Almacén de datos para el análisis de infraestructuras turísticas y pernoctaciones

Jonay Mora Bolaños
Grado en Ingeniería Informática

Carles Llorach Rius

16 Junio 2015

FICHA DEL TRABAJO FINAL

Título del trabajo:	Almacén de datos para el análisis de infraestructuras turísticas y pernoctaciones
Nombre del autor:	Jonay Mora Bolaños
Nombre del consultor:	Carles Llorach Rius
Fecha de entrega (mm/aaaa):	06/2015
Área del Trabajo Final:	Data warehouse
Titulación:	*Grado de Ingeniería Informática*

Resumen del Trabajo (máximo 250 palabras):

Las organizaciones de hoy en día no escatiman esfuerzos a la hora de realizar estudios de mercado antes de realizar algún tipo de inversión de mediana o gran escala. Por este motivo, el conocimiento se vuelve un bien de valor indiscutible, máxime si éste es novedoso o poco evidente a ojos de la competencia.

En este sentido, el presente trabajo de fin de grado trata de mostrar el diseño y desarrollo de un almacén de datos que permitirá posteriormente la búsqueda de conocimiento mediante el análisis multidimensional con cubos OLAP. De este modo, los analistas podrán consultar por sí mismos los datos de estudio desde varias perspectivas distintas a lo largo y ancho de todas las variables de estudio implicadas.

La dificultad de conseguir un almacén de datos funcional y que cumpla las expectativas depositadas en éste por parte de la dirección encargada de aprobar el proyecto, reside principalmente en la elaboración meticulosa de las tablas que contendrán los datos. No siempre se dispondrá de fuentes de datos fiables, ni tendrán el nivel de detalle que los analistas quisieran, así que el mayor o menor éxito del producto final depende principalmente de la fase de extracción, transformación y carga de datos (ETL) al futuro almacén.

La mayor parte de este trabajo estará enfocado principalmente a la etapa de adecuación de los datos y, en menor medida (pero no por ello menos importante), a la implementación de los esquemas que permitirán el acceso al almacén de datos.

Abstract (in English, 250 words or less):

Nowadays organizations spare no effort when it comes to a market research before making any attempt on a medium or large scale investment. Therefore, knowledge becomes an invaluable good, especially if it is new or little evident at competitor's eyes.

In this sense, the present final project aims to show the design and development of a data warehouse that will later allow seeking knowledge through multidimensional OLAP analysis cubes. This way, analysts can query for themselves the study data from several perspectives throughout all the key involved variables.

The difficulty of getting a functional data warehouse that meets the expectations placed on it by the directorate responsible for approving the project, lies partly in the meticulous development of the tables that will contain the data. Not always will be available sources of reliable data, nor will have the level of detail that analysts wish, so the degree of success of the final product depends mainly on the extraction, transformation and loading (ETL) stage of the future data warehouse.

Most of this paper will be focused mainly on the fitting data stage and to a lesser extent (but no less important), to the implementation of the schemes that will allow access to the data warehouse.

Palabras clave (entre 4 y 8):

Data warehouse, Extract-Transform-Load, OLAP, Dimensions, Measures, Fact Tables, Knowledge

Índice

Lista de figuras

1 Introducción

1.1 Contexto y justificación del Trabajo

La organización Grupo Líder en Turismo Familiar (GLTF) ha detectado una evolución negativa de las pernoctaciones en Cataluña, razón por la cual ha decidido invertir esfuerzos para detectar las variables claves que expliquen este fenómeno.

Hasta ahora, GLTF trataba de encontrar algún tipo de correlación entre el nivel de los equipamientos de los municipios, los tipos de establecimientos turísticos así como las plazas disponibles de los mismos y la percepción de seguridad en la zona (número de efectivos policiales). Si bien tienen a su disposición datos del INE e IDESCAT en formato de hojas de cálculo, lo cierto es que han tenido dificultades a la hora de realizar cálculos estadísticos por municipios o niveles superiores de delimitación territorial, por no mencionar que los datos más importantes (las pernoctaciones) sólo están disponibles a nivel de zona turística y no a nivel municipio.

Además, al querer incorporar la variable equipamientos al estudio, GLTF se ha dado cuenta que no tiene una estructura de datos que permita siquiera el desarrollo de cálculos simples (como de cuántos equipamientos deportivos consta un municipio, etc.).

Con todos estos factores en mente, se cae en la cuenta que es necesario un sistema capaz de almacenar todas estas variables y que permita contrastarlas unas con otras en un tiempo de respuesta aceptable.

La solución a esta necesidad pasa por desarrollar un almacén de datos, el cual permitirá el procesamiento analítico en línea (OLAP) de una gran cantidad de datos con el mejor tiempo de respuesta posible.

1.2 Objetivos del Trabajo

En principio, el enfoque del presente TFG está orientado a la satisfacción de los requerimientos de un hipotético cliente (la ya mencionada organización GLTF). Sin embargo, dada la naturaleza académica del TFG, es importante también tener en cuenta los objetivos inherentes al mismo. Así, tenemos por un lado:

1.2.1 Objetivos de la asignatura

Implica poner en práctica los métodos y, en general, los conocimientos adquiridos a lo largo de la carrera:

- Analizar y entender el alcance de las propuestas del enunciado.

- Planificar una estructura de distribución de trabajo (EDT) que se adecúe a los hitos propuestos en el plan de docencia y al alcance previamente analizado.
- Adquirir los conocimientos necesarios para entender la terminología de Data WareHouse y crear un entregable acorde al área en que se desarrolla.
- Adquirir los conocimientos necesarios para el manejo del software propuesto para el presente proyecto.
- Implementar una solución que se adecúe a los objetivos propuestos por el enunciado específico de este proyecto.
- Elaborar la presente memoria del proyecto.
- Realizar una presentación del desarrollo del trabajo y del producto final.

1.2.2 Objetivos generales

Son las líneas de trabajo principales de lo que ha de desarrollarse:

- Analizar las fuentes proporcionadas.
- Realizar el diseño (multidimensional) conceptual, lógico y físico del almacén de datos.
- Crear los procedimientos ETL necesarios para cargar los datos al almacén.
- Implementar los esquemas OLAP para el repositorio de la herramienta Bussiness Intelligence con la que han de trabajar los usuarios.
- Implementar las consultas que solicita GLTF.

1.2.3 Objetivos específicos

Son los requerimientos solicitados por GLTF. Esto es, el sistema implementado tiene que permitir la realización de las siguientes consultas:

- Total de pernoctaciones estimadas por municipio.
- Promedio de viajeros por tipo de establecimiento y comarca.
- % de ocupación por comarca turística.
- Ranking de municipios por categoría de equipamientos.
- Máximo y mínimo de efectivos policiales por tipología de establecimiento y municipio.
- “Top ten” de municipios por franja de pernoctaciones.
- % de pernoctaciones de un municipio sobre el total.
- Categorización de municipios A / B / C.
- Ratio de policías locales por habitante.

- Distribución mensual y estacionalidad de las pernoctaciones.

Notas adicionales:

- La información de los anteriores puntos debe proporcionarse por temporalidad a nivel de mes y año.
- Las consultas pueden ser agregadas, por demarcación territorial y tipo de establecimiento.

1.3 Enfoque y método seguido

Entre las metodologías de desarrollo de software, se pueden citar como más importantes las siguientes:

- Modelo en cascada.
- Prototipado.
- Incremental.
- Espiral.
- Desarrollo rápido de aplicaciones (RAD).

Sin entrar en mucho detalle sobre los principios básicos de cada metodología, se puede decir que el presente proyecto tiene un enfoque que concuerda más con el modelo en cascada.

No se trata, en este caso, de cuál es la estrategia más apropiada, sino de cuál está autoimpuesta por la naturaleza académica de este trabajo. Dicho de otro modo, la UOC ha establecido una serie de hitos que han de cumplirse y que son muy específicos para poder seguir la evolución del trabajo sin desviaciones. Si atendemos a los principios básicos que encontramos en la Wikipedia:

- *El proyecto está dividido en fases secuenciales, con cierta superposición y splashback aceptable entre fases.*
- *Se hace hincapié en la planificación, los horarios, fechas, presupuestos y ejecución de todo un sistema de una sola vez.*
- *Un estricto control se mantiene durante la vida del proyecto a través de la utilización de una amplia documentación escrita, así como a través de comentarios y aprobación / signoff por el usuario y la tecnología de la información de gestión al final de la mayoría de las fases antes de comenzar la próxima fase.*

Como se puede comprobar, las mencionadas fases o hitos corresponden en la UOC con las entregas parciales denominadas PECs. Los hitos y el contenido requerido está estipulado desde un principio, corriendo a cargo del alumno la propuesta de planificación para llevarla a cabo.

Por otro lado, el control se lleva a cabo precisamente con las PECs, que contienen una amplia documentación escrita para facilitar el seguimiento y verificar el cumplimiento de los puntos propuestos.

Por último, conviene entender algunos aspectos claves como son la nula experiencia en el área, el desarrollo de un producto desde cero y la necesidad de tener un seguimiento por parte del consultor del área TFG. En otras palabras, es necesaria la documentación exhaustiva y cumplir los plazos de entrega para dar tiempo a la revisión y a la posterior corrección de los fallos o posibles mejoras, por lo que el desarrollo en cascada sale de forma natural en este proceso.

1.4 Planificación del Trabajo

Según el plan docente del TFG, el trabajo se divide principalmente en 4 fases con sus respectivos entregables: “plan de trabajo”, “análisis y diseño”, “implementación” y “memoria y presentación virtual”.

Si bien es cierto que la estructura es la misma, hay ligeras modificaciones respecto de la segunda y tercera etapa si se comparan con el plan docente. Los hitos de las cuatro etapas tienen su correspondencia con el sistema de evaluación continua de la UOC, que se expondrá a continuación:

1.4.1 PEC1: Plan de trabajo

Lo que se pide es realizar un análisis preliminar de las fuentes, entender los objetivos perseguidos y, teniendo en cuenta ambos aspectos, realizar un modelo aproximado del futuro almacén de datos, atendiendo a las características más básicas del mismo.

Finalmente, teniendo en constancia el boceto general del modelo, se ha de crear un plan de trabajo que recoja todas las tareas detectadas así como el esfuerzo en días previsto (se usa para ello un diagrama de Gantt).

Entregable (11/03/2015): documento del plan de trabajo.

1.4.2 PEC2: Análisis y diseño

Hay tres puntos importantes que deben tenerse en cuenta en esta etapa:

- Hay que realizar un análisis detallado de las fuentes de datos.
- Hay que desarrollar el diseño multidimensional (conceptual y físico).
- Hay que diseñar los procesos de extracción, transformación y carga (ETL) para las fuentes proporcionadas (carga inicial).

Entregable (15/04/2015): documento de análisis y diseño.

1.4.3 PEC3: Implementación

Esta etapa se supone que es la concreción de la etapa anterior, esto es:

- La creación de la base de datos (tablas, índices, constraints, etc.) del almacén de datos.

- La construcción de los procesos de carga (ETL) y la carga (y propuesta de automatización) de los datos en el data warehouse.
- La creación de un conjunto de informes que se consideren interesantes según los indicadores solicitados.

Lo cierto es que, salvo los informes, el diseño de la base de datos y de los procesos ETL ya se realizó en la anterior fase, así que en esta etapa se podría identificar como tareas realizadas:

- Corregir errores detectados de los procesos implementados en la PEC2.
- La reconstrucción del almacén de datsos adaptándolo a los requerimientos solicitados.
- Los procesos de carga ETL necesarios para el nuevo modelo del almacén de datos.
- La implementación de los informes pedidos por GLTF.

Obviamente, esta parte no estaba planificada, sin embargo no hay desviación respecto del plan principal, sino una posibilidad de realizar una iteración en la etapa de diseño e implementación que ha permitido mejorar sustancialmente los resultados del trabajo.

Entregables (28/05/2015):

- Documento explicativo sobre cómo acceder a la base de datos del almacén de datos creado (incluidos usuarios y passwords para acceder a los esquemas creados) y los informes vía navegador (usuario y contraseña).
- Documento explicativo de cómo se ha llevado a cabo el trabajo, el funcionamiento de los procesos de carga y recomendaciones relevantes que se consideren oportunos, así como la justificación del cumplimiento de los requisitos funcionales.
- Documento con copia de pantalla de todos los informes que se han creado.

1.4.4 Memoria

Es la recta final del proyecto y consiste en los siguientes entregables:

- La memoria: el presente documento.
- Presentación virtual: en formato de vídeo, que deberá aportar una visión general del TFG y que debe permitir al Tribunal de evaluación formular las preguntas oportunas al autor del trabajo. Los límites de la presentación son: duración inferior a 20 minutos y espacio de almacenamiento inferior a 80Mb.
- El producto final: que estará instalado en la máquina virtual y debe ofrecer la versión definitiva del almacén de datos así como los accesos a la Suite BI de Pentaho para evaluar los informes.

1.4.5 Tabla de tareas y diagrama de Gantt

Ésta fue la planificación inicial propuesta en la PEC1 y su correspondiente diagrama de Gantt:

Nombre de tarea	Duración	Comienzo	Fin
Descarga y lectura módulos DW	**78 días**	**vie 27/02/15**	**mar 16/06/15**
PEC1	**9 días**	**vie 27/02/15**	**mié 11/03/15**
Lectura Plan docente	1 día	dom 01/03/15	dom 01/03/15
Análisis y arreglos archivos datos	6 días	dom 01/03/15	vie 06/03/15
Aprendizaje básico conceptos DW	4 días	mié 04/03/15	lun 09/03/15
Elaboración de la PEC1	3 días	lun 09/03/15	mié 11/03/15
Entrega PEC1	0 días	mié 11/03/15	mié 11/03/15
PEC2	**25 días**	**jue 12/03/15**	**mié 15/04/15**
Planteamiento de dudas PEC1	3 días	jue 12/03/15	lun 16/03/15
Lectura y comprensión PEC2	2 días	jue 12/03/15	vie 13/03/15
Familiarización entorno Amazon	6 días	lun 16/03/15	lun 23/03/15
Corrección PEC1	1 día	lun 23/03/15	lun 23/03/15
Análisis de requerimientos	3 días	mar 24/03/15	jue 26/03/15
Realización modelo de datos	3 días	vie 27/03/15	mar 31/03/15
Diseño conceptual	3 días	mié 01/04/15	vie 03/04/15
Diseño técnico	5 días	lun 06/04/15	vie 10/04/15
Revisión documentación	1 día	lun 13/04/15	lun 13/04/15
Documentación PEC2	2 días	mar 14/04/15	mié 15/04/15
Entrega PEC2	0 días	mié 15/04/15	mié 15/04/15
PEC3	**31 días**	**jue 16/04/15**	**jue 28/05/15**
Creación base de datos	3 días	jue 16/04/15	lun 20/04/15
Corrección definitiva y carga de datos	5 días	mar 21/04/15	lun 27/04/15
Correción PEC2	2 días	lun 27/04/15	mar 28/04/15
Familiarización Pentaho Business Analytics	2 días	mar 28/04/15	mié 29/04/15
Pruebas con herramientas	3 días	jue 30/04/15	lun 04/05/15
Análisis de información	2 días	mar 05/05/15	mié 06/05/15
Implementación del sistema	10 días	jue 07/05/15	mié 20/05/15
Construcción y realización de informes	4 días	jue 21/05/15	mar 26/05/15
Análisis de resultados	1 día	mié 27/05/15	mié 27/05/15
Finalizar documentación PEC3	1 día	jue 28/05/15	jue 28/05/15
Entrega PEC3	0 días	jue 28/05/15	jue 28/05/15
Memoria	**13 días**	**vie 29/05/15**	**mar 16/06/15**
Conclusiones	3 días	vie 29/05/15	mar 02/06/15
Revisión previa	4 días	mié 03/06/15	lun 08/06/15
Corrección PEC3	1 día	lun 08/06/15	lun 08/06/15
Revisión final	4 días	mar 09/06/15	vie 12/06/15
Síntesis	2 días	lun 15/06/15	mar 16/06/15

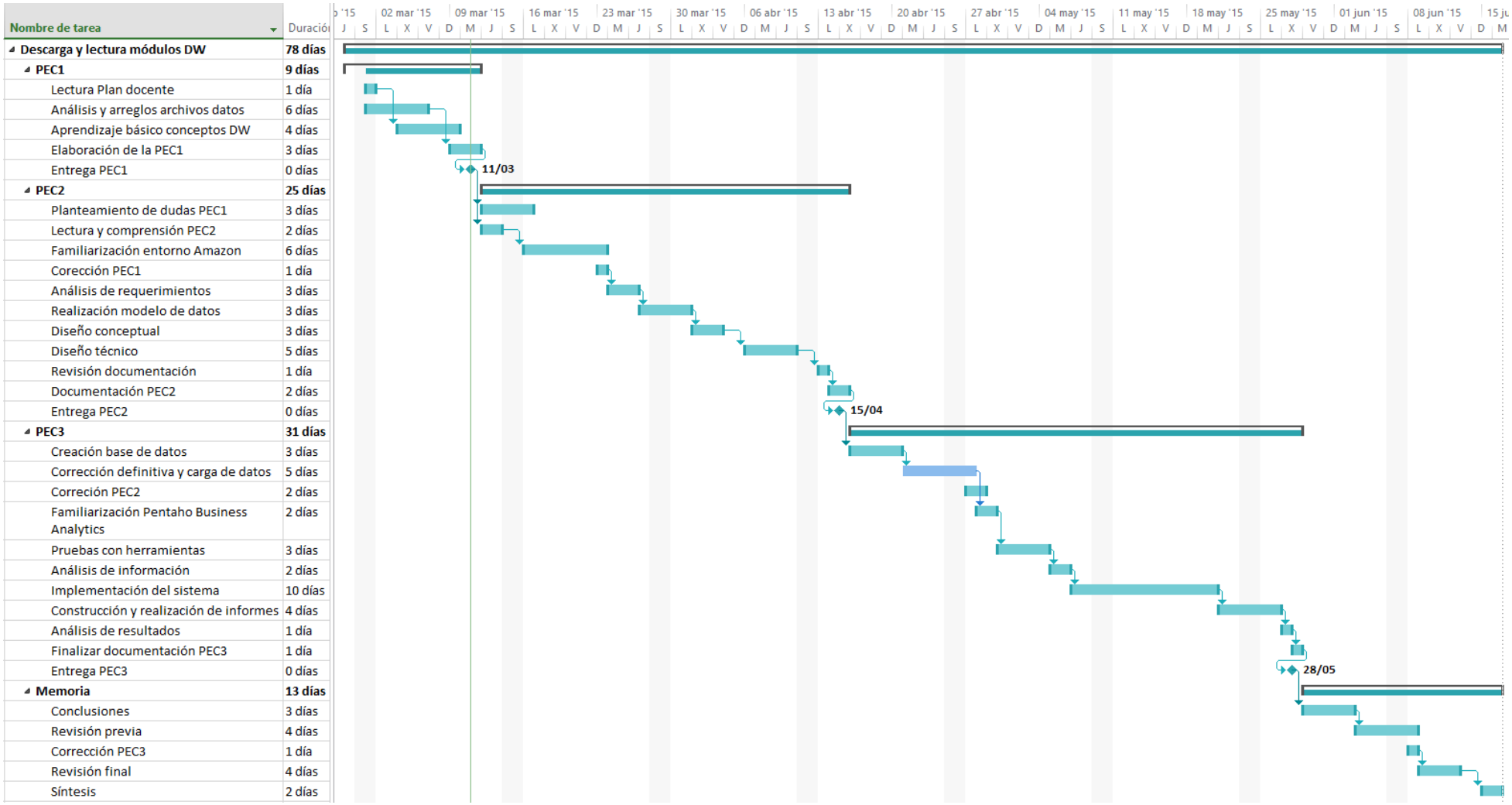


Ilustración 1 - Diagrama de Gantt

1.5 Breve sumario de productos obtenidos

El producto final consta de una serie de archivos estructurados en varias carpetas, incluyendo las fuentes de datos:

- Carpeta raíz **TFG_final** (C:\Users\Administrator\Documents\TFG_final)
    - Archivos Job que ejecutan secuencialmente todos los procesos ETL para la carga inicial y las actualizaciones.
    - Archivos log que muestran algunos estados intermedios del proceso ETL así como listas de elementos duplicados de equipamientos.
  - ETL_init:
    - Archivos ETL para la carga inicial.
    - Archivo de definición de datos (DDL) para la creación de la base de datos del almacén.
    - Archivo de lenguaje de manipulación de datos (DML) con los pesos para equipamientos.
  - ETL_update:
    - Archivos ETL para las actualizaciones anuales.
  - Reports: archivo de esquema Mondrian para la definición de los cubos OLAP.
  - CargaInicial:
    - Archivos CSV y hojas de cálculo suministrados por GLTF (cuya temporalidad abarca los años 2011 a 2013).
    - Municipios-geolocalizados: archivo SQL como fuente externa que se usará como apoyo en este trabajo.
  - Actualizacion: archivos en formato CSV de las mismas fuentes que la carga inicial (INE e IDESCAT) con los datos parciales del año 2014 (a modo de ejemplo).

Nota: la carpeta TFG_final puede renombrarse o cambiar de ubicación. La herramienta ETL (Kettle) permite definir variables de entorno como la ubicación relativa de los archivos, así que que todo seguirá funcionando mientras no se cambie la estructura interna de la solución propuesta.

1.6 Breve descripción de los capítulos de la memoria

A continuación, se pasa a explicar en líneas generales que es lo que se encontrará a partir de este punto:

- Recursos técnicos: explica de qué material se dispone para llevar a cabo el presente trabajo.
- Análisis y diseño: en el se encuentra todo el procedimiento que fue necesario llevar a cabo para crear el almacén de datos, pasando del análisis detallado de las fuentes para proceder a la carga inicial, al análisis conceptual que dará forma al modelo del sistema para, finalmente, proceder al diseño lógico y físico del almacén. Especial mención al primer apartado de este capítulo, que muestra cómo será la arquitectura de la solución propuesta.
- Procedimientos ETL: son las operaciones necesarias para cargar datos en el almacén a partir de las fuentes de datos. Los dos principales apartados de esta sección son la carga inicial y las actualizaciones anuales.

  La gran mayoría de los detalles, problemas, decisiones y otros aspectos de menor o mayor importancia encontrados en las etapas previas de análisis y diseño, así como a lo largo de la ejecución del proyecto están reflejadas en los esquemas ETL. Es por este motivo que se trata de la sección más "pesada" del trabajo y la razón por la que ha sido imposible poder resumirla más de lo que ya está.
- Los cubos OLAP: esta sección está dedicada a explicar qué son y cómo se definen los cubos OLAP en esta arquitectura, las herramientas que han de usarse y los resultados que GLTF espera encontrar con la explotación de este almacén de datos.

2 Recursos técnicos

Los recursos con los que cuenta el presente trabajo son los siguientes:

2.1 Recursos Hardware

Se cuenta principalmente con la computadora del autor y una máquina virtual ubicada en la red Amazon EC2:

- Máquina personal: con sistema operativo Windows 8.1 64 bits.
- Máquina virtual: con sistema operativo Windows Server 2012 64 bits. Equipo AMD 64 con una cuota de memoria de 3,75Gb y espacio de disco duro de 30 Gb (aproximadamente 3,2Gb libres para el desarrollo del TFG).

2.2 Recursos Software

Nuevamente se dividirá entre recursos personales y los ofertados por la UOC a través de la máquina virtual:

- Máquina personal
    - Procesador de textos: Microsoft Word.
    - Hojas de cálculo: Libre Office Calc.
    - Edición de imagen: GIMP.
    - Editor de textos: Geany.
    - Planificación: Microsoft Project.
    - Servicio de escritorio remoto: Conexión a escritorio remoto (accesorio Windows).
    - Google Drive: herramienta Drawing.
- Máquina virtual
    - Base de datos: MySQL Server.
    - Herramienta visual de BBDD MySQL: MySQL Workbench.
    - Suite BI: Pentaho Business Analytics.
        - Herramienta de ETL: Spoon-PDI (o Kettle).
        - Portal BI: Pentaho BI server (requiere navegador).
        - Motor MOLAP: Mondrian.
        - Herramienta de informes: Report Designer.
        - Análisis OLAP: Saiku (como plugin de Pentaho BI server).
        - Herramienta esquemas Mondrian: Schema Workbench.

Nota: el producto final se ha de desarrollar en la máquina virtual, por lo que todo diagrama que explique la arquitectura del producto final se referirá a ésta.

2.3 Recursos humanos

En principio, sólo es necesario el trabajo de una persona (el presente autor) que deberá asumir todos los roles posibles para llevar a cabo este trabajo, desde la gestión de proyectos a analista, arquitecto, técnico de pruebas…

Es importante destacar además la figura del consultor, que asumirá los roles de demandante del servicio (representando los intereses de GLTF) y de consultor propiamente dicho, permitiendo la resolución de dudas y propuestas de mejora.

3 Análisis y diseño

En este apartado se pretende esbozar gráficamente en qué consistirá la solución que busca GLTF, las partes claves del mismo y los pasos necesarios para su consecución.

3.1 Arquitectura del sistema

El diagrama que se muestra a continuación muestra la secuencia de pasos que se ha de seguir para abordar la problemática planteada por GLTF:

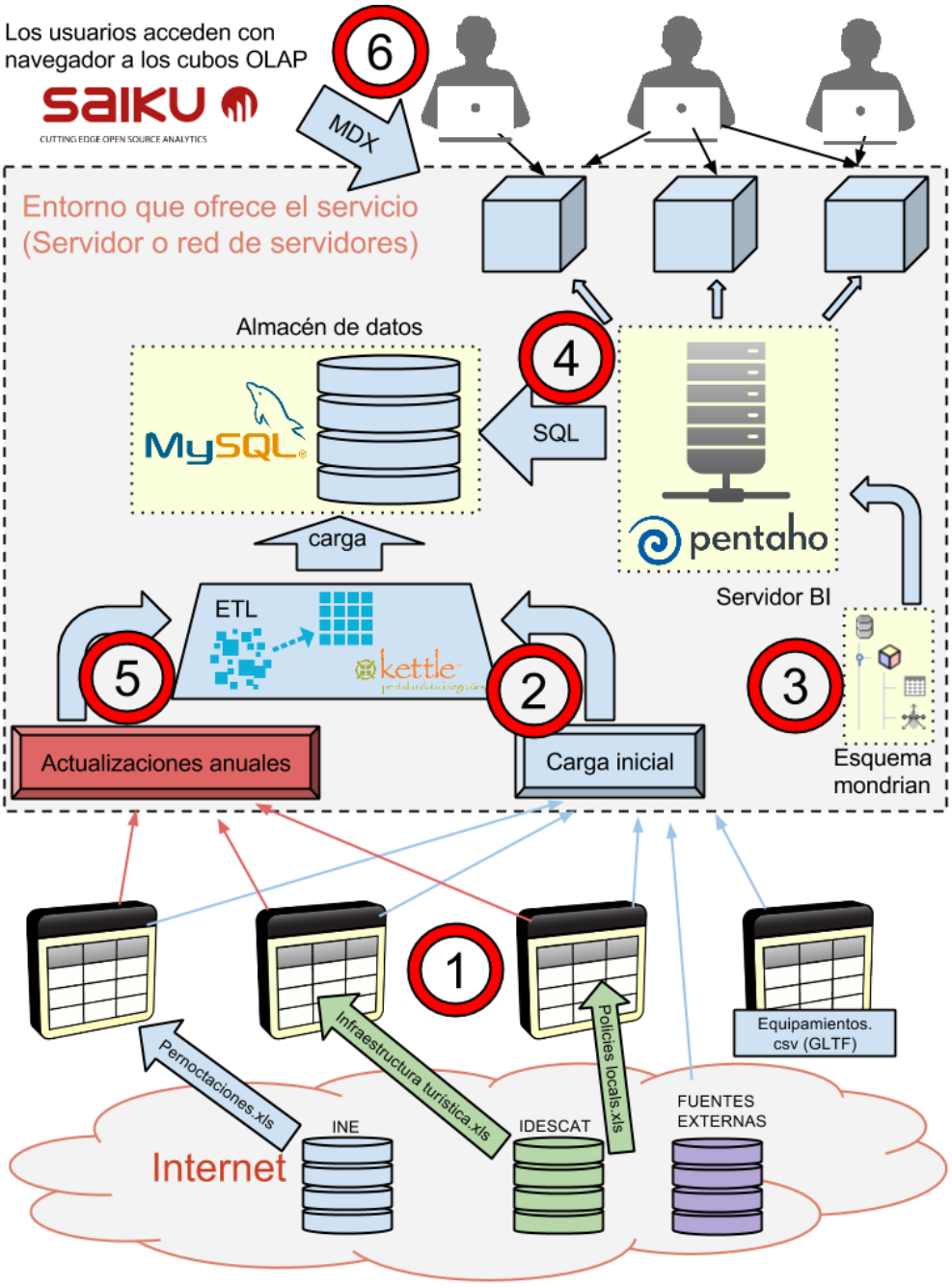


Ilustración 2 - Arquitectura del sistema

1. En primer lugar de lo único que se dispone son de las fuentes de datos accesibles desde Internet a través de las páginas web del INE y del IDESCAT. Todos los datos (a excepción de los equipamientos) tienen una temporalidad comprendida entre los años 2011 y 2013:
    a. Pernoctaciones.xls: proporcionado como hoja de cálculo, contiene los datos más importantes objetivos del estudio, las pernoctaciones y viajeros por zona turística de Cataluña y, a su vez, divididos en residentes españoles y residentes en el extranjero (fuente INE).
    b. Infraestructuras turísticas: otra hoja de cálculo, esta vez contiene los datos de los establecimientos hoteleros, campings y casas rurales de Cataluña. Para cada tipo de establecimiento y para cada año, existen dos tablas (cada una en una hoja) que contienen el número de establecimientos de cada subtipo (hoteles de 1, 2, 3 estrellas…) y el número de plazas total para cada subtipo de establecimiento en cada municipio de Cataluña (fuentes IDESCAT).
    c. Policies locals: una hoja de cálculo que contiene tres hojas (una por año) que lista, para cada municipio de Cataluña, el número de efectivos policiales por rango (fuente IDESCAT).
    d. Equipamientos.csv: se trata de un archivo que contiene un listado de equipamientos por municipio. El formato del archivo es bastante conflictivo y requiere un tratamiento intensivo para poder extraer con éxito la información que contiene (fuente GLTF).
    e. Población csv: no se muestra en este diagrama por ser una fuente posterior que tuvo que incluirse en el proyecto para realizar algunos cálculos. Se trata de una tabla de los municipios de Cataluña y el número de habitantes de estos en un año concreto (fuente IDESCAT).
2. La carga inicial: previamente ha de realizarse un análisis detallado de las fuentes de datos, seguidamente un análisis de diseño conceptual y físico para obtener finalmente la estructura necesaria del almacén de datos. Una vez cumplida esta etapa, se procede con la parte clave de este trabajo, que son los procesos ETL que permitirán la carga inicial de los datos al almacén.

