

Diseño Técnico

Plataforma BI para seguimiento de flotas

Edición: 1

Trabajo Final de Grado de Ingeniería Informática
Área de Business Intelligence
Jesús Vázquez Galán



Datos del documento	
Nombre	TFG_Diseño_20151029_Ed1
Tipo	Documento de diseño técnico
Fecha entrada en vigor	06/11/2015
Aplicabilidad	

Control de versiones							
Versión	Descripción de los cambios	Modificado por	Fecha	Revisado por	Fecha Revisión	Aprobado por	Fecha Aprobación
1.0	Documento inicial	Jesús Vázquez	29/10/2015	Xavier Martínez Fontes			

ÍNDICE

1	Introducción.....	4
1.1	Objeto.....	4
2	Definición de la Solución Tecnológica	5
2.1	Base de Datos Operacional	5
2.2	Software de BI: Pentaho	7
2.3	ETL	7
2.4	Datawarehouse.....	9
2.5	OLAP	9
2.6	Capa de Negocio	10
3	Modelo Físico de Datos – BBDD Operacional.....	11
4	Modelo Físico de Datos - DW	12
4.1	Modelo Estrella Puntualidad y Carga.....	12
4.2	Modelo Estrella Costes.....	13
4.3	Especificación de datos	13
5	Definición de ETLs.....	23
6	Arquitectura de Sistemas	23
7	Componentes de Cuadro de Mando	24
7.1	Tablas de Datos Dinámicas	24
7.2	Gráficos de sectores	24
7.3	Reloj de Indicador	25

1 INTRODUCCIÓN

1.1 OBJETO

El objeto del presente documento del presente documento es definir la implementación técnica del análisis realizado anteriormente. Para su realización se han tenido en cuenta los documentos previos de análisis y de elección de la solución BI.

En relación al contenido, se describen los siguientes elementos:

- Definición de la solución tecnológica establecida para el proyecto.
- Descripción del modelo físico de datos tanto para la base de datos operacional, como para el Datawarehouse.
- Definición de las transformaciones necesarias.
- Componentes visuales del cuadro de mando.

2 DEFINICIÓN DE LA SOLUCIÓN TECNOLÓGICA

Tomando como base el documento de Análisis, así como el estudio de la Elección de Soluciones BI planteadas para el proyecto, se ha determinado el siguiente esquema de componentes implicados en la solución:



A continuación se relaciona los elementos que componen la solución tecnológica diseñada para el proyecto:

2.1 BASE DE DATOS OPERACIONAL

Toda base de datos de información masiva debe contar con una serie de capacidades, tanto desde la perspectiva de almacenamiento y técnica de indexación, como de distribución de las consultas, escalabilidad y rendimiento en ambientes heterogéneos.

La naturaleza de la información almacenada en este lleva implícita una estructura heterogénea de información y una alta carga de acceso. Los datos almacenados en este tipo de proyectos suelen ser diferentes en los servicios disponibles, e incluso pueden ser cambiantes a lo largo del tiempo dentro de un mismo servicio.

Por estos motivos, **se ha como repositorio de almacenamiento de información transaccional, bases de datos NoSQL, en concreto MongoDB.**

MongoDB es un sistema de base de datos sin esquema (NoSQL) orientado a documentos. Esta base de datos en vez de guardar los datos en tablas como se hace en las base de datos relacionales, guarda estructuras de datos en documentos tipo JSON con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

Las principales características de esta base de datos son las siguientes:

- **Consultas Ad hoc**

Soporta la búsqueda por campos, consultas de rangos y expresiones regulares. Las consultas pueden devolver un campo específico del documento pero también puede ser una función JavaScript definida por el usuario.

- **Indexación**

Cualquier campo en un documento de MongoDB puede ser indexado, al igual que es posible hacer índices secundarios. El concepto de índices en MongoDB es similar a los encontrados en base de datos relacionales.

- **Replicación**

MongoDB soporta el tipo de replicación maestro-esclavo. El maestro puede ejecutar comandos de lectura y escritura. El esclavo puede copiar los datos del maestro y sólo se puede usar para lectura o para copia de seguridad, pero no se pueden realizar escrituras. El esclavo tiene la habilidad de poder elegir un nuevo maestro en caso de que se caiga el servicio con el maestro actual.

- **Balanceo de carga**

MongoDB se puede escalar de forma horizontal usando el concepto de "shard". El desarrollador elige una llave shard, la cual determina cómo serán distribuidos los datos en una colección. Los datos son divididos en rangos (basado en la llave shard) y distribuidos a través de múltiples shard. Un shard es un maestro con uno o más esclavos. MongoDB tiene la capacidad de ejecutarse en múltiple servidores, balanceando la carga y/o duplicando los datos para poder mantener el sistema funcionando en caso que exista un fallo de hardware. La configuración automática es fácil de implementar bajo MongoDB y nuevas máquinas pueden ser agregadas a MongoDB con el sistema de base de datos corriendo.

- **Almacenamiento de archivos**

MongoDB puede ser utilizado con un sistema de archivos, tomando la ventaja de la capacidad que tiene MongoDB para el balanceo de carga y la replicación de datos utilizando múltiples servidores para el almacenamiento de archivos. Esta función (que es llamada GridFS₁₁) está incluida en los drivers de MongoDB y disponible para los lenguajes de programación que soporta MongoDB. Esta base de datos expone funciones para la manipulación de archivos y contenido a los desarrolladores. En un sistema con múltiple servidores, los archivos pueden ser distribuidos y copiados entre los mismos varias veces y de una forma transparente, de esta forma se crea un sistema eficiente que maneja fallos y balanceo de carga.

