Facultad de Ciencias

Aplicación de Procesos de Ciencia de Datos para la Recopilación, Procesamiento y Análisis de Datos de una ONG

(Application of Data Science Processes for the Collection, Processing and Analysis of Data of a NGO)

Trabajo de Fin de Máster para acceder al

Máster en Ciencia de Datos

Autor: Laura Mónica Alcantar Martínez
Director: Diego García Saiz

Resumen

La transformación digital en una Organización No Gubernamental consiste en digitalizar sus procesos de trabajo mientras se adopta un cambio cultural dentro de la organización, y del mismo modo, la generación de valor agregado a través de los datos es una de las tecnologías clave para el diseño y la operación de sus procesos y servicios.

En este Trabajo de Fin de Máster se aplican diferentes elementos de la ciencia de datos que suponen la generación de una estructura digital para el almacenamiento, procesamiento y análisis de los datos. Estos procesos van desde el diseño e implementación de una base de datos para digitalizar y centralizar los datos de la organización y el análisis de los mismos utilizando herramientas de análisis y visualización como lo son los tableros e informes con la herramienta de Micrisoft Power BI.

Palabras clave: Base de Datos, Procesos ETL, Análisis de Datos, KPI, Power BI, ONG.

Abstract

Digital transformation in a Non Governmental Organization consists of digitizing its work processes while adopting a cultural change within the organization, and in the same way, the generation of added value through data is one of the key technologies for design and operation of its processes and services.

In this Master's Thesis, different elements of data science are applied that involve the generation of a digital structure for the storage, processing and analysis of data. These processes range from the design and implementation of a database to digitize and centralize the organization's data and its analysis using analysis and visualization tools such as dashboards and reports with the Micrisoft Power BI tool.

Palabras clave: Date Base, ETL Processes, Data Analysis, KPI, Power BI, NOG.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Contexto

Este trabajo de fin de máster se lleva a cabo en colaboración con la Empresa LIS Data Solutions[1] y la asociación ATTsF[2] (Siglas de: Asociación de Trabajadores y Técnicos sin Fronteras) en los meses de julio a septiembre del 2023.

LIS Data Solutions es una empresa tecnológica fundada en el año 2013, brindando servicios de consultoría con la idea de ser más rentables y sostenibles a las empresas mediante la implantación de tecnología puntera y emergente, y el uso de big data e inteligencia artificial para la mejora de procesos. Sus servicios se brindan también a través de herramientas de *Bussiness Intelligence* y *Data Science* para ayudar a las corporaciones a conocerse mejor, mejorar sus procesos y resultados.

Por otro lado, ATTsF, una organización no gubernamental (ONG), es una entidad sin ánimos de lucro, la cual ofrece soluciones técnicas para diferentes problemáticas que presentan países en vías de desarrollo.

Dentro del contenido de esta memoria y como parte del proyecto que se desarrolla, LIS Data Solutions ofrece servicios de transformación, almacenamiento y analítica de datos, para ayudar a mejorar los procesos y la toma de decisiones de la organización ATTsF.

[1] https://www.lisdatasolutions.com/es/

[2] https://www.attsf.org/

1.2. Motivación y Objetivos

ATTsF actualmente cuenta con un ERP (por sus sigla en inglés, *Enterprise Resource Planning*) diseñado e implementado a medida por su equipo de tecnologías de información, que recopila datos de dos de los principales procesos de la organización, por un lado el procesos de la distribución de alimentos de la canasta básica, y por otro lado el mantenimiento mecánico de los vehículos que utilizan para hacer estas distribuciones, se describen los detalles de estos procesos en la sección 2.1.1. Sin embargo, muchos de los datos que recopilan son capturados en ficheros excel, lo que da paso a que la calidad de estos datos no sea buena e incluso que se pierda información. Además, no se cuenta con una estrategia sólida de análisis de la información para favorecer los procesos ni la toma de decisiones.

Objetivos generales

Como objetivo general es implementar una arquitectura tecnológica que permita estandarizar la manera en la que se capturan y almacenan los datos para poder acceder a ellos, hacer consultas y principalmente analizarlos para favorecer la toma de decisiones de la organización.

Objetivos específicos

- Evitar la pérdida de información.
- Mejorar la calidad de los datos que se recopilan.
- Centralizar la recopilación de datos en una sola fuente de origen.
- Establecer los principales indicadores clave (KPIs por sus siglas en ingles, *Key Performance Indicators*) y detectar puntos críticos en los procesos
- Mejorar la toma de decisiones a partir del análisis de sus datos.

1.3. Propuesta y diseño de arquitectura

Por el hecho de que se tienen distintos orígenes de datos, se propone el diseño y la implementación de la una base de datos para centralizar y almacenar la información, y de este modo poder hacer consultas y que esta sea el nuevo origen para el análisis de datos y generación de tableros con la herramienta de *Business Intelligence* de Microsoft Power BI.

En la figura 1.1 se muestra el diseño de la arquitectura propuesta, la cual muestra los diferentes orígenes de datos, tanto los ficheros excel como el ERP de la organización. Los datos se procesan a través de procesos ETL (Extraer, Transformar, y Cargar, por sus siglas en inglés: *Extract, Transform, and Load*) para centralizarce en una base de datos alojada en SQL Server, la cual tiene como finalidad ser el *Data Mart* de la arquitectura, para finalmente alimentar el origen de datos en Power BI de Microsoft y poder analizar y visualizar la información.

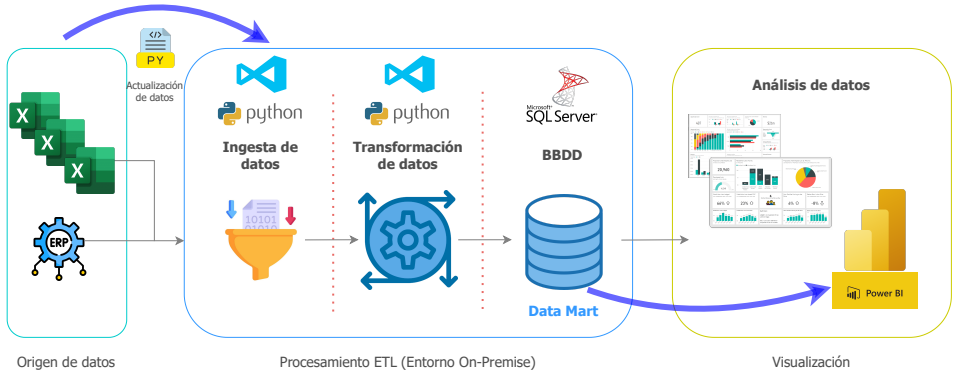


Figura 1.1: Diseño de arquitectura

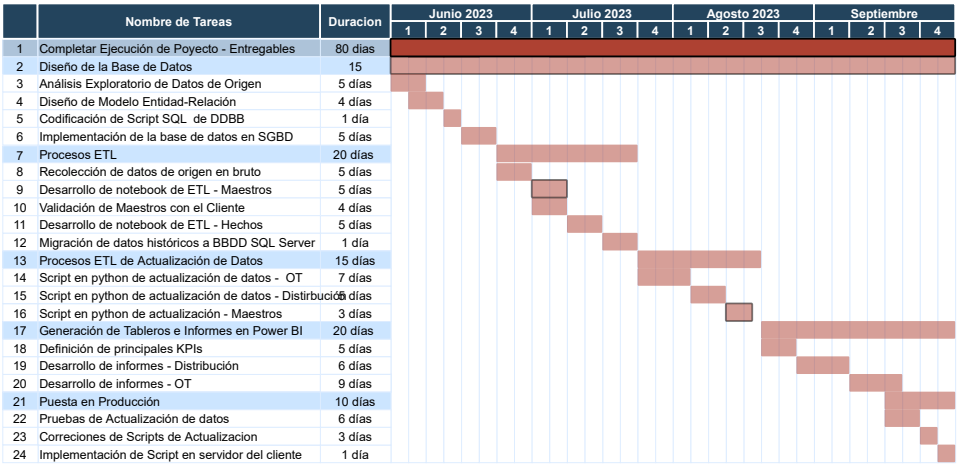
1.4. Entregables

Como resultado de la finalización del proyecto se obtiene los siguientes dos entregables que se enlistan a continuación:

- Script en Python de actualización de datos: este script consiste en mantener actualizados los datos centralizados en la base de datos desarrollada, es de ejecución semanal y asegura la calidad de los nuevos datos y el resguardo de los mismos.
- Herramienta de análisis y visualización de los datos: esta herramienta consiste en el desarrollo de tableros e informes que puede ser consultada por los usuarios encargados de la toma de decisiones en la organización.

1.5. Desarrollo temporal

El proyecto se desarrolla en cinco hitos principales, el diseño de la base de datos, la generación de procesos ETL, por un lado de los datos históricos y por otro lado los de la actualización de los datos, la generación de tableros e informes para el análisis de los datos y la puesta en producción de entregables del proyecto. En la figura 1.2 se muestra el Diagrama de Gantt simplificado del desarrollo temporal durante el cual se llevaron a cabo cada una de la tareas que conforman estos hitos.



	Nombre de Tareas	Duracion
1	Completar Ejecución de Poyecto - Entregables	80 dias
2	Diseño de la Base de Datos	15
3	Análisis Exploratorio de Datos de Origen	5 días
4	Diseño de Modelo Entidad-Relación	4 días
5	Codificación de Script SQL de DDBB	1 día
6	Implementación de la base de datos en SGBD	5 días
7	Procesos ETL	20 días
8	Recolección de datos de origen en bruto	5 días
9	Desarrollo de notebook de ETL - Maestros	5 días
10	Validación de Maestros con el Cliente	4 días
11	Desarrollo de notebook de ETL - Hechos	5 días
12	Migración de datos históricos a BBDD SQL Server	1 día
13	Procesos ETL de Actualización de Datos	15 días
14	Script en python de actualización de datos - OT	7 días
15	Script en python de actualización de datos - Distirbución	5 días
16	Script en python de actualización - Maestros	3 días
17	Generación de Tableros e Informes en Power BI	20 días
18	Definición de principales KPIs	5 días
19	Desarrollo de informes - Distribución	6 días
20	Desarrollo de informes - OT	9 días
21	Puesta en Producción	10 días
22	Pruebas de Actualización de datos	6 días
23	Correciones de Scripts de Actualizacion	3 días
24	Implementación de Script en servidor del cliente	1 día

Figura 1.2: Diagrama de Gantt del desarrollo temporal del proyecto

1.6. Software requerido

Python: es un lenguaje de programación creado por Guido van Rossum a principios de los años 90. Se trata de un lenguaje interpretado o de *script*, con tipado dinámico, multiplataforma y orientado a objetos, además , tiene una sintaxis muy limpia y que favorece un código legible [González Duque, 2011].

Visual Studio Code: Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es un *software* libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. Tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sin número de extensiones, que da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación [Flores, 2023].

