PROYECTO OLAP

Definición del desarrollo

Ainhoa Álvarez Fernández

Índice

1. Contextualización

a. Descripción del problema

Este proyecto se basará en el análisis de datos recogidos por la DGT sobre matriculaciones y diferentes trámites sobre vehículos en España entre los años 2015 a 2024. Inicialmente, vamos a realizar un tratamiento de los datos de la DGT, junto con otros relacionados a las provincias, comunidades autónomas, y municipios de España. El proyecto consiste en una parte de ETL, en la que se preparan las diferentes dimensiones que compondrán nuestro data warehouse, del que más adelante extraeremos información para hacer visualización de diferentes analíticas.

Pretendemos analizar las tendencias que hubo durante esta etapa en cuanto a información de los vehículos sobre los que se realizan los trámites, así como el perfil de las personas, o cómo puede afectar el número de ventas en una zona específica a dichos trámites. Esto ayudará a en un futuro, poder hacer análisis más exhaustivos y entender mejor las tendencias de estas acciones.

b. Descripción de los datos

Para este proyecto se utilizan cinco fuentes de información, todas csv. La primera, y la central sobre la que se realizará todo el proyecto, es la de la DGT. Esta fuente de información contiene columnas de todo tipo, desde información de los vehículos, hasta los datos del trámite y de la persona encargada. Además, tuve que modificar el archivo un poco, para lo cual usé un notebook, en el que de manera aleatoria, reducía su tamaño total a solo 3 millones de filas para que fuese manejable para trabajar con él. En este notebook además, se hace comprobación de que ninguna de las filas seleccionadas esté duplicada, para asegurar que toda la información que vayamos a usar para nuestras transformaciones sea correcta.

De este primer dataset, vamos a centrarnos mayoritariamente en la información que nos proporciona de los vehículos que se tramita, como la marca, el modelo, la potencia, la provincia de domiciliación, el tipo de vehículo, la cilindrada… De nuevo, este archivo será nuestro archivo central, sobre el que basaremos nuestro estudio.

Después, para la parte de la ubicación, se necesitaron dos csv, uno que relaciona comunidades autónomas con sus provincias, y otro que recoge todas las provincias de España y sus municipios. Estos dos archivos los usaremos para poder crear nuestra dimensión tiempo, y poder ubicar los trámites en lugares concretos de España.

Otro dataset que vamos a utilizar es el de ventas de vehículos en España entre los años 2015 y 2024, para poder comparar los datos y ver cómo afectaron al número de trámites. Este dataset era complicado de conseguir, ya que eran muchos años, entonces con información de la DGT y de ANFAC, generé un archivo python que me permitió crear este archivo con 2 millones de datos, con información de ventas de vehículos por provincia, siguiendo las tendencias económicas que sucedieron esos años, y con pesos asociados a cada comunidad autónoma.

Y por último, otro csv generado por python debido a la falta de uno, que analiza el número de personas cotizantes por provincia, para estudiar cómo puede afectar eso también al

mercado de los coches y los trámites que se hacen. Para generar este csv de nuevo nos basamos en datos reales como lo son la población por provincia en base a 2024, el porcentaje de personas trabajando en cada comunidad autónoma para darles un peso diferente, y la evolución anual que tuvo respecto a situaciones sociales económicas vividas.

Una vez elegidos los datasets, desarrollamos el diagrama de estrella sobre el que vamos a basar nuestro estudio de datos.

