

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



**“SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS
FINTECH MEDIANTE UN DATA WAREHOUSE ESCALABLE”**

RAÚL ANDRÉS MANUEL CUELLO OÑATE

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA**

Profesor Guía: Mauricio Figueroa Colarte
Profesor Correferente: Nelson Diaz

Julio - 2026

SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS FINTECH
MEDIANTE UN DATA WAREHOUSE ESCALABLE

**CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE
MONOGRAFÍA A REPOSITORIO ACADÉMICO**

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Solución a la dispersión de datos en entornos fintech mediante un Data Warehouse escalable

Nombre del candidato(a): Raúl Andrés Manuel Cuello Oñate

Carrera / Grado: Ingeniero Civil en Informática

Campus: Casa Central **Departamento:** Departamento de Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Mauricio Figueroa Colarte, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

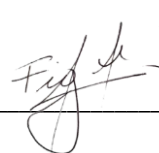
Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 17 de agosto de 2026

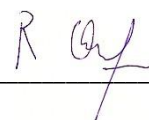
Firma: _____



Estudiante o Candidato(a):

Fecha: 14 de agosto de 2026

Firma: _____



Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

Para mi querida familia, mis amigos más cercanos, Florencia y su hermosa familia.

Papá, te prometo llevar el apellido Cuello lo más alto que pueda de forma honesta,
amable y soñadora.

“Ganar es un premio, darlo todo es obligación”.

AGRADECIMIENTOS

En primer lugar, agradezco a mis tíos, Juan Carlos Baeza y María Luisa Oñate, por subvencionarme esta carrera profesional con la que he logrado obtener conocimientos, herramientas, experiencias de vida y poder conectar con personas que me han fortalecido y apoyado durante todo este periodo. Gracias tíos por ayudarme a estar más cerca de mis sueños y enseñarme que la generosidad cambia vidas.

Agradezco a mi mamá, Leonor Oñate, por ayudarme y apoyarme sobre todo en este tiempo inmensamente agotador y estresante. Gracias mamá por hacerme el almuerzo cada día para que yo no tuviera que ocupar horas en ello y pudiese destinar ese tiempo en estudiar o descansar. Gracias por hacerme la cama cada día que iba a clases para que, al volver, pudiera descansar. Gracias por estar presente en las noches en las que venían ataques de ansiedad y encontraba consuelo en tu cariño. Sobre todo, gracias mamá por ser una increíble mamá.

Gracias Fabrizio Esteban Fernandini por ser el mejor hermano mayor. Quiero que sepas que valoro mucho ese trabajo silencioso que hiciste de apoyarme y fortalecerme, incluso, si no hablábamos por varias semanas. Gracias por estar para escucharme y debatir por horas sobre decisiones importantes y también pequeñas. Tu gran gesto de incentivar me a aprender cosas nuevas desde muy chico me ha guiado para encontrar mi vocación y pensar más allá de únicamente lo que enseñan los establecimientos.

Gracias a mi familia Oñate por siempre apoyarme incluso cuando hice falta por estar dedicado al deporte en mi etapa juvenil. Gracias Carla Baeza, familia Oñate-Tapia y Elias-Baeza por abrirme las puertas de sus respectivos hogares en momentos en los que lo necesité y brindarme de todas las comodidades para que no tuviese problemas de ningún tipo. Estaré eternamente agradecido con cada uno de ustedes.

Amigos, Cristóbal R., Tomás R., José T., Felipe S., gracias por cada palabra de aliento que tuvieron en mis momentos más bajos y las calurosas y eufóricas felicitaciones en cada uno de mis éxitos. “Hay amigos, hay familia y luego hay amigos que se convierten en familia”.

Gracias familia Figueroa-Valdivia por acogerme en su hogar y hacerme sentir parte desde el comienzo. Agradezco el apoyo que me han brindado a través de consejos, charlas reconfortantes y momentos felices que me han ayudado a sobrellevar malos ratos.

Por último, Florencia Silvia Figueroa Valdivia. Gracias por haber estado en todos los momentos en este camino que parecía eterno. Me has apoyado en los momentos difíciles, consolado en los peores y festejado los buenos, pero lo que más valoro es que sin importar lo que pasase siempre estuviste ahí para mí, con una sonrisa, un abrazo y ese cariño que tanto me hace sentir bien. Gracias por ayudarme a crecer y ser mejor persona, hemos compartido un tercio de nuestras vidas y ya no me imagino otra vida que no fuese a tu lado. Por último, gracias por tener paciencia debido a las numerosas semanas que no nos pudimos ver y días y fiestas que no pudimos compartir... por fin terminó la espera.

RESUMEN

Resumen—Las organizaciones del sector *fintech* operan con datos dispersos en múltiples fuentes heterogéneas, situación que genera inconsistencias en los reportes y dificulta la toma de decisiones. El presente trabajo aborda esta problemática en una *fintech* chilena del segmento *factoring* mediante el diseño e implementación de un *Data Warehouse* escalable en BigQuery que centraliza y estructura la información, materializado en una primera etapa sobre el dominio de las campañas de *marketing*. El desarrollo siguió la metodología HEFESTO, complementada con el modelado dimensional de Kimball, sobre una arquitectura por capas (*raw*, *staging*, *silver*, *gold* y semántica) desplegada en un *stack multinube* (AWS y Google Cloud Platform) bajo un enfoque *ELT*, e integró la base de datos operacional y la plataforma externa Brevo a través de procesos automatizados de extracción, transformación y carga. La validación, apoyada en el modelo de éxito de DeLone y McLean, se acotó al dominio de campañas de *marketing* evidenció para este caso una reducción del tiempo de consolidación de tres horas a veintinueve segundos, la eliminación de las inconsistencias entre reportes y un incremento de dos a diez indicadores disponibles en modalidad de autoservicio. Los resultados demuestran que la solución transforma un escenario de consolidación manual en un repositorio centralizado, automatizado y documentado.

Palabras Clave— *Data Warehouse*; inteligencia de negocios; modelado dimensional; *fintech*; *ELT*

ABSTRACT

Abstract— Organizations in the fintech sector operate with data scattered across multiple heterogeneous sources, a situation that generates inconsistencies in reports and hinders decision-making. This work addresses this problem in a Chilean fintech company from the factoring segment through the design and implementation of a scalable Data Warehouse in BigQuery that centralizes and structures information, materialized in a first stage on the marketing campaigns domain. The development followed the HEFESTO methodology, complemented with Kimball's dimensional modeling, over a layered architecture (*raw*, *staging*, *silver*, *gold* and semantic) deployed on a multi-cloud stack (AWS and Google Cloud Platform) under an *ELT* approach, and integrated the operational database and the external platform Brevo through automated extraction, transformation and loading processes. The validation, supported by the DeLone and McLean information systems success model, was scoped to the marketing campaigns domain and showed for this case a reduction in consolidation time from three hours to twenty-nine seconds, the elimination of inconsistencies between reports, and an increase from two to ten indicators available in self-service mode. The results demonstrate that the solution transforms a manual consolidation scenario into a centralized, automated and documented repository.

Keywords— Data Warehouse; Business Intelligence; dimensional modeling; fintech; *ELT*.

GLOSARIO

API (Application Programming Interface / Interfaz de programación de aplicaciones): Conjunto de reglas y especificaciones que permite que distintas aplicaciones de software se comuniquen e intercambien datos.

BI (Business Intelligence / Inteligencia de negocios): Conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar datos en información para apoyar la toma de decisiones.

CDC (Change Data Capture / Captura de cambios): Técnica que permite identificar y registrar inserciones, actualizaciones y eliminaciones realizadas sobre una fuente de datos.

Dataform: Herramienta de transformación y modelado de datos de Google Cloud que permite construir, documentar y automatizar pipelines SQL sobre BigQuery.

DW (Data Warehouse / Almacén de datos): Repositorio orientado a temas, integrado, no volátil y variable en el tiempo, diseñado para soportar el análisis y la toma de decisiones.

DWH (Data Warehousing): Proceso mediante el cual se recopilan, transforman y gestionan datos de diversas fuentes para construir y alimentar un *Data Warehouse*.

E2E (End-to-End / De principio a fin): Proceso que abarca todas las etapas, desde la extracción de los datos hasta su explotación final.

ELT (Extract, Load, Transform / Extracción, carga y transformación): Variante del *ETL* en la que los datos se cargan primero en el repositorio y se transforman dentro de él.

ETL (Extract, Transform, Load / Extracción, transformación y carga): Proceso de extraer datos de distintas fuentes, transformarlos a un formato consistente y cargarlos en un repositorio analítico.

GCP (Google Cloud Platform): Plataforma de servicios en la nube de Google que proporciona infraestructura, almacenamiento, procesamiento y análisis de datos.

KPI (Key Performance Indicator / Indicador clave de desempeño): Métrica utilizada para medir el rendimiento de un proceso o actividad del negocio.

MPP (Massively Parallel Processing / Procesamiento masivamente paralelo): Arquitectura informática que divide tareas complejas en múltiples procesos para ejecutarlas de forma paralela y eficiente.

OLAP (On-Line Analytical Processing / Procesamiento analítico en línea): Paradigma orientado a las consultas analíticas sobre grandes volúmenes de datos de forma multidimensional.

OLTP (On-Line Transaction Processing / Procesamiento transaccional en línea): Paradigma orientado al registro rápido y concurrente de las transacciones operativas del negocio.

SaaS (Software as a Service / Software como servicio): Modelo de distribución de software en el que el proveedor aloja la aplicación y la ofrece como un servicio a través de Internet.

SCD (Slowly Changing Dimensions / Dimensiones lentamente cambiantes): Técnica de modelado utilizada para gestionar cambios históricos en atributos de las dimensiones.

Schema (Esquema): Estructura lógica que define cómo se organizan y relacionan los datos dentro de una base de datos.

SII (Servicio de Impuestos Internos): Organismo público chileno encargado de la administración del sistema tributario.

Stakeholder: Cualquier individuo, grupo u organización que tiene un interés en una empresa, proyecto o decisión corporativa

SQL (Structured Query Language / Lenguaje de consulta estructurado): Lenguaje estándar utilizado para consultar, manipular y gestionar bases de datos relacionales.

UTFSM (Universidad Técnica Federico Santa María): Institución de educación superior donde se desarrolla este trabajo de título.

INDICE DE CONTENIDOS

AGRADECIMIENTOS.....	4
RESUMEN.....	5
ABSTRACT	5
GLOSARIO	6
INDICE DE CONTENIDOS	8
INDICE DE FIGURAS	11
INDICE DE TABLAS	13
INTRODUCCIÓN.....	14
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	16
1.1. Contexto.....	16
1.2. Antecedentes y soluciones en organizaciones similares	17
1.3. Objetivos y alcances	17
1.3.1. Objetivo general	17
1.3.2. Objetivos específicos	18
1.3.3. Alcance.....	18
CAPÍTULO 2: MARCO CONCEPTUAL.....	19
2.1. <i>Business Intelligence (BI)</i>	19
2.1.1. Paradigmas de procesamiento de datos	20
2.1.2. Sistema computacional de inteligencia de negocios.....	21
2.2. <i>Data Warehousing</i>	23
2.2.1 Definición	23
2.2.2. Obtención de datos (<i>ETL/ELT</i>)	24
2.2.3. Almacenamiento (<i>Data Warehouse</i>)	32
2.2.4. Análisis	40
2.2.5. Metodologías para el desarrollo de un Data Warehouse	44

2.2.6. Modelos de éxito	49
2.3. Datos e inteligencia de negocios en el sector FINTECH	50
2.3.1. Definición y CRECIMIENTO del <i>Fintech</i>	50
2.3.2. Data Warehousing en Fintech	50
CAPÍTULO 3: PROPUESTA DE SOLUCION.....	52
3.1. Visión general de la solución.....	52
3.2. Metodología de desarrollo aplicada	53
3.3. Análisis de requerimientos.....	53
3.3.1. Preguntas de negocio	54
3.3.2. Indicadores	54
3.3.3. Modelo conceptual de requerimientos.....	57
3.4. Fuentes de datos	57
3.5. Modelo lógico del Data Warehouse.....	60
3.6. Arquitectura de la solución	62
3.6.1. Arquitectura por capas	62
3.6.2. <i>Stack</i> tecnológico.....	63
3.7. Integración de datos	64
3.7.1. Ingesta de datos.....	64
3.7.2. Transformación con Dataform	65
3.7.3. Automatización, frecuencias y monitoreo	65
3.8. Análisis y visualización.....	66
3.9. Documentación de la solución	68
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	72
4.1. Criterios de validación.....	72
4.2. Validación técnica del sistema	73
4.2.1. Confiabilidad de los procesos automatizados	73

SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS FINTECH
MEDIANTE UN DATA WAREHOUSE ESCALABLE

4.2.2 Calidad de los datos	74
4.3. Validación funcional	75
4.3.1. Cobertura de las preguntas de negocio.....	75
4.3.2. Validación del <i>dashboard</i>	77
4.4. Comparativa antes y después	78
4.5 Síntesis según el modelo de DeLone & Mclean	80
CAPÍTULO 5: CONCLUSIONES	82
5.1 Cumplimiento de los objetivos.....	82
5.2 Resultados y aportes a la organización	83
5.3 Discusión y aprendizaje	84
5.3.1. Sobre la elección metodológica: Kimball VS Inmon	84
5.3.2. Sobre el análisis de requerimientos en Data Marts acotados.....	84
5.3.3. Sobre el rol de las claves subrogadas	85
5.3.4. Sobre la documentación de un Data Warehouse.....	85
5.4. Limitaciones.....	86
5.5. Trabajo futuro	86
5.6. Reflexión final.....	87
REFERENCIAS BIBLIOGRÁFICAS	88
ANEXOS	91
ANEXO A	91
ANEXO B	94

INDICE DE FIGURAS

Ilustración 1: Pirámide de procesos <i>BI</i>	19
Ilustración 2: Esquema convencional <i>BI</i>	22
Ilustración 3: Representación de la diferencia entre <i>DWH</i> y <i>DW</i>	23
Ilustración 4: Ejemplo de datos semiestructurados	25
Ilustración 5: Diferencias entre <i>ETL</i> y <i>ELT</i>	29
Ilustración 6: Ejemplo de esquema estrella (<i>star schema</i>)	35
Ilustración 7: Ejemplo de esquema copo de nieve (<i>Snowflake schema</i>)	36
Ilustración 8: Ejemplo de esquema constelación (<i>Constellation schema</i>)	36
Ilustración 9: Esquema de patrón arquitectónico medallón	37
Ilustración 10: Representación de cubo <i>OLAP</i>	40
Ilustración 11: Ejemplo <i>dashboard</i> en Data Studio	43
Ilustración 12: Diagrama de <i>Kimball Lifecycle</i>	45
Ilustración 13: Diagrama de arquitectura de Data Warehousing propuesta	52
Ilustración 14: Modelo conceptual de requerimientos	57
Ilustración 15: Modelo conceptual multidimensional	59
Ilustración 16: Modelo lógico de <i>Data Mart marketing</i>	60
Ilustración 17: Extracto de <i>dashboard</i> interactivo	67
Ilustración 18: Antecedente de convenciones de transformación	68
Ilustración 19: Grafo automático de flujo de datos	68
Ilustración 20: Extracto de documentación de catálogo de transformación	69
Ilustración 21: Extracto de documentación de diccionario de datos	69
Ilustración 22: Consultas frecuentes	70
Ilustración 23: Definición de KPI por pilar y subtipo	91
Ilustración 24: Antecedente de las capas	92
Ilustración 25: Antecedente de transformación en Dataform	92

Ilustración 26: Muestra de ejecuciones automáticas de transformación en Dataform	93
Ilustración 27: Muestra de ejecuciones automáticas de extracción y carga en AWS Glue .	93
Ilustración 28: Extracto de presentación de proyecto: Problema de negocio.....	94
Ilustración 29: Extracto de presentación de proyecto: Caso práctico	94
Ilustración 30: Vista resumen de la campaña N°629	94
Ilustración 31: Vista de listas de contactos incluídas en la campaña.....	95
Ilustración 32: Extracto de vista del detalle de la lista de contactos N°308 asociado a la campaña N°629 que indica el total de 661 contactos.....	95
Ilustración 33: Desglose de entregabilidad por lista de contactos	95
Ilustración 34: Desglose de clics por lista de contactos	95
Ilustración 35: Extracto de resultado de consulta del resumen de la campaña N°629 en formato JSON.....	96

INDICE DE TABLAS

Tabla 1: Comparación entre los paradigmas <i>OLTP</i> y <i>OLAP</i>	21
Tabla 2: Ejemplo de datos estructurados.....	24
Tabla 3: Comparación entre los enfoques <i>top-down</i> y <i>bottom-up</i>	33
Tabla 4: Ejemplo de tabla de hechos de desempeño de campañas	34
Tabla 5: Ejemplo de tabla de dimensión Campaña	34
Tabla 6: Comparación de metodologías de desarrollo de un Data Warehouse	48
Tabla 7: Mapeo de los indicadores según fuente(s) de origen	58
Tabla 8: Stack tecnológico de la solución	63
Tabla 9: Procesos de integración y sus frecuencias	65
Tabla 10: Componentes de la documentación de la solución.....	71
Tabla 11: Mapeo de objetivos específicos	72
Tabla 12: Registro de ejecuciones automáticas del flujo de extracción y carga (16–20 de julio de 2026).....	73
Tabla 13: Registro de ejecuciones automáticas del flujo de transformación (16–20 de julio de 2026).....	74
Tabla 14: Reglas de calidad aplicadas mediante aserciones de Dataform.....	74
Tabla 15: Reglas preventivas de calidad aplicadas en la capa de <i>staging</i>	75
Tabla 16: Verificación de respuestas a preguntas de negocio	76
Tabla 17: Comparativa del antes y el después de la implementación del DW	78
Tabla 18: Inconsistencias de la campaña N.º 629 según la vista consultada en Brevo	79
Tabla 19: Síntesis de la validación según el modelo de Delone & Mclean	80

INTRODUCCIÓN

En la actualidad, la información se ha consolidado como uno de los activos más valiosos para las organizaciones donde la velocidad y precisión en la toma de decisiones constituyen factores determinantes de competitividad. Las empresas modernas generan y consumen volúmenes de datos cada vez más grandes desde distintos sitios como bases de datos transaccionales, planillas colaborativas, plataformas SaaS¹, entre otros, por lo que se enfrentan al desafío de integrar y explotar dichas fuentes de datos que se encuentran dispersas, heterogéneas y en constante crecimiento.

Uno de los sectores más sensibles a esta problemática son aquellas organizaciones del segmento *fintech* que ofrecen servicios financieros apoyados intensivamente en tecnología. Su operación depende de integrar datos de productos, clientes, marketing y operaciones comerciales, como también fuentes externas como, por ejemplo, el Servicio de Impuestos Internos (SII), plataformas de marketing digital y herramientas de gestión de personas.

El presente trabajo se desarrolla en una empresa *fintech* chilena del segmento *factoring*. La compañía ha financiado cientos de miles de millones de pesos en facturas, opera con miles de empresas y respalda su trayectoria con decenas de miles de operaciones realizadas. Pese a este posicionamiento, la organización enfrenta dificultades estructurales en el manejo de sus datos. Conviven fuentes obsoletas con réplicas más recientes que, en ocasiones, contienen columnas con el mismo nombre pero distinto significado, duplicación de atributos o tablas lo que, en algunos casos, genera inconsistencias en reportes.

Para abordar estas problemáticas, se propone construir un *Data Warehouse (DW)*. Esto se presenta como una solución tecnológica y estratégica ya que permite centralizar, estructurar y transformar datos de distintas fuentes en un repositorio único, facilitando el análisis mediante herramientas de inteligencia de negocios. Su implementación no solo fortalece la trazabilidad y calidad de los datos, sino que también habilita la automatización de reportes, el monitoreo de métricas² y el desarrollo de modelos predictivos que agregan valor a los procesos organizacionales.

La metodología aplicada para desarrollar este proyecto se basa en una estructura de iniciativas de *business intelligence (BI)*, que incluye la recopilación de datos, el diseño de un modelo de datos robusto, la implementación del modelo de datos, la automatización del proceso *ETL/ELT (Extract, Transform, Load)* y la creación de *dashboards*³ interactivos para presentar información de manera clara y accesible.

¹ El software como servicio (*Software as a Service* o SaaS) es un modelo de entrega de software basado en la nube en el que los proveedores alojan aplicaciones y las ponen a disposición de los usuarios a través de Internet.

² Las métricas son medidas cuantitativas utilizadas para evaluar el rendimiento, progreso o éxito de procesos, estrategias o sistemas.

³ Herramienta visual que consolida y presenta de manera resumida los datos y métricas más relevantes de un proceso, área o negocio, facilitando el seguimiento del desempeño y la toma de decisiones basada en datos.

El principal desafío en esta primera etapa es visibilizar el rendimiento y desempeño de campañas de *marketing* para la empresa involucrada. Esta problemática abarca fuentes internas de operaciones y clientes, así como una fuente externa correspondiente a la plataforma digital desde la cual se ejecutan las campañas.

Si bien este proyecto de título tiene como desafío construir una arquitectura que permita automatizar reportes y monitoreo sobre el rendimiento y desempeño de campañas de *marketing*, es crucial que la implementación permita, posteriormente, facilitar el mantenimiento y escalabilidad para más áreas o procesos de la compañía.

El presente documento se organiza en cinco capítulos, además de las referencias bibliográficas y anexos.

En el Capítulo 1 se presenta la definición del problema, describiendo el contexto organizacional en el que se desarrolla el trabajo, los antecedentes de soluciones similares y los objetivos y alcances que delimitan la propuesta.

En el Capítulo 2 se expone el marco conceptual que sustenta el desarrollo de la solución, abordando los fundamentos de *Business Intelligence*, *Data Warehousing*, procesos *ETL/ELT*, modelos de datos y arquitecturas aplicables al sector *fintech*.

En el Capítulo 3 se describe la propuesta de solución, incluyendo la metodología de desarrollo aplicada, el análisis de requerimientos, las fuentes de datos involucradas, el diseño del modelo lógico, la arquitectura tecnológica, la integración de datos y los mecanismos de análisis y visualización implementados.

En el Capítulo 4 se presenta la validación de la solución, evaluando el cumplimiento de los objetivos planteados y analizando los resultados obtenidos a partir de su implementación.

Finalmente, en el Capítulo 5 se desarrollan las conclusiones del trabajo, donde se sintetizan los principales aportes, limitaciones y posibles líneas de trabajo futuro.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1. CONTEXTO

Este trabajo de título se desarrolla en una *fintech* chilena perteneciente al segmento del *factoring* que, por motivos de confidencialidad y resguardo de información sensible, no será identificada explícitamente. En adelante, se hará referencia a ella como "**Empresa Alfa**". En los últimos años la empresa Alfa ha financiado cientos de miles de millones de pesos, opera con miles de empresas y respalda su trayectoria con decenas de miles de operaciones realizadas.

El *factoring* o factoraje es una alternativa de financiamiento en la que una empresa traspasa el servicio de cobranza futura de los créditos y facturas a un tercero a cambio de un anticipo de fondos descontado a una tasa pactada, lo que le permite mejorar su flujo de caja sin recurrir a deuda bancaria. Por ejemplo, si la empresa XYZ ha hecho una venta y por ello tiene una factura para ser cancelada a 30 días por un valor de \$1.000.000, dicha factura se cede a la empresa de *factoring* y esta última le entrega de inmediato un monto de dinero equivalente al 90% de la factura. Cuando se cumpla el plazo de 30 días, la empresa de *factoring* cobrará la factura original y obtendrá el monto total (CMF, 2026).

La empresa Alfa sustenta su operación mediante múltiples sistemas de información que han sido incorporados de manera progresiva para responder a distintas necesidades del negocio. Entre ellos se encuentran bases de datos transaccionales que registran operaciones de *factoring*, información de clientes y actividades comerciales, además de planillas colaborativas y plataformas externas tipo *SaaS* utilizadas por distintas áreas de la organización. Como consecuencia, la información relevante para la gestión se encuentra distribuida en diversas fuentes con estructuras, formatos y mecanismos de acceso diferentes, dificultando su integración y análisis de manera constante.

Esta situación empeora aún más por la coexistencia de fuentes con versiones antiguas y otras más recientes, las cuales presentan diferencias en su estructura, nomenclatura y significado de los atributos almacenados. En algunos casos, una misma información puede encontrarse replicada en múltiples sistemas o representada de forma distinta según el área que la utilice. La ausencia de una visión unificada sobre el origen, transformación y uso de los datos genera dificultades para determinar cuál es la información válida y confiable, incrementando el riesgo de inconsistencias en reportes, análisis y procesos de toma de decisiones.

Los principales afectados por esta problemática son los usuarios de nivel gerencial y el equipo de datos de la organización. Por una parte, los gerentes y líderes de área requieren información confiable y consistente para respaldar decisiones estratégicas y operacionales, sin embargo, las discrepancias entre distintas fuentes pueden generar interpretaciones divergentes sobre el desempeño del negocio. Por otra parte, el área de *Data* enfrenta dificultades asociadas al entendimiento y administración de los datos corporativos, debido a la limitada documentación sobre los sistemas existentes, los flujos de información y las

dependencias entre fuentes. Esta situación aumenta la complejidad de las tareas de análisis, mantenimiento e incorporación de nuevos integrantes al equipo, elevando los tiempos de aprendizaje y el riesgo de errores en el uso de la información.

1.2. ANTECEDENTES Y SOLUCIONES EN ORGANIZACIONES SIMILARES

El problema de la dispersión de datos y de la dificultad para medir el desempeño no es exclusivo de la empresa Alfa. Este ha sido abordado en otras organizaciones mediante la integración y consolidación de datos en repositorios analíticos. La literatura sobre el sector *fintech* señala que la consolidación de datos en un *Data Warehouse* se asocia a beneficios como mayor eficiencia operacional, mejor calidad y consistencia de la información y mayor escalabilidad (Oktaviani & Mauritsius, 2024).

