

PROYECTO IMPLEMENTACION DE **UN DATAMART DE UN CASO DE** **ESTUDIO**

Elaborado por:

- **Díaz Danny**
- **Jaramillo Ronny**
- **Román Bolívar**

Profesor: María Hallo

Semestre: Sexto Semestre, 2019-B

I. PLANEACIÓN Y ADMINISTRACIÓN DEL PROYECTO

I.1. Introducción, descripción de la organización

Se ha elegido a la empresa *Herrajes & Tiraderas*, dedicada a la comercialización de productos de ferretería al por mayor y menor. Ubicada en el sur de Quito, la organización se ha caracterizado por su alta competitividad en precios y la excelente calidad de sus productos.

La organización está catalogada como *mediana empresa* según el volumen de ventas que maneja y la capacidad de sus instalaciones. Se busca que la organización eleve los réditos de sus actividades con el análisis de datos y se pueda tener una visión más clara de los productos y procesos que se deben optimizar dentro de la misma.

De manera general y a partir de los directivos de la empresa, el principal objetivo es tener más información sobre las necesidades de los clientes, para de esta manera ajustar a la empresa a estos requerimientos y hacerla más competitiva.

I.2. Título del proyecto

Implementación de un DataWarehouse para la empresa *Herrajes & Tiraderas*, con la finalidad de optimizar los procesos internos de la empresa.

I.3. Calendario tentativo

ACTIVIDAD	FECHA DE INICIO	FECHA DE FIN
Planeamiento del Proyecto	29-10-2019	01-11-2019
Definición de Requerimientos de Negocio	01-11-2019	02-11-2019
Diseño de Arquitectura Técnica	03-11-2019	06-11-2019
Selección de Instalación y Producto	07-11-2019	07-11-2019
Modelamiento Dimensional	08-11-2019	10-11-2019
Aplicación Especificación End-User	11-11-2019	12-11-2019
Diseño Físico	13-11-2019	14-11-2019
Desarrollo del Diseño y Datos	15-11-2019	18-11-2019
Desarrollo Aplicación End - User	19-11-2019	25-11-2019
Despliegue	26-11-2019	26-11-2019

I.4. Recursos: personal, hardware, software, costos

La empresa cuenta con 4 trabajadores en total dispuestos de la siguiente manera:

- Gerente general
- Auxiliar de ventas
- Auxiliar de bodega
- Contador

Cada uno de ellos tiene a su disposición un computador personal en su área, que tiene las características:

- Procesador: Intel Core i5 3.3 GHz
- Memoria RAM: 8 GB
- Disco Duro: HDD 500 GB
- SO: Windows 10 Pro

I.5. Autorización del proyecto

El presente proyecto ha sido autorizado por parte de la instructora de la materia tras la presentación de un avance de este durante una sesión de clases. Dentro de este avance, se redactaron las tres primeras secciones del presente documento. Aquí se logró definir las bases sobre las cuales el presente proyecto se desarrolló por lo que la autorización del proyecto fue exitosa.

II. ANÁLISIS DE REQUERIMIENTOS

II.1. El negocio

II.1.1. Objetivos del negocio

- Aumentar el volumen de ventas de la organización con conocimiento de cuáles son los productos que más necesitan los clientes.
- Variar el stock de la empresa con los productos de más venta.
- Mantener el stock de productos de mayor rotación estimando cantidades de venta según los datos.

II.1.2. Los problemas del negocio

- Muchas de las veces la organización deja de vender porque pierde el stock de los productos de mayor rotación.
- La organización adquiere mercadería que no es de alta rotación, por lo tanto, tiene nuevas deudas que son difíciles de cubrir posteriormente.

II.1.3. Descripción de los procesos a modelar

El proceso de compra de productos a proveedores de la organización debe ser administrado de mejor manera, para ello se utilizarán los datos de facturación, con lo que se pretende obtener información sobre cuáles son los productos de mayor rotación y enfocar a la organización en venta de éstos.

Posteriormente, se pretende hacer el uso de encuestas para saber directamente desde los consumidores cuáles son los productos de mayor necesidad.

Esto mejorará las ventas de la empresa y además elevará la satisfacción del cliente posicionando a la organización en un lugar estratégico del mercado y del sector en el que se encuentra.

II.1.4. Priorización de procesos

Como se ha podido observar, la organización quiere mejorar su proceso de compra a proveedores y el proceso de venta hacia sus clientes, por lo que estos procesos se sitúan en el siguiente orden según la administración de la empresa:

1. Gestión de Comercialización
2. Gestión de Relación con Proveedores
3. Gestión de Pedidos
4. Gestión de Demanda
5. Gestión de Relación con Clientes
6. Gestión de Servicio al Cliente
7. Gestión de Devoluciones

II.1.5. Necesidades de información de alto nivel

La información a alto nivel resulta mucho más familiar para los individuos que trabajen en este proyecto por lo que es elemental que, los datos sobre los cuales el datamart se va a construir, sean de alto nivel. Esto incluye todo tipo de registros para cada una de las dimensiones a incluir en la arquitectura del mismo. Así, la construcción del datamart será:

- Más sencilla
- Más rápida
- Más comprensible

II.1.6. Matriz de procesos vs. Entidades del negocio

	GERENCIA	VENTAS	CONTABILIDAD	BODEGA
GESTIÓN DE RELACIÓN CON CLIENTES		X		
GESTIÓN DE SERVICIO AL CLIENTE		X		
GESTIÓN DE RELACIÓN CON PROVEEDORES				X
GESTIÓN DE DEMANDA	X	X		
GESTIÓN DE PEDIDOS	X			X
COMERCIALIZACIÓN	X	X	X	
GESTIÓN DE DEVOLUCIONES		X	X	X

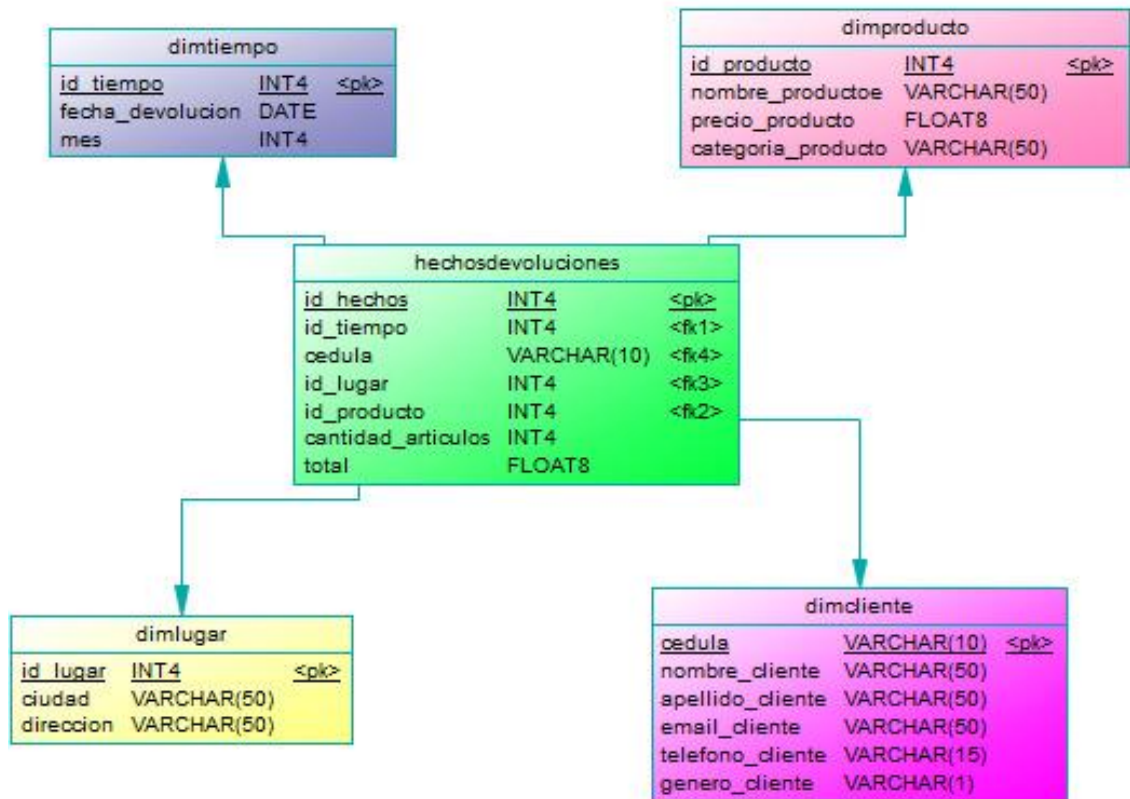
II.1.7. Definición de indicadores de rendimiento, dimensiones

