Construcció i explotació d'un magatzem de dades per a l'anàlisi d'informació sobre allotjaments turístics

Memoria

Ferrán Casasús Rodó
Enginyeria Tècnica d'Informàtica de Sistemes
Consultor: Carles Llorach Rius

Treball Fi de Carrera

Curs 2012-2013 segon semestre

Resumen

Trabajo centrado en el estudio y aplicación práctica de las técnicas conocidas bajo el nombre de "**Almacenes de Datos**" y concretamente en sistemas **ROLAP** para dar respuesta a las necesidades de los entornos decisorios. En él se detalla los requisitos para desplegar un "Análisis de información sobre establecimientos turísticos", se expone la planificación, método basado en UML, el análisis, diseño, los pasos de recuperación de los datos, para su posterior transformación y carga en un base de datos relacional **MySQL** y la presentación de las consultas **OLAP** e informes.

Índice de contenidos

Índice de ilustraciones

Índice de tablas

1. Capítulo. Introducción

Para el trabajo de final de la carrera d'Enginyeria Tècnica d'Informàtica de Sistemes hay que desarrollar un trabajo de síntesis aplicado en un área en concreto, en este caso es el análisis de las técnicas actuales para diseñar bases de datos para un almacén de datos (Data Warahouse), su consulta y manipulación.

1.1. Justificación del proyecto.

El proyecto de "Análisis de información sobre establecimientos turísticos" esta enfocado a plantear un reto con la intención de ofrecer un ejemplo de los problemas más comunes en el desarrollo de "Almacenes de datos".

1.2. Objetivos del TFC

En una sola frase y tal como se expone en el enunciado del trabajo el objetivo principal será poseer experiencia en el diseño, construcción y explotación de un almacén de datos a partir de la información disponible en una base de datos transaccional.

Si lo detallamos un poco más, tememos:

- Conocer e investigar en las técnicas actuales para diseñar bases de datos para un almacén de datos según los principios de los almacenes de datos físicos **ROLAP.** Estar en situación para considerar factores tales como desnormalización de tablas, inclusión de información agregada, históricos, etc.
- Comprender los problemas derivados de la recuperación de datos de múltiples fuentes y formatos distintos y duplicados.
- Aplicar un criterio metedológico para todo el proceso y utilizando una metodología integradora como el **UML** adaptado a los almacenes de datos.
- Estudiar un conjunto de herramientas concretas que son necesarias para crear y gestionar un sistema de información decisorio basado en almacenes de datos.
    - Sistema ETLs: **Spoon/Keetle**.
    - Servidor de Inteligencia Empresarial (Business Intelligence). **Pentaho BI Server**
    - Sistemas de consultas: **Pentaho Report Designer** y análisis OLAP con **jPivot** y **SAIKU** sobre .el servidor **Mondrian**

1.3. Enfoque y método

1.3.1. Arquitectura del proyecto

Este es un proyecto para los sistemas de información decisorios. Se entiende como sistema de información decisorios, a aquella parte del SI global de la organización dedicada a dar soporte a los diferentes tipos de procesos de toma de decisiones.

Se basará en software abierto de contrastada eficacia que cubrirá las 3 grandes capas: la importación de datos, su almacenamiento y posterior consulta. A un nivel de mas detalle tenemos:

- Almacén de datos, que está orientado a temas o conceptos , con información de largos periodos de tiempo, no volátil e integrado.
- Metadatos, que describen cual es la estructura de los datos que se almacenan y como se relacionan.
- Sistemas ETL "Extract, Transform, and Load" que lee la información de los datos primarios (fuentes de datos operativas), realiza procesos de transformación para ser almacenados (filtrado,adaptación,cambios de formato) y finalmente guardarlos.
- Sistema de consultas y explotación de usuario. Que permite de una forma ágil suministrar las consultas que precisa el usuario.

  El detalle de la pila de herramientas a utilizar se estudiará en la fase de diseño.

1.3.2. Infraestructura Informática.

La infraestructura informática se compone de 2 grandes áreas, por un lado están las herramientas para la documentación y presentación del proyecto y por otro las tecnologías necesarias para la implantación y explotación del producto a elaborar.

- Para el caso del Pla de Treball se ha utilizado un equipo en windows 97 con OpenOffice.org 3.3.0 para los documentos. Notepad++ v6.2.2 para notas y borradores. Ganttproject 2.0.10 para realizar los diagramas de Gantt y Argo UML para el como herramienta de diagramación del Análisis y Diseño (ver anexo sobre ArgoUML).
- Para caso del producto a elaborar, se realizará en la Maquina Virtual suministrada por la UOC bajo VirutalBox. Con ella se simulara el entorno de desarrollo y de usuario final.

1.3.3. Método de trabajo y herramientas.

Para presentar y desarrollar la fase de análisis se ha trabajado bajo el modelo ***Unified Modeling Language* (UML)** con la herramienta ***Argo UML*** [ref-argo],

Por que UML?

El utilizar el **UML** como sistema de diagramación y repositorio implica que había que tomar una decisión sobre como realizar el modelado: si se debía de utilizar el modelo ***Entidad/Relación (ER)*** y adaptarlo a la representación **UML** o utilizar una propuesta dentro del universo **UML**. Me he inclinado por lo segundo a partir de las siguientes lecturas:

[ref-prat-akoka] Que indica como transponer la visión de los 3 modelos utilizados en las bases de datos (conceptual, lógico y físico) a los almacenes de datos. Los autores argumentan que mucha de la literatura no aparece clara dicha distinción y ellos creen que es útil y necesaria que exista.

[ref-LujanS] Tesis de Sergio Luján que presenta y detalla una extensión del ***Unified Modeling Language* (UML)**, para desarrollar según el método ***Unified Process* (UP)** y aprovechando el uso de estándares en el desarrollo de los almacenes de datos. Luján S. expone un sistema de diagramación que cubre todas las partes del desarrollo de un almacén de datos.

[ref-LujanS] defiende la decisión de utilizar el UML por cinco consideraciones básicas:

- "***UML*** *sigue el paradigma* ***OO****, y que se ha demostrado tener una semántica mas rica y que otros paradigmas porque* ***OO*** *modelos (pueden) estar mas cerca a la concepción del usuario*"
- *"EL* ***UML*** *es un lenguaje conocido por los ingenieros informáticos"*
- *"El uso de los perfiles facilita el diseño, y así los desarrolladores de* ***Data Warehouse DW*** *pueden utilizar los conceptos que ahora ya aplican. Esto presenta la ventaja que se puede utilizar el* ***UML*** *para* ***DW*** *sin ser un experto en* ***UML****"*
- *"El* ***UML*** *es el estándar de* ***Object Management Group (OMG)*** *y unifica muchos años de esfuerzo en el análisis y diseño* ***OO"***
- ***"UML*** *se ha convertido en el modelo dominante en el mundo* ***OO"***

Como se va presentar el UML en la memoria?

El UML es un lenguaje **visual** utilizado como método sistemático para definir los aspectos de del análisis y diseño de un producto, pero para la memoria se precisa más bien un **relato** de los realizado y los gráficos son de soporte y comprensión del texto.. Para solucionar esta dicotomía entre los dos formatos de exposición, se ha decidido, dejar toda la documentación generada con el UML como anexos a la memoria. Así tenemos que en el anexo (Anexo Modelo Conceptual basado en UML) hay una exposición en detalle de los artefactos utilizados y los criterios de uso en el modelo UML.

1.4. Planificación

Para conseguir una correcta implementación del almacén de datos corporativo para ONdO, será construido por etapas, en cada uno de ellas se irá solucionando una problemática concreta.

Cada final de etapa se verá validada por la entrega de la PAC con el material elaborado en dicha fase.

Así tenemos 4 grandes fases:

- Plan de trabajo y análisis preliminar de requerimientos.
- Análisis de requerimientos y diseño conceptual y técnico
- Implementación
- Presentación

1.5. Tareas e hitos

Dentro de las tareas e hitos he tenido en cuenta, la necesidad de formación en tres ámbitos: en los almacenes de datos, en las tecnologías que los soportan, en la elaboración del proyecto.

Respecto a la asignación de jornadas y debido a mi trabajo como freelance, y por lo tanto dependiente de los plazos de entrega de mis trabajos con los clientes, puede implicar ciertos desajustes en la planificación, en todo caso, intentaré dar prioridad al TFC. La planificación está pensada en bloques semanales con jornadas de 4 horas.

La transcripción del diagrama de Gantt al Plan de Trabajo lo he realizado por secciones para que se pueda apreciar correctamente los días y jornadas asignadas a cada tarea.

Leyendas:

- Los hitos aparecen como rombos. En rojo los hitos de entrega.
- Los plazos de color ▮ indican tareas documentación.

1.5.1. Fase 1 Plan de trabajo.

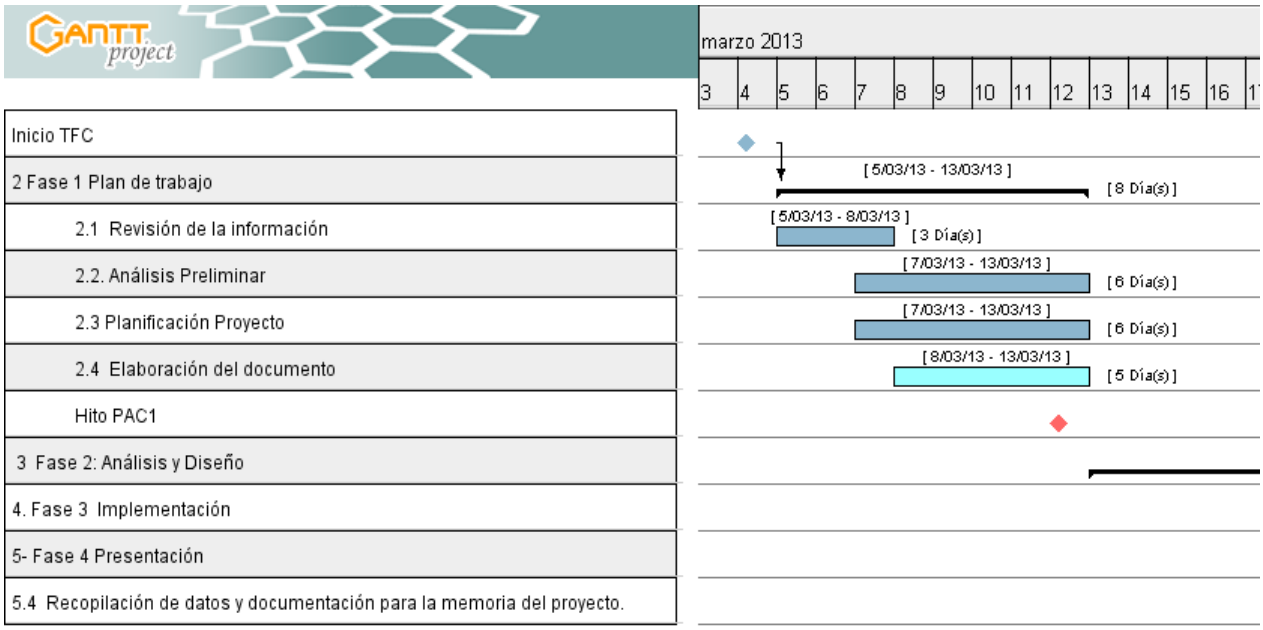


Ilustración 1: Gantt: Plan de trabajo

Como era de esperar esta fase se ha cumplido sin mayores desviaciones.

1.5.2. Fase 2 Análisis y diseño.

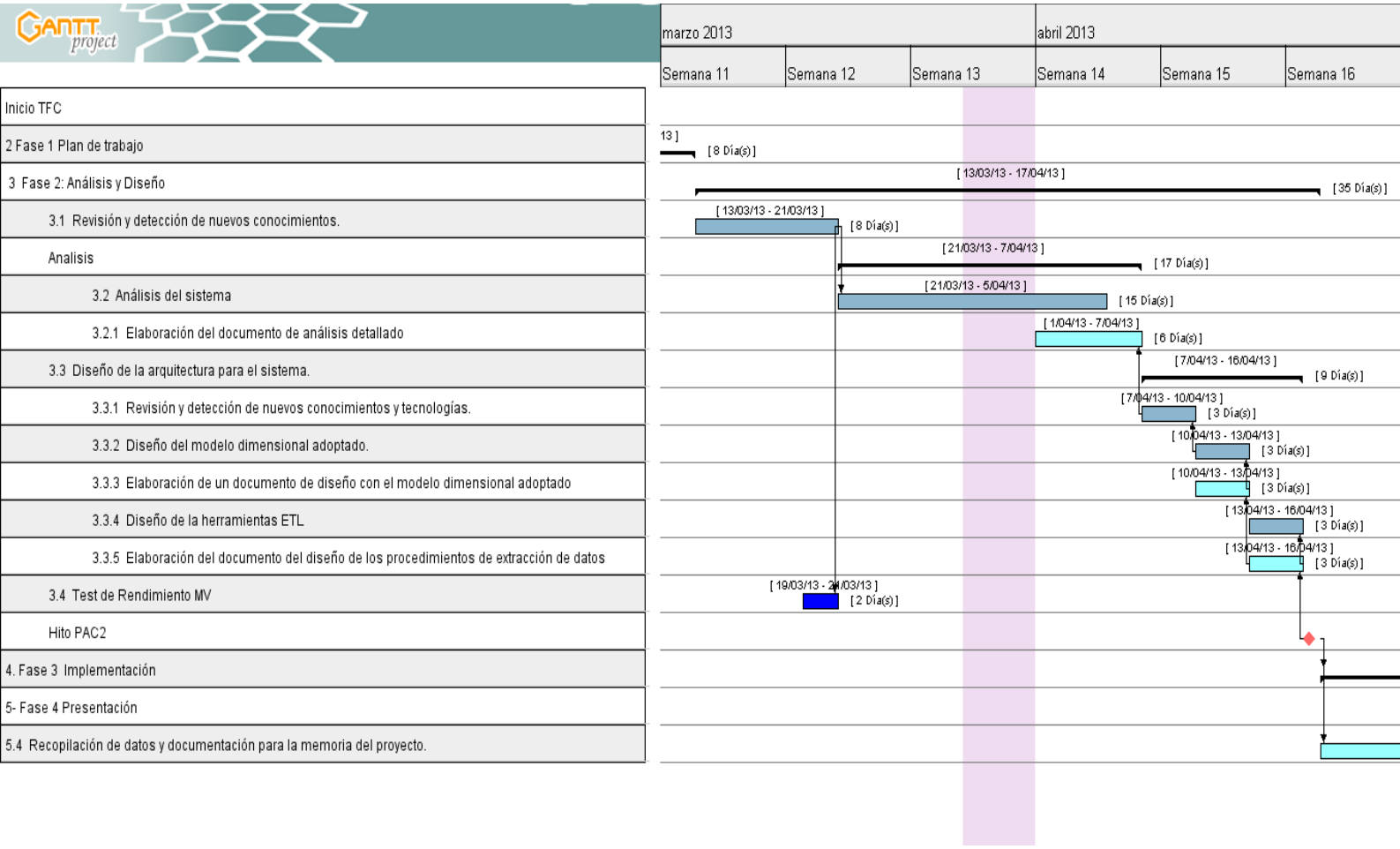


Ilustración 2: Gantt Fase de análisis y diseño

Al ser la fase con un desglose de tareas mayor, el proceso de cálculo de los plazos, se ha realizado empezando desde la fecha de entrega para ir asignando jornadas a las tareas en el orden inverso al que serán ejecutadas.

El plan se ha visto alterado ya que por razones de carga de trabajo, se tuvo que concentrar todo el proyecto en las tres últimas semanas, también se desvió el volumen de horas dedicado a formación ya que también implicó investigar en cual sería el mejor uso de la diagramación UML aplicado en toda la fase, por suerte el tiempo previsto para el análisis fue menor, y en cierta medida compenso. Como el análisis y diseño del proyecto se trabajo previamente con diagramas UML, incluyendo el diseño ETL, la redacción del documento quedó relegada al final de la fase.

1.5.3. Fase 3 Implementación

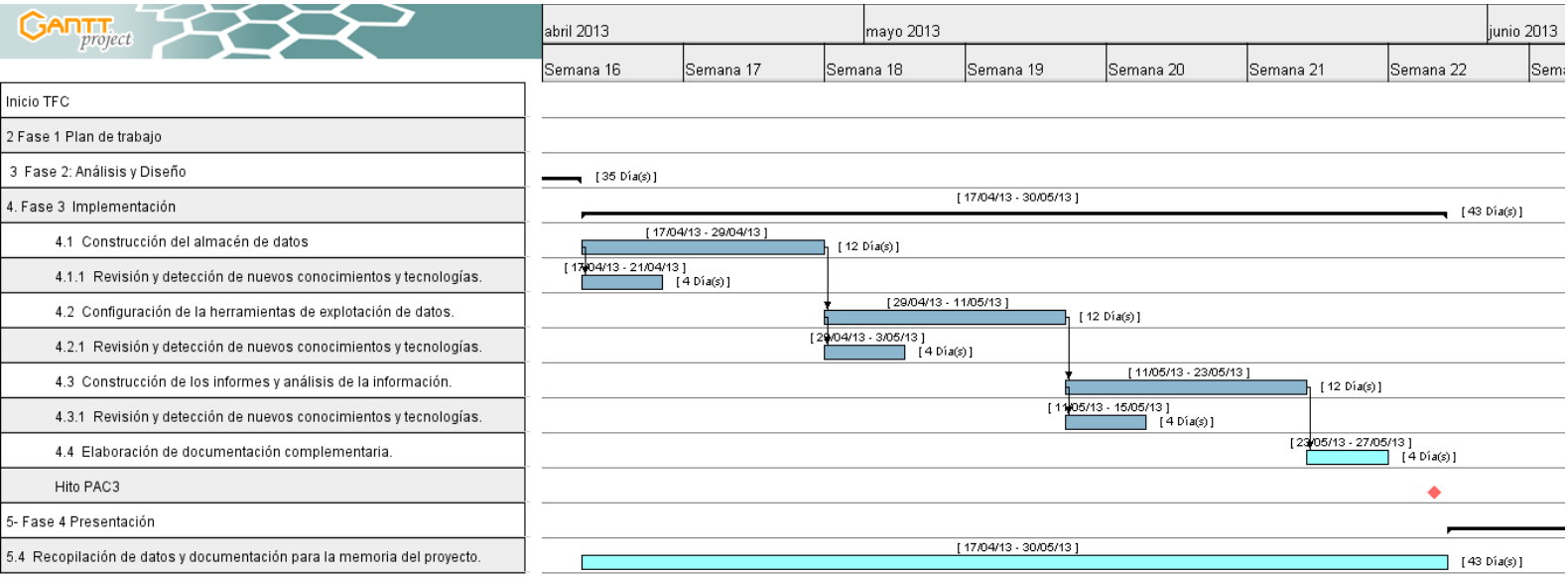


Ilustración 3: Gantt Fase Implementación

En esta fase, es donde ha habido, la mayor desproporción de horas y plazos, por otra parte normal ya que no se conocían todas las tecnologías que se utilizan, por lo que se realizaron varias aproximaciones, antes de encontrar el uso correcto de las mismas.

1.5.4. Fase 4. Presentación

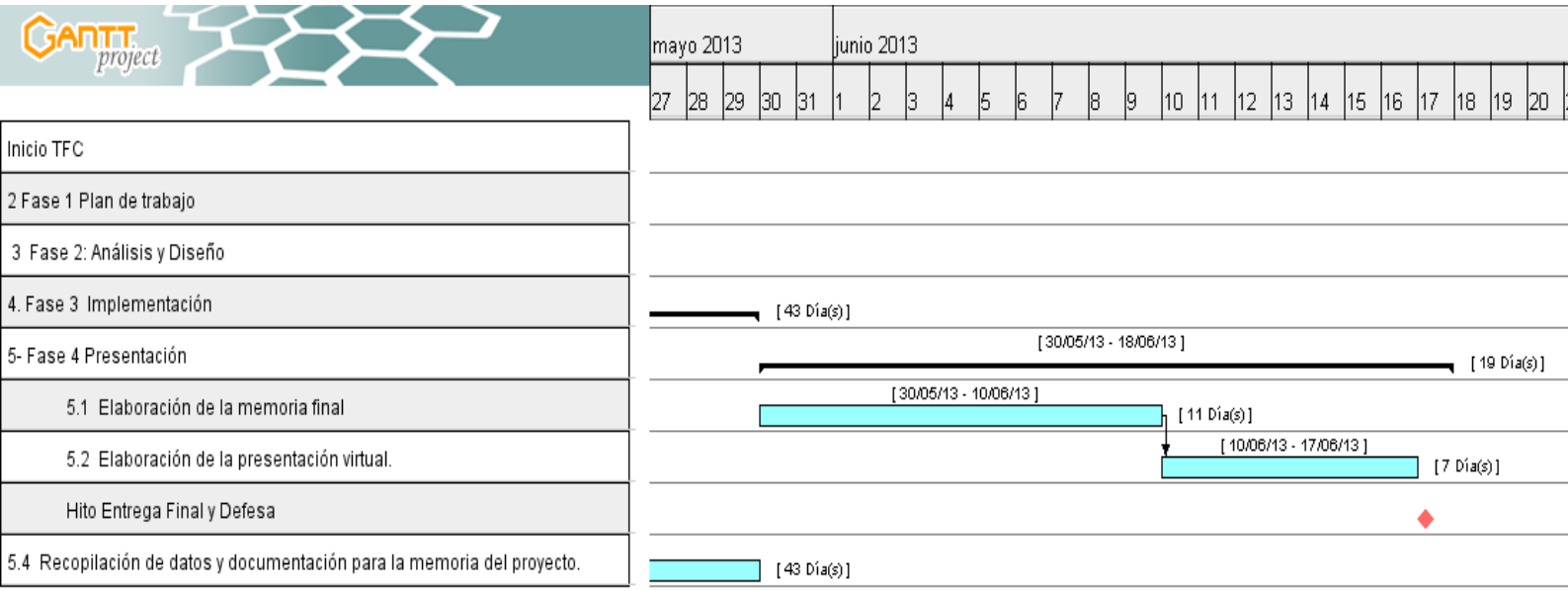


Ilustración 4: Gantt Fase presentación

1.6. Riesgos e incidencias

En la siguiente tabla se relacionan las incidencias mas destacadas que se pueden dar durante el desarrollo del Proyecto de Final de Carrera, con sus planes de contingencia.

Tabla 1: Plan de contingencias

• Incidencia	• Plan de Contingencia.
• El equipo no aguanta la maquina virtual	• Posibilidad de ampliar memoria. • Posibilidad de crear disco virtual exclusivo para memoria intermedia.
• Fallos en la maquina virtual	• Recuperar de las copias a disco duro externo de la Maquina virtual
• Perdida de ficheros de documentación del proyecto fuera de la maquina virtual	• Recuperar de las copias a disco duro externo.
• El desarrollo lleve a un camino sin salida, y es necesario deshacer lo hecho.	• Recuperar de versiones anteriores.
• Problemas de salud	• Si el tema es grave, no hay otra salida que parar el proyecto y dejarlo para el próximo semestre. • Solo hay margen de recuperación si después se amplia la disponibilidad horaria.
• Reducción de la disponibilidad horaria.	• Si la reducción es aguda no hay otra salida que parar el proyecto y dejarlo para el próximo semestre. • Solo hay margen de recuperación si después se amplia la disponibilidad horaria. • En todo caso el TFC tendrá prioridad sobre posibles trabajos que representen una disminución de la disponibilidad horaria.

1.7. Productos Obtenidos

- El fichero **UML** expuesto de forma expositiva en los anexos.
- El fichero **Spoon** para llevar a cabo el ETL que está en la máquina virtual
- Los informes
  - EstablimentsComarca
  - Establiments Comarca Homes Dones
  - Equipaments Comarca

- Las consultas OLAP
  - Ratios entre Establecimientos y Comarcas
  - Ratios entre Establecimientos habitantes por género y por grupo de equipamientos relacionados directamente con el turismo.
  - Ratios entre Equipamientos y Comarcas.
  - Consultas preparadas para generar nuevas consultas sobre los datos.
- Una maquina virtual en Windows que contiene tanto las herramientas cliente, como servidor
- Base de datos MySQL para el ***Stagging Area*** los procesos de transformación. Alojado en la máquina virtual
- Base de datos MySQL para el Proyecto Alojado en la máquina virtual

1.8. Descripción de los próximos capítulos.

En los próximos capítulos se desarrolla el análisis del proyecto (el modelo conceptual) presentando los grandes grupos de datos que de forman transversal se repiten en todas las fase de desarrollo. Se sigue con el diseño (el modelo lógico), hay un apartado dedicado al proceso ETL desde un punto de vista lógico (es decir enfocado a analizar la estructuras de las fuentes de datos y sus transformaciones para ser cargados en el almacén de datos), a continuación se relaciona de tablas creadas (el modelo físico), se planeta como se hizo la instalación y finalmente se presentan la lista de aplicaciones y informes generados.

2. Capítulo. Análisis

El proceso de análisis se ha centrado en los requerimientos actuales, pero a su vez se ha intentado dejar abierto el sistema a futuras necesidades, sin que estas no significasen una complicación mayor en el sistema[1].

2.1. Definición del contexto

Se requiere desarrollar un sistema de almacén de datos, para un estudio comparativo y evolutivo de los diversos tipos de establecimientos de alojamiento turístico, comparándolo con la zona geográfica, su dimensión, su población y con la oferta de equipamientos de la zona. Esto permitirá tener indicadores sobre el impacto de la industria de alojamiento turístico en una zona determinada.

En un principio, con la lectura de los requerimientos, puede entenderse que va ser un producto a ser utilizado de manera puntual en un proyecto de consultoría más amplio.

En todo caso, y analizando la información de las fuentes de datos, creo que si en vez de realizar un análisis puntual, el verdadero valor de las correlaciones que se estudian en este proyecto, estaría en un seguimiento durante los próximos años , y por lo tanto ir incorporando tanto información histórica, como nueva en el almacén de datos.

Y es sobre la anterior visión como se ha analizado y diseñado tanto el almacén de datos, como

1Dicha restricción parte de la idea de ver el proyecto, no solo como trabajo de final de carrera, sino también como el proyecto para unos usuarios reales, por lo tanto han de considerar el coste que supone tener un sistema mas o menos flexible a modificaciones futuras.

los procesos **ETL**

2.1.1. Consultas requeridas.

El número de informes que se requieren, son como mínimo los definidos en el enunciado, un total de 10, lo cuales los he agrupado en dos grandes grupos en función de las características de navegación

Hay un primer grupo que comparan establecimientos con zona geográficas y su población (habitantes y dimensión) , en todos ellos ha de poderse navegar por año y por tipo de establecimiento y agrupación geográfica.

”*Total d’establiments*”.

”*Total de places*”.

”*% de places respecte població*”.

”*Oferta mitjana de places*”.

”I*ndicador d’establiments vs habitants per gènere*”.

”*Indicador de places vs persones*”.

”*Quantitat de places ofertes / superfície del territori*”.

Tenemos un segundo grupo que compara las relaciones con los equipamientos, en esos casos ha de poderse, además, navegar por el árbol de tipos de equipamientos.

”*Nombre d’establiments/Nombre d’equipaments*”

”*% de població per equipament*”

”*Indicador d’equipaments vs població*”.

En todo caso, el sistema debería de permitir cualquier cruce geográfico y temporal entre la dimensión de establecimientos hoteleros y sus valores número y plazas, la dimensión de población y sus valores de número de habitantes y el área y la dimensión de equipamientos con sus valores de cantidad y tipos.

Lo anterior implica que hay que añadir la dimensión tiempo en equipamientos, lo que permitirá realizar importaciones anuales para que reflejen los cambios se produzcan en los equipamientos.

2.1.2. Fuentes de datos Disponibles

Las fuentes de datos que disponemos no son datos operacionales del sistema sino informes elaborados desde distintos origines, principalmente públicos.

Las fuentes proveen valores durante series de años, y se ha observado que en algunas series los nombres identificativos varían, así como también varían las categorías que incluyen.

Como criterio de diseño se ha establecido que siempre que sea posible utilizar la definición de nombres y categorización de los valores del años 2012.

Existen 3 diferentes fuentes principales:

- *poblacio.csv* con información sobre el número de habitantes entre los años 2012 y 2006 y su extensión en km2, además con el detalle en el último año 2012 del número de hombres y mujeres. Cada población tiene su código INE.

- *equipaments.csv* con información a fecha de 31/12/2012 sobre el n*ombre, dirección, municipio/población, comarca, código postal, teléfono, longitud,latitud, categorías (multi-nivel, por ejemplo "Equipaments | Cultura | Arxius") y localización.*
- *Establecimients año .txt*,. Los años son del 2006 al 2012. Contiene información proveniente de dos fuentes: *Federació Catalana d'Allotjaments Turístics* y *Idescat,* a partir de los datos del Departament d'Empresa i Ocupació
  - Fuente: ***Federació Catalana d'Allotjaments Turístics***: número de establecimientos y sus plazas, en formato desglosado y total por
    - Distribución geográfica: *Catalunya, àmbits, províncies, comarques*
    - Tipos: *Hotels, Càmpings, Turisme Rural*
    - Hoteles desglosado por *Hotels, pensions o estrellas oro, estrella argent,*
    - Càmpings desglosado por categorías: *luxe, 1ª,2ª,3ª*
    - Turismo Rural desglosando por: *Casa de poble compartida, independent, Masia, Masoveria*. Existen variaciones en las tipificaciones entre años.
  - Fuente: **Idescat**, a partir del **Departament d'Empresa i Ocupació**. solo año 2012 que da información sobre plazas por
    - Distribución geográfica: *Catalunya, àmbits, províncies, comarques*
    - Tipo de Turismo Rural: *Casa de poble compartida, independent, Masia, Masoveria.*

2.2. Criterios de un Almacén de Datos Decisional

Para poder dar solución a los requisitos del proyecto es necesario una herramiento que permita hacer consulta en linea y de una forma dinámica de la información (**OLAP**) lo que implica que el sistema de consulta y de almacenamiento ha de cumplir una serie de criterios

- Toda la información está centralizada y accesible bajo el sistema **OLAP**
- La información se puede actualizar al instante, el sistema ha de ofrecer la posibilidad de tener automatizado el proceso de extracción , transformación de los datos (**ETL**).
- El acceso a los datos ha de ser lo máximo de rápido. La estructura de la información se realiza en forma de estrella, se desnormalizan los datos, se diseñan sistemas cache para guardar información agregada.
- El sistema de almacenamiento ha de poder soportar altos volúmenes de datos.
- El acceso a los datos por parte del usuario ha de se fácil de entender y manipular

  Existe otra tipo de requisitos que son dependientes del diseño del problema

- Todas los datos históricos han de estar disponibles, es decir que se pueda seguir la cronología de un dato con sus sucesivas transformaciones.
- La definición cada dato es única, no existen varios nombres para referirse a ella.
- Los datos estan estandarizados, solo existe una definición y un conjunto de valores por cada dato.

Respecto al almacenamiento se va a utilizar una herramienta **OLAP** sobre base de datos relacional (**ROLAP**). existen otros modelos como en base de datos especializadas para contener almacenes de datos que son conocidas con el nombre (**MOLAP**) e incluso existen casos híbridos que combinan ambos mundos (**HOLAP**)

2.3. Casos de Uso de la Explotación.

Se han identificado los siguientes tipos de usuarios genéricos: el que consulta y navega con las herramientas de consultoría, el constructor de informes adicionales y finalmente quien administra el sistema y realiza la importación de datos

- Usuario: Utiliza el portal para consultar los informes, abre los listados predefinidos y aplica filtros. Sería el caso de quién accede a los informes definidos en los requerimientos, para después navegar por las diversas dimensiones. Además de los informes podrá manipular los cubos definidos en el sistema.
- Constructor: Puede crear informes adicionales y nuevas vistas tanto con las herramientas simples de usuario que dispone el sistema **Web Based Ad Hoc Query and Reporting**. como con herramientas mas sofisticados utilizando el **Report Designer** y crear nuevos modelos OLAP **Modrian** (**Pentaho**).
- Administrador: Responsable del servidor, y de validar el proceso de ETL, con lo que no solo tiene que tener un perfil técnico, si no, conocer las fuentes de datos, su estructura y estar preparado para detectar fuentes erróneas.

2.3.1. Casos de uso Usuario

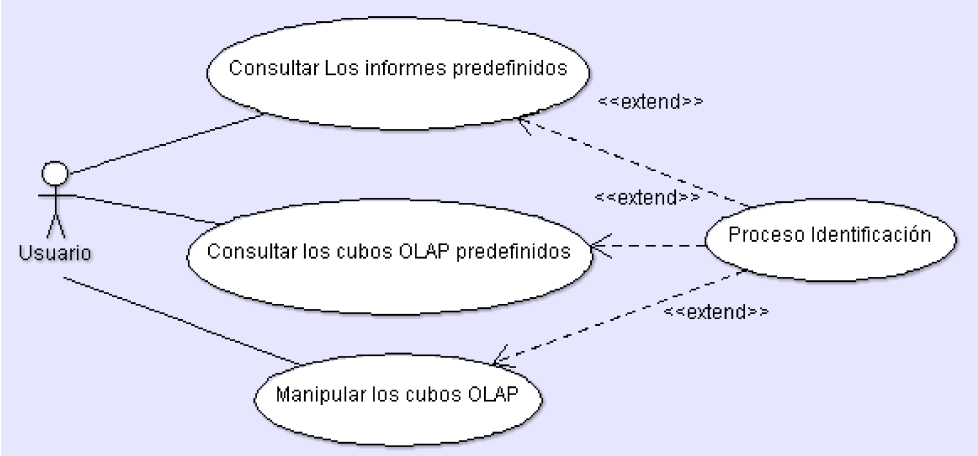


Ilustración 5: Caso de uso del Usuario

(Usuario) Consultar los informes predefinidos.