Existen distintos enfoques para enfrentar este tipo de problemas. Una alternativa son las soluciones puntuales de reportería integradas en cada plataforma, sin embargo, un gran número de estas solo describen lo que ocurre dentro de esa fuente y no permiten cruzar la información con sistemas distintos. Otra alternativa es continuar con la consolidación manual en plantillas lo que es flexible pero no escalable por lo que es propensa a errores y carece de trazabilidad. En el otro extremo se encuentran enfoques de mayor madurez, como los *data lakes* que requieren niveles de capacidades organizacionales y tecnológicas superiores a los disponibles actualmente en la empresa Alfa.

La revisión de estos antecedentes permite delimitar las soluciones existentes que resultan insuficientes para el escenario de la empresa Alfa y orienta la propuesta que se desarrolla en el capítulo 3.

1.3. OBJETIVOS Y ALCANCES

A partir del problema definido, esta sección precisa el objetivo general que orienta el trabajo. Luego se presentan los objetivos específicos, que lo descomponen en resultados verificables, y el alcance, que delimita lo abordado en esta etapa. Cabe señalar que, si bien el objetivo general se formula sobre el *Data Warehouse* en su conjunto, su primera materialización y validación se realiza sobre el dominio de las campañas de *marketing*, sin perjuicio de que la arquitectura se diseñe para escalar a otras áreas de la compañía.

1.3.1. OBJETIVO GENERAL

El objetivo principal del trabajo de título es diseñar e implementar un *Data Warehouse* escalable en *Bigquery*⁴ que centralice y estructure datos provenientes de múltiples fuentes internas y externas, de modo de habilitar reportería automatizada y dejar el sistema preparado para análisis avanzados y modelos predictivos en etapas futuras.

⁴ Servicio de almacenamiento y análisis de datos en la nube desarrollado por Google.

1.3.2. OBJETIVOS ESPECÍFICOS

1. Diseñar una arquitectura modular basada en capas que permita escalar el sistema y asegurar la trazabilidad y calidad de los datos.
2. Automatizar procesos de extracción, transformación y carga de datos, permitiendo definir frecuencias personalizadas por fuente, monitorear ejecuciones y detectar errores sin intervención manual.
3. Documentar la estructura del *Data Warehouse*, incluyendo flujos de datos, definiciones de tablas, reglas de transformación y un repositorio de consultas frecuentes.

1.3.3. ALCANCE

El alcance en esta etapa comprende el diseño e implementación de la arquitectura por capas del *Data Warehouse* y la automatización de los procesos *ETL/ELT* para un subconjunto priorizado de fuentes, correspondiente al dominio de las campañas de *marketing*. Incluye, además, la elaboración de la documentación técnica asociada, la construcción de *dashboards* y su validación. La arquitectura se concibe desde su diseño para escalar a nuevas fuentes y áreas de la compañía a futuro, en coherencia con ello, esta etapa entrega la infraestructura que habilita el análisis avanzado, pero no su explotación predictiva. Quedan fuera del alcance la incorporación de la totalidad de las fuentes de la organización, el entrenamiento y la puesta en producción de modelos predictivos, y el rediseño de los sistemas operacionales que producen los datos.

CAPÍTULO 2: MARCO CONCEPTUAL

En este capítulo se establecen los fundamentos teóricos y técnicos que sustentan el diseño e implementación del *Data Warehouse* propuesto. Se parte por el marco general de la inteligencia de negocios (*Business Intelligence*), que sitúa el problema dentro de su disciplina. A continuación, se aborda el *Data Warehousing*, recorriendo las tres fases de la arquitectura convencional, esto es, la obtención de los datos mediante procesos ETL y ELT, su almacenamiento en el *Data Warehouse*, junto con el modelado, la arquitectura por capas y las tecnologías asociadas, y, por último, su análisis. Luego se revisan las metodologías para el desarrollo de un *Data Warehouse* y los modelos de éxito que permiten evaluar la solución. Finalmente, el capítulo cierra aplicando estos fundamentos al sector *fintech*, ámbito en el que se inscribe el presente trabajo.

2.1. BUSINESS INTELLIGENCE (BI)

Antes de hablar sobre *Data Warehousing*, primero se debe entender en qué marco del conocimiento técnico se encuentra.

La inteligencia de negocios (*business intelligence*) apoya a la toma de decisiones del negocio. Según varios autores se define “*Business Intelligence* es el proceso de convertir datos en conocimiento, y el conocimiento en acción, para la toma de decisiones” (Silva Peñafiel, Zapata Yáñez, Morales Guamán, & Toaquiza Padilla, 2019)



Ilustración 1: Pirámide de procesos BI

Fuente: Tomado de <https://nexusintegra.io/es/business-intelligence-industria-inteligencia-negocio/>

Para efectos de este trabajo, *Business Intelligence* se define desde un punto de vista pragmático como “El conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar datos de los sistemas transaccionales e información desestructurada (interna y externa de la compañía) en información estructurada, para su explotación directa (reportes, análisis *OLTP*⁵/*OLAP*⁶, alertas) o para su análisis y conversión

⁵ *OLTP (On-Line Transactional Processing)*: Sistemas computacionales orientadas al proceso de operaciones diarias de un negocio.

⁶ *OLAP (On-Line Analysis Proccesing)*: Sistemas computacionales orientados al proceso analítico de un negocio desde registros históricos.

en conocimiento dando así soporte a la toma de decisiones sobre el negocio” (Sinnexus, 2007).

2.1.1. PARADIGMAS DE PROCESAMIENTO DE DATOS

La definición anterior menciona el análisis *OLTP* y *OLAP* como parte de la explotación de los datos. Estos términos corresponden a dos paradigmas de procesamiento que conviene detallar ya que reflejan los dos roles que cumplen los datos en una organización. La primera corresponde al registro operacional de las transacciones del negocio y la segunda al soporte del análisis para la toma de decisiones. Ambas funciones imponen requisitos contrapuestos sobre los sistemas que las soportan, ya que los sistemas operacionales se orientan en la captura y actualización de los datos, mientras que los sistemas analíticos lo hacen a su consulta y explotación (Kimball & Ross, 2013). De esta distinción surgen dos paradigmas de procesamiento de datos, denominados procesamiento transaccional en línea (*OLTP*) y procesamiento analítico en línea (*OLAP*).

- ***On-Line Transaction Processing (OLTP)***

Corresponde a las bases de datos operacionales, orientadas a la transacción. Están optimizadas para procesar grandes volúmenes de datos transaccionales de forma rápida y concurrente, operando normalmente sobre un registro a la vez. Para evitar anomalías de actualización, estas bases se encuentran altamente normalizadas, lo que las hace eficientes para operaciones diarias, pero de bajo rendimiento al ejecutar consultas complejas que requieren unir muchas tablas o *sumarizar*⁷ grandes volúmenes de datos. Además, suelen contener solo datos detallados y actuales sin mantener información histórica (Vaisman & Zimányi, 2014).

- ***On-Line Analytical Processing (OLAP)***

Para cubrir las carencias de *OLTP* surge el procesamiento analítico en línea (*OLAP*), un paradigma orientado a las consultas analíticas. Las bases *OLAP* deben soportar cargas de consultas pesadas que típicamente involucran *sumarizaciones* y, por lo tanto, recorrer la mayor parte de los registros de una tabla. En este escenario, la *normalización*⁸ resulta contraproducente, ya que fragmenta los datos en numerosas tablas y obliga a un alto número de *joins*⁹ para reconstruir la información. Esta necesidad de un modelo distinto fue precisamente la que dio origen a los *Data Warehouse*, que están diseñados y optimizados para responder eficientemente consultas *OLAP* sobre un modelo multidimensional (Vaisman & Zimányi, 2014).

Aunque ambos paradigmas operan sobre los datos de la organización, responden a objetivos distintos y, en consecuencia, presentan diferencias sustanciales en su diseño y funcionamiento. La Tabla 1 sintetiza estas diferencias según los criterios más relevantes.

⁷ La agregación o “*sumarización*” es el proceso de consolidar datos detallados en valores resumidos mediante operaciones como suma, promedio, conteo, máximo o mínimo, entre otros. El fin es representar información con un mayor nivel de granularidad.

⁸ La normalización es un proceso que se usa en el diseño de bases de datos para organizar los datos de manera eficiente. Puede ayudar a reducir la redundancia de datos (datos duplicados) y mejorar la integridad de los datos (exactitud y coherencia de los datos).

⁹ Los *joins* son una cláusula utilizada en bases de datos relacionales para combinar filas de dos o más tablas en una consulta.

Tabla 1: Comparación entre los paradigmas *OLTP* y *OLAP*
Fuente: Elaboración propia

Criterio	<i>OLTP</i>	<i>OLAP</i>
Propósito	Registro operacional	Análisis y toma de decisiones
Orientación	Transacciones	Consultas analíticas
Unidad de trabajo	Un registro a la vez	Miles de registros por consulta
Normalización	Alta	Baja
Temporalidad	Datos actuales	Datos históricos
Optimizado para	Escritura rápida y concurrente	Lectura y <i>sumarización</i> de grandes volúmenes de datos

2.1.2. SISTEMA COMPUTACIONAL DE INTELIGENCIA DE NEGOCIOS

Las definiciones anteriores describen qué es el *BI* y por qué resulta necesario, pero no cómo se lleva a una organización. Ese cómo se concreta en un sistema computacional que reúne los componentes tecnológicos requeridos para capturar, integrar y explotar los datos, y es lo que se describe en esta sección.

Un sistema computacional de inteligencia de negocios es el conjunto de componentes de software, infraestructura y procesos que implementan las capacidades de recopilación, integración, almacenamiento y explotación de datos propias de *BI*. Permiten transformar los datos dispersos de una organización en información estructurada y, finalmente, en conocimiento para la toma de decisiones.

La manera en que estos componentes se organizan e interconectan se define mediante una arquitectura tecnológica, cuya elección depende del volumen y la naturaleza de los datos, de los tiempos de respuesta requeridos y del tipo de análisis que busca habilitar.

Existen 3 tipos de arquitecturas tecnológicas para implementar un sistema computacional de inteligencia de negocio (Verdugo Mendoza, 2024):

- **Arquitectura convencional:** Diseñada para gestionar volúmenes de datos moderados mediante un *Data Warehouse* centralizado. Ideal para arquitecturas de análisis descriptivo, predictivo y sistemas estructurados.
- **Arquitectura operacional:** Orientada a procesos en tiempo real. Integra datos transaccionales y operativos, permitiendo respuestas rápidas a las operaciones del negocio.

- **Grandes volúmenes de datos (*Big Data*):** Enfocada en el manejo de datos masivos y no estructurados. Utiliza tecnologías avanzadas para análisis predictivo y en tiempo real.

Este trabajo de título aplica la arquitectura convencional ya que el objetivo principal se basa en el desarrollo y la implementación de un *Data Warehouse*. La razón se debe a que los sistemas de la empresa Alfa son sistemas estructurados, la cantidad de registros operativos en total no superan las decenas de millones y se espera, en un futuro, implementar análisis descriptivo y predictivo.

La arquitectura convencional se compone de tres fases:

- **Obtención de datos:** Desde las fuentes operativas internas y externas hasta el proceso ETL.
- **Almacenamiento:** Almacenamiento de datos enriquecidos en un *Data Warehouse*.
- **Análisis:** Explotación de datos para generar valor de negocio.

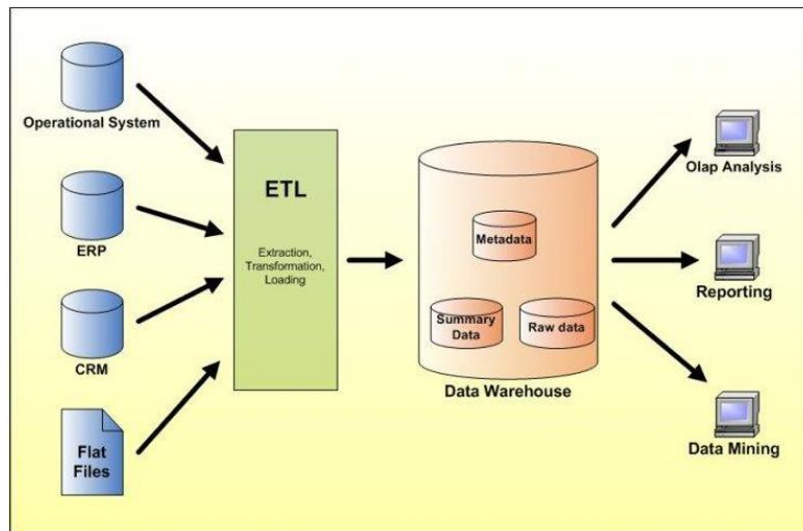


Ilustración 2: Esquema convencional *BI*

Fuente: Diapositiva de asignatura de “Bases Tecnológicas para Inteligencia de Negocios”;
profesor: Fernando Banz

Como se aprecia en la Ilustración 2, el *Data Warehouse* constituye el componente central de la arquitectura convencional, es donde convergen las tres fases descritas anteriormente. Por esta razón, antes de profundizar en cada fase, resulta necesario precisar qué se entiende por *Data Warehouse* y el proceso que lo construye y alimenta.

2.2. DATA WAREHOUSING

2.2.1 DEFINICIÓN

Tal como se anticipó, el *Data Warehouse* es el núcleo de la arquitectura convencional de inteligencia de negocios, mientras que el *Data Warehousing* corresponde al proceso que lo hace posible. Este último término es definido de muchas formas en la literatura, algunas de estas son:

- “*Data warehousing* se define como un conjunto de sistemas de toma de decisiones orientados a capacitar al especialista en información para mejorar la toma de decisiones y agilizarla.” (Linstedt & Olschimke, 2015).
- “*Data Warehousing* es el proceso mediante el cual los datos son transformados y agrupados dentro de un *Data Warehouse*, permitiendo una fusión de los mismos los cuales provienen de diversas fuentes hacia un solo contexto, lo cual permite un acceso rápido a la información y por consiguiente a la toma de decisiones.” (Silva Peñafiel, Zapata Yáñez, Morales Guamán, & Toaquiza Padilla, 2019).

Para efectos de este trabajo, se adopta la siguiente definición que es la más citada y la que mejor captura su condición de proceso.

“*Data Warehousing* es un proceso, no un producto, para recopilar y gestionar datos de diversas fuentes con el fin de obtener una visión única y detallada de una parte o la totalidad de una empresa.” (Gardner, 1998).

De esta definición se desprende una distinción clave: El *Data Warehousing* (*DWH*) es el proceso, mientras que el *Data Warehouse* (*DW*) es el repositorio resultante.



Ilustración 3: Representación de la diferencia entre *DWH* y *DW*
Fuente: Hefesto *Data Warehousing* (Bernabeu & García, 2017)

2.2.2. OBTENCIÓN DE DATOS (ETL/ELT)

2.2.2.1. FUENTES DE DATOS

La primera fase de la arquitectura convencional empieza por identificar las fuentes de datos, es decir, los sistemas que originan o almacenan la información que alimentará el repositorio analítico. Estas suelen clasificarse según su origen y su estructura.

- **Según origen**
 - **Fuentes internas:** Toda información recopilada, procesada y almacenada dentro de la propia organización. Por ejemplo: Sistemas transaccionales (OLTP) como bases de datos operativas y planillas colaborativas.
 - **Fuentes externas:** Toda información recopilada, procesada y almacenada fuera de los límites de la propia organización o sistema. Por ejemplo: Plataformas SaaS, Sistemas de terceros con conexión a través de APIs¹⁰.
- **Según estructura**
 - **Datos estructurados:** Son información altamente organizada que se ajusta a un esquema predefinido. Generalmente son almacenados en bases de datos relacionales.
Ejemplo:

Tabla 2: Ejemplo de datos estructurados
Fuente: Elaboración propia

Nombre	Apellido	Calle	Comuna	Región	País
Juan	Pérez	Av. Andrés Bello	Providencia	Metropolitana	Chile
Jhon	Doe	NULL ¹¹	NULL	New York	United States

- **No Estructurados:** Son información que no se ajusta a modelos predefinidos, lo que dificulta su organización y análisis mediante métodos tradicionales de bases de datos.
Ejemplo:

Juan Pérez vive en Av. Andrés Bello, comuna de Providencia en la Región Metropolitana de Chile. Por su parte, Jhon Doe reside en New York, United States, aunque no dispone de su dirección exacta.

¹⁰ API (Application Programming Interface): Conjunto de reglas y definiciones que permite que diferentes aplicaciones de software se comuniquen entre sí.

¹¹ NULL significa que un dato no tiene valor asignado.

- **Semiestructurados:** No siguen un esquema predefinido, pero incluyen metadatos que definen relaciones y jerarquías. Son más flexibles que los datos estructurados, pero conservan mayor organización que los datos no estructurados. Son ideales en fuentes de datos en constante evolución ya que su estructura puede cambiar dinámicamente. Ejemplo: (Notar que los campos vacíos se omiten en lugar de marcarlos como *NULL*)

```
[
  {
    "nombre": "Juan",
    "apellido": "Pérez",
    "direccion": {
      "calle": "Av. Andrés Bello",
      "comuna": "Providencia",
      "region": "Metropolitana",
      "pais": "Chile"
    }
  },
  {
    "nombre": "Jhon",
    "apellido": "Doe",
    "direccion": {
      "region": "New York",
      "pais": "Estados Unidos"
    }
  }
]
```

Ilustración 4: Ejemplo de datos semiestructurados
Fuente: Elaboración Propia

En síntesis, las fuentes que alimentan el repositorio analítico son diversas tanto en su origen, interno o externo, como en su estructura, que puede ser estructurada, semiestructurada o no estructurada. Esta heterogeneidad es la que justifica la existencia de un proceso capaz de reunir esos datos dispersos, homogeneizarlos y dejarlos listos para su almacenamiento y análisis. Dicho proceso es el que se aborda en la siguiente sección.

2.2.2.2. PROCESO ETL

En todo sistema de inteligencia de negocios, el proceso de extracción, transformación y carga (ETL) cumple un papel decisivo, ya que es el mecanismo mediante el cual los datos que residen dispersos en los sistemas de una organización se convierten en información lista para el análisis. A grandes rasgos, el ETL toma los datos desde sus fuentes de origen, los depura y los estructura conforme a reglas que aseguran su consistencia y calidad, y los deposita en un repositorio analítico como un *Data Warehouse* o *Data Mart*¹². Su objetivo principal es garantizar que la información disponible para la toma de decisiones sea precisa, coherente y confiable. Es importante resaltar que esta etapa suele absorber la mayor parte de los recursos de un proyecto, incluso, se estima que consume la mitad del esfuerzo total involucrado (Kimball & Caserta, 2004). El desarrollo de este proceso es un gran desafío porque se ven involucradas muchas restricciones en su diseño.

Establecer la arquitectura de un sistema ETL comienza por reunir y comprender todos los requisitos conocidos, las realidades y las limitaciones que afectan al sistema. Estos requisitos son en su mayoría restricciones con las que se debe convivir y a las que se debe adaptar el sistema. A continuación, se describen las principales categorías de requisitos.

- **Necesidades del negocio:** Son el contenido de información que necesitan los usuarios del sistema BI para tomar decisiones empresariales.
- **Cumplimiento:** Demostrar que los números informados son precisos, completos y no han sido manipulados.
- **Calidad de los datos:** Asegurar que los datos sean precisos, completos, consistentes y actualizados antes de su carga.
- **Latencia:** Definir qué tan rápido debe llegar la información a los usuarios del sistema BI.
- **Procesamiento End-to-end (E2E):** El proceso debe ser capaz de ejecutarse de manera programada y correcta, desde las extracciones hasta las interfaces de explotación.

2.2.2.2.1 EXTRACCIÓN (EXTRACT)

La extracción es la primera fase del proceso y consiste en leer y comprender los datos de las fuentes de origen y copiarlos al entorno donde el sistema ETL puede operar sobre ellos de forma independiente de los sistemas operacionales. A partir de ese punto, los datos pasan a formar parte del repositorio analítico (Kimball & Ross, 2013). Su diseño es crítico, ya que la forma, el orden y la frecuencia con que se extraen los datos condicionan todo el procesamiento posterior.

¹² Subconjunto de un Data Warehouse centrado en un proceso o área específica de una organización.

- **Mecanismos de acceso a las fuentes:** Dado que las fuentes son heterogéneas en formato y tecnología, esta fase debe contemplar distintas formas de conexión. Las más habituales son:
 - **Conectores de base de datos:** Interfaces estandarizadas como *ODBC* y *JDBC* que permiten acceder a bases de datos relacionales (Vaisman & Zimányi, 2014)
 - **APIs y servicios web:** Interfaces que exponen las plataformas externas y los sistemas de terceros de tipo *SaaS* para entregar sus datos de forma programática.
 - **Archivos planos:** Formatos de intercambio como *CSV*, *JSON* o *XML*, frecuentes en exportaciones y cargas manuales.
- **Modalidades de extracción:** Según el volumen de datos que se transfiere en cada ejecución, se distinguen dos modalidades:
 - **Extracción completa (*full*):** Transfiere la totalidad de los datos de la fuente en cada ejecución. Es propia de la carga inicial del repositorio o de migraciones masivas, pero resulta costosa en recursos y tiempo cuando se ejecuta sobre tablas grandes.
 - **Extracción incremental:** Transfiere solo los registros nuevos o modificados desde la última ejecución, reduciendo el impacto sobre los sistemas de origen y los tiempos de procesamiento. Es la modalidad adecuada para las operaciones recurrentes, como las actualizaciones diarias (Rainardi, 2008). La extracción incremental exige identificar qué datos cambiaron desde la última carga, tarea conocida como captura de datos modificados (*Change Data Capture, CDC*), cuyo objetivo es aislar las inserciones, ediciones y eliminaciones sin recurrir a una actualización completa.

La elección de la modalidad de extracción depende del tamaño de cada fuente, de la frecuencia de actualización requerida y del impacto admisible sobre los sistemas operacionales. En este trabajo, donde conviven fuentes internas y una fuente externa de campañas, esa decisión se toma de forma diferenciada por fuente, en línea con el objetivo de definir frecuencias de actualización personalizadas.

2.2.2.2 TRANSFORMACIÓN (*TRANSFORM*)

La transformación es la fase que convierte los datos extraídos, provenientes de formatos y orígenes diversos, en un conjunto uniforme y coherente apto para el análisis. Es donde el sistema *ETL* efectivamente agrega valor a la información, al modificarla y enriquecerla conforme a las reglas de negocio de la organización.

La transformación se suele dividir en tres tareas esenciales:

- **Limpieza:** Elimina errores e inconsistencias y estandariza los datos, corrigiendo elementos como errores ortográficos, conflictos de dominio o valores faltantes. También se identifican y eliminan los registros repetidos (*deduplicación*).

- **Integración:** Combina y unifica datos de fuentes distintas, resolviendo las diferencias de nomenclatura, formato y significado para que toda la información quede expresada bajo definiciones comunes. Aquí se estandarizan formatos.
- **Agregación (sumarización):** Resume los datos según el nivel de granularidad definido para el repositorio. También se crean nuevos campos a partir de cálculos sobre los datos existentes.

El conjunto de estas operaciones se rige por un protocolo de reglas alineado con los objetivos del proyecto, de manera que la información resultante sea precisa, consistente y fácil de interpretar. En este trabajo la transformación adquiere especial relevancia frente a la dispersión de datos de la empresa Alfa, donde conviven fuentes con nomenclaturas y significados distintos que deben reconciliarse antes de su análisis.

2.2.2.2.3 CARGA (LOAD)

La carga es la fase final del proceso y consiste en guardar los datos ya transformados en las estructuras de destino del repositorio analítico, dejándolos disponibles para su consulta y explotación. Más allá de la carga inicial, esta fase contempla la actualización periódica del repositorio, de modo que la información se mantenga vigente para la toma de decisiones.

- **Frecuencia de actualización:** La frecuencia con que se ejecuta la carga, conocida como *refreshing*, depende de las necesidades del negocio y puede variar desde una actualización mensual hasta varias veces al día o, incluso, en tiempo real. Cuanto menor es la latencia exigida, mayores son las exigencias sobre la arquitectura que soporta el proceso.
- **Integridad y monitoreo:** En todas sus modalidades, el sistema de carga debe garantizar la integridad y la consistencia de los datos, evitando pérdidas y duplicaciones. Asimismo, conviene que registre estadísticas de cada ejecución que permitan supervisar su funcionamiento y detectar errores de forma temprana, entre ellas:
 - Tiempos de carga de cada proceso.
 - Cantidad de registros procesados, insertados o rechazados.
 - Estado de la ejecución y los errores ocurridos.

Estas capacidades de monitoreo y trazabilidad se vinculan directamente con el objetivo de automatizar el proceso y gestionar los errores sin intervención manual planteado en este trabajo.

2.2.2.3 PROCESO ETL

El orden clásico de extraer, transformar y cargar (*ETL*) no es la única forma de alimentar un repositorio analítico. El enfoque de extracción, carga y transformación (*ELT*) invierte los dos últimos pasos, ya que los datos se cargan primero en el repositorio en su formato bruto y

las transformaciones se ejecutan después, dentro del propio motor (Rainardi, 2008). Esta variante ha ganado relevancia con la masificación de los Data Warehouses en la nube.

Aunque ambos enfoques persiguen el mismo fin, difieren en aspectos sustanciales:

- **Momento de la transformación:** en el *ETL* los datos se transforman antes de cargarse, mientras que en el *ELT* se cargan primero y se transforman dentro del repositorio.
- **Infraestructura:** el *ETL* suele requerir un servidor de transformación dedicado, mientras que el *ELT* aprovecha la capacidad de cómputo del propio motor de destino.
- **Tratamiento del dato bruto:** el *ELT* conserva los datos en su estado original dentro del repositorio, lo que permite reprocesarlos o transformarlos nuevamente según el caso de uso, sin volver a las fuentes.

