

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“DISEÑO E IMPLEMENTACIÓN DE UN DATA MART PARA
APOYAR LA TOMA DE DECISIONES EN UNA EMPRESA
DE COBRANZA EN SU LÍNEA DE NEGOCIO B2C”

JAVIERA PAZ VILLARROEL TOLOZA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Mauricio Figueroa Colarte
Profesor Correferente: José Luis Martí

Octubre - 2024

DEDICATORIA

A mis padres, quienes con su amor, esfuerzo y constante apoyo me han permitido llegar hasta aquí. Los amo.

AGRADECIMIENTOS

Agradezco a mi madre, Mónica, quien a lo largo de mi vida ha luchado por mis estudios, entregándome grandes oportunidades y enseñanzas, pero sobre todo, amor.

A mi padre, Francisco, con quien compartimos muchos gustos en común y quien me mostró el mundo geek, despertando mi interés en la informática.

A mi pareja, Peter, quien me ha apoyado en los buenos y malos momentos, creyendo en mí cuando más lo necesitaba y siempre alentándome con sus palabras y su amor.

A mi perrito, Cholo, por siempre ser tan regalón y divertido, brindándome alegría y cariño en momentos de estrés.

A NetPag, quienes me han recibido en mis dos prácticas y me han permitido realizar mi memoria en la empresa. Gracias a Carolina, mi mentora, quien siempre confió en mí y en mi proceso.

RESUMEN

Resumen— El presente trabajo tuvo como objetivo diseñar e implementar un *data mart* para el área B2C de la empresa de cobranza NetPag de forma que permitiese el acceso y procesamiento en línea de los datos con la finalidad de mejorar los procesos de toma de decisiones y análisis de datos. Para ello, se adoptó la metodología de Kimball, realizando entrevistas con *stakeholders* para comprender los requerimientos del negocio y diseñar un modelo de datos basado en estos. Teniendo en cuenta factores del negocio y las características de los datos, se seleccionó *SQL Server* como base de datos y *Pyspark* como la herramienta principal para los procesos *ETL*.

La implementación del *data mart* resultó en una reducción en los tiempos de respuesta de consultas frecuentes, en comparación con los métodos tradicionales. La consolidación y organización de los datos en el *data mart* permitieron no solo consultas más simples, sino también una integración más efectiva con herramientas de visualización como *Power BI*.

Palabras Clave— Data Mart; Inteligencia de Negocios; Spark; Cobranza; Kimball

ABSTRACT

Abstract— The objective of this project was to design and implement a data mart for the B2C area of the debt collection company NetPag, enabling online access and processing of data to improve decision-making processes and data analysis. The Kimball methodology was adopted, conducting interviews with stakeholders to understand business requirements and design a data model based on them. Considering business factors and data characteristics, *SQL Server* was selected as the database and *Pyspark* as the primary tool for ETL processes.

The implementation of the data mart resulted in a reduction in response times for frequent queries compared to traditional methods. The consolidation and organization of data in the data mart not only allowed for simpler queries but also enabled more effective integration with visualization tools like *Power BI*.

Keywords— Data Mart; Business Intelligence; Spark; Collection; Kimball

GLOSARIO

ACID: Atomicidad, Consistencia, Aislamiento y Durabilidad.

API: Application Programming Interface (Interfaz de programación de aplicación)

Back-end: El back-end son los datos y la infraestructura que permiten que una aplicación funcione.

B2B: Business To Business (Empresa a empresa)

B2C: Business To Consumer (Empresa a consumidor)

B4B: Business For Business (Empresa para empresa)

BI: Business Intelligence (Inteligencia de negocios)

BSON: Binary JSON, es una representación binaria de estructuras de datos y mapas.

CAP: Consistency, Availability and Partition Tolerance (Consistencia, Disponibilidad y Tolerancia a la Partición)

CPU: Unidad central de procesamiento, es un componente de hardware y la unidad computacional central de un servidor.

CQL: Cassandra Query Language

Data Lake: Repositorio centralizado diseñado para almacenar datos en su forma nativa.

Data Mart: Almacén de datos enfocado en una parte del negocio

Data Warehouse (DW): Almacén de datos

Dataset: Conjunto de datos.

DB2: Familia de productos de gestión de datos desarrollados por IBM.

Docker: Plataforma que permite el despliegue de aplicaciones dentro de contenedores de software.

DSO: Days Sales Outstanding (Periodo medio de cobro)

Excel: Programa que permite editar hojas de cálculo desarrollada por Microsoft.

ETL: Extraction, Transformation and Load (Extracción, transformación y carga)

Front-end: El front end es aquello que ven los usuarios e incluye elementos visuales. Permite a los usuarios interactuar con una aplicación.

GB: GigaBytes

HDFS: Hadoop Distributed File System (sistema de ficheros distribuido de Hadoop)

JSON: Formato de texto sencillo para el intercambio de datos.

Kimball: Metodología para gestionar proyectos analíticos creada por Ralph Kimball.

MQL: MongoDB Query Language

MySQL: Sistema de gestión de bases de datos relacional desarrollado bajo licencia dual por Oracle Corporation.

NoSQL: "No Solo SQL"

OLAP: Procesamiento analítica en línea.

Oracle: Sistema de gestión de base de datos relacional, desarrollado por Oracle Corporation.

Power BI: Servicio de análisis de datos de Microsoft para la inteligencia de negocios.

Python: Lenguaje de alto nivel de programación interpretado multiparadigma, caracterizado por la legibilidad de su código.

Pyspark: Interfaz de Apache Spark en Python

SFTP: Secure File Transfer Protocol (Protocolo de transferencia segura de archivos)

SGBD: Sistema gestor de base de datos

Sinacofi: Sistema Nacional de Comunicaciones Financieras.

SMS: Servicio de mensajes cortos.

Software: Programas informáticos que hacen posible la ejecución de tareas específicas dentro de un computador

SQL: Structured Query Language (Lenguaje de consulta estructurado)

SQL Server: Sistema de gestión de base de datos relacional, desarrollado por Microsoft.

Stakeholders: Partes interesadas internas o externas de una empresa

Subtel: Subsecretaría de Telecomunicaciones de Chile.

TI: Tecnología de la Información.

XML: Extensible Markup Language (Lenguaje de marcado extensible)

ÍNDICE DE CONTENIDOS

RESUMEN	4
ABSTRACT	4
GLOSARIO	5
ÍNDICE DE FIGURAS	10
ÍNDICE DE TABLAS	11
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	2
1.1 Contexto de la empresa	2
1.2 Situación actual	3
1.3 Actores involucrados	5
1.3.1 Área de Inteligencia de Negocios y Gobierno de Datos	6
1.3.2 Área de Operaciones	6
1.3.3 Área de Tecnología de la Información	6
1.4 Árbol del problema	6
1.5 Objetivos	9
1.6 Impacto de la solución	10
CAPÍTULO 2: MARCO CONCEPTUAL	11
2.1 Bases de datos	11
2.2 Data Warehouse y Data Mart	12
2.2.1 Modelo multidimensional	14
2.2.2 Arquitectura	16
2.2.3 Capa de <i>data warehouse</i>	17
2.2.4 Capa de OLAP	17
2.2.5 Capa de <i>front-end</i>	18
2.3 Operaciones OLAP	19
2.4 Diseño dimensional	19
2.4.1 Tablas de Hecho	20
2.4.2 Dimensiones	22
2.5 Extracción, transformación y carga (ETL)	24
2.5.1 PySpark	25
2.5.2 Talend	25
2.5.3 SQL Server Integration Services	26
2.6 Dashboard	27
2.7 Metodología Kimball	29
2.7.1 Planificación del proyecto	29
2.7.2 Definición de los requerimientos del negocio	30

2.7.3	Diseño de la arquitectura técnica	30
2.7.4	Modelado multidimensional	31
2.7.5	Diseño de la aplicación de inteligencia de negocios	31
2.7.6	Diseño físico	31
2.7.7	Diseño y desarrollo del ETL	32
2.7.8	Desarrollo de la aplicación de inteligencia de negocios	32
2.7.9	Crecimiento	32
2.7.10	Mantenición	32
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN		33
3.1	Consideraciones	33
3.2	Definición de Requerimientos del Negocio	33
3.3	Diseño de la arquitectura técnica	36
3.3.1	Ecosistema de la empresa	36
3.3.2	Estructura de los datos	36
3.3.3	Consultas y volumen de los datos	37
3.3.4	Selección de producto	37
3.4	Modelado Dimensional	38
3.4.1	Tablas de hechos	38
3.4.2	Dimensiones	47
3.4.3	Tablas de Reporte	55
3.5	Diseño físico	55
3.6	Diseño y desarrollo del ETL	57
3.6.1	Elección de tecnologías	57
3.6.2	ETL	58
3.7	Diseño de la aplicación de inteligencia de negocios	66
3.8	Mantenimiento	66
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		67
4.1	Diferencias entre el sistema antiguo y el data mart	67
4.1.1	Gestiones de llamada según categoría de contacto para el mes de abril del 2024	68
4.1.2	Gestiones de llamada por ejecutivo y categoría de contacto en abril del 2024	70
4.1.3	Recupero versus deuda en los últimos tres meses por mandantes del rubro de autopistas	72
4.2	Validación de los datos	75
4.3	Dashboard para toma de decisiones	77
CAPÍTULO 5: CONCLUSIONES		83
5.1	Trabajos Futuros	85
ANEXOS		88
A	Índexación	88

B	Particionamiento horizontal de tablas	88
C	Vistas materializadas	88
D	Función aplicada para la elección de nombres en Cliente_Deudor	89
E	Consulta SQL Completa	91
F	Acta de Validación	95
REFERENCIAS BIBLIOGRÁFICAS		96

ÍNDICE DE FIGURAS

1	Modelo de negocio B2C extraído de procesos internos. Fuente: (NetPag, 2023)	5
2	Árbol del Problema Fuente: Elaboración Propia	7
3	Reclamo por parte de un cliente extraído del área de Calidad de la empresa. Fuente: (NetPag, 2022)	8
4	Registro del contacto en la base de datos de NetPag. Fuente: (NetPag, 2022)	8
5	Un cubo tridimensional de ventas con dimensiones Tienda, Tiempo y Producto, con la medida cantidad. Fuente: [Malinowski y Zimányi, 2008]	14
6	Arquitectura típica de un <i>data warehouse</i> . Fuente: [Malinowski y Zimányi, 2008]	16
7	Proceso de <i>Talend</i> simple que inserta datos a una base de <i>SQL Server</i> . Fuente: [Training, 2023].	26
8	Dashboard para el análisis de agentes de un <i>call center</i> . Fuente: [Exchange,].	29
9	Ciclo de vida de Kimball. Fuente: [Kimball et al., 2008] y elaboración propia del diagrama traducido.	30
10	Modelado de la tabla de hecho <i>Gestion_Llamada</i> . Fuente: Elaboración propia.	39
11	Modelado de la tabla de hecho <i>Gestion_Bot</i> . Fuente: Elaboración propia.	41
12	Modelado de la tabla de hecho <i>Gestion_Correo</i> . Fuente: Elaboración propia.	42
13	Modelado de la tabla de hecho <i>Gestion_SMS</i> . Se omite algunos datos de <i>Cliente_Deudor</i> que ya se encuentran en figuras anteriores. Fuente: Elaboración propia.	43
14	Modelado de la tabla de hecho <i>Pago</i> . Fuente: Elaboración propia.	44
15	Modelado de la tabla de hecho <i>Cliente_Por_Mandante</i> . Fuente: Elaboración propia.	45
16	Modelado de la tabla de hecho <i>Compra_Correo</i> . Fuente: Elaboración propia.	45
17	Modelado de la tabla de hecho <i>Compra_Teléfono</i> . Fuente: Elaboración propia.	46
18	Modelado de la tabla <i>Bloqueo_Teléfono</i> . Fuente: Elaboración propia.	46
19	Modelado de la tabla <i>Bloqueo_Correo</i> . Fuente: Elaboración propia.	47
20	Modelado de la tabla de dimensión <i>Cliente_Deudor</i> . Fuente: Elaboración propia.	48
21	Modelado de la tabla <i>Deuda</i> . Fuente: Elaboración propia.	49
22	Modelado de la tabla <i>Teléfono_Deudor</i> . Fuente: Elaboración propia.	50
23	Modelado de la tabla <i>Correo_Deudor</i> . Fuente: Elaboración propia.	50
24	Modelado de la tabla <i>Teléfono_Deudor_Mandante</i> . Fuente: Elaboración propia.	51
25	Modelado de la tabla <i>Correo_Deudor_Mandante</i> . Fuente: Elaboración propia.	51
26	Modelado de la tabla <i>Usuario y Dotación</i> . Fuente: Elaboración propia.	52
27	Modelado de la tabla <i>Patente y Patente_Deudor</i> . Fuente: Elaboración propia.	54
28	Tablas del <i>Data Mart</i> utilizadas como reportes. Fuente: Elaboración propia.	56
29	Arquitectura del <i>data mart</i> Fuente: Elaboración propia.	59
30	Flujo de los <i>ETL</i> para el <i>data mart</i> Fuente: Elaboración propia.	60
31	Gestiones por Categoría y Ejecutivo en Abril del 2024.	72
32	Recupero versus Deuda para los periodos Febrero, Marzo y Abril	74

33	<i>Dashboard: Efectividad y Recupero. Fuente: Elaboración propia</i>	78
34	<i>Dashboard: Vista dos de Efectividad y Recupero Fuente: Elaboración propia .</i>	80
35	<i>Dashboard: Contactabilidad Fuente: Elaboración propia</i>	80
36	<i>Acta de Validación del Data Mart. Fuente: Elaboración propia</i>	95

ÍNDICE DE TABLAS

1	Cantidad de Gestiones por Categoría para abril del 2024	69
2	Datos de Mandantes, Periodos, Total Deuda, Total Pago y Porcentaje de Re- cupero	73
3	Comparación de rendimiento entre el sistema antiguo y Data Mart	75
4	Llamadas desde el origen de datos	76
5	Datos de llamadas desde el data mart	77
5	Datos de llamadas desde el data mart (<i>Continuación</i>)	77

INTRODUCCIÓN

En la actualidad, las empresas generan cada vez más datos debido a la velocidad de la operación y la capacidad de almacenar distinta información, enfrentándose al desafío de cómo organizar y aprovechar los datos para beneficiar al negocio.

Este es el caso de NetPag, empresa de cobranza que con distintas fuentes de datos se ha encontrado con la dificultad de cómo acceder y manejar la gran cantidad de datos de la comunicación con deudores en el área de cobranza a personas. Para la empresa, es esencial hacer un buen uso de los datos para el contacto a deudores con el objetivo de lograr la mayor cantidad de recuperación de deudas en relación a sus competidores.

Para solucionar esto, se diseñó e implementó un data mart, es decir, un almacén de datos para el área B2C de la empresa. Utilizando la metodología Kimball, se realizaron entrevistas para definir los requerimientos del negocio en los cuales basar el diseño del data mart, el cual se caracteriza por la importancia de las gestiones. Para la elección de la base de datos, se tuvo en consideración el ecosistema de la empresa, estructura de los datos, consultas frecuentes y volumen de los datos, donde finalmente se escogió *Microsoft SQL Server*. Considerando lo anterior, también fue necesario la elección de las tecnologías a utilizar para la extracción, transformación y limpieza de los datos, seleccionando el *framework Apache Spark* en su interfaz para *Python*, *Pyspark* debido a su simplicidad y potencial para trabajar con grandes volúmenes de datos.

Este documento consta de cinco capítulos, siendo el primero el que define el problema, contextualizando y explicando la situación actual de la empresa para definir los objetivos y el impacto de la solución. Luego, en el marco conceptual se presentan y explican distintos conceptos fundamentales para la comprensión del trabajo realizado tales como definiciones de la metodología Kimball, bases de datos, data warehouses, *ETL* y herramientas, entre otros.

En el capítulo tres, se propone la solución a implementar siguiendo las fases de la metodología Kimball, profundizando en la elección de la base de datos, tecnologías, modelado del data mart y en los *ETL* desarrollados. Posteriormente, en el capítulo cuatro se valida esta solución, comparando tres consultas haciendo uso del data mart versus utilizando el sistema antiguo consultando directamente las bases del proveedor. Además, se presenta un dashboard o tablero creado a partir de los datos del data mart mostrando métricas esenciales para la toma de decisiones en la empresa.

Finalmente, en el capítulo cinco se concluye sobre el trabajo realizado y se presentan trabajos futuros a realizar al data mart.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

1.1. Contexto de la empresa

NetPag es una empresa con 38 años de trayectoria en servicios especializados de recuperación de activos de empresa a empresa, ofreciendo un servicio de cobranza integral y multi-canal, enfocándose en entregar este servicio a otras empresas con un enfoque *B4B*, *Business for Business* a través de un modelo que ayuda a sus clientes a tomar mejores decisiones de negocio.

Sus servicios de cobranza se dividen en tres [NetPag,]:

- **Cobranza Personas B2C:** Cobranza especializada y personalizada dirigida al consumidor final, con herramientas omnicanales y de analítica predictiva junto con altos estándares de calidad y eficacia más un trato digno y justo a los deudores, con una propuesta diferenciada para los tramos de mora preventivo, prejudicial y judicial.
- **Cobranza B2B:** Agiliza la recuperación de las cuentas por cobrar de una empresa, reduciendo los días calle/*DSO* con el fin de rentabilizar el tiempo de la empresa cliente.
- **Cobranza Judicial:** Si no prosperan las acciones prejudiciales, NetPag cuenta con abogados expertos en cobranza a lo largo del país, que ejercen las acciones judiciales.

Estos tres servicios de cobranza también son tres líneas de negocio de la empresa de la que se encargan distintas áreas. La línea de negocio que abarca esta memoria corresponde a la línea de cobranza a personas (*B2C*).

Para cumplir con el enfoque *B4B*, NetPag ofrece el seguimiento en tiempo real de todos los deudores y cuentas por cobrar de la empresa cliente mediante una plataforma en línea las 24 horas, en el cual se podrá consultar múltiples opciones para conocer el estatus de la cartera asignada [NetPag,].

A su vez, hace uso de la tecnología e innovación usando modelos con inteligencia artificial para predecir el comportamiento de las carteras de sus clientes y así resolver al primer contacto.

La estrategia de NetPag busca generar el pago desde el momento que se emite la factura con una metodología y un sistema de trabajo que permite identificar y medir los quiebres que generan mayor impacto en los plazos de pago [NetPag, 2022], para así mejorar los tiempos de cobranza y mejorar las ventas de la empresa al identificar sus clientes de riesgo.

1.2. Situación actual

En NetPag, se realiza la cobranza diariamente según las carteras entregadas por la empresa que contrata los servicios de NetPag, es decir, el listado de deudores segmentado por criterios que define la empresa cliente, donde algunos de ellos pueden ser: Servicio ofrecido, monto y/o tramo de mora. Para acceder a estos listados, los ejecutivos encargados de realizar la cobranza acceden al *software* de cobranza para revisar los datos de los deudores y realizar el contacto según los distintos canales de comunicación que utiliza NetPag:

- Llamado telefónico
- Mensajes de texto
- Correos electrónicos
- Carta certificada a domicilio
- *WhatsApp*
- Visita en terreno

Cada uno de los contactos debe ser registrado en el *software* de cobranza, donde el ejecutivo debe completar el resultado de la gestión, indicando cuál fue la respuesta del deudor.

A su vez, NetPag hace uso de *bots* para contactar a los deudores mediante llamados telefónicos automáticos, mensajes de texto y correos electrónicos masivos.

Todos los contactos realizados deben ser considerados en el cumplimiento de la normativa legal dispuesta por la ley en Chile.

- Ley N° 21320, modificación a la ley N°19496 sobre protección de los derechos de los consumidores, en materia de cobranza extrajudicial y otros derechos del consumidor que plantea los siguientes puntos importantes con respecto a la frecuencia de cobro y datos de cobranza:
 1. Se entenderá que no se da cumplimiento a la ley cuando el proveedor del crédito o la empresa de cobranza efectúe **más de un contacto telefónico o visita por semana**. [MINISTERIO DE ECONOMÍA, 2021]
 2. Se entenderá que no se da cumplimiento a la ley cuando, respecto de otras actuaciones de cobranza extrajudicial realizadas a través de otros medios, tales como correspondencia por correo, mensajes de texto, correos electrónicos o aplicaciones de mensajería instantánea, se realicen **más de dos gestiones por semana**, las que deberán contar con una **separación de, al menos, dos días**. [MINISTERIO DE ECONOMÍA, 2021]

3. Los proveedores o las empresas de cobranza deberán registrar, almacenar y mantener disponible el tipo y frecuencia de las gestiones que realicen por cada deudor por un plazo de **al menos dos años**, contado desde su realización. [MINISTERIO DE ECONOMÍA, 2021]
- Ley N°21142, incorpora en el código del trabajo el contrato de teleoperadores, se encarga de regular el trabajo que se realiza en los *call centers*. [SOCIAL, 2019]

Para realizar la cobranza, cada cliente de NetPag, llamado mandante o cedente, envía los datos correspondientes a sus carteras, con las deudas, pagos y datos de los deudores mediante la transferencia de archivos con el protocolo *SFTP*. Esta información es cargada en el *software* de cobranza, con la importante característica de que todos los mandantes tienen formatos distintos para sus archivos de carga, brindando distintos datos de sus clientes o deudas.

Luego, a partir de estos datos, se diseña la estrategia de cobranza haciendo uso de *Excel*. Se revisan los datos de contacto enviados por el cliente, y en caso de no poseer números de contacto, NetPag compra bases telefónicas.

Una vez teniendo los datos de contacto necesarios, se carga la estrategia de barrido, es decir, el orden de deudores a quién llamar en el discador, que se encarga de marcar automáticamente los números para que los ejecutivos puedan realizar su labor de cobranza.

En cuanto a regulaciones por parte de NetPag sobre la frecuencia de llamados, se consideran **dos intentos de contacto**, los cuales deben realizarse en horarios y/o días distintos, por ejemplo: Si llamo al deudor Juan un lunes por la mañana, no puedo volver a llamarlo treinta minutos después; debo esperar a otro horario (en este caso, tarde o noche), o bien a otro día, para volver a contactarlo. Si ya hubo dos intentos de contacto, se bloquea el número para no seguir contactando hasta la próxima estrategia cargada. Por otro lado, el bloqueo también puede ocurrir en otros casos, por ejemplo, al notar un hostigamiento y/o para cumplir con la ley mencionada anteriormente.

En la figura 1 se presenta un diagrama con el modelo de negocio B2C, donde participa el jefe de negocios, supervisor de cartera y ejecutivo.

Los datos que se encuentran e ingresan en el *software* de cobranza se almacenan en una base de datos del proveedor, mientras que NetPag tiene una base de datos en un servidor local en *Microsoft SQL Server* con procedimientos almacenados que extraen los datos de la base de datos del *software* de cobranza.

Además del proveedor del *software* de cobranza, existen otros proveedores externos para los siguientes canales de contacto:

- Correos electrónicos masivos

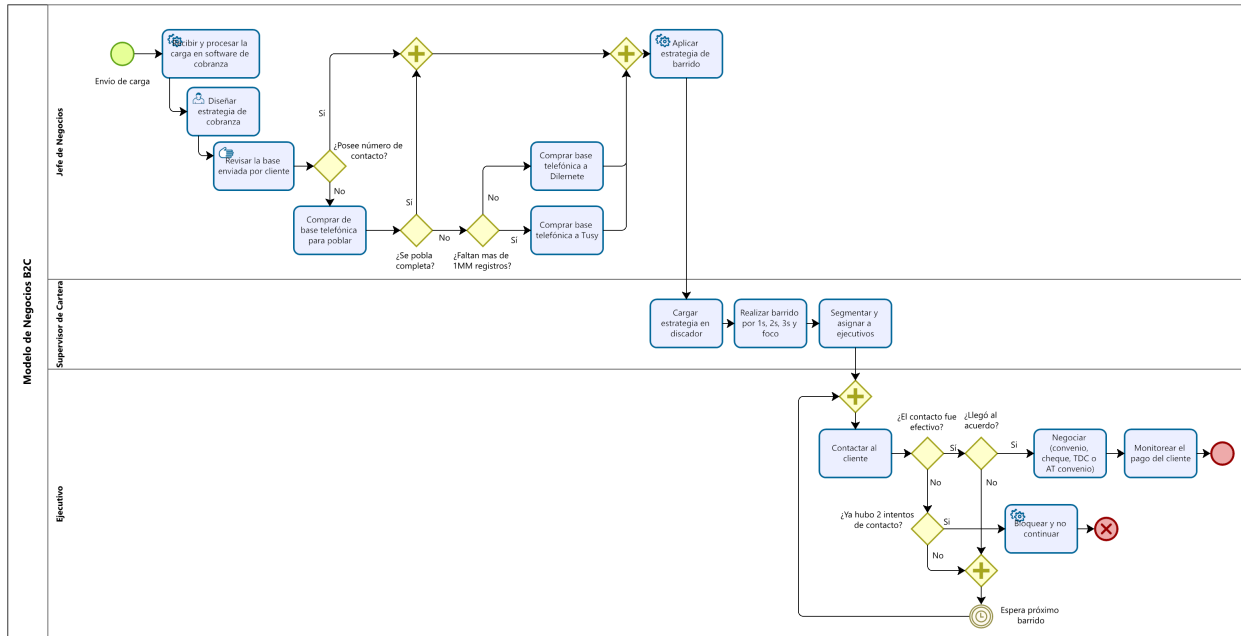


Figura 1: Modelo de negocio B2C extraído de procesos internos.
Fuente: (NetPag, 2023)

- Llamados telefónicos automáticos, o *bots*
- Mensajes de texto

Actualmente, se tienen 20 cedentes. Existen distintos reportes con los cuales NetPag analiza su operación, como también hay reportes que deben ser enviados a los cedentes. Cada reporte o tablero es creado para cada cedente, donde algunos de los más frecuentes son:

- Mejores gestiones según contactabilidad y efectividad
- Rendimiento de ejecutivos en nivel de gestiones y compromisos de pago
- A qué ejecutivo se le asigna el pago según reglas de negocio de cedentes

A su vez, se necesita extraer datos para el entrenamiento y uso de modelos de inteligencia artificial que permitan generar estrategias de contacto más eficientes, por ejemplo, clasificando a los deudores según su preferencia en medio de comunicación.

1.3. Actores involucrados

Al tratarse de datos, varias áreas de la empresa necesitan acceder a ellos para cumplir con sus funciones. Algunas de estas se presentan a continuación.

1.3.1. Área de Inteligencia de Negocios y Gobierno de Datos

El área de Inteligencia de Negocios utiliza *Microsoft Power BI* para la creación y revisión de reportes, conectándose a la base de datos local de NetPag.

En el caso de usar *Excel* usualmente ocurre que los archivos colapsan, por lo que los reportes deben ser limitados a una cierta cantidad de datos o a un cierto rango de fecha.

1.3.2. Área de Operaciones

El área de Operaciones pasa por la misma situación que el área de Inteligencia de Negocios al exportar reportes predefinidos en Excel desde el *software* de cobranza.

Por otro lado, esta área se encarga de estar en contactos con los clientes de NetPag, actualizándolos sobre el estado de la cobranza mediante reuniones donde presentan resultados haciendo uso de sus tableros en *Power BI*.

1.3.3. Área de Tecnología de la Información

El área de TI se ve afectada al ser contactada varias veces por el área de Operaciones, quienes consultan por los problemas tecnológicos que enfrentan al utilizar Excel o *Power BI*, puesto que por la cantidad de datos, ciertos reportes provocan que sus computadores no respondan.

Por otro lado, el área de TI tiene acceso directo a las bases de datos de NetPag, por lo que se encarga de crear vistas en *Microsoft SQL Server* y controlar el espacio que los datos ocupen en los servidores, siendo los encargados de conseguir más servidores y recursos en caso de ser necesario.

Al ser el área encargada de los servidores, se ven afectados al tener que solucionar el problema de la caída de estos debido a consultas exigentes que se realizan a la base de datos.

1.4. Árbol del problema

Las causas del problema son situaciones que se han mencionado en la situación actual de la empresa: tener que realizar exportación manual de los datos para generar los reportes, una descentralización de los datos actualmente al tener múltiples fuentes de información en donde a veces existe incongruencias y los tiempos de respuesta alto para consultas SQL debido al gran volumen de datos de esta área de la empresa.

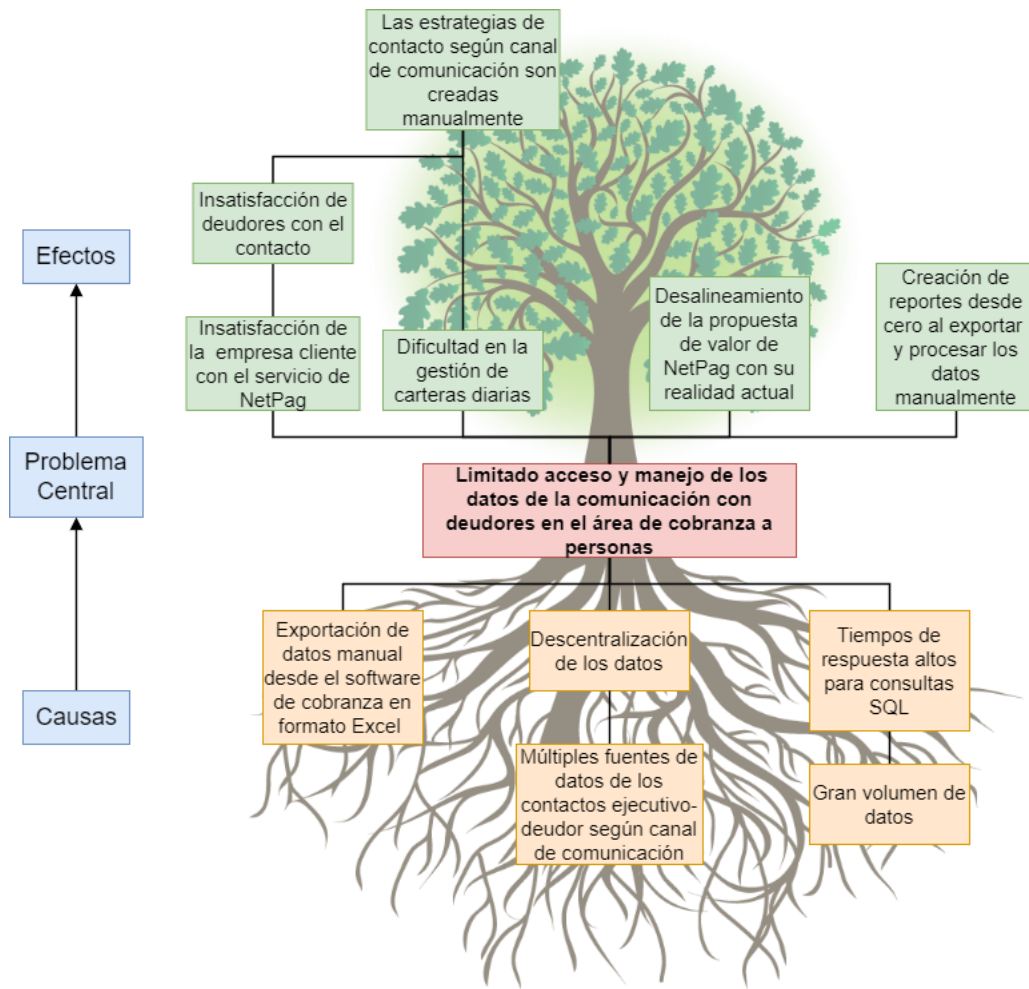


Figura 2: Árbol del Problema
Fuente: Elaboración Propia

Todo aquello conduce al problema central que enfrenta la empresa: un **limitado acceso y manejo de los datos de la comunicación con deudores en el área de cobranza a personas**.

