

Plataforma web BI: Automatización de informe y creación de tabla dinámica a partir de cubo de datos OLAP

Sergio González Granizo

Resumen— Este proyecto surge de la propia experiencia laboral. He podido observar lo importante que es para una empresa la información, como por ejemplo, los informes para tener un buen control y gestión sobre la misma. He comprobado que el proceso para obtener esta información no es siempre adecuado y debe mejorarse. Este proyecto es una solución que optimiza los recursos y mejora la información generada. La información se centraliza en una plataforma web, que además dispone de un control de accesos según credenciales. La plataforma dispone de un proceso que automatiza la generación de un informe. Esto permite que el tiempo que se dedica a la generación del informe, se pueda dedicar a otras tareas. Además, la plataforma dispone de una tabla dinámica JPivot, alimentada de un cubo de datos OLAP, que permite navegar desde datos globales hasta obtener el detalle deseado.

Palabras clave— Pentaho, Pentaho Data Integration, Pentaho Report Designer, Spoon, Pentaho BI Server, BDD Estrella, Automatización informe, Jpivot, Tabla dinámica, Cubo OLAP, Dimensiones.

Abstract— This project arises from my own professional experience. I've been able to observe how important is for a company the information. For example, the reports to have a good control and management over itself. I've come to see that the process to obtain this information is not always suitable and must be improved.

This project is a solution which optimizes the resources and improves the generated information. The information is centralized in a web platform which has an access control with credentials. The platform has an automatized report generator. This allows to dedicate the time in other tasks instead of the generation of it. Also, the platform contains a dynamic JPivot table, fed with a OLAP data cube which allows to navigate from global data to the desired detail.

Index Terms— Pentaho, Pentaho Data Integration, Pentaho Report Designer, Spoon, Pentaho BI Server, BDD Star, Report automatization, JPivot, Dynamic table, OLAP cube, Dimensions



1 INTRODUCCIÓN

La motivación de este trabajo, surge de la propia experiencia laboral. Trabajando varios años en el servicio informático de un conocido banco, y actualmente en una consultora de Software, he podido comprobar que se necesita disponer de información e informes, en los distintos departamentos, para tener un control sobre la actividad que se realiza y su estado.

Se ha observado que la forma de generar esta información y distribuirla es ineficiente y poco segura. Este trabajo pretende ser una solución eficiente y segura para generar la información e informes necesarios.

La información es muy valiosa y fundamental en la toma de decisiones. Es importante que sea accesible fácilmente y que se ajuste a la necesidad de la organización.

He querido que mi trabajo tuviera un caso real de una organización. Con el permiso de la empresa, he decidido utilizar datos reales de un proyecto.

Para cumplir con la LOPD (Ley Orgánica de Protección de Datos) y con el secreto profesional, he modificado o eliminado los datos que la empresa ha considerado oportunos. No haré mención al Proyecto, ni mención de ninguna de las empresas involucradas.

Para una mejor comprensión del documento ha sido estructurado en las siguientes secciones

- 2. Situación actual
- 3. Propuesta
- 4. Metodología
- 5. Planificación
- 6. Líneas Futuras
- 7. Valoración del resultado
- 8. Conclusiones

- E-mail de contacto: sergogra@gmail.com
- Mención realizada: Ingeniería del Software.
- Trabajo tutorizado por: Francisco Javier Sanchez Pujades
(Departamento de Ciencias de la Computación)
- Curso 2015/16

2 SITUACIÓN ACTUAL

Actualmente, el departamento genera la información manualmente. En concreto, la forma de generar los informes y distribuirlos, siguen el siguiente esquema:



Ilustración 1 - Esquema generación manual de informes

- Se realiza una extracción de datos de la aplicación en ficheros tipo Excel o CSV.
- Se manipulan manualmente los ficheros y se vuelcan a una base de datos.
- Mediante el Excel se generan los informes manualmente dándole el formato deseado.
- Se envían mediante correo electrónico utilizando listas de distribución.

Este proceso presenta varios posibles errores, los más destacados son los siguientes:

- Al ser manuales, influye el factor de error humano.
- No se explota suficiente la información que se podría extraer de los datos.
- Permiten pocas modificaciones en su diseño o información a mostrar.
- Distribuir por correo electrónico es ineficaz y poco seguro.
- No hay un control sobre quién debe tener acceso a la información.
- Un coste de tiempo y personas elevado para su generación.

Dependiendo del número de informes necesarios en el día, se requiere más tiempo y personas.

El informe diario, por ejemplo, requiere una persona durante un tiempo mínimo 4 horas. Puede aumentar si se producen errores.

3 PROPUESTA

La propuesta busca mejorar la información del departamento, así como mejorar todo el proceso para su generación.

