realiza transformaciones sin intervención humana y aumenta la eficiencia del modelado que genera predicciones. Este trabajo se relaciona con el TFM porque muestra que la automatización de la ingesta y la transformación de datos, junto con un diseño de arquitectura que puede crecer, es una necesidad para lograr cargas optimas y análisis precisos en entornos cloud como IICS.

Incremental Load Processing on ETL System through Cloud (2022) [17]

El estudio propone un sistema de carga incremental en la nube el cual mejora la eficiencia de los procesos ETL con la actualización de los registros que cambian, en vez de recargar todo el conjunto de datos. El modelo reduce el tiempo de ejecución y optimiza el uso de recursos con algoritmos de captura de cambios en entornos distribuidos. El enfoque se relaciona con el TFM mostrando cómo la automatización y la carga incremental, implementadas en plataformas cloud como IICS, mantienen la disponibilidad y la precisión de los datos sin afectar el rendimiento del sistema.

Requirements for DataOps to Foster Dynamic Capabilities in Organizations – A Mixed Methods Approach (2022) [18]

El estudio mira cómo la adopción de las prácticas DataOps puede fortalecer las capacidades de las organizaciones que se adaptan al fomentar la agilidad, la eficiencia y la calidad en la gestión del dato. Con un enfoque que combina métodos, los autores identifican los requisitos que se necesitan en estructura, tecnología y personas, resaltando la importancia de la automatización, colaboración y mejora continua. En este sentido la automatización contribuye a acelerar los procesos, la colaboración favorece la comunicación entre equipos y la mejora continua impulsa la innovación. Este enfoque se vincula con el TFM al coincidir en que la implementación de metodologías DataOps, como las integradas en IICS, impulsa la capacidad de adaptación y la toma de decisiones basadas en datos dentro de entornos empresariales en constante cambio.

Research on DataOps Capability: Practice and Development (2023) [19]

El trabajo estudia la construcción y desarrollo de las capacidades DataOps mostrando cómo las organizaciones pueden madurar los procesos de gestión de datos por automatización, integración continua, y colaboración entre los equipos técnicos y de negocio. Los autores presentan un modelo de práctica que permite evaluar el grado de madurez DataOps y muestra el impacto en su eficiencia operativa. Esta investigación está relacionada con el TFM reforzando la importancia de consolidar entornos de integración y despliegue automatizados como los que habilita IICS ayudando a mejorar la calidad y la gobernanza del dato en los ecosistemas cloud.

Zero-ETL Architectures for AI Workloads: Direct ML Model Access on Cloud-Native OLAP Systems (2025) [20]

El estudio propone una arquitectura que elimina las fases de extracción, transformación y carga con modelos de aprendizaje automático que acceden directamente a los datos en los sistemas OLAP nativos de la nube. Esa arquitectura reduce la latencia, los costos operativos y la complejidad del mantenimiento. Además, integra procesamiento en tiempo real y computación serverless. El planteamiento está relacionado con este TFM porque muestra la tendencia a usar arquitecturas de integración que funcionan sin intervención manual, similares a las de Informatica IICS donde el tratamiento de los

datos, la inmediatez y la eficiencia son los ejes del rendimiento de los analítico.

Los estudios más estrechamente vinculados con este TFM son Evolution of ETL Tools [6] y ETL-Based Billing System for Azure Services [8], ya que contextualizan el paso de arquitecturas on-premise a soluciones cloud como IICS y permiten comparar su impacto en eficiencia y costes. También resultan especialmente relevantes From Ingestion to Action [9] y The Agent Approach to the ETL Task [10], por su aportación a la automatización del control de calidad y la gestión de duplicados, aspectos clave en la carga hacia Salesforce. Finalmente, trabajos como Incremental Load Processing on ETL Systems through Cloud [17] y Dimensions of Automated ETL Management [11] refuerzan la importancia de la carga optimizada y de las prácticas DataOps, elementos centrales en el enfoque adoptado en este proyecto.

2.1.1 Evolución de los procesos ETL y su papel en la integración de datos.

A medida que las organizaciones han ido generando cada vez más información, también ha crecido la necesidad de poder mover, transformar y combinar esos datos para que sean útiles. Al principio, las empresas resolvían este problema de forma muy manual: leyendo ficheros, haciendo transformaciones con scripts o cargando datos a mano en bases de datos o sistemas analíticos. Este enfoque hacía que las organizaciones fueran muy dependientes de un equipo técnico, poco flexible y difícil de mantener en el tiempo [1].

Para cubrir esta necesidad de forma más estructurada, surgieron los primeros procesos conocidos como ETL, siglas que hacen referencia a Extract, Transform, Load. En la práctica, esto significa extraer los datos desde su lugar de origen (una base de datos, un fichero, una API...), aplicarles una serie de transformaciones (validaciones, conversiones, cálculos, etc.) y finalmente cargarlos en su destino, normalmente un almacén de datos o sistema de análisis. [21]

Este modelo ayudó a estandarizar cómo se gestionaba la integración de datos dentro de una organización donde en lugar de tener múltiples desarrollos sueltos, el enfoque ETL permitía construir procesos centralizados, más fáciles de controlar y con mayor trazabilidad. Su uso se extendió rápidamente en sectores donde los datos son fundamentales, como la banca, el comercio, los seguros o las telecomunicaciones. [7]

Con el paso del tiempo, los procesos ETL han ido evolucionando de forma significativa. Al principio se utilizaban únicamente como una solución técnica para mover los datos desde un punto A hasta un punto B. Sin embargo, en la actualidad se han convertido una

pieza clave en cualquier estrategia de gestión de la información. Los sistemas ETL aseguran que los datos lleguen al destino con el formato correcto, con la calidad adecuada y en el momento preciso pudiendo ser usados por los sistemas de negocio, cuadros de mando o algoritmos de análisis.

Además, el papel del proceso ETL ha crecido con la llegada del Business Intelligence ya que toda herramienta de análisis necesita datos limpios y estructurados para dar resultados válidos. El proceso ETL debe asegurar la calidad y consistencia de los datos puesto que si no lo hiciera los análisis perderían su valor, aunque la herramienta fuese muy buena.

En definitiva, el ETL ha pasado de ser una solución puntual para mover datos, a convertirse en un engranaje esencial en todo el ciclo de vida de la información dentro de las empresas. Su evolución ha ido en paralelo al crecimiento de la complejidad de los entornos de datos y al aumento de la necesidad de tomar decisiones basadas en información precisa y actualizada.

2.1.2 Enfoques tradicionales vs. modernos: PowerCenter frente a IICS

A la hora de hablar de integración de datos, una de las comparaciones más habituales es entre herramientas tradicionales que se ejecutan en infraestructura local, como PowerCenter, y soluciones más recientes pensadas directamente para la nube, como Informatica IICS. Aunque ambas pertenecen al mismo fabricante y comparten muchas bases conceptuales, en la práctica ofrecen formas bastante distintas de enfocar los procesos de integración.

PowerCenter lleva muchos años en el mercado y ha sido una de las plataformas más utilizadas en entornos empresariales donde se requiere estabilidad, trazabilidad y control. Se instala en servidores internos, depende de infraestructura propia y requiere de mantenimiento técnico por parte del equipo de sistemas. A cambio, ofrece un entorno maduro, con muchísimas funcionalidades, transformaciones complejas y una forma muy detallada de diseñar cada proceso ETL [1][4].

En este enfoque, todo se ejecuta en local: se extraen los datos, se transforman según las reglas definidas en el mapping, y se cargan una vez están listos. Es un modelo robusto y que da buenos resultados cuando los sistemas están dentro de la misma red o cuando el volumen de datos no requiere elasticidad constante [23].

Por otro lado, IICS (Informatica Intelligent Cloud Services) es la evolución lógica de esta filosofía, adaptada a los entornos cloud. Aquí ya no es necesario instalar ni mantener servidores locales. Toda la ejecución puede realizarse en la nube, y las conexiones con

orígenes y destinos se hacen de forma más directa simplificando mucho el despliegue y permitiendo que distintos equipos trabajen desde ubicaciones distintas sin necesidad de acceder a una red interna [24].

A nivel de diseño, IICS sigue una lógica muy parecida a PowerCenter ya que se definen flujos visuales, mappings y reglas de transformación. La gran diferencia está en la forma de ejecutarse, en la flexibilidad del entorno y en su capacidad para integrarse con servicios como Salesforce, bases de datos cloud o plataformas de almacenamiento distribuido. Además, al estar concebida para trabajar online, facilita la automatización, la escalabilidad dinámica y el acceso a nuevas funcionalidades que se van incorporando sin necesidad de actualizaciones manuales [4][5].

Este cambio de modelo también está en línea con una tendencia más amplia en el mundo de la integración de datos: el paso de los procesos ETL tradicionales hacia enfoques más orientados al ELT, donde la transformación ocurre en el propio destino. Aunque tanto PowerCenter como IICS permiten seguir el modelo ETL, es más natural en la versión cloud mover cargas hacia el sistema de destino y procesarlas allí, sobre todo cuando se trabaja con almacenes de datos modernos [2].

En definitiva, PowerCenter representa el enfoque clásico, fiable y muy detallado, mientras que IICS se adapta mejor a las necesidades actuales de flexibilidad, escalabilidad y conexión con servicios en la nube. Ambas herramientas responden a contextos distintos, y conocer sus diferencias es clave para elegir la más adecuada según el tipo de proyecto o infraestructura disponible.

2.1.3 Herramientas más relevantes en el mercado

El mundo de la integración de datos ha evolucionado mucho, y con él, la variedad de herramientas disponibles. Hoy en día, las empresas no solo buscan mover datos de un sitio a otro, sino hacerlo con garantías de calidad, en tiempos razonables y con capacidad para adaptarse rápidamente si el negocio cambia. Esa necesidad ha hecho que el mercado se llene de soluciones con distintos enfoques, cada una con sus particularidades.

Una de las más conocidas, sobre todo en grandes compañías, es Informatica PowerCenter, siendo una opción sólida para entornos donde se requiere control y trazabilidad, y se valora mucho su estabilidad. En empresas donde los procesos llevan años funcionando y ya están bien definidos, PowerCenter suele estar muy asentado aunque por su dependencia de infraestructura local y su menor flexibilidad, en algunos casos empieza a quedar un poco limitada frente a otras opciones más modernas [4].

En ese sentido, Informatica IICS ha ganado terreno en los últimos años, especialmente en organizaciones que están migrando hacia arquitecturas en la nube. Su gran ventaja es que conserva la lógica de trabajo que ya conocen los usuarios de PowerCenter, pero eliminando la parte de administración del sistema reduciendo tiempos y facilitando la gestión [5]. Además, al estar pensada para funcionar directamente online, se conecta sin problema con otros servicios en la nube, como Salesforce o Amazon Redshift, por citar algunos [25].

Otra herramienta muy presente en el mercado actual es Talend, que ha sabido posicionarse bien como solución de código abierto, pero también con versiones comerciales más completas. Su propuesta se basa en ofrecer una gran flexibilidad, permitiendo tanto procesos ETL clásicos como flujos más complejos siendo habitual encontrarla en proyectos donde se busca algo más personalizable, sobre todo cuando los equipos tienen cierta experiencia técnica [26].

También hay que mencionar Azure Data Factory y AWS Glue, dos servicios ofrecidos por Microsoft y Amazon respectivamente. Están pensados para integrarse de forma natural en sus propios ecosistemas cloud, lo que los hace muy prácticos si ya se trabaja en esas plataformas. A nivel funcional ofrecen lo necesario para construir procesos de transformación, aunque su diseño suele estar más orientado a perfiles técnicos, y no tanto a usuarios de negocio [27] [28].

Otra alternativa que está creciendo en popularidad es Matillion, sobre todo entre empresas que trabajan con almacenes de datos en la nube como Snowflake. Tiene un enfoque visual, con una interfaz clara y está muy centrada en procesos ELT, encajando bien con la filosofía moderna de cargar primero los datos y transformarlos luego directamente en el destino [29].

En definitiva, no hay una herramienta que sea “la mejor” para todos los casos. Cada una tiene sus puntos fuertes, y lo importante es elegir en función de lo que necesita el proyecto: si se busca robustez y continuidad PowerCenter puede seguir siendo útil, si se prioriza agilidad, integración con la nube y despliegue rápido, IICS o Matillion pueden encajar mejor. Lo que está claro es que el mercado ofrece alternativas para casi cualquier situación, y saber moverse entre ellas es clave para adaptarse al ritmo con el que evoluciona todo lo relacionado con los datos.

2.1.4 Tendencias actuales en la integración de datos: cloud, automatización, IA

La forma de integrar y trabajar con datos no ha dejado de evolucionar, y en los últimos

años han aparecido nuevas tendencias que están marcando el camino hacia donde se dirige todo este ámbito. Algunas de estas transformaciones están ligadas a los avances tecnológicos, mientras que otras vienen impulsadas por cambios en cómo las empresas entienden el valor de sus datos [30].

Uno de los cambios más importantes ha sido la transición hacia entornos en la nube donde cada vez, más compañías están dejando atrás infraestructuras propias y optando por plataformas que se gestionan directamente online. Esto no solo reduce los costes de mantenimiento, sino que también permite escalar con más facilidad y adaptarse a picos de demanda sin tener que hacer inversiones en hardware. Además, trabajar en la nube hace mucho más sencillo conectar distintas aplicaciones y fuentes de datos, que ya no siempre están en el mismo lugar físico.

Otra tendencia clara es la automatización ya que antes, muchas tareas dentro de un proceso ETL dependían de intervenciones manuales o de programación específica. Ahora, cada vez es más común que las herramientas ofrezcan opciones para que ciertas acciones se ejecuten de forma automática, como por ejemplo detectar nuevos datos, lanzar procesos en función de horarios o condiciones, o incluso corregir errores simples sin que nadie tenga que intervenir permitiendo ahorrar tiempo y reducir el riesgo de errores humanos.

También se empieza a notar la influencia de la inteligencia artificial. Aunque aún no está totalmente integrada en todos los sistemas ETL, ya hay herramientas que utilizan algoritmos para sugerir transformaciones, detectar patrones en los datos o anticiparse a problemas. En algunos casos, incluso se están desarrollando asistentes que ayudan a diseñar mappings o a validar la calidad del dato sin que sea necesario definir cada regla de forma explícita [31].

Más allá de estas tendencias técnicas, hay también un cambio de mentalidad. Ahora no se trata solo de mover datos, sino de hacerlo de forma rápida, flexible y con una visión clara de para qué se hace. Se busca integrar datos no solo para almacenarlos, sino para analizarlos, compartirlos o incluso actuar sobre ellos casi en tiempo real. Esto ha hecho que conceptos como DataOps, que mezcla ideas del desarrollo ágil con la gestión de datos, vayan ganando peso poco a poco [32].

En conjunto, estas tendencias están llevando a que la integración de datos deje de ser un proceso cerrado y técnico, para convertirse en una parte dinámica del funcionamiento de cualquier organización. La nube, la automatización y la inteligencia artificial no solo están cambiando las herramientas, sino también la forma de entender lo que significa

realmente integrar datos hoy.

2.2 Técnica empleada

2.2.1 Contexto funcional y descripción del caso real.

El presente trabajo parte de un escenario real dentro del sector asegurador, en el que la compañía implicada realiza campañas comerciales orientadas a la retención de clientes. Estas campañas, clasificadas como "Outbound", consisten en contactar de forma proactiva a asegurados con el objetivo de ofrecerles productos personalizados, reforzar la relación comercial y, en última instancia, evitar la fuga hacia otras entidades.

La solución funcional definida por el equipo de análisis establecía que toda la gestión de estas campañas debía centralizarse en Salesforce como BBDD. En este entorno, se cargarían tanto las campañas como los contactos vinculados a cada una de ellas, utilizando componentes específicos dentro del módulo de campañas Outbound. Esta operativa permite identificar el público objetivo, hacer seguimiento del estado de cada usuario, y actuar rápidamente si un cliente contacta con el centro de atención mientras tiene una campaña activa. Es decir, el objetivo principal es facilitar al agente asegurador la información de los clientes con campañas activas con fechas próximas a finalizar para poder retener al cliente y renovar la póliza.

El proceso que se toma como referencia para este trabajo surge precisamente de esa necesidad funcional. Se identificó que, durante la preparación de los ficheros de carga, un mismo cliente podía aparecer varias veces en una misma campaña debido a que tenía contratadas varias pólizas. Este hecho podía derivar en una carga redundante dentro del CRM, generando múltiples entradas para el mismo asegurado en la misma campaña, algo que se quería evitar.

Por este motivo, se diseñó un flujo que contemplara la detección de duplicados a partir de los campos clave CampaignId (el identificador de la campaña), ContactId (el identificador del contacto del cliente) y PolicyId (el identificador de la póliza). En los casos en los que un contacto figurara más de una vez dentro de la misma campaña con diferentes pólizas, el sistema debía registrar solo la primera entrada en Salesforce ya que la tabla donde se van a insertar los datos solo contempla un registro por cliente, mientras que el resto de registros se debían derivar a un fichero plano para su tratamiento posterior o revisión manual. Este enfoque permitía mantener la integridad de los datos en el CRM y evitaba conflictos en la ejecución de campañas automatizadas.

Si no existe duplicidad en el impacto del cliente en la campaña, se cargará en Salesforce un único registro de "Campaign Member" siguiendo el siguiente esquema:

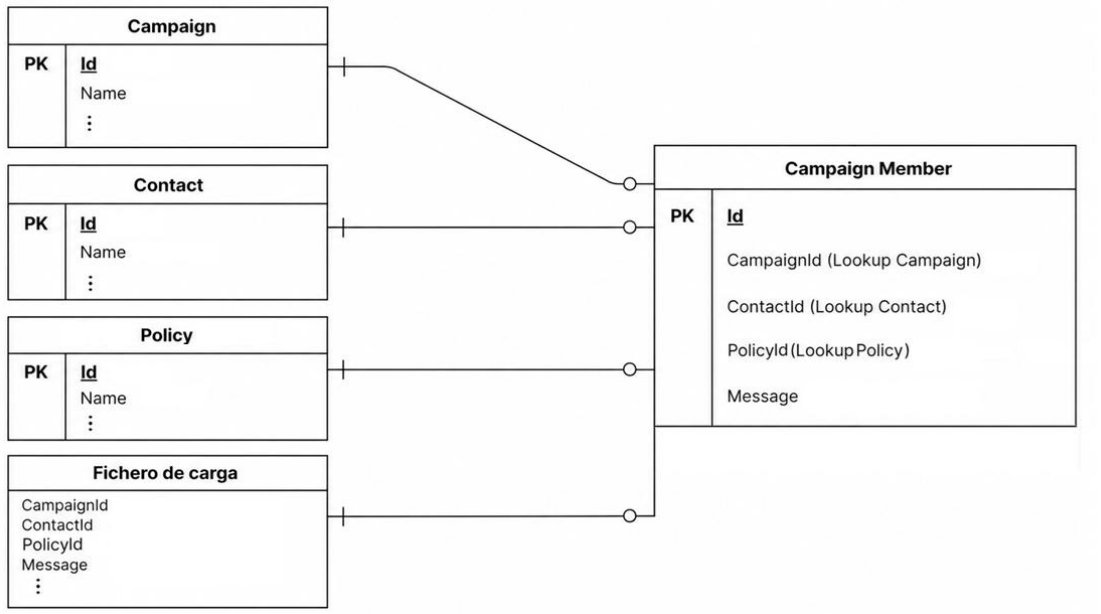


FIGURA 1: Esquema de carga cuando un cliente no tiene duplicidad

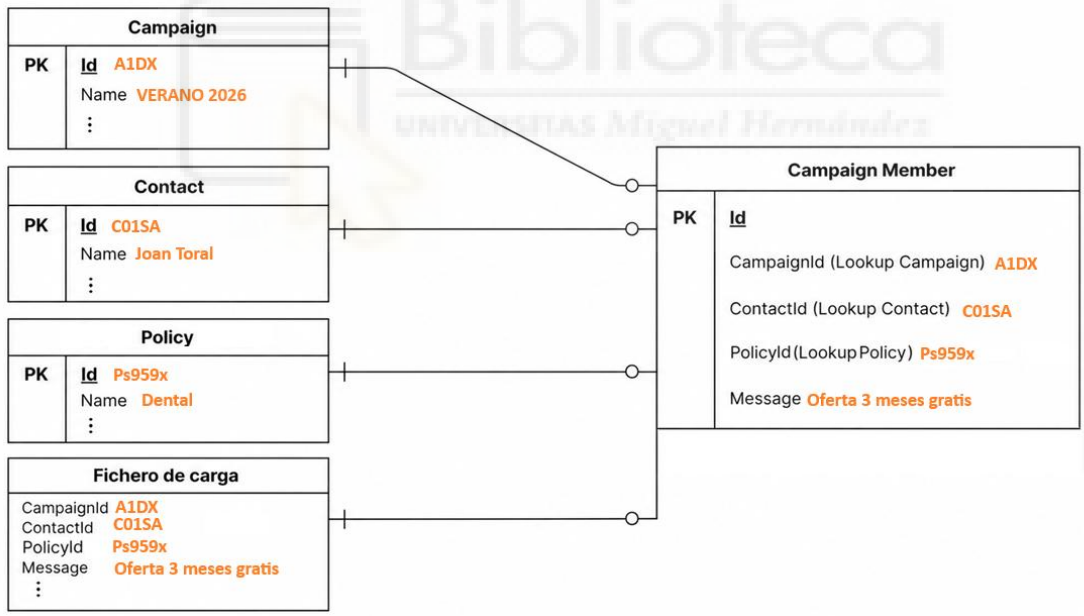


FIGURA 2: Esquema de carga cuando un cliente no tiene duplicidad, ejemplo visual.

En la Figura 2 se puede observar un ejemplo visual de lo que debería hacer la ETL. Si en el fichero de carga viene un registro con un cliente (Joan Toral) que solo tiene una póliza (Dental) y se va a ejercutar la campaña VERANO 2026, en la tabla de miembros de campaña ("Campaign Member") se insertarán el identificador de la campaña, el identificador del contacto del cliente y el identificador de la póliza.

Como se ha mencionado antes, la tabla de miembros de campaña solo debe tener un identificador (Id) por campaña y contacto. En caso de recibir varias pólizas para un mismo cliente, la información de las pólizas se deberá cargar en la tabla de detalles de miembro de campaña, "Campaign Member Detail".

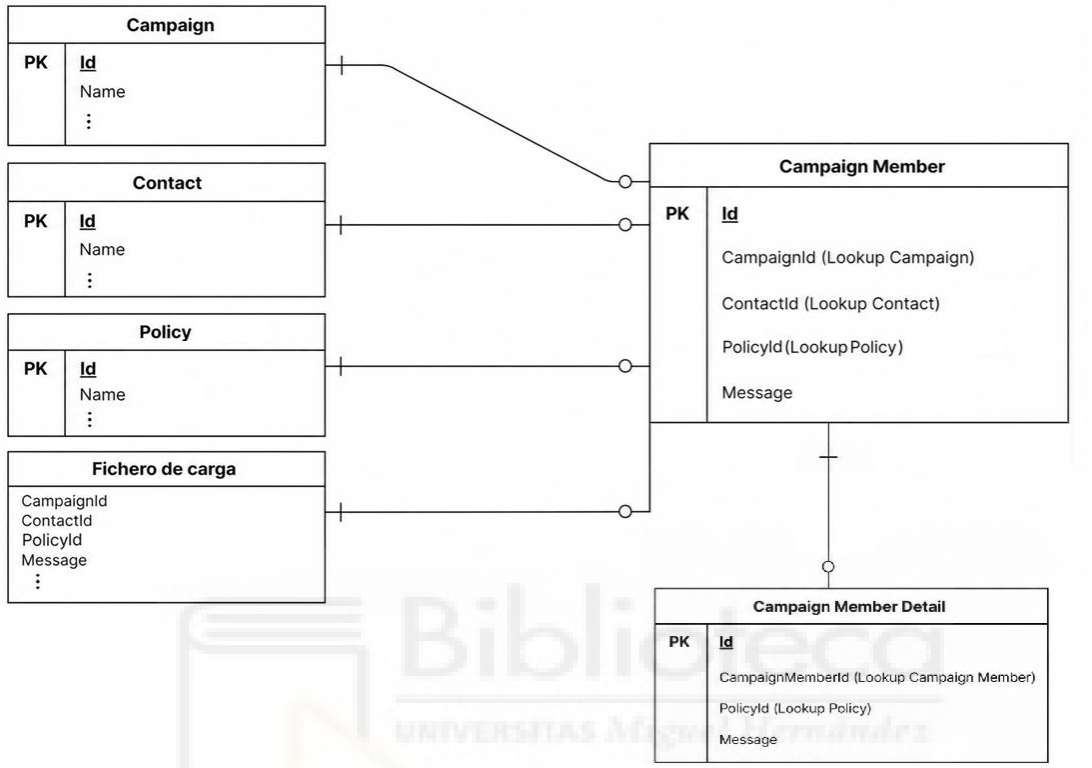


FIGURA 3: Esquema de carga cuando un cliente tiene duplicidad

Cuando en el mismo fichero se recibe información de un cliente para la misma campaña pero con varias pólizas, en la tabla de miembro de campaña se insertará la información de la campaña y del cliente, pero para evitar duplicados se deja en blanco la oferta y las pólizas, esta información se añadirá en la tabla de detalles de miembro de campaña e irá asociado con el id de la tabla Campaign Member.