Debido a esto, las estrategias de contacto en la actualidad son creadas manualmente entre el área de Operaciones y el área de Inteligencia de Negocios aplicando formulas simples en *Excel* para conseguir la lista de deudores a contactar cada día y cumplir con la ley, sin embargo, se ignoran varios factores que podrían influir en la creación de mejores estrategias de contacto, como tampoco es posible considerar todos los contactos realizados históricamente debido a la capacidad de *Excel*, provocando que las estrategias actuales no sean eficientes, agregando dificultad en la gestión de carteras diarias, tanto para los ejecutivos como los encargados de cada área que deben velar por cumplir la ley y no hostigar a los deudores.

El no tener buenas estrategias de contacto afecta a los **ejecutivos** que son recompensados si logran generar los pagos, a la **empresa cliente o mandante** que no verá reflejado el pago de sus deudas y a **NetPag** al enfrentar reclamos por parte de deudores, los cuales se dan

principalmente por hostigamiento, y en consecuencia, reclamos por parte de la empresa cliente.

Un ejemplo de aquella situación se presenta a continuación, donde una clienta reclama por el hostigamiento que recibe de la empresa de cobranza al ser contactada múltiples veces por una boleta no vencida. Los reclamos llegan directamente a la empresa cliente de NetPag, por lo que esta empresa contacta a NetPag para conocer la situación.



Figura 3: Reclamo por parte de un cliente extraído del área de Calidad de la empresa.
Fuente: (NetPag, 2022)

A pesar de lo anterior, NetPag solo contiene un registro del contacto, que corresponde a un contacto mediante bot, lo que evidencia un problema con los datos de los contactos que se están almacenando, y al mismo tiempo, una falla en las estrategias de contacto.

NetPag propone el uso de tecnología e innovación con modelos de inteligencia artificial [NetPag,] para conseguir en el menor tiempo posible el pago de las moras por parte de

MORA	ULTIMA_GESTION	FECHA_ULTIMA_GESTION	USUARIO_ULTIMA_GESTION	TELEFONO_ULTIMA_GESTION
NO_VENCIDA	BOT CONTACTO DIRECTO SIN ACUERDO	25-07-2022 15:25	[redacted]	[redacted]

Figura 4: Registro del contacto en la base de datos de NetPag.
Fuente: (NetPag, 2022)

los deudores. Actualmente, NetPag no cumple con aquello, existiendo un desalineamiento de su propuesta de valor con la realidad actual. Aquello afecta directamente la reputación de NetPag, la cual se debe mantener y mejorar para conservar el contrato con sus clientes y aspirar a una mejora continua.

1.5. Objetivos

Se vuelve necesario para NetPag tener una base de datos centralizada y consolidada que facilite el acceso a los datos. Se deben extraer estos datos para que se encuentren accesibles a las distintas áreas de la empresa, pudiendo ver la información mediante consultas y reportes que les permita conocer el estado de la empresa para la toma de decisiones y análisis de datos.

Por ello, se vuelve necesario escoger una base de datos adecuada para la implementación de un *data mart* enfocado en el área de negocio B2C que permita cumplir con sus necesidades y sentar las bases para el análisis de datos permitiendo tener un control sobre la situación de la empresa, así como para el uso de inteligencia artificial a partir de los datos, generando mejores estrategias de contacto con los deudores.

OBJETIVO GENERAL

- Diseñar e implementar un *data mart* que permita el acceso y procesamiento en línea de los datos con la finalidad de mejorar el proceso de toma de decisiones y análisis de datos.

OBJETIVOS ESPECÍFICOS

1. Elegir una base de datos que se ajuste al modelo de negocio de la empresa mediante la investigación de la diversidad de bases de datos existentes.
2. Diseñar la arquitectura y modelo que sostendrá el *data mart* según la base de datos escogida acorde al modelo de negocio y las consultas futuras a realizar.
3. Implementar el diseño propuesto, incluyendo las operaciones de extracción, transformación y limpieza de los datos para garantizar la calidad de estos.
4. Desplegar información del *data mart* mediante un dashboard que apoye la toma de decisiones.
5. Comparar el rendimiento entre el *data mart* propuesto y el antiguo sistema mediante consultas para verificar una mejora en los tiempos de respuesta.

1.6. Impacto de la solución

El método manual que realiza el área de Operaciones para obtener sus reportes utilizando *Excel* y realizando transformaciones y limpieza a grandes cantidades de datos absorbe tiempo con el cual podrían estar analizando los reportes y discutiendo la toma de decisiones.

El acceso y procesamiento de los datos en línea permitirá tener reportes en *Microsoft Power BI* que accedan al *data mart* y reflejen la situación de la empresa, ahorrando el tiempo con el cual exportan y crean nuevos reportes constantemente. Así, podrán revisar información actualizada en el momento que lo deseen y puedan tomar decisiones inmediatas, ya sea referente a la solución de algún inconveniente o mejoras en la estrategia.

Es importante diseñar el modelo de base de datos adecuado con tiempos de respuesta que reflejen este acceso y procesamiento en línea, con el fin de resolver el problema donde actualmente ciertas consultas *SQL* necesarias para la operación resultan demasiado complejas e incluso han provocado la caída del servidor.

Por otro lado, los modelos de inteligencia artificial podrán ser fácilmente alimentados con datos que permitan generar las mejores estrategias de contacto para conseguir el pago de las moras. Con el diseño de base de datos adecuada para el volumen de datos del área, se ahorrará tiempo en la extracción y procesamiento de datos, pudiendo abarcar mayor cantidad de datos históricos de los contactos para conseguir un mejor aprendizaje de los modelos de inteligencia artificial.

Estas estrategias serán seguidas por los **ejecutivos**, por lo que se vuelve más probable que tengan éxito en sus contactos, sintiéndose más satisfechos con su trabajo y recibiendo mayores comisiones.

Al tener los datos para el uso de inteligencia artificial, NetPag cumplirá con su propuesta de valor con respecto a la tecnología e innovación para predecir el comportamiento de las carteras. Las nuevas estrategias de contacto permitirán conseguir el pago de las moras en menor tiempo, lo cual aumentará la confianza de las empresas clientes en NetPag.

Los buenos resultados que se obtengan mejorará la reputación de NetPag, y por lo tanto, aumentará la posibilidad de conseguir nuevos clientes que contraten sus servicios de cobranza.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1. Bases de datos

Las bases de datos constituyen el componente central de los sistemas de información actuales. Una base de datos es una colección compartida de datos lógicamente relacionados, junto con la descripción de los datos, diseñada para cumplir con las necesidades de información y apoyar las actividades de una organización. Una base de datos es desplegada en un sistema gestor de base de datos (SGBD o DBMS en inglés) el cual es un sistema que permite a los usuarios definir, crear, manipular y administrar una base de datos. [Malinowski y Zimányi, 2008]

El diseño de una base de datos suele dividirse en cuatro fases:

1. **Especificación de requisitos**, que recoleta información sobre las necesidades de los usuarios con respecto al sistema de base de datos. Existen distintos enfoques para la realización de esta fase, donde es necesaria la activa participación de los usuarios para incrementar la satisfacción del consumidor con el sistema y evitar errores. [Malinowski y Zimányi, 2008]
2. **Diseño conceptual**, que tiene como objetivo construir una representación de la base de datos que no contenga consideraciones de implementación. Se identifica las entidades relevantes, relaciones y atributos del dominio de la aplicación. Se suele utilizar el modelo entidad-relación u orientado o objetivos basados en notación UML. [Malinowski y Zimányi, 2008]
3. **Diseño lógico**, que tiene como objetivo traducir la representación conceptual de la base de datos a un modelo de implementación (modelo lógico). Los modelos lógicos más comunes son el modelo relacional y el modelo objeto-relacional. A su vez, existen modelos como el orientado a objetos y modelos semi estructurados. Se deben definir reglas de mapeo adecuadas. [Malinowski y Zimányi, 2008]
4. **Diseño físico**, que tiene como objetivo personalizar la representación lógica de la base de datos a una implementación en una plataforma SGBD. Se especifica cómo los registros de la base de datos son guardados, accedidos y relacionados con el fin de asegurar un rendimiento adecuado, lo cual implica analizar los siguientes puntos:
 - Analizar las transacciones o consultas que son ejecutadas frecuentemente y que podrían tener un impacto en el rendimiento
 - Las transacciones que son críticas para las operaciones de la organización
 - Los períodos de tiempo en los que habrá una alta demanda en la base de datos

Algunas plataformas SGBD incluyen a SQL Server, Oracle, DB2 y MySQL. [Malinowski y Zimányi, 2008]

Si bien existen las bases de datos, también están los conceptos de *data warehouses* y *data lakes*, los cuales también se encargan de almacenar datos, sin embargo, se diferencian en lo siguiente:

- Una base de datos almacena datos actuales requeridos para hacer funcionar una aplicación. [MongoDB,]
- Un *data warehouse* almacena datos actuales e históricos de una o más fuentes en un esquema fijo y predefinido que permite realizar análisis de datos por parte de los científicos de datos y analistas. [MongoDB,]
- Un *data lake* almacena datos actuales e históricos de una o más fuentes en su forma bruta que permite realizar análisis de datos por parte de los científicos de datos y analistas. [MongoDB,]

2.2. Data Warehouse y Data Mart

Un *data warehouse* es un almacén de datos que integra datos obtenidos de distintas fuentes. Este repositorio tiene como propósito específico el análisis de datos multidimensional. Dicho de otra forma, es una colección de datos orientados a temas, integrados, no volátiles y variables en el tiempo con el fin de respaldar la toma de decisiones. [Vaisman y Zimányi, 2014] A continuación, se explican estas características

- **Orientado a temas** significa que se enfocan en las necesidades analíticas de distintas áreas de la organización. A diferencia de las bases de datos operacionales, donde el foco se encuentra en las funciones específicas que las aplicaciones deben llevar a cabo.
- **Integrado** significa que los datos que se obtienen, ya sea de sistemas operacionales o externos, se deben unir para agregarlos al *data warehouse*, lo que implica resolver problemas debido a las diferencias en la definición de datos, su contenido, formato, codificación y multiplicidad de ocurrencias de datos.
- **No volátiles** significa que se asegura la durabilidad de los datos al denegar la modificación y eliminado de datos.
- **Variables en el tiempo** indica la posibilidad de retener distintos valores para la misma información, así como el momento en el que ocurrió el cambio.

Un *data warehouse* está hecho para analizar los datos de toda una organización, por lo que es frecuente que departamentos o divisiones de una organización solo requieran una porción de los datos. Por ello, los *data warehouses* por departamento son llamados *data marts*. Un *data warehouse* puede ser visto como una colección de *data marts*.

Existen dos representaciones de *data warehouse*:

- **Bottom-up** donde primero se construyen los *data marts*, y luego estos se unen para obtener el *data warehouse*. [Vaisman y Zimányi, 2014]
- **Top-down** donde los *data marts* se obtienen del *data warehouse* que fue construido en primer lugar. Con este enfoque, a veces un *data mart* es una vista lógica del *data warehouse*. [Vaisman y Zimányi, 2014]

La tabla muestra aspectos en los que se diferencian los *data warehouses* de las bases de datos operacionales.

Los *data warehouses* permiten que los usuarios tomadores de decisiones puedan analizar los datos interactivamente mediante sistemas de procesamiento analítico en línea (**OLAP**). Además, los *data warehouses* son utilizados para reportería, minería de datos y análisis estadístico.

Profundizando en los *data marts*, estos corresponden a un almacén de datos que contiene información específica de la unidad de negocio de una organización. Las empresas suelen utilizar un *data mart* para analizar la información específica de cada departamento de manera más eficiente. [Amazon Web Services,] Al igual que con los *data warehouses*, los datos se obtienen de varios orígenes. Las características de integración, no volátil y variabilidad en el tiempo se mantienen para los *data marts*.

Al tener un *data mart* dedicado a una línea de negocio específica se obtienen varias ventajas, tales como:

- Una fuente única de verdad, puesto que ayuda a que todos los miembros de un departamento tomen decisiones basados en los mismos datos. Así, las partes interesadas pueden centrarse en tomar decisiones y medidas, en lugar de discutir la información misma. [Chile, 2021]
- Acceso más rápido a los datos, ya que una vez establecidas las conexiones con los orígenes de datos deseados, pueden obtener los datos del *data mart* cuando sea necesario. [Chile, 2021]
- Estadísticas más rápidas que permiten una toma de decisiones más rápidas a nivel de departamento. [Chile, 2021]

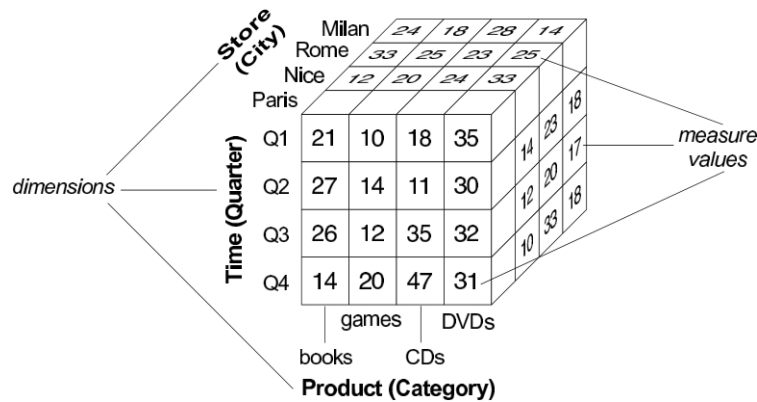


Figura 5: Un cubo tridimensional de ventas con dimensiones Tienda, Tiempo y Producto, con la medida cantidad.

Fuente: [Malinowski y Zimányi, 2008]

- Implementación más sencilla y rápida al centrarse en satisfacer las necesidades de una línea de negocio, lo que requiere acceso a menos fuentes de datos. [Chile, 2021]
- Creación de una gestión de datos ágil y escalable. [Chile, 2021]
- Análisis transitorio al tener la opción de configurar rápidamente un *data mart* para lograr nuevos proyectos. [Chile, 2021]

2.2.1. Modelo multidimensional

Los sistemas de *data warehouses* y OLAP están basados en el modelo multidimensional. Este modelo permite un mejor entendimiento de los datos para análisis y provee un mejor rendimiento para consultas analíticas complejas. El modelo multidimensional visualiza los datos en un espacio n-dimensional, usualmente llamado **cubo de datos** o **hipercubo**.

Un cubo de datos es definido por sus **dimensiones** y **hechos**. Las **dimensiones** son las perspectivas que se utilizan para analizar los datos. Las instancias de una dimensión se denominan **miembros**. Las dimensiones tienen **atributos** que la describen.

Por otro lado, las **celdas** del cubo de datos, o **hechos**, tienen asociados valores numéricos, llamados **medidas**. Las medidas permiten realizar una evaluación cuantitativa de varios aspectos del problema de análisis.

Un cubo de datos puede ser **disperso** o **denso** dependiendo de si tiene medidas asociadas a cada combinación de valores de las dimensiones. En las aplicaciones del mundo real, es común tener cubos dispersos.

Jerarquías

El nivel de detalle con que las medidas son representadas se denomina **granularidad de los datos**. Está determinada por una combinación de granularidades de cada dimensión del cubo. Por ejemplo, en la figura 5 la granularidad de cada dimensión se ve en paréntesis: Ciudad para la dimensión de Tienda, Cuarto para la dimensión de Tiempo, Categoría para la dimensión de Producto. Así, el cubo representa las ventas para las categorías de producto para varias ciudades durante distintos cuartos de tiempo.

Para extraer conocimiento estratégico de los cubos es necesario visualizar los datos en distintas granularidades, en el ejemplo podría ser visualizar a nivel de las tiendas individuales o a una granularidad general como la región o provincia. Las **jerarquías** dan la posibilidad de definir una secuencia de mapeos que relacionan conceptos detallados de bajo nivel con conceptos más generales de nivel superior. Dado dos niveles relacionados en la jerarquía, el nivel más bajo se denomina el **hijo** mientras que el de mayor nivel es el **padre**. [Malinowski y Zimányi, 2008] Un ejemplo de una jerarquía para la dimensión de Tienda se puede observar en la figura:

Existen varios tipos de jerarquías, las cuales se pueden clasificar como:

- **Balanceadas**, puesto que hay el mismo número de niveles desde el nivel más bajo hasta la raíz, tal como se observa en la figura
- **No cubiertas o irregulares**, donde el padre de al menos un miembro no pertenece al nivel inmediatamente superior de ese miembro. Por ejemplo, países de Europa como Luxemburgo o Mónaco no tienen una subdivisión como región o provincia, por lo que el padre de la ciudad Mónaco es el país es el mismo país Mónaco.
- **Desbalanceadas**, al no necesariamente tener instancias en los niveles inferiores. Un ejemplo de esta clasificación ocurriría si en la figura, algunas ciudades no tienen tiendas.

Las herramientas OLAP deben implementar mecanismos de agregación rápidos con el fin de que los usuarios puedan explorar el cubo de datos de forma interactiva a distinta granularidad. La eficiencia de esta agregación depende en el tipo de función utilizada. [Malinowski y Zimányi, 2008]

- Las **medidas distributivas** se definen por una función de agregación que puede ser computada de forma distribuida. Algunas de estas son: contar, sumar, mínimo y máximo.
- Las **medidas algebraicas** se definen por una función de agregación que tiene un número acotado de argumentos, donde cada uno es obtenido aplicando una función de distribución. Un ejemplo es el promedio, que puede ser obtenido dividiendo la suma y la cuenta, siendo las dos funciones distribuidas.

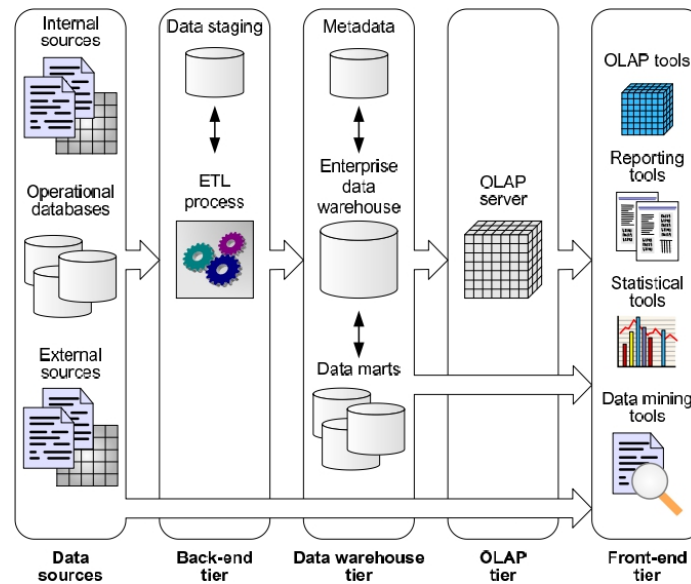


Figura 6: Arquitectura típica de un *data warehouse*.

Fuente: [Malinowski y Zimányi, 2008]

- Las **medidas holísticas** son medidas que no pueden ser obtenidas de otros subagregados, siendo costosas de computar. Algunos ejemplos son la mediana y rango.

2.2.2. Arquitectura

En la figura 6 se presenta una arquitectura típica de un *data warehouse*.

Como se puede notar, la arquitectura tiene distintos niveles:

- El nivel de **back-end** se compone de las herramientas de extracción, transformación y carga, que se encargan de alimentar el *data warehouse* desde las fuentes de datos. Dentro del marco conceptual, se profundiza en este concepto.
- El nivel de *data warehouse* se compone del *data warehouse empresarial* y/o varios *data marts*, junto con un repositorio de *metadata* que almacena información sobre el *data warehouse* y su contenido.
- El nivel de *OLAP* es un servidor *OLAP* que soporta datos multidimensionales y distintas operaciones.
- El nivel de *front-end* se encarga del análisis de datos y la visualización de datos. Contiene herramientas de *OLAP*, reportería, estadísticas y minería de datos.

2.2.3. Capa de *data warehouse*

En 6 se representa un *data warehouse* empresarial junto a varios *data marts*. Un ***data warehouse* empresarial** es un almacén de datos centralizado que engloba todos los departamentos o áreas funcionales en una organización. Por otro lado, un *data mart* es un almacén de datos especializado dirigido hacia un área funcional o a un grupo de usuarios de la empresa. Los datos en un *data mart* pueden ser obtenidos desde el *data warehouse* empresarial o recolectado de distintas fuentes de datos.

Otro componente de esta capa es el repositorio de *meta datos*. Los **meta datos** pueden ser definidos como "los datos sobre los datos". Tradicionalmente, los meta datos se han clasificado entre los técnicos y los comerciales. Los **meta datos comerciales** describen el significado o semántica de los datos, junto a sus reglas y políticas organizacionales, como también las restricciones con respecto a los datos. Por otro lado, los **meta datos técnicos** describen cómo los datos están estructurados y almacenados en el sistema, junto a las aplicaciones y procesos que manipulan los datos.

En el contexto de un *data warehouse*, los meta datos técnicos pueden describir el *data warehouse*, los sistemas fuentes como también el proceso de *ETL*.

- Descripción de la estructura del *data warehouse* y los *data marts*, ambos a nivel lógico (hechos, dimensiones, jerarquías, definiciones de datos, etc) como a nivel físico (índices, particiones y replicas). Además, puede contener información de seguridad (autorización de usuarios y control de acceso) e información de seguimiento (estadísticas de uso, reportes de error y pistas de auditoría)
- Descripción de las fuentes de datos, incluyendo sus esquemas e información descriptiva como propiedad, frecuencias de actualización, limitaciones legales y métodos de acceso.
- Descripción del proceso de *ETL*, incluyendo el linaje de los datos, extracción de los datos, limpieza, reglas de transformación y por defecto, actualización de datos, purga de datos, etc.

2.2.4. Capa de *OLAP*

Existen distintos enfoques para implementar un modelo multidimensional, dependiendo en cómo los datos del cubo estén almacenados. Estos enfoques son:

- Los sistemas ***OLAP* relacionales (ROLAP)** almacenan los datos multidimensionales en bases de datos relacionales, incluyendo características *SQL* y métodos de acceso especiales para implementar eficientemente las operaciones *OLAP*.

Las ventajas de estos sistemas se basa en el hecho de que las bases de datos relacionales se encuentran estandarizadas y proveen una gran capacidad de almacenamiento. Sin embargo, debido a que las operaciones *OLAP* deben ser ejecutadas en tablas relacionales, esto generalmente produce consultas *SQL* complejas. [Vaisman y Zimányi, 2014]

- Los sistemas **OLAP multidimensionales** (*MOLAP*) almacenan los datos en estructuras multidimensionales especializadas, como por ejemplo, en arreglos, combinandolo con técnicas de *hashing* y de indexación. Por ello, las operaciones *OLAP* pueden ser implementadas eficientemente, sin embargo, estos sistemas proveen menor almacenamiento que los sistemas *ROLAP*. [Vaisman y Zimányi, 2014]
- Los sistemas **OLAP híbridos** (*HOLAP*) combinan los dos enfoques anteriores con el objetivo de beneficiarse de la capacidad de almacenamiento de *ROLAP* junto a las capacidades de procesamiento de *MOLAP*. Por ejemplo, un servidor *HOLAP* puede almacenar sus datos en una base de datos relacional, mientras que las agregaciones se mantienen separados en *MOLAP*. [Vaisman y Zimányi, 2014]

Las herramientas de *OLAP* actuales soportan una combinación de los modelos presentados anteriormente, sin embargo, la mayoría se basa en un *data warehouse* implementado en una base de datos relacional.

2.2.5. Capa de *front-end*

Algunas herramientas típicas para que los clientes puedan aprovechar el contenido del *data warehouse* son los siguientes:

- Las **herramientas OLAP** permiten la exploración y manipulación interactiva del *data warehouse* con el fin de encontrar patrones o tendencias de importancia para la organización. Facilitan la formulación de consultas complejas que involucran grandes cantidades de datos. Estas consultas son llamadas **consultas *ad hoc***, ya que el sistema no tiene conocimiento previo de ellas.
- Las **herramientas de reportería** permiten la producción, entrega y manejo de los reportes. Estos usan **consultas predefinidas**, es decir, consultas pidiendo información específica en un formato específico que se realizan de manera regular.
- Las **herramientas estadísticas** son utilizadas para analizar y visualizar el cubo de los datos haciendo uso de métodos estadísticos.
- Las **herramientas de minería de datos** permiten a los usuarios analizar los datos con el fin de descubrir conocimiento valioso tales como patrones y tendencias, así como también predicciones a realizar en base a los datos actuales.

2.3. Operaciones OLAP

Las operaciones OLAP permiten visualizar los datos desde múltiples perspectivas a distintos niveles de detalle.

Un conjunto de operaciones OLAP son las siguientes:

- La operación de **roll-up** resume las medidas. Esto ocurre al moverse en una jerarquía o reducir una dimensión.
- La operación de **drill-down** es contraria a la de *roll-up*, puesto que se mueve de un nivel más general a uno detallado.
- La operación de **pivot** (o rotación) rota los ejes del cubo para proveer una presentación alternativa de los datos.
- La operación de **slice** realiza una selección en una dimensión del cubo, resultando en un subcubo.
- La operación de **dice** realiza una selección en dos o más dimensiones, nuevamente resultando en un subcubo.
- La operación de **drill-across** ejecuta consultas que involucran más de un cubo. Un ejemplo es la comparación de las ventas con las compras en otro cubo. Esta operación requiere que los cubos tengan al menos una dimensión en común.
- La operación de **drill-through** permite navegar desde los datos de nivel superior hasta los datos detallados en los sistemas operacionales de donde se originó el cubo. Un ejemplo de su uso es cuando se intenta identificar la causa de valores atípicos dentro de un cubo de datos.

Además de las operaciones básicas presentadas, las herramientas OLAP proveen una gran variedad de operadores matemáticos, estadísticos y financieros para computar radios, varianzas, medias móviles, rangos, depreciación, etc.

2.4. Diseño dimensional

Las cuatro decisiones clave que se toman durante el diseño de un modelo dimensional incluyen:

1. Seleccionar el proceso de negocio.
2. Declarar el nivel de granularidad.

3. Identificar las dimensiones.
4. Identificar los hechos.

Las respuestas a estas preguntas se determinan al considerar las necesidades del negocio junto con las realidades de los datos durante las sesiones de modelado. Después de declarar el proceso de negocio, la granularidad, las dimensiones y los hechos, el equipo de diseño determina los nombres de las tablas y columnas, los valores de dominio de muestra y las reglas de negocio. [Kimball y Ross, 2013]

2.4.1. Tablas de Hecho

Una tabla de hechos guarda datos numéricos de eventos medidos en el mundo real. Cada fila de la tabla representa un evento específico. Incluye medidas numéricas, claves foráneas que enlazan con las dimensiones asociadas, y a veces claves de dimensiones adicionales y marcas de fecha/hora. Las tablas de hechos son usadas principalmente para cálculos y resúmenes en los informes. [Kimball *et al.*, 2008]

Independientemente de la industria y de los procesos de negocio asociados, todos los eventos de medición se pueden caracterizar como transacciones, instantáneas periódicas o instantáneas acumulativas.

Tablas de Hechos Transaccionales

Las tablas de hechos más básicas y comunes son las transaccionales. La granularidad se especifica como una fila por transacción, o una fila por línea en una transacción. La granularidad transaccional es un punto en el espacio y en el tiempo; las mediciones en un grano transaccional deben ser verdaderas en ese momento específico. [Kimball *et al.*, 2008]

Tablas de Hechos de Instantáneas Periódicas

El segundo tipo más común de tabla de hechos es la instantánea periódica. Con las instantáneas periódicas, se toma un conjunto de “fotos”, como su nombre indica, capturando el rendimiento de un proceso de negocio durante un intervalo de tiempo bien definido y cargando estas fotos en una tabla de hechos. A intervalos predeterminados, como diarios, semanales o mensuales, se toman más fotos al mismo nivel de detalle y se apilan consecutivamente en la tabla de hechos. [Kimball *et al.*, 2008]

Instantáneas Acumulativas

El tercer tipo de tabla de hechos es el menos frecuente, pero cumple una función muy poderosa en ciertas aplicaciones.

A diferencia de las instantáneas periódicas, que almacenan mediciones capturadas regular-

mente, las instantáneas acumulativas se utilizan para representar la actividad a lo largo de un período de tiempo indeterminado para procesos que tienen un principio y un fin bien definidos. Por ejemplo, en una tabla de hechos de instantáneas acumulativas, se carga una fila por cada línea de un pedido cuando se recibe inicialmente. A medida que el pedido avanza en el proceso de cumplimiento, las filas de la tabla de hechos deben reflejar el estado más reciente. Finalmente, la actividad de la línea de pedido se considera completada cuando se genera la factura. [Kimball *et al.*, 2008]

Tablas de Hechos Sin Hechos

A veces, las organizaciones tienen procesos de negocio en los que ocurre un evento, pero no se generan mediciones cuantificables. Estos eventos pueden ser muy importantes para el negocio porque representan la existencia de una relación entre varias dimensiones. Estos eventos sin hechos dan lugar a tablas de hechos sin hechos.

Por ejemplo, una tabla de hechos sin hechos para capturar eventos de asistencia estudiantil. Se carga una fila en la tabla de hechos cada vez que un estudiante asiste a una clase. La granularidad de esta tabla es una fila por cada clase asistida por un estudiante en un día determinado.

Tablas de Hechos con Múltiples Granularidades

Pueden ocurrir casos en donde un solo proceso de negocio produce mediciones a diferentes niveles de detalle. Esto afecta directamente las tablas de hechos, donde por definición los datos deben tener la misma granularidad, sin embargo, existen formas de abordar estos casos.

- **Tabla de datos únicos**

El uso de una sola tabla de hechos con la menor granularidad posible para crear vistas o consultas que agreguen datos en diferentes niveles es un enfoque sencillo y coherente. Sin embargo, este enfoque tiene algunos inconvenientes. Por ejemplo, la tabla de hechos puede volverse grande y lenta de consultar, especialmente si hay muchas dimensiones y atributos. Además, es posible que los usuarios tengan que unir varias tablas y aplicar filtros y cálculos complejos en sus vistas o consultas, lo que puede afectar al rendimiento y la legibilidad. Además, es posible que los usuarios deban especificar la granularidad deseada en sus informes o paneles, lo que lo hace potencialmente confuso o propenso a errores. [LinkedIn, 2024]

- **Varias tablas de hecho**

Un enfoque alternativo consiste en utilizar varias tablas de hechos con diferentes granularidades y, a continuación, vincularlas mediante dimensiones conformadas. Esto tiene la ventaja de la flexibilidad y la escalabilidad, ya que puede personalizar cada tabla de hechos según sus necesidades de análisis. [LinkedIn, 2024]

- **Tabla de hechos híbridos**

Una tercera alternativa es emplear una tabla de hechos híbrida que integre múltiples niveles de granularidad en una sola tabla, utilizando columnas o indicadores especiales para señalar el nivel de detalle. Este enfoque ofrece la ventaja de conveniencia y compatibilidad, ya que permite almacenar y acceder a datos a diferentes niveles desde un único lugar. No obstante, esta solución presenta ciertos desafíos, como la posible aparición de datos dispersos o redundantes debido a que algunas columnas o filas solo son relevantes para determinadas granularidades. Además, puede ser necesario utilizar una clave principal compleja o una clave suplente para asegurar la unicidad e integridad de los datos. Los usuarios también pueden tener que aplicar filtros o condiciones para seleccionar la granularidad adecuada y evitar problemas como el doble conteo o la omisión de datos. [LinkedIn, 2024]

2.4.2. Dimensiones

En contraste con las cualidades simples de las tablas de hechos, que consisten solo en claves y medidas numéricas, las tablas de dimensiones están llenas de campos descriptivos grandes y voluminosos. Los atributos en una tabla de dimensiones tienen dos propósitos críticos: restringir/filtrar consultas y etiquetar los resultados de las consultas. En muchos aspectos, el poder del almacén de datos es proporcional a la calidad y profundidad de los atributos de las dimensiones; dimensiones robustas se traducen en capacidades robustas de consulta y análisis. [Kimball *et al.*, 2008]

Los atributos de dimensión deben ser:

- Verbosos (etiquetas que consisten en palabras completas)
- Descriptivos
- Completos (sin valores faltantes)
- Con valores discretos (cada fila en la tabla de dimensiones tiene solo un valor)
- Asegurados en calidad (sin errores de ortografía, valores imposibles, obsoletos)

Mientras que las tablas de hechos tienen una clave compuesta, las filas de las dimensiones se identifican de manera única por un solo campo clave. Kimball recomienda encarecidamente que las claves primarias de las tablas de dimensiones sean números enteros simples asignados en secuencia comenzando con 1, lo que significa que cuando se crea una clave para un nuevo registro de dimensión, simplemente se suma uno al último valor que se ha utilizado. Estas claves sustitutas no tienen significado puesto que solo sirven como campos de unión entre las tablas de hechos y las de dimensiones. [Kimball *et al.*, 2008]

Las **dimensiones conformadas** estandarizadas son el objetivo de cualquier *data warehouse* bien diseñado. A veces denominadas dimensiones maestras o dimensiones de referencia

comunes, las dimensiones conformadas se comparten en todo el entorno del *data warehouse* de la empresa, uniéndose a múltiples tablas de hechos que representan varios procesos de negocio. Las dimensiones conformadas vienen en dos variantes. Por ejemplo, la dimensión de producto referenciada por la tabla de hechos de pedidos de ventas es idéntica a la dimensión de producto unida a los hechos de inventario. Esta tabla de dimensiones proporciona el mismo contenido, interpretación y presentación, independientemente del proceso de negocio involucrado.