SQL Server: Microsoft SQL Server es un sistema de gestión de base

de datos relacional producido por Microsoft. Su principal lenguaje de consulta es Transact-SQL, una aplicación de las normas ANSI / ISO estándar *Structured Query Language (SQL)* utilizado por ambas Microsoft y Sybase [Santamaría and Hernández, 2016].

Dbeaver: es una herramienta gráfica de administración de bases de datos *open source*. Dbeaver puede ser utilizado para generar y administrar bases de datos en una amplia gama de disposiciones ya que trabaja con los sistemas gestores de bases de datos (DBMS, por sus siglas en inglés) más comunes, tales como MySQL, PostgreSQL, MariaDB, SQLite, Oracle, DB2, SQL Server, Sybase, Microsoft Access, Teradata, Firebird, Derby, y más [Anbumani et al., 2021].

SQLAchemy: SQLAlchemy SQL Toolkit y *Object Relational Mapper (ORM)* es un conjunto completo de herramientas para trabajar con bases de datos y Python. Tiene varias áreas distintas de funcionalidad que se pueden utilizar individualmente o combinadas [Bayer, 2010].

Las dos partes frontales más importantes de SQLAlchemy son el *Object Relational Mapper (ORM)* y el Lenguaje de Expresión SQL. Las expresiones SQL se pueden utilizar independientemente del ORM. Al utilizar el ORM, el lenguaje de expresión SQL sigue siendo parte de la API pública, ya que se utiliza dentro de configuraciones relacionales de objetos y consultas.

Power BI: es una herramienta para el análisis de datos basada en la nube y trabaja con una amplia gama de fuentes de dato. Esta herramienta puede ser utilizada para el análisis de datos y generación de informes o tableros. Power BI es amigable y puede ser utilizada de forma simple por el usuario, además es un software maduro y poderoso que pude ser utilizado por desarrolladores de inteligencia de negocio en sistemas de empresas para modelar escenarios y hacer la combinaciones de datos complejos [Gowthami and Kumar, 2017].

Algunos de los beneficios por los cuales se ha decidido implementar Power BI como solución de software para el análisis de datos de la organización son:

- Se pueden generar reportes y tableros para soluciones SaaS (*Software as a Service*).
- Actualización en tiempo real de tableros.
- Conexión con diversas fuentes de datos, ya sea *on-premise* o *cloud*.
- Seguridad, conexión híbrida, rápida implementación e integración con sistemas existentes de tecnologías de información.

Capítulo 2

Diseño de la base de datos

El diseño de una base de datos puede definirse como el proceso de capturar la información relevante y los requisitos de procesamiento de una empresa y asignarlos a un sistema de gestión de base de datos subyacente [Storey, 1991].

El tipo de base de datos que se desarrolla en este proyecto es de tipo relacional. Las bases de datos relacionales almacenan los datos de forma estructurada y normalizada, permitiendo definir restricciones y mecanismos que garanticen la consistencia, integridad y seguridad en el acceso, en la actualidad son de las bases de datos más populares [Sánchez, 2004].

Una de las principales características de las bases de datos de tipo relacional es que en sus transacciones se debe cumplir con las propiedades ACID (por sus siglas en ingles, *Atomicity, Consistency, Isolation y Durability*), pues ofrecen garantías en cuanto a [Antiñanco, 2014]:

- Atomicidad: todas las operaciones en la transacción serán completadas o ninguna lo será.
- Consistencia: la base de datos estará en un estado válido tanto al inicio como al final de la transacción.
- Aislamiento: la transacción se comportará como si fuera la única operación llevada a cabo sobre la base de datos (una operación no puede afectar a otras).
- Durabilidad: una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

2.1. Análisis de requisitos

Entre las principales fases del diseño de una base de datos se encuentra la fase del análisis de requisitos, en la cual se realiza un análisis de las necesidades de información dentro de la organización que da como resultado una especificación preliminar de las necesidades de información de los distintos usuarios, lo que hace importante conocer y analizar los procesos de la organización [Storey, 1991].

2.1.1. Descripción de procesos

ATTsF ofrece diversos servicios técnicos y logísticos en las regiones del Sahara, sin embargo este proyecto se enfoca en dos principales procesos, procesos de distribución y procesos de ordenes de trabajo, los cuales se describen a continuación.

Procesos de distribución Consisten en distribuir principalmente alimentos de la canasta básica, aunque pueden ser otro tipo de productos, desde un centro de origen hacia diferentes sitios en la región del Sahara llamadas Wilayas. Se utilizan como medio de transporte diferentes camiones que fueron donados para este fin. Por cada uno de los viajes realizados por cada camión se registra una orden de distribución en un libro de Excel que registra información cómo: el camión que hace la distribución, el conductor, la fecha y hora de salida y llegada, la wilaya a la que viaja, los kilómetros recorridos, las toneladas distribuidas, el tipo de producto que se distribuye, por mencionar la información más relevante.

A los camiones que hacen estas distribuciones se les conoce como Camiones de la flota de alimentos.

Ordenes de trabajo Antes de describir los procesos de ordenes de trabajo, es importante mencionar que así como hay camiones de la flota de alimentos hay camiones de la flota de agua, estos segundo son camiones que están equipados con cisternas que distribuyen agua a través de las diferentes Wilayas, sin embargo, la información de este proceso de distribución esta fuera del alcance de este proyecto.

Por otro lado se llevan registros de las órdenes de trabajo mecánico realizadas a todos los camiones, los de la flota de alimentos y la flota de agua, tanto de tipo correctivo como de tipo preventivo. Los datos de ordenes de trabajo para los camiones pertenecientes a la flota de agua se registran en un libro de Excel.

En el caso de las órdenes de trabajo correctivas se guarda información acerca de los camiones que han sufrido una avería (teniendo en cuenta qué avería ha tenido el camión, pueden ser varias), el día que se ha averiado dicho camión y hasta qué día ha estado dicho camión sin estar disponible, además de guardar los datos del mecánico que ha solucionado la avería y el taller dónde se ha realizado dicha reparación.

En cambio, en las órdenes preventivas se guardan los datos acerca de los repuestos que se les hacen a los camiones (y qué tipo de repuesto), cada una con cierta frecuencia. Al igual que en las ordenes correctivas, en las preventivas también se guardan los datos de fecha de inicio y fin de la orden.

Por otro lado los registros de mantenimientos, ya sea correctivos o preventivos para los camiones pertenecientes a la flota de alimentos, se almacenan directamente en el ERP de la organización, esto solo a partir de la información del año 2023, teniendo ordenes administrativas que recopilan la información antes mencionada de las ordenes de trabajo y por otro lado las ordenes operativas, las cuales recopilan información más detallada de los materiales sobre repuestos y averías, incluyendo sus precios.

2.2. Diseño conceptual

En la fase de diseño conceptual se modela y representa los puntos de vista de los usuarios y las aplicaciones sobre la información y, posiblemente, una especificación del procesamiento o el uso de la información. El objetivo de esta fase es producir una representación de alto nivel de los requisitos independientemente del sistema de gestión de bases de datos que se utilice. Esta representación de alto nivel se denomina esquema conceptual o diseño conceptual el cual es comúnmente representado como un modelo de entidad relación (ME-R).

Para comenzar con el diseño conceptual de la base de datos, se han tenido dos consideraciones generales, la primera se trata de separar conceptualmente el tipo de entidades o tablas de las que se compondrá la base de datos. Por un lado, se tiene lo que a partir de este punto se llamarán las tablas de **"Hechos"** , que corresponden a la información que se recopila de cada orden de distribución y cada orden de trabajo respectivamente, y por otro lado, las tablas **"Maestros"** las cuales se han conceptualizado de esta manera, ya que son tablas a las cuales no se añadirán registros de manera tan frecuente como se haría para las tablas de Hechos. Además, se considera que la principal función las tablas Maestros es clasificar y asignar una clave

única a algunas de las diferentes entidades que conforman la base de datos, esto quedará más claro conforme se va avanzando en el diseño de la base de datos.

Adicionalmente, el diseño conceptual se ha hecho en base a los componentes de un modelo Entidad - Relación, los cuales se detallan en la siguiente sección.

2.3. Modelo Entidad - Relación

El Modelo Entidad - Relación (ME-R) se basa en dos construcciones principales, entidades y sus atributos asociados [Storey, 1991]. Permite representar las entidades relevantes de un sistema de información, así como, las interrelaciones y propiedades [ADRIAN,].

2.3.1. Entidad

Una entidad es un "objeto" de interés en una base de datos; por ejemplo, un empleado. Una entidad es representada como una tabla en una base de datos relacional [Storey, 1991].

En el Modelo Entidad - Relación de la base de datos de este proyecto se tienen catorce entidades, dos corresponden a las tablas de Hechos y el resto a las tablas Maestros.

2.3.2. Atributos y campos

Los atributos o campos son características o propiedades que pueden identificarse tanto en las entidades como en las relaciones. Por ejemplo, "cargo", podría ser un atributo de Empleado y "duración de la asignación" un atributo de la relación anterior. Cada tipo de entidad desempeña un papel en particular en una relación: por ejemplo, los viajeros hacen reservas; las reservas no hacen viajeros[Storey, 1991].

En el modelo relacional, habrá al menos un atributo (o conjunto de atributos) que identifiquen de forma única a cada instancia de una tabla. A este campo se llama ***Primary Key***. A su vez, este campo sirve para que otras tablas lo referencien. Este campo referenciado se conoce como ***Foreign Key*** [Sánchez, 2004].

2.3.3. Relaciones

Una relación es una asociación entre entidades; por ejemplo, empleados asignados a proyectos es una asociación entre los tipos de entidad Empleado y Proyecto [Storey, 1991].

La relaciones en un modelo relacional definen como se relacionan las diferentes entidades en la base de datos y pueden ser de tres tipos [Sánchez, 2004]:

- Uno a uno (1:1).
- Uno a varios (1:N).
- Varios a varios (N:N).

En esta base de datos, solo se dan el tipo de relaciones de uno a varios (1:N) y de varios a varios (N:N).

2.4. Definición de las entidades y sus relaciones

De acuerdo con la descripción de los procesos del negocio en la sección 2.1.1, se definen a detalle algunas de las entidades que conforman parte del diseño de la base de datos, con el fin de esclarecer algunas de las decisiones tomadas para generar el diseño.

2.4.1. Puntos geográficos de distribución (tbl_wilayas)

La entidad tbl_wilaya, es una de las más sencillas en estructura, se refiere a la tabla que almacena cada uno de los datos que corresponden a las diferentes regiones por donde se llevan a cabo las distribuciones de alimentos, donde se encuentran los talleres mecánicos en los cuales se hacen los mantenimientos de los camiones y a su vez a que sitio pertenece un camión, por lo que esta entidad estará a su vez relacionada con otras más entidades. La figura 2.1 muestra como se define esta entidad.