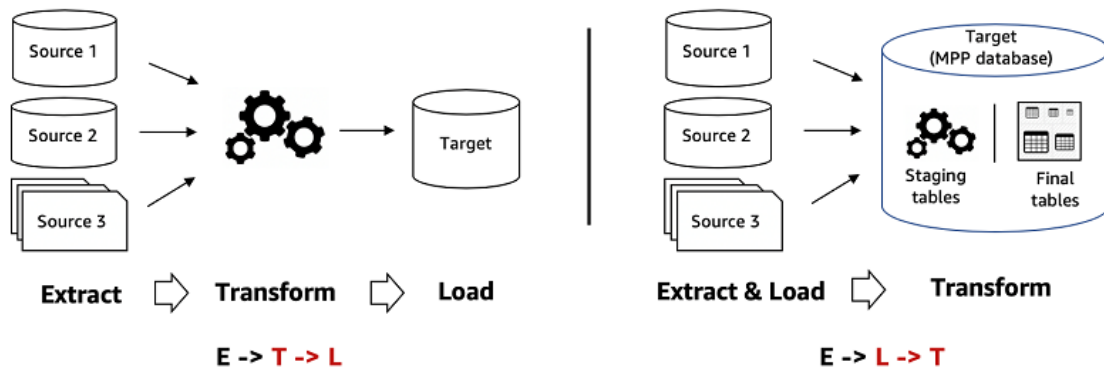


Ilustración 5: Diferencias entre *ETL* y *ELT*

Fuente: Tomado de <https://aws.amazon.com/es/compare/the-difference-between-etl-and-elt/>

En entornos de nube, este enfoque ofrece varias ventajas:

- **Escalabilidad:** Se apoya en motores de gran capacidad, como los sistemas de procesamiento masivamente paralelo (*MPP*¹³), capaces de transformar grandes volúmenes de forma eficiente.
- **Simplicidad operativa:** Al eliminar el servidor intermedio de transformación, reduce la complejidad de la arquitectura.
- **Flexibilidad:** Permite transformar el dato bruto en cualquier momento según las necesidades del análisis, en lugar de fijar la transformación como un paso rígido del *pipeline* (Nambiar & Mundra, 2022).

La selección entre ambos es una decisión de diseño relevante. El *ETL* resulta apropiado cuando se dispone de sistemas analíticos internos en los que se realiza un

¹³ MPP es una arquitectura informática que divide una tarea compleja en partes más pequeñas y las distribuye para que sean ejecutadas al mismo tiempo por múltiples procesadores o servidores, acelerando drásticamente el análisis de datos a gran escala.

preprocesamiento intensivo, mientras que el *ELT* es preferible cuando el destino es un *Data Warehouse* en la nube, donde una vez colocados los datos se puede aprovechar la potencia de un único motor para integrarlos y transformarlo.

2.2.2.4. TECNOLOGÍAS ETL/ELT

El mercado de herramientas para implementar procesos *ETL* y *ELT* es amplio y heterogéneo, y en los últimos años se ha desplazado mayoritariamente hacia servicios gestionados en la nube que evitan administrar infraestructura propia (Nambiar & Mundra, 2022). Ante esta diversidad, más que enumerar productos conviene clasificarlos según los atributos que determinan su ajuste a un proyecto. Las herramientas se distinguen principalmente por cuatro dimensiones (Kleppmann, 2017).

- **Modo de procesamiento:**
 - **Por lotes (*batch*):** Adecuado cuando los datos no requieren actualización inmediata.
 - **Vía *streaming*:** Cuando el análisis exige información en tiempo real.

Algunas soluciones admiten ambos.
- **Modelo de despliegue:**
 - ***On-premise*:** Ofrece control total sobre la seguridad a costa de mantener infraestructura propia.
 - **En la nube:** Más elástico y económico.
- **Modelo de licenciamiento:**
 - **Soluciones de código abierto:** Flexibles y personalizables, pero con mayor esfuerzo de operación.
 - **Propietarias:** Al ser propias, es más rápido el soporte, mantenimiento y actualizaciones frecuentes.
- **Grado de codificación requerido:**

Va desde plataformas *no-code/low-code* basadas en interfaces visuales de arrastrar y soltar, hasta herramientas totalmente programables orientadas a perfiles de ingeniería.

Sobre la base de estas dimensiones, la oferta de herramientas puede ordenarse en algunas familias representativas (Burdin, 2025):

- **Servicios nativos de la nube:** Diseñados para *Data Warehouses* en la nube. Trasladan la lógica de transformación al motor de destino.

Ejemplos:

 - AWS Glue, en el ecosistema de Amazon.
 - Azure Data Factory, en el de Microsoft.
 - Google Cloud Dataflow, en el ecosistema de Google Cloud.

- **Plataformas SaaS de integración de datos:** Privilegian la rapidez y la confiabilidad mediante amplios catálogos de conectores prediseñados y una operación mínima (DataCamp, 2026).

Ejemplos:

- Fivetran.
- Hevo.
- Airbyte de código abierto.

La combinación de Fivetran para la ingesta con dbt para la transformación constituye una de las pilas ELT más adoptadas en la actualidad.

- **Soluciones empresariales:** Se orientan a grandes organizaciones e industrias reguladas, donde la gobernanza, el linaje y el cumplimiento normativo resultan tan determinantes como el rendimiento.

Ejemplos:

- Informatica.
- IBM InfoSphere DataStage.
- Talend Data Integration.

- **Herramientas de orquestación:** No transforman datos por sí mismas, sino que coordinan, programan y monitorean los flujos que se ejecutan sobre los demás motores.

Ejemplo:

- Apache Airflow.

En síntesis, no existe una herramienta universalmente superior. La decisión depende del contexto de cada organización.

2.2.3. ALMACENAMIENTO (*DATA WAREHOUSE*)

Luego del proceso *ETL/ELT*, los datos enriquecidos residen en el *Data Warehouse*, el componente principal de la arquitectura. La definición de *Data Warehouse* más referenciada es la de Inmon, quien lo describe como “Una colección de datos orientada a temas, integrada, no volátil y variable en el tiempo, diseñada para soportar el proceso de toma de decisiones.” (Inmon, 2005). Estas cuatro características definen su naturaleza:

- **Orientado a temas:** Los datos se organizan en torno a las áreas de interés del negocio, como clientes, ventas o campañas, y no en torno a los procesos operacionales que los generan.
- **Integrado:** Los datos provenientes de fuentes heterogéneas se convierten a definiciones, unidades y formatos comunes, eliminando las inconsistencias de origen.
- **No volátil:** Los datos se cargan y se consultan, pero no se modifican ni eliminan, lo que preserva el histórico de la organización.
- **Variable en el tiempo:** El repositorio conserva registros históricos que permiten analizar la evolución de los indicadores y reconstruir el estado de los datos en cualquier punto del pasado.

2.2.3.1 ENFOQUES DE CONSTRUCCIÓN

Existen dos enfoques fundamentales para construir un *Data Warehouse*, atribuidos a Inmon y a Kimball, que difieren en el punto de partida y en el modelo de datos.

- **Enfoque *top-down* (Inmon):** Plantea construir primero un *Data Warehouse* corporativo centralizado y modelado de forma normalizada, que integra los datos de toda la organización. A partir de él se derivan los *Data Marts* departamentales como subconjuntos orientados a áreas específicas. Su ventaja es una visión única y consistente de los datos, a costa de un mayor esfuerzo y tiempo iniciales (Vaisman & Zimányi, 2014).
- **Enfoque *bottom-up* (Kimball):** Parte por construir *Data Marts* dimensionales orientados a procesos de negocio concretos, que luego se integran mediante dimensiones conformadas hasta formar el *Data Warehouse*, entendido como una colección de *Data Marts*. Permite obtener resultados más rápidos y de manera incremental, aunque exige orden y rigor en la conformación para evitar inconsistencias (Kimball & Ross, 2013).

La siguiente tabla resume las diferencias de ambos enfoques.

Tabla 3: Comparación entre los enfoques *top-down* y *bottom-up*
Fuente: Elaboración propia

Criterio	<i>Top-down</i> (Inmon)	<i>Bottom-up</i> (Kimball)
Punto de partida	<i>Data Warehouse</i> corporativo centralizado	<i>Data Marts</i> por proceso de negocio
Modelo de datos	Normalizado (3NF)	Dimensional
Orden de construcción	Primero el <i>DW</i> , luego los <i>Data Mart</i>	Primero los <i>Data Mart</i> , luego el <i>DW</i>
Rapidez de entrega de valor	Menor	Mayor
Integración	Garantizada desde el diseño	Lograda mediante dimensiones agrupadas
Costo y riesgo inicial	Más altos	Más bajos

2.2.3.2 TIPOS DE MODELOS DE DATOS

El modelo de datos define cómo se estructura la información dentro del repositorio. En el ámbito de los *Data Warehouse* coexisten dos paradigmas. El modelo normalizado, asociado al enfoque de Inmon, organiza los datos en tercera forma normal para minimizar la redundancia y suele emplearse en el repositorio corporativo. El modelo dimensional, asociado al enfoque de Kimball, organiza los datos en torno a los procesos de negocio y está optimizado para las consultas analíticas, por lo que predomina en las capas orientadas a los usuarios finales.

Este trabajo emplea el modelo dimensional. Su orientación a los procesos de negocio y su rendimiento en las consultas analíticas lo hacen el más adecuado para el objetivo de habilitar la reportería y los *dashboards* orientados a los usuarios finales. Además, su carácter incremental se ajusta a la estrategia del proyecto, que parte del dominio de las campañas de *marketing* y prevé escalar de forma progresiva a otras áreas, en línea con el enfoque ascendente de Kimball.

2.2.3.2.1 COMPONENTES DEL MODELO DIMENSIONAL

El modelo dimensional se construye a partir de dos tipos de tablas:

- **Tabla de hechos:** Es la tabla central del modelo y registra los eventos medibles del negocio. Cada fila representa una ocurrencia de un proceso de negocio en un nivel de detalle determinado y contiene dos clases de columnas. Por un lado, están las medidas, que son los valores numéricos que se desean analizar y, por otro lado, están las claves foráneas, que la conectan con las tablas de dimensión. Por ejemplo, en el dominio de campañas de marketing, una tabla de hechos podría registrar el

desempeño diario de cada campaña, con medidas como el número de impresiones, los clics, las conversiones y el costo, y con claves hacia las dimensiones fecha y campaña.

Tabla 4: Ejemplo de tabla de hechos de desempeño de campañas
Fuente: Elaboración propia

<i>FechaKey</i>	<i>CampañaKey</i>	Impresiones	Clics	Conversiones	Costo
20260601	361	15.000	480	35	250.000
20260601	378	9.200	310	22	180.000

- **Tablas de dimensión:** Almacenan los atributos descriptivos que dan contexto a los hechos y permiten filtrar, agrupar y describir la información al momento de analizarla, respondiendo al quién, qué, cuándo y dónde de cada evento. Siguiendo el ejemplo anterior, la dimensión fecha contendría atributos como día, mes, trimestre y año, la dimensión campaña incluiría el nombre, el tipo y el objetivo de cada campaña.

Tabla 5: Ejemplo de tabla de dimensión Campaña
Fuente: Elaboración propia

<i>CampañaKey</i>	Nombre	Tipo	Objetivo
361	<i>Black Friday 2026</i>	Promoción	Captación
378	<i>Newsletter Junio</i>	Fidelización	Retención

La clave *CampañaKey* de la dimensión es la que aparece como clave foránea en la tabla de hechos Tabla 4, lo que permite enlazar cada medición con la descripción completa de su campaña.

A ellos se asocian dos conceptos:

- **Medida:** Es cada valor numérico susceptible de análisis, como el costo o el número de conversiones del ejemplo que aparece en la Tabla 4.
- **Granularidad:** El grado de detalle con que se registra cada hecho, es decir, qué representa una sola fila de la tabla de hechos. En el ejemplo, la granularidad corresponde al desempeño de una campaña en un canal durante un día, de modo que una granularidad más fina, como el desempeño por hora, multiplicaría el volumen de registros, mientras que una más gruesa, como el desempeño mensual, lo reduciría.

2.2.3.2.2 ESQUEMAS DIMENSIONALES

La forma en que se relacionan la tabla de hechos y las tablas de dimensión da origen a distintos esquemas, que se diferencian principalmente por el grado de normalización de las dimensiones y por la cantidad de tablas de hechos involucradas.

- **Esquema estrella (*star schema*):**

Es el esquema más difundido y su nombre se debe a la disposición de sus tablas. Como se observa en la Ilustración 6, una tabla de hechos central se conecta directamente con un conjunto de tablas de dimensión *desnormalizadas*¹⁴, una por cada dimensión, formando una figura que recuerda a una estrella.

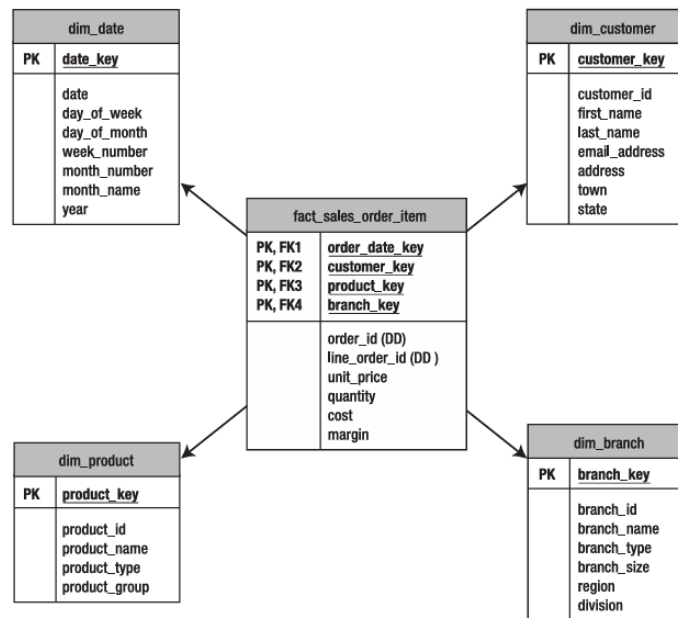


Ilustración 6: Ejemplo de esquema estrella (*star schema*)

Fuente: Building a Data Warehouse: With Examples in SQL Server (Rainardi, 2008)

Al estar *desnormalizadas*, cada dimensión concentra todos sus atributos en una sola tabla. Su principal ventaja es la simplicidad y el buen rendimiento de las consultas, ya que requiere pocas uniones para reconstruir la información. Como contrapartida, la desnormalización introduce cierta redundancia en las tablas de dimensión por lo que se eleva el espacio de almacenamiento.

- **Esquema copo de nieve (*snowflake schema*):**

Surge al normalizar las tablas de dimensión del esquema estrella. En lugar de concentrar todos los atributos en una sola tabla, cada dimensión se descompone en varias tablas relacionadas que reflejan sus jerarquías. Esta normalización reduce la redundancia y el espacio de almacenamiento, además de facilitar el mantenimiento

¹⁴ La desnormalización es una técnica de diseño que consiste en introducir intencionalmente redundancia de datos.

de la integridad. A cambio, el modelo se vuelve más complejo y las consultas requieren un mayor número de uniones, lo que puede afectar el rendimiento.

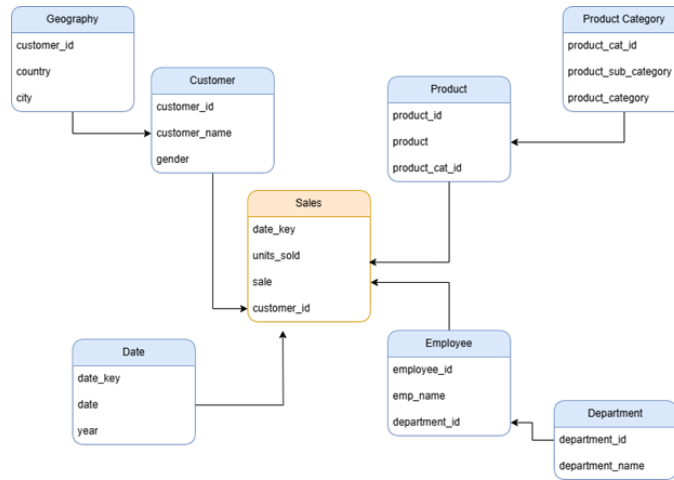


Ilustración 7: Ejemplo de esquema copo de nieve (*Snowflake schema*)

Fuente: Disponible en <https://help.pyramidanalytics.com/>

- **Esquema constelación (*fact constellation*):**

También llamado esquema galaxia, está compuesto por múltiples tablas de hechos que comparten tablas de dimensión comunes. Resulta apropiado cuando se modelan varios procesos de negocio relacionados. Estas dimensiones compartidas, conocidas como dimensiones conformadas, son las que permiten analizar de forma coherente distintos procesos bajo los mismos criterios.

Este esquema es el que da forma a un *Data Warehouse* construido como una colección de *Data Marts* agrupados.

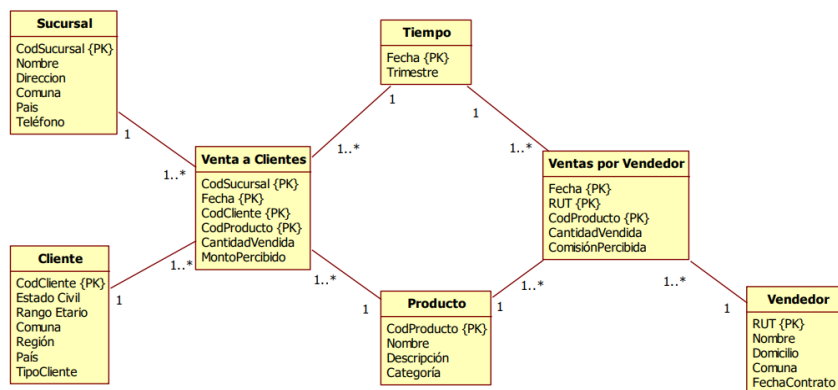


Ilustración 8: Ejemplo de esquema constelación (*Constellation schema*)

Fuente: Diapositiva de asignatura de “Bases Tecnológicas para Inteligencia de Negocios”; profesor: Fernando Banz

2.2.3.3. CONSTRUCCIÓN POR CAPAS DEL *DATA WAREHOUSE*

Los esquemas anteriores describen cómo se modelan y se relacionan las tablas dentro del repositorio, pero no cómo fluyen los datos hasta conformar ese modelo.

Una práctica habitual al construir un *Data Warehouse* moderno consiste en organizar los datos en capas sucesivas, donde cada una incrementa el nivel de calidad, estructura y valor de negocio de la información.

Una formulación contemporánea de este patrón es la arquitectura medallón, difundida en el contexto de las plataformas en la nube, que nombra las capas según el nivel de refinamiento de los datos y constituye un diseño lógico independiente de la tecnología (Microsoft, 2026). Adoptando esa nomenclatura, las capas consideradas en este trabajo son las siguientes.

- **Capa *Raw* (o bronce):** Almacena los datos tal como llegan desde las fuentes, sin transformación alguna. Conserva el dato original y su histórico, lo que permite reprocesarlo ante cualquier cambio en las reglas de negocio.
- **Capa *Staging*:** Zona de preparación transitoria donde se aplican limpiezas y normalizaciones básicas, sirviendo de puente entre el dato bruto y el dato ya tratado.
- **Capa *Silver* (o plata):** Contiene los datos ya limpios, validados, integrados y conformados, aptos para análisis intermedios y para alimentar las capas superiores.
- **Capa *Gold* (u oro):** Almacena los datos altamente refinados y agregados, modelados de forma dimensional.
- **Capa *semántica*:** Define de forma centralizada las métricas, los indicadores y las relaciones de negocio, de modo que las herramientas de BI consuman definiciones consistentes bajo un mismo lenguaje.

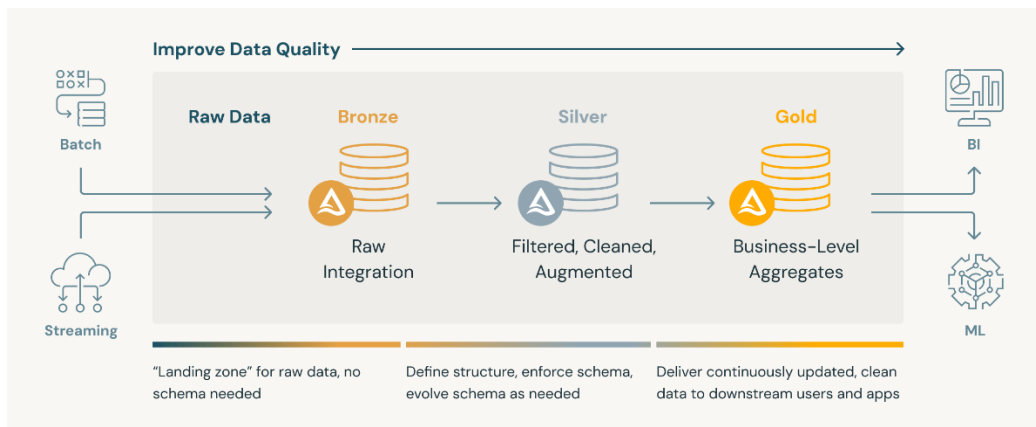


Ilustración 9: Esquema de patrón arquitectónico medallón

Fuente: Tomado de <https://www.databricks.com/blog/what-is-medallion-architecture>

Si bien la arquitectura medallón se popularizó en el ámbito de los *data lakes* y los *lakehouses*, corresponde a un patrón lógico de refinamiento progresivo cuyas raíces se encuentran en las prácticas tradicionales de *Data Warehousing*, por lo que es igualmente aplicable a un *Data Warehouse*.

Separar el flujo en capas facilita la trazabilidad, permite reprocesar los datos sin afectar a las capas superiores y habilita el trabajo en paralelo de los equipos. Además, esta organización materializa el flujo *ETL/ELT* descrito anteriormente, ya que los datos avanzan desde la capa *raw* hasta la *gold* a medida que son transformados, en línea con el enfoque adoptado en este trabajo.

2.2.3.4. TECNOLOGÍAS

Una vez definidos el enfoque de construcción y el modelo de datos, falta determinar la tecnología sobre la cual se materializa el *Data Warehouse*. Históricamente, estos repositorios se implementaban en sistemas *on-premise*, sobre infraestructura propia de la organización, pero, en la última década, la adopción se ha desplazado de forma sostenida hacia los almacenes de datos en la nube (*Cloud Data Warehouses*) (Nambiar & Mundra, 2022).

Los *Data Warehouse* tradicionales *on-premise* se ejecutan sobre servidores administrados por la propia organización (Rainardi, 2008):

- **Ventajas:** Control total sobre la seguridad, la configuración y los datos, además de una baja latencia al no depender de la red.
- **Limitaciones:** Inversión inicial elevada en hardware, mantenimiento continuo y escalamiento rígido, ya que ampliar la capacidad implica adquirir e instalar nueva infraestructura.

Ejemplos: Oracle, Teradata, IBM Db2 y Microsoft SQL Server.

Los almacenes de datos en la nube surgen para superar estas limitaciones. Su rasgo arquitectónico distintivo es la separación entre el almacenamiento y el cómputo, lo que permite escalar cada uno de forma independiente y elástica, junto con un modelo de pago por uso que evita la inversión inicial en infraestructura (Kleppmann, 2017). La mayoría se apoya en arquitecturas de procesamiento masivamente paralelo (*MPP*) y en almacenamiento columnar y, en algunos casos, en un modelo *serverless* en el que el proveedor gestiona por completo los recursos. Entre las plataformas más relevantes del mercado actual destacan (Nambiar & Mundra, 2022):

- **Snowflake:** Se distingue por una arquitectura de tres capas desacopladas (almacenamiento, cómputo y servicios) y por operar de forma *multinube* sobre AWS, Azure y Google Cloud. Es valorada por su facilidad de uso, su escalamiento elástico y sus capacidades de intercambio de datos (*data sharing*).
- **Amazon Redshift:** el almacén de datos de Amazon Web Services, basado en una arquitectura *MPP* madura y disponible en modalidad aprovisionada o *serverless*. Su

mayor fortaleza es la integración nativa con el ecosistema AWS, como S3, Glue e IAM (Fregly & Barth, 2021).

- **Google BigQuery:** servicio *serverless* de Google Cloud que separa almacenamiento y cómputo y se apoya en el motor Dremel para consultas analíticas a gran escala. Factura principalmente por la cantidad de datos procesados y se integra de forma nativa con el ecosistema de Google Cloud.
- **Azure Synapse Analytics:** la propuesta de Microsoft, que combina un almacén *MPP* con un espacio de trabajo analítico integrado, en estrecha alineación con el ecosistema Azure.

La elección entre estas alternativas depende de los siguientes criterios.

- Ecosistema de nube en que opera la organización.
- Modelo y previsibilidad de los costos.
- Escalabilidad ante volúmenes crecientes.
- Grado de administración requerido (*serverless* frente a *aprovisionado*).
- Riesgo de dependencia de un proveedor (*lock-in*).
- Tipo de cargas de trabajo predominantes, ya sean exclusivamente de inteligencia de negocios o también de *machine learning*.

En síntesis, no existe una plataforma universalmente superior. La decisión responde al contexto tecnológico y a las cargas de trabajo de cada organización.

2.2.4. ANÁLISIS

La tercera fase de la arquitectura convencional corresponde al análisis, donde los datos enriquecidos y almacenados en el *Data Warehouse* se explotan para generar valor de negocio. Es la capa en la que la información se transforma en conocimiento para la toma de decisiones, habitualmente a través de operaciones multidimensionales y de herramientas de visualización.

