Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Solución de Business Intelligence para empresa del sector transporte

Proyecto Fin de Grado

Grado en Ingeniería del Software

Junio 2024

Autor:
Borja Álvarez Bernardo

Tutor:
Pedro Pablo Alarcón Cavero

En primer lugar, quiero agradecer a mi tutor de TFG, Pedro Pablo Alarcón. Además de brindarme su apoyo en este proyecto personal, fue él quien me introdujo en este vasto campo de la Inteligencia de Negocio el cual me entusiasma.

Agradezco a mis padres por alentarme desde pequeño a hacer lo que me apasiona. A mis amigos que me han ayudado a revisar mi redacción aunque no tuvieran ni idea del tema. Y en especial, a Andrea, quien siempre me ha brindado su apoyo incondicional y me ha ayudado a seguir adelante durante mi carrera.

Resumen

La ingente cantidad de información con la que nos encontramos hoy en día nos abre una gran variedad de oportunidades, de las cuales muy pocas somos si quiera conscientes. El ser humano tiene una capacidad de comprensión limitada, y en consecuencia, ante grandes cantidades de datos desectructurados le es imposible obtener una información valiosa.

Conociendo nuestras debilidades, podemos reforzarlas. Como resultado, el análisis de datos ha adquirido una importancia cada vez mayor en la toma de decisiones tanto en grandes, como en medianas empresas. La posibilidad de obtener información útil de grandes cantidades de datos gracias a los denominados sistemas de Business Intelligence, es clave para el éxito de cualquier compañía.

El presente proyecto pretende abarcar directamente esta problemática, presentando una solución completa de Business Intelligence que permita el estudio de oportunidades anteriormente ocultas.

La solución incluye desde el análisis de datos de origen, hasta la generación de visualizaciones que permitan a los usuarios de negocio analizar los datos procesados. Los cuadros de mandos a desarrollar pretenden mostrar de forma holística el estado, rendimiento y posibles riesgos de una empresa.

El proyecto se desarrolla para una empresa del sector de transportes. Por temas de privacidad y políticas de protección de datos, se ha reemplazado el nombre de la empresa, así como los datos utilizados en los cuadros de mando, por un nombre ficticio "Pirnain".

Finalmente, se presentan líneas de ampliación para que el sistema propuesto abarque más áreas de la empresa.

Abstract

The enormous amount of information we are confronted with today opens up a wide variety of opportunities, of which we are hardly aware. Human beings have a limited capacity for comprehension, and consequently, when faced with large amounts of unstructured data, it is impossible for them to obtain valuable information.

By knowing our weaknesses, we can strengthen them. As a result, data analysis has become increasingly important in decision-making in both large and medium-sized companies. The possibility of obtaining useful information from large amounts of data thanks to so-called Business Intelligence systems is key to the success of any company.

This project aims to directly address this problem, presenting a complete Business Intelligence solution that allows the study of previously hidden opportunities.

The solution includes from the analysis of source data to the generation of visualisations that allow business users to analyse the processed data. The dashboards to be developed are intended to show holistically the status, performance and possible risks of a company.

The project is developed for a company in the transport sector. For privacy and data protection reasons, the company name and the data used in the dashboards have been replaced by a fictitious name 'Pirnain'.

Finally, lines of extension are presented in order for the proposed system to cover more areas of the company.

Índice

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

1.1. Contexto

En el actual contexto del sector del transporte, donde la demanda exige entregas rápidas y de calidad, las empresas se enfrentan al gran desafío de optimización de sus procesos de gestión y envío. Como resultado los tiempos de espera se ven reducidos, gracias a la eficiencia en la entrega de productos, y la satisfacción del cliente aumenta.

Ante esta situación se encuentra Pirnain. Pirnain es una empresa de transportes con múltiples franquicias en la península ibérica. La solución que se desarrolla durante el presente trabajo, se centra en una franquicia en concreto. Puesto que, a pesar de la existencia de una base de datos proporcionada por la empresa matriz, las franquicias no pueden acceder a sus propios datos de manera eficiente.

Aunque Pirnain no es una empresa pequeña, el acceso a los datos sobre clientes, envíos o servicios, entre otros datos de interés, es demasiado complejo y tedioso para las franquicias. Como resultado, los usuarios de negocio pierden mucho tiempo recopilando datos de distintas fuentes, aclarando el significado de campos ambiguos o campo homónimos que representan conceptos diferentes.

Actualmente, las franquicias no tienen un sistema que permita visualizar sus datos de manera útil o, si quiera, comprensible. Por consiguiente, los usuarios de negocio se ven obligados a ofertas a clientes en base a su intuición, sin tener una visión objetiva del estado de la empresa, ni de la viabilidad de la misma.

Tras cuantiosas pérdidas y problemas acarreados por esta desinformación, se ha tomado la decisión de crear una nueva estructura que permita el análisis de sus datos. De tal manera que, la información se encuentre centralizada y accesible para los usuarios que la requieran.

1.2. Solución propuesta

Se presenta una posible solución de Business Intelligence que permita visualizar todos los datos de interés de la franquicia que los usuarios de negocio necesiten. Para ello: se extraerán los datos, los cuales serán procesados (con el fin de homogeneizarlos y darles una coherencia como conjunto), para posteriormente almacenarlos en un nuevo sistema orientado al análisis. Finalmente, se propone generar una serie de visualizaciones sobre el nuevo conjunto de datos.

Con lo expuesto anteriormente, los datos siguen sin ser accesibles para los usuarios sin conocimiento sobre bases de datos. Por ende, se desarrollarán una serie de visualizaciones basadas en las necesidades de los usuarios de negocio.

Como resultado, obtendremos un conjunto de datos centralizados, accesibles y coherentes, los cuales pueden ser explotados mediante visualizaciones por parte de los usuarios de negocio.

1.3. Objetivos

A continuación, se exponen los objetivos que deben ser alcanzados, para que el desarrollo del proyecto sea considerado un éxito.

1.3.1. Objetivo principal

El principal objetivo del presente Trabajo de Fin de Grado es el diseño e implementación de una solución de Business Intelligence para una franquicia de la empresa Pirnain. Su principal motivación es poder presentar los datos, que actualmente se encuentra disgregados en la empresa, de forma clara, coherente y organizada.

1.3.2. Objetivos específicos

Para poder alcanzar el objetivo principal, primero se deben cumplir una serie de objetivos específicos. Una vez se hayan finalizado todos estos, se podrá dar como alcanzado el objetivo principal del proyecto. Los objetivos específicos definidos para este proyecto son los siguientes:

1. Análisis de la estructura y datos operacionales.
2. Describir el dominio del negocio.
3. Creación de un Data Warehouse para la franquicia.
4. Creación de procesos ETL.
5. Planificación del refresco de datos.
6. Creación de múltiples visualizaciones para los usuarios de negocio.

1.4. Contenido de la memoria

En esta sección se encuentra una breve introducción al contenido de cada apartado que compone la memoria.

1. Introducción

En este capítulo se presenta una descripción del contexto del proyecto, así como los objetivos del mismo. Para finalmente mostrar los diferentes apartados de esta memoria.

1.1. Contexto

En esta sección se presenta el contexto del problema, así como una breve introducción al mismo.

1.2. Solución propuesta

En esta sección se propone una solución para el problema expuesto anteriormente. Manteniendo una descripción a alto nivel de la solución que posteriormente se implementará.

1.3. Objetivos

En esta sección se exponen los objetivos del proyecto. Primero se presenta el objetivo principal que abarca todo el alcance, para posteriormente presentar los objetivos específicos que permitirán alcanzar este objetivo principal.

1.4. Contenido de la memoria

En esta sección se encuentra una breve introducción al contenido que se encuentra en cada apartado de la memoria.

2. Descripción del dominio

En este capítulo se presenta tanto el contexto de la empresa para la cual se desarrolla la proyecto, así como el dominio del mismo.

2.1. Contexto de negocio

En esta sección se realiza una breve descripción del contexto de negocio del proyecto. Para ello, se explican los conceptos principales del dominio y el funcionamiento de la empresa en términos generales.

2.2. Dominio del proyecto

En esta sección se expone el dominio del proyecto, tras la descripción del contexto de negocio. En este se expone la situación actual de la empresa, así como los problemas que tiene que afrontar, y serán tratados en el presente trabajo.

3. Tecnologías utilizadas

En este capítulo se listan las tecnologías usadas para cada fase del proyecto, así como una breve descripción de las mismas.

3.1. Knime

En esta sección se realiza una introducción a la herramienta seleccionada para implementar los procesos ETL. Donde se definirán sus posibles usos, contexto, evolución y como se ha implementado en la solución.

3.2. Excel

En esta sección se expone brevemente la herramienta Excel. Se resumirá: cómo y porque se utiliza en este proyecto, así como algunos conceptos más avanzados de utilidad.

3.3. Python

En esta sección se realiza una breve descripción del lenguaje de programación Python, haciendo especial hincapié en conceptos de web scraping utilizados en este proyecto.

3.4. MySQL

En se expone una breve descripción del sistema de gestión de bases de datos relacional seleccionado para implementar el proyecto. Se explicarán los conceptos más importantes, la importancia de su tipo relacional, y el porqué de su uso.

3.5. Power BI

En esta sección se realiza una breve introducción a la herramienta de visualización Power BI. Se explicarán sus principales usos, funcionalidades y ventajas respecto a otras herramientas.

4. Arquitectura del sistema de BI

En este capítulo se presenta la arquitectura diseñada para el proyecto, donde se especifican las fuentes de datos y servicios utilizados.

5. Diseño del Data Warehouse

En este capítulo se expone el diseño del Data Warehouse. Para ello se presenta el modelo conceptual de datos, los Data Marts por los que está compuesto, así como las tablas de cada uno de ellos.

5.1. Modelo conceptual de datos

En esta sección se presentan los diseños de los 3 Data Marts que componen el modelo conceptual de datos del Data Warehouse.

5.2. Modelo lógico de datos

En esta sección se desarrollan los 3 Data Marts que componen el modelo conceptual de datos, especificando los campos de cada tabla que los componen.

5.3. Relaciones entre los Data Marts del modelo de datos

En esta sección se exponen como se relacionan los diferentes Data Marts que componen el Data Warehouse.

5.4. Métricas

En esta sección se desarrollan todas las métricas identificadas para este modelo de datos.

5.5. Entidades

En esta sección se exponen todas las entidades que conforman el Data Warehouse.

5.6. Atributos

Finalmente, en esta sección se exponen todos los atributos de cada una de las tablas de modelo de datos del Data Warehouse.

6. Procesos ETL

En este capitulo se identifican los procesos de preparación de datos con lo que la solución propuesta cubrirá los requerimientos de negocio identificados.

6.1. Estructuración y aplicación de lógica de negocio

En esta sección se exponen el objetivo, condiciones de entrada, condiciones de salida y descripción que debe cumplir el sistema a desarrollar.

6.2. Planificación

Esta sección recoge la información relativa a la frecuencia de refresco planificada.

6.3. Obtención de los hechos

En esta sección se exponen los procesos encargados de poblar las tablas de hechos del Data Warehouse con los datos procesados. Además, se detallarán los principales nodos utilizados en la herramienta KNIME.

6.4. Obtención de las dimensiones

En esta sección se presentan los procesos necesarios para obtener las columnas de cada tabla dimensión a partir de las fuentes de datos.

6.5. Procesos de Staging

En esta sección se exponen los procesos desarrollados en KNIME cuyo fin es automatizar la ingesta de datos de las fuentes de entrada a la zona de staging del servidor.

6.6. Procesos generales

En esta sección se muestran los procesos complementarios que tienen como objetivo facilitar la ingesta de datos sobre dimensiones o tablas de hechos recurrentes.

7. Visualizaciones

En este capítulo se expondrán todas las visualizaciones implementadas para el proyecto.

7.1. Informe general Pirnain

En esta sección se desarrolla el informe general de Pirnain. Exponiendo todas las vistas que contiene el informe y su estructura.

8. Validación

En este capítulo se detallan los procesos de validación que se han llevado a cabo durante el proyecto.

8.1. Validación de proceso

En esta sección se explican los procesos llevados implementados para validar el correcto funcionamiento de los procesos desarrollados.

8.2. Validación de visualizaciones

En esta sección se desarrollan los procesos de validación a nivel de visualización.

9. Conclusiones

En este capítulo se exponen las principales conclusiones obtenidas del proyecto.

9.1. Resultados

En esta sección se presentan las conclusiones finales del proyecto desarrollado. Exponiendo los resultados obtenidos, los objetivos logrados y los problemas encontrados durante el desarrollo.

9.2. Lineas futuras

En esta sección se proponen una serie de posibles líneas de ampliación para el proyecto.

9.3. Impacto social y medioambiental

En esta sección se expone el impacto social y medioambiental que acarrea el proyecto.

Capítulo 2

Descripción del dominio

2.1. Contexto de negocio

Pirnain es una empresa de transportes, la cual cuenta con varias franquicias alrededor de la península ibérica. El dominio del problema se centra únicamente en una franquicia, la cual tiene dificultades para acceder y analizar sus propios datos.

Pirnain oferta principalmente servicios de mensajería, aunque también dispone de envíos de sobres. Conceptualmente, un envío se puede descomponer en: *recogida*, momento en el que el paquete se traspasa del emisor a Pirnain, y *entrega*, momento en el que el paquete se transfiere de Pirnain al destinatario. Tanto las recogidas como las entregas pueden ser o bien en oficina, o bien a domicilio.

Adicionalmente, existe la figura del *contratista*. En algunos casos una tercera entidad (normalmente una empresa) encarga el servicio para un usuario externo. Por ejemplo, una empresa de venta de teléfonos móviles puede contratar los servicios de Pirnain para recoger un móvil en su almacén y posteriormente entregarlo en el domicilio de su comprador. Ver Figura 2.1.

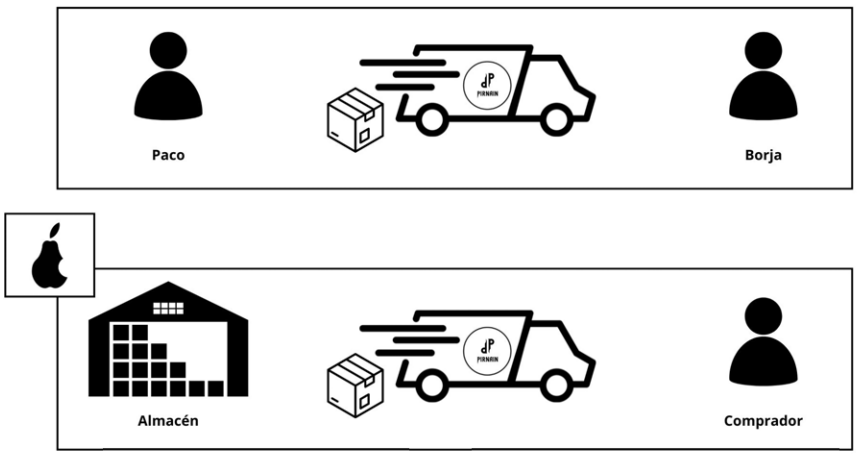

Figura 2.1: *Representación del contratista (Elaboración propia)*

Pirnain ofrece diferentes servicios dependiendo de la urgencia, peso, medidas y destino de cada envío. Pirnain funciona como cualquier tipo de franquiciador, es decir, por cada envío contratado la franquicia debe pagar unos arrastres a la matriz. En otras palabras, cuando un cliente contrata un servicio el precio se divide entre las franquicias que intervienen, y la empresa matriz.