El usuario tras identificarse elegirá uno de los informes predefinidos, entrando los filtros que el informe haya definido y el sistema devolverá el informe en el formato de documento que el usuario haya elegido entre una lista.

(Usuario) Consultar los cubos OLAP predefinidos.

El usuario tras identificarse elegirá uno de los cubos que el sistema le ofrecerá a escoger, tras la

selecciona, el sistema le mostrar el cubo, pudiendo el usuario navegar por las diferentes jerarquías y miembros.

(Usuario) Manipular los cubos.

Se ha incluido ese caso por la existencia de una herramienta interactiva y "usuable", permite al usuario avanzado manipular unas fuentes de datos. El usuario seleccionara uno de los cubos disponibles y generara la consulta.

El usuario podrá guardar la consulta para reutilizar en posterior ocasiones.

Restricción esté caso solo estará permitido a los estrellas que estén definidos en el sistema, actualmente son demográfico/territorial, establecimientos y equipamientos

2.3.2. Caso de uso Constructor

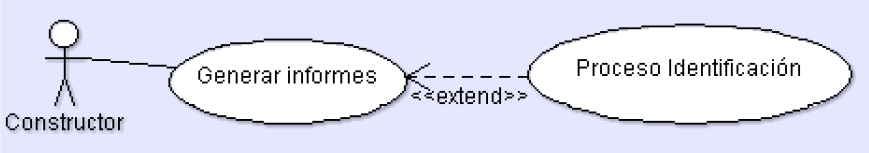


Ilustración 6: Caso de uso Constructor

(Constructor) Generar nuevos informes.

El constructor tras identificarse accederá al sistema para utilizar tanto un herramienta de generación de informes interactivos como una que permita documentos mas complejos..

2.3.3. Casos de uso Administrador

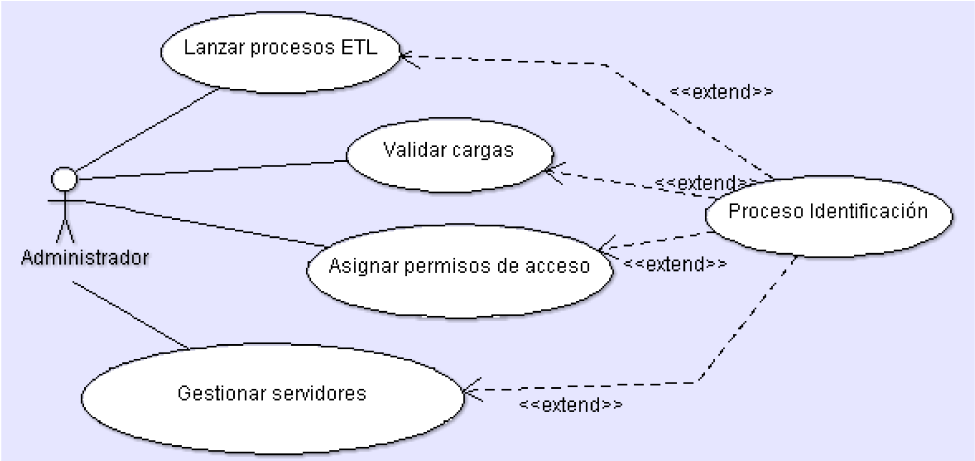


Ilustración 7: Caso de uso Administrador

(Administrador) Lanzar y controlar proceso ETL

El administrador tras identificarse accede al gestor ETL para realizar un proceso de transformación y carga, el sistema ha de mostrar por defecto la lista de tareas, de la cual existirá una que englobara a todas las otras. El administrado modificara los parámetros necesarios y lanzara

cada una de las tareas, el sistema ha de avisar de la entrada de incoherencia y del número de registros actualizados. En caso de incoherencia el sistema ha de parar el proceso.

(Administrador) Validar

El Administrator tras el proceso de ETL accede a la herramienta de análisis OLAP y de una manera interactiva podrá configurar un esquema OLAP para poder validar los resultados con los datos de entrada, mediante catas aleatorias de datos.

(Administrador) Asignar permisos de acceso a servidor BI

El administrador tras identificarse tendrá la opción de asignar a los documentos e informes creados los permisos de acceso para los usuarios finales.

(Administrador) Gestionar los servidores.

El administrador tras identificarse en una consola gestionara las programación de tareas, las configuraciones de los servidores y conexiones a las base de datos, el mantenimiento de los servicios.

2.4. Modelo Conceptual Multidimensional

2.4.1. Introducción al Modelo Conceptual

En síntesis y tal como se expresa [ref-LujanS] podemos definir la necesidad del modelo conceptual

- Entender el ámbito o dominio real en el que esta insertado el problema
- Dar razón sobre dicho ámbito
- Conseguir sincronizar la visión del problema de todos los implicados en el proyecto.

2.4.2. Criterios generales en el modelo conceptual

Guardar toda la información correcta que provea las fuentes de datos

Tener la información al mas bajo nivel de detalle que suministran las fuentes de datos

Nada se borra y se refleja su historia de cambios.

2.4.3. Nivel 1: modelo conceptual

En el apartado de “Definición del contexto” se han identificado dos grandes grupos de consultas que ha de responder el sistema:

- Comparativas de establecimientos con zona geográficas y su población (habitantes y km^2) , en todos ellos ha de poderse navegar por año y por tipo de establecimiento y agrupación geográfica.
- El sistema a de soportar comparativas y ser capaz de establecer relaciones entre los establecimientos y los equipamientos, Además ha de poderse , navegar por el árbol de tipos

de equipamientos y clases de equipamientos.

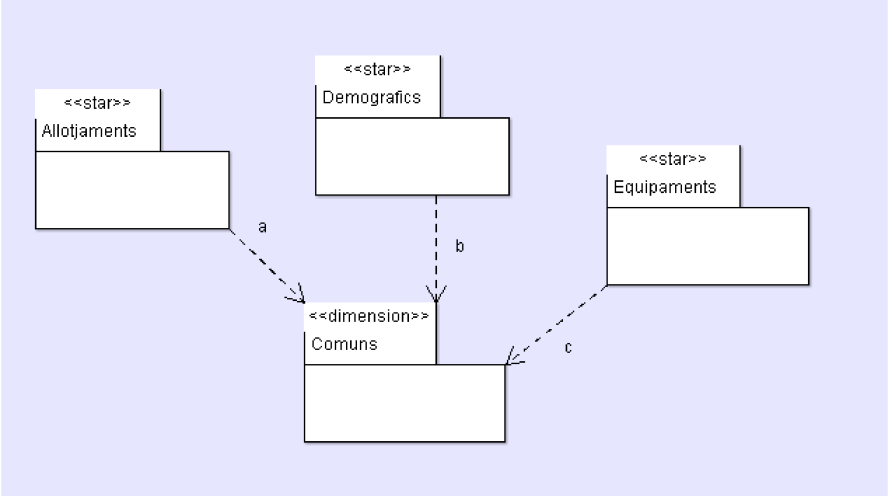


Ilustración 8: Modelo Conceptual

Se ha divido el analisis en 4 grandes grupos o **paquetes UML** con lo que tenemos 3 formaciones de estrellas (paquetes *<<start>>)*: *Allotjaments, Demografics, Equipaments.* Y una dimensión compartida por las tres estrellas llamada *Comuns*. Así tenemos en un primer nivel una visión de los diferentes tipos de valores que conforma el producto.

- El paquete *Allotjaments* contiene la estrella con los valores sobre el alojamiento hotelero y su dimensión de categorías de establecimientos y las dimensiones comunes: *Any* y *Comarca*
- El paquete *Demografics* contiene la estrella con los valores sobre la demografía y extensión geográfica, además de las dimensiones compartidas *Any* y *Municipi*
- El paquete *Equipaments* contiene la estrella con los valores agregados de equipamientos y su dimensiones de categorías de equipamientos (*Equipament)* e instalaciones *(Instalacions),* además de las dimensiones compartidas *Any y Municipi*

2.4.4. Nivel 2: Paquete dimensión *Comuns*

Dimensión temporal *Any:*

En este caso tenemos, a la dimensión *Any* que solo tiene el nivel base de *Any***.** Este es el -único caso en donde la clave no será asignado automáticamente por la base de datos, sino será el año de cuatro cifras. A su vez será utilizado como descriptor en el sistema OLAP. Aparte del año, se ha previsto un elementos más con la intención de permitir filtrar por antigüedad, es decir por año

Dimensión territorial *Municipi:*

Tiene como base de más bajo nivel al municipio (*Municipi*) que después asciende por *Comarca* que a su vez se puede ascender por *Ambit* y *Provincia*.

Dimensión territorial *Comarca*

Tiene como base de más bajo nivel a la *Comarca* desde la que se puede ascender por *Ambit* o *Provincia.*

Comarcas y Municipios: La razón de que existan dos puntos desde donde poderse “enganchar” al árbol de dimensiones territorial, es debido a que existen, *hechos* que tienen su nivel más bajo de

representación en el *Municipi* y otros en *Comarca.*

En el anexo (Anexo Modelo Conceptual basado en UML - Nivel 2: Paquete dimensión Comuns)del detalle en UML existe un modelado detallado del paquete

2.4.5. Nivel 2: Paquete estrella *Allotjaments*

La estrella *Allotjaments* tiene como el *hecho* de que exista un establecimiento y su número de plazas. No se puede acumular dichos valores por varios años ya que no tiene sentido, en todo caso establecer valores medios.

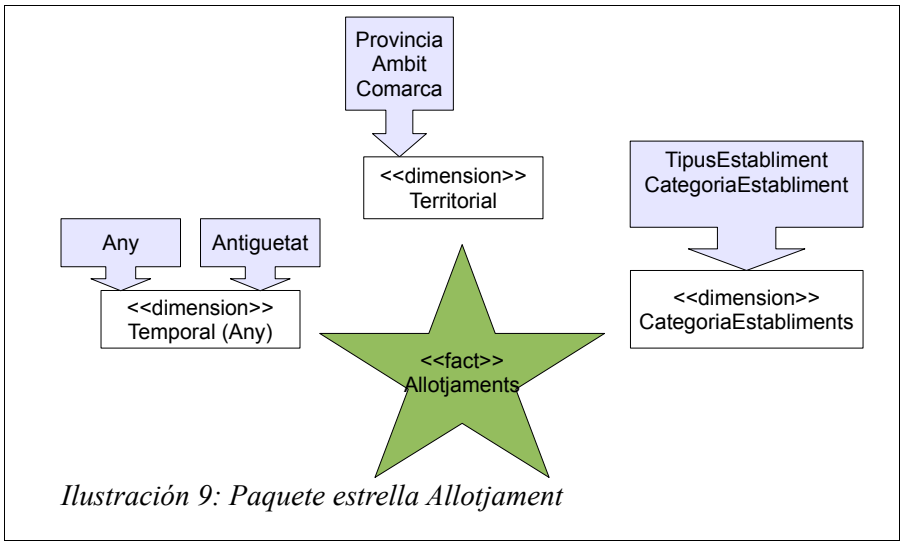


Ilustración 9: Paquete estrella Allotjament

Las dimensiones tienen como jerarquía de base, el año (A*ny)* ,comarca (*Comarca, Ambit, Provincia)* y categoría de establecimiento (*CategoriaEstabliment: TipusEstabliment, CategoriaEstabliment*)

La dimensión *CategoriesEstabliments* tiene la siguiente estructura.

TipusEstabliment	*CategoriaEstabliment*
Hotels	*Hotels (estrellas or)*
	Hostals (estrellas argent)
Campings	*Luxe*
	1
	2
	3
Turisme Rural	*Casa de poble compartida.*
	Casa de poble independent
	Masia
	Masoveria

Tabla 2: Tabla de tipos y categorías de Establecimiento

En el anexo (Anexo Modelo Conceptual basado en UML - Nivel 2: Paquete estrella

Allotjaments) del detalle en UML existe el correspondiente modelado detallado del paquete

2.4.6. Nivel 2: Paquete estrella Demográfics

La estrella *Demográfics* tiene como el hecho a los datos demográficos de una zona geográfica (*habitants,homes,dones*) y, por otra parte, el dato geográfico de los Km2 cuadrados (*extensio*.) En el caso de *homes* o *dones* las fuentes de datos solo suministran información del año 2012, pero el sistema está preparado para insertarla en otros años.

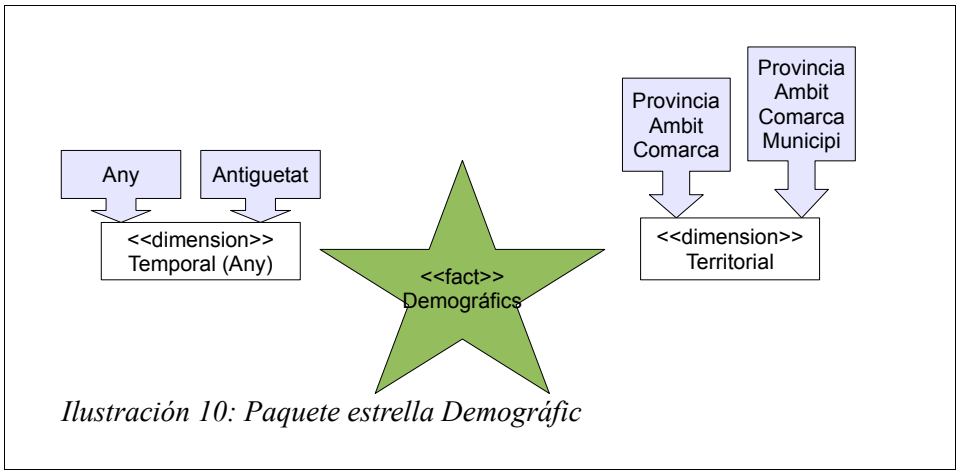


Ilustración 10: Paquete estrella Demográfic

Tenemos a las dimensiones temporal (*any*) y territorial sobre las que se puede analizar los hechos. En el caso del nivel territorial, puede ser tanto a nivel municipio (*municipi, comarca, ambit, provicia)* como a nivel comarca (*comarca, ambit, provicia)*. En los informes las correlaciones *demográfics* con *establiments* se realizará a nivel *comarca*

No se puede acumular los valores demográficos y geográficos por varios años ya que no tiene sentido, si que se puede calcular las medias, aunque en el caso de los Km^2 de un municipio no sea un dato que experimente cambio.

En el anexo (Anexo Modelo Conceptual basado en UML - Nivel 2: Paquete estrella Demográfics) del detalle en UML existe el correspondiente modelado detallado del paquete

2.4.7. Nivel 2: Paquete estrella *Equipaments*

La estrella *Equipaments* tiene como hecho la existencia de un determinado tipo de equipamiento y del tipo de instalaciones que posee. No se puede acumular los valores los equipamientos por varios años ya que no tiene sentido. Este es un “hecho” que tiene datos “degenerados de dimensión” (*nom, cp, telefon, longitud,latitud),* pero el valor es la cantidad de equipamientos que existen en una determinado tipo de equipamiento.

En las fuentes de datos la información esta reducida a un solo año, pero el sistema se a diseñado para que permita la entrada de otros años por eso se ha incluido la dimensión temporal (*Any*). De echo se cree, que enriquecería los análisis correlativos entre equipamientos y demográficos/establecimientos, si en un futuro, se inserta la lista de equipamientos de otros años.

Una observación a pesar de que incluya la dimensión temporal, los informes actuales no lo tiene en cuenta, o dicho de otro modo, cualquier correlación con demográficos o establecimientos en

donde intervenga la dimensión temporal, *Equipaments* tendrá los datos de 2012

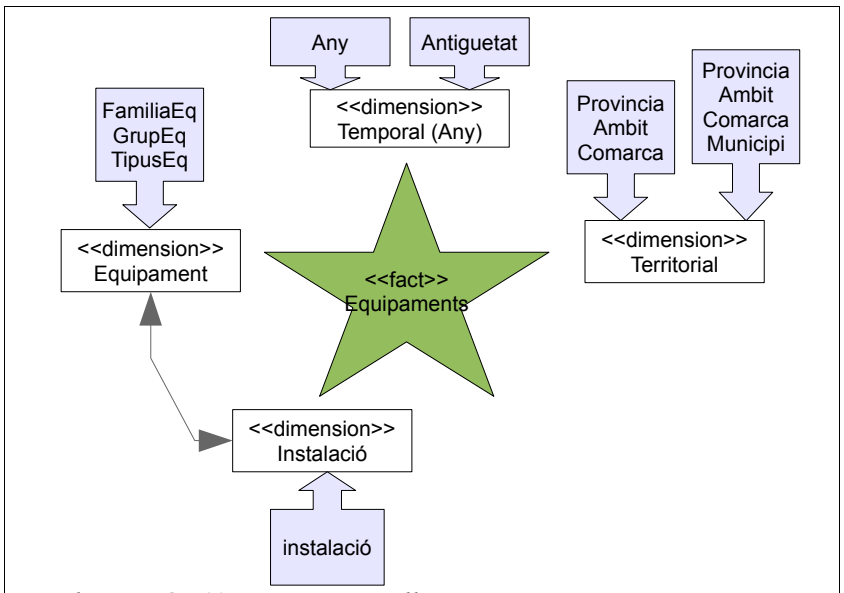


Ilustración 11: Paquete estrella Equipaments

Un equipamiento esta relacionado con la dimensión territorial, tanto a nivel municipio (*municipi, comarca, ambit, provicia)* como a nivel comarca (*comarca, ambit, provicia)*. En los informes, las correlaciones *equipaments* con *establiments* se realizará a nivel *comarca*. También es a nivel *comarca* en el caso de las correlaciones *equipaments* con d*emográfics,* aunque ambos posean el nivel municipio,ya que *equipaments* tiene muchos más municipios que *demográfics*.

El análisis de las fuentes de datos nos da que la estructura de clasificación de los equipamientos (*equipament <dimension>)* está formada por 3 niveles jerárquicos que los hemos identificado como *FamiliaEq*, *GrupEq* y *TipusEq*.

A mayores tras estos 3 niveles y en función de *TipusEq* puede el equipamiento contener unas instalaciones (*instalacions*) para realizar una serie de actividades. El concepto de instalaciones puede hacer referencia tanto a los espacios como a los servicios. Un ejemplo del segundo caso son los centros educativos que pueden dar unos o otros niveles de enseñanza.

La relación entre la dimensión de equipamientos (*equipament <dimension>: FamiliaEq*, *GrupEq* y *TipusEq)* e instalaciones es una relación N a M, es decir, una instalación puede estar en multiples *equipament <dimension>* y a su vez *equipament <dimension>* puede tener múltiples instalacions

En el anexo (Anexo Modelo Conceptual basado en UML - Nivel 2: Paquete estrella Equipaments) del detalle en UML existe el correspondiente modelado detallado del paquete

3. Capítulo. Diseño

3.1. Modelo lógico multidimensional.

El modelo lógico multidimensional se diseña siguiendo lo que sería el modelo lógico en el diseño de una base de datos relacional.

3.1.1. Criterios generales para el Modelo Lógico Multidimensional.

Como se ha comentado en el Modelo Conceptual el proceso de definir el modelo lógico multidimensional se ha realizado con la misma herramienta para el modelado conceptual **ArgoUML**, ya que como se explica previamente, unos de los objetivos del proyecto era tener un método y un modelado unificado e integrado.

En sección anexa (Anexo Modelo Lógico basado en UML) hay una explicación detallada del como se hizo el modelado, de como se explicita la relación entre los elementos del conceptual con los elementos del lógico, lo que posibilita la integridad entre ambos modelos. Además se creó un perfil de estereotipos específicos para poder representar tanto el modelo como su relación con el modelo conceptual.

3.1.2. Notas sobre la representación del modelo lógico

La representación del modelo lógico se ha extraído del repositorio UML,(ver anexo Anexo Modelo Lógico basado en UML) pero es una versión preparada para la exposición de la memoria. En ella utiliza muchos de los artefactos del perfil diseñado para poder representar el modelo de base de datos. Así tenemos:

- <<*tabla*>> a nivel de clase para indicar que es un tabla.
- <<*fk*>> a nivel atributo indica que es un clave foranea.
- <<*pk*>> a nivel atributo indica que es un clave primaria
- <<*unsigne*>> hace referencia a que es tipo numérico sin signo.

3.1.3. Nomenclatura de los elementos

Definimos en el modelo lógico los nombres finales que poseerán los diversas tablas y campos, por lo que se han establecido una serie de convenciones, siguiendo un versión libre del modelo propuesto en [ref-Bouman] .

- prefijo **dim** indica que la tabla es una dimensión.
- prefijo **fact** indica que la tabla es un hecho o valor
- prefijo **agg** indica que es tabla con valores agregados de otras tablas.
- perfijo **deg** indica que es un tabla resultado de una "*dimensión degenerada*"
- prefijo **clau** en un atributo indica que es una columna que forma parte de la clave.

3.1.4. Modelo lógico nivel 1: Visión general.

En el diagrama de la visión general del modelo lógico del almacén de datos, nos permite ver la derivación (relación) que existe con el modelo conceptual. Para buscar ser homogéneos se han aplicado los mismos criterios en la división en paquetes. Con ello, además, facilitamos el mostrar la relación entre los modelos. Así tenemos:

- Del modelo conceptual *Comuns* deriva el modelo lógico *Comuns*.
- Del modelo conceptual *Allotjaments* deriva el modelo lógico A*llotjaments*

- Del modelo conceptual *Demográfic* deriva el modelo lógico *Demográfic*
- Del modelo conceptual *Equipaments* deriva el modelo lógico *Equipaments*

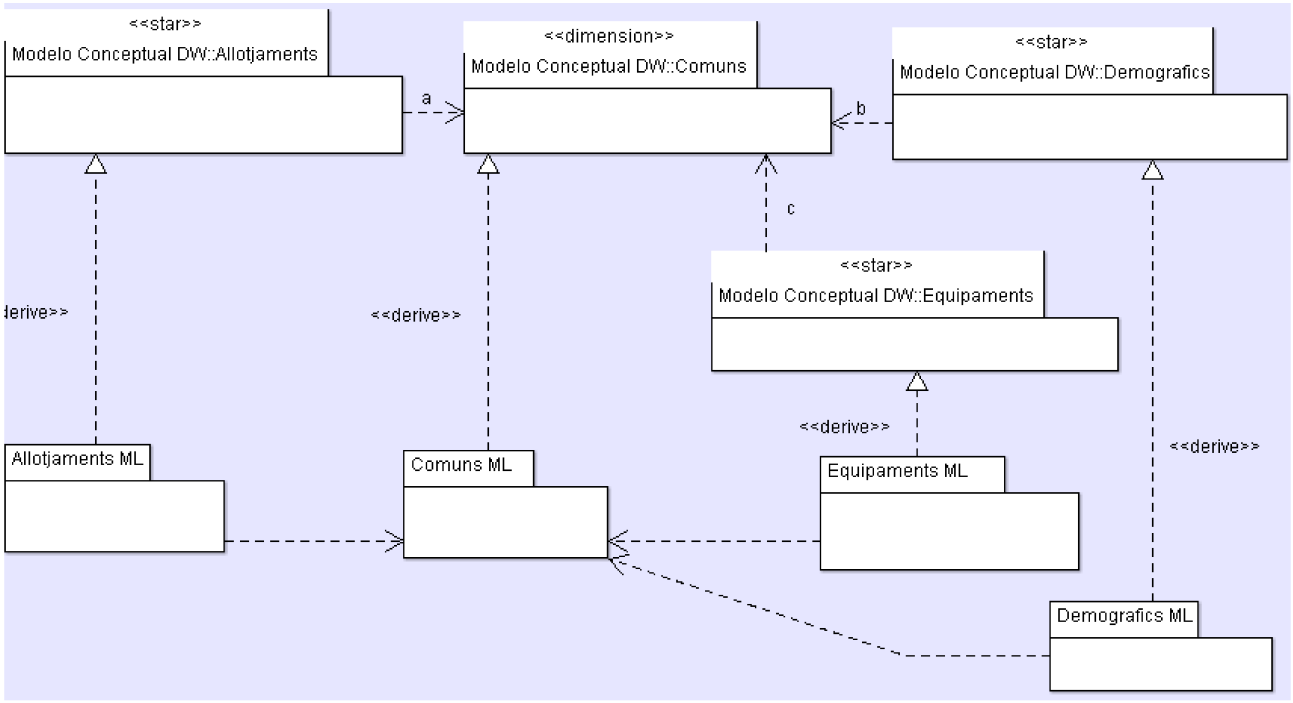


Ilustración 12: Nivel 1 Modelo lógico

3.1.5. Nivel 2: Paquete lógico *Comúns*.

En la diseño de la tabla del año (*dim_any)* se ha incluido el nuevo atributo *actual* que sirve para indicar cual es el año actual y cuales son los años anteriores, esto, por ejemplo, aporta rapidez en las selecciones del tipo comparativa entre este año y los 3 anteriores.

En el diseño de la tabla de comarca (*dim_comarca*) se ha decido desnormalizar y agrupar todos los elementos en un sola tabla y que en ella estén todas las combinaciones posibles. En esté caso al ser jerárquico solo hay una combinación por *comarca*, *ambit* y *provincia*.

Lo mismo pasa con la tabla *dim_muicipi* que incorpora *comarca*, *ambit* y *provincia.* También se ha decidido que una fila de *dim_municipi* solo puede existir si existe el correspondiente fila en *dim_comarca.*

Las claves de *dim_comaca* y *dim_muicipi* serán autogeneradas por el sistema gestor de bases de datos.

La representación de las tablas aparecen en las ilustraciones que acompañan a las descripciones de los demás modelos.

En el anexo (Anexo Modelo Lógico basado en UML - Nivel 2: Paquete Lógico *comúns*) del detalle en UML existe el correspondiente modelado detallado del paquete

3.1.6. Nivel 2: Paquete lógico *Allotjaments.*

En el diseño de la tabla de categorías (*dim_categoria_establiments*) se ha decido desnormalizar y agrupar todos los elementos en un sola tabla y que en ella estén todas las combinaciones posibles. En esté caso al ser jerárquico solo hay una combinación por *tipus y grup*.

En *fct_establiments* están los valores del número de plazas (*places*) y y el número de establecimientos (*establiments*)

Todas las claves primarias de las tablas sobre alojamientos serán generadas por el sistema gestor de base de datos.

La existencia de un valor en *fct_establiments* viene dada por la existencia en *dim_categoria_establiments, dim-any* y *dim_comarca* de referencia

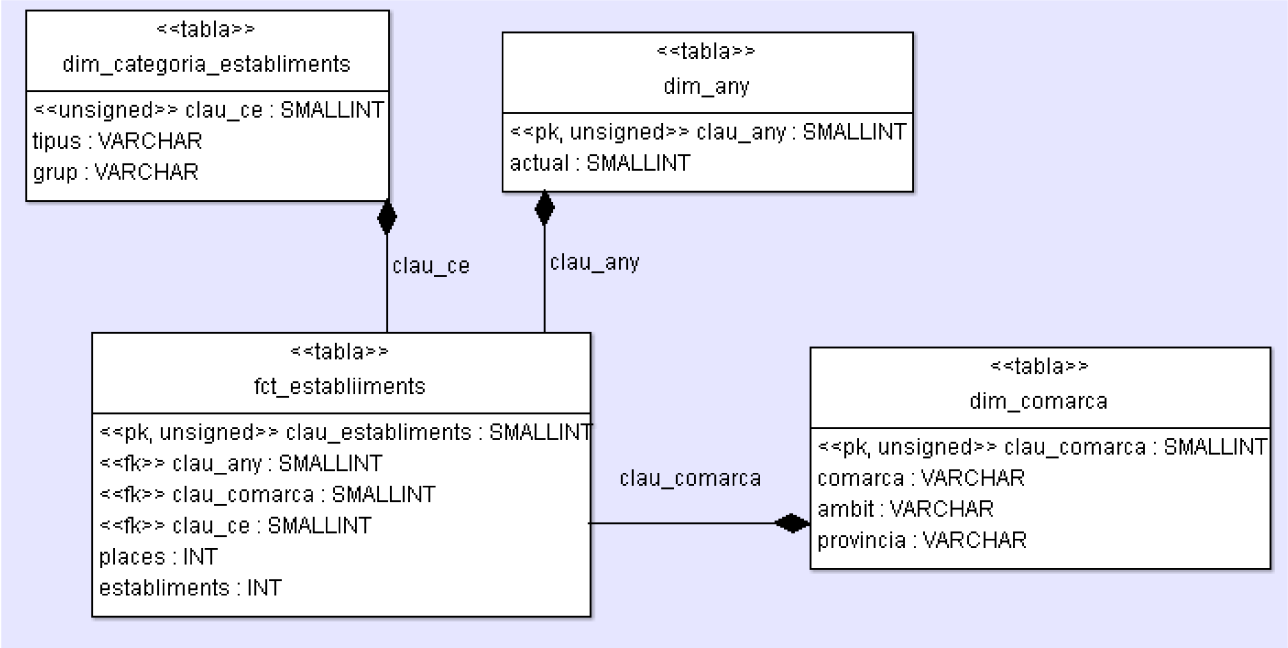


Ilustración 13: Paquete lógico Allotjaments

En el anexo (Anexo Modelo Lógico basado en UML - Nivel 2: Paquete Lógico Allotjamnets) del detalle en UML existe el correspondiente modelado detallado del paquete

3.1.7. Nivel 2: Paquete lógico *Demografic*.

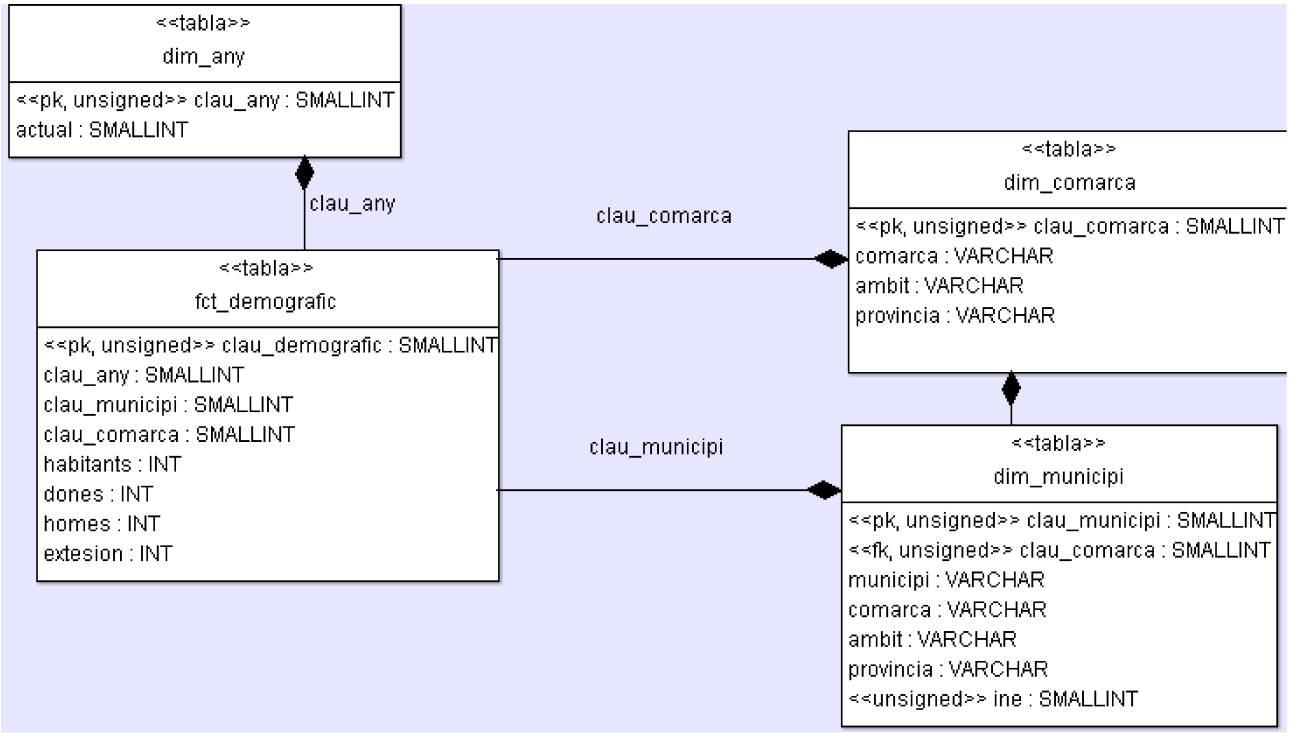


Ilustración 14: Paquete lógico Demográfic

La clave primarias *fct_demografic* será generada por el sistema gestor de base de datos.

