

Construcción y explotación de un almacén de datos para el análisis del sistema de ventas de una distribuidora farmacéutica

Alexandre Pereiras Magariños

2º Ciclo Ingeniería Informática

Xavier Plaza Sierra

17 Junio 2013

A Emilia, por su paciencia durante todos estos años de estudios.

A mi padre, por su insistencia en finalizar la Ingeniería Informática.

A toda mi familia, por su apoyo incondicional.

Resumen

El Business Intelligence, Inteligencia de Negocios en castellano, se define como el conjunto de estrategias y herramientas que permiten a las organizaciones obtener conocimiento mediante el análisis de los datos existentes, y poder así realizar tomas de decisiones sobre los procesos de negocio que llevan a cabo.

Una de las formas de obtener este conocimiento es a través de la creación de los Almacenes de Datos, Data Warehouse en inglés. Un almacén de datos es un conjunto de estructuras que permiten integrar datos procedentes de múltiples departamento y sistemas de información, permitiendo el análisis de múltiples procesos de negocio desde diferentes perspectivas y utilizando definiciones comunes entre los diferentes departamentos de la organización.

Los almacenes de datos se han convertido en la actualidad en parte indispensable de los departamentos de Tecnologías de la Información de las empresas, ya que permiten a los directivos entender cómo la organización se comporta y tomar decisiones de una manera más eficiente.

Dada la importancia de los almacenes de datos a día de hoy, se ha seleccionado el área de Almacenes de Datos para la realización del proyecto final de carrera de Ingeniería Informática, siendo el presente documento la memoria final de la propuesta de proyecto para la construcción de un almacén de datos para el análisis del sistema de ventas de una distribuidora farmacéutica.

Abstract

Business Intelligence is defined as a set of strategies and tools that allow organizations to obtain knowledge by performing analysis on existing data, so decisions can be made on current business processes.

One of the ways of obtaining this knowledge is by creating a Data Warehouse. A data warehouse is a set of structures which integrates data from multiple departments and information systems, allowing any type of analysis of multiple business processes from different perspectives and using common definitions between different company departments.

Currently, data warehouses have become more and more important in the Information Technology departments, because they allow executives to understand how the organization behaves and to make more efficient decisions.

Given the importance of data warehouses, I've selected the Data Warehouse area to deliver the final project of Engineering Degree, being the current document the final report of author's proposal to construct a data warehouse based on sales data for a pharmaceutical product distributor company.

Índice

1.	Introducción.....	8
	Contexto y Justificación	8
	Objetivos Generales.....	8
	Objetivos Específicos	9
	Metodología de Trabajo	10
	Planificación del Proyecto.....	10
	Análisis de Riesgos	11
	Entregables	12
	Requisitos de Hardware / Software.....	12
	Breve Descripción del Resto de Apartados.....	13
2.	Análisis	13
	Estudio de los Ficheros de Datos	13
	Datos Históricos	19
	Casos de Uso	19
	Enfoque de Diseño	19
	El Hecho	22
	La Granularidad.....	22
	Las Dimensiones. Atributos	23
	Identificación de las Medidas	25
	Diseño Conceptual	25
	Matriz de Dimensiones y Medidas.....	26
3.	Diseño	27
	Introducción a Microsoft SQL Server	27
	Diseño de la Arquitectura de Software.....	28
	Diseño de la Arquitectura de Hardware	29
	Diseño de la Base de Datos.....	30
	Diseño del Proceso de Carga.....	37
	Diseño del Cubo de Datos.....	43
	Diseño de Informes	47
4.	Conclusiones	57
5.	Líneas de Evolución Futura	58
6.	Glosario	59
7.	Bibliografía	59
8.	Apéndice A – Código SQL.....	60

Script SQL de creación de la base de datos	60
Script SQL de creación de los esquemas.....	61
Script SQL de creación de la cuenta de servicio “dw_etl”	61
Script SQL de creación de la cuenta de servicio “dw_consulta”	62
Script SQL de creación de tablas de control	62
Script SQL de creación de tablas temporales	63
Script SQL de creación de dimensiones	64
Script SQL de creación de tablas de hechos	70
Script SQL de creación de funciones.....	73
Script SQL de creación de procedimientos almacenados.....	73
Script SQL de creación de vistas	75
Script SQL de carga de fechas	75
9. Apéndice B – Ficheros de Configuración	76
Fichero de Configuración de Conexiones	76
Fichero de Configuración de Parámetros	77

Índice de Ilustraciones

<i>Ilustración 1. Diagrama de Gantt</i>	11
<i>Ilustración 2. Casos de Uso</i>	19
<i>Ilustración 3. Modelo en Estrella</i>	20
<i>Ilustración 4. Cubo OLAP</i>	20
<i>Ilustración 5. Diagrama Conceptual</i>	26
<i>Ilustración 6. Diagrama de Arquitectura de Software</i>	28
<i>Ilustración 7. Iconos para iniciar / parar los servicios Microsoft BI</i>	29
<i>Ilustración 8. Diagrama de Arquitectura de Hardware</i>	30
<i>Ilustración 9. Modelo Entidad-Relación</i>	30
<i>Ilustración 10. Propiedades de Dimensión Fecha</i>	31
<i>Ilustración 11. Propiedades de Dimensión Almacén</i>	31
<i>Ilustración 12. Propiedades de Dimensión Ruta Distribución</i>	31
<i>Ilustración 13. Propiedades de Dimensión Producto</i>	32
<i>Ilustración 14. Propiedades de Dimensión Cliente</i>	32
<i>Ilustración 15. Propiedades de Dimensión Región</i>	32
<i>Ilustración 16. Propiedades de Dimensión Tipo de Orden</i>	33
<i>Ilustración 17. Propiedades de Dimensión Fabricante</i>	33
<i>Ilustración 18. Propiedades de Dimensión Causa Crédito</i>	33
<i>Ilustración 19. Propiedades de Dimensión Categoría de Producto</i>	33
<i>Ilustración 20. Propiedades de Hechos Presupuesto</i>	34
<i>Ilustración 21. Propiedades de Hechos Línea Pedido</i>	34
<i>Ilustración 22. Acceso a SQL Server con usuario "Administrator"</i>	35
<i>Ilustración 23. Acceso a SQL Server con usuario "sa"</i>	35
<i>Ilustración 24. Patrón de Proceso de Carga de un Almacén de Datos</i>	37
<i>Ilustración 25. Paquete "Ventas_Maestro.dtsx"</i>	38
<i>Ilustración 26. Configuraciones del paquete maestro</i>	39
<i>Ilustración 27. Configuraciones del paquete de carga de la dimensión Cliente</i>	40
<i>Ilustración 28. Vista de datos de Ventas y Presupuesto</i>	44
<i>Ilustración 29. Dimensiones en Analysis Services</i>	45
<i>Ilustración 30. Combinaciones posibles de dimensiones y medidas en el cubo de datos</i>	46
<i>Ilustración 31. Procesado de dimensiones y cubo en Analysis Services</i>	46
<i>Ilustración 32. Base de datos "PFC Ventas AS" en Analysis Services</i>	47
<i>Ilustración 33. Ejemplo de informe en el browser de Analysis Service,</i>	47
<i>Ilustración 34. Carpetas e informes dentro de Reporting Services</i>	48
<i>Ilustración 35. Informe 01. Ventas vs Presupuesto por Almacén y Mes Venta</i>	48
<i>Ilustración 36. Informe 02. Ventas vs Presupuesto por Categoría y Mes Venta</i>	49
<i>Ilustración 37. Informe 03. Ventas Actuales vs Ventas Actuales -1 por Almacén y Categoría</i>	49
<i>Ilustración 38. Informe 04. 20 productos con más/menos demanda en Ventas 2012 por Categoría</i>	50
<i>Ilustración 39. Informe 05. 20 clientes más/menos demandantes en Ventas 2012 por Categoría</i>	51
<i>Ilustración 40. Informe 05. 20 clientes más/menos demandantes en Ventas 2012 para "PHARMACY"</i>	52
<i>Ilustración 41. Informe 06. Informe Operacional por Año de Venta para el año 2012</i>	53
<i>Ilustración 42. Informe 07. 20 rutas con más/menos beneficio Año 2012</i>	54
<i>Ilustración 43. Informe 08. Ventas y Margen por Región.</i>	55
<i>Ilustración 44. Informe 09. Causas Crédito en Ventas</i>	56
<i>Ilustración 45. Informe 10. Ventas y Margen por Distribuidor y Fabricante</i>	57

Índice de Tablas

<i>Tabla 1. Hitos</i>	10
<i>Tabla 2. Fichero de Clientes</i>	16
<i>Tabla 3. Fichero de Productos</i>	16
<i>Tabla 4. Fichero de Fabricantes</i>	17
<i>Tabla 5. Fichero de Categorías de Producto</i>	17
<i>Tabla 6. Fichero de Regiones</i>	17
<i>Tabla 7. Fichero de Rutas de Distribución</i>	17
<i>Tabla 8. Fichero de Tipos de Orden</i>	17
<i>Tabla 9. Fichero de Causa de Créditos</i>	18
<i>Tabla 10. Fichero de Líneas de Pedido Año 2012</i>	18
<i>Tabla 11. Fichero de Presupuesto</i>	18
<i>Tabla 12. Ejemplo de Dimensión Tipo II</i>	21

1. Introducción

Contexto y Justificación

El proyecto final de carrera, en adelante PFC, tiene como objetivo final la consolidación de todos los conocimientos adquiridos a lo largo de los estudios de Ingeniería Informática. El área elegida ha sido **Almacenes de Datos**, mediante la propuesta personal de un proyecto que ha permitido al alumno poner en práctica los conocimientos de asignaturas como *Bases de Datos* o *Sistemas de Gestión de Bases de Datos*, y de familiarizarse con uno de los productos Business Intelligence más utilizados en el mercado actual: **Microsoft SQL Server 2012**.

El proyecto fin de carrera presentado consiste en la **construcción y explotación de un almacén de datos para el análisis del sistema de ventas de una distribuidora farmacéutica**, que se evaluará como trabajo final de los estudios de Ingeniería Informática, curso 2012/2013 (Febrero) de la Universitat Oberta de Catalunya, y cuyo enunciado se presenta a continuación.

PharmaDis¹ es una empresa dedicada a la distribución de productos farmacéuticos al por mayor. Es la mayor distribuidora de medicamentos de la República de Irlanda, cubriendo un 80% del total de las farmacias de la isla. Cuenta con 3 almacenes localizados en Dublín, Galway y Cork (cada uno de ellos distribuyendo a áreas de los alrededores).

En estos momentos, PharmaDis realiza el análisis de ventas mediante programas codificados en el ERP que extraen la información que los directores y ejecutivos necesitan. Estos programas se codifican bajo demanda y suelen tardar entre 1 y 3 días dependiendo de la dificultad. Para los directores, este es un retraso en la obtención de la información que no llegan a entender. Además, los datos se presentan en un formato Excel que no es de su agrado, ya que necesitan visualizar los datos en gráficas que el programa no es capaz de generar, por lo que necesitan manipular los datos de forma manual.

Por otro lado, los usuarios del equipo de Ventas (que pueden lanzar ellos mismos los programas ya codificados) se encuentran con la problemática de comparar las ventas con los presupuestos asignados. Estos presupuestos se guardan en una hoja de cálculo y no en el ERP. Para poder realizar el análisis, necesitan realizar tareas de manipulación de datos y suelen tardar alrededor de 2 días al mes en obtener la información que desean. Esto es un tiempo que el departamento financiero necesita reducir para dedicarlo a otras tareas (marketing, contactar con los fabricantes, etc.).

Dada la problemática de obtener información en PharmaDis, la dirección se ha propuesto solucionar este problema mediante el diseño e implementación de un almacén de datos que les permita analizar sus ventas y realizar una comparativa con los presupuestos aprobados. Para ello, han invertido en un conjunto de herramientas Business Intelligence consolidadas en el mercado actual como es **Microsoft SQL Server 2012** y su suite de productos Business Intelligence, considerando dicha inversión como estratégica para la empresa, esperando un retorno en inversión de unos 2 años.

Objetivos Generales

El objetivo principal del proyecto es la adquirir experiencia en el diseño, construcción y explotación de un almacén de datos a partir de la información disponible en una base de datos transaccional. Para ello, como se ha indicado en el punto anterior, se ha propuesto la realización de un almacén para el análisis de

¹ Nombre de empresa ficticio

un sistema de ventas de una distribuidora farmacéutica que permitirá a los directivos tomar decisiones de negocio.

Para alcanzar estos objetivos, es necesario poner en práctica los conocimientos adquiridos a lo largo de los estudios de Ingeniería Informática, en concreto las siguientes asignaturas cursadas: *Bases de Datos 2*, *Sistemas de Gestión de Bases de Datos* y *Minería de Datos*. Para la gestión del proyecto, se procederá a poner en práctica los conocimientos adquiridos en la asignatura *Metodología y Gestión de Proyectos Informáticos*.

A nivel personal, el objetivo principal de este proyecto es el del aprendizaje de la suite de Business Intelligence **Microsoft SQL Server 2012**, herramienta que cuenta con una gran demanda de profesionales en el mercado laboral actual.

Objetivos Específicos

Los objetivos específicos de este proyecto son los siguientes:

- Permitir a PharmaDis analizar la siguiente información:
 - Ventas en Euro
 - Coste en Euro
 - Presupuesto en Euro
 - Variación (Ventas – Presupuesto) y % Variación (Variación / Presupuesto)
 - Margen (Ventas – Coste) y % Margen (Margen / Coste)
 - Número de órdenes de pedido
 - Número de líneas de pedido recibidas, despachadas y no despachadas (fuera de stock)
 - Número de unidades pedidas, despachadas y no despachadas (fuera de stock)
- La creación de una base de datos en SQL Server 2012 para almacenar los datos proporcionados.
- Desarrollo de los paquetes de carga de datos con **Integration Services**, que lean los ficheros de datos y los cargue en la base de datos.
- Creación de un cubo de datos OLAP con **Analysis Services** para la generación de informes.
- Proporcionar el siguiente conjunto predefinido de informes mediante **Reporting Services**:
 - Informe 01. Ventas vs Presupuesto por Almacén y Mes Venta
 - Informe 02. Ventas vs Presupuesto por Categoría y Mes Venta
 - Informe 03. Ventas Actuales vs Ventas Actuales - 1 por Almacén y Categoría
 - Informe 04. 20 productos con más/menos demanda en Ventas por Categoría
 - Informe 05. 20 clientes más/menos demandantes en Ventas 2012 por Categoría
 - Informe 06. Informe Operacional por Año de Venta para el año 2012
 - Informe 07. 20 Rutas con Más/Menos Beneficios Año 2012
 - Informe 08. Ventas y Margen por Región Año 2012
 - Informe 09. Causas Crédito en Ventas 2012
 - Informe 10. Ventas y Margen por Distribuidor y Fabricante

Metodología de Trabajo

La metodología de trabajo que se pretende llevar a cabo en este proyecto es la de **desarrollo en cascada**, ya que se trata de una de las metodologías más utilizadas en el mercado de construcción de software, que define que las siguientes actividades se realicen rigurosamente en secuencia: *Análisis de Requisitos, Diseño, Construcción, Pruebas e Implantación*.

Planificación del Proyecto

Las tareas a realizar en cada una de las etapas son las siguientes:

- Análisis Preliminar de Requisitos:
 - Analizar los datos de los ficheros.
 - Definición del hecho, granularidad, dimensiones y atributos.
 - Definición de las medidas.
- Análisis y Diseño:
 - Lectura de documentación sobre modelos dimensionales.
 - Análisis de Requisitos.
 - Diseño del modelo dimensional.
 - Diseño del proceso de carga, cubo e informes.
- Implementación:
 - Instalación del entorno de desarrollo y familiarización con el entorno.
 - Construcción de la base de datos, paquetes de carga de datos, cubo e informes.
 - Análisis de la información obtenida.
- Pruebas:
 - Pruebas unitarias para asegurar que todos los datos han sido cargados correctamente.
 - Pruebas unitarias para asegurar que los informes muestran la información correctamente.
- Implantación:
 - Implantación en una máquina de producción (entregable).
 - Creación del documento de la memoria y del video de presentación.

A continuación, en la *Tabla 1. Hitos* se especifican los hitos que se han definido en la planificación de este proyecto:

Hitos	Fecha
Inicio del Proyecto	27/02/2013
Plan de Trabajo	05/03/2012
PEC 1: Plan de Trabajo y Análisis de Requisitos	12/03/2013
PEC 2: Diseño del Modelo y Proceso de Carga	16/04/2013
PEC 3: Implementación y Pruebas Unitarias	29/05/2013
Entrega Final: Memoria e Implantación en Producción	17/06/2013
Debate Virtual	Finales Junio 2013

Tabla 1. Hitos

Tal y como se puede ver en la *Ilustración 1. Diagrama de Gantt*, la planificación total del proyecto requiere trabajar durante un período de aproximadamente 105 días (15 semanas) con una dedicación total de 48 días, comenzando el 27 de Febrero de 2013 y finalizando el 17 de Junio de 2013, considerando 1 día como 8 horas trabajadas. El **total de horas** a dedicar al proyecto suman **384 horas**.

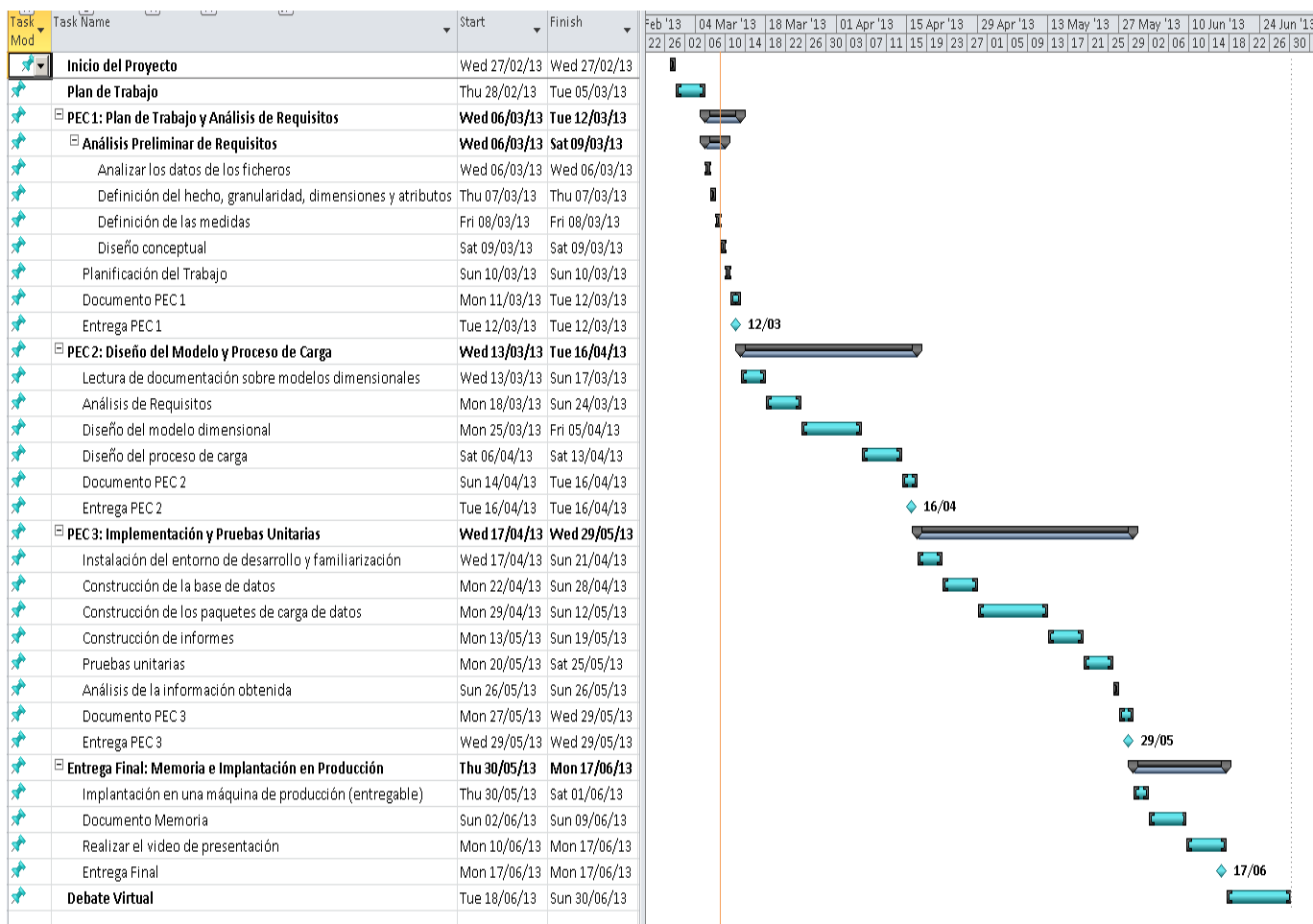


Ilustración 1. Diagrama de Gantt

Análisis de Riesgos

A continuación se identifican los riesgos que es posible que sucedan a lo largo del ciclo de desarrollo del proyecto, especificando en qué consisten y definiendo el plan de contingencia adecuado.

Equipo Informático

Es posible que, debido a circunstancias ajenas al proyecto, el equipo de trabajo se estropee, sufra algún deterioro, o incluso se extravíe o sea robado. Para evitar que el trabajo realizado hasta ese punto se pierda, se realizarán las siguientes tareas:

- Se realizará una copia de seguridad del proyecto con una frecuencia diaria, incluyendo código y documentación. Esta copia diaria será comprimida en formato ZIP, nombrada con la fecha en la que se ha realizado y guardada en los siguientes sistemas de almacenamiento:
 - o Un email en una cuenta personal (hasta 25MB de capacidad)
 - o Dropbox (hasta 17GB de capacidad), con copia a múltiples PC.

- Se realizará una copia de seguridad de la máquina virtual con el software base instalado sin el proyecto creado, para así poder reiniciar el trabajo con la copia de seguridad más reciente. Esta copia de seguridad se almacenará en un disco duro externo propiedad del autor del proyecto

Horario de Trabajo y Viajes Laborales No Planificados

El horario de trabajo estándar es de 9:00 a 18:00 de lunes a viernes. Es posible que, debido a exigencias de proyecto, este horario de trabajo se extienda ciertas horas debido a un incremento en la carga de trabajo o por viajes laborales, lo que implica que no sea posible cumplir el calendario de trabajo establecido. Como plan de contingencia, se establece que en caso de tener que realizar estas horas extras:

- Se compense con días libres en la empresa
- En último caso se intentará realizar tareas fuera del calendario de trabajo establecido.

Problemas de Salud

En caso de enfermedad leve, se prevé recuperar el tiempo no trabajado en el proyecto mediante la realización de horas extras o incluso solicitando días de vacaciones. En caso de enfermedad grave, se evaluará con el consultor de la asignatura si es posible continuar con el desarrollo del proyecto.

Entregables

Los entregables que se obtienen del desarrollo de este proyecto son los siguientes:

- Memoria del Proyecto: documento en el que se incluirán todas las tareas realizadas y la planificación establecida, y que se entregará al final del proyecto.
- Producto Instalado: una máquina virtual VirtualBox con el producto instalado y funcionando.
- Código Fuente: scripts de creación de la base de datos, la solución con los paquetes de carga de datos y ficheros de configuración, la solución del cubo y la solución con la definición de los informes.
- Presentación Virtual: vídeo de 20 minutos donde se explica todo el trabajo realizado, incluyendo una demostración de la funcionalidad de la solución.

Requisitos de Hardware / Software

El proyecto a realizar requiere de los siguientes elementos hardware y software.

Hardware

- Punto de trabajo estándar UOC.

Software

- **Oracle Virtual Box 4.1.6** para la instalación de una máquina virtual sobre la que trabajar para evitar software no deseado en la máquina host. La máquina virtual tendrá instalada un **sistema operativo Windows Server 2008** en inglés.
- Servidor de base de datos **Microsoft SQL Server 2012 SP1 Business Intelligence Edition**, en inglés, con las siguientes extensiones instaladas y configuradas:

- **Integration Services** para el desarrollo de la carga de datos.
- **Analysis Services** para la creación del cubo OLAP.
- **Reporting Services** para la creación de los informes y analíticas.
- **Power Designer 16.0** para la creación del modelo dimensional.
- **Microsoft Office 2010** para la generación de tablas, hojas de cálculo, documentación y realización de pruebas unitarias. Los ficheros CSV se pueden consultar y manipular utilizando MS Excel.
- **Microsoft Project 2010** para la creación del diagrama de Gantt.

Breve Descripción del Resto de Apartados

En el siguiente apartado, se realizará una descripción de la fase de análisis, donde se realiza un estudio de los ficheros dados y se define el enfoque de diseño utilizado, así como conceptos de almacenes de datos necesarios para entender el diseño de la solución. Se definen también los hechos y las dimensiones con sus atributos, la granularidad y las medidas, terminando con un diseño conceptual de la base de datos.

A continuación, se presentará el diseño completo de la solución, comenzando con una introducción al software Microsoft SQL Server, seguido de una breve explicación de la arquitectura hardware y software utilizadas, y terminando con una exposición detallada de las cuatro partes en que se divide la solución: diseño de base de datos, diseño del proceso de carga, diseño del cubo y diseño de los informes.

Por último, se concluye con una sección de conclusiones, así como el glosario y la bibliografía utilizada.

2. Análisis

Estudio de los Ficheros de Datos

La empresa PharmaDis nos ha encargado la creación de un almacén de datos para analizar información de sus ventas y de su presupuesto, y automatizar así la generación de informes mediante el uso de herramientas Business Intelligence. Para ello, nos ha provisto con una serie de ficheros en formato CSV² que necesitamos analizar para entender los datos que nos proporcionan.

El resultado de un primer análisis se presenta a continuación para cada uno de los ficheros:

- cliente.csv: datos acerca de los clientes de la distribuidora.
 - *account_number*: código de cuenta.
 - *account_name*: nombre de cuenta / farmacia.
 - *account_abb_name*: nombre abreviado de cuenta / farmacia.
 - *account_address*: dirección de la farmacia.
 - *account_phone_number*: número de teléfono de la farmacia.
 - *account_fax_number*: número de fax de la farmacia.
 - *account_contact_email*: email de contacto de la farmacia.
 - *account_contact_name*: nombre de contacto de la farmacia.
 - *account_ledger_code*: tipo libro mayor: (S)ales (Venta) o (R)echarge (Recargo).
 - *account_payment_terms*: términos de pago.

² Comma-separated values, formato de texto plano para representar datos en forma de tabla.

- *account_category*: categoría de la cuenta.
- *account_group*: grupo de la cuenta según la categoría.
- *account_location*: localización física de la farmacia.

Tras realizar un análisis sobre este fichero, se obtuvieron los siguientes resultados:

- Existe una jerarquía de datos: *Categoría* -> *Grupo* -> *Localización* -> *Cuenta*.

- *producto.csv*: datos acerca de los productos vendidos.

- *product_code*: código del producto.
- *product_desc*: descripción del producto.
- *product_pack_size*: formato del producto.
- *product_cost_price*: precio de coste unitario del producto.
- *product_trade_price*: precio de venta unitario del producto.
- *product_ean_code*: código EAN³ del producto.
- *product_classification_code*: código de categoría de producto.
- *product_classification_desc*: descripción de categoría de producto.

Tras realizar un análisis sobre este fichero, se obtuvieron los siguientes resultados:

- *product_pack_size* tiene diferentes formatos (pack, mililitros, gramos, etc.).
- *product_ean_code* parece tener ceros cuando no existe un código EAN.

- *fabricante.csv*: datos acerca del fabricante de los productos y de la distribuidora.

- *supplier_code*: código del fabricante.
- *supplier_name*: nombre del fabricante.
- *pre_wsaler_name*: nombre de la empresa que vende el producto a PharmaDis.

- *categoria_de_producto.csv*: datos acerca de la categoría del producto.

- *category_code*: código de la categoría de producto.
- *category_desc*: nombre de la categoría del producto.

- *region.csv*: información de los condados de la República de Irlanda.

- *region_code*: código de región / condado.
- *region_desc*: nombre de región / condado.

- *ruta_distribucion.csv*: datos de la ruta que se ha utilizado para distribuir los productos.

- *route_code*: código de ruta.
- *route_number*: número de ruta.
- *route_name*: nombre de ruta
- *route_delivery_day*: día de reparto en número.
- *route_van_number*: número de furgoneta.

³ European Article Number, generalmente se conoce como códigos de barra.

- *route_departure_time*: hora de salida de la furgoneta.