2.2.4.1. ANÁLISIS MULTIDIMENSIONAL

Los datos del *Data Warehouse* se analizan mediante un modelo multidimensional, representado normalmente como un cubo de datos, sobre el cual se aplican las operaciones *OLAP* que permiten observar la información desde distintas perspectivas y niveles de detalle (Vaisman & Zimányi, 2014).

El modelo multidimensional concibe los datos como un cubo, o como un hipercubo cuando intervienen más de tres dimensiones, definido por dimensiones y medidas (Vaisman & Zimányi, 2014). Las dimensiones son las perspectivas desde las cuales se analiza la información, y cada una se organiza en niveles de detalle que determinan su granularidad. Los valores que toma una dimensión se denominan miembros. Las medidas, en cambio, son los valores numéricos que se desean analizar. La Ilustración 10 presenta un ejemplo clásico de cubo aplicado a las ventas, con tres dimensiones, Producto, Tiempo y Cliente, y la medida Cantidad.

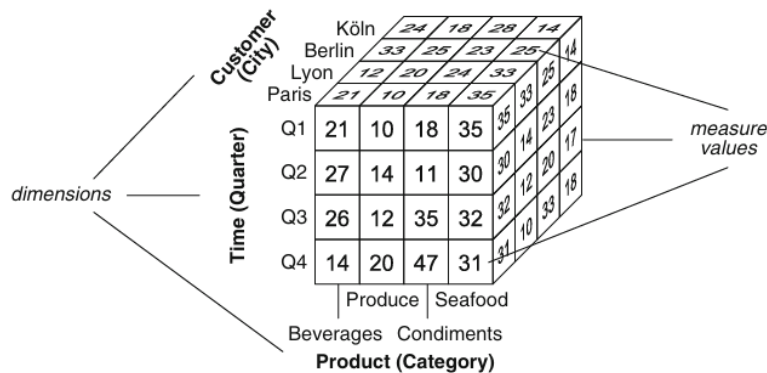


Ilustración 10: Representación de cubo *OLAP*

Fuente: Data Warehouse Systems: Design and Implementation (Vaisman & Zimányi, 2014)

En este ejemplo, cada dimensión se observa a un nivel de detalle determinado, el Producto a nivel de categoría con miembros como Bebidas o Mariscos, el Tiempo a nivel de trimestre con los miembros Q1 a Q4, y el Cliente a nivel de ciudad con miembros como Berlín, París, Lyon. Cada celda del cubo, denominada hecho, almacena el valor de la medida para una combinación específica de miembros de las dimensiones. De esta forma, una celda indicaría la cantidad de unidades vendidas de una categoría de producto, en un trimestre y para los clientes de una ciudad determinada, lo que permite responder preguntas como cuántas

unidades de la categoría Mariscos se vendieron a clientes de París durante el primer trimestre.

Un cubo puede ser denso o disperso según si todas las combinaciones posibles de miembros de las dimensiones tienen una medida asociada. Es denso cuando la mayoría de sus celdas contienen valores y disperso cuando muchas de ellas quedan vacías. En el ejemplo de la Ilustración 10, esto depende de si todos los productos son adquiridos por todos los clientes durante todos los trimestres considerados, lo que rara vez ocurre, ya que no todos los clientes ordenan productos de todas las categorías en cada período. De hecho, en las aplicaciones del mundo real los cubos suelen ser dispersos (Vaisman & Zimányi, 2014).

El cubo es una representación conceptual que facilita el análisis y no la forma física en que se guardan los datos. En la práctica se implementa sobre las tablas de hechos y de dimensiones descritas en la sección de modelado, ya sea en un esquema estrella o copo de nieve, donde las dimensiones del cubo se corresponden con las tablas de dimensión y las medidas con la tabla de hechos.

Una de las principales ventajas de representar los datos como un cubo es la posibilidad de analizarlos de forma flexible mediante las operaciones *OLAP*, que permiten observar la información desde distintas perspectivas y niveles de detalle. Las más habituales son las siguientes.

- **Roll-up:** Agrega los datos hacia un nivel más resumido, por ejemplo, de día a mes.
- **Drill-down:** Desagrega los datos hacia un mayor nivel de detalle, en sentido inverso al *roll-up*.
- **Slice:** Selecciona una porción del cubo fijando el valor de una dimensión.
- **Dice:** Selecciona un *subcubo* fijando valores en varias dimensiones.
- **Pivot:** Reorienta el cubo para analizar los datos desde otra disposición de sus dimensiones.

2.2.4.2. NIVELES DE ANÁLISIS

Las operaciones anteriores permiten explorar y describir lo que ha ocurrido en el negocio, pero la explotación de los datos puede ir más allá. Según la pregunta que se busca responder, la explotación de los datos puede alcanzar distintos niveles de madurez analítica, que suelen ordenarse de menor a mayor complejidad y valor para el negocio. Cada nivel se apoya en el anterior.

- **Descriptivo:** Responde a la pregunta de qué ha sucedido. Es el nivel más básico y se sustenta en la agregación de datos históricos para construir reportes, indicadores y *dashboards*. No explica las causas ni anticipa el futuro, solo resume lo ocurrido. En el caso de las campañas de *marketing*, correspondería a visualizar el total de conversiones, los clics y el costo de cada campaña durante el último mes. Se apoya en estadística descriptiva, como medias, sumas, percentiles y desviaciones, y en agregaciones *OLAP* sobre series históricas. No requiere algoritmos complejos, sino el resumen y la consolidación de los datos.

- **Diagnóstico:** Responde a la pregunta de por qué sucedió. Profundiza sobre el nivel descriptivo mediante técnicas como el *drill-down*, las comparaciones y el análisis de correlaciones, con el fin de identificar las causas y las relaciones detrás de un comportamiento. Por ejemplo, ante una caída en la tasa de conversión, este nivel permitiría cruzar el canal, el segmento de clientes y el período para determinar qué factor la explica.
Recorre a técnicas que buscan explicar los datos, como el análisis de correlación mediante el coeficiente de Pearson, las pruebas de hipótesis y el análisis de causa raíz. También se emplea la segmentación mediante algoritmos de agrupamiento, como *k-means*, para descubrir patrones que ayuden a entender un comportamiento.
- **Predictivo:** Responde a la pregunta de qué podría suceder. Emplea modelos estadísticos y de aprendizaje automático construidos sobre los datos históricos para estimar comportamientos futuros. En marketing podría proyectar el rendimiento esperado de una campaña o la probabilidad de conversión de un segmento, y en el negocio de *factoring* podría estimar el riesgo de impago de una operación.
Es el nivel que concentra la mayor variedad de algoritmos de aprendizaje automático. Entre los más representativos se encuentran la regresión lineal, para estimar variables continuas como la proyección de ingresos o de costos, y la regresión logística, para problemas de clasificación binaria como la probabilidad de conversión de un cliente o el riesgo de impago de una operación de *factoring*. A ellos se suman los árboles de decisión y los bosques aleatorios (*random forest*), los algoritmos de potenciación del gradiente, las redes neuronales y los modelos de series temporales como ARIMA para la elaboración de pronósticos.
- **Prescriptivo:** Responde a la pregunta de qué acción conviene tomar. Es el nivel más avanzado y, más allá de anticipar, recomienda decisiones mediante técnicas de optimización, simulación o sistemas de recomendación. Por ejemplo, podría sugerir cómo distribuir el presupuesto de marketing entre canales y segmentos para maximizar las conversiones.
Combina los modelos predictivos con técnicas de decisión. Entre ellas destacan la programación lineal y entera, para optimizar la asignación de recursos como el presupuesto entre canales, la simulación de Montecarlo, para evaluar escenarios bajo incertidumbre, los sistemas de recomendación basados en filtrado colaborativo y el aprendizaje por refuerzo.

A medida que se asciende por estos niveles aumentan tanto la complejidad técnica como el valor que aportan a la toma de decisiones. Este trabajo se sitúa inicialmente en el nivel descriptivo, dejando la arquitectura preparada para avanzar hacia el análisis predictivo en etapas futuras.

2.2.4.3. CAPA DE PRESENTACIÓN Y HERRAMIENTAS DE BI

El análisis se materializa para el usuario final en la capa de presentación, donde las herramientas de inteligencia de negocios ofrecen reportes, tableros (*dashboards*) interactivos y consultas de autoservicio que consolidan los indicadores más relevantes y permiten a los tomadores de decisión explorar la información de forma autónoma (Kimball & Ross, 2013).

Para materializar esta capa existe una amplia oferta de herramientas de visualización e inteligencia de negocios. En el espacio empresarial destacan las siguientes (Databricks, s.f.):

- **Microsoft Power BI:** Una de las más difundidas, con una interfaz intuitiva de arrastrar y soltar e integración directa con Excel.
- **Tableau, de Salesforce:** Reconocida por sus capacidades de visualización y analítica.
- **Data Studio, de Google:** Herramienta gratuita y de uso extendido para la creación de dashboards e informes, con conexión nativa a BigQuery y al ecosistema de Google. Anteriormente se llamó Looker Studio.
- **Databricks AI/BI:** Solución de BI nativa de plataforma, orientada a la analítica impulsada por inteligencia artificial.

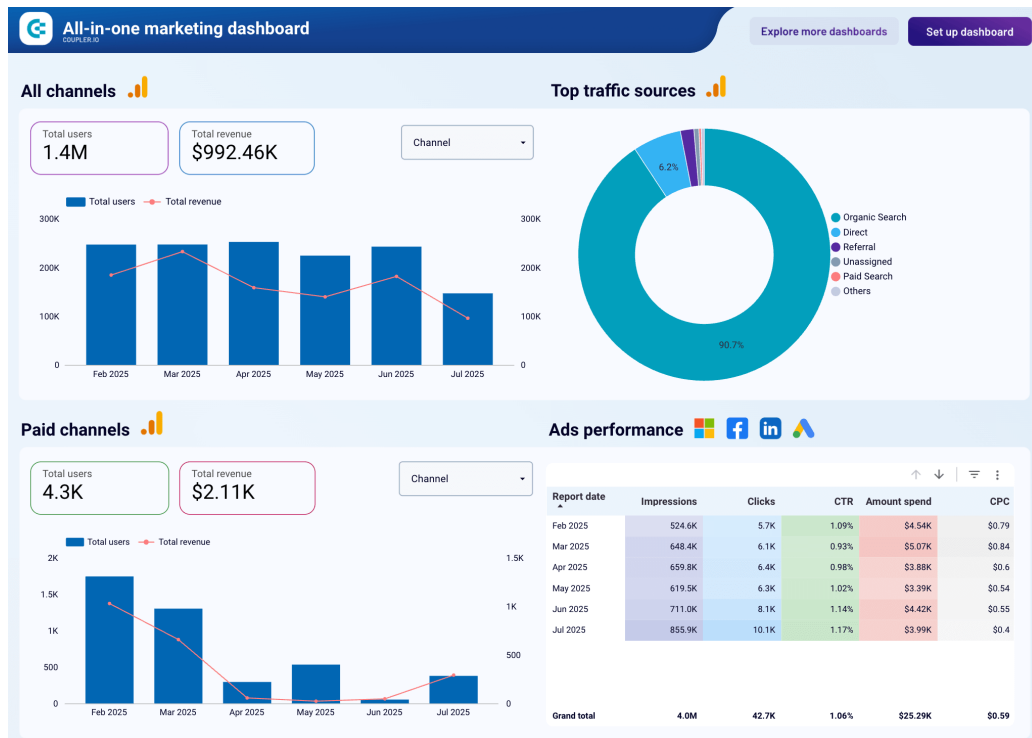


Ilustración 11: Ejemplo *dashboard* en Data Studio

Fuente: Tomado de <https://blog.coupler.io/looker-studio-dashboard-examples/>

2.2.5. METODOLOGÍAS PARA EL DESARROLLO DE UN DATA WAREHOUSE

Antes de empezar el desarrollo se debe tener en cuenta que construir un *Data Warehouse* de forma ordenada requiere seguir una metodología que organice el trabajo en pasos. La literatura ofrece varias propuestas y los estudios comparativos coinciden en que ninguna es universalmente superior, sino que la elección depende del tamaño del proyecto, de la madurez de la organización y de los recursos disponibles (Silva Peñafiel, Zapata Yáñez, Morales Guamán, & Toaquiza Padilla, 2019). A continuación, se describen las más relevantes para este trabajo.

2.2.5.1. CICLO DE VIDA DIMENSIONAL DE KIMBALL

Conocido como *Business Dimensional Lifecycle*, organiza el proyecto en torno al modelado dimensional. Se trata de una hoja de ruta que organiza las tareas del proyecto desde la planificación hasta el mantenimiento, situando la definición de los requerimientos del negocio como eje y la gestión del proyecto como una actividad transversal que acompaña todo el proceso (Kimball & Ross, 2013). Tras las actividades iniciales, el ciclo se despliega en tres vías paralelas que más tarde convergen en el despliegue.

- **Actividades iniciales**
 - **Planificación y gestión del proyecto:** Define el alcance, la justificación, los recursos y el plan de trabajo.
 - **Definición de los requerimientos del negocio:** Identifica las necesidades de información de los usuarios y es el paso central del que se derivan las tres vías siguientes.
 - **Vía de tecnología**
 - **Diseño de la arquitectura técnica:** Define la infraestructura y los componentes técnicos que soportarán la solución.
 - **Selección e instalación de productos:** Elige e instala las herramientas de bases de datos, *ETL* y *BI* acordes a esa arquitectura.
 - **Vía de datos**
 - **Modelo dimensional:** Diseña las tablas de hechos y de dimensiones a partir de los procesos de negocio identificados.
 - **Diseño físico:** Define la implementación física del modelo, como los índices, las particiones y las agregaciones.
 - **Diseño y desarrollo del *ETL*:** Construye los procesos que extraen transforman y cargan los datos en el modelo.
 - **Vía de aplicaciones de *BI***
 - **Especificación de aplicaciones *BI*:** Define el conjunto de reportes y análisis que se entregarán.
 - **Desarrollo de aplicaciones *BI*:** Construye aplicaciones sobre el repositorio.
- **Actividades de cierre**

- **Despliegue:** Pone el sistema a disposición de los usuarios, coordinando datos, aplicaciones e infraestructura.
- **Mantenimiento y crecimiento:** Opera el sistema y lo expande con nuevas fuentes, procesos y usuarios, lo que habitualmente reinicia el ciclo.

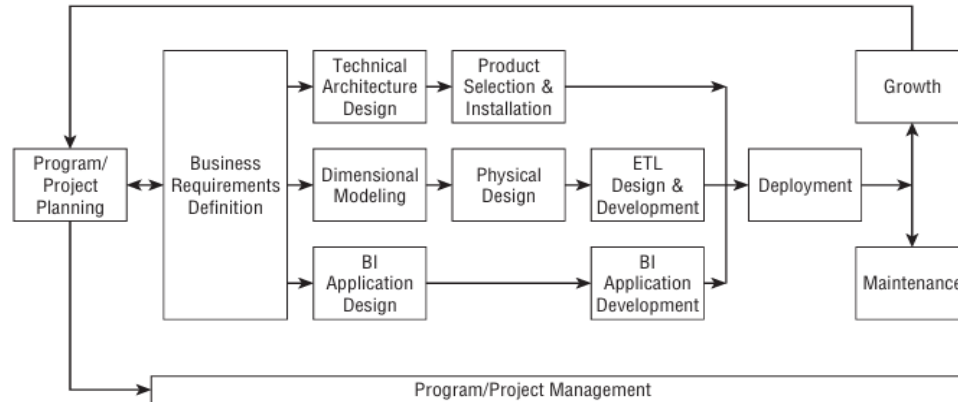


Ilustración 12: Diagrama de *Kimball Lifecycle*

Fuente: The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence, Second Edition **(Kimball, Becker, Mundy, & Thornthwaite, 2016)**

Observando la Ilustración 12, da cuenta que el ciclo de vida de Kimball se caracteriza por su naturaleza iterativa, por situar los requerimientos del negocio como eje de todas las vías y por su orientación dimensional y ascendente.

2.2.5.2. HEFESTO VERSIÓN 3

HEFESTO es una metodología orientada al desarrollo ágil de *Data Warehouses* y fundamentada en una extensa investigación y comparación de metodologías existentes. Parte de la recolección de los requerimientos y las necesidades de información de los usuarios y concluye con el esquema lógico del repositorio y sus respectivos procesos de extracción, transformación y carga. Se distingue por partir siempre de las preguntas del negocio, por su énfasis en la documentación y en comprender el porqué de cada paso, y por ser independiente de las herramientas y adaptable a cualquier ciclo de vida de desarrollo (Bernabeu & García, 2017). Su construcción se organiza en cuatro pasos.

- **Paso 1: Análisis de requerimientos**
 - **Preguntas de negocio:** Se formulan las preguntas cuyas respuestas ayudarán a tomar decisiones en a favor de los objetivos de la organización.
 - **Indicadores y perspectivas:** De cada pregunta se extraen los indicadores, que son las medidas a analizar, y las perspectivas, que son los puntos de vista desde los cuales se analizan esos indicadores.
 - **Modelo conceptual:** se representa de forma visual la relación entre los indicadores y las perspectivas identificados.

- **Paso 2: Análisis de los OLTP**
 - **Conformación de los indicadores:** Se define cómo se calcula cada indicador a partir de los datos disponibles.
 - **Mapeo:** Se relaciona el modelo conceptual con las fuentes, indicando de qué campo de cada sistema de origen se obtendrá cada indicador y perspectiva.
 - **Granularidad:** Se determina el nivel de detalle con que se almacenarán los datos.
 - **Modelo conceptual ampliado:** Se enriquece el modelo conceptual con los campos y la granularidad definidos.
- **Paso 3: Modelo lógico del Data Warehouse**
 - **Tipología:** Se elige el tipo de esquema, ya sea estrella, copo de nieve o constelación.
 - **Tablas de dimensiones:** Se diseñan a partir de las perspectivas.
 - **Tabla de hechos:** Se diseña a partir de los indicadores y la granularidad definida.
- **Paso 4: Integración de datos**

Pobla el modelo mediante los procesos de extracción, transformación y carga (*ETL* o *ELT*), contemplando tanto la carga inicial como las actualizaciones periódicas, junto con las reglas y los controles de calidad necesarios.

HEFESTO se caracteriza por su carácter ágil, por anclar todo el desarrollo en las preguntas del negocio y por su fuerte énfasis en la documentación, lo que la hace particularmente adecuada para equipos pequeños y para organizaciones que inician su recorrido analítico. Además, sus pasos integran de forma natural los conceptos ya presentados, ya que el modelo lógico del Paso 3 se apoya en los esquemas dimensionales descritos antes y el Paso 4 corresponde al proceso *ETL/ELT* desarrollado previamente.

2.2.5.3. CRISP PARA DATA WAREHOUSE

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) es una metodología concebida originalmente para proyectos de minería de datos, pero se propone una adaptación a la implementación de un *Data Warehouse*, aprovechando su organización en seis fases que no son terminantes, sino que conforman un ciclo en constante movimiento, donde es posible retroceder a fases previas según lo requiera el proyecto (Salcedo Parra, Milena Galeano, & Rodríguez B., 2010). Sus fases son las siguientes.

- **Fase 1: Comprensión del negocio:** Busca entender con claridad los requerimientos y objetivos del proyecto desde una visión de negocio. Comprende la determinación de los objetivos y los criterios de éxito, la evaluación de la situación mediante un inventario de recursos, riesgos y contingencias, terminología y costos, y la

elaboración del plan del proyecto con una valoración inicial de herramientas y técnicas.

- **Fase 2: Comprensión de los datos:** Consiste en obtener los datos y familiarizarse con ellos, reconociendo sus fortalezas para el análisis. Incluye la recolección inicial, la descripción, la exploración y la verificación de la calidad de los datos.
- **Fase 3: Preparación de los datos:** Trabaja sobre los datos realmente relevantes en un proceso de selección, depuración y transformación. Abarca la selección de los datos con sus motivos de inclusión o exclusión, la depuración, la estructuración mediante la generación de atributos y registros, la integración y el formateo.
- **Fase 4: Modelado:** Aplica las técnicas de modelado correspondientes. Comprende la selección de la técnica y sus supuestos, la generación del plan de pruebas, la construcción del modelo y su evaluación técnica.
- **Fase 5: Evaluación:** Valora si los modelos resultantes son verdaderamente útiles para los objetivos del negocio. Incluye la evaluación de los resultados respecto al éxito del negocio, la revisión del proceso y la determinación de los pasos siguientes.
- **Fase 6: Despliegue:** Implementa y divulga los modelos previamente evaluados. Comprende el plan de divulgación o implementación y el plan de monitoreo y mantenimiento.

CRISP se distingue por su raíz analítica y por su naturaleza cíclica, en la que ninguna fase es definitiva y el proceso se retroalimenta de forma continua.

2.2.5.4. COMPARACIÓN

Aunque las metodologías revisadas difieren en su origen y en su nivel de detalle, comparten un mismo patrón de fondo. Todas parten de los requerimientos o las preguntas del negocio y avanzan, a través de la comprensión y preparación de los datos y del diseño del modelo, hasta el despliegue y la explotación de la información. Las diferencias radican principalmente en el énfasis que cada una otorga a las distintas etapas del proceso.

El ciclo de vida de Kimball ofrece una visión integral del proyecto, articulada en torno al modelado dimensional y a tres vías paralelas de trabajo. HEFESTO pone el acento en el modelado y en la documentación, con un enfoque ágil que la hace accesible para equipos pequeños. CRISP, por su raíz en la minería de datos, destaca por su naturaleza cíclica y por la importancia que asigna a la evaluación de los resultados frente a los objetivos del negocio.

Tabla 6: Comparación de metodologías de desarrollo de un Data Warehouse
Fuente: Elaboración propia

Metodología	Énfasis principal	Más indicada para
<i>Kimball Lifecycle</i>	Modelado dimensional y visión integral	Proyectos que requieren una hoja de ruta completa de <i>DW/BI</i>
<i>HEFESTO</i>	Modelado y documentación	Equipos pequeños y organizaciones que inician su recorrido analítico
<i>CRISP</i>	Comprensión y evaluación de los datos	Proyectos centrados en el análisis y la minería de datos

La literatura comparativa coincide en que ninguna metodología es universalmente superior, sino que la más adecuada depende del contexto, los recursos y los objetivos de cada proyecto (Silva Peñafiel, Zapata Yáñez, Morales Guamán, & Toaquiza Padilla, 2019). Para este trabajo, el carácter incremental de la solución, el tamaño acotado del equipo y la implementación sobre una plataforma en la nube hacen que el pragmatismo de HEFESTO y el modelado dimensional de Kimball resulten especialmente afines, mientras que CRISP se conserva como referencia para la etapa de análisis predictivo que el proyecto deja habilitada a futuro. La elección definitiva y su justificación se desarrollan en el capítulo de propuesta de solución.

2.2.6. MODELOS DE ÉXITO

Por último, para determinar si un sistema de información, como un *Data Warehouse*, cumple sus objetivos, la literatura ha propuesto modelos de éxito que definen las dimensiones a evaluar. El más reconocido es el modelo de éxito de sistemas de información de DeLone y McLean, que en su versión actualizada identifica seis dimensiones interrelacionadas (DeLone & McLean, 2003)

- **Calidad del sistema:** Características técnicas como la facilidad de uso, la funcionalidad, la fiabilidad, la flexibilidad y el tiempo de respuesta.
- **Calidad de la información:** Atributos de la salida del sistema, como la exactitud, la completitud, la relevancia y la oportunidad de los datos.
- **Calidad del servicio:** El soporte que reciben los usuarios, como la capacidad de respuesta y la competencia del equipo de *TI*.
- **Uso:** El grado y la forma en que los usuarios aprovechan el sistema.
- **Satisfacción del usuario:** El nivel de conformidad de los usuarios con el sistema y su información.
- **Beneficios netos:** El impacto del sistema sobre los individuos y la organización, que constituye la medida última del éxito.

Más allá de definir el éxito, estas dimensiones ofrecen un marco concreto para evaluarlo por lo que orientan la validación de la solución propuesta.

Validar la solución consiste en demostrar que es válida para el entorno en que fue planteada. En un desarrollo de software, esto suele abordarse mediante la construcción y las pruebas del sistema, mediante su aplicación a un caso real, o comparando la situación previa con la situación posterior a la implementación sobre la base de un conjunto de indicadores cuantitativos. En este trabajo, la validación se apoya en las dimensiones del modelo de DeLone y McLean y se aborda mediante dos mecanismos:

- **Indicadores cuantitativos comparados antes y después:** Como la calidad de los datos, los tiempos de procesamiento y el esfuerzo manual de consolidación, asociados a las dimensiones de calidad del sistema y de la información.
- **Pruebas técnicas del sistema:** Verificación del correcto funcionamiento de los procesos *ETL/ELT* y de los *dashboards* construidos.

De este modo, los beneficios netos esperados, como una mayor disponibilidad de datos confiables y una reducción del esfuerzo manual, se traducen en criterios verificables que permiten evaluar el éxito de la solución.

2.3. DATOS E INTELIGENCIA DE NEGOCIOS EN EL SECTOR FINTECH

Los fundamentos presentados hasta aquí adquieren su pleno sentido al situarlos en el sector donde se aplica este trabajo. Esta sección caracteriza el ámbito fintech, primero definiendo el concepto y describiendo su crecimiento, y luego analizando el papel que cumple el Data Warehousing en el rubro. Con ello se cierra el marco conceptual y se tiende el puente hacia la propuesta de solución.

2.3.1. DEFINICIÓN Y CRECIMIENTO DEL FINTECH

El término *fintech*, contracción de las palabras inglesas *financial technology*, no cuenta con una definición única ni consensuada en el ámbito académico y empresarial (Matteo Avendaño, 2024). En un sentido amplio, la "expresión *Financial Technology (Fintech)* o Tecnología Financiera (*Fintech*) describe la aplicación de programas informáticos y otras tecnologías a los servicios bancarios y financieros" (Wilkins Binder, 2024). Otra definición plantea que "una *fintech* implica, no solo la aplicación de nuevas tecnologías digitales a los servicios financieros, sino también el desarrollo de modelos de negocio y productos que se basan en estas tecnologías y, en general, en plataformas y procesos digitales" (OECD, 2018, citado en Matteo Avendaño, 2024).

