Universitat Oberta de Catalunya

uoc.edu

Sistema de inteligencia de negocio: análisis de la publicidad en entornos digitales

María Lorena Talamé
Máster Universitario en Ingeniería Informática
Business Intelligence

Consultor: David Amorós Alcaraz
Profesora: María Isabel Guitart Hormigo

11/06/2018

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Sistema de inteligencia de negocio análisis de la publicidad en entornos digitales*
Nombre del autor:	*María Lorena Talamé Otero*
Nombre del consultor/a:	*David Amorós Alcaraz*
Nombre del PRA:	*María Isabel Guitart Hormigo*
Fecha de entrega (mm/aaaa):	06/2018
Titulación::	*Máster Universitario en Ingeniería Informática*
Área del Trabajo Final:	*Business Intelligence*
Idioma del trabajo:	*Castellano*
Palabras clave	*Business Intelligence Publicidad OLAP Multidimensional Red Social Datawarehouse*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El presente trabajo tiene como finalidad diseñar e implementar un sistema de Business Intelligence que facilite la adquisición, el almacenamiento y la explotación de los datos obtenidos durante la publicación de anuncios publicitarios en plataforma digitales. La metodología seleccionada fue la de Kimball y la suite de herramientas de BI, Pentaho.

Como resultado del TFM se obtuvieron distintos archivos relacionados con la explotación de los datos (ficheros ETL, cubos en XML y consultas en Saiku).

Se obtuvieron interesantes conclusiones sobre la eficiencia de la paramterización de los anuncios publicitarios destacando zonas geográficas, intereses, rango de edad y género de los grupos de usuarios.

Desde el punto de vista personal, el mayor fruto del presente trabajo, llevado a cabo con tanta satisfacción, fue la profundización sobre esta temática tan prometedora para el futuro que seguramente redundará en beneficios y oportunidades profesionales.

Abstract (in English, 250 words or less):

The purpose of this work is to design and implement a Business Intelligence system that facilitates the acquisition, storage and exploitation of data obtained during the publication of digital platform advertising. The selected methodology was Kimball method and the BI suite, Pentaho.

As a result of the TFM, different files related to the exploitation of the data were obtained (ETL files, XML cubes and Saiku queries).

Interesting conclusions were arrived on the efficiency of the parametrization of advertisements highlighting geographical areas, interests, age range and gender of user groups.

From a personal point of view, the greatest fruit of this work, carried out with such satisfaction, was the deepening on this subject so promising for the future that will surely result in benefits and professional opportunities.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

Las personas están cambiando la manera de relacionarse gracias al uso de las tecnologías. De un modo particular, las redes sociales están ocupando un rol primordial en la vida cotidiana de la gente. Por lo tanto, las empresas deben adaptarse a estos tiempos lo que necesariamente se traduce en nuevas formas de mercadeo. Actualmente, se están utilizando las redes sociales para realizar campañas publicitarias y así crear la fidelización del cliente o consumidor con el producto y/o marca.
El éxito de estas acciones se mide a través de indicadores que, al utilizar medios digitales, se obtienen de una manera más sencilla y en tiempo real. Además, en *social media*, la rapidez en adaptar la publicidad a una audiencia puede ser la diferencia entre éxito y fracaso. Con el presente trabajo, se pretende implementar una solución BI para analizar la eficiencia en la parametrización de anuncios publicitarios y ayudar en la elección de los mejores parámetros para lograr un mayor éxito en las campañas publicitarias.

1.2 Objetivos del Trabajo

El objetivo de este trabajo es diseñar e implementar un sistema de *Business Intelligence* que facilite la adquisición, el almacenamiento y la explotación de los datos obtenidos durante la publicación de anuncios publicitarios en plataforma digitales como Instagram, Twitter, Facebook o Youtube.
Los objetivos específicos del trabajo son:

- Diseñar e implementar un almacén de datos (*Data Warehouse*) que permita almacenar la información adquirida en los diferentes orígenes.
- Programar los procesos ETL (extracción, transformación y carga) que permitan alimentar el DWH a partir de los ficheros base facilitados.
- Analizar las diferentes plataformas BI *Open Source* disponibles en el mercado que permitan explorar la información almacenada.
- Elegir una de estas herramientas de tal forma que se disponga de una capa de software para el análisis de la información.
- Proveer una aplicación BI para analizar la eficiencia de la parametrización de anuncios publicitarios.

Con la aplicación BI se busca responder a preguntas como:

- ¿Qué regiones o ciudades tienen mejores indicadores de efectividad? ¿Hay alguna relación con el producto o familia de productos?

- ¿Existen una relación entre la mejora de los indicadores de efectividad con algún segmento de la población objetivo?
- ¿Hay alguna plataforma donde, bajo las mismas condiciones, se obtengan mejores tasas de visualización?
- ¿Existen relaciones entre plataformas y franjas de edad de usuarios que provoquen mejores tasas de visualización?
- ¿El conocimiento del grupo de interés de los usuarios podría ayudar a mejorar los indicadores para determinados productos?

1.3 Enfoque y método

A la hora de encarar un proyecto de BI es necesario definir si se hará siguiendo alguna metodología o, solamente, se realizará según el criterio del experto BI. La ventaja de seguir una metodología probada es que sirve como guía y sus buenas prácticas ayudan a la implementación del sistema haciendo la tarea más sencilla. El presente trabajo se llevará a cabo siguiendo el ***enfoque Kimball***.

Si bien los conceptos originales surgieron alrededor de 1980 y la primera edición del libro "*The Data Warehouse Lifecycle Toolkit*" en 1996, el ciclo de vida Kimball todavía se encuentra vigente para este tipo de proyecto. Además, las herramientas existentes en el mercado permiten aplicarla sin necesidad de grandes adaptaciones.

Esta metodología se centra en lo que Kimball denomina *"Ciclo de Vida Dimensional del Negocio"* [4] que básicamente consiste en analizar varias dimensiones en vez de seguir la 3º forma normal. Su ciclo de vida se ilustra en la Figura 1:

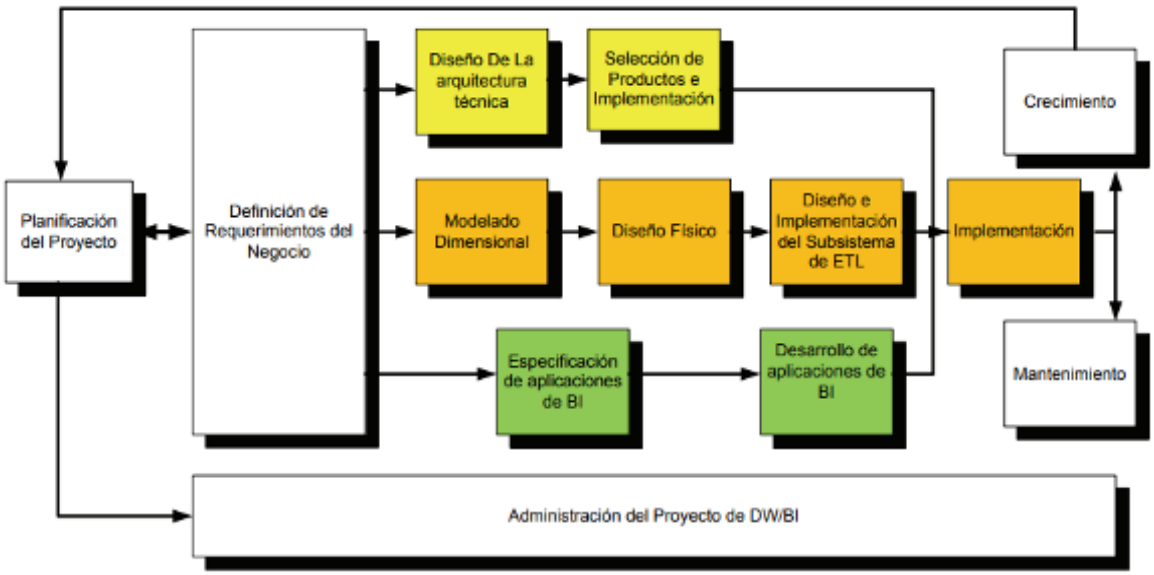


Figura 1: Ciclo de vida Kimball

(Fuente: http://www1.ucasal.edu.ar/htm/ingenieria/cuadernos/archivos/5-p56-rivadera-formateado.pdf)

Este diagrama sirve como guía para una implementación exitosa del sistema BI. Como en todo proyecto, la fase inicial es su planificación. Luego de la definición de requerimientos del negocio, se diferencian tres rutas:

1) *Ruta de la tecnología*: se diseña y seleccionan la arquitectura técnica y herramientas a utilizar.

2) *Ruta de datos*: se parte del diseño dimensional, se realizan los procesos ETL para su posterior implementación.
3) *Ruta de inteligencia de negocios*: se arman los cuadros de mandos, tableros y aplicaciones.

Las rutas convergen en una etapa de mantenimiento y crecimiento del sistema BI.

Otro punto importante a definir es la elección del modo de realización: si implementar una aplicación *ad hoc*, o hacer uso de alguna herramienta disponible en el mercado. Existen múltiples herramientas, inclusive *Open Source,* potentes y bien aceptadas por la comunidad. Estas plataformas dan soporte a cada etapa del proceso lo que facilitaría el trabajo y permitiría cumplir con los plazos establecidos de cada entrega. Por estos motivos, se utilizará alguna de ellas descartando la primera opción.

1.4 Planificación del Trabajo

El trabajo se va a realizar en diez etapas que, en algunos casos, se subdividirán para una mejor realización. Además, por motivos laborales, se modificó el calendario de los días lunes a sólo media jornada. Así mismo, si no fuera posible cumplir con la planificación entre semana, se deja abierta la posibilidad de recuperar el trabajo los fines de semana o los días no laborables.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesor	Nombres de los recursos
1	**Sistema de BI**	**70 días**	**mié 21/02/18**	**lun 11/06/18**		
2	**1. Planificación del Proyecto**	**70 días**	**mié 21/02/18**	**lun 11/06/18**		
3	1.1 Recopilación de documentación y bibliografía	4 días	mié 21/02/18	lun 26/02/18		
4	1.2 Lectura de la documentación y bibliografía	10 días	mié 21/02/18	mié 07/03/18		
5	1.3 Planificación del proyecto	2 días	jue 01/03/18	vie 02/03/18		
6	1.4 Entrega PEC 1: Planificación	0 días	lun 05/03/18	lun 05/03/18		
7	**2. Estudio y Selección de Herramientas BI**	**6 días**	**mié 07/03/18**	**jue 15/03/18**		
8	2.1 Investigación de Herramientas	5 días	mié 07/03/18	mié 14/03/18	4	
9	2.2 Selección de la herramienta	1 día	jue 15/03/18	jue 15/03/18	8	
10	**3. Instalación del entorno de trabajo**	**9 días**	**vie 16/03/18**	**jue 29/03/18**	**9**	
11	3.1 Instalación del Servidor	3 días	vie 16/03/18	mié 21/03/18		
12	3.2 Instalación del Software BI	2 días	mié 21/03/18	vie 23/03/18	11	
13	3.3 Aprendizaje del Software	4 días	vie 23/03/18	jue 29/03/18	12	
14	**4. Definición de Requerimientos del Negocio**	**4.5 días**	**mié 07/03/18**	**mar 13/03/18**		
15	4.1 Estudio de la documentación seleccionada	4 días	mié 07/03/18	mar 13/03/18	4	
16	4.2 Análisis de ficheros entregados	3 días	mié 07/03/18	vie 09/03/18		**Ficheros con datos de entrada**
17	**5. Modelado Dimensional**	**15 días**	**lun 12/03/18**	**mar 03/04/18**		
18	5.1 Diseño del modelo	10 días	lun 12/03/18	mar 27/03/18	16	
19	5.2 Revisión y Validación del modelo	3 días	jue 29/03/18	mar 03/04/18	18	
20	5.3 Entrega PEC 2: Memoria y Diseño del DW	0 días	lun 09/04/18	lun 09/04/18		
21	**6. Implementación del DW**	3 días	mié 04/04/18	vie 06/04/18	17,12	
22	**7. Diseño e Implementación de los procesos ETL**	**18 días**	**mar 03/04/18**	**mié 02/05/18**		
23	7.1 Diseño de los procesos ETL	9 días	mar 03/04/18	mar 17/04/18	19	
24	7.2 Implementación de procesos ETL	7 días	mar 17/04/18	jue 26/04/18	23	
25	7.3 Verificación de los procesos	2 días	vie 27/04/18	mié 02/05/18	24	
26	**8. Implantación de Aplicación BI**	**17 días**	**mié 02/05/18**	**mar 29/05/18**		
27	8.1 Implantación de Aplicación BI	17 días	mié 02/05/18	mar 29/05/18	22	
28	8.1 Entrega PEC3: Estado del TFM	0 días	lun 07/05/18	lun 07/05/18		
29	**9. Análisis de Resultados y Conclusiones**	**4 días**	**mar 29/05/18**	**lun 04/06/18**	**26**	
30	**10. Entrega de TFM**	**4 días**	**mar 05/06/18**	**lun 11/06/18**	**29**	
31	10.1 Preparación de Entrega del Producto	3 días	mar 05/06/18	jue 07/06/18		
32	10.2 Preparación de la Presentación	4 días	mar 05/06/18	vie 08/06/18		
33	10.3 Autoevaluación	1 día	vie 08/06/18	vie 08/06/18		
34	10.4 Entrega del TFM	0 días	lun 11/06/18	lun 11/06/18		
35	**Documentación en Memoria**	**68 días**	**mié 21/02/18**	**mié 06/06/18**		

Diagrama de Gantt

Diagrama de Gantt

Figura 2: Planificación del proyecto

La disponibilidad de los ficheros con los datos de entrada puede llegar a condicionar el plan del proyecto por lo que la tarea “4.2 Análisis de ficheros entregados” se considera crítica y se prevé su inicio el día 07 de Marzo.

En rojo se destacan las entregas según el cronograma indicado para cada PEC. En la *PEC1*, prevista para el día ***05/03/2018***, se entregará una ***primera versión del documento TFM*** con su respectiva introducción y explicación del proyecto. Asimismo, el enfoque y la metodología elegida, la ***planificación*** detallada y su correspondiente diagrama de Gantt.

La *segunda PEC* del día ***09/04/2018*** incluirá el ***diseño del data warehouse*** y una nueva versión del ***documento TFM***. Se habrá instalado el entorno de trabajo y elegida la herramienta BI. Se espera tener finalizadas desde la etapa 1 a la 5 para poder comenzar con la sexta: la implementación del DW.

El día ***07/05/2018*** está prevista la penúltima entrega del ***documento TFM*** junto con el ***diseño de los procesos ETL*** y del ***avance*** de la aplicación BI. Se espera terminar con las fases 6 y 7.

Finalmente, la última entrega consistirá en el ***TFM***, la ***presentación*** y ***autoevaluación*** prevista para el día *11/06/2018*.

En el Anexo 1 se puede ver el Diagrama de Gantt.

1.4.1 Costo estimado del proyecto

La valoración monetaria del presente proyecto se realiza teniendo en cuenta el costo de la jornada laboral según la escala salarial del convenio colectivo de trabajo correspondiente a la Unión Informática de la Confederación General del Trabajo de la República Argentina [1]. Se elige la categoría de Consultor *Business Intelligence Semi Senior* que según la Cámara de la Industria Argentina del Software corresponde a un profesional con experiencia e independencia para abordar problemas; puede descomponer problemas, buscar posibles soluciones y tiene idea del conjunto del proyecto [2]. Este profesional mensualmente debería cobrar como mínimo $ 30.868,75, a razón de $ 1.028,96 por día (equivalente a 1171,25 €/mes o 39,27€/día). Si este valor se multiplica por la cantidad de días que dura el proyecto, el costo estimado del mismo es de $ 72.027,20 o su equivalente en euros, 2.749,13 €.

1.5 Breve sumario de productos obtenidos

El principal producto que se obtendrá al finalizar el trabajo es la aplicación de BI que permitirá valorar los parámetros elegidos respecto a una campaña publicitaria en redes sociales. Además, como se indicó en el apartado de planificación, se obtendrán subproductos: la creación de un entorno de trabajo consistente en la herramienta seleccionada, la implementación del *data warehouse* y procesos ETL, la realización de cubos OLAP y consultas MDX.

1.6 Breve descripción de los otros capítulos de la memoria

En el primer capítulo, se introduce el problema planteado y sus objetivos.
En el segundo tendrá lugar el análisis de las herramientas *open source* y los criterios de selección de la plataforma a utilizar.
El modelo dimensional junto con sus pertinentes explicaciones se detalla en el tercer capítulo de esta memoria.
Los procesos ETL, su diseño e implementación son el objeto del cuarto capítulo.
En el quinto capítulo se plasma la creación del cubo OLAP junto con el análisis realizado de la información.
El sexto capítulo contiene las conclusiones del TFM.
El glosario y la bibliografía se encuentran en el séptimo y octavo capítulo, respectivamente.
Por último, se presentan todos los anexos de este trabajo.

2. Análisis y selección de herramientas BI

2.1 Análisis de herramientas *open source* para BI

Debido a la gran cantidad de herramientas BI que se pueden encontrar en el mercado, se toma como punto de partida aquellas propuestas por el tutor.

Las características a analizar en cada caso son:

- Soporte para todas las etapas del ciclo de vida (DWH, procesos ETL, análisis multidimensional, minería de datos, tableros o similar, diseño de informes).
- Gestión de múltiples orígenes de datos
- Versión gratuita
- Requisitos para su instalación

Pentaho Community Edition 8.0

Si bien **Pentaho** ahora está más orientado a lo comercial, todavía sigue proporcionando una versión *open source*, denominada *Community Edition*. Actualmente, se encuentra en la versión 8.0 y es multiplataforma.

Posee un enfoque simplificado e interactivo que permite a los usuarios acceder, explorar y combinar cualquier tipo de datos, independientemente de su tamaño. Proporciona potentes herramientas para las distintas fases. Entre ellas, se destaca *Pentaho Data Integration* (PDI) y *Pentaho Business Analytics*. PDI permite la obtención de datos de diversas fuentes y facilita la gestión de procesos ETL a través de una interfaz gráfica y un alto rendimiento. Además, provee una capa de abstracción para integrar minería de datos y soporte para big data.

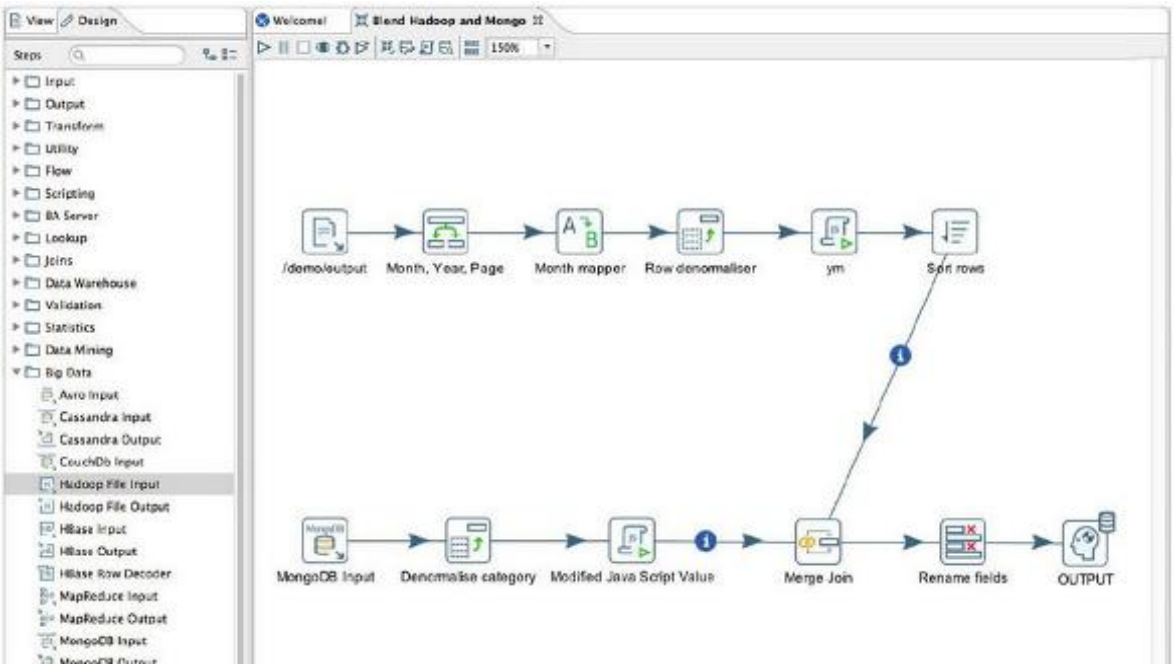


Figura 3: Spoon - Pentaho Data Integration

Pentaho Business Analytics es una herramienta que permite analizar la información con diferentes tipos de visualizaciones interactivas, generar informes y realizar análisis predictivo.

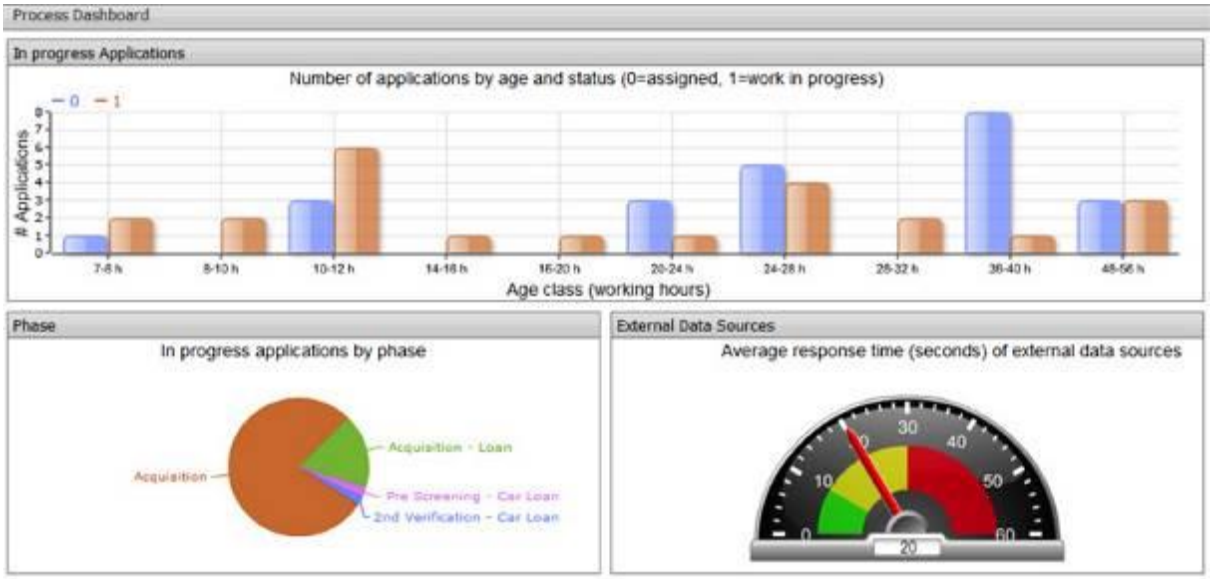


Figura 4: Pentaho Dashboard

Por defecto, utiliza PostgresSQL; pero se puede usar sobre MySQL, Oracle y MS SQL Server, entre otros. También soporta base de datos NoSQL como MongoDB.
Tanto a los reportes como a los análisis se puede acceder desde una pc de escritorio, una laptop o un móvil.
En cuanto a los requisitos de instalación para el servidor se necesita un procesador de 64bits, 8 GB de RAM y un espacio de 20 GB. Por otro lado, para el cliente, se requiere un procesador de 64 bits, 2GB de RAM y la misma cantidad de espacio libre en el disco rígido.

Ventajas:

- Está ampliamente aceptado por la comunidad.
- Según distintos portales, es una de las mejores alternativas *Open Source* del mercado.
- Su curva de aprendizaje no es tan pronunciada como en otros casos, ya que proporciona herramientas gráficas e intuitivas.
- Provee soporte para todas las fases.
- Su versión *Community* no tiene tiempo límite de uso.
- PDI es ampliamente aceptado por la comunidad y es muy potente.

Desventajas:

- La sección de descarga y la documentación de la edición *Community* no es simple de encontrar.
- Su versión por suscripción ofrece diseños interactivos de informes y otras mejoras que en la opción *Community* no están disponibles.

BIRT

Según [12], **Birt** es un *software open source* que permite crear informes. Es una solución muy adoptada en la comunidad para la visualización de

datos. Tiene dos componentes principales: un diseñador y un generador de informes que permite incrustarlos en aplicaciones del cliente, especialmente aquellas basadas en Java y Java EE. También incluye un motor de gráficos integrado en el diseñador, que incluso puede ser utilizado por separado.

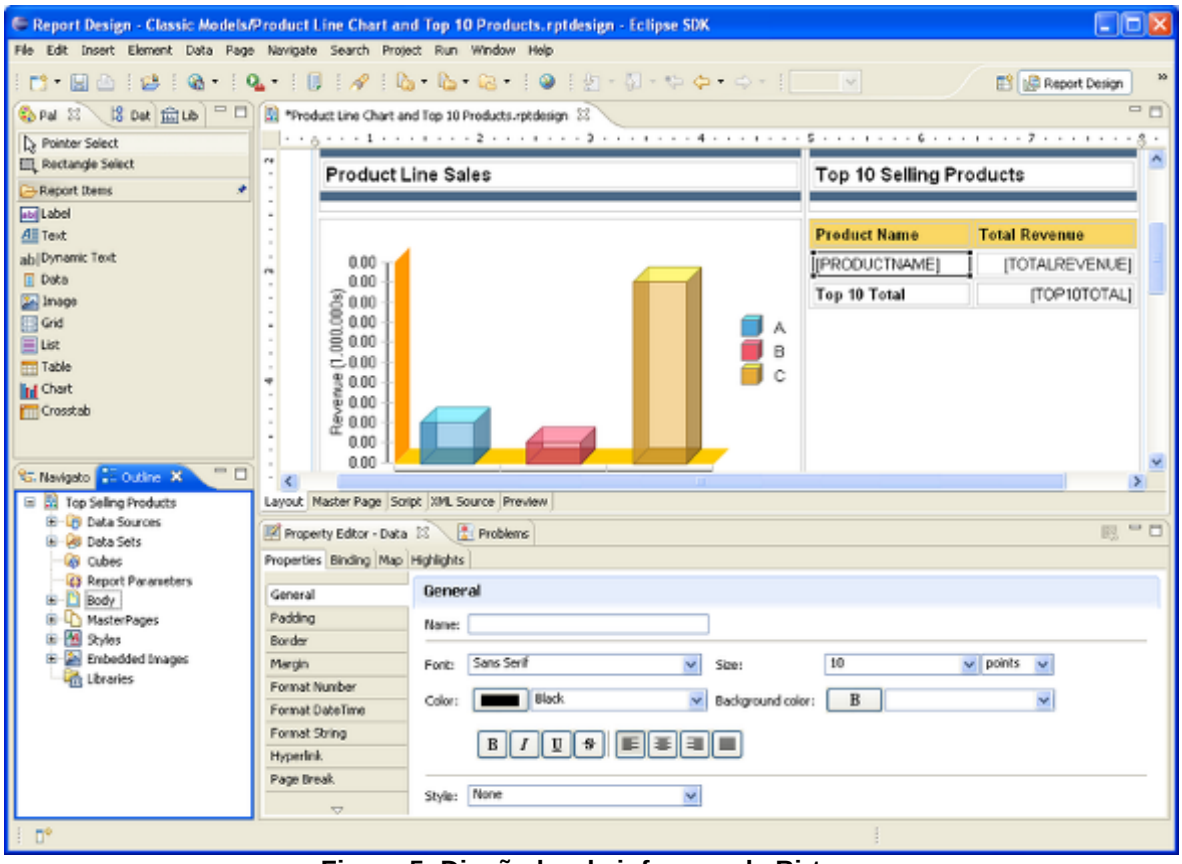


Figura 5: Diseñador de informes de Birt

No posee un módulo para ETL ni para otros procesos.
Los requisitos de instalación son mínimos ya que es un *plugin* para Eclipse 3.0.1 o superior.

Ventajas:

- Estándar de facto para la visualización de datos.
- Integración con Eclipse y otras herramientas.

Desventajas:

- No posee soporte para otras fases.

SpagoBI

La nueva versión de **Spago** es la Knowage 6.1. Es un software multiplataforma. Se autoproclama en [19] como la única plataforma cien por ciento *Open Source* del mercado. Como la mayoría de las plataformas analizadas, tiene una arquitectura modular donde cada módulo se encarga específicamente de un área. Entre ellas se destacan Big Data, análisis predictivo y generación de reportes. Además, posee un componente para enlazar la ubicación geoespacial con los datos

procesados. Por lo tanto, soporta todas las áreas de proyectos BI incluida la gestión de datos, administración y seguridad. Permite exportar reportes en diversos formatos y realizar análisis multidimensionales de la información. Ofrece soluciones para tareas de *Data mining*.

Figura 6: Dashboard de Knowage

Knowage se vale de TOS (Talend Open Studio) para la extracción, modificación y carga de datos en el data warehouse.

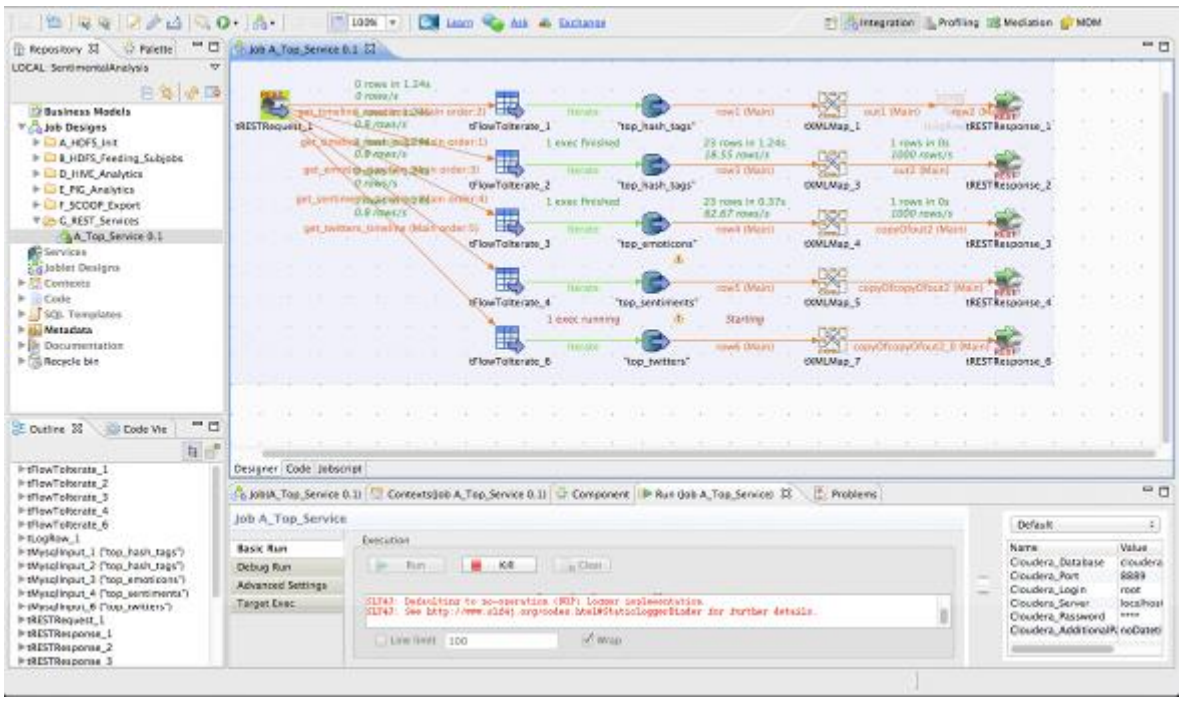


Figura 7: ETL con TOS

Knowage también ayuda al administrador del proyecto a controlar las versiones y a gestionar los flujos de trabajos, a fin de lograr un equipo de trabajo colaborativo.
Requiere para su instalación JDK 1.8, 2GB de Java *heap size* y también 2GB de espacio libre en el disco del lado del servidor. Del lado del cliente, sólo necesita un navegador que soporte Javascript.

Ventajas:

- Posee foros y documentación para iniciarse en su uso.
- Según distintos portales, es una de las mejores alternativas *Open Source* del mercado.
- Provee soporte para todas las fases.

Desventajas:

- Curva de aprendizaje pronunciada

Tibco Jaspersoft

Tibco Jaspersoft es una arquitectura flexible para soluciones de BI donde permite seleccionar distintas herramientas según el objetivo.
La herramienta para la integración de los datos se denomina ETL Essentials y posibilita de manera gráfica, al igual que Spoon (Pentaho), gestionar estos procesos de distintas fuentes, incluso desde la nube. Soporta base de datos relacionales y NoSQL. Tiene conexiones nativas a algunas aplicaciones ERP y CRM, como Salesforce.com, SAP y SugarCRM.

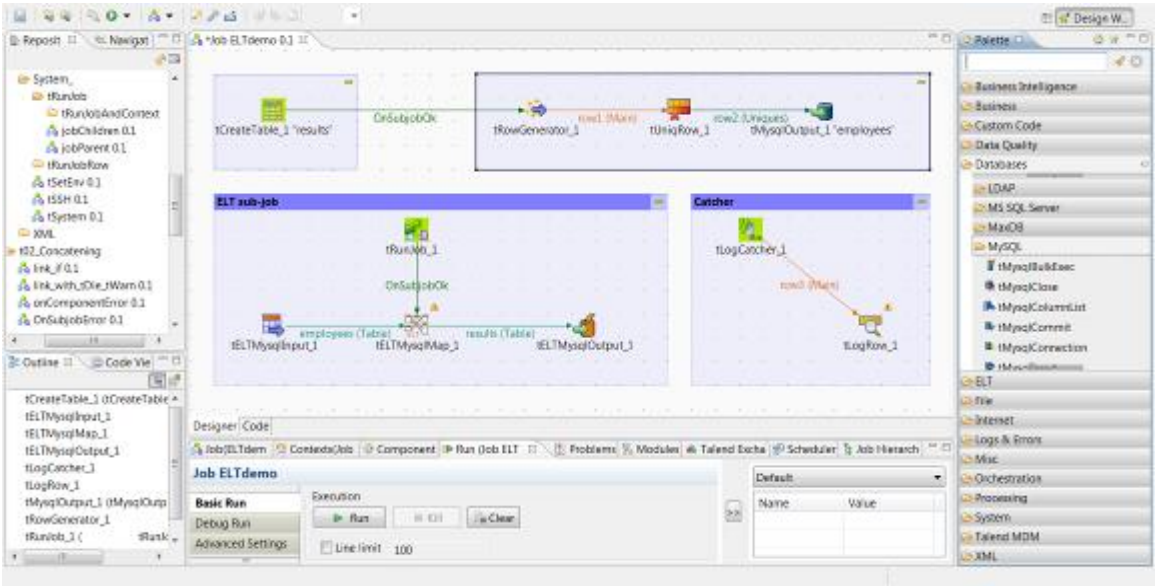


Figura 8: ETL Essentials

A pesar de que Tibco Jaspersoft posee este módulo para la integración de los datos, pone su énfasis en la visualización y creación de reportes o informes. Para este fin, varios módulos se encargan de mostrar los datos de forma interactiva, mediante análisis OLAP o cuadros de mandos. En la versión gratuita no se pueden crear dashboards.

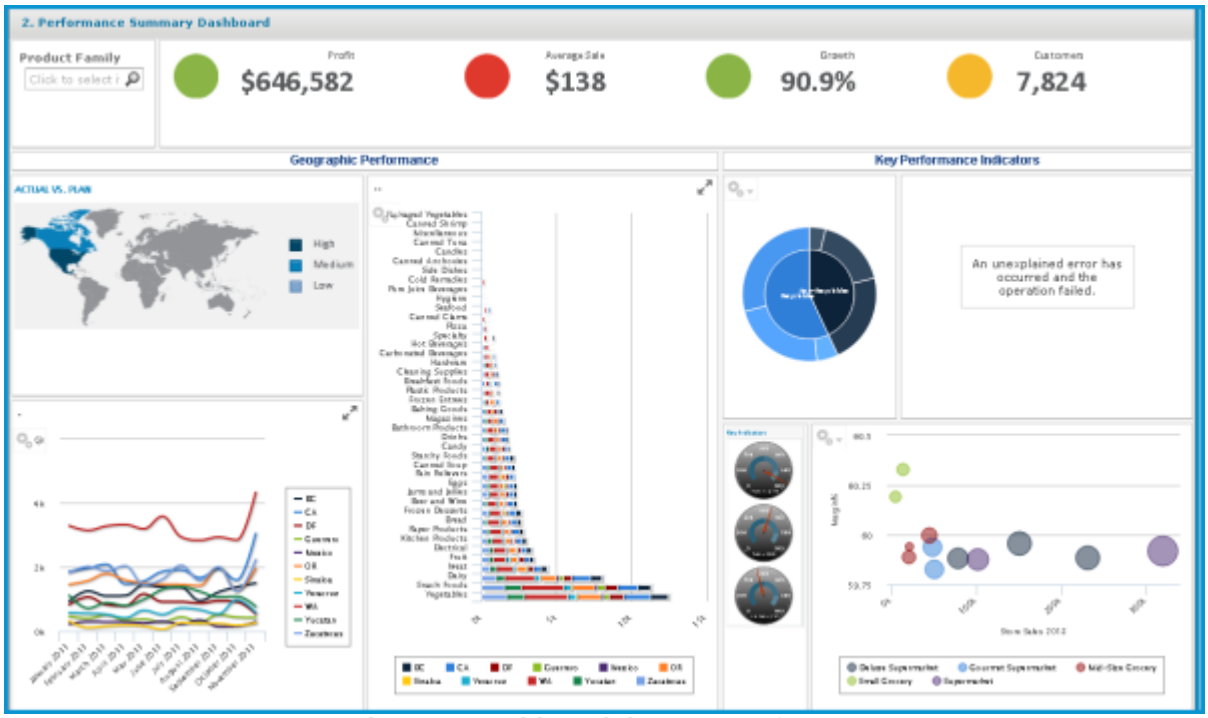


Figura 9: Dashboard de Jaspersoft

Jaspersoft es multiplataforma e incluye una aplicación nativa para dispositivos iPhone y Android. Con la versión paga, los informes se pueden incrustar en aplicaciones o en páginas web mediante Visualize.js. También, permite utilizar AWS (Amazon Web Services) como un servicio cloud con una suscripción de pago por hora de uso.

Ventajas:

- Posee soporte para la mayoría de las fases.
- Extensa comunidad y documentación

Desventajas:

- Integración con grandes almacenes de datas sólo en versiones comerciales.
- No posee soporte para minería de datos ni dashboards en la versión *Community*.

A modo de resumen, se muestra una tabla con las principales características analizadas en la versión gratuita de cada plataforma:

	Pentaho	Birt	Knowage 6.1	Jaspersoft
Versión gratuita	✓	✓	✓	✓
Gestión de múltiples fuentes de datos	✓	✓	✓	✓
Herramienta ETL	✓	✗	✓	✓

integrada				
Creación de informes	✓	✓	✓	✓
Informes interactivos y cuadros de mandos	✓	✓	✓	✗
Análisis OLAP	✓	✗	✓	✓
Soporte para Minería de datos	✓	✗	✓	✗

2.2 Herramienta Seleccionada: Pentaho

Analizando las características de las distintas opciones, en un principio se descartan aquellas que no poseen soporte para todas las fases, quedando SpagoBI y Pentaho como entornos posibles. Los criterios de selección entre ambas son los siguientes:

- Facilidad en la instalación
- Facilidad de uso y curva de aprendizaje
- Comunidad de soporte y documentación

Facilidad en la instalación:
La instalación de Pentaho sólo es necesario descargar y descomprimir los distintos componentes.
Para la instalación del servidor de Knowage, solamente se requiere descargar el instalador y seguir sus instrucciones. No ocurre lo mismo con el resto de los componentes.

Facilidad de uso y curva de aprendizaje:
Cabe destacar que la interfaz de Pentaho es más intuitiva que la de Knowage. Una ventaja respecto a Pentaho es que permite seleccionar el idioma español.

Comunidad de soporte y documentación:
Ambas herramientas tienen acceso a foros y blogs de soporte en sus páginas oficiales [9] y [19].
En la web, y en particular en Youtube, hay muchos videos y webinars para Knowage. Pero Pentaho tiene una gran comunidad que aporta constantemente ejemplos y documentación. Otra ventaja es que Pentaho es una suite más consolidada y tiene una herramienta ETL muy popular y de gran reputación.

La selección final se realizó una vez instaladas y probadas ambas plataformas. Por lo expresado hasta aquí, y sobre todo por la facilidad en el uso, se eligió Pentaho.

2.3 Entorno de Trabajo

Al contar con una PC cuyo sistema operativo es Windows 10 y continuando con el uso de software *Open Source*, se decide armar el

entorno de trabajo sobre una máquina virtual de Linux. La versión instalada es Ubuntu 16.04 LTS. El equipo anfitrión posee 12 GB de RAM por lo que se destinó a la máquina virtual 8 GB. Esta capacidad permite ejecutar sin problemas Pentaho.
En el anexo 2 se muestra la forma de instalación de Pentaho CE Edition.

3. Modelo dimensional

El modelado dimensional es una técnica ampliamente aceptada para presentar información analítica permitiendo rápidas consultas a la hora de entender el negocio. Provee la misma información que un modelo normalizado (por ejemplo, base de datos relacional), pero los datos se encuentran agrupados de tal manera que, aun manteniendo su simplicidad, igualmente favorecen dicha rapidez. Esto permite su modificación conforme cambien los requerimientos.

3.1 Diseño del modelo dimensional

Kimball, en [6], establece cuatro pasos para diseñar el modelo dimensional:

Paso 1: Seleccionar el proceso de negocio

Teniendo en cuenta los requerimientos y los datos disponibles, el primer paso consiste en decidir cuál es el proceso de negocio a modelar.

Para aplicar esta primera instancia en este trabajo hay que tener en cuenta su objetivo: analizar la eficiencia en la parametrización de anuncios, en las principales plataformas sociales (Facebook, Instagram, Youtube y Twitter). Por lo tanto, el proceso de negocio a modelar es el de la publicación de anuncios en dichas redes. Los datos de estos eventos permiten analizar el impacto de las publicidades según distintos parámetros: red social, sexo y ubicación, entre otros.

Paso 2: Selección de la granularidad (*Grain*)

Seleccionar la granularidad o *grain* significa especificar el nivel de detalle con el que se desea trabajar. La misma depende de las realidades del negocio que se capturan en los procesos seleccionados. Es un paso importante y debe ser definido en términos del negocio.

Para este trabajo, el *grain* estará dado por el registro de las publicaciones agrupadas por día, rango de edad, género, localización, red social, producto y gustos.

Paso 3: Elegir las dimensiones

Las dimensiones surgen naturalmente de los procesos de negocios. Las tablas de dimensiones contienen un conjunto de atributos (generalmente textuales) asociados a los hechos que se deseen medir. Describen el “quién”, “qué”, “dónde”, “cuándo”, “cómo” y “por qué” de los eventos y son los firmes candidatos para filtros y agrupamientos. Por lo general, las tablas de dimensiones tienden a tener múltiples atributos y pocas filas.

Una herramienta de diseño que se puede utilizar en esta etapa es la Bus Matrix. Esta matriz es una guía para todo el equipo de desarrollo y facilita el entendimiento y definición de requerimientos.

Para este caso, el punto de partida pueden ser las preguntas que se quieren resolver:

- **P1:** ¿Qué regiones o ciudades tienen mejores indicadores de efectividad? ¿Hay alguna relación con el producto o familia de productos?
- **P2:** ¿Existe una relación entre la mejora de los indicadores de efectividad con algún segmento de la población objetivo?
- **P3:** ¿Hay alguna plataforma donde, bajo las mismas condiciones, se obtengan mejores tasas de visualización?
- **P4:** ¿Existen relaciones entre plataformas y franjas de edad de usuarios que provoquen mejores tasas de visualización?
- **P5:** ¿El conocimiento del grupo de interés de los usuarios podría ayudar a mejorar los indicadores para determinados productos?

Para responder a la pregunta P1 hay que realizar una consulta donde intervengan las ciudades y los productos. Así, se pueden detectar cuatro dimensiones: **ciudad**, **producto**, **zona** y **familia**. Dichas ciudades están agrupadas por zonas. Lo mismo ocurre con los productos que se clasifican por familias. De aquí se desprende que existe una relación jerárquica entre ciudad – zona y entre producto – familia.
La pregunta P2 se puede responder analizando los indicadores (*clicks* y *hits*) en relación con el sexo, rango de edad y gustos. Por lo tanto, se agregan tres nuevas dimensiones: **sexo**, **rango de edad** y **gustos**.
Los datos de entrada están divididos por red social. De esta manera, para la P3, se puede analizar la efectividad de la publicidad en cada una de ellas. Por lo tanto, se necesita la dimensión **red social,** aunque también se podrían relacionar todas las otras dimensiones o sólo algunas de ellas.
Para dar respuesta a la P4, la consulta a realizar debe agrupar los datos por red social y rango de edad.
Para P5, habría que relacionar gustos con productos.

En definitiva, gracias a este análisis, las dimensiones detectadas son: producto, zonas geográficas, red social, rango de edad, sexo y gustos.
Además, como se quieren realizar agrupaciones semanales, mensuales, trimestrales y anuales, será necesaria una nueva dimensión: **date**, relacionada con todas las preguntas anteriores.
Este diseño dimensional es elástico y posibilita responder a todas las preguntas planteadas y a algunas otras que puedan surgir.
A modo de resumen y entendiendo cada pregunta como un proceso de negocio, se muestra la Bus Matrix:

	Date	Ciudad	Zona	Producto	Familia	Sexo	Rango de Edad	Gustos	Red Social
P1	X	X	X	X	X				
P2	X					X	X	X	
P3	X	X	X	X	X	X	X	X	X
P4	X						X		X
P5	X			X				X	

Paso 4: Identificar hechos

Los hechos se pueden identificar respondiendo a la pregunta "¿Qué es lo que se desea medir?". Se deben corresponder con la granularidad definida y resultan de los eventos de los procesos de negocio seleccionados. Generalmente, son numéricos y aditivos.

En los anuncios, los hechos son: el número de visualizaciones (*prints*) y el número de clicks (*hits*).

