



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

IMPLEMENTACIÓN DE UNA SOLUCIÓN DE INTELIGENCIA DE NEGOCIOS (BI) PARA
EL MÓDULO DE VENTAS DE CLARO UTILIZANDO LA HERRAMIENTA PENTAHO

Trabajo de Titulación presentado en conformidad con los requisitos establecidos
para optar por el título de Ingeniero en Sistemas de Computación e Informática

Profesor Guía

Ing. Sandra Johana Álvarez

Autor

John Carlos Tacco Meléndez

Año

2015

DECLARACIÓN DEL PROFESOR GUÍA

Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos para un adecuado desarrollo del tema escogido, y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

.....

Ing. Sandra Johana Álvarez
Ingeniera en Sistemas Informáticos y de Computación
C.I. 1720689445

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Yo, John Carlos Tacco Meléndez, declaro que el trabajo presentado en este proyecto de tesis es original y de mi autoría. Todas las fuentes en este documento han sido respectivamente citadas.

.....

John Carlos Tacco Meléndez

C.I. 1708444219

AGRADECIMIENTOS

Agradezco primero que todo a Dios, por bendecirme cada día de mi vida. Agradezco a mis padres Víctor y María Elena por su apoyo incondicional por inculcarme cada día valores y sentimientos como el respeto, honradez, amor, sinceridad, trabajo duro, cariño, humildad, en conclusión a ser una persona de bien. Agradezco a mis hermanos María Cristina y Alex por todo el apoyo brindado y por siempre ser ejemplos a seguir.

Quiero también agradecer a mis compañeros de trabajo, que me ayudaron con la información necesaria, entrevistas y demás temas en relación a mi tema de tesis. Un agradecimiento especial a Juan Manuel Ossa y Martha Ponce que gestionaron la respectiva autorización en Claro para el auspicio de mi proyecto de tesis y por la confianza depositada en mi persona.

DEDICATORIA

Dedico este proyecto de tesis a Dios, a mis padres Víctor y María Elena, porque gracias a sus enseñanzas, amor, esfuerzo he podido culminar con este proyecto. También dedico este trabajo a mis hermanos María Cristina y Alex, porque gracias a su apoyo y ejemplo me ha dado fuerzas para finalizar mi carrera en la Universidad.

Dedico adicionalmente a mis profesores que gracias a sus enseñanzas académicas he podido ponerlas en práctica en mi vida profesional y a esforzarme cada día más.

RESUMEN

Hoy en día, un componente importante en la toma de decisiones, es el manejo eficaz y eficiente de la información. En el mercado existen herramientas que ayudan a la implementación de soluciones de inteligencia de negocios, pero pocas empresas las utilizan. Una herramienta de inteligencia de negocios se puede implementar en toda empresa o lugar de trabajo en donde se requiera centralizar la información para su posterior análisis, el usuario puede acceder a los reportes desde la intranet en cualquier parte de la empresa. El actual proyecto de tesis permite la implementación de una solución de inteligencia de negocios para el módulo de ventas de Claro utilizando la herramienta *Pentaho*, basándose en el enfoque de uno de los padres del *Data Warehousing* Ralph Kimball. La herramienta que se va a utilizar para este proyecto es *Pentaho Community Edition suite* de código libre, esta plataforma está compuesta por varios componentes que dan soporte para soluciones de inteligencia de negocios.

Uno de los recursos más importantes de una empresa es la información, por tal motivo es necesario el correcto manejo de la misma. Las áreas gerenciales y directivas de una empresa necesitan de sistemas de apoyo que les permitan realizar análisis de ciertos indicadores que muestren cómo está el negocio y así poder tomar decisiones que ayuden a mejorar el desempeño del mismo. La aplicación de inteligencia de negocios y sus herramientas, apoyan al nivel gerencial en la toma de decisiones generando información importante, a la fecha e histórica, la cual puede ser mostrada a través de cubos *OLAP*, reportes, o tableros de mando integral.

El uso de inteligencia de negocios a largo plazo en una empresa tiene grandes ventajas, debido a que se pueden obtener:

- ✓ Generación de pronósticos
- ✓ Proyecciones
- ✓ Tendencias

ABSTRACT

Nowadays, an important decision-making component is an effective and efficient information management. There are tools that help to implement business intelligence solutions in the market, but few companies use them. A BI tool can be implemented in any company or workplace where the information is centralized for further analysis, the user can access reports from anywhere in the intranet of the company. The current thesis project will allow the implementation of a sales BI solution for Claro using Pentaho tool, based on the approach of one of the parents of Data Warehousing Ralph Kimball. The tool that will be used for this project is Pentaho Community Edition suite, which is an open source tool. This platform consists of several components that support BI solutions.

One of the most important resources of a company is the information, for that reason the correct management of that is necessary. The management and the director's area of a company need systems that support and enable them to perform analysis of certain indicators about the business. In this way they can make decisions that help improve the performance of the company. The BI application and its tools support the management level decision making generating important and historical information that can be displayed via OLAP cubes, reports, or dashboards.

Using BI in a Company have great advantages at long-term, because they can obtain:

- ✓ Forecasting
- ✓ Projections
- ✓ Trends

ÍNDICE

INTRODUCCIÓN	1
Capítulo I	1
1 Introducción a las tecnologías de inteligencia de negocios	2
1.1 Historia sobre inteligencia de negocios	2
1.2 Significado de inteligencia de negocios	2
1.2.1 Reportes	3
1.2.2 Procesamiento analítico en línea	4
1.2.3 Minería de datos	5
1.3 Arquitectura de la inteligencia de negocios	5
1.3.1 La Fábrica de Información Corporativa	5
1.3.2 Arquitectura Multidimensional	9
1.3.3 Comparativas sobre las arquitecturas: Fábrica de Información Corporativa y Multidimensional	11
1.4 Aplicaciones de inteligencia de negocios	13
1.5 Preferencias de la inteligencia de negocios dentro de las empresas	15
1.5.1 Centro de competencia de inteligencia de negocios (BICC)	15
1.6 Data Warehousing	16
1.6.1 Historia	16
1.6.2 Concepto del almacén de datos	17
1.6.2.1 Recuperación de datos	18
1.6.2.2 Consolidación de datos	19
1.6.2.3 Periodicidad	20
1.6.2.4 Reseña Histórica	20
1.6.2.5 Consultas	20
1.6.3 Concepto de un <i>data mart</i>	20

1.6.4	Diseño de un almacén de datos	22
1.6.4.1	La arquitectura de flujo de datos	22
1.6.4.1.1	Arquitectura de flujo de datos de un almacén de datos	24
1.6.4.1.2	Arquitectura <i>DDS</i> simple	26
1.6.4.1.3	Arquitectura NDS + DDS	27
1.6.4.1.4	Arquitectura ODS + DDS	27
1.6.4.1.5	Arquitectura de un almacén de datos federado (FDW)	28
1.6.4.2	Arquitectura del sistema	29
1.7	Metodología de desarrollo de un almacén de datos	30
1.7.1	Metodología en cascada	31
1.7.1.1	Estudio de factibilidad	31
1.7.1.2	Requerimientos	32
1.7.1.3	Arquitectura	32
1.7.1.4	Configuración de la infraestructura	32
1.7.1.5	Diseño	32
1.7.1.6	Desarrollo	33
1.7.1.7	Pruebas	33
1.7.1.8	Despliegue	33
1.7.1.9	Operación	33
1.7.1.10	Administración del proyecto	34
1.7.2	Metodología iterativa	34
1.8	La plataforma de inteligencia de negocios de Pentaho	35
1.8.1	Definición de <i>Pentaho</i>	35
1.8.2	Componentes de <i>Pentaho</i>	35
1.8.2.1	La plataforma	37
1.8.2.2	Componentes de inteligencia de negocios	38
1.8.2.3	La capa de presentación	39
1.8.3	Principales herramientas de <i>Pentaho</i>	40

1.8.4 Principales características de <i>Pentaho</i>	43
Capítulo 2	44
2 Diseño e implementación del data mart del módulo de ventas de Claro por medio de la herramienta Pentaho.....	44
2.1 Situación actual	44
2.1.1 Definición de cada objetivo	45
2.1.2 Delimitación del proyecto.....	45
2.1.3 Roles establecidos.....	46
2.1.4 Uso de metodología.....	46
2.1.5 Planificación.....	46
2.1.6 Recursos a utilizar: Hardware y software.....	47
2.1.6.1 Hardware	47
2.1.6.2 Software	48
2.1.7 Análisis de requerimientos.....	48
2.1.8 Reunión o entrevista con las personas involucradas.....	49
2.2 La Empresa de Telecomunicaciones	49
2.2.1 Definición	49
2.3 Funcionamiento de una empresa de telecomunicaciones	50
2.3.1 Organigrama de la Empresa Claro	50
2.3.2 Área de Sistemas.....	50
2.3.3 Departamento de Procesos y Calidad	51
2.3.4 Funciones del área Comercial, y su importancia	51
2.3.5 Problema	52
2.3.6 Solución.....	52
2.4 Metodología en cascada a seguir	53
a. Estudio de factibilidad.....	53
b. Requerimientos funcionales y no funcionales.....	53
c. Arquitectura de flujo de carga de datos y la arquitectura del sistema.....	56

d. Configuración de la infraestructura	58
e. Diseño del sistema del almacén de datos.....	61
f. Desarrollo	64
f.1 Creación de los almacenes de datos (<i>data stores</i>).....	64
f.2 Los procesos de extracción, transformación, y carga.....	64
f.2.1 ETLs (Fuente al <i>stage</i>), Transformaciones y JOBS	66
f.2.1.1 Transformación TRS_CARGA_ARCHIVOS_FUENTE_CSV.....	66
f.2.1.2 Transformación TRS_LIMPIEZA_DATOS_FUENTE	66
f.2.1.3 Transformación TRS_CARGA_ARCHIVO_FUENTE_XLS.....	67
f.2.1.4 Transformación TRS_PREPARACION_FINAL_DATOS	68
f.2.1.5 JOB_CARGA_STAGE	69
f.2.2 ETLs (<i>stage</i> al <i>data mart</i>), carga de tablas de dimensiones y de hechos.....	70
f.2.2.1 Transformación TRS_D_FECHA	70
f.2.2.2 Transformación TRS_D_CAC.....	70
f.2.2.3 Transformación TRS_D_PLAN	71
f.2.2.4 Transformación TRS_D_CLIENTE	72
f.2.2.5 Transformación TRS_D_ASESOR.....	73
f.2.2.6 JOB_CARGA_DIMENSIONES	74
f.2.2.7 Transformación TRS_F_VENTAS.....	75
f.2.2.8 JOB_CARGA_HECHOS	77
f.2.2.9 JOB_CARGA_DM_VENTAS	77
f.3 La interface de usuario final.....	77
f.3.1 Indicadores de gestión y soluciones de análisis.....	77
f.3.1.1 Reporte con los indicadores de gestión	78
f.3.1.2 Cubo OLAP con los indicadores de gestión.....	79
f.3.1.3 Tablero de mando con los indicadores de gestión	80
g. Pruebas	81

h. Despliegue.....	82
i. Operación	82
j. Administración del proyecto	83
2.5 Encuestas y análisis de resultados de las encuestas.....	83
2.5.1 Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de Pentaho.	83
2.5.2 La tabulación de los resultados de la encuesta se encuentra detallada en el anexo 7.	85
Capítulo 3	86
3.1 Cuadro comparativo de soluciones de inteligencia de negocios.....	86
3.2 Detalle de costos	88
3.3 Análisis costo – beneficio de la implementación de la solución de inteligencia de negocios para el módulo de ventas de Claro	89
3.3.1 Identificar los costos y beneficios.....	90
3.3.2 Análisis costo – beneficio Generación de reportes de ventas de los años 2012, 2013, y 2014 con y sin el uso de la herramienta de inteligencia de negocios	91
Capítulo 4	95
4. Conclusiones y Recomendaciones	95
4.1 Conclusiones	95
4.2 Recomendaciones	97
ANEXOS	101

ÍNDICE DE FIGURAS

Figura 1 La Fábrica de Información Corporativa propuesta por Bill Inmon.....	6
Figura 2 Arquitectura Multidimensional	9
Figura 3 Simplificación de la Fábrica de Información Corporativa (CIF).....	11
Figura 4 Comparación entre arquitecturas CIF y MD	12
Figura 5 Cuadrante mágico de Gartner.....	14
Figura 6 Data Mart dependiente propuesto por Bill Inmon	21
Figura 7 Data Mart independiente propuesto por Ralph Kimball	22
Figura 8 Arquitectura de flujo de datos de un almacén de datos	24
Figura 9 Arquitectura DDS	26
Figura 10 Arquitectura NDS + DDS.....	27
Figura 11 Arquitectura ODS + DDS.....	28
Figura 12 Arquitectura de un almacen de datos federado.....	29
Figura 13 Arquitectura de un sistema de un almacén de datos.....	30
Figura 14 Metodología en cascada	31
Figura 15 Metodología iterativa	34
Figura 16 Plataforma de <i>Pentaho BI</i>	36
Figura 17 Grafica conceptual de <i>Pentaho Community Edition</i>	40
Figura 18 Interface gráfica Spoon	41
Figura 19 Arquitectura de <i>Pentaho Data Integration</i>	42
Figura 20 Organigrama Claro.....	50
Figura 21 Organigrama departamento de Sistemas.....	51
Figura 22 Arquitectura de flujo de carga de datos.....	56
Figura 23 Arquitectura del sistema.....	57
Figura 24 Consola de usuario de Pentaho	59
Figura 25 Configuración de la consola administrativa de Pentaho.....	60
Figura 26 Configuración a la base de datos DM_VENTAS	60

Figura 27 Sistema del almacén de datos	61
Figura 28 Modelo dimensional de datos para el data mart DM_VENTAS	63
Figura 29 Transformación TRS_ARCHIVOS_FUENTE_CSV	66
Figura 30 Transformación TRS_LIMPIEZA_DATOS_FUENTE	66
Figura 31 Limpieza de datos campo cédula.....	67
Figura 32 Limpieza de datos campo fecha_ingreso	67
Figura 33 Transformación TRS_CARGA_ARCHIVO_FUENTE_XLS	68
Figura 34 Transformación TRS_PREPARACION_FINAL_DATOS.....	68
Figura 35 Código SQL para consolidación de datos	69
Figura 36 JOB_CARGA_STAGE	69
Figura 37 Transformación TRS_D_FECHA	70
Figura 38 Transformación TRS_D_CAC	71
Figura 39 Código SQL para mapeo de datos.....	71
Figura 40 Transformación TRS_D_PLAN	72
Figura 41 Código SQL para mapeo de datos.....	72
Figura 42 Transformación TRS_D_CLIENTE.....	73
Figura 43 Código SQL para mapeo de datos.....	73
Figura 44 Transformación TRS_D_ASESOR.....	74
Figura 45 Código SQL para mapeo de datos.....	74
Figura 46 JOB_CARGA_DIMENSIONES.....	75
Figura 47 Transformación TRS_F_VENTAS.....	76
Figura 48 Código SQL para mapeo y ofuscación de datos.....	76
Figura 49 JOB_CARGA_HECHOS	77
Figura 50 JOB_CARGA_DM_VENTAS.....	77
Figura 51 Reporte de Ventas Claro.....	79
Figura 52 Cubo OLAP.....	80
Figura 53 Tablero de mando de Ventas Claro.....	81
Figura 54 Reporte de ventas en Excel	82

ÍNDICE DE TABLAS

Tabla 1 Diferencias entre la fábrica de información corporativa y la arquitectura multidimensional	13
Tabla 2 Características del hardware para el <i>data mart</i> de ventas.....	47
Tabla 3 Características del software para el <i>data mart</i> de ventas.....	48
Tabla 4 Requerimientos funcionales	54
Tabla 5 Requerimientos no funcionales	55
Tabla 6 Indicadores de gestión	78
Tabla 7 Cuadro de costos de soluciones de inteligencia de negocios	86
Tabla 8 Detalle de costos	88
Tabla 9 Tipos de Costos y Beneficios	91
Tabla 10 Análisis costo – beneficio Generación de reportes de ventas año 2012.....	92
Tabla 11 Análisis costo – beneficio Generación de reportes de ventas año 2013.....	92
Tabla 12 Análisis costo – beneficio Generación de reportes de ventas año 2014.....	93
Tabla 13 Resumen Análisis costo – beneficio con BI.....	93

INTRODUCCIÓN

Las empresas hoy en día manejan grandes volúmenes de información, pero la misma se encuentran dispersos en diferentes fuentes como: Bases de datos, archivos planos, excel, csv, xml, entre otros. Este proyecto de tesis evidencia que el manejo de la inteligencia de negocios ayuda sustancialmente a que las empresas administren de manera eficiente todos los datos para transformarlas en información que ayude a las áreas gerenciales y jefaturas a tomar las mejores decisiones enfocadas a los objetivos del negocio.

Inteligencia de negocios con sus siglas en inglés *BI* reúne datos históricos de varias fuentes, y presenta la información de manera más sencilla, rápida y entendible.

La empresa de telecomunicaciones Claro brinda servicios como: Telefonía celular, internet móvil, fijo y televisión por cable. Por medio de una solución de inteligencia de negocios se puede analizar los datos existentes y hacer inteligencia de negocios en la empresa.

Para este proyecto de tesis se hace uso de la plataforma *Pentaho Community suite Edition*, la cual está compuesta por las herramientas necesarias para desarrollar una solución de inteligencia de negocios.

Por medio de esta herramienta se aplicará inteligencia de negocios, para el análisis de los requerimientos del módulo de ventas, con lo cual se construirá un *data mart* que apoye a la toma de decisiones gerenciales. Se seleccionó una metodología en cascada para la realización del proyecto y de esta forma se obtuvo las soluciones de análisis como: Un cubo de procesamiento analítico en línea (*OLAP*), un reporte y un tablero de mando que permitan aplicar inteligencia sobre el negocio.

La implementación de la solución se la va a realizar de manera local, únicamente en las agencias de Quito.

Capítulo 1

1 Introducción a las tecnologías de inteligencia de negocios

1.1 Historia sobre inteligencia de negocios

La inteligencia de negocios es un compendio de aplicaciones, que permiten extraer la información de las distintas fuentes de la compañía, almacenarla, analizarla y proveerla a los usuarios para la toma de decisiones.

La inteligencia de negocios engloba completamente a la compañía, debe apoyarse en un diseño robusto y escalable para soportar las distintas herramientas de inteligencia de negocios que existen en la actualidad y en el futuro. Los distintos modelos de inteligencia de negocios, fueron planteados por: Ralph Kimball y Bill Inmon.

Uno de los recursos más importantes de una empresa son sus datos, pero es importante que a partir de estos datos, se pueda generar información para su respectivo análisis, de esta manera la compañía se vuelve altamente competitiva.

Es importante tomar en cuenta que la inteligencia de negocios y sus herramientas han evolucionado de acuerdo a las necesidades del mercado. Estas herramientas disponen de distintas funcionalidades como la consolidación y el análisis de información además de la generación de reportes. Es posible visualizar toda esta información por medio de dispositivos móviles, *tablets*, o por correo. (Imhoff, Galemme y Geiger, 2003, p. 3-5)

1.2 Significado de inteligencia de negocios

La inteligencia de negocios consiste en recopilar datos desde distintas fuentes para transformarlas en información y posteriormente realizar su respectivo análisis y de esta manera ayudar para que las personas encargadas puedan tomar cada decisión de tipo: estratégico, táctico, y operacional. (Rainardi, 2008, p. 12)

A continuación se detallan algunas aplicaciones de inteligencia de negocios:

- Gestión del desempeño empresarial (*BPM*): Por medio de la creación de indicadores claves de rendimiento, los cuales permiten la toma de decisiones tácticas. (Rainardi, 2008, p. 13)
- Análisis de clientes: Por medio del uso de una herramienta *Customer Relationship Management (CRM)*, permite identificar los clientes potenciales para la empresa. (Rainardi, 2008, p. 13)
- Análisis estadístico: Al analizar una muestra de los datos permite proyectar algunos escenarios. Es decir, en una compañía de telecomunicaciones, en base a un análisis estadístico, se podría establecer promociones a ciertos productos y si se revisa el historial de ventas, se puede determinar que marca y modelo de celular se vende frecuentemente. (Rainardi, 2008, p. 13)
- Análisis predictivo: Permite identificar información importante dentro de volúmenes muy grandes de información de una manera precisa por medio de herramientas de minería de datos. Además de pronosticar las ventas, ingresos y costos con el fin de planificar los presupuestos del próximo año y tomando en cuenta otros factores tales como: El crecimiento orgánico, la situación económica, y la dirección futura de la empresa. (Rainardi, 2008, p. 13).

Tipos de herramientas de inteligencia de negocios:

- Reportes.
- Procesamiento analítico en línea (*OLAP*).
- Minería de datos. (Rainardi, 2008, p. 13 - 14)

1.2.1 Reportes

En el contexto de la inteligencia de negocios, un informe es una aplicación que recobra información obtenida del almacén de datos (*DWH*) o *data mart* y lo

presenta a los usuarios en pantalla o en papel. Bajo demanda los usuarios pueden solicitar que los reportes les sean enviados de manera automática a sus correos electrónicos y celulares para su respectivo análisis.

Los reportes se construyen de acuerdo a las especificaciones funcionales. Despliegan la información necesaria para ser analizado y entender las situaciones del negocio por medio del usuario. La forma más común de reporte es en forma de tabla que contiene columnas simples. Existe otra forma de reporte conocido como tabla de referencias cruzadas o de matriz, los cuales son como tablas dinámicas de Excel, en donde un atributo de dato se convierte en fila y otro atributo de dato se convierte en columna y cada celda en el reporte contiene el valor correspondiente a los atributos de fila y columna.

Los reportes del almacén de datos se utilizan para presentar los datos de negocio a los usuarios para la toma de decisiones basadas en información. (Rainardi, 2008, p. 13)

1.2.2 Procesamiento analítico en línea

Es el proceso de analizar de forma interactiva los datos de transacciones del negocio almacenados en el almacén de datos para tomar decisiones estratégicas y tácticas sobre la empresa. Las personas que realizan el procesamiento analítico en línea son los analistas del negocio, jefaturas y gerencias.

La funcionalidad típica del procesamiento analítico en línea incluye la agregación (un total), perforación hacia abajo (obtener los detalles), y *slice and dice* (cortar el cubo y sumar los valores en las celdas). El procesamiento analítico en línea que utiliza una base de datos relacional, se conoce como el procesamiento analítico en línea relacional (*ROLAP*). El procesamiento analítico en línea que utiliza una base de datos multidimensional, se conoce como el procesamiento analítico en línea multidimensional (*MOLAP*).

Un ejemplo del procesamiento analítico en línea es analizar la efectividad de una iniciativa de campaña de marketing en ciertos productos por medio de la medición del crecimiento de las ventas en un período determinado. Otro ejemplo es el de analizar el impacto de un aumento de precios para la venta de productos en las diferentes regiones y grupos de productos en el mismo período de tiempo. (Rainardi, 2008, p. 14)

1.2.3 Minería de datos

Son pasos para explorar datos y encontrar patrones y relaciones que describen los datos y de predecir los valores desconocidos o futuros de los datos. Lo más importante en la minería de datos es la capacidad de entender por qué algunas cosas sucedieron en el pasado y predecir lo que puede ocurrir en el futuro. Cuando se utiliza la minería de datos para exponer eventos en el presente o pasado, se denomina análisis descriptivo. Cuando se utiliza para predecir el futuro, se llama análisis predictivo. (Rainardi, 2008, p. 14)

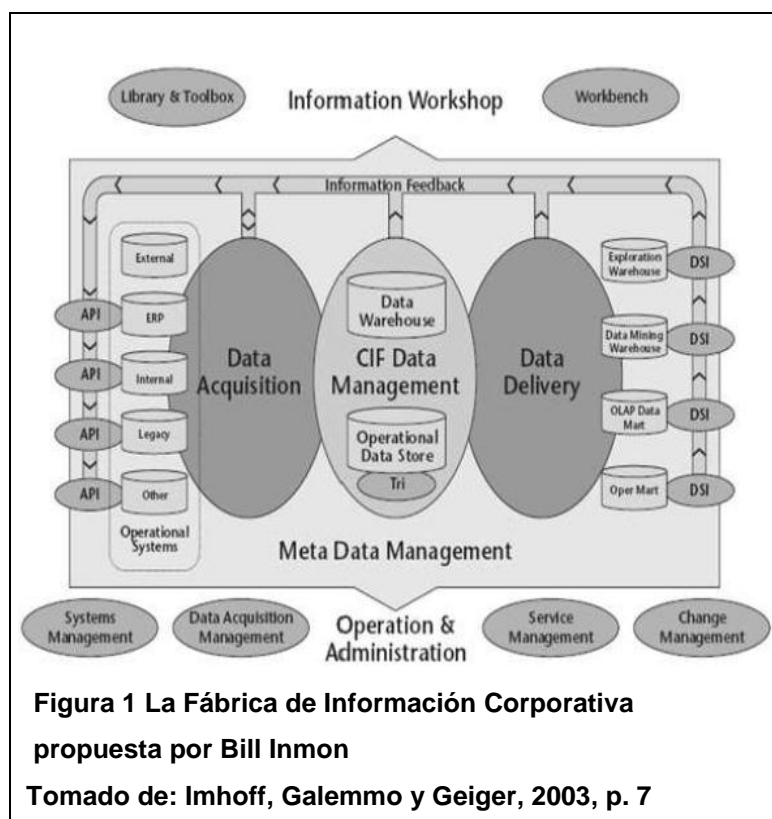
Por ejemplo en el negocio de telecomunicaciones detectar patrones que muestre que tan óptimas son las promociones que se publican cada mes o de ser necesario revisar si se debe incrementarlas para mejorar las ventas diarias.

1.3 Arquitectura de la inteligencia de negocios

1.3.1 La Fábrica de Información Corporativa

También denominada *CIF*. “Es una arquitectura conceptual ampliamente aceptada, la cual describe y clasifica los almacenes de información que se utilizan para operar y administrar una infraestructura de inteligencia de negocios robusta”. (Imhoff, Galemmeo y Geiger, 2003, p. 11)

Se observa la fábrica de información corporativa en la siguiente ilustración.



Estos almacenes de información soportan tres procesos de organización de alto nivel:

- Operaciones del negocio: Tienen que ver con las operaciones del día a día del negocio. Dentro de esta función se encuentran los sistemas de procesamiento de transacciones operacionales y datos externos, los cuales ayudan a manejar el negocio, y por lo general son altamente automatizados. Los procesos que apoyan esta función son bastante estáticos. (Imhoff, Galemno y Geiger, 2003, p. 11)
- Inteligencia de negocios: Comprende la búsqueda permanente de una mejor visión de la empresa, sus productos y de sus clientes. Incluye los procesos que están en constante mejora, los cuales pueden cambiar a medida que los analistas de negocio y los usuarios revisan la información disponible para ellos, el uso de esa información sirve para

desarrollar nuevos productos, medir la retención de clientes, analizar nuevos mercados potenciales. (Imhoff, Galemmo y Geiger, 2003, p. 11)

- Manejo del negocio: Es la función en la cual el conocimiento y las nuevas ideas desarrolladas en la inteligencia de negocios se institucionalizan y se introducen en las operaciones diarias del negocio en toda la empresa.

La fábrica de información corporativa se puede utilizar para identificar todas las actividades de gestión de la información que una empresa lleva a cabo. A continuación se explica brevemente cada componente: (Imhoff, Galemmo y Geiger, 2003, p. 12)

- Sistemas operacionales: Se enfocan en el procesamiento de transacciones, utilizan una gran variedad de tecnologías y arquitecturas. Los sistemas operacionales son estáticos por naturaleza, cambian sólo en respuesta a un cambio intencional en las políticas o las reglas del negocio o por razones técnicas, tales como el mantenimiento del sistema o la optimización del rendimiento. Estos sistemas soportan en tiempo real el procesamiento de transacciones, los sistemas operacionales representan la primera aplicación de las reglas del negocio hacia los datos de una empresa y la calidad de los datos de los sistemas operacionales tiene un impacto directo en la calidad de toda la información utilizada en la empresa. (Imhoff, Galemmo y Geiger, 2003, p. 12)
- Adquisición de datos: Es el conjunto de procesos que captura, integra, transforma, limpia y carga los datos fuente en el almacén de datos y en el *Data Store* operacional (*ODS*). (Imhoff, Galemmo y Geiger, 2003, p. 12)
- *Data Store* operacional (*ODS*): Se utiliza para la toma de decisiones tácticas, mientras que el almacén de datos soporta las decisiones estratégicas. (Imhoff, Galemmo y Geiger, 2003, p. 13)

- Entrega de datos: La entrega de datos se limita generalmente a operaciones tales como la agregación de los datos, filtrado por dimensiones específicas o reglas del negocio, cambio de formato de datos para facilitar el acceso a los usuarios finales o para dar soporte a las herramientas de inteligencia de negocios y finalmente entrega o transmisión de datos para toda la compañía. (Imhoff, Galemmo y Geiger, 2003, p. 14)
- Manejo de los meta datos: Es el conjunto de procesos en donde se recopila, administra e implementa la *metadata*, por medio de toda la fábrica de información corporativa. (Imhoff, Galemmo y Geiger, 2003, p. 1)

Está compuesto por:

- ✓ Metadatos técnicos.
- ✓ Metadatos de negocio.
- ✓ Metadatos administrativos.
- Retroalimentación de la información: Es el mecanismo que permite el intercambio de inteligencia y conocimientos adquiridos a través del uso de la fábrica de información corporativa para ser compartida con otros almacenes de datos. (Imhoff, Galemmo y Geiger, 2003, p. 15)
- Taller de información: Son el grupo de herramientas para los usuarios del negocio, para ayudarlos a utilizar recursos de la fábrica de información corporativa. Es una forma de organizar y categorizar los datos en la fábrica de información corporativa. Por ejemplo la intranet, el portal *web*. (Imhoff, Galemmo y Geiger, 2003, p. 15)
- Operaciones y administración: Se administran los constantes cambios, el manejo de los datos, sistemas, en general toda la fábrica de información corporativa. (Imhoff, Galemmo y Geiger, 2003, p. 16)

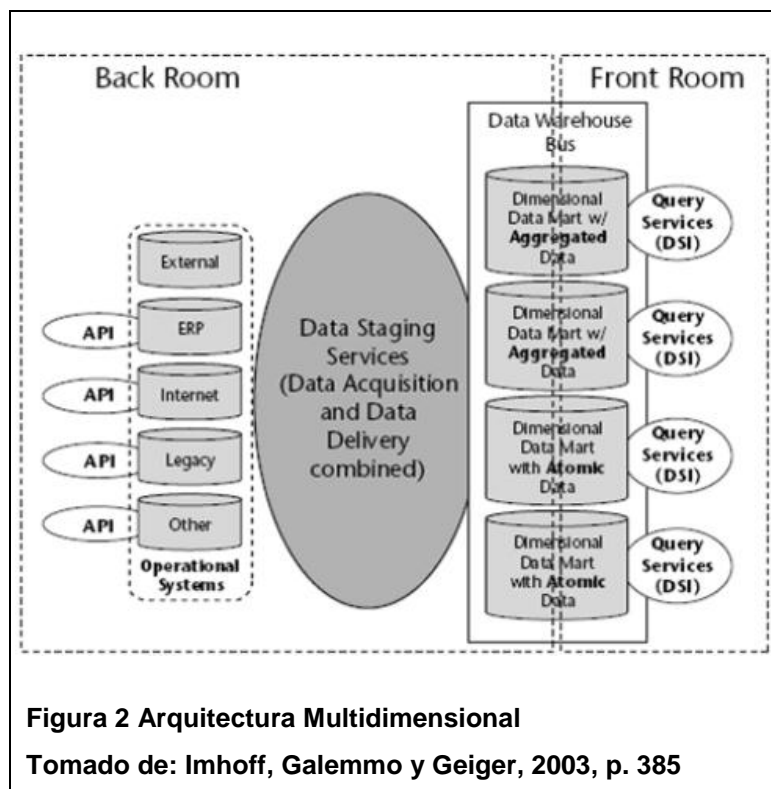
1.3.2 Arquitectura Multidimensional

También denominada *MD*. Un almacén de datos se construye a lo largo del tiempo a través de una serie de *data marts* que se enfocan en áreas específicas de la empresa, la suma de estos *data marts* se denomina bus del almacén de datos (*dwh Bus*), esto significa que los *data marts* pueden ser conectados según la necesidad de análisis.

La arquitectura multidimensional se basa en la condición que los análisis de inteligencia de negocio poseen como base un esquema multidimensional. El diseño estrella es el frecuentemente utilizado para la toma de decisiones estratégicas. Cabe señalar que la diferencia más importante entre la Fábrica de Información Corporativa y la Arquitectura Multidimensional es la definición del *data mart*. (Imhoff, Galemme y Geiger, 2003, p. 383 - 384)

La arquitectura Multidimensional se divide en dos grupos: El *back room* y el *front room*.

En la siguiente figura se observa la arquitectura multidimensional:



1. El *back room*: En este componente es donde se lleva a cabo el proceso de almacenamiento y adquisición de datos. Dentro de este componente se ubican los sistemas operacionales en donde esta data es transportada al área de almacenamiento temporal denominado *stage area*. Esta área está compuesta por las dimensiones conformadas y es el lugar en que las claves subrogadas (*Subrogate Keys - SK*) se generan. La mayor parte del trabajo realizado por los procesos *ETL* se produce aquí.

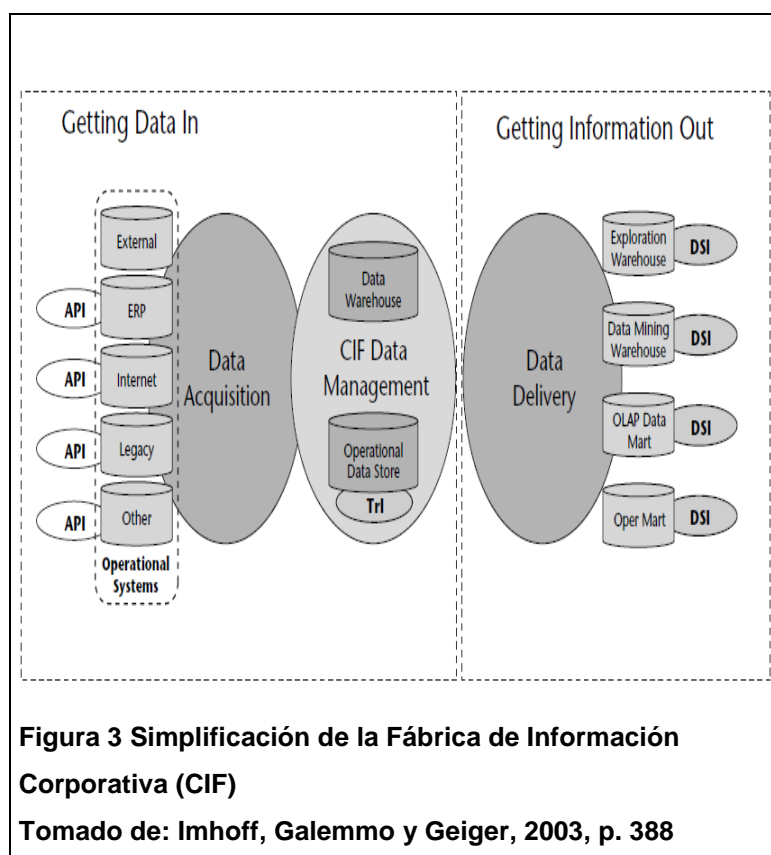
La arquitectura bus del almacén de datos está compuesta por dos tipos de *data marts*:

- ✓ *Data mart* atómico: Contienen datos multidimensionales en el nivel más bajo de detalle disponible en todo el ambiente. Pueden contener algunos datos agregados para mejorar el rendimiento de la consulta. Los datos son almacenados en un modelo de datos de esquema en estrella.
- ✓ *Data mart* agregado: Contienen datos relacionados con un núcleo de procesos de negocio como el marketing, las ventas, o las finanzas. Es posible crear un *data mart* agregado directamente desde el *Data stage area*, de igual manera los datos se almacenan en un esquema en estrella.