En *fct_demografic* están los valores sobre el número de *habitants, dones y homes* y la *extensión.* Los atributos *Dones* y *homes* pueden tener valores nulos si no se han importado o calculado dichos datos. Ver "Suposiciones aplicadas" en el "Procesos de extracción, transformación y carga (ETL)"

La existencia de un valor en *fct_demografic* viene dada por la existencia en *din_any, dim_municipi, dim_comarca* de referencia

En el anexo (Anexo Modelo Lógico basado en UML - Nivel 2: Paquete Lógico Demografic*)* del detalle en UML existe el correspondiente modelado detallado del paquete

3.1.8. Nivel 2: *Paquete lógico Equipaments.*

Como lo que es de información relevante es el número de establecimientos de un tipo y sus instalaciones, se han definido tablas agregadas que contienen dicho valor ya totalizado al nivel más bajo de la jerarquía.

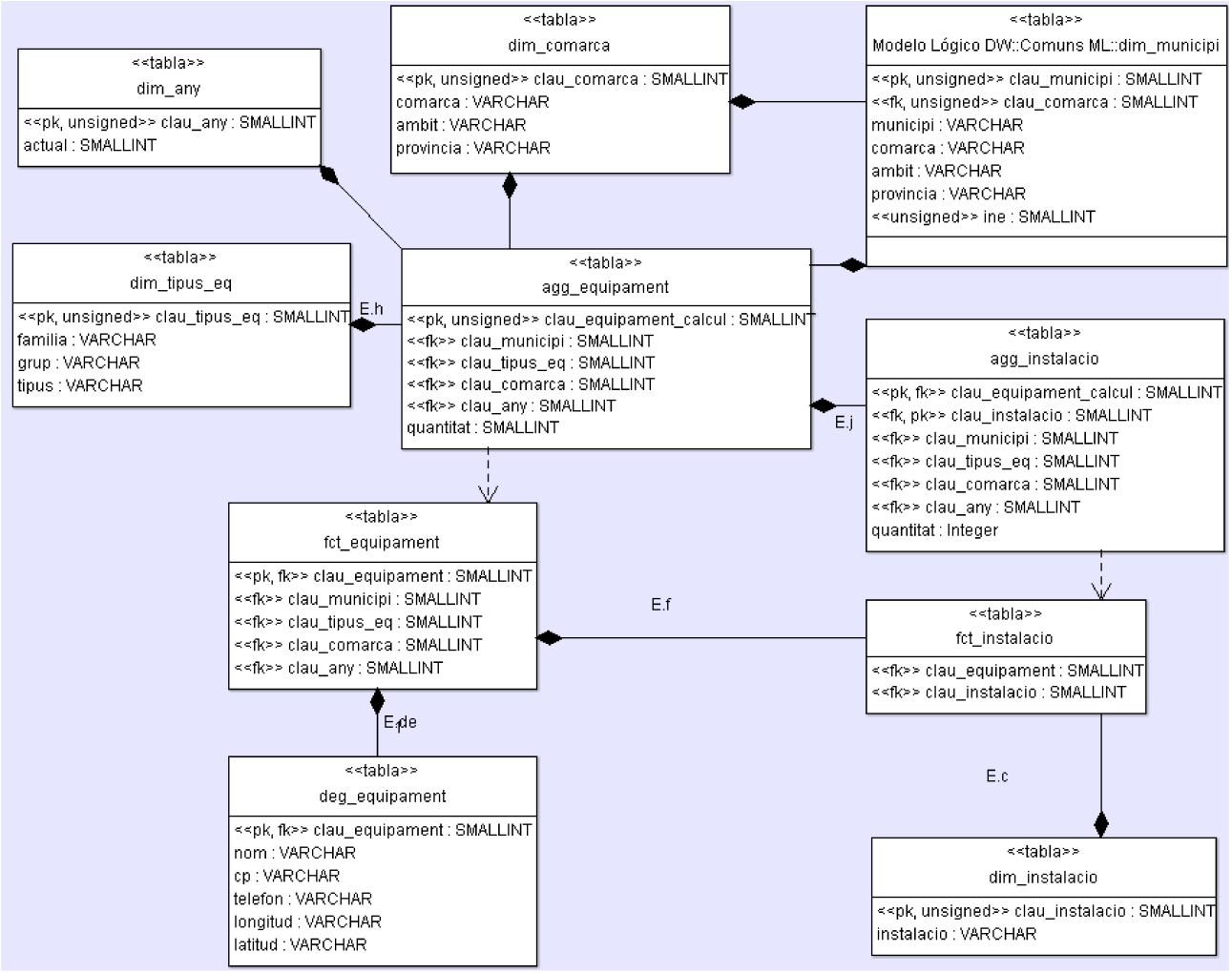


Ilustración 15: Paquete Lógico Equipaments

Así tenemos que *agg_equipament* es la suma de *fct_equipament* de un *dim_tipus_eq, dim_any, dim_comarca, dim_municipi.*

Se desnormalizado las categorías de equipamientos (*dim_tipus_eq*) colocando todas las combinaciones posibles de *familia, grup* y *tipus* que se dan en equipamientos.

El *fct_equipament* es solo una relación de claves foráneas y los datos que no son hechos (*nom,cp,telefon,longitud,latitud*) se han colocado en una tabla separada llamada *deg_equipament*

Después tenemos el *fct_instalacio* que contiene la relación N a M entre instalaciones (*dim_instalacio)* y equipamientos (*fct_equipament*), A su vez en vez, *fct_intalacio* está sumado en la tabla *agg_instalació* por cada combinación de *dim_any, dim_comarca_dim_municipi* y *dim_tipus_equipament*

En el anexo (Anexo Modelo Lógico basado en UML - Nivel 2: Paquete Lógico Demografic) del detalle en UML existe el correspondiente modelado detallado del paquete

3.2. Modelo lógico del proceso ETL

Para definir el proceso de extracción, transformación y carga de las fuentes al almacén de datos se ha creado un modelo lógico, en el que mediante una representación por niveles se han detallado

los diferentes elementos que intervienen en dicho proceso. Dicho modelo se ha basado en las propuestas de [ref-LujanS] más las recomendación para un buen proceso de **ETL** de [ref-Bouman]

Se ha decidido crear la definición del proceso **ETL** utilizando el modelado **UML,** con ello se consigue que en un único repositorio de información estén tanto la información sobre el almacén de datos como los procesos de carga y lo que es más importante como se relacionan entre ellos.

Otro aspecto no menos importante es que se definen y se tipifican todos los elementos implicados en un **ETL**, con lo cual, se consigue por un lado normalizar la nomenclatura utilizada y por otra facilitar el análisis y diseño ya que esté se centra en ver como se relacionan y se utilizan dichos componentes para que una fuente de datos concreta se traslade al almacén de datos definido.

El paradigma que persigue el modelo se puede resumir en los siguientes puntos (basados en [ref-LujanS] y [ref-Bouman])

- El proceso será descompuesto en unidades elementales llamados **mecanismos ETL**
- Los datos de cualquier fuente que no estén en tablas será previamente traspasado a una estructura de base de datos relacional.
- Existirá una zona en la que se almacenarán todas las tablas necesarias en el proceso de transformación. *Stagging Area*
- El proceso de transformación, será auditado y en el caso de errores en los datos o en su formato el flujo se detiene en el estado en que se ha producido el error. En nuestro caso cuando se detecte un error se mirará si es factible modificarlo definiendo un paso de conversión en el proceso de transformación.
- Para los fuentes de orígenes no relacionales se realizará un primer paso que será tener dichas fuente en un formato relacional y sin ningún tipo de manipulación. Es importante que los datos y su estructura no sean manipulados, eso se deja a los **mecanismos ETL** de transformación.
- Las tablas intermedias generadas durante el proceso de transformación se inicializarán en cada proceso de importación.

A continuación desglosamos las unidades elementales llamadas **mecanismos ETL** definidos por [ref-LujanS]:

- **Agregación (*Aggregation*)**,como su nombre indica se refiere cuando se totalizan o calculan datos basados en algún criterio
- **Conversión (*Conversion*)**: Cambia los tipos de datos y los formatea o genera nuevos datos a partir de datos existentes.
- **Filtro (*Filter)***: filtra y verifica los datos.
- **Incorrecto (*Incorrect*)**: Redirige (almacena) datos incorrectos.
- **Union (*Join*)** Se unen dos o mas fuentes en una nueva que sea la combinación de algunos de sus atributos.
- **Cargador (*loader*)** Realiza una carga de datos al almacén de datos.
- **Registrador (*log*)** Registra la actividad de los demás mecanismos.
- **Fusionar (*Merge*)** Integra dos o mas fuentes de datos con atributos compatibles

- **Sustituto (*Surrogate*)** Genera una clave primaria sustituta a la que sería la clave natural que define a la entidad. Por ejemplo agrupar los atributos en una clave natural en un única clave en formato numérico.
- **Envoltorio (*wrapper*)** Extrae los datos una fuente de datos externa para insertarlo en los registros de la base de datos del proceso de **ETL**

En el anexo (Anexo Modelo Lógico ETL basado en UML) existe el correspondiente ampliación de como se utilizó el UML.

3.2.1. Suposiciones generales aplicadas

- Como criterio de diseño se ha establecido que siempre que sea posible utilizar la definición de nombres y las clasificaciones que viene con las fuentes del año 2012. Existe algunos valores que su nombre varía con los años.
- El cálculo de las plazas de camping serán según la regla del 2011, esto implica que en años anteriores durante el proceso de transformación habrá que recalcular las plazas de campings
- El ajustes de los casas rurales en el 2006 ya que no existía la categoría de "*t.rural independen*t"
- La inclusión de todos los equipamientos independientemente de que los municipios no tuvieran su correlación con la fuente población.
- El cálculo de *homes* y *dones* en los años anteriores del 2012 a partir de la proporción del 2012

3.2.2. Paquete *ETL Comuns*.

La carga de la tabla *dim_Any* se produce cuando se decide importar un nuevo fichero csv de un año en concreto. Se indica que es un proceso manual, ya que se entraran los datos del *loader_any* manualmente para después realizar el proceso de carga, que puede implicar el cambio de los valores del año actual.

La lista de comarcas, ámbitos y provincias se entra manualmente, ya que son pocos datos

Para determinar la lista con los nombres normalizados del municipio se basará en la información que suministran las fuentes de *poblacio.csv* y *equipaments.csv* a las que se *extraerá* en cada uno de ellos el dato de población, para después realizar una *unión* en donde se relaciona el nombre de una fuente (*poblacio.csv)* con el nombre de la otra (e*quipaments.csv)* a continuación se realizará un *conversion* para generar el nombre normalizado que será utilizado en el almacén de datos. Con ello tendremos relacionados el nombre normalizado con el nombre de *equipaments.csv* y con el nombre *poblacion.csv.* Durante el proceso de *carga* a dim_municipi se incorporaran los datos de comarca.

En el anexo (Anexo Modelo Lógico ETL basado en UML -Nivel 2: Paquete *ETL Comuns*.) del detalle en UML existe el correspondiente modelado detallado del paquete

3.2.3. Paquete *ETL Allotjaments*.

Es un conjunto de fuentes de datos, uno por cada año,y con estructuras diferentes. Así tenemos un tipo de *envoltorio* para el año 2006, otro tipo para los años 2007,2008,2009,2011 y el año 2012

necesita antes aplicar múltiples transformaciones.

El 2006 necesita una paso adicional de *conversión* al formato de los otros años.

En el caso del 2012 se ha divido el fichero original en otros 3 ficheros en que se guardan cada uno de los tres grandes bloques que contiene:

- registro de hoteles y campings,
- registro de las casas rurales
- un segundo registro de casas rurales según la asociación de turismo rural

A cada uno ha de tener su propia *conversión* y tras ello se aplica un *fusionar* los datos de las dos fuentes de turismo rural y finalmente un *unión* que lo combine todo y se pueda seguir con el proceso de transformación ya unificado con los otros años.

Una vez unificados los formatos de los distintos años, se aplica un *filtro* para detectar errores, calcular (si es necesario) e insertar listas de ajustes, finalmente se realizará un *transposición* para pasar de columnas por tipo de establecimiento a un fila por cada tipo de establecimiento.

Las datos de sobre las categorías de alojamientos son entrados manualmente y surgen a partir de las cabeceras del año 2012.

En el anexo (Anexo Modelo Lógico ETL basado en UML -Nivel 2: Paquete *ETL Allotjaments*.) del detalle en UML existe el correspondiente modelado detallado del paquete

3.2.4. Paquete *ETL Demografics*.

Tras importar *poblacio.csv* se realizará una *transposición* para pasar de los datos de habitantes en columna por años a una fila por año, después se realizará un *filtro* para verificar errores y aplicar ajustes si son necesarios.

Indicar que cuando venga futuros ficheros habrá que revisar si la fuente vuelven a repetir los últimos 6 años o viene solo el nuevo año.

En el anexo (Anexo Modelo Lógico ETL basado en UML -Nivel 2: Paquete *ETL 3.2.4. Demografics*.) del detalle en UML existe el correspondiente modelado detallado del paquete

3.2.5. Paquete *ETL Equipaments*.

El paquete *ETL Equipaments* se desglosa en dos grandes áreas.. Así tenemos por un lado la carga de los equipamientos y por otro la carga de las categorías de los equipamientos. Primero hay que realizar la importación de las categorías y después de los equipamientos.

Para el proceso de las categorías lo primero es *envolver* la entrada categorías de tal forma que el primero es la *familia* el segundo el *grup* y el tercero es *tipus* y el resto asignarlo a instalacion_1... instalación_ después hay que seguir dos direcciones:

- *Filtrar* categorías eliminando duplicados y errores. para cargarlo en el *dim_tipus_eq*
- *Transponer* instalaciones de columnas a una fila por instalación y *filtrar* eliminando duplicados para *cargarlo* en el *dim_instalacio.*

Para el proceso de equipamientos, para cada fila hacemos una *unión* con las categorías de equipamientos, aplicamos una *conversión* para colocar el nombre del municipio normalizado, aplicamos *filtros* para detectar duplicados y fallos y finalmente es *cargado* en fct_equipament,

deg_equipament y en *agg_equipament*

Con los equipamientos correctos e insertados hay que generar la relación entre el equipamiento y las instalaciones, para ello hay que hacer una *unión* incluyendo a la fila todas las claves foráneas y la clave del establecimiento y con la nueva fila obtenida hay que hacer una *transposición* de columna de instalación a filas para finalmente cargarlo en *fct_instalacio* y *agg_instalacio*.

En el anexo (Anexo Modelo Lógico ETL basado en UML -Nivel 2: Paquete *ETL Equipaments*.) del detalle en UML existe el correspondiente modelado detallado del paquete

3.3. Modelo Físico.

3.3.1. Sistema Gestor de Base de Datos.

La dimensión de las tablas y el volumen de los datos hace que no haya problemas de rendimiento con las consultas a la bases de datos.

La descripción de las instrucciones SQL para la generación de tablas del sistema dimensional por falta de tiempo no se han insertado por el momento en este documento. En todo caso seguirán las siguientes consideraciones:

- Como motor de la base de datos se puede utilizar tanto el **MyISAM** como el **InnoDB** este último ofrece almacenamiento transaccional **ACID** y las restricciones de **FOREIGN KEY** pero ambos temas no son fundamentales para un almacén de datos. En cuanto a MyISAN permite la consulta de datos sin procesos de bloqueo y búsquedas a texto completo. En un principio con esta última es suficiente.
- Los indices: Se crearán tantos indicies como atributos filtrables existan en las dimensiones y los hechos.
- Como es una base de datos pequeña, no hace falta sistemas de particionado de tablas u otros para optimizar las consultas.
- En los caso de nombres y de descripciones estará establecido tamaño del varchar al máximo. 255, para evitar problemas futuros de truncamiento.

A continuación se detalla la definición de las tablas del almacén de datos, están por orden alfabético:

```
CREATE TABLE `agg_equipament` (
  `clau_equipament_calcul` smallint(5) unsigned NOT NULL
AUTO_INCREMENT,
  `clau_any` smallint(4) unsigned NOT NULL,
  `clau_municipi` smallint(5) unsigned NOT NULL,
  `clau_comarca` smallint(5) unsigned NOT NULL,
  `clau_tipus_eq` smallint(5) unsigned NOT NULL,
  `quantitat` int(5) unsigned DEFAULT '0',
  PRIMARY KEY (`clau_equipament_calcul`),
  KEY `ibuscar` (`clau_any`, `clau_municipi`, `clau_comarca`,
`clau_tipus_eq`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `agg_instalacio` (
  `clau_equipament_calcul` smallint(5) unsigned NOT NULL,
  `clau_instalacio` smallint(5) unsigned NOT NULL,
  `clau_any` smallint(4) unsigned NOT NULL,
  `clau_municipi` smallint(5) unsigned NOT NULL,
  `clau_comarca` smallint(5) unsigned NOT NULL,
  `clau_tipus_eq` smallint(5) unsigned NOT NULL,
  `quantitat` int(5) unsigned DEFAULT '0',
  PRIMARY KEY (`clau_equipament_calcul`,`clau_instalacio`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `dim_any` (
  `clau_any` smallint(4) unsigned NOT NULL,
  `actual` smallint(2) NOT NULL DEFAULT '0',
  PRIMARY KEY (`clau_any`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `dim_categoria_establiments` (
  `clau_ce` smallint(5) unsigned NOT NULL AUTO_INCREMENT,
  `tipus` varchar(255) NOT NULL,
  `grup` varchar(255) NOT NULL,
  PRIMARY KEY (`clau_ce`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `dim_comarca` (
```

```
CREATE TABLE `dim_instalacio` (
```

```
  `clau_comarca` smallint(5) unsigned NOT NULL
AUTO_INCREMENT,
  `comarca` varchar(255) NOT NULL,
  `ambit` varchar(255) NOT NULL,
  `provincia` varchar(255) NOT NULL,
  PRIMARY KEY (`clau_comarca`),
  KEY `icomarca` (`comarca`),
  KEY `iambit` (`ambit`),
  KEY `iprovincia` (`provincia`),
  KEY `iterritori` (`provincia`,`ambit`,`comarca`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

```
  `clau_instalacio` smallint(5) unsigned NOT NULL
AUTO_INCREMENT,
  `instalacio` varchar(255) NOT NULL,
  `version` datetime DEFAULT NULL,
  PRIMARY KEY (`clau_instalacio`),
  KEY `idx_dim_instalacio_lookup` (`instalacio`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `dim_municipi` (
  `clau_municipi` smallint(5) unsigned NOT NULL
AUTO_INCREMENT,
  `municipi` varchar(255) NOT NULL,
  `ine` smallint(5) unsigned NOT NULL,
  `version` datetime DEFAULT NULL,
  `comarca` varchar(255) NOT NULL DEFAULT 'no definido',
  `ambit` varchar(255) NOT NULL DEFAULT 'no definido',
  `provincia` varchar(255) NOT NULL DEFAULT 'no definido',
  PRIMARY KEY (`clau_municipi`),
  UNIQUE KEY `municipi` (`municipi`),
  KEY `iterritori` (`provincia`,`ambit`,`comarca`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `dim_tipus_eq` (
  `clau_tipus_eq` smallint(5) unsigned NOT NULL
AUTO_INCREMENT,
  `familia` varchar(255) NOT NULL,
  `grup` varchar(255) NOT NULL,
  `tipus` varchar(255) NOT NULL,
  `version` datetime DEFAULT NULL,
  PRIMARY KEY (`clau_tipus_eq`),
  KEY `idx_dim_tipus_eq_lookup` (`familia`,`grup`,`tipus`)
) ENGINE=InnoDB AUTO_INCREMENT=112 DEFAULT
CHARSET=utf8;
```

```
CREATE TABLE `fct_demografic` (
  `clau_demografic` smallint(5) unsigned NOT NULL
AUTO_INCREMENT,
  `clau_any` smallint(4) unsigned NOT NULL,
  `clau_municipi` smallint(5) unsigned NOT NULL,
  `clau_comarca` smallint(5) unsigned NOT NULL,
  `habitants` int(5) NOT NULL DEFAULT '0',
  `dones` int(5) DEFAULT '0',
  `homes` int(5) DEFAULT '0',
  `extension` int(5) NOT NULL DEFAULT '0',
  PRIMARY KEY (`clau_demografic`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `fct_equipament` (
  `clau_equipament` smallint(5) unsigned NOT NULL
AUTO_INCREMENT,
  `clau_any` smallint(4) unsigned NOT NULL,
  `clau_municipi` smallint(5) unsigned NOT NULL,
  `clau_comarca` smallint(5) unsigned NOT NULL,
  `clau_tipus_eq` smallint(5) unsigned NOT NULL,
  `version` datetime DEFAULT NULL,
  `clau_deg_equipament` smallint(5) DEFAULT NULL,
  PRIMARY KEY (`clau_equipament`),
  UNIQUE KEY `ibuscar`
(`clau_any`,`clau_tipus_eq`,`clau_deg_equipament`),
  KEY `idx_fct_equipament_lookup`
(`clau_equipament`,`clau_any`),
  KEY `iagregar` (`clau_any`,`clau_municipi`,`clau_comarca`,
`clau_tipus_eq`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `fct_establiments` (
  `clau_establiments` smallint(5) unsigned NOT NULL
AUTO_INCREMENT,
  `clau_any` smallint(4) unsigned NOT NULL,
  `clau_comarca` smallint(5) unsigned NOT NULL,
  `clau_ce` smallint(5) unsigned NOT NULL,
  `places` int(5) NOT NULL DEFAULT '0',
  `establiments` int(5) NOT NULL DEFAULT '0',
  PRIMARY KEY (`clau_establiments`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `fct_instalacio` (
  `clau_equipament` smallint(5) unsigned NOT NULL,
  `clau_instalacio` smallint(5) unsigned NOT NULL,
  PRIMARY KEY (`clau_equipament`,`clau_instalacio`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

La descripción de las tablas del proceso ETL, se aplicará lo expuesto en su modelo lógico creándose en el momento del proceso, pero sus tablas se mantendrán para futuras incorporaciones. .

Para claridad y diferenciar existirán dos bases de datos. una para el almacén de datos y otra para el proceso ETL, ambas estarán gestionadas por el mismo servidor de bases de datos, lo que facilitará el proceso de carga y consulta al almacén de datos.

3.3.2. EQUIPAMIENTO

Todo el software utilizable está concebido en una arquitectura cliente/servidor. Así que, para este proyecto el equipo servidor puede compartir recursos con otros servicios, ya que, el volumen no es grande y tampoco se espera una carga de trabajo intensa.

Solo en el caso de que se aportaciones de datos con volúmenes mayores e información mas compleja se podría pensar en utilizar un servidor dedicado (virtual o físico)

En el caso de que el producto creado sea un producto de un solo uso se puede pensar en utilizar los sistemas en la nube para alojar los datos y pagar por solo el tiempo de uso.

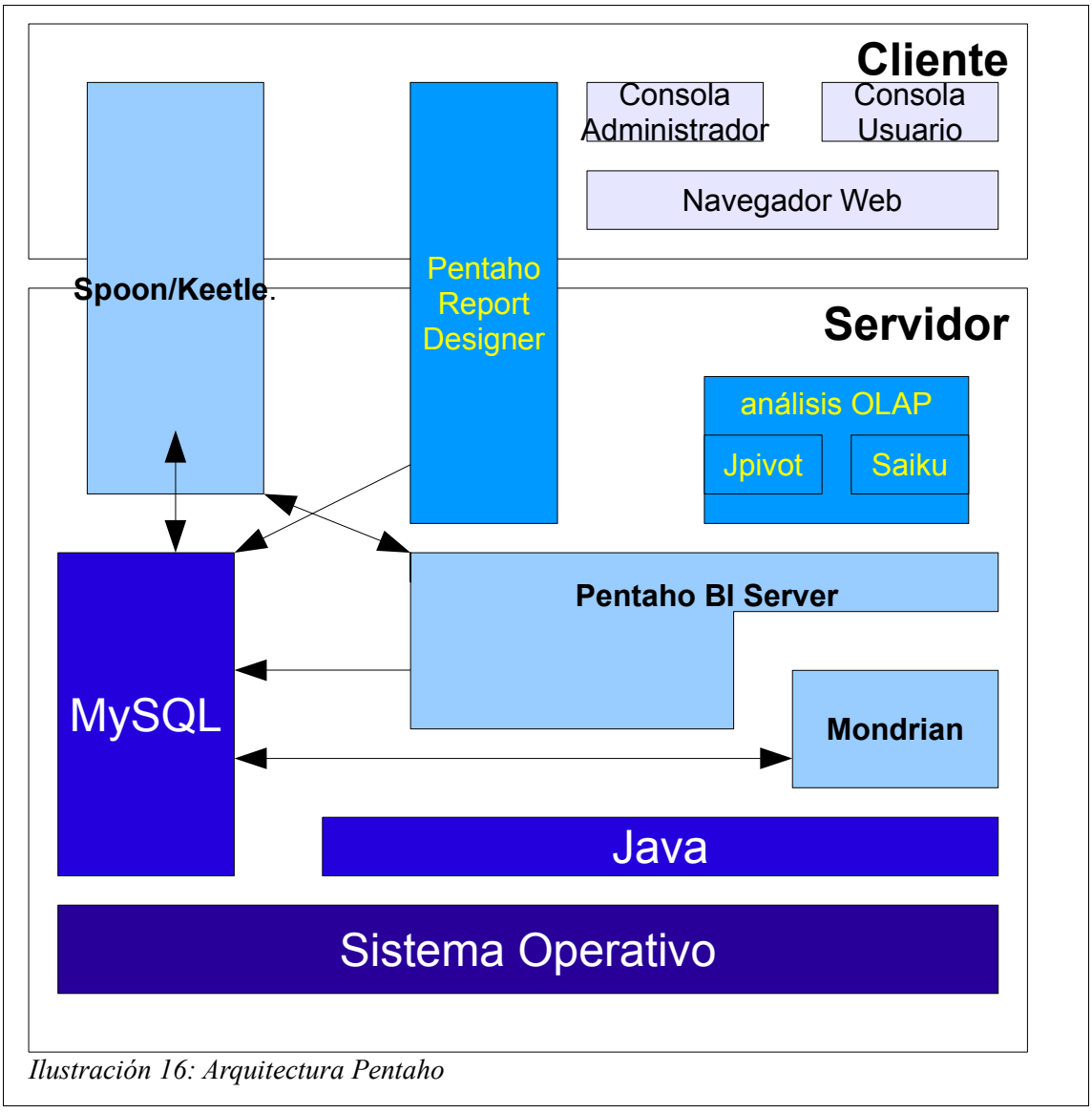


Ilustración 16: Arquitectura Pentaho

En cualquier caso la estructura ha de ser de cliente/servidor estando en la parte cliente las herramientas consulta y administración del servidor, ambas se ejecutan sobre cualquier navegador que permita AJAX, se podrán utilizar las herramienta de ETL Spoon / Keetle tanto en un equipo cliente, como en el mismo equipo servidor, lo mismo se puede decidir para Pentaho ReportDesigner y tanto el servidor de Pentaho BI como el de las base de datos ha de estar en un equipo con funciones de servidor (virtual o físico)

Al ser la pila de productos de Pentaho desarrollado en java se puede poner en cualquier S.O. (Windows, Unix, Linux, Mac) que soporte la maquina virtual java.

El MySQL se puede instalar entre otros en equipos Windows y Linux.

4. Capítulo Implementación

4.1. Exposición del trabajo de integración de los datos.

Todo el proceso de transformación se ha realizado en ***Spoon*** el ***Data Integracion de Penthao*** en una ***Stagging Area*** en el servidor **Mysql** llamada *tfcetl* con los mismos permisos de usuario que las demás bases de datos: *root > tfcmmd*

El primer paso es insertar los ficheros planos a la base de datos es realizado de forma manual, también es manual la separación de los datos de *establiments* del 2012 en 3 ficheros, el volcado a la base de datos se realiza mediante la orden de importación del **MySQL**, teniendo en cuenta los distintos tipos de caracteres que tienen cada uno de ellos.

Ejemplos de los comandos utilizados:

```
load data infile '/tmp/TFC/Equipaments.csv'into table equipaments fields terminated by ',' OPTIONALLY ENCLOSED BY '"' IGNORE 1 LINES;
```

```
load data infile '/tmp/TFC/est2006.csv'into table allotjaments_2006 fields terminated by '\t' TERMINATED BY '\r\n';
```

```
load data infile '/tmp/TFC/poblacioUTF8.csv' into table poblacio fields terminated by ';' IGNORE 1 LINES;
```

En este caso se hizo una transformación primero del fichero UTF8

A partir de ese momento en que están los datos cargados en la base de datos *tfcetl* se inicia todo el proceso de transformación con la herramienta **Spoon**.

En **Spoon** se ha creado un trabajo que sirve de guía y orientación de los pasos que hay que hacer, no es recomendable lanzarlo desde ahí (de hecho el siguiente paso a inicio es una cancelación de la tarea, impidiendo realizar los siguientes trabajos) Es mejor lanzar uno a uno, y más recomendable es realizar un seguimiento mediante la ventana de logs "***job metrics***", para ello es mejor lanzar cada trasformación desde la misma pagina de transformación. Así se obtienen información detallada de las "métricas del proceso" que muy son útiles ya que informan de los registros leídos, escritos, actualizados o borrados y con ello se tiene cumplida cuenta que se están procesando todos los registros entrados.

Para el proceso se ha seguido lo propuesto el documento lógico ETL del diseño con los ajustes para adaptarlo a la herramienta **Spoon**.

Si entramos en el **Spoon** y revisamos el trabajo etiquetado como "*Panel de trabajos migración TFCDW*" vemos que se ha divido en diversas tareas y estas a su vez en múltiples transformaciones.

Tememos como grandes bloques: las transformación "*dim_any*" y "*insertar Categories i Comarques*" y las transformaciones "*treball dim_poblacio*", "*treball fct_demografic*", "*treball allotjaments*", "*treball equipaments*".

A continuación se van a detallar dichas transformaciones agrupándolas por las misma división que se hizo en paquetes durante el análisis y el diseño: c*omunes, allotjaments, demografics y equipaments*

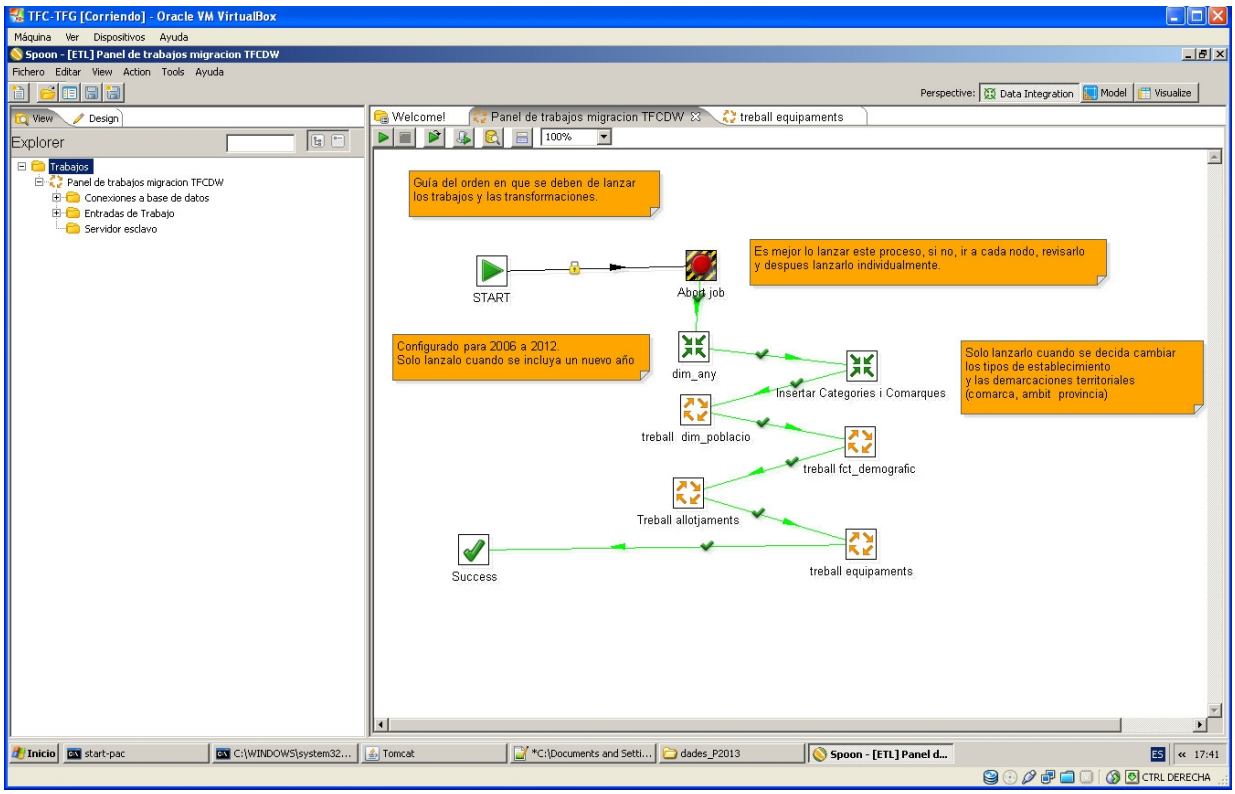


Ilustración 17: Imagen del trabajo que funciona como guía de los pasos a realizar

4.1.1. DATOS COMUNES.

Transformación para la dimensión temporal (DIM_any)

En este paso se aprovecha las funcionalidades de generar filas y secuenciar de **Spoon** para crear una secuencia desde el 2006 a 2012 y con actual de 0 a -6.