3.2 Modelo dimensional obtenido

El modelo dimensional final obtenido se muestra a continuación:

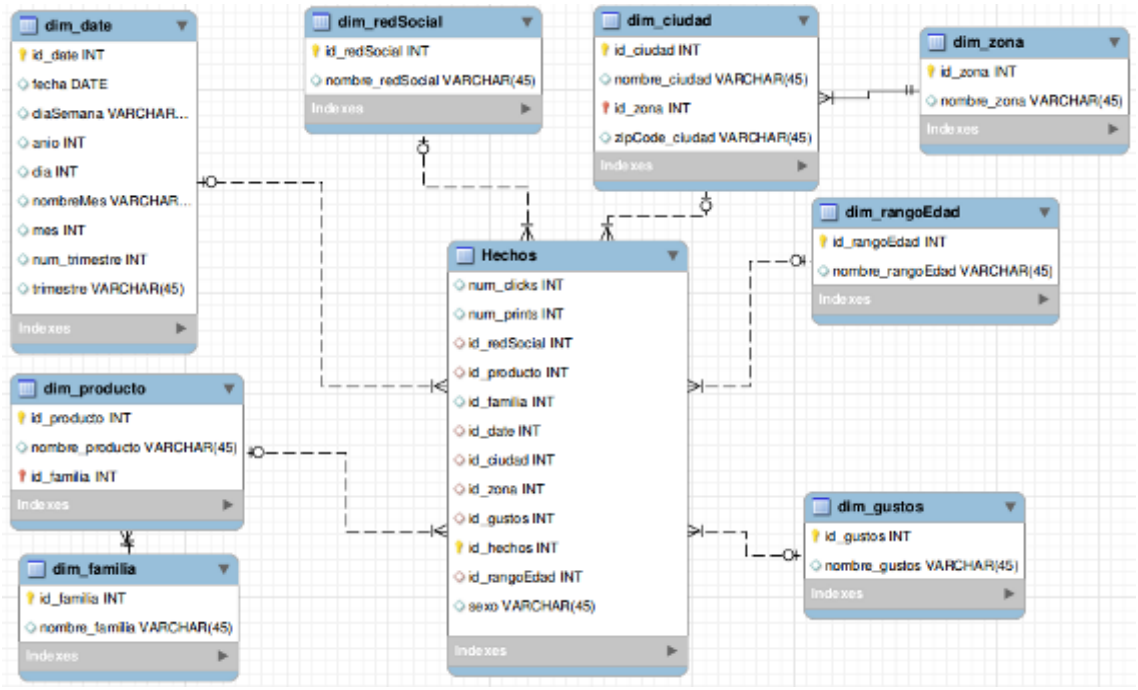


Figura 10: Modelo dimensional

Claramente se ve que se sigue un esquema **copo de nieve** donde no sólo la tabla de hechos se relaciona con otra sino también las tablas de dimensión por estar normalizadas.
En un primer momento, el diagrama se diseñó con una dimensión sexo. Pero debido a la poca cantidad de valores que puede tomar, se decidió

eliminarla y agregarlo como un atributo en la tabla de hechos. Esto facilita los procesos de carga y su posterior análisis.
Como ya fue mencionado, hay dimensiones que cuentan con una jerarquía como Producto – Familia y Ciudad – Zona.
Para una mayor claridad, todas las tablas, excepto la de hechos, contienen un campo del tipo texto para introducir el nombre del atributo.

3.3 Implementación del data warehouse

El diseño físico del data warehouse se implementa en un *schema* de MySQL con el nombre TFM que se reduce a la creación de las tablas correspondientes según el diseño anterior.
Gracias a la herramienta MySQL Workbench, el script de creación se obtiene de forma sencilla a través de la exportación del DER.

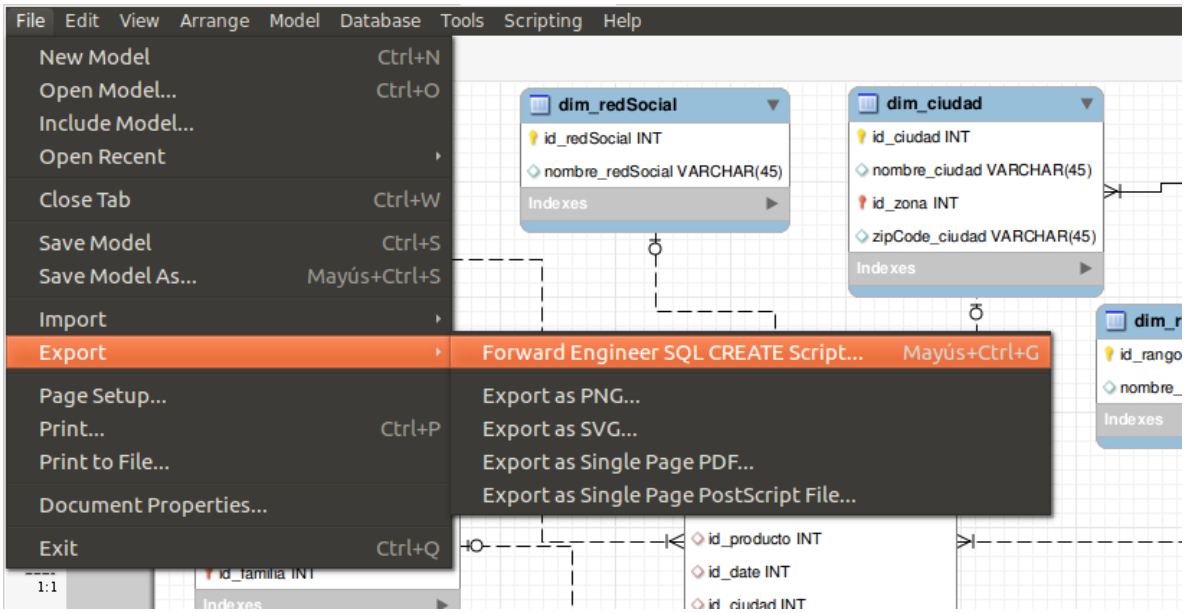


Figura 11: Exportar DER a SQL script

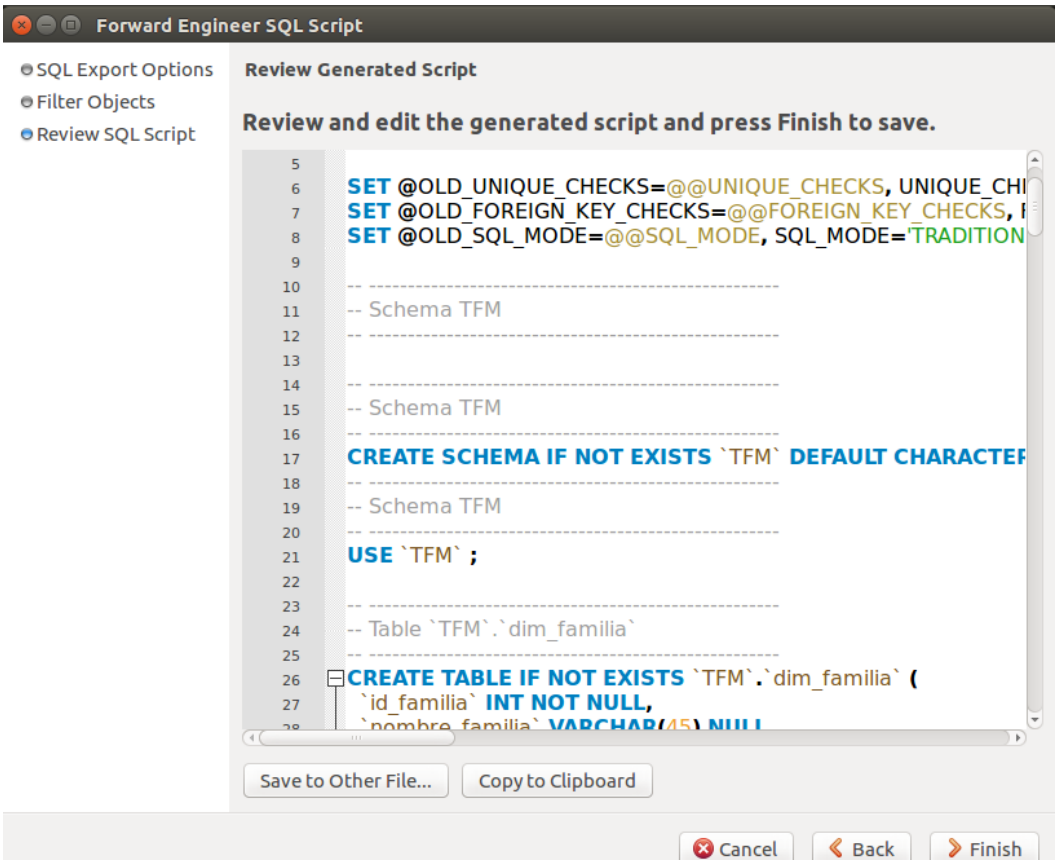


Figura 12: Script SQL

Al ejecutar el script exportado, se crean todas las tablas de dimensiones y la de hechos:

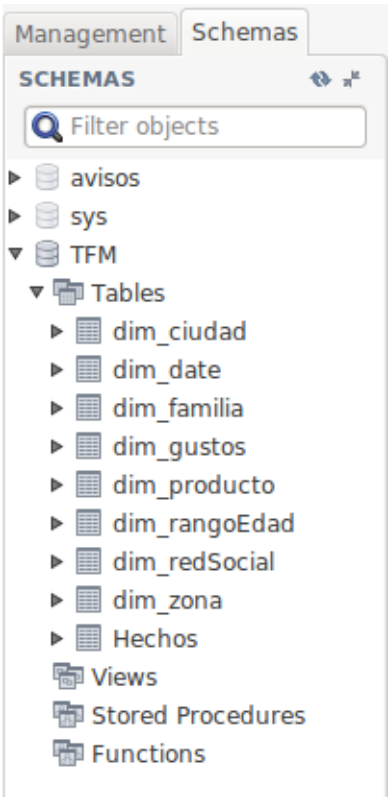


Figura 13: Schema MySQL

Gracias al monitor de MySQL se puede ver la sentencia SQL de creación de cada una de las tablas. Para ingresar, basta con escribir ***mysql – u usuario –p schema***:

```
lorena@lorena-VirtualBox:~/pentaho/pentaho-server$ mysql -u root -p TFM
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 5.7.21-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

Figura 14: Monitor MySQL

Las sentencias de creación de cada una de las tablas se muestran a continuación:

```
mysql> show create table dim_ciudad \G
*************************** 1. row ***************************
       Table: dim_ciudad
Create Table: CREATE TABLE `dim_ciudad` (
  `id_ciudad` int(11) NOT NULL,
  `nombre_ciudad` varchar(45) DEFAULT NULL,
  `dim_zona_id_zona` int(11) NOT NULL,
  PRIMARY KEY (`id_ciudad`,`dim_zona_id_zona`),
  KEY `fk_dim_ciudad_dim_zona1_idx` (`dim_zona_id_zona`),
  CONSTRAINT `fk_dim_ciudad_dim_zona1` FOREIGN KEY (`dim_zona_id_zona`) REFERENC
ES `dim_zona` (`id_zona`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0,00 sec)
```

Figura 15: Sentencia de creación tabla dim_ciudad

```
mysql> show create table dim_date\G
*************************** 1. row ***************************
       Table: dim_date
Create Table: CREATE TABLE `dim_date` (
  `id_date` int(11) NOT NULL AUTO_INCREMENT,
  `fecha` date DEFAULT NULL,
  `diaSemana` varchar(45) DEFAULT NULL,
  `mes` int(11) DEFAULT NULL,
  `anio` int(11) DEFAULT NULL,
  `dia` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_date`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0,00 sec)
```

Figura 16: Sentencia de creación tabla dim_date

```
mysql> show create table dim_familia \G
*************************** 1. row ***************************
       Table: dim_familia
Create Table: CREATE TABLE `dim_familia` (
  `id_familia` int(11) NOT NULL,
  `nombre_familia` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_familia`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0,00 sec)
```

Figura 17: Sentencia de creación tabla dim_familia

```
mysql> show create table dim_gustos \G
*************************** 1. row ***************************
       Table: dim_gustos
Create Table: CREATE TABLE `dim_gustos` (
  `id_gustos` int(11) NOT NULL,
  `nombre_gustos` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_gustos`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0,00 sec)
```

Figura 18: Sentencia de creación tabla dim_gustos

```
mysql> show create table dim_producto \G
*************************** 1. row ***************************
       Table: dim_producto
Create Table: CREATE TABLE `dim_producto` (
  `id_producto` int(11) NOT NULL,
  `nombre_producto` varchar(45) DEFAULT NULL,
  `id_familia` int(11) NOT NULL,
  PRIMARY KEY (`id_producto`,`id_familia`),
  KEY `fk_dim_producto_dim_familia_idx` (`id_familia`),
  CONSTRAINT `fk_dim_producto_dim_familia` FOREIGN KEY (`id_familia`) REFERENCES
 `dim_familia` (`id_familia`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0,00 sec)
```

Figura 19: Sentencia de creación tabla dim_producto

```
mysql> show create table dim_rangoEdad \G
*************************** 1. row ***************************
       Table: dim_rangoEdad
Create Table: CREATE TABLE `dim_rangoEdad` (
  `id_rangoEdad` int(11) NOT NULL,
  `nombre_rangoEdad` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_rangoEdad`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0,00 sec)
```

Figura 20: Sentencia de creación tabla dim_rangoEdad

```
mysql> show create table dim_redSocial \G
*************************** 1. row ***************************
       Table: dim_redSocial
Create Table: CREATE TABLE `dim_redSocial` (
  `id_redSocial` int(11) NOT NULL,
  `nombre_redSocial` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_redSocial`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Figura 21: Sentencia de creación tabla dim_redSocial

```
mysql> mysql> show create table dim_zona \G
*************************** 1. row ***************************
       Table: dim_zona
Create Table: CREATE TABLE `dim_zona` (
  `id_zona` int(11) NOT NULL,
  `nombre_zona` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_zona`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0,00 sec)
```

Figura 22: Sentencia de creación tabla dim_zona

```
mysql> mysql> show create table Hechos\G
*************************** 1. row ***************************
       Table: Hechos
Create Table: CREATE TABLE `Hechos` (
  `num_clicks` int(11) DEFAULT NULL,
  `num_prints` int(11) DEFAULT NULL,
  `id_redSocial` int(11) DEFAULT NULL,
  `id_producto` int(11) DEFAULT NULL,
  `id_date` int(11) DEFAULT NULL,
  `id_ciudad` int(11) DEFAULT NULL,
  `id_zona` int(11) DEFAULT NULL,
  `id_gustos` int(11) DEFAULT NULL,
  `id_rangoEdad` int(11) DEFAULT NULL,
  `sexo` varchar(45) DEFAULT NULL,
  `id_hechos` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id_hechos`),
  KEY `fk_Hechos_dim_producto1_idx` (`id_producto`),
  KEY `fk_Hechos_dim_redSocial1_idx` (`id_redSocial`),
  KEY `fk_Hechos_dim_date1_idx` (`id_date`),
  KEY `fk_Hechos_dim_ciudad1_idx` (`id_ciudad`,`id_zona`),
  KEY `fk_Hechos_dim_gustos1_idx` (`id_gustos`),
  KEY `fk_Hechos_dim_rangoEdad1_idx` (`id_rangoEdad`),
  CONSTRAINT `fk_Hechos_dim_ciudad1` FOREIGN KEY (`id_ciudad`, `id_zona`) REFERENCES `dim_ciudad` (`id_ciudad`, `id_zona`) ON DELETE NO ACTION
 ON UPDATE NO ACTION,
  CONSTRAINT `fk_Hechos_dim_date1` FOREIGN KEY (`id_date`) REFERENCES `dim_date` (`id_date`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_Hechos_dim_gustos1` FOREIGN KEY (`id_gustos`) REFERENCES `dim_gustos` (`id_gustos`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_Hechos_dim_producto1` FOREIGN KEY (`id_producto`) REFERENCES `dim_producto` (`id_producto`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
  CONSTRAINT `fk_Hechos_dim_rangoEdad1` FOREIGN KEY (`id_rangoEdad`) REFERENCES `dim_rangoEdad` (`id_rangoEdad`) ON DELETE NO ACTION ON UPDATE
 NO ACTION,
  CONSTRAINT `fk_Hechos_dim_redSocial1` FOREIGN KEY (`id_redSocial`) REFERENCES `dim_redSocial` (`id_redSocial`) ON DELETE NO ACTION ON UPDATE
 NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=152019 DEFAULT CHARSET=utf8
1 row in set (0,00 sec)
```

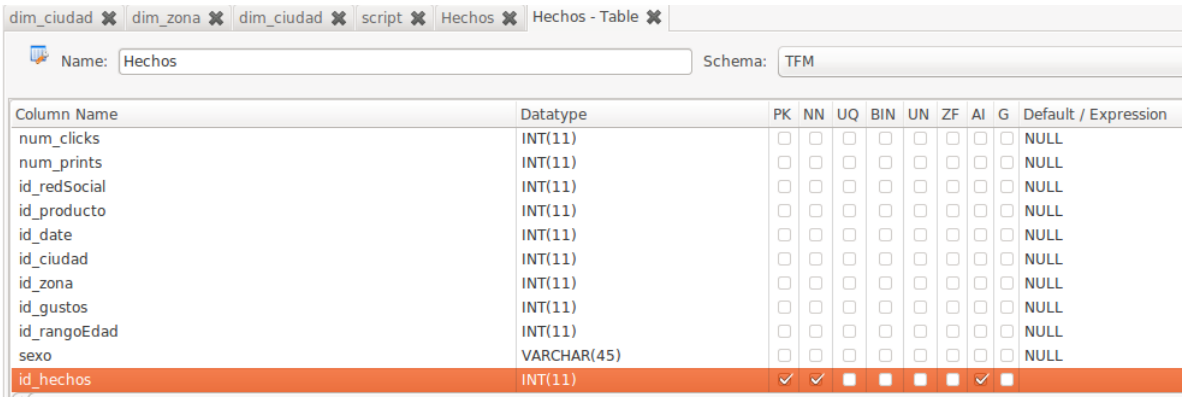


Figura 23: Sentencia de creación tabla Hechos

4. Diseño e Implementación de Procesos ETL

4.1 Análisis de datos de entrada

Los datos de entrada fueron dados en un fichero Excel que contiene seis hojas:

Hoja 1: Products

Campo	Descripción
Product	Tipo de producto
Family	Familia a la que pertenece el aviso publicitario

Hoja 2: Zones

Campo	Descripción
Zone	Zona geográfica donde se registra la visualización del anuncio publicitario
City	Ciudad del visitante
ZipCode	Código postal de la ciudad

Las cuatro hojas restantes, una por cada plataforma analizada: Instagram; Facebook; YouTube y Twitter

Campo	Descripción
Date	Día donde se registra las visualizaciones del aviso publicitario
ZipCode	Ciudad del visitante
Product	Tipo de producto
Age	Rango de edad del visitante
Gender	Género del visitante
Likes	Gustos del visitante
Prints	Número de visualizaciones
Hits	Número de clicks realizados

Para diseñar los procesos ETL, se separa cada hoja en un archivo Excel.

4.2 Procesos ETL

La preparación de los datos es una de las tareas que más tiempo y recursos puede requerir. Es importante y nunca se debe subestimar esta etapa ya que de ella depende el éxito del análisis final. En el contexto del Business Intelligence, dicha preparación se realiza con los denominados procesos ETL. Estos consisten en extraer (*extract*), transformar (*transform*) y cargar (*load*) los datos provenientes de distintas fuentes en el data warehouse. A los efectos de poder realizar los análisis correspondientes y de obtener información relevante para los procesos de negocios definidos en la etapa anterior, se buscan integrar los datos.
El primer paso del proceso consiste en extraer los datos. Muchas veces significa entender las fuentes y almacenar los datos necesarios en una

base de *staging* para su posterior manipulación. Sobre ella no se ejecuta ninguna consulta de muestra de resultados. No ofrece ningún servicio de visualización ni acceso al usuario final. Por ejemplo, podría emplearse en sistemas donde los datos provienen de diversas fuentes y en distintos momentos. En este caso, no fue necesario utilizarla debido a que las transformaciones no fueron muy complejas y los datos estaban siempre disponibles.
Cabe destacar que Kimball en [6] define un área de *staging* que consiste en todo lo que se encuentra entre las fuentes de datos y el data warehouse, incluida la base de datos definida líneas más arriba. En este trabajo, esta área corresponde al conjunto de transformaciones y *jobs* utilizados para cargar los datos en el data warehouse.
Una vez que los datos se encuentran en esta *staging area*, el siguiente paso consiste en aplicarles distintas transformaciones. Entre ellas, se pueden mencionar: la limpieza de los datos (completar valores faltantes, eliminar inconsistencias y duplicados, normalizarlos, estandarizarlos, entre otros), combinar distintas fuentes, eliminar registros duplicados, agregarle los identificadores necesarios según el diseño del data warehouse. Todas estas actividades son previas a la carga final, último paso del proceso.
Este último paso consiste simplemente en cargar masivamente en el data warehouse, los datos resultantes de la etapa anterior.
El proceso apenas descrito puede llevarse a cabo “a mano” o utilizando herramientas que ayudan a que esta tarea sea más sencilla. La plataforma seleccionada, Pentaho, trabaja con Kettle y, a su vez, ofrece una interfaz gráfica llamada Spoon la cual permite armar los ETL de una manera *drag and drop*.
En el Anexo 2 se puede ver en mayor detalle en qué consiste y cómo se ejecuta.

Para comenzar con la implementación, se debe configurar la conexión al data warehouse. En este caso, se trata de una base de datos MySQL.

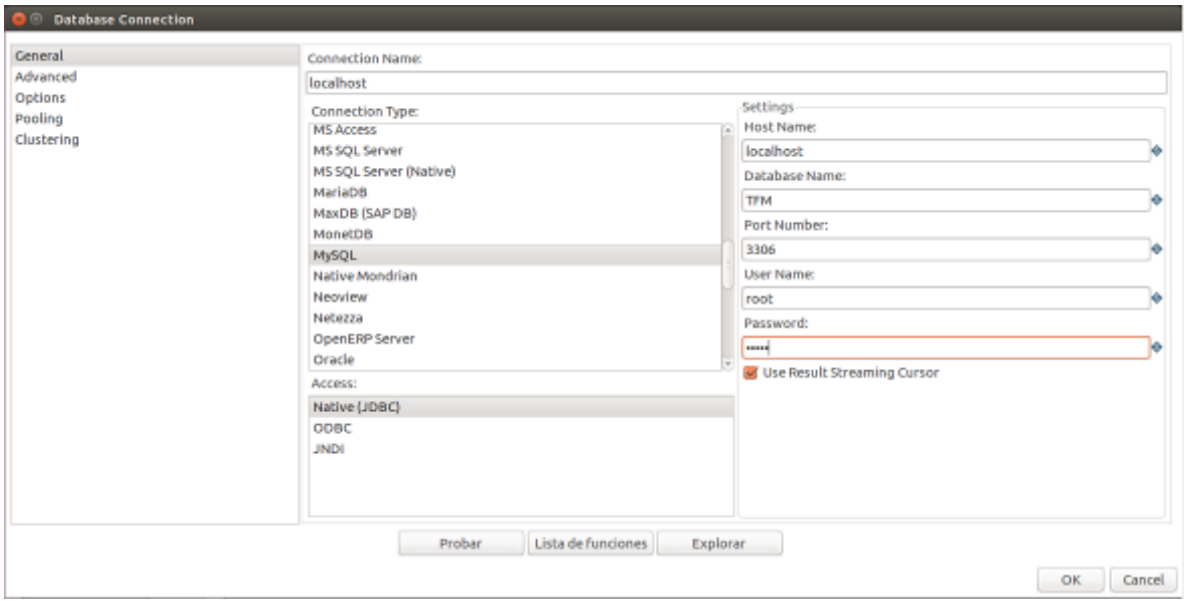


Figura 24: Creación de conexión al DWH

A continuación, se presentarán los procesos ETL necesarios para cargar los datos en el data warehouse. Para mayor detalle, se puede consultar el Anexo 3 donde se explica cada uno de ellos.

Dimensión Producto y Familia

El primer paso o salto como lo llama Spoon, consiste en leer el archivo correspondiente a los productos y sus familias. Como se dijo anteriormente, está compuesto por dos columnas de tipo String que contienen el nombre del producto y la familia a la que pertenece.
Para cargar los datos en ambas dimensiones, se necesita previamente prepararlos y agregarles un identificador. Para ello, utilizando “filas únicas”, se descartan aquellos productos o familias repetidos. Luego, se crea un identificador para familia y otro para producto.
Como se planteó una jerarquía entre Producto – Familia, en la dimensión producto se necesita asociar el identificador de la familia creado anteriormente. Esto se realiza fácilmente gracias a “Multiway Merge Join”.
Finalmente se cargan los datos en el data warehouse en sendas tablas.
A continuación, se muestra el diagrama resultante:

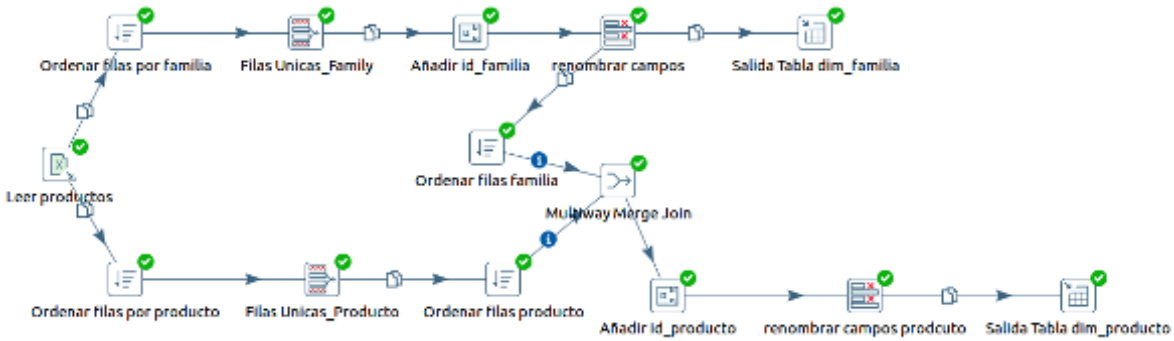


Figura 25: ETL Dimensión Producto y Familia

Se puede ver en las siguientes figuras que se cargaron correctamente los datos:

dim_familia | dim_producto

Limit to 1000 rows

```
1 • SELECT * FROM TFM.dim_familia;
```

Result Grid | Filter Rows: | Edit: | Export/

#	id_familia	nombre_familia
1	1	Accessory
2	2	Culture
3	3	Electronics
4	4	Sports
5	5	Wear
*	NULL	NULL

Figura 26: Dimensión familia

dim_familia | dim_producto

Limit to 1000 rows

```
1 • SELECT * FROM TFM.dim_producto;
```

Result Grid | Filter Rows: | Edit: | Export/

#	id_producto	nombre_producto	id_familia
1	1	Scarf	1
2	2	Watch	1
3	3	Theater	2
4	4	Trip	2
5	5	Mobile Phone	3
6	6	Sneakers	4
7	7	Dress	5
8	8	Sheatshirt	5
*	NULL	NULL	NULL

Figura 27: Dimensión producto

Dimensión Edad

El primer paso consiste en leer los datos de cualquier hoja correspondiente a las redes sociales. A través del operador “ordenar filas” se seleccionan los distintos rangos de edades. Luego, se añade un identificador.

Como los nombres de los campos de las hojas de Excel no coinciden con los de los atributos del data warehouse, se deben renombrar. Finalmente, se cargan los datos en la tabla dim_rangoEdad.

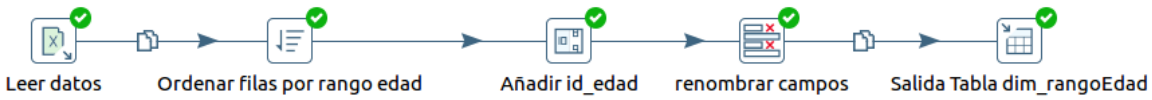


Figura 28: ETL Dimensión Edad

Como se puede ver en la siguiente figura, se cargaron correctamente los cuatro rango de edades disponibles:

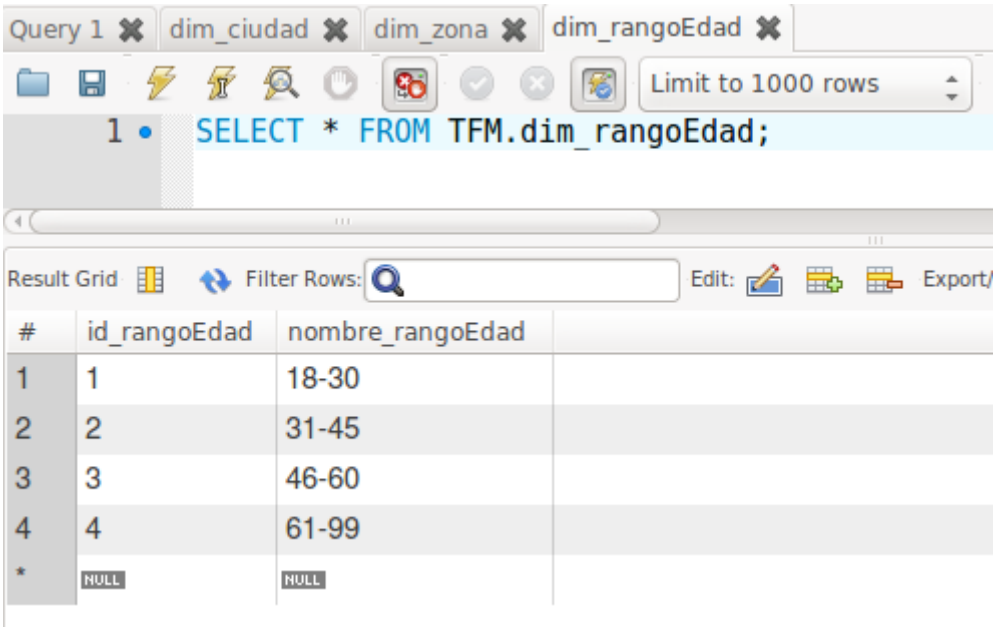


Figura 29: Dimensión edad

Dimensión Gustos

Esta dimensión es prácticamente igual a la anterior. Difiere en que se seleccionan los gustos o *likes* de cualquier hoja de red social.

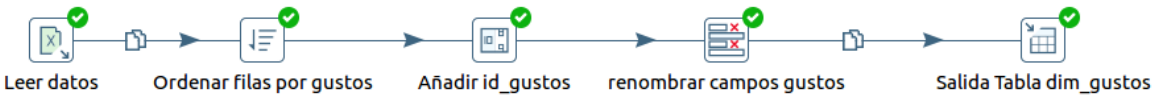


Figura 30: ETL Dimensión Gustos

En este caso, se obtuvieron ocho gustos:

Query 1 | dim_gustos

Limit to 1000 rows

```
1 • SELECT * FROM TFM.dim_gustos;
```

Result Grid | Filter Rows: | Edit: | Export

#	id_gustos	nombre_gustos
1	1	Business
2	2	Cars
3	3	Fashion
4	4	Garden
5	5	People
6	6	Sports
7	7	Technology
8	8	Travel
*	NULL	NULL

Figura 31: Dimensión gustos

Dimensión Ciudad y Zona

El primer salto consiste en leer el archivo correspondiente a las ciudades. Como se planteó dividir los datos en dos dimensiones, ciudades y zonas, se van a cargar ambas dimensiones en el mismo proceso. Para ello, utilizando “ordenar filas” se descartan aquellas zonas o ciudades (combinación ciudad – zipCode) repetidas. Luego, se crea un identificador para zona y otro para ciudad.

En un primer momento se planteó utilizar el zipCode como identificador. Pero a la hora de cargar la tabla de hechos, se producían errores en aquellas ciudades cuyo código postal comenzaban con 0.

Al diseñar una jerarquía entre ciudades – zona, en la dimensión ciudad se necesita asociar el identificador de la zona creada anteriormente. Esto se realiza gracias a “Multiway Merge Join”.

Finalmente se cargan los datos en el data warehouse en sendas tablas.

A continuación, se muestra el diagrama resultante:

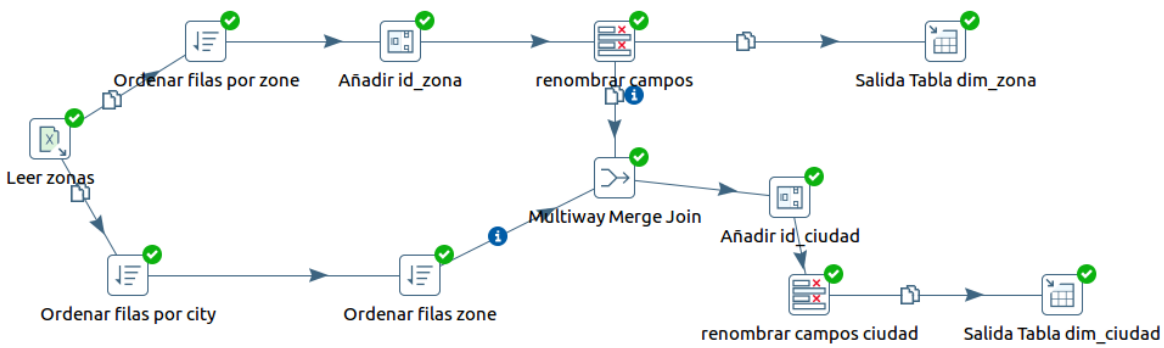


Figura 32: ETL Dimensión Zona y Ciudad

Con consultas SQL, se corrobora que se cargaron correctamente los datos:

Query 1 | dim_ciudad | dim_zona

Limit to 1000 rows

```
1 • SELECT * FROM TFM.dim_zona;
```

Result Grid | Filter Rows: | Edit: | Export

#	id_zona	nombre_zona
1	1	Centre
2	2	Coast
3	3	North
4	4	South
*	NULL	NULL

Figura 33: Dimensión zona

dim_ciudad

Limit to 50000 rows

```
1 • SELECT * FROM TFM.dim_ciudad;
```

Result Grid | Filter Rows: | Edit: | Export

#	id_ciudad	nombre_ciudad	id_zona	zipCode_ciudad
1	1	Madrid	1	28014,0
2	2	Madrid	1	28019,0
3	3	Toledo	1	45005,0
4	4	Valladolid	1	47011,0
5	5	Barcelona	2	08005
6	6	Barcelona	2	08029
7	7	Tarragona	2	43006,0
8	8	Valencia	2	46025,0
9	9	Bilbao	3	48003,0
10	10	Bilbao	3	48008,0
11	11	Gijon	3	33209,0
12	12	La Coruña	3	15010,0
13	13	Cadiz	4	11011,0
14	14	Malaga	4	29009,0
15	15	Sevilla	4	41011,0

Figura 34: Dimensión ciudad

Dimensión Date

La dimensión Date es muy común en los proyectos BI. En Kettle puede encontrarse un ejemplo como punto de partida. En particular, se utilizó la transformación “General – Populate date dimension.ktr” del directorio Samples/Transformations perteneciente al módulo Data Integration de Pentaho.

La transformación inicial es la que se muestra a continuación:

Figura 35: ETL de Kettle para dimensión date

Como se necesitan los 365 días del año 2017 para este trabajo, se deben cambiar algunos parámetros como la fecha inicial y la cantidad de “iteraciones” necesarias.

Inicialmente, se obtenían números para los días y meses. Como en el data warehouse se necesitan los nombres de los días y meses debieron mapearse. Los dos últimos pasos consisten en renombrar los campos y cargar los datos.

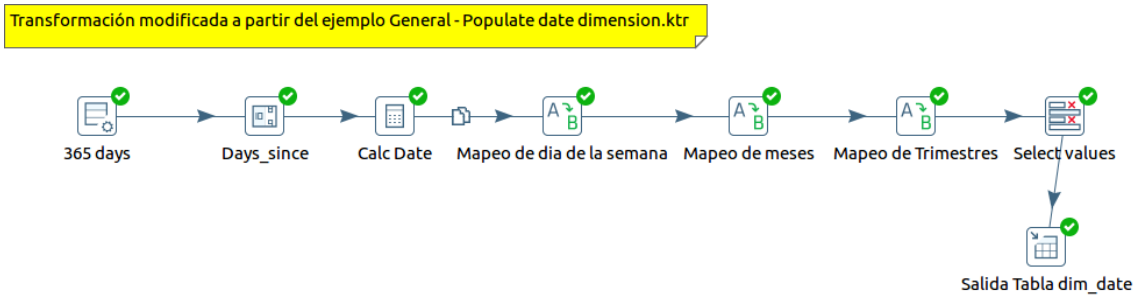


Figura 36: ETL Dimensión Date

Se ve en la siguiente figura como se cargan los datos en esta dimensión:

```
1 •  SELECT * FROM TFM.dim_date;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

#	id_date	fecha	diaSemana	anio	dia	nombreMes	mes	num_trimestre	trimestre
1	1	2017-01-01	Domingo	2017	1	Enero	1	1	Primer Trimestre 2017
2	2	2017-01-02	Lunes	2017	2	Enero	1	1	Primer Trimestre 2017
3	3	2017-01-03	Martes	2017	3	Enero	1	1	Primer Trimestre 2017
4	4	2017-01-04	Miércoles	2017	4	Enero	1	1	Primer Trimestre 2017
5	5	2017-01-05	Jueves	2017	5	Enero	1	1	Primer Trimestre 2017
6	6	2017-01-06	Viernes	2017	6	Enero	1	1	Primer Trimestre 2017
7	7	2017-01-07	Sábado	2017	7	Enero	1	1	Primer Trimestre 2017

Figura 37: Dimensión date

Accediendo a la información de la tabla en MySQL Workbench, se puede observar que se cargaron 365 filas correspondientes a todos los días del año 2017.

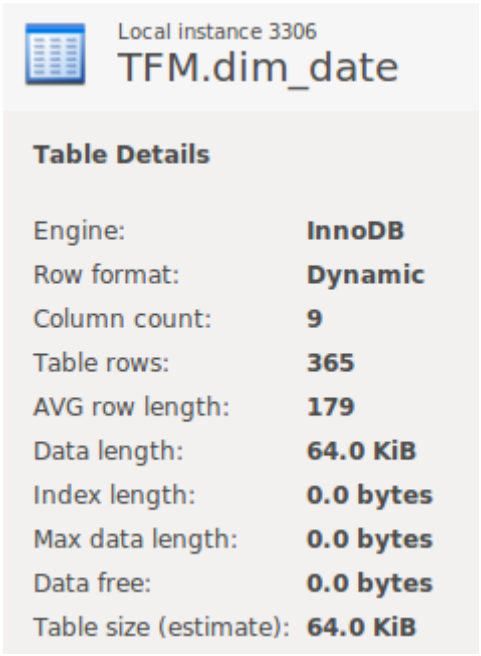


Figura 38: Información de la tabla dim_date

Dimensión Red Social

Debido a su simplicidad, se cargaron las redes sociales con un script SQL:

```
INSERT INTO dim_redSocial VALUES (1,'Facebook');
INSERT INTO dim_redSocial VALUES (2,'Instagram');
INSERT INTO dim_redSocial VALUES (3,'Twitter');
INSERT INTO dim_redSocial VALUES (4,'YouTube');
```

Hechos

Para cargar los datos en la tabla de Hechos, se realizó un trabajo encargado de ejecutar las transformaciones correspondientes a cada red social. Si bien Spoon ofrece la posibilidad de cargar distintas hojas de uno o varios libros Excel y debido a la cantidad de registros en cada una de ellas, la *heap memory* asignada a la JVM se satura y no permite procesar más de una. Por este motivo, se separaron en cuatro hojas distintas los datos y se hicieron cuatro transformaciones iguales para cada red social.
En la siguiente imagen se puede ver la transformación resultante:

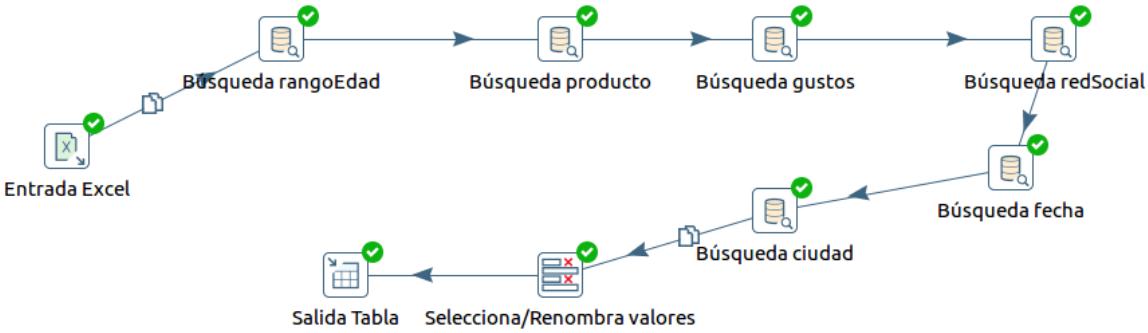


Figura 39: ETL Red Social

Como entrada, se carga cada hoja correspondiente a cada red social. Esto es lo único que varía en las cuatro transformaciones. Luego, se busca el identificador de rango de edad, producto, gustos, red social, fecha, ciudad y zona. Se acondicionan los nombres de los atributos para que coincidan con los de la tabla de hechos. Finalmente, se cargan los datos en dicha tabla.
El trabajo encargado de coordinar estas cuatro transformaciones es:

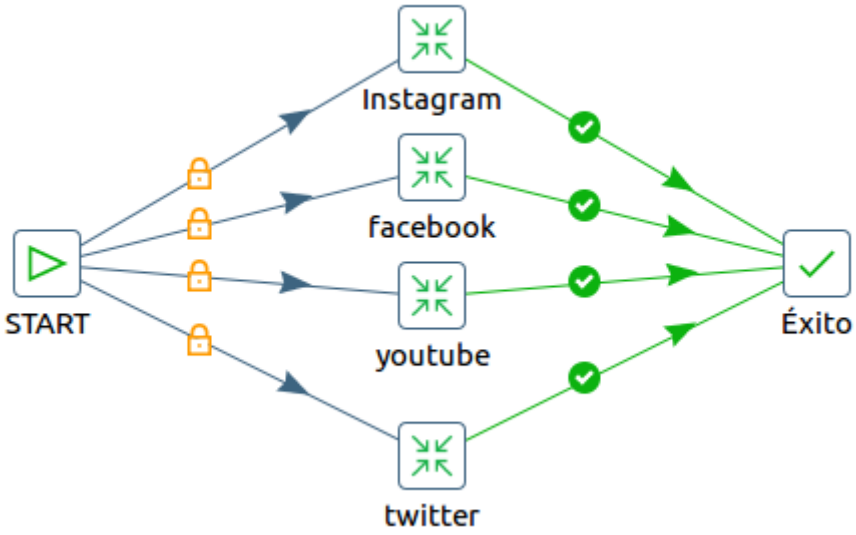


Figura 40: Trabajo para carga de tabla de hechos

La forma de constatar que se cargaron los 1.495.040 registros es a través de una sentencia SQL debido a que no se puede obtener usando la información de la tabla que provee MySQL Workbench. Para mayor detalle de este bug consultar el Anexo 3.

```
1 • SELECT count(*) FROM TFM.Hechos;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	count(*)
1	1495040

Figura 41: Tabla de hechos

Trabajo de Carga Final

Para realizar la carga final de los datos en el data warehouse, se crea un trabajo encargado de coordinar todas y cada una de las transformaciones y trabajo descriptos.

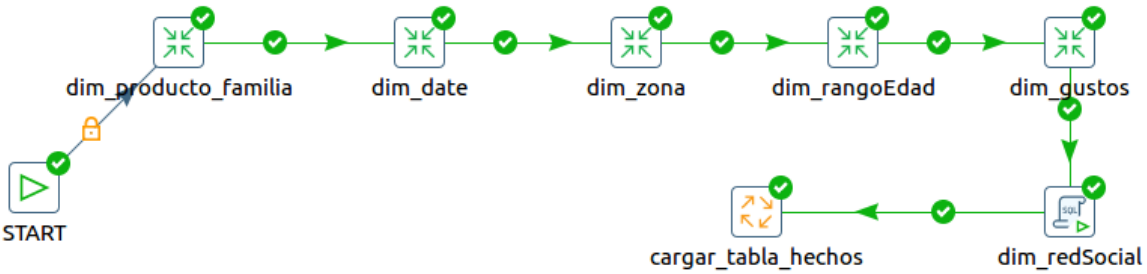


Figura 42: Trabajo carga final

Luego de 24 minutos de proceso se cargaron todos los datos en el data warehouse.

5. Implementación de la capa de análisis

Cubos OLAP

Una vez definidos todos los procesos ETL y cargados los datos al data warehouse, recién queda habilitada la posibilidad de analizarlos. Entre las opciones disponibles, se destaca la utilización de OLAP (*OnLine Analytical Processing* o su equivalente, procesamiento analítico en línea). En [5], OLAP se define como el método para organizar y consultar datos sobre una estructura multidimensional que proporciona mayor agilidad y flexibilidad para el análisis de la información. El hecho de permitir un análisis multidimensional implica que la información esté estructurada en ejes (puntos de vista) y celdas (valores), lo que comúnmente se denomina “cubo”.

Antes de continuar, es conveniente definir distintos conceptos relacionados con OLAP:

- *Esquema*: conjunto de cubos, dimensiones, tablas de hecho, métricas y roles.
- *Cubo*: conjunto de dimensiones relacionadas a una tabla de hecho.
- *Dimensión*: distintas perspectivas de análisis de un proceso de negocio.
- *Tabla de hechos*: representación del proceso de negocio que se desea analizar.
- *Métricas*: resultados de ese proceso de negocio. En el contexto de este trabajo, por ejemplo, serían los prints y hits, que son las medidas que se encuentran en la intersección de las diferentes dimensiones del cubo.
- *Jerarquía*: estructura para navegar por los posibles valores de una dimensión. Se compone de diferentes niveles.
- *Nivel*: posible agrupamiento de una jerarquía.

Las herramientas OLAP se componen de dos partes: el motor encargado de leer los cubos y ejecutar consultas MDX, y un visor que provee la interfaz para realizar el análisis. En Pentaho, el motor OLAP es Mondrian, el cual puede encontrarse en otras plataformas BI como, por ejemplo, Spago BI o Jasper y el visor definido por defecto es JPivot.

Como punto de partida se debe crear un cubo OLAP. Pentaho, a través de Schema Workbench, permite obtener un archivo xml que Mondrian se encargará de leer. Asimismo, posibilita la creación de dimensiones a nivel de cubo o a nivel de esquema. La diferencia estriba en que las primeras estarán definidas únicamente para el cubo en cuestión. En cambio, las dimensiones de esquema se pueden compartir en distintos cubos. Si bien en este trabajo sólo se definió un solo cubo para dar las respuestas pertinentes, para evitar su redundancia en el caso que se diseñen otros cubos, se crearon dimensiones a nivel de esquema. La única dimensión a nivel cubo es sexo.

La siguiente imagen detalla cómo quedó definido el cubo Publicidad, sus dimensiones y sus métricas. La explicación de cómo crear el cubo se encuentra en el anexo 4.

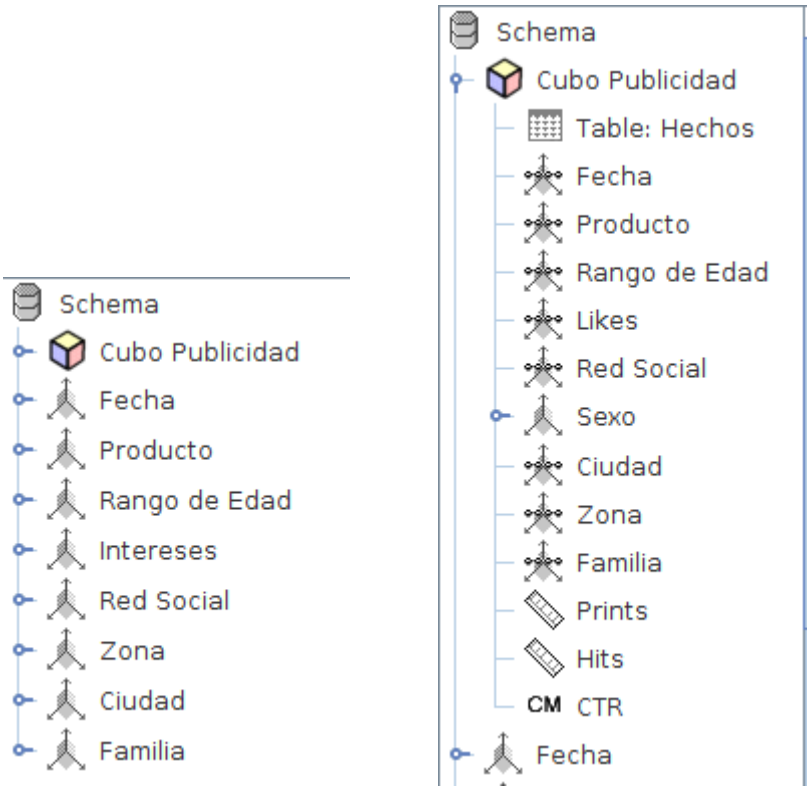


Figura 43: Cubo Publicidad

La única dimensión cuya jerarquía tiene más de un nivel es fecha. La definición de sus niveles sigue un criterio descendente: del más global al más detallado.