Un envío puede ser reservado en una franquicia: para realizar una recogida en su jurisdicción, y entregarlo en otra (o en la misma), lo cual se conoce como primera ciudad; para que otra franquicia realice la recogida y luego la misma franquicia donde se ha reservado entregue el pedido, lo que tiene el nombre de segunda ciudad; o para que otra franquicia realice la recogida y otra distinta (o la misma) realice la entrega, lo que se nombra como tercera ciudad.

Como en toda empresa, existen algunos clientes que realizan un elevado número de pedidos. Para favorecer las relaciones de trabajo con los mismos, se les ofrecen tarifas acorde a sus necesidades. Principalmente existen 3 grupos: los *usuarios normales*, quienes deben pagar el importe de la tarifa oficial, los *usuarios afiliados*, quienes suelen realizar un número de envíos relativamente constante, y finalmente los *usuarios clave*, quienes realizan un considerable número de envíos y son importantes para la empresa.

Los descuentos para los *usuarios afiliados* no necesitan estar aprobados por la matriz. En consecuencia, los descuentos se realizan sobre el beneficio de la franquicia. En cambio, para los *usuarios clave* las tarifas especiales deben ser aprobadas por la matriz, y el descuento se obtiene tanto del pago a central como del beneficio de la franquicia.

2.2. Dominio del proyecto

Tras haber presentado el marco sobre el que se desarrolla el proyecto, podemos iniciar con el dominio y alcance del mismo.

Los datos operacionales de las franquicias son gestionados por un programa que ofrece la empresa matriz. El programa es un portal web donde, a grandes rasgos, se registran los clientes, envíos y tarifas. El principal problema que presenta es la gran dificultad para analizar los datos propios de la franquicia. Aunque ofrece un histórico de todos los envíos, los datos solo son accesibles mediante informes en csv o Excel. Adicionalmente, los nombres de los servicios pueden diferir entre informes, así como los códigos de tarifa o de usuario. Ante tales incoherencias e inconsistencias en los datos, junto con su falta de accesibilidad, la tarea de realizar cualquier tipo de estudio es inviable.

En un principio la franquicia quiere conocer la evolución del número de pedidos a lo largo de los años, así como el tipo de servicio más popular. Además, la franquicia necesita conocer cuantas direcciones realiza un mensajero durante el mes para calcular su desempeño. Actualmente, esto se realiza contando manualmente el número de direcciones distintas a las que ha ido un mensajero, y anotándolo en otro Excel. Lo cual supone una gran perdida de tiempo para los usuarios.

Adicionalmente, la franquicia encuentra interesante poder analizar la situación actual de la empresa: envíos en tránsito, ingresos, clientes, entre otros. Para ello se estudiarán los estados de los envíos, la evolución de los ingresos de la empresa, así como el desempeño de los diferentes clientes.

Finalmente, se quiere conocer el estado de las devoluciones realizadas por la empresa matriz y si cumplen con lo estipulado. Puesto que, existe una creencia acerca de diversos impagos en envíos, pero debido a la falta de tiempo y dificultad para visualizar los datos, no se ha podido realizar ningún estudio.

Capítulo 3

Tecnologías utilizadas

3.1. KNIME

KNIME es una plataforma open source enfocada en la analítica de Business Intelligence, Machine Learning y ETL, la cual mediante una interfaz drag & drop (arrastrar y soltar) permite crear flujos de trabajo rápidamente.

KNIME fue diseñada y desarrollada en la Universidad de Constanza, Alemania, como una herramienta de minería de datos Open Source. Actualmente, la herramienta soporta el ciclo completo de data science y machine learning, lo cual incluye: una conexión a fuentes de datos heterogéneas, automatización de procesos de ETL, principales algoritmos y métodos de evaluación para la generación de modelos, visualizaciones, web scraping, entre las muchas posibilidades que pueden ofrecer los nodos creados por la comunidad.

Durante el proyecto nos enfocaremos principalmente en las funcionalidades relacionadas con la automatización de procesos ETL, web scraping, y visualización (útil para la depuración de procesos).

3.1.1. Características principales

Las principales características que destacamos de KNIME son:

1. **Facilidad de uso:** Gracias a su interfaz visual y actualizada, KNIME es altamente intuitiva. Con el uso de nodos para representar consultas y funciones, la creación de procesos es más amigable para todo tipo de usuarios.

2. **Herramienta Open Source:** A diferencia de otras herramientas de data science, KNIME es totalmente gratuita en su versión on premise. Además, ofrece una versión de pago "KNIME Server"para ejecutar y gestionar los procesos en sus servidores dedicados.

3. **Extensa funcionalidad:** Además de ofrecer una gran variedad de nodos para cubrir el ciclo completo de business intelligence. KNIME permite a los usuarios publicar los suyos propios, de esta manera la herramienta esta en constante desarrollo de la mano de miles de personas.

La gran comunidad que tiene puede ser considerada la mayor ventaja de la herramienta. KNIME tiene a disposición un enorme foro donde los mismos usuarios se ayudan entre sí, publican tutoriales y comparten ideas de desarrollo. Más aún de ofrecer una gran ayuda a nuevos usuarios, la misma comunidad crea nuevos nodos que mejoran la funcionalidad de la aplicación.

3.1.2. Descripción de la herramienta

Para comprender de manera más sencilla algunos términos utilizados durante el proyecto, se exponen los siguientes conceptos con los que se trabaja en KNIME:

1. **Workspace:** Un workspace es el directorio en el cual se va a almacenar todo tu proyecto, incluidos workflows, configuraciones, extensiones, etc.
2. **Workflow:** Un workflow es una secuencia de nodos interconectados que representan un proceso de análisis de datos. Los datos fluyen de un nodo a otro, donde cada uno realiza una función específica.
3. **Componente:** Es una agrupación de nodos, con una entrada y una salida.

3.1.3. Entorno de trabajo KNIME

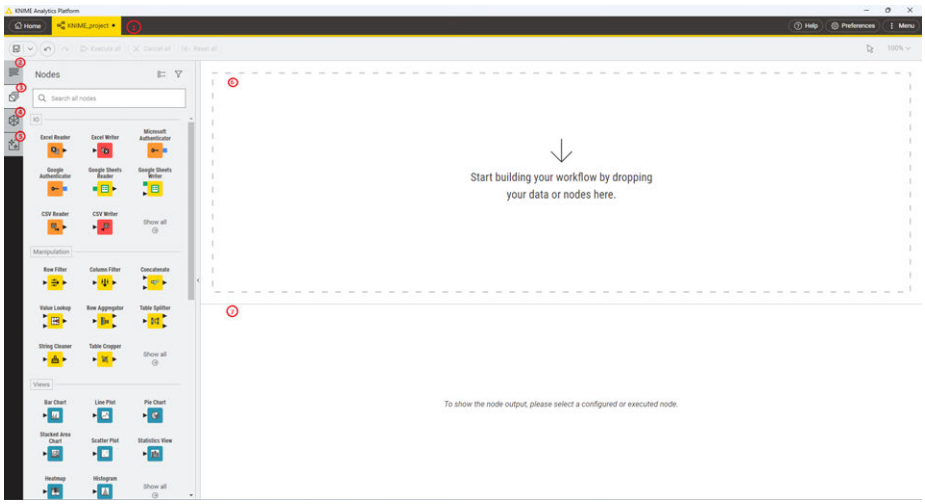

Figura 3.1: *Ventana inicial de KNIME (Elaboración propia)*

Una vez hemos instalado KNIME y abierto nuestro primer workspace, veremos la pantalla mostrada en la Figura 3.1. La cual se compone de las siguiente partes:

1. En esta zona se muestran los workflows que tenemos abiertos, en forma de pestañas.
2. **Description:** Se muestra la descripción del nodo seleccionado dentro del workflow.
3. **Node repository:** Aquí encontraremos todos los nodos. Los cuales están organizados en categorías y subcategorías. Se ofrece la opción de buscar un nodo por su nombre.
4. **Space explorer:** En este apartado se encuentran los archivos del proyecto en el que estamos trabajando.

5. **K-AI AI Assistant:** En esta ventana tenemos la posibilidad de realizar preguntas a la inteligencia artificial de KNIME. Esta, además de aclarar nuestras dudas, puede sugerirnos nodos, o realizar workflows simples.

6. **Workdlow view:** En esta zona se arrastran, conectan y visualizan los diferentes nodos del workflow.

7. **Output window:** En esta ventana se muestra la salida de un nodo ya configurado y ejecutado.

Debajo de cada nodo se encuentra un semáforo que indica su estado, ver Figura 3.2. La luz roja significa que le falta algún tipo de configuración al nodo, mientras que la luz amarilla indica que el nodo ya está configurado, pero aún no se ha ejecutado. Finalmente, la luz verde indica que se ha ejecutado correctamente. Adicionalmente, puede aparecer un triángulo con una exclamación indicando que el nodo puede estar generando una salida errónea, y un símbolo rojo con una cruz indicando la existencia de errores durante la ejecución.

Figura 3.2: *Visualización del estado de los nodos (Elaboración propia)*

3.2. Excel

Microsoft Excel es un programa desarrollado por Microsoft, el cual permite editar y visualizar hojas de cálculo. Excel cuenta con una interfaz amigable y se considera el programa para la gestión de cálculos y papeleo por antonomasia dentro de cualquier entorno laboral.

Excel divide sus hojas en celdas, sobre las cuales se pueden aplicar diferentes funciones. Las funciones que ofrece la aplicación sustituyen a las complicadas formulas matemáticas. Además de almacenar datos de manera ordenada, Excel permite generar gráficos sobre tablas creadas anteriormente.

Microsoft Excel cuenta con un lenguaje de programación llamado Visual Basic for Applications. VBA no suele ser utilizado por la mayoría de usuarios, pero puede llegar a ser de gran utilidad a la hora de automatizar diferentes procesos mediante las denominadas macros.

Microsoft VBA es un lenguaje de macros desarrollado por Microsoft Visual Basic, el cual viene integrado en Microsoft Office, y puede ser utilizado para desarrollar aplicaciones de Windows. VBA tiene acceso a varias funciones de Windows, y puede acceder a recursos como horarios, archivos o controles del propio ordenador.

Se debe usar Microsoft Excel para el proyecto, ya que es la principal herramienta de trabajo de los usuarios de negocio. En consecuencia, varias fuentes de datos provienen en este formato.

3.3. Python

Python es un lenguaje de programación interpretado, es decir, no necesita ser compilado antes de su ejecución; dinámico, esto es, que sus variables pueden tomar valores de distintos tipos; y multiplataforma.

Python es de los lenguajes más populares actualmente, siendo ampliamente utilizado en aplicaciones web, desarrollo de software, ciencia de datos y machine learning. Python ofrece una gran variedad de bibliotecas, lo que permite desarrollar una amplia gama de proyectos sobre diferentes áreas.

Aunque durante este proyecto no se utilice directamente la biblioteca Selenium, se cree conveniente mencionarla. Selenium, permite automatizar la interacción con las paginas web, mediante código. La biblioteca cuenta con varias funciones que facilitan su desarrollo.

Para este proyecto se utilizarán scripts de Python integrados en los workflows de KNIME, así como la extensión de KNIME de interacción web. Esta extensión esta desarrollada en Python y utiliza las funciones de Selenium.

3.4. Cron

Cron es un administrador de tareas de Linux que permite programar eventos y ejecutarlos en un momento determinado. Se trata de un servicio en segundo plano y sirve para programar tareas, evitando la intervención del usuario.

Para programar las tareas se debe modificar el archivo Crontab. Este documento es revisado cada minuto por Cron con dos objetivos:

1. Comprobar si existen nuevas tareas, con el fin de planificar las tareas que debe ejecutar y en qué momento hacerlo.
2. Verificar si existen tareas caducadas, con el fin de ejecutarlas en el caso correspondiente.

El archivo Crontab no debe ser modificado directamente, sino que ofrece una serie de comandos para administrarlo. Una vez configurado el sistema se encargará solo de ejecutar las tareas programadas.

Cron puede ser utilizado para ejecutar scripts de manera periódica, lanzar comandos en el servidor o, como es el caso del presente proyecto, ejecutar workflows de KNIME.

3.5. MySQL

MySQL es un sistema de gestión de base de datos relacional propiedad de Oracle. Actualmente, se considera la base de datos de código abierto más popular del mundo.

MySQL toma diferentes medidas para mantener la integridad y seguridad de la base de datos. Algunas de ellas son: la creación y gestión de usuarios, roles

y vistas, para administrar los permisos de acceso a datos; soporte de un modelo transaccional, lo que implica que, o bien, la consulta se ejecuta hasta el final, o bien, se descartan sus cambios.

La importancia de su tipo relacional reside en la necesidad de referenciar las distintas dimensiones desde la tabla de hechos. Al crear un Data Warehouse, buscamos analizar diferentes hechos de interés, cuyas propiedades se encontrarán en las dimensiones que referencia la tabla de hechos.

Se ha optado por MySQL como gestor de base de datos, por su gran versatilidad y fácil instalación. Además, gracias a su gran flexibilidad el Data Warehouse puede ser modificado con nuevas tablas y relaciones de manera sencilla.

3.6. Power BI

Microsoft Power BI es una herramienta de Business Intelligence que permite visualizar datos analíticos a través de paneles dinámicos e informes interactivos. Estos informes pueden mostrar los datos en tiempo real y de forma inmediata a todos los usuarios de negocio, según se configure la tasa de refresco y las fuentes de datos.

Power BI ofrece una gran cantidad de conexiones a fuentes de datos. Los datos pueden ser leídos desde una fuente externa sin almacenarlos en el propio informe, o por el contrario, pueden ser almacenados en Power BI en una estructura equivalente a la del origen, llamada modelo semántico.

Microsoft oferta toda una familia de aplicaciones alrededor de Power BI, la cual permite realizar los procesos ETL y prescindir de un Data Warehouse. Aunque esto no se considera una buena práctica, ya que desarrollar los procesos ETL y desplegar el Data Warehouse en un entorno externo brinda una mayor flexibilidad y eficiencia tanto en el manejo, como en el procesamiento de datos.

Se ha decidido implementar la visualización de los datos con Power BI, por su fácil uso y gran versatilidad en la creación de informes interactivos. La herramienta proporciona una interfaz de desarrollo cómoda y amigable, de tal manera que la tarea de la creación de informes se ve sumamente facilitada.

Capítulo 4

Arquitectura del sistema de BI

Para dar soporte a los requerimientos se plantea un sistema analítico basado en las tecnologías expuestas anteriormente.

