

Construcción y explotación de un almacén de datos para el análisis de información inmobiliaria.

Memoria del Proyecto.

Alumno:

Miguel Ángel Escalonilla González
Ingeniería Técnica en Informática de Gestión

Consultor:

Xavier Plaza Sierra

Fecha:

10/01/2012

Resumen

Los Almacenes de Datos, o Data Warehouse en inglés, se han convertido en pieza fundamental del denominado *Business Intelligence*, Inteligencia Empresarial en español, que consiste en la capacidad que tienen las organizaciones de tomar mejores decisiones, y lograr ventajas competitivas con ello, mediante la obtención de conocimiento tras el análisis de los datos que manejan.

Se trata de repositorios de datos cuya finalidad es recopilar, resumir y tratar eficientemente grandes volúmenes de datos existentes en los sistemas de información de las organizaciones, proporcionando un análisis de la información desde distintas perspectivas o dimensiones. El hecho de que los costes de almacenamiento masivo y conectividad se hayan reducido enormemente ha provocado que la industria piense en ellos como algo separado de las bases de datos orientadas al proceso.

Este documento es la memoria del trabajo final de la carrera de Ingeniería Técnica en Informática de Gestión, centrado sobre el área de Almacenes de Datos. Se describen los trabajos realizados para la construcción y explotación de un almacén de datos en el ámbito inmobiliario.

Abstract

The Data Warehouse have become in a very important element of the well-known field called *Business Intelligence*, which it consist in the ability of the companies to take better decisions, and to obtain competitive advantages by means of making knowledge after the analysis of the information they handle.

These data repositories have the purpose of collecting, summarize and effectively processing high volumes of existing data in the information systems of the organizations, providing an analysis of information from different perspectives or dimensions. The fact that mass storage and connectivity costs have been vastly reduced has led the industry think of them as something separate from the classic database process oriented.

This document is the report of the final work in Technical Engineering Degree in Computer Science, focused on the area of Data Warehouses. It describes the work done for the construction and deployment of a data warehouse in the realm of property business.

CONTENIDO

| | |
|--|----|
| Introducción y justificación del TFC | 6 |
| 1. Objetivos del TFC | 7 |
| 2. Enfoque y método seguido..... | 8 |
| 3. Planificación del proyecto | 9 |
| 3.1 Planificación global..... | 9 |
| 3.2 Planificación detallada..... | 10 |
| 3.3 Incidencias, riesgos y plan de contingencia..... | 11 |
| 3.4 Diagrama de Gantt..... | 12 |
| 4. Relación de productos obtenidos | 12 |
| 5. Breve descripción del resto de apartados..... | 14 |
| 6. Análisis..... | 15 |
| 6.1 Estudio de los datos origen | 15 |
| 6.2 Diagramas de casos de uso | 18 |
| 6.3 Diagrama del modelo conceptual | 19 |
| 6.3.1 Identificación de los Hechos | 19 |
| 6.3.2 Granularidad adecuada..... | 20 |
| 6.3.3 Identificación de las dimensiones y sus atributos | 20 |
| 6.3.4 Métricas e indicadores..... | 21 |
| 6.3.5 Modelo conceptual..... | 22 |
| 7. Diseño | 24 |
| 7.1 Diagrama de la arquitectura software | 24 |
| 7.2 Diagrama de la arquitectura hardware | 25 |
| 7.3 Diseño de la base de datos y diagrama del modelo físico | 25 |
| 7.4 Proceso ETL de carga y generación de la base de datos..... | 28 |
| 8. Informes implementados: Diseño y descripción..... | 37 |
| 8.1 Correcciones realizadas sobre proceso de carga | 37 |
| 8.2 Informes preparados | 39 |
| 8.2.1 Distribución de inmuebles | 40 |
| 8.2.1.1 Número de inmuebles en oferta | 40 |
| 8.2.1.2 Inmuebles vendidos y en oferta..... | 43 |
| 8.2.1.3 Metros cuadrados máximos, medios y mínimos | 44 |
| 8.2.2 Análisis precios de oferta..... | 46 |
| 8.2.2.1 Inmuebles con precio rebajado..... | 46 |
| 8.2.2.2 Evolución anual de precios | 47 |
| 8.2.2.3 Comparativa de precios por zona..... | 48 |
| 8.2.2.4 Precios de oferta máximos, medios y mínimos | 49 |
| 8.2.3 Ventas | 50 |
| 8.2.3.1 Precios y tiempo medios de venta | 50 |
| 8.2.3.2 Diferencia entre precios de venta y oferta..... | 52 |
| 8.2.3.3 Ingresos anuales | 54 |
| 8.2.4 Inmueble "tipo" | 55 |
| 9. Conclusiones | 57 |
| 10. Líneas de evolución futuras..... | 58 |
| Glosario | 59 |
| Bibliografía y referencias | 60 |
| Anexo 1. Breve teoría sobre almacenes de datos..... | 61 |
| Anexo 2. Código fuente fichero por lotes "GeneraDW.bat" | 64 |
| Anexo 3. PL/SQL "Crear_Tablas_Temp.sql"..... | 66 |
| Anexo 4. Ficheros .ctl cargas SQL *Loader tablas temp | 67 |
| Anexo 5. PL/SQL "Crear_Tablas_Definitivas.sql"..... | 68 |
| Anexo 6. Carga Tabla Definitivas..... | 69 |
| Anexo 7. Código fuente fichero por lotes "Carga_Mensual.bat" | 73 |
| Anexo 8. Carga Mensual de Mantenimiento..... | 75 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1. Diagrama del ciclo de vida del negocio dimensional. Fuente: The Data Warehouse Toolkit. Second Edition. Wiley Computer Publishing. 2002..... | 8 |
| Figura 2. Tabla con las fases y fechas clave del TFC..... | 9 |
| Figura 3. Tabla con desglose de tareas por y fechas por fase. | 10 |
| Figura 4. Diagrama de Gantt con la planificación del TFC..... | 12 |
| Figura 5. Estructura de carpetas del proyecto..... | 13 |
| Figura 6. Primeras 3 zonas distintas de aquellos registros con 5 zonas informadas..... | 16 |
| Figura 7. Zona 2 con descripción y caracteres añadidos..... | 16 |
| Figura 8. Diagrama de casos de uso del almacén de datos..... | 19 |
| Figura 9. Modelo conceptual del Almacén de datos..... | 23 |
| Figura 10. Arquitectura software del almacén de datos..... | 24 |
| Figura 11. Diagrama de la arquitectura hardware del almacén de datos..... | 25 |
| Figura 12. Tabla de zonas..... | 26 |
| Figura 13. Tabla de periodos..... | 26 |
| Figura 14. Tabla de tipos de inmueble..... | 26 |
| Figura 15. Tabla de inmuebles..... | 26 |
| Figura 16. Tabla de meses..... | 26 |
| Figura 17. Tabla de situación de inmuebles..... | 27 |
| Figura 18. Modelo de relaciones de la base de datos..... | 28 |
| Figura 19. Fases del proceso ETL..... | 28 |
| Figura 20. Primer subproceso ETL..... | 29 |
| Figura 21. Mensaje de proceso abortado..... | 30 |
| Figura 22. Mensaje de finalización de primer subproceso ETL..... | 31 |
| Figura 23. Segundo subproceso ETL..... | 31 |
| Figura 24. Mensaje de finalización de segundo subproceso ETL..... | 32 |
| Figura 25. Tercer subproceso ETL..... | 32 |
| Figura 26. Cuarto subproceso ETL..... | 33 |
| Figura 27. Quinto subproceso ETL..... | 33 |
| Figura 28. Mensaje de finalización de quinto subproceso ETL..... | 33 |
| Figura 29. Proceso ETL de mantenimiento evolutivo..... | 35 |
| Figura 30. Subproceso cuarto ETL de la carga mensual de mantenimiento..... | 35 |
| Figura 31. Ejemplo de diagrama de almacén de datos en estrella..... | 62 |
| Figura 32. Ejemplo de diagrama de almacén de datos en copo de nieve..... | 62 |
| Figura 33. Ejemplo cubo OLAP de 3 dimensiones: tiempo, producto y mercado..... | 63 |

ÍNDICE DE CAPTURAS DE PANTALLA ATLAS SBI

| | |
|--|----|
| Captura 1. Menú inicial al acceder a Atlas SBI Server..... | 39 |
| Captura 2. Elementos de carpeta de distribución de inmuebles..... | 40 |
| Captura 3. Tabla dinámica con el número de inmuebles en oferta..... | 41 |
| Captura 4. Número de inmuebles en oferta por país, provincia y tamaño. | 41 |
| Captura 5. Número de pisos en oferta por número de habitaciones, país y provincia..... | 42 |
| Captura 6. Tabla dinámica con el número de inmuebles en oferta y vendidos..... | 43 |
| Captura 7. Número de inmuebles en oferta y vendidos por país, provincia y número de habitaciones..... | 43 |
| Captura 8. Metros cuadrados máximos, medios y mínimos por provincia y desglose por resto de zonas. . | 44 |
| Captura 9. Metros cuadrados máximos, medios y mínimos por tipo de inmueble. | 44 |
| Captura 10. Metros cuadrados máximos, medios y mínimos por tamaño y por número de habitaciones. | 45 |
| Captura 11. Elementos de carpeta de análisis de precios de oferta..... | 46 |
| Captura 12. Lista de inmuebles en oferta con precio rebajado..... | 46 |
| Captura 13. Evolución de precios medios de oferta por metro cuadrado en provincia determinada y por tipo en dicha provincia. | 47 |
| Captura 14. Evolución de precios medios de oferta por metro cuadrado por tamaño y número de habitaciones en provincia determinada..... | 47 |
| Captura 15. Comparativa de precios medios de oferta por metro cuadrado por zonas..... | 48 |
| Captura 16. Precios medios de oferta por metro cuadrado por provincia y desglose por resto de zonas. . | 49 |
| Captura 17. Precios medios de oferta por metro cuadrado por tipo, tamaño y número de habitaciones en provincia determinada..... | 49 |
| Captura 18. Elementos de carpeta de ventas..... | 50 |
| Captura 19. Precios de venta máximos, medios, mínimos y tiempos de venta en junio del 2007. Presentación..... | 50 |
| Captura 20. Precios máximos, medios y mínimos de ventas reales y tiempos de venta en junio de 2007 en Menorca. Detalle por municipio, tipo de inmueble y tamaño. | 51 |
| Captura 21. Precios medios por metro cuadrado de venta y oferta por provincia..... | 52 |
| Captura 22. Precios medios por metro cuadrado de venta y oferta desglosados por zona y tipo de inmueble en Girona (enero 2006). | 53 |
| Captura 23. Precios medios por metro cuadrado de venta y oferta desglosados por tamaño y número de habitaciones en Girona (enero 2006). | 53 |
| Captura 24. Ingresos anuales por ventas realizadas por año, país y provincia..... | 54 |
| Captura 25. Ingresos anuales por ventas realizadas por año, país, provincia y tipo de inmueble..... | 55 |
| Captura 26. Tabla de inmuebles tipo de Andorra. | 56 |
| Captura 27. Tabla de inmuebles tipo de España..... | 56 |

Introducción y justificación del TFC

El trabajo final de carrera, en adelante TFC, está orientado a la puesta en práctica de distintos conocimientos que se han ido adquiriendo a lo largo del estudio de las diferentes asignaturas en el contexto de un trabajo concreto. El área elegida, almacenes de datos, conlleva que, fundamentalmente, se han puesto en práctica conocimientos obtenidos de las asignaturas relacionadas con las bases de datos o almacenamiento, si bien, como no puede ser de otra forma, otros conocimientos derivados de asignaturas como minería de datos, estadística o programación han tenido que ser utilizados.

En este caso se ha planteado un caso ficticio que nos lleve a la construcción y explotación de un almacén de datos dentro del ámbito inmobiliario. Concretamente, la empresa ficticia “Un techo para todos” ha encargado a nuestra consultora la creación de un *Data Warehouse* que permita poder explotar la información con la que cuentan mediante una herramienta de *Business Intelligence*, en adelante BI, y servir, de esta forma, como ayuda en la toma de decisiones de la compañía. Para ello, nos ha facilitado un fichero texto que contiene información de su extenso catálogo de inmuebles, y que han creado concatenando diferentes hojas de cálculo donde almacenan la información.

Actualmente, en un contexto social y económico como el que vivimos, el uso de la información para lograr conocimiento y poder tomar decisiones más correctas, obteniendo ventajas competitivas por ello, se ha convertido en una de las tareas más importantes para las organizaciones. El hecho de que cada vez se manejen y almacenen mayores volúmenes de datos por parte de las empresas, de que no haya prácticamente ninguna acción en las compañías que no quede registrada, y que el precio de la memoria no volátil cada vez sea más bajo, ha disparado el interés por disponer de modelos de almacenamiento mejor orientados que las bases de datos transaccionales para realizar consultas eficientes desde el punto de vista de negocio. Si a ello le aplicamos herramientas de obtención de conocimiento –conocidas como *Data Mining*– resulta evidente el interés que desde el punto de vista empresarial, o de otro tipo, puede despertar.

Este trabajo, dentro del área de la información inmobiliaria, permite a nuestra empresa ficticia poder comparar diferentes zonas donde mantienen inmuebles por distintos criterios, tipologías y precios de los mismos, realizar análisis de las ventas realizadas, y obtener información estadística de distribuciones de la venta y la oferta por diferentes criterios, con el objetivo de poder ajustar, oferta y demanda, y así ser más competitivos.

1. Objetivos del TFC

El objetivo más importante del TFC es que el alumno profundice en el diseño y construcción de un almacén de datos, y que se familiarice con las diferentes técnicas existentes para diseñar, tanto la base de datos, como las herramientas OLAP (*On Line Analytical Processing*), que permiten explotar la información bajo una perspectiva multidimensional y un tiempo de respuesta muy eficiente. En este TFC, en concreto, se solicita que el diseño físico sea ROLAP (*Relational On Line Analytical Processing*), que se trata de una implementación OLAP y bajo un esquema relacional donde las tablas pueden encontrarse desnormalizadas y la información agregada. Por lo demás, en el proyecto podemos diferenciar entre objetivos globales y específicos.

Los objetivos globales del TFC son:

- Entender qué son los almacenes de datos y cuál es su misión
- Preparar un plan de trabajo con la planificación estimada de tareas para todo el proyecto
- Análisis de los requerimientos del cliente y su contenido
- Análisis de las fuentes de datos desde las que alimentaremos el almacén
- Diseñar y construir el almacén de datos bajo una perspectiva multidimensional
- Entender las fases por las que todo almacén de datos debe pasar para poder alimentarlo
- Preparar una solución automatizada de generación y mantenimiento evolutivo del almacén construido
- Explotar la información almacenada mediante una herramienta de obtención de conocimiento

Y los objetivos específicos del proyecto son los requerimientos que el cliente nos ha solicitado, a saber, disponer de informes donde puede encontrar la siguiente información:

- Número de inmuebles por zona, tipología y características
- Lista de inmuebles que han bajado de precio. Mostrar la evolución de precios por zona, tipología y características
- Precios ofrecidos máximos, mínimos y medios por zona, tipología y características
- Precios de venta real máximos, mínimos y medios por zona, tipología y características
- Diferencia entre los precios ofrecidos y de venta real
- Metros cuadrados máximos, mínimos y medios por zona, tipología y características
- Distribución (número de pisos) por número de habitaciones, por zona, tipología y características
- Tiempo medio de venta por zona, tipología y características
- Número de inmuebles vendidos y existentes por zona, tipología y características

Todo ello deberá ofrecerse al usuario para un mes de un año determinado.

Además, debe ser posible obtener desde el almacén de datos una consulta o listado que nos muestre cuál es el piso “estándar” español y andorrano, mostrando su tipología, características y precio. Se explicará más adelante el criterio que hemos seguido para encontrar el piso “tipo”. Se debe destacar que el cliente opera, actualmente, sólo en España y Andorra.

2. Enfoque y método seguido

El proyecto ha sido dividido en etapas. La primera de ellas fue el plan de trabajo y el análisis preliminar de requerimientos. Un análisis más detallado de los mismos y el diseño conceptual y técnico conformaron la segunda etapa, y finalmente, la implementación fue la fase final.

El ciclo de vida del proyecto queda perfectamente descrito en este esquema propuesto por los conocidos autores Ralph Kimball y Margy Ross: plan de trabajo, definición de los requerimientos, diseño conceptual, técnico y físico, desarrollos e implementación, implantación y mantenimiento en un ciclo que se retroalimenta.

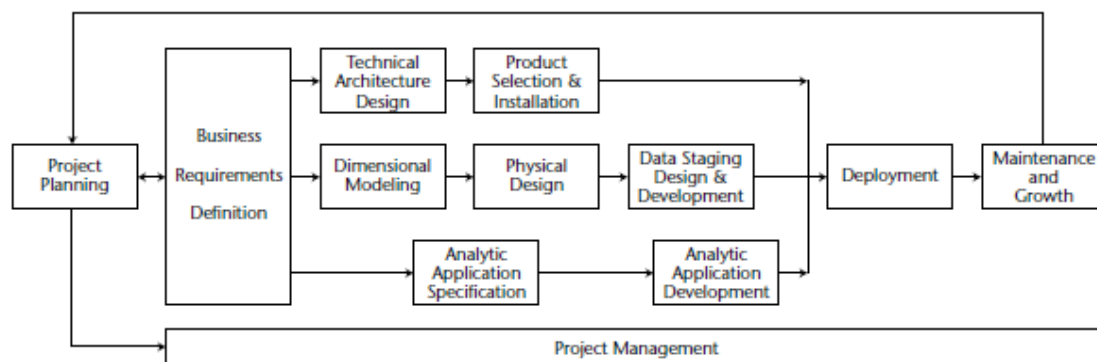


Figura 1. Diagrama del ciclo de vida del negocio dimensional. Fuente: The Data Warehouse Toolkit. Second Edition. Wiley Computer Publishing, 2002.

Es importante destacar la necesidad de darse cuenta del cambio de filosofía existente entre el almacenamiento transaccional y el almacenamiento orientado a lograr conocimiento por la organización. Esto es lo que los autores mencionados anteriormente quieren reseñar cuando hablan del “ciclo de vida de las dimensiones del negocio”. Este tipo de proyectos deben centrarse en las necesidades de negocio, los datos deben ser presentados al usuario de forma dimensional, y por último, aunque el almacenamiento de datos debe ser un proyecto en marcha y vivo (no olvidemos su mantenimiento), cada implementación debe tener un principio y un fin definidos¹. En este sentido, en este trabajo se ha seguido un método iterativo en el que con cierta frecuencia era necesario mirar hacia atrás, y redefinir ciertos aspectos correspondientes a anteriores etapas. Así mismo, la asimilación de los conceptos más importantes que subyacen en la teoría de almacenes de datos ha sido progresiva conforme avanzaba en el trabajo.

(1): Ralph Kimball / Margy Ross. The Data Warehouse Toolkit, Second Edition. The Complete Guide to Dimensional Modelling. Cap. 16: “Building the Data Warehouse”.

Por último, para finalizar este apartado, indicar que se han realizado entregas parciales de informes de cada una de las etapas antes de esta memoria, así como de los distintos entregables del producto final construido (almacén, procesos automáticos de construcción, carga y mantenimiento, código fuente de todo ello, documentos analíticos de la herramienta de BI, etc.). Además, se ha entregado una presentación virtual en formato de vídeo con el resumen de los aspectos más relevantes del trabajo realizado y del resultado obtenido.

También conviene indicar que toda la implementación se ha realizado sobre una máquina virtual, en adelante MV, entregada al cliente.

3. Planificación del proyecto

3.1 Planificación global

Las fases y fechas clave del proyecto son las siguientes.

| Fase Proyecto | Inicio | Fin |
|--|------------|------------|
| Fase I: Análisis Preliminar de Requerimientos y Plan de Trabajo (PEC1) | 22/09/2011 | 05/10/2011 |
| Fase II: Análisis de Requerimientos y Diseño Conceptual y Técnico (PEC2) | 06/10/2011 | 09/11/2011 |
| Fase III: Implementación | 10/11/2011 | 21/12/2011 |
| Entrega Final: memoria, presentación virtual y producto. | 22/12/2011 | 10/01/2012 |
| Debate virtual | 23/01/2012 | 26/01/2012 |

Figura 2. Tabla con las fases y fechas clave del TFC.

3.2 Planificación detallada

El proyecto se ha dividido en fases, y éstas a su vez, han sido divididas en tareas. A continuación, se muestra la planificación detallada por tareas.

| Tareas Proyecto | Inicio | Fin |
|---|-------------------|-------------------|
| Fase 0: Tareas iniciales | 15/09/2011 | 24/09/2011 |
| Lectura Módulos Teoría | 15/09 | 22/09 |
| Descarga Enunciado proyecto / recursos | 23/09 | 24/09 |
| | | |
| Fase I: Análisis Preliminar de Requerimientos y Plan de Trabajo (PEC1) | 24/09/2011 | 05/10/2011 |
| Lectura detallada enunciado TFC | 24/09 | 25/09 |
| Recopilación Información y Lectura | 25/09 | 03/10 |
| Instalación Software Fase I | 26/09 | 27/09 |
| Análisis Preliminar Elementos Identificados | 27/09 | 28/09 |
| Análisis Informes Requeridos | 29/09 | 30/09 |
| Análisis Fuentes Datos | 01/10 | 02/10 |
| Composición Documento entrega PEC1 | 03/10 | 04/10 |
| Revisión / Corrección / Entrega | 05/10 | 05/10 |
| | | |
| Fase II: Análisis de Requerimientos y Diseño Conceptual y Técnico (PEC2) | 06/10/2011 | 09/11/2011 |
| Recopilación Información | 06/10 | 12/10 |
| Instalación Software Fase II | 06/10 | 07/10 |
| Análisis Detallado Requerimientos | 13/10 | 16/10 |
| Diseño Modelo Dimensional Conceptual | 17/10 | 19/10 |
| Diseño Modelo Lógico/Físico | 20/10 | 22/10 |
| Diseño Procesos ETL | 23/10 | 29/10 |
| Composición Documento entrega PEC2 | 30/10 | 03/11 |
| Revisión / Corrección | 04/11 | 06/11 |
| Días Preparación Fase Siguiente | 07/11 | 09/11 |
| Entrega | 09/11 | 09/11 |
| | | |
| Fase III: Implementación | 10/11/2011 | 21/12/2011 |
| Recopilación Información | 10/11 | 15/11 |
| Instalación Software | 17/11 | 19/11 |
| Construcción Almacén de Datos (BB.DD, cargas,...) | 20/11 | 25/11 |
| Construcción Informes y Análisis información Obtenida | 26/11 | 06/12 |
| Composición Documento entrega PEC3 | 07/12 | 15/12 |
| Revisión / Corrección | 16/12 | 19/12 |
| Entrega | 20/12 | 21/12 |
| | | |
| Entrega Final | 22/12/2011 | 10/01/2012 |
| Composición Memoria Final | 22/12 | 02/01 |
| Revisión / Corrección | 03/01 | 03/01 |
| Composición Presentación virtual y Defensa | 04/01 | 09/01 |
| Entrega | 10/01 | 10/01 |
| | | |
| Debate virtual | 23/01/2012 | 26/01/2012 |

Figura 3. Tabla con desglose de tareas por y fechas por fase.

3.3 Incidencias, riesgos y plan de contingencia

En todo proyecto resulta necesario evaluar las posibles incidencias que puedan producirse, los riesgos a los que estamos expuestos, y disponer de un plan de contingencia en ese caso. En este caso, se advirtieron los siguientes riesgos y medidas a aplicar para solventarlo.

- **Motivos Técnicos**

- El trabajo técnico del proyecto se va a realizar sobre un PC de sobremesa que cumple con los requisitos de hardware y software solicitados por el cliente. No obstante, en caso de problemas, disponemos de otro portátil que también cumple con ellos y en donde deberemos instalar los componentes necesarios
- Se realizarán trabajos de backup o copias de seguridad con frecuencia diaria sobre disco duro externo desde el cual se pueda recuperar la información
- La instalación de la máquina virtual, así como su configuración, se instalará lo antes posible con el objetivo de adelantar este trabajo y que no genere problemas posteriores. Se dispone de un consultor técnico (laboratorio) con quien resolver las dudas. Una vez que dicha máquina virtual esté instalada y configurada, será salvada por si acaso debe ser restaurada

- **Motivos Laborales**

- Es posible que deba dedicar más tiempo a alguno de los proyectos en los que estoy trabajando en la empresa donde desarrollo mi actividad laboral. En caso de ser así, la solución pasará por dedicar horas en otros momentos: festivos, vacaciones o noche. En todo caso, en principio, no se trata de una eventualidad con riesgo alto para el proyecto pues lo voy a acometer en solitario precisamente por este tipo de cuestiones. Además, estas situaciones no son inmediatas, sino que podré anticiparlas para replanificar las tareas y así poder asumirlo

- **Motivos de Salud**

- En caso de contingencias no graves de este tipo la solución pasará de nuevo por dedicar horas en otros momentos u horarios. Para no tener problemas se llevará el trabajo día a día y por adelantado

Finalmente, no fue necesario tomar ninguna medida alternativa pues no tuve incidencias o problemas que desviaran la planificación prevista.

3.4 Diagrama de Gantt

A continuación se muestra el diagrama de Gantt con las tareas en las que se ha dividido el proyecto.

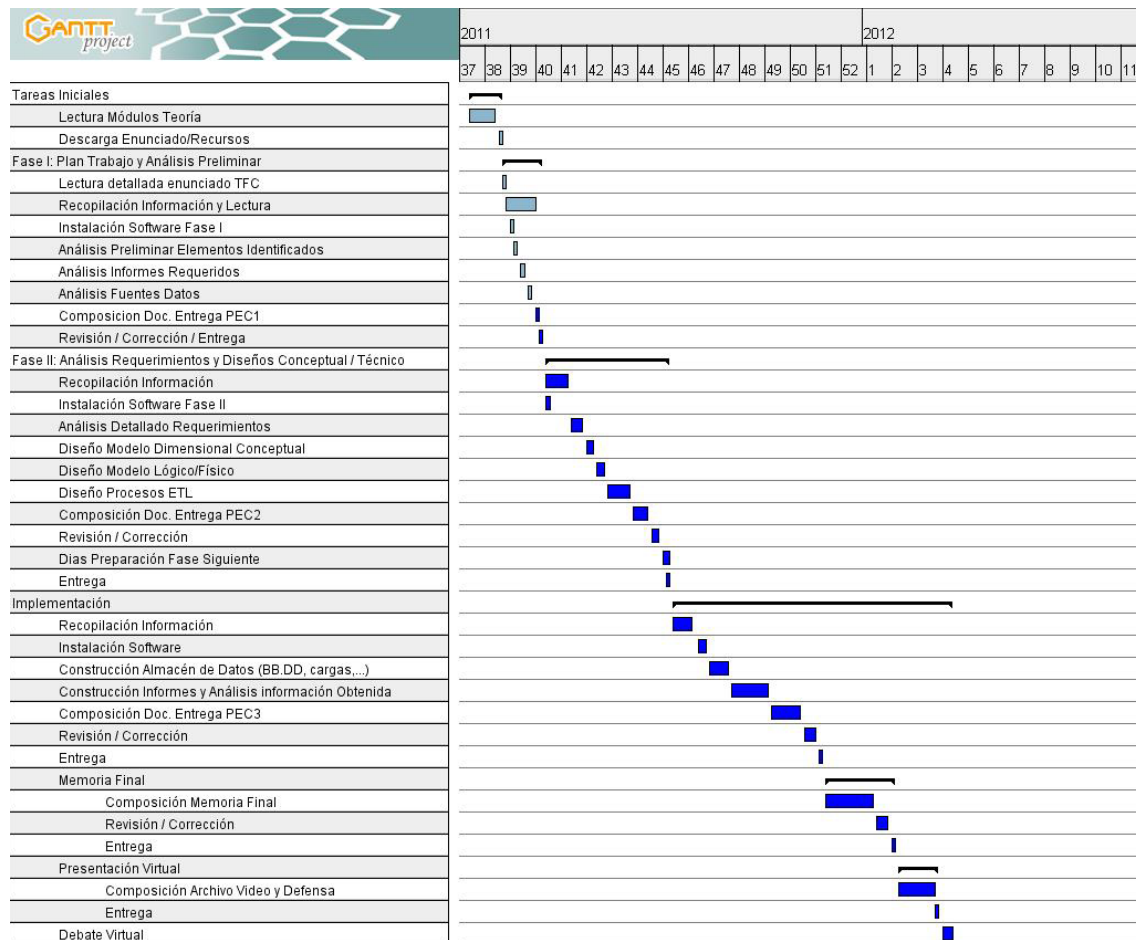


Figura 4. Diagrama de Gantt con la planificación del TFC.