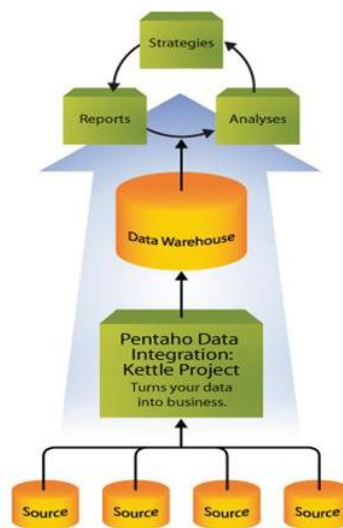
- **Agregación**

La función MapReduce puede ser utilizada para el procesamiento por lotes de datos y operaciones de agregación. Esta función permite que los usuarios puedan obtener el tipo de resultado que se obtiene cuando se utiliza el comando SQL "group-by".

2.2 SOFTWARE DE BI: PENTAHO

En base al estudio realizado para elegir el software de BI (TFG_Estudio_BI_20151013_Ed1), se ha seleccionado Pentaho para la implementación del presente proyecto.

Pentaho Business Analytics es una plataforma completa que cuenta con las herramientas necesarias para dar acceso, integración, visualización y análisis de toda la información, poniendo al alcance de los usuarios las herramientas necesarias para la analítica de negocio y Big Data.



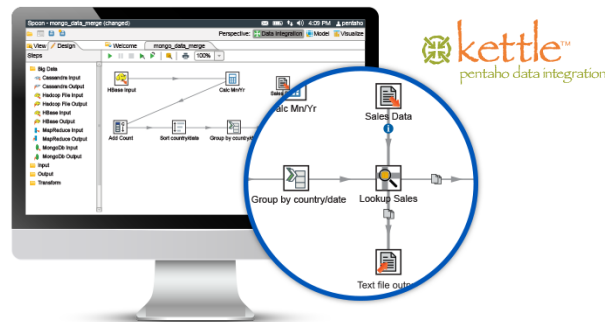
Los componentes de la Suite de Pentaho que se requieren para el proyecto son los siguientes:

- Pentaho Analysis Services (Mondrian)
- Pentaho Reporting
- Pentaho Dashboard
- Pentaho Data Integration (Kettle)

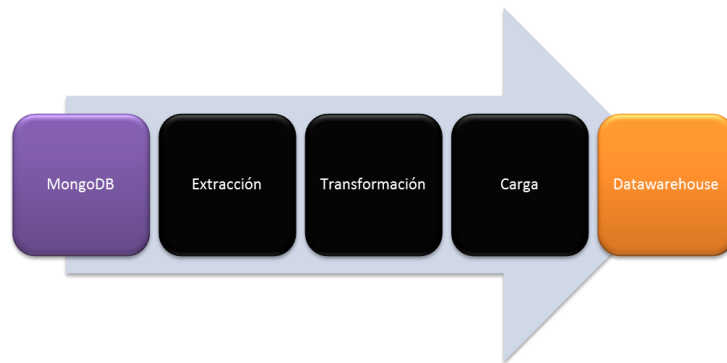
2.3 ETL

La capa de ETL (Extract, Transform and Load) o capa transformación de datos, es la encargada de procesar, limpiar y reorganizar los datos contenidos en la base de datos operacional (MongoDB) para poder ser analizados y estudiados en las capas siguientes.

El producto seleccionado para hacer este proceso es una herramienta denominada Kettle que forma parte de la suite de Pentaho. Este producto contiene todas las herramientas para ejecutar y configurar los procesos ejecutados por esta capa.



En la solución que se propone, esta capa será la encargada de obtener los datos del MongoDB, transformarlos y guardarlos en el Oracle utilizado como datawarehouse.



- **Extracción:**

Este paso es el encargado de extraer u obtener los datos desde el servidor MongoDB. Para ello, Kettle implementa las herramientas necesarias para realizar este proceso. Consta de las herramientas gráficas necesarias para configurar el proceso de extracción, así como un servidor que permite el lanzamiento automático de las tareas a realizar.

Kettle también consta de una gran variedad de conectores que le permiten extraer información de una amplia gama de fuentes de información como bases de datos, archivos, repositorios, directorios, etc.

- **Transformación:**

Una vez obtenidos los datos es necesario transformarlos y adaptarlos para poder ser almacenados en el Datawarehouse (Oracle) siguiendo el modelo de base de datos que tenga en su interior. Para ello, Kettle posee todo un conjunto de herramientas que facilitan el proceso de conversión, agrupación y transformación de los datos.

- **Carga:**

Con los datos transformados ya solo es necesario insertarlos en una base de datos para su posterior análisis. En este caso, Kettle, gracias al gran número de conectores que posee, permite guardar estos datos en una amplia variedad de bases de datos y formatos.

Kettle está compuesto por los siguientes componentes:

- SPOON: entorno gráfico para diseñar transformaciones ETL.
- KITCHEN: para ejecutar trabajos.
- CARTE: Servidor ligero que permite la ejecución en remoto de paquetes ETL.
- ENTERPRISE DATA INTEGRATION SERVER: Servidor que nos permite gestionar la ejecución de los paquetes ETL, administrarlos, versionarlos y supervisarlos.

2.4 DATAWAREHOUSE

El objetivo de esta capa es la de mantener y gestionar los datos elaborados. Esta capa debe tener siempre disponibles los datos almacenados para permitir el análisis de la información. **Para implementar esta capa se ha decidido instalar Oracle XE**, la cual va a ser optimizada para minimizar el tiempo de respuesta de las búsquedas.

En base al análisis realizado con anterioridad a este diseño, se ha definido un esquema en estrella por presentar mejor rendimiento.