El diseño de la arquitectura propuesto se muestra gráficamente en la Figura 4.1:

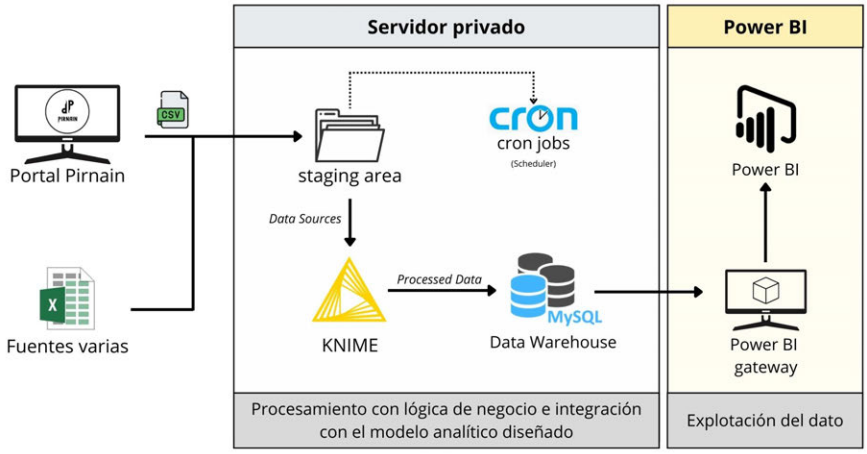

Figura 4.1: *Arquitectura Pirnain (Elaboración propia)*

En primer lugar, los datos necesarios para el análisis se extraen de 2 fuentes de datos identificadas:

- **Portal Pirnain:** El portal ofrece la posibilidad de descargar informes en formato csv son información de la franquicia. Estas descargas serán automatizadas mediante web scraping, para finalmente alojarlas en su correspondiente área de staging.
- **Fuentes varias:** Existen diversas fuentes de datos cuyo origen es difuso. En decir, los propios usuarios de negocio no tienen clara la procedencia de muchos de sus propios datos. Tras varias búsquedas, se aclaró que varios ficheros proceden de correos, mensajes archivados e informes físicos. Ante tal situación, se ha acordado en una carga manual de los propios archivos al área de staging del servidor.

Las fuentes de datos expuestas contienen los datos necesarios para dar soporte a los informes de explotación requeridos, que se implementarán con

Power BI. Sin embargo, para dar soporte a estos informes, es necesario preprocesar e integrar las fuentes de datos en el modelo de datos analítico diseñado.

Con el fin de aplicar la estructuración y lógica de negocio necesarios a las fuentes de datos, se ha identificado la necesidad de reservar un espacio de trabajo en un servidor privado que provee la franquicia. A continuación, describimos cada uno de los servicios utilizados:

- **Staging area:** Con el fin de almacenar los archivos sin procesar, se ha identificado la necesidad de reservar un área de staging para la solución. El área estará dividida en directorios donde, o bien, los archivos son cargados manualmente (fuentes con baja tasa de actualización u origen difuso), o bien, automáticamente, mediante procesos específicos.
- **KNIME:** Plataforma que permite desarrollar y ejecutar los procesos de transformación de datos requeridos, así como la automatización del staging de datos.
- **Cron jobs:** Servicio en segundo plano que permite ejecutar tareas de manera periódica, según se hayan planificado. Este servicio desplegado en el servidor permite automatizar la ingesta de datos según la planificación expuesta en la Sección 6.2.
- **MySQL server:** Almacenamiento de datos desplegado dentro del servidor privado, donde finalmente se cargan los datos procesados en el esquema del Data Warehouse.

Por último, los datos almacenados en el modelo de datos del DW son explotados mediante los informes que desarrollaremos en Power BI. Estos informes están conectados a un conjunto de datos común, cuyo refresco se implementa mediante la importación (import) incremental de datos desde el DW, tras una primera carga inicial.

Se dispone del servicio gratis de Power BI, el cual ofrece de un Área de Trabajo privada para el usuario. En consecuencia, se creará una cuenta dedicada, la cual será compartida con los usuarios de negocio.

Capítulo 5

Diseño del Data Warehouse

Para dar soporte a los informes se deben de integrar las fuentes de datos mediante el diseño de un modelo analítico. En este apartado se describirán los Data Marts que componen el Data Warehouse diseñado.

Para su diseño se han utilizado técnicas de modelado multidimensional, las cuales permiten representar el dominio de los problemas en hechos y dimensiones.

- Los **hechos** contienen métricas o información numérica que mide el rendimiento de Pirnain, como pueden ser el número de envíos, ganancias, etc.
- Las **dimensiones**, son conceptos que sirven para segmentar el análisis de las métricas, esto es, para filtrar y/o agregar datos. Algunas dimensiones identificadas son fecha, mensajero, cuenta, etc.

Los distintos Data Marts se relacionan a través de las dimensiones comunes entre sí. Además, los submodelos expuestos en este proyecto comparten una relación con el campo número de envío.

5.1. Modelo conceptual de datos

El modelo de datos a alto nivel se compone de tres Data Marts: facturación, mensajeros e incidencias. Estos Data Marts comparten algunas dimensiones comunes, además del campo `num_envio` de la tabla de hechos, por lo que se pueden relacionar y filtrar.

A continuación, mostramos el diseño de cada Data Mart del modelo de datos, identificando en color amarillo claro las tablas de hechos, en color azul señalamos las dimensiones compartidas entre los submodelos, y en gris claro las dimensiones que son únicas del propio Data Mart.

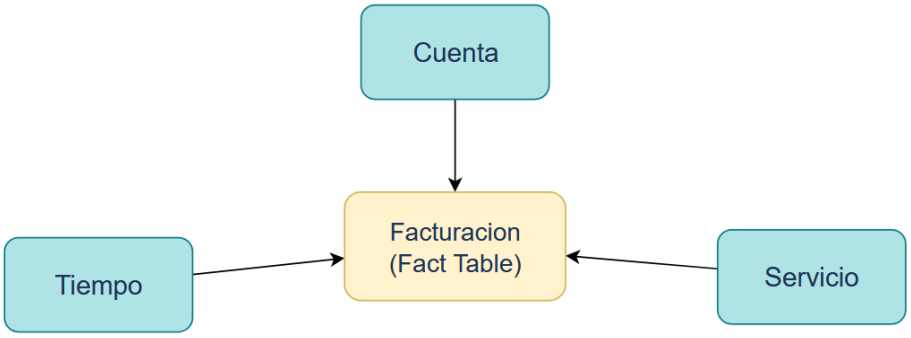

Figura 5.1: *Data mart para el estudio de la facturación (Elaboración propia)*

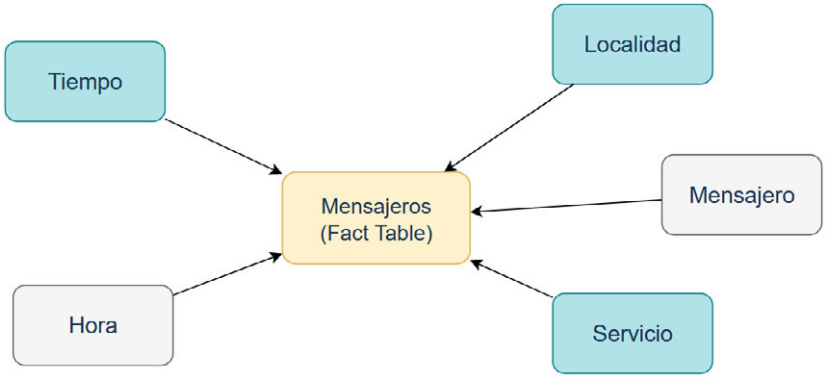

Figura 5.2: *Data Mart para el estudio de los mensajeros (Elaboración propia)*

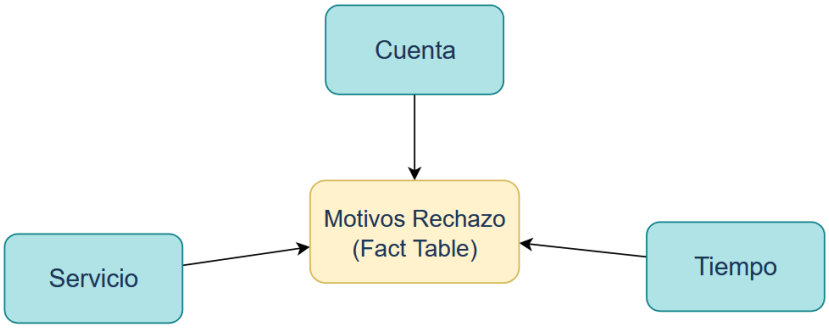

Figura 5.3: *Data Mart para el estudio de incidencias (Elaboración propia)*

5.2. Modelo lógico de datos

En esta sección, se muestra el modelo lógico de cada Data Mart del modelo de datos conceptual mostrado en el apartado anterior.

5.2.1. Data Mart Facturación

En la Figura 5.4 mostramos el modelo lógico diseñado para el análisis de los datos de `facturación`. Se identifican dos alternativas para representar las métricas de la tabla de hechos:

- El cálculo de las métricas principales (total_descuento, descuento_devuelto, etc.) se obtiene en tiempo de consulta.
- Precálculo de las métricas en el proceso de transformación de datos.

Optamos por la segunda opción con el fin de optimizar la latencia (tiempo) de ejecución de los informes de Power BI.

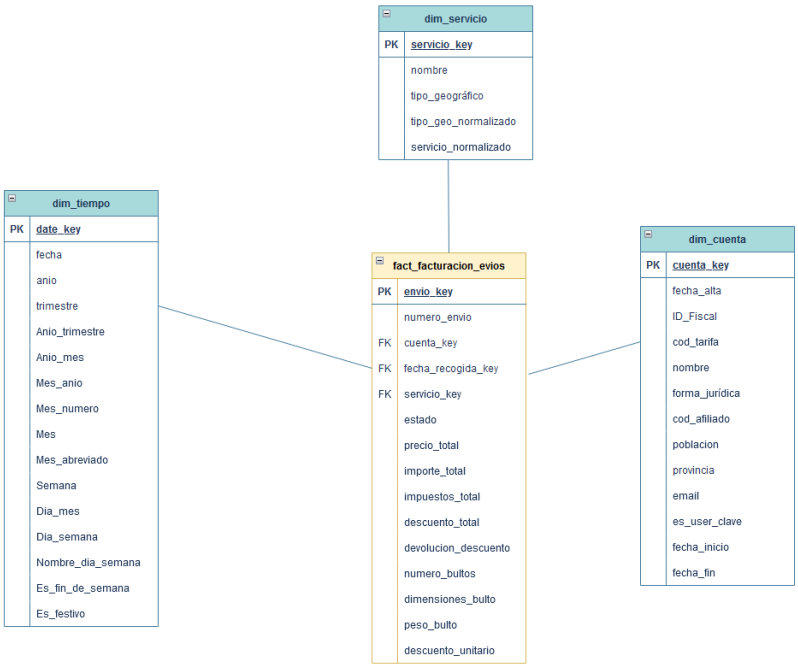

Figura 5.4: *Modelo lógico del submodelo facturación (Elaboración propia)*

5.2.2. Data Mart Mensajero

En la Figura 5.5 mostramos el modelo lógico para el análisis de los datos de `mensajeros`.

Una particularidad de este Data Mart es que en la fuente de datos de los envíos (portal Pirnain) no disponemos de la fecha prevista de entrega. Sin embargo esta fecha se puede calcular en base al servicio y la fecha de contratación o recogida,

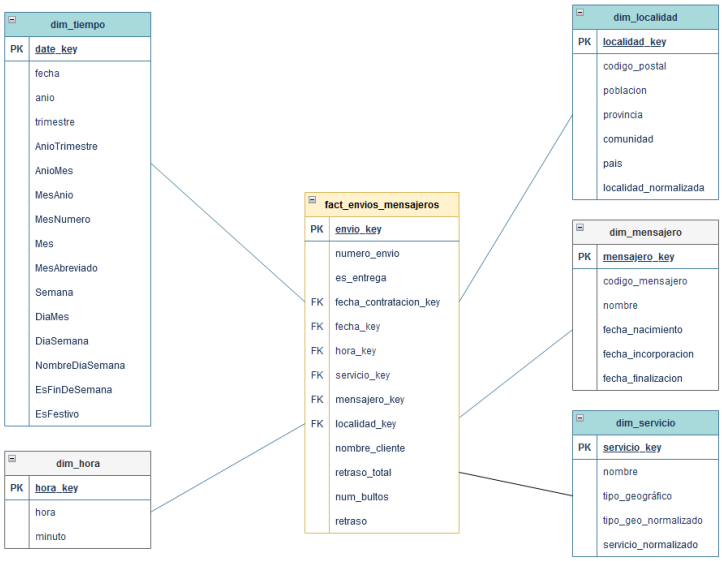

Figura 5.5: *Modelo lógico del submodelo mensajeros (Elaboración propia)*

5.2.3. Data Mart Incidencias

En la Figura 5.3 mostramos el modelo lógico del Data Mart de **incidencias**. En este submodelo no almacenamos los envíos individuales, sino que realizamos una agregación sobre los datos con el fin de optimizar el almacenamiento.

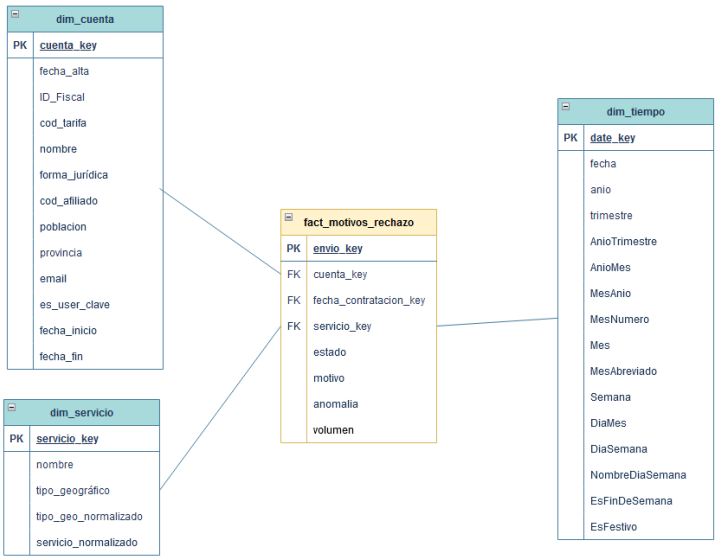

Figura 5.6: *Modelo lógico del submodelo incidencias (Elaboración propia)*

5.3. Relaciones entre los Data Marts del modelo de datos

Los Data Marts anteriores se pueden relacionar entre sí. Esta funcionalidad será necesaria para mostrar en un mismo informe de Power BI los datos de más de un Data Mart mediante tres vías de equivalencia:

- `Facturación` e `Incidencias` se relacionan con la tabla de hechos `Mensajeros` a través de las dimensiones `Tiempo` y `Servicio`
- Las tablas de hechos `Facturación` e `Incidencias` se relacionan entre sí a través de las dimensiones `Cuenta`, `Servicios` y `Tiempo`.
- Las tablas de hechos `Facturación` y `Mensajeros` se pueden relacionar entre sí, además de por las dimensiones, a través del campo `num_envío`, el cual es equivalente en las dos tablas.

5.4. Métricas

En la siguiente tabla recogemos la métricas identificadas que son calculadas mediante la fórmula indicada en la columna *Cálculo* en la herramienta de Power BI, en tiempo de ejecución del informe.

