

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Herramienta de Inteligencia de Negocio para la Gestión de Transporte Intermodal



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor: Samuel Nicuesa Carreras

Tutor: José Javier Astrain Escola

Pamplona, 23/03/2017

Agradecimientos

Me gustaría expresar mi agradecimiento a la Universidad Pública de Navarra por darme la oportunidad de realizar mis estudios y en especial al Departamento de Ingeniería Matemática e Informática.

También quiero agradecer a las personas que me han ayudado durante mi etapa en la Universidad que no ha sido fácil.

En primer lugar, agradecer a mi tutor de Trabajo Fin de Grado, José Javier Astrain Escola. Gracias por darme la oportunidad de realizar el trabajo contigo, por tus consejos, correcciones y por atender a todas mis preguntas e inquietudes.

También agradecer a Raúl Ecay Torres por ayudarme a resolver los problemas surgidos con el software utilizado.

Agradecer a mis compañeros de promoción por compartir conmigo estos años tan importantes de mi vida en los que he aprendido muchas cosas.

Por último, agradecer a mi padre Jacinto, que por desgracia no ha podido acompañarme hasta el final de mi etapa universitaria y a mi madre María por el esfuerzo económico que han tenido que hacer para ofrecerme la posibilidad de tener unos estudios.

Resumen

Diseño y Desarrollo de una herramienta informática de inteligencia de negocio (*Business Intelligence*, BI) que permite gestionar la información de transporte intermodal de un modo automatizado y dinámico para que las empresas e instituciones del sector puedan tomar decisiones que mejoren las infraestructuras e incrementen la productividad para permitir la evolución de los medios de transporte.

Los datos para la realización del trabajo son Open Data extraídos del Instituto Nacional de Estadística (INE) [1] y Datos del Gobierno de España [2]

Abstract

Design and Development of a Business Intelligence tool in order to manage transport information automatically and dynamically. This project will serve the companies and institutions to take good decisions in order to improve infrastructure and increase productivity.

This project contains a main Dashboard with images of different topics that we want to analyze. These images make reference to others Dashboards which will show graphics with information about the chosen topic.

The analysis of all this information will allow the evolution of transport making them more efficient, secure and better.

Lista de palabras clave

Big Data: Almacenamiento de grandes cantidades de datos y los procedimientos usados para encontrar patrones repetitivos dentro de esos datos.

Business Intelligence (BI): Se denomina inteligencia empresarial o inteligencia de negocios (BI) al conjunto de estrategias y aspectos relevantes enfocados a la administración y creación de conocimiento sobre el medio, a través del análisis de los datos existentes en una organización o empresa.

Dato: Los datos son la mínima unidad semántica, y se corresponden con elementos primarios de información que por sí solos son irrelevantes como apoyo a la toma de decisiones.

Información: La información se puede definir como un conjunto de datos procesados y que tienen un significado (relevancia, propósito y contexto), y que por lo tanto son de utilidad para quien debe tomar decisiones, al disminuir su incertidumbre.

Conocimiento: El conocimiento es una mezcla de experiencia, valores, información y know-how que sirve como marco para la incorporación de nuevas experiencias e información, y es útil para la acción. Se origina y aplica en la mente de los conocedores. En las organizaciones con frecuencia no sólo se encuentra dentro de documentos o almacenes de datos, sino que también está en rutinas organizativas, procesos, prácticas, y normas.

El conocimiento se deriva de la información, así como la información se deriva de los datos.[3]

Proceso ETL: *Extract, Transform and Load* es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, *data mart* o *data warehouse*, para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

OLTP: Es la sigla en inglés de Procesamiento de Transacciones En Línea (*OnLine Transaction Processing*) es un tipo de procesamiento que facilita y administra aplicaciones transaccionales, habitualmente para entrada de datos y recuperación y procesamiento de transacciones (gestor transaccional).

OLAP: Es el acrónimo en inglés de procesamiento analítico en línea (*On-Line Analytical Processing*). Es una solución utilizada en el campo de BI cuyo objetivo es agilizar la consulta de grandes cantidades de datos.

Cubo OLAP: Los cubos OLAP son estructuras multidimensionales (cubos) que permiten analizar bases de datos relacionales de gran volumen y variedad con una gran agilidad y rapidez, reduciendo enormemente el tiempo y los recursos empleados en el análisis. Existen distintos sistemas OLAP diferenciados entre sí, básicamente, por los tipos de bases de datos sobre los que se construyen y que dan lugar, entre otras y principalmente, a las siguientes categorías o sistemas:

- Sistemas **ROLAP** (Procesamiento Analítico Relacional en Línea).
- Sistemas **MOLAP** (Procesamiento Analítico Multidimensional en Línea).
- Sistemas **HOLAP** (Procesamiento Analítico Híbrido en Línea).

Datawarehouse (DW): Es una colección de datos (almacén de datos) orientada a un determinado ámbito (empresa, organización, etc.) integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

Dashboard (Cuadros de mando Integral): Herramienta de **gestión empresarial** muy útil para medir la evolución de la actividad de una compañía y sus resultados, desde un punto de vista estratégico y con una perspectiva general. La suelen utilizar los gerentes y altos cargos de las empresas por su valor, al contribuir de forma eficaz en la visión empresarial a medio y largo plazo.

Índice de contenidos

1. INTRODUCCIÓN	9
1.1. JUSTIFICACIÓN, CONTEXTO Y OBJETIVO DEL TRABAJO	9
1.2. ¿QUÉ ES BUSINESS INTELLIGENCE?	10
1.3. TECNOLOGÍA UTILIZADA	12
1.4. METODOLOGÍA ITERATIVA.....	14
1.5. APARTADOS DE LA MEMORIA	15
2. CONFIGURACIÓN DE PENTAHO COMMUNITY	16
2.1. CONEXIÓN A PROXMOX	16
2.2. JAVA	17
2.3. VARIABLE DE ENTORNO.....	18
2.4. PENTAHO 6.1	18
2.5. MYSQL.....	19
2.6. DATA SOURCES	21
2.7. AÑADIENDO COMPONENTES.....	24
3. PROCESO ETL (EXTRACT, TRANSFORM AND LOAD)	26
3.1. HERRAMIENTA DATA INTEGRATION.....	26
3.2. NUEVA TRANSFORMACIÓN.....	27
3.3. ENTRADA EXCEL	27
3.4. TRANSFORMACIÓN	29
3.5. SALIDA A TABLA	30
3.6. CARGA DE DATOS.....	33
3.7. OPCIÓN JOB	34
4. CREACIÓN DEL CUBO OLAP	38
4.1. CONFIGURACIÓN DE SCHEMA-WORKBENCH	38
4.2. CREACIÓN DE CUBO OLAP	40
5. CREACIÓN DE DASHBOARD	46
5.1. USO DE SAIKU	46
5.2. CREACIÓN DE DASHBOARD.....	49
5.3. EJEMPLOS DE DASHBOARDS	52
6. ELABORACIÓN DE REPORTES USANDO LA HERRAMIENTA PENTAHO REPORT DESIGNER.....	56
7. CONCLUSIONES Y REFLEXIONES PROPIAS.....	62
8. BIBLIOGRAFÍA	63

Listado de ilustraciones

Imagen 1 proceso bi.....	10
Imagen 2 conexión mediante putty	16
Imagen 3 conexión establecida con putty	16
Imagen 4 conexión mediante cliente x2go	17
Imagen 5 conexión mediante linux	17
Imagen 6 versión java instalada.....	17
Imagen 7 pentaho_java_home.....	18
Imagen 8 fichero start pentaho.....	18
Imagen 9 ejecutable pentaho	18
Imagen 10 arrancando pentaho	19
Imagen 11 pantalla inicio pentaho.....	19
Imagen 12 instalando mysql.....	20
Imagen 13 phpmyadmin	20
Imagen 14 roles usuario pentaho.....	21
Imagen 15 manage data sources.....	21
Imagen 16 new data source	22
Imagen 17 new data source	22
Imagen 18 credenciales mysql.....	23
Imagen 19 test data source	23
Imagen 20 conexión correcta data source	24
Imagen 21 vista data source	24
Imagen 22 marketplace	25
Imagen 23 saiku analytics	25
Imagen 24 instalación de saiku	25
Imagen 25 spoon.....	26
Imagen 26 inicio data integration	26
Imagen 27 nueva transformación.....	27
Imagen 28 entrada excel	27
Imagen 29 input entrada excel	28
Imagen 30 hojas entrada excel	28
Imagen 31 campos entrada excel	28
Imagen 32 transformación.....	29
Imagen 33 uso procedure	29
Imagen 34 procedure	30
Imagen 35 salida a tabla	30
Imagen 36 configuración salida a tabla.....	31
Imagen 37 conexión con bd salida a tabla	31
Imagen 38 error test conexión bd.....	32
Imagen 39 test conexión correcta	32
Imagen 40 create table data integration.....	32
Imagen 41 vista etl	33
Imagen 42 carga datos en bd.....	33
Imagen 43 carga de datos correcta.....	34
Imagen 44 creación job	34
Imagen 45 vista del job	35
Imagen 46 script borra tablas.....	35
Imagen 47 script crea tablas	36
Imagen 48 scheduling job	36
Imagen 49 ejecución job	37
Imagen 50 ejecución schema workbench	38
Imagen 51 conexión con bd en schema workbench	38
Imagen 52 error test conexión con bd en schema workbench	39
Imagen 53 test correcto.....	39
Imagen 54 new schema	40

Imagen 55 nombre schema.....	40
Imagen 56 add cube.....	40
Imagen 57 vista tabla hechos.....	41
Imagen 58 añadir dimensión	41
Imagen 59 añadir tabla de dimensión	42
Imagen 60 añadir nivel	42
Imagen 61 elementos de la jerarquía.....	42
Imagen 62 elementos del nivel.....	43
Imagen 63 dimensión usage	43
Imagen 64 añadir medida.....	43
Imagen 65 vista de los cubos.....	44
Imagen 66 publicar cubos	44
Imagen 67 credenciales pentaho	45
Imagen 68 publicación correcta	45
Imagen 69 acceso a saiku.....	46
Imagen 70 create new query.....	46
Imagen 71 vista de cubos en saiku	47
Imagen 72 uso del cubo en saiku.....	47
Imagen 73 tabla de datos extraída del cubo en saiku	47
Imagen 74 gráfico en saiku	48
Imagen 75 query mdx.....	48
Imagen 76 new dashboard	49
Imagen 77 opciones configuración dashboard	49
Imagen 78 query pentaho	50
Imagen 79 editor mdx.....	50
Imagen 80 gráfico en pentaho.....	51
Imagen 81 dashboard index.....	52
Imagen 82 dashboard transporte aéreo	53
Imagen 83 dashboard transporte carretera.....	54
Imagen 84 dashboard contaminación y consumo de energía	55
Imagen 85 report designer	56
Imagen 86 plantilla report designer	56
Imagen 87 data source.....	57
Imagen 88 conexión a nuestra bd	57
Imagen 89 conexión correcta	57
Imagen 90 creación de consultas.....	58
Imagen 91 uso de parámetros	58
Imagen 92 ejemplo de informe.....	59
Imagen 93 ejemplo de informe.....	59
Imagen 94 exportar informe	60
Imagen 95 informe con gráfica.....	60
Imagen 96 informe con gráfica.....	61

1. Introducción

1.1. Justificación, Contexto y Objetivo del Trabajo

Este trabajo surge de la inquietud por aprender qué es *Business Intelligence* (BI) y sus aplicaciones en la vida real. En distintas asignaturas de la carrera han aparecido los términos *Business Intelligence* y *Big Data* pero sin entrar en mucho detalle por lo que quería aprender más en profundidad debido a popularidad que están teniendo en estos años.

Lo que más me llamó la atención de estas herramientas de BI es el gran uso que tienen en las empresas ya que permiten transformar datos en información útil para la toma de decisiones y se pueden usar con datos de cualquier ámbito.

Un tema que me pareció útil para aplicar BI fue la medicina, debido a que se puede ver la respuesta que tienen los pacientes de diferentes enfermedades, como cáncer, alzheimer o parkinson, a distintos tratamientos con el fin de poder elegir el tratamiento más adecuado.

Este es un problema de la vida real, el caso es que la información no es fácilmente accesible, por lo que decidí utilizar datos abiertos (*open data*) de otro ámbito, en mi caso información de transporte intermodal para realizar un sistema de BI.

En este proyecto se va a aplicar *Business Intelligence* para gestionar la información de transporte intermodal de un modo automatizado y dinámico. Todo esto permitirá a las instituciones y empresas del sector de transportes analizar datos y tomar decisiones adecuadas.

Dependiendo de los informes se podrán mejorar infraestructuras, incrementar la productividad del sector, invertir en seguridad con el fin de mejorar la eficiencia y permitir la evolución de los medios de transportes.

Para la realización del proyecto se ha utilizado la herramienta de *Business Intelligence* Pentaho Community [4] que es gratuita y tiene todas las funcionalidades necesarias para poder llevar a cabo el estudio. La base de datos de la cual se van a extraer los datos es MySQL [5] y los datos con los que se ha poblado son los datos de transporte de la página web del INE y datos del gobierno de España.

1.2. ¿Qué es Business Intelligence?

BI es un conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

Las herramientas de BI permiten a una empresa reunir, depurar y transformar los datos obtenidos de operaciones diarias como pedidos, facturas, albaranes en información estructurada lista para su explotación directa mediante análisis y conversión en conocimiento que sirva para el soporte a la toma de decisiones de negocio.

Para reunir, depurar y transformar datos se utilizan herramientas ETL, los datos adquiridos anteriormente se almacenarán en el DW. La Imagen 1 muestra las fases del proceso BI.

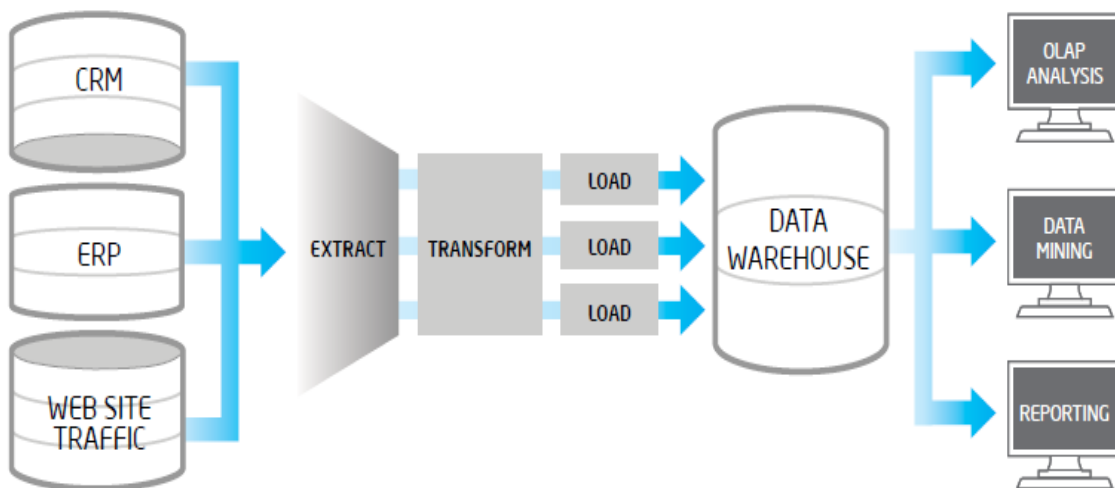


IMAGEN 1 PROCESO BI

Existen distintas herramientas de BI en la actualidad, tanto de software libre como de software de pago. A continuación, se citan algunas de ellas.

Software de pago:

- Microsoft SQL Server BI [6]
 - Herramienta Microsoft para Bi que incluye:
 - Integration Services.
 - Analysis Services.
 - Reporting Services.

- Tableau Business Intelligence [7]
 - Tableau Server ofrece permisos de seguridad en cualquier nivel que necesite el usuario.
 - Permite la gestión y el análisis de datos sin elevados conocimientos técnicos.

- SAP Lumira [8]
 - SAP Lumira para SAP Business One se instala en cualquier equipo o servidor y no requiere ningún hardware adicional.

- Oracle BI [9]
 - Características:
 - Cuadros de mandos interactivos
 - Informes interactivos individuales (auto-servicio)
 - Detección proactiva y alertas
 - Inteligencia de acción.
 - Integración con Office
 - Inteligencia Espacial vía visualizaciones basadas en mapas.

- Pentaho Enterprise Edition [10]
 - Posee herramientas como Kettle y Mondrian al igual que Pentaho Community Edition, pero al tratarse de un software de pago añade herramientas de gestión de carga de trabajo y planificación de tareas además de funciones de administración y seguridad como contraseñas AES.

Software Libre:

- Pentaho Community Edition [11]
 - Pentaho ha sido creado bajo las plataformas libres de Java y MySQL.
 - Ambas plataformas están en auge y cada vez más gente hace uso de ellas. Son fáciles de usar, compilar y programar.

- JasperReports [12]
 - Es una herramienta de creación de informes.
 - Está escrita completamente en java.

- Palo [13]
 - Utiliza una base de datos multidimensional.
 - Además de la versión para MS Excel, también hay una versión para *OpenOffice*.

- SpagoBI [14]
 - Es una multiplataforma integrada para BI de código abierto. Satisface todos los requisitos de BI, tanto en términos de análisis y de gestión de datos, administración y seguridad.

1.3. Tecnología utilizada

Para realizar este trabajo se ha utilizado Proxmox Virtual Environment, o Proxmox VE [15], que es un entorno de virtualización de servidor de código abierto. Proxmox se encuentra en distribuciones de Linux basadas en Debian con una versión modificada del Kernel RHEL y permite el despliegue y la gestión de máquinas virtuales y contenedores.

El proyecto se ha realizado en un Linux Ubuntu versión 14.04 LTS situado en un contenedor de Proxmox con memoria RAM 8 GB y tamaño de disco 20GB. El contenedor genera copias de seguridad automáticas todos los días.

La base de datos donde se almacenará la información sobre trasportes extraída de las fuentes INE y Datos del Gobierno de España es MySQL. Se ha elegido MySQL por disponer de experiencia previa en su manejo por su utilización en otras asignaturas del grado. Para gestionarla se ha utilizado phpMyAdmin [16] que es una herramienta escrita en lenguaje PHP para administrar MySQL a través de páginas web utilizando internet, y MySQL Workbench [17] que también es otra herramienta para administrar, diseñar y mantener sistemas de base de datos MySQL.

La herramienta de BI utilizada es Pentaho Community versión 6.1, que como se ha visto anteriormente, se trata de una herramienta de Software Libre que no tiene ningún tipo de coste. Además de eso ofrece una funcionalidad suficientemente potente como para realizar todo lo necesario para este proyecto.

Pentaho Community es, dentro de las herramientas de software libre, la más extendida y utilizada, lo que aporta una ventaja a la hora de documentarse, aprender y solucionar problemas sobre la misma.

La herramienta de Business Intelligence Pentaho está desarrollada en Java [18] y se ejecuta en un servidor Tomcat [19], lo cual la hace muy accesible puesto que es posible utilizarla desde cualquier dispositivo y desde cualquier sistema operativo. Esto es realmente importante porque las personas pueden utilizar el sistema operativo que quieran.

Pentaho Community es un conjunto de programas libres para generar inteligencia empresarial BI. Incluye herramientas integradas para generar informes, minería de datos, ETL... a partir de cualquier tipo de información.

Existen otras tecnologías que se han utilizado para el desarrollo del proyecto, a continuación, se muestran cuales son:

- **Tomcat8:** Es un servidor web que soporta la tecnología J2EE de Java, es gratuito y puede ser utilizado en cualquier sistema operativo, con su máquina virtual java correspondiente. Los encargados de su desarrollo y mantenimiento son los miembros de la *Apache Software Foundation*[20].
- **Java SE Development Kit 8:** Entorno de desarrollo para crear aplicaciones, applets y componentes utilizando el lenguaje de programación Java.