El uso para estos dos tipos de *data marts* depende de los requerimientos del negocio y del rendimiento de cada una de estas estructuras en el ambiente.

2. El *front room*: Este componente es la interfaz para la comunidad empresarial. Es similar a la función de “sacar la información” de la fábrica de información corporativa. Los dos tipos de *data marts* mencionados anteriormente atómico y agregado son la fuente de datos para estas interfaces y herramientas. (Imhoff, Galemno y Geiger, 2003, p. 384-386)

En la figura 1.3 se muestra una simplificación de la fábrica de información corporativa:



1.3.3 Comparativas sobre las arquitecturas: Fábrica de Información Corporativa y Multidimensional

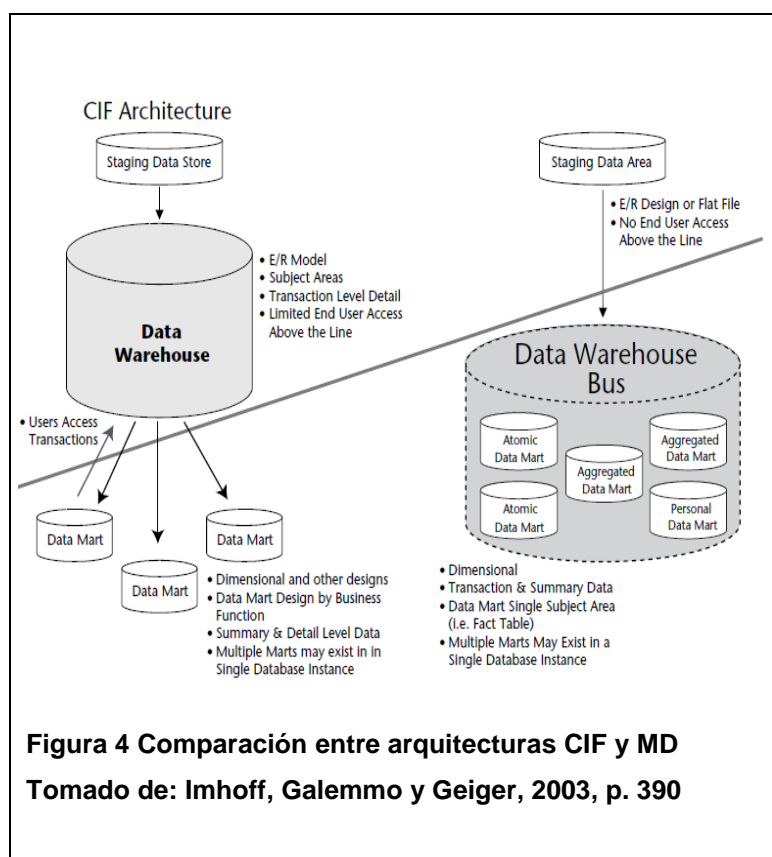
Ambas arquitecturas tienen un área de almacenamiento temporal, manejo de metadatos y procesamiento de adquisición de datos. Los diseños de los *data marts* son predominantemente multidimensionales para ambas arquitecturas, aunque la fábrica de información corporativa no se limita sólo a este diseño y puede soportar un conjunto mucho más amplio de técnicas de diseño de *data marts*.

Cada diseño maneja su propio esquema. La fábrica de información corporativa lo denomina *TOP – DOWN*, el cual provee una vista corporativa de todos los datos por medio de un almacén de datos y luego dependiendo de los requerimientos del negocio va hacia abajo construyendo los *data marts*.

requeridos. Por otro lado, la arquitectura multidimensional maneja un enfoque *BOTTOM – UP*, es decir para llegar a un almacén de datos completo se van construyendo una serie de *data marts* que se centran en áreas específicas de la empresa. La suma de los *data marts* compone el bus del almacén de datos.

Uno de los principales objetivos de cada una de las arquitecturas es minimizar el tiempo y costo de la implementación. (Imhoff, Galemno y Geiger, 2003, p. 389)

En la siguiente figura se observa una comparación entre las arquitecturas de la fábrica de información corporativa y la multidimensional.



En la siguiente tabla se observa las diferencias más importantes entre la fábrica de información corporativa y la multidimensional:

Tabla 1 Diferencias entre la fábrica de información corporativa y la arquitectura multidimensional

Característica	Fábrica de Información Corporativa (CIF)	Arquitectura Multidimensional (MD)
Técnica de modelamiento de Datos	Relacional y dimensional	Dimensional
Enfoque	TOP - DOWN	BOTTOM - UP
Vista empresarial de datos	Total	Parcial
Entrega de datos detallados	✓	✓
Entrega de datos resumizados	✓	✓
Tiempo de implementación	Alto	Bajo
Costo de implementación	Alto	Bajo

1.4 Aplicaciones de inteligencia de negocios

Según el cuadrante mágico de Gartner 2014, los líderes del mercado en plataformas de inteligencia de negocios y análisis son:

- ✓ Tableau
- ✓ QlikView
- ✓ Microsoft

La figura 1.5 muestra el cuadrante mágico de Gartner:



Actualmente en el mercado existen herramientas de inteligencia de negocios, las cuales tienen en común las siguientes características:

- ✓ Acceso a la información.
- ✓ Apoyo a la toma de decisiones.
- ✓ Orientación al usuario final.

Entre las herramientas de inteligencia de negocios con mayor uso se encuentran: *SAP Business Objects*, *IBM Cognos*, *MicroStrategy*, *Oracle BI*, *Oracle Hyperion*. Existen también herramientas de código libre tales como: *Pentaho*, *SpagoBI*, *Eclipse BIRT*, *JasperReports*. (Gartner, 2014)

Para este proyecto de tesis se va a utilizar la herramienta de código libre *Pentaho BI suite*, debido a que es una multiplataforma que contiene varias soluciones para inteligencia de negocios, adicionalmente la versión libre no tiene costo.

1.5 Preferencias de la inteligencia de negocios dentro de las empresas

El futuro de la inteligencia de negocios es amplio, está cambiando cómo se manejan las empresas, la toma de decisiones y cómo los empleados realizan sus trabajos. Han ido surgiendo nuevas herramientas y el conjunto de estas herramientas se ha consolidado como plataformas de inteligencia de negocios como por ejemplo *Pentaho BI Suite*.

De la misma manera existen nuevas tendencias de inteligencia de negocios tales como: Inteligencia de negocios de autoservicio, inteligencia de negocios móvil, prestaciones colaborativas y sociales, inteligencia de negocios en la nube, inteligencia de negocios de código libre, *Big Data*, percepción en tiempo real, pero uno de los principales y más eficaces son los *Business Intelligence Competency Center (BICC)*, que consisten en un equipo de trabajo que ha definido tareas, funciones, responsabilidades y procesos para apoyar y promover el uso efectivo de la inteligencia de negocios en una empresa, el objetivo es que la inteligencia de negocios sea considerado como una unidad de apoyo.

1.5.1 Centro de competencia de inteligencia de negocios (BICC)

Las empresas se han dado cuenta que existe más de la inteligencia de negocios que solo utilizar las herramientas tecnológicas. Es necesario dar una orientación integral y clave a la inteligencia de negocios, así como al capital humano, los procesos de conocimiento y la cultura, pues sin estrategias, los resultados son inconsistentes.

El centro de competencia de inteligencia de negocios es un equipo multifuncional con una estructura organizativa permanente. Su personal está conformado por los empleados de la organización en sí, pero algunos roles o funciones pueden ser contratados por empresas externas. El centro de competencia de inteligencia de negocios tiene la tarea de impulsar el uso de inteligencia de negocios en toda la empresa. Permite a la empresa coordinar y complementar esfuerzos existentes en el área de inteligencia de negocios,

mientras disminuye la redundancia, aumenta la efectividad. La centralización de estos esfuerzos logra que la información y mejores prácticas sean comunicadas y compartidas a través de toda la empresa. (Miller, Brautigam, Gerlach, 2006, p. 9-10)

Los motivos principales para establecer un centro de competencia de inteligencia de negocios en una empresa son:

- ✓ Conservar y explotar el valor de las inversiones en tecnología.
- ✓ Integrar y consolidar los procesos de inteligencia de negocios y análisis.
- ✓ Reducir el riesgo en la ejecución de proyectos
- ✓ Dar soporte a los usuarios del negocio para que puedan entender la información y realizar un análisis adecuado.
- ✓ Asegurar que lo aprendido por la inteligencia de negocios sea comunicado a toda la empresa.

1.6 Data Warehousing

1.6.1 Historia

En un inicio no existían mecanismos sencillos para el almacenamiento de información, había tarjetas perforadas y cintas de papel. El almacenamiento era muy caro y limitado. Luego con la cinta magnética, fue posible almacenar grandes volúmenes de datos a bajo costo, los datos podían ser escritos y sobrescritos. La cinta magnética representó un gran adelanto en comparación con los primeros métodos de almacenamiento antes mencionados, pero existían algunas desventajas como por ejemplo los datos solo se podían acceder secuencialmente y no era el medio más estable para escribir los datos. Entonces llegó el almacenamiento en disco, lo cual representó un adelanto para el almacenamiento de datos, los datos se podían acceder directamente y no secuencialmente y de forma simultánea. Pronto el almacenamiento en disco llegó de la mano de un software llamado sistema de gestión de base de datos, que consiste en un conjunto de programas que le permite almacenar, modificar y extraer información de una base de datos.

Así mismo llegaron las aplicaciones en línea como el procesamiento analítico en línea, debido a que los datos se podían acceder directamente por medio de almacenamiento en disco y un sistema de gestión de base de datos como por ejemplo se los cajeros automáticos y los sistemas de reservas de pasajes en línea.

En este punto el problema era que los usuarios finales no tenían acceso a su información directamente, entonces llegaron las computadoras personales, pero no bastaba con tener acceso a los datos y aparecieron otras complicaciones como: Los datos no eran correctos, datos incompletos, la información no llegaba a tiempo, redundancia de datos. (Inmon, Strauss, y Neushloss, 2010, p. 19-22)

Una vez corregidos los problemas anteriormente mencionados, los sistemas de procesamiento analítico en línea si bien es cierto tenían acceso a los datos almacenados pero se creó la necesidad de interpretar estos datos para poder analizarlos. Fue entonces cuando aparece la palabra almacén de datos, propuesta por Inmon, el cual indicó que el diseño del almacén de datos sea generado utilizando el modelo entidad- relación, mientras que Ralph Kimball proponía un modelo de datos dimensional.

Ambos autores Bill Inmon y Ralph Kimball son conocidos como los padres del *Data Warehousing*. Ambas metodologías se pueden implementar en una empresa, esto depende de las necesidades que cada una tiene en cuanto a aplicar inteligencia de negocios.

1.6.2 Concepto del almacén de datos

Existen dos definiciones sobre un almacén de datos.

- Bill Inmon propone que un almacén de datos es “un objeto orientado, integrado, no volátil, y variante en el tiempo de colección de datos en apoyo de toma decisiones”. (Rainardi, 2008, p. 16)

- Ralph Kimball propone que un almacén de datos es “un sistema que extrae, limpia, se ajusta, y entrega los datos de origen a un almacén de datos dimensional y luego da soporte e implementa consultas y análisis con el propósito de la toma de decisiones”. (Rainardi, 2008, p. 16)

Ambos autores coinciden en que un almacén de datos integra los datos de varios sistemas de origen. En el enfoque de Inmon, el almacén de datos se implementa físicamente como un almacén de datos normalizado. En el enfoque de Kimball, el almacén de datos se implementa físicamente como un almacén de datos dimensional. (Rainardi, 2008, p. 16)

Vincent Rainardi lo define como: "Un sistema que recupera y consolida los datos periódicamente desde los sistemas de origen a un almacén de datos dimensional o normalizado. Por lo general, almacena años de historia y es consultado para inteligencia de negocios u otras actividades de análisis". (Rainardi, 2008, p. 1)

A continuación se analiza cada proceso de un sistema de almacén de datos según Vincent Rainardi.

1.6.2.1 Recuperación de datos

Se realiza por un conjunto de rutinas conocido como un sistema *ETL* (*Extract, Transform and Load*), lo cual significa extracción, transformación y carga. El sistema *ETL* se compone de procesos, el cual recupera los datos de los sistemas de origen, los transforma y los carga a un sistema destino. Cada rutina tiene un proceso establecido. El de extracción obtiene la data desde los sistemas de origen, el proceso de transformación se utiliza para normalizar la data, los limpia y los verifica para establecer un formato y criterio en común y el proceso de carga deposita la data en los *data stores* dimensionales ó normalizados. (Rainardi, 2008, p. 4)

1.6.2.2 Consolidación de datos

Una empresa puede tener varios sistemas transaccionales. Un almacén de datos consolida dichos sistemas. La principal diferencia entre un almacén de datos y una aplicación transaccional es que los datos en un almacén de datos están integrados. Pero para que esto se lleve a cabo se debe tomar en cuenta lo siguiente:

- Disponibilidad de los datos: Cuando se consolida los datos de varios sistemas de origen, es posible que ciertos datos estén disponibles en un sistema A pero no en un sistema B. Es decir, el sistema A puede tener siete campos de dirección (dirección1, dirección2, dirección3, ciudad, código postal y país), pero el sistema B no tiene el campo dirección3 ni el campo país. (Rainardi, 2008, p. 5)
- Rangos del tiempo: Cada proceso tiene programado realizar cálculos en fechas establecidas. Por ejemplo el sistema A lo realiza semanalmente y el sistema B mensualmente. (Rainardi, 2008, p. 5)
- Definiciones: Dentro los sistemas de origen pueden existir datos con la misma información pero no necesariamente el mismo significado, esto depende de las reglas del negocio. Por ejemplo el término afluencia de clientes en un centro de atención en el sistema A significa ventas potenciales pero en el sistema B significa número de clientes atendidos. (Rainardi, 2008, p. 6)
- Conversiones: Cuando se consolida datos de diferentes sistemas de origen, es necesario validar si es necesario realizar la conversión de unidades de medida a una sola. Adicionalmente se debe de verificar en el período de tiempo en que se realizan las conversiones mencionadas. Por ejemplo la moneda que se maneja en el sistema A son dólares y en el sistema B son Euros. (Rainardi, 2008, p. 6)
- Empatar los datos (*Matching*): Para tener consistencia en los datos, es necesario que dichos datos coincidan entre los sistemas de origen. Por ejemplo el cliente 1 en el sistema A puede o no ser el mismo cliente 1 en

el sistema B. Para esto se debe de establecer una lógica para combinar los datos como número de cedula + nombres, apellidos + correo electrónico. (Rainardi, 2008, p. 6)

1.6.2.3 Periodicidad

Dentro de un almacén de datos se debe establecer los rangos en recuperar y consolidar la data, esto depende de los requerimientos de negocio de cada empresa pero también de los rangos que maneja los sistemas de origen. Por ejemplo si la gerencia requiere del reporte de ventas semanales, entonces los datos de todos los sistemas de origen se deben de consolidar semanalmente. (Rainardi, 2008, p. 6)

1.6.2.4 Reseña Histórica

La diferencia entre un sistema transaccional y un almacén de datos es que este último almacena datos históricos de varios años. La cantidad de información almacenada va a depender de los requerimientos del negocio de la empresa. Algo importante que se debe de mencionar es la técnica *Slowly Changing Dimension (SCD)*, la cual permite preservar información histórica sobre los datos dimensionales. (Rainardi, 2008, p. 10)

1.6.2.5 Consultas

Es el proceso de obtener datos de un almacén de datos que satisfaga algunos criterios. El propósito principal de la existencia de un almacén de datos es que sirve para que los usuarios puedan realizar consultas. El proceso de actualización no es permitido. Únicamente las herramientas *ETL* permiten actualizar el almacén de datos. (Rainardi, 2008, p. 11)

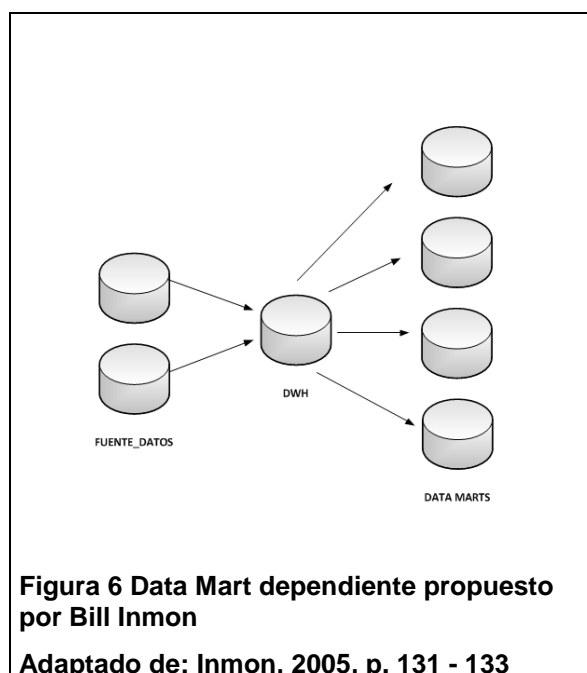
1.6.3 Concepto de un *data mart*

Es el lugar en donde el usuario final tiene acceso y control de sus datos para análisis. El *data mart* se compone de datos para un departamento. Por ejemplo finanzas, marketing, ventas y así sucesivamente. Un *data mart* ayuda a que un

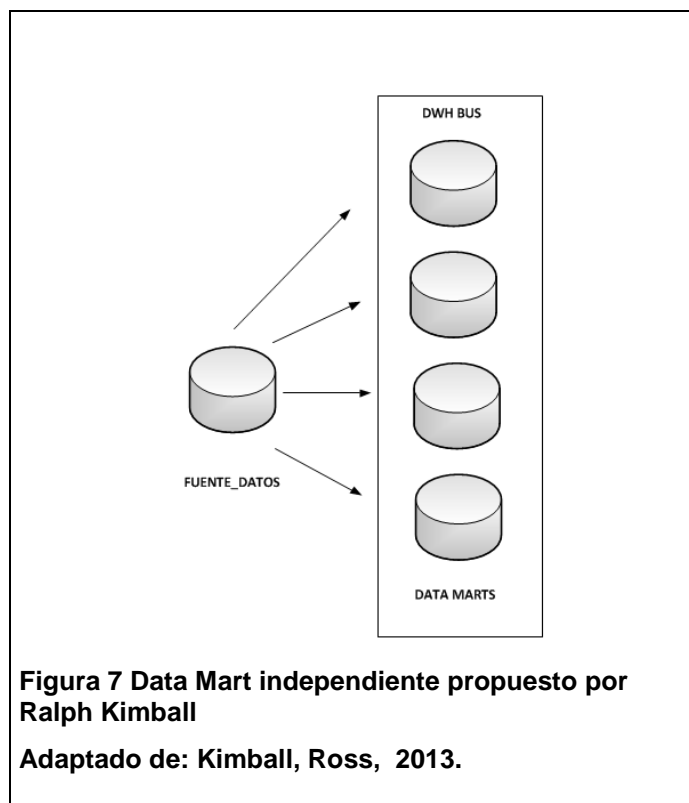
área específica de la empresa pueda tomar mejores decisiones. El *data mart* al igual que el almacén de datos están enfocados a la consulta. En resumen los *data marts* provienen del almacén de datos enfocados en un tema o área del negocio específico de una empresa. El tiempo de implementación de un *data mart* es corto en comparación a un almacén de datos. (Inmon, Strauss, y Neushloss, 2010)

Según lo propuesto por Inmon, un *data mart* puede ser dependiente, esto quiere decir que se deriva de un almacén de datos. Por otro lado lo propuesto por Kimball indica que puede ser un *data mart* independiente, el cual es construido en base a los sistemas de origen.

En la siguiente figura se observa el *data mart* dependiente.



En la siguiente figura se observa el *data mart* independiente.



1.6.4 Diseño de un almacén de datos

El sistema de almacén de datos se compone de:

- La arquitectura de flujo de datos: Indica cómo cada *data store* se organiza dentro del almacén de datos y de qué manera los datos fluyen desde los sistemas fuente hacia los usuarios.
- La arquitectura del sistema: Indica la disposición de cada servidor, disposición de red, programas, forma de almacenar y de los clientes. (Rainardi, 2008, p. 29)

1.6.4.1 La arquitectura de flujo de datos

Se compone de un arreglo de *data stores* al interior del sistema del almacén de datos, el cual muestra la manera en que fluyen los datos desde los sistemas

fuente a lo largo de los *data stores* hasta los programas utilizados por cada usuario final.

Los *data stores* son componentes importantes de la arquitectura de flujo de datos. Un *data store* se define como una o varias bases de datos o ficheros que contengan la data del almacén de datos estructurado en un formato definido e involucrados en los procesos del almacén de datos. (Rainardi, 2008, p. 29-30)

Basado en la accesibilidad de los usuarios los *data stores* del sistema del almacén de datos se clasifican en tres tipos:

- Orientada hacia el usuario (*User - facing data store*): Es un almacén de datos que está disponible para los usuarios finales y es consultado por los usuarios finales y las aplicaciones de usuario final. (Rainardi, 2008, p. 30)
- Interno (*Internal data store*): Se utiliza internamente por componentes del almacén de datos con el propósito de integrar, limpiar, conectar y la preparación de los datos. Estos *data stores* no disponen de acceso para consultas de cada usuario final. (Rainardi, 2008, p. 30)
- Híbrido (*Hybrid data store*): Se utiliza tanto para los mecanismos internos del almacén de datos como los de consulta por los usuarios finales y las aplicaciones de usuario final. (Rainardi, 2008, p. 30)

Basado en el formato de los datos, los *data stores* del almacén de datos se clasifican en cuatro tipos:

- Área temporal (*stage area*): Se utiliza para transformar y preparar la data que se obtiene de cada sistema fuente, antes que los datos sean cargados a otros *data stores*. (Rainardi, 2008, p. 30)
- *Data Store* normalizado (*NDS*): Se define como un *data store* maestro de forma de una o algunas bases de datos relacional normalizados con el objetivo de integrar los datos de diferentes sistemas de origen

almacenados en el *stage*, antes que los datos sean cargados en un *data store* orientado al usuario. (Rainardi, 2008, p. 30)

- *Data Store* operacional (*ODS*): Se define como un *data store* híbrido de forma de una o algunas bases de datos relacionales normalizadas, los cuales contienen la data de las transacciones además de la versión actualizada de la data maestra. (Rainardi, 2008, p. 30)
- *Data Store* dimensional (*DDS*): Se define como un *data store* orientado al usuario, de forma de una o algunas bases de datos relacionales, en la cual la data se encuentra organizado de manera dimensional.

Cuando se va a construir un sistema de almacén de datos se debe definir la arquitectura de flujo de datos a ser utilizada, debido a que afecta el tiempo y costo del proyecto. La arquitectura de flujo de datos muestra cómo los datos fluyen a través de los *data stores* dentro del almacén de datos.

A continuación se detallan las arquitecturas de flujos de datos:

- ✓ Arquitectura *DDS* simple.
- ✓ Arquitectura *NDS* + *DDS*.
- ✓ Arquitectura *ODS* + *DDS*.
- ✓ Arquitectura de *DWH* federado (*FDW*).

1.6.4.1.1 Arquitectura de flujo de datos de un almacén de datos

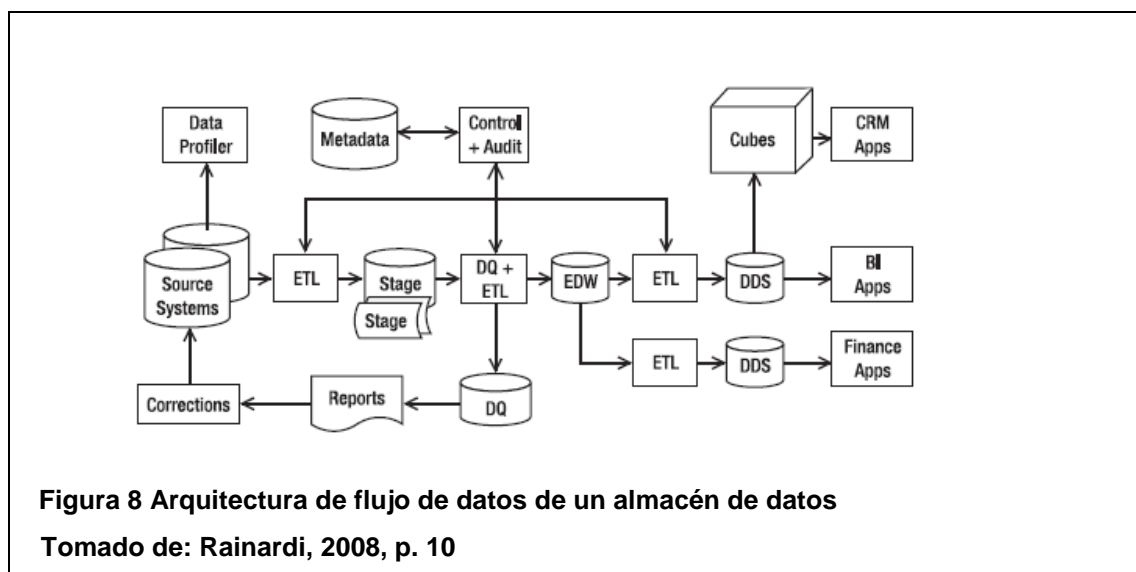


Figura 8 Arquitectura de flujo de datos de un almacén de datos

Tomado de: Rainardi, 2008, p. 10

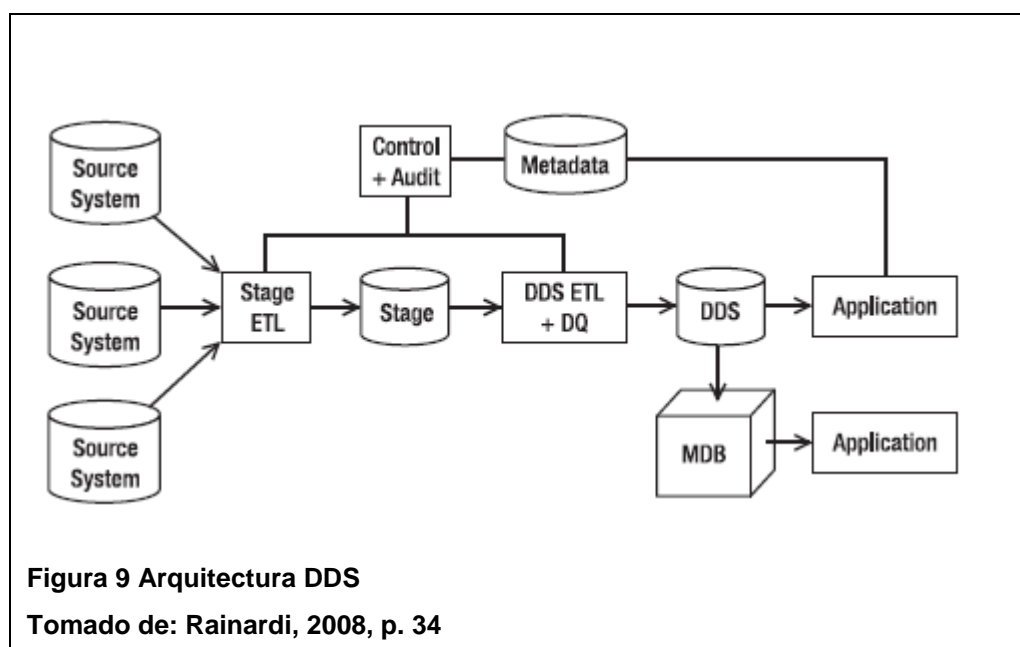
- El caracterizador de datos (*Data Profiler*) sirve para entender las características de los datos, además de realizar el análisis de los mismos.
- El sistema de extracción, transformación y carga (*ETL*) se enlazan a los sistemas de origen, en donde realizan el proceso de extraer, transformar y cargar la data a los sistemas finales, como el *stage*, *DDS*, *NDS* ó el almacén de datos.
- El *DDS* recopila los datos del almacén de datos en formato diferente al del *OLTP*. El motivo para obtener los datos desde los sistemas de origen hacia el *DDS* y luego consultar el *DDS* en lugar de consultar los sistemas de origen directamente se debe a que en un *DDS* la data está organizada de forma dimensional que es más adecuado para el análisis. La segunda razón es porque un *DDS* contiene datos integrados de varios sistemas de origen.
- La calidad de los datos (*DQ*) ejecutan diversas revisiones de control de la calidad de los datos. Los datos que no llegan a cumplir con cada regla es colocada en la base de datos de la calidad de los datos con el fin de que sea corregida en los sistemas de origen y nuevamente cargados al *stage*, almacén de datos normalizado, dimensional o al almacén de datos.
- El sistema de auditoría maneja un registro de cada operación de cada sistema y como es usado en la base de datos metadatos, se utiliza para conocer lo que pasó durante los procesos *ETL*.
- La metadata es una base de datos que almacena datos acerca de la estructuración de cada dato, que significa, utilización, la calidad de los datos, y demás información sobre cada dato.

- Las bases de datos multidimensionales (*MDB*) también denominados cubos. Es un tipo de base de datos, en el cual la data se almacena en celdas y la ubicación de la celda se determina por la cantidad de variables denominadas dimensiones. (Rainardi, 2008, p. 2-3)

1.6.4.1.2 Arquitectura *DDS* simple

Se manejan únicamente dos almacenes de datos. *Stage* y *DDS*. Los datos se extraen a través de los procesos *ETL* desde los sistemas de origen hacia el *stage*, luego de igual manera se extraen hacia el *DDS*. Se utiliza para una simple, rápida y sencilla solución de inteligencia de negocios, cuando se utiliza los datos sólo para alimentar a un almacén de datos dimensional. Una solución *DDS* es aplicable cuando se tiene sólo un sistema de origen debido a que no se necesita un *NDS* u *ODS* adicional para integrar los datos de los diferentes sistemas de origen. La arquitectura descrita tiene similitud con la definida por Ralph Kimball. (Rainardi, 2008, p. 34)

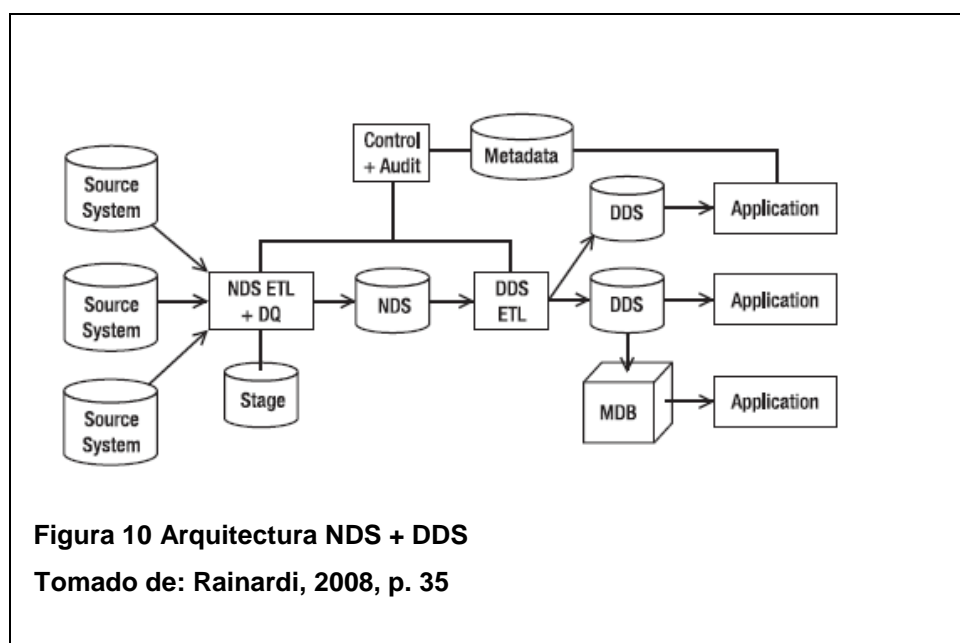
En la siguiente figura se observa la arquitectura simple con un *DDS*.



1.6.4.1.3 Arquitectura NDS + DDS

Es similar a la anterior pero además de los almacenes de datos *stage* y *DDS* también existe *NDS*, el cual está ubicado en frente del *DDS*. El *NDS* consolida los datos de los sistemas de origen, luego carga los datos en varias bases *DDS*. Una diferencia principal con la arquitectura *DDS* simple es que se pueden tener varias *DDS*. La arquitectura descrita tiene similitud con la definida por Bill Inmon debido a que primero se construye un almacén de datos normalizado, entonces de este se derivan los *data marts* necesarios como bases *DDS*. (Rainardi, 2008, p. 35)

En la siguiente figura se observa la arquitectura *NDS + DDS*.



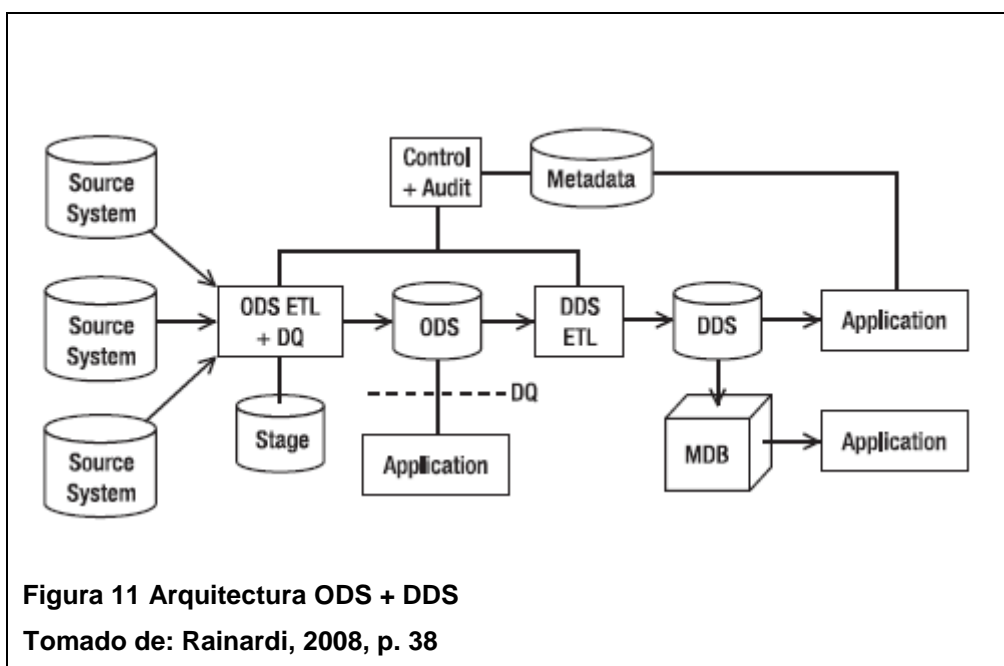
1.6.4.1.4 Arquitectura ODS + DDS

Es similar a la arquitectura anterior pero en lugar del *NDS* tiene un *ODS*, de igual manera se construyen los almacenes de datos dimensional (*DDS*) necesarios. El *ODS* está compuesta por la versión actual de los datos maestros, no contiene datos históricos y adicionalmente consolida la data de

los sistemas de origen. La data almacenada dentro del *ODS* son limpios e integrados además que ya pasaron por el proceso de calidad de datos (*DQ*).