2.5 OLAP

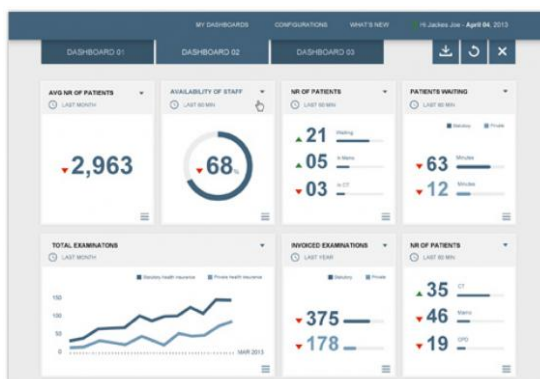
Esta capa es la responsable del análisis de la información que se encuentra contenida en el Datawarehouse. Su nombre viene derivado de las siglas en ingles de On-Line Analytical Processing y como su nombre indica, permite analizar la información on-line mediante las operaciones básicas del análisis: consolidación de la información (roll-up), navegación al detalle (drill-down) y reorganización.

Para proporcionar estas capacidades, es necesario disponer de un software que permita hacer estas operaciones de forma rápida y eficiente. En nuestra solución se ha propuesto la utilización de Mondrian que es un software que se encuentra dentro del conjunto de productos ofrecidos por Pentaho, el cual permite la ejecución de sentencias MDX (lenguaje de consulta para sistemas análisis de información) e implementa los principales interfaces estándares de acceso a la información en sistemas analíticos como XML for Analysis o olap4j.



2.6 CAPA DE NEGOCIO

El cuadro de mando proporciona un punto de acceso a todos los indicadores (KPIs) que permitirá identificar y analizar el estado de la red en cualquiera de los aspectos. Para la implementación de este cuadro de mando se hará uso del Pentaho Platform BI que proporciona un Framework con todos los componentes necesarios para la creación y representación de todos los indicadores.

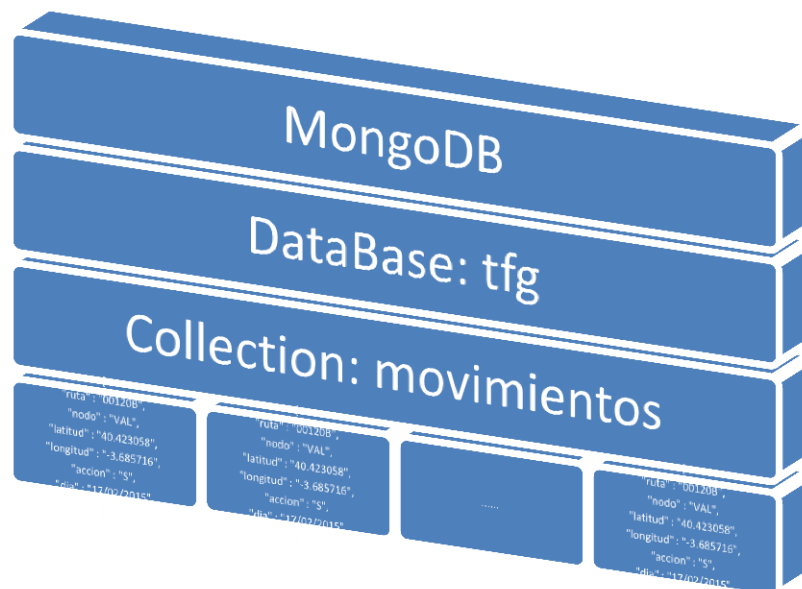


Por otro lado, este producto permite el análisis de la información gracias a que posee tablas dinámicas que permiten la creación de consultas dinámicas contra OLAP con una interfaz gráfica intuitiva, así como permite al usuario realizar on-line las operaciones básicas de análisis como la reordenación, la agrupación y segregación de la información proporcionando al usuario una buena experiencia de utilización. Algunos de estos componentes son:

- **Pivot4J:** Este componente de Pentaho es una tabla dinámica que permite la creación de queries MDX gráficamente al igual que permite bucear en la información mostrada. Este componente es especialmente útil para el diseño de queries a medida y el análisis de la información.
- **Saiku:** Este componente es de Pentaho, al igual que Pivot4J, permite generar queries dinámicas aunque está enfocado a un análisis gráfico de la información. Este permite cambiar rápidamente la representación de los gráficos y obtener el modelo más adecuado e intuitivo para su representación.

3 MODELO FÍSICO DE DATOS – BBDD OPERACIONAL

El arquitectura que deberá tener la base de datos operacional será la siguiente:



Como ya se ha indicado, MongoDB es una base de datos orientada a documentos y libre de esquema.

Como se puede observar en el esquema, la base de datos se denominada *tfg* (siglas de Trabajo Fin de Grado). Esta base de datos contendrá una colección de datos denominada *movimientos*. Dentro de la colección *movimientos* se encuentran todos los JSON correspondientes a los diferentes movimientos.