- **Gestión de Relación Con Clientes**
 - Número de clientes fijos que realizan transacciones con la organización.
- **Gestión de Servicio Al Cliente**
 - Número de quejas presentadas por los clientes que han adquirido productos.
- **Gestión de Relación con Proveedores**
 - Rating establecido por parte de proveedores con respecto a las adquisiciones
- **Gestión de Demanda**
 - Cantidad de productos vendidos clasificados por categorías.
- **Cumplimiento de Pedidos**
 - Número de incumplimientos, retrasos en la entrega de pedidos.
- **Comercialización**
 - Cantidad de órdenes cumplidas satisfactoriamente.
- **Gestión de Devoluciones**
 - Número de productos defectuosos que han sido devueltos a la empresa.

II.1.8. Fuentes de datos

Se usará la información proveniente de facturas electrónicas y las físicas.

III. EL MODELO LÓGICO: MODELO DIMENSIONAL



Para facilitar el proceso de visualización gráfica de un DW o DM, las herramientas de análisis, que serán producto de un estudio posterior, presentan al usuario un modelo multidimensional en el cual se muestra una actividad que es objeto de análisis y las dimensiones cuantificables que facilitarán dicho proceso de análisis.

III.1. Selección del Proceso sobre el que se construirá el DataMart.

Se selecciona el proceso de “Devolución” para la devolución de artículos comprados por clientes por diversas razones, puede ser por fallos y defectos de fábrica, garantía, etc.

III.2. Definir la granularidad

Se considera para el siguiente problema una granularidad diaria, de esta forma se puede analizar la cantidad de días que permaneció con el cliente, no tendría sentido una granularidad temporal de un mes, debido a que un artículo comprado por un cliente puede ser devuelto en el mismo mes que fue comprado causando confusión con las entregas y devoluciones.

III.3. Definir las dimensiones

Se consideran las dimensiones de tiempo (granularidad), lugar (donde fue devuelto el producto), cliente (quien realizó la devolución) y producto (que es lo que se devolvió).

III.4. Definir las tablas de hecho

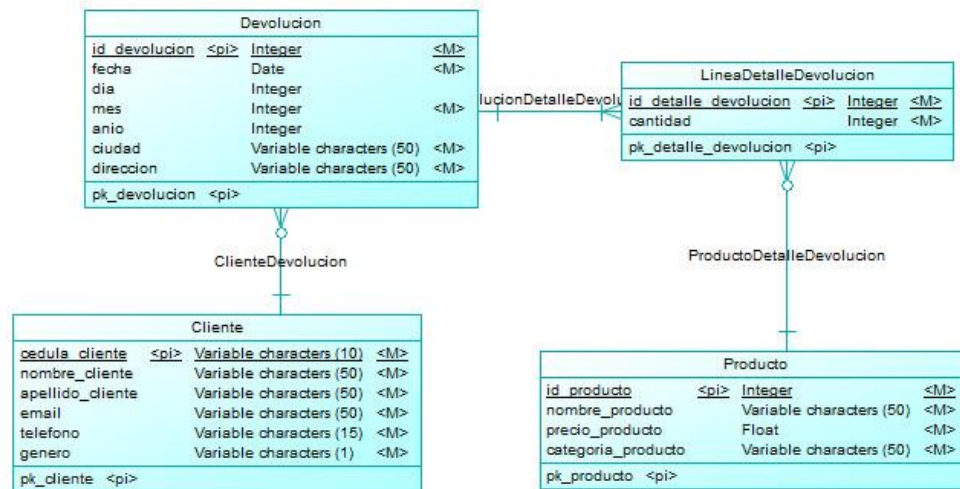
Hechos: Es la actividad que es objeto del análisis cuya información relevante se representa mediante un conjunto de indicadores que los llamaremos medidas o atributos del hecho; por ejemplo: “En una sucursal se recepta tantos artículos de un producto X en tal fecha de parte de un cliente”. El hecho o

actividad que será objeto de un análisis de las ventas en dólares, que se visualizará por almacén, por tiempo y por producto.

III.5. Modelo de Estrella

Para profundizar el conocimiento de la información se aplica un modelo copo de nieve, normalizando las tablas de productos y y

III.6. Diseño de base de datos intermedia



III.7. Diccionario de datos

Campo	Tamaño	Tipo de Dato	Descripción
fecha	10	Fecha	Fecha de registro de la devolución
dia	2	Numérico	Número de día de la fecha de devolución
mes	2	Numérico	Número de mes de la fecha de devolución
anio	4	Numérico	Número de año de la fecha de devolución
ciudad	50	Caracter	Ciudad donde se realiza el proceso de devolución
direccion	50	Caracter	Dirección hacia donde se realiza la devolución
nombre_cliente	50	Caracter	Nombre del cliente que requiere de la devolución
apellido_cliente	50	Caracter	Apellido del cliente que requiere de la devolución
email	50	Caracter	Dirección de correo electrónico del cliente
telefono	15	Catacter	Número de teléfono del cliente
genero	1	Caracter	Género del cliente
nombre_producto	50	Caracter	Nombre del producto del proceso de devolución
precio_producto	10	Flotante	Precio del producto del proceso de devolución
categoria_producto	50	Caracter	Categoría del producto del proceso de devolución

cantidad	10	Numérico	Cantidad numérica del producto
id_devolucion	-	Numérico	Clave de la devolución
cedula_cliente	10	Caracter	Número de cédula del cliente
id_detalle_devolucion	-	Numérico	Clave del detalle de la devolución
id_producto	-	Numérico	Clave del producto

IV. DISEÑO TÉCNICO DE LA ARQUITECTURA

IV.1. Infraestructura: servidores, equipos.

Para la implementación de DataMart, se cuentan con los equipos descritos anteriormente, en los cuales se ha instalado el software necesario para poder llevar a cabo la ejecución del modelo.

IV.2. Flujo técnico de la arquitectura: back room, front room.