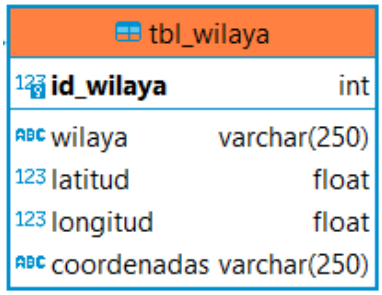


Figura 2.1: Entidad: tbl_wilaya

Sus campos son los siguientes:

Nombre del Campo	Apreciaciones
id_wilaya	Identificador único (Primary Key)
wilaya	Nombre de la región
latitud	Latitud (Medida en grados)
longitud	Longitud (Medida en grados)
coordenads	Latitud y longitud (Medida en grados)

Tabla 2.1: Campos de tbl_wilaya

- Campo *id_wilaya*

  Este capo corresponde a la ***Primary Key*** que se asignará a cada una de las instancias de la entidad. Este campo es un número entero, por lo que se define como tipo de dato int.

- Campo *wilaya*

  Este campo corresponde al nombre que se le da a la región, es de tipo texto y el tipo de datos se define como varchar(250).

- Campo *latitud* y *longitud*

  Estos campos almacenan la latitud y longitud respectivamente en tipo de dato float. Además se decide almacenar por separado teniendo en cuenta el requerimiento futuro de localizaciones geográficas.

- Campo *coordenadas*

Este campo comprende la latitud y la longitud de la ubicación geográfica de la wilaya. Este dato se ha asignado como formato de texto, varchar(250), ya que se tiene en cuenta los diferentes formatos de entrada de datos de ubicación geográfica que pudieran tener diferentes sistemas de visualización.

2.4.2. Camiones (tbl_camiones)

Esta es una de las entidades más potenciales para el modelo, ya que es la entidad de la cual parten los dos principales procesos que se describen de la organización, en primer lugar porque es el primer recurso que se utiliza para realizar las distribuciones y en segundo lugar porque es el objeto sobre el cual se realizan las ordenes de mantenimiento.

Esta entidad tiene una estructura un poco más compleja a diferencia de la entidad anteriormente descrita, ya que guarda relaciones con la tabla tbl_wilaya y la tabla tbl_tipo_vehiculo. La tabla tbl_camion se define con sus campos y relaciones como muestra la figura 2.2.

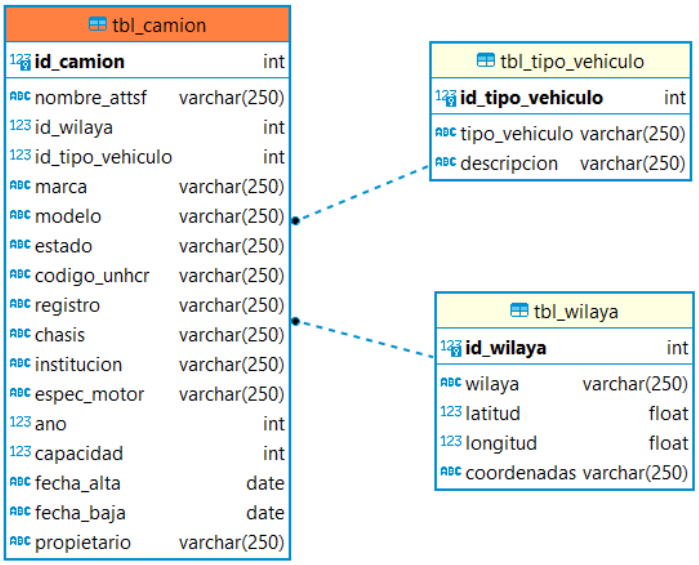


Figura 2.2: Entidad: tbl_camion

El tipo de relación que se dan entre la entidad tbl_camion y tbl_wilaya es de 1 a varios (1:N), de esta manera puede haber varios camiones en una Wilaya, pero un camión no puede tener más de una Wilaya como ubicación base.

Del mismo modo, la relación entre tbl_camión y tbl_tipo_vehiculo es de 1 a varios bajo el mismo principio que con las tbl_wilaya.

Sus campos se presentan a continuación en la tabla 2.2:

Nombre del Campo	Apreciaciones
id_camion	Identificador único *(Primary Key)*.
nombre_attsf	Nombre de identificación que usa la organización.
id_wilaya	Identificador que referencía a una instancia de la tabla tbl_wilaya *(Foreign Key)*.
id_tipo_vehiculo	Identificador que referencía a una instancia de la tabla tbl_tipo_vehiculo *(Foreign Key)*.
marca	Marca del vehículo.
modelo	Modelo del vehículo.
estado	Si es un camión activo ("Activo") o dado de baja ("Baja").
codigo_unhcr	Código de identificación por el propietario.
registro	Registro del propietario.
chasis	Numero de identificación del chasis.
institucion	Nombre dela institución.
espec_motor	Número de identificación del motor.
ano	Año del modelo del vehículo.
capacidad	Capacidad de transporte en toneladas.
fecha_alta	Fecha de alta del vehículo.
fecha_baja	Fecha de baja del vehículo.
propietario	Nombre del propietario.

Tabla 2.2: Campos de tbl_camion

- Campo *id_camion*

  Este capo corresponde a la ***Primary Key*** que se asignará a cada una de las instancias de la entidad. Este campo es un número entero, por lo que se define como tipo de dato int.

- Campo *nombre_attsf*

Este campo corresponde a un código de tipo alfanumérico con el cual la organización identifica y distingue a cada uno de los camiones. Este campo se ha descartado como *Primary Key* ya que en algunas ocasiones la identificación no era consistente, especialmente para identificar el "estado" de los camiones.

- Campo *id_wilaya*

  Identificador que referencía a una instancia de la tabla tbl_wilaya *(Foreign Key)*. A pesar de que los camiones realizan viajes a través de diferentes regiones, estos al final del día siempre vuelve a su región de base, por esta razón se relacionan directamente y no a diferentes niveles de relación como podría ser a través de otras entidades.

- Campo *id_tipo_vehiculo*

  Identificador que referencia a una instancia de la tabla tbl_tipo_vehiculo *(Foreign Key)*, el cual describe a qué flota pertenece un camión, la flota de agua, de alimentos o de residuos. No se describen los detalles de esta entidad con anterioridad, sin embargo se hace referencia al tipo de vehículo a través de otra entidad ya que se espera que en el futuro puedan añadirse más tipo de flotas de vehículos con otros propósitos.

- Campo *marca* y *modelo*

  Estos campos corresponden a la marca y modelo comercial de los camiones. No son atributos críticos para el estudio de los datos por lo que se asignan como cadenas de texto cortas (varchar(250)) y sin restricciones, también admiten valores nulos, como podrían ser una *Foreign Key* o posibles valores por defecto.

- Campo *estado*

  Este campo es un campo de texto corto, que se asigna como tipo de dato varchar(250). Puede almacenar dos cadenas de texto diferentes, "Activo" para un camión que sigue en operación, y "Baja" para un camión que ya no esta en operación o bien que ha sido cambiado de ubicación, es decir, un camión se da de baja cuando, a pesar de mantener los mismo atributos excepto el de id_wilaya, que es la ubicación, este ha sido dado de baja, sin embargo se sabe que en un periodo de tiempo ha estado en una ubicación distinta. No se hace referencia a estos dos estados con una nueva entidad, ya que difícilmente en el tiempo pueden existir múltiples estados. No admite valores nulos.

- Campo *codigo_unhcr*

  Este campo corresponde al código de identificación del fabricante, se asigna como tipo de dato de texto corto, varchar(250), y tampoco se considera un atributo crítico para el estudio de los datos, por lo que puede admitir valores nulos.

- Campo *registro*

  Este campo también se le asigna tipo de dato varchar(250) y al igual que los últimos anteriores no se considera un atributo crítico para la entidad, por lo que puede admitir valores nulos.

- Campo *chasis*

  Este campo corresponde al número de serie del chasis, se asigna como tipo de dato de texto corto, varchar(250).No se considera un atributo crítico para la entidad.

- Campo *institucion*

  Existen diferentes instituciones las cuales hacen las donaciones de camiones, este campo es un campo sin restricciones de texto corto, varchar(250) para asignar a la institución donante.

- Campo *espec_motor*

  Este campo se refiere a especificaciones del motor, es de tipo texto sin restricciones, varchar(250), por lo que puede admitir valores nulos.

- Campo *ano*

  Este campo se refiere al año de la marca y el modelo del camión del fabricante. Se le asigna como dato de tipo entero, int. Puede admitir valores nulos.

- Campo *capacidad*

  Es la capacidad en toneladas que puede transportar un camión. Es un dato de tipo entero, int. Puede admitir valores nulos.

- Campo *fecha_alta*

  Corresponde a la fecha en la que se dio de alta el registro del camión, es dato de tipo fecha (*date*) en formato 'YYYY-MM-DD'.

- Campo *fecha_baja*

Corresponde a la fecha en la que se dio de baja un camión, ya sea por fin de operaciones o bien por que ha cambiado su atributo id_wilaya, es dato de tipo fecha (*date*) en formato 'YYYY-MM-DD'.

Si un camión se da de baja por cambio de ubicación base, id_wilaya, un nuevo camión es registrado.

NOTA: no puede existir un camión con estado "Activo" que tenga información en el campo fecha_baja, por el estado, este debería ser de valor *null*. Por lo que este campo puede admitir valores nulos.

- Campo *propietario*

  Finalmente, este campo recopila información del propietario del camión, que podría ser la propia organización u otra organización. Se asigna como campo sin restricciones y de tipo texto corto, varchar(250).

2.4.3. Registro de ordenes de distribución y ordenes de trabajo (tbl_distribución y tbl_ot)

Como se ha descrito anteriormente, se han conceptualizado dos tablas de hechos que registran los procesos de la organización dentro del alcance del proyecto, la tabla tbl_distribución, la cual se relaciona con las tablas tbl_conductor, tbl_tipo_producto, tbl_camion, tbl_wilaya en relaciones 1 a varios (1:N) de la misma manera que se ha descrito la entidad tbl_camion en la sección 2.4.2, y la tabla tbl_ot, la cual guarda relaciónes con las tablas tbl_frecuencia, tbl_taller, tbl_wilaya, tbl_camion, tbl_personal, tbl_tipo_ot y especialmente con las tablas tbl_ot_averia y tbl_ot_repuesto. Se detallarán los componentes de la entidad tbl_ot con el fin de describir las relaciones de varios a varios (N:N) que se dan en el modelo.

Las relaciones de tipo varios a varios, se han gestionado añadiendo en el diseño un par de tablas intermedias, la tabla tbl_ot_averia y la tabla tbl_ot_repuesto. Para el caso particular de estas relaciones, es necesario crear estas tablas intermedias ya que, una orden de trabajo de tipo preventiva puede estar relacionada con varios repuestos, de igual forma una orden de tipo correctiva puede estar relacionada con varias averías, y a su vez los repuestos y averías pueden estar relacionadas con varias ordenes de trabajo de tipo preventivas y correctivas respectivamente.