   Los procesos ETL tienen como entrada las fuentes del paso 1 y como salida el almacén de datos, que es una base de datos relacional de MySQL. Los procesos ETL se desarrollan con la herramienta Spoon-PDI (o también conocida como Kettle) y que se explicarán más adelante. Conviene tener en cuenta que el proceso de carga inicial consiste en un proceso de escritura masiva que sólo se prevé realizar una sola vez. **Una vez termine la carga inicial, los datos no deben ser actualizados o borrados bajo ninguna circunstancia**.
3. El esquema Mondrian: la herramienta analítica que requiere GLTF se encuentra alojada en el servidor Business Intelligence de Pentaho. Sin

embargo, el almacén de datos se encuentra alojado en el SGBD MySQL. Aquí es donde entra en escena el esquema Mondrian, que no es sino un archivo XML que explica al motor Mondrian alojado en Pentaho cómo ha de leer los datos que se encuentran alojados en el almacén de datos para poder construir los cubos OLAP.

4. Una vez que el esquema Mondrian esté alojado en los repositorios de Pentaho BI y que tengamos los datos de conectividad a la base de datos MySQL, el motor Mondrian de Pentaho podrá acceder al almacén mediante consultas en el lenguaje SQL (previa consulta MDX).

   Es importante darse cuenta en este momento el porqué de la existencia de un almacén de datos con su característica estructura desnormalizada de datos, y es que MDX es un lenguaje de consulta que sólo usa sentencias SELECT. Es decir, **no se modifican datos, sólo se realizan operaciones de lectura**. Este aspecto es clave para entender el enfoque de la solución.

5. Adicionalmente, existe la posibilidad de realizar actualizaciones anuales del almacén de datos, que implica tener las fuentes de datos del paso 1 (a excepción de los equipamientos, cuyo carácter es atemporal) para un año determinado. Necesariamente han de ser tratados por procedimientos ETL específicos para este caso y su carga al almacén supone simplemente agregar a los datos ya existentes los nuevos.

   Este "módulo" adicional permitirá la escalabilidad de la historia de datos de Cataluña. Nótese que existen datos del INE disponibles a partir del año 1999.

6. Por último, el objetivo principal de este trabajo, que los usuarios analistas de GLTF puedan acceder a los cubos OLAP confeccionados por Pentaho BI a través del plugin de consulta MDX conocido como Saiku.

3.2 Análisis detallado de los datos

El primer paso que se puede observar en la ilustración 2 consiste en recopilar las fuentes, ahora bien, para poder proceder al siguiente paso, es necesario entender cómo están estructurados los datos, qué tipo de formato tienen y qué tipo de tratamiento han de seguir en caso de que se detecte una baja calidad por parte de estos.

A continuación se expondrá de forma resumida las características principales encontradas en cada archivo fuente sin entrar en todos los detalles, ya que será en los propios procesos ETL cuando se podrá ver todas y cada una de las incidencias detectadas.

3.2.1 Equipamientos.csv

Ésta es la fuente más problemática encontrada en el presente trabajo. Se trata de un archivo de datos separados por comas y con delimitadores de dobles comillas dobles ("") que no siempre cumple este formato. Es decir, se van a

encontrar datos que incluyen alguna coma en su interior y que no representa un separador de campo; esto es debido a que han omitido accidental o intencionadamente los delimitadores de campo (“”). Sin contar con los errores detectados, las características estructurales del archivo son las siguientes:

Atributo	Tipo	Rango / Tamaño
nombre	String	¿?
direccion	String	¿?
municipio	String	43 caracteres
cp	String	5 caracteres (hay nulos)
comarca	String	17 caracteres
telefono	String	12 caracteres (hay nulos)
longitud	Decimal	10 cifras 8 decimales
latitud	Decimal	10 cifras 8 decimales
categorias	String	368 caracteres
location	String	¿?

Y ahora, las principales incidencias detectadas:

- Los atributos nombre y direccion no siguen un patrón común: algunos datos fueron insertados sin delimitadores y otros usan caracteres que pueden crear conflictos (“;”). Lo cierto es que se trata de un dato prescindible de cara al almacén, ya que sólo interesa el dato estadístico “número de equipamientos de determinada categoría en un municipio”. Sin embargo, ambos datos ayudarán a resolver otro tipo de problemas como son las tuplas duplicadas, por lo que se hacen necesarios de cara a la detección de éstas. La solución tomada a este respecto es considerar ambos atributos como un único campo “nombreDireccion” que permita posteriores análisis manuales para establecer reglas que ayudarán filtrar las tuplas duplicadas.
- En el código postal (“cp”) hay que tener en cuenta que hay valores nulos que se indican con un guión (“-“).
- Los atributos longitud y latitud son campos prescindibles, aún así conviene usarlos para el filtrado de posibles tuplas duplicadas (por ejemplo, dos equipamientos con parecido nombreDireccion y misma longitud y latitud podrían tratarse del mismo equipamiento). No obstante existen valores erróneos muy alejados del rango de geoposicionamiento de Cataluña. Esta incidencia se puede tratar (y de hecho así se hizo), aunque no es necesario.
- Categorias: es un atributo clave en este trabajo, así que su tratamiento es prioritario. Se trata de elementos separados por el carácter “pipe” (“|”) y que siguen un patrón claro:

  “categoría_1 | categoría_2 | categoría_3 | categoría_3 …”

Esto es, cada equipamiento tiene (o debería tener) dos categorías antecesoras (niveles 1 y 2). En una misma tupla se pueden encontrar uno o más equipamientos de nivel 3 que tienen en común las dos primeras categorías. No se ha detectado ningún caso de categorías de nivel 1 y 2 múltiples con combinaciones de categoría 3 para una misma tupla (esto se supo una vez se pudo aplicar el proceso ETL).

Un aspecto destacado de este campo es que siempre empieza con la cadena de caracteres "Equipaments | " el cual es omitido por no aportar información alguna. Además, hubo que corregir algunas categorías que incluían espacios entre palabras (se supone que todas deben tener guión bajo "_" en vez de espacios).

- El campo Location (que es un derivado XML de los atributos longitud y latitud) se elimina por no aportar información relevante.

Respecto del atributo "nombreDireccion", se prevé que tendrá una longtitud máxima de 192 caracteres. Su extracción se verá facilitada enormemente por la correcta estructura del lado derecho de cada tupla, de modo que es posible crear expresiones regulares que permitan extraer la combinación de ambos atributos.

3.2.2 Infraestructura turística.xls

Se trata de un libro con 18 hojas de cálculo (más una de metadatos) donde cada hoja representa una tabla de datos de infraestructura turística. Realmente son tablas de 3 años (del 2011 al 2013), por lo que se puede simplificar la estructura a 6 hojas (o tablas) por año. Hay principalmente tres tipos de establecimientos: hoteles, campings y casas rurales, y de cada tipo de establecimiento hay dos tablas, una que indica la cantidad de establecimientos y otra que indica el número de plazas de cada establecimiento (de modo que hay 2 * 3 = 6 tablas u hojas de cálculo por año). Cada fila de la tabla corresponde a los datos de un municipio. Resumiendo:

Hoteles		Campings		Casas rurales	
Cantidad	Plazas	Cantidad	Plazas	Cantidad	Plazas
238	33695	12	11199	173	180

Los números que hay en la última fila de la tabla representan los valores máximos encontrados para cada tipo de establecimiento.

A su vez, cada tipo de establecimiento está subdividido en subtipos de establecimiento. La lista de subtipos es la siguiente:

Hoteles	Campings	Casas rurales
d'1 estrella	Luxe	Casa de poble compartida
de 2 estrelles	Primera	Casa de poble independent
de 3 estrelles	Segona	Masia

Hoteles	Campings	Casas rurales
de 4 estrelles	Tercera	Masoveria
de 5 estrelles		
Hostals i pensions		

Existen atributos de totales en cada tabla, pero estos se omitirán ya que no son relevantes para el almacén.

Un aspecto que conviene destacar es que no hay anomalías en las tablas: no hay valores nulos o errores tipográficos. Este hecho tiene relevancia puesto que, a diferencia de los equipamientos, los datos pueden importarse de forma directa en los procesos ETL.

3.2.3 Policies locals.xls

Otro libro con tres hojas de cálculo, una para cada año (2011-2013). Al igual que las hojas de cálculo de establecimientos, cada tabla tiene 947 filas que corresponden a los municipios de Cataluña. En este caso los atributos de las tablas corresponden a los posibles rangos de policías y los valores son la cantidad de estos por municipio.

Se podría presentar ahora una tabla con los valores máximos detectados para cada atributo pero lo cierto es que no es necesario puesto que sólo se busca el total de efectivos policiales por municipio. Esto es, de la lista de atributos, sólo interesa el "Total" tal como muestra la figura 3:

	A	B	C	D	E	F	G	H	I	J	K
1	Efectius d	les policies locals. Per graduació									
2	Catalunya. Distribució per municipis. Any 2013.										
3											
4		Super-	Intendent			Sots-					Per
5		intendent	major	Intendent	Inspector	inspector	Sergent	Caporal	Agent	Total	mil
6											
7	Abella de	0	0	0	0	0	0	0	0	0	0
8	Abrera	0	0	0	0	0	1	3	17	21	1,74
9	Ager	0	0	0	0	0	0	0	0	0	0

Ilustración 3 - Hoja de cálculo Policies locals

El atributo "Per mil" puede dar respuesta a uno de los requerimientos de GLTF pero lo cierto es que es mejor obtener este dato por combinación de efectivos policiales y población, de modo que permita la navegación por jerarquías en el cubo OLAP. Por otro lado, de los totales de los tres años, el máximo valor encontrado es de 2857 efectivos policiales.

3.2.4 Pernoctaciones.xls

En un principio, la fuente proporcionada por GLTF estaba en formato CSV, pero tras un análisis de la estructura de la hipotética tabla que contenía, se detectó que el orden de aparición de las celdas era demasiado complejo para su tratamiento, de modo que se optó por bajar la misma fuente pero en formato

de hoja de cálculo. Si bien esto es así para la carga inicial, más adelante se verá un tratamiento ETL para los mismos datos pero en formato CSV para las actualizaciones; esto es debido a que en la tercera etapa del presente trabajo ya se habían adquirido suficientes conocimientos de Kettle como para afrontar el tratamiento del archivo CSV.

En cuanto al contenido y estructura de este archivo, tenemos que se trata de las pernoctaciones y viajeros, tanto para los residentes españoles como para los extranjeros por mes (columnas) y por zona turística (filas). Gráficamente sería:

	Viajeros españoles			Viajeros extranjeros			Pernoct. españoles			Pernoct. extranjeros		
	M1	M2	…	M1	M2	…	M1	M2	…	M1	M2	…
Zona turística	205782			602601			813722			1788755		

Los datos numéricos son los valores máximos por zona turística y mes que se han encontrado para cada una de las cuatro variables de estudio (las celdas fueron combinadas para mostrar el valor en la misma columna que la variable).

El hecho de que existan dobles encabezados de columna es lo que probablemente complica mucho la forma de representar los datos en formato CSV. No obstante, como ya se comentó, esto no supondrá problemas para los procesos ETL de las actualizaciones anuales del almacén de datos.

3.2.5 Poblacion.csv

Aunque en un principio estas fuentes no fueron provistas por GLTF, fue necesario incluirlas para mejorar la capacidad de las consultas de los efectivos policiales por unidad territorial (no sólo por municipio). También resultó crucial para mejorar los cálculos relacionados con los ranking de equipamientos y la categorización de municipios.

Estructuralmente es análogo a la fuente “Policies locals.xls”, es decir, es una tabla normal con varios atributos (superficie, altitud…) del que sólo interesa uno, el de población. El valor más alto corresponderá obviamente al municipio de Barcelona, que cuenta con un número de habitantes del orden de varios millones (todos estos valores serán importantes de cara al diseño de las tablas ya que se busca representar numéricamente los valores en el menor espacio posible).

3.3 Análisis conceptual

Ahora que tenemos una visión general del conjunto de datos y de los rangos de valores que tienen, el siguiente paso es buscar cómo almacenarlos atendiendo a los requerimientos de GLTF y a la forma en que conviene crear un almacén de datos.

Es por este motivo por el que se expondrá a continuación una tabla que explique qué requerimientos pide GLTF y de qué nivel de detalle se dispone en

las fuentes. Los requerimientos han sido agrupados convenientemente para detectar cuáles serán las principales tablas de hechos que se van a obtener.

Requerimiento	Año	Mes	Equip.	Estab.	Municipio
Total de pernoctaciones estimadas por municipio		X		X	X
% de ocupación por marca turística		X		X	X
"Top ten" de municipios por franja de pernoctaciones		X		X	X
% de pernoctaciones de un municipio sobre el total		X		X	X
Distribución mensual y estacionalidad de las pernoctaciones		X		X	X
Promedio de viajeros por tipo de establecimiento y comarca		X	X	X	X
Ranking de municipios por categoría de equipamientos			X		X
Categorización de municipios A / B / C			X		X
Máximo y mínimo de efectivos policiales por tipología de establecimiento y municipio	X			X	X
Ratio de policías locales por habitante	X				X

Ilustración 4 - Tabla de requerimientos

Al visionar los datos de esta manera, y sin entender previamente los conceptos de granularidad, ya se puede entrever que no todos los datos se podrán guardar en las mismas tablas. En cualquier caso, se procede con el método convencional de modelado de almacenes de datos teniendo en mente esta tabla.

Por último, antes de comenzar con el modelo conceptual, hay que tener en mente que lo que se busca es diseñar una estrella, que consiste en una tabla de hechos que se relaciona con varias tablas llamadas dimensiones. A diferencia del “copo de nieve”, las dimensiones no tienen a su vez relaciones con otras tablas de dimensiones, de modo que si existen datos relacionados jerárquicamente, los de mayor nivel estarán repetidos en cada tupla de la tabla dimensión (desnormalizados). Se elige este modelo por motivos de eficiencia y porque no se actualizan los datos en el almacén, esto es, sólo hay operaciones de lectura que se ven facilitadas con esta estructura.

3.3.1 Elección del hecho

Es la unidad fundamental del almacén de datos y conforma el conjunto de acontecimientos con los datos numéricos de interés para GLTF (los que están relacionados intrínsecamente con los procesos de negocio).

Los acontecimientos que desea estudiar GLTF han sido convenientemente separados por colores en la ilustración 4, estos son:

- Las pernoctaciones (dará lugar a la tabla de hechos “h_pernoctaciones”).
- Los viajeros (“h_viajeros”).
- Los equipamientos (“h_equipamientos”).

- Los efectivos policiales (“h_policias”).
- La relación policías por habitante (“h_policias_ratio”).

3.3.2 Granularidad

La granularidad tiene que ver principalmente con el nivel de detalle máximo alcanzable para un tipo de dato y está intrínsecamente relacionado con el concepto dimensión.

Este parámetro se tiene que tratar desde dos puntos de vista que tienen que ver con la disponibilidad de las fuentes y con lo que pretende obtener GLTF. En pocas palabras, si GLTF busca datos cuyo nivel de detalle existe en las fuentes, habrá suerte; si no fuera así, estamos ante un problema que requiere la desagregación de los datos, lo cual implica distribuirlos con mejor o peor acierto a lo largo de las variables de interés. Se puede ver de la siguiente forma y usando la tabla de la ilustración 4:

- Granularidad óptima
    - “Ranking de municipios por categoría de equipamientos” y “Categorización de municipios A/B/C”: ambos necesitan la granularidad “equipamientos” y “municipios” (directamente obtenible desde “equipamientos.csv”).
    - “Ratio de policías locales por habitante”: requiere la granularidad “año” y “municipios” (directamene obtenibles desde “Policies locals.xls”).
- Necesita desagregación de datos
    - Pernoctaciones: todos los requerimientos piden granularidad “establecimientos” y “municipio”. No se dispone de estos datos en las fuentes para los establecimientos y, territorialmente hablando, sólo hay disponibilidad a nivel “zona turística”.
    - Viajeros: ocurre lo mismo que con las pernoctaciones, pero además se pide la granularidad “equipamientos”, lo que implica otro nivel más de desagregación.
    - Máximo y mínimo de efectivos policiales: de los tres niveles de granularidad que exige, dos de ellos están cubiertos “año” y “municipio”, sin embargo, es necesario hacer una desagregación a nivel de “equipamiento”.

3.3.3 Dimensiones

Ya con la tabla de la ilustración 4 se pueden obtener las variables que necesita GLTF para consultar los datos, que a partir de ahora serán llamadas dimensiones. Éstas son:

- Tiempo (dará lugar a la dimensión “d_tiempo”).
- Lugar (dimensión “d_lugares”).
- Equipamiento (dimensión “d_equipamiento”).

- Establecimiento (dimensión “d_establecimiento”).

Además, ya se vio en el análisis detallado de las fuentes que para cada una de estas dimensiones tenemos elementos que están relacionados entre sí por una jerarquía. Así por ejemplo, para los establecimientos había tipo (hotel, camping, casa rural) y subtipos (de una estrella, de lujo, masía…). En la siguiente ilustración se muestra el conjunto total de dimensiones con sus jerarquías:

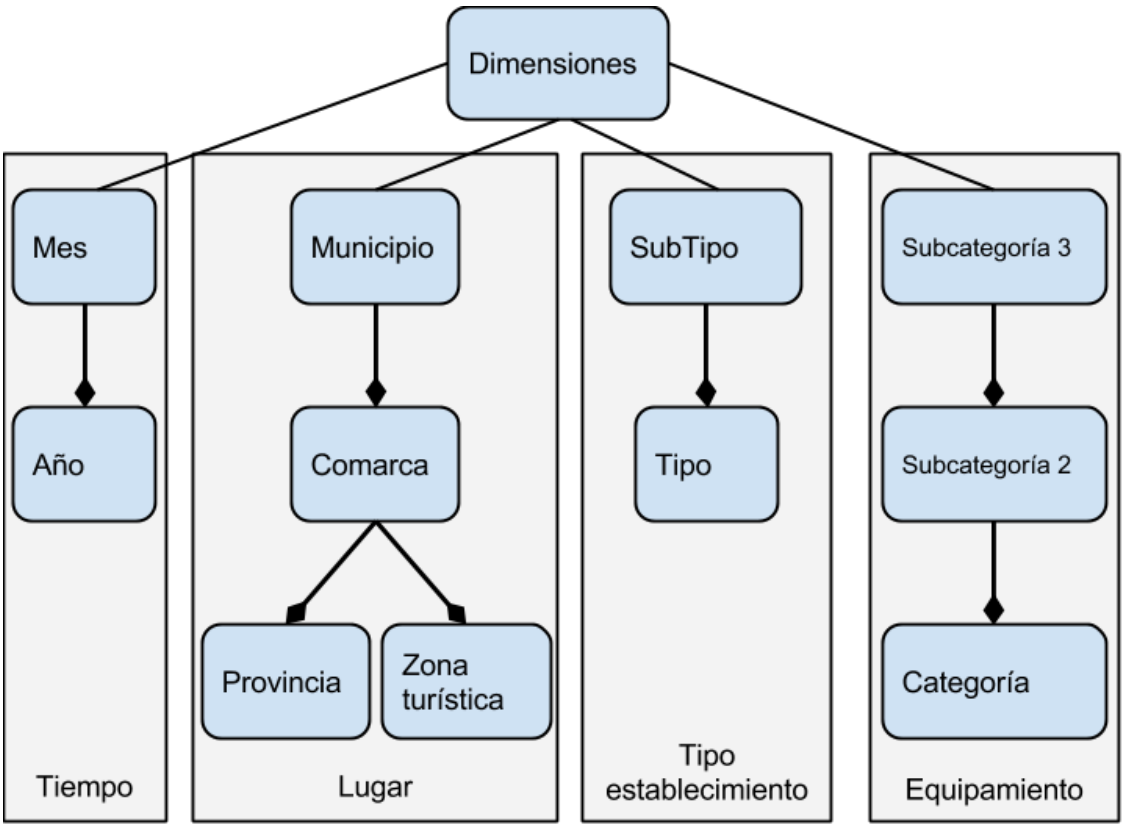


Ilustración 5 - Dimensiones y jeraquías

A partir de las fuentes podría haberse considerado una quinta jerarquía, la de los policías (con sus distintos rangos); sin embargo, GLTF no parece estar interesada al respecto, así que el dato “policía” sólo será tratado como una atributo más (o medida, si hablamos pensando ya en los cubos OLAP).

Otro detalle que se ha incluido en este diagrama es que, para la dimensión lugares, es posible añadir un elemento más a la jerarquía que “viene de regalo” al usar una fuente externa de datos de municipios, y es el nivel de agrupación “Provincias”. El único problema que tiene es que éste no puede ser compartido con las “zonas turísticas”, esto es, el nivel “comarca” puede tener dos formas de agrupación, por “zona turística” o por “provincia” pero no pueden compartirse ambas a la vez ya que origina conflictos de pertenencia (hay zonas turísticas que comparten dos provincias).

3.3.4 Atributos de las dimensiones

Existen varios atributos que se agregarán a las dimensiones y que cumplirán algún rol, ya sea como descriptor o como pieza clave en los procesos ETL para la realización de algunos cálculos. Sólo se indicarán los que no aparecen de forma implícita en la ilustración 5:

- Dimensión Tiempo:
  - lastDay: indica el último día de cada mes. Este dato es necesario para todo cálculo que implique pernoctaciones y plazas de establecimientos, ya que es importante cuantificar cuál es el máximo número de pernoctaciones posible que puede albergar la unidad territorial que se considere.
- Dimensión Equipamiento:
  - pesoLocal y pesoExtranjero: ambos atributos son **valores arbitrarios** que permiten ponderar el nivel de importancia de un equipamiento respecto de otros para un determinado colectivo, ya sea local (españoles) o extranjeros. Su inclusión en este trabajo está motivado en la necesidad de generar una distribución (desagregación) de los datos que sea diferente del reparto proporcional lineal.

3.3.5 Descriptores de las jerarquías

Para cada dimensión:

- Dimensión Tiempo:
  - Descriptor a nivel mes: “mes” cuyo formato es “aaaaMmm” tal como propone el INE.
  - Descriptor o agregador a nivel año: “anyo” en formato numérico “aaaa” (en algunas tablas de hechos será el descriptor porque no habrá granularidad fina a nivel mensual).
- Dimensión Lugar:
  - Descriptor: “municipio”.
  - Agregadores: comarca < marca (zona turística). También disponible: comarca < provincia.
- Dimensión Equipamientos:
  - Descriptor: “cat_nivel3”.
  - Agregadores: “cat_nivel2” < “cat_nivel1”.
- Dimensión Establecimientos:
  - Descriptor: “subTipo”.
  - Agregador: “tipo”.

3.3.6 Medidas

Dado que se dispondrá de 5 tablas de hechos, cada una tendrá las medidas o valores por los que está interesado GLTF. Hay que tener en cuenta que las consultas en los cubos OLAP siempre agrupan las medidas y aplican la función de agregación oportuna (generalmente la suma). Dicho en otras palabras, si los datos se consultan con la granularidad más fina (para todas las dimensiones) entonces los datos devueltos corresponden con los valores desagregados calculados en los procesos ETL. No obstante, si empezamos a subir en la jerarquía de las dimensiones, el motor Mondrian de los cubos OLAP empezará a agrupar los datos y sumar valores. Es importante tener en mente este hecho porque algunas medidas dejarán de tener sentido según el tipo de agregación que se aplique.

- h_pernoctaciones
  - plazasMesTotal: las plazas disponibles por mes, tipo de establecimiento y municipio.
  - plazasMesLibres: las plazas que no llegan a ocuparse por mes, tipo de establecimiento y municipio una vez han sido desagregadas las pernoctaciones. Esta medida y la anterior servirán para hallar los porcentajes de ocupación.
  - est_pern_local: es el valor estimado (desagregado) de pernoctaciones de españoles por mes, tipo de establecimiento y municipio.
  - est_pern_extr: es el valor estimado de pernoctaciones pero esta vez para extranjeros.
- h_viajeros
  - est_viaje_local: es el valor estimado (desagregado) de viajeros españoles por mes, tipo de establecimiento, categoría de equipamiento y municipio.
  - est_viaje_extr: ídem pero para extranjeros.
- h_equipamientos
  - rank: es un valor calculado que corresponde con la posición relativa en importancia de un municipio para una categoría de equipamiento determinada. Este cálculo sólo tiene sentido con los niveles de granularidad “municipio” y “cat_nivel3”. No es un campo calculado que permita navegación en el cubo OLAP con otros niveles de jerarquía.
  - classEqMun: es una lera (A/B/C) que indica a qué categoría pertenece un municipio en función del recuento de equipamientos total en el municipio.
  - classEqPob: es otra forma de categorizar el municipio, pero esta vez en función del número de equipamientos disponible por habitante. Estas dos últimas medidas sólo tienen sentido a nivel municipio.

- h_policias
    - minPolicias y maxPolicias: ambos atributos son, respectivamente, la cota inferior y superior de la estimación del atributo "policias" una vez ha sido desagregado por establecimientos. Realmente, el dato de interés es "policias" (o efectivos policiales por unidad territorial). La cuestión es que **los cálculos de desagregación producen mucho error por sucesivos redondeos** (no tiene sentido tener decimales en estas medidas).

      Más adelante se podrá ver que, en los cubos OLAP, estas dos medidas sirven de apoyo a un campo calculado que resulta ser la media de ambas medidas, dando lugar a una estimación que se acercará mucho más al valor real de la medida "policias".

      Lo ideal, es que todas las medidas estimadas usen esta técnica (viajeros, pernoctaciones…) que paliarán en gran medida los errores acumulados en el proceso de desagregación.
- h_policias_ratio
    - policias: es el valor que se obtiene directamente de "Policies locals.xls" del atributo "Total". Representa el total de policías que hay en el grano más fino (municipio).
    - poblacion: es el valor que se obtiene directamente de "Poblacion.csv" y representa el total de habitantes de un municipio.

      Ambas medidas en este caso sirven de apoyo para un campo calculado que se encargará de hallar el ratio (policías/población) para cualquier unidad territorial que se consulte.

Por último, hay una serie de aspectos que conviene tener en cuenta respecto a todas las estimaciones que se obtienen por desagregación:

- En las pernoctaciones, se le ha dado la misma importancia a todos los tipos de establecimiento, lo que implica que éstas se distribuirán por igual entre hoteles de 5 estrellas que en pensiones (siempre que el municipio disponga de dichos establecimientos).
- No se tiene en cuenta en ningún momento la estacionalidad de las pernoctaciones. Se entiende que debe haber algún tipo de preferencia por los tipos de establecimiento según la época del año (campings en verano…). Este hecho requeriría un sistema de pesos en la dimensión establecimientos que tenga en cuenta además la época del año.
- Para los viajeros sí se ha implementado un sistema de pesos que permitirá realizar una distribución no lineal que, además, afectará de forma distinta a residentes españoles y extranjeros. De este modo es posible observar los dos tipos de distribución (pernoctaciones y viajeros) y cómo afecta en las medidas calculadas (en el caso del tanto por ciento de ocupación en las pernoctaciones se repiten muchos valores).