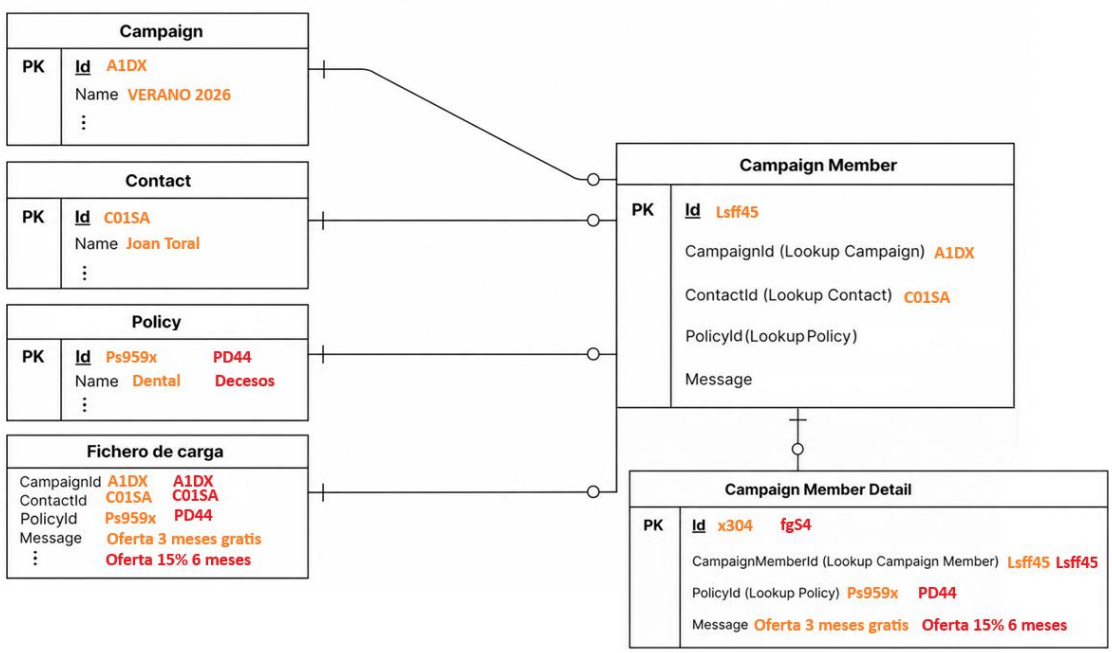


FIGURA 4: Esquema de carga cuando un cliente tiene duplicidad, ejemplo visual

En la imagen se muestran dos registros recibidos en el fichero de carga, el primero de color naranja y el segundo de color negro. Ambos coinciden que pertenecen a la misma campaña y cliente, pero tienen diferentes id de póliza. El primero corresponde a una póliza de dental y el segundo en una de decesos por lo que en la tabla de miembro de campaña solo se inserta un registro ya que el cliente pertenece a la misma campaña, VERANO 2026. Este sería el primer paso y en un segundo paso se cargarían en la tabla de detalles de miembro de campaña, que el cliente tiene dos pólizas y en cada una la oferta que ha recibido.

2.2.2 Simulación de una ETL manual

Antes de implementar el proceso con herramientas especializadas, se ha llevado a cabo una simulación manual utilizando el lenguaje de programación C. Esta elección no ha sido casual, aunque hoy en día existen lenguajes más modernos y herramientas específicas para manipulación de datos, el uso de C permite observar con claridad lo que ocurre “por debajo”, es decir, sin capas de abstracción que oculten los detalles del procesamiento.

El objetivo de esta simulación era reproducir la lógica de negocio del caso real de forma artesanal, gestionando cada paso del flujo de datos desde cero. Esto incluía: leer el archivo de origen, analizar los registros, detectar los duplicados y generar dos salidas diferentes con formatos específicos. Aunque parece simple, este proceso muestra las dificultades que las herramientas ETL modernas resuelven de forma nativa.

El código se escribió con estructuras simples, primero abre el fichero de entrada (f_campana_outbound_rel_poliza_PROCESO_MANUAL.csv), guarda los registros en la memoria y después los procesa. El código usa una estructura tipo Record para guardar los campos de cada línea, también incluye un mecanismo para detectar cuando un cliente con el mismo código de campaña aparece más de una vez, usando esta condición como criterio para identificar el registro como duplicado. Si el registro es duplicado, el programa vacía los campos de Poliza, Descripcion_producto y Oferta_Retencion al crear el archivo de destino para Salesforce.

El proceso también crea otro fichero de salida que contiene todos los registros de los duplicados y guarda todas las combinaciones que existen sin modificar los datos originales.

Uno de los retos más importantes fue garantizar la limpieza del formato, ya que errores como saltos de línea no deseados, comas mal posicionadas o registros vacíos provocaban salidas incorrectas. También fue necesario implementar una segunda función que limpiara el fichero final y corrigiera casos como comas iniciales no deseadas.

Otro aspecto crítico fue el control de la memoria ya que al no contar con herramientas que gestionan automáticamente el tratamiento de datos, todo el flujo se sostuvo sobre arrays estáticos, con una limitación deliberada para facilitar el desarrollo y mantener el foco en la lógica de negocio más que en la optimización a gran escala.

Aunque el código acabó funcionando de forma correcta y produjo los resultados esperados, esta simulación sirvió para comprobar que, si bien es posible desarrollar una ETL a mano, hacerlo conlleva mucho más esfuerzo, riesgo de error, y menor flexibilidad ante cambios. Esta experiencia dejó claro el valor que aportan las herramientas específicas, tanto en diseño como en ejecución, y preparó el terreno para comparar lo manual con lo profesional de forma justa.

2.2.3 Implementación del proceso en Powercenter

Una vez definida la lógica del proceso en su versión manual, se ha trasladado el mismo planteamiento a un entorno profesional utilizando Informatica PowerCenter, una de las herramientas más veteranas y extendidas en el mundo de la integración de datos.

El objetivo ha sido implementar de forma visual y estructurada un flujo que parte del mismo fichero de origen y aplique las mismas reglas de negocio: identificar registros únicos por cliente y campaña para su envío a Salesforce, y guardar en un fichero plano todos aquellos casos donde se detecten múltiples pólizas para el mismo cliente dentro de

una misma campaña.

El diseño del mapping comienza con la lectura del fichero de carga a través de un Source Qualifier en el cual, a partir de ahí se incorpora un Sorter que ordena los registros por combinación de IdCampaña, ContactId y PolicyId, lo que permite preparar los datos para la agrupación posterior. Esa agrupación se realiza mediante un Aggregator, en el que se detecta si un mismo cliente tiene más de una póliza mediante una lógica basada en el uso combinado de funciones como MAX y MIN. Si ambas devuelven resultados diferentes, se interpreta que existen varios productos asociados al mismo cliente y campaña [33][34]. Los registros que cumplen esta condición se marcan con un indicador, y se pasa a un Joiner que permite recuperar todos los campos originales (antes de haber realizado la agrupación y el ordenado de datos). A partir de ese punto, los datos se dividen en dos flujos en el que uno de ellos, el orientado a Salesforce, se filtra y transforma para dejar vacíos los campos relacionados con pólizas y ofertas en los casos en que hay duplicados. El otro flujo, destinado al fichero plano, conserva todos los registros originales y genera una salida completa para auditoría.

Para comprender mejor el contexto en el que se enmarca este desarrollo, a continuación se muestra un diagrama de flujo que representa el diseño completo del proceso de campañas outbound. Este flujo está compuesto por cuatro mappings distintos que actúan de forma coordinada.

La parte resaltada en azul dentro del diagrama corresponde al mapping m_SCA_campanas_outbound_miembro_campana, que es el que se analiza y detalla en este apartado. Como se observa, este subproceso es responsable de la lógica que identifica si un cliente participa con una o varias pólizas en una campaña determinada, y dirige los datos al destino adecuado según corresponda.

Esta visión de conjunto permite situar el mapping en su entorno funcional real y apreciar cómo se conecta con el resto del ecosistema de integración.

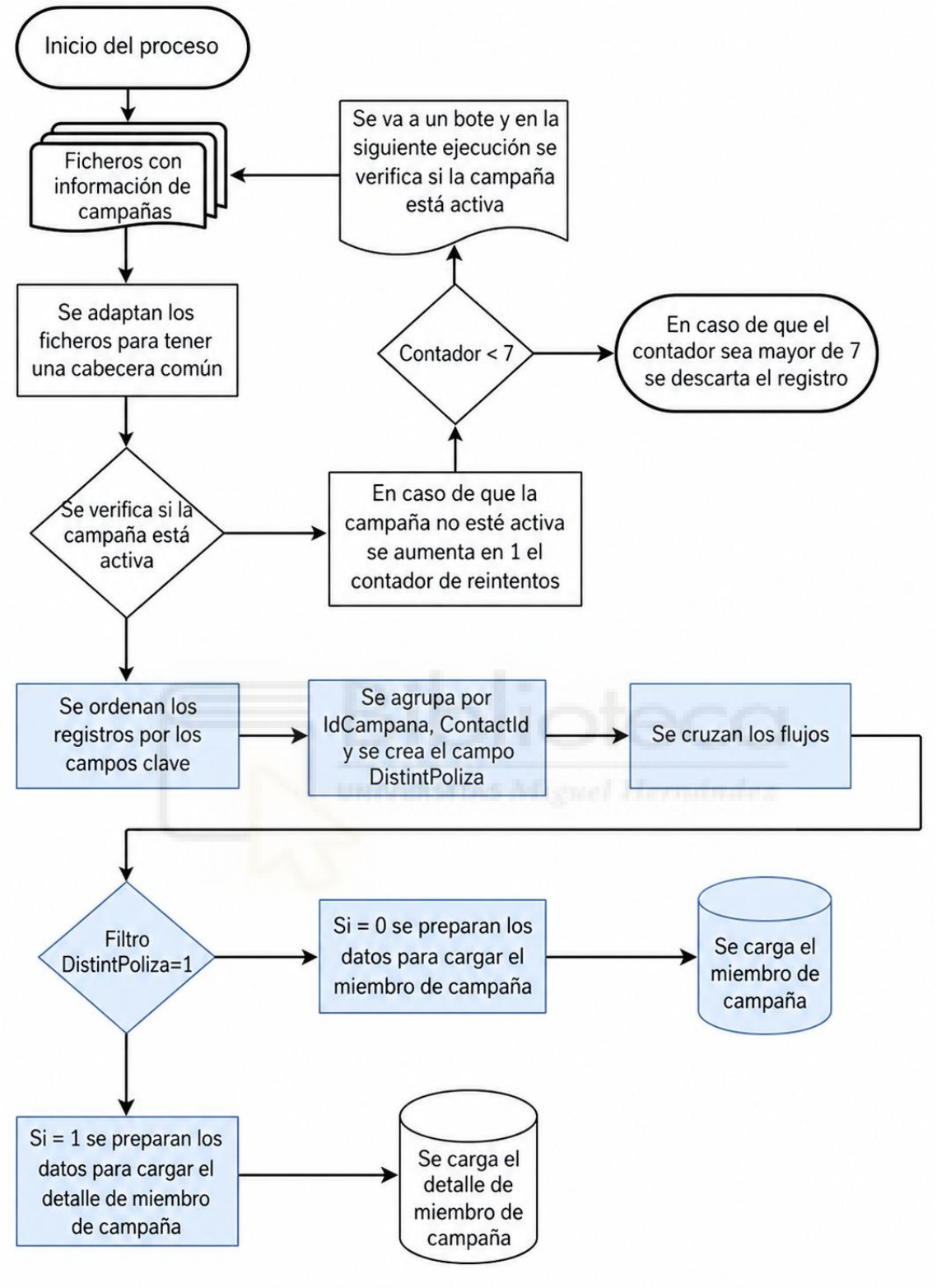


FIGURA 3: Diagrama general del proceso de campañas outbound.

Durante el proceso, la ETL utilizó transformaciones tipo Expression para formatear los valores, aplicar las condiciones con las funciones IIF y agrupar los datos antes de la carga. La interfaz de PowerCenter muestra el flujo paso a paso ayudando a comprender el flujo, mantenerlo y a validarlo en los entornos de producción. [33][34].

Una ventaja de esta implementación es la claridad visual del diseño en el que se muestran los objetos interconectados, facilitando al usuario ver si algo no está en su sitio y modificar una transformación con un doble clic, ajustando los parámetros, siendo diferente a navegar por funciones y estructuras en el lenguaje C ya que este te obliga a revisar el código.

Además, PowerCenter permite registrar cada ejecución, guardar logs, lanzar procesos bajo planificación y reutilizar mappings en otros proyectos. Todas estas capacidades aportan un nivel de fiabilidad y escalabilidad que resulta difícil alcanzar cuando se trabaja con scripts o desarrollos desde cero.

En definitiva, esta implementación demuestra cómo una herramienta ETL tradicional permite replicar con precisión la lógica diseñada a mano, pero con una carga de trabajo mucho menor y con una mayor facilidad para mantener y adaptar el proceso con el tiempo.

2.2.4 Implementación equivalente en IICS

Una vez desarrollada la solución en PowerCenter, el siguiente paso ha sido trasladar el mismo proceso a Informatica IICS, que es la versión en la nube de esta herramienta. La idea ha sido replicar exactamente la misma lógica de negocio, pero en un entorno con arquitectura cloud, para comprobar cómo se comporta en términos de diseño, ejecución y funcionalidad general.

A nivel de diseño, IICS conserva muchas similitudes con PowerCenter, lo cual facilita bastante la transición siendo el concepto de mappings, transformaciones y flujos visuales similar, aunque con una interfaz más moderna y simplificada. La creación de mappings se realiza a través de la sección de Mapping de IICS, y en este caso se ha utilizado también un fichero plano como fuente, cargado mediante un Source con formato delimitado.

Tal como se hizo en la versión anterior, en IICS se incorporan funciones para ordenar los datos y agruparlos. La agrupación se realiza con un Aggregator Transformation, en el que se comprueban las combinaciones de IdCampana y ContactId para detectar si hay más de una póliza por cliente dentro de una misma campaña. Esta transformación permite generar un campo nuevo que actúa como un indicador que detecta si un cliente debe considerarse duplicado o no [33].

A partir de ahí, los datos se ramifican: una de las salidas se prepara para Salesforce, vaciando los campos relacionados con Poliza y OfertaRetencion si se ha detectado duplicados, mientras que la otra mantiene todos los registros tal cual, generando una copia

completa para trazabilidad. Estas transformaciones se resuelven con expresiones condicionales y filtros configurados de forma visual, sin necesidad de escribir código manual.

Una diferencia importante con PowerCenter es la forma de usarlo ya que en IICS todo se maneja desde el navegador, pudiendo así entrar al entorno de trabajo desde cualquier lugar sin depender de una red interna ni de un cliente instalado. También, las actualizaciones y las mejoras se aplican automáticamente, sin que tengas que manejar parches o versiones en tu ordenador.

IICS facilita la conexión con los servicios de terceros en la nube por lo tanto, usuario configura el destino de Salesforce directamente desde la interfaz sin necesidad de instalar controladores ni definir las conexiones, en consecuencia, se acelera el despliegue y la integración con las plataformas de terceros, reduciendo el tiempo y evitando complicaciones. [35].

Otro aspecto que se ha valorado es la facilidad para supervisar la ejecución de los mappings ya que IICS cuenta con paneles de seguimiento que permiten ver en tiempo real si el proceso ha terminado correctamente, cuántos registros ha procesado, o si se ha producido algún error. Este control integrado, que en una solución manual habría que programar aparte, aporta mucha tranquilidad y reduce los tiempos de diagnóstico.

En resumen, la implementación en IICS ha permitido comprobar que se puede reproducir exactamente el mismo comportamiento que en PowerCenter, pero con las ventajas propias de un entorno moderno, más ligero y más conectado. Aunque ambos permiten resolver el problema de forma eficaz, la experiencia de uso y la facilidad de integración de IICS hacen que esta opción resulte más ágil en escenarios donde se trabaja con sistemas en la nube o se requiere flexibilidad en el despliegue.

2.2.5 Consideraciones funcionales y técnicas en ambos entornos

Después de haber implementado el mismo proceso en PowerCenter e IICS, resulta evidente que, aunque ambas herramientas permiten alcanzar el mismo objetivo, lo hacen desde perspectivas muy distintas en cuanto a funcionamiento, gestión técnica y experiencia de uso. A nivel funcional, el diseño de los mappings sigue una lógica parecida: se definen los orígenes, se aplican transformaciones y se direccionan los datos hacia los destinos correspondientes, sin embargo, los detalles prácticos de cada plataforma condicionan mucho la forma en la que se desarrolla y mantiene el flujo de datos [4] [5].

En PowerCenter, todo el entorno se encuentra instalado en servidores locales o virtuales dentro de la infraestructura de la empresa. Esto implica que la gestión de usuarios, actualizaciones, planificación de tareas y conectividad entre módulos dependen del equipo técnico interno. En muchos casos, esto da una sensación de control absoluto, pero también exige dedicar tiempo a tareas que, en herramientas modernas como IICS, están automatizadas o gestionadas desde el propio entorno cloud [4] [5].

Además, la propia interfaz de PowerCenter, aunque sólida, tiene un diseño más clásico y menos intuitivo en comparación con IICS. En este último, las transformaciones son más visuales y las acciones más directas, lo que facilita que incluso perfiles no técnicos puedan comprender o revisar un proceso sin necesidad de conocer la herramienta en profundidad [4] [5].

Desde el punto de vista del rendimiento, ambos entornos son capaces de manejar volúmenes considerables de datos. Sin embargo, IICS ofrece una ventaja, tiene una arquitectura distribuida y en la nube que permite que el sistema escale automáticamente según la demanda por eso, IICS no requiere preparar servidores antes, tampoco necesita preocuparse por cuellos de botella cuando el volumen de datos sube de repente. En cambio, PowerCenter necesita planificación de infraestructura, y su capacidad depende de cómo esté configurado el entorno donde se ejecuta [4] [5].

En cuanto al mantenimiento, PowerCenter requiere una mayor intervención del equipo de sistemas, actualizaciones, copias de seguridad, logs, configuraciones... todo esto debe gestionarse desde dentro. En IICS, gran parte de estas tareas están delegadas a la plataforma y se realizan de forma automática, en consecuencia esto no solo ahorra tiempo, sino que también reduce los riesgos asociados a errores humanos o configuraciones incorrectas [4] [5].

Otro punto relevante es la integración con otras plataformas. IICS está preparado de forma nativa para conectarse con servicios en la nube como Salesforce, Google BigQuery o Snowflake y en muchos casos, basta con seleccionar el conector y autenticarse para empezar a trabajar mientras que, en PowerCenter estas integraciones son posibles pero suelen requerir configuraciones adicionales, drivers específicos o incluso módulos externos [4] [5][35].

En el entorno original que se desarrolló con PowerCenter, la conexión a Salesforce se hacía con un conector externo que no era nativo y eso obligaba a poner varias configuraciones a mano en cada entorno, ya sea desarrollo, preproducción o producción. Una de esas configuraciones importantes era añadir el parámetro:

JVMOption1=-Djava.io.tmpdir=\\pretcappwc001\POWERCENTER\temp

Este parámetro se especificaba dentro del archivo de configuración del Integration Service de PowerCenter y servía para permitir la descarga de metadatos de Salesforce en una carpeta de red compartida. El proceso pedía que la carpeta tuviera permisos, la ruta existiera y se validara en cada servidor en el que cada vez que se hacía una promoción de entornos, la configuración cambiaba dependiendo de si estabas en desarrollo, preproducción y producción por lo que el equipo debía ajustar los valores manualmente.

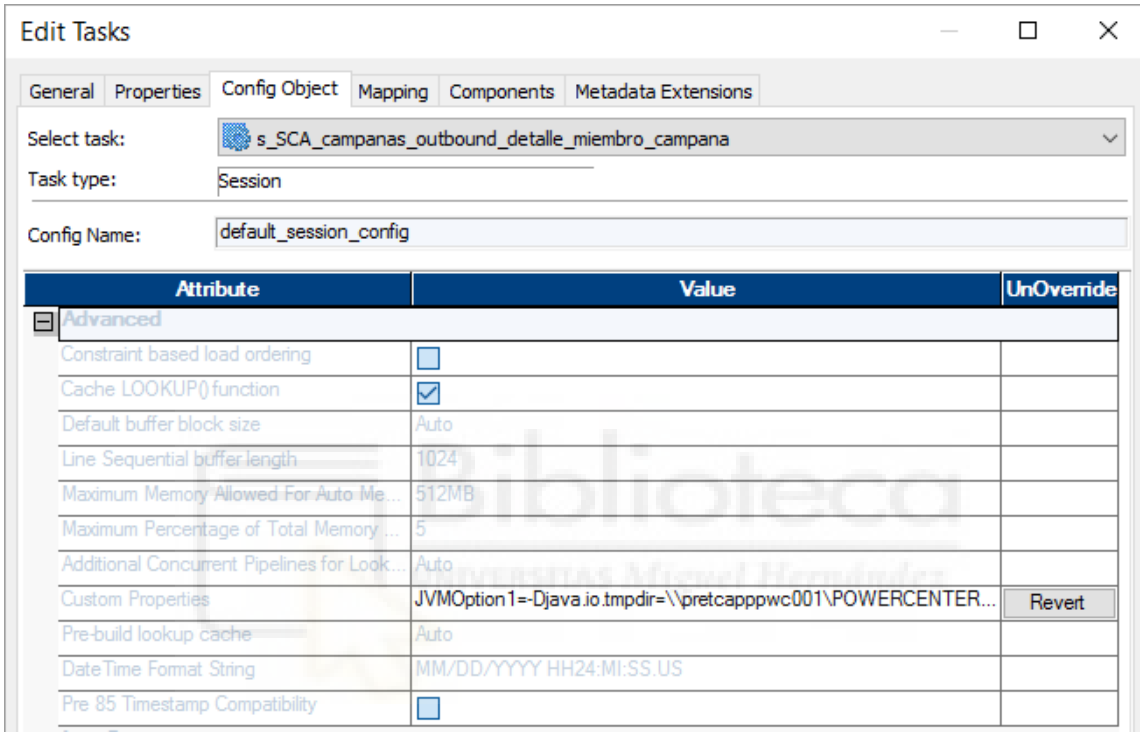


FIGURA 4: Configuración de la conexión Powercenter-Salesforce.

Este método genera más errores humanos y ha provocado incidencias en el uso real del proceso ya que en varias ocasiones se han producido pérdidas de conexión con Salesforce mientras se ejecutaban los mappings debido a que no estaba bien configurada la ruta provocando errores en tiempo de ejecución obligando a intervenir manualmente, teniendo el equipo que volver a ejecutar todo de nuevo. Estas incidencias estaban relacionadas con timeouts, errores de sesión o fallos en la autenticación, difíciles de diagnosticar por el carácter externo del conector y la falta de trazabilidad clara.