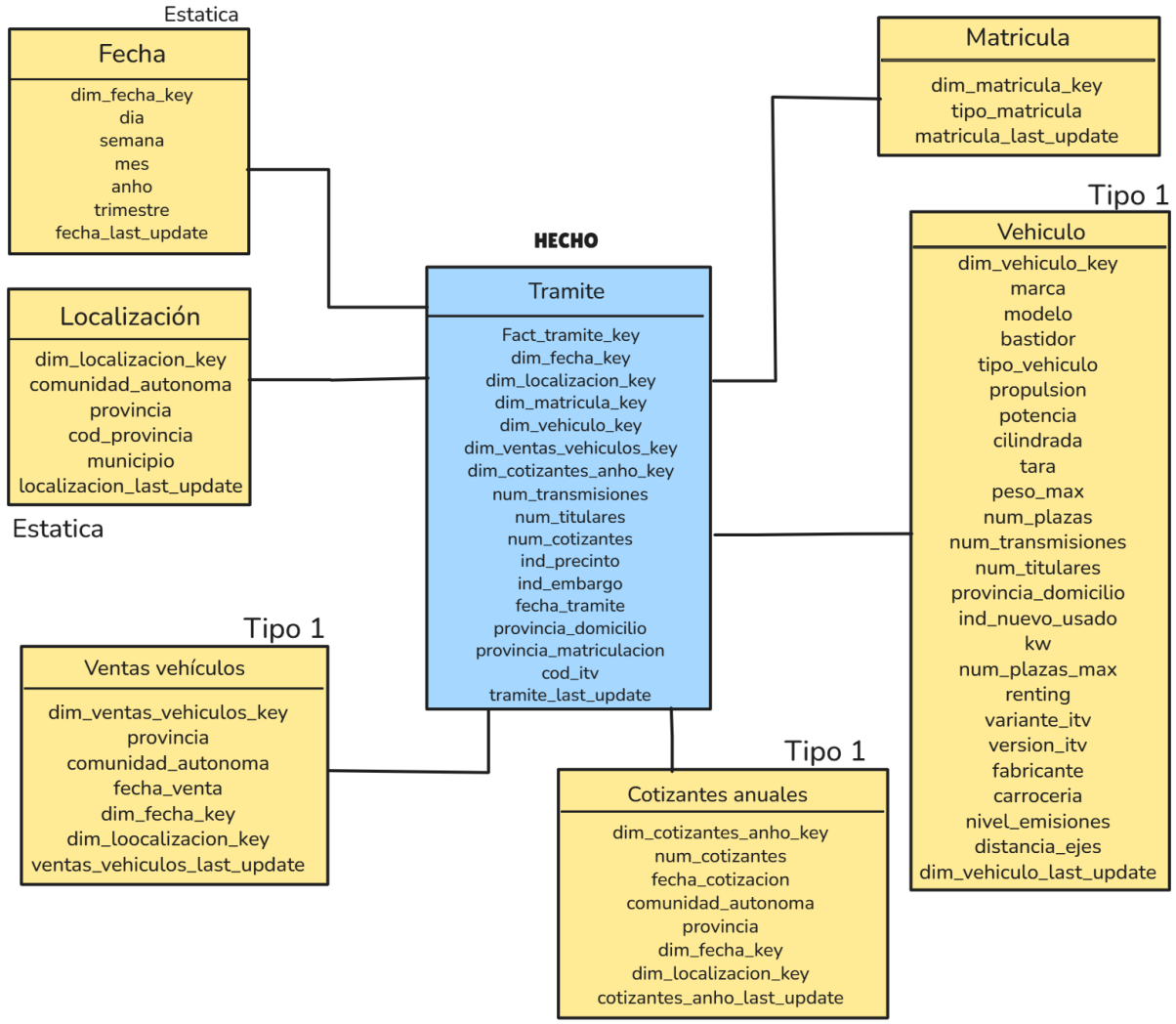

Figura 1: Diagrama estrella

Como podemos ver en la figura 1, el diagrama estrella está compuesto por 6 dimensiones y una tabla de hechos. Como se puede ver en la figura especificado, tres de las dimensiones son estáticas, las otras tres de tipo 1. Vamos a explicar una a una la decisión sobre el tipo.

En el caso de la fecha, decidí el uso de una dimensión estática, ya que los datos están agrupados en un intervalo muy específico, por lo cual se generan las filas necesarias para cubrir el intervalo entero, y ya no se vuelve a actualizar. De manera muy similar, se genera la

dimensión localización. Como el mapa de España no va a fluctuar, se genera de manera estática para todas las combinaciones de comunidad autónoma, provincia y municipio, añadiendo también la posibilidad de no conocer el municipio en el que ocurre el trámite.

Por otro lado tenemos las dimensiones de tipo 1. Decidí hacerlas de tipo 1 y no de tipo 2, porque no vi relevante mantener un historial de datos y modificaciones, sino que era más práctico y lógico desde mi punto de vista que si alguna de las tuplas sufría una modificación, se actualizase dicha tupla, y no generar una nueva versión de la misma con los datos actualizados.

En estas dimensiones de tipo 1 tenemos la dimensión vehículo, que es la más grande en cuanto a datos. En ella se recogen los datos de los vehículos que se tramitan. Esta dimensión es de tipo 1 porque un vehículo no va a modificar sus componentes principales como pueden ser la marca o el modelo, o la potencia que tiene, los únicos atributos que se cambiarán son las flags, como el número de titulares que tiene o si está nuevo o usado, y por tanto no tenía sentido hacerla de tipo 2 y generar una nueva fila. Por otro lado está la dimensión de ventas de vehículos. De nuevo, esta dimensión será de tipo 1, ya que se podrán actualizar los datos recogidos sobre ventas en años anteriores, pero no le veía relevancia en caso de que hubiese alguna modificación, el conservar los datos iniciales errados. Y por último, y de manera muy similar a la anterior, tenemos la dimensión de cotizantes al año. De nuevo, de tipo 1 porque si se hacen más estudios sobre esta información y hay que cambiarla, no tiene sentido mantener la tupla errada y generar una nueva. Además, tenemos la dimensión matrícula que será de tipo uno, ya que los tipos de matriculación podrían variar en el tiempo, pero si alguno se modifica, no tiene sentido mantener el valor anterior.

Por último y no menos importante, tenemos la tabla de hecho, que será la que acceda a todas las demás para optimizar las consultas. A mayores de las claves a tablas externas, como métricas elegí tres flags para estudiar, el indicador de embargo, el indicador de precinto, o si el vehículo es nuevo o usado. También me pareció interesante estudiar en relación al número de titulares de un vehículo y el número de transmisiones de titular que ha sufrido desde su primera matriculación. Analizar con respecto al número de personas cotizantes en ese período, e incluso el desplazamiento que hay de la provincia de domicilio del vehículo y en la que se matriculó.

Una vez tenemos este esquema montado, podemos empezar con las transformaciones etl para mejorar los datos y darle mejor forma para utilizar en los dashboards.

2. Proceso ETL

Vamos a analizar ahora las diferentes transformaciones etl que se han aplicado para cada dimensión, para entender cómo se han transformado estos datos para su uso.

2.1 Dimensión fecha

Vamos a analizar los pasos de creación para la dimensión fecha. Como mencionamos antes, esta es una dimensión de tipo estático, por lo que todos los datos se generarán solo cuando se ejecute su transformación y no sufrirán cambios.