- **Pentaho Data Integration (PDI):** Herramienta que permite implementar los procesos de extracción, transformación y carga de datos. Con esta herramienta se construyen las transformaciones (mínimo nivel de diseño) utilizando los pasos (*steps*). En un nivel superior se encuentran los trabajos (*Jobs*), que permiten ejecutar las transformaciones y otros componentes, además de orquestrar los procesos.
- **Saiku:** Herramienta OLAP destinada a usuarios finales de Pentaho, que permite visualizar y realizar análisis de datos de forma fácil e intuitiva. Es una mejora de la interfaz gráfica del portal web, que reemplazará Jpivot, mediante la cual se pueden construir vistas propias arrastrando y soltando campos.
- **Marketplace:** Lugar donde se encuentran todos los *plugins* disponibles, los cuales se pueden instalar y probar.
- **Schema Workbench:** Permite crear visualmente cubos OLAP.
- **Report Designer:** Herramienta de *reporting* fácil de utilizar y con multitud de aplicaciones. Los informes que genera se dividen en secciones o grupos de datos en los que los elementos del informe pueden ser posicionados. Esta forma de trabajar tiene algunas limitaciones, que se pueden superar con el uso de subinformes. Nos permite la salida de resultados en diferentes formatos como PDF, HTML, XLS, RTF y CSV.

1.4. Metodología Iterativa

Para la realización de este proyecto se ha utilizado una metodología ágil de desarrollo iterativo e incremental. Este proyecto se ha realizado en 3 iteraciones con tareas claramente diferenciadas.

En cada iteración se evoluciona el producto haciendo una entrega incremental, y al completar todas las iteraciones se proporciona un resultado completo sobre el producto final.

Este tipo de metodología ha sido muy útil en el desarrollo de este proyecto debido a que los resultados no se visualizan hasta la última iteración que es la creación de Dashboards. Por ello como la principal prioridad es satisfacer al cliente con entregas tempranas (resultados) he intentado terminar todas las iteraciones de manera rápida y simple para poder comprobar que soy capaz de realizar el proceso iterativo correctamente y ver el producto final correctamente, en este caso una gráfica.

Una vez hecho esto se volverán a realizadas todas las iteraciones otra vez, profundizando más en las tareas de cada iteración con el objetivo de mejorar el resultado final.

A continuación, se muestran las tareas correspondientes a las iteraciones:

- **Tarea 0:** Elección del Software utilizado.
- **Tarea 1:** Elección de los Datos a Tratar.
- **Tarea 2:** Familiarización con las herramientas.
- **Tarea 3:** Proceso ETL.
- **Tarea 4:** Creación de Cubos.
- **Tarea 5:** Realización de Dashboards e Informes.

Antes de empezar cada iteración era necesario aprender las distintas herramientas involucradas para que los entregables no se demoraran mucho en el tiempo. En mi caso, intentaba que cada dos semanas como mucho tuviese mejoras cuantitativas.

En cuanto a la familiarización con las herramientas, cabe destacar que me han sido de gran ayuda los ejemplos que vienen resueltos con cada una de las herramientas de Pentaho así como vídeos e información de distintos foros relacionados con el tema.

1.5. Apartados de la memoria

En el **capítulo 1** (Introducción) se describe el proyecto, así como la tecnología y metodología utilizada y las diferentes herramientas que serán necesarias durante su desarrollo.

En el **capítulo 2** (Configuración de Pentaho Community) se describen los pasos necesarios a seguir para configurar y usar correctamente Pentaho.

En el **capítulo 3** (Proceso ETL) se describe el proceso a seguir para extraer los datos almacenados en ficheros Excel y cargarlos en nuestro DW. Este es un paso muy importante ya que los datos generados se utilizarán en las iteraciones siguientes.

En el **capítulo 4** (Creación del Cubo OLAP) se describen los pasos a seguir para la correcta formación de los cubos OLAP, indicando en cada uno de ellos la tabla de hechos correspondiente, las dimensiones y medidas necesarias para obtener la información que se mostrará en la iteración 5, necesaria para la toma de decisiones.

En el **capítulo 5** (Creación de Dashboards) se describe el proceso para generar cuadros de mando, este también es un paso importante porque se reflejará el contenido obtenido de los pasos anteriores. Los Dashboards deben ser claros ya que se utilizarán para el proceso de toma de decisiones.

2. Configuración de Pentaho Community

En este capítulo se va a explicar cómo se configura correctamente Pentaho para su correcto funcionamiento.

Para este proyecto se ha decidido utilizar el contenedor Proxmox donde tendremos una máquina Linux con Pentaho instalado. Para una mejor interacción de nuestro equipo Windows con Proxmox es conveniente conectarse de alguna de las maneras que se explican a continuación.

2.1. Conexión a Proxmox

Debido a que la consola de Proxmox no es muy amigable, es conveniente acceder a la máquina de otra forma. A continuación se explicarán distintos modos de conexión para hacer más fácil el uso de la máquina.

El primer método para conectarse a la máquina desde un usuario Windows es usar el Programa Putty [21] (cliente ssh, Telnet, rlogin, y TCP raw, con licencia libre). Para ello necesitamos indicarle la dirección IP y el puerto para establecer la conexión ssh. La Imagen 2 muestra la configuración de la conexión con la aplicación Putty.

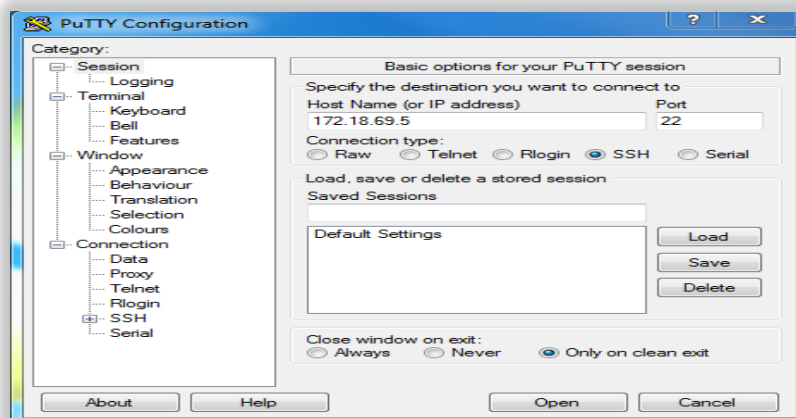


IMAGEN 2 CONEXIÓN MEDIANTE PUTTY

Una vez conectado a la máquina, podremos ejecutar comandos mediante el empleo de la consola (ver la Imagen 3).

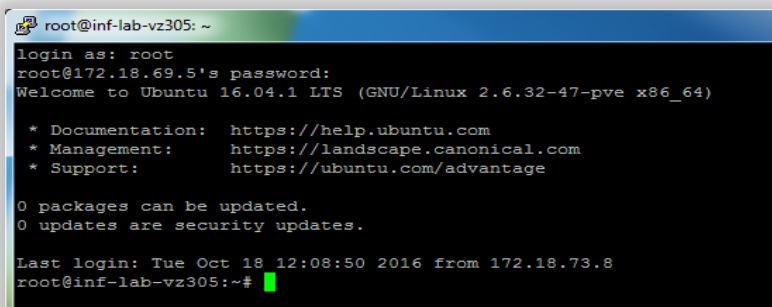


IMAGEN 3 CONEXIÓN ESTABLECIDA CON PUTTY

El segundo método de conexión es utilizar una gui xfce con el cliente X2go [22], que es un software que nos permite tener un acceso remoto gráfico vía ssh (ver Imagen 4).

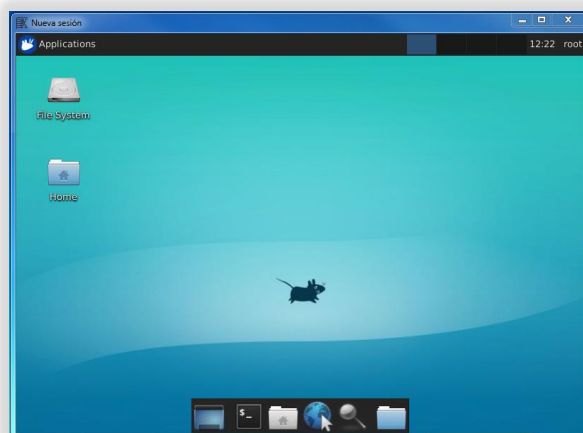


IMAGEN 4 CONEXIÓN MEDIANTE CLIENTE X2GO

El tercer método de conexión es conectarnos mediante una máquina Linux utilizando conexión ssh. La conexión se realizará de la siguiente manera: **ssh -X root@ip** como se indica en la Ilustración 5.

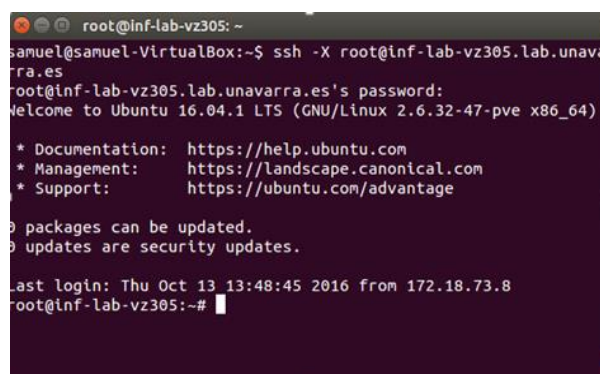


IMAGEN 5 CONEXION MEDIANTE LINUX

2.2. Java

Una vez que nos conectamos a Proxmox con alguno de los métodos anteriores estamos manejando la máquina Linux que está instalada en el contenedor.

Podemos navegar mediante consola, en este caso, por la máquina Linux y vemos que en la ruta `/usr/lib/jvm/java-8-oracle` está instalado JAVA 8.

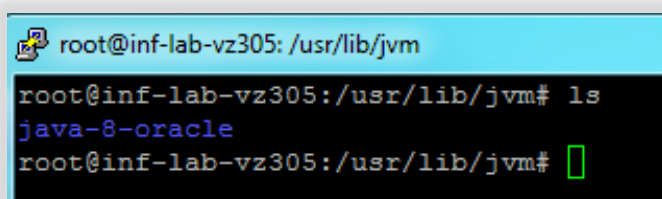


IMAGEN 6 VERSION JAVA INSTALADA

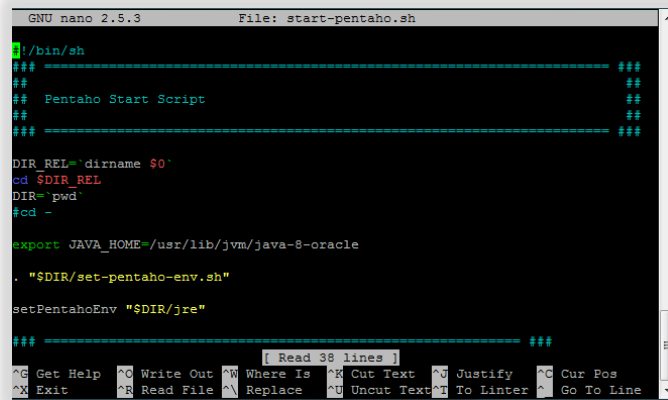
2.3. Variable de entorno

Tenemos una variable de entorno llamada **PENTAHO_JAVA_HOME** que apunta a la dirección donde está instalado JAVA, que es **/usr/lib/jvm/java-8-oracle**. Esta variable es necesaria para arrancar Pentaho, puesto que es necesario el intérprete de java (ver Imagen 7).

```
root@inf-lab-vz305:/opt/biserver-ce# $PENTAHO_JAVA_HOME
-bash: /usr/lib/jvm/java-8-oracle: Is a directory
root@inf-lab-vz305:/opt/biserver-ce#
```

IMAGEN 7 PENTAHO_JAVA_HOME

Dentro del fichero **start-pentaho.sh** localizado en **opt/biserver-ce** se añadirá la línea **export JAVA_HOME=/usr/lib/jvm/java-8-oracle** para que Pentaho arranque utilizando java (ver Imagen 8).



```
GNU nano 2.5.3 File: start-pentaho.sh
#!/bin/sh
### =====
## Pentaho Start Script
## =====
DIR_REL=$(dirname $0)
cd $DIR_REL
DIR=$(pwd)
#cd -

export JAVA_HOME=/usr/lib/jvm/java-8-oracle

. "$DIR/set-pentaho-env.sh"

setPentahoEnv "$DIR/jre"
### =====
```

IMAGEN 8 FICHERO START PENTAHO

2.4. Pentaho 6.1

Una vez realizados todos estos pasos, podemos proceder a iniciar Pentaho BI. Para arrancarlo se debe ejecutar el archivo **start-pentaho.sh** antes mencionado, que se encuentra en la maquina Linux instalada en el servidor Proxmox en la ruta **/opt/biserver-ce** (ver Imagen 9).

```
root@inf-lab-vz305:/opt/biserver-ce# ls
data                set-pentaho-env.bat    stop-pentaho.bat
import-export.bat   set-pentaho-env.sh     stop-pentaho.sh
import-export.sh     start-pentaho-debug.bat third-party-tools
licenses            start-pentaho-debug.sh tomcat
pentaho-solutions  start-pentaho.bat
promptuser.js       start-pentaho.sh
```

IMAGEN 9 EJECUTABLE PENTAHO

Al ejecutar ese archivo **./start-pentaho.sh** se iniciará el servidor Tomcat sobre el que se aloja Pentaho, y para acceder introduciremos en el navegador la url **“inf-lab-vz305.lab.unavarra.es:8080/pentaho/Login”** (ver Imagen 10).

```
root@inf-lab-vz305:/opt/biserver-ce# ./start-pentaho.sh
DEBUG: Using PENTAHO_JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/lib/jvm/java-8-oracle
DEBUG: _PENTAHO_JAVA=/usr/lib/jvm/java-8-oracle/bin/java
Using CATALINA_BASE: /opt/biserver-ce/tomcat
Using CATALINA_HOME: /opt/biserver-ce/tomcat
Using CATALINA_TMPDIR: /opt/biserver-ce/tomcat/temp
Using JRE_HOME: /usr/lib/jvm/java-8-oracle
Using CLASSPATH: /opt/biserver-ce/tomcat/bin/bootstrap.jar:/opt/biserver-ce/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

IMAGEN 10 ARRANCANDO PENTAHO

La primera vez que entremos a Pentaho utilizaremos el rol de administrador que trae por defecto Pentaho. Para poder acceder hay que introducir como **User Name:** Admin y como **Password:** password. Para evitar problemas de seguridad, crearemos un usuario para que no sea superusuario y cambiaremos la contraseña de administrador.

Introduciendo en la url: **http://inf-lab-vz305.lab.unavarra.es:8080/** aparece la ventana principal de Pentaho que se observa en la Imagen 11.



IMAGEN 11 PANTALLA INICIO PENTAHO

2.5. MySQL

Para alimentar de datos a Pentaho tenemos una base de datos llamada **transportes** en la que mediante la herramienta de ETL *data-integration*, como se verá más adelante, se creará la estructura de tablas necesaria en formato estrella con datos recogidos de los ficheros Excel de las fuentes INE y Datos del Gobierno de España.

En este caso se va a utilizar el sistema gestor de bases de datos MySQL. Para instalar MySQL lo que debemos hacer es teclear el siguiente comando **apt-get update && apt-get upgrade** para actualizar el repositorio de paquetes de la distribución de Linux instalada y **apt-get install mysql-server mysql-common mysql-client** para instalar el cliente y el servidor de MySQL.

La Imagen 12 muestra este proceso.

```
root@inf-lab-vz305:~# apt-get install mysql-server mysql-common mysql-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

IMAGEN 12 INSTALANDO MYSQL

Una vez que tenemos MySQL instalado vamos a instalar phpmyadmin para tener un entorno gráfico donde gestionar la base de datos mediante el comando **sudo apt-get install phpmyadmin**.

Para acceder a phpMyAdmin debemos poner en el navegador la dirección IP de la máquina Linux de nuestro contenedor seguido de phpmyadmin como indica la Imagen 13. Las credenciales a utilizar son las siguientes:

user = root y password = password

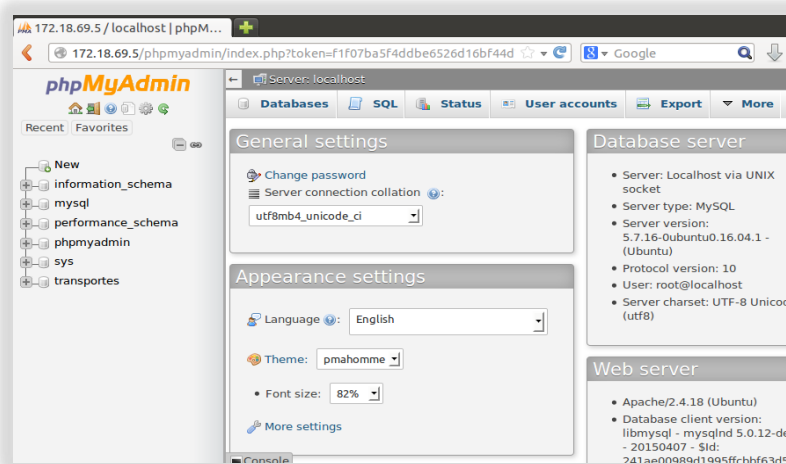


IMAGEN 13 PHPMYADMIN

El siguiente paso es acceder a Pentaho en **/opt/biserver-ce** y arrancar con el siguiente comando: **./start-pentaho.sh**.

Una vez dentro, nos *loguearemos* con las claves antes mencionadas: **User Name:** Admin y como **Password:** password.

Lo siguiente que tenemos que hacer es crear un usuario para no entrar siempre con el usuario administrador.

Las Imagen 14 muestra el acceso y la gestión de un usuario en Pentaho.

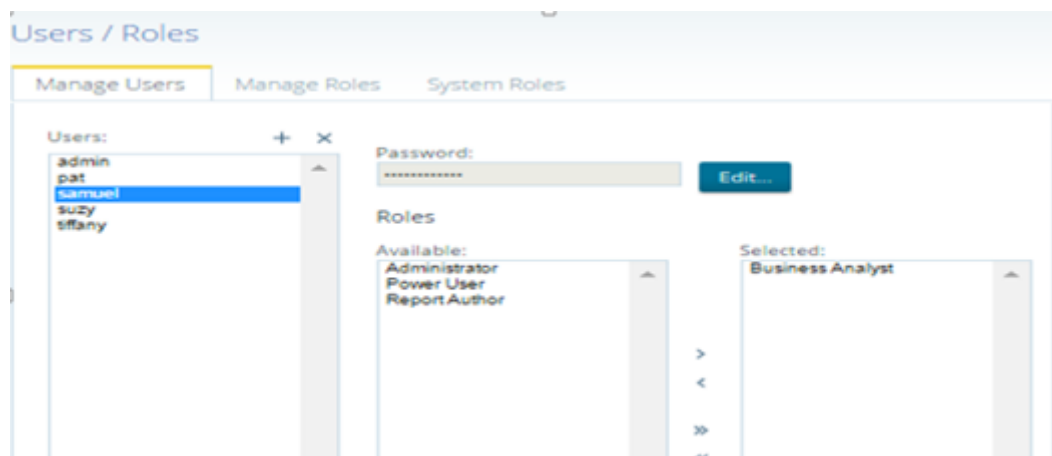


IMAGEN 14 ROLES USUARIO PENTAHO

2.6. Data Sources

Una vez dentro de Pentaho debemos asignarle un origen de datos, del cual extraerá los datos que posteriormente se utilizarán para crear el dashboard.

Para ello debemos ir a la pestaña **File** y pulsar en **Manage Data Sources** (ver Imagen 15).