En la figura 2.3 se muestra la relación de varios a varios que hay entre las averías y las ordenes de trabajo. La tabla intermedia tbl_ot_averia almacena los pares de valores (id_ot/id_avería), de forma que una avería puede estar relacionada con varias ordenes de trabajo y una orden de trabajo puede estar relacionada con varios tipos de averías.

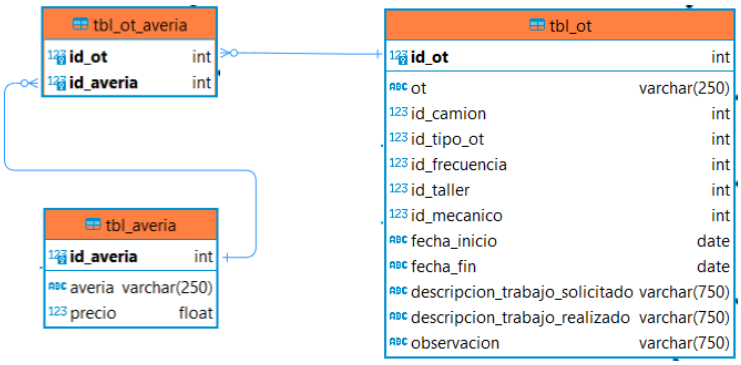


Figura 2.3: Relación varios a varios (N:N)

Los campos de la entidad tbl_ot se muestran a continuación en la tabla 2.3.

Nombre del Campo	Apreciaciones
id_ot	Identificador único *(Primary Key).*
ot	Número de identificación que usa la organización.
id_camion	Identificador que referencía a una instancia de la tabla tbl_camion *(Foreign Key).*
id_tipo_ot	Identificador que referencía a una instancia de la tabla tbl_tipo_ot *(Foreign Key).*
id_frecuencia	Identificador que referencía a una instancia de la tabla tbl_frecuencia *(Foreign Key).*
id_taller	Identificador que referencía a una instancia de la tabla tbl_taller *(Foreign Key).*
id_mecanico	Identificador que referencía a una instancia de la tabla tbl_mecanico *(Foreign Key)*
fecha_inicio	Fecha de inicio de la reparación.
fecha_fin	Fecha de fin de la reparación.
descripcion_trabajo_solicitado	Descripción del requerimiento de orden de trabajo.
descripcion_trabajo_realizado	Descripción del trabajo se que ha realizado al vehículo.
observacion	Observaciones adicionales

Tabla 2.3: Campos de tbl_ot

- Campo *id_ot*

  Este capo corresponde a la ***Primary Key*** que se asignará a cada una de las instancias de la entidad. Este campo es un número entero secuencial, por lo que se define como tipo de dato int.

- Campo *ot*

  Este campo corresponde a un código de tipo numérico con el cual la organización identifica las ordenes de trabajos realizadas, ya sean de tipo preventivas o correctivas. Este campo se ha descartado como *primary key* ya que en algunas ocasiones la identificación no era consistente, especialmente porque una orden de tipo correctiva podría detener el mismo número de identificación que una de tipo preventiva.

- Campos *id_camion, id_tipo_ot, id_frecuencia, id_taller, id_mecanico*

Como se establece en la tabla 2.3, cada uno de estos campos referencían una instancia de sus respectivas tablas. Estos campos en la entidad tbl_ot funciona como claves foráneas *(Foreign keys)* y son todos números enteros secuenciales, por lo que se les asigna el tipo de dato int.

- Campo *id_fecha_inicio*

  Corresponde a la fecha en la que se da inicio a una orden de trabajo, esta fecha no es igual a la fecha en la cual un camión podría dejar de estar disponible. Es dato de tipo fecha *(date)* en formato 'YYYY-MM-DD'.

- Campo *id_fecha_fin*

  Corresponde a la fecha en la que se da fin a una orden de trabajo, esta fecha podría ser igual a la fecha en la cual un camión podría comenzar a estar disponible. Es dato de tipo fecha (*date*) en formato 'YYYY-MM-DD'

- Campo *descripción_trabajo_solicitado*

  Este campo es un campo de texto más extenso, se asigna como tipo de dato varchar(750) y corresponde a una breve descripción del trabajo que se solicita hacer al vehículo, puede admitir valores nulos.

- Campo *descripcion_trabajo_realizado*

  Este campo es un campo de texto más extenso, se asigna como tipo de dato varchar(750) y corresponde a una breve descripción del trabajo que se ha realizado al vehículo, en algunas ocasiones no coincide con la descripción solicitada por falta de materiales, puede admitir valores nulos.

- Campo *observacion*

  Este campo también se le asigna tipo de dato varchar(750) y al igual que los últimos anteriores no se considera un atributo crítico para la entidad, por lo que puede admitir valores nulos.

Para cada una de las entidades que conforman el diseño de la base de datos se han establecido los atributos y relaciones de la misma manera en como se han establecido para las entidades en las secciones 2.4.1, 2.4.2 y 2.4.3, considerando el tipo de dato que almacenan, la información que describen y estableciendo la relación con las tablas de hechos que se han definido para los procesos de la organización.

2.5. Diseño lógico

Durante la fase del diseño lógico, un diseño lógico (o *schema*), que corresponde al modelo de datos, que a su vez corresponde al Sistema de Gestión de Base de Datos DBMS (DBMS por sus sigla en inglés: *Data Base Management System*) es creado; por ejemplo un modelo de datos relacional. Esta fase también se conoce como implementación del diseño ya que representa la transformación del esquema conceptual al esquema lógico del DBMS.

Para dar pie a la creación de la base de datos se ha hecho una conexión al *schema* de nombre AttsfTd a través de la herramienta para gestión de conexiones de bbdd DBeaver con el *driver* MS SQL Driver de SQL server. Dentro de este *schema* se encuentra la base de datos de nombre LisData y a la cual se le da la estructura del modelo desarrollado en las secciones anteriores. Para dar la estructura de tablas y relaciones de la base de datos, se ha creado y ejecutado en un *script* de tipo .sql con los comandos necesarios para la creación de tablas y establecer las restricciones y relaciones.

Este archivo de extensión .sql se encuentra almacenado en la carpeta del proyecto ATTsf Ordenes de Producción del repositorio de Git de LIS Data Solutions.

Finalmente la imagen 2.4 muestra el Modelo Entidad- Relación de la base de datos que se diseño e implementó para centralizar la información de los diferentes orígenes de los procesos de la organización. En el diagrama se muestra a cada una de las entidades (tablas), sus atributos, el tipo de dato que almacenan, y las relaciones entre ellas, siendo las tablas tbl_distribución y tbl_ot las tablas que registran la información de los hechos que se llevan a cabo en los dos principales procesos que están dentro del alcance de este proyecto.

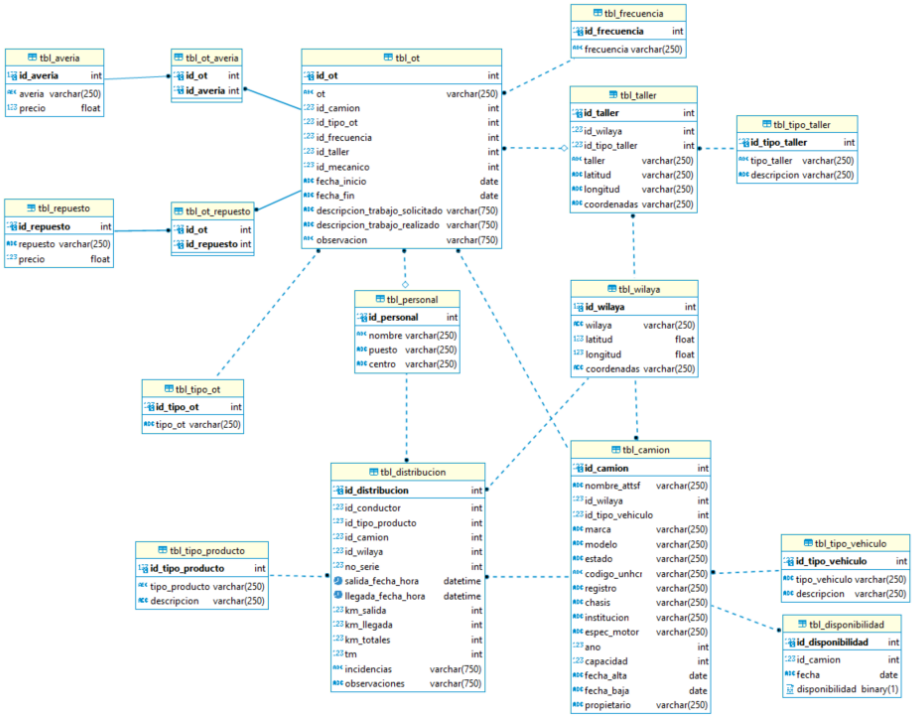


Figura 2.4: Modelo Entidad-Relación ATTsF

2.6. Diseño físico

El diseño físico de la base de datos transforma el diseño lógico en un formato adecuado para el *hardware* dado y el sistema de gestión de base de datos. Este mapea el esquema lógico en una representación almacenada apropiada y determina los parámetros físicos necesarios para optimizar el desempeño de la base de datos en contra de un conjunto de transacciones requeridas [Storey, 1991].

La naturaleza del volumen de datos históricos generados es de alrededor de 10 mil registros totales con valor de 1 kB. Se estima que cada año se añadan en promedio 2 mil registros equivalentes a 200 bytes, por lo que el proyecto no requiere gran capacidad de escalabilidad. Considerando el volumen de datos, se ha elegido en la arquitectura del proyecto un servidor *on–premise*, además de que el cliente ya tenía la accesibilidad a este.

Características del Servidor *On-Premise*:

- RAM Instalada: 8.00 GB.

- Tipo de sistema: Sistema operativo de 64 bits.
- Procesador: Intel(R) Xeon (R) CPU E3-1225 v6 @ 3.30 GHz.
- Disco de sistema: SSD Segate 1 Tb.

Capítulo 3

Procesos ETL

Un proceso ETL se define como el proceso que organiza el flujo de los datos entre diferentes sistemas de una organización y aporta métodos y herramientas necesarias para mover datos desde múltiples fuentes, reformatearlos, limpiarlos, o bien transfórmalos, y cargarlos en otra base de datos o almacén de datos [Martínez Trujillo, 2018].

El objetivo de esta primera fase de procesos ETL es poder preprocesar los datos históricos desde cierto rango temporal que se tienen en la organización y poder transformarlos para después cargarlos en la base de datos diseñada.

La otra fase de procesos ETL es la que se lleva a cabo en la actualización de los datos, y lo que será la puesta en producción de la propuesta solución y arquitectura que se ha planteado en la sección 1.3. Esta segunda fase de procesos ETL se detallará en la sección 3.4 de este capítulo.