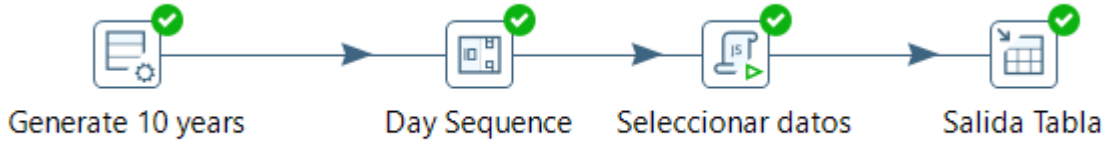

Figura 2: Transformación load_dim_fecha

Vamos a analizar los pasos seguidos en la figura 2. Tenemos primero un paso en el que generamos los valores necesarios para representar el intervalo de fechas desde 2015 a 2024 con todos los posibles días de todos los meses. A continuación, tenemos un paso que es el que se encargará de generar la clave de la dimensión, de manera autoincremental. Lo siguiente es un script en el cual, con el valor de fecha que entra, la transformamos en formato simple, y extraemos los valores día, semana dentro del mes, mes dentro del año, el año, el trimestre al que pertenece, y por último la fecha en formato adecuado para la base de datos. Por último, el paso final nos permite insertar los valores generados en la base de datos del data warehouse, completando así la primera dimensión estática.

2.2 Dimensión localización

Vamos a analizar los pasos de creación para la dimensión localización. Como mencionamos antes, esta es una dimensión de tipo estático, por lo que todos los datos se generarán solo cuando se ejecute su transformación y no sufrirán cambios.

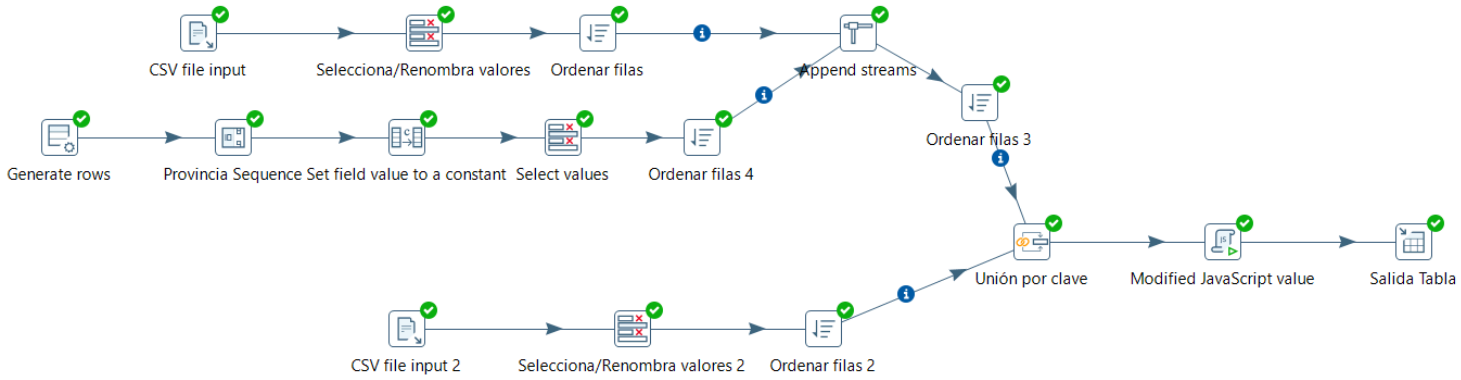

Figura 3: Dimensión load_dim_localización

En esta figura 3 podemos ver 3 diferentes entradas de datos. Hablaremos de los flujos de arriba hacia abajo. El primer flujo hace una lectura de nuestro csv con información sobre las provincias y sus municipios. Después de eso se hace una rápida limpieza de los datos, eliminando campos sobrantes y cambiando los nombres de dos atributos para trabajar más cómodamente con ellos. Una vez hecho esto, se ordenan las filas por orden de la provincia, ya que para el siguiente paso es necesario que los dos flujos tengan las provincias en el mismo orden.

En el segundo flujo lo que se hace es generar 52 filas de un atributo municipio con valor vacío. Esto es para poder insertar la posibilidad de tener una comunidad autónoma y

provincia, y desconocer el municipio en el que se ubica. Después de esto le generamos a estas filas con solo municipio, una columna provincia, con valores del 1 al 53, para que se correspondan con las provincias en el flujo de arriba. Después de generar el atributo provincia, hacemos un set to constant, y nos aseguramos de que el campo municipio esté vacío para todas las filas de la tabla. A continuación, e igual que en el caso del flujo de arriba, renombramos las variables, y ordenamos por provincia el flujo.

De ahí, se hace un append streams. Este campo lo que hace es unir dos flujos de datos por una clave. En este caso, por el código de provincia, se unían los dos flujos, teniendo así para cada provincia la lista con sus municipios y un municipio vacío como comodín. Después de ese append, se vuelve a ordenar todo por provincia, porque de nuevo, para el paso de la unión por clave, necesitaremos que los dos campos estén ordenados por dicha clave.

El tercer flujo, el de abajo de todo de la transformación, comienza con la lectura de un csv que contiene los valores de las comunidades autónomas y sus provincias asociadas. Después de cargar la información del archivo, realizamos una limpieza de datos y renombramos las variables que nos parezcan, y de ahí hacemos otro ordenar, por la clave de la provincia para que coincida el orden de los dos flujos.

Una vez tenemos el flujo 1 y el 3 ordenados, hacemos una unión por clave. Esta unión lo que permite es que al flujo que se marque como principal, se le añadan los atributos del flujo secundario, siguiendo la clave que se indique. En este caso hacemos la unión por provincia, y como nuestro flujo principal es el de las comunidades autónomas, a cada comunidad autónoma con su provincia se le van a añadir todos los municipios asociados que venían del flujo 1. Para finalizar esta transformación tenemos un script que nos permite sacar el código ine del municipio a partir de su nombre, lo cual nos será útil para otras dimensiones en el futuro, y después de eso, la salida a tabla, que de nuevo es insertar todos los valores generados en la tabla de dimensión localización de nuestro data warehouse.