Tras realizar un análisis sobre este fichero, se obtuvieron los siguientes resultados:

- Cada ruta es la combinación de un número de ruta y día de la semana.
 - El día de la semana es 1 para Lunes, 2 para Martes, etc.
 - *route_departure_time* es la hora de salida en segundos desde las 00:00.
- *tipo_de_orden.csv*: datos acerca del tipo de orden realizada (“Facturas” o “Créditos”).
 - *transaction_code*: código de transacción.
 - *transaction_desc*: nombre de transacción.
 - *transaction_category*: nombre de categoría de transacción.
 - *causa_creditos.csv*: información de las causas por las que se crea un crédito.
 - *credit_reason_code*: código de razón de crédito.
 - *credit_reason_desc*: nombre de razón de crédito.
 - *líneas_pedido_XXXX_YYYY.csv*: órdenes y líneas de pedido del año 2012.
 - *delivery_date*: fecha de reparto de la orden.
 - *sale_date*: fecha de venta de la orden.
 - *order_number*: número de orden.
 - *invoice_number*: número de factura.
 - *line_number*: número de línea de pedido.
 - *account_number*: número de cuenta de la farmacia.
 - *region_code*: región donde se ha producido la venta.
 - *transaction_code*: código de transacción.
 - *route_code*: código de ruta de reparto.
 - *line_status_code*: estado de la línea de pedido.
 - *credit_reason_code*: código de razón de crédito.
 - *product_code*: código de producto de la línea de pedido.
 - *supplier_code*: código de fabricante del producto.
 - *sales_line_value_eur*: montante de la línea de producto.
 - *sales_line_ordered_qty*: cantidad de producto ordenadas por el cliente.
 - *sales_line_dispatched_qty*: cantidad de producto despachada.

Tras realizar un análisis sobre este fichero, se obtuvieron los siguientes resultados:

- 3 ficheros identificados por código de almacén (*DUBL*, *CORK* y *GALW*) y año (2012).
 - *line_status_code* indica si la línea ha sido despachada (0) o si el producto no ha sido despachado por estar fuera de stock (10).
 - *sales_line_value_eur*, *sales_line_ordered_qty* y *sales_line_dispatched_qty* aparecen con valores negativos cuando se tratan de créditos.
- *presupuesto.csv*: datos acerca del presupuesto estimado para el año 2012.

- *date*: fecha de venta.
- *depot*: cogido de almacén.
- *category_code*: código de categoría de producto.
- *budget_amount_eur*: valor de presupuesto en Euros.

Tras realizar un análisis sobre este fichero, se obtuvieron los siguientes resultados:

- Resultado de la combinación de 365 días x 3 almacenes x 4 categorías.
- El código de los almacenes se indica por DUBL, CORK y GALW.

El resultado de un segundo análisis más detallado se presenta a continuación para cada uno de los ficheros. Se puede ver que en todos los campos con tipo de dato alfanumérico se han ampliado los caracteres para evitar problemas a la hora de cargar datos en el futuro (por ejemplo, que una de las descripciones sea más larga de la actual). La razón principal por la que se aplica esta configuración es porque desconocemos los tipos de datos origen, ya que han sido proporcionados en ficheros CSV.

cliente.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
account_number	5	7	NO	SI	Alfanumérico 10 caract.
account_name	3	40	NO	NO	Alfanumérico 50 caract.
account_abb_name	1	42	NO	NO	Alfanumérico 50 caract.
account_address	1	121	SI	NO	Alfanumérico 150 caract.
account_phone_number	1	18	SI	NO	Alfanumérico 30 caract.
account_fax_number	1	16	SI	NO	Alfanumérico 20 caract.
account_contact_email	1	45	SI	NO	Alfanumérico 50 caract.
account_contact_name	1	41	SI	NO	Alfanumérico 50 caract.
account_ledger_code	1	1	NO	NO	Alfanumérico 1 caract.
account_payment_terms	1	3	NO	NO	Alfanumérico 3 caract.
account_category	4	23	SI	NO	Alfanumérico 100 caract.
account_group	3	52	SI	NO	Alfanumérico 100 caract.
account_location	3	66	SI	NO	Alfanumérico 100 caract.

Tabla 2. Fichero de Clientes

producto.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
product_code	4	6	NO	SI	Entero
product_desc	7	35	NO	NO	Alfanumérico 50 caract.
product_pack_size	1	6	SI	NO	Alfanumérico 10 caract.
product_cost_prize	1	7	NO	NO	Moneda
product_trade_price	1	7	NO	NO	Moneda
product_ean_code	1	15	SI	NO	Alfanumérico 20 caract.
product_classification_code	3	3	NO	NO	Alfanumérico 5 caract.
product_classification_desc	6	9	NO	NO	Alfanumérico 20 caract.

Tabla 3. Fichero de Productos

fabricante.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
supplier_code	4	6	NO	SI	Alfanumérico 10 caract.
supplier_name	4	5	NO	NO	Alfanumérico 50 caract.
pre_wsaler_name	7	35	SI	NO	Alfanumérico 50 caract.

Tabla 4. Fichero de Fabricantes

categoria_de_producto.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
category_code	3	3	NO	SI	Alfanumérico 5 caract.
category_desc	6	9	NO	NO	Alfanumérico 20 caract.

Tabla 5. Fichero de Categorías de Producto

region.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
region_code	1	2	NO	SI	Entero
region_desc	4	20	NO	NO	Alfanumérico 30 caract.

Tabla 6. Fichero de Regiones

ruta_distribucion.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
route_code	3	4	NO	SI	Entero
route_number	3	5	NO	NO	Entero
route_name	4	36	NO	NO	Alfanumérico 20 caract.
route_delivery_day	1	1	NO	NO	Entero
route_van_number	1	3	NO	NO	Entero
route_departure_time	5	5	NO	NO	Entero

Tabla 7. Fichero de Rutas de Distribución

tipo_de_orden.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
transaction_code	4	8	NO	SI	Alfanumérico 10 caract.
transaction_desc	12	30	NO	NO	Alfanumérico 50 caract.
transaction_category	6	7	NO	NO	Alfanumérico 20 caract.

Tabla 8. Fichero de Tipos de Orden

causa_creditos.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
credit_reason_code	1	3	NO	SI	Alfanumérico 10 caract.
credit_reason_desc	7	26	NO	NO	Alfanumérico 30 caract.

Tabla 9. Fichero de Causa de Créditos

linea_pedido_XXXX_YYYY.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
delivery_date	10	10	NO	NO	Fecha
sale_date	10	10	NO	NO	Fecha
order_number	8	9	NO	SI	Entero
invoice_number	7	7	NO	NO	Entero
line_number	1	3	NO	SI	Entero
account_number	5	7	NO	NO	Alfanumérico 10 caract.
region_code	1	2	SI	NO	Entero
transaction_code	4	8	NO	NO	Alfanumérico 10 caract.
route_code	3	4	SI	NO	Entero
line_status_code	1	2	NO	NO	Entero
credit_reason_code	1	3	SI	NO	Alfanumérico 10 caract.
product_code	4	6	SI	NO	Entero
supplier_code	4	6	NO	NO	Alfanumérico 10 caract.
sales_line_value_eur	1	8	NO	NO	Moneda
sales_line_ordered_qty	3	4	NO	NO	Entero
sales_line_dispatched_qty	3	4	NO	NO	Entero

Tabla 10. Fichero de Líneas de Pedido Año 2012

presupuesto.csv					
Campo	No. Min. Caracteres	No. Max. Caracteres	Contiene Blancos/Nulos	Clave Primaria	Tipo Dato Candidato
date	10	10	NO	NO	Fecha
depot	4	4	NO	NO	Alfanumérico 30 caract.
category_code	3	3	NO	NO	Alfanumérico 5 caract.
budget_amount_eur	1	8	NO	NO	Moneda

Tabla 11. Fichero de Presupuesto

Vemos que existen casos en los que existen falta de datos, por lo que han de ser tratados a la hora de realizar el proceso de carga. Por otro lado, existen campos en los que parece que un tipo de dato puede encajar pero al final no sucede así, como por ejemplo el número de teléfono (caracteres alfanuméricos).

En relación a las líneas de pedido, hemos identificado que aunque la clave primaria consista en número de orden y número de línea, existe la posibilidad de que en un futuro el número de orden sea igual en alguno de los almacenes, lo que equivaldría a un gran problema a la hora de analizar los datos.

También es importante destacar que el presupuesto parece no tener una clave primaria. Podríamos establecer que la clave primaria es la combinación de almacén x categoría x día de venta, pero he decidido no hacerlo de esta manera para evitar posibles problemas a la hora de cargar datos.

Datos Históricos

La dirección ha acordado que no es necesario mantener el histórico de cambios en datos, es decir, no es necesario mantener por ejemplo, la categoría que un cliente tenía en el momento que la línea de pedido se había creado (aplica para todos los ficheros dados), y que desean ver siempre los datos que el sistema operacional presenta.

Casos de Uso

Para identificar las interacciones entre los usuarios del sistema a implementar y sus funcionalidades, se ha creado el siguiente caso de uso, en el que se pueden ver los roles de los usuarios, por un lado el operador del data Warehouse, y por otro lado, los usuarios que consumen informes, y las actividades que tienen asociadas cada uno de ellos .

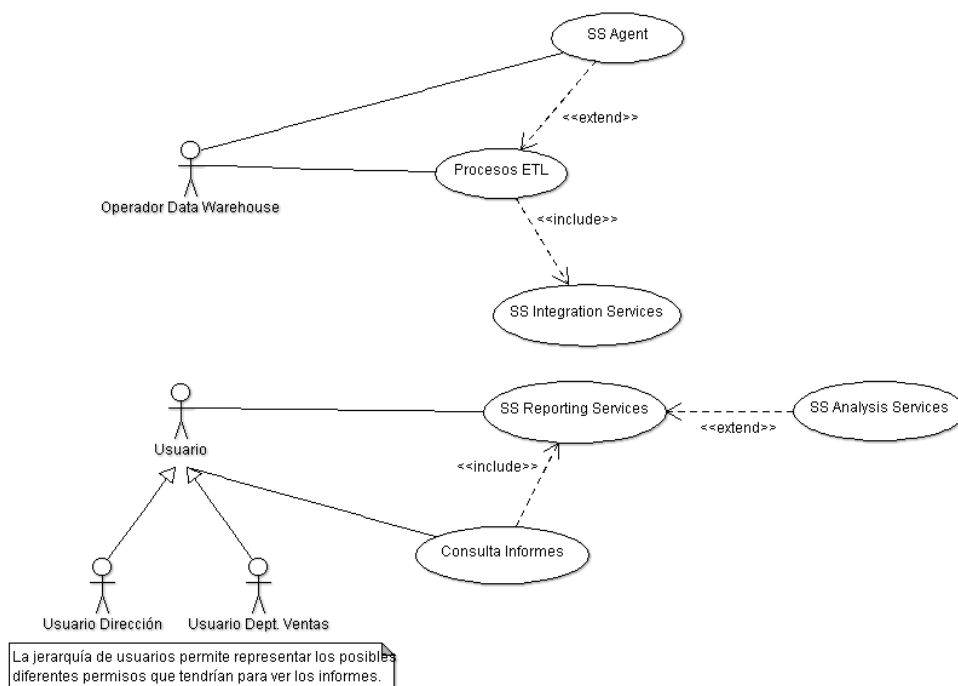


Ilustración 2. Casos de Uso

Enfoque de Diseño

El diseño de la solución construida se basa en el enfoque propuesto por **Ralph Kimball**. A grandes rasgos, este enfoque consiste en la definición de una serie de entidades denominadas **dimensiones** que almacenarán datos cualitativos (descripciones, códigos, categorías, tiempo, etc.) y una serie de entidades centrales llamadas **hechos**, en las que se almacenará información cuantitativa (montantes, cantidades).

La tabla de hechos es el resultado de una relación N-aria con las dimensiones, es decir, estas se relacionan con la tabla de hechos mediante relaciones 1 a N. A este tipo de estructuras se les llama **modelo en estrella** debido a que la forma gráfica de la estructura semeja a una estrella, como se puede ver en la *Ilustración 3. Modelo en Estrella*.

Generalmente, las dimensiones se caracterizan por tener tamaños de registros extensos (muchas columnas, tipos de datos alfanuméricos largos) pero con un número de registros medianamente bajo, mientras que las tablas de hechos se caracterizan por tener un número registros muy alto pero tamaños de registros reducidos (generalmente, solamente contiene tipos de datos numéricos). Como norma general, la información de las dimensiones se agrupará (mediante cláusulas GROUP BY) y la información de los hechos se agregará (mediante funciones de agregación como SUM, AVG o COUNT).

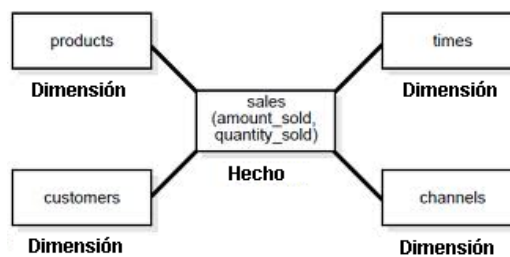


Ilustración 3. Modelo en Estrella

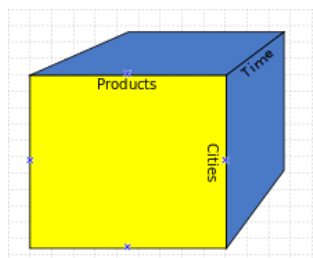


Ilustración 4. Cubo OLAP

Además de la tabla de hechos y de las dimensiones como parte del proyecto, es necesario construir un **cubo OLAP** (*On Line Analytical Processing*). Un cubo no es más que una base de datos multidimensional en el que los datos se almacenan en vectores multidimensionales, y que se caracteriza por tener al menos 2 o más dimensiones. En la *Ilustración 4. Cubo OLAP* se puede ver cómo se suelen representar los cubos OLAP gráficamente.

El resultado de la combinación de cada uno de los valores de todas las dimensiones (producto cartesiano) se denomina celda, y es donde se almacena el valor del hecho para dicha combinación de valores. Por ejemplo, la combinación de fecha de venta “25 Mayo 2012”, producto “REMICADE 100MG” y cliente “BOOTS CITYWEST” resulta en una celda donde se almacena, por ejemplo, el valor de ventas 100€ para la combinación de dichos valores.

Una de las razones principales para usar un cubo OLAP es la rapidez en ejecutar consultas, que permitirá a los usuarios recuperar y analizar datos más rápidamente.

Conceptos de Diseño Dimensional

Antes de iniciar con la definición del modelo, es necesario tener claros una serie de conceptos de diseño dimensional [1] que nos ayuden a entender mejor la estructura de estos sistemas.

- Claves subrogadas: se entiende por clave subrogada al identificador único de una entidad que no se deriva de los datos de la aplicación y que generalmente no es visible al usuario. Las claves subrogadas se utilizan en modelos dimensionales como claves primarias de las dimensiones con el fin de diferenciar la clave de negocio de la entidad (clave primaria de la entidad origen) de la clave primaria de la dimensión. Suelen construirse a partir de una secuencia numérica autogenerada en el que no existe una relación entre el significado del registro y dicha clave, aunque hay casos como en la dimensión Fecha en que puede tener un significado (por ejemplo, 20120101 representaría el registro con fecha 1 Enero 2012). Otro de los beneficios de usar claves subrogadas es el rendimiento en operaciones de consulta, ya que estas se hacen mediante valores numéricos, y el almacenamiento en las tablas de hecho, ya que se evita el usar códigos alfanuméricos que suelen ocupar más espacio que tipos de datos enteros.
- Roles de dimensiones: existen casos en que ciertas dimensiones pueden tener más de una relación con la misma tabla de hechos. Por ejemplo, la dimensión Fecha puede tener dos relaciones con una tabla de hechos de ventas, como puede ser Fecha de Venta y Fecha de Factura. En este ejemplo, la dimensión toma dos diferentes roles aunque la entidad física siga siendo una sola.

- Tipos de dimensiones: a la hora de modelar una dimensión, debemos decidir si queremos gestionar la historia de la entidad, o bien mantener siempre el último estado. A este mecanismo de gestionar la información se le denomina *Slowly Changing Dimension*, y según Kimball [1], existen diferentes tipos (mencionamos solamente los más importantes):
 - o *Tipo 0*: el valor del registro o campo en la dimensión no cambia y mantiene los valores que tenía cuando se cargó por primera vez.
 - o *Tipo I*: el valor del registro o campo se actualiza con la última versión de los datos del operacional. Este tipo no guarda ningún valor histórico y cualquier consulta sobre datos históricos mostrará los valores que el sistema operacional muestra tras la última carga. En cambio, este tipo es más fácil de mantener que el siguiente.
 - o *Tipo II*: si uno de los campos cambia su valor, se genera un nuevo registro con el cambio. Para diferenciar versiones (existirán múltiples registros con la misma clave natural), se suele implementar un campo *flag* que indica el número de versión o bien dos campos fecha indicando la fecha de comienzo y fin de la vida de dicho registro. Este método permite registrar histórico de cambios aunque requiere un mantenimiento de las tablas de hechos si los datos cambian constantemente. Un ejemplo puede verse a continuación en la tabla *Tabla 12. Ejemplo de Dimensión Tipo II*:

Producto_ID	Producto_Codigo	Producto_Nombre	Producto_Tipo	Fecha_Inicio	Fecha_Fin
123	PROD1	NUROFEN	CAPSULAS	01/01/1900	21/12/2004
124	PROD1	NUROFEN	SOBRES	22/12/2004	31/12/9999

Tabla 12. Ejemplo de Dimensión Tipo II

Existen otros tipos como *Tipo III* (2 columnas para almacenar el valor actual y el histórico respectivamente) y el *Tipo VI* (combinación de *Tipos I + II + II = VI*) que no veremos por ser técnicas de modelado más avanzadas y que no van a ser necesarias para este proyecto.

Por otro lado, existe otro tipo de clasificación de dimensiones según su finalidad:

- o *Conformada*: una dimensión conformada tiene relaciones con diferentes tablas de hechos cuando el significado de dicha entidad es el mismo para ambos hechos. Esto permite realizar consultas y analizar hechos de múltiples tablas de forma conjunta.
- o *No conformada*: al contrario que las dimensiones conformadas, estas mantienen relaciones con una única tabla de hechos.
- o *Degenerada*: suelen ser claves que no tienen atributos asociados, como número de factura o número de orden. Suelen almacenarse en las tablas de hechos, ya que generalmente son parte de la clave primaria del hecho.
- o *Junk (basura)*: una dimensión *junk* es aquella que generalmente almacena aquellos datos que no se corresponden con una dimensión pero que son necesarios para el análisis. Suelen ser agrupaciones de flags, indicadores o comentarios sin ninguna relación entre ellos que permiten evitar la creación de un gran número de dimensiones en el hecho y así tenerlos disponibles para el análisis.

- Tipos de tablas de hechos: en diseño dimensional, existen 3 tablas de hechos
 - *Transaccional*: es la más básica y fundamental, en el que el grano de la tabla de hechos se especifica como un registro por transacción, almacenando el nivel de detalle más bajo. Por ejemplo, una línea de pedido.
 - *Instantánea Periódica (Periodic Snapshot)*: esta tabla de hechos toma una fotografía o instantánea de un momento definido en un período de tiempo. Suele depender de una tabla de hechos transaccional. Por ejemplo, el balance de las cuentas bancarias a final de mes.
 - *Instantánea Acumulativa (Accumulating Snapshot)*: este tipo de hechos muestra la actividad de un proceso que tiene un principio y un fin, como por ejemplo una orden de compra. Este tipo suele tener múltiples fechas que marcan los diferentes hitos de la orden (fecha de creación, fecha de envío, fecha de llegada).

Existe un cuarto tipo de tabla de hechos denominada *tabla de hechos sin hecho (factless fact table)*, en las que la tabla no tiene ningún hecho definido. Suelen usarse para modelar relaciones muchos a muchos. Por ejemplo, para modelar qué permisos tienen ciertos usuarios en una aplicación.

- Tipos de medidas: por último, especificamos los diferentes tipos de medidas que existen:
 - *Aditivas*: las medidas aditivas permiten ser agregadas por cualquiera de las dimensiones. Por ejemplo, la suma de las ventas.
 - *Semi-Aditivas*: son medidas que permiten ser agregadas por algunas de las dimensiones, pero no para otras. Un ejemplo de medida semi-aditiva es el balance total de las cuentas bancarias en una tabla de hechos periódica, ya que permite agregar los balances por cuenta pero no por fecha.
 - *No Aditivas*: son medidas que no pueden ser agregadas por ninguna dimensión. Por ejemplo, un porcentaje (si bien es importante destacar que es una mala práctica almacenar porcentajes y medias como hechos).

Para la consecución de este proyecto ha sido muy importante entender estas definiciones para poder identificar y modelar las entidades de nuestro modelo dimensional.

El Hecho

Uno de los primeros pasos en la realización del diseño dimensional es identificar el hecho o hechos del modelo que vamos a construir, y que se traducirán en las tablas centrales que hemos mencionado anteriormente. En este proyecto, se han identificado los siguientes hechos:

- Línea de pedido: este hecho nos permite analizar las ventas, cantidades y líneas nos han solicitado. El fichero de líneas de pedido será el que se utilizará para cargar este hecho.
- Almacén x Categoría x Día de venta: este hecho nos permite analizar el presupuesto por almacén, categoría y día de venta. El fichero de presupuesto será el que se utilizará para cargar este hecho.

La Granularidad

La granularidad de las tablas de hecho identificadas es la siguiente:

- Línea de pedido: tendrá como granularidad una línea de pedido individual (cada fila representará una línea de pedido)
- Almacén x Categoría x Día de venta: tendrá la granularidad definida por el hecho donde cada entrada en la tabla de hechos será el presupuesto asignado a un almacén para una categoría en un día de venta concreto.

La granularidad de los ficheros de líneas y presupuesto es ya la adecuada, por lo que no requiere ningún tipo de agregación (cada fila del fichero se encuentra en el nivel atómico de la tabla de hechos).

Las Dimensiones. Atributos

Las dimensiones que se han identificado son las siguientes:

- Fecha de venta
- Fecha de reparto
- Almacén
- Categoría de Producto
- Causa de Crédito
- Cliente
- Fabricante
- Producto
- Región
- Ruta de distribución
- Tipo de Orden

Como requisito comentado anteriormente, se requiere que éstas almacenen la última instantánea del operacional. Esto nos lleva a la conclusión de que todas las dimensiones de este modelo deben de ser modeladas utilizando la técnica *Tipo I* (última instantánea del sistema operacional).

Los atributos de las dimensiones especificadas son los siguientes:

- Fecha de venta y Fecha de reparto: cada una de estas tendrá la siguiente estructura jerárquica:
 - Nivel 1: Año
 - Nivel 2: Cuarto (YYYY/QX)
 - Nivel 3: Mes (YYYY/MM)
 - Nivel 4: Fecha (YYYY-MM-DD)
- Almacén:
 - Código de almacén
 - Nombre de almacén
- Categoría de Producto:
 - Código de la categoría de producto.
 - Nombre de la categoría del producto.
- Causa de Crédito
 - Código de causa de crédito.
 - Nombre de razón de crédito.

- Cliente

- Código de cuenta.
- Nombre de cuenta / farmacia.
- Nombre abreviado de cuenta / farmacia.
- Dirección de la farmacia.
- Número de teléfono de la farmacia.
- Número de fax de la farmacia.
- Email de contacto de la farmacia.
- Nombre de contacto de la farmacia.
- Tipo de libro mayor: (S)ales (Venta) o (R)echarge (Recargo).
- Términos de pago.
- Y la siguiente jerarquía:
 - Nivel 1: Categoría de la cuenta.
 - Nivel 2: Grupo de la cuenta según la categoría.
 - Nivel 3: Localización física de la farmacia.
 - Nivel 4: Código de cuenta / nombre de cuenta.

- Fabricante

- Código del fabricante.
- Nombre del fabricante.
- Nombre de la empresa que vende el producto a PharmaDis.

- Producto

- Código del producto.
- Código del fabricante.
- Descripción del producto.
- Formato del producto (pack, mililitros, gramos, etc.)
- Precio de coste unitario del producto en Euro.
- Precio de venta unitario del producto en Euro.
- Código EAN del producto.

Los siguientes atributos se crearán en la dimensión, pero no se mostrarán al usuario, ya que se utilizan para calcular la “Categoría de producto” en el hecho:

- Código de categoría de producto.
- Descripción de categoría de producto.

- Región:

- Código de región / condado.
- Nombre de región / condado.

- Ruta de distribución

- Código de ruta.
- Número de ruta.
- Nombre de ruta
- Día de reparto en texto (Lunes, Martes, etc.)
- Número de furgoneta.
- Hora de salida de la furgoneta (en formato *hh:mm*)

- Tipo de Orden
 - Código de transacción.
 - Nombre de transacción.
 - Nombre de categoría de transacción.

Identificación de las Medidas

Una vez identificados los hechos y las granularidades, podemos pues definir las medidas que vamos a tener disponibles en las tablas de hechos definidas:

- € Ventas
- € Coste
- # Órdenes de pedido
- # Líneas de pedido recibidas
- # Líneas de pedido despachadas
- # Líneas de pedido no despachadas
- # Unidades pedidas
- # Unidades despachadas al cliente
- # Unidades no despachadas (producto fuera de stock)
- € Presupuesto

Es importante destacar que *Variación*, *% Variación*, *Margen* y *Margen %* no se considerarán hechos sino fórmulas que serán calculadas con la herramienta de Business Intelligence, ya que no es posible definir un hecho para esta información (por ejemplo, la variación puede ser entre ventas de diferentes meses, o entre ventas y presupuesto de un mismo mes).

Por otro lado se identifica la medida “*Líneas de Pedido No Despachadas*” como el cálculo “*Líneas de Pedido – Líneas de Pedido Despachadas*” y “*Unidades No Despachadas*” como el cálculo “*Unidades Pedidas – Unidades Despachadas*”. Ambos cálculos se realizarán en el entorno de la herramienta de Business Intelligence para evitar almacenar información redundante y no se almacenarán como hechos.

Por último, es importante destacar que todas las medidas definidas son aditivas (esto es, pueden ser agregadas por todas las dimensiones asociadas) excepto “# Órdenes de pedido”, que se trata de una medida semi-aditiva. Esta es un COUNT DISTINCT sobre una granularidad superior a la línea de pedido (orden), por lo que requiere un cuidado especial ya que existen dimensiones que están asociadas a la granularidad de líneas de pedido como “*Producto*”, “*Categoría de Producto*” o “*Fabricante*”. Esto significa que una misma orden podría contarse varias veces si una orden de pedido incluyo múltiples productos, productos de diferentes categorías o de diferentes fabricantes.

Diseño Conceptual

En la *Ilustración 5. Diagrama Conceptual*, podemos ver el diagrama conceptual completo con las dimensiones (fondo azul) y las tablas de hechos (fondo verde), cada una de estas con sus atributos, que requiere clarificar una serie de decisiones realizadas para asegurar un correcto funcionamiento del sistema:

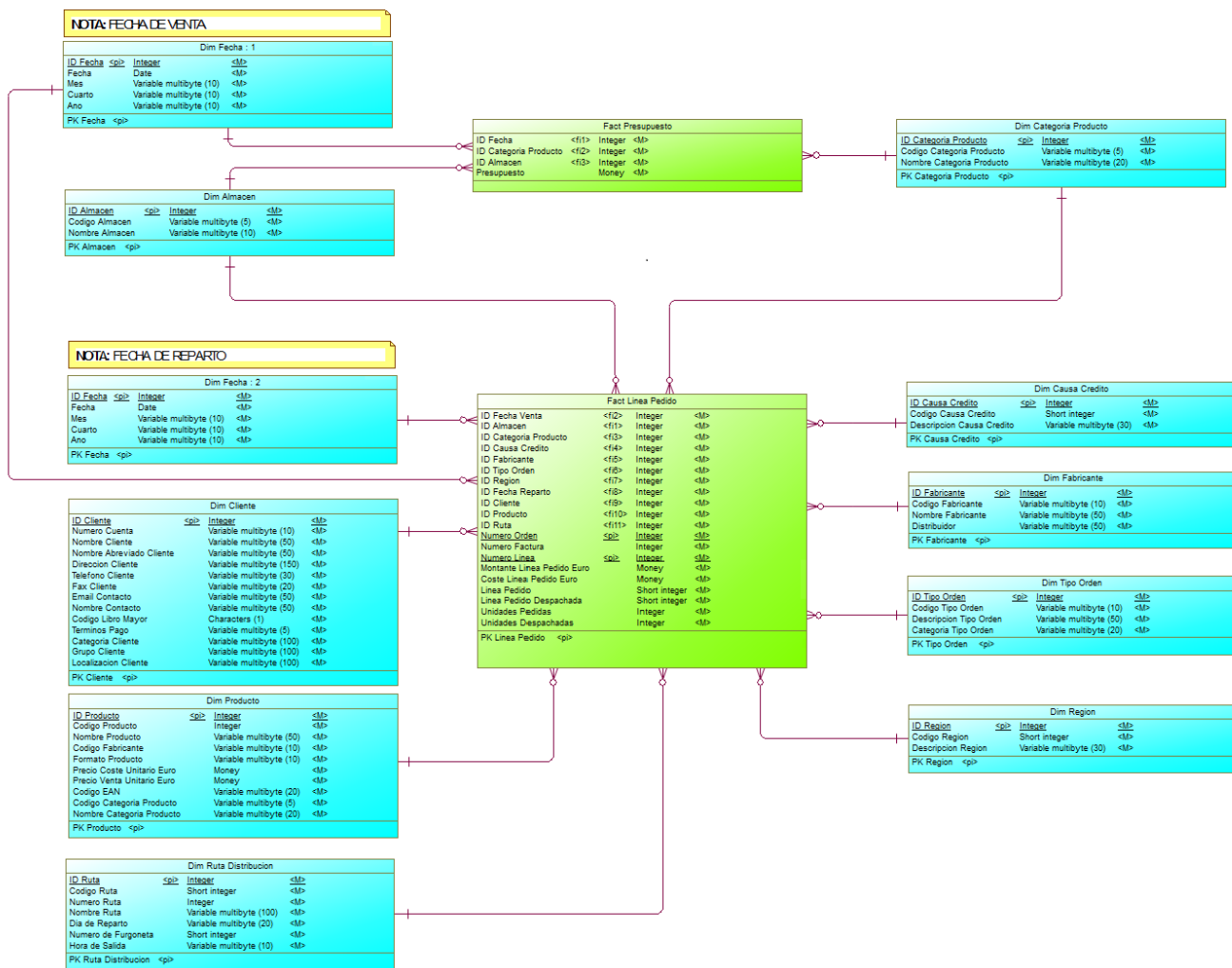


Ilustración 5. Diagrama Conceptual

- Los campos de dimensiones y hechos no permiten almacenar nulos. La razón detrás de esta decisión es que las dimensiones van a ser utilizadas para agrupar datos y realizar consultas que incluyen múltiples tablas de hechos (mediante el uso de dimensiones conformadas), y según el funcionamiento interno de la herramienta, estas consultas pueden ser creadas mediante un FULL OUTER JOIN sobre valores nulos, lo que podría proporcionar información incorrecta.
- La dimensión “Dim Fecha : 1” representa la fecha de venta, y la dimensión “Dim Fecha : 2” representa la fecha de reparto como se indica en las notas adjuntas. La razón por la cual se ha representado así es por la restricción en la herramienta de identificar los diferentes roles de una dimensión.
- Aunque es buena práctica utilizar el tipo de dato más pequeño, se ha optado por utilizar enteros de 4 bytes para almacenar las claves subrogadas para mantener la consistencia en claves subrogadas con el mismo tipo de dato.