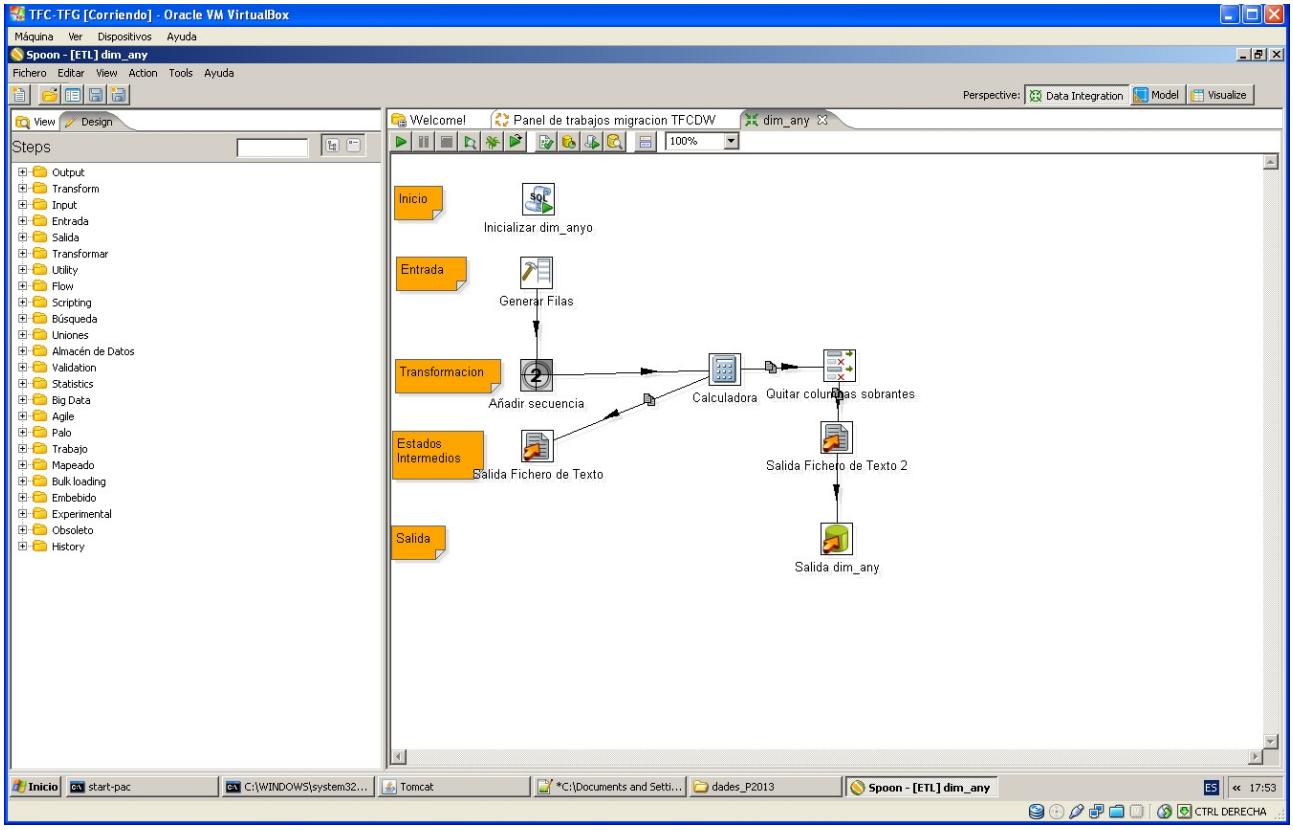


Ilustración 18: Generación de la dimensión temporal

Transformación para *Categories de establiments i Comarques*:

Contiene dos scripts SQL que mediante "**inserts**" **SQL** se entran las tablas de comarca y categorías de establecimientos.

Los nombres de las comarcas se han sacado de la documentación oficial, y su asignación al "*ambit*" se realiza según la convención del 2012.

Transformación *Municipi* dim_poblacio:

Responsable de identificar los municipios, tanto si provienen de equipamientos como de población, además tiene la responsabilidad de crear los ficheros intermedios que serán utilizados para recuperar correctamente el municipio durante las transformaciones para los datos demográficos y de equipamientos.

Tal como se indico en el modelo Lógico existe un fichero intermedio que contiene el nombre del municipio y comarca que se "viene" de cada una de las fuentes y unas columnas a mayores con el nombre y comarca "definitivos" que se utilizarán en el sistema.

Por lo tanto, hay que siempre asegurarse que se ha lanzado esté paso y de que los ficheros intermedios contienen la información correcta.

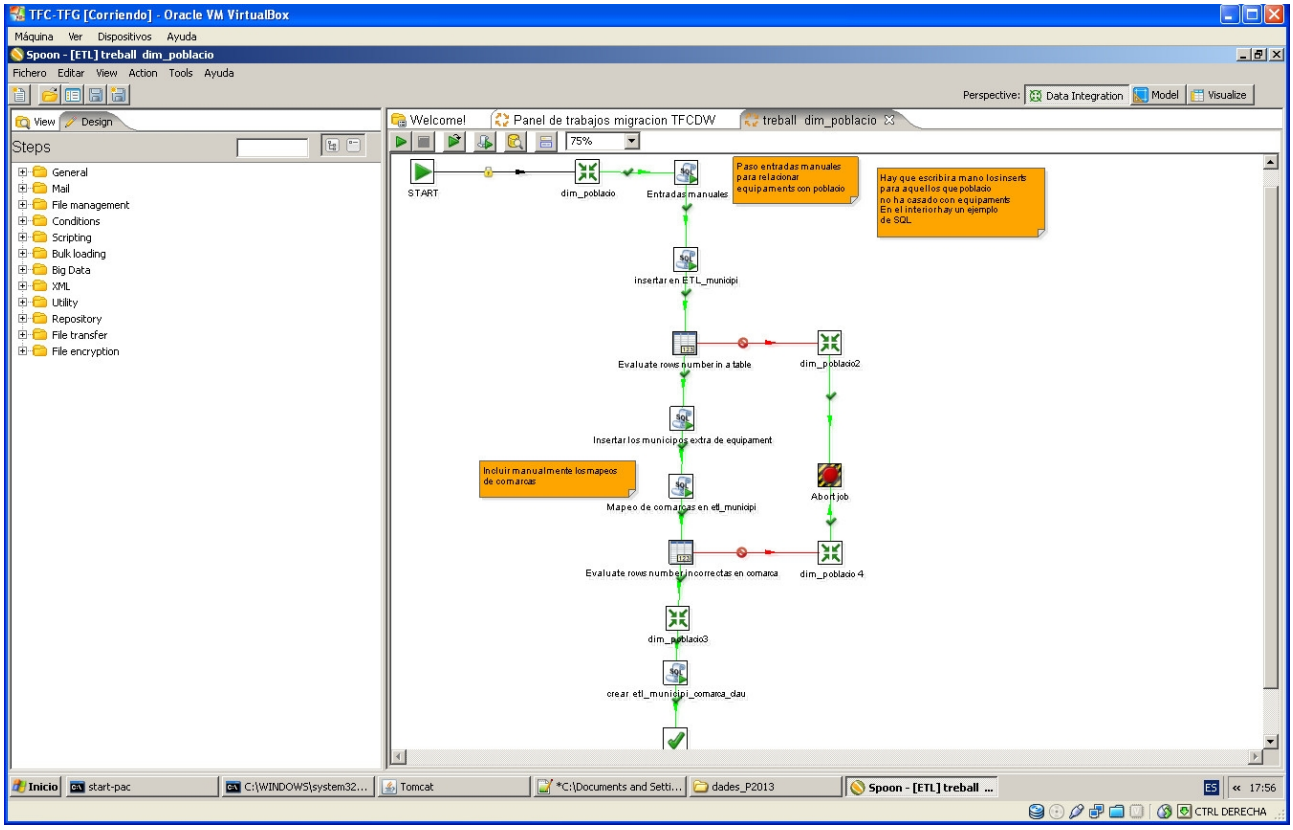


Ilustración 19: Generación de nombres de municipios

Como observación indicar que hay un número de municipios que están en *equipaments* y no en *població*, aún así se han incorporado en el sistema ya que se va a trabajar principalmente con información a nivel de comarca. Pero es importante saber que en habrán más equipamientos que los que hubiese si solo se daban por correctos los municipios existentes en demografía.

4.1.2. Demográficos.

Se realiza primero una transposición, para después limpiar de puntos y valores incorrectos en los campos de números, por ejemplo: los *"nd"* se han pasado a 0. Se recupera las claves para comarca y municipio y se verifica que lo tengan todos los registros (si el proceso de población ha ido bien ese paso no debería de dar error).

Finalmente se actualiza el *fct_demografic* y como un último paso debido a que los datos de los *homes* y *dones* solo están en el 2012 se ejecuta un ***SQL script*** para recalcular los de los años anteriores a partir de la proporción dada en el 2012

Si d se corresponde al registro del año que se esta procesando y d12 corresponde al registro del año 12 tenemos.

d.homes = TRUNCATE(d.habitans *d12.homes / d12.habitants,0)

d.dones = d.habitants = TRUNCATE(d.habitans *d12.homes / d12.habitants,0)

Observación el municipio **Canonja** solo tiene datos a partir del 2012

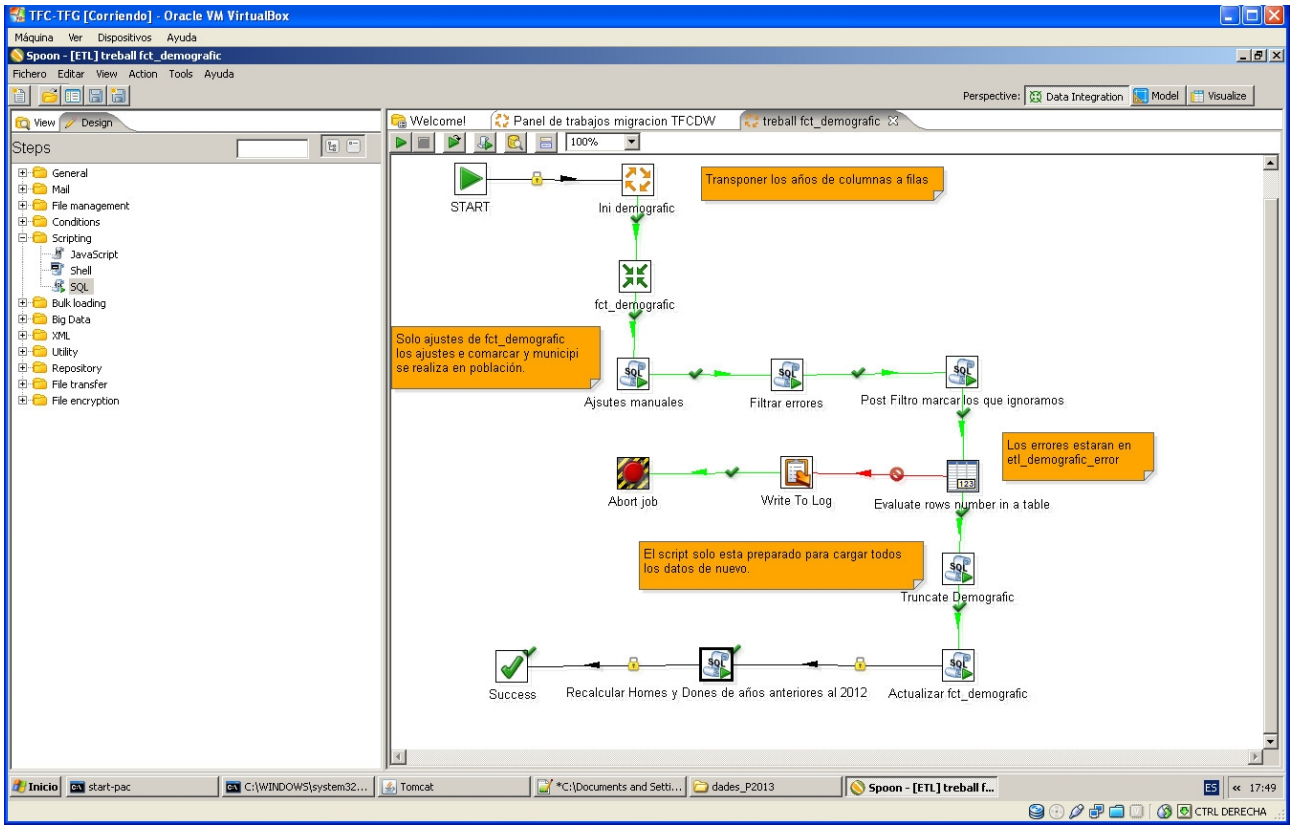


Ilustración 20: Transformación de datos demográficos

4.1.3. ALOJAMIENTOS

En el año 2006 los '*campigs privats*' son asignados camping de lujo y se recalcula las *Masoverias* e' *independents* con los datos del 2007: se pasa el mismo número de *masoveria* del 2007 al 2006 y el resto pasan a ser *independents*.

Para los demás para cada caso se normaliza la entrada de datos, es decir se pasa a una tabla intermedia que contiene las columnas normalizadas y se incluyen los ajustes manuales necesarios mediante scripts SQL . En el caso del 2012 se comparan las dos fuentes de datos que existen para los campings.

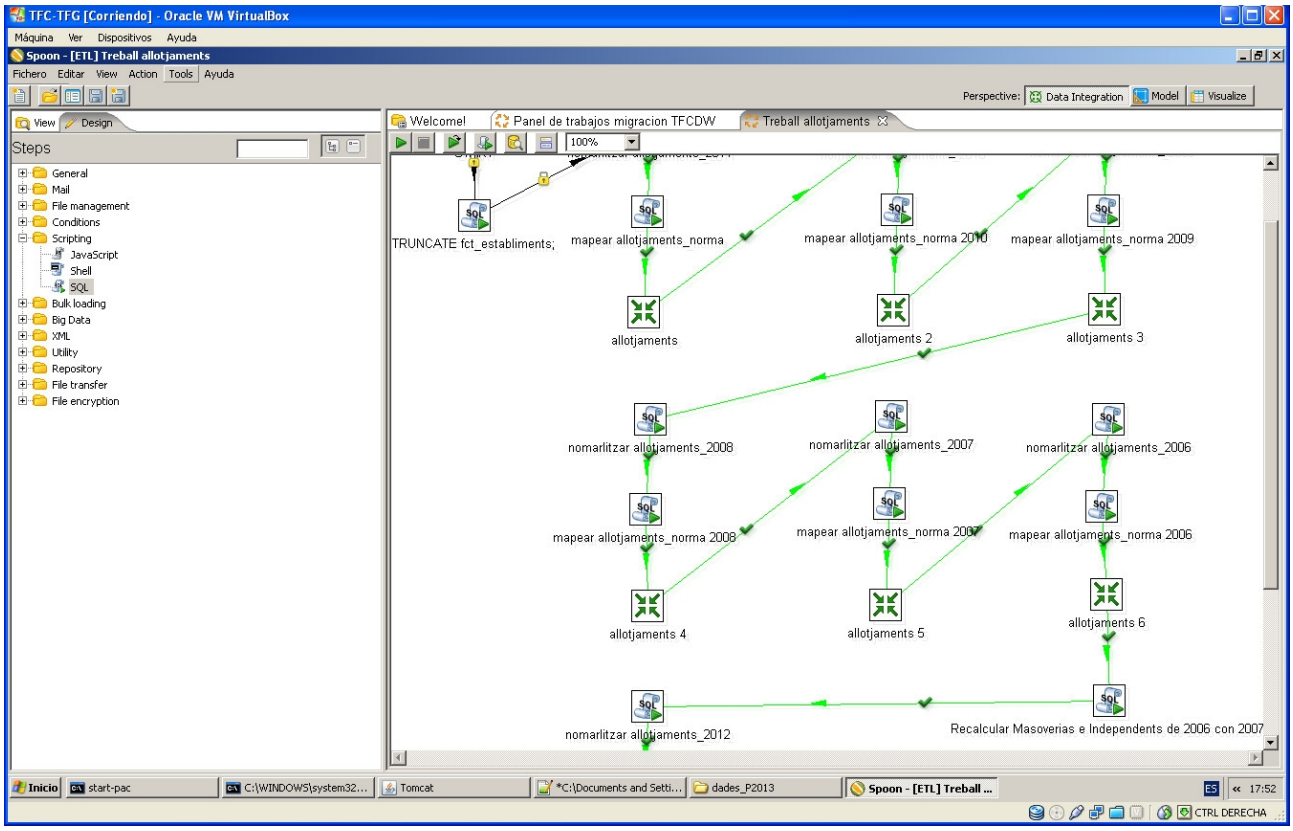


Ilustración 21: Transformación alojamientos

Después cada uno de los años (ver imagen en la siguiente página) se pasa al proceso de filtrado y validación y mediante un *javascript* se validan que los totales dentro del registro cuadren, o al menos que uno de ellos lo haga.

Se valida que los índices a *comarca* sean los correctos y se tipifica el registro de entrada indicando si se corresponde a una comarca, a un ámbito, a una provincia o al país

Finalmente se ignoran los que no corresponden a comarca, se registran aquellos en los que hay una fallo y con los que el registro corresponde a comarca se aplica otro *javascript* que convierte este registro en múltiples filas en la base de datos una por cada tipo y categoría de establecimiento y se normaliza el calculo de las plazas de camping según la regla del 2011

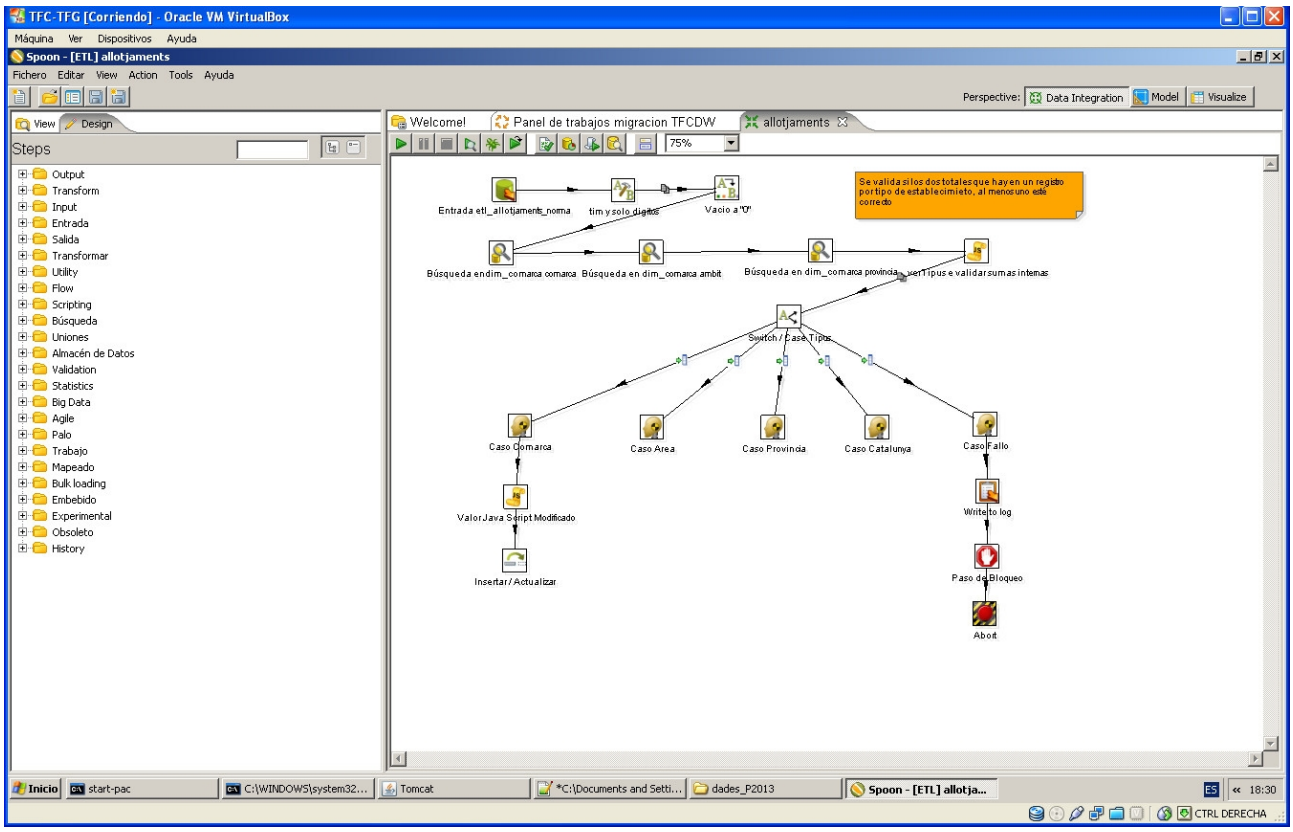


Ilustración 22: Transformación alojamientos descarte de filas.

4.1.4. EQUIPAMIENTOS

En esta transformación con respecto al modelo lógico de migración se ha realizado cambios agrupando operaciones en busca de optimizar el tiempo de procesos ya hay que procesar 31771 registros.

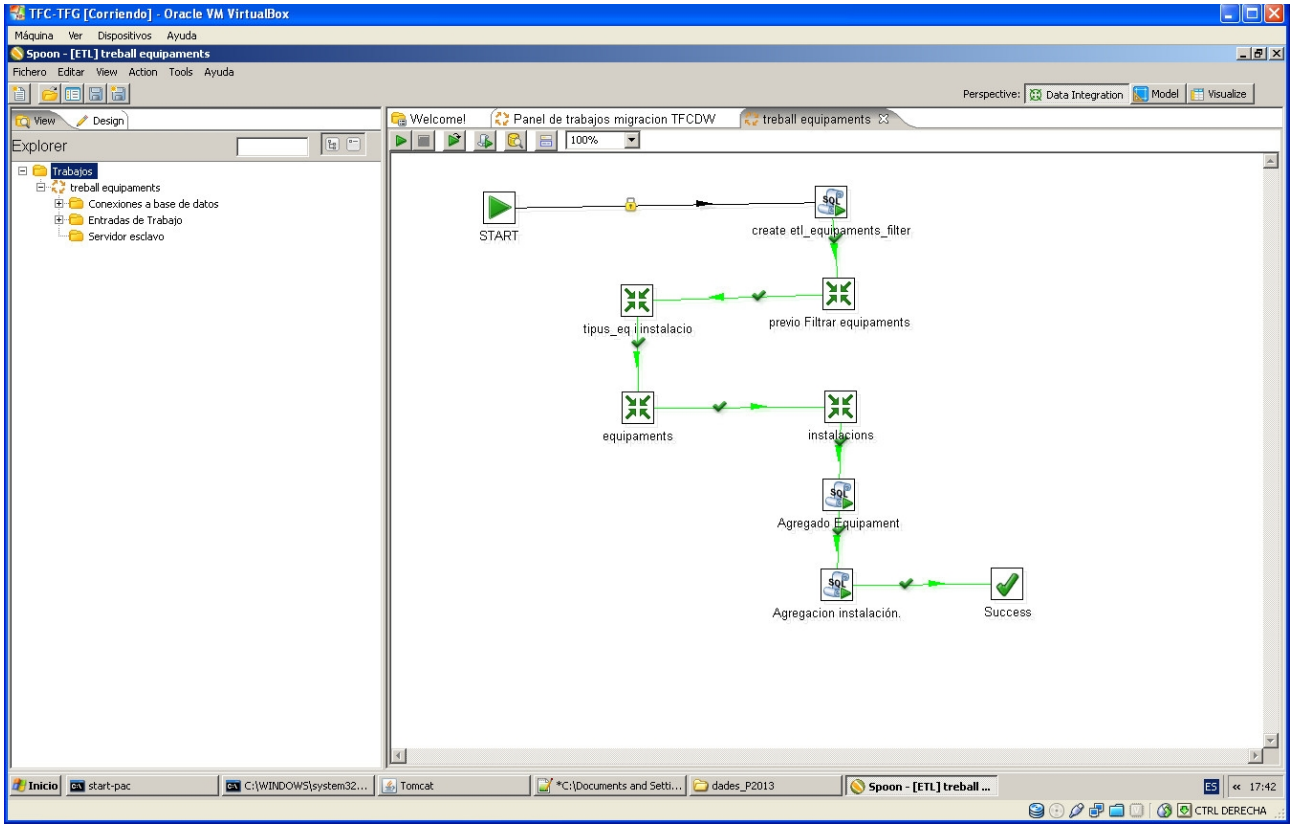


Ilustración 23: Transformación de equipamientos

Se realiza un transformación previa para retirar los registros duplicados y se le da al registro un valor único para ser referencia en todo el proceso de transformación, en ese instante también se les asigna las claves de municipio y comarca

Durante dicha transformación se encontraron 149 de repetidos, para detectarlos se creó una verificación por *nom* y *cp* (código postal), más el resultado de aplicar md5 a categoría, para así poder tener en cuenta el caso de que un mismo equipamiento perteneciera varios tipos de equipamiento.

El siguiente paso fue determinar los tipos de equipamiento e instalación, a partir del campo categorías, del que se aplico un *script* que recoge los 3 primeros campos de categoría como pertenecientes a al tabla de tipo de equipamiento (*familia, grup tipus*) y el restante para la tabla de instalación.

Una vez separados se aplicaron transformaciones para retirar las redundancias de familia y grupo y tipo por un lado y por otro las "instalaciones" para ser guardados en sus correspondientes dimensiones. A mayores se crean los ficheros intermediarios para que en posteriores pasos enlazar el equipamiento con los tipos de equipamiento e instalación que le corresponden.

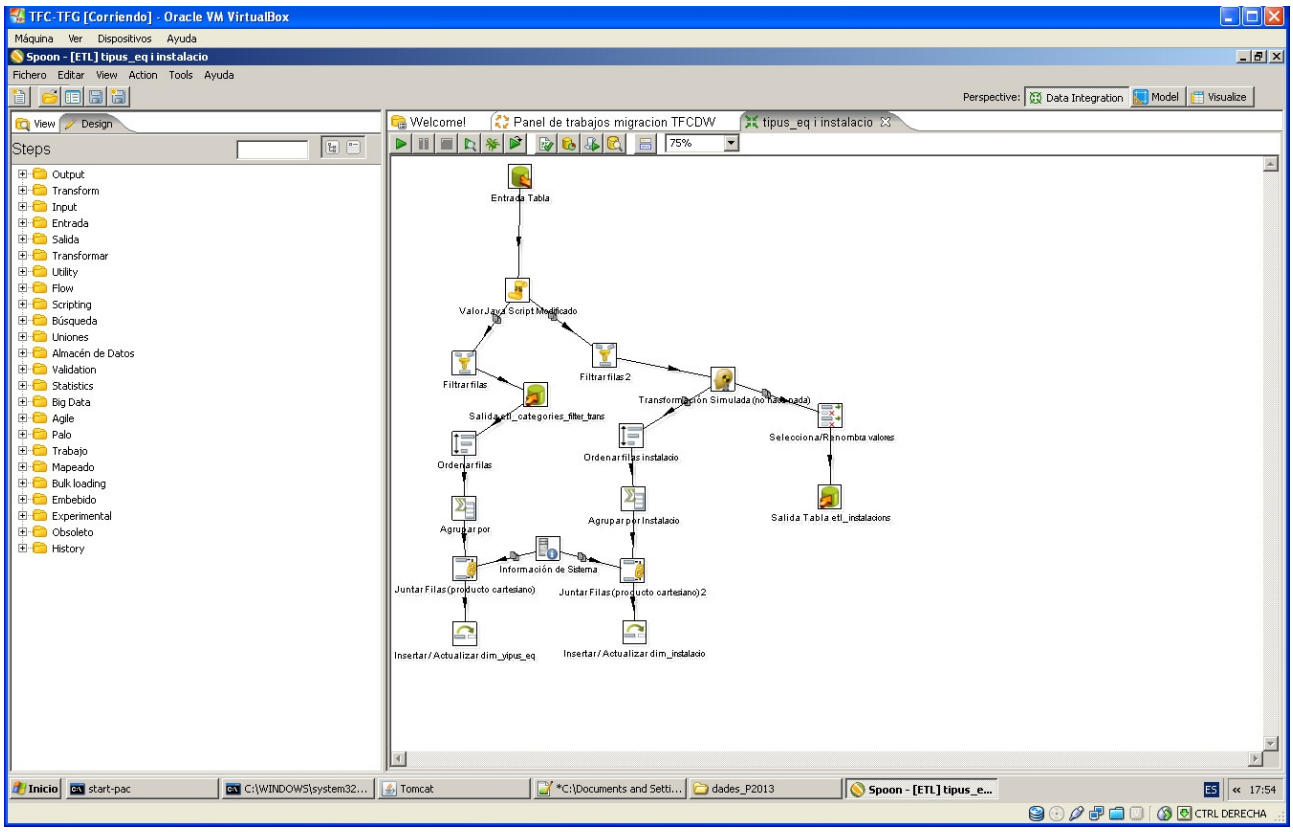


Ilustración 24: La generación de los categorías de equipamientos.

El siguiente paso *equipaments* es el encargado de guardar las tablas de *fct_equipaments*, *deg_equipaments* (en *deg_equipaments* estará los datos de dirección del equipamiento)

Durante el proceso se detectaron 75 equipamientos que tenían mas de un registro con al misma combinación de tipo de equipamiento, eso quería decir que la diferencia estaba en las categorías que pertenecen a *instalació*. Por lo tanto han sido ignorados, para la carga en equipamientos, pero siguen el proceso de transformación para asignar dichas instalaciones al equipamiento en cuestión.

El siguiente paso es a partir de los ficheros intermedios generados en los pasos anteriores asignar a cada equipamiento relación con su lista de instalaciones.

Finalmente se lanzan los ***scripts SQL*** para que se calcule los ficheros de agregación de equipamientos e instalación con la suma de equipamientos por comarca.

4.2. Proceso de creación de las consultas OLAP y de los informes.

El proceso se puede dividir en dos grandes secciones , aunque en realidad están intimamente relacionadas, por una lado el proceso de creación de los *metadatos* y por otro la creación de los informes.

4.2.1. Proceso de creación de los metadatos y esquemas.

Para el proceso de generación de los cubos de las consultas OLAP y de los informes, se intento

definir una única estructura única de *metadatos* y *esquemas OLAP*, común para todos los informes y consultas.

Los metadatos y los esquemas tienen por funcionalidad ser una capa intermedia entre los informes y la base datos, que permite añadir información sobre los datos a mayores de la que se puede definir en el DDL de las Bases de datos.

El *esquema* es utilizado por el servidor **Mondrian ROLAP** y los *metadatos* son utilizados por el gestor de informes. Cada uno tienen sus propias gestiones de configuración y a su vez ambas estructuras se pueden crear desde la consola de **Pentaho BI** cuando se define una fuente de datos**.**

Cuales son las diferencias de crearlos en la consola del **Pentaho BI** o en los gestores específicos:

- Los gestores específicos el **Pentaho Metadata Editor** para los informes y el **Pentaho Schema Workbench** para los esquemas OLAP a diferencia de la consola permiten definir la estructura como un todo, es decir, permite definir elementos que después son reutilizados en las distintos informes. Así por ejemplo se define una sola vez la dimensión territorial.
- La ventaja de la consola del **Pentaho BI** es que ofrece un configuración simple, fácil y de uso muy intuitivo.

Pero la idea se abandono ya que el esfuerzo para entrar los metadatos y esquemas, no compensaba con el uso posterior que se hacia de ellos , concretamente:

- Para la elaboración de los informes tanto los metadatos como los esquemas no valían de por se y se habían precisaban de utilizarla escritura de consultas en formatos **XML o MDX**, por lo tanto representaba el mismo esfuerzo que hacer un consulta **SQL**
- Los informes y cubos requeridos son de información cruzada entre las diferentes estrellas por lo que la única manera de hacerlo es que se definan los esquemas a partir de consultas SQL.
- La ventaja de tener una sola definición de los datos utilizando las herramientas especificas en esté almacén de datos no era representativa por el escaso número de jerarquías, por lo tanto no es problema duplicar la configuración varias veces.

Así que se decidió simplemente aplicar una convención de nombres a ser utilizado en cada configuración de fuente de datos en **Pentaho BI Server**.