3.1. Extracción de datos

En la fase de extracción se preprocesan los datos provenientes de la fuente, o varias fuentes de origen, a un formato homogéneo y consolidado para iniciar la siguiente fase, es decir, para llevarlos a la fase de transformación [Martínez Trujillo, 2018].

Algunas de las cosas ha tomar en cuenta para que una extracción de datos se lleve de manera adecuadas son [Martínez Trujillo, 2018]:

- Extraer la información desde el sistema o sistemas de origen.
- Analizar la información realizando una revisión previa.
- Verificar que la información extraída cumple con lo esperado.

- Convertir los datos a un formato requerido para iniciar con la fase de transformación.

Anteriormente ya se ha mencionado sobre las múltiples fuentes de datos, por un lado se trata de ficheros excel y por otro lado de datos provenientes del sistema ERP de la organización. Se considera importante mencionar los detalles de como se organizan estas diferentes fuentes de origen de datos.

Como se muestra en la figura 3.1 se cuenta con cuatro ficheros excel que recopilan la información de los procesos que se describieron en la sección 2.1.2 para las diferentes flotas de camiones, y a su vez, algunos de estos ficheros tienen varios libros u hojas que estructuran y organizan los datos para el caso de la flota de agua y residuos.

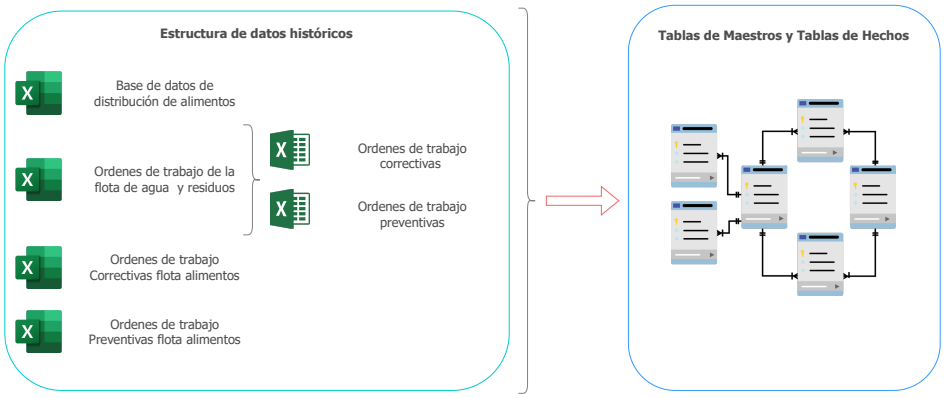


Figura 3.1: Estructura de datos históricos

De entre todo el universo de información de la organización existe información histórica desde el año 2011, sin embargo en el alcance de este proyecto se recopila solo información a partir del año 2019. Se tienen entonces cuatro ficheros excel diferentes por cada año de historificación hasta el mes de mayo del año 2023. Es a partir de los datos del mes de junio del 2023 en donde se incorpora el sistema ERP como fuente de datos, ya que este actualmente esta en implementación como sistema de información en la organización. De esto se habla más en detalle en la sección 3.4 sobre la actualización de los datos.

De acuerdo con la arquitectura propuesta, la ingesta de los datos de origen se hace de forma local, es decir, los ficheros son compartidos a través del *share point* de LIS Data Solutions, por lo que en el *share point* asignado al proyecto se descargan cada uno de los fichero excel de forma local.

Las herramientas que se utilizan para comenzar con la extracción de los datos son *notebooks* ejecutables en lenguaje de programación Python a través de la interfaz de Visual Studio Code.

Pandas es una de las librerías principales para el procesamiento de los datos de origen. Previamente se ha hecho una exploración de los datos directamente en los ficheros excel para definir cuales serian las columnas o campos seleccionado para la extracción de datos.

Al recopilar todos los ficheros y leer los archivos correspondientes se concatena la información de todos los años en dos objetos *dataframe* de la librería pandas de python, uno correspondiente a las ordenes de trabajo y otro correspondiente a las ordenes de distribución.

Una vez los datos han sido homogeneizados en esta estructura se procede a la fase de transformación.

3.2. Transformación de los datos

En esta fase se aplican una serie de reglas o funciones sobre la información que a sido extraída para convertirlos en datos que después serán cargados. Esta fase se conoce también la fase en la que se da la limpieza de los datos [Martínez Trujillo, 2018].

En la fase de transformación de los proceso ETL de este proyecto se llevan a cabo en dos principios generales, como ya se había mencionado, uno que consiste en la transformación de datos que serán cargados en las tablas Maestros y la otra que consiste en la transformación de datos que serán cargados en las tablas de Hechos.

Entre la información que se requiere cargar para las tablas Maestros, hay información que se transforma de forma directamente manual, esta información se transforma a través de acciones en lenguaje de programación de python. Un ejemplo de ello sería la información que se transforma para la entidad tbl_tipo_producto, ya que no se requieren demasiadas acciones para reestructurarla, pues solo existen tres tipo de productos que pueden ser transportados por los camiones. En la figura 3.2 se muestra también como se hace la agregación de la columna "descripción", la cual es planteada en el Modelo Entidad - Relación del diseño de la base de datos. Notese también que a este *dataframe* ya se le esta dando la estructura que se requiere para ser cargado en la entidad correspondiente en la base de datos, asignado un id o identificador único a cada registro de la entidad que en la base de datos será la *Primary Key* correspondiente.

	id_tipo_producto	tipo_producto	descripcion
0	1	CB	Canasta Basica
1	2	PF	Producto Fresco
2	3	E	Enseres

Figura 3.2: Dataframe: df_tipo_producto

Por otro lado, hay Maestros que requieren mayor número de acciones de transformación así como las tablas de Hechos para limpiar y organizar la información. Entre las acciones más relevantes para la limpieza de datos para maestros más complejos y tablas de Hechos se llevan a cabo las siguientes:

Gestión de valores nulos: Hay diversas formas de gestionar los valores nulos. Para el caso de las tablas Maestros, se ha considerado eliminar los registros con valores nulos correspondientes a campos como la identificación de un camión o el nombre de una wilaya, ya que estos son valores que repetidamente aparecen en los registros.

Agregación de columnas: se agregan columnas que contengan determinada información que se cree necesaria para clasificar la información. Como se vio anteriormente con el *dataframe* correspondinete a tbl_tipo_producto, se agregan por ejemplo columnas asignando el tipo de vehículo de un camión, el cual pude ser CC que se refiere a Camión Cisterna, CA que se refiere a Camión Alimentos, etc.

Dividir una columna en varias: esta acción consiste en dividir o extraer información de una columna a otra. Utilizando de nuevo el ejemplo de la tabla camiones, se ha revisado que, para identificar los camiones, se les asigna un número, se desconoce el criterio de asignación ya que no son números consecutivos, sin embargo otros podrían tener códigos alfanuméricos de tipo "CC5", lo que indica que "CC" corresponde al tipo de vehículo. Consecuentemente se extraen estos primeros caracteres para asignar así el atributo tipo de vehículo para cada camión. El código alfanumérico se conserva, ya que es la referencia que los usuarios utilizan para identificar los camiones en la organización.

Formateo de datos: esta acción consiste en eliminar los espacios en blanco al principio y al final de cada registro de cada campo, y a su vez, homogeneizar el formato de caracteres, de manera que si, se indicaba el tipo de camión, todo los registros del campo "tipo_camion" fueran en caracteres en mayúsculas ("CC"), pero si, de otro modo, el registro comprende al campo "marca" el formato de caracteres debería ser de tipo "title" ("Land Rover").

Asignación de identificadores únicos: finalmente a cada una de las Tablas de Maestros se les ha asignado un identificador único. Este identificador único consiste en un número de tipo entero, los cuales se asignaron de forma consecutiva teniendo como valor mínimo el número 1 hasta el número máximo de registros.

Unión de relaciones: De acuerdo con el diseño del Modelo Entidad - Relación, se requiere unir la información de una tabla con otra con la que tiene relación.

Esta acción se logra con la función merge() de python, se seleccionan solamente las columnas deseadas a unir, que para el caso de las tablas con relaciones es el identificados único y el nombre del campo asociado.

Selección de campos: Una vez que se han hecho las uniones de las tablas que tienen relaciones, se seleccionan los campos requeridos de cada una de las tablas del diseño del Modelo Entidad - Relación.

3.3. Carga de datos

Esta es la fase en la que los datos son cargados en el sistema de almacenamiento destino [Martínez Trujillo, 2018]. Dependiendo de los requerimientos de la organización este proceso puede abarcar una gran variedad de acciones. Para este proyecto, los *dataframes* de cada una de las tablas de Maestros y Hechos son exportadas en ficheros con extensiones .csv para almacenarlas de forma local proceder a ser cargados en el nuevo destino.

Para asegurar la carga correcta de los datos, se ha asegurado de que el tipo de datos de cada una de las columnas de cada uno de los *dataframes* sea el adecuado y el que ha sido asignado en el diseño de la base de datos.

3.3.1. Conexión con el servidor de base de datos

A través de Python en Visual Studio Code y con el paquete de herramientas de SQLAlchemy, se establece la conexión de el servidor de la base de datos SQL Server administrado en DBeaver. Se asegura que al introducir los parámetros necesarios, para la generación del motor en python de la base de datos la conexión se haya establecido correctamente.

3.3.2. Ejecución de carga de datos

Una vez es establecida la conexión, se llaman y leen los ficheros de extensión .scv generados previamente a partir de los *dataframes* para ejecutar

una acción que vuelque todos los datos en la tabla correspondiente de la base de datos en SQL Server.

3.4. Actualización de datos

En este punto, todos los datos históricos hasta el mes de mayo del 2023 ya fueron actualizados y centralizados en la base de datos en SQL Server, por lo que esta fase consiste en la actualización de los datos que, a partir de esta fecha, se añaden como nueva información y así poder mantener la base de datos con información actualizada. Esta fase sería parte de los entregables del proyecto y lo que es la puesta en producción de la solución propuesta y la arquitectura planteada en la sección 1.3.