Matriz de Dimensiones y Medidas

Por último, y relacionado con lo anterior, proporcionamos la *Tabla 12. Matriz de Dimensiones y Medidas*, que muestra las posibles combinaciones entre dimensiones y hechos. Las celdas resaltadas en amarillo muestran la problemática de la medida “# Órdenes de pedido”, que aunque la podamos combinar con todas las dimensiones, los resultados no necesariamente pueden ser correctos y hay que entender el contexto en el que se utiliza.

Por ejemplo, supongamos una orden 123456 con dos líneas de pedido 1 y 2, asociadas a los productos NUROFEN y LIPITOR. Si combinamos ésta medida con la dimensión Producto, obtendremos una orden para cada producto pero el total debe de marcar una sola orden. Esto significa que existe una orden que contiene dicho producto, pero que el total de órdenes no es necesariamente la suma de estas.

	Fecha Venta	Fecha Reparto	Almacén	Cat. Prod.	Causa Crédito	Cliente	Fabric.	Prod.	Región	Ruta Distrib.	Tipo Orden
€ Ventas	X	X	X	X	X	X	X	X	X	X	X
€ Coste	X	X	X	X	X	X	X	X	X	X	X
# Órdenes de pedido	X	X	X	X	X	X	X	X	X	X	X
# Líneas de pedido	X	X	X	X	X	X	X	X	X	X	X
# Líneas de pedido despachadas	X	X	X	X	X	X	X	X	X	X	X
# Líneas de pedido no despachadas	X	X	X	X	X	X	X	X	X	X	X
# Unidades pedidas	X	X	X	X	X	X	X	X	X	X	X
# Unidades despachadas	X	X	X	X	X	X	X	X	X	X	X
# Unidades no despachadas	X	X	X	X	X	X	X	X	X	X	X
€ Presupuesto	X		X	X							

Tabla 1. Matriz de Dimensiones y Medidas

3. Diseño

Introducción a Microsoft SQL Server

SQL Server es un software propietario de sistema de gestión de bases de datos de la multinacional Microsoft. SQL Server facilita la consulta y el almacenamiento de datos en bases de datos relacionales, que permiten ser referenciados por otras aplicaciones de software.

SQL Server, en el momento de la creación de este documento, se encuentra en su versión **SQL Server 2012**. Para la construcción de este proyecto, se ha seleccionado la edición **SQL Server 2012 Business Intelligence Edition**, que se focaliza en la generación de Business Intelligence Corporativo.

El software, que se ha optado por su versión en inglés, ha sido obtenido a través del programa **Microsoft DreamSpark for Academic Institutions** (<https://www.dreamspark.com/>), que permite la descarga de software Microsoft a estudiantes de la Universitat Oberta de Catalunya para propósitos educativos. Este software tiene una licencia de coste cero asignada al autor del proyecto.

SQL Server 2012 Business Intelligence Edition contiene una serie de servicios orientados al Business Intelligence denominados comúnmente como **Microsoft BI Stack 2012**. Esta pila de servicios permite a los usuarios integrar datos desde múltiples orígenes de datos a un destino SQL Server, construir cubos OLAP, y realizar informes y analíticas corporativas. Estas 3 principales actividades, que se modelan mediante la herramienta **Microsoft Visual Studio**, son llevadas a cabo mediante los servicios de **Integration, Analysis y Reporting**, que son los servicios utilizados para la carga de datos, construcción y despliegue del cubo, y generación de informes respectivamente.

Diseño de la Arquitectura de Software

Arquitectura de Software

En la *Ilustración 6. Diagrama de Arquitectura de Software*, podemos ver la arquitectura de software que se ha utilizado para el desarrollo de este proyecto. Como se puede ver, se han utilizado los servicios mencionados en la sección anterior, a los que se ha añadido Internet Explorer como plataforma de distribución de los informes.

Los datos se almacenarán en una instancia de bases de datos SQL Server, y serán cargados mediante procesos de carga de Integration Services. El cubo será generado por Analysis Services a partir de los datos de la base de datos, y los informes serán creados y actualizados mediante Reporting Services con datos procedentes del cubo de datos. Los usuarios simplemente tendrán que acceder al portal de Reporting Services para seleccionar el informe deseado y actualizarlo.

Todos los componentes de esta arquitectura han sido implementados en una máquina virtual que nos ha permitido el desarrollo en una plataforma diferente al sistema operativo host.

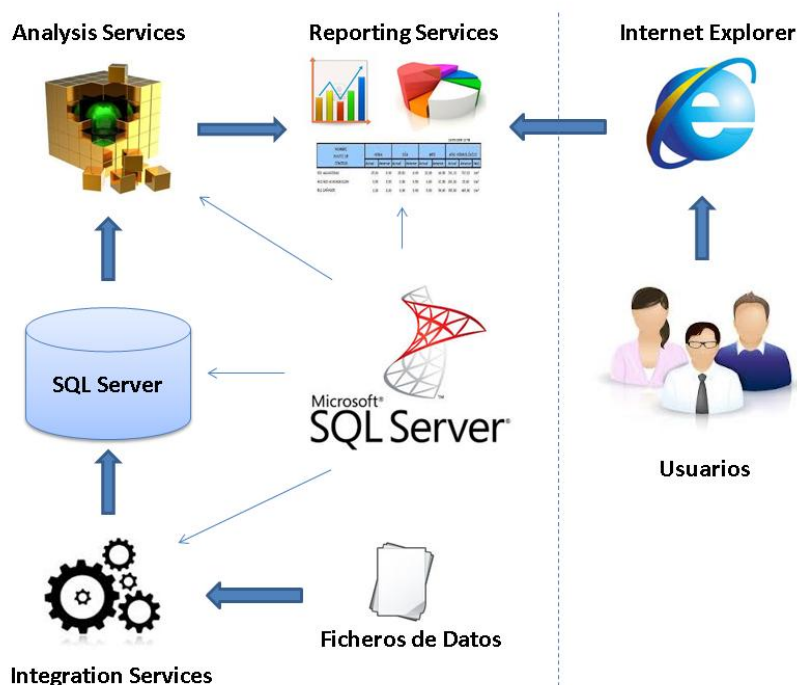


Ilustración 6. Diagrama de Arquitectura de Software

Máquina Virtual y Sistema Operativo

La máquina virtual se ha construido mediante el software de virtualización **Oracle VirtualBox Versión 4.1.6 r74713**. En el momento de la finalización del proyecto, con el software instalado y el proyecto finalizado, la máquina virtual ocupa unos 30GB, que se intentarán reducir para la entrega final.

La máquina virtual creada se llama **"W2008R2"** y tiene instalado el sistema operativo **Windows 2008 Server Enterprise SP2 Data Center Edition 32-bit** y configurada para usar **1GB de memoria** y **1 procesador**.

El sistema operativo tiene dos usuarios administradores creados:

- Administrator (contraseña: 123456789): usuario administrador por defecto en la instalación. Acceso total a la máquina. Es el usuario que se ha utilizado para la instalación del software y

para la construcción de este proyecto. Los informes en la web se podrán acceder mediante el uso de este usuario. Su contraseña no caduca.

- sqlserver (contraseña: sqlserver2012): usuario utilizado exclusivamente para ejecutar los servicios de Microsoft SQL Server.

Software Instalado

El software que se ha instalado en esta máquina virtual es el siguiente:

- Microsoft SQL Server 2012 Business Intelligence Edition, instalado en inglés.
- ArgoUML, para la creación de los casos de uso del proyecto.
- 7-zip, para comprimir los proyectos y realizar copias de seguridad.
- .NET Framework 3.5 SP1, necesario para la instalación de SQL Server.

Inicio de los Servicios Microsoft BI

Al tratarse de una máquina virtual con pocos recursos, se ha decidido no lanzar los servicios Microsoft BI al iniciar el sistema operativo ya que es posible que no se lancen correctamente. Para ello, se han habilitado dos iconos en el Escritorio del usuario “Administrador”, uno para iniciar los servicios y otro para pararlos, como se puede ver en la *Ilustración 7. Iconos para iniciar / parar los servicios Microsoft BI***Error! Reference source not found.**

Antes de iniciar los servicios se recomienda iniciar la máquina virtual totalmente y dejar al sistema operativo finalizar los procesos necesarios. En caso de que existan problemas, se recomienda parar primero los servicios e iniciarlos a continuación.

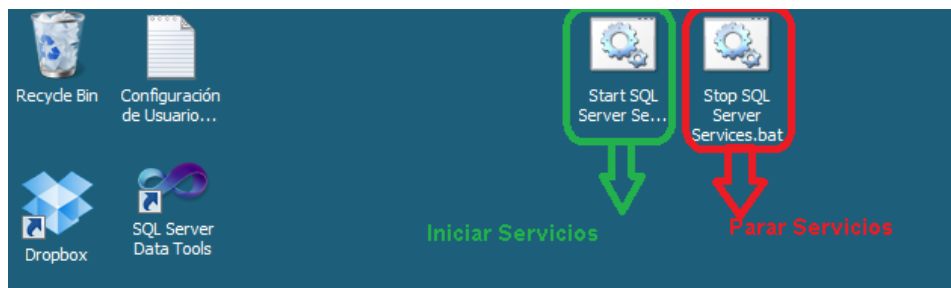


Ilustración 7. Iconos para iniciar / parar los servicios Microsoft BI

Diseño de la Arquitectura de Hardware

Arquitectura de Hardware

En la *Ilustración 8. Diagrama de Arquitectura de Hardware*, podemos ver la arquitectura de hardware que se ha utilizado para este proyecto. Si bien todos los servicios están instalados dentro de un mismo servidor, estos pueden ser perfectamente distribuidos en diferentes servidores, dependiendo de la configuración que los administradores de sistema consideren oportuna. Por ejemplo, es posible que los procesos de Integration Services estén en un servidor diferente al de la base de datos SQL Server para evitar que las cargas de datos utilicen recursos dedicados a la base de datos.

La solución implementada se puede desplegar tanto en arquitecturas con un solo servidor como en arquitecturas distribuidas.

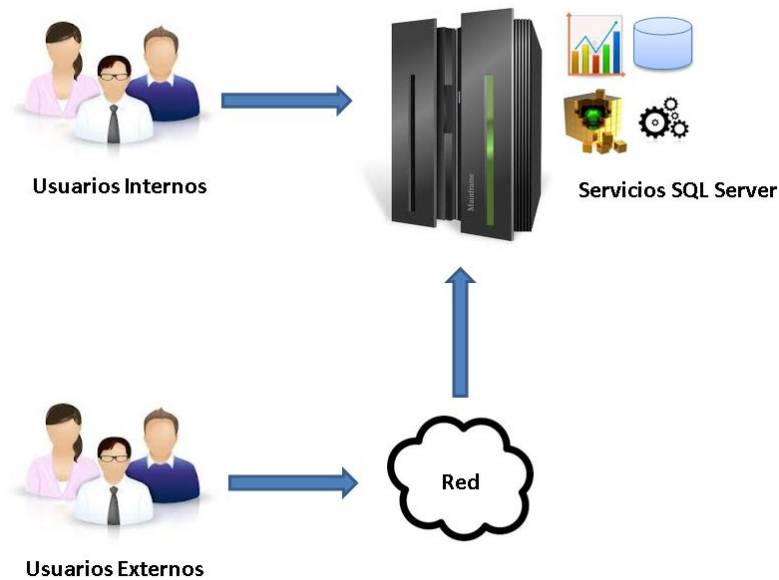


Ilustración 8. Diagrama de Arquitectura de Hardware

Diseño de la Base de Datos

Modelo Entidad / Relación

En la figura Ilustración 9. Modelo Entidad-Relación, podemos ver el modelo entidad relación generado a partir del modelo conceptual. Todas las relaciones entre las dimensiones y las tablas de hechos son 1..N mediante las claves subrogadas.

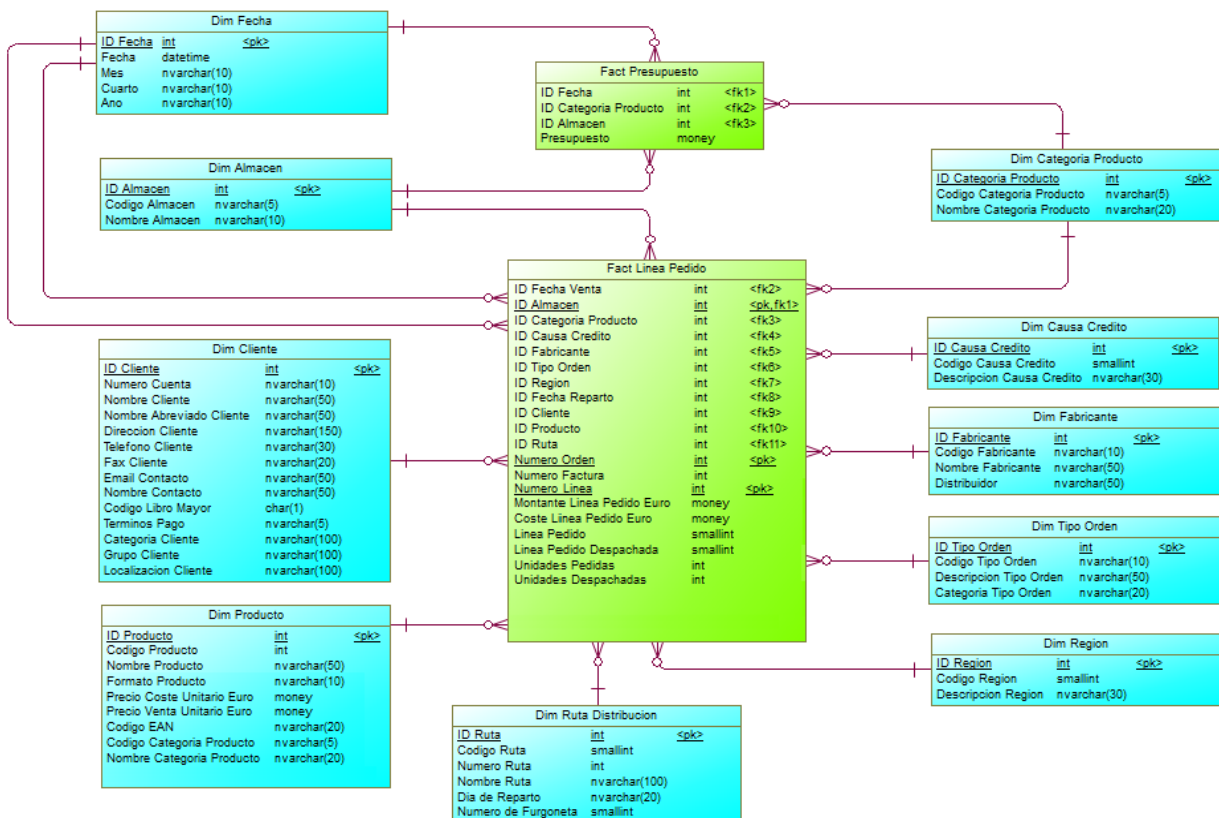


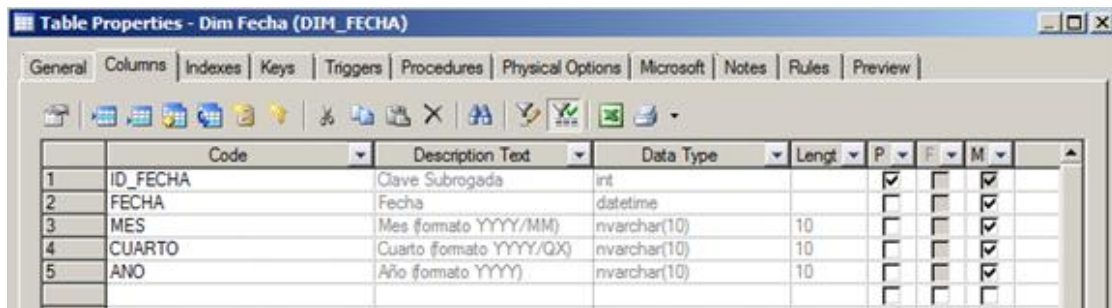
Ilustración 9. Modelo Entidad-Relación

Las características principales de este modelo son las siguientes:

- Se especifican los tipos de datos que SQL Server va a utilizar.
- La tabla “Fact Presupuesto” no tiene clave primaria como se ha indicado anteriormente.
- Existe una única dimensión Fecha con diferentes roles (Fecha Venta y Fecha Reparto).
- Ningún campo de las tablas del modelo aceptará nulos.

Para finalizar con el modelo entidad-relación, se muestran las propiedades de cada una de las tablas especificadas: se indica de qué clase de entidad se trata (dimensión/hecho), el tipo de dimensión si aplica (conformada o no, tipo X) y cuál es el origen de datos (fichero/carga manual).

DIM FECHA

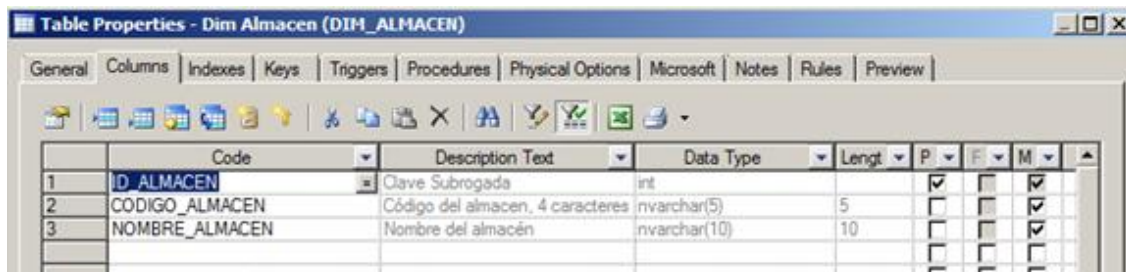


	Code	Description Text	Data Type	Length	P	F	M
1	ID_FECHA	Clave Subrogada	int		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	FECHA	Fecha	datetime		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	MES	Mes (formato YYYY/MM)	nvarchar(10)	10	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	CUARTO	Cuarto (formato YYYY/QX)	nvarchar(10)	10	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	AÑO	Año (formato YYYY)	nvarchar(10)	10	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 10. Propiedades de Dimensión Fecha

Dimensión de *Tipo I* con carga manual. Conformada en el rol Fecha de Venta con ambas tablas de hecho.

DIM ALMACEN

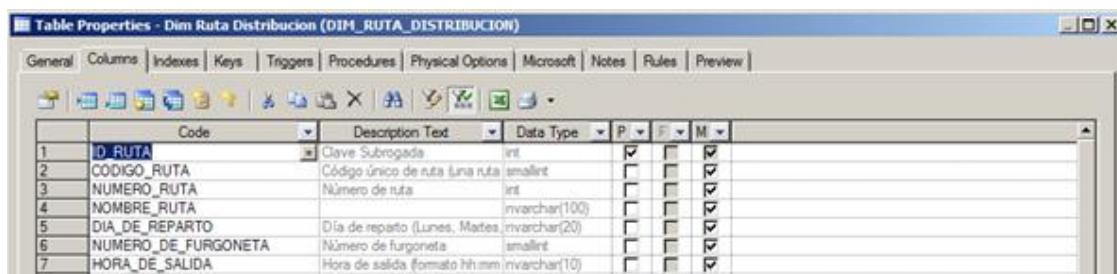


	Code	Description Text	Data Type	Length	P	F	M
1	ID_ALMACEN	Clave Subrogada	int		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	CODIGO_ALMACEN	Código del almacen, 4 caracteres	nvarchar(5)	5	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	NOMBRE_ALMACEN	Nombre del almacén	nvarchar(10)	10	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 11. Propiedades de Dimensión Almacén

Dimensión de *Tipo I* con carga manual (3 registros). Conformada con ambas tablas de hecho.

DIM RUTA DISTRIBUCION



	Code	Description Text	Data Type	Length	P	F	M
1	ID_RUTA	Clave Subrogada	int		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	CODIGO_RUTA	Código único de ruta (una ruta)	smallint		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	NUMERO_RUTA	Número de ruta	int		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	NOMBRE_RUTA		nvarchar(100)		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	DIA_DE_REPARTO	Día de reparto (Lunes, Mates)	nvarchar(20)		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	NUMERO_DE_FURGONETA	Número de furgoneta	smallint		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
7	HORA_DE_SALIDA	Hora de salida (formato hh:mm)	nvarchar(10)		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 12. Propiedades de Dimensión Ruta Distribución

Dimensión de *Tipo I* con datos del fichero “ruta_distribucion.csv”. No conformada.

DIM PRODUCTO

	Code	Description Text	Data Type	P	F	M
1	ID_PRODUCTO	Clave Subrogada	int	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	CODIGO_PRODUCTO	Código del producto	int	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	NOMBRE_PRODUCTO	Nombre del producto	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	FORMATO_PRODUCTO	Formato del producto (Pack,	nvarchar(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	PRECIO_COSTE_UNITARIO_EURO	Precio de coste unitario en Eur	money	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	PRECIO_VENTA_UNITARIO_EURO	Precio de venta unitario en Eu	money	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
7	CODIGO_EAN	Código EAN (European Article	nvarchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
8	CODIGO_CATEGORIA_PRODUCTO	Código de categoría	nvarchar(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
9	NOMBRE_CATEGORIA_PRODUCTO	Nombre de categoría	nvarchar(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 13. Propiedades de Dimensión Producto

Dimensión de *Tipo I* con datos del fichero "productos.csv". No conformada.

DIM CLIENTE

	Code	Description Text	Data Type	Length	P	F	M
→	ID_CLIENTE	Clave Subrogada	int		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	NUMERO_CUENTA	Número de cuenta del cliente. Nivel 4	nvarchar(10)	10	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	NOMBRE_CLIENTE	Nombre del cliente	nvarchar(50)	50	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	NOMBRE_ABREVIADO_CLIENTE	Nombre abreviado del cliente	nvarchar(50)	50	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	DIRECCION_CLIENTE	Dirección del cliente	nvarchar(150)	150	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	TELEFONO_CLIENTE	Teléfono del cliente	nvarchar(30)	30	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
7	FAX_CLIENTE	Fax del cliente	nvarchar(20)	20	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
8	EMAIL_CONTACTO	Email de contacto	nvarchar(50)	50	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
9	NOMBRE_CONTACTO	Nombre de contacto	nvarchar(50)	50	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10	CODIGO_LIBRO_MAYOR	Código de libro mayor (S o R)	char(1)	1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
11	TERMINOS_PAGO	Términos de pago	nvarchar(5)	5	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
12	CATEGORIA_CLIENTE	Categoría Cliente. Nivel 1	nvarchar(100)	100	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
13	GRUPO_CLIENTE	Grupo Cliente. Nivel 2	nvarchar(100)	100	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
14	LOCALIZACION_CLIENTE	Localización Cliente. Nivel 3	nvarchar(100)	100	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 14. Propiedades de Dimensión Cliente

Dimensión de *Tipo I* con datos del fichero "clientes.csv". No conformada.

DIM REGION

	Code	Description Text	Data Type	Length	P	F	M
1	ID_REGION	Clave Subrogada	int		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	CODIGO_REGION	Código de región	smallint		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	DESCRIPCION_REGION	Descripción de región	nvarchar(30)	30	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 15. Propiedades de Dimensión Región

Dimensión de *Tipo I* con datos del fichero "region.csv". No conformada.

DIM TIPO ORDEN

	Code	Description Text	Data Type	Length	P	F	M
1	ID_TIPO_ORDEN	Clave Subrogada	int		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	CODIGO_TIPO_ORDEN	Código de tipo de orden (SINVXX, SCRDXX)	nvarchar(10)	10	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	DESCRIPCION_TIPO_ORDEN	Descripción de tipo de orden	nvarchar(50)	50	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	CATEGORIA_TIPO_ORDEN	Categoría de tipo de orden (INVOICE, CREDIT)	nvarchar(20)	20	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 16. Propiedades de Dimensión Tipo de Orden

Dimensión de *Tipo I* con datos del fichero "tipo_de_orden.csv". No conformada.

DIM FABRICANTE

	Code	Description Text	Data Type	Length	P	F	M
1	ID_FABRICANTE	Clave Subrogada	int		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	CODIGO_FABRICANTE	Código de fabricante	nvarchar(10)	10	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	NOMBRE_FABRICANTE	Nombre de fabricante	nvarchar(50)	50	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	DISTRIBUIDOR	Distribuidora del fabricante	nvarchar(50)	50	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 17. Propiedades de Dimensión Fabricante

Dimensión de *Tipo I* con datos del fichero "fabricante.csv". No conformada.

DIM CAUSA CREDITO

	Code	Description Text	Data Type	Length	P	F	M
→	ID_CAUSA_CREDITO	Clave subrogada	int		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	CODIGO_CAUSA_CREDITO	Código de causa de crédito	smallint		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	DESCRIPCION_CAUSA_CREDITO	Descripción de causa de crédito	nvarchar(30)	30	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 18. Propiedades de Dimensión Causa Crédito

Dimensión de *Tipo I* con datos del fichero "causa_creditos.csv". No conformada.

DIM CATEGORIA PRODUCTO

	Code	Description Text	Data Type	Length	P	F	M
1	ID_CATEGORIA_PRODUCTO	Clave Subrogada	int		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	CODIGO_CATEGORIA_PRODUCTO	Código de categoría	nvarchar(5)	5	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	NOMBRE_CATEGORIA_PRODUCTO	Nombre de categoría	nvarchar(20)	20	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 19. Propiedades de Dimensión Categoría de Producto

Dimensión de *Tipo I* con datos del fichero "categoria_de_producto.csv".
Conformada con ambas tablas de hechos.

FACT PRESUPUESTO

	Code	Description Text	Data Type	Len	P	F	M
1	ID_FECHA	Clave Subrogada Dim Fecha	int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	ID_CATEGORIA_PRODUCTO	Clave Subrogada Dim Categoría Pr	int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	ID_ALMACEN	Clave Subrogada Dim Almacén	int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	PRESUPUESTO	Montante del presupuesto en Euros	money		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 20. Propiedades de Hechos Presupuesto

Tabla de hechos transaccional que se cargará a partir del fichero “*presupuesto.csv*”. No contiene dimensiones degeneradas.

FACT LINEA PEDIDO

	Code	Description Text	Data Type	P	F	M
1	ID_FECHA_VENTA	Clave Subrogada Dim Fecha (venta)	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	ID_ALMACEN	Clave Subrogada Dim Almacén	int	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	ID_CATEGORIA_PRODUCTO	Clave Subrogada Dim Categoría de Producto	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	ID_CAUSA_CREDITO	Clave Subrogada Dim Causa Crédito	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	ID_FABRICANTE	Clave Subrogada Dim Fabricante	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	ID_TIPO_ORDEN	Clave Subrogada Dim Tipo de orden	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	ID_REGION	Clave Subrogada Dim Región	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	ID_FECHA_REPARTO	Clave Subrogada Dim Fecha (reparto)	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	ID_CLIENTE	Clave Subrogada Dim Cliente	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	ID_PRODUCTO	Clave Subrogada Dim Producto	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	ID_RUTA	Clave Subrogada Dim Ruta	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	NUMERO_ORDEN	Número de orden	int	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
13	NUMERO_FACTURA	Número de factura	int	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
14	NUMERO_LINEA	Número de línea de pedido	int	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
15	MONTANTE_LINEA_PEDIDO_EURO	Montante de ventas de la línea de pedido en Euros	money	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
16	COSTE_LINEA_PEDIDO_EURO	Montante de coste de la línea de pedido en Euros	money	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
17	LINEA_PEDIDO	Contador de líneas de pedido (1 indica que la línea es de p	smallint	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
18	LINEA_PEDIDO_DESPACHADA	Contador de líneas de pedido despachadas (1 indica que la	smallint	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
19	UNIDADES_PEDIDAS	Contador de unidades pedidas en una línea de pedido.	int	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
20	UNIDADES_DESPACHADAS	Contador de unidades despachadas en una línea de pedido	int	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 21. Propiedades de Hechos Línea Pedido

Tabla de hechos transaccional que se cargará a partir de los ficheros “*líneas_pedido_XXXX_2012.csv*”. Contiene las siguientes dimensiones degeneradas: *no. orden*, *no. factura* y *no. línea*.