La **dimensión de fecha** ocupa un lugar especial en cada *data warehouse* porque prácticamente cada tabla de hechos es una serie temporal de observaciones; las tablas de hechos siempre tienen una o más dimensiones de fecha.

Además de la dimensión de fecha y la posible dimensión de hora del día, a veces se almacena una marca de tiempo completa en formato SQL en la tabla de hechos para admitir cálculos precisos de intervalos de tiempo entre filas de hechos.

Por definición, los atributos de las tablas de dimensiones cambian relativamente poco. Mientras que los valores en una tabla de hechos pueden cambiar con cada evento de medición, los atributos de las tablas de dimensiones son más estables y estáticos. Sin embargo, muchos de ellos aún están sujetos a cambios, aunque de manera más lenta e impredecible que los hechos. El modelo dimensional necesita rastrear los atributos de dimensión que varían con el tiempo según lo exijan los requisitos del negocio. Existen tres técnicas fundamentales para manejar las dimensiones que cambian lentamente, así como una multitud de enfoques híbridos.

Tipo 1: Sobrescritura del atributo de Dimensión

Cuando el valor de un atributo cambia, el valor antiguo simplemente se actualiza o se sobrescribe con el valor más actual. El atributo de dimensión refleja el estado más reciente, pero se pierden los valores históricos. Esta técnica es especialmente adecuada para procesar correcciones y se utiliza cuando el valor antiguo no tiene relevancia comercial, entendiendo que se pierden las asociaciones históricamente precisas con un tipo 1. En otras palabras, el tipo 1 no preserva el valor del atributo que estaba en vigor en el momento en que se creó la fila de hechos histórica. [Kimball *et al.*, 2008]

Tipo 2: Añadir una Nueva Fila de Dimensión

El tipo 2 es la técnica más popular y poderosa para rastrear con precisión los cambios en los valores de los atributos.

Con un cambio de tipo 2, se inserta una nueva fila con una nueva clave primaria sustituta en la tabla de dimensiones para reflejar los nuevos valores de los atributos. Tanto la fila anterior como la nueva incluyen la clave natural como un atributo, junto con una fecha de inicio de fila, una fecha de expiración de fila y un indicador de fila actual.

La respuesta de tipo 2 se utiliza cuando ha ocurrido un cambio significativo en la dimensión y

es apropiado dividir perfectamente la historia por el atributo cambiado. Cada clave sustituta corresponde a una versión única de la fila de dimensión que fue válida durante un período de tiempo. [Kimball *et al.*, 2008]

Tipo 3: Añadir un Nuevo Atributo de Dimensión

El tipo 3 se utiliza relativamente poco y se emplea generalmente cuando los cambios son menos drásticos y no afectan tanto la forma en que se analiza la historia. En lugar de crear una nueva fila para cada cambio, como en el tipo 2, el tipo 3 añade un nuevo atributo a la misma fila de dimensión. [Kimball *et al.*, 2008]

2.5. Extracción, transformación y carga (ETL)

- La **extracción** reúne los datos desde fuentes múltiples y heterogéneos, los cuales pueden ser de bases de datos operacionales, como también archivos en distintos formatos. Pueden ser internos o externos a la organización. [Malinowski y Zimányi, 2008]
- La **transformación** modifica los datos desde el formato de las fuentes de datos al formato definido en el *data warehouse*. [Malinowski y Zimányi, 2008] Para ello, se consideran los siguientes aspectos:
 - Se debe realizar una **limpieza** de los datos, removiendo errores e inconsistencias con la finalidad de que se encuentren en formato estandarizado.
 - **Integración** de los datos, que concilia los datos de las distintas fuentes a nivel de esquema y a nivel de datos.
 - **Agregación**, que resume los datos obtenidos desde las fuentes de datos acorde al nivel de detalle, o **granularidad**, del *data warehouse*.
- Finalmente, la **carga** que alimenta el *data warehouse* con los datos limpios y transformados. Se incluye el **refrescar** la base de datos, propagando actualizaciones de las fuentes de datos al *data warehouse* a una frecuencia determinada con el objetivo de proveer datos actualizados para el proceso de toma de decisiones. Esta frecuencia puede variar de mensual, varias veces al día o casi en tiempo real dependiendo de los requerimientos de la organización. [Malinowski y Zimányi, 2008]

El proceso de *ETL* suele utilizar un *data staging area*, o área de *staging*, una base de datos en donde se almacenan los datos del proceso de extracción para someterlos a las modificaciones necesarias para ser cargados en el *data warehouse*. [Malinowski y Zimányi, 2008]

Actualmente, existen muchas herramientas para realizar el proceso de *ETL*, ya sean *frameworks* de código abierto o soluciones comerciales robustas. Estas herramientas facilitan la extracción, transformación y carga de datos desde múltiples fuentes a destinos variados, optimizando el proceso de integración y preparación de datos.

2.5.1. PySpark

PySpark es la *API* de Python para Apache Spark. Permite realizar procesamiento de datos a gran escala en tiempo real en un entorno distribuido utilizando Python.

Pero, ¿qué es Apache Spark?

Apache Spark es un sistema de procesamiento distribuido utilizado para realizar tareas de big data y aprendizaje automático en conjuntos de datos grandes.

El procesamiento distribuido es un conjunto en el que se utilizan múltiples procesadores para ejecutar una aplicación. En lugar de intentar procesar grandes conjuntos de datos en una sola computadora, la tarea puede dividirse entre múltiples dispositivos que se comunican entre sí.

PySpark combina la facilidad de aprendizaje y uso de Python con la potencia de Apache Spark para facilitar el procesamiento y análisis de datos de cualquier tamaño para todos los que estén familiarizados con Python.

Para los procesos *ETL* tiene las siguientes ventajas:

- **Rendimiento:** PySpark aprovecha la computación en memoria, haciendo que los procesos *ETL* sean más rápidos que nunca. [Jain, 2023]
- **Facilidad de uso:** Los desarrolladores de Python pueden hacer la transición sin problemas a PySpark debido a su sintaxis. [Jain, 2023]
- **Escalabilidad:** Maneja conjuntos de datos masivos con facilidad gracias al procesamiento distribuido de Spark. [Jain, 2023]
- **Ecosistema:** PySpark se integra con herramientas y bibliotecas populares, lo que lo hace versátil para diversas tareas. [Jain, 2023]

2.5.2. Talend

Talend Open Studio es un software de *ETL* de código abierto en Java que se caracteriza por proporcionar un entorno gráfico en donde crear los procesos de *ETL*. Talend utiliza componentes que pueden ser de lectura, escritura y transformación de datos que se arrastran a la interfaz y se unen entre ellos para crear el proceso.

En la figura 7 se puede observar un ejemplo simple de un proceso en *Talend* que se conecta a una base de *SQL Server* (*tDBConnection_1*) para luego leer un archivo con datos (*tFileInputDelimited_1*) que se insertan en la base de datos de *SQL Server* (*tDbOutput_1*). Si todo resulta bien, se confirma la operación en la base de datos, mientras que si no, se revierte la operación. Finalmente, se cierra la conexión a la base de datos.

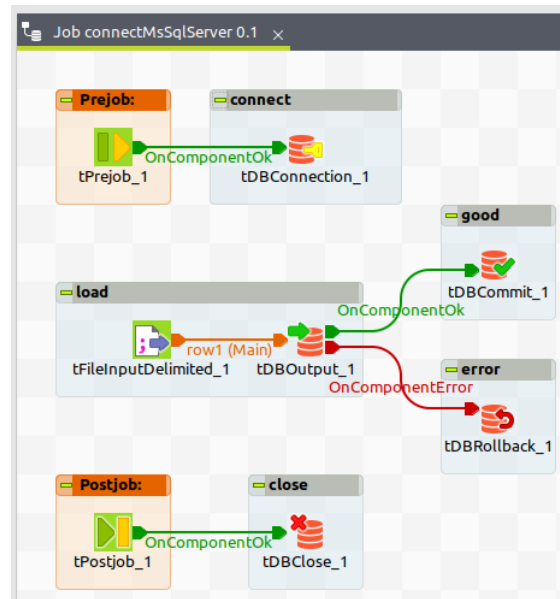


Figura 7: Proceso de *Talend* simple que inserta datos a una base de *SQL Server*.
Fuente: [Training, 2023].

2.5.3. SQL Server Integration Services

SQL Server Integration Services es una plataforma para la creación de soluciones empresariales de transformaciones de datos e integración de datos. [Chugugrace, 2024]

Ventajas

1. Integración con productos Microsoft, *SSIS* al ser parte de *Microsoft*, incluye soporte con *SQL Server* y otros productos. Esto facilita la integración de datos entre diferentes aplicaciones y sistemas *Microsoft*. En el contexto de *NetPag*, esto tiene gran utilidad al integrarse con *SQL Server* y con *SSAS* para los cubos OLAP.
2. Documentación detallada y un soporte extenso, al ser un producto de *Microsoft*.
3. Extensibilidad, al poder ampliar las capacidades de *SSIS* con componentes personalizados, tareas de script y complementos. [Guy, 2024]
4. Transformaciones avanzadas, ya que incluye tareas integradas para la limpieza de datos, agregación, indexación de datos y más. Tiene las funciones más importantes desde la instalación del software, lo que minimiza la necesidad de actualizaciones inmediatas. [Guy, 2024]
5. Integración con Visual Studio. Este entorno de desarrollo integrado (IDE) es familiar para muchos y permite una transición fácil a *SSIS*. [Guy, 2024]

Desventajas

1. Conectores limitados, SSIS puede tener dificultades a la hora de realizar conexiones con bases de datos específicas con las que no tenga soporte. [Airbyte, 2024]
2. Curva de aprendizaje y facilidad de uso, SSIS tiene una curva de aprendizaje elevada y su interfaz de usuario no es tan fácil de usar comparado con otras herramientas. Esto puede provocar ciclos de desarrollo más largos y de mayor complejidad. [Airbyte, 2024]
3. Limitado soporte con desarrollo de terceros, Aunque SSIS admite algunas extensiones de terceros y componentes personalizados, su ecosistema para integración de terceros es menos extenso que el de otras plataformas de integración de datos. Al utilizar desarrollo de terceros, se agrega dificultad al mantenimiento del código. Esto puede ocurrir particularmente cuando se trabaja con ciertos tipos de archivos como *SFTP*, puesto que SSIS no lo soporta de manera nativa. [Airbyte, 2024]
4. Problemas de escalabilidad, SSIS puede enfrentar desafíos para manejar eficientemente grandes volúmenes de datos, especialmente en escenarios de procesamiento distribuido o paralelo. A medida que aumentan los volúmenes de datos, el rendimiento puede degradarse. [Airbyte, 2024]

2.6. Dashboard

Un *dashboard*, también conocido como tablero o cuadro de mando, es una visualización de datos utilizada para monitorear condiciones y/o facilitar la comprensión de los datos [Wexler *et al.*, 2017]. Estos suelen incluir elementos como los indicadores clave de rendimiento (KPI) u otras métricas comerciales importantes que los interesados necesitan ver y entender rápidamente.

Los *dashboards* son útiles en diferentes industrias y sectores porque son altamente personalizables. Pueden incluir datos de todo tipo con rangos de fechas variados para ayudar a comprender qué sucedió, por qué sucedió, qué puede suceder y qué acción se debería tomar. Dado que los *dashboards* utilizan visualizaciones como tablas, gráficos y diagramas, otros que no están tan cerca de los datos pueden comprender rápida y fácilmente la historia que cuentan o las ideas que revelan. [Tableau,]

Para comprender mejor el objetivo de los *dashboards*, se puede comparar con reportes o visualizaciones de datos.

En general, los reportes suelen tener un enfoque más específico. Su propósito es entregar una visión detallada de un conjunto de datos y tienden a concentrarse en un solo elemento o evento.

Por otro lado, los *dashboards* tienden a ofrecer una vista de alto nivel a partir de grandes cantidades de datos y se crean para responder a una única pregunta. Esa pregunta puede ser amplia, como: “¿Cómo fue el rendimiento del sitio el mes pasado?” O más específica, como: “¿Cuántas unidades se vendieron?” e incluso una pregunta más difícil de responder sin experiencia especializada: “¿Está mejorando la eficiencia general?” [Tableau,]

En cuanto a la visualización de datos, esta es una forma de presentar los datos en un formato visual para hacerlos más fáciles de entender y analizar.

En cambio, los *dashboards* son un resumen de distintos conjuntos de datos, pero relacionados, presentados de manera que la información relacionada sea más fácil de comprender. Los *dashboards* son un tipo de visualización de datos y, a menudo, utilizan herramientas comunes de visualización como gráficos, tablas y diagramas. [Tableau,]

Los *dashboards* toman datos de diferentes fuentes y los agregan para que personas sin conocimientos técnicos puedan leerlos e interpretarlos. Con elementos interactivos, ayudan a que cualquier usuario del *dashboard* comprenda mejor ciertos puntos, explore áreas de mayor interés y fomente más preguntas para llegar a ideas clave o tomar decisiones importantes. [Tableau,]

Existen muchos gráficos que se pueden utilizar para un *dashboard*, un ejemplo es el gráfico de barras, el cual utiliza el largo para representar una medida y pueden ser horizontales o verticales. Este tipo de gráfico es ampliamente utilizado en la visualización de datos puesto a que usualmente son la representación más efectiva para comparar categorías. [Wexler et al., 2017]

Por otra parte, los gráficos de líneas de series temporales suelen mostrar el cambio a través del tiempo. El tiempo es representado en el eje horizontal, mientras que las medidas se muestran en el eje vertical. La altura y las pendientes de la línea nos permiten ver tendencias. [Wexler et al., 2017]

En la figura 8 se puede observar un ejemplo de un *dashboard* para el análisis de agentes de un *call center* proporcionando información sobre las llamadas entrantes, respondidas y el tiempo de conversación. Además, se muestran las principales cinco razones de las llamadas entrantes, como también los agentes ordenados por el porcentaje de llamadas respondidas para revisar su rendimiento.

Algunas de las preguntas clave para el negocio de *call center* que se pueden responder con este *dashboard* [Exchange,] son las siguientes:

- ¿Cuántos clientes llaman y cuántos cuelgan antes de que podamos responder?
- ¿Qué porcentaje de nuestros clientes se sienten satisfechos con su experiencia?
- ¿Cuál es el rendimiento de nuestros agentes de acuerdo con el volumen total de llamadas, la satisfacción de los clientes y las soluciones en la primera llamada?

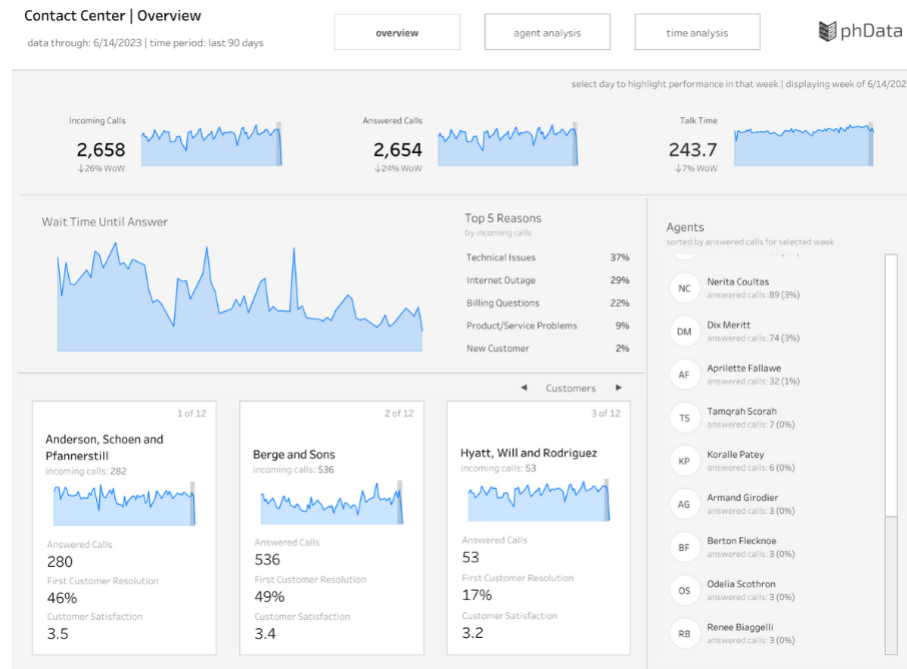


Figura 8: Dashboard para el análisis de agentes de un *call center*.
Fuente: [Exchange,].

2.7. Metodología Kimball

El Ciclo de Vida de Kimball proporciona el marco general que conecta las diversas actividades de una implementación de un sistema de *data warehouse* o inteligencia de negocios (*DW/BI*).

La implementación exitosa de un sistema *DW/BI* depende de la integración adecuada de numerosas tareas y componentes. No basta con tener el modelo de datos perfecto o la tecnología de última generación. Es necesario coordinar los muchos aspectos de un proyecto al igual que un director de orquesta debe guiar muchos instrumentos en una orquesta. [Kimball *et al.*, 2008]

El ciclo de vida de Kimball se puede apreciar en la figura 9.

2.7.1. Planificación del proyecto

Es necesario tener una comprensión básica de los requisitos del negocio para tomar decisiones apropiadas sobre el alcance; la flecha bidireccional entre las cajas de planificación del proyecto y requisitos del negocio en la figura muestra esta dependencia. Luego, la planificación del proyecto se centra en la asignación de recursos, junto con la identificación, asignación, duración y secuencia de las tareas del proyecto. El plan de proyecto integrado resultante identifica todas las tareas asociadas con el Ciclo de Vida de Kimball y las partes

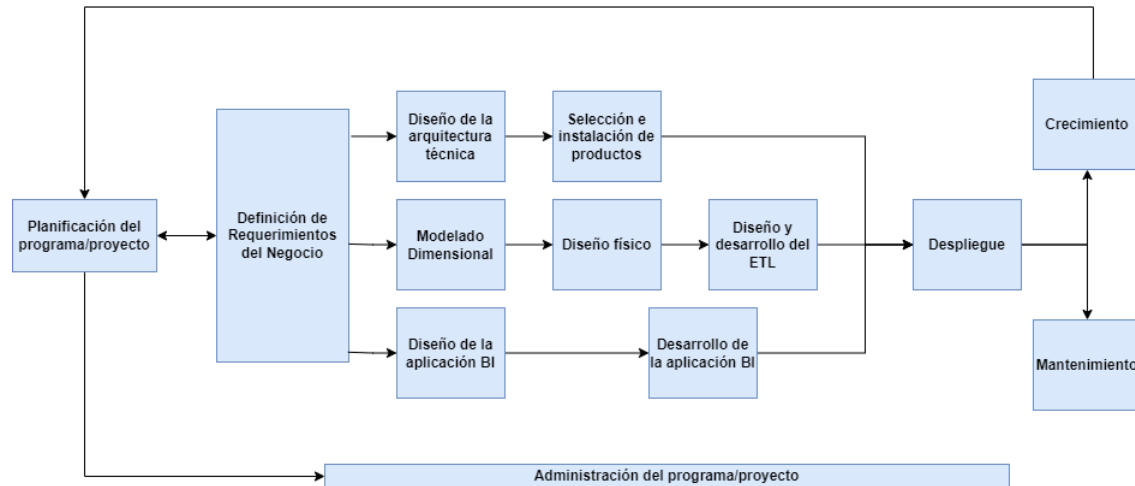


Figura 9: Ciclo de vida de Kimball.

Fuente: [Kimball et al., 2008] y elaboración propia del diagrama traducido.

responsables. [Kimball et al., 2008]

2.7.2. Definición de los requerimientos del negocio

La metodología Kimball para recopilar los requisitos analíticos de los trabajadores difiere significativamente de los enfoques más tradicionales basados en datos. Los analistas de *DW/BI* deben comprender los factores clave que impulsan el negocio para traducir con éxito los requisitos empresariales en consideraciones de diseño. Una definición eficaz de los requisitos empresariales es crucial, ya que establece la base para todas las actividades posteriores del ciclo de vida.

Para la recopilación de los requerimientos, se puede aplicar entrevistas con los usuarios finales para identificar qué información necesitan, cómo la usan y cuáles son sus objetivos de negocio, talleres entre los usuarios que fomenten la interacción y el intercambio de ideas, observación directa, encuestas y análisis de documentación existente y procesos de negocio. [Kimball et al., 2008]

2.7.3. Diseño de la arquitectura técnica

Los entornos *DW/BI* requieren la integración de numerosas tecnologías. El diseño de la arquitectura técnica establece el marco arquitectónico general y la visión. Se deben considerar tres factores, los requisitos empresariales, el entorno técnico actual y las direcciones estratégicas técnicas planificadas para establecer el diseño adecuado de la arquitectura técnica *DW/BI*. [Kimball et al., 2008]

2.7.4. Modelado multidimensional

Durante la recopilación de requisitos empresariales, se determinan y documentan las necesidades de datos de la organización en una matriz que representa los procesos empresariales clave de la organización y su dimensionalidad asociada. Esta matriz sirve como un plano de arquitectura de datos para asegurar que los datos *DW/BI* puedan integrarse y extenderse a lo largo de la organización con el tiempo.

Diseñar modelos dimensionales para apoyar las necesidades de informes y análisis del negocio requiere un enfoque diferente al utilizado para el diseño de procesamiento de transacciones. Tras un análisis de datos más detallado de una fila de la matriz de un proceso empresarial, los modeladores identifican la granularidad de la tabla de hechos, las dimensiones y atributos asociados, y los hechos numéricos. [Kimball *et al.*, 2008]

2.7.5. Diseño de la aplicación de inteligencia de negocios

Inmediatamente después de definir los requisitos empresariales, mientras algunos miembros del equipo *DW/BI* trabajan en la arquitectura técnica y los modelos dimensionales, otros deben colaborar con el negocio para identificar las aplicaciones de BI candidatas, así como las interfaces de navegación adecuadas para satisfacer las necesidades y capacidades de los usuarios. [Kimball *et al.*, 2008]

Selección de productos e instalación

Se deben evaluar y seleccionar componentes arquitectónicos específicos, como la plataforma de hardware, el sistema de gestión de bases de datos, la herramienta de extracción-transformación-carga (*ETL*) o la herramienta de consulta y generación de informes. Una vez seleccionados los productos, deben ser instalados y probados para garantizar una integración adecuada de extremo a extremo dentro del entorno *DW/BI*. [Kimball *et al.*, 2008]

2.7.6. Diseño físico

El diseño de la base de datos física se centra en definir las estructuras físicas, incluyendo la configuración del entorno de la base de datos y la implementación de medidas de seguridad apropiadas. Aunque el modelo de datos físico en la base de datos relacional será prácticamente idéntico al modelo dimensional, existen cuestiones adicionales a abordar, como las estrategias preliminares de ajuste de rendimiento, que van desde la indexación hasta la partición y las agregaciones. [Kimball *et al.*, 2008]

2.7.7. Diseño y desarrollo del ETL

El diseño y desarrollo del sistema de extracción, transformación y carga (ETL) sigue siendo uno de los desafíos más complicados que enfrenta un equipo de proyecto *DW/BI*; incluso cuando todas las demás tareas han sido bien planificadas y ejecutadas, el 70 % del riesgo y del esfuerzo en el proyecto *DW/BI* proviene de este paso. [Kimball *et al.*, 2008]

2.7.8. Desarrollo de la aplicación de inteligencia de negocios

Después de la especificación de la aplicación de inteligencia de negocios, las tareas de desarrollo de la aplicación incluyen configurar los metadatos empresariales y la infraestructura de herramientas, y luego construir y validar las aplicaciones de inteligencia de negocios analíticas y operacionales especificadas, junto con el portal de navegación. [Kimball *et al.*, 2008]

2.7.9. Crecimiento

Se deben establecer procesos de priorización para gestionar la demanda continua del negocio. Luego, se regresa al principio del ciclo de vida, aprovechando y construyendo sobre la base que ya se ha establecido, mientras se dirige la atención a los nuevos requisitos. [Kimball *et al.*, 2008]

2.7.10. Mantenimiento

Una vez que el sistema *DW/BI* está en producción, son necesarias tareas operativas técnicas para mantener el sistema funcionando de manera óptima, incluyendo la supervisión del uso, el ajuste del rendimiento, el mantenimiento de índices y las copias de seguridad del sistema. El enfoque también debe estar en los usuarios al brindar soporte continuo, educación y comunicación. [Kimball *et al.*, 2008]

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

3.1. Consideraciones

Como se mencionó en la definición del problema, la carga de datos de deudas e información del deudor depende de lo que envíe cada mandante, por lo que los campos y el nombre asociado a estos difiere por cada uno de ellos, requiriendo un entendimiento previo de la información que se envía, así como también del formato, para poder realizar el proceso de extracción, limpieza y carga al *data mart*.

En el *software* de cobranza, es necesario subir los archivos necesarios al *SFTP* para poblar con datos de contacto, deuda y los deudores a contactar, específicamente mediante llamada, ya sea predictivo o progresivo, con el fin de que el discador del *software* llame a los números de contacto automáticamente.

El sistema de bases de datos utilizado para el *data mart* debe tener soporte con *Microsoft Power BI*, *software* de análisis de datos utilizado en Netpag.

En la empresa, se suele utilizar *Talend* para los procesos de *ETL* y automatización.

3.2. Definición de Requerimientos del Negocio

De acuerdo a la metodología de *Kimball*, es necesario tener un conocimiento del negocio para desarrollar un *data mart* o *data warehouse* que cumpla con las necesidades de los usuarios.

Para ello, se realizaron entrevistas a los usuarios, con el objetivo de conocer su perspectiva, hablando sobre sus trabajos, sus objetivos y sus desafíos, entender lo qué hacen y cómo toman decisiones.

Los usuarios entrevistados fueron:

1. Analista de Inteligencia de Negocios, encargado de obtener los requerimientos del área de Operaciones para levantar los tableros en *Power BI* que apoyen la toma de decisiones.
2. Analista de Administración de Finanzas, encargado de definir las metas y con la que se evalúan a los ejecutivos de cobranza y calcular los bonos según cumplimiento, junto al análisis de costos de proyecto y de masividades como costos de envío de SMS y correos. Se comparten las metas definidas unificadas para todos los clientes.

3. Ingeniero de datos, encargado de darle utilidad a los datos y dejarlos en una ubicación que sea accesible a las partes interesadas que posteriormente los utilizan en sus reportes o análisis.
4. Científico de datos, encargado de generar modelos predictivos.

A partir de los analistas, se obtuvieron las siguientes métricas asociadas a reportes o metas que se utilizan hoy en día, siendo necesario que estén en el *data mart*.

- Recupero, porcentaje de cumplimiento de metas de recuperación de capital asociado a los clientes de cobranza B2C, de acuerdo con metas mensuales informadas por parte de Gerencia.
- Efectividad de pago, porcentaje de cumplimiento de la meta mensual para el porcentaje de total de pagos realizados sobre los compromisos de pago de clientes generados durante el mes de devengamiento.
- Efectividad de compromiso, porcentaje del total de compromisos sobre los contactos directos durante el mes.

Por otro lado, se descubrió una operación importante que se debe cubrir: la **asignación de pagos**, es decir, a qué ejecutivo se le asigna un pago de la deuda. Aquel cálculo es vital para la entrega de bonos a los ejecutivos por rendimiento, por lo que el *data mart* debe tener los datos necesarios para llegar al resultado, considerando las siguientes reglas de negocio:

- La categoría y/o subcategoría de las gestiones que se consideran para la asignación, por ejemplo, aquellas con compromiso de pago. Esto varía según mandante.
- Los días de protección para una gestión con compromiso de pago, esto quiere decir que el ejecutivo que realiza este tipo de gestión tiene una protección por una cierta cantidad de días para que el deudor haga el pago y se le considere. Para algunos mandantes se considera 7 o 3 días.
- Si varios ejecutivos tienen compromiso de pago con el mismo deudor y el pago llega en los días de protección, el bono es para el **primer ejecutivo** en conseguir el compromiso. Si ninguno está dentro de los días de protección y llega el pago, el pago se le asigna al **último ejecutivo** en conseguir el compromiso.
- Esta asignación se realiza **cada mes**, pero el mes depende de la **fecha de corte** de cada mandante.
- Se debe considerar el pago consolidado durante el mes, es decir, sumando los abonos en caso de existir, y quedarse con la última fecha de pago. Algunos mandantes envían los pagos ya consolidados, mientras que para otros se debe calcular.

- Pagos negativos son errores y tampoco se consideran pagos menores a 10 pesos.

Otro punto a considerar son los modelos predictivos que se buscan desarrollar haciendo uso del *data mart*:

- Probabilidad de pago, donde se utiliza los datos de pagos y deudores para entrenar al modelo el cual predice si un deudor pagará o no. Este modelo se ejecuta cada mes, donde por parte del mandante llega el archivo con los deudores, el cual se ingresa al modelo para predecir el pago de estos. Este modelo varía para cada mandante, puesto que se pueden utilizar columnas exclusivas según mandante.
- Estimación de esfuerzo por parte de ejecutivos
- Mejor canal de contacto para los deudores

En cuanto a la operación de cobranza, se debe tener en cuenta lo siguiente:

- Se bloquean contactos diariamente, es decir, aquellos que no hay que contactar debido a una lista negra por parte del mandante, porque ya hubo contacto directo el día anterior o porque ya hubo un pago.

Uno de los puntos más importantes para el modelo de negocios es conseguir la mayor **recuperación de las deudas y contactabilidad** con los deudores, que son las variables por las que los mandantes evalúan a NetPag. A su vez, se debe crear las estrategias considerando los recursos de NetPag, puesto que cada llamada, mensaje de texto o correo tiene un **costo asociado**.

El núcleo de la estrategia de NetPag está en encontrar aquellos deudores más propensos a pagar, junto al medio de contacto que más respondan o visualicen, con el fin de que estos sean a los que contactar para que realicen el pago.