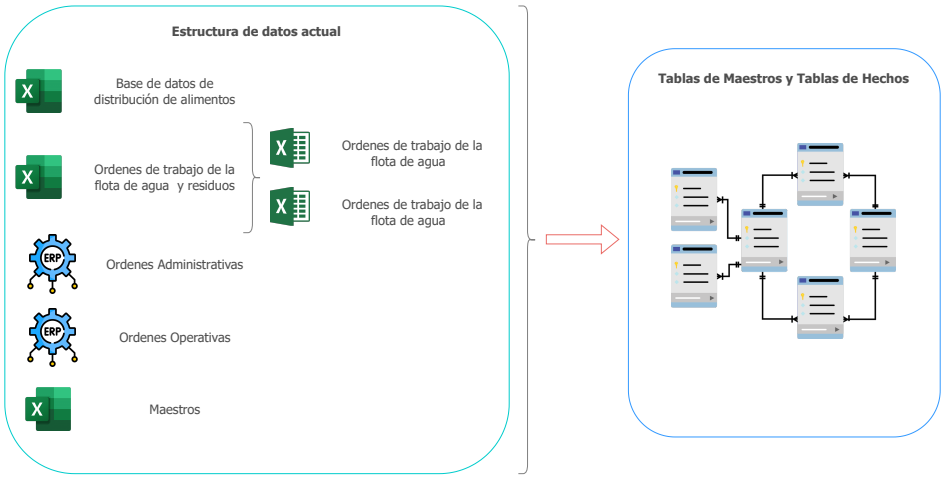


Figura 3.3: Estructura de nuevos datos

En la figura 3.3 se muestra gráficamente la estructura del nuevo origen de datos que, a diferencia del diagrama de la figura 3.1 este tiene integrado el sistema ERP que esta siendo adoptado como sistema de información en la organización. Esta nueva estructura difiere en que en esta ocasión ya no se cuenta con ficheros excel para las ordenes de trabajo correctivas y preventivas para la flota de camiones de alimentos, si no que ahora el origen de ésta información son lo que se conoce como ordenes administrativas y operativas que ya se han descrito en la sección 2.1.1. Además, la información proveniente del sistema ERP es llamada a partir de procedimientos almacenados en el mismo *schema* donde se encuentra almacenada la base de datos en SQL Server.

Para cumplir con el propósito de la actualización de los datos se desarrolla un *script* en python de ejecución automática con periodicidad semanal, el cual tiene la estructura que se muestra en la figura 3.4[1].

Este *script* consiste esencialmente en recibir los nuevos datos que se van añadiendo en los registros de los procesos día con día, ejecutar procesos ETL, similares a los que se han descrito para la historificación de los datos, hacer la conexión son el servidor de base de datos y posteriormente cargar los nuevos datos haciendo diferentes comprobaciones para asegurar la calidad de los nuevos datos.

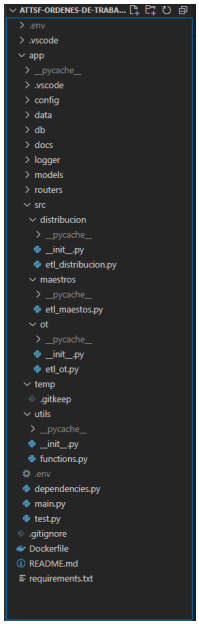


Figura 3.4: Estructura script ATTsF

[1] https://github.com/monalcmar/TFM-LIS-ATTsF

Capítulo 4

Análisis de Datos

El objetivo de esta última fase del proyecto, es poder tener la habilidad para transformar los datos en información que de valor, y que esta información se vuelva en conocimiento de forma que pueda optimizar los procesos de toma de decisiones en la organización. Es en esta última parte en donde el valor agregado se verá fuertemente reflejado.

Un concepto destacable es el que aporta Robert Stackowiak, define la inteligencia de negocios como el proceso de tomar datos, analizarlos y presentarlos en un conjunto de reportes que condensan la esencia de esta información en la base de las acciones del negocio, lo que permite a la gerencia tomar decisiones fundamentales sobre el negocio [Rivera Resina et al., 2018].

Por otro lado, el análisis que se da en esta fase del proyecto es un análisis mayormente exploratorio de los datos, tanto cuantitativo como cualitativo, y consiste en desarrollar el segundo los de los dos entrégales del proyecto, una herramienta de análisis desarrollada en tableros e informes de Power BI que ayude a los usuarios en la organización a tomar decisiones más informadas sobre sus procesos.

Algunos de los componentes que conforman un informe son los siguientes:

- **Dashboard**: un *dashboard* o tablero es una representación gráfica de los principales KPIs que intervienen en la consecución de los objetivos de negocio, y que está orientada a la toma de decisiones para optimizar la estrategia de la empresa. Un *dashboard* debe de representar las KIPs necesarias de forma que sean relevantes en el contexto de la organización [Rivera Resina et al., 2018].

- **KPIs**: KPI (por sus siglas en ingles, Key Performance Indicators) son un método para determinar el éxito del negocio utilizando métricas ob-

jetivas y directrices cuantificables. Pueden ser utilizados para evaluar el estado global de la organización, comparar el plan de negocio con el desempeño actual o hacer seguimiento al progreso de una actividad en particular, meta, departamento o producto.

- **Gráficos**: son un componente esencial en la descripción y análisis de datos. Consiste en obtener nueva información al cambiar de un sistema de representación a otro. Por ejemplo, al pasar de un listado de datos a un histograma, se puede percibir el valor de la moda, que antes no era visible en los datos brutos [Arteaga, 2008].

4.1. Carga y modelado de datos en Power BI

En esta fase los datos son cargados desde el *Data Mart* en SQL Server a la herramienta de análisis a través de una conexión con el servidor, en las siguientes secciones se describen los detalles de la carga y modelado de los datos en Power BI Desktop.

4.1.1. Carga de datos

Power BI tiene la facilidad de obtener datos a partir de diferentes fuentes de datos, tanto en arquitectura *on - premise* y en la nube. Para el caso de este proyecto, el origen o la carga de datos en Power BI se da a través de una conexión remota con SQL Server.

De este modo, a través de Power PI Desktop se establece la conexión con SQL introduciendo los parámetros necesarios para establecer la conexión.

Cuando la conexión con el servidor de Base de Datos se ha establecido correctamente, Power BI permite seleccionar las tablas que se quieren cargar al informe. Se seleccionan todas la tablas de la base de datos.

4.1.2. Modelado de datos

En en proceso de modelado se recogen los datos y se crea una estructura utilizable para Power BI. Existen dos tipos de estructuras utilizables básicas, el modelo en estrella y el Modelo Copo de Nieve.

Modelo Estrella: Es la estructura más sencilla. Hay una tabla de hechos central que contiene los datos para el análisis, rodeada de las tablas de dimensión. En este modelo la única tabla que tiene relación con otra es la de hechos, lo que significa que toda la información relacionada con una dimensión debe estar en una sola tabla [Rivera Resina et al., 2018].

Las tablas de dimensiones deben tener una clave primaria, la clave principal de la tabla de hechos estará compuesta por las claves principales de las tablas de dimensiones.

Modelo Copo de Nieve: este modelo se diferencia del Modelo Estrella principalmente en que no solo la tabla de hechos tiene relación con otras tablas, sino que hay otras tablas que se relacionan con las dimensiones sin tener relación con los hechos [Rivera Resina et al., 2018].

Para conseguir un esquema en copo de nieve se ha de tomar un esquema en estrella y conservar las tabla de hechos, centrándose únicamente en el modelado de las tablas de dimensión, que ahora se dividen en subtablas.

A pesar de que se han cargado todas las tablas de la base de datos, el modelo de datos no es adecuado para trabajar con Power BI, ya que, se establecen por defecto las relaciones tal y como están en el modelo de origen, sin embargo para gestionar los datos de una forma adecuada en Power BI es necesario establecer relaciones en la tablas de tal modo que sean consistentes con un Modelo Estrella o de Copo de Nieve.

Se ha establecido modelar los datos a modo de un Modelo de Copo de Nieve, ya que no son solo las tablas de hechos las que establecen relación con otras tablas. La figura 4.1 muestra el modelo con el cual se trabajara en Power BI para la generación de los informes. En este modelo, principalmente se ha duplicado la tabla de wilayas, ya que hay otras tablas que mantienen relación con esta misma, lo cual no es consistente con el modelado requerido para gestión de los datos en Power BI.

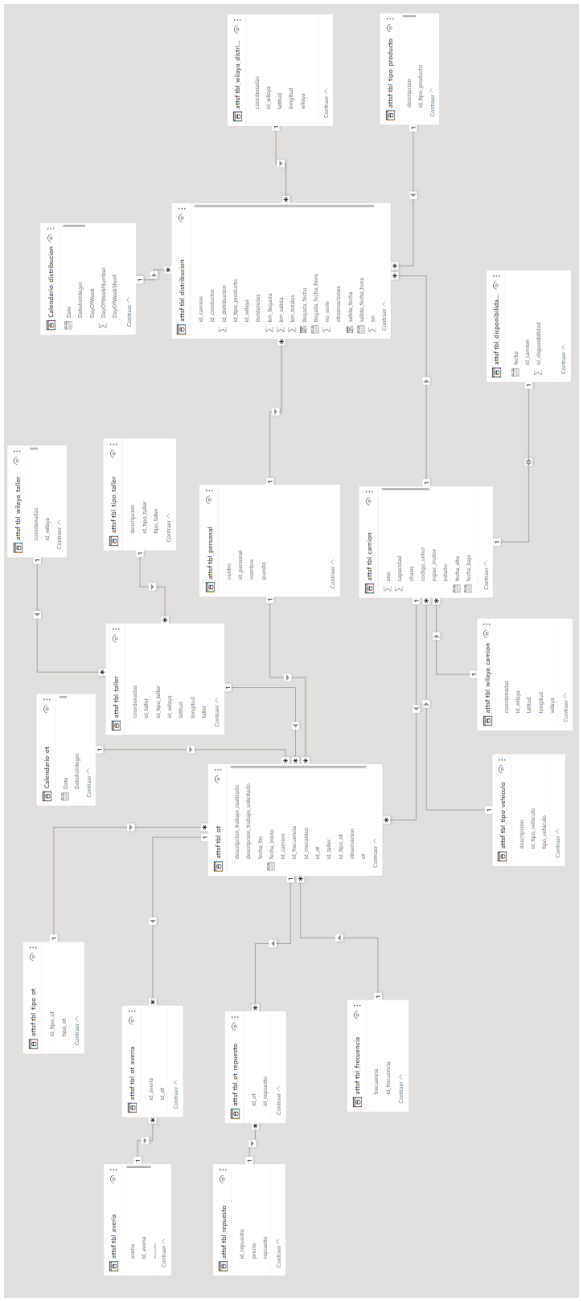


Figura 4.1: Modelo de Copo de Nieve

4.2. Procesamiento de datos

En esta fase se da formato a los datos en los cuales sea necesario. Desde que se han transformado los datos en los procesos ETL estos ya han sido gestionados en una forma casi totalmente adecuada, sin embargo para el gestión de estos datos en Power BI se han llevado un par de acciones para la gestión adecuada de los mismos.

Se han tenido que procesar los campos con tipo de datos de fecha, ya que en la base de datos el formato de fecha es YYYY/MM/DD HH:MM, sin embargo parecía no ser la forma más adecuada para la gestión de fechas en Power BI sobre todo si se quería hacer cálculos o agrupaciones por estas dimensiones.

A través de Power Query, uno de los componentes principales de Power BI para ejecutar procesos ETL, se transforman las columnas de datos de tipo fecha al formato YYYY/MM/DD. Adicionalmente, se agregan dos nuevas tablas de dimensión, el calendario maestro, una para cada tabla de hechos del modelo. El calendario maestro tiene como dimensión tipos de datos fecha en diferente granularidad, por ejemplo, año, mes, cuarto de año, semana del año, etcétera.