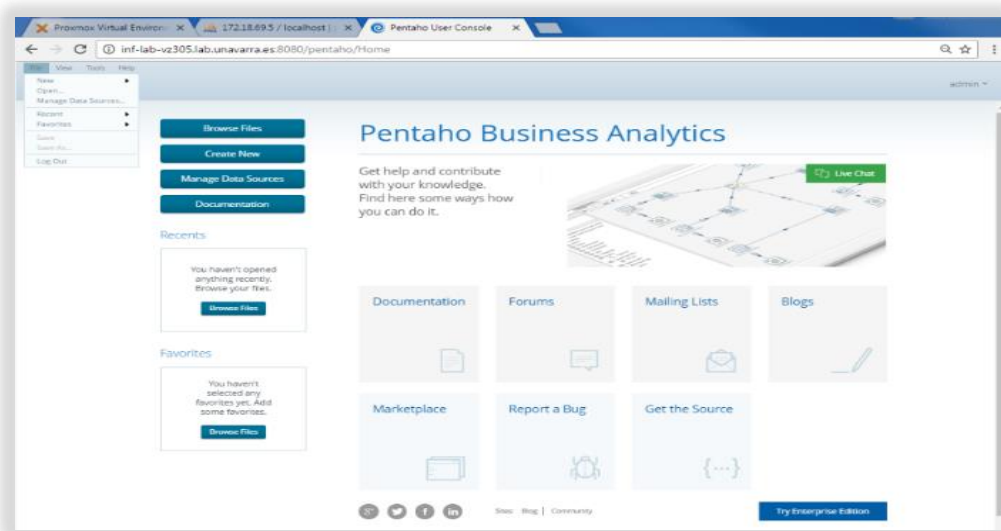


IMAGEN 15 MANAGE DATA SOURCES

En esa ventana hay que crear un **New Data Source** (ver Imagen 16) que será nuestra base de datos anteriormente creada.

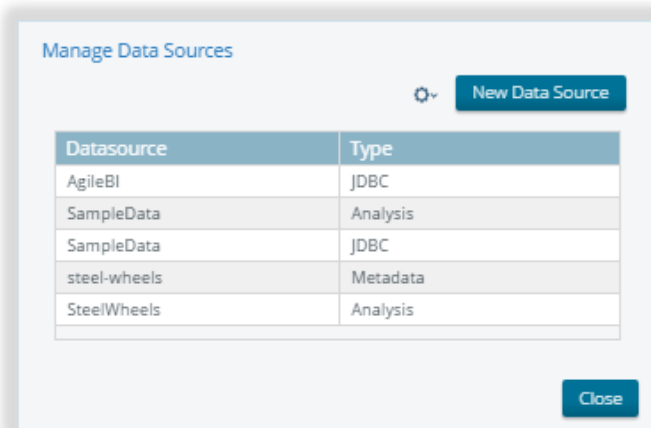


IMAGEN 16 NEW DATA SOURCE

Seleccionamos como *Source* -> **Type Database Table(s)** y pulsamos en + para añadir la conexión tal y como se aprecia en la Imagen 17.

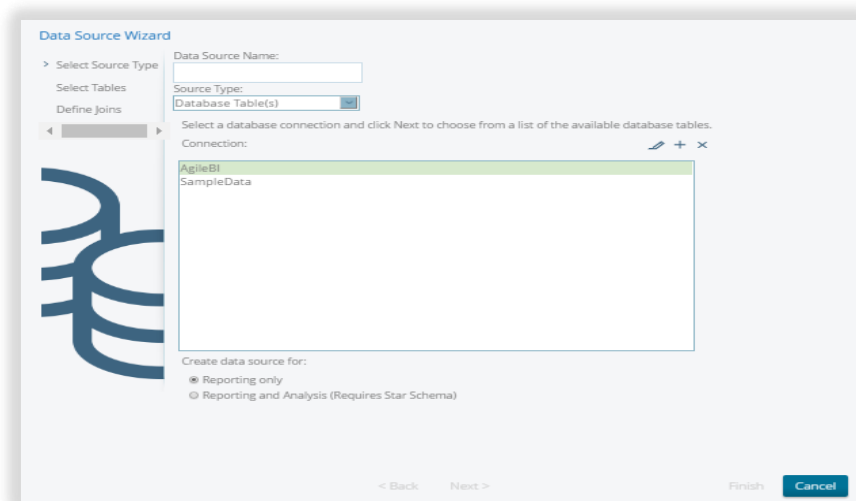


IMAGEN 17 NEW DATA SOURCE

El siguiente paso consiste en indicar las credenciales de nuestra base de datos para que la conexión sea satisfactoria tal y como se muestra en la Imagen 18. El nombre de la conexión será **transportes**, tipo de BBDD **MySQL**, acceso **JDBC**, hostame **localhost**, Nombre de base de datos **transportes**, puerto **3306** (por defecto el de MySQL), user **root** y contraseña **password**.

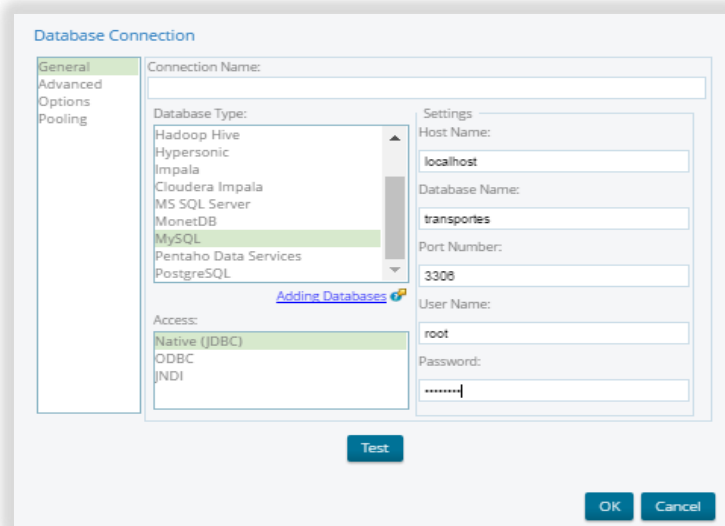


IMAGEN 18 CREDENCIALES MYSQL

Una vez completados todos los campos se pulsará la opción test para comprobar el estado de la conexión. Si ésta ha sido satisfactoria aparecerá el mensaje de la Imagen 19.

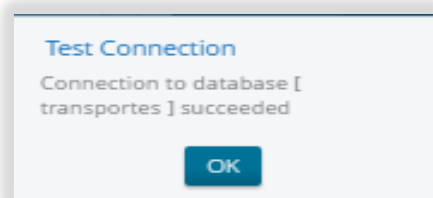


IMAGEN 19 TEST DATA SOURCE

Se podrá comprobar en la ventana de la Imagen 20 que aparece la conexión que acabamos de crear y ya podremos salir.

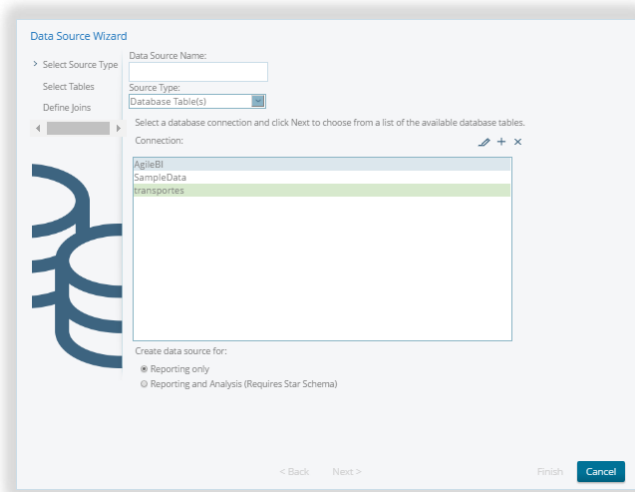


IMAGEN 20 CONEXIÓN CORRECTA DATA SOURCE

Observamos que ya tenemos creado nuestro *Data Source* en la Imagen 21.

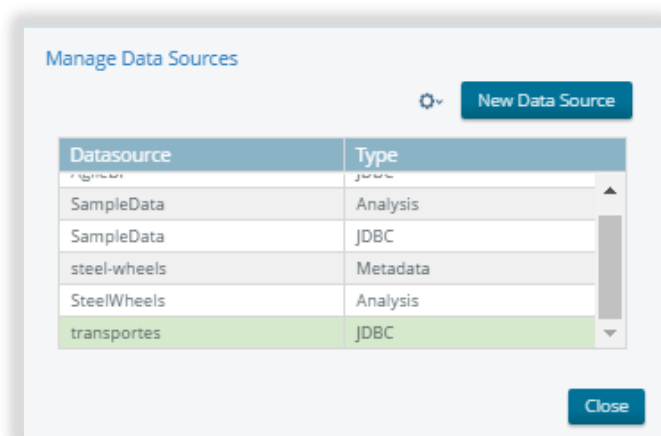


IMAGEN 21 VISTA DATA SOURCE

2.7. Añadiendo componentes

Necesitamos descargarnos de la página oficial de Pentaho el programa **data-integration** para poder seguir con el proceso de Business Intelligence. Una vez descargado hay que poner la carpeta en el directorio donde tenemos Pentaho instalado **/opt/data-integration**.

También necesitaremos instalar **Schema Workbench** para la creación de cubos Olap y **Report Designer** para la creación de informes. Todas estas herramientas se instalarán en el directorio **/opt/**.

Necesitamos descargarnos **Saiku Analytics** del Marketplace. Saiku nos va a permitir hacer consultas de una manera más intuitiva y gráfica. Para instalarlo nos dirigimos al Marketplace como indica la Imagen 22.

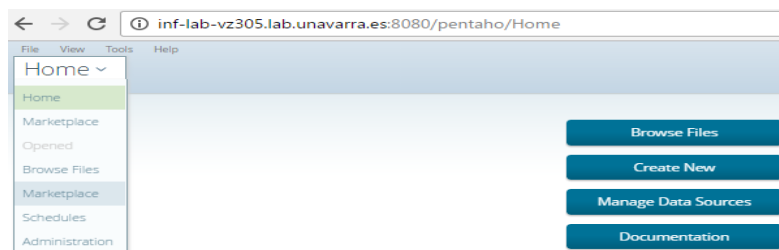


IMAGEN 22 MARKETPLACE

Una vez dentro buscamos Saiku y elegimos la primera opción, tal y como se aprecia en la Imagen 23.

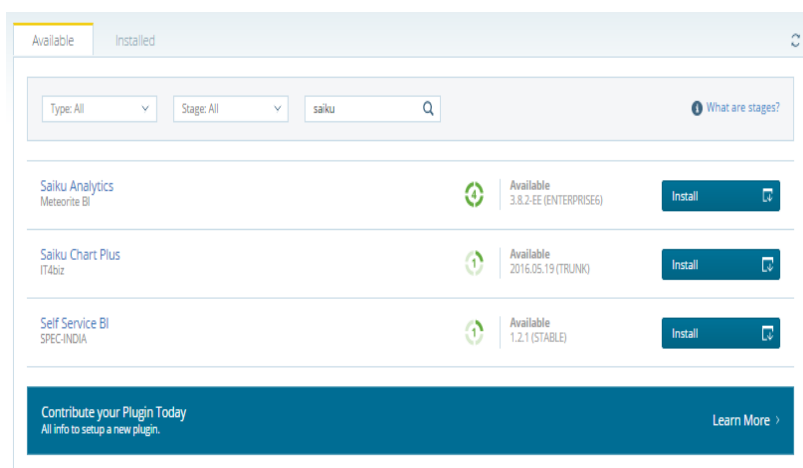


IMAGEN 23 SAIKU ANALYTICS

A continuación, instalamos *Saiku* eligiendo Pentaho 6 (ver Imagen 24).



IMAGEN 24 INSTALACIÓN DE SAIKU

3. Proceso ETL (Extract, Transform and Load)

Una vez configurado todo lo anterior para utilizar Pentaho, estamos preparados para realizar el sistema BI. Para ello empezaremos realizando el proceso ETL.

La realización de este proyecto se llevará a cabo con datos extraídos de ficheros de Excel de las fuentes antes citadas, mediante el proceso ETL. Estos datos se cargarán a una Base de datos MySQL llamada transportes.

3.1. Herramienta Data Integration

Para la realización del proceso ETL se necesitará descargar la herramienta *Data Integration*. La herramienta gráfica para diseñar y probar los procesos PDI es "spoon".

Iniciaremos la herramienta ejecutando el archivo **spoon.sh** que se encuentra en el directorio **/opt/data-integration/** como indica la Imagen 25.

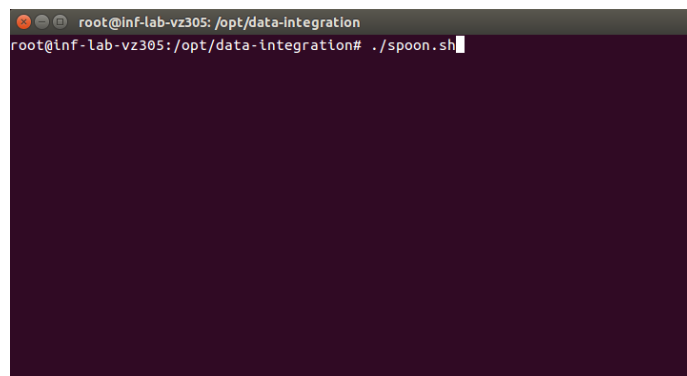


IMAGEN 25 SPOON

Una vez abierta la herramienta aparecerá la pantalla de inicio (ver Imagen 26).

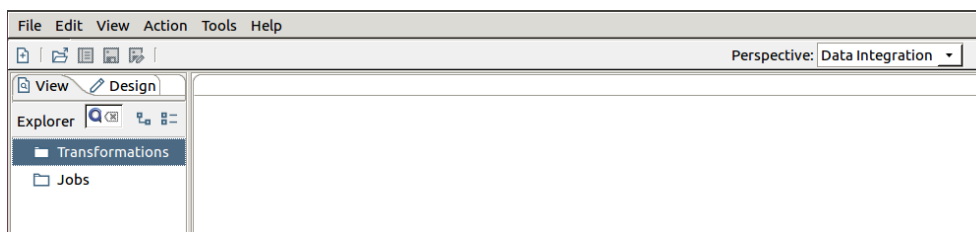


IMAGEN 26 INICIO DATA INTEGRATION

3.2. Nueva Transformación

Para realizar el proceso ETL tenemos que crear una nueva transformación para poder poblar la base de datos MySQL con los datos del Excel correctamente.

Realizamos los pasos de la Imagen 27 para proceder a la creación de una nueva transformación.

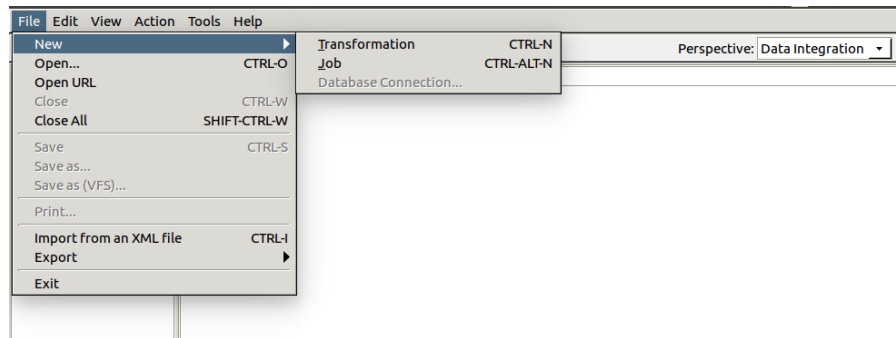


IMAGEN 27 NUEVA TRANSFORMACIÓN

3.3. Entrada Excel

La Imagen 28 muestra la opción de entrada de datos "Microsoft Excel Input". A cada una de estas entradas hay que indicarle la ruta del fichero Excel del que debe extraer los datos.

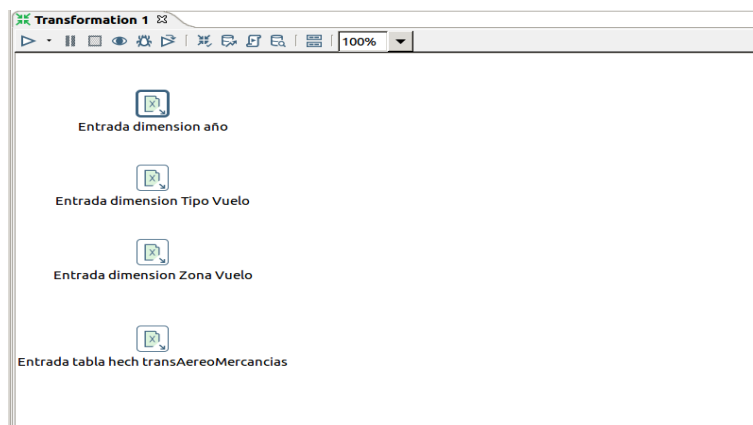


IMAGEN 28 ENTRADA EXCEL

Para ver todas las propiedades que tiene la opción de entrada de datos debemos seleccionar doble *click* sobre el icono.

Una vez hecho esto nos aparecerá una ventana como la de la Imagen 29 y es aquí cuando elegiremos la ruta del fichero Excel correspondiente.

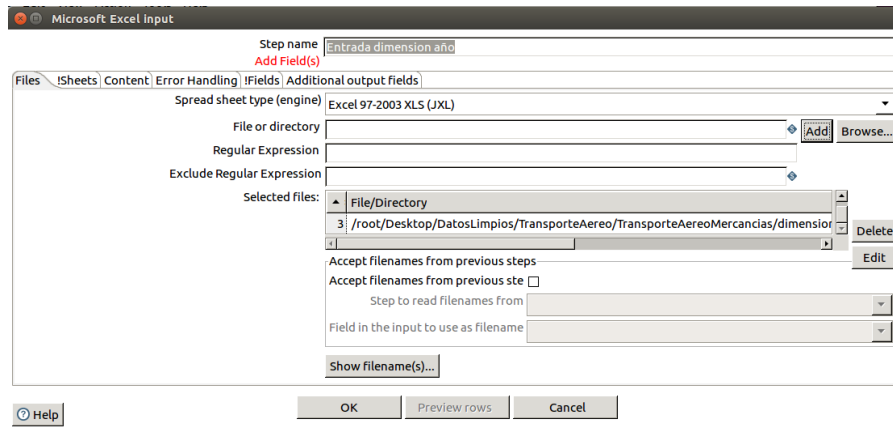


IMAGEN 29 INPUT ENTRADA EXCEL

Otra opción que nos permite la entrada de datos Excel es elegir de qué hoja se van a extraer los datos (ver Imagen 30).

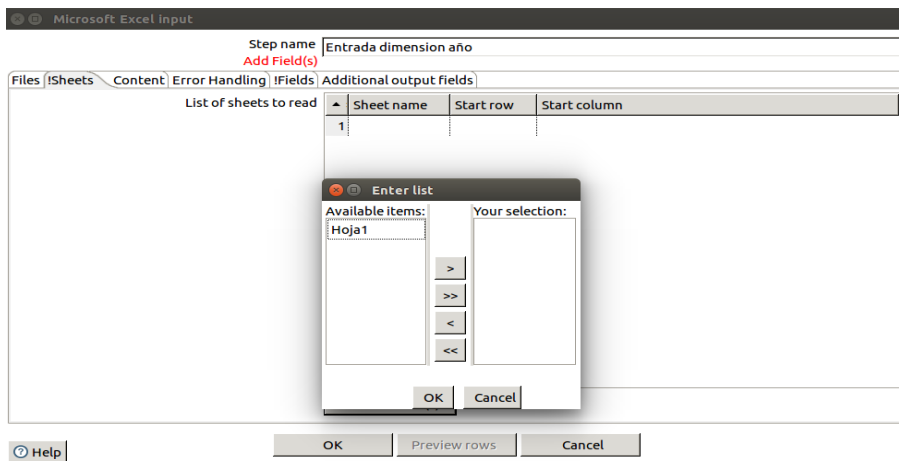


IMAGEN 30 HOJAS ENTRADA EXCEL

La opción *Fields* nos permite extraer los campos de la cabecera del fichero Excel y darles el formato deseado. Además, ofrece la posibilidad de añadir campos extra usando la opción *Additional output fields* (ver Imagen 31).