3.3.7 Estudio de viabilidad

Para cada una de las cinco tablas de hechos obtenidas en el diseño conceptual y usando los tipos de datos que ofrece MySQL (tamaño en bytes):

h_pernoctaciones

Atributo	Valor máximo	Tipo	Tamaño	Comentario
tiempo_id	65535	SMALLINT	2	1999M01 = 0, 1999M02 = 1…
municipio_id	947	SMALLINT	2	Sólo Cataluña
establecimiento_id	15	TINYINT	1	
plazasMesTotal	1044545	MEDIUMINT	3	(33695 plazas * 31 días máx)
plazasMesLibres	1044545	MEDIUMINT	3	Nadie ocupa las plazas máximas
est_pern_local	813722	MEDIUMINT	3	
est_pern_extr	1788755	MEDIUMINT	3	

h_viajeros

Atributo	Valor máximo	Tipo	Tamaño
tiempo_id	65535	SMALLINT	2
municipio_id	947	SMALLINT	2
categoriaEquip_id	249	TINYINT	1
establecimiento_id	15	TINYINT	1
est_viaje_local	205782	MEDIUMINT	3
est_viaje_extr	602601	MEDIUMINT	3

h_equipamientos

Atributo	Valor máximo	Tipo	Tamaño
municipio_id	947	SMALLINT	2
categoriaEquip_id	249	TINYINT	1
rank	947	SMALLINT	2
classEqMun	1	CHAR	1
classEqPob	1	CHAR	1

h_policias

Atributo	Valor máximo	Tipo	Tamaño	Comentarios
anyo	66535	SMALLINT	2	Rango [1999, 65535]
municipio_id	947	SMALLINT	2	
establecimiento_id	15	TINYINT	1	
minPolicias	2857	SMALLINT	2	

h_policias

Atributo	Valor máximo	Tipo	Tamaño	Comentarios
maxPolicias	2857	SMALLINT	2	
plazas	33695	SMALLINT	2	Necesario para otras tablas de hechos

h_policias_ratio

Atributo	Valor máximo	Tipo	Tamaño	
municipio_id	947	SMALLINT	2	
anyo	65535	SMALLINT	1	
policias	2857	SMALLINT	2	
poblacion	16777215	MEDIUMINT	3	Es inferior a la cota máxima

Ni que decir tiene que los valores máximos aquí expuestos corresponden a los encontrados en el período que abarca desde el 2011 hasta el 2013. Pudiera suceder que en alguna actualización haya algún atributo que supere estas cotas máximas, sin embargo el margen se considera más que suficiente.

En cuanto a los cálculos, tenemos:

Propiedad	h_pernoctaciones	h_viajeros	h_equipamientos	h_policias	h_policias_ratio
Atributos clave prim.	5 bytes	6 bytes	3 bytes	5 bytes	3 bytes
Atributos restantes	12 bytes	6 bytes	4 bytes	6 bytes	5 bytes
1 tupla tabla hechos	17 bytes	12 bytes	7 bytes	11 bytes	8 bytes
Registros máx 1 año	170.460	42.444.540	235.803	14.205	947
Volumen máx 1 año	2,76 Mb	485,74 Mb	1,57 Mb	152,59 Kb	7,40 Kb

Ilustración 6 - Tabla espacio teórico ocupado por tablas de hechos

Como se puede comprobar, la única tabla que tiene peligro debido a la "explosividad" de las combinaciones de dimensiones es "h_viajeros". Sin embargo, hay que tener en cuenta que existe una gran cantidad de municipios que tienen muy pocos equipamientos/establecimientos, de modo que la mayor parte de estas combinaciones no se va a dar nunca.

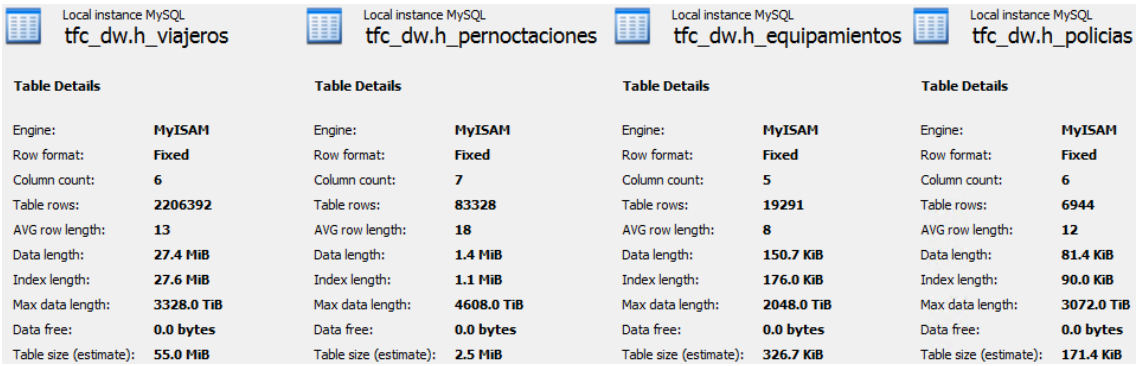


Ilustración 7 - Volumen datos real tablas de hechos

En la ilustración 7 se puede comprobar cuáles son los datos de espacio ocupado por cuatro de las cinco tablas de hechos. Téngase en cuenta que el tamaño estimado de las tablas que muestra es el resultado de sumar el tamaño de los datos y el del índice (el cual ocupa prácticamente el mismo volumen que los datos). Además, los datos de tamaño ocupado corresponden a la carga inicial, la cual contiene los datos del periodo del 2011 al 2013. Esto es:

Propiedad	h_pernoctaciones	h_viajeros	h_equipamientos	h_policias
Espacio teórico máx (1 año)	2,76 Mb	485,74 Mb	1,57 Mb	152,59 Kb
Tamaño tabla real (3 años)	1,4 Mb	27,4 Mb	150,7 Kb	81,4 Kb
Tamaño índice real (3 años)	1,1 Mb	27,6 Mb	176,0 Kb	90,0 Kb
Tamaño aprox. real (1 año)	477,9 Kb	9,1 Mb	50,2 Kb	27,1 Kb

Como se puede comprobar, tras la carga inicial, el volumen total de los datos (incluyendo índices) no supera los 60 Mb, en comparación con los casi 490 Mb teóricos (el doble con índices) si se dieran todas las combinaciones posibles de equipamientos y establecimientos en todos los municipios.

3.4 Diseño Lógico

Del diseño conceptual tenemos:

- Dimensiones
  - d_tiempo (**id**, mes, anyo, lastDay)
  - d_lugares (**id**, municipio, comarca, provincia, marca)
  - d_establecimientos (**id**, tipo, subTipo)
  - d_equipamientos (**id**, cat_nivel1, cat_nivel2, cat_nivel3, pesoLocal, pesoExtra)
- Tablas de hechos
  - h_equipamientos (**municipio_id, categoriaEquip_id**, rank, classEqMun, classEqPob)
  - h_pernoctaciones (**tiempo_id, municipio_id, establecimiento_id,** plazasMesTotal, plazasMesLibres, est_pern_local, est_pern_extr)
  - h_policias (**anyo, municipio_id, establecimiento_id**, plazas, minPolicias, maxPolicias)
  - h_policias_ratio (**municipio_id, anyo**, policias, poblacion)
  - h_viajeros (**tiempo_id, municipio_id, categoriaEquip_id, establecimiento_id**, est_viaje_local, est_viaje_extr)

En cuanto a los diagramas del modelo conceptual, se ha optado por cambiarlos por diagramas EER, que son mucho más explicativos y ricos en información. A

continuación se muestra cada tabla de hechos con sus relaciones con las dimensiones (modelo estrella) en vez de integrarlo todo en un solo esquema.

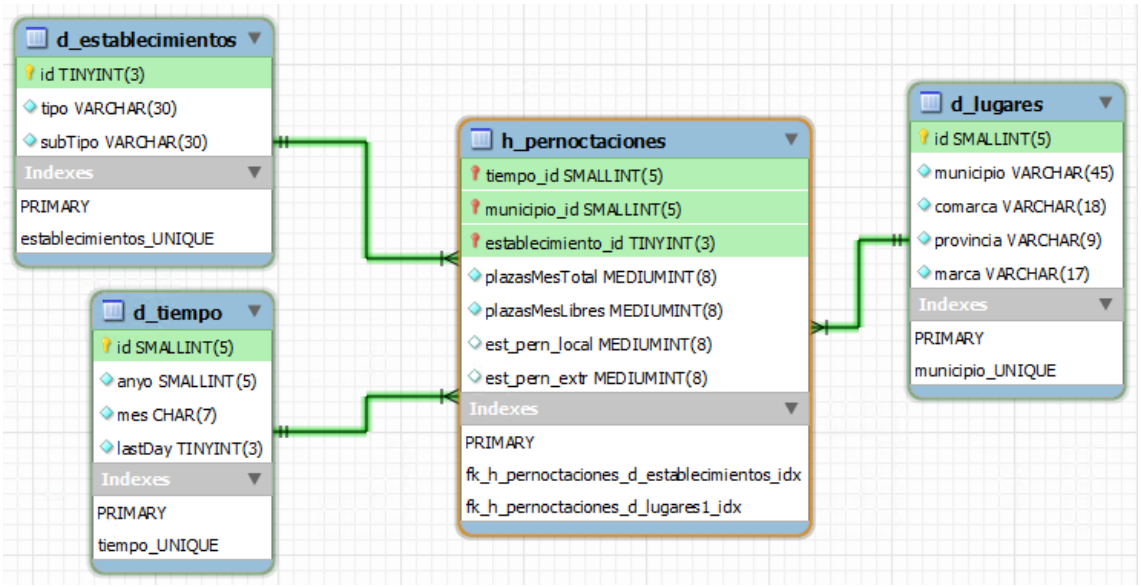


Ilustración 8 - Estrella tabla hechos pernoctaciones

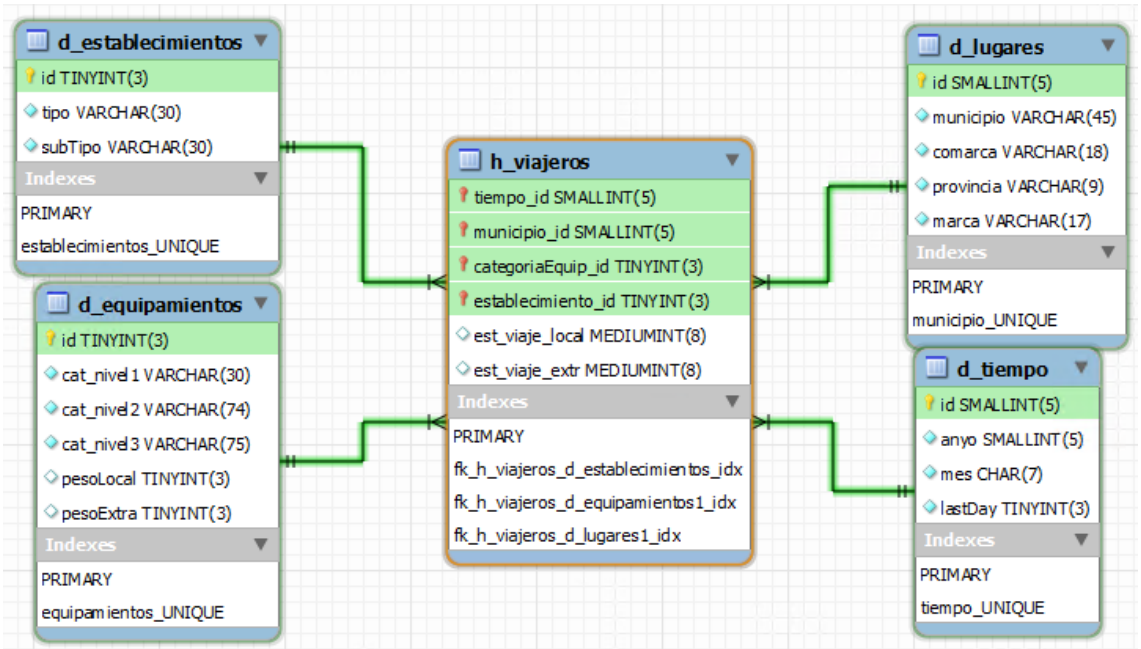


Ilustración 9 - Estrella tabla hechos viajeros

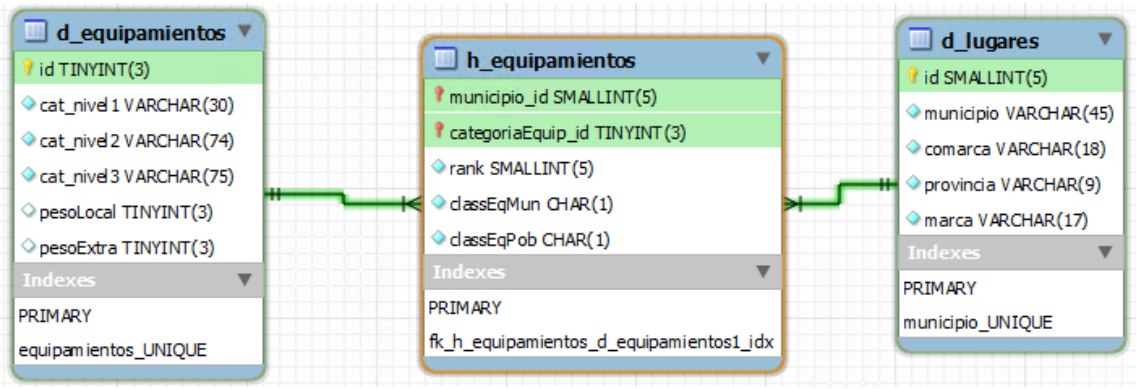


Ilustración 10 - Estrella tabla hechos equipamientos

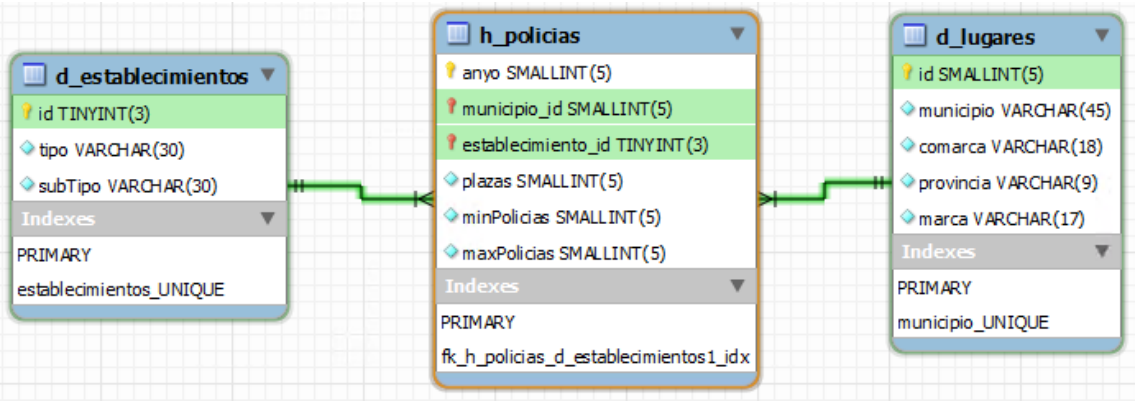


Ilustración 11 - Estrella tabla hechos equipamientos

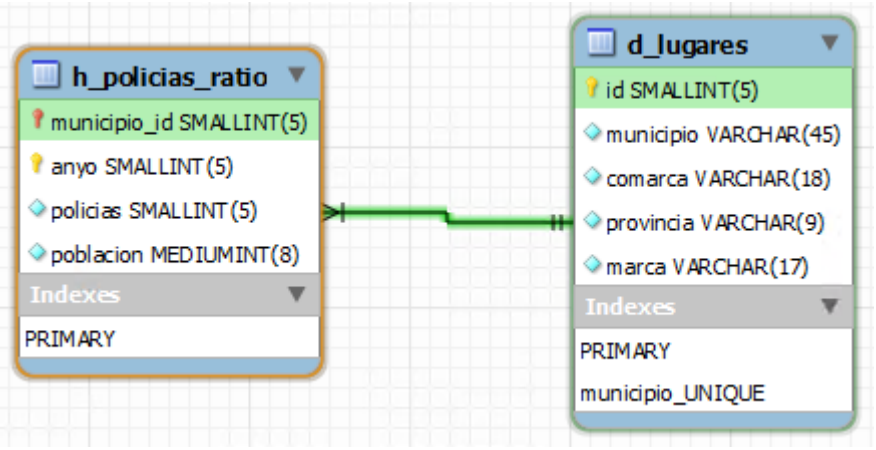


Ilustración 12 - Estrella tabla hechos ratio policías

Conviene tener en cuenta, para estos dos últimos diagramas, que en realidad el atributo “anyo” es de tipo degenerado y que, en los esquemas Mondrian para los cubos OLAP, la relación que tiene con la dimensión tiempo se establece a través de una consulta que obtiene los años filtrados (“SELECT DISTINCT…”).

3.5 Diseño físico

3.5.1 Dimensiones

d_lugares

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
id	SMALLINT	2	X			X	X
municipio	VARCHAR	45			X	X	
comarca	VARCHAR	18				X	
provincia	VARCHAR	9				X	
marca	VARCHAR	17				X	

d_equipamientos

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
id	TINYINT	1	X			X	
cat_nivel1	VARCHAR	30				X	
cat_nivel2	VARCHAR	74			X	X	
cat_nivel3	VARCHAR	75			X	X	
pesoLocal	TINYINT	1				X	
pesoExtra	TINYINT	1				X	

d_establecimientos

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
id	TINYINT	1	X			X	
tipo	VARCHAR	30				X	
subTipo	VARCHAR	30			X	X	

d_tiempo

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
id	SMALLINT	2	X			X	X
anyo	VARCHAR	4				X	
mes	VARCHAR	7			X	X	
lastDay	TINYINT	1				X	

Existe una serie de índices tipo UNIQUE en las dimensiones que corresponden con la granularidad más fina de cada dimensión. Estos son:

Índice dimensiones	Atributos	Descripción
municipio_UNIQUE	municipio	No puede haber 2 municipios con el mismo nombre.
equipamientos_UNIQUE	cat_nivel2, cat_nivel3	No puede existir una tupla con categorías de nivel 2 y 3 iguales (sí existen de categoría 3 con el mismo nombre).
tiempo_UNIQUE	mes	Con el formato aaaaMmm (como en el INE), no pueden existir 2 idénticos.
establecimientos_UNIQUE	subtipo	Evita la duplicidad de nombres de categoría de establecimientos a este nivel.

3.5.2 Tablas de hechos

h_pernoctaciones

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
tiempo_id	SMALLINT	2	X	X		X	
municipio_id	SMALLINT	2	X	X		X	
establecimiento_id	TINYINT	1	X	X		X	
plazasMesTotal	MEDIUMINT	3				X	
plazasMesLibres	MEDIUMINT	3				X	
est_pern_local	MEDIUMINT	3					
est_pern_extr	MEDIUMINT	3					

h_viajeros

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
tiempo_id	SMALLINT	2	X	X		X	
municipio_id	SMALLINT	2	X	X		X	
categoriaEquip_id	TINYINT	1	X	X		X	
establecimiento_id	TINYINT	1	X	X		X	
est_viaje_local	MEDIUMINT	3					
est_viaje_extr	MEDIUMINT	3					

h_equipamientos

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
municipio_id	SMALLINT	2	X	X		X	
categoriaEquip_id	TINYINT	1	X	X		X	

h_equipamientos

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
rank	SMALLINT	2				X	
classEqMun	CHAR	1				X	
classEqPob	CHAR	1				X	

h_policias

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
anyo	SMALLINT	2	X	X		X	
municipio_id	SMALLINT	2	X	X		X	
establecimiento_id	TINYINT	1	X	X		X	
plazas	SMALLINT	2				X	
minPolicias	SMALLINT	2				X	
maxPolicias	SMALLINT	2				X	

h_policias_ratio

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
municipio_id	SMALLINT	2	X	X		X	
anyo	SMALLINT	1	X	X		X	
policias	SMALLINT	2				X	
poblacion	MEDIUMINT	3				X	

Donde

Título campo	Significado
CP	Es Clave Primaria
CF	Es Clave Foránea
UQ	Índice único
NN	No Nulo
AI	Campo auto incrementado

Nota: todos los datos numéricos son de tipo UNSIGNED para duplicar su rango máximo. Además, el motor de almacenamiento escogido para todas las tablas es MyISAM (no hay actualizaciones de datos, sólo se prevé la lectura).

Aparte, hay algunas decisiones que quizás llamen la atención, por lo que conviene explicarlas:

- El tipo de datos del atributo “id” de “d_tiempo” es SMALLINT en vez de TINYINT (ya que con la carga inicial sólo se prevé que la dimensión contenga 36 tuplas). Esta decisión fue tomada a raíz de la inclusión de un sistema de actualización anual del almacén de datos y del hecho de que hay disponibles datos estadísticos de pernoctaciones en el INE a partir del año 1999. Para ofertar una escabilidad temporal razonable, es posible que el tipo TINYINT se quede corto, así que se amplía el rango del mismo por prevención.
- El atributo “plazas” en la tabla de hechos “h_policias”: en efecto, no es una medida de utilidad para las consultas en los cubos OLAP. La razón por la que se incluye es porque las tablas de hechos de pernoctaciones y de viajeros se construyen usando ésta como base (por comodidad). Se puede comprobar que esta tabla ocupa muy poco espacio en el almacén (ver tabla de la ilustración 6).

4 Procedimientos ETL

Son el núcleo del presente trabajo que decidirá el éxito o fracaso del almacén de datos y las expectativas depositadas en él. Esto es debido a la gran cantidad de tomas de decisión que implica su elaboración: detección de errores, tratamiento de resultados inesperados, ajustes de calidad, adecuación a los requerimientos…

4.1 Fuente de datos externa

Como se puede observar en la ilustración 2, se menciona la existencia de una fuente de datos externa. Esto es debido a que la lista de municipios encontrada en las distintas fuentes de datos difieren unas de otras, razón por la que se decidió buscar una fuente externa para contrastar y consensuar los nombres de municipio en todas las fuentes.

Como se puede observar en el esquema EER de la ilustración 13, la base de datos externa contiene además las tablas que definen la jerarquía territorial desde el municipio hasta las comunidades autónomas, pasando por las comarcas, marcas (zonas turísticas) y provincias.

En principio, esta fuente de datos no contenía toda esta información (la rama comarcas y marcas) y fue necesario introducir los datos manualmente tomando como referencia el IDESCAT, que fue finalmente la fuente de datos que se ha considerado como la más fiable de todas[1].

Del IDESCAT también fue necesario importar las tablas de población de los años 2011 al 2013 cuyos datos de poblacion eran necesarios para la realización de

algunos cálculos. Al ser la mayoría de las fuentes del IDESCAT, el problema principal de conflictos de integridad referencial respecto de los nombres de municipio quedará del lado de la fuente “equipamientos.csv”.

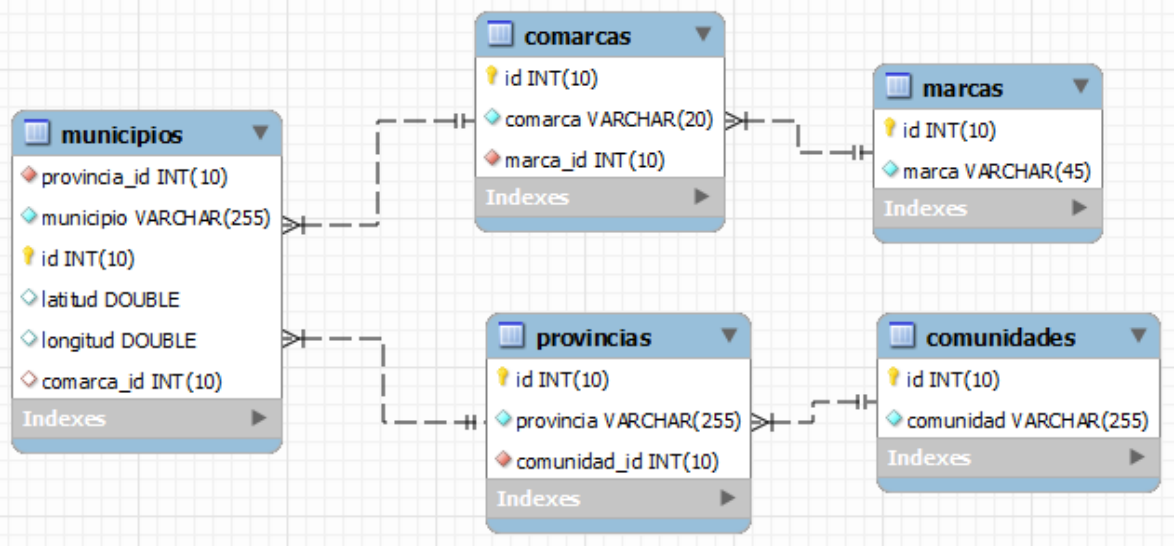


Ilustración 13 - Fuente de datos externa (municipios)

Para tener una idea de este problema, se presenta una tabla con los conflictos de nombres de municipios encontrados:

Municipios (fuente externa)	IDESCAT	Equipamientos.csv
Cabrera d’Anoia	Cabrera d’Anoia	Cabrera d’Igualada
Saus, Camallera i Llampies	Saus, Camallera i Llampies	Saus
Vimbodí i Poblet	Vimbodí i Poblet	Vimbodí
Roda de Barà	Roda de Berà	Roda de Barà
Santa Maria de Corcó	Esquirol, l’	Santa Maria de Corcó

Ilustración 14 - Tabla conflictos nombres de municipio

Como se podrá entender, los conflictos entre varias fuentes distintas deben ser resueltos en función de la fuente más fiable. Para el caso de la fuente externa, se optó por cambiar manualmente “Roda de Barà” y “Santa Maria de Corcó” para consensuar con IDESCAT; de este modo se pudo centrar la atención en resolver los conflictos en “Equipamientos.csv”.

4.2 Carga inicial

Para empezar el procedimiento ETL de la carga inicial, es necesario tener en cuenta que los archivos fuente se encuentran situados en la carpeta "TFG_final/CargaInicial" y los procesos ETL en "TFG_final/ETL_init". El archivo encargado de iniciar el proceso está directamente en la carpeta raíz "TFG_final" y se llama "Final_Job.kjb".

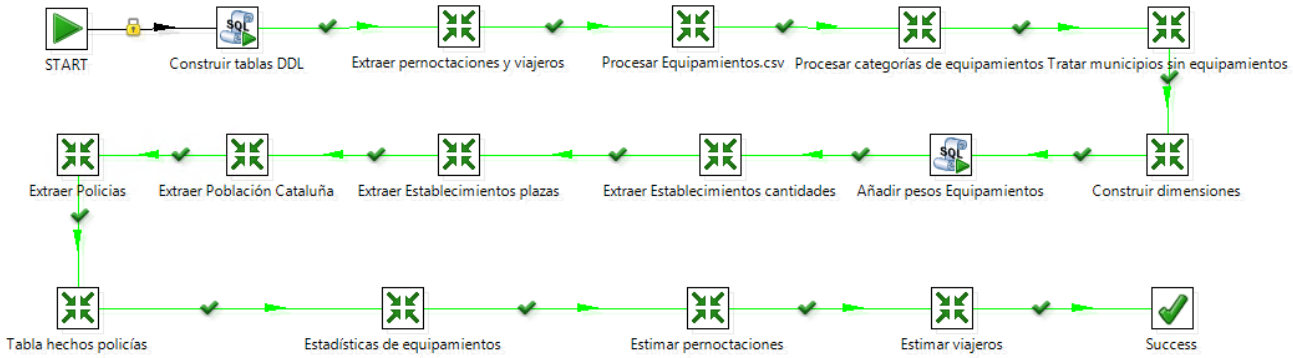


Ilustración 15 - Carga inicial (Final_Job.kjb)

Como se puede comprobar, el proceso global es una secuencia sencilla de transformaciones y dos scripts SQL sin ningún tipo de comprobaciones de error (como la entrada es siempre la misma y se ha probado bastantes veces no se ha invertido tiempo en vigilar este aspecto).

El proceso comienza con el script encargado de ejecutar el archivo DDL "Construir tablas DDL" para poder empezar el volcado de datos en las tablas de apoyo. Los siguientes procesos ETL son básicamente los que se encargan de extraer datos de las fuentes, realizando las transformaciones necesarias para adecuar los datos a las tablas de apoyo previamente creadas.

En un punto intermedio de las primeras extracciones (las de equipamientos), se procede a crear todas las dimensiones ("Construir dimensiones") para facilitar un poco el resto de extracciones de datos. Tras éste, se ejecuta el script "Añadir pesos Equipamientos", que básicamente lo que hace es completar la dimensión "d_equipamientos" asignando los atributos de pesos necesarios para las estimaciones de viajeros.

Seguidamente se realiza la segunda tanda de extracciones de datos del IDESCAT para, finalmente, empezar a construir las tablas de hechos con todos los cálculos pertinentes.

A continuación se explica con un poco más de detalle cada proceso.