Backroom: los datos fueron creados

Frontroom: los datos serán presentados

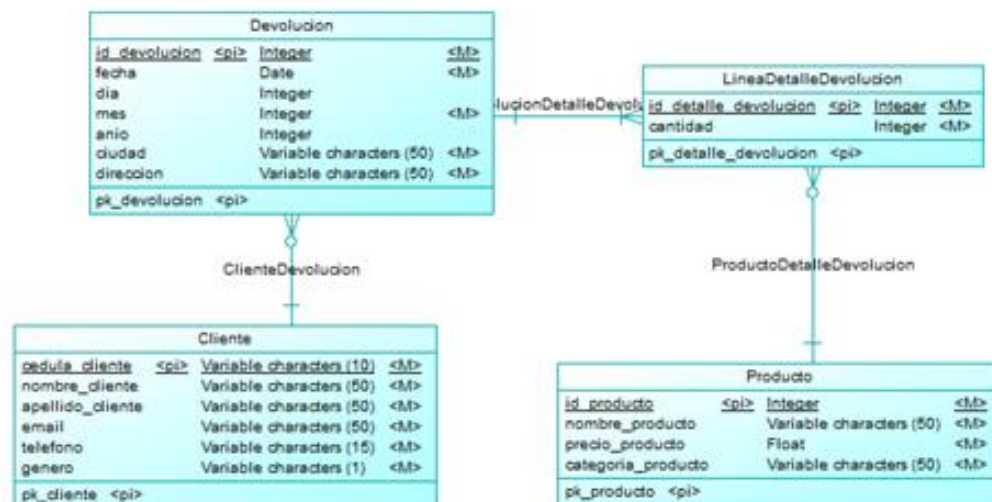
V. PROCESOS DE EXTRACCIÓN, TRANSFORMACIÓN Y CARGA

V.1. Identificar fuentes y destinos detallados

Fuentes

La fuente de datos principal será una base de datos de PostgreSQL.

Cuyo esquema conceptual es el siguiente:

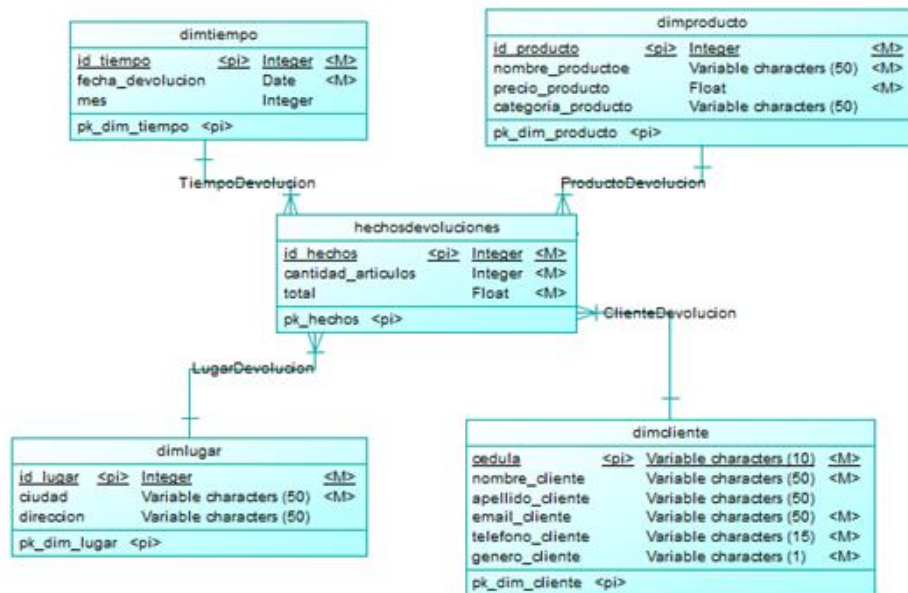


En esta fuente se almacena las devoluciones que realizan los clientes de uno o varios productos registrando la fecha, dirección y cantidad por producto devuelto.

Destinos

El destino será un Datamart compuesto de dimensiones y una tabla de hechos cuyo indicador principal es el número de artículos devueltos por cada producto y el dinero total que se perdió por esa devolución.

Su esquema conceptual es el siguiente:



V.2. Descripción de herramienta de ETL

La herramienta de Extracción-Transformación-Carga a utilizarse es PENTAHO. Pentaho es una plataforma opensource con componentes que incluyen tecnologías como Big Data o Internet de las Cosas (IoT). Algunas de estas herramientas o componentes más conocidos son Pentaho Business Analytics (Pentaho BA) para ejecutar recursos como los informes o cuadro de mandos, CTools para crear y gestionar Dashboards, o Pentaho Data Integration (PDI).

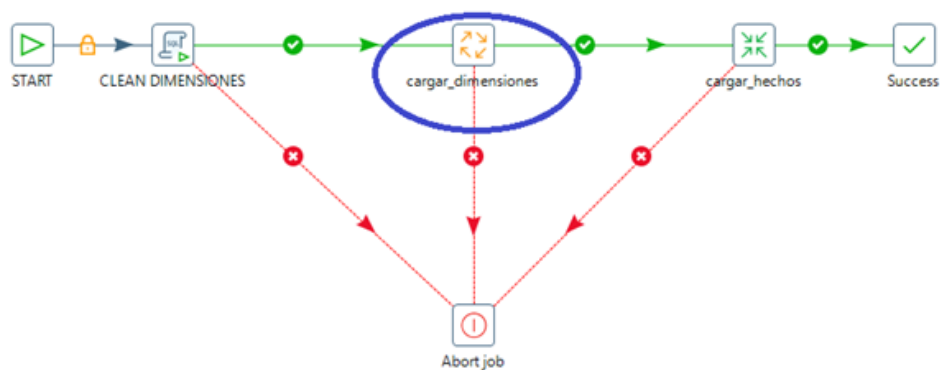


Pentaho Data Integration (PDI), cuyo nombre en clave es Kettle, es una de las herramientas o componentes de Pentaho Suite que permite que se utilicen técnicas ETL, es decir, poder implementar procesos de extracción, transformación y carga de datos. Kettle, además, ofrece datos analíticos muy precisos, eliminando las complejidades involucradas en la codificación al proporcionar bibliotecas en profundidad para el mismo.

Esta herramienta cuenta con módulos encargados de gestionar la entrada, salida, transformaciones, búsquedas, etc. Como se aprecia en la siguiente figura.

- > Entrada
- > Salida
- > Streaming
- > **Transformar**
- > Utility
- > Flow
- > Scripting
- > Pentaho Server
- > Búsqueda
- > Uniones
- > Almacén de Datos
- > Validation
- > Statistics
- > Big Data
- > Agile
- > Cryptography
- > Trabajo
- > Mapeado
- > Bulk loading
- > Embebido
- > Experimental
- > Obsoleto
- > Historio

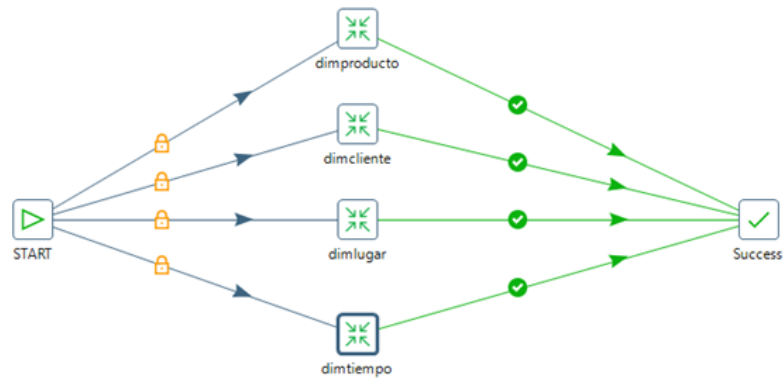
V.3. Cargar dimensiones



El proceso se detallará en el punto V.6 de esta sección

Primera dimensión: producto

Para la carga de esta dimensión:



1. Se extrae la tabla “producto” de la base de datos “Devoluciones” como entrada.
2. Se agrega un identificador único extra llamado ‘id’
3. Se seleccionan las columnas importantes para la dimensión (nombre, precio, categoría)
4. Se carga a la tabla “dimproducto” los datos transformados de la entrada.



