eman ta zabal zazu

Escuela Universitaria de Ingeniería Técnica Industrial de Bilbao

Grado en (Titulación)

Trabajo Fin de Grado

2014 / 2015

Software como un Servicio: Solución o Business Intelligence Memoria

Datos de la Alumna o del Alumno	**Datos del Director o de la Directora**
Nombre: Jose ignacio	Nombre: ana jesus
Apellidos: sánchez méndez	Apellidos: armendariz leunda
nombre: josu	Nombre: begoña
apellidos: rodríguez azpeleta	Apellidos: blanco jauregui
	Departamento: LENGUAJES Y SISTEMAS INFORMÁTICOS
Fdo.:	Fdo.:
Fecha:17-06-2015	Fecha:17-06-2015

Software como un Servicio: Solución Business Intelligence

Memoria

José Ignacio Sánchez Méndez y

Josu Rodríguez Azpeleta

19 de junio de 2015

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La rama de conocimiento de la Informática es probablemente uno de los campos de mayor y más veloz evolución a día de hoy. Su enfoque ha pasado en relativamente poco tiempo de proporcionar herramientas que facilitan cálculos a los humanos, a proveer innumerables aplicaciones en el día a día, introduciéndose y llegando a ser incluso esencial en ámbitos como la medicina, la gestión de empresas u organizaciones, e incluso el tiempo de ocio.

En este contexto, bien sea por necesidades o como mera consecuencia del uso de aplicaciones informáticas, se han generado volúmenes ingentes de datos a lo largo de los años de uso de las mismas. Debido a la abrumadora cantidad de estos, su uso podría parecer harto inaccesible hasta hace unas décadas; ahora, sin embargo, con un hardware de elevadas capacidades y accesible a cualquier persona o entidad dispuesta a usarlo, existe la oportunidad de explorar y explotar estos datos en busca de información, con fines tan dispares como los académicos o militares. Todas las teorías, técnicas y conocimiento al respecto han terminado por desembocar en una rama de conocimiento que podría considerarse independiente a cualquier otra, que es la Ciencia de los Datos (Data Science).

1.1. Propósito

El propósito del proyecto aquí descrito radica en, por una parte, sentar una base de un sistema de Business Inteligence adaptable a diversos casos de negocio, y por otra, diseñar e implementar una solución completa para una empresa específica fácilmente adaptable a otro caso, incluyendo desde los procesos de Extracción, Transformación y Carga, pasando por el *data warehouse* hasta el *Business Analysis* y la Minería de Datos.

1.2. Ámbito

En la actualidad, el entorno empresarial se halla repleto de datos, distribuidos a lo largo de múltiples y dispares fuentes, además de en muchos casos, inconexos. Esto ha creado la oportunidad de, mediante las más actuales metodologías, formación y personal adecuados, extraer información de dichos datos.

Este hecho, que a priori puede resultar incluso indiferente para aquel que no disponga de una perspectiva lo suficientemente abierta, ha creado unas oportunidades de negocio inéditas. Desde el más sencillo análisis visual o estadístico de los datos hasta la minería de los mismos, se abre un amplio abanico de posibilidades de lograr rendimiento de todo el enorme flujo de datos que circula a diario en los sistemas de gestión empresariales.

En la coyuntura económica actual, cualquier hecho diferenciador con respecto a la competencia puede suponer pasar de ser una mera empresa más a ser una referencia del sector. Aquí es dónde estos sistemas de Business Inteligence pueden brillar, y la visión que deseamos dar a este proyecto.

1.3. Contexto de negocio

El presente sistema de software pretende ser desarrollado de forma tal que permita la flexibilidad suficiente para poder ser adaptado a diversos negocios con relativa facilidad y mínimo coste económico-temporal.

El contexto de negocio concreto de la empresa cliente es el ámbito de los servicios de asistencia técnica informática, abarcando tanto servicios hardware como software, así como venta de suministros (p.ej.: tinta de impresora) y componentes o dispositivos informáticos. Cabe mencionar que el sistema también está siendo explotado actualmente por el proveedor, que se dedica al sector de la consultoría informática.

1.4. Definiciones, acrónimos y abreviaturas

Esta sección está dedicada a las definiciones y acrónimos de carácter técnico que merezcan se definidas o desplegadas:

- BI: Inteligencia Empresarial, del Inglés *Business Intelligence*. Business Intelligence es la habilidad para transformar los datos en información, y la información en conocimiento, de forma que se pueda optimizar el proceso de toma de decisiones en los negocios.
- SaaS (ScuS): El software como servicio (SaaS) es software que se usa a través de una red sin descargarlo en un ambiente de cómputo local. Se obtiene acceso a la aplicación de software a través de Internet desde un proveedor y se ejecuta en el ambiente de cómputo predefinido del proveedor.
- DWH: Almacén de datos, del Inglés Data Warehouse: Es una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis". También fue Kimball quien determinó que un data warehouse no era más que: "la unión de todos los Data marts de una entidad".[4] Defiende por tanto una metodología ascendente (bottom-up) a la hora de diseñar un almacén de datos.
- Datamart: Un Datamart es una base de datos departamental, especializada en el almacenamiento de los datos de un área de negocio específica. Se caracteriza por disponer la estructura óptima de datos para analizar la información al detalle desde todas las perspectivas que afecten a los procesos de dicho departamento.
- ETL: Del inglés *Extract, Transform and Load*. Comprende los procedimientos de obtención de datos dsde las fuentes siguiendo la estrategia de extraer, transformar y cargar.
- JDBC: Conector Java para poder acceder a un sistema de bases de datos concreto, diferente para cada sistema.

- Staging area: En *data warehousing* se utiliza un sistema de almacenamiento para traer todos los datos necesarios en crudo, se puede utilizar un sistema de base de datos relacional, o incluso un sistema de ficheros de texto plano. Estas decisiones serán tomadas por el equipo de análisis del proyecto, previo al diuseño y la implementación.
- ERP:(Por sus siglas en inglés,enterprise resource planning), Sistema de planificación de recursos empresariales gerenciales que integran y manejan muchos aspectos de los negocios asociados a las diferentes operaciones.

Capítulo 2

Planteamiento inicial

2.1. Objetivos

Según expertos en el campo del Business Intelligence, actualmente menos del 3 % de las empresas nacionales explotan el potencial del que disponen en sus orígenes de datos, recogido a través de las transacciones que realizan a diario a través de sus sistemas de gestión. No obstante algunas empresas, al igual que las grandes firmas, intentan descubrir los gustos y tendencias de sus actuales y potenciales clientes, buscando la estrategia que permita ofrecer productos o servicios adaptados a cada segmento, tratando de emular un trato individualizado con cada cliente. Otras organizaciones necesitan un control estratégico de la localización de su actividad, pero no disponen de la tecnología o los conocimientos necesarios para disponer de la información de una manera eficaz y clara. En algunos casos simplemente no se dispone de los mismos recursos que las grandes empresas para contratar técnicos expertos que desarrollen un proyecto ajustado a su negocio y un largo etcétera.

La oportunidad de explotar el dato aparece cuando muchas empresas se dan cuenta que tienen muchos datos en diferentes sistemas y archivos (ERP, CRM, hojas de cálculo, redes sociales, etc), y no lo explotan. Y aquí el principal problema está en que no hay una "explotación cerrada". Es decir, no hay un conjunto de utilidades o preguntas tipo. Cada conjunto de datos, cada realidad de empresa, es un proyecto nuevo. Es difícil industrializar esto. Hay tantos enfoques prácticamente como empresas.

Es por ésto que el objetivo principal del proyecto se trata de desarrollar un sistema con un coste de desarrollo asumible para estas empresas y que disponga del potencial suficiente para convertir en conocimiento los datos en crudo de los que actualmente disponen, añadiendo valor a sus negocios y posibilitando la reutilización de la mayor parte del sistema para diferentes empresas, permitiendo el desarrollo semi-industrializado, para el dominio de fuentes de datos origen definido.

Desde el punto de vista de la empresa que nos encarga el proyecto consiste en desarrollar un Sistema de Business Intelligence, de ahora en adelante BI, como un *Servicio* utilizando un modelo donde el soporte lógico y el *data warehouse* (DWH) se alojan en sus servidores. Esta empresa se dedica al mundo de la consultoría TIC y hasta ahora poseen una solida experiencia en servicios ERP de esta tipología con herramientas *Open Source*, pero no en el campo de BI y Big Data, así el objetivo del proyecto se convierte en un objetivo más general que consiste en ampliar el mercado al cual actualmente no tienen acceso con un nuevo servicio que permita crear un departamento dedicado a esta finalidad y consolidar las relaciones con los clientes actuales, mejorando los servicios de los que ya disponen actualmente.

Es prioridad disponer de la capacidad de análisis lo suficientemente flexible, que permita el estudio de los mismos hechos de negocio para diferentes clientes sin la condición necesaria de que exploten su actividad en el mismo sector. Tratándose de un *SaaS*, los clientes deben disponer la posibilidad de acceder desde cualquier dispositivo, se encuentren o no en las instalaciones corporativas, de una forma segura y explotando las posibilidades que ofrece un diseño híbrido que permita utilizar distintos dispositivos para su uso, sumando la ventaja de un sistema BI móvil.

Desde nuestro punto de vista se trata de aprovechar la oportunidad de utilizar no sólo los conocimientos técnicos adquiridos a lo largo de la carrera, si no también aquellos que nos permiten la auto-formación a través del *I+D+I* para desarrollar un proyecto innovador que nos permita avanzar en el mundo del BI y Big Data ya que nos encontramos en el convencimiento de que es hacia donde se dirige la atención de muchas de las empresas que aún tienen mucho potencial por explotar en los datos que poseen y tenemos la base suficiente para iniciarnos.

2.2. Antecedentes

Es sabido por todo experto en la disciplina que la explotación de los datos y su producto el Business Intelligence no son ningún tipo de materia recién descubierta. Existe desde hace años y ha evolucionado a pasos agigantados durante este tiempo. De todo este proceso de evolución han surgido casos sonados como la multinacional Amazon con su sistema de recomendación de compras[11].

Sin embargo, para bien o para mal, el sector del Business Intelligence aún se encuentra pasando desapercibido entre las PYMEs -en nuestro país al menos-[1], tanto a nivel de demanda como de oferta. Es por ello que las pocas empresas que decidan dar el paso y apostar por ello gozarán de un potencial económico sin par en el negocio del software de gestión empresarial.

[1] Fuente: conocimiento del estado del mercado laboral, la oferta y la demanda de puestos de trabajo y proyectos al respecto.

En este contexto es donde se desea encajar este proyecto: aunque el BI no es ningún tipo de novedad, no se encuentra reconocido a nivel de empresas de pequeño y mediano tamaño. Pese a compartir objetivos con los sistemas de empresas de gran calibre, las metodologías empleadas y el usuario objetivo distan mucho de los de estas últimas. Por ambas cuestiones se trata de un proyecto que, además de útil, goza de un factor innovativo considerable.

2.3. Arquitectura SaaS

La definición de la arquitectura se establece por la naturaleza del servicio en la nube, teniendo en cuenta los objetivos del proyecto. El producto final va a estar desplegado sobre una arquitectura cliente-servidor intentando volcar la mayor parte del trabajo en el servidor dejando el cliente unicamente como punto de visualización de la información.

El servidor constará de tres partes diferenciadas, por un lado se encuentra el módulo encargado de extraer, transformar y cargar los datos desde las fuentes de origen a un segundo módulo que es el *DWH* corporativo encargado de mantener el histórico para servir al resto de módulos de la aplicación. El tercer módulo es el propio servidor BI, encargado de procesar todas las consultas para obtener los datos desde el DWH corporativo y presentarlos en el lado del cliente.

Además se dispone de un cuarto módulo en el servidor que es el encargado de la extracción de conocimiento utilizando diferentes técnicas de minería de datos a determinar en función de la naturaleza de cada cliente, como ejemplo se plantea implementar un módulo capaz de predecir tiempos utilizando técnicas de regresión y otra parte de este módulo se encargará de aplicar técnicas de clasificación no supervisada para agrupar los clientes de la empresa en K grupos diferentes para ser estudiados con detenimiento, en busca de patrones de conducta que los relacione.

A continuación se muestra gráficamente la arquitectura descrita, figura 2.1.

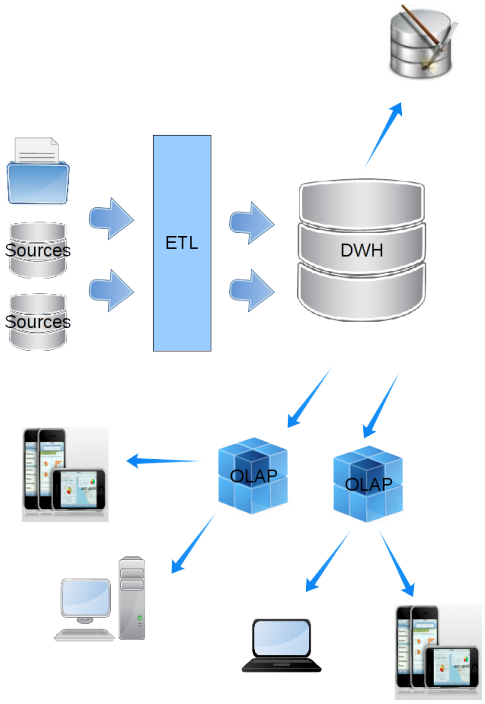


Figura 2.1: Arquitectura general del sistema

2.4. Herramientas y tecnologías

En esta sección se listan aquellas herramientas que se utilizarán a lo largo de todo el proyecto para todas y cada una de las tareas.

Para llevar a termino los paquetes de trabajo del proyecto son necesarias herramientas de diferente funcionalidad. Algunas de las tareas es posible solventarlas con con un procesador de textos, como es el caso de la documentación, otras en cambio necesitan herramientas de diseño e implementación más sofisticadas como *SQL Power Architect* para el diseño e implementación de la arquitectura *DWH* o *Kettle* para el diseño e implementación de los procedimientos *ETL*.

Herramientas de Bases de datos: *PGAdmin3 y Pentaho Data Integration (PDI).* PGAdmin es la herramienta gráfica de gestión de base de datos de PostgreSQL. Se emplea para verificar integridad de datos, implementar sentencias SQL que serán llevadas a dashboards, etcétera. PDI es una herramienta ETL mantenida por Pentaho, será la utilizada para la extracción, procesado y volcado de datos en el Data Warehouse.

Herramientas OLAP: Interfaz *Mondrian* y el lenguaje de consulta analítico *MDX*. Mondrian es el nombre de tanto el servidor OLAP como la interfaz de cubos multidimensionales de Pentaho. Su uso permite un acceso más dinámico a los datos que con un sistema de bases de datos relacional. MDX es el lenguaje de sentencias que permite hacer solicitudes a Mondrian.

Herramientas para la documentación, análisis y diseño:

- LaTeX: debido a su potencia, flexibilidad y facilidades a la hora de ser utilizado en un sistema de control de versiones, el lenguaje LaTeX ha sido la herramienta elegida para llevar a cabo la documentación del proyecto.
- LibreOffice Draw: con esta herramienta open-source se generarán los gráficos, imágenes o esquemas para facilitar la legibilidad de la documentación.
- SQL Power Architect: Herramienta utilizada para el diseño del Data Warehouse, y su posterior volcado a una base de datos real en PostgreSQL. Después de diseñar gráficamente la base de datos deseada, SQL Power Architect da la opción de crear un script SQL que creará dicha base de datos con todos los requisitos introducidos en el diseño.
- Pencil: Herramienta de prototipado de interfaces. No hemos valido de este software para desarrollar los prototipos de los cuadros de mando.

Herramientas y tecnologías para la implementación

- Geany: un editor de texto sencillo enfocado a la escritura o modificación de código fuente, incluye posibilidades de compilación de diversos tipos de archivos.
- Pentaho BI server: módulo central de la suite de Business Inteligence de Pentaho. Es usado tanto para el desarrollo de dashboards como para su presentación de cara al usuario final. Proporciona otras herramientas útiles para el desarrollo como Saiku Analytics.
- HTML5, Javascript, CSS3, Bootstrap: herramientas utilizadas para ampliar la funcionalidad de los dashboards y mejorar su diseño visual.
- Lenguajes SQL, PostgreSQL y Progress: Lenguajes y motores SQL de acceso a base de datos. PostgreSQL es usado como base de datos para el Data Warehouse, siendo necesario tanto para crear el DWH como para introducir y obtener datos de él. Progress es la base de datos orígen.
- XML - Mondrian: Interfaz XML para la implementación de cubos OLAP mediante los que se podrá realizar un acceso multidimensional a los datos.
- Saiku: Plug-in de análisis OLAP instalable en Pentaho BI Server. Es utilizado como herramienta de testeo de sentencias MDX.
- Lenguaje analítico de consulta MDX: Lenguaje de sentencias de acceso a bases de datos multidimensionales.
- OpenEdge Developer Studio: IDE para el desarrollo de aplicaciones de Progress. Se utiliza en el proyecto para explorar la base de datos origen.

Herramientas para el control de versiones

- Apache Subversion (SVN): Sistema de control de versiones utilizado para el control de, por una parte, diferentes componentes del sistema final como los esquemas Mondrian, y por otra parte, la gestión de la documentación del proyecto.
- Eclipse: IDE utilizado para facilitar el uso de SVN mediante el correspondiente plug-in.
- GitLab: Para el desarrollo de la memoria y el resto de la documentación realizada fuera del ámbito de la empresa, se hace uso de este sistema de control de versiones *privado* basado en git con una interfaz muy ligera y comoda.

2.5. Alcance del proyecto

La metodología utilizada para el proyecto se pretende acercar a las metodologías ágiles siguiendo el siguiente marco estratégico:

- Los requerimientos son por definición cambiantes, se permite que el cliente cambie de idea sobre lo que necesita.
- Entrega frecuente y progresiva de resultados, por lo que en todo momento se tiene una visión clara de la evolución de los resultados.

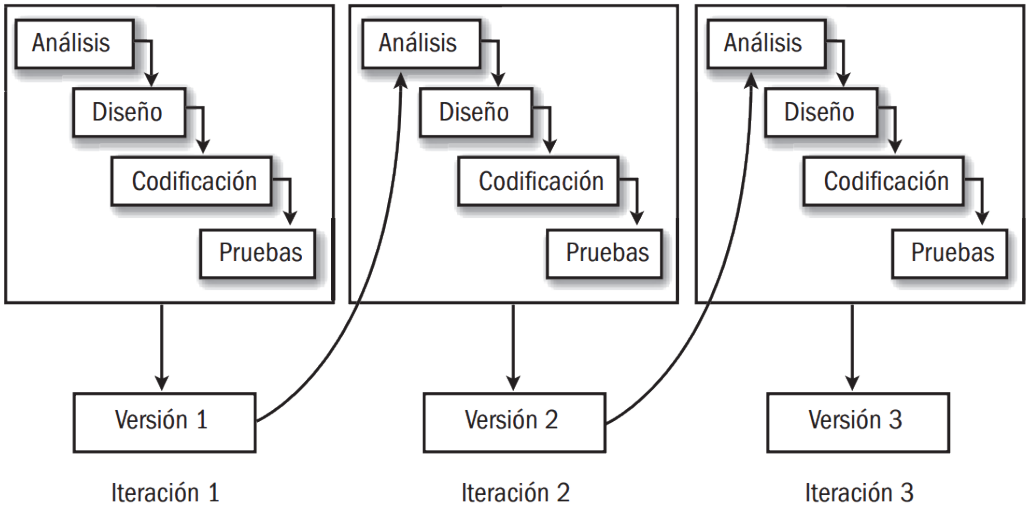


Figura 2.2: Diseño iterativo de software

[2]

Ésto posibilita enriquecer el proyecto con *feedback* periódico de los usuarios, incluyendo el desarrollo del proyecto en un proceso de mejora continua, figura 2.3.

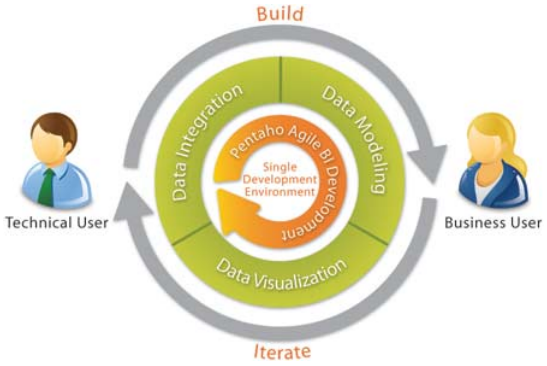


Figura 2.3: Proceso de mejora continua

[2]Imagen extraída de http://sings-ufps.blogspot.com.es/2012/04/ciclo-de-vida-conceptos.html

3

La *Estructura de Descomposición de Trabajo* 2.4 se presenta de una forma genérica, quedando cada una de las tareas inmersas en un proceso de mejora continua.

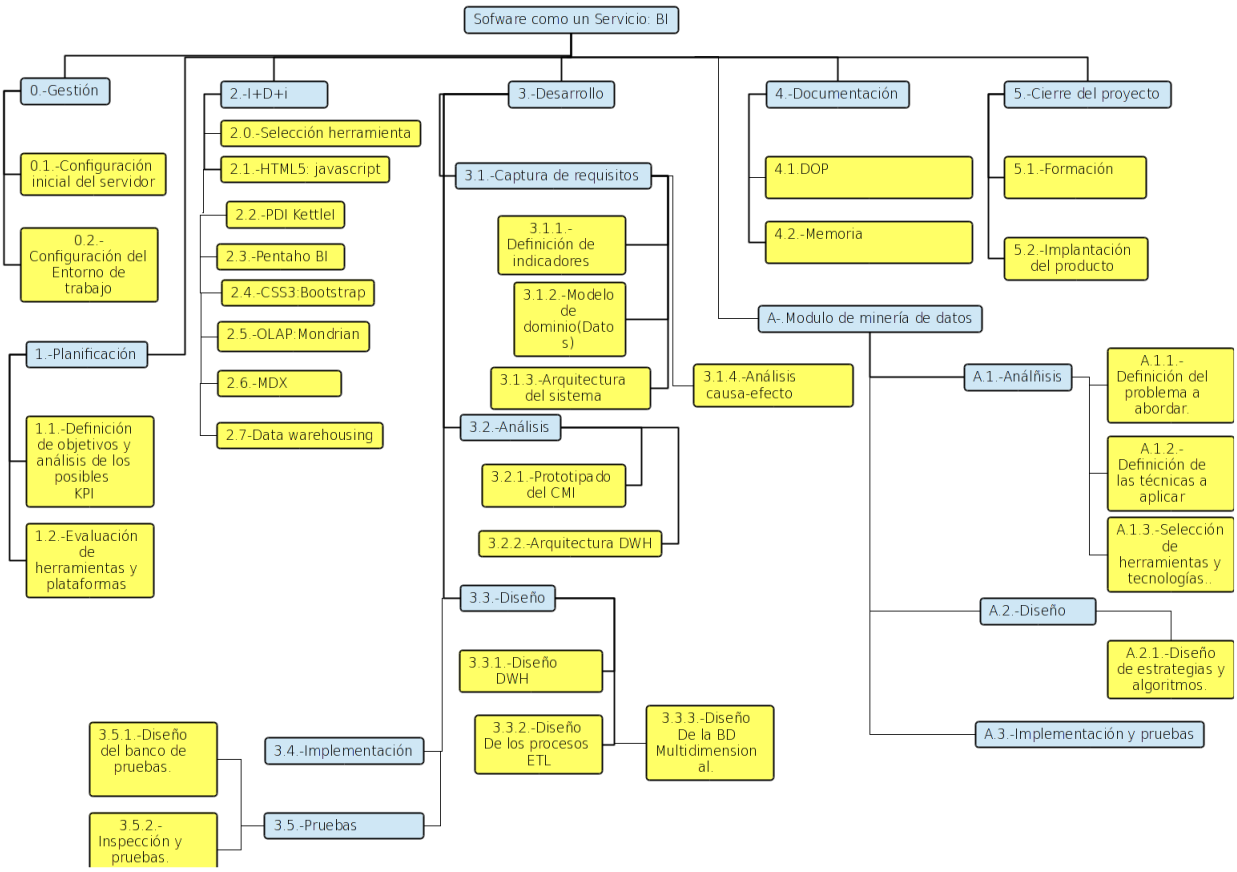


Figura 2.4: Estructura de Descomposición del Trabajo

[3]Imagen extraída de https://sqlbicro.wordpress.com/2013/02/13/agilna-poslovna-inteligencija/

2.5.1. Descripción de los paquetes de trabajo

A continuación se presenta la descripción en detalle de cada una de las tareas, el tiempo estimado y las precedencias para cada una de elllas, lo cual nos facilitará la información suficiente para finalizar la estimación de la planificación temporal.

Descripción del paquete de trabajo
Paquete de trabajo: 0.1. Configuración inicial del servidor de desarrollo **Responsable:** Jose Ignacio Sánchez **Duración estimada:** 2 **horas**
Descripción Configurar un servidor de desarrollo para los trabajos de implementación y pruebas. **Entradas** Ninguna **Salidas/Entregables** Ninguna **Recursos necesarios** Máquina virtual en la red de la empresa para simular el entorno de producción. Puesto con conexión a la red de la empresa para acceder a la Máquina virtual. **Precedencias** Ninguna

Tabla 2.1: Descripción del paquete de trabajo 0.1

Descripción del paquete de trabajo
Paquete de trabajo: 0.2. Configuración del entorno de trabajo **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez **Duración estimada:** 3 **horas**
Descripción Configurar todas las herramientas necesarias en el servidor y en los puestos de cada integrante del equipo. **Entradas** Servidor configurado y accesible desde el puesto de trabajo de los integrantes del equipo de desarrollo. **Salidas/Entregables** Ninguna **Recursos necesarios** Puesto con conexión a la red de la empresa para acceder a la Máquina virtual con el servidor, eclipse, PDI Pentaho (Kettle), Geany, PGAdmin3, conexión con los datos que simulan el entorno real. **Precedencias** Configuración inicial del servidor.

Tabla 2.2: Descripción del paquete de trabajo 0.2

Descripción del paquete de trabajo
Paquete de trabajo: 1.1. Definición de objetivos y análisis de los posibles KPI **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez **Duración estimada:** 8 **horas**
Descripción Esta tarea incluye reuniones con el cliente y con el consultor para estudiar los procesos de negocio a modelar para su análisis y estudiar los posibles indicadores. **Entradas** Ninguna. **Salidas/Entregables** Documento con los posibles indicadores a plantear en reuniones futuras con el cliente. **Recursos necesarios** Sala de reuniones. **Precedencias** Ninguna.

Tabla 2.3: Descripción del paquete de trabajo 1.1

Descripción del paquete de trabajo
Paquete de trabajo: 1.2. Evaluación de herramientas y plataformas. **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez **Duración estimada:** 20 **horas**
Descripción Esta tarea consiste en analizar y evaluar las mejores herramientas existentes en el mercado que mejor se ajuste a las necesidades del proyecto y a los requisitos del cliente. **Entradas** Requisitos del proyecto y el cliente. **Salidas/Entregables** Evaluación de las herramientas de BI seleccionadas como candidatas para implementar el proyecto. **Recursos necesarios** PC con conexión a internet. **Precedencias** Captura de requisitos.

Tabla 2.4: Descripción del paquete de trabajo 1.2

Descripción del paquete de trabajo
Paquete de trabajo: 2. I+D+i. **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez **Duración estimada:** 65 **horas**
Descripción Esta tarea contiene como subtareas las tecnologías y herramientas sobre las cuales se aplica y que quedan recogidas en ésta dada su descripción común: ▪ 2.0-Selección de herramientas. ▪ 2.1-HTML5:Javascript. ▪ 2.2-PDI-Kettle y procedimientos ETL. ▪ 2.3-Pentaho BI Server y sus componentes. ▪ 2.4-CSS3: Bootstrap. ▪ 2.5-OLAP Mondrian. ▪ 2.6-MDX. ▪ 2.7-Data warehousing. **Entradas** Ninguna. **Salidas/Entregables** Ninguna. **Recursos necesarios** PC con conexión a internet y las herramientas sobre las que se investiga en cada tarea instaladas. **Precedencias** Ninguna.

Tabla 2.5: Descripción del paquete de trabajo 2

Descripción del paquete de trabajo
Paquete de trabajo: 3.1.1.Definición de indicadores. **Responsable:** Jose Ignacio Sánchez. **Duración estimada:** 12 **horas**
Descripción Esta tarea consiste en definir los indicadores (KPI) definitivos a utilizar el proyecto. **Entradas** Posibles KPI. **Salidas/Entregables** Documento con los diagramas de causa-efecto. **Recursos necesarios** PC con software LibreOffice Draw instalado, Latex y el documento con los posibles indicadores. **Precedencias** Definición de objetivos.

Tabla 2.6: Descripción del paquete de trabajo 3.1.1

Descripción del paquete de trabajo
Paquete de trabajo: 3.1.2.Modelo de dominio origen. **Responsable:** Josu Rodríguez. **Duración estimada:** 15 **horas**
Descripción Dado que los sistemas de datos del origen no disponen de documentación, esta tarea consiste el estudio del modelo de dominio origen para extraer el conocimiento necesario que nos permita avanzar con éxito en el proyecto. **Entradas** KPI definidos. **Salidas/Entregables** Diseño conceptual que modela el hecho de negocio. **Recursos necesarios** PC con el software OpenEdge instalado que permite explorar la fuente de datos origen, Latex y LibreOffice Draw. **Precedencias** Definición de objetivos.

Tabla 2.7: Descripción del paquete de trabajo 3.1.2

Descripción del paquete de trabajo
Paquete de trabajo: 3.1.3.Arquitectura del sistema. **Responsable:** Jose Ignacio Sánchez. **Duración estimada:** 8 **horas**
Descripción Esta tarea consiste en definir una arquitectura eficiente para el sistema, que permita un funcionamiento eficaz del sistema. **Entradas** Modelo de dominio definido y arquitectura del sistema origen. **Salidas/Entregables** Diagrama con la arquitectura del sistema. **Recursos necesarios** PC con el software OpenEdge instalado que permite explorar la fuente de datos origen, Latex y LibreOffice Draw. Acceso al alojamiento de los datos del cliente con la finalidad de estudiar las conexiones remotas para la extracción de datos. **Precedencias** Modelo de dominio.

Tabla 2.8: Descripción del paquete de trabajo 3.1.3

Descripción del paquete de trabajo
Paquete de trabajo: 3.2.1.Prototipado del CMI. **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez. **Duración estimada:** 10 **horas**
Descripción Esta tarea consiste en realizar los prototipos iniciales para los *dashboards*. **Entradas** Ninguna. **Salidas/Entregables** Prototipos en papel o digitales. **Recursos necesarios** Papel y lápiz o PC con software para el prototipado en función de la elección del encargado de realizar la tarea. **Precedencias** Captura de requisitos.

Tabla 2.9: Descripción del paquete de trabajo 3.2.1

Descripción del paquete de trabajo
Paquete de trabajo: 3.2.2.Arquitectura del *DWH*. **Responsable:** Jose Ignacio Sánchez. **Duración estimada:** 6 **horas**
Descripción Esta tarea consiste en realizar la arquitectura del DWH. **Entradas** Arquitectura del sistema. **Salidas/Entregables** Diagrama con la arquitectura del DWH. **Recursos necesarios** PC con *LibreOffice Draw*. **Precedencias** Captura de requisitos.

Tabla 2.10: Descripción del paquete de trabajo 3.2.2

Descripción del paquete de trabajo
Paquete de trabajo: 3.3.1.Diseño del *DWH*..
Responsable: Jose Ignacio Sánchez y Josu Rodríguez.
Duración estimada: 16 **horas**
Descripción
Esta tarea consiste en realizar el diseño del DWH determinando las tablas de hechos y dimensiones necesarias en cada iteración.
Entradas
Diagrama de arquitectura del DWH, Diagramas de causa-efecto, Diagrama de la BD origen.
Salidas/Entregables
Proyecto architect con el diseño DWH.
Recursos necesarios
PC con *Power SQL Architect*.
Precedencias
Análisis.

Tabla 2.11: Descripción del paquete de trabajo 3.3.1

Descripción del paquete de trabajo
Paquete de trabajo: 3.3.2.Diseño de los procesos *ETL*.
Responsable: Jose Ignacio Sánchez y Josu Rodríguez.
Duración estimada: 20 **horas**
Descripción
Esta tarea consiste en realizar el diseño de los procesos necesarios para la extracción de los datos de las fuentes de origen, las transformaciones necesarias para obtener el formato requerido y la carga en el sistema objetivo.
Entradas
Diagrama de diseño del DWH y Diagrama de la BD origen.
Salidas/Entregables
Proyecto ETL PDI pentaho.
Recursos necesarios
PC con *Power SQL Architect*, OpenEdge y Kettle instalados.
Precedencias
Diseño DWH.

Tabla 2.12: Descripción del paquete de trabajo 3.3.2

Descripción del paquete de trabajo
Paquete de trabajo: 3.3.3.Diseño de la BD Multidimensional: *OLAP*.
Responsable: Josu Rodríguez.
Duración estimada: 10 **horas**
Descripción
Esta tarea consiste en realizar el diseño de la interfaz *mondrian* que permite el acceso multidimensional a al *DWH* construyendo los denominados cubos *OLAP*.
Entradas
Diagrama de diseño del DWH.
Salidas/Entregables
Base de datos multidimensional OLAP.
Recursos necesarios
PC con *Power SQL Architect*, mondrian y Geany instalados.
Precedencias
Diseño DWH.

Tabla 2.13: Descripción del paquete de trabajo 3.3.3

Descripción del paquete de trabajo
Paquete de trabajo: 3.4.Implementación. **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez. **Duración estimada:** 140 **horas**
Descripción Esta tarea consiste en la implementación de cada uno de los módulos necesarios: *DWH*, procedimientos *ETL*, cubos *OLAP*, modificaciones en el código del servidor y los componentes, implementación de la capa de presentación personalizada y la implementación de los dashboards. **Entradas** Diagrama de diseño del DWH, Diseño de los procesos ETL, diseño de los cubos OLAP,prototipado CMI y el documento con la definición de los indicadores. **Salidas/Entregables** Sistema implementado. **Recursos necesarios** PC con *Power SQL Architect*, mondrian, Geany, OpenEdge, Kettle, Pentaho BI Server, eclipse, svn y LibreOffice Draw instalados, también es necesario disponer de conexión con el servidor de desarrollo. **Precedencias** Diseño.

Tabla 2.14: Descripción del paquete de trabajo 3.4

Descripción del paquete de trabajo
Paquete de trabajo: 3.5.1 Diseño del banco de pruebas. **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez. **Duración estimada:** 16 **horas**
Descripción Esta tarea consiste en el diseño del banco de pruebas para cada módulo. **Entradas** Diagrama de diseño del DWH, Diseño de los procesos ETL, diseño de los cubos OLAP,prototipado CMI y el documento con la definición de los indicadores. **Salidas/Entregables** Sistema implementado. **Recursos necesarios** PC con *Power SQL Architect*, eclipse, svn y LibreOffice instalados. **Precedencias** Implementación.

Tabla 2.15: Descripción del paquete de trabajo 3.5.1

Descripción del paquete de trabajo
Paquete de trabajo: 3.5.2 Inspección y pruebas. **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez. **Duración estimada:** 20 **horas**
Descripción Esta tarea consiste llevar a cabo las pruebas diseñadas. **Entradas** Diseño del banco de pruebas. **Salidas/Entregables** Documento con el resultado de las pruebas realizadas. **Recursos necesarios** PC con el modulo a testear instalado y con LibreOffice. **Precedencias** Diseño del banco de pruebas.

Tabla 2.16: Descripción del paquete de trabajo 3.5.2

Descripción del paquete de trabajo
Paquete de trabajo: 4.1. Desarrollo del DOP. **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez. **Duración estimada:** 40 **horas**
Descripción Tarea que consiste en realizar el documento donde se encuentran recogidas las intenciones del proyecto. **Entradas** Ninguna. **Salidas/Entregables** Documento de Objetivos del Proyecto (DOP). **Recursos necesarios** PC con Latex y LibreOffice instalados. **Precedencias** Ninguna.

Tabla 2.17: Descripción del paquete de trabajo 4.1

Descripción del paquete de trabajo
Paquete de trabajo: 4.2. Memoria. **Responsable:** Jose Ignacio Sánchez y Josu Rodríguez. **Duración estimada:** 85 **horas**
Descripción Tarea que consiste en realizar el documento donde se encuentra toda la información asociada al proyecto. **Entradas** Ninguna. **Salidas/Entregables** Memoria del Trabajo de Fin de Grado. **Recursos necesarios** PC con Latex, LibreOffice, Eclipse, Power SQL Architect y acceso a los recursos implementados. **Precedencias** Análisis, Diseño, Implementación y pruebas de cada uno de los componentes del SaaS.

Tabla 2.18: Descripción del paquete de trabajo 4.2

Descripción del paquete de trabajo
Paquete de trabajo: 5.1. Formación. **Responsable:** Jose Ignacio Sánchez. **Duración estimada:** 12 **horas**
Descripción Tarea que consiste en formar a los usuarios del sistema. **Entradas** Ninguna. **Salidas/Entregables** Ninguno. **Recursos necesarios** Acceso al sistema SaaS. **Precedencias** Implantación del producto.

Tabla 2.19: Descripción del paquete de trabajo 5.1

Descripción del paquete de trabajo
Paquete de trabajo: 5.2. Despliegue del servidor. **Responsable:** Josu Rodriguez. **Duración estimada:** 6 **horas**
Descripción Tarea que consiste en desplegar el sistema y asegurar el correcto funcionamiento. **Entradas** Ninguna. **Salidas/Entregables** Ninguno. **Recursos necesarios** La infraestructura y los módulos necesarios para desplegar el sistema. **Precedencias** Pruebas.

Tabla 2.20: Descripción del paquete de trabajo 5.2

Descripción del paquete de trabajo
Paquete de trabajo: A.1.1. Definición del problema a abordar. **Responsable:** Jose Ignacio Sánchez. **Duración estimada:** 6 **horas**
Descripción Tarea que consiste en estudiar la información disponible y el negocio para definir que problema abordar para la aplicación de técnicas de *KDD*. **Entradas** SaaS. **Salidas/Entregables** Documento con la definición del problema a abordar. **Recursos necesarios** El sistema y los datos accesibles. **Precedencias** Pruebas.