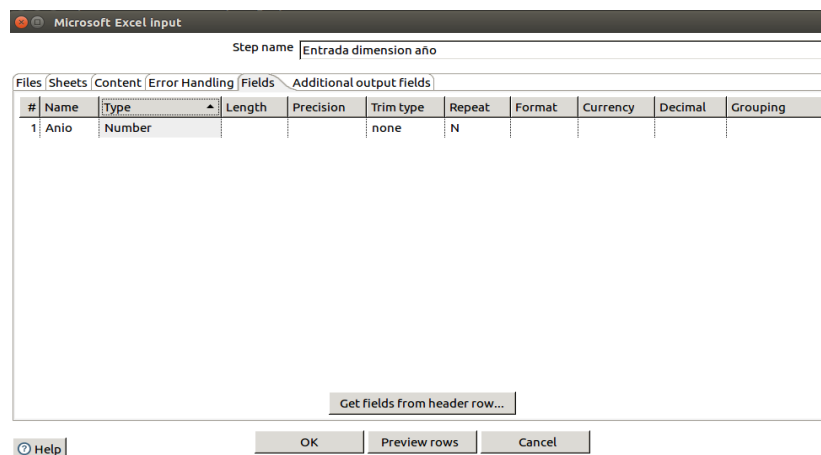


IMAGEN 31 CAMPOS ENTRADA EXCEL

Una vez completado el proceso de entrada de datos para un fichero Excel, hay que repetirlo de la misma manera para las demás entradas.

3.4. Transformación

Este es un paso muy importante en el proceso ETL ya que permite organizar la información de una manera correcta, cambiando el tipo de datos, nombres y la posición de los mismos para su correcta inserción en una tabla de la base de datos.

Cabe indicar que en la realización del primer dashboard del proyecto (prueba) no se utilizó ninguna transformación ya que los archivos Excel estaban bien estructurados y no era necesario.

Para la realización de los dashboards definitivos ha sido fundamental la utilización de transformaciones durante el proceso ETL.

La Imagen 32 muestra un ejemplo de una transformación.

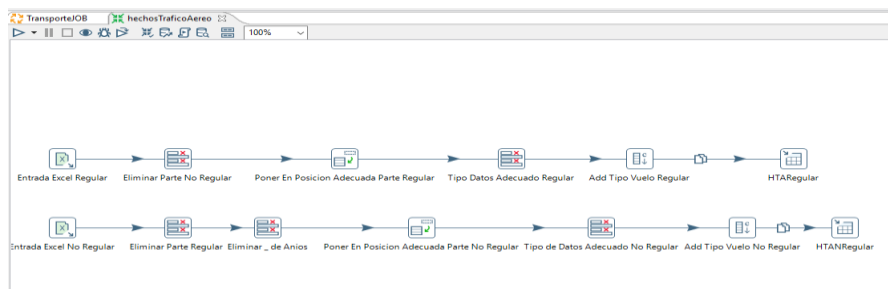


IMAGEN 32 TRANSFORMACIÓN

Hay ocasiones en las que no se pueden transformar los ficheros Excel utilizando las opciones que nos proporciona la herramienta *data-integration* y es necesario crear procedimientos que solucionen nuestros problemas. La Imagen 33 muestra una transformación utilizando procedimientos.

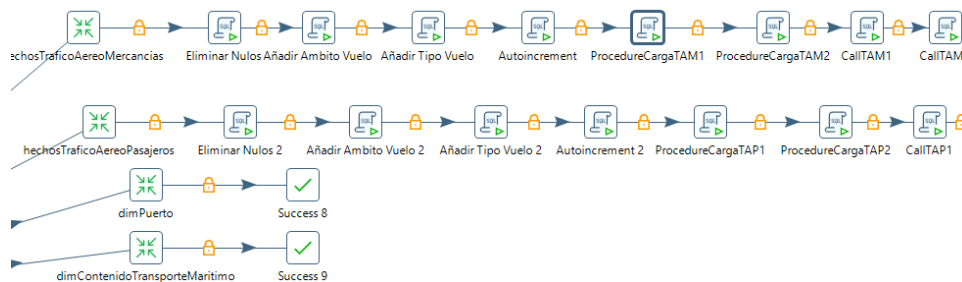


IMAGEN 33 USO PROCEDURE

La Imagen 34 muestra un procedimiento para cargar parte de los datos que necesitamos a una tabla en nuestra base de datos.

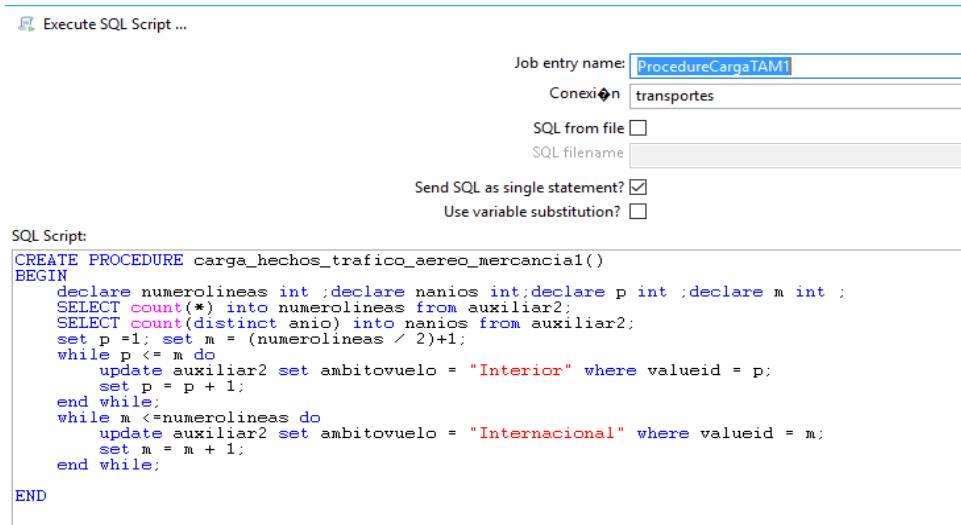


IMAGEN 34 PROCEDURE

3.5. Salida a Tabla

Una vez que tenemos los datos en el formato correcto, lo siguiente que hay que hacer es agregarlos a una tabla. Para ello elegimos la opción salida a tabla (ver Imagen 35).

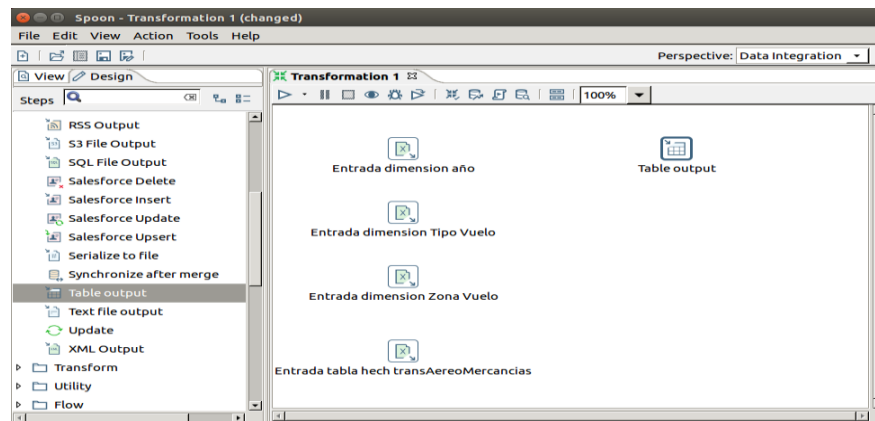


IMAGEN 35 SALIDA A TABLA

Para acceder a las opciones de Salida a tabla, al igual que en la opción entrada Excel, hay que pulsar dos veces en el icono. La Imagen 36 muestra las opciones que tiene la salida a tabla.

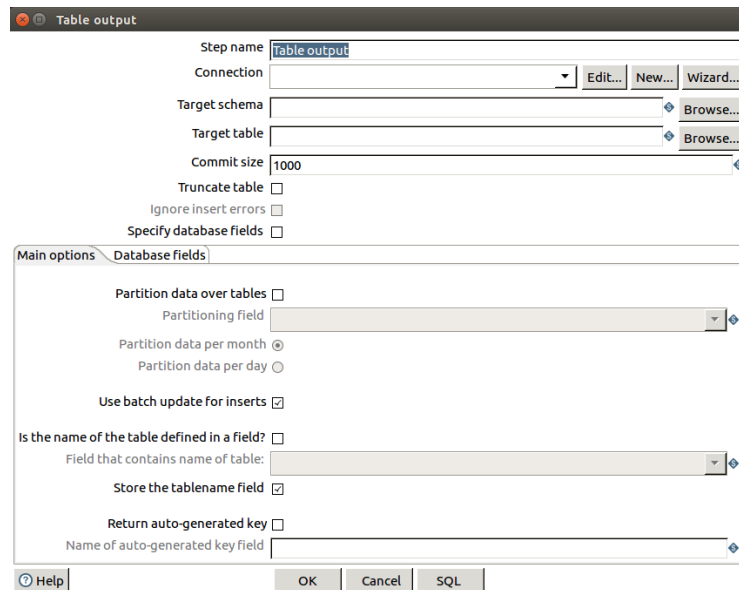


IMAGEN 36 CONFIGURACIÓN SALIDA A TABLA

Un paso muy importante que hay que realizar es agregar la conexión con nuestra base de datos pulsando el botón *new* en *connection*. La ilustración 37 muestra los datos necesarios para que se establezca la conexión con éxito.

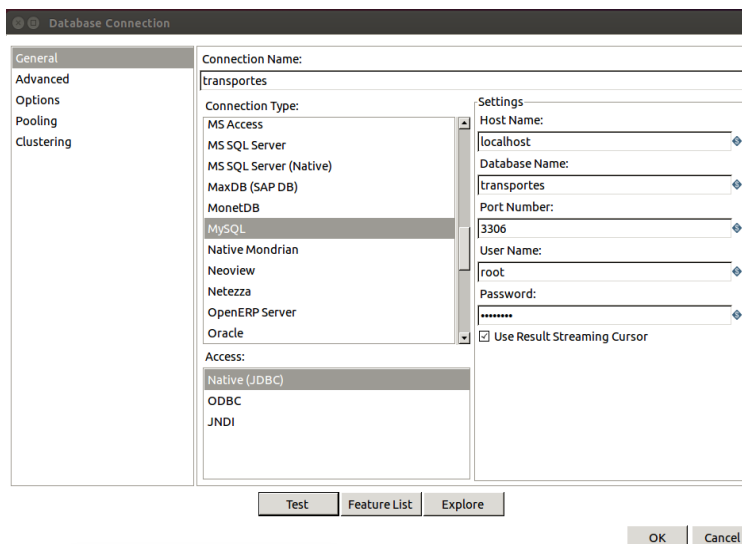


IMAGEN 37 CONEXIÓN CON BD SALIDA A TABLA

Una vez completados todos los datos procederemos a pulsar el botón de Test para ver si se ha establecido la conexión. En caso de que ocurra un error como el de la Imagen 38 hay que introducir el conector de MySQL [23] dentro del directorio **/data-integration/lib**.

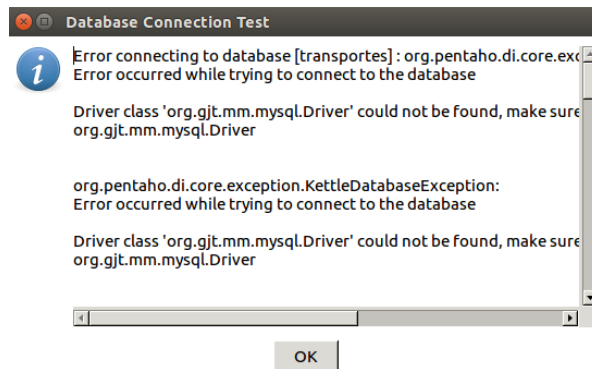


IMAGEN 38 ERROR TEST CONEXIÓN BD

Una vez hecho esto, pulsaremos el botón Test. Si todo ha salido correctamente aparecerá un mensaje como el de la Imagen 39.

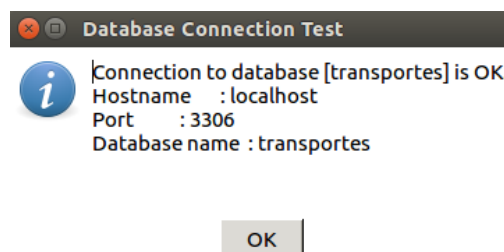


IMAGEN 39 TEST CONEXIÓN CORRECTA

Ya tenemos la conexión con nuestra base de datos así que procederemos a nombrar nuestra tabla usando la opción *Target table* y a continuación seleccionaremos el botón SQL que nos generará la instrucción SQL que creará nuestra tabla (ver ilustración 40).

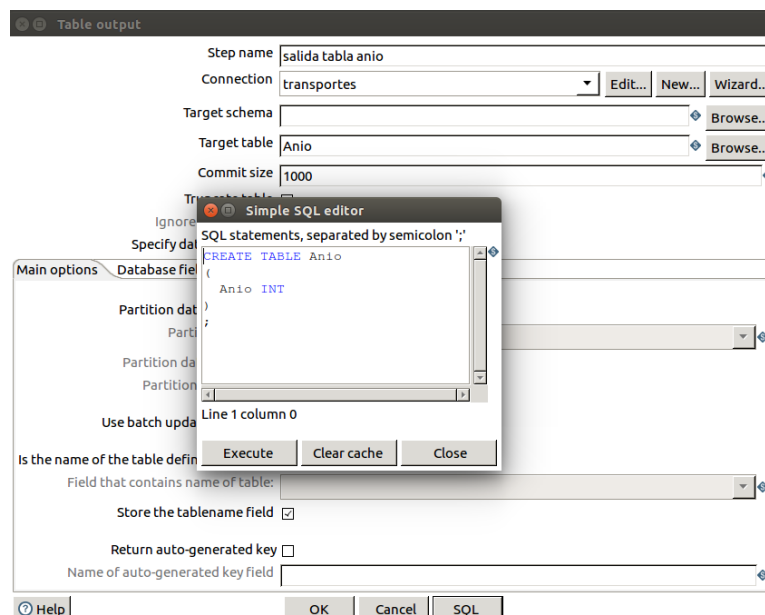


IMAGEN 40 CREATE TABLE DATA INTEGRATION

Una vez realizado el proceso, pulsaremos el botón *Execute* y se creará la tabla correspondiente. Este mismo proceso hay que realizarlo para todas las entradas Excel, ya que por cada fichero de Excel se debe generar una tabla.

Una vez que tengamos todas las tablas creadas ya tendremos una vista de cómo queda el ETL, tal y como se observa en la Imagen 41.

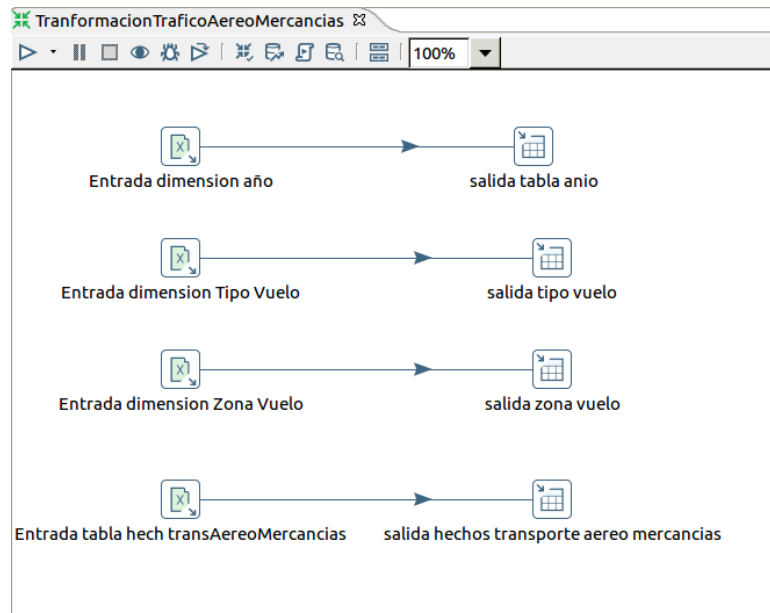


IMAGEN 41 VISTA ETL

3.6. Carga de Datos

Llegados a este punto podremos darle al botón *play* (ver Imagen 41, símbolo triangular). Una vez pulsado veremos una ventana como la de la Imagen 42 en la que pulsaremos el botón *run* para proceder a cargar los datos en las respectivas tablas.

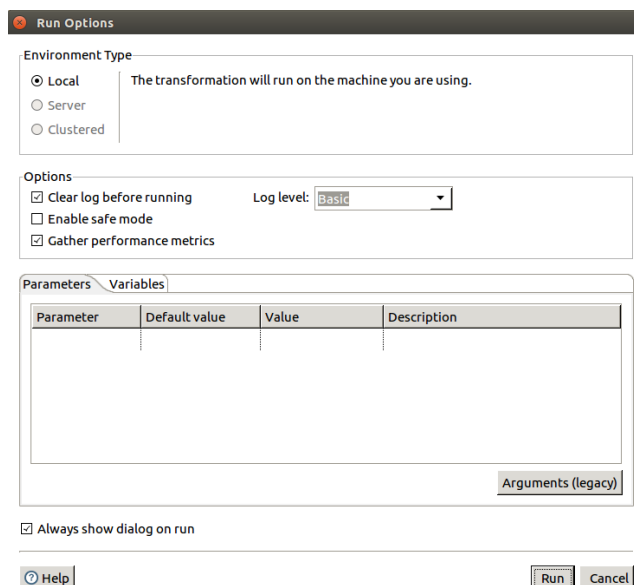


IMAGEN 42 CARGA DATOS EN BD

Después de pulsar el botón run empezará la carga de datos. Si en la vista general del ETL aparecen unos *ticks* en verde, quiere decir que todo ha funcionado correctamente y los datos se han cargado en sus correspondientes tablas de la base de datos (ver Imagen 43).

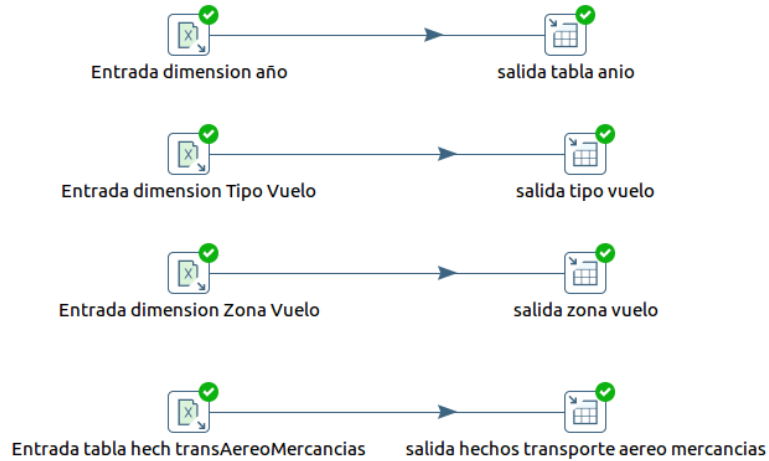


IMAGEN 43 CARGA DE DATOS CORRECTA

3.7. Opción Job

La herramienta *spoon* permite realizar *Jobs* o Trabajos que ejecutarán ficheros *.kjb* que contendrán distintas transformaciones (procesos de ETL). De esta manera, con un único JOB podremos ejecutar varios ETL facilitando el proceso de carga de datos llegándolo a hacer de manera autónoma si lo consideramos oportuno.

Para proceder a crear un nuevo trabajo pulsamos las opciones Fichero, Nuevo, Trabajo como indica la Imagen 44.