Instancia y Base de Datos

El nombre de la instancia del servidor de base de datos se denomina “**SQL2012W2K8**”. Esta instancia está creada con un *collation*⁴ **SQL_Latin1_General_CP1_CI_AS** y tiene asignada como máximo 1GB de memoria. Está configurada para aceptar tanto *Windows Authentication* como *SQL Server Authentication*. La primera de ellas permite acceder mediante las credenciales del usuario que ha iniciado sesión, mientras que la segunda se basa en los usuarios creados internamente en la base de datos.

A esta se puede acceder mediante la herramienta SQL Server Management Studio (SSMS) mediante dos usuarios con permisos de administración sobre esta instancia: el usuario “**Administrator**”, utilizando

⁴ Característica de los sistemas de gestión de base de datos que permite indicar el tipo de ordenación alfanumérica.

Windows Authentication (ver Ilustración 22. Acceso a SQL Server con usuario "Administrator"), o bien el usuario "sa" o súper-administrador utilizando SQL Server Authentication (ver Ilustración 23. Acceso a SQL Server con usuario "sa"). Para la creación de capturas de pantalla, utilizaremos el primero de ellos.

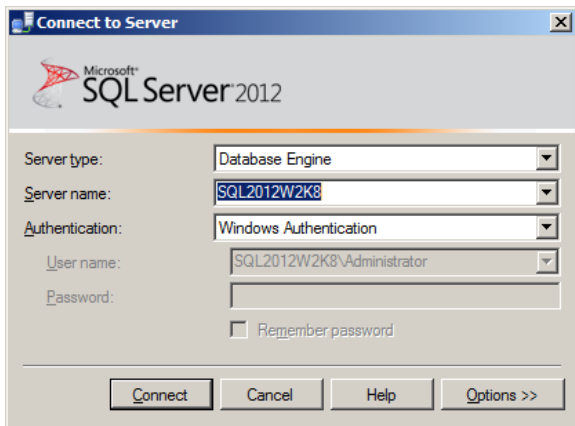


Ilustración 22. Acceso a SQL Server con usuario "Administrator"



Ilustración 23. Acceso a SQL Server con usuario "sa"

Para este proyecto se ha creado una base de datos llamada **dw_ventas**, configurada en modo de recuperación **simple** (modo en el que no se requieren copias de seguridad de los logs), con los siguientes parámetros físicos:

- **dw_ventas** está creada con la *collation SQL_Latin1_General_CP1_CI_AS*.
- Se han creado tres ficheros de datos, el primero de ellos denominado "*fd_primary.ndf*" (por defecto, obligatorio en SQL Server), el segundo "*fd_ventas.ndf*" que almacenará las tablas y datos de nuestro proyecto, y el tercero de ellos "*fd_log.ldf*", que almacenará el log de transacciones de nuestra base de datos. Estos ficheros están asociados cada uno de ellos a un grupo de ficheros concreto (*filegroup*, agrupaciones lógicas de ficheros de datos). En el caso del fichero "*fd_ventas.ndf*", este se ha asociado a un *filegroup* FG_VENTAS, creado para tal propósito. Esto es así para almacenar lógicamente los elementos creados en la base de datos y permitir un mantenimiento sencillo.
- Cada uno de los ficheros de datos están limitados en espacio. Así, el fichero que contendrá los datos de Ventas está limitado a 2GB de datos (suficientes para este proyecto). Lo mismo sucede para el fichero primario y de log, ambos limitados a un 1GB de datos máximo.

El código SQL utilizado para la generación de esta base de datos se puede ver en el **Apéndice A – Código SQL**, apartado **Script SQL de creación de la base de datos**.

Esquemas

Dentro de la base de datos, se han creado una serie de esquemas (estructuras lógicas dentro de la base de datos para agrupar elementos) que nos permitirán gestionar la base de datos de manera más eficiente. Los esquemas creados son los siguientes:

- **STG**: esquema que almacenará tablas temporales y vistas de control para la carga de datos.
- **DIM**: esquema que almacenará las tablas de dimensiones.
- **FACT**: esquema que almacenará las tablas de hechos.

La razón principal por la cual se utilizan esquemas es para conceder permisos a nivel de esquema (agrupación de elementos) y no de tabla, simplificando el mantenimiento de la seguridad.

El código SQL utilizado para la generación de esta base de datos se puede ver en el **Apéndice A – Código SQL**, apartado **Script SQL de creación de los esquemas**.

Cuentas de Servicio

Se han creado 2 cuentas de servicio en SQL Server para gestionar, por un lado la carga de datos y por otro, la lectura de datos desde Analysis Services. Estas cuentas son las siguientes:

- **dw_etl** (contraseña: PFCVentas2013): usuario creado para la realización de la carga de datos mediante Integration Services. Este usuario tendrá permisos de lectura, escritura y ejecución sobre los 3 esquemas definidos anteriormente.
- **dw_consulta** (contraseña: PFCVentasConsulta2013): usuario creado para la creación del cubo de datos y la creación de informes con SQL Server como origen de datos. Este usuario solamente tendrá permisos de lectura sobre los esquemas DIM y FACT (STG no es necesario).

Para la gestión de los permisos, se han generado dos roles:

- **rol_dw_etl**: gestiona los permisos del usuario **dw_etl**.
- **rol_dw_consulta**: gestiona los permisos del usuario **dw_consulta**.

El código SQL utilizado para la generación de esta base de datos se puede ver en el **Apéndice A – Código SQL**, apartados **Script SQL de creación de la cuenta de servicio “dw_etl”** y **Script SQL de creación de la cuenta de servicio “dw_consulta”**.

Tablas

Las siguientes tablas han sido creadas para almacenar la información de ventas y presupuesto (formato “nombre_esquema.nombre_tabla”):

- **Dimensiones**: todas las dimensiones disponen de un registro con identificador subrogado = -1 y asociado a descripciones UNKNOWN para asociar a valores que no existen en la dimensión.
 - o *DIM.DIM_ALMACEN*, para almacenar datos del almacén. Carga manual.
 - o *DIM.DIM_CATEGORIA_PRODUCTO*, para almacenar datos de categoría de producto.
 - o *DIM.DIM_CAUSA_CREDITO*, para almacenar datos de causa de crédito.
 - o *DIM.DIM_CLIENTE*, para almacenar datos del cliente.
 - o *DIM.DIM_FABRICANTE*, para almacenar datos del fabricante.
 - o *DIM.DIM_FECHA*, para almacenar fechas (venta / reparto). Carga manual.
 - o *DIM.DIM_PRODUCTO*, para almacenar los datos del producto.
 - o *DIM.DIM_REGION*, para almacenar los datos de regiones.
 - o *DIM.DIM_RUTA_DISTRIBUCION*, para almacenar los datos de las rutas de distribución.
 - o *DIM.DIM_TIPO_ORDEN*, para almacenar los datos de tipos de orden.
- **Hechos**:
 - o *FACT.FACT_LINEA_PEDIDO*, para almacenar las líneas de pedido.
 - o *FACT.FACT_PRESUPUESTO*, para almacenar las transacciones de presupuesto.
- **Temporales**:
 - o *STG.STG_LINEA_DE_PEDIDO*, para almacenar temporalmente las líneas de pedido leídas desde el fichero de datos. Se trunca en cada carga total / incremental.
- **Control**:
 - o *STG.CTL_FICHEROS*, para realizar la carga de los ficheros, tanto de dimensiones como hechos. Se trunca cada vez que se carga un fichero de datos.

El código SQL utilizado para la generación de las tablas indicadas se puede ver en la el **Apéndice A – Código SQL**, apartados **Script SQL de creación de dimensiones**, **Script SQL de creación de tablas de hechos**, **Script SQL de creación de tablas temporales** y **Script SQL de creación de tablas de control**.

Vistas

La única vista definida es la siguiente, *STG.VW_STG_LINEA_PEDIDO*, cuyo propósito es la de leer y transformar los datos de la tabla temporal *STG.STG_LINEA_DE_PEDIDO*.

El código SQL utilizado para la generación de la vista se puede ver en el **Apéndice A – Código SQL**, apartado **Script SQL de creación de vistas**.

Funciones

La única función definida es la siguiente, *DIM.FN_FECHA_TO_ID*, utilizada para transformar fechas en claves subrogadas enteras en formato YYYYMMDD. Ésta acepta un parámetro de tipo fecha y retorna un valor entero. Por ejemplo, si se le pasa una fecha 01/12/2013, el valor retornado será 20131201.

El código SQL utilizado para la generación de la vista se puede ver en el **Apéndice A – Código SQL**, apartado **Script SQL de creación de funciones**.

Procedimientos Almacenados

El único procedimiento almacenado definido es el siguiente, *DIM.SP_DIM_FECHA*, utilizado para cargar la dimensión *DIM.DIM_FECHA*. Éste acepta dos parámetros, el año de inicio y el año de fin. Por ejemplo, si se pasan por parámetro 2012 y 2013 como años de inicio y fin respectivamente, el procedimiento cargará fechas desde el 1 Enero 2012 hasta el 31 Diciembre 2013. Si la fecha ya existe, no se inserta. Si el año de inicio es superior al año de fin, el procedimiento no cargará nada.

El código SQL utilizado para la generación de la vista se puede ver en el **Apéndice A – Código SQL**, apartado **Script SQL de creación de procedimientos almacenados**.

Diseño del Proceso de Carga

Generalmente, los procesos de carga de almacenes de datos siguen un patrón común [2]. Este patrón, que se puede ver en la figura *Ilustración 24. Patrón de Proceso de Carga de un Almacén de Datos*, establece que primero hay que cargar las dimensiones, necesarias para vincular con los hechos (recordemos que las dimensiones se relacionan con los hechos mediante relaciones 1..N).



Ilustración 24. Patrón de Proceso de Carga de un Almacén de Datos

A continuación, se cargarán tablas temporales para cargar los hechos (preparación de claves naturales, limpieza y consolidación de datos en una tabla temporal). El siguiente paso es la carga de los hechos desde estas tablas temporales mediante operaciones de *lookup*, esto es, obtener la clave subrogada de un registro en una dimensión mediante su clave natural. De esta forma, podemos vincular el hecho a una dimensión por claves subrogadas, mejorando el rendimiento de las consultas cuando realizamos operaciones de JOIN.

Por último, aunque no necesario, se suelen calcular y actualizar las estadísticas de las dimensiones y de las tablas de hechos para que el optimizador de consultas del sistema de gestión de base de datos obtenga el mejor plan. Hay casos en los que también se reconstruyen los índices de las tablas de hechos, pero es una tarea que no va a ser realizada en este proyecto. También hay casos en los que, para mejorar el rendimiento de los informes se crea un cubo OLAP. En caso de que la herramienta lo permita, se podría añadir una tarea para actualizar el cubo OLAP una vez que el proceso de carga haya terminado.

En nuestro caso, realizaremos todas las etapas del patrón indicado: las dimensiones se cargarán a partir de los ficheros, se creará una tabla temporal que consolidará los datos de los hechos que proceden de los ficheros de presupuesto y líneas de pedido, se cargarán los hechos mediante operaciones de *lookup*, se actualizarán las estadísticas de las tablas de hechos y se actualizará el cubo OLAP.

Para ello se ha creado un paquete maestro llamado “*Ventas_Maestro.dtsx*” que incluye las cuatro primeras etapas del proceso de carga. Este paquete se encarga de llamar al resto de paquetes que cargan las dimensiones (un paquete por dimensión), las tablas temporales, la carga de hechos de líneas de pedido y el cálculo de las estadísticas. La carga del presupuesto se realiza aparte de forma manual mediante otro paquete dedicado a ello, por lo que no se ha incluido como parte del proceso ETL de carga inicial / incremental.

Como se puede ver en la *Ilustración 25. Paquete “Ventas_Maestro.dtsx”*, la carga de las dimensiones se realiza mediante una agrupación de los paquetes de carga de dimensiones, que se realizan de forma paralela entre ellas (no existen dependencias entre las dimensiones, por lo que se pueden cargar de forma independiente). Esta agrupación tiene una restricción de precedencia que obliga a todos los paquetes de dimensiones a finalizar correctamente antes de pasar a la siguiente etapa. Esta misma restricción, que es “Terminar satisfactoriamente”, está implementada en cada una de las etapas del paquete maestro.

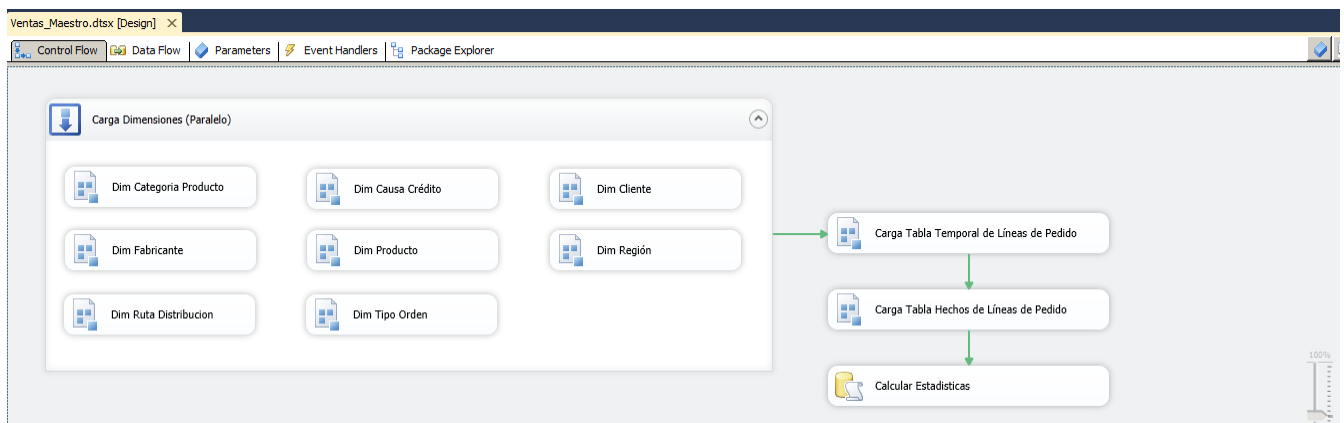


Ilustración 25. Paquete “Ventas_Maestro.dtsx”

La actualización del cubo, que es la última etapa del patrón que hemos visto, se realiza mediante otro paquete llamado “*Ventas_Procesar_Cubo.dtsx*”. Tanto el paquete maestro como el de actualización del cubo se lanzan secuencialmente mediante un SQL Agent job, implementando así el patrón. Este job se explicará en profundidad en las siguientes secciones.

Todos los paquetes implementados, incluyendo el proyecto de Integration Services, tienen definida una contraseña para poder acceder a ellos. Esta contraseña es “**ventas**”, y permite encriptar los datos sensibles relacionados con los orígenes de datos (contraseñas). Esto permite tener un mantenimiento más sencillo, ya que los paquetes por defecto encriptan esta información con la clave del usuario.

Mediante el uso de una contraseña común, el mantenimiento de estos paquetes es más sencillo y evitan la dependencia del usuario que los ha desarrollado.

Además de la contraseña, todos los paquetes utilizan variables con el fin de almacenar datos y referenciarlos de forma más sencilla. Estos datos, por ejemplo, pueden ser la ruta de acceso a los ficheros. Para usar apropiadamente estas variables, todos los paquetes tienen implementadas una serie de configuraciones. Estas configuraciones permiten a cada uno de los paquetes:

- Configurar las conexiones de forma dinámica, de forma que si necesitamos desplegar el proyecto en un nuevo entorno, nos basta con cambiar dicha configuración.
- En el caso del paquete maestro, existe una tercera configuración para gestionar todos estos parámetros que queremos pasar a los paquetes a los que llama.
- Paso de valores desde variables padre. Desde el paquete maestro, podemos pasar valores de variables para que estas se utilicen en el resto de paquetes, y así poder mantener dichas variables desde un solo lugar.

Las dos primeras configuraciones se realizan mediante ficheros de configuraciones. Estos ficheros con formato XML tienen una extensión “.dtsConfig” y nos permiten asignar valores a variables definidas en el paquete. En nuestro proyecto existen 2 ficheros de configuraciones.

- Fichero de conexiones: se llama “ventas_conexiones.dtsConfig” y contiene los parámetros necesarios para conectarse a la base de datos.
- Fichero de parámetros: se llama “ventas_parametros.dtsConfig” y contiene los parámetros necesarios para acceder a los paquetes hijo, rutas de almacenamiento de los ficheros, ruta de almacenamiento de log, nombre de fichero de log, etc.

Estos ficheros de configuración se pueden ver en detalle en el **Apéndice B – Ficheros de Configuración**, apartados **Fichero de Configuración de Conexiones** y **Fichero de Configuración de Parámetros**.

Para poder acceder a estos ficheros de configuración de forma dinámica y con fácil mantenimiento, se han creado dos variables de entorno:

- SSIS_VENTAS_CONEXIONES, que apunta a la ruta donde reside el fichero de conexiones.
- SSIS_VENTAS_PARAMETROS, que apunta a la ruta donde reside el fichero de parámetros.

En la *Ilustración 26. Configuraciones del paquete maestro* se pueden ver las configuraciones creadas para el paquete maestro.

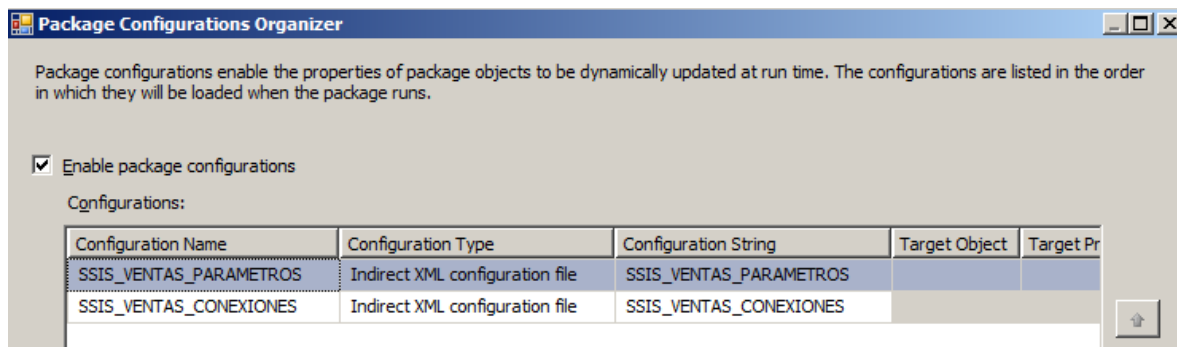


Ilustración 26. Configuraciones del paquete maestro

La última configuración, esto es, el paso de valores a variables hijas, se ha creado en cada uno de los paquetes que son llamados por el maestro y que requieren acceso a ficheros (en concreto, dimensiones y tablas temporales). En concreto, la variable que nos interesa es la ruta donde se han almacenado los

ficheros, como se puede ver en la *Ilustración 27. Configuraciones del paquete de carga de la dimensión Cliente*. De esta forma podemos controlar la ruta de los ficheros desde el fichero de configuraciones de forma dinámica.

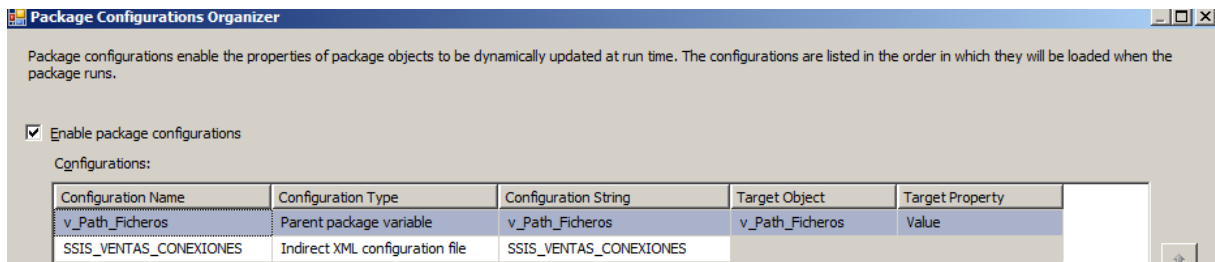


Ilustración 27. Configuraciones del paquete de carga de la dimensión Cliente

Por último, desde un punto de vista de diseño general, se ha creado la carpeta “**SSIS**”, localizada en la unidad C, para almacenar los componentes del proceso de carga. Esta carpeta contiene una carpeta de proyecto llamada “**Ventas**” que contendrá todos los paquetes, ficheros, ficheros de configuración y logs de este sistema. De esta manera, podremos incluir más proyectos y gestionarlos desde una misma ruta.

Cada una de las carpetas de proyecto tendrá 4 subcarpetas:

- *config*, para almacenar los ficheros de configuración. Es aquí a donde apuntan las variables de entorno creadas. En concreto:
 - o *SSIS_VENTAS_CONEXIONES* apunta a **C:\SSIS\Ventas\config\ventas_conexiones.dtsConfig**
 - o *SSIS_VENTAS_PARAMETROS* apunta a **C:\SSIS\Ventas\config\ventas_parametros.dtsConfig**
- *dtsx*, donde se almacenan cada uno de los paquetes *.dtsx* creados.
- *ficheros*, donde se almacenan los ficheros *.csv* dados para el proyecto.
- *log*, donde se almacena el fichero de log de las cargas y que veremos en la siguiente sección.

Gestión de Errores

Para gestionar la calidad de los datos en el almacén de datos, se realizarán las siguientes tareas dentro del proceso de carga:

- Si se encuentra un valor nulo/blanco en una dimensión:
 - Si es un tipo de dato *String*, se substituirá por el literal “*UNKNOWN*” (desconocido).
 - Si es un tipo de dato numérico, se substituirá por un valor cero.
- Si cuando se realiza una operación de *lookup* sobre una dimensión mediante clave natural no se encuentra un registro en la dimensión, esa fila será vinculada a un registro especial creado en las dimensiones para estos casos. Este registro debe existir en todas las dimensiones y tendrá, si es posible, un valor de clave subrogada -1 (en caso de fechas, se usará la fecha 19000101). Este registro tendrá códigos -1 y valores desconocidos (“*UNKNOWN*”). La razón principal por lo que se ha tomado esta decisión es para evitar fallos en la carga por problemas de calidad de datos. Si estos datos cambian en el sistema operacional podrán ser actualizados en sucesivas cargas incrementales, ya que, como se ha indicado anteriormente, el almacén de datos debe de mostrar la última instantánea del sistema operacional.
- Si uno de los campos numéricos del hecho es nulo, este se substituirá por un cero.

Mediante estas técnicas, permitiremos que las cargas finalicen y que los informes muestren los resultados con valores desconocidos, esto último muy útil para asegurar que se solucionen los problemas de datos en el sistema origen.

En caso de que los valores excedan el tipo de dato modelado (valores truncados), o que el valor sea de un tipo de dato diferente al esperado, la carga de datos producirá errores y esta se parará. La razón por la que se ha tomado esta decisión es para asegurar que todos los datos se carguen en la base de datos, de forma que se prime una carga total de los datos ante una incompleta, considerando que esta última podría ser perjudicial para el análisis de los datos al no contar con la totalidad de estos.

Si uno de los ficheros de datos no existe, al igual que en el caso anterior, el proceso de carga continuará sin producir errores, asumiendo que es posible que se mapeen todos los valores de una dimensión al registro especial de desconocidos.

En caso de que el proceso de carga falle por cualquier razón, los errores serán registrados en un fichero de log que podrá ser consultado para identificar y tratar los errores producidos. Tras la solución del problema, la carga puede volver a lanzarse desde el inicio sin tener que realizar ninguna operación de recuperación, ya que los valores de dimensiones y hechos se van a actualizar con los datos leídos.

Fichero de Log

Desde el paquete maestro se genera un fichero de log en formato *ventas_YYYY-MM-DD.log*, donde *YYYY-MM-DD* indica la fecha en la que se ha lanzado el proceso. Este log se genera automáticamente desde el paquete maestro en cada carga y almacena datos acerca de:

- fecha de inicio y fin del paquete maestro, y de los paquetes a los que llama
- si existen *warnings* o errores, estos se registrarán en este fichero de texto.

En caso de que en un mismo día se produzcan múltiples cargas de datos, se utilizará el mismo fichero de log para todas ellas, añadiendo la nueva información generada al final del fichero.

Jobs de SQL Agent

Para la carga de los datos y la actualización del cubo, se han creado 3 SQL Agent Jobs:

- 1. *Ventas - Dimensiones, Líneas de Pedido y Cubo*: este job incluye la carga de dimensiones, hechos de línea de pedido y actualización del cubo de datos. Este es el job que debe de programarse para realizar cargas incrementales.
- 2. *Ventas – Presupuesto*: este job se utiliza para la carga de datos del presupuesto, ya que se tiene que realizar de forma manual.
- 3. *Ventas - Cubo de Datos*: este job se utiliza solamente de forma auxiliar tras la carga del presupuesto, ya que es necesario actualizar el cubo una vez que las dimensiones y ambas tablas de hechos estén cargadas. No se requiere su uso después de cargar el presupuesto (salvo que se cargue el del año siguiente o se generen nuevas versiones).

Carga Inicial e Incremental

Un almacén de datos no se carga solamente una vez, si no que se carga de forma periódica. Por tanto, el proceso de carga tiene que gestionar ambas cargas iniciales, es decir, datos históricos, y cargas incrementales o también denominadas deltas.

Una carga inicial se realiza cuando el almacén de datos se despliega por primera vez en un entorno de producción, generalmente manipulando grandes cantidades de datos de años anteriores. Las cargas

incrementales suelen ser frecuentes una vez que se ha hecho la carga inicial, y generalmente manipula pequeñas cantidades de información, tratándose generalmente de datos que han sido modificados en los últimos días.

Para realizar la **carga inicial del sistema**, se deben de realizar los siguientes pasos:

1. Asegurarse que el proyecto se encuentra correctamente desplegado (estructura de carpetas, paquetes en su ubicación, ficheros de datos en su ubicación, etc.).
2. Asegurarse que las tablas no contienen datos (primera carga).
3. Ejecutar el job “1. Ventas - Dimensiones, Líneas de Pedido y Cubo”, que cargará todas las dimensiones, el hecho de líneas de pedido y actualizará el cubo de datos. El cubo todavía no tiene datos de presupuesto.
4. Ejecutar el job “2. Ventas – Presupuesto”, para cargar la tabla de hechos del presupuesto.
5. Ejecutar el job “3. Ventas - Cubo de Datos”, para actualizar el cubo con datos del presupuesto.

Para realizar **cargas incrementales** de líneas de pedido, basta con ejecutar o programar el job “1. Ventas - Dimensiones, Líneas de Pedido y Cubo”, asumiendo que los ficheros de datos de líneas de pedido han sido modificados.

Si los datos de presupuesto han sido cambiados, entonces se debe de realizar lo siguiente:

1. Eliminar los datos de la tabla de hechos de presupuesto mediante la consulta (utilizar un usuario administrador para ello):


```
truncate table fact.fact_presupuesto;
```
2. Ejecutar el job “2. Ventas – Presupuesto”, para cargar la tabla de hechos del presupuesto.
3. Ejecutar el job “3. Ventas - Cubo de Datos”, para actualizar el cubo con datos del presupuesto.

Paquetes Desarrollados

- *Ventas_Maestro.dtsx*, que ejecuta la carga de dimensiones, tablas temporales, hechos y cálculo de estadísticas.
- *Ventas_Dim_Producto.dtsx*, que lee los datos del fichero de productos, los compara con los datos existentes en la dimensión, y actualiza sus columnas o lo inserta dependiendo de si el registro existe o no respectivamente en la dimensión.
- *Ventas_Dim_Categoria_Producto.dtsx*, que lee los datos del fichero de categorías de producto, las compara con los datos existentes en la dimensión, y actualiza sus columnas o lo inserta dependiendo de si el registro existe o no respectivamente en la dimensión.
- *Ventas_Dim_Causa_Credito.dtsx*, que lee los datos del fichero de causas de crédito, las compara con los datos existentes en la dimensión, y actualiza sus columnas o lo inserta dependiendo de si el registro existe o no respectivamente en la dimensión.
- *Ventas_Dim_Cliente.dtsx*, que lee los datos del fichero de clientes, los compara con los datos existentes en la dimensión, y actualiza sus columnas o lo inserta dependiendo de si el registro existe o no respectivamente en la dimensión.
- *Ventas_Dim_Fabricante.dtsx*, que lee los datos del fichero de fabricantes, los compara con los datos existentes en la dimensión, y actualiza sus columnas o lo inserta dependiendo de si el registro existe o no respectivamente en la dimensión.