- ¿Cómo lograr que los clientes que contratan a NetPag se sientan satisfechos con el servicio? Mediante la contactabilidad a sus deudores y la recuperación de deudas. La evaluación de cada mandante varía, puesto que se puede enfocar en la contactabilidad y/o en la recuperación de deudas en distintas proporciones. Cada mandante envía al final de cada periodo un *benchmark* del resultado de NetPag según sus KPIs.
- ¿Cómo mejorar la contactabilidad?
 - Analizando las gestiones para identificar los mejores canales, horarios, contactos, entre otros por los que contactar a los clientes, aprovechando los intentos de contacto que se tienen antes incumplir la ley.

- Accediendo a todos los contactos entregados por mandantes o internos, considerando la posibilidad de que con otro mandante ya se haya conseguido un contacto valioso mediante teléfono, correo o mensaje de texto.
 - Al tener una mayor cantidad de contactos por deudor se vuelve más probable que alguno de estos generen un contacto directo.
- ¿Cómo lograr una mejor recuperación de las deudas?
 - Relacionado con el anterior, contactando a los deudores más propensos a pagar de forma oportuna para que estos realicen el pago.
 - Abarcando y priorizando ciertas carteras de los mandantes, identificando aquellas más urgentes a las que contactar según días de mora o tramo de mora y más probables de que se paguen.
 - ¿Cómo reducir costos y mejorar los ingresos? Analizando todo lo anterior en conjunto: las gestiones exitosas junto con los pagos para comparar los costos de las estrategias de contacto versus el recupero de las deudas.

3.3. Diseño de la arquitectura técnica

3.3.1. Ecosistema de la empresa

NetPag utiliza los productos *Microsoft*, trabajando con *Microsoft SQL Server* año 2016 y *Microsoft Power BI*. El escoger los productos con los que ya trabaja actualmente NetPag facilita el entendimiento de los usuarios que harán uso del *data mart* y facilita la obtención de soporte, tanto del área encargada como de *Microsoft*.

3.3.2. Estructura de los datos

Al revisar los datos disponibles y utilizados por NetPag, se puede identificar una estructura y relaciones definidas, destacando las principales:

- Cada **deuda** y **pago** está asociada a un mandante, periodo y deudor.
- Cada **dato de contacto** está asociada a un deudor.
- Cada **gestión** está asociada a un mandante, periodo, deudor y ejecutivo.
- Cada **gestión** se realiza mediante un canal específico.

Por otra parte, es crucial tener en cuenta el conocimiento del personal de NetPag, quienes ya están capacitados para trabajar con consultas SQL.

3.3.3. Consultas y volumen de los datos

El volumen de los datos es alto y aumenta diariamente, especialmente para las gestiones masivas como el envío de correos electrónicos, mensajes de texto y bots. Estos datos abarcan desde el año 2022 hasta el año presente, 2024.

Para hacer una estimación, en las gestiones de llamada, para el mes Octubre de 2023 se tienen 13.7 millones de datos aproximadamente. En promedio, 572 mil gestiones diarias aproximadamente.

Para ese mismo mes en gestiones masivas de correo, se enviaron 4.8 millones de correos. En promedio, 220 mil correos diarios.

En gestiones de bots, ese mes se realizaron 13.6 millones de gestiones. En promedio, 567 mil bots diarios.

En gestiones de mensajes de texto, se enviaron 572 mil SMS aproximadamente en ese mes. En promedio, 37 mil SMS diarios.

En un mes, se tienen en total aproximadamente 33 millones de gestiones.

Las consultas suelen realizarse por periodo y mandante, principalmente debido a los acuerdos comerciales que se evalúan mensualmente con cada uno de los mandantes. Sin embargo, si se quiere comparar, por ejemplo, los resultados de gestiones en varios periodos por parte de NetPag, se vuelve un desafío debido al gran consumo de recursos.

Existen varias opciones que podrían solucionar este desafío. Una de ellas podría ser la **creación de índices por periodo** A. Al indexar los datos por periodos específicos, en este caso, por el mes, se puede mejorar significativamente el rendimiento de las consultas. Esto permite que las búsquedas y comparaciones de datos entre diferentes periodos sean más rápidas y eficientes, reduciendo la carga sobre el sistema.

Otra opción es la **partición horizontal** de las tablas B, que consiste en dividir una tabla en múltiples tablas que contienen el mismo número de columnas, pero menos filas. En este caso, una opción es particionar las tablas por periodo.

Por otra parte, la **materialización de vistas** C en este contexto implicaría crear tablas distintas tablas por periodo.

3.3.4. Selección de producto

Después de analizar el ecosistema de la empresa, la estructura de los datos y las necesidades de la empresa, se ha decidido optar por una base de datos relacional SQL, específicamente, *Microsoft SQL Server*. Esta decisión se basa en los siguientes factores:

- **Compatibilidad con el sistema de la empresa:** El personal de NetPag ya está familiarizada con las bases de datos relacionales, teniendo servidores de *Microsoft SQL Server*.
- **Estructura de los datos:** Si bien el formato de los datos y cantidad de columnas varía por mandante, existen relaciones definidas en el modelo actual de cobranza por parte de NetPag.
- **Consultas y volumen de datos:** NetPag tiene un gran volumen de datos históricos, especialmente para gestiones masivas. Las consultas se dan principalmente por periodo, lo cual da la opción de crear índices y/o particiones en esta columna para mejorar los tiempos de respuesta y no sobrecargar los servidores.
- **Soporte:** La empresa cuenta con licencia de *Microsoft* y además el área de *TI* maneja los servidores de *Microsoft SQL Server*, encargándose de solucionar problemas que puedan surgir de almacenamiento o conexión.
- **Inteligencia de negocios:** Ya existe un *data mart* en otro servidor para el área *B2B* de la empresa [SEPÚLVEDA TAPIA, 2021]. Al implementar el *data mart* para el área *B2C* en la misma infraestructura se facilita la creación de un *data warehouse* que abarque distintas áreas de la empresa, siguiendo un modelo Bottom-up. [Vaisman y Zimányi, 2014]

3.4. Modelado Dimensional

3.4.1. Tablas de hechos

El principal desafío del modelado del *data mart* son las tablas de hecho para las gestiones.

Estas tablas de hecho se dividen según el canal de la gestión debido a que cada canal tiene información distinta que debe ser considerada. Por ejemplo, para las **llamadas** es importante manejar el usuario que la gestionó, mientras que para gestiones masivas como correo y mensajes de texto se debe almacenar la campaña asociada y fechas de apertura en el caso de correos.

Gestión_Llamada

Contiene todas las gestiones que se han realizado por llamadas para ser gestionadas por un ejecutivo, esto incluye las llamadas *inbound*, es decir, llamadas entrantes, como las llamadas salientes *outbound*. La granularidad es a nivel de cada llamada individual realizada. Esto significa que cada fila de la tabla representa una única llamada, ya sea entrante o saliente, incluyendo detalles específicos de la misma.

La fuente de datos son las bases de datos cargadas por el proveedor de *software* de cobranza.

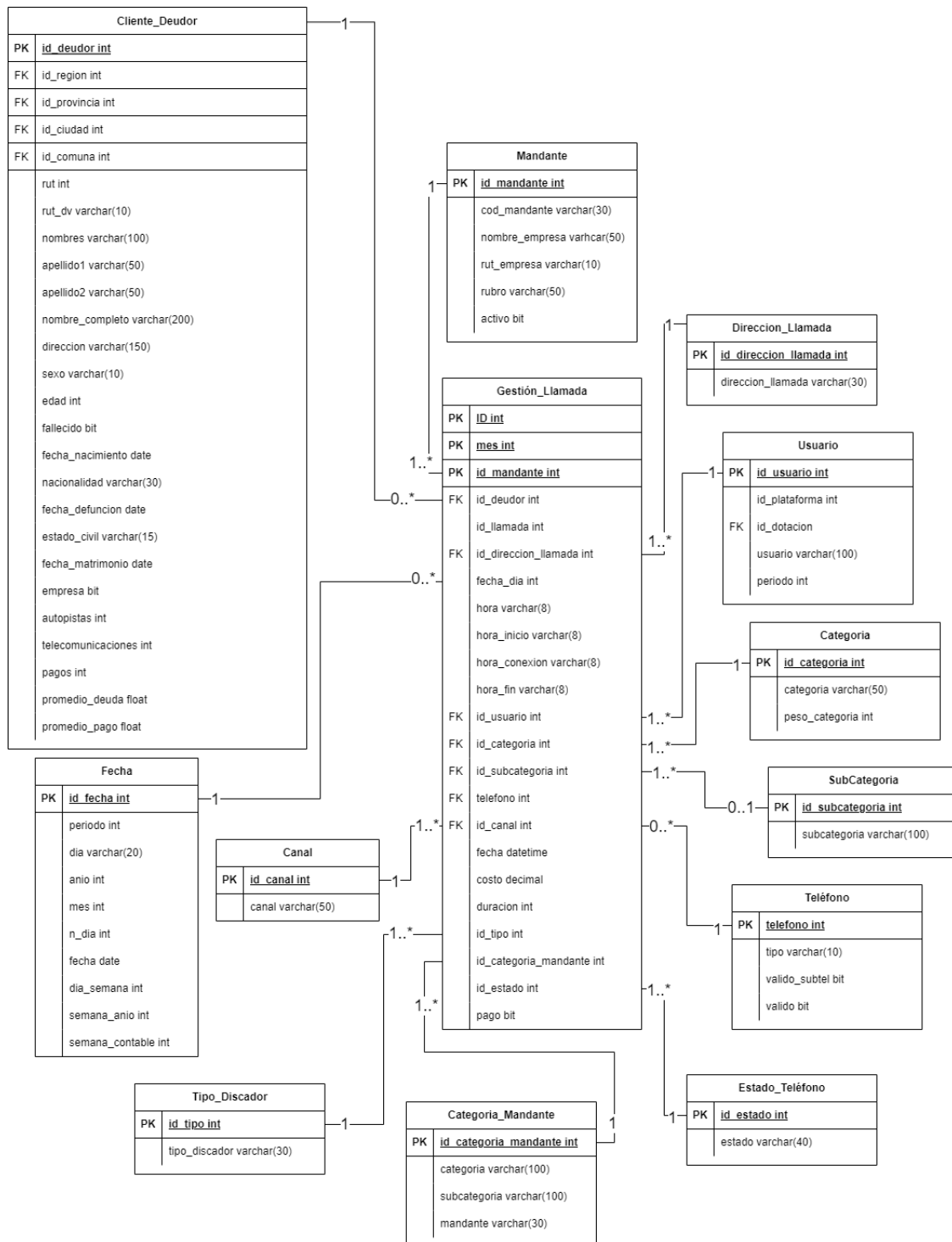


Figura 10: Modelado de la tabla de hecho Gestion_Llamada.
Fuente: Elaboración propia.

Gestión_Bot

Contiene todas las gestiones que se han realizado por el canal masivo de *bot*. La granularidad es a nivel de cada llamada de bot individual realizada. Esto significa que cada fila de la tabla representa una llamada de un bot, incluyendo detalles específicos del envío.

Las fuentes de datos son:

- API de proveedor de bots.
- Tablas de carga manual por parte de TI para bots enviados por un segundo proveedor debido a que no tienen API.

Gestión_Correo

Contiene todas las gestiones que se han realizado por el canal masivo de correo electrónico. La granularidad es a nivel de cada correo electrónico individual enviado. Esto significa que cada fila de la tabla representa un correo enviado, incluyendo detalles específicos del envío.

La fuente de datos son los datos de la API de proveedor de correos electrónicos.

Gestión_SMS

Contiene todas las gestiones que se han realizado por el canal masivo de mensajes de texto. La granularidad es a nivel de cada mensaje de texto individual enviado. Esto significa que cada fila de la tabla representa un único mensaje de texto, incluyendo detalles específicos del envío.

La fuente de datos son las tablas de carga manual por parte de TI para mensajes de texto enviados por el proveedor de mensajes de texto.

Pago

Contiene todos los pagos enviados por los distintos mandantes. Esta tabla de hecho tiene la dificultad de que un mandante envía los pagos con distinta granularidad a los demás, siendo un caso de tabla de hecho con múltiples granularidades. El mayor detalle de pago es por transacción, sin embargo, un mandante envía los pagos acumulados por deudor del periodo en curso. Para abordar esto, se decide utilizar una tabla de hecho híbrida al integrar las dos granularidades al ser solo un mandante el que tiene granularidad distinta. Esto queda documentado como una regla de negocio, especificando el envío de pagos por mandante.

La fuente de datos son las bases de datos cargadas por el proveedor de *software* de cobranza.

Cliente_Por_Mandante

Contiene todos los deudores enviados en la asignación de cada periodo según el mandante. Refleja qué deudores hay que contactar durante el periodo. La granularidad de esta tabla

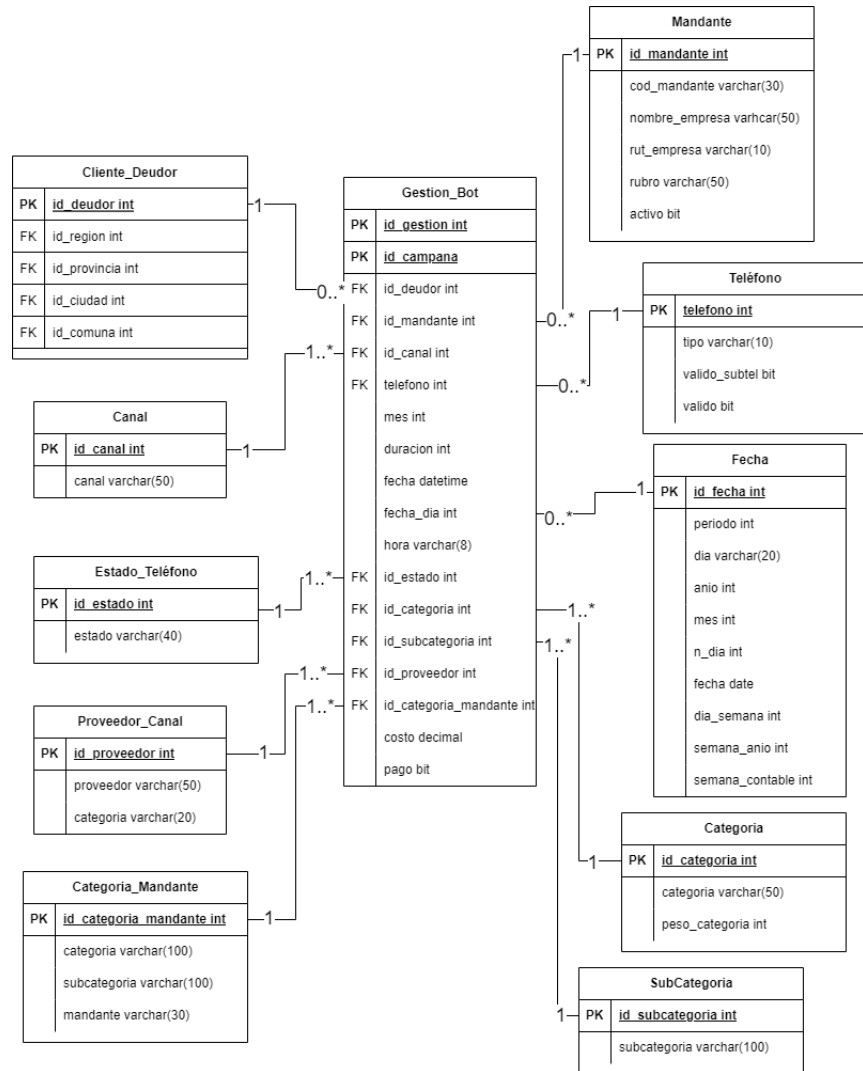


Figura 11: Modelado de la tabla de hecho Gestion_Bot.
Fuente: Elaboración propia.

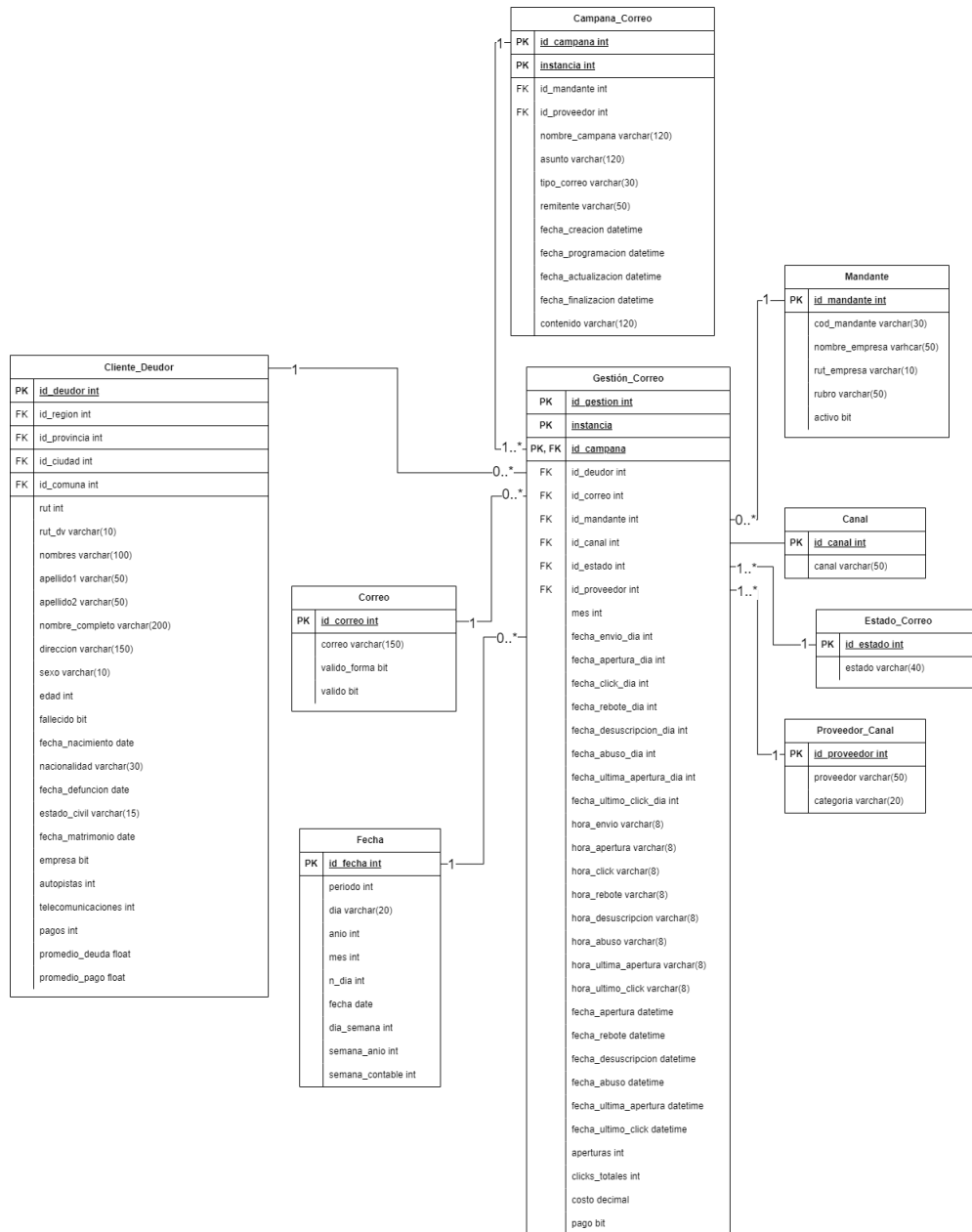


Figura 12: Modelado de la tabla de hecho Gestion_Correo.
Fuente: Elaboración propia.

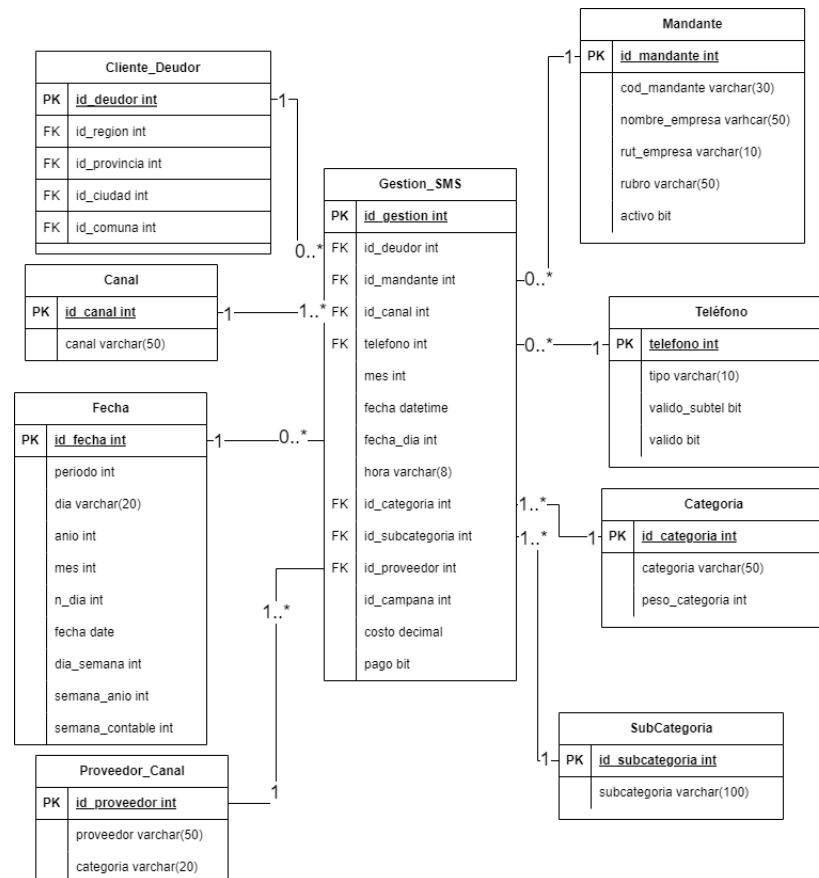


Figura 13: Modelado de la tabla de hecho Gestion_SMS. Se omite algunos datos de Cliente_Deudor que ya se encuentran en figuras anteriores.

Fuente: Elaboración propia.

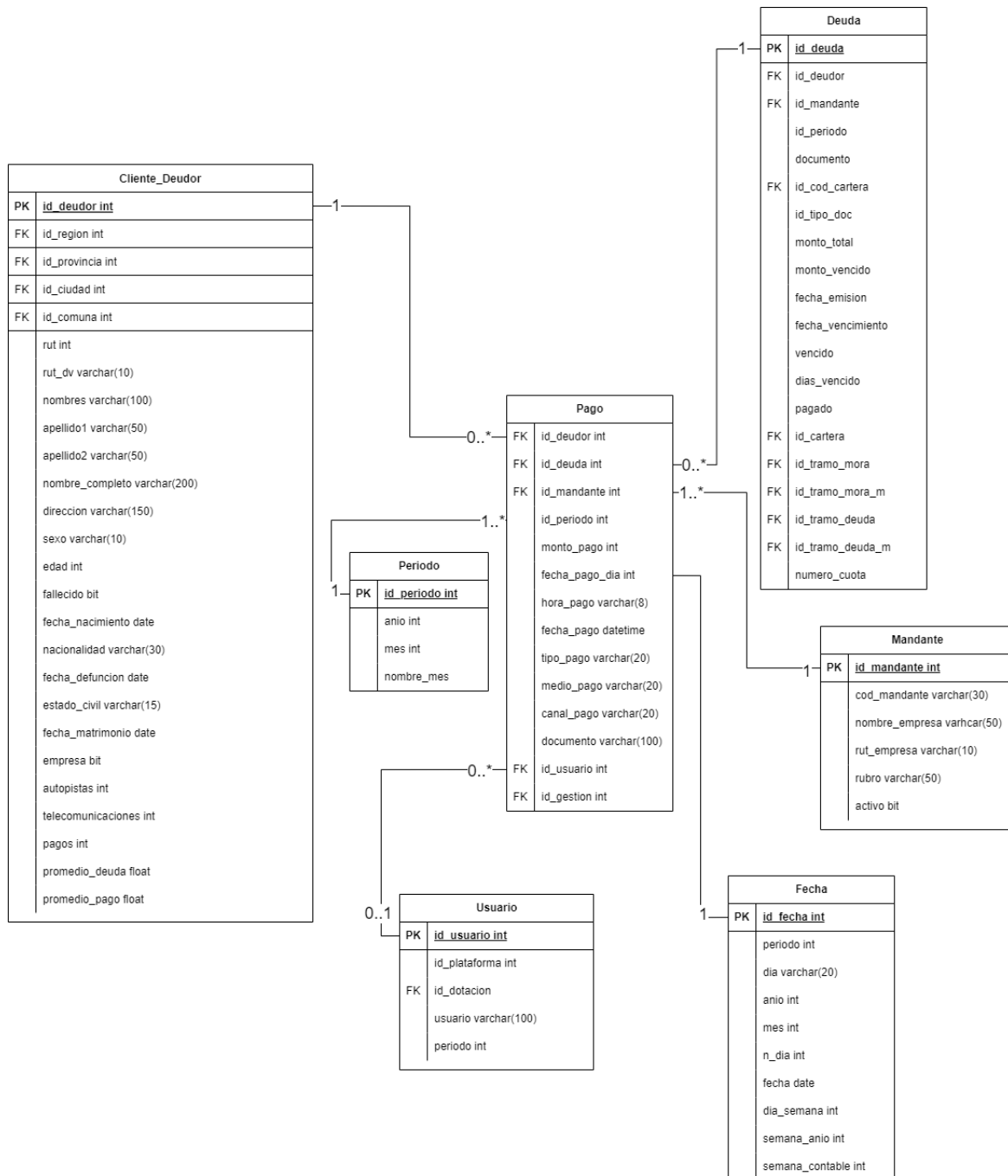


Figura 14: Modelado de la tabla de hecho Pago.
Fuente: Elaboración propia.

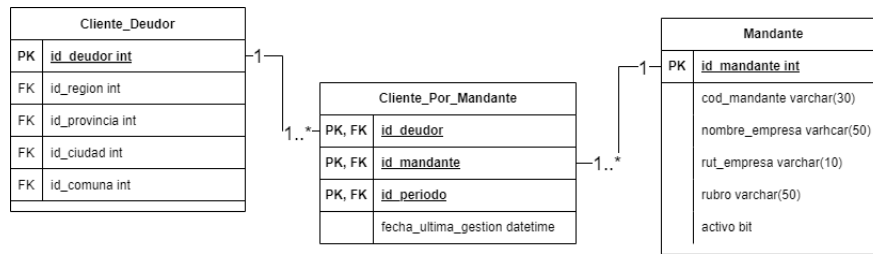


Figura 15: Modelado de la tabla de hecho Cliente_Por_Mandante.
Fuente: Elaboración propia.

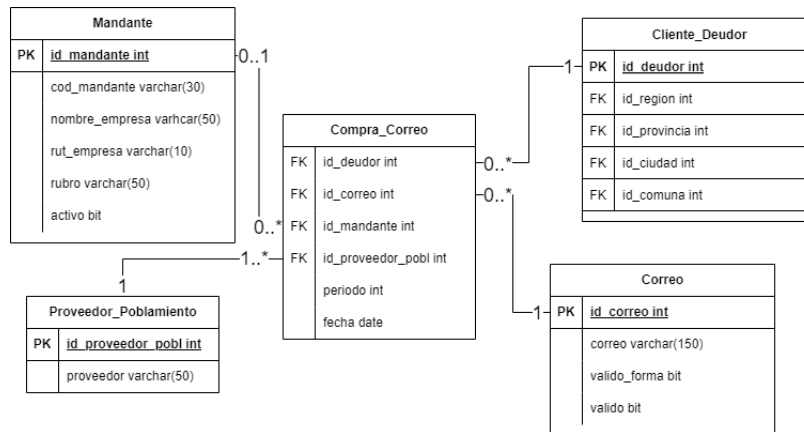


Figura 16: Modelado de la tabla de hecho Compra_Correo.
Fuente: Elaboración propia.

es por periodo, según las asignaciones enviadas. Se puede enviar más de una asignación durante el mes, donde si hay cliente deudores nuevo, se agregan a la tabla con el periodo correspondiente.

La fuente de datos son las bases de datos cargadas por el proveedor de *software* de cobranza.

Compra_Correo y Compra_Teléfono

Contiene todas las compras realizadas a proveedores para poblar los contactos de deudores. Estas compras se realizan a proveedores externos cuando los datos de contacto enviados por los mandantes no son suficientes o no han tenido éxito. La granularidad de estas tablas es a nivel de cada compra realizada. Cada fila representa una compra individual, incluyendo detalles sobre el proveedor y por parte de qué mandante se realizó.

La fuente de datos son tablas creadas por el área de *TI* para registrar las compras realizadas a distintos proveedores, a la cual mediante este proyecto se agregó un registro automático de compras realizadas por uno de los proveedores de contacto.

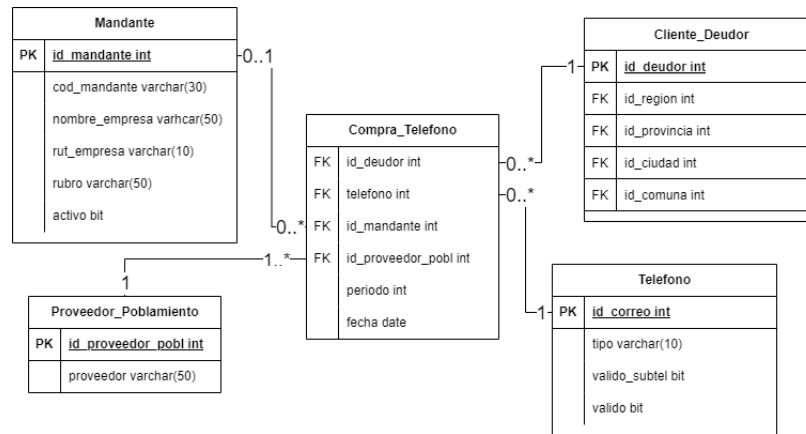


Figura 17: Modelado de la tabla de hecho Compra_Teléfono.
Fuente: Elaboración propia.

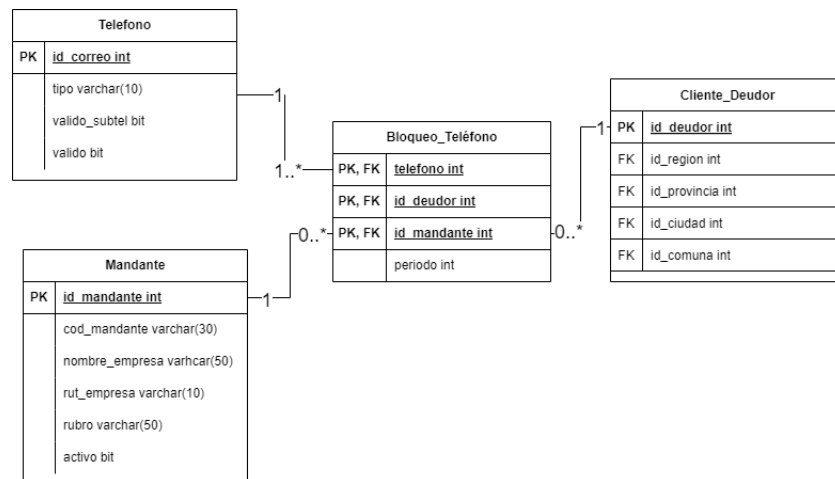


Figura 18: Modelado de la tabla Bloqueo_Teléfono.
Fuente: Elaboración propia.

Bloqueo_Correo y Bloqueo_Teléfono

Contiene registro de los bloqueos de contactos realizados. Los bloqueos son asociados a listas negras, lo que implica que los contactos (correos electrónicos o números de teléfono) han sido identificados como incorrectos o no pertenecientes al cliente deudor asignado. Este proceso asegura que no se utilicen contactos ineficaces o erróneos en futuras gestiones de cobranza. La granularidad de estas tablas es a nivel de cada bloqueo realizado. Cada fila representa un contacto específico que ha sido bloqueado.