Para este fin, se va a crear una plataforma web centralizada, que genere de forma automática los informes, que explote nuevas formas de mostrar la información y tenga un control de accesos para mejorar la disponibilidad y seguridad de los datos

La propuesta aporta las siguientes mejoras:

- Actualización automática diaria de la BDD.
- Generación automática de informes. Por lo que se

ahorran recursos de personas y tiempo.

- Modelado de datos en cubo OLAP, lo que permite visualizar los datos en tablas dinámicas.
- Centralización de los datos en la plataforma web. Lo que permite tener un control de accesos, mejorando así la seguridad y mejorando la accesibilidad y disponibilidad de los datos.
- Cubo de datos en la plataforma, añade funcionalidad. Permite utilizar herramientas para realizar informes personalizados (Por ejemplo, Saiku Analytics).

3.1 Herramientas Pentaho

Para que los siguientes apartados sean más comprensibles, vamos a hablar de las herramientas principales utilizadas para la realización del proyecto. Se trata de herramientas Pentaho. Estas herramientas sirven para generar Inteligencia Empresarial.



Entre otras posibles soluciones, la he elegido porque tiene muchas funcionalidades que resultan útiles para el proyecto y ayudan a cumplir los propósitos. Además, es de uso libre y tiene una gran comunidad detrás.

Las herramientas Pentaho utilizadas en el proyecto son:

- Pentaho Data Integration: Es un conjunto de herramientas. La principal es *Spoon*, una interfaz gráfica que permite diseñar transformaciones y trabajos que se pueden ejecutar con *Pan*. Pan es un motor de transformación de datos que realiza funciones como lectura, manipulación y escritura de datos desde fuentes de datos.
- Pentaho Report Designer: Es una herramienta que permite la creación automática de informes. Permite generarlos en HTML, Excel y PDF.
- Pentaho Schema Workbench: Es una herramienta gráfica que permite la creación de cubos OLAP.
- Pentaho BI Server: Es una herramienta que permite crear una plataforma web en la que centralizar todos los datos.

3.2 Plataforma web BI

Para tener la plataforma web BI se necesita realizar varias fases previas que se van a detallar en los siguientes apartados.

3.2.1 Extracción de datos

Los datos de los que se quiere obtener información pertenecen a una herramienta de gestión de incidencias. Se necesita extraer los datos de esta aplicación a una base de datos para poder trabajar con ellos posteriormente.

La aplicación de gestión de incidencias solo permite extraer los datos en ficheros con formato .csv

Se quiere crear un proceso automático que actualice diariamente los datos en la BDD.

Para la realización de esta fase se ha utilizado la herramienta Pentaho Data Integration – Spoon. Se ha creado el siguiente proceso:



Ilustración 2 - Proceso de extracción y carga de datos

- **Comprobación ficheros:** Comprueba que en la ruta especificada se encuentra el documento esperado. Es decir, el archivo .csv que se genera al extraer los datos de la aplicación de gestión de incidencias.
- **Transformación Jira:** Se especifica que columnas del archivo .csv vamos a insertar en BDD.
- **SQL:** Se insertan los datos en la BDD.
- **Move Files:** Mueve el fichero .csv ya procesado a una carpeta que contiene el histórico.

3.2.2 Generación automática del informe

En este paso tenemos los datos de la aplicación actualizados en la BDD. Pero los datos no están agrupados ni calculan las métricas que queremos en el informe.

Se podría hacer las agrupaciones y los cálculos de las métricas realizando *Querys* al generar el informe. Pero esto provocaría que el tiempo de generación del informe fuera muy elevado.

Por ello se ha decidido estructurar los datos con las agrupaciones y las métricas calculadas en la base de datos. Se ha creado un *Stored Procedure* que realiza la agrupación y cálculos. Se ejecuta cada día, justo después de actualizar los datos. Inserta los datos agrupados, con las métricas calculadas, en una nueva tabla.

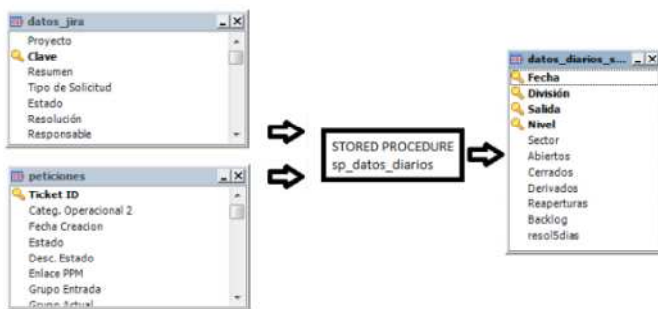


Ilustración 3 - Estructuración de la base de datos

Con los datos ya estructurados en la tabla, el siguiente paso es la automatización del informe. Para ello vamos a utilizar la herramienta Pentaho Report Designer.