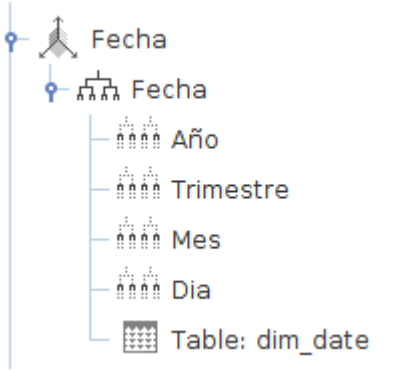


Figura 44: Definición de la dimensión Fecha

Una vez definido el cubo, se lo debe publicar en el servidor de Pentaho. Luego, haciendo uso del visor elegido, será posible realizar los distintos cortes y responder las preguntas planteadas.
Se llevaron a cabo distintas pruebas con JPivot pero debido a la lentitud en su respuesta y a lo poco intuitivo que resultó ser, se optó por instalar el plugin Saiku que provee tanto la información en forma de tabla como de gráfico.

Análisis de la información

Para realizar el análisis de los datos cargados en el data warehouse y haciendo uso del cubo diseñado, se partirá de las preguntas formuladas al inicio del trabajo.

Además, es conveniente recordar los indicadores:

- *Prints*: cantidad que representa al número de visualizaciones.
- *Hits*: cantidad que representa al número de clicks realizados en un anuncio.
- *CTR*: el indicador mide la eficacia de una campaña de publicidad online. Relaciona la cantidad de clicks (*hits*) con la cantidad de publicaciones (*prints*).

¿Qué regiones o ciudades tienen mejores indicadores de efectividad? ¿Hay alguna relación con el producto o familia de productos?

En un primer momento, la consulta consiste en agrupar los *prints* y los *hits* por zona geográfica (archivo P1_a). Se puede observar que la zona de la costa presenta mejores indicadores que el resto.

Zona	Prints	Hits
Centre	25.530.293	2.504.853
Coast	28.724.612	3.124.266
North	25.151.686	2.471.722
South	26.305.307	2.426.026

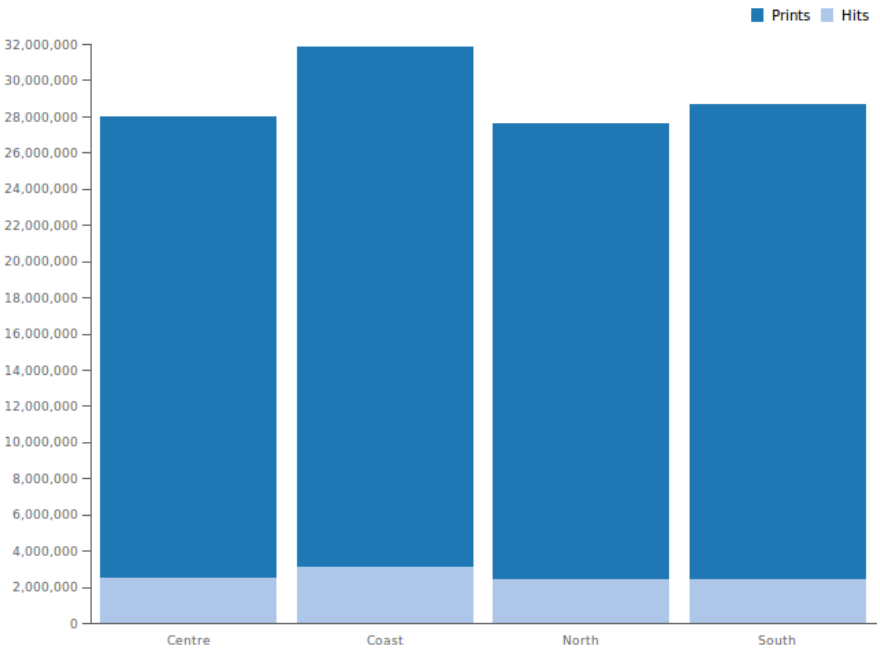


Figura 45: Relación de los indicadores y las zonas

Estos mismos datos se pueden mostrar en un gráfico de torta donde se aprecian los porcentajes asociados a cada una de estas medidas según la zona.

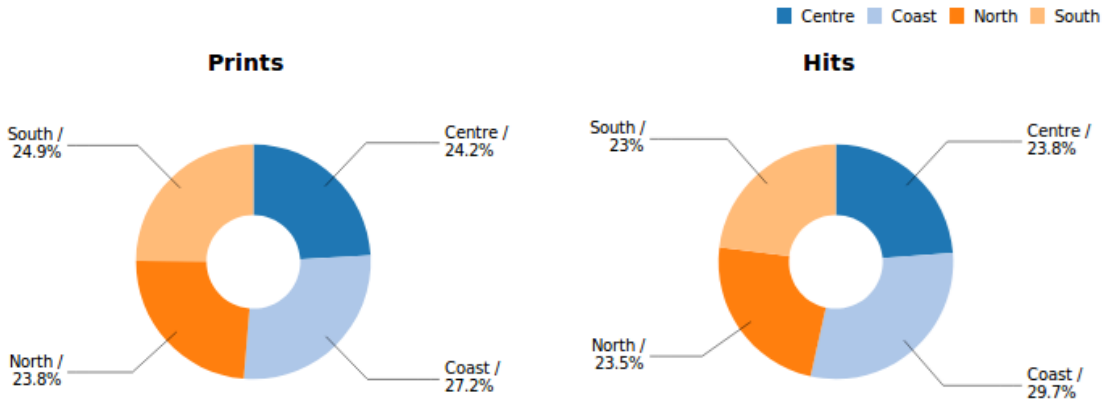


Figura 46: Relación de los indicadores y las zonas

Ejecutando otra consulta (archivo P1_b) se puede ver que el CTR, en la costa, es superior que en el resto de las zonas, siendo el sector sur el de menor valor.

Zona	CTR
Centre	0,098
Coast	0,109
North	0,098
South	0,092

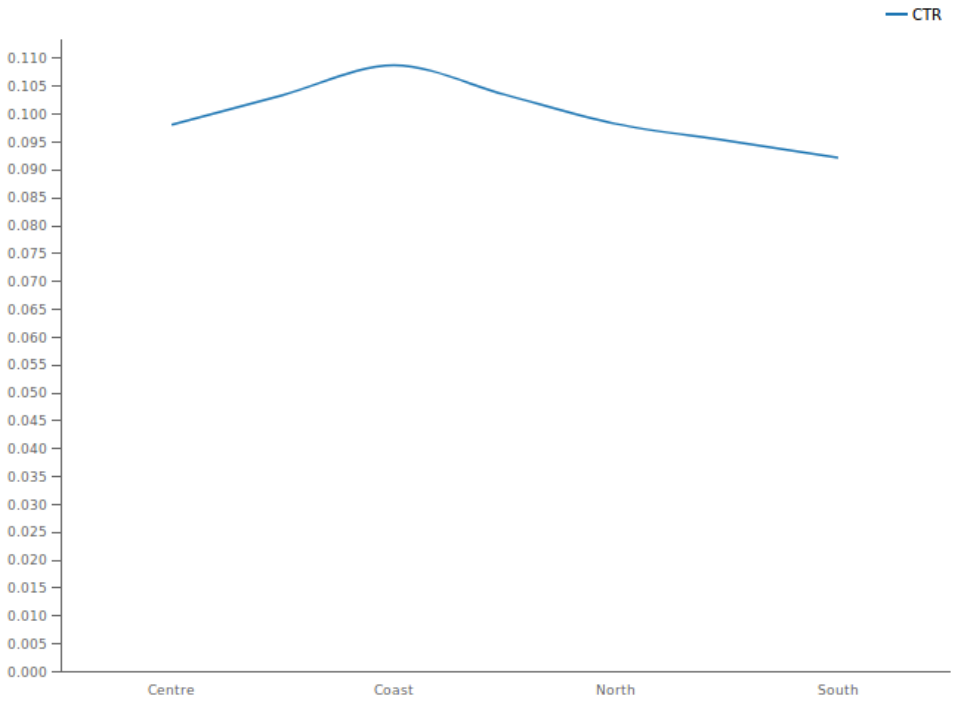


Figura 47: Relación CTR y zonas

Si se analiza a nivel de ciudades, agregando esta dimensión sin perder de vista la zona, el resultado sería el siguiente (archivo P1_c):

Zona	Centre		Coast		North		South	
Ciudad	Prints	Hits	Prints	Hits	Prints	Hits	Prints	Hits
Barcelona	-	-	15.211.350	1.721.906	-	-	-	-
Bilbao	-	-	-	-	13.342.599	1.463.287	-	-
Cadiz	-	-	-	-	-	-	6.099.472	489.193
Gijon	-	-	-	-	5.705.302	488.470	-	-
La Coruña	-	-	-	-	6.103.785	519.965	-	-
Madrid	15.248.125	1.731.760	-	-	-	-	-	-
Malaga	-	-	-	-	-	-	6.480.185	592.644
Sevilla	-	-	-	-	-	-	13.725.650	1.344.189
Tarragona	-	-	6.471.654	666.068	-	-	-	-
Toledo	4.957.407	380.472	-	-	-	-	-	-
Valencia	-	-	7.041.608	736.292	-	-	-	-
Valladolid	5.324.761	392.621	-	-	-	-	-	-

Figura 48: Relación de los indicadores, zonas y ciudades

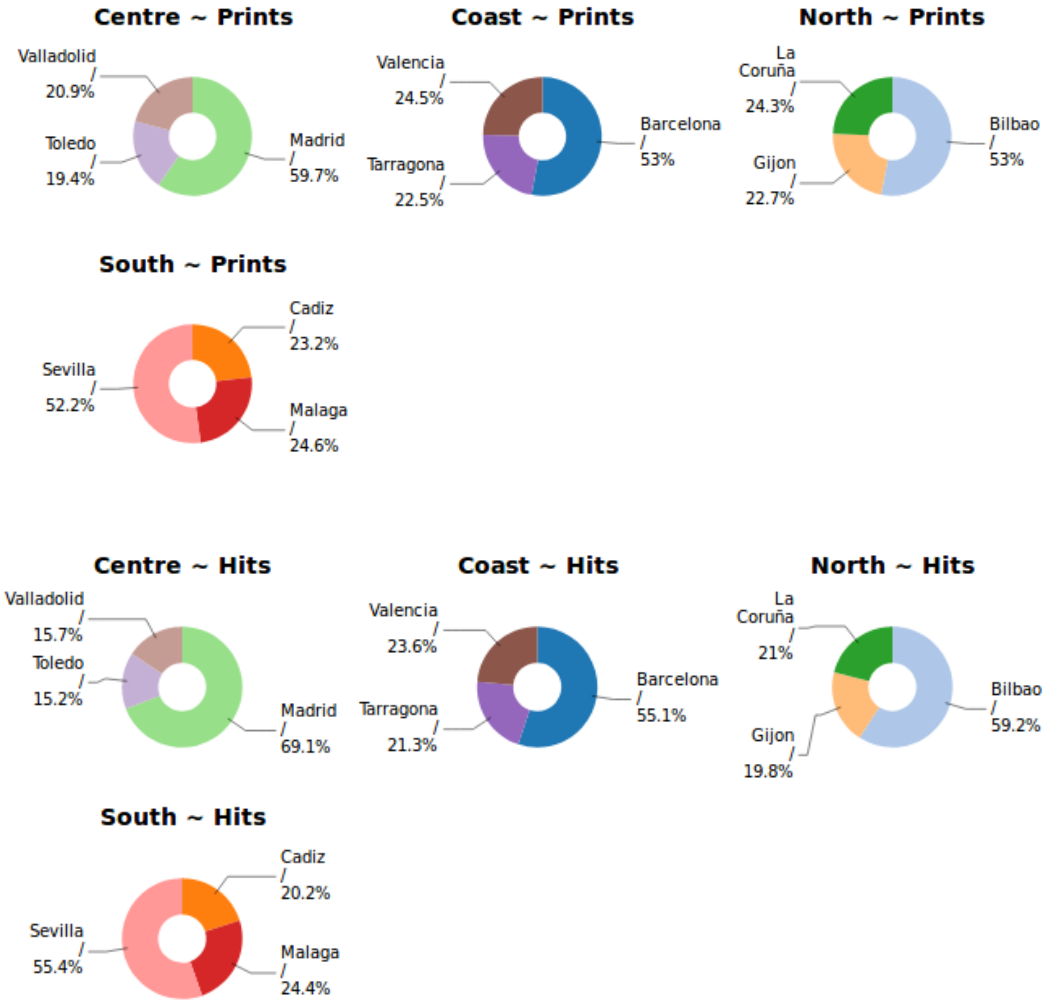


Figura 49: Relación de los indicadores, zonas y ciudades

Para el tercer indicador, la consulta (archivo P1_d) arroja los siguientes resultados:

Zona	Ciudad	CTR
Centre	Madrid	0,114
	Toledo	0,077
	Valladolid	0,074
Coast	Barcelona	0,113
	Tarragona	0,103
	Valencia	0,105
North	Bilbao	0,11
	Gijon	0,086
	La Coruña	0,085
South	Cadiz	0,08
	Malaga	0,091
	Sevilla	0,098

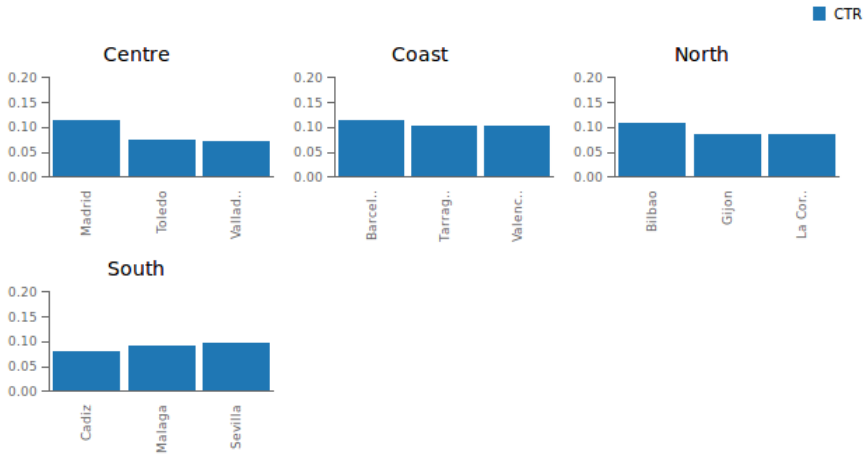


Figura 50: Relación CTR y ciudad

Claramente, en cada zona, existe una ciudad con mayor incidencia que otras: Madrid en la zona del centro; Barcelona en la de la costa; Bilbao en el norte y en el sur, se destaca Sevilla.
Por lo tanto, se puede afirmar no sólo que la zona de mayor incidencia es la costera sino que además cuenta con la más alta cantidad de interesados en la ciudad de Barcelona.

Por otro lado, se necesita analizar conjuntamente a la zona, la familia de los productos (archivo P1_e y archivo P1_f).

Zona	Centre		Coast		North		South	
Familia	Prints	Hits	Prints	Hits	Prints	Hits	Prints	Hits
Accessory	6.386.434	626.306	7.192.554	784.232	6.299.613	617.743	6.566.759	608.440
Culture	6.386.631	626.634	7.158.274	778.825	6.284.768	617.505	6.591.922	607.242
Electronics	3.177.984	311.798	3.590.269	389.823	3.134.875	308.141	3.284.497	304.685
Sports	3.190.371	313.576	3.577.242	388.468	3.148.665	308.580	3.286.201	302.813
Wear	6.388.873	626.539	7.206.273	782.918	6.283.765	619.753	6.575.928	602.846

Figura 51: Relación de las familias con las zonas

Zona	Centre	Coast	North	South
Familia	CTR	CTR	CTR	CTR
Accessory	0,098	0,109	0,098	0,093
Culture	0,098	0,109	0,098	0,092
Electronics	0,098	0,109	0,098	0,093
Sports	0,098	0,109	0,098	0,092
Wear	0,098	0,109	0,099	0,092

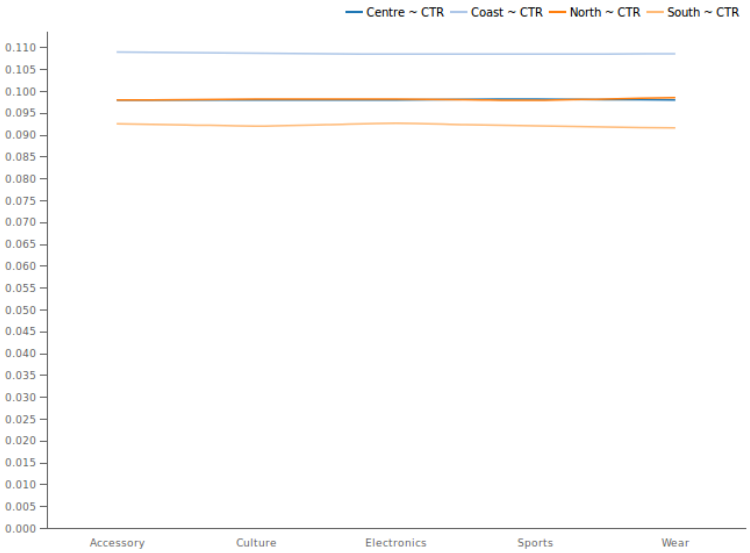


Figura 52: Relación de las familias con las zonas

Se descarta una relación entre la familia del producto y la zona.

Para saber si algún producto se destaca en la zona respecto a otros, se debe llevar a cabo una nueva consulta (archivo P1_g y archivo P1_h). Se obtiene:

Zona	Centre		Coast		North		South	
Producto	Prints	Hits	Prints	Hits	Prints	Hits	Prints	Hits
Scarf	3.194.843	312.769	3.597.217	391.152	3.147.328	308.139	3.287.380	304.838
Watch	3.191.591	313.537	3.595.337	393.080	3.152.285	309.604	3.279.379	303.602
Theater	3.193.861	312.625	3.579.300	389.337	3.147.703	310.026	3.290.953	302.004
Trip	3.192.770	314.009	3.578.974	389.488	3.137.065	307.479	3.300.969	305.238
Mobile Phone	3.177.984	311.798	3.590.269	389.823	3.134.875	308.141	3.284.497	304.685
Sneakers	3.190.371	313.576	3.577.242	388.468	3.148.665	308.580	3.286.201	302.813
Dress	3.201.525	314.511	3.604.377	392.538	3.142.765	312.179	3.280.992	300.043
Sheatshirt	3.187.348	312.028	3.601.896	390.380	3.141.000	307.574	3.294.936	302.803

Figura 53: Relación de productos y zonas

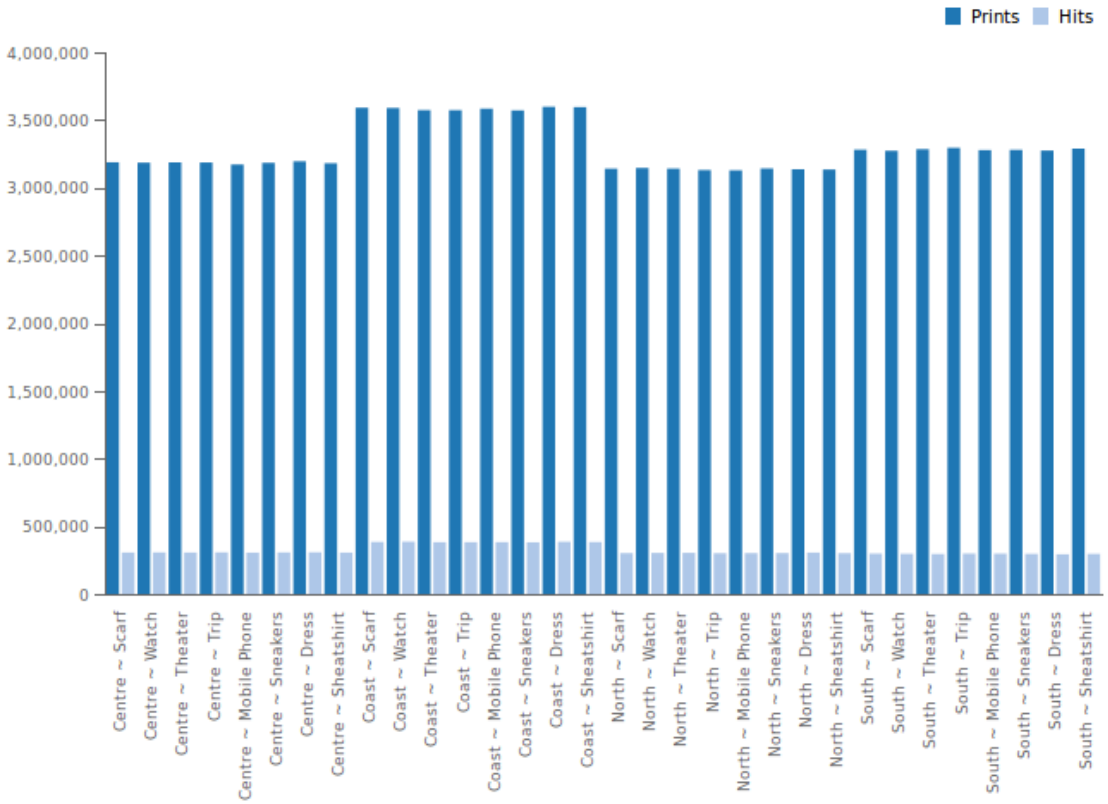


Figura 54: Relación de productos y zonas

Zona	Centre	Coast	North	South
Producto	CTR	CTR	CTR	CTR
Scarf	0,098	0,109	0,098	0,093
Watch	0,098	0,109	0,098	0,093
Theater	0,098	0,109	0,098	0,092
Trip	0,098	0,109	0,098	0,092
Mobile Phone	0,098	0,109	0,098	0,093
Sneakers	0,098	0,109	0,098	0,092
Dress	0,098	0,109	0,099	0,091
Sheatshirt	0,098	0,108	0,098	0,092

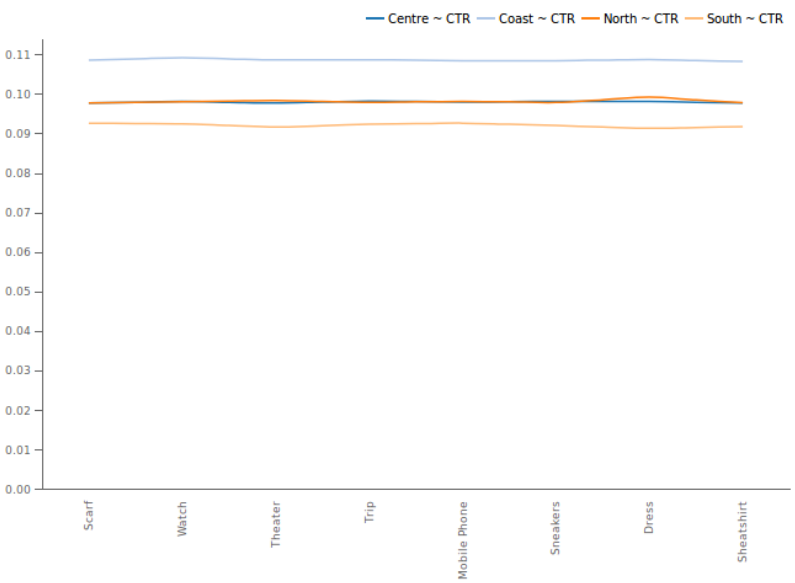


Figura 55: Relación entre producto y zona

La diferencia entre cada producto y la zona resulta mínima. En definitiva, también se descarta alguna relación entre un determinado producto y la zona.

¿Existe una relación entre la mejora de los indicadores de efectividad con algún segmento de la población objetivo?

Para responder a esta pregunta, lo primero a considerar será el rango de edad (archivo P2_a).

Rango de Edad	Prints	Hits
18-30	26.403.614	2.661.823
31-45	26.415.124	2.786.347
46-60	26.466.419	2.796.763
61-99	26.426.741	2.281.934

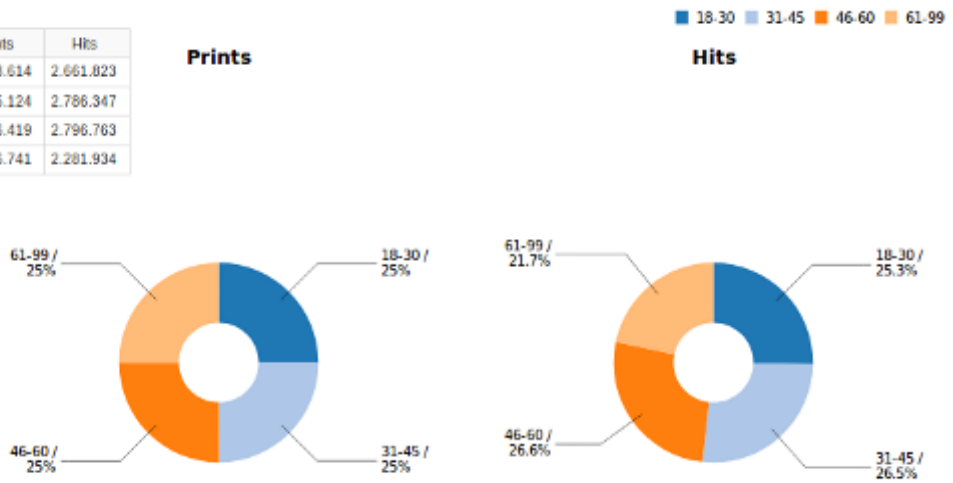


Figura 56: Relación con el rango de edad

El resultado no deja lugar a dudas. Se concluye que la edad no es un factor de mejora en los indicadores *Prints* y *Hits*.

Al enfocar el análisis en el indicador CTR (archivo P2_b), la mayor eficiencia se da en las personas de entre 31 y 60 años. En cambio, la peor tasa se da en el rango de mayor edad.

Rango de Edad	CTR
18-30	0,101
31-45	0,105
46-60	0,106
61-99	0,086

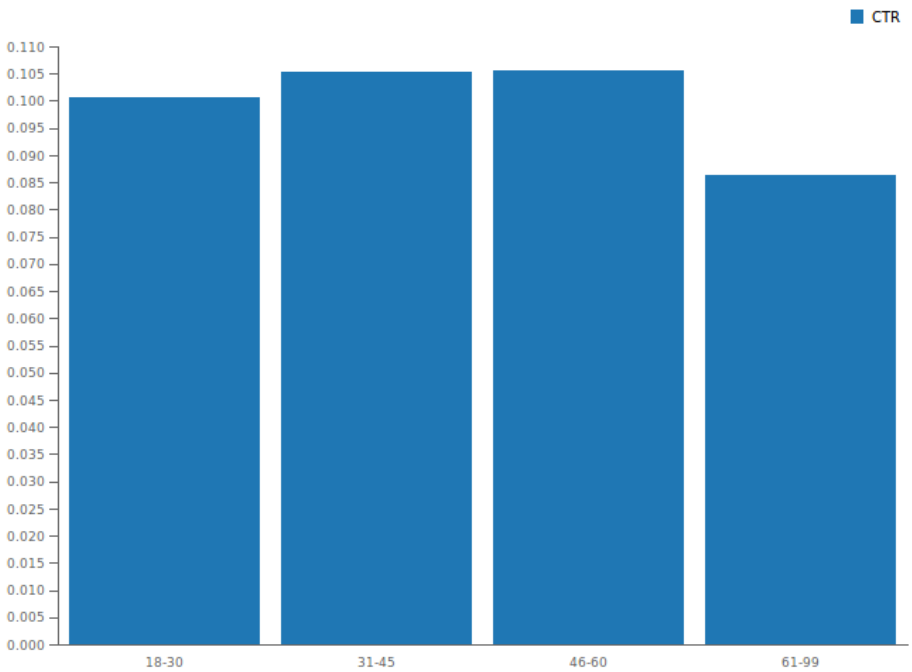


Figura 57: Relación con el rango de edad

La eficiencia no mejora respecto al género de las personas (archivo P2_c):

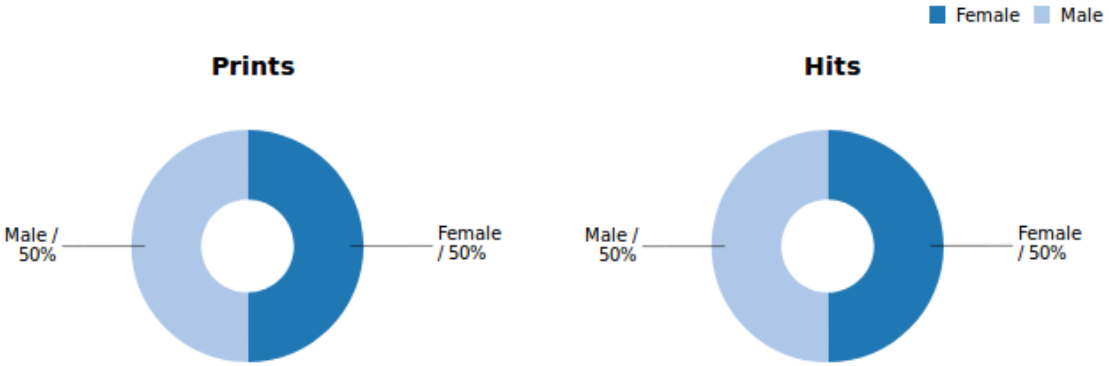


Figura 58: Relación con el genero

Sexo	CTR
Female	0,1
Male	0,1

Figura 59: Relación con el genero

Sin embargo, si se tienen en cuenta los intereses o gustos, la conclusión cambia radicalmente (archivo P2_d y archivo P2_e).

Interes	Prints	Hits
Business	13.254.628	1.004.188
Cars	13.211.002	1.409.463
Fashion	13.198.121	1.412.190
Garden	13.184.740	902.455
People	13.236.177	1.698.401
Sports	13.200.251	1.467.249
Technology	13.232.600	1.162.381
Travel	13.194.379	1.470.540

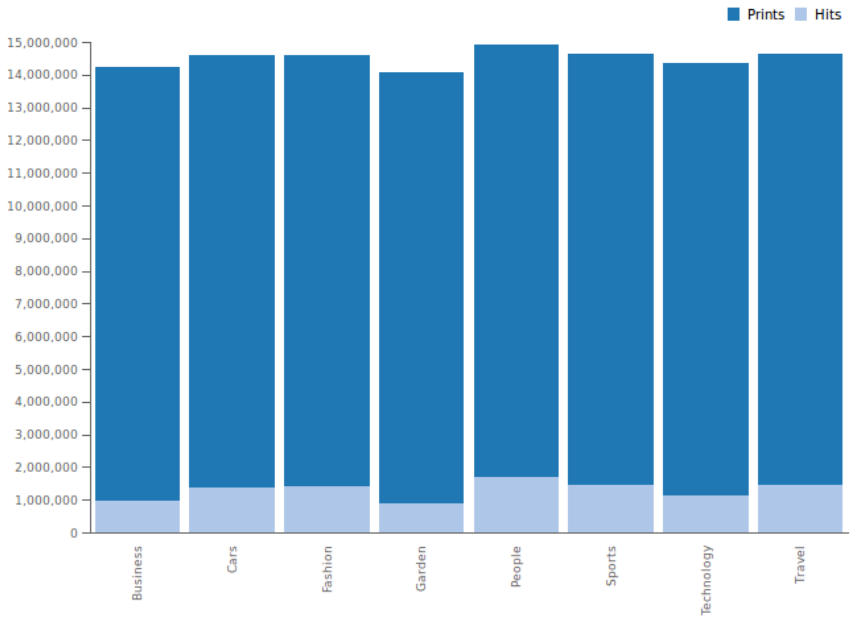


Figura 60: Relación con los intereses o gustos

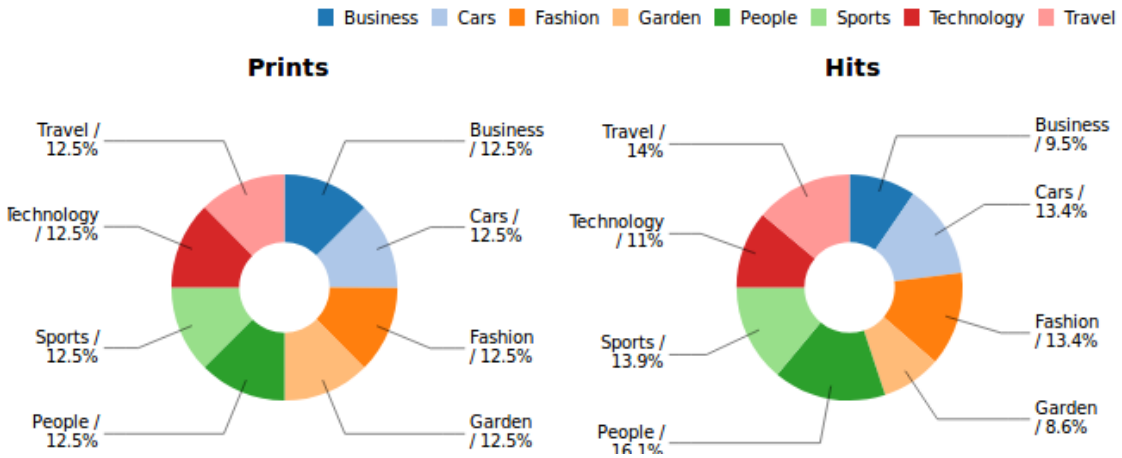


Figura 61: Relación con los intereses o gustos

Interes	CTR
Business	0,076
Cars	0,107
Fashion	0,107
Garden	0,068
People	0,128
Sports	0,111
Technology	0,088
Travel	0,111

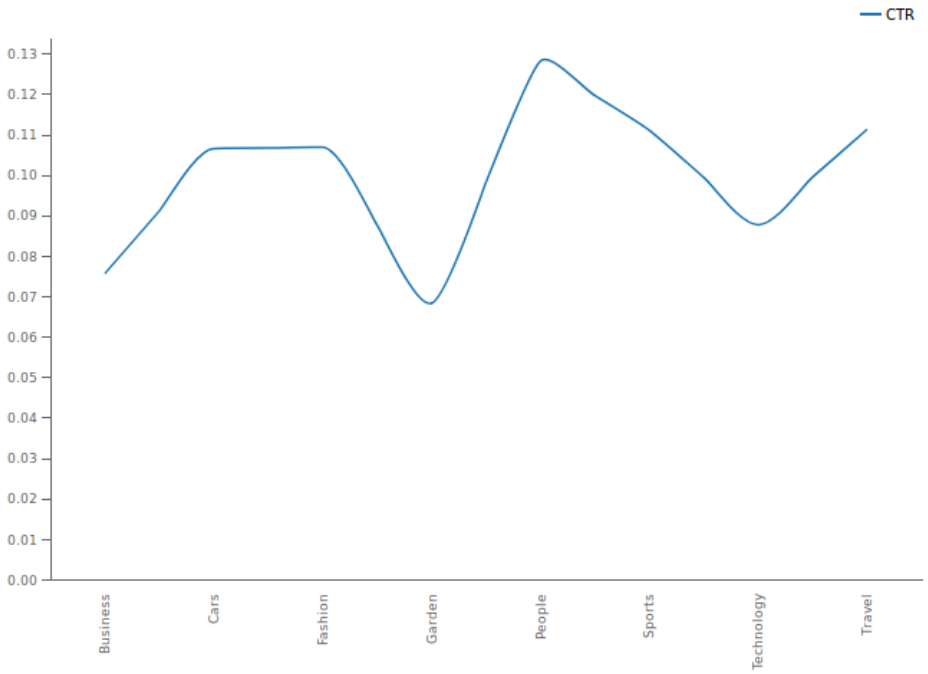


Figura 62: Relación con los intereses o gustos

Como se ven en los gráficos, hay una mejora en aquellas personas interesadas en People, seguida por Travel y Sports. Las que menos influencia tienen son aquellas a las que les gusta Garden.

¿Hay alguna plataforma donde, bajo las mismas condiciones, se obtengan mejores tasas de visualización?

Una primera consulta de análisis, únicamente de la red social (archivo P3_a), demuestra que no hay una plataforma con mejor tasa de visualización.

Red Social	Prints	Hits
Facebook	26.437.475	2.600.754
Instagram	26.444.417	2.662.052
Twitter	26.449.815	2.599.397
YouTube	26.380.191	2.664.664

Figura 63: Relación con redes sociales

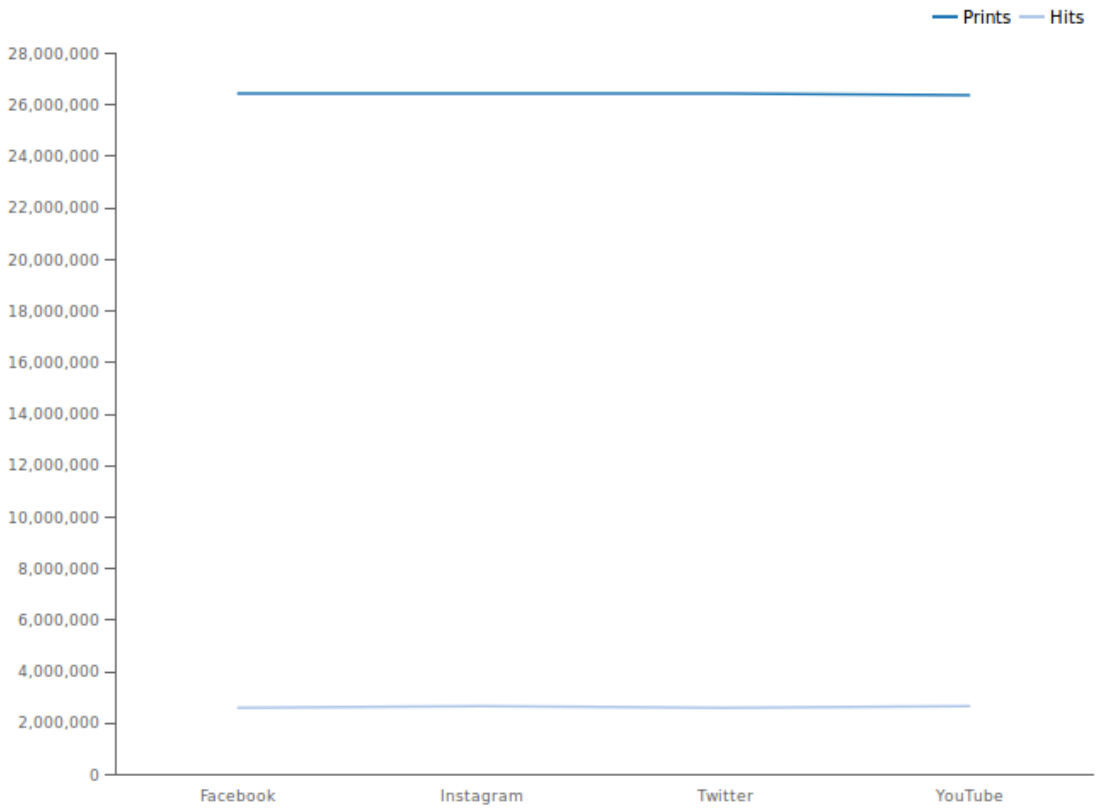


Figura 64: Relación con redes sociales

En contraposición, si se analiza el CTR, Instagram y YouTube presentan mejores tasas (archivo P3_b).

Red Social	CTR
Facebook	0,098
Instagram	0,101
Twitter	0,098
YouTube	0,101

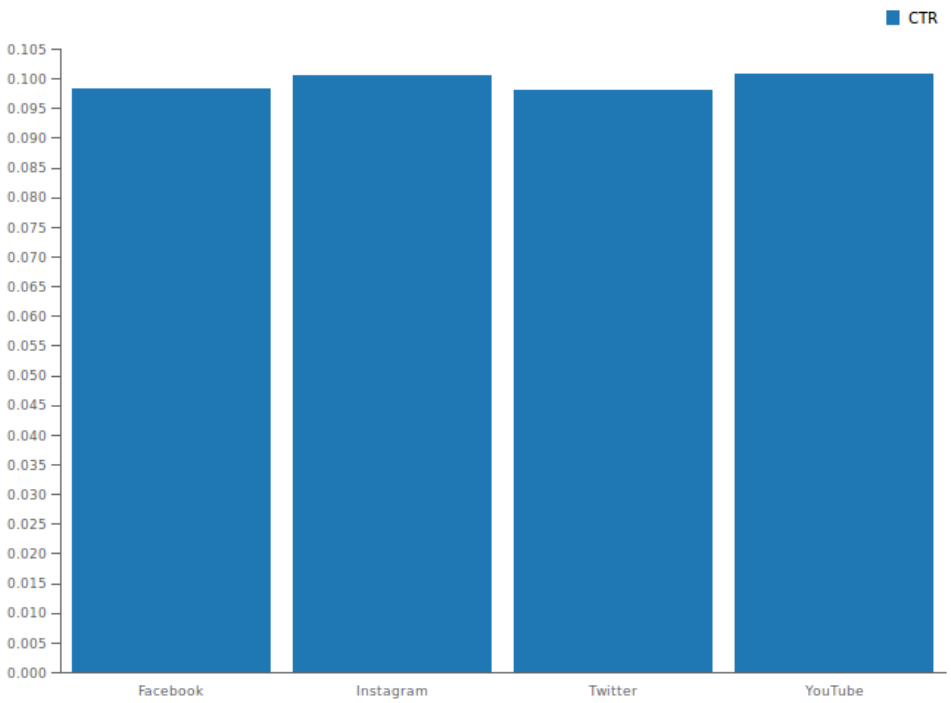


Figura 65: Relación con redes sociales

Para responder acabadamente la pregunta, habría que llevar a cabo un análisis más profundo. Por ejemplo, se podría ver si existe alguna red social preferida en alguna región (archivo P3_c). Analizando la cantidad

de clicks que se hicieron en las publicaciones, se concluye que en la zona costera y en el norte, las redes sociales con más hits son YouTube e Instagram. En el centro y en el sur, el orden es inverso: primero Instagram y luego YouTube. Cabe aclarar que las diferencias no son muy pronunciadas.

Red Social	Facebook	Instagram	Twitter	YouTube
Zona	Hits	Hits	Hits	Hits
Centre	617.927	634.092	618.983	633.851
Coast	772.533	790.521	768.013	793.199
North	610.571	623.387	610.652	627.112
South	599.723	614.052	601.749	610.502

Figura 66: Relación redes sociales y zonas

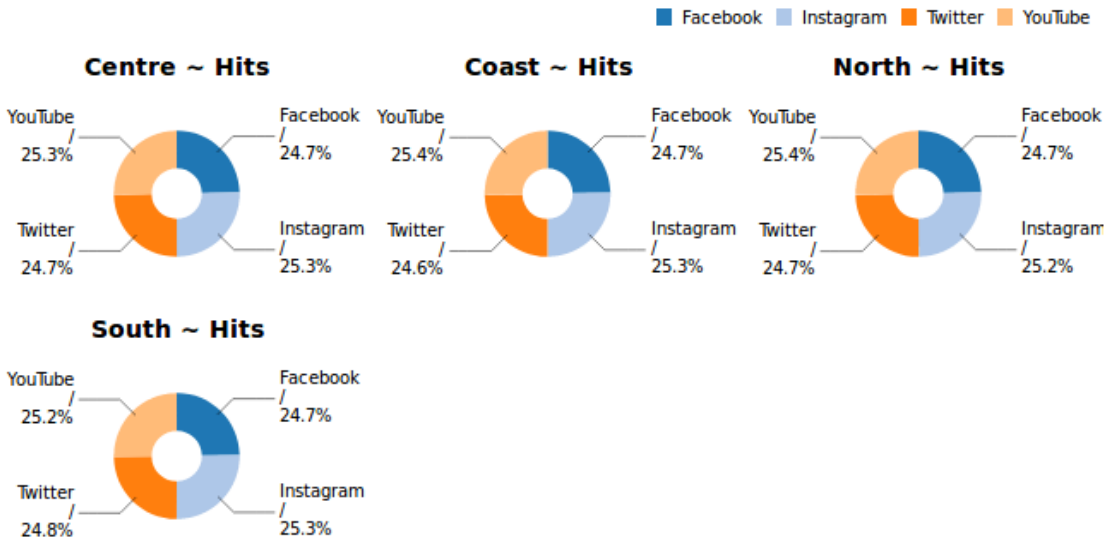


Figura 67: Relación redes sociales y zonas

En cuanto a la cantidad de visualizaciones, en ninguna zona hay predominio de alguna de las redes sociales.

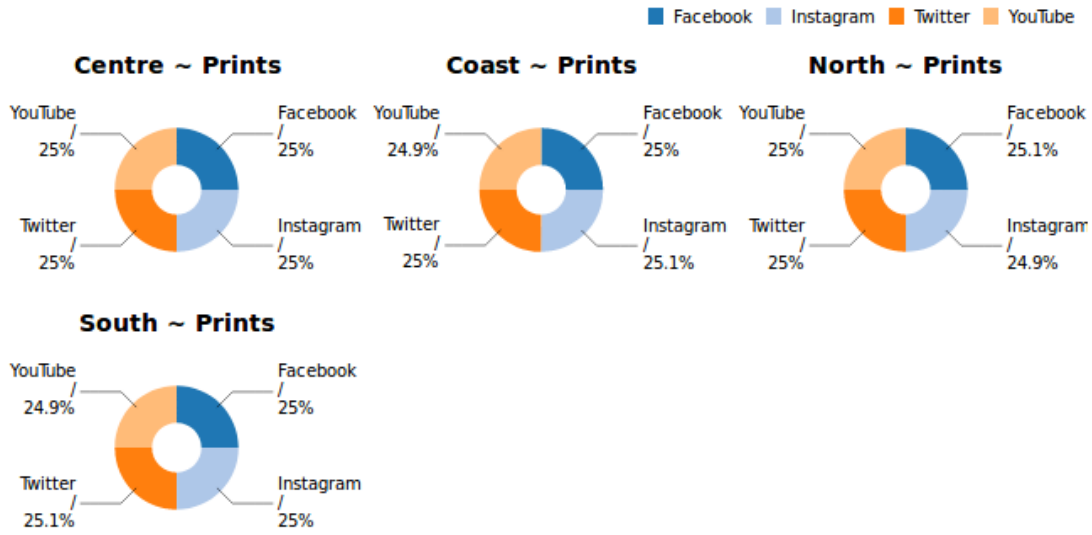


Figura 68: Relación redes sociales y zonas

Sin embargo, las redes sociales con mayor tasa de visualización son YouTube e Instagram, independientemente de la zona. La que menor tasa tiene es Twitter (archivo P3_d).

Red Social	Facebook	Instagram	Twitter	YouTube
Zona	CTR	CTR	CTR	CTR
Centre	0,097	0,099	0,097	0,099
Coast	0,108	0,11	0,107	0,111
North	0,097	0,099	0,097	0,1
South	0,091	0,093	0,091	0,093

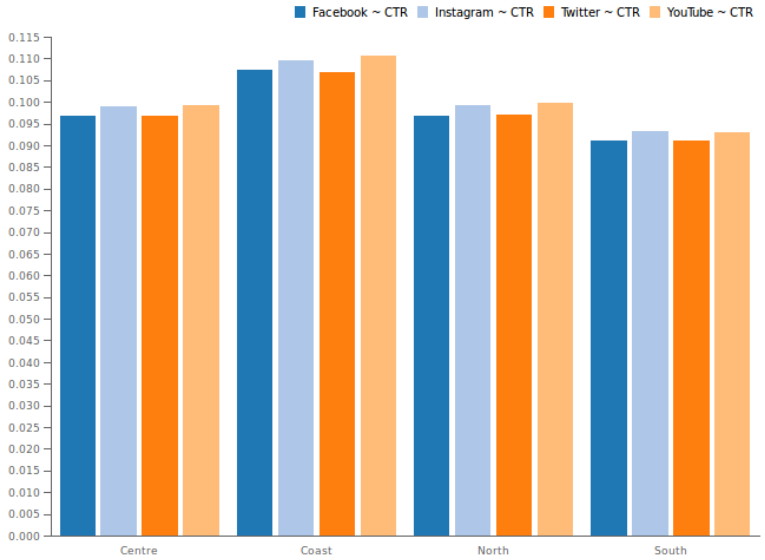


Figura 69: Relación redes sociales y zonas

Al analizar las familias de productos (archivo P3_e), la mayor cantidad de visualizaciones se da en las familias Wear, Accessory y Culture, sin que se note diferencias en cuanto a la red social donde se publicite.