2.3 Dimensión matrícula

Vamos a analizar los pasos de creación para la dimensión matrícula. Como mencionamos antes, esta es una dimensión de tipo 1, por lo que los datos se generarán al inicio, y en caso de modificación, serán sobreescritos.

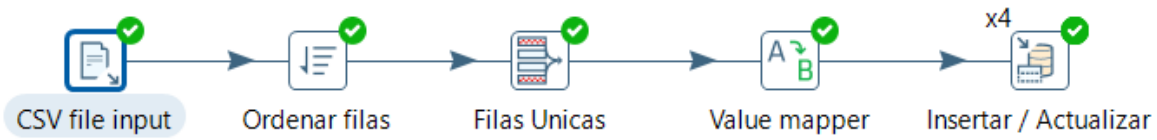

Figura 4: Transformación load_dim_matrícula

Esta transformación es muy sencilla.Como podemos observar en la figura 4, primero se lee el atributo cod_clase_matricula del csv de matriculaciones. Lo que haremos en esta transformación será simplificarlo para que su uso sea mejor. Después de eso, ordenamos las

millones de filas por el valor del campo, y de ahí lo pasamos por un unique rows. El paso este, se encarga de que solo una fila de cada valor pase en el flujo, por lo que nos quedaremos con 9 filas diferentes. Una vez hemos simplificado los datos de manera exponencial, hacemos un value mapper, y asignamos a cada código de matrícula, su valor real en palabras, para que después sea entendible por los usuarios. Por último, insertaremos todos los datos en la base de datos de nuestro data warehouse.

2.4 Dimensión ventas vehículos

Vamos a analizar los pasos de creación para la dimensión ventas vehículo. Como mencionamos antes, esta es una dimensión de tipo 1, por lo que los datos se generarán al inicio, y en caso de modificación, serán sobreescritos.

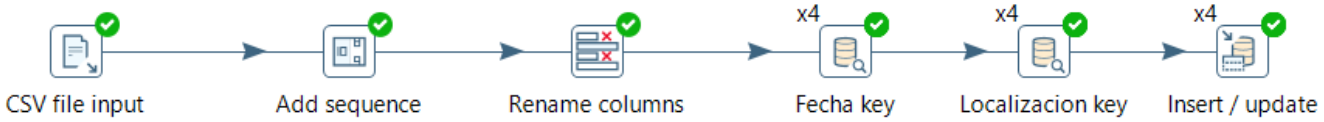

Figura 5: Transformación load_dim_ventas_vehiculo

Vamos a analizar lo que se ve en la figura 5. Esta dimensión comienza con la carga de datos del csv de ventas de vehículos. Leeremos los atributos todos relacionados con la fecha de la venta, la comunidad autónoma y la provincia. De ahí, usaremos el paso add sequence para añadir la clave de la dimensión, que será una autoincremental. Una vez tenemos la clave generada, pasamos por un select values para renombrar los atributos para trabajar con mayor comodidad, y de ahí pasamos a lo interesante. Vamos a hacer un LookUp a la dimensión de fecha, para retornar la clave de la tabla fecha y que sea más manejable a posteriori, y a su vez haremos un lookup de la dimensión localización por el valor de la provincia, sin municipio, por el mismo motivo que el caso de la fecha. Una vez tenemos todos estos datos, hacemos un inster update a la tabla en el data warehouse.

2.5 Dimensión cotizantes año

Vamos a analizar los pasos de creación para la dimensión cotizantes año. Como mencionamos antes, esta es una dimensión de tipo 1, por lo que los datos se generarán al inicio, y en caso de modificación, serán sobreescritos.

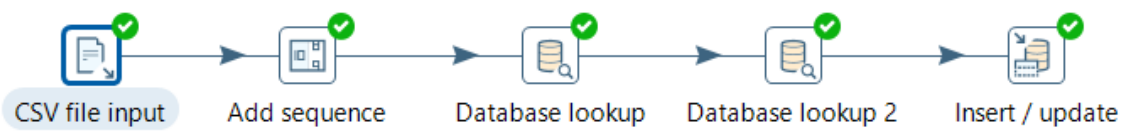

Figura 6: Transformación load_dim_cotizantes_anho

Como vemos en la figura 6, esta transformación es muy similar a la anterior. Cargaremos los datos del csv de cotizantes año. Con estos datos, añadiremos el campo clave

de la tabla, que será un autoincremental generado. Después de eso, se hacen dos lookups, el primero a la dimensión fecha y el segundo a la dimensión localización sin el municipio de nuevo. Esto de nuevo es para agilizar tiempo más adelante. Por último se hace un Insert/update para guardar todos los datos en el data warehouse.

2.6 Dimensión vehículo

Vamos a analizar los pasos de creación para la dimensión vehículo. Como mencionamos antes, esta es una dimensión de tipo 1, por lo que los datos se generarán al inicio, y en caso de modificación, serán sobreescritos.