4.2.1 Construir tablas DDL

En este proceso, además de las mencionadas tablas de dimensión y hechos que ya han sido explicadas en detalle en el diseño lógico y físico, existe otra serie de tablas de apoyo que se comentan a continuación:

tmp_equipamientos_full

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
Id	INT	4	X			X	X
nombreDireccion	VARCHAR	192				X	
municipio	VARCHAR	45				X	
cp	VARCHAR	5				X	
telefono	VARCHAR	12					
longitud	DECIMAL(10,8)	4				X	
latitud	DECIMAL(10,8)	4				X	
cat_nivel1	VARCHAR	30				X	
cat_nivel2	VARCHAR	74				X	
cat_nivel3	VARCHAR	75				X	

Esta tabla sirve para guardar los equipamientos que se obtienen en el proceso "Procesar categorías de equipamientos". Se usa como base para la construcción de aquellas tablas de hechos que tengan la dimensión equipamientos ("h_viajeros" y "h_equipamientos").

tmp_est_cantidad

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
municipio_id	SMALLINT	2	X			X	
Anyo	SMALLINT	2	X			X	
tipoEstablecimiento	VARCHAR	14	X			X	
Cantidad	SMALLINT	2				X	

Es la resultante del proceso "Extraer Establecimientos cantidades". Sirve de base para la creación de la primera tabla de hechos ("h_policias").

tmp_est_plazas

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
municipio_id	SMALLINT	2	X			X	
anyo	SMALLINT	2	X			X	
tipoEstablecimiento	VARCHAR	14	X			X	

tmp_est_plazas

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
plazas	SMALLINT	2				X	

Es la resultante del proceso “Extraer Establecimientos plazas”. Sirve de base para la creación de la primera tabla de hechos (h_policias). Se han creado dos tablas separadas en vez de una sola que contenga ambos atributos por “razones históricas”. Dado que importar hojas de cálculo resultó un proceso muy laborioso y el esquema resultante era demasiado grande, tuvo que crearse dos procesos ETL independientes ergo fue necesario la creación de dos tablas (no ocupan casi espacio y es fácil unirlas, así que no se volvió a tocar más esta parte).

tmp_poblacion

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
municipio_id	SMALLINT	2	X			X	
anyo	SMALLINT	2	X			X	
poblacion	MEDIUMINT	3				X	

Esta tabla es nueva y corresponde con la resultante del proceso “Extraer población Cataluña”. Sirve de base para la construcción de las tablas de hechos “h_equipamientos” y “h_policias_ratio”.

tmp_pernoctaciones

Atributo	Tipo	Tamaño	CP	CF	UQ	NN	AI
marca	VARCHAR	17	X			X	
mes	VARCHAR	7	X			X	
viajerosLocal	INT	4					
viajerosExtranjeros	INT	4					
pernoctacionesLocal	INT	4					
pernoctacionesExtranjeros	INT	4					

Del proceso “Extraer Pernoctaciones y Viajeros” (fuente de origen la web del INE). Como es de esperar, sirve de base para la construcción de las tablas de hechos “h_pernoctaciones” y “h_viajeros”.

4.2.2 Extraer pernoctaciones y viajeros

Es el proceso encargado de extraer los datos del INE, que son a su vez los datos principales del estudio que quiere llevar a cabo GLTF.

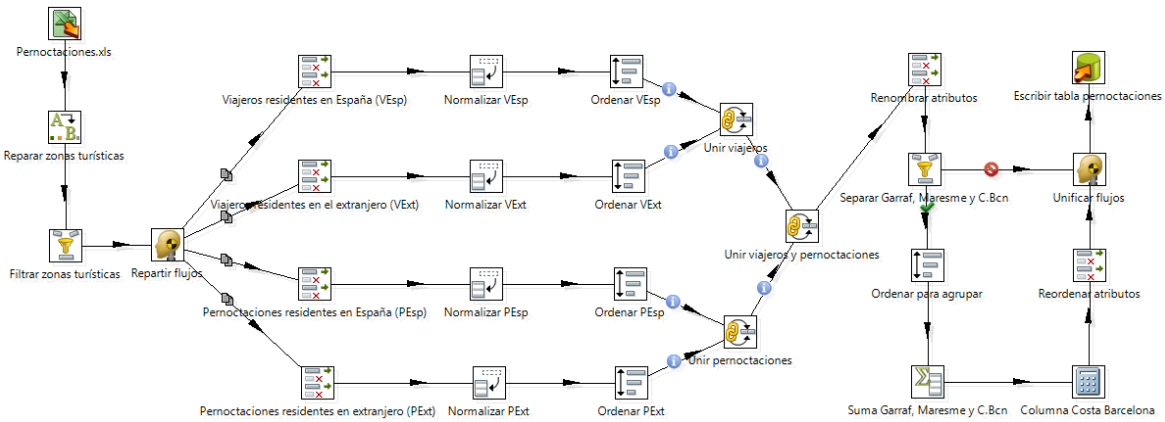


Ilustración 16 - ETL Extraer pernoctaciones y viajeros

Lo que hace el proceso es:

1. En la rama izquierda: leer la hoja de cálculo a partir del título de la columna "Zona turística". Seguidamente hace las correcciones necesarias para que las zonas turísticas correspondan con las que se guardarán en la dimensión "d_lugares". Finalmente realiza un filtro porque no todas las zonas turísticas se pueden estudiar debido al riesgo de solapamiento ("Costa Barcelona 2015" no tiene sentido y "Pirineo Catalán" debería quedar teóricamente cubierta por "Pirineus"). Las zonas turísticas válidas están definidas en un documento del IDESCAT[1].
2. La zona central se encarga de extraer los datos de cada una de las cuatro variables de estudio. Esto es:
    a. Viajeros residentes españoles.
    b. Viajeros residentes en el extranjero.
    c. Pernoctaciones residentes españoles.
    d. Pernoctaciones residentes en el extranjero.
3. La rama de la derecha resuelve un conflicto temporal de zonas turísticas: resulta que las zonas turísticas "Costa Del Garraf" y "Costa barcelona-Maresme" convergen en la zona turística "Costa Barcelona" a partir del año 2012 tal como se puede ver en la ilustración 17. Así que, para no perder los datos, se procede a unificar las tres zonas turísticas en una sola. Finalmente, los datos se vuelcan en la tabla "tmp_pernoctaciones".

Zona turística	Año 2011	Año 2012	Año 2013
Costa del Garraf	Disponible	No disponible	No disponible
Costa de Barcelona	No disponible	Disponible	Disponible
Costa Barcelona-Maresme	Disponible	No disponible	No disponible

Ilustración 17 - Tabla conflicto en zonas turísticas

4.2.3 Procesar Equipamientos.csv

Éste es el primero de una serie de tres procesos encargados de adecuar los datos de, sin duda, la fuente de datos más conflictiva de todas.

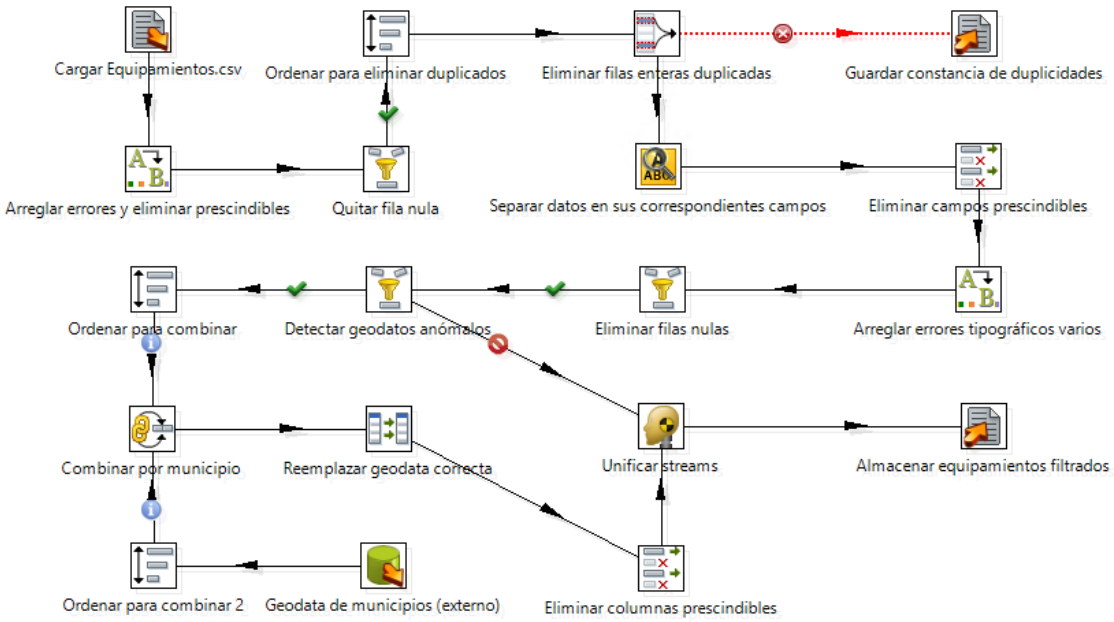


Ilustración 18 - ETL Reparar Equipamientos.csv

Lo que hace el proceso es:

1. Tener mucho cuidado con cómo lee los datos. Se ha usado como delimitador de campos el símbolo “;;;;;;”, de manera que obliga a Kettle a no separar ningún campo aunque encuentre un solo carácter “;” entre los caracteres de los supuestos atributos “nombre” y “direccion”. Gracias a esta medida se evitan algunos errores inesperados que dificultan la extracción de tuplas conflictivas (se da en 9 de los 31604 equipamientos válidos detectados en esta primera parte).
2. Eliminar tuplas enteras que se repiten (éstas se guardan en un archivo log con nombre “EquipamientosDup1”. En este punto se detectan 707 tuplas duplicadas.
3. Durante la primera etapa de las transformaciones únicamente hay un atributo “field1”. Como ya se explicó en el análisis detallado, este hecho se debe a la imposibilidad de importar los datos en sus correspondientes atributos debido a los conflictos de los atributos “nombre” y “direccion”. Este problema quedará resuelto a partir de la transformación “Separar datos en sus correspondientes campos” mediante el uso de expresiones regulares.
4. A partir de este punto y ya con el flujo de datos mínimamente estructurado, es posible realizar operaciones de limpieza básicas como la eliminación de atributos prescindibles o el tratamiento de tuplas que tienen datos de geolocalización anómalos, que son corregidos usando como apoyo la fuente de datos externa “municipios” comentada al inicio

de esta sección. El último paso consiste en volcar los datos en un fichero para ser importados en el siguiente proceso ETL.

4.2.4 Procesar categorías de equipamientos

Este proceso es el encargado de realizar el tratamiento del atributo “categorias” y, de paso, detectar más tuplas potencialmente duplicadas que no era posible detectar en la primera parte.

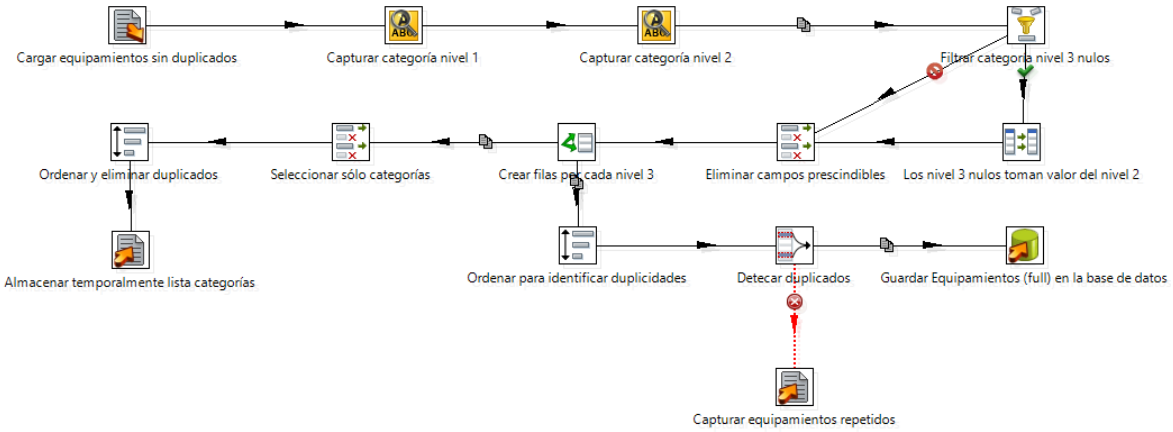


Ilustración 19 - ETL Procesar categorías de equipamientos

Lo que hace este proceso es:

1. Nada más importar los datos del archivo procesado en el anterior paso, aplica expresiones regulares en dos tandas: la primera es para obtener la “categoría 1” y la segunda la “categoría 2”. Como se explicó en el análisis detallado, el patrón de las categorías es:

   cat_nivel1 | cat_nivel2 (| cat_nivel3)*

   El hecho de tener el nivel 3 entre paréntesis y con asterisco es para dar a entender (al estilo de las expresiones regulares) que las categorías de nivel 3 pueden no aparecer, aparecer una vez o muchas más veces. Esto supone un problema, ya que las dimensiones no pueden contener descriptores nulos (mucho menos en el nivel de granularidad más fino).
2. Para resolver las categorías de nivel 3 nulas, el flujo se divide y se realiza la operación de copiar la categoría de nivel 2 a la 3. Finalmente se unifican los flujos de manera que se obtienen tuplas con los atributos “cat_nivel1”, “cat_nivel2” y “cat_nivel3”, donde este último contiene (al menos) una o muchas categorías de nivel 3.
3. En este punto ya es factible dividir cada tupla por tantas tuplas como equipamientos de categoría 3 existan, respetando los antecesores jerárquicos de categorías “cat_nivel1” y “cat_nivel2” de cada equipamiento.
4. Ya por último se crean dos copias del flujo: una servirá para crear un archivo externo con todos los equipamientos posibles etiquetados con sus 3 niveles de categorías; el otro flujo será almacenado en “tmp_equipamientos_full”, pero antes sufrirá un filtrado de tuplas

duplicadas. Dado que ahora tenemos tuplas con equipamientos específicos en el nivel de detalle máximo, es posible comparar unas tuplas con otras por “nombreDireccion”, “cp”, “latitud”,”longitud” y “cat3”, de modo que si todos estos atributos coinciden excepto (por ejemplo) el número de teléfono, es porque existen dos tuplas de exactamente el mismo equipamiento cuya única diferencia es que fueron registradas con 2 números de teléfono distintos.

Precisamente, este último caso es el se da con más frecuencia para los juzgados de 1ª instancia, cuyos duplicados quedan registrados en el archivo log “EquipamientosDefinitivoDup”.

4.2.5 Tratar municipios sin equipamientos

Uno de los detalles que no se comentó hasta ahora respecto del archivo Equipamientos.csv, es que no todos los municipios de Cataluña aparecen en éste. Existe una serie de municipios que carecen de equipamientos y que pueden suponer un problema a la hora de ser combinados con otras dimensiones. Para este tipo de casos es por lo que existe el “elemento nulo”, que vendría a ser como definir en la dimensión equipamientos una categoría especial de equipamiento denominada “Sin equipamiento”.

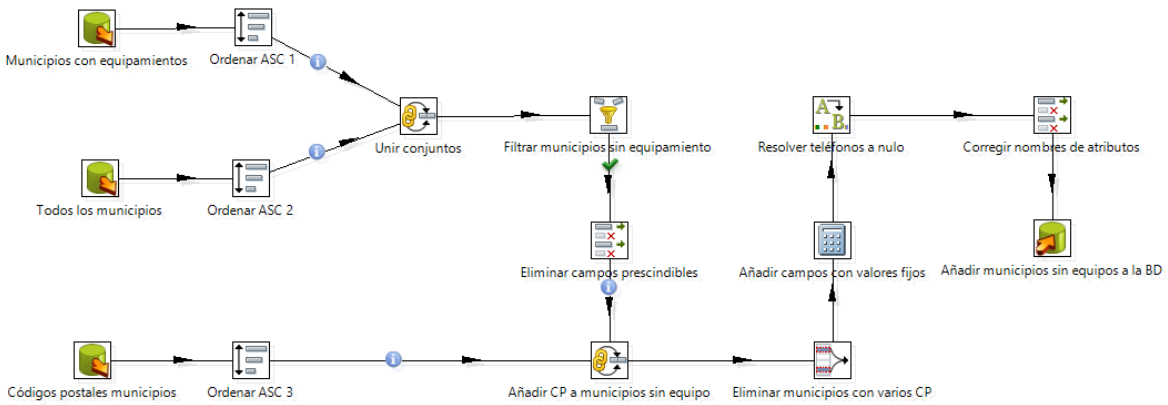


Ilustración 20 - ETL Tratar municipios sin equipamientos

El funcionamiento del proceso es el siguiente:

1. Combinar con JOIN OUTER la lista de todos los municipios encontrados en “tmp_equipamientos_full” con la lista de todos los municipios de la BBDD de apoyo “municipios”. De este modo se puede filtrar posteriormente aquellas tuplas que contengan determinados campos como null (que serán los municipios que no lograron combinarse).
2. Seguidamente se combina la lista de municipios que no están en “tmp_equipamientos_full” con los municipios de otra tabla de la BBDD de apoyo que contiene los códigos postales. No es una operación necesaria, puesto que el código postal es un campo prescindible. La intención de realizar todos estos arreglos fue para potenciar la búqueda de posibles equipamientos duplicados, pero lo cierto es que no tiene

mucho sentido esta operación, entre otras razones, porque hay municipios que pueden tener más de un código postal (razón por la que se usa una transformación que elimina municipios con varios CP).

3. El único paso de interés que queda es el que añade el atributo equipamiento al flujo de datos con el ítem especial “Sin equipamiento”. De este modo, cualquier consulta que se realice y que contenga la dimensión equipamientos siempre devolverá al menos 1 tupla representante de cada uno de los 947 municipios de Cataluña (a menos que se quiera filtrar específicamente por municipios “Sin equipamiento”).

   Hay un total de 10 municipios sin equipamientos.

4.2.6 Construir dimensiones

Tras el tratamiento de “Equipamientos.csv” es un buen momento para construir todas las tablas de dimensiones, ya que la de equipamientos era la única que presentaba incógnitas sobre su contenido (y ya ha sido extraído y debidamente estructurado en los anteriores procesos ETL).

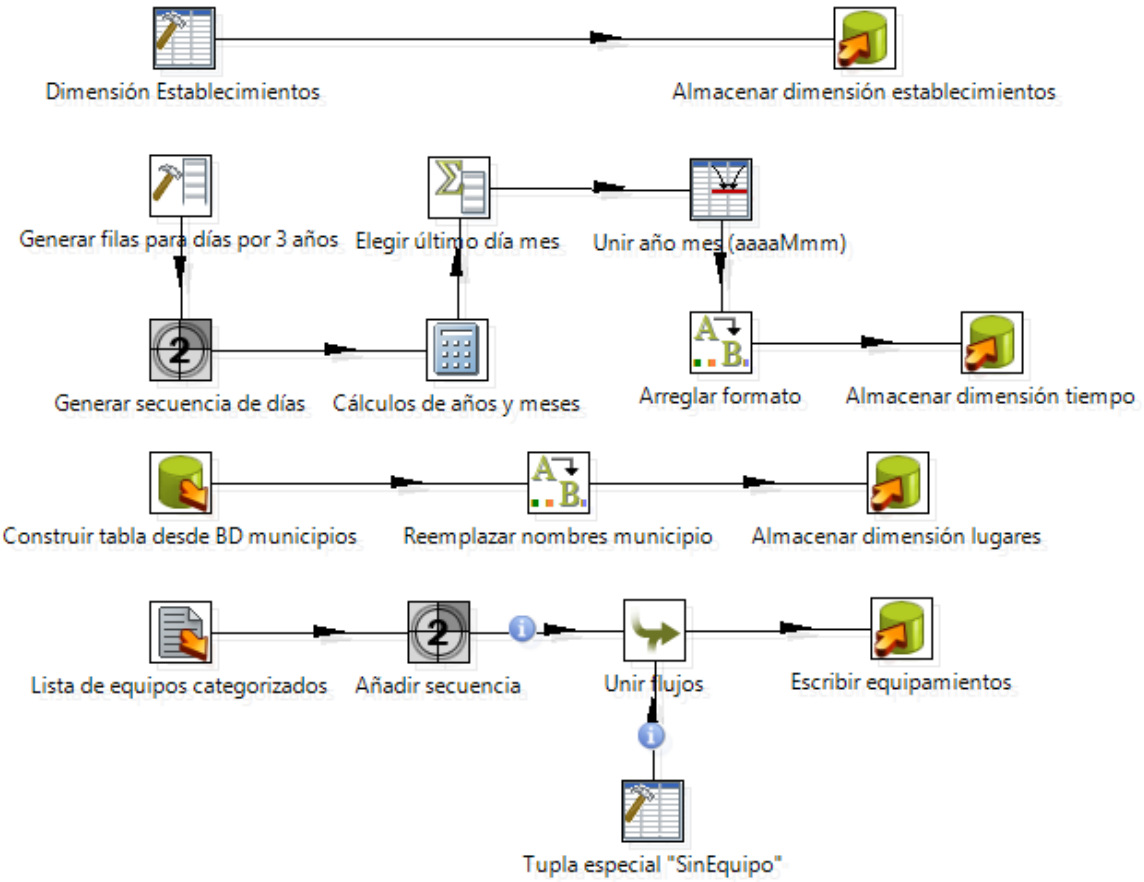


Ilustración 21 - ETL Construir dimensiones

Se trata de cuatro procesos independientes (uno por cada una de las cuatro dimensiones) que se han reunido en un único diagrama por comodidad.

1. Establecimientos: la entrada es manual y se incorpora de manera directa en la dimensión “d_establecimientos”.
2. Tiempo: ésta es la dimensión más compleja de tratar. Es una copia de uno de los procesos ETL que incorpora Pentaho como ejemplo de uso de Kettle pero transformado a conveniencia a los intereses de este trabajo.

   Se trata de una secuencia de días generada a partir de la fecha 1 de enero de 1999, momento en que sabemos que hay disponibles datos de pernoctaciones en el INE. Realmente sólo interesa el conjunto de meses y años, pero lo cierto es que esta secuencia ayuda a obtener los últimos días de todos los meses de todos los años (incluyendo bisiestos), que es un dato importante (lastDay) para el cálculo de pernoctaciones posibles según las plazas de establecimientos de un municipio.
3. Lugares: directamente, consultando de la BBDD de apoyo “municipios”, la cual contiene todos los niveles territoriales que se necesitan en la dimensión “d_lugares”. Aunque exista la transformación de reemplazar nombres de municipio, en realidad no es necesaria ya que en determinado punto del desarrollo de este trabajo se optó por cambiar los nombres de los municipios conflictivos en la fuente externa (la razón, porque esta BBDD se usa en varios pasos de los procesos ETL de los equipamientos).
4. Equipamientos: se obtiene de la lista de equipamientos que se almacena en un archivo externo (ver “Procesar categorías equipamientos”, ilustración 19).

   Tanto en equipamientos como en establecimientos, se añade el ítem nulo con id = 0 (“Sin equipamientos” y “Sin establecimientos”) necesario para que todas las combinaciones posibles de las cuatro dimensiones cubran todo el espacio de posibilidades en el dominio real de este trabajo.

4.2.7 Añadir pesos a equipamientos

Se trata de un script SQL que únicamente se limita a rellenar los atributos “pesoLocal” y “pesoExtra” de la tabla de la dimensión equipamientos. Conviene recordar, una vez más, la arbitrariedad con la que fueron asignados los valores de estos pesos (tomando como referencia el conocimiento adquirido en los trabajos estadísticos de FAMILITUR[2]).

```
UPDATE d_equipamientos SET pesoLocal=0, pesoExtra=0 WHERE id=0;
UPDATE d_equipamientos SET pesoLocal=1, pesoExtra=0 WHERE id=1;
UPDATE d_equipamientos SET pesoLocal=1, pesoExtra=0 WHERE id=2;
…
```

4.2.8 Extraer Establecimientos cantidades y plazas

Son dos procesos independientes en el diagrama global, pero realmente se podría haber tratado como un único proceso, ya que la cantidad de establecimientos y plazas son dos atributos que comparten el mismo municipio y año. La razón, como ya se ha explicado alguna vez, que la extracción es muy pesada en términos de elaboración en el entorno de Kettle, debido a la inflexibilidad de las hojas de cálculo.

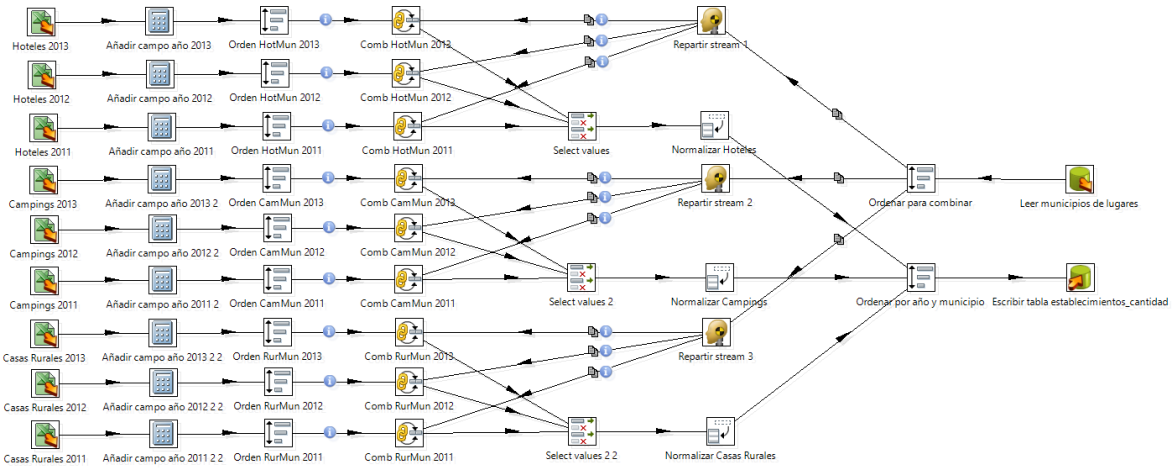


Ilustración 22 - ETL Extraer establecimientos cantidades y plazas

Las transformaciones que se usan son muy simples:

1. Leer la tabla en la hoja de cálculo correspondiente.
2. Añadir el año manualmente a todas las tuplas de cada tabla.
3. Ordenar y combinar con los municipios de la dimensión “d_lugares” para tener los identificadores numéricos de los municipios.
4. Normalizar filas: es decir, pasar todas las columnas de subtipos de establecimientos a filas, colocando el valor de cada una bajo el atributo “cantidad” o “plazas” dependiendo del proceso ETL que se esté viendo.
5. Escribir a la correspondiente tabla dependiendo del proceso ETL (esto es, a “tmp_est_cantidad” o “tmp_cantidad_plazas” según corresponda).

   Más adelante se podrá ver cómo es posible realizar este mismo tratamiento a partir de archivos CSV. El proceso es mucho más flexible, pero la contrapartida es la complejidad del tratamiento, ya que hay que tener mucho cuidado con la forma de extraer los metadatos.

4.2.9 Extraer población Cataluña

Éste es un proceso que debe tratar una fuente de datos en principio no prevista por GLTF. Se trata de extraer el número de habitantes de cada uno de los 947 municipios de Cataluña. Si bien es cierto que el ratio de policía por cada 1000 habitantes ya era un dato que existía de forma directa en las hojas de cálculo

"Policies locals.xls", lo cierto es que se trata de una solución muy poco flexible. Aparte, se ha encontrado que es necesario este dato para realizar cálculos más interesantes para la tabla de hechos "h_equipamientos".

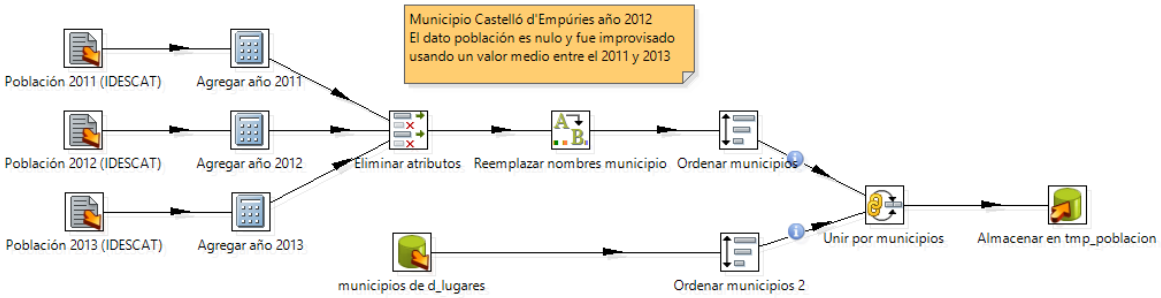


Ilustración 23 - ETL Extraer población Cataluña

El proceso es bastante sencillo, ya que los datos se importan de forma directa:

1. Se leen los archivos CSV y se importa la estructura de datos tal cual (no hay metadatos extras, ni adornos extraños en estas tablas).
2. Se agrega el año manualmente a todas las tuplas de cada tabla.
3. Se eliminan atributos prescindibles y se arreglan los municipios que tengan errores (en este caso, resulta que IDESCAT tiene en cuenta que "Santa Maria de Corcó" pasa a llamarse "Esquirol, L'" entre los años 2011 a 2013).
4. Un detalle importante y que da qué pensar para los procesos ETL de actualización, es que se ha encontrado un valor nulo de población para el municipio "Castelló d'Empuries" en el año 2012. En esta ocasión se ha podido solucionar de forma manual colocando el valor medio de los años 2011 y 2013 (cuando se haga actualizaciones y se detecten estos nulos, los valores medios se obtendrán a partir de los datos de la carga inicial, con lo que se podrá cubrir esta problemática).
5. Finalmente, se combina el flujo con la dimensión "d_lugares" para obtener los identificadores numéricos de los municipios y así almacenarlos en "tmp_poblacion".