Con la nueva implementación de Informatica Intelligent Cloud Services (IICS) el problema desaparece porque IICS incluye un conector nativo para Salesforce que, como está dentro del entorno cloud de IICS, permite conectar introduciendo las credenciales del usuario, el token de seguridad y la URL de la instancia de Salesforce no siendo

necesario definir parámetros extra, instalar drivers o cambiar configuraciones internas.

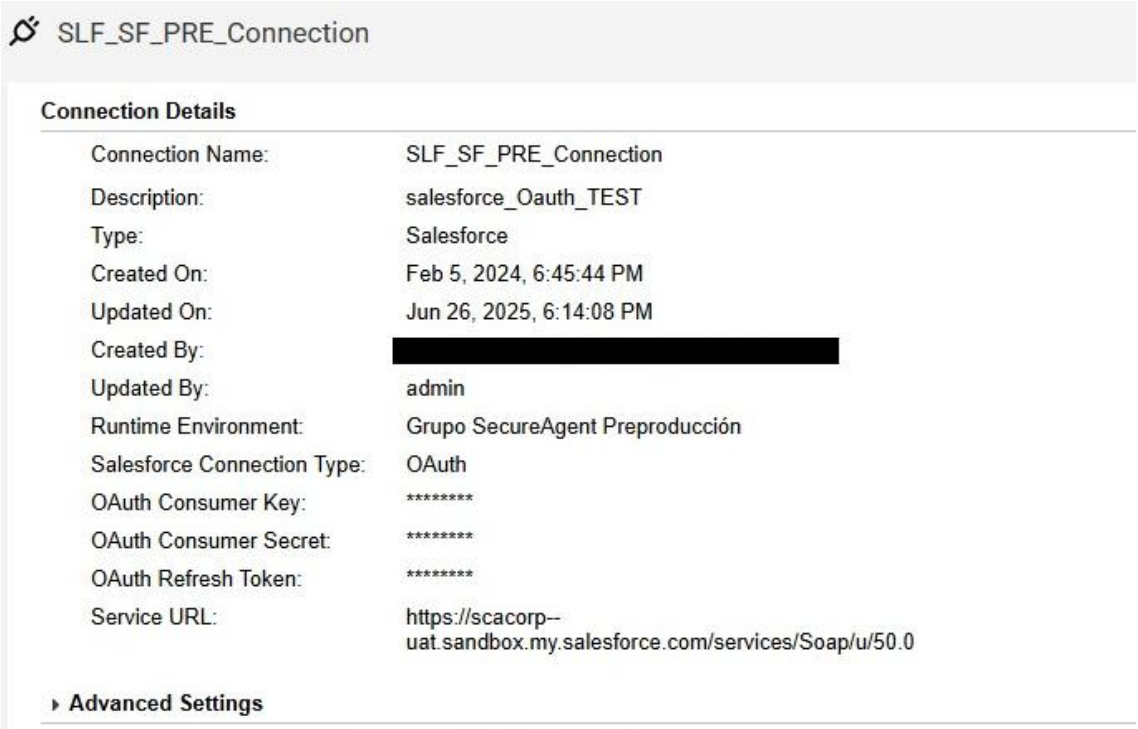


FIGURA 5: Configuración de la conexión IICS-Salesforce.

Este modelo simplifica la configuración inicial y añade más fiabilidad porque desde que se migró a IICS no se ha registrado ningún error de conectividad con Salesforce, incluso cuando se ejecutan procesos de alto volumen. El conector maneja la autenticación, controla los límites de API y se vuelve a conectar cuando es necesario, sin que el usuario o el equipo técnico tengan que intervenir.

También la gestión de conexiones en IICS se hace de forma centralizada y reutilizable ya que cuando se crea una conexión, esta se puede utilizar en diferentes tareas o mappings y las credenciales se guardan cifradas y gestionadas por el entorno lo que facilita la trazabilidad, seguridad y mantenimiento.

El cambio en el modelo de conectividad muestra una ventaja del paso de la arquitectura tradicional a la plataforma cloud porque ahora tiene menos complejidad técnica, menos errores operativos, más estabilidad y más escalabilidad. Lo que antes suponía una fuente de incidencias recurrentes, ahora está completamente resuelto gracias a un enfoque más moderno, estandarizado y gestionado por la propia plataforma.

También, hay que destacar la diferencia en los costes ocultos de cada solución. PowerCenter parece una opción que ya está implementada en muchas empresas pero utilizarlo supone una inversión en mantenimiento, escalado y adaptación de nuevas

necesitades o requerimientos consume tiempo y recursos técnicos sin embargo, IICS ofrece un modelo que se adapta, el usuario paga solo lo que necesita y ajusta el consumo según la carga de trabajo. [4] [5].

En definitiva, la comparación entre ambos entornos no se limita únicamente a lo funcional o visual, sino que se extiende a todo el ecosistema técnico que los rodea. La experiencia adquirida durante este proyecto ha permitido observar con claridad cómo una plataforma como IICS, concebida desde su origen para operar en la nube, ofrece una serie de ventajas estructurales que van más allá de la interfaz o la velocidad de ejecución. Al eliminar la dependencia de configuraciones complejas, como las necesarias para establecer conexiones externas en PowerCenter, y al centralizar la gestión de tareas clave como la autenticación, la escalabilidad o el monitoreo, IICS permite que los esfuerzos del equipo se centren en el desarrollo y optimización del proceso, y no en la resolución de incidencias técnicas. Esta transición no solo ha supuesto una mejora en términos de eficiencia, sino también un paso firme hacia una arquitectura más robusta, adaptable y preparada para evolucionar con las necesidades del negocio.

3. RESULTADOS Y DISCUSIÓN

3.1 Comparativa entre la solución manual y las herramientas ETL.

Una de las partes más reveladoras de este trabajo ha sido comprobar en primera persona cómo se comporta un proceso de integración de datos cuando se desarrolla a mano, utilizando un lenguaje de programación tradicional, frente a cuando se construye con una herramienta específica diseñada para ello. La diferencia no solo está en el resultado final, que en ambos casos puede ser funcional, sino en todo lo que implica llegar hasta ahí: tiempo invertido, facilidad de mantenimiento, control de errores, capacidad de adaptación y escalabilidad.

En la solución manual implementada, se ha utilizado C como lenguaje de desarrollo, esto ha implicado definir estructuras para almacenar los datos, escribir funciones para leer el fichero línea a línea, dividir cada registro por campos, controlar posibles errores de lectura, comparar elementos entre sí para detectar duplicados y, finalmente, generar dos salidas distintas: una para los registros únicos que irían a Salesforce, y otra para guardar todo el contenido, incluidos los duplicados. Todo esto, sin contar con herramientas auxiliares ni librerías externas.

Solo el proceso de identificar si un cliente aparece más de una vez en la misma campaña

ha requerido implementar comparaciones entre elementos, manejar bucles anidados y gestionar la memoria con cuidado para evitar fallos. Además, hubo que crear lógica condicional para decidir qué campos se debían vaciar en el caso de duplicados y cómo asegurarse de que la estructura del fichero de salida fuera correcta y a pesar de toda esa dedicación, surgieron errores: campos desalineados, saltos de línea inesperados, comas fuera de lugar… nada fuera de lo común cuando se trabaja a bajo nivel, pero que evidencia lo delicado que es construir este tipo de flujos sin una base pensada para ello.

Otro punto importante ha sido el control de calidad. En la solución manual, cualquier validación o verificación extra debe añadirse de forma explícita en el código. No hay avisos si un campo viene vacío, ni alertas si un valor se sale de los parámetros esperados ni tampoco un log del proceso a menos que se programe, ni posibilidad de revisar qué registros pasaron por cada paso sin añadir líneas específicas que los impriman o guarden en un archivo temporal. Esto no solo incrementa el esfuerzo técnico, sino que añade fragilidad al sistema: cualquier pequeño cambio puede romper lo anterior si no se gestiona con mucho cuidado.

En contraste, al usar una herramienta como PowerCenter o IICS, toda esta lógica se traslada a un entorno visual donde cada paso se representa como un componente claramente identificado donde las fuentes de datos se conectan mediante asistentes, las transformaciones se aplican con bloques configurables y las condiciones se definen a través de expresiones fáciles de seguir. No hace falta escribir ni una sola línea de código para replicar toda la lógica que antes llevó decenas de líneas en C y si algo falla, el sistema lo notifica, registra el error y permite depurarlo desde una consola.

Un aspecto que se valora mucho cuando se trabaja con estas herramientas es la posibilidad de reutilizar componentes, por ejemplo, una lógica de validación o una expresión de transformación puede guardarse y aplicarse en otro flujo sin necesidad de volver a escribirla. Esto, en la práctica, ahorra muchísimo tiempo, sobre todo cuando el volumen de mappings crece o cuando varios procesos deben cumplir con las mismas reglas de calidad.

Otro beneficio que se ha hecho evidente es la trazabilidad ya que en el entorno manual, si se quiere saber qué ocurrió con un registro concreto, hay que incluir trazas personalizadas o revisar el código línea a línea, sin embargo, en PowerCenter e IICS cada paso genera su propio log y es posible ver por dónde ha pasado cada registro.

Desde el punto de vista de mantenimiento, también hay una diferencia enorme dado que en un proyecto real, los procesos cambian con frecuencia: se añaden campos, se modifican

condiciones y cambian los destinos, ahora bien, en una solución manual todo esto implica abrir el código, entender lo que hizo quien lo desarrolló, y modificar con cuidado para no romper nada. En una herramienta ETL, basta con entrar al mapping, hacer los cambios en los componentes afectados y volver a desplegar, en otras palabras, el impacto es menor, el riesgo más controlado y el tiempo invertido mucho menor.

Tampoco hay que olvidar que las herramientas modernas ya están preparadas para gestionar múltiples fuentes y destinos, realizar transformaciones complejas, aplicar reglas condicionales, y ejecutar procesos en paralelo. Todo eso que en un enfoque manual se convierte en líneas y líneas de código, ya viene integrado y optimizado.

Por todo esto, la conclusión es clara: aunque es técnicamente posible desarrollar procesos ETL desde cero, y hacerlo puede ser útil como ejercicio formativo, no es un enfoque recomendable en entornos productivos. Las herramientas específicas no solo ahorran tiempo, sino que ofrecen robustez, escalabilidad, trazabilidad y facilidad de mantenimiento que son muy difíciles de conseguir con una solución desarrollada a medida.

3.2 Diferencias observadas entre PowerCenter e IICS.

Aunque PowerCenter e IICS son dos soluciones desarrolladas por la misma compañía, y comparten una base conceptual similar, lo cierto es que en la práctica representan dos formas distintas de entender y ejecutar los procesos de integración de datos. No se trata solo de una evolución técnica, sino también de un cambio en la forma de trabajar y en cómo se diseñan y gestionan estos procesos en el día a día.

Una de las diferencias más importantes tiene que ver con la arquitectura de ejecución dado que PowerCenter está diseñado para funcionar en un entorno local, es decir, en servidores que pertenecen a la propia empresa o están bajo su control directo. Esto implica que hay que dimensionar la infraestructura, instalar los distintos componentes del sistema (como el Integration Service, Repository Service, etc.), y mantenerlos operativos y actualizados. Este modelo tiene ventajas en términos de control, pero también requiere dedicar muchos más recursos técnicos a tareas que, en el fondo, no forman parte del objetivo principal: integrar datos.

En cambio, IICS se apoya en una arquitectura completamente gestionada desde la nube, lo que significa que todo el mantenimiento de infraestructura, los parches, la seguridad y el escalado están gestionados por el proveedor. El usuario se concentra únicamente en el diseño de los procesos, mientras que la plataforma se encarga de que todo funcione detrás.

Este modelo no solo reduce la carga técnica, sino que también hace posible trabajar desde cualquier ubicación, sin depender de una red corporativa ni de instalaciones específicas.

Otra diferencia fundamental aparece en la forma en que se gestiona el ciclo de vida de un proyecto, en PowerCenter, todo está muy orientado a entornos estables, con procesos definidos, donde los cambios son controlados y se aplican tras validaciones formales de ahí que la herramienta está muy bien preparada para este tipo de entornos, donde se valora la robustez por encima de la agilidad. En IICS, sin embargo, el planteamiento es más flexible ya que los entornos se crean rápidamente, las conexiones se gestionan con asistentes, y los cambios se pueden aplicar de forma casi inmediata. Esto hace que IICS encaje mucho mejor en proyectos ágiles o con ciclos de desarrollo más cortos.

En el diseño de procesos, los dos entornos usan la misma lógica de trabajar con mappings, con transformaciones y con flujos visuales. IICS ha simplificado esos conceptos al hacerlos más fáciles de usar y más rápidos de configurar donde una transformación condicional, una conexión con Salesforce o el despliegue de un proceso no necesitan conocimientos especiales ni configuraciones complicadas. En PowerCenter, esas mismas acciones existen, pero requieren más pasos y más conocimientos técnicos para ejecutarlas bien.

Hay que pensar también en la automatización y la escalabilidad, en PowerCenter, escalar un proceso implica preparar nuevos servidores, ajustar la configuración y asegurarse de que todo esté coordinado. El proceso es posible, pero necesita tiempo y experiencia, sin embargo, en IICS la escalabilidad ocurre de forma automática según la carga. Cuando un proceso necesita más recursos, la plataforma gestiona los recursos sin que el usuario tenga que intervenir, este detalle marca una diferencia en los contextos donde el volumen de datos cambia con frecuencia como por ejemplo, en las campañas comerciales o en los entornos multiusuario.

Desde el punto de vista del costo operativo hay matices, PowerCenter puede costar menos cuando ya está instalado y amortizado, pero su mantenimiento trae costos ocultos: el personal técnico, los recursos de infraestructura, los tiempos de parada, etc. IICS cobra por el uso de ahí que permite que sea más flexible para los proyectos que cambian de tamaño o que necesitan adaptarse a las nuevas realidades sin invertir en el hardware adicional.

Una diferencia más sutil pero importante está en la forma en que se adoptan mejoras y nuevas funcionalidades, en PowerCenter, los usuarios tienen que planear, instalar y validar cualquier actualización del sistema. En IICS, la plataforma integra esas mejoras

automáticamente haciendo que los usuarios puedan acceder antes a las funcionalidades nuevas y no necesiten interrumpir el servicio ni adaptar su infraestructura.

IICS ofrece una interfaz que sigue la modernidad e intuición alineándose con los estándares de la actualidad ya que facilita el aprendizaje de los usuarios recientemente incorporados pero también aumenta la productividad de los usuarios que usan la herramienta todos los días. PowerCenter sigue ofreciendo capacidad pero requiere un aprendizaje que lleva más tiempo, la interfaz de PowerCenter muestra signos de envejecimiento frente a las exigencias de la actualidad.

Además de las diferencias previamente detalladas en términos de arquitectura, mantenimiento, escalabilidad y usabilidad, es importante mostrar cómo estas diferencias se manifiestan en el momento de la ejecución del proceso. A través de capturas obtenidas durante el desarrollo del proyecto, es posible observar cómo se comporta la misma lógica de negocio implementada en PowerCenter y en IICS, y qué impacto práctico tiene en el entorno productivo.

En primer lugar, la siguiente captura muestra la ejecución del flujo m_SCA_campanas_outbound_miembro_campana desde la herramienta Workflow Monitor de PowerCenter. Este entorno, aunque consolidado y muy potente, requiere cierto conocimiento técnico para su interpretación y manejo. En el caso concreto representado, la tarea se ha ejecutado en el repositorio Rep_Produccion_SALESFORCE_IS, con una duración total de 1 minuto y 14 segundos, durante los cuales se han procesado un total de 2119 registros, sin errores ni rechazos.

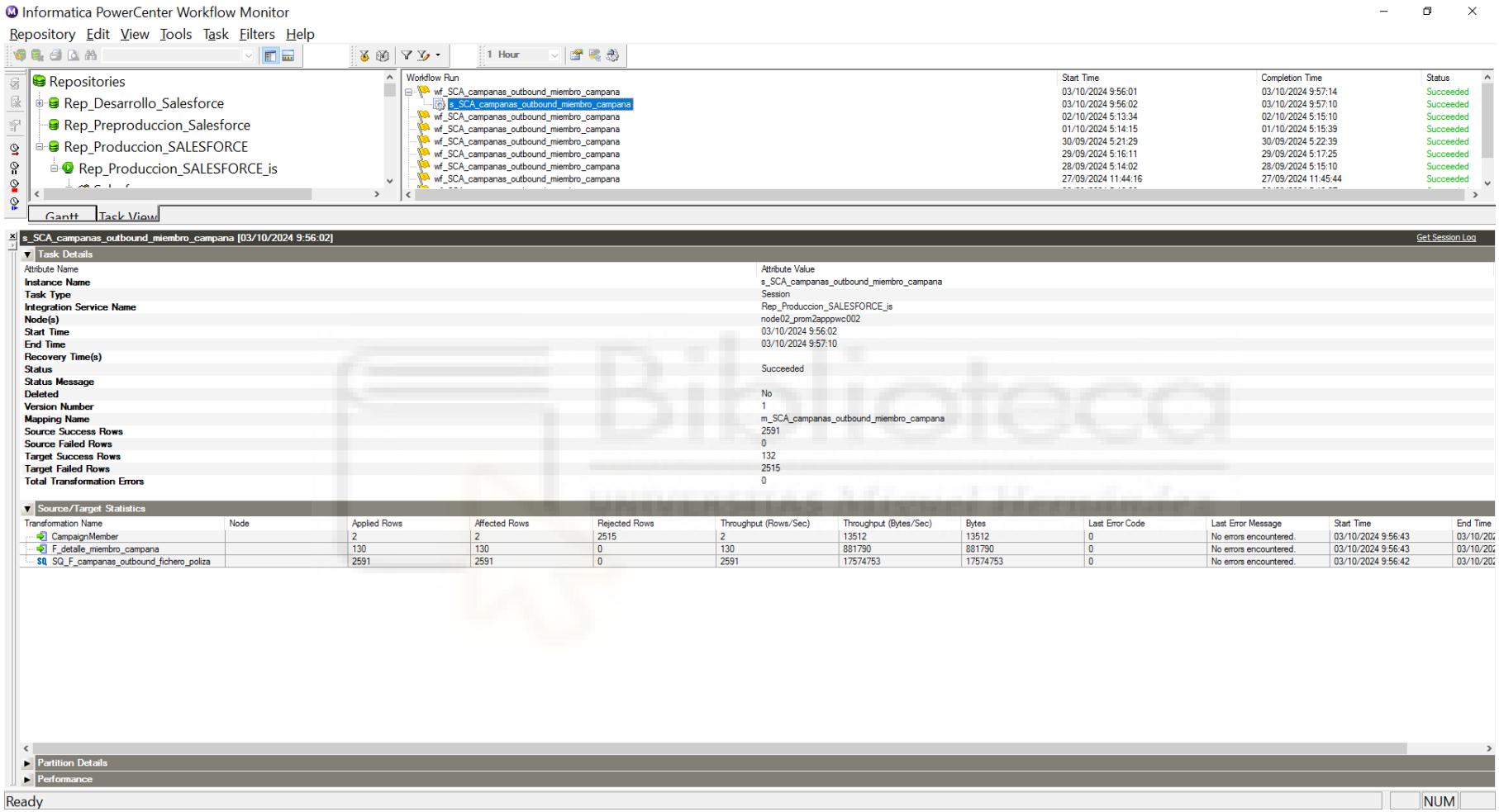


FIGURA 6: Ejecución del proceso en PowerCenter.

A continuación se muestra la misma ejecución, esta vez desde la plataforma Informatica Intelligent Cloud Services donde el mismo flujo se procesó en 26 segundos, la velocidad muestra una mejora en rendimiento y eficiencia. PowerCenter procesó 2119 registros sin errores ni registros rechazados e Informatica Intelligent Cloud Services procesó 2119 registros sin errores ni registros rechazados, en consecuencia, la comparación de resultados muestra que se migró correctamente el proceso a IICS.

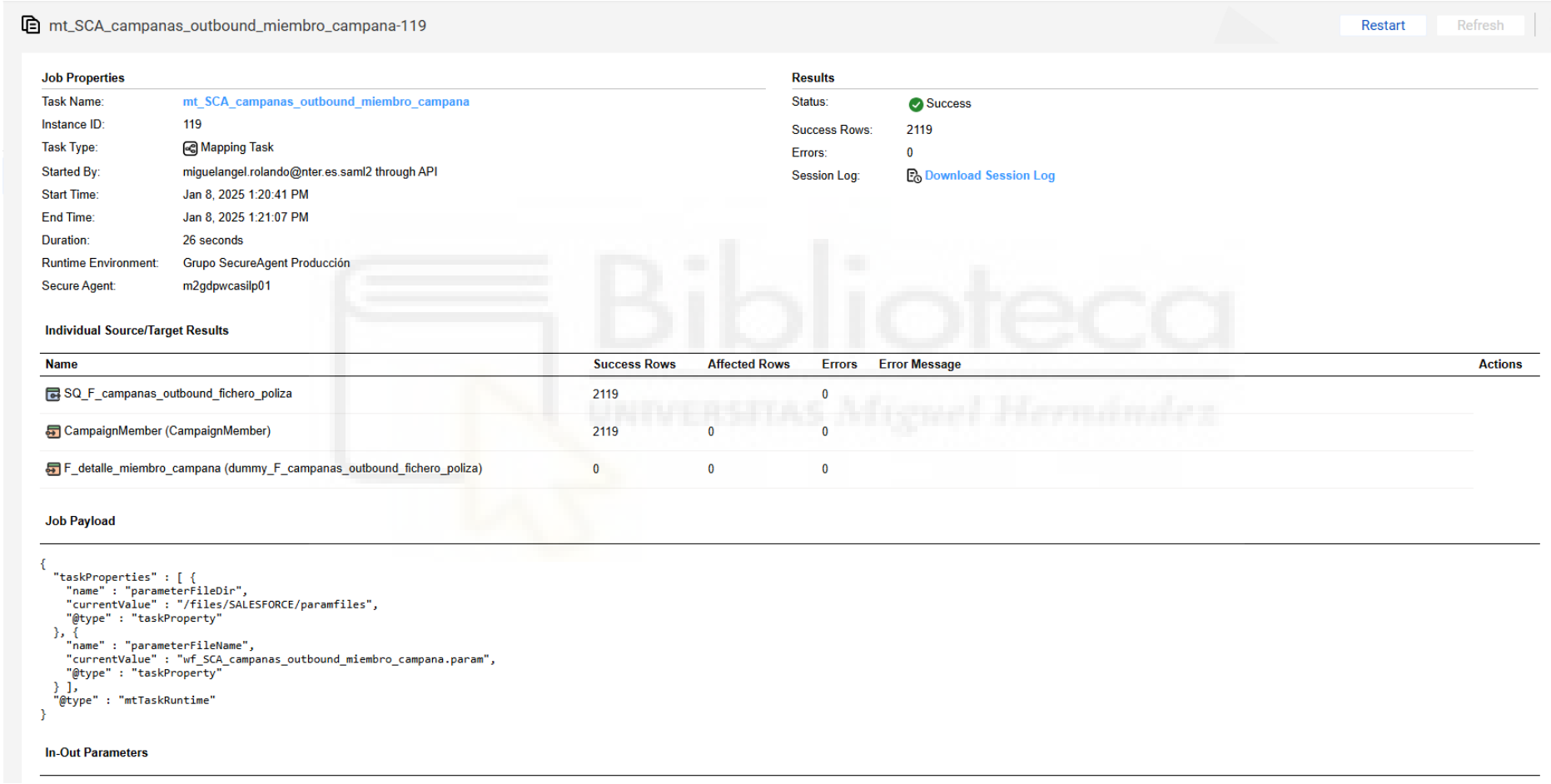
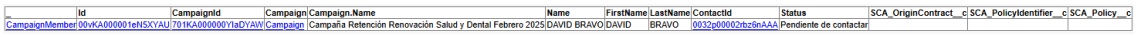


FIGURA 7: Ejecución del proceso en IICS

Aquí, tienes todos los detalles sobre el proceso en un formato mucho más amigable y comprensible, desde el nombre de la tarea y el entorno donde se ejecutó (Grupo SecureAgent Producción), hasta el hecho de que puedes descargar el log completo desde la interfaz con un solo clic. Esta forma de ejecutar el proceso permite un control mucho más dinámico, especialmente en contextos en los que se requiere supervisión continua o en tiempo real.