Tabla 2.21: Descripción del paquete de trabajo A.1.1

Descripción del paquete de trabajo
Paquete de trabajo: A.1.2. Definición de las técnicas a aplicar. **Responsable:** Jose Ignacio Sánchez. **Duración estimada:** 4 **horas**
Descripción Tarea que consiste en estudiar el problema a abordar para determinar que técnica de minería de datos sería conveniente aplicar. **Entradas** Documento con el problema a abordar. **Salidas/Entregables** Propuesta de la técnica de minería de datos a aplicar. **Recursos necesarios** El sistema y los datos accesibles. **Precedencias** Definición del problema a abordar.

Tabla 2.22: Descripción del paquete de trabajo A.1.2

Descripción del paquete de trabajo
Paquete de trabajo: A.1.3. Selección de herramientas y tecnologías. **Responsable:** Jose Ignacio Sánchez. **Duración estimada:** 4 **horas**
Descripción Seleccionar que tecnologías utilizar para aplicar las técnicas propuestas. **Entradas** Documento con el problema a abordar y las técnicas a aplicar. **Salidas/Entregables** Propuesta de las tecnologías y herramientas de minería de datos a aplicar. **Recursos necesarios** PC con conexión a internet. **Precedencias** Definición las técnicas a aplicar.

Tabla 2.23: Descripción del paquete de trabajo A.1.3

Descripción del paquete de trabajo
Paquete de trabajo: A.2.1. Diseño de estrategias y algoritmos. **Responsable:** Jose Ignacio Sánchez. **Duración estimada:** 4 **horas**
Descripción Una vez conocido que se va a abordar y con que herramientas, es necesario diseñar una estrategia y decidir que algoritmos se van a utilizar. **Entradas** Documento de análisis. **Salidas/Entregables** Documento con los algoritmos a utilizar y el diseño de la estrategia. **Recursos necesarios** Herramienta y tecnologías de minería de datos a utilizar instaladas en el PC del responsable de la tarea. **Precedencias** Análisis.

Tabla 2.24: Descripción del paquete de trabajo A.2.1

Descripción del paquete de trabajo
Paquete de trabajo: A.3. Implementación. **Responsable:** Jose Ignacio Sánchez. **Duración estimada:** 40 **horas**
Descripción Tarea que consiste en implementar los algoritmos escogidos y las técnicas de evaluación de estos para conocer las medidas de desempeño. Además se implementará un banco de pruebas en el que observar resultados experimentales. **Entradas** Documento de diseño. **Salidas/Entregables** Implementación del módulo de minería de datos. **Recursos necesarios** Herramienta y tecnologías de minería de datos a utilizar instaladas en el PC del responsable de la tarea y acceso a la BD del sistema SaaS. **Precedencias** Diseño.

Tabla 2.25: Descripción del paquete de trabajo A.3

Descripción del paquete de trabajo
Paquete de trabajo: B.1.1. Definición del problema a abordar. **Responsable:** Josu Rodríguez. **Duración estimada:** 6 **horas**
Descripción Tarea que consiste en estudiar la información disponible y el negocio para definir que problema abordar para la aplicación de técnicas de *KDD*. **Entradas** SaaS. **Salidas/Entregables** Documento con la definición del problema a abordar. **Recursos necesarios** El sistema y los datos accesibles. **Precedencias** Pruebas.

Tabla 2.26: Descripción del paquete de trabajo B.1.1

Descripción del paquete de trabajo
Paquete de trabajo: B.1.2. Definición de las técnicas a aplicar. **Responsable:** Josu Rodríguez. **Duración estimada:** 4 **horas**
Descripción Tarea que consiste en estudiar el problema a abordar para determinar que técnica de minería de datos sería conveniente aplicar. **Entradas** Documento con el problema a abordar. **Salidas/Entregables** Propuesta de la técnica de minería de datos a aplicar. **Recursos necesarios** El sistema y los datos accesibles. **Precedencias** Definición del problema a abordar.

Tabla 2.27: Descripción del paquete de trabajo B.1.2

Descripción del paquete de trabajo
Paquete de trabajo: B.1.3. Selección de herramientas y tecnologías. **Responsable:** Josu Rodríguez. **Duración estimada:** 4 **horas**
Descripción Seleccionar que tecnologías utilizar para aplicar las técnicas propuestas. **Entradas** Documento con el problema a abordar y las técnicas a aplicar. **Salidas/Entregables** Propuesta de las tecnologías y herramientas de minería de datos a aplicar. **Recursos necesarios** PC con conexión a internet. **Precedencias** Definición las técnicas a aplicar.

Tabla 2.28: Descripción del paquete de trabajo B.1.3

Descripción del paquete de trabajo
Paquete de trabajo: B.2.1. Diseño de estrategias y algoritmos. **Responsable:** Josu Rodríguez. **Duración estimada:** 4 **horas**
Descripción Una vez conocido que se va a abordar y con que herramientas, es necesario diseñar una estrategia y decidir que algoritmos se van a utilizar. **Entradas** Documento de análisis. **Salidas/Entregables** Documento con los algoritmos a utilizar y el diseño de la estrategia. **Recursos necesarios** Herramienta y tecnologías de minería de datos a utilizar instaladas en el PC del responsable de la tarea. **Precedencias** Análisis.

Tabla 2.29: Descripción del paquete de trabajo B.2.1

Descripción del paquete de trabajo
Paquete de trabajo: B.3. Implementación. **Responsable:** Josu Rodríguez. **Duración estimada:** 40 **horas**
Descripción Tarea que consiste en implementar los algoritmos escogidos y las técnicas de evaluación de estos para conocer las medidas de desempeño. Además se implementará un banco de pruebas en el que observar resultados experimentales. **Entradas** Documento de diseño. **Salidas/Entregables** Implementación del módulo de minería de datos. **Recursos necesarios** Herramienta y tecnologías de minería de datos a utilizar instaladas en el PC del responsable de la tarea y acceso a la BD del sistema SaaS. **Precedencias** Diseño.

Tabla 2.30: Descripción del paquete de trabajo B.3

2.6. Planificación temporal

Con la estructura de descomposición de tareas se obtiene una estimación del tiempo a emplear por cada una de ellas, conocido esto es necesario decidir como distribuir la carga a lo largo del tiempo. La duración prevista en el tiempo para cada una de las tareas de muestra en el siguiente diagrama de *Gantt*, construido con la herramienta Open source *ganttproject*, mostrado en la figura 2.5.

Para que sea posible visualizar la distribución temporal, se muestra semanalmente. Como se trabajará tanto los días laborales como los festivos éstos se contemplan como día laboral.

El diagrama está basado en la *Estructura de Descomposición de trabajo*, respetando tanto cada una de las tareas y sus sub-tareas como las precedencias. En la izquierda del diagrama se encuentra la lista de tareas que se pretenden desarrollar a lo largo de todo el proyecto. La duración en semanas se presenta a la derecha de cada una de las tareas representada por la barra horizontal con el mismo color correspondiente en el *EDT*. El proceso lejos de seguir una secuencia en serie pretende unas restricciones tales que permita el trabajo en paralelo de algunas de las tareas.

Cabe mencionar que aunque se considera la memoria en sí, el documento final, aunque para entonces ya habrá partes escritas. A medida que avanza el desarrollo o se implementan nuevos cambios, se irán registrando en la memoria con la finalidad de no dejarse ningún cambio por registrar. Con esto se espera dedicar una pequeña porción de cada día a la redacción de la memoria .

La estimación se intentará ajustar a la realidad lo máximo posible, creando los bloques temporales en los que se presume se va a disponer del tiempo necesario a emplear en cada tarea.

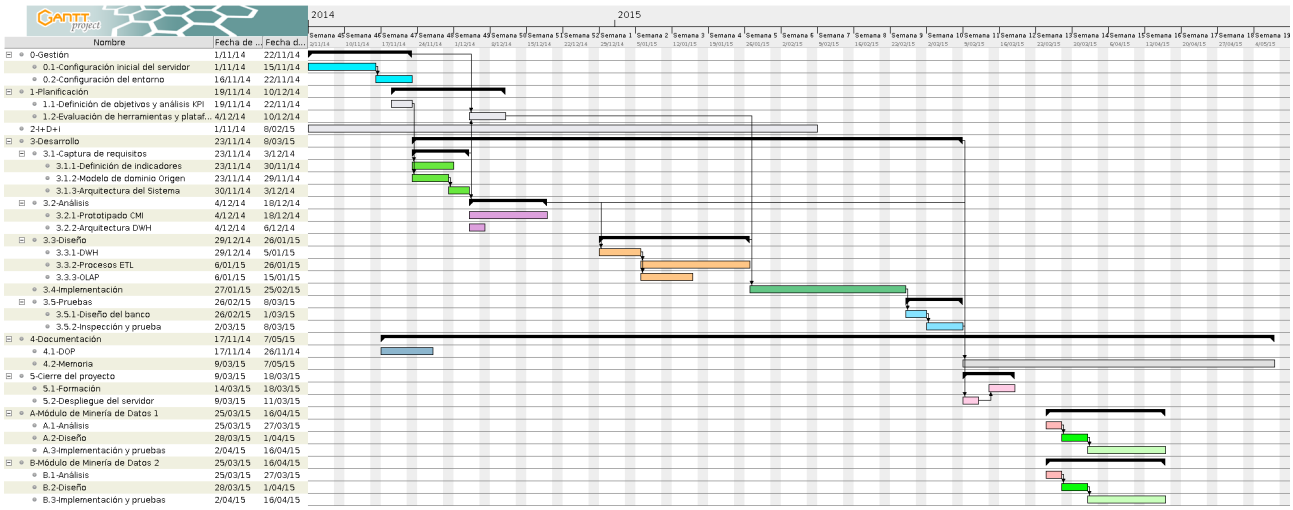


Figura 2.5: Diagrama de *Gantt*

Dado que en el proyecto hemos participado dos personas, esto suma a la complejidad del conjunto la tarea de coordinarse tanto en el tiempo como en volumen de trabajo, para ello también nos hemos valido de la herramienta *Ganttproject*, la cual permite gestionar los recursos y ver el porcentaje del tiempo total de la jornada que emplea de cada uno como puede verse en la figura 2.6.

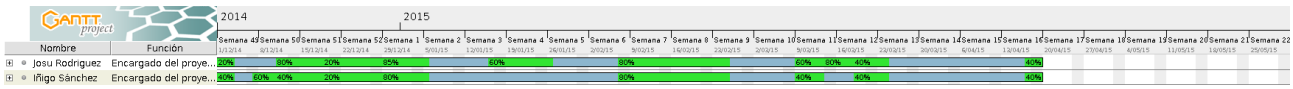


Figura 2.6: Diagrama de gestión de recursos

2.7. Evaluación de riesgos

En el presente apartado se analizarán los riesgos que puedan presentarse en el transcurso del proyecto, para ello se utilizará la siguiente escala de probabilidades:

- **Baja**: Probabilidad entre 0 - 33 %.
- **Media**: Probabilidad entre 34 - 63 %.
- **Alta**: Probabilidad entre 64 - 100 %.

2.7.1. Fallo electromecánico en el servidor

Análisis del riesgo de un fallo del hardware en el servidor.

Prevención

La prevención del riesgo se clasifica en dos secciones, la prevención de la pérdida de los datos contenidos dentro del servidor y la temprana detección de un fallo en el hardware.

Para prevenir la pérdida de datos se llevará a cabo una política de copias de seguridad basada en copias incrementales producidas de manera diaria y transmitidas a un servidor externo a media noche.

Es necesaria la monitorización del servidor mediante los programas que proporciona el sistema operativo contemplando las *temperaturas del procesador*, la memoria *RAM*, el *procesador gráfico* y los *discos duros* para comprobar que en ningún momento superen los umbrales recomendados por los fabricantes. Además, el disco duro será registrado de manera mensual en busca de sectores defectuosos, fallos en las cabezas lectoras y decrementos en sus rendimientos utilizando las herramientas ofrecidas por el fabricante.

De darse una situación anómala en alguno de estos registros el servidor será puesto a disposición del servicio técnico fuera del horario laboral de nuestra oficina para una revisión y para concretar los pasos para la resolución de un eventual problema.

Además, para minimizar el impacto sobre la producción que podría acarrear la pérdida del servidor contaremos con un servicio de emergencias consistente en un servidor remoto espejo que se activará en el momento en que el servidor no esté activo y que funcionará con las copias de seguridad proporcionadas.

Plan de contingencia

En caso de que se produzca un fallo electromecánico en el servidor se seguirán estos pasos:

1. Activación del servidor externo.
2. Comunicación a los empleados para que cambien la configuración de sus equipos para trabajar con el nuevo servidor.
3. Aviso al servicio técnico para evaluación de los daños.
4. Inicio de gestión administrativa para la gestión de la compra del reemplazo del material dañado.

Probabilidad

Media.

Impacto

La pérdida de las funciones que desarrolle en ese momento el servidor. Si además se ven afectados los discos duros, la pérdida parcial o total de la información contenida dentro del servidor.

El tiempo perdido por esta incidencia es de un día dado que las copias de seguridad solo pueden ser restauradas al día anterior.

2.7.2. Accidente laboral

Análisis del riesgo de un eventual accidente laboral en función de su gravedad.

Prevención

La empresa seguirá las recomendaciones recogidas dentro de la **OSHAS 18001** para la gestión de riesgos laborales emplazando bianualmente un inspector para la auditoría en esta materia dentro de la empresa.

Plan de contingencia

Para analizar la situación del accidentado y la gestión de sus necesidades evaluaremos el percance mediante el siguiente procedimiento:

- Si el accidente se ha producido en dependencias de la empresa:
  1. Evaluación de la necesidad de un traslado urgente a instalaciones sanitarias.
     a) Si se ve necesaria debido a la gravedad del accidente se procederá a llamar a una ambulancia.
  2. Aplicación de los primeros auxilios dentro de los conocimientos de los presentes.
  3. Realizar una llamada al seguro y a la mutua laboral para informar de la situación.
     a) Si el empleado así lo requiere o si hay un traslado hospitalario llamar al teléfono de contacto proporcionado por el empleado.
- Si el accidente se ha producido fuera de la empresa se realizará el paso número tres.

Probabilidad

Baja.

Impacto

Dependiendo de la gravedad se podría perder la producción del empleado mientras dure la incidencia o mientras esté de baja.

El tiempo perdido por esta incidencia es demasiado variable, desde unos minutos hasta la pérdida de todo el trabajo que realizaría el empleado hasta la nueva incorporación de otro empleado con todo lo que esto acarrea.

2.7.3. Bajas por enfermedad

Procedimiento ante la perspectiva de bajas por enfermedad.

Prevención

La empresa proveerá de un servicio de limpieza para la oficina que será explícitamente orientado a la limpieza del material informático periférico para evitar la transmisión de patógenos por vías cutáneas.

Además, en caso de enfermedad contagiosa de uno de los trabajadores que no requiera de una baja por su situación se proporcionará un gel antiséptico a disposición común con la recomendación de su uso frecuente mientras permanezca la situación.

Plan de contingencia

En caso de que se produzca una baja por enfermedad el gestor de la compañía se encargará de evaluar el impacto en función del tiempo estimado que dure la baja para estudiar la incorporación de un nuevo empleado temporal a la plantilla.

Si las pérdidas generadas por la baja son menores que el coste de contratar un nuevo empleado se desestimará la propuesta, sino se procederá a realizar la contratación.

Probabilidad

Media.

Impacto

En función del tiempo que dure la baja.

El tiempo perdido por esta incidencia es demasiado variable, desde unos días hasta la pérdida de todo el trabajo que realizaría el empleado hasta la nueva incorporación de otro empleado con todo lo que esto acarrea o hasta la recuperación del trabajador.

2.7.4. Fallos en el suministro de electricidad

Ante un fallo en el suministro eléctrico.

Prevención

La instalación de nuestra oficina deberá de estar previamente dotada con dispositivos *SAI* con un tiempo de batería suficiente para prevenir los daños que esta incidencia pudiera provocar sobre el trabajo en curso.

Los técnicos de la compañía eléctrica certificarán de manera anual que la instalación eléctrica no ha sufrido daños por la fatiga temporal.

Plan de contingencia

En caso de que se produzca esta contingencia se procederá a dar aviso a todos los empleados de que deben almacenar todo el trabajo que están realizando en ese momento y deberán apagar sus equipos.

Se procederá a llamar a la compañía suministradora para informar del suceso y para informarnos sobre el carácter de la incidencia. Si el una incidencia cuya resolución implique un tiempo superior al de la jornada restante, se procederá a realizar una reunión para abordar un posible cambio de horarios entre los trabajadores con el fin de intentar recuperar el tiempo perdido.

Además, se iniciará la gestión de los seguros concernientes para una posible indemnización por los daños sufridos.

Probabilidad

Baja.

Impacto

Durante el tiempo que dure el corte del suministro el trabajo se paraliza por completo debido a las características de la empresa.

2.7.5. Caída de red

En el caso de que la empresa se quede sin conexión a Internet.

Prevención

La empresa garantizará un proveedor de red que se comprometa a mantener un alto *uptime* y un servicio técnico que tenga experiencia en servicios técnicos inmediatos. Además, el equipamiento de red de la empresa se inspeccionará de manera anual para evaluar defectos.

Plan de contingencia

Se procederá a reportar la contingencia a la compañía proveedora para que provea los medio técnicos para la reparación de la red.

Probabilidad

Media.

Impacto

La caída en la conexión a Internet provocará un decremento de la productividad de los trabajadores puesto que dejarán de disponer de las herramientas de ayuda que proporciona esta herramienta como forma de consulta y de solución de problemas.

2.7.6. Fallo simultáneo de varios equipos

Evaluación del riesgo de fallos simultáneos en los equipos de la empresa.

Prevención

La empresa evaluará cada equipo de la empresa de manera trimestral para detectar posibles incidencias que afecten a los equipos y, en cualquier caso, mantendrá un número de ordenadores igual a uno por cada cinco empleados de reserva.

Plan de contingencia

Se asignará un ordenador de emergencia a cada empleado con un ordenador dañado. En caso de que no hubiera suficientes ordenadores disponibles procederíamos a preevaluar un número idéntico al número de ordenadores faltantes identificando, a grosso modo, aquellos equipos que no presenten posibilidad de reparación. Una vez identificados, se procederá a la compra de un número de ordenadores idénticos a los identificados.

Simultáneamente, se procederá a llamar al servicio técnico para evaluar los daños e identificar una posible solución.

Probabilidad

Baja.

Impacto

La posible pérdida de la información en los equipos afectados y la pérdida de la productividad en aquellos empleados que afecte la incidencia.

La pérdida económica que supone el reemplazo de los equipos dañado.

2.7.7. Cambio en las especificaciones del proyecto

Cambio en las especificaciones iniciales del proyecto.

Prevención

Las especificaciones del proyecto plantean un producto escalable y abierto a cambios por lo tanto este riesgo no debería plantear ninguna problemática insalbable.

Plan de contingencia

Replantear las especificaciones y reconducir el proyecto.

Probabilidad

Alta.

Impacto

Inexistente.

2.7.8. Planificación temporal incorrecta

La posibilidad de que se desvíe considerablemente la estimación temporal del proyecto.

Prevención

Ajustar la estimación a la realidad lo máximo posible.

Plan de contingencia

Ajustar la planificación temporal reduciendo el tiempo las tareas lo máximo posible.

Probabilidad

Alta.

Impacto

Aumento en los costes del proyecto y retraso en la entrega final.

2.7.9. Resumen

Después de analizar todos los riesgos que se consideran posibles de ocurrir, en la tabla 2.7.9 recoge un resumen del análisis de riesgos realizado en los anteriores apartados.

Riesgo	Probabilidad	Impacto
Fallo electromecánico del servidor	Media	Pérdida de datos total o parcial
Accidente laboral	Baja	Disminución temporal de la productividad
Bajas por enfermedad	Media	Disminución temporal de la productividad
Fallos en el suministro eléctrico	Baja	Se paraliza por completo el desarrollo del proyecto
Caída de red	Media	Perdida de los recursos de internet
Fallo simultaneo de varios equipos	Baja	Aumento del coste final.
Cambio en las especificaciones del proyecto	Alta	Ninguno
Planificación temporal incorrecta	Alta	Aumento del coste final y retraso en la entrega.

Tabla 2.31: Tabla resumen de riesgos

2.8. Evaluación económica

Con la estimación temporal se obtiene como resultado el diagrama de *Gantt* y las horas estimadas de cada tarea que nos permitirá realizar una evaluación económica lo más ajustada posible a la realidad. Los cálculos están basados en los precios que se presentan en la tabla de la figura 2.7, en la cual también se muestran los resultados de la evaluación. El **coste total del proyecto** asciende a 18872.8€para el cliente que lo solicitó. Para su desarrollo será necesaria una **inversión** de unos 6000 €por parte de la empresa.

La empresa obtendrá un **beneficio** de 4000 €por la realización del proyecto. El **ROI** se muestra a continuación, calculado mediante la siguiente fórmula en base a los datos de la hoja de cálculo:

$$ROI = \frac{Coste\ total\ al\ comprador - Coste\ total\ del\ proyecto}{Coste\ total\ del\ proyecto} * 100 = 26{,}9\,\%$$

A continuación se listan los recursos tenidos en cuenta para realizar la evaluación económica:

- **Trabajador**: El coste es el correspondiente a una semana completa de dedicación al proyecto. Por cada semana se ha estimado la dedicación de cada trabajador al proyecto (en tanto por 1) y, posteriormente, se ha sumado la dedicación de los 2 trabajadores para obtener el total por semana. En las semanas en que los dos componentes del equipo dedican todo su tiempo al proyecto, el valor es máximo, siendo éste de 2.
- **Equipo de desarrollo**: Incluye la amortización del equipo individual del trabajador. Se ha calculado la amortización por semana en base al precio de cada equipo, 500 €, y su tiempo de vida estimado, 4 años. Como en el caso anterior, se ha calculado el número de equipos utilizados por semana en base al uso que se espera de los mismos en el proyecto.
- **Servidor datos**: Servidor propio de la empresa utilizado para el almacenamiento de los datos relacionados con los proyectos. Incluye la gestión del código fuente de la aplicación durante el desarrollo del proyecto. Se incluye su amortización, partiendo de su precio de compra, 1000 €, y su duración estimada, 5 años. Su mayor uso se da durante la implementación de la aplicación.
- **Local**: Costes del alquiler. El cobro se realiza a fin de mes.
- **Servidor en la nube**: Al empezar con la fase de pruebas, será necesario realizar la instalación en un servidor accesible desde cualquier lugar. De esta forma se podrá emular la situación real de funcionamiento de la aplicación en el propio entorno de desarrollo. El precio al mes estimado asciende a unos 4 €.

Quedan acordados los pagos intermedios marcados en la tabla de la evaluación económica 2.7. Se efectuarán 6 pagos intermedios, en las semanas 46 y 49 del año 2014 y las semanas 2, 5, 10 y 13 del año 2015 de forma que se de soporte al periodo de desarrollo del producto. Finalmente se cobrará la cantidad restante.

Recursos	Coste Unidades	45	46	47	48	49	50	51	52
Trabajador	20 €/ hora	7	4	13	32	18	15	15	15
Equipo de desarrollo	2,5 €/ semana	2	2	2	2	2	2	2	2
Servidor datos	3,7 €/ semana	1	1	1	1	1	1	1	1
Licencias	0 €/ mes								
Local	75 €/ semana	1	1	1	1	1	1	1	1
Servidor	1 €/ semana	1	1	1	1	1	1	1	1
Consumo eléctrico	10 €/ semana	1	1	1	1	1	1	1	1
Formación	100 €/ semana								
Flujo de pagos		234,7	174,7	354,7	734,7	454,7	394,7	394,7	394,7
Flujo de ingresos			1500			1500			
Flujo de caja		-234,7	1325,3	-354,7	-734,7	1045,3	-394,7	-394,7	-394,7
Acumulado		-234,7	1090,6	735,9	1,2	1046,5	651,8	257,1	-137,6

Inversión necesaria:	234,7
Coste total del proyecto:	3137,6
Beneficios a alcanzar:	1500
Coste total al comprador:	4637,6

Recursos	Coste Unidades	1	2	3	4	5	6	7	8
Trabajador	20 €/ hora	20	25	45	45	40	40	40	40
Equipo de desarrollo	2,5 €/ semana	2	2	2	2	2	2	2	2
Servidor datos	3,7 €/ semana	1	1	1	1	1	1	1	1
Licencias	0 €/ mes								
Local	75 €/ semana	1	1	1	1	1	1	1	1
Servidor	1 €/ semana	1	1	1	1	1	1	1	1
Consumo eléctrico	10 €/ semana	1	1	1	1	1	1	1	1
Formación	100 €/ semana								
Flujo de pagos		494,7	594,7	994,7	994,7	894,7	894,7	894,7	894,7
Flujo de ingresos			1500			1500			
Flujo de caja		-494,7	905,3	-994,7	-994,7	605,3	-894,7	-894,7	-894,7
Acumulado		-494,7	410,6	-584,1	-1578,8	-973,5	-1868,2	-2762,9	-3657,6

Inversión necesaria:	3657,6
Coste total del proyecto:	6657,6
Beneficios a alcanzar:	1500
Coste total al comprador:	8157,6

Recursos	Coste Unidades	9	10	11	12	13	14	15	16
Trabajador	20 €/ hora	16	20	20	26	26	18	45	45
Equipo de desarrollo	2,5 €/ semana	2	2	2	2	2	2	2	2
Servidor datos	3,7 €/ semana	1	1	1	1	1	1	1	1
Licencias	0 €/ mes								
Local	75 €/ semana	1	1	1	1	1	1	1	1
Servidor en la nube	1 €/ semana	1	1	1	1	1	1	1	1
Consumo eléctrico	10 €/ semana	1	1	1	1	1	1	1	1
Formación	100 €/ semana								
Flujo de pagos		414,7	494,7	494,7	614,7	614,7	454,7	994,7	994,7
Flujo de ingresos			1500			1500			
Flujo de caja		-414,7	1005,3	-494,7	-614,7	885,3	-454,7	-994,7	-994,7
Acumulado		-414,7	590,6	95,9	-518,8	366,5	-88,2	-1082,9	-2077,6

Inversión necesaria:	2077,6
Coste total del proyecto:	5077,6
Beneficios a alcanzar:	1000
Coste total al comprador:	6077,6

ROI	26,894734 %

Figura 2.7: Evaluación económica

Capítulo 3

Captura de requisitos

En este capítulo presentaremos las principales problemáticas en las que se centrará el software a diseñar. Su estudio permitirá posteriormente la creación de un sistema de *Business Intelligence* que proporcionarán información esencial para la toma de decisiones en la empresa. Del presente análisis se obtendrá la información inicial clave que conducirá al equipo de desarrollo hacia las respuestas que el cliente necesita responder para conocer mejor su situación actual y gestionar de una manera eficaz su actividad empresarial.

3.1. Productividad

Siendo este un sistema implementado para su uso en una empresa, el rendimiento de la misma será siempre uno de los factores a tener en cuenta.

Concretamente, tratándose de una empresa en la que la producción depende en su mayor parte de la efectividad laboral de los empleados, el análisis de la productividad se centrará en estos, utilizando la producción por hora y el número de servicios como medida de referencia.

Tal y como ha sido mencionado, el eje del análisis será el rendimiento de las horas de trabajo por cada servicio que los empleados realizan.
Este análisis permite una gran flexibilidad ya que se puede extender a otras dimensiones analíticas como las temporales (p.ej.: análisis semanal, mensual, anual...), y profundizar o generalizar por empleados únicos, departamentos o todo el conjunto del personal de la empresa.

A este análisis también se le pueden ser sumados los costes de la actividad con facilidad, utilizando elementos como el precio de materiales de reparación, piezas de sustitución, gastos de desplazamiento y demás costes implicados en los servicios de la empresa.

Se realizará un análisis de la productividad por hora desglosada temporalmente y por trabajador. Estas dos dimensiones del análisis podrán variar en su nivel de profundidad analítica: el tiempo podrá ser analizado entre semanas, meses, trimestres y años, y los empleados, a su vez, entre empleados individuales, departamentos y el conjunto completo de los mismos.

Como medida se utilizarán los ingresos por hora como indicador básico. A su vez, se creará un indicador secundario generado a partir de este último, que serán los beneficios por hora de trabajo. Este análisis conlleva un mayor procesamiento debido a que cada servicio tendrá diferentes variables que afectarán al coste del mismo, lo que a su vez hará relevante un análisis de la rentabilidad de diferentes tipos de servicios.

3.2. Aumento/Reducción de ingresos

No es posible entender la situación actual de una empresa sin introducirnos en el análisis de la economía de la misma, el cual nos va a permitir tomar decisiones estratégicas, como por ejemplo lanzar ofertas de contratos de mantenimiento entre determinados segmentos de clientes o dejar de prestar servicio en localizaciones en las que el número de servicios prestados en horas no es capaz de soportar el coste de mantener tal servicio.

El enfoque de este análisis se realizará por cliente y localización. Analizaremos los mejores y los peores clientes que tiene la empresa en diferentes rangos temporales y permitirá la exploración entre la totalidad de la cartera.

Además es conveniente presentar un análisis global a lo largo del tiempo que permita ver cuál es la evolución de la empresa.

Nos centraremos en medir los ingresos proporcionados cada cliente y la evolución global a lo largo del tiempo, así como la evolución de la empresa a este respecto.

Para profundizar en el análisis y poder detectar por provincias y localidades cambios en la evolución de los ingresos o el número de servicios prestados, incluso la correcta correspondencia entre ambos, se pretende realizar el análisis por población.

Los principales indicadores los encontramos en formade ingresos realizados por cliente y población. También se analizarán los servicios prestados por población y provincia. Dada la importancia de estos datos, se obtendran diariamente para poder detectar cambios en la mayor brevedad posible.

3.3. Personal desmotivado

Este análisis fue propuesto por el cliente. Quién nos hizo saber que existía un problema de desmotivación entre los empleados de la plantilla que merecía especial interés, que debía ser tratado como un problema inde-

pendiente y no formando parte de ningún otro indicador como medida.

A este respecto el análisis fue apoyado en todo momento por el cliente, quién facilitó la tarea de encontrar que indicadores utilizar y bajo que ejes realizar el análisis.

La petición del cliente consistía en que pudiesen analizar cuanto de su trabajo realizado se invertía en actividad facturable y cuanto no lo es.

El equipo por su parte propuso medir el número de ordenes de trabajo por realizar y cuales de éstas son urgentes. Otra medida consiste en el número de horas invertidas en las ordenes de trabajo realizadas.

Nos centraremos en facilitar la gestión individual del trabajo, para que cada uno de los técnicos de la plantilla disponga de la información suficiente que mejore el reparto del tiempo que le dedica a los trabajos facturables y a los contratos de mantenimiento.

Los principales indicadores para este análisis fueron indicados por el cliente sumados a los propuestos por el equipo de analistas del proyecto. Para seguir las indicaciones se analiza el número de horas invertidas en los contratos de mantenimiento por que es la primera aproximación que nos permite el conjunto de datos que recoge la herramienta ERP que utiliza la empresa actualmente.

Como iniciativa para optimizar el análisis, siendo el proveedor de la herramienta ERP el mismo proveedor que financia el proyecto que nos ocupa, se propone incluir un campo de control en las ordenes de trabajo que permitirá identificar de manera unívoca las horas invertidas no facturables, es decir que forman parte de alguna garantía, contrato de mantenimiento o cualquier otra actividad por la que la empresa no genera ingresos.

3.4. Disminución de los servicios prestados

Este análisis se centrará en los artículos y las marcas que actualmente trabaja el cliente.
Se analizarán cual es el volumen de reparaciones y horas invertidas en cada artículo, para así permitir tomar decisiones estratégicas a este respecto.

El enfoqué estará dirigido en medir el trabajo invertido por cada artículo y cada categoría en los que la empresa los clasifica.

Los principales indicadores utilizados son el número de horas invertidas en las reparaciones de los artículos por categoriía y en un rango de fechas.

3.5. Variables y desglose

En las tablas que se muestran en la figura 3.1, quedan recogidos los diferentes indicadores (*KPI*) en los que se basará el análisis posterior.

Variables de productividad		
Concepto	**Desglose**	**Comparación**
Cumplimiento horas previstas	Medición por empleado	Media global
	Medición por servicio	Media global
	Medición por cliente	Detalle
	Medición por población	Detalle
	Medición por departamento	Media global
	Medición por estado	Total
horas invertidas por contrato	Medición por empleado	Total

Variables de rendimiento económico		
Concepto	**Desglose**	**Comparación**
Ingreso por servicio		
por cliente	Medición semanal	Objetivo
por empleado	Medición mensual	Medición mensual
por artículo	Medición mensual	Medición mensual
por población	Medición mensual	Mes anterior
Beneficio por hora de trabajo		
por empleado	Medición mensual	Media acumulada del año actual
por servicio	Medición a cierre de proyecto	Media anual de todos los proyectos
Rentabilidad de contratos anuales		
Ingresos por hora trabajada	Medición anual	Media ingresos/hora en contratos anuales
	Medición anual	Media ingresos/hora sin contrato anual
Beneficios por hora	Medición anual	Media ingresos/hora en contratos anuales
	Medición anual	Media ingresos/hora en sin contrato anual
Inversión en formación	gasto anual	años anteriores
Inversión en I+D+I	gasto anual	años anteriores

Variables de satisfacción del cliente		
Concepto	**Desglose**	**Comparación**
Eficacía		
Desvío temporal de la duración del servicio.	Medición en cada servicio realizado (horas)	Media de servicios mismo artículo

Variables del personal		
Concepto	Desglose	Comparación
Satisfacción	Resultado anual	Año anterior
	Encuesta anónima	Año anterior
Cumplimiento del horario	acumulado semanal	Semanas anteriores
	horas registradas	
Horas extra invertidas	Número de horas extra mensuales por empleado	Media global
Número de bajas	acumulado mensual	Meses anteriores
	informe de la baja	

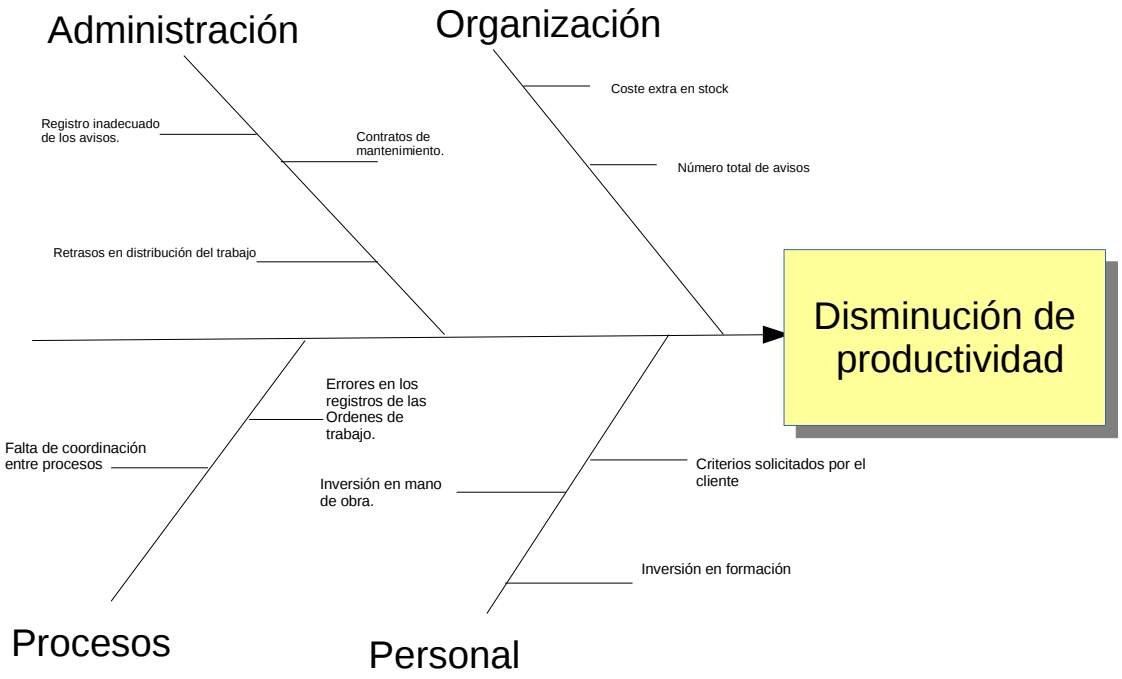
Figura 3.1: Desglose de variables

3.6. Análisis de causas y efectos

En el presente apartado se ddocumenta el análisis causa-efecto que se realiza con la problemática a estudiar extraida de las reuniones con el cliente.

3.6.1. Productividad

Dada la naturaleza de los procesos en los que la empresa realiza su actividad económica, es necesario analizar minuciosamente el trabajo que realiza cada empleado para ser capaces de obtener un análisis de la productividad de la empresa en su conjunto. De esta forma podremos tener una visión de cómo está funcionando la empresa y detectar cuando se está dando un mal funcionamiento, debido a la sobrecarga de trabajo o el extremo contrario.

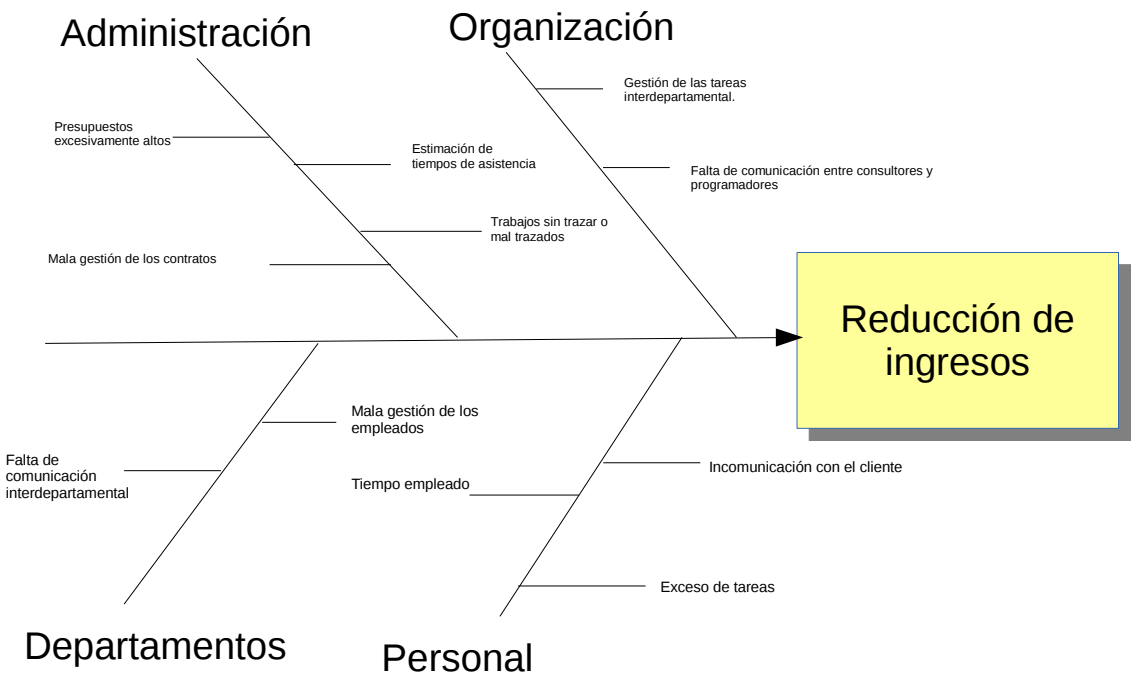


Las variables a tener en cuenta que nos permitirán medir la productividad de una manera lo suficientemente precisa que nos permita tener el control sobre la problemática aquí descrita, se encuentran recogidas en la primera tabla de la figura 3.5 *Variables de productividad*.