4.2.10 Extraer policías

Éste es el último proceso de extracción propiamente dicho de una fuente de datos, en este caso de la hoja de cálculo "Policies locals.xls" y la metodología es totalmente análoga a la vista en "Extraer establecimientos cantidades y plazas".

Aparte, una característica importante en este proceso es que se almacenarán los datos en la primera tabla de hechos "h_policias_ratio", que es la tabla más simple de todas las tablas de hechos que requiere GLTF para sus análisis.

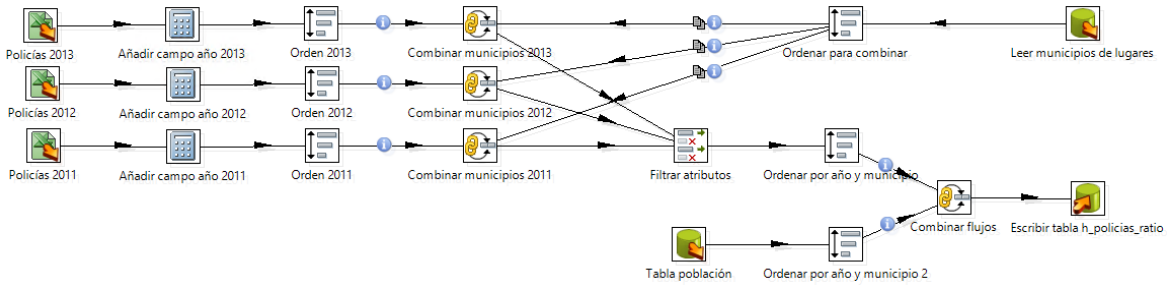


Ilustración 24 - ETL Extraer policías

Como ya se pudo ver en el análisis y diseño, lo que se busca en la tabla de hecho "h_policias_ratio" es tener los valores de número de policías y de habitantes para cada municipio en un año determinado.

Así, una vez se obtiene el número total de agentes de cada municipio de las hojas de cálculo, estos se combinan con la tabla de apoyo "tmp_poblacion" que se obtuvo en el proceso anterior (combinados por el id numérico de cada municipio).

La intención de tener los atributos "poblacion" y "policias" por el grano más fino "municipios" es que ofrece la posibilidad de generar un campo calculado en el cubo OLAP, el cual nos permitirá calcular el ratio de policías por habitante desde cualquier nivel de la jerarquía de la dimensión "d_lugares".

4.2.11 Tabla de hechos policías

Con este proceso ETL se da comienzo a los procesos encargados de generar atributos (medidas en los cubos OLAP) de valor para los análisis de GLTF.

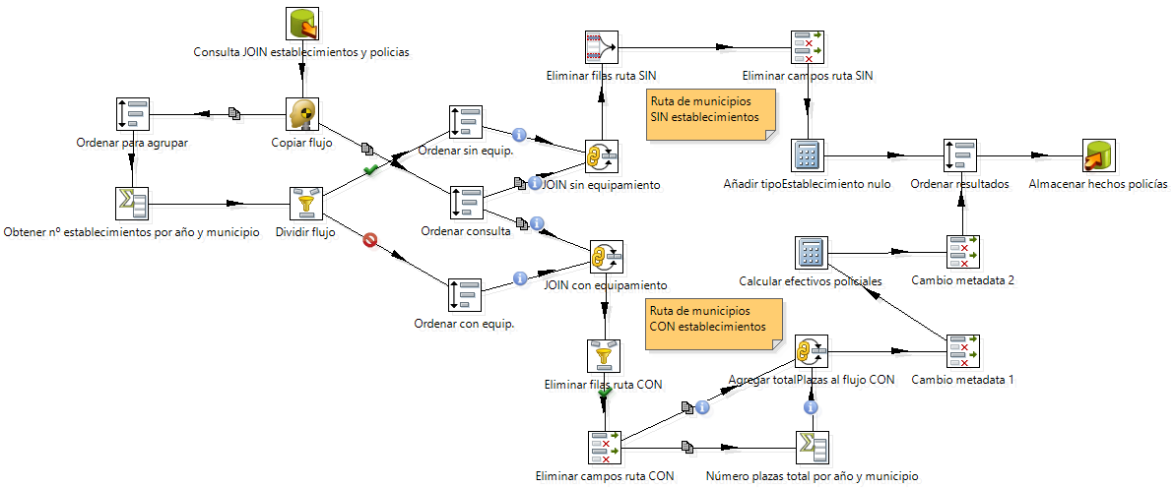


Ilustración 25 - ETL tabla hechos policías

El objetivo de este proceso ETL es generar los atributos "minPolicia" y "maxPolicia" que corresponden con el mínimo y máximo de efectivos policiales respectivamente para la tabla de hechos "h_policias". A su vez, debe ofrecer la desagregación de estos datos respecto de los establecimientos de cada

municipio para tener, finalmente, la posibilidad de poder trabajar con las dimensiones tiempo, lugares y establecimientos.

El proceso consiste en los siguientes pasos:

1. Se combinan los datos de la tabla de hechos “h_policias_ratio” (que contiene “policias” y “poblacion”, con los datos de las tablas de apoyo “tmp_est_cantidad” y “tmp_est_plazas”, que contienen respectivamente el número de establecimientos y plazas de cada subtipo de establecimiento en cada municipio.
2. El siguiente paso consiste en hallar el total de establecimientos por municipio. La intención de este cálculo en primera instancia es determinar si existe algún municipio que no tenga algún tipo de establecimiento (ya sea hotel, camping o casa rural).
3. Seguidamente, gracias a estos cálculos, se puede dividir el flujo en dos flujos que se denominarán CON y SIN (establecimientos). Estos se combinan con el flujo original de la consulta inicial para que ambos flujos contengan todos los atributos necesarios para la siguiente etapa.
4. Flujo CON
    a. Ahora que ya se sabe que en este flujo existe al menos un tipo de establecimiento para cada municipio, se puede hacer una limpieza de tuplas que tienen valor 0 para cada tipo de establecimiento. Dicho de otro modo, para el almacén de datos, que un municipio no tenga un determinado tipo de establecimiento quiere decir exactamente eso: no hay ese subtipo de establecimiento (no se indica que tiene 0).
    b. El siguiente paso que se realiza es hacer un cálculo de agregación del número de plazas en total de todos los establecimientos que hay en cada municipio y año. La intención de este cálculo es ofrecer la posibilidad de desagregar el número de polícias de cada municipio en función del número de plazas de cada establecimiento: habrá más policías donde haya más plazas que vigilar (reparto proporcional en función de “plazas”).
    c. La característica importante de este cálculo es que el redondeo final del mismo se hace a la baja y a la alta. Es decir, el concepto de mínimo y máximo de efectivos policiales se basa en dar la cota inferior y superior del resultado decimal, por lo que el valor de ambos atributos difiere como mucho en 1 unidad.

       Puede que parezca un cálculo trivial y sin sentido, pero este hecho cobra vital importancia a la hora de navegar entre los niveles de las dimensiones: a medida que subamos (comarcas, marcas, tipos de establecimientos…) y se vayan sumando (agregando) los valores de ambas medidas, mayor será la diferencia entre ambas medidas. Esto implica, por otro lado, que el valor medio entre ambas medidas será probablemente el

verdadero valor que tendría el atributo original “policias” si no se hubiera desagregado.

Éste sería el cálculo ideal para todas las estimaciones que se basan en la desagregación de datos.

5. Flujo SIN
    a. En cuanto al flujo de municipios sin establecimientos, lo que se hace en primer lugar es eliminar tuplas redundantes (sólo interesa saber qué municipios son). Posteriormente, se añaden los atributos “minPolicias” y “maxPolicias” con los valores del atributo “policias” (¡aunque no haya establecimientos, no hay que perder este dato!) y para el atributo “establecimiento_id” se coloca el valor del id del elemento nulo (0).
6. Finalmente, ya se pueden unificar ambos flujos CON y SIN y escribir los resultados en “h_policias”.

4.2.12 Estadísticas de equipamientos

Este proceso está pensado para generar los atributos “rank”, “classEqMun” y “classEqPob”, que son respectivamente el ranking de municipios (en función de la proporción de habitantes que pueden disfrutar de una determinada categoría de equipamiento), la categorización (A/B/C) por número de equipamientos en el municipio y la categorización (A/B/C) por proporción de habitantes que pueden disfrutar de determinada categoría de equipamiento.

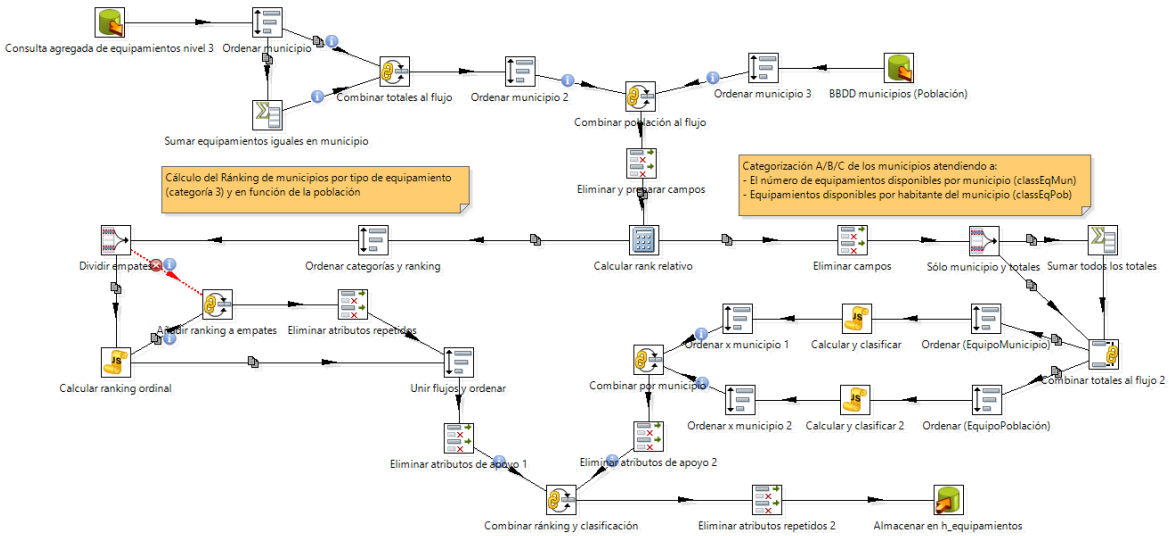
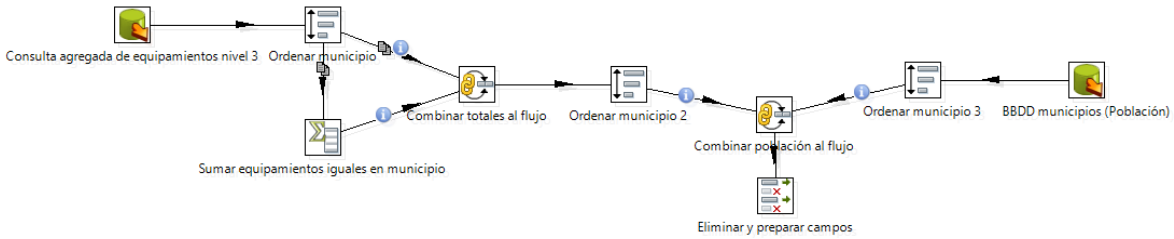


Ilustración 26 - ETL Estadísticas de equipamientos

El proceso aparenta ser complejo, así que será desgranado en cuatro partes con todo detalle para entender su funcionamiento.

1. En primer lugar, tenemos la entrada de datos:



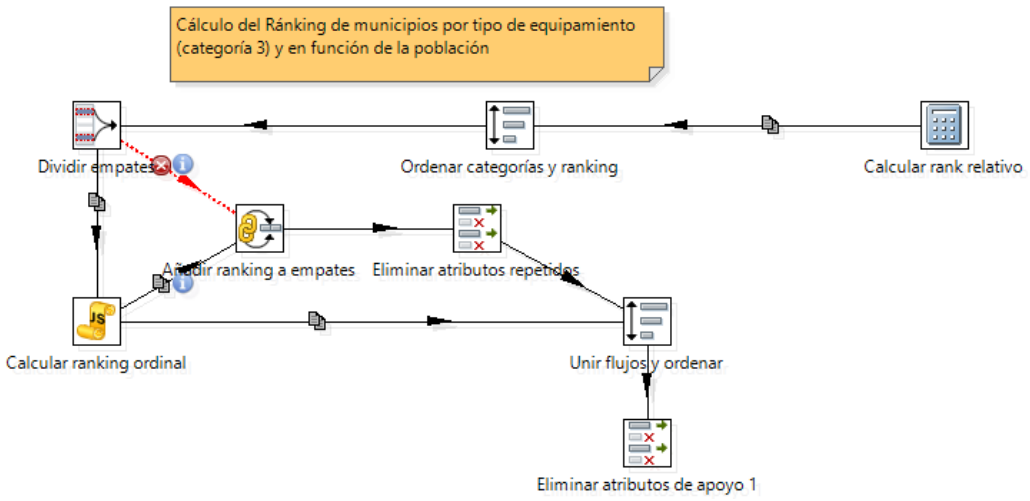
Por la rama izquierda se tiene como entrada de datos la tabla "tmp_equipamientos_full" y la dimensión "d_equipamientos". Se trata de una consulta agregada que devuelve, para cada municipio, todos los tipos de equipamientos existentes así como la cantidad de estos.

Seguidamente, se realiza una segunda consulta agregada que suma la cantidad de equipamientos de cada tipo ("eqMunCount") para crear el atributo "totalEqMun" (cantidad total de equipamientos existentes en el municipio).

En cuanto a la rama derecha, se trata de una consulta a la base de datos externa "municipios", que contiene datos de población de Cataluña del 2014 (IDESCAT). No hay ninguna razón que justifique por qué se usan estos datos, simplemente hay que tener en cuenta que los datos de equipamientos son atemporales y que, de hecho, no hay ranking ni categorizaciones para cada año. Es más, los cálculos se realizarán con la población simplemente por curiosidad y porque se considera que da una visión más objetiva para este tipo de cálculos.

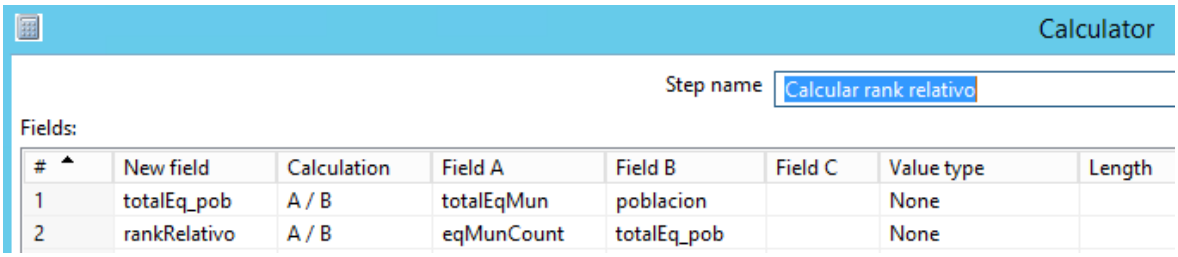
Por último, los dos flujos se combinan por el nombre del municipio, de modo que ahora tenemos un flujo de datos que contiene datos estadísticos de número de equipamientos del municipio y su población.

2. El cálculo del ranking de municipios por tipo de equipamiento:



a. El primer paso es calcular el ranking relativo de cada equipamiento respecto de la población del propio municipio. Esto es, no se trata de comparar equipamientos entre municipios, sino de averiguar cuál es la proporción de habitantes que puede disfrutar de un determinado tipo de equipamiento:

$$rankRelativo = poblacion \cdot \frac{eqMunCount}{totalEqMun}$$

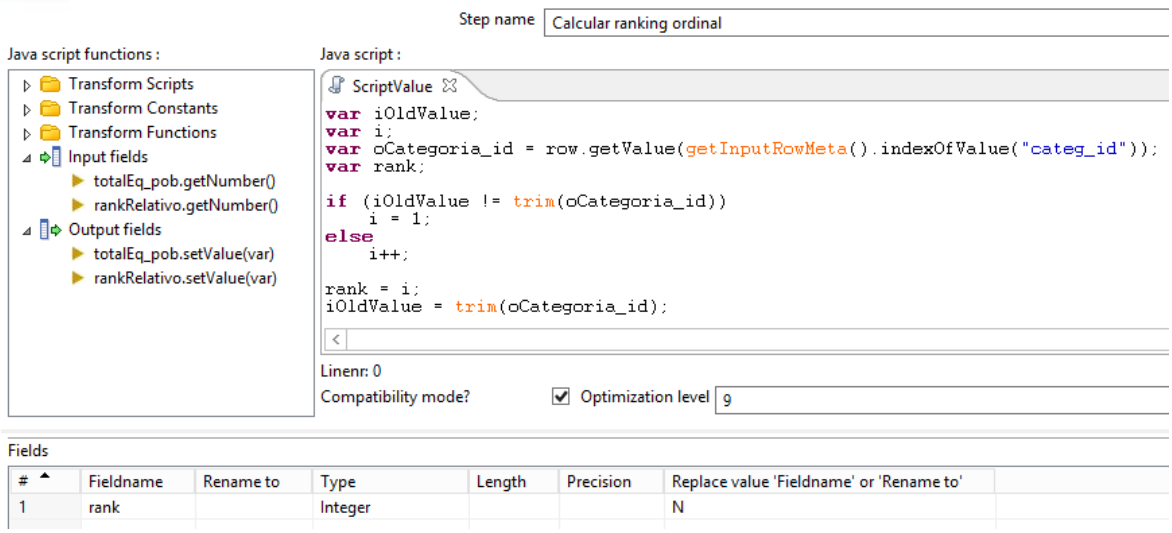


b. El siguiente paso es ordenar primero por categorías de equipamientos y luego por los valores de ranking relativo, de este modo se consigue que los municipios queden ordenados por categoría y por proporción de habitantes que disfrutan de un equipamiento. En este punto conviene aclarar que el atributo rankRelativo es una variable muy pobre en términos comparativos, ya que no es lo mismo un hospital de 100.000 camas en Barcelona que un hospital de 1000 camas en un municipio con menor número de habitantes: claramente vemos que se requiere de un factor adicional que permita darle un peso relativo a los equipamientos (ya sea por calidad, o cantidad de habitantes que puede soportar…).

c. Lo que viene a continuación es un truco que nos permite asignar números correlativos a todas las tuplas simulando el ranking final. El primer paso consiste en dividir el flujo de datos, apartando, para cada categoría, aquellos municipios que estén empatados en el ranking relativo:

d. Luego se creará el ranking, pero primero hay que hacer un paréntesis. En circunstancias “normales”, lo que se hace en casos de rankings es ordenar la lista por el atributo de interés y, seguidamente, emplear una transformación de secuencia que agrega el atributo Rank con la secuencia ordenada de números. Sin embargo, en este caso tenemos una lista jerarquizada (por categorías), con lo cual hay que idear otro método que permita reinicializar el ranking cada vez que se encuentre una nueva categoría.

   Lamentablemente, la única forma que se ha encontrado para realizar dicha operación pasa por el uso de un script que se ha denominado “Calcular ranking ordinal”. Gracias al uso de variables globales, es posible recordar la categoría de la anterior tupla, de este modo es posible saber cuándo debe reinicializarse el contador de ranking:

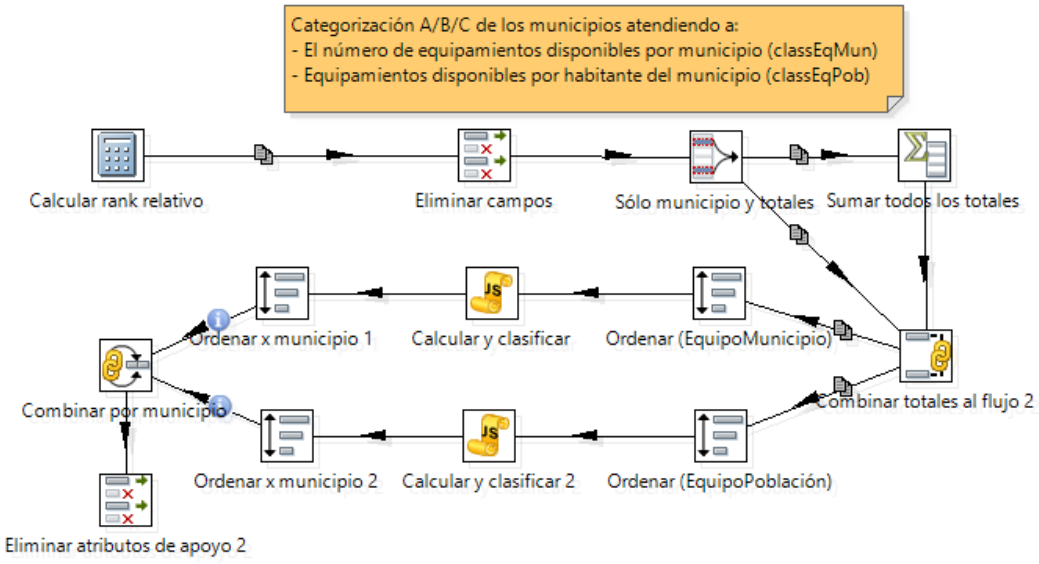


e. Ahora hay que tener en cuenta que hay dos flujos: uno que ya tiene la lista de municipios por categoría con el ranking definitivo, y otro que contiene una lista de categorías con municipios que tienen el mismo valor de ranking relativo que alguna de las tuplas del flujo principal. En otras palabras, el flujo de casos de empate.

   Con la transformación "Añadir ranking a empates" lo que se busca es combinar los empatados con las tuplas que ya tienen el ranking definitivo de modo que obtengan el atributo "rank".

f. Los últimos pasos que quedan consisten en eliminar atributos de cálculos y repetidos (resultantes de la combinación) para poder unir los flujos de datos.

3. Categorización A / B / C de municipios:



a. Se parte de la misma transformación que en el anterior apartado. Esto es, a partir del cálculo principal donde están los dos atributos:
    i. “totalEq_pob”: que resulta del total de equipamientos del municipio partido por la población.
    ii. “rankRelativo”: que indica la proporción relativa de una determinada categoría de equipamiento por habitante (o mejor dicho, indica cuántos habitantes pueden disfrutar de un determinado equipamiento).
b. Seguidamente se eliminan el “rankRelativo” y los demás atributos que no sean necesarios. Esto es, sólo debe quedar el municipio, el recuento total de equipamientos del municipio (“totalEqMun”) y la proporción “totalEq_pob”.

    La idea detrás de esta selección es que se va a crear dos tipos distintos de categorización de municipios:
    i. Por un lado, se va a categorizar en función del número de equipamientos total de un municipio (que será el parámetro a comparar con otros municipios).
    ii. Por el otro, se categoriza en función de la proporción de equipamientos por habitante, de modo si hubiera dos municipios con un solo equipamiento, uno será mejor que el otro en función de la cantidad de habitantes que puedan disfrutar de dicho equipamiento.

        Otra vez, conviene recordar que estas variables no son una solución idónea, pues no se tiene en cuenta la calidad de los equipamientos ni la proporción real de habitantes que puede disfrutar realmente del mismo.
c. Como el flujo contiene sólo municipios y atributos con totales, es lógico que la gran mayoría de las tuplas repitan valores (de hecho, sólo debe haber 947 tuplas, una por municipio). Luego han de filtrarse en “Sólo municipios y totales” (no se requiere ningún parámetro, se filtra por fila entera).
d. El flujo se divide en dos, ya que se necesitan los sumatorios totales de ambos atributos (“totalEqMun” y “totalEq_pob”) para poder hallar las proporciones relativas de cada municipio (no hay campo por el que agrupar, interesan todas las filas).

The fields that make up the group:

#	Group field
1	

Aggregates :

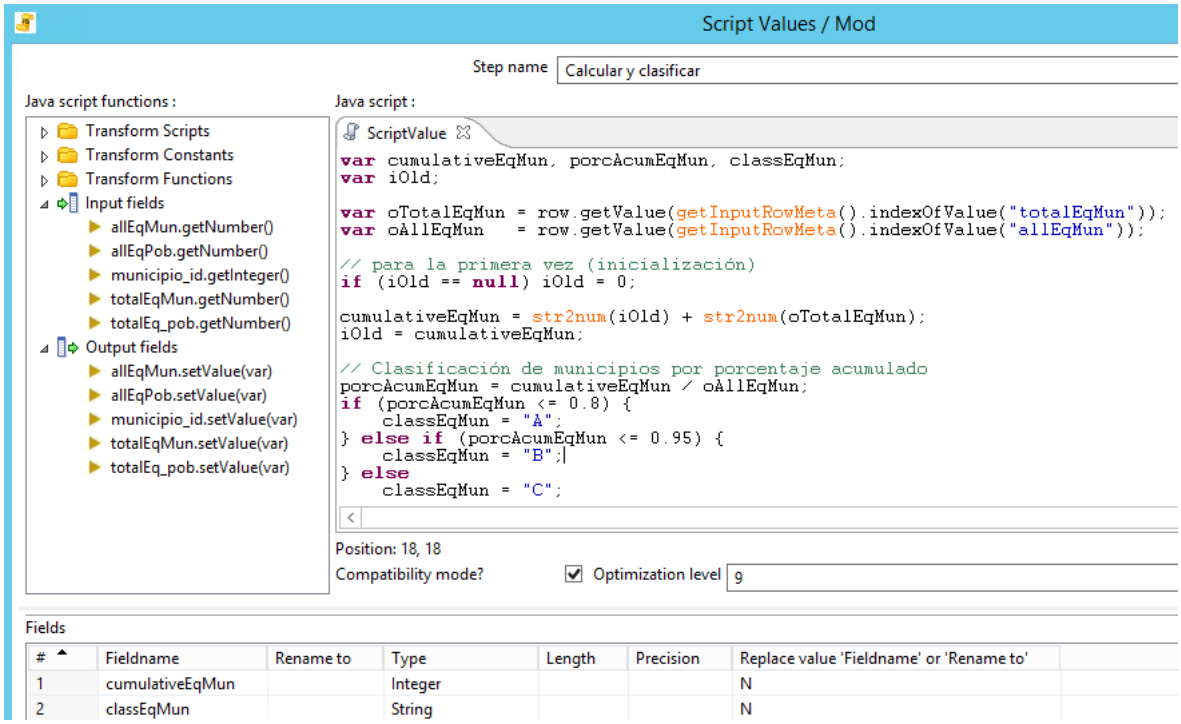
#	Name	Subject	Type
1	allEqMun	totalEqMun	Sum
2	allEqPob	totalEq_pob	Sum

e. Una vez calculados los “grandes totales”, se unen por combinación cartesiana al flujo para su posterior cálculo (ahora, cada tupla debería contener los atributos “allEqMun” y “allEqPob”).

f. Seguidamente, el flujo vuelve a dividirse en dos (sin ningún tipo de filtro, son dos flujos idénticos). Los cálculos que vienen a continuación son análogos, consisten en ordenar adecuadamente los datos para crear un nuevo atributo de valor acumulado. El motivo tiene que ver con el propio concepto de categorización: si la categoría A implica que un (aproximadamente) 20% de los municipios van a ser responsables del 80% de los resultados, significa que ese 20% de municipios “top” van a ser responsables del 80% del valor de alguno de los 2 grandes totales (“allEqMun” o “allEqPob”).

   En otras palabras, los municipios se ordenan (por “totalEqMun” o por “totalEq_pob”), luego se crea el atributo acumulado (“cumulativeEqMun” y “cumulativeEqPob” respectivamente). Y, sobre la marcha, se va comparando ese valor acumulado con el total (“allEqMun” y “allEqPob” respectivamente). La comparación se realiza dividiendo los valores acumulados por los totales (por ejemplo “cumulativeEqMun” / “allEqMun”). Dependiendo del valor encontrado, sabremos si pertenece a la categoría A, B o C (si la lista se ordena de forma descendente, entonces los primeros valores son clase A, hasta llegar al tope 0.8, luego B hasta 0.95 y el resto C).

   Los scripts son idénticos estructuralmente:



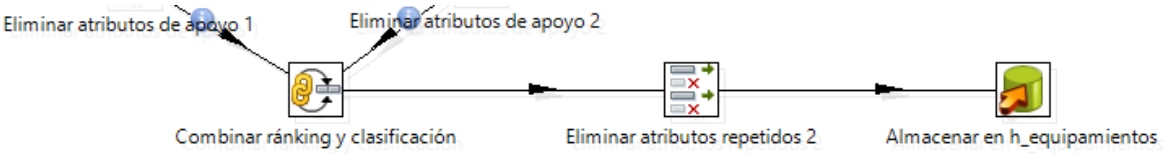
g. El resto de transformaciones se encargan de realizar las operaciones oportunas para unificar los flujos de datos, de modo

que tengamos una lista de 947 municipios con dos categorizaciones:

i. classEqMun: categorización por número de equipamientos.

ii. classEqPob: categorización por equipamientos disponibles por habitante del municipio.

4. La última parte sólo consiste en la combinación de los flujos resultantes del ranking y de la categorización y su posterior volcado en la tabla de hechos “h_equipamientos”.