Para comprobar el efecto final de ambos procesos, también se incluyen capturas tomadas directamente de Salesforce. En la primera, desde el objeto CampaignMember, se puede ver cómo un cliente con varias pólizas no muestra información de contrato, póliza ni mensaje, y eso no es casualidad, sino una consecuencia de la lógica del proceso que evita la visualización de información duplicada cuando hay más de una póliza vinculada al mismo cliente dentro de la misma campaña.



	Id	CampaignId	Campaign	Campaign.Name	Name	FirstName	LastName	ContactId	Status	SCA_OriginContract__c	SCA_PolicyIdentifier__c	SCA_Policy__c
CampaignMember	00vKA000001eN5XYAU	701KA000000Y1aDYAW	Campaign	Campaña Retención Renovación Salud y Dental Febrero 2025	DAVID BRAVO	DAVID	BRAVO	0032p00002rbz6nAAA	Pendiente de contactar			

FIGURA 8: Ejecución del proceso en IICS

En este caso, el objeto que recopila los datos detallados por póliza es CampaignMemberDetail, el cual como se puede observar en la imagen inferior, contiene los diferentes productos contratados, así como los códigos correspondientes y mensajes personalizados para cada uno de ellos.

	Id	CreatedDate	SCA_CampaignMemberId__c	SCA_Contact__c	SCA_Campaign__r	SCA_Campaign__r.Name	SCA_Message__c	SCA_OriginContract__c	SCA_PolicyIdentifier__c	SCA_Policy__c	SCA_Status__c
SCA_CampaignMemberDetail	a1pKA000000pqbiWYAQ	2024-11-29T01:30:58.000+0000	00vKA000001eN5XYAU	0032p00002rbz6nAAA	Campaign	Campaña Retención Renovación Salud y Dental Febrero 2025	ORDEN OFERTA: 30% DTO PRÓXIMOS 3 MESES - 9% DTO	9950485739280	282428091	a1pKA000000pqbiWYAQ	Pendiente de contacta
SCA_CampaignMemberDetail	a1pKA000000pqbiYYAQ	2024-11-29T01:30:58.000+0000	00vKA000001eN5YYAU	0032p00002rbz6nAAA	Campaign	Campaña Retención Renovación Salud y Dental Febrero 2025	ORDEN OFERTA: ORDEN OFERTA: 50% DTO PRÓXIMOS 3 MESES - 3 MES GRATIS - 15% DTO	9960485743965	282428560	a1pKA000000pQbBAAU	Pendiente de contacta

FIGURA 9: Ejecución del proceso en IICS

Las evidencias recopiladas durante el desarrollo del proyecto no solo demuestran que la migración de PowerCenter a IICS ha sido un éxito, sino que además muestran una mejora importante en factores clave como el rendimiento, la monitorización y la fiabilidad del sistema. En contraposición a un entorno tradicional donde eran necesarios ciertos ajustes delicados y un cierto grado de vulnerabilidad en las conexiones, como ocurría con PowerCenter, la solución basada en IICS ha demostrado ser más ágil, más robusta y mejor adaptada a los estándares actuales pero esto no es solo una evolución de la tecnología, sino un cambio profundo en el paradigma de la integración de datos: de un modelo basado en el control y la estabilidad, a otro basado en la flexibilidad, escalabilidad y eficiencia operativa. En función del tipo de proyecto, el nivel de exigencia técnica y los recursos

disponibles, cada organización debería evaluar cuál es la opción más adecuada para sus necesidades actuales y futuras.

3.3 Comparativa económica (TCO) entre PowerCenter e IICS

Realizar una comparativa económica del Coste Total de Propiedad (TCO) entre Informatica PowerCenter e Informatica Intelligent Cloud Services (IICS) implica analizar la diferencia fundamental entre dos modelos de despliegue: el modelo on-premise, basado en inversión de capital (CapEx), y el modelo cloud o iPaaS, fundamentado en gasto operativo (OpEx) [36].

En términos generales, IICS presenta un TCO inferior al de PowerCenter, principalmente debido a la eliminación de los costes de infraestructura física y a la reducción de los costes de mantenimiento y administración, que quedan incluidos en la suscripción al servicio [37].

El cambio esencial reside en que PowerCenter requiere una inversión inicial elevada (CapEx) en licencias y hardware, mientras que IICS sustituye este modelo por un gasto operativo variable (OpEx) basado en consumo.

Categoría	PowerCenter (On-Premise)	IICS (Cloud / iPaaS)
1. Coste de Licencias	Modelo CapEx (Inversión inicial). • Licencia Perpetua: Se realiza una compra única y elevada al inicio. • Coste Basado en Capacidad: El precio se basa en los núcleos de CPU (cores) de los servidores donde se ejecuta, las ediciones (Standard, Advanced) y los conectores (add-ons) adquiridos.	Modelo OpEx (Gasto operativo). • Suscripción: Pago recurrente (mensual o anual). • Coste Basado en Consumo: El precio se basa en el uso, medido en IPUs (Informatica Processing Units), volumen de datos procesados, número de conectores y los servicios específicos contratados (Data Integration, API Management, etc.).
2. Infraestructura	Coste Alto (CapEx + OpEx). • El cliente debe comprar, configurar y mantener toda la infraestructura física o virtual: • Servidores (para dominio, repositorio y nodos). • Bases de datos (para el repositorio). • Almacenamiento y redes.• Incluye costes de electricidad, refrigeración y espacio en el CPD.	Coste Cero o Mínimo. • Plataforma Gestionada: El dominio, repositorio y la gestión de la plataforma están alojados y mantenidos por Informatica en la nube (modelo SaaS). • Secure Agent: El único coste de infraestructura es el servidor (físico o virtual) para ejecutar el Secure Agent si se necesita acceder a datos on-premise, pero su huella es mínima en comparación.
3. Mantenimiento	Coste Alto y Obligatorio (OpEx). • Soporte Anual: Se paga una cuota anual (típicamente 18-22% del coste de la licencia) para tener derecho a soporte y nuevas versiones. • Actualizaciones Manuales: Las actualizaciones (parches, hotfixes, migraciones de versión) son proyectos complejos gestionados y ejecutados por el cliente, requiriendo planificación y ventanas de parada.	Incluido en la Suscripción (OpEx). • Soporte Integrado: El coste de soporte y mantenimiento está incluido en la cuota de suscripción. • Actualizaciones Automáticas: Informatica gestiona todas las actualizaciones de la plataforma de forma transparente y automática (modelo zero-downtime), liberando al cliente de esta carga.
4. Operación	Coste de Personal Alto (OpEx). • Requiere personal altamente especializado para la administración de la plataforma: • Administradores de PowerCenter. • DBAs (para gestionar la BBDD del repositorio). • Administradores de Sistemas (SO, hardware, redes).	Coste de Personal Reducido (OpEx). • Administración Simplificada: No se requiere gestión de BBDD de repositorio ni de servidores de plataforma. • El rol de administrador se centra en la configuración de servicios en la nube, usuarios y la gestión del Secure Agent, reduciendo drásticamente las horas de administración de infraestructura.

TABLA 1: Comparativa detallada del TCO

El TCO de PowerCenter está dominado por la inversión inicial (CapEx) y los costes fijos de operación (OpEx) asociados al mantenimiento de la infraestructura y al personal especializado. Aunque ofrece costes predecibles tras la inversión inicial, resulta muy costoso de escalar, ya que requiere adquirir nuevas licencias de CPU y ampliar el hardware [38]. Las licencias se conceden por usuario o por servidor, con una inversión inicial elevada y costes fijos anuales, fuentes del sector estiman un coste medio de unos 2.000 dólares mensuales para un usuario estándar, lo que equivale a más de 20.000 dólares anuales por entorno [39]. Otras plataformas de comparación, como Vendr, sitúan el gasto medio anual de las soluciones de Informatica en torno a 56.000 dólares, cifra que puede variar en función de los módulos contratados requiriendo además, la renovación periódica de licencias y la gestión manual de actualizaciones y soporte técnico, lo que incrementa los costes indirectos de mantenimiento [40].

El modelo IICS se basa en un OpEx variable y elástico, que elimina la inversión inicial y reduce los costes indirectos ya que los ahorros principales proceden de la eliminación del hardware on-premise, la reducción del personal de administración y la simplificación del mantenimiento [41]. No obstante, el gasto es variable y puede fluctuar según el volumen de uso, lo que exige una adecuada gobernanza del consumo (FinOps). En el AWS Marketplace se publica un ejemplo de 120 IPU mensuales por 131.760 dólares al año, que sirve como referencia del rango de precios para entornos corporativos [42].

Diversos estudios confirman una reducción del TCO del 20 % al 50 % tras migrar de PowerCenter a IICS. Adastra reportó un ahorro del 20 % anual al eliminar los costes de mantenimiento del data center [39]. En el caso de BMC Software, la reducción de costes operativos alcanzó un 40 % (≈ 265.000 USD anuales), gracias a la supresión del mantenimiento de hardware y las actualizaciones on-premise, asimismo, TelevisaUnivision redujo sus costes de integración en un 44 %, y un proveedor de seguros de salud reportó un ahorro anual de 4 millones USD tras migrar sus soluciones a la nube de Informatica [36].

En conjunto, la migración de PowerCenter a IICS supone la transformación de un modelo CapEx rígido a un modelo OpEx flexible y escalable, con un TCO global significativamente menor. Los beneficios más relevantes se concentran en las áreas de infraestructura, mantenimiento y operación, donde los costes se reducen o desaparecen por completo aunque el gasto en la nube puede ser variable, la reducción de la deuda técnica y de los costes hundidos de la infraestructura on-premise convierte a IICS en una alternativa más eficiente y sostenible a largo plazo [37].

3.4 Ventajas, retos y eficiencia de cada enfoque

Una vez que se han implementado y verificado mediante programación manual y a través de ETL, se puede parar a evaluar, con cierta perspectiva, las ventajas y desventajas de cada una de ellas. No se trata solo de comparar “qué hace cada cosa”, sino que debemos evaluar el esfuerzo que requiere cada una, su capacidad de adaptación, facilidad de modificarse por requerimientos nuevos, y comportamiento ante la realidad.

Comencemos por la solución manual, cuya mayor ventaja es la contención absoluta de cada paso ya que, al programar se sabe perfectamente qué sucede en cada línea, se puede saber cómo se procesa la información, qué lógica se aplica, y cómo se construye cada resultado. Esa transparencia total puede ser conveniente en entornos donde se desee trazabilidad absoluta, o cuando se necesite construir algo muy particular, que no tenga

implementación estándar en una herramienta ETL.

Además, el enfoque manual permite una optimización muy detallada, por ejemplo, se puede afinar el uso de memoria, reducir ciclos innecesarios o ajustar condiciones lógicas para maximizar el rendimiento en función del lenguaje utilizado y del sistema donde se ejecuta. Esto puede ser una ventaja en procesos críticos, muy sensibles al rendimiento o en sistemas con recursos limitados.

Ahora bien, junto a estas ventajas, aparecen una serie de retos importantes que deben tenerse en cuenta. El primero y más evidente es la complejidad del desarrollo, implementar un proceso de integración manualmente, aunque posible, requiere tiempo, experiencia técnica y una atención constante a los detalles. Basta con olvidar una condición o cometer un pequeño error en el manejo de cadenas o estructuras para que el resultado final no sea el esperado. En el caso concreto de este trabajo, los errores más habituales fueron saltos de línea mal gestionados, comas fuera de lugar o registros que no se procesaban al final del fichero.

Otro reto importante es el mantenimiento, una vez que el proceso está funcionando, cualquier cambio, por mínimo que sea, implica volver al código, entender lo que se hizo, modificarlo y volver a probarlo todo. Esto hace que la solución sea frágil, especialmente si el código pasa de mano en mano o si se utiliza tiempo después de haberlo desarrollado.

A nivel de eficiencia operativa, también hay una gran diferencia, en la solución manual, todo se ejecuta de forma secuencial, y cualquier intento de paralelizar el proceso requiere programación explícita, uso de hilos o estrategias que añaden complejidad. En cambio, las herramientas ETL ya están diseñadas para ejecutar varios procesos a la vez, o para dividir los datos en bloques y tratarlos en paralelo, lo que mejora el rendimiento sin que el usuario tenga que preocuparse por los detalles internos.

Cuando se observa el enfoque basado en herramientas como PowerCenter o IICS, las ventajas más evidentes están en la productividad, la facilidad para construir flujos y la rapidez con la que se pueden ajustar o rediseñar procesos. En lugar de codificar manualmente cada paso, el usuario puede trabajar con componentes visuales que encapsulan funcionalidades comunes: filtrado, ordenación, agrupaciones, condiciones, cálculos... Todo esto reduce drásticamente el tiempo de desarrollo y también el margen de error.

Otra ventaja fundamental es que estas herramientas incorporan funciones de monitorización y control que permiten saber en todo momento si un proceso ha funcionado correctamente, cuántos registros ha procesado, dónde ha fallado o en qué

punto exacto se encuentra. Esta visibilidad resulta clave en entornos productivos, donde no basta con saber que el fichero se ha generado, sino que es necesario tener trazabilidad completa de lo que ha pasado en cada etapa.

Además, muchas herramientas ETL incluyen mecanismos de versionado, reutilización de objetos y exportación de mappings, lo que permite trabajar de forma colaborativa y reutilizar lógica ya definida en otros proyectos, sin tener que empezar de cero cada vez. Este tipo de funcionalidades no existe de forma nativa en los desarrollos manuales, a menos que se cree un sistema propio de control y gestión, lo cual complica aún más el proyecto.

No obstante, también existen retos en las soluciones ETL, aunque de un tipo diferente. Uno de ellos es el aprendizaje inicial ya que, aunque las herramientas están diseñadas para ser intuitivas, requieren una inversión inicial para entender cómo se estructura cada tipo de proceso, cómo funcionan los distintos componentes y qué implicaciones tiene cada decisión de diseño. Especialmente en herramientas con mucha profundidad funcional como PowerCenter, puede llevar un tiempo considerable aprender a usarlas con soltura.

Otro reto está relacionado con la rigidez de los componentes predefinidos, si bien, las herramientas ETL cubren la mayoría de casos habituales, cuando se necesita una lógica muy específica, puede que no exista un componente que lo resuelva directamente. En esos casos, hay que recurrir a expresiones complejas, código embebido o integraciones externas, lo cual puede reducir parte de la ventaja que aporta la abstracción visual.

Por último, no hay que olvidar el aspecto económico, las soluciones ETL profesionales, especialmente en su versión on-premise, requieren licencias, mantenimiento y en ocasiones soporte especializado. En cambio, una solución manual puede ser más económica en términos de coste directo, aunque más cara a medio plazo por el tiempo que implica su desarrollo y mantenimiento.

Desde una perspectiva de eficiencia global, las herramientas ETL ganan por claridad, escalabilidad y robustez porque son especialmente útiles cuando se trabaja con volúmenes grandes de datos, múltiples fuentes, y equipos que necesitan colaborar en el desarrollo y mantenimiento de los procesos. En cambio, las soluciones manuales pueden ser válidas para tareas muy específicas, procesos simples o entornos donde no se justifica todavía una inversión en herramientas profesionales.

Lo que queda claro es que no existe un único enfoque válido para todos los casos. Cada situación requiere un análisis particular, teniendo en cuenta el entorno, los recursos

disponibles, la frecuencia con la que cambian los procesos, y la criticidad de los datos que se están tratando. Lo importante no es elegir la solución "más potente", sino la más adecuada al contexto donde se va a utilizar.

3.5 Impacto de la migración en la operativa real

Cuando una organización decide migrar sus procesos desde una herramienta tradicional como PowerCenter hacia una plataforma en la nube como IICS, el cambio no se limita a una cuestión técnica. Las implicaciones se extienden a la forma de trabajar, a la cultura del equipo, a la planificación de los despliegues y al modo en que se entienden los propios procesos de integración, en este caso concreto, la migración ha afectado tanto a la infraestructura tecnológica como al funcionamiento operativo diario del equipo que gestiona los datos.

Uno de los primeros cambios que se percibe tras una migración de este tipo es la eliminación de las tareas de mantenimiento de infraestructura debido a que, con PowerCenter instalado en servidores internos, había que dedicar tiempo y recursos a asegurarse de que los servicios estuvieran operativos, actualizados y bien configurados. Cualquier cambio en la red, modificación de seguridad o actualización del sistema operativo podía afectar al correcto funcionamiento del entorno y esto exigía una comunicación constante entre los equipos de datos y los departamentos de sistemas.

Con el paso a IICS, este tipo de dependencia desaparece, la plataforma está gestionada por el propio proveedor y, al estar basada en la nube, no requiere instalación local ni mantenimiento físico. Esto ha tenido un efecto directo en la operativa diaria: el equipo puede centrarse exclusivamente en el diseño, desarrollo y validación de procesos, sin tener que preocuparse por aspectos como caídas de servidor, gestión de espacio o backups traduciéndose así en más tiempo para mejorar procesos o implementar nuevas funcionalidades.

Otro impacto importante ha sido la agilidad en la puesta en marcha de nuevos desarrollos, con PowerCenter, desplegar un nuevo desarrollo, validarlo en un entorno de pruebas y llevarlo a producción implicaba una serie de pasos técnicos bien definidos, que podían alargarse en función de los recursos del entorno. En IICS, todo el proceso se simplifica, al trabajar directamente en la nube, el desarrollo puede comenzar en cuanto se define la lógica de negocio. Las conexiones a orígenes y destinos se configuran en minutos, y el despliegue puede hacerse desde cualquier ubicación, incluso de forma remota.

Esta agilidad ha cambiado también los tiempos de respuesta ante incidencias o peticiones

del negocio, si antes era necesario planificar con antelación los desarrollos o los cambios en los procesos, ahora es posible hacer ajustes sobre la marcha, probar en caliente y desplegar sin interrumpir otros procesos en ejecución. Esto ha mejorado notablemente la capacidad del equipo para adaptarse a cambios inesperados o necesidades urgentes, como por ejemplo la modificación de una campaña comercial en mitad de su ejecución.

También se ha notado una mejora en la visibilidad y trazabilidad de los procesos, IICS permite monitorizar los flujos de datos en tiempo real, ver métricas de ejecución, tiempos de procesamiento y errores, todo desde una única consola centralizada. Esta capacidad para observar lo que ocurre sin tener que acceder a múltiples logs o servicios ha facilitado mucho la labor de seguimiento y análisis ya que antes, detectar una caída o un fallo puntual requería revisar varios puntos del sistema; ahora, basta con consultar el panel de control.

En cuanto a la formación, el cambio ha implicado una curva de aprendizaje, principalmente para algunos perfiles acostumbrados a trabajar con PowerCenter pero como ambas herramientas son bastante similares, el cambio no ha sido especialmente brusco. Lo que sí ha cambiado es la facilidad de uso para introducir nuevos usuarios en la herramienta, ya que IICS es mucho más intuitivo y con mejor documentación en Internet. Esto nos ha permitido agilizar la incorporación de nuevos miembros al equipo, o incluso colaborar con perfiles menos técnicos, que pueden revisar mappings o procesos sin necesidad de tener un gran dominio de la herramienta.

Desde el punto de vista del negocio, la migración ha tenido un impacto positivo en términos de tiempo y flexibilidad. Poder desplegar procesos más rápidamente, adaptarse a nuevas fuentes de datos o integrar sistemas externos sin mayores complicaciones ha permitido al equipo de datos ser más ágil en la respuesta a las necesidades de la organización. En un entorno en el que las campañas cambian muy a menudo y los datos se actualizan constantemente, esta adaptabilidad marca una gran diferencia.

Naturalmente, el cambio también ha presentado desafíos, especialmente al principio puesto que reconfigurar procesos, redefinir conexiones o la necesidad de validar que el resultado de los procesos migrados es exactamente el mismo que el anterior ha requerido tiempo, pruebas y planificación pero una vez superado ese período inicial, el día a día del negocio se ha vuelto más fácil, ágil y enfocado en extraer valor de los datos.

En resumen, la migración de PowerCenter a IICS no ha sido solo un cambio de herramienta, sino un cambio de proceso, una forma de colaborar y entender el ciclo de vida del proceso de integración. Se ha ganado autonomía, velocidad, facilidad de

mantenimiento y adaptación a las nuevas formas de operar que exige el entorno digital actual.

3.6 Discusión crítica sobre la evolución de los procesos de integración

Analizar cómo han evolucionado los procesos de integración de datos no es únicamente un ejercicio técnico, sino también una forma de entender cómo han cambiado las prioridades, la cultura tecnológica y los entornos organizativos en los últimos años. Lo que antes se resolvía con soluciones a medida, mantenidas de forma casi artesanal por perfiles muy especializados, ha dado paso a una nueva forma de trabajar con los datos, donde la rapidez, la flexibilidad y la capacidad de adaptación tienen tanto peso como la robustez o la estabilidad.

En sus inicios, los procesos de integración eran muy rudimentarios, se construían con scripts o programas que simplemente movían datos desde un origen hacia un destino, con una mínima transformación por el camino. Estos desarrollos estaban estrechamente ligados a las personas que los creaban, y su mantenimiento dependía en gran medida de que esa persona siguiera disponible o hubiera documentado bien el código. En la práctica, esto generaba una gran dependencia tecnológica, poca escalabilidad y un riesgo elevado de error ante cualquier cambio.

La introducción de herramientas ETL como PowerCenter supuso un avance notable, por primera vez se ofrecía una solución pensada exclusivamente para orquestar este tipo de procesos, con una interfaz visual, componentes especializados y la posibilidad de diseñar flujos de forma modular. Esto no solo facilitó el desarrollo, sino que permitió centralizar la lógica de negocio, aplicar validaciones estandarizadas y garantizar cierto nivel de trazabilidad que era difícil de alcanzar. De esta forma se trabajó durante muchos años en las grandes corporaciones, pero conforme el entorno digital fue mutando comenzaron a surgir limitaciones. Los procesos ETL tradicionales, estables y detallados, no eran tan ágiles a la hora de implantar cambios o atender necesidades imprevistas y la dependencia de infraestructura on-premise, con servidores locales, configuraciones manuales y ciclos de actualización amplios, empezó a chocar con las nuevas formas tecnológicas.

Aquí es donde empiezan a tener cabida las soluciones en la nube, como IICS, no se trata de subir una herramienta a un servidor online, sino de replantear por completo cómo se construyen, gestionan y mantienen los procesos de integración. La nube proporciona un entorno en el que el escalado puede ser automático, el acceso no tiene fronteras y las actualizaciones no requieren intervenciones técnicas complejas. Además, por lo general,

estas plataformas están en constante evolución, lo que permite introducir mejoras sin necesidad de instalaciones manuales y paradas del servicio.

Todo este proceso influyó también en el perfil del equipo de datos en sí ya que hasta ahora se demandaba un perfil técnico, con conocimiento de la herramienta y su entorno de ejecución, ahora se trabaja hacia un enfoque más colaborativo. Los desarrolladores de procesos pueden centrarse en la lógica de negocio y la plataforma se encarga del resto e incluso perfiles menos técnicos pueden participar en ciertas fases del flujo de datos, revisar resultados o proponer mejoras sin tener que conocer todos los entresijos del sistema.

Al mismo tiempo, han aparecido nuevas prioridades, ya no basta con integrar datos una vez al día, en muchos casos se exige hacerlo en tiempo real, con procesos en streaming o con respuestas inmediatas a cambios en los sistemas de origen. Esto ha llevado a que se empiece a hablar más de ELT que de ETL, trasladando parte del esfuerzo de transformación hacia el destino, que suele ser un almacén de datos moderno con capacidades de procesamiento paralelo. Este cambio de paradigma tiene sentido cuando se dispone de plataformas como Snowflake o BigQuery, que permiten hacer transformaciones complejas directamente sobre los datos una vez cargados.