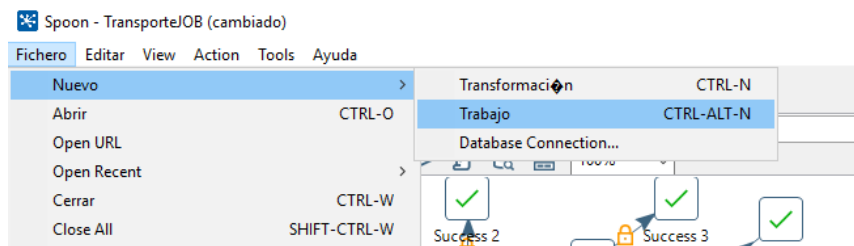


IMAGEN 44 CREACIÓN JOB

Dentro de este trabajo podemos añadir tantas transformaciones como queramos. La Imagen 45 muestra un ejemplo de JOB con sus respectivas transformaciones. Cabe destacar que es necesario indicarle la conexión a base de datos (ver Imagen 44). Para ejecutar el trabajo es necesario pulsar el triángulo que aparece en el menú de opciones.

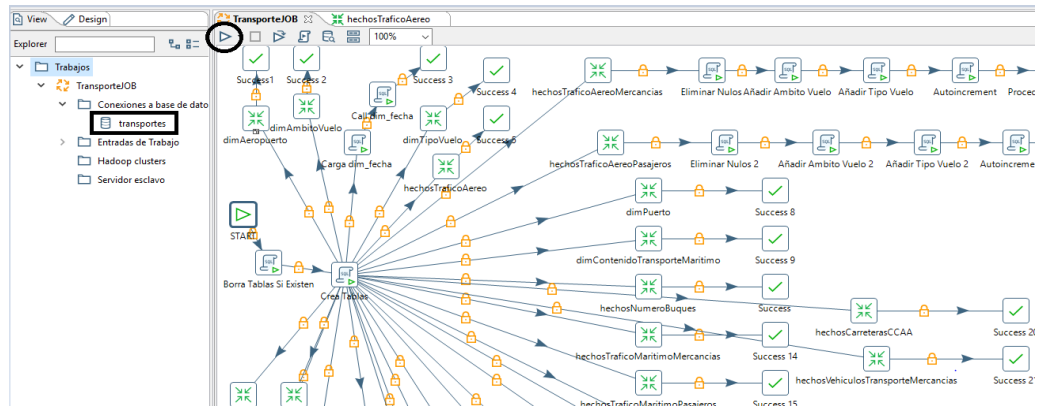


IMAGEN 45 VISTA DEL JOB

Una opción interesante en este job es que en cada ejecución se borra la base de datos y se crea de nuevo por si ha habido cambios de cualquier tipo. Eso se realizará mediante dos scripts SQL (ver Imagen 46 e Imagen 47).

Execute SQL Script ...

Job entry name: Borra Tablas Si Existen

Conexión: transportes

SQL from file:

SQL filename:

Send SQL as single statement?

Use variable substitution?

SQL Script:

```

drop procedure if exists carga_hechos_emisiones_gases1;
drop procedure if exists carga_hechos_emisiones_gases2;

drop procedure if exists carga_hechos_consumo_energia1;
drop procedure if exists carga_hechos_consumo_energia2;

drop procedure if exists poblar_dim_fecha;

/*Trafico Aereo*/
drop table if exists dim_aeropuerto cascade;
drop table if exists dim_ambitovuelo cascade;
drop table if exists dim_fecha cascade;
drop table if exists dim_tipovuelo cascade;
drop table if exists hechos_traficoaereo cascade;

drop table if exists hechos_trafico_aereo_mercancia cascade;
drop table if exists hechos_trafico_aereo_pasajeros cascade;

/*Auxiliares*/
drop table if exists auxiliar2 cascade;
drop table if exists auxiliar3 cascade;
drop table if exists auxiliar4 cascade;
drop table if exists auxiliar5 cascade;
drop table if exists auxiliar6 cascade;
drop table if exists auxiliar7 cascade;

/*Trafico Maritimo --- OK*/
drop table if exists dim_puerto cascade;
drop table if exists dim_mercanciamaritima cascade;

```

IMAGEN 46 SCRIPT BORRA TABLAS

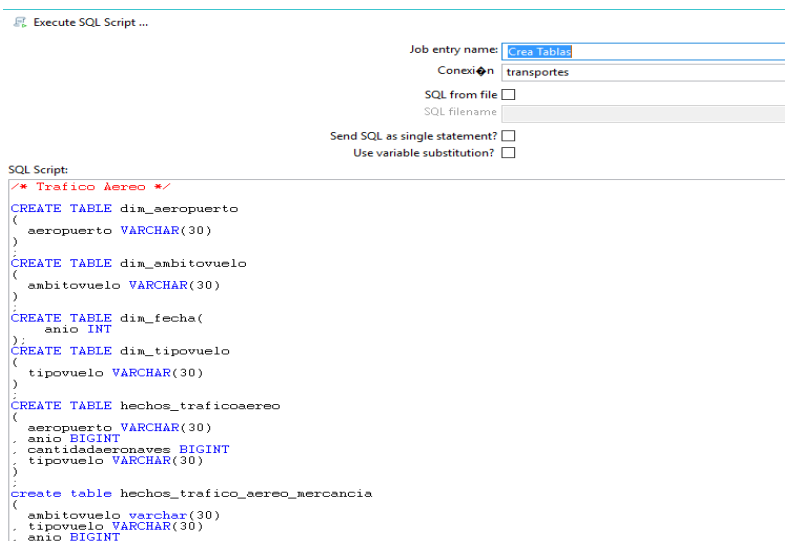


IMAGEN 47 SCRIPT CREA TABLAS

Otra funcionalidad interesante que tienen los *JOBS* es que se pueden programar para que se ejecuten automáticamente. En mi caso no he considerado esta opción relevante debido a que los datos se actualizan una vez al año (ver Imagen 48).

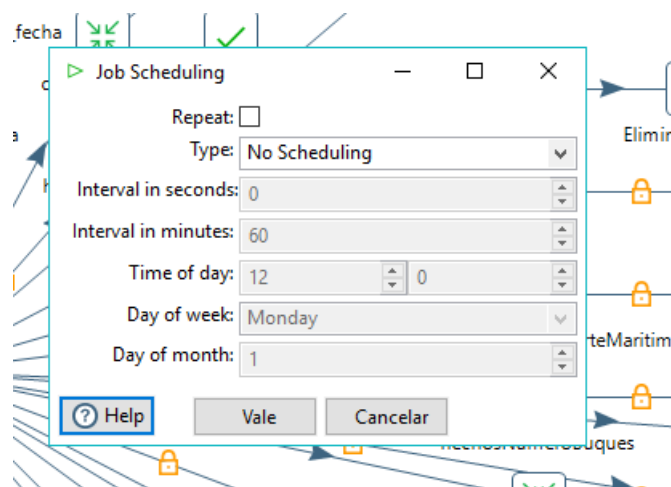


IMAGEN 48 SHEDULING JOB

Después de ejecutar el JOB, de la misma manera que una transformación, podemos observar que todo ha funcionado correctamente debido a que nos aparecen *ticks* en verde en cada paso que se va realizando, Además, en la consola se indica que el trabajo ha finalizado con éxito como se muestra en la Imagen 49.

The screenshot displays a workflow execution interface. At the top, a progress bar shows 100% completion. Below it, a complex flowchart illustrates the data pipeline, starting from a 'START' node and branching into multiple parallel paths. Each step in the flowchart is marked with a green checkmark, indicating successful execution. The paths include operations such as loading dimensions (e.g., 'Carga dim_fecha'), loading fact tables (e.g., 'hechosTráficoAereo'), and performing data transformations (e.g., 'Añadir Tipo Vuelo').

At the bottom of the interface, an 'Execution Results' table provides a detailed log of the job's progress. The table includes columns for the job name, comment, result, reason, file name, number of rows, and registration date. The final entry confirms that the job 'Trabajo: Transporte/OB' was completed successfully.

Trabajo / Entrada de Trabajo	Comentario	Resultado	Razón	Nombre Fichero	Núm	Fecha regis
Carga dim_fecha	Start of job execution		Followed unconditional link			2017/02/27 20:55
Carga dim_fecha	Job execution finished	Exito			0	2017/02/27 20:55
Call dim_fecha	Start of job execution		Followed unconditional link			2017/02/27 20:55
Call dim_fecha	Job execution finished	Exito			0	2017/02/27 20:55
Success 3	Start of job execution		Followed unconditional link			2017/02/27 20:55
Success 3	Job execution finished	Exito			0	2017/02/27 20:55
Trabajo: Transporte/OB	Job execution finished	Exito	finished		0	2017/02/27 20:55

IMAGEN 49 EJECUCIÓN JOB

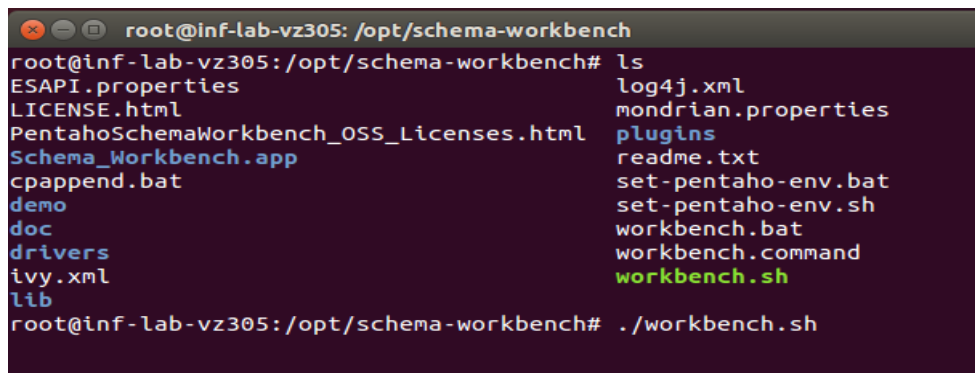
4. Creación del Cubo OLAP

Una vez que tenemos los datos almacenados en la base de datos de forma correcta, el siguiente paso será la creación del cubo OLAP.

Este proceso se realizará con la herramienta *Schema workbench* que previamente nos habíamos descargado.

4.1. Configuración de schema-workbench

Esta herramienta está alojada en el directorio */opt/*, para ejecutarla tendremos que ejecutar el siguiente comando *./workbench.sh* (ver Imagen 50).



```
root@inf-lab-vz305: /opt/schema-workbench
root@inf-lab-vz305: /opt/schema-workbench# ls
ESAPI.properties          log4j.xml
LICENSE.html              mondrian.properties
PentahoSchemaWorkbench_OSS_Licenses.html  plugins
Schema_Workbench.app      readme.txt
cpappend.bat              set-pentaho-env.bat
demo                       set-pentaho-env.sh
doc                         workbench.bat
drivers                    workbench.command
ivy.xml                    workbench.sh
lib
root@inf-lab-vz305: /opt/schema-workbench# ./workbench.sh
```

IMAGEN 50 EJECUCIÓN SCHEMA WORKBENCH

Para poder trabajar con *Schema Workbench* lo primero que tenemos que hacer es conectar nuestra base de datos con el programa, para ello nos vamos a la pestaña del menú opciones y rellenamos con nuestros datos las siguientes opciones.

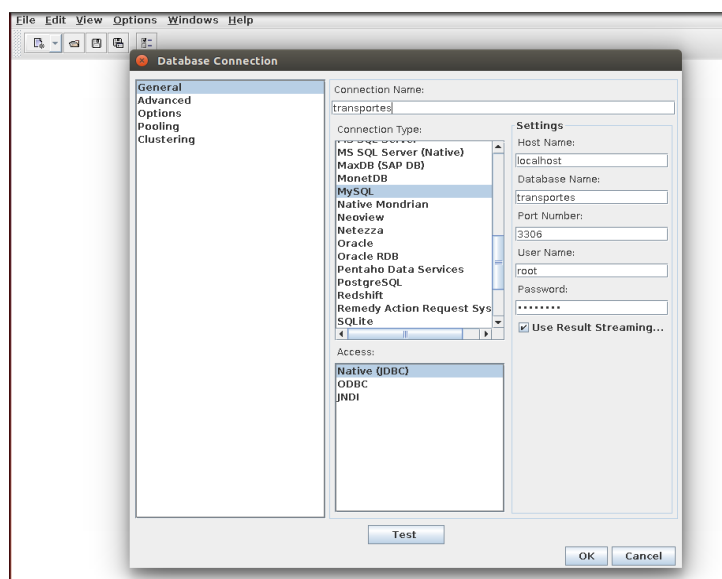


IMAGEN 51 CONEXIÓN CON BD EN SCHEMA WORKBENCH

Para comprobar que se ha conectado correctamente pulsaremos en el botón Test. Si nos aparece un error como el de la Imagen 52 tendremos que descargar el driver de conexión [22] y debemos copiar el archivo .jar en la carpeta drivers donde tenemos instalado *schema-workbench*, en nuestro caso */opt/schema-workbench/drivers*.

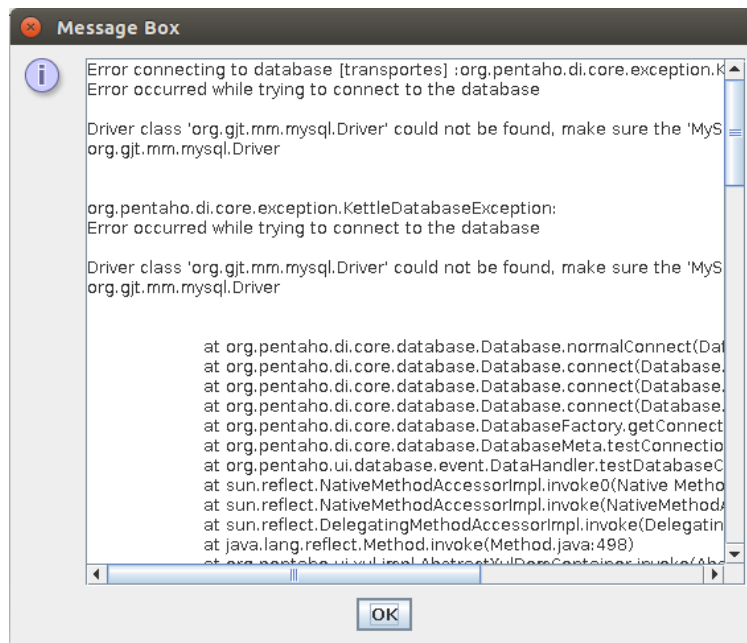


IMAGEN 52 ERROR TEST CONEXIÓN CON BD EN SCHEMA WORKBENCH

Una vez instalado el driver de conexión, se procederá a pulsar el botón test y la conexión se establecerá correctamente.

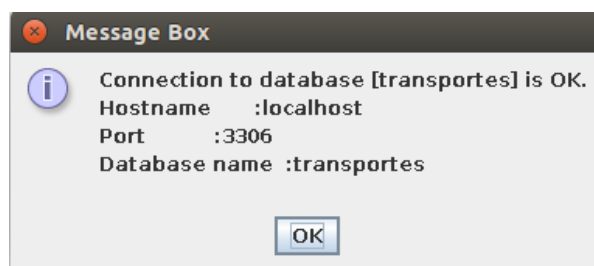


IMAGEN 53 TEST CORRECTO

Una vez realizado este proceso, ya estamos en disposición de realizar el cubo OLAP.

4.2. Creación de Cubo OLAP

Para crear el cubo OLAP debemos crear un nuevo *schema* pulsando en el botón del menú File **New Schema** como muestra la Imagen 54.

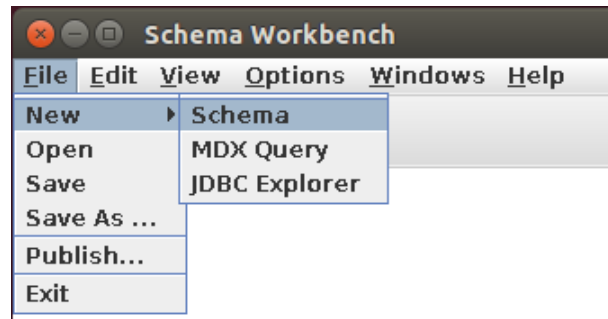


IMAGEN 54 NEW SCHEMA

A continuación le daremos un nombre al *schema*, en nuestro caso Cubos Transportes (ver Imagen 55).

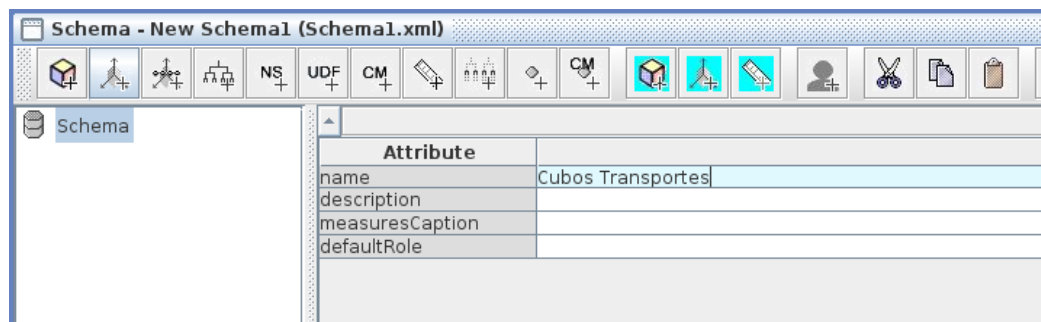


IMAGEN 55 NOMBRE SCHEMA

El siguiente paso es añadir un cubo al *schema*. Para ello seleccionamos con el botón derecho el *schema* y elegimos la opción *Add cube* como indica la Imagen 56.

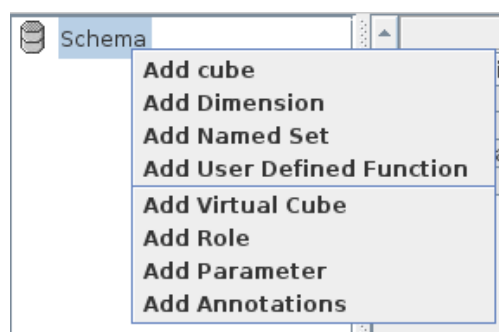


IMAGEN 56 ADD CUBE

A continuación asignaremos el nombre **Cubo_Hechos_Trafico_Aereo** al cubo e introduciremos la tabla de hechos correspondiente.

Para añadir la tabla de hechos pulsamos botón derecho en el cubo y seleccionamos la opción *Add Table* como indica la Imagen 56.

Una vez realizado el proceso, ya tendremos cargada nuestra tabla de hechos (ver Imagen 58).

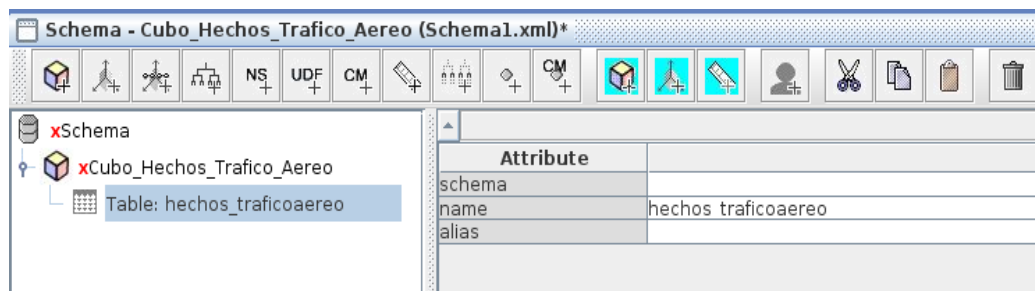


IMAGEN 57 VISTA TABLA HECHOS

Siguiendo los avisos de la herramienta debemos añadir las dimensiones. Esta herramienta tiene una opción interesante que permite crear dimensiones para el *schema*, de manera que cualquier cubo perteneciente al *schema* podrá utilizar esa dimensión.

Para poder utilizar estas dimensiones se necesitará crear *dimension usage* que se explicará más adelante.