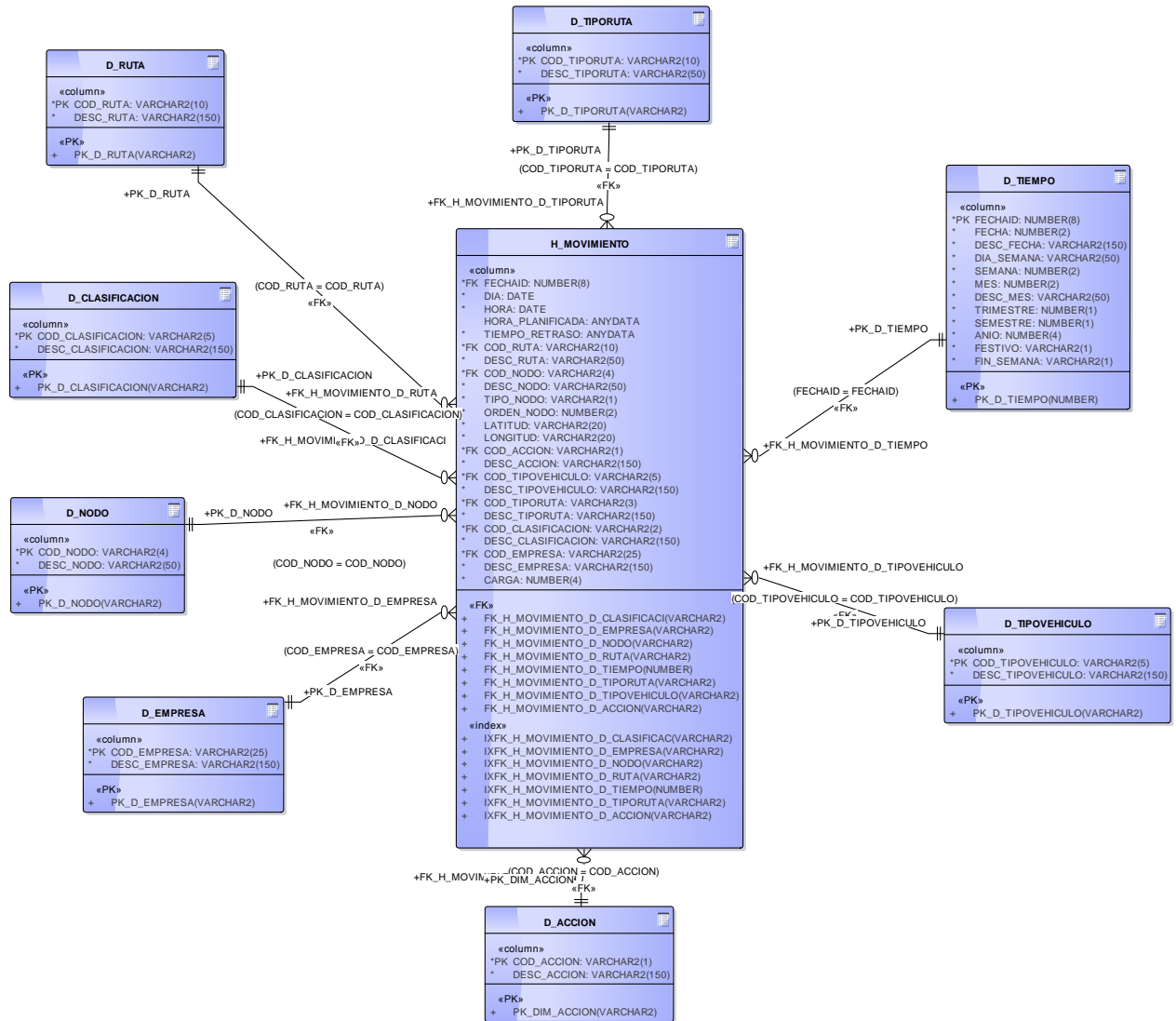
Para el presente proyecto, los datos provenientes del GPS, y que serán simulados por los programas de carga, se almacenarán en MongoDB. La estructura JSON definida para los eventos de GPS es la siguiente:

```
{
  "_id" : ObjectId("562943ebcc5876b6a3f07469"),
  "ruta" : "00120B",
  "nodo" : "VAL",
  "latitud" : "40.423058",
  "longitud" : "-3.685716",
  "accion" : "S",
  "hora" : "03:31",
  "dia" : "17/02/2015"
}
```

4 MODELO FÍSICO DE DATOS - DW

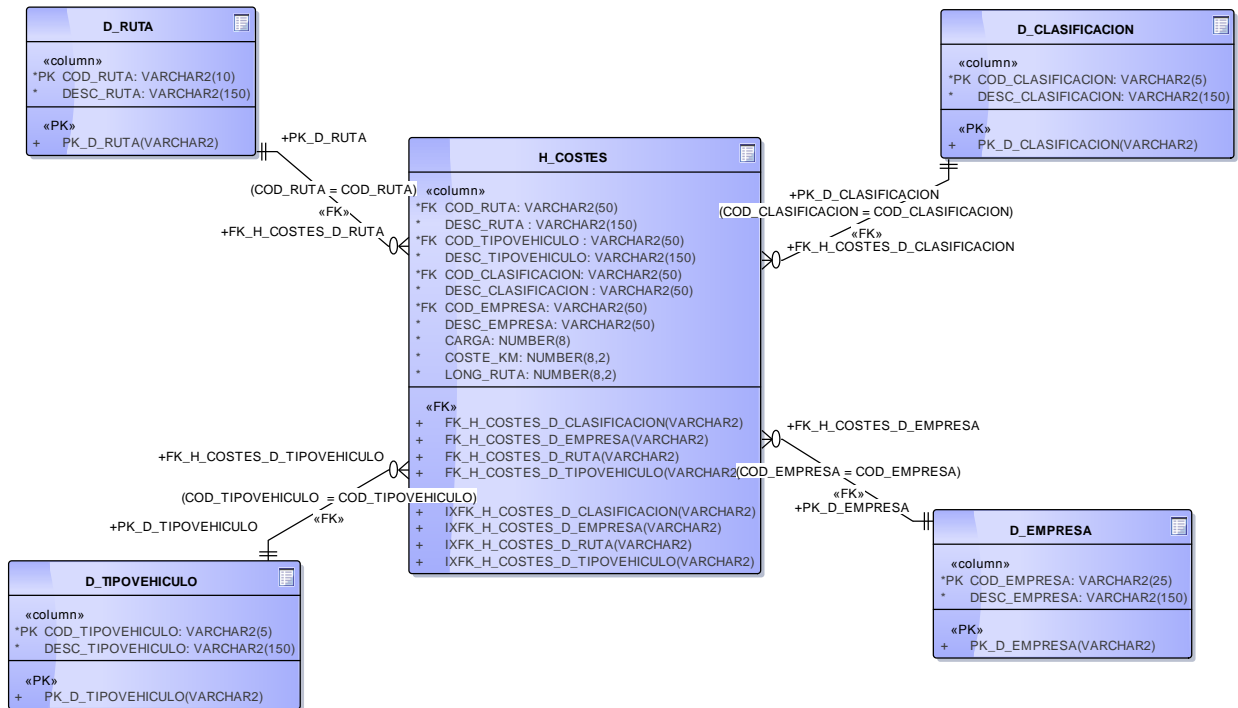
4.1 MODELO ESTRELLA PUNTUALIDAD Y CARGA

Diagrama: Modelo Estrella Puntualidad y Carga



4.2 MODELO ESTRELLA COSTES

Diagrama: **Modelo Estrella Costes**





4.3 ESPECIFICACIÓN DE DATOS

4.3.1 *D_ACCION* - «table»