Segunda dimensión: cliente

Para la carga de esta dimensión:

1. Se extrae la tabla “cliente” de la base de datos “Devoluciones” como entrada.
2. Se agrega un identificador único extra llamado ‘id’
3. Se seleccionan las columnas importantes para la dimensión (cedula, nombre, apellido, email, teléfono, genero)..
4. Se carga a la tabla “dimcliente” los datos transformados de la entrada.



Tercera dimensión: lugar

Para la carga de esta dimensión:

1. Se extrae la tabla “lugar” de la base de datos “Devoluciones” como entrada.
2. Se agrega un identificador único extra llamado ‘id’
3. Se seleccionan las columnas importantes para la dimensión (ciudad,dirección).
4. Se carga a la tabla “dimlugar” los datos transformados de la entrada.



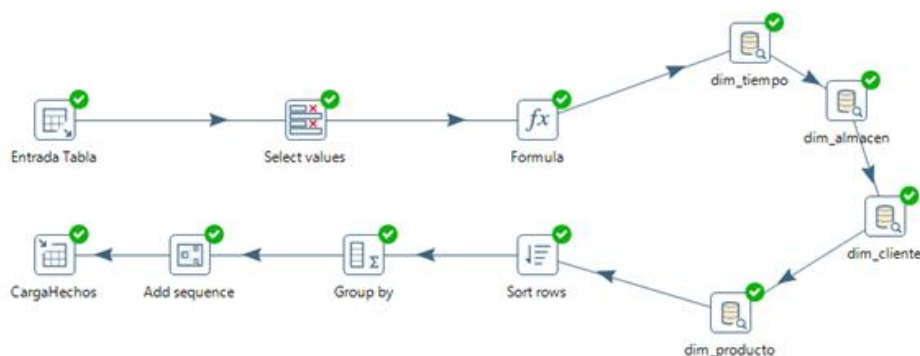
Cuarta dimensión: tiempo

Para la carga de esta dimensión:

1. Se extrae la tabla “cabecera devolucion” de la base de datos “Devoluciones” como entrada.
2. Se agrega un identificador único extra llamado ‘id’
3. Se seleccionan las columnas importantes para la dimensión (fecha, mes).
4. Se carga a la tabla “dimtiempo” los datos transformados de la entrada.



V.4. Cargar tabla de hechos



Para cargar los datos a la tabla de hechos se utiliza 11 pasos:

- Entrada, para la entrada se considera una consulta con varias juntas naturales sobre el documento de cabecera devolución, detalles devolución y tablas asociadas para

conseguir el id del detalle, id de la devolución, id del producto, la cédula del cliente, la fecha de la devolución, el mes (granularidad mensual), la cantidad de artículos devueltos y el precio de los productos.

- Luego se seleccionan los valores que se necesiten para cargar a la tabla de hechos.
- En el tercer paso se aplica una fórmula para crear una nueva columna la cuál será resultado de la multiplicación entre el precio de un producto y su cantidad, generando así el subtotal.
- En los siguientes 4 pasos se realizan junturas con las demás dimensiones para guardar congruencia en el almacenamiento de datos.
- Después se ordena la información recolectada con la siguiente jerarquía: tiempo, lugar, cliente y por último producto.
- Las agrupaciones se realizan con la misma jerarquía del punto anterior, pero se agrupa también por la cantidad, el precio y finalmente por el subtotal.
- A todo este proceso se le asigna un identificador único.
- Y al final cargamos todos los datos transformados en una tabla hechos del datamart.

V.5. Carga dimensiones y tabla de hechos incrementales.

Para una carga de dimensiones de forma incremental es importante utilizar la transformación 'Add sequence' de forma que desde el propio dominio del administrador se agreguen los identificadores únicos con una secuencia lineal.

Obtener valor de la secuencia de la base de datos

Nombre de paso

Add sequence

Nombre de valor

tbl_hecho_id

Utilizar una base de datos para generar la secuencia

Utilizar base datos para obtener secuencia?

☐

Conexión

Nombre de esquema

Nombre de secuencia

SEQ_

Utilizar un contador de la transformación para generar la secuencia

Utilizar contador para calcular secuencia?

☒

Nombre contador (opcional)

Valor inicial

1

Incremento

1

Valor máximo

99999999

Help

Vale

Cancelar

Utilizar una columna con valor auto incremental es también una excelente opción siempre y cuando la base de datos utilizada permita la desactivación del chequeo de 'identity', dado que para algunos transformaciones puede ser útil manipular esos identificadores únicos.

De todas formas, se aplica también una operación de valores únicos



Unique rows (HashSet)

Con el fin de salvaguardar los valores únicos y no sobreponer a la tabla. Para la carga continua se puede realizar de forma manual como se ha hecho en este trabajo, es decir realizar la consulta a la base de datos que actúa como fuente cada X tiempo. Se considera ese 'X' como un valor de 1mes a 6 meses dependiendo que tanta inserción de datos ocurra en las fuentes, para no sobre cargar al DataMart en una fecha determinada se recomienda cargar mensualmente y aprovechar esa granularidad.

Algo que no se hizo, pero queda para futuros trabajos es automatizar la carga utilizando 'scripts' escritos en PHP que se ejecuten a cierta hora del mes, para que empiecen a ordenar los datos, pero bloqueando la lectura de forma que se conserve la integridad de los datos.

V.6. Documentación de procesos

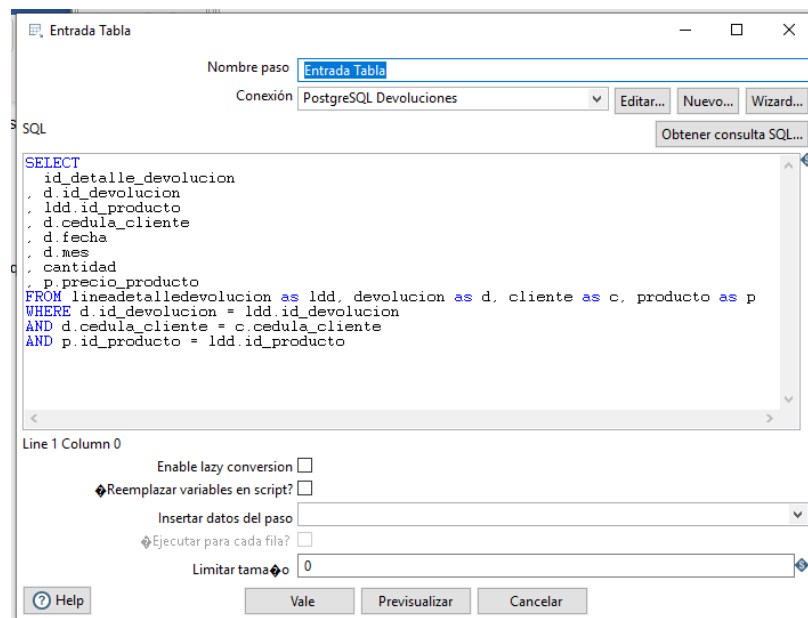
En los puntos V.3 y V.4 se explicó de forma general el procedimiento englobando lo más importante. En esta sección se profundizará en los procesos utilizados para cargar, transformar y cargar la información.