Se ha diseñado el informe de seguimiento diario siguiendo el modelo que se genera manualmente. Solicita como parámetro introducir la fecha de la que se desea generar el in-

forme y el formato en el que se quiere generar (HTML, Excel, PDF). Con esta información se genera de forma automática.

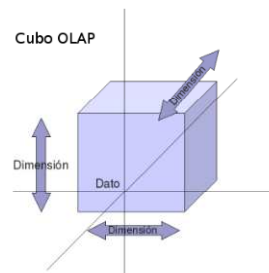
Seguimiento diario 03/02/2015

Tiqueting HDn2

NIVEL 2					
Fak	Módulo	Abiertos	Cerrados	Derivados	Backlog
ESPA	EP3	80	103	7	230
ESPA	GP3	10	7	2	28
ESPA	ATEC	8	19	1	17
ESPA	ATEC-4	13	10	1	22
MEXICO	EP3	12	19	1	29
BRASIL	EP3	9	5	1	17
TOTAL		132	168	13	348

Ilustración 4 - Ejemplo de informe diario

3.2.3 Cubo de datos OLAP



Uno de los objetivos de este trabajo es mejorar la información que se extrae de los datos. El uso del cubo de datos, permite realizar este objetivo.

Primero vamos a explicar brevemente qué es un cubo OLAP. El término cubo OLAP viene de “OnLine Analytical Processing” (Procesamiento Analítico en Línea). Se trata de una base de datos multidimensional.

Disponer los datos en cubos evita una limitación de las bases de datos relacionales, que no son muy adecuadas para el análisis instantáneo de grandes cantidades de datos.

El almacenar físicamente los datos de esta forma tiene sus pros y sus contras. Por ejemplo, en estas bases de datos las consultas de selección son muy rápidas (de hecho, casi instantáneas). Pero uno de los problemas más grandes de esta forma de almacenamiento es que una vez poblada la base de datos ésta no puede recibir cambios en su estructura. Para poder diseñar el cubo de datos, necesitamos una base de datos que siga el esquema en estrella.

Como podemos observar en la imagen, este modelo recibe el nombre tipo “Estrella” por la forma parecida a una estrella del esquema.

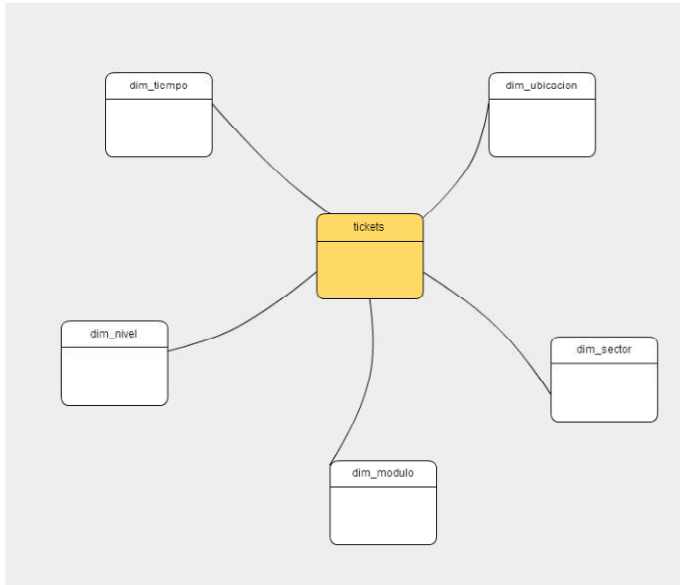


Ilustración 3 - Modelo de base de datos en estrella

Este modelo de datos que tiene una tabla de hechos (tabla que contiene los datos para el análisis), rodeada de las tablas de dimensiones.

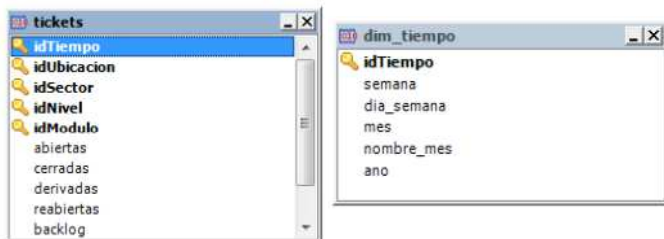


Ilustración 4 - Tabla de hechos "tickets" y tabla dimensión "Tiempo"

Una vez tenemos diseñado e implementado el nuevo modelo de datos en la base de datos. Tenemos que definir el cubo de datos. Para ello vamos a utilizar la herramienta Pentaho Schema Workbench.

La herramienta necesita que se defina que tabla corresponde a la tabla de “Hechos” y definir todas las métricas que se quieren utilizar. Además se deben definir todas las “Dimensiones” y definir los “Niveles” que se van a utilizar en cada una de ellas. La “Dimensión Tiempo” es particular, los niveles ya están definidos y se deben elegir los deseados. En nuestro caso, los niveles de la dimensión tiempo son: Año, Mes, Semana y Día.