- *Ventas_Dim_Region.dtsx*, que lee los datos del fichero de regiones, las compara con los datos existentes en la dimensión, y actualiza sus columnas o lo inserta dependiendo de si el registro existe o no respectivamente en la dimensión.
- *Ventas_Dim_Ruta_Distribucion.dtsx*, que lee los datos de rutas de distribución, las compara con los datos existentes en la dimensión, y actualiza sus columnas o lo inserta dependiendo de si el registro existe o no respectivamente en la dimensión.
- *Ventas_Dim_Tipo_Orden.dtsx*, que lee los datos de tipo de órdenes, las compara con los datos existentes en la dimensión, y actualiza sus columnas o lo inserta dependiendo de si el registro existe o no respectivamente en la dimensión.
- *Ventas_Stg_Linea_Pedido.dtsx*, que lee los datos de los tres ficheros de líneas de pedido, iterando con cada uno de ellos, y los inserta en la tabla temporal, que se trunca en cada carga inicial o temporal.
- *Ventas_Fact_Linea_Pedido.dtsx*, que lee los datos de la tabla temporal transformándolos mediante una vista (elimina nulos, transforma a mayúsculas, etc.), realiza las tareas de look-up con las dimensiones e inserta en los hechos de línea de pedido si la línea no existe, o actualiza la línea en caso contrario.
- *Ventas_Fact_Presupuesto.dtsx*, que lee los datos del fichero de presupuesto, realiza las tareas de look-up con las dimensiones e inserta en los hechos.
- *Ventas_Procesar_Cubo.dtsx*, que se encarga de actualizar el cubo de datos en su totalidad con los datos existentes en la base de datos, primero las dimensiones, a continuación los hechos

Diseño del Cubo de Datos

En la siguiente sección se explicará el diseño del cubo de datos construido en este proyecto, detallando las 4 fases de construcción: **definición del origen de datos**, **definición de la vista de datos y relaciones**, **construcción de las dimensiones** y **construcción del cubo de datos**, incluidas las medidas. Por último, se explicarán las acciones de **publicación** y **actualización** del cubo de datos.

Origen de Datos

Para poder acceder a los datos cargados por Integration Services en la base de datos SQL Server con el fin de generar el cubo de datos, se ha creado un origen de datos denominado “*DW Ventas.ds*”. Este origen de datos utiliza la cuenta de servicio “*dw_consulta*” para leer los datos y acceder a los metadatos de la base de datos (tablas, vistas, etc.).

Vista de Datos

Una vez definido el origen de datos, es necesario generar la **vista de datos y relaciones**, esto es, definir las tablas y campos que vamos a utilizar para la generación del cubo, y las relaciones que tienen estas tablas entre sí, de forma que podamos identificar las dimensiones y los hechos.

La vista que se ha definido se puede ver en la *Ilustración 28. Vista de datos de Ventas y Presupuesto*, en la que se incluyen las tablas de las dimensiones y de hechos para la creación el cubo.

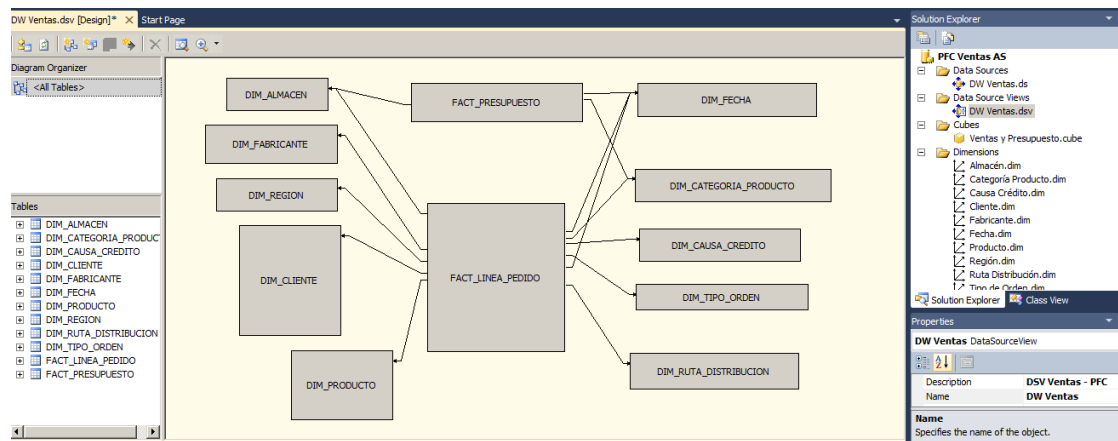


Ilustración 28. Vista de datos de Ventas y Presupuesto

En esta vista se han generado las siguientes variaciones para facilitar la generación del cubo y los informes:

- En la dimensión “DIM_CLIENTE” se ha añadido un campo calculado llamado “CUENTA_NOMBRE” y definido como la concatenación del número de cuenta y del nombre del cliente separados por un guión (“NUMERO_CUENTA + ' - ' + NOMBRE_CLIENTE”).
- En la dimensión “DIM_PRODUCTO” se ha añadido un campo calculado llamado “CODIGO_NOMBRE_PRODUCTO” y definido como la concatenación del código de producto y del nombre del producto separados por un guión (“CAST(CODIGO_PRODUCTO AS VARCHAR(10)) + ' - ' + NOMBRE_PRODUCTO”).
- En la dimensión “DIM_RUTA_DISTRIBUCIÓN” se ha añadido un campo calculado llamado “NUMERO_NOMBRE_DIA_RUTA” y definido como la concatenación del número de ruta, del nombre y del día de reparto separados por un guión (“CAST(NUMERO_RUTA AS VARCHAR(10)) + ' - ' + NOMBRE_RUTA + ' (' + DIA_DE_REPARTO + ')”).
- En la tabla de hechos “FACT_LINEA_PEDIDO” se han añadido los siguientes campos calculados:
 - o “NO_ORDEN” (“CAST(NUMERO_ORDEN AS VARCHAR(10)) + '#' + CAST(ID_ALMACEN AS VARCHAR(5))”), como medida para calcular el número de órdenes teniendo en cuenta que un número de orden puede estar repetido en varios almacenes.
 - o “LINEA_PEDIDO_NO_DESPACHADA” (“LINEA_PEDIDO - LINEA_PEDIDO_DESPACHADA”), como medida para calcular el número de líneas que no se han despachado.
 - o “UNIDADES_NO_DESPACHADAS” (“UNIDADES_PEDIDAS - UNIDADES_DESPACHADAS”), como medida para calcular el número de unidades que no se han despachado por estar fuera de stock.
 - o “MARGEN_EURO” (“MONTANTE_LINEA_PEDIDO_EURO - COSTE_LINEA_PEDIDO_EURO”), como medida para calcular el margen de la línea de pedido definido como montante de la línea menos el coste.

Dimensiones

Todas las dimensiones que se han generado utilizan un modo de almacenamiento MOLAP (base de datos multidimensional). Las dimensiones que se han generado se pueden ver en la Ilustración 29. Dimensiones en Analysis Services:

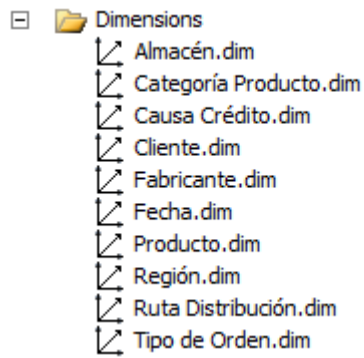


Ilustración 29. Dimensiones en Analysis Services

Cubo de Datos

Se ha definido un cubo de datos llamado “Ventas y Presupuesto.cube” que contendrá los datos de las líneas de pedido y de presupuesto. Este cubo está generado usando un modo de almacenamiento *MOLAP*, al igual que las dimensiones, esto es, el resultado se almacenará en una base de datos multidimensional.

En este cubo se han definido las siguientes medidas, que tendremos disponibles para la realización de los informes:

- € Ventas, definida como *FACT_LINEA_PEDIDO.MONTANTE_LINEA_PEDIDO_EURO*, con función de agregación *SUM* y formato *Moneda*.
- € Coste, definida como *FACT_LINEA_PEDIDO.COSTE_LINEA_PEDIDO_EURO*, con función de agregación *SUM* y formato *Moneda*.
- € Margen, definida como *FACT_LINEA_PEDIDO.MARGEN_EURO* (campo calculado), con función de agregación *SUM* y formato *Moneda*.
- # Líneas de Pedido Recibidas, definida como *FACT_LINEA_PEDIDO.LINEA_PEDIDO*, con función de agregación *SUM* y formato *Entero*.
- # Líneas de Pedido Despachadas, definida como *FACT_LINEA_PEDIDO.LINEA_PEDIDO_DESPACHADA*, con función de agregación *SUM* y formato *Entero*.
- # Líneas de Pedido No Despachadas, definida como *FACT_LINEA_PEDIDO.LINEA_PEDIDO_NO_DESPACHADA* (campo calculado), con función de agregación *SUM* y formato *Entero*.
- # Unidades Pedidas, definida como *FACT_LINEA_PEDIDO.UNIDADES_PEDIDAS*, con función de agregación *SUM* y formato *Entero*.
- # Unidades Despachadas, definida como *FACT_LINEA_PEDIDO.UNIDADES_DESPACHADAS*, con función de agregación *SUM* y formato *Entero*.
- # Unidades No Despachadas, definida como *FACT_LINEA_PEDIDO.UNIDADES_NO_DESPACHADAS* (campo calculado), con función de agregación *SUM* y formato *Entero*.
- € Presupuesto, definida como *FACT_PRESUPUESTO.PRESUPUESTO*, con función de agregación *SUM* y formato *Moneda*.
- # Órdenes de Pedido, definida como *FACT_LINEA_PEDIDO.NO_ORDEN* (campo calculado), con función de agregación *COUNT DISTINCT* y formato *Entero*.

- Se ha creado también un cálculo llamado “€ Ventas Mes Anterior (Fecha Venta)” que nos permite, dado un mes de venta, obtener las ventas del mes de venta anterior. Este cálculo está realizado en MDX y la formula es la siguiente: $[[Fecha\ Venta].[Fecha].PrevMember, [Measures].[\text{€ Ventas}]]$.

Una vez definidas las medidas y las particiones necesarias, tenemos que indicar qué dimensiones podemos utilizar con qué medidas. Esto se realiza en la pestaña “Dimension Usage” (uso de dimensiones), la cual nos permite establecer las relaciones entre las particiones de las medidas y las dimensiones. Véase pues la *Ilustración 30. Combinaciones posibles de dimensiones y medidas en el cubo de datos*, en la que destaca la imposibilidad de combinar las dimensiones “Producto”, “Categoría de Producto” y “Fabricante” con las medidas de “Órdenes de Pedido”, debido a la problemática comentada en el diseño dimensional.

Dimensiones	Hechos Líneas de Pedido	Hechos Presupuesto	Hechos Órdenes de Pedido
Almacén	Id Almacén	Id Almacén	Id Almacén
Categoría Producto	Id Categoría de Producto	Id Categoría de Producto	
Causa Crédito	Id Causa de Crédito		Id Causa de Crédito
Cliente	Id Cliente		Id Cliente
Fabricante	Id de Fabricante		
Fecha (Fecha Repa...)	Id Fecha		Id Fecha
Fecha (Fecha Venta)	Id Fecha	Id Fecha	Id Fecha
Producto	Id Producto		
Región	Id Región		Id Región
Ruta Distribución	Id Ruta Distribución		Id Ruta Distribución
Tipo de Orden	Id Tipo de Orden		Id Tipo de Orden

Ilustración 30. Combinaciones posibles de dimensiones y medidas en el cubo de datos

Despliegue del Cubo

Tras finalizar el diseño del cubo de datos, éste hay que desplegarlo en la base de datos multidimensional. Esta es una tarea que, una vez finalizada, no hay que volver a realizarla, ya que el proceso ETL se encargará de actualizar el cubo. Un ejemplo de procesado (y actualización) desde el proyecto generado en Visual Studio se puede ver en la *Ilustración 31. Procesado de dimensiones y cubo en Analysis Services*.

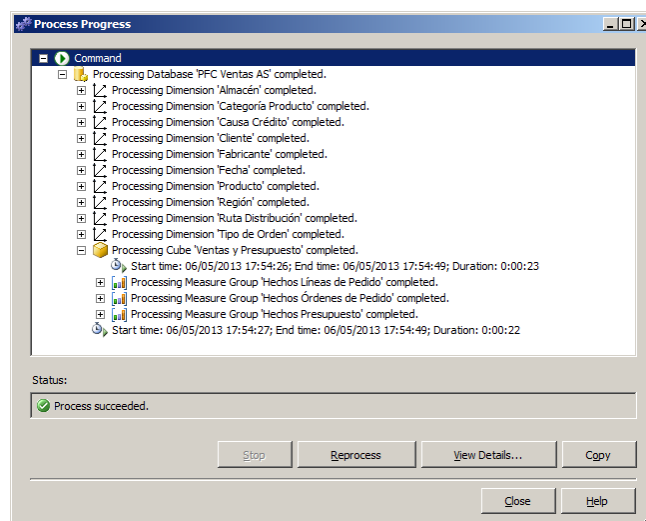


Ilustración 31. Procesado de dimensiones y cubo en Analysis Services

El cubo se ha publicado en una base de datos Analysis Services llamada "PFC Ventas AS", a la cual podemos acceder mediante SQL Server Management Studio y seleccionando la instancia de Analysis Services de la máquina virtual, como se puede ver en la *Ilustración 32. Base de datos "PFC Ventas AS" en Analysis Services.*

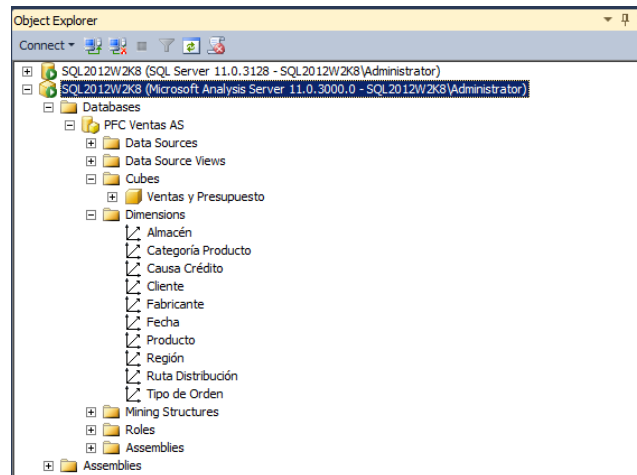


Ilustración 32. Base de datos "PFC Ventas AS" en Analysis Services

Por último, en la *Ilustración 33. Ejemplo de informe en el browser de Analysis Service,* podemos ver un ejemplo de informe realizado en el browser del cubo una vez desplegado y procesado, donde se muestran las líneas de pedido recibidas y despachadas en el almacén de Dublín por mes de venta:

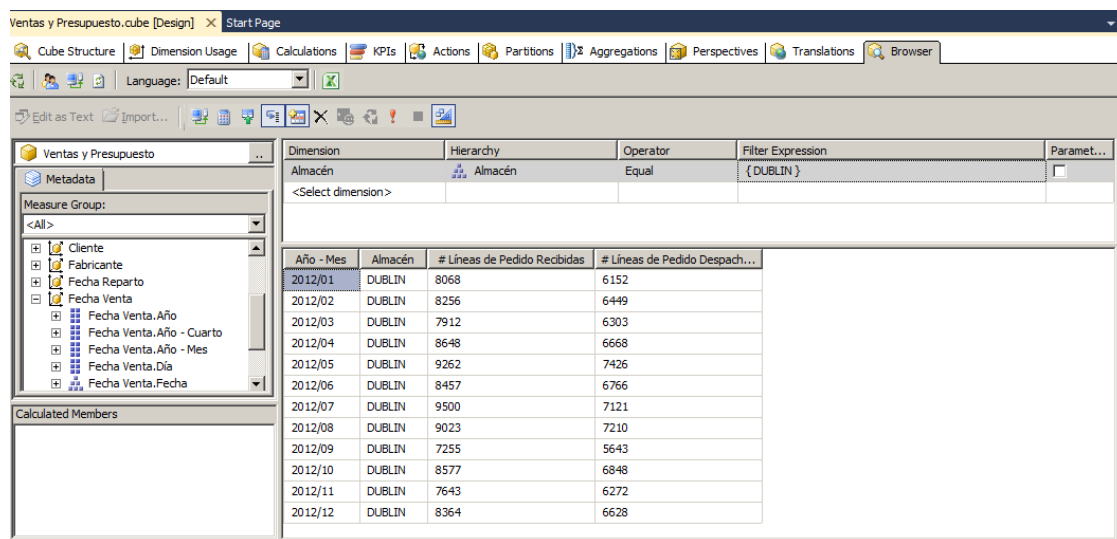


Ilustración 33. Ejemplo de informe en el browser de Analysis Service,

Diseño de Informes

En la siguiente sección se explicará cómo se realiza el acceso a Reporting Services, así como las capturas de pantallas de los informes construidos junto con una breve explicación de cada uno de ellos.

Acceso a SQL Server Reporting Services

El acceso a SQL Server Reporting Services se realiza a través de Internet Explorer, mediante la URL <http://sql2012w2k8/Reports/>. Por defecto, esta URL está configurada como página de inicio.

Una vez dentro de Reporting Services, nos encontraremos con 2 carpetas, una con los orígenes de datos llamada "Data Sources", y otra carpeta llamada "PFC Ventas RS" que es donde se almacenan los informes, tal y como podemos ver en la *Ilustración 34. Carpetas e informes dentro de Reporting Services*.

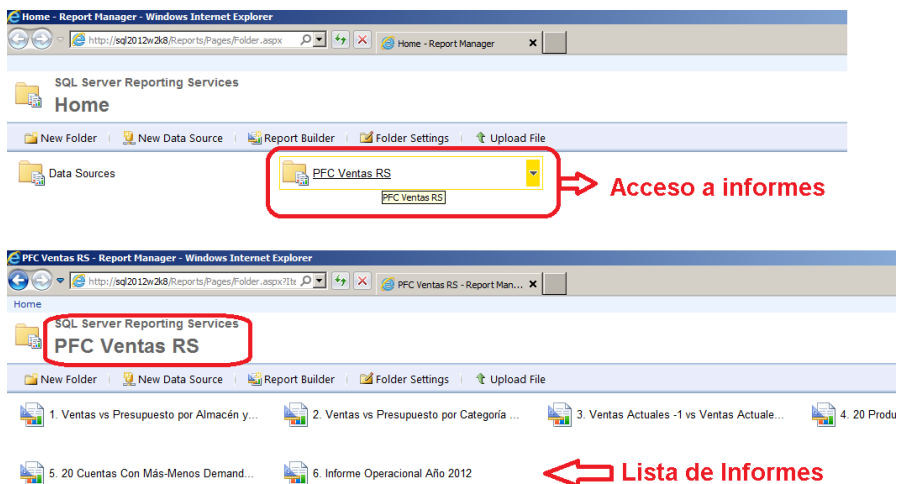


Ilustración 34. Carpetas e informes dentro de Reporting Services

Informe 01. Ventas vs Presupuesto por Almacén y Mes Venta

Este informe, que se puede ver en la *Ilustración 35. Informe 01. Ventas vs Presupuesto por Almacén y Mes Venta*, tiene como propósito analizar las ventas por almacén para determinar la variación y su porcentaje respecto al presupuesto aprobado a principios de año (que se asigna a cada almacén y categoría de producto).

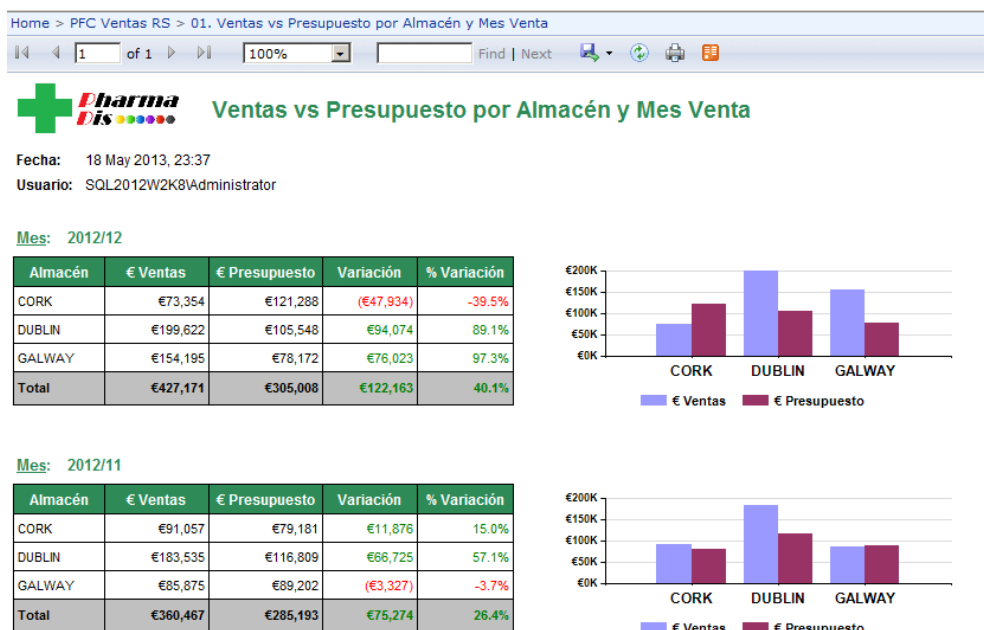


Ilustración 35. Informe 01. Ventas vs Presupuesto por Almacén y Mes Venta

Informe 02. Ventas vs Presupuesto por Categoría y Mes Venta

Este informe, que se puede ver en la *Ilustración 36. Informe 02. Ventas vs Presupuesto por Categoría y Mes Venta*, tiene como propósito analizar las ventas por categoría de producto para determinar la variación y su porcentaje respecto al presupuesto aprobado a principios de año (que se asigna a cada almacén y categoría de producto).

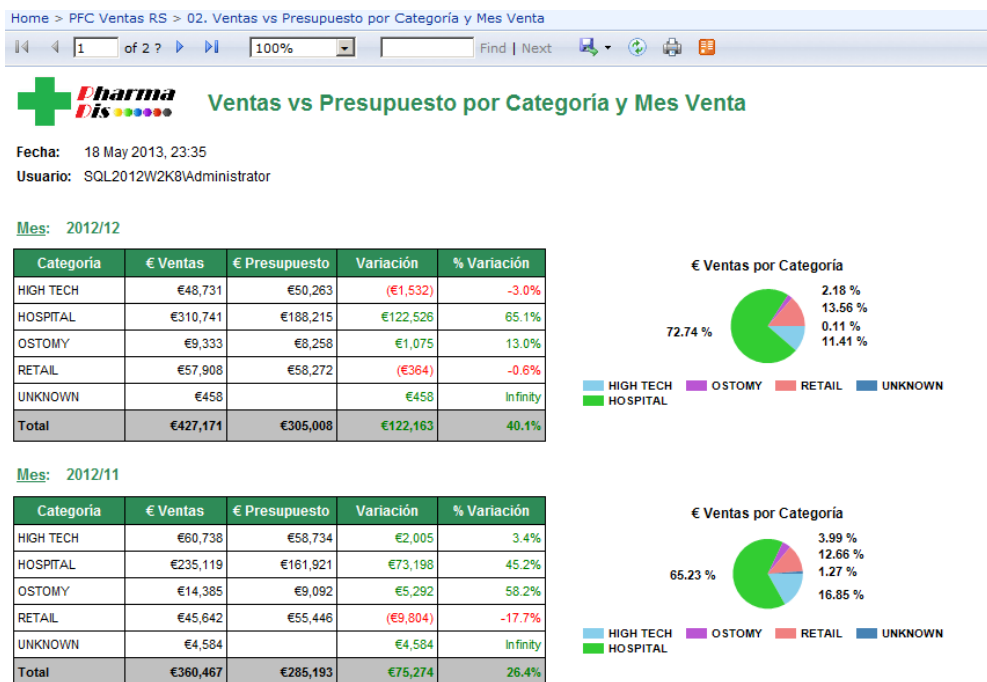


Ilustración 36. Informe 02. Ventas vs Presupuesto por Categoría y Mes Venta

Informe 03. Ventas Actuales vs Ventas Actuales -1 por Almacén y Categoría

Este informe, que se puede ver en la Ilustración 37. Informe 03. Ventas Actuales vs Ventas Actuales -1 por Almacén y Categoría (se presenta la comparativa entre los meses de Septiembre y Agosto 2012), tiene como propósito analizar las ventas por categoría de producto para determinar la variación y su porcentaje respecto al mes anterior. Este informe tiene la posibilidad de cambiar el mes actual, de forma que podamos analizar los diferentes meses del año 2012 (seleccionar el valor y hacer clic en "View Report"). Destacar que cuando se selecciona Enero 2012, al no existir datos para Diciembre 2011, no mostrará ninguna comparativa.

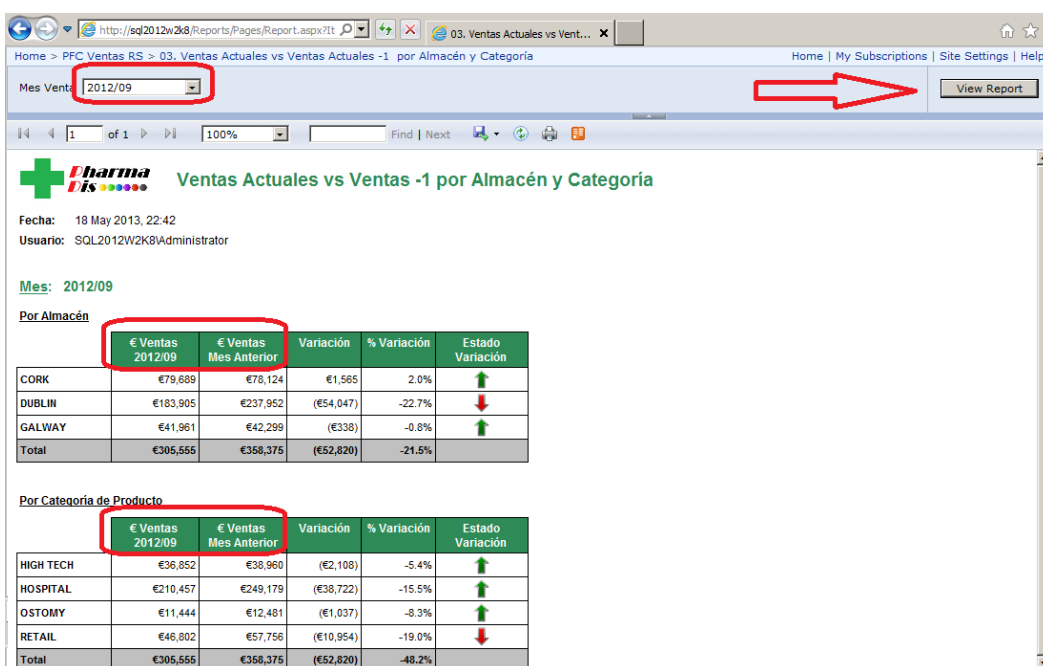
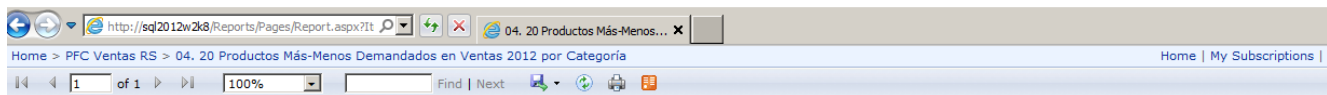


Ilustración 37. Informe 03. Ventas Actuales vs Ventas Actuales -1 por Almacén y Categoría

Informe 04. 20 productos con más/menos demanda en Ventas por Categoría

Este informe, que se puede ver en la *Ilustración 38. Informe 04. 20 productos con más/menos demanda en Ventas 2012 por Categoría* (solamente se presentan las categorías “HIGH TECH” y “HOSPITAL”), tiene como propósito analizar los productos con más / menos demanda en relación al volumen de ventas, así como su margen y porcentaje de margen, por cada una de las categorías de producto disponibles.

Vemos que la mayor parte de los productos tienen aproximadamente un 15% de margen establecido, y casos en los que el margen es o completo (del 100%, véase el tercer producto de los más demandados en HIGH TECH), o incluso un margen cero o negativo (véase los productos 3 y 10 de la lista de menos demandados en HIGH TECH).