4. Relación de productos obtenidos

Durante el trabajo a lo largo del proyecto se han generado distintos entregables que detallamos a continuación.

- **Memoria Final.** Se trata de este documento resumen de todo el trabajo realizado durante el semestre en el proyecto
- **Presentación virtual.** Se trata de un vídeo que describe brevemente el trabajo realizado durante las etapas del proyecto
- **Documento PEC1.** Entregado al término de la primera de las etapas del proyecto como informe del análisis previo de requerimientos y plan de trabajo
- **Documento PEC2.** Entregado al término de la segunda de las etapas del proyecto como informe del análisis definitivo de requerimientos y diseños conceptual y técnico

- **Documento PEC3.** Entregado al término de la tercera de las etapas del proyecto como informe de la fase de implementación
- **Almacén de datos.** Se entrega cargado en la MV con los datos que el cliente nos ha facilitado una vez transformados, normalizados y adaptados al modelo
- **Proceso ETL automático de carga y regeneración.** Se entrega al cliente código fuente y ejecutable del proceso por lotes, así como de todos los subprocesos que lo componen dentro de la MV (clase Java de extracción y transformación, ficheros script de generación y carga de tablas temporales y definitivas)
- **Proceso ETL automático de mantenimiento.** Se entrega al cliente código fuente y ejecutable de este proceso por lotes, así como de todos los subprocesos que lo componen dentro de la MV
- **Informes requeridos por el cliente.** Se entregan con la MV los documentos analíticos confeccionados con la herramienta Atlas SBI Studio elegida por el cliente. Se detalla cada uno de ellos más adelante

La MV se entrega con las herramientas Atlas SBI Studio, para creación y configuración de los documentos analíticos de reporte, y Atlas SBI Server, necesaria para acceder a la ejecución de los informes desarrollados, instaladas y configuradas.

Además, dentro de la máquina virtual se entrega en la unidad *E:*, una carpeta denominada *TFC* que es el directorio raíz de todo el proyecto. En este directorio se entregan todas las carpetas y elementos relacionados con el proyecto.

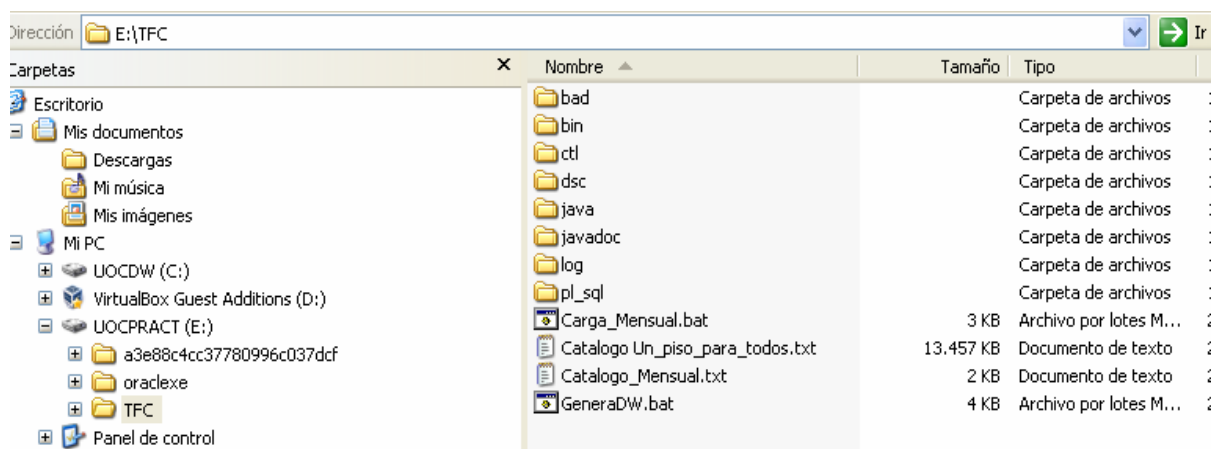


Figura 5. Estructura de carpetas del proyecto.

Carpetas:

- “*bad*” → Carpeta donde quedarán, en caso de generarse, archivos con registros rechazados en las cargas efectuadas con SQL *Loader (se explica en apartado 6). Se entrega vacía
- “*bin*” → Carpeta que contiene el archivo ejecutable de la clase Java usada en primer paso del proceso ETL
- “*ctl*” → Carpeta que contiene los ficheros que controlan las cargas SQL *Loader

- “*dsc*” → Carpeta donde quedan, en caso de generarse, archivos con registros no cargados por no cumplir el criterio de carga definido para SQL *Loader (se explica en apartado 6). Se entrega vacía
- “*java*” → Carpeta que contiene el archivo fuente de la clase Java usada en primer paso del proceso ETL
- “*javadoc*” → Carpeta que contiene la documentación de la clase Java anterior. Se puede acceder a dicha documentación mediante el archivo “*index.html*” que contiene
- “*log*” → Carpeta donde quedarán los archivos de auditoria de las cargas realizadas con SQL *Loader (se explica en apartado 6). Se entrega vacía
- “*pl_sql*” → Carpeta que contiene todos los archivos *script* PL/SQL utilizados en el proceso ETL

Archivos en raíz:

- “*Carga_Mensual.bat*” → Proceso ETL por lotes para la carga mensual de mantenimiento evolutivo del almacén (en realidad, no tiene porqué ser mensual, se pueden recibir datos de más meses)
- “*Catalogo_Un_Piso_para_todos.txt*” → Fichero original que nos ha suministrado el cliente para realizar la carga inicial
- “*Catalogo_Mensual.txt*” → Fichero ejemplo preparado para probar la carga periódica de mantenimiento. Este fichero ejemplo sólo contiene información de algunos inmuebles, pero, en realidad, debería tener información de todos los del catálogo (se explica más adelante)
- “*GeneraDW.bat*” → Proceso ETL por lotes para la carga inicial (o regeneración) del almacén de datos. Puede procesarse las veces que sea necesario para regenerar el almacén de nuevo

5. Breve descripción del resto de apartados

A partir de este punto de la memoria se describen las conclusiones de cada una de las fases en las que hemos dividido el proyecto: análisis, diseño e implementación. La memoria concluye con un apartado dedicado a las conclusiones del TFC, y otro, donde se aportan ideas para la mejora y la evolución a futuro del almacén de datos construido. Se finaliza con un glosario de términos técnicos y diferentes anexos que he creído interesante aportar.

6. Análisis

6.1 Estudio de los datos origen

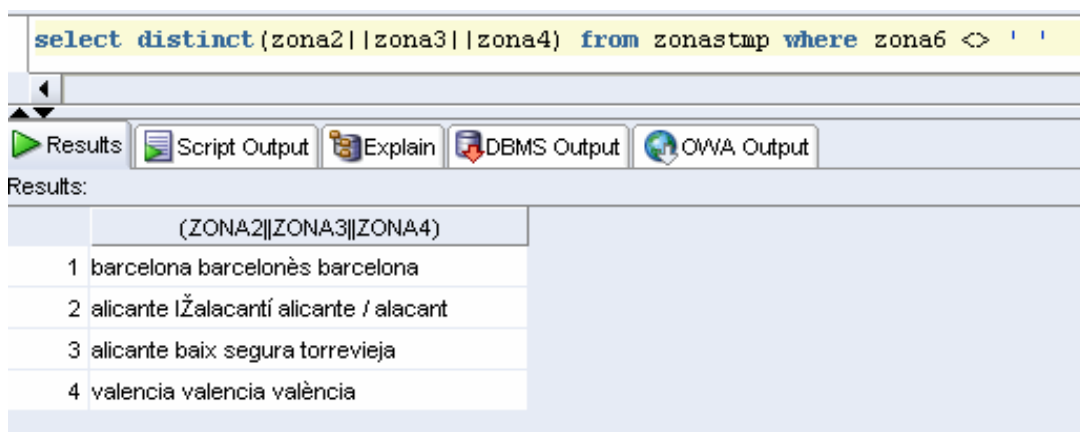
Para realizar el estudio de los datos hemos dividido el fichero texto en dos, uno de zonas (registros que comienzan por el carácter “#”), y otro de inmuebles (resto). En el fichero hay 2 tipos de filas o registros. Los que comienzan por una almohadilla (“#”) indican la zona donde están localizados los inmuebles, y los registros que vienen a continuación de estos primeros aportan datos relativos a los inmuebles.

Sobre los registros relativos a la zona geográfica (comienzan por “#”):

- Estos registros se componen de descripciones textuales en una jerarquía de 4 niveles. El cliente nos advirtió de ello y querrá poder desglosar la información según esta jerarquía
- Podemos asimilar la jerarquía a los conceptos *Provincia / Municipio / Área / Distrito*
- Siempre que ocurre un cambio de localización en alguno de los niveles aparece una cabecera informando de la nueva zona. Además, existe una relación entre las zonas y el dato “Zona” que hay en los registros de los inmuebles, si bien, como veremos más adelante, no podemos fiarnos del campo “Zona” del registro de inmuebles
- No siempre están los 4 niveles totalmente informados, pudiendo existir inmuebles que se localizan en el nivel 1, 2, 3 o 4
- A veces, aparecen tuplas con más de los 4 niveles informados. Esto deberemos tenerlo en cuenta. Ejemplo:

```
# ::: barcelona ::: barcelonÃ´s ::: barcelona ::: grÃ cia ::: vallcarca  
# ::: barcelona ::: barcelonÃ´s ::: barcelona ::: grÃ cia ::: grÃ cia
```

Mirando más detenidamente los registros que tienen cinco zonas informadas (son 123) se comprueba que responden en realidad a una misma casuística que aparece en todos ellos (ver Fig. 6). Se trata de ubicaciones de inmuebles que quedan bien definidas con su primera y tercera zona, pues éstas son la provincia y el municipio en la jerarquía que hemos adoptado. De esta forma, podemos prescindir de la segunda zona.

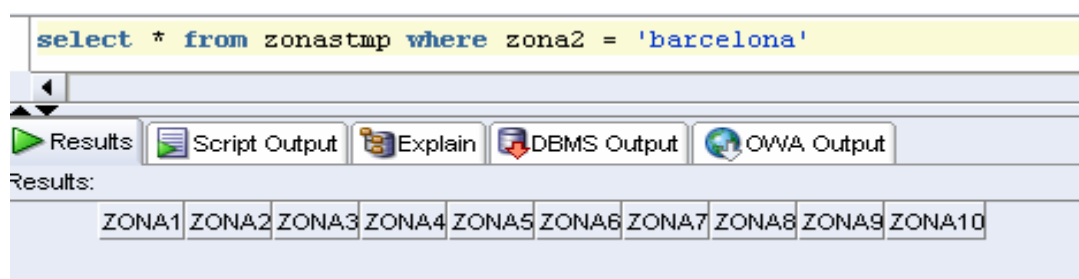


The screenshot shows a database query interface. At the top, a SQL query is entered: `select distinct (zona2||zona3||zona4) from zonastmp where zona6 <> ' '`. Below the query, there are tabs for "Results", "Script Output", "Explain", "DBMS Output", and "OWA Output". The "Results" tab is selected, and the results are displayed in a table with one column labeled "(ZONA2||ZONA3||ZONA4)".

| | (ZONA2 ZONA3 ZONA4) |
|---|--|
| 1 | barcelona barcelonès barcelona |
| 2 | alicante l'Alacantí alicante / alacant |
| 3 | alicante baix segura torrevieja |
| 4 | valencia valencia valència |

Figura 6. Primeras 3 zonas distintas de aquellos registros con 5 zonas informadas.

- La separación entre cada jerarquía se hace por medio de 3 caracteres dos puntos (":")
- El cliente nos ha dicho que los países con los que opera son España y Andorra. El literal "*españa*" como texto no aparece, pero sí "*andorra*", por lo que debemos entender que todas las zonas menos las que tienen "*andorra*" en primer nivel son de España. En todo caso, y por si acaso en un futuro se plantea ampliar el negocio a otros países, parece interesante establecer un nivel más alto en la zona geográfica que sea el país
- También se advierte que por algún motivo los literales que definen las zonas geográficas no vienen limpios del todo, de forma que al cargarlos en la tabla, los campos no sólo contienen la descripción literal geográfica, sino que deben tener algún carácter interno (como tabuladores por ejemplo). A consecuencia de ello, ocurre que una simple sentencia "*select * from tabla where zona = 'literal de zona'*" no devuelve ninguna fila, lo cual, no resulta saludable técnicamente para el manejo de la información. En la figura 7 se observa que esta consulta debería devolver resultados, pero sin embargo, no lo hace. Debemos conseguir almacenar los literales de las zonas geográficas sin ningún otro contenido que la propia cadena de caracteres



The screenshot shows a database query interface. At the top, a SQL query is entered: `select * from zonastmp where zona2 = 'barcelona'`. Below the query, there are tabs for "Results", "Script Output", "Explain", "DBMS Output", and "OWA Output". The "Results" tab is selected, and the results are displayed in a table with 10 columns labeled ZONA1 through ZONA10.

| ZONA1 | ZONA2 | ZONA3 | ZONA4 | ZONA5 | ZONA6 | ZONA7 | ZONA8 | ZONA9 | ZONA10 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| | | | | | | | | | |

Figura 7. Zona 2 con descripción y caracteres añadidos.

- Además, posiblemente motivado por las conversiones de códigos de caracteres entre sistemas, algunas zonas con apostrofes entre nombres nos vienen con un carácter raro. Ejemplo: "*l'Zhospitalet de llobregat*". Se deberán sustituir los caracteres "l'Z" por "l"

- Los campos de descripción de las zonas geográficas serán VARCHAR (100)

Sobre los registros relativos a los inmuebles:

- Los datos que aparecen en el fichero son:
 - Código de inmueble
 - Zona
 - Tipo
 - Planta
 - Dormitorio
 - Metros cuadrados
- El código del inmueble es una referencia formada por 2 letras al principio y 13 números. Se ha comprobado que esta referencia identifica de forma unívoca al inmueble en todos los registros (*en Oracle hemos definido la tabla de trabajo con esta referencia como clave primaria, y SQL *loader no ha dado problemas de integridad o restricciones al cargar*). Las 2 letras son siempre VP, VC o VW. Un ejemplo de uno de los códigos es: VW0000000546681. La Referencia del inmueble será un VARCHAR (15)
- La zona es una referencia de 20 posiciones que parece estar en el formato **pp-xxx-mm-aaa-dd-xxx** y que delimitan la jerarquía. Por el análisis realizado, podemos pensar que:
 - **pp** → indica un número que identifica a la provincia en España. En los inmuebles andorranos tal código es el 53 (no existe ninguna provincia española con tal código)
 - **mm** → indica el municipio o la comarca dentro de la provincia
 - **aaa** → indica el área o ayuntamiento dentro del municipio
 - **dd** → distrito o barrios

En realidad, pensamos que **no podemos fiarnos de esta referencia**. Existen datos incoherentes, fruto seguramente de errores manuales. Por ejemplo:

- Dentro de una misma zona, donde se suponen que los códigos deberían coincidir no lo hacen. En esta zona tenemos 2 códigos de área diferentes:

```
# ::: tarragona ::: tarragonÃ's ::: Ãrea de roda de barÃ  
VC0000000746118 43-XXX-02-001-XX-XXX  
VC0000000816314 43-XXX-02-007-XX-XXX
```

Al estar recogidos en la misma zona de tercer nivel, creemos que es un error que no coincidan el "001" y el "007".

Por lo tanto, cuando realicemos las cargas, **la ubicación de los inmuebles la tomaremos de la cabecera que tienen**, dado que esto es lo que el cliente nos ha asegurado que es correcto (enunciado).

- El tipo del inmueble es una cadena de caracteres donde los valores posibles son: *pisos, chalets, estudios, dúplex y áticos*. Se usará una variable VARCHAR (40) por si a futuro tenemos otros tipos con descripciones más extensas
- La planta es un dato que viene con un número entero indicando dicha planta, pero también puede contener letras (“bj” o “en”), 2 guiones (“--”), o en blanco. El valor 2 guiones (“--”) se transformará a blanco para homogeneizar ambos. Se usará una variable VARCHAR (5)
- El número de dormitorios es un número entero que en algunos registros no está informado y es cero. Va desde 0 hasta 99
- El número de dormitorios es un número entero que en algunos registros no está informado y es cero. El valor mínimo que nos llega es 0 y el máximo 99. Se usará una variable NUMBER (5,0) que parece suficiente para este dato
- Metros cuadrados es un número entero cuyo valor mínimo es 1 y el máximo 77895. Se usará una variable NUMBER (10,0)
- Después, vienen 25 campos con los precios de los inmuebles a la venta por mes y año, desde enero del 2006, hasta enero del 2008, es decir, todo el año 2006, 2007 y el mes de enero del 2008. Todos estos campos son importes enteros sin decimales que no tienen porque estar informados, de manera que, si están vacíos, significa que el inmueble no está en venta en ese mes y año. Esto implica que si un inmueble tiene informado precio en un mes y año, y al siguiente mes no lo tiene, se ha vendido durante este mes y año (el que sí tiene cargado el precio). Se usará una variable NUMBER (15,0) para almacenarlos

Aprovecharemos los procesos de transformación para incorporar en los registros de cada inmueble, al final, los datos de la zona, ya normalizada según se describió anteriormente, a la que pertenecen. Esta será el último registro del bloque de zona leído del fichero fuente y el país.

6.2 Diagramas de casos de uso

Podemos identificar dos tipos de perfil de usuarios del almacén de datos. El perfil del administrador, encargado de realizar las cargas y el mantenimiento del sistema. Y el perfil propio del usuario de las consultas del almacén, probablemente un empleado relacionado con la dirección de la empresa inmobiliaria.

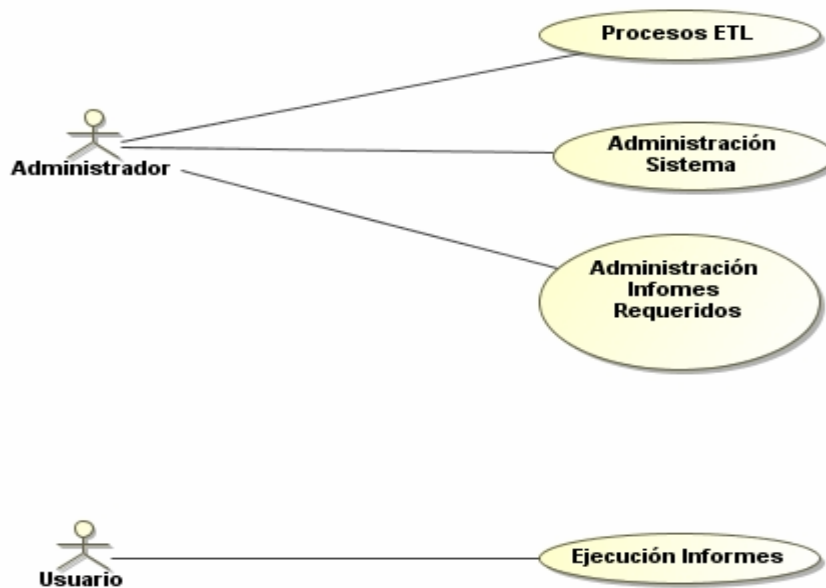


Figura 8. Diagrama de casos de uso del almacén de datos.

6.3 Diagrama del modelo conceptual

El diseño por el que se ha optado en este proyecto es de un modelo con diseño en estrella. En todo caso, lo primordial de los modelos multidimensionales es encontrar los hechos, dimensiones y sus medidas.

6.3.1 Identificación de los Hechos

En un modelo de datos dimensional el hecho representa el concepto de negocio que se quiere analizar. Tras un análisis profundo del escenario, y de los requisitos que nos solicita el cliente, el hecho a analizar que hemos definido lo hemos denominado:

- Situación Mensual de los Inmuebles

Esta será la tabla central de nuestro esquema en estrella alrededor de la cual situaremos las distintas perspectivas desde las que queremos agrupar o filtrar la información relativa a los inmuebles.

Sabemos que los requerimientos del cliente son estos:

- Número de inmuebles
- Inmuebles que han bajado de precio
- Precios de inmuebles máximos, mínimos y medios
- Precios de venta real
- Diferencia entre precios de venta y de oferta
- Metros cuadrados máximos, mínimos y medios
- Distribución (número de pisos)

- Tiempo medio de venta
- Número de inmuebles vendidos y en oferta

Todos ellos debemos presentarlos por zona, tipo de inmueble y características, y referidos a un mes de un año concreto. Además, en el caso de la distribución de los inmuebles, también se pide realizarla por número de habitaciones. Queremos significar que por el término *características* hemos acordado que hacen referencia al número de habitaciones y a los metros cuadrados del inmueble, no así a la planta que ocupa en el edificio donde se ubica, dado que el cliente no nos ha transmitido que sea un dato interesante para el tipo de informes que quiere obtener, y además, pensamos que el incorporar una variable más a la clasificación de la información puede hacer que los reportes finales resulten algo confusos. Dicho lo cual, veremos más adelante que sí incorporaremos este dato como un atributo más de la tabla de hechos, por si a futuro resultase interesante usarlo en otro tipo de informes.

Si repasamos los requisitos se aprecia que todos se refieren a mostrar la situación de los inmuebles del catálogo en un período concreto (mes), o a comparativas entre inmuebles vendidos y ofertados, desde diferentes dimensiones y métricas. Con esta única tabla de hechos pensamos dar soporte a todas las necesidades.

6.3.2 Granularidad adecuada

La granularidad define el nivel de detalle que tenemos en las dimensiones y en los hechos. Cuando hablamos de nivel de detalle nos referimos a la posible existencia de jerarquías, o también, a métricas (propiedades de los hechos) que no son coherentes con las dimensiones desde las cuales se quiere analizar el hecho. Por ejemplo, en nuestro proyecto, y con en el hecho que hemos planteado anteriormente, no tendría sentido disponer de una medida del precio medio anual por zona, puesto que como se verá ahora, el tiempo, concretado en un período de un mes, es una de nuestras dimensiones. La granularidad del hecho debe quedar definida por el nivel de detalle de la intersección de las dimensiones.

Así pues, es muy importante no mezclar diferentes niveles de granularidad en los hechos, pues esto puede conllevar a generar confusión en los usuarios que vayan a explotar el almacén de datos, y a poder generar resultados erróneos. Debemos tenerlo presente a la hora de diseñar la tabla de hechos.

Donde sí se aprecia claramente la existencia de niveles de jerarquías es en la dimensión Zona (los hemos denominado *provincia / municipio / área / distrito*). A esto hay que añadir que hemos incorporado un nivel superior más denominado *país*. Además, el cliente nos ha advertido que desea poder mantener los cinco niveles de jerarquía a la hora de explotar la información. Esto impacta en la decisión del modelo a adoptar. Se ha optado por crear una única tabla *zona* con la jerarquía de los niveles.

6.3.3 Identificación de las dimensiones y sus atributos

Las dimensiones son las diferentes perspectivas desde las que deseamos analizar nuestro hecho (situación mensual de los inmuebles). Nos sirven para agrupar, filtrar u ordenar la información. Cabe destacar que hemos considerado el número de habitaciones como una dimensión más, pues se trata de un concepto por el que deberemos ordenar y

agrupar. Del mismo modo podríamos haber considerado así los metros cuadrados, sin embargo, el hecho de que nos soliciten metros cuadrados máximos, mínimos y medios nos ha llevado a definirla conceptualmente como métrica, aunque bien podría haber sido una dimensión más.

Por tanto, las dimensiones que hemos definido para el proyecto son:

- **Período temporal**
- **Zona geográfica**
- **Tipo Inmueble**
- **Número de Habitaciones**

Los atributos que encontramos en cada una de ellas son los siguientes.

- Período Temporal:
 - Año
 - Mes
- Zona geográfica:
 - País
 - Provincia
 - Municipio
 - Área
 - Distrito
- Tipo Inmueble:
 - Tipo. Valores posibles (piso, chalets, estudios, áticos y dúplex)
- Número de Habitaciones:
 - Número de habitaciones

Se verá más adelante (en el diseño físico) que para esta última dimensión no hemos considerado conveniente disponer de una tabla real que almacene los diferentes valores, sino integrar el número de habitaciones de cada inmueble como una propiedad más de la tabla de hechos.

6.3.4 Métricas e indicadores

Métrica e Indicador son dos conceptos sinónimos, si bien, el primero lo usamos más desde un punto de vista técnico, y el segundo, más en términos de negocio. Se trata de los conceptos, o medidas de negocio, que hemos detectado que son necesarios disponer de ellos en la tabla de hechos para cumplir con los requisitos finales, y que quedan determinados (granularidad adecuada) por las dimensiones. Son las propiedades de los hechos. Se han dispuesto las siguientes métricas:

- **Precio actual.** El precio del mes correspondiente
- **Precio mes anterior**
- **Precio Venta.** Mientras el inmueble esté en oferta este atributo será cero. Quedará informado el mes en el que se haya vendido el inmueble
- **Meses en oferta.** Será el número de meses que lleva el inmueble en oferta

- o **Metros cuadrados del inmueble**

Adicionalmente, de cara a la preparación de forma más sencilla de alguno de los informes requeridos donde se pide una lista de inmuebles, y por si a futuro resulta interesante filtrar por el atributo planta, a pesar de desnormalizar la tabla de hechos, cosa que por otro lado es frecuente en los modelos de datos para almacenes de datos, incorporaremos como atributos de la tabla de hechos también:

- o **Referencia Inmueble**
- o **Planta**

Y algunos indicadores más que hemos creído interesante:

- o **Precio por metro cuadrado.** Será el cociente entre el precio del inmueble de ese mes y los metros cuadrados del mismo
- o **Inmueble en oferta².** Indicador que determinará si en ese mes el inmueble está en oferta
- o **Inmueble en venta².** Indicador que determinará si en ese mes el inmueble está en venta
- o **Tipo según metros cuadrados.** Indicador usado para categorizar la variable metros cuadrados del inmueble
- o **Tipo según número de habitaciones.** Indicador usado para categorizar la variable número de habitaciones del inmueble

6.3.5 Modelo conceptual

El modelo conceptual es un diagrama en estrella donde la tabla de hechos está en el centro y se relaciona con las dimensiones desde las que será analizada en cardinalidad (1:m), y en donde, hemos querido también reflejar la entidad Inmueble, a pesar de no ser dimensión, por su importancia como entidad y porque será una tabla del diseño físico final.