Ilustración 5 - Esquema definición cubo de datos

Lo que genera este esquema es un archivo XML con la definición del Cubo de datos. Pero la herramienta de forma gráfica ayuda a hacer un mejor diseño.

```
<?xml version="1.0"?>
<Schema name="Ticketing">
  <Table name="tickets">
    <Table>
      <Dimension name="Tiempo" visible="true" highCardinality="false" foreignKey="idTiempo" type="TimeDimension">
        <Table name="dim_tiempo">
          <Table>
            <Level name="Año" visible="true" type="String" hideMembers="Never" levelType="TimeYears" uniqueMembers="false" column="ano">
              <Level>
                <Level name="Mes" visible="true" type="String" hideMembers="Never" levelType="TimeMonths" uniqueMembers="false" column="mes">
                  <Level>
                    <Level name="Semana" visible="true" type="String" hideMembers="Never" levelType="TimeWeeks" uniqueMembers="false" column="semana">
                      <Level>
                        <Level name="Dia" visible="true" type="String" hideMembers="Never" levelType="TimeDays" uniqueMembers="false" column="idTiempo">
                          <Level>
                        </Level>
                      </Level>
                    </Level>
                  </Level>
                </Level>
              </Level>
            </Level>
          </Table>
        </Table>
      </Dimension>
      <Dimension name="Ubicacion" visible="true" highCardinality="false" foreignKey="idUbicacion" type="StandardDimension">
        <Table name="dim_ubicacion">
          <Table>
            <Level name="País" visible="true" type="String" hideMembers="Never" levelType="Regular" uniqueMembers="false" column="id_pais" nameColumn="nombre_pais">
              <Level>
            </Level>
          </Table>
        </Table>
      </Dimension>
      <Dimension name="Nivel" visible="true" highCardinality="false" foreignKey="idNivel" type="StandardDimension">
        <Table name="dim_nivel">
          <Table>
            <Level name="Modulo" visible="true" type="String" hideMembers="Never" levelType="Regular" uniqueMembers="false" column="id_modulo" nameColumn="nombre_modulo">
              <Level>
            </Level>
          </Table>
        </Table>
      </Dimension>
      <Dimension name="Sector" visible="true" highCardinality="false" foreignKey="idSector" type="StandardDimension">
        <Table name="dim_sector">
          <Table>
            <Level name="Sector" visible="true" type="String" hideMembers="Never" levelType="Regular" uniqueMembers="false" column="id_sector" nameColumn="nombre_sector">
              <Level>
            </Level>
          </Table>
        </Table>
      </Dimension>
      <Measure name="Abiertas" type="Number" column="abiertas">
      </Measure>
      <Measure name="Cerradas" type="Number" column="cerradas">
      </Measure>
      <Measure name="Derivadas" type="Number" column="derivadas">
      </Measure>
      <Measure name="Backlog" type="Number" column="backlog">
      </Measure>
    </Table>
  </Table>
</Schema>
```

Ilustración 6 - Código XML generado con Schema Workbench

3.2.4 Tablas dinámicas Jpivot

Uno de los beneficios de usar cubos de datos y tener un modelo de base de datos multidimensional, es usar tablas multidimensionales para mostrar los datos. Las tablas Jpivot permiten navegar cubos de datos OLAP y mostrar la información en forma de tabla, soportando funcionalidad como *drill-down*. El *drill-down* es la habilidad de poder navegar de lo general a lo particular. Permite a partir de una información mostrada, sumergirse a los detalles. Por ejemplo, visualizando una información agrupada por años, puede navegarse para visualizar el detalle de la información por meses.

Dado que tenemos creado un cubo de datos, podemos generar una tabla dinámica Jpivot.

Tiempo	Ubicación	Nivel	Modulo	Sector	Medidas			
					• Abiertas	• Cerradas	• Derivadas	• Backlog
All Tiempos	All Ubicacions	All Nivels	All Modulos	All Sectors	48.115	45.899	5.598	143.740
2014	All Ubicacions	All Nivels	All Modulos	All Sectors	16.825	16.765	1.487	32.692
2015	All Ubicacions	All Nivels	All Modulos	All Sectors	31.290	29.134	4.111	111.048
January	All Ubicacions	All Nivels	All Modulos	All Sectors	3.419	2.917	422	8.858
February	All Ubicacions	All Nivels	All Modulos	All Sectors	3.463	3.345	419	12.559
	España	All Nivels	All Modulos	All Sectors	2.746	2.686	333	10.765
		HDn2	All Modulos	All Sectors	2.540	2.481	282	10.129
		HDn3	All Modulos	All Sectors	206	205	51	636
	Mexico	All Nivels	All Modulos	All Sectors	549	486	31	1.382
	Brasil	All Nivels	All Modulos	All Sectors	168	173	55	412
March	All Ubicacions	All Nivels	All Modulos	All Sectors	4.058	4.149	472	16.310

Ilustración 7 - Tabla Jpivot mostrando datos del Cubo OLAP

3.2.5 Plataforma web

El siguiente objetivo es disponer de una plataforma en la que centralizar toda la información e informes. De esta forma se puede garantizar una mayor disponibilidad de los datos y facilidad de acceso a los mismos. Además, se puede mejorar la seguridad con un control de acceso.