Esta conjunción de tecnología y nuevos modelos de negocio ha impulsado un crecimiento sostenido del sector. A nivel mundial, las inversiones en *fintech* crecieron de 9 mil millones de dólares en 2010 a 247,2 mil millones en 2021, y hacia 2024 las Américas concentraban cerca de 13.100 empresas, el mayor número del mundo (Statista, 2024, citado en Matteo Avendaño, 2024). En Chile el panorama no ha sido distinto y el ecosistema *fintech* también ha alcanzado una magnitud considerable. Según los catastros de Finnovista, el número de *fintech* en el país pasó de 146 en 2020 a 348 en 2024, aunque con una desaceleración, ya que el crecimiento interanual entre 2023 y 2024 fue de 16,0%, frente al 34,1% registrado entre 2021 y 2022 (Finnovista, 2024, citado en Matteo Avendaño, 2024).

En el ámbito del *factoring*, gremios como la Asociación Chilena de Empresas de *Factoring* (ACHEF) promueven la digitalización y la adopción de tecnologías *fintech* para mejorar la eficiencia y la transparencia de las operaciones (ACHEF, 2024, citado en Matteo Avendaño, 2024), lo que evidencia la pertinencia de estas soluciones en el segmento donde se inscribe el presente trabajo.

2.3.2. DATA WAREHOUSING EN FINTECH

El sector *fintech* es intensivo en datos, por lo que el *Data Warehouse* adquiere en él un rol especialmente relevante. Su utilización resulta crucial en industrias como la de los préstamos entre pares, ya que permite obtener información sobre el comportamiento de los clientes, el desempeño de las operaciones y los indicadores de riesgo, lo que mejora la toma de decisiones y el desempeño del negocio (Oktaviani & Mauritsius, 2024).

El valor del *Data Warehouse* en este contexto radica en que proporciona una visión única y consistente de los datos al integrarlos desde diversas fuentes y organizarlos de forma

coherente. Sobre esa base integrada, una organización *fintech* puede sostener capacidades centrales para su operación, como el monitoreo de los flujos de caja y del gasto operativo, junto con la evaluación del riesgo de las operaciones, factores determinantes para su sostenibilidad. La consolidación de los datos en un repositorio analítico se asocia, además, a una mayor eficiencia operacional, a una mejor calidad y consistencia de la información y a una mayor escalabilidad (Oktaviani & Mauritsius, 2024).

Esta relevancia se enmarca en la creciente adopción de la analítica de datos en los servicios financieros, donde el *Big Data* resulta determinante para impulsar la innovación, mejorar la experiencia del cliente y elevar la eficiencia operacional, además de cumplir un rol crítico en la gestión de riesgo y la detección de fraude (Aderemi, y otros, 2024). La evidencia empírica respalda este aporte, ya que la integración de la inteligencia artificial y *Big Data* en la banca amplió la cobertura de la evaluación de riesgo, al reducir en un 40,1 % los casos que quedaban sin una calificación crediticia asignada y disminuyó en un 29,6 % la tasa de incumplimiento de los préstamos (Chen, Guo, Xia, & Zhang, 2025). Estos resultados son especialmente pertinentes para el negocio del *factoring*, donde la evaluación del riesgo de las operaciones es central.

Aprovechar estos beneficios exige una infraestructura de datos robusta. La literatura sobre finanzas digitales propone arquitecturas de referencia que integran las tecnologías de *Big Data* e inteligencia artificial para sostener estas capacidades analíticas en el sector (Soldatos & Kyriazis, 2022). No obstante, su adopción enfrenta desafíos propios del rubro, como las preocupaciones por la privacidad de los datos, el cumplimiento regulatorio, los requerimientos de infraestructura tecnológica y la escasez de profesionales calificados, lo que convierte a la gobernanza de datos en un componente esencial (Aderemi, y otros, 2024).

En síntesis, el sector *fintech* se caracteriza por un crecimiento sostenido y por una operación intensiva en datos, en la que la capacidad de integrar y explotar información dispersa se ha vuelto un factor determinante de competitividad. En este escenario, el *Data Warehouse* se consolida como una pieza clave, al ofrecer una visión única y consistente de los datos que habilita la reportería, el monitoreo del negocio y, de manera progresiva, el análisis del riesgo. La empresa Alfa, en su condición de *fintech* chilena del segmento *factoring*, comparte tanto los desafíos como las oportunidades descritas, ya que enfrenta la dispersión de sus datos entre múltiples fuentes y, al mismo tiempo, opera en un mercado donde la analítica resulta cada vez más decisiva. De esta forma, los fundamentos desarrollados a lo largo de este capítulo, desde la inteligencia de negocios y el *Data Warehousing* hasta su aplicación en el ámbito *fintech*, sustentan la solución que se presenta en el capítulo siguiente, orientada a centralizar y estructurar la información de la organización y a habilitar, de manera incremental, capacidades analíticas cada vez más avanzadas.

CAPÍTULO 3: PROPUESTA DE SOLUCION

3.1. VISIÓN GENERAL DE LA SOLUCIÓN

La solución propuesta consiste en el diseño e implementación de un *Data Warehouse* escalable que centraliza y estructura los datos dispersos de la empresa Alfa, con el fin de habilitar la reportería automatizada y sentar las bases para análisis más avanzados. Debido a la dispersión de la información entre sistemas operacionales, planillas colaborativas y plataformas externas descrita en el Capítulo 1, la propuesta consolida esas fuentes en un repositorio analítico único, organizado por capas y construido sobre una arquitectura en la nube.

El desarrollo se guía por la metodología HEFESTO, complementada con el modelado dimensional de Kimball, y adopta un enfoque orientado a *ELT*. La información avanza a través de capas sucesivas, desde el dato en bruto hasta el dato modelado y listo para el consumo, lo que asegura la trazabilidad y permite escalar de forma incremental. Esta primera materialización se acota al dominio de las campañas de *marketing*, sin perjuicio de que la arquitectura se diseñe para incorporar nuevas fuentes y áreas en el futuro.

La Ilustración 13 presenta una visión general de la arquitectura, desplegada sobre dos nubes. En AWS residen las fuentes y la extracción de la fuente externa, mientras que en Google Cloud Platform se alojan la replicación de las fuentes internas, el *Data Warehouse* y el monitoreo. La obtención de los datos sigue dos rutas complementarias, que reflejan los dos enfoques de integración descritos en el marco conceptual.

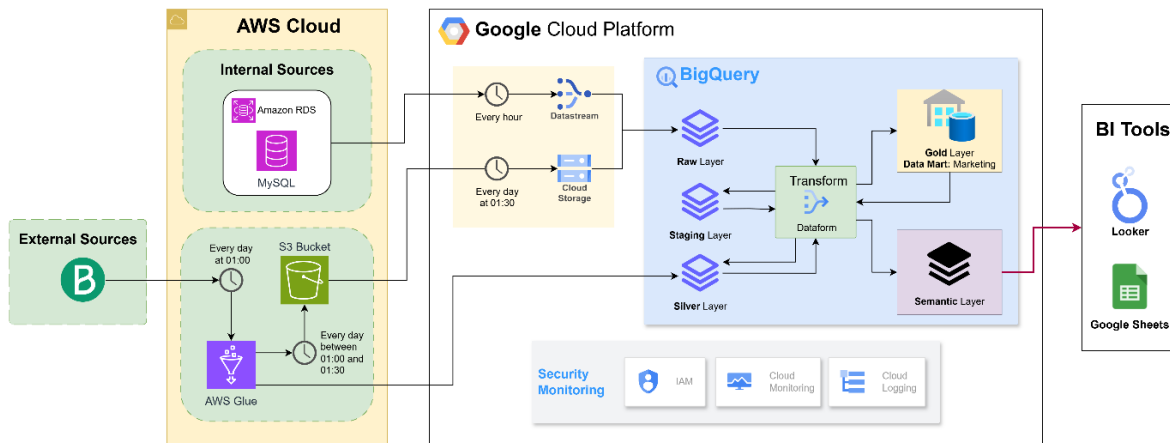


Ilustración 13: Diagrama de arquitectura de Data Warehousing propuesta
Fuente: Elaboración propia

Las fuentes internas, alojadas en Amazon RDS con MySQL, se replican cada hora mediante Datastream y aterrizan en la capa *raw* de BigQuery, donde posteriormente se transforman con Dataform siguiendo un enfoque *ELT*. Esta replicación de una nube a otra responde a

que el equipo de *Data* de la empresa Alfa cuenta con mayor experiencia en las herramientas de Google Cloud Platform, lo que facilita la operación y el mantenimiento del repositorio.

Por su parte, la fuente externa de campañas, proveniente de Brevo¹⁵, se procesa cada día a las 01:00 mediante un trabajo *ETL* en AWS Glue, cuyo resultado, ya transformado, se carga directamente a la capa *silver* en *GCP*. En paralelo, una copia de esos datos en formato *JSON* se conserva a través de Amazon S3 y Cloud Storage hasta la capa *raw*, lo que asegura la trazabilidad del dato original y proporciona un respaldo ante eventuales reprocesos.

Una vez en BigQuery, la información avanza por las capas *raw*, *staging*, *silver*, *gold* y *semantic* mediante las transformaciones de Dataform, hasta quedar disponible en el *Data Mart* de *marketing* y en la capa semántica. Desde allí, los datos se exponen a las herramientas de visualización, Data Studio y Google Sheets, para su explotación por los usuarios finales. De forma transversal, los servicios de IAM, Cloud Monitoring y Cloud Logging sostienen la seguridad y el monitoreo de la solución.

Los apartados siguientes desarrollan cada uno de estos componentes en el mismo orden en que se construyó la solución, desde la metodología y el análisis de los requerimientos del negocio hasta el diseño de la arquitectura, el modelo de datos, los procesos que la alimentan y, finalmente, la explotación y la documentación de la información.

3.2. METODOLOGÍA DE DESARROLLO APLICADA

El desarrollo de la solución se estructuró siguiendo la metodología HEFESTO, complementada con el modelado dimensional de Kimball. Como se justificó en el marco conceptual, esta combinación se ajusta a las características del proyecto, ya que HEFESTO ofrece un proceso ágil que parte de las preguntas de negocio y resulta adecuado para un equipo acotado y un desarrollo incremental, mientras que el modelado dimensional de Kimball aporta la estructura orientada a los procesos de negocio sobre la cual se construyen las capas analíticas del repositorio.

La construcción siguió los cuatro pasos de HEFESTO, que ordenan buena parte del capítulo y se apoyan en la arquitectura presentada más adelante.

3.3. ANÁLISIS DE REQUERIMIENTOS

Siguiendo el primer paso de HEFESTO, el desarrollo parte de las preguntas de negocio del dominio de las campañas de *marketing*, de las cuales se derivan los indicadores a medir y las perspectivas desde las cuales analizarlos. Este dominio se priorizó porque concentraba una necesidad concreta y porque su información se encontraba dispersa entre las fuentes internas y la plataforma externa de envíos lo que requería mayor esfuerzo para comprender los datos.

¹⁵ Brevo es una plataforma de marketing que está especializada en email marketing, permitiendo el envío automático de correos electrónicos y SMS y facilitando la segmentación de audiencias y personalización de los mensajes.

3.3.1. PREGUNTAS DE NEGOCIO

A partir de las dificultades identificadas, como la falta de una visión consolidada de las campañas y la dificultad para medir la interacción por cliente, se formularon las preguntas que el repositorio debe responder, entre ellas se presentan las principales y más complejas:

- ¿Cuál es el desempeño de cada campaña según el objetivo que persigue?
- ¿Cuál es el pilar¹⁶ y el subtipo¹⁷ que mejor funciona según la semana del mes?
- ¿Cómo se comporta la interacción, como las aperturas y los clics, al paso de los días por cliente?
- ¿Qué campañas derivan en registros¹⁸ y cesiones¹⁹?
- ¿Cuántos clientes nuevos generan las campañas?

3.3.2. INDICADORES

De estas preguntas se desprenden los indicadores, que corresponden a los KPIs oficiales definidos por el equipo en la nomenclatura estandarizada de envíos (antecedentes en el ANEXO A). Los principales son:

1. **Cantidad de aperturas:** Número de receptores que abrieron la campaña de email, contando una sola vez a cada receptor que registró al menos una apertura. Se expresa en receptores, se calcula a nivel de campaña y considera el total acumulado.

$$\text{Cantidad de aperturas} = \sum \text{receptores (con } \geq 1 \text{ apertura)}$$

2. **Tasa de apertura:** Proporción de los receptores de una campaña que la abrieron dentro de los 14 días posteriores al envío, plazo en que se concentra el impacto de una campaña de email. Se expresa como porcentaje y se calcula a nivel de campaña.

$$\text{Tasa de apertura} = \frac{\sum \text{receptores (con } \geq 1 \text{ apertura \& apertura } \leq 14 \text{ días)}}{\sum \text{envíos}}$$

3. **Clics promedio por campaña:** Cantidad promedio de clics que realiza un cliente por cada campaña que recibe. Se expresa en clics por campaña, se calcula a nivel de cliente y considera el total acumulado.

$$\text{Clics promedio por campaña} = \frac{\sum \text{clics del cliente}}{\sum \text{campañas recepcionadas}}$$

¹⁶ Pilar hace referencia al tema de la campaña. Ejemplos: Educación, Growth, Anuncio.

¹⁷ Refiere a segmentos de clientes como, por ejemplo, nuevos, retención, reactivación; o también a subcategorías de pilares como, por ejemplo, año nuevo (subtipo de anuncio), nuevos (subtipo de Growth).

¹⁸ Un registro significa haberse registrado en la plataforma de la empresa Alfa.

¹⁹ Una cesión en el rubro de *factoring* significa cursar una operación. Cesión significa que el cliente cede una factura.

4. **Tasa de clics (*Click-Through Rate, CTR*):** Proporción de los receptores de una campaña que hicieron al menos un clic, respecto del total de receptores. Mide el impacto de la campaña sobre todos quienes la recibieron. Se expresa como porcentaje y se calcula a nivel de campaña sobre el total acumulado.

$$CTR = \frac{\sum \text{receptores (con } \geq 1 \text{ clic)}}{\sum \text{receptores}} * 100$$

5. **Tasa de clics por apertura (*Click-To-Open Rate, CTOR*):** Proporción de los receptores que abrieron la campaña y que además hicieron al menos un clic. Aísla el impacto del contenido entre quienes efectivamente abrieron el correo, con independencia de la tasa de apertura. Se expresa como porcentaje y se calcula a nivel de campaña sobre el total acumulado.

$$CTOR = \frac{\sum \text{receptores (con } \geq 1 \text{ clic)}}{\sum \text{receptores (con } \geq 1 \text{ apertura)}} * 100$$

6. **Cantidad de registros:** Número de clientes que se registraron en la plataforma dentro de los 7 días posteriores a su interacción con la campaña. Se expresa en clientes y se calcula a nivel de campaña.

$$\text{Cantidad de registros} = \sum \text{clientes registrados atribuidos}$$

7. **Cantidad de operaciones:** Número de operaciones cursadas dentro de los 7 días posteriores a la interacción del cliente con la campaña, atribuidas a esta. Se expresa en operaciones y se calcula a nivel de campaña.

$$\text{Cantidad de operaciones} = \sum \text{operaciones atribuidas}$$

8. **Operaciones promedio por receptor:** Número medio de operaciones atribuidas por receptor de la campaña que al menos haya aperturado. Se expresa en operaciones por receptor y se calcula a nivel de campaña, con la misma ventana de 7 días.

$$\text{Operaciones promedio por receptor} = \frac{\sum \text{operaciones atribuidas}}{\sum \text{receptores (con } \geq 1 \text{ apertura)}}$$

9. **Monto total de operaciones:** Suma del monto de las operaciones atribuidas a la campaña dentro de la ventana de 7 días. Se expresa en pesos chilenos (CLP) y se calcula a nivel de campaña.

$$\text{Monto total de operaciones} = \sum \text{monto de operaciones atribuidas}$$

10. **Monto promedio de operación:** Monto medio por operación atribuida a la campaña. Se expresa en CLP por operación y se calcula a nivel de campaña, con la misma ventana de 7 días.

$$\text{Monto promedio de operación} = \frac{\sum \text{monto de operaciones atribuidas}}{\sum \text{operaciones atribuidas}}$$

Cada indicador puede calcularse desde la granularidad mínima (campaña, cliente y día) hasta niveles superiores de agregación. Las ventanas temporales difieren según la naturaleza del indicador. La interacción directa, como la apertura, se acota a los 14 días posteriores al envío, plazo en que se concentra el impacto de una campaña de email, mientras que la conversión, es decir, los registros y las operaciones, se atribuye dentro de los 7 días posteriores a la interacción del cliente con la campaña.

Adicionalmente, la labor del arquitecto de sistemas de inteligencia de negocios debe tener la capacidad de anticipar necesidades futuras de los usuarios por lo que, además, se habilita análisis tales como:

- Análisis por ubicación geográfica del cliente.
- Análisis por inversionista. La definición de “cliente” para el área de *marketing* es distinta a las de las otras áreas. Para *marketing*, un cliente es todo aquel a quien le envían campañas por lo que en esta etiqueta se agrupan: organizaciones que cursan operaciones, prospectos, clientes inactivos y también inversionistas.

3.3.3. MODELO CONCEPTUAL DE REQUERIMIENTOS

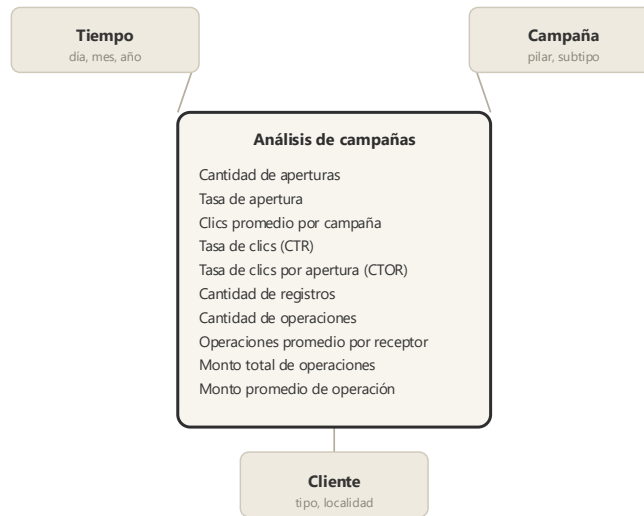


Ilustración 14: Modelo conceptual de requerimientos
Fuente: Elaboración propia

El modelo conceptual representa de forma visual la relación entre los indicadores y las perspectivas identificados, y constituye la base del modelo dimensional. Como muestra la Ilustración 14, en el centro se ubican los indicadores que el análisis busca medir, mientras que alrededor se disponen las perspectivas desde las cuales se los analiza, esto es, el tiempo, la campaña y el cliente. Cada perspectiva contempla además sus propios niveles de detalle, como el día, el mes y el año en el tiempo, el pilar y el subtipo en la campaña, o el tipo y la localidad en el cliente, lo que permite analizar cada indicador desde su granularidad mínima hasta su mayor nivel de agregación.

3.4. FUENTES DE DATOS

Definidos los requerimientos, se identifican las fuentes desde las cuales se obtienen los datos que alimentan los indicadores y se determina la granularidad. Intervienen dos tipos de fuentes, las internas, correspondientes a la base operacional alojada en Amazon RDS con MySQL, y la externa, correspondiente a la plataforma de campañas Brevo.

- **Mapeo:** Aquí se establece el origen de los datos que los alimentan, relacionando cada indicador con su fuente y sus campos. Las métricas de interacción provienen de Brevo, mientras que las de conversión requieren cruzar los envíos de Brevo con las operaciones y los clientes de la base operacional. La siguiente tabla resume este mapeo para los indicadores principales. Además, se omiten nombres específicos de los atributos de las tablas por razones de confidencialidad.

Tabla 7: Mapeo de los indicadores según fuente(s) de origen
Fuente: Elaboración propia

Indicador	Fuente	Origen del dato
Aperturas (cantidad y promedio)	Brevo	Cantidad de aperturas y total de envíos de cada campaña
CTR (por cliente y campaña)	Brevo	Clics y enlaces de cada envío
Cantidad de registros	Base operacional y Brevo	Registros de usuarios asociados al envío de campaña
Operaciones (cantidad y promedio)	Base operacional y Brevo	Operaciones cursadas respecto los receptores que leyeron el envío
Monto de operaciones (total y promedio)	Base operacional y Brevo	Monto de las operaciones respecto los receptores que leyeron el envío

- **Granularidad:** El nivel de detalle mínimo corresponde a la combinación de campaña, cliente y día. Desde esta granularidad, los indicadores pueden agregarse hacia niveles superiores, como el pilar en la campaña o el mes en el tiempo, tal como se anticipó en la sección 3.3.2.
- **Modelo conceptual ampliado**

El modelo conceptual ampliado enriquece el modelo de requerimientos con los atributos de cada perspectiva, sus jerarquías completas y las medidas, junto con la granularidad definida. Este modelo, presentado en la Ilustración 15, corresponde al esquema multidimensional que sirve de base para el modelo lógico desarrollado más adelante.

SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS FINTECH MEDIANTE UN DATA WAREHOUSE ESCALABLE

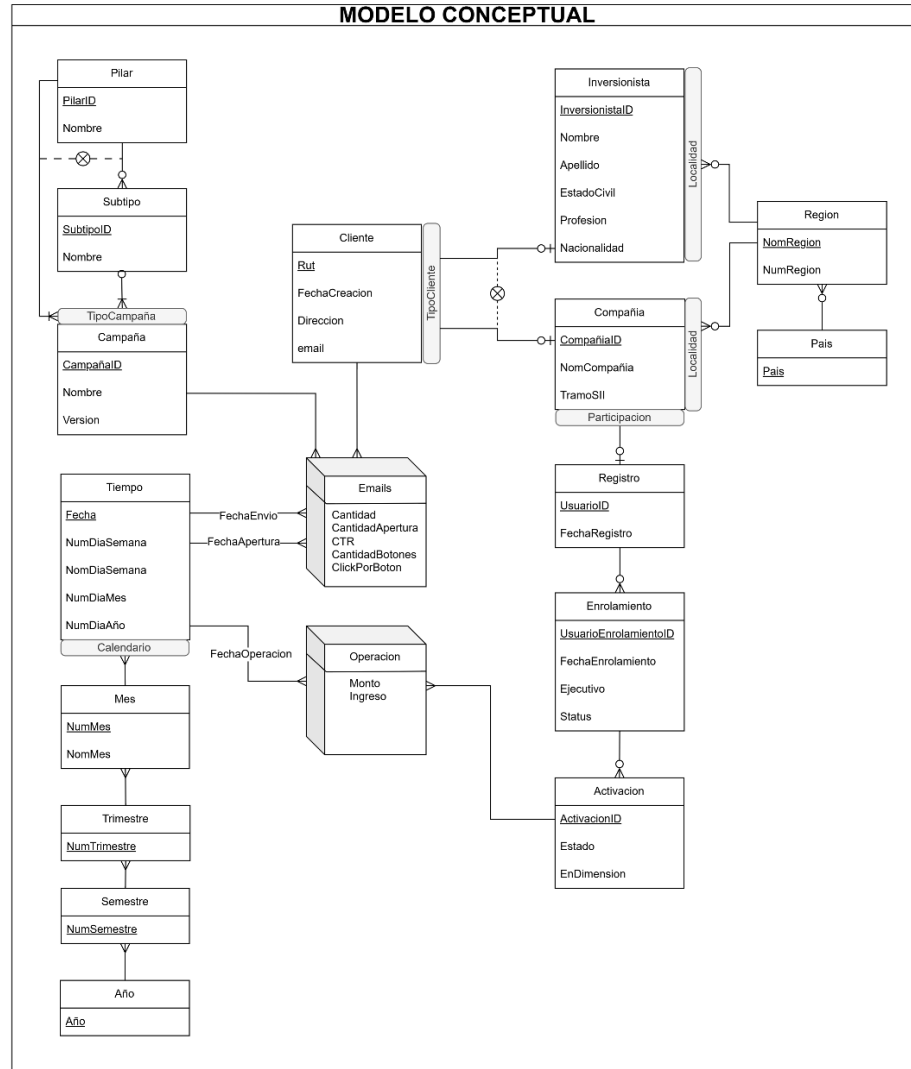


Ilustración 15: Modelo conceptual multidimensional
Fuente: Elaboración propia

El modelo conceptual emplea algunas notaciones que conviene precisar.

- Las cardinalidades se representan igual que en los modelos de bases de datos relacionales.
- Existe una relación de herencia. La localidad es común a las compañías y a los inversionistas, y estos, a su vez, son los padres de la dimensión cliente. Por tratarse de una herencia exclusiva, un cliente puede tomar los atributos de un inversionista o de una compañía, pero no de ambos a la vez.
- De forma análoga, la dimensión *tipoCampaña* hereda atributos del subtipo o directamente del pilar. Esta notación indica que los campos del subtipo son opcionales, mientras que el pilar es obligatorio, de modo que puede haber campañas con pilar pero sin subtipo.

3.5. MODELO LÓGICO DEL DATA WAREHOUSE

A partir del modelo conceptual ampliado de la sección anterior, se deriva el modelo lógico, presentado en la Ilustración 16, que traduce las perspectivas y los indicadores en tablas concretas implementadas en BigQuery.