Además se tendrá especial cuidado en tracear los retrasos y los servicios urgentes.

3.6.2. Aumento/Reducción de ingresos

El sentido intrínsenco de toda actividad empresarial se basa en los rendimientos económicos que es capaz de generar a medio o largo plazo. De ahí surge la necesidad de desarrollar el análisis de la economía de la empresa, que permitirá explorar a través de la situación económica real de la empresa en un periodo determinado, o incluso la evolución a lo largo del tiempo, y así poder tomar decisiones estructuradas que permitan mantener políticas económicas sostenibles.

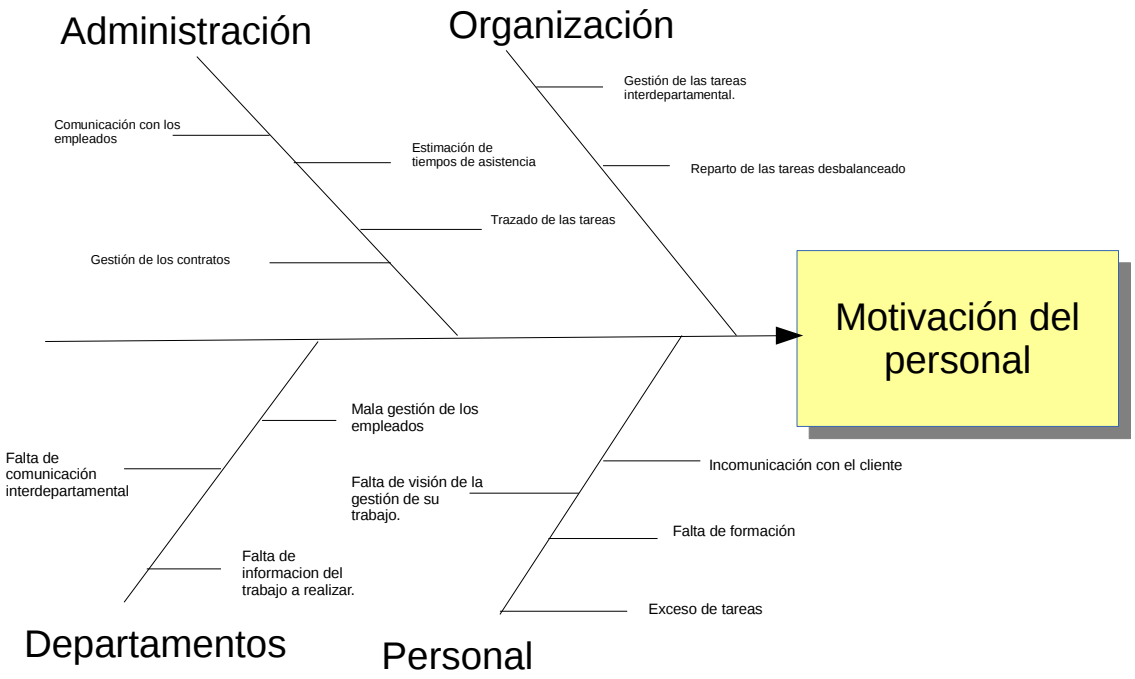


Para el análisis financiero se dispone de las variables de rendimiento económico de la empresa con la periodicidad de medición diaria, dado que la actualización del sistema tiene asignada esta frecuencia para evitar interferir en la actividad diaria de la empresa.

Si vamos a la figura 3.5, podremos ver las variables que se recogen para éste análisis en la tabla bajo el título *Variables de rendimiento económico*.

3.6.3. Personal desmotivado

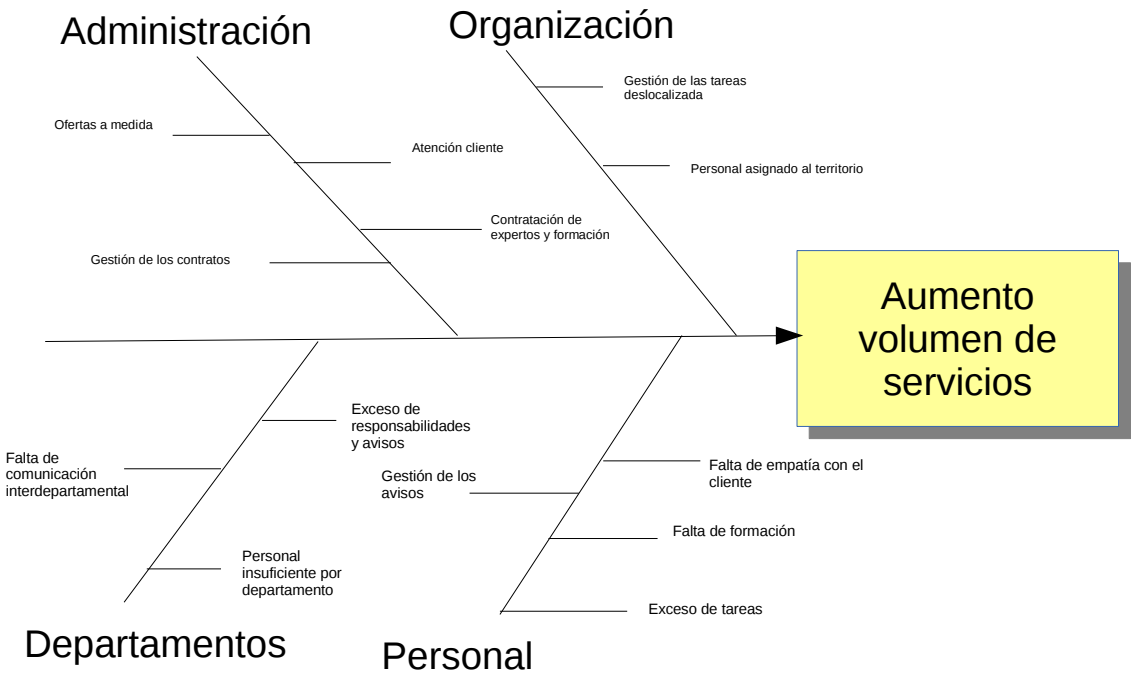
Como se ha mencionado en el apartado 3.1, éste análisis ha sido motivado por el cliente y llevado acabo con su soporte. La preocupación rádica en que percibe al personal desmotivado. Aunque desconoce cuales pueden ser las causas reales, presume intuir que se trata de una falta de visión de los rendimientos de trabajo que generan cada uno de ellos.



Para resolver éste análisis se hace uso principalmente de las variables dispuestas en la tabla *Variables de productividad* y *Variables del personal* presentadas en la figura 3.5

3.6.4. Volumen los servicios prestados

Si los rendimientos dinerarios son importantes para determinar la salud económica de una empresa, no lo es menos el volumen de actividad que es capaz de generar y gestionar. Para facilitar el análisis, se permitirá explorar a través de la información, para determinar en qué territorios genera más actividad o como evoluciona en aquellos en los que desarrolla de una manera "establecida" su actividad.



La definición de las variables a tener en cuenta resulta de utilizar aquellas que nos miden y comparan la productividad y la actividad económica. Para apoyar dicho análisis se tendrán en cuenta las variables recogidas en las tablas *Variables de productividad*, *Variables de rendimiento económico* y aquellas que nos permiten medir la satisfacción de los clientes y que quedan recoguidas en la tabla *Variables de satisfacción del cliente*.

Capítulo 4

Análisis y Diseño

Capítulo dedicado al análisis y diseño de todo el sistema. Desde la ingeniería inversa para obtener la documentación de la que no disponíamos como los diseños a los que se converge gracias al análisis previo y a las reuniones con el equipo del proyecto y el cliente.

4.1. Fuentes de datos origen

Las fuente de datos origen en las dos empresas que explotan el sistema actualmente, es el *ERP* que utilizan para gestionar sus respectivas actividades, el cual utiliza un sistema de base de datos relacional americano de la compañía *Progress*.

4.1.1. Diseño

Con la finalidad de conocer mejor la fuente de datos origen, ya que no la conociamos previamente, se aplican técnicas de ingeniería inversa para obtener el modelo de entidad relación.El resultado fue sorprendente, encontramos que, entre otras dificultades, no disponen ni siquiera de *Foreign keys*. Este problema que se presenta en un sólo parrafo ha supuesto un gran desafío a lo largo del proyecto ya que nos ha obligado a realizar una labor de investigación a través del cliente y el consultor, para ser capaces de comprender la realidad que representa el modelo lógico de la base de datos.

En la figura 4.1 se muestra el resultado de aplicar la ingeniería inversa y resolver la inexistencia de restricciones de integridad referencial de clave foránea.

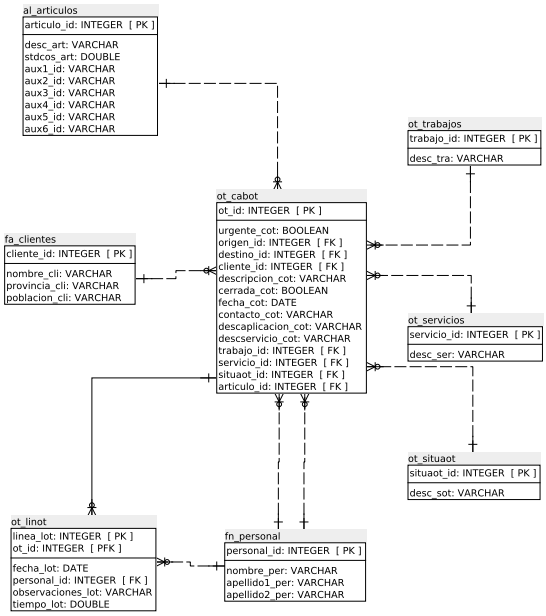


Figura 4.1: Ingeniería inversa

4.1.2. Aspectos relevantes

Cabe destacar que la fuente de datos original utilizada por las empresas clientes de este sistema ha conllevado algunos problemas debido a la implementación de la base de datos. El mayor escollo radica en el problema de la inexistencia de claves foráneas definidas anteriormente mencionado, dificultando así la búsqueda de relaciones entre tablas.

Este hecho queda agravado por la ausencia de consistencia entre diferentes campos que representan claves a lo largo de las tablas que les hacen referencia.

Otro problema hallado ha surgido de una característica de la implementación de una base de datos Progress: en los campos de este sistema, pese a que se defina un tamaño, se puede exceder el mismo sin que haya problemas. Sin embargo, hay otro atributo que es el tamaño máximo de campo para SQL, y cuando se trata de extraer datos vía JDBC, el campo sí debe respetar el tamaño. Al ejecutar los procesos ETL, ha surgido este problema con varios campos y ha sido necesario modificar el tamaño máximo para SQL de los mismos.

4.2. Data Warehouse

Un Data Warehouse es un sistema que extrae, limpia, ajusta y dispone los datos en un almacén de datos dimensional que ofrece soporte de consulta y análisis en el proceso de toma de decisiones.[4]

En el diseño de la arquitectura y la lógica de un **data warehouse** corporativo, es necesario partir de una serie de características:

- Administra grandes cantidades de información.
- Mantiene un histórico.
- Condensa y agrega información.
- Integra y asocia información de varias fuentes.

Generalmente se realizan una serie de procesos ETL, para obtener un modelo multidimensional y así poder realizar consultas analíticas de manera más óptima.

Si las dimensiones necesitan ser estructuradas en diferentes niveles de granularidad es posible definir jerarquías con las dimensiones. Por ejemplo la jerarquía para la fecha podría ser "día, semana, mes, año".

Este tipo de modelos generalmente consta de dos elementos:

- **Dimensiones**: Estructura que define los hechos y las medidas para capacitar a los usuarios a responder preguntas sobre su organización.
- **Hechos**: Son el objeto de los análisis y están relacionados con las dimensiones. Modelan un hecho real del negocio (por ejemplo una venta).

Los hechos contienen los datos de estudio y las dimensiones los metadatos.

Las jerarquías de las dimensiones presentan relaciones n-1 de manera que un valor de un nivel sólo puede ser agrupado por un único valor de cada nivel inmediatamente superior en la jerarquía. Esto facilita de manera rápida y sencilla el profundizar en el nivel de detalle (drill-down), disminuir el detalle(roll-up) que soncaracterísticos de informes creados a partir de un data warehouse. En la figura 4.2 se presenta un ejemplo sencillo de un datamart 1.4 multidimensional.

Figura 4.2: Cubo multidimensional.

Al fin y al cabo un *DWH* (1.4) puede considerarse como la unión de un diseño iterativo de diferentes *data-marts*.

En una empresa, sobre todo cuando hay múltiples data marts, es muy importante analizar las dimensiones previamente con las personas clave de las fuentes de datos y las usuarios clave de los datos finales además de revisar los diseños existentes de los que nos podremos valer, ya que las fuentes condicionarán los datos a proporcionar y las dimensiones se deberían reutilizar en las distintas tablas de hechos de distintos data mart. Una dimensión fecha debería tener el mismo diseño en todos los data marts, porque si no cuando se integren será mucho más complejo. Es una decisión tanto técnica como política.
Los hechos deben de tener unidades de medidas uniformes: mismas localizaciones, unidades de tiempo o moneda, si no fuera así no se podría definir correctamente un hecho único.

Es importante no trabajar con claves significativas, para evitar problemas, por si existe cambios en el futuro.

Un modelo multidimensional que no tiene jerarquías, se denomina modelo en estrella, si tuviera jerarquías, se denominaría modelo copo de nieve, vease la figura 4.3.

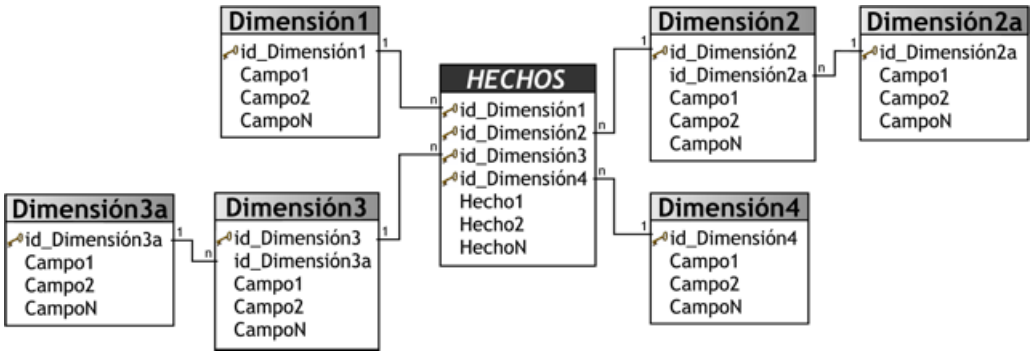


Figura 4.3: Modelo copo de nieve DWH

[1]

Por las características del módelo estrella,vease la figura 4.4, se ha optado por este diseño para la implementación del modelo lógico del *DWH* corporativo.
[2]

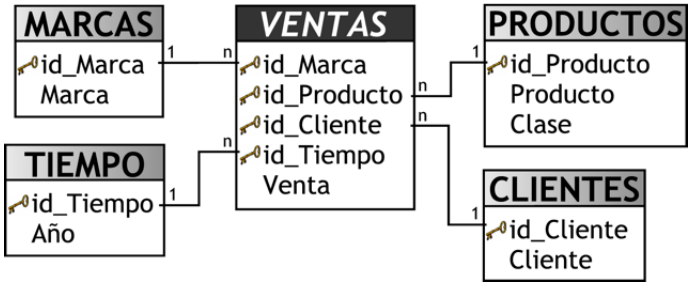


Figura 4.4: Modelo estrella DWH

[3]

4.2.1. Arquitectura

Para la elección de la arquitectura que se utilizará para implementar el *DWH*, se realizaron pruebas de carga en la red de datos, con los resultados de estas pruebas se tomo la decisión de prescindir de la denominada *staging area* 1.4 por que la red soporta la frecuencia y el volumen de actualización necesarios.

La arquitectura a implementar se muestra en la figura 4.5, el sistema obtendrá los datos de la fuente de origen, implementará los procedimientos *ETL* (1.4) necesarios para cargar los *datamarts* correspondientes en cada iteración, todos ellos juntos conforman el denominado *Data Warehouse Corporativo*.

[1] Imagen extraida de http://www.dataprix.com/data-warehousing-y-metodologia-hefesto/arquitectura-del-data-warehouse/34-datawarehouse-manager
[2] El tiempo de respuesta del servidor es más rápido debido a que se involucran menos tablas en los querys.
[3] Imagen extraida de http://www.dataprix.com/data-warehousing-y-metodologia-hefesto/arquitectura-del-data-warehouse/34-datawarehouse-manager

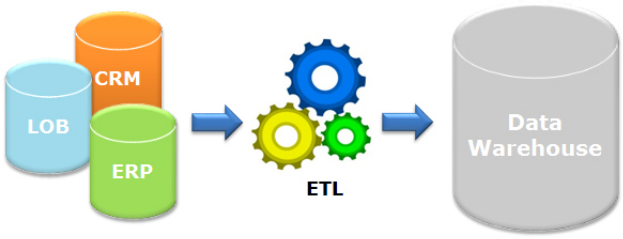


Figura 4.5: Arquitectura DWH

[4]

4.2.2. Diseño

Una característica muy importante en el diseño de un DWH es el factor granularidad. De hecho esta característica será permeada a todo la arquitectura del sistema. La granuralidad hace referencia a la atomicidad o nivel de detalle máximo al que se hace referencia. Por ejemplo se refiere a un nivel bajo de granularidad cuando se determina una única asistencia a domicilio como grano, mientras que el resumen de las asistencias realizadas el último mes representa un nivel más alto.

La granuralidad es un aspecto crítico en el diseño y esto es porque determina el volumen de datos a albergar y el tipo de preguntas que se podran realizar.
La clave para la reusabilidad en proyectos de gran embergadura puede residir en el grano implementado ya que la misma información puede ser reutilizada por diferentes departamentos de la organización: marketing, ventas, contabilidad. . . etcétera.

En este caso se a determinado la atomicidad máxima para ser capaces de llegar a analizar con el mayor detalle posible cada una de las transacciones. Conocido el volumen del tráfico de datos que generan los grupos de interés a los que se dirige el producto,se determina registrar con la mayor atomicidad posible, permitiendo un análisis profundo sin un coste de almacenamiento muy alto. Con esta decisión se aporta el valor de obtener en detalle cualquier cambio o anomalía detectada, por pequeña que esta sea.

Para modelar el problema que nos ocupa se ha optado por un diseño en estrella que agiliza las consultas en el servidor reduciendo el tiempo de carga y refresco. Actualmente se han realizado dos iteraciones que han permitido construir un data warehouse compuesto por tres tablas de hechos, representando cada una de elllas un datamart o cubo:

Las tablas que representan los hechos de negocio a analizar se listan a continuación:

- **fact_notice:** Esta tabla de hechos es reutilizable para distintos hechos de negocio en función de la actividad de la empresa a modelar. En el caso de las dos empresas que actualmente explotan el sistema, representa avisos para realizar asistencia técnica u ordenes de trabajo a realizar.
- **fact_purchase_receipt:** Esta tabla representa un hecho presente en cualquier realidad empresarial: **Compras** o facturación por gastos.
- **fact_sales_receipt:** Otra representación presente en todos y cada uno de los negocios presentes en cualquier mercado: **Ventas**.

El resto de tablas representan los metadatos del hecho modelado, es decir las dimensiones:

- **dim_date:** Es una dimensión estática, esto quiere decir que se carga una vez y no es necesario diseñar ninguna estrategia de actualización. Representa el tiempo; en este caso se ha optado por un diseño con una granularidad total que permite la máxima flexibilidad en cuanto análisis temporal se refiere.
- **dim_receipt:** Dimensión que representa la facturación, en este caso se ha optado por utilizar una única dimensión para representar tanto las compras como las ventas, para diferenciar unas de otras se ha optado por utilizar un atributo de tipo boolean, **is_purchase**, que determina cual de las dos transacciones representa.
- **dim_receipt_line:** Esta dimensión representa cada una de las líneas de las facturas.

[4]Imagen extraida de http://www.arcplan.com/en/blog/tag/etl/

- **header_to_line_bridge:** Tabla auxiliar que une dim_receipt y dim_line_receipt, optimizando la consulta utilizada en el caso de las tablas para obtener el detalle de la facturación, sin necesidad de pasar por la tabla de hechos con una sentencia **INNER JOIN** o estrategia similar.
- **dim_staff:** Dimensión que representa a los empleados de la organización.
- **dim_service:** Dimensión que puede describir tanto un servicio prestado como una orden de trabajo, en función de cual de las empresas esté obteniendo la información.
- **dim_service_line:**Representa las líneas de un servicio o dim_service.
- **dim_article:** En esta dimensión se encuentran los artículos involucrados en la actividad económica empresarial, tanto a nivel de ventas como los que se utilizan en las reparaciones o sustituciones.
- **dim_customer:** Esta dimensión representa a los clientes de la empresa.
- **dim_partner:** Los proveedores son estudiados en el análisis a través de esta dimensión.

4.3. Procedimientos de Extracción, Transformación y Carga

Se estima que el 70 % del tiempo se emplea en transformar el formato de las fuentes de datos origen en un formato dimensional.[2]

Se puede apreciar en el modelo de un sistema BI presentado en la figura 4.6, como un entorno Data Warehouse está compuesto por diversos componentes, cada uno de ellos con sus propias herramientas, técnicas, *suits* de diseño y productos. Por separado ninguna de ellas es un Data Warehouse. Las etapas y herramientas dedicadas a los procesos de extracción, transformación y carga se pueden considerar las más importantes y sofisticadas.

Los procedimientos ETL (1.4) se pueden subdividir en tres grandes etapas [2]:

1. Extraer: una de los desafíos principales lo encontramoes en esta primera fase, donde se trata de estraer los datos desde las fuentes de datos origen para dejarlos accesibles para las siguientes fases.
2. Transformar: se trata de las reglas de negocio a aplicar para obtener unos datos de alta calidad para resolver las cuestiones que se plantean. En esta fase son posibles diferentes operaciones sobre los datos entre las que encontramos: Integrar el dato con otras fuentes, modificar el contenido o la estructura del dato, calcular datos derivados o agregados,... etcétera.
3. Cargar: la tercera y última fase de esta etapa consiste en cargar los datos en el dispositivo objetivo y gestionar las dimensiones y jerarquías, con los desafíos que conlleva.

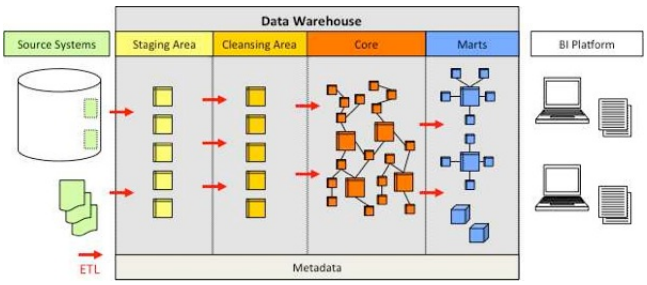


Figura 4.6: Data Warehousing

5

[5]Imagen extraida de http://www.dataprix.com/data-warehousing-y-metodologia-hefesto/arquitectura-del-data-warehouse/34-datawarehouse-manager

4.3.1. Diseño

Con la herramienta de Pentaho, *Pentaho Data Integration*, más conocido en la comunidad de desarrolladores de BI como *kettle*, se ofrece la posibilidad de realizar los diseños paralelamente a la implementación gracias a su potente capacidad *Drag Drop*. La figura B.14 representa un ejemplo de lo explicado: como es posible utilizar la misma herramienta gráfica tanto para el diseño de la *transformación* como para su implementación.

Por lo tanto el diseño de los procedimientos *ETL* podrían quedar inmersos en el apartado de implementación. No obstante y aunque no existen patrones de diseño estandarizados para éstos procesos, se ha pretendido mantender un sistema de buenas prácticas que permita al menos diseñar el mapeo desde los campos más importantes de la fuente de origen al sistema objetivo, permitiendo registrar una trazabilidad inicial como se puede observar en el anexo B.1, donde queda resgistrado dicho diseño.

4.3.2. Aspectos relevantes: Diseño de las estrategias de actualización

Las dimensión dim_date es estática, se carga una vez y se espera que no sea necesario actualizarla, al menos periodicamente. Sin embargo, no actualizar una dimensión no es lo habitual, ya que estas pueden sufrir modificaciones en sus datos orígenes por diferentes motivos. Por ejemplo, puede corregirse la fecha de nacimiento de un cliente, o éste puede cambiar de ciudad, o a una delegación se le puede ser asignado un delegado diferente... etcétera. En estos casos, se debe modelar la información de acuerdo a estos cambios.

En primer lugar, hay que considerar que la estrategia a utilizar dependerá de cada dimensión y de las necesidades de cada negocio. Por ejemplo si se actualiza el correo electrónico de un cliente, puede asumirse que el cambio debe realizarse en todo el histórico. No obstante, existen casos donde la información dimensional histórica es importante.Consideremos la siguiente situación:
Para realizar una previsión de asistencias por departamento, se debe considerar las asistencias históricas de los de los responsables de cada departamento, teniendo en cuenta que el responsable del departamento puede cambiar a lo largo del tiempo.

Por lo tanto en el ejemplo anterior se debe modelar la información para que tengamos la capacidad de conocer el responsable actual y el anterior. Con este fin existen diferentes técnicas de actualización que permiten tracear los cambios.

Dicho ésto, debemos saber que nos encontramos ante uno de los desafíos de mayor complejidad en el desarrollo e implementación de los procesos ETL, las denominadas **Slow Changing Dimensions** (SCD -"lentas", del inglés "slow")[4]. En general hay tres maneras de resolver este problema, aunque en este proyecto nos centramos en dos de ellas, ya que sabemos que son las que es posible implementar de una manera eficaz con las herramientas escogidas:

- Type1: El nuevo registro sustituye al viejo. No es necesario guardar diferentes versiones. Esto significa perder toda la historia del dato, y cuandohagamos un análisis veremos los datos desde un punto de vista actual.
- Type2: El nuevo registro es añadido a la dimensión. Es necesario tracearlo, guardar los cambios. El nuevo registro tiene una nueva clave subrogada, de forma que una entidad de sistema operacional (por ejemplo, un cliente), puede tener varios registros en la tabla de la dimensión según se van produciendo los cambios. Estamos gestionando un versionado, que ademas puede incluir unas fechas para indicar los periodos de validez, numerador de registros o un indicador de registro activo o no.

Inicialmente se diseña el sistema siguiendo una estrategia de **SCD Type2**, pero como se explica en apartados posteriores no es posible la implementación siguiendo este marco de actualización.

4.4. OLAP: Mondrian

Siguiendo con las herramientas de la suite que ofrece Pentaho, para el diseño e implementación de la base de datos OLAP se ha utilizado *Mondrian*, el cual implementa su propia sintaxis basada en XML para crear estas estructuras multidimensionales o como más correctamente debería denominarse, interfaz de acceso a la estructura multidimensional creada sobre un modelo en estreña en una base de datos relacional.

4.4.1. Diseño

El diseño trata de reunir cada una de las dimensiones que describirán los hechos representados como un cubo independiente y que utilizan la tabla de hechos correpondiente como eje principal.

Es decir, determinamos la capacidad de análisis a los ejes o dimensiones que participan en él y que pueden estar determinados por todos los que se relacionan con la tabla de hechos a traves de *foreign keys* o podemos prescindir de algunos de ellos en función del *datamart* a implementar.

Las medidas ha utilizar también se definen en el presente diseño, representan los KPI a analizar o las variables necesarias para su cálculo en tiempo de ejecución.

Una dificultad más, que se ha convertido en algo habitual a lo largo del proyecto, es la ausencia de estandares conocidos para el diseño de cubos OLAP. Siguiendo en la línea de todo el proyecto se ha optado por un diseño tabular, como puede apreciarse en las tablas 4.7 y 4.8 en el que se registran la tabla de hechos que representa el cubo, las dimensiones y las medidas a utilizar en los datamarts *Avisos* y *Ventas* respectivamente.

Cubo:	*Avisos*	
Tabla de hechos	**Dimensiones**	**Medidas**
fact_notice	Artículo Cliente Empleado cabecera Empleado línea Linea de servicio Servicio Fecha del servicio Fecha de la línea de servicio	Número de servicios Duración del servicio

Figura 4.7: Diseño datamart avisos

Cubo:	*Ventas*	
Tabla de hechos	**Dimensiones**	**Medidas**
fact_sales_receipt	Factura Cliente Fecha de factura Fecha de línea de factura	Importe Duración del servicio

Figura 4.8: Diseño datamart ventas

4.4.2. Aspectos relevantes

Para poder acceder al DWH haciendo uso de esta interfaz, es necesario conocer el lenguaje de consulta *MDX* que nos ofrece un poder analítico mucho mayor que *SQL*. Aunque cabe destacar que para lograr este objetivo es necesario definir muy bien el diseño, los agregados y medidas que las consultas deban utilizar en su ejecución, y que por nuestra experiencia la curva de aprendizaje es mucho mayor que la que proporciona *SQL*.

4.5. Cuadro de mando integral

La finalidad de todo el procesado anterior se alcanza en este apartado, en el que se define como presentar todos los datos obtenidos a lo largo de todo el *cocinado* mostrando la máxima información posible con un formato amigable y que de un sólo vistazo tengamos la oportunidad de conocer la *fotografía* actual de nuestro negocio.

4.5.1. Diseño de dashboards

En este apartado se pasa a describir cada uno de los prototipos iniciales diseñados con la finalidad de explotar la información. Para ello se pretende además de cumplir con los requisitos del cliente, seguir las directrices de usabilidad dentro de un marco de diseño actual y fácilmente reconocible con otras grandes aplicaciones, lo que convierte al sistema en un entorno más amigable y usable, teniendo en cuenta la complejidad que supone un sistema de BI.

4.5.1.1. Dashboard 1: Visión global de avisos

- 1: Parámetros de fechas: selección de medida (año/trimestre) y periodo correspondiente a la misma.
- 2: Tabla resumen de las OTs por trabajador, en el intervalo temporal seleccionado.
- 3: Análisis gráfico de los datos de la tabla.
- 4: Tabla de OTs clasificadas por estado.
- 5: Gráfico de OTs clasificadas por estado.
- 6: Tabla de OTs clasificadas por producción.
- 7: Gráfico de OTs clasificadas por producción.

Ver figura C.1

4.5.1.2. Dashboard 2: Avisos por empleado

- 1: Parámetros de fechas: selector de intervalo personalizado. Parámetro de selección del empleado.
- 2: Tabla con las OTs pendientes del trabajador seleccionado.
- 3: Gráfico de horas imputadas en cada día del intervalo seleccionado.
- 4: Gráfico de división por horas invertidas en OTs facturables/no facturables.

Ver figura C.2

4.5.1.3. Dashboard 3: Ventas globales

- 1: Parámetros de fechas: selector de intervalo personalizado.
- 2: Tabla resumen de las ventas anuales, desglosada por clientes. Pulsar sobre un cliente abrirá un gráfico que desglosará cada año por meses.
- 3: Gráfico de 5 mejores clientes.
- 4: Gráfico de 5 peores clientes.
- 5: Gráfico indicativo de facturación total anual.

Ver figura C.3

4.5.1.4. Dashboard 4: Ventas vs. avisos (localización)

- 1: Parámetros de fechas: selector de intervalo personalizado. Parámetro de selección de provincia.
- 2: Desglose de ingresos por localidades en la provincia seleccionada.
- 3: Desglose de horas imputadas por localidades en la provincia seleccionada.

Ver figura C.4

4.5.1.5. Dashboard 5: Análisis de artículos

- 1: Parámetros de fechas: selector de intervalo personalizado. Selección de categoría de artículos. Selección de valor de categoría de artículos.
- 2: Desglose de artículos bajo filtros seleccionados. Indicará varias medidas como la media de duración de servicio bajo cada artículo.
- 3: Tabla resumen de los miembros de la categoría seleccionada. Mostrará medidas estadísticas sobre la duración de los servicios de las mismas.
- 4: Gráfico estadístico de las categorías.
- 5: Cómputo por día en el intervalo seleccionado, indicando horas invertidas en cada categoría.

Ver figura C.5

4.5.2. Aspectos relevantes

En este sistema se pretendía, además de ofrecer unos Cuadros de Mando informativos y exhaustivos, aportar un remarcable componente de interactividad de cara al usuario final de los mismos. Esta visión conlleva que en la fase de diseño se tenga en cuenta la organización de los elementos en pantalla, de forma que a la hora de ser implementados, la funcionalidad esté acorde con la distribución de los diferentes objetos. Este requisito es importante para que no resulte confuso qué elemento interactúa con qué otro.

Capítulo 5

Selección de la herramienta de BI

En este capítulo se recoge el estudio realizado sobre las diferentes herramientas de Business Intelligence que han sido escogidas como candidatas. Para ello se ha recurrido a la documentación disponible de cada y herramienta y al uso personal en los casos en los que ha sido posible.

5.1. Evaluación de Software

Para el logro de los objetivos estratégicos recogidos en la etapa de análisis surge la necesidad de contar con un ambiente donde se tenga acceso a la información de manera oportuna, de una forma amigable, sin que ello signifique hacer requerimientos a sistemas o usuarios expertos o esperar tiempos de respuesta largos o escribir consultas complejas SQL. En este ambiente se debe poder explorar, visualizar, analizar, combinar con datos propios y generar reportes en tiempos razonables y con esquemas de seguridad.

En este contexto se plantea evaluar las herramientas de BI a considerar, que proporcione acceso controlado a la información, que cuente con funciones analíticas, de visualización, de exploración, de acceso, que permita al usuario final elaborar reportes, análisis, cuadros de mando, y compartir con otros usuarios y realizar sus propios análisis.

5.1.1. Factores a considerar

En la tabla 5.1 se exponen las principales características que serán tenidas en cuenta a la hora de realizar la evaluación del software de mercado que la empresa podría adoptar.

Id	Característica	Ranking	Descripción
		(1, 2, 3)	1 - Poca importancia, 2 - Normal, 3 - Mucha
1	**Requisitos básicos**		
1.1	Facilidad de uso	3	
1.2	Intuitividad	3	Capacidad de descubrir las posibilidades de la aplicación sin necesidad de ayuda
1.3	Tiempo de despliegue	2	El tiempo necesario para la implantación de la herramienta el la empresa
1.4	Soporte técnico continuo	3	Apoyo continuo de la empresa que suministra el software ante la aparición de problemas
1.5	Soporte de políticas de acceso a información	2	Capacidad de gestionar la información disponible a determinados usuarios y determinados grupos de usuarios
2	**Requisitos de organización**		
2.1	Formación de empleados	3	Formación de los empleados en el uso de la herramienta
2.2	Técnicos formados	3	Formación de los empleados de gestión de nuestra empresa en la administración de la herramienta
2.3	Soporte de mantenimiento	3	Disponibilidad de la empresa ante la sucesión de incidencias con la herramienta
3	**Requisitos del producto**		
3.1	Implantación en el mercado	1	Uso de la herramienta en el ámbito profesional
3.2	Opiniones de los clientes	1	Satisfacción de las empresas tras la implantación de la herramienta. Casos de implantación y resultados
4	**Requerimientos técnicos**		
4.1	**Orígenes de datos**		
4.1.1	Soporte de bases de datos	3	
4.1.2	Soporte para ficheros de texto	1	
4.1.3	Soporte de puertas de enlace	1	
4.1.4	Soporte para sistemas online	3	
4.2	**Destinos de datos**		
4.2.1	Ficheros de texto	2	
4.2.2	Hojas de cálculo	3	Exportación de datos a hojas de cálculo
4.3	Arquitectura cliente servidor	2	
4.4	**Servicios de rutinas**		
4.4.1	Soporte para cálculos del usuario	3	Posibilidad de que el usuario defina consultas personalizadas
4.4.2	Soporte para gráficos definidos por el usuario	3	Posibilidad de que el usuario defina gráficos personalizados. Posibilidad de modificación de los existentes
4.4.3	Envíos de correo a una lista de usuarios	2	
4.5	**Publicación de datos en web**		
4.5.1	HTML Estático	2	Generación de informes en HTML
4.5.2	Funcionalidad dinámica	3	Generación de informes dinámicos en base a los datos de cada momento
4.5.3	Gestión de aplicación web	3	Facilidad de la gestión del contenido mostrado en la web
4.5.4	Complejidad en el desarrollo	3	
4.8	**Integración con sistemas móviles**		
4.8.1	Interfaz adaptada	3	Visionado acorde con el tamaño de la pantalla del dispositivo (móvil, tablet) y las capacidades de uso del mismo (interfaz táctil, deslizamiento de páginas)
4.8.2	Integración completa	2	Disponibilidad de todas las capacidades de la herramienta estándar
4.8.3	Disponibilidad	2	Exclusivamente en línea, almacenamiento de datos en el dispositivo…
4.8.4	Seguridad en las conexiones	3	Cifrado de la información enviada, autenticación fiable
4.9	**Capacidad de realización de informes**		
4.9.1	Disponibilidad de plantillas	2	
4.9.2	Automatización de informes	3	Posibilidad de lanzar automáticamente informes predefinidos según un periodo de tiempo
4.9.3	Ajuste de tamaño del informe	1	
4.10	**Integración con softwares de terceros**		
4.10.1	Agendas y calendarios	2	
4.10.2	Aplicaciones de correo	2	
4.10.3	Hojas de cálculo	3	

Figura 5.1: Factores a evaluar

5.1.1.1. Alternativas

Se tiene en cuenta que la empresa posee una base de datos ya implantada y el software que gestiona el almacenamiento de datos en la misma. Es decir, se necesita implantar una solución que permita tanto **monitorizar el estado** de la empresa como el de sus **procesos**. La solución de software a implantar debe ser una herramienta de **análisis de datos** que permita su **visualización** de forma rápida y precisa, contando para ello con ayudas visuales.
Las soluciones propuestas se centran en el apartado de *Business Intelligence*. Este tipo de soluciones resultan idóneas para el trabajo que se debe llevar a cabo. Las opciones a evaluar son las siguientes:

- *Excel*.
- *Qlikview*.
- *SAP*
- *Pentaho BI*

5.1.1.2. Excel

Herramienta de hojas de cálculo de *Microsoft*. No es una herramienta de *Business Intelligence* propiamente dicha, pero puede ser adaptada para la tarea.