Dicho esto, procederemos a crear las dimensiones dentro del *schema* y no dentro del cubo pulsando con el botón derecho en el *schema* y eligiendo la opción *Add Dimension*.

La Imagen 59 muestra todas las dimensiones necesarias para realizar el cubo.

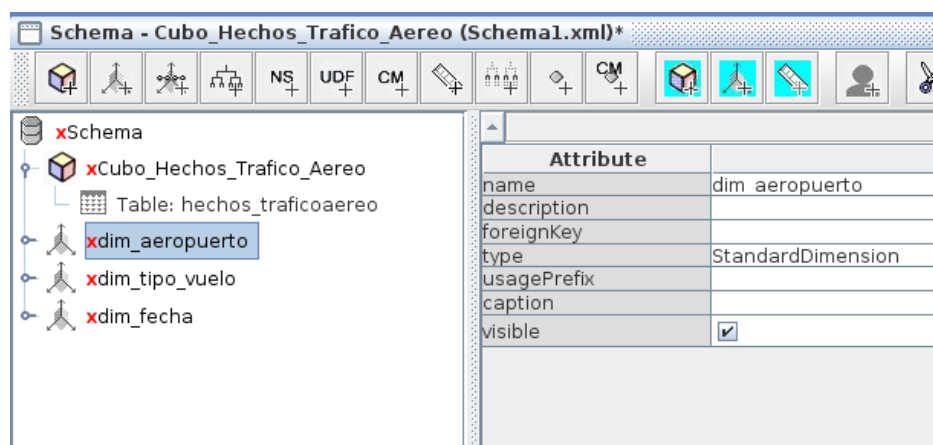


IMAGEN 58 AÑADIR DIMENSIÓN

A estas dimensiones hay que darles un nombre y añadirles una jerarquía seleccionando la opción *Add Hierarchy* sobre el icono de la dimensión.

Una vez realizado este paso se procederá a asignarle la tabla de la dimensión correspondiente, para ello sobre la jerarquía creada elegiremos *Add Table* como indica la Imagen 60.

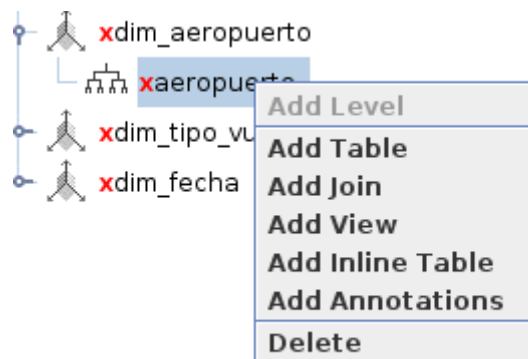


IMAGEN 59 AÑADIR TABLA DE DIMENSIÓN

A continuación vemos que se ha añadido la tabla de la dimensión y podemos proceder a crear el *Level* como indica la ilustración 61.

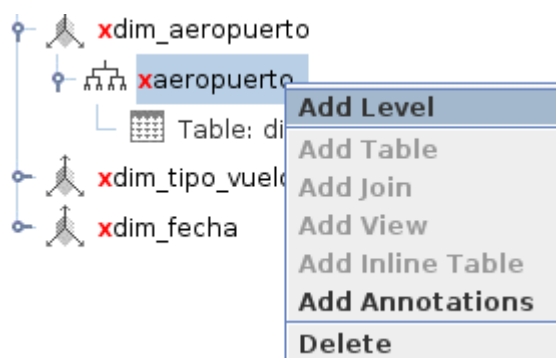


IMAGEN 60 AÑADIR NIVEL

Una vez que tenemos todos estos pasos finalizados, tenemos que volver a la jerarquía para indicarle la *primary key* de la tabla de la dimensión que posteriormente se ha añadido. La Imagen 62 ilustra el proceso.

Hierarchy for 'dim_aeropuerto' Dimension	
Attribute	Value
name	aeropuerto
description	
hasAll	<input checked="" type="checkbox"/>
allMemberName	
allMemberCaption	
allLevelName	
defaultMember	
memberReaderClass	
primaryKeyTable	
primaryKey	aeropuerto
caption	dim_aeropuerto->aeropuerto - VARCHAR(30)
visible	<input checked="" type="checkbox"/>

IMAGEN 61 ELEMENTOS DE LA JERARQUÍA

El siguiente paso será rellenar con los datos correctos la opción *Level*, asignándole un nombre, la tabla correspondiente, la columna de la tabla correcta y el tipo de datos adecuando. La Imagen 63 muestra todas las opciones.

Level for 'aeropuerto' Hierarchy	
Attribute	Value
name	aeropuerto
description	
table	dim aeropuerto
column	aeropuerto
nameColumn	
parentColumn	
nullParentValue	
ordinalColumn	
type	String
internalType	
uniqueMembers	<input type="checkbox"/>
levelType	Regular
hideMemberif	Never
approxRowCount	
caption	
captionColumn	
formatter	
visible	<input checked="" type="checkbox"/>

IMAGEN 62 ELEMENTOS DEL NIVEL

*Nota: Cabe destacar que una dimensión puede tener varios *Levels*, por ejemplo, la dimensión Localización puede tener *level* país y *level* ciudad

Llegados a este punto tenemos que indicarle a nuestro cubo que utilice las dimensiones que acabamos de crear. Aquí es donde se deben añadir las *dimensiones usages* en nuestro cubo (ver Imagen 64).

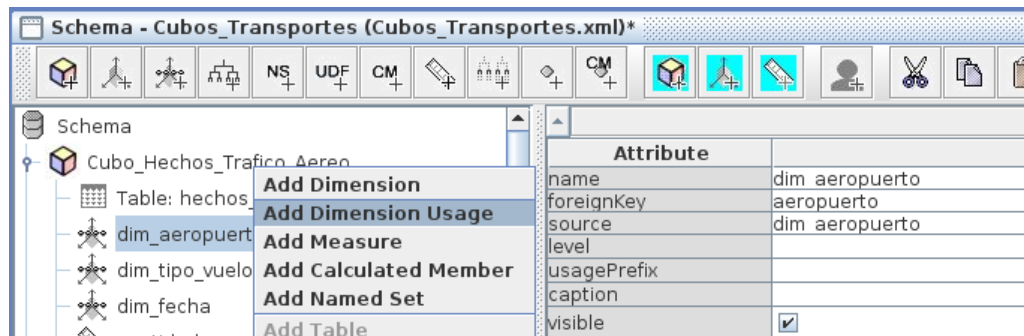


IMAGEN 63 DIMENSIÓN USAGE

Como se puede observar en la ilustración 66 para el caso de la dimensión *usage* aeropuerto hay que asignarle un nombre, la clave extranjera y la fuente que es la dimensión aeropuerto que hemos creado para el *schema*.

Por último necesitaremos añadir las medidas, para ello elegimos *Add Measure* seleccionando nuestro cubo como indica la ilustración 65.

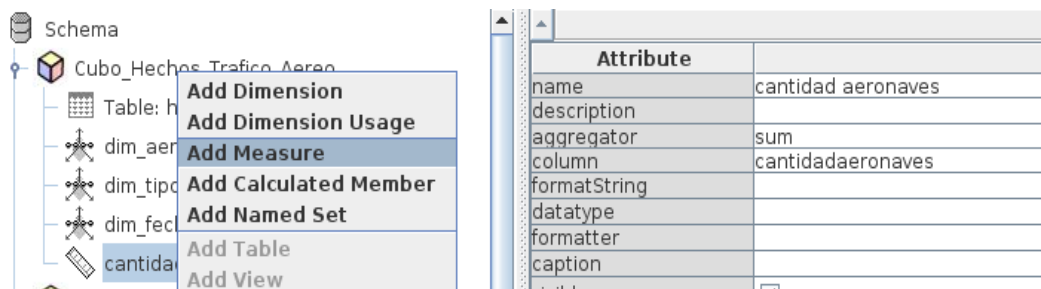


IMAGEN 64 AÑADIR MEDIDA

A la medida debemos asignarle un nombre, el agregador y el campo correspondiente en la tabla.

Este mismo proceso que se ha realizado para la dimensión aeropuerto hay que realizarlo para las demás dimensiones y una vez finalizado ya tendremos nuestro cubo construido.

Cabe destacar que la dim_fecha es una dimensión importante ya que esta debe estar presente en todos los cubos.

Después de repetir el proceso ya tenemos todos nuestros cubos y dimensiones creados como se puede ver en la Imagen 66.



IMAGEN 65 VISTA DE LOS CUBOS

Para poder usar el esquema con el cubo tenemos que publicarlo. Para ello tenemos que pulsar la opción *publish* como indica la Imagen 67.

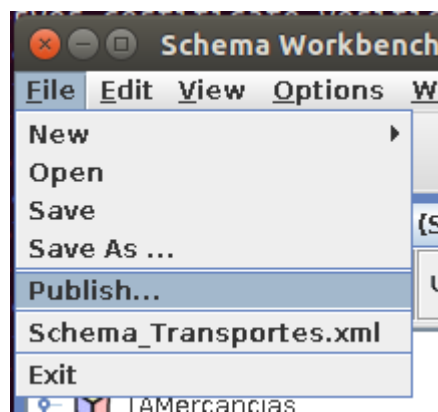


IMAGEN 66 PUBLICAR CUBOS

Indicaremos las credenciales de Pentaho y se publicará correctamente para poderlos utilizar en Pentaho (ver Imagen 68).

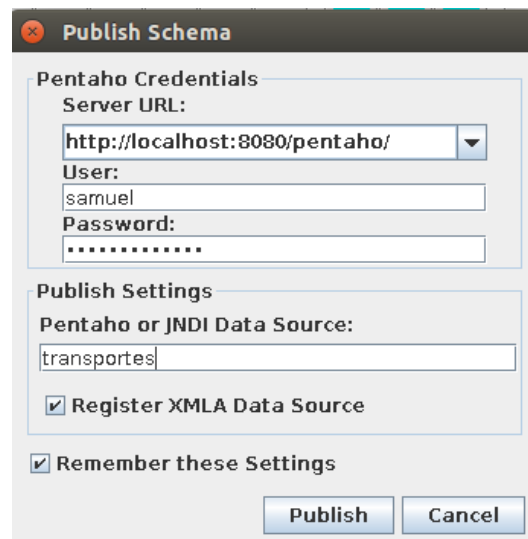


IMAGEN 67 CREDENCIALES PENTAHO

Pulsando el botón *Publish* se indica que la publicación ha sido correcta.

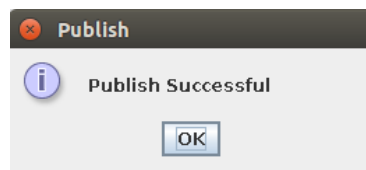


IMAGEN 68 PUBLICACIÓN CORRECTA

5. Creación de Dashboard

En este capítulo se va a proceder a explicar la construcción de Dashboards que son herramientas que permite visualizar los datos que se han obtenido de los procesos anteriores. Estos Dashboards servirán para detectar posibles problemas (sobre los datos que estamos tratando) y favorecer en la toma de decisiones.

5.1. Uso de Saiku

Para poder construir nuestros Dashboards (Cuadros de mando), la herramienta *Saiku* juega un papel fundamental ya que en ella se realizarán las consultas mdx. Mdx es un lenguaje de consulta para bases de datos multidimensionales sobre cubos Olap.

Los datos obtenidos de estas consultas servirán para mostrar las gráficas en el Dashboard. *Saiku* también es una herramienta importante para saber si los cubos construidos en *Schema Workbench* son correctos o no porque permite ver los cubos publicados, así como las medidas y las dimensiones.

Accedemos a esta herramienta desde Pentaho como se indica en la Imagen 70.



IMAGEN 69 ACCESO A SAIKU

Una vez dentro de Saiku, seleccionaremos la opción **Create a New Query** y seleccionamos el cubo que queremos de entre los que se han publicado anteriormente (ver ilustración 71).

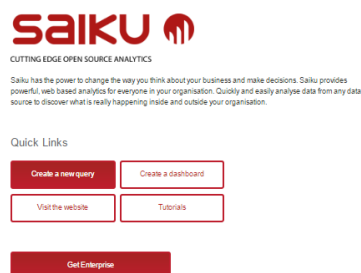


IMAGEN 70 CREATE NEW QUERY

Como podemos observar en la Imagen 72, tenemos todos los cubos que se han publicado, de los cuales extraeremos consultas mdx para posteriormente crear gráficos en los Dashboards.

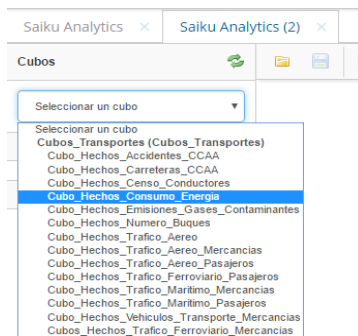


IMAGEN 71 VISTA DE CUBOS EN SAIKU

Seleccionaremos el cubo que queremos y una vez abierto aparecerá una ventana en la que habrá que arrastrar los datos que queremos mostrar hacia las posibilidades que se dan (Medidas, columnas, filas, filtro) tal y como se aprecia en la Imagen 73.

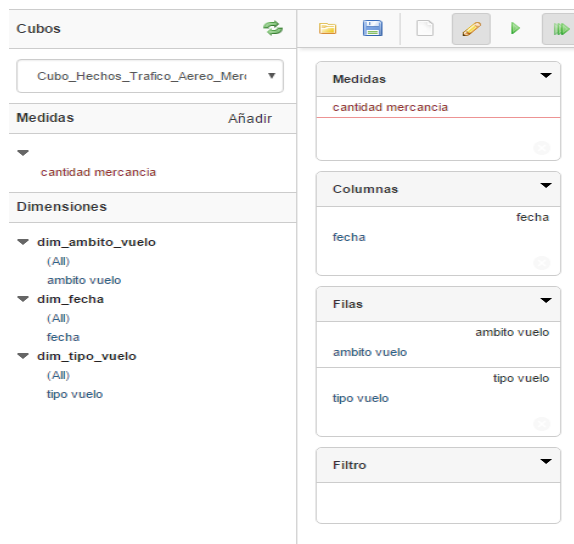


IMAGEN 72 USO DEL CUBO EN SAIKU

En este paso es cuando se comprueba si los cubos se han creado correctamente, ya que pueden fallar algunas medidas o dimensiones y esto implicará un retroceso al paso donde se crean los cubos con *Schema Workbench* y su revisión.

Una vez que hacemos esto, el resultado es una tabla con los datos solicitados (ver ilustración 74).

		1990	1991	1992	1993	1994	1995	1996
tipo vuelo	ambito vuelo	cantidad mercancia	cantidad mercancia	cantidad mercancia	cantidad mercancia	cantidad mercancia	cantidad mercancia	cantidad mercancia
No regular	Interior	10.466.490	13.739.110	13.110.348	23.820.241	29.314.004	29.504.631	35.560.170
	Internacional	27.828.602	17.712.293	21.487.038	25.399.680	38.522.955	50.534.541	60.323.256
Regular	Interior	189.840.212	180.155.410	181.088.140	185.526.910	164.917.551	172.201.087	178.808.912
	Internacional	196.597.836	171.220.198	181.096.731	184.935.735	180.388.786	200.564.880	217.185.925

IMAGEN 73 TABLA DE DATOS EXTRAÍDA DEL CUBO EN SAIKU

Saiku también ofrece la posibilidad de realizar gráficos a partir de datos. Sólo tenemos que elegir la opción gráfica y aparecerán varios que podremos elegir.

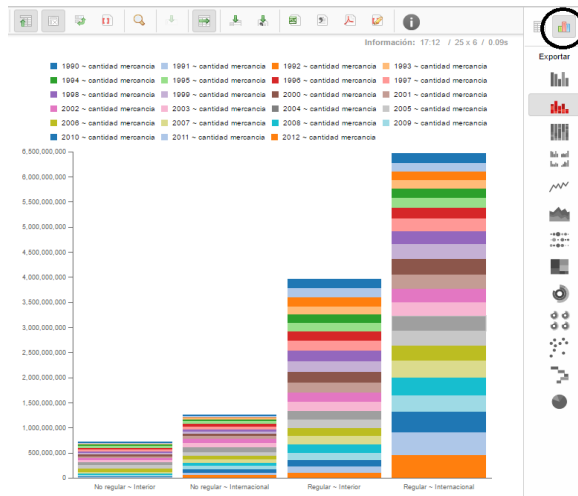


IMAGEN 74 GRÁFICO EN SAIKU

Para poder usar estos datos dentro de un Dashboard en Pentaho, es necesario recoger la consulta mdx que se genera en Saiku.

Para poder obtenerla, una vez que tenemos la tabla con los datos, hay que seleccionar la opción que se muestra en la Imagen 76.

```

MDX

WITH
SET [~COLUMNS] AS
  {[dim_fecha.fecha].[fecha].Members}
SET [~ROWS_dim_tipo_vuelo_dim_tipo_vuelo.tipo_vuelo] AS
  {[dim_tipo_vuelo.tipo_vuelo].[tipo_vuelo].Members}
SET [~ROWS_dim_ambito_vuelo_dim_ambito_vuelo.ambito_vuelo] AS
  {[dim_ambito_vuelo.ambito_vuelo].[ambito_vuelo].Members}
SELECT
NON EMPTY CrossJoin([~COLUMNS], {[Measures].[cantidad mercancia]}) ON
COLUMNS,
NON EMPTY
NonEmptyCrossJoin([~ROWS_dim_tipo_vuelo_dim_tipo_vuelo.tipo_vuelo],
[~ROWS_dim_ambito_vuelo_dim_ambito_vuelo.ambito_vuelo]) ON ROWS
FROM [Cubo_Hechos_Trafico_Aereo_Mercancias]
    
```

IMAGEN 75 QUERY MDX

5.2. Creación de Dashboard

Para la creación de un Dashboard es necesario elegir la opción **Create New** y **CDE Dashboard** tal como se ilustra en la Imagen 77.

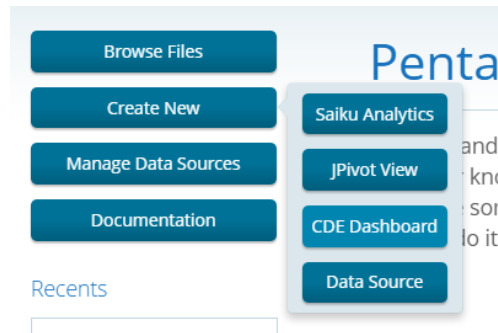


IMAGEN 76 NEW DASHBOARD

Una vez dentro de la configuración del Dashboard, aparecen cuatro opciones importantes que debemos utilizar.

- 1º. En esta opción se va a realizar la estructura de la vista del Dashboard.
- 2º. En esta opción se generan las gráficas, y si es necesario, los parámetros y selectores.
- 3º. Debemos generar consultas mdx dentro de esta opción para nutrir a las gráficas que hemos creado en el paso anterior. Copiaremos la consulta de *saiku* dentro de las mdx de este paso.
- 4º. El último paso sirve para visualizar lo que estamos haciendo.

La Imagen 78 muestra esas opciones.

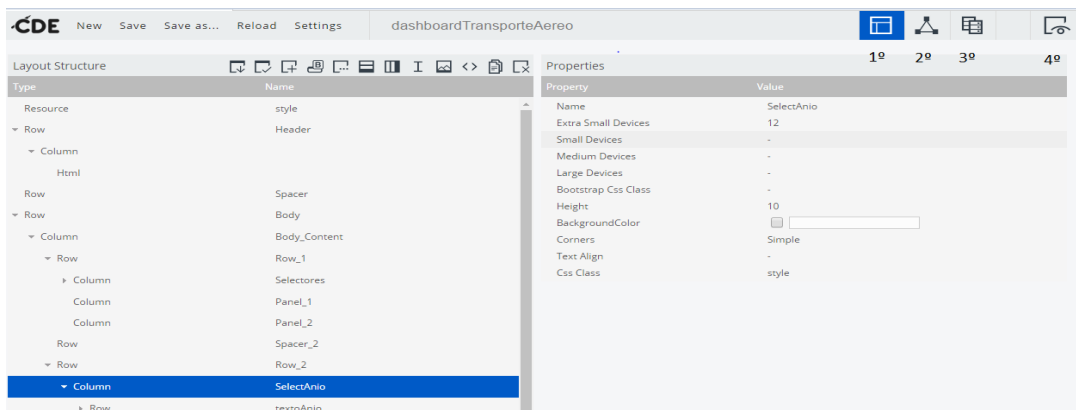


IMAGEN 77 OPCIONES CONFIGURACIÓN DASHBOARD

Después de que tengamos la estructura creada, tenemos que usar la consulta mdx de Saiku para poder pasársela a los gráficos que se crearán en la opción **2º**. Así que seleccionamos la opción **3º**.