Al combinar la arquitectura *NDS + DDS* con la arquitectura *ODS + DDS* se obtiene una fábrica de información corporativa definida por Inmon. (Rainardi, 2008, p. 38)

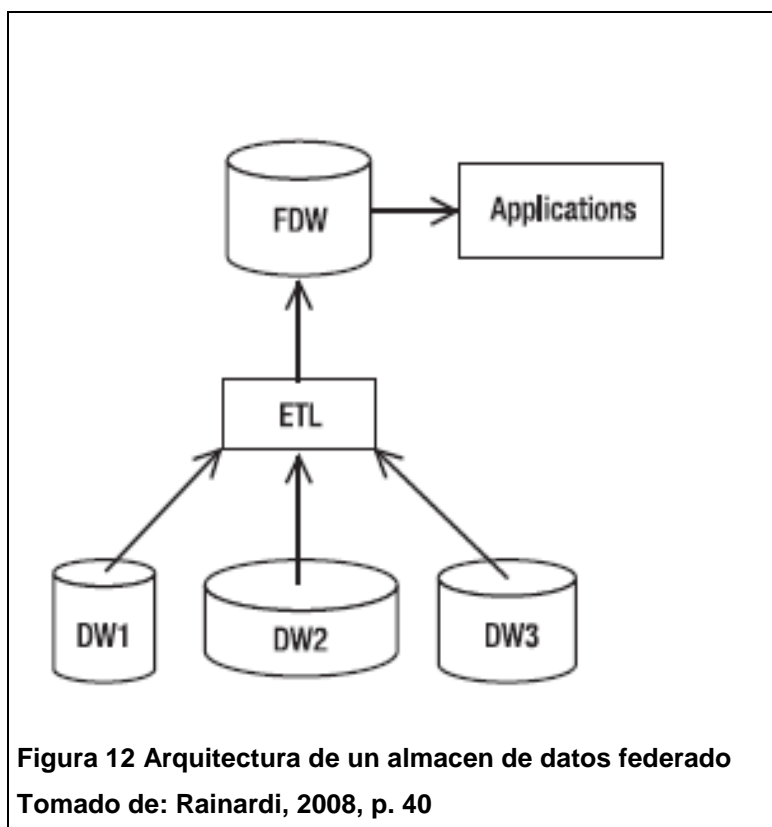
En la siguiente figura se observa la arquitectura *ODS + DDS*.



1.6.4.1.5 Arquitectura de un almacén de datos federado (*FDW*)

Está compuesto por varios almacenes de datos, los cuales pueden ser *DDS*, *NDS*, *ODS*. Un almacén de datos federada recupera los datos de los almacenes de datos existentes utilizando un proceso *ETL* y carga los datos en un nuevo almacén de datos dimensional (*DDS*).

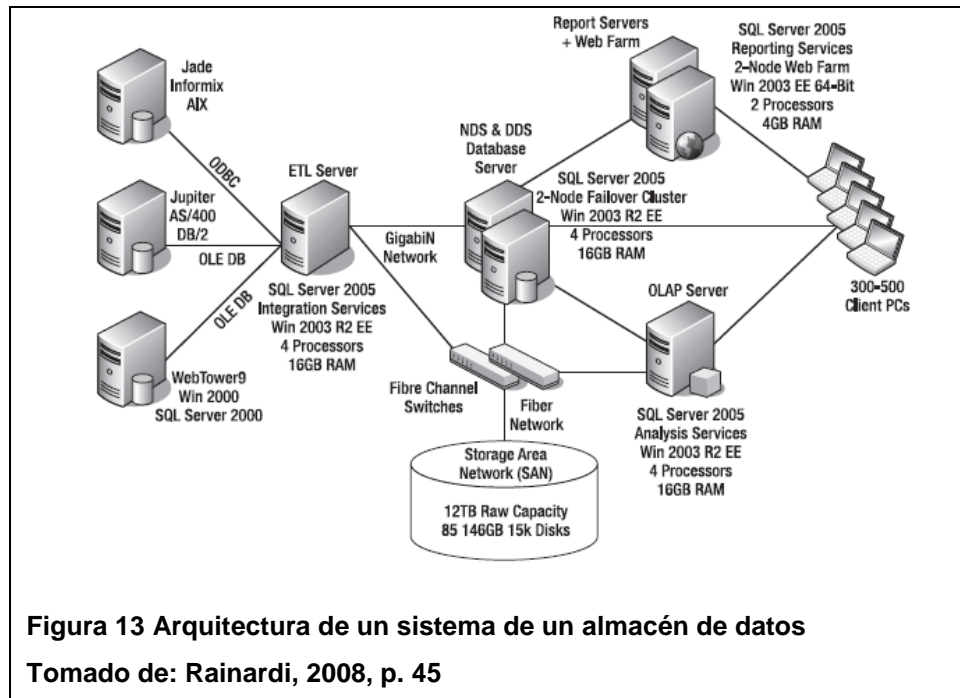
En la siguiente figura se observa el almacén de datos federado. (Rainardi, 2008, p. 39-40)



1.6.4.2 Arquitectura del sistema

Establece la estructura física y las conexiones entre los servidores, redes, software, sistema de almacenamiento, y los clientes. El diseño de una arquitectura de sistema requiere un conocimiento sobre el hardware que se refiere a los servidores que se van a utilizar en el sistema del almacén de datos. Las redes indican como se van a interconectar los servidores, deben ser seguros y de alto rendimiento. El sistema de almacenamiento indica los discos duros que se van a utilizar, arreglos, configuración y el plan de gestión de respaldos a ejecutar. (Rainardi, 2008, p. 42)

En la siguiente figura se observa la arquitectura del sistema.

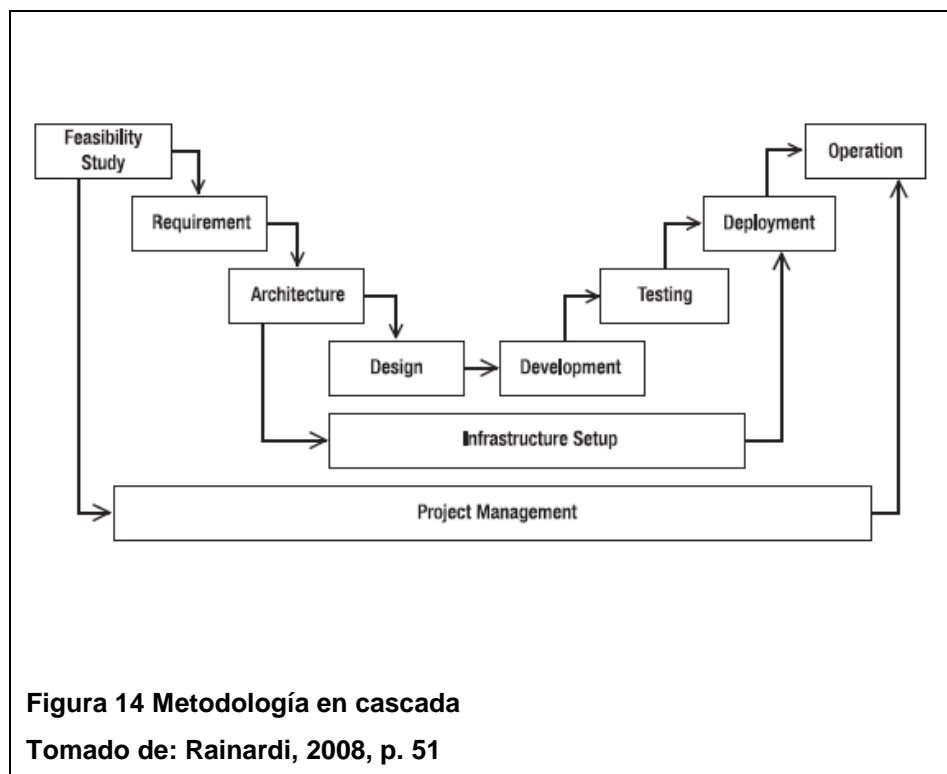


1.7 Metodología de desarrollo de un almacén de datos

Son los procesos que se deben de seguir de forma sistemática de manera que se pueda elaborar un sistema de almacén de datos. Dentro de la ingeniería de *software* se denomina como ciclo de vida de desarrollo de sistemas (*SDLC*). Existen dos metodologías: Cascada e iterativa. (Rainardi, 2008, p. 49)

1.7.1 Metodología en cascada

Se compone de algunos pasos de manera sistemática. En la siguiente figura se observa la metodología en cascada. (Rainardi, 2008, p. 49)



1.7.1.1 Estudio de factibilidad

En esta fase se recopilan los requerimientos en un alto nivel, se determina por qué es necesario un almacén de datos y si ésta es la solución adecuada, se visualiza rápidamente si los sistemas de origen contienen los datos necesarios, se obtiene los datos de muestra para evaluar la calidad de los datos, y se escribe una propuesta, algunos personas prefieren llamar a este documento el caso de negocio. En la propuesta es importante mencionar los beneficios, cuánto tiempo va a tomar el proyecto, cuánto va a costar, y las justificaciones, también se debe colocar los requisitos, la organización del proyecto, el cual incluye los recursos externos como consultores y contratistas, y el plan del proyecto, en donde es muy importante saber quién va a estar realizando qué actividad y en qué momento. (Rainardi, 2008, p. 51)

1.7.1.2 Requerimientos

En esta fase es necesario reunirse con los usuarios para entender las reglas del negocio, el significado de los datos, la interfaz de usuario, los requerimientos funcionales y no funcionales, y demás temas importantes para desarrollar el proyecto. Es necesario siempre documentar estas reuniones. (Rainardi, 2008, p. 52)

1.7.1.3 Arquitectura

En esta fase se debe establecer la arquitectura de flujo de datos y la arquitectura del sistema que se van a utilizar para construir el sistema de almacén de datos. Este paso es crítico debido a que si la arquitectura esta incorrecta, es posible que se deba rediseñar todo desde cero o reiniciar el proyecto desde el principio. (Rainardi, 2008, p. 52)

1.7.1.4 Configuración de la infraestructura

En esta fase es importante preparar el ambiente de producción en donde se va a ejecutar la aplicación. Este paso consiste en la creación de la arquitectura del sistema del almacén de datos, realizar el diseño técnico, la compra, instalación y configuración del software y hardware, conexión de redes, pruebas y documentación sobre la infraestructura y por último la entrega de todo al grupo de operaciones. (Rainardi, 2008, p. 53)

1.7.1.5 Diseño

En esta fase se debe modelar tres piezas principales del sistema del almacén de datos.

a) Los *data stores*

b) Los procesos *ETL* y cada aplicación de presentación.

c) Cada sistema de calidad de los datos y la metadata de cada sistema *ETL*.

(Rainardi, 2008, p. 52)

1.7.1.6 Desarrollo

En esta fase es necesario construir lo que se modeló y diseñó anteriormente,

a + b + c se pueden construir en paralelo. Lo más importante al construirlos en paralelo es definir las interfaces precisas entre las partes. (Rainardi, 2008, p. 52)

1.7.1.7 Pruebas

En esta fase se pone a prueba los *Data Stores*, cada sistema *ETL* y los programas. Este es el paso más crítico de todo el proyecto, debido a que ésta es la primera vez que todos los componentes trabajan en conjunto y la primera vez que toda la arquitectura funciona como un sistema. (Rainardi, 2008, p. 52)

1.7.1.8 Despliegue

En esta fase una vez finalizado el paso anterior, se ejecutan todos los componentes en producción: Los *Data Stores*, cada sistema *ETL* y las aplicaciones de usuario final. Se realiza la carga inicial de los datos desde los sistemas de origen, luego se realiza pruebas en el entorno de producción para asegurarse de que el almacén de datos se cargue correctamente y las aplicaciones de usuario muestren las cifras correctas. Adicionalmente se producen las guías de: Usuario, operación y de solución de problemas para el equipo de operaciones. Se entrena a cada usuario y al grupo de operaciones. Es importante facilitar soporte al sistema, a los usuarios y al equipo de operaciones. Por último se entrega el sistema al equipo de operaciones y se da por finalizado el proyecto. (Rainardi, 2008, p. 53)

1.7.1.9 Operación

En esta fase los usuarios continúan utilizando el almacén de datos y la aplicación. El equipo de operaciones continúa administrando el almacén de datos y da soporte a los usuarios. A menudo los usuarios van a proponer solicitudes de mejora, en donde el equipo de operaciones pasará estas

solicitudes al equipo de desarrollo, se colocan en orden de importancia y luego se rechazan o se les incluye en la próxima versión. (Rainardi, 2008, p. 53)

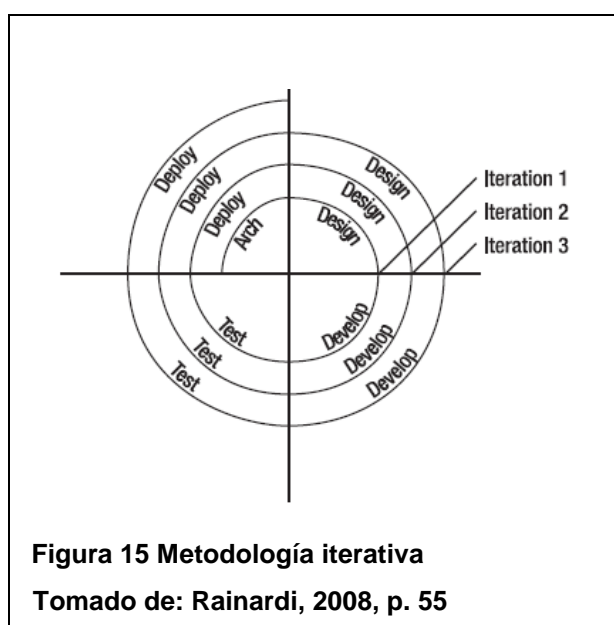
1.7.1.10 Administración del proyecto

En esta fase esto se da cuando se mantiene el plan del proyecto, lo cual significa que en cualquier momento se pueda conocer el estado de cada tarea, quien la realiza y cuando se va a llevar a cabo. Además de mantener una buena comunicación con todas las partes involucradas como los accionistas, los auspiciantes, los usuarios y los desarrolladores del proyecto. Es necesario detectar y atacar los posibles riesgos antes que se conviertan en problemas. (Rainardi, 2008, p. 53)

1.7.2 Metodología iterativa

Consiste en liberar porciones cada vez más grandes del proyecto completo con el objetivo de detectar dificultades a tiempo y poder realizar los arreglos necesarios. Es liberada una versión del sistema del almacén de datos varias veces en producción para ser utilizado por los usuarios. Se construye el almacén de datos horizontalmente en piezas. En cada iteración se realiza el diseño, desarrollo, pruebas e implementación y que los usuarios lo utilicen.

En la siguiente figura se observa la metodología iterativa.



Los pasos de diseño, desarrollo, pruebas y despliegue son similares a la metodología en cascada.

1.8 La plataforma de inteligencia de negocios de *Pentaho*

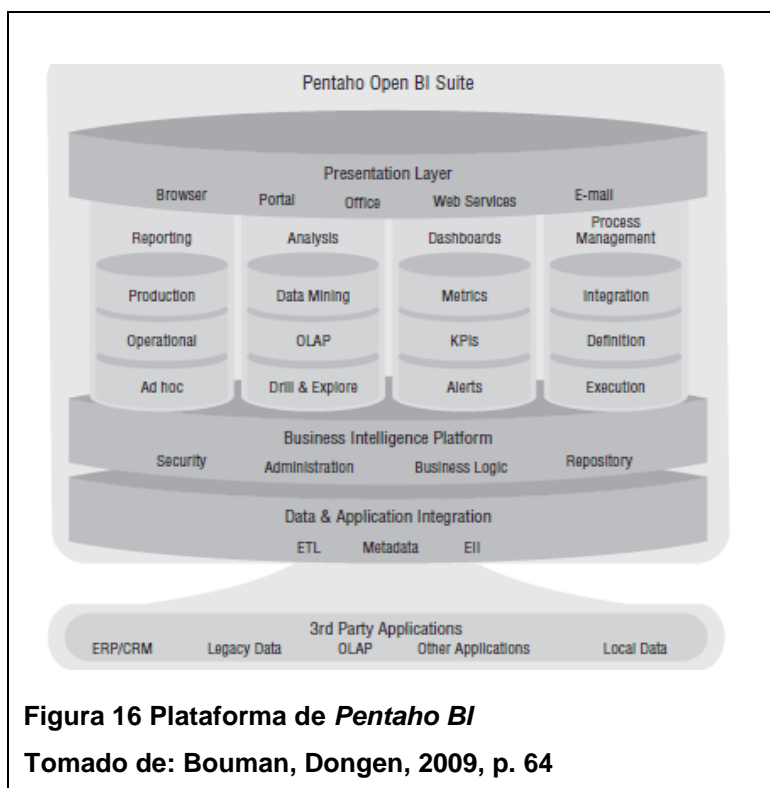
1.8.1 Definición de *Pentaho*

Es un conjunto de aplicaciones utilizadas para crear y entregar soluciones para la toma de decisiones gerenciales, es orientada a la solución y centrado en los procesos, cuando se habla de orientada a la solución es debido a que por medio de *Pentaho* se desarrolla o implementa una solución, en cambio cuando se habla de centrado en los procesos se debe a que cada componente de *Pentaho* tiene un motor de procesos en su núcleo que básicamente soporta entradas y salidas. Al mismo tiempo cada proceso de *Pentaho* puede ejecutarse dentro de otros procesos, lo que permite llevar a cabo tareas muy complejas y brinda una gran flexibilidad. (Mattío, Bernabeu, 2013)

1.8.2 Componentes de *Pentaho*

Pentaho está conformado por varios componentes independientes que ejecutan funciones específicas como: Envío de correo, visualización de gráficos, entre otros, funciona a través del uso combinado de estos componentes.

En la siguiente figura se puede observar cómo los componentes de *Pentaho* se relacionan entre sí.



En la figura 1.16 se pueden observar todos los componentes de la plataforma *Pentaho*, con esta estructura se pueden implementar la fábrica de información corporativa o la multidimensional. La integración de los datos realiza el proceso de obtención de datos de entrada por medio de las herramientas *ETL*. El segundo componente maneja las soluciones del repositorio y de los motores, brindando seguridad para el manejo de la autenticación de los usuarios, calendarización de las tareas, y servicios de correo electrónico. El tercero se denomina componentes de inteligencia de negocios, y está compuesto por reportes, análisis, tableros de mando y gestión de procesos. Aquí es donde se crean los reportes a demanda, y se realiza el análisis y las consultas. El último componente se denomina capa de presentación, el cual permite al usuario final visualizar la información por medio del navegador, portal, correo electrónico y servicios en el internet.

A nivel funcional, el servidor *Pentaho* se puede dividir en tres capas:

- ✓ La plataforma.
- ✓ Componentes de inteligencia de negocios.
- ✓ La capa de presentación.

1.8.2.1 La plataforma

La funcionalidad que ofrecen estos servicios es relativamente de bajo nivel y constituyen la infraestructura básica de la plataforma de la inteligencia de negocios.

Esta capa ofrece los siguientes servicios:

- **Repositorio y motor de soluciones.**

La plataforma de *Pentaho* organiza el contenido de *BI* en las denominadas soluciones. Se puede ver a una solución de *Pentaho* como un sistema de archivos de carpetas con todo el contenido de la inteligencia de negocios para resolver algún problema del negocio, también puede contener carpetas y elementos llamados secuencias de acción (AS). Las carpetas simplemente sirven para proporcionar el contenido de la inteligencia de negocios a una empresa, además de contener otras carpetas y secuencias de acción.

- **Administración de la conexión *pool* a la base de datos.**

En la mayoría de los casos, los datos presentados en las aplicaciones de inteligencia de negocios se almacenan en una base de datos relacional. Con el fin de acceder a los datos de la base de datos, las aplicaciones de *BI* necesitan establecer una conexión a la base de datos. Esta conexión se utiliza para enviar consultas al servidor de la base de datos, el cual envía de vuelta los datos como respuesta.

- **Autenticación de usuarios y servicios de autorización.**

La plataforma *Pentaho* utiliza seguridad *Spring* para manejar la autenticación y autorización de usuarios. Esta es la solución de seguridad estándar del sistema *Spring* de *Java*. La seguridad *Spring* proporciona muchos componentes para implementar diferentes esquemas de autenticación.

- **Programación de tareas.**

La plataforma *Pentaho* utiliza *Quartz* como componente para la programación de tareas. *Quartz* es creada por el proyecto *OpenSymphony* y liberado bajo una licencia Apache 2.0.

El programador de tareas se utiliza para una serie de cosas como la ejecución periódica de las tareas de mantenimiento, la ejecución en segundo plano de los reportes, y la programación de los trabajos *ETL*.

- **Servicios de correo electrónico.**

La plataforma de la inteligencia de negocios tiene la capacidad de enviar correos electrónicos mediante el uso de un servidor estándar por medio de un protocolo de transferencia de correo simple (*SMTP*). Primero se debe de configurar el servidor, antes que el correo pueda ser enviado, la configuración del correo se debe introducir en el archivo `email_config.xml`, el cual se encuentra ubicado en el directorio `<install-path>/pentaho-solutions/system/smtp-email`.

1.8.2.2 Componentes de inteligencia de negocios

Esta capa ofrece los siguientes componentes:

- **Capa de metadatos**

La función de la capa de metadatos *Pentaho* (*PML*) es para proteger a los usuarios finales de la complejidad del código *SQL* y de las bases de datos.

- **Servicio de reportes a demanda**

El servicio de consultas en el internet a demanda y de reportes (WAQR), brinda a los usuarios finales una manera fácil de crear reportes utilizando la capa de los metadatos.

- **Motor ETL**

El motor *ETL* de *Pentaho* realiza las tareas de integración de los datos y ejecuta los trabajos y las transformaciones creadas con las herramientas de integración de los datos *Pentaho*.

- **Motor de reportes**

La plataforma *Pentaho* tiene varios motores de reportes. Los motores nativos son el de la herramienta de consultas *ad hoc*, y del motor *JFreeReport*. Adicionalmente, *Pentaho* brinda soporte para *JasperReports* y *BIRT* ya incorporados.

- **Motor de procesamiento analítico en línea**

Mondrian es el motor de procesamiento analítico en línea de *Pentaho* y traduce las consultas *MDX* en *SQL* basado en un modelo multidimensional, también se encarga de receptor y almacenar los resultados intermedios y anteriores para optimizar el rendimiento. Esto significa que la primera vez que un análisis se ejecuta en un modelo multidimensional, se tardará más tiempo que el análisis posterior durante la misma sesión porque *Mondrian* trata de mantener los resultados anteriores, jerarquías, y los cálculos en la memoria.

- **Motor de minería de datos**

Se compone de una amplia colección de algoritmos de minería de datos, tales como los que se necesitan para la agrupación, árboles de decisión, regresión y de redes neuronales.

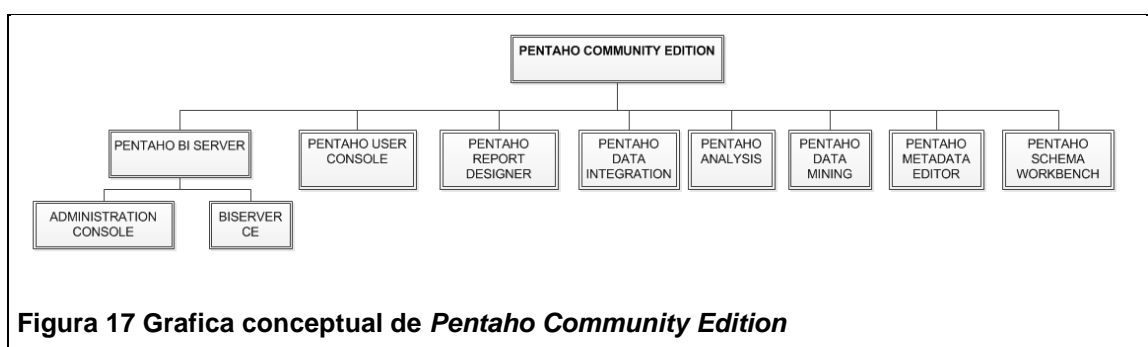
1.8.2.3 La capa de presentación

Pentaho tiene una interface de internet integrada llamado la consola de usuario de *Pentaho* (*PUC*). Esta consola permite al usuario interactuar con el servidor.

La capa de presentación se puede utilizar para navegar y abrir el contenido existente como reportes, cuadros de mando, y análisis pero adicionalmente se puede utilizar para crear nuevo contenido de inteligencia de negocios.

1.8.3 Principales herramientas de *Pentaho*

La plataforma de *Pentaho* se compone de todas las herramientas que se necesitan para implementar una arquitectura de inteligencia de negocios. A continuación se indica cada herramienta de la plataforma: (Mattío, Bernabeu, 2013)



- ***Pentaho BI Server***

Es una aplicación de internet, una de sus funciones principales es permitir al usuario acceder e interactuar con el contenido de inteligencia de negocios.

- ***Pentaho User console (PUC)***

Es una aplicación web a través de la cual el usuario puede navegar e interactuar con los componentes del negocio tales como reportes, cuadros de mando, análisis, etc., y también administrar y configurar el servidor de inteligencia de negocios.

- ***Pentaho Report Designer (PRD)***

Esta es la herramienta utilizada para crear reportes avanzados y de alta complejidad.

- **Pentaho Data Integration (PDI)**

Está compuesto por varios componentes, los cuales ayudan a cada proceso *ETL*. El diseñador de *Spoon*, tiene una interface gráfica, adicionalmente está compuesta por las siguientes áreas de trabajo: (Roldan, 2013, p. 243)

- ✓ Menú principal: Contiene las opciones archivo, editar, vista previa, acción, herramientas, y ayuda.
- ✓ Vista de diseño: Contiene todos los pasos de la herramienta, en el cual se diseñan las transformaciones y los trabajos.
- ✓ Barra de herramientas: Contiene los iconos para las transformaciones y los trabajos.
- ✓ Área de trabajo: Lugar de trabajo en donde se realizan las transformaciones y los trabajos.

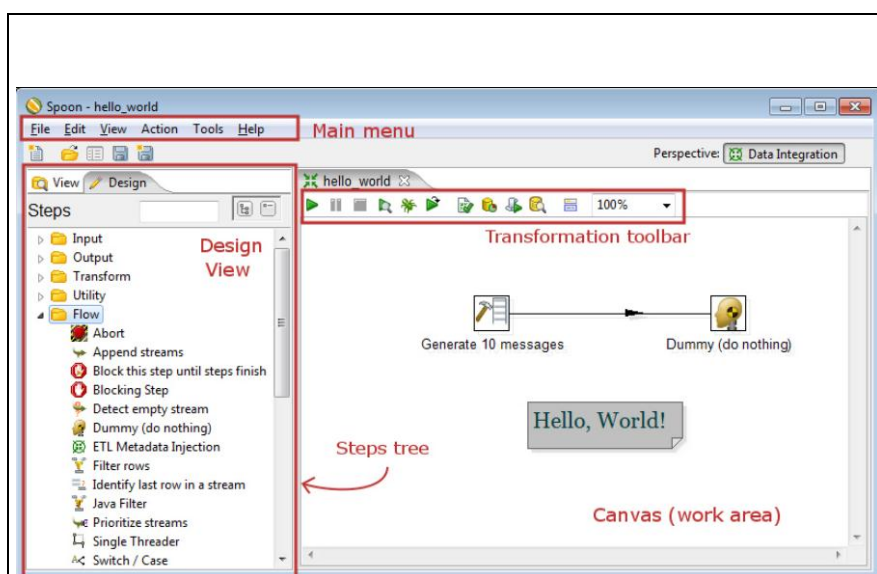


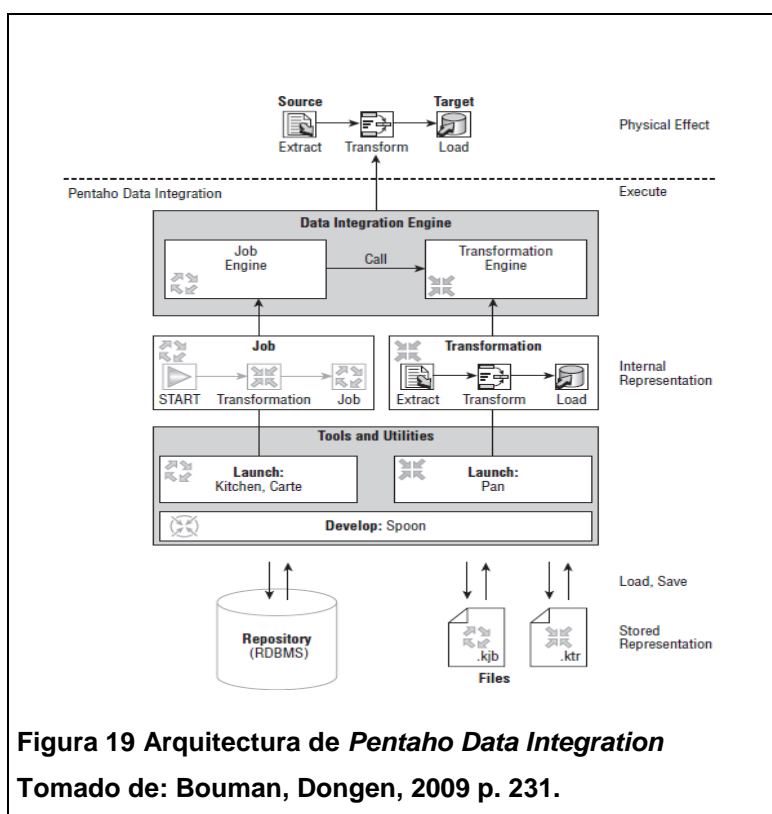
Figura 18 Interface gráfica Spoon

Tomado de: Herramienta Spoon de *Pentaho Data Integration*, 2014

➤ **Arquitectura de *Pentaho Data Integration***

- Interface gráfica.
- Trabaja con cada tecnología como: Java Script, Java.
- Se puede trabajar con varias plataformas como: Linux, Mac y Windows.

- Basado en dos tipos de objetos: Transformaciones (colección de pasos en un proceso *ETL*) y trabajos (colección de transformaciones).
- Incluye tres herramientas principales:
 - ✓ *Spoon*: Una herramienta gráfica de integración de datos para la creación de transformaciones y trabajos.
 - ✓ *Kitchen*: Una herramienta de línea de comandos para ejecutar trabajos.
 - ✓ *Pan*: Una herramienta de línea de comandos para ejecutar transformaciones. (Bouman, Dongen, 2009 p. 230)



- ***Pentaho Analysis (Mondrian OLAP Server)***

Es un motor multidimensional que resuelve análisis interactivo o consultas de análisis OLAP.

- ***Pentaho Data Mining (Weka)***

Es una herramienta de minería de datos.

- ***Pentaho Metadata Editor (PME)***

Es una herramienta que permite crear, editar, y publicar reportes ad hoc.

- ***Pentaho Schema Workbench (PSW)***

Es una herramienta que se utiliza para crear, editar, y publicar ya sea modelos multidimensionales o esquemas *Mondrian*.

- ***Saiku***

Es un visor *OLAP*, con una interface de usuario muy agradable.

- ***CTools***

Es un conjunto de herramientas utilizado principalmente para la creación de cuadros de mando.

1.8.4 Principales características de *Pentaho*

- ✓ Está completamente programado en *Java*. Esta característica hace que el software sea escalable, integrable y portable.
- ✓ Puede ejecutarse en cualquier servidor de aplicaciones compatible con arquitecturas *Java Enterprise Edition (JEE)*.
- ✓ Tiene acceso a la base de datos a través de controladores *JDBC*.
- ✓ Puede ser ejecutado en varios sistemas operativos, debido a que está programado en *Java*. Mientras se utilice una máquina virtual *Java (JVM)* para la plataforma. (Mattío, Bernabeu, 2013)

Capítulo 2

2 Diseño e implementación del *data mart* del módulo de ventas de Claro por medio de la herramienta *Pentaho*

2.1 Situación actual

El sector de telecomunicaciones en los últimos años ha tenido un crecimiento exponencial, uno de los objetivos principales de las empresas es el posicionamiento en el mercado por medio de las ventas de cada uno de sus productos y la contratación de los servicios.

Actualmente, un elemento importante cuando se va a tomar cada decisión, se debe a la administración correcta y oportuna de los datos. En el medio se puede encontrar aplicaciones, las cuales ayudan a la ejecución de soluciones de inteligencia de negocios, el problema es que no muchas empresas las implementan.

Una herramienta de inteligencia de negocios se puede implementar en toda empresa o lugar de trabajo en donde se requiera centralizar la información para su posterior análisis, el usuario puede acceder a los reportes desde la intranet en cualquier parte de la empresa. Cada vez más las áreas gerenciales requieren de un acceso más rápido a los datos para poder tomar cada decisión.

Hoy en día el departamento de Procesos y Calidad de Claro, no cuenta con una herramienta de inteligencia de negocios, por lo que se dificulta la toma de decisiones inmediatas, motivo por el cual se crea la necesidad de la construcción de un *data mart* que almacene datos históricos, que entregue la información de manera oportuna y principalmente ayude a la toma de decisiones a las áreas involucradas. Luego de la construcción del *data mart* enfocado al módulo de ventas de clientes actuales de la empresa en donde se va a realizar algunas tareas de análisis para poder detectar ventas potenciales como por ejemplo a un cliente que tiene un plan de celular solo minutos, se le puede ofrecer el servicio de datos que incluye internet, navegación, o a un

cliente que tiene un plan fijo de internet en su casa, se le puede ofrecer el servicio de televisión satelital. Existen varias oportunidades de ventas nuevas o de adquisición de servicios adicionales.

El *data mart* de ventas que va a ser implementado en este trabajo de tesis va a utilizar una metodología en cascada, la cual fue descrita anteriormente.