Es capaz de trabajar con varias fuentes de datos. Permite el trabajo con datos guardados en otras hojas de cálculo. Permite imprimir la propia hoja en formato *pdf* a modo de informe, seleccionando cabeceras, pies de página, área del documento a incluir en la impresión... Es una herramienta bastante versátil.

5.1.1.3. Qlickview

Herramienta de *Business Intelligence* [16], [5] de gran flexibilidad, con posibilidad de obtener soluciones adaptadas a áreas determinadas. A destacar su compatibilidad con gran variedad de fuentes de información con las que puede trabajar, incluyendo hojas de cálculo de *Excel*. Permite la creación de interfaces intuitivas con gráficos a través de los que se puede navegar y acceder a diferente información mediante filtros automáticos.

Presenta también propuestas para la integración del entorno móvil y la creación de *apps* para el mismo.

La evaluación se basa en: [1], [5], [22], [9], [17].

5.1.1.4. SAP Business Intelligence

En este caso no se trata de una herramienta, si no de varias. SAP ofrece múltiples soluciones [21], cada una adaptada a unas características determinadas, abarcando una gran variedad de fuentes de datos e integración con software de terceros.

Cabe destacar que las múltiples necesidades de software de una empresa pueden ser cubiertas mediante el software SAP, que ofrece soluciones para múltiples sistemas de información.

La evaluación se basa en: [21], [10], [18], [19], [20].

5.1.1.5. Pentaho Business Intelligence

La suite de Pentaho nos brinda un conjunto de potentes herramientas *Open source* capaces de cubrir todo el desarrollo de un proyecto de BI. Desde la integración de datos, haciendo uso de la herramienta *Pentaho Data Integration* pasando por la mienría de datos con el software *Weka* incluido en el proyecto *Pentaho*, hasta la implementación y desarrollo con la herramienta *Bi Server* y los múltiples componentes de los que dispone.

Cabe mencionar la potente solución de Integración de datos como una potente herramienta, no sólo en el ámbito de BI, si no también en otros como la migración de datos entre versiones y tecnologías. Todo ello libre de licencias y con una comunidad de usuarios detrás, que asegura el soporte y la actualización a corto-medio plazo.

La evaluación se basa en: [15], [1]

5.2. Resultados por aplicación

Para la evaluación se usan notas entre 0 y 10 y se ponderan de acuerdo a la importancia de la característica dada por el apartado “ranking”. Además se tendrán en cuenta otros aspectos relevantes como el cobro de licencias y la escalabilidad y el aumento potencial del número de usuarios.

Excel

Id	Característica	Ranking	Nota	Ponderada	Justificación
4.9.2	Automatización de informes	3	8	24	Posibilidad de automatizar el procesado y su distribución mediante soluciones y macros de terceros
4.9.3	Ajuste de tamaño del informe	1	8	8	Diferentes tamaños de hoja,
	Total			46	
4.10	**Integración con softwares de terceros**				
4.10.1	Agendas y calendarios	2	4	8	Pobre integración. Basada en que la misma hoja es el calendario
4.10.2	Aplicaciones de correo	2	6	12	Posible mediante el uso de macros

Id	Característica	Ranking	Nota	Ponderada	Justificación
1	**Requisitos básicos**				
1.1	Facilidad de uso	3	6	18	El uso de elementos y técnicas de cierta complejidad requiere conocimientos avanzados y uso de varios pasos
1.2	Intuitividad	3	7	21	Hace falta explorar mucho la interfaz para encontrar determinadas características
1.3	Tiempo de despliegue	2	8	16	Al ser una solución individual sin servidores, el tiempo de despliegue resulta muy bajo. Una vez instalada se puede empezar a editar archivos
1.4	Soporte técnico continuo	3	7	21	Basado principalmente en soporte online
1.5	Soporte de políticas de acceso a información	2	2	4	El acceso a los archivos queda en manos del Sistema Operativo. De forma complementaria se pueden usar diferentes servidores para su administración y almacenamiento
	Total			80	
2	**Requisitos de organización**				
2.1	Formación de empleados	3	8	24	Gran disponibilidad de cursos tanto online como presenciales
2.2	Técnicos formados	3	8	24	Gran disponibilidad de cursos tanto online como presenciales
2.3	Soporte de mantenimiento	3	7	21	Servicio online con recursos y comunidad de usuarios
	Total			69	
3	**Requisitos del producto**				
3.1	Implantación en el mercado	1	8	8	Herramienta muy extendida en el mercado. Principalmente como solución de análisis de datos a pequeña escala.
3.2	Opiniones de los clientes	1	8	8	Posee buena reputación como herramienta de uso en el entorno empresarial
	Total			16	
4	**Requerimientos técnicos**				
4.1	**Orígenes de datos**				
4.1.1	Soporte de bases de datos	3	8	24	Bases de datos Access, SQL Server y orígenes ODBC
4.1.2	Soporte para ficheros de texto	1	8	8	Soporte y posibilidad de definir la importación de datos a cada celda (formatos de separación de datos)
4.1.3	Soporte de puertas de enlace	1		0	
4.1.4	Soporte para sistemas online	3	8	24	Soporte de extracción de datos de páginas WEB
	Total			56	
4.2	**Destinos de datos**				
4.2.1	Ficheros de texto	2	10	20	Posibilidad de elegir diversos limitadores de datos
4.2.2	Hojas de cálculo	3	10	30	Archivo por defecto
4.3	Arquitectura cliente servidor	2	2	4	La aplicación funciona en local, aunque contempla la integración con servidores de datos
	Total			54	
4.4	**Servicios de rutinas**				
4.4.1	Soporte para cálculos del usuario	3	10	30	El usuario es el que tiene que definir todos los cálculos
4.4.2	Soporte para gráficos definidos por el usuario	3	10	30	La creación de gráficos la decide el usuario
4.4.3	Envíos de correo a una lista de usuarios	2	0	0	Este proceso se realizaría mediante software de terceros
	Total			60	
4.5	**Publicación de datos en web**				
4.5.1	HTML Estático	2	10	20	Posibilidad de guardar la hoja de cálculo como página web
4.5.2	Funcionalidad dinámica	3	0	0	Necesidad de actualizar manualmente la página
4.5.3	Gestión de aplicación web	3	0	0	No posee aplicación de gestión. Soluciones de servidor aparte
4.5.4	Complejidad en el desarrollo	3	7	21	En el caso de obtener informes en HTML estático, se consigue de forma simple
	Total			41	
4.8	**Integración con sistemas móviles**				
4.8.1	Interfaz adaptada	3	8	24	Disponibilidad de la suite para teléfonos y tablets
4.8.2	Integración completa	2	7	14	Versión más reducida que la de escritorio
4.8.3	Disponibilidad	2	7	14	Necesidad de importar los archivos al dispositivo. Posibilidad de uso de servidores con software adicional
4.8.4	Seguridad en las conexiones	3	0	0	No hay conexión, salvo que se use software adicional
	Total			52	
4.9	**Capacidad de realización de informes**				
4.9.1	Disponibilidad de plantillas	2	7	14	Disponibilidad de plantillas en línea, posibilidad de definirlas

Qlikview

Id	Característica	Ranking	Nota	Ponderada	Justificación
1	**Requisitos básicos**				
1.1	Facilidad de uso	3	8	24	Funciones avanzadas requieren conocimientos fuera del alcance para ciertos usuarios
1.2	Intuitividad	3	8	24	Navegación sencilla a través de los paneles de la herramienta
1.3	Tiempo de despliegue	2	7	14	Tiempo necesario de configuración de la herramienta en la empresa mayor a la semana
1.4	Soporte técnico continuo	3	9	27	Soporte online y portal de clientes para tramitación de solicitudes
1.5	Soporte de políticas de acceso a información	2	8	16	Posibilidad de establecer el nivel de acceso según tipos de usuario
	Total			105	
2	**Requisitos de organización**				
2.1	Formación de empleados	3	9	27	Cursos adaptados según perfil del empleado (diseño gráfico, desarrollador de aplicaciones, administración, usuario final)
2.2	Técnicos formados	3	9	27	Formación específica para técnicos de administración
2.3	Soporte de mantenimiento	3	9	27	Soporte y tramitación online. Equipos especializados de ayuda
	Total			81	
3	**Requisitos del producto**				
3.1	Implantación en el mercado	1	8	8	Cuota en alza. Herramienta innovadora
3.2	Opiniones de los clientes	1	9	9	Facilidad de uso, descubrimiento de relaciones entre los datos sencilla.
	Total			17	
4	**Requerimientos técnicos**				
4.1	**Orígenes de datos**				
4.1.1	Soporte de bases de datos	3	9	27	Comunicación con diversos tipos de bases de datos
4.1.2	Soporte para ficheros de texto	1	10	10	Importación de diversos orígenes de datos utilizando la información no estructurada de ficheros
4.1.3	Soporte de puertas de enlace	1	9	9	
4.1.4	Soporte para sistemas online	3	9	27	
	Total			73	
4.2	**Destinos de datos**				
4.2.1	Ficheros de texto	2	9	18	
4.2.2	Hojas de cálculo	3	9	27	Posibilidad de exportar objetos a hojas de cálculo
4.3	Arquitectura cliente servidor	2	9	18	Posibilidad de implementar qlikview en un servidor al que realizar peticiones
	Total			63	
4.4	**Servicios de rutinas**				
4.4.1	Soporte para cálculos del usuario	3	9	27	El usuario es capaz de definir que tipo de análisis se llevan a cabo con los datos una vez cargados
4.4.2	Soporte para gráficos definidos por el usuario	3	9	27	Gran variedad de gráficos en los que elegir
4.4.3	Envíos de correo a una lista de usuarios	2	8	16	Posibilidad contemplada
	Total			70	
4.5	**Publicación de datos en web**				
4.5.1	HTML Estático	2	9	18	
4.5.2	Funcionalidad dinámica	3	7	21	Posibilidad de automatizar la creación de informes HTML mediante software de terceros
4.5.3	Gestión de aplicación web	3	8	24	Posibilidad de implementar Qlikview Web Server
4.5.4	Complejidad en el desarrollo	3	7	21	
	Total			84	
4.8	**Integración con sistemas móviles**				
4.8.1	Interfaz adaptada	3	9	27	App de acceso disponible
4.8.2	Integración completa	2	9	18	Posibilidad de interacción con los datos, mayor sencillez de uso que la herramienta de escritorio
4.8.3	Disponibilidad	2	9	18	Acceso total a fuentes de datos
4.8.4	Seguridad en las conexiones	3	9	27	Conexiones móviles a través de Qlikview Server
	Total			90	
4.9	**Capacidad de realización de informes**				
4.9.1	Disponibilidad de plantillas	2	9	18	Capacidad de definir las plantillas
4.9.2	Automatización de informes	3	9	27	Capacidad de lanzar la creación de informes con plantillas predefinidas
4.9.3	Ajuste de tamaño del informe	1	8	8	Adaptación a diferentes tamaños de informe
	Total			53	
4.10	**Integración con softwares de terceros**				

Id	Característica	Ranking	Nota	Ponderada	Justificación
4.10.1	Agendas y calendarios	2		0	No evaluado
4.10.2	Aplicaciones de correo	2	7	14	Posible mediante servicios de distribución de terceros
4.10.3	Hojas de cálculo	3	10	30	Posibilidad de importar y exportar datos. Exportación de objetos
	Total			44	

Figura 5.2: Evaluación Qlikview

SAP

Id	Característica	Ranking	Nota	Ponderada	Justificación
1	**Requisitos básicos**				
1.1	Facilidad de uso	3	7	21	Herramienta compleja
1.2	Intuitividad	3	7	21	Menor intuitividad que las soluciones anteriores. Los usuarios finales sí que poseen dashboards preparados
1.3	Tiempo de despliegue	2	7	14	Tiempo necesario de configuración de la herramienta en la empresa mayor a la semana
1.4	Soporte técnico continuo	3	9	27	Planes de soporte independientes de la plataforma adoptada (Cloud, en servidores propios…). Diferentes tipos de plantes adaptados a las necesidades de la empresa
1.5	Soporte de políticas de acceso a información	2	8	16	Posibilidad de administración de usuarios y roles
	Total			99	
2	**Requisitos de organización**				
2.1	Formación de empleados	3	9	27	Formación centrada en la solución SAP adoptada
2.2	Técnicos formados	3	9	27	Cursos de mantenimiento centrados en la solución adoptada
2.3	Soporte de mantenimiento	3	9	27	Planes de soporte independientes de la plataforma adoptada (Cloud, en servidores propios…). Diferentes tipos de plantes adaptados a las necesidades de la empresa
	Total			81	
3	**Requisitos del producto**				
3.1	Implantación en el mercado	1	9	9	Suite ampliamente implantada en el mercado
3.2	Opiniones de los clientes	1	8	8	Clientes satisfechos. Gran versatibilidad de la herramienta
	Total			17	
4	**Requerimientos técnicos**				
4.1	**Orígenes de datos**				
4.1.1	Soporte de bases de datos	3	10	30	Integración de Bases de Datos internas (las propias utilizadas por las soluciones SAP) como externas (de otros proveedores, conectores ODBC y JDBC entre otros)
4.1.2	Soporte para ficheros de texto	1	8	8	Soporte en varios formatos de presentación de datos
4.1.3	Soporte de puertas de enlace	1		0	No evaluado
4.1.4	Soporte para sistemas online	3		0	No evaluado
	Total			38	
4.2	**Destinos de datos**				
4.2.1	Ficheros de texto	2	8	16	Soporte en varios formatos de presentación de datos
4.2.2	Hojas de cálculo	3	9	27	Compatibilidad tanto para exportación como importación
4.3	Arquitectura cliente servidor	2	9	18	Disponibilidad de servidores de recursos a los que se conectan diversos clientes de escritorio
	Total			61	
4.4	**Servicios de rutinas**				
4.4.1	Soporte para cálculos del usuario	3	8	24	Soporte de generación de consultas
4.4.2	Soporte para gráficos definidos por el usuario	3	7	21	Soportado
4.4.3	Envíos de correo a una lista de usuarios	2	8	16	Posible mediante soluciones de la misma suite. Distribución de informes
	Total			61	
4.5	**Publicación de datos en web**				
4.5.1	HTML Estático	2	9	18	Posibilidad contemplada
4.5.2	Funcionalidad dinámica	3	9	27	Acceso a paneles de información con información actualizada en todo momento
4.5.3	Gestión de aplicación web	3	8	24	Solución de servidor propia
4.5.4	Complejidad en el desarrollo	3		0	No evaluado
	Total			69	
4.8	**Integración con sistemas móviles**				
4.8.1	Interfaz adaptada	3	10	30	Visualización de datos
4.8.2	Integración completa	2	9	18	App específica para la conexión a servidores de datos. Posibilidad de navegar a través de datos. Plantillas previamente preparadas
4.8.3	Disponibilidad	2	8	16	Requiere conexión a un servidor SAP para realizar análisis de datos. Necesidad de conexión permanente a Internet
4.8.4	Seguridad en las conexiones	3	9	27	Transporte de datos por canales seguros mediante cifrado de la información
	Total			91	
4.9	**Capacidad de realización de informes**				
4.9.1	Disponibilidad de plantillas	2	9	18	Posibilidad de basar los informes en plantillas previas
4.9.2	Automatización de informes	3	8	24	Posibilidad de planificar el lanzamiento periódico de informes
4.9.3	Ajuste de tamaño del informe	1	8	8	Diferentes tamaños y tipos de documento
	Total			50	
4.10	**Integración con softwares de terceros**				
4.10.1	Agendas y calendarios	2		0	No evaluado
4.10.2	Aplicaciones de correo	2	8	16	Solución para integración de correo contemplada en la misma suite

Figura 5.3: Evaluación SAP

Pentaho

Id	Característica	Ranking	Nota	Ponderada	Justificación
1	**Requisitos básicos**				
1.1	Facilidad de uso	3	7	21	Si se desea un sistema interactivo y bien diseñado son necesarias capacidades de desarrollo
1.2	Intuitividad	3	8	24	Navegación sencilla, menús claros y explicativos
1.3	Tiempo de despliegue	2	6	12	El sistema tiene bastantes requisitos si se quiere tener un entorno adecuado para producción
1.4	Soporte técnico continuo	3	7	21	Comunidad on-line con actividad media
1.5	Soporte de políticas de acceso a información	2	9	18	Posibilidad de crear roles y usuarios, modificar accesos a nivel de archivo o carpeta. Se puede restringir mediante código extra (JavaScript)
		Total		96	
2	**Requisitos de organización**				
2.1	Formación de empleados	3	9	27	Cursos adaptados según perfil del empleado (diseño gráfico, desarrollador de aplicaciones, administración, usuario final)
2.2	Técnicos formados	3	9	27	Formación específica para técnicos de administración
2.3	Soporte de mantenimiento	3	9	27	Soporte y tramitación online. Equipos especializados de ayuda
		Total		81	
3	**Requisitos del producto**				
3.1	Implantación en el mercado	1	8	8	Suite de referencia en cuanto a open-source
3.2	Opiniones de los clientes	1	9	9	Flexibilidad y libertad de personalización, frecuencia de actualizaciones
		Total		17	
4	**Requerimientos técnicos**				
4.1	**Orígenes de datos**				
4.1.1	Soporte de bases de datos	3	9	27	Todo tipo de base de datos
4.1.2	Soporte para ficheros de texto	1	6	6	Es necesario utilizar la herramienta PDI
4.1.3	Soporte de puertas de enlace	1	9	9	
4.1.4	Soporte para sistemas online	3	9	27	
		Total		69	
4.2	**Destinos de datos**				
4.2.1	Ficheros de texto	2	7	14	Posibilidad de exportar objetos a ficheros de texto (sin parametrización)
4.2.2	Hojas de cálculo	3	7	21	Posibilidad de exportar objetos a hojas de cálculo (sin parametrización)
4.3	Arquitectura cliente servidor	2	9	18	La herramienta corre enteramente como un servidor
		Total		53	
4.4	**Servicios de rutinas**				
4.4.1	Soporte para cálculos del usuario	3	9	27	Posibilidad de análisis OLAP mediante plug-ins: Saiku Analytics, Analytics (Pentaho)
4.4.2	Soporte para gráficos definidos por el usuario	3	9	27	Gran variedad de gráficos en los que elegir
4.4.3	Envíos de correo a una lista de usuarios	2	8	16	Posibilidad contemplada
		Total		70	
4.5	**Publicación de datos en web**				
4.5.1	HTML Estático	2	10	20	Soporte total, cuadros de mando implementados en HTML
4.5.2	Funcionalidad dinámica	3	10	30	Posibilidad de implementar funciones Javascript personalizadas para dinamizar los cuadros de mando
4.5.3	Gestión de aplicación web	3	5	15	Limitada, es necesario crear un proyecto con el código fuente
4.5.4	Complejidad en el desarrollo	3	6	18	Es flexible pero requiere conocimientos avanzados de desarrollo web
		Total		83	
4.8	**Integración con sistemas móviles**				
4.8.1	Interfaz adaptada	3	10	30	Interfaz adaptada a dispositivos móviles mediante Bootstrap/Jquery.
4.8.2	Integración completa	2	9	18	Mismas funcionalidades que en un equipo de escritorio
4.8.3	Disponibilidad	2	9	18	Mismo acceso que en dispositivos de escritorio
4.8.4	Seguridad en las conexiones	3	9	27	Inherente al servidor donde se despliega la conexión
		Total		93	
4.9	**Capacidad de realización de informes**				
4.9.1	Disponibilidad de plantillas	2	8	16	Existen plantillas, pueden ser creadas
4.9.2	Automatización de informes	3	9	27	Existen "wizards" para ayudar a aplicar plantillas
4.9.3	Ajuste de tamaño del informe	1	8	8	Adaptación a diferentes tamaños de informe
		Total		51	
4.10	**Integración con softwares de terceros**				
4.10.1	Agendas y calendarios	2		0	No evaluado
4.10.2	Aplicaciones de correo	2	8	16	Pueden ser enviados e-mails. La herramienta PDI tiene funcionalidades extra a este respecto
4.10.3	Hojas de cálculo	3	8	24	Posibilidad de importar y exportar datos con personalización limitada
		Total		40	

Figura 5.4: Evaluación Pentaho

5.3. Resumen y conclusiones

La conclusión clara es que Excel, pese a ser una herramienta de gran difusión, no es por sí misma una solución de Business Intelligence adecuada para la aplicación en la totalidad de la empresa y se requieren otras mejor integradas para desempeñar dicha función.

El resto de herramientas cumplen con los requisitos necesarios del proyecto, por lo que la elección de la herramienta se basará en otros aspectos que las diferencie.

La herramienta Qlikview se presenta como un potente candidato tanto por los resultados de la evaluación como por la curva de aprendizaje reducida, la facilidad en cuanto a la integración de datos. Aunque la propia herramienta es capaz de soportar parte del ciclo de vida de los procesos *ETL*, proporcionando una potente solución, no es capaz de implementar la solución que permita desarrollar y mantener un *Data warehouse* corporativo y aunque si es capaz de funcionar como un servidor de escritorio no implementa la interfaz web que nos permite desarrollar nuestra solución como un *SaaS*. Cabe mencionar que se trata de software privativo con un coste de licencia de alrededor de 12000 €.

Como segunda herramienienta mejor valorada en el ranking se encuentra la suite BI que nos ofrece el proyecto *Pentaho*, software *Open Source* BI de referencia, con una comunidad de más de 17000 usuarios y desarrolladores. Como desventaja nos encontramos con una curva de aprendizaje mayor que la herramienta anterior dado que es necesario tener conocimientos de programación Web y librerías Javascript si queremos exprimir al máximo las funcionalidades que es capaz de ofrecer.

Por contrapartida a esto tenemos múltiples ventajas, a destacar: debido a la naturaleza libre del software, no es necesario pagar ninguna licencia, por lo que esto nos permite disponer de muchas más horas para el desarrollo del proyecto; de esta forma, la curva de aprendizaje queda compensada. Por otra parte, nos encontramos ante un proyecto *Open source* que nos permite obtener el código fuente para ajustar a las necesidades del proyecto, incluido el nucleo del servidor.

Teniendo en cuenta las conclusiones y el resultado del ranking en la figura 5.5, la herramienta escogida es *Pentaho BI*.

Id	Característica	Excel	Qlikview	SAP	Pentaho
1	Requisitos básicos	80	105	99	96
2	Requisitos de organización	69	81	81	81
3	Requisitos del producto	16	17	17	17
4	Requerimientos técnicos				
4.1	Orígenes de datos	56	73	38	69
4.2	Destinos de datos	54	57	61	53
4.4	Servicios de rutinas	60	70	61	70
4.5	Publicación de datos en web	41	76	69	83
4.8	Integración con sistemas móviles	52	90	91	93
4.9	Capacidad de realización de informes	46	53	50	51
4.10	Integración con softwares de terceros	44	38	40	40
	Total	518	660	607	653

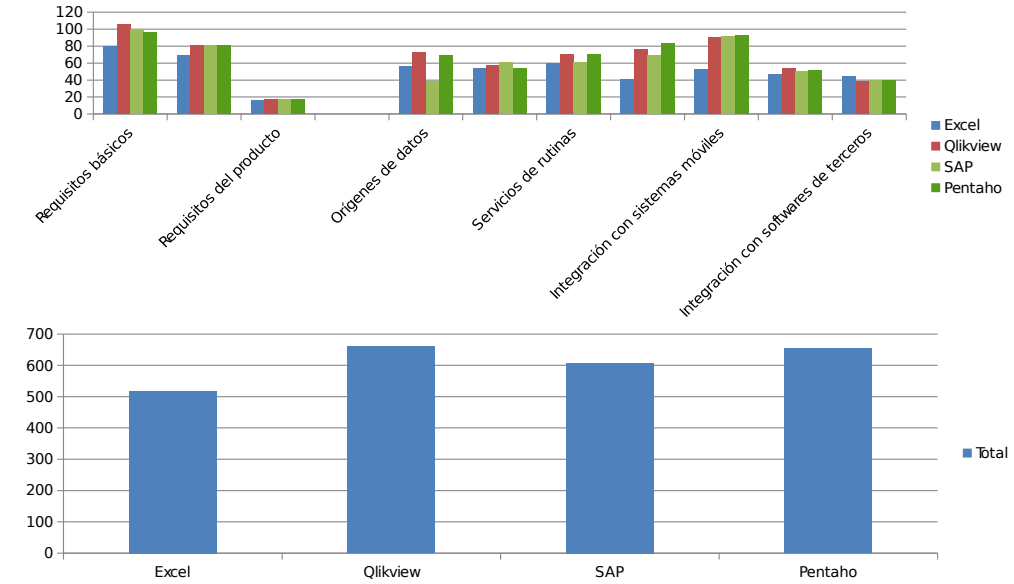


Figura 5.5: Resumen de resultados de la evaluación

Capítulo 6

Desarrollo

En este capítulo se cubrirán las diversas fases acometidas a lo largo del desarrollo del sistema, debidamente organizadas por partes de la arquitectura cliente/servidor y bloques funcionales.

6.1. Servidor

Esta sección trata las fases de implementación del sistema que hará de origen de datos y la configuración de la suite Pentaho para que esta se adecue a los requisitos del cliente.

6.1.1. Data Warehouse y procedimientos ETL

Con el objetivo de poder implementar el Data Warehouse, en primer lugar, ha sido necesario crear una base de datos en PostgreSQL. Para esto se ha utilizado el diseño de base de datos generado en la fase de análisis y diseño mediante la herramienta SQL Power Architect, ya que la misma nos da la opción de generar un script SQL que creará la estructura necesaria.

Una vez que la base de datos (de este punto en adelante, Data Warehouse o DWH) está operativa, se procede a crear los procesos ETL necesarios para obtener los datos de la fuente origen, tratarlos y cargarlos en el Data Warehouse, verificando siempre la consistencia de los mismos.

Con este fin se utiliza la herramienta Pentaho Data Integration (PDI) o Kettle. Este software dispone de una herramienta gráfica *Spoon* que permite la programación de una manera intuitiva y muy visual, Kettle clasifica los procesos ETL en dos diferentes: Las transformaciones serán los procesos de más bajo nivel, dedicados a realizar transacciones parciales para el Data Warehouse. Para desarrollar las transformaciones disponemos de tres elementos prncipales que se pueden ver en la figura 6.1.

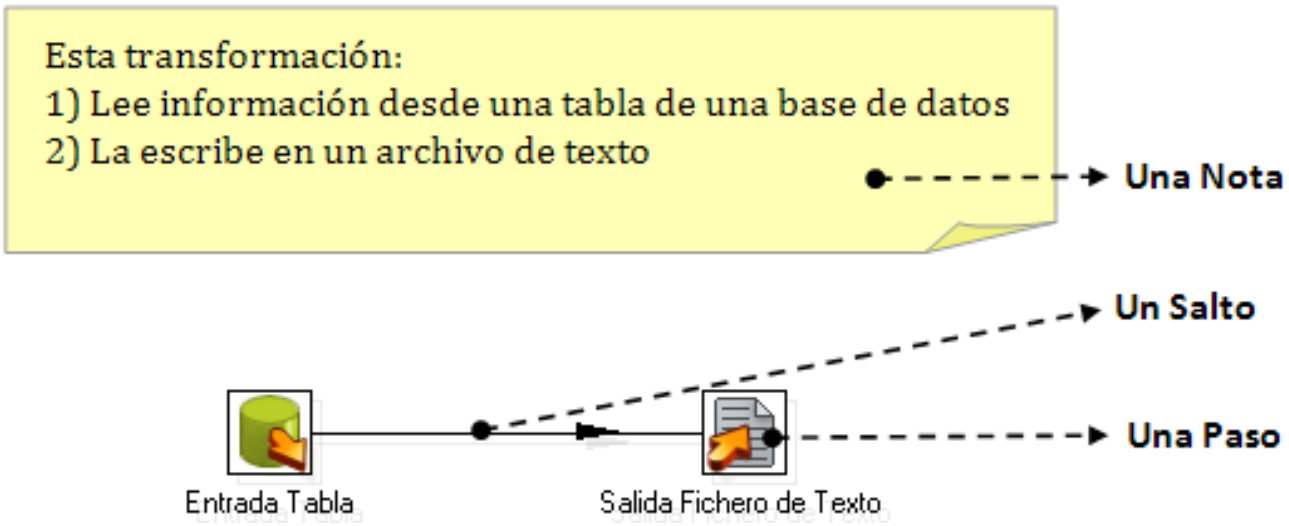


Figura 6.1: Transformación

[1]

Los pasos son los elementos con los que procedemos a programar la ejecución del proceso diseñado. Existen pasos predefinidos de entrada de datos, de transformación de cadenas de caracteres o string, estadísticos y muchos más. La potencia y escalabilidad es ampliada por medio de los pasos de transformación de *Scripting*. Conectando el flujo a estos pasos se puede tratar el flujo utilizando diferentes lenguajes de programación. Javascript ha sido el comunmente utilizado para este proyecto.

En la figura podemos ver el elemento gráfico que une los dos pasos que se encuentran en el lienzo. Este elemento se denomina "salto" y es el encargado de distribuir el flujo de datos a través de los diferentes pasos. Es significativo tener en cuenta el orden de la ejecución de la transformación, se trata de una ejecución paralela, algo a tener muy en cuenta en nuestra programación.

Los trabajos son utilizados para control de flujo a alto nivel, entre otras funcionalidades para ordenar transformaciones de forma que el Data Warehouse se poble de forma parcial, secuencial y consistente. Estos últimos también nos permiten programar ejecuciones en intervalos temporales o en momentos concretos. Esta última funcionalidad nos ha permitido realizar de una manera ágil y fiable la implementación de las estrategias temporales de actualización del DWH, limitadas por la actividad y el horario que la empresa desempeña. En las

[1] Imagen extraida de http://wiki.pentaho.com/pages

dos empresas que utilizan el sistema el volcado y actualización de datos es diarío y se realiza en horario nocturno.

Los elementos que nos encontramos en la programación de los trabajos es similar al de las transformaciones, salvo algunas particularidades inherentes a su función en un proceso ETL. Por ejemplo el orden de ejecución es secuencial y los saltos son de dos tipos: condicionados al éxito en la ejecución del paso o incondicionales. Además en un trabajo siempre debe existie el paso *start* y debe ser único como se puede apreciar en la figura 6.2.

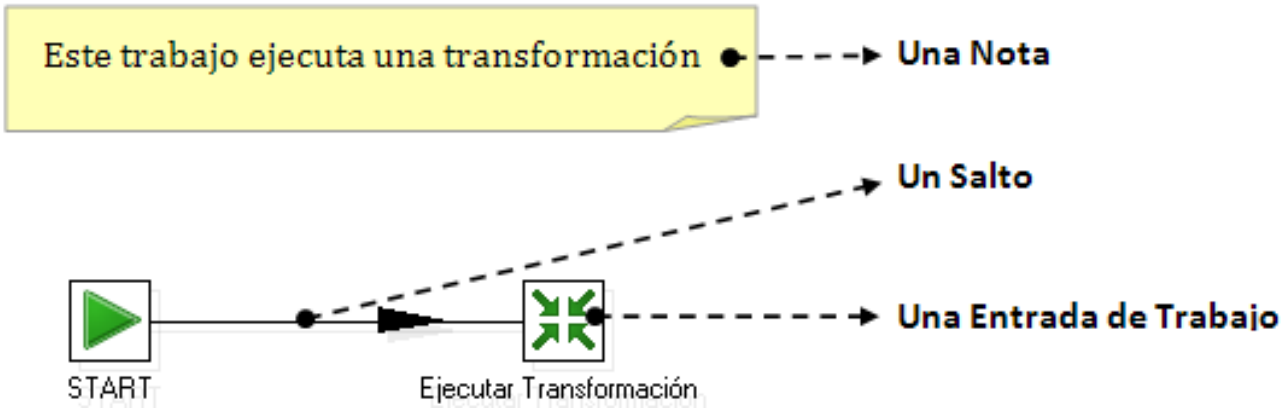


Figura 6.2: Trabajo

[2]

Para más información se puede consultar el manual de la herramienta en su versión online[7], o alguno de los libros de la bibliografía [2],[3].

En las siguientes sub-secciones se describirán las transformaciones más destacables del proceso, adjuntando instantáneas de su diseño y puntualizando aspectos destacables, así como diversas configuraciones o políticas llevadas a cabo durante la implementación:

6.1.1.1. Conexiones remotas

La extracción de los datos para el Data Warehouse ha implicado acceder a un servidor de bases de datos remoto, ajeno a la red local donde los procesos han sido lanzados. El acceso ha sido llevado a cabo mediante el driver JDBC para la base de datos Progress, al que se le deben indicar una dirección IP, un usuario y una contraseña. Una de las particularidades de este sistema es que no se nombran las bases de datos por un identificador de texto, sino que se debe utilizar un puerto de red con este motivo.

6.1.1.2. Transformaciones y trabajos

En este apartado se describirán las diferentes transformaciones que son necesarias para la correcta generación y mantenimiento de los datos del Data Warehouse. No se incluyen todas las implementadas debido a que parte de ellas siguen un mismo patrón, en cuyo caso será descrito y se citará un ejemplo.

Cabe destacar que un proceso ETL correctamente implementado debería tener en cuenta últimas fechas de actualización para determinar si los diversos registros ya presentes en el Data Warehouse deben ser actualizados. Sin embargo, ante la ausencia de dicho dato en la base de datos origen de la empresa cliente, se ha tenido que omitir su uso, actualizando todos los datos si ya están cargados, siendo la excepción algunas dimensiones que por reglas de negocio no variarán una vez introducidas en la base de datos.

6.1.1.2.1. Transformación estándar: Estas transformaciones siguen un esquema común, variando únicamente en los datos que manejan. Extraen los datos de la base de datos Progress, y comparan las claves obtenidas con aquellas almacenadas en la dimensión que se está tratando. Los registros que se hallen en el Data Warehouse serán actualizados, de no ser encontrados se introducirán como nuevos datos.

Estas transformaciones son aquellas que cargan las dimensiones de cliente, líneas de OT, líneas de factura, proveedores y artículos.

Ver figuras B.12, B.17, B.19, B.16, B.13

[2]Imagen extraida de http://wiki.pentaho.com/pages

6.1.1.2.2. Dimensión de fecha: Esta transformación solo será ejecutada en una ocasión, al inicializar el Data Warehouse. Subsecuentes ejecuciones solo serán necesarias en el caso de que se quiera modificar el formato de alguno de los campos de la dimensión, o se necesite añadir más fechas.

El proceso comienza creando tantas filas como días existen en el intervalo temporal deseado, y a cada una de estas filas se les añade un número en secuencia (1, 2, 3...). Tendiendo las filas procesadas se llega a un paso Javascript en el que mediante varias funciones, se completa la información para cada día generado. Estos datos engloban tanto diversos formatos de fecha como información temporal en diferentes formatos: trimestre, mes, día del año, año, etcétera, así como el campo que será la clave primaria de cada día. Con esta información creada se realiza la carga en la dimensión fecha del Data Warehouse.

Existe también la posibilidad de en lugar de habilitar la carga, habilitar la actualización de los datos ya presentes en el almacén de datos, para los casos anteriormente mencionados.

En este caso concreto, se generan 10 años de fechas, desde el 2006, año desde el que se tienen registros en la base de datos, hasta el 2016.

Ver figura B.14

6.1.1.2.3. Dimensión de factura: Esta dimensión es una de las que carecerá de proceso de actualización debido a las reglas de negocio: una vez una factura ha sido introducida en el sistema, no podrá ser modificada. Esta transformación tiene la peculiaridad de que se añade una constante que indicará si es una factura de compra o de venta, por lo tanto se generan dos transformaciones idénticas variando únicamente el valor del campo indicador del tipo.

Esto se debe a que a diferencia de en la base de datos original, el Data Warehouse tendrá una única tabla para almacenar las facturas, haciendo uso de un campo para almacenar el tipo de factura.

Para tratar los casos en los que una factura debe ser actualizada por haber sido introducido erróneamente en el ERP de la empresa, será comunicado dicho error al personal encargado y será actualizado manualmente mediante una transformación implementada para dicha situación.
Ver figura B.15

6.1.1.2.4. Dimensión de Orden de Trabajo: Las órdenes de trabajo conforman una de las dimensiones que más tratado de datos requerirá para obtener los mismos bajo el esquema diseñado en anteriores fases del proyecto.

Existe un campo original que es indicador de la situación de la orden de trabajo: en garantía o no en garantía. Sin embargo, pese a que de cara al usuario en el sistema se presenta así, en la base de datos únicamente se almacena este campo cuando el valor es en garantía -además el valor será '1', por lo que hay que modificar dicho campo en el proceso-. Debido a ello es necesario tratar el caso en el que el valor de dicho campo es nulo y convertirlo a "No en garantía", de forma que el Data Warehouse no requiera procesos extra a la hora de obtener los datos.

Por otra parte, el campo que contiene si la orden de trabajo está cerrada o no, toma los valores 'Y' o 'N'. Esto será modificado y se sustituirán por los valores 'true' o 'false', sucesivamente.

Por último, se desea capitalizar los campos estado y producción, esto se llevará a cabo mediante Javascript ya que tiene funciones nativas para este propósito.

Ver figura B.18

6.1.1.2.5. Dimensión de personal: La carga de los empleados de la empresa requiere ciertas modificaciones principalmente dirigidas a la presentación de cara al usuario final. Los campos para el nombre y los apellidos son individuales en la base de datos empresarial, sin embargo tras un breve estudio se ha concluido que en el Data Warehouse es conveniente almacenarlos en un solo campo, correctamente formateado.