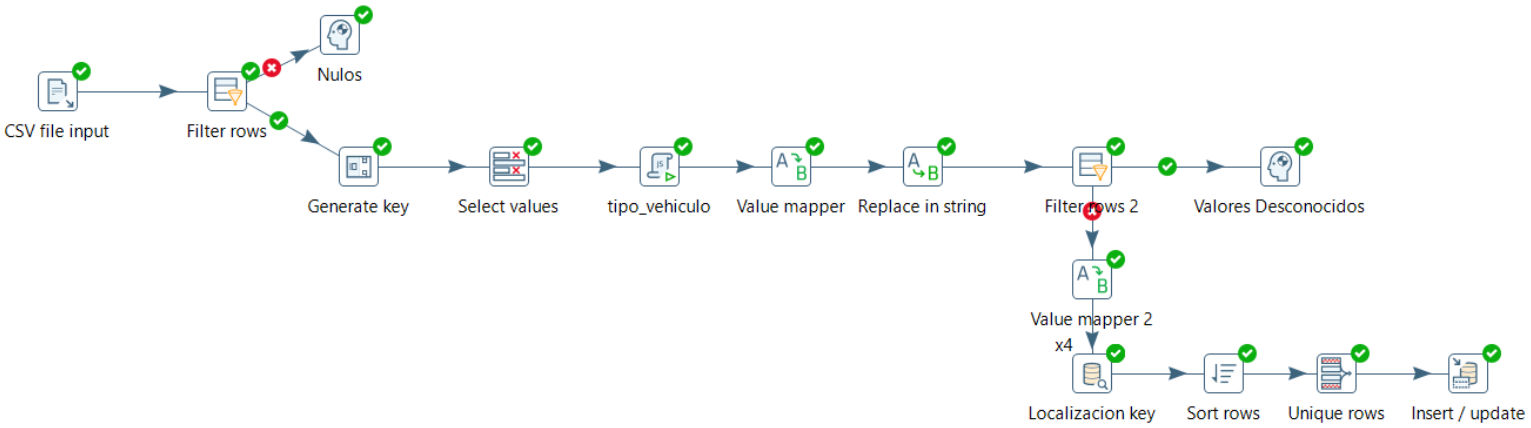

Figura 7: Transformación load_dim_vehículo

En la figura 7 podemos comprobar que esta transformación es un poco más compleja. Comenzamos leyendo todos los atributos del csv de matriculaciones reducido, y de ahí, después de hacer análisis, decidí que las filas en las que el valor del modelo fuese desconocido, o en las cuales la carrocería fuese desconocida, eran tan pocas que eran despreciables, por lo tanto, con un filter rows compruebo si dichos atributos no están vacíos. Si no están vacíos, el flujo continúa, y si lo están se van al paso de Nulos, en los que no pasa nada. Después de esta comprobación, inserto de nuevo la clave autoincremental generada en el paso de sequence, y después de eso, hago una limpieza de atributos y pongo nombres más comprensibles para trabajar con ellos.

Una vez limpios los campos, tengo un script que lo que hace es mapear el típo de vehículo, dependiendo del valor del primer carácter del campo. En el dataset original los tipos de vehículos son muchísimos, entonces los reduje por categorías más amplias, como camiones, remolques, maquinaria, y con este script es lo que compruebo. Una vez tengo establecidos los tipos de vehículo, hago un vlaue mapper, en el que mapeo los índices de código de propulsión a los valores reales del mismo, como gasolina o diesel.

El siguiente paso lo que comprueba es el campo renting. En el dataset este puede tomar valores ‘S’, ‘N’ o vacío, haciendo referencia a ‘N’. Lo que hago en este paso es buscar los campos renting vacíos, y asignarles el valor N. Después de eso hago otra limpieza más en la que compruebo el valor del atributo cod_provincia_domicilio. En caso que tome valores extranjero o desconocido, descarto esas filas. En el siguiente mapeo los códigos de las

provincias con sus nombres, para poder hacer a posterior un lookup en la dimensión de localización para que sea más manejable. Después de esto, ordeno todas las filas por el valor del bastidor, hago una comprobación de que ninguna se haya duplicado, y las inserto en la base de datos.

2.7 Hecho Trámite

Vamos a analizar ahora los pasos de la creación de la tabla de hechos trámite. En esta tabla se hará acceso a las diferentes dimensiones para conseguir los valores de acceso a las mismas.