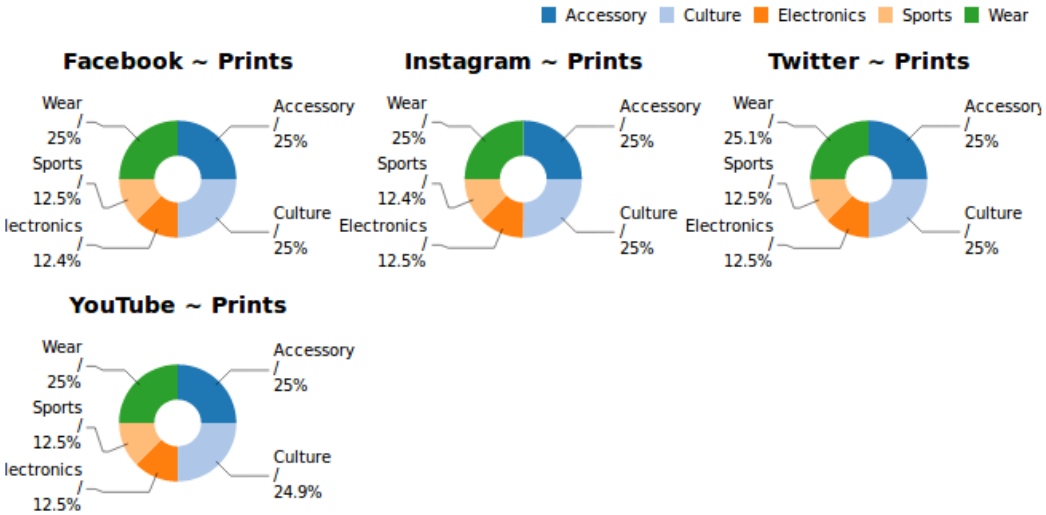


Figura 70 : Relación de prints entre redes sociales y familias

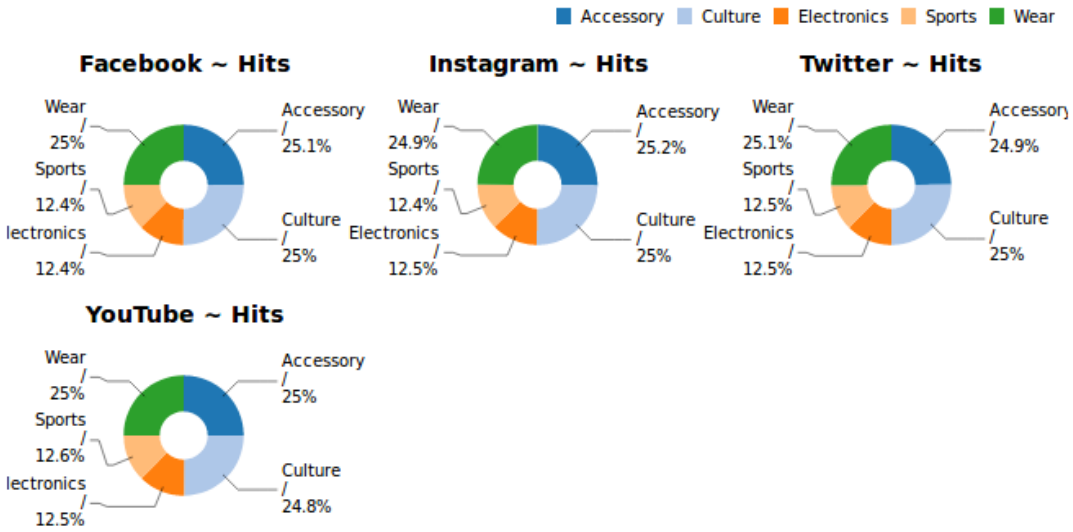


Figura 71: Relación redes sociales y familias

	Facebook	Instagram	Twitter	YouTube
Familia	CTR	CTR	CTR	CTR
Accessory	0,099	0,101	0,098	0,101
Culture	0,098	0,101	0,098	0,101
Electronics	0,098	0,101	0,098	0,101
Sports	0,098	0,101	0,098	0,102
Wear	0,098	0,1	0,098	0,101

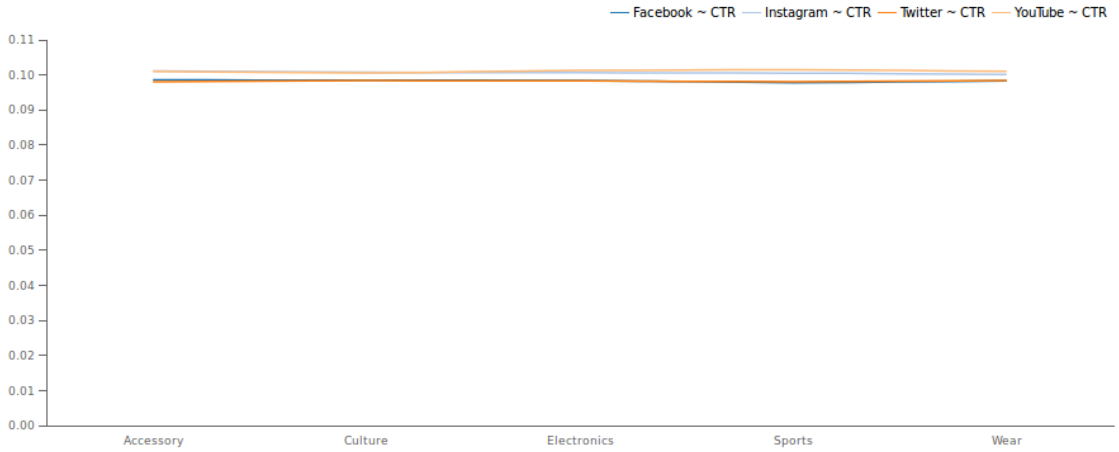


Figura 72: Relación redes sociales y familias

El análisis se vuelve más interesante al considerar cada trimestre del año (archivo P3_f y P3_g). A partir del segundo trimestre, pronunciándose aún más en el tercero, se ve una clara disminución en los indicadores. Se puede concluir que es más efectiva la campaña en cualquier red social en el primer trimestre. Se destaca Instagram y luego, Youtube.

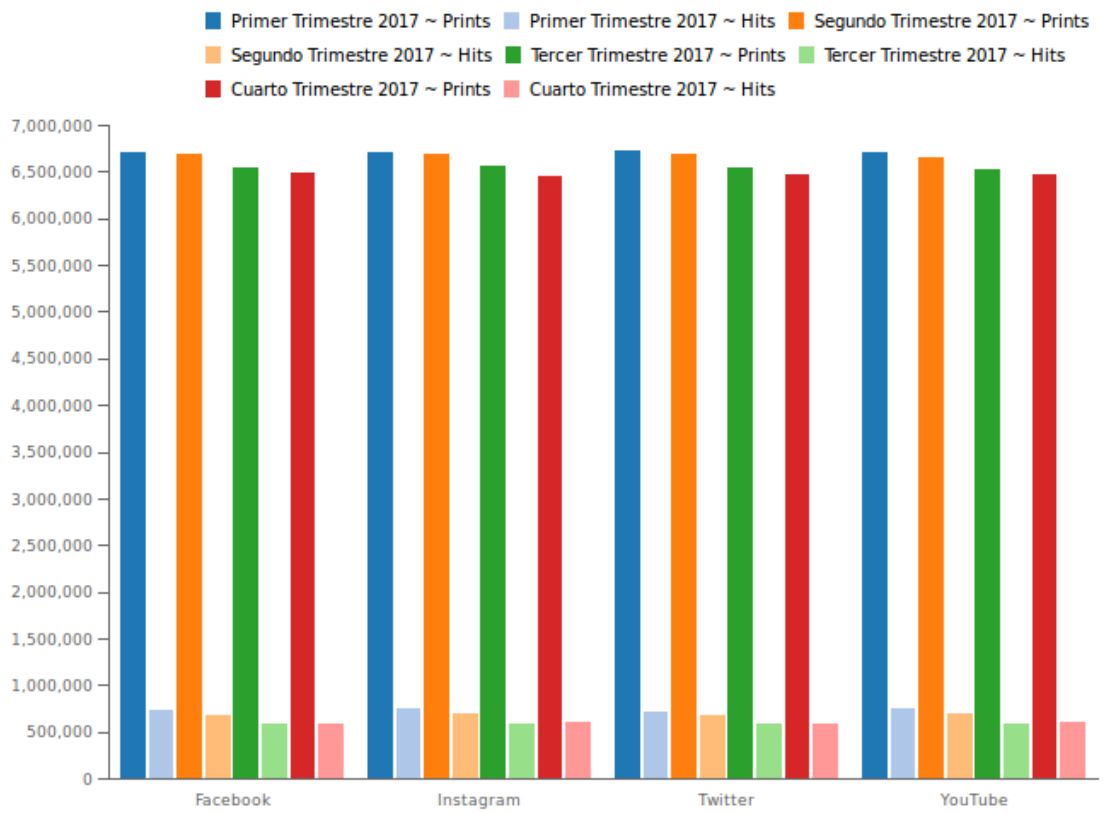


Figura 73: Relación redes sociales y trimestres

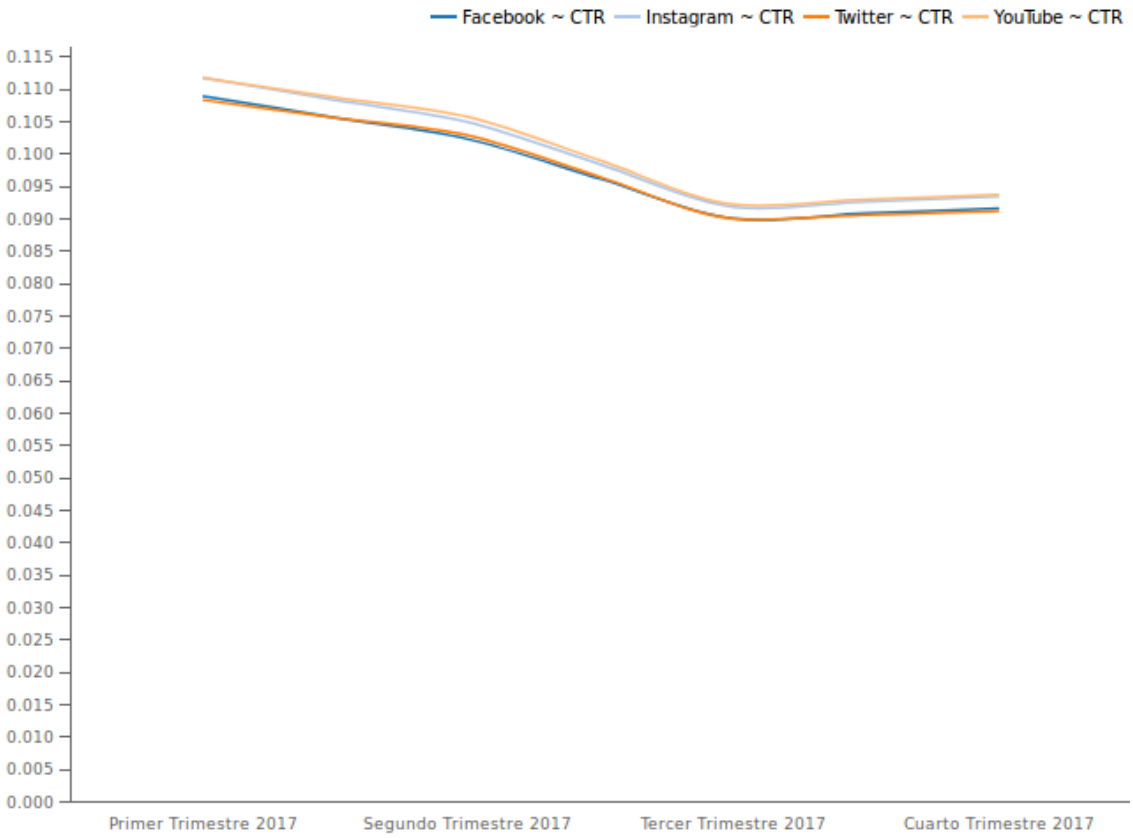


Figura 74: Relación redes sociales y trimestres

De todo lo dicho, la conclusión más contundente es que las dos plataformas con mejores resultados son Instagram y YouTube.

¿Existen relaciones entre plataformas y franjas de edad de usuarios que provoquen mejores tasas de visualización?

Para responder esta pregunta se realizó una consulta teniendo en cuenta la dimensión red social y rango de edad (archivo P4).

Rango de Edad	18-30	31-45	46-60	61-99
Red Social	CTR	CTR	CTR	CTR
Facebook	0,091	0,101	0,111	0,091
Instagram	0,11	0,11	0,101	0,081
Twitter	0,091	0,101	0,11	0,091
YouTube	0,111	0,11	0,101	0,082

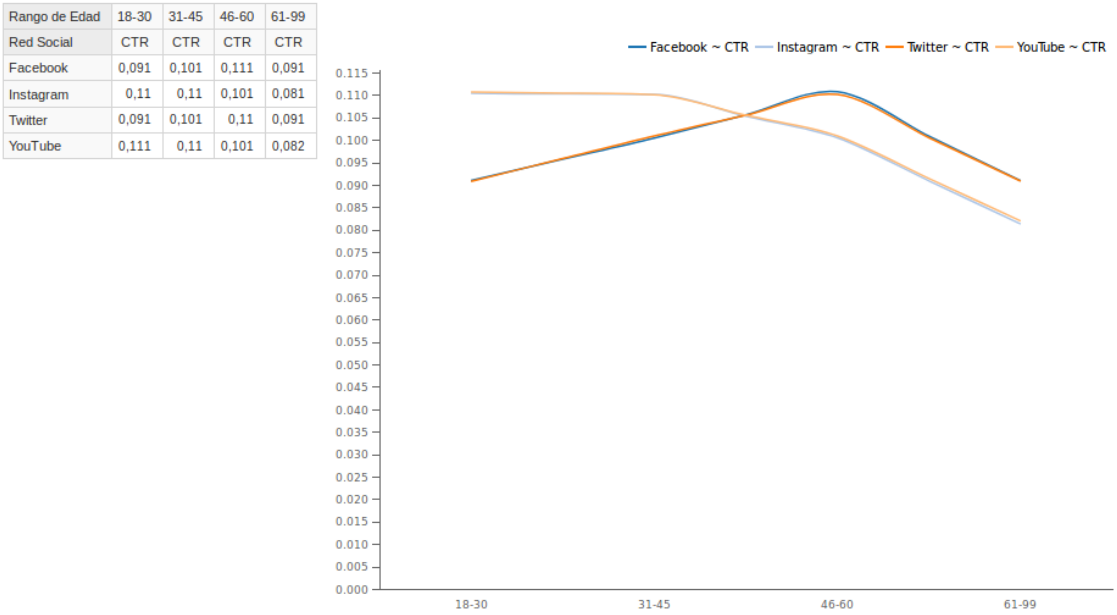


Figura 75: Relación redes sociales y rango de edad

De este gráfico se desprende que hay dos grupos de plataformas bien difrenciadas en cuanto a la tasa de visualización de las publicidades: por un lado, Facebook y Twitter y, por el otro, YouTube e Instagram. Para las personas entre 18 y 45 años tienen mejor resultado YouTube e Instagram y menor eficiencia Facebook y Twitter, tendencia que se invierte en las de mayor edad. Sin embargo, la diferencia más pronunciada entre estos dos conjuntos se da en el rango de 18-30 disminuyendo en el resto de las edades. Se concluye que hay una relación más que evidente entre las redes sociales y los rango de edades de los interesados.

¿El conocimiento del grupo de interés de los usuarios podría ayudar a mejorar los indicadores para determinados productos?

A priori, la sospecha es que exista una relación directa entre el interés y los productos. Para constatar si efectivamente es así, se lanza una consulta con agrupamiento de producto e intereses, primero teniendo en cuenta los *prints* (archivo P5_a), luego los *hits* (archivo P5_b) y por último, los CTR (archivo P5_c).

Prints:

Interes	Business	Cars	Fashion	Garden	People	Sports	Technology	Travel
Producto	Prints	Prints	Prints	Prints	Prints	Prints	Prints	Prints
Scarf	1.633.473	1.662.592	1.640.312	1.642.956	1.682.332	1.658.947	1.665.373	1.640.783
Watch	1.672.207	1.657.493	1.641.080	1.648.795	1.653.725	1.644.846	1.644.536	1.655.910
Theater	1.665.278	1.641.905	1.653.459	1.653.897	1.645.036	1.651.863	1.658.014	1.642.365
Trip	1.624.760	1.635.316	1.664.587	1.665.922	1.659.029	1.664.690	1.642.005	1.653.469
Mobile Phone	1.650.153	1.634.284	1.642.881	1.629.963	1.654.513	1.639.332	1.673.267	1.663.232
Sneakers	1.653.450	1.655.450	1.646.675	1.651.052	1.644.852	1.650.789	1.652.728	1.647.483
Dress	1.663.391	1.670.681	1.663.684	1.629.207	1.649.633	1.656.964	1.646.666	1.649.433
Sheatshirt	1.691.916	1.653.281	1.645.443	1.662.948	1.647.057	1.632.820	1.650.011	1.641.704

Figura 76: Relación producto e intereses

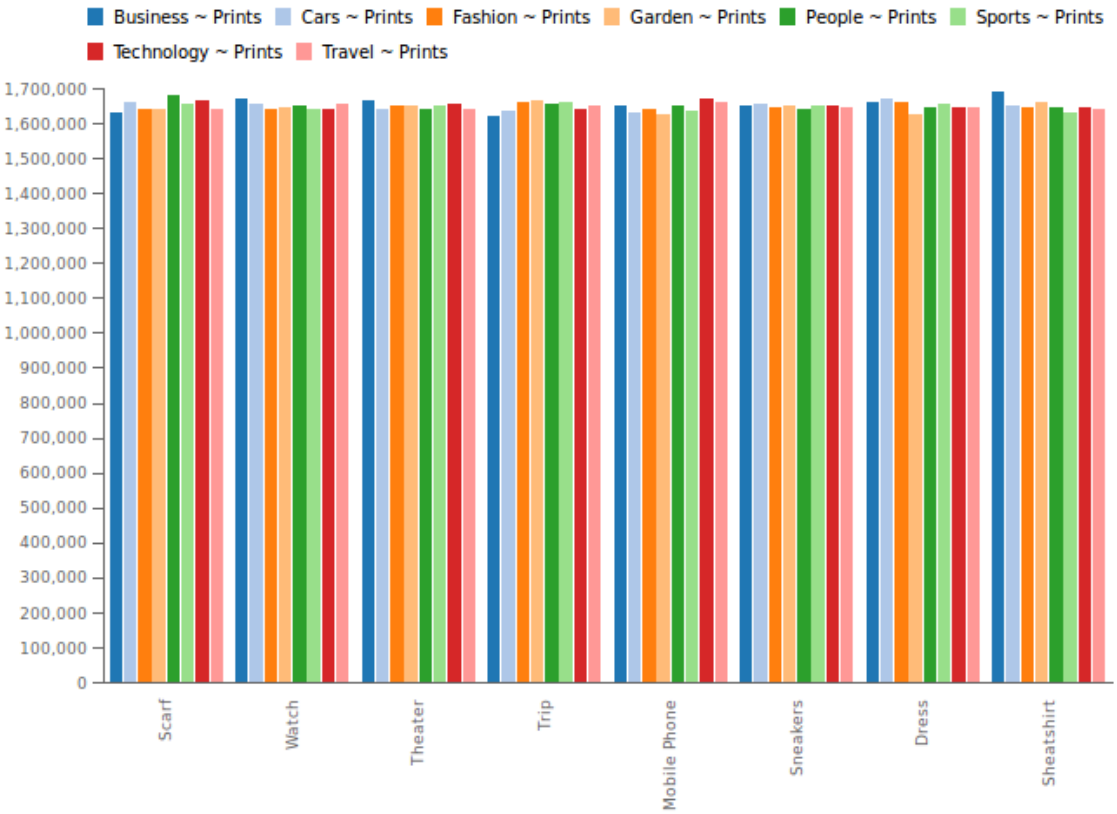


Figura 77: Relación producto e intereses

Hits:

Interes	Business	Cars	Fashion	Garden	People	Sports	Technology	Travel
Producto	Hits	Hits	Hits	Hits	Hits	Hits	Hits	Hits
Scarf	96.454	97.914	410.707	159.337	259.772	97.107	98.862	96.745
Watch	258.908	415.030	96.737	97.116	160.732	96.997	96.829	97.474
Theater	161.427	96.522	97.471	97.253	412.015	97.326	98.054	253.924
Trip	95.637	96.274	98.366	161.733	256.725	98.420	96.430	412.629
Mobile Phone	97.169	251.741	96.857	95.952	97.571	159.008	417.569	98.580
Sneakers	97.248	97.577	97.604	97.388	254.688	412.286	96.838	159.808
Dress	97.867	98.715	417.470	95.755	159.942	97.504	97.371	254.647
Sheatshirt	99.478	255.690	96.978	97.921	96.956	408.601	160.428	96.733

Figura 78: Relación producto e intereses

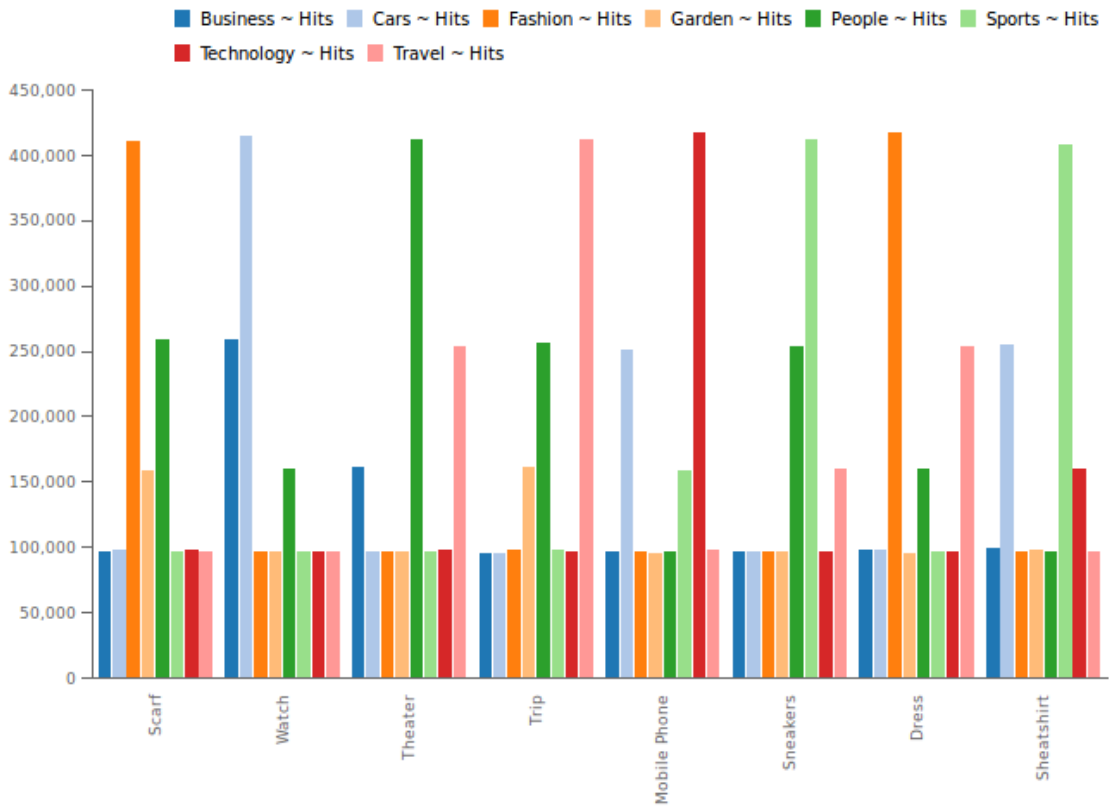


Figura 79: Relación producto e intereses

CTR:

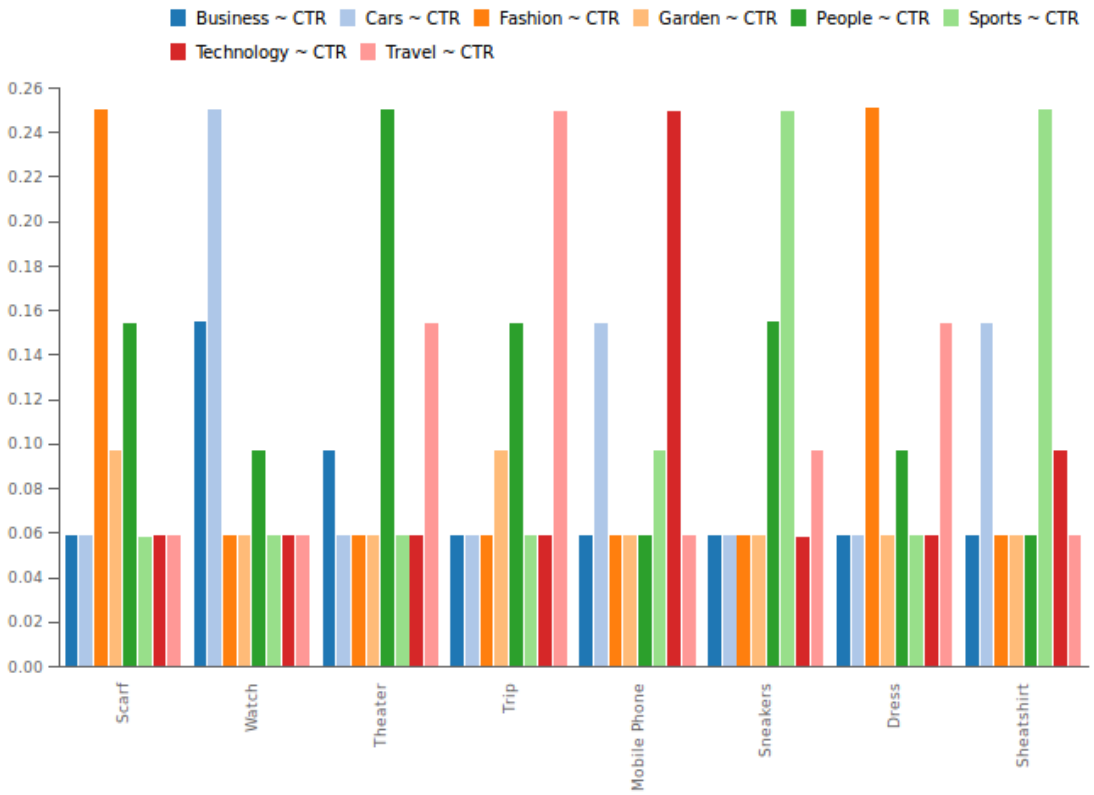


Figura 80: Relación producto e intereses

Analizando las visualizaciones (*prints*), se pueden detectar algunos productos que se destacan sobre el resto; pero no de manera tan significativa como en el caso de los números de clicks.
Los usuarios que más ingresan a las publicidades de Scarf y Dress están interesados en Fashion. Aquellas relacionadas con los teléfonos móviles son visitadas por quienes se interesan en tecnología y autos. Para mayor detalle, se adjunta la siguiente imagen.

Figura 81: Relación producto e intereses

Como era de esperar, se confirma la hipótesis de que existe una notable relación entre el grupo de interés de los usuarios con la publicidad de algunos productos, sobre todo de aquellos que ingresan a la misma.

Otras preguntas:

Al momento del análisis surgieron otras preguntas que en un principio no fueron formuladas: ¿existe alguna relación entre producto o familia de productos con el mes de publicación? ¿Influye el rango de edad en la selección de la familia de productos? Estos cuestionamientos podrían ser de utilidad para el negocio, por ejemplo, a la hora de establecer períodos de descuento.

El sistema de BI implementado bien puede dar respuestas a estas preguntas sin necesidad de agregar nuevas dimensiones ni de realizar cambios significativos en su diseño.

¿Existe alguna relación entre producto o familia de productos con el mes de publicación?

Para dar respuesta a este planteo, se realiza una nueva consulta (archivo P6) obteniendo los siguientes resultados:

Familia	Accessory	Culture	Electronics	Sports	Wear
Mes	CTR	CTR	CTR	CTR	CTR
Enero	0,11	0,11	0,11	0,111	0,11
Febrero	0,11	0,109	0,111	0,111	0,11
Marzo	0,112	0,111	0,109	0,11	0,11
Abril	0,11	0,111	0,11	0,111	0,11
Mayo	0,109	0,109	0,11	0,109	0,11
Junio	0,092	0,091	0,092	0,091	0,091
Julio	0,091	0,091	0,091	0,091	0,091
Agosto	0,091	0,091	0,092	0,091	0,091
Septiembre	0,091	0,091	0,091	0,091	0,091
Octubre	0,092	0,091	0,092	0,091	0,091
Noviembre	0,091	0,091	0,092	0,091	0,09
Diciembre	0,096	0,095	0,096	0,095	0,096

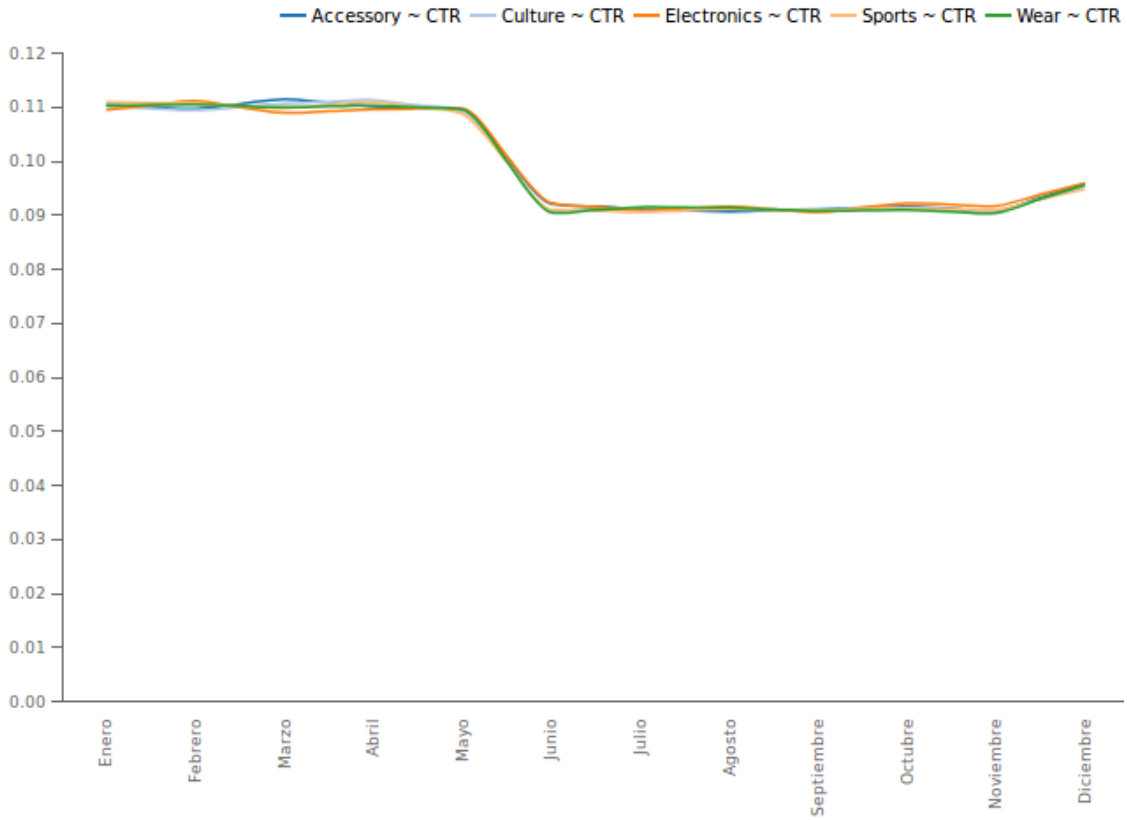


Figura 82: Relación familia y meses

Las tasas de visualizaciones son casi constantes en los primeros cinco meses del año sin depender de las familias de productos. Hay una clara disminución entre Junio y Noviembre, mejorando en el mes de Diciembre. En el mes de Marzo resalta un incremento en los CTR de Accessory.

¿Influye el rango de edad en la selección de la familia de productos?

La consulta (archivo P7) en este caso arroja los siguientes resultados:

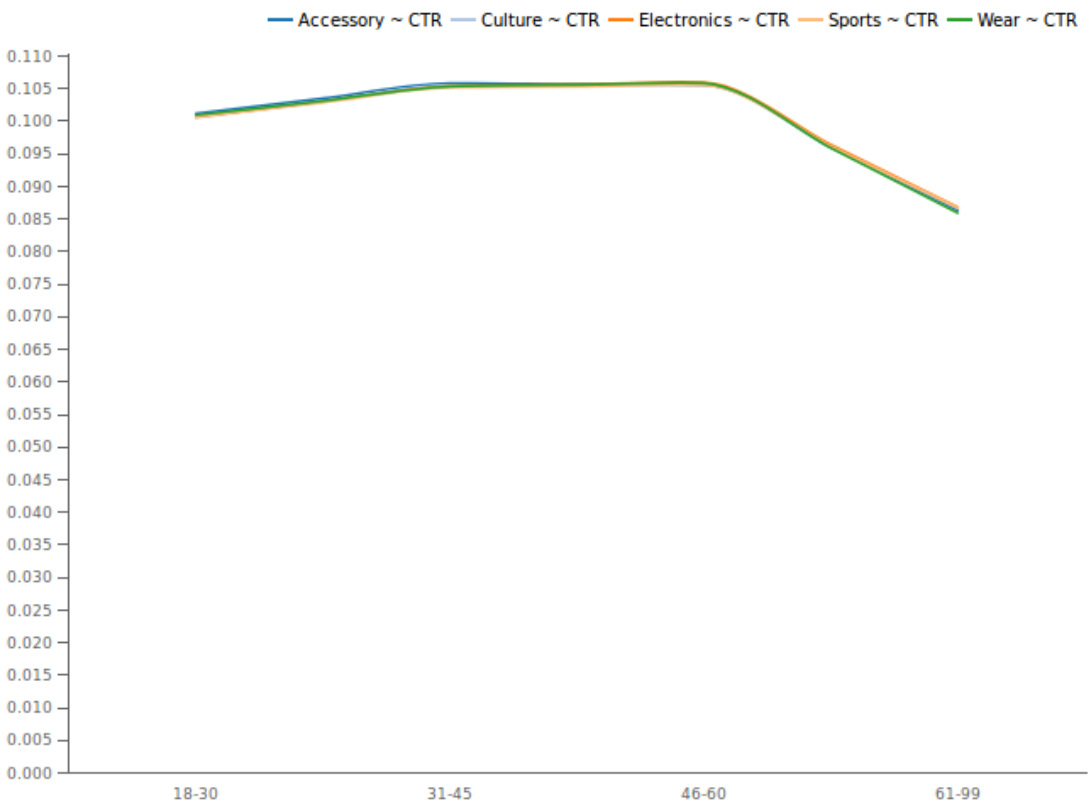


Figura 83: Relación rango de edad y productos

El rango de edad no modifica la tasa de visualización de la familia del producto.

Conclusiones del análisis

Se enumeran las siguientes conclusiones:

- La zona de la costa presenta mejores indicadores que el resto de las regiones, destacándose la ciudad de Barcelona.
- Teniendo en cuenta las zonas geográficas, no se observa una mejora ni en la familia de productos ni en los productos tomados individualmente.
- La mayor eficiencia de los anuncios se da en el rango de edad entre 31 y 60 años. En cambio, el género no altera los resultados.
- Existe relación entre el interés del grupo de usuario y la mejora de los indicadores. Se destacan los grupos de intereses People, Sports y Travel.
- Las redes sociales que presentan mejor tasa de visualización son YouTube e Instagram.
- Al centrar la atención en las familias de productos, no hay diferencia en cuanto a la red social donde se publique.
- En cualquier red social, en el primer trimestre, la campaña es más efectiva. Se destaca la de Instagram seguida por la de Youtube.

- Para las personas entre 18 y 45 años tienen mejor resultado YouTube e Instagram y menos eficiencia en Facebook y Twitter. La tendencia se invierte en las de mayor edad. Sin embargo, la diferencia más pronunciada entre estos dos conjuntos se da en el rango de 18-30 años, disminuyendo en el resto de las edades.
- Existe una relación directa entre el grupo de interés de los usuarios y la publicidad de algunos productos.
- Para todas las familias, la tasa de visualización es casi constante en los primeros cinco meses del año disminuyendo entre Junio y Noviembre. Tiene un leve incremento en Diciembre.
- La proporción de clicks no se modifica según el rango de edad.

6. Conclusiones

Como objetivo general del presente trabajo se planteó diseñar e implementar un sistema de *Business Intelligence* que, a partir de los datos proporcionados, permitiera evaluar el impacto de las campañas publicitarias en cuatro redes sociales. Tal finalidad se alcanzó gracias a la implementación de un d*ata* w*arehouse* que mediante procesos ETL almacenaron los datos de los ficheros base.

Luego de estudiar las diferentes plataformas BI *Open Source* disponibles en el mercado, fue seleccionada Pentaho. Esta elección resultó muy satisfactoria debido a la gran cantidad de foros y video tutoriales que existen y a su comunidad de usuarios que da soporte. La mayor dificultad afrontada fue la falta de experiencia en el uso de las herramientas para proyectos de este tipo, subsanada gracias a los módulos y plugins de Pentaho que hacen más intuitivo el proceso.

A pesar de no haber sido una meta específica, este trabajo y mayormente sus anexos pueden servir como base para personas que no tengan conocimiento y quieran utilizar esta suite para algún proyecto de BI.

Al inicio del TFM se creía que para implementar el sistema BI se debía desarrollar una determinada aplicación cuando, en realidad, dicho sistema es el resultado de la conjunción de la infraestructura, herramientas e informes que permiten analizar la información con el fin de mejorar y optimizar las decisiones del negocio.

Otro desafío fue cumplir con la planificación. Por problemas laborales y para no retrasar los plazos establecidos, se tuvieron que ajustar la cantidad de horas previstas y trabajar en días no contemplados anteriormente. A pesar de este inconveniente, se pudieron cumplir con las fechas y entregables definidos en cada fase.

El hecho de haber seleccionado la metodología de Kimball que prácticamente nació para el diseño de este tipo de sistemas favoreció la consecución de los objetivos planteados. Además, cada uno de sus libros ayudaron a subsanar posibles errores y a resolver rápidamente eventuales problemas.

Sobre todo, a la hora de diseñar el data warehouse, un punto a destacar es lo difícil que resulta pensar en términos de dimensiones dejando de lado la 3° forma normal. Sin embargo, una vez logrado este cambio de paradigma, se hace mucho más sencillo tanto la creación de los procesos ETL como el diseño de las consultas al almacén de datos.

Como premisas del trabajo se propusieron una serie de preguntas que fueron el motor para la creación de un cubo OLAP. Este análisis multidimensional posibilitó no sólo responder aquellos primeros interrogantes sino también otros que surgieron a lo largo del proceso, sin necesidad de realizar ningún cambio en el sistema planteado. La robustez y la flexibilidad del diseño elegido quedaron notoriamente demostradas.

Como líneas de trabajo futuro se podrían señalar varias posibilidades. En primer lugar, plantear la elaboración de procesos ETL que contemplen actualizaciones y modificaciones en los datos. El diseño del DW deja abierta la posibilidad de crear otros cubos OLAP. Asimismo,

cabría diseñar un cuadro de mando que permitiera a los usuarios consultar datos a demanda. En tercer lugar, sería factible realizar tareas de minería de datos. A través de un análisis predictivo, se podrán establecer reglas y detectar patrones que anticipen el comportamiento de la audiencia objetivo haciendo más eficiente los anuncios publicitarios.

El mayor fruto del presente trabajo, llevado a cabo con tanta satisfacción, fue la profundización sobre esta temática tan prometedora para el futuro que seguramente redundará en beneficios y oportunidades profesionales.

7. Glosario

ad hoc	Significa “para esto”. Se refiere a una solución específicamente elaborada para un problema o fin preciso.
BI	Acrónimo en inglés para referirse a *Business Intelligence*.
Business Intelligence	Significa “inteligencia de negocio”.
CTR	Acrónimo en inglés de tasa o proporción de clicks (*Click Trough Ratio*).
Data Warehouse	Repositorio de datos que proporciona una visión global, común e integrada de los datos de la organización.
DWH	Acrónimo en inglés de *data warehouse*
ETL	Acrónimo en inglés de extraer, transformar y cargar (*Extract*, *Transform and Load*).
MDX	Acrónimo en inglés de *multidimensional expression* o, también, *multidimensional query expression.* Lenguaje de consulta de estructuras OLAP.
OLAP	Acrónimo en inglés de procesamiento analítico en línea (*On-Line Analytical Proceessing*).
Social media	Medios de comunicación sociales.
Drag and drop	Significa “arrastrar y soltar”.
PDI	Acrónimo en inglés de Pentaho Data Integration.

8. Bibliografía

[1]<https://unioninformatica.org/institucional/convenio-colectivo-de-trabajo/#EscalaSalarial>. [Fecha de consulta: 02/03/2018].

[2]<http://www.cessi.org.ar/perfilesit/detalle-de-consultor-bi-business-intelligence-5>. [Fecha de consulta: 02/03/2018].

[3]<http://informatica.blogs.uoc.edu/2016/02/04/especialistas-en-business-intelligence-los-que-mas-ganan/>. [Fecha de consulta: 02/03/2018].

[4]<http://www1.ucasal.edu.ar/htm/ingenieria/cuadernos/archivos/5-p56-rivadera-formateado.pdf>. [Fecha de consulta: 03/03/2018].

[5] **Curto Díaz, J.** (2010). *Introducción al Business Intelligence*. Editorial UOC.

[6] **Kimball, R.; Ross, M.** (2013). *The Data Warehouse Toolkit* (3.ª ed.). Indianapolis: Wiley & Sons.

[7] **Kimball, R.** (2008). *The Data Warehouse Lifecycle Toolkit* (2.ª ed.). Indianapolis: Wiley & Sons.

[8] **Kimball, R.; Caserta, J.** (2004). *The Data Warehouse ETL Toolkit* (2.ª ed.). Indianapolis: Wiley Publishing, Inc.

[9]<http://www.pentaho.com>. [Fecha de consulta: 03/03/2018]

[10]<https://wiki.pentaho.com/display/COM/Community+Wiki+Home>. [Fecha de consulta: 03/03/2018]

[11]< https://sourceforge.net/projects/pentaho/files/>. [Fecha de consulta: 03/03/2018]

[12]<http://www.eclipse.org/birt/>. [Fecha de consulta: 03/03/2018]

[13]<http://www.exforsys.com/tutorials/msas/data-warehouse-design-kimball-vs-inmon.html>.[Fecha de consulta:03/03/2018]

[14]< http://www.dataprix.com/blog-it/business-intelligence/comparativa-costes-adquisicion-mantenimiento-plataformas-bi>.[Fecha de consulta: 07/03/2018]

[15]<http://www.dataprix.com/blog-it/business-intelligence/comparativa-costes-adquisicion-mantenimiento-plataformas-bi>.[Fecha de consulta: 07/03/2018]

[16]<https://www.captio.net/blog/en-que-consiste-un-proyecto-de-business-intelligence-bi>.[Fecha de consulta: 07/03/2018]

[17]<https://blog.capterra.com/top-8-free-and-open-source-business-intelligence-software/>.[Fecha de consulta: 07/03/2018]

[18]<http://rollupconsulting.com/cuadrante-magico-gartner-bi-2017/>.[Fecha de consulta: 07/03/2018]

[19]< https://www.spagobi.org/>. [Fecha de consulta: 07/03/2018]

[20]<http://reader.digitalbooks.pro/content/preview/books/43005/book/OEBPS/chapter02.xhtml >. [Fecha de consulta: 29/03/2018]

[21]<https://es.linkedin.com/pulse/kimball-e-inmon-y-el-dise%C3%B1o-de-data-warehouses-william-qui%C3%B1onez>.[Fecha de consulta: 07/04/2018]

[22]<http://www.interaktiv.cl/blog/wp-content/uploads/2012/04/4.-Metodologia_disegno_DW1.pdf>.[Fecha de consulta: 07/04/2018]

[23]<https://www.betterbuys.com/bi/reviews/pentaho-business-intelligence/>. [Fecha de consulta: 27/04/2018]

[24] **Trujillo, J. C.** (2013). *Diseño y explotación de almacenes de datos: conceptos básicos de modelado multidimensional.* Alicante: ECU.