El modelo corresponde a un esquema de constelación (también llamado galaxia), ya que está compuesto por dos tablas de hechos, *fact_email_history* y *fact_operacion*, que comparten las dimensiones conformadas *dim_cliente* y *dim_tiempo*. Las dimensiones se modelan de forma desnormalizada, concentrando cada jerarquía en una sola tabla, por lo que cada hecho con sus dimensiones constituye un esquema estrella y el conjunto forma una galaxia de estrellas.

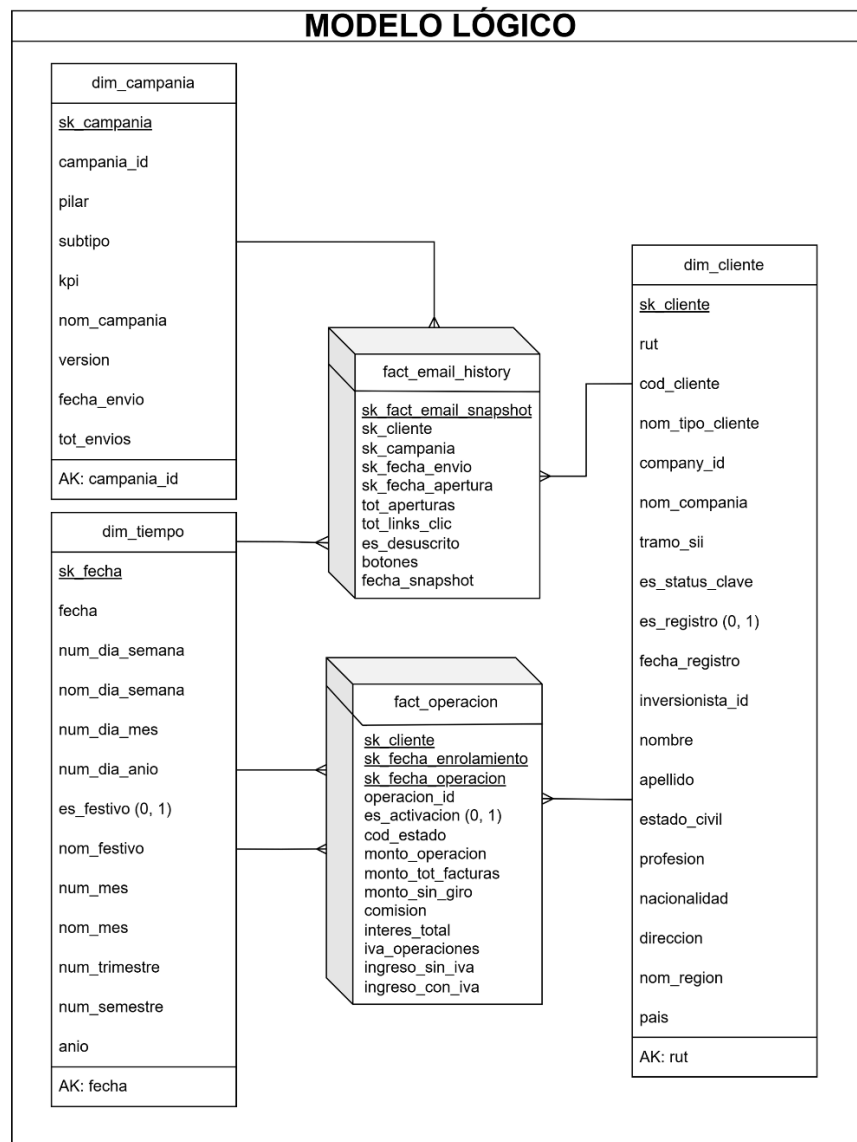


Ilustración 16: Modelo lógico de *Data Mart marketing*

Fuente: Elaboración propia

- **Tablas de dimensión**

Cada dimensión cuenta con una clave subrogada²⁰ (*sk*) como identificador y conserva la clave natural del sistema de origen como clave alternativa²¹ (*AK*), lo que aísla al repositorio de los cambios en las fuentes.

- ***dim_campania***: Identificada por *sk_campania* y con clave alternativa *campania_id*, reúne los atributos descriptivos de cada campaña, como el pilar, el subtipo, los KPI a medir, el nombre y la versión.
- ***dim_tiempo***: Identificada por *sk_fecha* y con clave alternativa fecha, contiene la jerarquía completa del calendario, desde el día hasta el año, junto con atributos de festivos.
- ***dim_cliente***: Identificada por *sk_cliente* y con clave alternativa *RUT*, consolida en una sola tabla los datos del cliente (compañía o inversionista), así como su localidad, región y país.

- **Tablas de hechos**

- ***fact_email_history***: Es una tabla de hechos de tipo *snapshot* periódico, en el sentido descrito por Kimball, ya que en lugar de registrar un único evento captura el estado de cada envío en sucesivas fotografías diarias. Por lo tanto, su granularidad es la combinación de campaña, cliente y fecha de *snapshot*. Cada fila representa el estado acumulado de la interacción de un cliente con una campaña tal como se observaba en una fecha determinada. La *fecha_snapshot* es el atributo que define esta periodicidad y hace que una misma combinación de campaña y cliente aparezca en múltiples filas a lo largo del tiempo. Almacena medidas como el total de aperturas, los clics en enlaces y los botones, además de indicadores como la desuscripción y se relaciona con *dim_cliente*, *dim_campania* y *dim_tiempo*.
- ***fact_operacion***: Registra las operaciones cursadas, con medidas como el monto de la operación, el monto total de facturas, la comisión, el interés y los ingresos con y sin IVA, y se relaciona con *dim_cliente* y *dim_tiempo*.

Las relaciones entre los hechos y las dimensiones se establecen mediante claves foráneas subrogadas. Un caso particular es *dim_tiempo*, que actúa como una dimensión de *role-playing*, ya que cada hecho la referencia más de una vez. El hecho *fact_email_history* la utiliza a través de tres referencias: *sk_fecha_envio*, *sk_fecha_apertura* y *sk_fecha_snapshot*, esta última correspondiente a la fecha de cada fotografía diaria, mientras que *fact_operacion* la utiliza a través de *sk_fecha_enrolamiento* y *sk_fecha_operacion*. Al modelar la fecha de *snapshot* como una referencia más al calendario, la evolución temporal de la interacción puede analizarse con la misma jerarquía de fechas que el resto de los eventos.

Cabe señalar que *fact_operacion* no se relaciona directamente con *dim_campania*, puesto que una operación es un evento de negocio que no registra la campaña que pudo originarla. Por ello, la asociación entre una operación y una campaña no se resuelve mediante una

²⁰ Una clave subrogada (*surrogate key*) es un identificador artificial y único que se asigna a cada fila en una tabla de un Data Warehouse. No posee ningún significado de negocio y su función es relacionar las tablas de hechos con las de dimensiones.

²¹ Una clave alternativa (*alternative key*) es el identificador original proveniente del sistema transaccional.

clave foránea, sino a través de una regla de atribución explícita, materializada como una vista puente en la capa semántica. Esta vista vincula cada operación con una campaña aplicando tres criterios: una ventana temporal, según la cual la operación se atribuye solo si ocurre dentro de los siete días posteriores a la interacción del cliente con la campaña, un tratamiento de múltiples campañas por cliente basado en el criterio de último contacto, que asigna la operación a la campaña con la que el cliente interactuó más recientemente dentro de la ventana y una prevención de doble conteo, que conserva una única atribución por operación, de modo que cada operación aporta a los indicadores una sola vez. Gracias a esta regla, indicadores como el monto de operación atribuido o el número de cesiones de factura por campaña se calculan sobre una base consistente en la que ninguna operación se contabiliza más de una vez.

3.6. ARQUITECTURA DE LA SOLUCIÓN

La solución se sustenta en una arquitectura por capas desplegada sobre un *stack multinube*, presentada de forma general en la Ilustración 13 de la sección 3.1. Esta sección detalla las capas que organizan el flujo de datos, en las cuales reside el modelo definido en la sección anterior, y las tecnologías que las materializan.

3.6.1. ARQUITECTURA POR CAPAS

La información se organiza en capas sucesivas, donde cada una incrementa su nivel de calidad y estructura, en línea con la arquitectura por capas descrita en el marco conceptual. En este proyecto se definieron cinco capas.

- **Capa *raw*:** Almacena los datos tal como llegan desde las fuentes, sin transformación. Recibe la replicación de las fuentes internas mediante Datastream y la copia de los datos externos a través de Cloud Storage, conservando el dato original para asegurar la trazabilidad.
- **Capa *staging*:** Zona transitoria donde se aplican limpiezas y normalizaciones básicas antes de la integración.
- **Capa *silver*:** Contiene los datos ya limpios e integrados, pero sin un contexto específico, lo que permite reutilizarlos en los modelos dimensionales de distintos *Data Marts*. Cada *Data Mart* toma estas tablas y les agrega el contexto según lo que se quiera filtrar o analizar, evitando reprocesar la limpieza y la integración para cada nuevo dominio.
- **Capa *gold*:** Aloja los *Data Marts* con los datos modelados de forma dimensional y agregados para el consumo. Aquí reside el *Data Mart* de *marketing* (*gold_mktng_campania*), que contiene el modelo definido en la sección 3.5, con los datos modelados y agregados para el consumo. Las dimensiones conformadas que se comparten entre los distintos *Data Marts*, actuales y futuros, se centralizan en un

conjunto de datos separado (*gold_shared*), como ocurre con la dimensión tiempo, lo que evita duplicarlas y facilita la incorporación de nuevos *Data Marts*.

- **Capa semántica:** Define de forma centralizada las métricas y los indicadores, entregando definiciones consistentes a las herramientas de *BI*. Esta capa es además relevante para el crecimiento del repositorio, ya que permite calcular nuevos indicadores sobre los datos existentes sin necesidad de incorporarlos en las tablas de hechos. De este modo, es posible medir nuevos *KPIs* a medida que surgen necesidades del negocio, sin rediseñar el modelo ni reprocesar los hechos.

Para revisar el antecedente revisar ANEXO A.

3.6.2. STACK TECNOLÓGICO

El *stack* combina servicios de AWS y de Google Cloud Platform. La Tabla 8 resume cada tecnología, la nube en que opera y el rol que cumple dentro de la arquitectura.

Tabla 8: Stack tecnológico de la solución
Fuente: Elaboración propia

Tecnología	Nube	Rol en la arquitectura
Amazon RDS (MySQL)	AWS	Base de datos operacional. Fuente interna
Datastream	GCP	Replicación <i>CDC</i> de las fuentes internas hacia la capa <i>raw</i>
AWS Glue (Python)	AWS	Extracción y transformación de la fuente externa
Amazon S3	AWS	Zona de aterrizaje de los datos externos en bruto
Cloud Storage	GCP	Respaldo de los datos externos y paso hacia la capa <i>raw</i>
BigQuery	GCP	<i>Data Warehouse</i> que aloja las capas
Dataform	GCP	Transformaciones <i>ELT</i> dentro de BigQuery
Data Studio	GCP	Visualización y construcción de <i>dashboards</i>
IAM, Cloud Monitoring y Cloud Logging	GCP	Seguridad, monitoreo y trazabilidad

3.7. INTEGRACIÓN DE DATOS

La integración de datos materializa los procesos que extraen, transforman y cargan la información desde las fuentes hasta el modelo dimensional, recorriendo las capas de la arquitectura. La solución combina dos enfoques, un *ETL* para la fuente externa y un *ELT* para las internas, y automatiza su ejecución con frecuencias diferenciadas, monitoreo y manejo de errores.

3.7.1. INGESTA DE DATOS

La ingesta se realiza por dos rutas complementarias que reflejan los dos enfoques de integración adoptados. Las fuentes internas se incorporan mediante replicación con captura de datos modificados, mientras que la fuente externa se procesa con un trabajo ETL.

- **Fuentes internas**

Las fuentes internas, alojadas en la base operacional de Amazon RDS con MySQL, se incorporan al repositorio mediante Datastream, el servicio de replicación y captura de datos modificados (*CDC*) de Google Cloud. Datastream lee de forma continua el registro de cambios de la base de datos y propaga hacia BigQuery únicamente las inserciones, actualizaciones y eliminaciones ocurridas desde la última ejecución, lo que reduce el impacto sobre el sistema operacional y mantiene los datos actualizados con una frecuencia horaria. Las tablas replicadas, que contienen la información de clientes, operaciones, registros, enrolamientos y activaciones, aterrizan sin transformar en la capa *raw*, desde donde son procesadas por Dataform siguiendo el enfoque *ELT* descrito en la sección 3.7.2. La elección de Datastream responde a su integración nativa con BigQuery, a la experiencia previa del equipo de *Data* con el ecosistema de Google Cloud y el costo asociado.

- **Fuente externa**

El trabajo de extracción de la fuente externa se implementó en AWS Glue mediante un script en Python que consume la API de Brevo. El proceso comienza recuperando las campañas activas dentro de una ventana de tiempo configurable, lo que permite recapturar campañas recientes y registrar las interacciones que se producen con posterioridad al envío, como las aperturas y los clics que ocurren días después. Esta recaptura es la que alimenta el diseño *snapshot* de *fact_email_history* descrito en la sección 3.5. Para cada campaña, el script solicita la exportación del reporte a nivel de destinatario, espera de forma asíncrona a que el proceso finalice y descarga el archivo resultante y luego enriquece los datos con los contactos de las listas asociadas para incorporar el *RUT* de cada destinatario. Finalmente, transforma los registros a formato *JSON* y los deposita en el *bucket* de Amazon S3, desde donde una copia se traspasa a Cloud Storage y alimenta la capa *raw*, mientras el resultado ya transformado se carga en la capa *silver*.

3.7.2. TRANSFORMACIÓN CON DATAFORM

Una vez que los datos residen en BigQuery, las transformaciones se ejecutan con Dataform siguiendo un enfoque *ELT*. Dataform permite definir las transformaciones en SQL, versionarlas mediante Git y orquestarlas como un conjunto de dependencias, de modo que cada tabla se construye en el orden correcto a partir de las que la preceden.

Las transformaciones hacen avanzar los datos por las capas de la arquitectura. En la capa *staging*, los datos replicados desde las fuentes internas se limpian y normalizan, corrigiendo formatos, tipos y valores inconsistentes. En la capa *silver* convergen los datos internos ya depurados y los datos externos provenientes del proceso *ETL*, donde se integran y se vuelven reutilizables sin un contexto específico de negocio de modo que puedan alimentar los modelos de distintos *Data Marts*. Finalmente, en la capa *gold* se construye el modelo dimensional, generando las tablas de hechos y de dimensiones, asignando las claves subrogadas y materializando las dimensiones conformadas en el conjunto de datos compartido. La capa semántica, por su parte, define los indicadores sobre las tablas resultantes.

Más allá de ejecutar las transformaciones, Dataform aporta capacidades que refuerzan la calidad y la mantenibilidad del proceso. La gestión de dependencias garantiza el orden de ejecución entre las tablas, el versionamiento con Git permite el trabajo colaborativo y la trazabilidad de los cambios, y las aserciones permiten validar reglas de calidad sobre los datos, como la unicidad de las claves o la ausencia de valores nulos, antes de que avancen a las capas superiores. Para revisar el antecedente revisar ANEXO A.

3.7.3. AUTOMATIZACIÓN, FRECUENCIAS Y MONITOREO

Los procesos de ingesta y transformación se ejecutan de forma automática y programada, con frecuencias diferenciadas por fuente que permiten ajustar la actualización de cada una a sus necesidades. Cada proceso se programa de manera independiente con su herramienta de orquestación correspondiente, de modo que la replicación de las fuentes internas, la extracción de la externa y las transformaciones se ejecutan sin intervención manual. La siguiente tabla resume estas frecuencias.

Tabla 9: Procesos de integración y sus frecuencias
Fuente: Elaboración propia

Proceso	Herramienta	Frecuencia
Replicación de las fuentes internas	Datastream	Cada hora
Extracción y carga de la fuente externa	AWS Glue	Diaria, a las 01:00
Respaldo a Cloud storage y carga a la capa Raw	Amazon S3 y Cloud Storage	Diaria, a las 01:30

SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS FINTECH
MEDIANTE UN DATA WAREHOUSE ESCALABLE

Transformación por capas	Dataform	Diaria, posterior a la ingesta
--------------------------	----------	--------------------------------

La operación del sistema se supervisa mediante los servicios de monitoreo de Google Cloud. Cloud Logging registra el detalle de cada ejecución, incluyendo su estado, los tiempos de carga, la cantidad de registros procesados y los errores ocurridos, mientras que Cloud Monitoring permite vigilar el comportamiento de los procesos y configurar alertas ante fallos o demoras. Por su parte, IAM gestiona los permisos de acceso a los datos y a los servicios, asegurando que cada componente opere con los privilegios mínimos necesarios. Gracias a este esquema, los errores se registran y detectan de forma temprana sin requerir intervención manual, y se mantiene la trazabilidad completa de las cargas, en cumplimiento del objetivo específico de automatización. Para ver antecedente revisar ANEXO A.

3.8. ANÁLISIS Y VISUALIZACIÓN

La última etapa de la solución corresponde al análisis y la visualización, donde la información consolidada en las capas *gold* y semántica se pone a disposición de los usuarios finales. Esta explotación se realiza mediante Data Studio, que se conecta directamente a BigQuery y permite construir *dashboards* interactivos sin requerir conocimientos técnicos avanzados.

Los *dashboards* presentan los indicadores definidos en el análisis de requerimientos, como las aperturas, el *CTR* y las operaciones, organizados según las perspectivas del modelo, esto es, la campaña con su pilar y subtipo, el tiempo y el cliente. De este modo, el área de *marketing* puede visualizar el rendimiento de cada campaña y compararlo a lo largo del tiempo o entre pilares.

La visualización es de autoservicio. Los usuarios pueden filtrar y desglosar la información según sus necesidades, por ejemplo, por período, por pilar o por segmento de cliente, y profundizar desde una vista agregada hacia el detalle de cada campaña, aprovechando las jerarquías definidas en el modelo dimensional.

Con ello se cumple el objetivo de habilitar la reportería automatizada y se aborda el desafío inicial de visibilizar el rendimiento de las campañas. En esta etapa el análisis se sitúa en un nivel descriptivo, mientras que la arquitectura queda preparada para incorporar análisis predictivo en el futuro, tal como se anticipó en el marco conceptual.

SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS FINTECH MEDIANTE UN DATA WAREHOUSE ESCALABLE

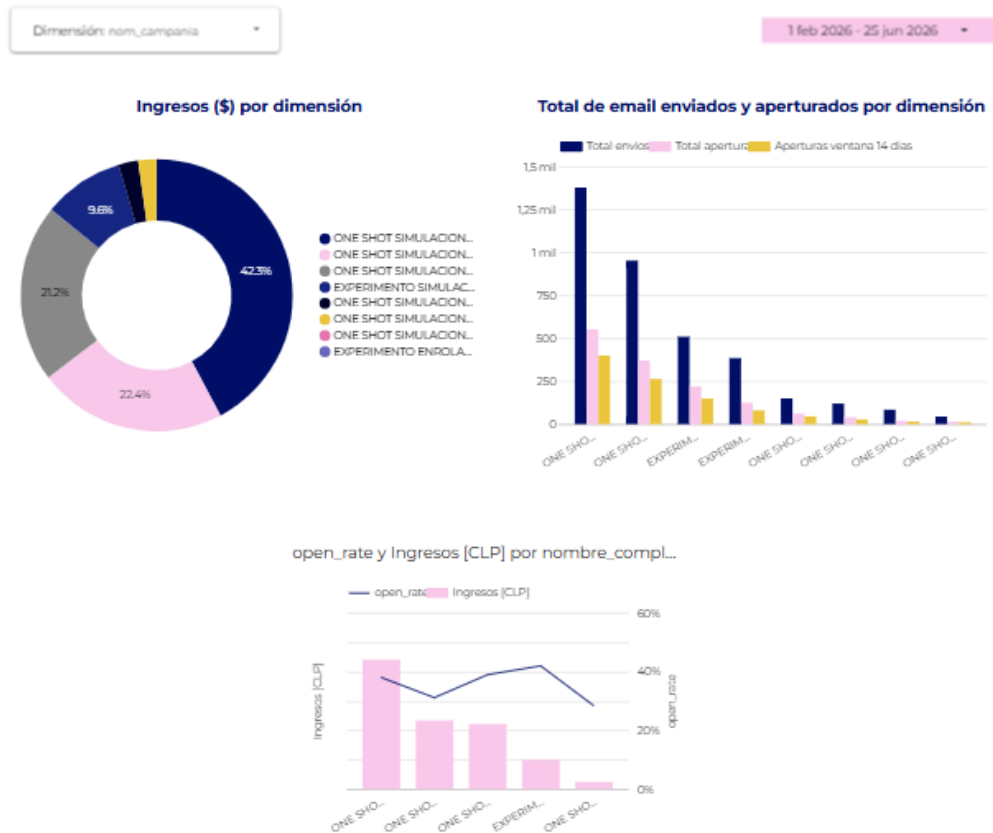


Ilustración 17: Extracto de *dashboard* interactivo
Fuente: Documento de Data Studio de la empresa Alfa

Por razones de privacidad de datos y confidencialidad solo se pueden mostrar los indicadores de la Ilustración 17.

3.9. DOCUMENTACIÓN DE LA SOLUCIÓN

En cumplimiento del tercer objetivo específico, que establece documentar la estructura del Data Warehouse incluyendo flujos de datos, definiciones de tablas, reglas de transformación y un repositorio de consultas frecuentes, se elaboró una documentación integral que cubre cada uno de estos elementos. Esta documentación se mantiene como un documento vivo en Notion y se complementa con la que Dataform genera de forma automática a partir del código de las transformaciones. La responsabilidad de mantenerla actualizada recae en el equipo de Data y se revisa cada vez que se incorporan nuevas fuentes, reglas de transformación o áreas de la compañía al repositorio. La documentación se organiza en los siguientes componentes:

- **Convenciones de nomenclatura:** Las capas, los prefijos de los objetos y la estructura de los nombres de campaña.

1.5.1 Convenciones

Los modelos residen en la capa **staging** y construyen las dimensiones y los hechos del data mart. Las dimensiones llevan el prefijo **dim_**, los hechos **fact_** y las claves subrogadas **sk_**. Cada dimensión conserva su clave natural como clave alternativa (AK).

Ilustración 18: Antecedente de convenciones de transformación

Fuente: Notion de empresa Alfa

- **Flujo de datos o linaje:** El recorrido de cada dato desde la fuente hasta el consumo, apoyado en el grafo de dependencias que Dataform genera automáticamente.

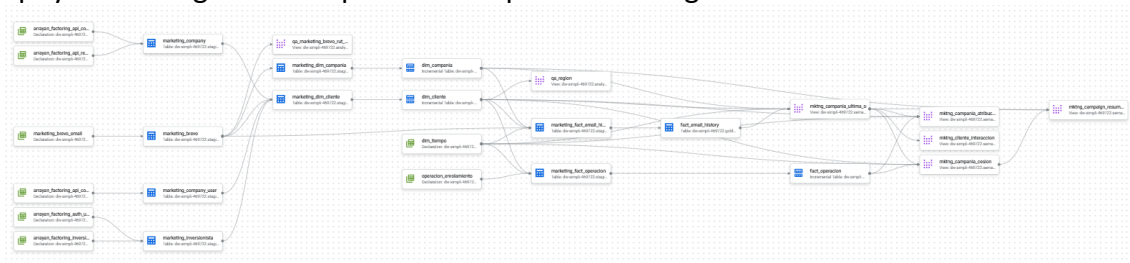


Ilustración 19: Grafo automático de flujo de datos

Fuente: Proyecto de la empresa Alfa en Dataform

- **Catálogo de transformaciones:** El propósito y la lógica de cada modelo en BigQuery.

1.5.1.1 marketing_brevo

- **Propósito:** Dejar los envíos de Brevo limpios y válidos para ingresar al data mart.
- **Lógica:**
 1. Normaliza el RUT pasándolo a mayúsculas y conservando solo dígitos y la letra K.
 2. Descarta los registros con RUT nulo.
 3. Valida que el nombre de campaña cumpla la nomenclatura `{PILAR}_{SUBTIPO}_{KPI}_{NOMBRE}_{V#}` mediante una expresión regular; solo pasan los envíos que la cumplen.
- **Salida:** campaign_id, campaign_name, rut, email, fechas de envío, entrega, apertura y desuscripción, total_opens, clicked_links_count, button y last_update.

Ilustración 20: Extracto de documentación de catálogo de transformación
Fuente: Notion de empresa Alfa

- **Diccionario de datos:** La definición de cada tabla y columna, con su tipo, descripción y origen.

1.6. Diccionario de datos

1.6.1. dim_campania

☐	Nombre del campo	Tipo	Modo	Descripción
☐	sk_campania	INTEGER	NULLABLE	Clave subrogada que no tiene valor útil de negocio. Sirve como identificador único para los registros de esta dimensión
☐	campania_id	INTEGER	NULLABLE	mismo id de campaña de Brevo
☐	pilar	STRING	NULLABLE	Tema o área de la campaña
☐	subtipo	STRING	NULLABLE	segmento de clientes a quien se les envía la campaña
☐	kpi	STRING	NULLABLE	indicadores que se desean medir
☐	nom_campania	STRING	NULLABLE	Nombre de la campaña de Brevo
☐	version	INTEGER	NULLABLE	versionado de campaña para aquellas que se quieran repetir. Caso de las campañas mensuales
☐	fecha_envio	DATE	NULLABLE	Fecha del primer registro de envío
☐	tot_envios	INTEGER	NULLABLE	Cantidad total de correos enviados

Ilustración 21: Extracto de documentación de diccionario de datos
Fuente: Notion de empresa Alfa

- **Repositorio de consultas frecuentes:** Las consultas SQL recurrentes que responden las preguntas de negocio definidas en la sección 3.3.1. Las consultas recurrentes se materializan como vistas en la capa semántica, de modo que los usuarios las consultan directamente sin reescribir la lógica. Las tres vistas presentadas en la Ilustración 22 cubren las preguntas de negocio prioritarias.