Otro elemento que ha cobrado fuerza en los últimos tiempos es la automatización y la inteligencia artificial dentro de los propios procesos de integración. Algunas plataformas modernas empiezan a incorporar motores que sugieren transformaciones, detectan patrones de datos erróneos o incluso ajustan automáticamente ciertas condiciones para mejorar el rendimiento. Aunque todavía es pronto para considerar que estos sistemas funcionen sin supervisión humana, ya están ayudando a reducir tareas repetitivas y a enfocar el trabajo en los puntos que realmente requieren criterio profesional.

Sin embargo, este camino hacia la modernización no está exento de riesgos, depender de soluciones en la nube implica ceder parte del control técnico, y obliga a confiar en que el proveedor gestiona correctamente la seguridad, la disponibilidad y la privacidad de los datos. Además, la abundancia de herramientas y enfoques puede llevar a una cierta dispersión si no se define bien la arquitectura y los criterios de diseño ya que no todo lo nuevo es automáticamente mejor, y la elección de la herramienta adecuada sigue dependiendo del contexto, los objetivos y los recursos disponibles.

También hay que tener en cuenta que no todas las organizaciones están en el mismo punto de madurez digital de ahí que algunos ya operan en entornos de nube y otros todavía están en sistemas clásicos no migrables a corto plazo. En este último supuesto, la convivencia

de lo viejo con lo nuevo es inevitable y se aboca a la creación de flujos híbridos que mezclan procesos on-premise con otros en la nube que puede ser temporal o puede dilatarse en el tiempo según el sector y su velocidad de transformación interna, por lo tanto, no podemos hablar propiamente de sustitución directa de herramientas tradicionales, sino de una evolución progresiva. PowerCenter todavía tiene sentido en un proyecto, sobre todo cuando la estabilidad, la trazabilidad y la jerarquía de procesos son fundamentales, IICS surge como una respuesta más ágil, más flexible, capaz de asumir el ritmo de cambio actual y las nuevas formas de consumo y gestión de la información. En definitiva, la evolución de los procesos de integración no es solo una cuestión tecnológica, sino una evolución profunda en la gestión de datos. Hemos pasado de una lógica basada en el control técnico absoluto a una visión colaborativa, más automatizada y orientada al valor de los datos para el negocio. Esa transformación, todavía en curso, seguirá redefiniendo los roles, las herramientas y estrategias con las que las organizaciones gestionan los datos para convertirlos en decisiones.

Por eso, más que hablar de un reemplazo directo de las herramientas tradicionales, lo que se está observando es una evolución progresiva. Las plataformas como PowerCenter siguen teniendo su espacio, especialmente en proyectos donde la estabilidad, la trazabilidad y la estructura jerárquica de los procesos son claves, IICS por su parte, se posiciona como una respuesta más ágil y flexible, capaz de adaptarse al ritmo actual de cambio y a las nuevas formas de consumir y gestionar información.

A lo largo de este capítulo se ha llevado a cabo una profunda radiografía de los resultados extraídos al abordar la integración de datos desde diferentes frentes. Desde un ejercicio de simulación a bajo nivel a implementaciones con herramientas especializadas, el camino recorrido no ha servido solo para validar una lógica de negocio determinada, sino también para poner en perspectiva las verdaderas diferencias que supone hacerlo todo a mano frente a confiar en plataformas consolidadas como PowerCenter e IICS. La primera gran conclusión extraída de este ejercicio es el coste oculto de la programación manual que podría parecer una solución flexible, económica o incluso “a la carta”, pero la experiencia práctica nos dice que no siempre eso se traduce en eficiencia. El tiempo dedicado a escribir, testar, depurar, mantener código es alto y lo peor de todo, cada cambio por menor que sea, conlleva tener que volver a abrir ese código y repetir el ciclo. Esto limita la evolución del proceso y lo hace vulnerable a los cambios personales o de contexto. Frente a esta rigidez, el empleo de herramientas ETL profesionales como PowerCenter ha demostrado su robustez estructural, la posibilidad de modelar procesos

visualmente, aplicar validaciones en cada etapa, controlar la ejecución a través de logs detallados y programar tareas desde una consola centralizada ha permitido tener una visión completa del ciclo de vida de cada proceso.

Pero ese mismo poder conlleva un coste, una curva de aprendizaje más complicada, una interfaz más clásica y la necesidad de una infraestructura que ahora, en escenarios más avanzados, empieza a quedarse obsoleta y aquí es donde IICS es el punto de inflexión de estas herramientas, con su modelo nativo de nube, interfaz simplificada y escalabilidad, es una herramienta especialmente diseñada para empresas que están o planean estar en entornos de nube. Las ganancias operativas de menor mantenimiento, entrega más rápida de procesos y fácil conexión a otros servicios son evidentes y aportan un beneficio directo al día a día pero, más allá de lo técnico, IICS también aporta otra forma de entender el trabajo del equipo de datos, menos ligado a la infraestructura y más al valor del negocio. Comparando estas dos, de forma práctica, podemos ver que ambas están diseñadas para satisfacer diferentes necesidades, el desarrollo manual puede utilizarse como prueba de concepto, de forma académica o en escenarios muy puntuales, PowerCenter, como la herramienta perfecta para empresas que quieren tener el máximo control y ya están invertidas en una infraestructura robusta e IICS, como la herramienta para una forma de tratamiento de datos más flexible, más conectada, más alineada con la velocidad a la que se mueven los datos hoy en día, en cualquier tipo de empresa.

En el lado organizacional, el impacto de pasar de PowerCenter a IICS también ha sido enorme, no se trataba solo de cambiar una herramienta por otra, sino de cambiar el proceso, reducir el tiempo de respuesta, permitir que más equipos colaboren y la posibilidad de empezar a explorar nuevas formas de gestión de datos. El empoderamiento que aporta IICS permite descentralizar algunas tareas, distribuir la propiedad de los datos entre más roles y eliminar muchos de los cuellos de botella que un proceso más rígido genera. En la misma línea de evolución, también es interesante ver cómo las nuevas plataformas de integración, como IICS, están preparadas para integrarse con los nuevos ecosistemas analíticos que lideran el mercado. Hoy en día, IICS no es solo una plataforma de integración, sino un componente central en una arquitectura de datos avanzada que incluye herramientas como Snowflake, Databricks o Azure Synapse. Estas plataformas ofrecen entornos de nube diseñados para análisis, aprendizaje automático y ejecución de cargas de trabajo de datos masivas, y la interoperabilidad con IICS aumenta enormemente las posibilidades del proceso de integración. El uso de conectores nativos permite a IICS comunicarse directamente con estas plataformas y realizar procesos ELT, donde las

transformaciones se realizan en el destino, aprovechando las capacidades de computación de las plataformas analíticas, por ejemplo, en Snowflake IICS emplea el "Snowflake Data Cloud Connector" que habilita la lectura y escritura directas hacia Snowflake desde mappings de IICS, en Databricks la integración oficial de IICS permite conectar con el Lakehouse y realizar transformaciones optimizadas, y aunque la documentación pública de Azure Synapse-específica es más limitada, IICS admite conectividad hacia entornos Azure de análisis y procesamiento estructurado/no estructurado [43][45][46].

Más allá de la integración con herramientas de análisis, el avance hacia procesos en tiempo real se presenta como la siguiente frontera natural, IICS incorpora funcionalidades orientadas al real-time streaming mediante su servicio "Streaming Ingestion and Replication", que permite ingerir datos de fuentes como Kafka, MQTT, REST o Event Hubs y enviarlos a destinos analíticos o lago de datos en tiempo real [46]. Esto aborda el requisito de toma de decisiones a demanda y menor latencia en escenarios en tiempo real. Según los últimos desarrollos en DataOps, la automatización, el monitoreo en tiempo real y la integración con servicios de eventos harán que IICS se mueva hacia un modelo más reactivo y predictivo donde los flujos de integración serán en tiempo real, inteligentes y completamente orquestados desde la nube [47].

En resumen, esta capacidad de integrarse con fuentes de datos modernas y flujos de datos en tiempo real demuestra que IICS no es solo una mejora técnica de PowerCenter, sino también un movimiento estratégico hacia una arquitectura de datos y análisis más conectada, automatizada e inteligente. Esta es la dirección para el futuro de la integración de datos, donde la colaboración de los servicios en la nube y los modelos de DataOps se establecen como los bloques de construcción centrales de los sistemas empresariales de próxima generación.

4. CONCLUSIONES

4.1 Síntesis de los hallazgos del TFM

Me he sumergido totalmente en el mundo de la integración de datos mientras desarrollaba este trabajo, abordando el tema tanto desde el punto de vista teórico como práctico. Con el desarrollo de un caso real, he podido observar de primera mano, con ejemplos y resultados, cómo el enfoque, el esfuerzo requerido y el resultado obtenido cambian en función de la herramienta o metodología empleada.

Una de las mayores lecciones aprendidas ha sido la diferencia entre abordar un proceso

de integración a mano, utilizando un lenguaje de bajo nivel como C, y hacerlo con herramientas ETL específicas. El hecho a mano, aunque técnicamente válido, ha mostrado la complejidad de tener que controlar explícitamente cada fase del flujo de datos: desde la lectura del archivo de entrada línea por línea, el procesamiento lógico para la detección de duplicados y la escritura de las salidas requeridas. Esto me ha permitido ver de primera mano cómo un simple error, como un salto de línea mal gestionado o una coma mal colocada, puede llevar a un resultado final erróneo aunque la lógica general sea correcta. Además, la necesidad de prever y programar cada detalle hace que este tipo de soluciones sean algo frágiles, poco escalables y difíciles de mantener a medio o largo plazo.

Por otro lado, el uso de herramientas como PowerCenter e IICS ha cambiado radicalmente la forma de concebir e implementar estos procesos ya que, a nivel funcional, ambas plataformas han permitido la reproducción fiel y precisa de la lógica de negocio definida, pero con un esfuerzo muy reducido, menos propenso a errores humanos y con mayor facilidad para verificar cada fase. Especialmente destacable ha sido la posibilidad que ofrecen las herramientas ETL de incorporar transformaciones complejas, gestionar visualmente condiciones, aplicar reglas de negocio a través de sencillas expresiones y obtener logs detallados sin tener que implementar funciones personalizadas. Todo ello implica una mejora sustancial de la productividad que va más allá del tiempo de desarrollo puesto que también influye en la documentación, el mantenimiento posterior y la capacidad de adaptar el proceso a nuevos requisitos.

La comparación entre PowerCenter, como herramienta consolidada en entornos tradicionales y IICS, como plataforma orientada a la nube, también ha proporcionado una visión realista de cómo está evolucionando el mercado de las herramientas de integración. Aunque comparten elementos comunes, el salto de una a otra no es solo técnico, sino también organizativo. IICS presenta una lógica más ágil, ligera y conectada a los entornos que se manejan hoy en día, donde los datos provienen de múltiples fuentes y se consumen en tiempo real o con latencia mínima. También ha quedado claro que la elección de la herramienta no debe basarse únicamente en su potencia o presencia en el mercado, sino en su idoneidad para las necesidades reales de la organización. Una herramienta puede ser enormemente potente, pero si implica la necesidad de mantener infraestructuras sobredimensionadas o dificulta los ajustes sobre la marcha, es ineficiente. En este sentido, el análisis del impacto de la migración en la actividad diaria ha sido particularmente esclarecedor.

Por lo tanto, como conclusión transversal, este trabajo demuestra que el valor de una solución de integración no reside solo en su capacidad para mover datos, sino en todo lo que se habilita a su alrededor, facilidad de concepción, rapidez de implementación, trazabilidad de errores, adaptabilidad al cambio y adecuación al entorno técnico y organizativo. Esta visión, que combina la experiencia práctica con el análisis comparativo, ha permitido construir una visión sólida del estado actual de las soluciones de integración de datos, así como de los retos y oportunidades que existen en este campo. La integración de datos ya no es un proceso de fondo de segundo orden, sino una parte estratégica del sistema de información, y su correcta implementación tiene un impacto real en la calidad de la toma de decisiones y en la eficiencia operativa.

4.2 Valoración de la experiencia práctica

La realización de este proyecto ha sido mucho más que un ejercicio técnic, ha sido en cierto modo, un proceso de aprendizaje haciendo, donde cada etapa ha planteado desafíos que no siempre se contemplan en guías ni se abordan en cursos teóricos. Partir de cero con las particularidades, necesidades y limitaciones de la integración de datos ha sido una lección inmensa sobre lo que significa planificar, ejecutar y mantener un proceso ETL en condiciones reales.

Uno de los mayores enriquecimientos ha sido precisamente el de enfrentarse a un caso de uso real extraído del mundo profesional. Esto ha implicado tomar decisiones con información parcial, priorizar por plazos de entrega y adaptarse a necesidades que no siempre eran definitivas desde el principio. A diferencia de los ejercicios prácticos, en este proyecto fue necesario captar los requisitos del negocio, averiguar cómo traducirlos al lenguaje de los datos y equilibrar entre rendimiento y simplicidad.

Abordar el proceso primero de forma manual, desarrollando el flujo en C, ha sido un ejercicio enormemente didáctico no solo para asimilar cada paso del proceso, desde abrir un archivo, leer secuencialmente, identificar duplicados, limpiar y escribir archivos segregados, sino para darse cuenta de cómo cada pequeña elección técnica tiene repercusiones muy tangibles en el resultado, elementos tan aparentemente menores como los saltos de línea, la gestión de comas o el tratamiento de registros vacíos, que son prácticamente invisibles en las herramientas ETL, aquí exigieron atención y correcciones repetidas.

Esta primera etapa afianzó el concepto de no subestimar la complejidad que puede tener un proceso aparentemente simple cuando se lleva a cabo desde cero, la experiencia de

escribir, depurar, reescribir y validar cada línea de código dejó muy claro por qué existen herramientas específicas para ello: no por comodidad, sino por necesidad. El tiempo ahorrado, los errores evitados, la seguridad ganada con herramientas que automatizan y estructuran las tareas no es un privilegio, sino una condición que posibilita el mantenimiento a medio y largo plazo.

A nivel personal, también ha sido un ejercicio de cotejar y solidificar aprendizajes previos con escenarios reales, muchas veces, al trabajar con plataformas visuales como PowerCenter o IICS, se pierde la conciencia de lo que ocurre en el "backend". Haber emulado ese proceso de forma artesanal me permitió volver a lo básico, cómo se leen los datos, en qué consiste comparar registros, cómo se maneja la memoria, qué ocurre si un campo es nulo o si una línea está incompleta. La vuelta al origen no solo fortaleció la visión técnica, sino que proporcionó herramientas para detectar errores o inconsistencias con mayor facilidad al trabajar con plataformas más evolucionadas.

Durante la fase de implementación en PowerCenter e IICS, la experiencia fue diferente pero igualmente enriquecedora. Aquí, el desafío fue más organizativo que técnico, comprender cómo se organizan los mapeos, cómo se orquestan las transformaciones, qué implicaciones tiene cada elección de diseño y cómo todo ello se enlaza con destinos reales como Salesforce o archivos para exportar. En este contexto, lo más destacable ha sido la perspectiva ganada cuando los procesos están bien mapeados y visualizados. La posibilidad de trazar cada campo, validar transformaciones en cada paso y aislar fácilmente errores proporciona una seguridad operativa difícil de lograr con desarrollos manuales.

También ha sido interesante experimentar cómo la herramienta influye en la forma de trabajar, PowerCenter ofrece un entorno más estructurado, más jerárquico, con una lógica heredada de una época en la que los desarrollos seguían un proceso de maduración más largo y controlado. Por otro lado, IICS ofreció una experiencia decididamente más ágil, donde los cambios se pueden realizar sobre la marcha y donde todo parece diseñado para facilitar el trabajo en equipo y la integración rápida con otros sistemas.

Desde la perspectiva del crecimiento personal, el proyecto también ha sido un medio para potenciar habilidades transversales: gestión del tiempo, priorización de tareas, documentación clara y, sobre todo, resolución de problemas sin una solución inmediata. Muchas veces, el camino no estaba trazado y había que tomar decisiones basándose en pruebas, comparaciones y sentido común y esa incertidumbre, lejos de ser un escollo, fue parte fundamental del proceso de aprendizaje.

En definitiva, más allá del resultado técnico, lo que deja esta travesía es una comprensión mucho más integral del proceso ETL, no como una serie de operaciones técnicas, sino como un proceso vivo, donde cada decisión importa, cada herramienta tiene su momento, y donde lo verdaderamente importante no es aplicar recetas, sino captar el por qué detrás de cada acción.

4.3 Posibles líneas de mejora o continuación

Aunque este trabajo ha podido cubrir los objetivos que inicialmente se plantearon y ha podido realizar un estudio detallado sobre las diferentes aproximaciones de integración de datos, son muchos los puntos que aún se pueden profundizar o son susceptibles de servir como continuación a este. La integración de datos, en sí, es un tema amplio, mutable y con una fuerte dependencia de la evolución tecnológica, por lo que siempre se puede ampliar su alcance o dar otro enfoque.

Un primer punto a mejorar es, profundizar en la automatización de ciertas tareas, especialmente en el control de calidad del dato. En un entorno donde los datos son cada vez más diversos y llegan desde múltiples fuentes, asegurar que se cumplen ciertas reglas de validación, formatos o integridad referencial se vuelve una necesidad crítica. Si bien las herramientas utilizadas ya permiten implementar reglas, un paso siguiente podría consistir en construir módulos específicos de control que funcionen como bloques reutilizables y que detecten patrones de error, inconsistencias o anomalías con mayor autonomía.

Otra línea de desarrollo interesante sería la integración de flujos ETL con entornos de procesamiento en tiempo real. Las soluciones analizadas en este TFM trabajan, principalmente, en modo batch, es decir, procesan los datos en bloques una vez estos están disponibles. Sin embargo, en determinados contextos (como plataformas de e-commerce, sistemas de monitorización o servicios financieros), los datos se generan y deben procesarse en el mismo instante en que ocurren. Explorar herramientas o arquitecturas que combinen lo mejor de los procesos batch con capacidades de streaming sería una evolución natural y alineada con las exigencias actuales del mercado.

Además, con el auge de tecnologías como machine learning o la inteligencia artificial aplicada al dato, se abre un campo muy prometedor en el que los procesos de integración no solo transportan datos, sino que también toman decisiones inteligentes sobre ellos, por ejemplo, ajustar reglas de negocio en función de patrones históricos, detectar comportamientos anómalos en los datos durante el proceso de carga o adaptar los

mappings automáticamente en función de la calidad de los registros. Explorar este tipo de inteligencia integrada puede llevar a procesos más dinámicos, más fiables y menos dependientes de intervención humana constante.

Otra mejora potencial está relacionada con la experiencia de usuario en el diseño de procesos, aunque las herramientas actuales ofrecen interfaces visuales muy completas, aún existe cierta barrera para usuarios no técnicos. Una evolución futura podría orientarse a crear asistentes que, a partir de una descripción funcional o de ejemplos concretos, propongan estructuras iniciales de mappings o transformaciones ya que esto permitiría democratizar aún más el acceso al diseño de procesos, reduciendo la dependencia de perfiles técnicos especializados.

Por último, en términos de continuidad, este trabajo podría servir como base para definir una guía práctica de migración de procesos entre herramientas, muchas organizaciones se encuentran en este momento, ante la decisión de si migrar sus procesos ETL tradicionales a soluciones en la nube. Contar con un modelo probado, con casos reales documentados y con buenas prácticas derivadas de la experiencia podría ayudar a otros equipos a tomar decisiones más informadas y evitar errores comunes en este tipo de transiciones.

De cara a futuros desarrollos, una línea especialmente prometedora consiste en profundizar en la integración de Informatica IICS con plataformas analíticas de nueva generación como Snowflake, Databricks o Azure Synapse. Los posibles nuevos flujos híbridos que permitan integrar, transformar y analizar en un mismo entorno cloud, suprimiendo al mismo tiempo intermediaciones, son conexiones nativas [43][44][45] que en un futuro pueden fluir hacia arquitecturas completamente automatizadas donde IICS sea el núcleo central de todo el ciclo de vida de los datos. De la misma manera, la progresiva implantación de pipelines en tiempo real y la adopción progresiva de modelos Zero-ETL constituyen un campo de investigación altamente relevante. Profundizar en el tratamiento de flujos en streaming por parte de IICS y su capacidad de adaptación a entornos event-driven permitirá explorar nuevas estrategias de integración continua y analítica predictiva alineadas con los nuevos paradigmas modernos de DataOps [46][47]. De esta forma, la plataforma podrá perfilarse no solo como un elemento de integración, sino como una pieza clave en la creación de ecosistemas de datos inteligentes, resistentes y conectados.

En resumen, pese a que en este trabajo se han cumplido los objetivos marcados, se han dejado varias puertas abiertas a caminos que merecen ser transitados. La integración de

datos, lejos de ser un tema cerrado, es una materia en plena transformación, y estar a la par de su evolución es fundamental para seguir generando soluciones útiles, sostenibles y adaptadas a un entorno en permanente cambio.

Como parte del cierre del trabajo es pertinente mostrar la planificación seguida a lo largo de la realización del TFM. A continuación, se muestra un cronograma que resume las distintas etapas realizadas, desde la definición inicial del proyecto hasta la redacción final del informe, y que permite la visualización estructurada de la organización temporal de la labor académica emprendida.

	2024																													2025															
Tarea	05-feb	12-feb	19-feb	26-feb	04-mar	11-mar	18-mar	25-mar	01-abr	08-abr	...	03-jun	10-jun	17-jun	24-jun	01-jul	08-jul	15-jul	22-jul	29-jul	05-ago	12-ago	19-ago	...	09-sep	16-sep	23-sep	...	14-oct	21-oct	...	03-mar	10-mar	17-mar	24-mar	...	13-oct	20-oct	27-oct	03-nov	10-nov	17-nov	24-nov	01-dic	08-dic	15-dic
Análisis y diseño técnico Power Center	x																																											
Construcción	x	x	x	x	x	x	x																																					
Nueva malla Control-M							x	x																																				
Pruebas integradas y conjuntas								x	x																																			
Despliegue en producción										x																																		
Validación										x																																		
Migración de Power Center a IICS												x	x	x	x																													
Pruebas integradas y conjuntas																	x	x	x	x	x	x	x																					
Modificación mallas Control-M																									x	x	x																		
Puesta en producción																													x	x															
Definición del proyecto del TFM y análisis del caso real																																x	x												
Revisión del estado del arte																																		x	x										
Desarrollo de la simulación manual																																				x	x								
Análisis comparativo y estudio del TCO																																						x	x	x					
Redacción de la memoria del TFM																																									x	x	x	x	x

FIGURA 10: Cronograma del TFM

5. ANEXOS

5.1 Código de la simulación del proceso ETL manual

El archivo de entrada, denominado f_campana_outbound_rel_poliza_PROCESO_MANUAL.csv, tiene el siguiente formato:

Encabezados:

CodigoCampana: Código único que identifica la campaña comercial.

NombreTomador: Nombre del cliente asociado a la campaña.

DNITomador: Documento de identidad del cliente (clave única por cliente).

Email: Correo electrónico del cliente.

Telefono_1: Número de contacto del cliente.

Poliza: Código de la póliza asociada a la campaña.

Descripcion_producto: Descripción del producto ofrecido en la campaña.

Oferta_Retencion: Información sobre la oferta realizada o descuento aplicado.