- Dimensión "*Temporal*" para din_any
  - Jerarquía *Anual* {*anual*}
  - Jerarquía *Antiguetat* {*actual*}.
- Dimensión "*Territorial*" desde comarca.
  - Jerarquía territorial {*Provincia, Ambit, Comarca* }
- Dimensión "*Territorial*" desde municipio
  - Jerarquía territorial {*Provincia, Ambit, Comarca, Municipio* }

- La dimensión tipos de equipamiento: "*Categoría*"
  - Jerarquía *categoría* {*Familia, Grup, Tipus* }
- Dimensión categorías de establecimiento: "*Categoría*"
  - Jerarquía *categoría* {*Tipus, Grup* }

4.2.2. Proceso de creación de los informes

Para realizar los listado primero se intentaron con la herramienta **Web Based Ad Hoc Query and Reporting** disponible en la consola de **Pentaho BI** pero nos encontramos con el problema desde las opciones de pantalla no permite incluir datos calculados, así que se descarto. En todo caso, se aprovecho para realiza una maquetación aceptable de los informes a modo de prototipo para después incluir los datos que faltan en el generador de informes **Pentaho Report Designer**

Con el **Pentaho Report Designer** se esperaba que con la importación de los dos prototipos al gestor de informes también incluyera la información de los metadatos y de la fuente de datos, pero eso no fue así, se probaron otras opciones como realizar conexiones vía metadatos y esquemas OLAP pero tampoco se ajustó a lo que se necesitaba, así que finalmente se utilizo como fuentes de datos las consultas SQL.

Para consulta al cubo OLAP se utilizo el frontal **JPivot** que accede a los cubos de **Mondrian Server** disponible en la consola **Pentaho BI,** existe un segundo frontal **Saiku,** pero no permite entrar campos calculados a partir del formato **MDX** y después, a mayores, se encontró con el problema que una vez guardado daba errores internos en el log de la consola al volverlo a cargar y olvidaba lo realizado

4.3. *Configuraciones* y accesos

Para simular un usuario final se ha creado el usuario *l'Observatori Nacional d'Ocupació* **ondo** (clave **ondo**) con el perfil de **ceo**

Se han asignado permisos de ejecución y poder programar procesos en diferido para los consultas OLAP creadas y para el caso de los ejemplos de acceso a los tres cubos (*estrellaDemograficTerritorial*, *estrellaEquipaments*, *estrellaEstabliments*) tiene todos los permisos. (de echo no he podido restringir los permisos de los cubos generados por **Saiku**

El caso de los informes no hay ningún tipo de permisos al no poder publicarlos dentro del **Servidor BI**

Las bases de datos creadas son la *tfcetl* para la ***Stagging Area*** y la *tfcdw* para el almacén de datos.: *root* > *tfcmmd*

5. Presentación de los informes realizados.

5.1. *Consultas OLAP*

Se han creado los siguientes cubos que se acceden desde la consola **Pentaho BI Server** y estan en la carpeta TFCDW, han sido creados por el usuario tfcmmd y usuario ondo puede consultarlos al tener un perfil de *ceo.*

Observación: me he encontrado con muchos problemas, con las consultas OLAP ya que en ocasiones desaparecen o el servidor no responde. Y en ciertas ocasiones el usuario ondo no ve todas las consultas a las que tiene acceso. Supongo que son problemas debido a los recurso escasos de memoria y procesador.

5.1.1. Consulta Establiments Comarques Ratios

Cumple con los requerimientos de tener las siguientes ratio.

- Total d'establiments
- Total de places
- Oferta mitjana de places
- % de places respecte població
- Indicador de places vs persones
- Quantitat de places ofertes / superfície del territori
- Nombre d'establiments/Nombre d'equipaments

Se puede navegar en territorial, hasta nivel de comarca y por las categorías de los establecimientos.

Los datos de habitantes y equipamientos no son validos en todas las combinación del cubo ya que el sistema no permite elegir la función de agregación según la dimensión que se intervenga, es decir habitantes y equipamientos solo deberían de sumar a nivel territorial, a nivel de años debería de realizar promedio y con tipos de equipamiento no debería de hacer nada.

La fuente de datos utilizada es una consulta SQL.

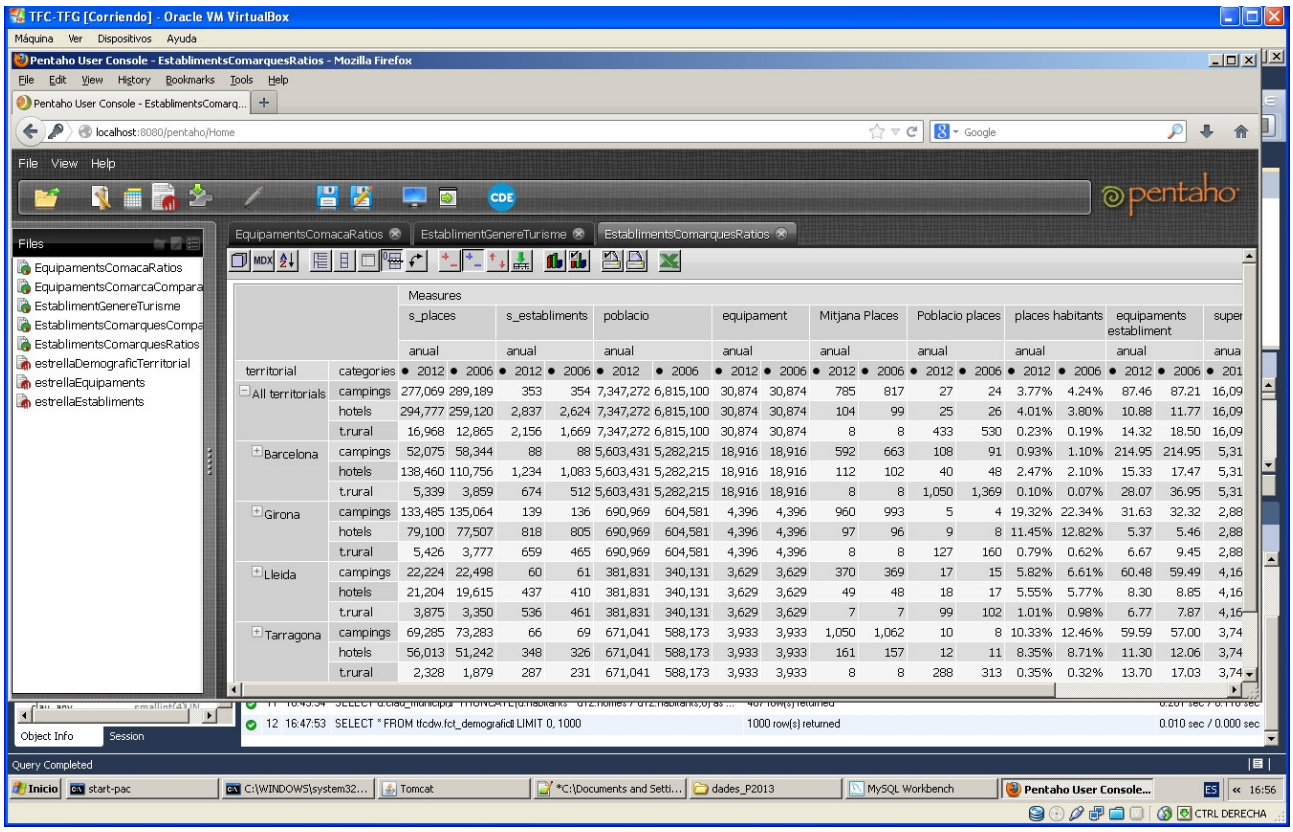


Ilustración 25: Ratios Establecimientos comarca lado izquierdo

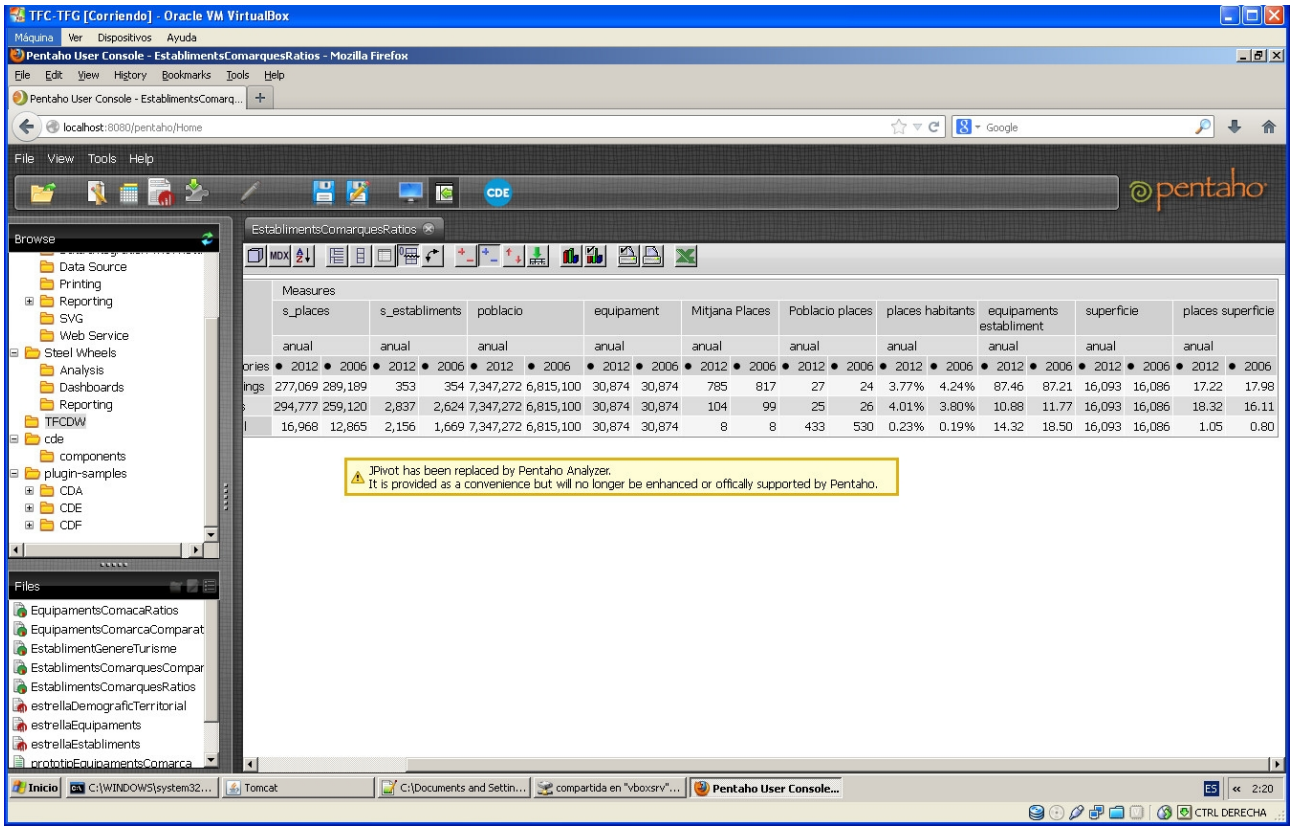


Ilustración 26: Ratios Establecimientos comarca lado derecho

5.1.2. Consulta Establiments Genere Turisme

Cumple con los requerimientos de tener los siguientes ratio.

- Total d'establiments
- Total de places
- Indicador d'establiments vs habitants per gènere
- Nombre d'establiments/Nombre d'equipaments

Ademas muestra el total por género, el total de la población y en vez de mostrar todos los equipamientos se muestra una selección de ellos bajo el criterio que tengan que ver con turismo: *Esport i lleura, cultura, turismo.* Y finalmente un ratio el % de mujeres.

Observación en teoría el ratio del % de mujeres debería de ser constante en todos los años, pero hay diferencias decimales por el redondeo que se ha realizado para calcularlos.

Los datos de habitantes y equipamientos no son válidos en todas las combinaciones del cubo ya que el sistema no permite elegir la función de agregación según la dimensión que se intervenga, es decir habitantes y equipamientos solo deberían de sumar a nivel territorial, a nivel de años debería de realizar promedio y con tipos de equipamiento no debería de hacer nada.

La fuente de datos utilizada es una consulta SQL.

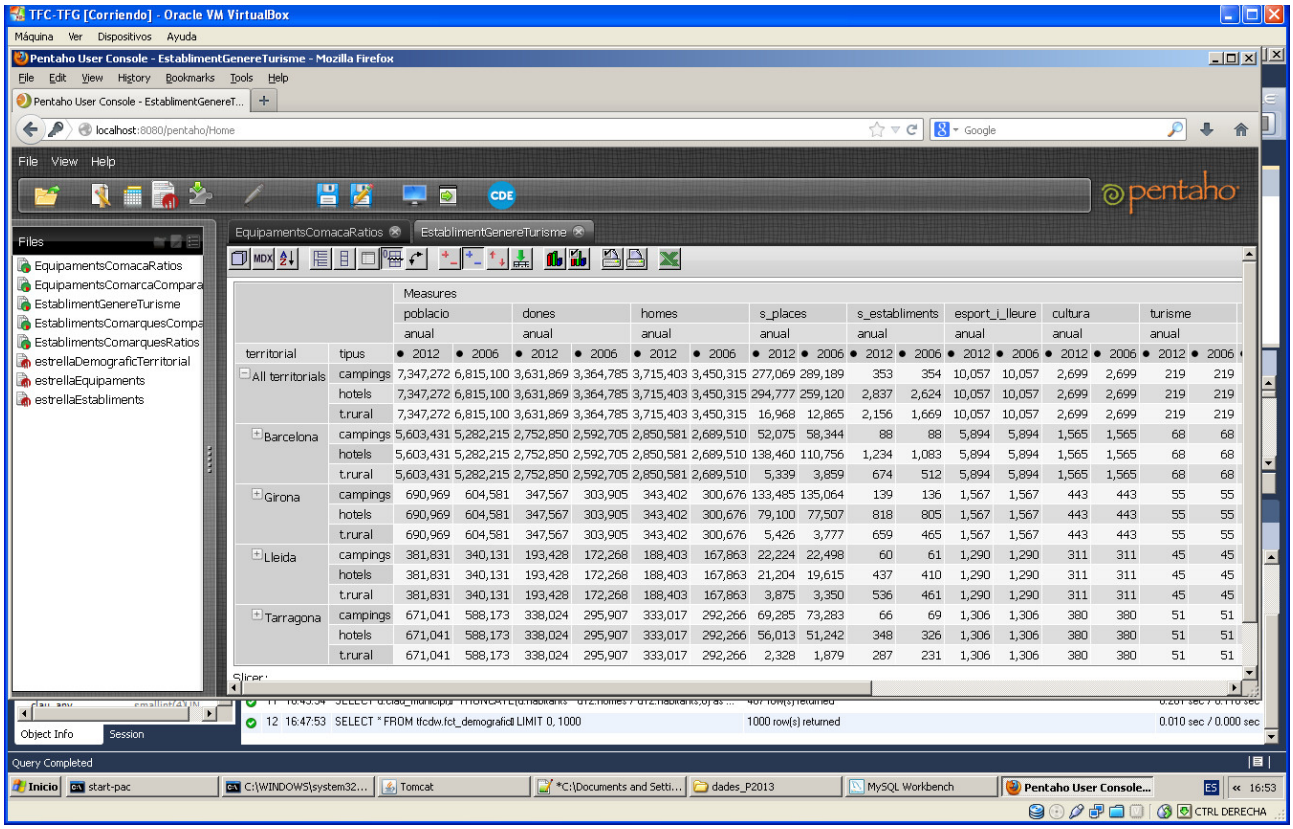


Ilustración 27: Volcado de la pantalla de la consulta establecimientos género lado izquierdo

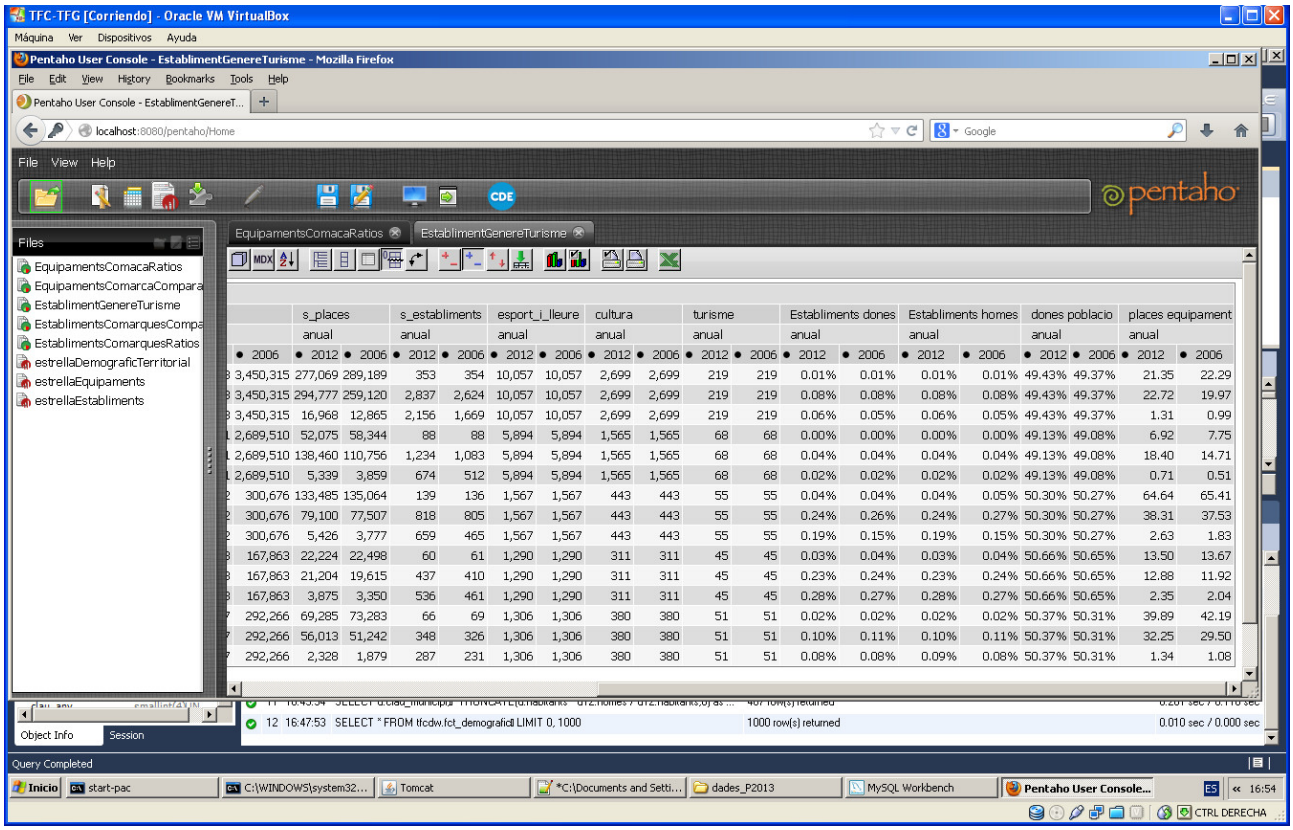


Ilustración 28: Volcado de la pantalla de la consulta establecimientos género lado derecho

5.1.3. Consulta Equipaments Comarca Ratios

Cumple con los requerimientos de tener los siguientes ratio.

- Nombre d'establiments/Nombre d'equipaments
- % de població per equipament
- Indicador d'equipaments vs població

Ademas muestra los agregados de equipamientos, habitantes y establecimientos y se puede navegar por las dimensiones anual, tipos de equipamientos y territorial hasta comarca.

Los datos de habitantes y establecimientos no son validos en todas las combinación del cubo ya que el sistema no permite elegir la función de agregación según la dimensión que se intervenga, es decir habitantes y establecimientos solo deberían de sumar a nivel territorial, a nivel de años debería de realizar promedio y con tipos de equipamiento no debería de hacer nada.

La fuente de datos utilizada es una consulta SQL.

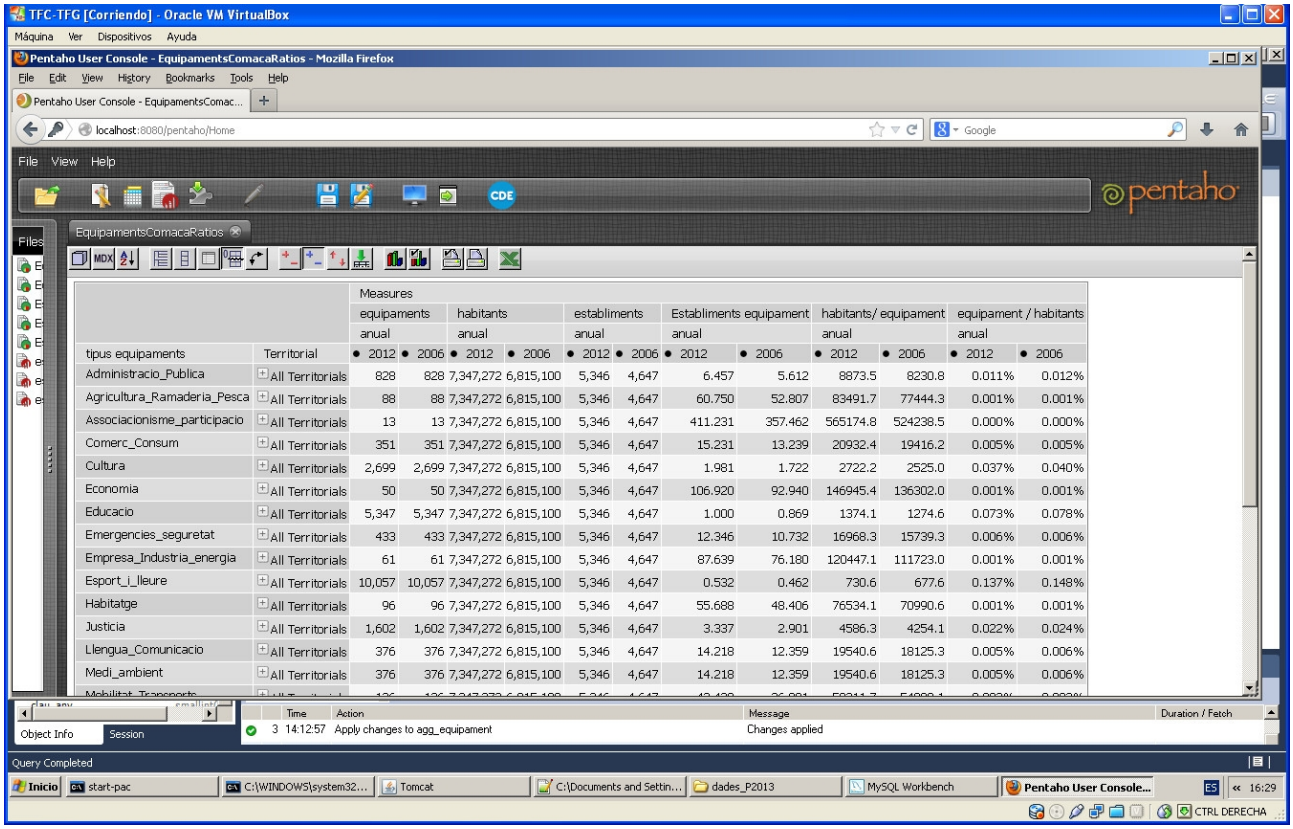


Ilustración 29: Volcado de pantalla con la consulta equipamientos comarca

5.1.4. Consultas OLAP de los data-marts.

A mayores se ha creado las fuentes de conexión y los esquemas de análisis para poder acceder mediante consultas OLAP a las tres estrellas definidas, el usuario tiene libertad de crear las consultas que deseé, se han dejado accesibles 3 consultas. Han sido creadas con **Saiku** por que es una herramienta intuitiva y apta para el usuario, aunque, tiene un problema ya que ciertas consultas guardadas en **Saiku** después no son recuperables, sobre todo si se realizan con varios niveles de

profundidad.

Las consultas son:

- estrellaDemograficTerritorial Que hace uso de la fuente/esquema DemograficTerritorial, que permite navegar hasta nivel de municipio.
- estrellaEquipaments Que hace uso de la fuente/esquema: estrellaEquipaments, que permite navegar hasta nivel de municipio
- estrellaEstabliments, Que hace uso de la fuente/esquema: estrellaEstabliments

En las capturas de pantalla a parte de la consulta se puede apreciar la estructuras de las dimensiones y medidas.

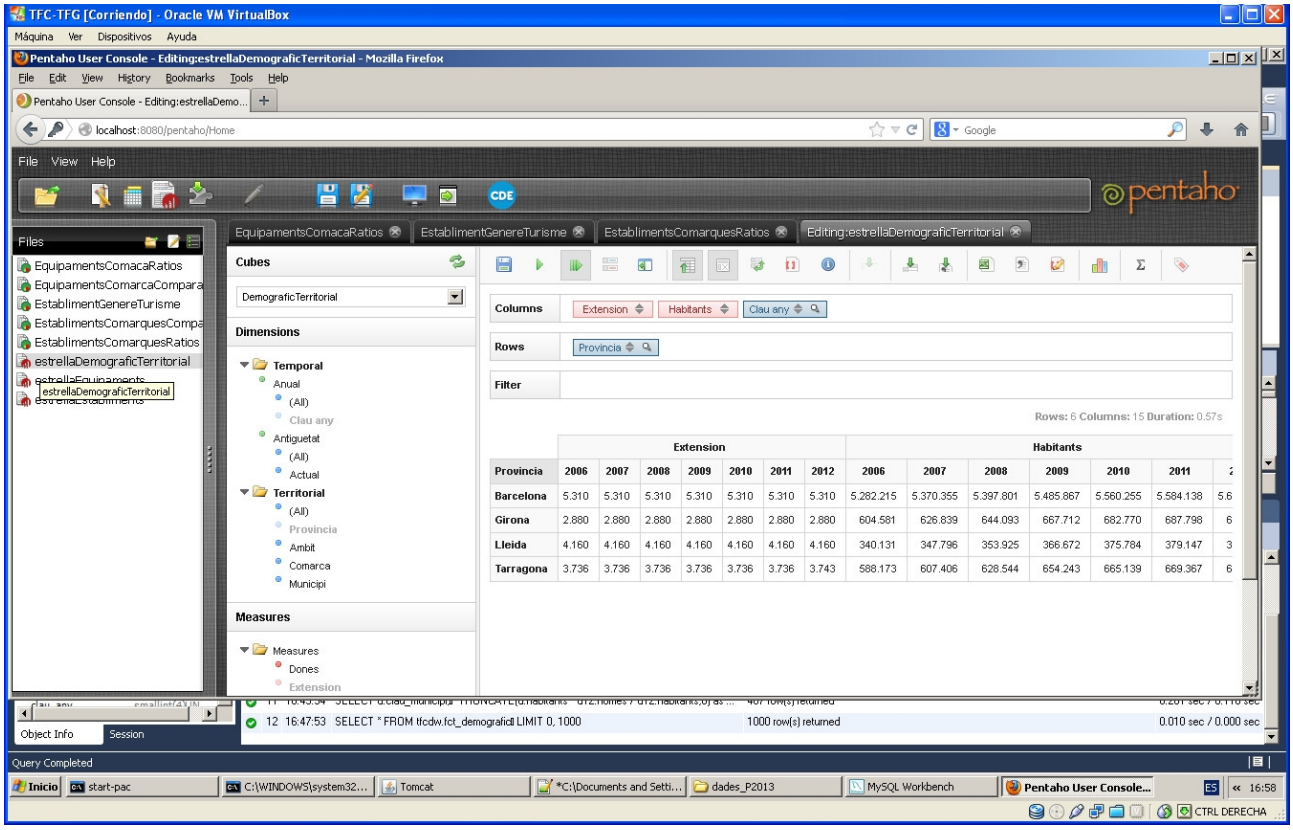


Ilustración 30: El cubo de equipamientos.

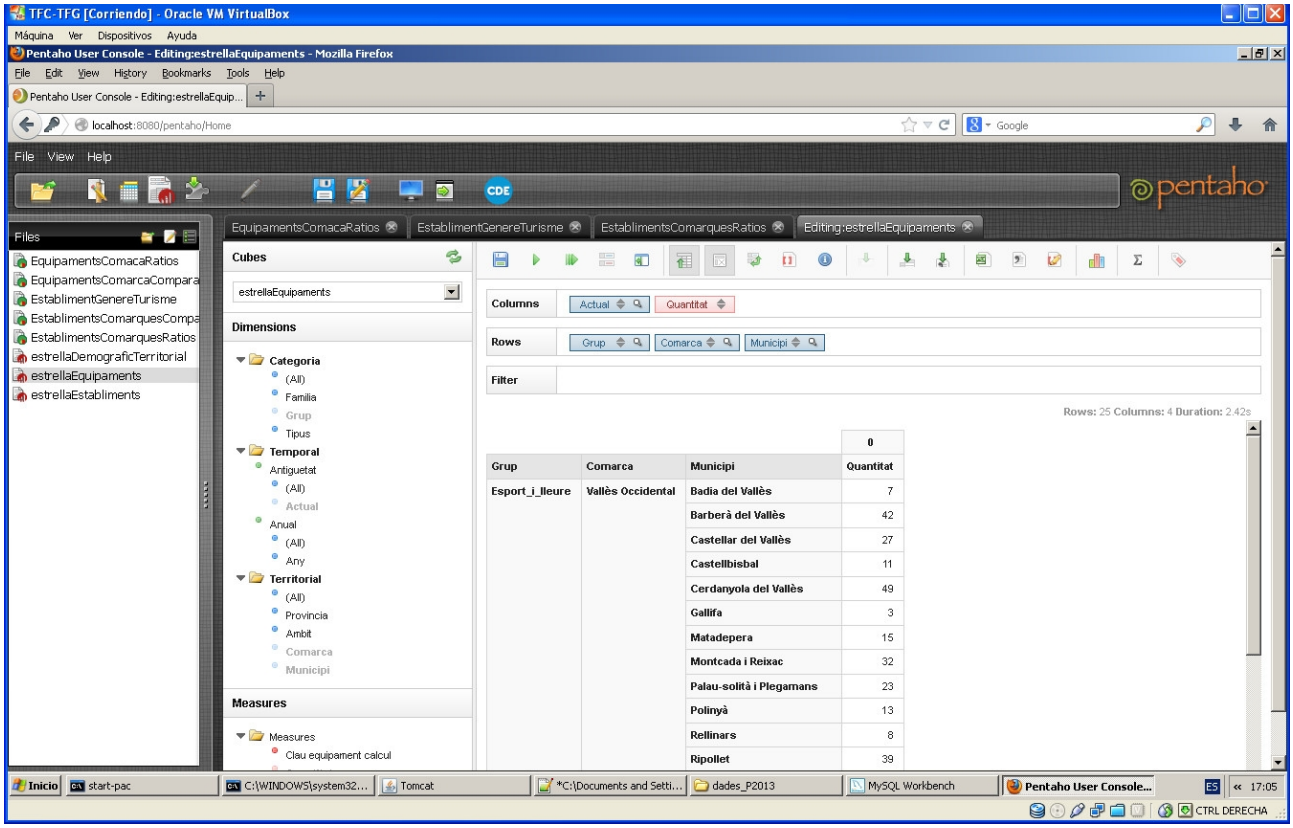


Ilustración 31: El cubo de Demográfico

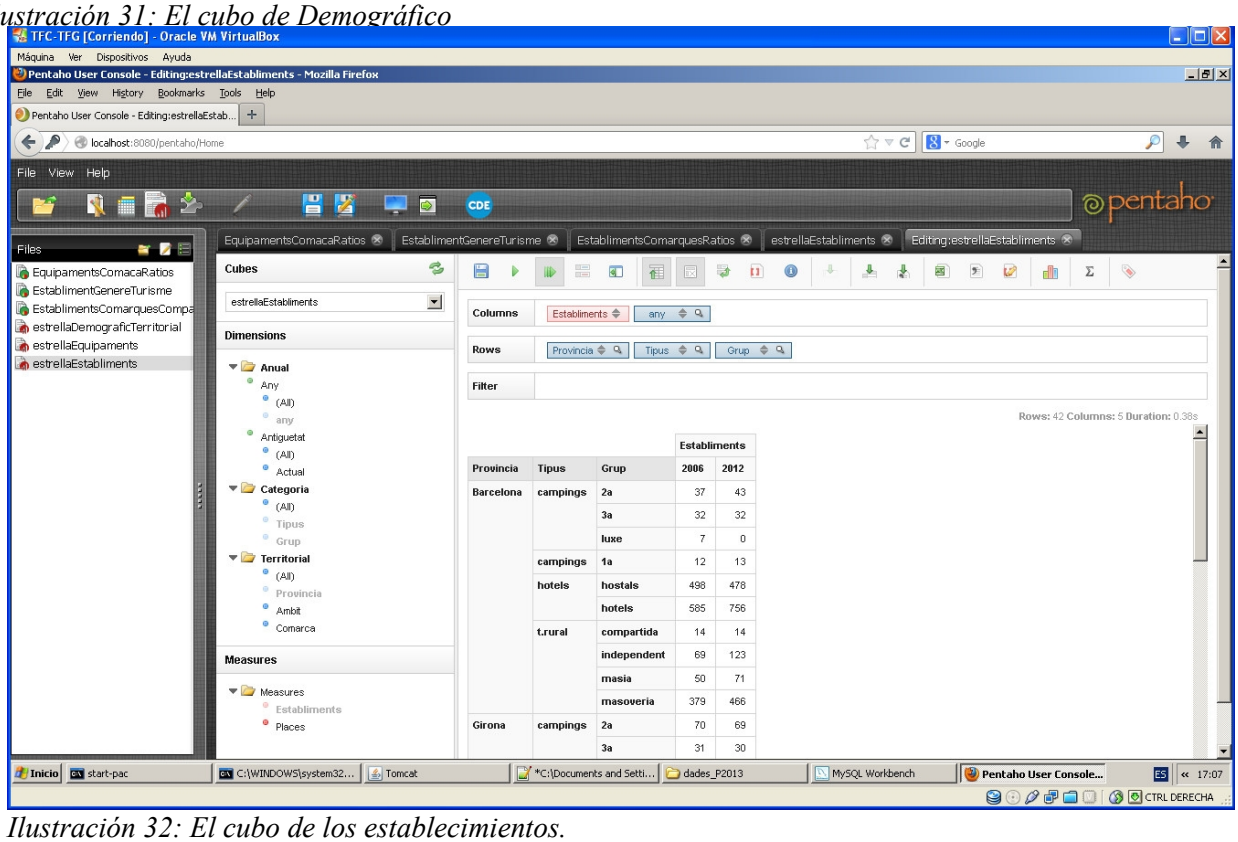


Ilustración 32: El cubo de los establecimientos.