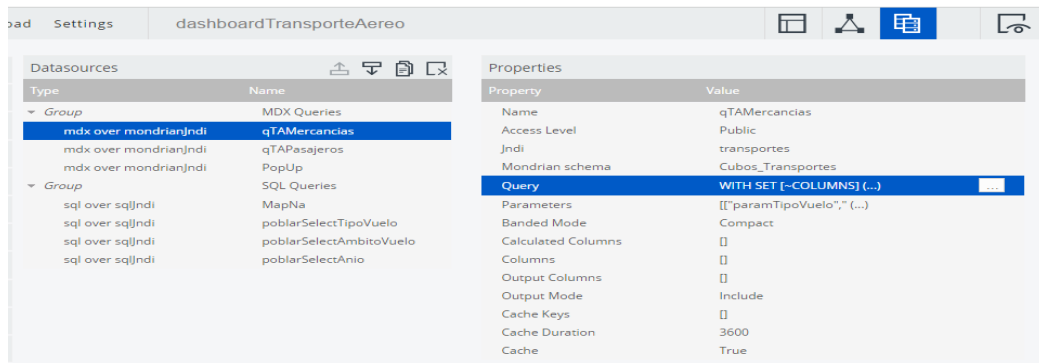


IMAGEN 78 QUERY PENTAHO

En este paso (Imagen 79) hay que rellenar las opciones:

- **Jndi (Java Naming and Directory Interface):** Permite guardar y recoger datos identificados por un nombre, organizando dichos datos en una estructura que puede ser similar a una estructura de directorios.
- **Mondrian Schema:** Es el esquema donde están los cubos Olap realizados anteriormente y sobre el que se va a realizar las consultas.

Para nuestro ejemplo **Jndi** será **transportes** que es la Base de Datos donde tenemos los datos y para **Mondrian Schema** tenemos **Cubos_Transportes** que es el schema donde tenemos nuestros cubos.

La nomenclatura de todas las consultas realizadas en los dashboards seguirá la norma de comenzar con la letra **q** para mayor facilidad a la hora de localizarlas.

Seleccionamos la consulta qTAMercancias y seleccionamos en la opción Query para poner la mdx.

En la Imagen 80 se puede ver un ejemplo de consulta mdx.

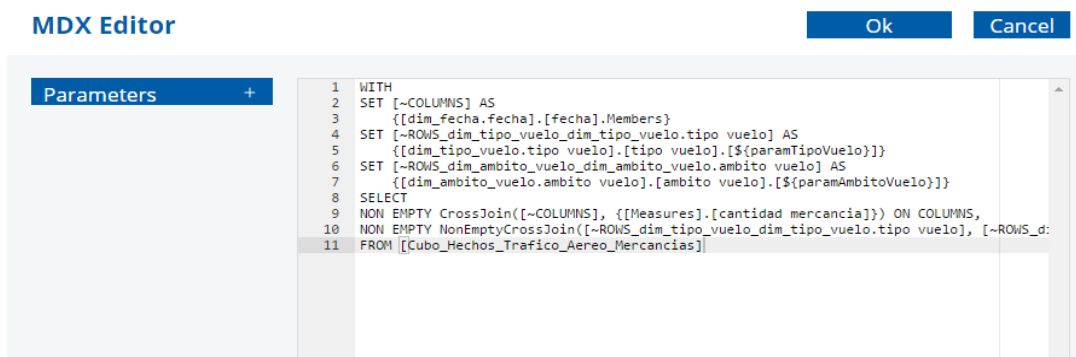


IMAGEN 79 EDITOR MDX

Como se puede apreciar, si queremos usar parámetros, en este caso mediante selectores, hay que indicar que la consulta haga caso a el parámetro usando $\{\text{parámetro}\}$ en vez de *Members*. Estos parámetros habrá que crearlos en el paso 2º.

En el paso 2º hay que crear las gráficas (indicando como *DataSource* la mdx creada en el paso anterior), parámetros y selectores.

Otras opciones a tener en cuenta en la gráfica son los *listeners* que escuchen al parámetro adecuado y hay que indicarle en *parameters* los parámetros que se usan.

En el paso 2º, en este caso para las gráficas y selectores, hay que indicar dónde se ubicarán estos elementos en el paso 1º. Para ello se utiliza la opción *HtmObject*, donde hay que introducir el nombre del elemento html del paso 1º donde queremos posicionar el componente.

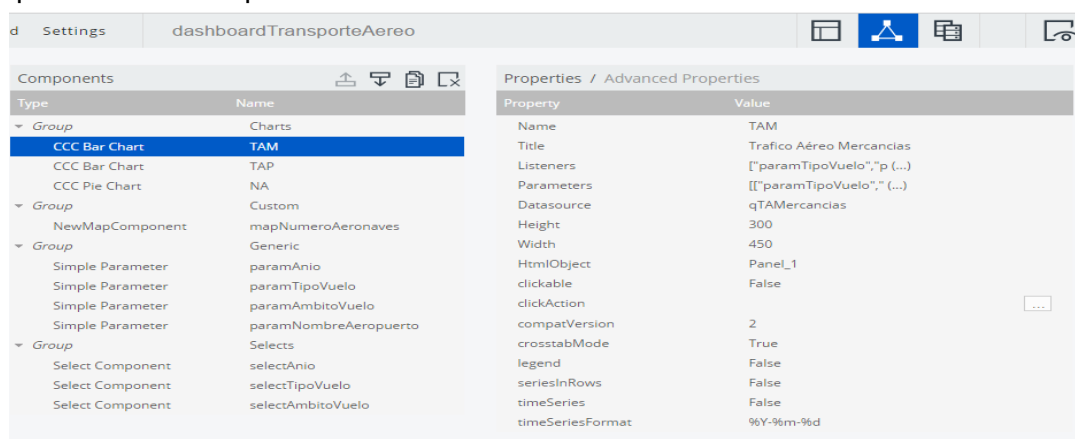


IMAGEN 80 GRÁFICO EN PENTAHO

Una vez completado todo el proceso, podemos seleccionar la opción 4ª para visualizar cómo ha quedado la gráfica **Tráfico Aéreo de Mercancías**, y siguiendo el proceso podemos crear Dashboards tan complejos como queramos.

En este caso he pensado en hacer una especie de *index* relacionando los distintos Dashboards para tener toda la información organizada por tipo de transporte.

El proceso anterior describe cómo se ha realizado el Dashboard Transporte Aéreo, pero se han realizado más. Como se puede observar en el Dashboard Index del siguiente apartado, cada imagen hace referencia a otro Dashboard que contendrá información de su temática.

5.3. Ejemplos de Dashboards

Este es el Dashboard principal que servirá para acceder a los otros Dashboards seleccionando la imagen según el tema que se quiera consultar.



IMAGEN 81 DASHBOARD INDEX

Si seleccionamos la imagen del avión, que es la correspondiente al Dashboard Transporte Aéreo, nos lleva a dicho Dashboard.

Este Dashboard dispondrá de unos filtros en la parte superior, así como un botón que nos llevará otra vez a el Dashboard index. Seleccionando las distintas opciones de filtrado se podrá ver la cantidad de tráfico aéreo existente, ya sea de pasajeros o de mercancías.

La parte inferior del Dashboard muestra un mapa en el que aparecen los aeropuertos que hay en España, pulsando en cada uno de ellos se podrá ver la cantidad de aviones que han utilizado el aeropuerto mediante la aparición de una gráfica a modo de *PopUp*.

Como se puede observar la imagen 83 muestra el Dashboard referente al transporte aéreo.

La parte superior del Dashboard contiene los selectores: Año, Ámbito de Vuelo y Tipo de Vuelo, según el valor que tengan estos selectores los gráficos variarán pudiendo así ver los cambios de una manera visual.

En la parte superior también tenemos un botón llamado Volver a Index que servirá para regresar al Dashboard principal.

Los gráficos de barras muestran el tráfico de mercancías y de pasajeros que soporta el transporte aéreo, cada barra corresponde a un año en concreto.

El tráfico aéreo de mercancías se observa que no tiene una tendencia clara, sin embargo, en el gráfico de transporte aéreo de pasajeros se puede observar una tendencia ascendente conforme pasan los años. Después de llegar al pico más alto los últimos 5 años tienen una tendencia descendente.

Por último la parte inferior del Dashboard contiene un mapa con todos los aeropuertos de España, seleccionando cada uno de ellos se mostrará un popup con la cantidad de aviones que realizan vuelos regulares y no regulares.

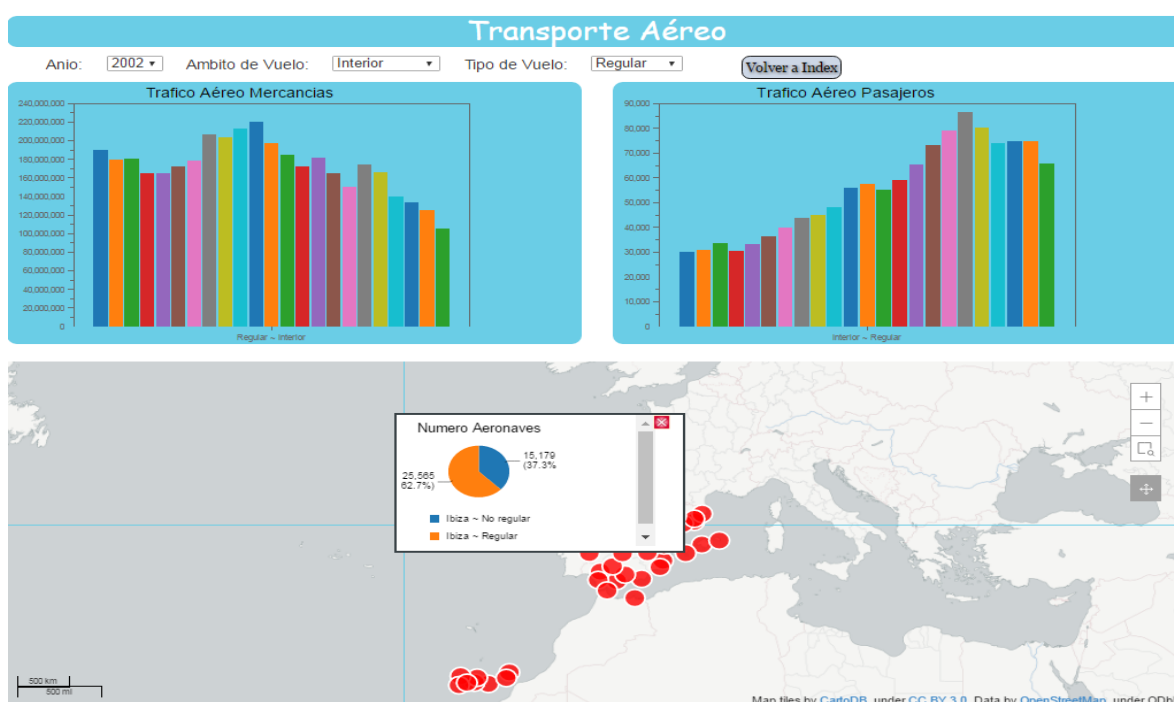


IMAGEN 82 DASHBOARD TRANSPORTE AÉREO

La imagen 84 muestra un Dashboard con información de transporte por Carretera. En la parte superior solo tendremos el botón para volver al Dashboard principal y un selector que nos permitirá filtrar por año.

El primer gráfico es un gráfico *Hear Grid* y nos muestra la cantidad de vehículos que están autorizados a transportar mercancías. Se trata de un gráfico estático, el eje (x) muestra los años y el eje (y) muestra la comunidad autónoma correspondiente.

El gráfico tomará distintos colores según la cantidad de vehículos, el color verde indica que en esa comunidad autónoma hay muchos vehículos autorizados, mientras que el rojo indica que hay pocos.

Por último el gráfico circular muestra los kilómetros de carretera pertenecientes a cada comunidad autónoma, como es lógico las comunidades autónomas más grandes tendrán más kilómetros de carreteras.

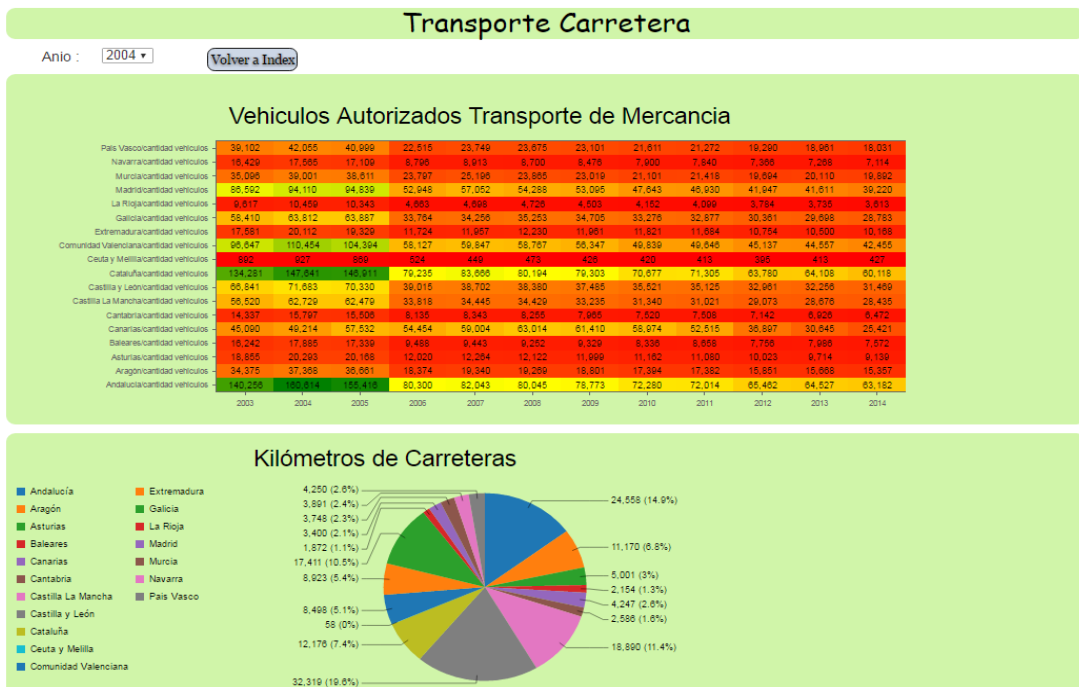


IMAGEN 83 DASHBOARD TRANSPORTE CARRETERA

La imagen 85 muestra un Dashboard que es muy interesante ya que trata el tema del consumo de energía y la contaminación en el ámbito de los transportes.

En la parte superior, al igual que los Dashboards anteriores, están los selectores que nos ayudarán a filtrar la información que se mostrará en las gráficas.

Podremos filtrar la información por modo de transporte, tipo de tráfico, tipo de combustible y tipo de gas contaminante.

Para mostrar la información se ha empleado el gráfico de barras porque resulta más intuitivo para apreciar la evolución del consumo de energía y la contaminación con respecto al tiempo.

El primer gráfico muestra el consumo de energía que supone el modo carretera, tipo de tráfico nacional y combustible gasolina. Se puede observar una tendencia descendente con respecto al tiempo en el consumo de energía.

El segundo gráfico muestra la cantidad de emisiones de gases contaminantes emitidos, esta información al igual que el consumo de energía muestra una tendencia descendente con respecto al tiempo.

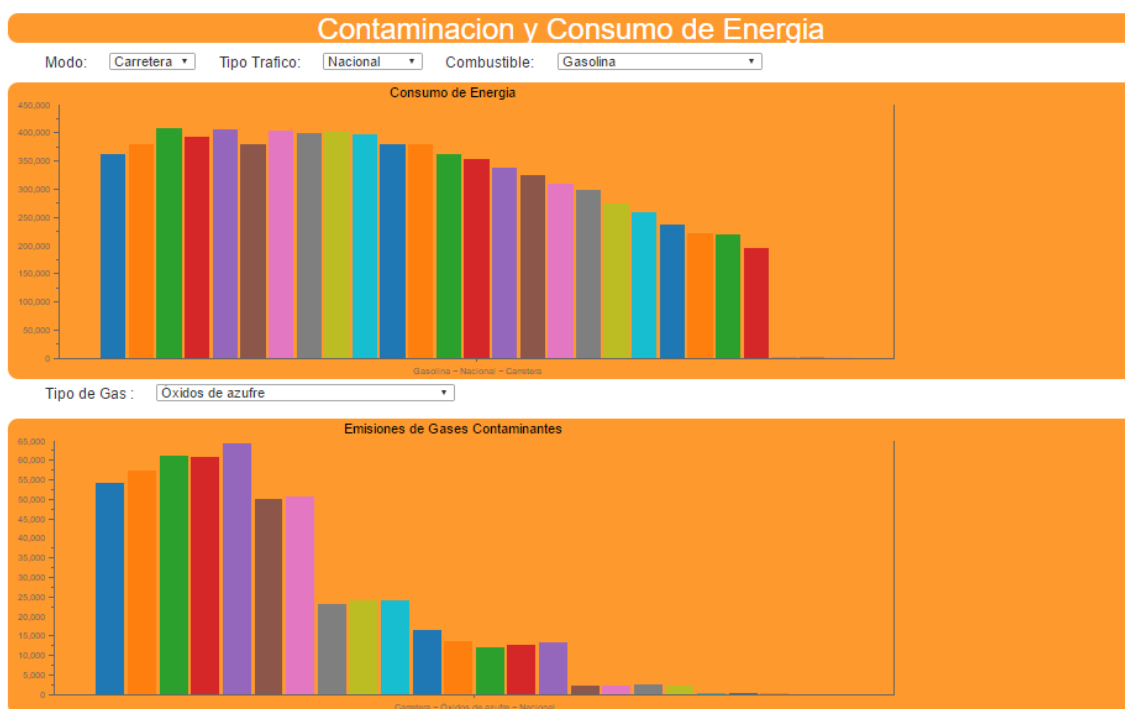


IMAGEN 84 DASHBOARD CONTAMINACIÓN Y CONSUMO DE ENERGÍA

6. Elaboración de Reportes usando la herramienta Pentaho Report Designer

Como ya se ha explicado anteriormente, *Report Designer* es una herramienta de *reporting* fácil de utilizar pero con un gran potencial, ya que se pueden crear informes dinámicos de una manera formal y cuidada para poder presentar al cliente.

Además, esta herramienta permite la incorporación tanto de informes de texto como de gráficos y estos informes se pueden exportar en diferentes formatos como PDF, HTML, XLS, RTF y CSV.

El uso de esta herramienta es muy similar al de las anteriores herramientas. Una vez descargado el programa lo alojaremos en la carpeta /opt como los anteriores y procederemos a ejecutarlo.

Como se puede ver en la Imagen 86, aparecerá la página principal en la que se debe seleccionar Report Wizard.