1. Entrada de datos

Para la entrada de datos se puede utilizar desde CSV, Excel, Ficheros, GZIP, información de tablas de cualquier base de datos hasta información extraída de Google Analytics.

Además, al trabajar con bases de datos SQL se puede utilizar consultas como requiera la extracción de datos que necesitemos, incluso realizar junturas por medio de código como se ve en la siguiente figura.



2. Seleccionar/Renombrar valores



Este proceso se utiliza para filtrar columnas de un conjunto de datos y para adecuar los nombres para estandarizar la información que llegará al DataMart. En la siguiente figura se aprecia como se filtra una gran cantidad de campos y se renombre 'precio_producto' a 'precio'.

#	Nombre campo	Renombrar a	Longitud	Precisión
1	id_detalle_devolucion			
2	id_producto			
3	cedula_cliente			
4	fecha			
5	mes			
6	cantidad			
7	precio_producto	precio		
8	id_devolucion			

3. Añadir secuencia de valores



Dado que al trabajar con diversas fuentes muchas pueden tener identificadores repetidos, es necesario crear una secuencia nueva para etiquetar a los datos que ingresan a las dimensiones y a los hechos

Por ejemplo, en la siguiente figura se aprecia una inserción de identificadores desde el número 1 hasta 999999999. La secuencia deberá ser adecuada a la cantidad de tuplas que se vaya a insertar.

Nombre de paso: Add sequence

Nombre de valor: tbl_hecho_id

Utilizar una base de datos para generar la secuencia

Utilizar base de datos para obtener secuencia? ☐

Conexión: [dropdown]

Nombre de esquema: [text]

Nombre de secuencia: SEQ_

Utilizar un contador de la transformación para generar la secuencia

Utilizar contador para calcular secuencia? ☒

Nombre contador (opcional): [text]

Valor inicial: 1

Incremento: 1

Valor máximo: 999999999

4. Salida de datos



En la salida de datos se apunta hacia el destino que en este caso será la tabla de hechos del Datamart.

Es importante recalcar como los nombres de los campos entre la entrada y la salida difieren a pesar de referirse al mismo 'tipo' de información.

Salida de Tabla

Nombre paso: CargaHechos

Conexión: PostgreSQL Datamart

Esquema destino:

Tabla destino: hechosdevoluciones

Tamaño de transacción: 1000

Vaciar tabla: ☐

Ignorar errores de inserción: ☐

Specify database fields: ☒

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	id_tiempo	tbl_tiempo_id
2	cedula	tbl_cliente_id
3	id_producto	tbl_producto...
4	cantidad_ar...	cantidad
5	total	sub_total
6	id	tbl_hecho_id
7	id_hechos	tbl_hecho_id
8	id_lugar	tbl_lugar_id

Get fields

Enter field mapping

Help Vale Cancelar SQL

5. Fórmula



La fórmula sirve para incluir una nueva columna en base a los datos que se reciba para mejorar el análisis de datos.

Para este trabajo se utiliza la fórmula de cantidad multiplicado por precio de producto para generar una nueva columna con el subtotal de la devolución para en un futuro averiguar cuánto dinero le ha costado a la empresa esa devolución.

Formula

Nombre paso: Formula

Fields:

#	New field	Formula	Value type	Length	Precision	R
1	sub_total	[cantidad]*[precio]	Number		2	

Help Vale Cancelar

6. Búsqueda de valor en la base de datos



Este proceso ayuda a comparar identificadores únicos entre las dimensiones y la nueva consulta realizada a la fuente de datos de forma que la integridad y congruencia de datos se mantenga en la relación de hechos.

La siguiente figura ilustra el procedimiento de comparación y la reasignación de nombre para el identificador que se incluirá en la tabla de hechos.

Búsqueda de valor en base de datos

Nombre paso:

Conexión:

Esquema de búsqueda:

Tabla de búsqueda:

Habilitar cache? ☐

Tamaño de cache en filas (0=todas):

Load all data from table ☐

La clave(s) para realizar búsqueda de valor(es):

#	Campo de tabla	Comparador	Campo1	Campo2
1	id_tiempo	=	id_devolucion	

Valores a devolver de la tabla de búsqueda:

#	Campo	Nuevo nombre	Defecto	Tipo
1	id	tbl_tiempo_id		BigNumber

No procesar la fila si la búsqueda falla ☒

Producir error si se obtienen múltiples ☐

Ordenar por:

7. Ordenar filas



Como su nombre lo indica, permite realizar un ordenamiento de forma ascendente para los identificadores de forma que la información se vaya organizando en el tiempo.

En el trabajo se ordena con la siguiente jerarquía: tiempo, lugar, cliente y finalmente producto.

8. Agrupación



Esta operación se debe utilizar siempre que se realice agregaciones sobre un mismo campo de forma que la información no se replique innecesariamente, tiene una semejanza con el group by del lenguaje de consultas SQL.

Agrupar

Nombre de paso

☒ Incluir todas las

Directorio temporal

Prefijo para ficheros

Añadir número de

Nombre de campo

☐ Always give back a

Campos que forman la agrupación:

#	Campo agrupación
1	tbl_tiempo_id
2	tbl_lugar_id
3	tbl_cliente_id
4	tbl_producto_id
5	cantidad

VI. DISEÑO FISICO

VI.1. Determinar la longitud de la BD del DWH

La longitud para la base de datos del Datawarehouse se compone de la longitud de las dimensiones y de la tabla de hechos.

Para la dimensión de tiempo:

Id_tiempo -> Integer -> 4 bytes
 Fecha_devolución -> Date -> 8 bytes
 Mes -> Integer -> 4 bytes
 Fecha_compra -> Date -> 8 bytes

Para la dimensión de producto:

id_producto -> Integer -> 4 bytes
 Nombre_producto -> varchar(50)-> 50 bytes
 precio_producto -> float-> 4 bytes
 categoria_producto-> varchar(50) -> 50 bytes

Para la dimensión de almacén (lugar):

id_lugar -> Integer -> 4 bytes
 Ciudad -> varchar(50)-> 50 bytes
 dirección -> varchar(50) -> 50 bytes

Para la dimensión de cliente:

Cedula -> varchar(10)-> 10 bytes
 Nombre_cliente -> varchar(50)-> 50 bytes
 apellido_cliente -> varchar(50)-> 50 bytes

Email_cliente -> varchar(50) -> 50 bytes
 telefono_cliente -> varchar(15) -> 15 bytes
 genero_cliente -> varchar(1) -> 1 bytes

Finalmente, para Hechos de devoluciones:

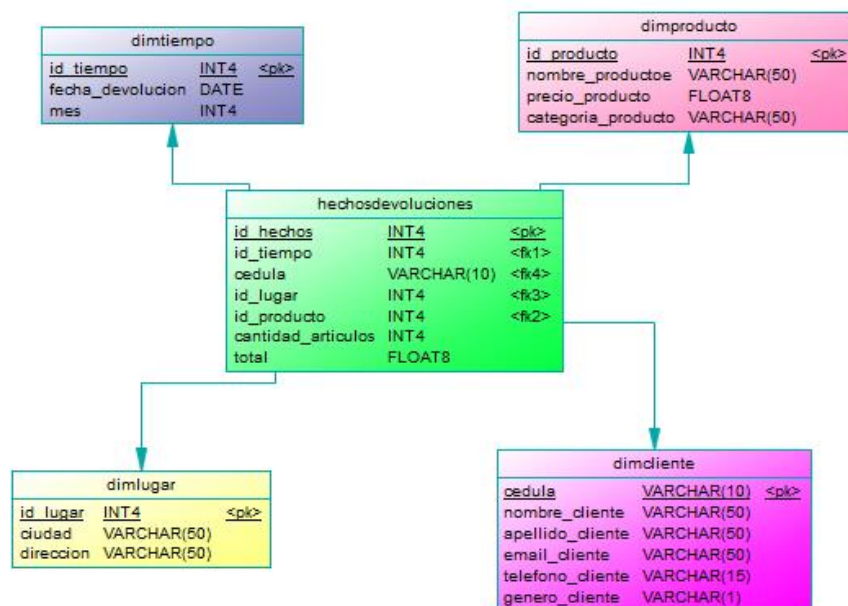
Identificadores de las dimensiones -> Integer * 4 -> 16 bytes
 Id_hechos -> Integer -> 4 bytes
 Cantidad_articulos -> Integer -> 4 bytes
 Total_dinero -> Float -> 4 bytes
 Dias_entre_compra_devolución -> Float -> 4 bytes

En total la longitud para una inserción con todas sus relaciones es de 444 bytes.

De forma que para insertar un millón de datos se requerirá 0,4135 GB aproximadamente. Para una base de datos estándar hosteada de 25GB con precio de 10\$ al mes, se podrán almacenar mas de 60 millones de registros hasta requerir una nueva expansión.

VI.2. Trasladar el modelo dimensional al modelo físico. Aplicando nombres y estándares de la base de datos, identificar el tipo de datos y la longitud de las columnas, etiquetar atributos como nulos y no nulos.

El siguiente modelo corresponde a uno a exportarse para PostgreSQL, con sus tipos de datos primitivos.



Al pasar del modelo conceptual al físico todas las relaciones de 1:N trasladan su llave primaria hacia la tabla de 'N' transformándolas en llaves foráneas.

Para este trabajo se consideró la cantidad de artículos y el total de dinero perdido por devolución en un tiempo. Indicadores de crecimiento como capacitación a los empleados para manejar el tema de las devoluciones se omite de forma que la tabla de hechos tenga información suficiente para que permita un análisis profundo de datos con alta escalabilidad.

En el diagrama del modelo conceptual se ha establecido los que son mandatorios '<M>' que en este caso son todos, al transformarlo al modelo de PostgreSQL estas características se han puesto ocultas.

VI.3. Determinar la estrategia de indexación

Para Data Warehouse hay varias opciones de indexación, entre las que se incluyen índices de almacén de columnas, índices agrupados e índices no agrupados y una opción que no es de índice también conocida como "montón".

La estrategia de 'Tablas del montón' se descarta dado que no se usará tablas de apilamiento ni se trabaja con datos temporales.

La mejor estrategia de indexación para no expertos siempre será el de índices de almacén de columnas en clúster. Aunque los índices clúster suelen ser mejores en cuanto se requiera recuperar rápidamente una sola fila, esto significa que esa estrategia beneficia solo a los que utilicen un filtro muy selectivo en la columna del índice agrupado

Las tablas del almacén de columnas en clúster ofrecen el máximo nivel de compresión de datos, así como el mejor rendimiento general de las consultas. Por lo general, las tablas del almacén de columnas en clúster funcionan mejor que las tablas de montón o de índice agrupado, y suelen ser la mejor opción para tablas grandes. Por estos motivos, el almacén de columnas agrupado es el mejor lugar por el que empezar cuando no se sabe cómo indexar una tabla.

Al crear la tabla será importante agregar esta instrucción:

WITH (CLUSTERED COLUMNSTORE INDEX);

Adecuado para la base de datos de destino, en este caso PostgreSQL.

Las tablas de almacén de columnas en clúster organizan los datos en segmentos. El hecho de que los segmentos sean de gran calidad es fundamental para conseguir un rendimiento óptimo de las consultas en una tabla de almacén de columnas. La calidad de los segmentos se puede medir por el número de filas de un grupo de filas comprimido. La calidad de los segmentos es mayor donde hay al menos 100 000 filas por grupo de filas comprimido y su rendimiento aumenta a medida que el número de filas por grupo se aproxima a 1.048.576, que es el número máximo de filas que puede contener un grupo.

VI.4. Sentencias SQL para crear el modelo

PDM Generation Options

General Detail Target Models Selection

☒ Generate new Physical Data Model

DBMS: PostgreSQL 9.x

Name: ConceptualDatamart

Code: CONCEPTUALDATAMART

Configure Model Options...

☐ Update existing Physical Data Model

Select model: <None>

DBMS:

☒ Preserve modifications

Aceptar Cancelar Aplicar Ayuda

```

/*=====*/
/* TABLE: DIMCLIENTE */
/*=====*/

CREATE TABLE DIMCLIENTE (
CEDULA          VARCHAR(10)          NOT NULL,
NOMBRE_CLIENTE  VARCHAR(50)          NOT NULL,
APELLIDO_CLIENTE VARCHAR(50)          NULL,
EMAIL_CLIENTE   VARCHAR(50)          NOT NULL,
TELEFONO_CLIENTE VARCHAR(15)         NOT NULL,
GENERO_CLIENTE  VARCHAR(1)           NOT NULL,
CONSTRAINT PK_DIMCLIENTE PRIMARY KEY (CEDULA)
);

```

```

/*=====*/
/* INDEX: DIMCLIENTE_PK */
/*=====*/

CREATE UNIQUE INDEX DIMCLIENTE_PK ON DIMCLIENTE (
CEDULA
);

```

```

/*=====*/
/* TABLE: DIMLUGAR */
/*=====*/

CREATE TABLE DIMLUGAR (
ID_LUGAR          INT4          NOT NULL,
CIUDAD            VARCHAR(50)    NOT NULL,
DIRECCION         VARCHAR(50)    NULL,
CONSTRAINT PK_DIMLUGAR PRIMARY KEY (ID_LUGAR)
);

```

```

/*=====*/
/* INDEX: DIMLUGAR_PK */
/*=====*/

CREATE UNIQUE INDEX DIMLUGAR_PK ON DIMLUGAR (
ID_LUGAR
);

```

```

/*=====*/
/* TABLE: DIMPRODUCTO */
/*=====*/

CREATE TABLE DIMPRODUCTO (
ID_PRODUCTO       INT4          NOT NULL,
NOMBRE_PRODUCTOE  VARCHAR(50)   NOT NULL,
PRECIO_PRODUCTO   FLOAT8        NOT NULL,
CATEGORIA_PRODUCTO VARCHAR(50)   NULL,
CONSTRAINT PK_DIMPRODUCTO PRIMARY KEY (ID_PRODUCTO)
);

```

```

/*=====*/
/* INDEX: DIMPRODUCTO_PK */
/*=====*/

CREATE UNIQUE INDEX DIMPRODUCTO_PK ON DIMPRODUCTO (
ID_PRODUCTO
);

```

```

/*=====*/
/* TABLE: DIMTIEMPO */
/*=====*/

CREATE TABLE DIMTIEMPO (
ID_TIEMPO         INT4          NOT NULL,
FECHA_DEVOLUCION  DATE          NOT NULL,
MES               INT4          NULL,