(2): Estas características se hubieran podido implementar mediante un único campo pues son datos excluyentes, pero, por motivos de sencillez en la creación de los informes, y otros derivados de las correcciones a realizar detectadas se ha decidido así.

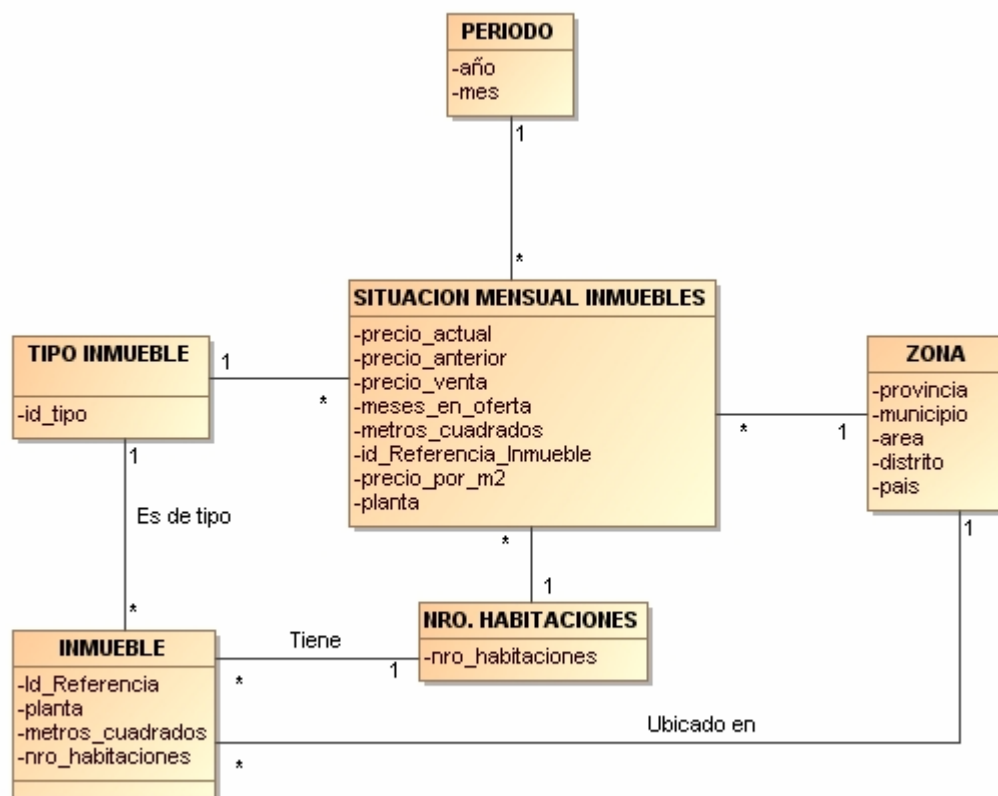


Figura 9. Modelo conceptual del Almacén de datos.

7. Diseño

7.1 Diagrama de la arquitectura software

El trabajo se ha desarrollado sobre una máquina virtual *Oracle VM VirtualBox* con los siguientes componentes instalados.

- Sistema gestor de bases de datos Oracle DataBase Express Edition 10g Release 2 (10.2)
- Oracle SQL Developer 1.0.015 (Oracle Corporation)
- Atlas SBI Studio (Business Intelligence Studio)
- Atlas SBI Server
- Microsoft SQL Server 2005 Express Edition

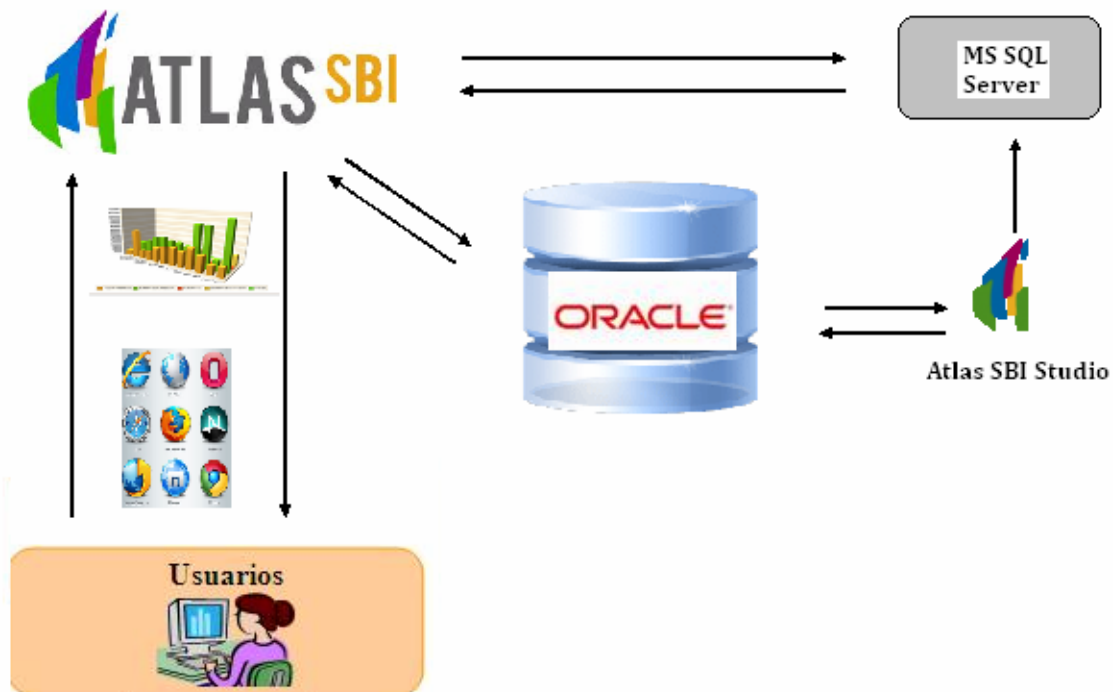


Figura 10. Arquitectura software del almacén de datos.

El almacén de datos se ha construido sobre Oracle. La herramienta Atlas SBI Studio accede a MS SQL Server para guardar los documentos analíticos creados para este proyecto, y al almacén, para poder comprobar las consultas definidas. A su vez, la herramienta de BI Atlas SBI Server, accede a recuperar los documentos analíticos de sus metadatos almacenados sobre SQL Server, a Oracle, para recuperar del almacén los datos de cada consulta definida en el documento solicitado, y devuelve al usuario el resultado mediante una sesión desde un navegador.

7.2 Diagrama de la arquitectura hardware

Entendemos que los usuarios del almacén serán personas del equipo directivo o responsables de departamentos, y que por tanto, no demasiadas personas accederán a consultar los documentos analíticos desarrollados. La propia esencia de los almacenes de datos, y su función de generadores de conocimiento para la toma de decisiones a través de herramientas de BI, hacen que en principio no sea necesario prever un gran número de accesos concurrentes.

Así pues, en este contexto, la arquitectura más apropiada es la de disponer de un servidor en red donde estará ubicado Oracle y el almacén de datos. Dispondremos de otro servidor de aplicaciones donde se instalarán los dos productos Atlas SBI. Los usuarios con acceso a los documentos analíticos se conectarán por medio de la red. No se prevén conexiones más allá de la red local. El servidor Oracle debe preverse que pueda ser fácilmente ampliable en memoria para que pueda servir de mejor manera las consultas solicitadas, mantener los sistemas de indexación, cursores y páginas caché.

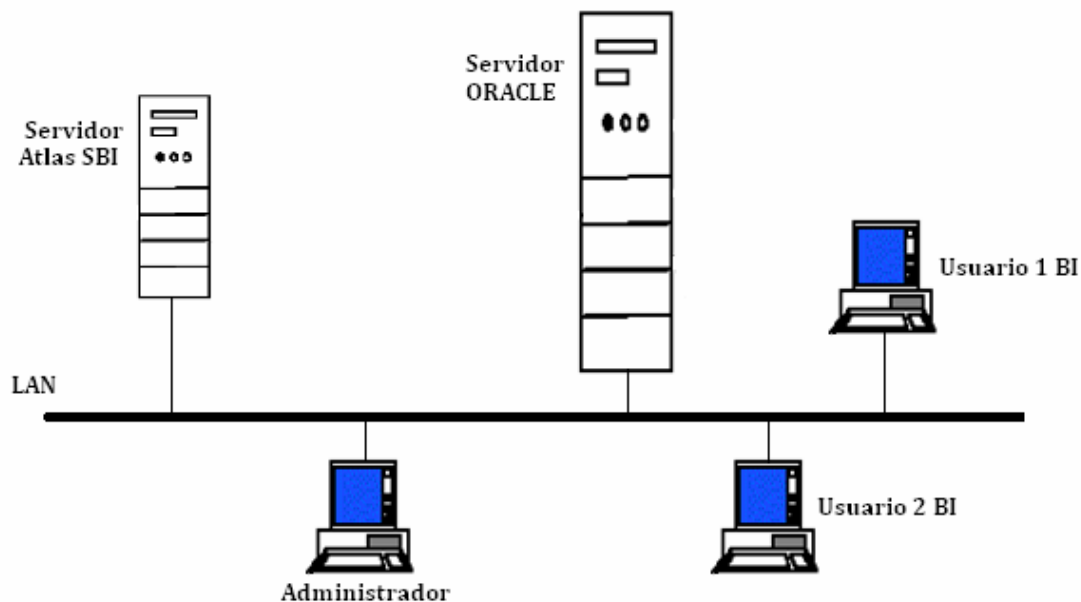


Figura 11. Diagrama de la arquitectura hardware del almacén de datos

7.3 Diseño de la base de datos y diagrama del modelo físico

A continuación, se describen las tablas del modelo de datos, así como un diagrama del diseño físico, del almacén. En las tablas siguientes la columna “*Tipo*” admite los valores “A”, referido a campos alfanuméricos, o “N” para campos numéricos. La columna “*Lon*” es la longitud del campo, y la columna “*Dec*”, indica los decimales en campos numéricos. La columna “*Campo*” tiene la descripción del campo. Se muestran en **negrita** los campos que formarán parte de la clave primaria de cada tabla. Dispondremos de una tabla de hechos, tres tablas de dimensiones, una tabla para la entidad inmueble y otra para la entidad meses.

Tabla de Dimensión **Zonas** (*zonas*).

| Campo | Tipo | Lon | Dec | Observaciones |
|--------------------|------|-----|-----|--|
| País | A | 100 | | De momento, contendrá España o Andorra |
| Provincia | A | 100 | | Descripción provincia |
| Municipio | A | 100 | | Descripción municipio |
| Área | A | 100 | | Descripción área |
| Distrito | A | 100 | | Descripción distrito |
| Latitud provincia | N | 20 | 15 | |
| Longitud provincia | N | 20 | 15 | |

Figura 12. Tabla de zonas.

Tabla de Dimensión **Períodos** (*periodos*).

| Campo | Tipo | Lon | Dec | Observaciones |
|------------|------|-----|-----|---------------|
| Año | N | 4 | 0 | |
| Mes | N | 2 | 0 | |

Figura 13. Tabla de periodos.

Tabla de Dimensión **Tipos de inmuebles** (*tipo inmueble*).

| Campo | Tipo | Lon | Dec | Observaciones |
|-------------|------|-----|-----|---------------|
| Tipo | A | 40 | | |

Figura 14. Tabla de tipos de inmueble.

Tabla de Entidad **Inmueble** (*inmuebles*).

| Campo | Tipo | Lon | Dec | Observaciones |
|--------------------|------|-----|-----|------------------------------------|
| Referencia | A | 15 | | Código de referencia del inmueble. |
| Planta | A | 5 | | |
| Numero Dormitorios | N | 5 | 0 | |
| Metros cuadrados | N | 10 | 0 | |

Figura 15. Tabla de inmuebles.

Tabla de Entidad **Meses** (*meses*).

| Campo | Tipo | Lon | Dec | Observaciones |
|--------------|------|-----|-----|---------------|
| IdMes | N | 2 | 0 | |
| Nombre mes | A | 10 | | |
| Nombre corto | A | 3 | | |

Figura 16. Tabla de meses.

Tabla de Hecho **Situación Mensual de Inmuebles** (*Situ Mes Inmueble*).

| Campo | Tipo | Lon | Dec | Observaciones |
|-----------------------------------|------|-----|-----|--|
| Año | N | 4 | 0 | Clave foránea de tabla de períodos |
| Mes | N | 2 | 0 | Clave foránea de tabla de períodos |
| País | A | 40 | | Clave foránea de tabla de zonas |
| Provincia | A | 100 | | Clave foránea de tabla de zonas |
| Municipio | A | 100 | | Clave foránea de tabla de zonas |
| Área | A | 100 | | Clave foránea de tabla de zonas |
| Distrito | A | 100 | | Clave foránea de tabla de zonas |
| Tipo | A | 40 | | Clave foránea de tabla de tipos de inmueble |
| Referencia inmueble | A | 15 | | Clave foránea de tabla de inmuebles |
| Numero habitaciones | N | 5 | 0 | |
| Metros cuadrados | N | 10 | 0 | |
| Planta | A | 5 | | |
| Precio actual | N | 15 | 0 | Precio en este mes |
| Precio anterior | N | 15 | 0 | Precio mes anterior |
| Precio venta | N | 15 | 0 | Informado si se ha vendido en este mes |
| Meses en oferta | N | 5 | 0 | Número de meses en oferta |
| Precio metro cuadrado | N | 10 | 0 | Precio actual / metros cuadrados |
| En oferta | A | 1 | | Indicativo de si el inmueble está o no en oferta. ('1' → Si / '0' → No) |
| Vendido | A | 1 | | Indicativo de si el inmueble está o no vendido. ('1' → Si / '0' → No) |
| Tipo según metros cuadrados | A | 1 | | Para categorizar inmuebles por metros cuadrados: Desconocido → 0 (Hasta 70 m2) → 1 (>70 y <=140 m2) → 2 (>140 y <=300 m2) → 3 (>300 m2) → 4 |
| Tipo según número de habitaciones | A | 1 | | Para categorizar inmuebles por número de habitaciones: Desconocido → 0 (<=2) → 1 (>2 y <=4) → 2 (>4 y <=9) → 3 (>9) → 4 |

Figura 17. Tabla de situación de inmuebles.

Este es el diagrama físico de relación entre tablas.

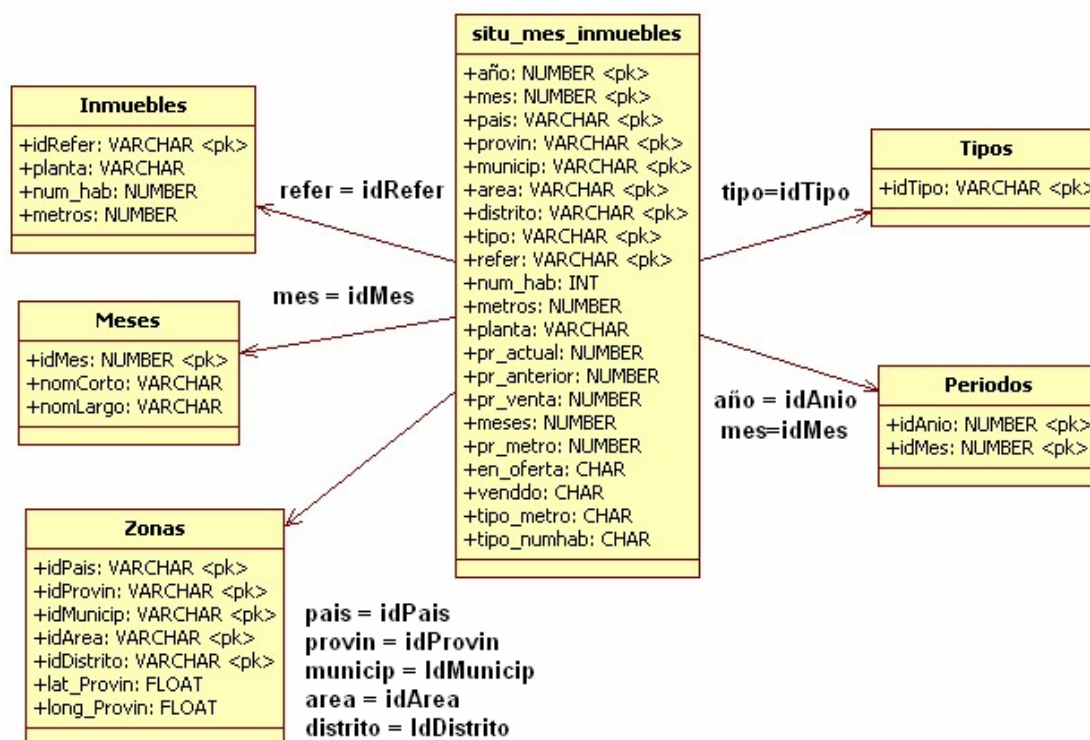


Figura 18. Modelo de relaciones de la base de datos.

7.4 Proceso ETL de carga y generación de la base de datos

Se han preparado dos procesos diferentes de extracción, transformación y carga. Uno es **para la carga del almacén por primera vez, o bien, para regenerarlo**, cuando así se necesite, y otro, **para el mantenimiento periódico evolutivo** que el cliente estime. En ambos casos, la estructura final del proceso ETL se divide en subprocesos cuyas tareas principales quedan recogidas en este esquema.



Figura 19. Fases del proceso ETL.

A continuación, se detalla el funcionamiento de los procesos de carga desarrollados.

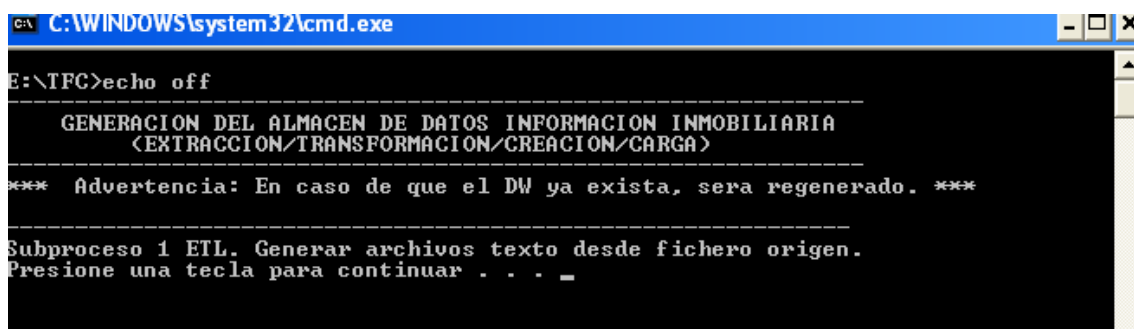
Proceso de Carga del Almacén (carga inicial o regeneración) “GeneraDW.bat”.

- Este ejecutable se entrega en la raíz del proyecto (ruta *E:\TFC*). Puede verse su código en el **anexo 2**. Para lanzarlo, simplemente, se deberá hacer doble *click* sobre él, o bien, llamarlo desde línea de comandos. Una vez ejecutado, el proceso solicita que se vaya pulsando INTRO conforme va avanzando por las diferentes etapas o subprocesos que lo componen, y que, en algún momento, pueden generar mensajes de error controlados que a continuación explicamos junto al detalle de las fases por las que pasa.
- Se ha probado la ejecución del proceso (de hecho, la MV se entrega ya cargada) sobre un procesador de características *Pentium 4 CPU 3.00 GHz* y 512 MB de RAM, que es la memoria con la que se definió la MV, y el tiempo aproximado que dura el proceso en total son unos 10 minutos.

Al ejecutar el proceso de carga, éstas son las fases por las que se pasa:

1.- Subproceso 1 ETL. Extracción de datos del fichero Origen “Catalogo Un piso para todos.txt” y obtención de los ficheros “NewCatalogo.txt”, “NewPrecios.txt” y “NewInmueblesVendidos.txt”.

Nada más ejecutar el proceso por lotes el sistema nos muestra esta pantalla de presentación y advertencia, indicándonos que el almacén será regenerado en caso de continuar con la ejecución. Además, se informa que se va a lanzar la primera fase del mismo, que consiste en extraer los datos del fichero origen y cargarlos normalizados y transformados sobre otros tres ficheros texto temporales. Esta acción la realiza una clase Java denominada “*TransformacionPrevia.class*” que se entrega en la ruta *E:\TFC\bin*.



```
C:\WINDOWS\system32\cmd.exe
E:\TFC>echo off
-----
GENERACION DEL ALMACEN DE DATOS INFORMACION INMOBILIARIA
(EXTRACCION/TRANSFORMACION/CREACION/CARGA)
-----
*** Advertencia: En caso de que el DW ya exista, sera regenerado. ***
-----
Subproceso 1 ETL. Generar archivos texto desde fichero origen.
Presione una tecla para continuar . . . _
```

Figura 20. Primer subproceso ETL.

Al término de este subproceso se habrán generado los siguientes archivos temporales (al finalizar el proceso ETL serán eliminados):

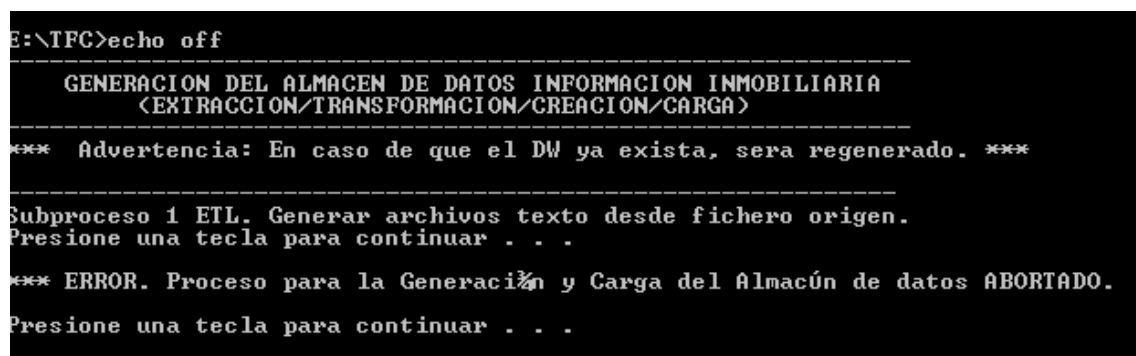
- Fichero “NewCatalogo.txt”, que contendrá filas de inmuebles en las cuales se ha eliminado el valor del campo “zona” de los registros del fichero original, así como los precios de los diferentes periodos, y se habrá incorporado por el final la jerarquía de la zona a la que pertenecen ya normalizada

- Fichero “NewPrecios.txt”, que contendrá la referencia del inmueble, el año y mes al que pertenece el precio, y el precio. Tendrá un registro por cada período con precio informado para cada inmueble. Se ha decidido estructurar de esta forma (más filas) para evitar que en futuras cargas pueda resultar un problema el hecho de que pudieran venir más de 25 meses con precios
- Fichero “NewInmueblesVendidos.txt”, que tendrá un registro por cada inmueble vendido. Contendrá la referencia del inmueble, año y mes de la venta, así como el precio de venta

Estos archivos se crean en el directorio raíz del proyecto, y en ellos, todos los datos estarán separados por el carácter punto y coma (“;”) para facilitar los siguientes pasos. Al finalizar esta fase, el procedimiento muestra un mensaje de finalización correcta y solicita que el usuario pulse INTRO para continuar. En caso de producirse algún error en la clase Java se generará un fichero de incidencias en el directorio raíz denominado “Incidencias.txt”. Las incidencias controladas son:

- El proceso se debe ejecutar pasándole 3 parámetros: año y mes de inicio de los precios³, y tipo de proceso (1 = carga inicial / 2 = carga mensual)
- Fichero de entrada “Catalogo Un_piso_para_todos.txt” no encontrado
- Problemas en la generación de fichero “NewCatalogo.txt”
- Problemas en la generación de los ficheros de precios

Estos errores anteriores suponen que el proceso **se aborte en su totalidad**, informando antes al usuario de ello. En este caso, **deben ponerse en contacto con nosotros** para resolver el problema.



```
E:\TFC>echo off
-----
GENERACION DEL ALMACEN DE DATOS INFORMACION INMOBILIARIA
(EXTRACCION/TRANSFORMACION/CREACION/CARGA)
-----
*** Advertencia: En caso de que el DW ya exista, sera regenerado. ***
-----
Subproceso 1 ETL. Generar archivos texto desde fichero origen.
Presione una tecla para continuar . . .
*** ERROR. Proceso para la Generación y Carga del Almacén de datos ABORTADO.
Presione una tecla para continuar . . .
```

Figura 21. Mensaje de proceso abortado

También puede producirse el siguiente aviso que no hace abortar el proceso:

- Existen más de 5 grupos territoriales en jerarquía de zonas (en este caso, al igual que cuando hay exactamente 5 zonas, ya anunciamos en la fase anterior que nos quedamos con primera, tercera, cuarta y quinta, pero se advierte de ello no obstante).

(3): es importante destacar que la clase Java recibe por parámetros el año y mes de inicio de los precios del fichero original y un parámetro que indica el tipo de proceso (1 = Carga Inicial / 2 = Carga Mensual). Estos parámetros los pasa automáticamente este proceso ETL por lotes, por tanto, si alguna vez se tuviera que usar este proceso de creación o regeneración de nuevo, con un fichero que tuviera diferente inicio de precios a enero de 2006, se deben modificar los parámetros de la llamada sobre E:\TFC\GeneraDW.bat.

Al finalizar este subproceso se muestra el siguiente mensaje:

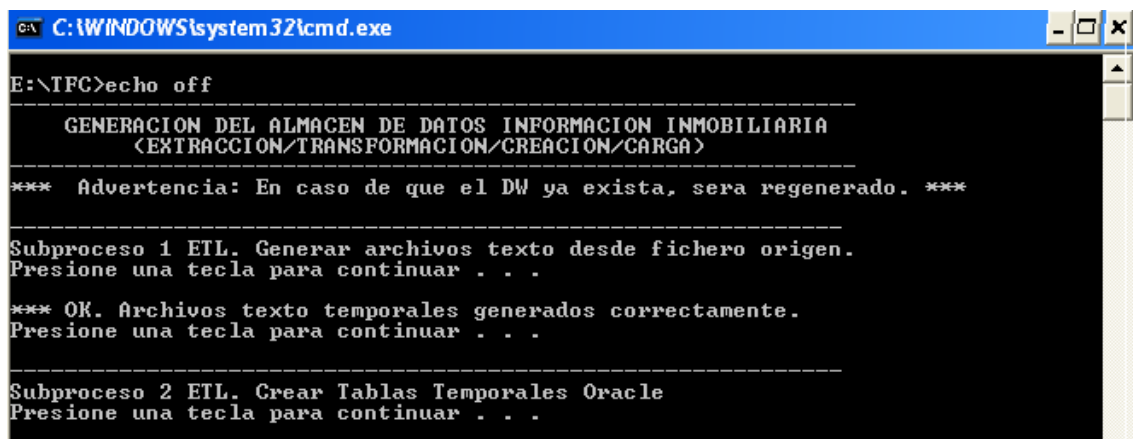
```
*** OK. Archivos texto temporales generados correctamente.
Presione una tecla para continuar . . .
```

Figura 22. Mensaje de finalización de primer subproceso ETL.

Debido a la extensión del código fuente de la clase Java que realiza este paso hemos pensado que no tiene sentido incorporarlo en este informe como un anexo más. Pueden encontrar su código fuente en la ruta *E:\TFC\java* y su documentación en la carpeta *E:\TFC\javadoc*.

2.- Subproceso 2 ETL. Crear Tablas Temporales Oracle.