Ejemplo de los datos de entrada:

CodigoCar	Nombre_Tomador	DNI_Tomador	Email	Telefono_1	Poliza	Descrip	Oferta_Retencion
CAMP001	JUAN PÉREZ	12345678A	juan.perez@e	600123456	P001	Auto	2 MESES GRATIS
CAMP001	JUAN PÉREZ	12345678A	juan.perez@e	600123456	P002	Hogar	DESCUENTO 20%
CAMP002	ANA GARCÍA	98765432B	ana.garcia@e	600987654	P003	Vida	3 MESES GRATIS
CAMP002	ANA GARCÍA	98765432B	ana.garcia@e	600987654	P004	Salud	DESCUENTO 20%
CAMP003	PEDRO LÓPEZ	11223344C	pedro.lopez@	600112233	P005	Auto	DESCUENTO 20%
CAMP003	PEDRO LÓPEZ	11223344C	pedro.lopez@	600112233	P005	Hogar	3 MESES GRATIS
CAMP003	PEDRO LÓPEZ	11223344C	pedro.lopez@	600112233	P006	Salud	DESCUENTO 10%
CAMP003	PEDRO LÓPEZ	11223344C	pedro.lopez@	600112233	P006	Hogar	3 MESES GRATIS
CAMP004	MARÍA LÓPEZ	44556677D	maria.lopez@	600334455	P006	Vida	DESCUENTO 10%
CAMP005	CARLOS RUIZ	22334455E	carlos.ruiz@e	600445566	P006	Hogar	3 MESES GRATIS
CAMP005	CARLOS RUIZ	22334455E	carlos.ruiz@e	600445566	P007	Auto	3 MESES GRATIS
CAMP006	CARLOS RUIZ	22334455E	carlos.ruiz@e	600445566	P007	Auto	3 MESES GRATIS

TABLA 2: Ejemplo de datos de entrada

Salidas esperadas:

Se deberá generar un archivo para Salesforce (salesforce_target.csv) que contenga únicamente un registro por cada combinación única de CodigoCampana y DNITomador. En aquellos casos en los que existan duplicados, los campos Poliza, Descripcion_producto y Oferta_Retencion permanecerán vacíos.

CodigoCar	Nombre_Tomador	DNI_Tomador	Email	Telefono_1	Poliza	Desc	Oferta_Retencion
CAMP001	JUAN PÉREZ	12345678A	juan.perez@	600123456			
CAMP002	ANA GARCÍA	98765432B	ana.garcia@	600987654			
CAMP003	PEDRO LÓPEZ	11223344C	pedro.lopez(	600112233			
CAMP004	MARÍA LÓPEZ	44556677D	maria.lopez(	600334455	P006	Vida	DESCUENTO 10%
CAMP005	CARLOS RUIZ	22334455E	carlos.ruiz@	600445566			
CAMP006	CARLOS RUIZ	22334455E	carlos.ruiz@	600445566	P007	Auto	3 MESES GRATIS

TABLA 3: Salida esperada a cargar en Salesforce

Por otro lado, también se generará un archivo plano (archivo_target.csv) que incluya todos los registros del archivo original, manteniendo la información completa incluso en los casos de duplicidad.

Salida esperada (ejemplo):

CodigoCar	Nombre_Tomador	DNI_Tomador	Email	Telefono_1	Poliza	Descri	Oferta_Retencion
CAMP001	JUAN PÉREZ	12345678A	juan.perez@	600123456	P001	Auto	2 MESES GRATIS
CAMP001	JUAN PÉREZ	12345678A	juan.perez@	600123456	P002	Hogar	DESCUENTO 20%
CAMP002	ANA GARCÍA	98765432B	ana.garcia@	600987654	P003	Vida	3 MESES GRATIS
CAMP002	ANA GARCÍA	98765432B	ana.garcia@	600987654	P004	Salud	DESCUENTO 20%
CAMP003	PEDRO LÓPEZ	11223344C	pedro.lopez(	600112233	P005	Auto	DESCUENTO 20%
CAMP003	PEDRO LÓPEZ	11223344C	pedro.lopez(	600112233	P005	Hogar	3 MESES GRATIS
CAMP003	PEDRO LÓPEZ	11223344C	pedro.lopez(	600112233	P006	Salud	DESCUENTO 10%
CAMP003	PEDRO LÓPEZ	11223344C	pedro.lopez(	600112233	P006	Hogar	3 MESES GRATIS
CAMP005	CARLOS RUIZ	22334455E	carlos.ruiz@	600445566	P006	Hogar	3 MESES GRATIS
CAMP005	CARLOS RUIZ	22334455E	carlos.ruiz@	600445566	P007	Auto	3 MESES GRATIS

TABLA 4: Salida esperada del fichero

Código para la extracción:

Aquí está el código que realiza la extracción de datos:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_LINE 1024
#define MAX_RECORDS 1000

typedef struct {
    char CodigoCampana[20];
    char NombreTomador[100];
```

```
    char DNITomador[20];
    char Email[100];
    char Telefono[20];
    char Poliza[20];
    char DescripcionProducto[100];
    char OfertaRetencion[100];
    int is_duplicate;
} Record;

void process_file(const char *input_file, const char *salesforce_file, const char
*archivo_file) {
    FILE *infile = fopen(input_file, "r");
    if (!infile) {
        perror("No se pudo abrir el archivo de entrada");
        exit(EXIT_FAILURE);
    }

    FILE *salesforce_outfile = fopen(salesforce_file, "w");
    FILE *archivo_outfile = fopen(archivo_file, "w");
    if (!salesforce_outfile || !archivo_outfile) {
        perror("No se pudo crear el archivo de salida");
        fclose(infile);
        exit(EXIT_FAILURE);
    }

    Record records[MAX_RECORDS];
    int record_count = 0;

    char line[MAX_LINE];

    // Leer la cabecera y escribirla en ambos archivos de salida
    if (fgets(line, MAX_LINE, infile)) {
        fprintf(salesforce_outfile, "%s", line);
```

```
        fprintf(archivo_outfile, "%s", line);
    } else {
        printf("El archivo de entrada está vacío o tiene un formato incorrecto.\n");
        fclose(infile);
        fclose(salesforce_outfile);
        fclose(archivo_outfile);
        return;
    }

    // Leer y procesar cada línea del archivo de entrada
    while (fgets(line, MAX_LINE, infile)) {
        Record record;
        char *token = strtok(line, ",");
        strcpy(record.CodigoCampana, token ? token : "");
        token = strtok(NULL, ",");
        strcpy(record.NombreTomador, token ? token : "");
        token = strtok(NULL, ",");
        strcpy(record.DNITomador, token ? token : "");
        token = strtok(NULL, ",");
        strcpy(record.Email, token ? token : "");
        token = strtok(NULL, ",");
        strcpy(record.Telefono, token ? token : "");
        token = strtok(NULL, ",");
        strcpy(record.Poliza, token ? token : "");
        token = strtok(NULL, ",");
        strcpy(record.DescripcionProducto, token ? token : "");
        token = strtok(NULL, ",");
        strcpy(record.OfertaRetencion, token ? token : "");

        record.is_duplicate = 0;

        // Verificar duplicados
```

```
        for (int i = 0; i < record_count; i++) {
            if (strcmp(records[i].CodigoCampana, record.CodigoCampana) == 0 &&
                strcmp(records[i].DNITomador, record.DNITomador) == 0) {
                records[i].is_duplicate = 1;
                record.is_duplicate = 1;
                break;
            }
        }

        records[record_count++] = record;
    }

    // Procesar registros para Salesforce
    for (int i = 0; i < record_count; i++) {
        Record *record = &records[i];

        // Verificar si es duplicado y si no es el primer duplicado de este grupo
        if (record->is_duplicate && i > 0 &&
            strcmp(records[i - 1].CodigoCampana, record->CodigoCampana) == 0 &&
            strcmp(records[i - 1].DNITomador, record->DNITomador) == 0) {
            continue; // Saltar duplicados adicionales
        }

        // Escribir registro en Salesforce
        if (record->is_duplicate) {
            fprintf(salesforce_outfile, "%s,%s,%s,%s,%s,,,\n",
                    record->CodigoCampana, record->NombreTomador, record-
>DNITomador,
                    record->Email, record->Telefono);
        } else {
            fprintf(salesforce_outfile, "%s,%s,%s,%s,%s,%s,%s,%s",
                    record->CodigoCampana, record->NombreTomador, record-
>DNITomador,
```

```
                record->Email, record->Telefono, record->Poliza,
                record->DescripcionProducto, record->OfertaRetencion);
        }
    }

    // Procesar duplicados para Archivo Plano
    for (int i = 0; i < record_count; i++) {
        Record *record = &records[i];
        if (record->is_duplicate) {
            fprintf(archivo_outfile, "%s,%s,%s,%s,%s,%s,%s,%s",
                record->CodigoCampana, record->NombreTomador, record->DNITomador,
                record->Email, record->Telefono, record->Poliza,
                record->DescripcionProducto, record->OfertaRetencion);
        }
    }

    fclose(infile);
    fclose(salesforce_outfile);
    fclose(archivo_outfile);

    printf("Procesamiento completado. Salidas generadas: %s, %s\n", salesforce_file, archivo_file);
}

void clean_file(const char *input_file, const char *output_file) {
    FILE *infile = fopen(input_file, "r");
    if (!infile) {
        perror("No se pudo abrir el archivo de entrada");
        exit(EXIT_FAILURE);
    }

    FILE *outfile = fopen(output_file, "w");
```

```
    if (!outfile) {
        perror("No se pudo crear el archivo de salida");
        fclose(infile);
        exit(EXIT_FAILURE);
    }

    char line[MAX_LINE];

    // Leer y procesar cada línea
    while (fgets(line, MAX_LINE, infile)) {
        // Comprobar si la línea comienza con una coma
        if (line[0] == ',') {
            // Eliminar la coma inicial desplazando la cadena
            memmove(line, line + 1, strlen(line));
        }

        // Escribir la línea procesada en el archivo de salida
        fprintf(outfile, "%s", line);
    }

    fclose(infile);
    fclose(outfile);

    printf("Limpieza completada. Archivo generado: %s\n", output_file);
}

int main() {
    const char *input_file =
"f_campana_outbound_rel_poliza_PROCESO_MANUAL.csv";
    const char *salesforce_file = "salesforce_target.csv";
    const char *archivo_file = "archivo_target_previo.csv";
            const char *cleaned_file = "archivo_target.csv";
```

```
    // Procesar el archivo de entrada y generar los targets
    process_file(input_file, salesforce_file, archivo_file);

    // Limpiar el archivo archivo_target_previo.csv para eliminar comas iniciales
    clean_file(archivo_file, cleaned_file);

