

GRADO EN INGENIERÍA INFORMÁTICA DE GESTIÓN Y SISTEMAS DE INFORMACIÓN

DESARROLLO DE UN SISTEMA DE BI CON PENTAHO

TRABAJO FIN DE GRADO

DOCUMENTO 1- MEMORIA

Alumno: Bahillo Gil, Xabier
Directora: Blanco Jauregui, Begoña

Curso 2020-2021

1. Resumen

Durante este proyecto se ha implementado un sistema de Business Intelligence haciendo uso de la plataforma Pentaho, con el objetivo de explotar los datos obtenidos principalmente de un sistema de gestión de almacenes y extender las funcionalidades ofrecidas en este.

Este documento corresponde con la memoria del Trabajo de Fin De Grado de Xabier Bahillo, desarrollado para la titulación Ingeniería Informática de Gestión y Sistemas de Información en la Escuela de Ingeniería de Bilbao de la UPV.

Índice

1. Resumen	1
2. Introducción	9
2.1. Descripción del Proyecto	9
2.2. Motivación	9
2.3. Definiciones	10
3. Planteamiento inicial	11
3.1. Objetivos	11
3.1.1. Objetivos del proyecto	11
3.1.2. Objetivos personales	11
3.2. Marco del proyecto	11
3.3. Alcance	12
3.3.1. Fases	12
3.3.2. Ciclo de Vida	12
3.4. Herramientas	13
3.4.1. Hardware	13
3.4.2. Software	13
3.5. Planificación Temporal	15
3.5.1. EDT	15
3.5.2. Listado de Tareas	16
3.5.3. Gantt	22
3.6. Gestión de Riesgos	24
3.7. Planificación Económica	27
3.7.1. Gastos	27
3.7.2. Beneficios	28
4. Antecedentes	29
4.1. Situación actual	29
4.2. Definición de Business Intelligence	29
4.3. Componentes de un Business Intelligence	30
4.3.1. Datawarehouse	30
4.3.2. Cubos OLAP	31
4.3.3. Cuadros de mando	32
4.4. Análisis de Herramientas	32
4.4.1. Pentaho CE	33
5. Captura de requisitos	35
5.1. Requisitos funcionales	35
5.2. Requisitos no funcionales	35
5.3. Jerarquía de actores	36
5.4. Casos de uso	37
6. Análisis y diseño	39
6.1. Análisis de la base de datos origen	39
6.2. Diseño del Datawarehouse	42
6.3. Diseño de los cubos OLAP	50
6.4. Diseño de los informes predefinidos	58

7. Desarrollo	62
7.1. Creación del esquema de base de datos	62
7.2. Implementación del proceso ETL	64
7.3. Implementación de los cubos OLAP	80
7.4. Implementación de informes predefinidos	84
7.5. Despliegue de la aplicación web	87
7.5.1. Conexión de la aplicación con el Datawarehouse	88
7.5.2. Despliegue de los cubos OLAP	89
7.5.3. Despliegue de la herramienta para consultas ad-hoc	91
7.5.4. Creación de metadatos para consultas ad-hoc	96
7.5.5. Despliegue de informe predefinidos	100
7.5.6. Instalación de la aplicación BI	101
8. Evaluación	104
8.1. Evaluación del proceso ETL	104
8.2. Evaluación medidas cubos OLAP	106
8.3. Evaluación consultas ad-hoc	113
9. Conclusiones y trabajo futuro	117
9.1. Planificación final	117
9.2. Reflexiones personales	119
9.3. Trabajo futuro	119
10. Referencias	120
11. Anexo I: Casos de uso extendidos	123
12. Anexo II: Definición de metadatos	143
12.1. Compras	143
12.2. Ventas	144
12.3. Maestros	146
12.4. Control	147
13. Anexo III: Pruebas unitarias del proceso ETL	149
14. Anexo IV: Manual de usuario	153
14.1. Introducción	153
14.2. Acceso a la aplicación	153
14.3. Consola de usuario	154
14.4. Menú ficheros	156
14.5. Menú programaciones	158
14.6. Menú administración	159
14.7. Consultas ad-hoc	161
14.7.1. Diseñar consulta	162
14.7.2. Aplicar filtro	162
14.8. Consultas OLAP	164
14.9. Informes predefinidos	166

Índice de figuras

1.	Logotipo InLOG Consultoría y Soluciones	11
2.	Ciclo de vida	13
3.	EDT	15
4.	Diagrama Gantt	23
5.	Proceso ETL	30
6.	Estructura Cubo OLAP	31
7.	Diagrama Arquitectura Pentaho	34
8.	Jerarquía de actores	36
9.	Casos de Uso	37
10.	Esquema de estrella	42
11.	Esquema copo de nieve	42
12.	Esquema en copo de nieve de la tabla Movimientos	43
13.	Esquema en copo de nieve de la tabla Paletas	44
14.	Esquema en copo de nieve de la tabla Cabecera de Pedido de Compra	44
15.	Esquema en copo de nieve de la tabla Líneas de Pedidos de Compra	44
16.	Esquema en copo de nieve de la tabla Cabecera de Pedido de Venta	45
17.	Esquema en copo de nieve de la tabla Líneas de Pedidos de Venta	45
18.	Esquema en copo de nieve de la tabla Cabecera Devolución Cliente	46
19.	Esquema en copo de nieve de la tabla Líneas Devolución Cliente	46
20.	Esquema en copo de nieve de la tabla Cabecera Devolución Pro- veedor	47
21.	Esquema en copo de nieve de la tabla Líneas Devolución Proveedor	47
22.	Esquema en copo de nieve de la tabla Ordenes de Palet Completo	48
23.	Esquema en copo de nieve de la tabla Ordenes de Picking	48
24.	Esquema en copo de nieve de la tabla Líneas de Picking	49
25.	Esquema en copo de nieve de la tabla Ubicación de Picking	49
26.	Esquema en copo de nieve de la tabla Ubicación de Almacenaje .	50
27.	Esquema tabla Festivos y pseudocódigo procedimiento almacenado	52
28.	Diseño informe paletas caducadas	58
29.	Diseño informe pedidos de compra	59
30.	Diseño informe pedidos de venta	60
31.	Diseño informe productividad operarios	61
32.	Vista esquema DWH desde PL/SQL Developer	63
33.	Transformación de la tabla de dimensiones Artículos	64
34.	Transformación de las tablas de dimensiones Área descarga y Pasillos	65
35.	Transformación de la tabla de hechos Movimientos	65
36.	Transformación de la tabla de hechos Paletas	66
37.	Transformación de las tablas de hechos Pedidos de Compra	67
38.	Transformación de las tablas de hechos Pedidos de Venta, Ordenes de Palet Completo y Picking	69
39.	Transformación de las tablas de hechos Devoluciones Cliente	71
40.	Transformación de las tablas de hechos Devoluciones Proveedor .	72
41.	Transformación de las tablas de hechos Ubicación Picking	73
42.	Transformación de las tablas de hechos Ubicación de Almacenaje	73
43.	Transformación de las tablas de dimensiones Almacenes	74

44.	Transformación de las tablas de dimensiones Clientes y Proveedores	75
45.	Transformación de la tabla de dimensiones Operarios	75
46.	Transformación de las tablas de dimensiones propietarios	76
47.	Transformación de las tablas de dimensiones tipos movimiento y pedido	76
48.	Transformación de las tablas de dimensiones tipos pedido proveedor y paleta	77
49.	Job para la carga de datos	77
50.	Ejecución del job principal mediante Kitchen	78
51.	Script ejecución proceso ETL	79
52.	Fichero Crontab	79
53.	Consulta Vista Tiempo	80
54.	Operaciones sobre un schema	81
55.	Operaciones sobre jerarquía de una dimensión	81
56.	Operaciones sobre un cubo	82
57.	Atributos de una dimensión	82
58.	Schema con los cubos y dimensiones compartidas	83
59.	Añadir fuente de datos en Report Designer	84
60.	Creación de sentencia del informe	84
61.	Workspace diseño de informe	85
62.	Panel de datos en Report Designer	86
63.	Organización del proyecto en Eclipse	87
64.	Pantalla administración de fuentes de datos	88
65.	Asistente creación de conexión a base de datos	88
66.	Asistente publicación cubos OLAP	89
67.	Fuente de datos cubos OLAP	89
68.	Clonación del proyecto Pivot4J	90
69.	Vista Menú Crear nuevo	90
70.	Consulta contra cubo Cabecera Pedido Compra	91
71.	Herramienta para el desarrollo de Plugins	92
72.	Componentes de un cuadro de mando	92
73.	Código HTML del popup para aplicar filtros	93
74.	Implementación de la función wdu.filter	94
75.	Implementación de la función wdu.PostChange	94
76.	Implementación de la función wdu.removeFilter	95
77.	Popup selector de filtros	95
78.	Opciones sobre conexión en la herramienta PME	96
79.	Busqueda de tablas a importar en la herramienta PME	97
80.	Modificación de tabla ARTICULOS en la herramienta PME	97
81.	Asistente para la definición de relaciones entre tablas	98
82.	Asistente publicación dominio de metadatos	99
83.	Fuente de datos dominio de metadatos	99
84.	Ejemplo de consulta sobre pedidos de compra	99
85.	Asistente publicación de informe	100
86.	Selección ruta para publicación de informe	100
87.	Búsqueda informes en la aplicación web	101
88.	Ejemplo informe paletas caducadas	101
89.	Subida de la aplicación web al servidor	102
90.	Comprobación subida de la aplicación web al servidor	102

91.	Configuración puerto de la aplicación web Pentaho	103
92.	Pantalla login aplicación web Pentaho	103
93.	Resultado pruebas unitarias primera carga de datos	105
94.	Resultado pruebas unitarias carga de datos incremental	105
95.	Resultado medidas cubo movimientos mediante consulta SQL	106
96.	Resultado medidas cubo movimientos mediante consulta MDX	106
97.	Resultado consulta ad-hoc pedidos de compra	113
98.	Resultado consulta SQL pedidos de compra	114
99.	Resultado consulta ad-hoc pedidos de venta	114
100.	Resultado consulta SQL pedidos de venta	115
101.	Resultado consulta ad-hoc maestros	115
102.	Resultado consulta SQL maestros	115
103.	Resultado consulta ad-hoc control	116
104.	Resultado consulta SQL control	116
105.	Gráfica comparación horas estimadas y reales	117
106.	Diagrama metadato de compras	143
107.	Diagrama metadato de ventas	144
108.	Diagrama metadato de maestros	146
109.	Diagrama metadato de control	147
110.	Diagrama de clases JUnits proceso ETL	149
111.	Estructura proyecto JUnits del proceso ETL	152
112.	Pantalla de login de la aplicación web	153
113.	Mensaje de carga de la consola de usuario	153
114.	Interfaz consola de usuario	154
115.	Pestaña de navegación entre perspectivas	154
116.	Desplegable Crear Nuevo	155
117.	Interfaz menú ficheros con acciones sobre carpeta	156
118.	Interfaz menú ficheros con acciones sobre fichero	157
119.	Interfaz menú programaciones	158
120.	Popup asistente configuración de tiempo de interrupción	158
121.	Interfaz opciones menú administración	159
122.	Interfaz gestión usuarios y roles del menú administración	159
123.	Interfaz gestión servidor correo del menú administración	160
124.	Interfaz configuración en menú administración	160
125.	Interfaz pantalla consultas ad-hoc	161
126.	Diseño de consultas ad-hoc	162
127.	Resultado de consulta ad-hoc	162
128.	Popup selección de filtros sobre un campo	163
129.	Filtros aplicados en una consulta ad-hoc	163
130.	Pantalla selección de cubo OLAP	164
131.	Pantalla visor OLAP	164
132.	Ejemplo de consulta con el visor OLAP	165
133.	Configuración parámetros de informe predefinido	166
134.	Valores tipo de salida de informe predefinido	166
135.	Ejemplo de informe predefinido	167

Índice de tablas

1.	Plantilla Definición Paquetes de Trabajo	16
2.	Tarea Definición de Objetivos y Alcance del proyecto	16
3.	Tarea Planificación Temporal del proyecto	16
4.	Tarea Gestión de Riesgos	17
5.	Tarea Planificación Económica	17
6.	Tarea Antecedentes	17
7.	Tarea Aprender conceptos básicos de Business Intelligence	18
8.	Tarea Aprender funcionamiento de la plataforma Pentaho	18
9.	Tarea Aprender funcionamiento de la plataforma Pentaho	18
10.	Tarea Identificación de funcionalidades	19
11.	Tarea Casos de Uso	19
12.	Tarea Diseño del Datawarehouse	19
13.	Tarea Diseño de los cubos OLAP	20
14.	Tarea Diseño de informes	20
15.	Tarea Implementación e integración del Datawarehouse	20
16.	Tarea Implementación de cubos OLAP y visor	21
17.	Implementación e integración del Datawarehouse	21
18.	Tarea Despliegue del servidor web	21
19.	Tarea Despliegue del servidor web	22
20.	Tarea Realización de la memoria	22
21.	Tarea Pruebas de implementación	22
22.	Riesgo Perdida del proyecto	24
23.	Retraso en el desarrollo del proyecto	24
24.	Riesgo Desarrollo diferente al pedido	24
25.	Riesgo Mala planificación temporal	25
26.	Riesgo Ordenador defectuoso	25
27.	Riesgo Problemas en la implementación	25
28.	Diseño inadecuado	26
29.	Problemas en reunión con el cliente	26
30.	Gastos en personal	27
31.	Gastos en personal	27
32.	Gastos software	27
33.	Gastos indirectos	28
34.	Gastos totales	28
35.	Comparativa entre Datawarehouse y base de datos operacional [6] [7]	31
36.	Comparativa entre herramientas de BI [13] [14] [15] [16] [17] [18]	33
37.	Estructura Vista Dimensión Tiempo	52
38.	Resultado pruebas medidas del cubo movimientos	107
39.	Resultado pruebas medidas del cubo paletas	107
40.	Resultado pruebas medidas del cubo cabecera pedido de compra	108
41.	Resultado pruebas medidas del cubo líneas pedido de compra	108
42.	Resultado pruebas medidas del cubo cabecera pedido de venta	109
43.	Resultado pruebas medidas del cubo líneas pedido de venta	109
44.	Resultado pruebas medidas del cubo cabecera devolución cliente/ proveedor	110
45.	Resultado pruebas medidas del cubo líneas devolución cliente	110

46.	Resultado pruebas medidas del cubo líneas devolución proveedor	111
47.	Resultado pruebas medidas del cubo ordenes de paleta completa	111
48.	Resultado pruebas medidas del cubo ordenes de picking	112
49.	Resultado pruebas medidas del cubo líneas de picking	112
50.	Resultado pruebas medidas cubos ubicación de picking y almace- naje	113
51.	Comparación horas estimadas y reales	118
52.	Caso de Uso: Iniciar Sesión	124
53.	Caso de Uso: Programar Tareas	126
54.	Caso de Uso: Realizar consulta	127
55.	Caso de Uso: Crear cuadro de mando	128
56.	Caso de Uso: Visualizar cubos OLAP	129
57.	Caso de Uso: Abrir cuadros de mando	130
58.	Caso de Uso: Abrir informes	132
59.	Caso de Uso: Administrar fuentes de datos	133
60.	Caso de Uso: Administrar usuarios	134
61.	Caso de Uso: Administrar Roles de usuario	136
62.	Caso de Uso: Crear cubo OLAP	137
63.	Caso de Uso: Desplegar cubo OLAP	138
64.	Caso de Uso: Definir transformaciones	139
65.	Caso de Uso: Definir jobs	140
66.	Caso de Uso: Diseñar informe	141
67.	Caso de Uso: Publicar informe	142
68.	Relaciones metadato de compras	144
69.	Relaciones metadato de ventas	145
70.	Relaciones metadato de maestros	146
71.	Relaciones metadato de control	148

2. Introducción

2.1. Descripción del Proyecto

El proyecto que se desarrolla consiste en crear, desarrollar y desplegar una aplicación de Business Intelligence (BI) para la empresa *InLOG Consultoría y Soluciones*, con el objetivo de poder aportar a las empresas clientes de InLOG una visión del estado actual de su negocio, y poder generar información útil para tomas de decisiones en el futuro. Para el desarrollo se aplicarán distintas herramientas de Business Intelligence de código abierto de la plataforma Pentaho.

Para lograr este objetivo, es necesario, por un lado generar un almacén de datos, el cual contendrá datos previamente tratados de los distintos sistemas de gestión que los clientes poseen, entre ellos el sistema InLOG WMS, un sistema de gestión de almacenes propio de la empresa, siendo este el sistema de información preferente a la hora de tratar los datos al ser el propio de la empresa, y al ser el BI un producto que se ofrecerá junto al sistema.

Una vez generado el almacén de datos, se procederá a explotar los datos de la siguiente manera:

- Generar informes predefinidos parametrizables sobre distintos datos útiles para el usuario.
- Ofrecer la posibilidad de realizar informes-adhoc con los datos obtenidos en el almacén de datos, con el objetivo de que el cliente pueda realizar consultas de una forma sencilla sin conocer la estructura del almacén de datos.
- Desarrollo de cubos OLAP e integración de un visor para poder realizar consultas de una forma sencilla y óptima utilizando las posibilidades de los cubos generados.
- Integración del servidor web proporcionado por Pentaho, para ofrecer al cliente una forma sencilla de acceder a todas las características implementadas.

Para el desarrollo, se aplicara todos los conocimientos adquiridos durante el recorrido académico. Como sistema de gestión de base de datos se utilizará Oracle, al ser el sistema con el que trabaja InLOG WMS, y estar familiarizado con el desarrollo en ese entorno. Se realizará una formación sobre Pentaho, al desconocer su funcionamiento.

2.2. Motivación

Hoy en día, muchas de las empresas clientes de InLOG WMS, solicitan la posibilidad de explotar los datos generados por el sistema utilizando un Business Intelligence, al ser una buena oportunidad para explotar los múltiples datos generados por el sistema de gestión de almacenes.

Esto ha provocado el interés en la empresa por implementar una solución de Business Intelligence para ofrecer a los clientes, y además, atraer a nuevos clientes.

2.3. Definiciones

Se van a definir algunos conceptos que se utilizarán a lo largo del proyecto, para facilitar la lectura de este:

- **ERP:** *Enterprise Resource Planning*, son programas para gestionar distintas operaciones de una empresa.
- **WMS:** *Warehouse Management System*, es un programa para gestionar almacenes.
- **Datawarehouse:** Sistema para almacenar datos que se pueden analizar para toma de decisiones mediante herramientas de Business Intelligence.
- **Cubo OLAP:** Estructura de datos que proporciona un acceso y análisis rápido de los datos.
- **Informe ad-hoc:** Permite a los usuarios realizar consultas sobre la base de datos de una forma sencilla

3. Planteamiento inicial

3.1. Objetivos

3.1.1. Objetivos del proyecto

A continuación, se definen los siguientes objetivos a lograr con el proyecto, los cuales serán desarrollados a lo largo de la memoria:

- Definición del proceso de extracción y transformación de los datos procedentes del WMS, ERP y sistema de recursos humanos.
- Desarrollo del Datawarehouse en Oracle
- Integración del Datawarehouse con el servidor web de Pentaho
- Desarrollo de informes predefinidos parametrizables
- Integración de informes ad-hoc en el servidor web de Pentaho
- Desarrollo de cubos OLAP e integración de un visor para dichos cubos en el servidor web de Pentaho
- Despliegue del servidor web de Pentaho

3.1.2. Objetivos personales

En este proyecto, se pondrán en practica todos los conocimientos adquiridos durante el recorrido académico, lo cual es una buena oportunidad para reforzar estos conocimientos con un proyecto real en una empresa.

Además, la posibilidad de aprender a implementar una solución de Bussiness Intelligence con Pentaho es una gran oportunidad para entrar en un ámbito que cada vez mas empresas están interesadas, lo cual es muy útil para mi futuro profesional.

3.2. Marco del proyecto

InLOG Consultoría y Soluciones es una empresa de desarrollo de software, cuya principal actividad consiste en el diseño e implementación de soluciones en la industria, retail, transporte y sectores de servicios.



Figura 1: Logotipo InLOG Consultoría y Soluciones

Posee un software propio para la gestión de almacenes llamado InLOG WMS, el cual permite automatizar la gestión de las operaciones del almacén, gestionar de forma inteligente las prioridades, optimizar recorridos, reducir los errores operativos... logrando así aumentar y optimizar la productividad de los almacenes en los que se implanta el software.

Este software, al ser el propietario de la empresa, y generar una gran fuente de datos sobre todo lo relacionado con el almacén, será la principal fuente de extracción de datos para la implementación de este proyecto.

3.3. Alcance

3.3.1. Fases

El proyecto será dividido en diferentes fases, las cuales a su vez contarán con un listado de tareas a realizar, mostradas más adelante mediante un diagrama EDT (estructura de desglose del trabajo). Las fases son las siguientes:

- **Planificación y Gestión Inicial:** Fase inicial del proyecto, donde se definirán los objetivos del proyecto, las tareas necesarias para llevarlo a cabo y una planificación temporal estimada del proyecto, mediante el uso del diagramas EDT y Gantt.
- **Formación:** Fase donde se adquirirán los conocimientos necesarios para desarrollar el proyecto.
- **Captura de Requisitos:** Fase donde se definirán las distintas funcionalidades a implementar realizando reuniones con la empresa.
- **Análisis y Diseño:** Fase donde se identificarán las diferentes herramientas para implementar las funcionalidades, y se realizará el diseño de las funcionalidades.
- **Implementación e Integración:** Fase donde se desarrollarán e integrarán en la plataforma las distintas funcionalidades definidas.
- **Despliegue:** Fase donde se instalará y verificará que la aplicación funcione correctamente y se adecue a las necesidades establecidas por la empresa.
- **Documentación:** Fase donde se realizará la documentación correspondiente. Esta fase se realizará a lo largo de todo el proyecto.

3.3.2. Ciclo de Vida

Para la realización del proyecto se ha empleado un ciclo de vida clásico llamado ciclo de vida en cascada. En este, se ejecuta cada fase una tras otra de forma lineal, no pudiendo empezar con la segunda fase hasta no terminar la primera, y así sucesivamente. En caso de encontrar algún fallo, es posible volver a una fase anterior para corregirlo.

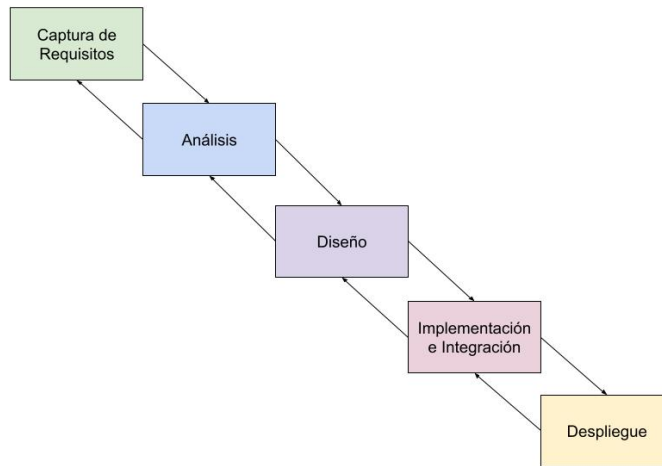


Figura 2: Ciclo de vida

3.4. Herramientas

Las herramientas hardware y software a emplear son las siguientes:

3.4.1. Hardware

- Ordenador portatil con Windows 10

3.4.2. Software

- Pentaho BI Suite Community Edition: Conjunto de software libre para generar Business Intelligence. Se utilizarán los siguientes programas:
 - BI Server 7.0: Aplicación web desarrollada en Java2EE en la que podremos gestionar todas las funcionalidades como informes y vista OLAP implementadas de Pentaho creando un cuadro de mando
 - Pentaho Data Integration (PDI): Aplicación para transformar datos desde diferentes fuentes y unificarlos en datawarehouse
 - Pentaho Report Designer: Aplicación para diseñar informes
 - Pentaho Schema Workbench: Aplicación para diseñar cubos OLAP
 - Pentaho Metadata Editor: Aplicación para definir los metadatos del Datawarehouse
- Texmaker: Editor de textos multiplataforma de código abierto para desarrollar documentos con LaTeX, con el cual se desarrollará la memoria del proyecto

- GanttProject: Software de código abierto empleado para la gestión de proyectos usando el diagrama de Gantt.
- PL/SQL Developer: Software para administrar bases de datos de Oracle.
- PuTTY: Software de código abierto para poder usar el protocolo SSH.
- Google Drive: Servicio de alojamiento de archivos en la nube.
- Google Drawings: Software para realizar diagramas, con el cual se realizarán las figuras de la memoria.
- Microsoft 365: Conjunto de software ofimático. Se utilizarán los siguientes programas:
 - Microsoft Access: Software de gestión de base de datos
 - Microsoft PowerPoint: Software para realizar presentaciones
- Visual Paradigm: Herramienta para modelado UML, con el cual se representaran los casos de uso del sistema.
- TortoiseGit: Software cliente para el control de versiones Git de código abierto.
- Eclipse IDE for Enterprise Java and Web Developers: Plataforma de código abierto para el desarrollo de aplicaciones Java EE y web.

3.5. Planificación Temporal

3.5.1. EDT

A continuación se muestra el diagrama EDT con el desglose del trabajo:

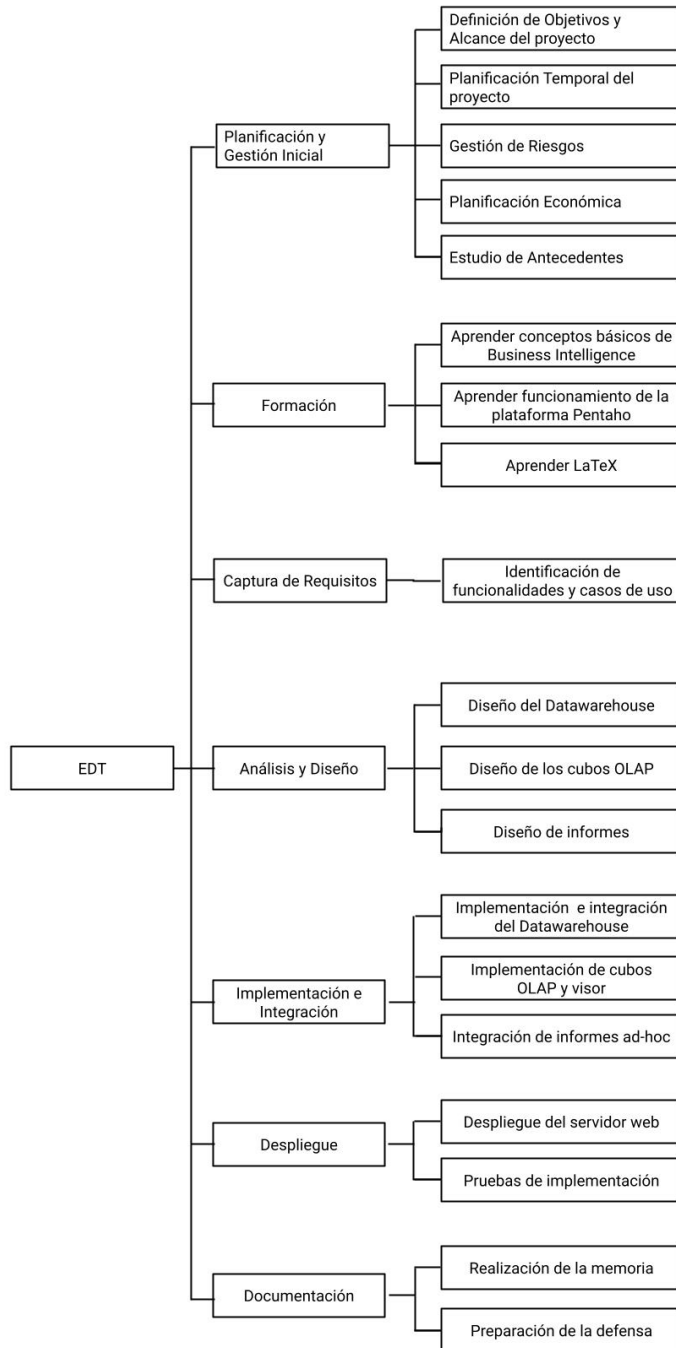


Figura 3: EDT

3.5.2. Listado de Tareas

A continuación se van a definir el listado de tareas de los cuales están compuesta cada fase. Para ello, utilizaremos como plantilla la siguiente tabla:

NOMBRE TAREA	
Descripción	Breve explicación de la tarea.
Esfuerzo	Duración estimada en horas de la tarea.
Recursos necesarios	Listado con las herramientas a utilizar en la tarea.
Precedente	Listado de tareas que deben estar completadas para poder comenzar.
Salidas:	Explicación de los hitos obtenidos con la tarea.

Tabla 1: Plantilla Definición Paquetes de Trabajo

Teniendo en cuenta las tareas definidas en el EDT:

1. Planificación y Gestión Inicial

Definición de Objetivos y Alcance del proyecto	
Descripción	Fijar los objetivos a lograr con el proyecto y establecer el alcance de este.
Esfuerzo	16 horas.
Recursos necesarios	Texmaker.
Precedente	-
Salidas:	Objetivos del proyecto y Alcance del proyecto.

Tabla 2: Tarea Definición de Objetivos y Alcance del proyecto

Planificación Temporal del proyecto	
Descripción	Identificar fases y tareas del proyecto, y estar el tiempo a emplear para realizar cada una de ella.
Esfuerzo	6 horas.
Recursos necesarios	Texmaker y GanttProject.
Precedente	Definición de Objetivos y Alcance del proyecto.
Salidas:	Diagrama EDT y listado de tareas junto con el tiempo estimado y fechas máximas de inicio y fin.

Tabla 3: Tarea Planificación Temporal del proyecto

Gestión de Riesgos	
Descripción	Identificar aquellos riesgos que pueden ocurrir a lo largo del desarrollo del proyecto.
Esfuerzo	3 horas.
Recursos necesarios	Texmaker.
Precedente	Definición de Objetivos y Alcance del proyecto.
Salidas:	Gestión de riesgos.

Tabla 4: Tarea Gestión de Riesgos

Planificación Económica	
Descripción	Estimación de costes y beneficios que supone el desarrollo del proyecto.
Esfuerzo	3 horas.
Recursos necesarios	Texmaker.
Precedente	Definición de Objetivos y Alcance del proyecto.
Salidas:	Planificación Económica.

Tabla 5: Tarea Planificación Económica

Estudio de Antecedentes	
Descripción	Análisis de la situación antes de realizar el proyecto y técnicas para realizarlo.
Esfuerzo	14 horas.
Recursos necesarios	Internet y Texmaker.
Precedente	Definición de Objetivos y Alcance del proyecto.
Salidas:	Antecedentes.

Tabla 6: Tarea Antecedentes

2. Formación

Aprender conceptos básicos de Business Intelligence	
Descripción	Adquirir conocimientos y conocer las herramientas que se utilizan en los proyectos de Business Intelligence.
Esfuerzo	14 horas.
Recursos necesarios	Internet
Precedente	Estudio de Antecedentes.
Salidas:	-

Tabla 7: Tarea Aprender conceptos básicos de Business Intelligence

Aprender funcionamiento de la plataforma Pentaho	
Descripción	Conocer el funcionamiento de las herramientas de la suite de Pentaho.
Esfuerzo	28 horas.
Recursos necesarios	Internet
Precedente	Aprender conceptos básicos de Business Intelligence.
Salidas:	-

Tabla 8: Tarea Aprender funcionamiento de la plataforma Pentaho

Aprender LaTeX	
Descripción	Aprender el uso de LaTeX para poder realizar la memoria del proyecto.
Esfuerzo	10 horas.
Recursos necesarios	Internet
Precedente	-
Salidas:	-

Tabla 9: Tarea Aprender funcionamiento de la plataforma Pentaho

3. Captura de Requisitos

Identificación de funcionalidades	
Descripción	Identificación de las funcionalidades a implementar.
Esfuerzo	10 horas.
Recursos necesarios	Texmaker.
Precedente	Formación.
Salidas:	Listado con las funcionalidades

Tabla 10: Tarea Identificación de funcionalidades

Casos de Uso	
Descripción	Definir los casos de uso.
Esfuerzo	40 horas.
Recursos necesarios	Visual Paradigm.
Precedente	Identificación de funcionalidades.
Salidas:	Casos de Uso

Tabla 11: Tarea Casos de Uso

4. Análisis y Diseño

Diseño del Datawarehouse	
Descripción	Diseño del proceso de extracción, transformación de datos, estructura de las tablas del Datawarehouse y carga de datos.
Esfuerzo	40 horas.
Recursos necesarios	Texmaker y Pentaho Data Integration.
Precedente	Captura de Requisitos.
Salidas:	Diagramas y tablas de base de datos

Tabla 12: Tarea Diseño del Datawarehouse

Diseño de los cubos OLAP	
Descripción	Diseño de la estructura de los cubos.
Esfuerzo	30 horas.
Recursos necesarios	Google Drawing.
Precedente	Diseño del Datawarehouse.
Salidas:	Diagramas de cubos

Tabla 13: Tarea Diseño de los cubos OLAP

Diseño de informes	
Descripción	Diseño de informes parametrizables.
Esfuerzo	28 horas.
Recursos necesarios	Pentaho Report Designer.
Precedente	Diseño del Datawarehouse.
Salidas:	Informes parametrizables

Tabla 14: Tarea Diseño de informes

5. Implementación e Integración

Implementación e integración del Datawarehouse	
Descripción	Implementación del datawarehouse en Oracle, implementación del proceso ETL e integración con el servidor web de Pentaho
Esfuerzo	15 horas.
Recursos necesarios	Pentaho Data Integration, Oracle PLSQL y BI Server 7.0
Precedente	Análisis y Diseño.
Salidas:	Datawarehouse

Tabla 15: Tarea Implementación e integración del Datawarehouse

Implementación de cubos OLAP y visor	
Descripción	Implementación de los cubos OLAP
Esfuerzo	10 horas.
Recursos necesarios	Pentaho Schema Workbench y BI Server 7.0
Precedente	Implementación e integración del Datawarehouse.
Salidas:	Cubos y visor OLAP

Tabla 16: Tarea Implementación de cubos OLAP y visor

Integración de informes ad- hoc	
Descripción	Instalación, modificación de plugins e integración de los datos para poder realizar informes ad-hoc desde el servidor web
Esfuerzo	8 horas.
Recursos necesarios	BI Server 7.0
Precedente	Implementación e integración del Datawarehouse.
Salidas:	Informes ad-hoc desde el servidor web

Tabla 17: Implementación e integración del Datawarehouse

6. Despliegue

Despliegue del servidor web	
Descripción	Despliegue del servidor web de Pentaho para poder ser utilizado y publicación de informes predefinidos.
Esfuerzo	15 horas.
Recursos necesarios	BI Server 7.0
Precedente	Implementación e integración.
Salidas:	Servidor Web Pentaho

Tabla 18: Tarea Despliegue del servidor web

Pruebas de implementación	
Descripción	Prueba del correcto funcionamiento de las funcionalidades implementadas
Esfuerzo	20 horas.
Recursos necesarios	BI Server 7.0
Precedente	Despliegue del servidor web.
Salidas:	-

Tabla 19: Tarea Despliegue del servidor web

7. Documentación

Realización de la memoria	
Descripción	Realizar la memoria del proyecto
Esfuerzo	80 horas.
Recursos necesarios	Texmaker
Precedente	Aprender LaTeX
Salidas:	Memoria del proyecto

Tabla 20: Tarea Realización de la memoria

Preparación de la defensa	
Descripción	Realizar la presentación para la defensa del proyecto
Esfuerzo	10 horas.
Recursos necesarios	Microsoft PowerPoint
Precedente	Realización de la memoria
Salidas:	Presentación de la defensa

Tabla 21: Tarea Pruebas de implementación

3.5.3. Gantt

Para gestionar las fechas en las que tienen que completar las tareas definidas anteriormente se utilizará el diagrama Gantt. Se ha estimado que para la realización del proyecto se empleara aproximadamente 15 horas semanales, trabajando en el proyecto 3 horas al día de lunes a viernes. En caso de no ser suficiente, se aumentará la cantidad de horas semanales trabajando los fines de semana en el proyecto.

Se han tenido en cuenta un periodo de vacaciones por navidad entre el 22/12/2020 y 6/12/2021, y vacaciones por semana santa entre el 28/03/2021 y 4/04/2021.

Nombre	Fecha de inicio	Fecha de fin
Planificación y Gestión Inicial	11/12/20	26/01/21
Definición de Objetivos y Alcance del proyecto	11/12/20	22/12/20
Planificación Temporal del proyecto	20/01/21	22/01/21
Gestión de Riesgos	25/01/21	25/01/21
Planificación Económica	25/01/21	26/01/21
Estudio de Antecedentes	11/01/21	13/01/21
Formación	7/12/20	9/02/21
Aprender conceptos básicos de Business Intelligence	14/01/21	19/01/21
Aprender funcionamiento de la plataforma Pentaho	27/01/21	9/02/21
Aprender LaTeX	7/12/20	10/12/20
Captura de Requisitos	10/02/21	26/02/21
Identificación de funcionalidades	10/02/21	15/02/21
Casos de Uso	16/02/21	26/02/21
Análisis y Diseño	1/03/21	16/04/21
Diseño del Datawarehouse	1/03/21	15/03/21
Diseño de cubos OLAP	16/03/21	25/03/21
Diseño de informes	7/04/21	16/04/21
Implementación e Integración	19/04/21	29/04/21
Implementación e integración del Datawarehouse	19/04/21	22/04/21
Implementación de cubos OLAP y visor	23/04/21	27/04/21
Integración de informes ad-hoc	28/04/21	29/04/21
Despliegue	30/04/21	12/05/21
Despliegue del servidor web	30/04/21	5/05/21
Pruebas de implementación	6/05/21	12/05/21

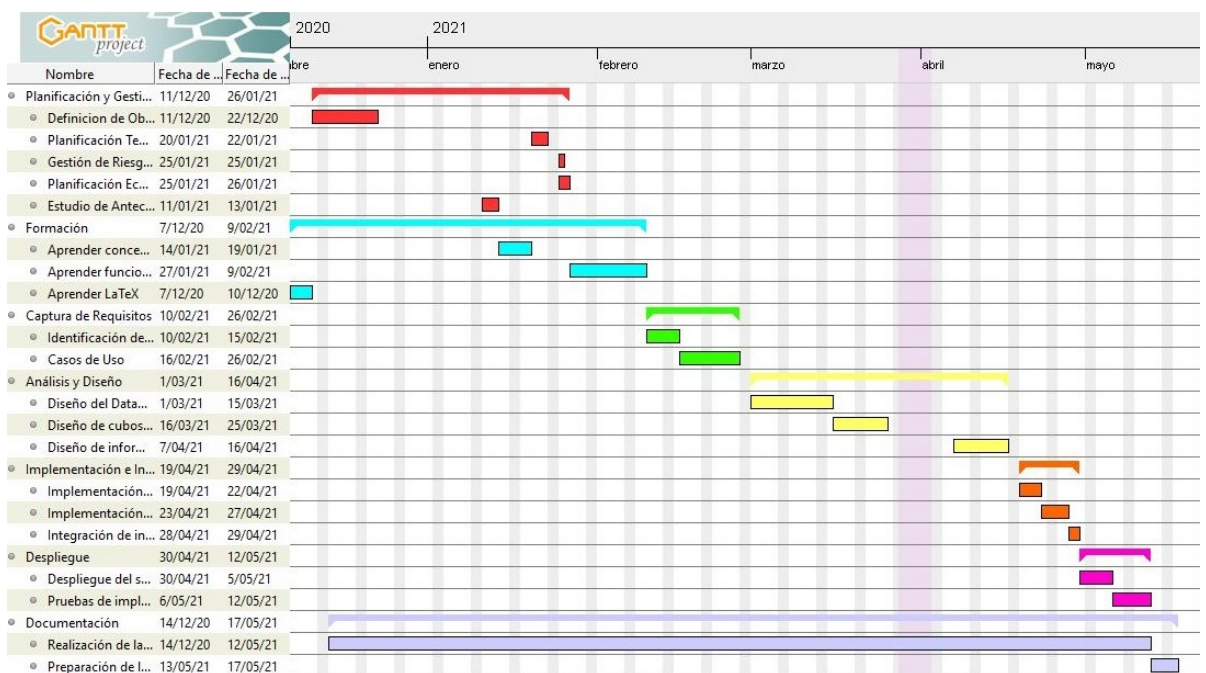


Figura 4: Diagrama Gantt

3.6. Gestión de Riesgos

A la hora de definir un proyecto es importante tener en cuenta los distintos riesgos que pueden surgir a lo largo del proyecto, los cuales si no los tenemos en cuenta pueden afectar a la planificación temporal.

Cada riesgo se expondrá de la siguiente manera: una breve descripción del riesgo, medidas que se emplearan para prevenir el riesgo, plan de contingencia a aplicar en caso de que el riesgo ocurra, la probabilidad de que ocurra y el impacto que puede suponer en el proyecto. Estas dos últimas medidas se medirán en cinco rasgos: muy bajo, bajo, medio, alto y muy alto.