Métrica	Tipo	Agregación	`Cálculo`
Conteo de entregas	Integer	count	`CALCULATE( COUNT( fact_mensajeros [numero_envio]), fact_mensajeros [es_entrega] = TRUE( ))`
Conteo de recogidas	Integer	count	`CALCULATE( COUNT( fact_mensajeros [numero_envio]), fact_mensajeros [es_entrega] = FALSE( ))`
Conteo Servicios	Integer	count	`COUNT( 'fact_envios_mensajeros' [servicio_key])`
Total impuestos	Float	sum	`SUM( fact_factura [impuestos_total])`
Ingresos totales	Float	sum	`SUM( fact_factura [importe_total])`
Total descuentos	Float	sum	`SUM( fact_factura [descuento_total])`
Total devoluciones	Float	sum	`SUM( fact_factura [devolucion_descuento])`

Métrica	Tipo	Agregación	Cálculo
Total bultos	Integer	sum	SUM(fact_factura [numero_bultos])
Total precio	Float	sum	SUM(fact_factura [precio_total])
Retraso medio en recogidas (H)	Float	average	CALCULATE(AVERAGE(fact_mensajeros [retraso]), fact_mensajeros [es_entrega] = FALSE( ))
Retraso medio en entregas (H)	Float	average	CALCULATE(AVERAGE(fact_mensajeros [retraso]), fact_mensajeros [es_entrega] = TRUE( ))
Retraso medio en envios (H)	Float	average	AVERAGEX(VALUES('fact_mensajeros' [numero_envio]), CALCULATE(MAX('fact_mensajeros' [retraso_total])))
Total finalizados	Integer	sum	CALCULATE(SUM(fact_motivos [volumen]), fact_motivos [Estado] == .Envio finalizado")
Envios completados	Integer	sum	CALCULATE(SUM(fact_motivos [volumen]), fact_motivos [Estado] = .envio completado")
En transito totales	Integer	sum	CALCULATE(SUM(fact_motivos [volumen]), fact_motivos [Estado] = .en tránsito")
Ingresos YTD	Float		TOTALYTD([Ingresos totales], 'dim_tiempo' [Fecha])
Ingresos MTD	Float		TOTALMTD([Ingresos totales], 'dim_tiempo' [Fecha])
Ingresos P-1	Float		CALCULATE([Ingresos totales], PARALLELPERIOD(dim_tiempo [Fecha], -12, MONTH))
Ingresos % Growth	Float		DIVIDE(([Ingresos totales] - [Ingresos P-1]), [Ingresos P-1])*100
Descuadres Descuentos	Float		DIVIDE([Total descuentos] - [Total devoluciones], [Total descuentos])

Métrica	Tipo	Agregación	Cálculo
EnviosAnulados	Integer	sum	`CALCULATE( SUM( fact_motivos [volumen]), fact_motivos[Estado] IN {"devuelto", çancelado"})`
Evolución ingresos	Float		`[Ingresos totales] - [Ingresos P-1]`
TotalDiasUltimoMes	Integer		`DATESINPERIOD( 'dim_tiempo' [Fecha], LASTDATE( 'dim_tiempo' [Fecha]), -1, MONTH)`
TotalEnviosUltimoMes	Integer	count	`CALCULATE( DISTINCTCOUNT( fact_factura [Número de envío]), DATESINPERIOD( 'dim_tiempo' [Fecha], LASTDATE( 'dim_tiempo' [Fecha]), -1, MONTH))`
PromedioEnviosUltimoMes	Integer		`[TotalEnviosUltimoMes] / TotalDiasUltimoMes`
% Envios cancelados	Float		`[EnviosAnulados] / [Conteo envios totales]`
Total a devolver	Float		`[Total descuentos] - [Total devoluciones]`
Conteo envios totales	Integer	count	`DISTINCTCOUNT( fact_mensajeros[numero_envio])`
Conteo cuentas	Integer	count	`DISTINCTCOUNT( dim_cuenta [Código afiliado])`
Conteo envios cuentas	Integer	count	`DISTINCTCOUNT( fact_factura [Número de envío])`
Volumen envíos	Integer		`SUM( fact_motivos [volumen])`
Media bultos	Integer		`DIVIDE( [Total bultos], [Conteo envios totales])`

Tabla 5.1: *Medidas calculadas en Power BI*

5.5. Entidades

Como comentamos previamente, en nuestro modelo analítico identificamos dos tipos de entidades: dimensiones y tablas de hechos. A continuación, se enumeran y describen las entidades que constituyen el modelo de datos analítico:

- **Dimensiones**
  - **Cuenta:** Contiene información de las cuentas de los clientes. Se trata de una dimensión de variación lenta *tipo 4*.
  - **Localidad:** Contiene el nombre de los municipios y su respectivo país. Se trata de una dimensión de variación lenta *tipo 1*.
  - **Servicio:** Contiene el nombre de los servicios. Se puede considerar una dimensión de variación lenta *tipo 2*, al insertar una nueva fila en el momento en el que se modifica un campo, sin necesidad de mantener un histórico.
  - **Mensajero:** Contiene el nombre de los mensajeros. Se trata de una dimensión de variación lenta *tipo 4*, la cual mantiene un histórico de los mensajeros mediante el archivo de entada, sin necesidad de tratarlo en el proceso ETL.
  - **Tiempo:** Trata la información de fechas. Si se tuviera que catalogar dentro de un tipo de dimensión de variación lenta sería en el *tipo 1*, debido que sus modificaciones sólo podrían tratarse de errores.
  - **Hora:** Contiene las horas y minutos. Si se tuviera que catalogar dentro de un tipo de dimensión de variación lenta sería en el *tipo 1*, dado que sus modificaciones sólo podrían tratarse de errores.
- **Tablas de hechos**
  - **Facturación:** Hechos de los envíos en torno a sus gastos e ingresos.
  - **Mensajeros:** Análisis del rendimiento de los mensajeros.
  - **Incidencias:** Análisis de las incidencias de los diferentes envíos.

5.6. Atributos

5.6.1. Dimensión Cuenta

Campo	Tipo	Descripción
Cuenta_key	Numérico	Generado en el proceso ETL
fecha_alta	Date	Fecha de incorporación de la cuenta
ID_Fiscal	Texto	ID fiscal de la cuenta
cod_tarifa	Texto	Codigo de la tarifa de la cuenta
nombre	Texto	Nombres de la cuenta
forma_jurídica	Texto	Forma jurídica de la cuenta
cod_afiliado	Texto	Código de afiliado de la cuenta
poblacion	Texto	Población de facturación
provincia	Texto	Provincia de facturación
email	Texto	Email de contacto
es_user_clave	Boolean	es un usuario clave?
fecha_inicio	Date	Fecha de inicio de la cuenta
fecha_fin	Date	Fecha de baja de la cuenta

Tabla 5.2: *Descripción de los atributos de la dimensión Cuenta*

5.6.2. Dimensión Localidad

Campo	Tipo	Descripción
localidad_key	Numérico	Generado por el proceso ETL
codigo_postal	Texto	Código postal
poblacion	Texto	Población
provincia	Texto	Provincia
comunidad	Texto	Comunidad autónoma
pais	Texto	Nombre del país
localidad_normalizada	Texto	Localidad normalizada (codigo_postal+poblacion)

Tabla 5.3: *Descripción de los atributos de la dimensión Localidad*

5.6.3. Dimensión Servicio

Campo	Tipo	Descripción
servicio_key	Numérico	Generado por el proceso ETL
nombre	Texto	Nombre del servicio
tipo_geografico	Texto	Tipo (Cercanal, Nacional, Provincial)
tipo_geo_normalizado	Texto	Tipo geografico normalizado
servicio_normalizado	Texto	Servicio normalizado (nombre + tipo_geo_normalizado)

Tabla 5.4: *Descripción de los atributos de la dimensión Servicio*

5.6.4. Dimensión Mensajero

Campo	Tipo	Descripción
mensajero_key	Numérico	Generado por el proceso ETL
codigo_mensajero	Texto	Clave operacional distintiva del mensajero
nombre	Boolean	Nombre del mensajero
fecha_nacimiento	Numérico	Fecha de nacimiento
fecha_incorporacion	Numérico	Fecha de incorporación
fecha_finalización	Numérico	Fecha de finalización

Tabla 5.5: *Descripción de los atributos de la dimensión Mensajeros*

5.6.5. Dimensión Hora

Campo	Tipo	Descripción
hora_key	Numérico	Generado por el proceso ETL
hora	Numérico	Hora
minuto	Numérico	Minutos

Tabla 5.6: *Descripción de los atributos de la dimensión Hora*

5.6.6. Dimensión Tiempo

Campo	Tipo	Descripción
Date_key	Numérico	Generado por el proceso ETL
fecha	Date	fecha completa
Anio	Numérico	Año
Trimestre	Numérico	Trimestre
Anio_trimestre	Texto	Año + Trimestre
Anio_mes	Texto	Año + numero mes
Mes_anio	Texto	Nombre mes + año
Mes_numero	Numérico	Numero del mes
Mes	Texto	Nombre de un mes
Mes_abreviado	Texto	Primeras 3 letras del mes
Semana	Numérico	Numero de la semana del año
Dia_mes	Numérico	Día del mes
Dia_semana	Numérico	Número del día de la semana
Nombre_dia_semana	Texto	Nombre del día de la semana
Es_fin_de_semana	Boolean	Es fin de semana?
Es_festivo	Boolean	Es festivo?

Tabla 5.7: *Descripción de los atributos de la dimensión Tiempo*

5.6.7. Tabla de hechos Facturación

Campo	Tipo	Descripción
envio_key	Numérico	Generado por el proceso ETL
numero_envio	Texto	Clave operacional
cuenta_key	Numérico	id de la cuenta
fecha_recogida_key	Numérico	id de la fecha de recogida
servicio_key	Numérico	id del servicio
precio_total	Numérico	Precio de venta final
importe_total	Numérico	Importe total sin impuestos
impuestos_total	Numérico	Impuestos totales pagados
descuento_total	Numérico	Total descuentos
devolucion_descuento	Numérico	Total devuelto
numero_bultos	Numérico	Número de bultos
dimensiones_bulto	Numérico	Dimensiones del bulto
peso_bulto	Numérico	Peso del bulto
descuento_unitario	Numérico	Descuento aplicado al bulto

Tabla 5.8: *Descripción de los atributos de la tabla de hechos Facturación*

5.6.8. Tabla de hechos Mensajeros

Campo	Tipo	Descripción
envio_key	Numérico	Generado por el proceso ETL
numero_envio	Texto	Clave operacional
es_entrega	Boolean	True: Entrega, False: Recogida
fecha_contratacion_key	Numérico	id de la fecha de contratación
fecha_key	Numérico	Id de la fecha de la acción
hora_key	Numérico	Id de la hora de la acción
servicio_key	Numérico	Id del servicio
mensajero_key	Numérico	Id del mensajero
localidad_key	Numérico	Id de la localidad
nombre_cliente	Texto	Nombre del cliente que entrega/recibe
retraso_total	Numérico	Retraso total del envío
num_bultos	Numérico	Número de bultos
retraso	Numérico	Retraso en la entrega/recogida

Tabla 5.9: *Descripción de los atributos de la tabla de hechos Mensajeros*

5.6.9. Tabla de hechos Incidencias

Campo	Tipo	Descripción
envio_key	Numérico	Generado por el proceso ETL
cuenta_key	Numérico	id de la cuenta
fecha_contratacion_key	Numérico	id de la fecha de contratación
servicio_key	Numérico	id del servicio
estado	Texto	Estado del envío
motivo	Texto	Motivo del rechazo
anomalia	Texto	Nombre de la anomalía
volumen	Numérico	Recuento de envíos

Tabla 5.10: *Descripción de los atributos de la tabla de hechos Incidencias*

Capítulo 6

Procesos ETL

En este capitulo se identifican los procesos de preparación de datos con los que se cumplirán los objetivos identificados.

6.1. Estructuración y aplicación de lógica de negocio

OBJETIVO	Preprocesamiento y carga/actualización de los datos en el modelo analítico diseñado.
CONDICIONES DE ENTRADA	Disponibilidad de los datos de envíos en Staging.
CONDICIONES DE SALIDA	Datos cargados y actualizados en el modelo de datos.
DESCRIPCIÓN	Es necesario automatizar la ejecución de este proceso. Para ello se programa una ejecución periódica que inicie el procesamiento y carga/actualización de datos en el modelo de datos que se implementa en DW.

Tabla 6.1: Objetivo de los procesos

6.2. Planificación

En esta sección recogemos la información relativa a la frecuencia de refresco planificada. A continuación, recogemos la programación del refresco de datos en la arquitectura y procesos específicos implementados para la solución digital.

6.2.1. Refresco de datos en DWH

1. **Continuos**, cada 7 días: Tablas de hechos `fact_envíos_mensajeros`, `fact_facturación_envíos` y `fact_motivos_rechazo`, mediante la programación de sus ejecuciones en el servicio Cron.

2. **Mensuales**, cada 30 días: Dimensiones `dim_servicio` y `dim_cuenta`, mediante la programación de sus procesos ETL en Cron.

3. **Bajo demanda**, activadas por la carga de archivos manuales en el area de staging, alertada por el usuario de negocio. Dimensiones `dim_mensajero` y `dim_localidad`

En el siguiente diagrama se describen a alto nivel las tareas a realizar en el proceso de envíos que se realizan con una frecuencia de 7 días:

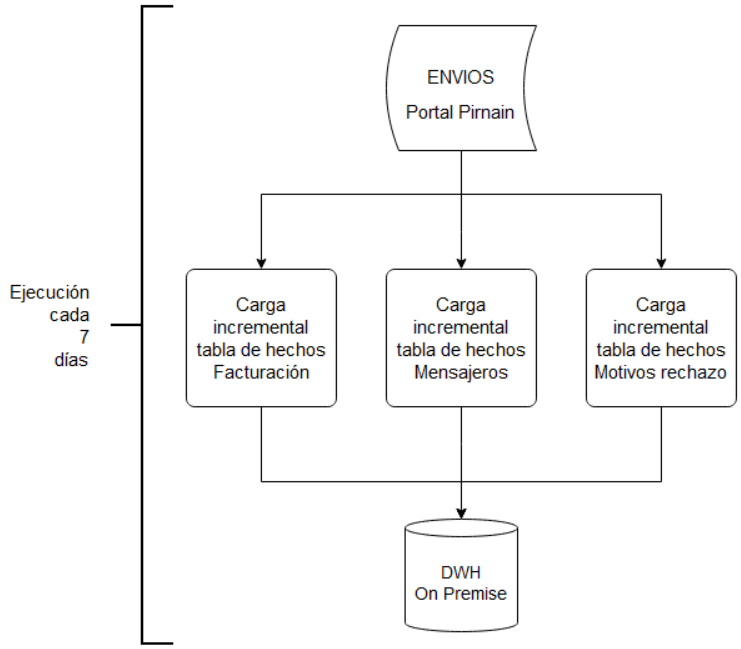

Figura 6.1: *Procesos semanales (Elaboración propia)*

La carga o actualización de las dimensiones que se realiza con frecuencia mensual o manual.

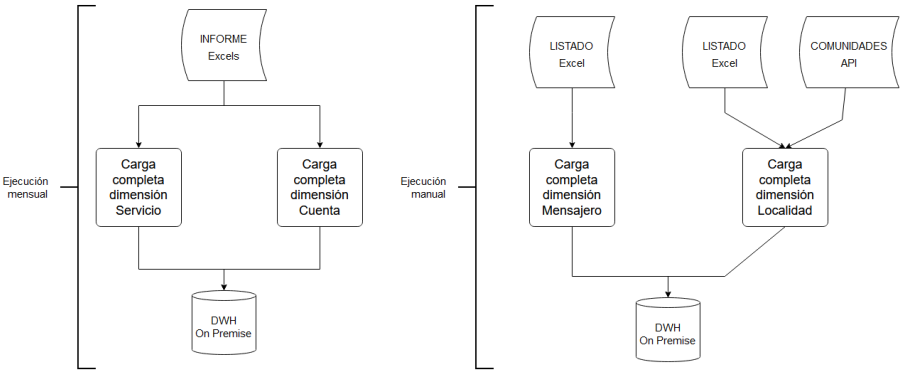

Figura 6.2: *Procesos mensuales y manuales (Elaboración propia)*

La generación de las dimensiones Tiempo y Hora se realiza una única vez, pudiendo ejecutar de nuevo el proceso si se desea regenerar las dimensiones.