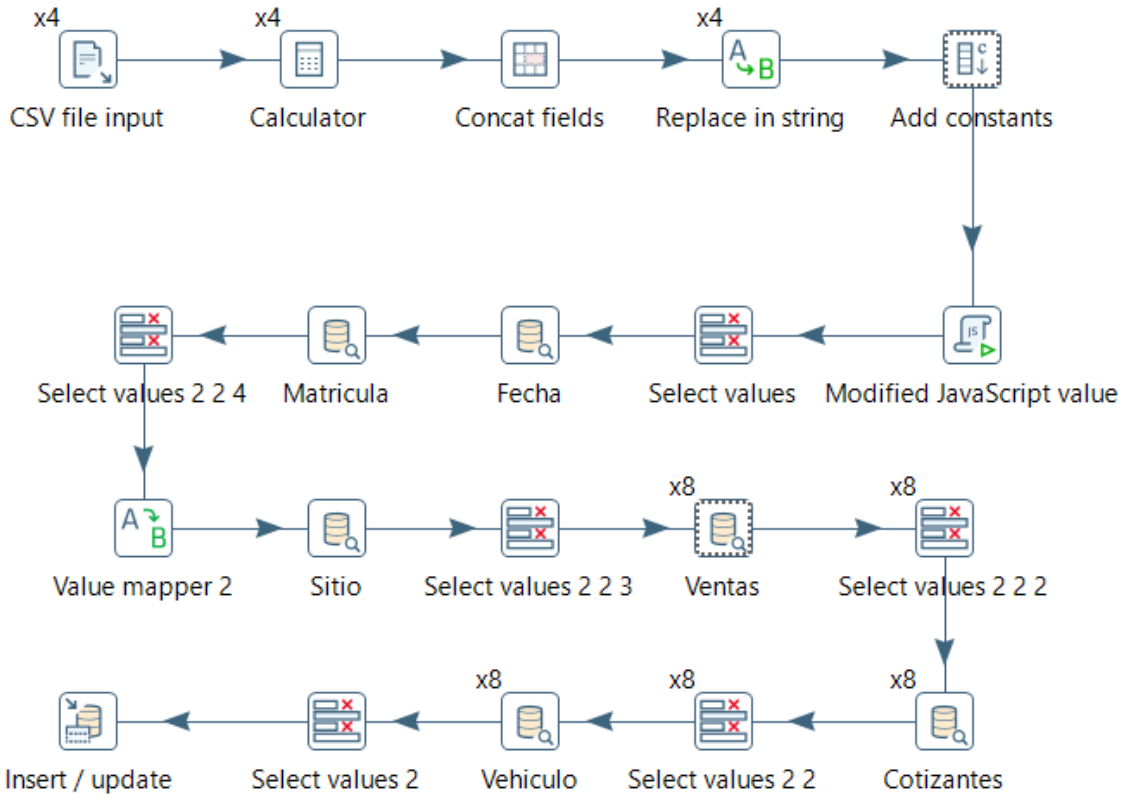

Figura 8: Transformación load_fact_tramite

En la figura 8 podemos comprobar que esta transformación carga datos de matriculaciones de vehículos en la tabla de hechos fact_tramite. Comenzamos leyendo todos los atributos del CSV de matriculaciones, que incluye bastidor, fecha matriculación, cilindrada, potencia, tara, peso máximo, plazas, etc.. Después de la carga inicial calculamos el peso neto restando tara del peso máximo con Calculator, concatenamos bastidor ITV más fecha matriculación para generar la clave compuesta fact_tramite_key del hecho, y limpiamos los indicadores binarios en Replace in string convirtiendo "SI" a "1" para cuando hay precinto en el vehículo,y para cuando hay embargo en el vehículo, y mapeando "N"/"U" del campo ind_nuevo_usado a nuevo o usado respectivamente. En el add constant añadiremos un valor count_tramite que nos servirá más adelante para las visualizaciones.

Una vez generada la clave del hecho, normalizamos la fecha matriculación con un JavaScript que reformatea fec_matricula de numérico a string con formato de fecha adecuado para la base de datos, que luego se convierte a tipo Date en Select values para los lookups dimensionales. Eliminamos bastidor temporalmente con Select values 2 para evitar duplicados en los lookups, y realizamos cinco lookups a dimensiones diferentes: Matrícula une cod_clase_matricula con dim_matricula, Fecha une fecha_matricula_norm con dim_fecha, Sitio combina provincia y municipio INE con dim_localizacion, Vehículo une bastidor con dim_vehiculo, Ventas combina provincia y fecha con dim_ventas_vehiculos, y Cotizantes une provincia con dim_cotizantes_anho.

Después de los lookups, haremos un Value mapper que traduce los códigos de provincia a su respectivo nombre para poder usarlo después en el lookup de localización. Entre todos estos lookups vamos ejecutando de manera recursiva unos select values para eliminar los atributos que ya no necesitamos y mejorar el procesamiento de la transformación. Finalmente Insert/Update carga en fact_tramite usando fact_tramite_key como clave única, recuperando las claves foráneas de las dimensiones.

2.8 Job Cargar dimensiones

Por último, vamos a comentar el funcionamiento del job que se encarga de hacer la carga de todas las dimensiones implicadas en este proyecto.