Riesgo: Perdida del proyecto	
Descripción	Se pierden los datos del proyecto por algún fallo en el disco duro del ordenador
Medidas de Prevención	Realizar copias de seguridad periódicamente de todo el proyecto en un servicio en la nube (Google Drive)
Plan de Contingencia	Recuperar la copia de seguridad
Probabilidad	Baja
Impacto	Muy alto

Tabla 22: Riesgo Perdida del proyecto

Riesgo: Retraso en el desarrollo del proyecto	
Descripción	Debido a una enfermedad, confinamiento o motivos familiares se produce un retraso en el desarrollo del proyecto
Medidas de Prevención	Tener en cuenta un margen de tiempo por motivos personales en la planificación temporal
Plan de Contingencia	Realizar horas extra
Probabilidad	Media
Impacto	En función del tiempo perdido

Tabla 23: Retraso en el desarrollo del proyecto

Riesgo: Desarrollo diferente al pedido	
Descripción	El desarrollo realizado es diferente al que la empresa esperaba
Medidas de Prevención	Realizar reuniones periódicas mostrando los avances del proyecto
Plan de Contingencia	Realizar una reunión para aclarar las diferencias con lo esperado e invertir tiempo extra para ponerse al día
Probabilidad	Baja
Impacto	Muy alto

Tabla 24: Riesgo Desarrollo diferente al pedido

Riesgo: Mala planificación temporal	
Descripción	Se tarda mas tiempo en realizar una tarea del estimado
Medidas de Prevención	Tener en cuenta un margen de tiempo en la planificación temporal
Plan de Contingencia	Realizar horas extra
Probabilidad	Media
Impacto	En función del tiempo perdido

Tabla 25: Riesgo Mala planificación temporal

Riesgo: Ordenador defectuoso	
Descripción	Se estropea el ordenador con el que se está desarrollando el proyecto
Medidas de Prevención	Tener un ordenador alternativo disponible
Plan de Contingencia	Continuar el desarrollo desde el otro ordenador
Probabilidad	Baja
Impacto	Alto

Tabla 26: Riesgo Ordenador defectuoso

Riesgo: Problemas en la implementación	
Descripción	A la hora de implementar una determinada parte se encuentran problemas de los que no se conoce la solución
Medidas de Prevención	-
Plan de Contingencia	Buscar la solución en internet
Probabilidad	Alta
Impacto	Medio

Tabla 27: Riesgo Problemas en la implementación

Riesgo: Diseño inadecuado	
Descripción	El diseño realizado es insuficiente para implementar el proyecto
Medidas de Prevención	Realizar la captura de requisitos de la mejor manera posible para poder realizar el diseño mas ajustado posible
Plan de Contingencia	Volver a realizar el diseño en horas extra
Probabilidad	Baja
Impacto	Alto

Tabla 28: Diseño inadecuado

Riesgo: Problemas en reunión con el cliente	
Descripción	Debido al trabajo no se pueda realizar la reunión con el cliente y esta es cancelada
Medidas de Prevención	-
Plan de Contingencia	Continuar desarrollando otro apartado evitando paralizar el proyecto
Probabilidad	Media
Impacto	Alto

Tabla 29: Problemas en reunión con el cliente

3.7. Planificación Económica

En este apartado se va a hacer un cálculo de los gastos y beneficios que otorgará el proyecto, para poder así analizar su viabilidad.

3.7.1. Gastos

Tenemos los siguientes gastos:

- a. Gasto en personal: Al ser un proyecto realizado aprovechando la beca en la empresa, se ha estimado que un becario trabajando 3h al día cobra aproximadamente 330€ al mes.

	Coste mensual	Meses trabajados	Total
Becario	330€	6 meses	1980€
		Gasto total	1980€

Tabla 30: Gastos en personal

- b. Gasto en hardware: Para realizar el proyecto se ha utilizado un ordenador portátil Asus valorado aproximadamente en 650€, y puntualmente un ordenador sobremesa personal valorado en aproximadamente 1000€.

	Precio	Vida media	Tiempo de uso	Amortización
Ordenador portatil	650€	5 años	6 meses	64,99€
Ordenador sobre-mesa	1000€	6 años	1 mes	13,88€
			Gasto total	78,87€

Tabla 31: Gastos en personal

- c. Gasto en software: Para realizar el proyecto, la mayoría del software utilizado es gratuito, excepto Microsoft Access, el cual está incluido en el Microsoft Office, el cual tiene un coste de 7€ al mes.

	Coste mes	Meses trabajados	Total
Microsoft 365 Personal	7€/mes	6 meses	63€
		Gasto total	63€

Tabla 32: Gastos software

- d. Gastos indirectos: Son gastos que no tienen influencia directa en el desarrollo del proyecto, pero que son esenciales para poder realizarlo, como los gastos de luz e Internet.

En este caso, al ser un trabajo que se realizará en su mayor medida en la oficina de la empresa, no tendremos en cuenta los gastos de Internet, al ser un gasto independiente de la realización de este proyecto.

	Coste mes	Meses trabajados	Total
Luz	10€/mes	6 meses	60€
		Gasto total	60€

Tabla 33: Gastos indirectos

- e. Gasto total: Agrupando todos los costes, los costes totales en el proyecto quedan de la siguiente manera:

	Gasto
Personal	1980€
Hardware	78,87€
Software	63€
Indirectos	60€
Total	2181,87€

Tabla 34: Gastos totales

3.7.2. Beneficios

En cuanto a los beneficios que puede otorgar el desarrollo del proyecto, este es un producto que se espera ofrecer integrado junto al sistema actual de gestión de almacenes InLOG WMS, y no se va a ofrecer como servicio independiente, sino que irá integrado en las nuevas versiones del sistema para aumentar la competitividad de este producto en el mercado.

Con el desarrollo del proyecto se espera aumentar el abanico funcional del producto, logrando una expansión del sistema. Como los indicadores de cada empresa cliente de InLOG WMS son diferentes, al contar cada cliente con una versión personalizada de este adaptada a sus necesidades, se abre la posibilidad de realizar nuevos proyectos para adaptar el producto a cada cliente.

Además, desde el punto de vista comercial, con la expansión del sistema aumenta la posibilidad de venta de este, al ser una opción mas llamativa para posibles nuevos clientes, y por tanto podemos considerarlo como un proyecto generador de nuevas oportunidades.

Con lo cual, se espera recuperar la inversión realizada con los nuevos proyectos tanto de adaptación como de posibles nuevos clientes que se puedan concretar a partir de la implantación de este producto en las nuevas versiones del sistema.

4. Antecedentes

4.1. Situación actual

Actualmente, los clientes de InLOG WMS no disponen de ningún Datawarehouse que unifique los datos de sus diferentes sistemas. Las únicas conexiones que existen entre los sistemas ERP y WMS es mediante procedimientos de carga de datos implementados en InLOG WMS para la carga de pedidos, devoluciones... procedentes del ERP utilizando ficheros con una estructura concreta.

Lo máximo que actualmente el cliente puede hacer es imprimir los informes ya establecidos en la aplicación y hacer consultas contra la base de datos operativa mediante ODBC, perjudicando al rendimiento del sistema de gestión de almacenes en caso de hacer consultas muy pesadas, al realizar las operaciones contra la misma base de datos.

Al implementar un Datawarehouse, no solo lograremos que el cliente pueda realizar consultas e informes de datos procedentes de distintos sistemas de una manera sencilla, sino que al separar la base de datos operativa del Datawarehouse, el cliente podrá realizar consultas muy pesadas sin ralentizar el sistema.

4.2. Definición de Business Intelligence

Se define Business Intelligence como el conjunto de estrategias que realizan el análisis de los datos de una empresa para poder comprender el funcionamiento de los distintos procesos de la empresa y en base a esto, poder obtener los conocimientos suficientes para facilitar la toma de decisiones, acciones a tomar y lograr un incremento en el rendimiento y competitividad de la empresa. [1]

Para ello, las herramientas de business intelligence utilizan datos extraídos de los distintos sistemas de información que la empresa posee, generalmente . Estos datos son transformados mediante el proceso conocido como ETL (Extraer, Transformar y Cargar) para que puedan generar información útil para la toma de decisiones. [1]

El proceso de ETL consiste en obtener datos de múltiples fuentes, transformarlos y centralizarlos en un único repositorio, normalmente en un almacén de datos o Datawarehouse: [2]

1. Proceso de extracción: Se extraen los datos en bruto de fuentes diversas
2. Proceso de transformación: Se modifican los datos en bruto mediante una serie de normas para garantizar datos de calidad y prepararlos para poder utilizarlos
3. Proceso de carga: Se cargan los datos extraídos y transformados en el nuevo destino.



Figura 5: Proceso ETL

Una vez integrados los datos, se procede a trabajar sobre estos datos utilizando herramientas propias de la tecnología Business Intelligence mediante la creación de cuadros de mando, gráficos, indicadores e informes. Estas herramientas permitirán al usuario explorar los datos de una forma sencilla. [3]

Con la implementación de un Business Intelligence podemos destacar los siguientes beneficios: [4]

- Proporciona herramientas de análisis para la toma de decisiones
- Los usuarios pueden realizar consultas contra sus datos de una manera sencilla
- Reduce el tiempo necesario para realizar consultas contra datos procedentes de diferentes sistemas al estar estos unificados en una misma base de datos
- Posibilita realizar consultas pesadas sin ralentizar los sistemas de la empresa al ser una base de datos aislada al sistema
- Se pueden identificar aquellos factores que inciden en el funcionamiento de la empresa

4.3. Componentes de un Business Intelligence

4.3.1. Datawarehouse

Un Datawarehouse, traducido literalmente como “almacén de datos” es una base de datos que se caracteriza por ser: [5]

- **Integrado:** Datos de una o múltiples fuentes distintas en una misma base de datos.
- **Organizado por temas:** Los datos están clasificados en temas para facilitar su acceso y entendimiento por parte de los usuarios.

- **Histórico:** Guarda los distintos variables que toma una variable en el tiempo para poder ver evoluciones, comparaciones y buscar patrones.
- **No volátil:** La información es permanente y solo de lectura.

Otra característica interesante de los datawarehouse es que contiene metadatos. Los metadatos son datos que describen y dan contexto otros datos. Contienen información como por ejemplo la procedencia, estructura de datos, formato, etc. La gestión de metadatos puede ser de gran ayuda para mejorar la gestión de los datos, facilitando las búsquedas y análisis de datos. [5] [12]

En la siguiente tabla se puede ver una comparativa entre un Datawarehouse y una base de datos operacional:

	Datawarehouse	Base de datos Operacional
Objetivo	Análisis de datos	Operaciones del día a día
Datos	De todo el negocio	Relacionada con la aplicación
Valor	Importancia de los valores historicos	Importancia de los valores actuales
Actividad	Consultas grandes y masivas	Actualizaciones y pequeñas consultas
Volatilidad	Solo carga y lectura	Actualizable
Organización	Esquema en estrella o copo de nieve	Relacional
Rendimiento	Importancia a la respuesta masiva	Importancia a una respuesta instantánea

Tabla 35: Comparativa entre Datawarehouse y base de datos operacional [6] [7]

Se define como Data Mart al conjunto de datos que representa un único proceso o área de negocios especifica. Un Datawarehouse puede verse como el conjunto de los Data Marts de una organización [19]

4.3.2. Cubos OLAP

Un cubo OLAP es una estructura de datos multidimensional, ampliando las posibilidades y limitaciones que se ofrecían en las bases de datos relacionales, donde los datos son vistos como cubos. Las distintas dimensiones de este cubo consisten en las diferentes categorías en las que interesa estructurar los datos. Con esta estructura, logramos realizar consultas complejas con gran rapidez, y poder procesar grandes cantidades de información, los cuales sería imposible de procesar con una base de datos tradicional. [8] [9]

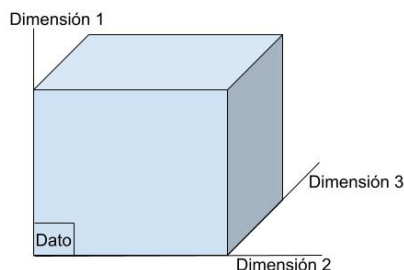


Figura 6: Estructura Cubo OLAP

Para realizar consultas contra los cubos, se utilizan consultas MDX, las cuales son unas consultas similares a las utilizadas en el lenguaje SQL, pero adaptadas para trabajar con estructuras de datos multidimensionales. [22]

Como principal desventaja es importante destacar la imposibilidad de realizar cambios en su estructura debido a su forma de almacenamiento de la información, teniendo que rediseñar el cubo OLAP en caso de querer realizar modificaciones. [9]

4.3.3. Cuadros de mando

Un cuadro de mando es la herramienta que muestra los indicadores clave de una forma visual con el objetivo de facilitar la toma de decisiones. Se distinguen dos tipos de cuadro de mando: [10]

- **Cuadro de Mando Operativo:** Enfocado al control de indicadores pertenecientes a los procesos de un área o departamento concreto de la empresa. [10] [11]
- **Cuadro de Mando Integral:** Muestra los datos desde cuatro perspectivas diferentes, financiera, cliente interna y aprendizaje/crecimiento. [10] [11]

4.4. Análisis de Herramientas

Al existir una gran cantidad de herramientas para el desarrollo de proyectos BI es importante realizar un análisis de las distintas herramientas existentes.

El estudio se ha basado principalmente en las herramientas Power BI, Pentaho CE, QlikView y SAP Business Intelligence, por ser de las principales herramientas en el mercado:

- **Power BI:** Servicio de Business Intelligence propietario de Microsoft. Permite realizar los procesos BI de una forma muy visual y simple. [13]
- **Pentaho CE:** Conjunto de programas de Business Intelligence de código abierto. Ofrece una amplia gama de herramientas para realizar los distintos procesos, y estás cuenta con complementos de código abierto para distintos objetivos. [16]
- **QlikView:** Herramienta para visualizar y generar cuadros de mando de una forma rápida. [14]
- **SAP Business Intelligence:** Herramienta de Business Intelligence de pago propietaria de SAP. El precio varía en función de las necesidades, siendo necesario establecer contacto con la empresa para estimar el precio. [15]

En la siguiente tabla se puede ver una comparativa entre las herramientas, teniendo en cuenta aquellos puntos indispensables para realizar el proyecto.

	Power BI	Pentaho CE	QlikView	SAP BI
Procesos ETL	SI	SI	NO	SI
Cuadros de mando	SI	SI	SI	SI
Informes	SI	SI	SI	SI
Cubos OLAP	NO	SI	NO	SI
Open Source	NO	SI	NO	NO
Precio	Gratuito	Gratuito	Gratuito	Pago

Tabla 36: Comparativa entre herramientas de BI [13] [14] [15] [16] [17] [18]

Como se puede ver, entre todas las herramientas analizadas, las únicas con las que se podría implementar el proyecto son Pentaho CE y SAP Business Intelligence. Teniendo en cuenta que la herramienta de SAP es de pago, no es de código abierto, y que con Pentaho podemos implementar prácticamente las mismas funcionalidades, y al ser código abierto podemos realizar modificaciones en caso de verlo necesario, la herramienta utilizada para implementar el proyecto será Pentaho CE. La única diferencia que podemos encontrar entre ellas es que al ser SAP Business Intelligence una aplicación de pago, el soporte y la documentación será mucho mayor que el de Pentaho, pero al ser esta última una solución BI muy popular, existe una comunidad muy activa donde poder adquirir una buena formación.

4.4.1. Pentaho CE

La herramienta BI escogida para implementar este proyecto es Pentaho BI, en concreto, Community Edition (CE) 7.0, pues esta es una de las versiones Open Source con mayor estabilidad, mayor soporte y comunidad activa.

Pentaho BI CE consiste en conjunto de herramientas programadas en su gran mayoría en Java, independientes entre si, y que permiten la implementación de todos los procesos de Business Intelligence. Las herramientas que se utilizaran para el desarrollo del proyecto son las siguientes:

- Pentaho Data Integration (PDI): Es una aplicación de escritorio para diseñar y crear los procesos ETL. Permite realizar transformaciones, que son los procesos que permiten extraer, transformar y cargar los datos al Datawarehouse, y ejecutar trabajos, los cuales se utilizan para ejecutar los procesos ETL periódicamente. [21]
PDI permite trabajar tanto con archivos, como mediante el uso de un repositorio proporcionado por Pentaho, el cual es interesante para mantener un control de versiones del proceso, y además poder realizar un uso colaborativo de la herramienta. [22]
- Pentaho Report Designer (PRD) : Es una aplicación de escritorio que proporciona un entorno de diseño visual para crear informes tanto estáticos como con datos dinámicos a partir de filtros. Una vez creados los informes, permite publicarlos en el servidor web de Pentaho para que el usuario pueda visualizarlos desde allí. [23]
Los reportes generados tienen el formato .prpt (Pentaho Reporting Archive) [27]

- Pentaho Schema Workbench (PSW): Es una aplicación de escritorio que proporciona una interfaz gráfica para diseñar y crear los cubos OLAP, generando un fichero XML donde se define la estructura de la base de datos multidimensional. Proporciona un servidor OLAP en el que permite realizar consultas MDX contra los cubos creados, y permite una vez creados los cubos, publicar estos en el servidor web de Pentaho. [24]
- Pentaho Metadata Editor: Es una aplicación de escritorio que permite diseñar un modelo de metadatos a partir del Datawarehouse. [25]
- Pentaho BI Server: Es una aplicación web programada con el framework Spring. Esta es el núcleo de Pentaho, donde se alojan todos los recursos generados para poder ser utilizados de una manera sencilla por el usuario. Permite entre otros crear cuadros de mando, visualizar informes, gestionar los usuarios y roles de usuario. La suite proporciona un servidor Tomcat para poder desplegar la aplicación de una forma sencilla. [26]

Podemos ver la arquitectura de Pentaho resumida en el siguiente diagrama:

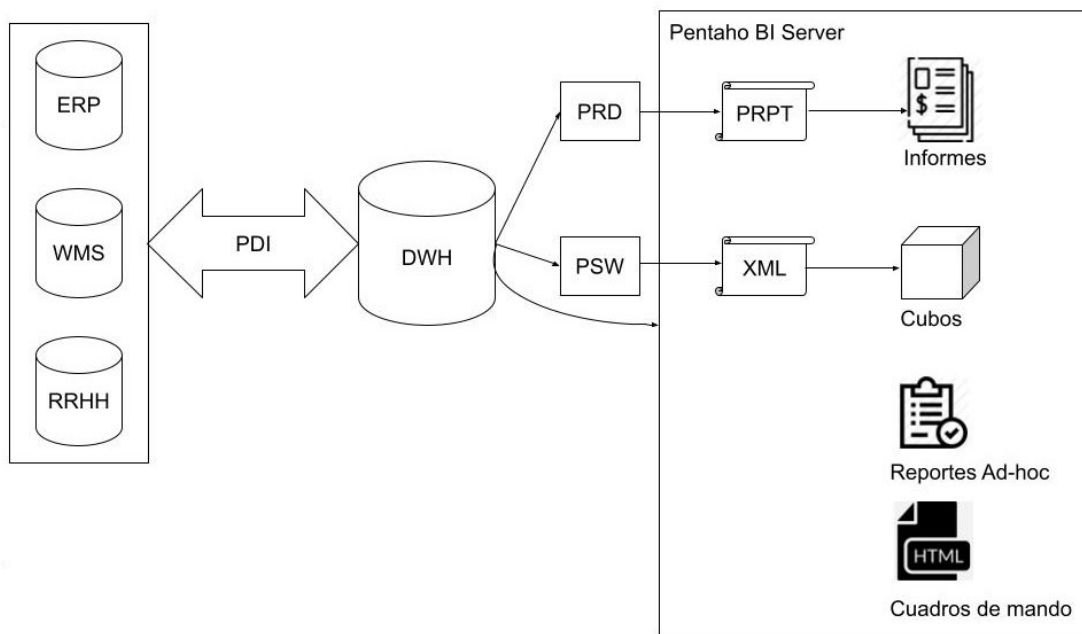


Figura 7: Diagrama Arquitectura Pentaho

5. Captura de requisitos

En este apartado se detallaran los requisitos funcionales y casos de uso que deberá cumplir el proyecto, para poder alcanzar las necesidades y objetivos del proyecto definidos anteriormente.

5.1. Requisitos funcionales

Los requisitos funcionales definen las funcionalidades del sistema:

- La aplicación solo debe poder usarse con un usuario registrado.
- Permitir carga de datos en el Datawarehouse desde distintas fuentes.
- Realizar el proceso ETL para la carga de datos en el Datawarehouse periódicamente.
- Generar una serie de indicadores que sirva para el control directo mediante la consulta de estos indicadores.
- Realizar consultas contra el Datawarehouse de una forma sencilla.
- Generación de distintos informes para la alta gerencia con el objetivo de dar una visión general de lo que ocurre en su empresa.
- Gestión de roles de usuario.
- Ofrecer la posibilidad de exportar las consultas e informes realizados a otras herramientas ofimáticas como Excel.
- Ofrecer la posibilidad de programar tareas.
- Ofrecer la posibilidad de crear y visualizar cuadros de mando.
- Flexibilidad del sistema para que el usuario final pueda formatear la información a visualizar en los cuadros de mando.

5.2. Requisitos no funcionales

Los requisitos funcionales son aquellos requisitos que el sistema debe cumplir independientemente de las funcionalidades a implementar:

- El sistema debe ser estable y fácil de utilizar para personas sin conocimientos técnicos de informática.
- La aplicación debe mantener los datos utilizados seguros y protegidos.
- La aplicación debe ser multiplataforma.
- La aplicación no debe penalizar el rendimiento de los sistemas operacionales.

5.3. Jerarquía de actores

Los tipos de usuarios del sistema son los siguientes: Por un lado, en la aplicación web, el "Usuario sin registrar" es aquel que accede a la aplicación y aún no ha iniciado sesión. Una vez que decide iniciar sesión pasa a ser "Usuario" o "Administrador" en función de su condición. Ambos tienen opción de hacer uso de las funcionalidades del sistema, pero el "Administrador" además podrá gestionar los ajustes del sistema, dar de alta nuevos usuarios y gestionar sus permisos.

Por otro lado, se define el actor "AdministradorCubos", quien accede a la aplicación Pentaho Cube Designer y puede crear nuevos cubos OLAP y desplegarlos en la aplicación web, "AdministradorETL", el cual es el actor que accede a la aplicación Pentaho Data Integration y puede definir y modificar el proceso de carga de datos ETL al Datawarehouse, y el actor "AdministradorReport", el cual accede a la aplicación Pentaho Report Designer y puede crear y modificar informes y subirlos a la aplicación web.

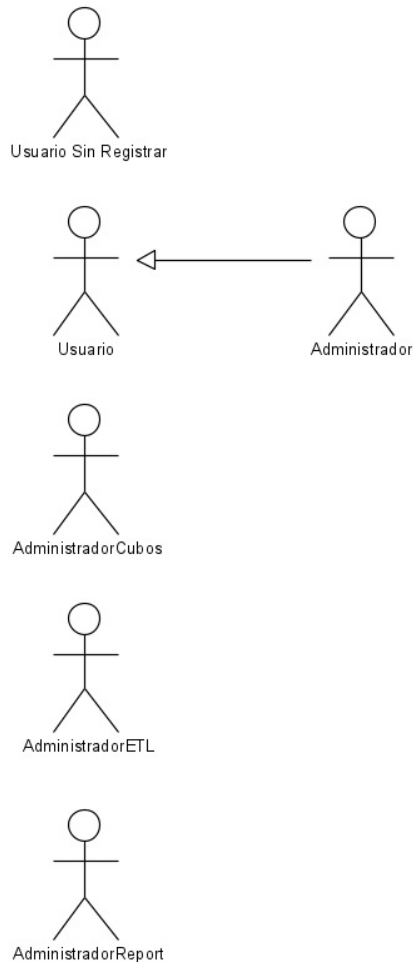


Figura 8: Jerarquía de actores

5.4. Casos de uso

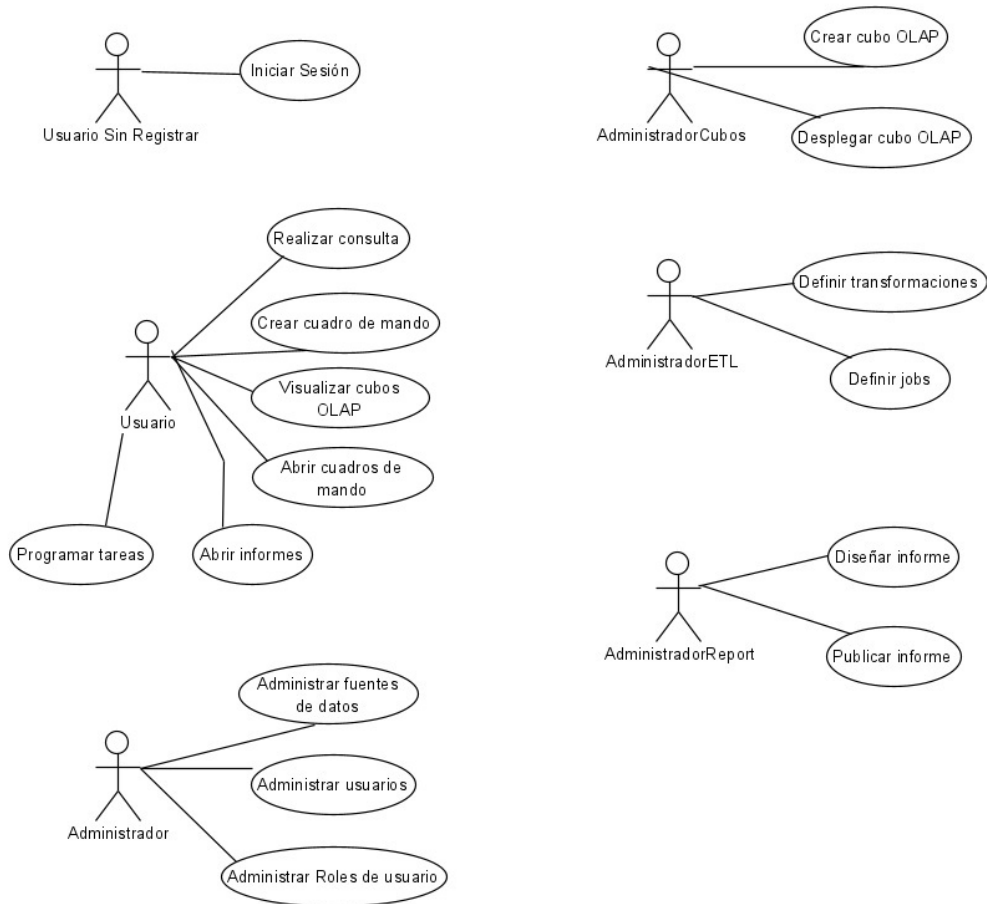


Figura 9: Casos de Uso

En el diagrama se observan las siguientes funcionalidades:

- **Iniciar sesión:** La función es identificarse en la aplicación web.
- **Realizar consulta:** Permite realizar consultas contra el Datawarehouse.
- **Crear cuadro de mando:** Permite diseñar un cuadro de mando.
- **Visualizar cubos OLAP:** Permite visualizar cubos OLAP desplegados en el servidor utilizando un visor OLAP.
- **Abrir cuadros de mando:** Permite abrir cuadros de mando diseñados anteriormente.
- **Abrir informes:** Permite abrir y exportar a diferentes herramientas ofimáticas informes predefinidos y publicados en el servidor web.

- **Programar tareas:** Permite programar tareas a ejecutar en el servidor web.
- **Administrar fuentes de datos:** Permite añadir y modificar las diferentes fuentes de datos con las que funciona la aplicación, y formatear los datos utilizados para las consultas.
- **Administrar usuarios:** Permite crear y eliminar los usuarios que pueden acceder a la aplicación.
- **Administrar Roles de usuario:** Permite modificar los permisos de los distintos usuarios de la aplicación.
- **Crear cubos OLAP:** Permite diseñar cubos OLAP a partir del Datawarehouse.
- **Desplegar cubos OLAP:** Permite desplegar los cubos OLAP diseñados en el servidor web para poder visualizarlos desde el visor OLAP.
- **Definir transformaciones:** Permite definir y modificar las diferentes transformaciones de datos que conforma el proceso de ETL.
- **Definir jobs:** Permite definir tareas para carga de datos en el Datawarehouse de forma periódica.
- **Diseñar informe:** Permite diseñar informes predefinidos y parametrizables a partir de los datos del Datawarehouse.
- **Publicar informe:** Permite publicar los informes diseñados en el servidor web para poder utilizarlos.

Podemos ver los casos de uso extendidos definidos en el Anexo I: Casos de uso extendidos

6. Análisis y diseño

6.1. Análisis de la base de datos origen

El primer paso del desarrollo de un proyecto BI es construir el Datawarehouse. Para ello, es importante analizar la información de los orígenes de datos, para así poder determinar aquella información relevante a exportar para poder realizar unos informes y cuadros de mando útiles para el cliente.

En este caso, el origen de datos principal es el de InLOG WMS el cual trabaja con una base de datos de tipo objeto-relacional Oracle. Al haber estado trabajando durante aproximadamente medio año con el sistema InLOG WMS y disponer de ayuda de expertos, el análisis y selección de las tablas relevantes se convierte en una tarea bastante sencilla.

Las tablas que se van a extraer van a ser tablas de hechos (tablas que almacena los datos de los movimientos que se actualizan continuamente) y tablas de dimensiones (tablas maestras por las cual se va a poder filtrar o navegar dentro de la información almacenada en las tablas de hechos).

A continuación, se listan todas las tablas de InLOG WMS relevantes para el desarrollo del Datawarehouse, junto con la forma de carga de estas al Datawarehouse:

1. Tablas de hechos:

- **MOVIMIENTOS:** Tabla de los movimientos realizados en el sistema. Al tener fecha y hora fin del movimiento, se extraera toda la información de la tabla que esté comprendida en el rango de fechas marcado por el histórico que se quiera tener en las tablas historicas del Datawarehouse.
- **PALETAS:** Tabla de paletas existentes con mercancía. Se extraerán todos los campos de la tabla. Para mantener un histórico del alta de las paletas, se realizará un cruce con la tabla movimientos.
- **CABECERA DE PEDIDO DE COMPRA:** Tabla de cabeceras de pedidos de compra. Se extraerá toda la información de la tabla que esté comprendida en el rango de fechas marcado por el histórico que se quiera tener en las tablas históricas del Datawarehouse, teniendo en cuenta la fecha de alta del pedido en el sistema.
- **LINEAS DE PEDIDOS DE COMPRA:** Tabla de lineas de pedidos de compra. Se extraerán todas las lineas de pedidos correspondientes a las cabeceras extraidas.
- **CABECERA DE PEDIDO DE VENTA:** Tabla de cabeceras de pedidos de ventas. Se extraerá toda la información de la tabla que esté comprendida en el rango de fechas marcado por el histórico que se quiera tener en las tablas históricas del Datawarehouse, teniendo en cuenta la fecha de alta del pedido en el sistema.
- **LINEAS DE PEDIDOS DE VENTA:** Tabla de lineas de Pedidos de Venta. Se extraerán todas las lineas de pedidos correspondientes a las cabeceras extraidas.

- **CABECERA DEVOLUCIÓN CLIENTE:** Tabla de cabeceras de devoluciones de clientes. Se extraerá toda la información de la tabla que esté comprendida en el rango de fechas marcado por el histórico que se quiera tener en las tablas históricas del Datawarehouse, teniendo en cuenta la fecha de alta de la devolución en el sistema.
- **LINEAS DEVOLUCIÓN CLIENTE:** Tabla de líneas de devoluciones de clientes. Se extraerán todas las líneas de devoluciones correspondientes a las cabeceras extraídas.
- **CABECERA DEVOLUCIÓN PROVEEDOR:** Tabla de cabeceras de devoluciones de proveedores. Se extraerá toda la información de la tabla que esté comprendida en el rango de fechas marcado por el histórico que se quiera tener en las tablas históricas del Datawarehouse, teniendo en cuenta la fecha de alta de la devolución en el sistema.
- **LINEAS DEVOLUCIÓN PROVEEDOR:** Tabla de líneas de devoluciones de proveedores. Se extraerán todas las líneas de devoluciones correspondientes a las cabeceras extraídas.
- **ORDENES DE PALET COMPLETO:** Tabla de Ordenes Paleta Completa. Se extraerán los registros de los pedidos de venta extraídos.
- **ORDENES DE PICKING:** Tabla de cabecera de órdenes de Picking. Se extraerán los registros de los pedidos de venta extraídos.
- **LINEAS DE PICKING:** Tabla de líneas de picking. Se extraerán todas las líneas correspondientes a las cabeceras cargadas en ORDENES DE PICKING.
- **UBICACIÓN DE PICKING:** Tabla de ubicaciones de picking. Se extraerá toda la tabla, al no disponer de ningún histórico del alta de ubicaciones.
- **UBICACIÓN DE ALMACENAJE:** Tabla de ubicaciones de almacenaje. Se extraerá toda la tabla, al no disponer de ningún histórico del alta de ubicaciones.

2. Tablas de dimensiones:

- **ALMACENES:** Tabla de almacenes. Se extraerá toda la información de la tabla.
- **ÁREA DESCARGA:** Tabla de área descarga. Se extraerá toda la información de la tabla.
- **ARTÍCULOS:** Tabla de artículos. Se extraerá toda la información de la tabla.
- **CLIENTES:** Tabla de clientes. Se extraerá toda la información de la tabla.
- **OPERARIOS:** Tabla de operarios. Se extraerá toda la información de la tabla.
- **PASILLOS:** Tabla de pasillos. Se extraerá toda la información de la tabla.
- **PROPIETARIOS:** Tabla de propietarios. Se extraerá toda la información de la tabla.

- **PROVEEDORES:** Tabla de proveedores. Se extraerá toda la información de la tabla.
- **TIPOS MOVIMIENTO:** Tabla de tipos de movimientos. Se extraerá toda la información de la tabla.
- **TIPOS PEDIDO PROVEEDOR:** Tabla de tipos de pedidos de proveedor. Se extraerá toda la información de la tabla.
- **TIPOS PEDIDO:** Tabla de tipos de pedidos de venta. Se extraerá toda la información de la tabla.
- **TIPOS PALETA:** Tabla de tipos de paleta. Se extraerá toda la información de la tabla.

Las tablas de dimensiones no tienen fecha por la que se controlará cuando se ha dado de alta un registro. El criterio que deben de seguir todas las tablas de dimensiones es que sus datos no serán borrados en ningún momento y serán mantenidos en las tablas históricas del Datawarehouse.

Como fuentes de datos secundarias, tenemos una base de datos procedente de un ERP, y otra procedente de un sistema de recursos humanos.

La base de datos del ERP es una base de datos relacional MySQL, de la cual extraeremos los datos de facturación de los pedidos de compra y venta. A continuación, se listan las modificaciones en las tablas de hechos y dimensiones definidas anteriormente:

- **CABECERA DE PEDIDO DE COMPRA:** Se añade un nuevo campo correspondiente al coste del pedido. Se extraerá la información de aquellos pedidos extraídos en InLOG WMS.
- **CABECERA DE PEDIDO DE VENTA:** Se añade un nuevo campo correspondiente al precio del pedido. Se extraerá la información de aquellos pedidos extraídos en InLOG WMS.
- **ARTICULOS.** Se añade un nuevo campo correspondiente al precio y coste del artículo. Se extraerá la información de aquellos artículos extraídos en InLOG WMS

La base de datos del sistema de recursos humanos es una base de datos relacional MySQL, de la cual extraeremos el coste/hora de los operarios. Modificaremos la tabla de dimensiones OPERARIOS, añadiendo un nuevo campo correspondiente al coste/hora.

6.2. Diseño del Datawarehouse

Una vez analizadas las distintas fuentes de datos que van a formar los datos almacenados en el Datawarehouse, el siguiente paso es realizar un diseño del modelo dimensional y el modo de carga de datos.

Existen dos modos diferentes de esquemas de base de datos para llevar a cabo el modelado dimensional del Datawarehouse: [19]

- Esquema en estrella: Está formado por una tabla de hechos, la cual contiene datos para el análisis, rodeada de tablas de dimensiones. [19]

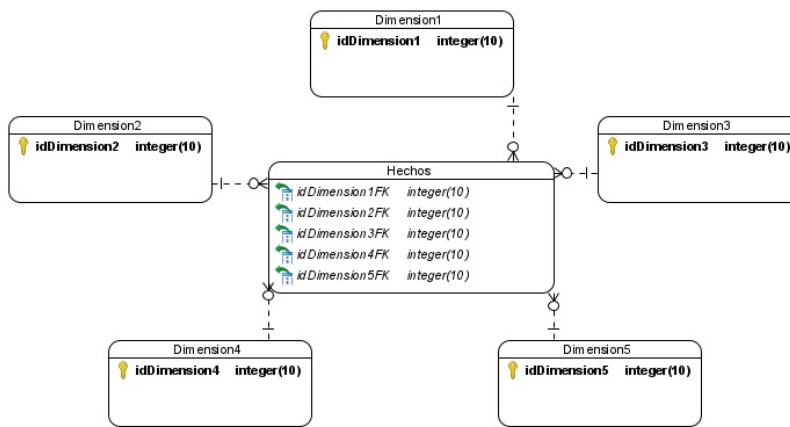


Figura 10: Esquema de estrella

Es el esquema utilizado habitualmente para modelar la base de datos. [20]

- Esquema en copo de nieve: Es una estructura mas compleja que el esquema estrella. Está formado por una tabla de hechos conectada con dimensiones anidadas. [19]

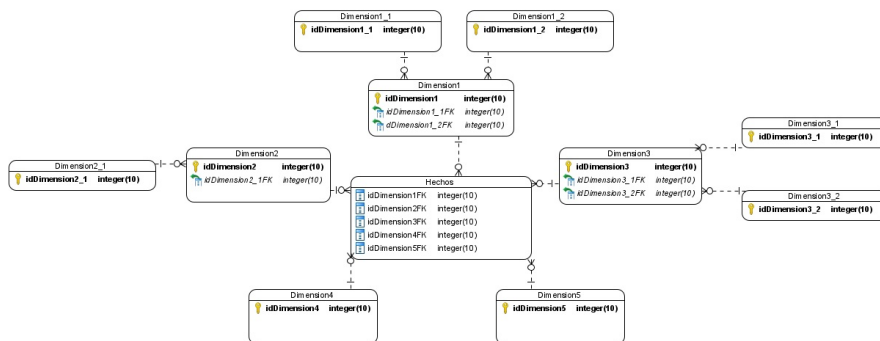


Figura 11: Esquema copo de nieve

Es utilizado cuando alguna de las dimensiones se implementa con más de una tabla de datos, normalizando así estas tablas y eliminando la redundancia entre los datos. [19]

En este caso, el Datawarehouse será diseñado con esquemas en copo de nieve, pues varias de las dimensiones están relacionadas entre si, y por ello, no puede utilizarse el esquema en estrella. Al crearse los esquemas en copo de nieve entorno a las tablas de hechos, necesitamos realizar un esquema diferente por cada una de las tablas de hechos.

Hasta las últimas versiones de Mondrian, el servidor OLAP de Pentaho, solo se permitía establecer la relación entre las tablas en función de un único campo, y debido a que Pentaho aun sigue trabajando con una versión mas antigua de este, necesitamos que todas las claves primarias de las tablas de dimensiones estén formadas por un solo campo, con lo cual, añadiremos un campo numérico de identificación en las siguientes tablas de dimensiones:

- **ARTÍCULOS:** Hasta ahora identificado por almacén, propietario, articulo y variables del articulo. Se añade un campo numérico ART_IDENTI para identificar el articulo.
- **ÁREA DESCARGA:** Hasta ahora identificado por almacén y código de área descarga. Se añade un campo numérico ADE_IDENTI para identificar el área de descarga.
- **PASILLOS:** Hasta ahora identificado por almacén y código de pasillo. Se añade un nuevo campo numérico PAS_IDENTI para identificar el pasillo.