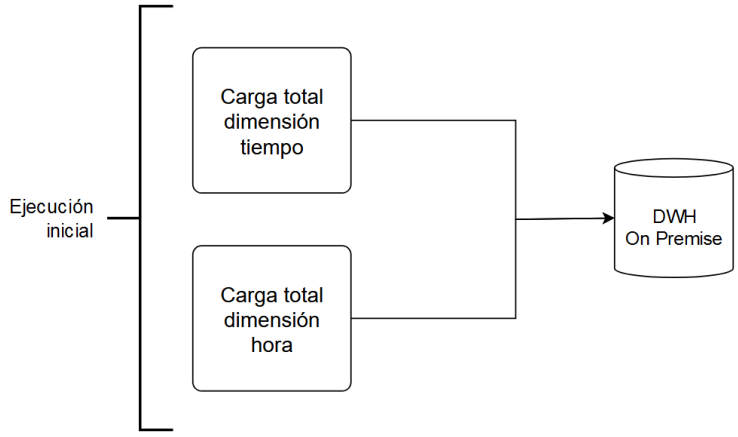

Figura 6.3: *Procesos iniciales (Elaboración propia)*

6.3. Obtención de los hechos

En esta sección se exponen los procesos encargados de poblar las tablas de hechos del Data Warehouse con los datos procesados. Además, para el primer proceso se detallarán algunos de los nodos/partes más importantes del proceso, con el fin de facilitar la comprensión del desarrollo con la herramienta KNIME.

6.3.1. 00_fact_facturación_envíos

Este proceso es el encargado de la extracción y carga de los datos referentes a la facturación de la franquicia de Pirnain.

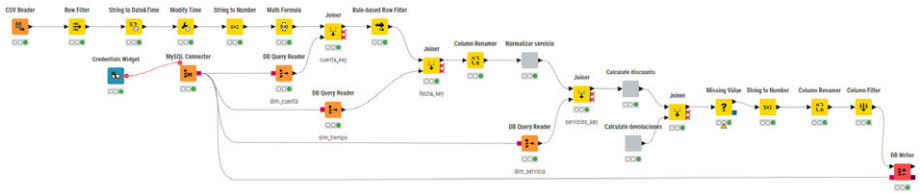

Figura 6.4: *Proceso ETL fact facturación (Elaboración propia)*

Como podemos observar en la Figura 6.4 el proceso inicia leyendo de un CSV de entrada, para posteriormente aplicar diferentes transformaciones a los datos, se calculan las medidas derivadas y finalmente se cargan los datos en la tabla de hechos del Data Warehouse.

Algunos de los nodos a destacar son:

1. **CSV Reader**: Este nodo permite leer datos de uno o varios ficheros csv, siempre y cuando tengan la misma estructura. La ruta de los ficheros puede ser absoluta, o relativa al workspace o Hub Space, gracias a esto se puede migrar un proyecto entre varios entornos de forma eficaz.

   El mismo nodo permite, seleccionar las columnas de la entrada, así como renombrarlas o especificar su tipo (String, Integer, etc.). Adicionalmente,

se puede especificar la codificación a utilizar (UTF-8, UTF-16, ISO-8859-1, etc.).

2. **Row Filter**: Este nodo permite filtrar las filas de entrada mediante varios métodos. Estos son: comparación del valor de una columna con una expresión regular o valor objetivo, por su número de fila, o por su RowID.

3. **Joiner**: Este nodo permite realizar un JOIN entre dos entradas. Lo interesante del mismo, es la posibilidad de separar la salida en tres dataflows distintos, en función del emparejamiento (las filas coincidentes, las no coincidentes de la izquierda, y las no coincidentes de la derecha).

   Además, se puede configurar para que realice un INNER JOIN, LEFT JOIN o RIGHT JOIN. Así como unificar las columnas con el mismo nombre y seleccionar únicamente las requeridas en la salida.

4. **Credentials Widget**: Este nodo puede encargarse, o bien, de almacenar las credenciales (nombre de usuario y contraseña) débilmente cifradas en la memoria de KNIME, o bien, de solicitar las credenciales de usuario al iniciar el proceso.

5. **MySQL Connector**: Este nodo ofrece una sesión de MySQL sobre la cual se pueden ejecutar operaciones en una determinada base de datos. La sesión que proporciona, sirve para realizar consultas, escrituras, y actualizaciones como se puede observar en los nodos conectados al mismo.

6. **Componente**: KNIME ofrece la posibilidad de crear agrupaciones de nodos, las cuales son denominadas componentes. Estos componentes deben recibir un nombre y son de gran utilidad para organizar los workflows y reutilizar secuencias de nodos. Como podemos observar en la Figura 6.4, para este proceso se han generado 3 componentes en este nivel (`Normalizar servicio`, `Calculate discounts`, `Calculate devoluciones`).

Dentro de este proceso `00_fact_facturación_envíos` podemos adentrarnos en el nodo `Calculate discounts`, para observar las transformaciones que ejecuta.

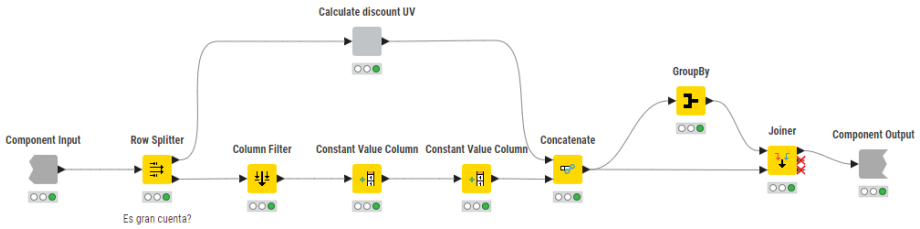

Figura 6.5: *Componente Calculate discounts (Elaboración propia)*

Como podemos observar en la Figura 6.5, dentro de un componente se puede generar otro componente. Los componentes se almacenan como ramificaciones del workflow donde se utilizan, por lo que ahora mismo nos encontraríamos en el nivel 1.

El componente `Calculate discounts` trata de generar los descuentos aplicados de los pedidos, en función de la cuenta que lo ha solicitado. Como las grandes cuentas requieren de un proceso complejo para calcular sus descuentos, este se agrupa en un nuevo componente.

Como se muestra en la Figura 6.6, KNIME nos indica en que nivel y nodo nos encontramos editando. Para la realización de este proceso, se ha necesitado un análisis previo de los datos del cual se ha obtenido un mapeo con las tarifas y códigos de descuento de los diferentes usuarios.

Adicionalmente, las hojas de descuentos han sido previamente procesadas mediante una macro de Excel, cuyo desarrollo fue previo a este proyecto. Esta macro permite organizar los datos en tablas, de tal manera que se puedan utilizar como input en procesos automatizados. Anteriormente, los descuentos se encontraban en hojas des-estructuradas inservibles para automatizaciones.

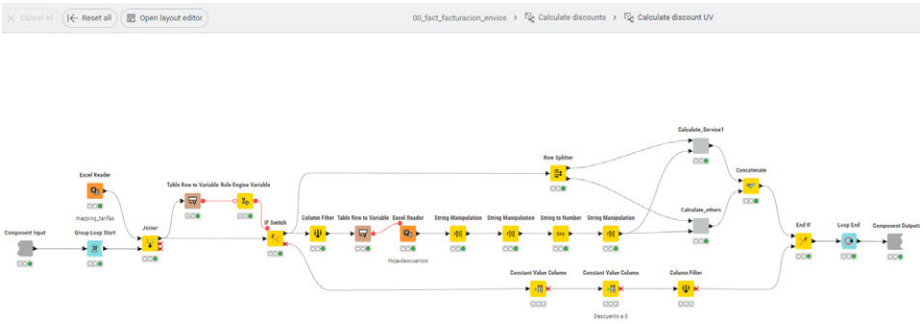

Figura 6.6: *Componente Calculate discount UV (Elaboración propia)*

En este proceso es interesante destacar como se ha derivado el valor de la columna cuenta a una variable. Posteriormente, esta variable se ha utilizado para leer la hoja correspondiente a los descuentos del cliente. Además, se quieren destacar los siguientes nodos:

1. **Group Loop Start y Loop End**: El nodo `Group Loop Start` permite iniciar un bucle, donde en cada iteración se procesan un grupo de datos definidos. Por ejemplo, en la Figura 6.6 se procesan todos los envíos de una cuenta por iteración. De esta manera mejoramos la eficiencia de lectura de las hojas de descuentos, ya que solo accedemos al fichero una única vez por cuenta.

2. **Rule Engine Variable**: Este nodo junto con el nodo `Column Expresssion` pertenecen a la extensión `KNIME Javasnippet`. Estos nodos permiten aplicar diferentes una gran variedad funciones a variables o columnas, respectivamente.

6.3.2. 00_fact_motivos_rechazo

Este proceso es el encargado de transformar y cargar los datos relativos a los motivos de rechazo de los envíos relacionados con la franquicia de Pirnain, en el Data Warehouse.

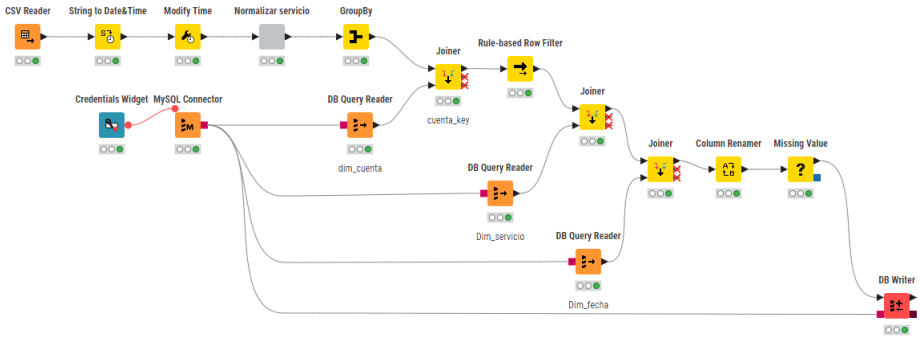

Figura 6.7: *Proceso ETL fact motivos rechazo(Elaboración propia)*

Como se puede observar en la Figura Figura 6.7, los datos son agrupados de tal manera que se reducen considerablemente los registros de la tabla de hechos. La agrupación se produce por los campos `cuenta_key`, `fecha_contratacion_key`,`estado`, `motivo` y `anomalía`, y se realiza un recuento de la agrupación para posteriormente poder visualizar el volumen de anomalías.

6.3.3. 00_fact_envíos_mensajeros

Este procedimiento se encarga de procesar y cargar los datos relativos a los envíos realizados por los mensajeros de la franquicia de Pirnain. Ver Figura 6.8.

Figura 6.8: *Proceso ETL fact envíos mensajeros(Elaboración propia)*

Para este proceso se han generado 2 componentes principales: `Crear recogida` y `Crear entrega`, los cuales han sido reutilizados dentro de `Crear recogida_entrega`. Este proceso se encarga de separar los diferentes envíos según su naturaleza (entrega, recogida o ambas), en función de varios campos definidos anteriormente por los usuarios de negocio.

Como se expuso en la descripción del dominio los envíos pueden clasificarse en primera, segunda y tercera ciudad. En este caso solo se tienen en cuenta los envíos de primera y segunda ciudad, en los que están involucrados los mensajeros de la franquicia.

Adicionalmente, se han encapsulado los JOINs con las dimensiones en un nuevo componente `JOINs dimensiones`. Se encuentra de interés mostrar este componente ya que utiliza un nodo particularmente útil en la mayoría de procesos ETL, el nodo `Value Lookup`.

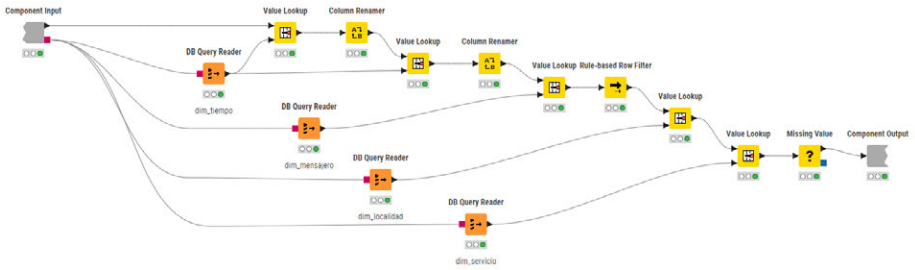

Figura 6.9: *Componente JOINs dimensiones(Elaboración propia)*

El nodo `Value Lookup` tiene dos entradas: la tabla de datos y la tabla diccionario. En la salida se añaden las columnas requeridas de la tabla diccionario en la tabla de datos, en función de una columna que permite la unión entre ambas.

Como se puede apreciar en las Figuras 6.4 y 6.7, este nodo ha sido reemplazada por el nodo `Joiner`. Esto se debe a que, aunque el nodo `Value Lookup` ofrece un mejor rendimiento, en cuestión de configuración no ofrece una personalización tan amplia como su competidor. Por ende, en los procesos `00_fact_facturación_envíos` y `00_fact_motivos_rechazo` no se han implementado con este nodo.

6.4. Obtención de las dimensiones

En esta sección se presentan los procesos necesarios para obtener las columnas de cada tabla dimensión a partir de los ficheros de las fuentes de datos.

A continuación, se describen los procesos de obtención de las dimensiones.

6.4.1. 00_dim_mensajero

Este proceso es el encargado de la extracción y carga de datos referentes a los mensajeros en la dimensión mensajero del Data Warehouse.

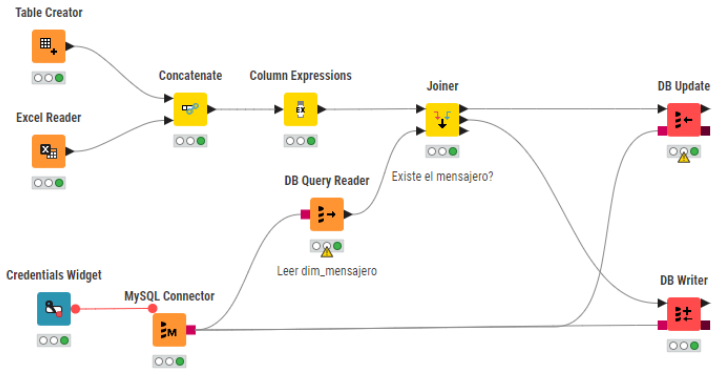

Figura 6.10: *Proceso ETL mensajero (Elaboración propia)*

En este procedimiento no se transforman los datos de entrada. Únicamente se realiza un select sobre el Excel de entrada para obtener las columnas requeridas. Posteriormente se genera un registro vacío para evitar inconsistencias en el Data Warehouse.