El resultado final se desea estructurado de forma "Apellido 1, Apellido 2, Nombre". Esto se condiciona a que es posible que no todos los apellidos existan en la base de datos, cubriendo toda casuística posible. Además el

resultado será capitalizado para mejorar la visibilidad del mismo en la interfaz del sistema.

Pese a esta transformación, se siguen almacenando los campos de forma individual. La principal ventaja de tener un campo único es que este es identificativo del empleado, facilitando así futuros usos de los datos en los cuadros de mando.
Ver figura B.20

6.1.1.2.6. Tabla puente de facturas y líneas de factura: Debido a la teoría del Data Warehousing[12], cuando un modelo que posee una relación de uno a varios no va a ser almacenado en la tabla de hechos, es necesario implementar una llamada tabla puente; en ella se almacenan pares de las claves correspondientes a la relación modelada.

En este caso, se necesita una tabla puente para enlazar facturas y líneas de facturas. En la primera versión del proyecto no se contempla dar uso a las líneas pero se ha decidido implementar su inclusión para acelerar su uso si en un futuro se decide incluirlas en alguna funcionalidad del sistema. Esta transformación solo tiene una pequeña particularidad, y es que es necesario haber cargado las dimensiones factura y línea de factura en el Data Warehouse, ya que en la tabla se almacenarán las claves primarias de las dimensiones, no las claves presentes en la base de datos origen.
Ver figura B.25

6.1.1.2.7. Tabla de hechos de OT: Esta transformación tiene como función crear y actualizar la tabla de hechos de órdenes de trabajo una vez que todas las dimensiones de las que depende han sido cargadas.

Este proceso se lleva a cabo siguiendo determinados pasos para cada dimensión. Una vez obtenidos los datos necesarios de la base de datos origen, mediante las claves primarias se hace la búsqueda de los mismos en el Data Warehouse, obteniendo la clave correspondiente. Una vez recogidos todos los identificadores, se introducen o actualizan los registros en la tabla de hechos del Data Warehouse.

Un detalle surgido durante la implementación ha sido que en algunas órdenes de trabajo se hace referencia a dos identificadores de empleado que sin embargo no se hallan en la base de datos. Se ha concluido que se debe a la eliminación posterior de los mismos, lo que ha sido confirmado posteriormente por la empresa contratante. Para resolver este problema se ha decidido ignorar las órdenes de trabajo enlazadas a alguno de estos empleados.
Ver figura B.21

6.1.1.2.8. Tabla de hechos de facturas: El procedimiento para poblar la tabla de hechos de facturas es el mismo que para la tabla de hechos de órdenes de trabajo. La particularidad de esta transformación reside en que el campo fecha de vencimiento es un vector en la base de datos, por lo tanto tenemos que extraer la fecha de la primera posición mediante Javascript.
Ver figura B.22

6.1.1.2.9. Trabajo de carga de dimensiones: El trabajo de carga de dimensiones nos facilita el hecho de programar el proceso ETL en su conjunto, ya que se ejecutan las fases de carga una a una, en un orden concreto que dará lugar a un Data Warehouse consistente. De haber algún fallo durante la ejecución, hay un paso que envía un correo electrónico a una dirección programada, este será recibido por el personal encargado del sistema para así poder actuar lo más rápido posible ante esta circunstancia.

Además de esto, el trabajo nos permite ser programado desde la misma interfaz de PDI, siendo el caso una ejecución diaria a las dos de la noche, de forma que a primera hora de la mañana los datos sean lo más recientes posible. En futuras iteraciones se contempla programar una carga parcial al mediodía, sin embargo esto requiere reuniones con el cliente para ser tratado y acordado de forma que no se generen inconsistencias durante el proceso.
Ver figura B.23

6.1.1.2.10. Trabajo de carga de tablas de hechos: Debido a que las transformaciones de tablas de hechos requieren de haber procesado previamente las dimensiones, se decide crear un trabajo separado para ellas. Este carece de programación ya que es llamado desde el trabajo de las dimensiones si el proceso ha transcurrido con éxito.
Ver figura B.24

6.1.1.3. Automatización

Una vez generada la estructura del Data Warehouse y verificados los resultados y consistencia de los procesos ETL, es necesario definir un lanzamiento periódico del sistema de carga de datos, para ello, PDI ofrece la opción de programar la ejecución de trabajos. Este procedimiento se llevará a cabo a las doce de la noche, horario en el que la empresa no está operativa. De esta forma, no habrá modificaciones en la base de datos origen mientras se ejecuta la extracción de datos, por lo que queda asegurado que no se vulnerará la integridad del Data Warehouse siempre y cuando los datos extraídos sean consistentes.

Es destacable mencionar que en este proceso se define un envío de correo a una dirección corporativa, en el caso de que alguna de las transformaciones falle; de esta forma se podrán atender con celeridad los posibles problemas que puedan surgir durante el uso del sistema en el entorno de producción.

6.1.2. OLAP: Mondrian

Sobre el Data Warehouse ya definido y creado en el sistema de base de datos correspondiente (PostgreSQL), es necesario implementar una interfaz Mondrian para poder realizar un acceso multidimensional a los datos.

Esta implementación se realiza mediante un fichero xml siguiento el esquema correspondiente, para después ser cargado en el servidor Pentaho. En él, se deben definir diferentes elementos que darán cuerpo a los cubos OLAP, siendo estos:

- Dimensiones: Con el elemento Dimension se especificarán los datos necesarios para que una dimensión quede definida, siendo necesario indicar la tabla de la base de datos y la clave primaria. A esto se le añaden jerarquías y niveles, mediante las cuales los datos quedarán organizados con estructura padre-hijo, lo que permite generar análisis flexible de las dimensiones.

```
<Dimension type="TimeDimension" visible="true" highCardinality="false" name="Tiempo">
  <Hierarchy name="Jerarquia de fechas" visible="true" hasAll="true" primaryKey="dim_date_key">
    <Table name="dim_date" schema="public">
    </Table>
    <Level name="Anio" visible="true" table="dim_date" column="year4" nameColumn="year4" type="Numeric" uniqueMembers="false" levelType="TimeYears" hideMemberIf=
    "Never">
    </Level>
    <Level name="Cuatrimestre" visible="true" table="dim_date" column="year_quarter" nameColumn="year_quarter" type="String" uniqueMembers="false" levelType=
    "TimeQuarters" hideMemberIf="Never">
    </Level>
    <Level name="Mes" visible="true" table="dim_date" column="month_number" nameColumn="month_name" type="Numeric" uniqueMembers="false" levelType="TimeMonths"
    hideMemberIf="Never">
    </Level>
    <Level name="Etiqueta de fecha" visible="true" table="dim_date" column="date_value" nameColumn="date_medium" type="String" uniqueMembers="false" levelType=
    "TimeDays" hideMemberIf="Never">
    </Level>
    <Level name="Dia del mes" visible="true" table="dim_date" column="day_in_month" nameColumn="day_in_month" type="Numeric" uniqueMembers="false" levelType=
    "TimeDays" hideMemberIf="Never">
    </Level>
  </Hierarchy>
</Dimension>
```

Figura 6.3: Dimensión de fecha

- Cubos: El epicentro de un análisis OLAP, en Mondrian el cubo se define indicando qué dimensiones de las creadas anteriormente se van a utilizar con el fin de obtener los datos, además de indicar la tabla de hechos en la que se hayan las claves de las dimensiones. A esto se le añaden las medidas que se explican en el siguiente punto.

```
<Cube name="Notice" visible="true" cache="true" enabled="true">
  <Table name="fact_notice" schema="public">
  </Table>
  <DimensionUsage source="Lineas de aviso" name="Lineas de aviso" visible="true" foreignKey="service_line_key" highCardinality="false">
  </DimensionUsage>
  <DimensionUsage source="Cliente" name="Cliente" visible="true" foreignKey="customer_key" highCardinality="false">
  </DimensionUsage>
  <DimensionUsage source="Articulo" name="Articulo" visible="true" foreignKey="article_key" highCardinality="false">
  </DimensionUsage>
  <DimensionUsage source="Empleado" name="Empleado aviso" visible="true" foreignKey="service_staff_key" highCardinality="false">
  </DimensionUsage>
  <DimensionUsage source="Empleado" name="Empleado linea aviso" visible="true" foreignKey="service_line_staff_key" highCardinality="false">
  </DimensionUsage>
  <DimensionUsage source="Servicio" name="Servicio" visible="true" foreignKey="service_key" highCardinality="false">
  </DimensionUsage>
  <DimensionUsage source="Tiempo" name="Fecha servicio" visible="true" foreignKey="dim_date_key" highCardinality="false">
  </DimensionUsage>
  <DimensionUsage source="Tiempo" name="Fecha linea servicio" visible="true" foreignKey="dim_date_line_key" highCardinality="false">
  </DimensionUsage>
  <Measure name="Numero OTs" column="count_notices" aggregator="sum" visible="true">
  </Measure>
  <Measure name="Duracion servicio" column="repair_duration" aggregator="sum" visible="true">
  </Measure>
</Cube>
```

Figura 6.4: Cubo de avisos

- Medidas y medidas calculadas: Las medidas básicas son sencillas, se limitan a apuntar a una columna de la tabla de hechos y definen un agregador. Sin embargo, las medidas calculadas dan gran flexibilidad a la hora de crear indicadores, ya que se pueden emplear diferentes operaciones matemáticas y utilizar datos de las diferentes dimensiones. Un ejemplo sencillo es obtener el porcentaje de horas realizadas por un trabajador sobre el total de la plantilla.

```
<CalculatedMember name="Porcentaje de categoria" formatString="###,###%" dimension="Measures">
  <Formula>
    <![CDATA[([Articulo.Categorias].CurrentMember, Measures.[Duracion servicio]) /
            ([Articulo.Categorias].CurrentMember.Parent, Measures.[Duracion servicio])]]>
  </Formula>
</CalculatedMember>
```

Figura 6.5: Medida calculada

6.1.3. Administración: Gestión de usuarios y accesos

El sistema desarrollado maneja información delicada de la empresa, y por políticas internas, esta no debe poder ser visualizada por cualquier usuario que tenga acceso a la aplicación. Por este motivo se hace necesario implementar una gestión de acceso para los usuarios.

El servidor BI Pentaho da soporte nativo a la gestión de permisos, mediante roles y usuarios que serán asociados a dichos roles. Desde su interfaz de Administración, solo accesible a aquellos usuarios con el rol de administrador, se puede modificar, crear y eliminar tanto usuarios como roles. Se ha hecho uso de esta funcionalidad para implementar los requisitos del cliente, que son en una primera aproximación, roles de empleado y de gerente.

Los empleados solo tendrán acceso al cuadro de mando de Órdenes de Trabajo por empleado, y pudiendo visualizar únicamente los datos correspodientes a su usuario. Los gerentes, en cambio, podrán tener acceso a todos los cuadros, sin restricción alguna. Este control se realiza mediante Javascript: los cuadros de mando tienen funciones propias para obtener el usuario y el rol, de esta forma se ocultan enlaces y se modifican diversos parámetros para personalizar la información dispuesta en el sistema.

6.1.4. Personalización y modificación de los componentes

Con el objeto de adecuar el sistema a la casuística del proyecto y la empresa productora, se han llevado a cabo varias modificaciones en el servidor BI de Pentaho. Como personalización visual del sistema, se ha incluído el logo empresarial en la pantalla de inicio de sesión y en el menú superior de la interfaz del sistema.

Dejando de un lado estos pequeños cambios de interfaz, las principales modificaciones han implicado cambios en complementos del sistema, ampliando o mejorando funcionalidades ya existentes. Los cambios realizados han sido los siguientes:

6.1.4.1. Export Button

Este complemento nos permite añadir un botón configurable que exportará el Datasource indicado al formato de fichero que se defina. Sin embargo, la configuración del mismo es limitada ya que solo ofrece elegir el formato de salida del fichero. Debido a los requisitos del cliente, se necesita obtener una hoja de cálculo en formato Microsoft Excel, con los datos ordenados por campos concretos y pudiendo elegir el nombre del fichero.

Para obtener esta funcionalidad, se ha modificado el código Javascript de este complemento, añadiendo varios parámetros a su configuración que se podrán determinar en el código pre-execution, dentro de los ajustes que hay disponibles en la interfaz de CDE. Estos parámetros determinan el nombre de fichero, número de columna y orden en el que se quieren ordenar los datos, y si se quiere solicitar el nombre del fichero mediante um prompt de Javascript. De no estar determinado alguno de estos parámetros, el complemento llevará a cabo su funcionamiento por defecto.

```
function f(){
    this.parameters.filename = 'HorasPorEmpleado-Año'+p_tiempo.substring(0,4)+'.xls';
    this.parameters.sort = '13D';
    }
```

Figura 6.6: Código de parametrización

6.1.4.2. Date Range Picker

Entre los complementos de selector que nos ofrece Pentaho, está el selector de rango de fechas, nos permite elegir entre diversos rangos predefinidos, o introducir nosotros un rango mediante dos calendarios desplegables.

Este selector nos ofrece una funcionalidad muy útil, sin embargo, tiene un problema importante de cara a crear un sistema de cuadros de mando con interfaz consistente, y es que su intefaz está en inglés, y no se modifica dependiendo de la localización del navegador. El componente hace uso del selector de fechas de JQuery, que da la opción de cambiar el idioma en el que se mostrará. Además de esto, posee unos textos indicativos de los rangos de fecha predefinidos, que se indican en el código Javascript del complemento. Para abordar esta problemática, se analizan los idiomas establecidos en el navegador; si el castellano se encuentra entre ellos, se realiza la función JQuery necesaria para modificar el idioma de los selectores de fecha, y se modifican las cadenas de texto que identifican los rangos predefinidos.

Figura 6.7: Componente modificado

6.2. Cliente

Una vez el servidor está funcional y el Data Warehouse ha sido poblado, se procede a implementar el conjunto de cuadros de mando definidos en las fases de análisis y diseño, siguiendo los requisitos funcionales indicados por el cliente.

Cabe mencionar que este proceso no ha sido complemente aislado del desarrollo de los elementos del servidor, si no que necesidades surgidas durante la creación de cuadros de mando han retroalimentado principalmente el Data Warehouse, añadiendo o modificando partes del proceso ETL.

6.2.1. Saiku Analytics

Saiku Analytics permite a los usuarios avanzados explorar de una forma mucho más técnica y flexible los datos, usando una interfaz drag & drop intuitiva pero a la vez potente.

Para facilitar la integración con este sistema, Saiku Analytics tiene un plugin instalable desde el market de Pentaho; como casi todo el ecosistema de Pentaho disponemos de una versión libre, que es la utilizada para el proyecto, y de una versión empresa que nos ofrece funcionalidades ampliadas y sobre todo mayor libertad de personalización.
[13]

Entre otras utilidades, podremos analizar de una manera gráfica y veloz cual es la distribución de la actividad de nuestra empresa en un mapa del mundo como puede verse en la figura 6.8.

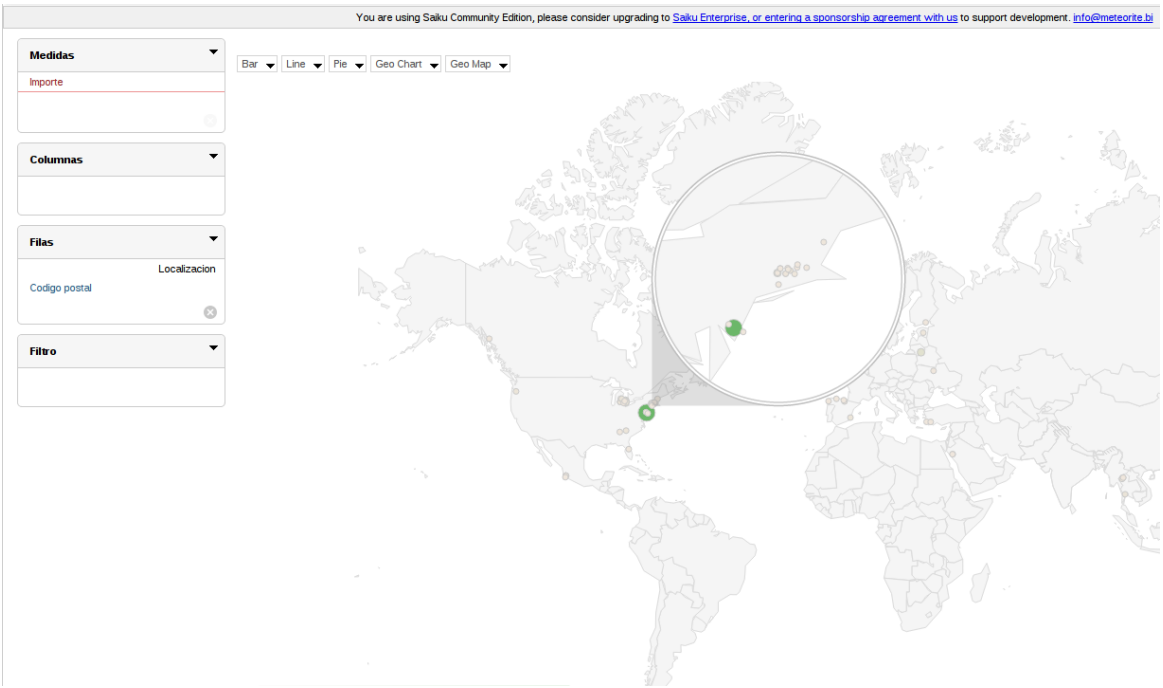


Figura 6.8: Distribución georáfica de las ventas

En las figuras 6.10 y 6.9 vemos la captura de un ejemplo de consulta del sistema Saiku implementado para este proyecto. En ellas se puede apreciar como esta herramienta nos permite obtener visualización de forma ágil con muy pocos clicks, pudiendo ajustar la consulta de los datos que queremos visualizar aplicando filtros con gran flexibilidad. Además, es posible utilizar manualmente el lenguaje de consulta MDX, véase la figura **??**.

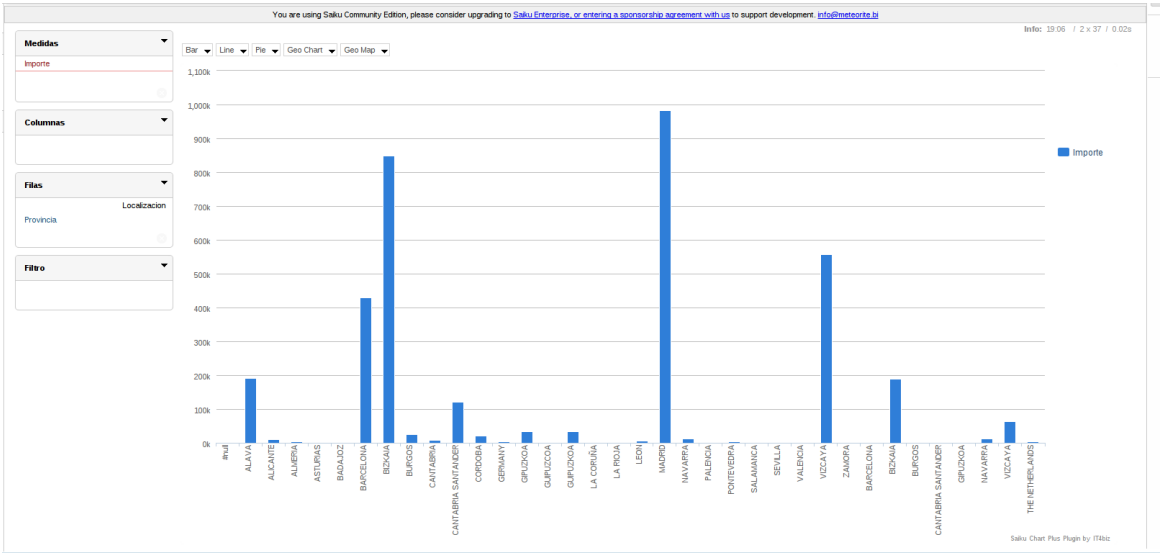


Figura 6.9: Gráfico de barras

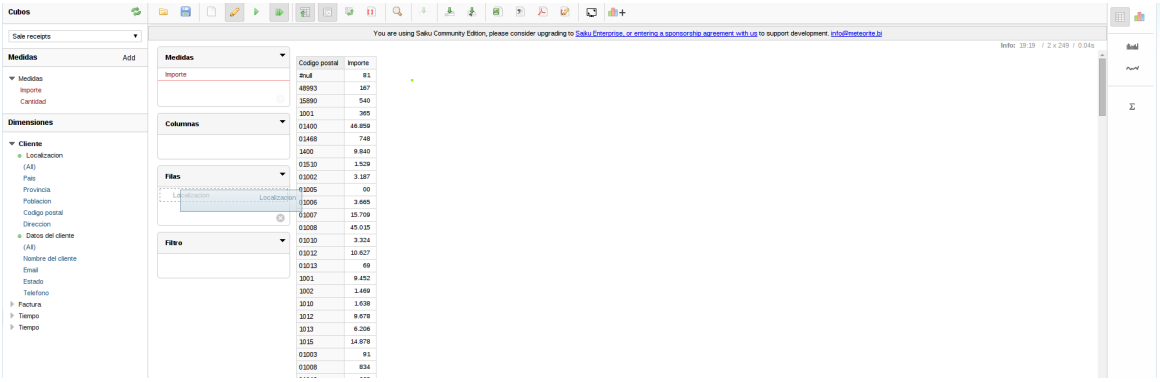


Figura 6.10: Consultas personalizadas

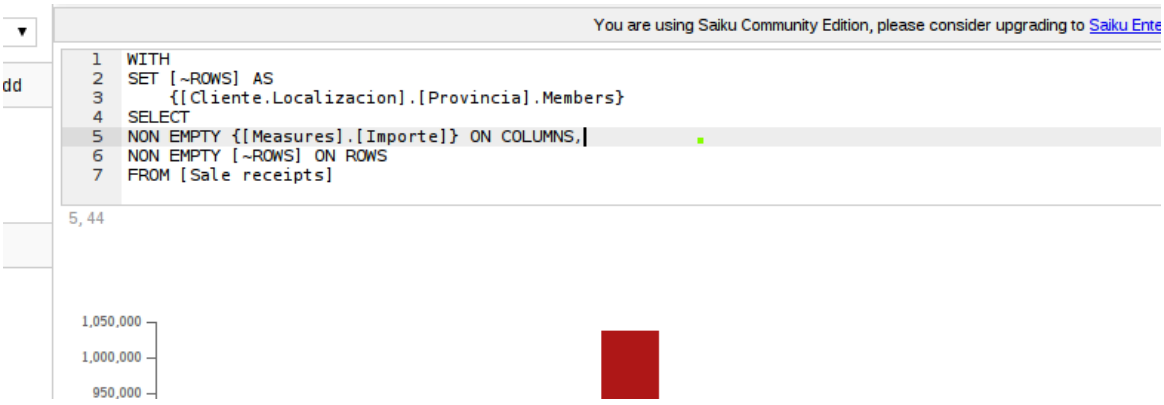


Figura 6.11: Consultas MDX personalizadas

6.2.2. Dashboards

La implementación de un dashboard mediante el plug-in Community Dashboard Editor del servidor BI Pentaho se divide en tres apartados:

- Estructura del dashboard ("Layout"): En esta sección se crea la disposición en la que se distribuirán los componentes, así como es posible añadir código HTML propio y aplicar estilos personalizados o del framework Bootstrap.

- Componentes: Aquí se definen los elementos que formarán la interfaz funcional y visual del cuadro de mando. No solo se genera una interfaz de visualización pasiva, si no que mediante código Javascript en los componentes se genera interactividad entre componentes e incluso entre cuadros de mando.
- Fuentes de datos: Finalmente, se deben indicar los datos que son utilizados por los componentes. En este apartado, mediante las conexiones definidas anteriormente se implementan sentencias en los lenguajes requeridos para obtener los datos necesarios. Estas fuentes de datos pueden utilizar parámetros que son enviados desde los componentes del cuadro de mando.

Para aportar un nivel de interactividad superior y lograr un sistema que vaya más allá de mostrar y filtrar datos, se ha hecho un uso intensivo de código Javascript. El entorno Pentaho provee de herramientas muy potentes para este propósito: se puede gestionar el dashboard a nivel global o cada componente uno a uno. Es posible decidir en qué momento o bajo qué circunstancias se ejecutarán las funciones creadas: al cargar el cuadro de mando, tras cambiar un parámetro, al hacer click en un sector de un gráfico, etcétera. A todas estas posibilidades también se les añade la potencia de poder utilizar la librería JQuery.

Se ha implementado el uso del botón de exportación en todos aquellos cuadros que poseen tablas de datos. Cada uno de estos botones ha sido parametrizado vía Javascript, con las utilidades generadas al modificar el complemento, de forma que el nombre del archivo sea consistente con los datos exportados, y además, estén ordenados de forma intuitiva.

A continuación se describen los diferentes cuadros de mando implementados, así como las funcionalidades dinámicas que han sido creadas para maximizar la utilidad de los mismos de cara al usuario final.

6.2.2.1. Dashboard 1: Visión global de avisos

La tabla de resumen de órdenes de trabajo por empleado está programada para que, en el caso de pulsar en alguno de los registros, seamos redireccionados al cuadro de mando de OTs por empleado, parametrizado por el empleado que hayamos seleccionado. El gráfico situado a la derecha posee la misma funcionalidad que esta tabla.

Los gráficos de órdenes de trabajo por estado y producción tienen ambos implementadas dos funcionalidades diferentes pero ejecutadas bajo las mismas circunstancias. Pulsando en alguna de las barras de los gráficos, se tomará el valor seleccionado y se filtrará la tabla adjunta por el campo estado o producción correspondiente. Además, el gráfico presente se ocultará para dar paso a un gráfico de análisis temporal del valor seleccionado. Volviendo a pulsar sobre el gráfico se reestablecerán los parámetros y serán retirados los filtros.

```
function f(scene) {

    color = this.pvMark.fillStyle().color;

    var vars = scene.vars;
    var c = vars.category.value;
    var v = vars.series.value;

    Dashboards.fireChange('p_estado',v);
    Dashboards.fireChange('p_chart_color',color);

    document.getElementById("Panel4").setAttribute("hidden","true");
    document.getElementById("PanelHiddenEstado").removeAttribute("hidden","false");


    if(p_periodo=="Anio")
    {
        render_table_status_detail.chartDefinition.dataAccessId="sql_status_detail_table_year";
    }
    else
    {
        render_table_status_detail.chartDefinition.dataAccessId="sql_status_detail_table_quarter";
    }
}

```

Figura 6.12: Código del gráfico de estado

```
function f(){
    document.getElementById("PanelHiddenEstado").setAttribute("hidden","true");
    document.getElementById("Panel4").removeAttribute("hidden","false");
    Dashboards.fireChange('p_estado',p_estado);

    if(p_periodo=="Anio")
    {
        render_table_status_detail.chartDefinition.dataAccessId="sql_status_detail_table_year_nofilter";
    }
    else
    {
        render_table_status_detail.chartDefinition.dataAccessId="sql_status_detail_table_quarter_nofilte
    }
}
```

Figura 6.13: Código del gráfico de evolución temporal del estado

Los parámetros de este cuadro de mando sirven para seleccionar qué medida temporal se utilizará para el análisis (año o trimestre) y qué valor de dicha medida será utilizado en los componentes. Se encuentran programados para mostrar el último trimestre del que se dispongan datos.

6.2.2.1.1. Problemática surgida: Debido a que este cuadro de mando ha sido el primero con el que se ha empezado a desarrollar, es el que más problemas ha acarreado ya que se partía de unos conocimientos limitados del sistema. Por contrapartida, este hecho ha conllevado que los siguientes desarrollos tengan una más rápida solución ante problemas comunes.

El primer bache encontrado al implementar el cuadro de mando ha estado en desarrollar una interfaz visualmente agradable pero sobre todo, fácil de modificar mientras se está desarrollando. En un principio únicamente se utilizaban los componentes básicos del servidor Pentaho, que están un tanto limitados. Esto añadido a la ausencia de documentación al respecto ha dificultado mucho llegar a dar con una metodología adecuada.

Este problema ha sido solucionado al encontrar que Pentaho habilita nativamente el uso del framework de desarrollo web Bootstrap. Esto nos permite mediante clases predefinidas crear elementos visuales como cabeceras, espacios para parámetros, pestañas de navegación o paneles. Con esto se facilita en gran medida la creación de una buena interfaz ya que el desarrollo se centra en saber qué se quiere mostrar y de qué forma, no en cómo lograr implementar esto.

El siguiente escollo ha radicado en el uso de parámetros dentro de los cuadros de mando. Pentaho nos permite definir parámetros, variables usables a lo largo del dashboard, tanto para lectura como para escritura. Sin embargo para que los componentes puedan hacer uso de ellos es necesario modificar varias de sus propiedades, dependiendo de si se desea acceso solo de lectura o de escritura. Además, si se quiere acceder a estos vía Javascript y modificarlos, es necesario utilizar funciones específicas que proporciona el entorno de Pentaho:

```
function f(scene) {
    var vars = scene.vars;
    var s = vars.series.value;
    var c = vars.category.value;
    var v = vars.value.value;

    Dashboards.fireChange('param_fecha',c);
}
```

Figura 6.14: Código necesario para modificar un parámetro en Javascript

Una vez resueltos estos dos problemas, el desarrollo ha empezado a ser más fluido, pasando a ser los contratiempos más específicos de la funcionalidad concreta que se está implementanto, y no de manejo global de la plataforma.

6.2.2.2. Dashboard 2: Avisos por empleado

La funcionalidad dinámica de este cuadro de mando radica en poder profundizar (drill-down) en los datos que se muestran en cada momento.

En la tabla de órdenes de trabajo no cerradas se puede hacer click sobre los registros de órdenes de trabajo. Se desplegará una sub-tabla oculta que muestra las líneas de orden de trabajo imputadas sobre la seleccionada, con diversa información al respecto.

Al pulsar sobre algún día del gráfico de tiempos, se abrirá un pop-up, que mostrará las líneas de orden de trabajo imputadas por el trabajador en dicha fecha. En el caso del gráfico de clasificación por garantía, se visualizan las líneas correspondientes a dicha situación de OT.

Como es habitual, el cuadro de mando de órdenes de trabajo por empleado implementa un selector de fechas. Sin embargo, en este caso también se incluye un selector de empleado; este será habilitado únicamente para aquellos usuarios con permisos de Gerente. Los usuarios que entren como Empleado solo tendrán acceso a los datos correspondientes al empleado asociado a su nombre de usuario, el selector no mostrará más valores.

6.2.2.2.1. Problemática surgida: En este cuadro es donde ha surgido por primera vez la conveniencia de hacer uso de componentes dentro de pop-ups. Aún no se había utilizado esta funcionalidad de Pentaho por lo que ha sido necesario realizar una exhaustiva búsqueda de información en la web. Las pautas halladas no fueron necesarias, y se ha hecho necesario realizar diversas pruebas hasta dar con la solución.

El uso de estos componentes requiere de tres elementos. Por una parte, se debe definir un panel en el apartado “Layout”, este panel requiere de tener el atributo HTML “hidden” a “true”.

Figura 6.15: Código HTML para el componente pop-up

Una vez hecho esto, se define un “Pop-up component” de la siguiente manera, añadiéndolo al panel que acabamos de crear:

Properties / Advanced Properties

Property	Value
Name	popup_realizadas
Listeners	[]
Gravity Parameter	Top
Parameters	[["param_fecha","par (...)
Priority	5
Draggable	True
Horizontal scrollbar	False
Vertical scrollbar	False
Close on click outside	True
HtmlObject	${p:Panel5}
Execute at start	True
Pre Execution	
Post Execution	
Tooltip	

Figura 6.16: Configuración del componente pop-up

Por último, solo queda crear el componente que deseamos mostrar en el pop-up, y añadirlo también al panel creado:

Properties / Advanced Properties

Property	Value
Name	tabla_realizadas
Listeners	['param_fecha']
Column Headers	[]
Parameters	[["param_fecha","par (...)
Column Types	[]
Datasource	query_tabla_realizadas
HtmlObject	Panel5
clickAction	

Figura 6.17: Componente tabla que se mostrará

6.2.2.3. Dashboard 3: Ventas globales

Este cuadro de mando tiene la capacidad de en la tabla de ingresos por clientes, haciendo click sobre alguno de ellos se muestre un gráfico de detalle sobre este cliente. En él aparecerán las ventas anuales desglosadas por meses, en los años mostrados en la tabla.

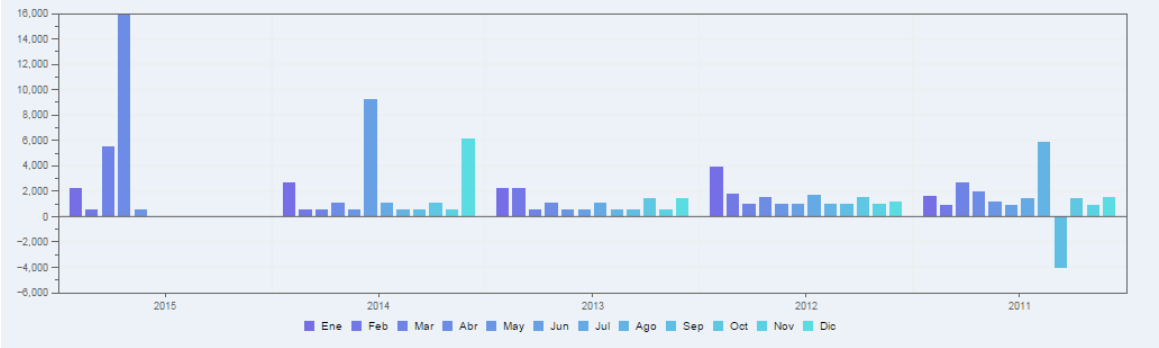


Figura 6.18: Gráfico que aparece al desplegar un cliente

Los dos selectores de fecha modifican el rango temporal de los gráficos de cinco mejores y cinco peores clientes. El resto de componentes no se hayan parametrizados, a excepción del gráfico de desglose del cliente.

6.2.2.3.1. Problemática surgida: El desarrollo de este cuadro de mando ha resultado sencillo tras los dos anteriores, debido a que carece de elementos de interacción complejos. Sin embargo, el gráfico de desglose temporal del cliente seleccionado ha supuesto un reto importante, no en su implementación sino en el origen de datos. Debido a las exigencias visuales (meses en formato Ene, Feb...) y el límite de años establecido, ha derivado en una sentencia SQL de complejidad considerable:

```
SELECT CASE WHEN sel.mes=1 THEN CAST('Ene' AS varchar(3))
 WHEN sel.mes=2 THEN CAST('Feb' AS varchar(3))
 WHEN sel.mes=3 THEN CAST('Mar' AS varchar(3))
 WHEN sel.mes=4 THEN CAST('Abr' AS varchar(3))
 WHEN sel.mes=5 THEN CAST('May' AS varchar(3))
 WHEN sel.mes=6 THEN CAST('Jun' AS varchar(3))
 WHEN sel.mes=7 THEN CAST('Jul' AS varchar(3))
 WHEN sel.mes=8 THEN CAST('Ago' AS varchar(3))
 WHEN sel.mes=9 THEN CAST('Sep' AS varchar(3))
 WHEN sel.mes=10 THEN CAST('Oct' AS varchar(3))
WHEN sel.mes=11 THEN CAST('Nov' AS varchar(3))
 ELSE CAST('Dic' AS varchar(3)) END AS Mes, sel.Ano AS "Año", sel.Importe

 FROM
(
 SELECT extract(month from dim_date.date_value) AS Mes,
 extract(year from dim_date.date_value) AS Ano,

 SUM(fact_sale_receipt.amount_taxes) AS Importe
 FROM fact_sale_receipt
 LEFT JOIN dim_date ON fact_sale_receipt.receipt_date_key = dim_date.dim_date_key
 INNER JOIN dim_customer
 ON dim_customer.customer_key = fact_sale_receipt.customer_key
 WHERE dim_customer.customer_first_name=${columnaCliente}
 AND extract(year from dim_date.date_value)
 IN (extract(year from now()), extract(year from now())-1, extract(year from now())-2, extract(year from now())-3, extract(year from now())-4)
 GROUP BY extract(year from dim_date.date_value), extract(month from dim_date.date_value)
 ORDER BY  extract(month from dim_date.date_value) ASC)
 AS sel
 ORDER BY sel.Mes asc, sel.Ano desc
```

Figura 6.19: Gráfico que aparece al desplegar un cliente

6.2.2.4. Dashboard 4: Ventas vs. Avisos (localización)

En el cuadro de comparación de ventas contra avisos de nuevo se ha implementado la funcionalidad de los pop-ups. En este caso, al pulsar sobre alguna de las localidades en ambos gráficos, se abrirá una tabla con las órdenes de trabajo realizadas en dicha localidad, en el rango de fechas seleccionado.

De nuevo se hace uso de un selector de rango de fechas, que filtra ambos gráficos. Además de esta parametrización se dispone de un filtro de provincias, el cual se hace necesario dada la cantidad de diferentes poblaciones que hay registradas en este contexto.

6.2.2.5. Dashboard 5: Análisis de artículos

La funcionalidad de este cuadro de mando radica en el uso de parámetros de filtrado tanto para las fechas como para los datos que se muestran. En concreto, el filtrado se realiza por dos campos diferentes pero relacionados: el nivel de categoría que se desea analizar, y el valor de la categoría a usar como filtro.

De esta forma, es posible hacer "drill-down" en los datos, pasando de una vista global de todos los artículos manejados en el sistema a un subconjunto de interés para el usuario. Además, para mejorar la interacción, se incluye la opción de reestablecer los filtros eligiendo el valor "Todos" en el selector de categoría.

6.3. De-identificación de datos

La deidentificación (traducción literal del término inglés "de-identification") es un concepto conocido en la medicina[14], que sin embargo está probándose ser cada vez más relevante en el ámbito de la Minería de Datos[6], entre otras razones, porque habilita la liberación de datos que de otra forma no podrían ser hechos públicos.

6.3.1. Motivación

El ámbito en este proyecto es diferente: forma parte del sistema desarrollado por los autores para una empresa cliente. Ello implica que los datos almacenados en el Data Warehouse son reales, e incluyen información personal e identificativa sobre sujetos y empresas. Sin embargo, la función de la deidentificación viene a ser la mencionada con anterioridad: poder habilitar los datos a terceras personas sin vulnerar leyes ni principios morales.

En este contexto, nos encontramos con dos límites legales que hacen necesario tener una base de datos completamente anónima: por una parte, el acuerdo de confidencialidad firmado con la empresa empleadora, y por otra, la Ley Orgánica de Protección de Datos.