2.1.1 Definición de cada objetivo

- Identificar las necesidades del área de ventas de la empresa Claro, para un correcto análisis de requerimientos.
- Realizar un análisis para la selección de los equipos que se utilizarían en la implementación y configuración del servidor *Pentaho*.
- Elaborar un diseño dimensional para el *data mart* de ventas
- Implementar una solución de inteligencia de negocios para el módulo de ventas de la empresa Claro.
- Realizar un análisis de costo - beneficio del proyecto.

2.1.2 Delimitación del proyecto

En el presente proyecto se va a aplicar inteligencia de negocios con la herramienta *Pentaho*, para el análisis de los requerimientos del módulo de ventas de la empresa Claro, con lo cual se va a construir un *data mart* que brinde soporte cuando el área gerencial necesite tomar decisiones importantes. Es importante seleccionar la metodología de desarrollo en cascada para la realización del proyecto, debido a que propone un enfoque ordenado y secuencial. De esta forma se obtendrá cada reporte y cada tablero para que las personas encargadas realicen el respectivo análisis.

Es importante mencionar que la implementación se va a realizar de manera local en todas las agencias de Quito.

2.1.3 Roles establecidos

- Patrocinador: Claro Ecuador
- Administrador de base de datos (*DBA*): Ing. Sistemas *dataware house*
- Analistas de Calidad y Procesos: Analista de información y analista de calidad.
- Usuarios finales: Director, gerente y jefes de los centros de atención a clientes de Quito.

2.1.4 Uso de metodología

Se va a realizar la implementación de una solución de inteligencia de negocios para el módulo de ventas de Claro con el uso de la herramienta *Pentaho Community Edition (PCE)*. Se va a desarrollar un reporte estándar con la herramienta *Pentaho Report Designer (PRD)*. Adicionalmente se va a implementar un cubo de procesamiento analítico en línea, en donde se va a realizar las distintas combinaciones de análisis para entregar la información requerida por cada usuario final, además se construye un tablero de mando integral.

Se va a utilizar la metodología de Ralph Kimball descrito en el capítulo I debido a que se va a construir un *data mart* independiente a partir de los sistemas fuente y siguiendo un alcance por fases, el cual entrega una constante retroalimentación de las reglas establecidas, define el tamaño del proyecto, las correcciones son desarrolladas en periodos establecidos, adicionalmente el tiempo de implementación es corto.

2.1.5 Planificación

En este subcapítulo se definen, planifican y explican las actividades que se van a ejecutar por fases en el diseño del *data mart* de ventas y las herramientas de inteligencia de negocios a utilizar.

2.1.6 Recursos a utilizar: *Hardware y software*

Para seleccionar tanto el *hardware* como el *software*, se tomó en cuenta las recomendaciones del fabricante desde la página oficial de *Pentaho*. (Pentaho, 2014)

Las herramientas de *Pentaho BI* suite y el servidor de base de datos *PostgreSQL* v9.2 se instalaron en una computadora personal, la cual cumple con los requerimientos para este proyecto de tesis.

2.1.6.1 *Hardware*

Tabla 2 Características del hardware para el *data mart* de ventas

Características	Detalle
Marca	Dell
Modelo	OptiPlex 3010
Procesador	Intel(R) Core(TM) i3-3200 CPU @ 3.30 GHz
Disco duro	500 Gb
Memoria Ram	4 Gb

2.1.6.2 Software

Tabla 3 Características del software para el *data mart* de ventas

Características	Detalle
Sistema operativo	Microsoft Windows 7 Enterprise Edition
Java	Java Development Kit v1.7.0._45
Herramienta para modelamiento de datos	SQL Power Architect v1.0.7
Servidor de base de datos	PostgreSQL versión 9.2 para Windows 32 bits
Herramientas <i>BI</i>	Pentaho BI suite Community Edition v4.8.0 Pentaho Data Integration v4.4.0 Pentaho Report Designer v.3.9.1 Pentaho Schema Workbench v3.4.1.1 Community Dashboard Framework v4.8.0
Navegador	Mozilla Firefox v32

2.1.7 Análisis de requerimientos

En las reuniones con las personas involucradas y los usuarios finales se documentó y definió los requerimientos para establecer que información necesitan obtener para su respectivo análisis.

2.1.8 Reunión o entrevista con las personas involucradas

En las entrevistas con cada usuario final, se definió que información requieren revisar frecuentemente.

Los usuarios finales son:

- ✓ La directora de servicio al cliente región sierra Javiera Robles.
- ✓ El gerente de servicio al cliente región sierra Juan Manuel Ossa.
- ✓ Los jefes de los centros de atención a clientes de Quito.

La información que los usuarios finales requieren revisar frecuentemente son las ventas generadas por los centros de atención a clientes de la región sierra, como se mencionó anteriormente para este proyecto de tesis se va a delimitar el análisis a los centros de atención a clientes de Quito.

También se entrevistó al analista de información, en donde él detallo todo el proceso de la generación de los reportes de ventas.

2.2 La Empresa de Telecomunicaciones

2.2.1 Definición

Una empresa de telecomunicaciones presta servicios como: Telefonía celular, internet móvil y fijo y televisión por cable. Uno de los principales objetivos es que los usuarios tengan acceso a productos y servicios a precios asequibles y que se ajusten a sus necesidades. En Claro se hace énfasis tanto en la satisfacción del cliente como en la atención personalizada.

2.3 Funcionamiento de una empresa de telecomunicaciones

2.3.1 Organigrama de la Empresa Claro

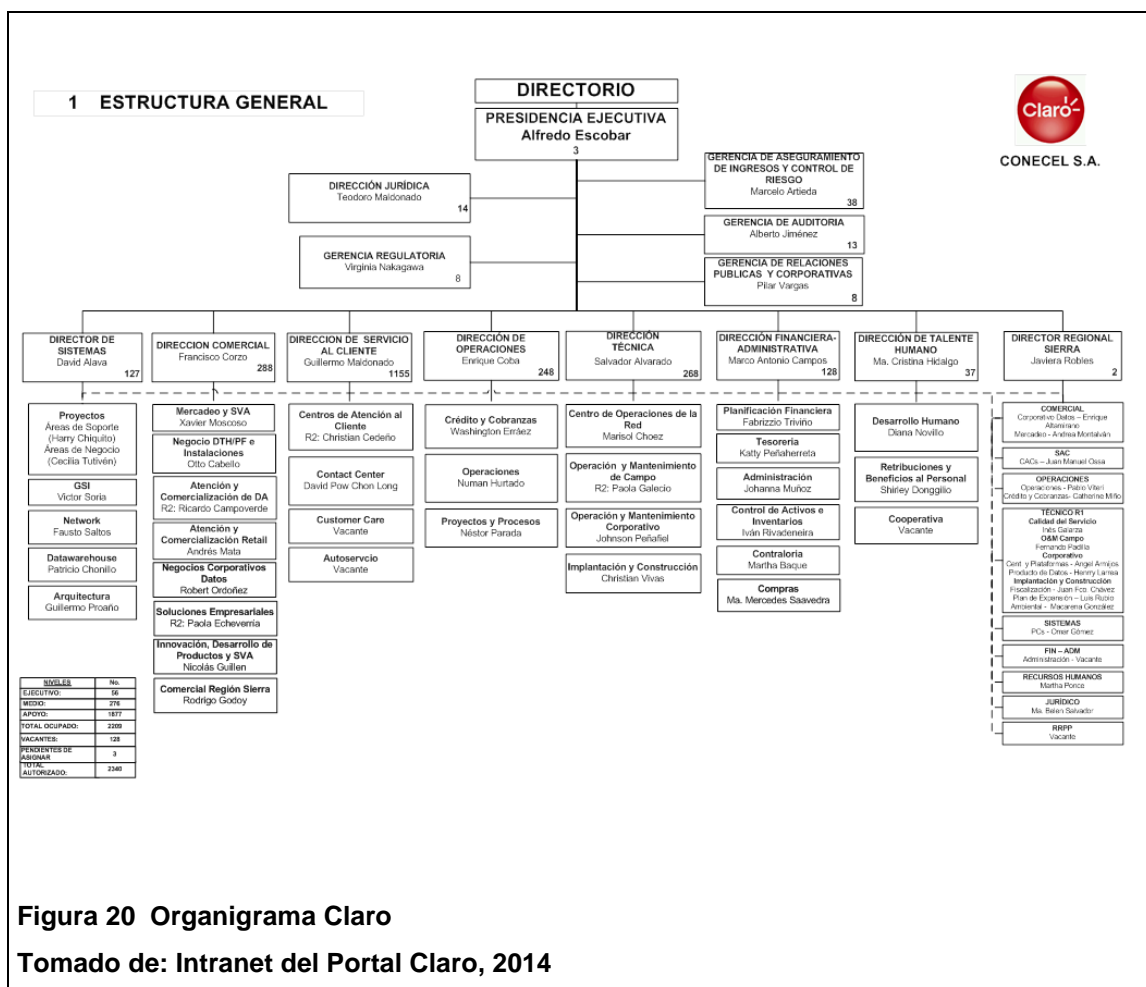


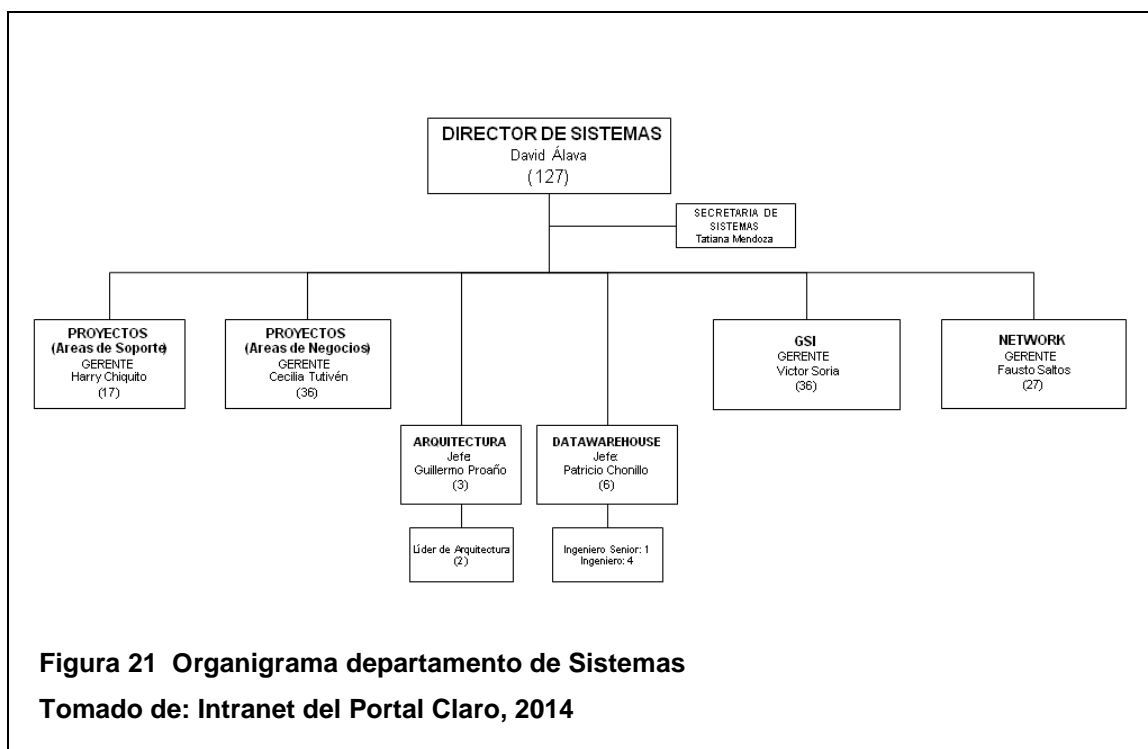
Figura 20 Organigrama Claro

Tomado de: Intranet del Portal Claro, 2014

2.3.2 Área de Sistemas

El área de Sistemas está conformada por ciento veinte seis personas, de los cuales nueve están a cargo de Región 1, el departamento se llama Pc's Nac Redes / Servidores / Seguridades R1. El departamento de *data warehouse*, está conformado por seis personas, los cuales están ubicados en Guayaquil.

En la figura 2.36 se observa el organigrama del departamento de sistemas de Claro.



2.3.3 Departamento de Procesos y Calidad

El departamento de Procesos y Calidad, se encuentra dentro del área de Servicio al Cliente, está conformado por once analistas, los cuales están ubicados diez en Guayaquil y uno en Quito. Cuando la directora, el gerente o los jefes de la empresa requieren información sobre las ventas generadas en un período específico es necesario que lo soliciten al área de Procesos y Calidad, este proceso puede tomar desde un día y medio hasta 2 días laborables, dependiendo de la carga de trabajo y solicitudes que le realicen al analista de información.

2.3.4 Funciones del área Comercial, y su importancia

La función principal del área es la venta de productos y servicios, estas ventas se generan por medio de los centros de atención a clientes, distribuidores y de los asesores que cuentan con cartera de clientes empresariales (ases). Otras funciones son elaborar promociones, establecer precios, investigación de

mercado, generar pronósticos de ventas, control de inventario, abastecimiento de stock en los puntos de ventas de productos.

El objetivo de cualquier negocio es obtener una rentabilidad, el área de ventas de toda empresa siempre debe de innovar su portafolio de productos y servicios para poder competir en el mercado.

2.3.5 Problema

En la actualidad existen algunos problemas en cuanto a la recolección, administración, y análisis de los datos como por ejemplo no existe el acceso a la información de manera oportuna, existen procesos manuales, la información no se encuentra consolidada, y la toma de decisiones a destiempo.

El analista de información envía todos los días Lunes y solo una vez por semana el reporte de ventas de los centros de atención a clientes. Hoy en día para poder revisar en detalle las ventas generadas por los centros de atención a clientes de Quito, los usuarios finales deben solicitar al analista de información los datos, él tiene acceso a los sistemas fuente y genera la información solicitada en hojas de cálculo, el problema principal es que para levantar la información de ventas es necesario realizar un proceso manual en donde una vez generado el reporte se pueda visualizar las ventas por cada centro de atención a clientes de Quito, por asesor y por producto. Otro problema es que el proceso actual no permite tomar decisiones inmediatas, debido a que no se cuenta con una herramienta de inteligencia de negocios para las ventas. Dependiendo del volumen de peticiones de información, los tiempos de entrega se incrementan, ocasionando que muchas veces se obtenga la información a destiempo.

2.3.6 Solución

Se construyó un *data mart* cuyo propósito es ayudar a la toma de decisiones al director, al gerente, y los jefes de los centros de atención a clientes de Quito, que necesitan establecer un control y monitoreo sobre las ventas diarias de manera detallada.

Se construyó cada proceso de extracción, transformación y carga para extraer los datos de los sistemas fuente y cargarlos al *data mart* de análisis de ventas.

Se construyó un reporte, un cubo *OLAP* y un tablero de mando como herramientas de entrega de información y análisis de inteligencia de negocios. De ésta manera los usuarios pueden visualizar el reporte y realizar actividades de análisis *OLAP*. El reporte y cubo *OLAP* tienen funcionalidades para representar los datos de forma gráfica, facilitando el análisis a los usuarios finales.

2.4 Metodología en cascada a seguir

Se utiliza una metodología en cascada para construir el *data mart* de ventas denominado DM_VENTAS, a continuación se detalla cada fase:

a. Estudio de factibilidad

En la actualidad Claro no cuenta con una herramienta que permita visualizar en un solo lugar las ventas generadas por los centros de atención a clientes de Quito, motivo por el cual es necesario la construcción de un *data mart*. Se puede mencionar algunos beneficios que se van a obtener por medio de esta implementación como el acceso a la información consolidada de ventas de manera rápida, además que los datos siempre van a estar actualizados. En el capítulo 3 se va a detallar el costo – beneficio de este proyecto.

b. Requerimientos funcionales y no funcionales

Es necesario definir tanto los requerimientos funcionales como los no funcionales en la construcción de un *data mart*, el objetivo es asegurar que el sistema permita a los usuarios cumplir con los objetivos del negocio. Los requerimientos funcionales definen lo que realiza el sistema, además contienen las características que el *data mart* debe tener. Los requerimientos no funcionales se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar. (Rainardi, 2008, p. 61)

Tabla 4 Requerimientos funcionales

Número	Requerimientos	Prioridad
1	Los usuarios finales necesitan que la información se encuentre disponible en línea, y que por medio de un gráfico se pueda realizar el respectivo análisis sobre el módulo de ventas, el uso del grafico permite visualizar de manera rápida las ventas generadas por cada centro de atención a clientes de Quito.	Alta
2	Los usuarios finales necesitan que la información se encuentre consolidada en un solo lugar. Por ejemplo por medio de la consola de usuario de <i>Pentaho</i> .	Alta
3	Los usuarios finales necesitan analizar las ventas generadas por cada centro de atención al cliente de Quito.	Alta
4	Los usuarios finales necesitan analizar las ventas generadas por cada asesor.	Alta
5	Los usuarios finales necesitan analizar las ventas generadas por cada producto.	Alta
6	Los usuarios finales necesitan analizar la ganancia de ventas generadas por producto.	Alta
7	Los usuarios finales necesitan analizar las ventas generadas por cantidad por cada producto.	Alta
8	Los usuarios del negocio necesitan comparar las ventas generadas del mes del año anterior con el mes del año actual.	Alta
9	Los usuarios del negocio necesitan comparar las ventas generadas entre meses por producto.	Alta
10	Los usuarios del negocio necesitan acceder, visualizar y filtrar los datos en el reporte.	Alta
11	Los usuarios del negocio deben tener habilitado la opción para imprimir reportes y poder exportarlos a formato .pdf/.xls.	Alta

Adaptado de: Rainardi, 2008, p. 63

Tabla 5 Requerimientos no funcionales

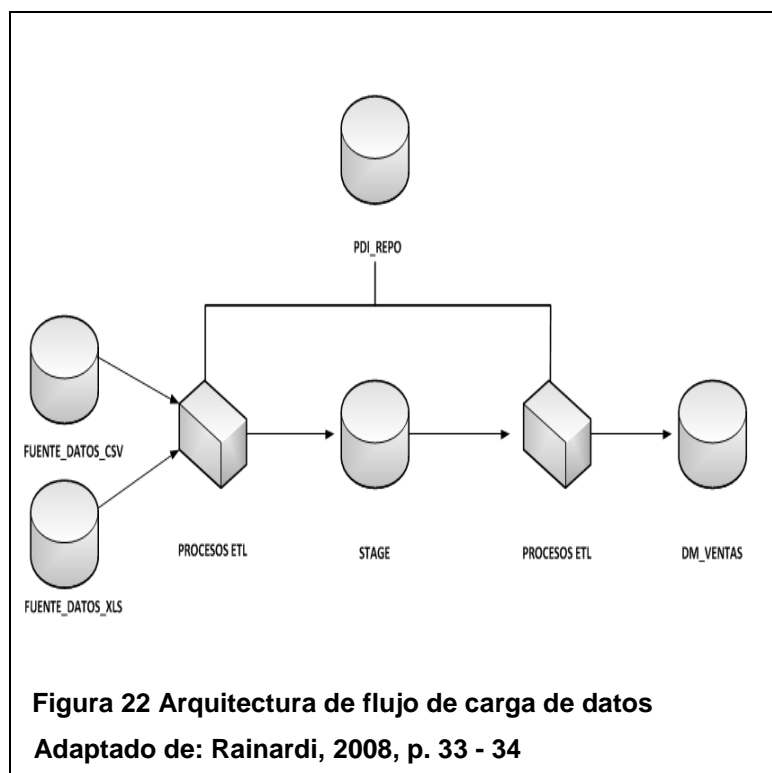
Número	Requerimiento
1	La interface de usuario debe contener las normas y directrices de la empresa.
2	La interface de usuario se va a poder acceder por medio del navegador. Pero su acceso se encuentra limitado a la red interna de la empresa.
3	Los usuarios finales van a acceder a la información por medio de la consola de usuario de <i>Pentaho</i> .
4	Los usuarios finales necesitan que el reporte, el cubo <i>OLAP</i> , y el tablero de mando tengan una interface amigable.
5	Los usuarios del negocio deben tener habilitado la opción para mostrar gráficos en el reporte.
6	La directora, el gerente, y los jefes van a tener acceso a toda la información sobre las ventas de los centros de atención a clientes de Quito
8	El tiempo de respuesta máximo del reporte, del cubo <i>OLAP</i> , o del tablero de mando es de 30 segundos.
9	La especificación estándar mínima para el cliente es Internet Explorer 6 sobre Windows XP que se ejecute en una computadora personal o portátil con Intel Pentium D 820 (o móvil / AMD equivalente) con 512 MB de memoria RAM.
10	El <i>data mart</i> de ventas debe estar disponible durante las 24 horas del día, y 7 días a la semana. El tiempo de inactividad no debe ser mayor a una hora al mes.
14	Los usuarios del negocio necesitan crear reportes dinámicos, sin necesidad de escribir código de lenguaje para realizar consultas.

Adaptado de: Rainardi, 2008, p. 66

c. Arquitectura de flujo de carga de datos y la arquitectura del sistema

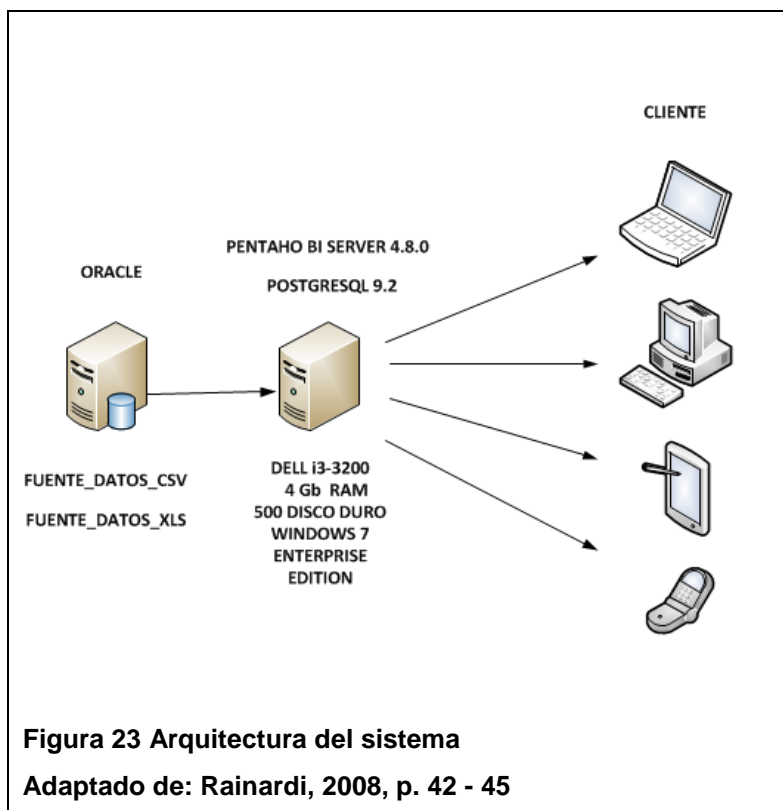
En esta fase se detalla cada proceso de carga de datos del *data mart* DM_VENTAS. Se va a utilizar una arquitectura de flujo de carga de datos simple con un *data store* dimensional denominado DM_VENTAS.

En la siguiente figura se observa la arquitectura de flujo de carga de datos.



En cuanto al diseño de la arquitectura del sistema, se compone de un servidor de base de datos Oracle que almacena los sistemas fuente FUENTE_DATOS_CVS y FUENTE_DATOS_XLS. Una computadora personal Dell OptiPlex 3010 i3-3200 con una base de datos PostgreSQL v9.2 y sistema operativo Microsoft Windows 7 Enterprise Edition 32 bits, en donde se encuentra instalado el Pentaho BI server. Finalmente en el diseño se encuentra los medios por los cuales los usuarios finales van a acceder a la información.

En la siguiente figura se observa la arquitectura del sistema.



Para este proyecto de tesis se trabaja con la información recolectada de la base de datos de ventas, del departamento comercial de la empresa Claro Ecuador. La base de datos se encuentra ubicada físicamente en la matriz Claro en la ciudad de Guayaquil.

La fuente de datos FUENTE_DATOS_CSV se extrae del reporte denominado Reporte de líneas activas en SCT, el tipo de archivo es .csv, los cuales se obtienen del sistema axis de Claro. En el anexo 1 se detalla el diccionario de datos del archivo Reporte de líneas activas en SCT.

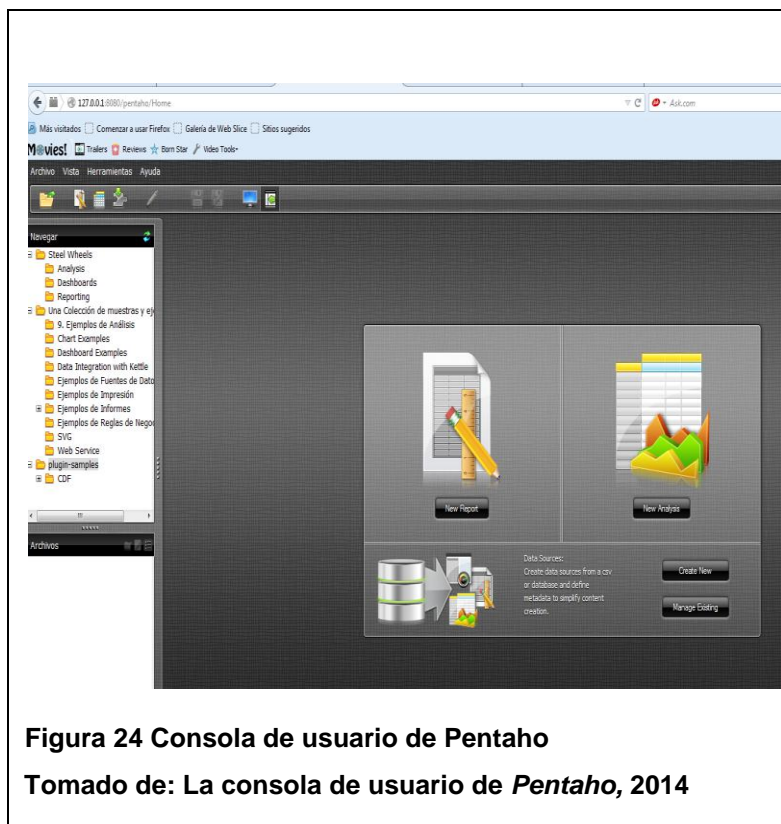
La fuente de datos FUENTE_DATOS_XLS, se extrae del reporte denominado Reporte Detalle de planes y códigos BP, el tipo de archivo es xls, los cuales se obtienen del sistema axis de Claro. En el anexo 2 se detalla el diccionario de datos del archivo Reporte Detalle de planes y códigos BP.

d. Configuración de la infraestructura

Se realizaron los siguientes pasos y la selección de las herramientas se hicieron en base a que son de código libre y cumplen con lo requerido para este proyecto de tesis:

- ✓ Se descargó e instaló el *Java Development Kit (JDK)* v1.7.0_45 desde la página oficial de java y se configuró las variables de entorno JAVA_HOME y Path.
- ✓ Se creó una carpeta denominada *pentaho* en la raíz del disco C: en donde se descomprimió e instaló todas las herramientas descargadas
- ✓ Se descargó, instaló, y configuró la herramienta para modelamiento de datos *SQL Power Architect* v1.0.7.
- ✓ Se descargó, instaló, y configuró la base de datos *PostgreSQL* v9.2 desde la página oficial de PostgreSQL.
- ✓ Se descargó y configuró la plataforma *Pentaho BI Server Community Edition* v4.8.0 desde la página oficial de *Pentaho*, el cual brinda la infraestructura para las soluciones de inteligencia de negocios.
- ✓ Antes de arrancar *Pentaho BI Server* es necesario copiar el driver *postgresql-9.2-1004.jdbc4.jar* en la ruta *C:\pentaho\biserver-ce-4.8.0-stable\biserver-ce\tomcat\lib*.
- ✓ En la carpeta *C:\pentaho\biserver-ce-4.8.0-stable\biserver-ce*, se encuentra ubicado el ejecutable *start-pentaho.bat*. Una vez levantado el servidor *Tomcat* se tiene acceso a la consola de usuario de *Pentaho* ingresando a la dirección: <http://localhost:8080/pentaho/Login>. Es necesario ingresar usuario y contraseña por motivos de seguridad.

En la siguiente figura se observa la consola de usuario de *Pentaho* (PUC).



- ✓ En la carpeta `C:\pentaho\biserver-ce-4.8.0-stable\administration-console`, se encuentra ubicado el ejecutable `start-pac.bat`. Por medio de esta aplicación se va hacer uso de la consola administrativa de *Pentaho* (PAC). La cual sirve para la administración de usuarios, roles y servicios, además de agendar tareas.

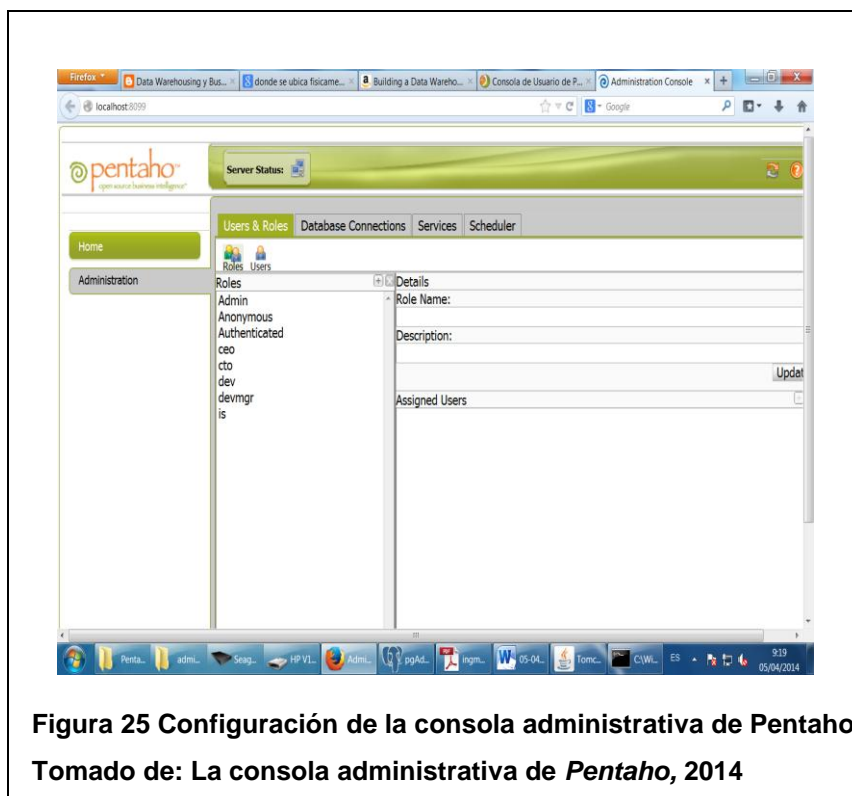


Figura 25 Configuración de la consola administrativa de Pentaho
Tomado de: La consola administrativa de Pentaho, 2014

- ✓ Por medio de la consola administrativa de *Pentaho*, se configuró la conexión al *data mart* DM_VENTAS con el cual se va a trabajar.

En la siguiente figura se observa la conexión a DM_VENTAS.

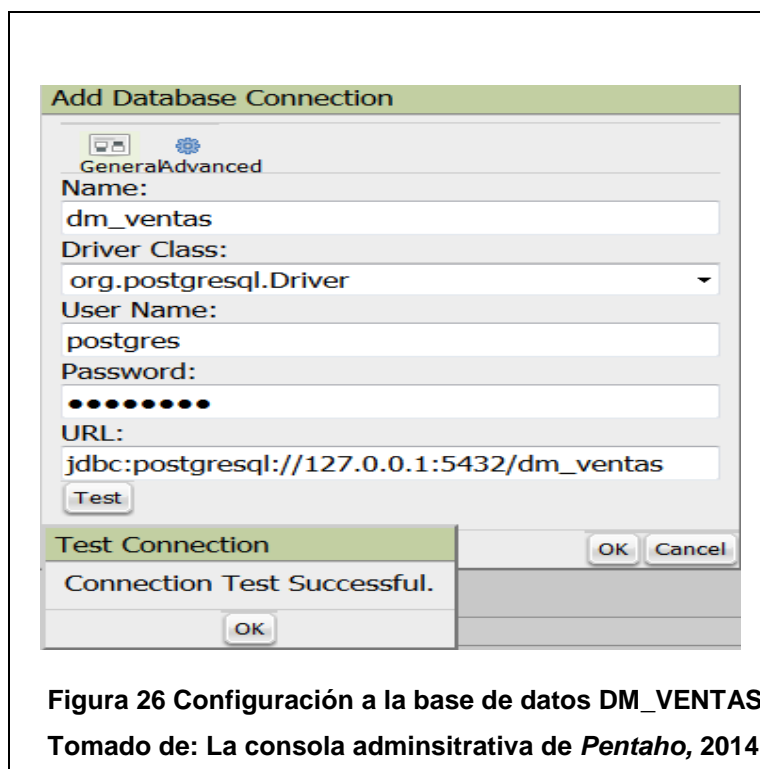


Figura 26 Configuración a la base de datos DM_VENTAS
Tomado de: La consola adminsitrativa de Pentaho, 2014

- ✓ Se descargó y configuró la herramienta *Pentaho Data Integration* v4.4.0 desde la página oficial de *Pentaho*.
- ✓ Se descargó el controlador JDBC postgresql-9.2-1004.jdbc4.jar desde la página oficial de *PostgreSQL* y se copió en la ruta C:\pentaho\pdi-ce-4.4.0\data-integration\libext\JDBC, para establecer la conexión entre *PDI* y la base de datos *PostgreSQL*.
- ✓ En la carpeta C:\pentaho\pdi-ce-4.4.0\data-integration, se encuentra ubicado el ejecutable Spoon.bat, por medio de esta aplicación se realizan todos los procesos *ETL* de forma gráfica.
- ✓ Se configuró la conexión desde SPOON hasta la base de datos PDI_REPO.
- ✓ Se configuró la conexión desde SPOON hasta la base de datos STAGE.
- ✓ Se configuró la conexión desde SPOON hasta la base de datos DM_VENTAS.
- ✓ Se descargó y configuró *Pentaho Report Designer* v3.9.1 desde la página oficial de *Pentaho*.
- ✓ Se descargó y configuró *Pentaho Schema Workbench* v3.4.1.1 desde la página oficial de *Pentaho*.
- ✓ Se descargó y configuró *Community Dashboard Framework* v4.8.0.

e. Diseño del sistema del almacén de datos

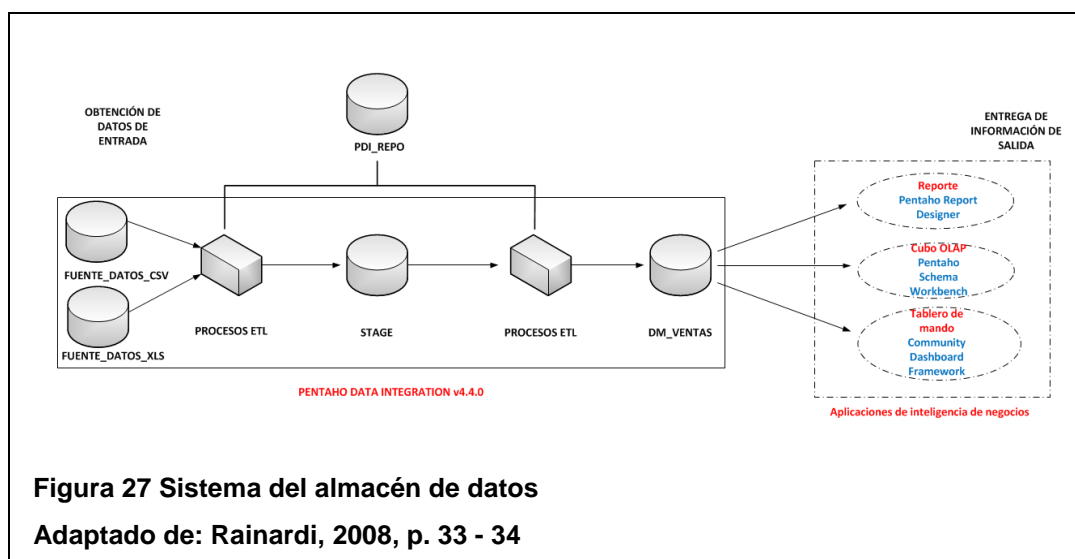


Figura 27 Sistema del almacén de datos

Adaptado de: Rainardi, 2008, p. 33 - 34

En los sistemas fuente FUENTE_DATOS_CSV y FUENTE_DATOS_XLS están ubicados los archivos tipo csv y xls. Se utiliza como fuente de datos los archivos tipo .csv y xls, los cuales contienen información desde el 01-08-2013 como fecha inicial hasta el 31-08-2014 como fecha final, las fechas mencionadas fueron seleccionadas como referencia para este proyecto de tesis. Se va a extraer los datos desde los sistemas fuente FUENTE_DATOS_CSV Y FUENTE_DATOS_XLS hacia la base de datos *STAGE*, esta base de datos es temporal y sirve para normalizar, limpiar, y consolidar los datos, además de la creación de claves subrogadas y de la creación de tablas temporales antes de la carga de datos al *data mart* DM_VENTAS. La definición de los procesos de carga y los metadatos manejados por la herramienta *Pentaho Data Integration* se almacenarán en la base de datos PDI_REPO.