Para la creación de la plataforma vamos a utilizar Pentaho BI Server. Incluye toda la base web y servidor. Se puede personalizar mediante cambios en el código HTML o personalizar el estilo modificando los archivos CSS. Incluye módulos para poder añadir el generador de informes o poder visualizar la tabla dinámica Jpivot que hemos creado en las fases anteriores.

Se ha realizado algunos cambios en el código para personalizar la plataforma web y se ha configurado el control de acceso de los usuarios. Se han creado distintos perfiles para realizar pruebas y comprobar el acceso que tiene cada uno de ellos.

Con el servidor de la plataforma levantado, además de la base de datos. Desde el navegador web, se debe acceder a la IP del equipo en el que está arrancado el servidor (se podría configurar para establecer una URL). El puerto de acceso por defecto es el 8080.

Al acceder a la plataforma desde el navegador aparece la siguiente pantalla de acceso.

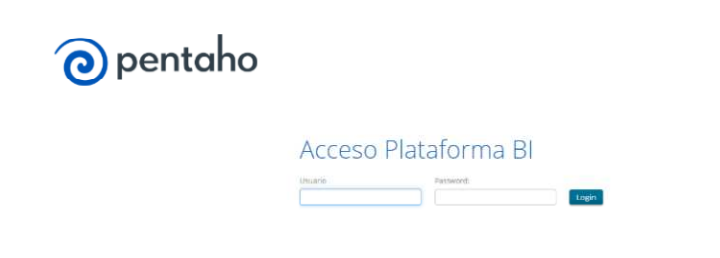


Ilustración 8 - Acceso Plataforma BI

Solicita introducir las credenciales para acceder a la plataforma. Introducidas las credenciales correctas accedemos a la plataforma BI.



Ilustración 9 - Página principal Plataforma BI

En la página principal se muestra un menú con las opciones principales. La visualización del usuario actual es de tipo administrador. Por lo tanto, tiene acceso a todas las funcionalidades.

Para visualizar el generador de informes y la tabla Jpivot debemos acceder a la opción del menú "Browse Files".

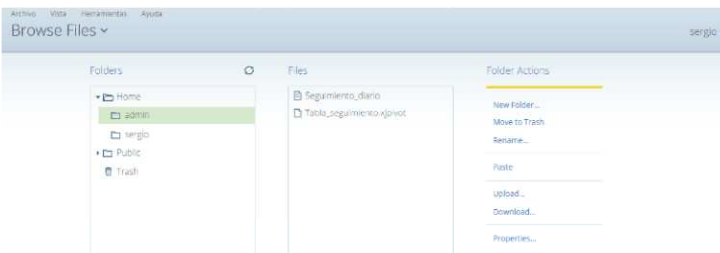


Ilustración 10 - Visualización de Informes y tablas dinámicas publicadas

Como podemos observar en la imagen, está publicado el informe de seguimiento diario y la tabla dinámica creados en las fases anteriores.

El contenido queda accesible de una manera sencilla y se puede organizar por carpetas de la forma deseada.

4 METODOLOGÍA

Para la realización del proyecto, he seguido la metodología de desarrollo de Software "Extreme Programming (XP)". Esta metodología ágil se diferencia de otras metodologías por su gran adaptabilidad.

Los cambios de requisitos sobre la marcha son inevitables en el desarrollo de proyectos. Ser capaz de adaptarse fácilmente a estos cambios es más realista que definir todos los requisitos al comienzo.

Se puede considerar esta metodología como la adopción de las mejores prácticas, aplicándolas de forma dinámica durante el proyecto.

Algunas de las características de esta metodología más destacables en el desarrollo del proyecto son:

- **Simplicidad:** Se simplifica el diseño para agilizar el desarrollo. Un diseño complejo del código hace que la complejidad aumente exponencialmente.
- **Retroalimentación:** La opinión del cliente, o quienes van a utilizar el programa, debe conocerse con frecuencia. Se deben realizar ciclos cortos, mostrar los resultados y obtener *feedback*.

5 PLANIFICACIÓN

Para la realización del proyecto, se planificaron las siguientes tareas.