Este es un proceso incremental, de tal manera que se comprueba si existe un registro en la dimensión, antes de insertarlo. Si ya existe el registro, pero se ha modificado el campo `fecha_finalización` se actualiza en la dimensión. Aún tratándose de una dimensión tipo 4, el proceso no gestiona el histórico de los mensajeros, ya que el periodo de contratación de los mismos es reflejado en el archivo de entrada.

6.4.2. 00_dim_localidad

Este proceso se encarga de la extracción, transformación y carga de datos sobre la dimensión localidad del Data Warehouse.

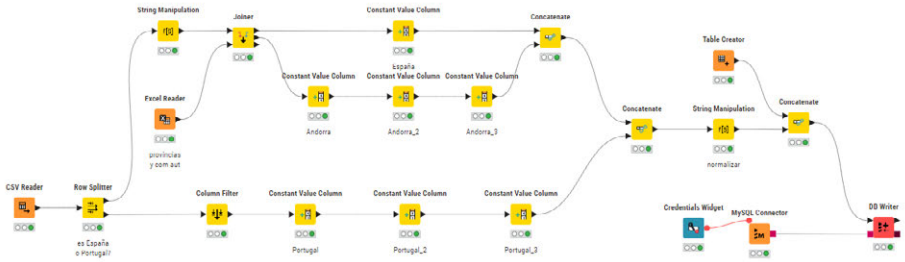

Figura 6.11: *Proceso ETL localidad (Elaboración propia)*

En este procedimiento se cargan todas las localidades y municipios en los que Pirnain ofrece servicio. Como los datos de origen no cumplen los requerimientos, se tienen que aplicar ciertas transformaciones para su posterior uso en consultas, y visualizaciones sobre mapas.

6.4.3. 00_dim_servicio

Este proceso es el encargado de tratar los diferentes servicios que ofrece Pirnain, junto con su carga en la dimensión servicio.

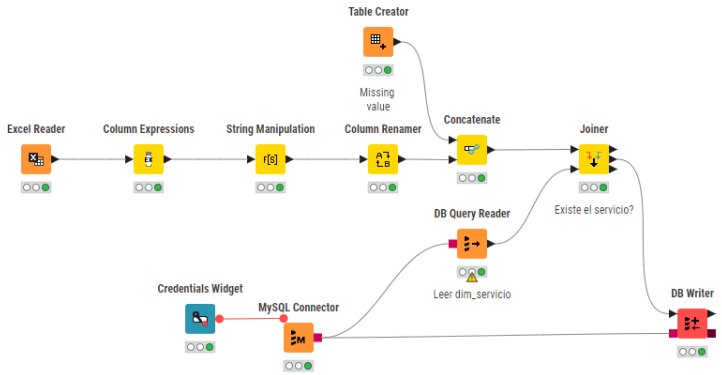

Figura 6.12: *Proceso ETL servicio (Elaboración propia)*

En este procedimiento podemos apreciar una carga incremental. Es decir, los servicios ya existentes en la dimensión no son cargados nuevamente, sino actualizados en el caso de que hayan sido modificados. El proceso únicamente genera un atributo derivado el cual normaliza el nombre del servicio y su tipo geográfico.

6.4.4. 00_dim_cuenta

Este proceso es el encargado de tratar todas las cuentas asociadas con la franquicia de Pirnain, junto con la carga en su dimensión correspondiente al Data Warehouse.

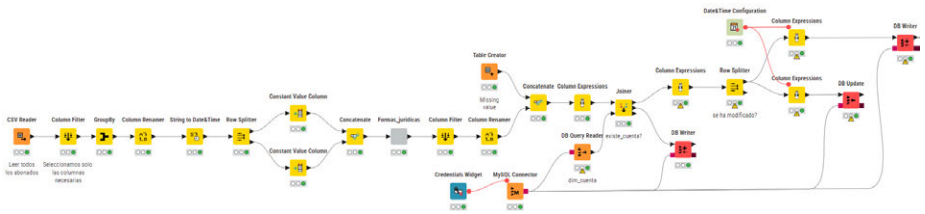

Figura 6.13: *Proceso ETL cuenta (Elaboración propia)*

Como podemos observar en la Figura 6.13, se ha creado un nuevo nodo componente `Formas_jurídicas` el cual se encarga de corregir los campos nombre y `forma_jurídica` de tal manera que mantengan una coherencia. Por ejemplo, algunos registros no cuentan con el campo `forma_jurídica`, sin embargo, esta se encuentra incluida en su nombre.

Este es un proceso incremental sobre una dimensión tipo 4, es decir, que las modificaciones sobre los registros deben reflejar una evolución histórica. Para ello se puede apreciar como se realiza un JOIN con la dimensión del Data Warehouse al final del proceso. Gracias a esta operación podemos comprobar la existencia del registro. Si el registros se encuentra en la dimensión, se comprueba que haya sido modificado y en caso afirmativo se añade una nueva fila, y se modifica la anterior para que refleje el fin de su validez en el campo `fecha_fin`.

6.4.5. 00_dim_tiempo

Este proceso es el encargado de generar todas las fechas y campos derivados para la dimensión tiempo del Data Warehouse.

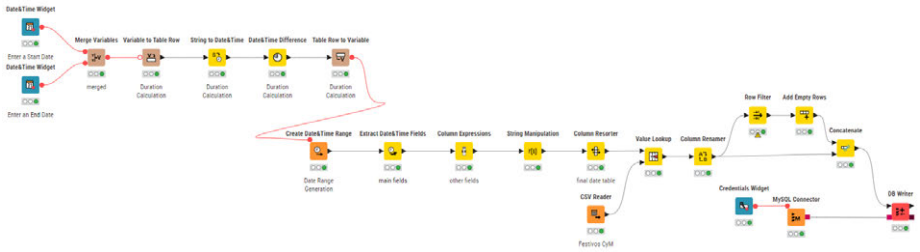

Figura 6.14: *Proceso ETL tiempo (Elaboración propia)*

Como se puede observar en la Figura 6.14 el proceso genera todas las fechas dentro de un rango introducido como parámetro al iniciar el proceso. Si no se especifica ninguna fecha manualmente, se utilizan los valores por defecto (2020-2025). Además, el proceso lee de un CSV el cual contiene los días festivos de la comunidad de Castilla-La Mancha.

6.4.6. 00_dim_hora

Este proceso se encarga de generar todos los registros necesarios para poblar la dimensión hora del Data Warehouse.

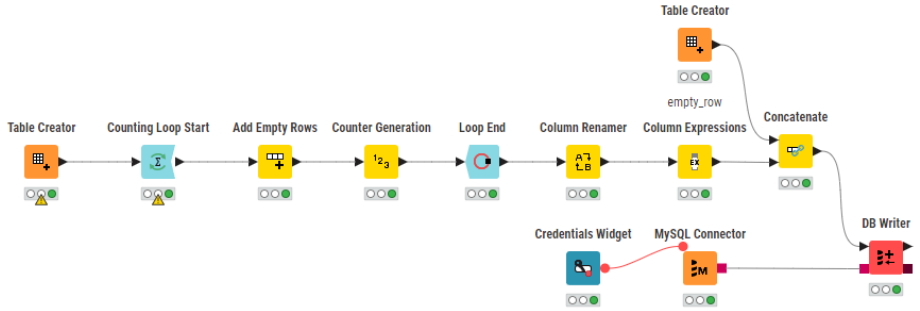

Figura 6.15: *Proceso ETL hora (Elaboración propia)*

Como se puede apreciar en la figura, el proceso genera todas las combinaciones de horas y minutos posibles. Esto se realiza mediante un bucle y un contador que generan los 1440 registros necesarios. Además, posteriormente se realiza una derivación que permite obtener la clave de la dimensión dentro del Data Warehouse.

6.5. Procesos de Staging

En esta sección se exponen los procesos desarrollados en KNIME cuyo fin es automatizar la ingesta de datos de las fuentes de entrada a la zona de staging del servidor.

6.5.1. 09_staging_envíos

Este proceso se encarga de interactuar con el portal de Pirnain con el objetivo de descargar los ficheros relativos a los envíos para un rango de fechas determinado.

Figura 6.16: *Proceso staging envíos (Elaboración propia)*

Para implementar este Workflow se ha utilizado la extensión `KNIME Web Interaction (Labs)`, para los nodos que interactúan con la web. Junto con la extensión `KNIME Python Integration`, la cual permite ejecutar scripts de Python dentro del mismo Workflow.

Como se puede observar en la Figura 6.16 se aprovecha la secuencialidad que ofrece traspaso de variables entre nodos, para poder realizar bucles y diferentes flujos mediante los nodos `Group Loop Start` y `If switch`. Ante la ausencia de compatibilidad en las entradas y salidas de los anteriores nodos, se ha propuesto esta solución que ofrece resultados aceptables.

Cabe destacar que este proceso tiene que ser llamado con el rango de fechas sobre el cual se quieren obtener los datos relativos a los envíos. Como se puede ver en la Figura 6.17, en el workflow se encuentra el nodo `Container Input (Variable)`, este nodo recibe las variables de flujo introducidas en la llamada del proceso.

Figura 6.17: *Input rango de fechas (Elaboración propia)*

Como el portal de Pirnain únicamente permite descargar los archivos de los envíos con un rango máximo de 90 días, el proceso se encarga de dividir el rango de fechas total en grupos con esta duración máxima. Posteriormente, se itera en un bucle sobre estos subrangos obteniendo el rango total introducido.

6.5.2. 09_staging_festivos

Con el objetivo de simplificar el mantenimiento del sistema, se ha automatizado la descarga de los días festivos en la Comunidad de Castilla-La Mancha.

Figura 6.18: *Proceso staging festivos (Elaboración propia)*

Este proceso se encarga de descargar el calendario de festivos del portal de Castilla-La Macha, e iterar sobre el mismo hasta marcar todos los días del año como festivos o no. El proceso tiene como entrada un rango de años, como resultado se pueden obtener los días festivos de varios años en una misma ejecución.

El workflow genera un csv de salida, el cual coloca en la zona de staging. Posteriormente, este fichero será utilizado por el proceso `00_dim_tiempo`

6.6. Procesos generales

En esta sección se muestran los procesos complementarios que tienen como objetivo facilitar la ingesta de datos sobre dimensiones o tablas de hechos recurrentes.

6.6.1. 02_full_load

Con el objetivo de facilitar la carga completa del Data Warehouse se ha desarrollado un workflow que permite realizar la ingesta de datos completa de forma sencilla.

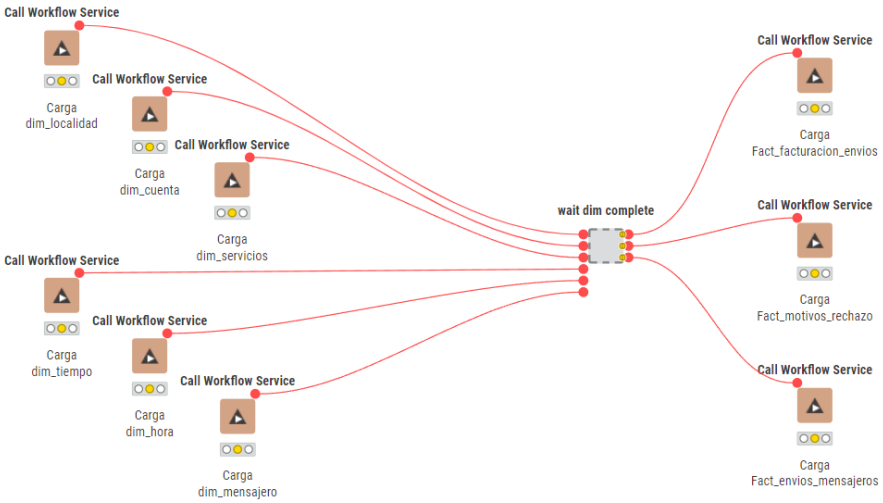

Figura 6.19: *Proceso de carga completa (Elaboración propia)*

Para cumplir con las restricciones de dependencia entre las diferentes tablas, primero se cargan las dimensiones, y finalmente se ingestan los datos de las tablas de hechos. Para ello se ha vuelto a hacer uso de la secuencialidad que ofrecen las variables de los nodos, como se ha mostrado en la Subsección 6.5.1.

6.6.2. 03_weekly_processes

Este proceso se encarga de configurar y ejecutar los procesos necesarios para cumplir con los requerimientos de actualización semanal de las tablas de hechos.

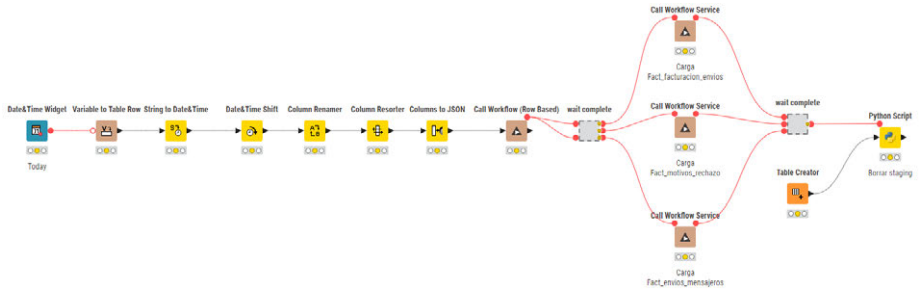

Figura 6.20: *Procesos semanales (Elaboración propia)*

Para ello, el proceso coloca en la zona de staging los ficheros referentes a los envíos relativos a los últimos 7 días. Una vez los datos preprocesados se encuentran listos, se procede a la carga en las tablas de hechos. Finalmente, se eliminan los ficheros de la zona de staging con un script Python.

6.6.3. 03_mothly_processes

Este procesos se encarga de ingestar los datos relativos a las dimensiones cuenta y servicios. Como se ha especificado en la Sección 6.2, únicamente son necesarios los procesos de carga de estas dos dimensiones a nivel mensual.

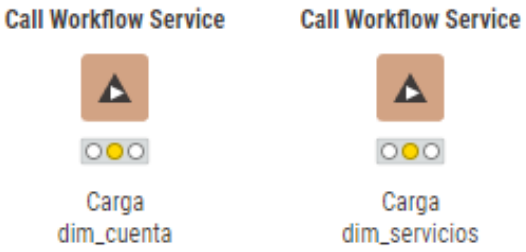

Figura 6.21: *Procesos mensuales (Elaboración propia)*

El resto de procesos que no son incluidos ni en la ejecución semanal, ni en la ejecución mensual son ejecutados manualmente.

Capítulo 7

Visualizaciones

El principal objetivo del sistema es dar soporte a un conjunto de informes que se han desarrollado en el servicio de Power BI. En esta sección se detallarán las visualizaciones requeridas.

Por motivos de confidencialidad no se pueden mostrar los datos reales de la empresa. Por ende, se ha desarrollado un script de Python que permite poblar el Data Warehouse con datos ficticios. Algunas gráficas como consecuencia pueden parecer inservibles o poco entendibles, pero con los datos reales esto no ocurre.

Para un mejor desarrollo de la memoria de este proyecto se ha generado un logo para la empresa ficticia que se ha creado para este proyecto, ver Figura 7.1

Figura 7.1: *Logo Pirnain (Elaborado con Logo.com)*

7.1. Informe general Pirnain

En primer lugar, identificamos la necesidad de disponer de un **informe compuesto por 6 pantallas** o secciones principales, a las que se puede navegar mediante un menú lateral. Las 6 pantallas del informe se enumeran a continuación:

1. Vista General
2. Vista de Ingresos
3. Vista de Servicios
4. Vista de Cuentas
5. Vista de Mensajeros
6. Vista de Incidencias

Las diferentes visualizaciones se detallan a continuación.