Para simplificar los esquemas, únicamente mostramos los campos correspondiente a la clave primaria de la tabla, antiguas claves primarias de aquellas tablas de dimensiones de las cuales se ha añadido un campo de identificación, claves extranjeras y campos nuevos obtenidos en la carga de datos desde diferentes fuentes:

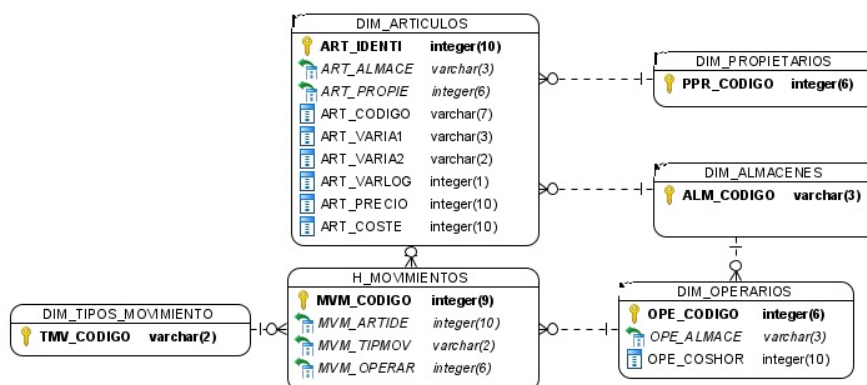


Figura 12: Esquema en copo de nieve de la tabla Movimientos

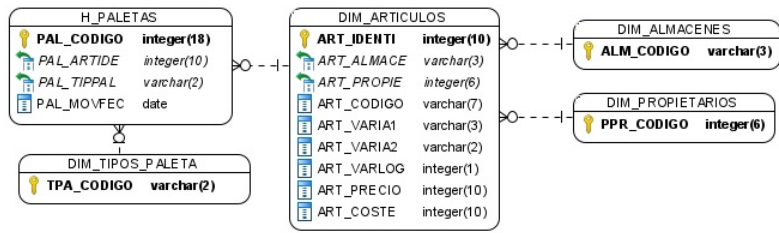


Figura 13: Esquema en copo de nieve de la tabla Paletas

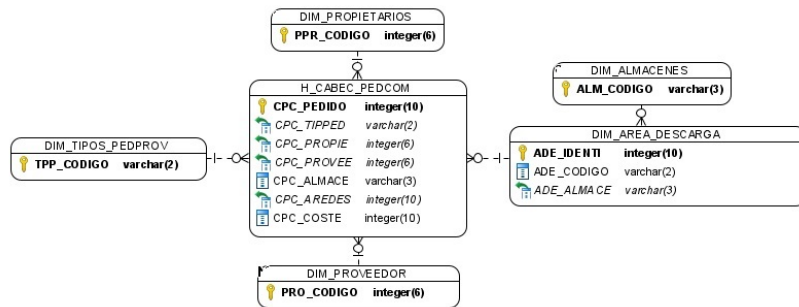


Figura 14: Esquema en copo de nieve de la tabla Cabecera de Pedido de Compra

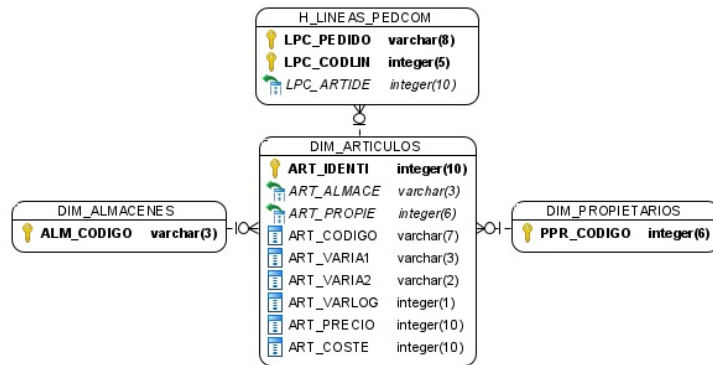


Figura 15: Esquema en copo de nieve de la tabla Lineas de Pedidos de Compra

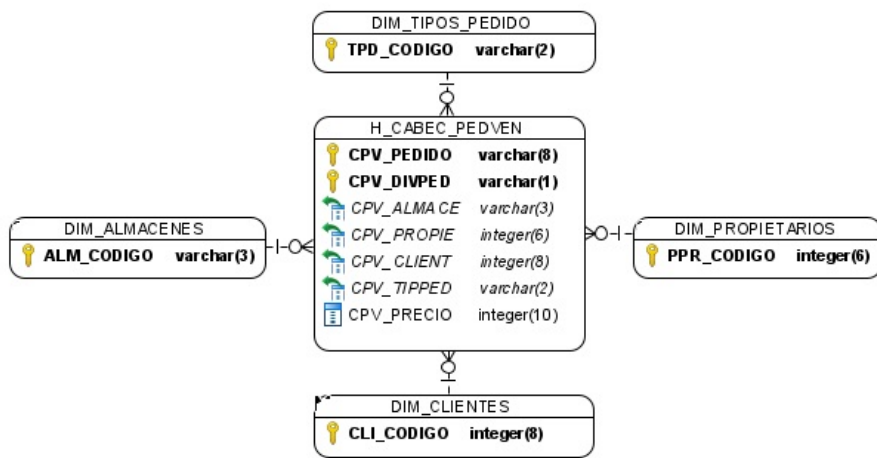


Figura 16: Esquema en copo de nieve de la tabla Cabecera de Pedido de Venta

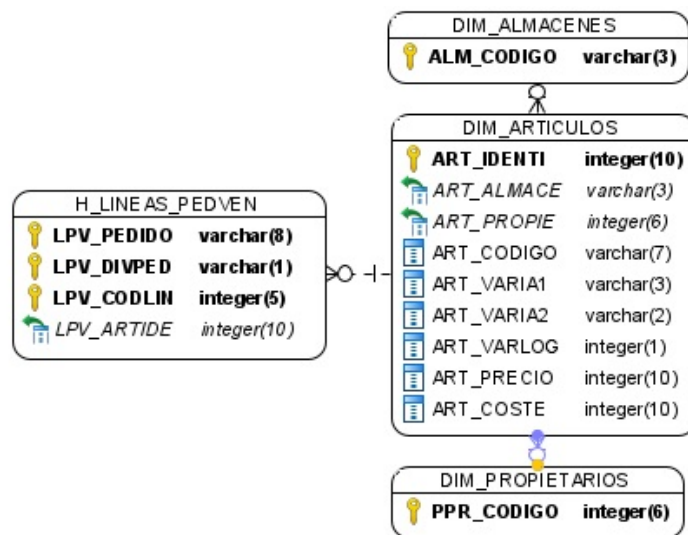


Figura 17: Esquema en copo de nieve de la tabla Lineas de Pedidos de Venta

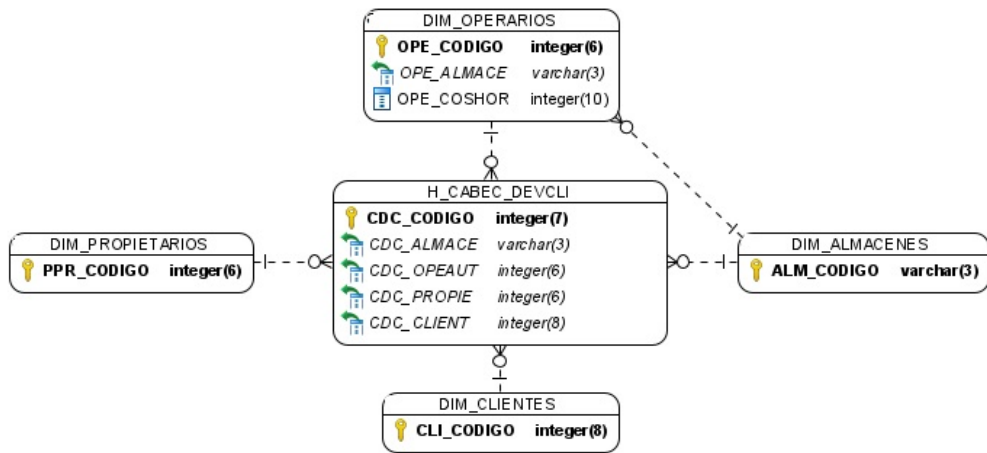


Figura 18: Esquema en copo de nieve de la tabla Cabecera Devolución Cliente

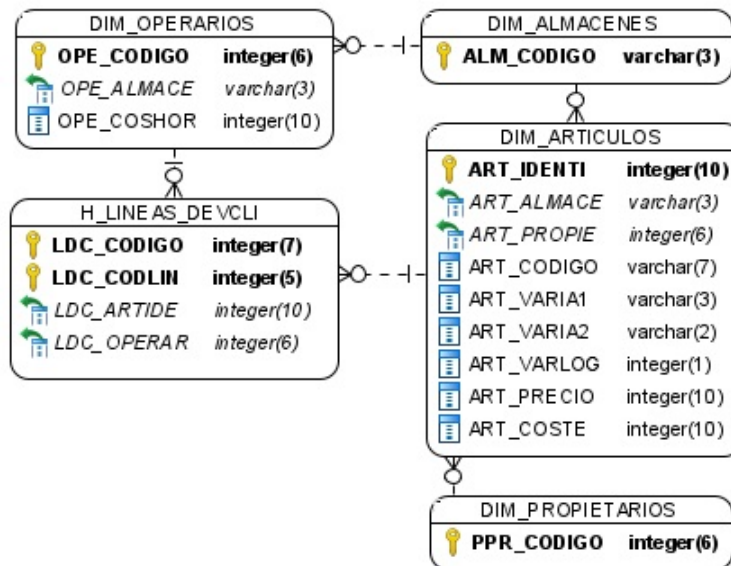


Figura 19: Esquema en copo de nieve de la tabla Lineas Devolución Cliente

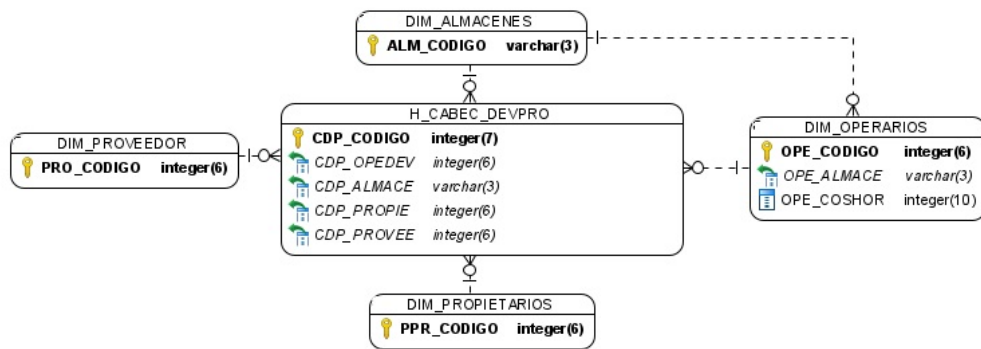


Figura 20: Esquema en copo de nieve de la tabla Cabecera Devolución Proveedor

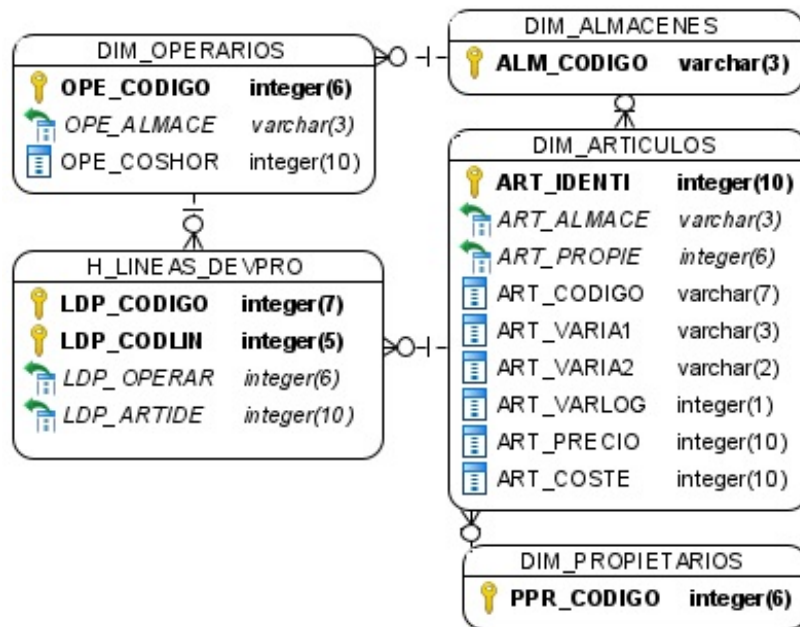


Figura 21: Esquema en copo de nieve de la tabla Lineas Devolución Proveedor

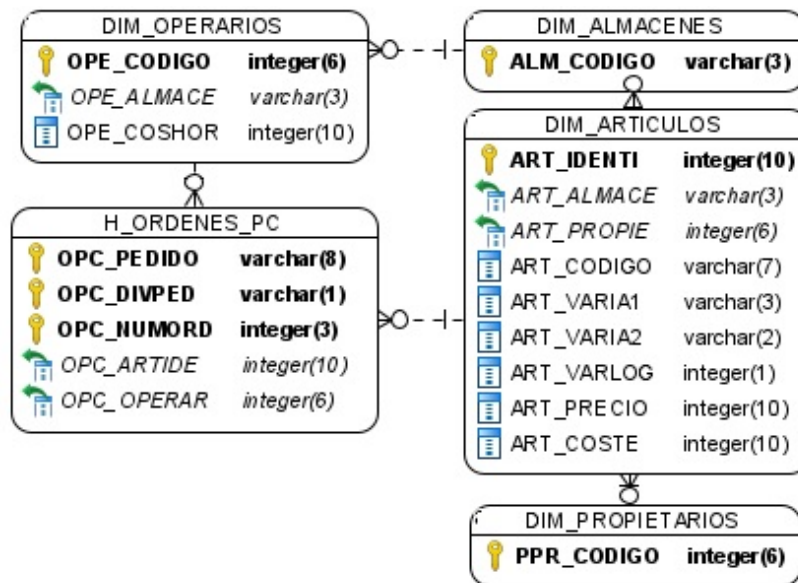


Figura 22: Esquema en copo de nieve de la tabla Ordenes de Palet Completo

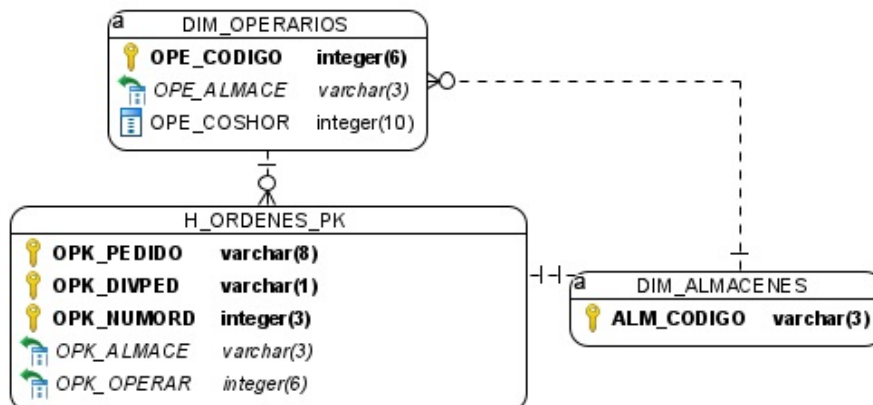


Figura 23: Esquema en copo de nieve de la tabla Ordenes de Picking

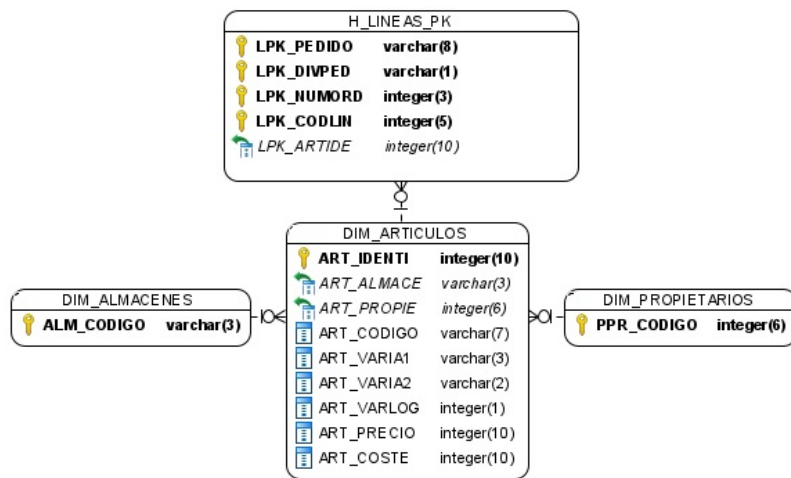


Figura 24: Esquema en copo de nieve de la tabla Lineas de Picking

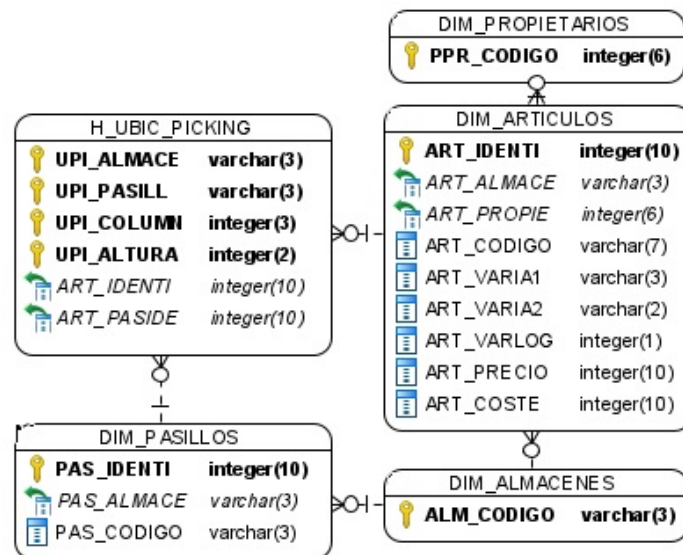


Figura 25: Esquema en copo de nieve de la tabla Ubicación de Picking

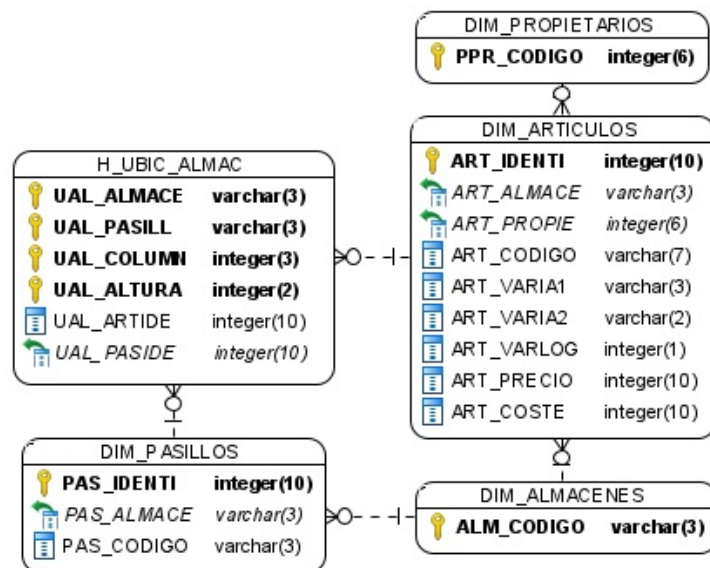


Figura 26: Esquema en copo de nieve de la tabla Ubicación de Almacenaje

6.3. Diseño de los cubos OLAP

Tras diseñar el Datawarehouse, se dispone en este de toda la información necesaria para diseñar los cubos OLAP, los cuales permitirán acceder a la información del Datawarehouse de una forma más rápida, permitiendo así realizar consultas complejas contra este.

Al igual que en el paso anterior, se realiza un cubo por cada una de las tablas de hechos. Para cada cubo se define las dimensiones que lo componen, ya sean compartidas o propias, y las medidas. Las dimensiones compartidas son aquellas dimensiones que son utilizadas por más de un cubo, y las dimensiones propias son aquellas dimensiones únicamente utilizadas en el cubo donde se definen.

En primer lugar, definimos las dimensiones compartidas. Por cada dimensión se definen las jerarquías con sus respectivos niveles:

- Dimensión Artículos
 - Jerarquía Artículos
 - Nivel Almacén
 - Nivel Propietario
 - Nivel Nombre
- Dimensión Operarios
 - Jerarquía Operarios
 - Nivel Nombre
 - Nivel Almacén

- Nivel Código
- Dimensión Propietarios
 - Jerarquía Propietarios
 - Nivel Nombre
 - Nivel Código
 - Nivel Localidad
 - Nivel Provincia
 - Nivel País
 - Nivel Código Postal
- Dimensión Proveedor
 - Jerarquía Proveedor
 - Nivel Nombre
 - Nivel Código
 - Nivel Localidad
 - Nivel Provincia
 - Nivel País
 - Nivel Código Postal
- Dimensión Cliente
 - Jerarquía Cliente
 - Nivel Nombre
 - Nivel Código
 - Nivel Localidad
 - Nivel Provincia
 - Nivel País
 - Nivel Código Postal
- Dimensión Almacenes
 - Jerarquía Almacenes
 - Nivel Código
 - Nivel Denominación
 - Nivel Localidad
 - Nivel Provincia
 - Nivel País
 - Nivel Código Postal
- Dimensión Pasillos
 - Jerarquía Pasillos
 - Nivel Almacén
 - Nivel Código
 - Nivel Denominación

o Nivel Sección

Una dimensión muy importante que debe estar en todo Datawarehouse es la dimensión tiempo, pues casi todos los análisis necesitan poder hacerse desde una perspectiva temporal. Esta dimensión suele estar relacionada con la mayoría de sus tablas de hechos, siendo frecuente que lo haga por mas de una columna. [29]

En este caso, en lugar de implementar la dimensión tiempo mediante el uso de una tabla física, se ha decidido implementarlo mediante una vista que calcule los distintos atributos correspondientes al tiempo. El rango de fechas que se calculará en la vista será desde el 1 de enero de 2020, hasta un mes después de la fecha del sistema en el momento de hacer la consulta.

La estructura de la vista es la siguiente:

V_DIM_TIEMPO		
CAMPO	FORMATO	DESCRIPCIÓN
FECHA	DATE	Fecha con formato DD/MM/YYYY
AÑO	NUMBER	Año de la fecha
MES	VARCHAR	Nombre del mes
DIA	VARCHAR	Nombre del día
SEMANA	NUMBER	Número de la semana
SEMESTRE	NUMBER	Número de semestre
TRIMESTRE	NUMBER	Número de trimestre
DIAJULIANO	NUMBER	Fecha juliana
FESTIVO	VARCHAR	Indica si es día festivo o no

Tabla 37: Estructura Vista Dimensión Tiempo

Para poder determinar si un día es festivo o no, se creará la tabla FESTIVOS, en la que cada cliente podrá introducir sus días que habitualmente son declarados festivos, y mediante un procedimiento almacenado se determinará si el día es festivo o no, teniendo en cuenta esta tabla y el día de la semana no laborable.

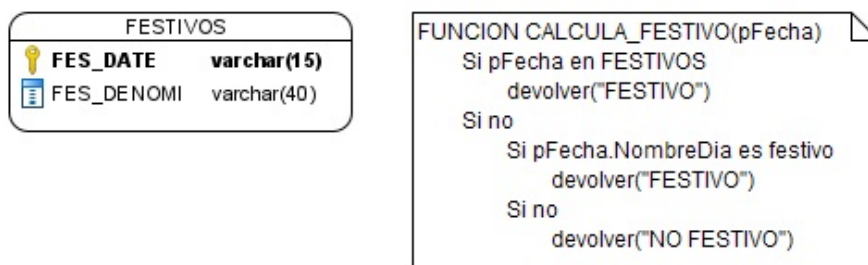


Figura 27: Esquema tabla Festivos y pseudocódigo procedimiento almacenado

A continuación, definimos la dimensión tiempo:

- Dimensión Tiempo
 - Jerarquía Tiempo
 - Nivel Año
 - Nivel Semestre
 - Nivel Trimestre
 - Nivel Mes
 - Nivel Semana
 - Nivel Día
 - Nivel Día Juliano
 - Nivel Festivo

Una vez definidas las dimensiones compartidas, ya podemos definir los cubos:

- Cubo Movimientos
 - Dimensión Artículos
 - Dimensión Operarios
 - Dimensión Tipo de Movimiento
 - Jerarquía Tipo de Movimiento
 - ◇ Nivel Código Movimiento
 - ◇ Nivel Nombre Movimiento
 - Dimensión Tiempo: Fecha Inicio
 - Dimensión Tiempo: Fecha fin
 - Medida Cantidad
 - Medida Número artículos diferentes
 - Medida Operarios diferentes
 - Medida Número movimientos
- Cubo Paletas
 - Dimensión Artículos
 - Dimensión Tipos de Paleta
 - Jerarquía Tipos de Paleta
 - ◇ Nivel Código
 - ◇ Nivel Nombre
 - ◇ Nivel Propietario
 - Medida Número paletas
 - Medida Número artículos diferentes
 - Medida Stock paleta
 - Medida Número movimientos de paleta
 - Medida Stock mínimo paleta
 - Medida Stock máximo paleta

- Cubo Cabecera de Pedido de Compra
 - Dimensión Propietarios
 - Dimensión Proveedor
 - Dimensión Tipos Pedido Proveedor
 - Jerarquía Tipos Pedido Proveedor
 - ◇ Nivel Código
 - ◇ Nivel Nombre
 - ◇ Nivel Propietario
 - Dimensión Área Descarga
 - Jerarquía Área Descarga
 - ◇ Nivel Código
 - ◇ Nivel Denominación
 - ◇ Nivel Almacén
 - Dimensión Tiempo: Fecha asignada a pedido
 - Medida Coste
 - Medida Media Coste
 - Medida Número de proveedores diferentes
 - Medida Número de paletas
 - Medida Operarios diferentes
 - Medida Número de pedidos
- Cubo Lineas de Pedido de Compra
 - Dimensión Artículos
 - Medida Cantidad pedida
 - Medida Cantidad recibida
 - Medida Número de paletas
 - Medida Número de lineas
 - Medida Número de pedidos
 - Medida Artículos diferentes
- Cubo Cabecera de Pedido de Venta
 - Dimensión Almacenes
 - Dimensión Propietarios
 - Dimensión Clientes
 - Dimensión Tipos Pedido
 - Jerarquía Tipos Pedido
 - ◇ Nivel Código
 - ◇ Nivel Denominación
 - ◇ Nivel Procesamiento automático
 - ◇ Nivel Agrupación automática

- Dimensión Tiempo: Fecha de transmisión
- Dimensión Tiempo: Fecha de servicio
- Medida Coste
- Medida Media coste
- Medida Número de paletas
- Medida Número de pedidos
- Medida Clientes diferentes
- Medida Peso paletas completas
- Medida Peso picking
- Medida Volumen paleta completa
- Medida Volumen picking
- Cubo Lineas de Pedido de Venta
 - Dimensión Artículos
 - Medida Cantidad pedida
 - Medida Cantidad servida
 - Medida Número de pedidos
 - Medida Artículos diferentes
- Cubo Cabecera Devolución Cliente
 - Dimensión Almacenes
 - Dimensión Propietarios
 - Dimensión Clientes
 - Dimensión Operarios
 - Dimensión Tiempo: Fecha devolución
 - Medida Clientes diferentes
 - Medida Operarios diferentes
 - Medida Muelles
 - Medida Número devoluciones
- Cubo Lineas Devolución Cliente
 - Dimensión Artículos
 - Dimensión Operarios
 - Medida Cantidad total
 - Medida Cantidad apta
 - Medida Cantidad no apta
 - Medida Número operarios
 - Medida Artículos diferentes

- Cubo Cabecera Devolución Proveedor
 - Dimensión Almacenes
 - Dimensión Propietarios
 - Dimensión Clientes
 - Dimensión Operarios
 - Dimensión Tiempo: Fecha devolución
 - Medida Proveedores diferentes
 - Medida Operarios diferentes
 - Medida Muelles
 - Medida Número devoluciones
- Cubo Lineas Devolución Proveedor
 - Dimensión Artículos
 - Dimensión Operarios
 - Medida Cantidad
 - Medida Número paletas
 - Medida Número operarios
 - Medida Artículos diferentes
- Cubo Ordenes de paleta completa
 - Dimensión Operarios
 - Dimensión Artículos
 - Medida Artículos diferentes
 - Medida Cantidad paletas pedidas
 - Medida Cantidad recibida
 - Medida Operarios diferentes
- Cubo Ordenes de Picking
 - Dimensión Operarios
 - Dimensión Almacenes
 - Medida Número pedidos
 - Medida Operarios diferentes
 - Medida Número de lineas
 - Medida Número de contenedores
 - Medida Volumen contenedores
 - Medida Peso contenedores

- Cubo Lineas de Picking
 - Dimensión Artículos
 - Medida Cantidad pedida
 - Medida cantidad servida
 - Medida Artículos diferentes
- Cubo Ubicación de Picking
 - Dimensión Pasillos
 - Dimensión Artículos
 - Medida Número paletas ubicadas
 - Medida Artículos diferentes
- Cubo Ubicación de Almacenaje
 - Dimensión Pasillos
 - Dimensión Artículos
 - Medida Número paletas ubicadas
 - Medida Artículos diferentes

6.4. Diseño de los informes predefinidos

Una vez construido el Datawarehouse, un paso para poder realizar un análisis sobre los datos extraídos a este es mediante los informes predefinidos:

1. INFORME PALETAS CADUCADAS:

Muestra los datos sobre aquellas paletas que han caducado en un periodo determinado.

Parámetros:

- Fecha inicio periodo (Obligatoria)
- Fecha fin periodo (Obligatoria)
- Código de propietario (Opcional)

Se mostrará el código de paleta, nombre de propietario, código de artículo, nombre de artículo, stock de paleta e importe de la paleta (multiplicación stock de paleta por precio del artículo). Los datos se obtendrán de las tablas de H.PALETAS, DIM_ARTÍCULOS Y DIM_PROPIETARIOS.

Se mostrará una fila de totales, calculando el total de las columnas stock de paleta e importe de la paleta.

Fecha report

INFORME PALETAS CADUCADAS

FECHA INICIO: Parametro fecha inicio
FECHA FIN: Parametro fecha fin

PALETA	PROPIETARIO	ARTICULO	NOMBRE ARTICULO	STOCK	IMPORTE
PAL_CODIGO	PPR_NOMBRE	ART_CODIGO	ART_DENOMI	PAL_STOCPA	ART_PRECIO * PAL_STOCPA
PAL_CODIGO	PPR_NOMBRE	ART_CODIGO	ART_DENOMI	PAL_STOCPA	ART_PRECIO * PAL_STOCPA
TOTAL				SUMA(STOCK)	SUMA(IMPORTE)

© In-LOG Consultoría y Soluciones S.L.

Figura 28: Diseño informe paletas caducadas

2. INFORME PEDIDOS DE COMPRA:

Muestra los datos de un pedido de compra.

Parámetros:

- Pedido (Obligatorio)

Se mostrará por un lado, en la cabecera del informe el código de pedido, tipo de pedido, fecha asignada del pedido al sistema, nombre de almacén, nombre de propietario, nombre de proveedor, número de paletas, número de líneas y situación del pedido. Los datos se obtendrán de las tablas H_CABEC_PEDCOM, DIM_TIPOS_PEDPROV, DIM_ALMACENES, DIM_PROPIETARIOS y DIM_PROVEEDOR.

En las líneas del pedido, se mostrará el número de línea, código de artículo, nombre de artículo, cantidad teórica a recibir, cantidad real recibida, situación de la línea y coste. Los datos se obtendrán de las tablas H_LINEAS_PEDCOM y DIM_ARTICULOS.

Se mostrará una fila de totales, calculando el total del coste del pedido.

Fecha report

INFORME PEDIDO DE COMPRA

Pedido: CPC_PEDIDO Tipo Pedido: TPP_DENOMI Fecha Asignada: CPC_FECASI
Almacén: ALM_DENOMI Propietario: PPR_NOMBRE Proveedor: PRO_NOMBRE
Número de paletas: CPC_NUMPAL Número de líneas: CPC_TOTLIN Situación: CPC_SITPED

LINEA	ARTICULO	NOMBRE ARTICULO	C. TEO	C REAL	SITUACION	COSTE
LPC_CODLIN	ART_CODIGO	ART_DENOMI	LPC_CANTEO	LPC_CANREA	LPC_SITLIN	ART_COSTE * LPC_CANTEO
LPC_CODLIN	ART_CODIGO	ART_DENOMI	LPC_CANTEO	LPC_CANREA	LPC_SITLIN	ART_COSTE * LPC_CANTEO
TOTAL						SUMA(COSTE)

© In-LOG Consultoría y Soluciones S.L.

Figura 29: Diseño informe pedidos de compra

3. INFORME PEDIDOS DE VENTA:

Muestra los datos de un pedido de venta.

Parámetros:

- Pedido (Obligatorio)
- División (Obligatorio)

Se mostrará por un lado, en la cabecera del informe el código de pedido, tipo de pedido, fecha de transmisión del pedido al sistema, nombre de almacén, nombre de propietario, nombre del cliente, nombre de operario y situación. Los datos se obtendrán de las tablas H.CABEC_PEDVEN, DIM.TIPOS_PEDIDO, DIM.ALMACENES, DIM.PROPIETARIOS, DIM.CLIENTES y DIM.OPERARIOS.

En las líneas del pedido, se mostrará el número de línea, código de artículo, nombre de artículo, cantidad pedida, formato de la cantidad pedida, cantidad servida (siempre en unidades), situación de la línea, coste de compra del artículo, coste de la preparación de la línea y precio. Los datos se obtendrán de las tablas H.LINEAS_PEDVEN, H.LINEAS_PK, DIM.ARTICULOS y DIM.OPERARIOS.

Se mostrará una fila de totales, calculando el total del coste de compra de los artículos, coste de la preparación de las líneas y precios. Además, se mostrará una fila indicando los beneficios obtenidos con el pedido.

Fecha report

INFORME PEDIDO VENTA

Pedido: CPV_PEDIDO + "/" + CPV_DIVPED Tipo Pedido: TPD_DENOMI Fecha Transmision: CPV_FECTRA
 Almacén: ALM_DENOMI Propietario: PPR_NOMBRE Cliente: CU_NOMBRE
 Operario: OPE_NOMBRE Situación: CPV_SITPED

LINEA	ARTICULO	NOMBRE ARTICULO	C.PED F.	C.SER	SITUACION C.ART	C.PREP	PRECIO
LPV_CODLIN	ART_CODIGO	ART_DENOMI	LPV_CANPED	LPV_CANSER	LPV_SITLIN	ART_COSTE * LPV_CANSER	ART_PRECIO * LPV_CANSER
LPV_CODLIN	ART_CODIGO	ART_DENOMI	LPV_CANPED	LPV_CANSER	LPV_SITLIN	ART_COSTE * LPV_CANSER	ART_PRECIO * LPV_CANSER
TOTAL						SUMA(C.ART)	SUMA(PRECIO)
BENEFICIOS:						SUMA(PRECIO) - SUMA(C.ART) - SUMA(C.OPE)	

© In-LOG Consultoría y Soluciones S.L.

Figura 30: Diseño informe pedidos de venta

4. INFORME PRODUCTIVIDAD OPERARIOS:

Muestra los datos de productividad sobre los operarios del almacén en un periodo determinado

Parámetros:

- Fecha inicio periodo (Obligatoria)
- Fecha fin periodo (Obligatoria)
- Código de operario (Opcional)

Se mostrará el nombre de operario, número de movimientos realizados, total de tiempo empleado en formato (HH:MI:SS) y coste de operario. Los datos se obtendrán de las tablas de H_MOVIMIENTOS Y DIM_OPERARIOS.

Fecha report

PRODUCTIVIDAD OPERARIOS

FECHA INICIO: Parametro fecha inicio
FECHA FIN: Parametro fecha fin

NOMBRE	MOVIMIENTOS	TIEMPO	COSTE
OPE_NOMBRE	COUNT(MVM_CODIGO)	SUM(MVM_FEHOFI - MVM_FEHOIN)	TIEMPO (en horas) * OPE_COSHOR

© In-LOG Consultoría y Soluciones S.L.

Figura 31: Diseño informe productividad operarios

7. Desarrollo

Para el desarrollo del proyecto, se ha utilizado una maquina con el sistema operativo CentOS y el cliente Oracle Database 11g instalado.

La administración de la maquina se realizará utilizando el protocolo SSH, con el cual se podrá acceder de forma remota al servidor y además realizar transferencias de archivos a este, mientras que la administración de la base de datos instalada en la maquina se realizará mediante el entorno PL/SQL Developer. Para diferenciar el acceso las diferentes funcionalidades instaladas en la maquina, tenemos los siguientes usuarios:

- ORACLE: Permite gestionar la base de datos Oracle.
- ETL: Permite gestionar el proceso de carga de datos al Datawarehouse.
- SERVER: Permite gestionar el servidor web de Pentaho.

7.1. Creación del esquema de base de datos

El primer paso en el desarrollo es crear el esquema de base de datos correspondiente al Datawarehouse.

En Oracle, un schema contiene todos los objetos creados por un usuario específico de la base de datos. Esos objetos pueden incluir tablas, vistas, sinónimos, triggers... [28]

Para crear el schema, nos conectamos a la maquina mediante SSH con el usuario Oracle, y accedemos a la base de datos en modo administrador ejecutando el siguiente comando:

```
sqlplus "/as_sysdba"
```

En este caso, el Datawarehouse se creará en un schema que llamaremos *pentaho*. Para ello, ejecutaremos el siguiente comando:

```
CREATE USER pentaho IDENTIFIED BY pentaho;
```

El siguiente paso es asignar permisos de conexión a base de datos, utilizando los siguientes comandos:

```
GRANT "CONNECT" TO pentaho;  
GRANT "RESOURCE" TO pentaho;
```

Finalmente, asignamos permisos específicos para permitir realizar todas las operaciones necesarias sobre objetos:

```
GRANT ALTER ANY INDEX TO pentaho;  
GRANT ALTER ANY SEQUENCE TO pentaho;  
GRANT ALTER ANY TABLE TO pentaho;  
GRANT ALTER ANY TRIGGER TO pentaho;  
GRANT CREATE ANY INDEX TO pentaho;  
GRANT CREATE ANY SEQUENCE TO pentaho;  
GRANT CREATE ANY TABLE TO pentaho;  
GRANT CREATE ANY TRIGGER TO pentaho;
```

```

GRANT CREATE ANY VIEW TO pentaho;
GRANT CREATE PROCEDURE TO pentaho;
GRANT CREATE TRIGGER TO pentaho;
GRANT CREATE VIEW TO pentaho;
GRANT DELETE ANY TABLE TO pentaho;
GRANT DROP ANY INDEX TO pentaho;
GRANT DROP ANY SEQUENCE TO pentaho;
GRANT DROP ANY TABLE TO pentaho;
GRANT DROP ANY TRIGGER TO pentaho;
GRANT DROP ANY VIEW TO pentaho;
GRANT INSERT ANY TABLE TO pentaho;
GRANT QUERY REWRITE TO pentaho;
GRANT SELECT ANY TABLE TO pentaho;

```

Una vez creado el schema, ya podemos acceder a este mediante PL/SQL Developer.

El siguiente paso es crear la estructura de tablas de hechos y dimensiones. Para facilitar una posible futura implementación en otros clientes, realizaremos un script de instalación de base de datos, donde guardaremos todas las instrucciones necesarias para crear la estructura del datawarehouse.

Utilizando la instrucción CREATE TABLE, creamos todas las tablas necesarias para formar el Datwarehouse, añadiendo el prefijo H a las tablas de hechos, y añadiendo el prefijo DIM a las tablas de dimensiones.

Una vez creado el script de instalación, lo ejecutamos en la base de datos utilizando PL/SQL Developer.

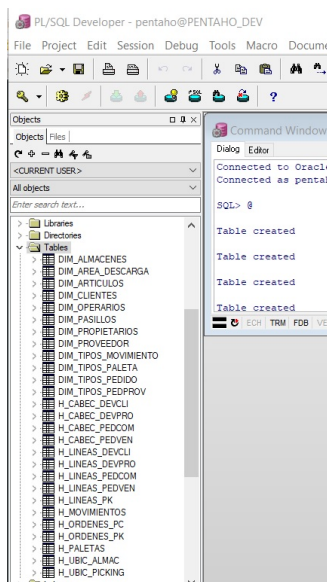


Figura 32: Vista esquema DWH desde PL/SQL Developer

7.2. Implementación del proceso ETL

El proceso ETL se realizará mediante la herramienta Pentaho Data Integration (PDI), incluida en la suite de Pentaho CE.

Esta herramienta proporciona una interfaz gráfica en la que se pueden diseñar transformaciones (conjunto de pasos para realizar la transformación desde el origen de datos hasta el Datawarehouse) y jobs (definir la ejecución de transformaciones)

A continuación, se muestran las transformaciones implementadas teniendo en cuenta las condiciones de carga de datos definidas anteriormente para cada una de las tablas:

1. TABLA DIMENSIONES ARTÍCULOS



Figura 33: Transformación de la tabla de dimensiones Artículos

La transformación de la tabla ARTÍCULOS consiste en unir las tablas procedentes del ERP y WMS, y además insertar un código numérico secuencial que servirá para identificar al artículo, al estar este originalmente identificado por varios campos.

Se definen los siguientes pasos:

- ARTÍCULOS ERP: Extrae la información de la tabla ARTÍCULOS del ERP.
- ARTÍCULOS WMS: Extrae la información de la tabla ARTÍCULOS de InLOG WMS.
- Unir: Une la información de la tabla ARTÍCULOS extraída del ERP y ARTÍCULOS extraída de InLOG WMS.
- Selecciono valores: Selecciona aquellos campos a insertar, eliminando aquellos campos del ERP que no son necesarios.
- Inserto secuencia: Añade el campo numérico secuencial para identificar cada artículo.
- Insertar / Actualizar Artículos: Inserta los datos a la tabla histórica de ARTÍCULOS del Datawarehouse.