    return 0;
}
```

TABLA 5: Código empleado para la ETL simulada de forma manual

El código utiliza la función strtok, que permite dividir cada línea del archivo en campos separados por comas. En caso de que algún campo esté vacío, este se almacena como una cadena vacía ("").

Los datos de cada línea se organizan mediante la estructura Record, que facilita su acceso y lectura de forma ordenada.

Para controlar la carga de información, se establece un límite de registros definido por MAX_RECORDS (en este caso, 1000), lo que evita posibles desbordamientos de memoria.

Finalmente, la primera línea del archivo, correspondiente a la cabecera, se lee únicamente para ser descartada, ya que no forma parte del conjunto de datos procesados.

5.2 Capturas adicionales de PowerCenter e IICS

Este apartado tiene como finalidad mostrar los elementos visuales que forman parte del diseño del proceso ETL en ambas herramientas, permitiendo al lector visualizar cómo están organizados los componentes, qué transformaciones se aplican y cómo fluye la información desde el origen hasta los destinos.

Primero con los componentes utilizados en PowerCenter:

Fuente de datos (Source Qualifier): Captura del objeto que representa el fichero de entrada. Indica si se han aplicado filtros, ordenaciones o condiciones al leer los datos.

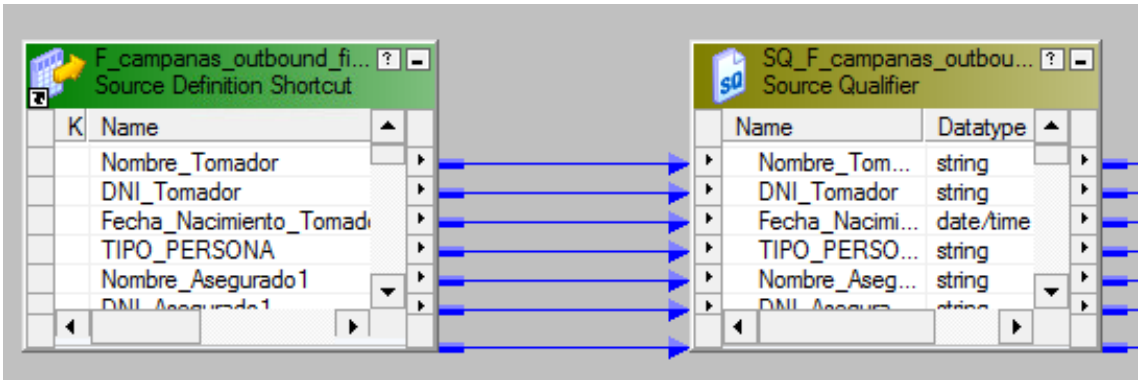


FIGURA 11: Origen de los datos en Powercenter

Ordenación y agrupación (Sorter + Aggregator): Imagen del flujo donde se aplica el orden por IdCampaña y ContactId, seguido de la agrupación para identificar duplicados.

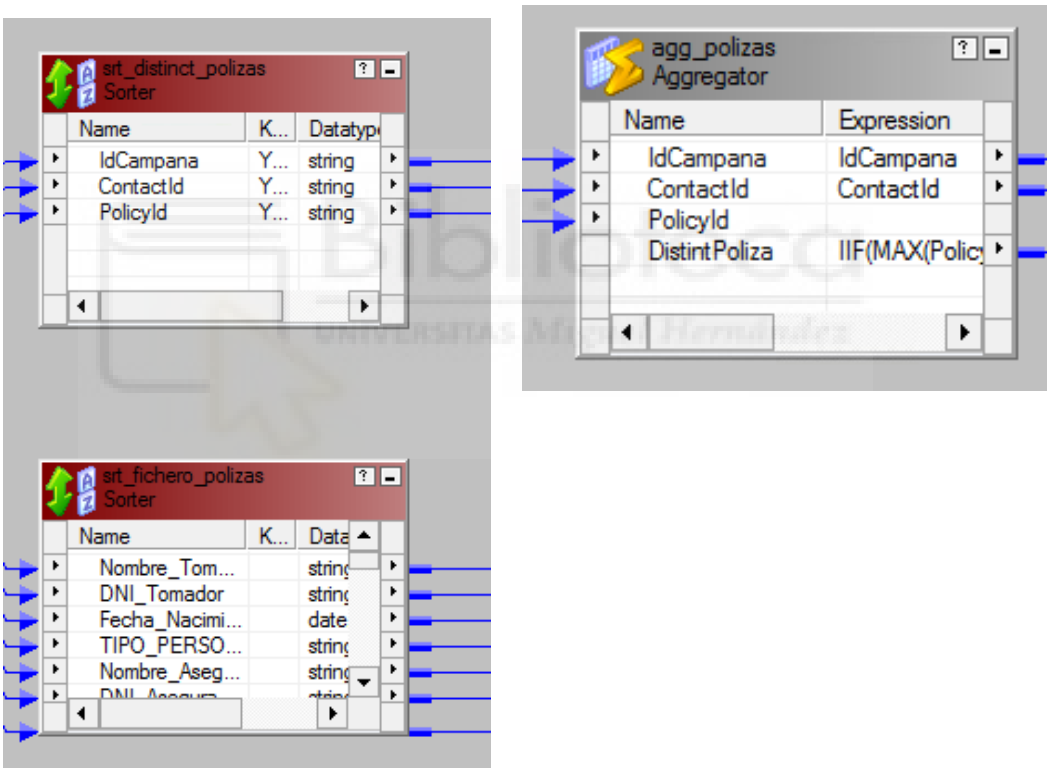


FIGURA 12: Sorters y Aggregator en Powercenter

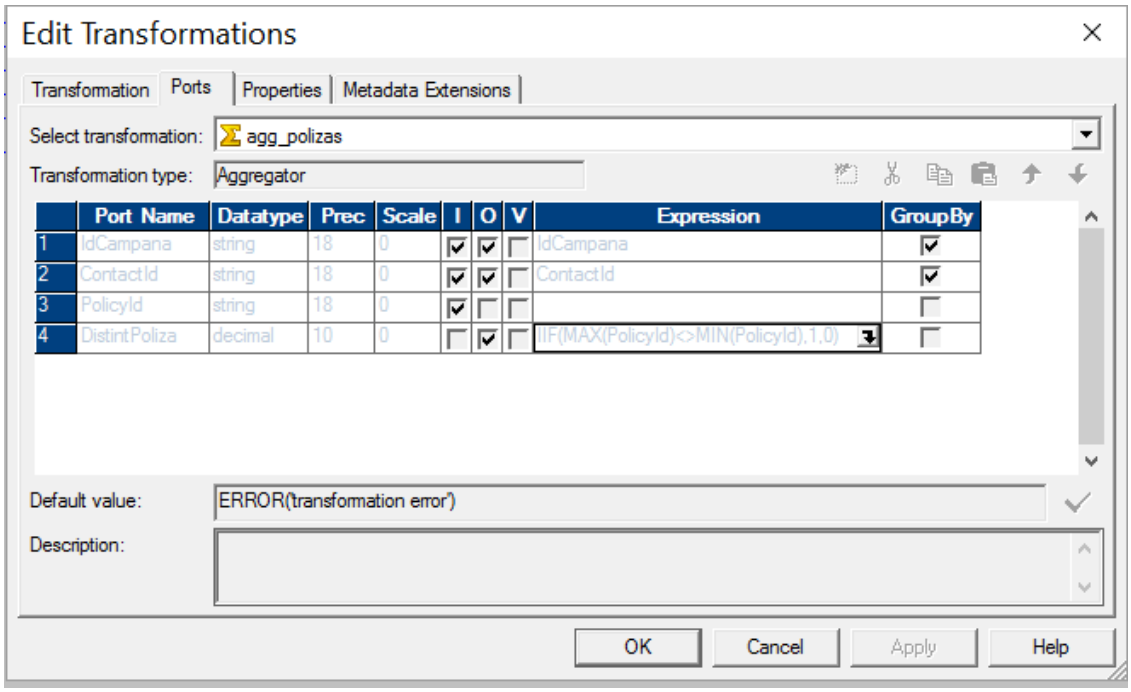


FIGURA 13: Detalles del Aggregator de Powercenter

En esta sección se representan los pasos de ordenación y agregación realizados sobre los datos obtenidos del fichero de entrada. El objetivo de este bloque es determinar si un mismo cliente (ContactId) ha sido incluido más de una vez dentro de la misma campaña (IdCampana) con distintas pólizas.

En primer lugar, mediante el componente Sorter, se ordenan los datos en función de los campos clave IdCampana, ContactId y PolicyId ya que es necesario para que la transformación siguiente, el Aggregator, funcione correctamente agrupando registros que pertenecen a una misma unidad lógica.

A continuación, se aplica el componente Aggregator agg_polizas, donde se agrupan los datos por IdCampana y ContactId. Dentro de este bloque se calcula un campo adicional denominado DistintPoliza, que evalúa si existen múltiples pólizas distintas asociadas a un mismo cliente en una misma campaña. La lógica empleada en su definición es la siguiente:

IIF(MAX(PolicyId) <> MIN(PolicyId), 1, 0)

Esta expresión permite identificar rápidamente si, dentro de cada grupo definido por campaña y cliente, existe más de una póliza. Si el valor máximo y el mínimo de PolicyId difieren, significa que hay al menos dos registros con distintas pólizas, por lo que DistintPoliza toma el valor 1. En caso contrario, el valor será 0.

Esta variable se utiliza más adelante en el flujo para filtrar aquellos registros que deben

tratarse de forma distinta: los clientes con varias pólizas serán dirigidos a un fichero plano, mientras que aquellos con una única póliza completa se enviarán a Salesforce con toda la información.

Joiner o expresión para marcar duplicados: Captura que muestra cómo se asocian los datos originales con la marca que identifica registros únicos o repetidos.

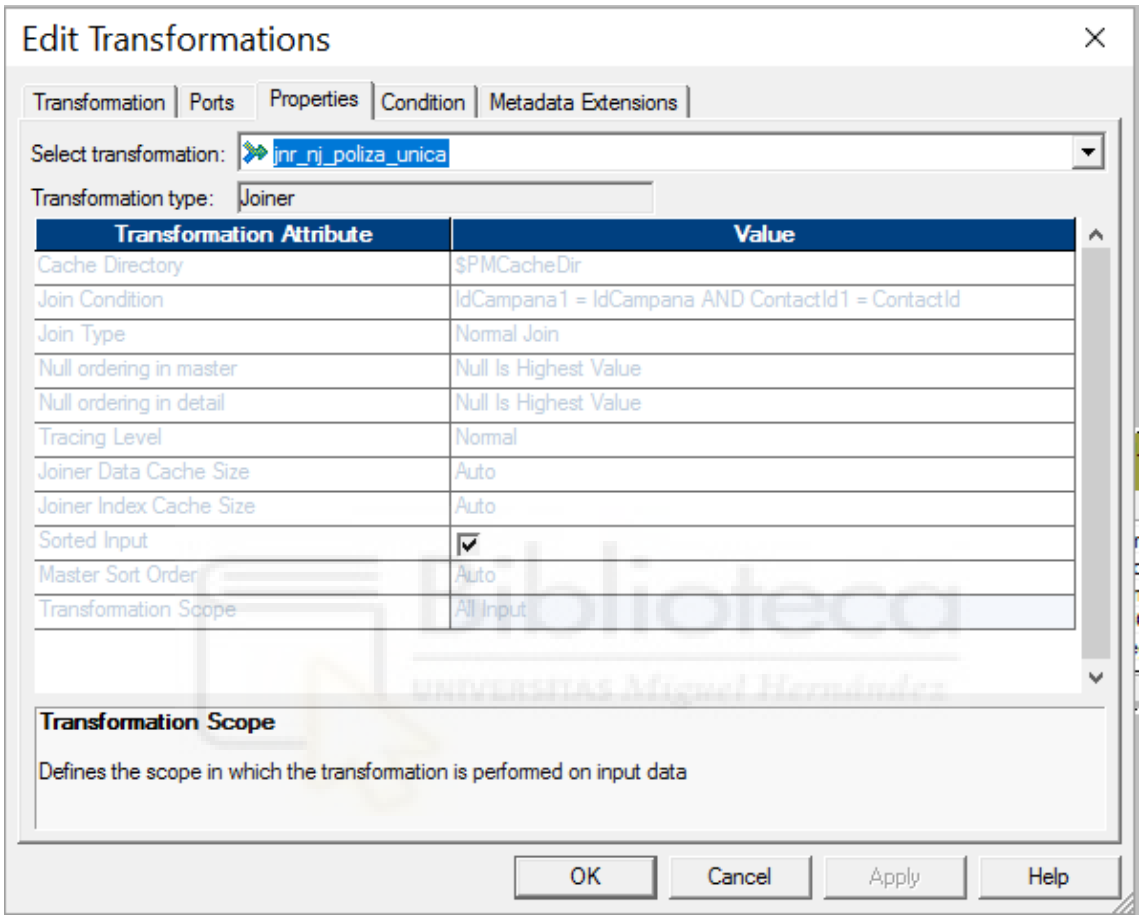


FIGURA 14: Detalles del joiner de Powercenter

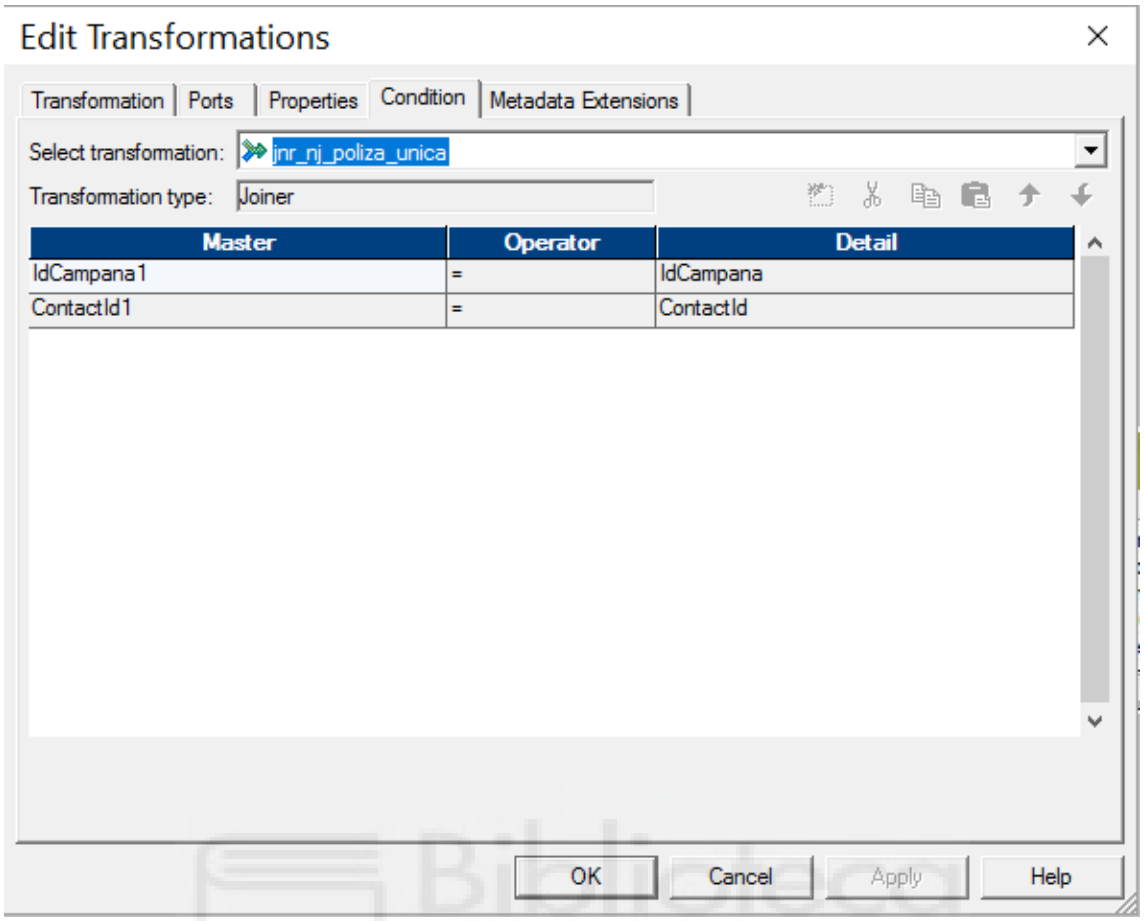


FIGURA 15: Información de los crucen en el joiner de Powercenter

Split del flujo: Imagen donde se visualiza claramente la bifurcación: una salida para Salesforce (con campos vacíos si hay duplicado), otra para fichero plano (registro completo).

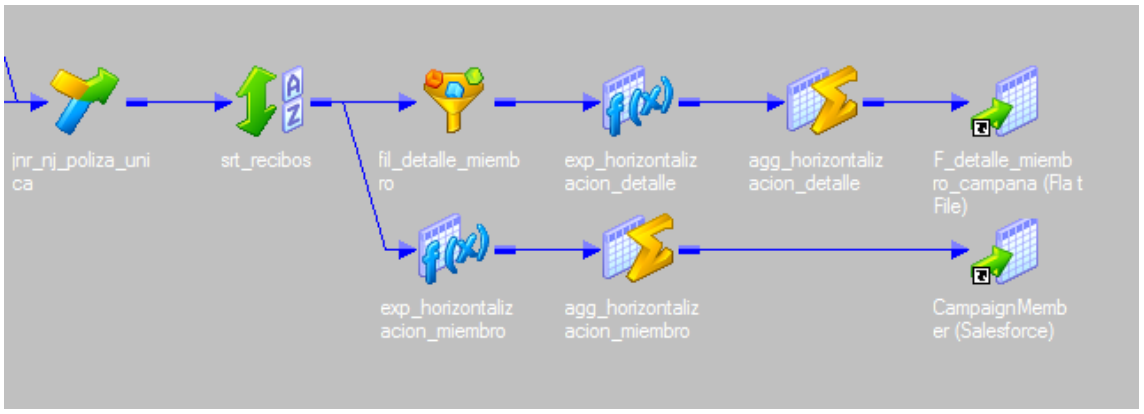


FIGURA 16: Vista del mapping a partir del cruce en Powercenter

El flujo de datos pasa por una transformación de tipo Filter denominada fil_detalle_miembro cuya finalidad dividir el flujo en función del número de pólizas detectadas para cada combinación única de campaña y cliente.

La condición aplicada en esta transformación es:

DistintPoliza = 1

Esto significa que solo se permiten pasar por esta rama aquellos registros que han sido clasificados previamente como duplicados, es decir, aquellos clientes que tienen más de una póliza asociada a una misma campaña. El resto de registros, aquellos donde DistintPoliza vale 0, son descartados en esta transformación y procesados en otra rama del flujo, donde serán enviados directamente a Salesforce con todos sus campos informados.

Este filtrado es fundamental para separar los casos en los que debe omitirse información sensible o redundante en el destino Salesforce, y derivarla en su lugar a un fichero plano. La lógica detrás de esta decisión se basa en la necesidad de evitar insertar múltiples veces al mismo cliente en Salesforce con diferentes datos de póliza. Al hacerlo así, se evita duplicidad de información y se mantiene una sola entrada por campaña y cliente, dejando los detalles adicionales registrados por separado.

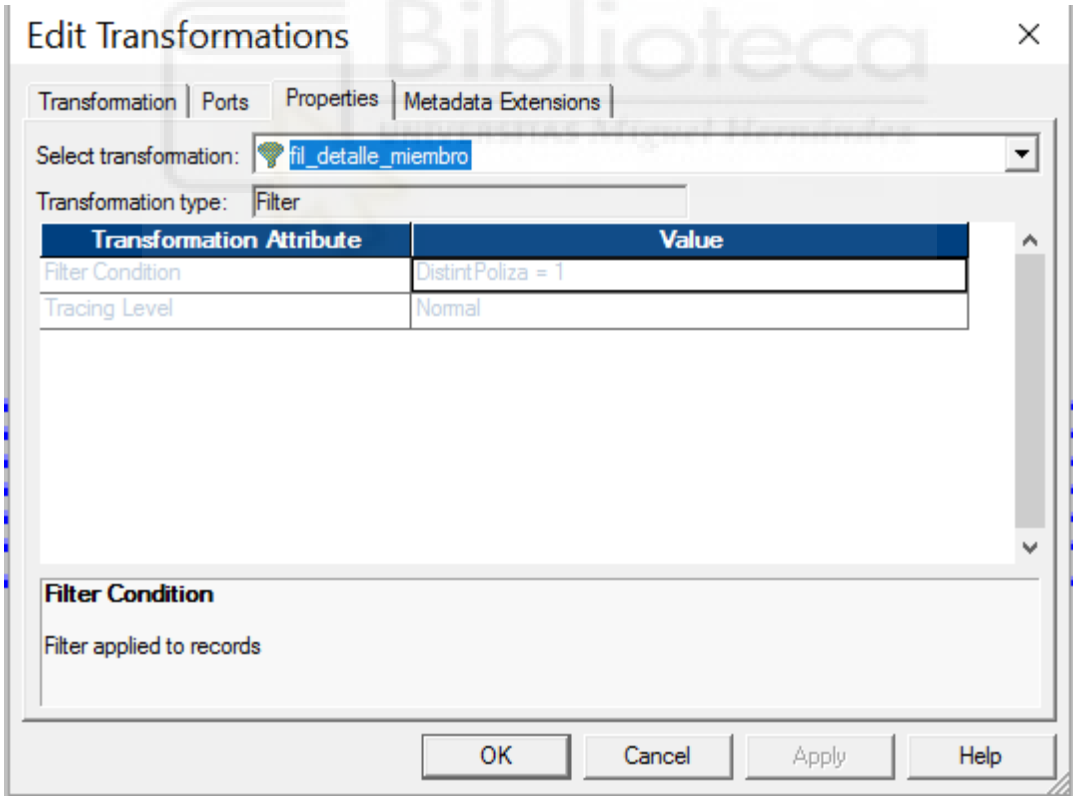


FIGURA 17: Detalles del filtro en Powercenter

Targets:

Captura final del diseño que muestra los destinos definidos y su relación con los flujos anteriores.

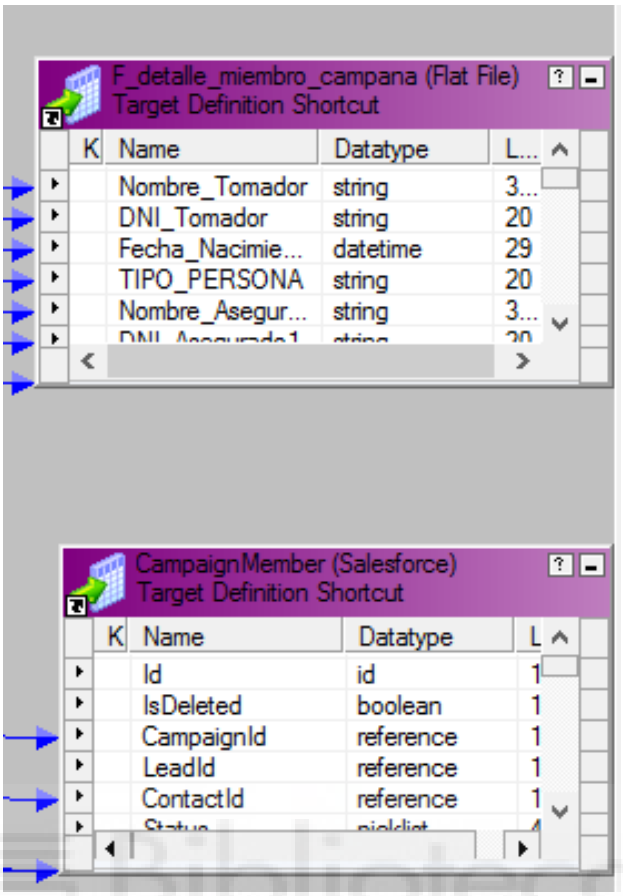


FIGURA 18: Targets de Powercenter

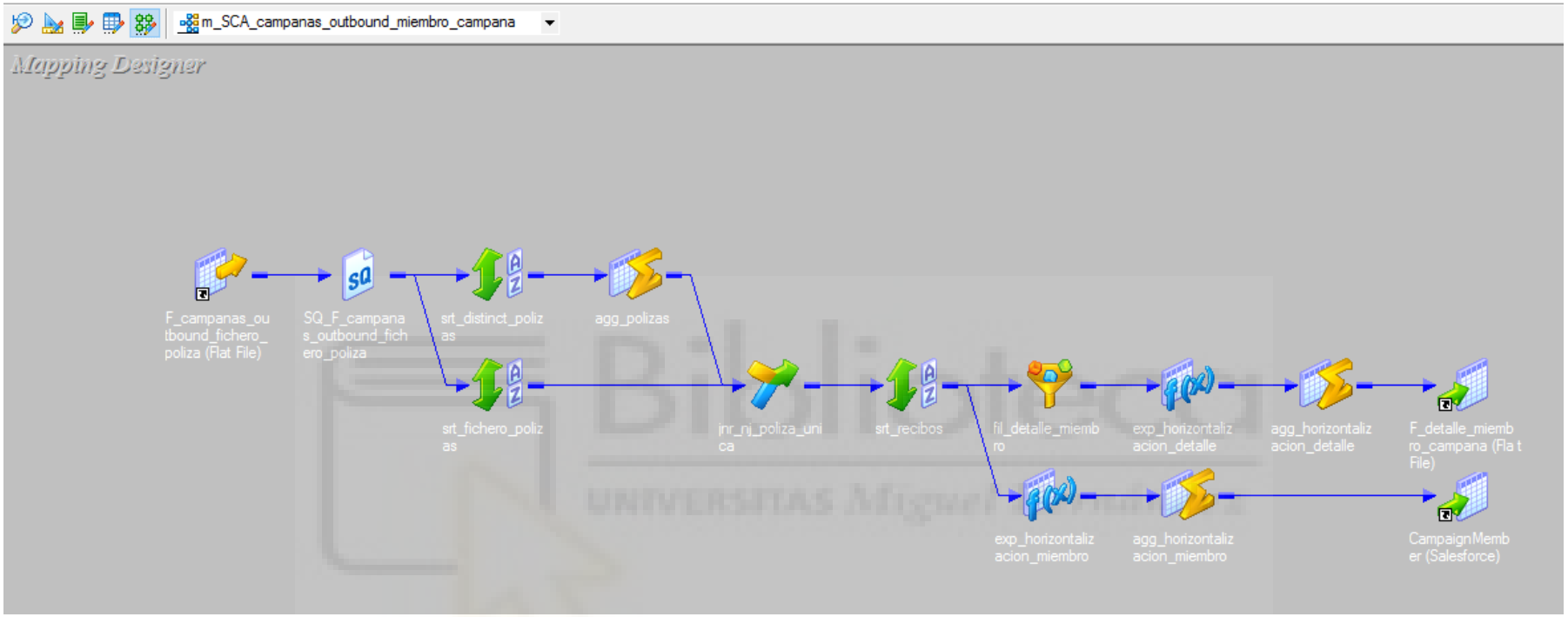


FIGURA 19: Vista del mapping completo en Powercenter

Ahora con los componentes utilizados en IICS;

Origen de datos (Source Qualifier): La fuente inicial corresponde a un componente que importa la información desde un archivo plano. Este objeto representa el punto de entrada al sistema y contiene los registros de campañas y clientes que serán procesados. La configuración del origen puede incluir delimitadores específicos, tipos de datos por columna y estructuras que facilitan el reconocimiento automático de los campos.

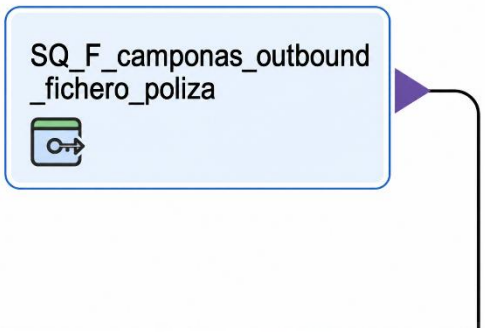


FIGURA 20: Source en IICS

Ordenación y agrupación (Sorter + Aggregator): Una vez leídos los datos, se procede a organizar los registros mediante un componente de ordenación que establece el orden por IdCampana, ContactId y PolicyId. Esta preparación garantiza que la agrupación posterior funcione de manera precisa, respetando la lógica de comparación entre registros relacionados.

El siguiente paso lo realiza un componente de tipo Aggregator, que agrupa los datos según las claves de campaña y cliente. Dentro de este bloque se genera un nuevo campo, DistintPoliza, cuya finalidad es detectar si existe más de una póliza vinculada al mismo cliente dentro de la misma campaña. Para ello, se emplea una expresión que compara los valores mínimo y máximo del campo PolicyId dentro de cada grupo y si hay discrepancia entre ambos, se asigna el valor 1, indicando la presencia de duplicados; si no, se asigna 0 de la siguiente forma:

IIF(MAX(PolicyId) <> MIN(PolicyId), 1, 0)

Esta lógica permite diferenciar entre casos en los que un cliente tiene una única póliza (situación sencilla) y aquellos donde aparece varias veces, con distinta información contractual, lo que requiere un tratamiento más específico en etapas posteriores del flujo.

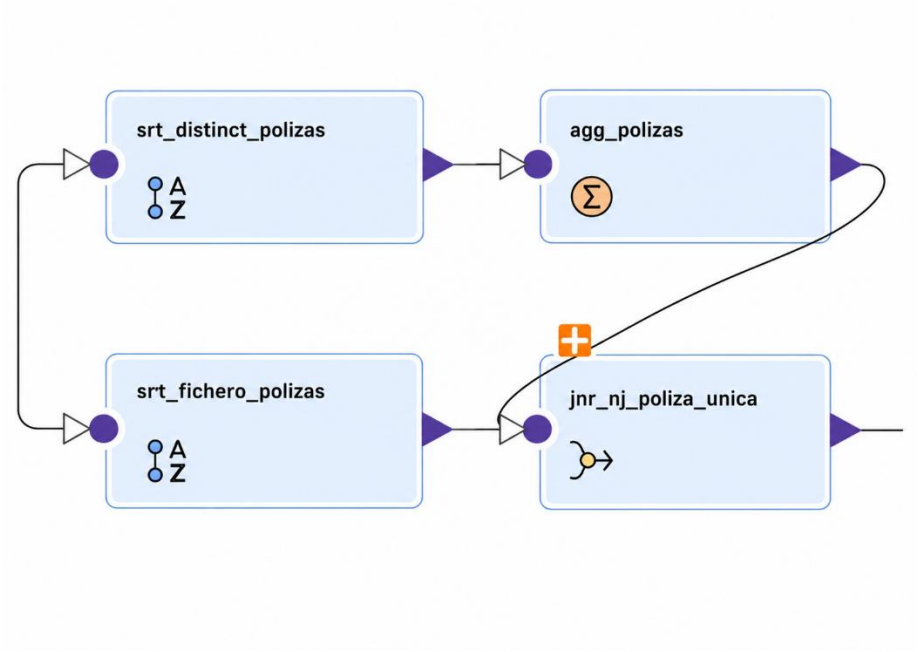


FIGURA 21: Distinct, agrgregator y joiner en IICS

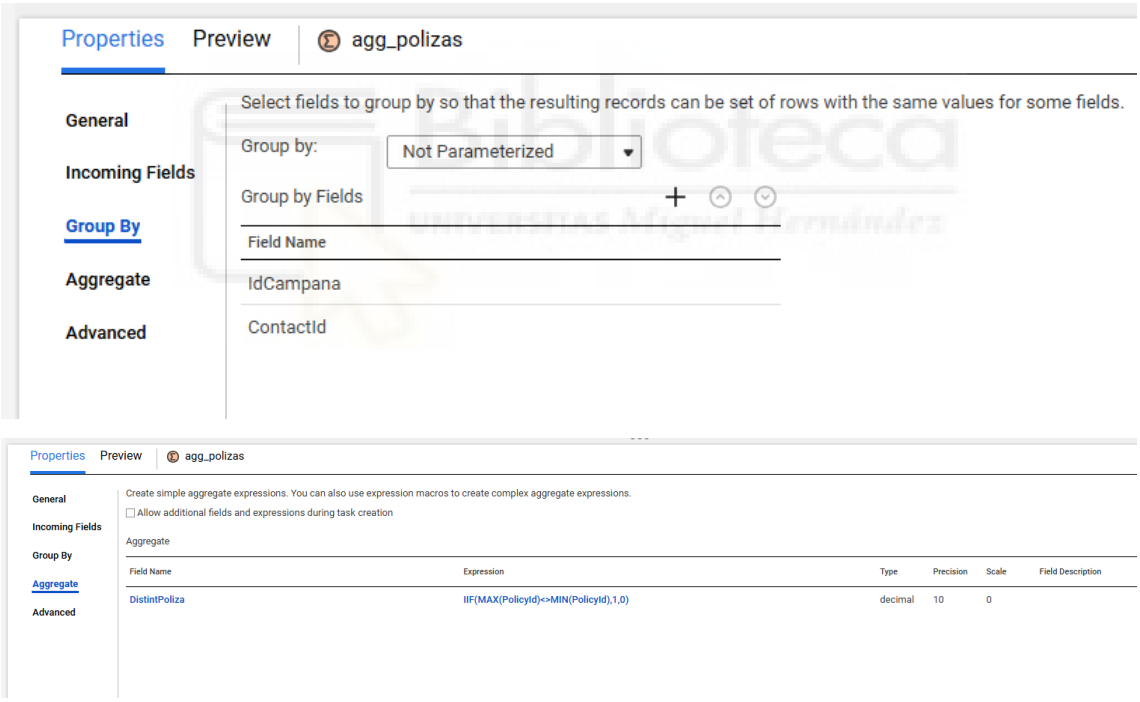


FIGURA 22: Detalles del aggregator en IICS

Joiner o expresión para marcar duplicados: Una vez calculado el campo que identifica la duplicidad (DistintPoliza), se aplica una transformación de tipo Filter que actúa como bifurcador lógico, separa los registros que contienen múltiples pólizas (valor igual a 1) del resto. Solo los casos marcados como duplicados continúan por una de las rutas, mientras que aquellos sin duplicidad toman otra dirección.

Este diseño facilita la personalización del tratamiento de datos, los registros únicos son procesados normalmente y enviados directamente a Salesforce, mientras que los duplicados siguen un camino distinto, donde se omite información sensible como PolicyId o Oferta_Retencion, para posteriormente ser almacenados en un archivo de salida alternativo.

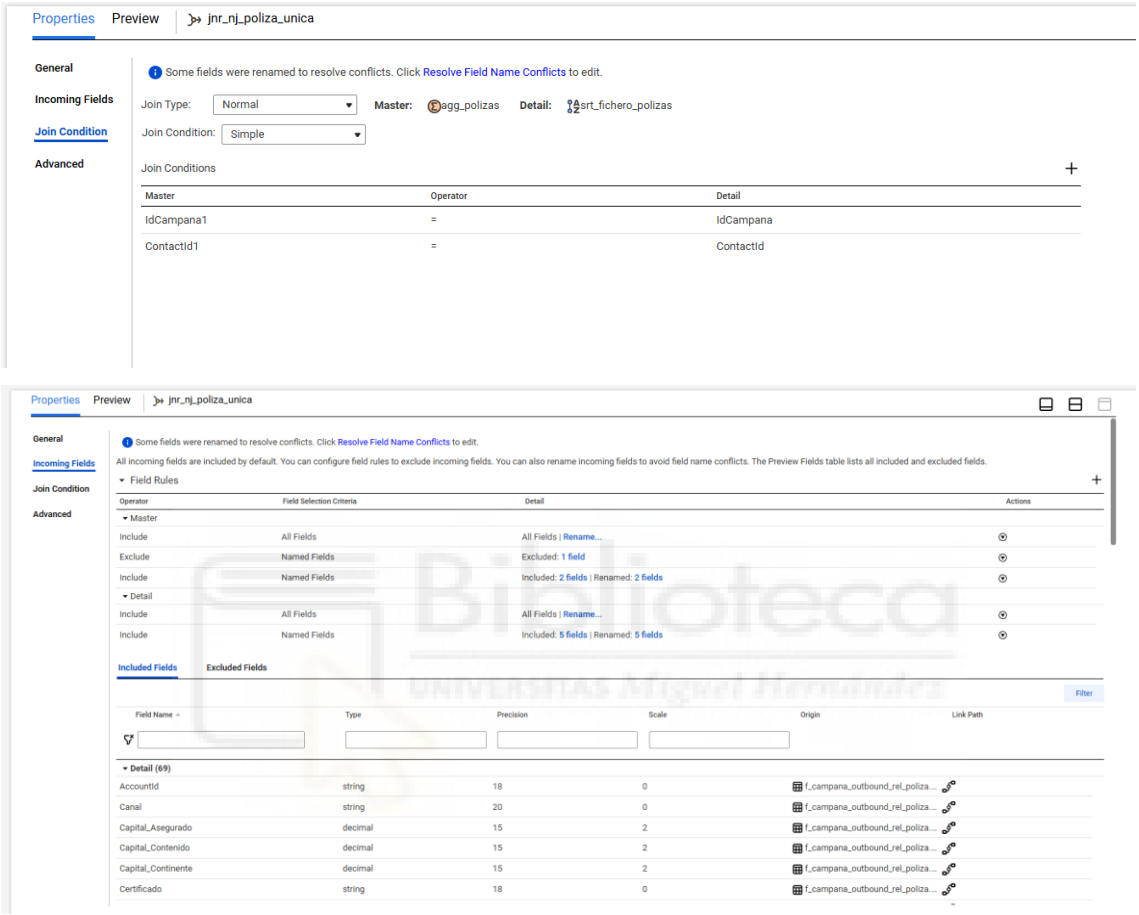


FIGURA 23: Detalles del joiner en IICS

Split del flujo: Imagen donde se visualiza claramente la bifurcación: una salida para Salesforce (con campos vacíos si hay duplicado), otra para fichero plano (registro completo).

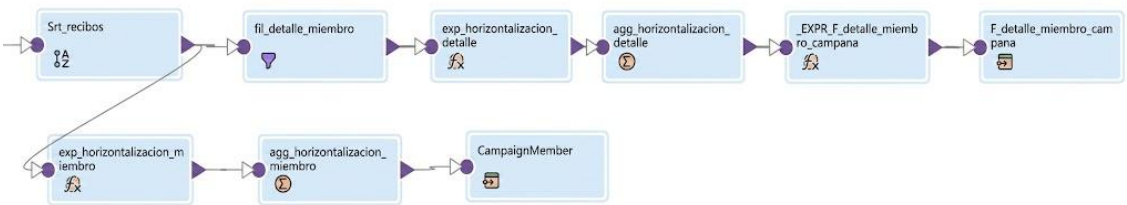


FIGURA 24: Vista del mapping a partir del cruce en IICS

Una vez calculado el campo que identifica la duplicidad (DistintPoliza), se aplica una transformación de tipo Filter que actúa como bifurcador lógico: separa los registros que contienen múltiples pólizas (valor igual a 1) del resto. Solo los casos marcados como duplicados continúan por una de las rutas, mientras que aquellos sin duplicidad toman otra dirección.

Este diseño facilita la personalización del tratamiento de datos. Los registros únicos son procesados normalmente y enviados directamente a Salesforce, mientras que los duplicados siguen un camino distinto, donde se omite información sensible como PolicyId o Oferta_Retencion, para posteriormente ser almacenados en un archivo de salida alternativo.

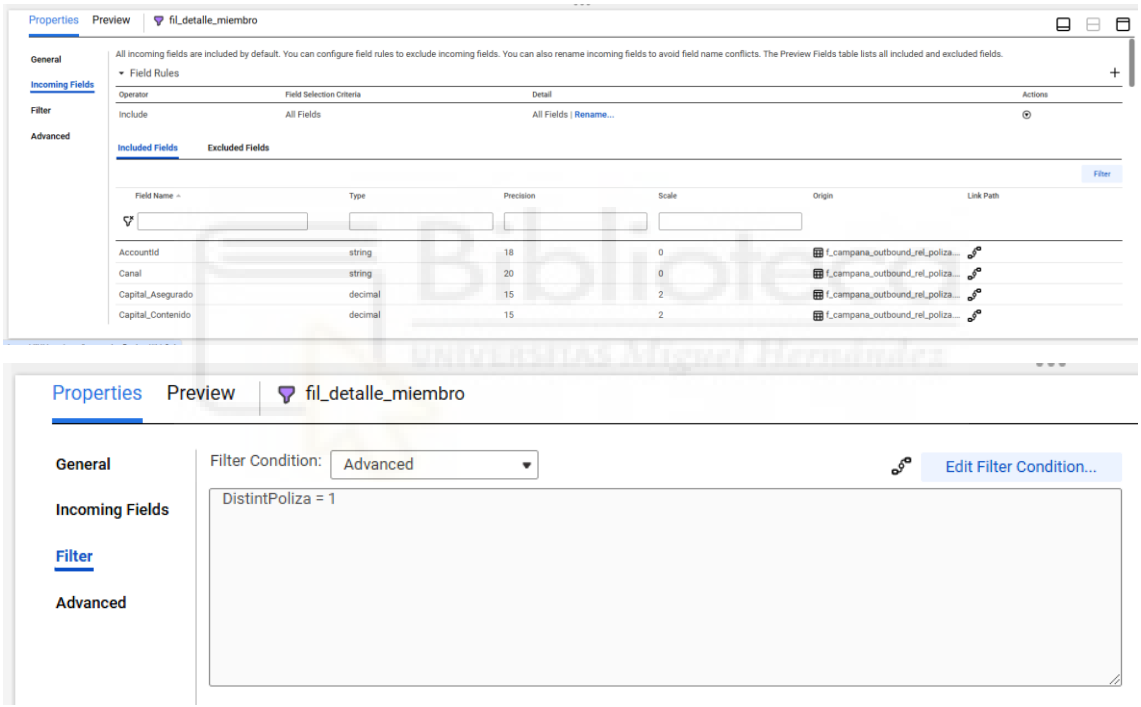


FIGURA 25: Detalles del filtro en IICS

Targets:

Captura final del diseño que muestra los destinos definidos y su relación con los flujos anteriores.

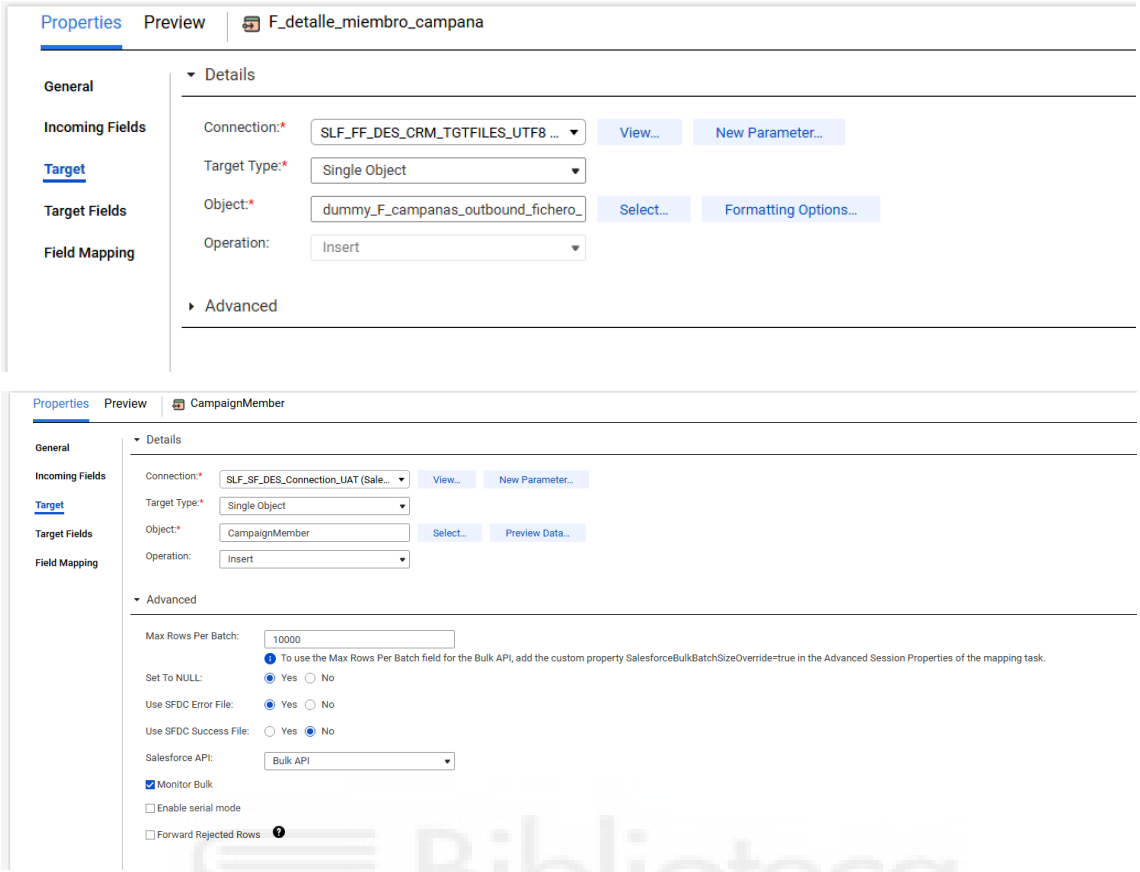


FIGURA 26: Target en IICS

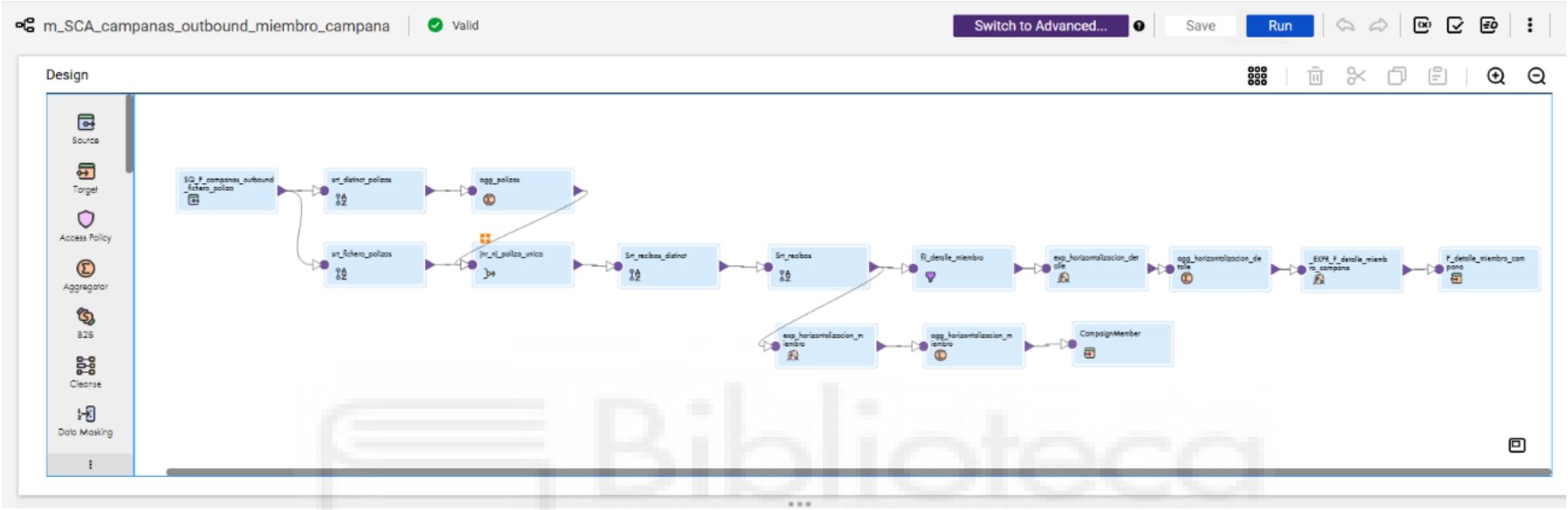


FIGURA 27: Vista del mapping completo en IICS

5.3 Especificaciones técnicas del entorno y documentos de soporte

Durante el desarrollo y ejecución de los procesos de integración de datos, se ha contado con dos plataformas tecnológicas bien diferenciadas, cada una con sus propias particularidades a nivel de sistema operativo, configuración y arquitectura subyacente.

En el entorno de PowerCenter, la infraestructura utilizada estaba basada en un servidor con sistema operativo Windows el cual ofrecía una integración directa con herramientas locales y facilitaba la gestión tradicional de procesos ETL, apoyado en una instalación on-premise de Informatica PowerCenter. La administración de los servicios, así como el despliegue de mappings y workflows, se realizaba directamente sobre este servidor, en un modelo de gestión centralizado basado en recursos internos. En contraposición, la migración a IICS supone un cambio arquitectónico considerable ya que está diseñado para funcionar en un entorno Linux. Este servidor actúa como intermediario entre sistemas cloud y locales (Secure Agent) permitiendo procesar y ejecutar tareas de forma remota. Este enfoque ofrece mayor flexibilidad y escalabilidad ya que gran parte de la lógica de orquestación se externaliza a los servicios cloud de Informatica. Además, el uso de un sistema operativo Linux aporta mayor estabilidad, menor consumo de recursos y mayor compatibilidad con entornos de integración continua y automatizada.