5.2. INFORMES

Se han creando los siguientes informes, que se encuentran en el directorio *mis documentos/* a continuación se detalla que datos de los solicitados en el proyecto contiene cada uno de ellos.

Observación: no se ha podido incluirlos en la consola del **Pentaho BI Server**

5.2.1. INFORME ESTABLIMENTS COMARCA (INFORMEESTABLIMENTCOMARCA.PRPT)

Cumple con los requerimientos de tener los siguientes ratio.

- Total d'establiments
- Total de places
- Oferta mitjana de places
- % de places respecte població
- Indicador de places vs persones
- Quantitat de places ofertes / superfície del territori

Con detalle a nivel de comarca y acumulados a nivel de provincia, tipo de establecimiento (camping,hotel,t. rural) y año.

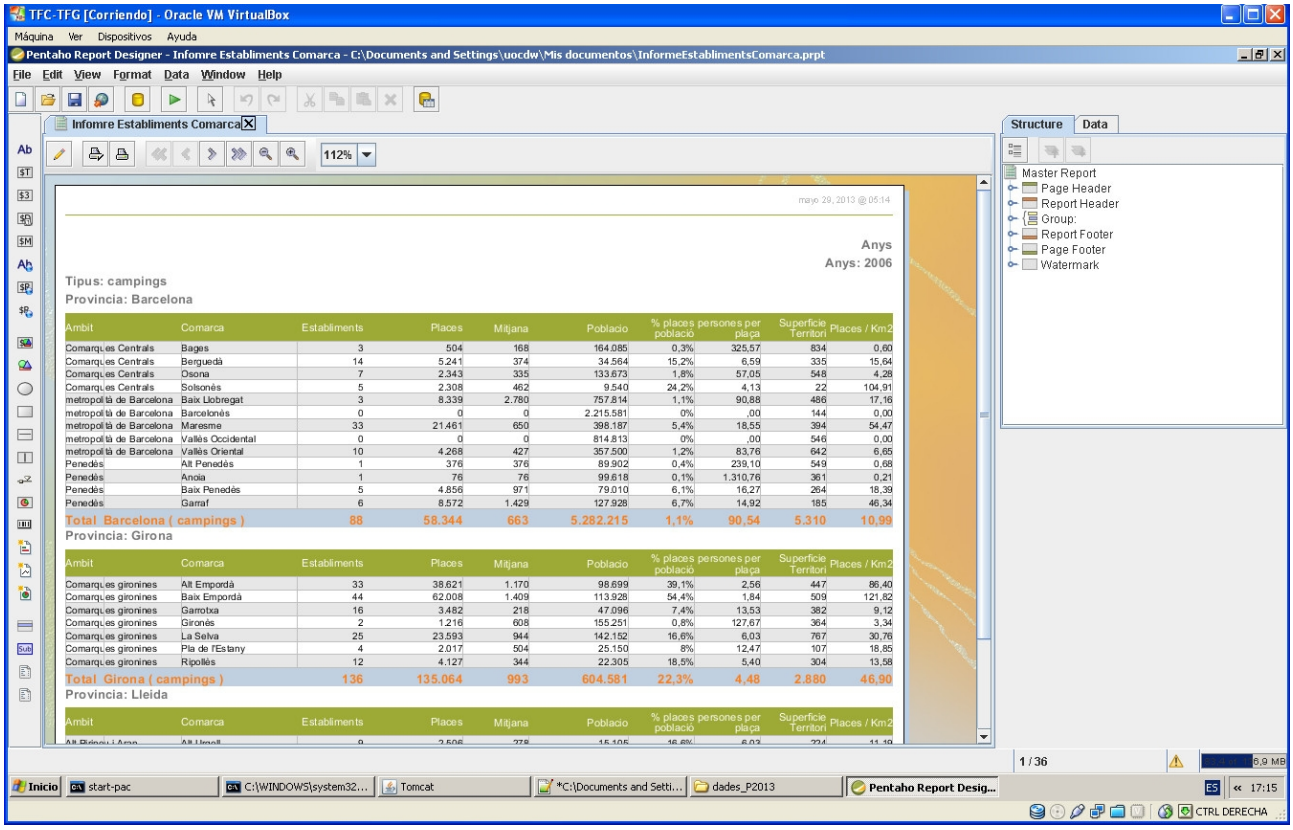


Ilustración 34: El detalle del informe establecimientos comarca

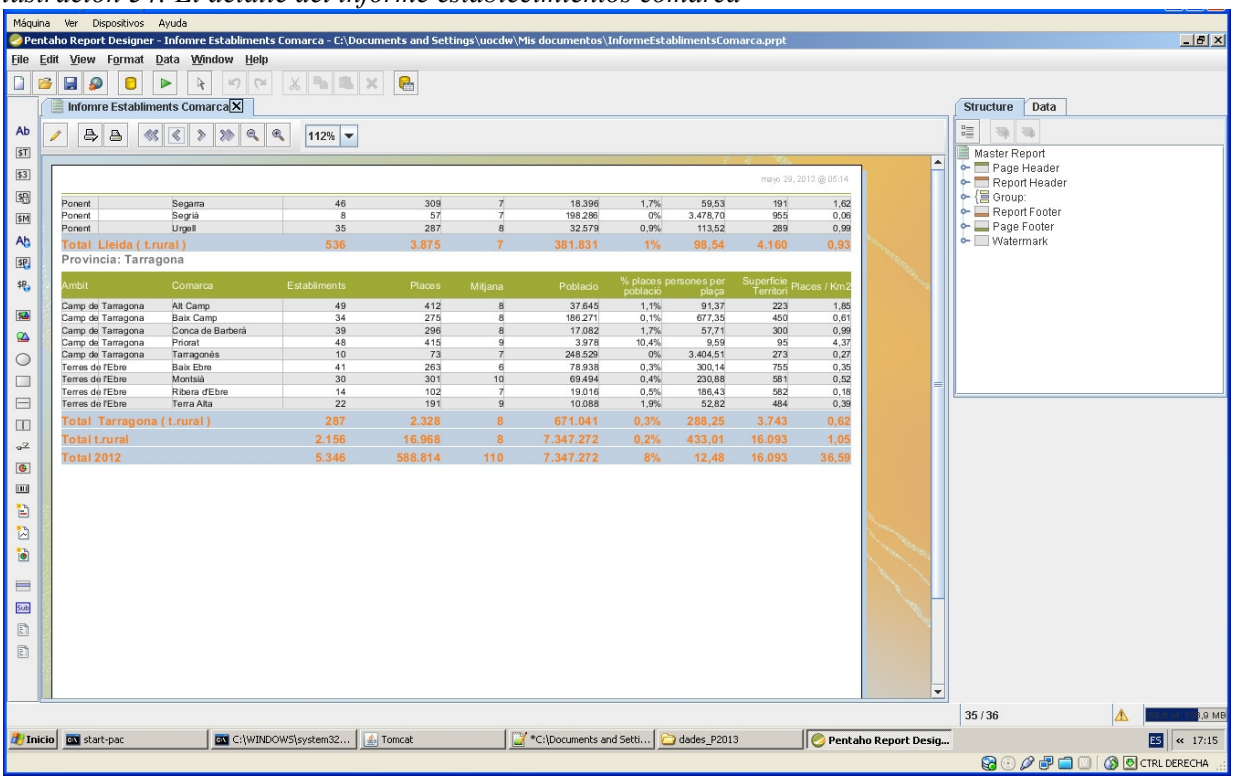


Ilustración 33: Los totales a nivel año tipo y provincia establecimientos comarca

5.2.2. *Informe Establiments Comarca Homes Dones Equipaments* (InformeEstablimentsComarcaB.prpt)

Cumple con los requerimientos de tener los siguientes ratio.

- Total d'establiments
- Total de places
- Indicador d'establiments vs habitants per gènere
- Nombre d'establiments/Nombre d'equipaments

Con detalle a nivel de comarca y acumulados a nivel de provincia, tipo de establecimiento (camping,hotel,t. rural) y año.

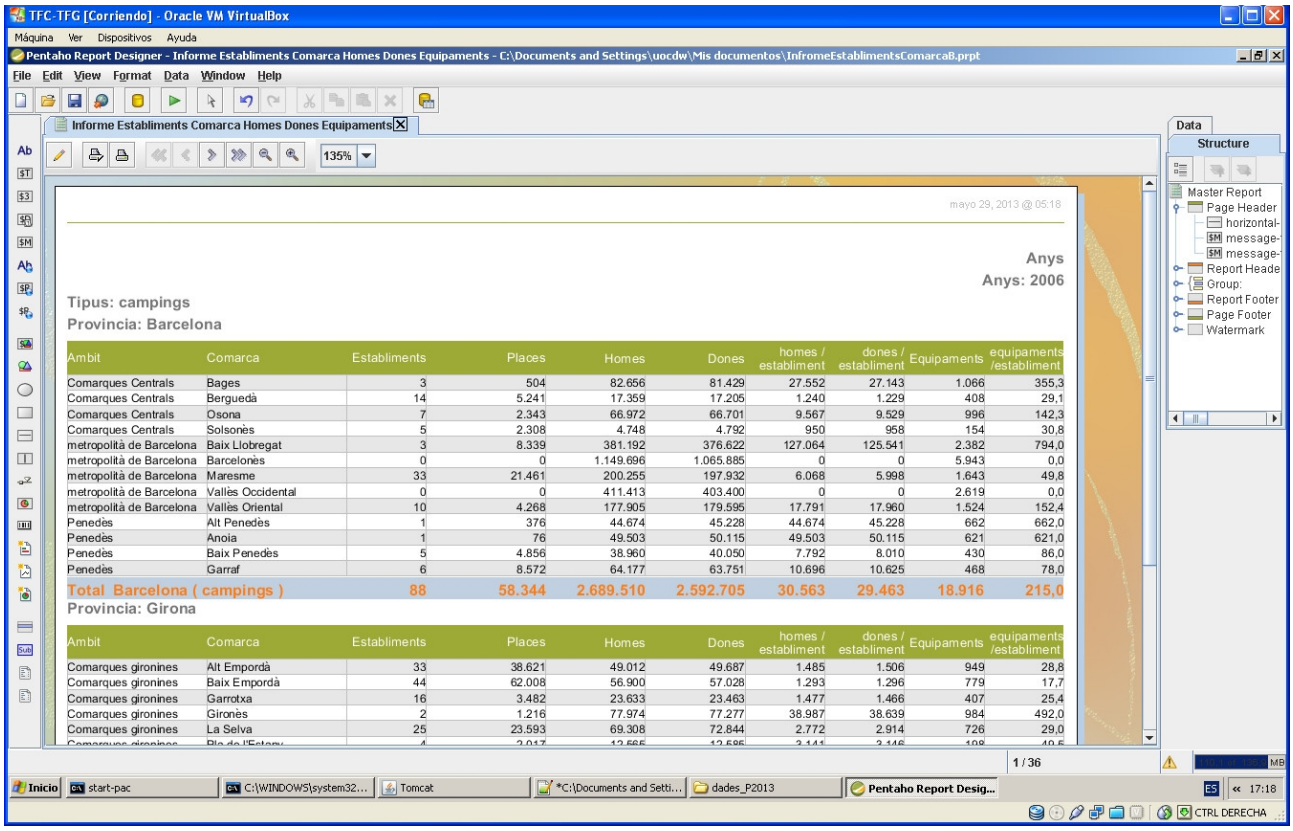


Ilustración 35: El detalle del informe Establiments Comarca Homes Dones Equipaments

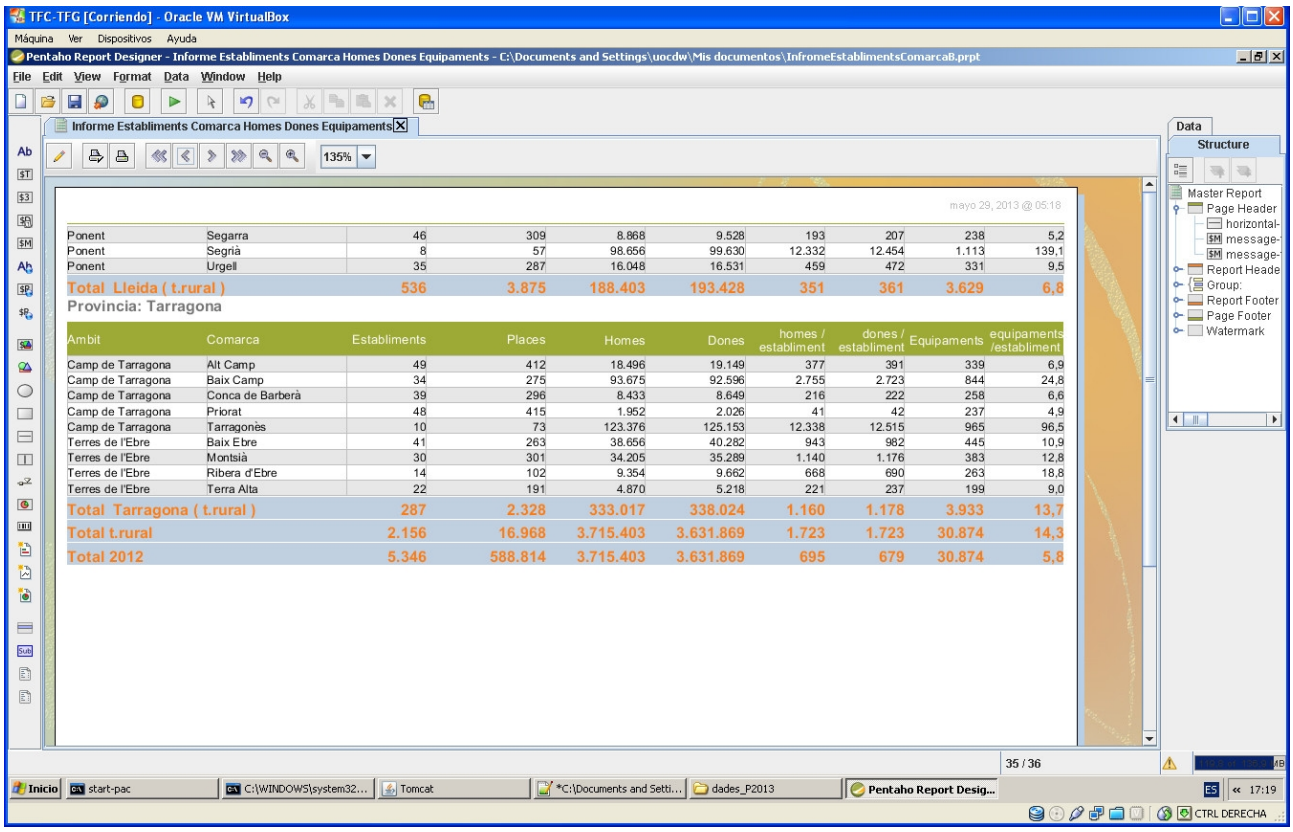


Ilustración 36: Totales provincia, categoría y año informe Establiments Comarca Homes Dones Equipaments.

5.2.3. *INFORME EQUIMAMENTS COMARCA* (INFORMEEQUIPAMENTSCOMARCA.PRPT)

Cumple con los requerimientos de tener los siguientes ratio.

- Nombre d'establiments/Nombre d'equipaments
- % de població per equipament
- Indicador d'equipaments vs població

Con detalle a nivel de comarca y acumulados a nivel de provincia, a nivel de *grupo* de equipamiento (el nivel 2) ya que todos pertenecen a la misma familia de "*equipamients*", y año

Muestra los equipamientos y su relación con habitantes y establecimientos (totalizados por comarca) y los ratios "*equipaments/ poblacio %"*, "*habitants / equipaments"* y "*establiment / equipament*".

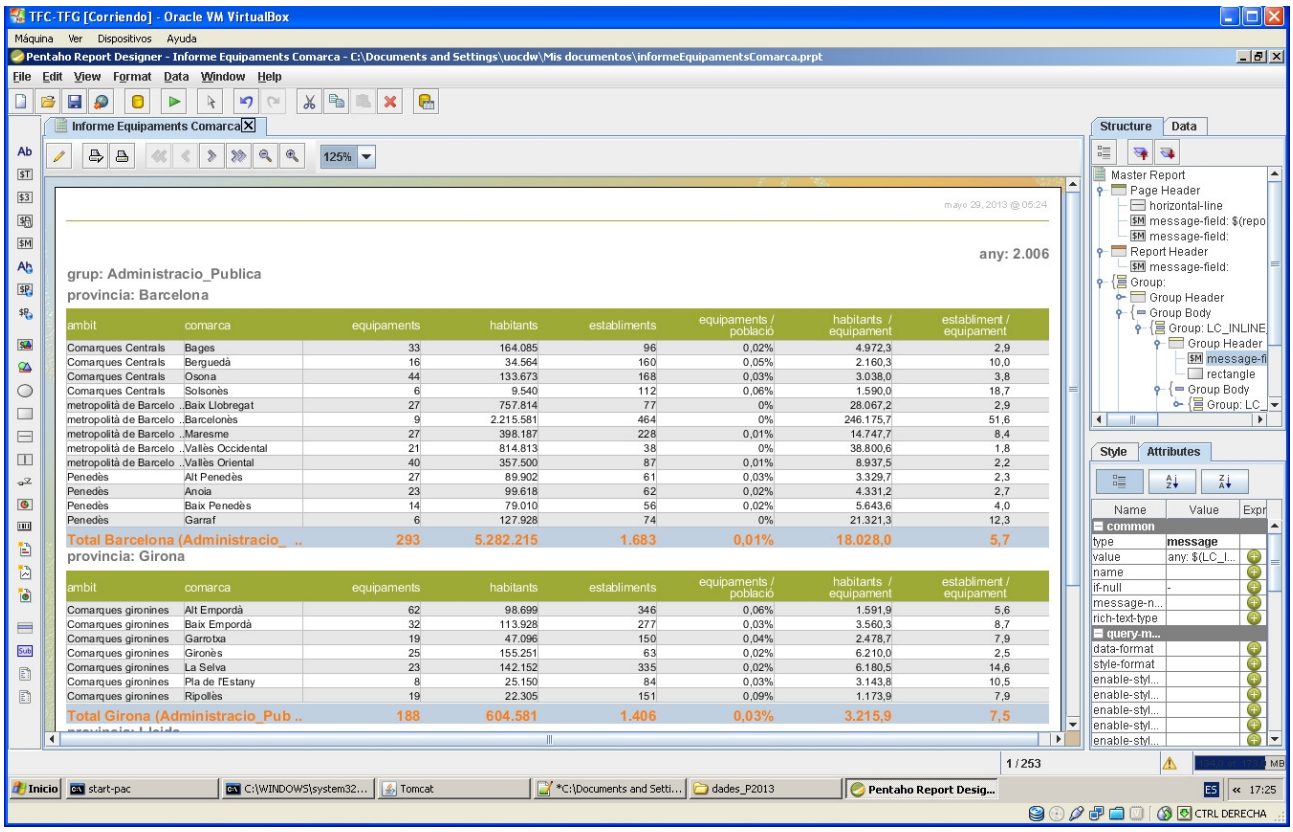


Ilustración 37: Cabecera y cuerpo del informe Equimaments Comarca

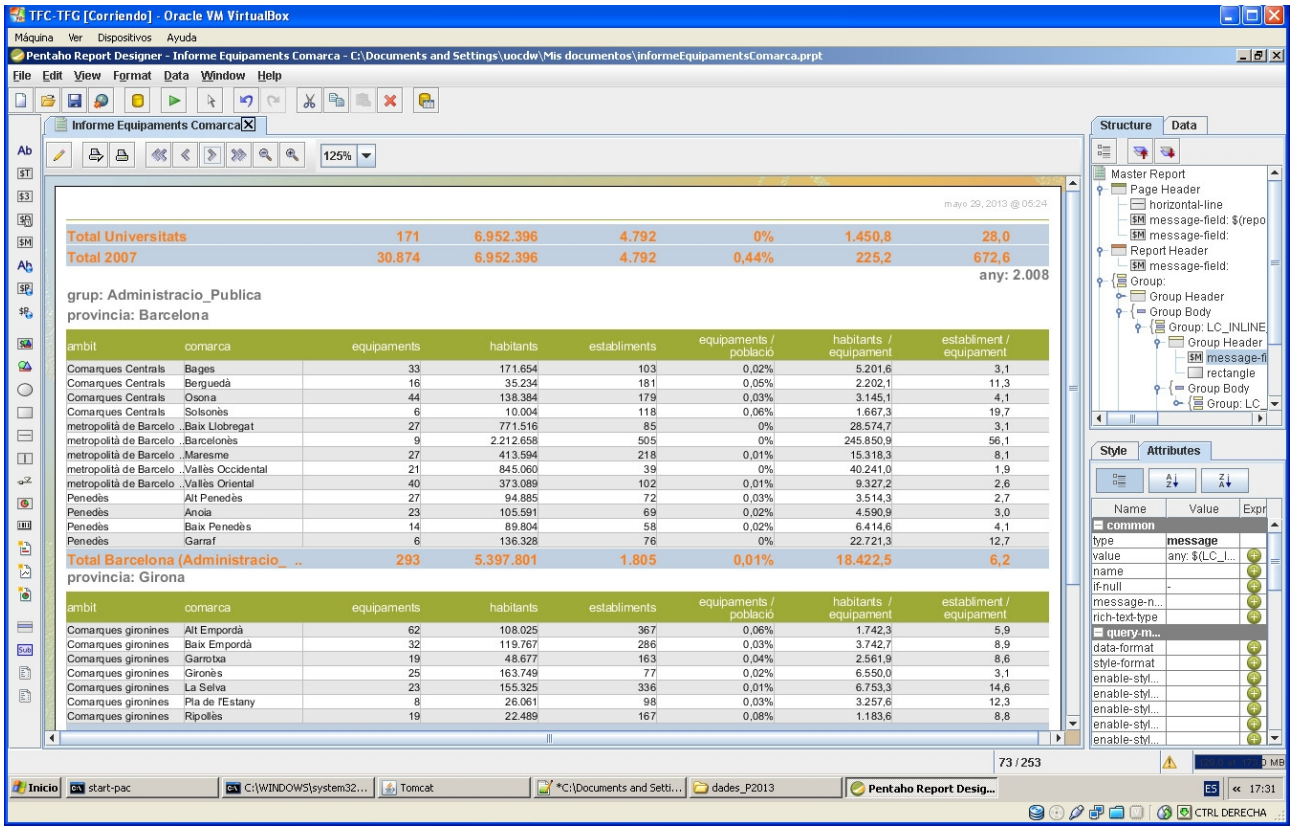


Ilustración 38: Los totales a nivel tipo y año del informe Equimaments Comarca

6. Conclusiones.

En este documento no solo se ha presentado un análisis y diseño del proyecto solicitado, sino que, también muestra como todo el proceso ha sido realizado bajo el paraguas de la diagramación **UML,** con el beneficio de tener toda la definición y diagramación integrada en un único repositorio.

A nivel personal el seguir esté método basado principalmente en [ref-LujanS] (pero tambien de [ref-prat-akoka]) me ha servido y me sirve de guía para afrontar de forma focal un parte del problema, pero a la vez tenerlo dentro de un visión conjunta. Por otra parte la lectura de [ref-Bouman] me ha expuesto las diferentes consideraciones que hay que tener al desarrollar un almacén de datos.

Respecto a la consulta de datos por OLAP al ser este un proyecto de datos cruzados entre varios tipos de estrellas, obliga a que la conexión fuera una consulta SQL y por definición de tablas, lo que se pierde parte del potencial de los esquemas, por otra parte, los datos cruzados ,que no dejan de ser hacer combinaciones entre manzanas y peras, hacen que ciertos valores calculados en el cubo sea ilógicos, debido al no poder indicarle a un valor que operación hay que utilizar cuando se agrega en cada una de las dimensiones que le afecta, el sistema solo permite elegir una que aplicará en todas las dimensiones.

7. Lineas de evolución futura.

Los informes cumplen los requisitos del proyecto y además ofrecer nuevas consultas OLAP, aunque en el enunciado no se habla de gráficos creo que haría falta diseñar uno que resumiera el valor de la información presentada destacando de una manera visual y reveladora la razón de ser de este proyecto.

Para el proceso de transformación, se han creado múltiples reglas de validación, pero no estoy satisfecho del registro de los cambios, en este caso es un proyecto de una sola carga y por lo tanto el control de modificaciones, no es importante, y no he creído necesario utilizar el sistema para *las tablas que cambian lentamente,**Slowly Changing Dimensions (SCD),* aunque si me hubiese gustado presentar una sistema de registro de altas y escrituras tal como describí en el modelo lógico del ETL, pero la falta de tiempo me lo ha impedido.

Glosario

Cubo OLAP Es una expresión que surge de ver el OLAP como un cubo de varias dimensiones en donde en cada una de las caras hay una dimensión del hecho que se analiza.

Desnormalizar, Una vez aplicados los conceptos relaciones en la tablas descerlos para conseguir una optimización de tiempo en las consultas.

Dimensión. Una entidad sobre la que se puede medir.

Estrella. Es el diseño de crear las tablas de hechos con sus dimensiones, de tal forma que la tabla de hecho solo tiene valores y referencias a las distintas dimensiones.

Esquemas. Véase metadatos.

ETL. Proceso de extracción, transformación y carga de los datos desde sus distintos formatos y fuentes hasta el almacén de datos.

Hecho. ver valor

HOLAP Modelo híbrido de almacenamiento OLAP que utiliza base de datos relacional y multidimensional

Máquina Virtual. Software que simula un equipo determinado sobre el cual se le instala un sistema operativo funcional

Mecanismo ETL, Una unidad elemental de transformación, se ha identificado los siguientes mecanismos: Agregación, conversión, filtro, incorrecto, unión (join), cargador, registrador, fusionar (merge), sustituto, envoltorio.

Metadatos. Datos sobre los datos, describen cual es la estructura de los datos que se almacenan y como se relacionan. Incluye esquemas basados en la estructura y el lenguaje del negocio.

Modelo Conceptual Resultado de aplicar el análisis centrado en buscar entender el ámbito o dominio real en el que está insertado el problema.

Modelo Entidad/Relacion. Modelo en donde se identifican la entidades o elementos que conforman las estructuras de datos y sus relaciones.

Modelo Lógico. Es el modelo conceptual transformado para adaptarlo al sistema gestor de la base de datos y teniendo en cuenta los criterios de desnormalización.

MOLAP Es un sistema OLAP que su almacenamiento se realizan en base de datos especificas conocidas como multidimensionales.

OLAP. (Procesamiento analítico en linea) es la herramienta y las técnicas que permiten realizar consultas decisorias en línea.

Paquete UML, Agrupación de varios elementos del dominio del problema o diseño de la solución, se representa como una carpeta.

ROLAP Es un sistema OLAP que su almacenamiento se realiza en una base de datos relacional

Stagging Area. Espacio de almacenamiento utilizado durante el proceso ETL.

UML Lenguaje Modelado Unificado, es una método de desarrollo de proyectos informáticos y un conjunto de técnicas de diagramación

Valor / Hecho unidad de información mensurable en varias dimensiones.

Bibliografía y Enlaces

Enlaces web

- [ref-LujanS] **Sergio Luján-Mora**. *Data Warehouse Design with UML PhD Thesis*. Departament of Software and Computing Systems University of alicante. Junio del 2005. Formato PDF. Enlace web: http://www.dlsi.ua.es/~slujan/files/thesis.pdf
- [ref-prat-akoka] **Nicolas Prat* Jacky Akoka**** *Form UML to Rolap multidimensional databases using a pivot model* Formato PDF. Enlace web: http://faculty.essec.fr/n.prat/BDA02_Prat_Akoka.pdf
- [ref-argoUML] http://argouml.tigris.org/ *ArgoUML is the leading open source UML modeling tool and includes support for all standard UML 1.4 diagrams. It runs on any Java platform*

Libros

- [ref-Bouman] **Roland Bouman Jos van Dongen** "Pentaho Solutions: Busines Intelligence and Data Warehousing with Pentaho and MySQL". Ed. WILEY.

Referencias

- [ref-hüse] **B. Hüsemann, J. Lechtenbörger and G. Vossen**. Conceptual Data Warehouse Modeling. In *Proceedings of te 2nd International Workshop on Design and Management of data Warehouses (DMDW'00),* páginas 6.1-6-11, Stockholm, Sweden, 5 Junio del 2000

Anexo: Por que **ArgoUML**?

Un primer motivo era tener un herramienta de análisis que permitiese definir todo el proyecto y así conseguir tener un repositorio único con todo el análisis/diseño del proyecto,

Un segundo motivo era que que quería evitar utilizar varios tipos de herramientas diferentes, con los previsibles problemas de sincronización y manteamiento.

Otros motivos:

- Vengo utilizando **ArgoUML** para mis proyectos profesionales y por la tanto me evita incluir en el **TFC** la planificación de tiempo que requeriría para superar la curva de aprendizaje
- Es una herramienta código abierto, y por lo tanto, colaborativo.

Anexo Modelo Conceptual basado en UML

Notas sobre la representación del modelo conceptual

Para el modelo conceptual no utilizamos el modelo ***Entidad/Relación*** **ER**, si no, aprovechamos de que de hecho el **UML** es un superconjunto de las notaciones **ER** y utilizamos los diagramas de clases con notaciones extendidas, principalmente estereotipos.

En esta ocasión aprovechamos la versatilidad de crear un perfil **UML** especifico para los almacenes de datos siguiendo la propuesta de nomenclatura de [ref-LujanS]

Para la representación del modelo conceptual hemos utilizado la propuesta de [ref-LujanS] de dividirla en niveles utilizando la agrupación por paquetes que ofrece el UML, que tiene la ventaja adicional de permitir definir diferentes dominios conceptuales. En la propuesta de [ref-LujanS] indica la utilización de 3 niveles, con la idea de dar cabida a poder diagramar estructuras complejas, y a su vez que estás se representen con el mínimo número de niveles.

En nuestro proyecto siguiendo la filosofía de mostrar la complejidad de forma entendible y con el menor número de niveles posibles y viendo que la magnitud de nuestro proyecto nos permite quedamos en dos niveles:

- Nivel 1: Definición del modelo, en donde el artefacto **UML paquete** representa el esquema estrella del modelo conceptual. Identificamos con el estereotipo <<*star*>> aquellos que contiene en su interior una estructura de estrella con sus valores y dimensiones (como mínimo a de tener una dimensión) y <<*dimension*>> para aquellos paquetes que en su interior solo contienen dimensiones. Dichas dimensiones serán compartidas por los demás paquetes estrella. Las flechas de dependencia indican que el paquete como minino contiene una dimensión que comparte la estrella
- Nivel 2: Definición de la estrella: Se representan como **clases UML** cual es el hecho ,cuales son sus dimensiones y sus correspondientes niveles de jerarquía. Para el nivel 2 utilizamos los siguientes artefactos para los siguientes conceptos.
  - Hecho o valor : clase con estereotipo <<*fact*>>.Puede tener atributos
  - Dimensión: clase con estereotipo <<*dimension*>>. No podrá tener atributos, ver más

adelante la explicación sobre el uso de base y dimensión.

- Niveles de jerarquías de agregación: clase con estereotipo <<*base*>>, puede tener atributos
- Existe otra clase del tipo hecho que tiene un etiquetado especial <<*degenerateFact*>> para indicar que realmente es una asociación Muchos a Muchos entre un hecho y una dimensión
- Clasificación de las jerarquías: utilizamos el dibujo de la agregación con el estereotipo que indica que permite la navegación <<*rolls-upto*>>
- Las relaciones entre clases pueden tener indicadores de cardinalidad.
- Estereotipo para el atributo que hace funciones de clave primaria: <<*dwkey*>>
- Estereotipo para el atributo que se utilizará como identificador para la navegación por el cubo dimensional en herramientas **OLAP** <<*descriptor*>>
- Estereotipo para el atributo que contiene un valor, <<*factAttribute*>> evidentemente se utilizará en una <<*fact*>>
- Estereotipo para indicar una dimensión que no existe como una entidad independiente, si no, que esta dentro de el hecho <<*degenerateDimension*>>.
- En el nivel conceptual para los tipos de los atributos se utilizan los que están en el perfil de predefinido en **ArgoUML**, *Integer* y *String* para diferenciar se trata de un numero o de un texto, sin entrar en mas detalle.