Pharma 20 Productos Más/Menos Demandados en € Ventas

Fecha: 18 May 2013, 22:48
 Usuario: SQL2012W2K8\Administrator

Año 2012

HIGH TECH				
20 PRODUCTOS MÁS DEMANDADOS				
Producto	€ Ventas	Margen	% Margen	Rank
892117 - AVONEX PEN 30UG/0.5ML	€40,540.9	€6,081.3	15.0%	1
889905 - CIMZIA PRE FILLED SYRINGE 200MG	€33,105.9	€4,965.9	15.0%	2
880339 - REVLIMID CAPS 10MG	€30,094.1	€30,094.1	100.0%	3
891710 - NEULASTA PFS 6MG/0.6ML WITH N/GUARD	€29,088.0	€4,363.2	15.0%	4
872415 - AVONEX PRE-FILLED SYR 30MG/0.5ML	€23,908.7	€3,586.4	15.0%	5
874173 - COPAXONE PFS 20MG/ML (TEVA)	€21,960.0	€2,415.6	11.0%	6
889820 - SIMPONI 50MG PRE-FILLED (PEN)	€18,917.6	€1,832.6	9.7%	7
701287 - PULMOZYME AMPOULES 2.5MG	€17,707.1	€2,656.3	15.0%	8
887848 - FORSTEO COL INJ 250MCG/ML 2.4ML X 1	€14,443.5	€2,166.5	15.0%	9
857145 - REBIF PF SYRINGE 44MCG REBIJECT	€12,572.4	€1,885.9	15.0%	10
880340 - REVLIMID CAPS 15MG	€12,571.8	€12,571.8	100.0%	11
875153 - PUREGON CARTRIDGE 900 I.U.	€11,613.3	€1,742.0	15.0%	12
855311 - DECAPEPTYL 11.25MG 3 MONTH (IPSEN)	€11,601.4	€1,740.3	15.0%	13
886944 - REBIF MULTIDOSE CART 44MCG REBISMAR	€11,315.2	€1,697.3	15.0%	14
874178 - NEORECORMON PF SYRINGE 30 000 IU	€10,730.0	€1,609.5	15.0%	15
892361 - PROSTAP 3 DCS PFS 11.25MG	€9,774.3	€1,466.3	15.0%	16
878129 - NEXAVAR TABS 200MG	€9,524.2	€1,278.6	15.0%	17
853677 - CELLCEPT TABS 500MG	€8,337.0	€1,250.3	15.0%	18
875130 - XAGRID CAPS 0.5MG (SHIRE)	€8,058.9	€1,208.8	15.0%	19
874899 - MIMPARA TABS 30MG (AMGEN)	€7,234.7	€954.5	13.2%	20

20 PRODUCTOS MENOS DEMANDADOS				
Producto	€ Ventas	Margen	% Margen	Rank
892238 - CASOMIDE FC TABS 50MG (CLONMEL)	€45.0	€6.8	15.0%	1
888476 - TEMODAL CAPS SACHETS 5MG	€46.4	€7.0	15.0%	2
884317 - INTRON A SOLUTION FOR INJ 10 MU	€52.7	€0.0	0.0%	3
868043 - XELODA TABLETS 150MG (ROCHE)	€64.2	€9.6	15.0%	4
858948 - ZEFFIX TABS 100MG	€75.5	€11.3	15.0%	5
882919 - RETACRIT 4 000IU IN 0.4ML (HOSPIRA)	€106.5	€16.0	15.0%	6
855198 - NEORECORMON PF SYRINGE 2 000 IU	€120.6	€18.1	15.0%	7
885659 - MIRCERA PFS 30MCG/0.3ML	€142.2	€21.3	15.0%	8
882921 - RETACRIT 6 000IU IN 0.6ML (HOSPIRA)	€159.7	€24.0	15.0%	9
893707 - EPORATIO SFI PFS 2000IU/0.5ML(TEVA)	€165.2	€(23.3)	-14.1%	10
854279 - SUPREFACT INJECTION 5.5ML	€168.0	€25.2	15.0%	11
868683 - ARANESP PFS 10MCG/0.4ML	€174.7	€26.2	15.0%	12
875712 - ARANESP SURE CLICK 40MCG	€174.7	€26.2	15.0%	13
875711 - ARANESP SURE CLICK 20MCG	€174.7	€26.2	15.0%	14
872730 - GENOTROPIN MINIQUICK 0.2MG	€235.9	€35.4	15.0%	15
882018 - MIRCERA PFS 50MCG/0.3ML	€253.4	€38.0	15.0%	16
868412 - RAPAMUNE ORAL (SOLUTION 1MG/ML)	€271.1	€40.7	15.0%	17
876973 - ELIGARD INJ MONTHLY (ASTELLAS)	€277.9	€41.7	15.0%	18
890037 - MYCOLAT CAPS 250MG (ROWEX)	€302.1	€45.3	15.0%	19
870189 - VIRAFERON PEGPEN 150MCG	€313.7	€47.1	15.0%	20

HOSPITAL				
20 PRODUCTOS MÁS DEMANDADOS				
Producto	€ Ventas	Margen	% Margen	Rank
871268 - REMICADE INJECTION 100MG/ML	€192,284.8	€1,552.0	0.8%	1
868297 - SERETIDE EVOHALER 250MCG	€97,993.6	€14,459.1	14.8%	2
874808 - ALIMTA VIAL 500MG VL7623	€52,466.7	€0.1	0.0%	3
874806 - CYMBALTA CAPS 60MG	€44,103.7	€6,406.9	14.5%	4
874317 - LYRICA CAPS 100MG	€37,643.4	€5,646.2	15.0%	5
874315 - LYRICA CAPS 50MG	€35,272.6	€5,150.4	14.6%	6
871462 - CIALIS TABS 20MG	€29,129.4	€4,345.2	14.9%	7
889899 - PROLIA SOLN FOR INJ 60MG PFS	€28,972.1	€(143.2)	-0.5%	8
868229 - SYMBICORT TURBOHALER 200/6	€28,067.5	€4,078.7	14.5%	9
869991 - SPIRIVA CAPS 18MCG + HANDIHALER	€27,077.9	€3,935.9	14.5%	10
874316 - LYRICA CAPS 75MG	€26,843.9	€4,027.8	15.0%	11
872073 - SYMBICORT TURBOHALER 400/12	€22,804.9	€3,244.2	14.2%	12
871151 - LEXAPRO TABS 10MG	€22,565.9	€2,854.9	12.7%	13
880942 - ORENCIA L INJECTION 250MG/ML	€22,486.2	€190.4	0.8%	14
890467 - VERSATIS PLASTERS 5%	€21,320.0	€2,963.0	13.9%	15
874318 - LYRICA CAPS 150MG	€20,634.3	€3,097.5	15.0%	16
886726 - ISENTRESS TABS 400MG (HOSP ONLY)	€20,124.0	€(9,379.2)	-46.6%	17
855985 - SERETIDE DISKUS 500MCG	€19,576.2	€2,195.4	11.2%	18
885040 - VIDAZA VIAL 100MG	€19,050.2	€0.2	0.0%	19
874319 - CELSTRON TABS 10MG	€18,603.2	€2,469.0	13.2%	20

20 PRODUCTOS MENOS DEMANDADOS				
Producto	€ Ventas	Margen	% Margen	Rank
621585 - TAMOX TABS 10MG (ROWEX)	€0.5	€0.5	100.0%	1
857214 - GERAMOX CAPS 250MG (GERARD)	€0.5	€(0.8)	-150.9%	2
890689 - RISONTEL TABS 35MG (CLONMEL)	€1.4	€1.4	100.0%	3
884952 - MOBICAM TABS 7.5MG (PINEWOOD)	€1.5	€0.2	14.9%	4
622130 - VERISOP TABS 40MG (GERARD)	€1.6	€(0.5)	-30.2%	5
887356 - FRESUBIN CREME (PRALINE)	€1.6	€0.2	15.0%	6
879180 - ZIRTENE TABS 10MG (GERARD)	€2.2	€0.3	15.2%	7
885714 - FAMCICLOVIR TABS 125MG (TEVA)	€2.3	€(4.6)	-197.0%	8
881558 - FLUNGASIL CREAM 10MG (CLONMEL) TUBE	€2.4	€0.7	30.0%	9
860219 - TRADOL INJECTION 50MG/ML (ROWEX)	€2.5	€0.4	15.0%	10
622131 - VERISOP TABS 80MG (GERARD)	€2.5	€(0.8)	-30.7%	11
876662 - RAZOLAGER CAPS 15MG (GERARD)	€2.6	€(10.2)	-385.9%	12
622212 - VOLTAROL INJECTION 75MG/3ML(25MG/ML)	€2.8	€0.4	15.1%	13
875345 - GLUTAFIN SELECT H/FIBRE SEEDED LOAF	€2.9	€0.4	14.9%	14
601467 - CIPROXIN TABS 250MG	€3.0	€3.0	100.0%	15
871207 - ZESTAN TABS 2.5MG (CLONMEL)	€3.2	€(5.1)	-34.7%	16
894152 - IRBESAN FC TABS 75MG (ROWEX)	€3.2	€(5.1)	-157.8%	17
889783 - ENSURE PLUS BTL (PEACH)	€3.4	€0.5	14.7%	18
621210 - SLOW K TABS	€3.5	€0.5	14.9%	19
610764 - BENEICAN AMP 250MG	€3.5	€(3.2)	-91.4%	20

Ilustración 38. Informe 04. 20 productos con más/menos demanda en Ventas 2012 por Categoría

Informe 05. 20 clientes más/menos demandantes en Ventas 2012 por Categoría

Este informe, que se puede ver en la *Ilustración 39. Informe 05. 20 clientes más/menos demandantes en Ventas 2012 por Categoría* (solamente se presentan las categorías “HIGH TECH” y “HOSPITAL”), tiene como propósito analizar los clientes con más / menos demanda en relación al volumen de ventas, así como su margen y porcentaje de margen, por cada una de las categorías de producto disponibles.

Al contrario que en el informe anterior, vemos disparidad de márgenes con las cuentas de cliente. Hay casos en los que el margen es del 100% y casos en los que es negativo, quizá debido a la creación de algún crédito cuya cantidad pueda referirse al año anterior.

20 Cuentas con Más/Menos Demanda en € Ventas (2012)

Fecha: 18 May 2013, 22:53
 Usuario: SGL2012W2K0Administrator
 Año 2012

HIGH TECH

20 CUENTAS CON MÁS DEMANDA					20 CUENTAS CON MENOS DEMANDA				
Cuenta Cliente	€ Ventas	Margen	% Margen	Rank	Cuenta Cliente	€ Ventas	Margen	% Margen	Rank
11007 - GIBBS MEDICAL HALL	€19,086.8	€2,863.1	15.0%	1	1000 - BEAUFORT HOSP PHARM	€24.0	(€38.8)	-161.8%	1
11008 - FIDELITY PHARMACY LTD	€9,427.2	€6,530.1	69.3%	2	1002 - BOOTS CAWNER ETHICAL	€39.3	€5.7	15.0%	2
11009 - STRANG HOSPITAL LTD	€8,155.7	€1,223.5	15.0%	3	1006 - S. HILL CHEMIST LTD	€42.0	€6.3	15.0%	3
11010 - G. HARRINGTON HOSP	€6,817.7	€6,817.7	100.0%	4	1008 - HARRINGTON HOSP LIMITED	€43.7	€6.6	15.0%	4
11018 - THE PHARMACY COOP AND BOOTS HOSPITAL	€6,737.3	(€1,971.0)	-29.3%	5	1009 - CARBON PHARMACY LTD	€43.7	€6.6	15.0%	5
11020 - KIM BERRY LTD	€6,493.1	€974.0	15.0%	6	1010 - ST LOUIS HOSP HATHAM PHARM	€45.0	(€121.5)	-269.9%	6
11040 - ANKERS LTD	€6,363.5	€954.6	15.0%	7	1016 - ST VINCENTS PRIVATE HOSP COVENTRY ROAD	€52.7	€0.0	0.0%	7
11042 - HILL OF THE GLEN HOSP	€6,285.9	€6,285.9	100.0%	8	1000 - HULLHALL GENERAL HOSP PHARM	€57.0	€3.2	5.6%	8
11043 - HARTON HALL HOSP GENERAL LIMITED	€6,285.9	€6,285.9	100.0%	9	1000 - HULLHALL LTD	€62.4	€6.9	11.0%	9
11040 - HULL HALL LTD	€6,237.1	€935.6	15.0%	10	1004 - TUNERS PHARMACY LTD	€62.4	€6.9	11.0%	10
11040 - T. P. & S. J. GENERAL PHARM S.L.	€6,200.1	€6,046.0	97.5%	11	1005 - CARBON PHARMACY LIMITED	€62.4	€6.9	11.0%	11
11040 - HEDGECOCK PHARMACY LIMITED	€6,018.8	€6,018.8	100.0%	12	1002 - PORTFORD PHARMACY LTD	€62.4	€6.9	11.0%	12
11040 - HEDGECOCK HOSP PHARM	€6,018.8	€6,018.8	100.0%	13	1006 - HARTON HALL HEALTH CARE HALL	€62.4	€6.9	11.0%	13
11040 - HEDGECOCK PHARMACY LTD	€6,018.8	€6,018.8	100.0%	14	1002 - BOYD WILLIAMS HOSPITAL BARNHURST	€63.3	€9.5	15.0%	14
11040 - LANCASH PHARMACY	€5,724.7	€5,724.7	100.0%	15	1002 - BRISTOL HOSPITAL	€63.3	€9.5	15.0%	15
11040 - POLYCLINIC PHARMACY LTD (SUPPORT)	€5,466.8	€780.1	14.3%	16	1000 - LANE HOSP LTD	€63.3	€9.5	15.0%	16
11040 - COBBS PHARMACY	€5,301.6	€795.3	15.0%	17	1010 - COCKLANDS ACUTHE	€63.3	€9.5	15.0%	17
11040 - HILL GARDEN DISPENSARY (HULL)	€5,172.8	€775.9	15.0%	18	1000 - HULL PHARMACY	€63.3	€9.5	15.0%	18
11010 - HULL HALL LTD	€4,668.2	€700.2	15.0%	19	1000 - HULL PHARMACY LTD	€63.3	€9.5	15.0%	19
11020 - BRIDGEMAN PROPERTY CONSTRUCTION (PDR)	€4,548.6	€682.3	15.0%	20	1000 - HATHOR HOSP PHARMACY	€63.3	€9.5	15.0%	20

HOSPITAL

20 CUENTAS CON MÁS DEMANDA					20 CUENTAS CON MENOS DEMANDA				
Cuenta Cliente	€ Ventas	Margen	% Margen	Rank	Cuenta Cliente	€ Ventas	Margen	% Margen	Rank
10000 - THE PHARMACY COOP AND BOOTS HOSPITAL	€102,727.2	(€1,516.3)	-1.5%	1	1000 - BRISTOL PHARMACY LTD	€1.3	€0.2	15.3%	1
10000 - GENERAL HOSPITAL (LONDON)	€53,421.3	(€9,588.2)	-11.5%	2	1000 - BOWEN PHARMACY LTD	€1.3	€0.2	15.3%	2
10000 - GENERAL HOSP. (LONDON)	€44,990.0	(€991.5)	-2.2%	3	1000 - BOOTS CAWNER (OTC BEAUTY)	€1.7	(€0.9)	-50.3%	3
10010 - HEDGECOCK LTD	€32,873.6	(€721.6)	-2.2%	4	1000 - BOOTS CAWNER (OTC BEAUTY)	€1.7	(€0.9)	-50.3%	4
10010 - BEAUFORT HOSP PHARM	€27,913.5	(€10,910.6)	-39.1%	5	1000 - BOOTS HALF HOLLOW (OTC BEAUTY)	€1.7	(€0.9)	-50.3%	5
10016 - ST VINCENTS PRIVATE HOSP COVENTRY ROAD	€27,832.3	(€1,707.2)	-6.1%	6	1000 - BOOTS BRACKNELL (OTC BEAUTY)	€1.7	(€0.9)	-50.3%	6
10000 - TOM HADLEY HOSP. (SHEFFIELD) ETHICAL	€27,617.5	€4,126.3	14.9%	7	1000 - BOOTS BRACKNELL (OTC BEAUTY)	€1.7	(€0.9)	-50.3%	7
10000 - HULL PHARMACY LTD (SUPPORT)	€23,439.0	€1,813.8	7.7%	8	1000 - CARNE CAULY (SHEFFIELD)	€1.7	€0.3	15.4%	8
10020 - HARTON GENERAL HOSP PHARM	€19,851.3	(€15,652.7)	-78.8%	9	1000 - HILL GARDEN HOSP PHARM ST	€1.9	(€14.3)	-760.6%	9
10010 - TOM HADLEY (SHEFFIELD) LTD	€16,560.3	€2,477.2	15.0%	10	1000 - BOOTS HARTON (OTC)	€2.1	€0.1	2.4%	10
10000 - BOWEN PHARMACY LIMITED	€16,409.1	€2,333.2	14.2%	11	1000 - BOOTS LUTTERHAM (OTC BEAUTY)	€2.1	€0.1	2.4%	11
10010 - BRACKNELL HALL PHARMACY	€15,549.8	(€57.1)	-0.4%	12	1000 - BOOTS BRACKNELL (OTC)	€2.1	€0.1	2.4%	12
10000 - GALAXY CARE PHARMACY ACCOUNTS	€14,854.6	€75.4	0.5%	13	1000 - BOOTS SHEFFIELD (OTC BEAUTY)	€2.1	€0.1	2.4%	13
10000 - BOYD WILLIAMS HOSP PHARM	€14,124.4	(€45.9)	-0.3%	14	1000 - G. HILL PHARMACY (SHEFFIELD)	€2.6	€0.4	15.3%	14
10010 - GENERAL HOSP. (LONDON)	€13,481.0	(€5,369.9)	-39.8%	15	1000 - BOOTS BRACKNELL (OTC BEAUTY)	€2.9	€0.1	2.4%	15
10010 - COVENTRY PHARMACY	€12,090.0	€770.4	6.4%	16	1000 - BOOTS CARBON S.L. LTD	€2.9	€0.1	2.4%	16
10020 - BOWEN PHARMACY	€11,292.2	€395.7	3.5%	17	1000 - BOOTS BRACKNELL (OTC)	€2.9	€0.1	2.4%	17
10010 - HULL HALL (COMPANY) PHARMACY LIMITED	€10,585.4	€1,613.7	14.7%	18	1000 - BOOTS BRACKNELL (OTC)	€3.0	€0.1	2.3%	18
10010 - THORNTON (SHEFFIELD) LIMITED	€10,867.5	€1,621.3	14.9%	19	1000 - BOOTS BRACKNELL (OTC)	€3.0	€0.1	2.3%	19
10000 - BRISTOL PHARMACY LTD	€10,594.4	€1,564.0	14.8%	20	10000 - BOOTS (SHEFFIELD) OTC	€3.0	€0.1	2.3%	20

Ilustración 39. Informe 05. 20 clientes más/menos demandantes en Ventas 2012 por Categoría

Este informe en concreto permite navegar por la jerarquía de cliente, de manera que podemos analizar las cuentas con más / menos demanda para una categoría, grupo o incluso localización del cliente. Para ello se han generado una serie de prompts que el usuario podrá utilizar para seleccionar el valor/es que desee. Como ejemplo, la *Ilustración 40. Informe 05. 20 clientes más/menos demandantes en Ventas 2012 para "PHARMACY"* presenta el mismo informe generado para la categoría de cliente "PHARMACY".

Machine View Devices Help
 Home > PPC Ventas RS > 05. 20 Cuentas Con Más-Menos Demanda en Ventas 2012 por Categoría Home | My Subscriptions

Categoría: **PHARMACY** Grupo: ABBEY HEALTHCARE LTD, ABBO
 Localización: TERNE PHARMACY LTD, BOYDS

14 of 1 of 1 > > | 100% Find | Next

20 Cuentas con Más/Menos Demanda en € Ventas (2012)

Fecha: 18 May 2013, 23:00
 Usuario: SQL2012V2KBAdministrator
 Año 2012

HIGH TECH					
20 CUENTAS CON MÁS DEMANDA					
Cuenta Cliente	€ Ventas	Margen	% Margen	Rank	
12307 - LEWIS MEDICAL HALL	€19,095.8	€2,863.1	15.0%	1	
14005 - STRAND NETWORK LTD.	€9,155.7	€1,223.5	15.0%	2	
12230 - G. PARADITION HPS	€6,817.7	€6,817.7	100.0%	3	
11380 - JAMES BARRY LIMITED	€6,493.1	€974.0	15.0%	4	
12460 - AMARON LTD	€6,363.5	€954.6	15.0%	5	
16380 - MR. J.P. MC GIBBE HPS	€6,285.9	€6,285.9	100.0%	6	
16040 - MARTIN WALSH HOLDINGS LIMITED	€6,285.9	€6,285.9	100.0%	7	
14460 - ATLABAZ LIMITED	€6,237.1	€935.6	15.0%	8	
12580 - T.P. & M.J. DONNELLY H.P. S.L.	€6,200.1	€6,046.0	97.5%	9	
12190 - HORGANS PHCY (BANTLAW)	€6,018.8	€6,018.8	100.0%	10	
12640 - BOPFALLEN LATE NIGHT PHCY	€6,018.8	€6,018.8	100.0%	11	
12680 - BORGCOCKANE PHARMACY LTD.	€6,018.8	€6,018.8	100.0%	12	
14040 - LINCAGE PHARMACY	€5,724.7	€5,724.7	100.0%	13	
12440 - POLLEVEN PHARMACY LTD. (LIPPOND)	€5,466.8	€780.1	14.3%	14	
14050 - CORRS PHARMACY	€5,301.6	€795.3	15.0%	15	
12700 - MC SHERRY DISPENSERS (NLT)	€5,172.8	€775.9	15.0%	16	
12070 - WALLEN LTD	€4,668.2	€700.2	15.0%	17	
11030 - BALBRIGGAN PROPERTY CONSTRUCTION (PINK)	€4,548.6	€622.3	15.0%	18	
14070 - WALSHS PHARMACY ST. LTD.	€4,467.3	€670.1	15.0%	19	
12380 - HEDDYS PHARMACY LTD	€4,361.1	€374.3	13.2%	20	

HIGH TECH					
20 CUENTAS CON MENOS DEMANDA					
Cuenta Cliente	€ Ventas	Margen	% Margen	Rank	
13630 - BOOTS (CAHILL) ETHICAL	€38.3	€5.7	15.0%	1	
14740 - B. MAC SHEARNEY LTD	€42.0	€6.3	15.0%	2	
12760 - HERRICKS LTD	€43.7	€6.6	15.0%	3	
12780 - CARA PHARMACEUTICALS LTD.	€43.7	€6.6	15.0%	4	
14080 - BELLAMY LTD	€62.4	€6.9	11.0%	5	
12940 - CLARKE PHARMACY LTD	€62.4	€6.9	11.0%	6	
12880 - CARROLLERS PHARMACY LIMITED	€62.4	€6.9	11.0%	7	
12830 - POWFORD PHARMACY LTD.	€62.4	€6.9	11.0%	8	
14090 - WATERFORD HEALTH CARE PHCY	€62.4	€6.9	11.0%	9	
14620 - BRADY'S CROSSBEDIA	€63.3	€9.5	15.0%	10	
14010 - LURE BANTON LTD.	€63.3	€9.5	15.0%	11	
12170 - COLLINS-BRYN AGATHA	€63.3	€9.5	15.0%	12	
12090 - HOLLY PHARMACY	€63.3	€9.5	15.0%	13	
12910 - MC NALLY PHARMACEUTICALS LTD	€63.3	€9.5	15.0%	14	
12940 - BATHURST PHARMACY	€63.3	€9.5	15.0%	15	
12030 - BRADY'S BLACKROCK	€76.5	€11.5	15.0%	16	
12420 - AHS PHARMACY LTD (PINK)	€87.4	€13.1	15.0%	17	
12940 - BORGCOCKANE PHARMACY LTD	€87.4	€13.1	15.0%	18	
12840 - MARY BERRY LTD.	€87.4	€13.1	15.0%	19	
12010 - WELPHARM LTD	€87.4	€13.1	15.0%	20	

HOSPITAL					
20 CUENTAS CON MÁS DEMANDA					
Cuenta Cliente	€ Ventas	Margen	% Margen	Rank	
21040 - TOM TREACY HPSL (WATERFORD) ETHICAL	€27,617.5	€4,126.3	14.9%	1	
24090 - TOM TREACY (WATERFORD) LTD.	€16,560.3	€2,477.2	15.0%	2	
26090 - BYRNE & POWERS LTD	€16,409.1	€2,333.2	14.2%	3	
21410 - BLACKROCK HALL PHARMACY	€15,549.8	€57.1	-0.4%	4	
31700 - COSTIGANS PHARMACY	€12,090.0	€770.4	6.4%	5	
24020 - BIRNIS PHARMACY	€11,292.2	€595.7	5.3%	6	
31230 - WILLOW COMMUNITY PHARMACY LIMITED	€10,985.4	€1,615.7	14.7%	7	
31210 - THOMAS J BROSQUAN LIMITED	€10,867.5	€1,621.3	14.9%	8	
12080 - ORELLY'S PHARMACY LTD.	€10,594.4	€1,564.0	14.8%	9	
12460 - BRIDGEMOOR (SULLIVAN)	€10,495.0	€1,476.6	14.1%	10	
31380 - JOHN O BROWNE LTD.	€9,993.7	€1,503.8	15.0%	11	
12080 - MRS YOUNG MORRISON HPS	€9,708.8	€669.5	6.9%	12	
13003 - MAPLE PHARMACY LTD	€9,672.5	€311.9	3.2%	13	
31400 - NORTHFACE LTD.	€9,482.8	€1,298.5	13.7%	14	
12000 - ABBEY HEALTHCARE LTD.	€9,454.5	€1,324.0	14.0%	15	
31130 - F.J. & M. BOLAND LTD. (PINK)	€9,420.8	€1,372.0	14.6%	16	
21100 - STANCO PHARMACY (SANDWICH)	€9,283.0	€625.7	-8.9%	17	
24090 - WATERFORD HEALTH CARE PHCY	€9,843.7	€110.7	-1.3%	18	
10700 - BLACKFORD BRIDGE PHARMACY LTD.	€8,793.5	€6,552.4	-74.5%	19	
12720 - CLARKE'S PHARMACY	€8,782.6	€1,198.5	13.6%	20	

HOSPITAL					
20 CUENTAS CON MENOS DEMANDA					
Cuenta Cliente	€ Ventas	Margen	% Margen	Rank	
13002 - GOLDSON PHARMACY LTD	€1.3	€0.2	15.3%	1	
12080 - SINGH'S PHARMACY LTD	€1.3	€0.2	15.3%	2	
12140 - BOOTS (CAHILL) OPT BEAUTY	€1.7	€0.9	-50.3%	3	
12110 - BOOTS (CAHILL) OPT BEAUTY	€1.7	€0.9	-50.3%	4	
12100 - BOOTS (WALTON ST) OPT BEAUTY	€1.7	€0.9	-50.3%	5	
12140 - BOOTS (WATERFORD) OPT BEAUTY	€1.7	€0.9	-50.3%	6	
12110 - BOOTS (WATERFORD) OPT BEAUTY	€1.7	€0.9	-50.3%	7	
12090 - SAMPSON GALLERY (SAMPSON)	€1.7	€0.3	15.4%	8	
12070 - BOOTS (WATERFORD) OPT	€2.1	€0.1	2.4%	9	
12140 - BOOTS (WATERFORD) OPT BEAUTY	€2.1	€0.1	2.4%	10	
12080 - BOOTS (WATERFORD) OPT	€2.1	€0.1	2.4%	11	
12110 - BOOTS (WATERFORD) OPT BEAUTY	€2.1	€0.1	2.4%	12	
12110 - G. SHEAR PHARMACY (SHEAR)	€2.6	€0.4	15.3%	13	
12114 - BOOTS (BLACKROCK) OPT BEAUTY	€2.9	€0.1	2.4%	14	
12090 - BOOTS (SHEAR) S.C.J OPT	€2.9	€0.1	2.4%	15	
12080 - BOOTS (BLACKROCK) OPT	€2.9	€0.1	2.4%	16	
12070 - BOOTS (WATERFORD) OPT	€3.0	€0.1	2.3%	17	
12080 - BOOTS (WATERFORD) OPT	€3.0	€0.1	2.3%	18	
12070 - BOOTS (WATERFORD) OPT BEAUTY	€3.0	€0.1	2.3%	19	
12070 - BOOTS (WATERFORD) OPT BEAUTY	€3.0	€0.1	2.3%	20	

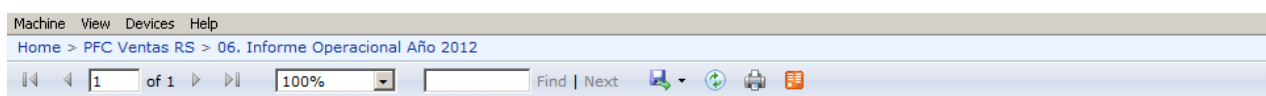
Ilustración 40. Informe 05. 20 clientes más/menos demandantes en Ventas 2012 para "PHARMACY"

Informe 06. Informe Operacional por Año de Venta para el año 2012

Este informe, que se puede ver en la *Ilustración 41. Informe 06. Informe Operacional por Año de Venta para el año 2012*, tiene como propósito analizar las operaciones de la empresa para el año 2012, de forma que puedan saber el número de líneas y unidades que han recibido, despachado y no despachado, así como el porcentaje de ventas perdidas y el número total de órdenes procesadas. Se presenta la información de forma tabular por almacén y tipo de orden, así como dos gráficas que muestran la tendencia de líneas recibidas / despachadas a lo largo de los meses para los 3 almacenes.