#	municipio_id	categ_id	eqMunCount	poblacion	rank	classEqMun	classEqPob
1	1	99	1	170	431	C	B
2	1	101	1	170	336	C	B
3	2	2	1	12125	63	A	C
4	2	17	1	12125	44	A	C
5	2	26	1	12125	117	A	C
6	2	29	1	12125	88	A	C
7	2	62	1	12125	49	A	C
8	2	65	2	12125	107	A	C

Como se puede observar en esta tabla final, ambas categorizaciones devuelven resultados muy distintos para cada municipio.

4.2.13 Estimar pernoctaciones

En este procedimiento se van a generar una serie de atributos que serán claves para la generación de campos calculados en el cubo OLAP que se alimenta de la tabla de hechos “h_pernoctaciones”. Estos son “plazasMesTotal”, “plazasMesLibres”, “est_pern_local” y “est_pern_extr”. Es mejor explicarlos sobre la marcha siguiendo el esquema de este proceso ETL:

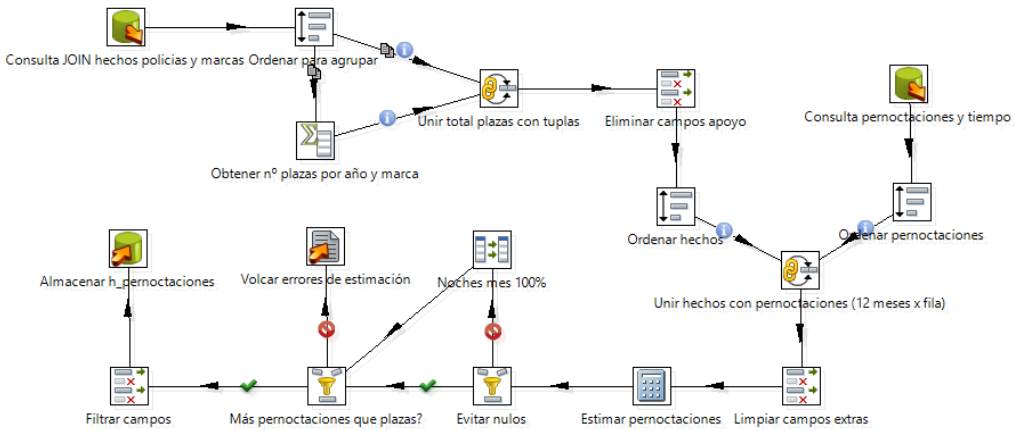


Ilustración 27 - ETL Estimar Pernoctaciones

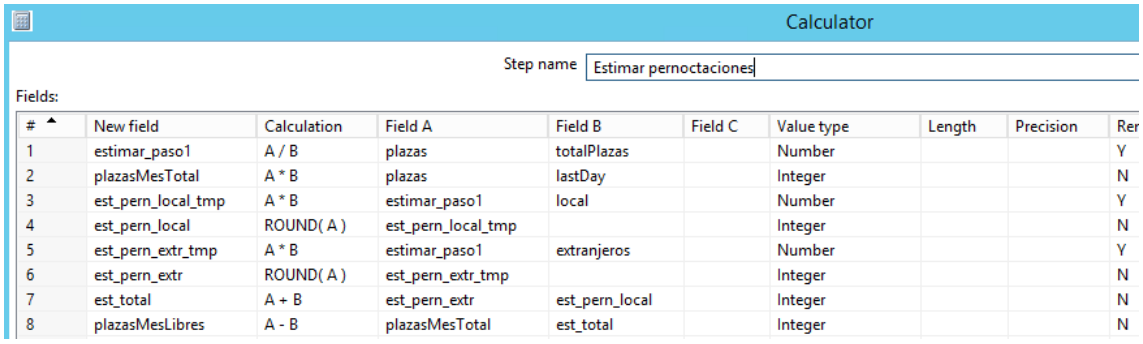
El proceso realiza lo siguiente:

1. Se tiene como punto de partida la tabla de hechos “h_policias”, ya que contiene las dimensiones “d_lugares” y “d_establecimientos” con los datos en su nivel de granularidad más fino.
2. El siguiente paso es calcular el número de plazas total disponible en cada marca turística (de todos los establecimientos) y que se denominará “totalPlazas”. Al recombinarlo con el flujo de datos permitirá que cada tupla tenga un tipo de establecimiento con sus plazas para ese municipio y, además, el “totalPlazas” de la marca a la que pertenece ese municipio.

   La idea que hay detrás de estos cálculos es análoga a la que se planteó en los efectivos policiales: si hay que desagregar datos de pernoctaciones y estos están a nivel de marca turística, entonces la contribución (reparto proporcional) de cada tipo de establecimiento de cada municipio debe plantearse respecto de “totalPlazas”.
3. El siguiente paso es consultar la tabla de apoyo “tmp_pernoctaciones” y combinar los datos de pernoctaciones (españoles y extranjeros) con el flujo principal de datos.

   Hay que tener presente en este punto que el flujo que viene de la consulta de “h_policias” tiene temporalidad anual, mientras que los datos de “tmp_pernoctaciones” tienen temporalidad mensual. En otras palabras, se producirá una “explosión de tuplas” en la combinación (12 tuplas por cada tupla de la consulta “h_policias”). Esto puede dar una idea de lo que sucederá cuando se haga la estimación con los viajeros (que incluye una dimensión más, la de equipamientos).
4. Llegados a este punto, habrá subconjuntos de tuplas de municipios con establecimientos que tienen en común tres datos, el “totalPlazas”, “pernoctacionesEspañoles” y “pernoctacionesExtranjeros”. Así pues, llega el momento en que deben desagregarse las pernoctaciones.

   Conviene aclarar que los cálculos que hay a continuación no van a generar parejas de atributos con cotas superior e inferior como en el caso de los efectivos policiales (lo cual habría sido una buena idea).


Calculator

Step name: Estimar pernoctaciones

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Rer
1	estimar_paso1	A / B	plazas	totalPlazas		Number			Y
2	plazasMesTotal	A * B	plazas	lastDay		Integer			N
3	est_pern_local_tmp	A * B	estimar_paso1	local		Number			Y
4	est_pern_local	ROUND(A)	est_pern_local_tmp			Integer			N
5	est_pern_extr_tmp	A * B	estimar_paso1	extranjeros		Number			Y
6	est_pern_extr	ROUND(A)	est_pern_extr_tmp			Integer			N
7	est_total	A + B	est_pern_extr	est_pern_local		Integer			N
8	plazasMesLibres	A - B	plazasMesTotal	est_total		Integer			N

5. Ahora la explicación de los cálculos:
   a. “estimar_paso1” es la contribución que tiene el tipo de establecimiento del municipio respecto del total de plazas que hay en la marca turística.

b. “plazasMesTotal” indica qué cantidad de pernoctaciones puede absorber un tipo de establecimiento del municipio en el mes que se está considerando (hay que recordar que cada tupla contiene un dato de pernoctaciones para un mes específico, de ahí la importancia del atributo “lastDay”).
c. “est_pern_*_tmp” es la estimación de las pernoctaciones (* = españoles o extranjeros) que resulta de multiplicar “estimar_paso1” por el dato de pernoctaciones correspondiente. Es un cálculo de reparto proporcional sencillo:

$$est\_pern\_local = \frac{plazas}{totalPlazas} * pernoctaciones\_local$$

d. “est_pern_*” es el redondeo del cálculo anterior. Hay que darse cuenta que este tipo de redondeo va a generar tasas de error que pueden ser apreciables a medida que se ejecuten operaciones de agregación en los datos.
e. “est_total” es la suma de las estimaciones de pernoctaciones de residentes españoles y extranjeros (o estimación de pernoctaciones a secas).
f. “plazasMesLibres” viene a ser la variable que permitirá controlar si esta operación ha generado un error conceptual del tipo: no es posible que, para el tipo de establecimiento que se está considerando, haya más pernoctaciones que “plazasMesTotal” (o como reza el dicho “no hay cama para tanta gente”).

6. Una vez realizados los cálculos, hay que controlar que el valor del atributo “plazasMesLibres” no sea nulo, hecho que ocurre cuando tenemos tuplas que contienen valores de plazas y establecimientos pero no de pernoctaciones (porque no hay datos del INE para la zona turística en las fechas consideradas).

   Para resolver este hecho, se divide el flujo de tuplas con los nulos para realizar una transformación de copia del valor “plazasMesTotal” a “plazasMesLibres”. Recuérdese que este atributo indica la cantidad de pernoctaciones que no se han realizado y que era capaz de absorber el tipo de establecimiento de ese municipio (al ser idéntico al valor de “plazasMesTotal” equivale a decir que nadie ha pernoctado en este municipio).
7. El siguiente paso consistirá en analizar si el valor de “plazasMesLibres” es negativo. En caso de ser así, se habrá encontrado un error que no debería haber sido posible y se registra en un archivo log con el nombre “EstimacionesPernoctacionesErroneas”.
8. Finalmente, se almacenan los datos en “h_pernoctaciones”.

4.2.14 Estimar viajeros

Es el último proceso ETL de la carga inicial y el que genera la tabla de hechos ("h_viajeros") de mayor volumen debido a las combinaciones de las cuatro dimensiones.

La principal misión de este procedimiento es crear los atributos "est_viaje_local" y "est_viaje_extr" que corresponden a las estimaciones de viajeros residentes en España y en el extranjeros respectivamente. El punto de partida de este proceso es similar a la estimación de las pernoctaciones, sin embargo, en esta ocasión se usará un sistema diferente de reparto proporcional, el cual incluirá un elemento arbitrario (peso) que conferirá una menor o mayor importancia a según qué equipamientos, de modo que desbalanceará la predilección de los viajeros hacia unos u otros municipios dependiendo de cuán atractivos sean los equipamientos para estos.

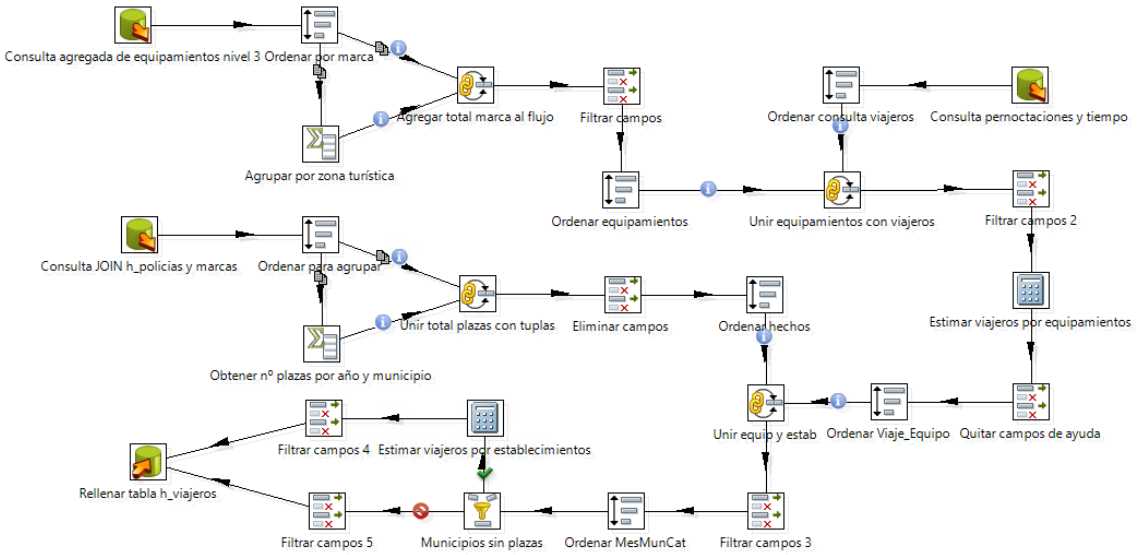
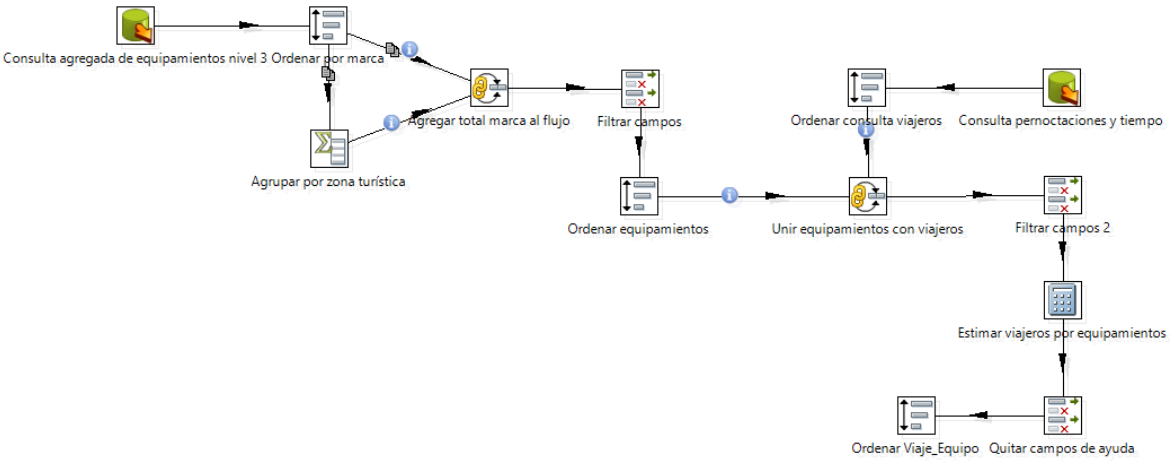


Ilustración 28 - ETL Estimar viajeros

Se procederá a explicar con detalle. El proceso tiene tres entradas de datos:

1. Consulta agregada de equipamientos nivel 3: contiene los datos de "tmp_equipamientos_full" agregados por municipio y categoría de equipamiento. La cantidad de dichos equipamientos se almacena en el atributo "total".
2. Consulta de pernoctaciones y tiempo: son los datos de viajeros ofertados por el INE.
3. Consulta JOIN "h_policias" y marcas: es la misma consulta que la original del proceso ETL "Estimar pernoctaciones" pero con ligeras modificaciones para adecuar los cálculos a la estimación de viajeros.

Ahora se explicará el proceso en tres bloques:



1. Lo primero que conviene tener en cuenta de la consulta agregada de equipamientos de categoría 3, es la necesidad de mantener clara la integridad referencial de la dimensión "d_equipamientos" con las categorías tal como están almacenadas en "tmp_equipamientos_full". Para ello, es necesaria una cláusula WHERE que iguale en ambas tablas las categorías 2 y 3. Tener cuidado en este punto se debe a que el descriptor "cat_nivel3" no es un campo de valores únicos, existen varios valores repetidos y es necesario diferenciarlos apoyándose en "cat_nivel2".

Table input

Step name: Consulta agregada de equipamientos nivel 3

Connection: TFC

SQL

```
SELECT l.id AS municipio_id, t1.categ_id, l.marca, t1.total,
    (t1.total * t1.pesoLocal) AS tot_eq_mun_local,
    (t1.total * t1.pesoExtra) AS tot_eq_mun_extra
FROM (
    SELECT f.municipio, f.cat_nivel3, e.id as categ_id,
        e.pesoLocal, e.pesoExtra, COUNT(e.id) AS total
    FROM equipamientos_full f, equipamientos e
    WHERE f.cat_nivel2 = e.cat_nivel2
    AND   f.cat_nivel3 = e.cat_nivel3
    GROUP BY f.municipio, e.id) AS t1
JOIN lugares l ON l.municipio = t1.municipio
ORDER BY l.id, cat_nivel3
```

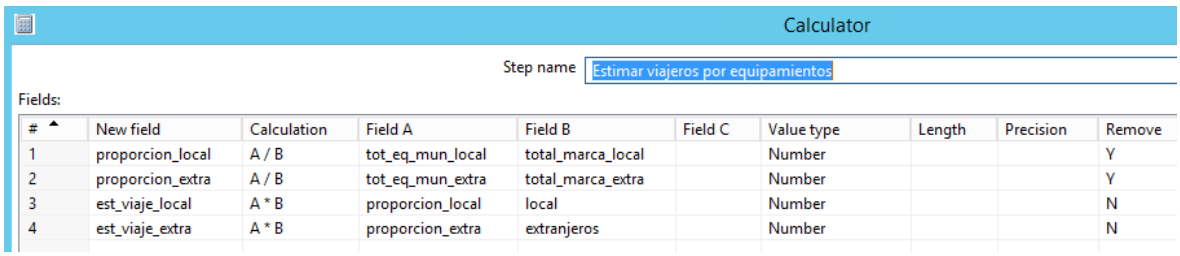
   a. Además, con esta consulta se generan dos nuevos atributos que serán clave en la primera estimación: "tot_eq_mun_local" y "tot_eq_mun_extra", que corresponden con el sumatorio de los pesos de los equipamientos (de una categoría determinada) de un municipio para residentes de España y residentes en el extranjero respectivamente. Nótese que ya a partir de este punto comienzan a prevalecer aquellos municipios que tengan los equipamientos más atractivos (al tener valores más altos).

   b. En la transformación "Agrupar por zona turística" lo que se realiza es la suma agregada de todos los pesos de cada equipamiento y municipio agrupándolos por su zona turística. El objetivo es tener el peso total ("total_marca_local" y

"total_marca_extra") para los dos tipos de viajeros y que serán la base para realizar los repartos proporcionales.

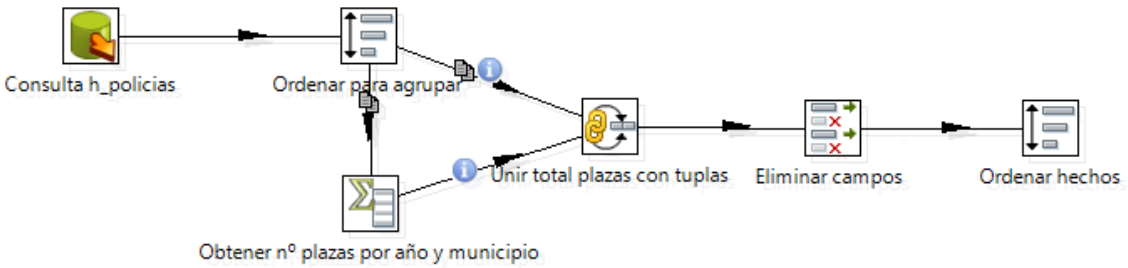
c. Seguidamente se realiza un JOIN con los datos de viajeros del INE. Por un lado hay 19339 tuplas de equipamientos por municipio que, combinadas con los 36 meses de los datos del INE (del año 2011 al 2013) genera da un total de 19339 * 36 = 696204 tuplas.

d. La primera operación de estimación en base a los equipamientos debe tener en cuenta a los locales y a los extranjeros. Sin embargo, esta vez no se realizará ningún redondeo de los resultados, ya que hay prevista una segunda estimación en base a los diferentes tipos de establecimientos.



#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove
1	proporcion_local	A / B	tot_eq_mun_local	total_marca_local		Number			Y
2	proporcion_extra	A / B	tot_eq_mun_extra	total_marca_extra		Number			Y
3	est_viaje_local	A * B	proporcion_local	local		Number			N
4	est_viaje_extra	A * B	proporcion_extra	extranjeros		Number			N

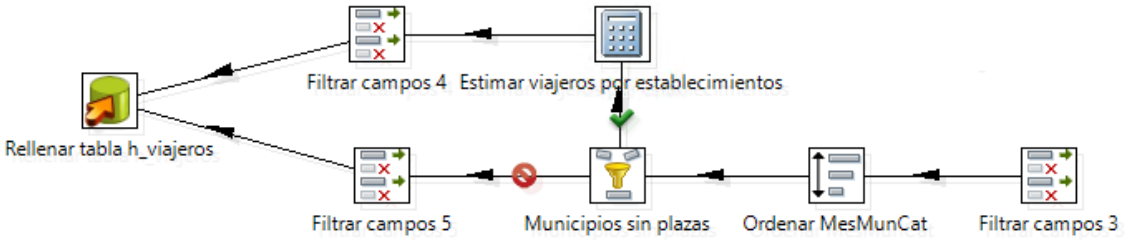
2. En el segundo bloque se parte de la rama que consulta los datos de "h_policias".



a. La consulta de partida no tiene nada en especial, como mucho destacar que el atributo "plazas" es de vital mportancia para la siguiente estimación.

b. Una característica de la suma agregada que hay a continuación, es que no se realiza agrupando por zona turística (marca) sino por municipios. Esto es así porque no hay que desagregar los datos de viajeros desde el nivel de zonas turísticas, ya se ha realizado esta operación en el flujo del primer bloque. Aparte, lo que se busca principalmente con esta consulta es obtener los datos necesarios para calcular las proporciones relativas de las plazas a nivel municipio, de modo que en el tercer bloque, cada tipo de equipamiento de un determinado municipio debe repartir los viajeros entre los establecimientos disponibles.

3. En el tercer y último bloque se realizan las operaciones de estimación de viajeros realizando un reparto proporcional en base a las plazas disponibles por tipo de establecimiento y municipio:



a. Lo primero que se realiza es la combinación de los flujos de los dos bloques anteriores, de modo que se produce la máxima multiplicidad posible al combinar todos los equipamientos de un municipio con los establecimientos del mismo. En esta ocasión hay 6944 tuplas que resultan de la combinación de municipios con sus establecimientos disponibles y las 696204 que provienen del primer bloque. La combinación final da un total de 2.210.928 tuplas (da una media aproximada de 3 tipos diferentes de establecimientos por municipio).

b. El siguiente paso es el filtro “Municipios sin plazas”, que impide que la estimación se realice en municipios que tienen el tipo de establecimiento especial “None” (daría errores de división por cero al tener un total de plazas igual a cero).

   Nota importante: conviene aclarar en este punto que existen municipios sin ningún tipo de establecimientos que tienen viajeros, dado que estos fueron calculados en primera instancia en base a los equipamientos disponibles. En este punto, se entiende que los viajeros se sienten atraídos por los equipamientos, razón por la que no se realizó este filtro en el primer bloque. Para el requerimiento “Promedio de viajeros por tipo de establecimiento y comarca”, existirá el establecimiento “None”, que indicará el promedio de viajeros que fueron a esa comarca sin estar relacionados directamente con algún tipo de establecimiento.

c. “Estimar viajeros por establecimiento”: se dispone de las plazas de un tipo determinado de establecimiento en un municipio y del total de plazas de dicho municipio, por tanto, se puede obtener la proporción de plazas que aporta cada tipo de establecimiento.

Calculator

Step name: Estimar viajeros por establecimientos

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remov
1	proporcionPlazas	A / B	plazas	totalPlazas		Number			N
2	est_local_tmp	A * B	proporcionPlazas	est_viaje_local		Number			N
3	est_local	ROUND(A)	est_local_tmp			Integer			N
4	est_extra_tmp	A * B	proporcionPlazas	est_viaje_extra		Number			N
5	est_extra	ROUND(A)	est_extra_tmp			Integer			N

d. El último paso consiste en unir ambos flujos y almacenarlos en la tabla de hechos “h_viajeros”.

Error en la estimación de viajeros por reparto con pesos

Incluir un sistema de pesos arbitrario no deja de presentar controversias debido a los desajustes que provocará, especialmente, en aquellos municipios que no cuentan con "equipamientos atractivos" o bien que disponen de muchos tipos de establecimientos para un muy reducido número de viajeros que han sido repartidos proporcionalmente por una gran cantidad de equipamientos (lo que origina una árbol de equipamientos y establecimientos donde cada rama sufre un redondeo en la estimación):

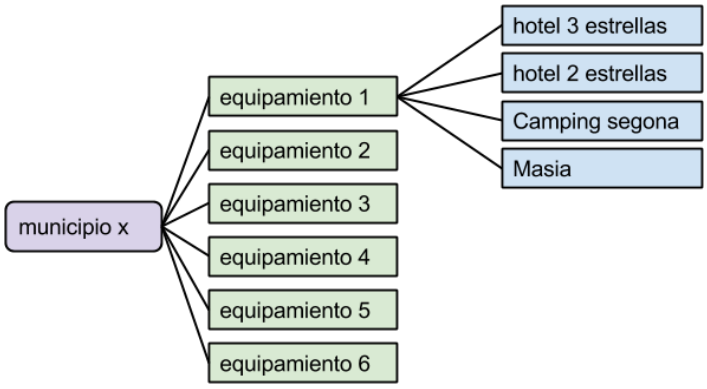


Ilustración 29 - Ejemplo ramificación equipamientos y establecimientos

Tomando como ejemplo la ilustración anterior, tenemos un "municipio x" con seis tipos distintos de equipamientos. Si este municipio cuenta además con 4 tipos de establecimientos, tenemos que se generarán 6 * 4 = 24 tuplas por cada mes, lo que equivale a decir que habrá 24 estimaciones que redondear (porque el número de viajeros es un valor entero).

Dicho de otro modo, la desagregación de los datos del INE en estas estimaciones conlleva a una serie de errores de cálculo que deben ser tenidos en cuenta, especialmente si se usa un reparto proporcional no lineal (por pesos) como el que se ha efectuado en este trabajo.

Si se realiza la operación inversa y se agrega el total de viajeros de cada municipio por zona turística, se podrán obtener los valores de viajeros con los que poder comparar los originales del INE y, por ende, es posible cuantificar los errores generados por estas estimaciones.

Concretamente, se trata de la siguiente consulta:

```
SELECT t2.*, p.viajerosLocal, p.viajerosExtranjeros,
       ((p.viajerosLocal - t2.localTotal) / p.viajerosLocal * 100) AS errorLocal,
       ((p.viajerosExtranjeros - t2.extraTotal) / p.viajerosExtranjeros * 100) AS
       errorExtra
FROM (
       SELECT t.mes, l.marca, sum(t1.local) AS localTotal, sum(t1.extra) AS extraTotal
       from (
              SELECT h.tiempo id, h.municipio id, sum(est viaje local) AS local,
                     sum(est_viaje_extr) AS extra
              FROM h_viajeros h
              GROUP BY h.tiempo_id, h.municipio_id
       ) AS t1
       JOIN d_lugares l ON l.id = t1.municipio_id
```

```
        JOIN d_tiempo t ON t.id = t1.tiempo_id
        GROUP BY t1.tiempo_id, l.marca
) AS t2, tmp_pernoctaciones p
WHERE t2.mes = p.mes
AND t2.marca = p.marca
HAVING errorExtra > 5
ORDER BY errorExtra DESC
```

Con la cláusula "HAVING errorExtra > 5" conseguimos filtrar aquellos errores que superan una variación del 5% respecto del valor original. A continuación, una muestra de los resultados:

mes	marca	localTotal	extraTotal	viajerosLocal	viajerosExtranjeros	errorLocal	errorExtra
2013M01	Pirineus	29983	1292	30485	1590	1.6467	18.7421
2013M11	Pirineus	27800	1795	28232	2182	1.5302	17.7360
2011M11	Pirineus	26395	1478	26962	1787	2.1030	17.2916
2012M01	Pirineus	34907	1585	35259	1916	0.9983	17.2756
2012M12	Pirineus	40599	1627	41008	1963	0.9974	17.1167
2011M12	Pirineus	44151	1581	44535	1902	0.8622	16.8770
2012M11	Pirineus	24965	1941	25448	2291	1.8980	15.2772
2013M01	Terres de l'Ebre	4688	812	4859	946	3.5192	14.1649
2011M01	Pirineus	34565	2218	35023	2573	1.3077	13.7971

Ilustración 30 - Tabla errores de estimación en viajeros

La explicación de los resultados de la consulta:

- localTotal: suma de las estimaciones de viajeros residentes en España.
- extraTotal: suma de las estimaciones de viajeros residentes en el extranjero.
- viajerosLocal: valor original del INE de viajeros residentes en España.
- viajerosExtranjeros: valor original del INE de viajeros residentes en el extranjero.
- errorLocal: porcentaje de error entre los viajeros españoles estimados y los del INE.
- errorExtra: porcentaje de error entre los viajeros extranjeros estimados y los del INE.

Como se puede observar, se ha hecho especial hincapié en los valores de error de viajeros extranjeros, ya que es donde se produce una mayor asimetría en la asignación de pesos por equipamientos. En algunos casos se llegan a errores de casi el 19%, si bien conviene tener en cuenta que las cifras de viajeros extranjeros del INE para estas zonas turísticas son especialmente bajas en comparación con los residentes españoles (y tal como se explicó, repartir pocos viajeros entre muchos equipamientos y establecimientos conduce a muchos más errores de redondeo).

Una técnica de reparto como la ya vista en los efectivos policiales probablemente devolvería resultados muchos más próximos a los reales .

4.3 Actualizaciones anuales

Según el esquema visto en la ilustración 2, el presente apartado de actualizaciones anuales sería un paso posterior y, de hecho, su implementación fue una de las últimas tareas efectuadas en este trabajo.

Dado que la longitud de la memoria no permite entrar en mucho detalle, se procederá a explicar de manera somera los aspectos más relevantes de los procesos ETL.

Para el caso del procedimiento ETL de actualizaciones anuales, los archivos fuente se encuentran situados en la carpeta “TFG_final/Actualizacion” y los procesos ETL en “TFG_final/ETL_update”. El archivo encargado de iniciar el proceso está directamente en la carpeta raíz “TFG_final” y se llama “Update_Job.kjb”.