En cuanto a la documentación técnica de soporte, se han consultado manuales oficiales de Informatica para PowerCenter e IICS, así como guías de buenas prácticas de la compañía. También se han utilizado documentos internos de requisitos y flujos funcionales, los cuales han sido esenciales para comprender la lógica previa y su posterior adaptación al nuevo entorno. Estos documentos han sido el punto de referencia continuo para garantizar la fidelidad del proceso migrado y asegurar el cumplimiento de las reglas de negocio definidas.

6. BIBLIOGRAFÍA

[1] IBM, *What is ETL (extract, transform, load)?* Disponible en: https://www.ibm.com/think/topics/etl [Último acceso 22 de noviembre de 2025]

[2] IBM, *ELT vs. ETL: What's the Difference?* Disponible en: https://www.ibm.com/think/topics/elt-vs-etl [Último acceso 22 de noviembre de 2025]

[3] Informatica (2024), *PowerCenter Customer ALERT: Migrate Now — Informatica PowerCenter 10.4 Support Ends on March 31.* Disponible en: https://www.informatica.com/blogs/powercenter-customer-alert-migrate-now-informatica-powercenter-104-support-ends-on-march-31.html [Último acceso 22 de noviembre de 2025]

[4] Informatica (2020), Documentación PowerCenter 10.4. Disponible en: https://docs.informatica.com/es_es/data-integration/powercenter/10-4-0.html [Último acceso 22 de noviembre de 2025]

[5] Informatica, *Integración de aplicaciones de Informatica Cloud: Descripción de las funcionalidades.* Disponible en: https://www.informatica.com/content/dam/informatica-com/es/collateral/white-paper/cloud-application-integration_white-paper_3407es.pdf [Último acceso 22 de noviembre de 2025]

[6] Kashetty Sunag Dinesh & Soma Ghosh (2025). IEEE, *Evolution of ETL Tools: Trends and Insights from On-Premises to Cloud Solutions*. DOI:10.1109/ICSCDS65426.2025.11166785 [Último acceso 22 de noviembre de 2025]

[7] Vangipuram Radhakrishna, Vangipuram SravanKiran & K. Ravikiran (2012). IEEE, *Automating ETL Process with Scripting Technology*. DOI:10.1109/NUICONE.2012.6493217 [Último acceso 22 de noviembre de 2025]

[8] Vaishnavi Pawar, Shubham Kumawat, Madhuri Ahire, Rutika Musmade, Rohit R. Nikam & P. William (2023). IEEE, *ETL-Based Billing System for Azure Services with Cost Estimation.* DOI:10.1109/ICAISS58487.2023.10250628 [Último acceso 22 de noviembre de 2025]

[9] Aditya Gupta, Sana Zia Hassan, Gokul Narain Natarajan, Raghavender Reddy Tuniki & Satya Manesh Veerapaneni (2025). IEEE, *From Ingestion to Action: Architecting Intelligent DataOps Pipelines for Real-Time Consumer Systems.* DOI:10.1109/ICOCT64433.2025.11118485 [Último acceso 22 de noviembre de 2025]

[10] V.P. Khranilov, P.V. Misevich, P.S. Kulyasov & E.N. Pankratova (2024). IEEE, *The Agent Approach to the ETL Task.* DOI:10.1109/SUMMA64428.2024.10803737 [Último acceso 22 de noviembre de 2025]

[11] G. Sunil Santhosh Kumar & M. Rudra Kumar (2022). IEEE, *Dimensions of Automated ETL Management: A Contemporary Literature Review*. DOI:10.1109/ICACRS55517.2022.10029274 [Último acceso 22 de noviembre de 2025]

[12] Shubhada Labde, Nidhi Bhanushali, Rahul Thaker & Romit Singh (2022). IEEE, *A Project on Statistical Analysis, ETL Automation, and Intelligence Reporting*. DOI:10.1109/CONIT55038.2022.9847751 [Último acceso 22 de noviembre de 2025]

[13] Shivareddy Devarapalli, Venumadhav Goud Vathsavai, Venugopal Katkam & Rajesh Kumar Kanji (2025). IEEE, *Cloud-Native LLMOps Meets DataOps: A Unified Framework for High-Volume Analytical Systems.* DOI:10.1109/ICCTDC64446.2025.11158069 [Último acceso 22 de noviembre de 2025]

[14] Milan Szilveszter, György Molnár & Enikő Nagy (2024). IEEE, *Comparison of Commercial and Open Source ETL Tools*. DOI:10.1109/ICCC62278.2024.10582972 [Último acceso 22 de noviembre de 2025]

[15] Gokul Narain Natarajan, Shazia Hassan, Raghavender Reddy Tuniki & Sana Zia Hassan (2025). IEEE, *Cross-Domain DataOps with Federated Learning: Unlocking AI in Regulated and Consumer-Facing Systems.* DOI:10.1109/ICCTDC64446.2025.11158826 [Último acceso 22 de noviembre de 2025]

[16] Kristina Cormier, Keona Gagnier, Joshua Padron-Uy, Dolcy Sareen, Ajitesh Parihar, Youry Khmelevsky, Gaétan Hains & Albert Wong (2025). IEEE, *Data Extraction, Transformation, and Loading (ETL) Process Automation and Data Warehouse Implementation for Algorithmic Trading Machine Learning Modelling*. DOI:10.1109/SysCon64521.2025.11014812 [Último acceso 22 de noviembre de 2025]

[17] M. Vijayalakshmi & R.I. Minu (2022). *IEEE, Incremental Load Processing on ETL System through Cloud.* DOI:10.1109/ICONAT53423.2022.9726039 [Último acceso 22 de noviembre de 2025]

[18] Inan Gür, Frederik Möller, Marius Hupperz, Dilara Uzun & Boris Otto (2022). IEEE, *Requirements for DataOps to Foster Dynamic Capabilities in Organizations – A Mixed Methods Approach.* DOI:10.1109/CBI54897.2022.00025 [Último acceso 22 de noviembre de 2025]

[19] Zheng Yin, Shengwen Zhou, Jingjing Zhou, Minghui Tian, Musen Lin & Sida Liu (2023). IEEE, *Research on DataOps Capability: Practice and Development*. DOI:10.1109/TrustCom60117.2023.00303 [Último acceso 22 de noviembre de 2025]

[20] Ramesh Somayajula, Rakesh Ramakrishna Pai, Nirmal Sajanraj & Kushal Shah

(2025). IEEE, *Zero-ETL Architectures for AI Workloads: Direct ML Model Access on Cloud-Native OLAP Systems.* DOI:10.1109/ICOCT64433.2025.11118928 [Último acceso 22 de noviembre de 2025]

[21] Oracle, *Data Movement/ETL Overview*. Disponible en: https://docs.oracle.com/en/database/oracle/oracle-database/23/dwhsg/etl-overview.html#GUID-50E33D2C-6F47-4449-8E49-90E4D7974A08 [Último acceso 20 de noviembre]

[22] TDWI (2010), *Data Integration and Data Warehousing Defined*. Disponible en: https://tdwi.org/articles/2010/05/06/data-integration-and-data-warehousing-defined.aspx [Último acceso 27 de noviembre de 2025]

[23] Informatica (2023), *Configuring Resources*. Disponible en: https://docs.informatica.com/data-quality-and-governance/informatica-data-quality/10-4-0/application-service-guide/powercenter-integration-service/load-balancer-for-the-powercenter-integration-service/configuring-resources.html [Último acceso 22 de noviembre de 2025]

[24] Informatica (2023), *CDI-Elastic and Advanced Serverless.* Disponible en: https://www.informatica.com/content/dam/informatica-cxp/techtuesdays-slides-pdf/CDI-Elastic%20and%20Advanced%20Serverless.pdf [Último acceso 27 de noviembre de 2025]

[25] IICS Docs (2025), *Amazon Redshift Serverless connectivity.* Disponible en: https://onlinehelp.informatica.com/IICS/prod/CDI/en/index.htm?contextID=CDI_REDSHIFTV2_SERVERLESS_CONNECTIVITY [Último acceso 27 de noviembre de 2025]

[26] Talend *Talend Open Studio*. Disponible en: https://www.talend.com/products/talend-open-studio [Último acceso 28 de noviembre de 2025]

[27] Microsoft Learn (2025), *Introduction to Azure Data Factory*. Disponible en: https://learn.microsoft.com/en-us/azure/data-factory/introduction [Último acceso 28 de noviembre de 2025]

[28] AWS Docs (2025), *What is AWS Glue?.* Disponible en: https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html [Último acceso 28 de noviembre de 2025]

[29] Matillion Docs (2025), *Matillion ETL product overview*. Disponible en: https://docs.matillion.com/metl/docs/1975061/ [Último acceso 28 de noviembre de 2025]

[30] IBM, *Modern ETL: The Brainstem of Enterprise AI.* Disponible en: https://www.ibm.com/think/insights/modern-etl [Último acceso 29 de noviembre de 2025]

[31] Informatica Docs (2023), *CLAIRE Recommendations and Insights*. Disponible en: https://docs.informatica.com/data-quality-and-governance/informatica-data-quality/10-4-1/new-features-guide/new-features--10-4-0-/informatica-mappings/claire-recommendations-and-insights.html [Último acceso 29 de noviembre de 2025]

[32] DataOps Manifesto (2021). Disponible en: https://dataopsmanifesto.org/es/ [Último acceso 29 de noviembre de 2025]

[33] Informatica PowerCenter Docs (2022), *Preface*. Disponible en: https://docs.informatica.com/data-integration/powercenter/10-4-0/transformation-guide/preface.html [Último acceso 29 de noviembre de 2025]

[34] Informatica PowerCenter Docs (2019), *Functions*. Disponible en: https://docs.informatica.com/data-integration/powercenter/10-4-0/transformation-language-reference/functions/max--string-.html [Último acceso 29 de noviembre de 2025]

[35] Informatica Docs (2025), *Introduction to Salesforce Connector*. Disponible en: https://docs.informatica.com/integration-cloud/data-integration-connectors/current-version/salesforce-connector/introduction-to-salesforce-connector.html [Último acceso 29 de noviembre de 2025]

[36] Informatica Intelligent Cloud Services (2023), *Cost and Licensing*. Disponible en: https://docs.informatica.com/cloud-common-services/administrator/h2l/1382-how-to--install-and-configure-informatica-intelligent-cloud/install-and-configure-informatica-intelligent-cloud-services-in-/cost-and-licensing.html [Último acceso 1 de diciembre de 2025]

[37] Syntax Minds (2024), *Essential Difference Between Informatica PowerCenter and IICS*. Disponible en: https://syntaxminds.com/what-is-the-difference-between-informatica-pc-and-iics/ [Último acceso 1 de diciembre de 2025]

[38] DataTerrain (2025), *Informatica PowerCenter vs. Informatica Intelligent Cloud Services (IICS): Key Feature Differences*. Disponible en: https://dataterrain.com/informatica-powercenter-vs-iics-key-feature-differences [Último acceso 1 de diciembre de 2025]

[39] Panoply Blog (2021*), Top Data Integration Tools for 2021: A Cost-Benefit Analysis.*

Disponible en: https://blog.panoply.io/data-integration-tools [Último acceso 1 de diciembre de 2025]

[40] Vendr, *How much does Informatica cost?* Disponible en: https://www.vendr.com/marketplace/informatica [Último acceso 1 de diciembre de 2025]

[41] Adastra, *20 % Less TCO through Cloud Migration on Informatica Intelligent Cloud Platform*. Disponible en: https://adastracorp.com/success-stories/cloud-migration-on-informatica-intelligent-cloud-services [Último acceso 1 de diciembre de 2025]

[42] AWS Marketplace (2025), *Informatica Intelligent Data Management Cloud*. Disponible en: https://aws.amazon.com/marketplace/pp/prodview-vetdnosd4u6oe [Último acceso 1 de diciembre de 2025]

[43] Informatica (2025), *Snowflake Data Cloud Connector*. Disponible en: https://docs.informatica.com/integration-cloud/data-integration-connectors/current-version/snowflake-data-cloud-connector/preface.html [Último acceso 1 de diciembre de 2025]

[44] Informatica (2025), *Introduction to Databricks Connector*. Disponible en: https://docs.informatica.com/integration-cloud/data-integration-connectors/current-version/databricks-connector/introduction-to-databricks-connector.html [Último acceso 3 de diciembre de 2025]

[45] Informatica (2023), *Ecosystem - Azure*. Disponible en: https://docs.informatica.com/integration-cloud/data-integration-connectors/current-version.html?subsection=Ecosystem+%E2%80%93+Azure [Último acceso 3 de diciembre de 2025]

[46] Informatica (2025), *Streaming Ingestion and Replication.* Disponible en: https://docs.informatica.com/integration-cloud/data-ingestion-and-replication/current-version/streaming-ingestion-and-replication/streaming-ingestion-and-replication.html [Último acceso 3 de diciembre de 2025]

[47] Medium (2025), *Real-Time Data Ingestion in IDMC/IICS: How to Design It Right.* Disponible en: https://medium.com/%40adilshk047/real-time-data-ingestion-in-iics-how-to-design-it-right-f9458d0e86e8 [Último acceso 3 de diciembre de 2025]