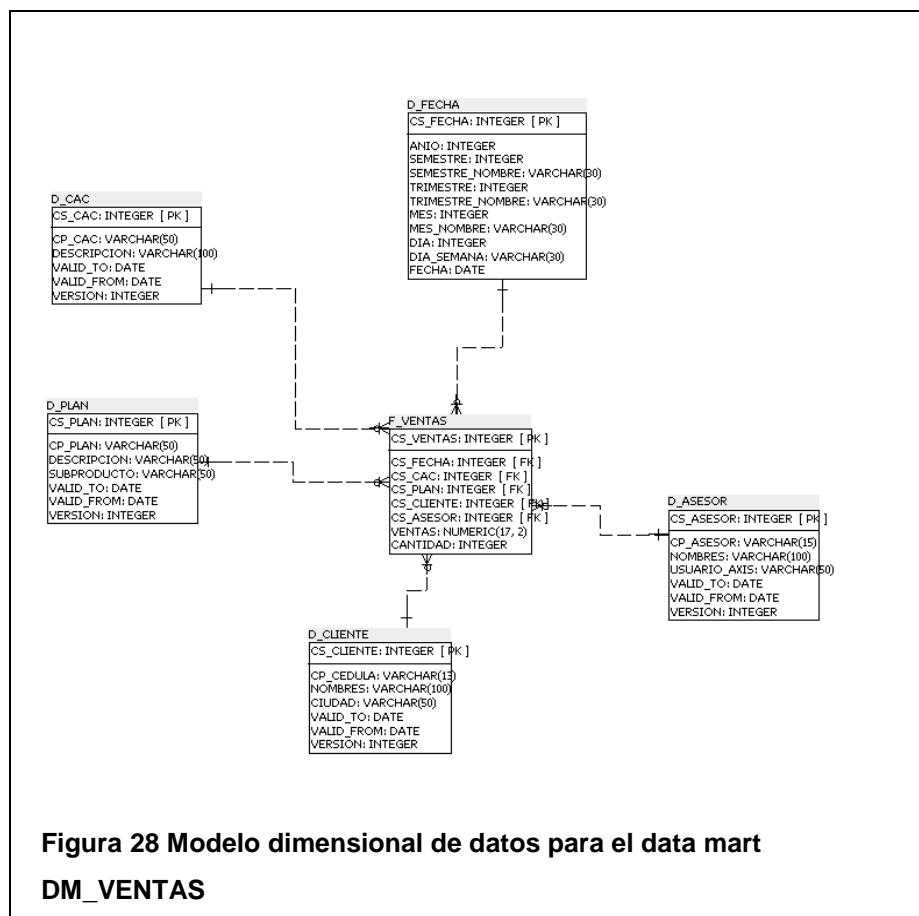
En el capítulo anterior se mencionó que se va a utilizar la plataforma de inteligencia de negocios *Pentaho BI server v4.8.0 Community Edition*. Para el proceso de extracción, transformación y carga de datos se va hacer uso de la herramienta *Pentaho Data Integration v4.4.0*. La entrega de información se va a realizar a través del portal del *Pentaho User Console*, por medio de este portal *web* los usuarios finales van a visualizar las tres soluciones de análisis como el reporte con la herramienta *Pentaho Report Designer v3.9.1*, el cubo *OLAP* con la herramienta *Pentaho Schema Workbench v3.4.1.1*, y el tablero de mando integral con la herramienta *Community Dashboard Framework v4.8.0*.

Se utilizará la base de datos *PostgreSQL v9.2* para almacenar los datos de los sistemas fuente, de los procesos *ETL*, de los metadatos, y los datos del *data mart* DM_VENTAS.

Se diseñó el modelo dimensional de datos para el *data mart* DM_VENTAS en relación a cada requerimiento funcional. Cada entregable se detalla a continuación:

- Modelo dimensional de datos para DM_VENTAS.
- Diccionario de datos del modelo dimensional DM_VENTAS (Anexo 3).

Modelo dimensional de datos para DM_VENTAS



Cada tabla se encuentra implementada con un esquema tipo estrella. F_VENTAS tiene relación con cada dimensión del modelo de ventas. La tabla en mención es una tabla transaccional la cual almacena cada registro de las ventas generadas por los centros de atención a clientes de Quito, por producto y por asesor, al igual que la cantidad de ventas por día, mes y año. Por medio del modelo dimensional se visualiza el producto por cada venta.

Cada registro de F_VENTAS pertenece a toda venta procesada. Las dimensiones que corresponden a F_VENTAS son: D_FECHA, D_CAC, D_PLAN, D_CLIENTE Y D_ASESOR, en el anexo 3 se detalla el diccionario de datos de la tabla de hechos y de cada dimensión. En el anexo 4 se detalla el código SQL utilizado para la creación de la tabla de hechos y de cada dimensión.

f. Desarrollo

En esta fase se va a construir las tres partes principales del sistema del almacén de datos:

- ✓ Los *data stores* PDI_REPO, STAGE, y DM_VENTAS.
- ✓ Los procesos *ETL*.
- ✓ La interface de usuario final.

Adicionalmente se va a detallar los indicadores de gestión y las soluciones de análisis como el reporte, el cubo OLAP, y el tablero de mando.

f.1 Creación de los almacenes de datos (*data stores*)

Para este proyecto de tesis se instaló la base de datos *PostgreSQL v9.2* y se creó la instancia de base de datos denominado DWH. En el anexo 5 se detalla cada paso para la instalación y configuración del servidor de la base de datos *PostgreSQL*.

Se crearon los *data stores* PDI_REPO, STAGE, y DM_VENTAS. El *data store* PDI_REPO es el repositorio de la herramienta *Pentaho Data Integration*, en el ANEXO 6 se detallan los pasos para la configuración de la herramienta de integración de datos *Pentaho Data Integration*.

f.2 Los procesos de extracción, transformación, y carga

En el siguiente subcapítulo, se realizó los respectivos procesos *ETL*, cada proceso está compuesto por dos elementos:

- Transformaciones: Conjunto de *steps* (pasos) que se ejecutan de manera sistemática.
- *JOBS*: Conjunto de transformaciones que se ejecutan de manera sistemática. (Bouman, Dongen, 2009 p. 233-235)

En la herramienta Spoon se creó el repositorio PDI_REPO, dentro de éste repositorio se crearon los *JOB*S: JOB_CARGA_STAGE, JOB_CARGA_DIMENSIONES, JOB_CARGA_HECHOS, y JOB_CARGA_DM_VENTAS.

- El *JOB* JOB_CARGA_STAGE almacena las transformaciones TRS_CARGA_ARCHIVOS_FUENTE_CSV, TRS_LIMPIEZA_DATOS_FUENTE, TRS_CARGA_ARCHIVO_FUENTE_XLS, y TRS_PREPARACION_FINAL_DATOS de los procesos de extracción, transformación y carga de datos para transportar los datos desde los sistemas fuente hasta el *data store* STAGE de manera periódica.
- El *JOB* JOB_CARGA_DIMENSIONES almacena las transformaciones de las dimensiones TRS_D_CAC, TRS_D_PLAN, TRS_D_CLIENTE, TRS_D_ASESOR, y TRS_D_FECHA de los procesos de extracción, transformación y carga de datos para transportar los datos desde el *data store* STAGE hasta el *data store* DM_VENTAS de manera periódica.
- El *JOB* JOB_CARGA_HECHOS almacena las transformaciones de la tabla de hechos TRS_F_VENTAS de los procesos de extracción, transformación y carga de datos para transportar los datos desde el *data store* STAGE hasta el *data store* DM_VENTAS de manera periódica.
- El *JOB* JOB_CARGA_DM_VENTAS almacena los *JOB*S JOB_CARGA_STAGE, JOB_CARGA_DIMENSIONES, y JOB_CARGA_HECHOS de los procesos de extracción, transformación y carga de datos para transportar los datos desde el *data store* STAGE hasta el *data store* DM_VENTAS de manera periódica.

f.2.1 ETLs (Fuente al *stage*), Transformaciones y JOBS

f.2.1.1 Transformación TRS_CARGA_ARCHIVOS_FUENTE_CSV

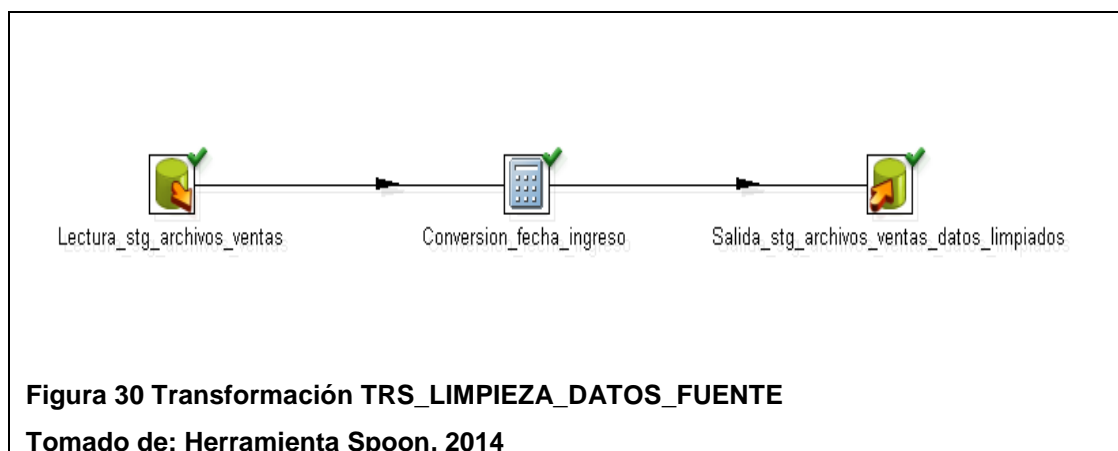
Esta transformación realizó la lectura de datos desde los archivos fuente csv y los carga en la tabla `stg_archivos_ventas` de la base de datos *stage*.



f.2.1.2 Transformación TRS_LIMPIEZA_DATOS_FUENTE

Esta transformación realizó la lectura de datos desde la tabla `stg_archivos_ventas` de la base de datos *stage*. En la figura 2.11 se observa los procesos de limpieza de datos en el campo cédula.

En la figura 2.13 se observa los procesos de limpieza de datos en el campo `fecha_ingreso`. Una vez terminado cada proceso, se realizó la carga de datos en la tabla `stg_archivos_ventas_datos_limpiados` del *data mart* DM_VENTAS.



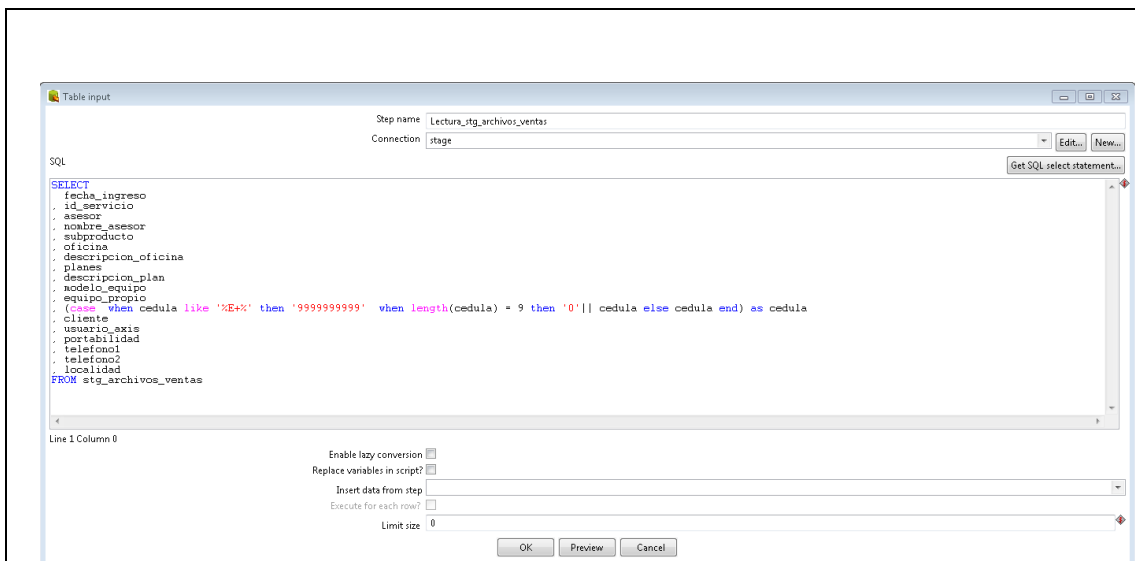


Figura 31 Limpieza de datos campo cédula

Tomado de: Herramienta Spoon, 2014

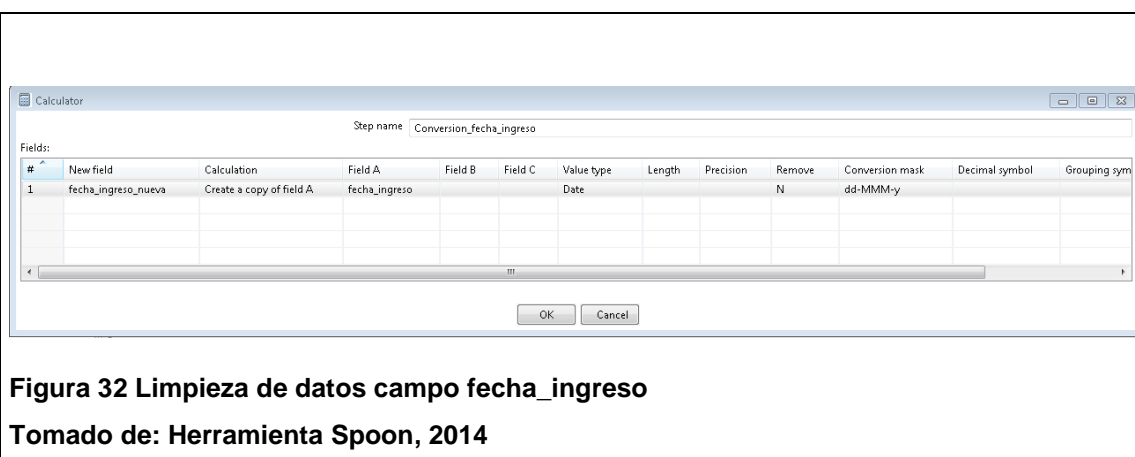
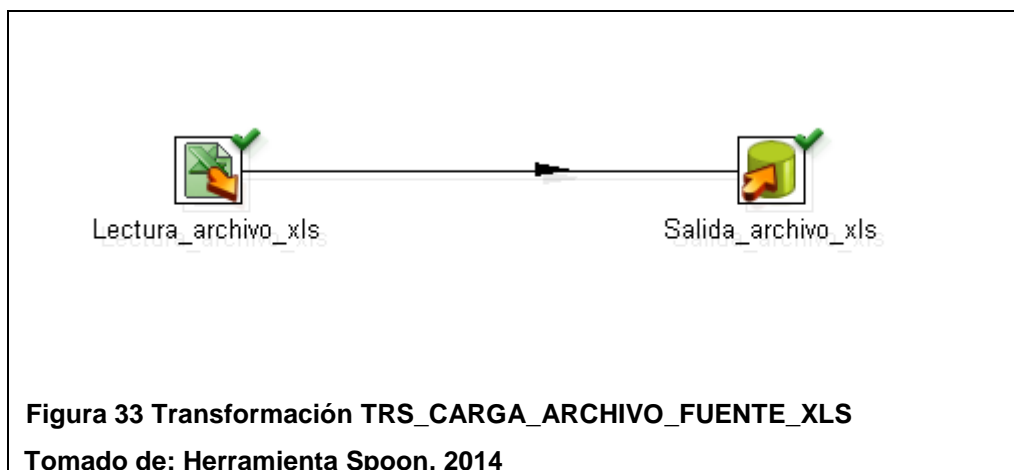


Figura 32 Limpieza de datos campo fecha_ingreso

Tomado de: Herramienta Spoon, 2014

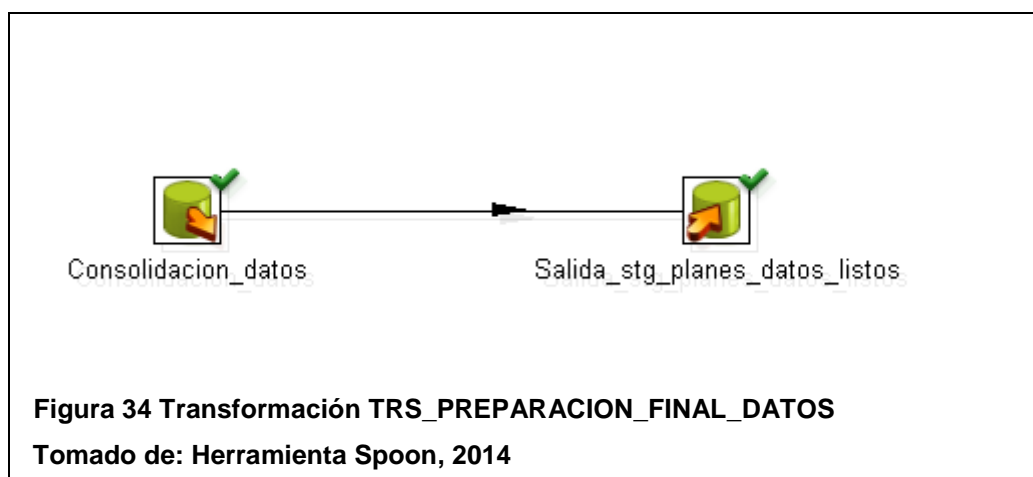
f.2.1.3 Transformación TRS_CARGA_ARCHIVO_FUENTE_XLS

Esta transformación realizó la lectura de datos desde el archivo fuente xls y los cargó en la tabla stg_planes de la base de datos stage.



f.2.1.4 Transformación TRS_PREPARACION_FINAL_DATOS

Esta transformación consolidó la información de la tabla `stg_archivos_ventas_datos_limpiados` con `stg_planes`. Ambas tablas se encuentran ubicadas en la base de datos `stage`.



En la figura 2.16 se observa la transformación en donde se consolidó la información de las tablas `stg_archivos_ventas_datos_limpiados` con `stg_planes` y se mapea el campo `planes` con `id_plan` respectivamente, una vez verificado que los datos sean consistentes, se realizó la carga en la tabla `stg_planes_datos_listos`, todas estas transformaciones se ejecutaron en la base de datos `stage`.

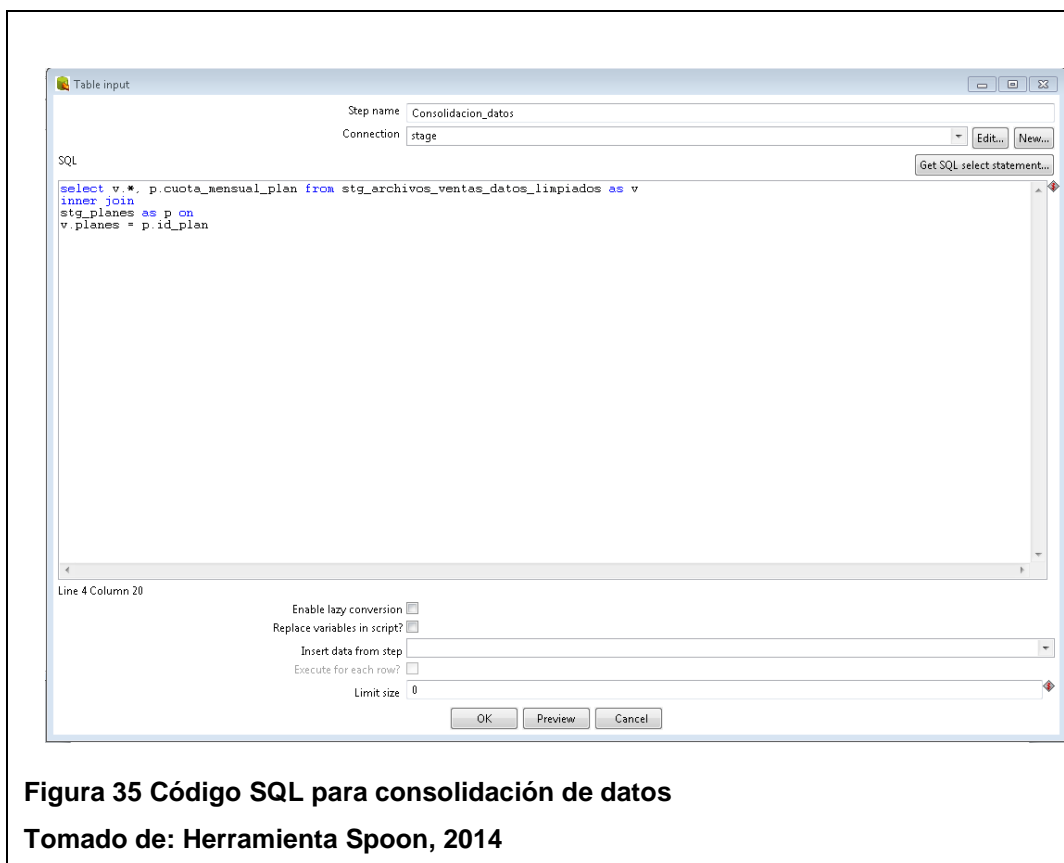


Figura 35 Código SQL para consolidación de datos

Tomado de: Herramienta Spoon, 2014

f.2.1.5 JOB_CARGA_STAGE

En el siguiente *JOB* (trabajo), se valida que todas las transformaciones se han ejecutado de manera exitosa y se verificó que todos los 684.543 registros se encuentran contabilizados. Esto quiere decir que no se ha perdido información de los archivos fuente.

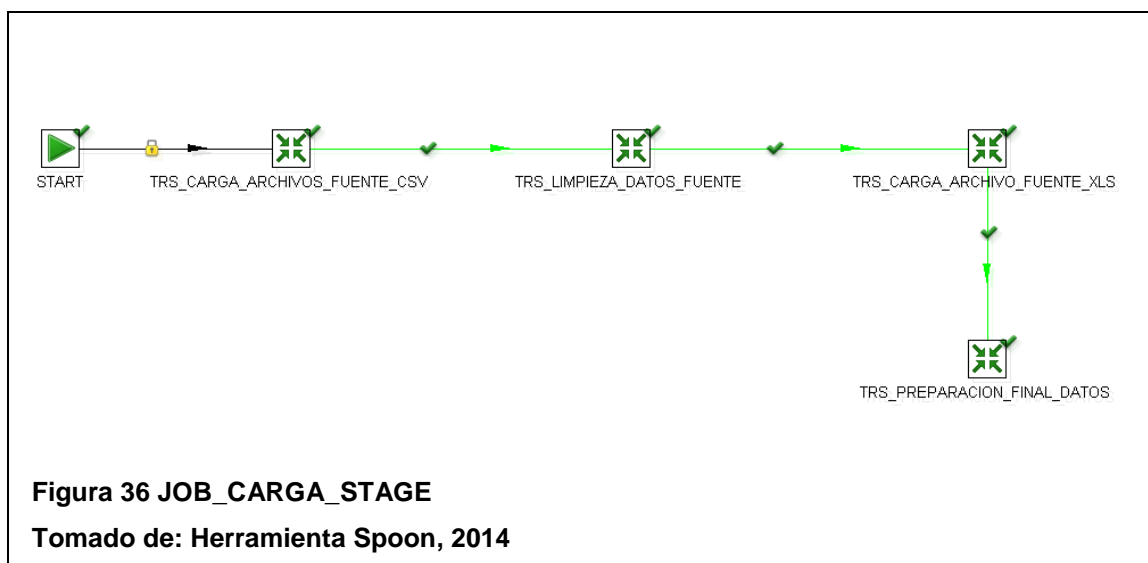


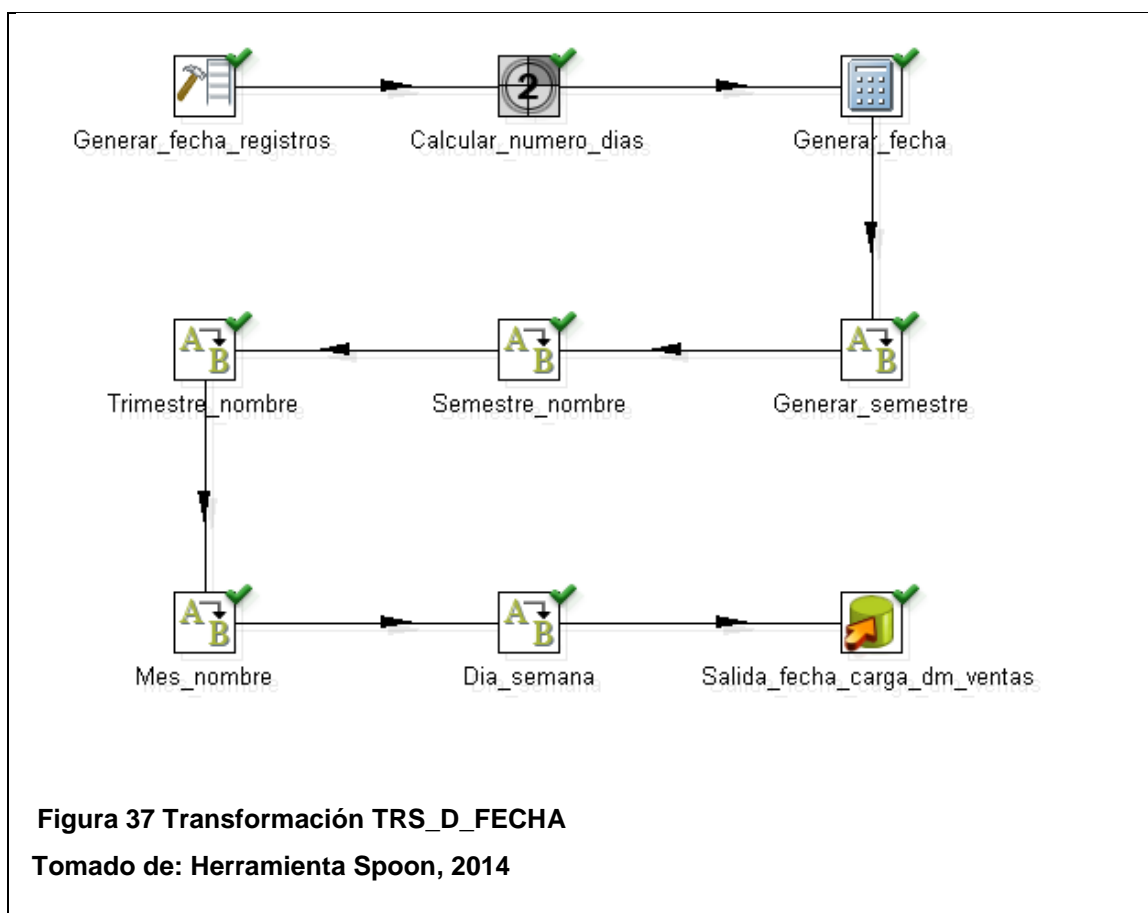
Figura 36 JOB_CARGA_STAGE

Tomado de: Herramienta Spoon, 2014

f.2.2 ETLs (stage al data mart), carga de tablas de dimensiones y de hechos

f.2.2.1 Transformación TRS_D_FECHA

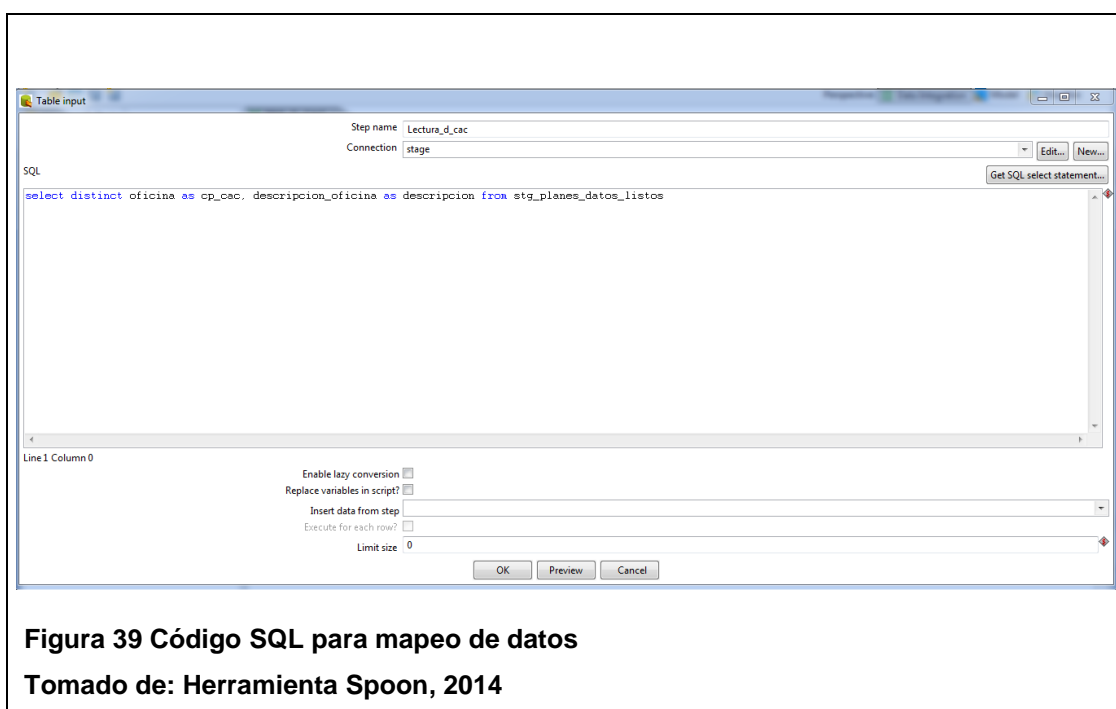
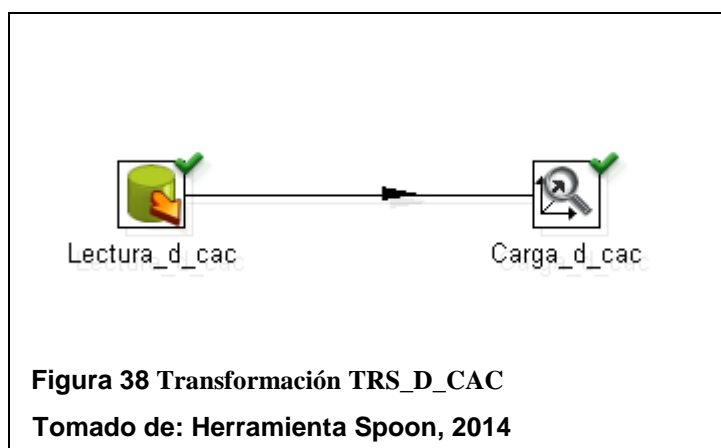
Esta transformación realizó todos los procesos respectivos para cargar la fecha en la dimensión D_FECHA del *data mart* DM_VENTAS. Es importante tomar en cuenta que D_FECHA, solo se carga una sola vez en la tabla.



f.2.2.2 Transformación TRS_D_CAC

Esta transformación realizó el mapeo todos los campos de la dimensión D_CAC desde la tabla stg_planes_datos_listos hasta cargar los datos en DM_VENTAS.