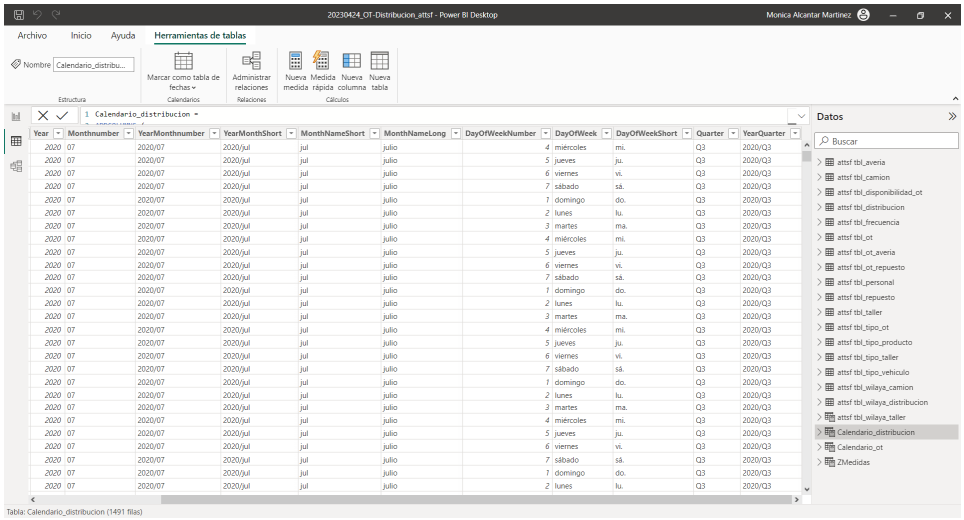


Figura 4.2: Calendario Maestro

4.3. Creación de los informes y tableros

En esta fase es en la que se le da valor a los datos creando conocimiento a través de recoger la información, transfórmala y finalmente crear los informes en función de los requisitos del cliente [Rivera Resina et al., 2018].

Es necesaria una buena comunicación con el cliente para definir cual sería la información que se requiere representar en los informes y si hay preguntas que responder a partir de la información o bien, que decisiones se quieren tomar. Es muy común que en ocasiones los usuarios que hacen la toma de decisiones tengan una idea de lo que quieren analizar o visualizar, sin embargo no siempre esta idea suele ser muy clara, el cual es el caso para el desarrollo de este análisis.

Para esta tarea se fueron generando propuestas de informes y tableros que fueron evolucionando a través de *workshops* con los usuarios. Algunos de los tableros desarrollados, se detalla en las secciones 4.3.2 y 4.3.3 destacando los resultados obtenidos.

4.3.1. Definición de los principales KPIs

La imagen en la figura 4.3 muestra los principales indicadores, de los diferentes procesos, que se solían utilizar en la organización. Ya que la idea del nuevo análisis no era del todo clara, se partió de la información que brinda esta figura esperando que, conforme se hubiese ido desarrollando el análisis y a partir de *workshops* con el cliente de las propuestas que se plantearon, las ideas se fueran esclareciendo. De esta manera se han modelado los tableros e informes que forman parte de la herramienta desarrollada.

Dentro del área geográfica en la cual se hacen las distribuciones, existen seis regiones base (Wilayas), por lo que es importante poder analizar la información a nivel de cada una de ellas. Esto ayuda a darse cuenta cual de estas regiones es la más relevante en los procesos. Por otro lado uno de los indicadores más importantes en el análisis son los kilómetros totales recorridos.

Adicional a los kilómetros recorridos, otros de los indicadores más relevantes es el número de viajes realizado y la cantidad de toneladas distribuidas.

Para los usuarios de la organización es indispensable poder analizar las ordenes de trabajo en base al tipo de vehículo, ya que el tipo de vehículo describe básicamente a que flota o grupo de camiones pertenece el vehículo. Por lo que el número de ordenes de trabajo por cada tipo de ellas se convierte en un indicador relevante.

RESUMEN DE LA ACTIVIDAD DE LA BdT ENERO 2021

DATOS DE DISTRIBUCIÓN Y TALLER ENERO 2021	
TOTAL DE VIAJES REALIZADOS	235
TOTAL TONELADAS REPARTIDAS	2.828.56 Tm
TOTAL TONELADAS CANASTA BÁSICA	2.315.31 Tm
TOTAL TONELADAS PRODUCTO FRESCO	513.25 Tm
TOTAL TONELADAS EXTRAS	0.0 Tm
PORCENTAJE DE TONELADAS DISPONIBLES	87.22%
ORDENES DE TRABAJO CORRECTIVAS VEHÍCULOS	31
ORDENES DE TRABAJO CORRECTIVAS INSTALACIONES	04
ORDENES DE TRABAJO PREVENTIVO CAMIONES	10
ORDENES DE TRABAJO PREVENTIVO VEHÍCULOS AUXILIARES	00
ORDENES DE TRABAJO PREVENTIVO MAQUINARIA	00

Figura 4.3: Principales indicadores ATTsF

4.3.2. Resultados de análisis de los procesos de Distribución

En el tablero a) de la figura 4.4 se muestra que la Wilaya en la que más kilómetros se recorren es la Wilaya de nombre Dajla, pero por otro lado la Wilaya con mayor número de viajes realizado es la de nombre Aaiun, la cual a sus vez tiene casi la misma cantidad de kilómetros recorridos que Dajla, esta información indica que en la región de Dajla, los viajes que se realizan son más extensos respecto a otras regiones, lo cual podría suponer también menor eficiencia en las toneladas de producto transportadas respecto a otras regiones.

El nivel más sencillo de análisis es el tipo de producto distribuido, ya que previo al análisis de los datos se sabe que el producto mayormente distribuido es el que corresponde al de la canasta básica de alimentos, sin se cree conveniente analizar los datos a nivel geográfico.

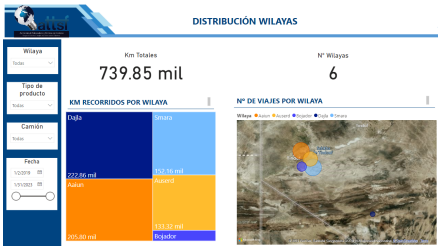
En tablero b) se muestra el total de toneladas de producto distribuido, mayormente son los productos de la canasta básica los de mayor volumen transportado, y como ya lo había mostrado el tablero a) la Wilaya en donde menos toneladas son distribuidas es Dajla.

En el tablero c) y en el tablero d) se analizan las toneladas, el número de viajes realizados y los kilómetros totales recorridos por cada camión. Los datos en este tablero muestran que hay camiones que a pesar de tener similar número de kilómetros recorridos, unos más que otros, distribuyen mayor cantidad de producto. Esto indica que hay camiones con mayor capacidad

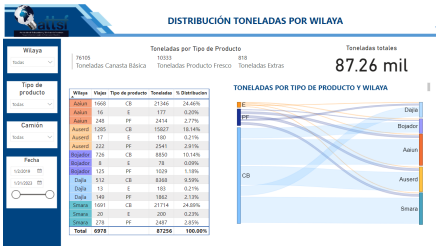
y que por lo tanto su eficiencia respecto a los viajes realizados y kilómetros recorridos es mejor.

En el tablero e) se analizan los kilómetros recorrido a nivel de camión y día del mes. En este tablero se puede detectar los camiones con más actividad y también los camiones con menos. Lo más relevante de este tablero es que se detecta patrones en los cuales se puede ver que entre los días 5 y 15 de cada mes del año es donde se da el mayor número de kilómetros recorridos.

Finalmente en lo que respecta al análisis de la distribución, el tablero f) muestra los resultados anuales de los tres principales indicadores para los procesos de distribución. En este tablero se tiene un gráfico que muestra la evolución de estos indicadores a lo largo de los datos históricos, y por otro lado, los indicadores muestran los resultados anules del año actual respecto al año anterior.

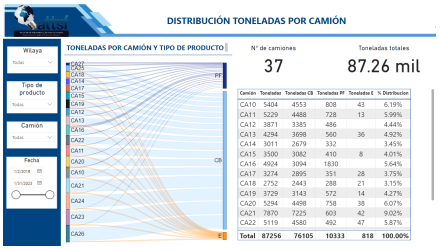


a) Tablero de la Distribución a nivel Wilaya

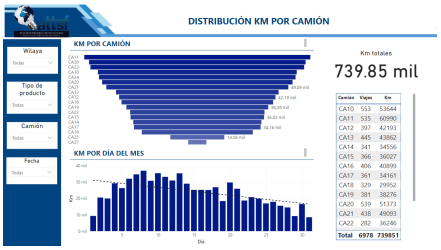


b) Tablero de la Distribución en toneladas a nivel Wilaya

c) Tablero de la Distribución a nivel Camión



d) Tablero de la Distribución en toneladas a nivel Camión



e) Tablero de la Distribución en KM a nivel Camión

f) Tablero Indicadores de la Distribución Anual

Figura 4.4: Análisis de los datos de la Distribución

4.3.3. Resultados de análisis de los procesos de Ordenes de Trabajo

El tablero a) describe la cantidad total de camiones y como se distribuyen por tipo de camión y en las diferentes Wilayas. Se puede ver que el hay más camiones de tipo cisterna y le sigue en cantidad los camiones de alimentos. Y la mayoría de camiones se concentran en la Wilaya Rabouni.

Partiendo del análisis general que se hace en el tablero a) se comienza a seccionar el análisis por camiones de tipo alimentos. El tablero b) indica el número total de ordenes de trabajo que corresponden a esta flota, del cual el 33 % corresponden a ordenes de tipo preventivas y el 66 % a ordenes de

tipo correctivas. Este dato es crítico, ya que lleva a analizar en profundidad si este resultado de da a falta de una buena planificación de mantenimiento preventivo o bien las razones podrían ser ajenas a este motivo.

Así mismo este mismo tablero, muestra por camión que tipo de orden de trabajo se les realiza mayormente. Como es de esperarse a la mayoría de camiones se les realizan más ordenes de trabajo de tipo correctivas.

Haciendo un análisis más detallado sobre las ordenes de trabajo de tipo correctivas, en el tablero c) se aprecia que el mayor motivo de una avería en este tipo de orden es por causa de los neumáticos, seguido de fallos eléctricos del vehículo. Esta información ayuda a establecer medidas preventivas respecto a la calidad de los neumáticos y revisión sobre las condiciones eléctricas de los vehículos. En este tablero, además se puede visualizar como evolucionan la cantidad de averías que se registran por mes, este dato se mantiene casi constante excepto por el mes de mayo en donde hay un pico de menos averías y el mes de enero y septiembre donde existe un poco de mayor número averías. Además es importante saber cuantos camiones de la flota han sufrido una avería en determinado tiempo.