```

```
CONSTRAINT PK_DIMTIEMPO PRIMARY KEY (ID_TIEMPO)
);
```

```
/*=====*/
/* INDEX: DIMTIEMPO_PK */
/*=====*/
CREATE UNIQUE INDEX DIMTIEMPO_PK ON DIMTIEMPO (
ID_TIEMPO
);
```

```
/*=====*/
/* TABLE: HECHOSDEVOLUCIONES */
/*=====*/
CREATE TABLE HECHOSDEVOLUCIONES (
ID_HECHOS          INT4          NOT NULL,
ID_TIEMPO          INT4          NOT NULL,
CEDULA             VARCHAR(10)   NOT NULL,
ID_LUGAR           INT4          NOT NULL,
ID_PRODUCTO        INT4          NOT NULL,
CANTIDAD_ARTICULOS INT4          NOT NULL,
TOTAL              FLOAT8        NOT NULL,
CONSTRAINT PK_HECHOSDEVOLUCIONES PRIMARY KEY (ID_HECHOS)
);
```

```
/*=====*/
/* INDEX: HECHOSDEVOLUCIONES_PK */
/*=====*/
CREATE UNIQUE INDEX HECHOSDEVOLUCIONES_PK ON HECHOSDEVOLUCIONES (
ID_HECHOS
);
```

```
/*=====*/
/* INDEX: TIEMPODEVOLUCION_FK */
/*=====*/
CREATE INDEX TIEMPODEVOLUCION_FK ON HECHOSDEVOLUCIONES (
ID_TIEMPO
);
```

```
/*=====*/
/* INDEX: PRODUCTODEVOLUCION_FK */
/*=====*/
CREATE INDEX PRODUCTODEVOLUCION_FK ON HECHOSDEVOLUCIONES (
ID_PRODUCTO
);
```

```
);
```

```
/*=====*/  
/* INDEX: LUGARDEVOLUCION_FK */  
/*=====*/  
CREATE INDEX LUGARDEVOLUCION_FK ON HECHOSDEVOLUCIONES (  
ID_LUGAR  
);
```

```
/*=====*/  
/* INDEX: CLIENTEDEVOLUCION_FK */  
/*=====*/  
CREATE INDEX CLIENTEDEVOLUCION_FK ON HECHOSDEVOLUCIONES (  
CEDULA  
);
```

```
ALTER TABLE HECHOSDEVOLUCIONES  
ADD CONSTRAINT FK_HECHOSDE_CLIENTEDE_DIMCLIEN FOREIGN KEY (CEDULA)  
REFERENCES DIMCLIENTE (CEDULA)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE HECHOSDEVOLUCIONES  
ADD CONSTRAINT FK_HECHOSDE_LUGARDEVO_DIMLUGAR FOREIGN KEY (ID_LUGAR)  
REFERENCES DIMLUGAR (ID_LUGAR)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE HECHOSDEVOLUCIONES  
ADD CONSTRAINT FK_HECHOSDE_PRODUCTOD_DIMPRODU FOREIGN KEY (ID_PRODUCTO)  
REFERENCES DIMPRODUCTO (ID_PRODUCTO)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE HECHOSDEVOLUCIONES  
ADD CONSTRAINT FK_HECHOSDE_TIEMPODEV_DIMTIEMP FOREIGN KEY (ID_TIEMPO)  
REFERENCES DIMTIEMPO (ID_TIEMPO)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

VII. APLICACIONES PARA USUARIOS FINALES: CARACTERÍSTICAS

VII.1. Descripción de la herramienta para el procesamiento analítico.

Para realizar el proceso de analítico de datos se ha usado la herramienta Qlik Sense, es una solución de desarrollo de visualización de datos de autoservicio de vanguardia que permite a las personas de negocios desarrollar fácilmente aplicaciones de descubrimiento de negocios flexibles e intuitivas basadas en el propio conjunto de ideas y formas de ver la información de los datos.

Permite a los usuarios crear rápidamente impresionantes imágenes, gráficos, cuadros; explore datos y revele sus conexiones ocultas y descubra información comercial desde cualquier ángulo.

Características de Qlik Sense:

- Revelar datos ocultos con visualización interactiva de arrastrar y soltar
- Descubrir relaciones de datos con Smart Search
- Conectarse a múltiples fuentes de datos
- Accesible desde cualquier lugar, en cualquier momento y desde cualquier dispositivo
- Colaborar con Rich Data Storytelling
- Bibliotecas de datos consistentes y gobernadas
- Diseñado arquitectónicamente para implementación empresarial

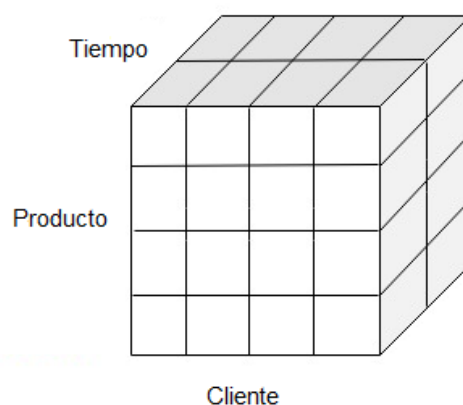
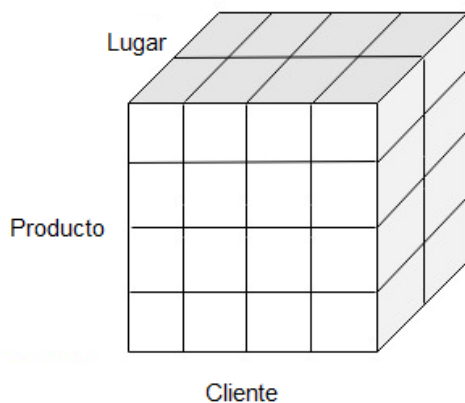
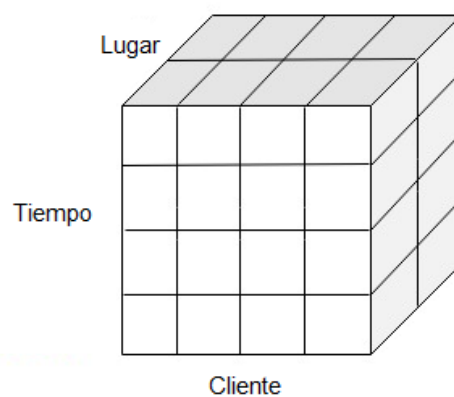
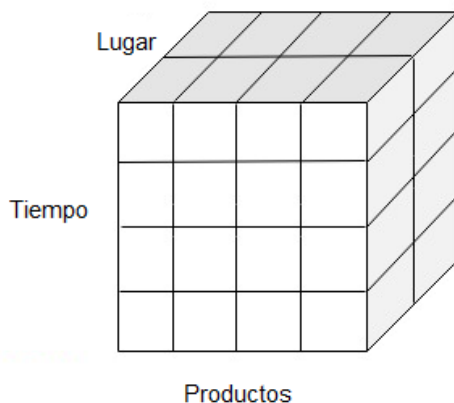
VII.2. Listar los resúmenes de información requeridos por los usuarios.

Los usuarios del DataMart requieren cuadros de información o gráficos estadísticos en donde se muestre la siguiente información:

- Número de ventas por producto
- Número de ventas por categoría
- Número de ventas por lugar
- Número de ventas por mes.

VII.3. Determinar los cubos.

Para la creación de los cubos, debemos tomar en cuenta que existen cuatro dimensiones que resultan en los siguientes cubos:



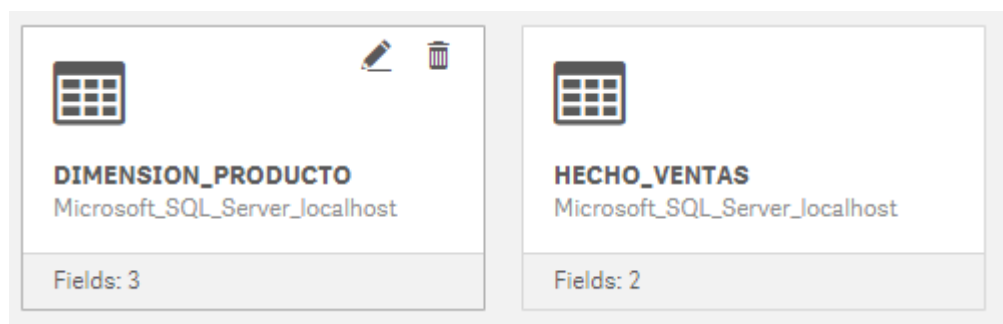