Al pulsar INTRO tras el anterior mensaje de finalización correcta, se muestra aviso anunciando la segunda de las etapas del proceso, consistente en lanzar un *script* PL/SQL que borra, en caso de existir, las tres tablas temporales Oracle utilizadas, y se crean nuevamente. Esta acción se realiza con el usuario *tfc* para que las tablas queden bajo su esquema. El bloque sql se entrega en la ruta *E:\TFC\pl_sql* y se denomina “*Crear_Tablas_Temp.sql*” (ver anexo 3).



```
C:\WINDOWS\system32\cmd.exe
E:\TFC>echo off

-----
GENERACION DEL ALMACEN DE DATOS INFORMACION INMOBILIARIA
(EXTRACCION/TRANSFORMACION/CREACION/CARGA)
-----
*** Advertencia: En caso de que el DW ya exista, sera regenerado. ***
-----
Subproceso 1 ETL. Generar archivos texto desde fichero origen.
Presione una tecla para continuar . . .
*** OK. Archivos texto temporales generados correctamente.
Presione una tecla para continuar . . .
-----
Subproceso 2 ETL. Crear Tablas Temporales Oracle
Presione una tecla para continuar . . .
```

Figura 23. Segundo subproceso ETL.

Las tres tablas a crear tienen los nombres: “*inmuebles_tmp*”, “*precios_tmp*” e “*inmueblesvendidos_tmp*”. En todo caso, antes de finalizar el proceso ETL, serán eliminadas.

El proceso muestra un mensaje por cada tabla borrada o creada. Si las tablas temporales a borrar no existen se muestra aviso de que no se han podido borrar, pero se continúa sin problemas. En realidad, esto ocurre siempre, puesto que son borradas al finalizar. Sin embargo, **por si acaso alguna vez no se completase el proceso y no quedarán eliminadas**, hemos preferido dejar las sentencias de borrado aunque generen el error de no existencia de la tabla a eliminar.

Al finalizar este subproceso, se muestra el siguiente mensaje:

```
*** OK. Tablas temporales creadas correctamente.  
Presione una tecla para continuar . . .
```

Figura 24. Mensaje de finalización de segundo subproceso ETL.

3.- Subproceso 3 ETL. Cargar Tablas Temporales Oracle.

Una vez finalizado el paso anterior es necesario informar las tablas temporales creadas desde los archivos texto que generamos en el primer paso. De ello se encarga SQL *Loader, el cual, invocamos en tres ocasiones en esta etapa con el usuario tfc y con sus correspondientes archivos de control (“inmuebles.ctl”, “precios.ctl” e “inmueblesvendidos.ctl”). Estos archivos se entregan en la ruta *E:\TFC\ctl* (**anexo 4**).

```
*** OK. Tablas temporales creadas correctamente.  
Presione una tecla para continuar . . .  
  
-----  
Subproceso 3 ETL. Cargar Tablas Temporales Oracle  
Presione una tecla para continuar . . . _
```

Figura 25. Tercer subproceso ETL.

Al finalizar esta carga se habrá generado un archivo log por cada una de las tres tablas cargadas en la ruta *E:\TFC\log*.

- En caso de que se rechace algún registro se generará algún fichero en la ruta *E:\TFC\bad*
- En caso de encontrar algún registro que no concuerda con el criterio de carga especificado en los ficheros de control se generará un fichero en la ruta *E:\TFC\dsc*
- Previamente se eliminan los ficheros .log, .dsc o .bad que pudieran existir en estos directorios por anteriores ejecuciones
- Al procesar el fichero origen que el cliente nos ha suministrado, los datos de carga que deben generarse en los archivos log son:
 - Tabla “inmuebles_tmp”. 92.555 filas cargadas correctamente
 - Tabla “precios_tmp”. 1.091.388 filas cargadas correctamente
 - Tabla “inmueblesvendidos_tmp”. 43.258 filas cargadas correctamente

En pantalla van apareciendo los mensajes que genera SQL *Loader por cada registro cargado. Al finalizar la carga de cada tabla se mostrará mensaje indicando que ha acabado correctamente, y se nos pedirá confirmación para continuar con la siguiente fase.

4.- Subproceso 4 ETL. Crear Tablas Definitivas Oracle.

Al pulsar INTRO tras el mensaje de finalización correcta anterior se muestra aviso anunciando el lanzamiento del *script* PL/SQL que borra, en caso de existir, las seis tablas definitivas del modelo, y las genera de nuevo. Esta acción se realiza con el usuario tfc para que las tablas queden bajo su esquema. El bloque sql se entrega en la ruta *E:\TFC\pl_sql* y se denomina “*Crear_Tablas_Definitivas.sql*” (**ver anexo 5**).


```

*** OK.Carga de tabla temporal de Ventas de Inmuebles realizada correctamente.
Presione una tecla para continuar . . .

-----
Subproceso 4 ETL. Crear Tablas Definitivas Oracle
Presione una tecla para continuar . . .

```

Figura 26. Cuarto subproceso ETL.

Las tablas a crear son las Dimensiones: “zonas”, “periodos” y “tipo_inmueble”, el Hecho “situ_mes_inmuebles”, y las entidades “inmuebles” y “meses”. El proceso mostrará mensaje por cada tabla borrada o creada. Si las tablas a borrar no existen (primera ejecución, por ejemplo), se muestra aviso de que no se han podido borrar, pero se continúa sin problemas.

5.- Subproceso 5 ETL. Cargar Tablas Definitivas Oracle.

En este subproceso final se realiza la inserción de los registros sobre las tablas del modelo del almacén. Consiste en lanzar dos script PL/SQL que se entregan en la ruta *E:\TFC\pl_sql*, denominados “Carga_Modelo_Datos_1.sql” y “Carga_Modelo_Datos_2.sql” (ver anexo 6). El primero de ellos informa las tablas de dimensiones y entidades, mientras que el segundo, carga la tabla de hechos.

Antes de lanzarse esta fase, el proceso avisa de que esta carga puede demorarse unos 10 minutos. En realidad, los tiempos empleados en todas las pruebas realizadas han sido ligeramente menores (7 u 8 minutos).

```

*** OK. Tablas definitivas creadas correctamente.
Presione una tecla para continuar . . .

-----
Subproceso 5 ETL. Cargar Tablas Definitivas Oracle
***¡Atención! Este proceso puede tardar varios minutos (10 min. aprox). ***
Presione una tecla para continuar . . .

```

Figura 27. Quinto subproceso ETL.

Una vez pulsado INTRO el proceso lanza automáticamente los dos *script*. El primero de ellos, la carga de entidades y dimensiones, tarda menos de 1 minuto y consiste en rellenar las tablas de tipos de inmuebles, períodos y meses mediante *inserts* de valores distintos de las tablas temporales, y las tablas de zonas e inmuebles, mediante bucles de lectura de dos cursores que apuntan a las tablas temporales. El verdadero coste temporal de esta fase está en la segunda carga, la de la tabla de hechos, porque se hace desde la lectura por bucle de la tabla de precios, que recordemos contiene muchos registros (más de 1 millón), y por la necesidad de calcular las métricas e indicadores.

Al finalizar, se lanza mensaje mostrando que el subproceso ha terminado correctamente, y se informa al usuario de la eliminación de los archivos y tablas temporales.

```

*** OK. Carga de tablas Definitivas realizada correctamente.
Ahora se van a eliminar Archivos texto y Tablas temporales.
Presione una tecla para continuar . . .

```

Figura 28. Mensaje de finalización de quinto subproceso ETL.

Una vez pulsado INTRO, el proceso elimina dichos objetos y muestra mensaje antes de finalizar.

Proceso de Carga Mensual Evolutiva del Almacén “Carga Mensual.bat”.

- Este ejecutable se entrega en la raíz del proyecto (ruta *E:\TFC*). Puede verse su código en el **anexo 7**. Para lanzarlo, simplemente, se deberá hacer doble *click* sobre él, o bien, llamarlo desde línea de comandos. Una vez ejecutado, el proceso solicita que se vaya pulsando INTRO conforme va avanzando por las diferentes etapas o subprocesos que lo componen
- Este proceso para el mantenimiento evolutivo está pensado para las cargas mensuales de los precios de los inmuebles. El proceso es válido tanto si se reciben datos correspondientes a un único mes, como si son de más de un mes
- Es importante destacar que el fichero origen a recibir debe cumplir ciertas pautas convenidas con el cliente:
 - Su formato es el mismo que el utilizado en la carga inicial. La única diferencia es el número de columnas relativas a meses, aunque ya hemos indicado que pueden recibirse más de una columna por si el cliente decide realizar el mantenimiento del almacén no cada mes, sino con otra frecuencia
 - Al igual que se cumplía en el fichero original, si un inmueble se vendió el mes pasado, este mes debe venir sin precio
 - Es necesario grabar en el fichero mensual datos de todos los inmuebles del catálogo, independientemente de si han variado o no su precio, o si se han vendido en ese mes. Se ha pensado de esta forma, pues de otra manera, obligaría a que el proceso de carga mensual tuviera que acceder a la tabla de hechos por cada período e inmueble para el que no llegase información, con el objetivo de recuperar ciertos datos a cargar en el nuevo registro, y esto no parece óptimo. No obstante, el proceso tiene una parte al final que se entrega comentada por si se decide cambiar este criterio, si bien, este código debería optimizarse
 - Se entrega en la raíz del proyecto un fichero de pruebas denominado “Catalogo_Mensual.txt”. No hemos incluido todos los inmuebles y precios que se repiten, sino, simplemente, algunos registros para realizar las pruebas (algún inmueble nuevo, alguno vendido, alguno que repite precio y alguno que cambia de precio)

En realidad, este proceso de mantenimiento, es muy similar al descrito anteriormente. Las fases por las que pasa son:

- Subproceso 1 ETL. Extracción de datos del fichero mensual y obtención de los ficheros temporales texto.

Esta fase es completamente igual que la descrita anteriormente. Las únicas diferencias son que, en este caso, se solicita por pantalla el año y mes de inicio de los datos del fichero que vamos a procesar.

```
E:\TFC>echo off

-----
CARGA MENSUAL ALMACEN DE DATOS INFORMACION INMOBILIARIA
-----

Subproceso 1 ETL. Generar archivos texto desde fichero origen.
Presione una tecla para continuar . . .
Introduzca el Año de inicio de los datos:2008
Introduzca el Mes de inicio de los datos:02
Año tecleado: 2008
Mes tecleado: 02

*** OK. Archivos texto temporales generados correctamente.
Presione una tecla para continuar . . . _
```

Figura 29. Proceso ETL de mantenimiento evolutivo.

Una vez tecleados estos datos, se llama internamente a la misma clase Java pasándole como parámetros los datos de año y mes de inicio, y además, el parámetro del tipo de proceso con el valor “2” de carga mensual.

Otra diferencia en este primer paso, con respecto a la carga inicial, es que, en carga mensual, no se genera el archivo temporal de inmuebles vendidos. Este archivo tenía sentido en el contexto de la carga inicial pues servía de apoyo para, aprovechando la lectura de los precios de los meses, guardarnos en ese mismo momento el mes de la venta y no tener que recuperarlo después mediante accesos que penalizasen el proceso de carga. En este caso, como el número de accesos es mucho menor, pues sólo se hace por los inmuebles vendidos en los meses para los que viene información (normalmente sólo un mes), se ha decidido obtener dicho dato accediendo a la base de datos.

- Los subprocesos 2 y 3, referentes a la creación de las tablas Oracle temporales, y su carga, son exactamente iguales, con la única salvedad que ahora no se rellena la tabla temporal Oracle de inmuebles vendidos tampoco.
- El siguiente subproceso 4 corresponde a la carga mensual de las tablas del *Data Warehouse*.

```
*** OK. Carga de tabla temporal de Precios realizada correctamente.
Presione una tecla para continuar . . .

-----

Subproceso 4 ETL. Carga Mensual Tablas DW
Presione una tecla para continuar . . .
```

Figura 30. Subproceso cuarto ETL de la carga mensual de mantenimiento.

En este caso, consiste en lanzar un único script PL/SQL denominado “*Carga_Mensual_1.sql*” (**ver anexo 8**), que se entrega en la ruta *E:\TFC\pl_sql*, y que se encarga de alimentar las tablas del almacén con los nuevos datos que llegan.

La tabla periodos se alimenta mediante los distintos valores nuevos de año y mes que nos llegan. El resto de dimensiones, así como la tabla de hechos, son cargadas desde la lectura que se hace de cada inmueble para el que se reciben datos, verificando, en el caso de las dimensiones, si existen o no los nuevos valores, y actualizando la situación de los inmuebles desde la lectura de los precios recibidos asociados a cada inmueble leído. En caso de ser un inmueble

nuevo, se usan variables de trabajo para llevar el control de los meses en oferta y el precio del mes anterior, y en caso de ya existir, se recuperan esos datos del último registro grabado en la tabla para el inmueble. En ambos casos, si algún precio leído es cero, se busca el último registro grabado para actualizar el precio de venta de aquel mes.

Nota importante: La MV se entrega con el almacén cargado únicamente con los datos originales proporcionados por el cliente. No se incluyen, lógicamente, estos datos de pruebas usados para verificar el proceso de carga de mantenimiento.

8. Informes implementados: Diseño y descripción

8.1 Correcciones realizadas sobre proceso de carga

El cliente nos advirtió que el fichero original podía contener errores. Durante el trabajo en esta fase se han detectado **ciertas incidencias** en los datos que queremos informar junto a la solución que hemos adoptado.

- El inmueble de referencia “VP0000000647987” tiene 77895 metros cuadrados y un precio de 1.563.000 euros. El hecho de que sea una cifra de metros cuadrados tan elevada (máxime para ser un piso) hace que el precio por metro cuadrado sea de 20 euros, cuando en su misma zona, salvo 2 pisos, todos superan los 2000 euros por metro cuadrado. Por tanto, parece un error la cifra de metros cuadrados. Creemos que puede estar multiplicada por 100. Así pues, se pone en conocimiento del cliente por si hay que corregirlo. Para la carga nos ha parecido como mejor solución actualizar la cifra de 77895 a 779 metros cuadrados. Lo mismo ocurre con el inmueble de referencia “VP0000000874346”, que trae 44700 metros cuadrados y un precio de 960.200. Hemos actualizado sus metros cuadrados a 447.
- Creemos que llegan errores importantes en algunos de los precios informados. En ciertos inmuebles ocurre que varía mucho el precio de un mes a otro. Se ha hecho una búsqueda orientativa de dichos errores por medio de la sentencia:

```
select * from situ_mes_inmuebles where
((precio_act > (2 * precio_ant)) OR
(precio_ant > (2 * precio_act))) AND
precio_ant > 0
```

Es decir, buscar meses en los que los inmuebles cambian su precio con respecto al mes anterior el doble, bien al alza, o bien, a la baja.

Se encuentran 45 registros que cumplen con la consulta, lo cual, significa que hay 45 posibles inmuebles afectados, pues aquí solo se recoge 1 registro por cada inmueble y mes en donde se produce la incidencia, la variación del precio al alza o baja. Al analizar caso a caso se comprueba que la mayoría parecen errores claros. En unos casos por tener un cero de más o de menos, como por ejemplo, los inmuebles de referencias: VP0000001023861, VW0000000702300, VC0000000722282. Otros, como la referencia VW0000000826757, parece raro que tenga un salto en un mes de 300.000 euros a 2.335.000 euros, o al contrario, como sucede en éste otro (VC0000000441530), donde el precio cae de 3.925.000 a 695.000 (un 464%). Parece claramente un error. Quizás el precio bueno sea 3.965.000, puesto que se trata de un ático de 522 metros cuadrados. Sin embargo, esta alta variabilidad en los posibles errores, y el **no encontrar un patrón de error**, sino que para corregirlo correctamente deberíamos tomar decisiones junto al cliente, hace que se informe del hecho en esta memoria para intentar solucionarlo en origen y ahora tomemos la decisión de informar a cero todos los precios de cada registro de dichos inmuebles, así como el precio por metro cuadrado. De esta forma no afectarán a los documentos analíticos donde se analicen precios. No borramos los registros porque para otro tipo de informes

nos sirven la ubicación, características, tipo, o meses en oferta que entendemos sí son correctos. Se realiza la corrección mediante esta sentencia SQL.

```
update situ_mes_inmuebles set precio_act = 0, precio_ant = 0,
precio_venta = 0, pr_metro = 0, vendido = 0, en_oferta = 0
WHERE refer IN
(select distinct(refer) from situ_mes_inmuebles where
((precio_act > (2 * precio_ant)) OR
(precio_ant > (2 * precio_act))) AND
precio_ant > 0)
```

- Vienen inmuebles con precios realmente muy bajos. Por ejemplo, esta referencia “VC0000000914039” viene con un precio de 460 euros, o esta otra “VC0000000919174”, de 1.800 euros. Parecen, más bien, cifras mensuales de alquileres. Si se consultan los inmuebles ordenados por precio ascendente, se ve que los precios van en aumento desde 460 euros pasando por casi todas las cifras. Hay un salto de 3000 a 9000 euros, que parece un importe alto para ser alquiler (se trata de un inmueble de 55 metros cuadrados), aunque muy bajo para ser un precio real. Como de nuevo, no se encuentra un patrón, ni lógica, se pone en conocimiento del cliente y se decide poner a cero los precios de los inmuebles que tienen precios por debajo de 8000 euros mediante esta sentencia:

```
update situ_mes_inmuebles set precio_act = 0, precio_ant = 0,
precio_venta = 0, pr_metro = 0,
vendido = 0, en_oferta = 0
WHERE refer IN
(select distinct(refer) from situ_mes_inmuebles where
precio_act < 8000 and precio_act > 0)
```

- En el fichero vienen todo tipo de inmuebles con el dato número de habitaciones a cero. Esto puede tener sentido en el caso de inmuebles de tipo estudios, pero parecen errores claros en el resto de casos. Para evitar que esto sesgue los documentos analíticos donde se analizan los inmuebles por número de habitaciones, se decide informar la columna con el valor -1 para estos casos que no son estudios y tienen cero habitaciones. De esta forma, no entran en la categorización realizada en los documentos analíticos.
- Pensamos que el dato metros cuadrados tampoco es correcto en bastantes casos. Por ejemplo, vienen inmuebles de tipo áticos, chalets, estudios y pisos con menos de 25 metros cuadrados, lo cual, no parece correcto. Si buscamos los diferentes tipos de inmuebles cuyos metros cuadrados están entre 26 y 40, de nuevo, salen todos los anteriores tipos, y además, aparece también dúplex. Tampoco parece tener mucho sentido. Por tanto, creemos que esta anomalía también se debe poner de manifiesto en esta memoria y nosotros, en la carga, para evitar un sesgo importante de los documentos analíticos por esta variable, optamos por informar con el valor -1 los casos de inmuebles con menos de 25 metros cuadrados. De esta forma, no entran en la categorización realizada sobre los documentos analíticos. También se actualiza a cero el campo precio medio por metro cuadrado por razones obvias, no siendo tratados, por ello, en los análisis de precios.
- Por último, simplemente indicar que por cuestiones de calidad de ciertos informes analíticos con desglose por zona, hemos decidido que cuando el municipio, área y distrito nos llegan sin informar, cargamos como municipio el

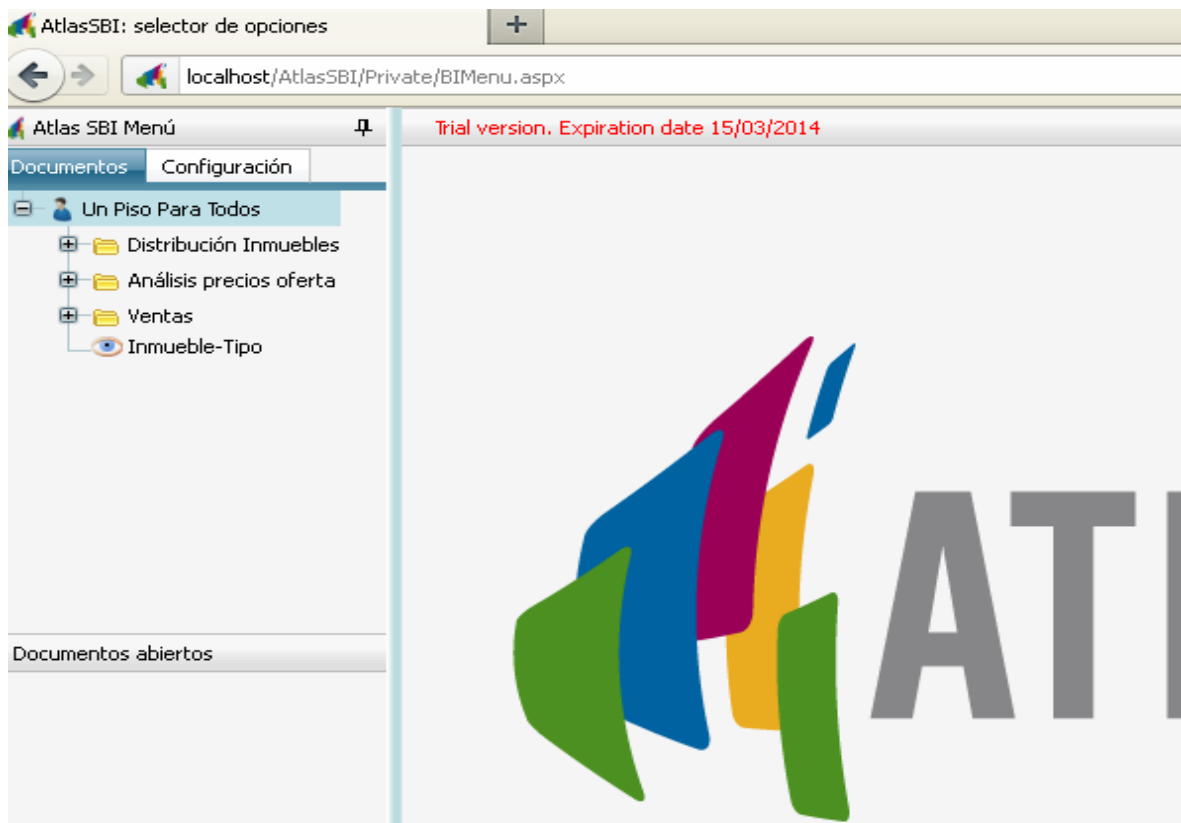
literal ‘Sin informar’. De esta forma, no aparecen en ciertos documentos tablas con registros de desglose por zona en blanco.

Queremos destacar que la sensación es que, **tal y como se nos advirtió**, en el fichero original **vienen muchísimos errores**, y además, no tienen un patrón común, suponemos por las manualidades que nos han indicado que se llevaron a cabo para su confección. Debe revisarse este problema con el cliente y ser subsanado en origen para que los documentos analíticos sean veraces y puedan ayudar a su objetivo: extraer conocimiento.

Todas estas soluciones descritas anteriormente se han implementado dentro del proceso ETL por si se vuelve a generar el almacén de nuevo.

8.2 Informes preparados

Para acceder a los documentos analíticos debemos ir a la página <http://localhost/atlasSBI/> con un programa navegador. En la MV entregada, tanto el navegador *Explorer*, como *Mozilla*, tienen definida dicha página como la de inicio. Una vez que nos hemos acreditado en la página inicial, aparece este menú con los documentos preparados.



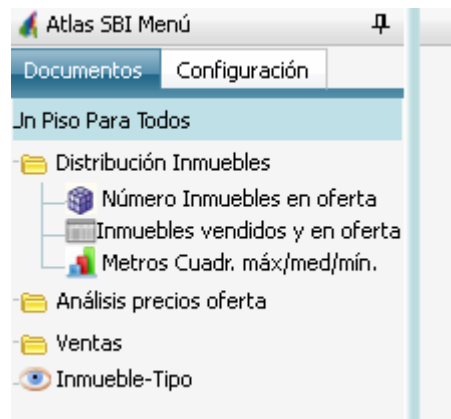
Captura 1. Menú inicial al acceder a Atlas SBI Server.

Se puede apreciar el menú creado en el marco de la izquierda. Se ha intentado agrupar la información solicitada por su relación en estas tres carpetas: “distribución de inmuebles”, “análisis de precios de oferta” y “ventas”. Finalmente, aparece un

documento analítico creado para encontrar el inmueble tipo de cada país. A continuación, vamos a ir desgranando cada uno de los reportes preparados.

8.2.1 Distribución de inmuebles

Al desplegar la carpeta “distribución de inmuebles” nos aparecen tres documentos analíticos: número de inmuebles en oferta, inmuebles vendidos y en oferta, y metros cuadrados máximos, medios y mínimos (captura 2).



Captura 2. Elementos de carpeta de distribución de inmuebles

Los tres documentos analíticos aquí integrados muestran información referente a inmuebles en oferta.

8.2.1.1 Número de inmuebles en oferta

El documento número de inmuebles en oferta responde a los requerimientos del cliente de disponer del **número de inmuebles por zona, tipología y características** y del **número de pisos por número de habitaciones, zona, tipología y características**. Al ejecutarlo, inicialmente, presenta una tabla dinámica con el número de inmuebles en oferta por país y provincia (captura 3), pudiendo filtrar la información por el tipo de inmueble, además de por el resto de dimensiones que aparecen en la cabecera de la tabla. Se puede observar esto en la captura 4, donde hemos clasificado la información, además de por país y provincia (clasificación inicial), por la dimensión metros. Dispone de los selectores año y mes para seleccionar el período al que se refiere la información

Número Inmuebles en oferta

Trial version. Expiration date 15/03/2014

Número de inmuebles en oferta por zona/tipo/características

Año 2006

Mes Enero

MUNICIP

AREA

DISTRITO

TIPO

PLANTA

METROS

NRO. HABITACIONES

NRO. INMUEBLES

Suelta Campos de Columna Aquí

PAIS

PROVIN

NRO. INMUEBLES Total

andorra

andorra

60

alicante

5.321

barcelona

15.520

castellón

1.274

girona

1.613

ibiza-formentera

146

lleida

253

mallorca

1.054

menorca

114

tarragona

1.803

valencia

7.880

españa Total

34.978

Total General

35.038

Captura 3. Tabla dinámica con el número de inmuebles en oferta.

Número Inmuebles en oferta Trial version. Expiration date 15/03/2014

Número de inmuebles en oferta por zona/tipo/características

Año 2006 Mes Enero

| | | | | | | | |
|----------------|------------------|----------|-----------------|----------------|-------------------|------------|---------------|
| MUNICIP | AREA | DISTRITO | TIPO | PLANTA | NRO. HABITACIONES | | |
| NRO. INMUEBLES | | METROS | | | | | |
| PAIS | PROVIN | Desc. | Entre 140 y 300 | Entre 70 y 140 | Hasta 70 | Más de 300 | Total General |
| andorra | andorra | | 5 | 26 | 24 | 5 | 60 |
| españa | alicante | 4 | 936 | 2.859 | 1.300 | 222 | 5.321 |
| | barcelona | 18 | 2.498 | 7.805 | 4.621 | 578 | 15.520 |
| | castellón | | 178 | 698 | 360 | 38 | 1.274 |
| | girona | 6 | 499 | 635 | 329 | 144 | 1.613 |
| | ibiza-formentera | | 34 | 60 | 43 | 9 | 146 |
| | lleida | | 76 | 116 | 43 | 18 | 253 |
| | mallorca | | 258 | 577 | 147 | 72 | 1.054 |
| | menorca | | 26 | 64 | 22 | 2 | 114 |
| | tarragona | | 427 | 894 | 420 | 62 | 1.803 |
| valencia | | 1.550 | 5.003 | 1.058 | 269 | 7.880 | |
| españa Total | | 28 | 6.482 | 18.711 | 8.343 | 1.414 | 34.978 |
| Total General | | 28 | 6.487 | 18.737 | 8.367 | 1.419 | 35.038 |

Captura 4. Número de inmuebles en oferta por país, provincia y tamaño.