En la primera tabla (por almacén) vemos que aproximadamente no se despacha el 25% de las líneas que se reciben, lo que nos indica a preguntarnos el por qué estas líneas estaban fuera de stock y cómo podríamos reducir este porcentaje.

En la segunda tabla (por tipo de orden), vemos que hemos procesado unos 500 créditos que nos ha proporcionado un retorno al almacén de cerca de unas 4300 unidades.



Fecha: 18 May 2013, 23:05
 Usuario: SQL2012W2K8Administrator

Año 2012

Por Almacén

Almacén	# Órdenes	# Líneas Recibidas	# Líneas Despachadas	# Líneas No Despachadas	% Perdidas	# Unidades Pedidas	# Unidades Despachadas	# Unidades No Despachadas	% Perdidas
CORK	3,060	43,904	30,927	12,977	29.6%	127,875	90,209	37,666	29.5%
DUBLIN	11,157	100,964	79,486	21,478	21.3%	310,168	234,557	75,611	24.4%
GALWAY	1,847	36,420	27,013	9,407	25.8%	103,134	78,515	24,619	23.9%
Total	16,064	181,288	137,426	43,862	25.5%	541,177	403,281	137,896	24.2%

Por Tipo de Orden

Tipo de Orden	# Órdenes	# Líneas Recibidas	# Líneas Despachadas	# Líneas No Despachadas	% Perdidas	# Unidades Pedidas	# Unidades Despachadas	# Unidades No Despachadas	% Perdidas
CREDIT	496	0	0	0	NaN	(4,286)	(4,286)	0	0.0%
INVOICE	15,568	181,288	137,426	43,862	24.2%	545,463	407,567	137,896	25.3%
Total	16,064	181,288	137,426	43,862	25.5%	541,177	403,281	137,896	24.2%

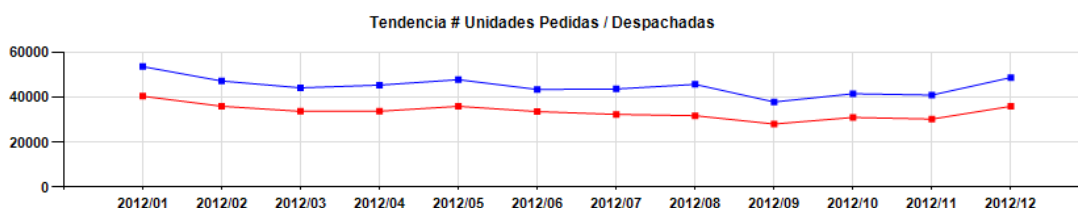
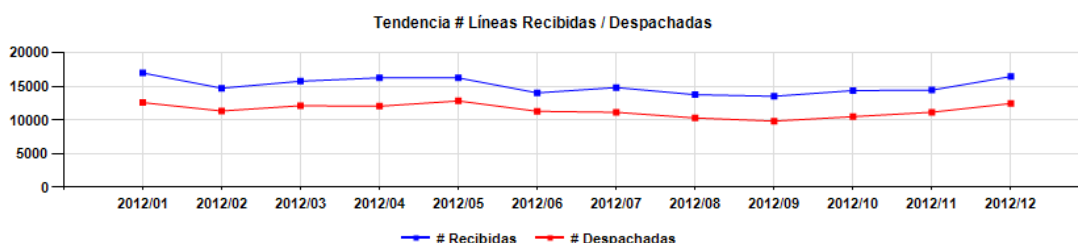


Ilustración 41. Informe 06. Informe Operacional por Año de Venta para el año 2012

Informe 07. 20 Rutas con Más/Menos Beneficios Año 2012

Este informe, que se puede ver en la *Ilustración 42. Informe 07. 20 rutas con más/menos beneficio Año 2012*, tiene como propósito analizar las rutas más / menos rentables para la empresa, de forma que puedan entender si han acumulado pérdidas y analizar la posibilidad de asignar más furgonetas a las más rentables, o cerrar las rutas con más pérdidas.

Vemos que existen una serie de rutas que proporcionan una gran cantidad de pérdidas, por lo que conviene preguntar por qué el margen es negativo y si conviene mantener activas dichas rutas.

20 RUTAS CON MÁS BENEFICIO				
Ruta	€ Ventas	Margen	% Margen	Rank
16 CORK PAL 2 AM (38688)	€40,599.8	€9,128.7	22.5%	1
GALWAY CITY (38911)	€18,479.6	€8,288.3	44.9%	2
SWIRIOS (480)	€27,569.7	€7,994.7	29.0%	3
DAY NORTH CITY SPLIT (865)	€29,586.9	€7,932.8	26.8%	4
RING OF HERRY (107 HERRY) (38916)	€55,970.4	€7,333.6	13.1%	5
SATURDAY WAINSWORTH 10 (832)	€6,817.7	€6,817.7	100.0%	6
AIRPORT 2 (380)	€15,394.0	€6,699.5	43.5%	7
SANTRYCLOUTAMP 11 (811)	€17,848.0	€6,544.4	36.7%	8
FORTLACIDE FREDDER 12 (846)	€52,280.3	€6,340.5	12.1%	9
SLIGO (740)	€12,019.3	€6,294.7	52.4%	10
EAST CORK PAL 5 AM (38912)	€47,511.1	€6,233.7	13.1%	11
DINGLE (910)	€31,017.8	€5,862.5	18.9%	12
CORK AGE PAL 3 AM (38911)	€56,426.0	€5,621.2	10.0%	13
DROGHEDA BALRIGG 14 (38930)	€46,485.3	€5,215.2	11.2%	14
BAMP 8 (380)	€23,250.0	€5,212.3	22.4%	15
DUBLIN SANDYFORD 9 (340)	€40,287.0	€4,530.5	11.2%	16
CORK 5 CITY PAL 1 AM (38914)	€56,224.5	€4,377.3	7.8%	17
DINGLE (38910)	€43,113.2	€4,180.6	9.7%	18
BAYLISK RUN 1 (38920)	€30,686.6	€4,004.3	13.0%	19
PERMOY (38920)	€31,373.4	€3,916.5	12.5%	20

20 RUTAS CON MENOS BENEFICIO				
Ruta	€ Ventas	Margen	% Margen	Rank
NORTH CITY HOSP 14 (887)	€50,628.0	(€14,149.2)	-27.9%	1
BRALMOUNT HOSP 16 (811)	€35,609.8	(€11,480.4)	-32.2%	2
TERENURE TALLAGHT 12 (890)	€46,893.3	(€10,355.5)	-22.1%	3
UNIONDAUN (-)	€33,863.4	(€9,830.5)	-29.0%	4
SOUTH CITY HOSP 13 (886)	€43,362.5	(€8,147.9)	-18.8%	5
DROGHEDA (238)	€58,377.0	(€5,732.8)	-9.8%	6
SOUTH HOSPITAL 2 (286)	€36,233.5	(€5,023.1)	-13.9%	7
DUNDALK DROGHEDA 13 (807)	€8,870.2	(€4,664.1)	-52.6%	8
TIPPERARY (38938)	€24,641.8	(€4,591.2)	-18.6%	9
TALLAHORE NAAS 10 (340)	€32,440.7	(€4,491.7)	-13.8%	10
SOUTH LIMERICK CITY (38925)	€62,116.0	(€3,947.2)	-6.4%	11
HOSPITALS TALLAHORE/MALLINGAR (240)	€12,410.5	(€3,906.4)	-31.5%	12
WIDLOW 7 (838)	€47,406.1	(€3,584.2)	-7.6%	13
EAST LIMERICK CITY (920)	€20,746.0	(€3,312.6)	-16.0%	14
SANDYFORD PM (380)	€26,029.6	(€2,489.3)	-9.6%	15
ENNIS RUN 1 (910)	€5,920.4	(€2,488.0)	-42.0%	16
NAAS (401)	€21,675.9	(€2,294.0)	-10.6%	17
BLANCHARDSTOWN (480)	€41,360.3	(€2,074.4)	-5.0%	18
SAT LUX/SERRAS (320)	€4,917.9	(€1,825.3)	-37.1%	19
CORK AGE PAL 3 AM (910)	€4,874.2	(€1,788.3)	-36.7%	20

Ilustración 42. Informe 07. 20 rutas con más/menos beneficio Año 2012

Informe 08. Ventas y Margen por Región Año 2012

Este informe, que se puede ver en la *Ilustración 43. Informe 08. Ventas y Margen por Región.*, tiene como propósito analizar las regiones de Irlanda en las que más ventas se producen, así como entender el margen de estas.

En este informe podemos ver que hay ciertas regiones que se podrían agrupar. Por ejemplo, DUBLIN 5, DUBLIN 9 y DUBLIN 24 se podrían agrupar bajo una región DUBLIN, o CORK CITY y CORK COUNTY N bajo el condado CORK. Este proceso de categorización no se ha realizado como parte de este proyecto, pero se identifica como una posible mejora (y casi podríamos decir necesidad) para el análisis de los datos de este almacén de datos.

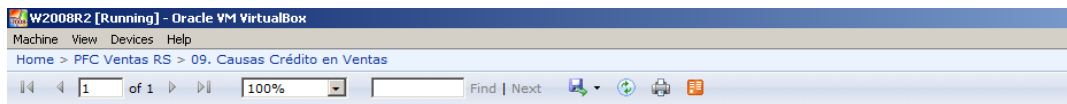
Región	€ Ventas	€ Margen	Unid. Vendidas	Líneas Vendidas
Cataluña	€425,763	€19,539	26,509	7,945
Comunidad Valenciana	€279,933	€27,642	21,403	8,497
Madrid	€179,946	(€8,836)	9,183	3,135
País Vasco	€165,237	€15,997	13,932	3,800
Castilla-La Mancha	€161,737	€14,257	13,729	4,723
Extremadura	€158,547	€8,281	13,056	4,499
Castilla y León	€145,546	€29,953	10,287	4,116
Galicia	€138,924	€2,519	11,558	3,450
Comunidad de Madrid	€136,711	€11,203	10,915	3,848
Comunidad de Castilla y León	€136,365	€18,526	11,607	4,594
Valencia	€133,383	€11,215	11,339	4,490
Extremadura	€131,894	€7,128	10,269	3,146
La Rioja	€127,494	€5,652	13,175	5,147
Saragoza	€107,154	€5,102	5,724	1,708
La Rioja	€105,415	€12,777	9,218	3,618
Comunidad de Madrid	€95,182	(€1,215)	7,312	2,448
País Vasco	€94,471	€8,026	9,012	3,850
Castilla	€88,628	€11,758	6,962	3,033
País Vasco	€87,583	€14,001	6,486	2,205
Valencia	€85,430	€4,166	8,629	3,311
La Rioja	€84,219	(€11,892)	23,723	5,398
País Vasco	€83,489	€4,104	8,359	3,435
Castilla	€81,068	(€10,456)	4,542	1,257
Castilla	€70,437	(€7,800)	6,583	2,501
Valencia	€68,321	(€177)	7,378	2,481
País Vasco	€65,986	€6,592	5,893	2,254
Castilla	€64,324	(€3,188)	4,673	1,244

Ilustración 43. Informe 08. Ventas y Margen por Región.

Informe 09. Causas Crédito en Ventas 2012

Este informe, que se puede ver en la Ilustración 44. Informe 09. Causas Crédito en Ventas, tiene como propósito analizar las diferentes causas por las que se ha concedido un crédito, así como analizar los montantes de créditos por categoría de producto, almacén y categoría de cliente para entender dónde residen la mayor parte de los créditos que concede la empresa.

En este informe llama la atención que, aunque la categoría de producto **HOSPITAL** tenga más de €40,000 en créditos, la categoría de cliente que más créditos tenga es **PHARMACY**. Esto es debido a que existen productos categorizados como **HOSPITAL** que se venden en farmacias, como por ejemplo geles, gasas, cremas, etc. Y si observamos la causa de crédito más común (**ORDERED BY ERROR (RTS)**), podemos entender que las farmacias son más propensas a equivocarse en sus pedidos, por lo que es posible que PharmaDis necesite cobrar a nuestros clientes el servicio de distribución de productos que por esta razón tienen que ser devueltos al almacén.



Causas de Crédito en Ventas

Fecha: 18 May 2013, 23:31
 Usuario: SQL2012W2K8Administrator

Año 2012

Causa Crédito	€ Créditos	Unid. Devueltas
ORDERED IN ERROR (RTS)	(€30,802)	(3,037)
PICKING ERROR	(€5,257)	(314)
TELESALES ERROR	(€4,571)	(343)
CASELINE CLEANED REPORT	(€3,535)	0
WRONG PRICING (RE-INV)	(€3,267)	0
NOT DELIVERED	(€2,980)	(3)
DUPLICATE ERROR	(€1,972)	(167)
FRIDGE LINE NOT DELIVERED	(€1,726)	0
AUTHORISED LEVY WAIVER SM	(€801)	(47)
SHORT DELIVERED	(€744)	(9)
DUPLICATE ORDER	(€686)	(10)
CHARGED TO WRONG ACCOUNT	(€587)	(28)
FICI 15% DUE (NRTS)	(€531)	0
MISSING RETURN/ QUERIES	(€494)	0
CRUSHED	(€457)	(68)
IPHA RETURNS	(€454)	(64)
CREDIT LEVY	(€297)	(25)
DISCOUNT	(€246)	(2)
ORDER NOT DELIVERED	(€205)	(2)
BOTTLE DAMAGE	(€182)	(34)
CREDIT & RE INVOICE	(€147)	0
STOCK NOT FOUND	(€145)	(38)
REPS ERROR	(€133)	(43)
OUTSTANDING QUERIES (NRTS)	(€126)	0
GOODS RECD SHORT DATED	(€121)	(1)
DAMAGED/FAULTY (NRTS)	(€110)	(28)
PLC CREDIT/RECHARGE NRTS	(€85)	0
FRIDGE DAMAGED IN TRANSIT	(€82)	(16)
PRICING ERROR	(€78)	0
GOODS RECEIVED DAMAGED	(€34)	0
SHORT DATED	(€18)	0
UNKNOWN	(€10)	(1)
PL OOS IN WAREHOUSE	(€5)	(6)
DAMAGED	(€3)	0

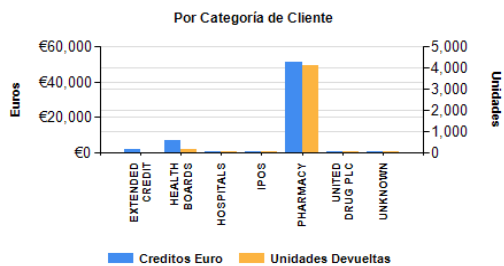
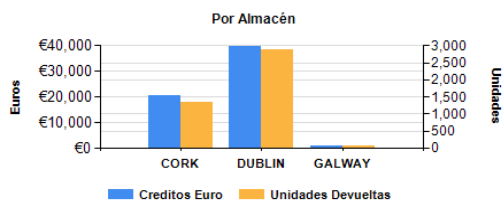
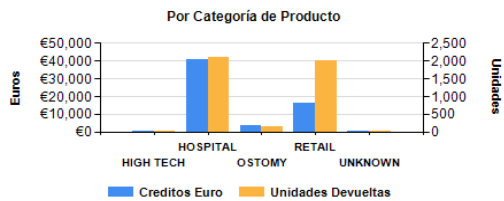


Ilustración 44. Informe 09. Causas Crédito en Ventas

Informe 10. Ventas y Margen por Distribuidor y Fabricante

Este informe, que se puede ver en la Ilustración 45. Informe 10. Ventas y Margen por Distribuidor y Fabricante, tiene como propósito analizar las ventas y margen por distribuidores (que se muestra mediante gráficas) y de los fabricantes (listados). En las gráficas se agrupan bajo la categoría **OTHER** todos aquellos distribuidores con ventas y margen inferiores al 5% del total respectivamente.

Vemos que cerca del 75% ventas proceden de productos distribuidos por **UNITED DRUG DISTRIBUTORS**, del cual curiosamente se obtiene un % Margen notablemente inferior a **ALLPHAR SERVICES**, que aproximadamente distribuye productos que copan un 15% de las ventas.

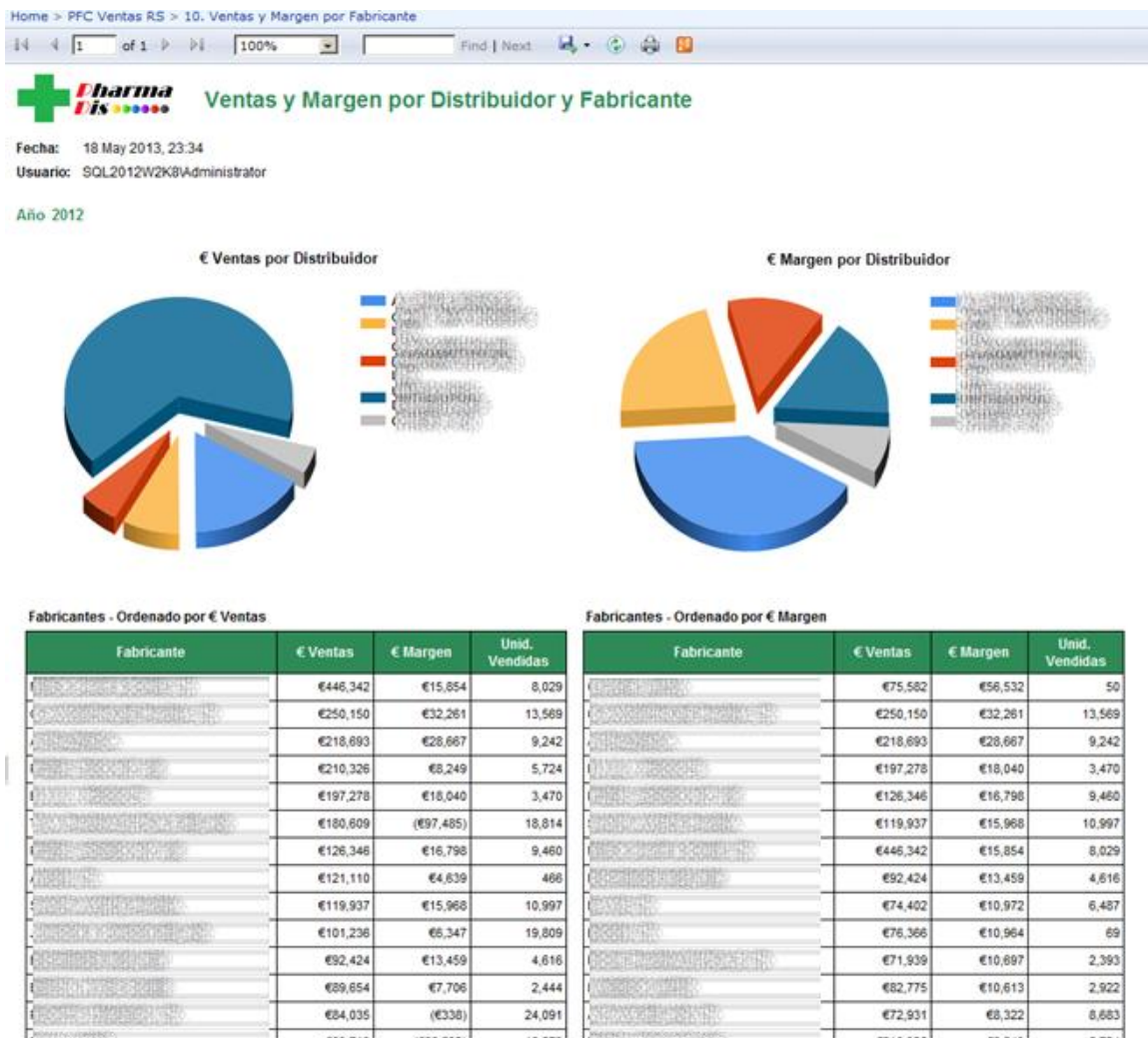


Ilustración 45. Informe 10. Ventas y Margen por Distribuidor y Fabricante

4. Conclusiones

Una vez terminado el proyecto, se puede afirmar que el grado de cumplimiento de los objetivos hasta la fecha de entrega de la memoria ha sido del 100%, además de extender satisfactoriamente los requisitos preestablecidos al inicio del proyecto mediante la creación de nuevas medidas y nuevos informes. Además, en relación a la calidad del producto, podemos decir que responde perfectamente a todas las especificaciones que el cliente ha dado.

Por otro lado, también es importante destacar que este trabajo ha servido para adquirir un gran conocimiento tanto técnico como práctico de la herramienta Microsoft SQL Server 2012 y de su plataforma Business Intelligence, ya que está considerado como uno de los productos más importantes en el mercado.

En relación a la implementación del proyecto, una de las partes más costosa de todas ha sido la del aprendizaje de la plataforma, ya que ha requerido más tiempo de lo esperado, especialmente en la creación del cubo, si bien las partes que más tiempo me han llevado han sido el proceso de carga de datos y el desarrollo de los informes, esta última especialmente al aplicar detalles cosméticos.

Como conclusión final, decir que ha sido un proyecto muy interesante y que puede abrir las puertas a futuros proyectos profesionales que se realicen con dicha plataforma.

5. Líneas de Evolución Futura

Como futuras evoluciones del sistema, podríamos considerar las siguientes:

- Añadir datos de años anteriores al 2012 para poder realizar comparaciones “*Year over Year*”. De esta forma podríamos analizar tendencias para algunas de las medidas presentadas, e incluso podríamos aplicar técnicas de *data mining* para obtener conocimiento.
- Realización de nuevas tablas de hechos con datos sobre *compras*, que permitan cruzar datos con, por ejemplo, producto y distribuidora, o *deuda*, que permitan cruzar datos con cliente y fecha de venta. De esta manera, nuestro almacén de datos corporativos podría analizar procesos de negocio de forma conjunta.
- Mejoras en la base de datos:
 - o Particionamiento de las tablas de hechos dependiendo del volumen de datos de estas para asegurar un mejor mantenimiento y rendimiento tanto en la carga de datos como en las consultas.
- Mejoras en el cubo de datos:
 - o Incremento en el rendimiento mediante la creación de particiones de datos por año de venta en los grupos de medidas, o creación de cálculos agregados, lo que haría que las consultas fuesen aún más rápidas.
 - o Creación de vistas de datos y asignación a grupos de usuarios que no puedan ver ciertos datos. Por ejemplo, los datos de un almacén no deberían de ser vistos por usuarios de otro almacén. También podría darse el caso que ciertos fabricantes deseen obtener los datos de ventas, por lo que habría que restringir el conjunto de datos dependiendo del usuario que acceda.
- Mejoras en los informes:
 - o Mayor interactividad con los usuarios. Por ejemplo, informes tipo *dashboard* donde el usuario cambia valores de una variable y todo el informe se actualiza para dicho valor.
 - o Realización de tarjetas de puntaje.
 - o Añadir mecanismos de distribución de informes vía email.
 - o Implementar un mecanismo de seguridad para que conjuntos de informes no sean vistos por ciertos grupos de usuarios.
- Integración con SharePoint mediante la herramienta y servicios de *PerformancePoint* para la generación de *dashboards* y analíticas corporativas. De esta manera, podríamos publicar datos en la Intranet de la empresa para todos los empleados. Ver el siguiente enlace para más información: <http://blogs.msdn.com/b/performancepoint/archive/2012/08/03/what-s-new-in-performancepoint-services-2013.aspx>

6. Glosario

Almacén de datos (*Data Warehouse*) Colección de datos orientada a un determinado ámbito, integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

Base de Datos Multidimensional. Tipo de base de datos orientada a OLAP que almacena los datos de dimensiones y medidas en vectores.

Dimensión. Entidad que se vincula a una tabla de hecho y que contiene atributos descriptivos.

ETL (*Extract, Transform and Load*). Proceso de extracción, transformación y carga.

Grano. Representa el nivel más atómico que permite definir los hechos.

Hecho. Actividad objeto de análisis.

HOLAP (*Hybrid OLAP*). Tipo de arquitectura OLAP resultante de la combinación de MOLAP y ROLAP.

Medida. Es la información relevante sobre el hecho y que se almacena en la tabla de hechos.

MOLAP (*Multidimensional OLAP*). Tipo de arquitectura OLAP en la que el cubo de datos se almacena en una base de datos multidimensional.

OLAP (*Online Analytical Processing*). Procesamiento analítico en línea.

ROLAP (*Relational OLAP*). Tipo de arquitectura OLAP que accede a los datos en la base de datos relacional mediante consultas SQL y no genera ningún cubo.

Tabla de Hechos. Tabla central en un esquema dimensional que contiene los hechos.

7. Bibliografía

1. **Kimball, R. y Ross, M.** *The Data Warehouse Toolkit*. s.l. : Wiley Publishing, Inc.
2. **Mundy, J. y Thornthwaite, W.** *The Microsoft Data Warehouse Toolkit*. s.l. : Wiley Publishing, Inc.
3. **Caserta, J. y Kimball, R.** *The Data Warehouse ETL Toolkit*. s.l. : Wiley Publishing, Inc.
4. **Wikipedia.** *Microsoft SQL Server*. [En línea] http://en.wikipedia.org/wiki/Microsoft_SQL_Server.
5. **Microsoft.** *Microsoft SQL Server*. [En línea] <http://www.microsoft.com/en-us/sqlserver/default.aspx>.
6. —. *SQL Server Forums*. [En línea] <http://social.msdn.microsoft.com/Forums/en-US/category/sqlserver>.