El contrato de confidencialidad con la empresa es un acuerdo legal en el que el empleado se compromete a cumplir varias cláusulas, a destacar la responsabilidad sobre la custodia de la información y la obligación de no divulgación de la misma. De ser incumplida alguna de las cláusulas se habilita a la parte no infractora a emprender acciones legales dentro del marco legislativo nacional.

Los artículos de la LOPD que atañen al proyecto se consideran fuera del alcance de este documento, ya que han sido eliminados todos aquellos datos que puedan en cualquier caso permitir identificar a una persona o institución real. Por ello, se considera que en ningún caso existirán datos sobre los que aplicar dicha ley.

6.3.2. Análisis y desarrollo

Para llevar a cabo el proceso de deindentificación, se ha realizado un análisis del Data Warehouse en busca de qué datos identificativos se almacenan. Una vez detectados los campos que deben ser modificados, se decide la estrategia a utilizar para volver estos datos anónimos.

Tabla	Campo	Tratamiento
dim_staff	staff_first_name	generar valores de formato "Empleado #"
	staff_last_name	eliminar datos
dim_customer	customer_first_name	generar valores de formato "Cliente #"
	customer_last_name	eliminar datos
	customer_email	generar valores de formato "diente_#@clientes.com"
	customer_phone_number	generar números aleatorios
	customer_address	eliminar datos
dim_partner	partner_first_name	generar valores de formato "Proveedor #"
	partner_last_name	eliminar datos
	partner_email	generar valores de formato "proveedor_#@proveedores.com"
	partner_phone_number	generar números aleatorios
	partner_address	eliminar datos
	partner_fiscal_name	asignar valor de partner_first_name
dim_service	service_origin_staff_name	eliminar datos
dim_receipt	billing_customer_name	eliminar datos
	billing_address	eliminar datos

Figura 6.20: Tablas, campos y estrategia de modificación

Una vez se tienen qué campos, de qué origen, y cómo van a ser modificados, se implementa una transformación única que lleve a cabo estos cambios. No es necesario ningún cambio secuencial por el hecho de que las relaciones están creadas mediante claves foráneas, no hay referencias a campos informativos entre tablas.

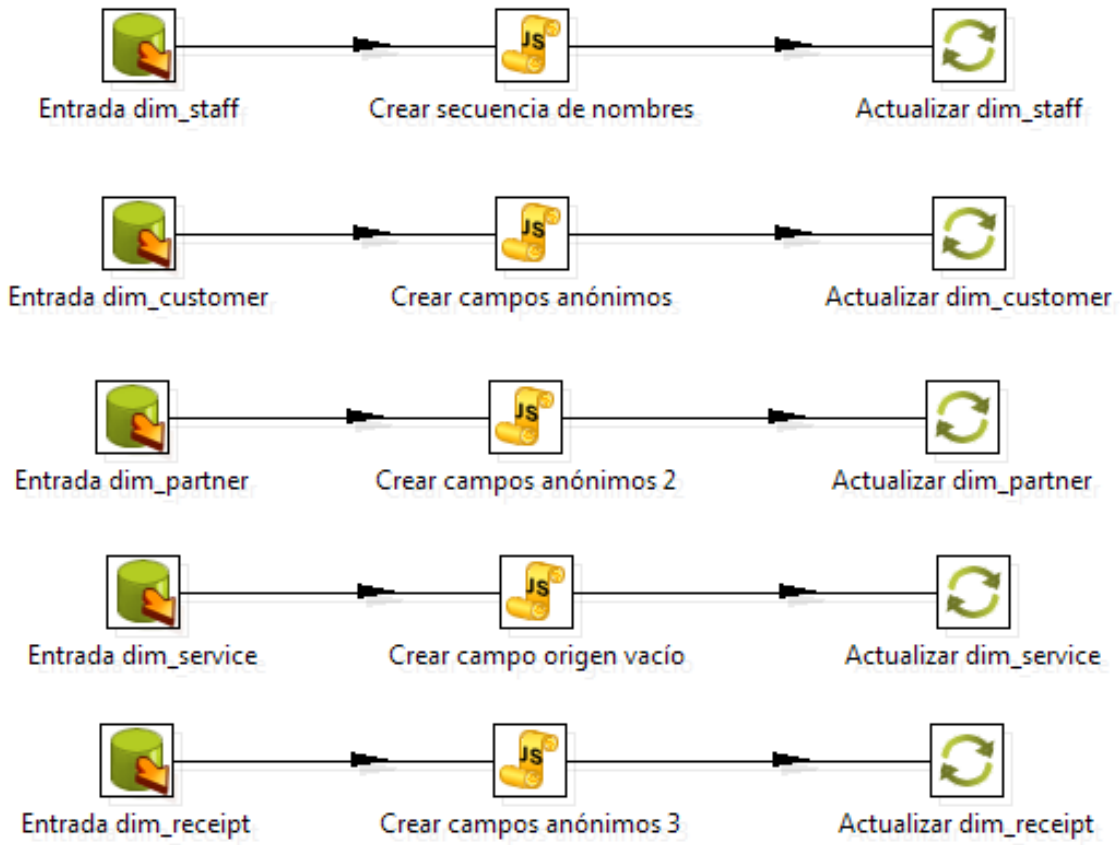
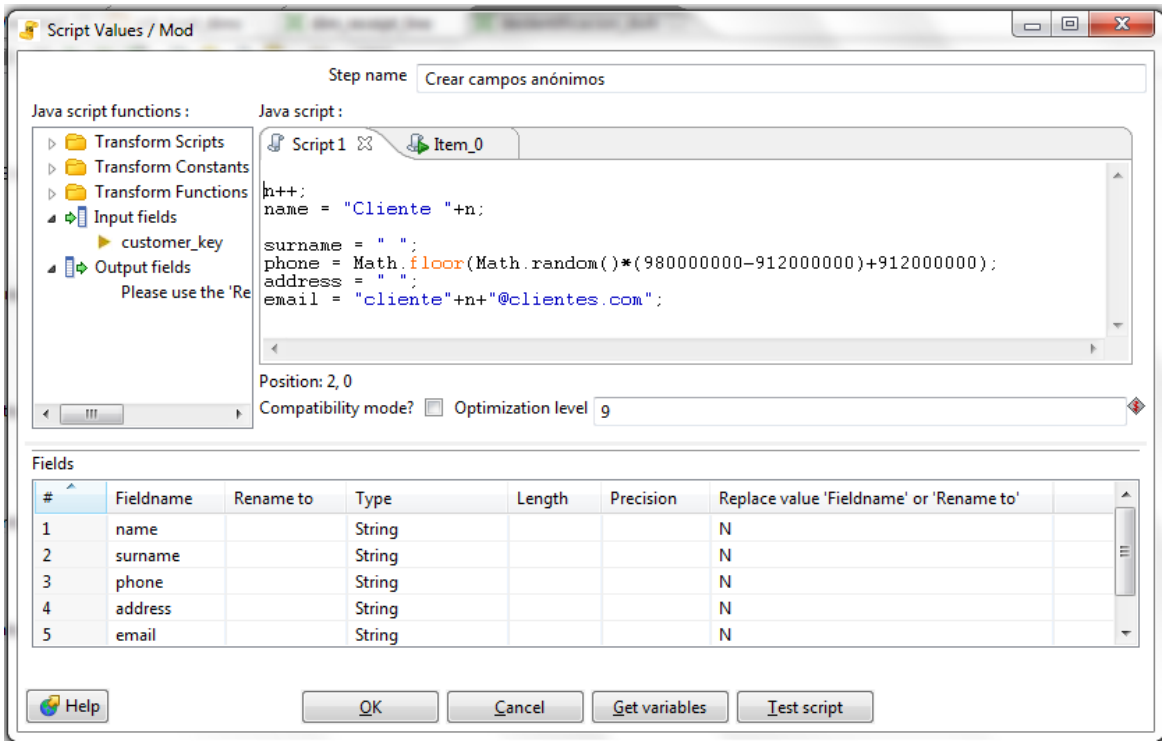


Figura 6.21: Transformación encargada de todo el proceso ETL de deidentificación

En este proceso se hace uso de la potencia del paso Javascript de PDI, con él generamos todos los campos que queremos modificar en las dimensiones. También nos facilita el crear los nombres con el formato "Empleado 1, Empleado 2... Empleado N", ya que permite crear un script inicial. En él creamos una variable "n" que será asignada a cero, y por cada fila procesada se le añadirá uno a su valor, creando así una secuencia de

nombres modificando la cifra.



Capítulo 7

Minería de datos

Estamos inundados con cantidades ingentes de datos generados por las miles de transacciones que generamos cada hora. El volumen de datos que se genera en nuestro entorno está en continuo crecimiento, y no parece que esta tendencia se vaya a detener a corto-medio plazo.

El hardware que actualmente existe en el mercado posibilita guardar los datos que no hace tanto hubiéramos desechado. Además de sus capacidades, el precio actual de estos dispositivos y el almacenamiento en la nube nos permite posponer las decisiones sobre qué hacer con estos datos. Nuestras acciones permanecen almacenadas en los registros de las distintas empresas en las que participamos como clientes, se almacenan nuestras compras en el supermercado, nuestros movimientos financieros y los datos de geoposicionamiento transmitidos desde nuestros *Smarthphones*.

La *World Wide Web* (WWW) nos abruma con información, además todos nuestros movimientos a través de la red quedan almacenados en forma de registro en una base de datos. Es incuestionable que estamos ante un nuevo cambio, somos testigos de la evolución desde la era de Internet y los datos hacía la generación del conocimiento. Sin embargo a medida que los datos crecen inexorablemente la cantidad de personas capaces de entender y procesar esos datos no aumenta en la misma proporción. En todos estos datos permanece oculta información potencialmente utilizable y todavía son pocas las empresas que explotan todo el potencial que reside en sus sistemas de almacenamiento.

El desenfrenado crecimiento de las bases de datos en los últimos años, bases de datos para almacenar las elecciones de los clientes, los emails que envían al servicio de atención telefónica y aquellas que albergan nuestras opiniones y sentimientos expresados a través de las redes sociales, pone la minería de datos a la vanguardia de las nuevas tecnologías de negocio en cuanto a potencial económico.

Cuando hablamos de minería de datos nos estamos refiriendo a la búsqueda de conocimiento entre toda esta cantidad de datos que albergan las bases de datos u otras tecnologías de almacenamiento de la información. Estas técnicas constituyen una herramienta por si mismas que nos proporciona la consecución de predicciones futuras apoyadas en la inferencia de los datos que se generan en el pasado a lo largo del tiempo.

Dentro de la minería de datos existen técnicas de clasificación que nos permiten agrupar muestras de acuerdo a criterios, estás técnicas son conocidas como clasificación supervisada y no supervisada. La clasificación es una técnica ampliamente utilizada en diversos campos como el reconocimiento de patrones o la segmentación, sin embargo, cada una de las técnicas está enfocada a un propósito diferente: a grandes rasgos, la clasificación supervisada radica en predecir información, y la no supervisada, en descubrirla.

7.1. Motivación y contexto de negocio

Tal y como se ha descrito anteriormente en este documento, el propósito de la clasificación no supervisada consiste en, opuestamente a la clasificación supervisada, descubrir información, en lugar de predecirla. Explorar los datos en busca de patrones de comportamiento. Los campos y casos a los que es aplicable son múltiples y diversos: en sociología, análisis genético, reconocimiento facial, etcétera. Todo área de conocimiento capaz de recopilar gran cantidad de datos es susceptible y puede beneficiarse de metodologías de clustering.

Tras una fase de análisis sobre los datos disponibles en la base de datos del cliente, se ha determinado que el concepto sobre el que más datos indicativos se pueden recabar es el cliente, así como otros hechos de negocio que giran en torno a este: facturas y servicios.

Partiendo de esta base, se consideran dos técnicas comunmente empleadas dada esta información: segmentación de clientes o predicciones sobre futuros clientes. Se decide realizar ciertas preguntas a la empresa contratante y resulta indicativo el hecho de que la clientela es relativamente invariante, es decir, no se incorpora una cantidad significativa de nuevos clientes. Dado esto, se decide que la segmentación de clientes encaja mejor en el proyecto.

El propósito principal de una segmentación de clientes es clasificarlos en diferentes conjuntos en base a rasgos comunes, de forma que se puedan tomar distintas estrategias de cara a cada grupo, acercándose así al ideal de un trato individualizado para cada cliente.[8]

7.2. Diseño

El diseño de éste módulo esta construido bajo el mismo marco de modularidad con el que construimos el resto del proyecto, permitiendo el desarrollo por los dos componentes del equipo.

En la 7.1 está representado el mapa de diseño donde se muestran las dependencias y se documentan las rutinas.

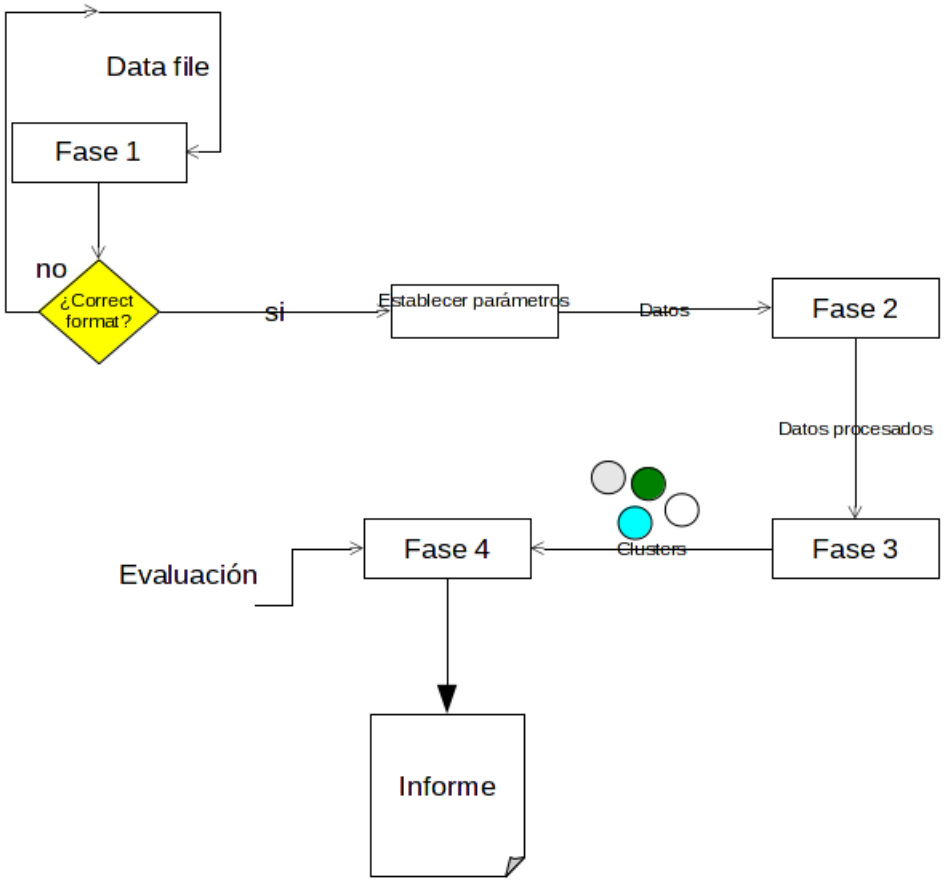


Figura 7.1: Dependencias del sistema

7.3. Implementación

La implementación está realizada en el lenguaje de programación *JAVA* haciendo uso de librerías para la visualización, **JFreeChart**, y para la generación automática de informes.

7.4. Carga de datos y configuración

Inicialmente se carga la configuración establecida por el usuario en el fichero **kmeans.conf**, es decir: path del fichero y su formato, tipo de inicialización para el *codebook*, número de clusters, distancia a utilizar, número de clústers deseados, elección manual o automática sobre la normalización y diversas opciones más, especificando datos sobre el fichero que se utilizará para las instancias.

7.5. Preproceso de datos

En el preproceso de datos se normalizará o no, dependiendo del parámetro indicado por el usuario. Si el parámetro es 0 no se normalizará, si es 1 se normaliza y si es 2 se hará uso del método experimental para decidir la idoneidad de la normalización.

7.6. Algoritmo K-means

En esta fase se implementa el algoritmo **K-means**.

1. En primer lugar inicia los *centroides* con el criterio establecido por el usuario, o la matriz de bits de pertenencias.
2. Recorre las instancias del conjunto y calcula la distancia a cada uno de los *codeword* actualizando la matriz de bits de pertenencia,el valor del bit es uno si es el centroide más cercano a la instancia.
3. Se calcula de nuevo el vector promedio para cada cluster.
4. Iterar los pasos dos y tres hasta converger.

7.6.1. Algoritmo en pseudocódigo

```
Let k be the number of clusters to partition the data set
Let X = x_1, x_2, ..., x_n be the data set to be analyzed
Let M = m_1, m_2, ..., m_k be the code-book associated to the clusters
Let dist(a, b) be the desired distance metric
Let B = B_11, B_12, ..., B_nk be the temporary pertenece bit matrix

Ensure: C = C_1, C_2, ..., C_k set of clusterized instances

Begin:

  //randomly initialize the first centroids
  for each m_j
    m_j = randomsample(X)
  end

  //assign dataset instances to each cluster generated by the centroids
  for each x_n
    B_nj = 1 if argmin dist(x_n, m_j) = m_j \foreach m_j else B_nj = 0
  end

  for each B_nj
    if B_nj == 1
      C_j.add(x_i)
    end
  end

  //iterate the algorithm generatin new centroids based on previously clusterized instances until
      there are no changes between iterations
  while changes in M do
    for each m_j
      m_jnew = calculatecentroid(C_j)
      if m_jnew == m_j
        changes = false
      else
        changes = true
      end
      m_j = m_jnew
    end

    for each x_n
      B_nj = 1 if argmindist(x_n, m_j) = m_j \foreach m_j else B_nj = 0
    end

    for each B_nj
      if B_nj == 1
        C_j.add(x_i)
      end
    end
  end

  return C = C_1, C_2, ..., C_k
end
```

7.7. Implementación: Formato de entrada de datos

Este particular es el que menos tiempo y esfuerzo nos ha llevado en la implementación, ya que el manejo de archivos se encuentra resuelto con clases disponibles en el API de java.

Gracias al alto nivel de configuración del sistema, que se pasará a explicar a continuación, éste es capaz de tratar ficheros de entrada de tres formatos diferentes: ARFF (utilizado por la librería de minería de datos Weka), TXT y CSV.

Tanto para los ficheros con extensión arff como txt se ha realizado una implementación propia, para los ficheros de entrada se ha decidido hacer uso de la librería OpenCSV disponible de manera libre en Internet. Pese a que podrían ser tratados como los ficheros txt de la implementación, se decide hacer uso de esta librería ya que ha sido el primer formato tratado, antes de llevar a cabo el sistema propio de carga de datos.

7.8. Implementación: Configuración del sistema

Cómo ha sido mencionado anteriormente en este documento, el sistema es altamente configurable, por lo que los argumentos de configuración son numerosos y es sabido, dada la experiencia propia, que no es eficiente tratar un número alto de argumentos de entrada a través de la línea de comandos.

La decisión ante este inconveniente ha sido crear un fichero de configuración a través del cual el usuario tiene la posibilidad de ajustar la ejecución a sus datos o incluso realizar diferentes ejecuciones con distintos parámetros en la búsqueda de una ejecución óptima.

Parámetros:

- file : donde indicaremos el path del fichero que contiene el conjunto de datos.
- k: donde indicaremos el número de clusters para la ejecución.
- iterations: si este es 0 la parada ejecución se decidirá por disimilitud de los codebook.
- difference: valor para ponderar la diferencia de las distancias entre las instancias y los centroides. Es decir el cambio de pertenencia.
- distance: el exponente de la distancia Minkowski.
- initialize: para inicializar con una matriz de pertenencia aleatoria(0) o con instancias del conjunto escogidas aleatoriamente como codewords(1).
- file_ extension: indica la extensión del archivo.
- data_ line_ start: permite al usuario indicar la línea en la que comienzan los datos. Este parámetro nos pareció interesante por dos motivos. El primero es que permite manejar archivos con cualquier información antes de comenzar a extraer los datos y el segundo es que se pueden obtener datos conjuntos de datos de diferentes tamaños extraídos de un mismo fichero.
- delimiter: para indicar el delimitador entre los distintos atributos.
- normalize: para dejar que el usuario decida si normalizar(1) o no(0).Por otra parte cabe la posibilidad de dejar que el sistema decida si normalizar o no(2).
- ratio_ max: para indicar la disimilitud entre codebooks(0.0 distintos, 1.0 iguales).

7.8.1. Análisis del conjunto de datos para decidir la conveniencia de la normalización

Debido a las dudas surgidas en torno a la normalización de los atributos y su conveniencia, se ha considerado adecuado para la tarea tratar de buscar algún método que fuese de alguna manera indicador de la utilidad de la normalización.

Inicialmente el planteamiento se basaba en utilizar la media de la desviación típica de los valores de cada atributo con el objeto de poder analizar los rangos de las diferencias entre valores de cada atributo. Sin embargo la media por separado no es siempre indicativa, ya que por ejemplo, si la media es alta pero la desviación es baja, en realidad, puede no haber mucha variación en los rangos. Esto ha llevado a hallar lo que se denomina

Coeficiente de Variación:

$$C_V = \frac{\sigma}{\bar{x}} \tag{7.1}$$

De esta forma se logra un indicador más preciso sobre el rango buscado ya que indica la proporción de la variabilidad de las desviaciones, en lugar de la mera cantidad de desviación.

La motivación de este análisis viene dada más por el interés de hallar una forma de conocer la utilidad que pueda tener la normalización en un conjunto de datos, ya que objetivamente, el beneficio principal que puede tener e s que si no afecta demasiado al resultado final, puede interesar más tener los atributos en su rango numérico inicial (p.ej.: es más visual ver un g ráfico con edades entre 0 y 100 que entre 0 y 1).

Por otra parte, se ha tratado de hallar una cifra del Coeficiente de Variación que esté comúnmente aceptada como baja, pero no se ha encontrado ninguna fuente fidedigna para ello. Por lo tanto, se ha decidido establecer una cifra pequeña (0,1) teniendo en cuenta que ante la posibilidad de que haya rangos muy variados, puede ser más conveniente normalizar.

Huelga decir que este método y sus resultados son experimentales, pese a tener cierta base empírica se carece de la certeza sobre su eficacia, siendo el objetivo principal investigar respecto a la normalización en lugar de simplemente aplicarla por estar aceptada como conveniente.

7.9. Implementación: Evaluación

El objetivo principal de éste módulo consiste en obtener un algoritmo capaz de evaluar los resultados de aplicar clasificación no supervisada (Clustering) en un conjunto de datos que representa los clientes de la empresa.

Las técnicas de clasificación no supervisada consiten en aquellas en las que no existe conocimiento a priori, donde se agrupa las instancias, a los clientes en este caso, sin atributos dependientes pre-especificados. Los algoritmos de “clustering” son un método común de aprendizaje no supervisado.

Para el proyecto el algoritmo de clustering utilizado es el conocido como K-means o K-medias, pero el módulo de evaluación es independiente del algoritmo aplicado.

En la figura 7.1 se presenta el diseño del flujo para el módulo completo de segmentación de usuarios, la fase cuatro representa la fase de evaluación.

Actualmente no existe una metodología de referencia para esta tarea, por lo tanto se trata de un trabajo experimental, el cual se basa en resultados obtenidos empiricamente. Como se puede ver en la figura el diseño inicial para evaluar los clusters consistía en comparar ejecuciones anteriores con la que se desea evaluar. Lo que parecía una buena aproximación, no tardo mucho es desecharse porque si queremos evaluar un sistema, y no conocemos si el funcionamiento es correcto o no, de poco sirve evaluar dicho sistema contra si mismo.

Este pensamiento me empujo a buscar como cuantificar la cohesión de los elementos de un mismo grupo. En general encontré diferentes metricas de validación de clustering agrupadas en dos categorías:

Métricas de validación internas:

- Cohesión y separación
- Coeficiente Silhoutte
- SSW
- SSB

Métricas de validación externas:

- Precision
- Recall
- F-measure
- Entropía

- Pureza
- Mutual Información

En la actualidad se ha escogido una métrica de validación interna para evaluar los clusters, coeficiente Silhouette, que relaciona las métricas de cohesión y separación y que se explica en el siguiente apartado.

7.9.1. Evaluación: Silhouette Coefficient

Es una combinación de las medidas de separación y cohesión:

Figura 7.2: Medidas de cohesión y separación

El coeficiente Silhouette puede ser calculado para puntos independientes y para clusters. Para un punto individual,a=la distancia promedio de i a los puntos del mismo cluster;b=la distancia promedio de i a los puntos de los otros clusters 7.3.

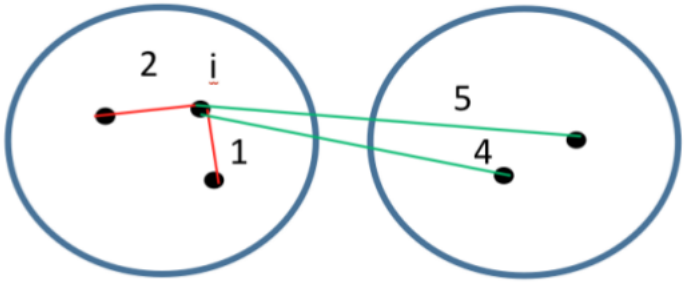


Figura 7.3: Coheficiente silhouette para un punto

Fuente e imágenes extraídas de MOA[pags 3,4]
Para calcular el coeficiente del conjunto total de los clusters inicialmente se calcula el coeficiente para cada miembro de un mismo cluster y se calcula la media de todos los coeficientes hallados,se calcula ésto para cada cluster y se devuelve la media de todos los coeficientes hallados para todos los clusters.

7.9.2. Visualización

Con el fin de analizar los resultados de una forma visual, se presenta la representación gráfica de la matriz de pertenencias,figura 7.4.
Para este fin hacemos uso de la librería **JFreeChart** para realizar los gráficos y de **iTex** para generar el informe en PDF.
Al finalizar el sistema muestra una gráfica de la matriz de pertenencias, permitiendo analizar de una forma visual a que cluster pertenece cada instancia:

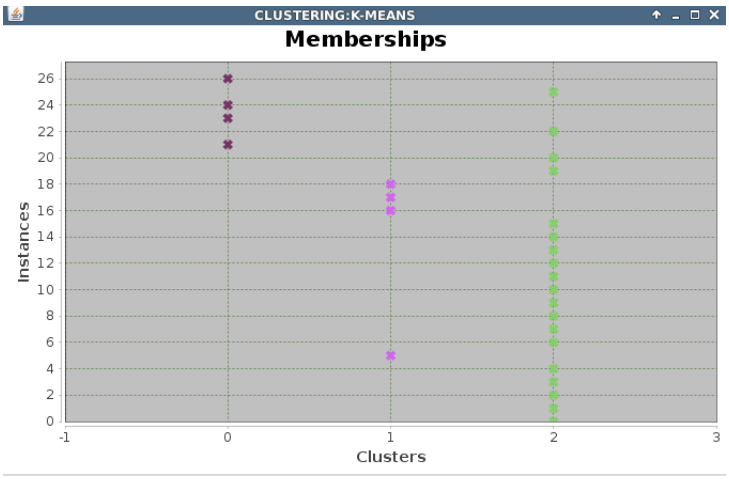


Figura 7.4: Pertenencias

7.10. Diseño del banco de pruebas

Los informes de resultados se engloban en un marco teórico experimenta ya que actualmente nos encontramos en la fase de recopilación de los atributos que nos permita definir mejor a cada cliente.

File	k	iterations	difference	distance	initialize	normalize	disimilitud
bank-data.csv	1	0	0.0	2	0	0	0.5
bank-data.csv	2	0	0.0	2	1	1	0.8
bank-data.csv	3	10	0.3	1	0	2	x
bank-data.csv	25	100	0	3.5	1	0	x
bank-data.csv	25	0	0	7.5	1	2	1.0
bank-data.csv	25	0	0	7.5	1	0	0.6
colon.arff	1	0	0.0	2	0	0	0.5
colon.arff	2	0	0.0	2	1	1	0.8
colon.arff	3	10	0.3	1	0	2	x
colon.arff	10	30	0	3.5	1	0	x
colon.arff	10	0	0	7.5	1	2	1.0
colon.arff	10	0	0	7.5	1	0	0.6
ClusterData.atributos.txt	1	0	0.0	2	0	0	0.5
ClusterData.atributos.txt	2	0	0.0	2	1	1	0.8
ClusterData.atributos.txt	3	10	0.3	1	0	2	x
ClusterData.atributos.txt	30	40	0	3.5	1	0	x
ClusterData.atributos.txt	30	0	0	7.5	1	0	1.0
ClusterData.atributos.txt	30	0	0	7.5	1	2	0.6

Tabla 7.1: Banco de pruebas experimental

Tras resolver los problemas aparecidos en la implementación, el sistema pasa el banco de pruebas completo con éxito.

7.11. Análisis de resultados

Analizar los resultados no es tarea fácil, no existe un método de evaluación estandarizado para el clustering, por lo que nos basamos únicamente en el coeficiente ya que nos acerca a una medida de lo bien que el algoritmo agrupa las instancias.

Dicho esto y a la vista de los resultados aportados como anexo, se puede ver que el algoritmo es bastante eficiente. Podemos observar que para agrupar en un único cluster, la evaluación siempre nos dará el mismo resultado 1.0, que es el mejor que podemos obtener.

Si nos centramos la atención en los conjuntos de datos de los que disponemos la clase, con distancia euclidea y con disimilitud 1.0 aunque no obtenemos más de 0.3 de coeficiente, podemos analizar los datos con respecto a la ejecución y ver que el número de instancias de cada clase es similar al número de instancias en cada cluster. Esto nos índica que quizás merezca analizar los datos a partir de un índice no muy alto.

7.11.1. Modificando inicializaciones

Debido a que las inicializaciones posibles son matriz de pertenencia aleatoria o codebook inicial escogiendo instancias del conjunto de datos, los resultados no varían demasiado con el cambio de este método, pero el resultado si depende de como se inicializa el codebook.

7.11.2. Criterios de convergencia

En las subsiguientes secciones se pasa a realizar el análisis de los resultados en función de los distintos criterios de convergencia.

7.11.2.1. Número fijo de iteraciones

Por lo general el aumento de iteraciones es proporcional al resultado, aunque llegado a un número de iteraciones el resultado no varía. Para los datos manejados generalmente a partir de la iteración diez el coeficiente silhouette no varía.

7.11.2.2. Disimilitud entre *codebooks*

Este punto se podría analizar igual que el punto anterior, dado que la disimilitud escogida lo que permite es un mayor número de iteraciones. Es decir que aunque pongamos que la disimilitud entre codebooks sea 0.2 y no 1.0 que es la máxima similitud, es decir que no para hasta que sean iguales, cuando alcanza el número de iteraciones a partir del cual el coeficiente no varía, se alcanza el mejor resultado posible.

7.11.3. Distintas métricas

Tras hacer diferentes ejecuciones, tanto con distancias mahattan y euclidea como con diversos exponentes para la distancia Minkowski, se observa que los mejores resultados se obtienen con las dos primeras, para los tipos de problemas de los que disponemos. Esto podría justificar porque algunas librerías de minería de datos ya existentes como Weka no implementen Minkowski para este tipo de algoritmo ya que a partir de un exponente de tres, los resultados del coeficiente disminuyen casi hasta cero.

Capítulo 8

Validación y pruebas

La fase de pruebas es una parte esencial en todo proyecto de sistemas informáticos, y el Business Intelligence no es una excepción a esta regla. Debido a la alta complejidad que caracteriza estos proyectos, la ausencia de un plan de pruebas progresivo tras cada etapa que modifique alguno de los componentes clave del proyecto puede suponer un fracaso rotundo del mismo. Esto se debe a que un pequeño cambio puede suponer inumerables horas de trabajo para adaptar el proyecto en su conjunto, debido a la propagación que tienen los cambios a lo largo de los diferentes elementos del sistema.

8.1. Data Warehouse y procedimientos ETL

Es esencial tener un almacén de datos consistente en un sistema centrado en explotar los datos. Debido a la naturaleza cambiante del proyecto a medida que se va avanzando en el desarrollo, ha sido necesario realizar pruebas de integridad en cada ocasión que se ha modificado el Data Warehouse de alguna forma susceptible de crear incogruencias.

Con este motivo se ha recurrido a diversas estrategias de evaluación. La primera y la más utilizada en las fases iniciales de la implementación del proceso ETL ha sido la contrastación de los datos cargados en el Data Warehouse con los mismos datos consultados en el sistema origen. Gracias a que la información integrada en el almacén de datos es fácilmente contrastable mediante varias funcionalidades de este software, esta metodología ha supuesto buena parte de la comprobación de la consistencia.

Para verificaciónes de datos más complejas se ha recurrido a utilizar sentencias SQL contra la base de datos origen. Este método se basa en el simple hecho de que una sentencia que solicite un determinado resultado en este entorno y en el Data Warehouse, debe en todo momento devolver los mismos datos. De no ser así, se tendrán localizados los posibles puntos en los que se ha generado la inconsistencia y será más sencillo subsanarla.

En siguientes iteraciones a partir del primer diseño y carga del Data Warehouse se ha utilizado la comparación con este último de referencia, teniendo la certeza de su consistencia, las siguientes versiones deben arrojar la misma información ante las mismas solicitudes de datos.

Para esta fase también se ha contado como apoyo con la ayuda de un consultor experto en el sistema origen de los datos, de forma que la validación de la consistencia se ha podido realizar con menos inversión temporal en el estudio de dicho sistema.

En los aspectos en los que ha sido posible y ya que la fuente de origen en este caso es el ERP de la empresa, también se ha contrastado parte de la información con aquella que se obtiene con alguna de la funcionalidad extra de las que dispone el ERP.

8.2. Dashboards

Para llevar a cabo las pruebas de los cuadros de mando, se ha establecido un pre-requisito, que será a su vez trasladado al cliente: hacer uso de los navegadores Mozilla Firefox o Google Chrome. Esto se debe a que gran parte de la funcionalidad y apariencia se encuentra implementada en Javascript, CSS3 y elementos HTML5, tecnologías relativamente novedosas, que pueden ocasionar problemas con el navegador Internet Explorer 9 en Windows XP y Windows Vista. Se han realizado pruebas y este navegador no es compatible al cien por cien con el sistema, y dado que Windows XP el sistema operativo que se utilizará en alguno de los PCs del cliente final, se ha decidido no garantizar el correcto funcionamiento en ninguna versión de Internet Explorer.

Las pruebas en los cuadros de mando han sido realizadas una vez creada su estructura, componentes y orígenes de datos necesarios. Estos últimos, en el caso de las sentencias SQL, han sido contrastados mediante la herramienta PGAdmin 3 de PostgreSQL, validando la correcta sintaxis y resultados consistentes. En el caso de los orígenes de datos de MDX, se ha recurrido al plug-in Saiku Analytics, herramienta que además de permitir probar sentencias MDX, facilita la creación de las mismas con una sencilla interfaz:

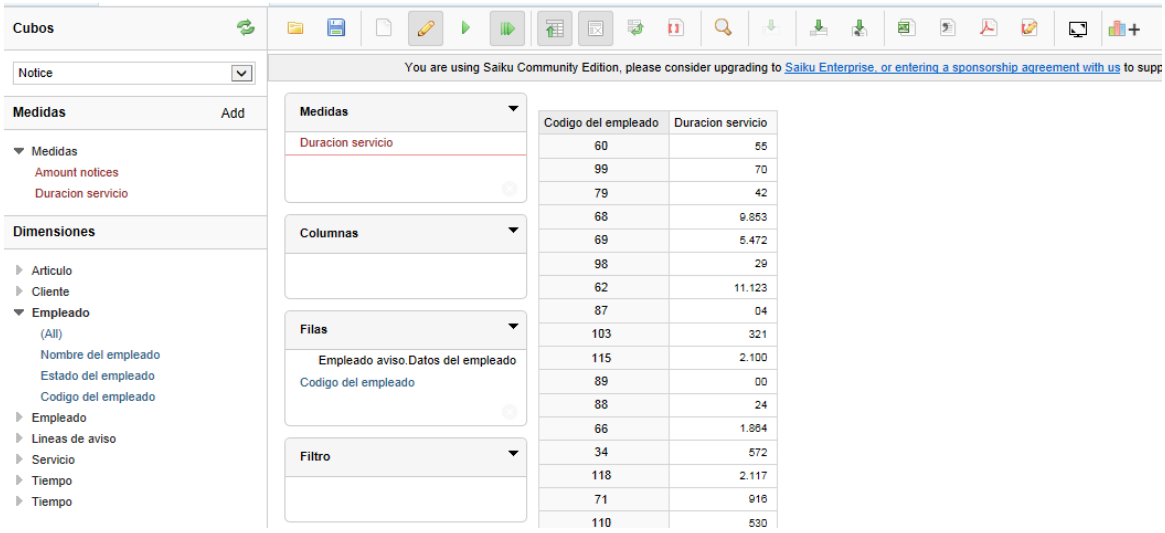


Figura 8.1: Interfaz principal de Saiku Analytics

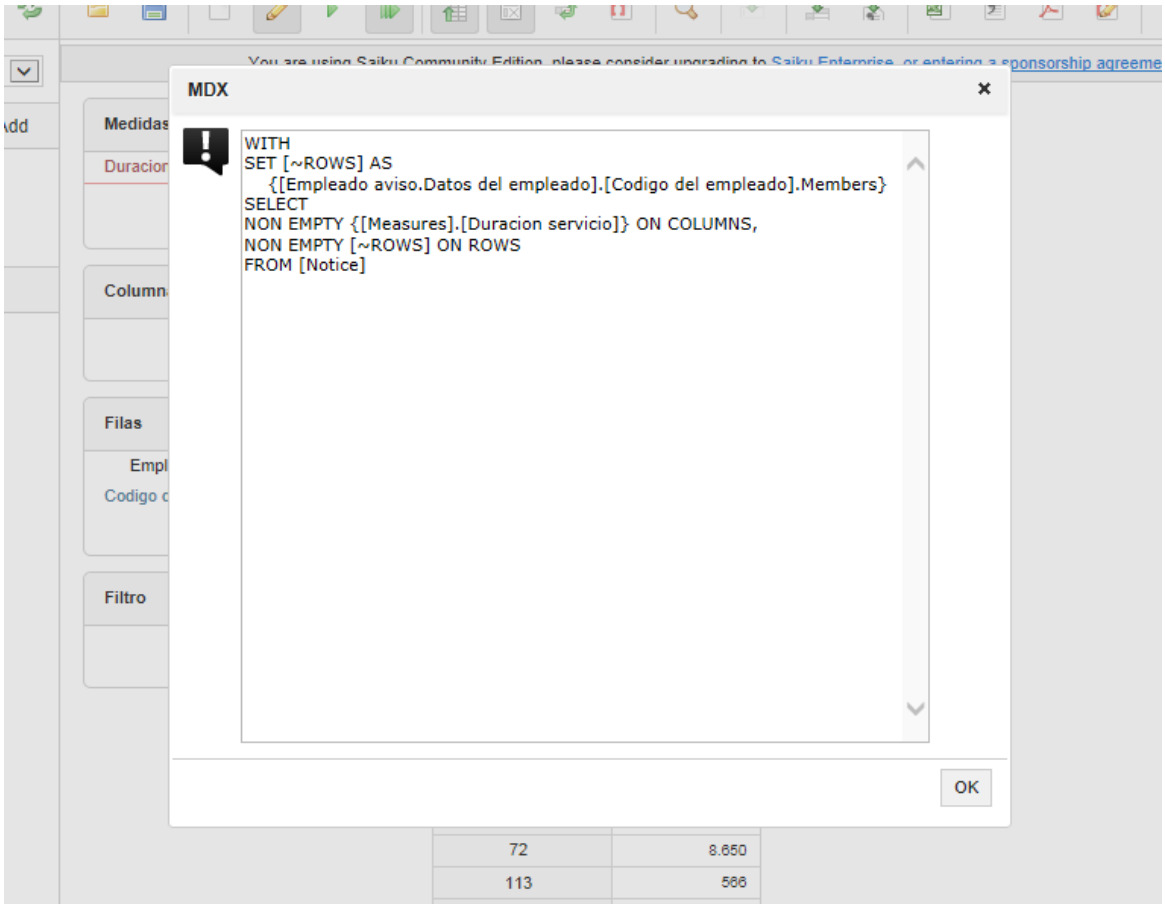


Figura 8.2: Opción de ver MDX en Saiku Analytics

Durante la fase de pruebas, se han generado unas tablas recogiendo las principales comprobaciones de funcionalidad realizadas, así como los problemas surgidos hasta lograr obtener el resultado esperado: **Ver figuras A.2, A.3, A.4, A.5, A.6.**

Las verificaciones más sencillas como por ejemplo comprobar la correcta colocación de un elemento en la interfaz, tamaños de componentes y similares no han sido documentadas debido a que por su número y sencillez, resultaría en demasiado tiempo consumido con pocos o nulos beneficios.