Versión: 1.0 // Status: Proposed

Tabla correspondiente a la dimensión de la acción (Entrada, Salida o Parada)






ATRIBUTOS / COLUMNAS	DESCRIPCIÓN
<p>◆ COD_ACCION (VARCHAR2)</p>	Código correspondiente a la acción (E, S, P)
<p>◆ DESC_ACCION (VARCHAR2)</p>	Texto descriptivo de la acción (Entrada, Salida, Parada)
OPERACIONES	DESCRIPCIÓN

 PK_DIM_ACCION (VARCHAR2 COD_ACCION,)		Clave principal de la tabla
RELACIONES DEL ELEMENTO		
H_MOVIMIENTO «table»	 ForeignKey «FK»	D_ACCION «table»

4.3.2 D_CLASIFICACION - «table»

Versión: 1.0 // Status: Proposed

Tabla correspondiente a la dimensión Clasificación

ATRIBUTOS / COLUMNAS	DESCRIPCIÓN	
 COD_CLASIFICACION (VARCHAR2)	Código correspondiente al tipo de clasificación	
 DESC_CLASIFICACION (VARCHAR2)	Texto descriptivo correspondiente a la clasificación (aéreo, marítimo, terrestre)	
OPERACIONES	DESCRIPCIÓN	
 PK_D_CLASIFICACION (VARCHAR2 COD_CLASIFICACION,)	Clave principal de la tabla	
RELACIONES DEL ELEMENTO		
H_COSTES «table»	 ForeignKey «FK»	D_CLASIFICACION «table»
H_MOVIMIENTO «table»	 ForeignKey «FK»	D_CLASIFICACION «table»

4.3.3 D_EMPRESA - «table»

Versión: 1.0 // Status: Proposed

Tabla correspondiente a la dimensión Empresa

ATRIBUTOS / COLUMNAS	DESCRIPCIÓN
 COD_EMPRESA (VARCHAR2)	Código de la empresa, correspondiente al

		NIF
◆ DESC_EMPRESA (VARCHAR2)		Texto descriptivo de la empresa (razón social)
OPERACIONES		DESCRIPCIÓN
◆ PK_D_EMPRESA (VARCHAR2 COD_EMPRESA,)		Clave principal de la tabla
RELACIONES DEL ELEMENTO		
H_COSTES «table»	⇒ ForeignKey «FK»	D_EMPRESA «table»
H_MOVIMIENTO «table»	⇒ ForeignKey «FK»	D_EMPRESA «table»

4.3.4 D_NODO - «table»

Versión: 1.0 // Status: Proposed

Tabla correspondiente a la dimensión Nodo

ATRIBUTOS / COLUMNAS		DESCRIPCIÓN
◆ COD_NODO (VARCHAR2)		Código del nodo
◆ DESC_NODO (VARCHAR2)		Texto descriptivo del nodo
OPERACIONES		DESCRIPCIÓN
◆ PK_D_NODO (VARCHAR2 COD_NODO,)		Clave principal de la tabla
RELACIONES DEL ELEMENTO		
H_MOVIMIENTO «table»	⇒ ForeignKey «FK»	D_NODO «table»

4.3.5 D_RUTA - «table»

Versión: 1.0 // Status: Proposed

Tabla correspondiente a la dimensión ruta

ATRIBUTOS / COLUMNAS		DESCRIPCIÓN
◆ COD_RUTA (VARCHAR2)		Código correspondiente a la ruta
◆ DESC_RUTA (VARCHAR2)		Texto descriptivo de la ruta
OPERACIONES		DESCRIPCIÓN
◆ PK_D_RUTA (VARCHAR2 COD_RUTA,)		Clave principal de la tabla
RELACIONES DEL ELEMENTO		
H_COSTES «table»	ForeignKey «FK»	D_RUTA «table»
H_MOVIMIENTO «table»	ForeignKey «FK»	D_RUTA «table»

4.3.6 D_TIEMPO - «table»

Versión: 1.0 // Status: Proposed

Tabla correspondiente a la dimensión Tiempo

ATRIBUTOS / COLUMNAS	DESCRIPCIÓN
◆ FECHAID (NUMBER)	Pk de la tabla de dimensión tiempo (20151010->10/10/2015)
◆ FECHA (NUMBER)	Fecha
◆ DESC_FECHA (VARCHAR2)	Fecha en texto
◆ DIA_SEMANA (VARCHAR2)	Texto de día correspondiente a la semana (Lunes, Martes, ...)
◆ SEMANA (NUMBER)	Número de semana dentro del año
◆ MES (NUMBER)	Número del mes
◆ DESC_MES (VARCHAR2)	
◆ TRIMESTRE (NUMBER)	Número de trimestre
◆ SEMESTRE (NUMBER)	Número de semestre
◆ ANIO (NUMBER)	Año

◆ FESTIVO (VARCHAR2)	Flag que indica si el día es festivo	
◆ FIN_SEMANA (VARCHAR2)	Flag que indica si el día era fin de semana	
OPERACIONES		
DESCRIPCIÓN		
◆ PK_D_TIEMPO (NUMBER FECHAID,)	Clave principal de la tabla	
RELACIONES DEL ELEMENTO		
H_MOVIMIENTO «table»	⇨ ForeignKey «FK»	D_TIEMPO «table»