La fuente de datos son tablas creadas por el área de *TI* actualizadas manualmente y mediante una plataforma interna para el registro de reclamos.

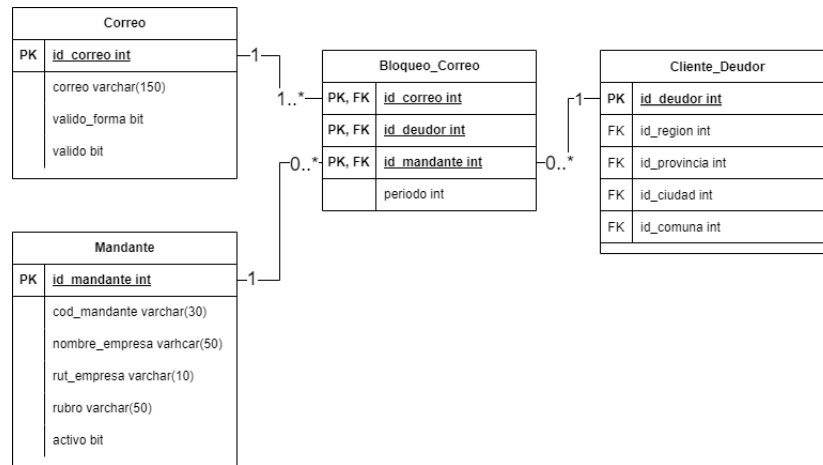


Figura 19: Modelado de la tabla Bloqueo_Correo.
Fuente: Elaboración propia.

3.4.2. Dimensiones

Mandante

Todos los mandantes, o clientes de NetPag. Almacena el código del mandante según la base en la que se encuentra, el rubro de la empresa y si esta activo o no como cliente de NetPag.

Cliente_Deudor

Contiene los datos de todos los deudores enviados por asignaciones de los mandantes. Se acumula la información de todos los mandantes para escoger la opción más completa para **nombre, dirección, sexo y fecha de nacimiento**.

Fuentes de datos:

- Asignaciones de cada mandante cargadas en bases de *SQL Server*.
- Extracciones de reporte *Sinacofi*.
- *Dataset* de nombres registrados en Chile entre 1920 y 2021. [Rivaquiroga,]

Ciudad, Comuna, Provincia y Región

Dimensiones que contienen todas las ciudades, comunas, provincias y regiones de Chile respectivamente que se asocian con Cliente_Deudor.

Una forma de abordar estas dimensiones sería unir la tabla de Comuna con Ciudad, Ciudad con Provincia y así sucesivamente, sin embargo, en este diseño se decide unir la tabla de

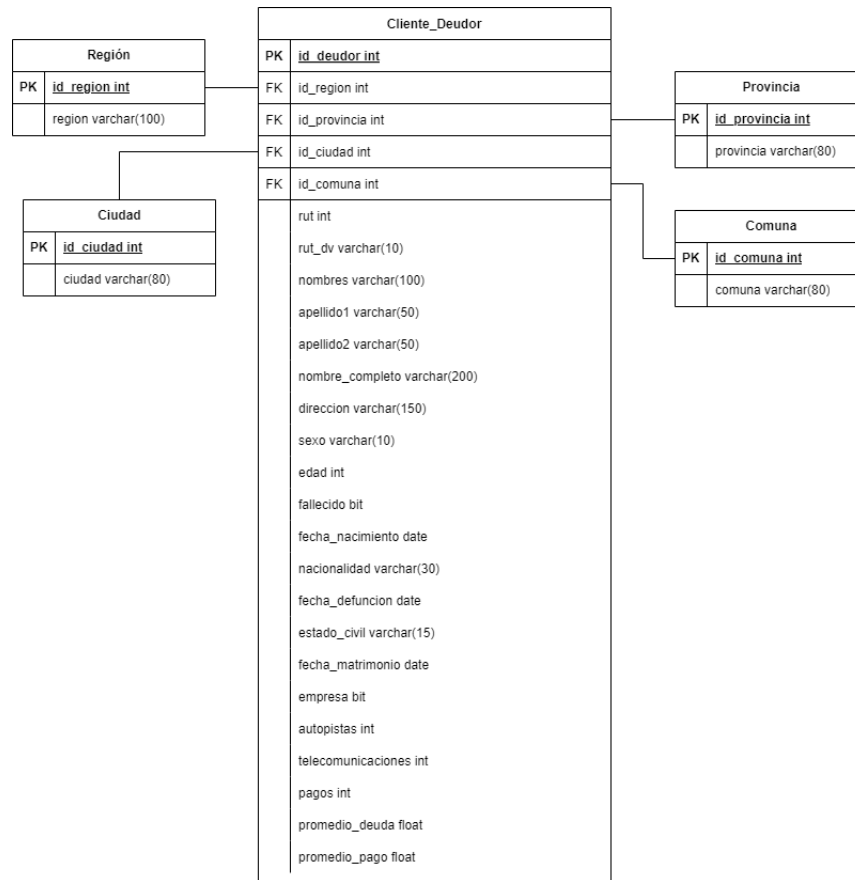


Figura 20: Modelado de la tabla de dimensión Cliente_Deudor.
Fuente: Elaboración propia.

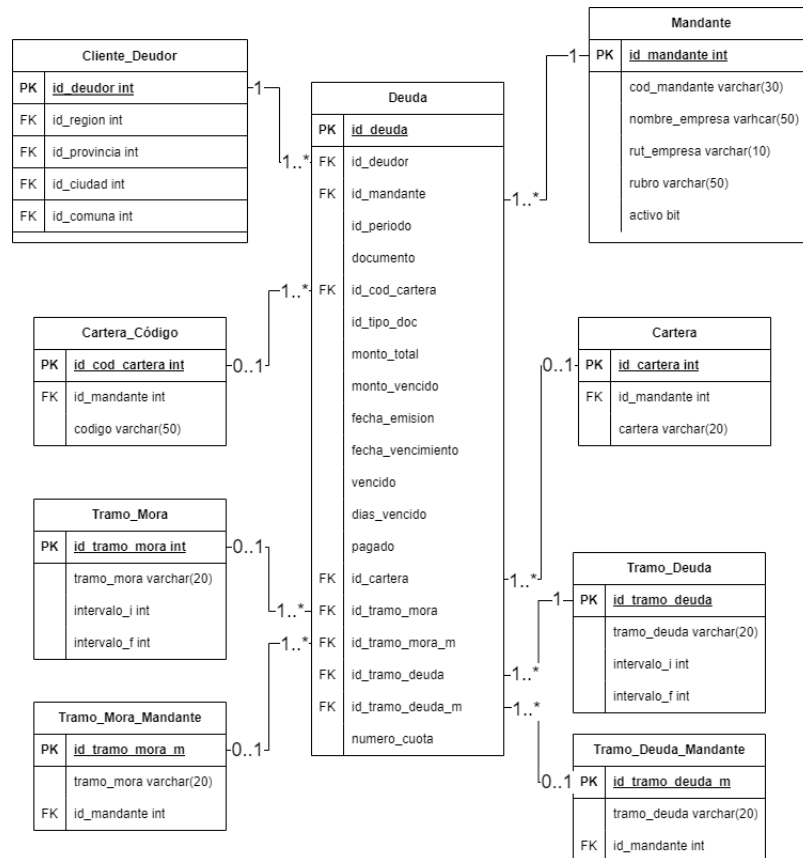


Figura 21: Modelado de la tabla Deuda.

Fuente: Elaboración propia.

Cliente_Deudor con cada una de estas localidades debido a que no siempre se va a tener el mismo detalle en la dirección.

Deuda

Contiene las deudas de cada cliente deudor según el periodo y mandante. La granularidad depende de este último, puesto que algunos envían la deuda según documento y otros el acumulado de la deuda por cliente deudor.

La fuente de datos son las bases de datos cargadas por el proveedor de *software* de cobranza.

Teléfono y Correo

Contiene todos los teléfonos y correos enviados en asignaciones o compras.

Fuentes de datos:

1. Bases de datos del proveedor del *software* de cobranza

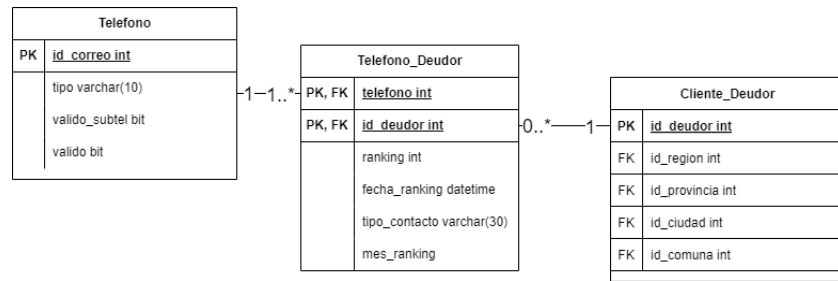


Figura 22: Modelado de la tabla Teléfono_Deudor.
Fuente: Elaboración propia.

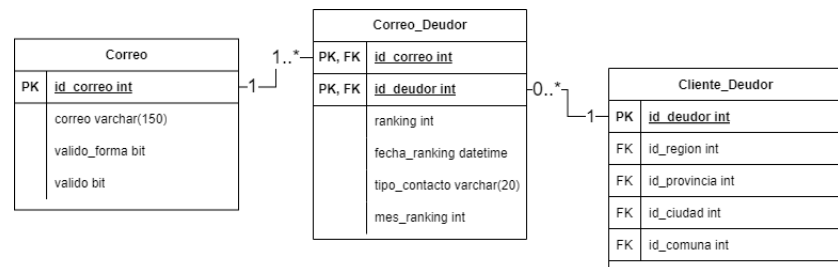


Figura 23: Modelado de la tabla Correo_Deudor.
Fuente: Elaboración propia.

2. Base de numeración telefónico de *Subtel*. [de Telecomunicaciones de Chile, 2024]
3. Tablas ya existentes de teléfonos y correos de compra.
4. Tablas ya existentes de poblamiento, es decir, confirmaciones externas del contacto y su deudor.
5. API de uno de los proveedores de teléfonos y correos.

Teléfono_Deudor y Correo_Deudor

Contiene todos los teléfonos y correos respectivamente de los deudores enviados en la asignación de cada mandante. Esta tabla permite identificar los teléfonos asociados a un deudor para realizar las gestiones.

Las fuentes de datos son las mismas de Teléfono y Correo.

Teléfono_Deudor_Mandante y Correo_Deudor_Mandante

Al igual que en la tabla anterior, contiene los teléfonos y correos respectivamente de los deudores agregando el mandante que lo envió. Esto permite tener un control de los teléfonos enviados por cada mandante en caso de terminar el acuerdo, además de poder clasificar el ranking del teléfono según mandante.

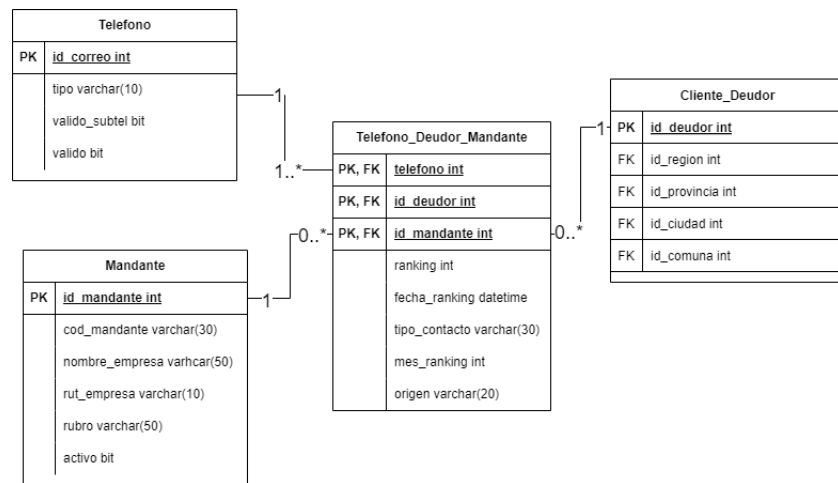


Figura 24: Modelado de la tabla Teléfono_Deudor_Mandante.
Fuente: Elaboración propia.

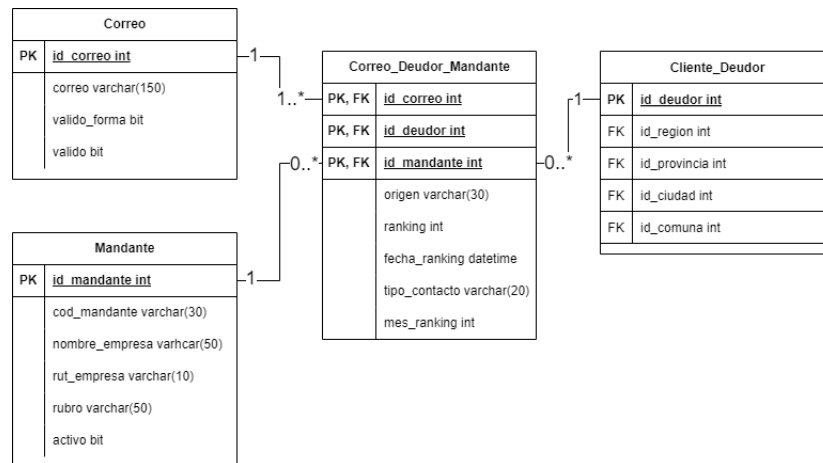


Figura 25: Modelado de la tabla Correo_Deudor_Mandante.
Fuente: Elaboración propia.

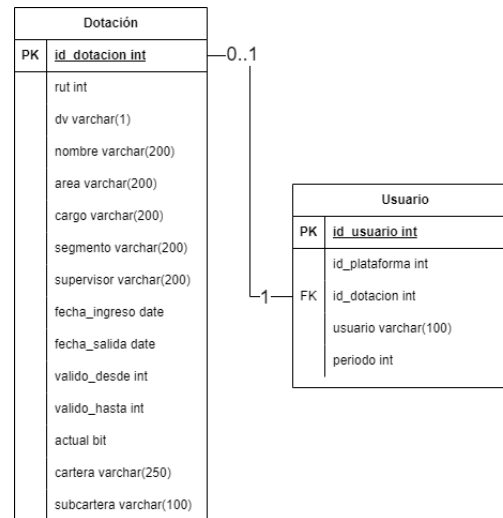


Figura 26: Modelado de la tabla Usuario y Dotación.
Fuente: Elaboración propia.

Las fuentes de datos son las mismas de Teléfono y Correo.

Canal

Contiene los canales por los que se realiza la gestión. Estos pueden ser: Llamada, *bot*, SMS o Mail.

Cartera

Contiene las distintas divisiones de deudores que envía cada mandante en su asignación.

Dotación

Contiene los datos de los trabajadores en la empresa, junto con su cargo, área, jefe y periodos de validez del registro. Si hay algún cambio en el cargo, área o jefe se agrega un nuevo registro con nuevos periodos de validez, representando una tabla de dimensión de tipo 2 con la diferencia de que la validez de los registros es por periodos en vez de fechas.

La fuente de datos es una tabla ya existente que se carga a partir de un *Excel* con la dotación de cada mes y es actualizada manualmente por Recursos Humanos.

Usuario

Contiene los usuarios del *software* de cobranza que realizan las gestiones. La fuente de datos son las mismas gestiones de las bases de datos cargadas por el proveedor de *software* de cobranza.

Categoría

Contiene las categorías que puede tomar el resultado de una gestión, estas pueden ser las siguientes:

- Compromiso de Pago: Se logró un compromiso de pago por parte del deudor.
- Contacto Directo: Se logró contactar con el titular de la deuda.
- Contacto Indirecto: Se logró contactar con algún familiar o conocido del titular.
- Sin Contacto: No se logró contactar con el titular de la deuda.

SubCategoría

Contiene la categorización del resultado de las gestiones en subcategorías para proporcionar una clasificación más detallada. Algunos ejemplos son: Insultos, Molestia, Buzón de Voz o Fallecido.

Campaña_Correo y Campaña_Bot

Contiene todas las campañas enviadas por correo y bot respectivamente. Cada campaña representa un lote de correos electrónicos enviados a un conjunto específico de destinatarios.

La fuente de datos para estas campañas son las APIs respectivas de cada canal.

Estado_Correo y Estado_Teléfono

Contiene los estados de los correos electrónicos y los teléfonos según el resultado de las gestiones. Por ejemplo, para los correos este estado puede ser **abierto, rebotado o sin apertura**, mientras que para los teléfonos puede tener valores como **ocupado, contestado, error de red, buzón de voz, en cola, etc.** Esta dimensión se diferencia de la categoría al representar el estado del medio de contacto más que el resultado de la gestión de cobranza.

Tramo_Deuda_Mandante y Tramo_Mora_Mandante

Esta tabla almacena los intervalos de deuda y los intervalos de días de mora utilizados por los mandantes. Cada registro representa un tramo específico de deuda o mora, facilitando la categorización y análisis de las deudas según su monto y los días de retraso en el pago. El tramo de deuda define el rango en el que se encuentra la deuda en términos monetarios, mientras que el tramo de mora especifica el intervalo de días que la deuda ha estado en mora.

Tramo_Deuda y Tramo_Mora

Contiene la homologación de los tramos de deuda y tramos de mora a tramos utilizados por NetPag.

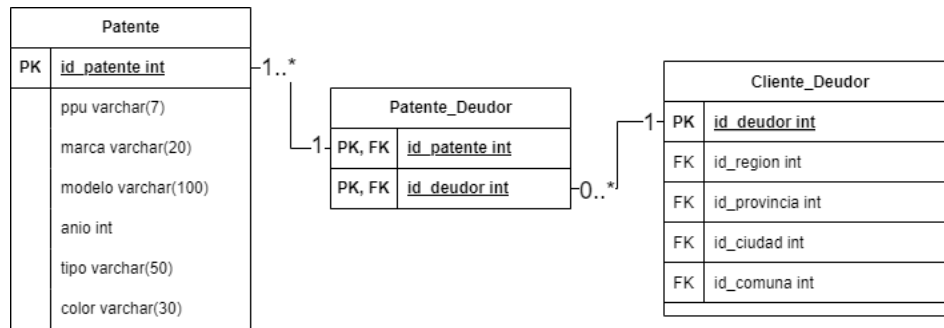


Figura 27: Modelado de la tabla Patente y Patente_Deudor.
Fuente: Elaboración propia.

Para el tramo de mora, se incluyen valores como: No vencido, 1 a 30, 31 a 60, 61 a 90, etc. Mientras que el tramo de deuda puede ser: 1 a 50M, 50M a 100M, 100M a 200M, 200M a 500M, etc.

Proveedor_Poblamiento

Contiene todos los proveedores de datos de contacto a los que se ha realizado una compra. Esta dimensión se asocia con la tabla de hechos de Compra_Telefono y Compra_Correo.

Proveedor_Canal

Contiene todos los proveedores de canales de contacto como SMS, correo electrónico o bots.

Patente y Patente_Deudor

La tabla Patente contiene todas las patentes encontradas en mandantes relacionados con vehículos junto con información del vehículo como modelo, año y marca. La tabla Patente_Deudor contiene las patentes asignadas a cada deudor.

Las fuentes de datos son las asignaciones de mandantes relacionados con vehículos como autopistas y compra/préstamo de vehículos junto a datos extraídos del reporte Sinacofi.

Estas tablas son útiles para buscar relaciones entre el tipo de vehículo, deudor, deuda y fechas de pago. Por otra parte, puede ser utilizada para reportes del área Comercial para analizar acuerdos de recuperación de deudas, por ejemplo, recuperación de deudas de estacionamiento donde estos solo tienen el vehículo con deuda, pero sin información del deudor.

Fecha

Tabla de dimensión temporal. Contiene fechas por día desde el 2022 hasta 2031, especificando el periodo (mes y año), día, año, mes, día del mes, día de la semana, semana del año y semana contable.

3.4.3. Tablas de Reporte

Gestión_Teléfono_Mes

Para abordar los casos de consulta por mes y ranking de los teléfonos, se utiliza una vista materializada al juntar todas las gestiones realizadas por teléfono para un mes en específico. Esta vista se va creando por cada mes en curso y une las gestiones de llamada, mensaje de texto y bots. Se extraen columnas más importantes y que se tengan en común en estos tres canales y se incluyen columnas necesarias para realizar el ranking.

Pago_Asignado

Esta es una tabla de reporte que además, tiene su propio proceso para asignar los pagos. Utiliza como fuente las tablas Gestión_Llamada , Gestión_Bot y Pago, donde cada pago es asignado a un ejecutivo dependiendo de si la gestión cae en ciertas categorías y días de protección (diferencia de días entre la gestión y el pago).

Se diferencia de la tabla de hecho de Pago al tener distinta granularidad. La tabla Pago_Asignado tiene los pagos acumulados por deudor y periodo, siendo la forma en que el negocio asigna los pagos a los ejecutivos.

Además, existe la columna **mes** y la columna **id_periodo**, se diferencian puesto que el mes corresponde al mes calendario mientras que el periodo se refiere al mes que considera en curso el mandante, los cuales pueden variar en algunos casos dependiendo de las fechas de corte de cada mandante. Estas fechas de corte se almacenan en otra tabla para realizar la asignación correspondiente.

En la figura 28 se observa la estructura de estas tablas de reporte.

Por otra parte, se creó una base de datos específica para guardar datos sobre las tablas del *data mart* como las fechas de inicio y fin de ejecución de los *ETL*, cuántos datos se cargaron y e información utilizada para los *ETL* como por ejemplo el mapeo de categorías y subcategorías de gestión.

3.5. Diseño físico

Luego de definir el modelado dimensional del *data mart*, se procede a crear las tablas correspondientes en *Microsoft SQL Server*.

Uno de los desafíos del *data mart* es cómo sobrellevar la gran cantidad de datos, especialmente de las gestiones. Como se ha mencionado anteriormente, las consultas en NetPag se suelen realizar por periodo y por mandante.

Para ello, se crean particiones en las tablas Gestión_Llamada, Gestión_SMS, Gestión_Bot,

Pago_Asignado	
FK	id_deudor int
FK	id_deuda int
FK	id_mandante int
	documento varchar(100)
	id_periodo int
	monto_pago int
	fecha_pago_dia int
	hora_pago varchar(8)
	fecha_pago datetime
	tipo_pago varchar(20)
	medio_pago varchar(20)
	canal_pago varchar(20)
	mes int
	fecha datetime
FK	id_usuario int
FK	id_gestion int
FK	id_categoria int
FK	id_subcategoria int
FK	id_categoria_mandante int
FK	id_canal int

Gestión_Teléfono_{Mes}	
FK	id_deudor int
	id_gestion int
	id_llamada int
	id_campana varchar(15)
	fecha_dia int
	hora varchar(8)
	fecha datetime
	mes int
FK	id_usuario int
FK	id_categoria int
FK	id_subcategoria int
FK	telefono int
FK	id_canal int
	costo decimal
	duracion int
FK	id_tipo int
FK	id_categoria_mandante int
FK	id_estado int
FK	id_proveedor int
	peso_categoria int
	ranking int

Figura 28: Tablas del *Data Mart* utilizadas como reportes.
Fuente: Elaboración propia.

Gestión_Correo, Deuda y Pago por la columna mes o periodo.

Además, se agregan índices agrupado por mes/periodo como también índices no agrupado por mandante.

Por otra parte, se definen los usuarios actuales que tendrán acceso a la base de datos para evitar la filtración de datos delicados como información de los deudores (RUTs, números de teléfonos, correos, direcciones, etc.).

3.6. Diseño y desarrollo del ETL

3.6.1. Elección de tecnologías

Luego de estudiar las diferentes tecnologías comúnmente utilizadas para los procesos *ETL*, se escogió Apache Spark, específicamente para Python, Pyspark.

Se mencionó el uso de *Talend* en la empresa, el cual es utilizado principalmente para procesos de automatización y *ETL* para el área *B2B*, sin embargo, esta área trabaja con una cantidad menor de datos que el área *B2C* ya que esta última se caracteriza por el envío de masividades.

Se decide utilizar PySpark por sus beneficios al tratar con grandes cantidades de datos, el cual es el caso del área *B2C*, donde los procesos *ETL* requerirán tratamientos complejos de estos datos. Para ello, se aprovecha una estructura distribuida utilizando contenedores de *Docker*.

PySpark permite el procesamiento distribuido de datos, lo que mejora significativamente la velocidad y eficiencia del procesamiento de datos en comparación con soluciones no distribuidas.

El uso de contenedores de *Docker* facilita la implementación y escalabilidad de los procesos *ETL* con PySpark. Permiten un despliegue consistente y aislado, asegurando que los procesos se ejecuten de manera uniforme en diferentes entornos y facilitando la migración de las máquinas en caso de que ocurra, solo teniendo que configurar las conexiones pero manteniendo las configuraciones e instalaciones necesarias para ejecutar los procesos *ETL*.

Por otra parte, *Python* es un lenguaje de programación fácil y simple de aprender, facilitando que el personal de NetPag pueda agregar o mantener procesos *ETL* al *data mart* como facilita el desarrollo de procesos *ETL* complejos y personalizados según las necesidades específicas del área.

Apache Spark tiene una comunidad activa, una documentación detallado y gran cantidad de recursos, sirviendo como apoyo para el desarrollo y entendimiento del *data mart*.

3.6.2. ETL

La extracción de los datos para el *data mart* se realizará utilizando las siguientes fuentes de datos ya mencionadas anteriormente:

- API de proveedor de correos electrónicos masivos.
- API de proveedor de llamadas de bots.
- Base de datos en nube de *Microsoft SQL Server* cargada por proveedores del *software* de cobranza.
- Tablas ya existentes en base de datos de NetPag.
- API de proveedor de teléfonos y correos.

Además, se utilizarán *datasets* externos para complementar información, tales como:

- Nombres registrados en Chile entre 1920 y 2021 con su sexo. [Rivaquiroga,]
- Bases de numeración telefónica de *Subtel*. [de Telecomunicaciones de Chile, 2024]
- Regiones, provincias, ciudades y comunas de Chile.

Es importante destacar que los mandantes cuentan con el consentimiento necesario para el tratamiento de los datos personales de sus deudores, lo cual permite a NetPag gestionar información sensible como RUTs, nombres, direcciones, números de teléfono y correos electrónicos. Estos datos son esenciales para llevar a cabo el proceso de cobranza, respetando la normativa vigente en cuanto a protección de datos.

Además, otro proceso que involucra datos sensibles es la gestión de reclamos de deudores. En estos casos, cuando un deudor reporta haber sido contactado erróneamente, se solicita su RUT, junto con el número de teléfono o correo electrónico afectado, para asegurar que no se le vuelva a contactar indebidamente. Esta información se almacena en las tablas *Bloqueo_Correo* y *Bloqueo_Teléfono* presentadas anteriormente.

Para los procesos *ETL* se utilizará *PySpark*, aprovechando su capacidad en el procesamiento masivo de datos, especialmente para carga y lectura de los datos.

Por otra parte, los *ETL* que requieren operaciones de actualización o cargas ligeras harán uso de *SQLAlchemy*, otra librería de *Python*.

La arquitectura del data mart se resume en la figura 6.

El proceso de *ETL* se realizará al final del día cuando el servidor no esté siendo utilizado para la Operación. En primer lugar, se cargan los datos extraídos de *APIs* y archivos fuente, para

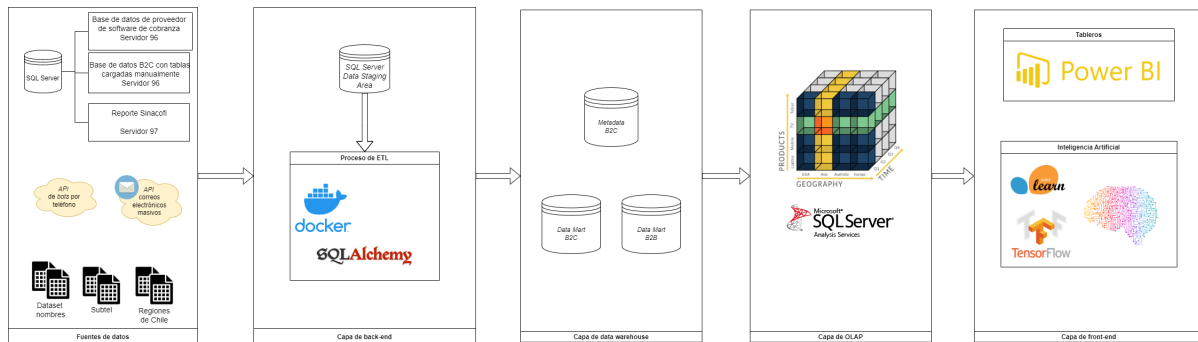


Figura 29: Arquitectura del data mart
Fuente: Elaboración propia.

luego enfocarse en los datos extraídos de la base de datos del proveedor de *software* después de la actualización de las bases de datos fuente con los datos del día.

El flujo de los *ETL* se presenta en 30, mostrando el origen de los datos para cada *ETL* y las tablas del *data mart* que son alimentadas. El orden de cada *ETL* es importante para realizar la unión entre tablas.

Se puede notar cómo los *ETL* utilizan como origen de datos algunas tablas ya existentes en bases internas de NetPag, que a su vez se alimentan de otras fuentes como *Excel* o *APIs*.

Existen *ETL* donde el tratamiento de los datos varía mucho dependiendo del formato de las tablas dependiendo del mandante. Un ejemplo de esto son los pagos y las deudas donde para estandarizar nombres de columnas se crean vistas en *SQL Server* previamente.