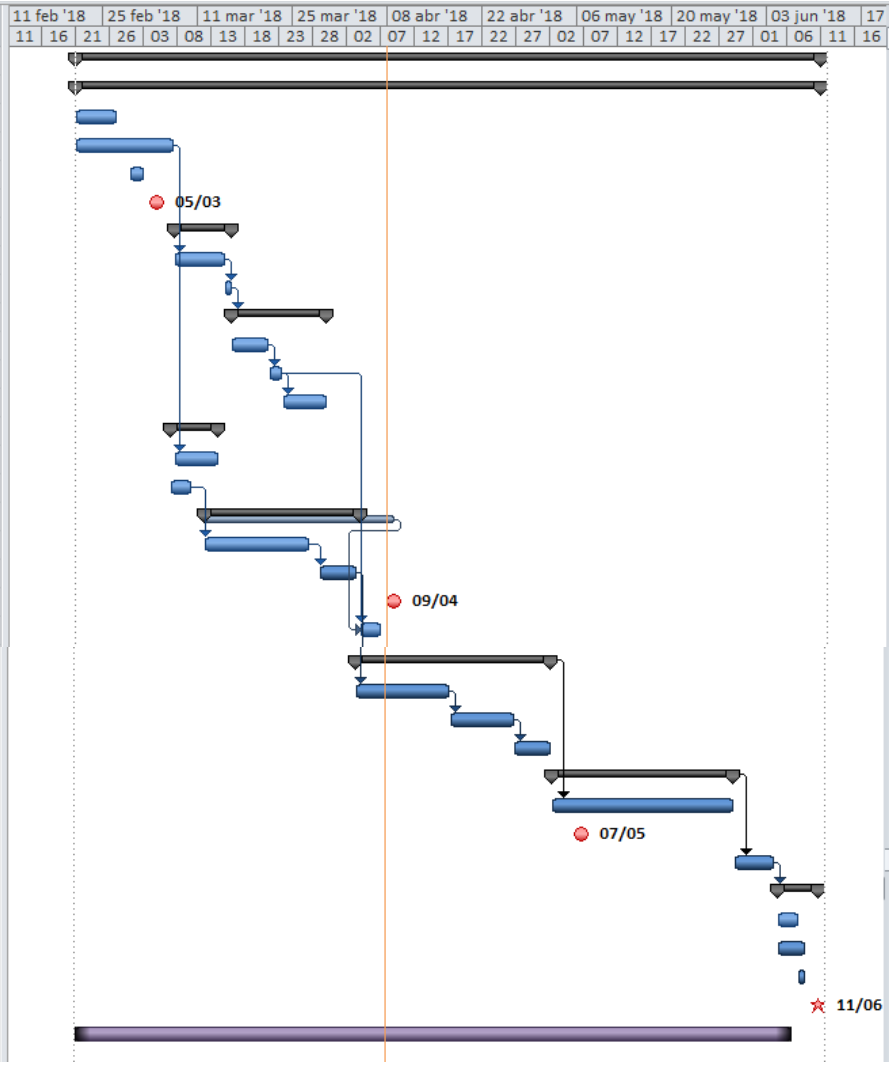
[25]<http://blog.bi-geek.com/arquitectura-enfoque-de-william-h-inmon/>. [Fecha de consulta: 27/04/2018]
[26]<http://tdan.com/data-warehouse-design-inmon-versus-kimball/20300>.[Fecha de consulta: 27/04/2018]
[27]<https://churriwifi.wordpress.com/2010/06/01/comparativa-talend-vs-kettle-pdi/>. [Fecha de consulta: 28/04/2018]
[28]<https://blog.es.logicalis.com/analytics/cubos-olap-y-estructuras-multidimensionales-todo-lo-que-hay-que-saber>. [Fecha de consulta: 23/05/2018]

9. Anexos

Anexo 1: Planificación: Diagrama de Gantt

Diagrama de Gantt

	Nombre de tarea	Duración	Comienzo	Fin
1	**Sistema de BI**	**70 días**	**mié 21/02/18**	**lun 11/06/18**
2	**1. Planificación del Proyecto**	**70 días**	**mié 21/02/18**	**lun 11/06/18**
3	1.1 Recopilación de documentación y bibliografía	4 días	mié 21/02/18	lun 26/02/18
4	1.2 Lectura de la documentación y bibliografía	10 días	mié 21/02/18	mié 07/03/18
5	1.3 Planificación del proyecto	2 días	jue 01/03/18	vie 02/03/18
6	1.4 Entrega PEC 1: Planificación	0 días	lun 05/03/18	lun 05/03/18
7	**2. Estudio y Selección de Herramientas BI**	**6 días**	**mié 07/03/18**	**jue 15/03/18**
8	2.1 Investigación de Herramientas	5 días	mié 07/03/18	mié 14/03/18
9	2.2 Selección de la herramienta	1 día	jue 15/03/18	jue 15/03/18
10	**3. Instalación del entorno de trabajo**	**9 días**	**vie 16/03/18**	**jue 29/03/18**
11	3.1 Instalación del Servidor	3 días	vie 16/03/18	mié 21/03/18
12	3.2 Instalación del Software BI	2 días	mié 21/03/18	vie 23/03/18
13	3.3 Aprendizaje del Software	4 días	vie 23/03/18	jue 29/03/18
14	**4. Definición de Requerimientos del Negocio**	**4.5 días**	**mié 07/03/18**	**mar 13/03/18**
15	4.1 Estudio de la documentación seleccionada	4 días	mié 07/03/18	mar 13/03/18
16	4.2 Análisis de ficheros entregados	3 días	mié 07/03/18	vie 09/03/18
17	**5. Modelado Dimensional**	**15 días**	**lun 12/03/18**	**mar 03/04/18**
18	5.1 Diseño del modelo	10 días	lun 12/03/18	mar 27/03/18
19	5.2 Revisión y Validación del modelo	3 días	jue 29/03/18	mar 03/04/18
20	5.3 Entrega PEC 2: Memoria y Diseño del DW	0 días	lun 09/04/18	lun 09/04/18
21	**6. Implementación del DW**	3 días	mié 04/04/18	vie 06/04/18
22	**7. Diseño e Implementación de los procesos ETL**	**18 días**	**mar 03/04/18**	**mié 02/05/18**
23	7.1 Diseño de los procesos ETL	9 días	mar 03/04/18	mar 17/04/18
24	7.2 Implementación de procesos ETL	7 días	mar 17/04/18	jue 26/04/18
25	7.3 Verificación de los procesos	2 días	vie 27/04/18	mié 02/05/18
26	**8. Implantación de Aplicación BI**	**17 días**	**mié 02/05/18**	**mar 29/05/18**
27	8.1 Implantación de Aplicación BI	17 días	mié 02/05/18	mar 29/05/18
28	8.1 Entrega PEC3: Estado del TFM	0 días	lun 07/05/18	lun 07/05/18
29	**9. Análisis de Resultados y Conclusiones**	**4 días**	**mar 29/05/18**	**lun 04/06/18**
30	**10. Entrega de TFM**	**4 días**	**mar 05/06/18**	**lun 11/06/18**
31	10.1 Preparación de Entrega del Producto	3 días	mar 05/06/18	jue 07/06/18
32	10.2 Preparación de la Presentación	4 días	mar 05/06/18	vie 08/06/18
33	10.3 Autoevaluación	1 día	vie 08/06/18	vie 08/06/18
34	10.4 Entrega del TFM	0 días	lun 11/06/18	lun 11/06/18
35	**Documentación en Memoria**	**68 días**	**mié 21/02/18**	**mié 06/06/18**

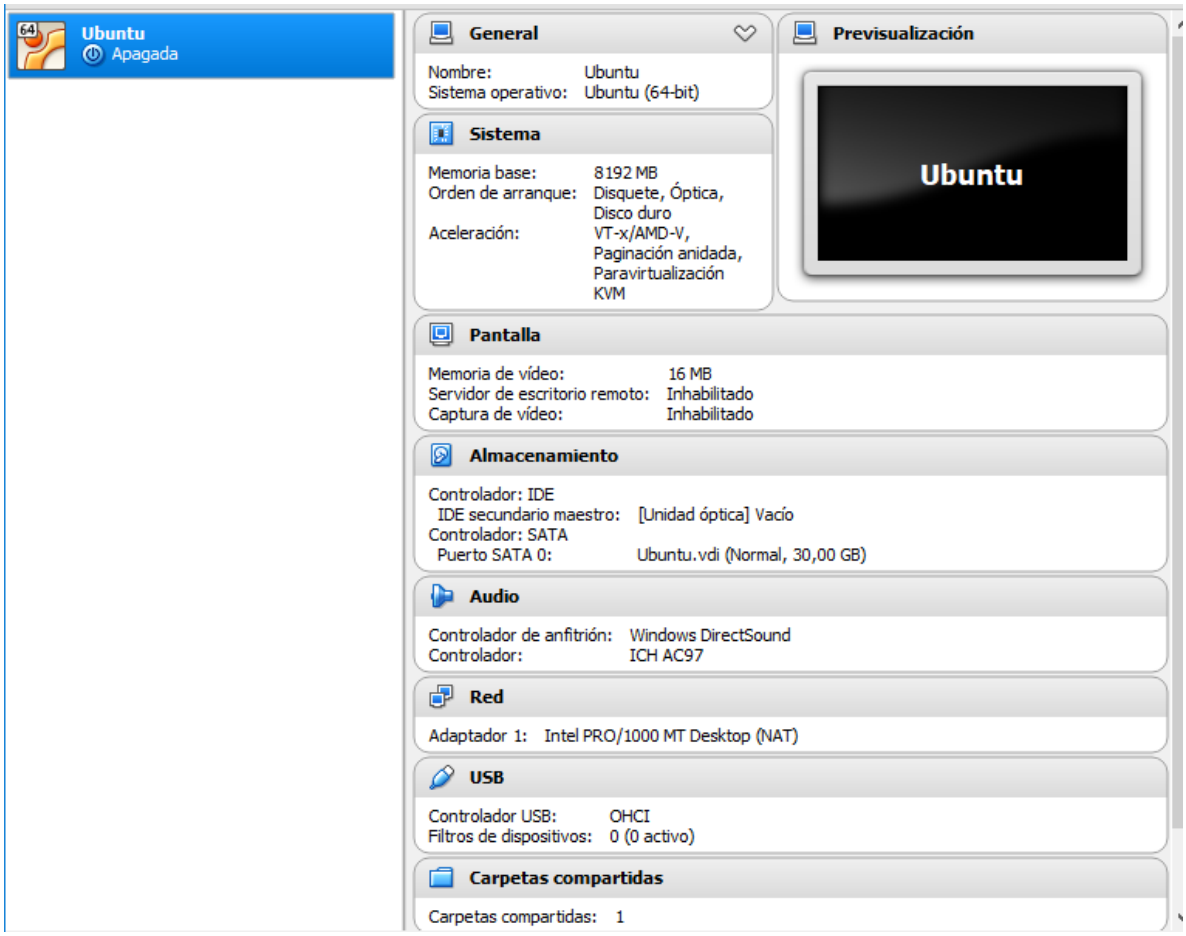


Anexo 2: Preparación del entorno de trabajo

Instalación de plataforma elegida: Pentaho

Se pueden descargar los distintos componentes de https://community.hds.com/docs/DOC-1009931-downloads.

La instalación se realiza sobre una máquina virtual donde corre el sistema operativo Linux Ubuntu 16.04 LTS. Se asignó 30 GB de espacio en el disco y 8GB de memoria RAM.



Como requisito, Pentaho necesita de Oracle Java 8 para funcionar. Se instala con los siguientes comandos:

```
lorena@lorena-VirtualBox:~$ sudo add-apt-repository ppa:webupd8team/java
lorena@lorena-VirtualBox:~$ sudo apt-get update
lorena@lorena-VirtualBox:~$ sudo apt-get install oracle-java8-installer
```

Luego de leer y aceptar la licencia, se instala la máquina virtual y para verificar si se instaló correctamente se invoca a `lorena@lorena-VirtualBox:~$ java -version`. En la siguiente imagen se puede observar que está instalada la version 1.8.0_161.

```
lorena@lorena-VirtualBox:~$ java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
```

Una vez descomprimido el archivo correspondiente al servidor, se ejecuta el siguiente comando para iniciarlo:

```
lorena@lorena-VirtualBox:~/pentaho-server$ sudo ./start-pentaho.bat
```

Cuando esté funcionando el servidor tomcat, desde un navegador, por ejemplo, Firefox, se accede con http://localhost:8080/pentaho.

Instalación de otras aplicaciones

Pentaho Data Integration

Pentaho Data Integration, más conocido como Kettle (acrónimo de Kettle Extraction, Transformation, Transportation, and Load Environment) permite crear transformaciones y trabajos para construir, actualizar y mantener un data warehouse.

A su vez provee una interfaz gráfica Spoon que facilita el diseño de los procesos ETL. La ejecución se hace sobre otras dos herramientas de Kettle, Pan y Kitchen que se encargan de ejecutar, generalmente por lotes (modo batch), transformaciones y trabajos, respectivamente.

Como se dijo anteriormente, es una herramienta muy popular y con gran reputación.

Para lanzar Spoon, se escribe en la consola:

```
lorena@lorena-VirtualBox:~/pentaho/data-integration$ ./spoon.sh
```

Plugin Saiku

Saiku es un plugin que permite realizar análisis OLAP de una manera muy sencilla. Para instalarlo se debe descargar el paquete del repositorio de Pentaho. Luego, se descomprime la carpeta en pentaho/pentaho-server/pentaho-solutions/system. Si bien es gratuito, para su uso, se debe gestionar una licencia que se guarda en el mismo directorio de Saiku.

Instalación de MySQL Server y MySQL Workbench

Para instalar el servidor de MySQL, se utilizó como guía https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-16-04.

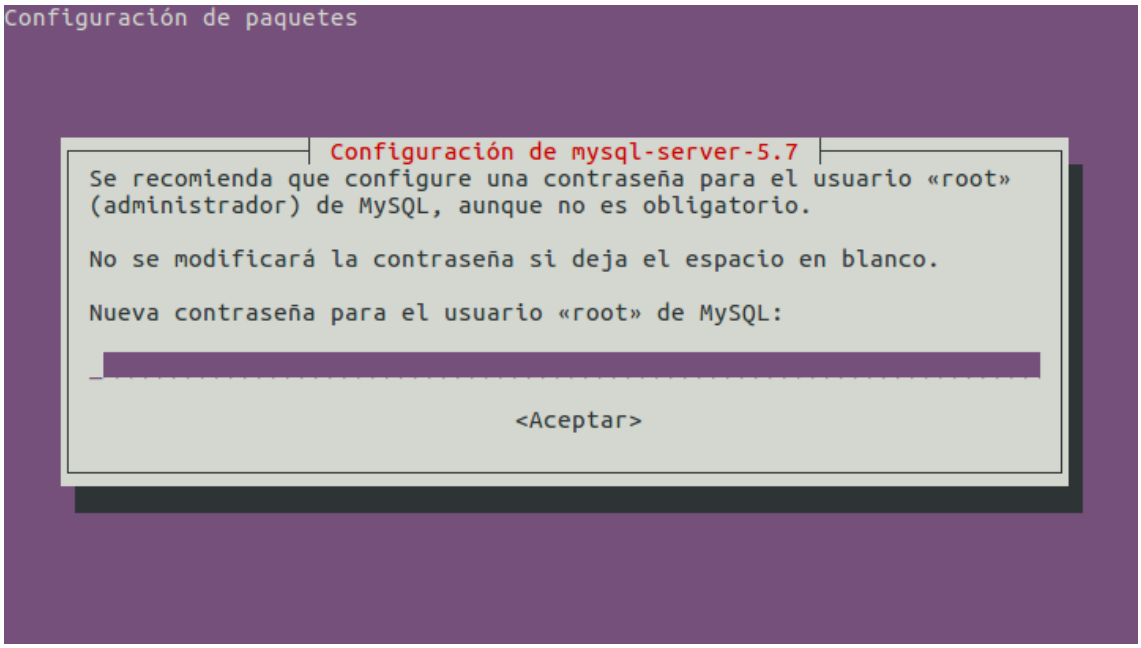
Como primer paso, se actualiza el repositorio del gestor apt-get con el siguiente comando:

```
lorena@lorena-VirtualBox:~$ sudo apt-get update
```

A continuación, se instala el servidor de MySQL. La versión disponible en el momento de realizar este trabajo es la 5.7.

```
lorena@lorena-VirtualBox:~$ sudo apt-get install mysql-server
```

Automáticamente se inicia la configuración del mismo que consiste en asignarle una contraseña al usuario “root”.



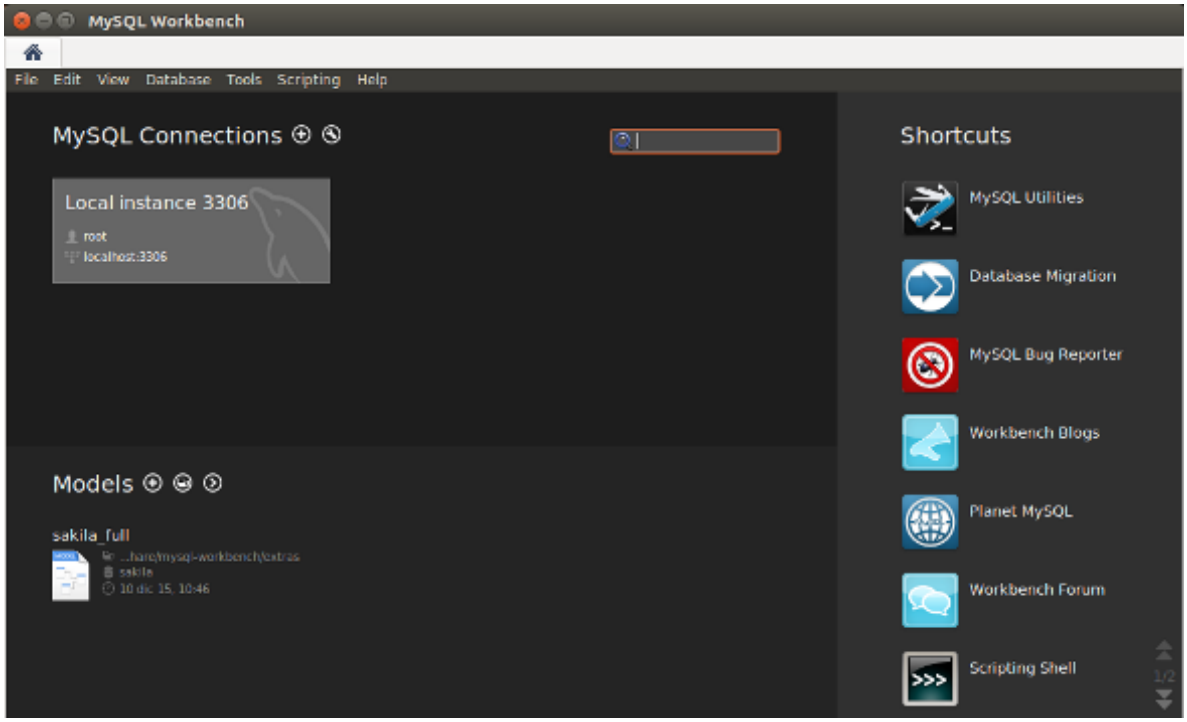
Luego de confirmarla, MySQL Server ya está listo para usar.

MySQL Workbench

Dado que MySQL Workbench, al igual que MySQL Server, se encuentra en los repositorios de Ubuntu, se puede instalar a través del gestor apt-get. Como en el momento de instalar el servidor ya se actualizó el repositorio, no es necesario actualizarlo y es suficiente sólo ejecutando el siguiente comando:

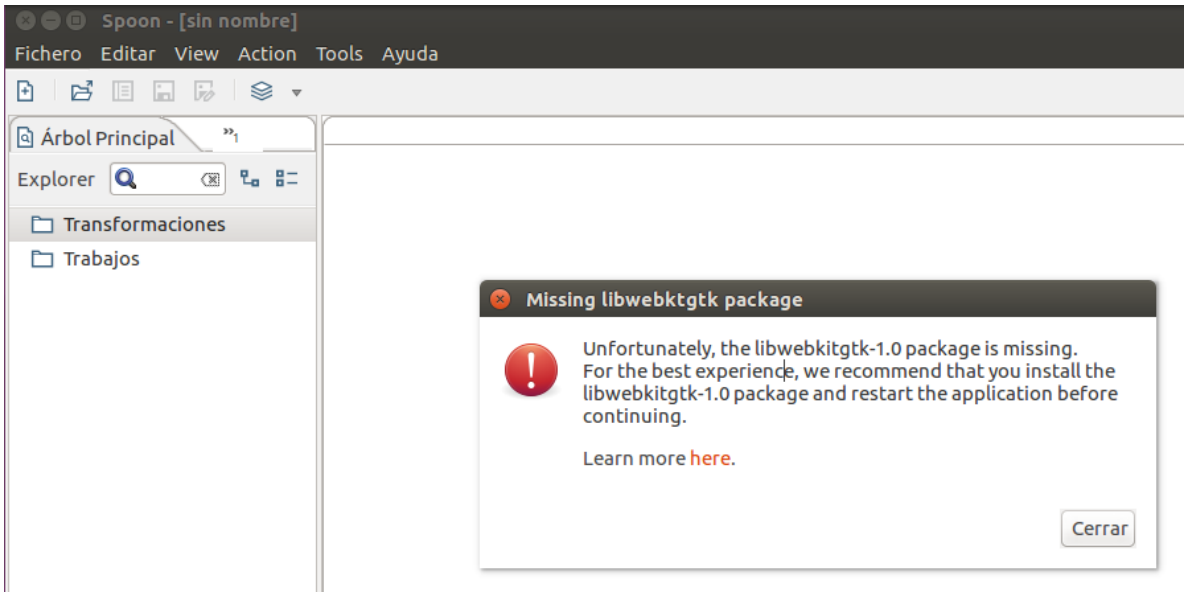
```
lorena@lorena-VirtualBox:~$ sudo apt-get install mysql-workbench
```

Cabe destacar que la conexión local se configura automáticamente.



Dificultades enfrentadas

Librería libwebkitgtk

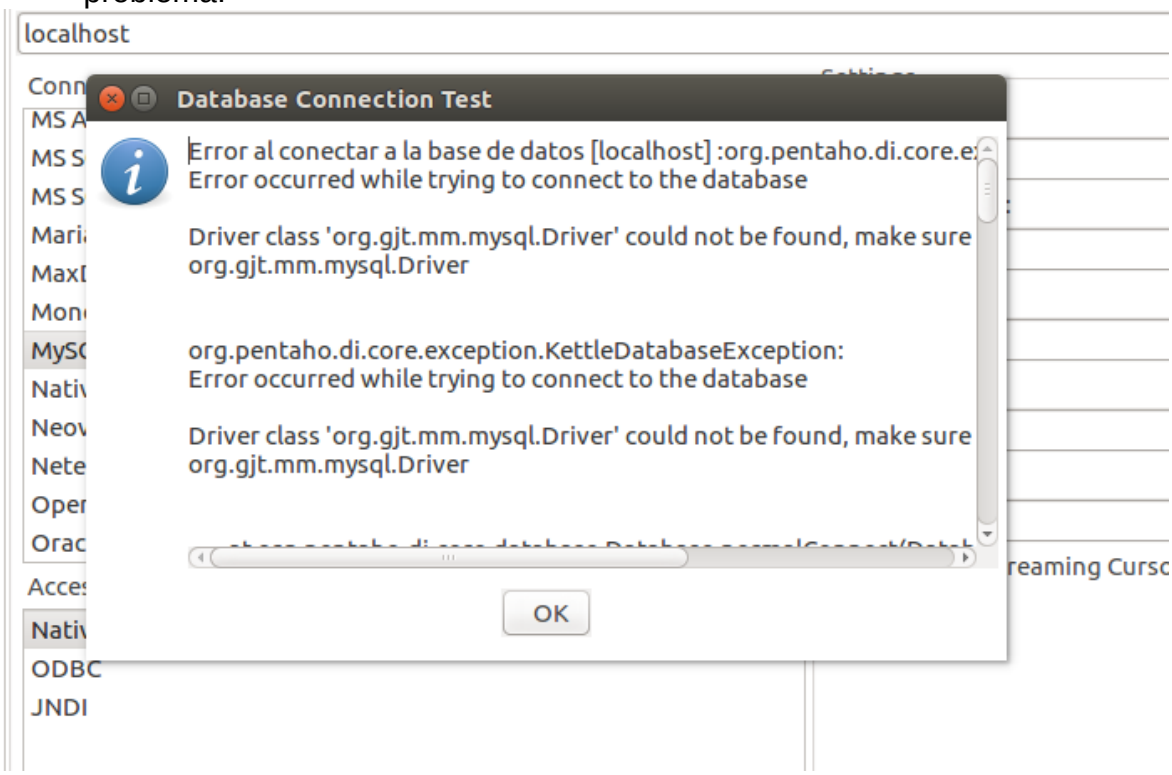


Para subsanar este inconveniente, se debe instalar el paquete faltante con el comando:

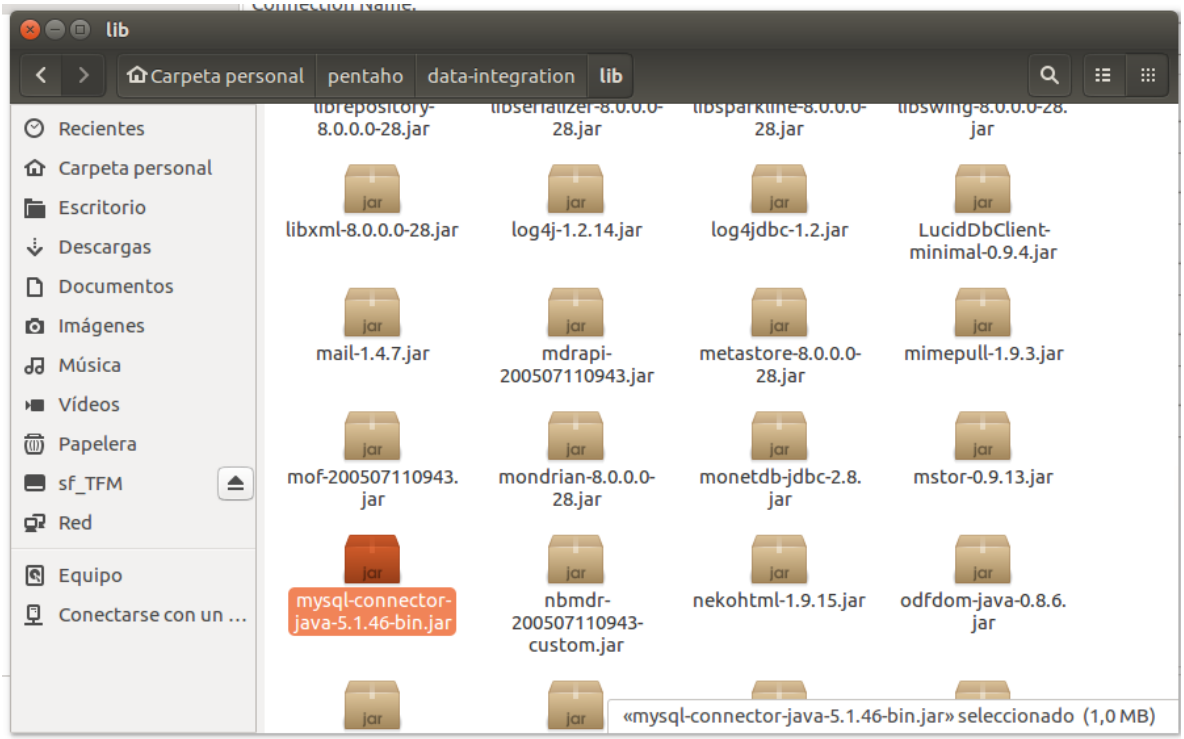
```
lorena@lorena-VirtualBox:~$ sudo apt-get install libwebkitgtk-1.0-0
```

Driver MySQL

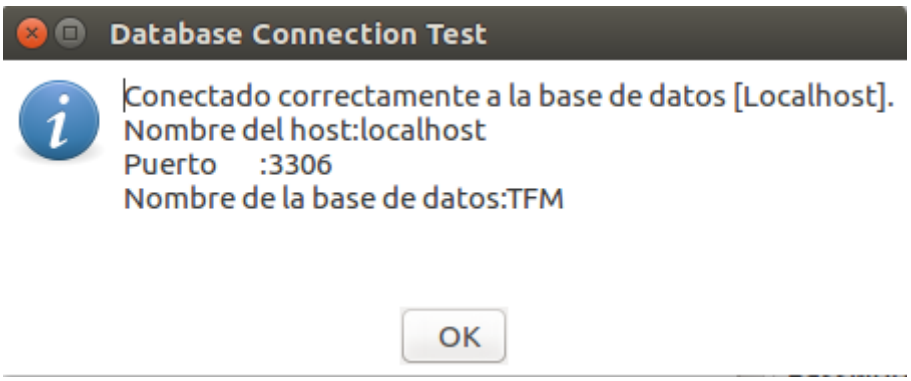
Al crear la conexión con la base de datos MySQL, surgió el siguiente problema:



Esto se debe a que Pentaho, a pesar de soportar MySQL, utiliza como predeterminado el servidor PostgresSQL. Para ello y siguiendo lo indicado en https://stackoverflow.com/questions/11634181/pentaho-data-integration-sql-connection.
Se descarga el conector de MySQL de: https://dev.mysql.com/downloads/connector/j/.
A continuación, se copia el archivo *mysql-connector-java-5.1.46-bin.jar* en la carpeta *lib* de *data-integration*.

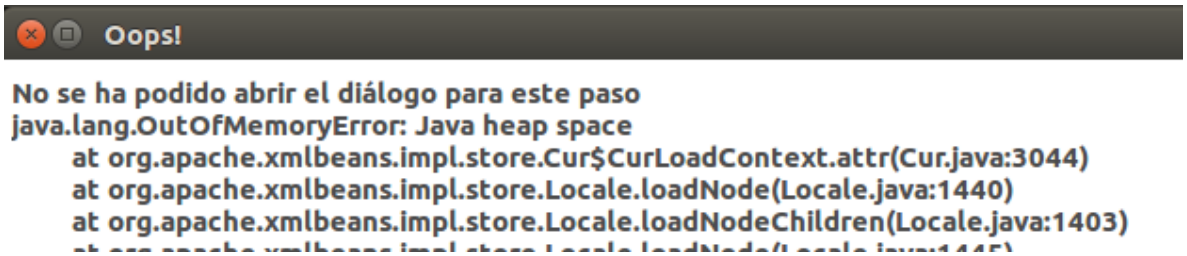


Luego se reinicia Spoon y se vuelve a probar la conexión:



Si el mismo problema ocurre en *Schema Workbench,* se aplica la misma solución.

Problema con heap memory



Spoon.bat (~/pentaho/data-integration) - gedit

Abrir Guardar

```
REM ***********************

set _cmdline=
:TopArg
if %1==! goto EndArg
set _cmdline=%_cmdline% %1
shift
goto TopArg
:EndArg

REM *****************************************************************
REM ** Set java runtime options                                    **
REM ** Change 2048m to higher values in case you run out of memory  **
REM ** or set the PENTAHO_DI_JAVA_OPTIONS environment variable      **
REM *****************************************************************

if "%PENTAHO_DI_JAVA_OPTIONS%"=="" set PENTAHO_DI_JAVA_OPTIONS="-Xms1024m" "-Xmx2048m" "-
XX:MaxPermSize=256m"
```

```
REM *****************************************************************
REM ** Set java runtime options                                    **
REM ** Change 2048m to higher values in case you run out of memory  **
REM ** or set the PENTAHO_DI_JAVA_OPTIONS environment variable      **
REM *****************************************************************

if "%PENTAHO_DI_JAVA_OPTIONS%"=="" set PENTAHO_DI_JAVA_OPTIONS="-Xms1024m" "-Xmx3072m" "-
XX:MaxPermSize=256m"
```

*spoon.sh (~/pentaho/data-integration) - gedit

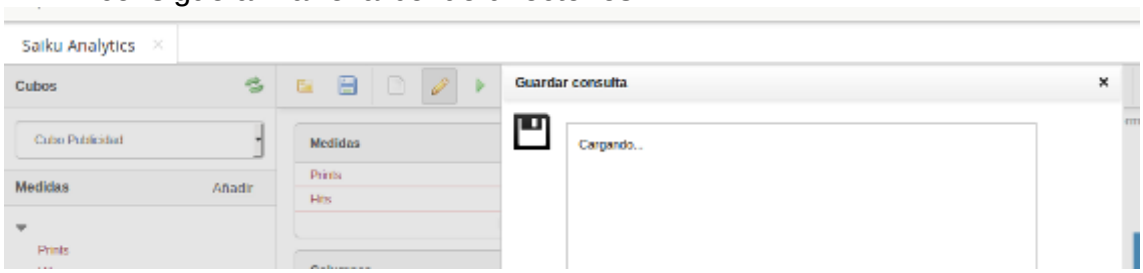
Abrir

```
                ./Spoon.bat
                exit
                ;;

        *)
                echo Spoon is not supported on this hosttype : `uname -s`
                exit
                ;;
esac

export LIBPATH

# *****************************************************************
# ** Set java runtime options                                    **
# ** Change 2048m to higher values in case you run out of memory  **
# ** or set the PENTAHO_DI_JAVA_OPTIONS environment variable      **
# *****************************************************************

if [ -z "$PENTAHO_DI_JAVA_OPTIONS" ]; then
    PENTAHO_DI_JAVA_OPTIONS="-Xms1024m -Xmx3072m -XX:MaxPermSize=256m"
fi
```

A pesar de aumentar el tamaño de la memoria para la máquina virtual a un valor superior al recomendado, cuando se carga el archivo excel original, Spoon sigue dando el mismo error de memoria insuficiente. Por esta razón, se decidió dividir cada hoja en un archivo diferente.

Saiku: Error al guardar las queries

El error que surge al intentar guardar una consulta se produce porque no consigue armar el árbol de directorios:



La solución oficial se puede encontrar en https://jira.pentaho.com/browse/BISERVER-13666. Consiste en copiar los archivos *cpf-core-8.0.0.0-28.jar*; *cpf-pentaho-8.0.0.0-28.jar*; *cpk-core-8.0.0.0-28.jar* y *cpk-pentaho5-8.0.0.0-28.jar* en la carpeta *lib* de Saiku y eliminar de la misma ruta los archivos anteriores *cpf-core-xxx.jar* y *cpf-pentaho-xxx.jar*.

Los archivos *cpk-core-8.0.0.0-28.jar* y *cpk-pentaho5-8.0.0.0-28.jar* se descargaron de https://github.com/oncase/portalboilerplate/tree/master/lib.