2. TABLA DIMENSIONES ÁREA DESCARGA Y PASILLOS

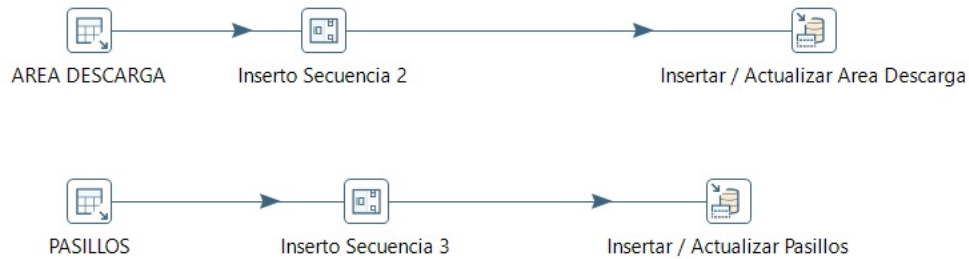


Figura 34: Transformación de las tablas de dimensiones Área descarga y Pasillos

Se definen los siguientes pasos:

- **ÁREA DESCARGA:** Extrae la información de la tabla **ÁREA DESCARGA** de InLOG WMS.
- **Insero Secuencia 2:** Añade el campo numérico secuencial para identificar cada área de descarga.
- **Insertar / Actualizar Área Descarga:** Inserta los datos a la tabla de dimensión **ÁREA DESCARGA** del Datawarehouse.
- **PASILLOS:** Extrae la información de la tabla **PASILLOS** de InLOG WMS.
- **Insero Secuencia 3:** Añade el campo numérico secuencial para identificar cada pasillo.
- **Insertar / Actualizar Almacenes:** Inserta los datos a la tabla de dimensión **PASILLOS** del Datawarehouse.

3. TABLA HECHOS MOVIMIENTOS:

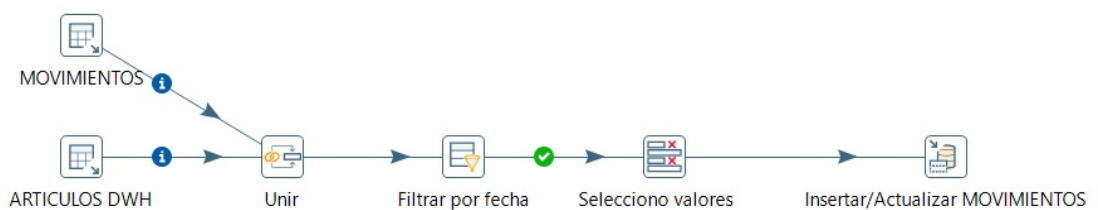


Figura 35: Transformación de la tabla de hechos Movimientos

La transformación de la tabla **MOVIMIENTOS** consiste en cargar aquellos movimientos cuya fecha fin de movimiento es superior a la fecha estipulada. Debido a que el movimiento contiene información sobre artículos, es necesario obtener el código de identificación de este desde la nueva tabla de **ARTICULOS** en el Datawarehouse.

Se definen los siguientes pasos:

- MOVIMIENTOS: Extrae la información de la tabla movimientos de InLOG WMS
- ARTICULOS DWH: Extrae la información de la tabla ARTÍCULOS del Datawarehouse
- Unir: Une los datos de los movimientos con los artículos del DWH para obtener el código de identificación del artículo.
- Filtrar por fecha: Filtra aquellas filas en las que la fecha fin del movimiento es superior al 1/01/2021
- Selecciono valores: Selecciona aquellos valores a insertar, eliminando los campos de referencia al artículo antiguos y añadiendo el código numérico de identificación del articulo como referencia.
- Insertar/Actualizar MOVIMIENTOS: Inserta los datos a la tabla histórica de MOVIMIENTOS del Datawarehouse

4. TABLA HECHOS PALETAS:

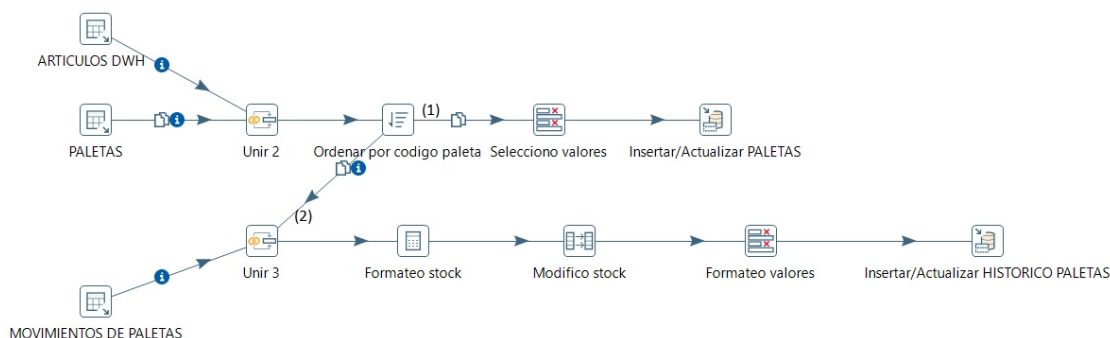


Figura 36: Transformación de la tabla de hechos Paletas

Por un lado, insertamos el valor actual de la tabla PALETAS, sin realizar ningún cambio (1), y por otro lado, obtendremos los cambios de stock que ha sufrido la paleta, realizando un cruce con la tabla movimientos (2)

Se definen los siguientes pasos:

- ARTICULOS DWH: Extrae la información de la tabla ARTÍCULOS del Datawarehouse.
- PALETAS: Extrae la información de la tabla paletas de InLOG WMS.
- Unir 2: Une los datos de las paletas con los artículos del DWH para obtener el código de identificación del artículo.
- Ordenar por código paleta: Ordena los datos por código de paleta.
- Selecciono valores: Selecciona aquellos valores a insertar, eliminando los campos de referencia al artículo antiguos y añadiendo el código numérico de identificación del articulo como referencia.
- Insertar / Actualizar Paletas: Inserta los datos a la tabla histórica de PALETAS del Datawarehouse.

- **MOVIMIENTOS DE PALETAS:** Extrae la información de aquellos movimientos relacionado con paletas de InLOG WMS
- **Unir 2:** Une los datos de las paletas con los movimientos realizados sobre estas.
- **Formateo stock:** Calcula el nuevo stock de la paleta y formatea el tipo de datos para hacerlo compatible con el campo STOCK DE PALETA.
- **Modifico stock:** Actualiza el campo STOCK DE PALETA introduciendo el nuevo stock calculado.
- **Formateo valores:** Selecciona únicamente aquellos campos correspondientes a la tabla PALETAS, eliminando la información de los movimientos.
- **Insertar/Actualizar HISTORICO PALETAS:** Inserta los datos a la tabla histórica de PALETAS del Datawarehouse.

5. TABLAS HECHOS PEDIDOS DE COMPRA

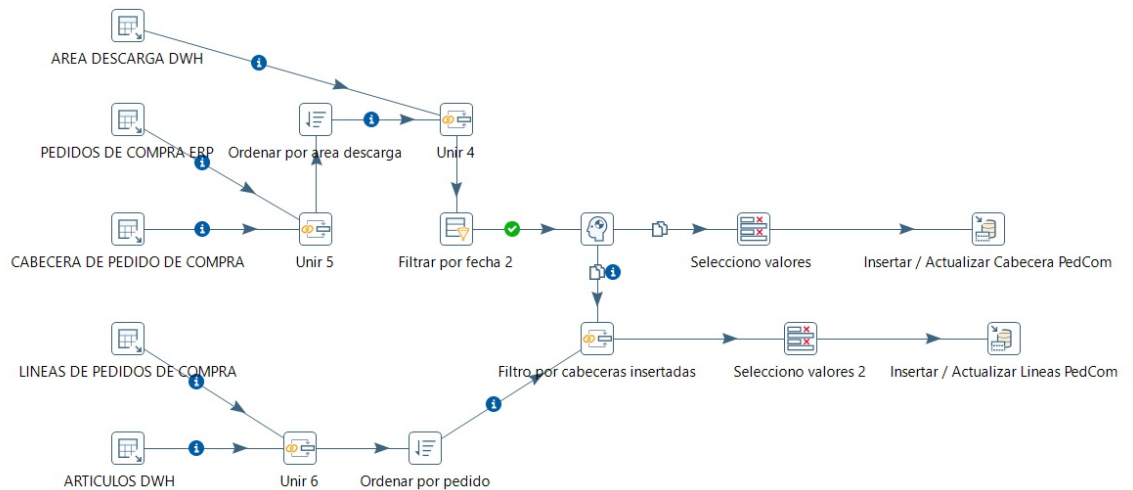


Figura 37: Transformación de las tablas de hechos Pedidos de Compra

La transformación consiste en, por un lado cargar aquellos pedidos de compra uniendo las tablas procedentes del ERP y WMS cuya fecha de alta de pedido sea superior a la fecha estipulada. Debido a que las cabeceras contienen información del área de descarga, es necesario obtener el código secuencial que las identifica.

Por otro lado, en el caso de las líneas de compra, se cargarán aquellas correspondientes a los pedidos cargados en las cabeceras, por ello será necesario aplicar un filtro consultando las cabeceras. Debido a que las líneas contienen información de los artículos, es necesario obtener el código secuencial que los identifica.

Se definen los siguientes pasos:

- AREA DESCARGA DWH: Extrae la información de la tabla ÁREA DESCARGA del Datawarehouse.
- PEDIDOS DE COMPRA ERP: Extrae la información de la tabla PEDIDOS COMPRA del ERP.
- CABECERA DE PEDIDO DE COMPRA: Extrae la información de la tabla CABECERA DE PEDIDO DE COMPRA de InLOG WMS.
- Unir 4: Une los datos de las cabeceras de pedido de compra con las áreas de descarga del Datawarehouse para obtener el código de identificación del área de descarga.
- Unir 5: Une la información de la tabla PEDIDOS COMPRA extraída del ERP y CABECERA DE PEDIDO DE COMPRA extraída de InLOG WMS.
- Ordenar por área descarga: Ordena los campos en función del área de descarga para poder unir con ÁREA DESCARGA DWH
- Filtrar por fecha 2: Filtra aquellas filas en las que la fecha de alta del pedido en InLOG WMS es superior al 1/01/2021.
- Selecciono valores: Selecciona aquellos campos a insertar, eliminando aquellos campos del ERP que no son necesarios.
- Insertar / Actualizar Cabecera PedCom. Inserta los datos a la tabla histórica de CABECERA DE PEDIDO DE COMPRA del Datawarehouse.
- LINEAS DE PEDIDOS DE COMPRA: Extrae la información de la tabla LINEAS DE PEDIDOS DE COMPRA de InLOG WMS.
- ARTICULOS DWH: Extrae la información de la tabla ARTÍCULOS del Datawarehouse.
- Unir 6: Une los datos de las líneas de pedido de compra con los artículos del Datawarehouse para obtener el código de identificación del artículo.
- Ordenar por pedido: Ordena los campos en función del pedido.
- Filtro por cabeceras insertadas: Filtra aquellas líneas de pedidos de las cuales se han extraído las cabeceras.
- Selecciono valores 2: Selecciona aquellos campos a insertar.
- Insertar / Actualizar Lineas PedCom: Inserta los datos a la tabla histórica de LINEAS DE PEDIDO DE COMPRA del Datawarehouse.

6. TABLAS HECHOS PEDIDOS DE VENTA, ORDENES DE PALET COMPLETO Y PICKING

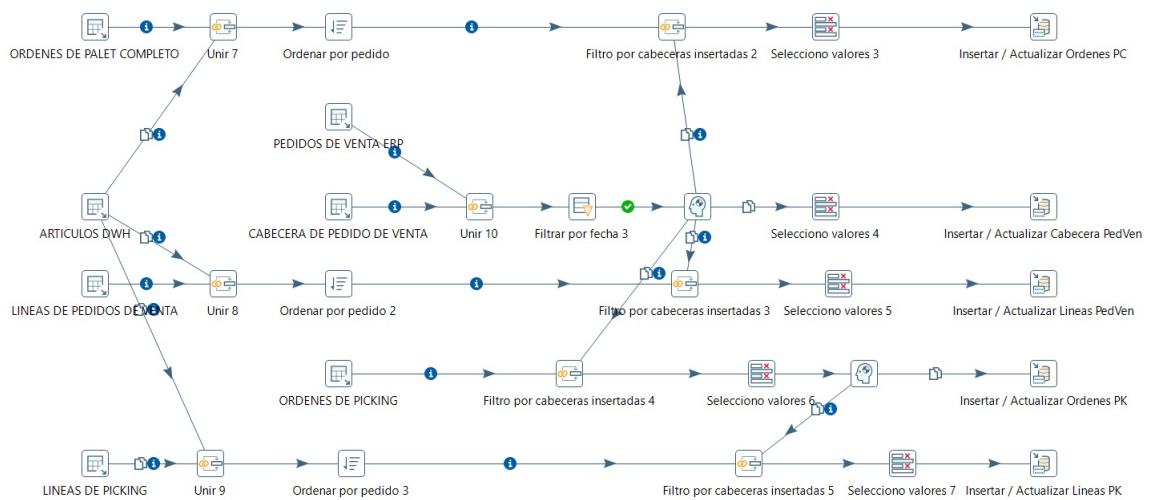


Figura 38: Transformación de las tablas de hechos Pedidos de Venta, Ordenes de Palet Completo y Picking

Se definen los siguientes pasos:

- **ORDENES DE PALET COMPLETO:** Extrae la información de la tabla ORDENES DE PALET COMPLETO de InLOG WMS.
- **ARTICULOS DWH:** Extrae la información de la tabla ARTÍCULOS del Datawarehouse.
- **Unir 7:** Une las ordenes de palet completo con los artículos del Datawarehouse para obtener el código de identificación del artículo.
- **Ordenar por pedido:** Ordena los campos en función del pedido.
- **Filtro por cabeceras insertadas 2:** Filtra aquellas ordenes de picking de las cuales se han extraído los pedidos de venta asociados.
- **Selecciono valores 3:** Selecciona aquellos campos a insertar.
- **Insertar / Actualizar Ordenes PC:** Inserta los datos a la tabla histórica de ORDENES DE PALET COMPLETO del Datawarehouse.
- **PEDIDOS DE VENTA ERP:** Extrae la información de la tabla PEDIDOS VENTA del ERP.
- **CABECERA DE PEDIDO DE VENTA:** Extrae la información de la tabla CABECERA DE PEDIDO DE VENTA de InLOG WMS.
- **Unir 3:** Une la información de la tabla PEDIDOS VENTA extraída del ERP y CABECERA DE PEDIDO DE VENTA extraída de InLOG WMS
- **Filtrar por fecha 3:** Filtra aquellas filas en las que la fecha de alta del pedido en InLOG WMS es superior al 1/01/2021.

- Selecciono valores 4: Selecciona aquellos campos a insertar, eliminando aquellos campos del ERP que no son necesarios.
- Insertar / Actualizar Cabecera Pedven Inserta los datos a la tabla histórica de CABECERA DE PEDIDO DE VENTA del Datawarehouse.
- LINEAS DE PEDIDOS DE VENTA: Extrae la información de la tabla LINEAS DE PEDIDO DE VENTA de InLOG WMS.
- Unir 8: Une las lineas de pedidos de venta con los artículos del Datawarehouse para obtener el código de identificación del articulo.
- Ordenar por pedido 2: Ordena los campos en función del pedido.
- Filtro por cabeceras insertadas 3: Filtra aquellas líneas de pedidos de las cuales se han extraído las cabeceras.
- Selecciono valores 5: Selecciona aquellos campos a insertar.
- Insertar / Actualizar Lineas Pedven Inserta los datos a la tabla histórica de LINEAS DE PEDIDO DE VENTA del Datawarehouse.
- ORDENES DE PICKING: Extrae la información de la tabla ORDENES DE PICKING de InLOG WMS.
- Filtro por cabeceras insertadas 4: Filtra aquellas ordenes de picking de las cuales se han extraído las cabeceras.
- Selecciono valores 6: Selecciona aquellos campos a insertar.
- Insertar / Actualizar Ordenes PK Inserta los datos a la tabla histórica de ORDENES DE PICKING del Datawarehouse.
- LINEAS DE PICKING: Extrae la información de la tabla LINEAS DE PICKING de InLOG WMS.
- Unir 9: Une las lineas de picking con los artículos del Datawarehouse para obtener el código de identificación del articulo.
- Ordenar por pedido 3: Ordena los campos en función del pedido.
- Filtro por cabeceras insertadas 5: Filtra aquellas líneas de picking de las cuales se han extraído las ordenes de picking.
- Selecciono valores 7: Selecciona aquellos campos a insertar.
- Insertar / Actualizar Lineas PK Inserta los datos a la tabla histórica de LINEAS DE PICKING del Datawarehouse.

7. TABLAS HECHOS DEVOLUCIONES CLIENTE

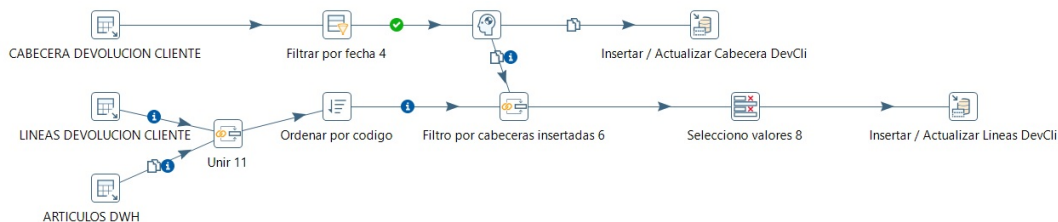


Figura 39: Transformación de las tablas de hechos Devoluciones Cliente

Se definen los siguientes pasos:

- **CABECERA DEVOLUCIÓN CLIENTE:** Extrae la información de la tabla CABECERA DEVOLUCIÓN CLIENTE de InLOG WMS.
- **Filtrar por fecha 4:** Filtra aquellas filas en las que la fecha de alta de la devolución en InLOG WMS es superior al 1/01/2021.
- **Insertar / Actualizar Cabecera DevCli** Inserta los datos a la tabla histórica de CABECERA DEVOLUCIÓN CLIENTE del Datawarehouse.
- **LINEAS DEVOLUCIÓN CLIENTE:** Extrae la información de la tabla LINEAS DEVOLUCION CLIENTE de InLOG WMS.
- **ARTICULOS DWH:** Extrae la información de la tabla ARTÍCULOS del Datawarehouse.
- **Unir 11:** Une las líneas de devolución cliente con los artículos del Datawarehouse para obtener el código de identificación del artículo.
- **Ordenar por código:** Ordena los campos en función del código de devolución.
- **Filtro por cabeceras insertadas 6:** Filtra aquellas líneas de devolución de cliente de las cuales se han extraído las cabeceras.
- **Selecciono valores 8:** Selecciona aquellos campos a insertar.
- **Insertar / Actualizar Lineas DevCli** Inserta los datos a la tabla histórica de LINEAS DEVOLUCIÓN CLIENTE del Datawarehouse.

8. TABLAS HECHOS DEVOLUCIONES PROVEEDOR

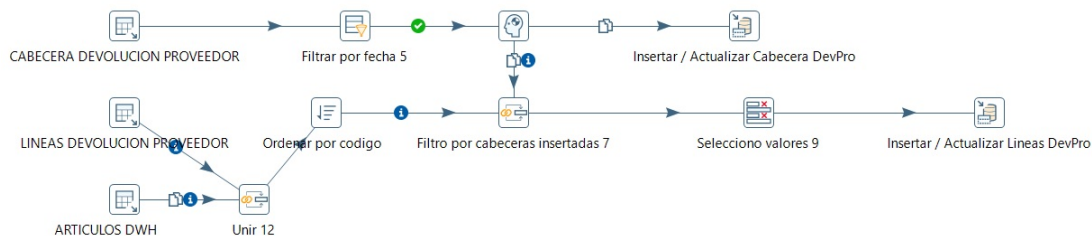


Figura 40: Transformación de las tablas de hechos Devoluciones Proveedor

Se definen los siguientes pasos:

- **CABECERA DEVOLUCIÓN PROVEEDOR:** Extrae la información de la tabla CABECERA DEVOLUCIÓN PROVEEDOR de InLOG WMS.
- **Filtrar por fecha 5:** Filtra aquellas filas en las que la fecha de alta de la devolución en InLOG WMS es superior al 1/01/2021.
- **Insertar / Actualizar Cabecera DevPro** Inserta los datos a la tabla histórica de CABECERA DEVOLUCIÓN PROVEEDOR del Datawarehouse.
- **LINEAS DEVOLUCIÓN PROVEEDOR:** Extrae la información de la tabla LINEAS DEVOLUCIÓN PROVEEDOR de InLOG WMS.
- **ARTICULOS DWH:** Extrae la información de la tabla ARTÍCULOS del Datawarehouse.
- **Unir 12:** Une las líneas de devolución proveedor con los artículos del Datawarehouse para obtener el código de identificación del artículo.
- **Ordenar por código:** Ordena los campos en función del código de devolución.
- **Filtro por cabeceras insertadas 7:** Filtra aquellas líneas de devolución de proveedor de las cuales se han extraído las cabeceras.
- **Selecciono valores 9:** Selecciona aquellos campos a insertar. Se eliminan las referencias antiguas a los artículos y se añade el nuevo código de identificación del artículo.
- **Insertar / Actualizar Lineas DevPro** Inserta los datos a la tabla histórica de LINEAS DEVOLUCIÓN PROVEEDOR del Datawarehouse.

9. TABLA HECHOS UBICACIONES DE PICKING

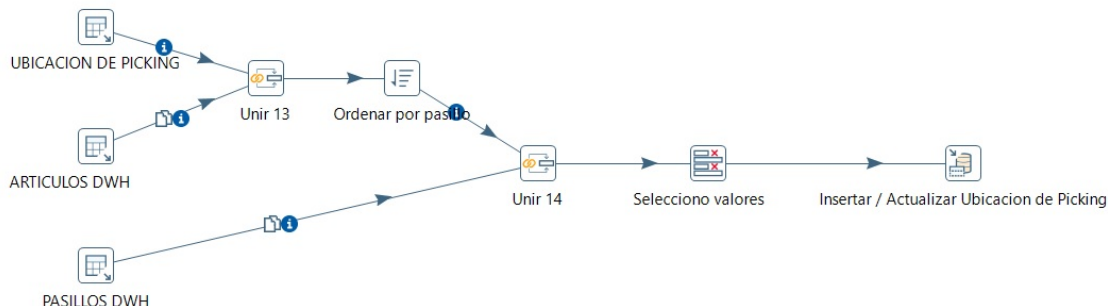


Figura 41: Transformación de las tablas de hechos Ubicación Picking

Se definen los siguientes pasos:

- **UBICACIÓN DE PICKING:** Extrae la información de la tabla UBICACIÓN DE PICKING de InLOG WMS.
- **ARTICULOS DWH:** Extrae la información de la tabla ARTÍCULOS del Datawarehouse.
- **Unir 13:** Une las ubicaciones de picking con los artículos del Datawarehouse para obtener el código de identificación del artículo.
- **Ordenar por pasillo:** Ordena las ubicaciones de picking en función del pasillo para poder unir con la tabla pasillos.
- **Unir 14:** Une las ubicaciones de picking con los pasillos del Datawarehouse para obtener el código de identificación del artículo.
- **Selecciono valores:** Selecciona aquellos campos a insertar. Se eliminan las referencias antiguas a los artículos y pasillos y se añaden los nuevos códigos de identificación del artículo y pasillo.
- **Insertar / Actualizar Ubicación de picking:** Inserta los datos a la tabla histórica de UBICACIÓN DE PICKING del Datawarehouse.

10. TABLA HECHOS UBICACIONES DE ALMACENAJE

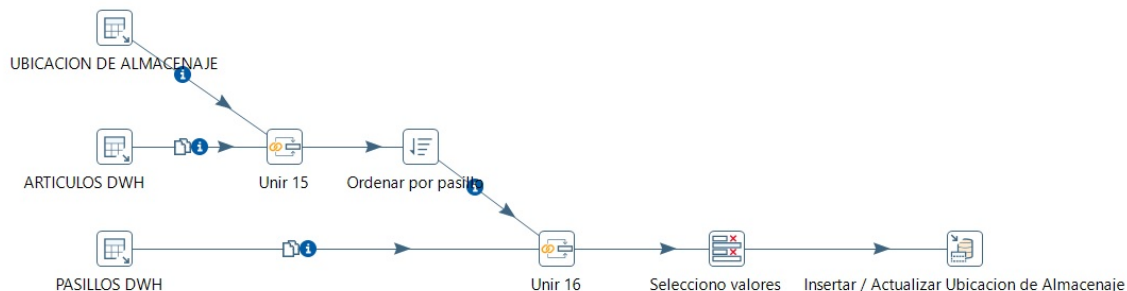


Figura 42: Transformación de las tablas de hechos Ubicación de Almacenaje

Se definen los siguientes pasos:

- UBICACIÓN DE ALMACENAJE: Extrae la información de la tabla UBICACIÓN DE ALMACENAJE de InLOG WMS.
- ARTICULOS DWH: Extrae la información de la tabla ARTÍCULOS del Datawarehouse.
- Unir 15: Une las ubicaciones de almacenaje con los artículos del Datawarehouse para obtener el código de identificación del artículo.
- Ordenar por pasillo: Ordena las ubicaciones de almacenaje en función del pasillo para poder unir con la tabla pasillos.
- Unir 16: Une las ubicaciones de almacenaje con los pasillos del Datawarehouse para obtener el código de identificación del artículo.
- Selecciono valores: Selecciona aquellos campos a insertar. Se eliminan las referencias antiguas a los artículos y pasillos y se añaden los nuevos códigos de identificación del artículo y pasillo.
- Insertar / Actualizar Ubicación de almacenaje: Inserta los datos a la tabla histórica de UBICACIÓN DE ALMACENAJE del Datawarehouse.

11. TABLA DIMENSIONES ALMACENES



Figura 43: Transformación de las tablas de dimensiones Almacenes

En esta tabla no es necesario realizar ninguna operación sobre los datos, al cargarse toda la tabla en el Datawarehouse. Se definen los siguientes pasos:

- ALMACENES: Extrae la información de la tabla ALMACENES de InLOG WMS.
- Insertar / Actualizar Almacenes: Inserta los datos a la tabla de dimensión ALMACENES del Datawarehouse.

12. TABLA DIMENSIONES CLIENTES Y PROVEEDORES

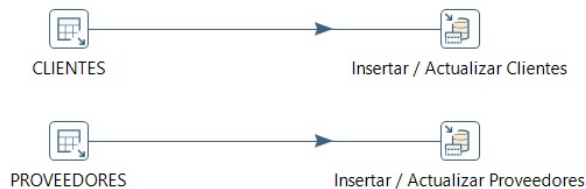


Figura 44: Transformación de las tablas de dimensiones Clientes y Proveedores

En estas tablas no es necesario realizar ninguna operación sobre los datos, al cargarse toda la tabla en el Datawarehouse. Se definen los siguientes pasos:

- **CLIENTES:** Extrae la información de la tabla CLIENTES de InLOG WMS.
- **Insertar / Actualizar Clientes:** Inserta los datos a la tabla de dimensión CLIENTES del Datawarehouse.
- **PROVEEDORES:** Extrae la información de la tabla PROVEEDORES de InLOG WMS.
- **Insertar / Actualizar Proveedores:** Inserta los datos a la tabla de dimensión PROVEEDORES del Datawarehouse.

13. TABLA DIMENSIONES OPERARIOS



Figura 45: Transformación de la tabla de dimensiones Operarios

Se definen los siguientes pasos:

- **OPERARIOS WMS:** Extrae la información de la tabla OPERARIOS de InLOG WMS.
- **OPERARIOS RRHH:** Extrae la información de la tabla OPERARIOS del sistema de recursos humanos.
- **Unir 5:** Une la información de la tabla OPERARIOS extraída de InLOG WMS y OPERARIOS extraída del sistema de recursos humanos.
- **Selecciono valores 11:** Selecciona aquellos campos a insertar, eliminando aquellos campos del sistema de recursos humanos que no son necesarios.
- **Insertar / Actualizar Operarios:** Inserta los datos a la tabla de dimensión OPERARIOS del Datawarehouse.

14. TABLAS DIMENSIONES PROPIETARIOS



Figura 46: Transformación de las tablas de dimensiones propietarios

En esta tabla no es necesario realizar ninguna operación sobre los datos, al cargarse toda la tabla en el Datawarehouse. Se definen los siguientes pasos:

- PROPIETARIOS: Extrae la información de la tabla PROPIETARIOS de InLOG WMS.
- Insertar / Actualizar Propietarios: Inserta los datos a la tabla de dimensión PROPIETARIOS del Datawarehouse.

15. TABLAS DIMENSIONES TIPOS MOVIMIENTO Y PEDIDO

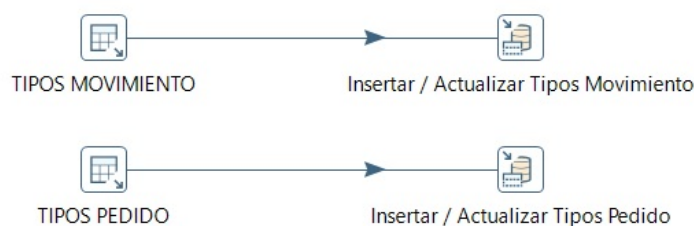


Figura 47: Transformación de las tablas de dimensiones tipos movimiento y pedido

En estas tablas no es necesario realizar ninguna operación sobre los datos, al cargarse toda la tabla en el Datawarehouse. Se definen los siguientes pasos:

- TIPOS MOVIMIENTO: Extrae la información de la tabla TIPOS MOVIMIENTO de InLOG WMS.
- Insertar / Actualizar Tipos Movimiento: Inserta los datos a la tabla de dimensión TIPOS MOVIMIENTO del Datawarehouse.
- TIPOS PEDIDO: Extrae la información de la tabla TIPOS PEDIDO de InLOG WMS.
- Insertar / Actualizar Tipos Pedido: Inserta los datos a la tabla de dimensión TIPOS PEDIDO del Datawarehouse.

16. TABLAS DIMENSIONES TIPOS PEDIDO PROVEEDOR Y PALETA

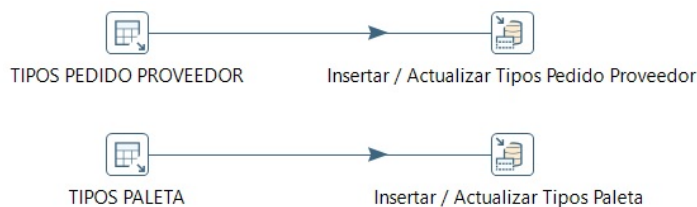


Figura 48: Transformación de las tablas de dimensiones tipos pedido proveedor y paleta

En estas tablas no es necesario realizar ninguna operación sobre los datos, al cargarse toda la tabla en el Datawarehouse. Se definen los siguientes pasos:

- TIPOS PEDIDO PROVEEDOR: Extrae la información de la tabla TIPOS PEDIDO PROVEEDOR de InLOG WMS.
- Insertar / Actualizar Tipos Pedido Proveedor: Inserta los datos a la tabla de dimensión TIPOS PEDIDO PROVEEDOR del Datawarehouse.
- TIPOS PALETA: Extrae la información de la tabla TIPOS PALETA de InLOG WMS.
- Insertar / Actualizar Tipos Paleta: Inserta los datos a la tabla de dimensión TIPOS PALETA del Datawarehouse.

Una vez creadas las transformaciones, definimos los jobs para ejecutar el proceso ETL.

En este caso, el job será muy sencillo, primero ejecutará transformación en la que se cargaran los datos de todas las tablas de dimensiones de las cuales se calcula un nuevo campo numérico de identificación, y después se ejecutará la transformación en la que se cargarán los datos del resto de tablas de hechos y y dimensiones al datawarehouse. Se enviará un correo electrónico informando del resultado del proceso.

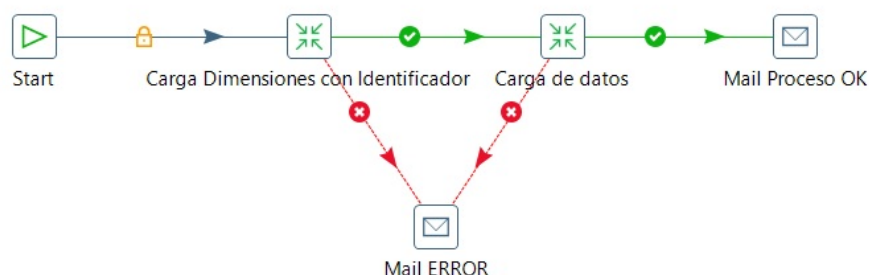


Figura 49: Job para la carga de datos

Para lanzar los jobs, se ofrece junto a la aplicación Pentaho Data Integration utilizada para crear tanto las transformaciones como los jobs un módulo llamado Kitchen.

Kitchen permite ejecutar jobs diseñados por Pentaho en formato .kjb, o almacenados en un repositorio desde la consola de comandos. Para ello, es necesario que todos los scripts de la aplicación Pentaho Data Integration tengan permisos de ejecución. Para otorgar los permisos, ejecutamos el siguiente comando estando situados en la carpeta de la aplicación: [30]

```
chmod +x *.sh
```

Ejecutando el script kitchen.sh desde la consola de comandos podemos lanzar los jobs:

```
sh kitchen.sh [options]
```

Las opciones interesantes para este comando son las siguientes:

- /file:nombreFichero.kjb
Lanza el job definido en nombreFichero.kjb
- /logfile:archivoLog
Especifica un archivo donde guardar los logs de la ejecución. Si no se especifica, se muestran en la consola de comandos.
- /level:LogginLevel
Especifica el nivel de log.

En este caso, hemos situado tanto el fichero del job como de las transformaciones en la carpeta de Pentaho Data Integration, y ejecutamos la carga de datos con el siguiente comando:

```
sh kitchen.sh /file:"InlogJob.kjb"
```

```
[etl@inlogbd pdi]$ sh kitchen.sh /file:"InlogJob.kjb"
19:05:15,902 INFO [KarafBoot] Checking to see if org.pentaho.clean.karaf.cache is enabled
19:05:15,960 INFO [KarafInstance]
*****
*** Karaf Instance Number: 1 at /home/etl/pdi/./system/karaf/caches/kitchen ***
*** /data-1 ***
*** Karaf Port:8802 ***
*** OSGI Service Port:9051 ***
*****
7:05:17 PM org.apache.karaf.main.Main$KarafLockCallback lockAcquired
INFORMACIÃO: Lock acquired. Setting startlevel to 100
19:05:17 - Kitchen - Start of run.
```

Figura 50: Ejecución del job principal mediante Kitchen

Para definir una carga de datos al Datawarehouse de manera periodica, haremos uso del demonio de programación de tareas Cron y el comando ejecutado anteriormente.

Cron es un demonio que se encuentra de forma nativa en gran parte de las distribuciones de Linux, el cual permite ejecutar comandos o scripts en función la programación personalizada que le dé el usuario, con lo cual lo hace un demonio muy útil para automatizar tareas. [31]

Cron se configura mediante el fichero de texto crontab, donde se especifica que procesos que deben ejecutarse y la hora a la que deben hacerlo. Para modificar el archivo crontab del usuario actual, ejecutamos el siguiente comando en la consola de comandos: [32]

```
crontab -e
```

La sintaxis para determinar la tarea programada es la siguiente:

```
"minuto" "hora" "días" "día" "comando"
```

Disponemos además del carácter especial *, el cual representa el valor todo. [32]

Para facilitar el proceso, crearemos el script run_etl, el cual se encargará de lanzar el job utilizando el comando Kitchen, y guardar los resultados en un log cuyo nombre será la fecha actual y estará guardado en la carpeta logs situada en la carpeta personal del usuario etl.

```
[etl@inlogbd ~]$ vi run_etl.sh
#!/usr/bin/env bash
FECHA=`date "+%d%b%Y"`
sh /home/etl/pdi/kitchen.sh /file:"/home/etl/pdi/InlogJob.kjb" /logfile:"/home/etl/logs/$FECHA"
~
```

Figura 51: Script ejecución proceso ETL

En este caso, se ha decidió hacer la carga de datos todos los domingos a las 00:00, al ser un día festivo en el que no suele haber movimientos en la base de datos operativa. Con lo cual, el valor introducido en el archivo crontab será el siguiente:

```
00 00 * * SUN sh /home/etl/run_etl.sh
```

```
[etl@inlogbd ~]$ crontab -e
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

#run-parts
00 00 * * SUN sh /home/etl/run_etl.sh
~
```

Figura 52: Fichero Crontab

Con esto, se realizará la carga de datos al Datawarehouse todos los domingos, logrando así una carga periódica de los datos.

7.3. Implementación de los cubos OLAP

El primer paso para implementar los cubos es desarrollar la vista para la dimensión tiempo.

Para obtener todas las fechas desde el 1 de enero de 2020 hasta un mes después de la fecha del sistema en el momento de hacer la consulta haremos uso de la siguiente función:

```
SELECT TO_DATE( '01/01/2020 ')+rownum-1 FECHA
FROM dual
CONNECT BY LEVEL <= (SELECT TRUNC(SYSDATE+30-TO_DATE( '01/01/2020 '))
FROM DUAL)
```

El siguiente paso será definir una consulta que calcule los distintos atributos de las fechas obtenidas con la función anterior utilizando la función EXTRACT definida en PLSQL, la cual permite extraer los valores de diferentes elementos de un dato de tipo DATE (YEAR, MONTH, DAY...), y el procedimiento almacenado CALCULA_FESTIVO, el cual determina si una fecha es festivo o no haciendo uso de la tabla auxiliar FESTIVOS, tabla en la que el cliente puede insertar los días que considera como festivos, y teniendo en cuenta sábados y domingos como festivos.

Una vez creada la consulta, creamos una vista sobre esta utilizando la función CREATE OR REPLACE VIEW.

	FECHA	AÑO	MES	DIA	SEMANA	SEMESTRE	TRIMESTRE	DIAJULIANO	FESTIVO
1	01/01/2020	2020	Enero	Miercoles	1	1	1	2458850	FESTIVO
2	02/01/2020	2020	Enero	Jueves	1	1	1	2458851	NO FESTIVO
3	03/01/2020	2020	Enero	Viernes	1	1	1	2458852	NO FESTIVO
4	04/01/2020	2020	Enero	Sabado	1	1	1	2458853	FESTIVO
5	05/01/2020	2020	Enero	Domingo	1	1	1	2458854	FESTIVO
6	06/01/2020	2020	Enero	Lunes	2	1	1	2458855	FESTIVO
7	07/01/2020	2020	Enero	Martes	2	1	1	2458856	NO FESTIVO
8	08/01/2020	2020	Enero	Miercoles	2	1	1	2458857	NO FESTIVO
9	09/01/2020	2020	Enero	Jueves	2	1	1	2458858	NO FESTIVO
10	10/01/2020	2020	Enero	Viernes	2	1	1	2458859	NO FESTIVO
11	11/01/2020	2020	Enero	Sabado	2	1	1	2458860	FESTIVO
12	12/01/2020	2020	Enero	Domingo	2	1	1	2458861	FESTIVO
13	13/01/2020	2020	Enero	Lunes	3	1	1	2458862	NO FESTIVO
14	14/01/2020	2020	Enero	Martes	3	1	1	2458863	NO FESTIVO
15	15/01/2020	2020	Enero	Miercoles	3	1	1	2458864	NO FESTIVO
16	16/01/2020	2020	Enero	Jueves	3	1	1	2458865	NO FESTIVO

Figura 53: Consulta Vista Tiempo

La implementación de los cubos se realizará mediante la herramienta Pentaho Schema Workbench (PSW) incluida en la suite de Pentaho.

Esta herramienta permite implementar y probar los cubos OLAP de una manera gráfica, generando un fichero .xml donde se define la estructura de los cubos y las conexiones a la base de datos, ofreciendo además la posibilidad de publicarlos en la aplicación web directamente desde la aplicación

En Schema Workbench, todos los cubos y dimensiones compartidas forman parte de un Schema, con lo cual, el primer paso es generar un schema nuevo. Las operaciones que ofrece sobre un schema son las siguientes:

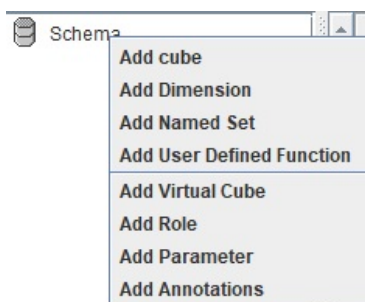


Figura 54: Operaciones sobre un schema

El primer paso es generar las dimensiones compartidas. Para ello, tenemos la opción “Add Dimension”. Las dimensiones pueden ser de dos tipos, Standard-Dimension, TimeDimension, siendo este último tipo utilizado para la dimensión tiempo.