Como el lector debe haber apreciado, lo de actualización anual parece condicionar bastante la forma en que se pueden actualizar los datos. En efecto, el título anual implica que los archivos fuente que se vayan a usar deben tener todos los datos de un año concreto (el que se vaya a actualizar). Esto implica que los datos de pernoctaciones y viajeros deben tener las cuatro variables implicadas por los 12 meses del año; además, deben estar presentes los archivos del IDESCAT para ese mismo año (“Policies locals”, “Poblacion” e “Infraestructuras turisticas”). Sobre el formato del nombre de cada archivo ya se explicará debidamente en cada esquema ETL.

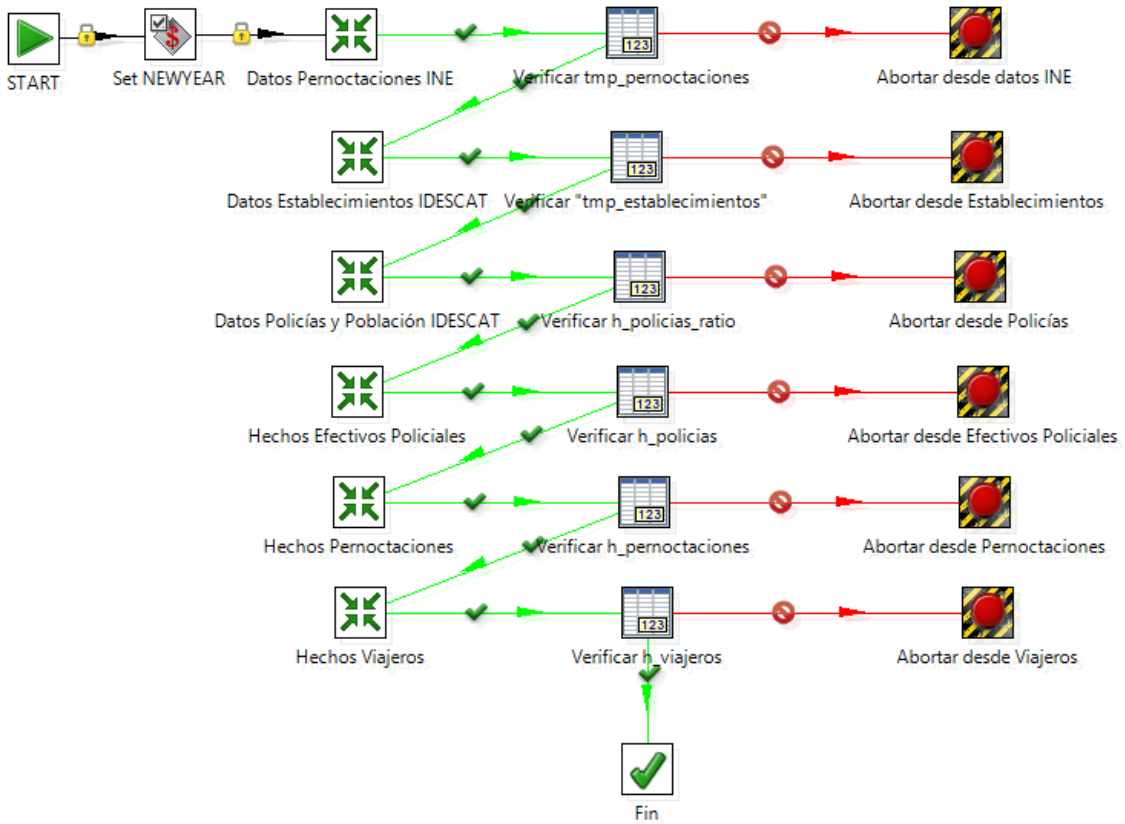


Ilustración 31 - ETL Actualización Job (Update_Job.kjb)

Como se puede observar, el esquema del Job de actualización únicamente tiene seis transformaciones, sus seis respectivas comprobaciones y un proceso abortador en caso de que no superen el visto bueno de cada verificador. Además, nada más empezar se crea una variable global "NEWYEAR" que contendrá el año de trabajo de la actualización, cuyo valor será asignado en la primera transformación y será usado por cada verificador para comprobar si las tablas actualizadas contienen efectivamente datos del año que se está actualizando.

Realmente, el sistema no es nada robusto y no hace todas las comprobaciones que realmente serían necesarias para asegurar el éxito de la actualización, pero contiene lo mínimo para informar que la tarea no se pudo llevar a cabo con pleno éxito (el tema de tener archivos mezclados de varios años o con datos incompletos / parciales es un tema que podría llevar muchas horas de trabajo).

Un último detalle no menos importante, es que este proceso **actualiza**, lo que significa que podría ejecutarse tantas veces como se quisiera para cualquier año aunque ya exista en el almacén de datos. Esto podría suponer un problema si alguien lleva a cabo una actualización porque desea borrar datos incorrectos y quiere refrescar con nuevos datos: si en los nuevos datos no hay combinaciones de dimensiones que coincidan con lo que se desea eliminar, las viejas tuplas permanecerán en el almacén de datos y convivirán con los nuevos datos. Para llevar a cabo un "borrado" de posibles tuplas erróneas, lo mejor sería empezar de cero con la carga inicial y luego realizar todas las actualizaciones anuales que se necesiten (no es un método fantástico, pero es hasta la fecha de implementación la única forma posible de resolverlo).

4.3.1 Datos Pernoctaciones INE

Se encarga de extraer los datos del INE a partir, esta vez, de ficheros CSV.

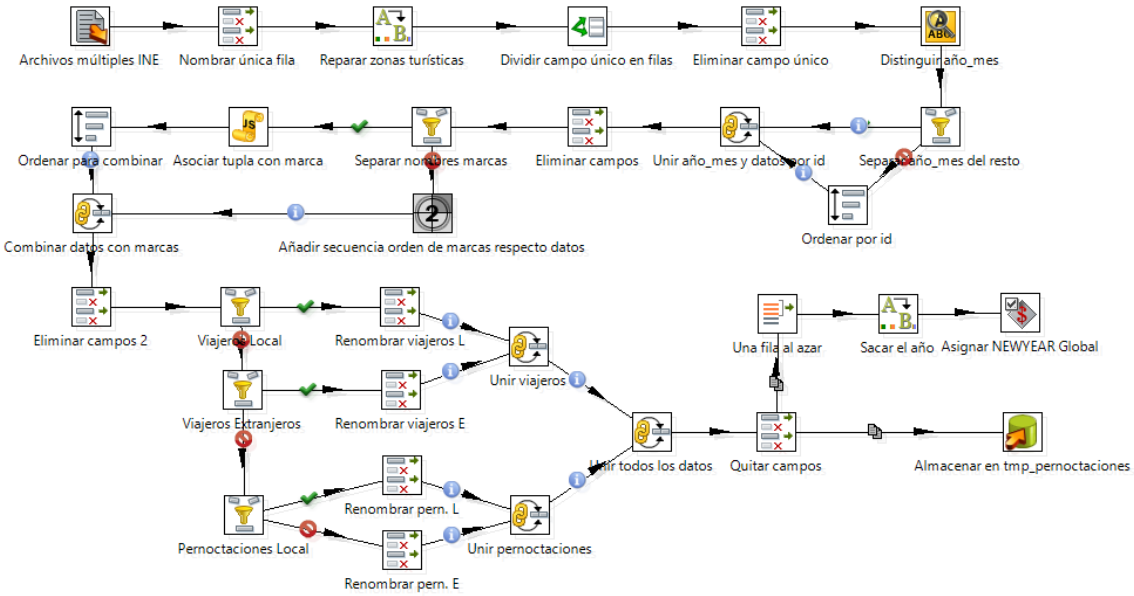
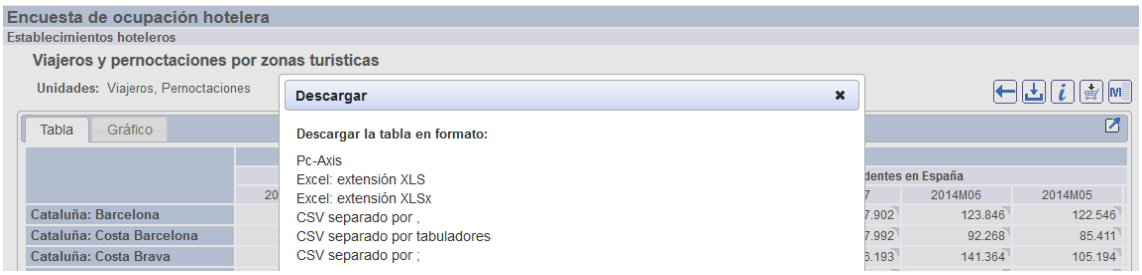


Ilustración 32 - ETL2 Datos pernoctaciones INE

El principal problema que tiene obtener datos del INE, es que la página web devuelve consultas limitadas, de tal modo que si queremos todos los datos de viajeros y pernoctaciones para residentes en España y en el extranjero para los 12 meses del año, es posible que tengamos que exportar los datos en varios archivos. Por este motivo, la entrada de datos permite la posibilidad de incluir varios archivos, cuyo formato ha de ser "tabla_2039*.csv", que es el nombre del archivo por defecto al exportar desde el INE. La idea es que los archivos estén nombrados de la siguiente forma:

- tabla_2039.csv
- tabla_2039 (1).csv
- tabla_2039 (cualquier comentario).csv
- …

El formato csv debe ser el ofertado por el INE (con separación de campos por ";"):



El motivo de elegir CSV y no una hoja de cálculo Excel se debe a las limitaciones en la importación de datos de estas últimas: es necesario especificar las "coordenadas" de la esquina superior izquierda de la tabla (fila, columna) y no siempre vamos a tener la tabla como la espera Kettle, esto es, puede fallar precisamente la primera celda al no tener el nombre de la columna, que en este caso particular debería ser "Zonas turísticas".

Sin embargo, uno de los problemas que tiene importar datos en formato csv, es que hay que darle muchas más vueltas para conseguir los metadatos (nombres de columnas, el tipo de tabla que se está importando, el año de los datos…). Ésta es la razón por la que el proceso ETL de una fuente CSV se ve más complejo que su análogo en Excel.

En cuanto al proceso en sí, tenemos:

1. Nombrar única fila: al principio, sólo tenemos un campo. Esto se debe a que cada fila del archivo CSV del INE contiene datos de diferentes filas de una tabla bien estructurada.
2. Reparar zonas turísticas: lo que se busca es reparar algunas palabras de las zonas turísticas que ya se sabe que causarán conflictos. Además, se elimina el nombre de las columnas agrupadoras porque estorban tal y como fueron exportadas.
3. Dividir campo único en filas: en este punto sólo tenemos los nombres de columnas que interesan (los meses del año) y los datos, todos separados

por ";". Se pasan absolutamente todos los valores de celda a filas; esto es, una celda = una fila, de modo que se consigue una secuencia (columna "id") de filas cada una con un dato (columna "datos"). Conviene tener en cuenta que hay una tercera columna con los datos originales que se repite tantas veces como celdas contenga.

4. Eliminar campo único: se elimina la columna con datos originales.
5. Distinguir año_mes: se crea una nueva columna ("result") que evalúa si la tupla contiene datos o nombre de columna (se establece a "true" si es un nombre de columna tipo "aaaaMmm").
6. Separar año_mes del resto: se divide el flujo para tener los nombres de columna por un lado y los datos por el otro.
7. Unir año_mes y datos por id: la parte clave de todas estas transformaciones. En el punto 3, se creó una secuencia por cada campo que se dividía. La idea es que cada campo que se divide contiene 48 celdas que corresponden a los 12 meses del año * las 4 variables de estudio (viajeros y pernoctaciones * españoles y extranjeros). Además, puede darse el caso de que exista una celda más (la 49) que corresponde al nombre de una zona turística. En cualquier caso, siempre se respeta el hecho de que las 48 primeras celdas corresponden a las secuencias correctas de nombres de columna y datos, por lo que los identificadores generados sirven para combinar de nuevo los datos.
   Importante darse cuenta que se unen los datos con su correspondiente mes del año, pero aún falta relacionar a qué variable de estudio corresponde cada par mes-dato, y a qué zona turística corresponde la secuencia global. Nótese que, exceptuando los datos (que ya los tenemos controlados) el resto de tuplas no han sufrido ningún tipo de ordenamiento, es decir, conservan el orden de secuencia en que fueron leídos del archivo CSV original.
   a. Ordenar por id: el id indica el orden de cada secuencia durante la división de los datos en el punto 3. Es necesario que ahora sean ordenados para la operación JOIN que viene a continuación. Lo más importante que hay que comentar en este punto es que, a pesar de que haya ids que se repitan, el ordenamiento es secuencial, es decir, se respeta el orden de llegada de las tuplas, de modo que corresponde con la secuencia de zonas turísticas que se van encontrando a lo largo de todas las tuplas.
8. Eliminar campos: sólo interesan los campos "id", "mes" y "data". Nótese que ahora hay sucesiones de x tuplas con un mismo "id" y que tienen el mismo mes del año. Es decir, habrá tantas tuplas con el mismo id como zonas turísticas se hayan importado (si el archivo contiene 4 zonas turísticas, se observará que hay series de 4 tuplas con el mismo id y mes del año).
9. Separar nombres marcas: en el punto 7, la combinación se realiza con una operación JOIN OUTER, de modo que combinan los meses con los datos (con ids del 1 al 48) y quedan huérfanas las tuplas que no tienen asociados datos (con id 49) que corresponden con las zonas turísticas. Es

decir, las zonas turísticas son tuplas cuyo id es null, motivo por el que se pueden separar fácilmente.

a. Asociar tupla con marca: en realidad aún no se pueden asociar, pero lo que sí se puede hacer es asignar un número de secuencia que permitirá la posterior combinación. Y esto es lo que hace este script: comprueba el campo "id" para ir asignando una secuencia en un nuevo campo "idMarca". Cada vez que "id" cambia se reinicia la secuencia.

   Además, aprovechando que aún se dispone del campo "id" y que estamos en un script, se crea un atributo más ("iTipoMedida") que indicará a qué variable de estudio corresponde cada tupla. Para ello hay que entender que el campo "id" (que va del valor 1 al 48), tiene la secuencia correcta de aparición de los datos, de modo que:

   i. Del 1 al 12: son datos de viajeros residentes en España.
   ii. Del 13 al 24: son datos de viajeros extranjeros.
   iii. Del 25 al 36: son datos de pernoctaciones de españoles.
   iv. Del 36 al 48: son datos de pernoctaciones de extranjeros.

b. Ordenar para combinar: como en este momento ya tenemos identificados el orden secuencial original por zona turística de cada tupla, es posible realizar esta operación de ordenamiento sin miedo a perder la estructura original de los datos. La idea es ordenar por "idMarca" para asociar de forma definitiva cada tupla a su correspondiente zona turística.

c. Añadir secuencia orden de marcas respecto datos: simplemente se trata de asignar un identificador ("idMarca") secuencial por orden de aparición (tal como llegaron respecto del archivo original).

10. Combinar datos con marcas: los dos flujos ya son combinables en este punto y asocian las zonas turísticas con los datos.
11. Eliminar campos 2: sólo interesa tener los campos "id", "mes", "data", "marca" e "iTipoMedida".
12. Lo que sigue a continuación son filtros para separar flujos de datos y transformaciones para cambiar el nombre de la columna "iTipoMedida" a la correspondiente columna estática conocida ("viajerosLocal", "viajerosExtranjeros", "pernoctacionesLocal" y "pernoctacionesExtranjeros").
13. Finalmente, todos los datos de los diferentes flujos se van recombinando por "mes" y "marca", de modo que la estructura final de los datos concuerda con la esperada por la tabla "h_pernoctaciones".
14. Una fila al azar: este paso es clave de cara al proceso global. Se trata de escoger una fila (una muestra) para los siguientes pasos, que se encargarán de obtener el año a partir del nombre de la columna (ya que ésta tiene el formato aaaaMmm). Finalmente, se asigna el valor encontrado (el año) a la variable global NEWYEAR.

A partir de este momento, todos los procesos serán evaluados con el valor de esta variable. Por ejemplo, si “NEWYEAR=2014”, entonces, para todos los procesos subsiguientes, sólo se darán por válidos si introducen datos correspondientes al año 2014.

4.3.2 Datos establecimientos IDESCAT

Este proceso extrae los datos de establecimientos del IDESCAT en formato CSV.

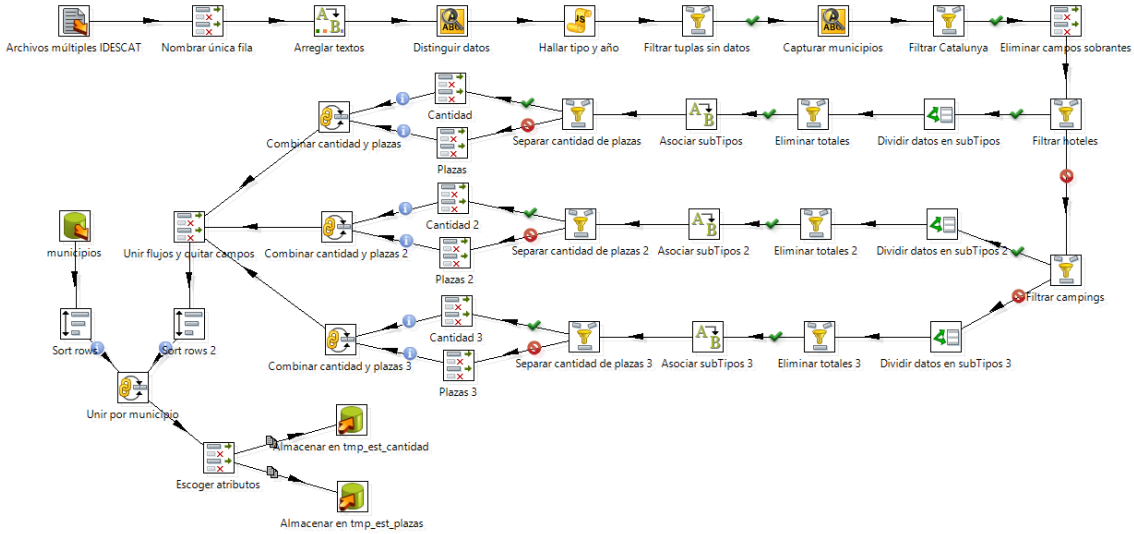


Ilustración 33 - ETL 2 Datos establecimientos IDESCAT

1. Archivos Múltiples IDESCAT: en este caso hay una probabilidad del 100% de que haya que hacer la lectura a partir de 6 archivos (los nombres de archivo deben ser de la forma “Infraestructura tur*.csv”), correspondientes a los obtenibles por la web del IDESCAT que son:
    a. Establecimientos hoteleros
    b. Plazas hoteleras
    c. Establecimientos de campings
    d. Plazas de campings
    e. Establecimientos de turismo rural
    f. Plazas de turismo rural

    Nota: de la web del IDESCAT se obtiene un libro de Excel con todas las tablas necesarias (una por hoja del libro). Por tanto, es necesario exportar los datos al formato CSV (con separación “;”). La razón es análoga al INE: la inflexibilidad a la hora de importar los datos de la hoja de cálculo debido a los problemas de formato estructural de los mismos.
2. Nombrar fila única: sólo se dispone de un campo por fila (se ha provocado esta situación para evitar la dispersión de datos por diferentes campos).
3. Arreglar textos: esta transformación está pensada para que la tupla que contiene el año de los datos tenga únicamente el valor del año en la

única celda disponible (sin texto adicional que dificulte su posterior extracción).

4. Distinguir datos: siguiendo el mismo procedimiento que en el INE, se trata de distinguir lo que son datos de lo que no. En este caso particular, los nombres de las futuras columnas sí son conocidos (no como en el INE), así que esta transformación más bien diferencia los metadatos de las tablas respecto de los datos en sí.
5. Hallar tipo y año: la mejor solución encontrada en este punto pasa por el uso de un script. Lo que se realiza básicamente es la identificación secuencial del tipo de tabla y del año de la misma. Del proceso 4 se distingue con la columna "result" lo que no son datos, de modo que en el script se evalúan estas tuplas y se obtienen los metadatos.

   Conviene darse cuenta que para identificar las tablas ha sido necesario "hardcodear" los nombres de las mismas en el script en un array, de modo que se almacena en la variable "tipo" la posición de cada nombre de tabla encontrado. Esta solución permite poder tratar el conjunto de archivos fuente sin tener necesidad de conocer previamente qué archivo corresponde a qué tabla.

   Por último, de las tuplas que contienen metadatos se extrae también el año, que se guarda en la variable "anyo". Tanto "tipo" como "anyo" se propagan a lo largo de los datos hasta encontrar el siguiente encabezado de tabla o final de secuencia de tuplas.
6. Filtrar tuplas sin datos: como en el anterior paso ya se han extraído los metadatos, se puede prescindir de las tuplas que conforman los encabezados de tabla (y de paso, otras tuplas que hay al final de cada tabla).
7. Capturar municipios: ahora "field1" contiene los datos de interés, que están compuestos por el municipio y los datos estadísticos en sí separados por ";". Así pues, se separa en una primera columna los nombres de municipios mediante el uso de expresiones regulares.
8. No se realiza el mismo proceso con el resto de datos por dos razones: la primera, porque hay diferente número de columnas por cada uno de los 3 tipos de establecimientos principales (hoteles, campings y turismo rural); y la segunda, porque precisamente por la misma razón, tienen nombres de columna diferentes.
9. Filtrar Catalunya: esta transformación específica elimina la tupla final que contiene cada tabla con los totales de cada columna para la provincia Cataluña.
10. Eliminar campos sobrantes: sólo interesan los atributos "municipio", "anyo", "tipo" y "data".
11. A continuación viene el tratamiento específico para cada tipo de establecimiento. Cada rama sigue exactamente las mismas transformaciones, lo único que diferencia unas de otras son los nombres de las futuras columnas que indican el subtipo de establecimientos. A continuación se explica el ejemplo de la rama "camping"

a. Filtrar campings: se filtra a través del campo “tipo”. Hay que recordar que son 6 tablas en total, 2 por cada tipo de establecimiento, ya que una tabla contiene el número de subtipo de establecimientos y la otra contiene el número de plazas disponible de cada subtipo de establecimiento.
b. Dividir datos en subTipos: con la transformación “Split field to rows” usando como separador “;” se consigue crear tantas tuplas como valores separados por “;” haya en la celda “data”. En el caso de los campings son 5 valores, luego habrá 5 veces más tuplas en este flujo.

   Adicionalmente, conviene tener en cuenta que se crea el campo “subTipo”, que contendrá la secuencia de cada uno de los valores encontrados en “data”. Esta secuencia es conocida y corresponde a los nombres de los subtipos de establecimientos (“Luxe”, “Primera”, “Segona”, “Tercera” y “Total”).
c. Eliminar totales 2: es decir, eliminar las tuplas que tienen el subtipo “Total”.
d. Asociar subTipos 2: se crea el campo “subTipoFull” con el nombre real del subtipo de establecimiento (con real se quiere decir el que corresponde con la dimensión “d_establecimientos”).
e. Separar cantidad de plazas: éste es el flujo de datos de camping, que a su vez se puede subdividir en “número de establecimientos” (tipo 3) y “plazas” (tipo 4) para este caso particular.
f. Renombrar el atributo “tipo”: como “plazas” y “cantidad”
g. Combinar cantidad y plazas: ahora se combinan los dos flujos (por “municipio”, “anyo” y “subTipo”) de modo que cada tupla contenga los dos atributos “plazas” y “cantidad”.
h. Unir flujos y quitar campos: se unifican los flujos de hoteles, campings y turismo rural. De paso se eliminan los campos repetidos o prescindibles.
i. Lo que queda se sustituir los nombres de municipio por los id correspondientes de la dimensión “d_lugares” y almacenar los datos en las dos tablas de apoyo “tmp_est_cantidad” y “tmp_est_plazas”. La razón de hacerlo en dos tablas en vez de una sola es histórica (debido a los procesos ETL de carga inicial que usan como fuente hojas de cálculo).

Por razones de espacio, no es posible continuar explicando las decisiones tomadas en cada paso para el resto de procesos ETL de actualización. El resto de diagramas ETL se pondrán en el anexo.

Una aclaración importante respecto de los últimos tres procesos ETL, es que a la hora de escribir los datos a las tablas, se realiza una operación INSERT/UPDATE, de modo que se crean nuevas tuplas o se actualizan las que correspondan.

5 Los cubos OLAP

Realmente, esta sección debería haber sido titulada “Informes” o algo similar, pero lo cierto es que la solución final está tan enfocada a los resultados de análisis a través de cubos OLAP que se ha optado por nombrarlo de este modo.

En la ilustración 2, tras los procesos ETL y la carga al almacén de datos, quedaría por explicar en qué consisten los 3, 4 y 6. Para ello, es necesario entender de qué herramientas se disponen y cómo se interrelacionan entre sí.

5.1 El entorno

Antes de empezar, es conveniente tener una definición (extraída de la Wikipedia) sobre la definición de OLAP:

“Es una solución utilizada en el campo de la llamada Inteligencia de negocios (o Business Intelligence) cuyo objetivo es agilizar la consulta de grandes cantidades de datos. Para ello utiliza estructuras multidimensionales (o Cubos OLAP) que contienen datos resumidos de grandes Bases de datos o Sistemas Transaccionales (OLTP). Se usa en informes de negocios de ventas, marketing, informes de dirección, minería de datos y áreas similares.”

Los “*datos resumidos de grandes Bases de datos*” hace referencia a los resultados de los procesos ETL que se han obtenido a través de las fuentes de datos (que a su vez proceden del INE e IDESCAT, que son los sistemas transaccionales y que quedan fuera de la “jurisdicción” de este trabajo).

En cuanto a la referencia “Business Intelligence”, se trata del software necesario para llevar a cabo los análisis mediante las mencionadas consultas a grandes cantidades de datos. Según la ilustración 2, correspondería con el servidor “Pentaho BI”, al cual puede accederse a través de su interfaz vía web:

Ilustración 34 - Pentaho BI interfaz

El siguiente punto "*Para ello utiliza estructuras multidimensionales (o Cubos OLAP)*", correspondería en la ilustración 2 con el paso 3, que consiste en la realización de un esquema Mondrian. Éste puede realizarse de forma manual o mediante el uso de alguna herramienta destinada a este fin ("Schema Workbench"). Una vez que se crea un esquema, hay que incluirlo en el repositorio de esquemas del servidor Pentaho. Si todo ha ido bien los usuarios podrán tener acceso a los cubos OLAP (dependiendo de los roles y las directivas de seguridad definidas en el esquema Mondrian y en el propio servidor Pentaho).

Los puntos 4 y 6 de la ilustración 2 vendrían a ser el proceso de consulta en 2 pasos, de modo que hay que tener en cuenta quién interactúa con qué:

- En el paso 6, los usuarios acceden a los cubos OLAP a través de una herramienta de visualización llamada Saiku. Ésta se integra como un plugin dentro del entorno de Pentaho y es accesible a través del menú de la interfaz de la ilustración 34:

  Saiku interacciona a través de consultas MDX contra los cubos OLAP y el motor Mondrian que se aloja en Pentaho es el encargado de interpretar estas consultas.

- En el paso 4, el servidor Pentaho actúa como un cliente del servidor MySQL. El motor Mondrian, una vez interpreta la consulta MDX realizada por el usuario, realiza una consulta SQL a la base de datos MySQL, de modo que recupera los valores necesarios y que debe procesar el motor Mondrian, el cual a su vez devuelve los resultados a Saiku que finalmente los muestra con el formato adecuado al usuario.