Esta captura 4 da pie a explicar la categorización que hemos realizado de la dimensión “metros cuadrados” y “número de habitaciones”. Como se aprecia en la captura 3, para

metros cuadrados se han establecido cinco categorías: “desconocido”, que son los inmuebles que no tienen informado los metros cuadrados, o bien, que tenían menos de 25 y fueron considerados errores, “hasta 70” metros cuadrados, “entre 70 y 140”, “entre 140 y 300”, y “más de 300”. Para el número de habitaciones también hemos planteado cinco categorías: “desconocido”, que incluye los inmuebles que no son estudios y no traían el número de habitaciones informado –error descrito en apartado anterior–, “hasta 2” dormitorios, “de 3 y 4”, “de 5 a 9”, y “más de 9” dormitorios. En otras capturas podremos apreciar estas categorías también.

Gracias a la flexibilidad de la tabla dinámica podemos dar soporte con este mismo documento analítico al requisito de disponer del número de pisos por número de habitaciones, zona, tipo y características. Se puede ver en la siguiente captura donde hemos seleccionado como tipo “pisos”, y clasificamos la información, por ejemplo, por número de habitaciones, país y provincia.

Número Inmuebles en oferta Trial version. Expiration date 15/03/2014

| | | | | | |
|-------------------|-----------|------------------|-------------------------------|--------|--------|
| MUNICIP | AREA | DISTRITO | TIPO | METROS | PLANTA |
| NRO. INMUEBLES | | | Suelta Campos de Columna Aquí | | |
| NRO. HABITACIONES | PAIS | PROVIN | NRO. INMUEBLES Total | | |
| De 3 a 4 | andorra | andorra | 13 | | |
| | españa | alicante | 1.318 | | |
| | | barcelona | 7.549 | | |
| | | castellón | 359 | | |
| | | girona | 315 | | |
| | | ibiza-formentera | 17 | | |
| | | lleida | 57 | | |
| | | mallorca | 356 | | |
| | | menorca | 20 | | |
| | tarragona | 525 | | | |
| valencia | 3.666 | | | | |
| españa Total | | 14.182 | | | |
| De 3 a 4 Total | | | 14.195 | | |
| De 4 a 9 | españa | alicante | 20 | | |
| | | barcelona | 298 | | |
| Total General | | | 22.517 | | |

Captura 5. Número de pisos en oferta por número de habitaciones, país y provincia.

8.2.1.2 Inmuebles vendidos y en oferta

Este documento analítico responde al requerimiento del cliente de disponer del **número de inmuebles vendidos y existentes por zona, tipología y características**. Su planteamiento ha sido muy similar al anterior pues nos ha parecido que la tabla es un mecanismo gráfico óptimo para ver estas cifras.

Atlas SBI Menú

Inmuebles vendidos y en oferta

Trial version, Expiration date 15/03/2014

Documentos

Configuración

Un Piso Para Todos

Distribución Inmueble:

Número Inmueble

Inmuebles vendido

Metros Cuadr. mé

+

Análisis precios oferta

+

Ventas

+

Inmueble-Tipo

Documentos abiertos

Inmuebles vendidos y en of

Año

2006

Mes

1

MUNICIPIO

AREA

DISTRITO

METROS

NRO. HABITAC.

EN OFERTA

VENDIDOS

Suelto Campos de Columna Aquí

PAIS

PROVINCIA

Total General

EN OFERTA

VENDIDOS

andorra

andorra

60

0

españa

alicante

5.321

109

barcelona

15.520

428

castellón

1.274

15

girona

1.613

43

ibiza-formentera

146

3

lleida

253

9

mallorca

1.054

28

menorca

114

2

tarragona

1.803

40

valencia

7.880

184

españa Total

34.978

861

Total General

35.038

861

Captura 6. Tabla dinámica con el número de inmuebles en oferta y vendidos.

Del mismo modo, se han dispuesto distintas dimensiones para poder filtrar, clasificar u ordenar la información. En este caso, se ha filtrado además por número de habitaciones.

Inmuebles vendidos y en oferta

Trial version. Expiration date 15/03/2014

Mes

1

PIPO

AREA

DISTRITO

METROS

ERTA

VENDIDOS

NRO. HABITAC. ▲

▲

▼

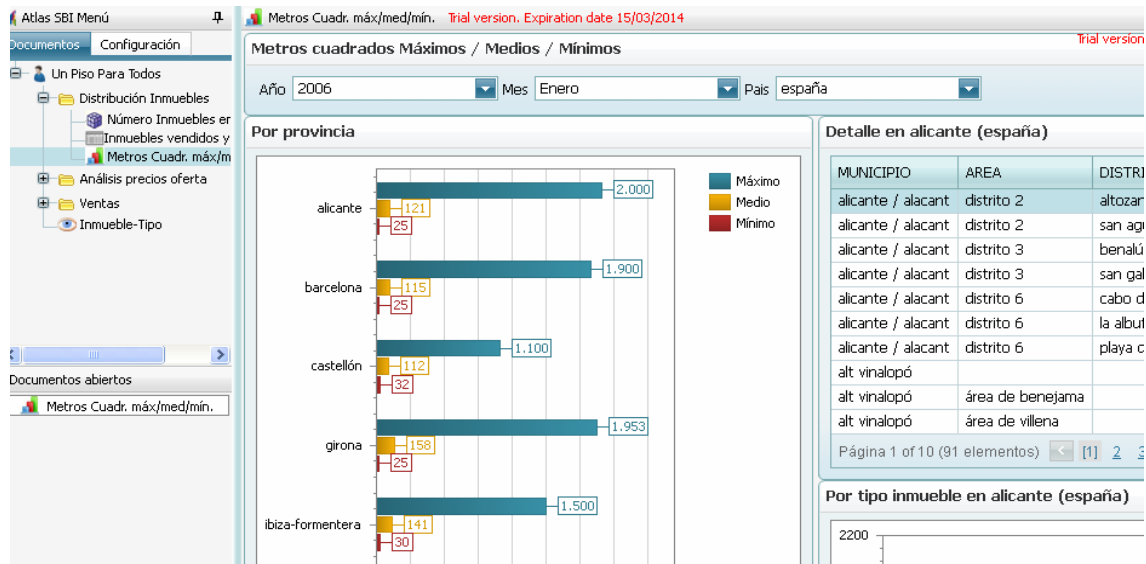
PROVINCIA ▲▼

| | | De 3 a 4 | | De 4 a 9 | | Desc. | | Hasta 2 | | Más de 9 | | Total General | |
|----------|------------------|-----------|----------|-----------|----------|-----------|----------|-----------|----------|-----------|----------|---------------|----------|
| | | EN OFERTA | VENDIDOS | EN OFERTA | VENDIDOS | EN OFERTA | VENDIDOS | EN OFERTA | VENDIDOS | EN OFERTA | VENDIDOS | EN OFERTA | VENDIDOS |
| espa | andorra | 21 | 0 | 1 | 0 | | | 38 | 0 | | | 60 | 0 |
| | alicante | 2.600 | 62 | 254 | 0 | 2 | 0 | 2.459 | 47 | 6 | 0 | 5.321 | 109 |
| | barcelona | 10.581 | 290 | 1.135 | 22 | 36 | 1 | 3.757 | 115 | 11 | 0 | 15.520 | 428 |
| | castellón | 610 | 10 | 50 | 0 | 2 | 0 | 612 | 5 | | | 1.274 | 15 |
| | girona | 979 | 23 | 186 | 7 | 2 | 1 | 437 | 12 | 9 | 0 | 1.613 | 43 |
| | ibiza-formentera | 55 | 2 | 8 | 1 | | | 83 | 0 | | | 146 | 3 |
| | lleida | 147 | 6 | 35 | 1 | 1 | 0 | 70 | 2 | | | 253 | 9 |
| | mallorca | 630 | 19 | 81 | 1 | 0 | 1 | 341 | 7 | 2 | 0 | 1.054 | 28 |
| | menorca | 65 | 1 | 4 | 0 | | | 45 | 1 | | | 114 | 2 |
| | tarragona | 1.133 | 23 | 124 | 2 | 1 | 1 | 543 | 14 | 2 | 0 | 1.803 | 40 |
| valencia | 5.488 | 139 | 532 | 9 | 11 | 0 | 1.842 | 36 | 7 | 0 | 7.880 | 184 | |
| Total | | 22.288 | 575 | 2.409 | 43 | 55 | 4 | 10.189 | 239 | 37 | 0 | 34.978 | 861 |

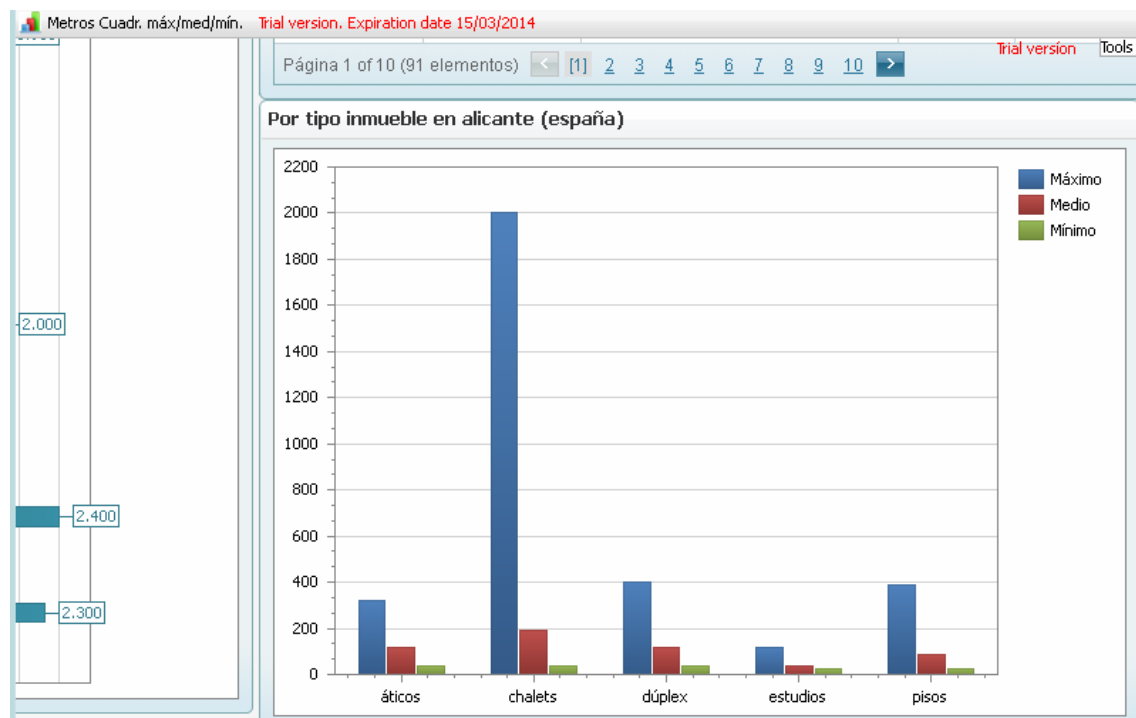
Captura 7. Número de inmuebles en oferta y vendidos por país, provincia y número de habitaciones.

8.2.1.3 Metros cuadrados máximos, medios y mínimos

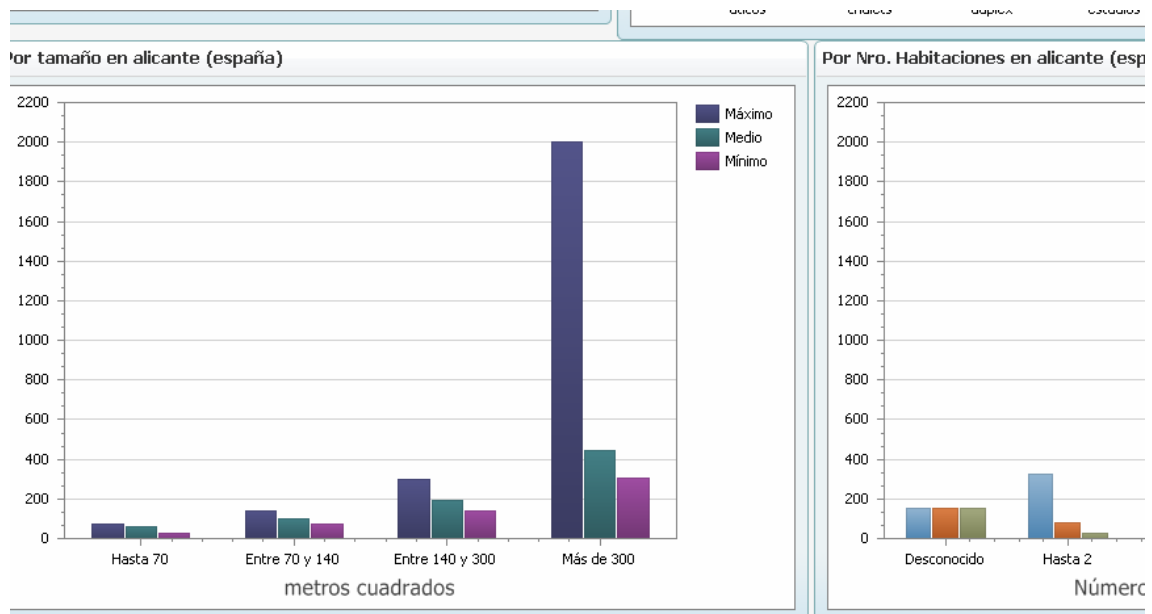
Este informe responde al requerimiento del cliente de disponer de esta información por zona, tipología y características de los inmuebles. Para presentarlo, se ha usado este formato compuesto que se aprecia en las capturas 8, 9 y 10, y el cual, hemos reutilizado en otro tipo de reportes que veremos más adelante pues nos ha parecido que gráficamente representa de forma clara y sencilla este tipo de información.



Captura 8. Metros cuadrados máximos, medios y mínimos por provincia y desglose por resto de zonas.



Captura 9. Metros cuadrados máximos, medios y mínimos por tipo de inmueble.



Captura 10. Metros cuadrados máximos, medios y mínimos por tamaño y por número de habitaciones.

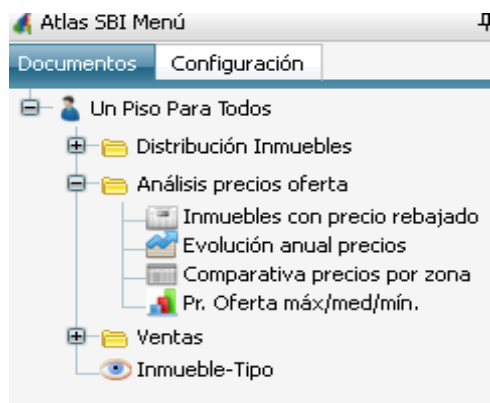
Este formato de documento se compone de:

- Selectores en cabecera del año, mes y país para seleccionar la información
- Gráfica de barras en formato vertical con los datos de cada provincia. Si se pulsa sobre alguna de las provincias, las gráficas restantes se actualizan con la información de dicha provincia, año y mes
- A su derecha, tabla de desglose de los datos por el resto de zonas: municipio, área y distrito
- Debajo de ésta, se muestra una gráfica de barras por tipo de inmueble
- En la parte inferior del documento, se muestran las gráficas de barras por tamaño (metros cuadrados) y por número de habitaciones

Por tanto, la información siempre se refiere a un año, mes, país y provincia concreta. En caso de cambiar los valores de los selectores, las gráficas y tablas se actualizan automáticamente.

8.2.2 Análisis precios de oferta

Al desplegar esta carpeta nos aparecen cuatro documentos analíticos: inmuebles con precio rebajado, evolución anual de precios, comparativa de precios por zona, y precios de oferta máximos, medios y mínimos.



Captura 11. Elementos de carpeta de análisis de precios de oferta.

8.2.2.1 Inmuebles con precio rebajado

Como indica su nombre, este documento responde al requisito de disponer de una **lista de inmuebles que han bajado de precio**. Para esta información se solicita el año, mes y el país. Se presenta en una tabla ordenada por provincia, con el resto de áreas geográficas de la zona a la que pertenece el inmueble, las dimensiones tipo, metros cuadrados y número de habitaciones para seleccionar o agrupar, y los datos precio actual del mes y el del mes anterior, y el porcentaje de bajada. Lógicamente, sólo se muestran los inmuebles cuyo precio actual es inferior al anterior. En caso de que el porcentaje de bajada sea igual o mayor a un 10% se muestra en rojo esta columna, tal y como puede verse en la siguiente captura, donde se están consultando los inmuebles con precio rebajado en Ibiza-Formentera en enero del 2008. También se muestra una columna con la referencia del inmueble por si se quiere saber exactamente la referencia del inmueble que ha bajado de precio.

Inmuebles con precio rebajado

Año

2008

Mes

Enero

País

españa

Lista de inmuebles con precio rebajado en españa (2008/ 1)

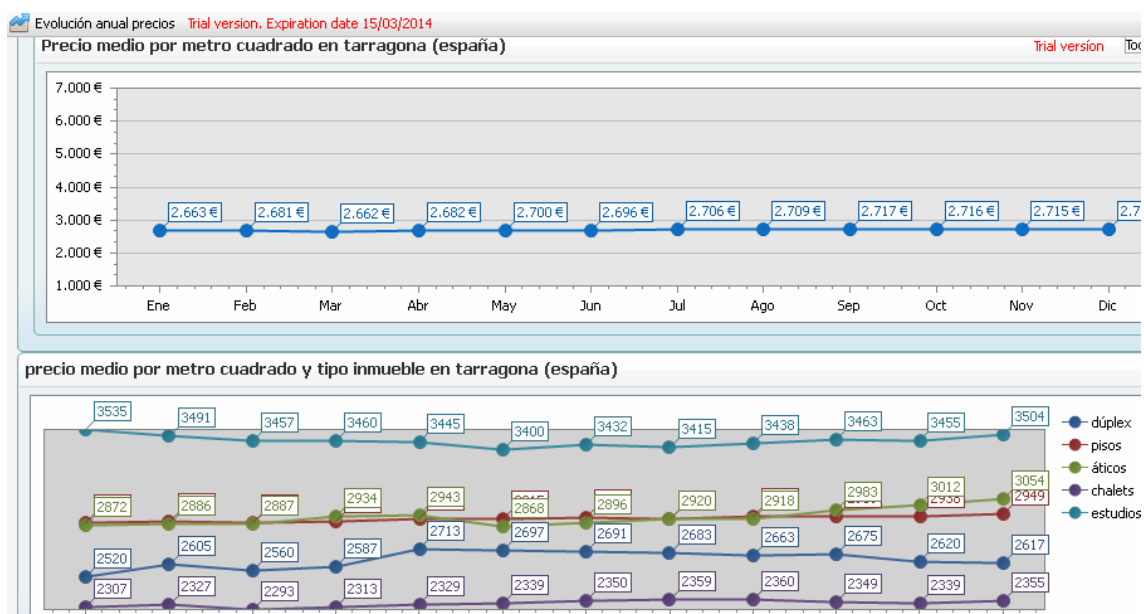
| | AREA | DISTRITO | TIPO | METROS | NRO. HABITAC. | PRECIO_ACT | PRECIO_ANT | % BAJADA | REFER |
|----------------------------------|------|----------|--------|----------------|---------------|------------|------------|----------|---------|
| | | | | | | | | | |
| PROVINCIA: alicante | | | | | | | | | |
| PROVINCIA: barcelona | | | | | | | | | |
| PROVINCIA: castellón | | | | | | | | | |
| PROVINCIA: girona | | | | | | | | | |
| PROVINCIA: ibiza-formentera | | | | | | | | | |
| MUNICIPIO: área de eivissa | | | | | | | | | |
| | | | pisos | Hasta 70 | Hasta 2 | 170.000 € | 175.000 € | -2,86 % | VC00000 |
| MUNICIPIO: área de santa eulària | | | | | | | | | |
| | | | pisos | Entre 70 y 140 | Hasta 2 | 350.000 € | 395.000 € | -11,39 % | VW00000 |
| | | | dúplex | Entre 70 y 140 | De 3 a 4 | 269.000 € | 270.000 € | -0,37 % | VW00000 |
| PROVINCIA: lleida | | | | | | | | | |
| PROVINCIA: mallorca | | | | | | | | | |
| PROVINCIA: menorca | | | | | | | | | |
| PROVINCIA: tarragona | | | | | | | | | |

Captura 12. Lista de inmuebles en oferta con precio rebajado.

8.2.2.2 Evolución anual de precios

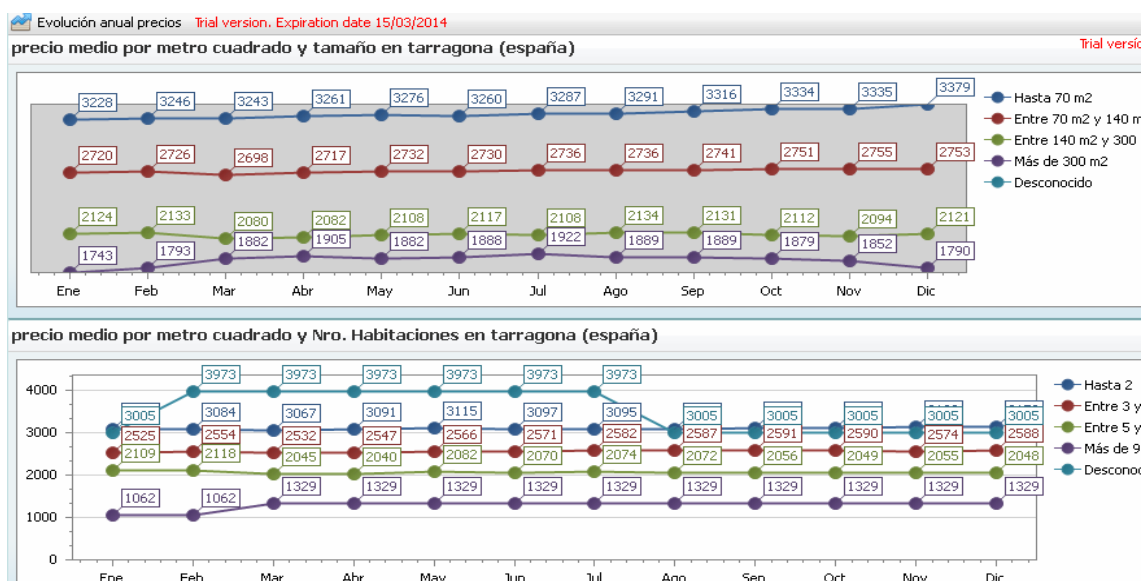
Se trata de otro requisito del cliente. Hemos decidido presentarlo mediante gráficas de líneas con la evolución mensual de los **precios medios por metro cuadrado** de los inmuebles en oferta para un año, país y provincia.

Son en total cuatro gráficas. En la captura 13 se pueden observar las dos primeras: evolución del precio medio de oferta por metro cuadrado en la provincia seleccionada y por tipo de inmueble.



Captura 13. Evolución de precios medios de oferta por metro cuadrado en provincia determinada y por tipo en dicha provincia.

En la siguiente figura se observa la evolución de los precios medios de oferta por metro cuadrado por tamaño y número de habitaciones, en la misma provincia y año seleccionado.



Captura 14. Evolución de precios medios de oferta por metro cuadrado por tamaño y número de habitaciones en provincia determinada.

8.2.2.3 Comparativa de precios por zona

Este reporte no es un requisito directo del cliente, si bien, hemos decidido añadirlo al proyecto porque pensamos puede ser de interés. Ofrece una tabla comparativa de los precios medios por metro cuadrado de inmuebles ofertados por zonas y por meses de un año concreto. La tabla permite también seleccionar el mes hasta el cual se desea que aparezcan los datos del año seleccionado, y si la comparativa se quiere por meses, trimestres, semestres, etc. En caso de elegir otra periodificación diferente a la mensual, los datos de los precios medios por metro cuadrado se acumulan, no se calculan precios medios nuevamente, pero permite de una forma sencilla ver la evolución y comparar precios por país, provincia y municipio. Por supuesto, también se puede filtrar por las columnas de la tabla.