4.3.7 D_TIPORUTA - «table»

Versión: 1.0 // Status: Proposed

Tabla correspondiente a la dimensión de tipo de ruta (ej: INTERZONAL, ZONAL, PROVINCIAL)





ATRIBUTOS / COLUMNAS		DESCRIPCIÓN
◆ COD_TIPORUTA (VARCHAR2)	Código correspondiente al tipo de ruta	
◆ DESC_TIPORUTA (VARCHAR2)	Texto descriptivo correspondiente al tipo de ruta	
OPERACIONES		DESCRIPCIÓN
◆ PK_D_TIPORUTA (VARCHAR2 COD_TIPORUTA,)	Clave principal de la tabla	
RELACIONES DEL ELEMENTO		
H_MOVIMIENTO «table»	⇨ ForeignKey «FK»	D_TIPORUTA «table»

4.3.8 D_TIPOVEHICULO - «table»

Versión: 1.0 // Status: Proposed

Tabla correspondiente al tipo de vehículo (ej: TRAILER, CAMION, FURGONETA)






ATRIBUTOS / COLUMNAS		DESCRIPCIÓN
◆ COD_TIPOVEHICULO (VARCHAR2)	Código correspondiente al tipo de vehículo	

 DESC_TIPOVEHICULO (VARCHAR2)		Texto descriptivo correspondiente al tipo de vehículo
OPERACIONES		DESCRIPCIÓN
 PK_D_TIPOVEHICULO (VARCHAR2 COD_TIPOVEHICULO,)		Clave principal de la tabla
RELACIONES DEL ELEMENTO		
H_MOVIMIENTO «table»	 ForeignKey «FK»	D_TIPOVEHICULO «table»
H_COSTES «table»	 ForeignKey «FK»	D_TIPOVEHICULO «table»

4.3.9 H_COSTES - «table»

Versión: 1.0 // Status: Proposed

Tabla que modela la tabla de hechos correspondiente a los costes

ATRIBUTOS / COLUMNAS	DESCRIPCIÓN
 COD_RUTA (VARCHAR2)	Código de la Ruta a la que corresponde el movimiento
 DESC_RUTA (VARCHAR2)	Texto descriptivo de la ruta
 COD_TIPOVEHICULO (VARCHAR2)	Código correspondiente al tipo de vehículo
 DESC_TIPOVEHICULO (VARCHAR2)	Texto correspondiente al tipo de vehículo
 COD_CLASIFICACION (VARCHAR2)	Código correspondiente a la clasificación de la ruta
 DESC_CLASIFICACION (VARCHAR2)	Descripción correspondiente a la clasificación (aéreo, marítimo o terrestre)
 COD_EMPRESA (VARCHAR2)	Código identificativo de la empresa (CIF)
 DESC_EMPRESA (VARCHAR2)	Razón social de la empresa
 CARGA (NUMBER)	Carga que transportaba la ruta en Kg

◆ COSTE_KM (NUMBER)	Coste por Km de la ruta	
◆ LONG_RUTA (NUMBER)	Longitud de la ruta en Km	
OPERACIONES		
◆ IXFK_H_COSTES_D_EMPRESA (VARCHAR2 COD_EMPRESA,)		
◆ IXFK_H_COSTES_D_RUTA (VARCHAR2 COD_RUTA,)		
◆ FK_H_COSTES_D_CLASIFICACION (VARCHAR2 COD_CLASIFICACION,)		
◆ FK_H_COSTES_D_EMPRESA (VARCHAR2 COD_EMPRESA,)		
◆ FK_H_COSTES_D_TIPOVEHICULO (VARCHAR2 COD_TIPOVEHICULO,)		
◆ IXFK_H_COSTES_D_TIPOVEHICULO (VARCHAR2 COD_TIPOVEHICULO,)		
◆ FK_H_COSTES_D_RUTA (VARCHAR2 COD_RUTA,)		
◆ IXFK_H_COSTES_D_CLASIFICACION (VARCHAR2 COD_CLASIFICACION,)		
RELACIONES DEL ELEMENTO		
H_COSTES «table»	⇒ ForeignKey «FK»	D_CLASIFICACION «table»
H_COSTES «table»	⇒ ForeignKey «FK»	D_RUTA «table»
H_COSTES «table»	⇒ ForeignKey «FK»	D_EMPRESA «table»
H_COSTES «table»	⇒ ForeignKey «FK»	D_TIPOVEHICULO «table»

4.3.10 H_MOVIMIENTO - «table»

Versión: 1.0 // Status: Proposed

Tabla que modela la tabla de hechos correspondiente a los movimientos de los vehículos de transporte

ATRIBUTOS / COLUMNAS	DESCRIPCIÓN
◆ FECHAID (NUMBER)	Fecha en formato numérico
◆ DIA (DATE)	Día en el que se produce el movimiento
◆ HORA (DATE)	Hora en la que se produce el movimiento
◆ HORA_PLANIFICADA (ANYDATA)	Hora a la que estaba planificado el evento de entrada o salida
◆ TIEMPO_RETRASO (ANYDATA)	Tiempo de retraso ya sea a la entrada o a la salida del nodo
◆ COD_RUTA (VARCHAR2)	Código de la Ruta a la que corresponde el movimiento
◆ DESC_RUTA (VARCHAR2)	Texto descriptivo de la ruta
◆ COD_NODO (VARCHAR2)	Código del nodo sobre el que se realiza el registro de entrada o salida
◆ DESC_NODO (VARCHAR2)	Texto descriptivo del nodo
◆ TIPO_NODO (VARCHAR2)	Indica si el nodo es inicial, final o intermedio
◆ ORDEN_NODO (NUMBER)	Orden del nodo dentro de la ruta
◆ LATITUD (VARCHAR2)	Latitud en la que se registra el evento, asociada al ámbito de un nodo
◆ LONGITUD (VARCHAR2)	Latitud en la que se registra el evento, asociada al ámbito de un nodo
◆ COD_ACCION (VARCHAR2)	Código de la acción asociada al evento
◆ DESC_ACCION (VARCHAR2)	Descripción de la acción: E-> Entrada, S->Salida, P->Parada
◆ COD_TIPOVEHICULO (VARCHAR2)	Código correspondiente al tipo de vehículo que realiza el movimiento
◆ DESC_TIPOVEHICULO (VARCHAR2)	Texto correspondiente al tipo de vehículo
◆ COD_TIPORUTA (VARCHAR2)	Código correspondiente al tipo de ruta