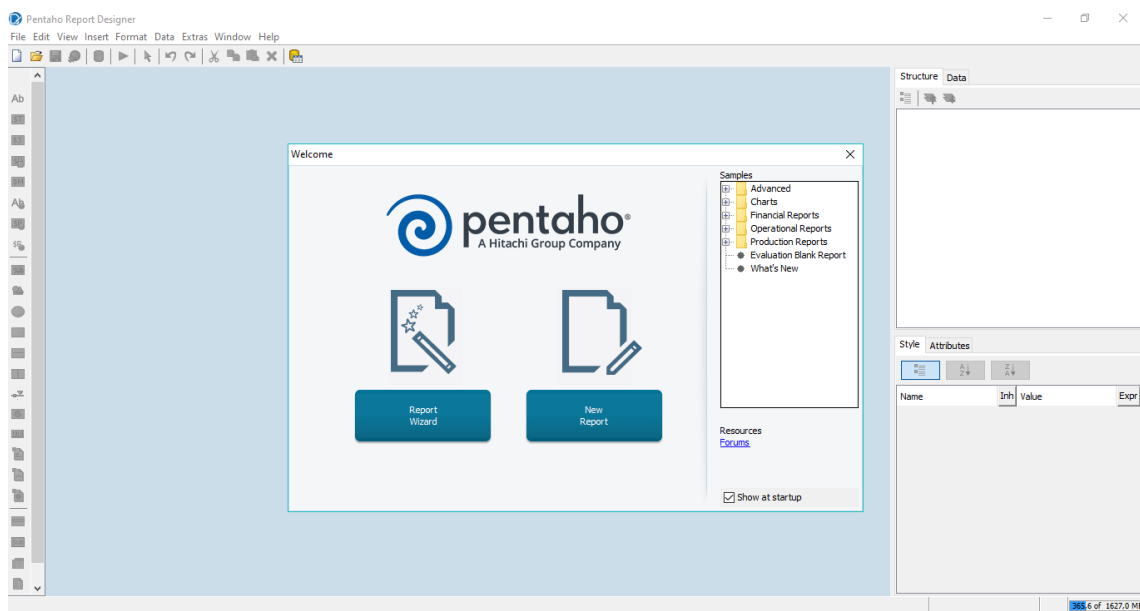


IMAGEN 85 REPORT DESIGNER

La ventana de la Imagen 87 permite elegir una plantilla para nuestro informe.

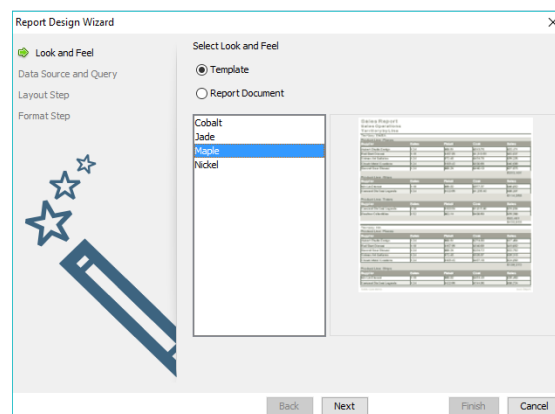


IMAGEN 86 PLANTILLA REPORT DESIGNER

Al igual que en las herramientas anteriores, es necesario añadirle el Data Source del cual se extraerán los datos a analizar. Para ello pulsaremos la opción + y conector JDBC, tal y como se indica en la Imagen 88.

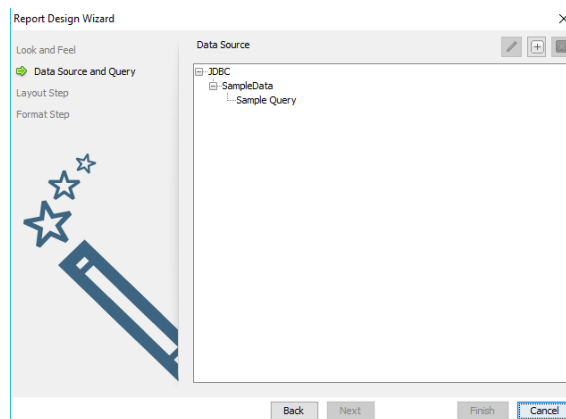


IMAGEN 87 DATA SOURCE

Una vez realizados estos pasos nos aparecerá una ventana (ver Imagen 89) en la que debemos añadir la conexión a nuestro DW pulsando la opción +.

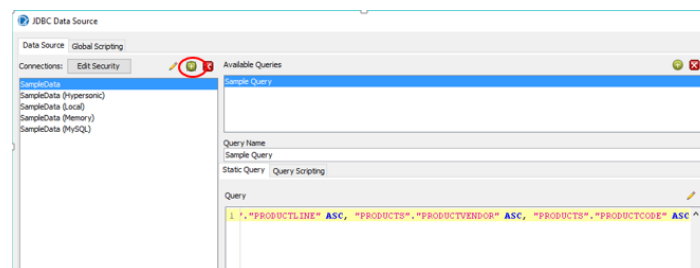


IMAGEN 88 CONEXIÓN A NUESTRA BD

El siguiente paso es rellenar los datos para proceder a establecer la conexión, para comprobar que la conexión se ha establecido correctamente pulsaremos la opción Test.

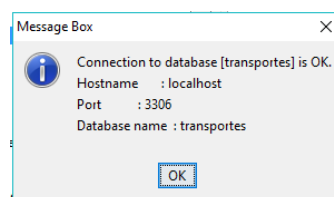


IMAGEN 89 CONEXIÓN CORRECTA

En la Imagen 91 se comprueba que se ha añadido nuestro DW de forma correcta y podemos proceder a añadir todas las consultas que vamos a necesitar utilizando la opción +.

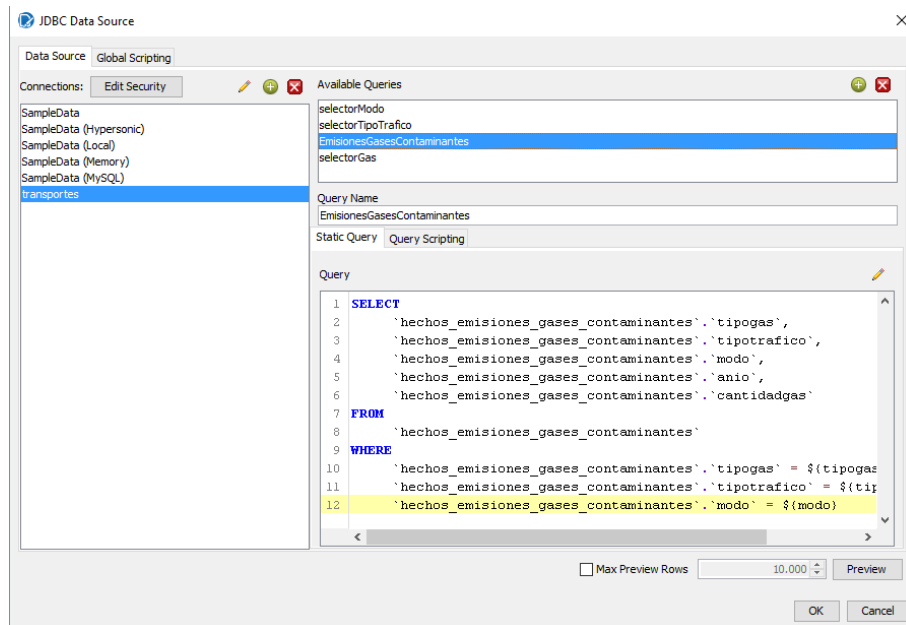


IMAGEN 90 CREACIÓN DE CONSULTAS

Como va a ser un informe dinámico necesitaremos crear selectores y parámetros, de una forma similar a como se realizan en los Dashboards.

La Imagen 92 muestra un ejemplo de selector utilizando un parámetro.

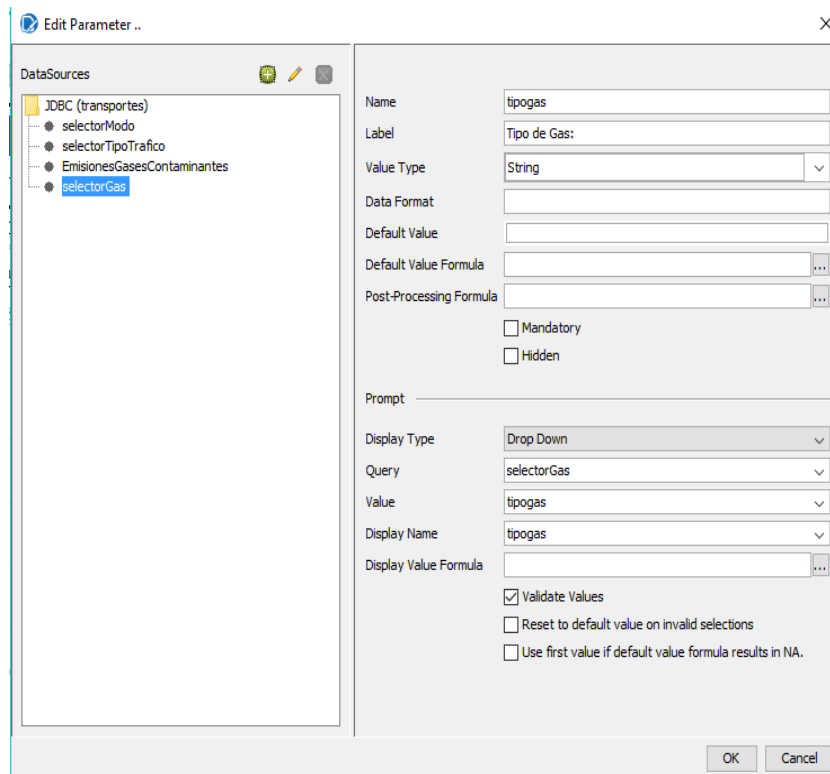


IMAGEN 91 USO DE PARÁMETROS

Con todos los pasos anteriores realizados de manera correcta podemos proceder a la realización de los informes.

Los atributos de la consulta principal en este caso **Emisiones de Gases Contaminantes** se deben colocar en la parte *Details* del informe para que se ejecuten como un bucle.

The screenshot shows a web browser window with the title 'EmisionesGasesConstaminantes'. The page header includes the 'upna' logo and the text 'Análisis Trafico Intermodal' and 'Emisiones Gases Contaminantes'. Below the header is a table with the following structure:

Tipo de Gas	Tipo de Trafico	Modo	Año	Cantidad de Gas
tipogas	tipotrafico	modo	año	cantidadgas

IMAGEN 92 EJEMPLO DE INFORME

Para visualizar el informe debemos seleccionar la imagen del ojo azul y aparecerán los selectores para poder elegir la información que queremos mostrar. La Imagen 94 muestra la visualización del informe.

The screenshot shows a web application interface with the following filter options:

- Tipo de Gas: Óxidos de nitrógeno
- Tipo Trafico: Nacional
- Modo: Carretera
- Auto-Update on selection

Below the filters is a table with the following data:

Tipo de Gas	Tipo de Trafico	Modo	Año	Cantidad de Gas
Óxidos de nitrógeno	Nacional	Carretera	1.990	585.291,324
Óxidos de nitrógeno	Nacional	Carretera	1.991	618.224,2142
Óxidos de nitrógeno	Nacional	Carretera	1.992	659.981,2205
Óxidos de nitrógeno	Nacional	Carretera	1.993	635.804,1217
Óxidos de nitrógeno	Nacional	Carretera	1.994	627.019,0373
Óxidos de nitrógeno	Nacional	Carretera	1.995	586.172,5399
Óxidos de nitrógeno	Nacional	Carretera	1.996	586.379,2242

IMAGEN 93 EJEMPLO DE INFORME

Como ya se ha comentado anteriormente, una opción interesante que tiene esta herramienta de informes es que podemos exportar nuestros informes a distintos formatos pulsando en la opción del triángulo verde (ver Imagen 95).

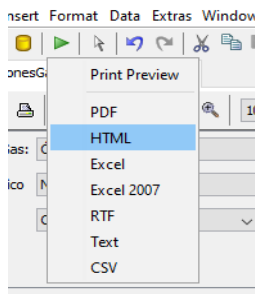


IMAGEN 94 EXPORTAR INFORME

En mi opinión esta herramienta de informes es útil para mostrar la información que tenemos en formato Excel de una manera clara y formal al cliente, pudiendo filtrar como hemos visto la información utilizando los selectores.

Las imágenes 96 y 97 muestran los informes gráficos que se pueden realizar también con esta herramienta.

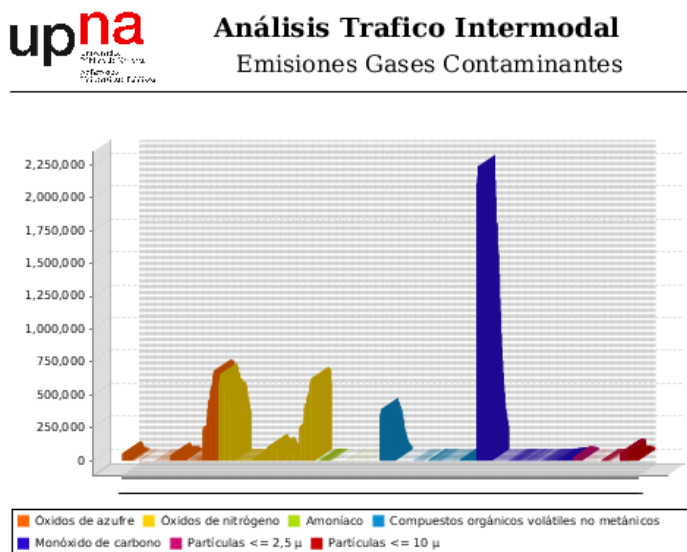


IMAGEN 95 INFORME CON GRÁFICA

Combustible : Gasolina
Tipo de Trafico: Nacional **Modo:** carretera

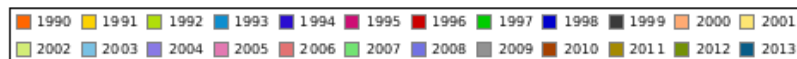
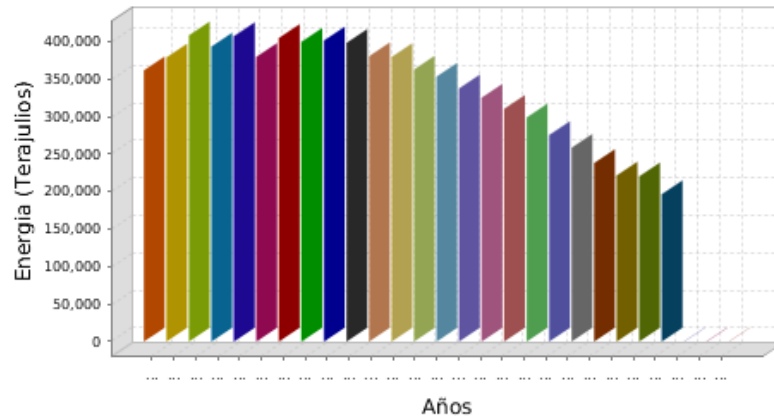
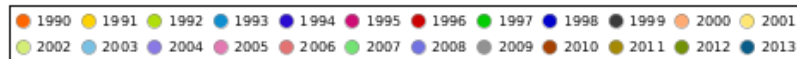
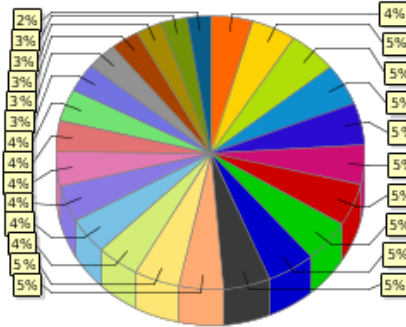


IMAGEN 96 INFORME CON GRÁFICA

7. Conclusiones y reflexiones propias

Hoy en día, las empresas se están moviendo hacia adelante para integrar la inteligencia de negocio (BI) en su infraestructura, debido a que estas herramientas aportan una ventaja competitiva sobre otras empresas que no usan BI. Permiten predecir ventas según informes de años anteriores, conocer beneficio de la empresa fácilmente mediante graficas ingresos – gastos, gestión de la producción de los empleados, y gestión de la inversión, entre otras funciones.

Cabe destacar que las empresas e instituciones que utilizan herramientas de BI obtienen ventajas sobre otras que no utilizan estas herramientas. Algunas de estas ventajas son: agilidad en el proceso de toma de decisiones, eliminación de controles manuales, centralización de la información de gestión, rapidez en la recolección de información. Además en este proyecto todas las herramientas son *Open Source* por lo que su utilización no supone ningún gasto.

El campo de *Business intelligence* y *Big Data* cada día está creciendo más, debido a que las empresas y las organizaciones disponen de mucha información que no saben aprovechar y estas herramientas les ayudan a explotar dicha información para su provecho.

Lo que más me ha llamado la atención de esta herramienta es que se pueden analizar datos de distintos ámbitos, como en mi caso información de transportes, no solo de tipo empresarial.

Tengo que decir que todas las herramientas que he utilizado para realizar este proyecto han sido muy útiles y cada una tenía un gran potencial que al juntarlo al final ofrecen grandes resultados.

Me gustaría prestar especial atención a la herramienta *Pentaho data – integration* utilizada para realizar el proceso ETL que es una parte muy importante del proyecto ya que permite extraer y cargar información. Me ha sorprendido la capacidad que tiene para gestionar la información, pudiendo extraer información de distintas fuentes y formatos juntarla y almacenarla de diferentes formas. Permite recuperar cualquier tipo de información y almacenarla en una BD para hacerla dinámica, sea cual sea su formato.

En general, estoy satisfecho con este proyecto y con la elección del tema, he aprendido mucho durante este tiempo y he tenido la suerte de poder utilizar otros programas de BI en otras asignaturas cosa que me ha ayudado mucho a comprender el funcionamiento.

Para finalizar, me gustaría seguir aprendiendo más cosas de BI pudiendo utilizar este proyecto para seguir mejorándolo ya que durante su desarrollo se me han ocurrido algunas ideas. Se podría utilizar esta misma estructura de Dashboards pero con información de diferentes ámbitos, por ejemplo información de agricultura, medio ambiente, economía de España...

Con este proyecto se podrá acceder a gran cantidad de información mediante un único Dashboard.

8. Bibliografía

- [1] <<Instituto Nacional de Estadística>> Available: <http://www.ine.es/>
- [2] <<Datos Gobierno de España>> Available:
http://observatoriotransporte.fomento.es/OTLE/lang_castellano/
- [3] <<Información>> Available:
http://www.sinnexus.com/business_intelligence/piramide_negocio.aspx
- [4] <<Pentaho Community >> Available: <http://community.pentaho.com/>
- [5] <<MySQL>> Available: <https://www.mysql.com/>
- [6] << Microsoft SQL Server BI >> Available: <https://www.microsoft.com/es-es/sql-server/sql-business-intelligence>
- [7] <<Tableau Business Intelligence>> Available: <https://www.tableau.com/>
- [8] <<SAP Lumira>> Available: <http://saplumira.com/>
- [9] <<Oracle BI>> Available: <https://www.oracle.com/es/solutions/business-analytics/index.html>
- [10] <<Pentaho Enterprise Edition>> Available:
<http://www.pentaho.com/solutions/enterprise>
- [11] <<Pentaho Community Edition>> Available: <http://community.pentaho.com/>
- [12] <<JasperReports>> Available: <http://community.jaspersoft.com/>
- [13] <<Palo>> Available: <https://sourceforge.net/p/palo/code/HEAD/tree/>
- [14] <<SpagoBI>> Available: <https://www.spagobi.org/>
- [15] <<Proxmox>> Available: <https://www.proxmox.com/en/>
- [16] <<phpMyAdmin>> Available: <https://www.phpmyadmin.net/>
- [17] <<MySQL Workbench>> Available: <https://dev.mysql.com/downloads/workbench/>
- [18] <<Java>> Available:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- [19] <<Tomcat>> Available: <https://tomcat.apache.org/>
- [20] <<Apache Software Foundation>> Available : <https://www.apache.org/>
- [21] <<Putty>> Available: <http://www.putty.org/>
- [22] <<X2go>> Available: <http://wiki.x2go.org/doku.php/download:start>
- [23] <<Conector MySQL>> <http://dev.mysql.com/downloads/connector/j/>