Capítulo 9

Conclusiones y líneas futuras

El presente capítulo trata de echar la vista atrás con una mirada crítica para puntualizar los aspectos susceptibles de mejorar y los puntos fuertes que pueden hacer destacar el proyecto frente a otros.

9.1. Planificación inicial frente a final

La inexperiencia en la estimación del coste temporal de un proyecto de software, inluso más si cabe de la naturaleza del presente trabajo, debido que se ha estudiado de una manera superficial a lo largo de la carrera el campo en el que queda enmarcado, el Business Intelligence. Esto fué lo que nos empujó a considerar el fallo en la estimación temporal como un riesgo con una probabilidad alta, y así ha sido como se puede ver en la tabla 9.1 que enfrenta la duración estimada con la duración real del proyecto:

Tarea	Duración estimada/horas	Duración real/horas
0-Gesión	5	12
1-Planificación	5	16
2-I+D+i	65	84
3.1-Captura requisitos	33	32
3.2-Análisis	16	20
3.3-Diseño	46	45
3.4-Implementación	140	204
3.5-Pruebas	36	31
4.1-DOP	40	36
4.2-Memoria	85	87
5.1-Formación	12	0
6.1-Despliege del servidor	6	8
X-Modulo de minería de datos	58	50
Total	547	625

Figura 9.1: Planificación estimada frente a real

La desviación total es de menos de 80 horas. Si analizamos en detalle el gráfico de la figura 9.2 nos vamos a encontrar con que la máxima desviación ha sido en las horas de implementación con 64 horas reales por encima de las estimadas.

El resto de las tareas la duración real no se aleja demasiado de la estimada, excepto quizás las tareas de I+D+i que tienen una diferencia de 19 horas.

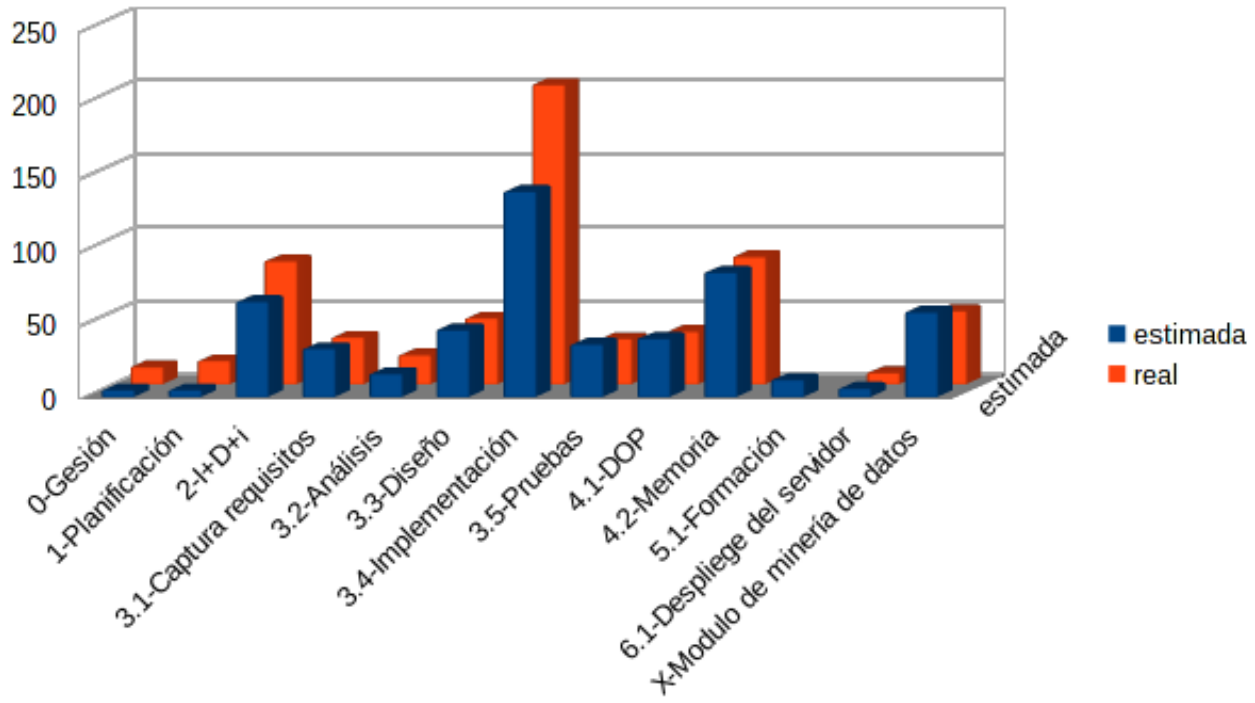


Figura 9.2: Comparativa estimada frente a real

9.2. Conclusiones

Después de finalizar el trabajo, al menos las iteraciones que serán entregadas como trabajo de fin de grado, llega el momento de ser críticos, hacer repaso del trabajo realizado y en lugar de concluir, incluir el proyecto en un proceso de mejora continua que nos permita alcanzar la excelencia con nuestro producto. Lo cierto es que el propio proyecto en sí mismo representa un proceso de mejora continua en el que el equipo del proyecto y los clientes convergen en un mismo objetivo, alcanzar un producto que satisfaga las necesidades del cliente, resolviendo las preguntas surgidas sobre la problemática de la empresa antes de concluir el proyecto y proporcionando una herramienta que permita conocer en todo momento la situación pasada y presente de la empresa, y proporcione un sistema de apoyo a la decisión en el que apoyarse para la toma de decisiones estructuradas e informadas.

Volviendo la vista atrás se puede entender como alcanzado el objetivo principal ya que radica en sentar una base de un sistema de Business Intelligence adaptable a diferentes casos de negocio, y se resuelve como alcanzado por que actualmente son dos las empresas que explotan este sistema. Por un lado está la empresa dedicada a las asistencias técnicas a domicilio, por el otro la empresa proveedora dedicada a la actividad de consultoría informática y que también a añadido valor a su negocio a través del presente *SaaS*. Otro de los objetivos principales consistía en desarrollar el producto que la empresa proveedora buscaba desarrollar, y así ha sido. No sólo se ha alcanzado si no que existe otro desarrollo de las mismas características técnicas, pero con diferencias funcionales, por lo que además de un nuevo producto se ha ampliado la cuota de mercado del proveedor con un nuevo proyecto. No ha sido posible "enganchar" al nuevo cliente a este SaaS porque la actividad económica de éste se desarrolla e un ámbito industrial productivo por lo que se a optado por desarrollar un nuevo producto que abarque esta problemática tan distinta a la del proyecto desarrollado como trabajo de fin de grado.

Bajo la perspectiva del cliente todavía es pronto para hablar de grandes resultados, pero nos encontramos ante una mejoría importante sobre todo en cuanto a la accesibilidad de la información que los empleados del cliente necesitaban. Ahora tienen más autonomía para auto-gestionar su trabajo, facilitando a su vez la labor del departamento de administración.

Huelga decir que se han adquirido nuevas competencias en la realización y gestión del proyecto. Por mi parte he logrado desarrollarme como profesional en el mundo del Business Intelligence, adquiriendo novedosos conocimientos en la materia por el camino y también aumentar mi capacidad de investigar, desarrollar e implementar dado el alto grado de I+D+i que ha implicado el trabajar en este proyecto. En resumen me ha aportado los conocimientos teóricos y técnicos suficientes para emprender el camino hacia la excelencia en un campo tan emergente.

Es importante hablar de los continuos cambios en las especificaciones que mayormente afectaban a la visualización y disposición de los datos en pantalla. En el análisis de riesgos fueron estimados poco probables, pero también de muy bajo o ningún impacto, predicción que también se ha ajustado a la realidad ya que se han acometido sin mayor repercusión para el resto del proyecto.

Si bien es cierto que una vez adquirida más experiencia será posible realizar unas estimaciones más precisas y desarrollar funcionalidades con una mayor flexibilidad analítica, la experiencia me dice que cada proyecto será distinto, que la problemática de un negocio puede ser similar a la de otro con la misma actividad, pero las necesidades, las preguntas a las que necesité responder cada empresa y las fuentes de datos origen de las que disponen, con una muy alta serán dispares, y por ello nos encontraremos ante nuevos desafíos que resolver y que quizás no los hallamos enfrentado hasta ese momento. Ya sea de nuevas estructuras lógicas que modelar, arquitecturas de sistemas con políticas de seguridad complejas... etcétera

9.3. Líneas futuras

Las líneas futuras que se pretenden seguir desarrollando son varias. Actualmente estamos trabajando en definir a los clientes con el mayor número de atributos que nos permitan hacerlo con la mayor precisión posible para poder automatizar y aplicar la segmentación de usuarios desarrollada para incluir el resultado en el análisis, en forma de gráfica resumen en un dashboard e incluso en la tabla de detalle de los clientes, en un estudio posterior.

Mas hallá de la segmentación de clientes, existen análisis de datos más potentes para conseguir inteligencia de negocio. Entre estos destacan los modelos de propensión de compra o de fuga. Son modelos que estiman la probabilidad de que esta conducta ocurra para cada uno de nuestro clientes, y permite generar modelos para tomar decisiones en tiempo real.

La línea a seguir está enfocada hacia ese cámino, conocer mejor las conductas y necesidades de los clientes para tomar decisiones que ayuden a las empresas a prosperar y a los clientes a obtener lo que quieren.

Existen otros frentes que quedan abiertos. Por ejemplo se les ha facilitado un campo en el ERP que permitirá registrar cuando un aviso es facturable y cuando no lo es. Así podremos, no solo identificar los avisos cubiertos por garantías si no todos aquellos que no generan ingreso.

Por otro lado se ha dispuesto de dos campos de fecha y tiempo para que los técnicos puedan registrar la hora de inicio y finalización de la asistencia. La idea es generar un histórico que nos permita obtener un volumen de datos suficiente para entrenar un modelo predictivo para estimar el tiempo de asistencia y así acercarse un poco más hacia una gestión excelente y disminuir los costes.

Bibliografía

[1] Experiencia personal de la herramienta.

[2] *Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration.* Wiley, 2010.

[3] *Pentaho Data Integration 4 Cookbook.* Packt Publishing, 2011.

[4] *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition.* Wiley, 2013.

[5] Segundas jornadas de software de gestión empresarial. http://www.qlik.com/es/explore/experience/free-download, 2014.

[6] Daniel Castro Ann Cavoukian. Deidentificación en la minería de datos. http://www2.itif.org/2014-big-data-deidentification.pdf.

[7] Pentaho Corporation. Manual on-line pdi. http://wiki.pentaho.com/display/EAIes/Manual+del+Usuario+de+Spoon.

[8] Universidad de Cantabria. Teoría sobre segmentación de mercado. http://ocw.unican.es/ciencias-sociales-y-juridicas/direccion-comercial/Tema3_Segmentacion.pdf.

[9] Grupo Euclides. Qlikview. http://www.grupoeuclides.com/es/soluciones/reporting-y-analisis-de-negocio/qlikview.

[10] Aníbal Goicochea. Conectividad de los nuevos componentes de la plataforma sap bo bi 4.0. http://anibalgoicochea.com/2012/09/15/conectividad-de-los-nuevos-componentes-de-la-plataforma-sap-bo-bi-4-0/.

[11] Brent Smith y Jeremy York Greg Linden. Artículo en torno a sistemas de recomendación y amazon. http://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf.

[12] Kimball Group. Teoría sobre tablas puente. http://www.kimballgroup.com/2014/05/design-tip-166-potential-bridge-table-detours/.

[13] Meteorite.bi. Saiku analytics. http://www.meteorite.bi/products/saiku.

[14] U.S. Department of Health Human Services. Deidentificación en la medicina. http://www.hhs.gov/ocr/privacy/hipaa/understanding/coveredentities/De-identification/deidentificationpanels.html.

[15] pentaho. Pentaho. http://www.pentaho.com/.

[16] Qlik. Business discovery. http://www.qlik.com/es, 2014.

[17] Qlikview. Servicvios de formación qlikview. http://www.qlik.com/es/services/training.

[18] SAP. Manual del usuario de businessobjects xi. http://help.sap.com/businessobject/product_guides/boexir31SP2/es/xi31_sp2_bip_sap_user_es.pdf.

[19] SAP. Planes de soporte. http://www.sap.com/services-support/support/plans.html.

[20] SAP. Sap training and education courses for your sap solution. http://www.sap.com/training-education/overview/solution.html.

[21] SAP. Business intelligence solutions. http://www.sap.com/pc/analytics/business-intelligence/software/overview/index.html, 2014.

[22] Vizuvi. Distribución qlikview. http://www.vizubi.com/es/blog-es/distribucion-qlikview/.

Anexos

Anexos A

Data Warehouse

A.1. Diseño Data Warehouse

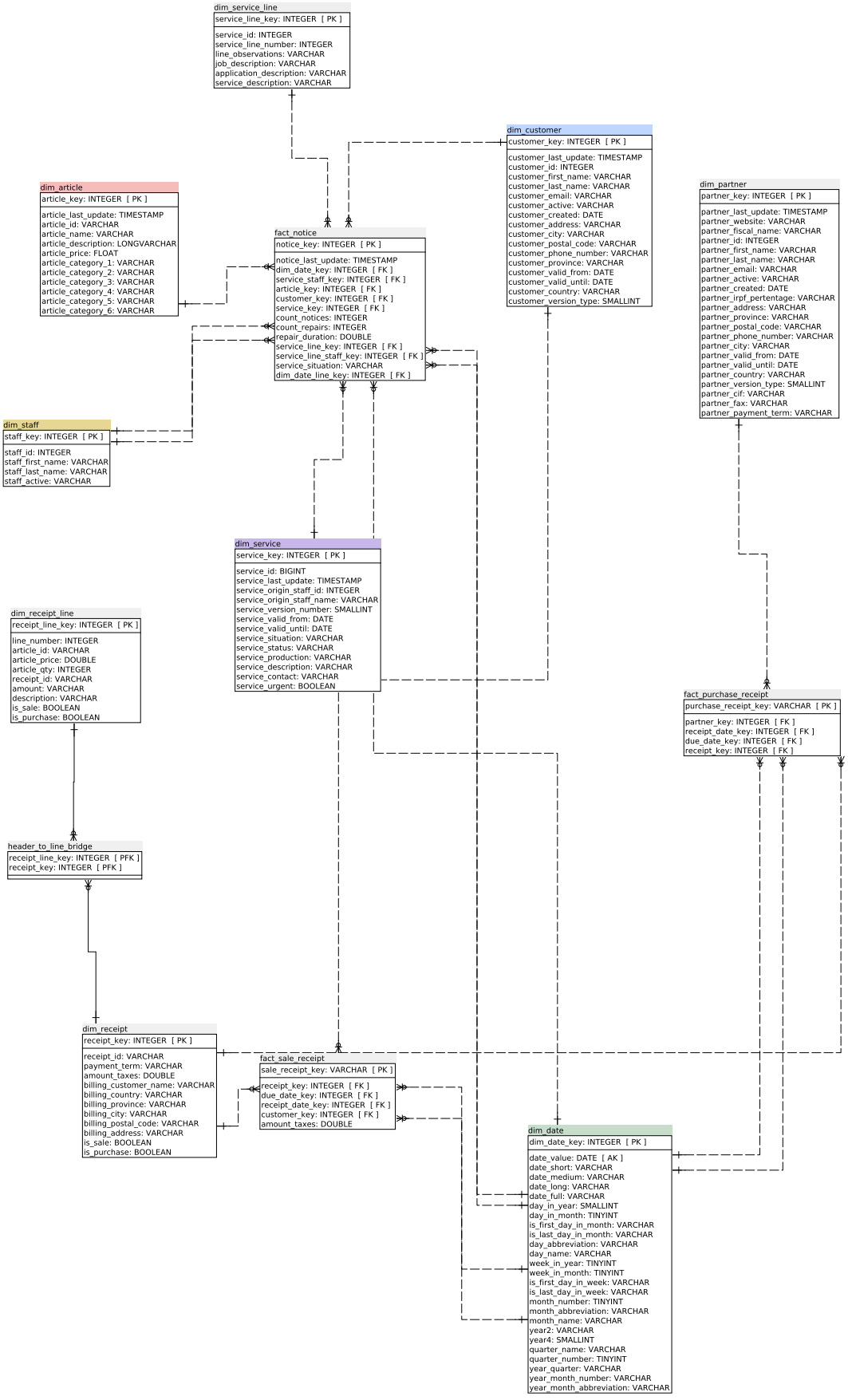


Figura A.1: Data Warehouse: diseño

A.2. Validación y pruebas de Dashboards

Dashboard:		OTs General	
Acción	**Prueba**	**Resultado esperado**	**Resultado obtenido**
Utilizar filtro de medida temporal	1.a	Se modifica el filtro de periodo	Resultado esperado
Utilizar filtro de periodo	2.a	Se aplica el filtro a todos los componentes	Resultado esperado
Hacer click en tabla resumen por trabajador	3.a	Redirección a CM OTs por empleado	Se abre la tabla y no se muestra nada
	3.b		Resultado esperado
Hacer click en gráfico por estado	4.a	Filtrar tabla por dicho estado, mostrar gráfico temporal por estado	Desaparece el gráfico por estado
	4.b		Resultado esperado
Hacer click en gráfico temporal por estado	5.a	Eliminar filtro de la tabla estado, mostrar gráfico estado	Resultado esperado
Hacer click en gráfico por producción	6.a	Filtrar tabla por dicha producción, mostrar gráfico temporal por producción	Resultado esperado
Hacer click en gráfico temporal por producción	7.a	Eliminar filtro de la tabla producción, mostrar gráfico producción	Resultado esperado

Figura A.2: OTs General

Dashboard:		OTs por Empleado	
Acción	**Prueba**	**Resultado esperado**	**Resultado obtenido**
Hacer click en una OT de la tabla pendientes	1.a	Se abre una sub-tabla con las líneas	Mensaje: Error processing component
	1.b		Resultado esperado
Hacer click en gráfico temporal	2.a	Se abre un pop-up con las líneas de ese día	Se abre pop-up sin contenido
	2.b		Se abre pop-up con mensaje de error
	2.c		Resultado esperado
Utilizar filtro de fechas	3.a	Se filtran todos los componentes salvo la tabla	El gráfico temporal no está filtrado
	3.b		Deja de funcionar el click en la tabla
	3.c		Resultado esperado

Figura A.3: OTs por Empleado

Dashboard:		Ventas General	
Acción	**Prueba**	**Resultado esperado**	**Resultado obtenido**
Hacer click en un cliente en la tabla resumen	1.a	Se abre un gráfico con análisis temporal de dicho cliente	Fallo en la tabla: Error processing component
	1.b		Se abre, aparece gráfico pero la query es incorrecta
	1.c		Resultado esperado
Utilizar filtro de fechas	2.a	Se filtran todos los componentes salvo la tabla y el gráfico de años	Falla la tabla, el gráfico por años se filtra incorrectamente
	2.b		Resultado esperado

Figura A.4: Ventas General

Dashboard:		Ventas vs. OTs	
Acción	**Prueba**	**Resultado esperado**	**Resultado obtenido**
Hacer click en una localidad, gráfico tiempos	1.a	Se abre un pop-up con las OTs en esa fecha	Desaparece el gráfico
	1.b		Se abre el pop-up pero se muestran líneas
	1.c		Resultado esperado
Hacer click en una localidad, gráfico ingresos	2.a	Se abre un pop-up con las OTs en esa fecha	Resultado esperado
Utilizar filtro de fechas	3.a	Se filtran los gráficos	Se filtran, pero los pop-up fallan al hacer click
	3.b		Resultado esperado

Figura A.5: Ventas vs. OTs

Dashboard:		OTs por Articulos	
Acción	**Prueba**	**Resultado esperado**	**Resultado obtenido**
Utilizar filtro de fechas	1.a	Se filtran todos los componentes	Error en gráfico temporal: Error processing component
	1.b		Resultado esperado
Utilizar filtro de categorías	2.a	Se modifica el filtro de valor categoría. Se filtran los componentes	Error en el filtro de valor
	2.b		No se aplica el filtro de valor
	2.c		Resultado esperado
Utilizar filtro de valor de categoría	3.a	Se filtran todos los componentes	Resultado esperado

Figura A.6: Articulos

Anexos B

Procedimientos ETL

B.1. Diseño de los procedimientos ETL

Objetivo	Fuente	Observaciones
fact_notice.notice_key	autoincremental	
fact_notice.dim_date_key	fecha_cot	comparar en dim_date con data_value
fact_notice.service_staff_key	destino_cot	comparar en dim_staff con staff_first_name
fact_notice.service_line_staff_key	destino_lot	comparar en dim_staff con staff_first_name
fact_notice.dim_date_line_key	fecha_lot	comparar en dim_date con data_value
fact_notice.customer_key	cliente_id	Tabla fa_clientes, comparar en dim_customer con customer_first_name
fact_notice.repair_duration	tiempo_lot	tiempo de línea, no de servicio completo
fact_notice.service_situation	situaot_id	desc_sot en tabla ot_situaot

Figura B.1: Tabla de hechos de avisos

Tabla origen:	*tabla fa_cabfac en ENBOR*	
Objetivo	**Fuente**	**Observaciones**
fact_sale_receipt.receipt_key	autoincremental	
fact_sale_receipt.customer_key	Buscar customer_id	Busqueda de customer_id en dim_customer
fact_sale_receipt.sale_receipt_key	vfactura_id	Busqueda de receipt_id en dim_receipt además si en DWH issale==True
fact_sale_receipt.due_date_key	echavencimiento_c	Busqueda de fecha en dim_date comparar data_value
fact_sale_receipt.receipt_date_key	fechafact_cff	Busqueda de fecha en dim_date comparar data_value

Figura B.2: Tabla de hechos de facturas de ventas

Tabla origen:	*tabla gc_cabfac en ENBOR*	
Objetivo	**Fuente**	**Observaciones**
fact_purchase_receipt.receipt_date_key	fechafact_cfc	comparar en dim_date con data_value
fact_purchase_receipt.partner_key	proveedor_id	comparar en dim_partner con partner_id=proveedor_id
fact_purchase_receipt.receipt_key	factura_id	sí dwh ispurchase==True
fact_purchase_receipt.purchase_receipt_key	autoincremental	
fact_purchase_receipt.due_date_key	fechavto_cfc	comparar en dim_date con data_value

Figura B.3: Tabla de hechos de facturas de compra

Tabla origen:	*tabla ot_cabot en ENBOR*	
Objetivo	**Fuente**	**Observaciones**
dim_service.service_key	autoincremental	
dim_service.service_id	fecha_cot	comparar en dim_date con data_value
dim_service.origin_staff_id	origen_id	comparar en dim_staff con staff_first_name
dim_service.origin_staff_name	origen_id	comparar en dim_staff con staff_first_name
dim_service.service_situation	situaot_id	Tabla ot_situaot, campo desc_sot
dim_service.service_status	servicio_id	Tabla ot_servicios, campo desc_ser
dim_service.service_production	trabajo_id	Tabla ot_trabajos, campo desc_tra
dim_service.service_contact	contacto_cot	
dim_service.service_urgent	urgente_cot	

Figura B.4: Dimensión de orden de trabajo

Tabla origen:	*tabla ot_linot en ENBOR*	
Objetivo	**Fuente**	**Observaciones**
dim_service_line.service_line_key	autoincremental	
dim_service_line.service_id	ot_id	
dim_service_line.service_line_number	linea_lot	
dim_service_line.job_description	desctrabajo_cot	comparar en dim_staff con staff_first_name
dim_service_line.service_description	descservicio_cot	Tabla ot_situaot, campo desc_sot
dim_service_line.application_description	descaplicacion_cot	Tabla ot_servicios, campo desc_ser
dim_service_line.line_observations	observaciones_lot	Tabla ot_trabajos, campo desc_tra

Figura B.5: Dimensión de línea de orden de trabajo

Tabla origen:	*tabla fn_personal en ENBOR*	
Objetivo	**Fuente**	**Observaciones**
dim_customer.customer_key	autoincremental	
dim_customer.customer_first_name	nombre_cli	
dim_customer.customer_email	email_cli	
dim_customer.customer_active	tipocliente_cli	
dim_customer.customer_country	pais_id	tabla loc_pais, campo nombre_pa
dim_customer.customer_postal_code	poblacion_id	el id es el código postal
dim_customer.customer_created	fechalta_cli	
dim_customer.customer_province	provincia_cli	
dim_customer.customer_id	cliente_id	
dim_customer.customer_phone_number	telefono_cli	
dim_customer.customer_address	direccion_cli	
dim_customer.customer_city	poblacion_cli	

Figura B.6: Dimensión de cliente

Tabla origen:	*tabla gc_proveedores en ENBOR*	
Objetivo	**Fuente**	**Observaciones**
dim_partner.partner_city	poblacion_pro	
dim_partner.partner_phone_number	telefono_pro	
dim_partner.partner_website	www_pro	
dim_partner.partner_province	provincia_pro	
dim_partner.partner_fiscal_name	nombrefiscal_pro	
dim_partner.partner_id	proveedor_id	
dim_partner.partner_cif	cif_pro	
dim_partner.partner_key	autoincremental	
dim_partner.partner_first_name	nombre_pro	
dim_partner.partner_payment_term	pago_id	
dim_partner.partner_postal_code	poblacion_id	el id es el código postal
dim_partner.partner_created	fechalta_pro	es un array, extraer primer valor
dim_partner.partner_fax	fax_pro	
dim_partner.partner_country	pais_id	tabla loc_pais, campo nombre_pa
dim_partner.partner_email	email_pro	
dim_partner.partner_irpf_pertentage	porcenirpf_pro	
dim_partner.partner_address	direccion_pro	

Figura B.7: Dimensión de proveedor

Tabla origen:	*tabla al_articulos en ENBOR*	
Objetivo	**Fuente**	**Observaciones**
dim_article.article_name	desc_art	
dim_article.article_key	autoincremental	
dim_article.article_category_2	aux2_id	tabla al_aux2 campo desc_aux
dim_article.article_id	articulo_id	
dim_article.article_category_5	aux5_id	tabla al_aux5 campo desc_aux
dim_article.article_category_3	aux3_id	tabla al_aux3 campo desc_aux
dim_article.article_category_1	aux1_id	tabla al_aux1 campo desc_aux
dim_article.article_category_6	aux6_id	tabla al_aux6 campo desc_aux
dim_article.article_category_4	aux4_id	tabla al_aux4 campo desc_aux
dim_article.article_price	stdcos_art	

Figura B.8: Dimensión de artículo

Tabla origen:		
Objetivo	**Fuente**	**Observaciones**
dim_staff.staff_first_name	nombre_per	
dim_staff.staff_last_name	apellido1_per	contiene ambos apellidos si los hay
dim_staff.staff_active	activo	
dim_staff.staff_key	autoincremental	
dim_staff.staff_id	personal_id	se guarda como string en ENBOR

Figura B.9: Dimensión de empleado

Tabla origen:	*fa_cabfac y gc_cabfac*	
Objetivo	**Fuente**	**Observaciones**
dim_receipt.is_purchase	-	Generado en función de la tabla origen
dim_receipt.receipt_id	vfactura_id	
dim_receipt.billing_customer_name	nombrecli_cff	
dim_receipt.billing_province	provincia_cff	
dim_receipt.receipt_key	autoincremental	
dim_receipt.payment_term	pago_id	tabla fa_pagos campo desc_pago
dim_receipt.billing_address	direccion_cff	
dim_receipt.billing_country	pais_cff	
dim_receipt.amount_taxes	totalfact_cff	array, obtener primer valor
dim_receipt.is_sale	-	Generado en función de la tabla origen
dim_receipt.billing_city	poblacion_cff	

Figura B.10: Dimensión de factura

Tabla origen:	*fa_linfac y gc_linfac*	
Objetivo	**Fuente**	**Observaciones**
dim_receipt_line.is_purchase	-	Generado en función de la tabla origen
dim_receipt_line.is_sale	-	Generado en función de la tabla origen
dim_receipt_line.receipt_id	vfactura_id	
dim_receipt_line.amount	importe_lff	
dim_receipt_line.article_qty	cantidad_lff	
dim_receipt_line.description	desc_lff	
dim_receipt_line.article_id	articulo_id	
dim_receipt_line.receipt_line_key	autoincremental	
dim_receipt_line.line_number	linea_lff	
dim_receipt_line.article_price	precio_lff	