Comparativa precios por zona Trial version. Expiration date 15/03/2014

Comparativa precio medio del metro cuadrado por zona

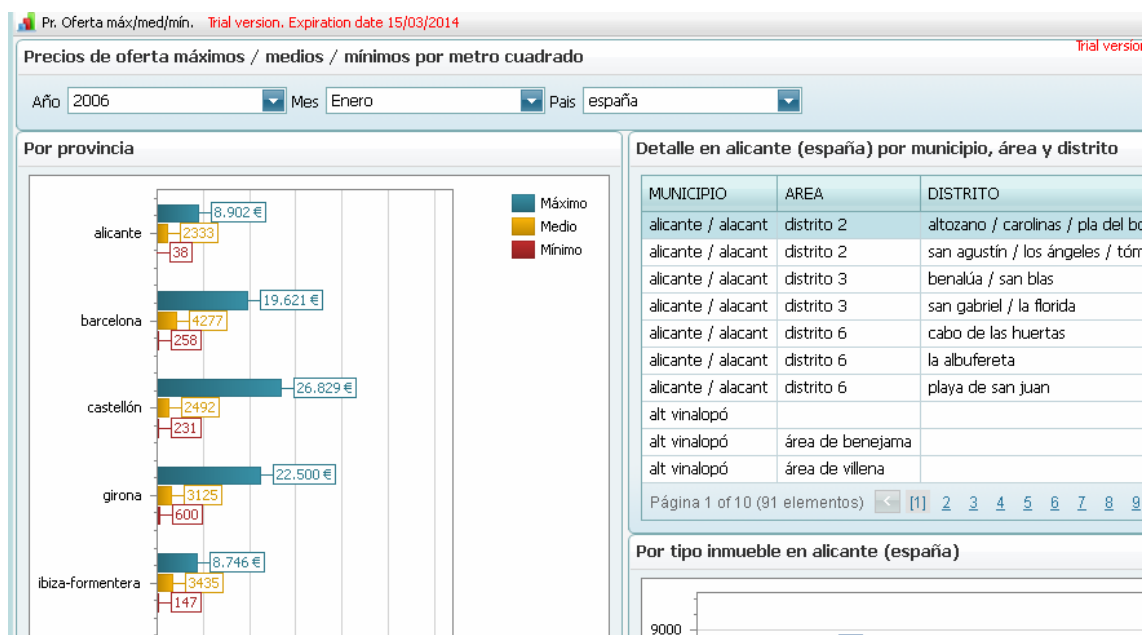
Periodificación: Mensual Año: 2006 Mes: Noviembre

| PAIS | PROVIN | MUNICIP | Año | Enero PRECIO_METRO | Febrero PRECIO_METRO | Marzo PRECIO_METRO | Abril PRECIO_METRO |
|---------|-----------|------------------------------|------|-----------------------|-------------------------|-----------------------|-----------------------|
| andorra | andorra | encamp - canillo - ordino | 2006 | 4.069 € | 4.069 € | 4.136 € | 4.136 € |
| andorra | andorra | massana - andorra - escaldes | 2006 | 4.796 € | 4.796 € | 4.411 € | 4.411 € |
| andorra | andorra | Sin informar | 2006 | 4.253 € | 4.262 € | 4.286 € | 4.286 € |
| españa | alicante | alicante / alacant | 2006 | 2.254 € | 2.248 € | 2.250 € | 2.250 € |
| españa | alicante | alt vinalopó | 2006 | 1.355 € | 1.377 € | 1.400 € | 1.400 € |
| españa | alicante | baix segura | 2006 | 2.004 € | 1.995 € | 1.979 € | 1.979 € |
| españa | alicante | baix vinalopó | 2006 | 2.263 € | 2.257 € | 2.233 € | 2.233 € |
| españa | alicante | comtat | 2006 | 1.476 € | 1.597 € | 1.597 € | 1.597 € |
| españa | alicante | l'alacantí | 2006 | 2.336 € | 2.327 € | 2.322 € | 2.322 € |
| españa | alicante | l'alcoya | 2006 | 1.297 € | 1.283 € | 1.289 € | 1.289 € |
| españa | alicante | marina alta | 2006 | 2.552 € | 2.552 € | 2.540 € | 2.540 € |
| españa | alicante | marina baixa | 2006 | 2.829 € | 2.844 € | 2.832 € | 2.832 € |
| españa | alicante | torrevieja | 2006 | 1.970 € | 1.952 € | 1.946 € | 1.946 € |
| españa | alicante | vinalopó mitjà | 2006 | 1.378 € | 1.360 € | 1.382 € | 1.382 € |
| españa | barcelona | alt penedès | 2006 | 2.603 € | 2.640 € | 2.630 € | 2.630 € |
| | | | | 2.474 € | 2.480 € | 2.467 € | 2.467 € |

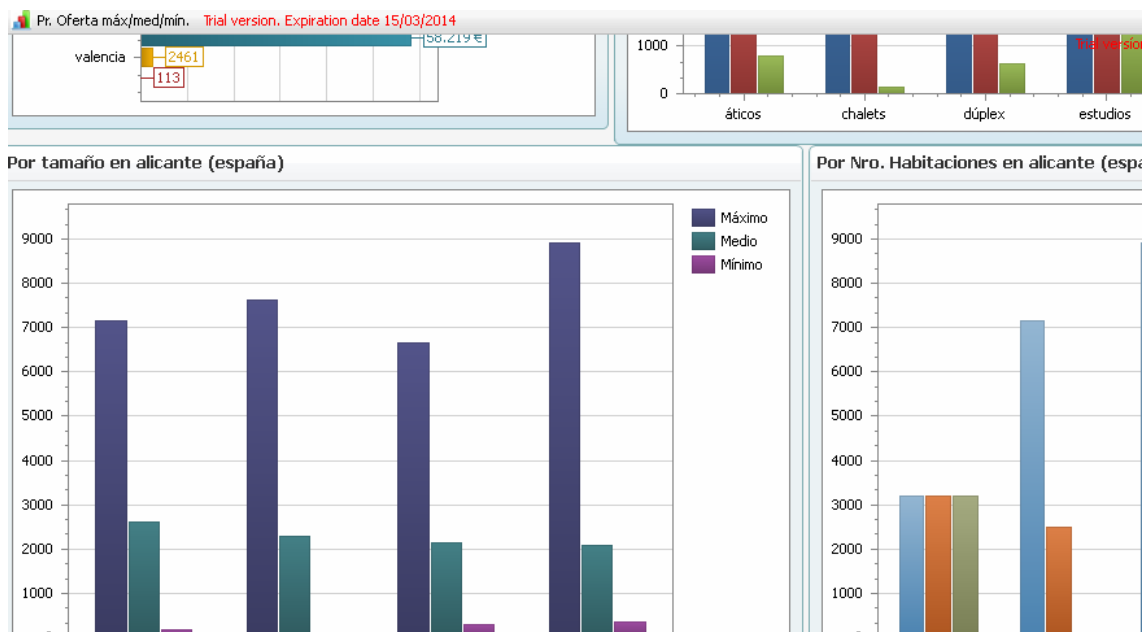
Captura 15. Comparativa de precios medios de oferta por metro cuadrado por zonas.

8.2.2.4 Precios de oferta máximos, medios y mínimos

Este documento sí responde a un requerimiento solicitado por el cliente. Para presentarlo hemos usado el mismo formato explicado en el apartado 8.2.1.3. Creemos que no es necesario volver a explicarlo con tanto detalle, simplemente, indicar que el dato que presenta es el precio medio por metro cuadrado de oferta máximo, medio y mínimo en cada provincia del país, año y mes seleccionado, con su desglose por municipio, áreas y distritos en tabla a su derecha, y por tipo de inmueble, tamaño y número de habitaciones en gráficas más abajo.



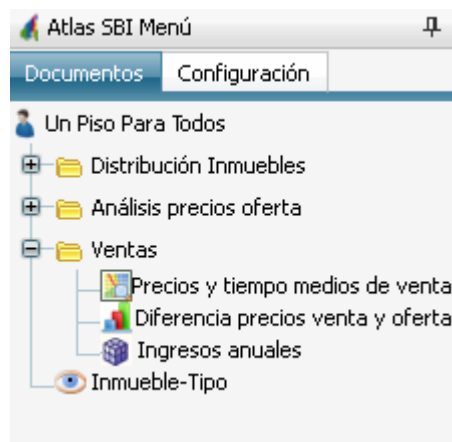
Captura 16. Precios medios de oferta por metro cuadrado por provincia y desglose por resto de zonas.



Captura 17. Precios medios de oferta por metro cuadrado por tipo, tamaño y número de habitaciones en provincia determinada.

8.2.3 Ventas

Al desplegar esta carpeta nos aparecen tres documentos analíticos: precios y tiempos medios de venta, diferencia entre precios de venta y oferta, e ingresos anuales.

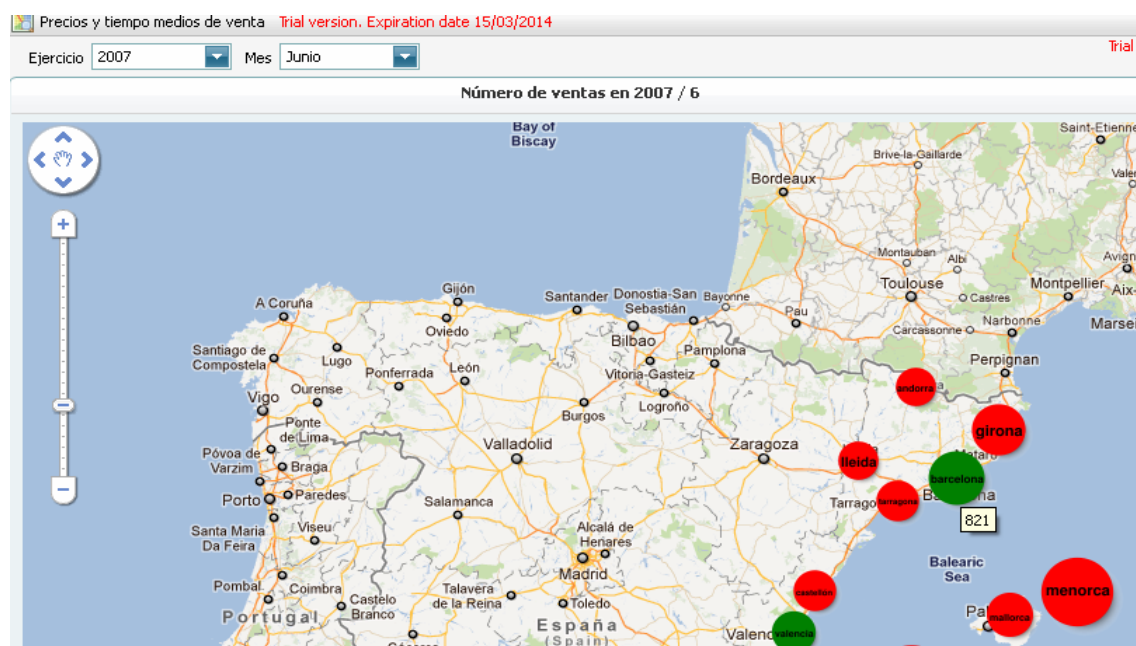


Captura 18. Elementos de carpeta de ventas.

8.2.3.1 Precios y tiempo medios de venta

En este documento analítico hemos incorporado dos requisitos solicitados por el cliente: **precios de venta real máximos, medios y mínimos, así como, el tiempo medio de venta**, todo ello, por zona, tipología y características.

Para implementarlo hemos acudido a una presentación en formato mapa *GoogleMap* donde se muestran en círculos las zonas (provincias) donde la empresa ha realizado ventas en el año y mes seleccionado.

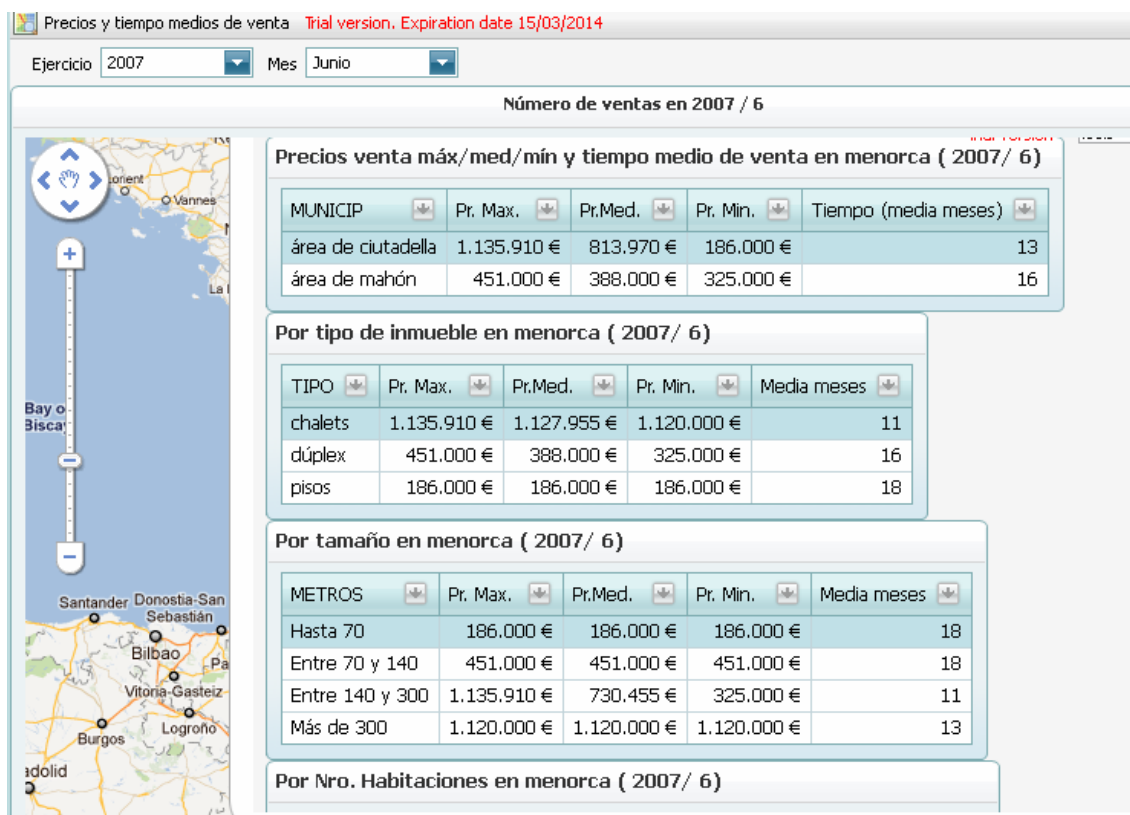


Captura 19. Precios de venta máximos, medios, mínimos y tiempos de venta en junio del 2007. Presentación.

Los círculos se muestran con las siguientes características:

- En verde, aquellas provincias que han vendido más de 100 inmuebles en dicho mes. De hecho, al pasar el ratón por ellos se muestra el número de inmuebles. En la captura anterior, se puede ver que en Barcelona, en junio de 2007, se vendieron 821 inmuebles
- El diámetro del círculo viene marcado por el precio medio real de las ventas realizadas, de forma que, a mayor diámetro, mayor será el precio medio de las ventas realizadas en el mes. Así pues, en la figura anterior, podemos ver que aunque en Menorca se han vendido menos de 100 inmuebles en ese mismo periodo (concretamente 5, y por eso, es un círculo rojo), sin embargo, deben haber sido inmuebles de precio alto pues aparece con un diámetro mayor que la propia Barcelona. Así es, el precio medio de venta real en Menorca en ese mes es 643.582 euros, mientras que en Barcelona es 432.401 euros. Hemos preferido usar este dato pues aportaba algo más de información que haber usado el total de ventas. Parece obvio que si hay muchas ventas, círculo en verde, los ingresos en estas provincias seguramente sean mayores que en las otras. Al usar este otro parámetro, nos puede inducir a pensar que aunque el número de ventas no sea el de provincias en verde, han sido ventas de calidad (con un precio medio alto)

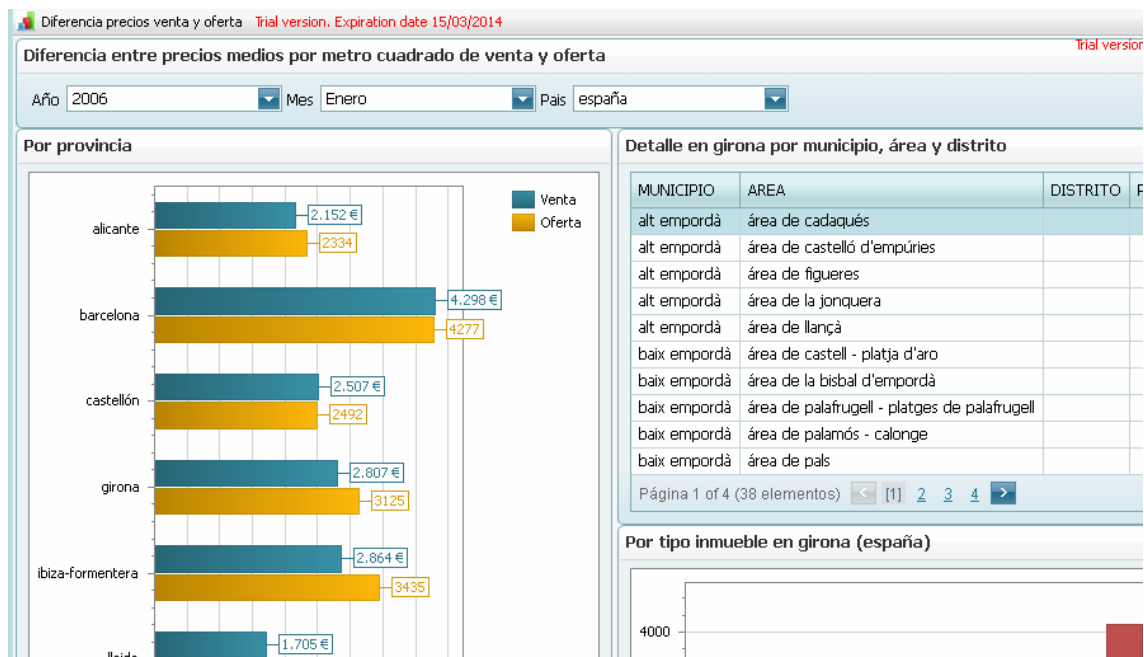
En todo caso, al seleccionar una provincia, se muestra otro documento analítico sobre dicha provincia con los datos de precios máximos, medios y mínimos reales de ventas en ese mes y año, así como, el tiempo medio en meses de las ventas, todo ello clasificado por municipios, tipo de inmueble, tamaño y número de habitaciones.



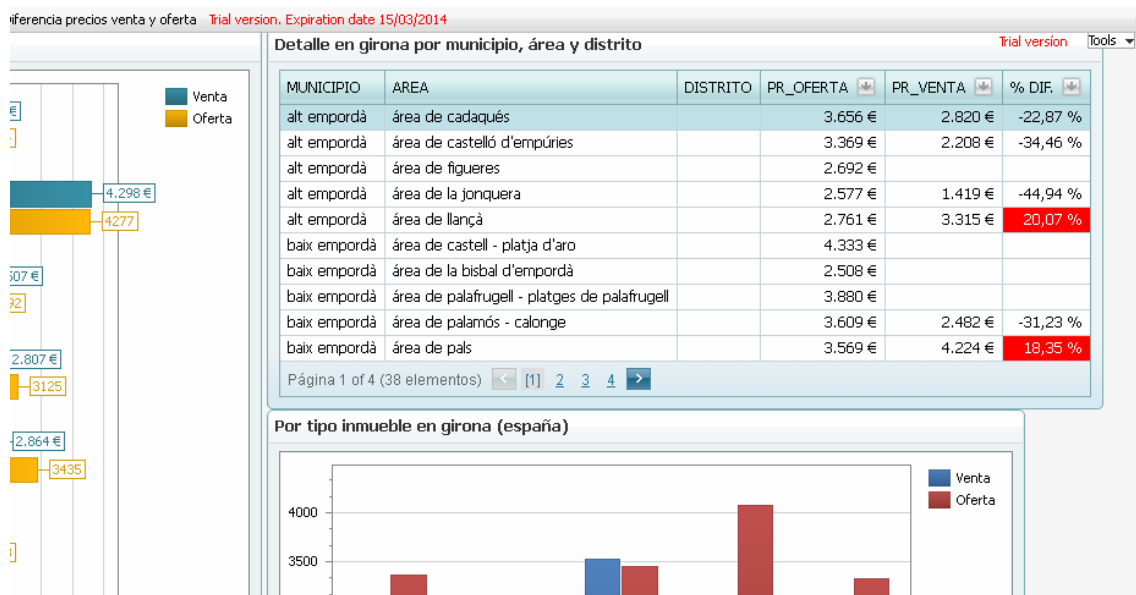
Captura 20. Precios máximos, medios y mínimos de ventas reales y tiempos de venta en junio de 2007 en Menorca. Detalle por municipio, tipo de inmueble y tamaño.

8.2.3.2 Diferencia entre precios de venta y oferta

Se trata de otro requisito del cliente en donde se busca obtener información de la diferencia entre los precios de venta y de oferta. Nosotros lo hemos analizado a partir del precio medio por metro cuadrado en inmuebles ofertados y vendidos en el periodo seleccionado. Nos ha parecido que la mejor forma de representarlo es a través de un documento similar al usado para obtener la información de máximos, medios y mínimos. Es decir, se solicitan el año, mes y país, y se muestra un documento analítico con un gráfico de barras a su izquierda con los precios medios por metro cuadrado de venta y de oferta de las diferentes provincias, una tabla de desglose por municipios, áreas y distritos a su derecha, con sendos datos de la provincia seleccionada y la diferencia en porcentaje, mostrándose en rojo aquellos casos en los que el precio de venta es un 15% superior al de oferta. Debajo, gráficas de barras con los precios medios por metro cuadrado de venta y de oferta de esta provincia seleccionada según el tipo de inmueble, tamaño y número de habitaciones.



Captura 21. Precios medios por metro cuadrado de venta y oferta por provincia.



Captura 22. Precios medios por metro cuadrado de venta y oferta desglosados por zona y tipo de inmueble en Girona (enero 2006).



Captura 23. Precios medios por metro cuadrado de venta y oferta desglosados por tamaño y número de habitaciones en Girona (enero 2006).

8.2.3.3 Ingresos anuales

Dentro de esta carpeta de ventas hemos incluido un informe no solicitado para comprobar los ingresos anuales por ventas realizadas, pues nos ha parecido interesante poder clasificar estos ingresos por diferentes criterios.

Para presentarlo se ha usado una tabla dinámica que inicialmente presenta las cifras de ventas por año, país y provincia.

Ingresos anuales Trial version. Expiration date 15/03/2014

Ingresos anuales por ventas de inmuebles

MUNICIP AREA DISTRITO TIPO METROS NRO. HABITACIONES

INGRESOS

Sulte Campos de Columna Aquí

| ANIO | PAIS | PROVIN | INGRESOS Total |
|---------------|----------|------------------|----------------|
| 2006 | andorra | andorra | 3.354.781 |
| | españa | alicante | 584.498.518 |
| | | barcelona | 3.857.763.332 |
| | | castellón | 119.907.711 |
| | | girona | 313.427.742 |
| | | ibiza-formentera | 29.381.041 |
| | | lleida | 33.016.830 |
| | | mallorca | 238.751.277 |
| | | menorca | 18.195.458 |
| | | tarragona | 261.155.520 |
| | valencia | 1.319.339.242 | |
| españa Total | | 6.775.436.671 | |
| 2006 Total | | | 6.778.791.452 |
| 2007 | andorra | andorra | 8.133.347 |
| | españa | alicante | 636.762.222 |
| Total General | | | 14.744.911.448 |

Captura 24. Ingresos anuales por ventas realizadas por año, país y provincia.

Como en anteriores tablas, en su cabecera, se han ubicado distintas dimensiones que permitan clasificar la cifra de ventas por diferentes criterios. Por ejemplo, en la siguiente captura se ha incorporado la dimensión “tipo” para obtener la información por año, país, provincia y tipo de inmueble.

Ingresos anuales Trial version. Expiration date 15/03/2014

| MUNICIP | | | | | AREA | DISTRITO | METROS | NRO. HABITACIONES | |
|---------------|---------|---------------|-----------------|-------------------------------|-------------|---------------|--------|-------------------|--|
| INGRESOS | | | | Suelta Campos de Columna Aquí | | | | | |
| ANIO | PAIS | PROVIN | TIPO | INGRESOS Total | | | | | |
| 2006 | andorra | andorra | áticos | 260.000 € | | | | | |
| | | | estudios | 319.000 € | | | | | |
| | | | pisos | 2.775.781 € | | | | | |
| | | andorra Total | | | 3.354.781 € | | | | |
| | españa | alicante | áticos | 47.598.461 € | | | | | |
| | | | chalets | 225.049.813 € | | | | | |
| | | | dúplex | 19.041.577 € | | | | | |
| | | | estudios | 3.230.627 € | | | | | |
| | | | pisos | 289.578.040 € | | | | | |
| | | | alicante Total | | | 584.498.518 € | | | |
| | | barcelona | áticos | 306.112.576 € | | | | | |
| | | | chalets | 802.512.143 € | | | | | |
| | | | dúplex | 167.990.533 € | | | | | |
| | | | estudios | 13.363.686 € | | | | | |
| pisos | | | 2.567.784.394 € | | | | | | |
| Total General | | | | 14.744.911.448 € | | | | | |

Página 1 of 9 (131 elementos) 1 2 3 4 5 6 7 8 9

Captura 25. Ingresos anuales por ventas realizadas por año, país, provincia y tipo de inmueble.

8.2.4 Inmueble “tipo”

El cliente también nos comunicó que explicásemos cual es el piso-tipo español y andorrano. Aunque se especificó “piso”, nosotros hemos decidido ampliar la búsqueda de la información a todos los posibles tipos.

El criterio usado para ello ha sido el recuento histórico del número de inmuebles diferentes de cada país para todos los periodos de los que tenemos datos, independientemente de si se ha vendido o no, teniendo en cuenta que tuviera precio informado, de forma que no tuviéramos en cuenta aquellos descartados por tener precios erróneos, que se hubiera podido categorizar correctamente por tamaño y número de habitaciones para poder asignar bien sus características, y clasificando el recuento por tipo de inmueble, tamaño y número de habitaciones (dimensiones categorizadas). Así pues, aquel grupo con mayor número de inmuebles es el que presentamos como “inmueble-tipo”. Además, se ha obtenido el precio medio para cada categoría.

Se puede observar en las siguientes capturas que en Andorra, el inmueble tipo son los pisos de hasta 70 metros cuadrados y con 2 habitaciones o menos. El precio medio a lo largo de los meses de los que disponemos datos es de 239.345 euros. Sin embargo, en España, se trata de pisos con un tamaño de entre 70 y 140 metros cuadrados, y de 3 o 4 habitaciones. Tienen un precio medio de 314.478 euros.

Inmueble-Tipo Trial version. Expiration date 15/03/2014

Relación de inmuebles tipo por país

Pais

Tabla de inmuebles tipo en andorra

| TIPO | METROS | NRO. HABITACIONES | NRO. INMUEBLES | PRECIO MEDIO |
|----------|-----------------|-------------------|----------------|--------------|
| pisos | Hasta 70 | Hasta 2 | 100 | 239.345 € |
| pisos | Entre 70 y 140 | De 3 a 4 | 60 | 472.966 € |
| pisos | Entre 70 y 140 | Hasta 2 | 44 | 359.177 € |
| chalets | Entre 140 y 300 | De 3 a 4 | 21 | 862.983 € |
| estudios | Hasta 70 | Hasta 2 | 13 | 184.147 € |
| pisos | Entre 140 y 300 | De 3 a 4 | 11 | 702.056 € |
| chalets | Más de 300 | De 3 a 4 | 10 | 2.239.322 € |
| chalets | Más de 300 | De 4 a 9 | 5 | 1.770.126 € |
| dúplex | Entre 70 y 140 | De 3 a 4 | 5 | 586.265 € |
| dúplex | Entre 70 y 140 | Hasta 2 | 4 | 353.643 € |
| áticos | Entre 70 y 140 | De 3 a 4 | 4 | 515.690 € |
| áticos | Entre 70 y 140 | Hasta 2 | 4 | 378.075 € |
| áticos | Hasta 70 | Hasta 2 | 4 | 257.645 € |
| dúplex | Entre 140 y 300 | De 3 a 4 | 3 | 685.934 € |
| chalets | Entre 70 y 140 | De 3 a 4 | 2 | 607.033 € |
| 300 | | | | |

Página 1 of 2 (23 elementos) < [1] 2 >

Captura 26. Tabla de inmuebles tipo de Andorra.

Inmueble-Tipo Trial version. Expiration date 15/03/2014

Relación de inmuebles tipo por país

Pais

Tabla de inmuebles tipo en españa

| TIPO | METROS | NRO. HABITACIONES | NRO. INMUEBLES | PRECIO MEDIO |
|---------|-----------------|-------------------|----------------|--------------|
| pisos | Entre 70 y 140 | De 3 a 4 | 30.201 | 314.478 € |
| pisos | Hasta 70 | Hasta 2 | 13.743 | 216.720 € |
| chalets | Entre 140 y 300 | De 3 a 4 | 8.228 | 504.671 € |
| pisos | Entre 70 y 140 | Hasta 2 | 7.533 | 275.259 € |
| pisos | Hasta 70 | De 3 a 4 | 6.794 | 257.557 € |
| chalets | Entre 70 y 140 | De 3 a 4 | 4.028 | 335.705 € |
| áticos | Entre 70 y 140 | De 3 a 4 | 2.516 | 362.633 € |
| chalets | Entre 140 y 300 | De 4 a 9 | 2.108 | 621.342 € |
| dúplex | Entre 70 y 140 | De 3 a 4 | 1.935 | 355.781 € |
| pisos | Entre 140 y 300 | De 3 a 4 | 1.818 | 564.143 € |
| chalets | Más de 300 | De 4 a 9 | 1.630 | 1.223.366 € |
| áticos | Hasta 70 | Hasta 2 | 1.231 | 248.477 € |
| chalets | Más de 300 | De 3 a 4 | 1.166 | 961.287 € |
| chalets | Entre 70 y 140 | Hasta 2 | 1.076 | 254.681 € |
| áticos | Entre 70 y 140 | Hasta 2 | 1.060 | 312.327 € |
| 91.922 | | | | |

Página 1 of 4 (55 elementos) < [1] 2 3 4 >

Captura 27. Tabla de inmuebles tipo de España.

9. Conclusiones

En primer lugar, quiero reseñar que, a mi juicio, el TFC ha cumplido plenamente el gran objetivo global que plantea: la introducción del alumno en el área de los almacenes de datos. El trabajo realizado ha servido para entender el cometido que tienen a la hora de explotar la información desde diferentes perspectivas de análisis, y su ayuda para la toma de decisiones en las organizaciones. En este sentido, ha quedado muy clara la diferenciación entre la filosofía del modelo clásico relacional y la filosofía del modelo dimensional, orientado al **hecho** del negocio, y su porqué (necesidad de filtrar, agrupar, resumir, calcular, etc.).