7.1.1. Vista General

Pantalla en la que se recoge una visión general de los ingresos y envíos a nivel de franquicia. En esta visualización se recoge información de ventas, envíos e indicadores de rendimiento.

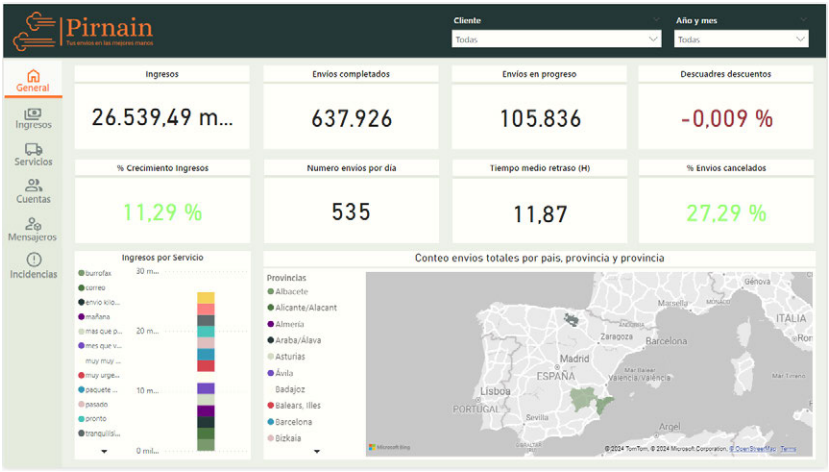

Figura 7.2: *Visualización General Power BI (Elaboración propia)*

El mapa es interactivo, y se puede entrar en detalle de cada comunidad autónoma y provincia, hasta la población. La vista se puede segmentar según el cliente o el año y mes.

7.1.2. Vista de Ingresos

Pantalla que recoge la información relativa a los ingresos de la franquicia. En esta se pueden visualizar la evolución de los ingresos y las devoluciones de la empresa matriz a la franquicia.

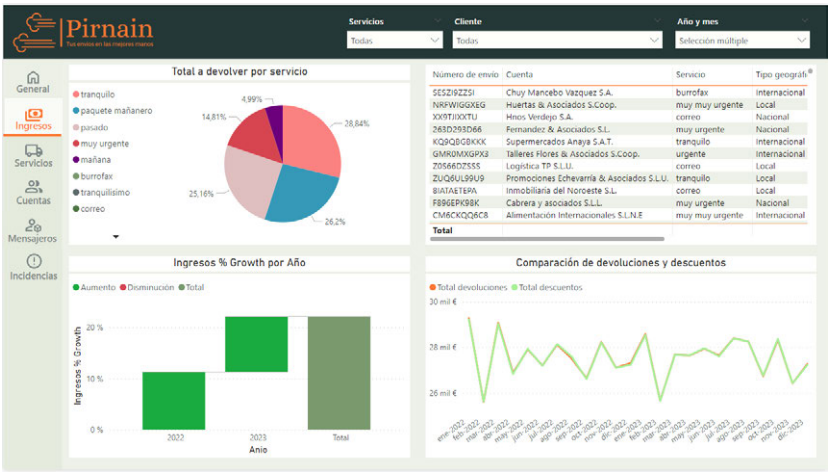

Figura 7.3: *Visualización de Ingresos Power BI nivel de años(Elaboración propia)*

Tanto para la gráfica `Ingresos % Growth por Año` como para la gráfica `Comparación de devoluciones y descuentos` se ha creado una jerarquía de atributos. Esto quiere decir que se pueden visualizar las gráficas a nivel de años o bien a nivel de mes dentro de un año en concreto, ver Figura 7.4.

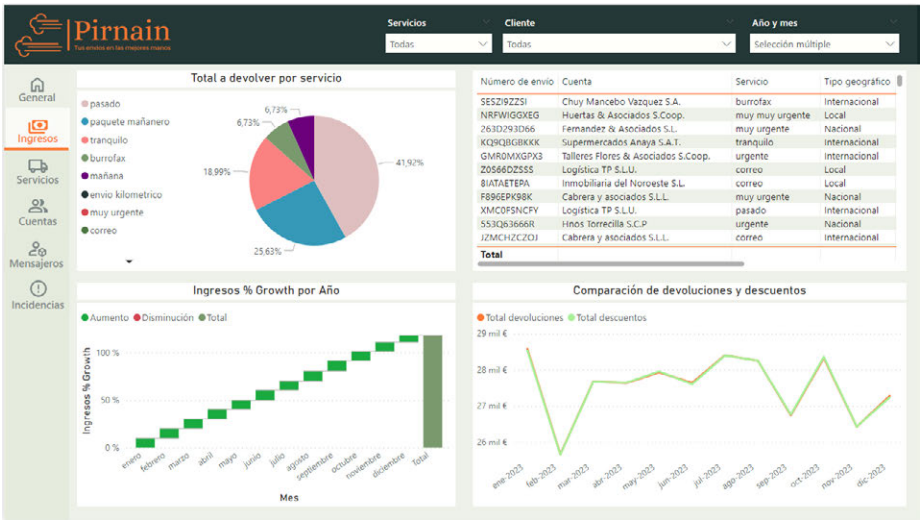

Figura 7.4: *Visualización de Ingresos Power BI en 2023 (Elaboración propia)*

7.1.3. Vista de Servicios

Pantalla en la que se recoge la información relativa a número de envíos, ingresos, retrasos y número de bultos por tipo de servicio.

Figura 7.5: *Visualización de Servicios Power BI (Elaboración propia)*

La visualización puede segmentarse en función del servicio, el cliente o el año y mes que se realizó el envío.

7.1.4. Vista de Cuentas

En esta pantalla se recoge la información relativa al número de envíos e ingresos por cuenta. Mostrando principalmente las cuentas clave, en relación a su importancia para la empresa. Además se añade una visualización en forma de tabla para facilitar la exportación de datos en periodos de facturación.

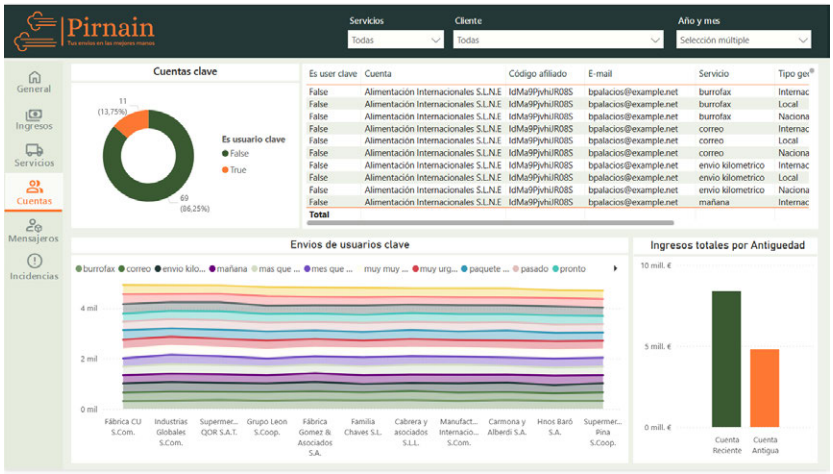

Figura 7.6: *Visualización de Cuentas Power BI (Elaboración propia)*

La visualización, al igual que la anterior, puede segmentarse en función del servicio, el cliente o el año y mes que se realizó el envío.

7.1.5. Vista de Mensajeros

En esta pantalla se recoge la información relativa al número de envíos, bultos y retrasos en función de los mensajeros.

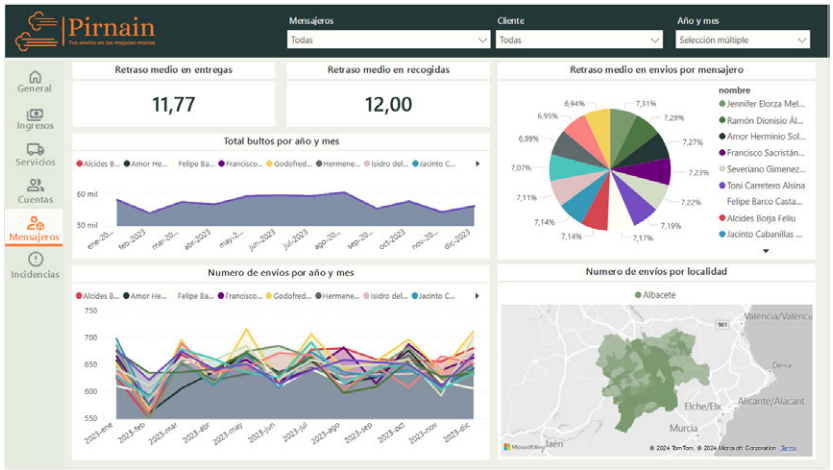

Figura 7.7: *Visualización de Mensajeros Power BI (Elaboración propia)*

7.1.6. Vista de Incidencias

Pantalla que recoge la información relativa al número de envíos por estado y cliente. Esta visualización permite un rápido estudio de la salud general de los envíos, pudiendo identificar puntos críticos en incidencias de forma accesible.

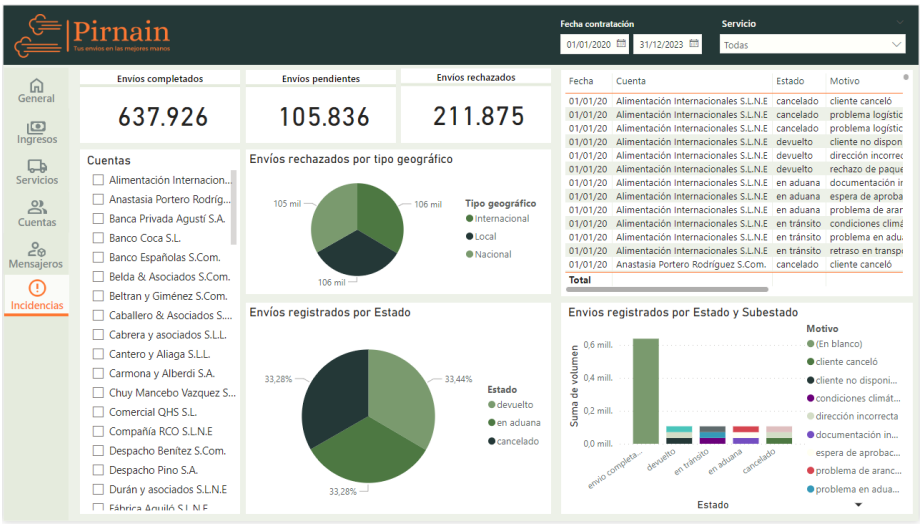

Figura 7.8: *Visualización de Incidencias Power BI (Elaboración propia)*

Esta visualización puede ser segmentada en función de un rango de fechas, una serie de clientes y un conjunto de servicios.Además, la figura `Envíos registrados por Estado y Subestado` es interactiva, es decir, si posas el cursor encima de una columna con datos se muestra una pequeña pantalla con el número de envíos con ese estado y divido por servicio, ver Figura 7.9.

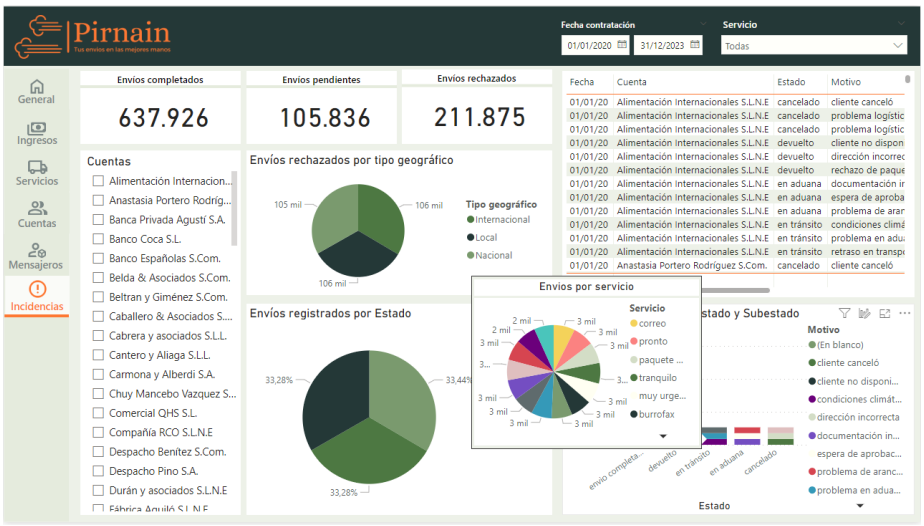

Figura 7.9: *Visualización de Incidencias Power BI con Tool Tip (Elaboración propia)*

Capítulo 8

Validación

En este capítulo se abarca la validación realizada durante el proyecto. El trabajo ha sido desarrollado sobre los objetivos expuestos en la Sección 1.3. Objetivos, por lo que no se realizará una validación a nivel de requisitos, sino a nivel de proceso y visualización.

8.1. Validación de proceso

Para llevar a cabo la validación a nivel de proceso se han seguido los siguientes métodos para cada una de las tablas finales del Data Warehouse.

8.1.1. Dimensiones

8.1.1.1. Dimensión Cuenta

En primer lugar se han contabilizado las filas distintas en el origen y las filas de la dimensión final. Esto quiere decir que, si son iguales, existen tantos clientes en el Data Warehouse como número de clientes tienen documentados los usuarios de negocio de Pirnain.

Una vez se ha realizado esta primera comprobación de manera exitosa, se procede a comprobar que los registros de los usuarios clave sean conformes al origen. Se revisa este segmento de datos debido a su gran importancia para la empresa y reducido número de registros.

Finalmente, se ha generado un listado Excel, el cual ha enviado a los usuarios de negocio para su validación. Una vez los usuarios de negocio han aprobado el listado, se acepta la dimensión.

8.1.1.2. Dimensión Localidad

En primer lugar se han comparado el número de filas distintas en el origen con las filas de la dimensión final, con el fin de asegurar que existe el mismo número de localidades tanto en el origen como en el Data Warehouse.

Tras las primera comprobación, se exponen los datos en una visualización de tipo mapa. De esta manera aseguramos que todos los valores se encuentren su zona geográfica correspondiente.

Adicionalmente, se realiza una consulta sobre la zona principal de operación de la franquicia. Con el objetivo de realizar una comprobación exhaustiva de los valores de provincia, localidad y código postal debido a su criticidad para la empresa.

El resultado de la consulta es validado por los usuarios de negocio, aportando observaciones sobre la escritura habitual del nombre de las localidades en la base de datos operacional, con el fin de normalizar posteriormente los datos.

8.1.1.3. Dimensión Servicio

En primer lugar se han comparado el número de filas distintas en el origen con las filas de la dimensión final. Si la comparación genera el mismo resultado, significa que el número de servicios es correcto.

Tras la primera verificación, se ha realizado una comprobación sobre el listado de servicios de Pirnain, comparando que existan todos los servicios ofrecidos por la empresa junto con su tipo geográfico correspondiente.

8.1.1.4. Dimensión Mensajero

En primer lugar se han contabilizado los registros del origen y los de la dimensión final. Tras esta comprobación podemos asegurar que existen como mínimo el número de mensajeros que debería.