Profundizando en el código *ETL*, hay operaciones de transformación y limpieza que se repiten en los procesos, por ejemplo:

- **Transformación de ruts:** Dependiendo del origen de datos y mandante, el rut viene en distintos formatos ya sea con ceros adelante del número, con o sin dígito verificador. El rut se estandariza al formato entero sin dígito verificador.

```
#Se crea una nueva columna, rut_gestion para el rut sin dígito
verificador. Para ello, se separa la parte del rut antes y
después del guión para luego utilizar la primera parte y
transformarla a entero.
df_gestiones = df_gestiones.withColumn('rut_gestion',
    (split(df_gestiones['RUT'], '-').getItem(0)).cast("int"))

if((cod_mandante=='CLIENTE10') or (cod_mandante=='CLIENTE11') or
    (cod_mandante=='CLIENTE12')):
    #Estos son casos donde el rut viene con dígito verificador pero
    sin guión. Se quita el ultimo caracter del rut y se transforma
    a entero.
```

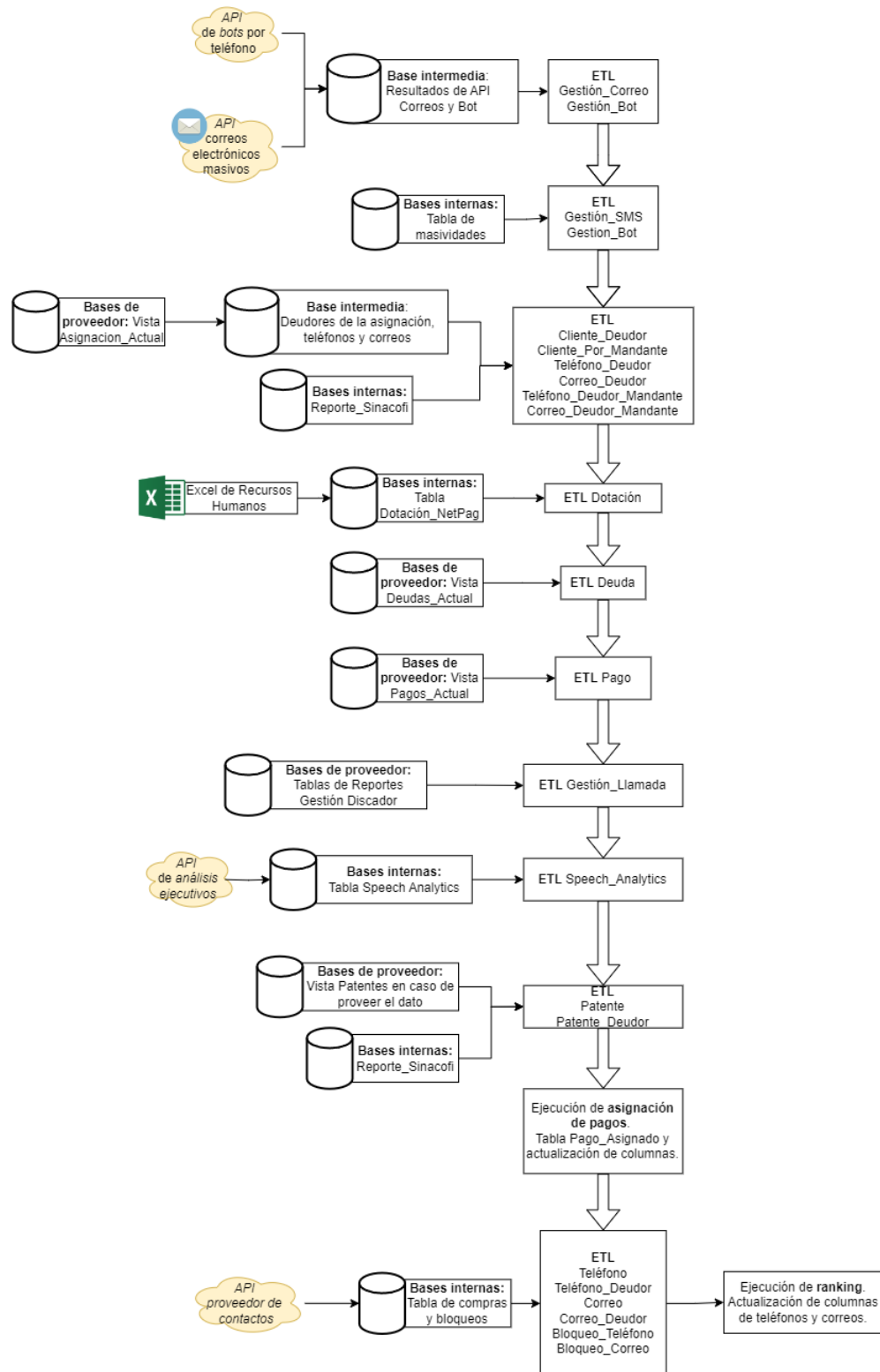


Figura 30: Flujo de los ETL para el data mart
Fuente: Elaboración propia.

```
df_gestiones = df_gestiones.withColumn('rut_gestion',
    expr("substring(rut, 1, length(rut)-1)").cast("int"))

#El rut es una columna esencial para unir con la tabla de
#Cliente_Deudor
df_gestiones = df_gestiones.join(df_clientedeudor,
    df_clientedeudor['rut']==df_gestiones['rut_gestiones'], 'left')
```

Código 1: ETL en Pyspark para la limpieza de ruts.

- **Limpieza de nombres y unión con tablas:** Este proceso es necesario cuando hay que unir el usuario del ejecutivo con la dotación de NetPag, como en la carga de la tabla Usuario y Speech_Analytics. Se realiza utilizando expresiones regulares para encontrar coincidencias entre el nombre del usuario y de dotación, como también aplicando funciones de similitud para abarcar casos borde.

```
"""Funcion para verificar coincidencias entre dos nombres
Recibe la expresión regular del primer nombre y el segundo nombre
Retorna True si encuentra coincidencias y False en caso contrario
"""
def cumple_regex_usuario(reg, nombre):
    #Si se recibe un nombre nulo, se retorna False
    if nombre == None:
        return False

    #Se busca coincidencias entre la expresion regular del primer
    nombre y el segundo nombre
    x = re.search(reg, nombre)

    #Se invierten los casos, expresión regular del segundo nombre y
    el primer nombre normal
    nombre2 = reg.replace('.', '+', ' ')
    reg2 = nombre.replace(' ', '.+')

    #Se busca coincidencia entre la expresión regular del segundo
    nombre y el primer nombre
    y = re.search(reg2, nombre2)

    if x or y:
        #Si alguno de los casos coincide, se retorna True
        return True
    else:
        #En caso contrario, se utiliza función de similitud de nombres
        similarity = jellyfish.jaro_winkler_similarity(nombre, nombre2)

        #Si la similitud es mayor al 90, se retorna True
        if (similarity>0.9):
            return True
        return False
```

Código 2: Función de *Python* para buscar coincidencias entre dos nombres

Luego, en el código 3 se aplica la función en el *dataframe* correspondiente en *Pyspark*.

```
#Creación de la expresion regular para el nombre del usuario/ejecutivo
df_usuario = df_usuario.withColumn("usuario_nombre",
    regexp_replace(upper(col("usuario")), ' ', '.+'))

#Se realiza el join con dotación, buscando coincidencias de nombre y
que el registro de dotación sea valido con el mes de la gestión.
df_usuario_join = df_usuario.join(df_dotacion,
    (cumple_regex_usuario_udf(df_usuario['usuario_nombre'],
    df_dotacion['nombre'])) &
```

```
(df_usuario['MES']>=df_dotacion['valido_desde']) &  
(df_usuario['MES']<=df_dotacion['valido_hasta']), 'inner')
```

Código 3: ETL en Pyspark para realizar la unión con Dotación

- **Transformación de fechas:** El dato de fechas y hora se repite en varios ETL, representando información valiosa para NetPag, especialmente en el análisis de las gestiones. En Pyspark, la transformación de fechas se puede realizar fácilmente como se presenta en el código 4.

```
#Transformación de horas a partir de fecha original  
df = df.withColumn("hora", date_format(col("FECHA"), "HH:mm:ss"))  
df = df.withColumn("hora_inicio", when(col("INICIO").isNotNull(),  
    date_format(col("INICIO"), "HH:mm:ss"))).\  
    withColumn("hora_inicio", when(col("hora_inicio")=='00:00:00',  
        lit(None).cast("string")))  
df = df.withColumn("hora_conexion",  
    when(col("CONEXION").isNotNull(),  
        date_format(col("CONEXION"), "HH:mm:ss"))).\  
    withColumn("hora_conexion",  
        when(col("hora_conexion")=='00:00:00',  
            lit(None).cast("string")))  
df = df.withColumn("hora_fin", when(col("FIN").isNotNull(),  
    date_format(col("FIN"), "HH:mm:ss"))).\  
    withColumn("hora_fin", when(col("hora_fin")=='00:00:00',  
        lit(None).cast("string")))  
  
#Transformación del día a partir de fecha original  
df = df.withColumn("fecha_dia",  
    date_format(col("FECHA"), "yyyyMMdd").cast("int"))  
  
#Transformación de fecha original a formato aceptado por SQL Server  
df = df.withColumn("fecha", date_format(col("FECHA"), "dd-MM-yyyy  
    HH:mm:ss"))
```

Código 4: ETL en Pyspark para transformación de fechas

Una de las tablas con mayor procesamiento es **Cliente_Deudor** debido a que los deudores se pueden repetir por mandantes. Al juntar la información de los mandantes, se obtienen varios nombres, direcciones y otros datos como fecha de nacimiento. Por ello, se debe incluir en el ETL la elección de uno de estos datos para agregar a la tabla. En el código 5 se presenta un ejemplo de elección de nombre.

```
#Se agrupan los clientes deudores por rut, obteniendo un set de todos los  
    nombres, direcciones y fechas de nacimiento y escogiendo el máximo de  
    edad y género.
```

```
df_ruts = df_ruts.groupBy("rut").agg(
    collect_set("nombre").alias("nombre"),
    collect_set("direccion").alias("direccion"),
    collect_set("fecha_nacimiento").alias("fecha_nacimiento"),
    max("edad").alias("edad"),
    max("genero").alias("genero")
)

#Se aplica la función definida
df_ruts = df_ruts.withColumn("nombre_limpio",
    when(col("nombre").isNotNull(),
        eleccion_nombre_udf(col("nombre")))).drop("nombre")

#Columnas apellido_pat, apellido_mat, nombres, nombre_completo
df_ruts = df_ruts.select("?", "nombre_limpio.nombres",
    "nombre_limpio.apellido_pat", "nombre_limpio.apellido_mat",
    "nombre_limpio.nombre")\
    .withColumnRenamed("nombre", "nombre_completo") \
    .withColumnRenamed("apellido_pat", "apellido1") \
    .withColumnRenamed("apellido_mat", "apellido2") \
    .drop("nombre_limpio")
```

Código 5: ETL en Pyspark para aplicar la elección de nombres en Cliente_Deudor

En el anexo D, se presenta la función elección_nombre aplicada.

Por otra parte, para operaciones de actualización, por ejemplo, para la actualización del ranking según su categoría y fecha se utiliza directamente consultas SQL ejecutándolas mediante la librería *SQLAlchemy* de *Python*. En el código 6 se presenta la consulta SQL utilizada, donde el mes es una variable de *Python* que representa el mes calendario en curso.

Esta consulta utiliza la función ROW_NUMBER de SQL Server para obtener la gestión con el mejor ranking para cada teléfono. Luego, a partir de esas gestiones se calcula el ranking de teléfonos para un cliente deudor según el mandante en el que se encuentra.

```
WITH cte_ranked AS (
    SELECT
        gl.id_deudor,
        gl.telefono,
        gl.id_mandante,
        gl.mes,
        gl ranking,
        gl.fecha,
        cat.categoria,
        ROW_NUMBER() OVER (
            PARTITION BY gl.id_deudor, gl.id_mandante, gl.mes, gl.telefono
            ORDER BY gl ranking ASC, gl.fecha DESC
```

```
        ) AS row_num
FROM
    Gestion_Telefono_{mes} gl
INNER JOIN
    Categoria cat ON gl.id_categoria = cat.id_categoria
WHERE
    gl.mes = {mes}
),
cte_final_ranking AS (
    SELECT
        id_deudor,
        telefono,
        id_mandante,
        mes,
        ranking as ranking_gestion,
        fecha,
        categoria,
        ROW_NUMBER() OVER (
            PARTITION BY id_deudor, id_mandante, mes
            ORDER BY ranking ASC, fecha DESC
        ) AS ranking
    FROM
        cte_ranked
    WHERE
        row_num = 1
)
UPDATE tdm
SET
    tdm.ranking = cte_final_ranking.ranking,
    tdm.fecha_ranking = cte_final_ranking.fecha,
    tdm.mes_ranking = cte_final_ranking.mes,
    tdm.tipo_contacto = cte_final_ranking.categoria
FROM
    Telefono_Deudor_Mandante tdm
INNER JOIN
    cte_final_ranking
    ON tdm.id_deudor = cte_final_ranking.id_deudor
    AND tdm.telefono = cte_final_ranking.telefono
    AND tdm.id_mandante = cte_final_ranking.id_mandante
WHERE
    cte_final_ranking.mes >= tdm.mes_ranking OR tdm.mes_ranking IS NULL
```

Código 6: Código SQL para la actualización del ranking en la tabla Telefono_Deudor_Mandante

3.7. Diseño de la aplicación de inteligencia de negocios

Considerando el uso de aplicaciones *Microsoft* y la experiencia con *Power BI* en la empresa, se decide utilizar esta aplicación para la inteligencia de negocios, específicamente, para la creación de *dashboards*.

3.8. Mantenimiento

Si bien los procesos en el *data mart* están automatizados, es importante estar atento a que las ejecuciones de *ETL* y carga de datos se hayan realizado correctamente. En caso de error, los procesos envían un correo de notificación al área de datos e inteligencia de negocios.

Por otra parte, puede ocurrir que los procesos terminen su ejecución correctamente pero el valor de las columnas no sea consistente, por ejemplo, al tener muchos valores nulos. En esos casos, es necesario realizar una revisión sobre qué puede estar sucediendo.

Por lo anterior, es necesario una revisión y mejora continua del *data mart*, poniendo atención a la retroalimentación de los usuarios para identificar oportunidades de mejora.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

Al tener los datos cargados en el *data mart*, se generan las instancias para realizar las consultas.

Este cambio tiene como objetivo simplificar las consultas utilizadas para el análisis del rendimiento y panorama de NetPag, permitiendo abarcar periodos más amplios, proporcionando más información para el análisis, ahorrando tiempo en el tratamiento de los datos y mejorando los tiempos de respuesta en consultas grandes.

Considerando lo anterior, al tener el *data mart* diseñado e implementado, se necesita verificar su funcionamiento mediante los siguientes objetivos presentados anteriormente:

- Desplegar información del *data mart* mediante un *dashboard* que apoye la toma de decisiones.
- Comparar el rendimiento entre el *data mart* propuesto y el antiguo sistema mediante consultas para verificar una mejora en los tiempos de respuesta.

En el primer punto, el *dashboard* diseñado será en *Microsoft Power BI*, demostrando que es posible realizar la conexión entre este *software* y el *data mart*. Este *dashboard* inicial pretende servir como guía para que el área de Operaciones e Inteligencia de Negocios continúe creando sus propios tableros según sus necesidades y para que puedan observar el funcionamiento del *data mart* y el potencial de obtener información relevante para su toma de decisiones.

En el segundo punto, se comparará cómo se obtendrían los datos de una misma consulta entre el sistema antiguo y el *data mart* propuesto. La forma de realizar consultas varía puesto que los datos se encuentran almacenados de distinta manera. Para el sistema antiguo, no hay una base de datos que centralice los datos del área de negocios a tratar, por lo que las consultas requieren una unión de varias tablas localizadas en distintas bases de datos.

4.1. Diferencias entre el sistema antiguo y el *data mart*

Para abordar el segundo punto, se realizará la comparación de las siguientes consultas SQL

1. Gestiones de llamada según categoría de contacto para el mes de abril del 2024
2. Gestiones de llamada por ejecutivo y categoría de contacto en abril del 2024

3. Recupero versus deuda en los últimos tres meses por mandante

El rendimiento entre el *data mart* y el sistema antiguo será comparado en las siguientes métricas:

1. Tiempos de respuesta de las consultas
2. Complejidad de la consulta

4.1.1. Gestiones de llamada según categoría de contacto para el mes de abril del 2024

Para esta consulta, se limitan los mandantes al rubro de autopistas, debido a que requiere la estandarización de las categorías, el cual es parte de los procesos *ETL* y no se puede replicar directamente con consultas *SQL* utilizando el sistema antiguo.

Consulta mediante el *data mart* en *SQL Server*

```
SELECT c.categoria as Categoria,
count(*) as Cantidad_Gestiones
FROM Gestion_Llamada g
INNER JOIN Categoria c on c.id_categoria = g.id_categoria
INNER JOIN Mandante m on m.id_mandante = g.id_mandante
WHERE m.rubro='AUTOPISTA'
and g.mes=202404
GROUP BY c.categoria
```

Consulta mediante el sistema antiguo en *SQL Server*

```
SELECT Categoria, count(*) as Cantidad_Gestiones from
((SELECT CASE
WHEN UPPER(categoria) = 'CLIENTE FALLECIDO' THEN 'CONTACTO INDIRECTO'
WHEN UPPER(CATEGORIA) = 'CONTACTO TERCERO' THEN 'CONTACTO INDIRECTO'
WHEN UPPER(CATEGORIA) = 'COMPROMISO PAGO' THEN 'COMPROMISO DE PAGO'
WHEN UPPER(CATEGORIA) = 'INBOUND' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'RESOLUCIONES DE SISTEMA' THEN 'SIN CONTACTO'
WHEN UPPER(CATEGORIA) = 'DISCADOR' THEN 'SIN CONTACTO'
WHEN UPPER(CATEGORIA) = 'EQUIVOCADO' THEN 'SIN CONTACTO'
ELSE UPPER(CATEGORIA) END AS Categoria, FONO
FROM PROVEEDOR_CLIENTE1.dbo.REPORTE_GESTION_DISCADOR_ID_202404
WHERE SUBCATEGORIA NOT LIKE '%MAIL%')
UNION ALL
(SELECT CASE
WHEN UPPER(CATEGORIA) = 'TERCERO FAMILIAR' THEN 'CONTACTO INDIRECTO'
WHEN UPPER(CATEGORIA) = 'TERCERO OTRO' THEN 'CONTACTO INDIRECTO'
```

```

WHEN UPPER(CATEGORIA) = 'TITULAR' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'INBOUND' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'RESOLUCIONES DE SISTEMA' THEN 'SIN CONTACTO'
WHEN UPPER(CATEGORIA) = 'DISCADOR' THEN 'SIN CONTACTO'
ELSE UPPER(categoria) END AS Categoria, FONO FROM
    PROVEEDOR_CLIENTE2.dbo.REPORTE_GESTION_DISCADOR_202404
WHERE SUBCATEGORIA NOT LIKE '%MAIL%')
UNION ALL
(SELECT CASE
WHEN UPPER(subcategoria) = 'MENSAJE A TERCERO' THEN 'CONTACTO INDIRECTO'
WHEN UPPER(categoria)='CONTACTADO' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'INBOUND' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'RESOLUCIONES DE SISTEMA' THEN 'SIN CONTACTO'
WHEN UPPER(CATEGORIA) = 'DISCADOR' THEN 'SIN CONTACTO'
ELSE UPPER(CATEGORIA) END AS Categoria, FONO FROM
    PROVEEDOR_CLIENTE3.dbo.REPORTE_GESTION_DISCADOR_202404
WHERE SUBCATEGORIA NOT LIKE '%MAIL%')) g
WHERE Categoria NOT IN ('MAIL', 'SMS', 'BOT')
GROUP BY Categoria

```

El resultado de la consulta se presenta en la tabla 1

Categoría	Cantidad_Gestiones
COMPROMISO DE PAGO	8206
CONTACTO DIRECTO	16272
CONTACTO INDIRECTO	2281
SIN CONTACTO	1302089

Tabla 1: Cantidad de Gestiones por Categoría para abril del 2024

El tiempo de respuesta para la consulta mediante el sistema antiguo fue de 2.598 segundos, mientras que con el *data mart* fue de 0.566 milisegundos.

Se realiza otro experimento al obtener las gestiones de marzo y abril, es decir, dos meses, donde para el sistema antiguo el tiempo fue de 4.055 segundos y para el *data mart* de 0.929 segundos.

Finalmente, consultando por tres meses: febrero, abril y mayo se obtiene un tiempo de 5.371 segundos para el sistema antiguo y de 2.068 segundos para el *data mart*.

Con respecto a la **complejidad** de la consulta, se puede notar como la consulta mediante el *data mart* es considerablemente más breve a diferencia del sistema antiguo, puesto que este último requiere de algunas transformaciones de categoría para poder homologar todo a las cuatro categorías principales: Compromiso de pago, contacto directo, contacto indirecto y sin contacto. En cambio, en el *data mart* esta transformación de datos ya está aplicada y de forma mucho más flexible, puesto que las categorías pueden ir variando a lo largo del

tiempo. Por otra parte, la consulta del sistema antiguo requiere consultar a cada base distinta dependiendo del mandante.

Otro punto importante es que para el *data mart* solo es necesario consultar por el rubro de autopista para obtener los datos de estos mandantes, a diferencia del sistema antiguo que se requiere saber exactamente cuáles son los mandantes que pertenecen a autopistas, teniendo que modificar y aumentar la longitud de la consulta en caso de tener un nuevo mandante.

4.1.2. Gestiones de llamada por ejecutivo y categoría de contacto en abril del 2024

Esta consulta es común en la empresa para revisar el rendimiento de los ejecutivos y analizar cuántos compromisos de pago han logrado durante el mes. Generalmente, esta consulta se realiza por mandante. En esta ocasión, se utilizan los datos de los mandantes en el sector de autopistas para evaluar el rendimiento general de los ejecutivos.

Consulta mediante el *data mart* en SQL Server

```
SELECT US.usuario, Cat.categoria, COUNT(*) as Cantidad_Gestiones
FROM GESTION_LLAMADA GL
LEFT JOIN USUARIO US ON US.id_usuario = GL.id_usuario
LEFT JOIN Categoria Cat ON GL.id_categoria = Cat.id_categoria
INNER JOIN Mandante ON GL.id_mandante = Mandante.id_mandante
WHERE Mandante.rubro = 'AUTOPISTA' AND MES=202404
GROUP BY US.usuario, Cat.categoria
```

Consulta mediante el sistema antiguo en SQL Server

```
SELECT Categoria,Usuario, count(*) as Cantidad_Gestiones from
((SELECT CASE
WHEN UPPER(categoria) = 'CLIENTE FALLECIDO' THEN 'CONTACTO INDIRECTO'
WHEN UPPER(CATEGORIA) = 'CONTACTO TERCERO' THEN 'CONTACTO INDIRECTO'
WHEN UPPER(CATEGORIA) = 'COMPROMISO PAGO' THEN 'COMPROMISO DE PAGO'
WHEN UPPER(CATEGORIA) = 'INBOUND' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'RESOLUCIONES DE SISTEMA' THEN 'SIN CONTACTO'
WHEN UPPER(CATEGORIA) = 'DISCADOR' THEN 'SIN CONTACTO'
WHEN UPPER(CATEGORIA) = 'EQUIVOCADO' THEN 'SIN CONTACTO'
WHEN UPPER(SUBCATEGORIA) = 'SEGUIMIENTO PROMESA' THEN 'COMPROMISO DE PAGO'
ELSE UPPER(CATEGORIA) END AS Categoria, 'CLIENTE1' AS Mandante, Usuario
FROM PROVEEDOR_CLIENTE1.dbo.REPORTE_GESTION_DISCADOR_ID_202404 Gestiones
WHERE Categoria NOT IN ('MAIL', 'SMS', 'BOT') AND SUBCATEGORIA NOT
LIKE '%MAIL%')
UNION ALL
(SELECT CASE
WHEN UPPER(CATEGORIA) = 'TERCERO FAMILIAR' THEN 'CONTACTO INDIRECTO'
```

```
WHEN UPPER(CATEGORIA) = 'TERCERO OTRO' THEN 'CONTACTO INDIRECTO'
WHEN UPPER(CATEGORIA) = 'TITULAR' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'INBOUND' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'RESOLUCIONES DE SISTEMA' THEN 'SIN CONTACTO'
WHEN UPPER(CATEGORIA) = 'DISCADOR' THEN 'SIN CONTACTO'
ELSE UPPER(categoria) END AS Categoria, 'CLIENTE2' as Mandante, Usuario
FROM PROVEEDOR_CLIENTE2.dbo.REPORTE_GESTION_DISCADOR_202404 Gestiones
    WHERE Categoria NOT IN ('MAIL', 'SMS', 'BOT') AND SUBCATEGORIA NOT
    LIKE '%MAIL%')
UNION ALL
(SELECT CASE
WHEN UPPER(subcategoria) = 'MENSAJE A TERCERO' THEN 'CONTACTO INDIRECTO'
WHEN UPPER(categoria)='CONTACTADO' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'INBOUND' THEN 'CONTACTO DIRECTO'
WHEN UPPER(CATEGORIA) = 'RESOLUCIONES DE SISTEMA' THEN 'SIN CONTACTO'
WHEN UPPER(CATEGORIA) = 'DISCADOR' THEN 'SIN CONTACTO'
ELSE UPPER(CATEGORIA) END AS Categoria, 'CLIENTE3' AS Mandante, Usuario
FROM PROVEEDOR_CLIENTE3.dbo.REPORTE_GESTION_DISCADOR_202404 Gestiones
    WHERE Categoria NOT IN ('MAIL', 'SMS', 'BOT') AND SUBCATEGORIA NOT
    LIKE '%MAIL%')) g
GROUP BY Categoria, Usuario
```

El resultado de la consulta se puede resumir en la figura 31.

El tiempo de respuesta para la consulta mediante el sistema antiguo fue de 2.810 segundos, mientras que con el *data mart* fue de 1.428.

Nuevamente, se realiza el experimento para 2 meses, abril y marzo, obteniendo un tiempo para el sistema antiguo de 2.925 segundos y de 2 segundos para el *data mart*.

Si se considera tres meses: febrero, marzo y abril, se llega a un tiempo de 4 segundos para el sistema antiguo y de 3 segundos para el *data mart*.

Con respecto a la **complejidad** de la consulta, se puede notar nuevamente como la consulta mediante el *data mart* es más concisa con respecto al sistema antiguo. Al igual que antes, una de las grandes diferencias es la transformación ya aplicada en el *data mart* de las categorías. Por otra parte, esta consulta en el sistema antiguo accede a las mismas bases y tablas que la consulta anterior, por lo que requiere lectura a tres bases de datos distintas.

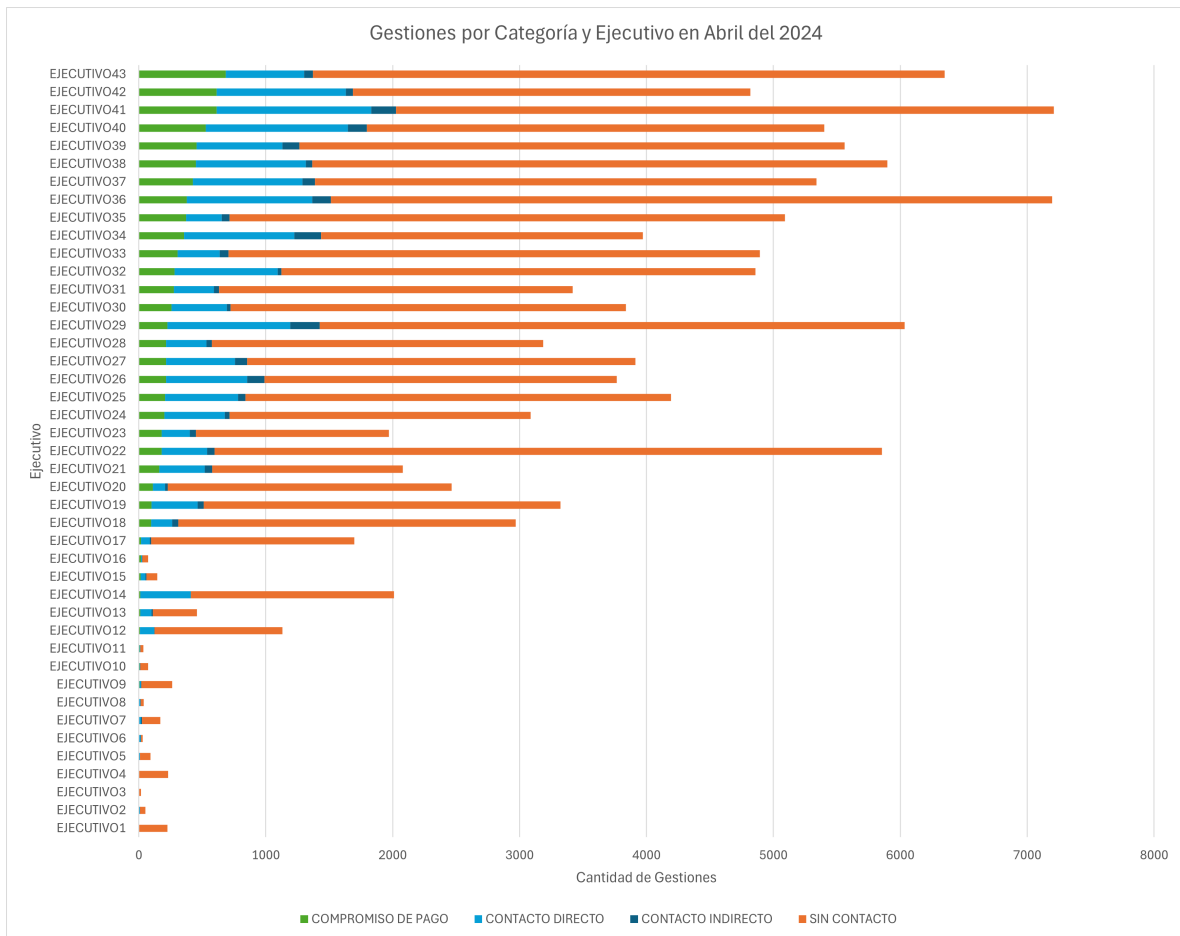


Figura 31: Gestiones por Categoría y Ejecutivo en Abril del 2024.

4.1.3. Recupero versus deuda en los últimos tres meses por mandantes del rubro de autotopistas

Consulta mediante el *data mart* en SQL Server

```
WITH DeudaAcumulada AS (
    SELECT
        id_mandante,
        id_perodo AS Periodo,
        SUM(CAST(monto_total AS BIGINT)) AS DeudaTotal
    FROM
        Deuda
    WHERE id_perodo >= 202402 AND id_perodo <= 202404
    GROUP BY id_mandante, id_perodo),
PagoAcumulado AS (
    SELECT
        id_mandante,
        id_perodo AS Periodo,
```

```

        SUM(CAST(monto_pago AS BIGINT)) AS PagoTotal
FROM
    Pago
WHERE id_periodo >= 202402 AND id_periodo <= 202404
GROUP BY id_mandante, id_periodo)
SELECT
    Mandante.cod_mandante as Mandante,
    d.Periodo,
    ISNULL(d.DeudaTotal, 0) AS TotalDeuda,
    ISNULL(p.PagoTotal, 0) AS TotalPago,
    ROUND((CAST(p.PagoTotal AS FLOAT) /
    CAST(d.DeudaTotal AS FLOAT)) * 100,2) AS PorcentajeRecupero
FROM
    DeudaAcumulada d
LEFT JOIN
    PagoAcumulado p ON d.PERIODO = p.PERIODO and d.id_mandante =
        p.id_mandante
INNER JOIN Mandante ON d.id_mandante = Mandante.id_mandante
WHERE Mandante.rubro = 'AUTOPISTA'

```

Consulta mediante el sistema antiguo en SQL Server

Esta consulta se presenta en el anexo E.

Como se puede notar, la consulta es extensa y se deben incluir las reglas de negocio necesarias para obtener los datos correctos. Por ejemplo, los pagos menores a 10 pesos no son validos puesto que son cargos por el uso de tarjeta. Además, es necesario agrupar los pagos para eliminar los datos duplicados.

El resultado para ambas consultas se presenta en la tabla 2 y se puede resumir en el gráfico 32.