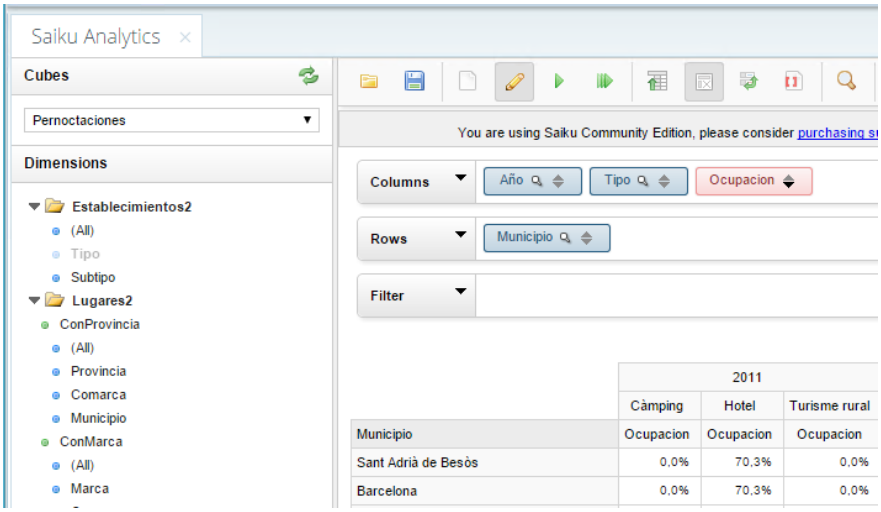
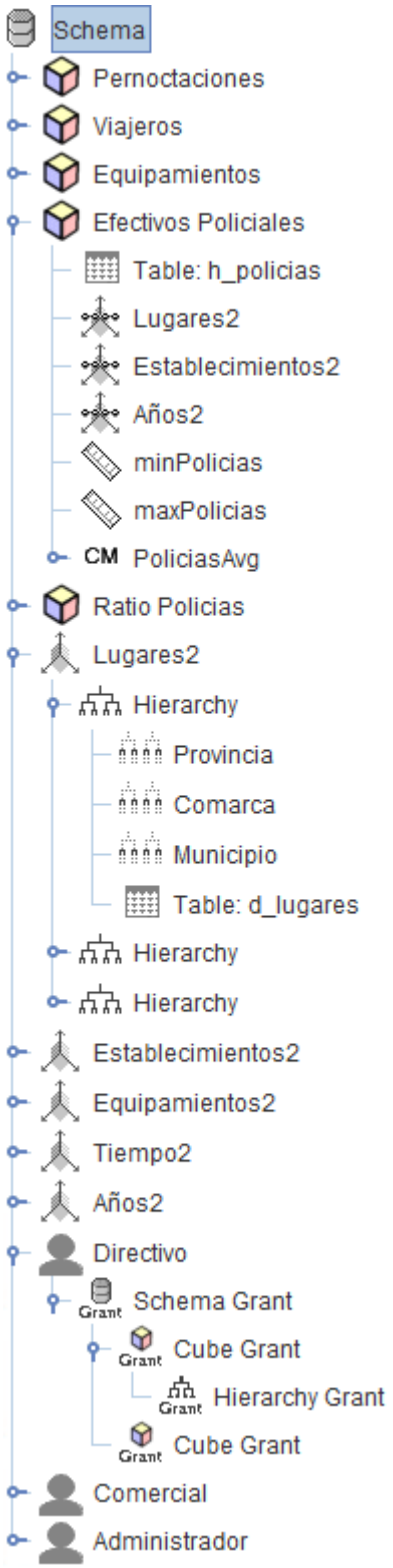


Ilustración 35 - Ejemplo interfaz Saiku Analytics

5.2 El esquema Mondrian

Se trata de un archivo en formato XML que define todo lo necesario para conformar los cubos OLAP. En este trabajo, el esquema consta de las siguientes secciones:

- Schema: es la raíz, define el nombre del esquema Mondrian.
  - Dimension: lo primero que se define son las dimensiones. Éstas podrían haber sido definidas dentro de los propios cubos, pero al definirlas fuera de estos es posible ahorrar espacio y tener un mejor control de posibles errores o cambios.
    - Jeraquías: cada dimensión puede tener a su vez una o varías jerarquías. Las razones para tener varias jerarquías puedenser múltiples. En el caso del presente trabajo, fue necesario crear tres jerarquías en la dimensión "d_lugares", dos por los solapamientos territoriales entre "marcas" y "provincias", y una por problemas inherentes a la versión de evaluación de Pentaho, la cual tiene algunas limitaciones que impiden el funcionamiento esperado de las cláusulas ORDER de las consultas.
  - Cube: en este trabajo, cada cubo OLAP está intrínsecamente relacionado con una única tabla de hechos, comparte una serie de dimensiones y posee una serie de medidas.
    - Table: relaciona el cubo con una tabla de hechos.
    - DimensionUsage: todas las entradas de este tipo relacionan alguna dimensión definida previamente con la tabla de hechos del cubo, de manera que se define únicamente el campo de la tabla de hechos que contiene la clave foránea (la propia dimensión tiene definida la clave primaria).
    - Measure: las medidas son los atributos de las tablas de hechos. Definir varias medidas equivale a seleccionar los atributos que se desea que estén disponibles en el cubo.
    - CalculatedMember: son medidas que no están definidas en la tabla de hechos y que se crean en el cubo OLAP a medida que se navega por éste. Pueden considerarse como fórmulas que usan las medidas previamente definidas para dar lugar a cálculos de interés. Nótese que las medidas se agregan dependiendo de las dimensiones consultadas, de modo que las medidas calculadas deberían ofrecer los valores esperados según las dimensiones que se han usado.
  - Role: este elemento tiene que ver con las directivas de seguridad que se quieren establecer en cualquier nivel del esquema, de modo que se puede limitar el acceso de los usuarios a información sensible. Los parámetros básicos de acceso son "all" y "none" (acceso permitido y no permitido respectivamente).



- SchemaGrant: es el nivel más alto de acceso y vendría a definir las dos filosofías propias de sistema de seguridad en cuanto a accesos. Si se establece “all” a este nivel, entonces lo lógico es que se creen directivas que restrinjan el acceso a determinados cubos (y, por defecto, los demás cubos son accesibles). Si por el contrario se establece a “none”, habrá que crear los accesos a los cubos que se le permita consultar. En este trabajo se emplea la directiva “none”, ya que es el mejor modo de controlar que nadie tenga acceso a cualquier cubo que se cree a posteriori.
    - CubeGrant: tal como se definió SchemaGrant, estas entradas estarán disponibles para dar acceso (“all”) al cubo para el rol en cuestión.

Existen más reglas que permiten restringir el acceso a granos más finos del cubo (DimensionGrant, HierarchyGrant y MemberGrant), pero para este trabajo basta como ejemplo permitir el acceso a los diferentes cubos.

Además, como se puede comprobar en la imagen lateral que ha acompañado este texto, las dimensiones se han tenido que nombrar añadiendo un 2 al final. Esto se debe a los problemas técnicos que se han encontrado en la realización de este trabajo y a los continuos bugs acumulados en el repositorio de esquemas Mondrian.

5.3 Los campos calculados

Estos elementos del esquema Mondrian se merecen una sección aparte debido a la importancia que tienen. Y es que algunos de los requisitos propuestos por GLTF implican necesariamente el uso de campos calculados, como podría ser la medida “Ocupacion” del cubo “Pernoctaciones”.

```
<Cube name="Pernoctaciones">
  <Table name="h_pernoctaciones"/>
  <DimensionUsage name="Lugares2" source="Lugares2" foreignKey="municipio_id"/>
  <DimensionUsage name="Establecimientos2" source="Establecimientos2"
     foreignKey="establecimiento_id"/>
  <DimensionUsage name="Tiempo2" source="Tiempo2" foreignKey="tiempo_id"/>
  <Measure name="Espa&#241;oles" column="est_pern_local" aggregator="sum"/>
  <Measure name="Extranjeros" column="est_pern_extr" aggregator="sum"/>
  <Measure name="Plazas x dia total" column="plazasMesTotal" aggregator="sum"/>
  <Measure name="Plazas x dia libres" column="plazasMesLibres" aggregator="sum"/>
  <CalculatedMember name="Pernoctaciones" dimension="Measures">
    <Formula>Measures.[Espa&#241;oles] + Measures.Extranjeros</Formula>
    <CalculatedMemberProperty name="FORMAT_STRING" value="0"/>
  </CalculatedMember>
  <CalculatedMember name="Ocupacion" dimension="Measures">
    <Formula>IIF([Measures].[Plazas x dia total] > 0,
       1 - ([Measures].[Plazas x dia libres] / [Measures].[Plazas x dia total]), 0)
    </Formula>
```

```
      <CalculatedMemberProperty name="FORMAT_STRING" value="0.0%"/>
    </CalculatedMember>
    <CalculatedMember name="% Espa&#241;oles sobre total" dimension="Measures">
      <Formula>Measures.[Espa&#241;oles] / (Measures.[Espa&#241;oles] +
       [Lugares2.ConMarca].[All Lugares])</Formula>
      <CalculatedMemberProperty name="FORMAT_STRING" value="0.0%"/>
    </CalculatedMember>
    <CalculatedMember name="% Extranjeros sobre total" dimension="Measures">
      <Formula>Measures.Extranjeros / (Measures.Extranjeros + [Lugares2.ConMarca].[All
       Lugares])</Formula>
      <CalculatedMemberProperty name="FORMAT_STRING" value="0.0%"/>
    </CalculatedMember>
    <CalculatedMember name="% Pernoctaciones sobre total" dimension="Measures">
      <Formula>(Measures.Extranjeros + Measures.[Espa&#241;oles]) /
               (Measures.Extranjeros + Measures.[Espa&#241;oles] +
               [Lugares2.ConMarca].[All Lugares])
      </Formula>
      <CalculatedMemberProperty name="FORMAT_STRING" value="0.0%"/>
    </CalculatedMember>
  </Cube>
```

Ilustración 36 - Esquema Mondrian (cubo Pernoctaciones)

En la ilustración 36 se puede observar parte del esquema Mondrian, concretamente la definición del cubo “Pernoctaciones”.

Aparte de las cuatro medidas que ya contenía la tabla de hechos “h_pernoctaciones” (atributos “est_pern_local”, “est_pern_extr”, “plazasMesTotal” y “plazasMesLibres”), fueron necesarios los campos calculados siguientes:

- [Pernoctaciones]: es la suma de las medidas [Españoles] y [Extranjeros] (o lo que es lo mismo, la suma de las estimaciones “est_pern_local” y “est_pern_extr”).
- [Ocupacion]: primero comprueba si la medida [Plazas x Dia Total] es distinta de 0, caso de ser cierto, realiza la operación que indica qué porcentaje de las plazas se han ocupado con pernoctaciones.
- [% * sobre el total]: donde “*” puede ser “Españoles”, “Extranjeros” o “Pernoctaciones”. Realiza el cálculo de porcentaje de pernoctaciones “*” respecto de todos los municipios.

  Conviene darse cuenta que para este cálculo se está usando la jerarquía [Lugares2.ConMarca], lo que implica que si se cruzan estas medidas con la jerarquía [Lugares2.ConProvincia] no se van a obtener los esultados esperados. Sería deseable transmitir este cálculo a todas las jerarquías implicadas en la dimensión “Lugares2”, pero esto no ha sido posible debido a problemas técnicos.

Otro campo calculado que merece especial mención es [PoliciasAvg] que se encuentra en el cubo “Efectivos Policiales”, cuya fórmula:

```
<Formula>([Measures].[minPolicias] + [Measures].[maxPolicias]) / 2</Formula>
```

Permite el cálculo de lo que sería la medida [Policias] (el atributo “policias” desagregado a nivel de establecimientos y comentado en la sección ETL).

5.4 Los roles definidos

En el presente trabajo se han definido tres usuarios en el servidor Pentaho BI. Cada uno de ellos debe (o debería) tener al menos un rol definido que le confiere permisos de operación como la publicación de contenido, la ejecución de cosultas, creación de contenido… Estos son:

Cuenta	Rol	Permisos
TFC_Admin	Administrador	Todos
TFC_Comercial	Comercial	Leer contenido Publicar contenido Crear contenido
TFC_Directivo	Directivo	Leer contenido Publicar contenido Ejecutar Programar contenido Crear contenido

Para mayor comodidad, la contraseña de todas las cuentas es la misma: 1234.

Los roles “Administrador”, “Comercial” y “Directivo” están definidos también en el esquema de Mondrian, ya que en el archivo de configuración “Objects.spring.xml” de Pentaho está activo el ‘bean id=Mondrian-UserRoleMapper”’, el cual establece que el motor Mondrian mapee los roles de Pentaho con los del esquema Mondrian.

5.5 Los análisis de cubos OLAP

Éste sería el apartado final con el que se concluye este trabajo y la razón de su existencia. La organización GLTF proporcionó un listado de requisitos y espera que el almacén de datos creado junto con las herramientas explicadas en esta sección puedan dar respuesta a sus necesidades. Así pues, a continuación se mostrará el listado de consultas pedidas con su respectiva formalización en el lenguaje de consulta MDX. Para ver los resultados de las consultas y cómo se llegan hasta ellas es mejor ver el vídeo que se adjuntará como presentación de la memoria.

- Total de pernoctaciones estimadas por municipio: se usan las medidas [Españoles], [Extranjeros] o la suma de ambas [Pernoctaciones]. Está permitida la navegación por todos los niveles de todas las dimensiones implicadas (no hay dimensión equipamientos).

  Ejemplo 1: estimaciones de pernoctaciones en los municipios de Barcelona y Girona en los tres primeros meses del 2011:

```
SELECT
NON EMPTY Hierarchize (
Union(CrossJoin({[Tiempo2].[2011].[2011M01]},{[Measures].[Españoles]}),
Union(CrossJoin({[Tiempo2].[2011].[2011M01]}, {[Measures].[Extranjeros]}),
Union(CrossJoin({[Tiempo2].[2011].[2011M02]}, {[Measures].[Españoles]}),
Union(CrossJoin({[Tiempo2].[2011].[2011M02]}, {[Measures].[Extranjeros]}),
Union(CrossJoin({[Tiempo2].[2011].[2011M03]}, {[Measures].[Españoles]}),
```

```
CrossJoin({[Tiempo2].[2011].[2011M03]}, {[Measures].[Extranjeros]})))))))) ON
COLUMNS,
       NON EMPTY
       Hierarchize(Union(CrossJoin({[Lugares2.ConMarca].[Barcelona].[Barcelonès]
       .[Barcelona]}, [Establecimientos2].[Subtipo].Members),
       CrossJoin({[Lugares2.ConMarca].[Costa Brava].[Gironès].[Girona]},
       [Establecimientos2].[Subtipo].Members))) ON ROWS
FROM [Pernoctaciones]
```

Ejemplo 2: reconstrucción de la tabla de pernoctaciones del INE con los meses del año en columnas y las zonas turísticas en filas:

```
SELECT
NON  EMPTY  Union(CrossJoin([Tiempo2].[Mes].Members,  {[Measures].[Españoles]}),
CrossJoin([Tiempo2].[Mes].Members, {[Measures].[Extranjeros]})) ON COLUMNS,
       NON EMPTY [Lugares2.ConMarca].[Marca].Members ON ROWS
FROM [Pernoctaciones]
```

- Tanto por ciento de ocupación por marca turística: para esta consulta se usa el campo calculado [Ocupacion]. Está permitida la navegación de todos los niveles de las dimensiones a excepción de la jerarquía [Lugares2.ConProvincia].

  Ejemplo: ocupación por zona turística para todos los meses del año 2011

```
SELECT
NON EMPTY Hierarchize(Union(CrossJoin({[Tiempo2].[2011].[2011M01]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M02]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M03]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M04]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M05]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M06]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M07]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M08]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M09]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M10]},
{[Measures].[Ocupacion]}), Union(CrossJoin({[Tiempo2].[2011].[2011M11]},
{[Measures].[Ocupacion]}), CrossJoin({[Tiempo2].[2011].[2011M12]},
{[Measures].[Ocupacion]}))))))))))))) ON COLUMNS,
       NON EMPTY {Hierarchize({[Lugares2.ConMarca].[Marca].Members})} ON ROWS
FROM [Pernoctaciones]
```

- Top ten de municipios por franja de pernoctaciones: implica el uso de la función “TopCount” en la medida [Pernoctaciones], si bien es importante señalar que no debe mezclarse la dimensión tiempo en la misma columna que la medida, ya que Saiku no parece ordenar correctamente los resultados resultantes de la operación crossjoin. Para filtrar por tiempo, es mejor usar el tiempo como “slicer”.

  Ejemplo: los 10 municipios top del primer mes del 2011.

```
SELECT
NON EMPTY {Hierarchize({[Measures].[Pernoctaciones]})} ON COLUMNS,
       NON EMPTY
       TopCount({Hierarchize({[Lugares2.SoloMunicipo].[Municipio].Members})},
       10, [Measures].[Pernoctaciones]) ON ROWS
FROM [Pernoctaciones]
```

```
WHERE {[Tiempo2].[2011].[2011M01]}
```

- Tanto por ciento de pernoctaciones de un municipio sobre el total: se pueden usar los 3 campos calculados [% Españoles sobre total], [% Extranjeros sobre total] y/o [% Pernoctaciones sobre total]. El cálculo se realiza con la medida que proceda en cada municipio ("Españoles", "Extranjeros" o la suma de ambos) dividido por el sumatorio total de dicha medida en todos los miembros de la dimensión Lugares (definido como "All Lugares").

  Ejemplo: las tres medidas para el primer mes del 2011 y escogiendo el top 10 de los resultados en función de [% Pernoctaciones sobre total].

```
SELECT
NON EMPTY {Hierarchize({{[Measures].[% Españoles sobre total], [Measures].[%
Extranjeros sobre total], [Measures].[% Pernoctaciones sobre total]}})} ON
COLUMNS,
       NON EMPTY
       TopCount({Hierarchize({[Lugares2.ConMarca].[Municipio].Members})}, 10,
       [Measures].[% Pernoctaciones sobre total]) ON ROWS
FROM [Pernoctaciones]
WHERE {[Tiempo2].[2011].[2011M01]}
```

- Distribución mensual y estacionalidad de las pernoctaciones: básicamente se trata de usar las medidas [Españoles], [Extranjeros] y/o [Pernoctaciones] y enfrentarlas en consultas sencillas con las otras dimensiones.

  Ejemplo: distribución mensual de españoles por provincia en el 2011.

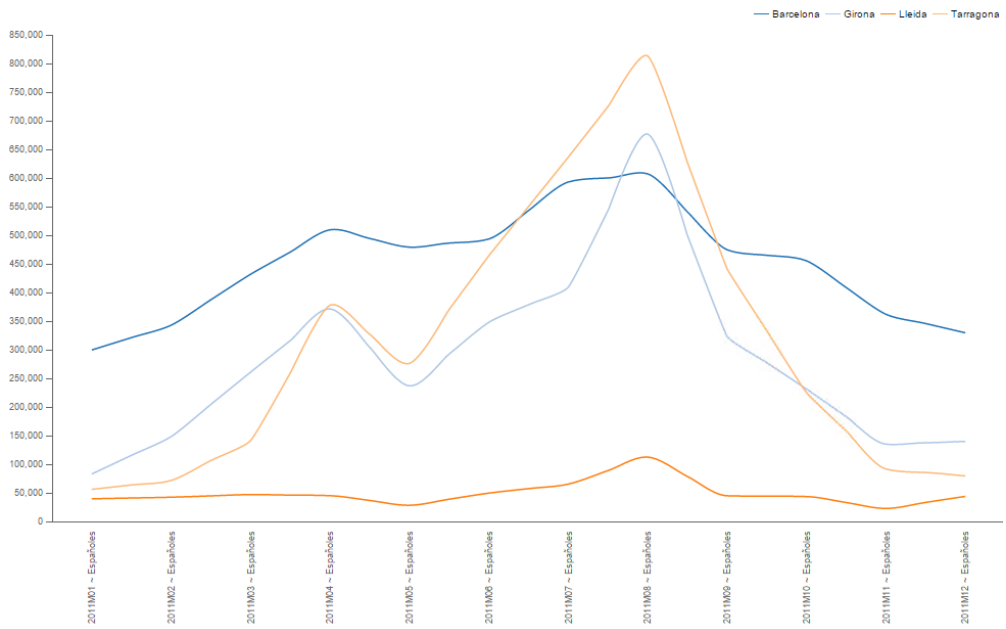


Ilustración 37 - Distribución mensual pernoctaciones españoles

- Promedio de viajeros por tipo de establecimiento y comarca: en esta ocasión se trabaja con el cubo “Viajeros” y con las medidas [Españoles] y [Extranjeros]. Se puede navegar por todas las dimensiones.

  Ejemplo: la consulta pedida, por tipo de establecimiento y comarca.

```
SELECT
NON EMPTY CrossJoin([Establecimientos2].[Subtipo].Members,
{[Measures].[Viajeros]}) ON COLUMNS,
       NON EMPTY CrossJoin([Lugares2.ConMarca].[Comarca].Members,
       {[Tiempo2].[2011].[2011M01]}) ON ROWS
FROM [Viajeros]
```

- Ranking de municipios por categoría de equipamientos: se trabaja con el cubo “Equipamientos” y con la medida [rank]. Es importante señalar que sólo se puede trabajar con la jerarquía [Lugares2.SoloMunicipios], ya que de otro modo Saiku no devuelve los valores ordenados de la forma esperada (no interesa romper la jerarquía con “BASC”).

```
SELECT
NON EMPTY {Hierarchize({[Measures].[Ranking]})} ON COLUMNS,
       NON EMPTY Order(CrossJoin([Equipamientos2.ConTodo].[Categoria 3].Members,
       [Lugares2.SoloMunicipo].[Municipio].Members), [Measures].[Ranking], ASC)
       ON ROWS
FROM [Equipamientos]
```

- Categorización de municipios A/B/C: se trabaja con el cubo “Equipamientos” y con las medidas [Categ. Equipo poblacion] y [Categ. numero equipos]. Sólo hay una dimensión (“Lugares2”), de modo que sólo hay un tipo de consulta con mayor o menor información jerárquica, como por ejemplo:

```
SELECT
NON EMPTY {Hierarchize({{[Measures].[Categ. equipo poblacion],
[Measures].[Categ. numero equipos]}})} ON COLUMNS,
       NON EMPTY {Hierarchize({{[Lugares2.ConMarca].[Marca].Members},
       {[Lugares2.ConMarca].[Comarca].Members},
       {[Lugares2.ConMarca].[Municipio].Members}})} ON ROWS
FROM [Equipamientos]
```

- Máximo y mínimo de efectivos policiales por tipología de establecimiento y municipio: el cubo con el que se trabaja es “Efectivos policiales” y las medidas pedidas son [minPolicias] y [maxPolicias]. Existe una tercera medida “de regalo” [PoliciasAvg] que corresponde con el valor medio de las anteriores. La consulta pedida:

```
SELECT
NON EMPTY Hierarchize(Union(CrossJoin([Establecimientos2].[Tipo].Members,
{[Measures].[PoliciasAvg]}), Union(CrossJoin([Establecimientos2].[Tipo].Members,
{[Measures].[minPolicias]}), CrossJoin([Establecimientos2].[Tipo].Members,
{[Measures].[maxPolicias]})))) ON COLUMNS,
       NON EMPTY
```

```
       Filter({Hierarchize({[Lugares2.ConMarca].[Municipio].Members})},
       [Measures].[PoliciasAvg] > 0) ON ROWS
FROM [Efectivos Policiales]
WHERE {[Años2].[2011]}
```

- Ratio de policías locales por habitante: se trabaja con el cubo "Ratio policias" y en principio sólo se pide el campo calculado [Ratio]. Aún así también están disponibles las medidas [Poblacion] y [Policias]; ambas medidas son las que hacen posible que el campo calculado [Ratio] devuelva los valores esperados aún navegando por otros niveles de la jerarquía de "Lugares2".

```
SELECT
NON EMPTY Hierarchize(Union(CrossJoin([Años2].[Año].Members,
{[Measures].[Ratio]}), CrossJoin([Años2].[Año].Members,
{[Measures].[Policias]}))) ON COLUMNS,
       NON EMPTY {Hierarchize({[Lugares2.ConMarca].[Marca].Members})} ON ROWS
FROM [Ratio Policias]
```

6 Conclusiones

El presente trabajo ha significado, entre otras cosas, una buena forma de acercarse a una signatura pendiente que aún no había cursado, a la par que ha servido para entender lo problemático que es crear procesos automatizados que sean capaces de cargar al almacén los datos de interés.

Se ha lidiado con dos tipos de fuentes, hojas de cálculo y archivos CSV. Si bien en principio las hojas de cálculo eran más intuitivas, a la larga se ha visto que los archivos CSV son mucho más flexibles si se tiene conocimiento de cómo empezar a abordarlos y qué estrategias conviene aplicar en según qué orden. Especialmente relevante es la captura de metadatos usando únicamente las herramientas de transformación que proporciona Kettle.

Uno de los principales puntos conflictivos de este trabajo ha sido la desagregación de datos. La distribución de los mismos se ha efectuado usando tres técnicas en tres tablas de hechos distintas:

- Para las pernoctaciones se realizó un reparto proporcional a secas.
- Para los efectivos policiales, el reparto proporcional se realizó de forma que se generan dos atributos con redondeos por abajo y por arriba. Ambas cotas permiten crear un campo calculado que es la media de ambas medidas y que dan el valor más aproximado de la medida real.
- En cuanto a los viajeros, el reparto proporcional se hizo usando pesos en los equipamientos, de modo que la distribución no resulta lineal y permite distribuir los datos de un modo teóricamente más realista.

De todas las técnicas, la que usa cotas inferiores y superiores resulta imprescindible para aminorar las tasas de error generadas por redondeos. Ésta

podría combinarse con el reparto por pesos, ya que el reparto lineal ofrece resultados poco creíbles. Eso sí, conviene recordar que la asignación de pesos es arbitraria y que no resulta fácil decidir qué valores asignar a cada categoría de equipamiento. En este sentido, si se pudiera tener los datos reales de pernoctaciones y viajeros de un año a nivel municipio, podría ser un buen punto de partida para hacer cálculos inversos que permitan determinar la distribución de los pesos. De este modo, con una lista de pesos contrastada con datos reales, se podrían crear desagregaciones que fueran un poco más realistas.

En cuanto a los objetivos y la planificación, el único comentario que se me ocurre es que resulta imposible planificar sin conocimiento del campo. Sólo a medida que he avanzado en el proyecto he podido entender realmente cuáles son los pasos necesarios y dónde es necesario invertir más tiempo. La planificación en sí fue elaborada siguiendo el ejemplo de otros y poco y nada se ha ajustado a la realidad de los hechos. Sin embargo, eso no ha sido impedimento para cumplir con los objetivos de GLTF y la mayor parte de los planteados por el consultor, que con mejor o peor acierto han permitido aumentar la calidad del presente trabajo.

Respecto a los requerimientos planteados por el consultor, no ha sido posible llevar a cabo el control de actualizaciones (llevar un historial de actualizaciones) ni crear un script que facilite la ejecución de los procesos necesarios (comprobación de ficheros, ejecutar archivo Job y mostrar el feedback oportuno para saber si hubo errores de consideración).

Por último, merece la pena comentar que la carga de esquemas Mondrian al repositorio del servidor Pentaho es un punto conflictivo, cualquier error tipográfico o error de sintaxis puede derivar en la imposibilidad de eliminar el esquema del repositorio debido a posibles errores en su almacenamiento. Es más, es posible que este hecho también incida en la aparición de errores inesperados de conflicto de nombres de elementos como pueden ser las dimensiones.

7 Glosario

CSV: Acrónimo inglés de Archivo de datos separados por comas (*Comma-Separated Values*). Es uno de los formatos posibles de los archivos fuente que se usarán para cargar datos al almacén.

ETL: Acrónimo inglés de Extracción, Transformación y Carga de datos (*Extract, Transform & Load)*. Es un proceso clave en el presente trabajo.

GLTF: Grupo Líder en Turismo Familiar, la organización que contrata los servicios de consultoría para la realización de este trabajo.

IDESCAT: Acrónimo catalán de Instituto de Estadística de Cataluña (*Institut d'Estadística de Catalunya*). Es la segunda fuente principal de datos con la que se alimentará el almacén de datos.

INE: Instituto Nacional de Estadística (de España). Es la principal fuente de datos estadísticos de la que se nutre este trabajo (suministra también a IDESCAT).

MDX: Acrónimo inglés de Expresiones Multidimensionales (*MultiDimensional eXpressions*). Es un lenguaje de consulta para bases de datos multidimensionales sobre cubos OLAP.

OLAP: Acrónimo inglés de Procesamiento Analítico en Línea (*On-Line Analytical Processing*). Es una solución informática cuyo punto fuerte es la velocidad de ejecución de consultas de tipo SELECT, en contraposición con OLTP.

OLTP: Acrónimo inglés de Procesamiento de Transacciones En Línea (*On-Line Transaction Processing*). Es la solución informática que se encuentra mayormente en los Sistemas Gestores de Bases de Datos (SGBD). La orientación de estos está sujeta a buscar el mejor rendimiento en las operaciones INSERT, UPDATE y DELETE de los sistemas relacionales, en contraposición con las prestaciones ofrecidas por los sistemas OLAP.

RAD: Acrónimo inglés de Desarrollo Rápido de Aplicaciones (*Rapid Application Development*).

TFG: Trabajo de Fin de Grado.

8 Bibliografía

Libros de texto

- William D. Back, Nicholas Goodman y Julian Hyde; *Mondrian in Action, Open Source Business Analitycs*; Manning Publications Co., 2014.
- Brian C. Smith y C. Ryan Clay; *SQL Server 2008 MDX, Step by Step*; Hitachi Consulting, 2009.

Páginas web **(todas accesibles en la fecha 15/06/2015)**

- Mondrian
  - Manual Schema Workbench: http://mondrian.pentaho.com/documentation/schema_workbench.pdf
  - Documentación oficial: http://mondrian.pentaho.com/documentation/
- IDESCAT
  - [1] Fuente de datos de comarcas y marcas: http://www.idescat.cat/cat/idescat/publicacions/cataleg/pdfdocs/geocat2013s.pdf (página 115)
  - Población de Cataluña: http://www.idescat.cat/pub/?id=aec&n=925
- INE
  - Fuente de datos pernoctaciones: http://www.ine.es/jaxiT3/Tabla.htm?t=2039
- FAMILITUR
  - [2] Informe anual 2011 para sacar ideas de cómo asignar pesos por equipamientos: http://www.iet.tourspain.es/es-ES/estadisticas/familitur/Anuales/Informe%20anual%20de%20Familitur.%20A%C3%B1o%202011.pdf
- Principales webs de referencia (Spoon/Kettle, MDX, Mondrian…)
  - Youtube: http://www.youtube.com
  - Stack Overflow: http://www.stackoverflow.com
  - Foros Pentaho: http://forums.pentaho.com