Figura B.11: Dimensión linea de factura

B.2. Transformaciones

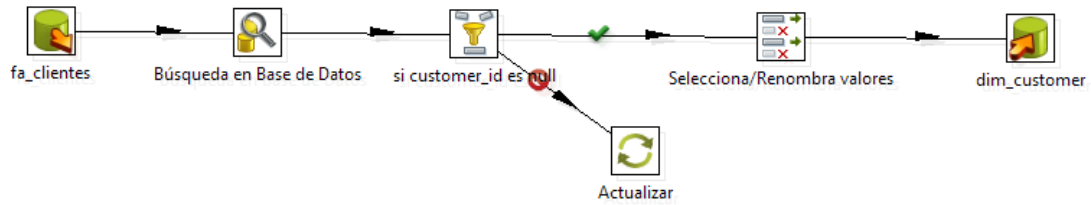


Figura B.12: Dimensión de cliente

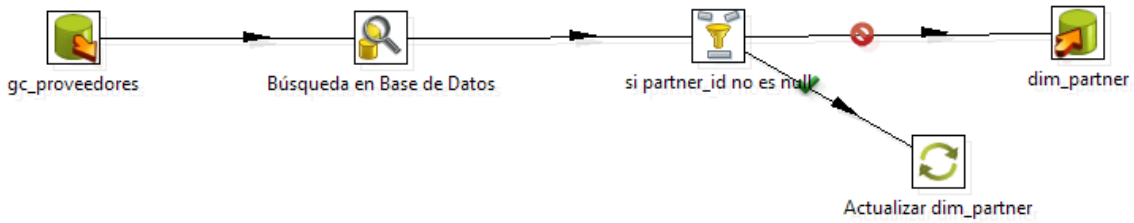


Figura B.13: Dimensión de proveedor

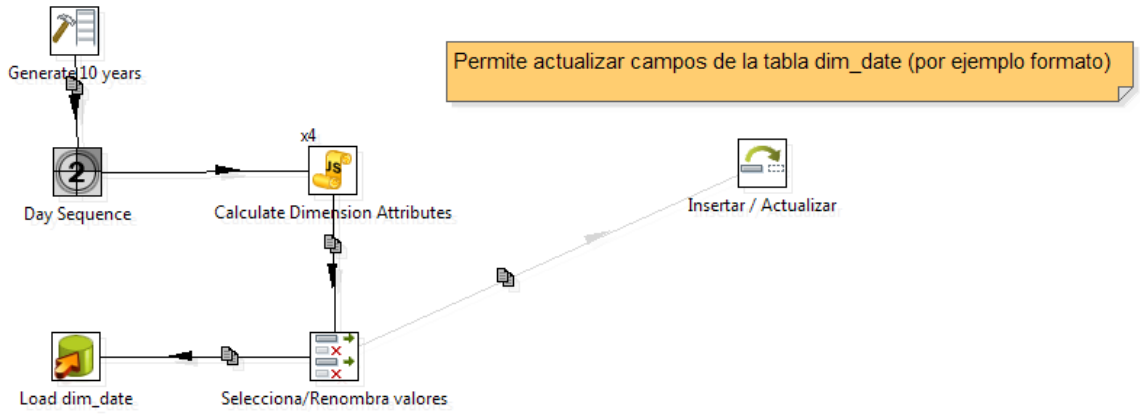


Figura B.14: Dimensión de fecha

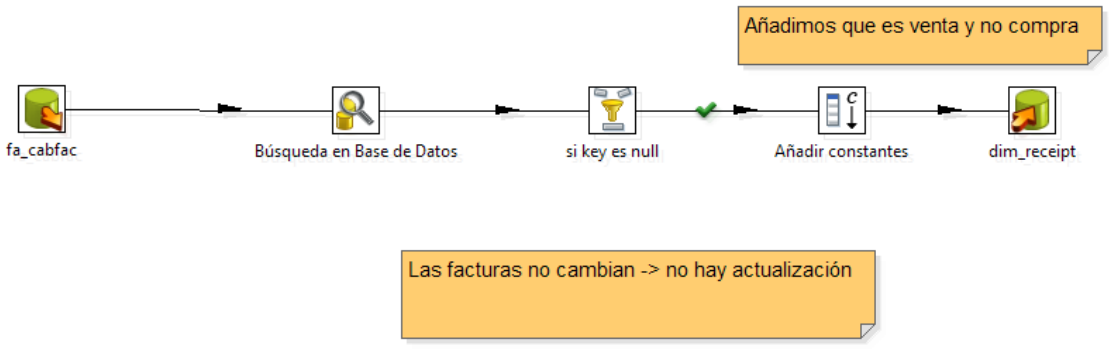


Figura B.15: Dimensión de factura

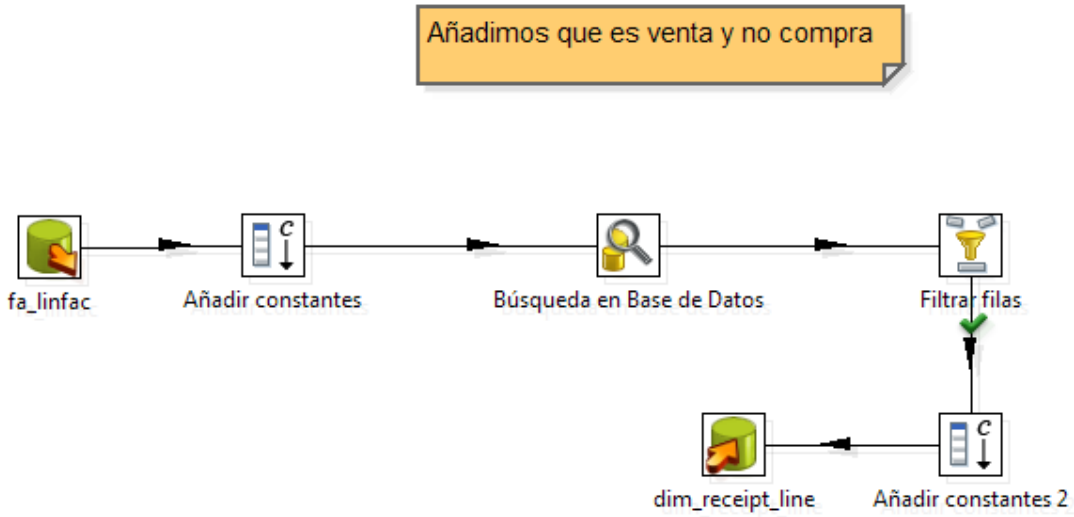


Figura B.16: Dimensión de línea de factura

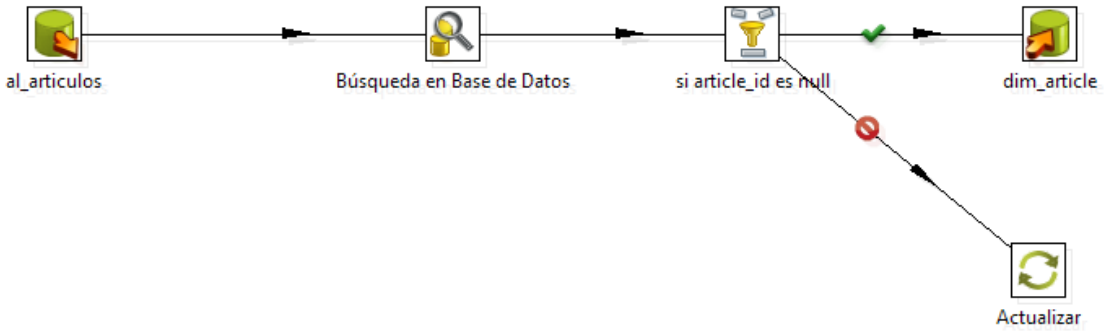


Figura B.17: Dimensión de artículo

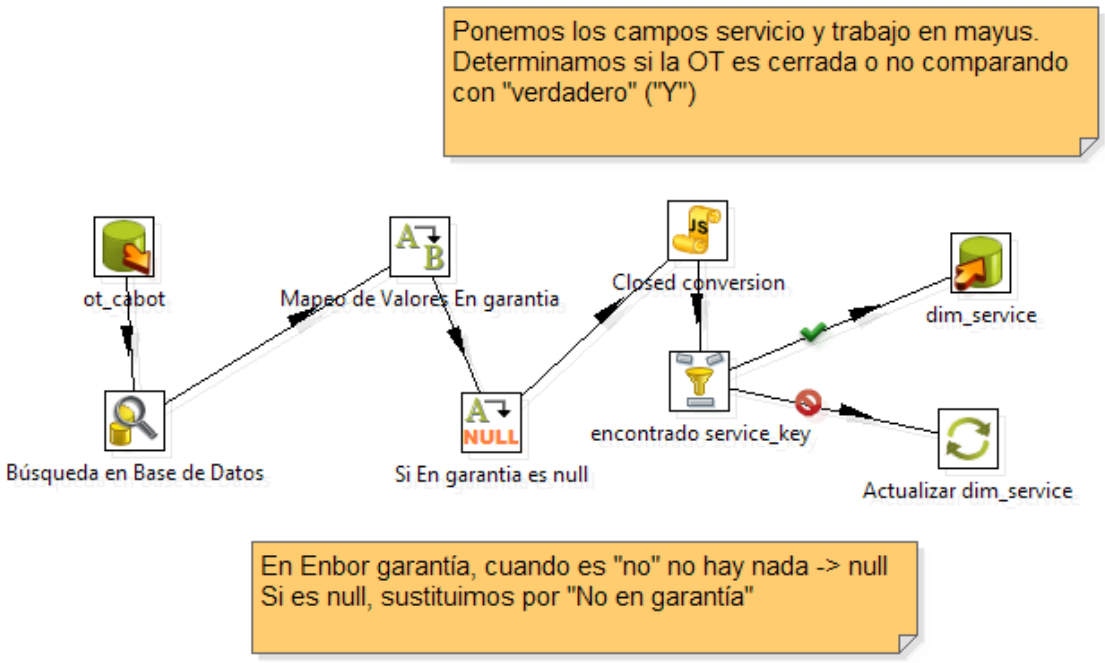


Figura B.18: Dimensión de orden de trabajo

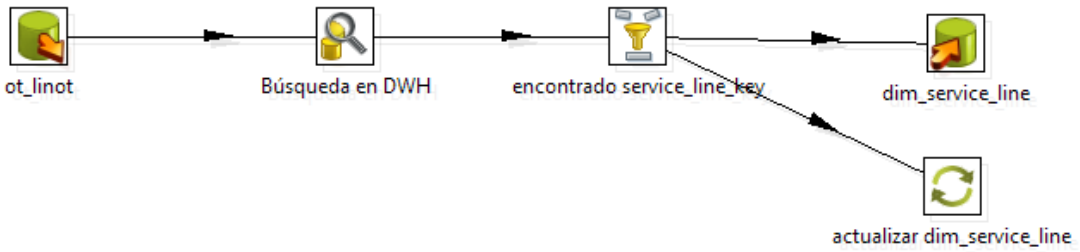


Figura B.19: Dimensión de línea de orden de trabajo

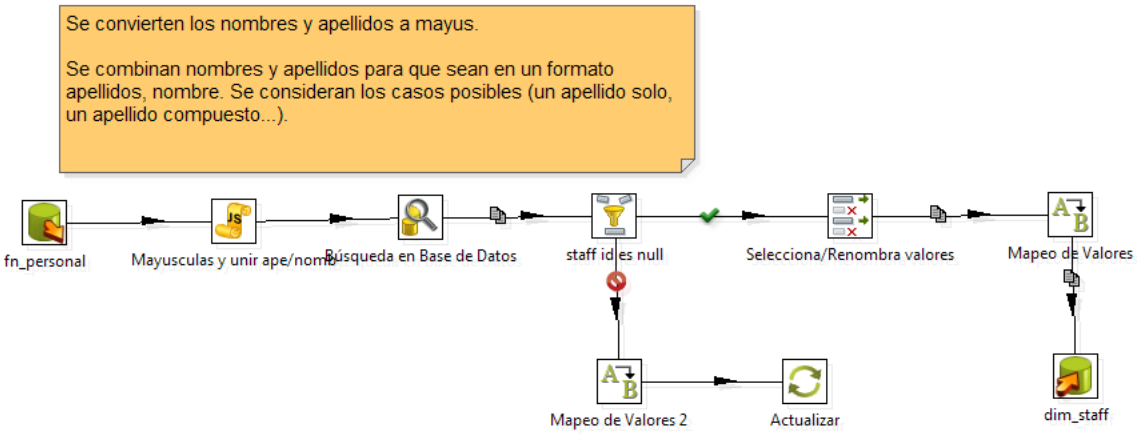


Figura B.20: Dimensión de empleado

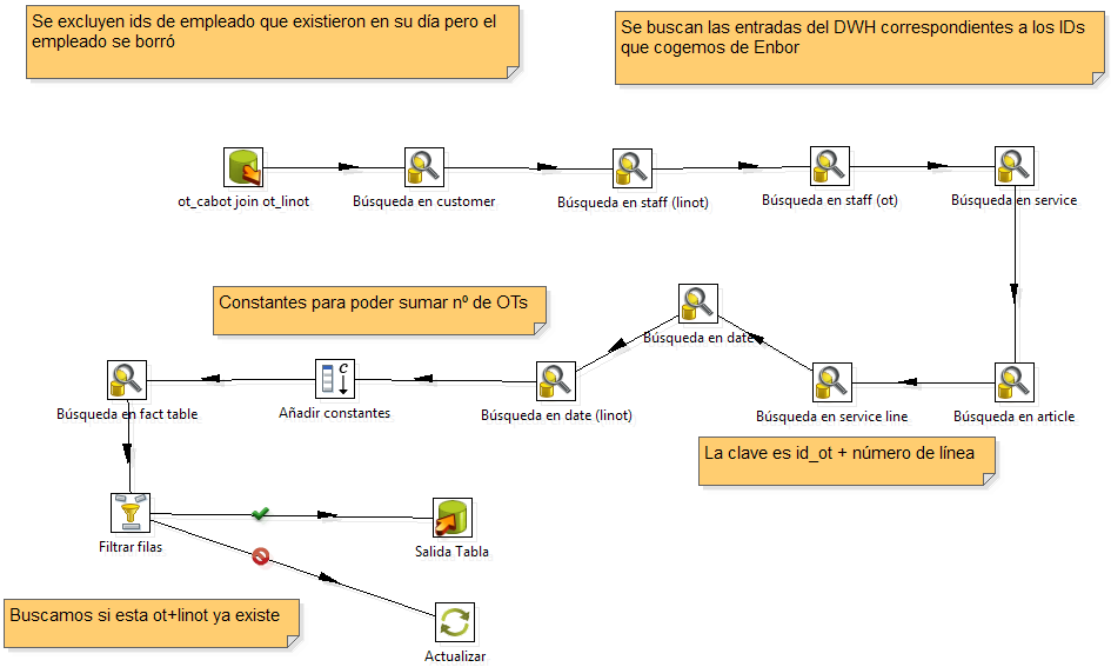


Figura B.21: Tabla de hechos de avisos

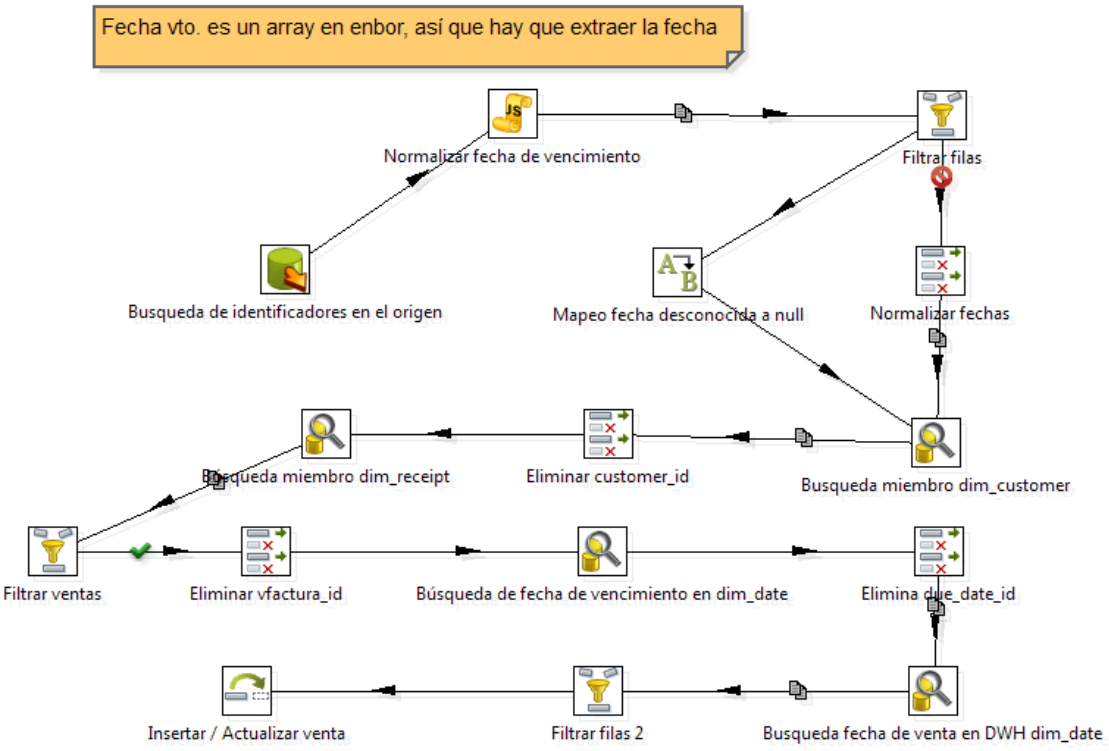


Figura B.22: Tabla de hechos de facturas

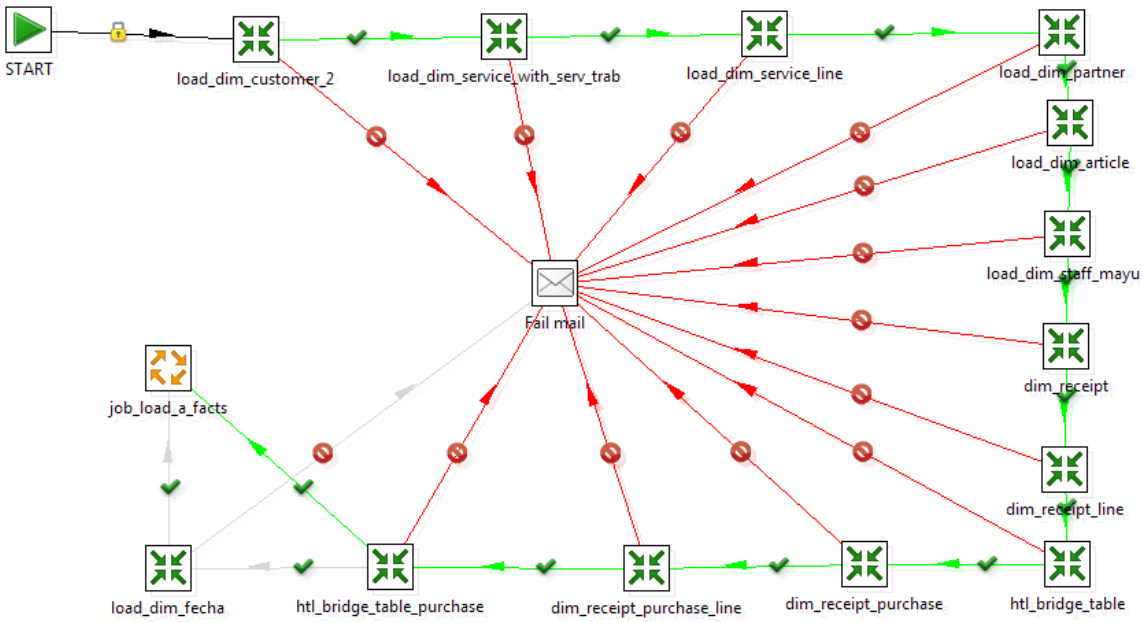


Figura B.23: Trabajo de carga de dimensiones

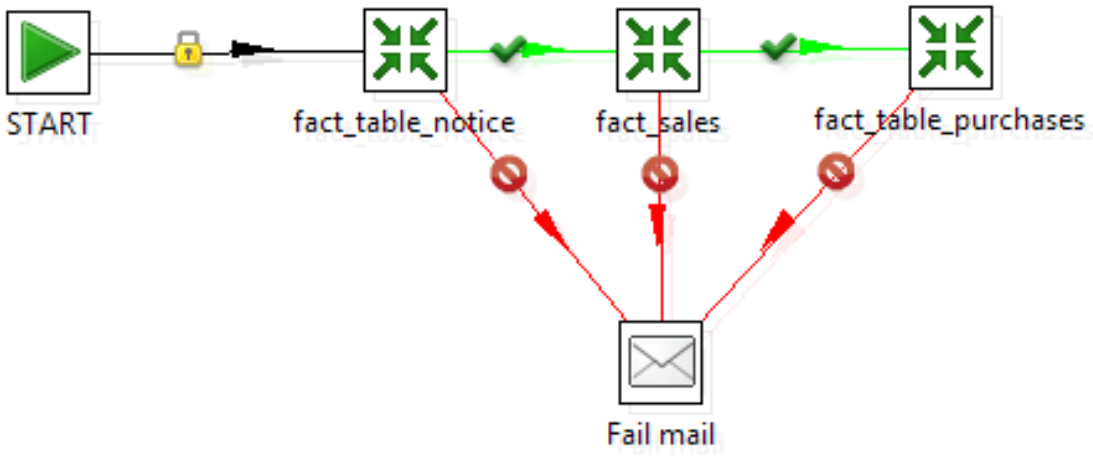


Figura B.24: Trabajo de carga de tablas de hechos

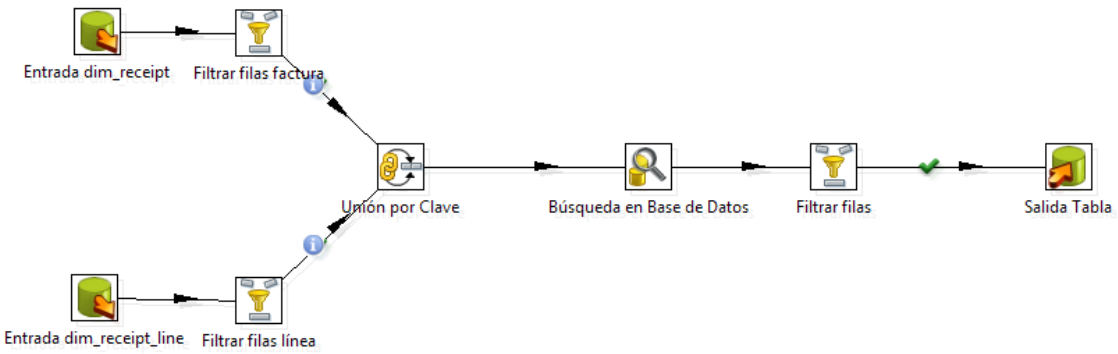


Figura B.25: Tabla puente de facturas y líneas

Anexos C

Cuadro de mando integral

C.1. Prototipos cuadros de mando

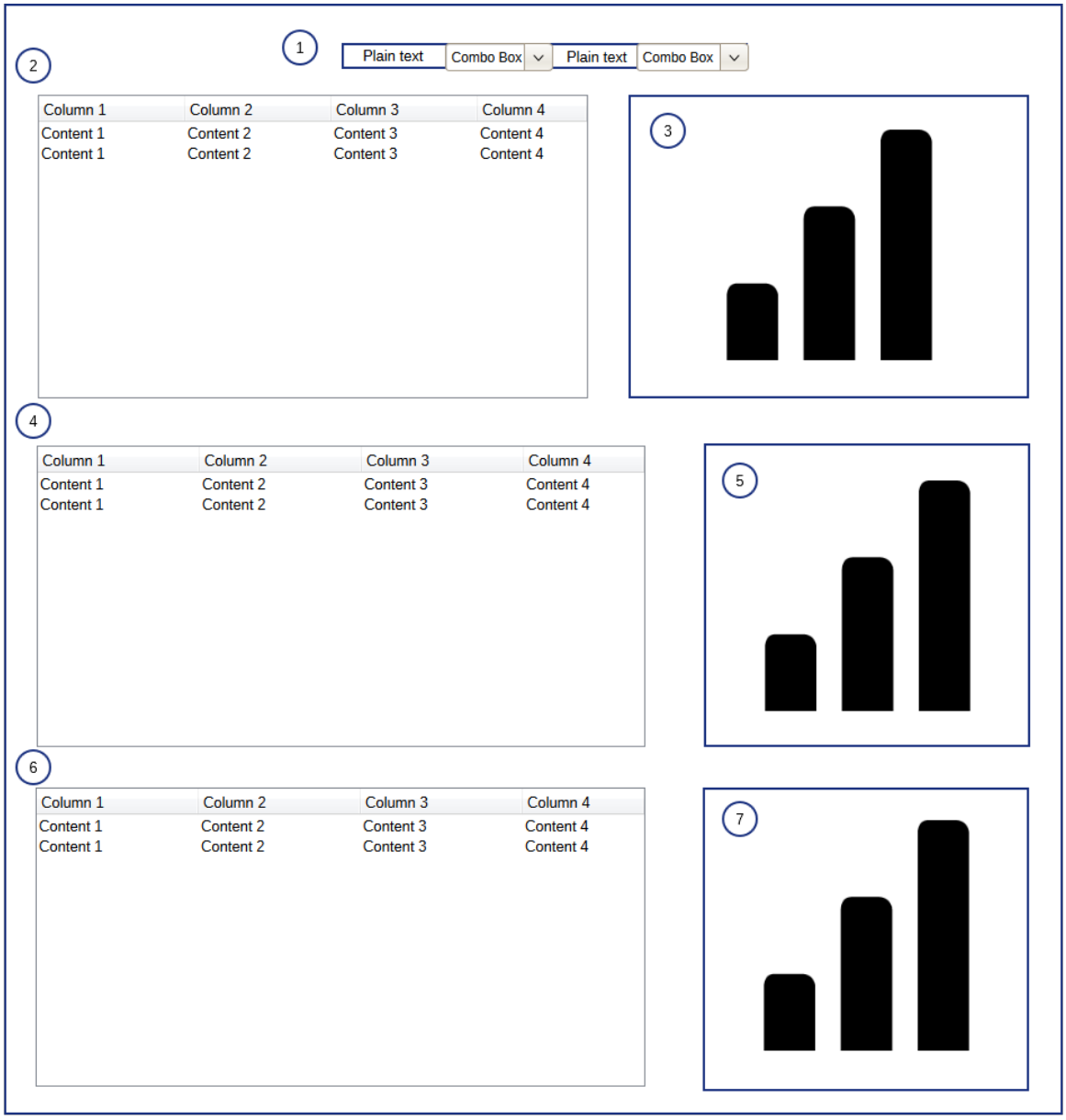


Figura C.1: OTs general

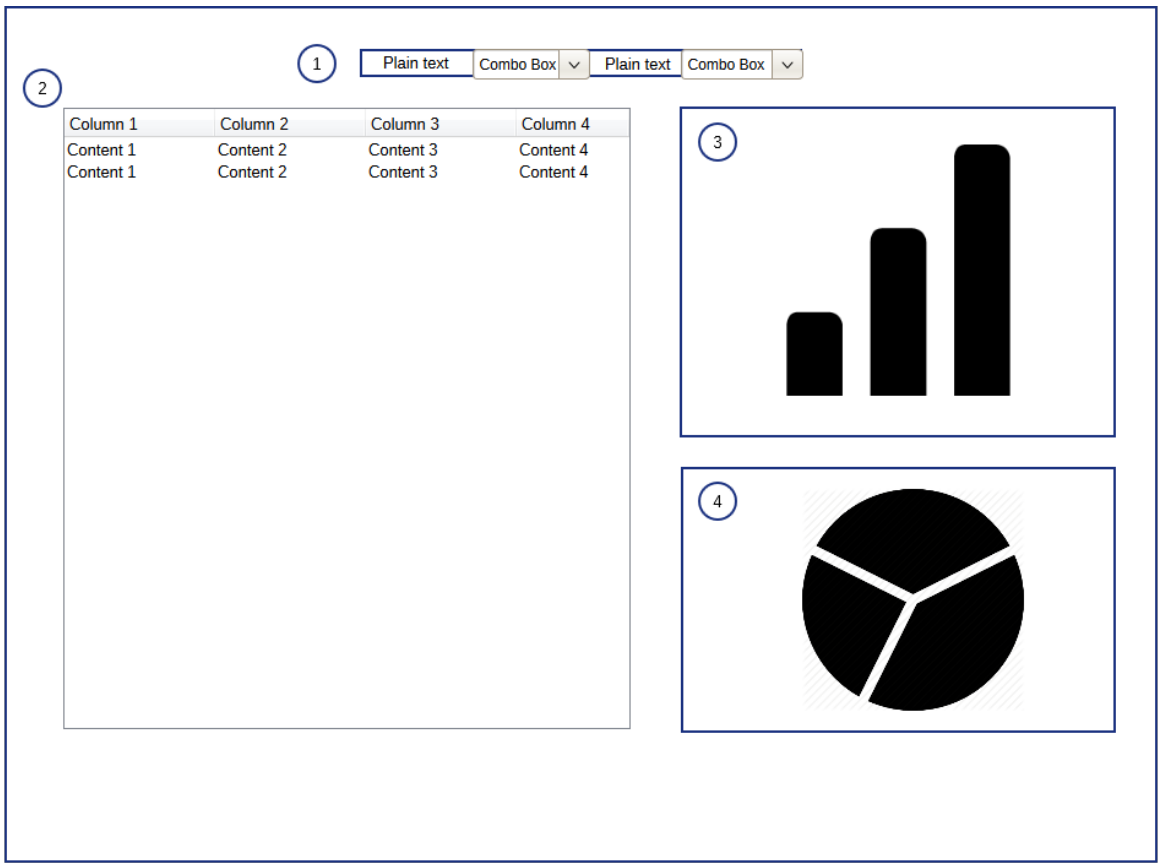


Figura C.2: OTs por empleado

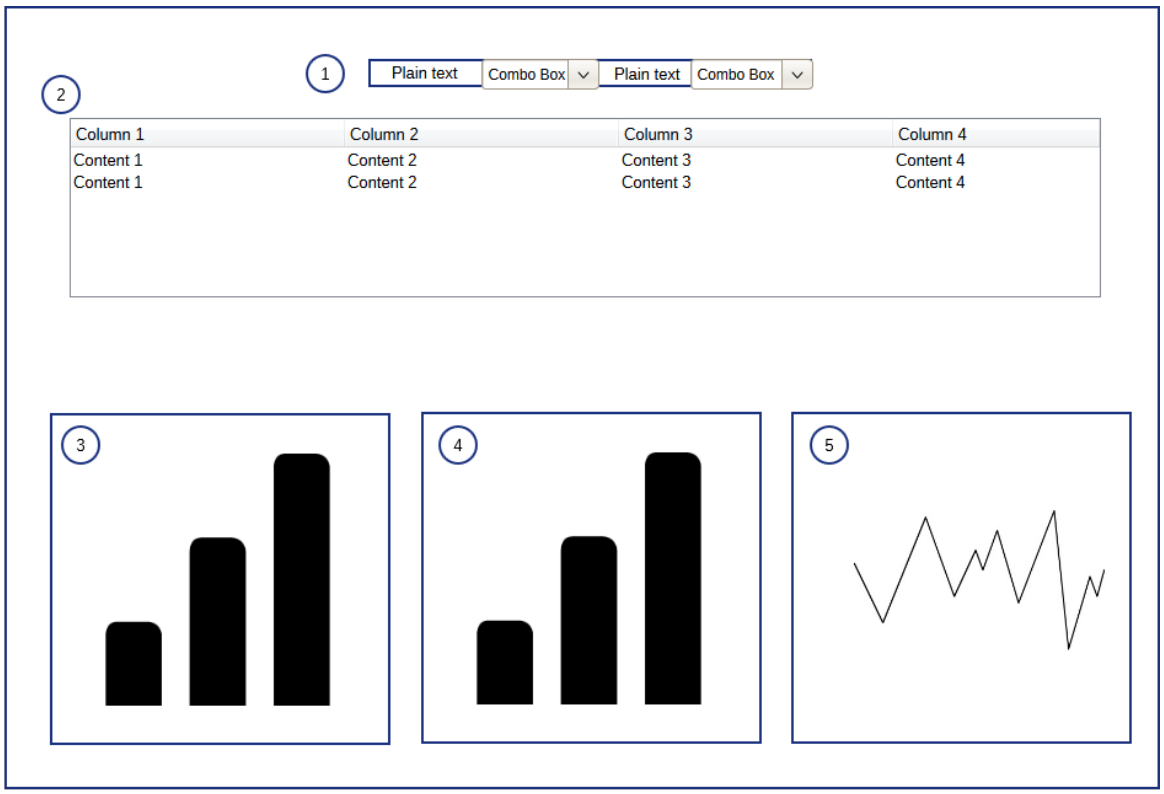


Figura C.3: Ventas general

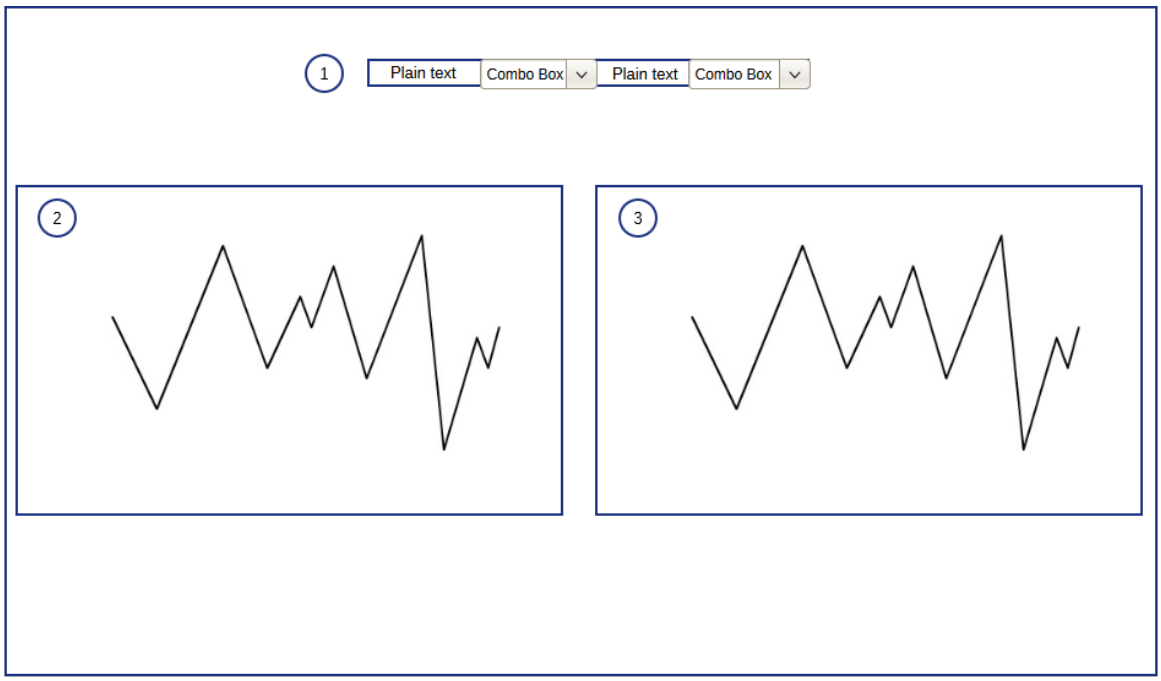


Figura C.4: Ventas vs OTs

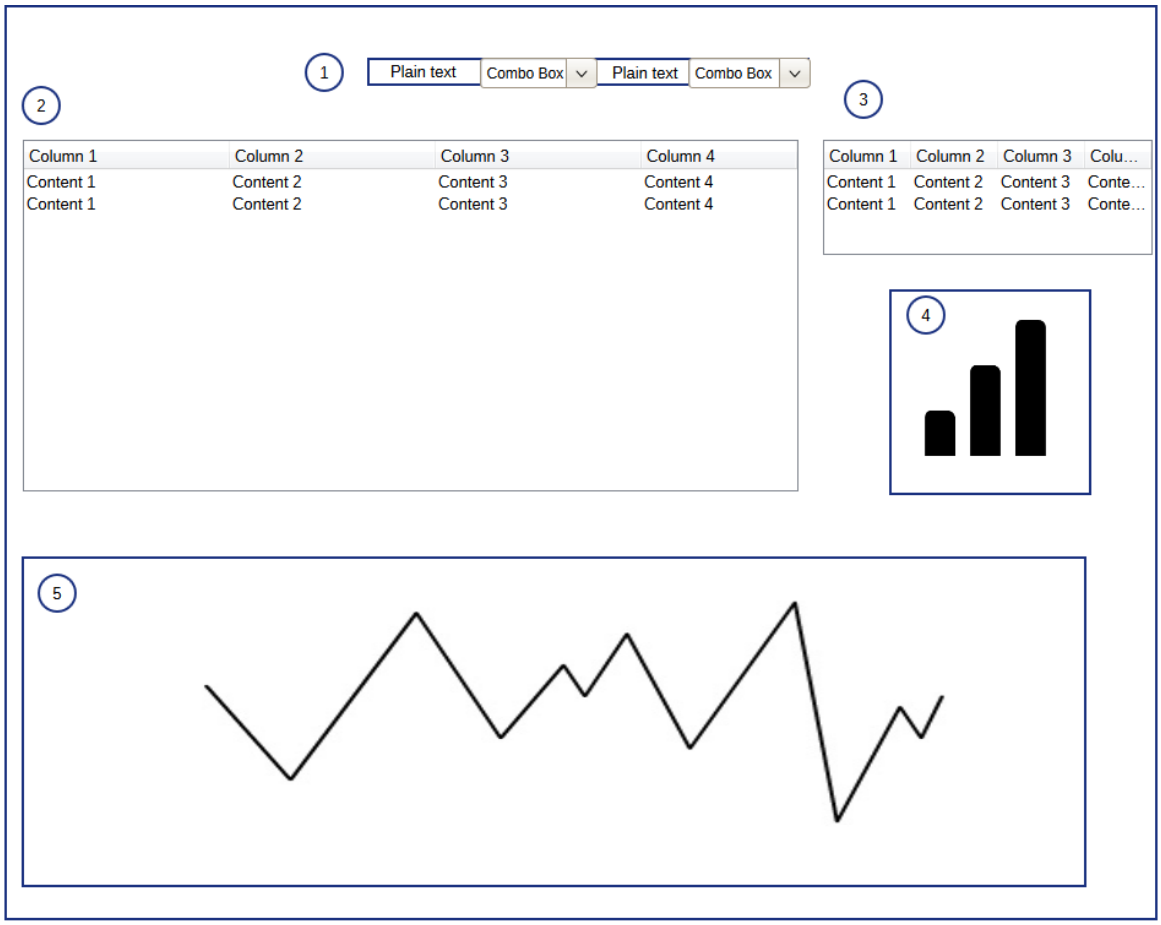


Figura C.5: Artículos

Anexos D

Principales problemas encontrados y soluciones adoptadas

D.1. MDX

Este apartado recoge los proncipales problemas encontrados haciendo uso de latecnología MDX y cuales han sido las soluciones adoptadas.

Descripción del problema: Celdas múltiples repetidas
Responsable: Josu Rodríguez
Descripción detallada
Según la documentación de Mondrian, MDX puede devolver celdas repetidas en varias ocasiones, con los mismos datos. No se concreta la causa.

Solución
Utilizar la función "Distinct" en la sentencia MDX. En la sentencia originaria del problema, quedaría de la siguiente forma:

```
WITH
SET [~ROWS\_Servicio.Encargado] AS
    {[Servicio.Encargado].[Encargado].Members}
SET [~ROWS\_Tiempo.Jerarqua de fechas] AS
    {[Tiempo.Jerarqua de fechas].[Anio].Members}
SELECT
NON EMPTY {[Measures].[Amount notices]} ON COLUMNS,
NON EMPTY Distinct(NonEmptyCrossJoin([~ROWS\_Servicio.Encargado],
[~ROWS\_Tiempo.Jerarqua de fechas])) ON ROWS FROM [Notice]
```

Descripción del problema: Función Order
Responsable: Iñigo Sánchez
Descripción detallada
Al ejecutar la función order y limitar el resultado del set al que se debería aplicar, el resultado no es el esperado. El orden respetado es el global porque se aplica antes de obtener el resultado del set.

Solución
Utilizar otra función en la sentencia MDX dentro de la función set. Esto permite aplicar el order sobre el resultado de la función aplicada previamente, quedaría de la siguiente forma:

```
 Order(TopCount([Servicio].[Estado].Members,[Servicio].[Estado].Members.count)
...
```

Descripción del problema: Miembros duplicados con determinadas sentencias
Responsable: Josu Rodríguez
Descripción detallada
Al ejecutar sentencias que se refieren a miembros de una jerarquía que no sean del nivel superior, se producen duplicados por cada valor del nivel padre que esté asociado a dicho miembro.

Solución
PENDIENTE
Enlaces
No disponible

Descripción del problema: La función tail se aplica sobre la dimensión.
Responsable: Iñigo Sánchez
Descripción detallada
La sentencia NON EMPTY se aplica sobre el resultado (del set), p. e. que si queremos los últimos 5 años con datos y tenemos una dimensión fecha con metadatos en fechas futuras, la consulta OBTENDRÁ LOS ÚLTIMOS CINCO AÑOS CARGADOS EN LA DIMENSIÓN.
Solución
PENDIENTE

D.2. BI Server

Este apartado recoge los principales problemas encontrados al desarrollar los cuadros de mando con la herramienta *Pentaho* y el componente *CDE*.

Descripción del problema: Acceso a variables de entorno
Responsable: Josu Rodríguez
Descripción detallada
Se necesita acceder a las variables de entorno del BI Server, en este caso a los roles y al nombre de usuario.

Solución
Utilizar la función "Dashboards.context.x" (p.ej.: Dashboards.context.roles) en la función Javascript. Si se desea acceder en una MDX o SQL, se usaría env::username.

Descripción del problema: Cambiar valor de parámetro y refrescar
Responsable: Josu Rodríguez, Iñigo Sánchez
Descripción detallada
Se desea cambiar el valor de un parámetro al hacer click en una tabla, y que los gráficos que escuchan a ese parámetro se actualizen.

Solución
Utilizar la función "Dashboards.fireChange("parámetro", valor)" en la función Javascript del onClick.

Descripción del problema: Cambiar DataSource de componentes
Responsable: Josu Rodríguez
Descripción detallada
Se necesita modificar las fuentes de datos de los componentes vía JavaScript, y no necesariamente desde el mismo componente.

Solución
Utilizar la función "*.queryDefinition.DataAccessId" (p.ej.: render_SelectEmpleado.queryDefinition.dataAccessId="mdx_lista_empleados";) en la función Javascript.

Descripción del problema: Uso de parámetros del *DataRange*.
Responsable: Iñigo Sánchez
Descripción detallada
Conviene tener en cuenta el orden en el que se definen los parámetros que capturan las fechas desde y hasta, de otra manera la consulta los recibirá invertidos ofreciendo un resultado NO esperado, generalmente vacío.

Solución
Definir primero el parámetro que captura la fecha desde y segundo el que captura la fecha hasta en la sección de componentes.

Descripción del problema: Implementación incorrecta de *PopUps*.
Responsable: Iñigo Sánchez
Descripción detallada
Al implementar un componente popup igual que el resto, con los parámetros definidos y los listenner el popup deja de mostrar el contenido.

Solución
No incluir el *listenner* del parámetro en el popup, pero si definir el parámetro.

Descripción del problema: Alert javascript:Nombres de las columnas del componente *Table*.
Responsable: Iñigo Sánchez
Descripción detallada
Si definimos los nombres de las columnas en la tabla y creamos una consulta que nos devuelva los registros no vacíos, al cargar la tabla e intentar incluir valores nulos en la fila y columna correspondiente, aparece un molesto alert de aviso.

Solución
No incluir en la consulta *NON EMPTY*.

Descripción del problema: Expand de las tablas
Responsable: Josu Rodríguez
Descripción detallada
Al hacer un expand, buscar en la tabla y luego intentar hacer expand otra vez, el gráfico desaparece.

Solución
Es un bug. Pendiente evaluar en la versión 5.3.

Descripción del problema: Problema al usar dos popup en un dashboard
Responsable: Josu Rodríguez
Descripción detallada
Al implementar dos popups en un mismo dashboard, solo se ve uno de ellos.

Solución
Los divs en los que se metan no tienen que tener atributo hidden. El mismo componente popup se encarga de ello.

Descripción del problema: No es posible cambiar el valor del atributo hidden
Responsable: Iñigo Sánchez
Descripción detallada
Al usar la función *setAtributte* de javascript para cambiar el valor al atributo *hidden* y ponerlo a *False* el elemento del DOM sigue oculto.

Solución
Para volver a mostrar un elemento oculto aplicamos la función javascript *removeAtributte("hidden")*.
No disponible

Descripción del problema: Cambiar Datasources de componentes
Responsable: Josu Rodríguez
Descripción detallada
Diferentes componentes requieren un tratamiento distinto del cambio de datasource.

Solución
En caso de componentes que no sean gráficos: this.queryDefinition.dataAccessId=iddatasource.
Si son gráficos: this.chartDefinition.dataAccessId=iddatasource.

D.3. PostgreSQL

En este apartado quedan recogidos los principales problemas que involucran PostgreSQL y las soluciones adoptadas.

Descripción del problema: Uso de parámetros de fecha con PostgreSQL
Responsable: Josu Rodríguez
Descripción detallada El uso de parámetros de fecha con PostgreSQL no funciona simplemente haciendo una llamada al mismo. **Solución** Hacer uso de la función todate(${param1_FromDate}, 'YYYY MM DD') en la sentencia SQL y determinar en los parámetros de entrada de la sentencia que serán de tipo String, pese a que estén definidos como Date en el dashboard.