	Nombre de tarea
1	Información y documentación inicial
2	▲ Extracción de datos a Base de datos
3	Análisis
4	Desarrollo
5	Pruebas / Verificación
6	▲ Generación Informes
7	▲ Informe automatico
8	Análisis
9	Desarrollo
10	Pruebas / Verificación
11	▲ Cubo de datos
12	Análisis
13	Desarrollo
14	Pruebas / Verificación
15	▲ Plataforma web
16	Análisis
17	Desarrollo
18	Pruebas / Verificación

Ilustración 13 - Tareas planificadas del desarrollo del proyecto

En cada una de las fases, se ha estimado un tiempo para cada parte del desarrollo.

Es decir, una estimación para el análisis, el desarrollo y las pruebas. Hay flexibilidad para moverse entre ellas siempre que el proyecto lo ha requerido.

Para visualizar la planificación de forma gráfica se puede observar el diagrama de *Gantt*.

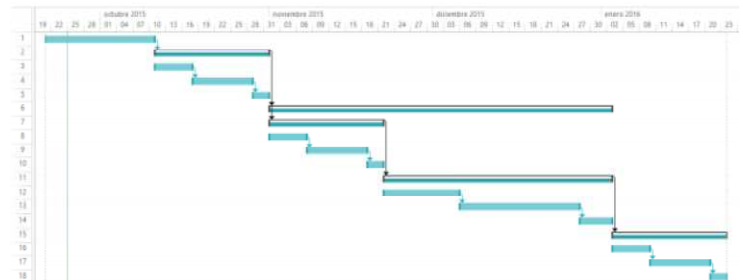


Ilustración 14 - Diagrama de Gantt del proyecto

6 LÍNEAS FUTURAS

Actualmente, la plataforma solo dispone de un informe de seguimiento diario y de una tabla JPivot que obtiene los datos del cubo OLAP. Se podría automatizar todos los informes que se realizan en el departamento o crear nuevos según las necesidades. De la misma forma, se podrían crear los cubos de datos necesarios para crear las tablas Jpivot que puedan aportar información útil.

La línea futura que más interesante puede resultar, es la creación de cuadros de mandos (*Dashboards*).

Un cuadro de mando te permite ver de forma gráfica y rápida la situación global de los datos que se están analizando.

Utilizan indicadores (*KPI*) acompañados de una representación gráfica para mostrar el estado actual.

Un cuadro de mandos sirve para optimizar el proceso de toma de decisiones estratégicas o tácticas.

Un ejemplo aplicado a la herramienta de gestión de incidencias, podría ser detectar que el valor de incidencias que se han generado durante el día, respecto a las incidencias cerradas es bajo. Detectar esto rápidamente, permite gestionar mejor el equipo y dedicar más recursos donde es necesario para solucionarlo.



Ilustración 15 - Ejemplo de Dashboard

7 VALORACIÓN DEL RESULTADO

El objetivo de este proyecto era mejorar la situación anterior solucionando todos los errores encontrados y aportar mejoras.

Con la plataforma web, se ha conseguido tener un lugar centralizado en el que tener toda la información. Esto evita la distribución de la información por correo electrónico. Además permite tener seguridad y privacidad sobre los datos al tener un control de accesos por credenciales.

Por otra parte, la automatización del informe ahorra recursos diariamente. El tiempo que se dedica a crear el informe manualmente puede ser invertido en otras tareas. Sin olvidar también el factor de error humano, al estar automatizado desaparece.

Como último punto, la estructuración de los datos en cubos OLAP permite crear tablas dinámicas JPivot. Estas permiten visualizar la información de manera distinta. Supone una mejora para obtener la información necesaria de una forma sencilla navegando por las dimensiones que se quieran consultar.

8 CONCLUSIONES

Con la valoración de los resultados previa, la primera conclusión que puedo extraer, es que estoy satisfecho con el trabajo realizado. Se han cumplido los objetivos propuestos, corrigiendo los errores existentes y aportando mejoras.

La siguiente conclusión, es que este proyecto puede aplicarse a casi cualquier departamento o empresa que no tenga un sistema o proceso para generar información o informes adecuado. Su desarrollo aporta valor para la toma de decisiones, además de reducir recursos invertidos.

Considero que puede aplicarse a casi cualquier sector empresarial, tanto en grandes empresas como en PYMES.

Como punto crítico, añadir que la plataforma tendría más utilidad con más tipos de informes automatizados y más tablas dinámicas. El tiempo invertido investigando y aprendiendo nuevos conceptos desconocidos hasta ahora me lo han impedido. Sin embargo, considero que los conocimientos adquiridos con la realización del proyecto son de gran valor y utilidad.

Para finalizar, mencionar que el desarrollo de este proyecto me ha animado a seguir investigando en soluciones y nuevas mejoras en la gestión de la información. Por ejemplo, como he mencionado en las líneas futuras del proyecto, los cuadros de mandos (*Dashboards*).

AGRADECIMIENTOS

Para concluir este artículo, quiero dar las gracias al tutor del proyecto Javier Sanchez Pujades por sus consejos y la libertad dada para realizar el proyecto con mis ideas.