Mandante	Periodo	TotalDeuda	TotalPago	PorcentajeRecupero
CLIENTE1	202402	51218904006	1045471751	2.04
CLIENTE1	202403	65137072270	11641045600	17.87
CLIENTE1	202404	54465575924	6960842295	12.78
CLIENTE2	202404	24483691431	585190171	2.39
CLIENTE2	202403	21113767114	561718189	2.66
CLIENTE2	202402	20362189626	527535148	2.59
CLIENTE3	202403	1636375077	702312030	42.92
CLIENTE3	202404	2339185551	821346633	35.11
CLIENTE3	202402	1940981638	958007769	49.36

Tabla 2: Datos de Mandantes, Periodos, Total Deuda, Total Pago y Porcentaje de Recupero

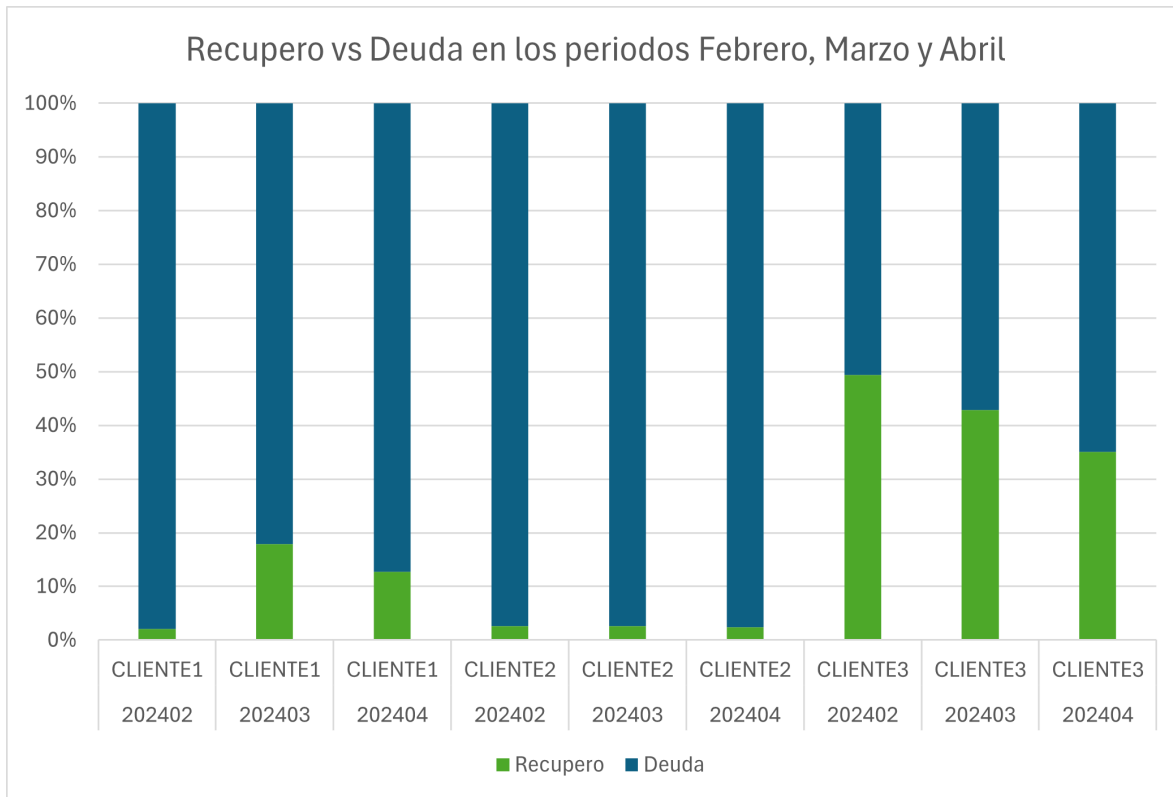


Figura 32: Recupero versus Deuda para los periodos Febrero, Marzo y Abril

El tiempo de respuesta para la consulta mediante el sistema antiguo fue de 4.620 segundos, mientras que con el *data mart* fue de 1 segundo.

Con respecto a la **complejidad** de la consulta, las dos son complejas puesto que requieren comparar valores de tablas distintas, sin embargo, se puede notar cómo la consulta mediante el sistema antiguo se vuelve muy extensa y requiere agregar las reglas de negocio explícitamente, lo cual ya está incluido en el *data mart*. Además, la consulta antigua debe consultar en distintas bases y tablas para obtener los datos, a diferencia del *data mart* que accede solo a la tabla Deuda y Pago.

La tabla 3 resume el rendimiento de las consultas presentadas tanto por tiempo total, de CPU y lecturas físicas y lógicas.

Se puede notar que los tiempos de CPU y total para el *data mart* son considerablemente menores en comparación con el sistema antiguo, por lo que el procesamiento en el *data mart* es más rápido. Por otra parte, el menor valor de las lecturas físicas y lógicas para el *data mart* significa un manejo más eficiente de los datos, esto debido a la simplicidad de la consulta como al uso de particiones por mes en las tablas.

Al tener una menor carga de CPU se liberan recursos del servidor, permitiendo que varios usuarios de NetPag puedan realizar consultas sin que se degrade el rendimiento.

Consulta	Método	Tiempo CPU (ms)	Tiempo total (ms)	Lecturas lógicas	Lecturas físicas
Deuda vs Pago	Sistema antiguo	15807	4465	501539	498976
Deuda vs Pago	Data Mart	6083	1343	168697	167030
Usuarios en 1 mes	Sistema antiguo	15079	4272	611724	483166
Usuarios en 1 mes	Data Mart	4124	1311	163364	112755
Usuarios en 2 meses	Sistema antiguo	20144	2803	620590	314687
Usuarios en 2 meses	Data Mart	6198	2079	211220	0
Usuarios en 3 meses	Sistema antiguo	30666	3985	900649	461400
Usuarios en 3 meses	Data Mart	10300	3767	266134	53702
Categorías en 1 mes	Sistema antiguo	33798	2519	305862	305654
Categorías en 1 mes	Data Mart	2860	395	163359	0
Categorías en 2 meses	Sistema antiguo	96434	3998	620590	314834
Categorías en 2 meses	Data Mart	4625	853	211280	29039
Categorías en 3 meses	Sistema antiguo	161041	5289	900649	280005
Categorías en 3 meses	Data Mart	6874	1539	266192	0

Tabla 3: Comparación de rendimiento entre el sistema antiguo y Data Mart

4.2. Validación de los datos

Debido a la complejidad de los datos y de los procesos *ETL*, durante el desarrollo se enfrentaron desafíos con respecto a la consistencia de los datos transformados con los datos de origen nativo, donde la mayoría se iban solucionando a medida que se encontraba el origen del error.

Una inconsistencia desde el origen de datos que se traspasa al data mart son RUTs mal ingresados, generalmente números muy pequeños, debido a que no corresponden con la persona. Este error en el origen de datos se mantiene en las tablas de Contactos como también de Gestiones. El especificar la clave primaria de Cliente_Deudor como un número autoincrementable permite modificar RUTs en caso de realizar las correcciones sin tener que modificar las tablas relacionadas. Como solución, se crea además una tabla intermediaria de correcciones de RUTs, especificando el RUT mal ingresado, RUT corregido y mandante al que pertenece para poder aplicar esta corrección a datos futuros durante el *ETL*. El inconveniente de esta solución es que son correcciones que se deben ingresar manualmente.

Aunque esta inconsistencia tiene un impacto leve, puede afectar el análisis de datos, especialmente si la misma persona ya se encuentra registrada en la tabla Cliente_Deudor con otro RUT. Esto puede influir en el ranking de la tabla Teléfono_Deudor, ya que al contener distintos registros de deudores y teléfonos, el ranking se calcula por separado para lo que en realidad debería ser un único deudor. Como resultado, se podría llegar a conclusiones erróneas sobre la frecuencia de contacto o la efectividad de las gestiones.

Por otra parte, se verificó que la cantidad de datos fuese consistente entre el origen de datos y el data mart, como también en los valores de sus columnas.

En la tabla 4 se puede visualizar algunas gestiones desde el origen de datos, mientras que en la tabla se pueden visualizar estas mismas gestiones en el data mart, notando cómo en

este último se puede obtener una mayor cantidad de información referente al ejecutivo y al deudor contactado. Se censura información sensible como nombres y RUTs de personas.

Algunas diferencias que se pueden notar directamente en estos datos son la diferencia de formatos en las fechas de inicio, conexión y fin de la llamada, notando cómo en el origen de datos cuando no hay conexión con el destinatario se visualiza la fecha como 2000-01-001 00:00:00Z, lo cual puede ser confuso a simple vista o se puede creer que hay un error en los datos, a diferencia del *data mart* donde el horario de conexión se encuentra vacío, pudiendo notar fácilmente que en esa llamada no hubo una conexión. Además, para una mejor visualización se almacena solo los horarios de inicio, conexión y fin y no la fecha completa, ya que esta se encuentra definida en la columna fecha o fecha_dia. Esta transformación de fechas se puede revisar en el código 4.

Por otra parte, en estas tablas se puede visualizar la transformación de las categorías mencionada anteriormente, donde el *data mart* guarda tanto la categoría desde el origen de datos como la categoría asignada por ETL en el *data mart*. A su vez, al unir la gestión con la tabla Telefono_Deudor_Mandante se puede conocer el ranking del teléfono contactado.

En conclusión, si bien puede haber errores o inconsistencias en los datos que vienen de la operación como RUTs mal ingresados, se mantiene una consistencia entre los datos del origen y del *data mart* luego de las limpiezas y transformaciones correspondientes para la carga al *data mart*, cumpliendo con el tercer objetivo “Implementar el diseño propuesto, incluyendo las operaciones de extracción, transformación y limpieza de los datos para garantizar la calidad de estos”.

id_llamada	rut	fecha	usuario	categoría	subcategoría	fono	inicio
165011130	966867XX-X	2024-07-15 14:24:51Z	1071 - Georgina Bxxxx	EQUIVOCADO	EQUIVOCADO	22540XXXX	
165010812	274607XX-X	2024-07-15 14:26:00Z	1619 - Eduarver Bxxxx	SIN CONTACTO	NO CONTESTA	99932XXXX	
162691338	9403XXX-X	2024-07-06 10:06:56Z	1088 - Cesar Txxxx	SIN CONTACTO	BUZON DE VOZ	93280XXXX	2024-07-06 10:06:51Z
162691506	76557XXX-X	2024-07-06 10:08:07Z	1549 - Emperatriz Bxxxx	CONTACTO DIRECTO	SEGUIMIENTO PROMESA	99905XXXX	2024-07-06 10:07:23Z
162691699	19341XXX-X	2024-07-06 10:08:32Z	1088 - Cesar Txxxx	SIN CONTACTO	BUZON DE VOZ	99023XXXX	2024-07-06 10:07:51Z
168562394	14305XXX-X	2024-07-29 11:39:47Z	1162 - Cecilia Rxxxx	SIN CONTACTO	NO CONTESTA	93655XXXX	2024-07-29 11:39:27Z
168562119	75593XXX-X	2024-07-29 11:39:17Z	1071 - Georgina Bxxxx	SIN CONTACTO	NO CONTESTA	98541XXXX	2024-07-29 11:39:01Z
164599720	78297XXX-X	2024-07-13 11:00:24Z	1567 - Sofia Sxxxx	SIN CONTACTO	FUERA DE SERVICIO	92415XXXX	2024-07-13 11:00:14Z
164599191	77163XXX-X	2024-07-13 10:59:52Z	1071 - Georgina Bxxxx	SIN CONTACTO	NO CONTESTA	99862XXXX	2024-07-13 10:59:26Z
167053401	18665XXX-X	2024-07-23 16:18:35Z	1619 - Eduarver Bxxxx	SIN CONTACTO	NO CONTESTA	93002XXXX	2024-07-23 16:11:08Z

Tabla 4: Llamadas desde el origen de datos

id_llamada	conexion	fin	direccion	costo1	duracion	tipo_discador
165011130						
165010812						
162691338	2000-01-01 00:00:00Z	2024-07-06 10:06:58Z	SALIENTE	0.0000	0	Predictive
162691506	2000-01-01 00:00:00Z	2024-07-06 10:07:30Z	SALIENTE	0.0000	0	Predictive
162691699	2024-07-06 10:07:57Z	2024-07-06 10:08:09Z	SALIENTE	0.0000	12	Predictive
168562394	2000-01-01 00:00:00Z	2024-07-29 11:39:30Z	SALIENTE	0.0000	0	Predictive
168562119	2000-01-01 00:00:00Z	2024-07-29 11:39:15Z	SALIENTE	0.0000	0	Predictive
164599720	2000-01-01 00:00:00Z	2024-07-13 11:00:14Z	SALIENTE	0.0000	0	Predictive
164599191	2024-07-13 10:59:40Z	2024-07-13 10:59:40Z	SALIENTE	0.0000	0	Predictive
167053401	2024-07-23 16:11:24Z	2024-07-23 16:11:29Z	SALIENTE	0.0000	5	IVR

Tabla 4: Llamadas desde el origen de datos (Continuación)

DISEÑO E IMPLEMENTACIÓN DE UN DATA MART PARA APOYAR LA TOMA DE DECISIONES EN UNA EMPRESA DE COBRANZA EN SU LÍNEA DE NEGOCIO B2C

id_llamada	fecha_dia	hora	hora_inicio	hora_conexion	hora_fin	telefono	fecha	costo	duracion	tipo_discador	rut_dv	nombre_completo
165011130	20240715	14:24:51				225408XXX	15-07-2024 14:24:51.8170000				96686XXX-X	MATERIALES DE CONSTRUCCION SOXXXXXXXX S A
165010812	20240715	14:26:00				999323XXX	15-07-2024 14:26:00.2300000				27460XXX-X	EJOBE ANTONIO MXXXXXX
162691338	20240706	10:06:56	10:06:51		10:06:58	932802XXX	06-07-2024 10:06:56.7130000	0	0	DISCADOR PREDICTIVO	940XXX-X	MARIA LORENA MXXXXXXXX RXXXXXXXX
162691506	20240706	10:08:07	10:07:23		10:07:30	999056XXX	06-07-2024 10:08:07.7200000	0	0	DISCADOR PREDICTIVO	76557XXX-X	INGXXXXXXXX CHILE LIMITADA
162691699	20240706	10:08:32	10:07:51	10:07:57	10:08:09	990236XXX	06-07-2024 10:08:32.5970000	0	12	DISCADOR PREDICTIVO	19341XXX-X	LETICIA CXXXXX SXXX
168562394	20240729	11:39:47	11:39:27		11:39:30	936551XXX	29-07-2024 11:39:47.1830000	0	0	DISCADOR PREDICTIVO	14305XXX-X	MANUEL FRANCISCO JXXXXXX
168562119	20240729	11:39:17	11:39:01		11:39:15	985417XXX	29-07-2024 11:39:17.4130000	0	0	DISCADOR PREDICTIVO	7559XXX-X	JULIO COSTA AXXXXXX
164599720	20240713	11:00:24	11:00:14		11:00:14	924150XXX	13-07-2024 11:00:24.8530000	0	0	DISCADOR PREDICTIVO	78297XXX-X	RENXX UNO LIMITADA
164599191	20240713	10:59:52	10:59:26	10:59:40	10:59:40	998626XXX	13-07-2024 10:59:52.2530000	0	0	DISCADOR PREDICTIVO	77163XXX-X	TRANSPORTES Y REPARACIONES DXXXXX FXXXX
167053401	20240723	16:18:35	16:11:08	16:11:24	16:11:29	930028XXX	23-07-2024 16:18:35.2600000	0	5	IVR	18665XXX-X	CRISTIAN ALEXANDER IXXXXXX RXXXXXXXX

Tabla 5: Datos de llamadas desde el data mart

id_llamada	direccion_llamada	categoria	subcategoria	categoria_original	subcategoria_original	id_usuario_p	usuario	nombre	cargo	ranking
165011130		SIN CONTACTO	EQUIVOCADO	EQUIVOCADO	EQUIVOCADO	1071	GEORGINA BXXXX	GEORGINA AMELIA BXXXX VXXXXXXXX	EJECUTIVO	1
165010812		SIN CONTACTO	SIN RESPUESTA	SIN CONTACTO	NO CONTESTA	1619	EDUARVER BXXXX	EDUARVER RAFAEL BXXXX CXXXXXX	EJECUTIVO	2
162691338	SALIENTE	SIN CONTACTO	BUZÓN DE VOZ	SIN CONTACTO	BUZÓN DE VOZ	1088	CESAR TXXXX	CESAR TXXXX	EJECUTIVO	4
162691506	SALIENTE	COMPROMISO DE PAGO		CONTACTO DIRECTO	SEGUIMIENTO PROMESA	1549	EMPERATRIZ BXXXXX	EMPERATRIZ BXXXXX	EJECUTIVO	4
162691699	SALIENTE	SIN CONTACTO	BUZÓN DE VOZ	SIN CONTACTO	BUZON DE VOZ	1088	CESAR TXXXX	CESAR TXXXX	EJECUTIVO	3
168562394	SALIENTE	SIN CONTACTO	SIN RESPUESTA	SIN CONTACTO	NO CONTESTA	1162	CECILIA RXXXXXXXX	CECILIA RXXXXXXXX	EJECUTIVO	1
168562119	SALIENTE	SIN CONTACTO	SIN RESPUESTA	SIN CONTACTO	NO CONTESTA	1071	GEORGINA BXXXX	GEORGINA AMELIA BXXXX VXXXXXXXX	EJECUTIVO	6
164599720	SALIENTE	SIN CONTACTO	ERROR DE RED/FUERA DE SERVICIO	SIN CONTACTO	FUERA DE SERVICIO	1567	SOFIA SXXXXXXXX	SOFIA SXXXXXXXX	EJECUTIVO	7
164599191	SALIENTE	SIN CONTACTO	SIN RESPUESTA	SIN CONTACTO	NO CONTESTA	1071	GEORGINA BXXXX	GEORGINA AMELIA BXXXX VXXXXXXXX	EJECUTIVO	1
167053401	SALIENTE	SIN CONTACTO	SIN RESPUESTA	SIN CONTACTO	NO CONTESTA	1619	EDUARVER BXXXX	EDUARVER RAFAEL BXXXX CXXXXXX	EJECUTIVO	1

Tabla 5: Datos de llamadas desde el data mart (Continuación)

4.3. Dashboard para toma de decisiones

Diariamente, se debe subir al *software* de cobranza los contactos a discar para que los ejecutivos comiencen con la gestión.

En paralelo, los encargados de las masividades se encargan de obtener los correos y teléfonos para el envío de SMS, bots y correos masivos.

Por ello, es necesario tener una visión general de cómo va el recupero y las gestiones a los distintos mandantes. Usualmente, existe un tablero para cada mandante, el cual se utiliza para esta toma de decisiones como también para mostrarle al cliente cómo va el rendimiento de NetPag.

En este *dashboard* se presentará la situación de NetPag general, tomando en consideración puntos como:

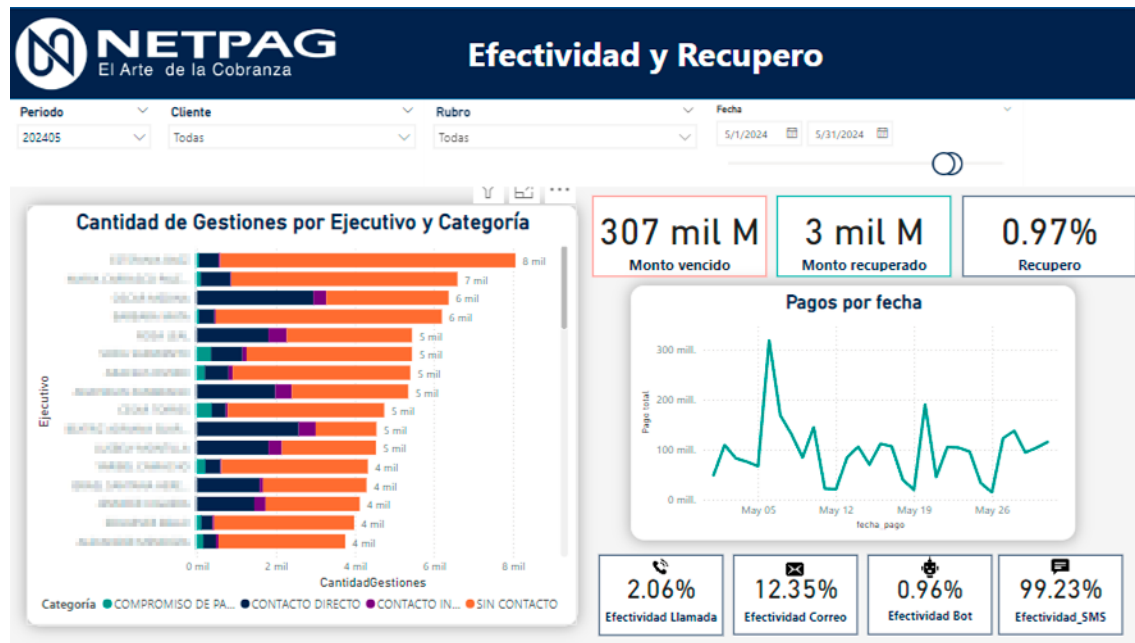


Figura 33: Dashboard: Efectividad y Recupero.
Fuente: Elaboración propia

- Rendimiento de los ejecutivos: Cantidad de Gestiones por Ejecutivo y Categoría
- Efectividad de los canales: Llamadas, Bot, SMS y Correo
- Porcentaje de recupero

Estos puntos permitirán que NetPag pueda ir cambiando sus estrategias y tomando decisiones para mejorar el recupero y efectividad de sus contactos.

El dashboard final se presenta en la figura 33.

Se decidió utilizar dos gráficos, uno de barras apiladas para las gestiones por ejecutivo, y otro gráfico de líneas para visualizar la evolución de los pagos durante un periodo de tiempo determinado por el filtro.

Las tarjetas “Monto vencido”, “Monto recuperado”, y “Recupero” junto con el gráfico de “Pagos por día” permiten revisar rápidamente los montos vencidos y recuperados junto con el porcentaje de recupero, es decir, el monto recuperado a partir del monto vencido.

El gráfico de “Cantidad de Gestiones por Ejecutivo y Categoría” y las tarjetas de efectividad por canal muestran el resultado de la estrategia de cobranza tomada por NetPag. Con estos datos, se pueden tomar decisiones como: Guiar a ciertos ejecutivos para que mejoren su efectividad en los contactos y enfocarse en la gestión mediante canales específicos que mejoren la efectividad.

La efectividad en este tablero se definió como la cantidad de gestiones con compromiso de pago o contacto directo dividido por las gestiones totales del canal. Es importante señalar la categorización de estas categorías para los canales de SMS y Correo, ya que para mensajes de texto se considera el mensaje entregado como contacto directo y mensaje con error como sin contacto, mientras que para los correos se considera como contacto directo el correo abierto, contacto indirecto el correo sin apertura y sin contacto los correos rebotados.

Los datos mostrados se ven afectados por los filtros que se encuentran en la parte superior del tablero, pudiendo filtrar por período, cliente (mandante), rubro y fecha.

Estos filtros permiten mostrar la situación general de NetPag o enfocarse en algún mandante o rubro en particular. De esta forma, el *dashboard* puede ser utilizado por distintas personas de Operaciones que quieran revisar la efectividad de sus estrategias según su cartera asignada para tomar decisiones, cumpliendo con el objetivo específico cuatro: “Desplegar información del *data mart* mediante un *dashboard* que apoye la toma de decisiones.”

Una segunda vista del *dashboard* “Efectividad y Recupero” se puede revisar en la figura 34 la cual tiene el mismo objetivo pero presentando nuevos gráficos.

El gráfico de “Recupero por Ejecutivo” representa los pagos totales (recupero) que se han asignado a cada ejecutivo, pudiendo notar rápidamente cuáles son los ejecutivos que han conseguido un mayor recupero de las deudas a partir de compromisos de pagos de deudores contactados.

Por otra parte, complementando la efectividad de la vista anterior, se puede visualizar un gráfico de “Efectividad por Canal” que muestra la cantidad de gestiones que se han realizado por canal desglosado por el resultado del contacto (Compromiso de Pago, Contacto Directo, Contacto Indirecto y Sin Contacto). Este gráfico permite visualizar con mayor detalle la efectividad de cada canal, identificando aquellos más utilizados como también los que han conseguido mejor resultado en las gestiones.

Luego, se presenta una tabla que complementa las tarjetas anteriores de “Monto vencido”, “Monto recuperado”, y “Recupero” pero esta vez desglosando los montos y recupero por tramo de mora (días desde el vencimiento de la deuda), lo cual permite visualizar en qué tramo de mora se concentran los pagos.

Además, se rediseñó un *dashboard* previamente desarrollado en NetPag llamado “Contactabilidad” presentado en la figura 35, utilizando el *data mart* como fuente de datos. Este *dashboard* se concentra en las gestiones de llamada y de bot mostrando gráficos comparativos entre contactos directos y compromisos de pago realizados a través de ambos canales aplicando filtros por cliente, periodo y fecha. Los datos se agrupan y analizan según el horario en el que se realizaron los contactos, lo que permite evaluar la efectividad de cada canal en diferentes momentos del día.

Con la implementación del *data mart*, se presenta la oportunidad de crear varios gráficos que

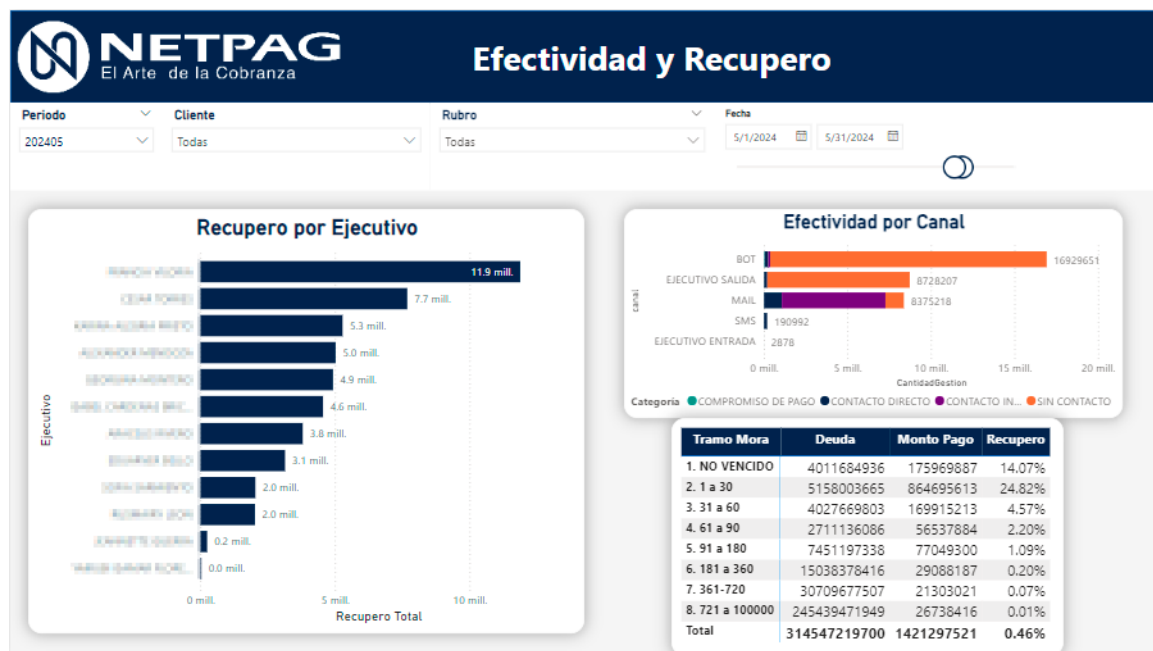


Figura 34: Dashboard: Vista dos de Efectividad y Recupero
Fuente: Elaboración propia

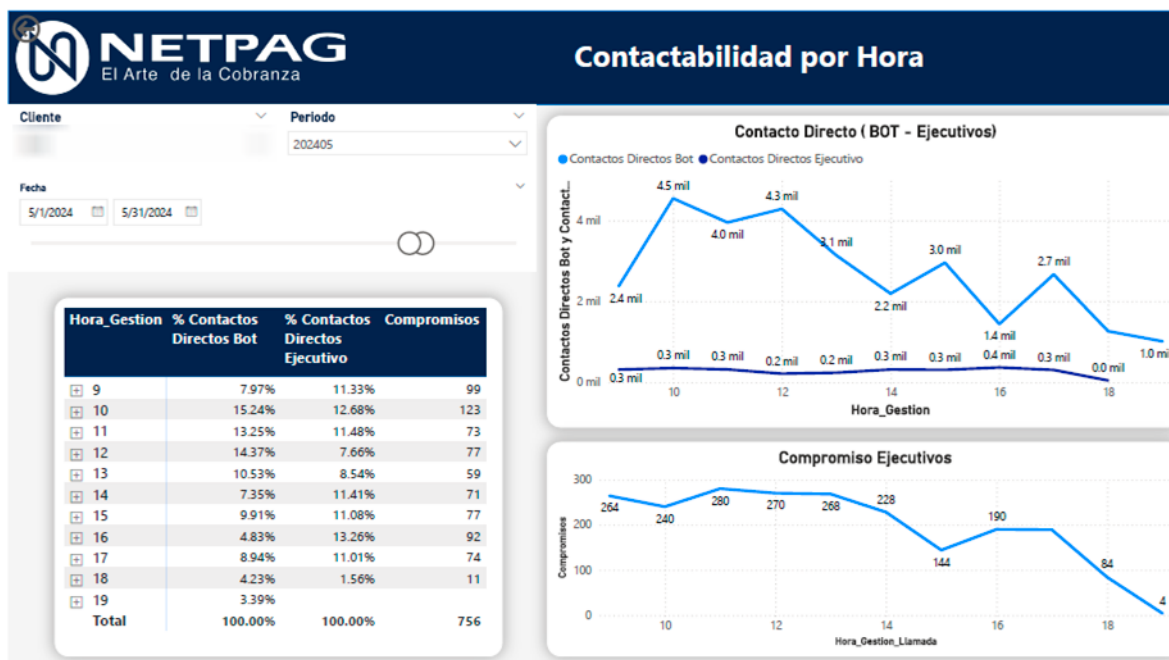


Figura 35: Dashboard: Contactabilidad
Fuente: Elaboración propia

permitan al área de Inteligencia de Negocios conocer a sus deudores, comportamiento de pago y generar propuestas de estrategias de cobranza al área de Operaciones. Por ejemplo, la tabla de Cliente_Deudor contiene información relevante de los deudores donde uno de los gráficos posibles es un mapa coroplético que muestra las regiones en las que vive cada deudor.

En la entrega final del proyecto, el jefe del área *TI* firmó un acta para validar la correcta implementación y funcionamiento del *data mart* en la empresa, la cual se puede visualizar en el anexo F, figura 36.

Finalmente, se presentan algunos casos donde el *data mart* ha permitido mejorar el proceso de toma de decisiones y análisis de datos, cumpliendo con el objetivo general “Diseñar e implementar un *data mart* que permita el acceso y procesamiento en línea de los datos con la finalidad de mejorar el proceso de toma de decisiones y análisis de datos.”

1. Respuesta rápida a solicitudes del área Comercial

En ocasiones, el área Comercial consulta al área de Inteligencia de Negocios por cruces de datos de deudores enviados por un posible nuevo cliente de NetPag y los datos existentes actualmente en la base de NetPag para revisar si se tienen datos de contacto con los cuales realizar gestiones. En esos casos, es necesario revisar la base de cada cliente con sus tablas de Contactos, considerando los distintos formatos para luego hacer el cruce, tomando un tiempo considerable dependiendo de la cantidad de datos.

Con el *data mart*, esto se soluciona con las tablas de Cliente_Deudor, Telefono_Deudor y Correo_Deudor al contener los datos de todos los deudores históricos, teniendo que realizar solo una consulta para el cruce. Además, se vuelve más simple poder acceder a otra información de los deudores, como por ejemplo, consultar los pagos de estos deudores para revisar su comportamiento de pago. Esto permite realizar una evaluación más completa del nuevo cliente, mejorando el proceso de **toma de decisiones** al disminuir los tiempos de consulta y al tener suficiente información para tomar una decisión justificada acerca de si brindar el servicio de cobranza o no.

2. Facilidad en la creación de *dashboards*

Como se mencionó anteriormente, en NetPag existe un tablero por cada cliente y periodo con el que se evalúa la efectividad de las gestiones y el recupero, en donde por cada cliente y periodo es necesario crear un nuevo *dashboard* apuntando a las tablas de la base de cada cliente. Esto agrega un tiempo extra al desarrollo de los tableros, puesto que hay que revisar el formato de los datos y crear el *dashboard* cada mes.

Con el *data mart*, igualmente es necesario revisar el formato de los datos pero solo cuando ingrese el nuevo cliente, desarrollando las vistas necesarias para que el *ETL* extraiga los datos. Luego, al tener los datos consolidados, es posible crear tableros que filtren por cliente y periodo, tal como se presentó en el *dashboard* creado, ahorrando tiempo en la creación de nuevos tableros por periodo. Además, si se desea modificar

los *dashboards* o crear nuevos, con el *data mart* se puede acceder a otros datos interesantes, como datos demográficos de los deudores o pagos de todos los clientes. Todo lo anterior permite que el **proceso de análisis de datos** sea más simple pero completo en información.