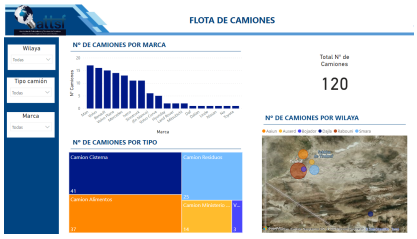
En el tablero d) se muestra la interacción entre el número de averías y los kilómetros recorridos para cada camión. Se presta atención en sí los kilómetros recorridos tienen relación con el número de averías, para esto se comparan este par de indicadores entre camiones, sin embargo con la ayuda de las visualizaciones, no se puede detectar un patrón entre estas dos medidas. Entre los camiones con mayor número de kilómetros recorridos no se registran el mayor número de averías. Sin embargo, los camiones con mayor número de averías tiene relativamente la misma distancia recorrida.

Por otro lado, se registran ordenes de tipo preventivas para los camiones de alimentos, sin embargo no se registran otros detalles de los tipo de mantenimientos preventivos que se hicieron, por lo que el tablero e) tendrá información un poco incompleta.

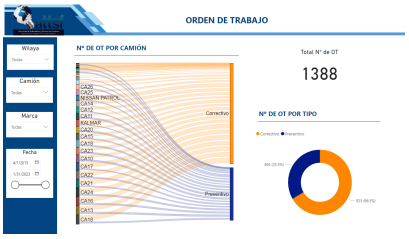
A partir de ahora se analizan los datos del tipo de camiones que no son parte de la flota de alimentos, los camiones cisterna, de residuos y otros tipos. Este conjunto de tipos de camiones suman en total de 83 camiones e igual que los camiones de alimentos, el porcentaje de ordenes de tipo correctivas es mayor por el 64 % que las de tipo preventiva con el 36 %. Esta información se muestra en el tablero f).

Como era de esperarse, en el tablero g), se muestra que el mayor motivo por el que un vehículo de las demás flotas se le efectúa una orden de tipo correctiva es por avería de neumáticos, seguido de falla eléctrica del vehículo. Además se puede ver que hay un pico de mayor número de averías en el mes de enero.

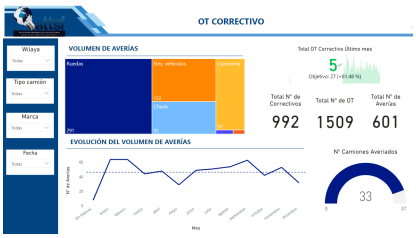
Finalmente, para el resto de vehículos y con ordenes de tipo preventivas si es posible tener más información de los datos que se tienen. Los datos muestran que 65 de los 83 vehículos tienen al menos una orden de trabajo preventiva. Se cuestiona este dato, ya que teóricamente se deberían realizar trabajos de mantenimiento a todos lo vehículos. Por otro lado, se muestra en el tablero h) que el mayor consumo de materiales en los mantenimientos preventivos es en anticongelante seguido de aceite para la caja de cambios. En estas ordenes también se detecta un pico de mayor número de ordenes de tipo preventivo en los meses del primer trimestre del año.



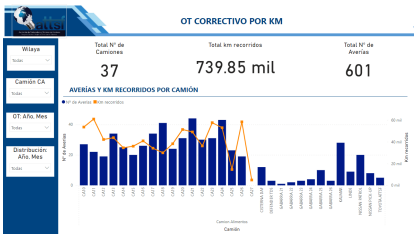
a) Tablero de OT por Número de Camiones a nivel de Flota y Wilaya



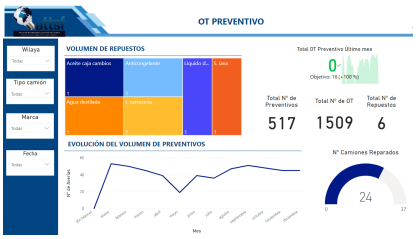
b) Tablero de OT de Camiones de Alimentos por Tipo de Orden de Trabajo



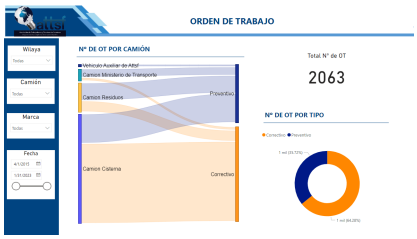
c) Tablero de OT de Tipo Correctivo para Camiones de Alimentos



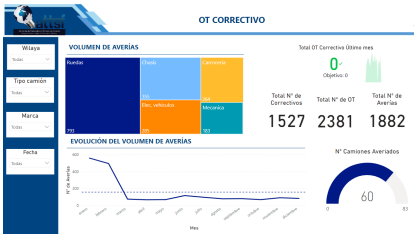
d) Tablero de OT de Camoines de Alimentos de Tipo Correctivo. No. Averias vs Km Recorridos



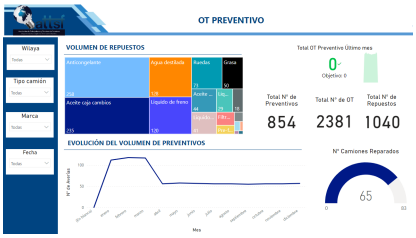
e) Tablero de OT de Tipo Preventivo para Camiones de Alimentos



f) Tablero de OT de Camiones de Agua por Tipo de Orden de Trabajo



g) Tablero de OT de Tipo Correctivo para Camiones de Agua



h) Tablero de OT de Tipo Preventivo para Camiones de Agua

Figura 4.5: Análisis de los datos de las Ordenes de Trabajo

Capítulo 5

Conclusiones

A lo largo de este proyecto, se han implementado soluciones para lograr que la organización ATTsF cuente con una nueva arquitectura tecnológica en la forma en la que gestionan sus datos. El diseño de su nueva base de datos, ayuda a que sus datos estén mejor estructurados, sean más consistentes y sean de mejor calidad, esto último implementando los procesos de transformación adecuados para darle a los datos un formato homegéneo y también más legible. Además, esta nueva base de datos tiene la función de ser la nueva fuente de datos para que estos puedan ser accesibles y puedan ser consultados y analizados por los usuarios que lo requieran. Finalmente esta solución permitirá que los datos sean resguardados y actualizados de forma que el riesgo de perder información se minimice.

En cuando a los resultado del análisis realizado, es importante considerar que los KPIs en sí mismos no son metas ni objetivos para la organización, si no que son una forma de medir el correcto funcionamiento de sus procesos y sobretodo poder detectar los puntos críticos en donde se necesita implementar medidas de acción para mejorar en estos puntos. Sin el uso de los KPIs establecidos se tardaría mucho tiempo y esfuerzo en realizar la recopilación y procesamiento de los datos para obtener información sobre el estado generar de los procesos de la empresa y esto afecta que que se demoraría la toma de acciones en el tratamiento de los problemas que se puedan presentar.

A partir de la definición de los principales KPIs en la organización y la generación de los tableros e informes de análisis, se han podido detectar puntos críticos y retos que resolver en los procesos tanto de distribución como de ordenes de trabajo, entre lo más importantes:

- Los camiones con mayor capacidad de distribución, no están siendo utilizados para realizar viajes de distancias más largas. Si se conside-

ran estos camiones para las distancias más larga, se podría reducir el número de viajes realizados, teniendo mayor alcance en kilómetros recorridos y toneladas de producto distribuidas, haciendo de esta manera los procesos de distribución más eficientes.

- Los informes indican que hay días del mes, en que se da mayor actividad en los procesos de distribución, por lo que valdría la pena prestar más atención en cuales son los factores que favorecen este comportamiento.
- El mayor número de ordenes de trabajo son de tipo correctivas, más que preventivas, esto índica que se están llevando a cabo acciones a posteriori que a priori en cuando a los mantenimientos que se les da a los vehículos.
- Son los vehículos de la flota de alimentos los que mayormente se averían, sin embargo falta más información para conocer cuales son los factores externos o internos que hacen que esto ocurra.

5.1. Trabajo futuro

La finalización de este proyecto es apenas el comienzo de una de las primeras fases en las que LIS Data Solutions tiene colaboración con la organización ATTsF. En esta proyecto se han abordado los primeros pasos para ayudar a la organización a conocerse mejor y fortalecer el conocimiento que se tiene sobre sus datos. Sin embargo, el análisis que se ha hecho hasta ahora, es un análisis poco robusto y con áreas de oportunidad mayores.

En trabajos futuros se espera poder poder abordar los siguientes aspectos generales:

- Integración de información sobre los costes en la Ordenes de Trabajo y datos sobre la disponibilidad de los vehículos.
- Implementación de modelos de *data mining* como reglas de asociación con el objetivo de generar planes de mantenimiento estratégicos en los camiones de las diferentes flotas.
- Integración de los datos de *tracking* de GPS en los vehículos para poder estudiar factores externos e identificar puntos de rotura en rutas de distribución.

Bibliografía

[ADRIAN,] ADRIAN, T. E. Modelo conceptual-entidad relación.

[Anbumani et al., 2021] Anbumani, V., Geetha, V., Kumar, V. P., Sabaree, D., and Sivanantham, K. (2021). Development of cloud-based agriculture marketing system with intellectual weigh machine. In *IOP Conference Series: Materials Science and Engineering*, volume 1055, page 012016. IOP Publishing.

[Antiñanco, 2014] Antiñanco, M. J. (2014). *Bases de Datos NoSQL: Escalabilidad y alta disponibilidad a través de patrones de diseño*. PhD thesis, Universidad Nacional de La Plata.

[Arteaga, 2008] Arteaga, P. (2008). Análisis de gráficos estadísticos elaborados en un proyecto de análisis de datos. *Trabajo fin de Master. Departamento de Didáctica de la Matemática*.

[Bayer, 2010] Bayer, M. (2010). Sqlalchemy documentation. *URL: https://www. sql. alchemy. org.(last.*

[Flores, 2023] Flores, F. (2023). Qué es visual studio code y qué ventajas ofrece.

[González Duque, 2011] González Duque, R. (2011). Python para todos.

[Gowthami and Kumar, 2017] Gowthami, K. and Kumar, M. P. (2017). Study on business intelligence tools for enterprise dashboard development. *International Research Journal of Engineering and Technology*, 4(4):2987–2992.

[Martínez Trujillo, 2018] Martínez Trujillo, T. (2018). Gestión de datos empresariales utilizando procesos etl.

[Rivera Resina et al., 2018] Rivera Resina, F. J. et al. (2018). Aplicación de busines intelligence en una pequeña empresa mediante el uso de power bi.

[Sánchez, 2004] Sánchez, J. (2004). Principios sobre bases de datos relacionales. *Informe, Creative Commons*, 11:20.

[Santamaría and Hernández, 2016] Santamaría, J. and Hernández, J. (2016). Microsoft sql server. *SQL SER vs MY SQL*, pages 1–6.

[Storey, 1991] Storey, V. C. (1991). Relational database design based on the entity-relationship model. *Data & knowledge engineering*, 7(1):47–83.