VII.4. Diseñar los reportes a mostrar.

Al incorporarse toda la información dentro de un dashboard, no es necesario el diseño de un reporte, ya que todos los gráficos se muestran en una sola interfaz, es importante escoger adecuadamente el tipo de gráfico que se quiere mostrar, en este caso para los gráficos en donde se visualizan totales se usarán diagramas de pastel en donde se muestren porcentajes de totales, mientras que en los casos en los que se quieran observar tendencias por productos, será mejor utilizar gráficos de barras, de esta manera se clarificarán las tendencias.

Todo esto se hace con el objetivo de mostrar la información gerencial más importante a los usuarios del DataMart.

VII.5. Implementación

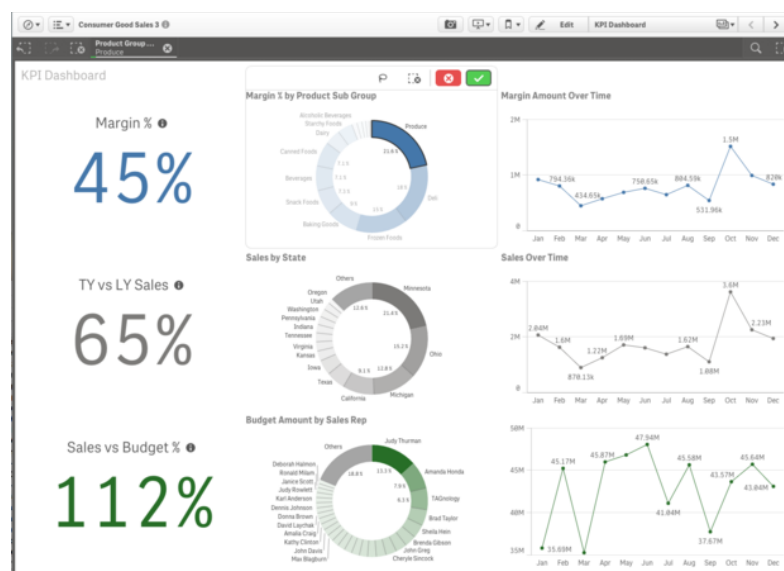
VII.5.1. Crear las tablas dinámicas.



Aquí se seleccionan las tablas necesarias y se establecen las relaciones entre las diferentes entidades, de esta manera se relacionarán los campos y se podrá visualizar la información en un gráfico.

VII.5.2. Construir los gráficos dinámicos.

Para la construcción de gráficos únicamente se debe escoger el tipo de gráfico que se quiere observar y luego establecer las relaciones y los criterios para los parámetros de cada dato, luego se mostrarán los gráficos en un panel de la siguiente manera:



VII. REFERENCIAS


[1] Hitachi, "Pentaho - Data Integration and Analytics Platform", *Hitachivantara.com*, 2019. [Online]. Available: <https://www.hitachivantara.com/en-us/products/data-management-analytics/pentaho-platform.html>. [Accessed: 24- Nov- 2019].

[2] Qlik, "Data Analytics for Modern Business Intelligence", *Qlik.com*, 2019. [Online]. Available: <https://www.qlik.com/us>. [Accessed: 24- Nov- 2019].

[3] PostgreSQL, "PostgreSQL: About", *Postgresql.org*, 2019. [Online]. Available: <https://www.postgresql.org/about/>. [Accessed: 24- Nov- 2019].

VIII. ANEXOS

VIII.1. Factura perteneciente a la organización



IMPORTADORES DIRECTOS
Cerraduras - Bisagras - Garruchas -
Rieles y todo para acabado de muebles
DOCUMENTO CATEGORIZADO: NO

JORDY ALEXIS JÁCOME GAROFALO
DOCUMENTO CATEGORIZADO: NO
WWW.HERRAJESYTIRADERAS.COM
Dirección: Av. Mariscal Sucre 3715 y Carapungo, (frente a Edimca) Sector La Raya. Telf.: (02) 284 - 2391 Cel.: (+593) 099 920 9241
E-mail: info@herrajesytiraderas.com
j.jacome@herrajesytiraderas.com
Quito - Ecuador

FACTURA
RUC: 1206522490001
Aut. SRI. 1120583715
Fecha de Autorización: 18 ABRIL 2017
001-001-Nº **0017851**

Fecha: Quito: 28 Febrero 2018 Vendedor: [Firma]
Cliente: Silverio Corredor
Dirección: Cumbayá Calle del Pepe lote 29 y del Establo

RUC/CI.: 1706794003001
Telf.: 022 8958896
Forma de pago:

Cantidad	CODIGO	DETALLE	Desc %	V.Unit.	Valor Total
300		tornillos negros 2x6		1.25	3.75
300		" " 2 1/2 x 8		2.10	6.30
300		" " 1 x 8		0.90	2.70
56		pares bisagra recta 4cv.		1.00	56.00
2		tupas blancas.		1.00	2.00
14		soportes ovalados.		0.25	3.50
7		tubos ovalados 3 metros.		5.50	38.50
31		trud. tubulares 128 g. Sn.		1.10	34.10
300		tuos. PG.		1.15	3.45

3

Debo y Pagare incondicionalmente y sin protesto a la orden de **JORDY JÁCOME**. En la ciudad de Quito, el valor de esta factura en las condiciones en ella establecidas. En caso de mora pagare intereses al máximo legal vigente en el sistema financiero y bancario sin protesto. La mercadería antes mencionada he recibido conforme y en perfectas condiciones. Declaro no tener ningún reclamo que hacer por el estado ni por la cantidad de las mismas. Las mercaderías de la presente Factura son propiedad de **JORDY JÁCOME** quién se reserva el dominio de las mismas hasta su entera cancelación.

FAVOR PAGAR CON CHEQUE CRUZADO A LA ORDEN DE JORDY JÁCOME

ELABORADO

CIUDAD

[Firma]

FIRMA AUTORIZADA

FIRMA CONFORME

JOSE DAVID SINGAÑA RIVERA - R.U.C. 1716123805001 / TELF.: 0995 497149 / AUT. 4229 DEL 0015201 AL 0018200

SUB-TOTAL 134.20

DESCUENTO

SUB-TOTAL NETO 134.20

12 %IVA 16.10

TOTAL 150.30

CADUCA 18 ABRIL 2018