También queda entendida la lógica evolución que este tipo de arquitecturas, y herramientas de BI en general, han sufrido de un tiempo a esta parte. Una vez superado y asimilado por parte de las organizaciones la necesidad de almacenar la información, una vez superado el coste económico que conllevaban los grandes volúmenes de memoria, parece claro que el siguiente paso es el uso de la información almacenada para extraer conocimiento de ella, y para eso, es necesario almacenarla de forma eficiente de cara a los requisitos que tengamos, y así, poder responder de forma más rápida y sencilla a las consultas planteadas. Esto es obvio, y de hecho, es el gran motivo del auge de *Data Warehouse*, CRM (*Customer Relationship Management*) o herramientas de *Data Mining*, pero no por ello quería dejar de exponer como conclusión el aporte que el TFC da a su comprensión.

Otra conclusión no menos lógica, aunque solo sea pensando en las posibilidades que se abren para el usuario de negocio que visualiza alguna herramienta de BI aplicada sobre un almacén, es la importancia del buen análisis conceptual y físico, y el dimensionado del hardware que lo soporte, pues a buen seguro se solicitará que crezca rápido con nuevas funcionalidades.

En cuanto al resto de objetivos planteados en el primer apartado de esta memoria, creo que en mayor, o menor medida, se han cumplido todos. Con respecto a los que califiqué como objetivos globales, que no son más que un repaso de las fases por las que hemos avanzado durante este periodo, se puede concluir la importancia que cada una de ellas tiene. En realidad, no están nada alejadas de las del ciclo de vida de cualquier proyecto software. Es primordial recoger y entender bien los requisitos que ha de cumplir el almacén, ser conscientes de la necesidad de los procesos ETL, entender que como toda migración de datos encontraremos errores que debemos informar y depurar, y finalmente, descubrir las posibilidades de las herramientas de BI aplicadas sobre el almacén. Respecto de los objetivos específicos solicitados pienso que también se han cumplido, si bien, me hubiera gustado disponer de algo más de tiempo para mejorar ciertos aspectos que indico en el próximo apartado.

Por último, simplemente destacar que, como esperaba al elegir esta área de TFC, el proyecto me ha parecido muy interesante, y me ha permitido profundizar en esta rama que siempre me ha atraído mucho.

10. Líneas de evolución futuras

En mi opinión, el almacén que he construido puede mejorar en cuatro aspectos fundamentalmente.

En primer lugar, por falta de tiempo, no he podido ampliar el diseño para dar cabida jerarquías de zonas compuestas por más de cuatro elementos geográficos. Como indiqué en apartados anteriores, se reciben algunos inmuebles ubicados en una localización compuesta por cinco zonas geográficas, una más de las previstas por el almacén y que son las que nos comunicó el cliente. La idea era ampliar el número de zonas posibles en un número con el cual no tuviéramos problemas a futuro, sin embargo, finalmente no tuve tiempo de rehacer modelos y cargas.

Otra de las posibles evoluciones del almacén es en su diseño físico. Finalmente, no implementé totalmente el modelo conceptual presentado que establecía relaciones entre la tabla de inmuebles y su tipo, así como con la jerarquía de zonas a la que pertenece. De nuevo por motivos de tiempo no pude entrar a implantar esta mejora.

Ambas líneas de evolución futuras anteriores demuestran el ciclo iterativo de todo proyecto de software que, como no, también está presente aquí, pues su implementación requiere revisar de nuevo el alcance en fase de análisis, modelo, proceso ETL e informes finales, lo cual, confirma una de las conclusiones expuestas anteriormente: la importancia del análisis y del modelo conceptual.

Tras ver las posibilidades de la herramienta de BI Atlas SBI, creo también que la tabla de hechos puede reducirse en algunos indicadores y con ello ahorrar tiempo en los procesos ETL.

Otra línea clara de evolución a futuro es trabajar en la optimización del código para el mantenimiento evolutivo del almacén.

Glosario

Data Warehouse o Almacén de datos. Repositorios de datos cuya finalidad es recopilar, resumir y tratar eficientemente grandes volúmenes de datos existentes en los sistemas de información de las organizaciones, proporcionando un análisis de la información desde distintas perspectivas o dimensiones.

Business Intelligence. Conjunto de herramientas y estrategias enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización.

Hecho. En almacenes de datos representa los eventos o conceptos de negocio que se quieren analizar.

Dimensión. Representan diferentes perspectivas desde las cuales se va a realizar el análisis. Permiten agrupar, filtrar u ordenar la información.

Métrica o Indicador. Se trata de propiedades del Hecho, medidas, o conceptos calculados, y con los cuales generalmente se puede operar, que resultan interesantes para el análisis y cumplir con los objetivos requeridos. Quedan asociados al Hecho en el modelo dimensional.

Granularidad. Define las dimensiones que deberán tener el almacén de datos y el nivel de las medidas o métricas.

OLAP. *On-Line Analytical Processing.* Solución usada en el campo de la Inteligencia Empresarial cuyo objetivo es agilizar la consulta de grandes volúmenes de datos.

ROLAP. *Relational On-Line Analytical Processing.* Implementación OLAP construida sobre bases de datos relacionales.

Procesos ETL. Proceso para la extracción, transformación y carga (*extract, transform and load*) de los datos originales sobre el almacén de datos.

Máquina Virtual. Software que emula una computadora y que puede ejecutar procesos de la misma forma que una computadora real, si bien, se hace bajo las características, abstracciones y recursos definidos para este software.

Java. Lenguaje de programación orientado a objetos.

Clase Java. Es el núcleo del lenguaje Java. Se trata de una construcción lógica que define la forma y el comportamiento en la aplicación Java de una entidad a la que modela.

PL/SQL. *Procedural Language Structured Query Language.* Se trata de una extensión Oracle del estándar *Structured Query Language (SQL)* en forma de lenguaje de programación para crear procedimientos o funciones que pueden ser almacenados como objetos también de la base de datos.

Bibliografía y referencias

[1] **The Data Warehouse Toolkit (2nd Edition). The complete guide to dimensional modeling.** Ralph Kimball and Margy Ross. Willey Computer Publishing, Inc. 2002.

[2] **Data Warehousing for Dummies (2nd Edition).** Thomas C. Hammergren and Alan R. Simon. Willey Publishing, Inc. 2009.

[3] **Kimball University: The 10 Essential Rules of Dimensional Modeling.** Margy Ross. InformationWeek. 2009

<http://www.informationweek.com/news/software/bi/217700810?pgno=1>

[4] **Modelamiento Multidimensional.** Carmen Gloria Wolff. Universidad de Concepción. Chile. 2002.

<http://www.inf.udec.cl/~revista/ediciones/edicion4/modmulti.PDF>

[5] **Atlas SBI Manual de instalación. Proyectos Gestión Conocimiento. Octubre 2010.**

[6] **La investigación en OLAP y Data Warehousing: pasado, presente y futuro.** Alejandro Vaisman. Universidad de Buenos Aires. Jornadas de Data Mining. Septiembre 2006.

<http://www.dc.uba.ar/events/jadm/2006/docs/VaismanjornadasDM.pdf/view>

[7] **Todo lo que querías saber sobre Datawarehouse.** Christian Herrera. Tutorial. Octubre 2007.

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=datawarehouse>

Otros enlaces consultados:

http://es.wikipedia.org/wiki/Esquema_en_copo_de_nieve

http://es.wikipedia.org/wiki/Esquema_en_estrella

http://en.wikipedia.org/wiki/Dimensional_Fact_Model

www.pgconocimiento.com

A Dimensional Modeling Manifesto. Ralph Kimball. August 1997.

http://www.kimballgroup.com/html/articles_search/articles1997/9708d15.html

<http://www.pensamientoscomputables.com/entrada/data-warehouse/mineria-datos/almacen-datos/base-de-datos-multidimensionales>

Anexo 1. Breve teoría sobre almacenes de datos

Podríamos definir Almacén de Datos, o *Data Warehouse* en inglés, como un sistema donde guardamos datos que han sido extraídos, y probablemente transformados con el objetivo de conseguir cargar las bases de datos de forma más eficiente para el análisis, desde otros sistemas, y desde el cual, podemos explotar la información con el objetivo de generar conocimiento que ayude a la decisión empresarial u organizativa. Más allá de entrar en polémicas en cuanto a la definición en sí –existen autores de gran prestigio como Kimball⁴ que simplemente lo definen desde la perspectiva de los datos–, lo importante es entender su función y porqué se han vuelto tan populares.

La realidad sobre los almacenes de datos gira entorno a una serie de conceptos que podríamos aglutinar en lo que se conoce como *Data Warehousing*, que se podría definir como las diferentes técnicas que consiguen que podamos copiar datos de fuentes externas sobre un entorno optimizado para el análisis y procesamiento de la información de forma eficiente. Si a ello le aplicamos herramientas de obtención de conocimiento – conocidas como *Data Mining*– resulta evidente el interés que desde el punto de vista empresarial, o de otro tipo, puede despertar.

La teoría fundamental sobre la que sustentan los almacenes de datos es la teoría del Modelo de Hechos Dimensionales (*DFM, Dimension Fact Model*), la cual, difiere bastante con la clásica teoría Entidad – Relación en su concepto como el propio Kimball advierte en este artículo⁵. Sus componentes básicos son el Hecho, sobre el que gira todo, la Dimensión y los Indicadores. Podemos definir el **Hecho** como un evento que sucede en la realidad que estamos estudiando y que queremos analizar, por ejemplo, una venta de nuestra empresa inmobiliaria. No tiene porqué ser un proceso de negocio, puede ser simplemente un concepto que se considere importante para la toma de decisiones. La **Dimensión** es la perspectiva desde la cual queremos analizar los hechos. Por ejemplo, una dimensión casi siempre presente en estos modelos es el tiempo. En el ejemplo anterior, puede ser interesante analizar las ventas sucedidas en un período marcado por la dimensión tiempo. Los **indicadores o medidas** son las propiedades de los hechos. Es decir, atributos de los hechos. En una supuesta venta, la cantidad de producto vendido sería una medida o indicador. A la hora de implementar físicamente los hechos y las dimensiones ambos conceptos serán tablas de base de datos.

En el modelado dimensional de hechos existen básicamente dos configuraciones: estrella y en copo de nieve. La principal diferencia entre ambos esquemas es que en copo de nieve se dispone de las tablas normalizadas. Esto suele ser interesante cuando tenemos una dimensión con varios niveles de jerarquía. Al crear más tablas y normalizarlas se ahorra espacio, sin embargo, el rendimiento de las consultas se ve perjudicado por el hecho de los *join* a realizar entre tablas de dimensiones para las consultas. El **esquema en estrella** se denomina así por disponer una tabla de hechos en el centro que se relaciona con las dimensiones con cardinalidad (1:m).

(4): **Ralph Kimball** define un almacén de datos como: "una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis". Wikipedia. http://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos#Definici.C3.B3n_de_Ralph_Kimball.

(5): **Ralph Kimball**. A Dimensional Modeling Manifesto. August 1997 http://www.kimballgroup.com/html/articles_search/articles1997/9708d15.html

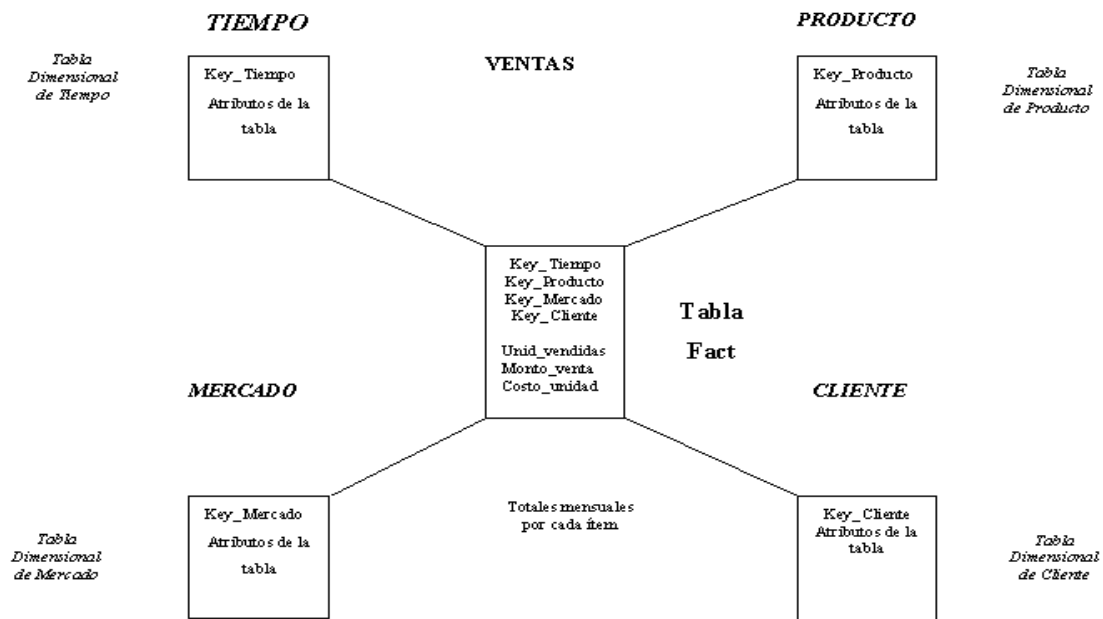


Figura 31. Ejemplo de diagrama de almacén de datos en estrella.

El **esquema en copo de nieve** (*snowflake*) es algo más complejo porque tiene las tablas normalizadas, si bien, también dispone la tabla de hechos en el centro y las dimensiones a su lado en cardinalidad (1:m).

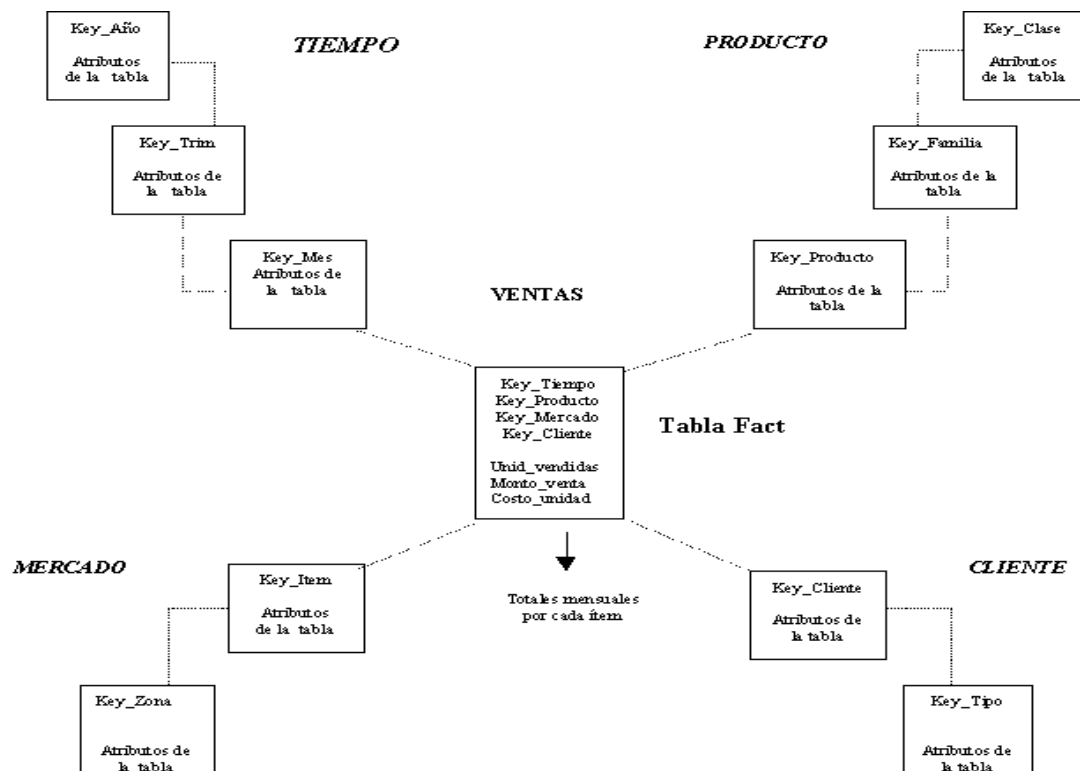


Figura 32. Ejemplo de diagrama de almacén de datos en copo de nieve.

Figuras tomadas de <http://www.inf.udec.cl/~revista/ediciones/edicion4/modmulti.PDF> donde se muestran los esquemas en estrella y en copo de nieve para un hecho con las dimensiones tiempo, producto, mercado y cliente.

Un concepto importante a tener en cuenta en este modelado es la granularidad o nivel de detalle en la tabla de hechos. La granularidad del hecho define las dimensiones que deberá tener el almacén de datos, y esto, lógicamente viene marcado por los requerimientos que tenemos que satisfacer con dicho almacén. Gráficamente, el elemento que describe perfectamente el modelado dimensional de hechos son los denominados **cubos OLAP** (*Online Analytical Processing*) donde cada cubo, a su vez, es un hecho analizable desde varias perspectivas.

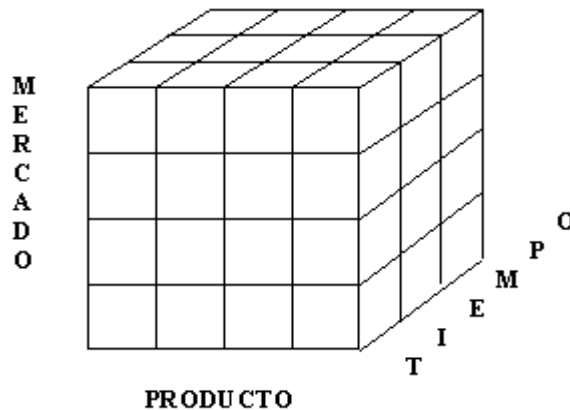


Figura 33. Ejemplo cubo OLAP de 3 dimensiones: tiempo, producto y mercado.

Para finalizar este breve repaso, quiero mencionar otro punto importante en Data Warehousing, que es la necesidad de extraer, transformar y cargar sobre nuestro almacén de datos la información a partir de unos datos originales, que normalmente recogeremos de las bases de datos transaccionales de negocio de la organización. Estos procesos son denominados procesos ETL (*Extract, transform and Load*). Tienen gran importancia, y de hecho, forman parte del ciclo de vida de cualquier proceso de generación de conocimiento a partir de datos como una etapa más.

Anexo 2. Código fuente fichero por lotes “GeneraDW.bat”

```
cls
echo off
echo -----
echo          GENERACION DEL ALMACEN DE DATOS INFORMACION INMOBILIARIA
echo          (EXTRACCION/TRANSFORMACION/CREACION/CARGA)
echo -----

rem      echo -----
rem      echo Se va a crear el usuario TFC / TFC para administrar Proyecto
rem      pause
rem      SQLPLUS system/uoc @pl_sql/CrearUser.sql

echo ***  Advertencia: En caso de que el DW ya exista, sera regenerado. ***

echo.
echo -----
echo Subproceso 1 ETL. Generar archivos texto desde fichero origen.
echo      pause

echo      java -cp bin TransformacionPrevia 2006 01 1

echo      if ERRORLEVEL 1 goto FINAL

echo.
echo *** OK. Archivos texto temporales generados correctamente.
echo      pause

echo.
echo -----
echo Subproceso 2 ETL. Crear Tablas Temporales Oracle
echo      pause

SQLPLUS TFC/TFC @pl_sql/Crear_Tablas_Temp.sql

echo.
echo *** OK. Tablas temporales creadas correctamente.
echo      pause

echo.
echo -----
echo Subproceso 3 ETL. Cargar Tablas Temporales Oracle
echo      pause

DEL bad\*.bad
DEL log\*.log
DEL dsc\*.dsc

SQLLDR USERID=TFC/TFC CONTROL=ctl/Inmuebles.ctl LOG=log/inmuebles.log

echo.
echo *** OK. Carga de tabla temporal de Inmuebles realizada correctamente.
echo      pause

SQLLDR USERID=TFC/TFC CONTROL=ctl/Precios.ctl LOG=log/precio.log

echo.
echo *** OK. Carga de tabla temporal de Precios realizada correctamente.
echo      pause

SQLLDR USERID=TFC/TFC CONTROL=ctl/InmueblesVendidos.ctl LOG=log/inmueblesvendidos.log

echo.
echo *** OK.Carga de tabla temporal de Ventas de Inmuebles realizada correctamente.
echo      pause

echo.
echo -----
echo Subproceso 4 ETL. Crear Tablas Definitivas Oracle
echo      pause

SQLPLUS TFC/TFC @pl_sql/Crear_Tablas_Definitivas.sql
```



```
echo.
echo *** OK. Tablas definitivas creadas correctamente.
      pause

echo.
echo -----
echo Subproceso 5 ETL. Cargar Tablas Definitivas Oracle
echo *** ¡Atención! Este proceso puede tardar varios minutos (10 min. aprox).
      pause

SQLPLUS TFC/TFC @pl_sql/Carga_Modelo_Datos_1.sql
SQLPLUS TFC/TFC @pl_sql/Carga_Modelo_Datos_2.sql

echo.
echo *** OK. Carga de tablas Definitivas realizada correctamente.
echo      Ahora se van a eliminar Archivos texto y Tablas temporales.
      pause

DEL NewCatalogo.txt
DEL NewPrecios.txt
DEL NewInmueblesVendidos.txt

SQLPLUS TFC/TFC @pl_sql/Eliminar_Tablas_Temp.sql

echo.
echo *** Archivos texto y Tablas temporales ELIMINADOS.
echo      Pulse INTRO para finalizar.
      pause >nul

GOTO endpgm

:FINAL
echo.
echo *** ERROR. Proceso para la Generación y Carga del Almacén de datos ABORTADO.
      pause

:ENDPGM
```

Anexo 3. PL/SQL “Crear_Tablas_Temp.sql”

```
DROP TABLE "INMUEBLES_TMP";

CREATE TABLE INMUEBLES_TMP (
    REFER VARCHAR(15),
    TIPO VARCHAR(40),
    PLANTA VARCHAR(5),
    NUM_DORM NUMBER (5,0),
    METROS NUMBER (10,0),
    PAIS VARCHAR(100),
    PROVIN VARCHAR (100),
    MUNICIP VARCHAR(100),
    AREA VARCHAR(100),
    DISTRITO VARCHAR(100),
    CONSTRAINT PK_inmuebles_tmp PRIMARY KEY (REFER)
);

DROP TABLE "PRECIOS_TMP";

CREATE TABLE PRECIOS_TMP (
    REFER VARCHAR(15),
    ANIO NUMBER (4,0),
    MES NUMBER (2,0),
    PRECIO NUMBER (15,0),
    CONSTRAINT PK_PRECIOS_TMP PRIMARY KEY (REFER, ANIO, MES)
);

DROP TABLE "INMUEBLESVENDIDOS_TMP";

CREATE TABLE INMUEBLESVENDIDOS_TMP (
    REFER VARCHAR(15),
    ANIO NUMBER (4,0),
    MES NUMBER (2,0),
    PR_VENTA NUMBER (15,0),
    CONSTRAINT PK_INMUEBLESVENDIDOS_TMP PRIMARY KEY (REFER, ANIO, MES)
);

COMMIT;
QUIT;
```

Anexo 4. Ficheros .ctl cargas SQL *Loader tablas temp

“Inmuebles.ctl”:

```
LOAD DATA
CHARACTERSET 'WE8MSWIN1252'
INFILE 'NewCatalogo.txt'
BADFILE 'bad/Inmuebles.bad'
DISCARDFILE 'dsc/Inmuebles.dsc'
REPLACE
INTO TABLE INMUEBLES_TMP
FIELDS TERMINATED BY ';' TRAILING NULLCOLS
(
    REFER,
    TIPO,
    PLANTA,
    NUM_DORM,
    METROS,
    PAIS,
    PROVIN,
    MUNICIP,
    AREA,
    DISTRITO
)
```

“Precios.ctl”:

```
LOAD DATA
CHARACTERSET 'WE8MSWIN1252'
INFILE 'NewPrecios.txt'
BADFILE 'bad/Precios.bad'
DISCARDFILE 'dsc/Precios.dsc'
REPLACE
INTO TABLE PRECIOS_TMP
FIELDS TERMINATED BY ';' TRAILING NULLCOLS
(
    REFER,
    ANIO,
    MES,
    PRECIO
)
```

“Inmueblesvendidos.ctl”:

```
LOAD DATA
CHARACTERSET 'WE8MSWIN1252'
INFILE 'NewInmueblesVendidos.txt'
BADFILE 'bad/InmueblesVendidos.bad'
DISCARDFILE 'dsc/InmueblesVendidos.dsc'
REPLACE
INTO TABLE INMUEBLESVENDIDOS_TMP
FIELDS TERMINATED BY ';' TRAILING NULLCOLS
(
    REFER,
    ANIO,
    MES,
    PR_VENTA
)
```

Anexo 5. PL/SQL “Crear_Tablas_Definitivas.sql”

```
DROP TABLE "ZONAS";

CREATE TABLE ZONAS (
    PAIS VARCHAR (100),
    PROVIN VARCHAR (100),
    MUNICIP VARCHAR(100),
    AREA VARCHAR (100),
    DISTRITO VARCHAR (100),
    CONSTRAINT PK_ZONAS PRIMARY KEY (PAIS, PROVIN, MUNICIP, AREA, DISTRITO)
);

DROP TABLE "PERIODOS";

CREATE TABLE PERIODOS (
    ANIO NUMBER (4,0),
    MES NUMBER (2,0),
    CONSTRAINT PK_PERIODOS PRIMARY KEY (ANIO, MES)
);

DROP TABLE "TIPO_INMUEBLE";

CREATE TABLE TIPO_INMUEBLE (
    TIPO VARCHAR(40),
    CONSTRAINT PK_TIPO_INMUEBLE PRIMARY KEY (TIPO)
);

DROP TABLE "INMUEBLES";

CREATE TABLE INMUEBLES (
    REFER VARCHAR(15),
    PLANTA VARCHAR(5),
    NUM_HAB NUMBER (5,0),
    METROS NUMBER (10,0),
    CONSTRAINT PK_INMUEBLES PRIMARY KEY (REFER)
);

DROP TABLE "SITU_MES_INMUEBLES";

CREATE TABLE SITU_MES_INMUEBLES (
    ANIO NUMBER (4,0),
    MES NUMBER (2,0),
    PAIS VARCHAR (100),
    PROVIN VARCHAR (100),
    MUNICIP VARCHAR(100),
    AREA VARCHAR (100),
    DISTRITO VARCHAR (100),
    TIPO VARCHAR(40),
    REFER VARCHAR(15),
    NUM_HAB NUMBER (5,0),
    METROS NUMBER (10,0),
    PLANTA VARCHAR (5),
    PRECIO_ACT NUMBER (15,0),
    PRECIO_ANT NUMBER (15,0),
    PRECIO_VENTA NUMBER (15,0),
    MESES_OFERTA NUMBER (5,0),
    PR_METRO NUMBER (10,0),
    CONSTRAINT PK_SITU_INMUEBLES PRIMARY KEY (ANIO, MES, PAIS, PROVIN, MUNICIP,
        AREA, DISTRITO, TIPO, REFER)
);

COMMIT;
QUIT;
```