Scripts

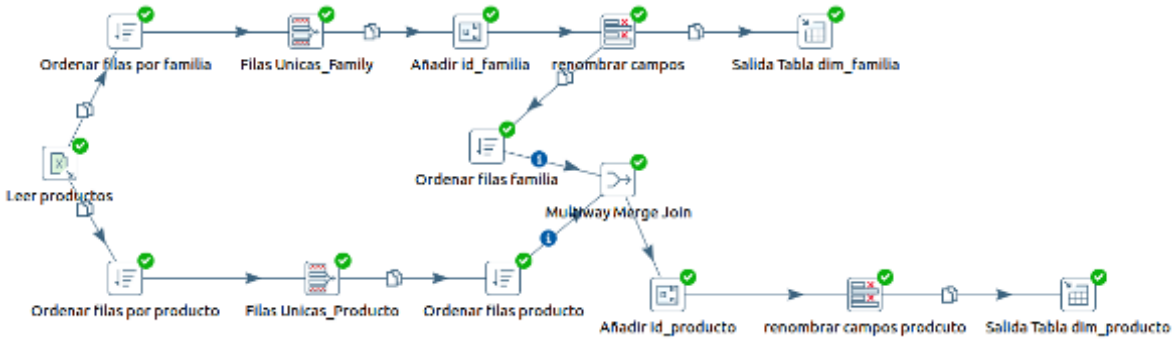
Un error con el que el lector se puede encontrar es la falta de permiso al ejecutar los scripts. Se soluciona como se ve en la siguiente captura:

```
lorena@lorena-VirtualBox:~$ ./iniciar_servidor.sh
bash: ./iniciar_servidor.sh: Permiso denegado
lorena@lorena-VirtualBox:~$ chmod u+x iniciar_servidor.sh
lorena@lorena-VirtualBox:~$ ./iniciar_servidor.sh
DEBUG: Using JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/lib/jvm/java-8-oracle
DEBUG: _PENTAHO_JAVA=/usr/lib/jvm/java-8-oracle/bin/java
Using CATALINA_BASE:   /home/lorena/pentaho/pentaho-server/tomcat
Using CATALINA_HOME:   /home/lorena/pentaho/pentaho-server/tomcat
Using CATALINA_TMPDIR: /home/lorena/pentaho/pentaho-server/tomcat/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-oracle
Using CLASSPATH:       /home/lorena/pentaho/pentaho-server/tomcat/bin/bootstrap.
jar:/home/lorena/pentaho/pentaho-server/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

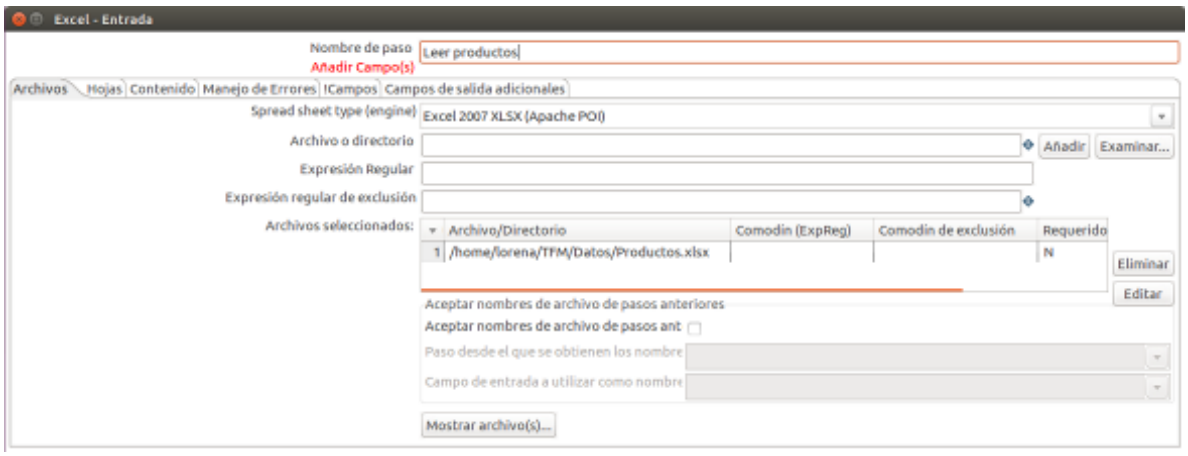
Anexo 3: Procesos ETL

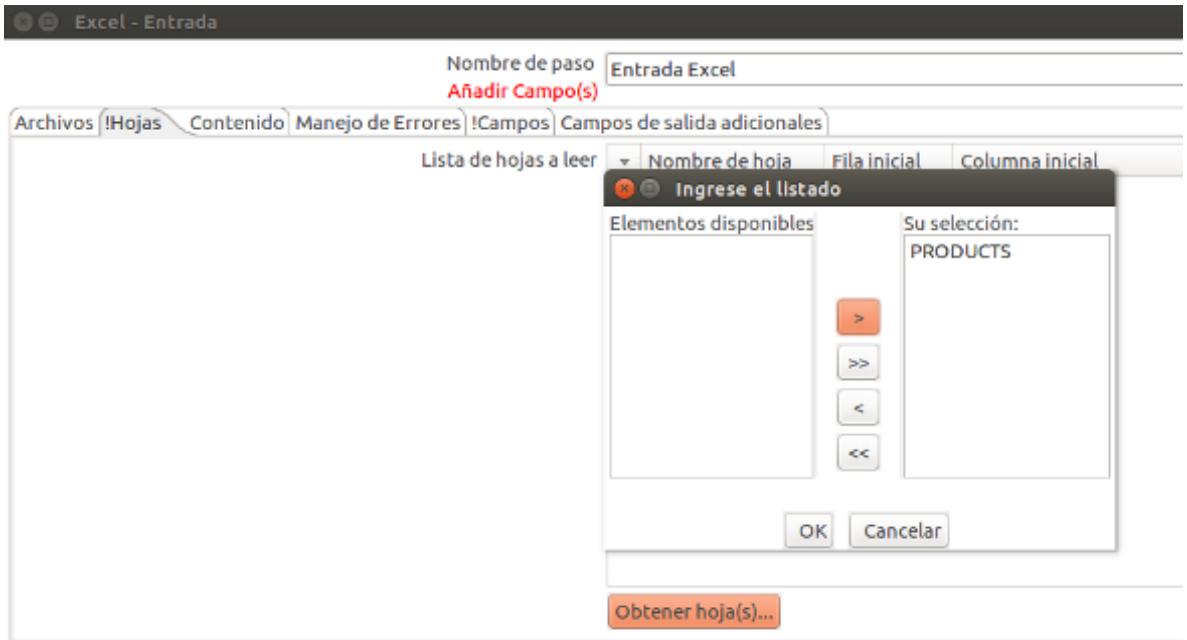
Dimensión Producto y Familia

El archivo correspondiente a este proceso es **t_d_producto_familia.ktr**.



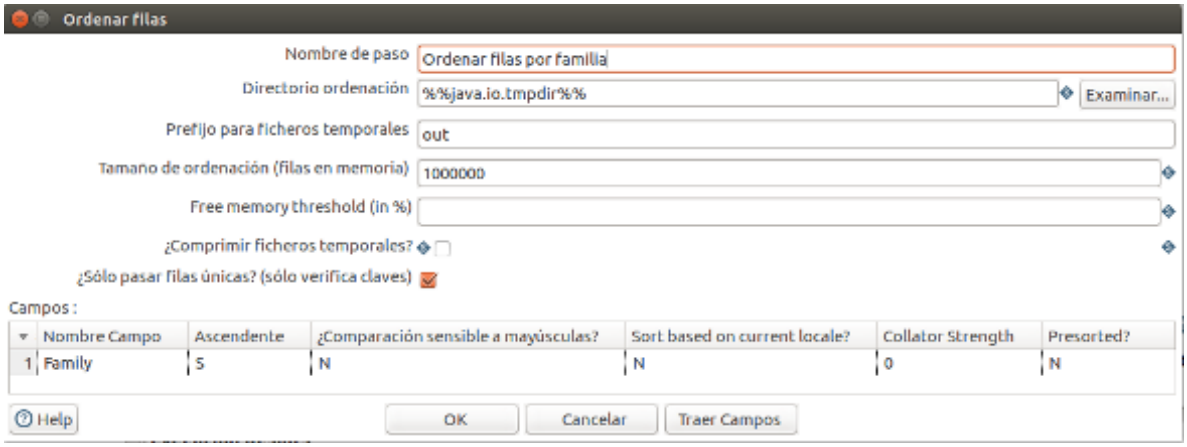
Paso 1: Leer entrada de datos



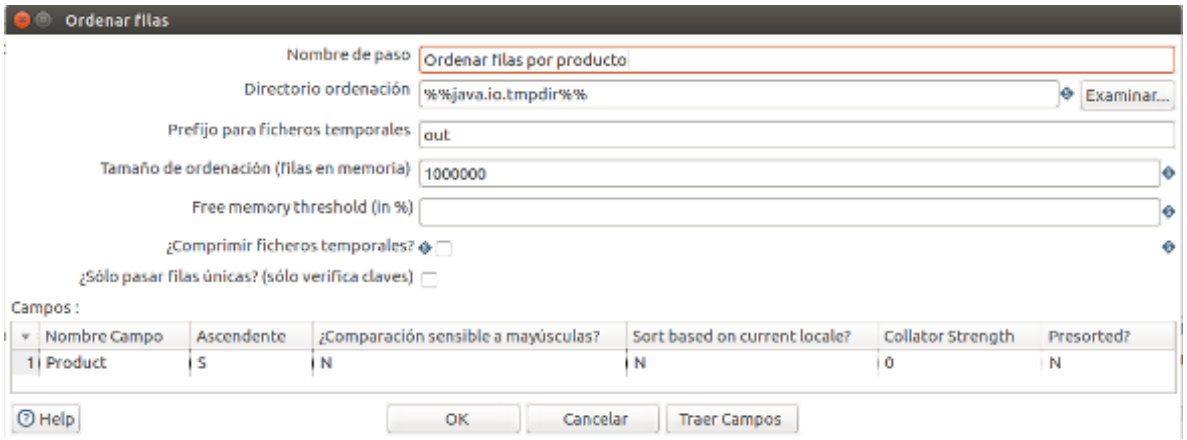


Paso 2: Ordenación de los datos
Es necesario ordenar los datos por el atributo que se desea filtrar con el objeto Filas Únicas.

Para la dimensión familia (rama superior del diagrama) se lo hace ordenando por Family:

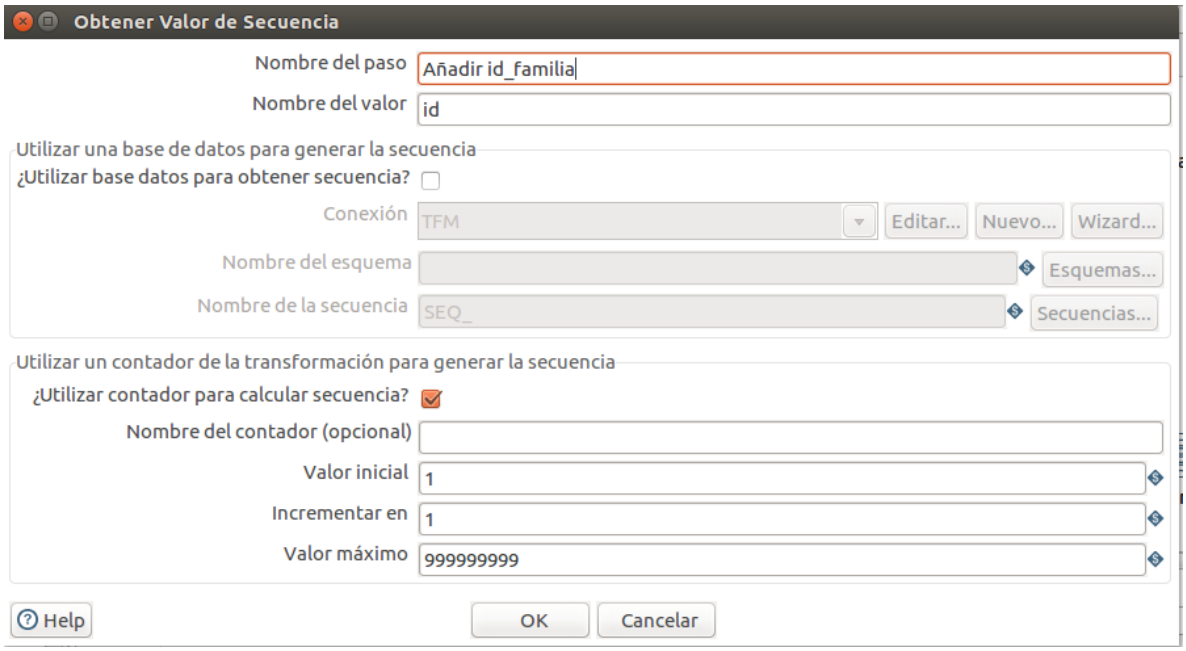


Para la otra dimensión se lo ordena por Product:



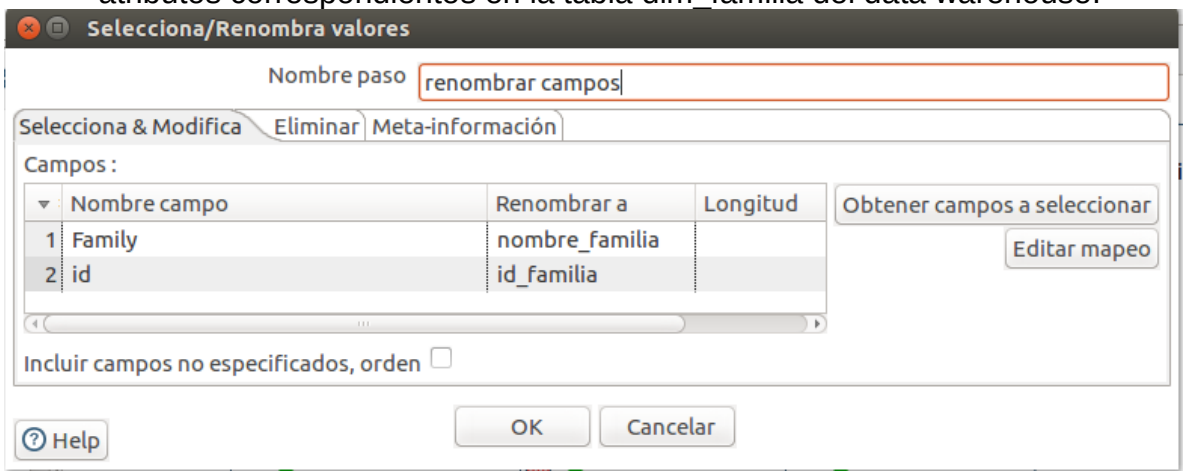
Paso 3: Filas únicas
Se filtran aquellas filas repetidas, tanto para familia como para producto. Cabe destacar que no se repiten los productos por lo que se esperan obtener 8 filas. No ocurre lo mismo con las familias que sólo hay cinco.

Rama superior:
Paso 4: Añadir id
Se necesita añadir un identificador para cada familia. Esto se hizo usando el objeto añadir secuencia definiéndolo como se muestra a continuación:



Paso 5: Renombrar campos

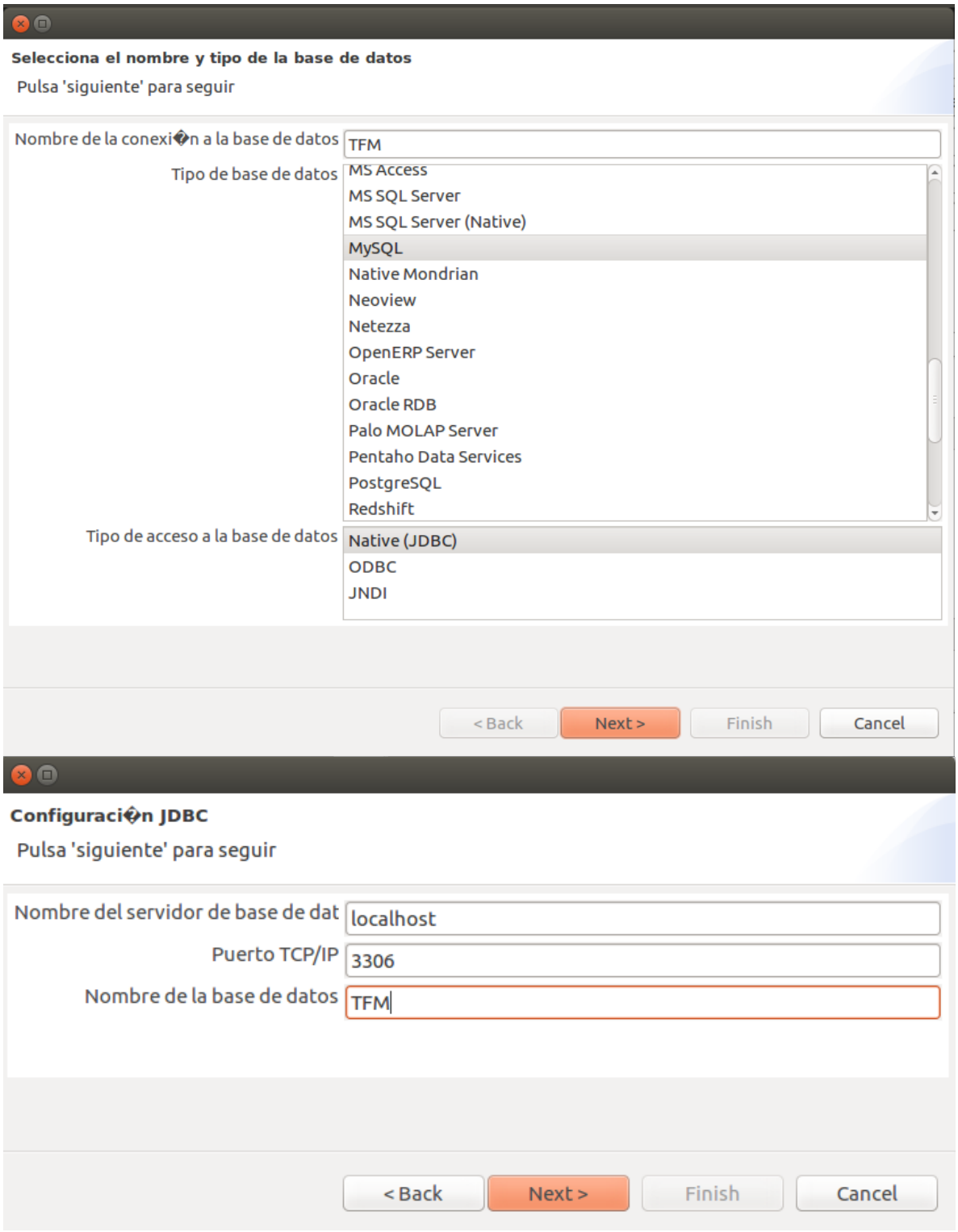
Es necesario renombrar los campos de la entrada de datos según los atributos correspondientes en la tabla dim_familia del data warehouse.

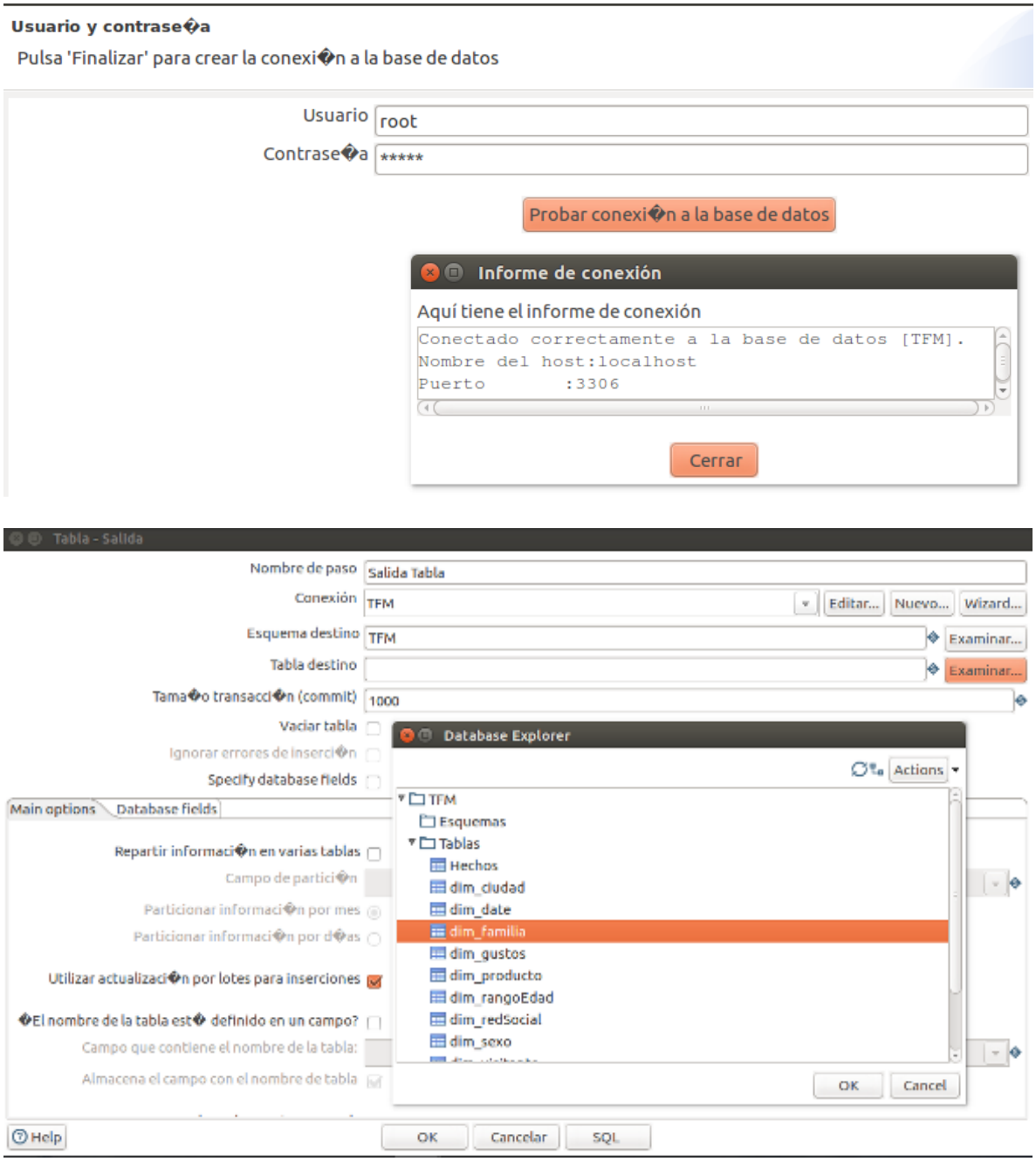


Paso 6: Salida tabla

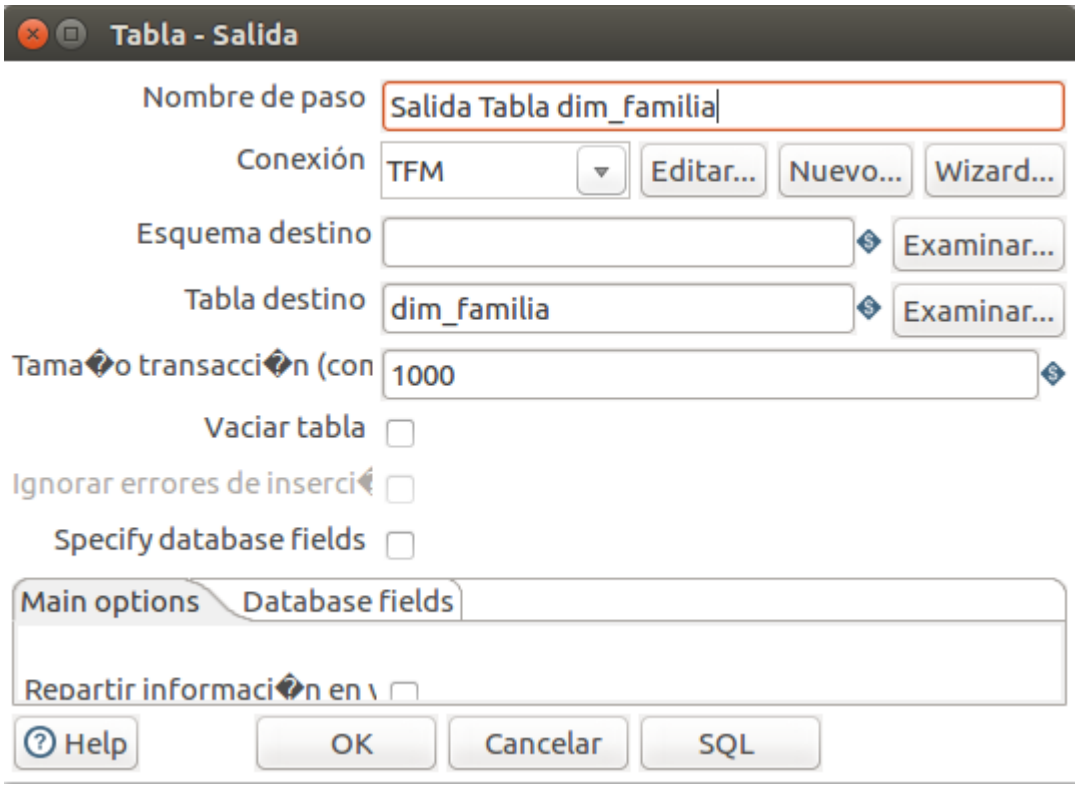
Finalmente, para cargar los datos en el data warehouse se deben definir la conexión al schema MySQL y la tabla en donde guardarlos.

Hay dos maneras de crear la conexión al data warehouse: utilizando el wizard o a través de la opción nueva. En esta oportunidad, se siguió la primera.





Los parámetros quedan definidos como se ve a continuación:

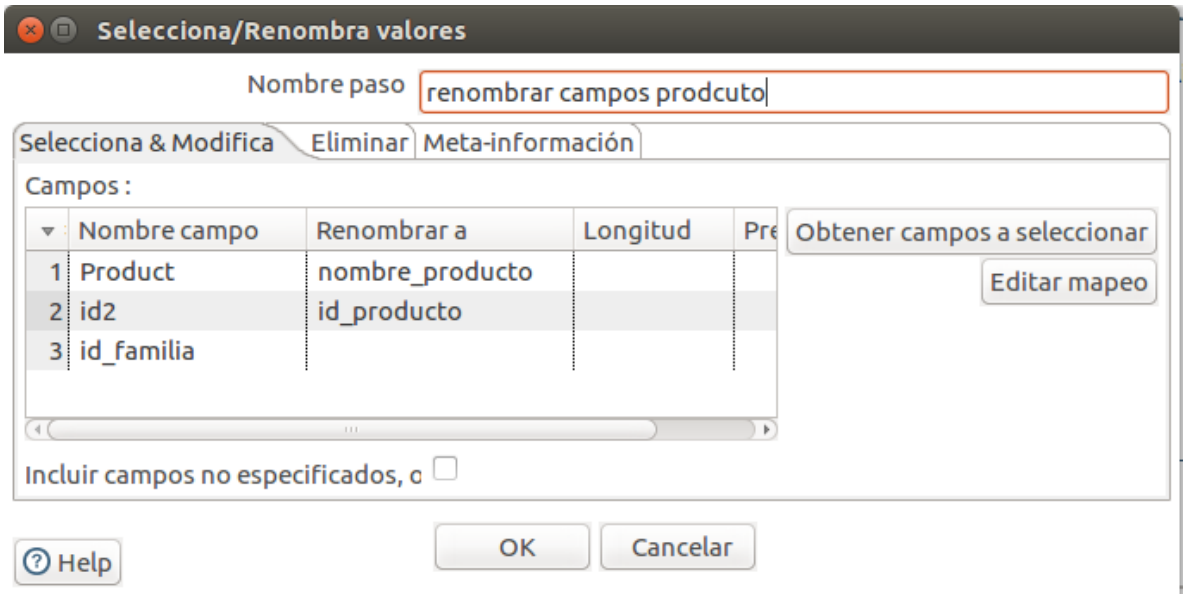


Rama inferior:
Paso 4: Ordenación de los datos
La dimensión producto tiene una relación con la de familia por lo que se debe asociar el identificador de la familia con el producto. Así, se vuelven a ordenar los datos, pero en esta oportunidad, por familia.

Paso 5: Multiway Merge Join
Se deben unir los datos de los productos con el identificador de la familia a la que pertenece.

Paso 6: Añadir id
Se necesita añadir un identificador para cada producto. Nuevamente se utiliza añadir secuencia.

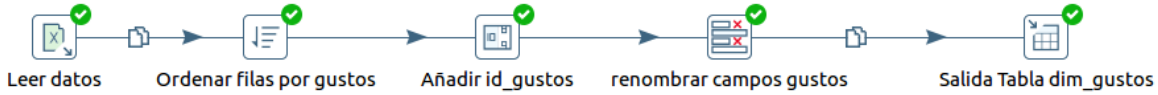
Paso 7: Renombrar campos
Es necesario renombrar los campos de la entrada de datos según los atributos correspondientes en la tabla dim_prodcuto del data warehouse.



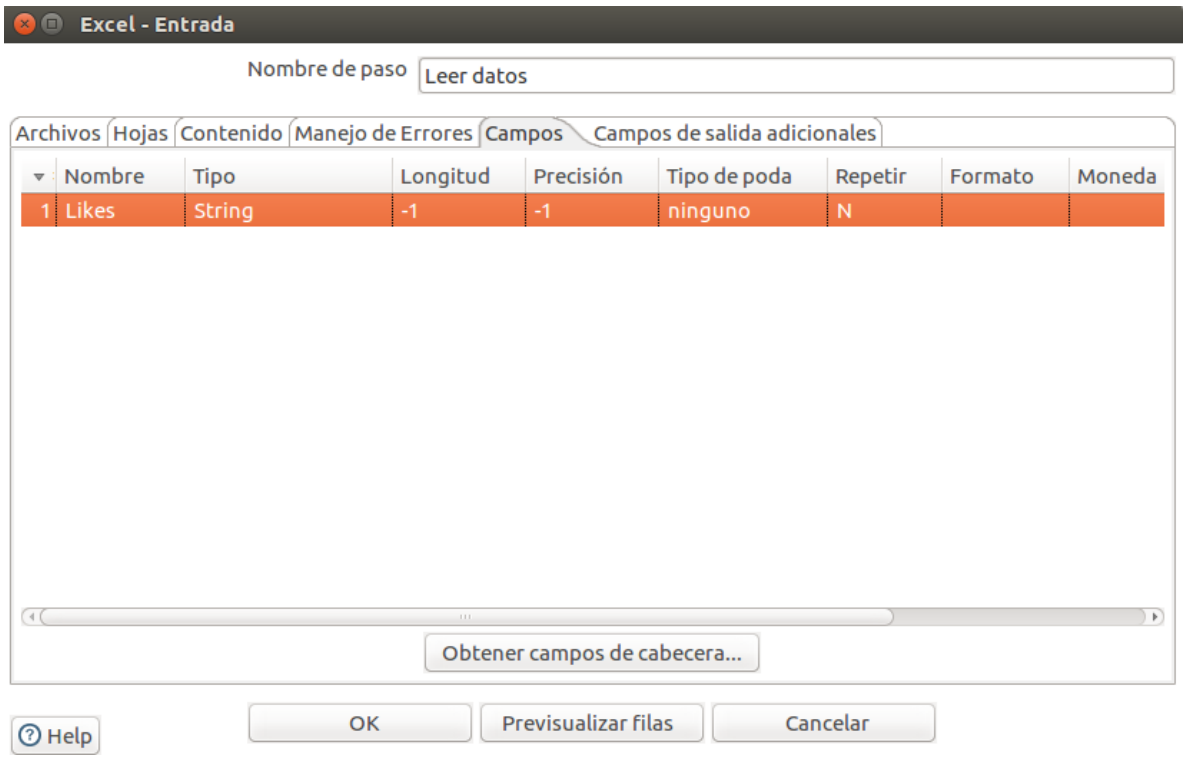
Paso 8: Salida tabla
Finalmente, se cargan los datos en el data warehouse.

Dimensión Gustos

El archivo correspondiente a este proceso es **t_d_gustos.ktr**.

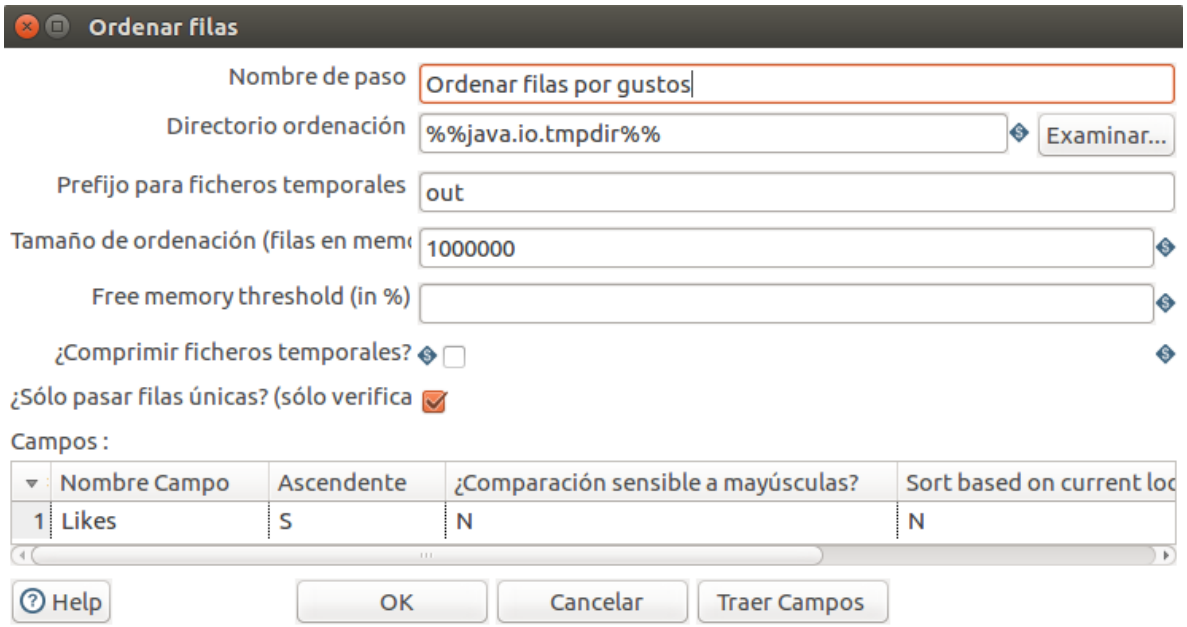


Paso 1: Leer entrada de datos



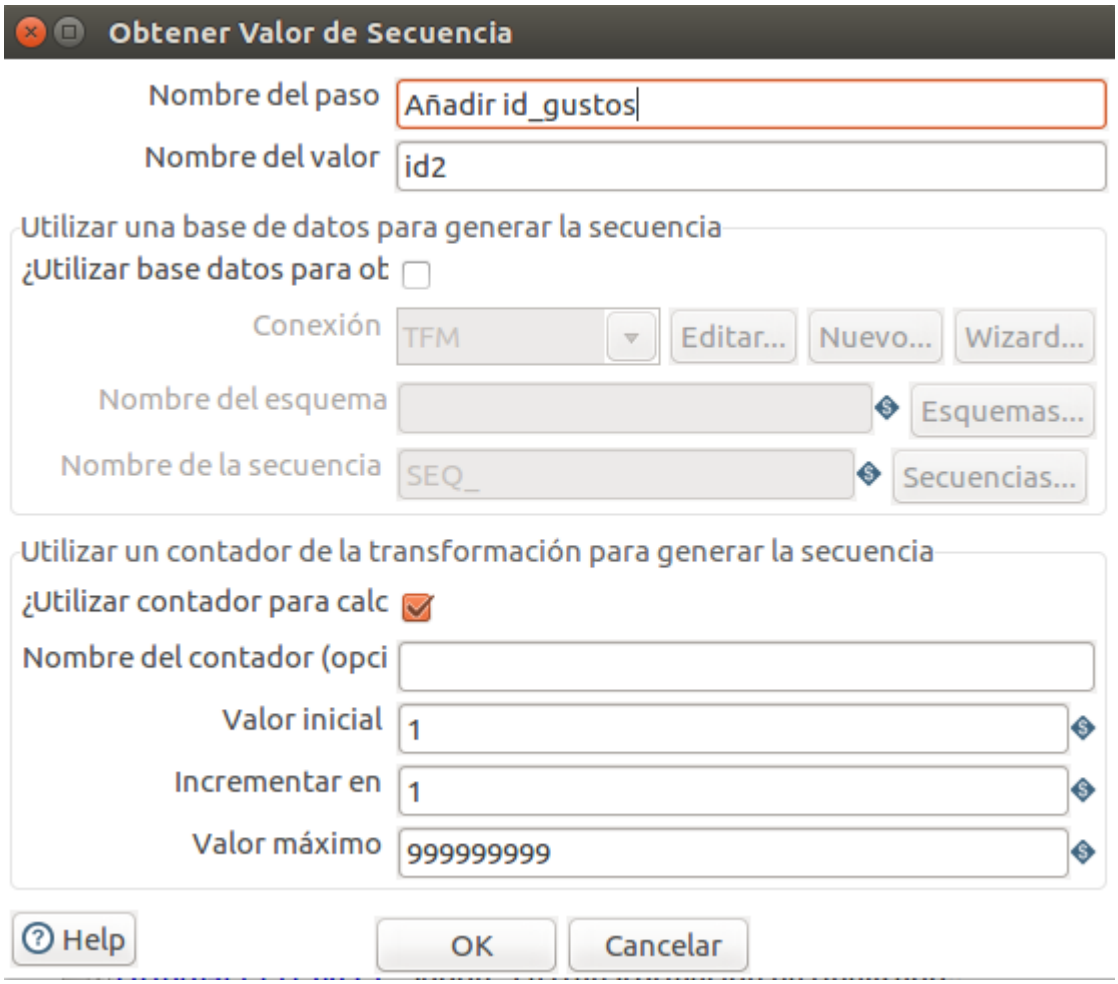
Paso 2: Ordenación de los datos

En este paso lo que se busca es ordenar los datos y a su vez, eliminar los duplicados del campo *Likes* tildando la opción “¿Sólo pasar filas únicas?”.

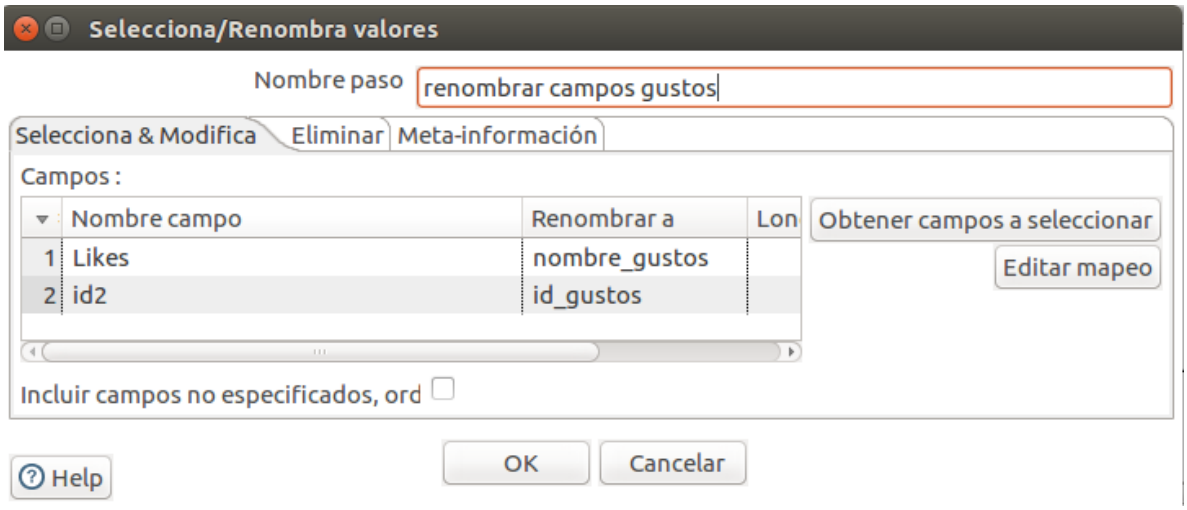


Paso 3: Añadir id

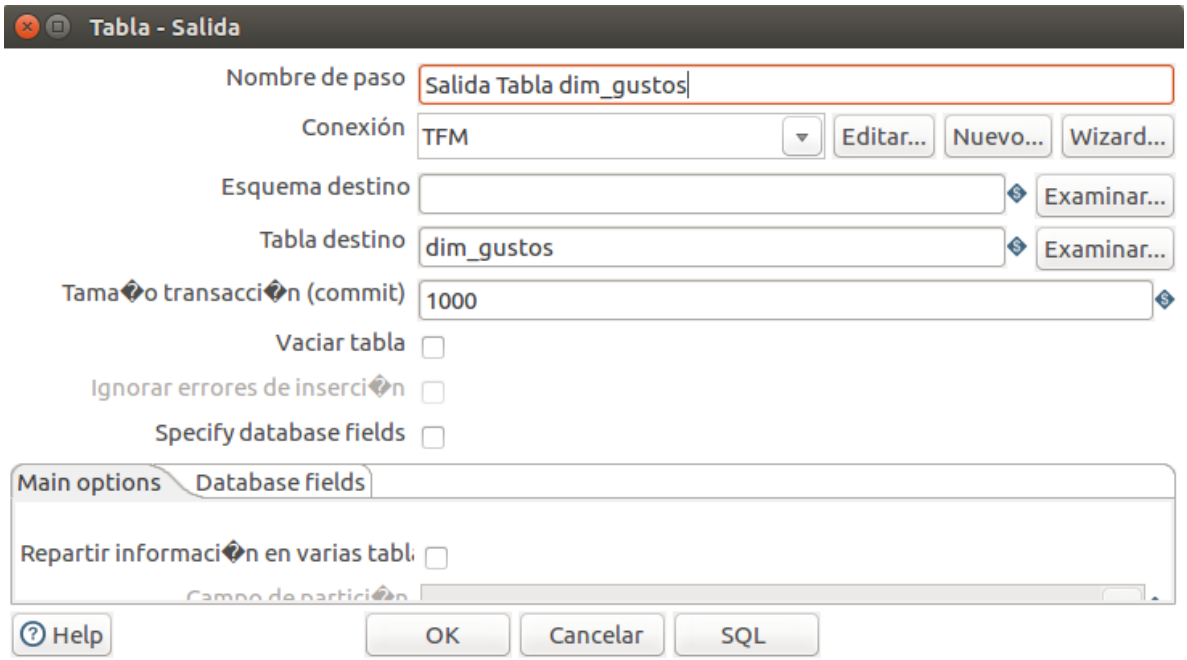
Se necesita añadir un identificador para cada gusto. Esto se hizo usando el objeto añadir secuencia definiéndolo como se muestra a continuación:



Paso 4: Renombrar campos
Es necesario renombrar los campos de la entrada de datos según los atributos correspondientes en la tabla dim_gustos del data warehouse.

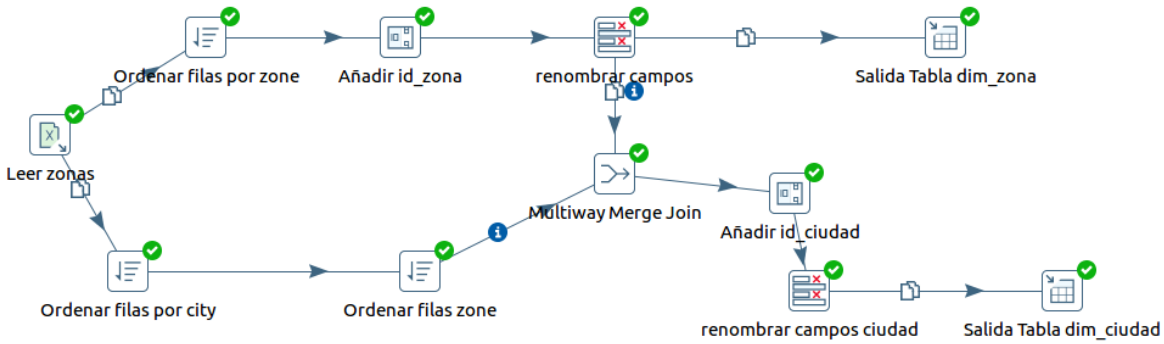


Paso 5: Salida tabla
Finalmente, para cargar los datos en el data warehouse se deben definir la conexión al schema MySQL y la tabla en donde guardarlos.

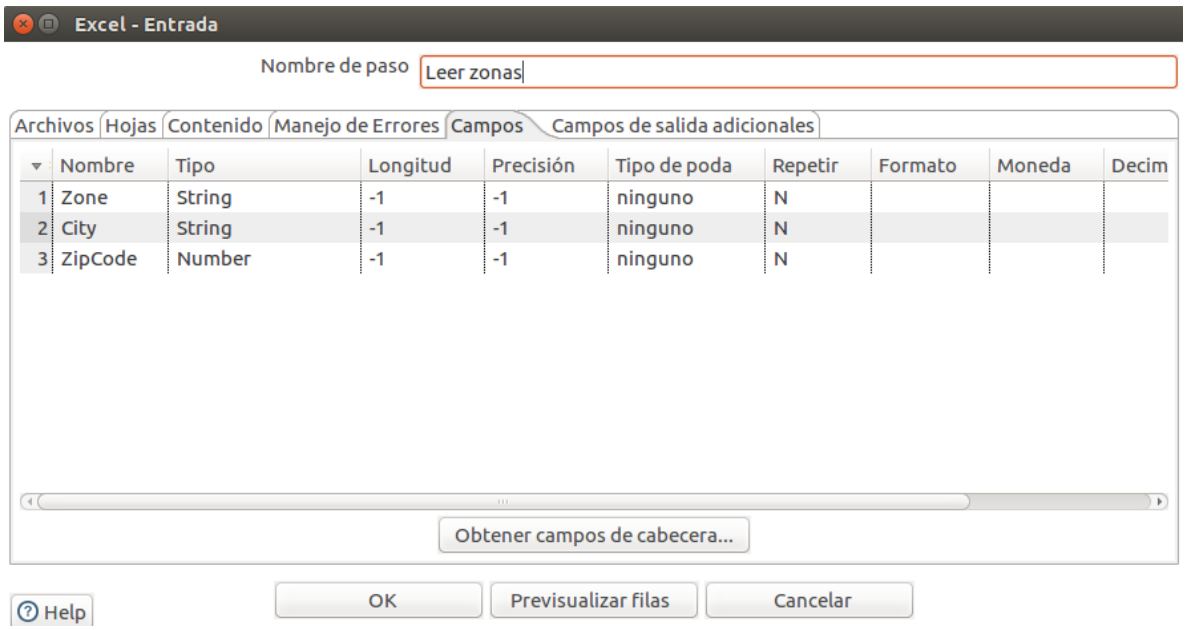


Dimensión Ciudad y Zona

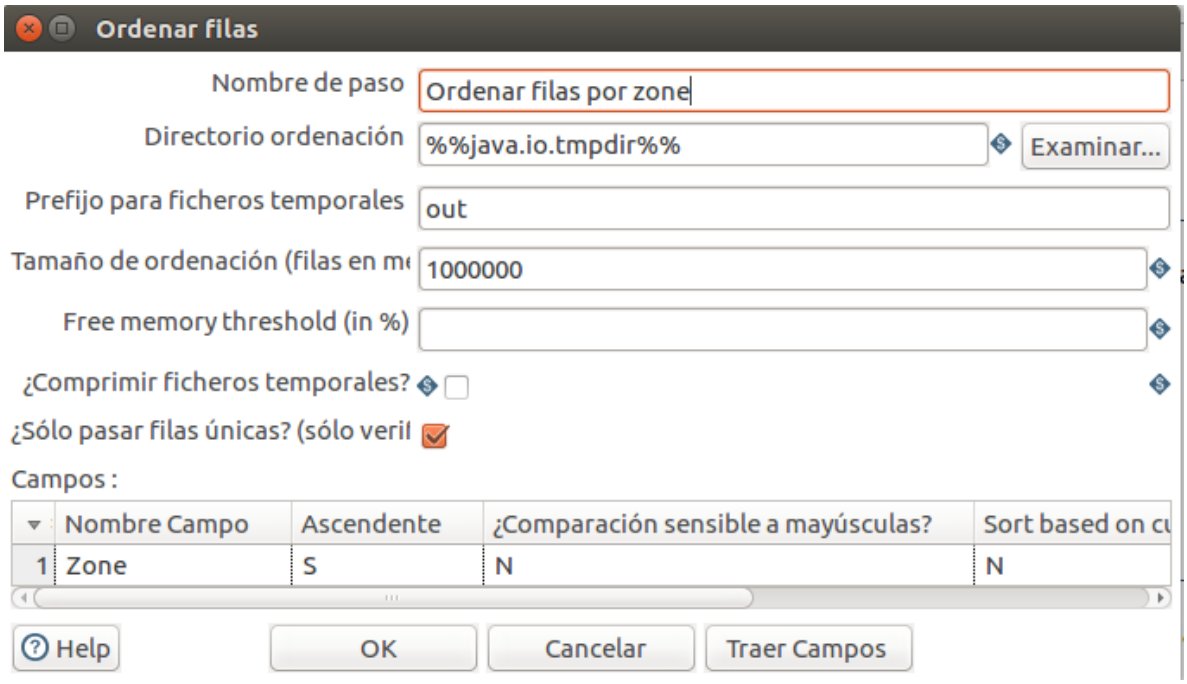
El archivo correspondiente a este proceso es **t_d_zonas.ktr**.



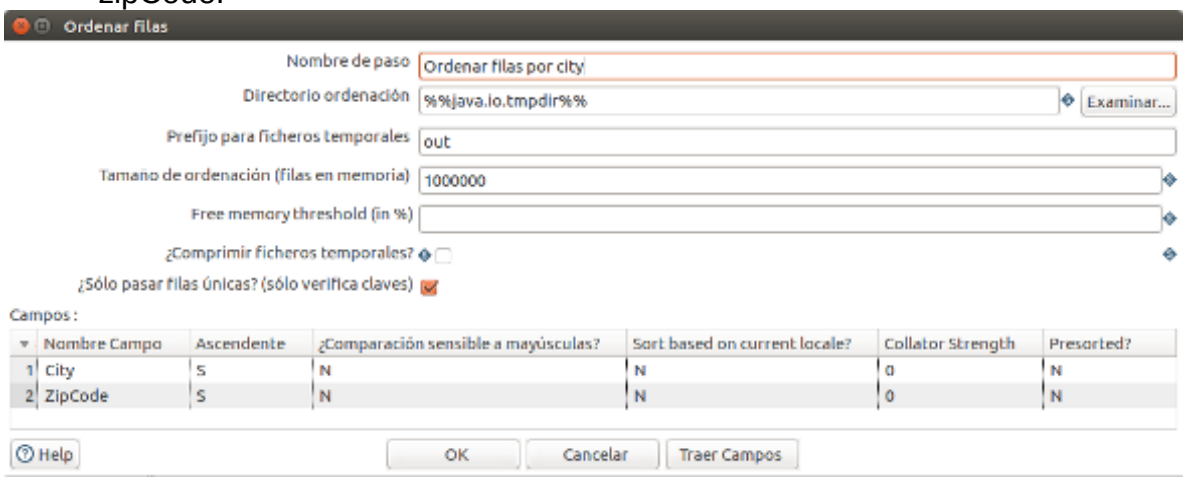
Paso 1: Leer entrada de datos



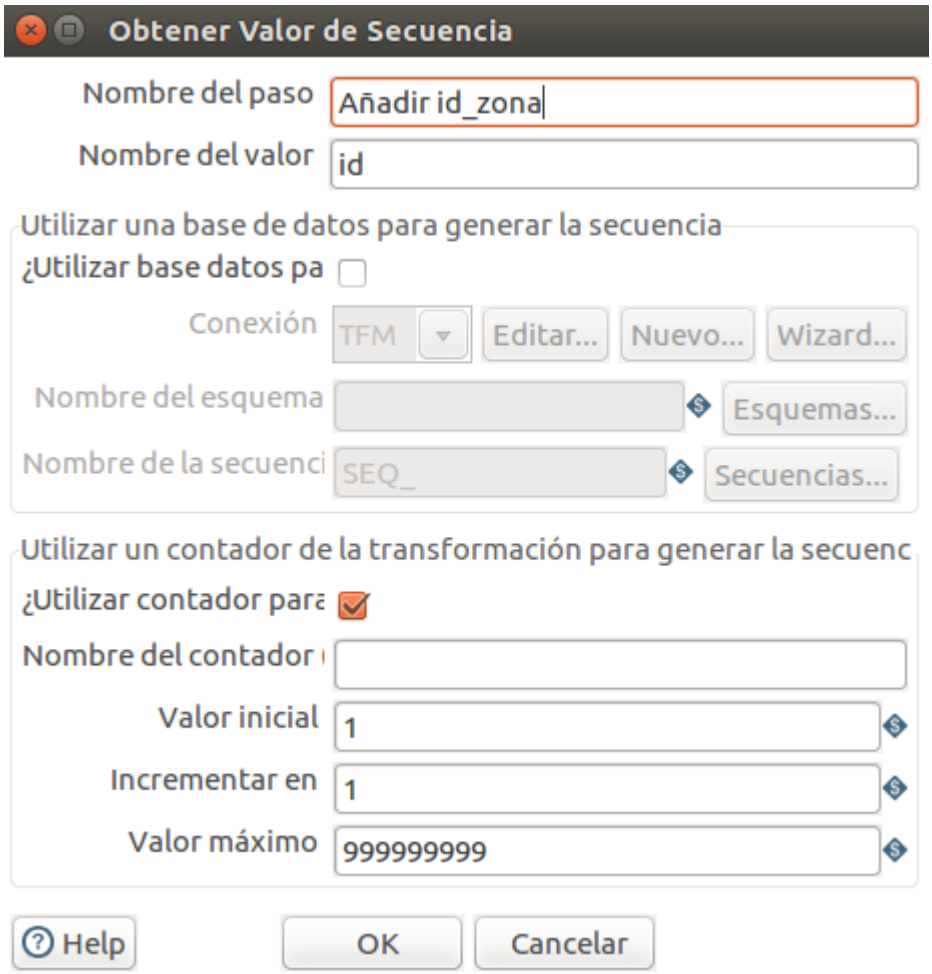
Paso 2: Ordenación de los datos
En este paso lo que se busca es ordenar los datos y a su vez, eliminar los duplicados del campo *Zone* tildando la opción "¿Sólo pasar filas únicas?". Esto ocurre en la rama superior.



En la rama inferior, lo que se elimina sería la combinación de City – zipCode:

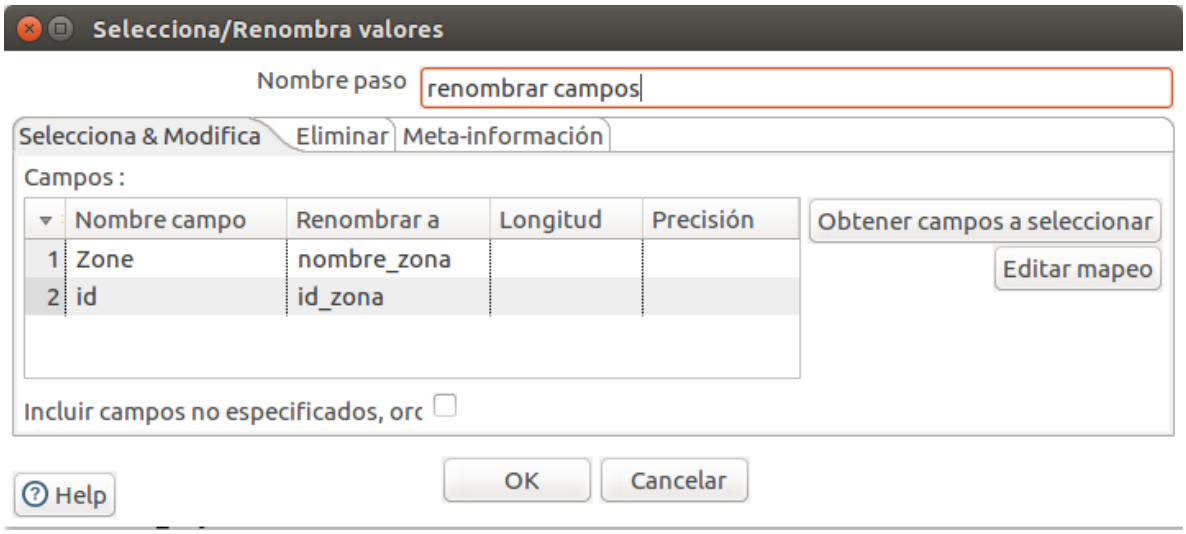


Rama superior:
Paso 3: Añadir id
Se necesita añadir un identificador para cada zona. Esto se hizo usando el objeto añadir secuencia:



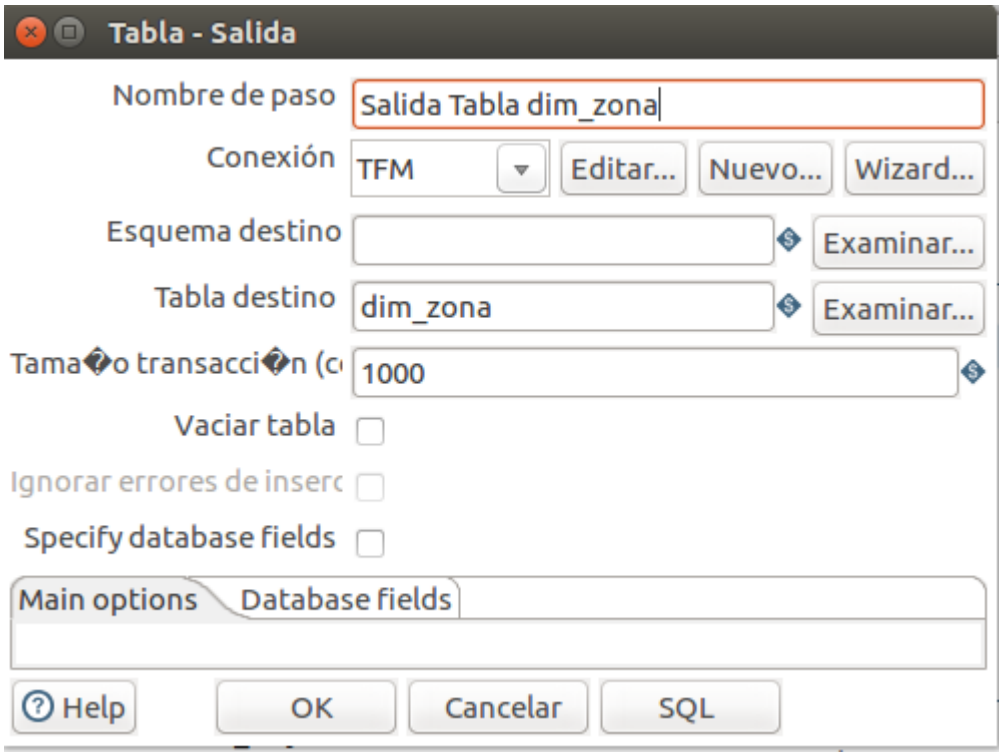
Paso 4: Renombrar campos

Es necesario renombrar los campos de la entrada de datos según los atributos correspondientes en la tabla dim_zona del data warehouse.

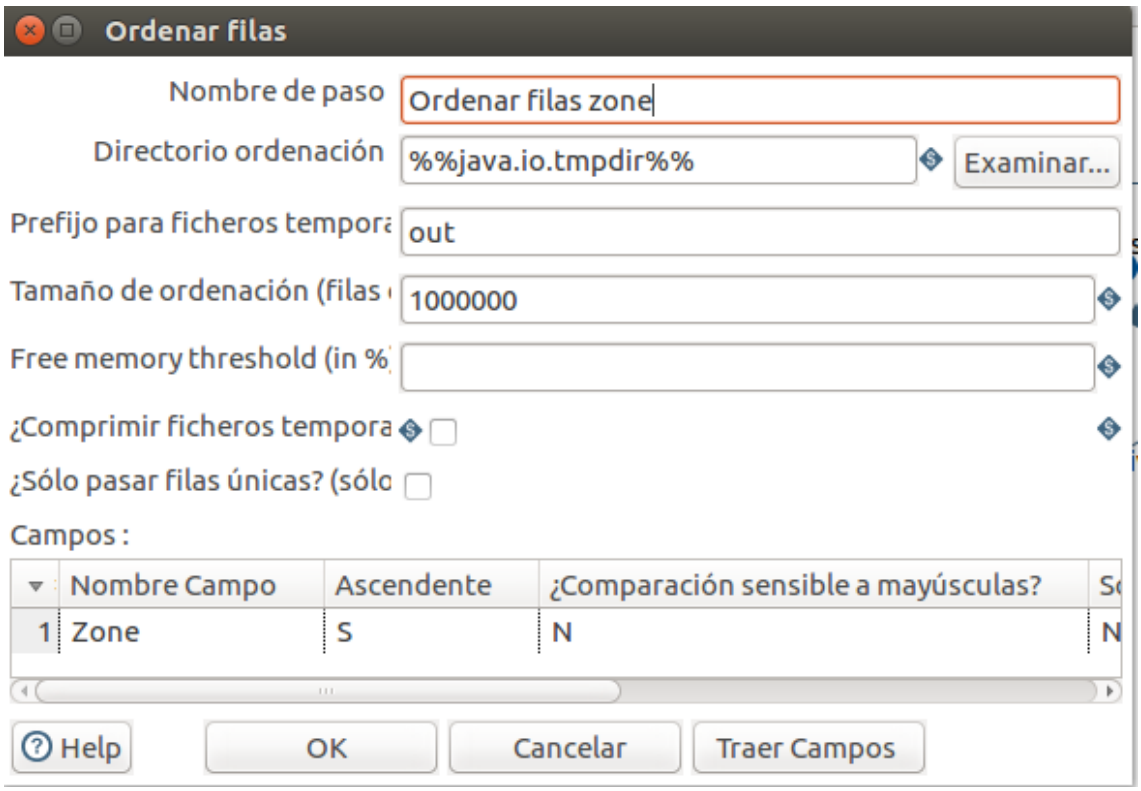


Paso 5: Salida tabla

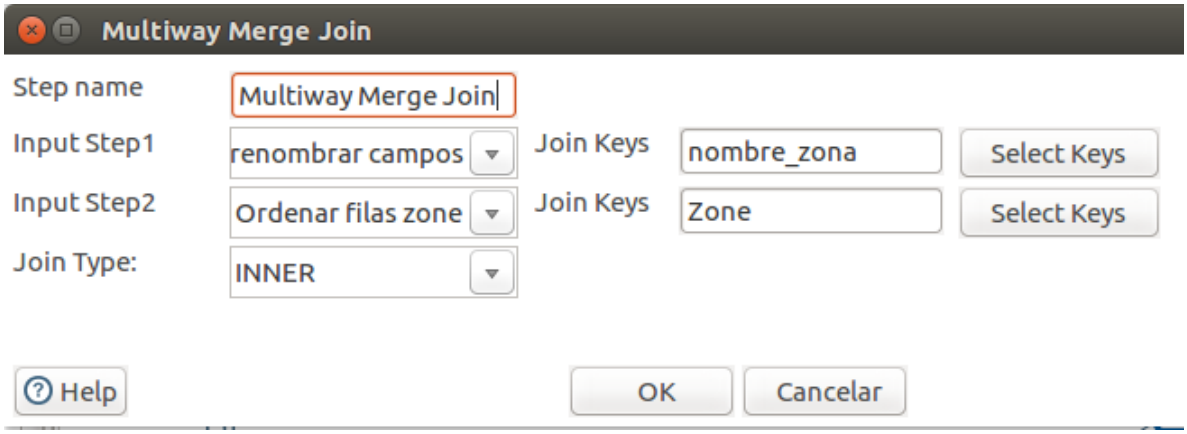
Finalmente, para cargar los datos en el data warehouse se deben definir la conexión al schema MySQL y la tabla en donde guardarlos.



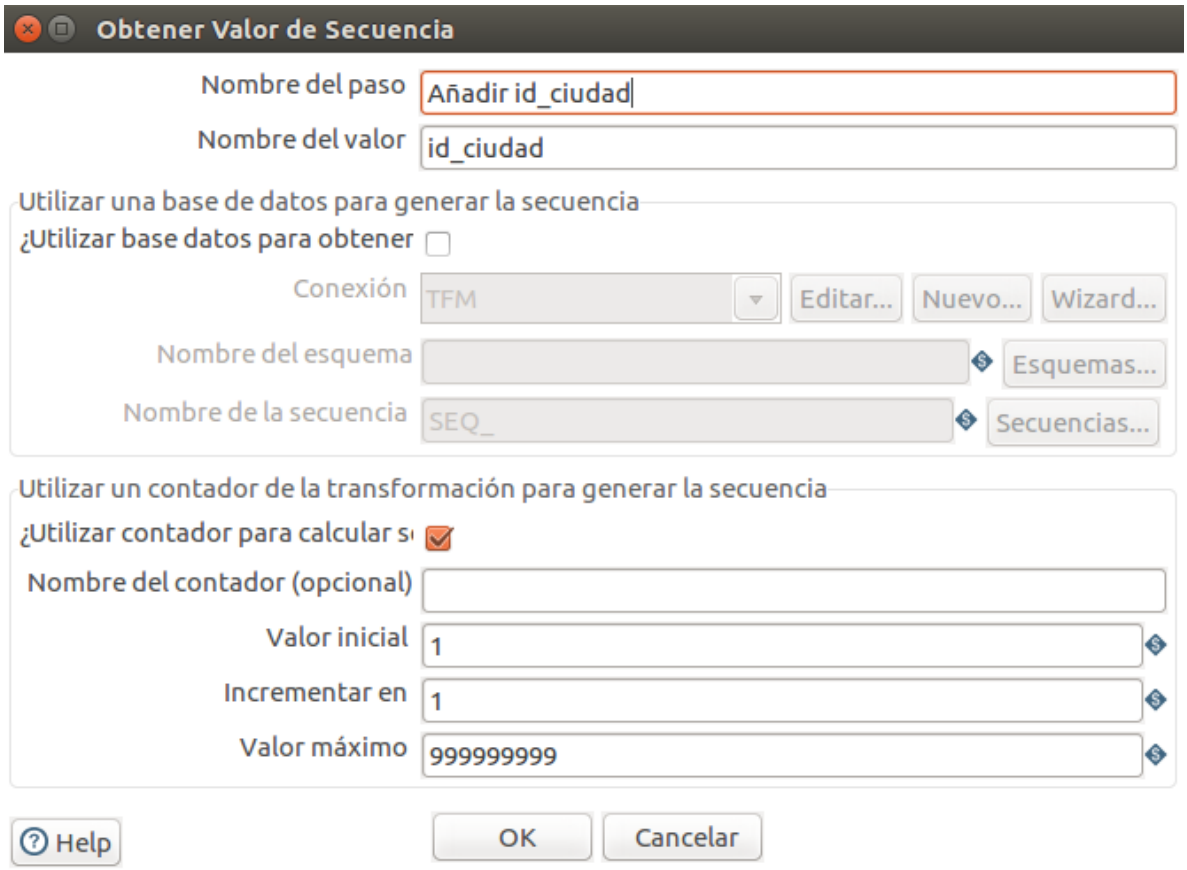
Rama inferior:
Paso 3: Ordenación de los datos
La dimensión ciudad tiene una relación con la de zona por lo que se debe asociar el identificador de la zona con la ciudad. Así, se vuelven a ordenar los datos, pero en esta oportunidad, por zona ya que en el próximo salto los datos deben estar ordenados por la clave de unión.



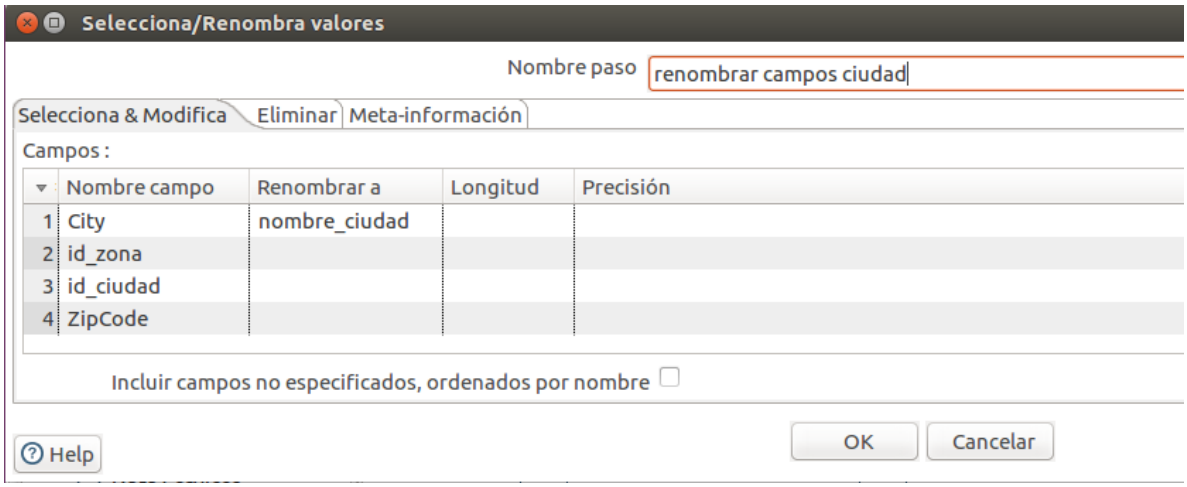
Paso 4: Multiway Merge Join
Se deben unir los datos de las ciudades con el identificador de la zona a la que pertenece.



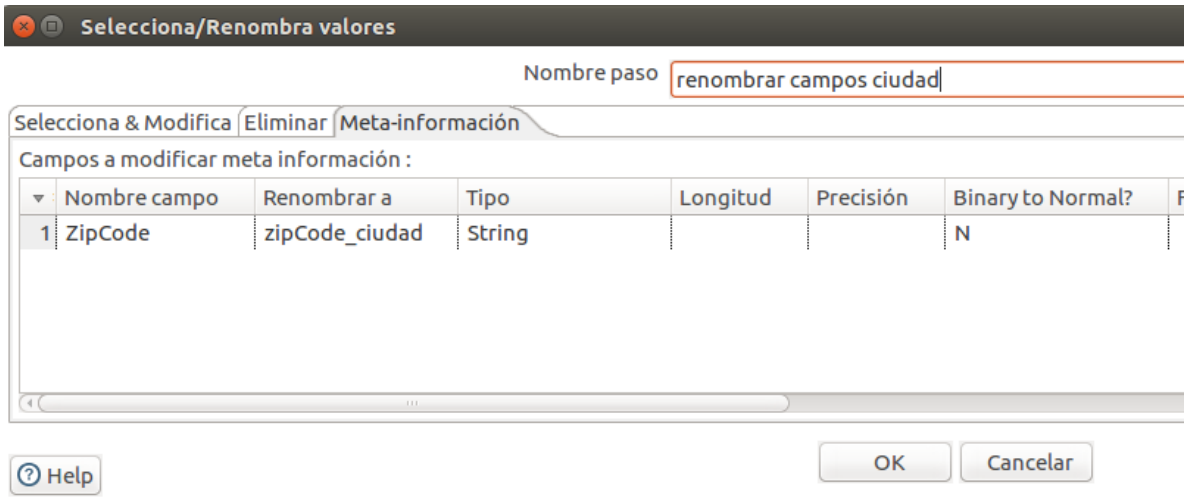
Paso 5: Añadir id
Se necesita añadir un identificador para cada ciudad usando el objeto añadir secuencia:



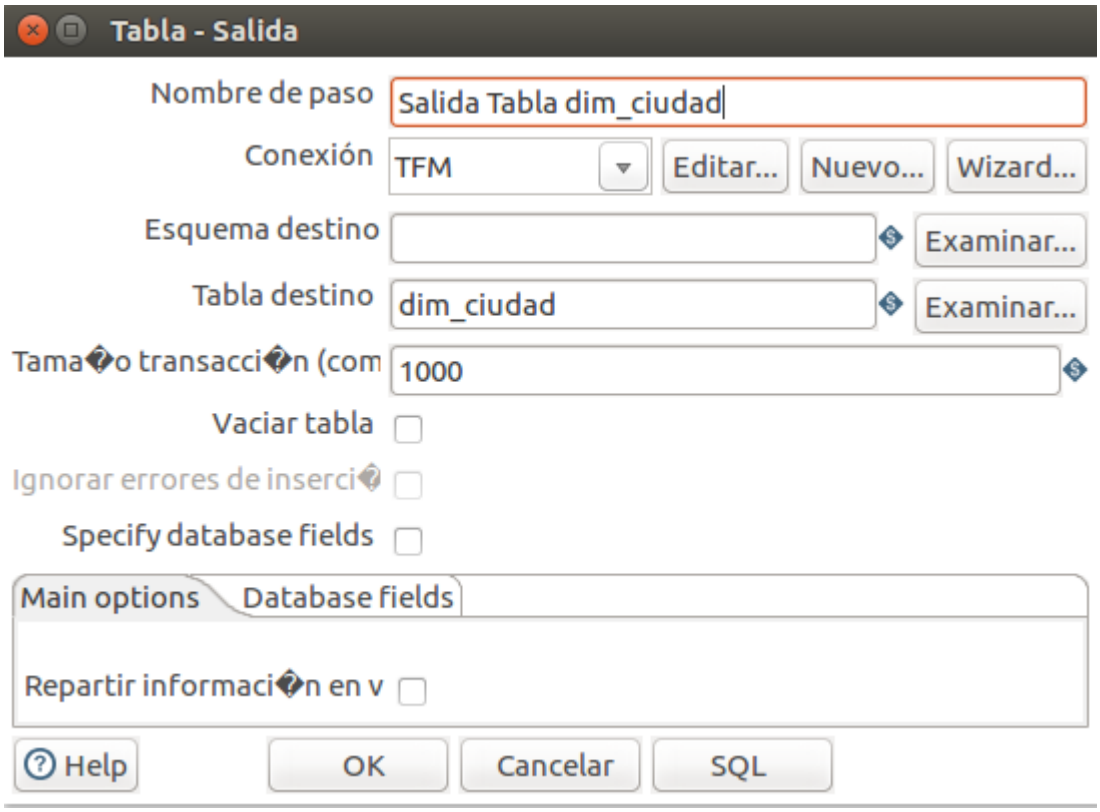
Paso 6: Renombrar campos
Es necesario renombrar los campos de la entrada de datos según los atributos correspondientes en la tabla dim_ciudad del data warehouse.