Una vez creada la dimensión, es necesario crear las jerarquías de la dimensión utilizando la opción Add “Hierarchi”, ofrecida para la dimensión, y dentro de esta definir la tabla que forma la dimensión “Add Table”, y los distintos niveles de la dimensión utilizando la opción “Add Level”

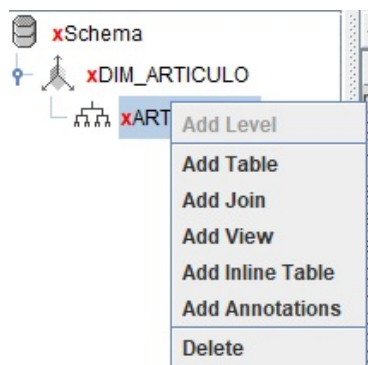


Figura 55: Operaciones sobre jerarquía de una dimensión

Una vez creadas las dimensiones compartidas, creamos los cubos utilizando la opción “Add cube”. Las opciones que sobre un cubo son las siguientes:

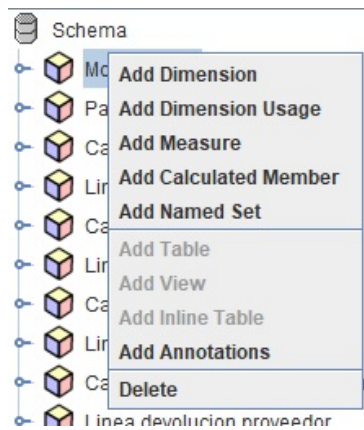


Figura 56: Operaciones sobre un cubo

Para cada cubo, definimos las dimensiones utilizando “Add Dimensión”, o en caso de ser una dimensión compartida, utilizando “Add Dimension Usage”. Se definen las tabla de hechos del cubo utilizando “Add Table”, y se añaden las medidas utilizando “Add Measure”.

Para cada dimensión del cubo es necesario indicar en sus atributos cual es el campo de la tabla de hechos que se relaciona con la dimensión.

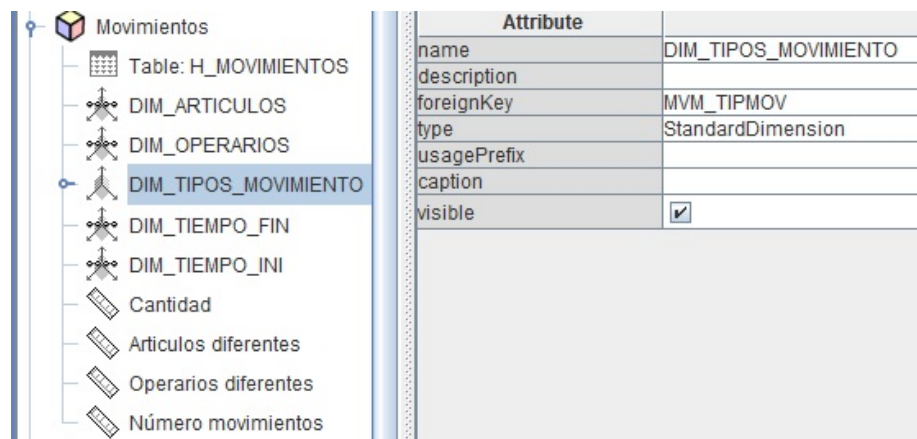


Figura 57: Atributos de una dimensión

El schema queda definido de la siguiente manera:

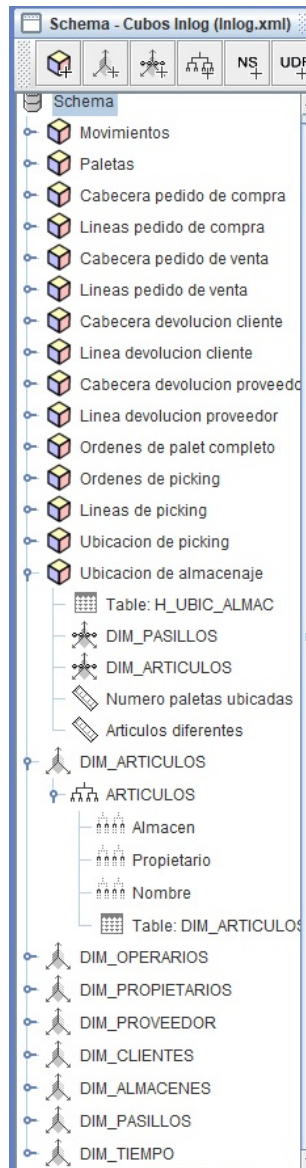


Figura 58: Schema con los cubos y dimensiones compartidas

7.4. Implementación de informes predefinidos

La implementación de los informes se realizará mediante la herramienta Pentaho Report Designer (PRD) incluida en la suite de Pentaho.

Esta herramienta permite diseñar los informes de una forma gráfica y ejecutarlos directamente desde la propia herramienta en múltiples formatos (PDF, HTML, XLS y CSV) o publicarlos en la aplicación web para poder lanzarlos desde esta. [33]

El primer paso es definir la fuente de datos de la cual se obtendrán los datos a mostrar en el informe. Para ello, se utiliza la opción “Data / Add Datasource / JDBC”.

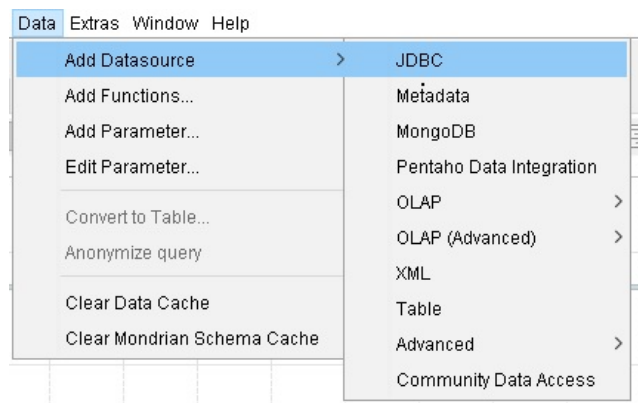


Figura 59: Añadir fuente de datos en Report Designer

Se mostrará una ventana en la que se permitirá establecer la conexión con la base de datos y definir la sentencia SQL con la que se obtendrán los diferentes datos. En caso de tener una condición definida por un parámetro, esta se incluirá en la cláusula WHERE de la forma $\${PARAMETRO}$

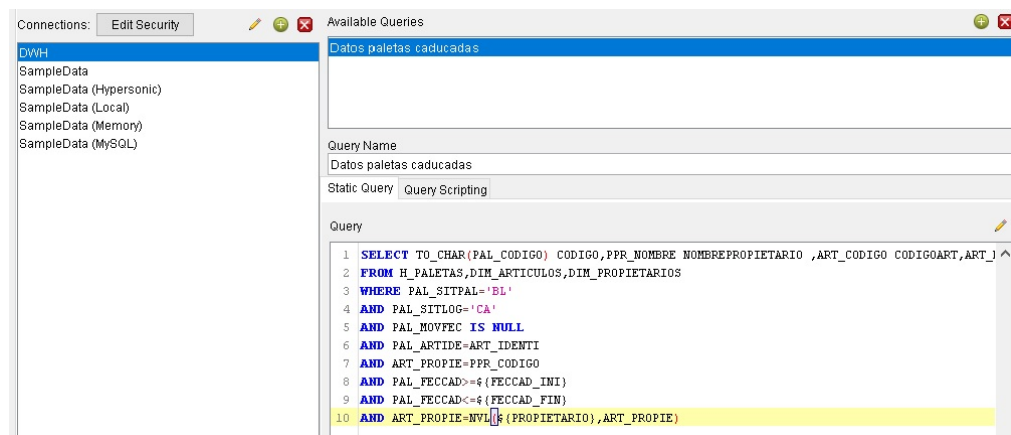


Figura 60: Creación de sentencia del informe

Una vez definida la fuente de datos, comenzamos a diseñar el informe. Para ello, disponemos de un workspace de diseño, en el cuál, podremos añadir aquellos elementos necesarios para diseñar el informe.

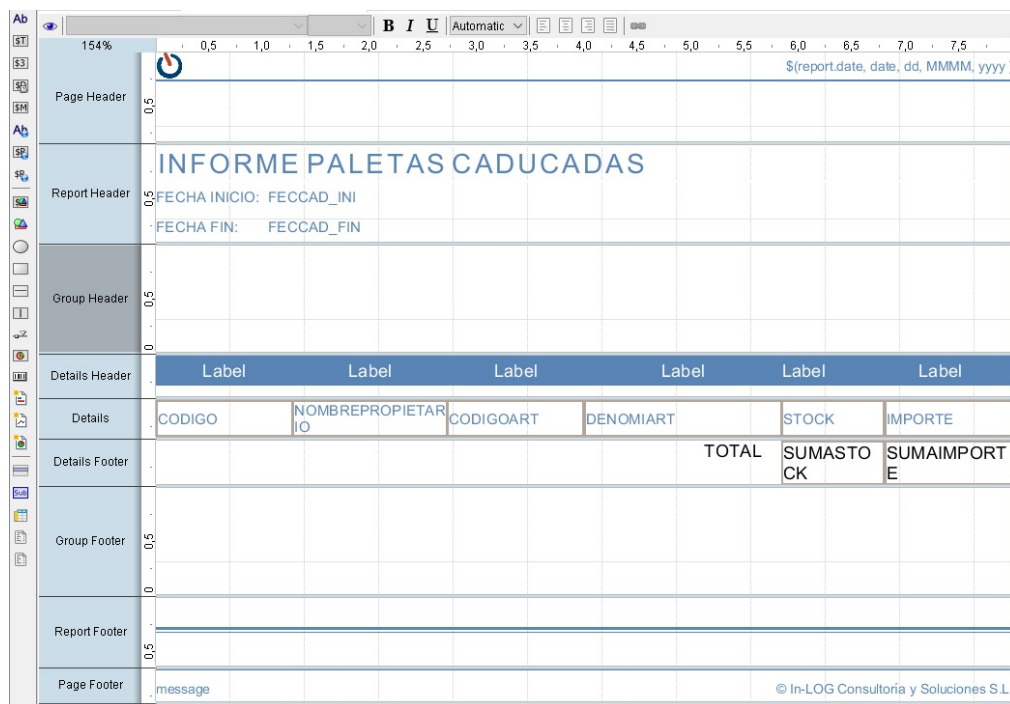


Figura 61: Workspace diseño de informe

Este workspace está dividido en varias categorías. [34]

- Page Header: Representa la cabecera de cada una de las páginas del informe. [34]
- Report Header: Representa la cabecera del informe, mostrándose únicamente en la primera página del informe. [34]
- Group Header: Representa la cabecera de los grupos de información definidos en el informe, en caso de trabajar con estos. [34]
- Details: Formada por Details Header, Details y Details Footer, representa los datos del informe. Contiene la información fila por fila recuperada de la base de datos. [34]
- Group Footer: Representa el pie de los grupos de información definidos en el informe, en caso de trabajar con estos. [34]
- Report Footer: Representa el pie del informe, mostrándose únicamente en la última página del informe. [34]
- Page Footer: Representa el pie de cada página del informe. [34]

Disponemos además de un panel de datos, en el cual podremos definir campos calculados a partir de una fórmula, y los parámetros del informe:

- Para definir un parámetro, hacemos clic sobre “Parameters / Add Parameter”. Se abrirá una ventana para definir la estructura del parámetro.
- Para definir un campo calculado, hacemos clic sobre “Functions / Add Function”, y se abrirá una ventana para definir el tipo y la fórmula a aplicar.

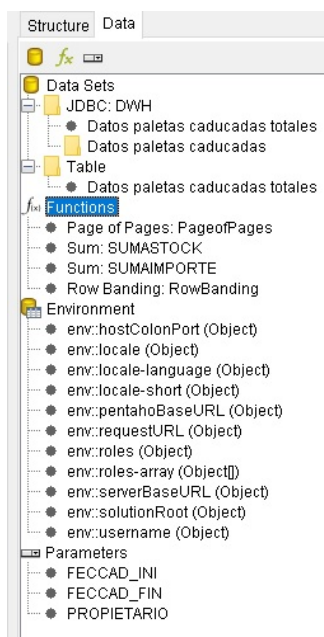


Figura 62: Panel de datos en Report Designer

7.5. Despliegue de la aplicación web

El corazón de Pentaho es la aplicación web, la cual será la forma en la que los clientes podrán utilizar los distintos recursos desarrollados de una forma sencilla. Está desarrollada en Java EE con el framework Spring. Al ser open source, podemos modificar y personalizar todas las características de esta. [26]

Para realizar las modificaciones sobre la aplicación, importaremos el proyecto de la aplicación web a un IDE para poder trabajar de forma local sobre él. En este caso, utilizaremos Eclipse IDE for Enterprise Java and Web Developers.

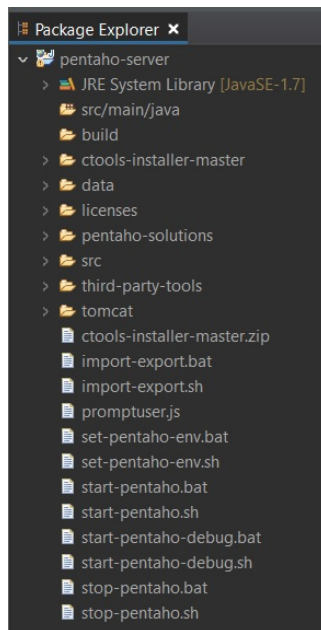


Figura 63: Organización del proyecto en Eclipse

La mayoría de los componentes de Pentaho, como el visor de informes predefinidos, motor OLAP, herramienta de edición y visor de cuadros de mandos se encuentran en pentaho-solutions, mientras que los elementos web se encuentran en tomcat/webapps.

La pantalla de login está definida en tomcat/webapps/pentaho/jsp/PUCLogin.jsp, y la pantalla de la consola principal está definida en tomcat/webapps/pentaho/mantle/home/content/welcome/index.html. Modificaremos ambas pantallas para obtener la apariencia deseada.

Además, por defecto Pentaho solo está disponible en inglés, por lo que modificaremos todos los textos de los distintos elementos web a utilizar para traducir la aplicación a castellano. En la mayoría de los componentes de Pentaho tenemos una carpeta messages en la que se encuentran los textos y en el caso de los elementos web modificaremos los jsp y html para traducir la aplicación.

7.5.1. Conexión de la aplicación con el Datawarehouse

Para poder hacer uso de los cubos e informes predefinidos es necesario conectar la aplicación web con el Datawarehouse.

Para ello, crearemos una nueva conexión desde la consola principal de la aplicación haciendo clic sobre el la opción “Gestionar Fuente de Datos”. Se mostrará la pantalla para la administración de fuentes de datos.



Figura 64: Pantalla administración de fuentes de datos

En esta pantalla se podrán crear y gestionar las diferentes fuentes de datos con las que trabajará la aplicación, pudiendo crear nuevas conexiones a base de datos, importar los schemas de los cubos o crear/importar metadatos para los informes ad-hoc.

Para crear la conexión con el Datawarehouse, hacemos clic sobre Nueva conexión y se abrirá un asistente para realizar la configuración. Definiremos una nueva conexión llamada DWH.

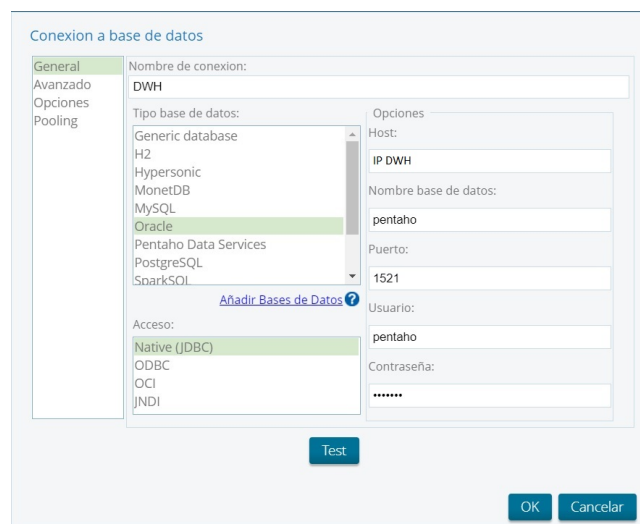


Figura 65: Asistente creación de conexión a base de datos

7.5.2. Despliegue de los cubos OLAP

Para desplegar el cubo OLAP en el servidor web, podemos realizarlo directamente desde la aplicación Schema Workbench en la que se definieron los cubos. Para ello, teniendo el schema de los cubos abiertos, hacemos clic en File / Publish

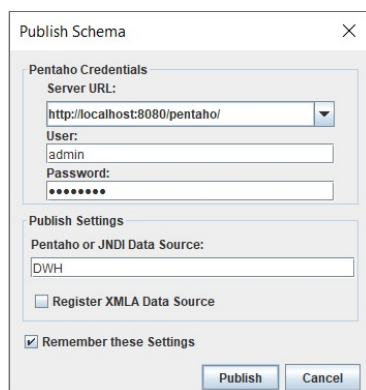


Figura 66: Asistente publicación cubos OLAP

Introducimos la dirección del servidor Pentaho, usuario y contraseña del usuario administrador, el nombre de la conexión a la base de datos definida (en este caso, hemos definido la conexión DWH en el punto anterior) y pulsamos en el botón “Publish” para publicar el schema de los cubos en el servidor.

Una vez publicados, podremos ver una nueva fuente de datos del tipo Analysis definida en la pantalla de gestión de fuentes de datos correspondiente al schema de los cubos.

Fuente de datos	Tipo
Cubos Inlog	Analysis
DWH	JDBC

Figura 67: Fuente de datos cubos OLAP

Para poder realizar consultas contra los cubos de una forma visual, se incluye con Pentaho el visor JPivot. JPivot es una librería de componentes JSP que se utiliza para generar tablas OLAP de una forma gráfica. [35]

Al quedar JPivot discontinuado, surgió una nueva librería llamada Pivot4j, basado en JPivot, la cual es una herramienta bastante más amigable a la hora de generar las tablas OLAP. [36]

Pivot4j está disponible tanto como una aplicación web independiente, como un plugin para instalar en Pentaho, el cual se integra con la aplicación web. Este plugin se puede descargar desde el Github del desarrollador, o empleando el Marketplace incluido en el servidor de Pentaho.

Para instalar el plugin, clonamos el proyecto git en la carpeta pentaho-solutions/system y reiniciamos el servidor. Para ello, haremos un git-clone del proyecto empleando la herramienta TortoiseGit.

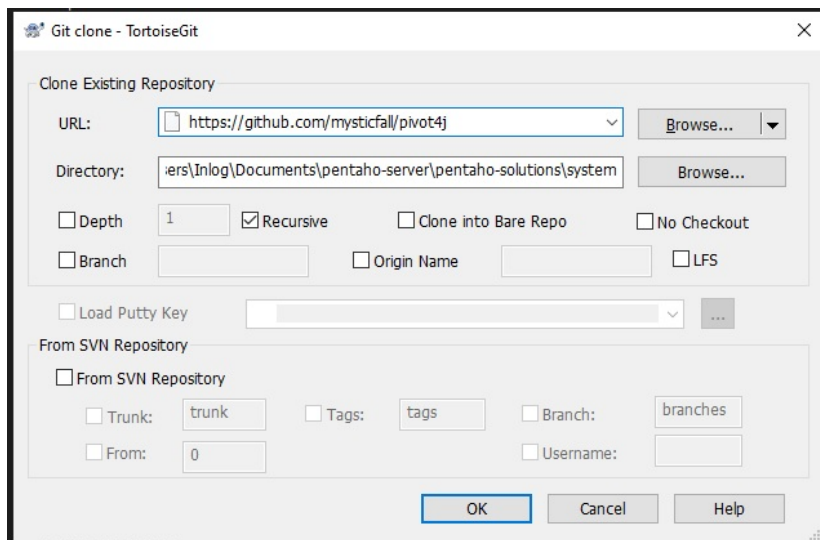


Figura 68: Clonación del proyecto Pivot4J

Una vez reiniciado el servidor, aparecerá una opción correspondiente a Pivot4J View en el menú Crear Nuevo. Modificamos el fichero de propiedades plugin.xml ubicando en la carpeta del plugin para cambiar el nombre del menú por Visor OLAP.

La generación del menú Crear Nuevo se realiza dinámicamente en el fichero pentaho-server/tomcat/webapps/pentaho/mantle/home/js/createNew.js. Modificaremos este fichero para dejar de mostrar la opción correspondiente a el antiguo visor OLAP JPivot.



Figura 69: Vista Menú Crear nuevo

Con este visor, podemos crear las consultas MDX contra los cubos definidos anteriormente de una forma gráfica.

The screenshot shows the OLAP Navigator interface. On the left, the 'Estructura de cubo' (Cube Structure) pane shows a hierarchy: Proveedor, TIPOS_PEDPROV, Area descarga, and Tiempo. The 'Estructura pivote' (Pivot Structure) pane shows 'Columnas' with 'Measures' (Numero pedidos) and 'Filas' with 'Tiempo' (Año, Semestre, Trimestre, Mes). The 'Resultado de la consulta' (Query Result) pane displays a table with columns 'Tiempo' and 'Measures'. The table contains data for years 2020 and 2021, and months Enero, Febrero, and Marzo. The 'Consulta MDX' pane at the bottom shows the MDX query: `SELECT {[Measures].[Numero pedidos]}`.

Tiempo	Measures
+ 2020	
- 2021	131
- 1	131
- 1	131
+ Enero	18
+ Febrero	90
+ Marzo	23
+ 2	

Figura 70: Consulta contra cubo Cabecera Pedido Compra

7.5.3. Despliegue de la herramienta para consultas ad-hoc

La versión Community de Pentaho no ofrece ninguna característica para explorar los metadatos del Datawarehouse realizando consultas ad-hoc, es por ello que es necesario realizar un plugin que permita hacer estas consultas contra el Datawarehouse de una forma sencilla.

Para realizar este plugin, nos hemos basado en el plugin WAQE (Web Adhoc Query Executor) desarrollado por Jortilles, el cual estuvo disponible hasta la versión 7 de Pentaho. Este plugin tiene una licencia open source que permite tratar el software sin restricciones y todo el código está disponible para ser modificado en el Github del creador.

Este plugin está desarrollado mediante la herramienta Sparkl, incluida en la aplicación web de Pentaho, la cual permite desarrollar plugins de una manera similar a como se desarrollaría un cuadro de mando, siendo los plugins realizados con Sparkl un conjunto de cuadros de mando.

Para instalar el plugin, haremos al igual que con el visor OLAP, un git-clone del proyecto Github en la carpeta pentaho-solutions/system y reiniciaremos el servidor.

WAQE está formado por un único cuadro de mando basado en el framework Bootstrap. La parte visual de este plugin está diseñada principalmente utilizando componentes HTML, Javascript y CSS, mientras que la lógica de generación de las consultas SQL y su guardado / exportación está implementada en Javascript y Java respectivamente.



Figura 71: Herramienta para el desarrollo de Plugins

Para el desarrollo de los cuadros de mando utilizaremos la herramienta Community Dashboard Editor incluida en la aplicación web de Pentaho. Esta herramienta se abrirá automáticamente al comenzar a modificar el cuadro de mando del plugin.

El desarrollo de un cuadro de mando se divide en tres componentes Layout, Components y Datasources. [37]

- Layout: Es la representación gráfica de los datos. Se define el diseño y la estructura del cuadro de mando, haciendo uso de estilos css, código javascript y html
- Components: Se definen los componentes con los que se mostrarán los datos obtenidos por base de datos (checkbox, cuadros de selección, gráficas...)
- Datasources: Se definen las consultas a realizar contra la base de datos

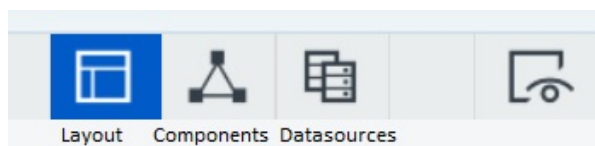


Figura 72: Componentes de un cuadro de mando

El primer paso, será modificar el layout para obtener la vista deseada. En cuanto a este, tenemos los siguientes componentes:

- Fila cabecera: Se encuentran los botones Exportar a Excel y Guardar Consulta
- Fila principal: Compuesta por dos columnas: columna de selección de datos, en la que se permite al usuario seleccionar los datos a consultar y resultados de consulta, en la que se muestra la consulta realizada.

Realizaremos los siguientes cambios:

- Crearemos un array llamado `wdu.buttonFilter` donde guardaremos los filtros introducidos manualmente
- Modificaremos la función `wdu.filter`, la cual se ejecuta al abrir el popup de los filtros. Vaciamos el cuadro de texto y añadiremos un evento `onclick` sobre el botón Añadir para añadir el filtro introducido.

```
327 wdu.buttonFilter=[];
328 wdu.filter = function(col){
329     //console.log(col);
330     wdu.filtering = col;
331     $('#div_filter_lbl').text( 'Filtro en columna: ' + col.split('|-|')[4]);
332     document.getElementById('filtroTxt').value = ''; //XBG: Pongo el textfield vacio
333     //XBG: Evento onclick en boton añadir
334     document.getElementById('filtroButton').onclick = function(){
335         var filterText= document.getElementById('filtroTxt').value;
336         if (filterText!=''){ //Si hay algun filtro, se añade
337             wdu.buttonFilter.push(col+"-"+filterText);
338             wdu.multisel_filter_elements_PostChange();
339         }
340     }
341     render_multisel_filter_elements.update();
342 }
```

Figura 74: Implementación de la función `wdu.filter`

- Modificaremos la función `wdu.PostChange`, la cual se ejecuta tras añadir / borrar un filtro, y se encarga de añadir los filtros seleccionados a la consulta SQL a generar. Trataremos también los filtros introducidos manualmente en la lista `wdu.buttonFilter`.

```
368 wdu.multisel_filter_elements_PostChange = function(){
369     var filter = {};
370     filter = {
371         column: null,
372         elems:[]
373     };
374     filter.column = wdu.filtering;
375     for ( var i=0; i < param_filter_elements.length; i++){
376         filter.elems.push(param_filter_elements[i] );
377     }
378     //XBG: Compruebo filtros añadidos manualmente
379     for ( var i=0; i < wdu.buttonFilter.length; i++){
380         var elFiltro=wdu.buttonFilter[i].split(';');
381         if (elFiltro[0]==filter.column) {
382             filter.elems.push(elFiltro[1]);
383         }
384     }
385
386     if(param_filter_elements.length == 0){
387         w.removeColumnFiltering(wdu.filtering);
388     }
389     /*
390     console.log('wdu.multisel_filter_elements_PostChange');
391     console.log('lenght' + filter.elems.length );
392     console.log(filter);
393     */
394
395     if(filter.elems.length > 0){
396         w.addFilter( filter );
397     }
398     Dashboards.fireChange('param_run_query', param_run_query + 1 );
399 }
```

Figura 75: Implementación de la función `wdu.PostChange`

- Modificaremos la función `wdu.removeFilter`, la cual se ejecuta para borrar un filtro. Consultaremos si el filtro a borrar está en la lista de filtros añadidos manualmente, para eliminarlo en ese caso.

```

190 wdu.removeFilter = function( column, elem){
191     //console.log('wdu.removeFilter llega column: ' + column + ' elem: ' + elem );
192     for ( var i=0; i < w.query.filter.length; i++){
193         if( w.query.filter[i].column == column){
194             $.each( w.query.filter[i].elems, function(j){
195                 if(w.query.filter[i].elems[j] === elem ) {
196                     w.query.filter[i].elems.splice(j,1);
197                     return false;
198                 }
199             });
200         }
201     }
202     for ( var i=0; i < w.query.filter.length; i++){
203         if( w.query.filter[i].elems.length == 0){
204             w.query.filter.splice(i,1);
205         }
206     }
207     //XBG: Compruebo si el filtro a eliminar esta entre los añadidos manualmente
208     for (var i=0; i<wdu.buttonFilter.length; i++){
209         if (wdu.buttonFilter[i]==column+' '+ elem){
210             wdu.buttonFilter.splice(i,1);
211         }
212     }
213
214     Dashboards.fireChange('param_run_query', param_run_query + 1 );
215     render_multisel_filter_elements.update();
216
217 };

```

Figura 76: Implementación de la función `wdu.removeFilter`

El resultado final es el siguiente:



Figura 77: Popup selector de filtros

7.5.4. Creación de metadatos para consultas ad-hoc

Una vez implementada la herramienta que nos permitirá realizar las consultas ad-hoc, el siguiente paso es definir los metadatos sobre los que haremos las consultas. Para ello, utilizaremos la herramienta Pentaho Metadata Editor (PME) incluida en la suite de Pentaho.

Pentaho Metadata Editor es una herramienta de escritorio que permite desarrollar de forma gráfica los dominios y modelos de metadatos. Un dominio, el cual se genera como un fichero .xmi, representa todos los modelos de metadatos que tienen alguna entidad común. [38]

El primer paso es definir la conexión a la base de datos, para ello, haremos clic en File / New / Connection e introduciremos los datos de conexión a el Datawarehouse, de una manera similar a la definición de la conexión realizada en las herramientas Pentaho Report Designer, Schema Workbench y Pentaho Data Integration.

Una vez creada la conexión al Datawarehouse, el siguiente paso es importar las tablas necesarias para formar los distintos modelos de datos. Para ello, desde la conexión creada haremos clic en la opción “Import From Explorer”

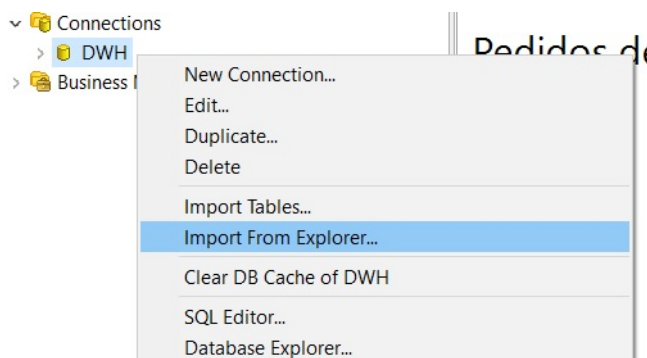


Figura 78: Opciones sobre conexión en la herramienta PME

Se mostrará el esquema de base de datos, permitiendo seleccionar aquellas tablas que formarán los diferentes modelo de datos a definir.

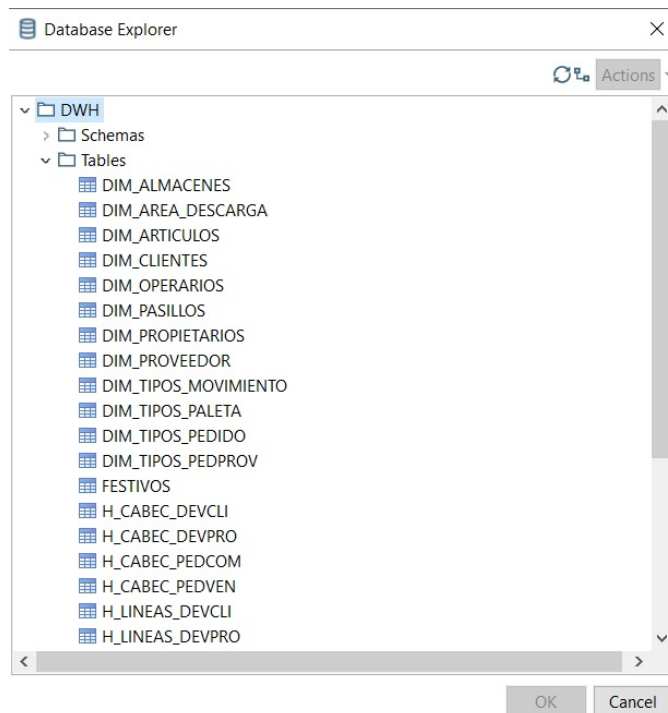


Figura 79: Búsqueda de tablas a importar en la herramienta PME

Una vez importadas las tablas, podremos modificar sus características para añadir un nombre personalizado a los campos que se mostrarán desde la aplicación de consultas, crear campos calculados ...

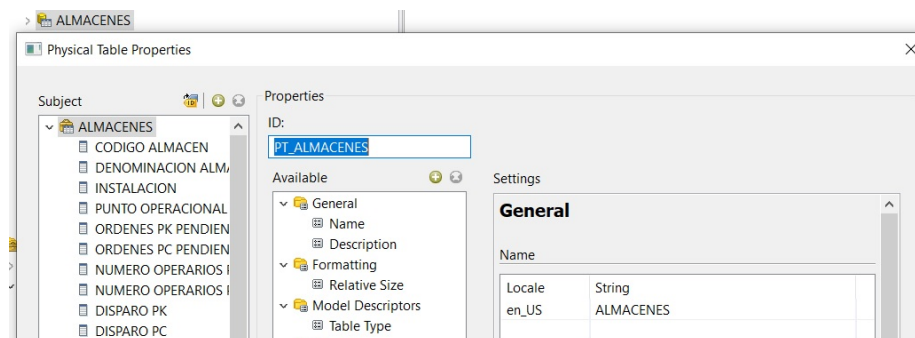


Figura 80: Modificación de tabla ARTICULOS en la herramienta PME

Una vez modificadas todas las tablas importadas, creamos un nuevo modelo de datos haciendo clic en Business Models / New Business Model. Crearemos los siguientes modelos de datos: Compras, Ventas, Maestros y Control.

Una vez creado el modelo de datos, arrastraremos todas las tablas necesarias para crear el modelo de datos a el área grafica. Definimos las relaciones entre las tablas haciendo clic en New Relationship...

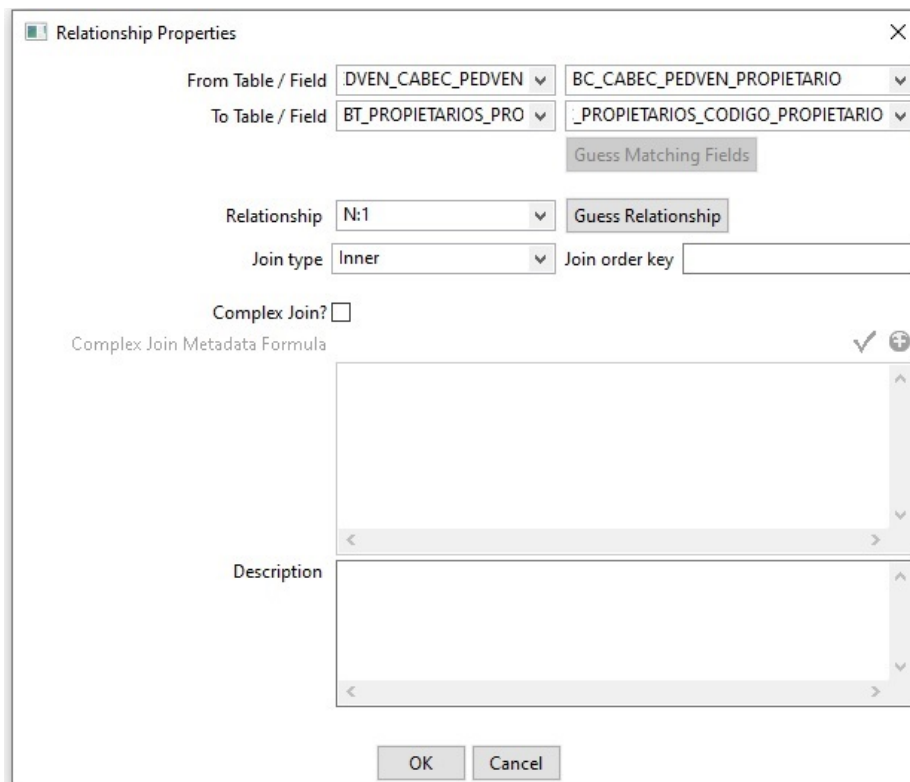


Figura 81: Asistente para la definición de relaciones entre tablas

Por cada relación será necesario definir:

- Tablas que forman la relación
- Campos relacionados (en caso de ser una unión compuesta por varios campos, se realiza declarando la fórmula en Complex Join)
- Cardinalidad de la relación: 1:N, N:1, 1:1 ...
- Tipo de relación: Inner (intersección entre ambas tablas) u outer (unión entre ambas tablas).

Una vez definidos los modelos de metadatos, la aplicación ofrece una opción para subir directamente el dominio a la aplicación web, haciendo clic en File / Publish To Server. Introduciendo los datos de acceso al servidor, subimos el fichero .xmi a la aplicación.

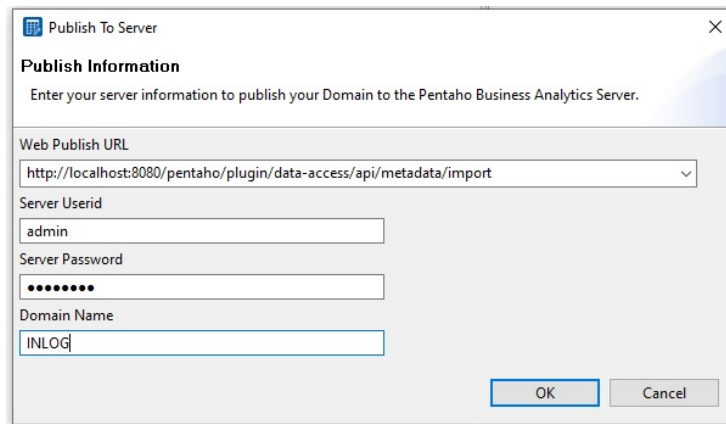


Figura 82: Asistente publicación dominio de metadatos

Una vez importado, podremos ver una nueva fuente de datos del tipo metadata definida en la pantalla de gestión de fuentes de datos correspondiente al dominio de metadatos.

Fuente de datos	Tipo
DWH	JDBC
INLOG	Metadata

Figura 83: Fuente de datos dominio de metadatos

Utilizando la herramienta de consultas ad-hoc desplegada anteriormente y los metadatos desarrollados, podemos realizar consultas contra el Datawarehouse de una forma sencilla.

Seleccionar modelo de datos:

Seleccionar categoría:

Seleccionar columna:

CODIGO PROVEEDOR	RAZON SOCIAL	NOMBRE
NOMBRE CORTO	TIPO DE ENTREGA	
DIRECCION PROVEEDOR	PROVINCIA PROVEEDOR	
LOCALIDAD PROVEEDOR		
CODIGO POSTAL PROVEEDOR	PAIS PROVEEDOR	
ALMACEN PROVEEDOR	AREA DESCARGA	
PEDIDOS ENTREGA	AGENCIA	
PERSONA DE CONTACTO	TELEFONO	
NUMERO DE FAX	PUNTO OPERACIONAL	
CODIGO EXTERNO	GESTION CONTENEDORES	

Show 50 entries

PEDIDO	DENOMINACION ALMACEN	CODIGO LINEA	ARTICULO	NOMBRE CORTO	CANTIDAD TEORICA	FORMATO VENTA
93	ALMACEN DE PRUEBAS	1	0000001	SEMIEJE MAZDA T	10	U
99	ALMACEN DE PRUEBAS	1	0000001	SEMIEJE MAZDA T	4	U

Showing 1 to 2 of 2 entries Previous Next

Filtros: SITUACION PEDIDO: PEARTICULO: 0000001 0000002 0000003 0000004 0000005

Figura 84: Ejemplo de consulta sobre pedidos de compra

7.5.5. Despliegue de informe predefinidos

Para desplegar los informes predefinidos en el servidor web, podemos realizarlo directamente desde la aplicación Pentaho Report Designer, haciendo clic en File / Publish por cada uno de los informes a desplegar.

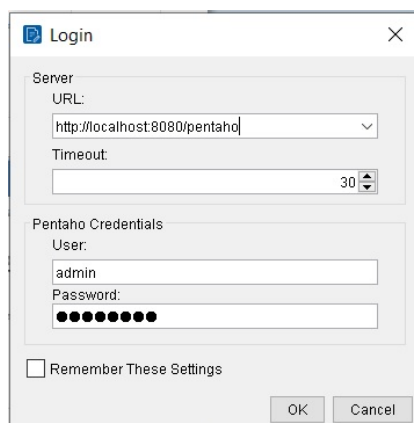


Figura 85: Asistente publicación de informe

Será necesario introducir la dirección del servidor pentaho, usuario y contraseña del usuario administrador para subir el informe a la aplicación web. Tras introducir los datos, se abrirá un asistente para seleccionar la ruta del servidor donde se guardará el informe.

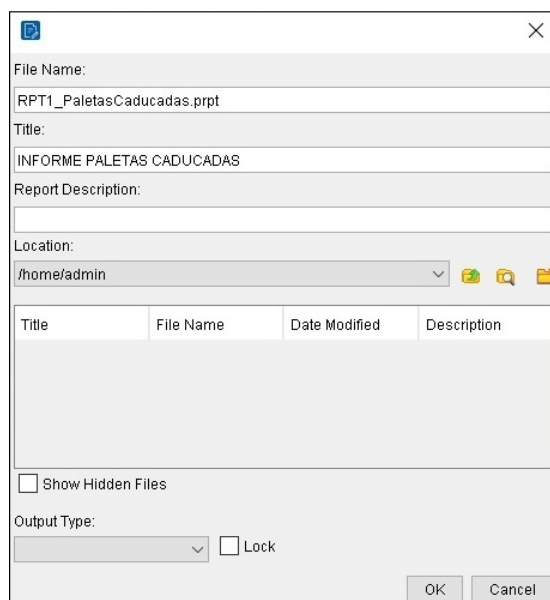


Figura 86: Selección ruta para publicación de informe

El informe será accesible desde la web haciendo clic en Ficheros y buscando el informe en la carpeta donde se ha guardado anteriormente.