CRITERIOS PARA EL USO DE <<*DIMENSION*>> Y <<*BASE*>>

Para la representación de la dimensión y los niveles de jerarquía seguimos la recomendación de [ref-LujanS] de utilizar la definición de Hüsemman [ref-huse] en la que una dimensión contiene una primera jerarquía llamada *dimensión de nivel terminal*, (la representamos con el estereotipo <<*dimension*>>) desde la que se asciende "*Rolls-upTo*" hasta el nivel de "todos", que aunque esta última no se representa, implícitamente está en toda dimensión.

La *dimensión de nivel terminal s*e utiliza como clase enganche entre estrella y los niveles de jerarquías de clasificación que contiene dicha dimensión. Funciona como "un representante "de la dimensión y quienes definen los niveles de jerarquía son las clases <<*base*>> mediante sus relaciones de tipo agregación entre ellas . Con ello conseguimos 2 objetivos acercarnos al modelo de cubo y por otro facilita la representación del modelo. Para ver de una manera gráfica ir al apartado *"Nivel 2: Paquete dimensión comuns"*

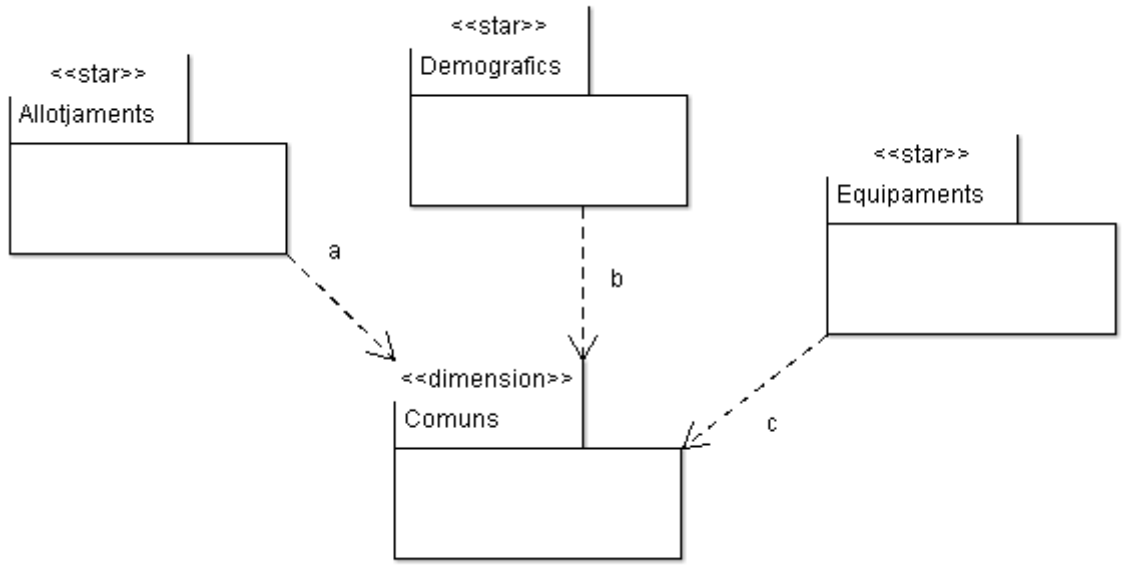
NIVEL 1: MODELO CONCEPTUAL

En el apartado de "Definición del contexto" se han identificado dos grandes grupos de consultas que ha de responder el sistema:

- Comparativas de establecimientos con zona geográficas y su población (habitantes y km^2) , en todos ellos ha de poderse navegar por año y por tipo de establecimiento y agrupación geográfica.
- El sistema a de soportar comparativas y ser capaz de establecer relaciones entre los establecimientos y los equipamientos, Además ha de poderse , navegar por el árbol de tipos de equipamientos y clases de equipamientos.

A mayores hemos propuesto diseñar el sistema para que permita

- Comparativas anuales entre equipamientos.
- Comparativas entre los equipamientos y los datos geográficos y demográficos.



Diagr1: Modelo Conceptual.

A partir de lo anterior expuesto, en nuestra representación tenemos 3 formaciones de estrellas (paquetes *<<start>>)*: *Allotjaments, Demografics, Equipaments.* Y una dimensión compartida por las tres estrellas llamada *Comuns*. Así tenemos en un primer nivel una visión de los diferentes tipos de valores que conforma el producto.

- El paquete *Allotjaments* contiene la estrella con los valores sobre el alojamiento hotelero y su dimensión de categorías de establecimientos y las dimensiones comunes: *Any* y *Comarca*
- El paquete *Demografics* contiene la estrella con los valores sobre la demografía y extensión geográfica, además de las dimensiones compartidas *Any* y *Municipi*
- El paquete *Equipaments* contiene la estrella con los valores agreagados de equipamenintos y

su dimensiones de categorías de equipamientos (*Equipament)* e instalaciones *(Instalacions),* además de las dimensiones compartidas *Any y Municipi*

Nivel 2: Paquete dimensión Comuns

Nota: El diagrama se encuentra en la siguiente página y en formato apaisado

Para la representación de las dimensiones y como ya se indica en el apartado "Criterios para el uso de <<*dimension*>> y <<*base*>>*"* definimos una que funciona como representante y después están los elementos de la jerarquía de niveles.

Dimensión *Any:*

En este caso tenemos, a la dimensión *Any* que solo tiene el nivel base de *Any.* Este es el -único caso en donde la clave no será asignado automáticamente por la base de datos, sino será el año de cuatro cifras. A su vez será utilizado como descriptor en el sistema OLAP.

Dimensión *Municipi:*

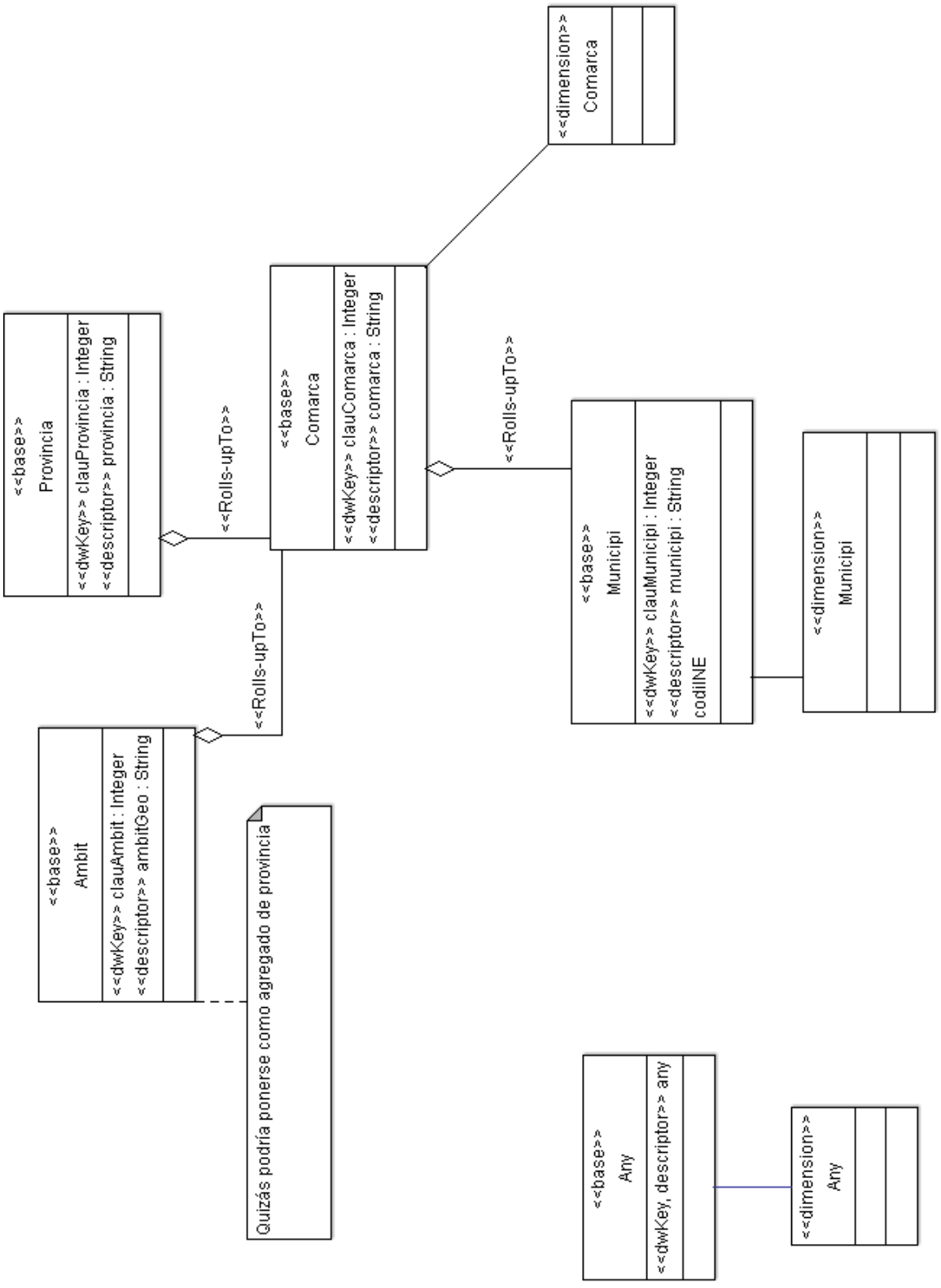
Tiene como base de más bajo nivel al municipio (*Municipi*) que después asciende por *Comarca* que a su vez se puede ascender por *Ambit* y *Provincia*.que como se ve en diagrama comparte con la dimensión *Comarca*

Dimensión Comarca

Tiene como base de más bajo nivel a la *Comarca* desde la que se puede ascender por *Ambit* o *Provincia.*

Quizás *Ambit* se podría poner como agregado de provincia, pero como *Ambit* parte de conceptos administrativos diferentes que las provincias, por el momento, se dejan como dos visiones diferentes de agrupar las comarcas.

Comarcas y Municipios: La razón de que existan dos puntos desde donde se "engancha" al árbol de clasificación administrativa, es debido a que existen, *hechos* que tienen su nivel más bajo de representación en el *Municipi* y otros en *Comarca*



Diagr 2: Comuns

Nivel 2: Paquete estrella Allotjaments

Nota: El diagrama se encuentra en la siguiente página

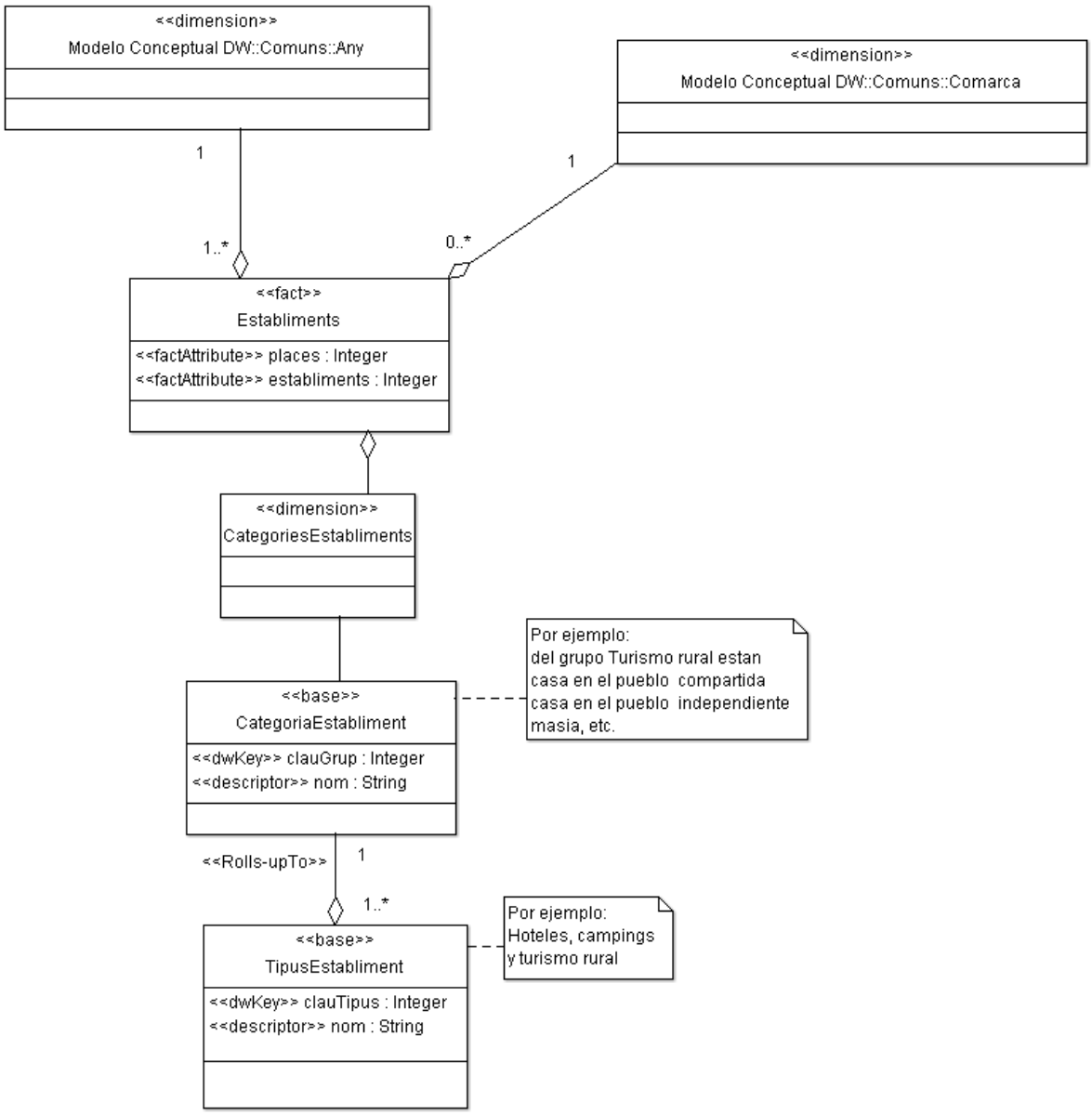
La estrella Allotjaments tiene como el hecho de que exista un establecimiento y su número de plazas. No se puede acumular dichos valores por varios años ya que no tiene sentido.

Las dimensiones tienen como jerarquia de base, el año (A*ny)* comarca (*Comarca)* y categoría de establecimiento (*CategoriaEstabliment*) que se puede ascender a los tres tipos básicos (*TipusEstabliment*)

La dimensión *CategoriesEstabliments* tiene la siguiente estructura.

TipusEstabliment	*CategoriaEstabliment*
Hotels	*Hotels (estrellas or)*
	Hostals (estrellas argent)
Campings	*Luxe*
	1
	2
	3
Turisme Rural	*Casa de poble compartida.*
	Casa de poble independent
	Masia
	Masoveria

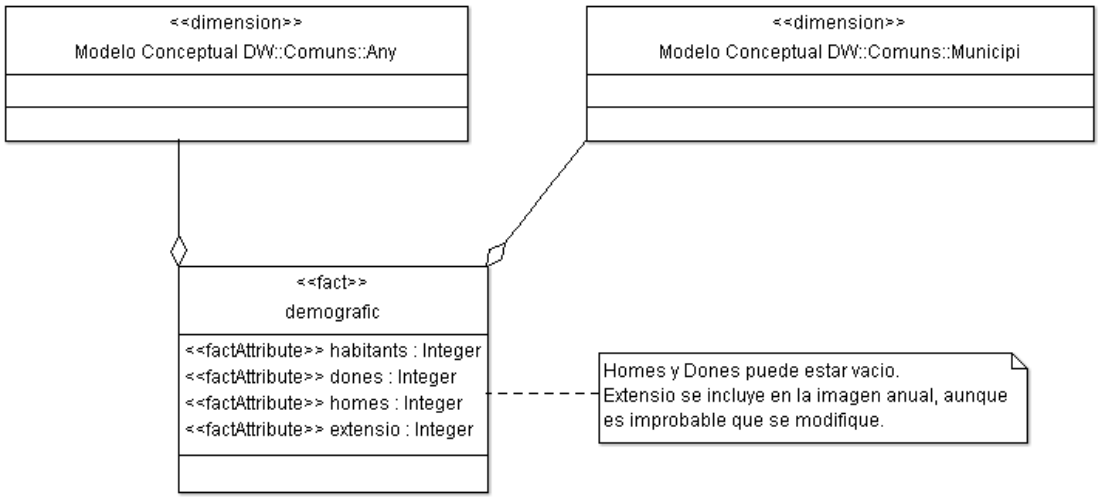
•



Diag3: Paquete estrella *Allotjaments*

NIVEL 2: PAQUETE ESTRELLA DEMOGRÁFICS

La estrella *Demográfics* tiene como el hecho que en una determina zona geográfica viva un individuo, que este sea hombre o mujer y, por otra parte, el dato geográfico de los Km2 cuadrados. En el caso de hombre o mujer las fuentes de datos solo suministran información del año 2012, pero el sistema está preparado para insertarla en otros años. No se puede acumular los valores geográficas por varios años ya que no tiene sentido.



Diagr4 Paquete estrella *Demográfics*

Nivel 2: Paquete estrella *Equipaments*

Nota: El diagrama se encuentra en la siguiente página

La estrella *Equipaments* tiene como hecho la existencia de un determinado tipo de equipamiento y de el tipo de instalaciones que posee. No se puede acumular los valores los equipamientos por varios años ya que no tiene sentido.

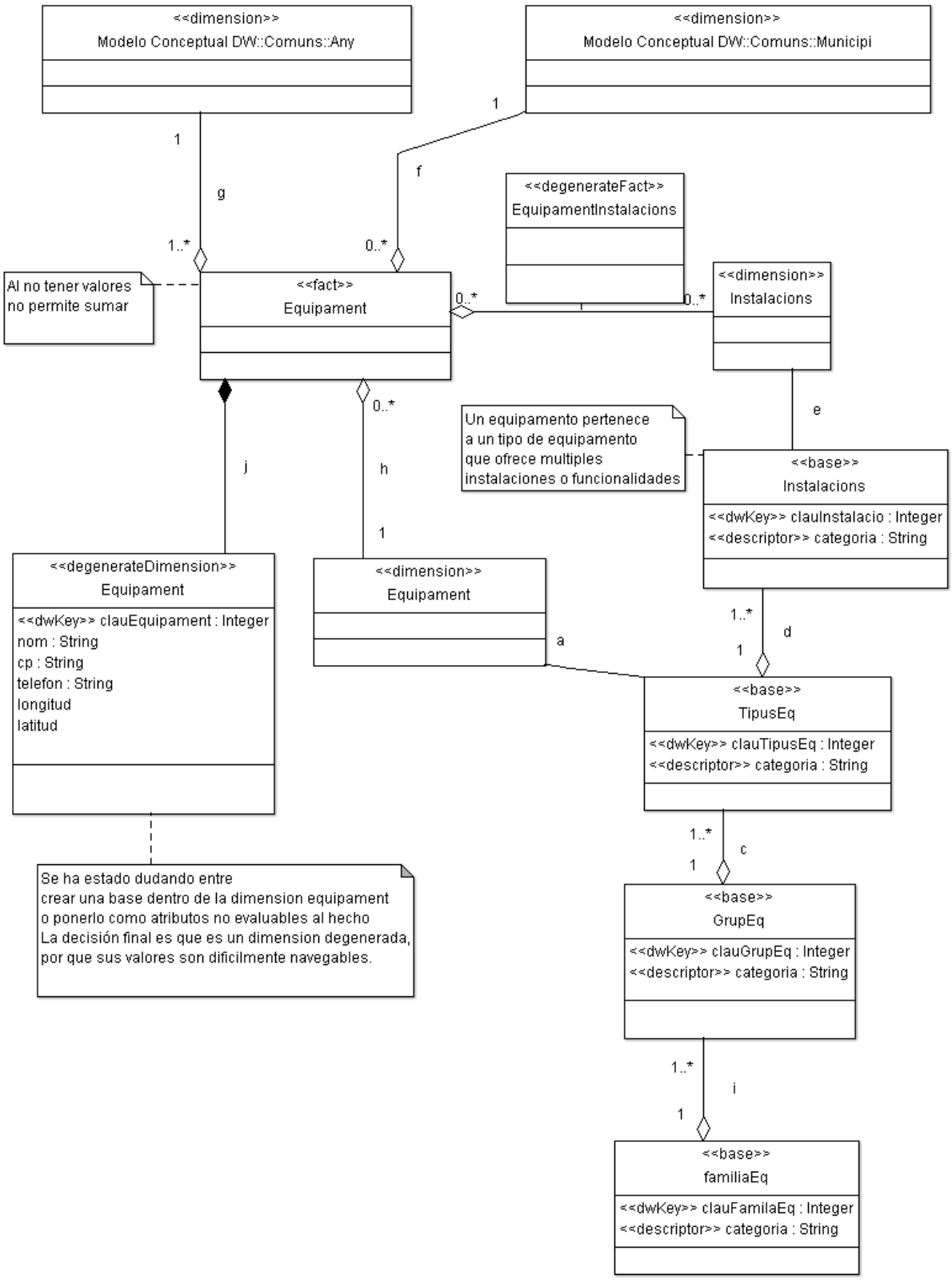
En las fuentes de datos la información esta reducida a un solo año, pero el sistema se a diseñado para que permita la entrada de otros años.

El análisis de las fuentes de datos nos da que la estructura de clasificación de los equipamientos esta formada por 3 niveles jerárquicos que los hemos identificado como *FamiliaEq*, *GrupEq* y *TipusEq*.

A mayores tras estos 3 niveles y en función de *TipusEq* puede el equipamiento contener unas instalaciones para realizar una serie de actividades. El concepto de instalaciones puede hacer referencia tanto a los espacios como a los servicios. Un ejemplo del segundo caso son los centros educativos que pueden dar unos o otros niveles de enseñanza.

En este diagrama existen varias decisiones sobre la definición del dominio a destacar:

- Como se analiza el hecho de que exista un equipamiento de un tipo y con unas instalaciones sea un valor, la información que acompaña al equipamiento se puede decir que es adjunta al hecho,y por lo tanto, se entiende que no se utilizará para navegar por el cubo multimensional así que se ha agrupado en una dimensión del tipo degenerada
- Como la lista de instalaciones (*Instalacions*) que tiene el hecho equipamiento (*Equipament)* es múltiple y un instalacion puede estar en múltiples equipamientos existe el patrón de que un hecho *Equipament* con respecto a una dimensión *Instalacions* tiene una relación N a M y por lo tanto hay que indicar que es una hecho degenerado.



Diagr5: Paquete estrella *Equipaments*

Anexo Modelo Lógico basado en UML

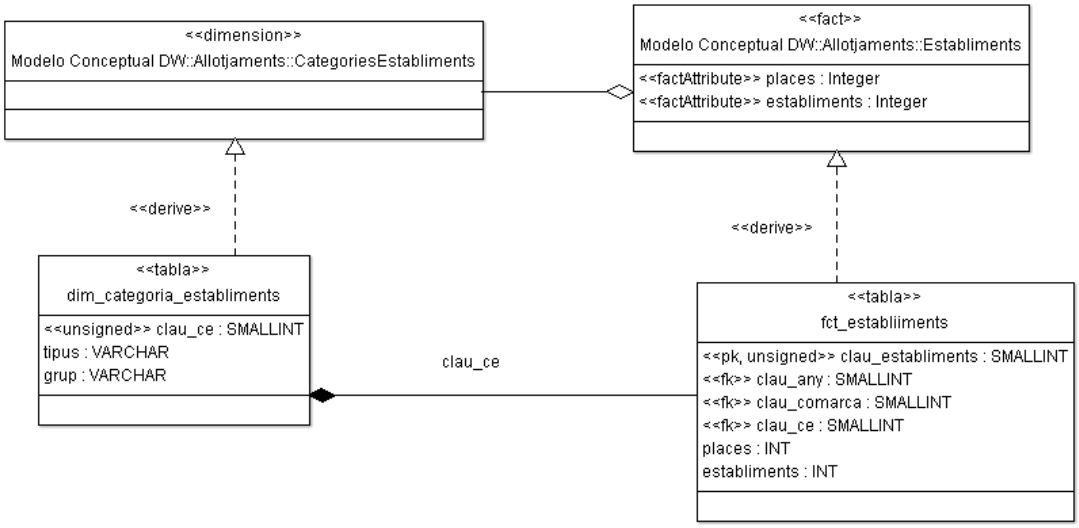
Notas sobre la representación del modelo lógico

Para el modelo lógico también se ha creado un perfil especifico para poder representar el modelo de base de datos. Así tenemos:

- <<*derive*>> Se utiliza conjuntamente con la flecha de tipo realización (en **OO UML** es utilizada para marcar la relación entre una interfaces y su implementación) en esté caso es para indicar la relación entre el elemento del modelo conceptual y el modelo lógico. Apunta a que dimensión o hecho se deriva la tabla,
- <<*tabla*>> a nivel de clase para indicar que es un tabla.
- <<*fk*>> a nivel atributo indica que es un clave foranea.
- <<*pk*>> a nivel atributo indica que es un clave primaria
- <<*unsigne*>> hace referencia a que es tipo numérico sin signo.
- Están definidos los tipos de atributos mas comunes en base de datos y utilizados por **MySQL**: BIGINT, BINARY, BLOB, BOOL, CHAR, DATE, DATETIME, DECIMAL, DOUBLE, ENUM, FLOAT, INT, MEDIUINT, SMALLINT, TEXT, TIME, TIMESTAMP, VARBYNARY, VARCHAR, YEAR.

Expongo en la página siguiente una parte del diagrama del modelo lógico para *Allotjaments* para destacar dos tipos de conexiones que pueden parecer contradictorias entre le modelo conceptual y el modelo lógico.

- En el modelo conceptual las dimensiones son agregados de los hechos.
- En el modelo lógico las tablas de hecho son composición de las dimensiones.



Diagr6 Contradicción en las relaciones.

Esto es resultado de la diferente visión que se aplica en ambos casos, en el modelo conceptual se muestra la centralidad del diseño estrella y por lo tanto, las dimensiones son componentes del hecho o valor, pero en el diseño lógico, se muestra que no puede existir un valor si no existen las correspondientes tablas de dimensiones. Así lo marcan las claves foráneas y queda reforzado por la relación de tipo composición

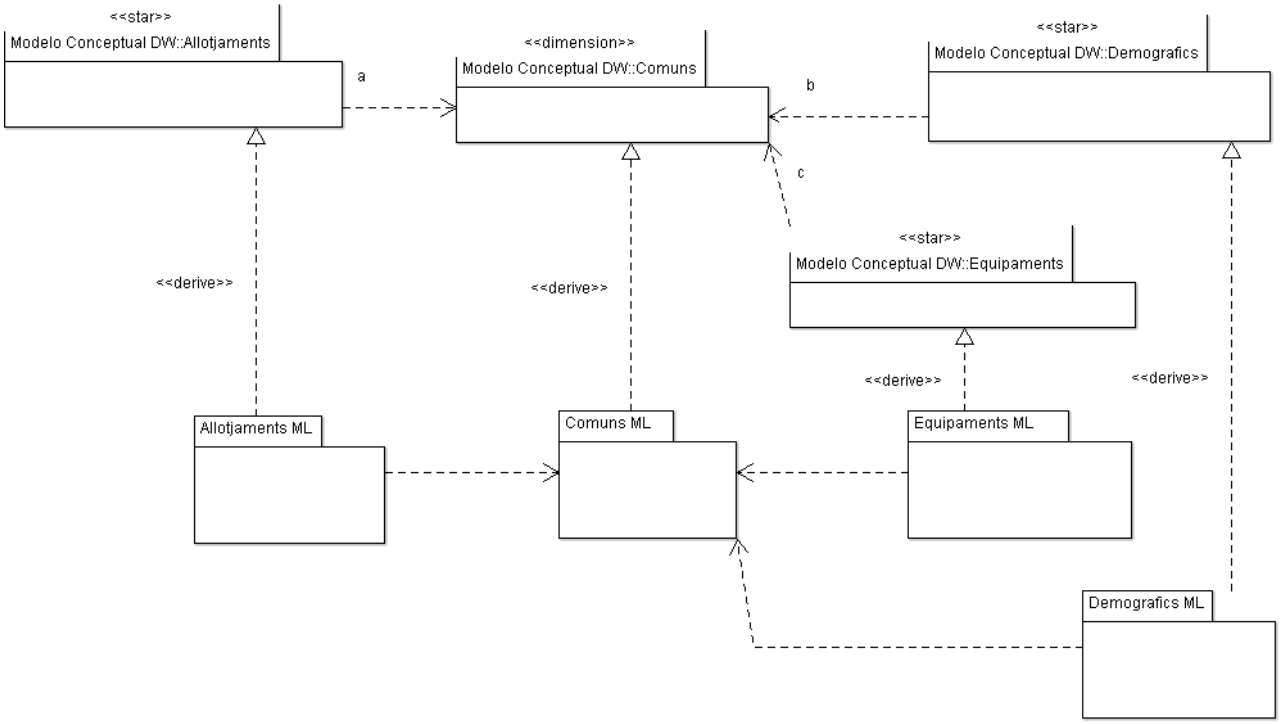
NOMENCLATURA DE LOS ELEMENTOS

Definimos en el modelo lógico los nombres finales que poseerán los diversas tablas y campos, por lo que se han establecido una serie de convenciones, siguiendo un versión libre del modelo propuesto en [ref-Bouman] .

- prefijo **dim** indica que la tabla es una dimensión.
- prefijo **fact** indica que la tabla es un hecho o valor
- prefijo **agg** indica que es tabla con valores agregados de otras tablas.
- perfijo **deg** indica que es un tabla resultado de una "*dimensión degenerada*"
- En atributos:
- prefijo **clau** indica que es una columna que forma parte de la clave.

MODELO LÓGICO NIVEL 1: VISIÓN GENERAL.

En el diagrama de la visión general del modelo lógico del almacén de datos, nos permite ver la derivación (relación) que existe con el modelo conceptual. Para buscar ser homogéneos se han aplicado los mismos criterios en la división en paquetes. Con ello, además, facilitamos el mostrar la relación entre los modelos.



diagr7: Modelo Lógico Multidimensional

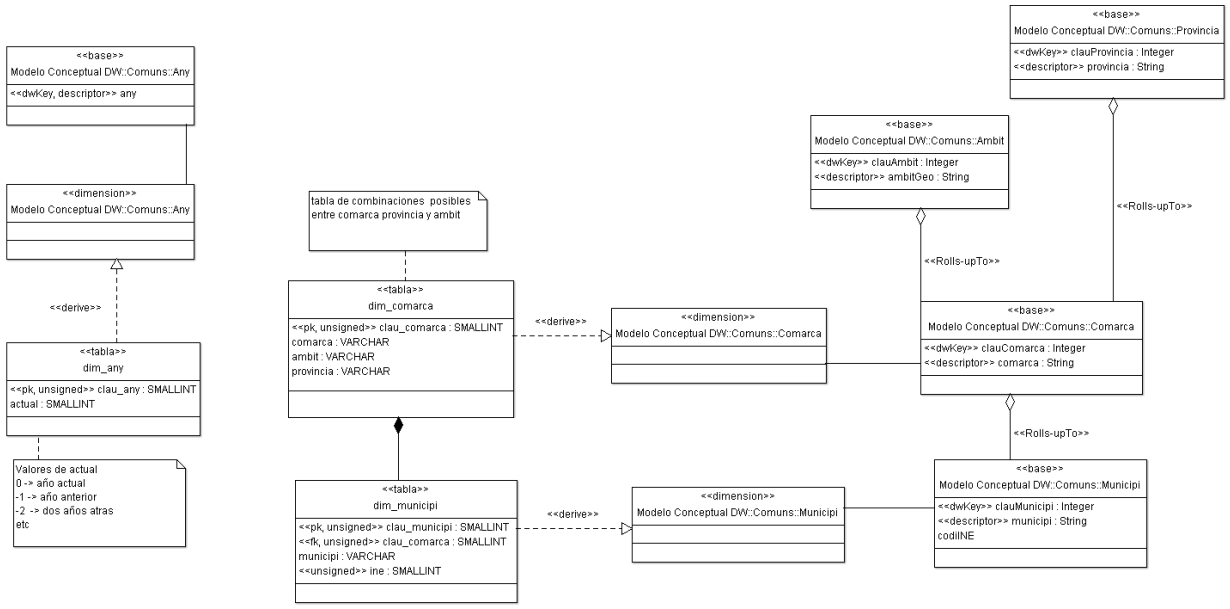
NIVEL 2: PAQUETE LÓGICO COMÚNS.

Nota: El diagrama se encuentra en la siguiente página y en apaisado.

En la diseño de la tabla del año (*dim_any)* se ha incluido el nuevo atributo *actual* que sirve para indicar cual es el año actual y cuales son los años anteriores, esto, por ejemplo, aporta rapidez en las selecciones del tipo comparativa entre este año y los 3 anteriores.

En el diseño de la tabla de comarca (*dim_comarca*) se ha decido desnormalizar y agrupar todos los elementos en un sola tabla y que en ella estén todas las combinaciones posibles. En esté caso al ser jerárquico solo hay una combinación por *comarca*, *ambit* y *provincia*.

Las claves de *dim_comaca* y *dim_muicipi* serán autogeneradas por el sistema gestor de bases de datos.