Por defecto, Spoon toma al campo zipCode como Integer. Se debe modificar esto para que las ciudades cuyo código postal comiencen con 0, sean encontradas a la hora de cargar los datos en la tabla de hechos. Por lo tanto, el tipo de datos debe ser String.

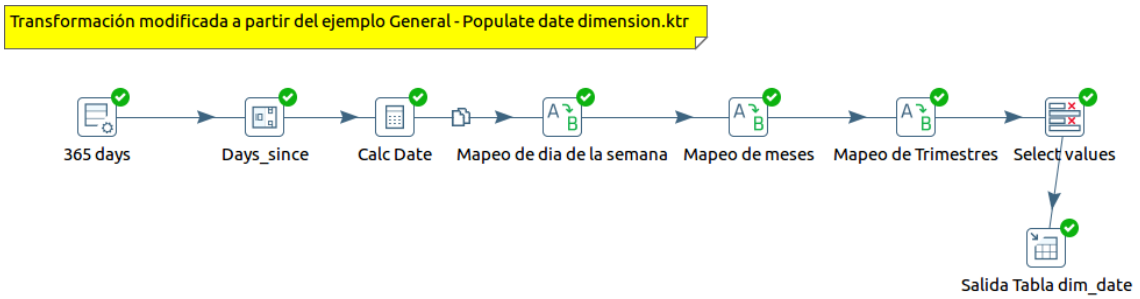


Paso 7: Salida tabla
Finalmente, para cargar los datos en el data warehouse se deben definir la conexión al schema MySQL y la tabla en donde guardarlos.



Dimensión Date

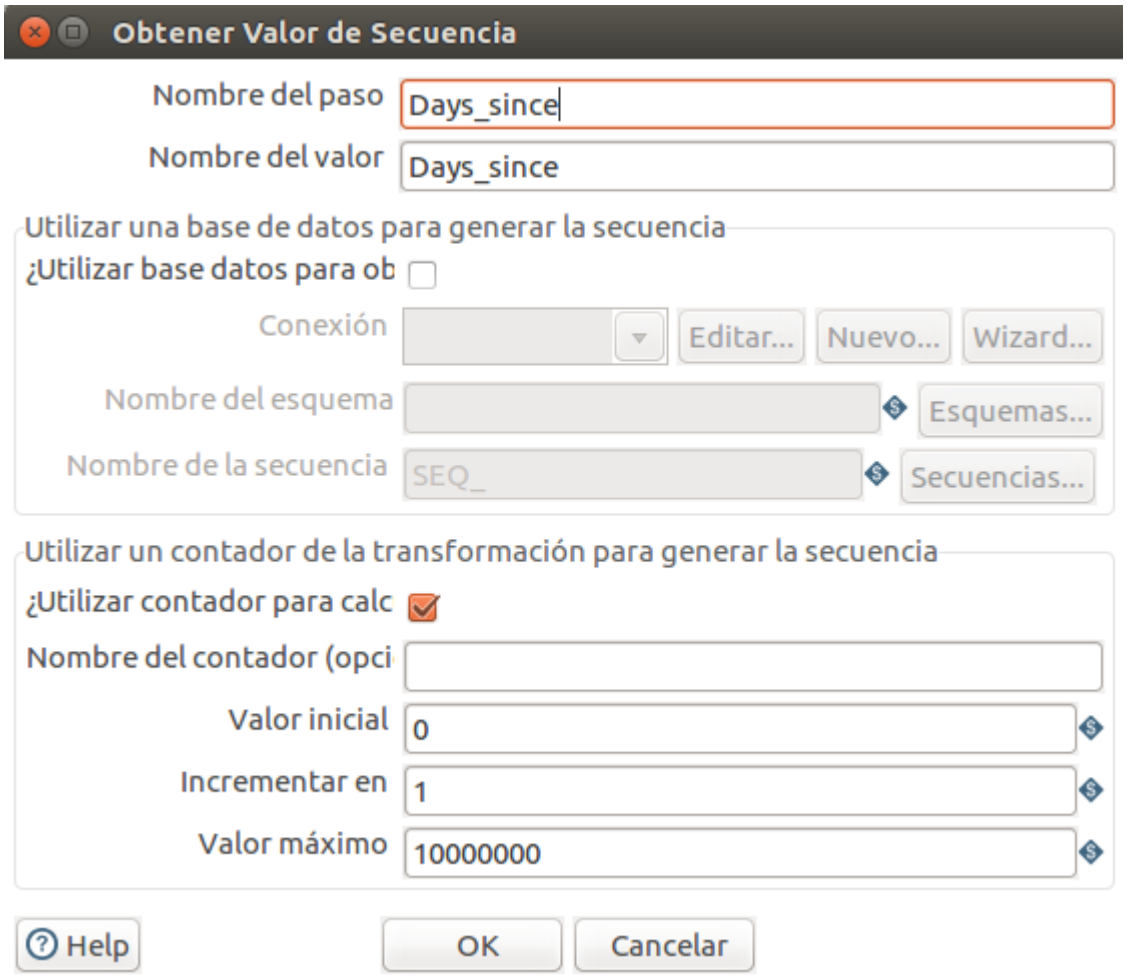
El archivo correspondiente a este proceso es **t_d_date.ktr**. Se utilizó como punto de partida un ejemplo que provee Kettle: la transformación General – Populate date dimension.ktr que se encuentra en el directorio Samples/Transformations del módulo Data Integration de Pentaho.



Paso 1: Generación de días

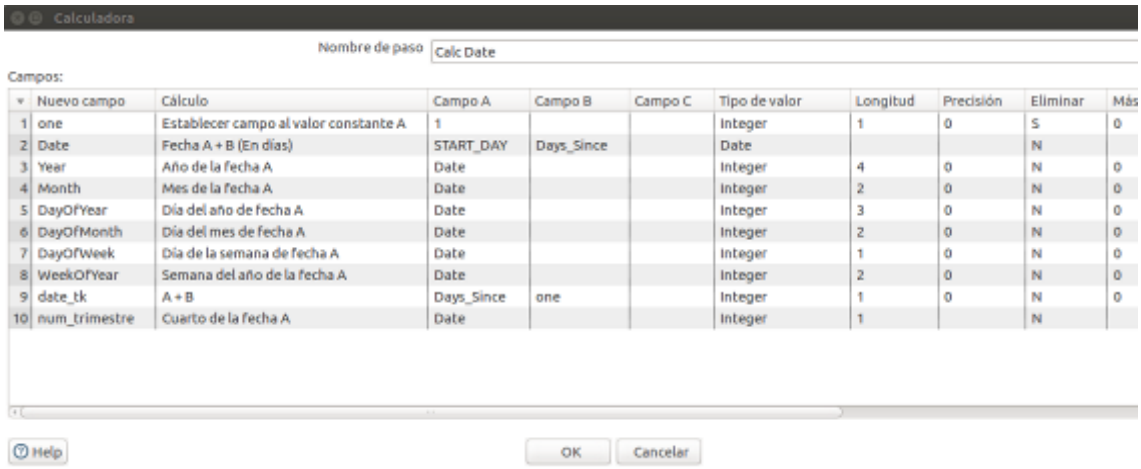
El objeto generar filas permite generar registros vacíos o iguales. En este caso, se busca generar los 365 días del año 2017. Se define un campo START_DAY con el valor inicial del 1 de Enero del año 2017. En la figura siguiente se puede ver como se definen estos parámetros:

Paso 2: Añadir secuencia
Se añade un contador que se va a sumar al campo definido previamente para obtener las fechas.

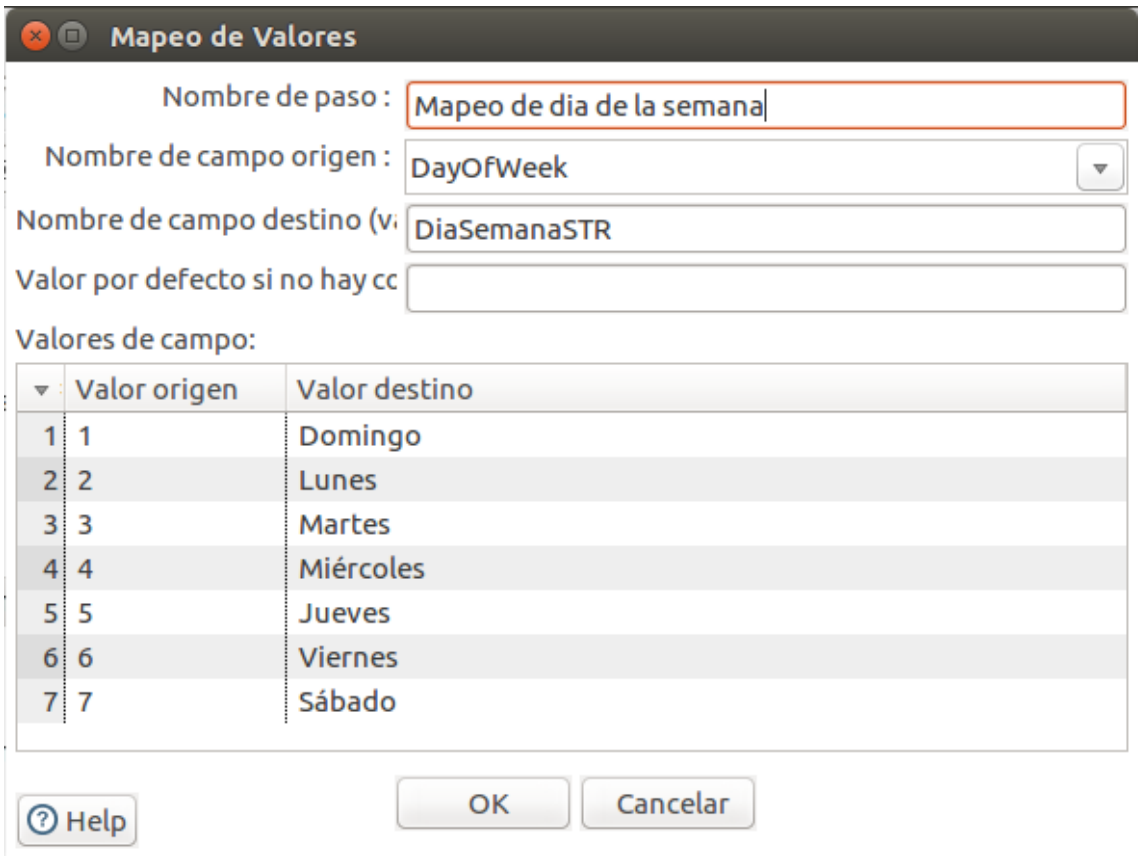


Paso 3: Cálculos
Con el objeto Calculadora se pueden realizar todos los cálculos necesarios para obtener las fechas. Como se ve a continuación, se obtienen los distintos campos correspondientes al Año, Mes, Día del mes, día del año, entre otros. También, se crea el campo Date que surge

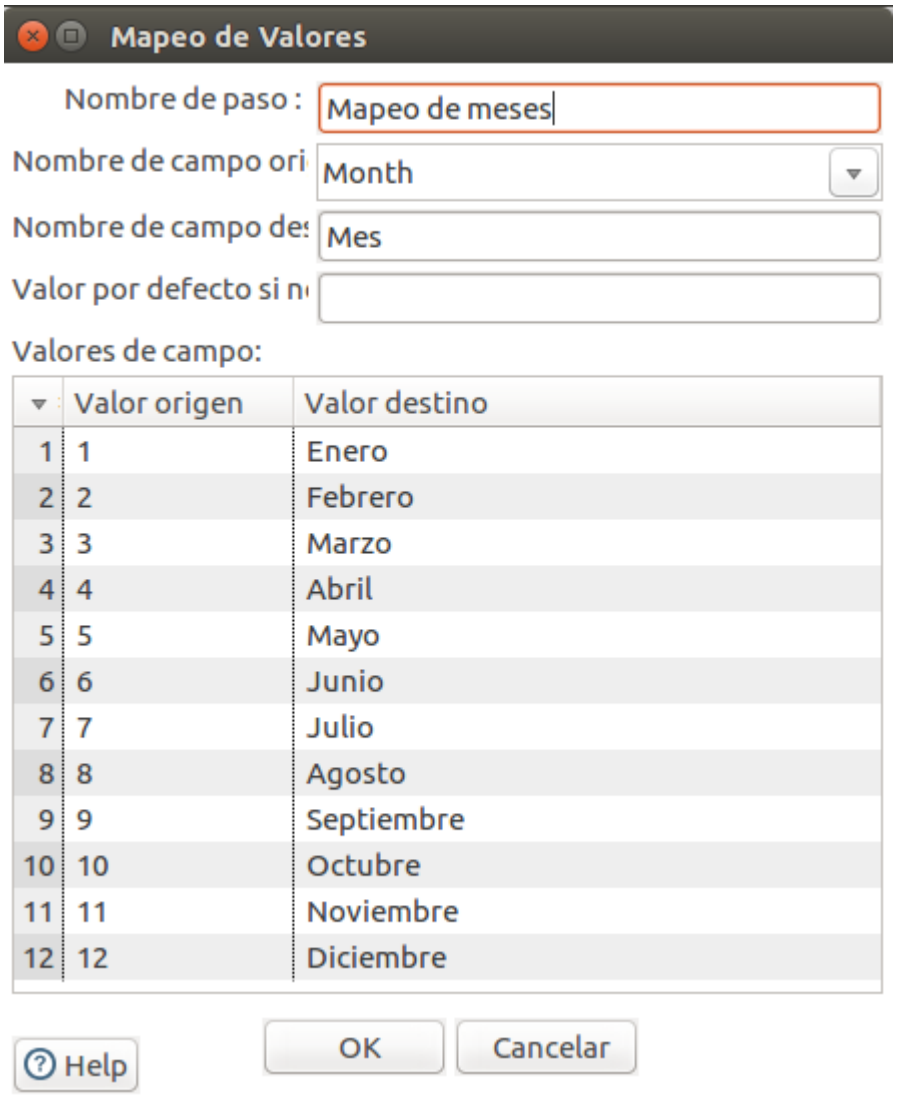
de la suma del valor Start_Day definido en el paso 1 con el valor del contador Days_since del paso 2.



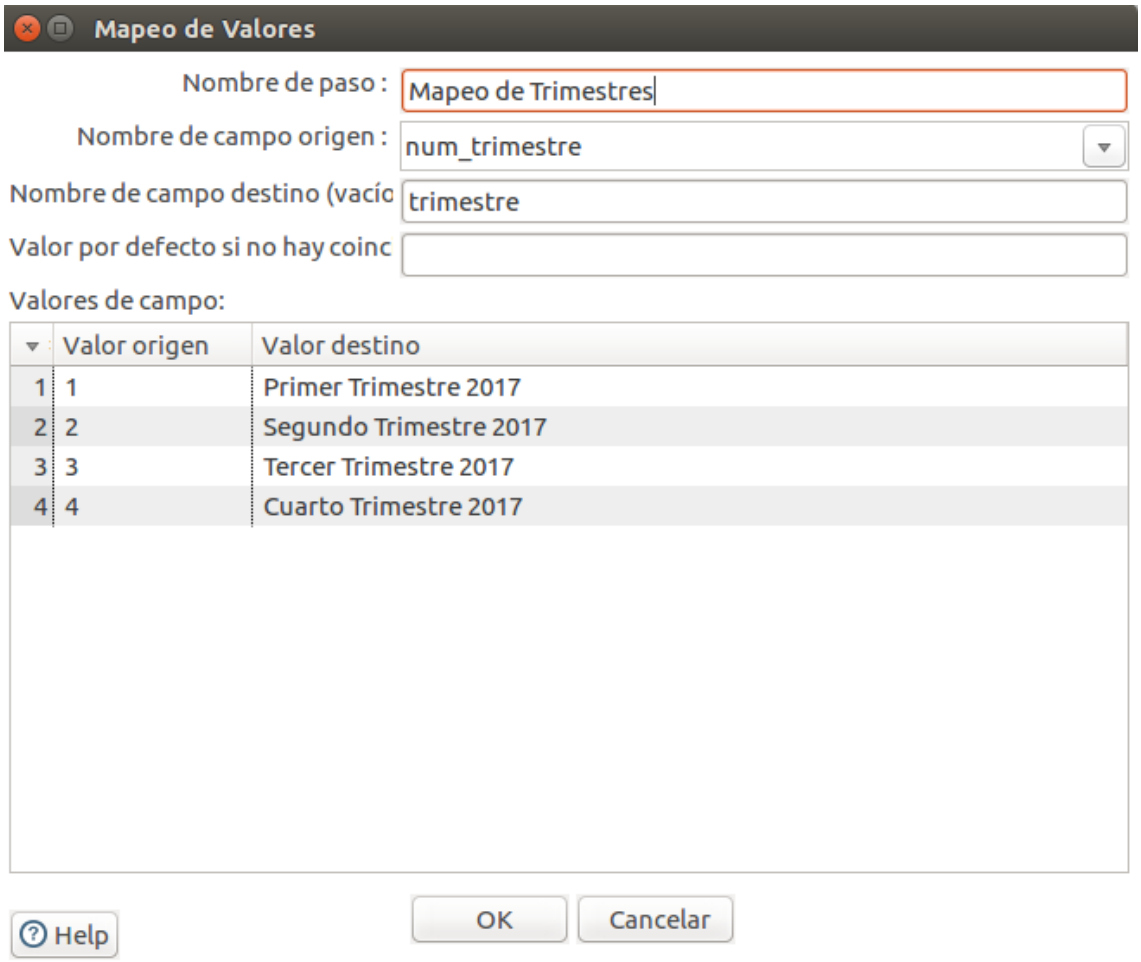
Paso 4: Mapeo del día de la semana
En la dimensión del data warehouse, se desea almacenar el nombre del día de la semana y no un número. Por lo tanto, se debe mapear el valor del día de la semana obtenido del paso anterior.



Paso 5: Mapeo del mes
Al igual que en el paso anterior, se desea almacenar el nombre del mes y no un número. Por lo tanto, se debe mapear el valor obtenido en el paso 3.

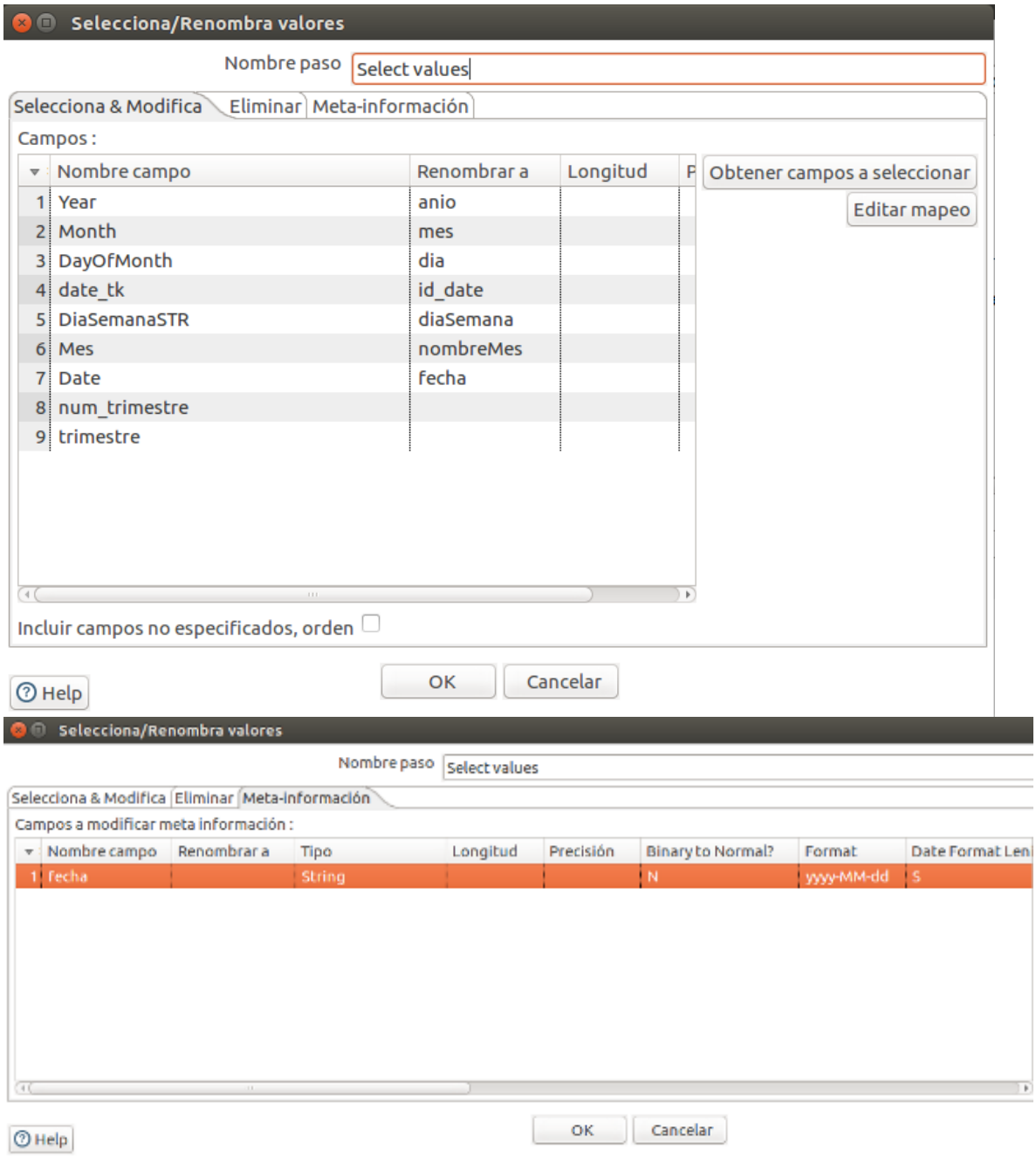


Paso 6: Mapeo del trimestre
Al igual que en el paso anterior, se desea almacenar una descripción del trimestre y no sólo un número. Por lo tanto, se debe mapear el valor obtenido en el paso 3.



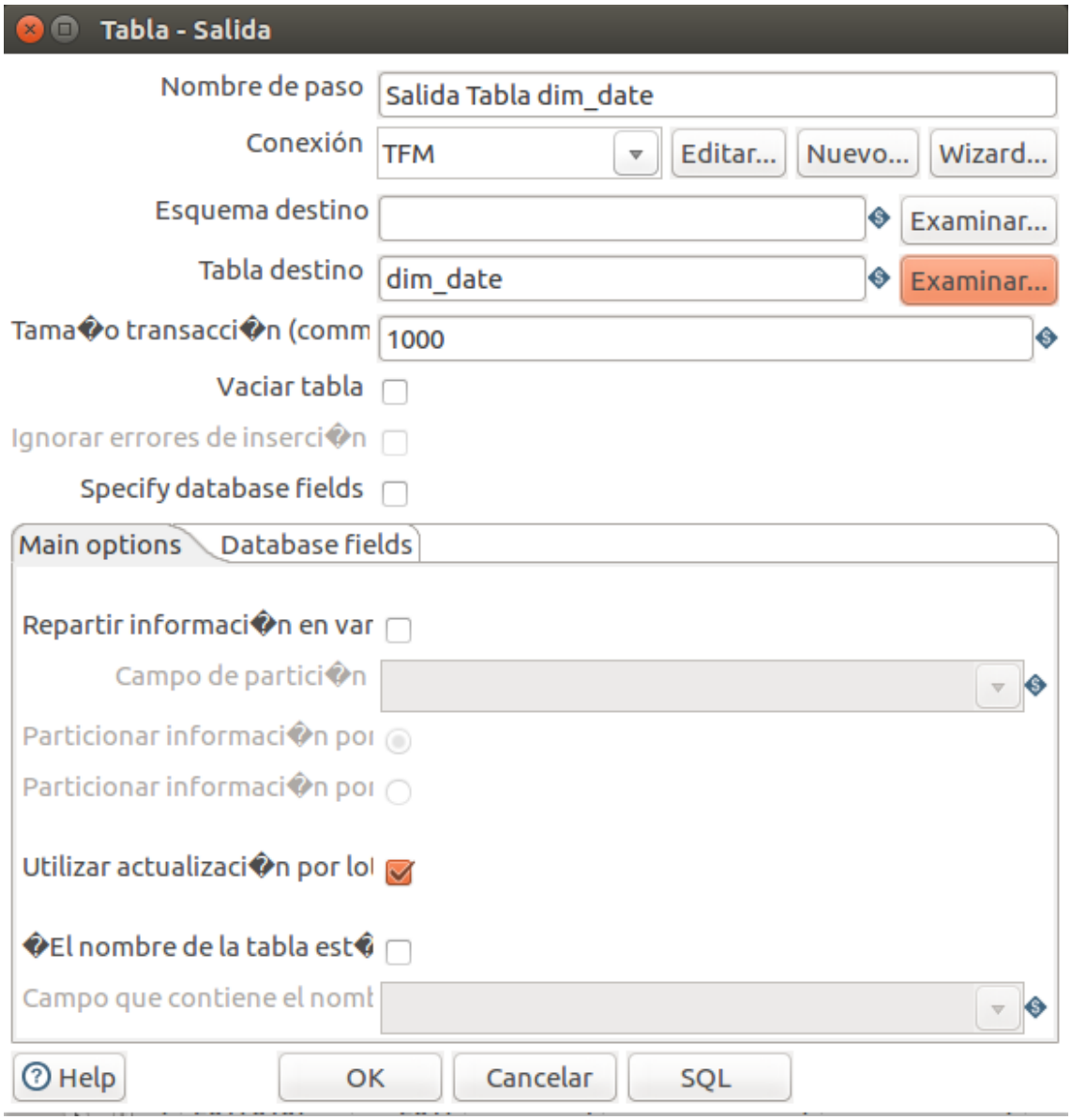
Paso 7: Renombrar campos

Es necesario renombrar los campos de la entrada de datos según los atributos correspondientes en la tabla dim_date del data warehouse.



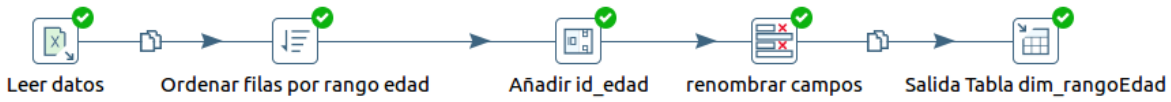
Paso 8: Salida tabla

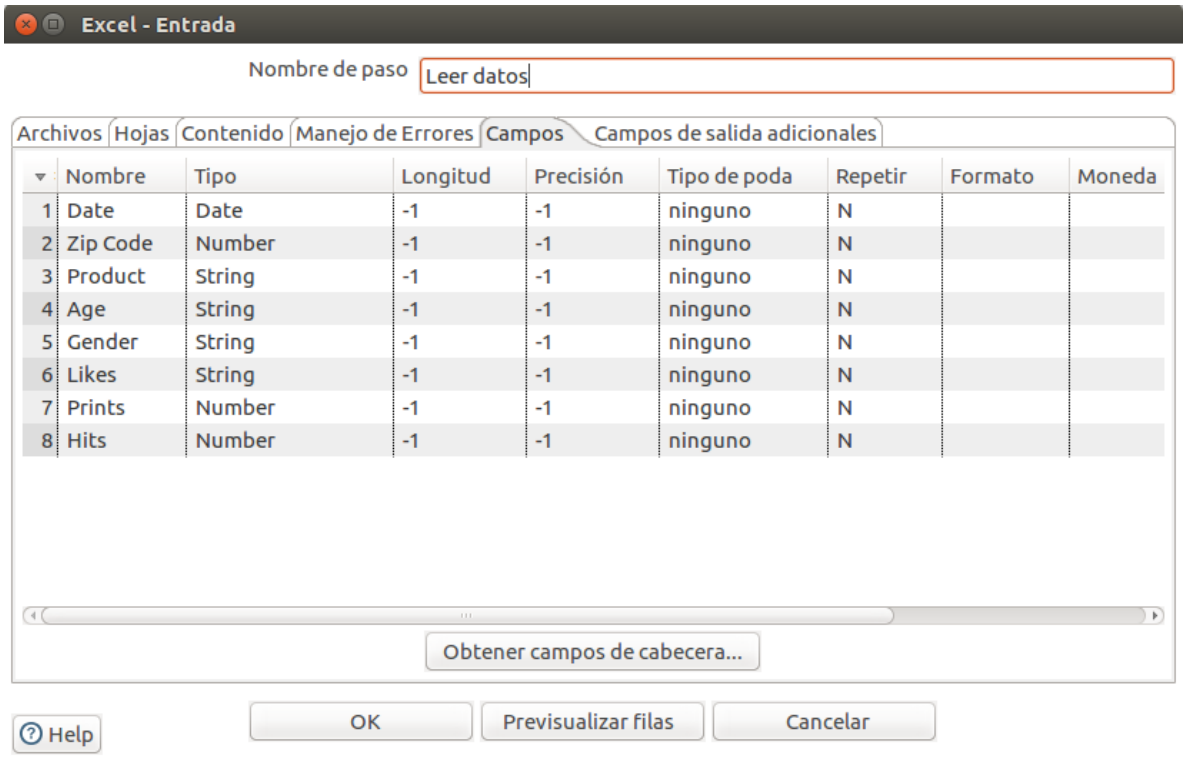
Finalmente, para cargar los datos en el data warehouse se deben definir la conexión al schema MySQL y la tabla en donde guardarlos.



Dimensión Edad

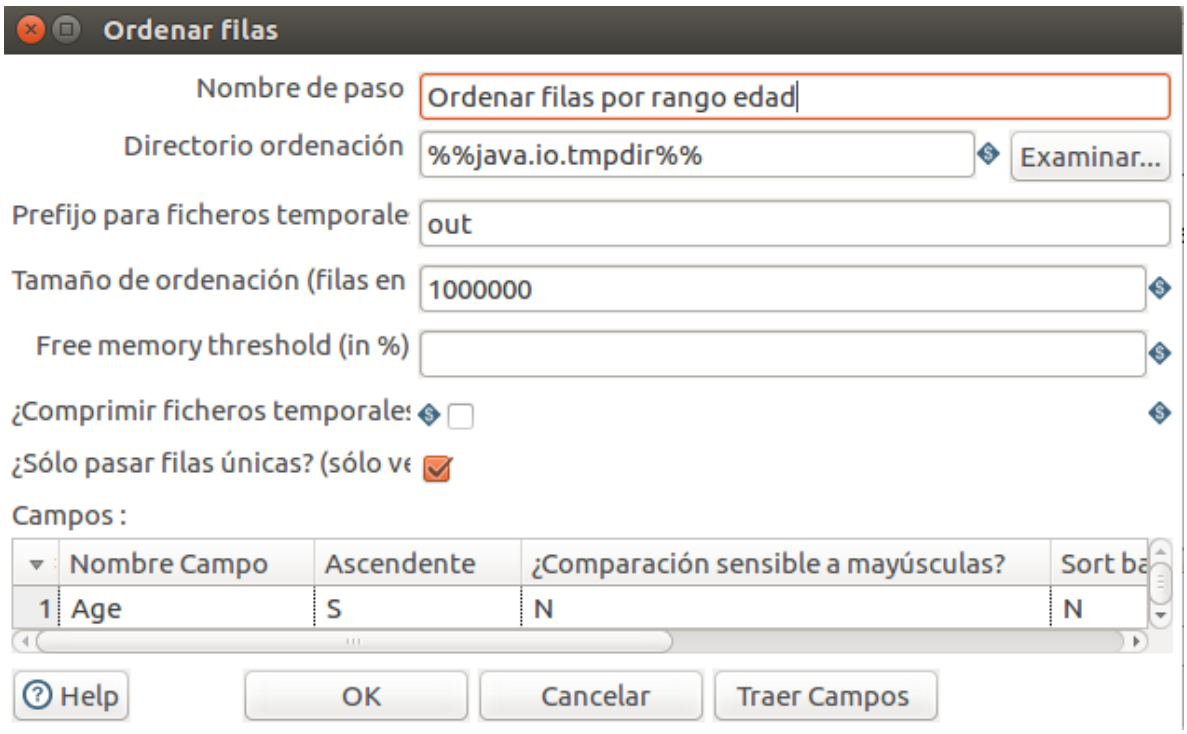
El archivo correspondiente a este proceso es **t_d_edad.ktr**.
Paso 1: Leer entrada de datos





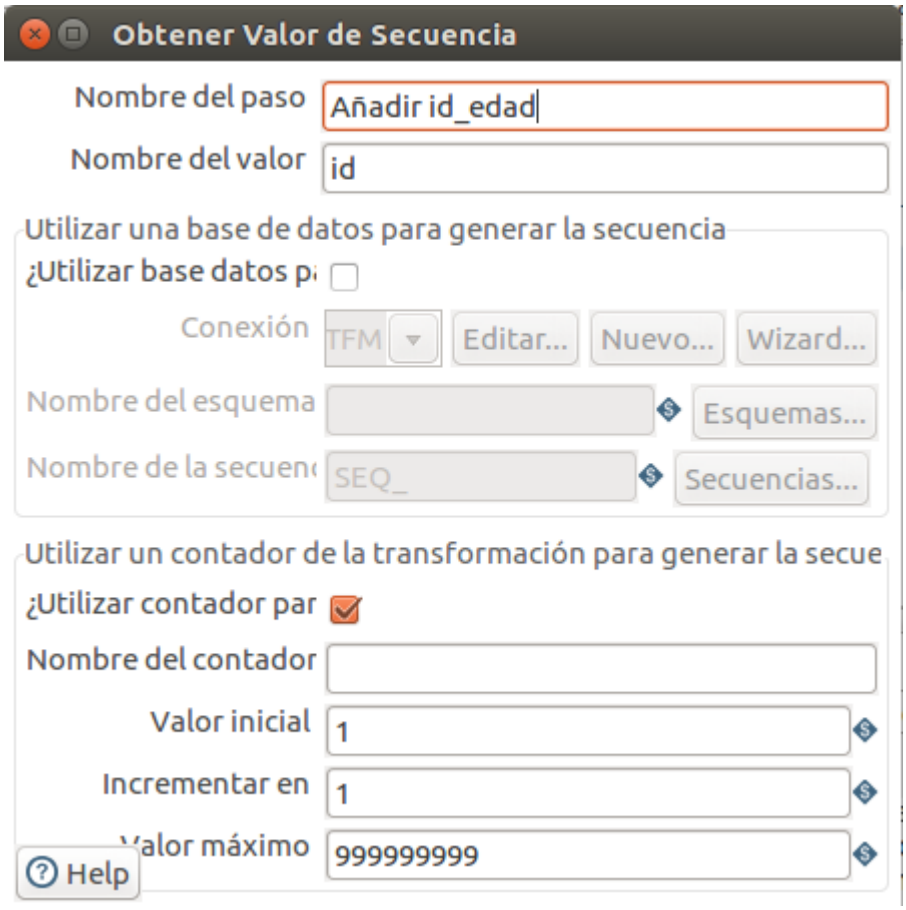
Paso 2: Ordenación de los datos

En este paso lo que se busca es ordenar los datos y a su vez, eliminar los duplicados del campo *Age* tildando la opción "¿Sólo pasar filas únicas?".

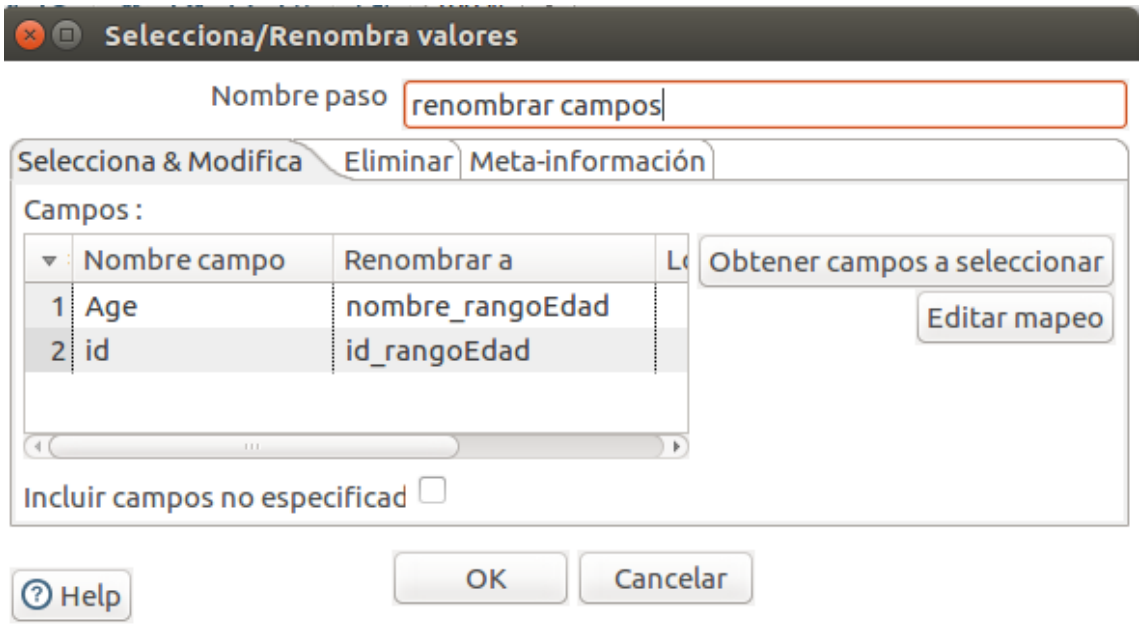


Paso 3: Añadir id

Se necesita añadir un identificador para cada rango de edad usando el objeto añadir secuencia:

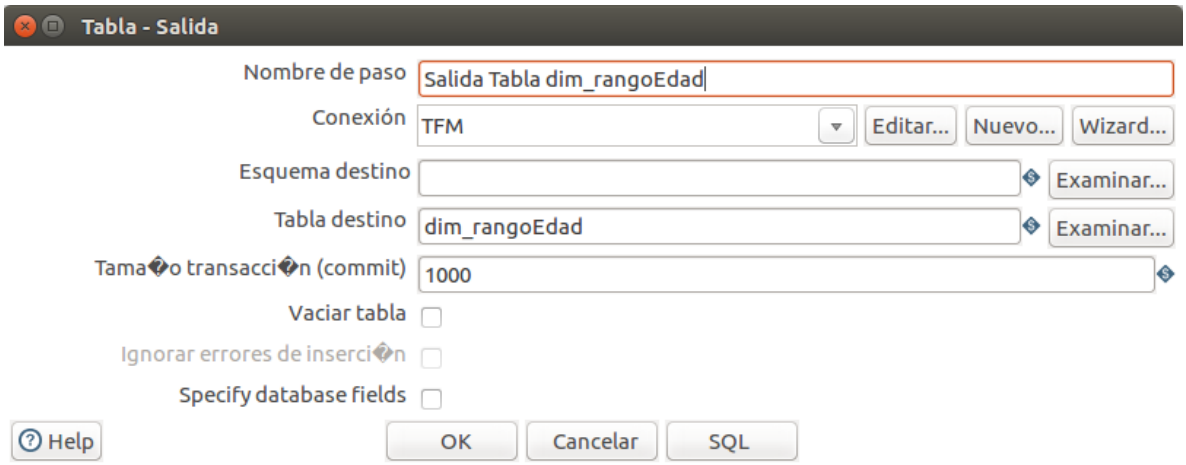


Paso 4: Renombrar campos
Es necesario renombrar los campos de la entrada de datos según los atributos correspondientes en la tabla dim_edad del data warehouse.



Paso 5: Salida tabla

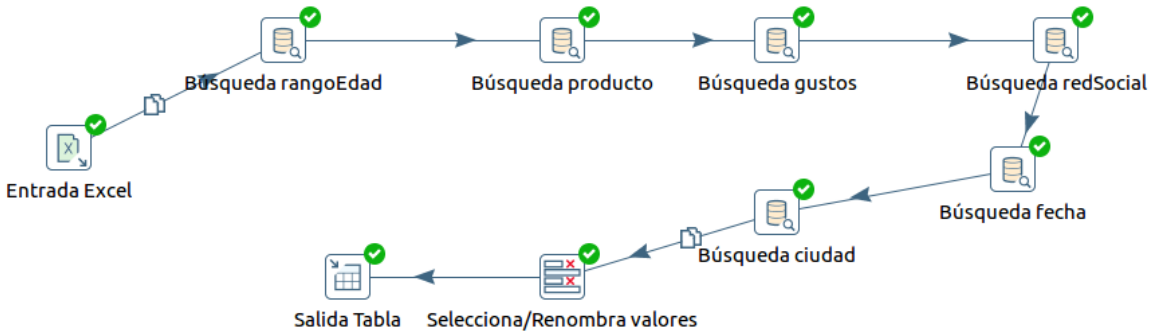
Finalmente, para cargar los datos en el data warehouse se deben definir la conexión al schema MySQL y la tabla en donde guardarlos.

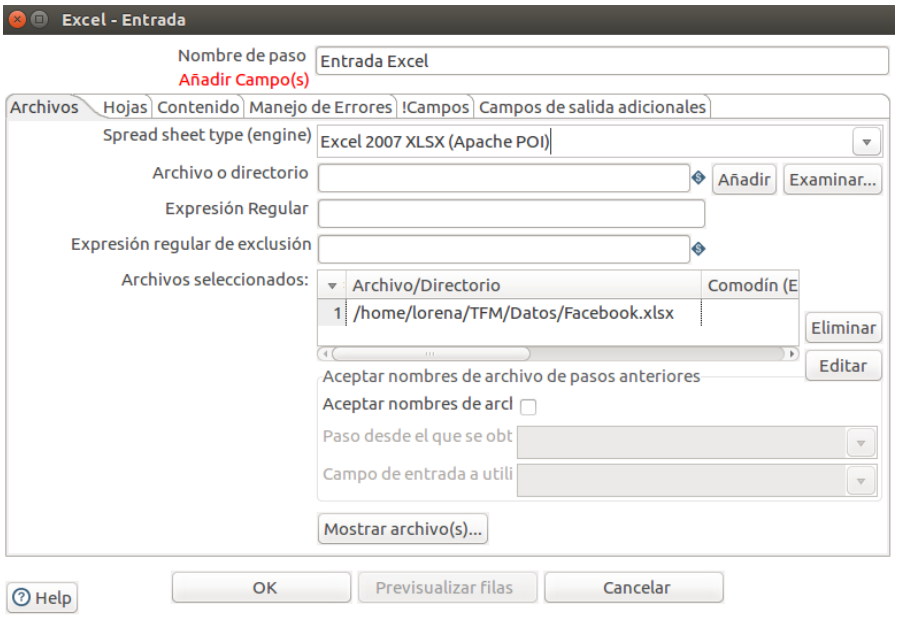


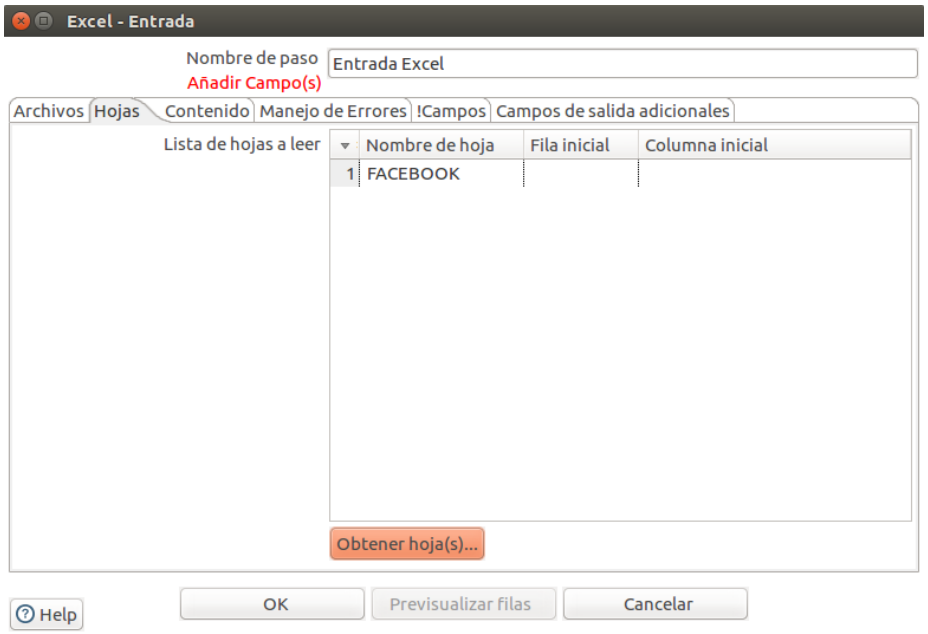
Hechos

El archivo correspondiente a este proceso es **t_h_facebook.ktr.** Los archivos de las transformaciones de las otras redes sociales son **t_h_twitter.ktr**, **t_h_instagram.ktr** y **t_h_youtube.ktr**.
Se mostrarán los pasos usando la red social Facebook pero son similares para los otras redes.

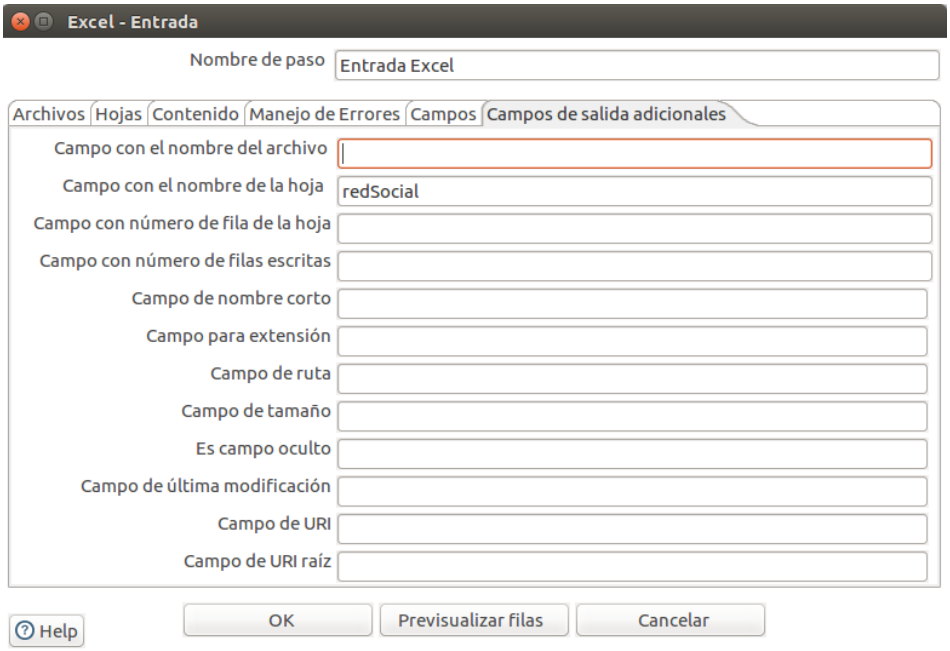
Paso 1: Leer entrada de datos







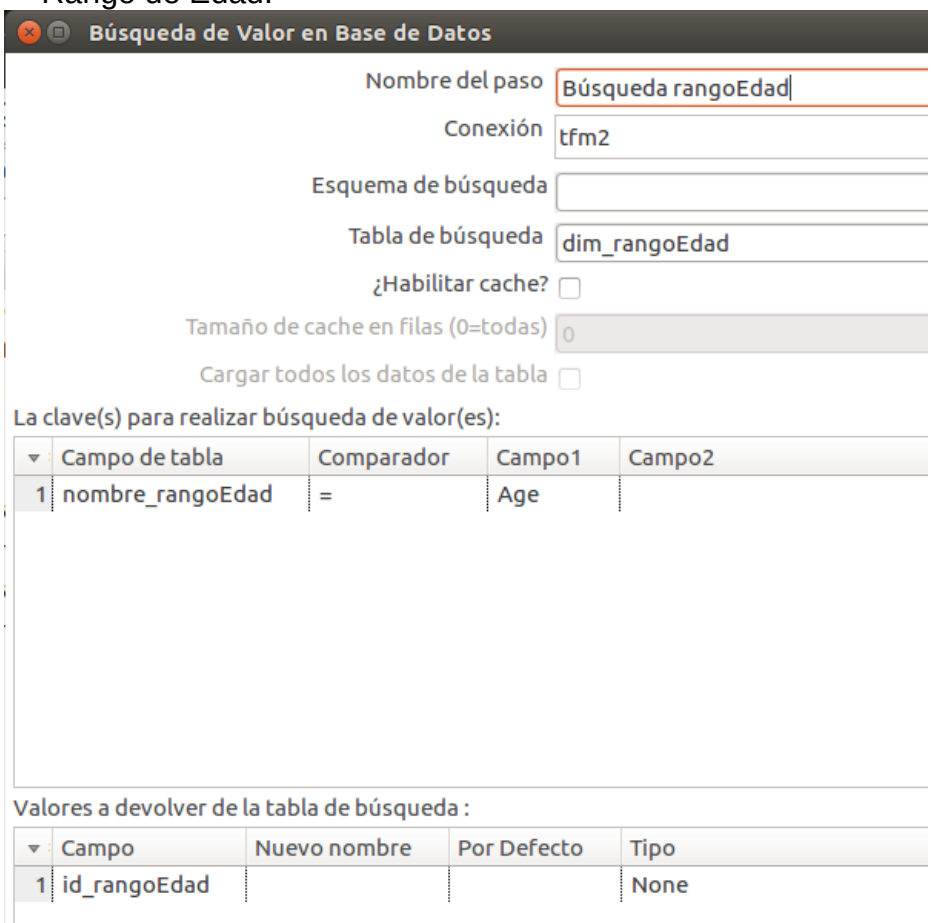
Se necesita saber de que red social se obtuvieron los datos. Por lo tanto, se crea un campo de salida denominado redSocial que va a contener el nombre de la Hoja Excel que se lee como entrada.



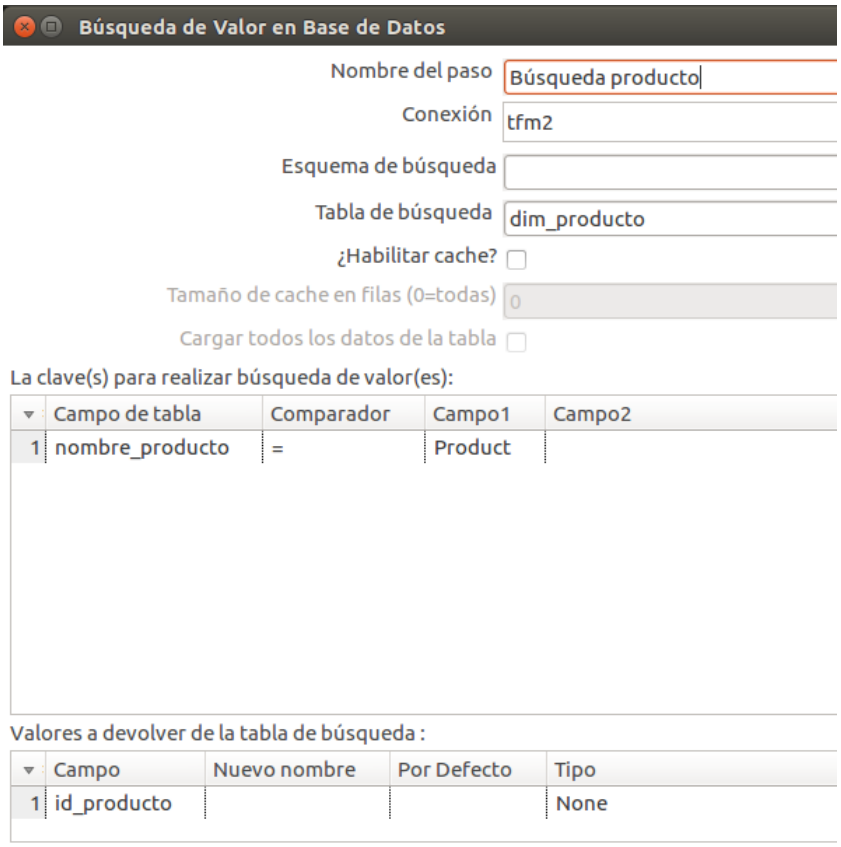
Paso 2 al 7: Búsqueda de los identificadores

En la tabla de hechos no se desea guardar los datos tal cual llegan en la hoja de datos. Al contrario, se busca asociarlos con las distintas dimensiones a traves de los identificadores correspondientes.

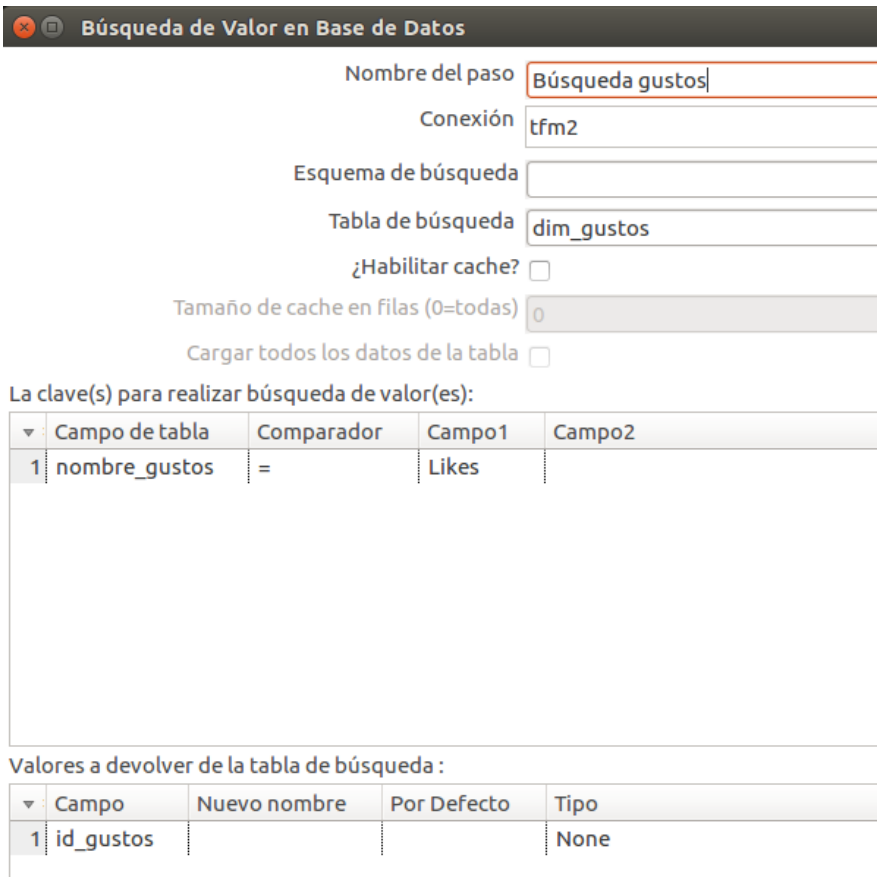
- Rango de Edad:



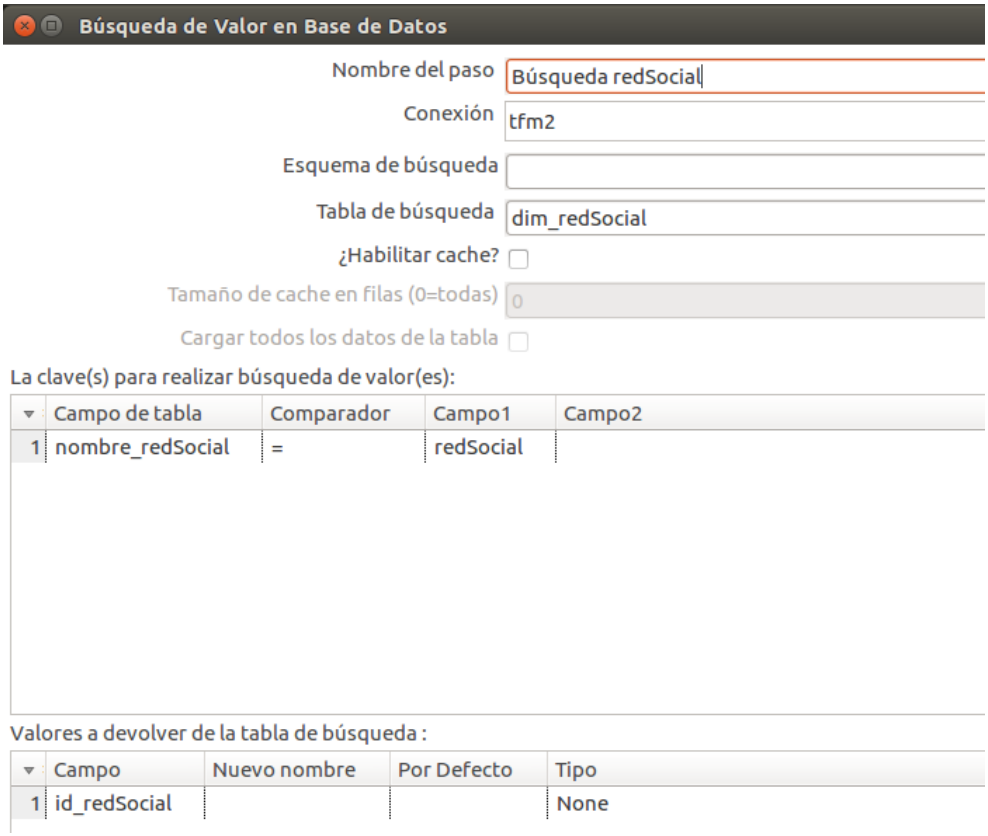
- Producto:



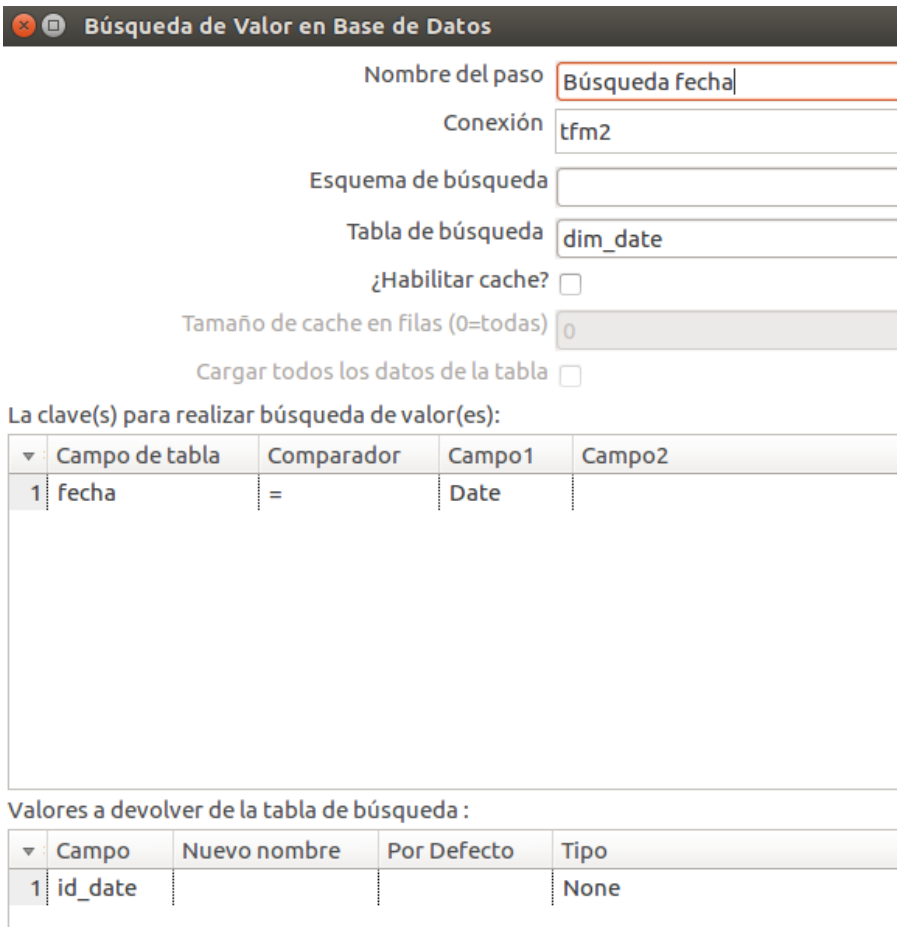
- Gustos:



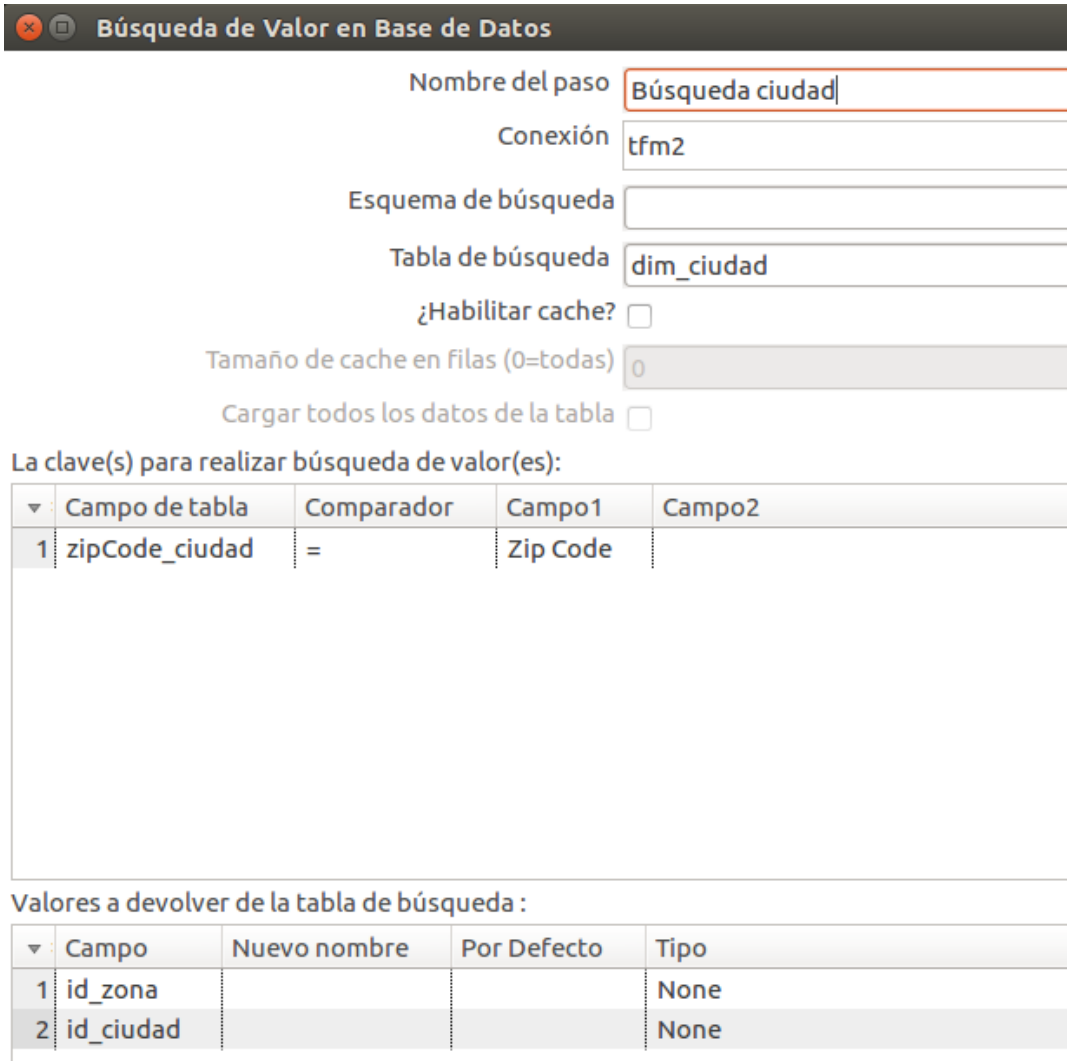
- Red Social:



- Fecha:



- Ciudad:



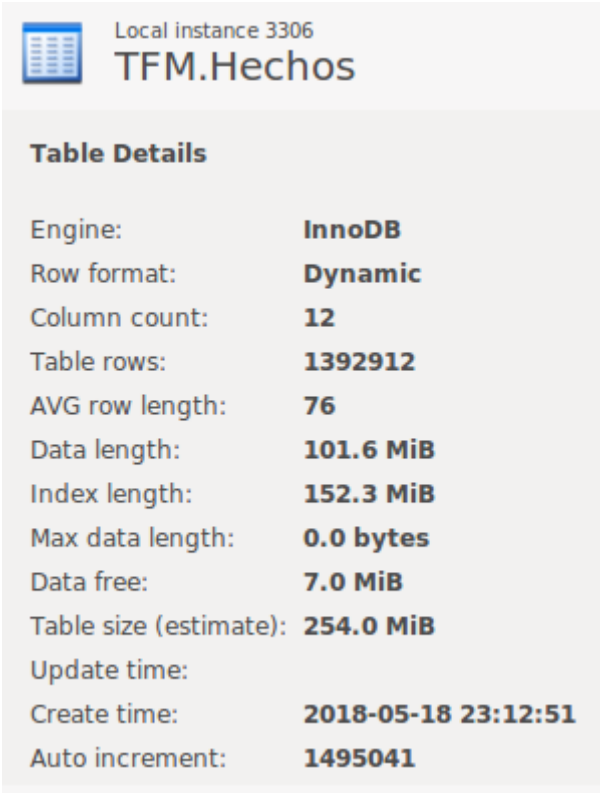
Paso 8: Renombrar campos

Es necesario renombrar los campos de la entrada de datos según los atributos correspondientes en la tabla hechos del data warehouse.

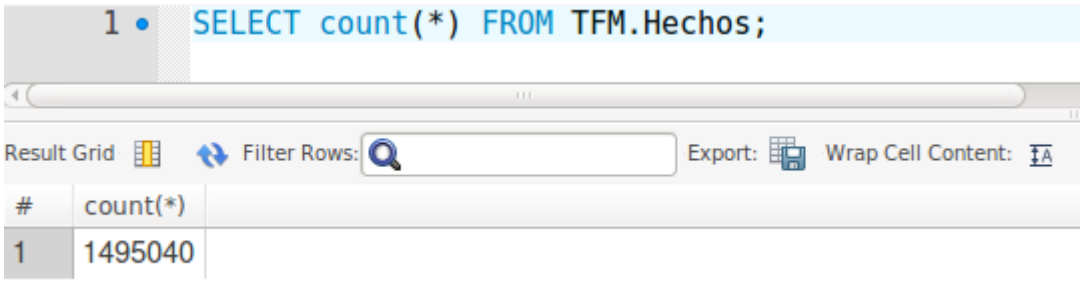
Paso 9: Salida tabla

Finalmente, para cargar los datos en el data warehouse se deben definir la conexión al schema MySQL y la tabla en donde guardarlos.

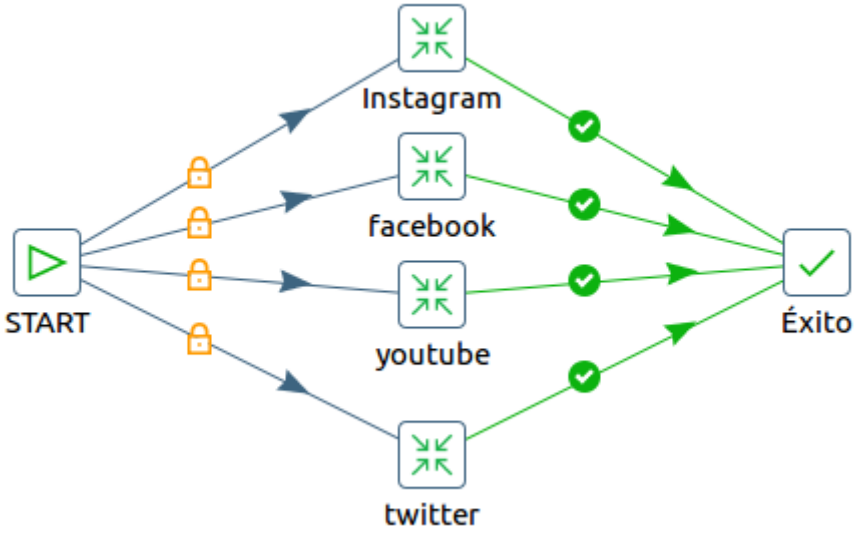
Antes de crear el trabajo que coordinará las transformaciones de las cuatro redes sociales, se debe verificar que se cargaron correctamente los datos de la red social Facebook.
Al acceder a la información de la tabla que provee MySQL Workbench, se verifica que el próximo valor del identificador autoincremental está correcto, corresponde a 1.495.041. Pero la cantidad total de filas, no: 1.392.912.



Esto se debe a un bug de la herramienta que calcula aproximadamente el número de filas para esquemas InnoDB, como es este caso. Se puede ahondar más de este bug en https://bugs.mysql.com/bug.php?id=46422.
La solución es utilizar una sentencia SQL para contar los registros:



Una vez verificado que el proceso anterior carga correctamente los datos, se crea un trabajo para coordinar las cuatro transformaciones correspondientes a los datos de las cuatro redes sociales analizadas. Simplemente consiste en agregar un punto de inicio que lanzará las transformaciones. El diagrama resultante puede verse a continuación:


Instagram
facebook
youtube
twitter
START
Éxito

Anexo 4: Cubo OLAP

La creación del cubo OLAP para el análisis de datos se hará utilizando el módulo *Schema Workbench* de Pentaho.

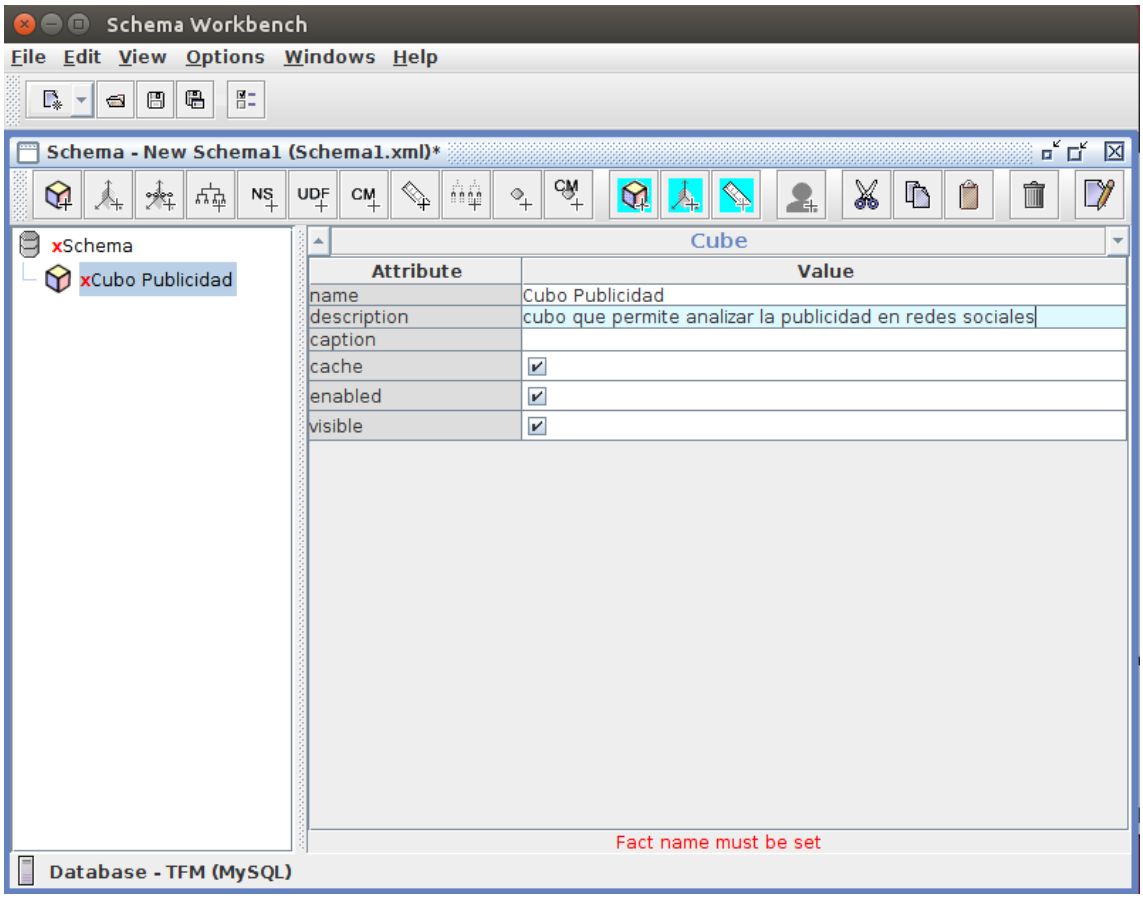
Para iniciarlo, basta con escribir:

```
lorena@lorena-VirtualBox:~$ cd pentaho/schema-workbench/
lorena@lorena-VirtualBox:~/pentaho/schema-workbench$ ./workbench.sh
```

El primer paso consiste en seleccionar o crear una conexión al data warehouse. En este caso, una conexión MySQL. Si el lector recibe el error de la falta del driver correspondiente, en el Anexo 2, sección dificultades enfrentadas, puede encontrar la solución.

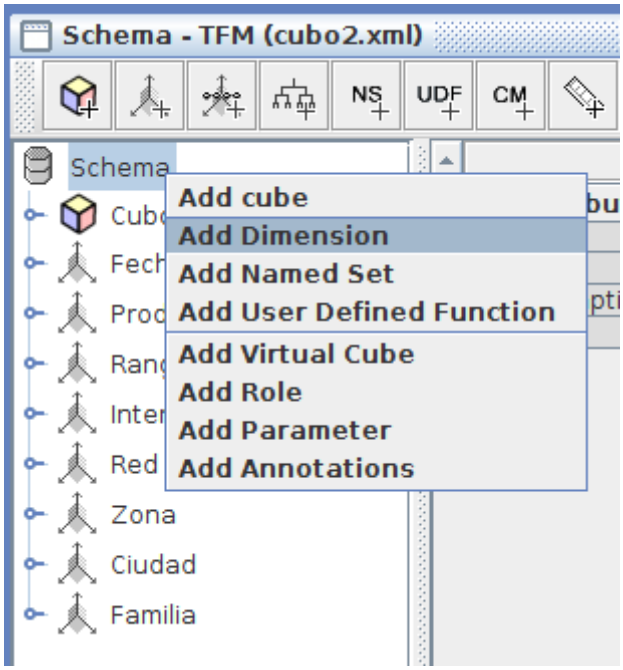
Una vez creada la conexión, se necesita iniciar un esquema, simplemente seleccionando del menú File, New Schema.

Por último, se agrega un nuevo cubo como se puede ver a continuación:

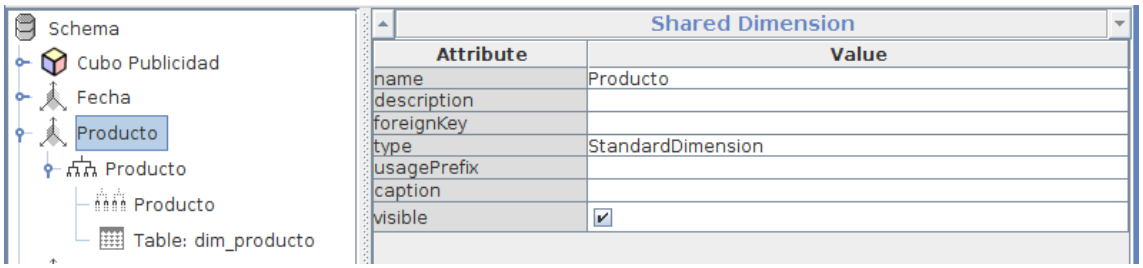


Cabe aclarar que un cubo solo puede estar asociado a una tabla de hechos.

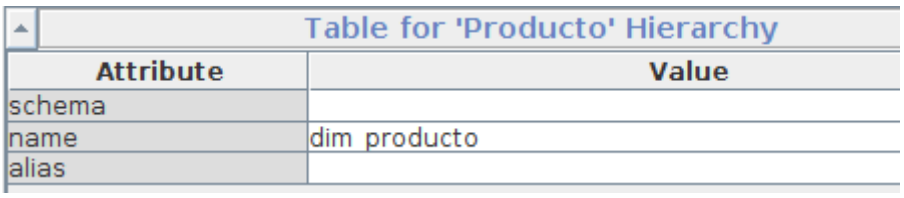
Para agregar una nueva dimensión a nivel de esquema, se debe selccionar el mismo y haciendo click con el botón derecho, elegir Add Dimension.



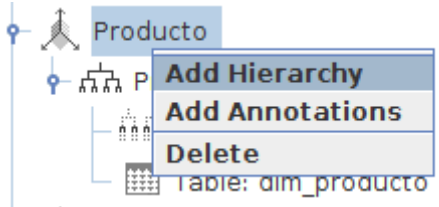
Se deben definir sus propiedades, como se ve a continuación:

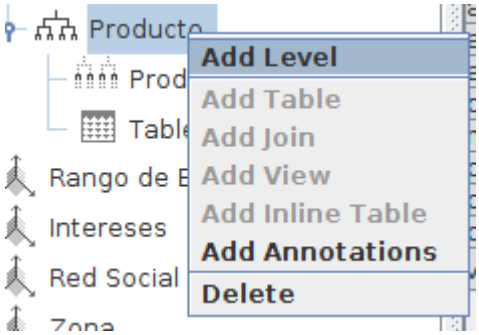


En el caso que se esté definiendo una dimensión temporal, se puede elegir como Type, TimeDimension.
Una vez hecho esto, se debe agregar una jerarquía, un nivel y definir a que tabla de dimensión estará asociada.
Primero, se define la tabla de dimensión, eligiendo Add Table y definiendo a qué tabla estará asociada. En este caso, dim_producto:



Para agregar una jerarquía se debe seleccionar Add Hierarchy como se ve a continuación:





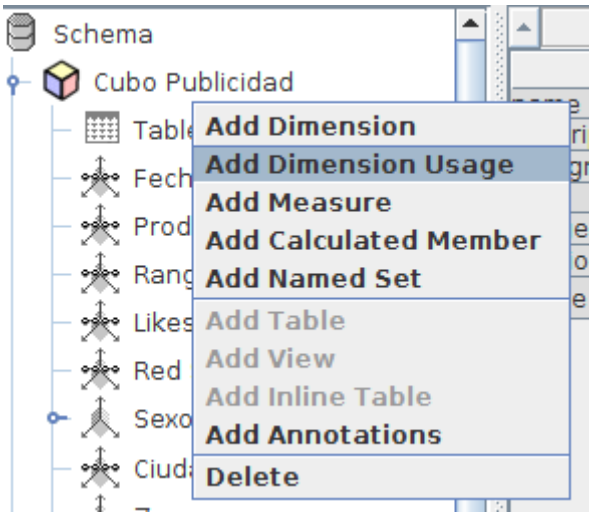
Luego, se agrega un nivel y se definen sus atributos:

Level for 'Producto' Hierarchy

Attribute	Value
name	Producto
description	
table	
column	id_producto
nameColumn	nombre_producto
parentColumn	
nullParentValue	
ordinalColumn	
type	String
internalType	
uniqueMembers	☑
levelType	Regular
hideMemberIf	Never
approxRowCount	
caption	
captionColumn	
formatter	
visible	☑

De esta forma, se define cada dimensión, ya sea del cubo o a nivel de esquema.

Una vez definidas todas las dimensiones, hay que asociar las dimensiones del esquema a las que se desea acceder. Esto se hace eligiendo la opción Add Dimension Usage:

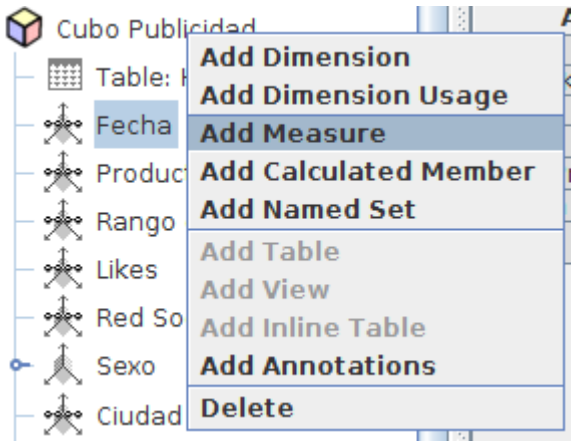


Se indica el campo de la tabla de Hechos que se relaciona con la dimensión (foreign key) y la dimensión definida previamente (source). En este caso, se hace uso de la dimensión Fecha (source) y se la relaciona a través del cambio id_fecha de la tabla de hechos.

Dimension Usage for 'Cubo Publicidad' Cube

Attribute	Value
name	Fecha
foreignKey	id_date
source	Fecha
level	
usagePrefix	
caption	
visible	☑

Por último, resta definir las métricas o medidas del cubo. Se selecciona la opcion Add Measure

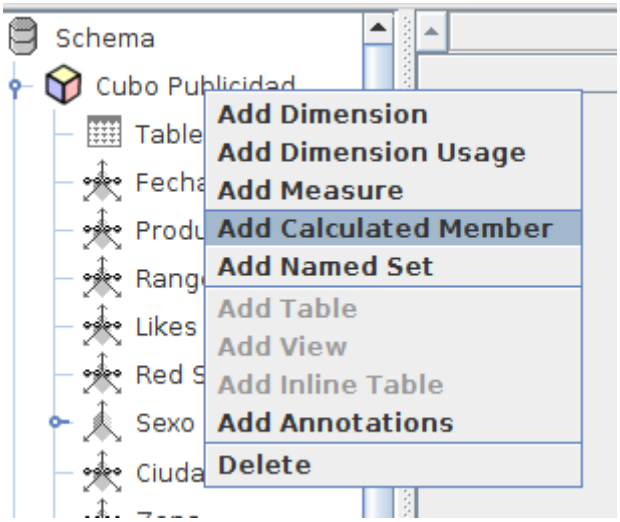


Luego, se define el nombre, el campo de la tabla de hechos correspondiente y qué tipo de cálculo se desea realizar sobre él.

Measure for 'Cubo Publicidad' Cube

Attribute	Value
name	Prints
description	
aggregator	sum
column	num_prints
formatString	
datatype	
formatter	
caption	
visible	☑

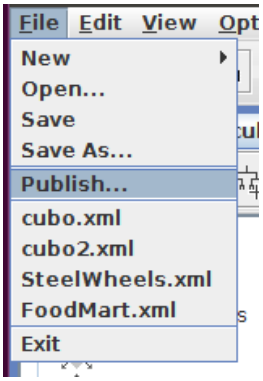
El indicador CTR se puede calcular a partir del número de prints y hits disponible. Por este motivo, se agrega un miembro calculado:



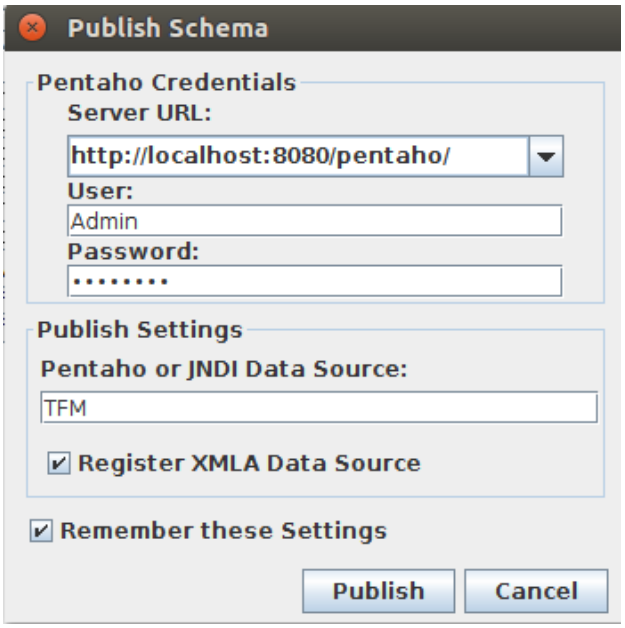
Calculated Member for 'Cubo Publicidad' Cube

Attribute	Value
name	CTR
description	
caption	
dimension	Measures
hierarchy	
parent	
visible	☑
formula | formulaElem...	[Measures].[Hits]/[Measures].[Prints]
formatString	

Una vez diseñado el cubo, se lo publica eligiendo la opción Publish del menú File:



Luego, se debe definir la conexión al servidor de Pentaho:


Publish Schema
Pentaho Credentials
Server URL:
http://localhost:8080/pentaho/
User:
Admin
Password:
Publish Settings
Pentaho or JNDI Data Source:
TFM
Register XMLA Data Source
Remember these Settings
Publish
Cancel