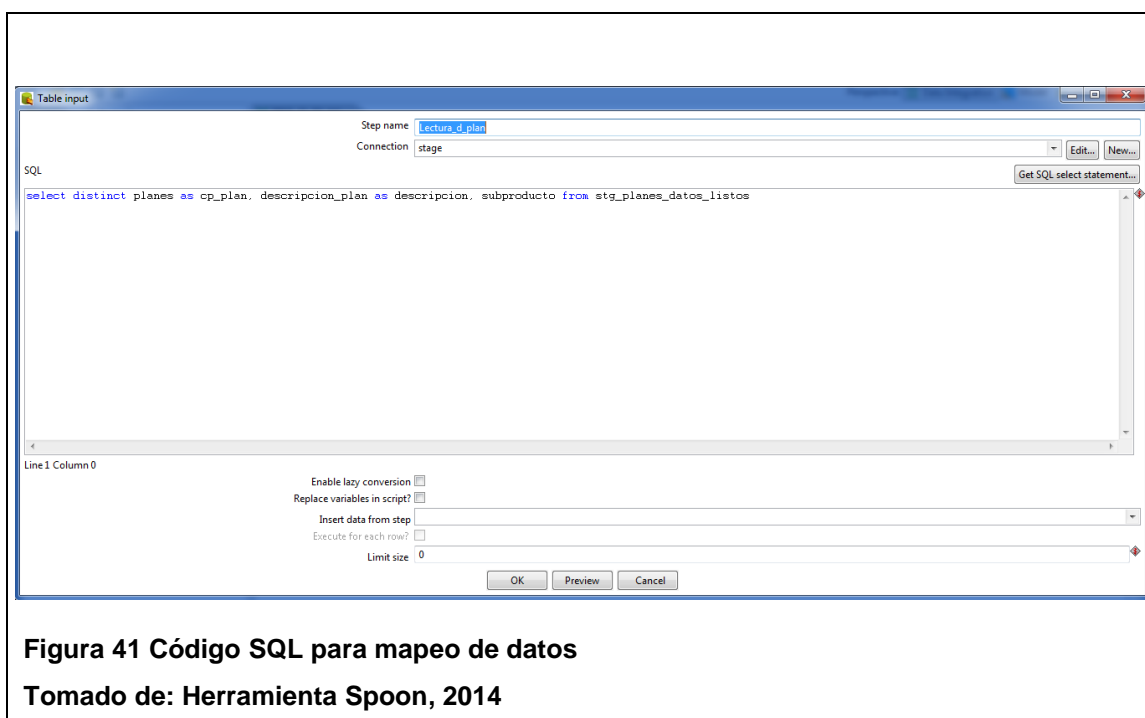
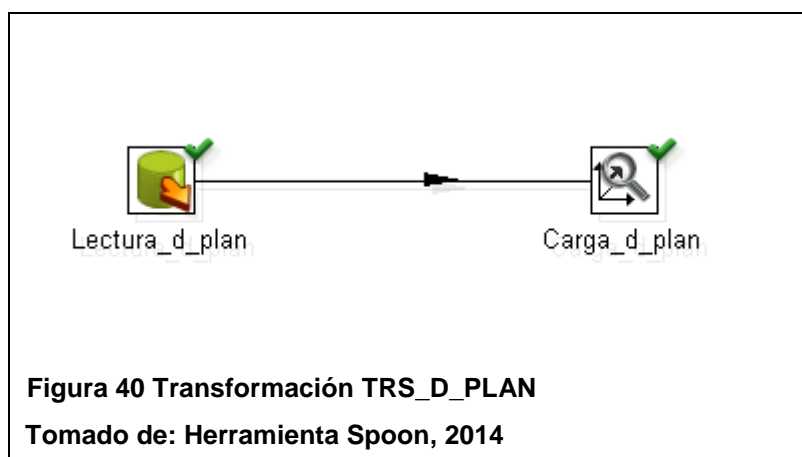
En la figura 2.20 se observa lo mencionado.



f.2.2.3 Transformación TRS_D_PLAN

Esta transformación realizó el mapeo todos los campos de la dimensión D_PLAN desde la tabla stg_planes_datos_listos hasta cargar los datos en DM_VENTAS.

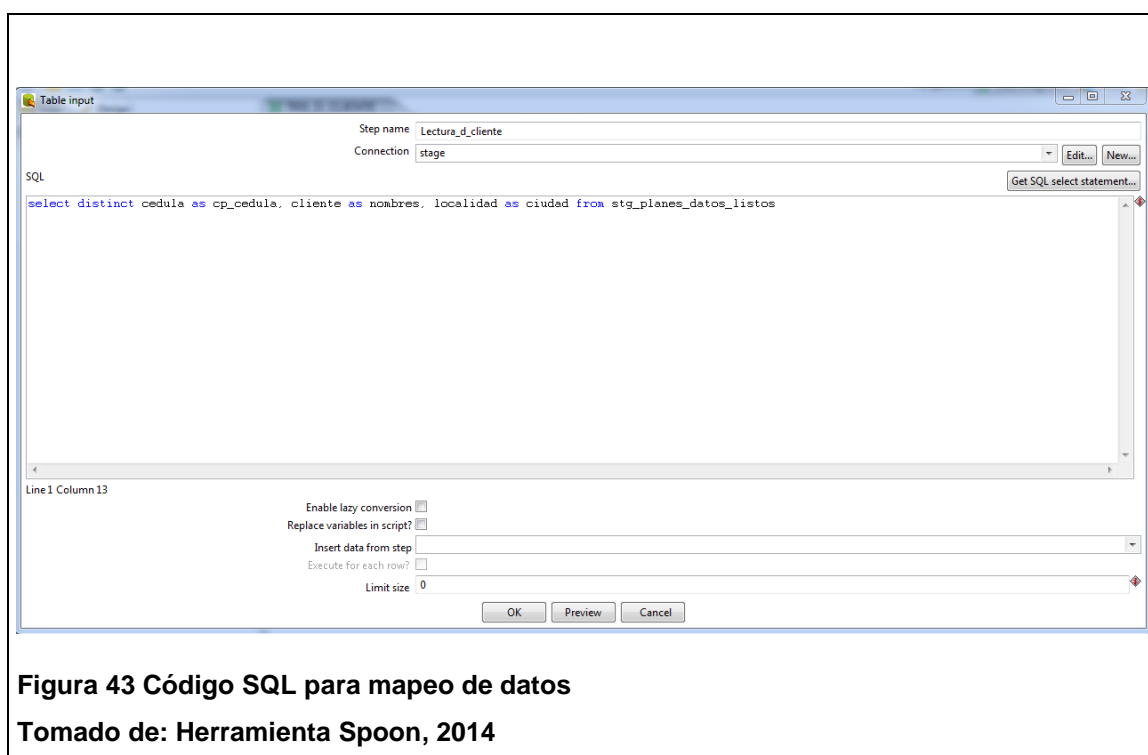
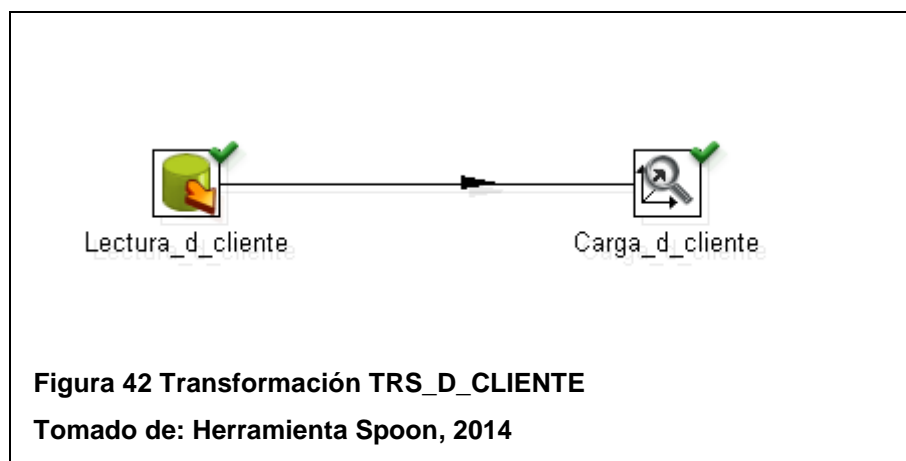
En la figura 2.22 se observa lo mencionado.



f.2.2.4 Transformación TRS_D_CLIENTE

Esta transformación realizó el mapeo todos los campos de la dimensión D_CLIENTE desde la tabla stg_planes_datos_listos hasta cargar los datos en DM_VENTAS.

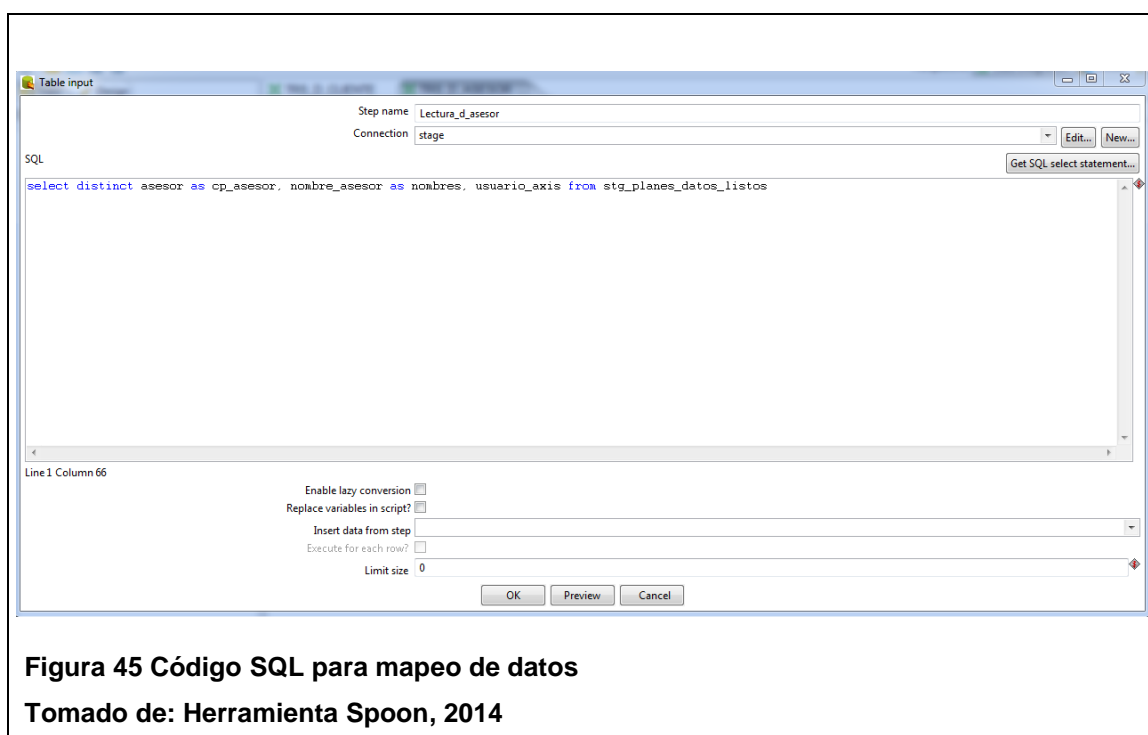
En la figura 2.24 se observa lo mencionado.



f.2.2.5 Transformación TRS_D_ASESOR

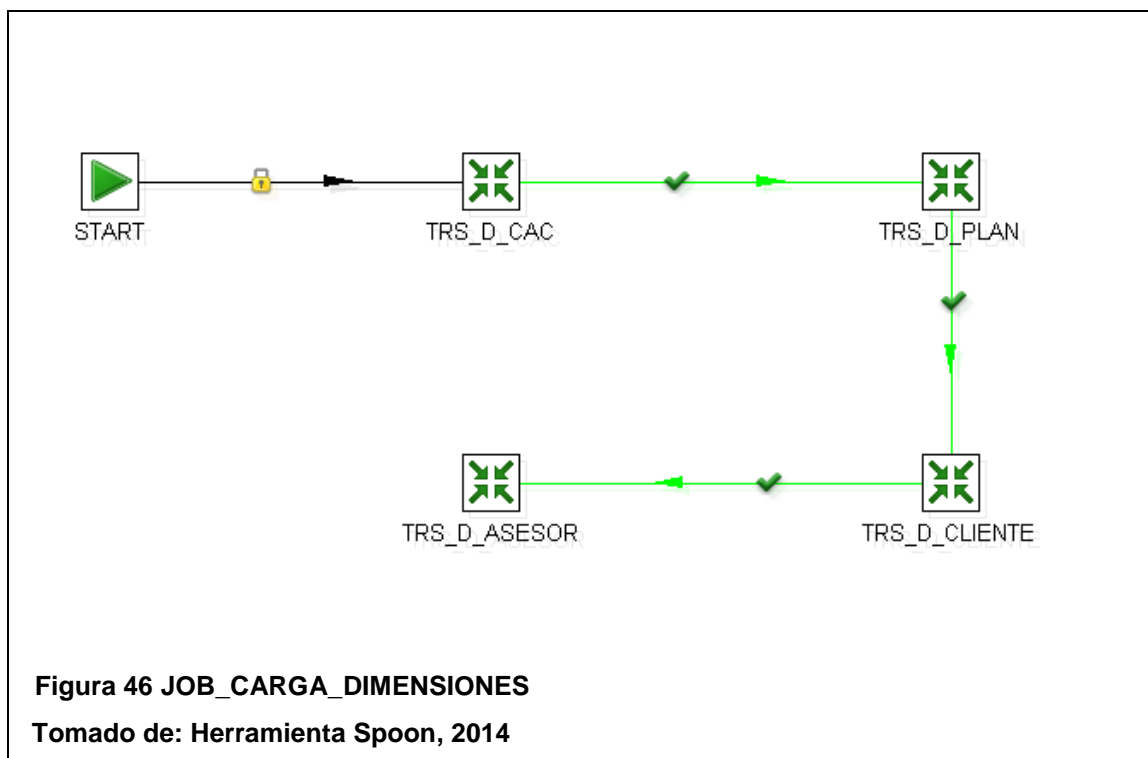
Esta transformación realizó el mapeo todos los campos de la dimensión D_CAC desde la tabla stg_planes_datos_listos hasta cargar los datos en DM_VENTAS.

En la figura 2.26 se observa lo mencionado.



f.2.2.6 JOB_CARGA_DIMENSIONES

El siguiente *JOB* (trabajo), realizó la carga de todas las tablas de dimensiones. La dimensión D_FECHA solo se ejecuta una vez, motivo por el cual no se cargó en el *JOB*.



f.2.2.7 Transformación TRS_F_VENTAS

Esta transformación primero realizó el mapeo de todas las claves subrogadas de cada dimensión y luego ejecutó la carga de datos en la tabla de hechos F_VENTAS del *data mart* DM_VENTAS. Por temas de confidencialidad de la información, se realizó un cálculo, el cual modificó los datos de ventas pero se manejan en proporciones iguales para no alterar el respectivo análisis.

En la figura 2.29 se observa lo mencionado.

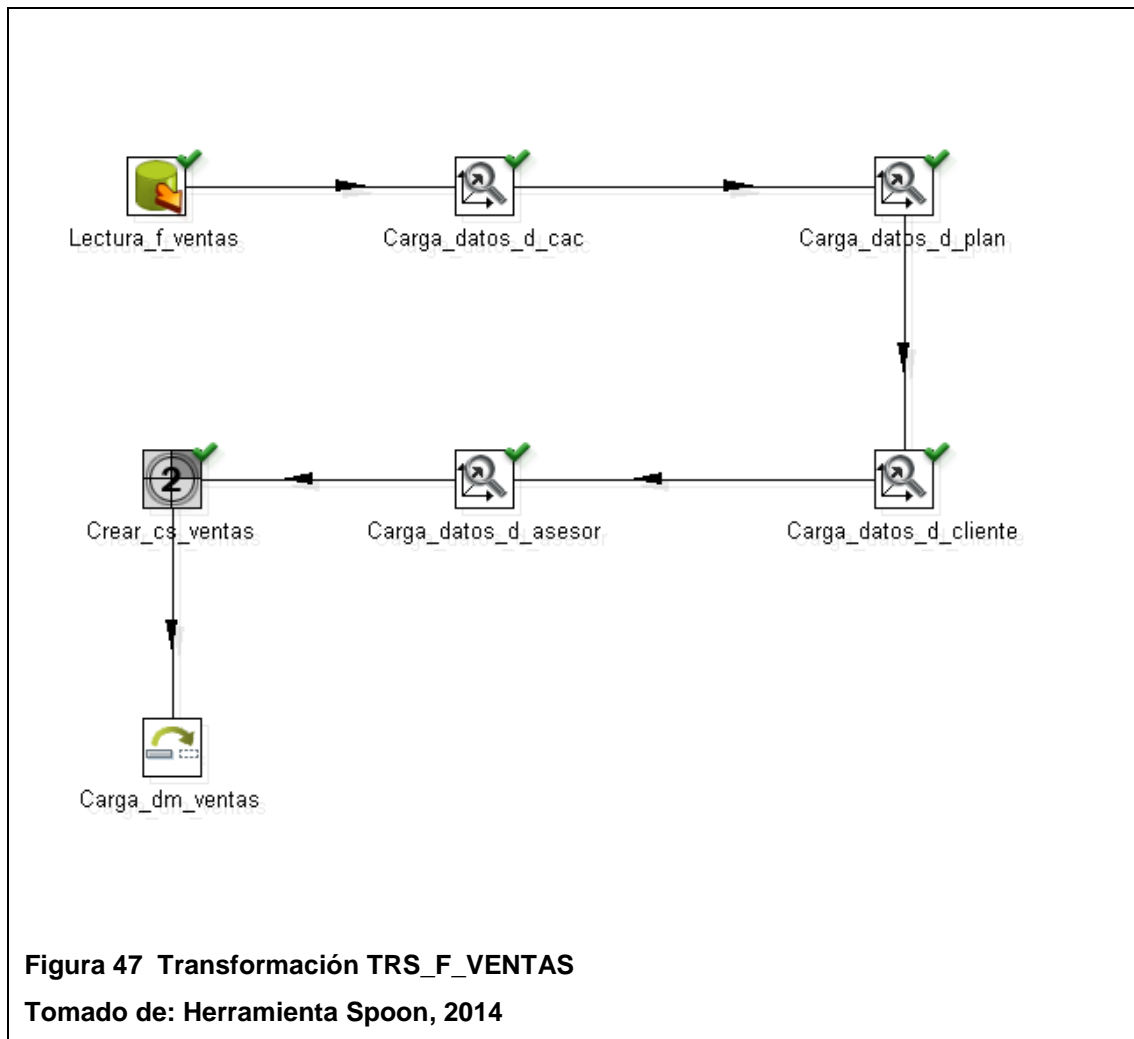


Figura 47 Transformación TRS_F_VENTAS

Tomado de: Herramienta Spoon, 2014

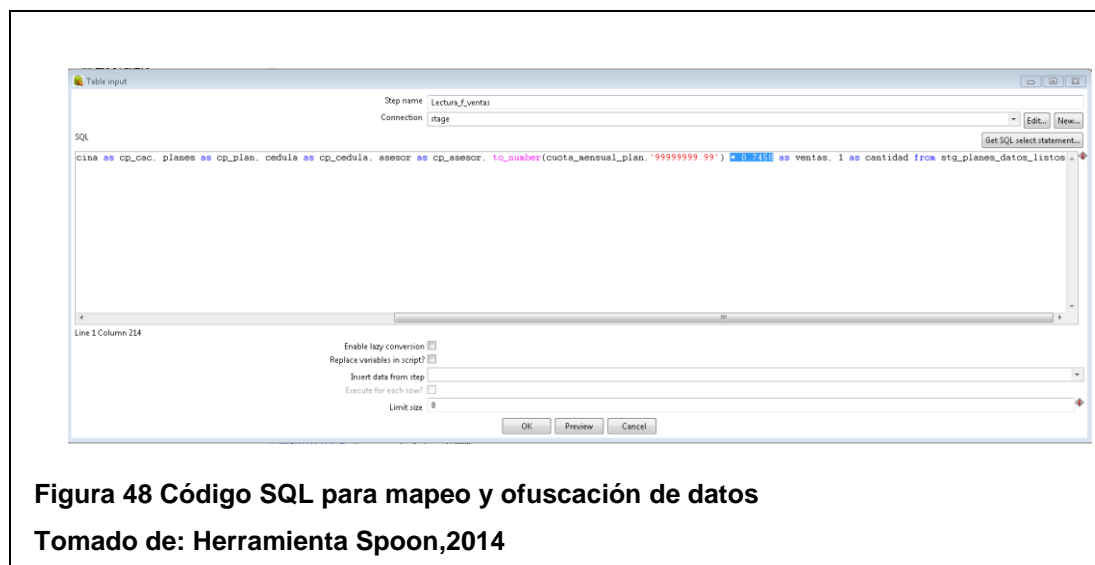
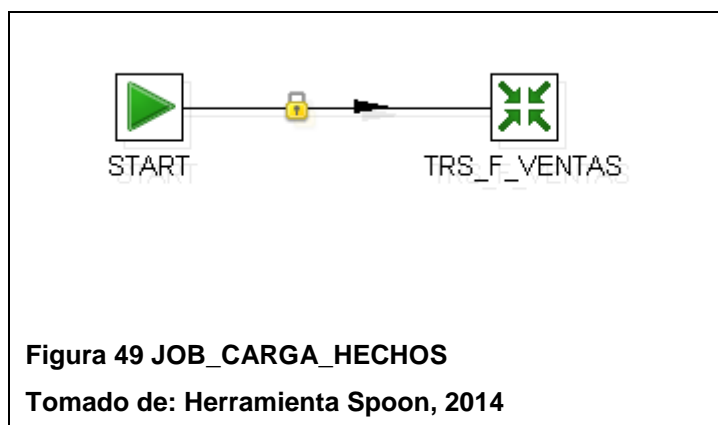


Figura 48 Código SQL para mapeo y ofuscación de datos

Tomado de: Herramienta Spoon,2014

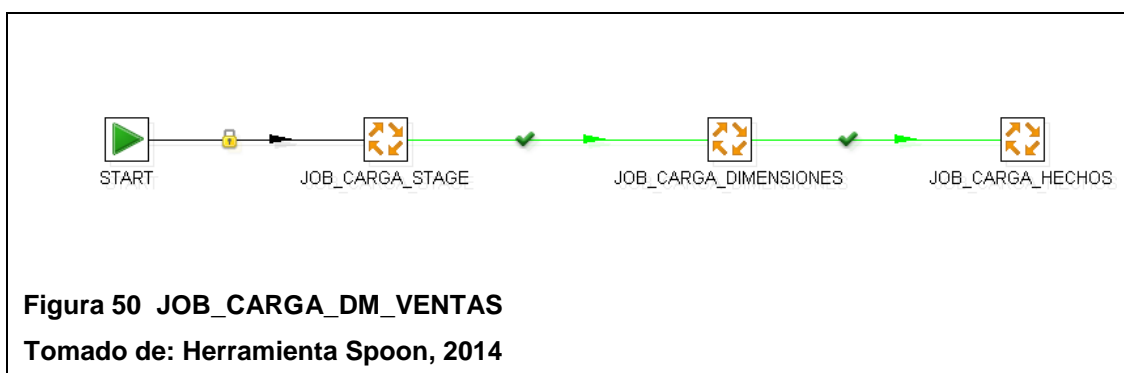
f.2.2.8 JOB_CARGA_HECHOS

El siguiente *JOB*, realizó la carga de la tabla de hechos F_VENTAS.



f.2.2.9 JOB_CARGA_DM_VENTAS

El siguiente *JOB*, realizó la carga de todos los trabajos: JOB_CARGA_STAGE, JOB_CARGA_DIMENSIONES Y JOB_CARGA_HECHOS.



f.3 La interface de usuario final

f.3.1 Indicadores de gestión y soluciones de análisis

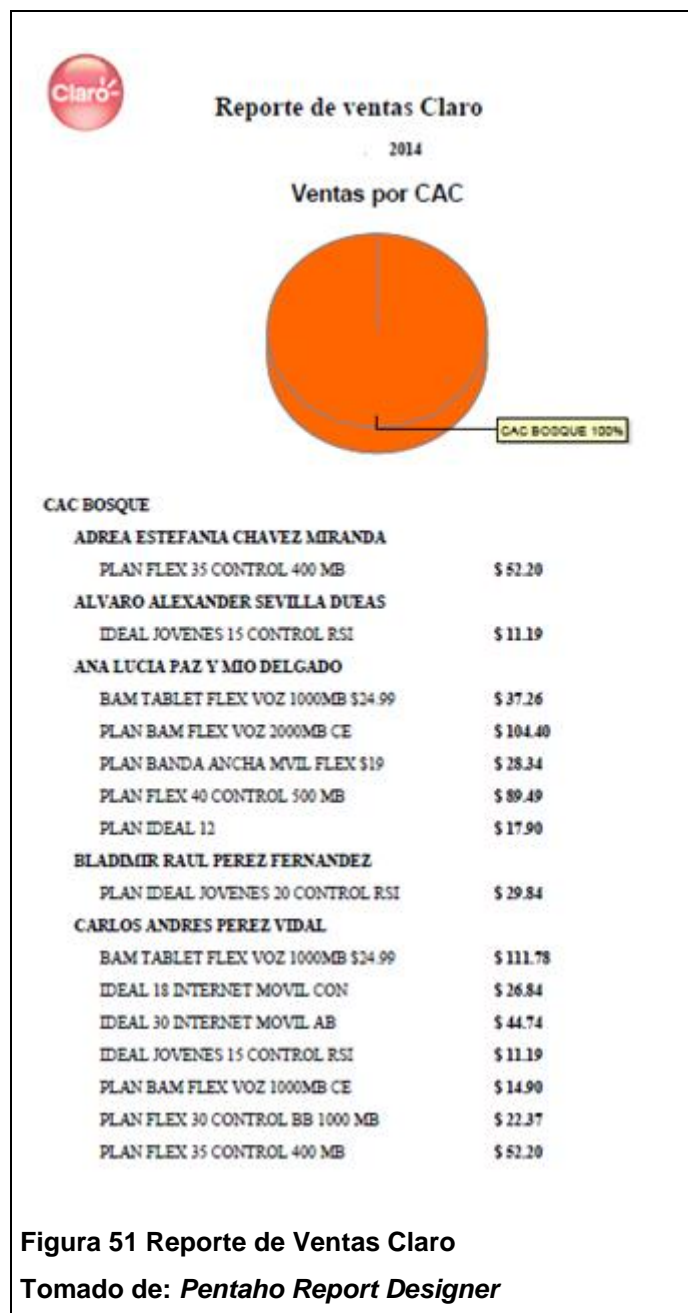
En la tabla 2.5 se detallan los indicadores de gestión

Tabla 6 Indicadores de gestión

Indicador	Detalle
Ventas	Ventas generadas en dólares por los centros de atención a clientes de Quito.
Cantidad	Número de ventas generadas por los centros de atención a clientes de Quito.

f.3.1.1 Reporte con los indicadores de gestión

Se creó un reporte en donde se puede responder las preguntas definidas en los requerimientos funcionales. Se construyó este reporte por medio de la herramienta *Pentaho Report Designer*.



f.3.1.2 Cubo OLAP con los indicadores de gestión

Se creó un cubo *OLAP* en donde se puede responder las preguntas definidas en los requerimientos funcionales. Se construyó este cubo por medio de la herramienta *Pentaho Schema Workbench*.

					Measures
Fecha	CAC	Plan	Cliente	Asesores	● Cantidad
- Todas las fechas	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	684543
- 2013	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	265376
- 2014	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	419167
- SEMESTRE I	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	158646
- TRIMESTRE I	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	158646
+ 1	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	51113
+ 2	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	51132
+ 3	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	56401
- SEMESTRE II	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	174515
- TRIMESTRE II	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	174515
+ 4	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	57198
+ 5	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	63517
+ 6	+ Todas los CACs	+ Todos los Planes	+ Todos los Clientes	+ Todos los asesores	53800

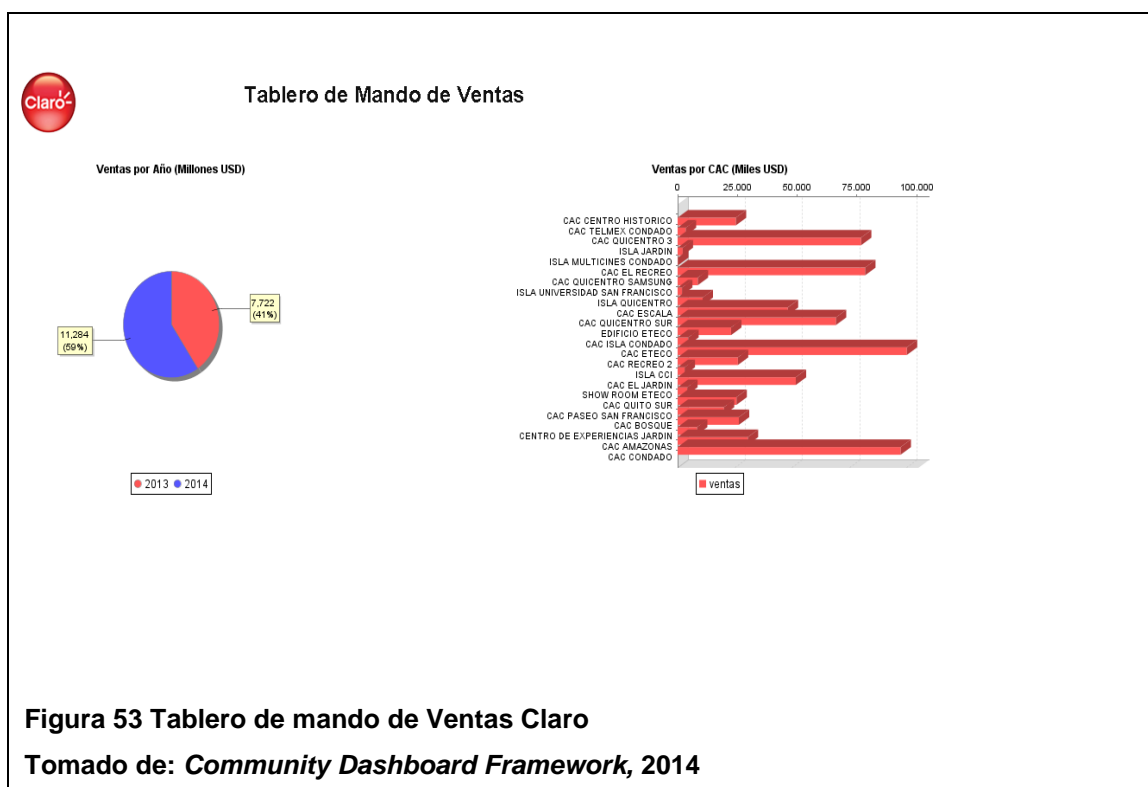
Figura 52 Cubo OLAP

Tomado de: *Pentaho Schema Workbench, 2014*

f.3.1.3 Tablero de mando con los indicadores de gestión

Se creó un tablero de mando en donde se puede responder las preguntas definidas en los requerimientos funcionales. Se construyó este tablero por medio de la herramienta *Community Dashboard Framework*.

Las tres soluciones se visualizan a través de la consola de usuario de *Pentaho*.



g. Pruebas

Luego de construir los *data stores*, los procesos *ETL*, y la interface de usuario final es necesario realizar las respectivas pruebas para validar que la información cargada en el *data mart* DM_VENTAS sea consistente en relación a los sistemas fuente. Estos sistemas usualmente poseen reportes preestablecidos, de los cuales los usuarios finales pueden obtener cierto tipo de información, la principal desventaja es que los usuarios finales no pueden generar reportes a demanda para responder preguntas específicas del negocio.

Para la validación de las pruebas de carga de datos se utilizó dos reportes de los sistemas fuente. El reporte denominado Reporte de líneas activas en SCT y el Reporte Detalle de planes y códigos BP. Se compara los datos del reporte en excel que el analista de información envía a la directora, el gerente y los jefes con el reporte generado con *Pentaho Report Designer* y se valida que son los mismos resultados. Es necesario recordar que en la transformación

TRS_F_VENTAS, detallado en la figura 2.28 se realizó un cálculo sobre las ventas para generar ofuscación de los datos por temas de confidencialidad.

En la siguiente figura se observa el reporte de ventas realizado en Excel sin la ofuscación de los datos.

	A	B	C	D	E
1	FECHA_INGRESO	(Todas)			
2	DESCRIPCION_OFICINA	CAC BOSQUE			
3					
4	NOMBRE_ASESOR	DESCRIPCION_PLAN	TARIFA_BASICA_PLAN	Cuenta de DESCRIPCION_PLAN	VALOR USD
5	ADREA ESTEFANIA CHAVEZ MIRANDA	PLAN FLEX 35 CONTROL 400 MB	35	2	70
6	ALVARO ALEXANDER SEVILLA DUEÑAS	IDEAL JOVENES 15 CONTROL RSI	15	1	15
7	ANA LUCIA PAZ Y MIÑO DELGADO	BAM TABLET FLEX VOZ 1000MB \$24.9	24,99	2	49,98
8		PLAN BAM FLEX VOZ 2000MB CE	34,99	4	139,96
9		PLAN BANDA ANCHA MÓVIL FLEX \$1	19	2	38
10		PLAN FLEX 40 CONTROL 500 MB	40	3	120
11		PLAN IDEAL 12	12	2	24
12	BLADIMIR RAUL PEREZ FERNANDEZ	PLAN IDEAL JOVENES 20 CONTROL RSI	20	2	40
13	CARLOS ANDRES PEREZ VIDAL	BAM TABLET FLEX VOZ 1000MB \$24.9	24,99	6	149,94
14		IDEAL 18 INTERNET MOVIL CON	18	2	36
15		IDEAL 30 INTERNET MOVIL AB	30	2	60
16		IDEAL JOVENES 15 CONTROL RSI	15	1	15
17		PLAN BAM FLEX VOZ 1000MB CE	19,99	1	19,99
18		PLAN FLEX 30 CONTROL BB 1000 MB	30	1	30
19		PLAN FLEX 35 CONTROL 400 MB	35	2	70
20		PLAN FLEX 40 CONTROL 500 MB	40	2	80
21		PLAN IDEAL 12	12	2	24
22		PLAN IDEAL 22 CONT	22	1	22
23		PLAN JOVENES 15 CONTROL RSI EP	15	2	30
24		PLAN NETBOOK FLEX 2000 MB PROM	29	1	29

Figura 54 Reporte de ventas en Excel

Tomado de: Reporte de Líneas Activas en SCT, 2014

h. Despliegue

En esta fase en el ambiente de producción se realiza la carga inicial de datos desde los sistemas fuente, hacia el stage y al *data mart* de ventas.

i. Operación

En esta fase los usuarios finales están utilizando la aplicación por medio de la consola de usuario de *Pentaho*. La administración del *data mart* y el soporte a los usuarios finales los administra mi persona.

En el caso que los usuarios finales soliciten opciones de mejora, se colocan en orden de importancia y luego se rechazan o se les incluye en la próxima versión.

j. Administración del proyecto

Esto se da cuando se mantiene el plan del proyecto, lo cual significa que en cualquier momento se pueda conocer el estado de cada tarea, quien la realiza y cuando se va a llevar a cabo.

Es importante mencionar que se mantiene una buena comunicación con todas las partes involucradas como la directora, el gerente y los jefes de los centros de atención a clientes de Quito.

2.5 Encuestas y análisis de resultados de las encuestas

Por medio del siguiente banco de preguntas se realizó las encuestas a algunos de los usuarios finales sobre el uso de la aplicación de la consola de usuario de *Pentaho*. En el anexo 7 se encuentran escaneadas las encuestas realizadas.

2.5.1 Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de *Pentaho*.

Fecha: _____

Nombre: _____

1 El uso de la consola de usuario de *Pentaho* es sencillo y amigable?

Si _____ No _____ ¿Por qué?

2 Los reportes muestran la información necesaria para poder realizar un análisis sobre el negocio?

Si _____ No _____ ¿Por qué?

3 La información consolidada y centralizada en un solo lugar, ¿les permitirá un mejor análisis en sus reportes?

Si _____ No _____ ¿Por qué?

4 Según su opinión esta solución de inteligencia de negocios, ¿les disminuirá los tiempos de respuesta a las consultas de los usuarios finales?

Si _____ No _____ ¿Por qué?

5 Luego de haber probado la aplicación, ¿la seguiría utilizando?

Si _____ No _____ ¿Por qué?