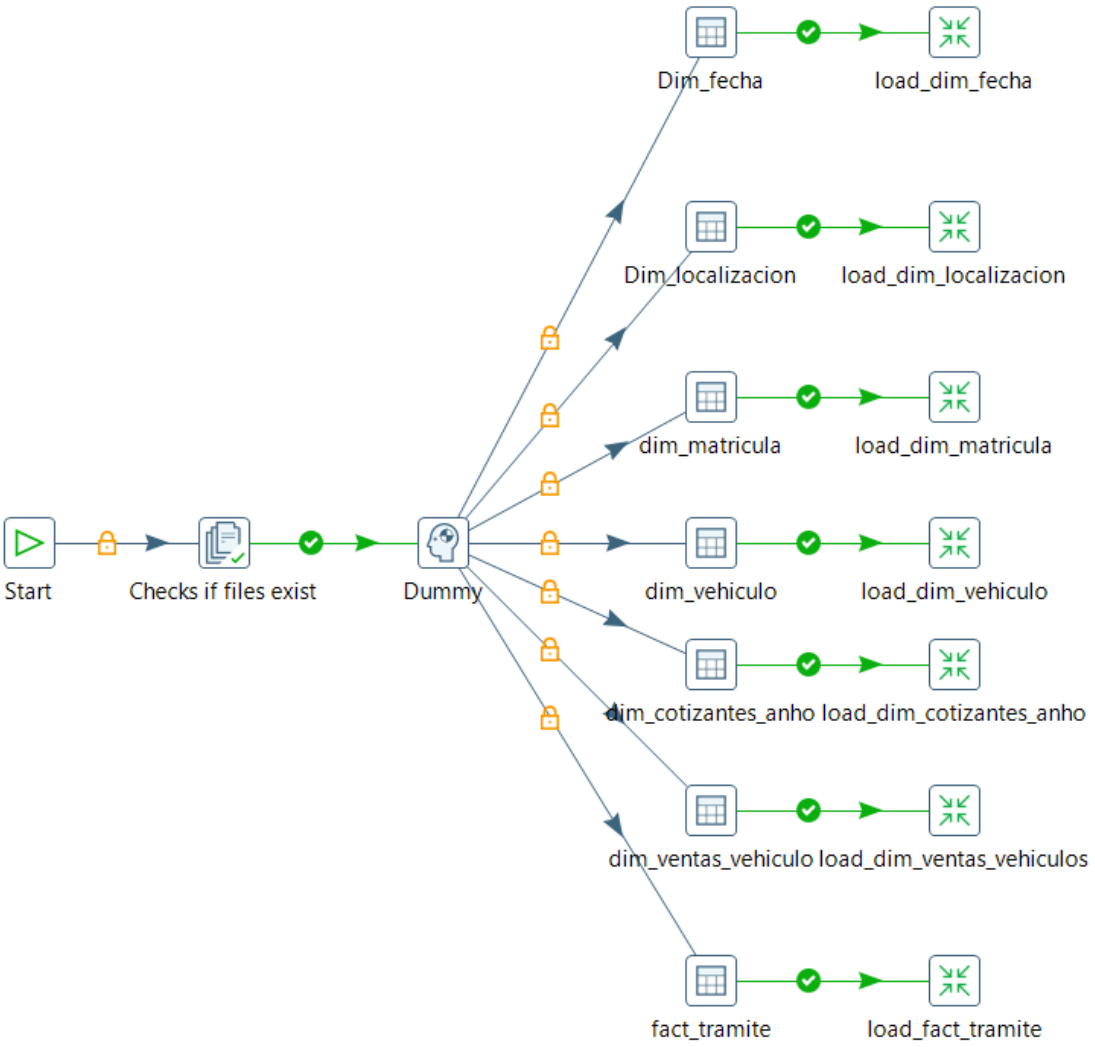

Figura 9: Job Load_dimensions

Como podemos ver en la figura 9, el job comienza en el botón inicio. De ahí, hay una comprobación, con la cual nos aseguramos que todos los csv referenciados en las transformaciones existen, y en los directorios correspondientes. De ahí vamos a un Dummy para centralizar toda la ejecución, y vamos de manera secuencial, comprobando que existe la tabla en el data warehouse, y en caso de que exista, se ejecuta la transformación asociada a la misma. Este job lo que pretende es agilizar el trabajo, y evitar el tener que ejecutarlo todo de uno en uno.

3. Representación gráfica en Tableau

Vamos a analizar ahora los diferentes dashboards que salieron a resultado de la preparación de los datos.