Como segunda comprobación, se ha enviado el listado de mensajeros a los usuarios de negocio de Pirnain para que validen los resultados obtenidos. En el momento que acepten el listado se da como validada la dimensión.

8.1.2. Tablas de hechos

Para validar las tablas de hechos se ha empleado un subconjunto de los datos finales. De esta manera las primeras pruebas de validación son más manejables y rápidas, en comparación con el uso del conjunto entero de datos.

8.1.2.1. Tabla de hechos Facturación

Utilizando el subconjunto de datos se han comparado las filas del origen y de la salida. Además, se han comprobado que las claves foráneas no contengan nulos, los datos sean coherentes entre sí y no existan envíos duplicados.

Tras esta primera comprobación, se ha enviando un subconjunto de datos, filtrados por un cliente en específico, a los usuarios de negocio. Los usuarios deben comparar el importe, descuento y devolución total del cliente.

Si los usuarios aceptan este subconjunto, se da por validado el proceso hasta la implementación del cuadro de mandos referente a estos hechos.

Tras el desarrollo del cuadro de mando, se vuelven a validar los datos sobre el mismo, esta vez de manera global por parte de los usuarios de negocio. Si es aceptada, se da por terminada la validación de la tabla de hechos.

8.1.2.2. Tabla de hechos Mensajeros

En este proceso no se pueden comparar el número de registros de origen y de salida de forma directa. Ya que en esta tabla existen envíos duplicados que representan la entrega y la recogida. Por ende, se debe contabilizar el número de registros de origen con el número de códigos de envío distintos en la tabla de hechos final.

Los datos utilizados en la primera validación están filtrados por un mensajero en concreto. Se realiza una comprobación sobre el número de bultos entregados, las zonas de entrega, y el tiempo de retraso. Tras aceptar los datos se da por validado hasta la implementación del cuadro de mandos.

Una vez está desarrollado el cuadro de mando, se vuelven a validar los datos, esta vez para todos los mensajeros, por parte de los usuarios de negocio. Una vez se ha comprobado la validez de los datos, se acepta el proceso.

8.1.2.3. Tabla de hechos Motivos Rechazo

En primer lugar se valida el número de envíos contabilizando los registros en el origen y, posteriormente, calculando la suma de la columna volumen de la tabla de hechos. Si los dos datos coinciden, significa que el proceso produce el número de registros adecuado.

Para validar este proceso se han filtrado los datos según varias combinaciones de los campos pertenecientes a la agrupación. De tal manera que se comprueba por lo menos un registro para cada estado, con el campo volumen igual a 1 ó mayor que 1.

Finalmente, tras implementar el cuadro de mando, se somete a una nueva validación por parte de los usuarios de negocio. Quienes, gracias a la visualización, pueden comparar los datos finales los propios de origen y comprobar su validez. Una vez los usuarios finales han aprobado los datos, se toma por terminado el proceso.

8.2. Validación de visualizaciones

Con el objetivo de crear una serie de visualizaciones funcionales para los usuarios de negocio se ha sometido a un proceso de validación a todos los informes creados en Power BI.

La validación se lleva a cabo a nivel de página del informe. Cuando finalmente se validan las 6 páginas expuestas en el Capítulo 7. Visualizaciones, se acepta el informe completo y se toma por terminado.

Se ha realizado el mismo proceso de validación para todos los informes de manera individual. Primero se ha realizado, en base a las necesidades aparentes de los usuarios finales, una primera versión con diferentes medidas y figuras.

Una vez desarrollado el informe, se otorgó acceso a los usuarios de negocio. En este momento los usuarios generaron una serie de propuestas de mejora, las cuales fueron implementadas.

De esta manera, se vuelve a enviar el informe a los usuarios de negocio, los cuales tienen como responsabilidad aprobar la visualización. Si no se acepta la visualización se vuelve a repetir el proceso, hasta que el informe sea validado correctamente.

Capítulo 9

Conclusiones

En esta sección se presentan las conclusiones finales del proyecto desarrollado. Exponiendo los resultados obtenidos, los objetivos logrados y los problemas encontrados durante el desarrollo.

9.1. Resultados

9.1.1. Resultados obtenidos

El principal resultado del proyecto es la generación del informe general de Pirnain en Power BI. Para alcanzar este resultado final, se han tenido que completar diferentes fases y resultados. A continuación se enumeran los principales resultados del proyecto:

1. **Análisis de los datos de franquicia:** Uno de los resultados más útiles para el propio desarrollo del proyecto ha sido el análisis de las distintas fuentes de datos de Pirnain. Gracias a su documentación, actualmente se tiene acceso a los diferentes datos de forma controlada.

2. **Descripción detallada del dominio de negocio**: Uno de los principales problemas a enfrentar fue la compleja lógica de negocio. Tras varias reuniones con los usuarios de negocio se pudo documentar el dominio, y gracias a esto se pudo iniciar el proyecto.

3. **Arquitectura del sistema:** El proyecto ha dado como resultado el diseño e implementación de una arquitectura completa de Business Intelligence. Esta arquitectura documenta entre otros: las fuentes de datos, tasa de refresco, planificación de procesos, diseño del Data Warehouse, etc.

4. **Procesos ETL:** Como resultado del desarrollo del sistema, se han implementado una serie de procesos ETL los cuales permiten la integración del nuevo Data Warehouse al sistema operacional. Estos procesos han sido desplegados en un servidor privado, los cuales son ejecutados según una agenda expuesta anteriormente.

5. **Data Warehouse:** El proyecto tiene como resultado el diseño e implementación de un Data Warehouse dedicado a la franquicia de Pirnain. Este diseño ha sido refinado a lo largo del proceso, y ha sido desplegado en el mismo servidor privado en el que se encuentran los procesos ETL.

6. **Visualizaciones de datos:** Este es el resultado visible para los usuarios de negocio. Con el fin de explotar los datos reunidos en el Data Warehouse se ha desarrollado un informe en Power BI, el cual permite a los usuarios finales acceder a los datos requeridos de forma rápida y sencilla.

7. **Solución de Business Intelligence:** Como resultado general del proyecto se ha diseñado e implementado una estructura de BI completa para el caso de Pirnain. Esta ha sido desplegada, y actualmente se encuentra en uso por parte de los usuarios de negocio.

9.1.2. Objetivos logrados

En esta sección se especifican los objetivos logrados. En primer lugar se enumeran los objetivos específicos:

1. **Análisis de la estructura y datos operacionales:** El objetivo se ha cumplido con éxito. Como resultado se ha obtenido la documentación necesaria para entender la estructura y datos operacionales. Se puede constatar en la arquitectura del mismo proyecto, y en su correcto tratamiento de los datos.

2. **Describir el dominio del negocio:** Se ha completado el objetivo, y como resultado del mismo se tiene la descripción del dominio de negocio en la Sección 2.1.

3. **Creación de un Data Warehouse para la franquicia:** Se ha alcanzado exitosamente el objetivo. Primero se realizó el diseño del mismo, como puede observarse en el Capítulo 5. Para posteriormente implementarlo y desplegarlo en un servidor MySQL.

4. **Creación de procesos ETL:** El objetivo ha sido completado con éxito. Con el fin de poblar el Data Warehouse se han implementado una serie de procesos ETL, expuestos en el Capítulo 6. Los cuales han sido correctamente validados por los usuarios de negocio.

5. **Planificación del refresco de datos:** Se ha cumplido con el objetivo. Como puede observarse en el Capítulo 6 se ha diseñado e implementando la planificación de los diferentes procesos ETL. Como resultado, los datos referentes a los envíos son actualizados conforme a los requerimientos.

6. **Creación de múltiples visualizaciones para los usuarios de negocio:** Se ha completado el objetivo de manera exitosa. Como se han mostrado en el Capítulo 7, se han desarrollado las visualizaciones requeridas por los usuarios de negocio, englobadas en un único informe en Power BI.

Como se ha expuesto en los resultados, se ha diseñado e implementado una solución de Business Intelligence completa para la empresa Pirnain. De esta manera, junto con el cumplimiento de todos los objetivos específicos, se puede afirmar que el objetivo principal ha sido completado exitosamente.

9.1.3. Problemas encontrados

Se han encontrado diversos problemas a lo largo del desarrollo del proyecto. Los cuales han tenido diferentes y variadas naturalezas y soluciones. A continuación se exponen los principales problemas enfrentados durante el proyecto:

1. **Complejidad de negocio:** El primer y gran problema surgió al inicio del proyecto. En este aspecto se juntaron diversos factores: falta de documentación, desconocimiento de los propios usuarios de negocio, y conceptos y dominio complejos. Ante tales dificultades, realizar el análisis del sistema operacional fue una tarea ardua, así como, el posterior diseño del Data Warehouse y métricas. Tras varias reuniones, documentos desactualizados e ingeniería inversa se pudo realizar una documentación del dominio de negocio y la estructura de los datos operacionales.

2. **Falta de organización de datos:** Al tener que definir las diferentes fuentes de datos nos dimos cuenta de la gran cantidad de información útil que se encontraba desperdigada. Algunos ficheros con datos de interés podían encontrarse en correos electrónicos archivados, carpetas en local aleatorias, o en los mismos recuerdos de los usuarios de negocio.

3. **Falta de accesibilidad a los datos:** Cuando se realizó el estudio de la estructura operacional, se llegó a la conclusión de que realizar cualquier análisis sobre los datos obtenidos en los diferentes portales de Pirnain era imposible. Muchos datos no eran coherentes entre diferentes informes, existían columnas con un mismo nombre, pero que representaban conceptos distintos, la descarga de informes era lenta, entre otros varios inconvenientes más.

9.2. Lineas futuras

El proyecto ha abarcado las necesidades más críticas que Pirnain requería. Aun así, por la gran complejidad de negocio de la empresa se pueden desarrollar varias extensiones del mismo. Entre otras lineas futuras se plantean:

1. **Estudio de direcciones de mensajeros:** Pirnain realiza un conteo de envíos diferente para los mensajeros contratados. Para poder implementarlo en el proyecto, se debe realizar un estudio de la lógica de negocio y, posteriormente, diseñar e implementar los procesos, junto con las modificaciones necesarias. Esto puede parecer trivial, teniendo en cuenta el trabajo desarrollado. Pero tras un estudio inicial de los datos y lógica referentes a este área, se ha optado por mantenerlo como una línea futura.

2. **Sistema de recomendación de clientes clave:** Actualmente Pirnain ofrece descuentos a sus clientes, basándose en su intuición. Actualmente se ha desarrollado una herramienta que permite ver la actividad de las diferentes cuentas, pero siguen sin conocer la viabilidad de los usuarios clave. Por ende, se propone una solución que permita clasificar a los diferentes usuarios en función de sus características, estudiando posibles ofertas que mantengan la viabilidad tanto de antiguos como de nuevos usuarios clave.

3. **Sistema de ayuda de facturación:** Como consecuencia de las incoherencias de los datos del sistema, la facturación no puede ser realizada automáticamente. Como resultado, un usuario de negocio tiene que revisar cliente por cliente que los envíos realizados estén correctamente

definidos y tengan sentido según el usuario de negocio. Se propone realizar una solución que permita visualizar los datos de todos los clientes con el formato adecuado para la facturación, así como corregir los datos incoherentes según X reglas definidas por los usuarios de negocio mes a mes.

4. **Sistema de predicción de envíos:** Se propone desarrollar un sistema de predicciones de envíos para los clientes de Pirnain. Conociendo el sector al que pertenece el cliente, junto con su tendencia de envíos, se cree provechoso la realización de un sistema de recomendaciones de potenciales clientes clave.

9.3. Impacto social y medioambiental

El impacto medioambiental del presente Trabajo de Fin de Grado puede considerarse como nulo, dado el coste computacional resultante no integra máquinas de alto consumo. Además, el sistema está desplegado sobre un servidor privado, el cual ya estaba operativo anteriormente, por lo que no se deben adquirir nuevas máquinas para este proyecto.

En lo que respecta al impacto social, el proyecto desarrollado en este Trabajo de Fin de Grado sirve de gran ayuda para los usuarios de Pirnain. Conozco personalmente a los usuarios de negocio desde hace varios años, y los problemas a los que se enfrentan semanalmente. Este fue el gran aliciente para iniciar el presente proyecto, y con ello me complace haber mejorado la calidad de trabajo de conocidos mios.

Para concluir, durante todo el trabajo se han utilizado una empresa y datos ficticios, por lo que no se pone en riesgo ningún tipo de información sensible de la empresa real.

Bibliografía

[1] StrateBI, "Manual de introduccion a knime," https://todobi.com/introduccion-a-kine/, accedido: 24 de abril de 2024.

[2] Python, "Applications for python — python.org," https://www.python.org/about/apps/, accedido: 24 de abril de 2024.

[3] L. D. Solutions, "¿qué es knime?" https://www.lisdatasolutions.com/es/que-es-knime/, accedido: 24 de abril de 2024.

[4] J. de Castilla y León, "A2.4.1.excel como herramienta de evaluación nivel a2 web," https://www.educa.jcyl.es/educacyl/cm/gallery/CCD/Area_4/A2.4_Excel_como_herramienta_de_evaluacion/1_qu_es_excel_aplicaciones.html, accedido: 24 de abril de 2024.

[5] Proximahost, "Qué es cron y cómo funciona el administrador de tareas de linux," https://proximahost.es/blog/cron-administrador-tareas-linux/, accedido: 24 de abril de 2024.

[6] Dinahosting, "¿qué es cron y para qué sirve?" https://dinahosting.com/ayuda/como-configurar-tareas-cron-de-forma-manual/, accedido: 24 de abril de 2024.

[7] MySQL, "Qué es mysql: Características y ventajas — openwebinars," https://openwebinars.net/blog/que-es-mysql/, accedido: 24 de abril de 2024.

[8] Microsoft, "¿qué es power bi y cuáles son sus características? - xms," https://www.xmslatam.com/que-es-power-bi-y-cuales-son-caracteristicas/, accedido: 24 de abril de 2024.

[9] Logo.com, "Free logo maker," https://logo.com/, accedido: 24 de abril de 2024.

[10] S. Overflow, "Stack overflow," https://stackoverflow.com/, accedido: 24 de abril de 2024.

[11] P. P. Alarcón Cavero, "Tema 2. data warehouse. sistemas de soporte para la toma de decisiones," Diapositivas de PowerPoint, Universidad Politécnica de Madrid, 2023.

[12] ——, "Tema 3. procesos etl. sistemas de soporte para la toma de decisiones," Diapositivas de PowerPoint, Universidad Politécnica de Madrid, 2023.

[13] KNIME, "Knime community forum," https://forum.knime.com/latest, accedido: 24 de mayo de 2024.

[14] ——, "Knime best practices guide," https://docs.knime.com/latest/analytics_platform_best_practices_guide/index.html#_what_is_knime_software, accedido: 24 de abril de 2024.

[15] ——, “Knime call from cmd,” https://forum.knime.com/t/how-can-i-execute-a-knime-workflow-from-windows-cmd-when-knime-is-not-actually-installed-because-i-always-start-it-with-windows-cmd/62017, accedido: 24 de abril de 2024.

[16] ——, “Knime python api — knime python api documentation,” https://knime-python.readthedocs.io/en/stable/, accedido: 24 de mayo de 2024.

[17] Microsoft, “Referencia de visual basic para aplicaciones (vba) para office,” https://learn.microsoft.com/es-es/office/vba/api/overview/, accedido: 24 de mayo de 2024.