6 Mencione algunas ventajas y desventajas del aplicativo, para el desarrollo de su trabajo.

2.5.2 La tabulación de los resultados de la encuesta se encuentra detallado en el anexo 7.

Capítulo 3

3. Análisis de costos

3.1 Cuadro comparativo de soluciones de inteligencia de negocios

Tabla 7 Cuadro de costos de soluciones de inteligencia de negocios

HERRAMIENTA	CARACTERISITCAS	COSTO USD
Tableau Desktop Profesional	Instalación hasta en dos pc's, cliente Windows o mac)	\$ 2.100
Tableau Server	Con 10 usuarios interactores (Mínimo 10 usuarios, se puede adquirir usuarios unitarios a partir del usuario 11)	\$ 13.050
Pentaho Enterprise Edition	Pentaho BA: <ul style="list-style-type: none"> - Versión limited (20 usuarios / 4 cores) - Versión basic (usuarios ilimitados / 4 cores) - Versión Professional Standard (usuarios ilimitados / 8 cores) - Versión Professional Premium (nivel de soporte Premium 7x24x365 / usuarios ilimitados / 8 cores) 	\$ 16.000 \$ 34.000 \$ 44.000 \$ 50.000

	<p>- Versión Enterprise (nivel de soporte Premium 7x24x365 / usuarios ilimitados / 16 cores / hasta 25 nodos hadoop)</p> <p>Pentaho Data Integration:</p> <p>- Versión profesional estándar (usuarios ilimitados / 8 cores)</p> <p>- Versión Professional Premium (nivel de soporte Premium 7x24x365 / usuarios ilimitados / 8 cores)</p> <p>- Versión Enterprise (nivel de soporte Premium 7x24x365 / usuarios ilimitados / 16 cores / soporte para hadoop y NoSQL)</p>	<p>\$ 70.000</p> <p>\$ 22.000</p> <p>\$ 30.000</p> <p>\$ 42.000</p>
Pentaho Community Edition BI suite	Software libre	\$ 0

Tomado de: Tableau y Datalytics

3.2 Detalle de costos

Todos los costos detallados fueron asumidos por Claro.

Tabla 8 Detalle de costos

Detalle	Cantidad	Precio por unidad	Total USD
PC Dell OptiPLex 3010	1	500	\$ 500
Pentaho Community Edition BI suite	1	0	\$ 0
PostgreSQL para Windows 32 bits	1	0	\$ 0
Centro de capacitación continua – Escuela Politécnica Nacional Data Warehousing Básico con Pentaho - Módulo 01	1	250	\$ 250
Centro de capacitación continua – Escuela Politécnica	1	250	\$ 250

Nacional			
Business Intelligence con Pentaho - Módulo 02			
Sueldo Egresado	1	800	\$ 800
		Total	\$ 1.800

3.3 Análisis costo – beneficio de la implementación de la solución de inteligencia de negocios para el módulo de ventas de Claro

Se definió dos escenarios para medir si el análisis de costo – beneficio de la implementación de inteligencia de negocios es beneficioso o no para la empresa. En el primer escenario se obtiene el costo por la generación de los ocho reportes generados por semana, por mes, y por año sin la implementación de la herramienta de inteligencia de negocios. En el segundo escenario se obtiene el costo por la generación de n número de reportes de ventas con la implementación de la herramienta de inteligencia de negocios.

Los costos y beneficios se pueden dividir en cuatro categorías:

1) Los costos de desarrollo

Son aquellos gastos tangibles que se incurren durante la creación del sistema, tales como los sueldos del equipo del proyecto, los gastos de hardware y software, los honorarios de consultoría, la capacitación, el espacio de la oficina, y de los equipos. Los costos de desarrollo usualmente son considerados como gastos no recurrentes.

2) Los costos operativos

Son aquellos costos tangibles que se requieren para operar el sistema, tales como los sueldos del personal de operaciones, los derechos de las licencias de software, las actualizaciones de los equipos, y los gastos en comunicaciones. Los costos operativos son considerados como gastos en curso.

3) Los beneficios tangibles

Incluyen los ingresos que el sistema va a generar para la empresa como el aumento de las ventas. Además, el sistema incluye otro beneficio tangible como el ahorro de costos. Por ejemplo, si el sistema produce una reducción en las horas necesarias para que el analista de información genere los reportes de ventas, esto quiere decir que la empresa va a ahorrar en el pago de horas al analista pero adicionalmente el analista puede dedicar ese tiempo adicional a realizar otras actividades igual de productivas para la empresa.

4) Los beneficios intangibles

Por supuesto, un proyecto también puede afectar a los resultados finales de la organización por la obtención de beneficios intangibles o incurrir en costos intangibles. Los costos y beneficios intangibles son más difíciles de incorporar en el análisis de la viabilidad económica, debido a que se basan en percepción. Algunos ejemplos de costos intangibles son el aumento de la participación en el mercado, el aumento en el reconocimiento de marca, el aumento en la calidad de los productos, el mejoramiento en el servicio al cliente, y mejores relaciones con los proveedores. (Dennis, Wixom, Roth, 2008)

3.3.1 Identificar los costos y beneficios

El primer paso cuando se va a desarrollar un análisis de costo – beneficio es identificar los posibles tipos de costos y beneficios. Para este capítulo se va a enfocar el análisis costo - beneficio sobre la relación:

Costo: Hora/Hombre

Beneficio: Ahorro en tiempo

En la siguiente tabla se detalla cada uno de ellos.

Tabla 9 Tipos de Costos y Beneficios

Costos de desarrollo
Hora/Hombre
Beneficios tangibles
Ahorro en tiempo

Adaptado de: (Dennis, Wixom, Roth, 2008)

Para este análisis de costo – beneficio se va a realizar de los últimos tres años: 2012, 2013, y 2014.

3.3.2 Análisis costo – beneficio Generación de reportes de ventas de los años 2012, 2013, y 2014 con y sin el uso de la herramienta de inteligencia de negocios

Como se explicó anteriormente en el subcapítulo 2.3.5, el analista de información envía todos los días lunes y una vez por semana el reporte de ventas a todos los centros de atención a clientes. En el siguiente análisis se detalla la información por la generación del reporte de ventas enviado una vez por semana, tomando en cuenta que al analista de información le toma aproximadamente una hora en realizar este reporte. En una entrevista con el analista de información, él indicó que los usuarios finales le solicitan el reporte de ventas cualquier día de la semana, motivo por el cual también se incluye estos datos en el análisis, en promedio el analista genera siete reportes de ventas adicionales al reporte que envía una vez por semana. Esto quiere decir que se envían ocho reportes de ventas en promedio por semana.

Tabla 10 Análisis costo – beneficio Generación de reportes de ventas año 2012

COSTO VALOR HORA/HOMBRE	
\$ 4,87	Analista de información

ANÁLISIS COSTO HORA/HOMBRE SIN BI		
HORA	COSTO POR OCHO REPORTE	
\$ 4,87	\$ 38,96	Analista de información
	\$ 155,80	COSTO TOTAL POR MES
	\$ 1.869,60	COSTO TOTAL POR AÑO

ANÁLISIS COSTO HORA/HOMBRE CON BI		
HORA	COSTO POR OCHO REPORTE	
\$ 4,87	\$ 0	Analista de información
	\$ 0	COSTO TOTAL POR MES
	\$ 0	COSTO TOTAL POR AÑO

Tomado de: Analista de información de Claro

Tabla 11 Análisis costo – beneficio Generación de reportes de ventas año 2013

COSTO VALOR HORA/HOMBRE	
\$ 4,99	Analista de información

ANÁLISIS COSTO HORA/HOMBRE SIN BI		
HORA	COSTO POR OCHO REPORTE	
\$ 4,99	\$ 39,95	Analista de información
	\$ 159,80	COSTO TOTAL POR MES
	\$ 1.917,59	COSTO TOTAL POR AÑO

ANÁLISIS COSTO HORA/HOMBRE CON BI		
HORA	COSTO POR OCHO REPORTE	
\$ 4,99	\$ 0	Analista de información
	\$ 0	COSTO TOTAL POR MES
	\$ 0	COSTO TOTAL POR AÑO

Tomado de: Analista de información de Claro

Tabla 12 Análisis costo – beneficio Generación de reportes de ventas año 2014

COSTO VALOR HORA/HOMBRE	
\$ 5,12	Analista de información

ANÁLISIS COSTO HORA/HOMBRE SIN BI		
HORA	COSTO POR OCHO REPORTE	
\$ 5,12	\$ 40,96	Analista de información
	\$ 163,84	COSTO TOTAL POR MES
	\$ 1.966,08	COSTO TOTAL POR AÑO

ANÁLISIS COSTO HORA/HOMBRE CON BI		
HORA	COSTO POR OCHO REPORTE	
\$ 5,12	\$ 0	Analista de información
	\$ 0	COSTO TOTAL POR MES
	\$ 0	COSTO TOTAL POR AÑO

Tomado de: Analista de información de Claro

Tabla 13

Resumen Análisis costo – beneficio con BI

RESUMEN ANÁLISIS COSTO - BENEFICIO CON BI			
AÑO	BENEFICIO	COSTO	AHORRO
2012	Ahorro en tiempo	Hora/Hombre	\$ 1.869,60
2013	Ahorro en tiempo	Hora/Hombre	\$ 1.917,59
2014	Ahorro en tiempo	Hora/Hombre	\$ 1.966,08
	TOTAL AHORRO		\$ 5.753,27

En resumen en los años 2012, 2013, y 2014 la empresa pudo haber ahorrado \$ 5.753,27 dólares por año en la generación de los reportes de ventas de los centros de atención a clientes de Quito.

Es importante tomar en cuenta que con el uso de inteligencia de negocios en la empresa, el costo por la generación de reportes es de \$ 0 dólares. Adicionalmente, como otro beneficio el analista de información puede destinar su tiempo en otras actividades productivas para la empresa.

Capítulo 4

4. Conclusiones y Recomendaciones

4.1 Conclusiones

- Se ha concluido que antes de iniciar cualquier proyecto de inteligencia de negocios es necesario definir las herramientas que se van a utilizar, la metodología a implementarse y los requerimientos funcionales y no funcionales, de esta manera se evita contratiempos y pérdida de tiempo.
- Por medio de la implementación de una solución de inteligencia de negocios para el módulo de ventas de Claro, los usuarios finales pueden visualizar en línea la información sobre las ventas generadas en todos los centros de atención a clientes en Quito y según los resultados obtenidos, las respectivas áreas de dirección, gerencia, y jefaturas pueden tomar las decisiones tanto tácticas como operativas.
- Para la implementación de la solución de inteligencia de negocios se estableció el uso de la plataforma *Pentaho Community Edition*, debido a que contiene todas las herramientas necesarias para el desarrollo de este proyecto de tesis.
- Para este proyecto de tesis se utilizó la metodología en cascada, debido a que se basa en etapas y cada etapa debe necesariamente finalizarse antes que la siguiente etapa inicie, con esto se puede detectar cualquier problema en etapa temprana y de esta manera evitar costos y tiempo adicional en el desarrollo del proyecto.
- Con el uso de la herramienta *Pentaho Data Integration* se realizó todos los pasos necesarios para la implementación del *data mart* de ventas de Claro, desde la recopilación de información de los sistemas fuente, realizar todos los procesos *ETL* necesarios hasta la carga de todos los datos al *data mart* DM_VENTAS. Cabe mencionar que el uso de *Pentaho Data Integration* permite que todas las transformaciones mantengan la información siempre actualizada y de manera automática.

- Los aplicativos para la visualización y análisis de la información que se entregue a los usuarios finales deben de ser necesariamente de un manejo simple. La idea principal es que los usuarios en base a los indicadores claves del negocio, puedan generar inteligencia de negocio que apoye a las tomas de decisiones.
- Una vez finalizado la implementación del *data mart* para el módulo de ventas de Claro, se puede aplicar esta misma solución a otros módulos como: Renuncias, calidad de servicio, servicio técnico entre otros. De igual manera otros departamentos pueden hacer uso de esta solución para mejorar los tiempos de respuesta y tomar decisiones de manera oportuna.
- El uso de herramientas de inteligencia de negocios para este proyecto de tesis permiten administrar el tiempo y mantener un monitoreo constante sobre las ventas generadas por los centros de atención a clientes de Quito.
- Por medio del análisis costo – beneficio, claramente se pudo observar que con el uso de la herramienta de inteligencia de negocios la empresa pudo haber ahorrado dinero en la generación de los reportes de ventas. Adicionalmente es importante mencionar que esta solución es de gran ayuda y permite a los usuarios finales analizar la información de manera oportuna y actualizada, para tomar las respectivas decisiones en favor de la empresa en base a las reglas del negocio.

4.2 Recomendaciones

- Es necesario mantener constante contacto con las personas involucradas en el proceso, para así, identificar a tiempo los cambios necesarios que se deban de realizar al proyecto.
- Se recomienda al personal de sistemas mantener una mejor organización y estructura de los sistemas fuente, debido a que se detectó que se pueden realizar mejoras como por ejemplo: Definir los nombres de cada campo de los sistemas fuente de manera más precisa.
- Los usuarios finales pueden solicitar mejoras o aportar con ideas sobre la herramienta de inteligencia de negocios de *Pentaho* que se va a implementar en la empresa.
- Se recomienda el uso de una herramienta de inteligencia de negocios en los demás departamentos de Claro, de esta manera las áreas gerenciales pueden tomar las decisiones respectivas.
- Se recomienda a los usuarios finales el uso continuo de esta herramienta de inteligencia de negocios, debido a que se les va a facilitar el análisis diario sobre las ventas generadas de los centros de atención a clientes de Quito.

REFERENCIAS

- Beighley, L. (2007). *Head First SQL*. (1ª. ed.). USA: O'Reilly Media, Inc.
- Bhatnagar, V. (2014). *Data Mining and Analysis in the Engineering Field*. (1ª. ed.). USA: IGI Global.
- Bouman, R., Dongen, J. (2009). *Business Intelligence and Data Warehousing with Pentaho and MySQL*. (1ª. ed.). Indianapolis, USA: Wiley Publishing, Inc.
- Cadle, J., Ahmed, T., Cox, J., Girvan, L., Paul, A., Paul, D., y Thompson, P., (2014). *Developing Information Systems: Practical guidance for IT professionals*. (1ª. ed.). Swindon, UK: BCS Learning & Development Limited.
- Cruz, M., Moreira, F. y Varajao, J. (2013). *Handbook guide of Research on Enterprise 2.0*. (1ª. ed.). USA: IGI Global.
- Dennis, A. Wixom, B. y Roth, R. (2008). *Systems Analysis and Design*. (1ª. ed.). New Jersey, USA: John Wiley & Son.
- Ericsson, R., Cline J. (2006). *SQL Server 2005 For Developers*. (1ª. ed.). Boston, USA: Course Technology PTR.
- Gartner, (2015). Cuadrante mágico de Gartner. Recuperado el 6 de marzo del 2015 de <http://www.gartner.com/technology/reprints.do?id=1-2ADAAYM&ct=150223&st=sb>Bouman, R.
- Imhoff, C., Galemno, N. y Geiger, J. (2003). *Mastering Data Warehouse Design Relational and Dimensional Techniques*. (1ª. ed.). Indiana, USA: Wiley Publishing, Inc.
- Inmon, W. (2005). *Building the Data Warehouse*. (4ª. ed.). Indianapolis, USA: John Wiley & Son.
- Inmon, W. Strauss, D. y Neushloss, G. (2010). *DW 2.0: The Architecture for the Next Generation of Data Warehousing*. (1ª. ed.). Burlington, USA: Morgan Kaufmann.

- Kimball, R. y Caserta, J. (2004). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. (1ª. ed.). Indianapolis, USA: John Wiley & Sons.
- Kimball, R., Ross, M. (2013). *The Data Warehouse Toolkit - The Definitive to Dimensional Modeling*. (3ª. ed.). USA: John Wiley & Son.
- Kimball, R., Ross, M., Thornthwaite, W., Mundy, J., y Becker, B. (2010). *The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence*. (1ª. ed.). Indianapolis, USA: John Wiley & Sons.
- Mattío, M. Bernabeu, D. (2013). *Pentaho 5.0 Reporting by Example Beginner's Guide*. (1ª. ed.). Birmingham, UK: Packt Publishing.
- Miller, G. Brautigam, D. y Gerlach, S. (2006). *Business Intelligence Competency Centers: A Team Approach to Maximizing Competitive Advantage*. (1ª. ed.). New Jersey, USA: John Wiley & Sons.
- Mundy, J., Thornthwaite, W, y Kimball, R. (2011). *The Microsoft® Data Warehouse Toolkit: With SQL Server 2008 R2 and the Microsoft® Business Intelligence Toolset, Second Edition*. (2ª. ed.). Indianapolis, USA: John Wiley & Sons.
- Pentaho, (2014). Página oficial de Pentaho. Recuperado el 2 de febrero del 2014 de <http://www.pentaho.com>
- Pentaho, (2014). Soporte Pentaho. Recuperado el 3 de febrero del 2014 de <http://wiki.pentaho.com/display/COM/Community+Wiki+Home>
- Pentaho, (2014). Comunidad Pentaho. Recuperado el 3 de febrero del 2014 de <http://community.pentaho.com/>
- Pentaho, (2014). Soporte PDI Pentaho. Recuperado el 4 de febrero del 2014 de <http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation>
- Pentaho, (2014). Infocenter Pentaho. Recuperado el 4 de febrero del 2014 de http://infocenter.pentaho.com/help/index.jsp?topic=%2Fsupported_components%2Freference_supported_components.html
- Pulvirenti, A. Roldán, M. (2011). *Pentaho Data Integration 4 Cookbook*. (1ª. ed.). Birmingham, UK: Packt Publishing.

- Rainardi, V. (2008). *Building a Data Warehouse With Examples in SQL Server*. (1^a. ed.). New York, USA: Apress.
- Roldan, M. (2013). *Pentaho Data Integration Beginner's Guide*. (2nd. ed.). New Jersey, USA: Packt Publishing.

ANEXOS

ANEXO 1

Diccionario de datos del Reporte de líneas activas en SCT

Nombre	Tipo	Detalle
FECHA_INGRESO	DATE	Campo fecha en donde se generó una venta.
ID_SERVICIO	INTEGER	Campo número de servicio.
ASESOR	INTEGER	Campo código de vendedor del asesor.
NOMBRE_ASESOR	VARCHAR (30)	Campo nombre del asesor.
SUBPRODUCTO	VARCHAR (3)	Campo en donde indica si la línea es controlada o abierta.
OFICINA	VARCHAR (6)	Campo código del centro de atención a clientes.
DESCRIPCION_OFICINA	VARCHAR (30)	Campo nombre del centro de atención a clientes.
PLANES	VARCHAR (30)	Campo código del plan.
DESCRIPCION_PLAN	VARCHAR (30)	Campo nombre del plan.
MODELO_EQUIPO	VARCHAR (30)	Campo nombre del modelo de equipo.
EQUIPO_PROPIO	VARCHAR (6)	Campo en donde indica si el equipo es propio del cliente o no.

CEDULA	VARCHAR (13)	Campo número de cedula o RUC del cliente.
CLIENTE	VARCHAR (30)	Campo nombre del cliente.
USUARIO_AXIS	VARCHAR (15)	Campo usuario del asesor que genera la venta.
PORTABILIDAD	VARCHAR (1)	Campo en donde indica se la línea proviene de una portabilidad o no.
TELEFONO1	INTEGER	Campo número de contacto 1 del cliente.
TELEFONO2	INTEGER	Campo número de contacto 2 del cliente.
LOCALIDAD	VARCHAR (30)	Campo ubicación del centro de atención a clientes.

ANEXO 2

Diccionario de datos del Reporte Detalle de planes y códigos BP

Nombre	Tipo	Detalle
ID_PLAN	VARCHAR(30)	Campo código de plan.
DESCRIPCION	VARCHAR (30)	Campo nombre del plan.
COUTA_MENSUAL_PLAN	NUMERIC (17,2)	Campo tarifa básica mensual del plan

ANEXO 3

Diccionario de datos del modelo dimensional DM_VENTAS

Dimensión D_FECHA

La dimensión FECHA siempre va a estar implementado en un *data mart*. Por medio de D_FECHA se realiza el análisis histórico de las ventas generadas. Esta dimensión almacena las fechas en donde se generaron cada transacción por cada venta.

Nombre	Tipo	Detalle
CS_FECHA	INTEGER	Campo clave subrogada de D_FECHA.
ANIO	INTEGER	Campo año de D_FECHA.
SEMESTRE	INTEGER	Campo # de semestre de D_FECHA .
SEMESTRE_NOMBRE	VARCHAR (30)	Campo nombre del semestre de D_FECHA.
TRIMESTRE	INTEGER	Campo # de trimestre de D_FECHA.
TRIMESTRE_NOMBRE	VARCHAR (30)	Campo nombre del trimestre de D_FECHA.
MES	INTEGER	Campo # del mes de D_FECHA.
MES_NOMBRE	VARCHAR (30)	Campo nombre del mes D_FECHA.
DIA	INTEGER	Campo # del día del mes de D_FECHA.
DIA_SEMANA	VARCHAR (30)	Campo nombre del día de la semana

		D_FECHA.
FECHA	DATE	Campo fecha en donde se generó una venta.

Dimensión D_CAC

La dimensión CAC indica el cac en donde se generaron las ventas. Por medio de D_CAC se realiza un análisis de ventas por cac.

Nombre	Tipo	Detalle
CS_CAC	INTEGER	Campo clave subrogada de D_CAC.
CP_CAC	VARCHAR (50)	Campo clave primaria de D_CAC.
DESCRIPCION	VARCHAR (100)	Campo nombre completo del cac de D_CAC.
CIUDAD	VARCHAR (50)	Campo ciudad en donde se encuentra ubicado el cac de D_CAC.
VALID_FROM	DATE	Campo fecha desde cuándo es válido el registro.
VALID_TO	DATE	Campo fecha hasta cuándo es válido el registro.
VERSION	INTEGER	Versión

Dimensión D_PLAN

La dimensión PLAN indica el plan que tiene contratado el cliente, adicionalmente representa el enfoque de análisis principal para este caso de estudio. Por medio de D_PLAN se genera un análisis de ventas, de tal manera que apoye a la toma de decisiones.

Nombre	Tipo	Detalle
CS_PLAN	INTEGER	Campo clave subrogada de D_PLAN.
CP_PLAN	VARCHAR (50)	Campo clave primaria de D_PLAN.
DESCRIPCION	VARCHAR (50)	Campo nombre completo del plan de D_PLAN.
SUBPRODUCTO	VARCHAR (50)	Campo indica si el plan es abierto (TAR) o controlado (AUT) de D_PLAN.
VALID_FROM	DATE	Campo fecha desde cuándo es válido el registro.
VALID_TO	DATE	Campo fecha hasta cuándo es válido el registro.
VERSION	INTEGER	Versión

Dimensión D_CLIENTE

La dimensión CLIENTE indica que cliente realizo la contratación del plan. Por medio de D_CLIENTE se detecta a los clientes potenciales para ofrecerles otros planes o servicios adicionales.

Nombre	Tipo	Detalle
CS_CLIENTE	INTEGER	Campo clave subrogada D_CLIENTE.
CP_CEDULA	VARCHAR (13)	Campo clave primaria de la tabla de D_CLIENTE.
NOMBRES	VARCHAR (100)	Campo nombres completos del cliente de D_CLIENTE.
VALID_FROM	DATE	Campo fecha desde cuándo es válido el registro.
VALID_TO	DATE	Campo fecha hasta cuándo es válido el registro.
VERSION	INTEGER	Versión

Dimensión D_ASESOR

La dimensión ASESOR indica el asesor que gestiona y procesa la venta hacia el cliente. Por medio de D_ASESOR se analiza cuál es la productividad de cada asesor.

Nombre	Tipo	Detalle
CS_ASESOR	INTEGER	Campo clave subrogada de D_ASESOR.
CP_ASESOR	VARCHAR(15)	Campo clave primaria de la tabla de D_ASESOR.
USUARIO_AXIS	VARCHAR (50)	Campo usuario único que tiene asignado cada asesor de D_ASESOR.
VALID_FROM	DATE	Campo fecha desde cuándo es válido el registro.
VALID_TO	DATE	Campo fecha hasta cuándo es válido el registro.
VERSION	INTEGER	Versión

Tabla de hechos F_VENTAS

La tabla de hechos F_VENTAS contiene su clave subrogada, además de todas las claves foráneas de las dimensiones D_FECHA, D_CAC, D_CAC, D_PLAN, D_CLIENTE Y D_ASESOR. Esta tabla también contiene las medidas ventas y cantidad.

Nombre	Tipo	Detalle
CS_VENTAS	INTEGER	Campo clave subrogada de F_VENTAS.
CS_FECHA	INTEGER	Campo clave externa que hace referencia a D_FECHA.
CS_CAC	INTEGER	Campo clave externa que hace referencia a D_CAC.
CS_PLAN	INTEGER	Campo clave externa que hace referencia a D_PLAN.
CS_CLIENTE	INTEGER	Campo clave externa que hace referencia a D_CLIENTE.
CS_ASESOR	INTEGER	Campo clave externa que hace referencia a D_ASESOR.
VENTAS	NUMERIC(17, 2)	Campo valor correspondiente a las ventas generadas en USD.
CANTIDAD	INTEGER	Campo número de planes vendidos en USD.

ANEXO 4

Script SQL para la creación de las dimensiones D_FECHA, D_CAC, D_PLAN, D_CLIENTE, D_ASESOR y la tabla de hechos F_VENTAS.

```
CREATE TABLE d_fecha
```

```
(  
  cs_fecha INTEGER PRIMARY KEY  
  , anio INTEGER  
  , semestre INTEGER  
  , semestre_nombre VARCHAR(30)  
  , trimestre INTEGER  
  , trimestre_nombre VARCHAR(30)  
  , mes INTEGER  
  , mes_nombre VARCHAR(30)  
  , dia INTEGER  
  , dia_semana VARCHAR(30)  
  , fecha DATE  
)
```

```
;
```

```
CREATE TABLE d_cac
```

```
(  
  cs_cac INTEGER PRIMARY KEY  
  , cp_cac VARCHAR(50) NOT NULL DEFAULT '0'  
  , descripcion VARCHAR(100) NOT NULL DEFAULT 'DESCONOCIDO'  
  , valid_to DATE  
  , valid_from DATE  
  , version INTEGER NOT NULL  
)
```

```
;
```

```
CREATE UNIQUE INDEX idx_d_cac_pk  
ON d_cac
```

```
(
```

```
    cp_cac
)
;
CREATE INDEX idx_d_cac_lookup
ON d_cac
(
    cp_cac
, descripcion
, ciudad
)
;
CREATE TABLE d_plan
(
    cs_plan INTEGER PRIMARY KEY
, cp_plan VARCHAR(50) NOT NULL DEFAULT '0'
, descripcion VARCHAR(50) NOT NULL DEFAULT 'DESCONOCIDO'
, subproducto VARCHAR(50) NOT NULL DEFAULT 'DESCONOCIDO'
, valid_to DATE
, valid_from DATE
, version INTEGER NOT NULL
)
;
CREATE UNIQUE INDEX idx_d_plan_pk
ON d_plan
(
    cp_plan
)
;
CREATE INDEX idx_d_plan_lookup
ON d_plan
(
    cp_plan
```

```
, descripcion
, subproducto
)
;
CREATE TABLE d_cliente
(
  cs_cliente INTEGER PRIMARY KEY
, cp_cedula VARCHAR(13) NOT NULL DEFAULT '0'
, nombres VARCHAR(100) NOT NULL DEFAULT 'DESCONOCIDO'
, ciudad VARCHAR(50) NOT NULL DEFAULT 'DESCONOCIDO'
, valid_to DATE
, valid_from DATE
, version INTEGER NOT NULL
)
;
CREATE UNIQUE INDEX idx_d_cliente_pk
ON d_cliente
(
  cp_cedula
)
;
CREATE INDEX idx_d_cliente_lookup
ON d_cliente
(
  cp_cedula
, nombres
)
;
CREATE TABLE d_asesor
(
  cs_asesor INTEGER PRIMARY KEY
, cp_asesor VARCHAR(15) NOT NULL DEFAULT '0'
```

```
, nombres VARCHAR(100) NOT NULL DEFAULT 'DESCONOCIDO'
, usuario_axis VARCHAR(50) NOT NULL DEFAULT 'DESCONOCIDO'
, valid_to DATE
, valid_from DATE
, version INTEGER NOT NULL
)
;
CREATE UNIQUE INDEX idx_d_asesor_pk
ON d_asesor
(
  cp_asesor
)
;
CREATE INDEX idx_d_asesor_lookup
ON d_asesor
(
  cp_asesor
, nombres
, usuario_axis
)
;
CREATE TABLE F_VENTAS (
    CS_VENTAS INTEGER NOT NULL,
    CS_FECHA INTEGER NOT NULL,
    CS_CAC INTEGER NOT NULL,
    CS_PLAN INTEGER NOT NULL,
    CS_CLIENTE INTEGER NOT NULL,
    CS_ASESOR INTEGER NOT NULL,
    VENTAS NUMERIC(17,2) NOT NULL,
    CANTIDAD INTEGER NOT NULL,
    CONSTRAINT CS_VENTAS PRIMARY KEY (CS_VENTAS)
);
```



```
ALTER      TABLE      F_VENTAS      ADD      CONSTRAINT
DIM_FECHA_FACT_VENTAS_fk
FOREIGN KEY (CS_FECHA)
REFERENCES D_FECHA (CS_FECHA)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER      TABLE      F_VENTAS      ADD      CONSTRAINT
DIM_ASESOR_FACT_VENTAS_fk
FOREIGN KEY (CS_ASESOR)
REFERENCES D_ASESOR (CS_ASESOR)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER      TABLE      F_VENTAS      ADD      CONSTRAINT
DIM_CLIENTE_FACT_VENTAS_fk
FOREIGN KEY (CS_CLIENTE)
REFERENCES D_CLIENTE (CS_CLIENTE)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER      TABLE      F_VENTAS      ADD      CONSTRAINT
DIM_PLAN_FACT_VENTAS_fk
FOREIGN KEY (CS_PLAN)
REFERENCES D_PLAN (CS_PLAN)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE F_VENTAS ADD CONSTRAINT DIM_CAC_FACT_VENTAS_fk
FOREIGN KEY (CS_CAC)
REFERENCES D_CAC (CS_CAC)
ON DELETE NO ACTION
```

ON UPDATE NO ACTION
NOT DEFERRABLE;

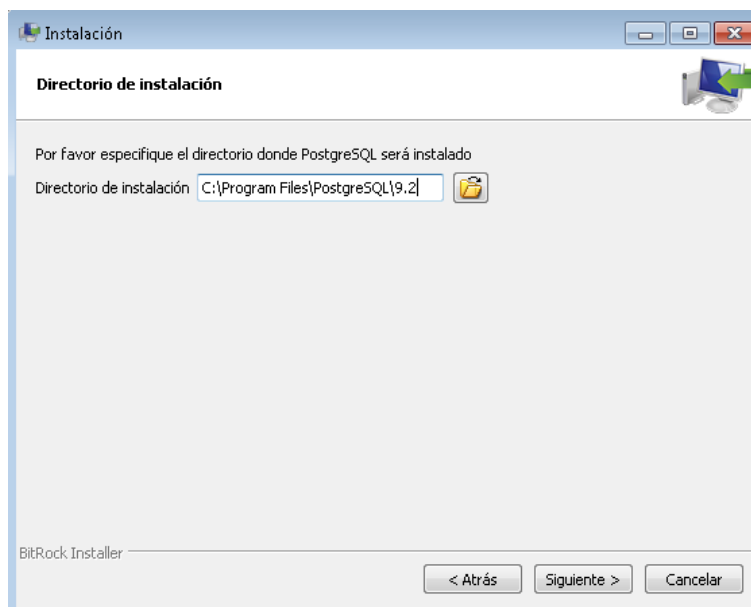
Anexo 5

Instalación y configuración del servidor de base de datos PostgreSQLv9.2

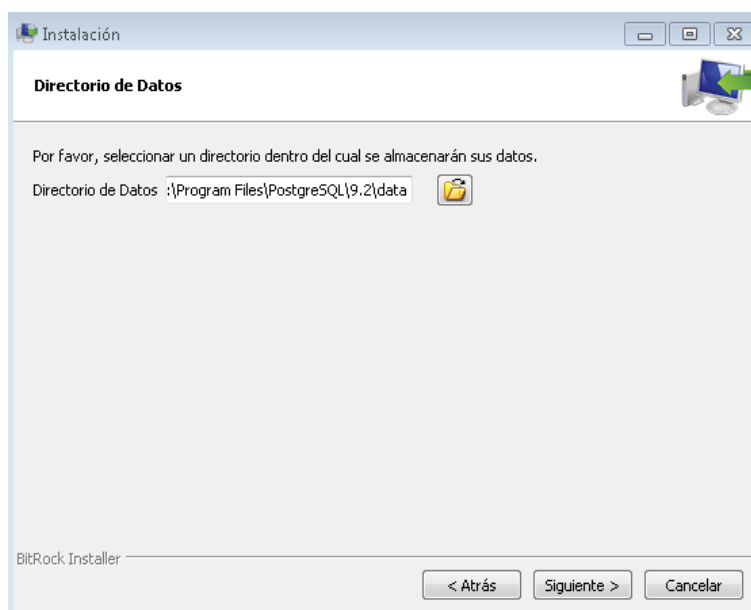
- ✓ Descargar e instalar la base de datos PostgreSQL v9.2 32 bits para Windows desde la página oficial <http://www.postgresql.org/download/windows/>.



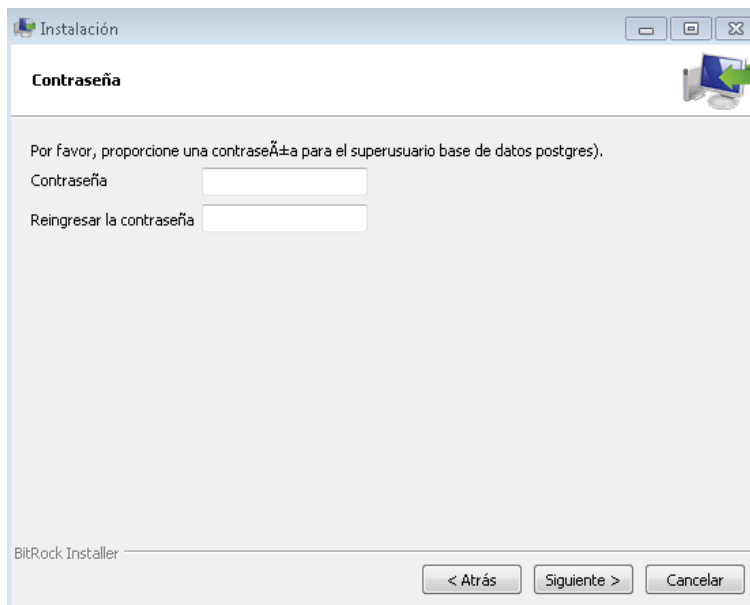
- ✓ Se especifica el directorio en donde se va a instalar PostgreSQL.
C:\Program Files\PostgreSQL\9.2



- ✓ Se especifica el directorio en donde se van a guardar los datos.
C:\Program Files\PostgreSQL\9.2\data.