3. Optimización en la creación de estrategias

Como se mencionó anteriormente, NetPag suele crear un tablero por cada cliente, sin embargo, también es necesario tener un panorama general sobre la efectividad de las gestiones totales en NetPag, considerando la totalidad de los clientes, y al igual que en los casos anteriores, el tener los datos separados por cliente y en distintos formatos agregan tiempo extra al desarrollo.

Con el *data mart*, esto se resuelve con consultas a la base de datos sin necesidad de desarrollos extras, pudiendo extraer distinta información clave para NetPag, ya sean gestiones, pagos y/o deudas, ahorrando tiempo en la extracción para poder tener una visión global del rendimiento de NetPag. Esto permite implementar o ajustar estrategias de contacto de manera más ágil, mejorando el proceso de análisis de datos y posibilitando decisiones más informadas y eficientes.

CAPÍTULO 5

CONCLUSIONES

NetPag es una empresa con trayectoria en servicios especializados de recuperación de activos de empresa a empresa, enfrentando desafíos para gestionar su área B2C debido al volumen de datos y la falta de una base consolidada con la cual poder acceder fácilmente a los datos para realizar análisis.

NetPag en su propuesta de valor plantea el “uso de inteligencia artificial y modelos predictivos a la gestión de cobranza para mejorar la probabilidad de pago y efectividad de gestión” [NetPag,], lo cual hoy en día no es una realidad de la empresa, siendo un problema el desalineamiento de su propuesta de valor con la situación actual.

Como solución se propuso el desarrollo e implementación de un *data mart* para el área B2C, la cual es el área con más volumen de datos por el contacto a personas, incluyendo el uso de masividades como correos electrónicos, SMS y bots.

Para la definición de los requerimientos del negocio, se realizaron entrevistas a distintos usuarios para conocer su perspectiva, sus objetivos y sus dificultades en la empresa. Considerando esta información, se diseñó el *data mart* de forma que cumpla con las necesidades de los usuarios.

Este *data mart* requirió de una arquitectura y diseño que permitiese a distintas áreas de la empresa acceder a los datos con poco tiempo de espera para el desarrollo de la estrategia de cobranza del día, definiendo las llamadas, correos y SMS que se enviarán y en qué horarios.

Para ello, se decidió trabajar con el sistema de base de datos utilizado en la empresa, *SQL Server* por la naturaleza relacional de los datos, la capacitación del personal en *SQL*, el soporte dado por el área de *TI* y por la existencia de un *data mart* del área B2B en la empresa.

Este *data mart* se caracteriza por su enfoque en las Gestiones, segmentándolas según el canal por el que se realizan debido a las diferencias en la información capturada por distintos canales, por ejemplo, una gestión de llamada tiene un usuario, teléfono y una duración asociada, mientras que una gestión por correo electrónico masivo no tiene ni teléfono ni duración asociada, pero sí un correo electrónico. La separación de estas tablas no solo simplifica el acceso a los datos según el canal deseado, sino que también optimiza el rendimiento al gestionar volúmenes variables de información donde, por ejemplo, el volumen de gestiones por correo electrónico es mucho mayor comparado al volumen de las llamadas.

Entre las principales tablas de hecho del *data mart* se encuentran Gestión_Llamada, Gestión_Bot, Gestión_SMS, Gestión_Correo, Cliente_Por_Mandante y Pago.

Se tienen tablas de dimensiones como Cliente_Deudor, Región, Provincia, Ciudad, Comuna, Teléfono, Correo, Teléfono_Deudor, Correo_Deudor, Dotación, Deuda, Categoría (de Ges-

tión) y Subcategoría (de Gestión)

Este diseño no solo mejora la accesibilidad y eficiencia en el manejo de grandes volúmenes de datos, sino que también proporciona una estructura flexible para adaptarse a las necesidades cambiantes del análisis de datos en la organización.

Considerando el volumen de datos del área B2C, se decidió cambiar la tecnología en la cual se realizan los ETL generalmente en la empresa, escogiendo la interfaz de *Apache Spark* en Python, *Pyspark* la cual se caracteriza por realizar operaciones masivas de datos de forma eficiente aprovechando el procesamiento distribuido.

Uno de los mayores desafíos en la implementación del *data mart* fueron los ETL, puesto que requirieron de procesos complejos y extensos al tener en cuenta los distintos orígenes de datos y su variedad de formatos. Entre estos orígenes se incluye extracción desde APIs y bases de datos de la empresa junto a documentos de texto externos que complementan los datos existentes, destacando los bloques de numeración de *Subtel* para la identificación de números telefónicos definidos como también nombres registrados en Chile para la identificación del sexo del deudor.

Por otra parte, el *data mart* facilita el cumplimiento de la ley sobre protección de los derechos de los consumidores [MINISTERIO DE ECONOMÍA, 2021], con la cual NetPag ha tenido problemas al contactar excesivamente a algunos deudores. Al consolidar las gestiones y los datos de contacto en una base de datos, se puede verificar mediante consultas simples cuántas veces se ha contactado a una persona durante la semana, permitiendo llevar un control de la cantidad de contactos para cumplir con la ley y disminuir los reclamos de clientes.

Con respecto a la Ley 19.628 sobre Protección de la Vida Privada [PRESIDENCIA, 1999], el *data mart* solo recolecta los datos que han sido autorizados para el tratamiento por parte del deudor. Esta autorización es otorgada por los mandantes, y en casos específicos, como la gestión de reclamos, NetPag solicita directamente el consentimiento al deudor.

Además, el acceso a los datos del *data mart* está restringido a usuarios previamente autorizados, quienes cuentan con permisos específicos según su rol dentro de la organización. Solo un número muy limitado de usuarios tiene permisos para modificar los datos, con el objetivo de garantizar tanto la confidencialidad como la integridad de la información.

Por otro lado, la base de datos cuenta con respaldos diarios gestionados por el área de TI, lo que permite la recuperación rápida de la información en caso de pérdida de datos o fallos en el sistema. Esto permite una alta disponibilidad de los datos, facilitando su uso para los fines autorizados y permitiendo atender de manera eficiente las solicitudes de acceso o corrección por parte de los titulares, en cumplimiento con la ley.

Finalmente, al validar la solución se realizaron distintos experimentos al consultar al *data mart* y a la fuente de datos para comparar los tiempos de respuesta, facilidad de la consulta y uso de recursos.

- **Rendimiento de las consultas:** Las consultas ejecutadas en el *data mart* presentaron mejores tiempos de respuesta que al consultar con el método antiguo, incluso al aumentar la cantidad de meses que abarca la consulta, destacando la rapidez del *data mart* debido al uso de particiones por mes.
- **Simplicidad y mantenimiento de las consultas:** Las consultas del *data mart* son más breves y menos complejas que al consultar al sistema antiguo, puesto que este último requiere varias uniones al consultar distintas bases de datos de origen. Además, hay consultas que en el sistema antiguo requieren estandarización y transformación de columnas, mientras que en el *data mart* esto ya se encuentra integrado, sin requerir procesamiento adicional.
- **Uso de recursos:** Las métricas de rendimiento que se pueden observar en la tabla 3 indican que el *data mart* utiliza menos recursos de CPU y realiza menos lecturas físicas y lógicas en comparación con el sistema antiguo. Esto libera recursos del servidor, permitiendo que más usuarios realicen consultas simultáneamente sin degradar el rendimiento.

Por otra parte, la integración con herramientas de visualización como *Microsoft Power BI* facilita la creación de *dashboards* interactivos al tener consolidados los datos de distintos mandantes y periodos logrando consultas más simples, como también al aprovechar la estructura del *data mart* para obtener distintas columnas de tablas relacionadas, como por ejemplo, complementar la información de las gestiones al unir con los deudores para obtener la región en la que vive. Esto es un apoyo para la toma de decisiones de distintas áreas de la empresa al facilitar su exploración de los datos mediante los tableros en *Power BI*.

Uno de los desafíos para los analistas de negocios al crear estos tableros de *Power BI* es la generación de consultas que accedan a las tablas de forma eficiente, por ejemplo, aprovechando las particiones por periodo. Es importante en la creación de estos tableros preocuparse de las consultas que se ejecuten y la cantidad de usuarios que accedan para no sobrecargar la base de datos.

5.1. Trabajos Futuros

Uno de los principales desafíos a corto plazo para trabajar con el *data mart* orientado al área con personas trata sobre la protección de los datos personales, puesto que se almacenan datos como el RUT, nombre, dirección, sexo y datos de contacto de personas. Es importante que la recolección y uso de estos datos continúe cumpliendo con las regulaciones en Chile.

Considerando la información que se almacena en el *data mart* y la facilidad de su acceso a datos consolidados de deudores, es esencial que NetPag adopte prácticas éticas en el manejo de los datos personales en complemento con las medidas ya tomadas para la protección de datos, enfocándose en el uso de los datos con un fin específico de gestión de cobranza y

recuperación de deudas como también en la confidencialidad de los datos, garantizando que quienes tengan acceso a los datos personales mantengan la confidencialidad de la información y no la divulguen indebidamente. Además, como trabajo futuro, es necesario revisar y adaptar los procesos de tratamiento de datos en NetPag para asegurar el cumplimiento de la nueva Ley de Protección de Datos Personales, que entrará en vigencia en los próximos años. Esta normativa exigirá un mayor grado de protección en el manejo de los datos personales en comparación con la actual Ley 19.628 sobre Protección de la Vida Privada ya mencionada. [Chile, 2024]

En cuanto al área de Inteligencia de Negocios y desarrollo de tableros con *Power BI*, el trabajo desarrollado en esta memoria sienta las bases para la creación de una gran variedad de tableros con distintos objetivos que ayuden a conocer y mejorar la estrategia de cobranza de NetPag, como por ejemplo:

- Efectividad de los canales, medido por las gestiones exitosas en el contacto versus las gestiones total o mediante un gráfico de gestiones por canal y resultado de la gestión.
- Recupero durante un periodo, para revisar qué días acumulan más pagos y considerarlo en el ajuste de la estrategia.
- Gestiones por ejecutivo y categoría, para analizar el rendimiento de los ejecutivos según sus resultados de gestión.
- Comportamiento y características de los deudores, mostrando un resumen general de cuál canal de comunicación tiene más efectividad en los pagos según las reglas de negocio, horarios en los que hay más contacto e incluso revisando la distribución geográfica de los deudores.
- Análisis de costo-beneficio de las gestiones, mostrando gráficos de costo total versus recupero conseguido por cada canal o costo versus contactabilidad por canal.

El *data mart* también permite responder rápidamente a otras consultas que van naciendo de parte del área Comercial hacia el área de Inteligencia de Negocios sin necesidad de desarrollo extra, como la evaluación del ingreso de nuevos mandantes comparando sus bases de deudores con la base histórica de deudores que tiene NetPag para poder estimar el recupero al que se podría llegar.

Además, se abre la posibilidad de proyectos de creación de modelos de *machine learning* al complementar los datos de los deudores, deudas, gestiones y pagos, tales como:

- Análisis predictivo de los pagos: Uso de técnicas de análisis predictivo para identificar tendencias y proyectar comportamientos futuros de los deudores con el objetivo de armar una estrategia que contacte a los deudores más propensos a pagar.

- Segmentación avanzada de deudores: Uso de *clustering* para agrupar deudores con características y comportamientos similares, facilitando la implementación de estrategias de cobranza específicas para cada segmento.
- Recomendación de estrategias de cobranza: Sugerencias de la mejor estrategia de cobranza para cada deudor, considerando la efectividad histórica de diferentes métodos y características de los deudores.
- Optimización de recursos: Sugerencias sobre cuántas gestiones por distintos canales realizar dependiendo del costo y de la efectividad del contacto.

El uso de estos nuevos tableros y modelos de inteligencia artificial para la mejora de la estrategia de cobranza tiene un impacto directo en el recupero, tiempos de cobranza y resultado de las gestiones, logrando alinear la realidad de NetPag con su propuesta de valor sobre el uso de tecnología e innovación para predecir el comportamiento de las carteras. Al mejorar el rendimiento de NetPag, mejora su posición en los *benchmark* de cada mandante, posicionándose como un fuerte candidato en la cobranza de carteras.

Considerando la validación de los datos discutida en el capítulo anterior, un trabajo futuro es la aplicación de una solución más robusta para la inconsistencia en los RUTs, pudiendo aplicar funciones de similitud de nombres para encontrar a la persona si es que ya fue ingresada con otro RUT, como también validación de RUTs mediante servicios externos como APIs que entreguen información sobre un RUT y/o nombre consultado, siempre y cuando esto no afecte la privacidad de los deudores.

Por otra parte, aún es posible mejorar el rendimiento de los ETL gracias a la arquitectura distribuida de *Apache Spark*, aprovechando máquinas locales de NetPag para ser utilizadas como nodos.

Queda espacio aún para crear distintos reportes a partir de las tablas del *data mart*, creando nuevos procesos aprovechando el rendimiento de *Pyspark*.

Con la existencia del *data mart* de B2C y B2B se abren nuevos proyectos como la creación de *data mart* para el área Judicial o directamente a la creación de un *data warehouse* a partir de los *data marts* existentes con el método *bottom-up*. Esto permitirá acceder más fácil a los datos consolidados y tener una visión aun más completa del negocio.

Otro trabajo futuro a realizar es la creación de cubos OLAP aprovechando la licencia con Microsoft de la empresa utilizando *SQL Server Analysis Services (SSAS)*. La capa OLAP dentro de una arquitectura de un *data warehouse* se puede visualizar en la figura 6. Las herramientas OLAP permiten la exploración y manipulación interactiva del *data mart* con el fin de encontrar patrones o tendencias de importancia para la empresa, existiendo distintas operaciones que se pueden realizar haciendo uso de SSAS tales como roll-up, drill-down o pivote que se presentan en el marco conceptual.

ANEXOS

A. Índexación

Un índice SQL es una tabla de búsqueda rápida que tiene como objetivo encontrar los registros que los usuarios necesitan buscar con mayor frecuencia.

SQL Server reconoce varios tipos de índices, pero uno de los más comunes es el índice agrupado. Esta clase de índice se crea automáticamente con una clave principal.

Un índice único es usado para mantener la integridad de la información de las columnas sobre las cuales es creado al asegurar que no hay valores duplicados en el índice clave, y la filas de la tabla, sobre la cual el índice es creado. [Yaseen, 2019]

Se puede definir como único el índice agrupado y no agrupado, estos se diferencian en que el agrupado reorganiza físicamente los datos de la tabla según el orden de la clave de índice, mientras que el no agrupado crea una estructura de índice separada de los datos de la tabla donde cada entrada en el índice no agrupado contiene la clave del índice y un puntero a las filas de datos correspondientes.

B. Particionamiento horizontal de tablas

El particionamiento horizontal divide una tabla en múltiples tablas que contienen el mismo número de columnas, pero menos filas.

Las tablas son particionadas horizontalmente basadas en una columna que será usada para particionar y los rangos asociados a cada partición. La columna de particionamiento es usualmente una columna de fecha pero todos los tipos de datos que son válidos para usarse como columnas de índice pueden ser usados como columna de partición. [Medic, 2020]

Hay dos enfoques diferentes que se pueden usar para lograr la partición de la tabla. El primero es crear una nueva tabla particionada y simplemente copiar los datos desde su tabla existente en la nueva tabla y renombrarla. El segundo enfoque es particionar una tabla existente reconstruyendo o creando un índice agrupado en la tabla.

C. Vistas materializadas

Una vista materializada es una tabla de datos duplicada que se crea mediante la combinación de datos de varias tablas existentes para una recuperación de datos más rápida.

En las bases de datos relacionales, una vista es una tabla temporal que se crea cuando se transforman y combinan los datos en varias tablas base. Es una tabla virtual que no almacena ningún dato por sí misma. En cambio, se define mediante una consulta en una o más tablas de origen.

Una vista materializada, por otro lado, almacena los resultados de una consulta específica como una tabla física en la base de datos. Los datos de la vista materializada se calculan con anterioridad y se almacenan, lo que significa que los resultados ya están disponibles sin necesidad de volver a calcular la consulta cada vez que se accede a la vista. [Services,]

D. Función aplicada para la elección de nombres en Cliente_Deudor

```
def eleccion_nombre(nombres_dic):
    #Variables para inicializar el nombre y mandante al que pertenece el
    nombre.
    max_len = 0
    max_nombre = ''
    max_cliente = ''

    #Variables para separar el nombre en apellidos y nombre
    apellido_mat = None
    apellido_pat = None
    nombres = None

    #Si tiene alguna de estas palabras, no se considera un nombre valido
    omitir = ['NORMALIZADO', 'ELIMINADO', 'SIN INFORMACION']

    #Variable para identificar si el nombre ya viene separado en apellido
    y nombre
    separados=False

    #Variable para identificar si el nombre se ha omitido
    omitido=False

    for dic in nombres_dic:

        if (dic['apellido_mat'] is not None and (dic['apellido_pat'] is not
        None) and dic['nombres'] is not None):
            """Se escoge el primer diccionario que tenga completitud en
            apellido mat, pat y nombres. Se limpian los nombres quitando
            caracteres especiales"""
            apellido_pat=limpieza_nombre(dic['apellido_pat'])
            apellido_mat=limpieza_nombre(dic['apellido_mat'])
            nombres=limpieza_nombre(dic['nombres'])
```

```
separados=True
break

if (dic['nombre'] is not None):
    #Si tiene nombre, se verifica que no contenga las palabras
    omitidas, se limpia y se escoge el nombre más largo
    for palabra in omitir:
        if palabra in dic['nombre']:
            omitido=True

    if omitido:
        continue

    #Se limpia el nombre quitando caracteres especiales como puntos
    y coma
    nombre_dic = limpieza_nombre(dic['nombre'])
    len_n = len(nombre_dic)

    #Si el largo es mayor al actual, se escoge como nombre
    if (len_n > max_len):
        max_len = len_n
        max_nombre = nombre_dic
        max_cliente = dic['cliente']

#Si los nombres ya vienen separados por mandante, solo se agrega el
campo de nombre_completo
if separados:
    nombre_completo = nombres + ' ' + apellido_pat + ' ' + apellido_mat

    return {'apellido_mat': apellido_mat, 'apellido_pat': apellido_pat,
            'nombres': nombres, 'nombre_completo': nombre_completo}

else:
    #Revisar caso de nombres invertidos. Mandantes que entregan el
    nombre en formato apellido nombre.
    if max_cliente in ['CLIENTE3', 'CLIENTE5', 'CLIENTE6', 'CLIENTE7']:
        nombres, apellido_pat, apellido_mat =
            SplitNombresInverso(max_nombre)
        nombre_completo = max_nombre
        return {'apellido_mat': apellido_mat, 'apellido_pat':
            apellido_pat, 'nombres': nombres, 'nombre_completo':
            nombre_completo}

#Separar en apellido y nombres
nombres, apellido_pat, apellido_mat = SplitNombres(max_nombre)
nombre_completo = max_nombre
```

```
return {'apellido_mat':apellido_mat, 'apellido_pat': apellido_pat,  
        'nombres': nombres, 'nombre': nombre_completo}
```

E. Consulta SQL Completa

La consulta SQL utilizando el sistema antiguo para obtener “Recupero versus deuda en los últimos tres meses por mandantes del rubro de autopistas” se realiza en varias etapas, calculando la deuda y los pagos acumulados por mandante y periodo.

En primer lugar, se obtiene el total de la deuda y pagos para el cliente CLIENTE2.

```
WITH DeudaAcumulada_CLIENTE2 AS (  
    SELECT  
        'CLIENTE2' AS MANDANTE,  
        MES AS PERIODO,  
        SUM(CAST(DEUDA AS BIGINT)) AS DeudaTotal  
    FROM  
        (SELECT  
            MAX(CAST(CD.DEUDA AS int)) AS DEUDA, CD.MES  
        FROM PROVEEDOR_CLIENTE2.dbo.CONTACTOS_DEUDA CD  
        GROUP BY CASE WHEN LEN(CD.RUT) > 2  
            THEN LEFT(CD.RUT, LEN(CD.RUT)-2) ELSE null END,  
            CD.FOLIO, CD.MES) DEUDA_CLIENTE2  
        WHERE MES >= 202402 AND MES <= 202404  
        GROUP BY MES  
    ),  
    PagoAcumulado_CLIENTE2 AS (  
        SELECT  
            'CLIENTE2' AS MANDANTE,  
            MES AS PERIODO,  
            SUM(CAST(MONTOCOMPEN AS BIGINT)) AS PagoTotal  
        FROM  
            PROVEEDOR_CLIENTE2.dbo.PAGOS  
        WHERE MES >= 202402 AND MES <= 202404  
        AND TRY_CAST(MONTOCOMPEN AS INT) > 10  
        GROUP BY MES  
    ),
```

Luego, se realiza lo mismo para CLIENTE3.

```
DeudaAcumulada_CLIENTE3 AS (  
    SELECT  
        'CLIENTE3' AS MANDANTE,  
        MES AS PERIODO,
```

```
        SUM(CAST(TOTAL AS BIGINT)) AS DeudaTotal
FROM
    PROVEEDOR_CLIENTE3.dbo.CONTACTOS_DEUDA
WHERE MES >= 202402 AND MES <= 202404
GROUP BY MES),
PagoAcumulado_CLIENTE3 AS (
    SELECT
        'CLIENTE3' AS MANDANTE,
        MES AS PERIODO,
        SUM(CAST(MONTO AS BIGINT)) AS PagoTotal
    FROM
        (SELECT MES, CAST(MONTO AS INT) AS MONTO
        FROM PROVEEDOR_CLIENTE3.dbo.PAGOS
        WHERE MES >= 202402 AND MES <= 202404
        AND TIPODOC != 'NC'
        AND TRY_CAST(MONTO AS INT) > 10
        GROUP BY RUT, FOLIO, CAST(MONTO AS int),
        FECHAPAGO, MES, fechacarga) PAGO_CLIENTE3
    GROUP BY MES
),
```

Y también para el cliente CLIENTE1. Este varía un poco puesto que los pagos se extraen de las tablas REPORTE.

```
DeudaAcumulada_CLIENTE1 AS (
    SELECT
        'CLIENTE1' AS MANDANTE,
        MES AS PERIODO,
        SUM(DEUDA) AS DeudaTotal
    FROM
        (SELECT SUM(CAST(MONTO AS BIGINT)) AS DEUDA,
        202404 AS MES
        FROM PROVEEDOR_CLIENTE1.dbo.DEUDA_CONSOLIDADA_202404
        UNION ALL
        SELECT SUM(CAST(MONTO AS BIGINT)) AS DEUDA,
        202403 AS MES
        FROM PROVEEDOR_CLIENTE1.dbo.DEUDA_CONSOLIDADA_202403
        UNION ALL
        SELECT SUM(CAST(MONTO AS BIGINT)) AS DEUDA,
        202402 AS MES
        FROM PROVEEDOR_CLIENTE1.dbo.DEUDA_CONSOLIDADA_202402) AS DEUDA
    WHERE MES >= 202402 AND MES <= 202404
    GROUP BY MES
),
PagoAcumulado_CLIENTE1 AS (
    SELECT
```

```
'CLIENTE1' AS MANDANTE,
PERIODO AS MES,
SUM(CAST(PAGO AS BIGINT)) AS PagoTotal
FROM
(SELECT TRY_CAST(MONTO_COM AS INT) AS PAGO,
202404 AS PERIODO
FROM PROVEEDOR_CLIENTE1.dbo.REPORTE_202404
UNION ALL
SELECT TRY_CAST(MONTO_COM AS INT) AS PAGO,
202403 AS PERIODO
FROM PROVEEDOR_CLIENTE1.dbo.REPORTE_202403
UNION ALL
SELECT TRY_CAST(MONTO_COM AS INT) AS PAGO,
202402 AS PERIODO
FROM PROVEEDOR_CLIENTE1.dbo.REPORTE_202402) AS PAGOS
WHERE PERIODO >= 202402 AND PERIODO <= 202404
AND TRY_CAST(PAGO AS INT) > 10
GROUP BY PERIODO)
```

Para finalmente, seleccionar los resultados en conjunto.

```
SELECT
d.MANDANTE AS Mandante,
d.PERIODO AS Periodo,
ISNULL(d.DeudaTotal, 0) AS TotalDeuda,
ISNULL(p.PagoTotal, 0) AS TotalPago,
ROUND((CAST(p.PagoTotal AS FLOAT) / CAST(d.DeudaTotal AS FLOAT)) *
100,2)
AS PorcentajeRecupero
FROM
DeudaAcumulada_CLIENTE2 d
LEFT JOIN
PagoAcumulado_CLIENTE2 p ON d.PERIODO = p.PERIODO
UNION ALL
SELECT
d.MANDANTE AS Mandante,
d.PERIODO AS Periodo,
ISNULL(d.DeudaTotal, 0) AS TotalDeuda,
ISNULL(p.PagoTotal, 0) AS TotalPago,
ROUND((CAST(p.PagoTotal AS FLOAT) / CAST(d.DeudaTotal AS FLOAT)) *
100,2)
AS PorcentajeRecupero
FROM
DeudaAcumulada_CLIENTE3 d
LEFT JOIN
PagoAcumulado_CLIENTE3 p ON d.PERIODO = p.PERIODO
```

```
UNION ALL
SELECT
    d.MANDANTE AS Mandante,
    d.PERIODO AS Periodo,
    ISNULL(d.DeudaTotal, 0) AS TotalDeuda,
    ISNULL(p.PagoTotal, 0) AS TotalPago,
    ROUND((CAST(p.PagoTotal AS FLOAT) / CAST(d.DeudaTotal AS FLOAT)) *
        100,2)
    AS PorcentajeRecupero
FROM
    DeudaAcumulada_CLIENTE1 d
LEFT JOIN
    PagoAcumulado_CLIENTE1 p ON d.PERIODO = p.MES
```

F. Acta de Validación



Acta de Validación de Data Mart B2C

Nombre del Proyecto: "Diseño e implementación de un Data Mart para el área B2C de la empresa NetPag"

Mediante esta acta se valida el diseño, implementación y funcionamiento del data mart para el área B2C de NetPag, con los siguientes alcances:

- ✓ El data mart se encuentra en un servidor de NetPag y en una base de datos SQL Server, a la que solo los usuarios autorizados pueden acceder.
- ✓ El diseño implementado es coherente con los modelos de negocio de NetPag.
- ✓ Se presentan tableros en Power BI conectados al data mart, lo que valida su funcionamiento y utilidad para la toma de decisiones.
- ✓ Las pruebas de rendimiento realizadas han demostrado que el data mart responde de manera eficiente a las consultas, mostrando mejoras en los tiempos de respuesta en comparación con el sistema anterior de obtención de datos.

Con fecha 17 de octubre de 2024, el data mart ha sido validado satisfactoriamente y está listo para su uso en el área B2C de NetPag.


Miguel Verdugo
Jefe de Tecnología,
Implementación y
Continuidad Operacional

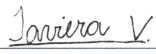

Javiera Villarroel
Memorista

Figura 36: Acta de Validación del Data Mart.
Fuente: Elaboración propia

REFERENCIAS BIBLIOGRÁFICAS

- [Airbyte, 2024] Airbyte (2024). A Beginners Guide to SQL Server Integration Services (SSIS).
- [Amazon Web Services,] Amazon Web Services, I. ¿Qué es Data Mart? <https://aws.amazon.com/es/what-is/data-mart/>. Recuperado el 9 de noviembre de 2022.
- [Chile, 2024] Chile, G. D. (2024). Se aprueba Ley de Protección de Datos Personales: Revisa de qué se trata.
- [Chile, 2021] Chile, O. (2021). ¿Qué es un data mart? <https://www.oracle.com/cl/autonomous-database/what-is-data-mart/>. Recuperado el 9 de noviembre de 2022.
- [Chugugrace, 2024] Chugugrace (2024). SQL Server Integration Services - SQL Server Integration Services (SSIS).
- [de Telecomunicaciones de Chile, 2024] de Telecomunicaciones de Chile, S. (2024). Servicios públicos.
- [Exchange,] Exchange, T. Call Center Agent Analytics | Tableau Exchange. <https://exchange.tableau.com/es-es/products/662>. Recuperado el 6 de octubre de 2024.
- [Guy, 2024] Guy, S. D. (2024). What Is SSIS and Should You Use It?
- [Jain, 2023] Jain, P. (2023). Introducing the Power of ETL with PySpark! - Priyam Jain - Medium.
- [Kimball y Ross, 2013] Kimball, R. y Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley, Indianapolis, IN, 3 edición.
- [Kimball et al., 2008] Kimball, R., Ross, M., Thorthwaite, W., Becker, B., y Mundy, J. (2008). *THE DATA WAREHOUSE LIFECYCLE TOOLKIT, 2ND ED.* John Wiley Sons.
- [LinkedIn, 2024] LinkedIn (2024). How can you handle a fact table with multiple granularities?
- [Malinowski y Zimányi, 2008] Malinowski, E. y Zimányi, E. (2008). *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Data-Centric Systems and Applications. Springer, Berlin.
- [Medic, 2020] Medic, M. (2020). Particionamiento de tablas de bases de datos en SQL Server.
- [MINISTERIO DE ECONOMÍA, 2021] MINISTERIO DE ECONOMÍA, F. Y. T. (2021). Ley 21320 (20-abr-2021) M. de Economía, Fomento y Turismo | Ley Chile. Biblioteca del Congreso Nacional de Chile. <https://www.bcn.cl/leychile/navegar?idNorma=1158548&idParte=10217460>.

- [MongoDB,] MongoDB. Databases vs. data warehouses vs. Data lakes. <https://www.mongodb.com/databases/data-lake-vs-data-warehouse-vs-database>. Recuperado el 11 de enero de 2023.
- [NetPag,] NetPag. Cobranza para Empresas - NETPAG. <https://www.netpag.cl/cobranza-para-empresas-servicios-de-cobranza/>. Recuperado el 3 de noviembre de 2022.
- [NetPag, 2022] NetPag (2022). Cobranza Administrativa. <https://www.netpag.cl/servicios-de-cobranza/cobranza-administrativa/>. Recuperado el 3 de noviembre de 2022.
- [PRESIDENCIA, 1999] PRESIDENCIA, M. S. G. D. L. (1999). Ley 19628 SOBRE PROTECCION DE LA VIDA PRIVADA.
- [Rivaquiroga,] Rivaquiroga. GitHub - rivaquiroga/guaguas: Un paquete de R con datos sobre nombres de bebés registrados en Chile entre 1920 y 2021.
- [SEPÚLVEDA TAPIA, 2021] SEPÚLVEDA TAPIA, M. A. (2021). IMPLEMENTACIÓN DE UNA SOLUCIÓN TECNOLÓGICA DE ARQUITECTURA DE DATA WAREHOUSE PARA NETPAG, QUE DIFERENCIE Y AGREGUE VALOR A LA FIRMA EN EL MERCADO DE LA COBRANZA ADMINISTRATIVA.
- [Services,] Services, A. W. ¿Qué es una vista materializada? Explicación de la vista materializada: AWS.
- [SOCIAL, 2019] SOCIAL, M. D. T. Y. P. (2019). Ley 21142 (01-mar-2019) M. del Trabajo y Previsión Social | Ley Chile. Biblioteca del Congreso Nacional de Chile. <https://www.bcn.cl/leychile/navegar?idNorma=1129381>.
- [Tableau,] Tableau. What is a dashboard? a complete overview. <https://www.tableau.com/learn/articles/dashboards/what-is>. Recuperado el 3 de octubre de 2024.
- [Training, 2023] Training, D. (2023). How to use Microsoft SQL Server in Talend Open Studio.
- [Vaisman y Zimányi, 2014] Vaisman, A. y Zimányi, E. (2014). *Data Warehouse Systems: Design and Implementation*. Springer, Heidelberg.
- [Wexler et al., 2017] Wexler, S., Shaffer, J., y Cotgreave, A. (2017). *The Big Book of Dashboards*. John Wiley Sons.
- [Yaseen, 2019] Yaseen, A. (2019). Uso de diferentes tipos de índices SQL Server.