1.7. Consultas frecuentes

1.7.1 Resumen de campañas - `mktnng_campaign_resumen`

- **Pregunta que responde:** ¿Cuál es el desempeño agregado de cada campaña según su objetivo?
- **Qué entrega:** Una fila por campaña con sus atributos (pilar, subtipo, KPI y versión) y sus métricas agregadas, como el total de enviados, las aperturas, las aperturas dentro de 14 días, las desuscripciones, los clics y los montos e ingresos de las operaciones asociadas.
- **Consulta:** `SELECT * FROM semantic.mktnng_campaign_resumen`

1.7.2 Atribución por cliente - `mktnng_campania_atribucion_cliente`

- **Pregunta que responde:** ¿Qué campañas derivaron en registros, enrolamientos o activaciones, y qué operaciones generaron?
- **Qué entrega:** Una fila por campaña y cliente, con el contexto del cliente y las banderas de atribución, esto es, registro dentro de 7 días, enrolamiento y activación dentro de 5 días, aplicando las reglas de negocio, además de los montos de las operaciones.
- **Consulta:** `SELECT * FROM semantic.mktnng_campania_atribucion_cliente`

1.7.3 Última snapshot de campañas - `mktnng_campania_ultima_snapshot`

- **Pregunta que responde:** ¿Cuál es el estado más reciente de la interacción de cada envío?
- **Qué entrega:** El snapshot más reciente de cada envío por campaña y cliente, con las métricas de aperturas, clics y desuscripción, junto con el contexto de la campaña y del cliente.
- **Consulta:** `SELECT * FROM semantic.mktnng_campania_ultima_snapshot`

Ilustración 22: Consultas frecuentes
Fuente: Notion

La Tabla 10 resume los componentes de la documentación descritos, indicando su ubicación, el responsable de su mantenimiento, la frecuencia con que se actualizan y la evidencia visual asociada a cada uno.

Tabla 10: Componentes de la documentación de la solución
Fuente: Elaboración propia

Componente documentado	Ubicación	Responsable de mantenimiento	Frecuencia de actualización	Evidencia asociada
Convenciones de nomenclatura	Notion (espacio de la empresa Alfa)	Equipo de Data	Ante cada cambio relevante (nueva fuente, regla o área)	Ilustración 18
Flujo de datos o linaje	Dataform (grafo de dependencias del proyecto)	Automático (Dataform, desde el código)	Automática, en cada compilación del proyecto	Ilustración 19
Catálogo de transformaciones	Notion (espacio de la empresa Alfa)	Equipo de Data	Ante cada cambio relevante (nueva fuente, regla o área)	Ilustración 20
Diccionario de datos	Notion (espacio de la empresa Alfa)	Equipo de Data	Ante cada cambio relevante (nueva fuente, regla o área)	Ilustración 21
Repositorio de consultas frecuentes	Notion + capa semántica en BigQuery	Equipo de Data	Ante cada cambio relevante (nueva fuente, regla o área)	Ilustración 22

Cabe destacar que, mientras los componentes alojados en Notion se mantienen de forma manual por el equipo de Data, el flujo de datos o linaje se regenera automáticamente desde el código en cada compilación del proyecto en Dataform, por lo que permanece siempre sincronizado con el estado real de las transformaciones. De este modo, la documentación combina la mantención de las definiciones de negocio con la actualización automática de la estructura técnica, lo que reduce el riesgo de que quede desactualizada.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

Como se estableció en la sección 2.2.6, la validación de la solución se apoya en el modelo de éxito de DeLone y McLean y se aborda mediante dos mecanismos complementarios: pruebas técnicas del sistema y una comparación cuantitativa de indicadores antes y después de la implementación. Este capítulo aplica ambos mecanismos sobre el dominio de las campañas de *marketing*, que constituye la primera materialización del *Data Warehouse*, y contrasta los resultados con los tres objetivos específicos definidos en la sección 1.3.2.

4.1. CRITERIOS DE VALIDACIÓN

Para orientar la validación, la Tabla 11 mapea cada objetivo específico con el mecanismo de verificación correspondiente y con la dimensión del modelo de DeLone y McLean que lo evalúa.

Tabla 11: Mapeo de objetivos específicos
Fuente: Elaboración propia

Objetivo específico	Mecanismo de verificación	Dimensión de DeLone y McLean
1. Arquitectura modular por capas con trazabilidad y calidad	Prueba técnica: Linaje y aserciones en Dataform	Calidad del sistema Calidad de la información
2. Automatización de procesos con frecuencias, monitoreo y gestión de errores	Prueba técnica: Registros de ejecución en Cloud Logging y Dataform	Calidad del sistema Uso
3. Documentación de la estructura, flujos, transformaciones y consultas	Revisión de cobertura documental	Calidad del servicio

Adicionalmente, los beneficios netos, dimensión final del modelo, se evalúan mediante la comparación del antes y después desarrollada en la sección 4.4.

4.2. VALIDACIÓN TÉCNICA DEL SISTEMA

4.2.1. CONFIABILIDAD DE LOS PROCESOS AUTOMATIZADOS

El objetivo específico 2 establece que los procesos deben ejecutarse de forma automática con frecuencias personalizadas y sin intervención manual. Para verificarlo se analizaron los registros de ejecución de los dos flujos que componen el proceso ELT: el flujo de extracción y carga de AWS Glue y el flujo de transformación de Dataform. Ambos se encuentran en operación estable desde el 12 de febrero de 2026, ejecutándose de forma diaria y automática, lo que a la fecha de esta evaluación acumula cerca de cien ejecuciones consecutivas. Dado que cada flujo se ejecuta bajo una configuración fija, todas las ejecuciones son equivalentes en su comportamiento, por lo que basta una muestra acotada para caracterizarlo. En consecuencia, se analizan las cinco últimas ejecuciones, correspondientes al período del 16 al 20 de julio de 2026, presentadas en la Ilustración 26 e Ilustración 27 (Ver ANEXO A).

Tabla 12: Registro de ejecuciones automáticas del flujo de extracción y carga (16–20 de julio de 2026)

Fuente: Elaboración propia

Fecha	Hora de inicio	Duración	Cantidad de registros procesados	Resultado
20/07/2026	00:00:39	2m 57s	5454	Exitoso
19/07/2026	00:00:39	3m 15s	6008	Exitoso
18/07/2026	00:00:39	3m 36s	6578	Exitoso
17/07/2026	00:00:39	3m 56s	6736	Exitoso
16/07/2026	00:00:39	3m 50s	6736	Exitoso

Observación: La hora difiere de la que indica la Ilustración 13 porque la programación de AWS Glue se rige por la zona horaria del servidor. Debido a ello, durante el horario de verano de Chile la ejecución ocurre a las 01:00 y, durante el horario de invierno, a las 00:00. Por otra parte, la cantidad de registros procesados se repite entre ejecuciones porque cada campaña se extrae durante una ventana de seguimiento de 30 días, por lo tanto, mientras no se incorpore una campaña nueva ni ninguna concluya dicho período, se seguirá procesando la misma cantidad de registros. La coincidencia en la cantidad de registros entre el 16 y el 17 de julio se explica porque en esos días no se incorporó ni concluyó el seguimiento de ninguna campaña, a diferencia de lo que sí ocurre en las jornadas posteriores, donde la cifra varía.

Tabla 13: Registro de ejecuciones automáticas del flujo de transformación (16–20 de julio de 2026)

Fuente: Elaboración propia

Fecha	Hora de inicio	Duración	Cantidad de acciones ejecutadas	Resultado
20/07/2026	03:15:00	29s	21	Exitoso
19/07/2026	03:15:01	28s	21	Exitoso
18/07/2026	03:15:00	29s	21	Exitoso
17/07/2026	03:15:00	28s	21	Exitoso
16/07/2026	03:15:00	32s	21	Exitoso

Los registros de Cloud Logging complementan esta evidencia al dejar constancia automática del estado de cada ejecución y sus tiempos, lo que permite auditar el comportamiento de los flujos sin intervención manual y satisface el requerimiento de monitoreo del objetivo específico 2.

4.2.2 CALIDAD DE LOS DATOS

La calidad de los datos se valida mediante aserciones formales de Dataform, incorporadas como acciones dentro del flujo de transformación. Cada aserción expresa una regla como una consulta que debe retornar cero filas, si la consulta devuelve registros, la regla se considera incumplida y la ejecución del *workflow* se marca como fallida. La Tabla 14 resume las reglas implementadas y la tabla sobre la que operan.

Tabla 14: Reglas de calidad aplicadas mediante aserciones de Dataform

Fuente: Elaboración propia

Tabla evaluada	Regla
<i>dim_campania</i>	Unicidad de la clave subrogada (<i>sk_campania</i>)
<i>dim_cliente</i>	Unicidad de la clave subrogada (<i>sk_cliente</i>)
<i>fact_email_history</i>	Campos obligatorios no nulos (<i>sk_cliente</i> , <i>sk_campania</i> , <i>sk_fecha_envio</i> , <i>fecha_snapshot</i>)
<i>fact_email_history</i>	Integridad referencial (<i>sk_cliente presente en dim_cliente</i>)

De forma complementaria, la calidad se refuerza de manera preventiva en la capa de *staging*, donde las transformaciones limpian y estandarizan los datos antes de que alimenten el modelo dimensional. La Tabla 15 detalla estas reglas.

Tabla 15: Reglas preventivas de calidad aplicadas en la capa de *staging*
Fuente: Elaboración propia

Tabla de <i>staging</i>	Regla preventiva	Propósito
<i>marketing_brevo</i> , <i>marketing_company</i> , <i>marketing_inversionista</i>	Normalización del RUT (solo dígitos y K mayúscula)	Estandariza el identificador para permitir el cruce fiable entre fuentes
<i>marketing_brevo</i> , <i>marketing_company</i> , <i>marketing_inversionista</i>	Descarte de registros con RUT nulo	Evita poblar las dimensiones con registros sin identificador válido
<i>marketing_company</i> , <i>marketing_inversionista</i>	Deduplicación por RUT, conservando el registro más antiguo	Impide la duplicación de clientes provenientes de una misma fuente
<i>marketing_company_user</i>	Deduplicación por <i>company_id</i> , conservando el primer registro	Asegura un único registro de enrolamiento por compañía
<i>marketing_brevo</i>	Validación de la nomenclatura de campaña mediante expresión regular	Solo admite campañas cuyo nombre permite extraer pilar, subtipo, KPI y versión

Cabe precisar el alcance actual de este control. Las aserciones se ejecutan automáticamente y señalan cualquier anomalía marcando la ejecución como fallida, pero no están declaradas como dependencias de las acciones posteriores, por lo que una falla no bloquea por sí sola la escritura de los datos en las capas superiores. Por esta razón, hoy operan como un mecanismo de detección más que como una barrera preventiva.

4.3. VALIDACIÓN FUNCIONAL

4.3.1. COBERTURA DE LAS PREGUNTAS DE NEGOCIO

La validación funcional verifica que el sistema construido sea capaz de responder las preguntas de negocio definidas en la sección 3.3.1. La Tabla 16 mapea cada pregunta con la vista de la capa semántica que la responde y con el indicador o conjunto de indicadores que la sustenta.

Tabla 16: Verificación de respuestas a preguntas de negocio
Fuente: Elaboración propia

<i>Pregunta de negocio</i>	<i>Vista semántica</i>	<i>Indicadores involucrados</i>
¿Cuál es el desempeño de cada campaña según su objetivo?	<i>mktng_campaign_resumen</i>	Aperturas, <i>CTR</i> por campaña, conversiones
¿Cuál es el pilar y subtipo que mejor funciona según la semana del mes?	<i>mktng_campaign_resumen</i>	Aperturas y <i>CTR</i> agrupados por pilar/subtipo y semana
¿Cómo se comporta la interacción al paso de los días por cliente?	<i>mktng_campania_ultima_snapshot</i>	<i>CTR</i> por cliente a lo largo del tiempo (<i>snapshot</i> diario)
¿Qué campañas derivan en registros y cesiones?	<i>mktng_campania_atribucion_cliente</i>	Cantidad de registros, cantidad y monto de operaciones
¿Cuántos clientes nuevos generan las campañas?	<i>mktng_campania_atribucion_cliente</i>	Indicador <i>es_registro</i> con fecha dentro de ventana de atribución

Las cinco preguntas prioritarias de la sección 3.3.1 quedan cubiertas por las tres vistas materializadas en la capa semántica, confirmando que el modelo dimensional diseñado es suficiente para responder las necesidades de información del área de *marketing* en esta etapa.

4.3.2. VALIDACIÓN DEL *DASHBOARD*

El *dashboard* construido en Data Studio, cuyo extracto se presenta en la Ilustración 17, fue revisado con el grupo de *stakeholders*²². La revisión verificó que los valores de los indicadores mostrados en el *dashboard* coinciden con los que el área venía calculando de forma manual, lo que confirma la correcta implementación de las reglas de negocio en las transformaciones.

La revisión se realizó el 12 de febrero de 2026 con la participación de:

- **Franco Cisternas** – *Chief Revenue Officer (CRO)*
- **Germaine Álvarez** – Gerente de *marketing*
- **Matías Yañez** – Líder del área de *Growth*
- **Almendra Tamayo** – Líder del área de datos

Los participantes validaron que los valores de los indicadores de aperturas, *CTR* y operaciones correspondían a los esperados para el período analizado.

La revisión se llevó a cabo en el marco de una reunión de presentación del proyecto en todas sus etapas, cuyos materiales se incluyen en el Anexo A. Al término de la sesión, los participantes indicaron que el *Data Warehouse* implementado, junto con su arquitectura, procesos de integración y herramientas de visualización, responde a las necesidades de información actuales del área de *marketing*. Asimismo, se presentaron de forma explícita las limitaciones de la etapa actual. La solución cubre únicamente el canal de *email marketing* gestionado a través de Brevo y no integra aún los canales digitales, como Meta Ads, Google analytics y Microsoft Ads. Esta limitación queda registrada como trabajo futuro en la arquitectura diseñada, que se concibió desde su origen para incorporar nuevas fuentes de forma incremental.

²² Un stakeholder es cualquier individuo, grupo u organización que tiene un interés en una empresa, proyecto o decisión corporativa.

4.4. COMPARATIVA ANTES Y DESPUÉS

Para evaluar los beneficios netos de la solución, se compara la situación anterior a la implementación del *Data Warehouse* con la situación posterior, sobre la base de los indicadores descritos en la 2.2.6. Modelos de éxito.

Tabla 17: Comparativa del antes y el después de la implementación del DW
Fuente: Elaboración propia

Indicador	Antes	Después
Fuentes de datos integradas para análisis	0 (sin integración)	2 (base operacional MySQL, Brevo)
Proceso de consolidación de datos para reporte de campañas	Manual, en planillas	Automatizado
Tiempo de ejecución del proceso de actualización	Según el equipo de <i>Data</i> , actualizar los reportes tardaba alrededor de 3 horas	29 segundos aproximadamente (Dataform)
Frecuencia de actualización de los datos	Sin definir. Dependía de disponibilidad del analista	Definida y automatizada por fuente (Tabla 9). La mayor frecuencia es la actualización diaria
Consistencia de los datos entre reportes	Variable. Los reportes podían discrepar	Precisa. Única fuente de la verdad en la capa <i>gold</i>
Documentación del flujo de datos	Ausente	Completa (linaje, diccionario, catálogo, convenciones y reglas de negocio)
Indicadores disponibles en autoservicio para el área de marketing	2 (aperturas, tasa de clics por campaña)	10 (todas las definidas en la sección 3.3.2)
Tiempo para responder una pregunta de negocio	Medio día (consolidación manual)	Segundos (Consulta directa a vista semántica o <i>dashboard</i>)

A modo de ilustración de la fila “Consistencia de los datos” de la Tabla 12, se presenta el caso de la campaña N.º 629, en el que una misma métrica arroja valores distintos según la vista de Brevo que se consulte.

Tabla 18: Inconsistencias de la campaña N.º 629 según la vista consultada en Brevo
Fuente: Elaboración propia

<i>Vista consultada en Brevo</i>	Correos enviados	Clics
<i>Resumen de la campaña</i>	684	42
<i>Lista de contactos</i>	661	-
<i>Desglose de entrega</i>	685 (668 recepciones + 17 <i>soft bounces</i>)	-
<i>Detalle de interacción</i>	-	2

Como muestra la Tabla 18 (Ver antecedentes en ANEXO B), la cantidad de correos enviados varía entre 661, 684 y 685 según se consulte la lista de contactos, el resumen o el desglose de entrega, mientras que los clics descienden de 42 a 2 al pasar del resumen al detalle de interacción. Estas discrepancias se explican por bloqueos de correos, listas de contactos que se agregan y luego se eliminan, actualizaciones en los contactos de las listas, y aperturas y clics generados por bots, entre otros factores.

Esta problemática se resuelve reconociendo que cada métrica es confiable en un momento distinto de su ciclo de vida. La cantidad de destinatarios es confiable desde el primer día del envío, las aperturas alcanzan un valor representativo en una ventana intermedia, pues se estima que dos semanas bastan para capturar el grueso del impacto y los clics, que caracterizan la interacción del cliente con la campaña, se consolidan de principio a fin de forma acumulativa. Con este criterio, el valor que se entrega al usuario como fuente verídica se construye tomando cada indicador en el instante en que resulta estable y consolidándolo en la capa *gold*.

En síntesis, la comparación evidencia beneficios netos en cuatro planos: eficiencia operativa, con la reducción del tiempo de actualización de horas a segundos y la respuesta a preguntas de negocio en segundos. Autonomía del usuario, con el paso de 2 a 10 indicadores en autoservicio. Confiabilidad de la información, mediante su consolidación en una única fuente de la verdad. Y, por último, gobernanza, gracias a la documentación completa del flujo de datos.

4.5 SÍNTESIS SEGÚN EL MODELO DE DELONE & MCLEAN

La Tabla 19 sintetiza los resultados de la validación según las seis dimensiones del modelo de éxito de DeLone y McLean, indicando para cada una la evidencia en que se sustenta y el resultado observado.

Tabla 19: Síntesis de la validación según el modelo de Delone y Mclean
 Fuente: Elaboración propia

Dimensión	Evidencia de evaluación	Resultado
Calidad del sistema	Registros de ejecución de Dataform del 16 al 20 de julio de 2026 (Ver ANEXO A). Cinco ejecuciones consecutivas exitosas, iniciadas a las 03:15 hrs, con duraciones de 28 a 32 segundos y sin intervención manual.	Satisfactorio
Calidad de la información	Coincidencia de los valores de los indicadores con los cálculos manuales previos del área. Control de calidad mediante aserciones de Dataform (unicidad de claves subrogadas, campos obligatorios no nulos e integridad referencial) que se ejecutan automáticamente en cada corrida del flujo, complementadas con reglas preventivas en la capa de <i>staging</i> .	Satisfactorio pero el control reactivo (aserciones no bloqueantes) queda como limitación.
Calidad del servicio	Documentación íntegra y vigente en Notion: Convenciones de nomenclatura, linaje, catálogo de transformaciones, diccionario de datos y repositorio de consultas frecuentes.	Satisfactorio
Uso	<i>Dashboard</i> de autoservicio operativo en Data Studio y tres vistas semánticas disponibles para consulta directa, que cubren las cinco preguntas de negocio prioritarias.	Satisfactorio
Satisfacción del usuario	Revisión del 12 de febrero de 2026 con el equipo de marketing y de datos, que validó los valores de aperturas, <i>CTR</i> y operaciones como correctos para el período analizado.	Satisfactorio
Beneficios netos	Comparativa antes/después (Tabla 17): automatización del proceso de consolidación, reducción del tiempo de actualización de cerca de tres horas a unos 29 segundos y eliminación de las inconsistencias entre reportes.	Satisfactorio

Los resultados de la validación permiten concluir que la solución implementada es válida para el entorno en que fue planteada. Los tres objetivos específicos quedan verificados. La arquitectura por capas demostró ser funcional y trazable, los procesos automatizados operaron de forma confiable durante el período evaluado y la documentación cubre la totalidad de los elementos comprometidos. En conjunto, la solución transforma una situación de dispersión de datos y consolidación manual en un repositorio centralizado, automatizado y documentado, y deja la infraestructura preparada para el análisis predictivo planteado como trabajo futuro.

No obstante, la validación también delimita el estado actual de madurez de la solución. El control de calidad de los datos se ejecuta de forma automática mediante aserciones de Dataform incorporadas al flujo de transformación, que verifican la unicidad de las claves, la ausencia de nulos en los campos obligatorios y la integridad referencial. Sin embargo, estas aserciones no están declaradas como dependencias de las acciones posteriores, por lo que detectan las anomalías y marcan la corrida como fallida, pero no bloquean por sí solas el avance de los datos hacia las capas superiores. Por esta razón operan hoy como un mecanismo de detección reactivo más que como una barrera preventiva, lo que constituye una limitación que se proyecta resolver declarando dichas aserciones como dependencias del flujo de transformación. Asimismo, el alcance funcional se acota al canal de *email marketing* gestionado a través de Brevo y no integra aún los canales de pauta digital. Ambas limitaciones quedan registradas como trabajo futuro, en coherencia con una arquitectura concebida desde su origen para escalar de forma incremental.

CAPÍTULO 5: CONCLUSIONES

El presente trabajo de título abordó el problema de la dispersión de datos en la empresa Alfa, una *fintech* chilena del segmento *factoring* que sustentaba su operación en múltiples fuentes heterogéneas cuya coexistencia generaba inconsistencias en los reportes, dificultaba la trazabilidad de la información y elevaba los tiempos de análisis. Para enfrentar esta situación se diseñó e implementó un *Data Warehouse* escalable sobre una arquitectura en la nube, materializado en una primera etapa sobre el dominio de las campañas de *marketing*. Este capítulo recoge el aprendizaje final del trabajo. En él se discute el cumplimiento de los objetivos planteados, se sintetizan los resultados y aportes obtenidos, se exponen las principales reflexiones surgidas del diseño de la solución, se reconocen sus limitaciones y se plantean lineamientos para su ampliación futura.

5.1 CUMPLIMIENTO DE LOS OBJETIVOS

El objetivo general consistía en diseñar e implementar un *Data Warehouse* escalable en BigQuery que centralizara y estructurara datos provenientes de múltiples fuentes internas y externas, habilitando la reportería automatizada y dejando el sistema preparado para análisis avanzados en etapas futuras. A la luz de los resultados presentados en el Capítulo 4, este objetivo se considera cumplido. La solución integró en un único repositorio analítico la base operacional alojada en Amazon RDS y la plataforma externa de campañas Brevo, dos fuentes que antes operaban de forma aislada. Sobre esa base, organizó los datos en capas de refinamiento progresivo, los puso a disposición de los usuarios mediante *dashboards* de autoservicio y dejó la infraestructura preparada para sumar nuevas fuentes y, en etapas posteriores, capacidades de análisis predictivo.

Respecto de los objetivos específicos, el primero: diseñar una arquitectura modular basada en capas que permitiera escalar el sistema y asegurar la trazabilidad y la calidad de los datos; se cumplió mediante la arquitectura de cinco capas (*raw*, *staging*, *silver*, *gold* y semántica) desplegada sobre un *stack multinube*. La separación en capas demostró ser funcional y trazable, y la materialización de las dimensiones conformadas en un conjunto de datos compartido habilitó la reutilización de componentes entre *Data Marts*, que es la base de la escalabilidad buscada. Cabe señalar que la trazabilidad quedó plenamente verificada a través del linaje generado por Dataform, mientras que el aseguramiento de la calidad, si bien operativo, se sustenta en un esquema de monitoreo reactivo que se discute más adelante como una limitación de la etapa actual.

El segundo objetivo: automatizar los procesos de extracción, transformación y carga, permitiendo frecuencias personalizadas por fuente, el monitoreo de ejecuciones y la gestión de errores sin intervención manual; se verificó a partir de los registros de ejecución del flujo de Dataform durante el período del 16 al 20 de julio de 2026, que mostraron cinco ejecuciones consecutivas completadas con éxito, iniciadas de forma consistente a la hora programada y con duraciones acotadas, sin requerir intervención humana. La adopción de frecuencias diferenciadas por fuente, horaria para la replicación interna mediante

Datastream y diaria para la extracción de la fuente externa mediante AWS Glue, confirmó que el sistema puede ajustar la actualización de cada origen a sus necesidades particulares.

El tercer objetivo: documentar la estructura del Data Warehouse, incluyendo flujos de datos, definiciones de tablas, reglas de transformación y un repositorio de consultas frecuentes; se materializó en una documentación íntegra mantenida en Notion y complementada con la que Dataform genera automáticamente. Esta documentación cubre la totalidad de los elementos comprometidos y aborda de forma directa una de las causas originales del problema, esto es, la escasa documentación de los sistemas existentes.

En conjunto, la formulación verificable de los objetivos facilitó su contraste con mecanismos concretos de validación y la decisión de acotar la primera materialización al dominio de las campañas de *marketing* resultó acertada, ya que permitió concentrar el esfuerzo, demostrar valor de forma temprana y validar la arquitectura antes de extenderla a toda la organización.

5.2 RESULTADOS Y APORTES A LA ORGANIZACIÓN

El impacto de la solución sobre la empresa Alfa se aprecia con claridad al contrastar la situación previa con la posterior a la implementación. Antes del proyecto no existía integración alguna entre fuentes y la consolidación de los datos para el reporte de campañas se realizaba de forma manual sobre planillas, un proceso que, según el equipo de *Data*, tomaba alrededor de tres horas, carecía de una frecuencia definida y producía reportes que podían discrepar entre sí. Tras la implementación, el proceso de consolidación quedó completamente automatizado, su tiempo de ejecución se redujo a unos veintinueve segundos, la actualización pasó a estar programada por fuente y los datos convergen en una única fuente de la verdad en la capa *gold*, lo que eliminó las inconsistencias entre reportes.