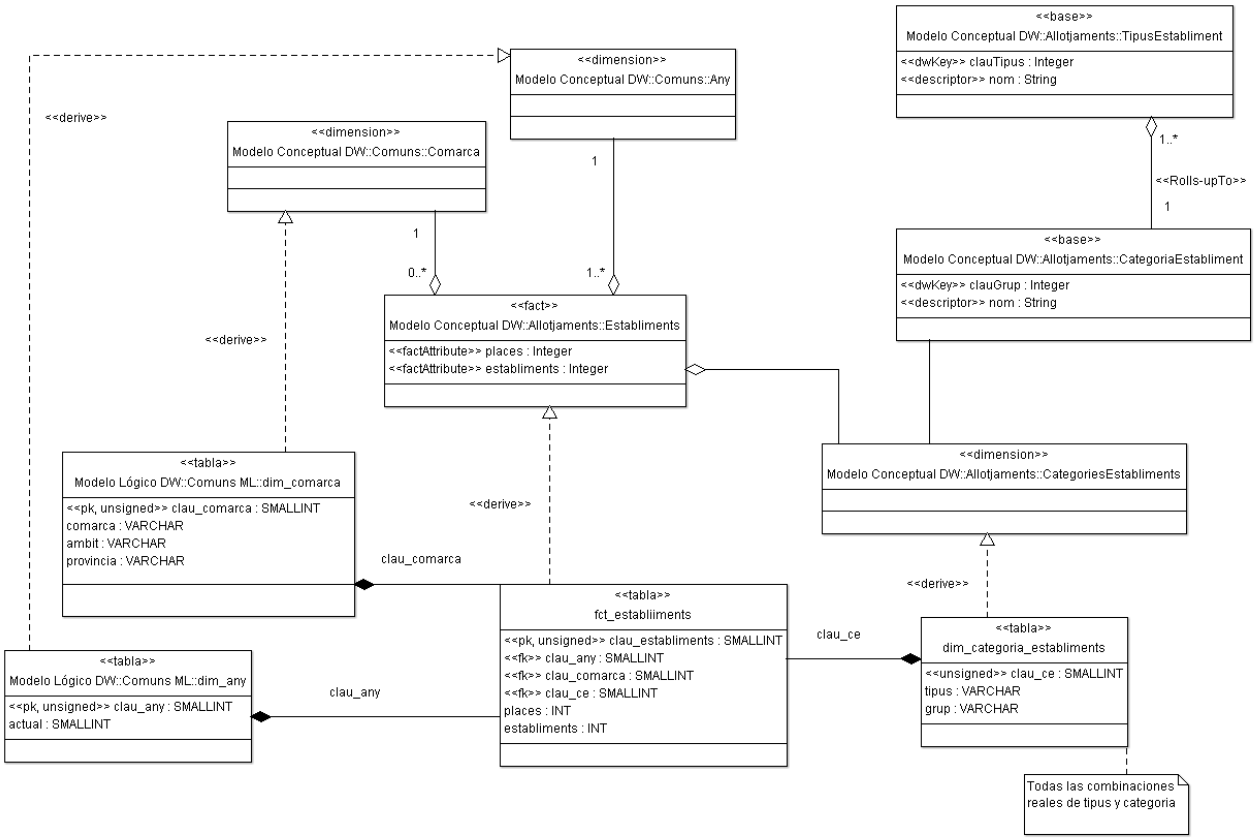


diagr8: Paquete Lógico común

NIVEL 2: PAQUETE LÓGICO ALLOTJAMNETS.

En el diseño de la tabla de categorias (*dim_categoria_establiments*) se ha decido desnormalizar y agrupar todos los elementos en un sola tabla y que en ella estén todas las combinaciones posibles. En esté caso al ser jerárquico solo hay una combinación por *tipus y grup*.

Todas las claves primarias de las tablas sobre alojamientos serán generadas por el sistema gestor de base de datos.

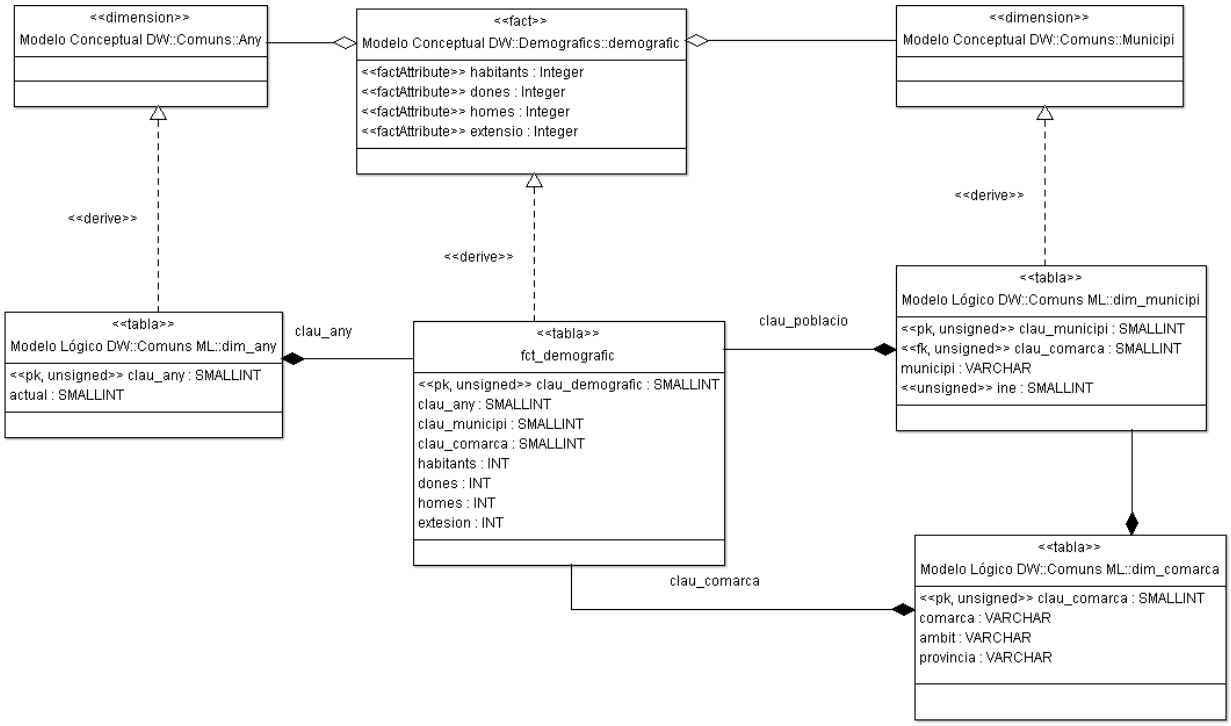


Diagr9

NIVEL 2: PAQUETE LÓGICO DEMOGRAFICS.

La clave primarias *fct_demografic* será generada por el sistema gestor de base de datos.

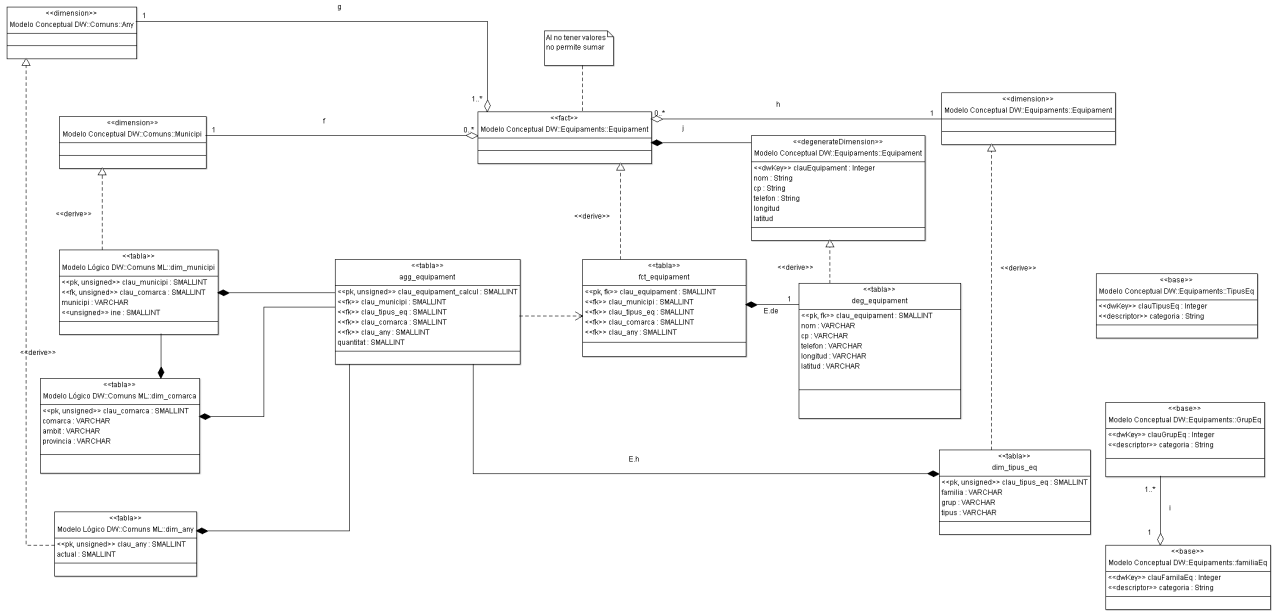
Los atributos Dones y homes pueden tener valores nulos. Es importante entonces que los informes sepan si se va consultar ese dato en un año que no tenga dichos valores.



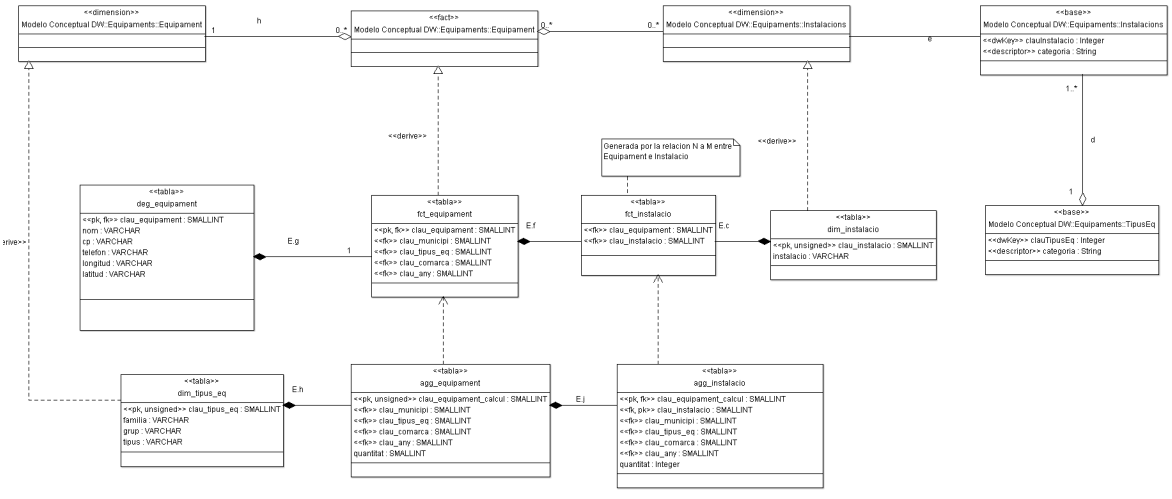
Diagr10

NIVEL 2: *PAQUETE* LÓGICO *EQUIPAMENTS.*

Como lo que es de información relevante es el número de establecimientos de un tipo y sus instalaciones, se han definido tablas agregadas que contienen dicho valor ya totalizado al nivel más bajo de la jerarquía.



Diagr11



Diagr12

Anexo Modelo Lógico ETL basado en UML

Para definir el proceso de extracción, transformación y carga de las fuentes al almacén de datos se ha creado un modelo lógico, en el que mediante una representación por niveles se han detallado los diferentes elementos que intervienen en dicho proceso. Dicho modelo se ha basado en las propuestas de [ref-LujanS] más las recomendación para un buen proceso de **ETL** de [ref-Bouman]

Modelo Lógico del proceso ETL

Se ha decido crear la definición del proceso **ETL** utilizando el modelado **UML,** con ello se consigue que en un único repositorio de información estén tanto la información sobre el almacén de datos como los procesos de carga y lo que es más importante como se relacionan entre ellos.

Otro aspecto no menos importante es que se define y se tipifican todos los elementos implicados en un **ETL**, con lo cual, se consigue por un lado normalizar la nomenclatura utilizada y por otra facilitar el análisis y diseño ya que esté se centra en ver como se relacionan y se utilizan dichos componentes para que una fuente de datos concreta se traslade al almacén de datos definido.

El paradigma que persigue el modelo se puede resumir en los siguientes puntos (basados en [ref-LujanS] y [ref-Bouman])

- El proceso será descompuesto en unidades elementales llamados **mecanismos ETL**
- Los datos de cualquier fuente que no estén en tablas será previamente traspasado a una estructura de base de datos relacional.
- Existirá una zona en la que se almacenarán todas las tablas necesarias en el proceso de transformación,
- El proceso de transformación, será auditado y en el caso de errores en los datos o en su formato el flujo se detiene en el estado en que se ha producido el error.
- Para los fuentes de orígenes no relacionales se realizara un primer paso que será tener dichas fuente en un formato relacional y sin ningún tipo de manipulación. Es importante que los datos y su estructura no sean manipulados, eso se deja a los **mecanismos ETL** de transformación.
- Las tablas intermedias generadas durante el proceso de transformación se inicializarán en cada proceso de importación.

A continuación desglosamos las unidades elementales llamadas **mecanismos ETL** definidos por [ref-LujanS],

- **Agregación (*Aggregation*)**,como su nombre indica se refiere cuando se totalizan o calculan datos basados en algún criterio
- **Conversión (*Conversion*)**: Cambia los tipos de datos y los formatea o genera nuevos datos a partir de datos existentes.
- **Filtro (*Filter)***: filtra y verifica los datos.
- **Incorrecto (*Incorrect*)**: Redirige (almacena) datos incorrectos.
- **Union (*Join*)** Se unen dos o mas fuentes en una nueva que sea la combinación de algunos de sus atributos.

- **Cargador (*loader*)** Realiza una carga de datos al almacén de datos.
- **Registrador (*log*)** Registra la actividad de los demás mecanismos.
- **Fusionar (*Merge*)** Integra dos o mas fuentes de datos con atributos compatibles
- **Sustituto (*Surrogate*)** Genera una clave primaria sustituta a la que sería la clave natural que define a la entidad. Por ejemplo agrupar los atributos en una clave natural en un única clave en formato numérico.
- **Envoltorio (*wrapper*)** Extrae los datos una fuente de datos externa para insertarlo en los registro de la base de datos del proceso de **ETL**

Puede pensarse que los mecanismos **ETL** son procesos y no estructuras de datos, está sería un visión limitada, realmente hay que verlo como estructuras de información que han sufrido una manipulación o selección, es decir, reflejan un estado en el proceso de transformación. Por lo tanto, con toda probabilidad, en la implementación tendrá una tabla que contendrá los datos en dicho estado, es decir, tras aplicar las acciones del mecanismo **ETL**. El único que escapa a dicha visión podría ser el mecanismo cargador.

Notas sobre la representación del modelo ETL

Para la representación se utiliza el diagrama de clases, en donde los mecanismos ETL son clases identificadas con estereotipos. Las operaciones que se realizan sobre los mecanismos ETL estarán comentadas como notas a la entidad. Se utilizará las conexiones del modelo de clases para representar las relaciones entre los diferentes mecanismos.

Para el modelo **ETL** también se ha creado un perfil especifico con todos los estereotipos que intervienen en el proceso **ETL**: *<<etl.agregation>>, <<etl.convesion>>, <<etl.filter>>, <<etl.incorrect>>, <<etl.join>>, <<etl.loader>>, <<etl.log>>, <<etl.merge>>, <<etl.surrogate>>, << etl.wrapper>>*

Se ha añadido: otros indicativos para diferenciar fuentes de datos: *<<etl.csv>>, <<etl.txt>>* y otro para indicar si es importante remarcar que es un proceso manual *<<etl.manual>>*

Se utilizaran las flechas de dependencia para indicar de que otros mecanismos y tablas depende un mecanismo o a que tabla del modelo multidimensional van a actualizar.

Criterios generales para los procesos ETL

Hay que ir con cuidado con la importación ya que las fuentes proveen valores durante de varios años, y en algunas de estos años los nombres identificativos varían, así como las categorías que incluyen.

Como criterio de diseño se ha establecido que siempre que sea posible utilizar la definición de nombres y las clasificaciones que viene con las fuentes del año 2012.

Todo el proceso de ETL se desarrollará en una zona diferente al almacén de datos.

Los fuentes que vengan exportados de otros sistemas, en la primera fase será importarlos sin cambios en una tabla mimética a la fuente.. Las transformaciones posteriores se realizan en nuevas tablas, dejando la original (la tabla mimética) sin cambios.

Para llevar una auditoria de cuando se produce una ETL en una tabla se ha centralizado en un

tabla *aud_tables* en donde se indicara la fecha, el número de versión y si es una inserción, una modificación o un borrado (lógico ya que no se producen borrados de registros)

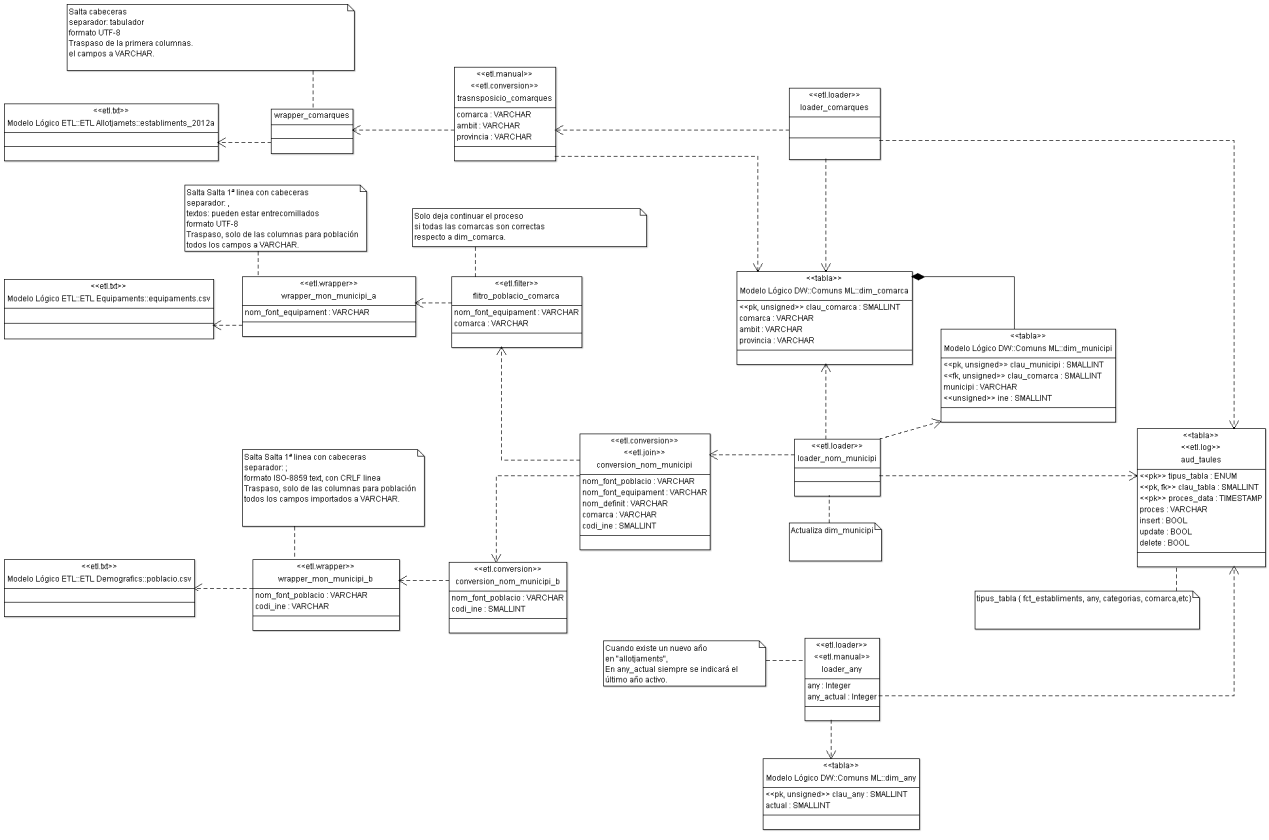
NIVEL 2: PAQUETE ETL COMUNS.

La carga de la tabla *dim_Any* se produce cuando se decide importar un nuevo fichero csv de un año en concreto. Se indica que es un proceso manual, ya que se entraran los datos del *loader_any* manualmente para después realizar el proceso de carga, que puede implicar el cambio de los valores del año actual.

Para determinar la lista con los nombres normalizados del municipio se basará en la información que suministran las fuentes de *poblacio.csv* y *equipaments.csv.*

Para determinar la lista de comarcas, ámbitos y provincias se utilizará la información que suministra la fuente de datos *establiments_2012* y con ella se normalizarán los nombres, para futuras importaciones.

La tabla del *conversion_nom_municipi* como es una tabla que contiene información de la diferentes formas en que se nombra a un municipio quizás sea interesante que se mantenga para futuras importaciones y no se inicialice al final o al principio de un proceso.



Diagr4

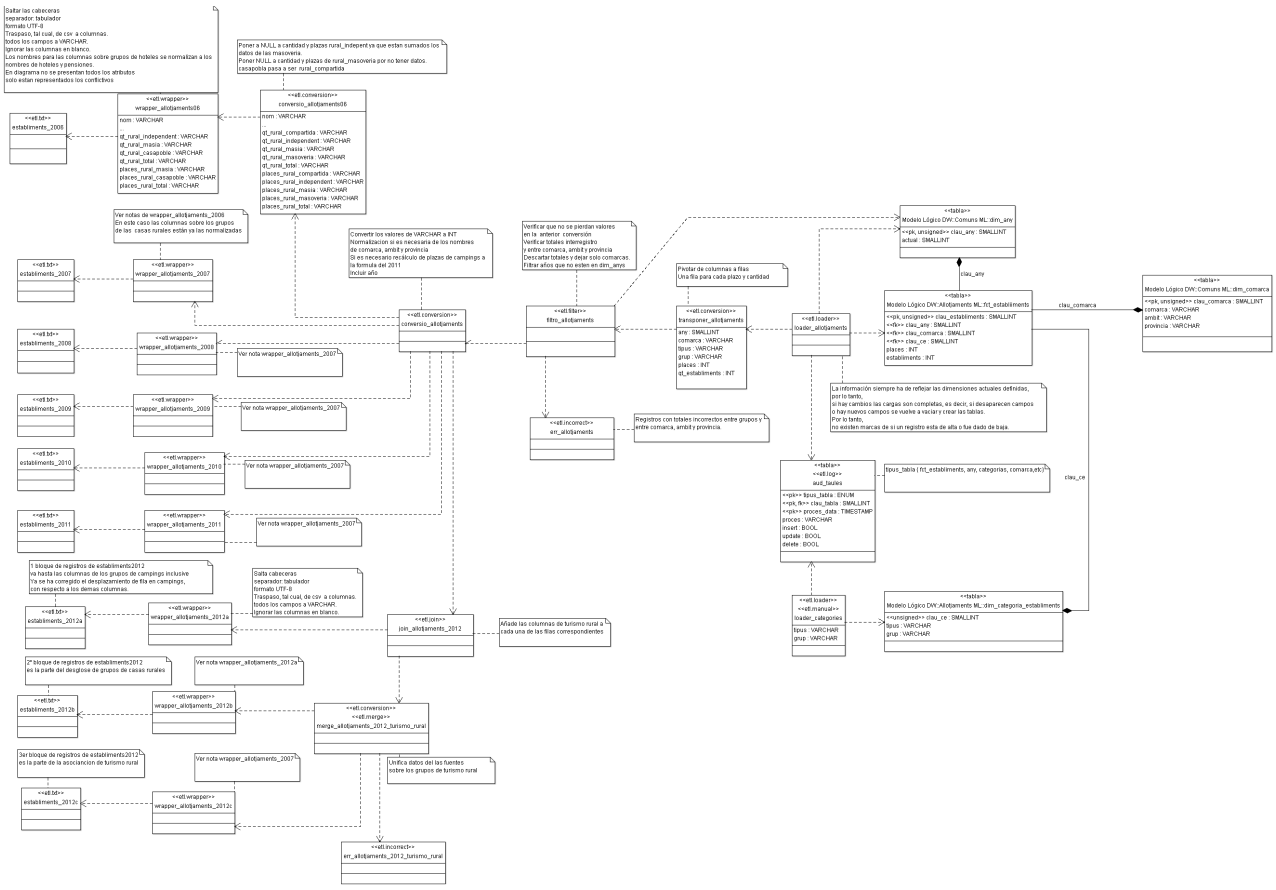
NIVEL 2: PAQUETE *ETL ALLOTJAMENTS*.

Este es un diagrama más complejo ya que se analiza cada uno de los ficheros planos con los datos de cada año, haciendo indicaciones precisas de que ajustes hay que realizar en cada uno de ellos.

En el caso del 2012 se ha divido el fichero original en otros 3 ficheros en que se guardan cada uno de los tres grandes bloques que contiene.

Los datos para el mecanismo *loader_categories* son entrados manualmente y surgen a partir de las cabeceras del año 2012l.

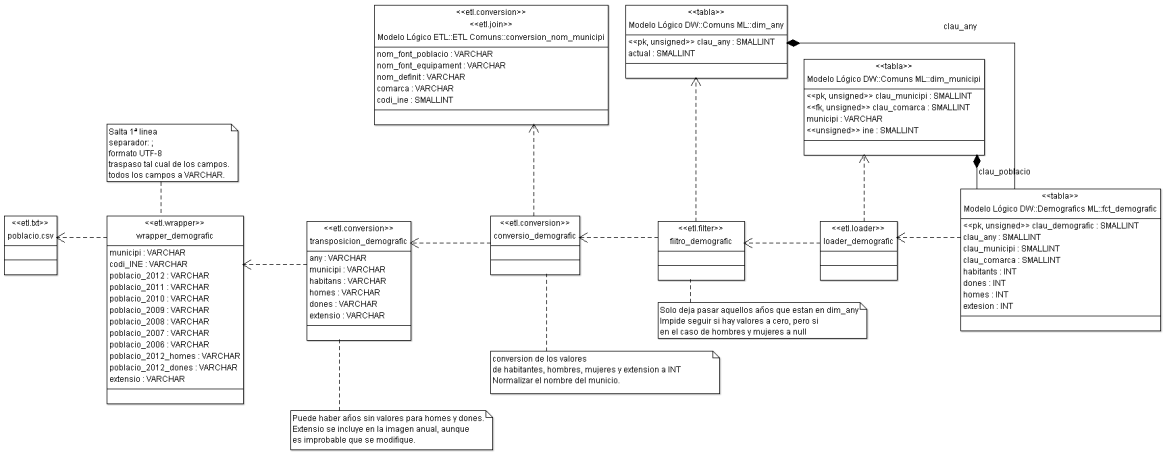
Para importación de los próximos años se aconseja conseguir una versión más próxima a un fichero csv, con lo que se facilitará mucho el trabajo.



Diagr15

NIVEL 2: PAQUETE *ETL DEMOGRAFICS*.

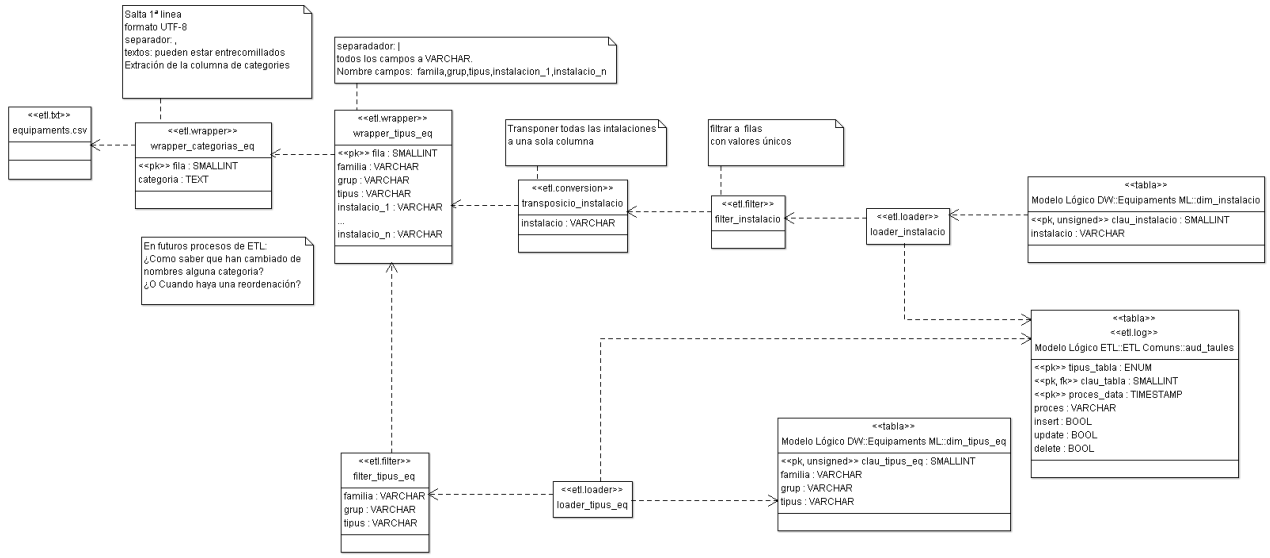
No hay información adicional que exponer sobre el diagrama presentado. Solo indicar que cuando venga futuros ficheros habrá que revisar si vuelven a repetir los últimos 6 años o viene solo el nuevo año.



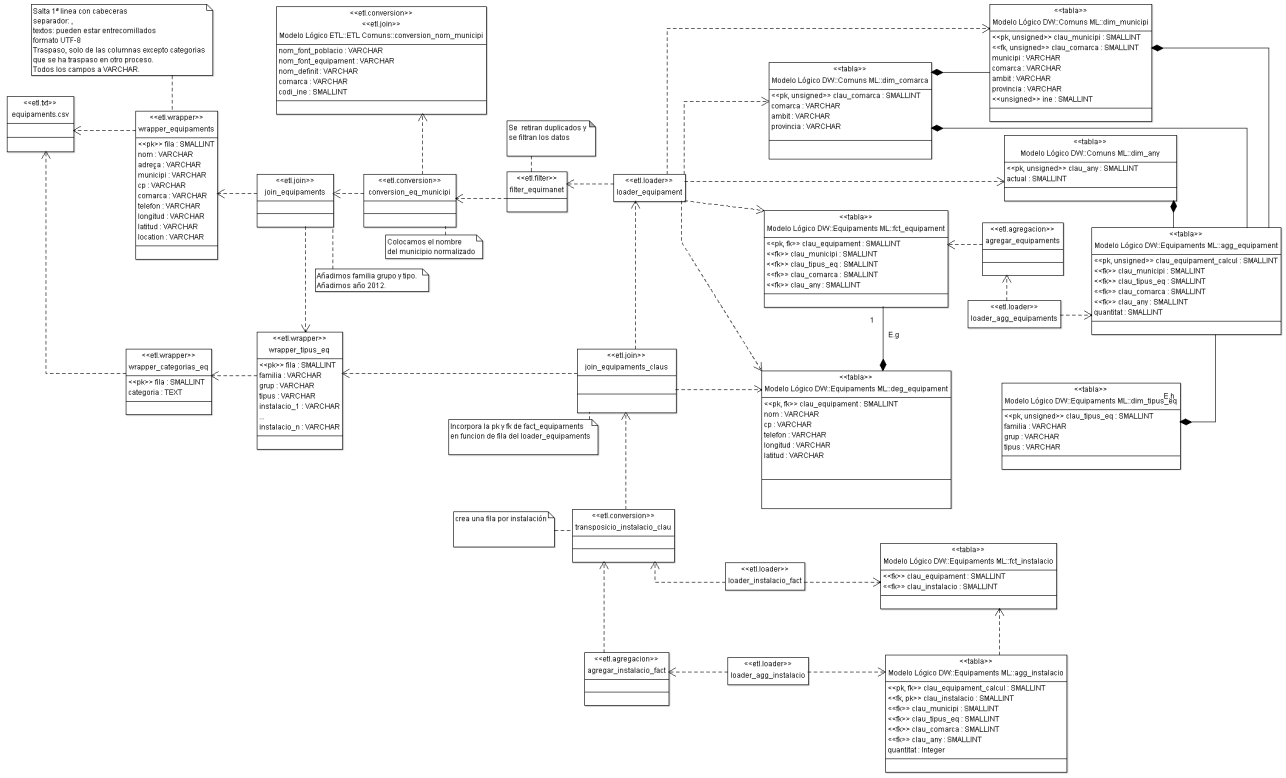
Diagr16

NIVEL 2: PAQUETE *ETL EQUIPAMENTS*.

El paquete *ETL Equipaments* se desglosa en dos diagramas para facilitar la lectura del proceso. Así tenemos por un lado la carga de los equipamientos y por otro la carga de las categorías de los equipamientos. Primero hay que realizar la importación de las categorías y después de los equipamientos.



Diagr17



Diagr18