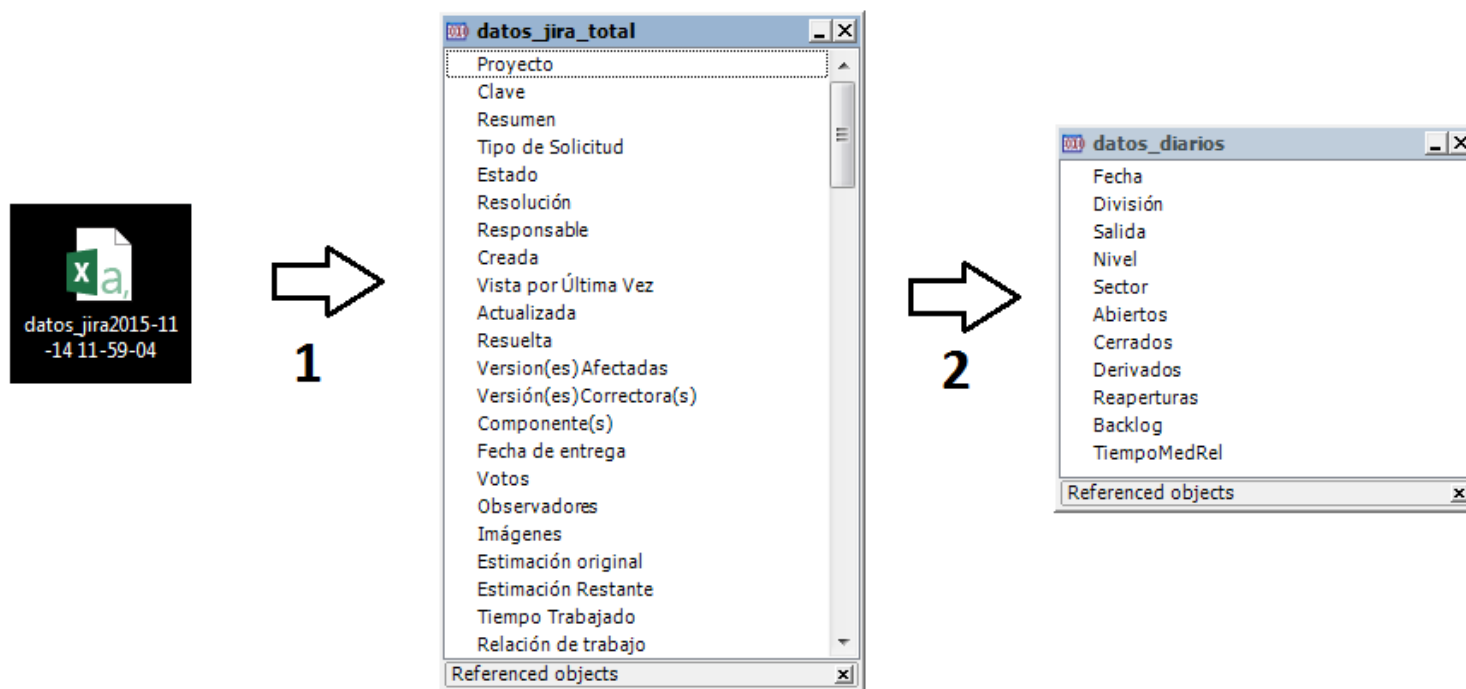
También dar las gracias a mi hermana por revisar los documentos y darme su opinión crítica para que pudiera mejorarlos.

BIBLIOGRAFÍA

- [1] Documentación Pentaho Data Integration
<http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation>
[Última consulta: 15/10/2015]
- [2] Documentación Pentaho Report Designer
<http://infocenter.pentaho.com/help/index.jsp>
[Última consulta: 20/12/2015]
- [3] MySQL Stored Procedures
<http://dev.mysql.com/doc/connector-net/en/connector-nettutorialsstored-procedures.html>
[Última consulta: 15/10/2015]
- [4] Documentación Spoon
<http://wiki.pentaho.com/display/EAI/Spoon+User+Guide>
[Última consulta: 15/10/2015]
- [5] Pentaho InfoCenter 5.0
<http://infocenter.pentaho.com/help/index.jsp>
[Última consulta: 20/12/2015]
- [6] Documentación Pentaho Schema Workbench
<http://mondrian.pentaho.com/documenttion/workbench.php>
[Última consulta: 20/12/2015]
- [7] Wikipedia - Definición cubo de datos
https://es.wikipedia.org/wiki/Cubo_OLAP
[Última consulta: 20/12/2015]
- [8] Modelo de base de datos en estrella contra modelo copo de nieve
<http://mundodb.es/disenio-data-warehouse-hechos-y-dimensiones-modelo-estrella-vs-copo-de-nieve>
[Última consulta: 20/12/2015]
- [9] Cubos OLAP y tablas Jpivot
<https://churriwifi.wordpress.com/2010/07/20/17-5-cubos-olap-y-navegacion-dimensional-con-mondrian-jpivot-y-stpivot/>
[Última consulta: 20/12/2015]
- [10] Documentación Pentaho BI Server
<http://wiki.pentaho.com/display/ServerDoc2x/BI+Platform+5.0>
[Última consulta: 08/01/2016]