◆ DESC_TIPORUTA (VARCHAR2)	Texto descriptivo correspondiente al tipo de ruta
◆ COD_CLASIFICACION (VARCHAR2)	Código correspondiente a la clasificación de la ruta
◆ DESC_CLASIFICACION (VARCHAR2)	Descripción correspondiente a la clasificación (aéreo, marítimo o terrestre)
◆ COD_EMPRESA (VARCHAR2)	Código identificativo de la empresa (CIF)
◆ DESC_EMPRESA (VARCHAR2)	Razón social de la empresa
◆ CARGA (NUMBER)	Carga que transportaba la ruta en Kg
OPERACIONES	DESCRIPCIÓN
◆ IXFK_H_MOVIMIENTO_D_ACCION (VARCHAR2 COD_ACCION,)	Índice de la dimensión ACCION
◆ IXFK_H_MOVIMIENTO_D_TIEMPO (NUMBER FECHAID,)	Índice de la dimensión TIEMPO
◆ FK_H_MOVIMIENTO_D_TIEMPO (NUMBER FECHAID,)	FK de la tabla TIEMPO
◆ FK_H_MOVIMIENTO_D_ACCION (VARCHAR2 COD_ACCION,)	FK de la tabla ACCION
◆ FK_H_MOVIMIENTO_D_EMPRESA (VARCHAR2 COD_EMPRESA,)	FK de la tabla EMPRESA
◆ IXFK_H_MOVIMIENTO_D_CLASIFICAC (VARCHAR2 COD_CLASIFICACION,)	Índice de la dimensión CLASIFICACION
◆ FK_H_MOVIMIENTO_D_CLASIFICACI (VARCHAR2 COD_CLASIFICACION,)	FK de la tabla CLASIFICACION
◆ IXFK_H_MOVIMIENTO_D_NODO (VARCHAR2 COD_NODO,)	Índice de la dimensión NODO
◆ FK_H_MOVIMIENTO_D_NODO (VARCHAR2 COD_NODO,)	FK de la tabla NODO
◆ FK_H_MOVIMIENTO_D_TIPOVEHICULO (VARCHAR2 COD_TIPOVEHICULO,)	FK de la tabla TIPOVEHICULO
◆ IXFK_H_MOVIMIENTO_D_TIPORUTA (VARCHAR2 COD_TIPORUTA,)	Índice de la dimensión TIPORUTA

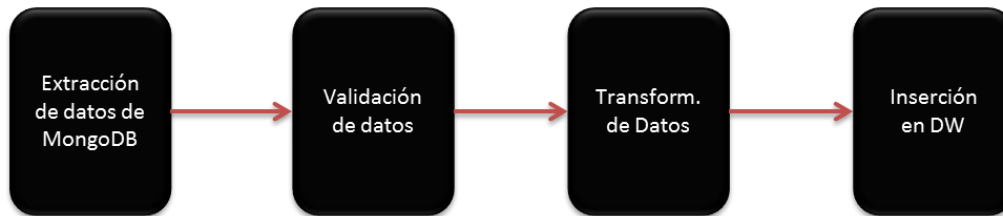
<p>◆ FK_H_MOVIMIENTO_D_TIPORUTA (VARCHAR2 COD_TIPORUTA,)</p>	FK de la tabla TIPORUTA
<p>◆ IXFK_H_MOVIMIENTO_D_RUTA (VARCHAR2 COD_RUTA,)</p>	Índice de la dimensión RUTA
<p>◆ FK_H_MOVIMIENTO_D_RUTA (VARCHAR2 COD_RUTA,)</p>	FK de la tabla RUTA
<p>◆ IXFK_H_MOVIMIENTO_D_EMPRESA (VARCHAR2 COD_EMPRESA,)</p>	Índice de la dimensión EMPRESA

RELACIONES DEL ELEMENTO

<p>H_MOVIMIENTO «table»</p>	<p>⇒ ForeignKey «FK»</p>	<p>D_TIPOVEHICULO «table»</p>
<p>H_MOVIMIENTO «table»</p>	<p>⇒ ForeignKey «FK»</p>	<p>D_TIEMPO «table»</p>
<p>H_MOVIMIENTO «table»</p>	<p>⇒ ForeignKey «FK»</p>	<p>D_RUTA «table»</p>
<p>H_MOVIMIENTO «table»</p>	<p>⇒ ForeignKey «FK»</p>	<p>D_TIPORUTA «table»</p>
<p>H_MOVIMIENTO «table»</p>	<p>⇒ ForeignKey «FK»</p>	<p>D_EMPRESA «table»</p>
<p>H_MOVIMIENTO «table»</p>	<p>⇒ ForeignKey «FK»</p>	<p>D_NODO «table»</p>
<p>H_MOVIMIENTO «table»</p>	<p>⇒ ForeignKey «FK»</p>	<p>D_ACCION «table»</p>
<p>H_MOVIMIENTO «table»</p>	<p>⇒ ForeignKey «FK»</p>	<p>D_CLASIFICACION «table»</p>