8. Apéndice A – Código SQL

Script SQL de creación de la base de datos

```

/*****
*
* Creación de la base de datos "dw_ventas"
*
*****/

CREATE DATABASE dw_ventas
ON PRIMARY -- Filegroup primario
( NAME = FD_PRIMARY,
  FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL11.BIPFC\MSSQL\DATA\fd_primary.mdf',
  SIZE = 500MB,
  MAXSIZE = 1024MB,
  FILEGROWTH = 10MB),

FILEGROUP FG_VENTAS
( NAME = FD_VENTAS,
  FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL11.BIPFC\MSSQL\DATA\fd_ventas.ndf',
  SIZE = 500MB,
  MAXSIZE = 2048MB,
  FILEGROWTH = 100MB)
LOG ON
( NAME = FD_LOG,
  FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL11.BIPFC\MSSQL\DATA\fd_log.ldf',
  SIZE = 500MB,
  MAXSIZE = 1024MB,
  FILEGROWTH = 50MB)
COLLATE SQL_Latin1_General_CP1_CI_AS
go

/*****
*
* Cambiar el modo de recuperación a Simple (no queremos mantener log de transacciones)
*
*****/

ALTER DATABASE dw_ventas SET RECOVERY SIMPLE
go

```

Script SQL de creación de los esquemas

```

/*****
*
* Creación de los esquemas necesarios en la base de datos "dw_ventas"
*
*****/

use dw_ventas
go
create schema STG
go
create schema DIM
go
create schema FACT
go

```

Script SQL de creación de la cuenta de servicio "dw_etl"

```

/*****
***
*
* Creación de la cuenta de servicio para carga de datos
*
*****/

use dw_ventas;
go

create login dw_etl with password = 'PFCVentas2013';
create user dw_etl for login dw_etl;

/*****
***
*
* Asignar permisos en esquemas a la cuenta de servicio para carga de datos
* Crear rol de base de datos para asignar dichos permisos
*
*****/

create role rol_dw_etl;

grant select, insert, update, delete, execute, alter          on schema :: DIM to rol_dw_etl;
grant select, insert, update, delete, execute, alter          on schema :: STG to rol_dw_etl
grant select, insert, update, delete, execute, alter          on schema :: FACT to rol_dw_etl;
go

-- Asignar rol a cuenta de servicio de carga de datos

```

```
exec sp_addrolemember 'rol_dw_etl', 'dw_etl';
```

Script SQL de creación de la cuenta de servicio “dw_consulta”

```

/*****
***
*
* Creación de la cuenta de servicio para consulta de datos (cubo de datos e informes)
*
*****/
use dw_ventas;
go

create login dw_consulta with password = 'PFCVentasConsulta2013';
create user dw_consulta for login dw_consulta;

/*****
***
*
* Asignar permisos en esquemas a la cuenta de servicio para consulta de datos
* Crear rol de base de datos para asignar dichos permisos
*
*****/
create role rol_dw_consulta;

grant select on schema :: DIM to rol_dw_consulta;
grant select on schema :: FACT to rol_dw_consulta;
go

-- Asignar rol a cuenta de servicio de consulta de datos
exec sp_addrolemember 'rol_dw_consulta', 'dw_consulta';

```

Script SQL de creación de tablas de control

```

/*****
***
*
* Creación de una tabla de control para cargar los nombres de los ficheros de datos
*
*****/
use dw_ventas
go

CREATE TABLE STG.CTL_FICHEROS (
    NOMBRE_FICHERO          varchar(100)          not null,

```

```

        FECHA_CARGA            datetime            not null
    )
on FG_VENTAS
go

```

Script SQL de creación de tablas temporales

```

/*****
***
*
* Creación de una tabla temporal para almacenar los datos de las líneas de pedido
*
*****
**/
use dw_ventas
go

CREATE TABLE STG.STG_LINEA_DE_PEDIDO (
    CODIGO_ALMACEN            nvarchar(5)  not null,
    FECHA_DE_REPARTO         date          null,
    FECHA_DE_VENTA           date          null,
    NUMERO_ORDEN             int           not null,
    NUMERO_FACTURA          int           null,
    NUMERO_LINEA             int           not null,
    NUMERO_CUENTA            nvarchar(10) null,
    CODIGO_REGION            smallint     null,
    CODIGO_TIPO_ORDEN        nvarchar(10) null,
    CODIGO_RUTA              smallint     null,
    CODIGO_ESTADO_LINEA     smallint     null,
    CODIGO_CAUSA_CREDITO    smallint     null,
    CODIGO_PRODUCTO         int           null,
    CODIGO_FABRICANTE        nvarchar(10) null,
    MONTANTE_LINEA_PEDIDO_EUR money       null,
    CANTIDAD_ORDENADA       int           null,
    CANTIDAD_DESPACHADA     int           null,
    constraint PK_STG_LINEA_DE_PEDIDO primary key
        (CODIGO_ALMACEN, NUMERO_ORDEN, NUMERO_LINEA) on FG_VENTAS
) ON FG_VENTAS
go

```

Script SQL de creación de dimensiones

```

use dw_ventas
go

/*=====*/
/* Tabla: DIM_CATEGORIA_PRODUCTO */
/*=====*/

create table DIM.DIM_CATEGORIA_PRODUCTO (
    ID_CATEGORIA_PRODUCTO      int                identity,
    CODIGO_CATEGORIA_PRODUCTO  nvarchar(5)        not null,
    NOMBRE_CATEGORIA_PRODUCTO  nvarchar(20)       not null,
    constraint PK_DIM_CATEGORIA_PRODUCTO primary key (ID_CATEGORIA_PRODUCTO) on FG_VENTAS
)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
SET IDENTITY_INSERT DIM.DIM_CATEGORIA_PRODUCTO ON
go
INSERT INTO DIM.DIM_CATEGORIA_PRODUCTO
(ID_CATEGORIA_PRODUCTO, CODIGO_CATEGORIA_PRODUCTO, NOMBRE_CATEGORIA_PRODUCTO)
VALUES (-1, 'UNK', 'UNKNOWN');
go
SET IDENTITY_INSERT DIM.DIM_CATEGORIA_PRODUCTO OFF
go
DBCC CHECKIDENT('DIM.DIM_CATEGORIA_PRODUCTO', RESEED, 0)
go

/*=====*/
/* Tabla: DIM_FECHA */
/*=====*/

create table DIM.DIM_FECHA (
    ID_FECHA      int                not null,
    FECHA         datetime           not null,
    MES           nvarchar(10)       not null,
    CUARTO        nvarchar(10)       not null,
    ANO           nvarchar(10)       not null,
    constraint PK_DIM_FECHA primary key (ID_FECHA) on FG_VENTAS
)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
INSERT INTO DIM.DIM_FECHA VALUES (19000101, '1900-01-01', '1900/01', '1900/Q1', '1900')

/*=====*/
/* Tabla: DIM_ALMACEN */

```



```

/*=====*/
create table DIM.DIM_ALMACEN (
    ID_ALMACEN          int          identity,
    CODIGO_ALMACEN      nvarchar(5)  not null,
    NOMBRE_ALMACEN      nvarchar(10) not null,
    constraint PK_DIM_ALMACEN primary key (ID_ALMACEN) on FG_VENTAS
)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
SET IDENTITY_INSERT DIM.DIM_ALMACEN ON
go
INSERT INTO DIM.DIM_ALMACEN (ID_ALMACEN, CODIGO_ALMACEN, NOMBRE_ALMACEN) VALUES (-1, 'UNKN',
'UNKNOWN')
go
SET IDENTITY_INSERT DIM.DIM_ALMACEN OFF
go

-- Insertar registros para los diferentes almacenes. Carga manual
INSERT INTO DIM.DIM_ALMACEN (CODIGO_ALMACEN, NOMBRE_ALMACEN) VALUES ('DUBL', 'DUBLIN')
go
INSERT INTO DIM.DIM_ALMACEN (CODIGO_ALMACEN, NOMBRE_ALMACEN) VALUES ('CORK', 'CORK')
go
INSERT INTO DIM.DIM_ALMACEN (CODIGO_ALMACEN, NOMBRE_ALMACEN) VALUES ('GALW', 'GALWAY')
go
DBCC CHECKIDENT('DIM.DIM_ALMACEN', RESEED, 0)
go

/*=====*/
/* Tabla: DIM_CLIENTE */
/*=====*/
create table DIM.DIM_CLIENTE (
    ID_CLIENTE          int          identity,
    NUMERO_CUENTA       nvarchar(10) not null,
    NOMBRE_CLIENTE      nvarchar(50) not null,
    NOMBRE_ABREVIADO_CLIENTE nvarchar(50) not null,
    DIRECCION_CLIENTE   nvarchar(150) not null,
    TELEFONO_CLIENTE    nvarchar(30) not null,
    FAX_CLIENTE         nvarchar(20) not null,
    EMAIL_CONTACTO     nvarchar(50) not null,
    NOMBRE_CONTACTO    nvarchar(50) not null,
    CODIGO_LIBRO_MAYOR  char(1)      not null,
    TERMINOS_PAGO       nvarchar(5)  not null,
    CATEGORIA_CLIENTE   nvarchar(100) not null,
    GRUPO_CLIENTE       nvarchar(100) not null,
    LOCALIZACION_CLIENTE nvarchar(100) not null,
    constraint PK_DIM_CLIENTE primary key (ID_CLIENTE) on FG_VENTAS
)

```

```

)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
SET IDENTITY_INSERT DIM.DIM_CLIENTE ON
go
INSERT INTO DIM.DIM_CLIENTE
(ID_CLIENTE, NUMERO_CUENTA, NOMBRE_CLIENTE, NOMBRE_ABREVIADO_CLIENTE, DIRECCION_CLIENTE,
TELEFONO_CLIENTE, FAX_CLIENTE,
EMAIL_CONTACTO, NOMBRE_CONTACTO, CODIGO_LIBRO_MAYOR, TERMINOS_PAGO, CATEGORIA_CLIENTE,
GRUPO_CLIENTE, LOCALIZACION_CLIENTE)
VALUES (-1, '-1', 'UNKNOWN', 'UNKNOWN', 'UNKNOWN', 'UNKNOWN', 'UNKNOWN',
'UNKNOWN', 'UNKNOWN', '', '', 'UNKNOWN', 'UNKNOWN', 'UNKNOWN');
go
SET IDENTITY_INSERT DIM.DIM_CLIENTE OFF
go
DBCC CHECKIDENT('DIM.DIM_CLIENTE', RESEED, 0)
go

/*=====*/
/* Tabla: DIM_PRODUCTO */
/*=====*/
create table DIM.DIM_PRODUCTO (
ID_PRODUCTO int identity,
CODIGO_PRODUCTO int not null,
NOMBRE_PRODUCTO nvarchar(50) not null,
FORMATO_PRODUCTO nvarchar(10) not null,
PRECIO_COSTE_UNITARIO_EURO money not null,
PRECIO_VENTA_UNITARIO_EURO money not null,
CODIGO_EAN nvarchar(20) not null,
CODIGO_CATEGORIA_PRODUCTO nvarchar(5) not null,
NOMBRE_CATEGORIA_PRODUCTO nvarchar(20) not null,
constraint PK_DIM_PRODUCTO primary key (ID_PRODUCTO) on FG_VENTAS
)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
SET IDENTITY_INSERT DIM.DIM_PRODUCTO ON
go
INSERT INTO DIM.DIM_PRODUCTO
(ID_PRODUCTO, CODIGO_PRODUCTO, NOMBRE_PRODUCTO, FORMATO_PRODUCTO, PRECIO_COSTE_UNITARIO_EURO,
PRECIO_VENTA_UNITARIO_EURO, CODIGO_EAN, CODIGO_CATEGORIA_PRODUCTO, NOMBRE_CATEGORIA_PRODUCTO)
VALUES (-1, -1, 'UNKNOWN', 'UNKNOWN', 0.00, 0.00, 'UNKNOWN', 'UNK', 'UNKNOWN');
go
SET IDENTITY_INSERT DIM.DIM_PRODUCTO OFF
go
DBCC CHECKIDENT('DIM.DIM_PRODUCTO', RESEED, 0)

```

```

/*=====*/
/* Tabla: DIM_RUTA_DISTRIBUCION */
/*=====*/
create table DIM.DIM_RUTA_DISTRIBUCION (
    ID_RUTA          int          identity,
    CODIGO_RUTA      smallint     not null,
    NUMERO_RUTA      int          not null,
    NOMBRE_RUTA      nvarchar(100) not null,
    DIA_DE_REPARTO  nvarchar(20)  not null,
    NUMERO_DE_FURGONETA smallint  not null,
    HORA_DE_SALIDA   nvarchar(10)  not null,
    constraint PK_DIM_RUTA_DISTRIBUCION primary key (ID_RUTA) on FG_VENTAS
)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
SET IDENTITY_INSERT DIM.DIM_RUTA_DISTRIBUCION ON
go
INSERT INTO DIM.DIM_RUTA_DISTRIBUCION
(ID_RUTA, CODIGO_RUTA, NOMBRE_RUTA, NUMERO_RUTA, DIA_DE_REPARTO, NUMERO_DE_FURGONETA,
HORA_DE_SALIDA)
VALUES (-1, -1, 'UNKNOWN', -1, 'UNKNOWN', -1, '00:00');
go
SET IDENTITY_INSERT DIM.DIM_RUTA_DISTRIBUCION OFF
go
DBCC CHECKIDENT('DIM.DIM_RUTA_DISTRIBUCION', RESEED, 0)
go

/*=====*/
/* Tabla: DIM_REGION */
/*=====*/
create table DIM.DIM_REGION (
    ID_REGION        int          identity,
    CODIGO_REGION    smallint     not null,
    DESCRIPCION_REGION nvarchar(30) not null,
    constraint PK_DIM_REGION primary key (ID_REGION) on FG_VENTAS
)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
SET IDENTITY_INSERT DIM.DIM_REGION ON
go
INSERT INTO DIM.DIM_REGION
(ID_REGION, CODIGO_REGION, DESCRIPCION_REGION) VALUES (-1, -1, 'UNKNOWN')
go

```

```

SET IDENTITY_INSERT DIM.DIM_REGION OFF
go

DBCC CHECKIDENT('DIM.DIM_REGION', RESEED, 0)
go

/*=====*/
/* Tabla: DIM_TIPO_ORDEN */
/*=====*/
create table DIM.DIM_TIPO_ORDEN (
    ID_TIPO_ORDEN          int          identity,
    CODIGO_TIPO_ORDEN      nvarchar(10) not null,
    DESCRIPCION_TIPO_ORDEN nvarchar(50) not null,
    CATEGORIA_TIPO_ORDEN  nvarchar(20) not null,
    constraint PK_DIM_TIPO_ORDEN primary key (ID_TIPO_ORDEN) on FG_VENTAS
)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
SET IDENTITY_INSERT DIM.DIM_TIPO_ORDEN ON
go
INSERT INTO DIM.DIM_TIPO_ORDEN
(ID_TIPO_ORDEN, CODIGO_TIPO_ORDEN, DESCRIPCION_TIPO_ORDEN, CATEGORIA_TIPO_ORDEN) VALUES (-1,
'UNKNOWN', 'UNKNOWN', 'UNKNOWN')
go
SET IDENTITY_INSERT DIM.DIM_TIPO_ORDEN OFF
go
DBCC CHECKIDENT('DIM.DIM_TIPO_ORDEN', RESEED, 0)
go

/*=====*/
/* Tabla: DIM_FABRICANTE */
/*=====*/
create table DIM.DIM_FABRICANTE (
    ID_FABRICANTE          int          identity,
    CODIGO_FABRICANTE      nvarchar(10) not null,
    NOMBRE_FABRICANTE     nvarchar(50) not null,
    DISTRIBUIDOR          nvarchar(50) not null,
    constraint PK_DIM_FABRICANTE primary key (ID_FABRICANTE) on FG_VENTAS
)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
SET IDENTITY_INSERT DIM.DIM_FABRICANTE ON
go
INSERT INTO DIM.DIM_FABRICANTE
(ID_FABRICANTE, CODIGO_FABRICANTE, NOMBRE_FABRICANTE, DISTRIBUIDOR) VALUES (-1, 'UNKNOWN',

```

```
'UNKNOWN', 'UNKNOWN')
go
SET IDENTITY_INSERT DIM.DIM_FABRICANTE OFF
go
DBCC CHECKIDENT('DIM.DIM_FABRICANTE', RESEED, 0)
go

/*=====*/
/* Tabla: DIM_CAUSA_CREDITO */
/*=====*/
create table DIM.DIM_CAUSA_CREDITO (
    ID_CAUSA_CREDITO          int          identity,
    CODIGO_CAUSA_CREDITO      smallint    not null,
    DESCRIPCION_CAUSA_CREDITO nvarchar(30) not null,
    constraint PK_DIM_CAUSA_CREDITO primary key (ID_CAUSA_CREDITO) on FG_VENTAS
)
on FG_VENTAS
go

-- Insertar registro por defecto para códigos no identificados
SET IDENTITY_INSERT DIM.DIM_CAUSA_CREDITO ON
go
INSERT INTO DIM.DIM_CAUSA_CREDITO
(ID_CAUSA_CREDITO, CODIGO_CAUSA_CREDITO, DESCRIPCION_CAUSA_CREDITO) VALUES (-1, -1, 'UNKNOWN')
go
SET IDENTITY_INSERT DIM.DIM_CAUSA_CREDITO OFF
go
DBCC CHECKIDENT('DIM.DIM_CAUSA_CREDITO', RESEED, 0)
go
```

Script SQL de creación de tablas de hechos

```

use dw_ventas
go

/*=====*/
/* Table: FACT_PRESUPUESTO */
/*=====*/

create table FACT.FACT_PRESUPUESTO (
    ID_FECHA          int          not null,
    ID_CATEGORIA_PRODUCTO int      not null,
    ID_ALMACEN        int          not null,
    PRESUPUESTO       money        not null
)
on FG_VENTAS
go

create clustered index IDX_PRESUPUESTO on FACT.FACT_PRESUPUESTO (ID_FECHA ASC,
ID_CATEGORIA_PRODUCTO ASC, ID_ALMACEN ASC) on FG_VENTAS
go

alter table FACT.FACT_PRESUPUESTO
    add constraint FK_PRES_ALMACEN foreign key (ID_ALMACEN)
        references DIM.DIM_ALMACEN (ID_ALMACEN)
go

alter table FACT.FACT_PRESUPUESTO
    add constraint FK_PRES_CAT_PRODUCTO foreign key (ID_CATEGORIA_PRODUCTO)
        references DIM.DIM_CATEGORIA_PRODUCTO (ID_CATEGORIA_PRODUCTO)
go

alter table FACT.FACT_PRESUPUESTO
    add constraint FK_PRES_FECHA_VENTA foreign key (ID_FECHA)
        references DIM.DIM_FECHA (ID_FECHA)
go

/*=====*/
/* Table: FACT_LINEA_PEDIDO */
/*=====*/

create table FACT.FACT_LINEA_PEDIDO (
    ID_FECHA_VENTA      int          not null,
    ID_ALMACEN          int          not null,
    ID_CATEGORIA_PRODUCTO int      not null,
    ID_CAUSA_CREDITO    int          not null,
    ID_FABRICANTE        int          not null,
    ID_TIPO_ORDEN        int          not null,
    ID_REGION            int          not null,
    ID_FECHA_REPARTO     int          not null,

```

```

ID_CLIENTE                int                not null,
ID_PRODUCTO               int                not null,
ID_RUTA                   int                not null,
NUMERO_ORDEN              int                not null,
NUMERO_FACTURA           int                not null,
NUMERO_LINEA              int                not null,
MONTANTE_LINEA_PEDIDO_EURO money           not null,
COSTE_LINEA_PEDIDO_EURO  money           not null,
LINEA_PEDIDO              smallint           not null,
LINEA_PEDIDO_DESPACHADA  smallint           not null,
UNIDADES_PEDIDAS         int                not null,
UNIDADES_DESPACHADAS     int                not null,
constraint PK_FACT_LINEA_PEDIDO primary key nonclustered
        (ID_ALMACEN, NUMERO_ORDEN, NUMERO_LINEA) on FG_VENTAS
)
on FG_VENTAS
go

create clustered index IDX_LINPED_FECHA_VENTA on FACT.FACT_LINEA_PEDIDO (ID_FECHA_VENTA ASC) on
FG_VENTAS
go

create index IDX_LINPED_ALMACEN on FACT.FACT_LINEA_PEDIDO (ID_ALMACEN ASC) on FG_VENTAS
go

create index IDX_LINPED_CAT_PRODUCTO on FACT.FACT_LINEA_PEDIDO (ID_CATEGORIA_PRODUCTO ASC) on
FG_VENTAS
go

create index IDX_LINPED_CAUSA_CREDITO on FACT.FACT_LINEA_PEDIDO (ID_CAUSA_CREDITO ASC) on
FG_VENTAS
go

create index IDX_LINPED_FABRICANTE on FACT.FACT_LINEA_PEDIDO (ID_FABRICANTE ASC) on FG_VENTAS
go

create index IDX_LINPED_TIPO_ORDEN on FACT.FACT_LINEA_PEDIDO (ID_TIPO_ORDEN ASC) on FG_VENTAS
go

create index IDX_LINPED_REGION on FACT.FACT_LINEA_PEDIDO (ID_REGION ASC) on FG_VENTAS
go

create index IDX_LINPED_FECHA_REPARTO on FACT.FACT_LINEA_PEDIDO (ID_FECHA_REPARTO ASC) on
FG_VENTAS
go

create index IDX_LINPED_CLIENTE on FACT.FACT_LINEA_PEDIDO (ID_CLIENTE ASC) on FG_VENTAS
go

create index IDX_LINPED_PRODUCTO on FACT.FACT_LINEA_PEDIDO (ID_PRODUCTO ASC) on FG_VENTAS
go

create index IDX_LINPED_RUTA on FACT.FACT_LINEA_PEDIDO (ID_RUTA ASC) on FG_VENTAS
go

alter table FACT.FACT_LINEA_PEDIDO
    add constraint FK_LINPED_ALMACEN foreign key (ID_ALMACEN)
        references DIM.DIM_ALMACEN (ID_ALMACEN)
go

```

```
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_CAT_PRODUCTO foreign key (ID_CATEGORIA_PRODUCTO)
  references DIM.DIM_CATEGORIA_PRODUCTO (ID_CATEGORIA_PRODUCTO)
go
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_CAUSA_CREDITO foreign key (ID_CAUSA_CREDITO)
  references DIM.DIM_CAUSA_CREDITO (ID_CAUSA_CREDITO)
go
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_CLIENTE foreign key (ID_CLIENTE)
  references DIM.DIM_CLIENTE (ID_CLIENTE)
go
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_FABRICANTE foreign key (ID_FABRICANTE)
  references DIM.DIM_FABRICANTE (ID_FABRICANTE)
go
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_FECHA_REPARTO foreign key (ID_FECHA_REPARTO)
  references DIM.DIM_FECHA (ID_FECHA)
go
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_FECHA_VENTA foreign key (ID_FECHA_VENTA)
  references DIM.DIM_FECHA (ID_FECHA)
go
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_PRODUCTO foreign key (ID_PRODUCTO)
  references DIM.DIM_PRODUCTO (ID_PRODUCTO)
go
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_REGION foreign key (ID_REGION)
  references DIM.DIM_REGION (ID_REGION)
go
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_RUTA foreign key (ID_RUTA)
  references DIM.DIM_RUTA_DISTRIBUCION (ID_RUTA)
go
alter table FACT.FACT_LINEA_PEDIDO
  add constraint FK_LINPED_TIPO_ORDEN foreign key (ID_TIPO_ORDEN)
  references DIM.DIM_TIPO_ORDEN (ID_TIPO_ORDEN)
go
```


Script SQL de creación de funciones

```

use dw_ventas
go

/*****
***
*
* Función para transformar fechas en claves subrogadas
*
*****/

CREATE FUNCTION dim.FN_FECHA_TO_ID (@p_Fecha AS DATE)
RETURNS int
AS
BEGIN

RETURN CAST(YEAR(@p_Fecha) AS VARCHAR(10)) +
        CASE WHEN MONTH(@p_Fecha) < 10 THEN '0' + CAST(MONTH(@p_Fecha) AS VARCHAR(10))
              ELSE CAST(MONTH(@p_Fecha) AS VARCHAR(10))
        END +
        CASE WHEN DAY(@p_Fecha) < 10 THEN '0' + CAST(DAY(@p_Fecha) AS VARCHAR(10))
              ELSE CAST(DAY(@p_Fecha) AS VARCHAR(10))
        END

END

;

```

Script SQL de creación de procedimientos almacenados

```

use dw_ventas
go

/*****
***
*
* Creación del procedimiento de la dimensión Fecha
* Este admite dos parámetros: año desde y año hasta
*
*****/

CREATE PROCEDURE dim.sp_dim_fecha
        @p_AnoDesde int
        , @p_AnoHasta int
AS

-- Declaración de variables

```

```

DECLARE @v_FechaDesde DATE
DECLARE @v_FechaHasta DATE

-- Inicialización de variables a partir de los parámetros pasados
-- Cálculo de la fecha inicial y fecha final basado en los años pasados como parámetro
SET @v_FechaDesde = CAST(CAST(@p_AnoDesde AS VARCHAR(10)) + '-01-01' AS DATE)
SET @v_FechaHasta = CAST(CAST(@p_AnoHasta AS VARCHAR(10)) + '-12-31' AS DATE)

BEGIN

-- Iteración desde la fecha inicial hasta la fecha final
-- Si la fecha final es menor que la inicial, el procedimiento no hará nada
WHILE @v_FechaDesde <= @v_FechaHasta
BEGIN

-- Verificamos si el registro para el día de la iteración existe o no.
-- Si existe, no lo insertamos. Si no existe, lo insertamos.
IF (SELECT ISNULL(COUNT(*), 0) FROM DIM.DIM_FECHA WHERE FECHA = @v_FechaDesde) = 0
BEGIN
INSERT INTO dim.DIM_FECHA
(
ID_FECHA
,FECHA
,MES
,CUARTO
,ANO
)
SELECT
ID_FECHA = YEAR(@v_FechaDesde) * 10000 + MONTH(@v_FechaDesde) * 100 + DAY(@v_FechaDesde)
,FECHA = @v_FechaDesde
,MES = CAST(Year(@v_FechaDesde) AS VARCHAR(4)) + '/' + RIGHT('00' +
CAST(MONTH(@v_FechaDesde) AS VARCHAR(2)),2)
,CUARTO = CAST(Year(@v_FechaDesde) AS VARCHAR(4)) + '/Q'+CAST(DATEPART('QQ',@v_FechaDesde)
AS VARCHAR(1))
,ANO = YEAR(@v_FechaDesde)
;
END -- END IF

-- Incrementamos la fecha de inicio
SET @v_FechaDesde = DATEADD(DAY,1,@v_FechaDesde)

END -- END WHILE

END -- END PROCEDURE

```

Script SQL de creación de vistas

```

use dw_ventas
go

/*****
 *
 * Vista sobre la tabla temporal con limpieza de datos
 *
 *****/

CREATE VIEW stg.VW_STG_LINEA_PEDIDO
AS
SELECT
    CODIGO_ALMACEN                AS CODIGO_ALMACEN
    ,dim.FN_FECHA_TO_ID(
        ISNULL(FECHA_DE_REPARTO
            , CAST('1900-01-01' AS DATE))) AS FECHA_DE_REPARTO
    ,dim.FN_FECHA_TO_ID(
        ISNULL(FECHA_DE_VENTA
            , CAST('1900-01-01' AS DATE))) AS FECHA_DE_VENTA
    ,NUMERO_ORDEN                AS NUMERO_ORDEN
    ,NUMERO_FACTURA              AS NUMERO_FACTURA
    ,NUMERO_LINEA                 AS NUMERO_LINEA
    ,ISNULL(NUMERO_CUENTA, '')    AS NUMERO_CUENTA
    ,ISNULL(CODIGO_REGION, -1)   AS CODIGO_REGION
    ,ISNULL(CODIGO_TIPO_ORDEN, '') AS CODIGO_TIPO_ORDEN
    ,ISNULL(CODIGO_RUTA, -1)     AS CODIGO_RUTA
    ,ISNULL(CODIGO_ESTADO_LINEA, -1) AS CODIGO_ESTADO_LINEA
    ,ISNULL(CODIGO_CAUSA_CREDITO, -1) AS CODIGO_CAUSA_CREDITO
    ,ISNULL(CODIGO_PRODUCTO, -1) AS CODIGO_PRODUCTO
    ,ISNULL(CODIGO_FABRICANTE, '') AS CODIGO_FABRICANTE
    ,ISNULL(MONTANTE_LINEA_PEDIDO_EUR, 0) AS MONTANTE_LINEA_PEDIDO_EUR
    ,ISNULL(CANTIDAD_ORDENADA, 0) AS CANTIDAD_ORDENADA
    ,ISNULL(CANTIDAD_DESPACHADA, 0) AS CANTIDAD_DESPACHADA
FROM
    STG.STG_LINEA_DE_PEDIDO
;

```

Script SQL de carga de fechas

```

use dw_ventas
go
/* Carga la dimensión Fecha con datos del 2012 */
exec dim.sp_dim_fecha 2012, 2012

```

9. Apéndice B – Ficheros de Configuración

Fichero de Configuración de Conexiones

```

<?xml version="1.0"?>
<DTSTConfiguration>
  <DTSTConfigurationHeading>
    <DTSTConfigurationFileInfo>
      GeneratedBy="SQL2012W2K8\Administrator"
      GeneratedFromPackageName="Ventas_Dim_Categoria_Producto"
      GeneratedFromPackageID="{9CC793C2-C48A-4753-A47D-7D66DDBD494E}"
      GeneratedDate="04/05/2013 23:45:41"/>
    </DTSTConfigurationHeading>
  <Configuration>
    ConfiguredType="Property"
    Path="\Package.Connections[dw_ventas].Properties[InitialCatalog]"
    ValueType="String">
      <ConfiguredValue>dw_ventas</ConfiguredValue>
    </Configuration>
  <Configuration>
    ConfiguredType="Property"
    Path="\Package.Connections[dw_ventas].Properties>Password"
    ValueType="String">
      <ConfiguredValue>PFCVentas2013</ConfiguredValue>
    </Configuration>
  <Configuration>
    ConfiguredType="Property"
    Path="\Package.Connections[dw_ventas].Properties[ServerName]"
    ValueType="String">
      <ConfiguredValue>SQL2012W2K8</ConfiguredValue>
    </Configuration>
  <Configuration>
    ConfiguredType="Property"
    Path="\Package.Connections[dw_ventas].Properties[UserName]"
    ValueType="String">
      <ConfiguredValue>dw_etl</ConfiguredValue>
    </Configuration>
</DTSTConfiguration>

```

Fichero de Configuración de Parámetros

```

<?xml version="1.0"?>
<DTSConfiguration>
  <DTSConfigurationHeading>
    <DTSConfigurationFileInfo
      GeneratedBy="SQL2012W2K8\Administrator"
      GeneratedFromPackageName="Ventas_Maestro"
      GeneratedFromPackageID="{9BD67B03-1713-4BF0-9724-98CA0A1AFD2B}"
      GeneratedDate="06/05/2013 18:44:06"/>
    </DTSConfigurationHeading>
  <Configuration
    ConfiguredType="Property"
    Path="\Package.Variables[User::v_Log_Ext_Fichero].Properties[Value]"
    ValueType="String">
    <ConfiguredValue>.log</ConfiguredValue>
  </Configuration>
  <Configuration
    ConfiguredType="Property"
    Path="\Package.Variables[User::v_Log_Nombre_Fichero].Properties[Value]"
    ValueType="String">
    <ConfiguredValue>ventas_</ConfiguredValue>
  </Configuration>
  <Configuration
    ConfiguredType="Property"
    Path="\Package.Variables[User::v_Path_Ficheros].Properties[Value]"
    ValueType="String">
    <ConfiguredValue>C:\SSIS\Ventas\ficheros</ConfiguredValue>
  </Configuration>
  <Configuration
    ConfiguredType="Property"
    Path="\Package.Variables[User::v_Path_Log].Properties[Value]"
    ValueType="String">
    <ConfiguredValue>C:\SSIS\Ventas\log</ConfiguredValue>
  </Configuration>
  <Configuration
    ConfiguredType="Property"
    Path="\Package.Variables[User::v_Path_Paquetes].Properties[Value]"
    ValueType="String">
    <ConfiguredValue>C:\SSIS\Ventas\dtsx</ConfiguredValue>
  </Configuration>
</DTSConfiguration>

```