El aporte se extiende a los distintos actores involucrados. Los usuarios de nivel gerencial disponen ahora de información confiable y consistente para respaldar sus decisiones, superando las interpretaciones divergentes que generaba la coexistencia de fuentes. El área de *marketing* accede a diez indicadores en modalidad de autoservicio, donde antes no contaba con ninguno, y puede responder una pregunta de negocio en segundos frente al medio día que demandaba la consolidación manual. Por su parte, el equipo de *Data* se beneficia de una documentación completa que reduce los tiempos de aprendizaje y el riesgo de errores, así como la complejidad de incorporar nuevos integrantes.

Más allá del beneficio inmediato sobre el dominio de *marketing*, el principal aporte estructural del trabajo es haber dejado instalada una base reutilizable. La arquitectura por capas, las dimensiones conformadas centralizadas y la documentación viva constituyen un activo que disminuye el costo marginal de incorporar cada nuevo dominio, de modo que el valor de la solución no se agota en la etapa actual, sino que se proyecta sobre el crecimiento analítico futuro de la compañía.

5.3 DISCUSIÓN Y APRENDIZAJE

El desarrollo de la solución dejó aprendizajes que trascienden el caso particular de la empresa Alfa y que conviene explicitar, pues constituye el conocimiento de mayor valor obtenido durante el trabajo.

5.3.1. SOBRE LA ELECCIÓN METODOLÓGICA: KIMBALL VS INMON

Uno de los aprendizajes centrales es que la elección entre los enfoques de construcción de un *Data Warehouse* debe responder a la madurez y la dinámica de la organización, y no únicamente a una preferencia técnica. Para un contexto dinámico y de baja madurez corporativa, como el de la empresa Alfa, el enfoque de Kimball resultó claramente el más adecuado, debido a su rapidez de mantención y escalamiento. Al partir de *Data Marts* orientados a procesos de negocio concretos, este enfoque permitió comenzar por el dominio de *marketing*, entregar valor de forma temprana y crecer de manera incremental hacia otras áreas, sin exigir el modelado normalizado de toda la organización por adelantado, algo para lo que la compañía aún no está preparada.

Por otro lado, El enfoque de Inmon, constituye un gran aporte cuando las reglas de negocio están bien establecidas y existe una sólida gobernanza de datos. Su *Data Warehouse* corporativo centralizado y normalizado garantiza una visión única y consistente desde el diseño, pero a costa de un esfuerzo inicial elevado que solo se justifica cuando la organización posee la estabilidad y la disciplina necesarias para sostenerlo. Aplicar el enfoque de Inmon a un entorno cambiante y de baja madurez habría sido contraproducente, ya que la inversión inicial en un modelo corporativo asume una madurez mayor respecto al que se encuentra la empresa Alfa en este momento. En conclusión, ambos enfoques no compiten, sino que cada uno responde a un momento organizacional distinto.

5.3.2. SOBRE EL ANÁLISIS DE REQUERIMIENTOS EN DATA MARTS ACOTADOS

La aplicación práctica de la metodología HEFESTO permitió constatar que no todos sus artefactos aportan el mismo valor cuando el alcance es reducido. En particular, el modelo conceptual de requerimientos resulta prácticamente innecesario en Data Marts pequeños, ya que su contenido es casi trivial y queda absorbido por el modelo conceptual dimensional, que es más rico y detallado. En estos casos, resulta preferible avanzar directamente hacia el modelo conceptual dimensional, que incorpora los atributos, las jerarquías completas, las cardinalidades y la granularidad, y que sirve de base efectiva para el modelo lógico. Esto no invalida la metodología, sino que confirma que toda metodología debe adaptarse con criterio al tamaño del proyecto, evitando producir material cuyo costo de elaboración no se traduce en un beneficio proporcional.

5.3.3. SOBRE EL ROL DE LAS CLAVES SUBROGADAS

El diseño del modelo dimensional dejó aprendizajes sobre el uso de las claves subrogadas. El primero es que la clave subrogada se incorpora a la tabla de hechos cuando se desea almacenar actualizaciones de los datos a lo largo del tiempo. Esto se observa con claridad al contrastar las dos tablas de hechos del modelo. *fact_operacion* registra eventos discretos y se identifica mediante la combinación de sus claves foráneas, mientras que *fact_email_history*, al ser una tabla de tipo *snapshot* que captura el estado de cada envío día a día, requiere una clave subrogada propia, pues la combinación de campaña, cliente y fecha de envío se repite a través de las sucesivas fotografías y deja de ser un identificador único. Si bien se podría haber hecho una combinación de claves foráneas agregando la fecha de *snapshot* no aporta mucha más información más allá de la misma fecha.

El segundo aprendizaje se refiere a la naturaleza misma de la clave subrogada. Esta debe ser genérica y carente de todo significado de negocio, y no puede construirse a partir de un atributo conformado ni de la clave alternativa del sistema de origen. La razón es estructural, si se pretende conservar el histórico de actualizaciones de un registro, una misma clave natural debe poder aparecer en varias filas y, si una clave subrogada deriva de esa clave natural entonces se duplicaría, rompiendo la unicidad. Es la forma genérica de la clave subrogada la que habilita el almacenamiento de versiones históricas de una misma entidad.

De lo anterior surge una excepción que conviene precisar para no incurrir en contradicción. La dimensión tiempo admite que su clave subrogada se genere con una lógica explícita, por ejemplo, el entero 20260626 para el 26 de junio de 2026. Esto es válido porque el tiempo es inmutable. Una fecha nunca cambia sus atributos ni requiere versiones históricas, de modo que no existe riesgo de duplicación. La regla general sobre que no deben tener un significado aplica a las dimensiones susceptibles de cambiar en el tiempo, mientras que la dimensión tiempo establece la excepción, incluso, es ventajosa, pues facilita el ordenamiento, la partición de las tablas y la legibilidad durante la depuración.

5.3.4. SOBRE LA DOCUMENTACIÓN DE UN DATA WAREHOUSE

Un último aprendizaje es que no existe un estándar único y universalmente aceptado para documentar un *Data Warehouse*, lo que deriva que esta tarea se posponga o se omita. No obstante, es posible definir un conjunto mínimo de componentes que toda documentación debiera contemplar para ser funcional: el flujo de datos o linaje, el catálogo de transformaciones, el diccionario de datos y el repositorio de consultas frecuentes. La relevancia de este conjunto se aprecia al recordar que la escasa documentación fue una de las causas raíz del problema original, por lo que establecer una base mínima no es solo un formalismo, sino a una respuesta directa a una de las dificultades estructurales que motivaron el proyecto.

5.4. LIMITACIONES

La solución desarrollada presenta limitaciones que delimitan su estado de madurez. En primer lugar, el control de calidad de los datos, si bien está automatizado mediante aserciones de Dataform que verifican unicidad, ausencia de nulos e integridad referencial en cada corrida del flujo, no se encuentra configurado para bloquear el avance de la información. Las aserciones no están declaradas como dependencias de las acciones posteriores, por lo que detectan las anomalías y marcan la corrida como fallida, pero no impiden por sí solas que los datos defectuosos se escriban en las capas superiores. En segundo lugar, el alcance funcional se acota al canal de *email* gestionado a través de Brevo y no integra aún los canales digitales, como Meta ads, Google Analytics y Microsoft Ads. En tercer lugar, la primera materialización abarca un único dominio, el de las campañas de *marketing*, por lo que la escalabilidad de la arquitectura, aunque diseñada y fundamentada, queda por demostrarse empíricamente con la incorporación de nuevas áreas. Por lo anterior, el cumplimiento del objetivo general se sitúa en el plano del diseño y la habilitación de la escalabilidad, y no en su comprobación operativa a través de múltiples dominios. Finalmente, el nivel de explotación alcanzado es descriptivo, de modo que las capacidades de análisis predictivo, si bien habilitadas por la infraestructura, no forman parte de la entrega actual.

5.5. TRABAJO FUTURO

A partir de las limitaciones reconocidas, se identifican los siguientes trabajos futuros:

- Declarar las aserciones de calidad ya integradas al flujo como dependencias de las acciones que alimentan, de modo que una falla bloquee el avance de datos defectuosos hacia las capas superiores, evolucionando desde la detección automática hacia un control preventivo que impida el procesamiento de datos que no cumplen las reglas.
- Integrar los canales digitales pendientes, ampliando la cobertura del análisis de *marketing* más allá del correo electrónico.
- Extender la solución a nuevos dominios y áreas de la compañía, aprovechando las dimensiones conformadas ya centralizadas para validar empíricamente la escalabilidad de la arquitectura.
- Reevaluar el enfoque de construcción a medida que la organización gane madurez, considerando que una eventual consolidación de las reglas de negocio y de la gobernanza de datos podría justificar la adopción de elementos del enfoque de Inmon para prevenir la proliferación inconsistente de Data Marts.

5.6. REFLEXIÓN FINAL

El trabajo demuestra que es posible transformar una situación de dispersión de datos y consolidación manual en un repositorio centralizado, automatizado y documentado, mediante una solución técnica y estratégicamente fundamentada y ajustada a la realidad de una organización en crecimiento. Más allá del resultado concreto sobre el dominio de *marketing*, el principal valor del proyecto radica en haber sentado las bases sobre las cuales la empresa Alfa puede construir, de manera incremental, capacidades analíticas cada vez más avanzadas. En un sector *fintech* intensivo en datos y crecientemente competitivo, la capacidad de integrar y explotar información dispersa deja de ser una ventaja accesorio para convertirse en un factor determinante, y este trabajo constituye un primer paso sólido en esa dirección.

REFERENCIAS BIBLIOGRÁFICAS

- CMF. (29 de Mayo de 2026). *¿Qué es el Factoring?* Recuperado el 29 de Mayo de 2026, de CMF EDUCA: <https://www.cmfchile.cl/educa/621/w3-article-27145.html>
- Sinnexus. (2007). *Sinnexus*. Obtenido de Sinnexus: https://sinnexus.com/business_intelligence/
- Verdugo Mendoza, C. J. (Junio de 2024). Un enfoque moderno para proyectos de la feria de software - Analítica Avanzada. *Memoria para optar al título de Ingeniero Civil en Informática*. Valparaíso, Chile.
- Oktaviani, N., & Mauritsius, T. (2024). Data Warehouse in Financial Technology Peer-to-Peer Lending Business: A Literature review. *Journal of Logistics, Informatics, and Service Science*, 11(2), 118-141.
- Silva Peñafiel, G. E., Zapata Yáñez, V. M., Morales Guamán, P. K., & Toaquiza Padilla, L. M. (2019). Análisis de metodologías para desarrollar Data Warehouse aplicado a la toma de decisiones. *Ciencia Digital*, 397-418.
- Linstedt, D., & Olschimke, M. (2015). *Building a Scalable Data Warehouse with Data Vault 2.0*. Morgan Kaufmann.
- Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit. The Definitive Guide to Dimensional Modeling, Third Edition*. Indianapolis: John Wiley & Sons, Inc.
- Vaisman, A., & Zimányi, E. (2014). *Data Warehouse System: Design and Implementation*. Springer.
- Kimball, R., & Caserta, J. (2004). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Canada: Wiley Publishing Inc.
- Rainardi, V. (2008). *Building a Data Warehouse: With Examples in SQL Server*. New York: Apress.
- Nambiar, A., & Mundra, D. (2022). *An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management*. Suiza: MDPI.
- Inmon, W. H. (2005). *Building the Data Warehouse, 4th ed*. Indianapolis: Wiley.
- Kimball, R. R., Becker, B., Mundy, J., & Thornthwaite, W. (2016). *The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence*, (Second Edition ed.). Indianapolis: John Wiley & Sons, Inc.
- Databricks. (s.f.). *¿Cuáles son las mejores herramientas de visualización de datos?* Obtenido de databricks: <https://www.databricks.com/es/blog/data-visualization-tools>

- DataCamp. (27 de Febrero de 2026). *A List of The 24 Best ETL Tools And Why To Choose Them*. Obtenido de DataCamp: https://www.datacamp.com/blog/a-list-of-the-16-best-etl-tools-and-why-to-choose-them?utm_cid=21057859163&utm_aid=157296745417&utm_campaign=230119_1-ps-other~dsa-tofu~all_2-b2c_3-latam-en_4-prc_5-na_6-na_7-le_8-pdsh-go_9-nb-e_10-na_11-na&utm_loc=9048045-&u
- Databricks. (s.f.). *ETL Tools for Fast and Reliable Data Management*. Obtenido de Databricks: <https://www.databricks.com/discover/etl/tools>
- Matia. (2 de Septiembre de 2025). *15 best ETL tools for 2025: A complete guide*. Obtenido de Matia.io: <https://www.matia.io/blog/15-best-etl-tools-for-2025-complete-guide>
- Burdin, R. (16 de Septiembre de 2025). *A List of The 23 Best ETL Tools And Why To Choose Them*. Obtenido de Stacksync: <https://www.stacksync.com/blog/list-23-best-etl-tools-choose-them>
- Kleppmann, M. (2017). *Designing Data-Intensive Applications*. Sebastopol, CA: O'Reilly Media Inc.
- Microsoft. (7 de Mayo de 2026). *What is the medallion lakehouse architecture?* Obtenido de Microsoft learn: <https://learn.microsoft.com/en-us/azure/databricks/lakehouse/medallion>
- Databricks. (s.f.). *What is Medallion Architecture?* Obtenido de Databricks: <https://www.databricks.com/blog/what-is-medallion-architecture>
- DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. *Journal of Management Information Systems*, 19(4), 9-30.
- Matteo Avendaño, A. N., & Cancino del Castillo, C. (2024). *Determinación de factores de éxito en Fintech chilenas*. Santiago: Universidad de Chile.
- Wilkins Binder, J. (2024). *Desarrollo regulatorio de las Fintech*. Biblioteca del Congreso Nacional de Chile.
- Aderemi, S., Olanrewaju, D., Innocent, U., Omamode, H., Tolulope, E., & Odetunde, S. (2024). Big data analytics in the financial services industry: Trends, challenges, and future prospects: A review. *International Journal of Science and Technology Research Archive*, 147-166.
- Chen, B., Guo, D., Xia, J., & Zhang, Z. (2025). The Transformative Role of Artificial Intelligence and Big Data in Banking. *The 24th Annual Bank Research Conference*.
- Soldatos, J., & Kyriazis, D. (2022). *Big Data and Artificial Intelligence in Digital Finance: Increasing Personalization and Trust in Digital Finance using Big Data and AI*. Cham, Switzerland: Springer.

- Salcedo Parra, O. J., Milena Galeano, R., & Rodriguez B., L. G. (Noviembre de 2010). Metodología crisp para la implementación Data Warehouse. *Tecnura*, 14(26), 35-49.
- Bernabeu, D., & García, M. (28 de Septiembre de 2017). Hefesto Data Warehousing. *Guía completa de aplicación teórico-práctico; metodología Data Warehouse*.
- Gardner, S. R. (Septiembre de 1998). Building the Data Warehouse. *Communications of the ACM*, 41(9), 52-80.
- Fregly, C., & Barth, A. (2021). *Data Science on AWS: Implementing End-to-End, Continuous AI and Machine Learning Pipelines*. Sebastopol, CA: O'Reilly Media Inc.

ANEXOS

ANEXO A

NOMENCLATURA ESTANDARIZADA DE CORREOS Y MENSAJES [ACTUALIZADA]

Documento realizado por Germaine Álvarez & Matías Yañez

Documentación revisada y actualizada por Raúl Cuello

1. OBJETIVO

Definir una nomenclatura unificada para todos los envíos de Mail y WhatsApp realizados desde Brevo, con el fin de permitir trazabilidad clara por pilar, subtipo y objetivo, facilitar extracción y modelado de datos, medir KPIs clave y estandarizar el trabajo del equipo.

2. ESTRUCTURA GENERAL DE NOMENCLATURA

Formato:

{PILAR}_{SUBTIPO (opcional)}_{KPI}_{NOMBRE}_{V#}

Observación: las definiciones entre llaves ({ }) se deben reemplazar sin las llaves, pero la definición “subtipo” que va entre corchetes ([]) debe mantener los corchetes que lo rodean. Además, todas las definiciones que contengan espacios entre palabras, estos espacios deben ser reemplazados por un guion (-).

3. PILARES & KPIs OFICIALES

3.1. Pilar: POSIBLES DESACTIVADOS

KPI: Cesión post envío (más apertura y clics)

Ejemplo: POSIBLES-DESACTIVADOS_[]_CESION_factura-en-pausa_V1

3.2. Pilar: FECHAS SENSIBLES

KPI: Cesión post envío

Ejemplo: FECHAS-SENSIBLES_[]_CESION_liquidez-por-iva_V1

3.3. Pilar: PROSPECCIÓN

Subtipos: REGISTRO / ENROLAMIENTO / ACTIVACIÓN

KPI: según subtipo* (Registro, Enrolamiento o Cesión)

Ejemplo: PROSPECCION_[REGISTRO]_KPI-REGISTRO_empieza-con-su-rut_V1

Página 1 de 2

Ilustración 23: Definición de KPI por pilar y subtipo

Fuente: Documento conformado por stakeholders y el autor de este trabajo de título

SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS FINTECH MEDIANTE UN DATA WAREHOUSE ESCALABLE

- ▼ gold_mktng_campania
 - dim_campania
 - dim_cliente
 - fact_email_history
 - fact_operacion
- ▼ gold_shared
 - dim_tiempo
- raw_layer_prod
- semantic
- silver_layer
- staging

Ilustración 24: Antecedente de las capas

Fuente: Proyecto de la empresa Alfa en BigQuery (por razones de confidencialidad no se pueden mostrar las tablas de todas las capas)

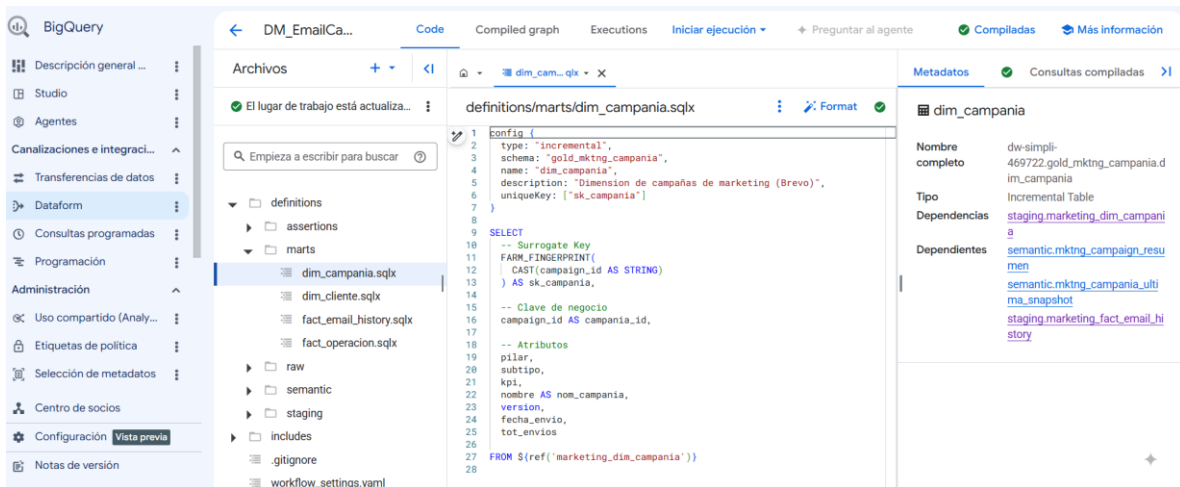


Ilustración 25: Antecedente de transformación en Dataform

Fuente: Proyecto de la empresa Alfa en Dataform

SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS FINTECH MEDIANTE UN DATA WAREHOUSE ESCALABLE

Estado	Hora de inicio	Duración	Tipo de origen	Fuente	Contenido
✓	20 jul 2026, 3:15:00 a.m.	29 segundos	Configuración del flujo de trabajo	dm-mktng-campania-email-breve	Todas las acciones
✓	19 jul 2026, 3:15:01 a.m.	28 segundos	Configuración del flujo de trabajo	dm-mktng-campania-email-breve	Todas las acciones
✓	18 jul 2026, 3:15:00 a.m.	29 segundos	Configuración del flujo de trabajo	dm-mktng-campania-email-breve	Todas las acciones
✓	17 jul 2026, 3:15:00 a.m.	28 segundos	Configuración del flujo de trabajo	dm-mktng-campania-email-breve	Todas las acciones
✓	16 jul 2026, 3:15:00 a.m.	32 segundos	Configuración del flujo de trabajo	dm-mktng-campania-	Todas las acciones

Ilustración 26: Muestra de ejecuciones automáticas de transformación en Dataform
Fuente: Proyecto de la empresa Alfa en Dataform

Run status	Retries	Start time (Local)	End time (Local)	Duration
✓ Succeeded	0	07/20/2026 00:00:39	07/20/2026 00:04:16	2 m 57 s
✓ Succeeded	0	07/19/2026 00:00:39	07/19/2026 00:04:23	3 m 15 s
✓ Succeeded	0	07/18/2026 00:00:39	07/18/2026 00:05:03	3 m 36 s
✓ Succeeded	0	07/17/2026 00:00:39	07/17/2026 00:04:54	3 m 56 s
✓ Succeeded	0	07/16/2026 00:00:39	07/16/2026 00:05:16	3 m 50 s

Ilustración 27: Muestra de ejecuciones automáticas de extracción y carga en AWS Glue
Fuente: Proyecto de la empresa Alfa en AWS Glue

SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS FINTECH MEDIANTE UN DATA WAREHOUSE ESCALABLE

1. Problema de Negocio

- Fuentes Heterogéneas.
- Data sucia.
- Queries muy complejas.
- Falta de autonomía en equipo de Data.
- Monitoreo de costos limitado.
- Falta de visión consolidada de campañas.
- Datos dispersos.
- Dificultad para medir interacción por cliente.
- Dificultad para medir ingresos generados por marketing.



Ilustración 28: Extracto de presentación de proyecto: Problema de negocio
Fuente: Elaboración propia

6. Casos Prácticos

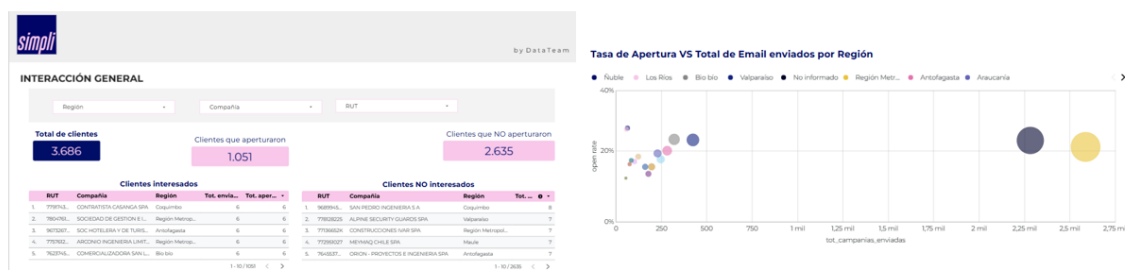


Ilustración 29: Extracto de presentación de proyecto: Caso práctico
Fuente: Elaboración propia

ANEXO B

STAKEHOLDERS_[CLIENTES]_APERTURA_██████-WRAPPED-2025					
Enviada Enviada el 31 dic 2025 15:00 #629	Destinatarios	Aperturas	Clics	Suscripción cancelada	Conversiones
	684	270	42	0	-
	100%	40,48%	6,3%	0%	0%

Ilustración 30: Vista resumen de la campaña N°629
Fuente: CRM Brevo cuenta de la empresa Alfa

SOLUCIÓN A LA DISPERSIÓN DE DATOS EN ENTORNOS FINTECH MEDIANTE UN DATA WAREHOUSE ESCALABLE

Audiencia de campaña		
Listas incluidas		
#308 [redacted] Wrapped 2...	684 contactos	Ver

Ilustración 31: Vista de listas de contactos incluídas en la campaña
Fuente: CRM Brevo cuenta de la empresa Alfa

661 contactos	ⓘ
El recuento de contactos se actualiza cada 5 minutos. Recarga la página para ver el último recuento.	

Ilustración 32: Extracto de vista del detalle de la lista de contactos N°308 asociado a la campaña N°629 que indica el total de 661 contactos
Fuente: CRM Brevo cuenta de la empresa Alfa

Desglose de entregabilidad por listas			
Exportar			
Nombre de la lista	Índice de entrega	Soft bounces	Hard bounces
[redacted] Wrapped 2025 2.0 #308 • 684 contactos - 100% de todos	97,66% 668 contactos	2,49% 17 contactos	0% 0 contacto

Ilustración 33: Desglose de entregabilidad por lista de contactos
Fuente: CRM Brevo cuenta de la empresa Alfa

Desglose de clics por listas		
Los Clics de bots están excluidos. Exportar		
Nombre de la lista	Porcentaje de clics	Total de clics
[redacted] Wrapped 2025 2.0 #308 • 684 contactos - 100% de todos	0,3% 2 clics	0,3% 2 clics

Ilustración 34: Desglose de clics por lista de contactos
Fuente: CRM Brevo cuenta de la empresa Alfa

```
[{"campania_id": "629",  
  "pilar": "STAKEHOLDERS",  
  "subtipo": "CLIENTES",  
  "kpi": "APERTURA",  
  "nom_campania": "[REDACTED] WRAPPED 2025",  
  "version": null,  
  "tot_enviados": "696",  
  "tot_aperturas": "265",  
  "aperturas_14d": "181",  
  "tot_desuscripciones": "0",  
  "tot_clicks": "2",  
  "fecha_envio": "2025-12-30",
```

Ilustración 35: Extracto de resultado de consulta del resumen de la campaña N°629 en formato JSON

Fuente: Resultado en proyecto de la empresa Alfa de BigQuery