ANEXO A

Transformación de datos para generación automática del informe

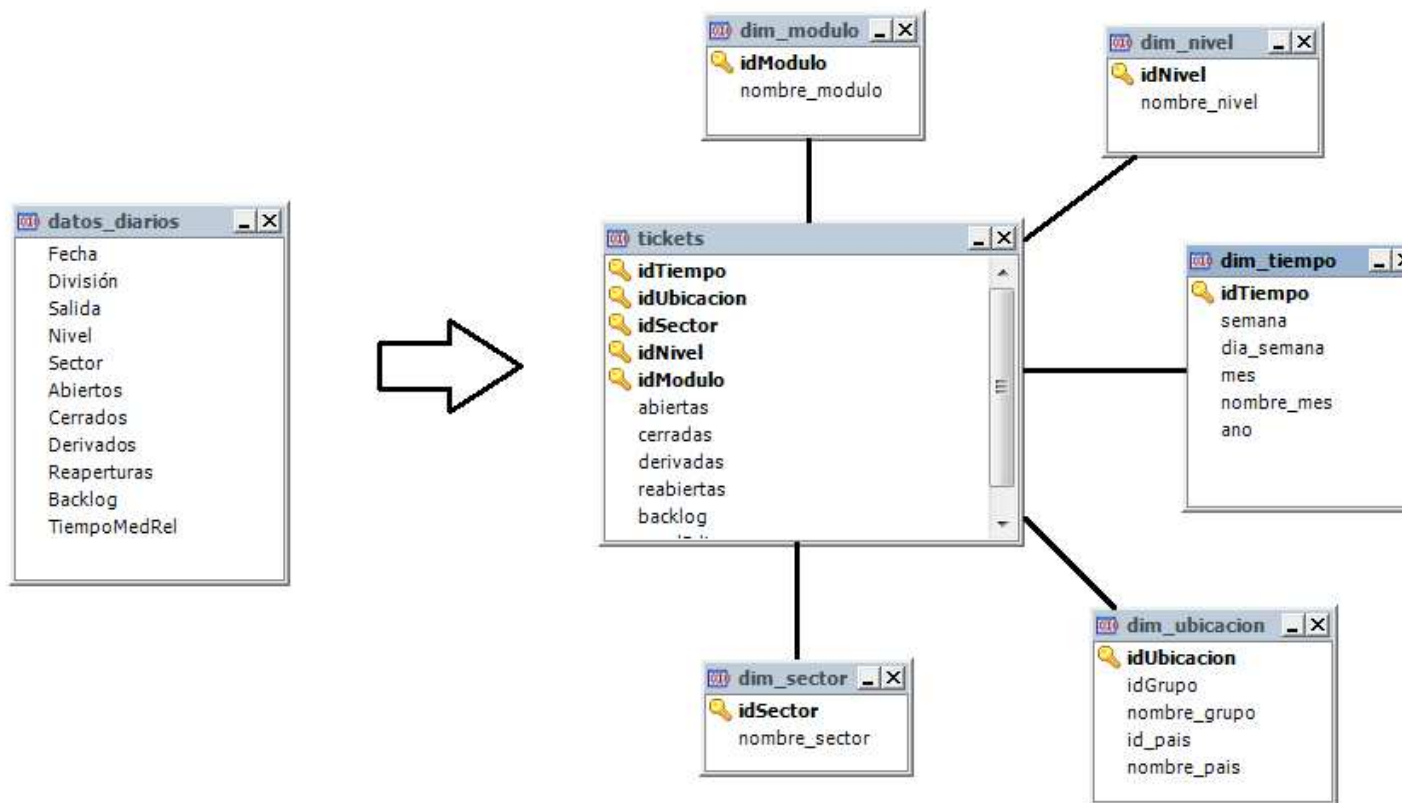


1 - El proceso creado con la herramienta “Pentaho Data Integration”, procesa el archivo Excel extraído de la aplicación, e inserta los valores en la tabla.

2 - Mediante un “Stored Procedure” se manipulan los datos, se calculan métricas y se agrupan los datos. Se insertan en la tabla “datos_diarios”.

ANEXO B

Transformación de tabla a modelo de base de datos tipo Estrella



El diseño de un cubo de datos OLAP requiere un modelo de base de datos tipo Estrella. Por eso se ha creado un “Stored Procedure” que inserta los valores en las tablas mediante el uso de “Foreign Keys” con identificadores.