Anexo 6. Carga Tabla Definitivas

“Carga_Modelo_Datos_1.sql”:

```

/* ===== */
/* Script para la carga de las tablas del modelo MultiDimensional */
/* diseñado: TIPO_INMUEBLE, INMUEBLE, MESES, ZONAS Y PERIODOS */
/* ===== */
DECLARE
    v_Refer          INMUEBLES_TMP.REFER%TYPE;
    v_Planta         INMUEBLES_TMP.PLANTA%TYPE;
    v_NumHab         INMUEBLES_TMP.NUM_DORM%TYPE;
    v_Metros         INMUEBLES_TMP.METROS%TYPE;
    v_Pais           INMUEBLES_TMP.PAIS%TYPE;
    v_Provin         INMUEBLES_TMP.PROVIN%TYPE;
    v_Municip        INMUEBLES_TMP.MUNICIP%TYPE;
    v_Area           INMUEBLES_TMP.AREA%TYPE;
    v_Distrito       INMUEBLES_TMP.DISTRITO%TYPE;
    v_NumReg         NUMBER;

    -- Cursor para recuperar registro de Inmuebles_tmp:
    CURSOR r_InmueblesTMP IS SELECT REFER, PLANTA, NUM_DORM, METROS FROM
INMUEBLES_TMP;

    -- Cursor para recuperar registro de Inmuebles_tmp:
    CURSOR r_ZonasTMP IS SELECT UNIQUE PAIS, PROVIN, MUNICIP, AREA, DISTRITO FROM
INMUEBLES_TMP;

BEGIN

    -- actualizo campos de grupos territoriales no informados para que sean blancos
    y no nulos:
    update inmuebles_tmp set provin = ' ' where provin is null;
    update inmuebles_tmp set municip = ' ' where municip is null;
    update inmuebles_tmp set area = ' ' where area is null;
    update inmuebles_tmp set distrito = ' ' where distrito is null;

    -- Correcciones:
    update inmuebles_tmp set metros = 779 where refer = 'VP0000000647987' AND metros
= 77895;
    update inmuebles_tmp set num_dorm = -1 where num_dorm = 0 AND tipo <>
'estudios';
    update inmuebles_tmp set metros = -1 where metros < 25;
    -- Mejora para la visualización de documentos analíticos:
    update inmuebles_tmp set municip = 'Sin informar' where municip = ' ' AND area =
' ' and distrito = ' ';
    COMMIT;

    -- Inserción de diferentes TIPOS de INMUEBLES:
    INSERT INTO tipo_inmueble (tipo_inmueble.TIPO) SELECT
DISTINCT(inmuebles_tmp.TIPO) FROM inmuebles_tmp;
    COMMIT;

    -- Inserción de diferentes PERIODOS:
    INSERT INTO periodos SELECT UNIQUE anio, mes FROM PRECIOS_TMP order by anio,
mes;
    COMMIT;

    -- Inserción de diferentes ZONAS:
    OPEN r_ZonasTMP;
    LOOP
        FETCH r_ZonasTMP INTO v_Pais, v_Provin, v_Municip, v_Area, v_Distrito;

        EXIT WHEN r_ZonasTMP%NOTFOUND;

        INSERT INTO zonas VALUES (v_Pais, v_Provin, v_Municip, v_Area,
v_Distrito, 0, 0);

    END LOOP;
    CLOSE r_ZonasTMP;
    COMMIT;

```

```
-- Inserción de INMUEBLES:
OPEN r_InmueblesTMP;
LOOP
    FETCH r_InmueblesTMP INTO v_Refer, v_Planta, v_NumHab, v_Metros;

    EXIT WHEN r_InmueblesTMP%NOTFOUND;

    -- Inserción en INMUEBLES:
    INSERT INTO inmuebles VALUES (v_Refer, v_Planta, v_NumHab, v_Metros);

END LOOP;
CLOSE r_InmueblesTMP;
COMMIT;

UPDATE zonas SET latitud = 38.344887442462806, longitud = -0.457305908203125
WHERE provin = 'alicante';
UPDATE zonas SET latitud = 42.511336337601854, longitud = 1.5277862548828125
WHERE provin = 'andorra';
UPDATE zonas SET latitud = 41.38814294931545, longitud = 2.1697998046875 WHERE
provin = 'barcelona';
UPDATE zonas SET latitud = 39.986068, longitud = -0.036024 WHERE provin =
'castellón';
UPDATE zonas SET latitud = 41.981796, longitud = 2.8237 WHERE provin = 'girona';
UPDATE zonas SET latitud = 38.78941577989049, longitud = 1.5106201171875 WHERE
provin = 'ibiza-formentera';
UPDATE zonas SET latitud = 41.61441561788573, longitud = 0.626220703125 WHERE
provin = 'lleida';
UPDATE zonas SET latitud = 39.695263, longitud = 3.017571 WHERE provin =
'mallorca';
UPDATE zonas SET latitud = 39.949629, longitud = 4.110445 WHERE provin =
'menorca';
UPDATE zonas SET latitud = 41.118663, longitud = 1.24533 WHERE provin =
'tarragona';
UPDATE zonas SET latitud = 39.470239, longitud = -0.376805 WHERE provin =
'valencia';
UPDATE zonas SET latitud = 40.423167191915596, longitud = -3.7109756469726562
WHERE provin = 'madrid';

INSERT INTO meses VALUES (1, 'Enero', 'Ene');
INSERT INTO meses VALUES (2, 'Febrero', 'Feb');
INSERT INTO meses VALUES (3, 'Marzo', 'Mar');
INSERT INTO meses VALUES (4, 'Abril', 'Abr');
INSERT INTO meses VALUES (5, 'Mayo', 'May');
INSERT INTO meses VALUES (6, 'Junio', 'Jun');
INSERT INTO meses VALUES (7, 'Julio', 'Jul');
INSERT INTO meses VALUES (8, 'Agosto', 'Ago');
INSERT INTO meses VALUES (9, 'Septiembre', 'Sep');
INSERT INTO meses VALUES (10, 'Octubre', 'Oct');
INSERT INTO meses VALUES (11, 'Noviembre', 'Nov');
INSERT INTO meses VALUES (12, 'Diciembre', 'Dic');
COMMIT;

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('No se ha podido realizar la carga de datos en
tablas DEFINITIVAS');

END;
/

QUIT;
```

“Carga_Modelo_Datos_2.sql”:

```

/* ===== */
/* Script para la carga de las tablas del modelo MultiDimensional */
/*          SITU_MES_INMUEBLES (HECHO) */
/* ===== */

DECLARE
    v_Refer          PRECIOS_TMP.REFER%TYPE;
    v_Anio           PRECIOS_TMP.ANIO%TYPE;
    v_Mes            PRECIOS_TMP.MES%TYPE;
    v_Precio         PRECIOS_TMP.PRECIO%TYPE;
    v_Pais           INMUEBLES_TMP.PAIS%TYPE;
    v_Provin         INMUEBLES_TMP.PROVIN%TYPE;
    v_Municip        INMUEBLES_TMP.MUNICIP%TYPE;
    v_Area           INMUEBLES_TMP.AREA%TYPE;
    v_Distrito       INMUEBLES_TMP.DISTRITO%TYPE;
    v_Tipo           INMUEBLES_TMP.TIPO%TYPE;
    v_Metros         INMUEBLES_TMP.METROS%TYPE;
    v_Planta         INMUEBLES_TMP.PLANTA%TYPE;
    v_NumHab         INMUEBLES_TMP.NUM_DORM%TYPE;

    v_Aux_Precio     NUMBER(15,0);
    v_Aux_Refer      VARCHAR(15);
    v_Precio_Venta   INMUEBLESVENDIDOS_TMP.PR_VENTA%TYPE;
    v_NumMeses       NUMBER;
    v_NumReg         NUMBER;
    v_Precio_Metro   NUMBER;
    v_En_Oferta      NUMBER;
    v_Vendido        NUMBER;
    v_TipoMetros     NUMBER;
    v_TipoNumHab     NUMBER;

    -- Cursor para recuperar registro de precios_tmp:
    CURSOR r_PreciosTMP IS
        SELECT REFER, ANIO, MES, PRECIO FROM PRECIOS_TMP;

BEGIN

    v_NumMeses := 0;
    v_Aux_Precio := 0;
    v_Aux_Refer := '#####';

    OPEN r_PreciosTMP;
    LOOP
        FETCH r_PreciosTMP INTO v_Refer, v_Anio, v_Mes, v_Precio;

        EXIT WHEN r_PreciosTMP%NOTFOUND;

        IF v_Aux_Refer <> v_Refer THEN
            SELECT pais, provin, municip, area, distrito, tipo, metros,
planta, num_dorm
                                INTO v_Pais, v_Provin, v_Municip, v_Area, v_Distrito,
v_Tipo, v_Metros, v_Planta, v_NumHab
                                FROM inmuebles_tmp WHERE refer = v_Refer;
            v_NumMeses := 0;
            v_Aux_Refer := v_Refer;
            v_Aux_Precio := 0;
        END IF;

        -- Mejora para la visualización docs. analíticos:
        IF v_Municip = ' ' AND v_Area = ' ' AND v_Distrito = ' ' THEN
            v_Municip := v_Provin;
        END IF;

        -- Carga de tabla de Hechos.

        v_NumMeses := v_NumMeses + 1;

        v_Precio_Metro := 0;
        IF v_Metros > -1 THEN
            v_Precio_Metro := v_Precio / v_Metros;
        END IF;

        SELECT COUNT(*) INTO v_NumReg FROM inmueblesvendidos_tmp WHERE refer =

```

```

v_Refer AND anio = v_Anio AND mes = v_Mes;

        IF v_NumReg > 0 THEN
            SELECT pr_Venta INTO v_Precio_Venta FROM inmueblesvendidos_tmp
WHERE
            refer = v_Refer AND anio = v_Anio AND mes = v_Mes;
            v_En_Oferta := 0;
            v_Vendido := 1;
        ELSE
            v_Precio_Venta := 0;
            v_En_Oferta := 1;
            v_Vendido := 0;
        END IF;

        -- Categorización de Metros:
        v_TipoMetros := 0;
        IF v_Metros > -1 AND v_Metros <= 70 THEN
            v_TipoMetros := 1;
        END IF;
        IF v_Metros > 70 AND v_Metros <= 140 THEN
            v_TipoMetros := 2;
        END IF;
        IF v_Metros > 140 AND v_Metros <= 300 THEN
            v_TipoMetros := 3;
        END IF;
        IF v_Metros > 300 THEN
            v_TipoMetros := 4;
        END IF;

        -- Categorización de Número de habitaciones:
        v_TipoNumHab := 0;
        IF v_NumHab > -1 AND v_NumHab <= 2 THEN
            v_TipoNumHab := 1;
        END IF;
        IF v_NumHab > 2 AND v_NumHab <= 4 THEN
            v_TipoNumHab := 2;
        END IF;
        IF v_NumHab > 4 AND v_NumHab <= 9 THEN
            v_TipoNumHab := 3;
        END IF;
        IF v_NumHab > 9 THEN
            v_TipoNumHab := 4;
        END IF;

        INSERT INTO situ_mes_inmuebles VALUES (v_Anio, v_Mes, v_Pais, v_Provin,
v_Municip, v_Area, v_Distrito,
            v_Tipo, v_Refer, v_NumHab, v_Metros, v_Planta, v_Precio,
v_Aux_Precio, v_Precio_Venta,
            v_NumMeses, v_Precio_Metro, v_En_Oferta, v_Vendido, v_TipoMetros,
v_TipoNumHab);

        v_Aux_Precio := v_Precio;

    END LOOP;
    CLOSE r_PreciosTMP;

    -- Correcciones sobre precios cargados por incoherencias entre meses. Se ponen a
    cero:
    update situ_mes_inmuebles set precio_act = 0, precio_ant = 0, precio_venta = 0,
pr_metro = 0,
        vendido = 0, en_oferta = 0
    WHERE refer IN
        (select distinct(refer) from situ_mes_inmuebles where
            ((precio_act > (2 * precio_ant)) OR
            (precio_ant > (2 * precio_act))) AND
            precio_ant > 0);

    COMMIT;

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('No se ha podido realizar la carga de datos en
tablas DEFINITIVAS');
    END;

/QUIT;

```


Anexo 7. Código fuente fichero por lotes “Carga_Mensual.bat”

```
cls
echo off
echo -----
echo      CARGA MENSUAL ALMACEN DE DATOS INFORMACION INMOBILIARIA
echo -----

echo.
echo -----
echo Subproceso 1 ETL. Generar archivos texto desde fichero origen.
pause

      SET /P ANIO=Introduzca el Anio de inicio de los datos:
      SET /P MES=Introduzca el Mes de inicio de los datos:

echo Anio tecleado: %ANIO%
echo Mes tecleado: %MES%

      java -cp bin TransformacionPrevia %ANIO% %MES% 2

      if ERRORLEVEL 1 goto FINAL

echo.
echo *** OK. Archivos texto temporales generados correctamente.
pause

echo.
echo -----
echo Subproceso 2 ETL. Crear Tablas Temporales Oracle
pause

SQLPLUS TFC/TFC @pl_sql/Crear_Tablas_Temp.sql

echo.
echo *** OK. Tablas temporales creadas correctamente.
pause

echo.
echo -----
echo Subproceso 3 ETL. Cargar Tablas Temporales Oracle
pause

DEL bad\*.bad
DEL log\*.log
DEL dsc\*.dsc

SQLLDR USERID=TFC/TFC CONTROL=ctl/Inmuebles.ctl LOG=log/inmuebles.log

echo.
echo *** OK. Carga de tabla temporal de Inmuebles realizada correctamente.
pause

SQLLDR USERID=TFC/TFC CONTROL=ctl/Precios.ctl LOG=log/precio.log

echo.
echo *** OK. Carga de tabla temporal de Precios realizada correctamente.
pause

echo.
echo -----
---
echo Subproceso 4 ETL. Carga Mensual Tablas DW
pause

SQLPLUS TFC/TFC @pl_sql/Carga_Mensual_1.sql

echo.
echo *** OK. Carga MENSUAL realizada correctamente.
echo      Ahora se van a eliminar Archivos texto y Tablas temporales.
pause
```

```
DEL NewCatalogo.txt
DEL NewPrecios.txt

SQLPLUS TFC/TFC @pl_sql/Eliminar_Tablas_Temp.sql

echo.
echo *** Archivos texto y Tablas temporales ELIMINADOS.
echo          Pulse INTRO para finalizar.
      pause >nul

GOTO endpgm

:FINAL
echo.
echo *** ERROR. Proceso para la Carga Mensual del Almacén de datos ABORTADO.
      pause

:ENDPGM
```

Anexo 8. Carga Mensual de Mantenimiento

```

/* ===== */
/* Script para la carga MENSUAL de las tablas del modelo MultiDimensional */
/* TIPO_INMUEBLE, INMUEBLE, ZONAS, PERIODOS y SITU_MES_INMUEBLES */
/* ===== */
DECLARE
    v_Refer          INMUEBLES_TMP.REFER%TYPE;
    v_Tipo           INMUEBLES_TMP.TIPO%TYPE;
    v_Planta        INMUEBLES_TMP.PLANTA%TYPE;
    v_NumHab        INMUEBLES_TMP.NUM_DORM%TYPE;
    v_Metros        INMUEBLES_TMP.METROS%TYPE;
    v_Pais          INMUEBLES_TMP.PAIS%TYPE;
    v_Provin        INMUEBLES_TMP.PROVIN%TYPE;
    v_Municip       INMUEBLES_TMP.MUNICIP%TYPE;
    v_Area          INMUEBLES_TMP.AREA%TYPE;
    v_Distrito      INMUEBLES_TMP.DISTRITO%TYPE;
    v_Anio          PRECIOS_TMP.ANIO%TYPE;
    v_Mes           PRECIOS_TMP.MES%TYPE;
    v_Precio        PRECIOS_TMP.PRECIO%TYPE;
    v_Precio_Metro  NUMBER;
    v_NumReg        NUMBER;
    v_Aux_Precio    NUMBER(15,0);
    v_Precio_Venta  NUMBER(15,0);
    v_NumMeses      NUMBER;
    v_TipoMetros    NUMBER;
    v_TipoNumHab    NUMBER;

    -- Cursor para recuperar registro de Inmuebles_tmp:
    CURSOR r_InmueblesTMP IS SELECT REFER, TIPO, PLANTA, NUM_DORM, METROS, PAIS,
        PROVIN, MUNICIP, AREA, DISTRITO FROM INMUEBLES_TMP;

    -- Cursor para recuperar registro de precios_tmp condicional:
    CURSOR r_Precios IS SELECT ANIO, MES, PRECIO FROM PRECIOS_TMP WHERE REFER =
v_Refer;

    -- Cursor para detectar los distintos períodos que llegan:
    -- CURSOR r_Periodos IS SELECT UNIQUE anio, mes FROM precios_tmp;

    -- Cursor para lectura de inmuebles de los que no se han recibido datos:
    -- CURSOR r_Inmuebles IS SELECT REFER, PLANTA, NUM_HAB, METROS FROM INMUEBLES
        --WHERE REFER NOT IN (SELECT REFER FROM INMUEBLES_TMP);

BEGIN

    -- actualizo campos de grupos territoriales no informados para que sean blancos
    y no nulos:
    update inmuebles_tmp set provin = ' ' where provin is null;
    update inmuebles_tmp set municip = ' ' where municip is null;
    update inmuebles_tmp set area = ' ' where area is null;
    update inmuebles_tmp set distrito = ' ' where distrito is null;
    update precios_tmp set precio = 0 where precio is null;

    -- Correcciones:
    update inmuebles_tmp set metros = 779 where refer = 'VP0000000647987' AND metros
= 77895;
    update inmuebles_tmp set num_dorm = -1 where num_dorm = 0 AND tipo <>
'estudios';
    update inmuebles_tmp set metros = -1 where metros < 25;
    -- Mejora para la visualización de documentos analíticos:
    update inmuebles_tmp set municip = 'Sin informar' where municip = ' ' AND area =
' ' and distrito = ' ';
    COMMIT;

    insert INTO periodos select UNIQUE anio, mes FROM precios_tmp WHERE
        anio||mes NOT IN (select anio||mes from periodos);

    -- Lectura de INMUEBLES:
    OPEN r_InmueblesTMP;
    LOOP
        FETCH r_InmueblesTMP INTO v_Refer, v_Tipo, v_Planta, v_NumHab, v_Metros,
v_Pais, v_Provin, v_Municip,
        v_Area, v_Distrito;
        EXIT WHEN r_InmueblesTMP%NOTFOUND;

```

```

-- Grabo Tipo sino existe:
SELECT COUNT(*) INTO v_NumReg FROM TIPO_INMUEBLE WHERE tipo = v_Tipo;

IF v_NumReg <= 0 THEN
    INSERT INTO tipo_inmueble VALUES (v_Tipo);
END IF;

-- Grabo Zona sino existe:
SELECT COUNT(*) INTO v_NumReg FROM ZONAS WHERE pais = v_Pais AND provin =
v_Provin AND municip = v_Municip
    AND area = v_Area AND distrito = v_Distrito;
IF v_NumReg <= 0 THEN
    INSERT INTO zonas VALUES (v_Pais, v_Provin, v_Municip, v_Area,
v_Distrito, 0, 0);
END IF;

-- Grabo Inmueble sino existe. Si no existe, debe traer al menos un
precio.
SELECT COUNT(*) INTO v_NumReg FROM INMUEBLES WHERE refer = v_Refer;

IF v_NumReg <= 0 THEN
    INSERT INTO inmuebles VALUES (v_Refer, v_Planta, v_NumHab,
v_Metros);

    -- Grabo nuevos precios en tabla de Hechos:
    v_Aux_Precio := 0;
    v_Precio_Venta := 0;
    v_NumMeses := 0;

    OPEN r_Precios;
    LOOP
        FETCH r_Precios INTO v_Anio, v_Mes, v_Precio;
        EXIT WHEN r_Precios%NOTFOUND;

        IF (v_Precio = 0) THEN
            -- actualizo precio de venta
            UPDATE situ_mes_inmuebles SET PRECIO_VENTA =
PRECIO_ACT, En_OFERTA = 0, VENDIDO = 1
                WHERE refer = v_Refer AND anio||mes IN (select
max(anio||mes)
                    FROM situ_mes_inmuebles where refer = v_Refer);

        ELSE
            -- Categorización de Metros:
            v_TipoMetros := 0;
            IF v_Metros > -1 AND v_Metros <= 70 THEN
                v_TipoMetros := 1;
            END IF;
            IF v_Metros > 70 AND v_Metros <= 140 THEN
                v_TipoMetros := 2;
            END IF;
            IF v_Metros > 140 AND v_Metros <= 300 THEN
                v_TipoMetros := 3;
            END IF;
            IF v_Metros > 300 THEN
                v_TipoMetros := 4;
            END IF;

            -- Categorización de Número de habitaciones:
            v_TipoNumHab := 0;
            IF v_NumHab > -1 AND v_NumHab <= 2 THEN
                v_TipoNumHab := 1;
            END IF;
            IF v_NumHab > 2 AND v_NumHab <= 4 THEN
                v_TipoNumHab := 2;
            END IF;
            IF v_NumHab > 4 AND v_NumHab <= 9 THEN
                v_TipoNumHab := 3;
            END IF;
            IF v_NumHab > 9 THEN
                v_TipoNumHab := 4;
            END IF;

            v_NumMeses := v_NumMeses + 1;
        
```

```

                                v_Precio_Metro := 0;
                                IF v_Metros > -1 THEN
                                    v_Precio_Metro := v_Precio / v_Metros;
                                END IF;

                                INSERT INTO situ_mes_inmuebles VALUES (v_Anio,
                                v_Mes, v_Pais, v_Provin, v_Municip,
                                    v_Area, v_Distrito, v_Tipo, v_Refer, v_NumHab,
                                v_Metros, v_Planta, v_Precio,
                                    v_Aux_Precio, v_Precio_Venta, v_NumMeses,
                                v_Precio_Metro, 1, 0, v_TipoMetros, v_TipoNumHab);
                                    v_Aux_Precio := v_Precio;

                                END IF;
                                END LOOP;
                                CLOSE r_Precios;
ELSE
    -- Inmuebles Ya existe, se graban precios,pero hay que recuperar algún
    dato del último registro de tabla de hechos.
    OPEN r_Precios;
    LOOP
        FETCH r_Precios INTO v_Anio, v_Mes, v_Precio;
        EXIT WHEN r_Precios%NOTFOUND;

        IF (v_Precio = 0) THEN
            -- actualizo precio de venta
            UPDATE situ_mes_inmuebles SET PRECIO_VENTA =
PRECIO_ACT , En_OFERTA = 0, VENDIDO = 1
            WHERE refer = v_Refer AND anio||mes IN (select
max(anio||mes)
            FROM situ_mes_inmuebles where refer = v_Refer);

        ELSE
            SELECT precio_act, meses_oferta, tipometros,
tiponumhab INTO v_Aux_Precio, v_NumMeses, v_TipoMetros, v_TipoNumHab
            FROM situ_mes_inmuebles WHERE refer =
v_Refer AND
            anio||mes IN (select max(anio||mes) FROM
situ_mes_inmuebles where refer = v_Refer);

            v_NumMeses := v_NumMeses + 1;
            v_Precio_Metro := 0;
            IF v_Metros > -1 THEN
                v_Precio_Metro := v_Precio / v_Metros;
            END IF;
            v_Precio_Venta := 0;

            INSERT INTO situ_mes_inmuebles VALUES (v_Anio,
            v_Mes, v_Pais, v_Provin, v_Municip,
                v_Area, v_Distrito, v_Tipo, v_Refer, v_NumHab,
            v_Metros, v_Planta, v_Precio,
                v_Aux_Precio, v_Precio_Venta, v_NumMeses,
            v_Precio_Metro, 1, 0, v_TipoMetros, v_TipoNumHab);
            END IF;

        END LOOP;
        CLOSE r_Precios;
    END IF;
END LOOP;
CLOSE r_InmueblesTMP;
-- Grabar precio de los inmuebles que no han venido en carga mensual
-- Es el mismo precio. Ojo, que pueden haber venido varios meses.
-- Se comenta todo este código pues SI vendrá el precio en fichero mensual
-- Además, debería optimizarse.
/*
OPEN r_Periodos;
LOOP
    FETCH r_Periodos INTO v_Anio, v_Mes;
    EXIT WHEN r_Periodos%NOTFOUND;

    OPEN r_Inmuebles;
    LOOP

        FETCH r_Inmuebles INTO v_Refer, v_Planta, v_NumHab, v_Metros;
        EXIT WHEN r_Inmuebles%NOTFOUND;

```

```

        SELECT pais, provin, municip, area, distrito, tipo, precio_act,
precio_ant, precio_venta,
        meses_oferta, pr_metro, tipometros, tiponumhab
        INTO v_Pais, v_Provin, v_Municip, v_Area, v_Distrito, v_Tipo,
v_Precio, v_Aux_Precio, v_Precio_Venta,
        v_NumMeses, v_Precio_Metro, v_TipoMetros, v_TipoNumHab FROM
situ_mes_inmuebles WHERE refer = v_Refer AND
        anio||mes IN (select max(anio)||mes) FROM situ_mes_inmuebles
where refer = v_Refer);

        v_NumMeses := v_NumMeses + 1;
        INSERT INTO situ_mes_inmuebles VALUES (v_Anio, v_Mes, v_Pais,
v_Provin, v_Municip,
        v_Area, v_Distrito, v_Tipo, v_Refer, v_NumHab,
v_Metros, v_Planta, v_Precio,
        v_Aux_Precio, v_Precio_Venta, v_NumMeses,
v_Precio_Metro, 1, 0, v_TipoMetros, v_TipoNumHab);

        END LOOP;
        CLOSE r_Inmuebles;
    END LOOP;
    CLOSE r_Periodos;
    */
    COMMIT;

    UPDATE zonas SET latitud = 38.344887442462806, longitud = -0.457305908203125
WHERE provin = 'alicante';
    UPDATE zonas SET latitud = 42.511336337601854, longitud = 1.5277862548828125
WHERE provin = 'andorra';
    UPDATE zonas SET latitud = 41.38814294931545, longitud = 2.1697998046875 WHERE
provin = 'barcelona';
    UPDATE zonas SET latitud = 39.986068, longitud = -0.036024 WHERE provin =
'castellón';
    UPDATE zonas SET latitud = 41.981796, longitud = 2.8237 WHERE provin = 'girona';
    UPDATE zonas SET latitud = 38.78941577989049, longitud = 1.5106201171875 WHERE
provin = 'ibiza-formentera';
    UPDATE zonas SET latitud = 41.61441561788573, longitud = 0.626220703125 WHERE
provin = 'lleida';
    UPDATE zonas SET latitud = 39.695263, longitud = 3.017571 WHERE provin =
'mallorca';
    UPDATE zonas SET latitud = 39.949629, longitud = 4.110445 WHERE provin =
'menorca';
    UPDATE zonas SET latitud = 41.118663, longitud = 1.24533 WHERE provin =
'tarragona';
    UPDATE zonas SET latitud = 39.470239, longitud = -0.376805 WHERE provin =
'valencia';
    UPDATE zonas SET latitud = 40.423167191915596, longitud = -3.7109756469726562
WHERE provin = 'madrid';

    -- Correcciones sobre precios cargados por incoherencias entre meses. Se ponen a
cero:
    update situ_mes_inmuebles set precio_act = 0, precio_ant = 0, precio_venta = 0,
pr_metro = 0,
        vendido = 0, en_oferta = 0
    WHERE refer IN
        (select distinct(refer) from situ_mes_inmuebles where
        ((precio_act > (2 * precio_ant)) OR
        (precio_ant > (2 * precio_act))) AND
        precio_ant > 0);

    COMMIT;

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('No se ha podido realizar la carga MENSUAL de
datos');

END;
/

QUIT;

```