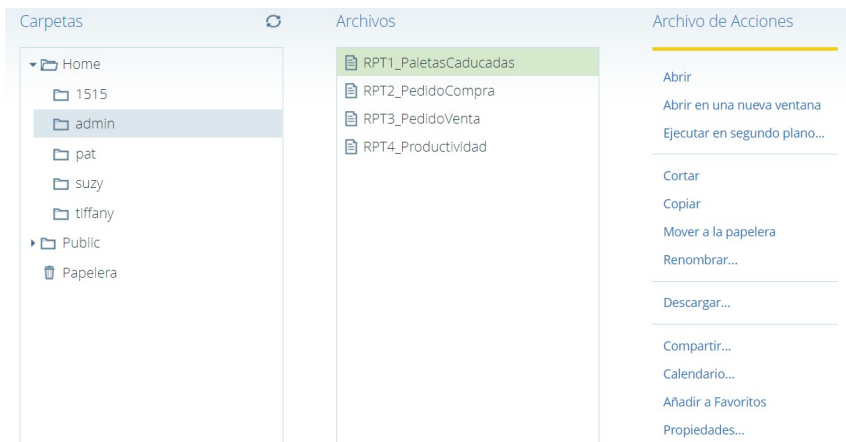


Figura 87: Búsqueda informes en la aplicación web

Seleccionando el informe deseado se abre mediante el componente Pentaho Reporting incluido en la aplicación web, en el cual se podrá seleccionar los parámetros para generar el informe y lanzarlo en múltiples fuentes de datos, como HTML (visualizado en la propia herramienta), PDF y CSV.

The screenshot shows the Pentaho Reporting tool interface for generating a report titled 'INFORME PALETAS CADUCADAS'. The interface includes a header with the report name and a date '30,abril,2021'. Below the header, there are several input fields for report parameters: 'Fecha Inicio' (05/04/2021), 'Fecha Fin' (30/04/2021), 'Propietario', and 'Tipo de Salida' (HTML (Paginado)). There is also a 'Ver Informe' button and a checked 'Auto-Enviar' checkbox. The main content area displays a table with the following data:

PALETA	PROPIETARIO	ARTICULO	NOMBRE ARTICULO	STOCK	IMPORTE
453251	MASSA REIG	0110	SALTXIXO CULAR -VILLAR-	22	110,00 €
453253	MASSA REIG	0110	SALTXIXO CULAR -VILLAR-	77	385,00 €
453256	MASSA REIG	0110	SALTXIXO CULAR -VILLAR-	77	385,00 €
453258	MASSA REIG	0110	SALTXIXO CULAR -VILLAR-	77	385,00 €

Figura 88: Ejemplo informe paletas caducadas

7.5.6. Instalación de la aplicación BI

Una vez desarrolladas todas las características de la aplicación web, subiremos esta al servidor.

Para ello, copiaremos mediante el comando SCP el contenido de la carpeta pentaho-server en nuestro servidor. Para poder utilizar el comando SCP en Windows, utilizaremos el bash de Linux incluido en las últimas versiones de Windows 10. Situándonos en la carpeta pentaho-server, ejecutamos el siguiente comando:

```
scp -r $PWD usuario@servidor:/rutaDestino
```

En este caso, instalaremos la aplicación web en la carpeta home del usuario SERVER.

```
Inlog@DESKTOP-275DQ9P MINGW64 ~/Documents/pentaho-server
$ scp -r $PWD server@192.168.56.102:/home/server
server@192.168.56.102's password:
.jsdtscope                                100% 572    242.6KB/s  00:00
org.eclipse.jdt.core.prefs                 100% 364    186.4KB/s  00:00
org.eclipse.wst.common.component           100% 509    318.6KB/s  00:00
org.eclipse.wst.common.project.facet.core.xml 100% 252    136.6KB/s  00:00
org.eclipse.wst.jsdt.ui.superType.container 100% 44     24.7KB/s   00:00
org.eclipse.wst.jsdt.ui.superType.name     100% 6      3.0KB/s    00:00
dist.zip                                   100% 5169KB 58.3MB/s   00:00
dist.zip                                   100% 13MB   60.2MB/s   00:00
dist.zip                                   100% 17MB   53.7MB/s   00:00
dist.zip                                   100% 10MB   58.2MB/s   00:00
ctools-installer.sh                       100% 27KB   13.4MB/s   00:00
cda-16.08.18-sources.jar                   100% 28KB   13.4MB/s   00:00
cda-16.08.18-sources.tar.gz               100% 17KB   9.0MB/s    00:00
cda-16.08.18-sources.zip                  100% 28KB   13.4MB/s   00:00
cda-16.08.18.jar                           100% 95KB   19.7MB/s   00:00
cda-16.08.18.zip                          100% 2526KB 39.7MB/s   00:00
cda-disabled-16.08.18.zip                 100% 2526KB 59.5MB/s   00:00
cda-samples-16.08.18.zip                  100% 42KB   16.1MB/s   00:00
```

Figura 89: Subida de la aplicación web al servidor

Accediendo desde ssh a la maquina con el usuario server, podemos comprobar que se ha subido la aplicación web correctamente:

```
[server@inlogbd ~]$ cd pentaho-server/
[server@inlogbd pentaho-server]$ ls
build                pentaho-solutions    start-pentaho-debug.sh
ctools-installer-master  promptuser.js        start-pentaho.sh
ctools-installer-master.zip  set-pentaho-env.bat  stop-pentaho.bat
data                 set-pentaho-env.sh   stop-pentaho.sh
import-export.bat     src                   third-party-tools
import-export.sh      start-pentaho.bat    tomcat
licenses              start-pentaho-debug.bat
```

Figura 90: Comprobación subida de la aplicación web al servidor

Antes de lanzar el servidor, es necesario modificar el puerto en el que se lanzará el servidor, ya que por defecto Pentaho se lanza en el servidor 8080, y en el caso de esta máquina, este puerto lo tiene ocupado por otra aplicación por lo que será necesario lanzarlo en otro puerto, en este caso el puerto 5000. Para ello, tenemos que modificar el fichero de configuración del servidor Tomcat ubicado en pentaho-server/tomcat/conf

```

<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector URIEncoding="UTF-8" port="5000" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--

```

Figura 91: Configuración puerto de la aplicación web Pentaho

Para lanzar el servidor, ubicándonos en la carpeta pentaho-server ejecutamos el siguiente comando:

```
./start --pentaho .sh
```

Tras ejecutar el comando, se lanzará el servidor Tomcat de la aplicación web. Una vez lanzado el servidor, accedemos desde el navegador a la dirección ipServidor:5000/pentaho. Se mostrará la pantalla de login de Pentaho

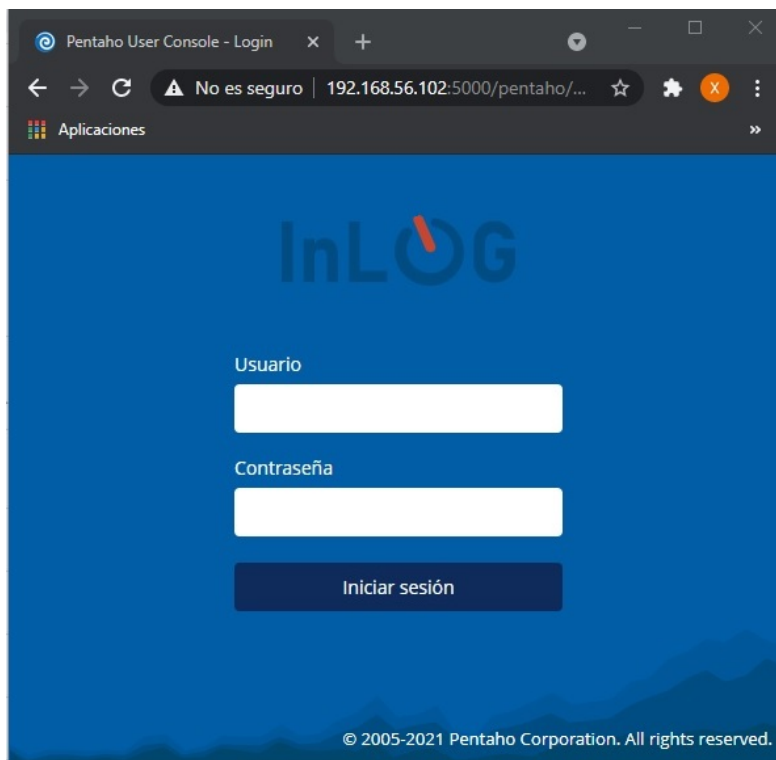


Figura 92: Pantalla login aplicación web Pentaho

Para parar el servidor, ubicándonos en la carpeta pentaho-server ejecutamos el siguiente comando:

```
./stop --pentaho .sh
```

8. Evaluación

Durante el desarrollo del proyecto, se han sometido las diferentes funcionalidades a una serie de pruebas para comprobar el buen funcionamiento de la solución.

Las funcionalidades a probar son las siguientes:

- Proceso ETL
- Medidas de los cubos OLAP
- Informes ad-hoc

Para realizar las pruebas, se ha utilizado un entorno de desarrollo, el cual es un reflejo del entorno de producción pero con un número pequeño y limitado de datos para facilitar las pruebas. Esto es necesario ya que en caso de realizar las pruebas contra el entorno de producción en un periodo de trabajo, los datos se modificarían constantemente haciendo imposible realizar pruebas contra estos.

8.1. Evaluación del proceso ETL

El proceso de carga de datos al Datawarehouse es uno de los procesos más importantes del proyecto, ya que una mala carga de datos puede suponer que se muestren datos erróneos en el resto de funcionalidades.

Por un lado, comprobaremos que el número de registros cargados a cada tabla de hechos y dimensiones sea correcto, comparándolo con el número de registros en la base de datos operacional.

Por otro lado, al haberse realizado cambios sobre las claves primarias de ARTICULOS, AREA_DESCARGA y PASILLO, añadiendo un campo numérico secuencial como clave, será necesario comprobar que las referencias a estas entidades realizadas en las tablas de hechos sean correctas y correspondan a la entidad original.

Debido a la gran importancia del proceso y para facilitar futuras implementaciones en distintos clientes en los que se pueda modificar ligeramente el proceso de carga de datos para añadir otras entidades, se han implementado unas pruebas unitarias automáticas haciendo uso de JUnit para poder comprobar rápidamente si el proceso de carga de datos funciona correctamente.

Se realizarán las siguientes pruebas:

1. Primera carga de datos completa, teniendo todas las tablas del Datawarehouse vacías.

Ejecutando las pruebas unitarias, comprobamos que la primera carga de datos al Datawarehouse se ha realizado correctamente. Por cada prueba ejecutada, se observa el resultado obtenido y el tiempo que ha tardado en ejecutarse:

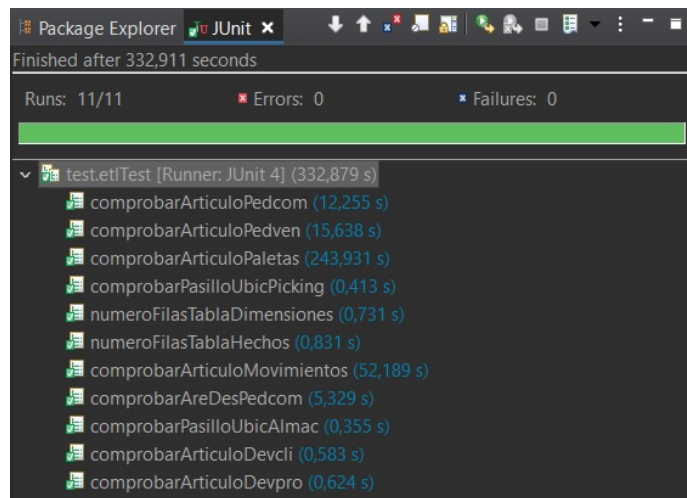


Figura 93: Resultado pruebas unitarias primera carga de datos

2. Carga de datos incremental, dando de alta en el sistema al menos un nuevo artículo, área de descarga y pasillo. Crearemos un pedido de compra cuyo área de descarga sea la nueva creada, y que entre sus líneas tenga al menos una referencia al nuevo artículo. Además, crearemos al menos una nueva ubicación de picking con el nuevo pasillo y volveremos a realizar la carga de datos.

Ejecutando las pruebas unitarias, podemos ver que la carga de datos se ha realizado correctamente, tratando correctamente los nuevos datos introducidos.

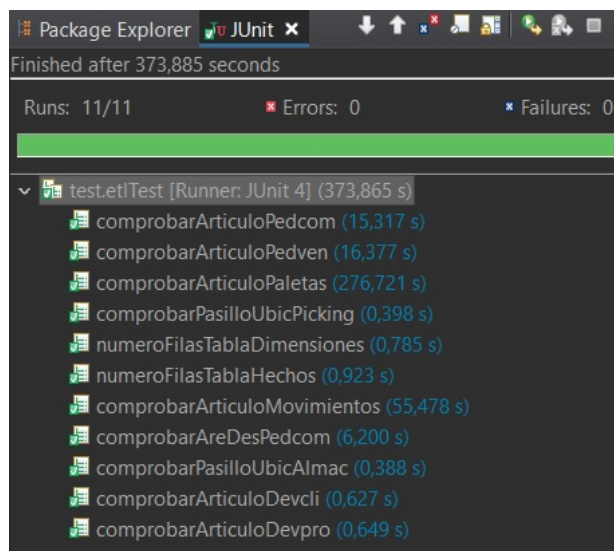


Figura 94: Resultado pruebas unitarias carga de datos incremental

8.2. Evaluación medidas cubos OLAP

Una vez comprobada que la carga de datos al Datawarehouse es correcta, el siguiente paso es comprobar que el cálculo de las medidas por cada uno de los cubos OLAP es correcto.

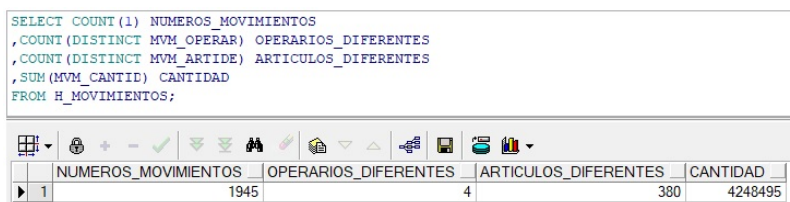
Para realizar las comprobaciones, se calculará por un lado los valores de las medidas de los cubos manualmente contra el Datawarehouse haciendo uso de consultas SQL desde PLSQL Developer, y se compararán estos valores con los obtenidos por las consultas MDX contra los cubos generadas en el visor OLAP.

Los resultados obtenidos son los siguientes:

1. CUBO MOVIMIENTOS

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

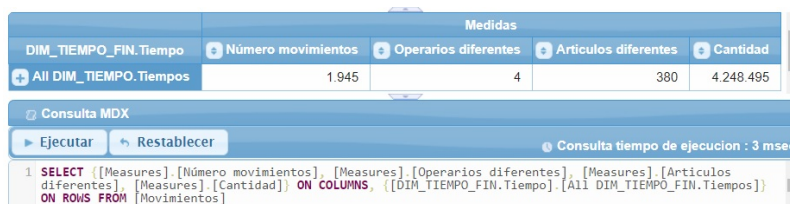
```
SELECT COUNT(1) NUMEROS.MOVIMIENTOS
, COUNT(DISTINCT MVM.OPERAR) OPERARIOS.DIFERENTES
, COUNT(DISTINCT MVM.ARTIDE) ARTICULOS.DIFERENTES
, SUM(MVM.CANTID) CANTIDAD
FROM H.MOVIMIENTOS;
```



	NUMEROS_MOVIMIENTOS	OPERARIOS_DIFERENTES	ARTICULOS_DIFERENTES	CANTIDAD
1	1945	4	380	4248495

Figura 95: Resultado medidas cubo movimientos mediante consulta SQL

Realizando la consulta MDX desde el visor OLAP obtenemos el siguiente resultado:



DIM_TIEMPO_FIN.Tiempo	Medidas			
	Número movimientos	Operarios diferentes	Articulos diferentes	Cantidad
All DIM_TIEMPO.Tiempos	1.945	4	380	4.248.495

Consulta MDX

Ejecutar Restablecer Consulta tiempo de ejecucion : 3 msec

```
1 SELECT {[Measures].[Número movimientos], [Measures].[Operarios diferentes], [Measures].[Articulos diferentes], [Measures].[Cantidad]} ON COLUMNS, {[DIM_TIEMPO_FIN.Tiempo].[All DIM_TIEMPO_FIN.Tiempos]} ON ROWS FROM [Movimientos]
```

Figura 96: Resultado medidas cubo movimientos mediante consulta MDX

Comparando los resultados obtenidos mediante ambas consultas, obtenemos los siguientes resultados:

	Consulta SQL	Consulta MDX	Resultado
Número movimientos	1945	1945	Correcto
Operarios diferentes	4	4	Correcto
Artículos diferentes	380	380	Correcto
Cantidad	4.248.495	4.248.495	Correcto

Tabla 38: Resultado pruebas medidas del cubo movimientos

2. CUBO PALETAS

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT COUNT(1) NUMERO.PALETAS
, COUNT(DISTINCT PAL.ARTIDE) ARTICULOS.DIFERENTES
, SUM(PAL.STOCPA) STOCK.PALETA
, COUNT(PAL.MOVFEC) NUM.MOVIMIENTOS
, MIN(PAL.STOCPA) STOCK.MINIMO
, MAX(PAL.STOCPA) STOCK.MAXIMO
FROM H.PALETAS;
```

	Consulta SQL	Consulta MDX	Resultado
Número paletas	6345	6345	Correcto
Artículos diferentes	3419	3419	Correcto
Stock paleta	4.714.587	4.714.587	Correcto
Número movimientos	1	1	Correcto
Stock mínimo	0	0	Correcto
Stock máximo	1.246.600	1.246.600	Correcto

Tabla 39: Resultado pruebas medidas del cubo paletas

3. CUBO CABECERA PEDIDO DE COMPRA

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT SUM(CPC.COSTE) COSTE
, COUNT(DISTINCT CPC.PROVEE) NUM.PROVEEDORES
, AVG(CPC.COSTE) MEDIA.COSTE
, SUM(CPC.NUMPAL) NUMERO.PALETAS
, COUNT(DISTINCT CPC.USUARI) NUM.OPERARIOS
, COUNT(1) NUMERO.PEDIDOS
FROM H.CABEC.PEDCOM;
```


	Consulta SQL	Consulta MDX	Resultado
Coste	510	510	Correcto
Proveedores diferentes	3	3	Correcto
Media coste	3	3	Correcto
Número paletas	1383	1383	Correcto
Operarios diferentes	5	5	Correcto
Número pedidos	170	170	Correcto

Tabla 40: Resultado pruebas medidas del cubo cabecera pedido de compra

4. CUBO LINEAS PEDIDO DE COMPRA

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT SUM(LPC.CANTEO) CANT_PEDIDA
, SUM(LPC.CANREA) CANT_RECIBIDA
, SUM(LPC.NUMPAL) NUM_PALETAS
, COUNT(1) NUM_LINEAS
, COUNT(DISTINCT LPC_PEDIDO) NUM_PEDIDOS
, COUNT(DISTINCT LPC_ARTIDE) NUM_ARTICULOS
FROM H.LINEAS_PEDCOM;
```

	Consulta SQL	Consulta MDX	Resultado
Cantidad pedida	1.717.381	1.717.381	Correcto
Cantidad recibida	1.206.898	1.206.898	Correcto
Número paletas	1383	1383	Correcto
Número líneas	355	355	Correcto
Número pedidos	170	170	Correcto
Artículos diferentes	48	48	Correcto

Tabla 41: Resultado pruebas medidas del cubo líneas pedido de compra

5. CUBO CABECERA PEDIDO DE VENTA

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT SUM(CPV_PRECIO) COSTE
, AVG(CPV_PRECIO) MEDIA_COSTE
, SUM(CPV_NUMPAL) NUM_PALETAS
, COUNT(1) NUM_PEDIDOS
, COUNT(DISTINCT CPV_CLIENT) NUM_CLIENTES
, SUM(CPV_PESOPC) PESO_PC
, SUM(CPV_PESOPK) PESO_PK
, SUM(CPV_VOLUPC) VOL_PC
, SUM(CPV_VOLUPK) VOL_PK
FROM H.CABEC_PEDVEN;
```

	Consulta SQL	Consulta MDX	Resultado
Coste	605	605	Correcto
Media coste	5	5	Correcto
Número paletas	22	22	Correcto
Número pedidos	121	121	Correcto
Clientes diferentes	5	5	Correcto
Peso paletas completas	105.950	105.950	Correcto
Peso picking	6.604.230	6.604.230	Correcto
Volumen paletas completas	186.680	186.680	Correcto
Volumen picking	21.894.048	21.894.048	Correcto

Tabla 42: Resultado pruebas medidas del cubo cabecera pedido de venta

6. CUBO LINEAS PEDIDO DE VENTA

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT SUM(LPV_CANPED) CANT_PEDIDA
,SUM(LPV_CANSER) CANT_SERVIDA
,COUNT (DISTINCT LPV_ARTIDE) NUMARTICULOS
,COUNT (DISTINCT LPV_PEDIDO || LPV_DIVPED) NUM_PEDIDOS
FROM H_LINEAS_PEDVEN;
```

	Consulta SQL	Consulta MDX	Resultado
Cantidad pedida	28.179	28.179	Correcto
Cantidad servida	29.614	29.614	Correcto
Artículos diferentes	28	28	Correcto
Número pedidos	121	121	Correcto

Tabla 43: Resultado pruebas medidas del cubo líneas pedido de venta

7. CUBOS CABECERA DEVOLUCIÓN CLIENTE/PROVEEDOR

La consulta SQL a realizar para obtener el cálculo de las medidas del cubo cabecera devolución cliente es la siguiente:

```
SELECT COUNT(DISTINCT CDC_CLIENT) NUM_CLIENTES
,COUNT(DISTINCT CDC_OPEAUT) NUM_OPERARIOS
,COUNT(DISTINCT CDC_MUELLE) NUM_MUELLES
,COUNT(1) NUM_DEVOLUCIONES
FROM H_CABEC_DEVCLI;
```

La consulta SQL a realizar para obtener el cálculo de las medidas del cubo cabecera devolución proveedor es la siguiente:

```
SELECT COUNT(DISTINCT CDP_PROVEE) NUMPROVEEDORES
,COUNT(DISTINCT CDP_OPEDEV) NUMOPERARIOS
,COUNT(DISTINCT CDP_MUELLE) NUMMUELLES
,COUNT(1) NUMDEVOLUCIONES
FROM H.CABEC_DEVPRO;
```

	Consulta SQL Cliente	Consulta MDX Cliente	Consulta SQL Proveedor	Consulta MDX Proveedor	Resultado
Cli/Prov diferentes	1	1	2	2	Correcto
Operarios diferentes	3	3	3	3	Correcto
Muelles	2	2	3	3	Correcto
Número devoluciones	4	4	3	3	Correcto

Tabla 44: Resultado pruebas medidas del cubo cabecera devolución cliente/proveedor

8. CUBO LINEAS DEVOLUCIÓN CLIENTE

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT SUM(LDC.CANTOT) CANT_TOTAL
,SUM(LDC.CANTAP) CANT_APTA
,SUM(LDC.CANTNA)
,COUNT(DISTINCT LDC_OPERAR) NUMOPERARIOS
,COUNT(DISTINCT LDC_ARTIDE) NUMARTICULOS
FROM H.LINEAS_DEVCLI;
```

	Consulta SQL	Consulta MDX	Resultado
Cantidad total	333	333	Correcto
Cantidad apta	293	293	Correcto
Cantidad no apta	40	40	Correcto
Número operarios	2	2	Correcto
Artículos diferentes	3	3	Correcto

Tabla 45: Resultado pruebas medidas del cubo líneas devolución cliente

9. CUBO LINEAS DEVOLUCIÓN PROVEEDOR

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT SUM(LDP_CANTID) CANTID
, COUNT(DISTINCT LDP_PALETA) NUMPAL
, COUNT(DISTINCT LDP_ARTIDE) NUMARTICULOS
, COUNT(DISTINCT LDP_OPERAR) NUMOPERARIOS
FROM H.LINEAS.DEVPRO;
```

	Consulta SQL	Consulta MDX	Resultado
Cantidad	25	25	Correcto
Número paletas	5	5	Correcto
Artículos diferentes	3	3	Correcto
Número operarios	2	2	Correcto

Tabla 46: Resultado pruebas medidas del cubo líneas devolución proveedor

10. CUBO ORDENES DE PALETA COMPLETA

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT COUNT(DISTINCT OPC_ARTIDE) NUMARTICULOS
, SUM(OPC_CANPAL) CANT_PALPED
, SUM(OPC_CANREC) CANT_RECIBIDA
, COUNT(DISTINCT OPC_OPERAR) NUMOPERARIOS
FROM H.ORDENES_PC;
```

	Consulta SQL	Consulta MDX	Resultado
Artículos diferentes	7	7	Correcto
Cantidad paletas pedidas	1059	1059	Correcto
Cantidad recibida	245	245	Correcto
Número operarios	2	2	Correcto

Tabla 47: Resultado pruebas medidas del cubo ordenes de paleta completa

11. CUBO ORDENES DE PICKING

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT COUNT(DISTINCT OPK_PEDIDO || OPK_DIVPED) NUMPEDIDOS
, COUNT(DISTINCT OPK_OPERAR) NUMOPERARIOS
, SUM(OPK_TOTLIN) NUMLINEAS
, SUM(OPK_NUMCON) NUMCONTEN
, SUM(OPK_VOLCON) VOLCONTEN, SUM(OPK_PESCON) PESCONTEN
FROM H.ORDENES_PK;
```

	Consulta SQL	Consulta MDX	Resultado
Número pedidos	80	80	Correcto
Operarios diferentes	3	3	Correcto
Número de líneas	408	408	Correcto
Número de contenedores	820	820	Correcto
Volumen contenedores	1.127.220.000	1.127.220.000	Correcto
Peso contenedores	662.295.000	662.295.000	Correcto

Tabla 48: Resultado pruebas medidas del cubo ordenes de picking

12. CUBO LÍNEAS DE PICKING

La consulta SQL a realizar para obtener el cálculo de las medidas es la siguiente:

```
SELECT SUM(LPK_CANTID) CANT.PEDIDA
,SUM(LPK_CANREC) CANT.SERVIDA
,COUNT(DISTINCT LPK_ARTIDE) NUMARTICULOS
FROM H.LINEAS_PK ;
```

	Consulta SQL	Consulta MDX	Resultado
Cantidad pedida	23.179	23.179	Correcto
Cantidad servida	15.010	15.010	Correcto
Artículos diferentes	25	25	Correcto

Tabla 49: Resultado pruebas medidas del cubo líneas de picking

13. CUBOS UBICACIÓN DE PICKING/ALMACENAJE

La consulta SQL a realizar para obtener el cálculo de las medidas del cubo ubicación de picking es la siguiente:

```
SELECT SUM(UPLPALUBI) NUMPALUBI
,COUNT(DISTINCT UPLARTIDE) NUMARTICULOS
FROM H.UBIC_PICKING ;
```

La consulta SQL a realizar para obtener el cálculo de las medidas del cubo ubicación de almacenaje es la siguiente:

```
SELECT SUM(UALPALUBI) NUMPALUBI
,COUNT(DISTINCT UALARTIDE) NUMARTICULOS
FROM H.UBIC_ALMAC ;
```

	Consulta SQL Picking	Consulta MDX Picking	Consulta SQL Almacén	Consulta MDX Almacén	Resultado
Número paletas ubicadas	2352	2352	3218	3218	Correcto
Artículos diferentes	3262	3262	3	3	Correcto

Tabla 50: Resultado pruebas medidas cubos ubicación de picking y almacenaje

8.3. Evaluación consultas ad-hoc

Para verificar que los datos mostrados en las consultas ad-hoc son correctos, se definirá por cada modelo de metadatos una consulta que obtenga información de varias tablas, y se obtendrán los datos tanto realizando directamente una consulta SQL contra el Datawarehouse utilizando PLSQL Developer, como realizando la consulta ad-hoc mediante Pentaho, comparando que los resultados obtenidos mediante ambas consultas sean iguales.

Debido al límite de 1000 registros añadido en las consultas ad-hoc, añadiremos filtros suficientes para mostrar resultados con un número de registros inferior a este número.

Los resultados obtenidos son los siguientes:

1. PEDIDOS DE COMPRA

Mostraremos el código de pedido, código línea, nombre propietario, denominación artículo de cada línea y cantidad teórica a recepcionar de los pedidos de compra confirmados del propietario con nombre Xabier. Ordenaremos los resultados obtenidos según la cantidad teórica a recepcionar de forma descendente.

PEDIDO	CODIGO LINEA	NOMBRE PROPIETARIO	DENOMINACION ARTICULO	CANTIDAD TEORICA
2605	6	XABIER	FORVEN C CON RET	140000
2604	6	XABIER	FORVEN C CON RET	116000
94	3	XABIER	PACK NADAL SIR EDWARD'S WHISKY 1 LTR7890	59994
2706	1	XABIER	FORVEN UDS	38240
2605	1	XABIER	FORVEN UDS	26000
2605	5	XABIER	FORVEN C CON RET	26000
2706	2	XABIER	FORVEN C SIN RET	25975
2605	2	XABIER	FORVEN C SIN RET	25000
2701	1	XABIER	FORVEN UDS	20000
2605	3	XABIER	FORVEN C SIN RET	20000

Showing 1 to 10 of 276 entries Previous **1** 2 3 4 5 ... 28 Next

Figura 97: Resultado consulta ad-hoc pedidos de compra

Realizaremos la misma consulta mediante SQL para comprobar si los datos obtenidos son los mismos:

```

SELECT CPC_PEDIDO PEDIDO, LPC_CODLIN CODIGOLINEA
,PFR_NOMBRE PROPIETARIO
,ART_DENOMI DENOM_ARTICULO, LPC_CANTEO CANT_TEOERICA
FROM H_CABEC_PEDCOM, H_LINEAS_PEDCOM, DIM_PROPIETARIOS, DIM_ARTICULOS
WHERE CPC_PEDIDO=LPC_PEDIDO
AND CPC_PROPIE=PFR_CODIGO
AND LPC_ARTIDE=ART_IDENTI
AND PFR_NOMBRE='XABIER'
AND CPC_SITPED='CO'
ORDER BY LPC_CANTEO DESC;

```

PEDIDO	CODIGOLINEA	PROPIETARIO	DENOM_ARTICULO	CANT_TEOERICA
1	2605	6 XABIER	FORVEN C CON RET	140000
2	2604	6 XABIER	FORVEN C CON RET	116000
3	94	3 XABIER	PACK NADAL SIR EDWARD'S WHISKY 1 LTR7890	59994
4	2706	1 XABIER	FORVEN UDS	38240
5	2605	1 XABIER	FORVEN UDS	26000
6	2605	5 XABIER	FORVEN C CON RET	26000
7	2706	2 XABIER	FORVEN C SIN RET	25975
8	2605	2 XABIER	FORVEN C SIN RET	25000
9	2701	1 XABIER	FORVEN UDS	20000
10	2605	3 XABIER	FORVEN C SIN RET	20000

Figura 98: Resultado consulta SQL pedidos de compra

El número registro y datos mostrados coinciden entre la consulta realizada manualmente por SQL y la consulta realizada mediante la herramienta de consultas ad-hoc.

2. PEDIDOS DE VENTA

Mostraremos pedido, división, denominación del almacén del pedido, nombre de propietario, número de orden de picking, denominación artículo de la orden, cantidad a servir y cantidad recogida del pedido 106/Z. Ordenaremos los resultados obtenidos según el número de orden.

PEDIDO	DIVISION	DENOMINACION ALMACEN	NOMBRE PROPIETARIO	NUMERO ORDEN	DENOMINACION ARTICULO	CANTIDAD A SERVIR	CANTIDAD RECOGIDA
106	Z	ALMACEN 001	XABIER	1	SID GOT PER LOTS	6	6
106	Z	ALMACEN 001	XABIER	2	PACK NADAL SIR EDWARD'S WHISKY 1 LTR7890	270	270
106	Z	ALMACEN 001	XABIER	2	LOT SIR EDWARD'S 1L + SIR EDWARD'S 0,50L	225	225
106	Z	ALMACEN 001	XABIER	2	LOT SIR EDWARD'S 1,5L + SIR EDWARD'S 0,5	30	30

Showing 1 to 4 of 4 entries Previous Next

Figura 99: Resultado consulta ad-hoc pedidos de venta

Realizaremos la misma consulta mediante SQL para comprobar si los datos obtenidos son los mismos:

```

SELECT CPV_PEDIDO PEDIDO, CPV_DIVPED DIVISION
,ALM_DENOMI DEN_ALMACEN, PPR_NOMBRE PROPIETARIO
,LFK_NUMORD NUM_ORDEN, ART_DENOMI DENOM_ARTICULO
,LFV_CANSEC CANT_SER, LFK_CANSEC CANT_REC
FROM H_CABEC PEDVEN, H_LINEAS PEDVEN
H_LINEAS PK, DIM_ALMACENES ,DIM_PROPIETARIOS, DIM_ARTICULOS
WHERE CPV_PEDIDO=LFV_PEDIDO
AND CPV_DIVPED=LFV_DIVPED
AND CPV_ALMACE=ALM_CODIGO
AND CPV_PROPIE=PPR_CODIGO
AND LFV_PEDIDO=LFK_PEDIDO
AND LFV_DIVPED=LFK_DIVPED
AND LFV_CODLIN=LFK_LINPED
AND LFV_ARTIDE=ART_IDENTI
AND CPV_PEDIDO='106' AND CPV_DIVPED='Z'
ORDER BY LFK_NUMORD;

```

PEDIDO	DIVISION	DEN_ALMACEN	PROPIETARIO	NUM_ORDEN	DENOM_ARTICULO	CANT_SER	CANT_REC
1	106	Z	ALMACEN 001	XABIER	1	SID GOT PER LOTS	6
2	106	Z	ALMACEN 001	XABIER	2	PACK NADAL SIR EDWARD'S WHISKY 1 LTR7890	270
3	106	Z	ALMACEN 001	XABIER	2	LOT SIR EDWARD'S 1L + SIR EDWARD'S 0.50L	225
4	106	Z	ALMACEN 001	XABIER	2	LOT SIR EDWARD'S 1.5L + SIR EDWARD'S 0.5	30

Figura 100: Resultado consulta SQL pedidos de venta

Los datos mostrados coinciden entre la consulta realizada manualmente por SQL y la consulta realizada mediante la herramienta de consultas adhoc.

3. MAESTROS

Mostraremos código artículo, denominación artículo, denominación del almacén del artículo y nombre del propietario del artículo con código BIC_001.

CODIGO ARTICULO	DENOMINACION ARTICULO	DENOMINACION ALMACEN	NOMBRE PROPIETARIO
BIC_001	BIC NARANJA	ALMACEN 001	ANDOLAC

Showing 1 to 1 of 1 entries Previous **1** Next

Figura 101: Resultado consulta ad-hoc maestros

Realizaremos la misma consulta mediante SQL para comprobar si los datos obtenidos son los mismos:

```

SELECT ART_CODIGO ARTICULO, ART_DENOMI DENOM_ARTICULO
, ALM_DENOMI DENOM_ALMACE, PPR_NOMBRE PROPIETARIO
FROM DIM_ARTICULOS, DIM_ALMACENES, DIM_PROPIETARIOS
WHERE ART_ALMACE=ALM_CODIGO
AND ART_PROPIE=PPR_CODIGO
AND ART_CODIGO='BIC_001';

```

ARTICULO	DENOM_ARTICULO	DENOM_ALMACE	PROPIETARIO
1	BIC_001	BIC NARANJA	ALMACEN 001
			ANDOLAC

77:2 pentaho@PENTAHD_DEV 1 row selected in 0.015 seconds

Figura 102: Resultado consulta SQL maestros

Los datos mostrados coinciden entre la consulta realizada manualmente por SQL y la consulta realizada mediante la herramienta de consultas adhoc.

4. CONTROL

Mostraremos código movimiento, denominación del tipo movimiento, nombre propietario y denominación de artículo de aquellos movimientos de tipo picking y artículo con código 0000001. Ordenaremos los resultados obtenidos según el código de movimiento.

CODIGO MOVIMIENTO	DENOMINACION TIPO MOVIMIENTO	NOMBRE PROPIETARIO	DENOMINACION ARTICULO
1549	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO
1563	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO
1567	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO
1571	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO
1603	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO

Showing 1 to 5 of 5 entries Previous **1** Next

Figura 103: Resultado consulta ad-hoc control

Realizaremos la misma consulta mediante SQL para comprobar si los datos obtenidos son los mismos:

```
SELECT MVM_CODIGO CODMOV, TMV_DENOMI DEN_TIPMOV
, PPR_NOMBRE PROPIETARIO, ART_DENOMI DENOM_ARTICULO
FROM H_MOVIMIENTOS, DIM_TIPOS_MOVIMIENTO, DIM_PROPIETARIOS, DIM_ARTICULOS
WHERE MVM_TIPMOV=TMV_CODIGO
AND MVM_TIPMOV='PI'
AND MVM_ARTIDE=ART_IDENTI
AND ART_PROPIE=PPR_CODIGO
AND ART_CODIGO='0000001'
ORDER BY MVM_CODIGO;
```

	CODMOV	DEN_TIPMOV	PROPIETARIO	DENOM_ARTICULO
1	1549	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO
2	1563	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO
3	1567	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO
4	1571	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO
5	1603	Picking	XABIER	IPHONE 8 GRIS PLATA 125 GB. EUROPEO

95.3 | pentaho@PENTAH0_DEV | 5 rows selected in 0,015 seconds

Figura 104: Resultado consulta SQL control

Los datos mostrados coinciden entre la consulta realizada manualmente por SQL y la consulta realizada mediante la herramienta de consultas adhoc.

9. Conclusiones y trabajo futuro

9.1. Planificación final

En primer lugar, se ha logrado alcanzar todos los objetivos definidos en el planteamiento del proyecto.

A continuación, se muestra un gráfico comparativo en el que se puede ver la desviación temporal que ha sufrido la planificación del proyecto por cada uno de los apartados definidos:

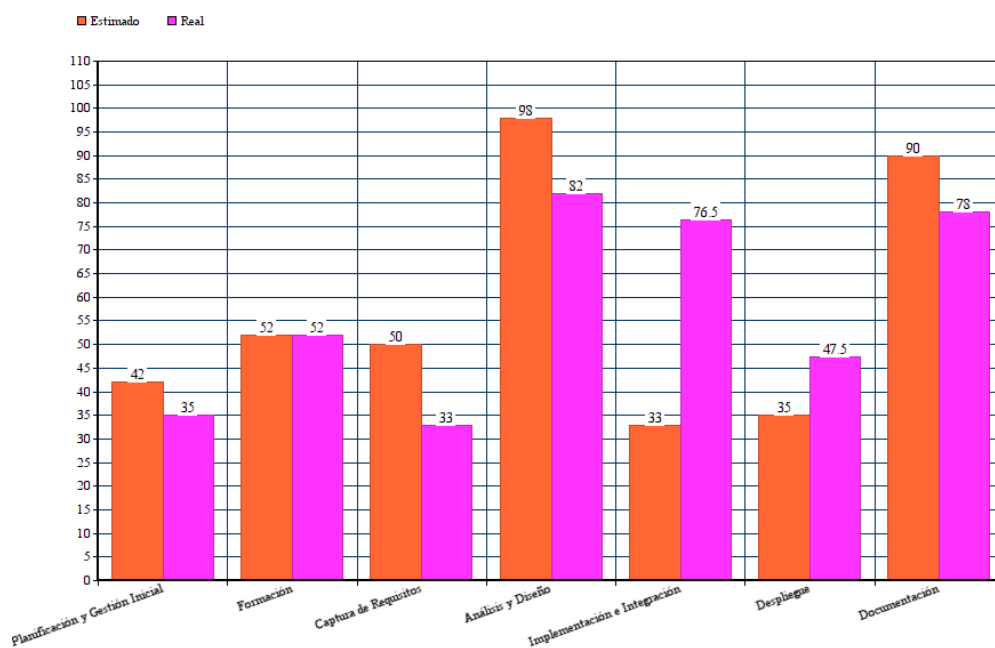


Figura 105: Gráfica comparación horas estimadas y reales

En la siguiente tabla podemos ver desglosado el tiempo total dedicado a cada tarea:

TAREA	DURACIÓN ESTIMADA	DURACIÓN REAL
Definición de Objetivos y Alcance del Proyecto	16 horas	12 horas
Planificación temporal del proyecto	6 horas	4,5 horas
Gestión de riesgos	3 horas	3,5 horas
Planificación económica	3 horas	3 horas
Estudio de antecedentes	14 horas	12 horas
Aprender conceptos básicos de Business Intelligence	14 horas	14 horas

Aprender funcionamiento de la plataforma Pentaho	28 horas	28 horas
Aprender LaTeX	10 horas	10 horas
Identificación de funcionalidades	10 horas	8 horas
Casos de Uso	40 horas	25 horas
Diseño del Datawarehouse	40 horas	36 horas
Diseño de los cubos OLAP	30 horas	28 horas
Diseño de los informes	28 horas	18 horas
Implementación e integración del Datawarehouse	15 horas	28 horas
Implementación de cubos OLAP y visor	10 horas	18,5 horas
Integración de informes ad-hoc	8 horas	30 horas
Despliegue del servidor web	15 horas	25 horas
Pruebas de implementación	20 horas	22,5 horas
Realización de la memoria	80 horas	70 horas
Preparación de la defensa	10 horas	8 horas
TOTAL	400 horas	404 horas

Tabla 51: Comparación horas estimadas y reales

Como se puede observar tanto en la gráfica como en la tabla, se ha sobrestimado el tiempo necesario para realizar las tareas de planificación, análisis y diseño del proyecto, siendo necesario un tiempo menor al estimado en la mayoría de las tareas.

Sin embargo, debido en gran parte al desconocimiento del funcionamiento de Pentaho durante las fases de planificación del proyecto, ha sido necesario emplear mucho más tiempo del estimado en las fases de desarrollo, despliegue y evaluación de la aplicación, al ser necesario modificar y añadir nuevas funcionalidades al sistema no incluidas por defecto en Pentaho.

Pese a esto, el número de horas totales empleadas para realizar el proyecto es similar al estimado, y además se ha logrado terminar el proyecto en el plazo establecido, con lo cual, no hay ninguna modificación en la planificación económica, y pese a haber surgido el riesgo de una mala planificación estimada, finalmente el impacto en el proyecto ha sido pequeño.

9.2. Reflexiones personales

Tras finalizar el proyecto, la sensación general hacia el resultado obtenido es satisfactoria.

Por un lado, pese a contar con los consejos de personas con años de experiencia en puestos de gerencia, me he dado cuenta lo difícil que es planificar y llevar adelante un proyecto de esta magnitud, siendo necesario ser constante en la realización de los distintos puntos del proyecto para poder llevarlo a cabo.

Además, el trabajar implementando un sistema de Business Intelligence, una tecnología cada vez más demandada por las empresas, creo que ha sido muy enriquecedor para mi formación como profesional, y además, al realizar el proyecto sobre un entorno real proporcionado por la empresa en la que estoy trabajando como becario, he podido aprender con un proyecto que va más allá del entorno académico en el cual había trabajado hasta ahora, viendo los resultados de mi proyectos aplicados a un cliente real.

Al ser Pentaho una plataforma open source, no solo he implementado la solución de Business Intelligence, sino que haciendo uso de la documentación oficial de Pentaho, consultando la documentación publicada por la comunidad en foros y mis conocimientos de programación adquiridos durante mi periodo académico y laboral, poder añadir modificar las características de este para ajustarlo a mis necesidades, aprendiendo así a trabajar con software de código abierto.

9.3. Trabajo futuro

Los datos cargados al Datawarehouse y utilizados en el sistema, son aquellas entidades más importantes y comunes en el sistema InLOG WMS, pero estas son una pequeña parte del conjunto de entidades pertenecientes al sistema, con lo que, quedaría como trabajo futuro ir incorporando las distintas entidades al proyecto.

En cuanto a la herramienta para realizar consultas ad-hoc, sería interesante extender las funcionalidades que esta ofrece, añadiendo la posibilidad de realizar agrupaciones, introducir campos calculados (sumatorios, medias...) y poder añadir filtros más complejos (menor que, contiene, empieza por...) para poder realizar consultas de un nivel mayor al ofrecido actualmente.

Pese a haber traducido la aplicación web directamente a Español, al ser el idioma hablado por la mayoría de los clientes de InLOG actualmente, sería interesante desarrollar la aplicación en varios idiomas, como Castellano, Euskera e Inglés, y ofrecer la posibilidad de permitir al cliente elegir el idioma desde la propia aplicación.

Por último, queda como trabajo futuro realizar un proyecto de formación tanto al resto de desarrolladores, como a los clientes de InLOG WMS para que puedan usar correctamente la aplicación.

10. Referencias

- [1] Inteligencia empresarial: transformación de datos en decisiones. <https://www.captio.net/blog/inteligencia-empresarial-la-transformacion-de-los-datos-en-decisiones-optimas> (Accedido el 12/02/2021)
- [2] ¿En qué consiste un proceso de ETL (Extraer, Transformar y Cargar)? <https://www.talend.com/es/resources/what-is-etl/> (Accedido el 12/02/2021)
- [3] Proceso de BI — Dataprix TI <https://www.dataprix.com/es/data-warehousing-y-metodologia-hefesto/13-proceso-bi> (Accedido el 12/02/2021)
- [4] Beneficios — Dataprix TI <https://www.dataprix.com/es/data-warehousing-y-metodologia-hefesto/14-beneficios> (Accedido el 12/02/2021)
- [5] Datawarehouse https://www.sinnexus.com/business_intelligence/datawarehouse.aspx (Accedido el 12/02/2021)
- [6] Data base y Data warehouse ¿En qué se diferencian? <https://smartable.es/es/blog/41-data-base-y-data-warehouse-en-que-se-diferencian> (Accedido el 12/02/2021)
- [7] La base de datos analítica (el Datawarehouse o Almacén de Datos) <https://churriwifi.wordpress.com/2009/11/28/3-la-base-de-datos-analitica-el-datawarehouse-o-almacen-de-datos/> (Accedido el 12/02/2021)
- [8] Uso de cubos OLAP para análisis <https://docs.microsoft.com/es-es/system-center/scsm/olap-cubes-overview?view=sc-sm-2019> (Accedido el 12/02/2021)
- [9] Cubos OLAP - EcuRed https://www.ecured.cu/Cubos_OLAP (Accedido el 12/02/2021)
- [10] Cuadro de Mando Integral https://www.sinnexus.com/business_intelligence/cuadro_mando_integral.aspx (Accedido el 12/02/2021)
- [11] El cuadro de mando en Business Intelligence — EAE <https://retos-operaciones-logistica.eae.es/el-cuadro-de-mando-como-herramienta-de-business-intelligence/> (Accedido el 12/02/2021)
- [12] ¿Qué son los metadatos y cuál es su utilidad? <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-son-los-metadatos-y-cual-es-su-utilidad> (Accedido el 12/02/2021)

- [13] ¿Qué es Power BI? — Deloitte España
<https://www2.deloitte.com/es/es/pages/technology/articles/que-es-power-bi.html> (Accedido el 13/02/2021)
- [14] ¿Qué es QlikView?
https://help.qlik.com/es-ES/qlikview/April2020/Content/QV_HelpSites/what-is.htm (Accedido el 13/02/2021)
- [15] SAP Business Intelligence - Cibernetica
<https://www.cibernetica.net/sap-business-intelligence/> (Accedido el 13/02/2021)
- [16] Pentaho Big Data - Business Intelligence España - Stratebi
<https://www.stratebi.com/pentaho> (Accedido el 12402/2021)
- [17] Características y diferencias entre Pentaho y Power BI
<https://openwebinars.net/blog/caracteristicas-y-diferencias-entre-pentaho-y-power-bi/>
 (Accedido el 14/02/2021)
- [18] Comparativa Qlikview vs Pentaho
<https://todobi.com/comparativa-qlikview-vs-pentaho/> (Accedido el 14/02/2021)
- [19] Modelo Dimensional - BI Geek Blog
<https://blog.bi-geek.com/modelo-dimencional/> (Accedido el 24/03/2021)
- [20] Data Warehouse y Data Marts - Conoce las diferencias
<https://blogs.solidq.com/es/business-analytics/data-warehouse-y-data-marts-esquema-en-estrella-11/> (Accedido el 25/03/2021)
- [21] Pentaho Data Integration - Pentaho Documentation
https://help.pentaho.com/Documentation/8.2/Products/Data_Integration (Accedido el 29/03/2021)
- [22] Informes parametrizados con report designer en Pentaho CE <https://openwebinars.net/blog/informes-parametrizados-con-report-designer-en-pentaho-ce>
 (Accedido el 29/03/2021)
- [23] Aspectos básicos de las consultas MDX <https://docs.microsoft.com/es-es/analysis-services/multidimensional-models/mdx/mdx-query-fundamentals-analysis-services> (Accedido el 29/03/2021)
- [24] Pentaho Schema Workbench - Pentaho Documentation
https://help.pentaho.com/Documentation/8.2/Products/Schema_Workbench (Accedido el 29/03/2021)
- [25] Pentaho Metadata Editor
https://help.pentaho.com/Documentation/8.0/Products/Metadata_Editor (Accedido el 29/03/2021)
- [26] Qué es Pentaho BI Server <https://openwebinars.net/blog/que-es-pentaho-bi-server/> (Accedido el 29/03/2021)

- [27] Archivo PRPT
<https://www.filesuffix.com/es/extension/prpt> (Accedido el 02/04/2021)
- [28] Qué es un esquema de base de datos
<https://www.lucidchart.com/pages/es/que-es-un-esquema-de-base-de-datos>
(Accedido el 03/04/2021)
- [29] La Guía Definitiva Sobre La Dimension Tiempo En Un Data Warehouse
<https://www.elfuturodelosdatos.com/dimension-tiempo-data-warehouse>
(Accedido el 05/04/2021)
- [30] Kitchen User Documentation
<https://wiki.pentaho.com/display/EAI/Kitchen+User+Documentation> (Accedido el 17/04/2021)
- [31] Cron – Configuración básica para programar tareas mediante crontab <https://laguialinux.es/cron-configuracion-basica-para-programar-tareas-mediante-crontab/>
(Accedido el 18/04/2021)
- [32] Cómo usar Cron para automatizar tareas en CentOS 8
<https://www.digitalocean.com/community/tutorials/how-to-use-cron-to-automate-tasks-centos-8-es> (Accedido el 18/04/2021)
- [33] Pentaho Report Designer
https://help.pentaho.com/Documentation/8.2/Products/Report_Designer (Accedido el 24/04/2021)
- [34] Pentaho Report Designer User Guide
https://docs.huihoo.com/pentaho/pentaho-business-analytics/4.8/report_designer_user_guide.pdf (Accedido el 24/04/2021)
- [35] JPivot
<http://jpivot.sourceforge.net/> (Accedido el 29/04/2021)
- [36] Pivot4J para visualizar cubos OLAP en Java
<https://rubenjgarcia.cloud/pivot4j-cubos-olap-java/> (Accedido el 29/04/2021)
- [37] CDE Dashboard Overview
https://help.pentaho.com/Documentation/7.0/0R0/CTools/CDE_Dashboard_Overview (Accedido el 30/04/2021)
- [38] https://help.pentaho.com/Documentation/8.0/Products/Metadata_Editor (Accedido el 01/05/2021)

11. Anexo I: Casos de uso extendidos

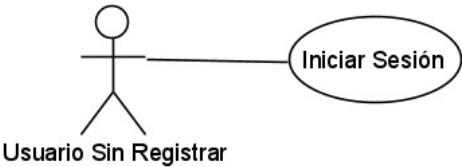
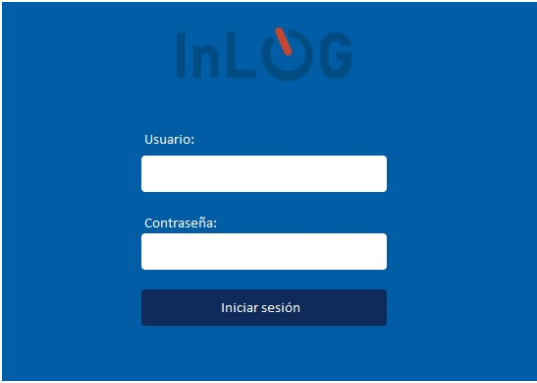
Iniciar Sesión	
 <p style="text-align: center;">Usuario Sin Registrar</p>	
Descripción	El usuario se identifica en el sistema mediante una cuenta dada de alta en el sistema
Precondiciones	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario introduce el usuario y contraseña 2. El usuario pulsa el botón identificarse: 3. [Si el registro es correcto:] <ol style="list-style-type: none"> 3.1 Se accede al sistema. [Si algo falla] <ol style="list-style-type: none"> 3.2 Se muestra un mensaje indicando el error.
Postcondiciones	-
Interfaz gráfica	



Tabla 52: Caso de Uso: Iniciar Sesión

Programar Tareas	
<pre> graph LR Usuario((Usuario)) --- ProgramarTareas([Programar tareas]) </pre>	
Descripción	Permite al usuario programar tareas a ejecutar desde la aplicación web.
Precondiciones	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario, desde el menú Ficheros, selecciona el recurso del cual programar una tarea. 2. Pulsa en Programar tarea y se abre una ventana para programar la tarea. 3. El usuario introduce un nombre para la tarea programada y la ruta donde se guardara el resultado y pulsa en Continuar. 4. El usuario introduce la periodicidad de repetición de la tarea y la fecha-hora en la que se ejecutará y pulsa en Siguiente. 5. El usuario escoge el formato de la salida y pulsa en Finalizar. <p>El usuario puede consultar las tareas programadas desde la opción Programaciones.</p>
Postcondiciones	Se programa la ejecución de una tarea en el sistema en la fecha seleccionada durante el periodo especificado.

Interfaz gráfica

Archivos

- Report Ejemplo

Acciones

- Abrir
- Abrir en nueva ventana
- Ejecutar en background

- Cortar
- Copiar
- Eliminar
- Renombrar

- Compartir
- Programar tarea
- Añadir a favoritos
- Propiedades

Nueva programación

Nombre de la programación

Ubicación

Nueva programación

Recurrencia

Hora de inicio

Fecha de inicio

Nueva programación

Esta programación correrá usando los siguientes parámetros:

Parametro A

Tipo de Salida

Volver
Finalizar
Cancelar

Programaciones admin

Administrar programaciones 🔍 📄 🗑️ 🔄

Nombre de la programación	Repeticiones	Archivo fuente	Ubicación de salida	Última ejecución	próxima ejecución	Creado por	Estado
No hay programaciones.							

Tiempo de interrupción
 No se han creado restricciones de fechas para la programación.

[Crear tiempo de interrupción](#)

Tabla 53: Caso de Uso: Programar Tareas

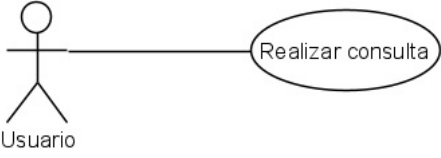
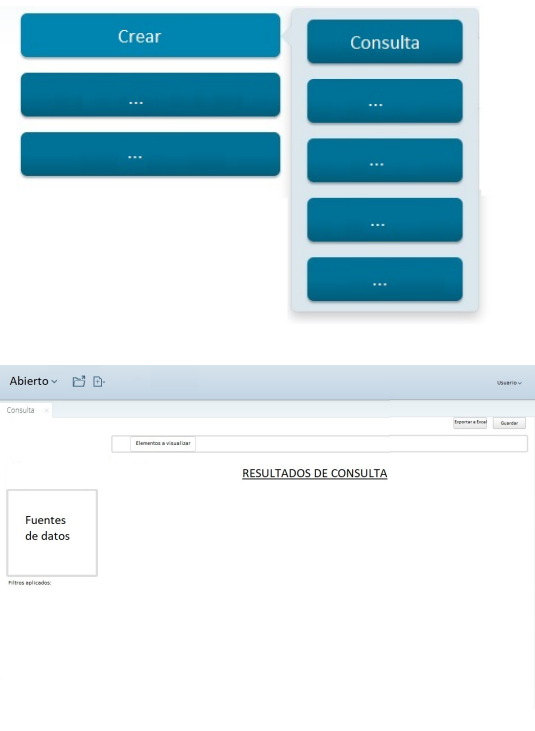
Realizar consulta	
	
Descripción	Permite realizar al usuario consultas contra el Datawarehouse
Precondiciones	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario pulsa en Crear / Consulta 2. El usuario selecciona los datos a consultar. 3. Se muestran los datos correspondientes a la consulta introducida.
Postcondiciones	-
Interfaz gráfica	

Tabla 54: Caso de Uso: Realizar consulta

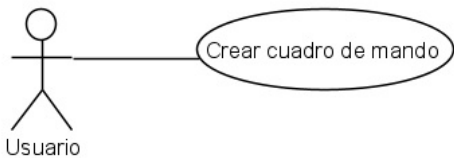

Crear cuadro de mando	
	
Descripción	Permite diseñar y crear cuadros de mando personalizados
Precondiciones	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario pulsa en Crear / Cuadro de mando 2. Se muestra un dashboard para diseñar el cuadro de mando
Postcondiciones	Se crea un nuevo cuadro de mando
Interfaz gráfica	

Tabla 55: Caso de Uso: Crear cuadro de mando

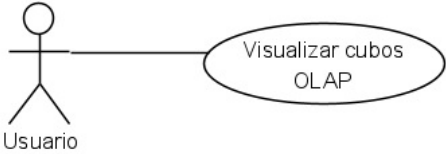
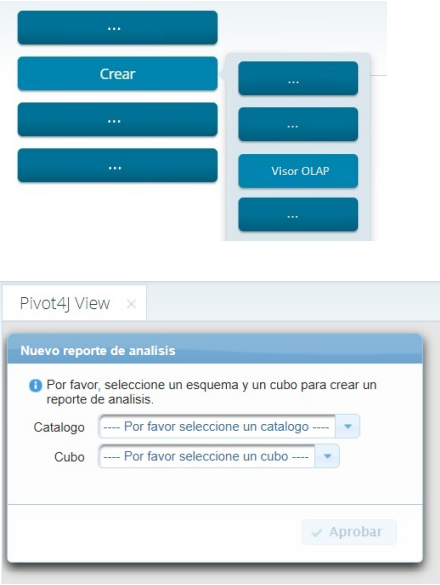
Visualizar cubos OLAP	
	
Descripción	Permite visualizar los cubos OLAP diseñados y desplegados.
Precondiciones	Debe haber algún cubo desplegado en la aplicación.
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario pulsa en Crear / Visor OLAP. 2. El usuario selecciona el cubo a visualizar y pulsa el botón Aprobar. 3. Se muestra la interfaz en la que se visualiza el cubo OLAP, permitiendo al usuario realizar consultas sobre el.
Postcondiciones	-
Interfaz gráfica	

Tabla 56: Caso de Uso: Visualizar cubos OLAP


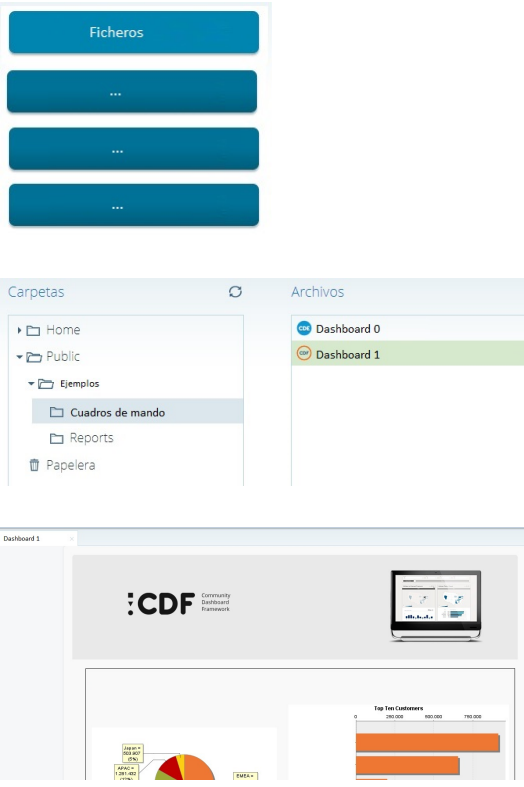
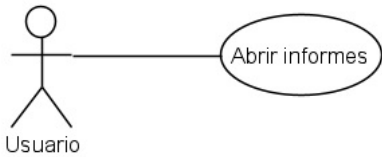
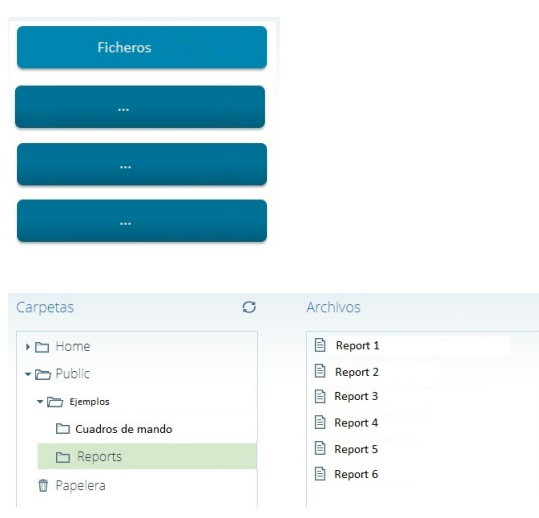
Abrir cuadros de mando	
	
Descripción	Permite abrir y visualizar los cuadros de mando creados anteriormente.
Precondiciones	Debe haber algún cuadro de mando creado.
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario pulsa en Ficheros . 2. El usuario selecciona y abre el cuadro de mando a visualizar. 3. Se visualiza el cuadro de mando.
Postcondiciones	-
Interfaz gráfica	

Tabla 57: Caso de Uso: Abrir cuadros de mando

Abrir informes	
	
Descripción	Permite abrir y visualizar los informes creados y desplegados anteriormente.
Precondiciones	Debe haber algún informe desplegado en la aplicación.
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario pulsa en Ficheros. 2. El usuario selecciona y abre el informe a visualizar. 3. Se visualiza el informe, permitiendo al usuario modificar los diferentes parámetros para el informe.
Postcondiciones	-
Interfaz gráfica	

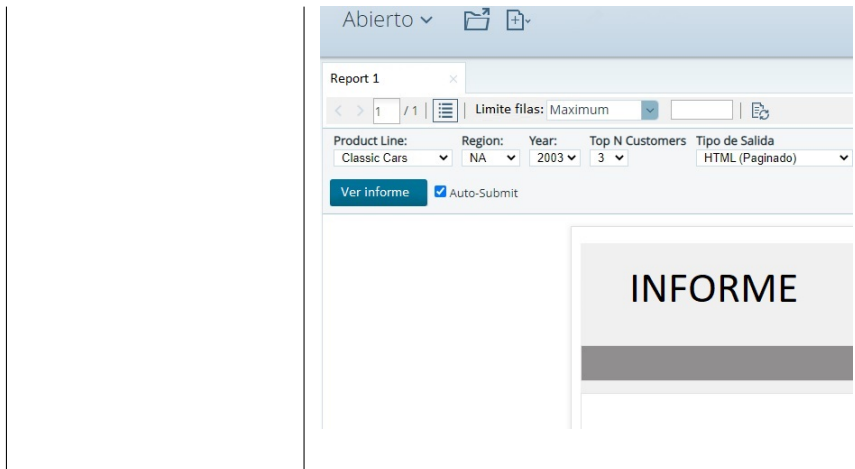


Tabla 58: Caso de Uso: Abrir informes

Administrar fuentes de datos	
Descripción	Permite crear y modificar las fuentes de datos con las que se trabaja desde la aplicación web.
Precondiciones	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario pulsa en Administrar fuentes de datos. 2. Se despliega una ventana para crear o modificar las diferentes fuentes de datos con las que trabaja la aplicación web.
Postcondiciones	-
Interfaz gráfica	

Fuentes de datos ⚙️ Nuevo

Fuente de datos	Tipo
Inlog DWH	JDBC
Tabla 1	Data Source Wizard

Cerrar

Nuevo

Selección origen

Selección tablas

Definir joins

Nombre:

Tipo: Tabla de base de datos

Conexión: ✎ + ✕

Inlog DWH

Crear fuente de datos para:

Solo informes

Informes y análisis

Tabla 59: Caso de Uso: Administrar fuentes de datos

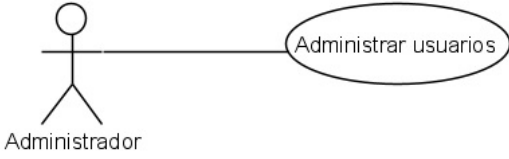
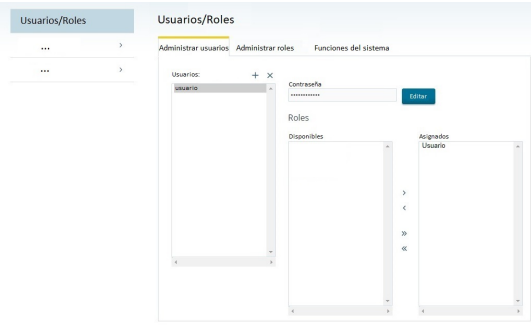
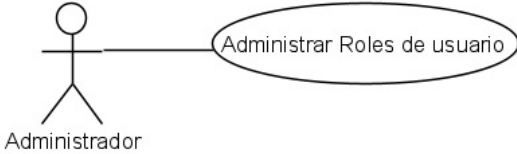
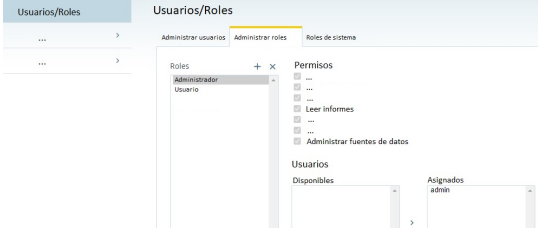
Administrar usuarios	
	
Descripción	Permite crear y modificar los usuarios de la aplicación web.
Precondiciones	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. El administrador pulsa en Administración 2. Se muestra el menú para administrar el sistema. 3. El administrador pulsa en Usuarios / Roles 4. Se muestra el menú para administrar los usuarios de la aplicación web.
Postcondiciones	-
Interfaz gráfica	

Tabla 60: Caso de Uso: Administrar usuarios

Administrar Roles de usuario	
	
Descripción	Permite modificar los permisos de cada usuario de la aplicación web.
Precondiciones	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. El administrador pulsa en Administración 2. Se muestra el menú para administrar el sistema. 3. El administrador pulsa en Usuarios / Roles 4. El administrador pulsa en Administrar roles o Funciones del sistema 5. Se muestra el menú para administrar los distintos roles y asignárselos a los distintos usuarios del sistema.
Postcondiciones	-
Interfaz gráfica	

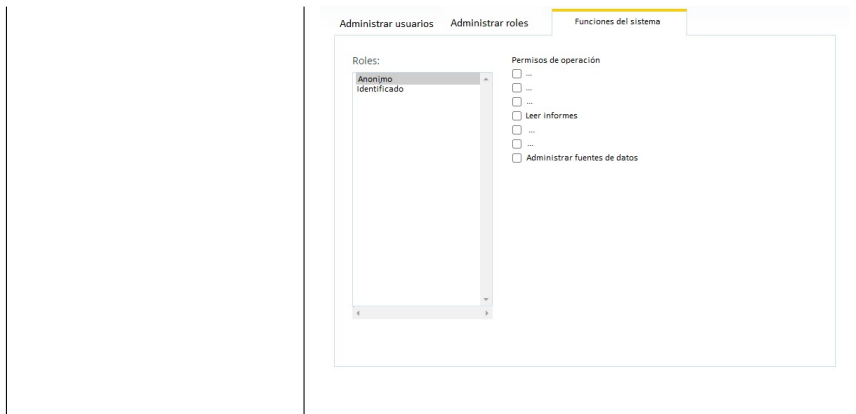


Tabla 61: Caso de Uso: Administrar Roles de usuario

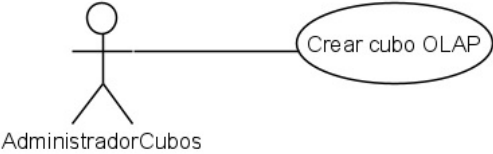
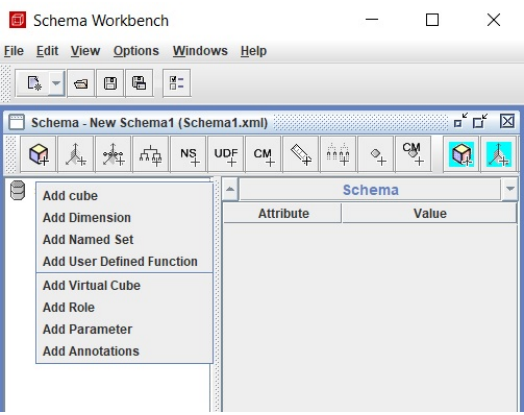
Crear cubo OLAP	
	
Descripción	Permite crear y diseñar cubos OLAP.
Precondiciones	Debe estar creado el Datawarehouse
Flujo de eventos	<ol style="list-style-type: none"> 1. Abrir la aplicación de escritorio Pentaho Schema Workbench 2. Abrir un schema existente o crear uno nuevo
Postcondiciones	Se crea un fichero .xml con los datos del schema creado
Interfaz gráfica	

Tabla 62: Caso de Uso: Crear cubo OLAP

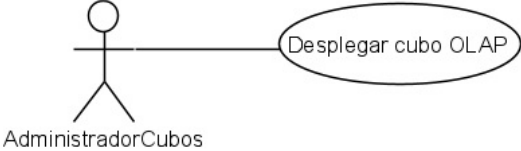
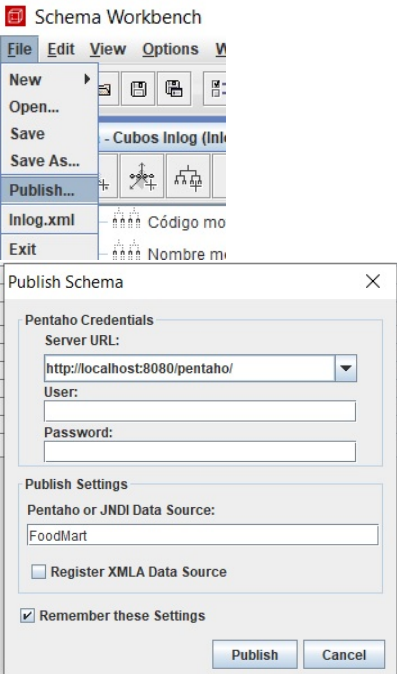
Desplegar cubo OLAP	
	
Descripción	Permite desplegar un cubo OLAP en la aplicación web.
Precondiciones	Debe haberse creado un schema con los datos de los cubos a desplegar
Flujo de eventos	<ol style="list-style-type: none"> 1. Abrir la aplicación de escritorio Pentaho Schema Workbench 2. Abrir el schema con los datos de los cubos 3. Pulsar File/Publish
Postcondiciones	Se publican los cubos definidos en la aplicación web Pentaho
Interfaz gráfica	

Tabla 63: Caso de Uso: Desplegar cubo OLAP

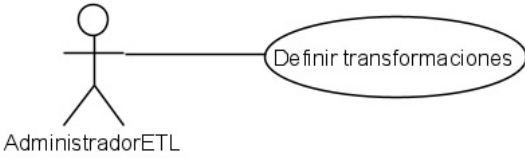
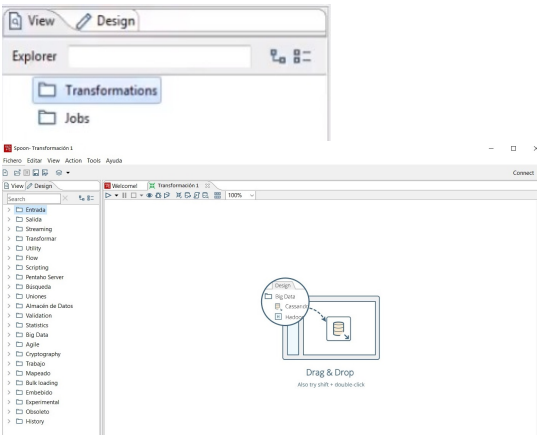
Definir transformaciones	
	
Descripción	Permite definir y modificar el proceso de carga de datos al Datawarehouse desde las diferentes fuentes
Precondiciones	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. Abrir la aplicación de escritorio Pentaho Data Integration 2. En la pestaña View, seleccionar la opción Transformations o abrir una transformación desde File / Open
Postcondiciones	Se crea un fichero de extensión .ktr con la transformación definida
Interfaz gráfica	

Tabla 64: Caso de Uso: Definir transformaciones

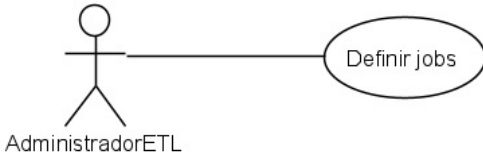
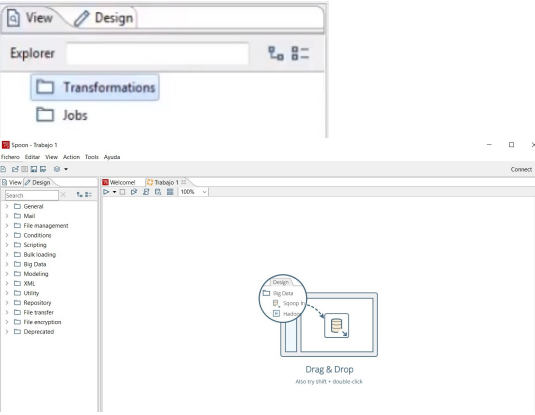
Definir jobs	
 <p style="text-align: center;">Administrador ETL</p>	
Descripción	Permite modificar la forma y la periodicidad de carga de datos al Datawarehouse
Precondiciones	Ninguna
Flujo de eventos	<ol style="list-style-type: none"> 1. Abrir la aplicación de escritorio Pentaho Data Integration 2. En la pestaña View, seleccionar la opción Jobs o abrir un job ya definido desde File / Open
Postcondiciones	Se crea un fichero de extensión .kjb con la transformación definida
Interfaz gráfica	

Tabla 65: Caso de Uso: Definir jobs

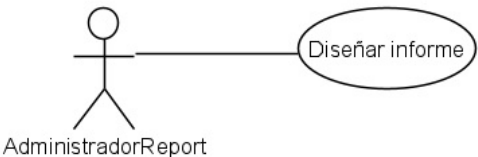
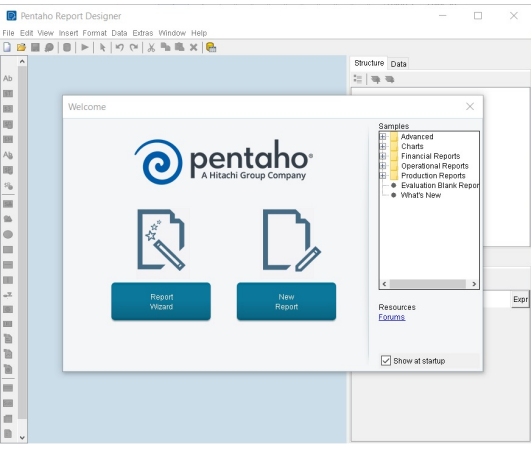
Diseñar informe	
	
Descripción	Permite diseñar informes predefinidos parametrizables
Precondiciones	Debe estar creado el Datawarehouse
Flujo de eventos	<ol style="list-style-type: none"> 1. Abrir la aplicación de escritorio Pentaho Report Designer 2. En la pestaña View, seleccionar la opción Jobs o abrir un job ya definido desde File / Open
Postcondiciones	Se genera un fichero de extensión .prpt con el report diseñado
Interfaz gráfica	

Tabla 66: Caso de Uso: Diseñar informe

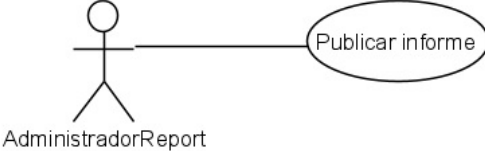
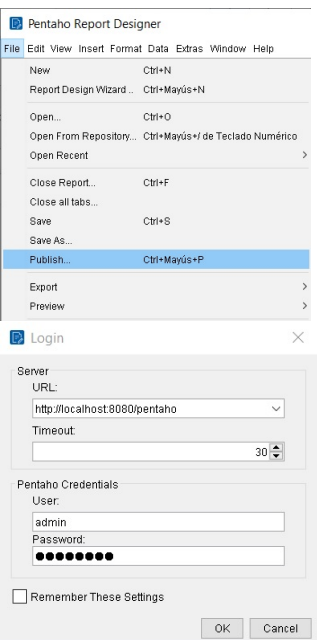
Publicar informe	
	
Descripción	Permite desplegar informes creados anteriormente en la aplicación web
Precondiciones	Debe haber un informe creado
Flujo de eventos	<ol style="list-style-type: none"> 1. Abrir la aplicación de escritorio Pentaho Report Designer 2. Abrir el informe a publicar 3. Pulsar File/Publish
Postcondiciones	El informe es accesible desde la aplicación web
Interfaz gráfica	

Tabla 67: Caso de Uso: Publicar informe

12. Anexo II: Definición de metadatos

12.1. Compras

Metadato correspondiente a los pedidos de compra del sistema.

Pedidos de Compra

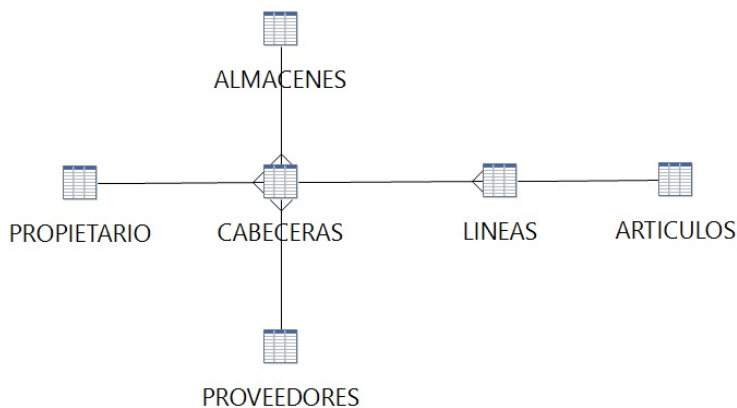


Figura 106: Diagrama metadato de compras

Está formado por las siguientes tablas:

- Cabecera de pedido de compra (H_CABEC_PEDCOM)
- Líneas de pedido de compra (H_LINEAS_PEDCOM)
- Almacenes (DIM_ALMACENES)
- Propietarios (DIM_PROPIETARIOS)
- Proveedores (DIM_PROVEEDOR)
- Artículos (DIM_ARTICULOS)

Se definen las siguientes relaciones:

Tabla desde	Tabla hasta	Campos relacionados	Tipo	Cardinalidad
H_CABEC_PEDCOM	H_LINEAS_PEDCOM	CPC_PEDIDO = LPC_PEDIDO	Inner	1:N
H_CABEC_PEDCOM	DIM_ALMACENES	CPC_ALMACE = ALM_CODIGO	Inner	N:1
H_CABEC_PEDCOM	DIM_PROPIETARIOS	CPC_PROPIE = PPR_CODIGO	Inner	N:1

H.CABEC_PEDCOM	DIM.PROVEEDOR	CPC_PROVEE = PRO_CODIGO	Inner	N:1
H.LINEAS_PEDCOM	DIM.ARTICULOS	LPC_ARTIDE = ART_IDENTI	Inner	N:1

Tabla 68: Relaciones metadato de compras

12.2. Ventas

Metadato correspondiente a los pedidos de venta del sistema.

Pedidos de Venta

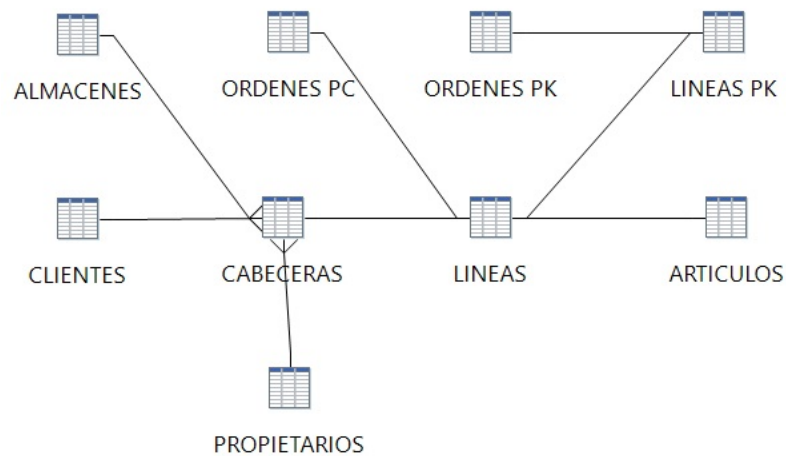


Figura 107: Diagrama metadato de ventas

Está formado por las siguientes tablas:

- Cabecera de pedido de venta (H.CABEC_PEDVEN)
- Líneas de pedido de venta (H.LINEAS_PEDVEN)
- Almacenes (DIM.ALMACENES)
- Propietarios (DIM.PROPIETARIOS)
- Clientes (DIM.CLIENTES)
- Artículos (DIM.ARTICULOS)
- Ordenes de paleta completa (H.ORDENES_PC)
- Ordenes de picking (H.ORDENES_PK)
- Líneas de picking (H.LINEAS_PK)

Se definen las siguientes relaciones:

Tabla desde	Tabla hasta	Campos relacionados	Tipo	Cardinalidad
H_CABEC_PEDVEN	H.LINEAS_PEDVEN	CPV_PEDIDO = LPV_PEDIDO, CPV_DIVPED = LPV_DIVPED	Inner	1:N
H_CABEC_PEDVEN	DIM.ALMACENES	CPV_ALMACE = ALM_CODIGO	Inner	N:1
H_CABEC_PEDVEN	DIM.PROPIETARIOS	CPV_PROPIE = PPR_CODIGO	Inner	N:1
H_CABEC_PEDVEN	DIM.CLIENTES	CPV_CLIENT = CLI_CODIGO	Inner	N:1
H.LINEAS_PEDVEN	DIM.ARTICULOS	LPV_ARTIDE = ART_IDENTI	Inner	N:1
H.ORDENES_PC	H.LINEAS_PEDVEN	OPC_PEDIDO = LPV_PEDIDO, OPC_DIVPED = LPV_DIVPED, OPC_CODLIN = LPV_CODLIN	Inner	N:1
H.ORDENES_PK	H.LINEAS_PK	OPK_PEDIDO = LPK_PEDIDO, OPK_DIVPED = LPK_DIVPED, OPK_NUMORD = LPK_NUMORD	Inner	1:N
H.LINEAS_PK	H.LINEAS_PEDVEN	LPK_PEDIDO = LPV_PEDIDO, LPK_DIVPED = LPV_DIVPED, LPK_LINPED = LPV_CODLIN	Inner	N:1

Tabla 69: Relaciones metadato de ventas

12.3. Maestros

Metadato correspondiente a los maestros de las entidades del sistema.

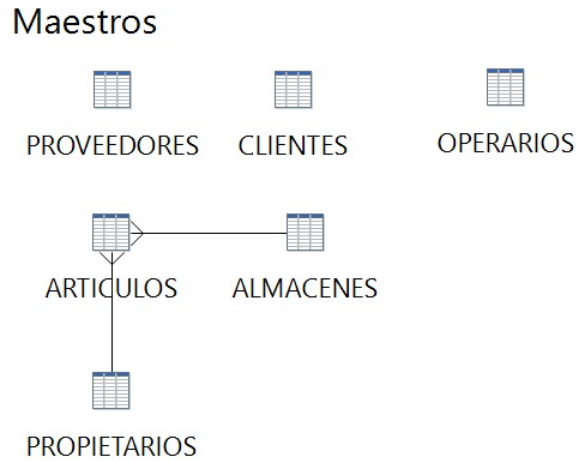


Figura 108: Diagrama metadato de maestros

Está formado por las siguientes tablas:

- Proveedores (DIM_PROVEEDOR)
- Almacenes (DIM_ALMACENES)
- Propietarios (DIM_PROPIETARIOS)
- Clientes (DIM_CLIENTES)
- Artículos (DIM_ARTICULOS)
- Operarios (DIM_OPERARIOS)

Se definen las siguientes relaciones:

Tabla desde	Tabla hasta	Campos relacionados	Tipo	Cardinalidad
DIM_ARTICULOS	DIM_ALMACENES	ART_ALMACE = ALM_CODIGO	Inner	N:1
DIM_ARTICULOS	DIM_PROPIETARIOS	ART_PROPIE = PPR_CODIGO	Inner	N:1

Tabla 70: Relaciones metadato de maestros

12.4. Control

Metadato correspondiente al control de los movimientos en el almacén. Está

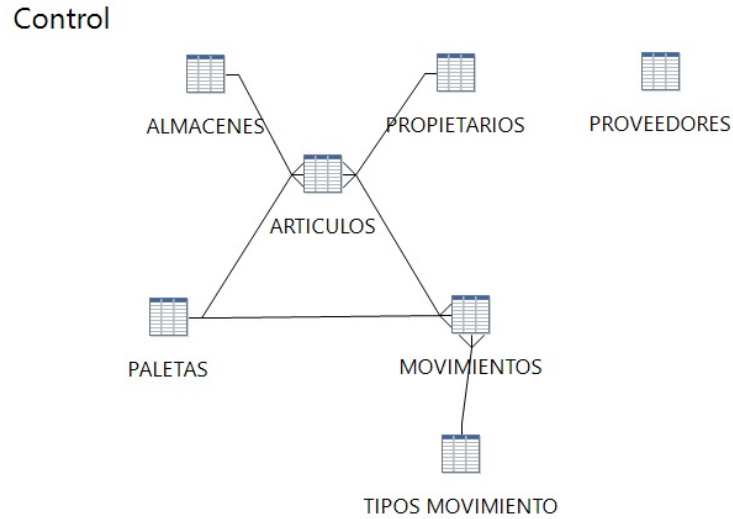


Figura 109: Diagrama metadato de control

formado por las siguientes tablas:

- Movimientos (H.MOVIMIENTOS)
- Tipos Movimiento (H.TIPOS_MOVIMIENTO)
- Paletas (H.PALETAS)
- Proveedores (DIM.PROVEEDOR)
- Almacenes (DIM.ALMACENES)
- Propietarios (DIM.PROPIETARIOS)
- Clientes (DIM.CLIENTES)
- Artículos (DIM.ARTICULOS)

Se definen las siguientes relaciones:

Tabla desde	Tabla hasta	Campos relacionados	Tipo	Cardinalidad
H.MOVIMIENTOS	H.TIPOS_MOVIMIENTO	MVM_TIPMOV = TMV_CODIGO	Inner	N:1
H.PALETAS	H.MOVIMIENTOS	PAL_CODIGO = MVM_PALETA	Inner	1:N

H_MOVIMIENTOS	DIM_ARTICULOS	MVM_ARTIDE = ART_IDENTI	Inner	N:1
DIM_ARTICULOS	H_PALETAS	ART_IDENTI = PAL_ARTIDE	Inner	N:1
DIM_ARTICULOS	DIM_ALMACENES	ART_ALMACE = ALM_CODIGO	Inner	N:1
DIM_ARTICULOS	DIM_PROPIETARIOS	ART_PROPIE = PPR_CODIGO	Inner	N:1

Tabla 71: Relaciones metadato de control

13. Anexo III: Pruebas unitarias del proceso ETL

Para la implementación de las pruebas del proceso ETL, se utilizará la herramienta de testing automático en Java JUnit 4.

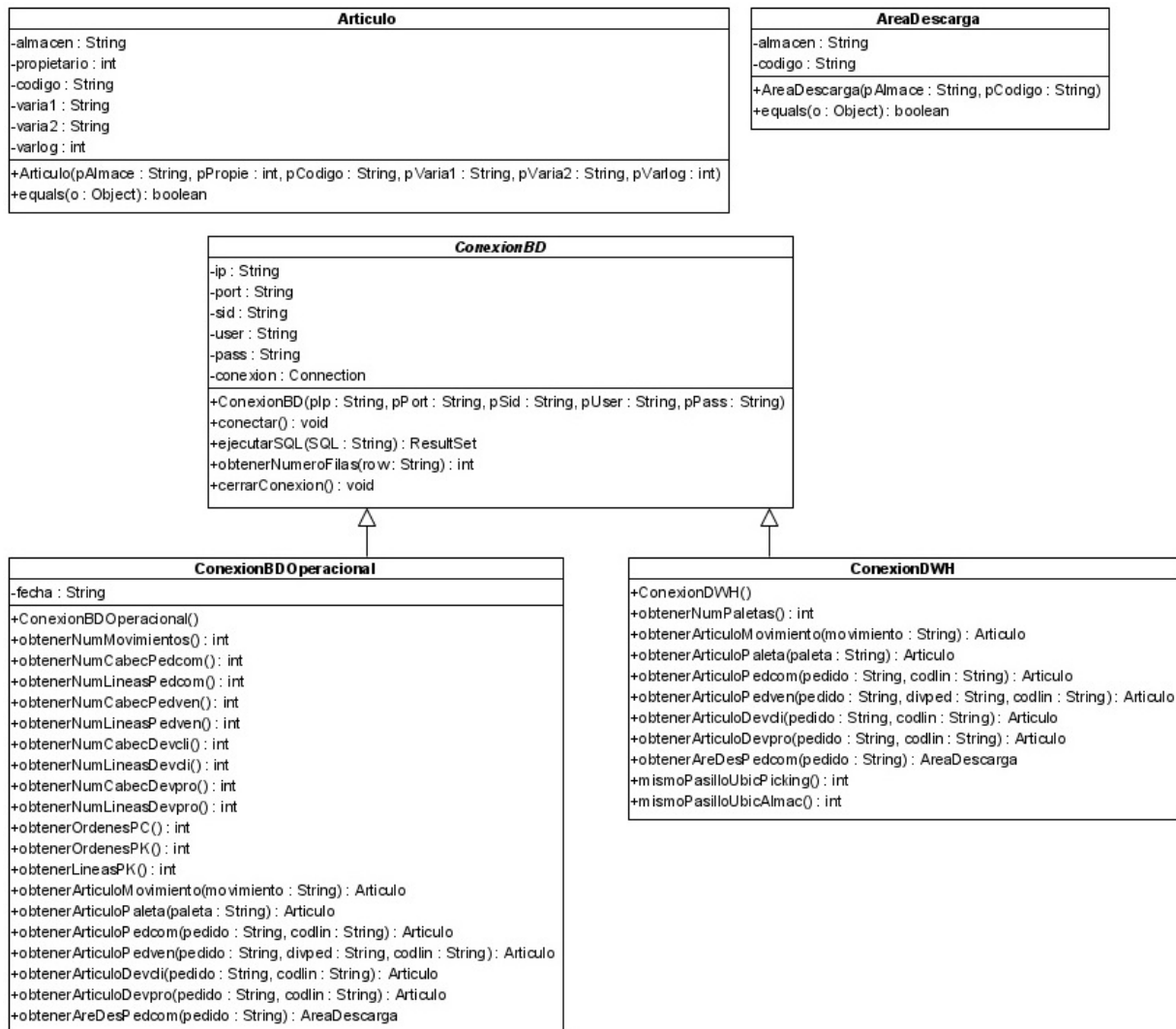


Figura 110: Diagrama de clases JUnits proceso ETL

La clase Articulo contendrá los datos de un artículo del sistema. Se reemplazará el método equals para comprobar si dos artículos son iguales.

La clase AreaDescarga contendrá los datos de un área de descarga del sistema. Se reemplazará el método equals para comprobar si dos artículos son iguales.

La clase `ConexionBD`, es una clase abstracta la cual contendrá los métodos comunes necesarios para trabajar con ambas bases de datos.

- **Constructor:** Recibe como parámetros los datos de conexión a la base de datos, se encarga de recogerlos y llamar al método `connect` para iniciar la conexión.
- **Conectar:** Conecta con la base de datos.
- **ejecutarSQL:** Ejecuta y devuelve el resultado de la sentencia SQL recibida como parámetro.
- **obtenerNumeroFilas:** Devuelve el número de registros en la tabla recibida como parámetro.
- **cerrarConexion:** Cierra la conexión una vez finalizada las operaciones contra la base de datos para evitar fugas de memoria.

La clase `ConexionDWH`, es una clase que hereda de `ConexionBD`, la cual contendrá los métodos necesarios para trabajar contra el Datawarehouse.

- **Constructor:** Llama al constructor de la clase padre pasándole como parámetro los datos de conexión al Datawarehouse.
- **obtenerNumPaletas:** Obtiene el número de registros en la tabla paletas sin contar históricos.
- **obtenerArticuloMovimiento:** Devuelve un objeto del tipo `Artículo` con la referencia al artículo del movimiento pasado como parámetro.
- **obtenerArticuloPaleta:** Devuelve un objeto del tipo `Artículo` con la referencia al artículo de la paleta pasada como parámetro.
- **obtenerArticuloPedcom:** Devuelve un objeto del tipo `Artículo` con la referencia al artículo de la línea de pedido de compra pasada como parámetro.
- **obtenerArticuloPedven:** Devuelve un objeto del tipo `Artículo` con la referencia al artículo de la línea de pedido de venta pasada como parámetro.
- **obtenerArticuloDevcli:** Devuelve un objeto del tipo `Artículo` con la referencia al artículo de la línea de devolución de cliente pasada como parámetro.
- **obtenerArticuloDevpro:** Devuelve un objeto del tipo `Artículo` con la referencia al artículo de la línea de devolución de proveedor pasada como parámetro.
- **mismoPasilloUbicPicking:** Devuelve el número de filas de las ubicaciones de picking cuyo pasillo es el mismo al pasillo referenciado.
- **mismoPasilloUbicAlmac:** Devuelve el número de filas de las ubicaciones de almacenaje cuyo pasillo es el mismo al pasillo referenciado.

La clase `ConexionBDOperacional` es una clase que hereda de `ConexionBD`, la cual contendrá un atributo que indica la fecha de carga de datos al Datawarehouse y los métodos necesarios para trabajar contra la base de datos operacional.

- Constructor: Llama al constructor de la clase padre pasándole como parámetro los datos de conexión a la base de datos operacional.
- `obtenerNumMovimientos`: Devuelve el número de registros en la tabla `MOVIMIENTOS` cuya fecha es superior a la fecha de carga de datos al Datawarehouse establecida.
- `obtenerNumCabecPedcom`: Devuelve el número de registros en la tabla `CABEC_PEDCOM` cuya fecha asignada es superior a la fecha de carga de datos al Datawarehouse establecida.
- `obtenerNumLineasPedcom`: Devuelve el número de registros de la tabla `LINEAS_PEDCOM` cuya fecha de cabeceras es superior a la fecha de carga de datos al Datawarehouse establecida.
- `obtenerNumCabecPedven`: Devuelve el número de registros en la tabla `CABEC_PEDVEN` cuya fecha de transmisión es superior a la fecha de carga de datos al Datawarehouse establecido.
- `obtenerNumLineasPedven`: Devuelve el número de registros de la tabla `LINEAS_PEDVEN` cuya fecha de cabeceras es superior a la fecha de carga de datos al Datawarehouse establecida.
- `obtenerNumCabecDevcli`: Devuelve el número de registros en la tabla `CABEC_DEVCLI` cuya fecha de devolución es superior a la fecha de carga de datos al Datawarehouse establecida.
- `obtenerNumLineasDevcli`: Devuelve el número de registros de la tabla `LINEAS_DEVCLI` cuya fecha de cabeceras es superior a la fecha de carga de datos al Datawarehouse establecida.
- `obtenerNumCabecDevpro`: Devuelve el número de registros en la tabla `CABEC_DEVPRO` cuya fecha de devolución es superior a la fecha de carga de datos al Datawarehouse establecida.
- `obtenerNumLineasDevpro`: Devuelve el número de registros de la tabla `LINEAS_DEVPRO` cuya fecha de cabeceras es superior a la fecha de carga de datos al Datawarehouse establecida.
- `obtenerOrdenesPC`: Devuelve el número de registros de la tabla `ORDENES_PC` cuyos pedidos han sido cargados al Datawarehouse.
- `obtenerOrdenesPK`: Devuelve el número de registros de la tabla `ORDENES_PK` cuyos pedidos han sido cargados al Datawarehouse.
- `obtenerLineasPK`: Devuelve el número de registros de la tabla `LINEAS_PC` cuyas ordenes han sido cargadas al Datawarehouse.
- `obtenerArticuloMovimiento`: Devuelve un objeto del tipo `Artículo` con la referencia al artículo del movimiento pasado como parámetro.

- obtenerArticuloPaleta: Devuelve un objeto del tipo Artículo con la referencia al artículo de la paleta pasada como parámetro.
- obtenerArticuloPedcom: Devuelve un objeto del tipo Artículo con la referencia al artículo de la línea de pedido de compra pasada como parámetro.
- obtenerArticuloPedven: Devuelve un objeto del tipo Artículo con la referencia al artículo de la línea de pedido de venta pasada como parámetro.
- obtenerArticuloDevcli: Devuelve un objeto del tipo Artículo con la referencia al artículo de la línea de devolución de cliente pasada como parámetro.
- obtenerArticuloDevpro: Devuelve un objeto del tipo Artículo con la referencia al artículo de la línea de devolución de proveedor pasada como parámetro.

Se implementarán los distintos test automáticos en el fichero etlTest.java. Los métodos implementados son los siguientes:

- numeroFilasTablasHechos y numeroFilasTablasDimensiones comprobarán que el número de filas insertadas al Datawarehouse de las diferentes tablas de hechos y dimensiones sea correcto.
- comprobarArticuloMovimientos, comprobarArticuloPaletas, comprobarArticuloPedcom, comprobarArticuloPedven, comprobarArticuloDevcli, comprobarArticuloDevpro comprobarán que la nueva referencia al artículo basada en un campo numérico sea correcta.
- comprobarAreDesPedcom comprobará que la nueva referencia a el área de descarga basada en un campo numérico sea correcta.
- comprobarPasilloUbicPicking y comprobarPasilloUbicAlmac comprobarán que la nueva referencia a el pasillo basada en un campo numérico sea correcta.

Para obtener los datos necesarios en estos métodos se utilizarán las clases definidas anteriormente.

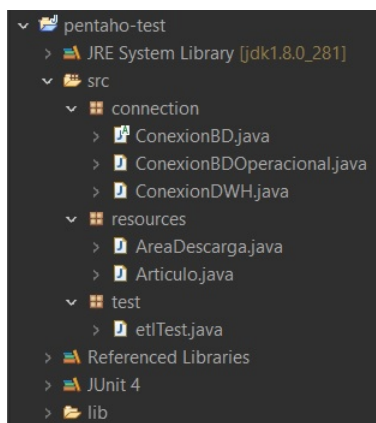


Figura 111: Estructura proyecto JUnits del proceso ETL

14. Anexo IV: Manual de usuario

14.1. Introducción

En este manual se procede a explicar cómo hacer uso de la aplicación web de Pentaho.

14.2. Acceso a la aplicación

Acceda a la dirección web de la aplicación Pentaho proporcionada. Se mostrará la siguiente pantalla:



Figura 112: Pantalla de login de la aplicación web

Introduzca un usuario y contraseña válidos y pulse en el botón “Iniciar sesión”.

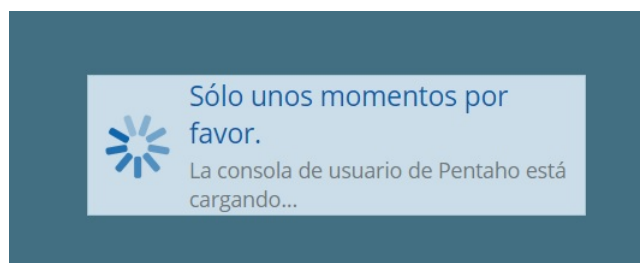


Figura 113: Mensaje de carga de la consola de usuario

Se iniciará sesión en la aplicación y se mostrará la consola de usuario.

14.3. Consola de usuario

La interfaz desde la perspectiva “Inicio” de la consola de usuario es la siguiente:



Figura 114: Interfaz consola de usuario

Las posibles opciones para realizar desde la consola de usuario son las siguientes:

1. Pestaña de navegación. Permite navegar entre las diferentes perspectivas de la aplicación.

Haciendo clic se despliega el menú de navegación:

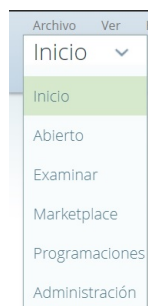


Figura 115: Pestaña de navegación entre perspectivas

- Inicio: Muestra la interfaz de inicio de la aplicación
 - Abierto: Muestra los recursos abiertos (consultas ad-hoc, visores olap, informes...)
 - Examinar: Se muestra la pantalla menú ficheros. Permite navegar entre los directorios de archivos de la aplicación.
 - Marketplace: Permite acceder al marketplace de Pentaho para instalar plugins desarrollados y publicados por la comunidad.
 - Programaciones: Se muestra la pantalla programaciones. Permite gestionar las tareas programadas en la aplicación.
 - Administración: Se muestra la pantalla administración. Permite administrar las opciones de la aplicación.
2. Muestra el nombre de usuario identificado en la aplicación. Tras pulsar sobre él se abre un menú desplegable que permite cerrar la sesión.
3. Menú de recursos de la aplicación:
- Ficheros. Se muestra la pantalla menú ficheros. Permite navegar entre los ficheros de la aplicación.
 - Crear nuevo. Permite crear una nueva consulta contra el Datawarehouse, cubo OLAP o crear un cuadro de mando. Tras situar el cursor sobre “Crear nuevo” se abre un desplegable mostrando las opciones posibles:



Figura 116: Desplegable Crear Nuevo

- Consultas: Se muestra la pantalla de consultas ad-hoc.
 - Visor OLAP: Se muestra la pantalla de consultas olap.
 - Tablero CDE: Permite crear un cuadro de mando.
 - Gestionar Fuentes de Datos: Permite gestionar las fuentes de datos con las que trabaja la aplicación.
 - Documentación: Descarga este manual.
4. Muestra los recursos abiertos recientemente.
5. Muestra los recursos marcados como favoritos.

14.4. Menú ficheros

Permite navegar entre las carpetas y ficheros alojados en el servidor, como consultas ad-hoc guardadas, cuadros de mando creados e informes predefinidos.

La interfaz del menú ficheros es la siguiente:

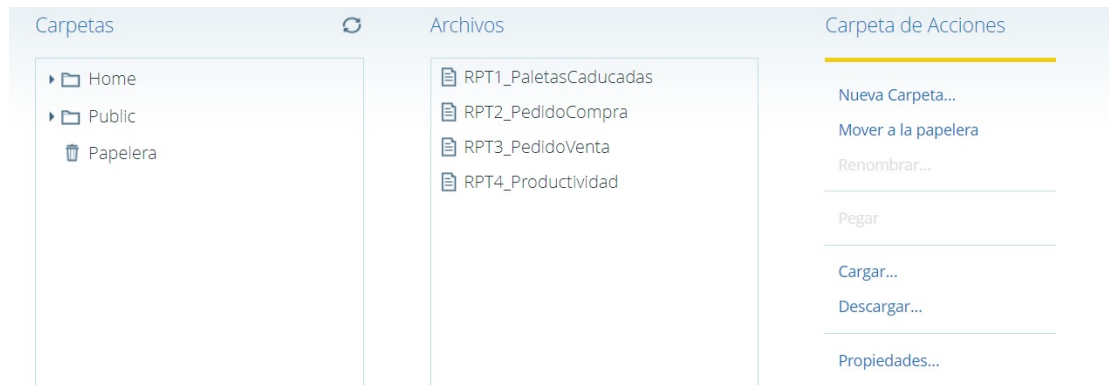


Figura 117: Interfaz menú ficheros con acciones sobre carpeta

Las acciones sobre una carpeta son las siguientes:

- Nueva Carpeta: Crea una nueva carpeta en la ubicación actual
- Mover a la papelera: envía la carpeta a la papelera
- Renombrar: Cambia el nombre de la carpeta
- Pegar: Pega un fichero cortado o copiado en la carpeta actual
- Cargar: Sube un fichero desde el equipo al servidor
- Descargar: Descarga la carpeta al equipo
- Propiedades: Consulta de propiedades de la carpeta

En caso de seleccionar un fichero, las posibles acciones son las siguientes:

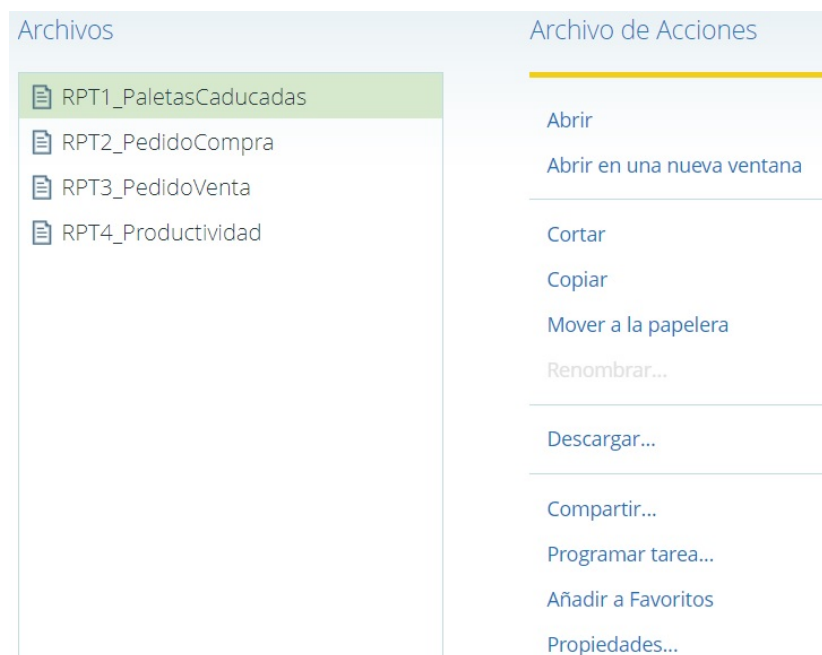


Figura 118: Interfaz menú ficheros con acciones sobre fichero

- Abrir: Abre el fichero seleccionado
- Abrir en una nueva ventana: Abre una nueva ventana con el fichero seleccionado
- Cortar: Traslada el fichero a otro lugar
- Copiar: Duplica el fichero seleccionado en otro lugar
- Mover a la papelera: envía el fichero a la papelera
- Renombrar: Cambia el nombre del fichero
- Descargar: Descarga el fichero seleccionado al equipo
- Compartir: Permite gestionar el acceso al fichero a otros usuarios
- Programar tarea: Permite programar una tarea a realizar con el fichero seleccionado
- Añadir a favoritos: Añade el fichero a la lista de favoritos
- Propiedades: Consulta de propiedades del fichero

14.5. Menú programaciones

Permite administrar las tareas programadas en la aplicación.

La interfaz del menú programaciones es la siguiente:

The screenshot shows a web interface titled 'Administrar programaciones'. It features a table with the following data:

Nombre	Repeticiones	Archivo fuente	Ubicación de salida	Última ejecución	Próxima ejecución	Creado por	Estado
PentahoSystemVersionCheck	cada día at 2:08:55 PM	PentahoSystemVersionCheck	-	Jun 5, 2021 2:08:55 PM	Jun 6, 2021 2:08:55 PM	admin	Normal

Below the table, there is a section titled 'Tiempo de interrupción' with the text 'No se han creado restricciones de fechas para la programación.' and a button labeled 'Crear tiempo de interrupción'.

Figura 119: Interfaz menú programaciones

Administrar programaciones: Muestra las programaciones en la aplicación, permitiendo editar y modificarlas.

Tiempo de interrupción: Muestra los tiempos de interrupción establecidos en la aplicación, durante el cual no se ejecutará ninguna programación.

Para crear un nuevo tiempo de interrupción, tras pulsar sobre el botón “Crear tiempo de interrupción”, se mostrará un asistente para configurarlo de una forma sencilla.

The screenshot shows a configuration popup titled 'Nueva programación de interrupción de tiempo'. It contains the following fields and options:

- Recurrencia: Solo una vez (dropdown)
- Inicio: 12:00 AM, Hora de verano de Europa Central (UTC+0200) (dropdown)
- Fin: Duración (selected), 0 día(s), 0 hora(s), 1 minuto(s)
- Fecha inicio: 6/6/21

At the bottom right, there are two buttons: 'Finalizar' and 'Cancelar'.

Figura 120: Popup asistente configuración de tiempo de interrupción

14.6. Menú administración

Permite administrar las diferentes opciones de administrador de la aplicación:

- Gestión de usuarios.
- Gestión de roles.
- Configuración del servidor de correo electrónico.

La interfaz del menú administración es la siguiente:



Figura 121: Interfaz opciones menú administración

Las opciones son las siguientes:

- Usuarios & Roles: Permite administrar los usuarios, los roles y permisos de usuarios.



Figura 122: Interfaz gestión usuarios y roles del menú administración

- **Servidor de correo:** Permite configurar un servidor de correo electrónico para enviar correos automáticamente.

Figura 123: Interfaz gestión servidor correo del menú administración

- **Configuración:** Permite eliminar archivos generados sin utilizar para optimizar el consumo de recursos de la aplicación, o configurar una programación para eliminarlos automáticamente en función de un periodo.

Figura 124: Interfaz configuración en menú administración

14.7. Consultas ad-hoc

Pantalla accesible desde Crear Nuevo / Consultas o Herramientas / Consultas.

Permite realizar consultas de una manera sencilla contra el Datawarehouse, siguiendo una filosofía drag & drop (arrastrar los elementos que forman las consultas) para formar las consultas de una forma visual.

La interfaz de la pantalla de consultas ad-hoc es la siguiente:

The screenshot shows a web interface for creating ad-hoc queries. At the top, there is a tab labeled 'Consultas' with a close button. To the right are two buttons: 'Exportar a Excel' and 'Guardar consulta'. Below this is a large rectangular area with the text 'Arrastra aquí los elementos'. Underneath are three dropdown menus: 'Seleccionar modelo de datos:' with 'INLOG.xmi - Pedidos de Compra' selected, 'Seleccionar categoría:' with 'PROPIETARIO' selected, and 'Seleccionar columna:'. Below the dropdowns is a grid of buttons representing available columns: 'CODIGO PROPIETARIO', 'RAZON SOCIAL PROPIETARIO', 'NOMBRE PROPIETARIO', 'DIRECCION PROPIETARIO', 'LOCALIDAD PROPIETARIO', 'PROVINCIA PROPIETARIO', 'CODIGO POSTAL PROPIETARIO', 'PAIS PROPIETARIO', 'PERSONA DE CONTACTO', 'TELEFONO PROPIETARIO', 'NUMERO DE FAX PROPIETARIO', and 'MUELLE'. At the bottom left, there is a label 'Filtros:'.

Figura 125: Interfaz pantalla consultas ad-hoc

Las opciones sobre la pantalla son las siguientes:

- Seleccionar modelo de datos: Muestra los diferentes modelos de metadatos creados.
- Seleccionar categoría: Muestra las tablas definidas en el modelo de metadatos seleccionado
- Seleccionar columna: Muestra las columnas de la tabla seleccionada
- Filtros: Muestra los filtros aplicados
- Exportar a Excel: Permite exportar la consulta actual a un fichero .xls
- Guardar consulta: Guarda la consulta realizada en un fichero .waqe para poder abrirla posteriormente desde el menú ficheros

14.7.1. Diseñar consulta

Para realizar una consulta, es necesario seleccionar el modelo de datos sobre el que se realizará la consulta, no siendo posible realizar consultas contra mas de un modelo de datos a la vez.

Es necesario ir arrastrando las columnas de las diferentes tablas que formarán la consulta a la pestaña “Arrastra aquí los elementos”.

The screenshot shows a query design interface. At the top, there are two tabs: 'CODIGO PROPIETARIO' and 'PEDIDO'. Below the tabs, there is a label 'Seleccionar modelo de datos:' followed by a dropdown menu containing 'INLOG.xmi - Pedidos de Compra'. Below that, there is a label 'Seleccionar categoría:' followed by a dropdown menu containing 'CABECERAS'.

Figura 126: Diseño de consultas ad-hoc

El resultado de la consulta realizada se mostrará en una tabla en la parte derecha de la pantalla:

The screenshot shows the query result interface. On the left, there are tabs for 'CODIGO PROPIETARIO' and 'PEDIDO'. Below the tabs, there are labels for 'Seleccionar modelo de datos:', 'Seleccionar categoría:', and 'Seleccionar columna:'. The 'Seleccionar columna:' label is followed by a grid of buttons representing different columns: TIPO PEDIDO, PROPIETARIO, PROVEEDOR, ALMACEN, AREA DESCARGA, TIPO ENTREGA, NUMERO PALETAS, TOTAL LINEAS, SITUACION PEDIDO, FECHA ASIGNADA, and AREA DESCARGA PLANIFICADA. On the right, there is a table with the following data:

CODIGO PROPIETARIO	PEDIDO
0	PRUJZA11
0	PRUJZA12
0	159
0	3007

Below the table, there is a label 'Showing 1 to 4 of 4 entries' and a pagination control with 'Previous', '1', and 'Next' buttons.

Figura 127: Resultado de consulta ad-hoc

Es posible modificar el número de registros mostrados por tabla, hacer búsquedas sobre la tabla y ordenar en función de un campo haciendo clic sobre la cabecera en la tabla de este.

14.7.2. Aplicar filtro

Para aplicar un filtro sobre un campo, es necesario hacer clic sobre este, abriéndose un popup para definir los filtros.

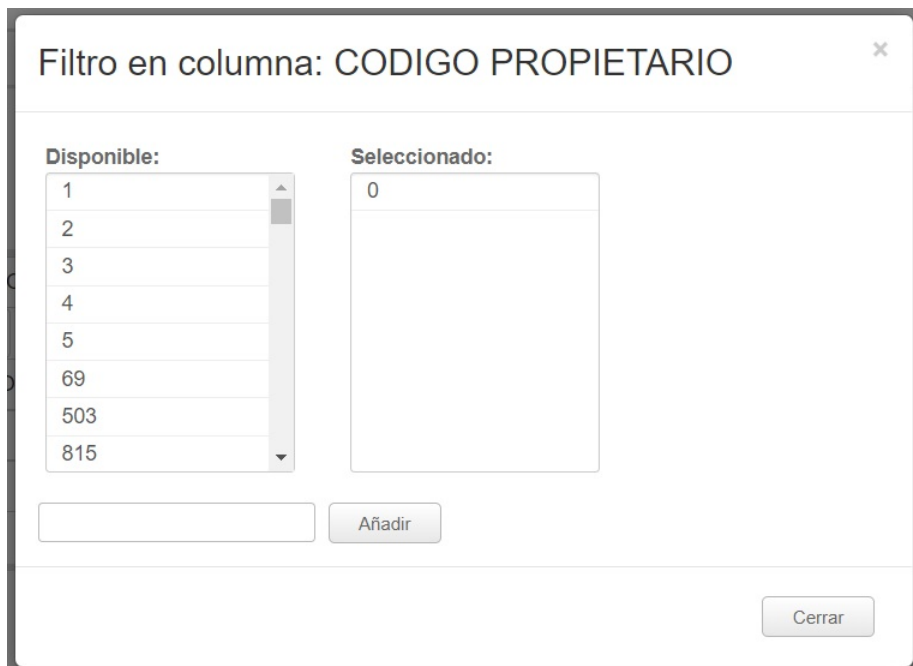


Figura 128: Popup selección de filtros sobre un campo

- Disponible: Muestra los valores posibles. Haciendo clic sobre un valor se aplicará un filtro sobre este.
- Seleccionado: Muestra los valores sobre los cuales se aplicará el filtro. Haciendo clic sobre un valor se dejará de aplicar un filtro sobre este.
- Es posible añadir un valor manualmente, escribiendo el valor en el cuadro de texto debajo de la columna “Disponible”. Una vez escrito el valor, pulsar el botón “Añadir” para aplicar el filtro sobre este valor.
- Cerrar: Cierra el popup de filtros.

Tras aplicar un filtro sobre algún campo, este aparecerá en el apartado filtros de la pantalla consultas ad-hoc. Los filtros se mostrarán de la forma CÓDIGO CAMPO: VALORES:

Filtros: CODIGO PROPIETARIO: 0

Figura 129: Filtros aplicados en una consulta ad-hoc

Es posible eliminar un filtro haciendo clic sobre el valor a eliminar.

14.8. Consultas OLAP

Pantalla accesible desde Crear Nuevo / Visor OLAP o Archivo / Nuevo / Consulta OLAP.

Tras acceder al visor OLAP se muestra la siguiente pantalla:



Figura 130: Pantalla selección de cubo OLAP

Seleccionar catalogo y cubo contra el que se realizarán las consultas y pulsar el botón “Aprobar”.

Tras pulsar el botón, se muestra la pantalla para realizar la consulta contra el cubo seleccionado:



Figura 131: Pantalla visor OLAP

Tenemos las siguientes pestañas:

- Navegador OLAP: Muestra los cubos contra los que realizar una consulta.
- Estructura de cubo: Muestra la estructura (medida y dimensiones) del cubo seleccionado.
- Estructura pivote: Muestra la estructura de la consulta a realizar.
- Resultado de la consulta: Muestra la consulta realizada.
- Consulta MDX: Muestra la sentencia MDX realizada para realizar la consulta OLAP.

Para realizar una consulta contra el cubo seleccionado es necesario mostrar al menos una medida y dimensión, y al menos mostrar un elemento del cubo en filas y columnas. Para mostrar un elemento del cubo, seleccionar el elemento desde “Estructura de cubo” y arrastrar a Columnas o Filas de “Estructura pivote”.

The screenshot displays an OLAP viewer interface with the following components:

- Navegador OLAP:** Shows the selected cube as "Cabecera pedido de compra".
- Estructura de cubo:** Lists dimensions: Trimestre, Mes, Semana, Dia, Dia Juliano, and Festivo.
- Estructura pivote:** Shows the pivot structure with "Numero pedidos" in the Columns area and "Año" and "Mes" in the Rows area.
- Resultado de la consulta:** Displays a table with the following data:

Tiempo	Measures
2021	Numero pedidos: 170
Enero	18
Febrero	87
Marzo	23
Abril	2
Junio	
Mayo	40
Julio	
- Consulta MDX:** Shows the MDX query:


```
1 SELECT {[Measures].[Numero pedidos]} ON COLUMNS, Hierarchize({[DIM_TIEMPO.Tiempo].[2021], [DIM_TIEMPO.Tiempo].[2021].[1].[1].[Enero], [DIM_TIEMPO.Tiempo].[2021].[1].[1].[Febrero], [DIM_TIEMPO.Tiempo].[2021].[1].[1].[Marzo], [DIM_TIEMPO.Tiempo].[2021].[1].[2].[Abril], [DIM_TIEMPO.Tiempo].[2021].[1].[2].[Junio], [DIM_TIEMPO.Tiempo].[2021].[1].[2].[Mayo], [DIM_TIEMPO.Tiempo].[2021].[2].[3].[Julio]}) ON ROWS FROM [Cabecera pedido de compra]
```

Figura 132: Ejemplo de consulta con el visor OLAP

14.9. Informes predefinidos

Para abrir un informe predefinido, es necesario abrir un fichero .prpt desplegado en el servidor desde el menú ficheros.

Una vez abierto el fichero, se muestra la pestaña para configurar los parámetros del informe:

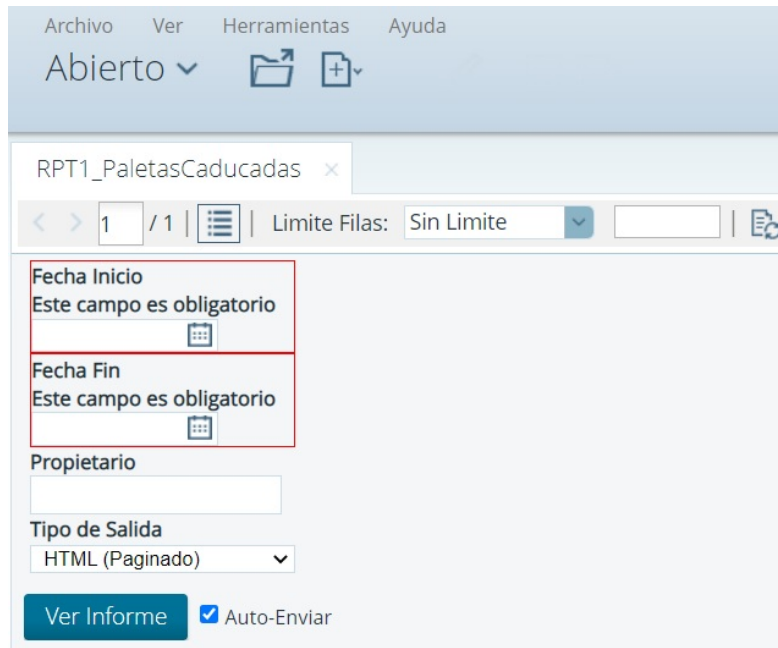


Figura 133: Configuración parámetros de informe predefinido

Se muestran:

- Campos obligatorios para poder lanzar un informe.
- Campos opcionales.
- Tipo de Salida: Indica el formato en el que se mostrará el informe (HTML, Excel, PDF...). En caso de seleccionar la opción HTML se mostrará en la misma página debajo de la pestaña de configuración de informe.

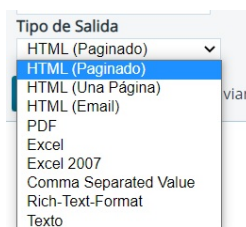


Figura 134: Valores tipo de salida de informe predefinido

Una vez seleccionados los parámetros, pulsar el botón “Ver informe” para lanzar el informe.

En el caso de seleccionar la opción “Auto-Enviar”, al cambiar algún parámetro, se lanzará el informe con las nuevas opciones automáticamente

Fecha Inicio
03/05/2021

Fecha Fin
04/06/2021

Propietario

Tipo de Salida
HTML (Paginado)

Auto-Enviar

08, junio, 2021

INFORME PALETAS CADUCADAS

FECHA INICIO: 03/05/2021
FECHA FIN: 04/06/2021

PALETA	PROPIETARIO	ARTICULO	NOMBRE ARTICULO	STOCK	IMPORTE
389837	XABIER	476	Aigua EVIAN PET -Amp. 0,33 L	1776	8.880,00 €
459596	ANDOLAC	81118	Caldo POLLASTRE DOY 0,5L - GB	156	780,00 €
448276	ANDOLAC	83203	CONFITURA PRUNA 256x25GRS HELIOS	27	135,00 €
469068	ANDOLAC	80489	HELIOS Mel Pot Vidre Mini 24x 30 Gr	29	145,00 €

Figura 135: Ejemplo de informe predefinido