5 DEFINICIÓN DE ETLs

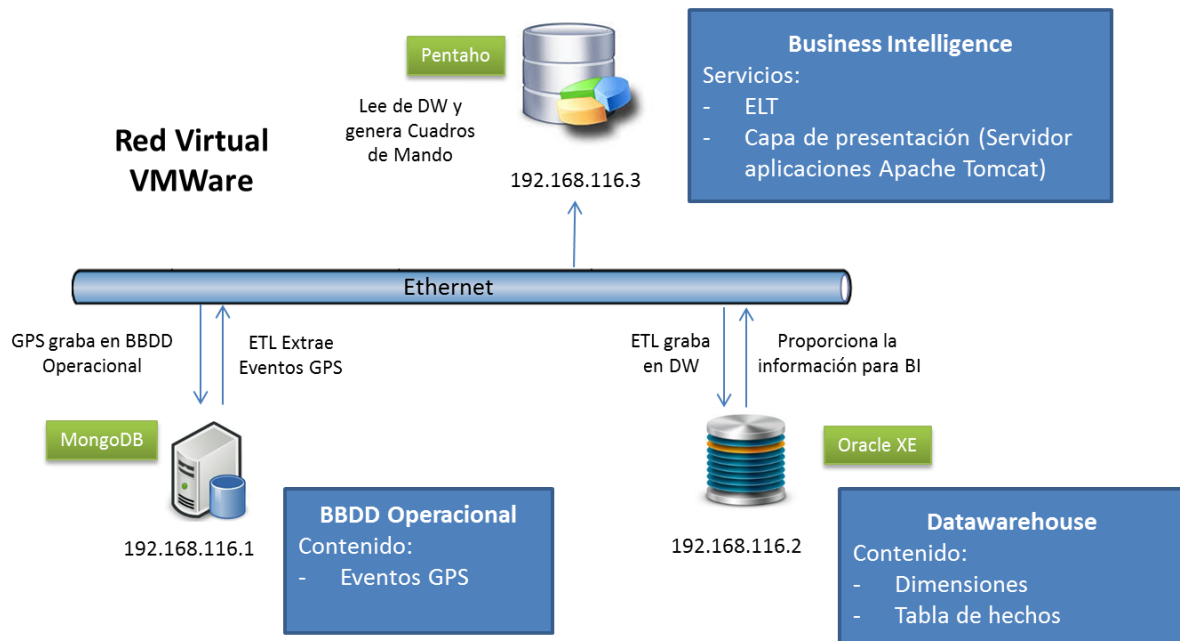
El proceso de ETL necesario para la consecución del proyecto debe realizar las siguientes acciones:



- Extracción de los datos de la base de datos de MongoDB. Para ello, se deberá disponer del driver jdbc para MongoDB.
- Validación de la información para cumplir las restricciones de datos del DW.
- Transformación de la información en donde se adaptan los tipos de datos extraídos del MongoDB, y combinados con información de la estructura de la red.
- Inserción en el datawarehouse para su posterior explotación.

6 ARQUITECTURA DE SISTEMAS

Para la realización del proyecto se ha establecido una arquitectura de sistemas basada en máquinas virtuales.



7 COMPONENTES DE CUADRO DE MANDO

Los diferentes cuadros de mando estarán diseñados bajo las herramientas de Pentaho, y estarán formados por los siguientes componentes.

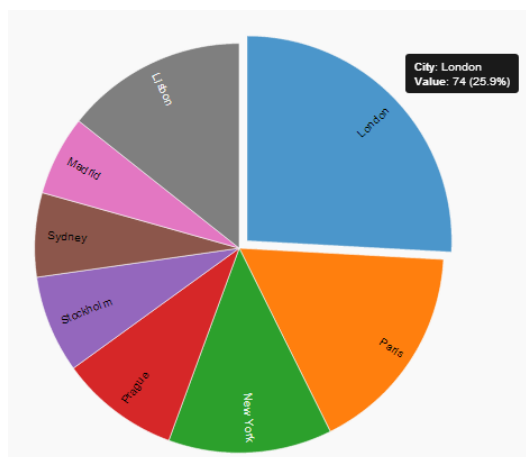
7.1 TABLAS DE DATOS DINÁMICAS

Este componente de Pentaho es una tabla dinámica que permite la creación de queries MDX gráficamente al igual que permite bucear en la información mostrada. Este componente es especialmente útil para el diseño de queries a medida y el análisis de la información

Product	Time								
	All Years	2003	2004	QTR1	QTR2	QTR3	QTR4	2005	
All Products	105,331	36,439	49,417	8,694	8,443	11,311	20,969	19,475	
Classic Cars	35,552	12,762	16,085	2,916	2,593	3,947	6,629	6,705	
Autoart Studio Design	933	337	416	68	52	115	181	180	
Carousel DieCast Legends	2,895	1,043	1,320	216	222	311	571	532	
1958 Chevy Corvette Limited Edition	983	338	453	54	80	97	222	192	
1966 Shelby Cobra 427 S/C	915	357	387	81	53	113	140	171	
1982 Camaro Z28	997	348	480	81	89	101	209	169	
Classic Metal Creations	5,830	2,056	2,601	465	454	629	1,053	1,173	
Exoto Designs	2,610	983	1,183	209	198	338	438	444	
Gearbox Collectibles	4,715	1,746	2,147	399	373	547	828	822	
Highway 66 Mini Classics	1,923	639	930	123	139	210	458	354	

MDX Query: `SELECT Hierarchize(Union(Union([Time].[All Years],[Time].[All Years].Children],[Time].[2004].Children)) ON COLUMNS, Hierarchize(Union(Union([Product].[All Products],[Product].[All Products].Children],[Product].[Classic Cars].Children)) ON ROWS FROM [SteelWheelsSales])`

7.2 GRÁFICOS DE SECTORES



7.3 RELOJ DE INDICADOR