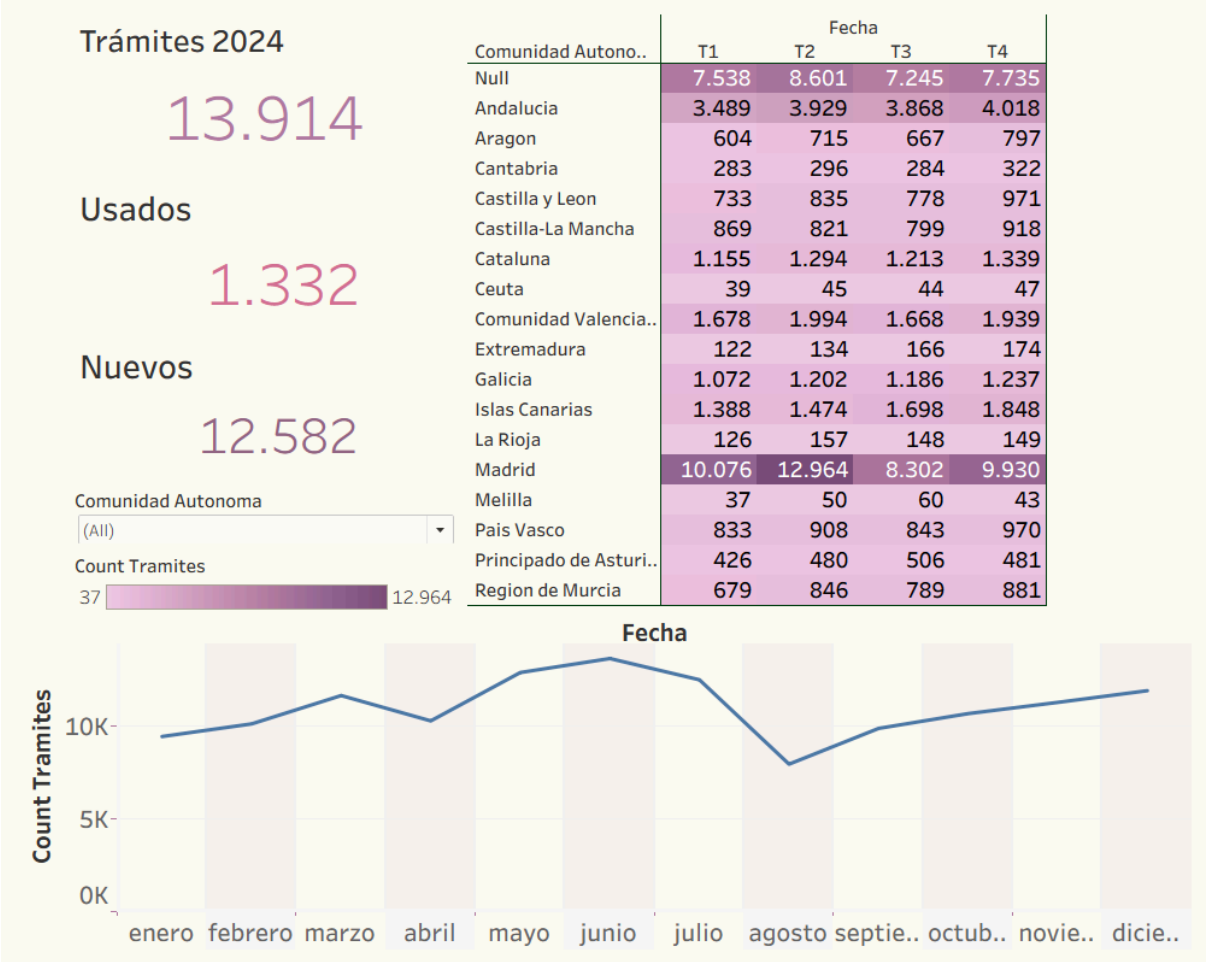

Figura 10: Dashboard 1

En la figura 10 podemos observar el dashboard principal de análisis de trámites vehiculares 2024. En primer lugar, los KPIs desglosados muestran el total de trámites realizados en 2024, clasificándolos después en vehículos usados, y vehículos nuevos, con filtro interactivo por comunidad autónoma que permite drill-down.. La tabla superior compara trámites totales por comunidades autónomas, destacando Castilla-La Mancha como líder, seguida de Aragón y Cantabria . La gráfica inferior de evolución temporal visualiza trámites por mes, revelando crecimiento estacional desde enero hasta máximo en verano, estabilizándose en otoño, lo que refleja patrones de matriculaciones post-vacaciones.

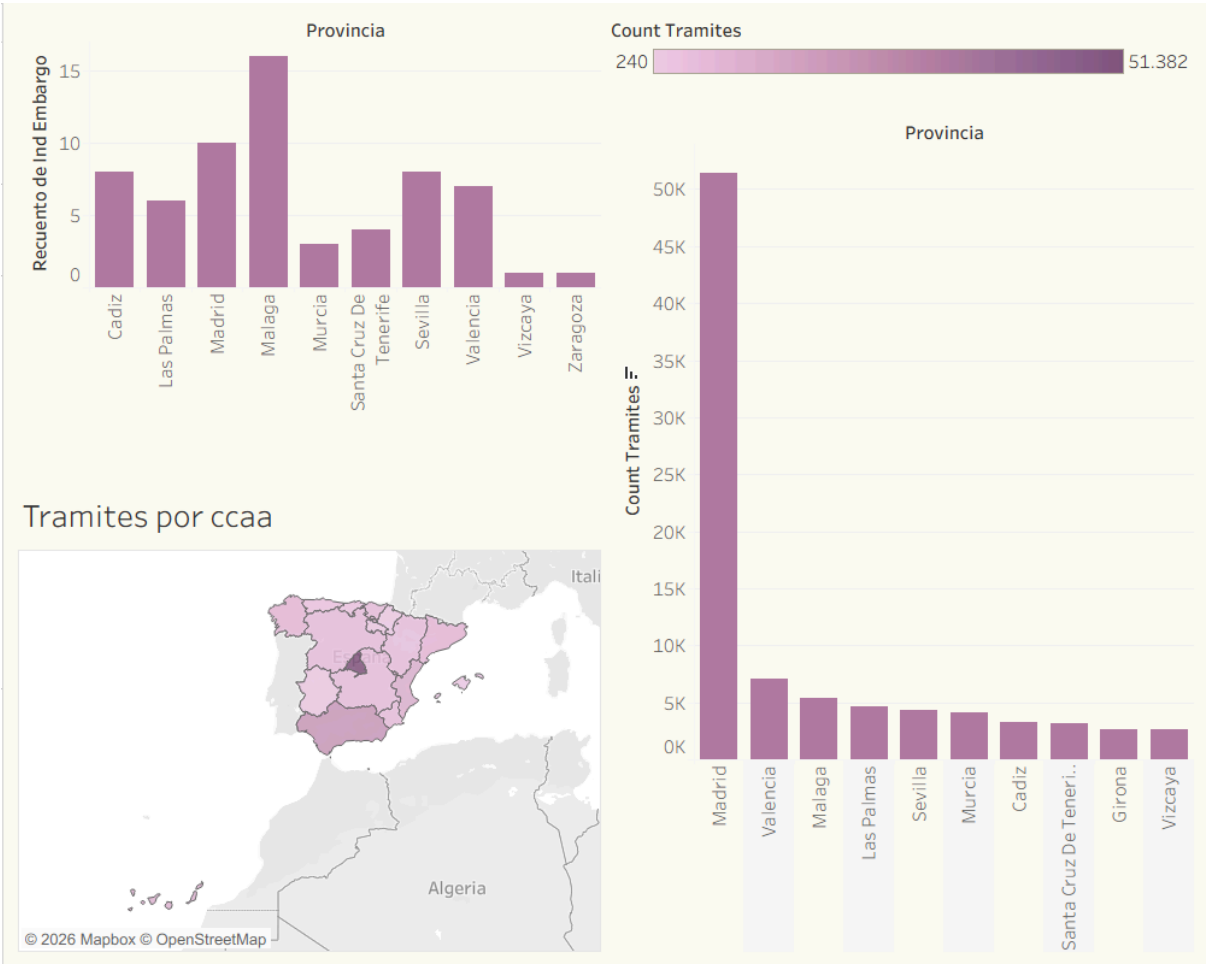

Figura 11: Dashboard 2

En la figura 11 podemos observar el dashboard principal de análisis geográfico de trámites vehiculares. En primer lugar, tenemos una gráfica de barras que nos indica la cantidad de trámites con vehículos embargados que se realizan por provincia. A la derecha, tenemos el top 10 provincias con más trámites en el intervalo 2015 a 2024. Y por último, en la esquina inferior izquierda, tenemos un mapa, que nos permite ver la distribución de trámites por comunidad autónoma, lo que además nos permite hacer un filtro geográfico a todo el dashboard.