- ✓ Se especifica una contraseña con privilegios de administrador para PostgreSQL.



Instalación

Contraseña

Por favor, proporcione una contraseña para el superusuario base de datos postgres).

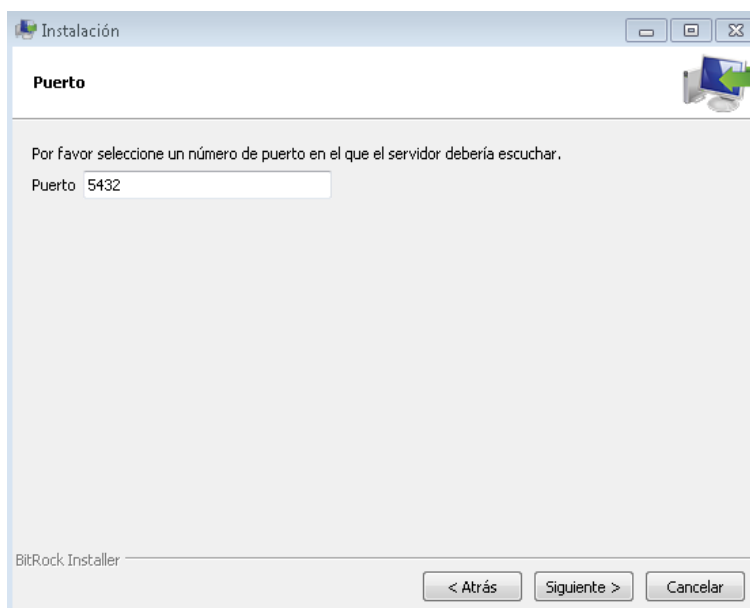
Contraseña

Reingresar la contraseña

BitRock Installer

< Atrás Siguiete > Cancelar

- ✓ Se especifica el número de puerto 5432 para que el servidor lo pueda escuchar.



Instalación

Puerto

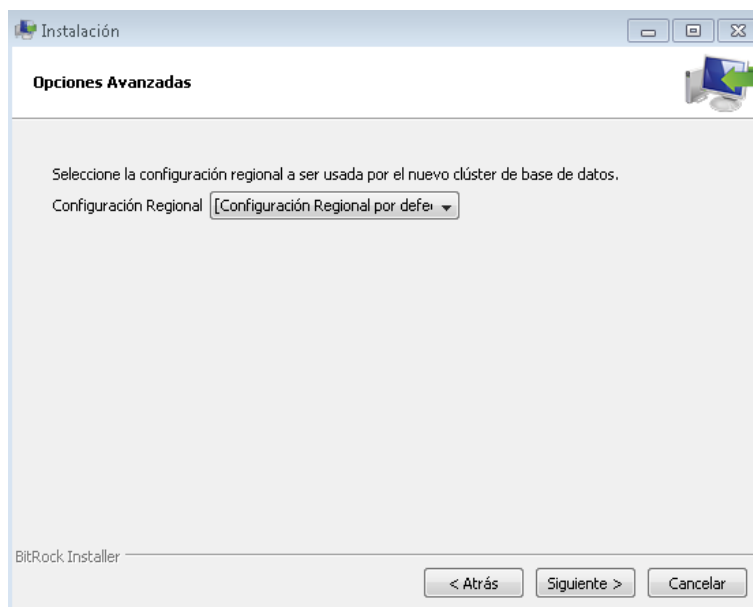
Por favor seleccione un número de puerto en el que el servidor debería escuchar.

Puerto 5432

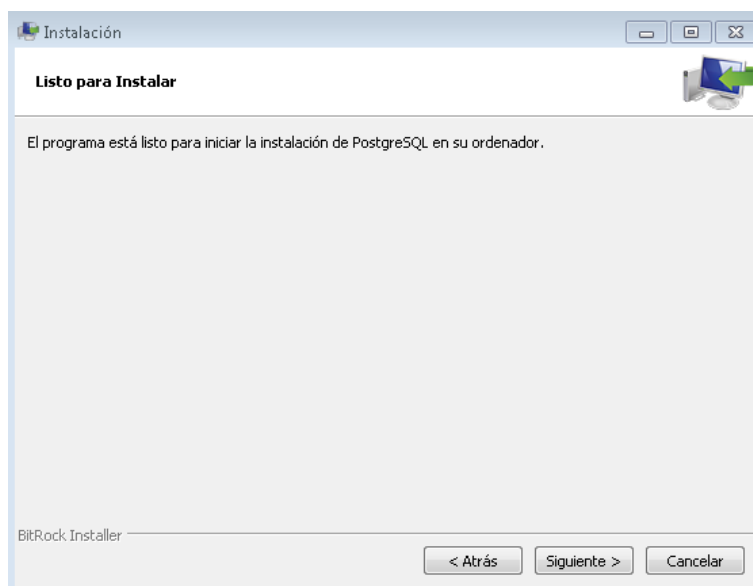
BitRock Installer

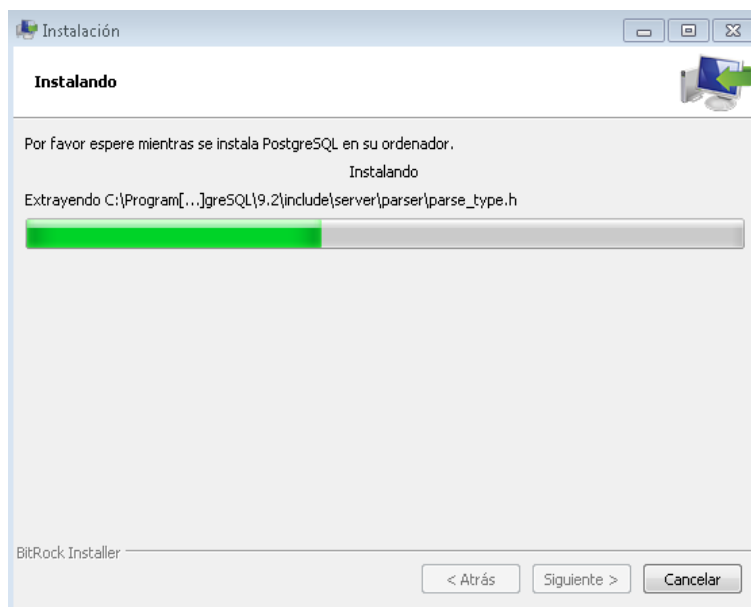
< Atrás Siguiete > Cancelar

- ✓ Se especifica la configuración regional por defecto para el cluster de la base de datos.



- ✓ Inicia la instalación.

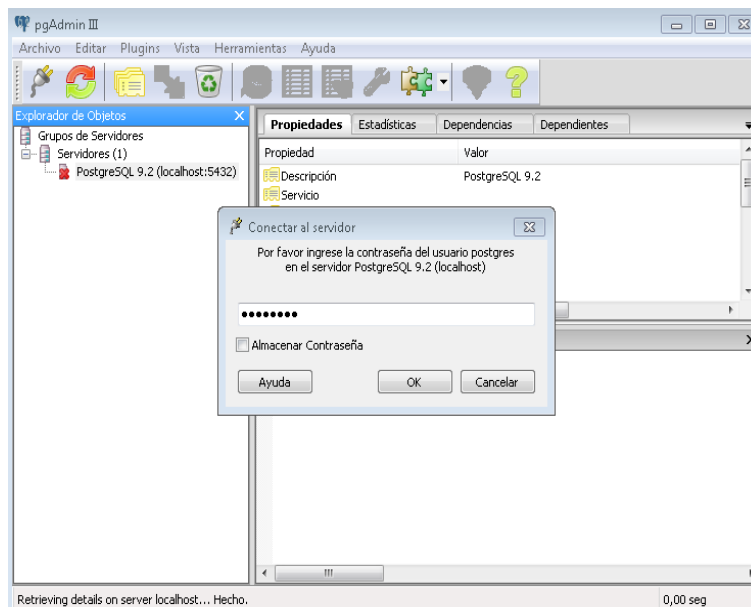




✓ Finalizó la instalación.



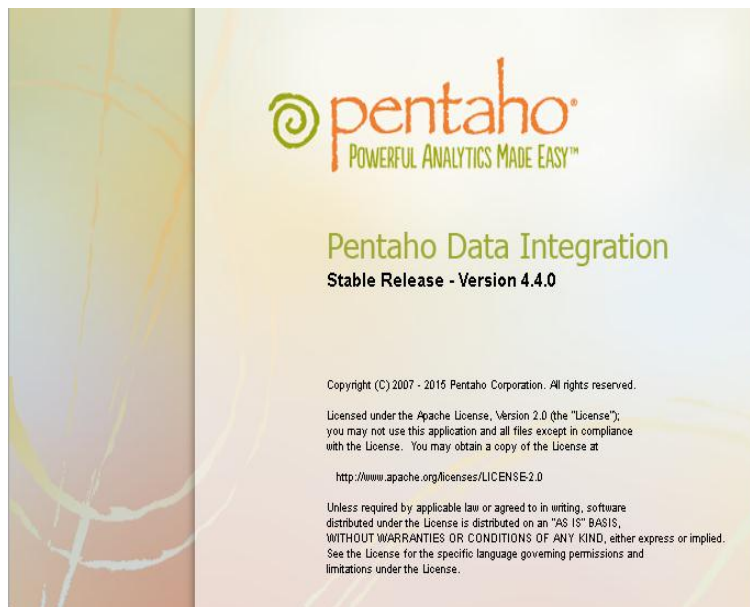
- ✓ Se ingresa la contraseña de administrador del usuario PostgreSQL.



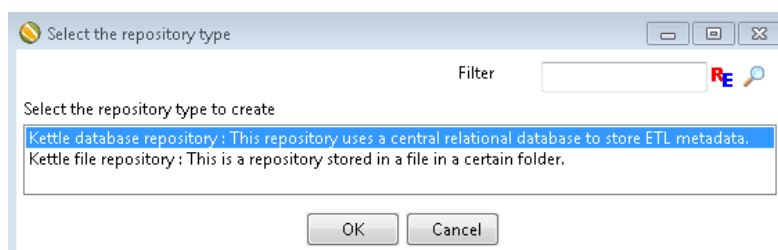
ANEXO 6

Configuración de Pentaho Data Integration v4.4.0

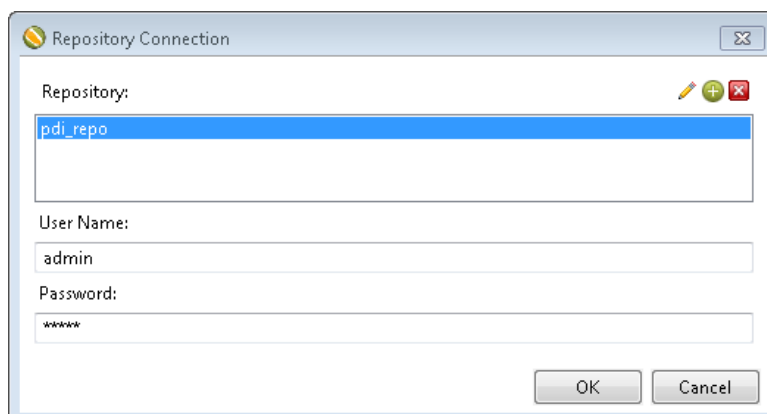
- ✓ Se ingresa a la ruta `C:\pentaho\pdi-ce-4.4.0\data-integration` y se ejecuta el archivo `Spoon.bat`, para arrancar la herramienta *PDI*.



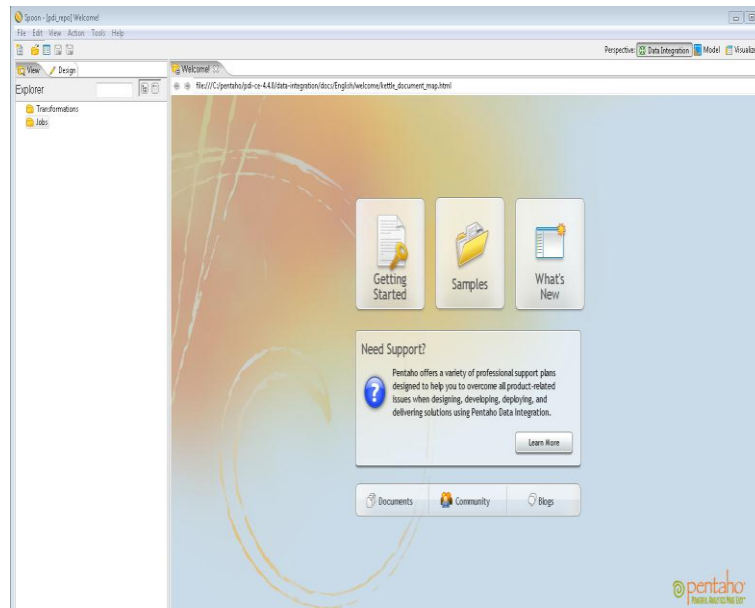
- ✓ Se selecciona el repositorio base de datos Kettle.



- ✓ Se crea el repositorio denominado PDI_REPO y se ingresa el usuario y contraseña.



✓ Interface de *Pentaho Data Integration*.



ANEXO 7

Tabulación de los resultados obtenidos:

Pregunta	Respuesta Si	Respuesta No	%
1	2	0	100
2	2	0	100
3	2	0	100
4	2	0	100
5	2	0	100

Encuestas escaneadas de los usuarios finales

Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de *Pentaho*.

Fecha: 21-OCT-2014

Nombre: EDMUNDO ARIAS

1 El uso de la consola de usuario de *Pentaho* es sencillo y amigable?

Si X No ¿Por qué?

LUEGO DE LA INDUCCIÓN DE JOHN, EL USO
FUE SENCILLO.

2 Los reportes muestran la información necesaria para poder realizar un análisis sobre el negocio?

Si X No ¿Por qué?

SI LA INFORMACIÓN QUE SE SOLICITÓ A JOHN
EN LAS EMPRESAS.

3 La información consolidada y centralizada en un solo lugar, ¿les permitirá un mejor análisis en sus reportes?

Si X No ¿Por qué?

PERMITE OBTENER UNA IDEA GLOBAL DE LA
SITUACIÓN DE LOS CAC'S Y CON LA
INFORMACIÓN ACTUALIZADA.

- 4 Según su opinión esta solución de inteligencia de negocios, ¿les disminuirá los tiempos de respuesta a las consultas de los usuarios finales?

Si X No ¿Por qué?

LA INFORMACIÓN CONSULTADA ES EN LÍNEA.

- 5 Luego de haber probado la aplicación, ¿la seguiría utilizando?

Si X No ¿Por qué?

PERMITE TOMAR DECISIONES DE MANERA INMEDIATA

- 6 Mencione algunas ventajas y desventajas del aplicativo, para el desarrollo de su trabajo.

VENTAJAS: LA INFORMACIÓN SE ENCUENTRA
ACTUALIZADA Y EN LÍNEA.

Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de Pentaho.

Fecha: 21- Octubre - 2014

Nombre: Jamiera Robles

1 El uso de la consola de usuario de Pentaho es sencillo y amigable?

Si No ¿Por qué?

El acceso a la información es ordenado y se puede visualizar los reportes requeridos.

2 Los reportes muestran la información necesaria para poder realizar un análisis sobre el negocio?

Si No ¿Por qué?

Por medio de los gráficos se puede ver claramente y de manera general los principales índices sobre las ventas.

3 La información consolidada y centralizada en un solo lugar, ¿les permitirá un mejor análisis en sus reportes?

Si No ¿Por qué?

Se tiene acceso a toda la información sobre las ventas en un solo lugar, de esta manera se logra optimizar tiempo.

- 4 Según su opinión esta solución de inteligencia de negocios, ¿les disminuirá los tiempos de respuesta a las consultas de los usuarios finales?

Si No ¿Por qué?

Actualmente el reporte de ventas se genera hasta 2 días laborables, pero con esta solución la información se obtiene en línea y de manera actualizada

- 5 Luego de haber probado la aplicación, ¿la seguiría utilizando?

Si No ¿Por qué?

Este sistema me ayuda a tomar decisiones diariamente.

- 6 Mencione algunas ventajas y desventajas del aplicativo, para el desarrollo de su trabajo.

Ventaja: Información consolidada y en línea.

Desventaja: Se sugiere que se incluya un módulo para poder visualizar las renuncias y calidad de servicio de los Cac's.

Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de *Pentaho*.

Fecha: 18/03/2015

Nombre: JORGE PATIÑO

1 El uso de la consola de usuario de *Pentaho* es sencillo y amigable?

Si No ¿Por qué?

LA INTERFACE ES MUY AMIGABLE.

2 Los reportes muestran la información necesaria para poder realizar un análisis sobre el negocio?

Si No ¿Por qué?

SE PUEDE VISUALIZAR LAS VENTAS POR CAC, PRODUCTO, ASESOR.

3 La información consolidada y centralizada en un solo lugar, ¿les permitirá un mejor análisis en sus reportes?

Si No ¿Por qué?

TEMPO ACCESS A LA INFORMACIÓN EN CUALQUIER MOMENTO Y EN UN SOLO LUGAR.

- 4 Según su opinión esta solución de inteligencia de negocios, ¿les disminuirá los tiempos de respuesta a las consultas de los usuarios finales?

Si No ¿Por qué?

EL REPORTE LO PUEDO VER EN LINEA, SIN LA NECESIDAD DE SOLICITARLE AL ANALISTA DE INFORMACIÓN (FELIX MIRANDA)

- 5 Luego de haber probado la aplicación, ¿la seguiría utilizando?

Si No ¿Por qué?

PUEDO REVISAR LAS VENTAS DEL CAZ A DIARIO.

- 6 Mencione algunas ventajas y desventajas del aplicativo, para el desarrollo de su trabajo.

COMO SUGERENCIA SERIA INTERESANTE SI ESTA IMPORTACION NOS LLEVARIA AL CELULAR/TABLET.

Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de *Pentaho*.

Fecha: 18-03-2015

Nombre: VICTORIA PACHECO

1 El uso de la consola de usuario de *Pentaho* es sencillo y amigable?

Si X No ¿Por qué?

SE OBTIENE LA INFORMACION DE MANERA RAPIDA
y FACIL.

2 Los reportes muestran la información necesaria para poder realizar un análisis sobre el negocio?

Si X No ¿Por qué?

LOS REPORTES ME INDICAN LA INFORMACION
REQUERIDA.

3 La información consolidada y centralizada en un solo lugar, ¿les permitirá un mejor análisis en sus reportes?

Si X No ¿Por qué?

ME AHORRA TIEMPO EN BUSCAR EN MI CORREO
LOS REPORTES DE VENTAS, PERO CON ESTA
APLICACION, LA INFORMACION SE ENCUENTRA
CONSOLIDADA

- 4 Según su opinión esta solución de inteligencia de negocios, ¿les disminuirá los tiempos de respuesta a las consultas de los usuarios finales?

Si X No ¿Por qué?

SE PUEDE VISUALIZAR LA INFORMACION DE MANERA INMEDIATA.

- 5 Luego de haber probado la aplicación, ¿la seguiría utilizando?

Si X No ¿Por qué?

DEFINITIVAMENTE, PORQUE ME AHORRA TIEMPO.

- 6 Mencione algunas ventajas y desventajas del aplicativo, para el desarrollo de su trabajo.

VENTAJA: PUEDO REVISAR LA INFORMACION A DIARIO.

Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de Pentaho.

Fecha: 19/03/2015

Nombre: Juan Manuel Ossa

1 El uso de la consola de usuario de Pentaho es sencillo y amigable?

Si No ¿Por qué?

Sí debido a que se puede visualizar la información de ventas de una manera rápida y la interface es bastante amigable.

2 Los reportes muestran la información necesaria para poder realizar un análisis sobre el negocio?

Si No ¿Por qué?

Comparado al reporte de ventas que nos envía el analista de información, este reporte por medio de Pentaho facilita la visualización de los índices de ventas.

3 La información consolidada y centralizada en un solo lugar, ¿les permitirá un mejor análisis en sus reportes?

Si No ¿Por qué?

Actualmente los reportes de ventas no se encuentran consolidados en un sólo lugar, esta aplicación ofrece acceso centralizado a la información.

- 4 Según su opinión esta solución de inteligencia de negocios, ¿les disminuirá los tiempos de respuesta a las consultas de los usuarios finales?

Si No ¿Por qué?

Actualmente el analista de información envía el reporte de ventas una vez por semana, pero con esta solución la información va a estar disponible en cualquier momento.

- 5 Luego de haber probado la aplicación, ¿la seguiría utilizando?

Si No ¿Por qué?

Esta herramienta es de mucha utilidad para las decisiones que requiero tomar diariamente.

- 6 Mencione algunas ventajas y desventajas del aplicativo, para el desarrollo de su trabajo.

La ventaja principal es obtener la información sobre las ventas, en tiempo real y de manera consolidada. Sería importante incluir una solución para obtener información sobre otros ítem como calidad de servicio, renuncias, servicio técnico, inventario.

Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de Pentaho.

Fecha: 20/MARZO/15.

Nombre: DAVID LOPEZ.

1 El uso de la consola de usuario de Pentaho es sencillo y amigable?

Si No ¿Por qué?

SÍ ES AMIGABLE.

2 Los reportes muestran la información necesaria para poder realizar un análisis sobre el negocio?

Si No ¿Por qué?

SÍ PORQUE PUEDO VER LAS VENTAS
POR AGENCIA, POR ASESOR, POR
PRODUCTO.

3 La información consolidada y centralizada en un solo lugar, ¿les permitirá un mejor análisis en sus reportes?

Si No ¿Por qué?

SÍ PORQUE PUEDO REVISAR LAS VENTAS
EN LINEA.

- 4 Según su opinión esta solución de inteligencia de negocios, ¿les disminuirá los tiempos de respuesta a las consultas de los usuarios finales?

Si No ¿Por qué?

LA INFORMACIÓN SE ACCEDERÍA DE MANERA
INMEDIATA.

- 5 Luego de haber probado la aplicación, ¿la seguiría utilizando?

Si No ¿Por qué?

POEDO REALIZAR UN ANALISIS DE
MANERA RAPIDA.

- 6 Mencione algunas ventajas y desventajas del aplicativo, para el desarrollo de su trabajo.

UNA VENTAJA ES QUE PUEDO REVISAR
LAS VENTAS EN LINEA.

Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de *Pentaho*.

Fecha: Marzo 20, 2015

Nombre: Fabian Hidalgo

1 El uso de la consola de usuario de *Pentaho* es sencillo y amigable?

Si No ¿Por qué?

Se obtiene las ventas desde la consola de una forma rápida y sencilla.

2 Los reportes muestran la información necesaria para poder realizar un análisis sobre el negocio?

Si No ¿Por qué?

Esta aplicación me permite revisar las ventas y los indicadores necesarios.

3 La información consolidada y centralizada en un solo lugar, ¿les permitirá un mejor análisis en sus reportes?

Si No ¿Por qué?

Obtengo los indicadores de ventas en tiempo real y de una sola fuente.

- 4 Según su opinión esta solución de inteligencia de negocios, ¿les disminuirá los tiempos de respuesta a las consultas de los usuarios finales?

Si No ¿Por qué?

Debido a que la información se encuentra disponible en línea.

- 5 Luego de haber probado la aplicación, ¿la seguiría utilizando?

Si No ¿Por qué?

Puede revisar los indicadores a diario y tomar decisiones en base a éstos.

- 6 Mencione algunas ventajas y desventajas del aplicativo, para el desarrollo de su trabajo.

Como ventaja se puede mencionar que esta aplicación nos ayuda diariamente a generar mayores ventas.

Banco de preguntas a los usuarios finales sobre el uso de la consola de usuario de Pentaho.

Fecha: 20/MARZO/15.

Nombre: DAVID LOPEZ.

- 1 El uso de la consola de usuario de Pentaho es sencillo y amigable?

Si No ¿Por qué?

SÍ ES AMIGABLE.

- 2 Los reportes muestran la información necesaria para poder realizar un análisis sobre el negocio?

Si No ¿Por qué?

SÍ, PORQUE PUEDO VER LAS VENTAS
POR AGENCIA, POR ASESOR, POR
PRODUCTO.

- 3 La información consolidada y centralizada en un solo lugar, ¿les permitirá un mejor análisis en sus reportes?

Si No ¿Por qué?

SÍ PORQUE PUEDO REVISAR LAS VENTAS
EN LÍNEA.

- 4 Según su opinión esta solución de inteligencia de negocios, ¿les disminuirá los tiempos de respuesta a las consultas de los usuarios finales?

Si No ¿Por qué?

LA INFORMACIÓN SE ACCEDERÍA DE MANERA INMEDIATA.

- 5 Luego de haber probado la aplicación, ¿la seguiría utilizando?

Si No ¿Por qué?

PUEDO REALIZAR UN ANÁLISIS DE MANERA RÁPIDA.

- 6 Mencione algunas ventajas y desventajas del aplicativo, para el desarrollo de su trabajo.

UNA VENTAJA ES QUE PUEDO REVISAR LAS VENTAS EN LÍNEA.

ANEXO 8

Código SQL de los ETL's

ETLs (Fuente al *stage*), Transformaciones y JOBS

Transformación TRS_CARGA_ARCHIVOS_FUENTE_CSV

-- Step : Salida_archivos_csv

-- Database Connection : stage

-- SQL :

```
ALTER TABLE stg_archivos_ventas ADD COLUMN FECHA_INGRESO_KTL
TEXT;
```

```
UPDATE stg_archivos_ventas SET
FECHA_INGRESO_KTL=FECHA_INGRESO;
```

```
ALTER TABLE stg_archivos_ventas DROP COLUMN FECHA_INGRESO;
```

```
ALTER TABLE stg_archivos_ventas RENAME FECHA_INGRESO_KTL TO
FECHA_INGRESO;
```

;

```
ALTER TABLE stg_archivos_ventas ADD COLUMN ID_SERVICIO_KTL TEXT;
```

```
UPDATE stg_archivos_ventas SET ID_SERVICIO_KTL=ID_SERVICIO;
```

```
ALTER TABLE stg_archivos_ventas DROP COLUMN ID_SERVICIO;
```

```
ALTER TABLE stg_archivos_ventas RENAME ID_SERVICIO_KTL TO
ID_SERVICIO;
```

;

```
ALTER TABLE stg_archivos_ventas ADD COLUMN ASESOR_KTL TEXT;
```

```
UPDATE stg_archivos_ventas SET ASESOR_KTL=ASESOR;
```

```
ALTER TABLE stg_archivos_ventas DROP COLUMN ASESOR;
```

```
ALTER TABLE stg_archivos_ventas RENAME ASESOR_KTL TO ASESOR;
```

;

```
ALTER TABLE stg_archivos_ventas ADD COLUMN NOMBRE_ASESOR_KTL
TEXT;
```

```
UPDATE stg_archivos_ventas SET
NOMBRE_ASESOR_KTL=NOMBRE_ASESOR;
```

```
ALTER TABLE stg_archivos_ventas DROP COLUMN NOMBRE_ASESOR;
```

```
ALTER TABLE stg_archivos_ventas RENAME NOMBRE_ASESOR_KTL TO
NOMBRE_ASESOR;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN SUBPRODUCTO_KTL
TEXT;
UPDATE stg_archivos_ventas SET SUBPRODUCTO_KTL=SUBPRODUCTO;
ALTER TABLE stg_archivos_ventas DROP COLUMN SUBPRODUCTO;
ALTER TABLE stg_archivos_ventas RENAME SUBPRODUCTO_KTL TO
SUBPRODUCTO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN OFICINA_KTL TEXT;
UPDATE stg_archivos_ventas SET OFICINA_KTL=OFICINA;
ALTER TABLE stg_archivos_ventas DROP COLUMN OFICINA;
ALTER TABLE stg_archivos_ventas RENAME OFICINA_KTL TO OFICINA;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN
DESCRIPCION_OFICINA_KTL TEXT;
UPDATE stg_archivos_ventas SET
DESCRIPCION_OFICINA_KTL=DESCRIPCION_OFICINA;
ALTER TABLE stg_archivos_ventas DROP COLUMN
DESCRIPCION_OFICINA;
ALTER TABLE stg_archivos_ventas RENAME DESCRIPCION_OFICINA_KTL
TO DESCRIPCION_OFICINA;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN PLANES_KTL TEXT;
UPDATE stg_archivos_ventas SET PLANES_KTL=PLANES;
ALTER TABLE stg_archivos_ventas DROP COLUMN PLANES;
ALTER TABLE stg_archivos_ventas RENAME PLANES_KTL TO PLANES;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN
DESCRIPCION_PLAN_KTL TEXT;
```

```
UPDATE          stg_archivos_ventas          SET
DESCRIPCION_PLAN_KTL=DESCRIPCION_PLAN;
ALTER TABLE stg_archivos_ventas DROP COLUMN DESCRIPCION_PLAN;
ALTER TABLE stg_archivos_ventas RENAME DESCRIPCION_PLAN_KTL TO
DESCRIPCION_PLAN;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN MODELO_EQUIPO_KTL
TEXT;
UPDATE          stg_archivos_ventas          SET
MODELO_EQUIPO_KTL=MODELO_EQUIPO;
ALTER TABLE stg_archivos_ventas DROP COLUMN MODELO_EQUIPO;
ALTER TABLE stg_archivos_ventas RENAME MODELO_EQUIPO_KTL TO
MODELO_EQUIPO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN EQUIPO_PROPIO_KTL
TEXT;
UPDATE          stg_archivos_ventas          SET
EQUIPO_PROPIO_KTL=EQUIPO_PROPIO;
ALTER TABLE stg_archivos_ventas DROP COLUMN EQUIPO_PROPIO;
ALTER TABLE stg_archivos_ventas RENAME EQUIPO_PROPIO_KTL TO
EQUIPO_PROPIO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN CEDULA_KTL TEXT;
UPDATE stg_archivos_ventas SET CEDULA_KTL=CEDULA;
ALTER TABLE stg_archivos_ventas DROP COLUMN CEDULA;
ALTER TABLE stg_archivos_ventas RENAME CEDULA_KTL TO CEDULA;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN CLIENTE_KTL TEXT;
UPDATE stg_archivos_ventas SET CLIENTE_KTL=CLIENTE;
ALTER TABLE stg_archivos_ventas DROP COLUMN CLIENTE;
ALTER TABLE stg_archivos_ventas RENAME CLIENTE_KTL TO CLIENTE;
;
```

```
ALTER TABLE stg_archivos_ventas ADD COLUMN USUARIO_AXIS_KTL
TEXT;
UPDATE stg_archivos_ventas SET USUARIO_AXIS_KTL=USUARIO_AXIS;
ALTER TABLE stg_archivos_ventas DROP COLUMN USUARIO_AXIS;
ALTER TABLE stg_archivos_ventas RENAME USUARIO_AXIS_KTL TO
USUARIO_AXIS;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN PORTABILIDAD_KTL
TEXT;
UPDATE stg_archivos_ventas SET PORTABILIDAD_KTL=PORTABILIDAD;
ALTER TABLE stg_archivos_ventas DROP COLUMN PORTABILIDAD;
ALTER TABLE stg_archivos_ventas RENAME PORTABILIDAD_KTL TO
PORTABILIDAD;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN TELEFONO1_KTL TEXT;
UPDATE stg_archivos_ventas SET TELEFONO1_KTL=TELEFONO1;
ALTER TABLE stg_archivos_ventas DROP COLUMN TELEFONO1;
ALTER TABLE stg_archivos_ventas RENAME TELEFONO1_KTL TO
TELEFONO1;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN TELEFONO2_KTL TEXT;
UPDATE stg_archivos_ventas SET TELEFONO2_KTL=TELEFONO2;
ALTER TABLE stg_archivos_ventas DROP COLUMN TELEFONO2;
ALTER TABLE stg_archivos_ventas RENAME TELEFONO2_KTL TO
TELEFONO2;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN LOCALIDAD_KTL TEXT;
UPDATE stg_archivos_ventas SET LOCALIDAD_KTL=LOCALIDAD;
ALTER TABLE stg_archivos_ventas DROP COLUMN LOCALIDAD;
ALTER TABLE stg_archivos_ventas RENAME LOCALIDAD_KTL TO
LOCALIDAD;
;
```

Transformación TRS_LIMPIEZA_DATOS_FUENTE

```
-- Step          : Salida_stg_archivos_ventas_datos_limpiados
```

```
-- Database Connection : stage
```

```
-- SQL          :
```

```
ALTER TABLE stg_archivos_ventas_datos_limpiados ADD COLUMN
cedula_KTL TEXT;
```

```
UPDATE stg_archivos_ventas_datos_limpiados SET cedula_KTL=cedula;
```

```
ALTER TABLE stg_archivos_ventas_datos_limpiados DROP COLUMN cedula;
```

```
ALTER TABLE stg_archivos_ventas_datos_limpiados RENAME cedula_KTL
TO cedula;
```

```
;
```

Transformación TRS_CARGA_ARCHIVO_FUENTE_XLS

```
-- Step          : Salida_archivo_xls
```

```
-- Database Connection : stage
```

```
-- SQL          :
```

```
ALTER TABLE stg_planes ADD COLUMN ID_PLAN_KTL TEXT;
```

```
UPDATE stg_planes SET ID_PLAN_KTL=ID_PLAN;
```

```
ALTER TABLE stg_planes DROP COLUMN ID_PLAN;
```

```
ALTER TABLE stg_planes RENAME ID_PLAN_KTL TO ID_PLAN;
```

```
;
```

```
ALTER TABLE stg_planes ADD COLUMN DESCRIPCION_KTL TEXT;
```

```
UPDATE stg_planes SET DESCRIPCION_KTL=DESCRIPCION;
```

```
ALTER TABLE stg_planes DROP COLUMN DESCRIPCION;
```

```
ALTER TABLE stg_planes RENAME DESCRIPCION_KTL TO DESCRIPCION;
```

```
;
```

```
ALTER TABLE stg_planes ADD COLUMN CUOTA_MENSUAL_PLAN_KTL
DOUBLE PRECISION;
```

```
UPDATE          stg_planes          SET
```

```
CUOTA_MENSUAL_PLAN_KTL=CUOTA_MENSUAL_PLAN;
ALTER TABLE stg_planes DROP COLUMN CUOTA_MENSUAL_PLAN;
ALTER TABLE stg_planes RENAME CUOTA_MENSUAL_PLAN_KTL TO
CUOTA_MENSUAL_PLAN;
;

JOB_CARGA_STAGE
-- Step          : Salida_archivos_csv
-- Database Connection : stage
-- SQL           : ALTER TABLE stg_archivos_ventas ADD COLUMN
FECHA_INGRESO_KTL TEXT;
UPDATE          stg_archivos_ventas          SET
FECHA_INGRESO_KTL=FECHA_INGRESO;
ALTER TABLE stg_archivos_ventas DROP COLUMN FECHA_INGRESO;
ALTER TABLE stg_archivos_ventas RENAME FECHA_INGRESO_KTL TO
FECHA_INGRESO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN ID_SERVICIO_KTL TEXT;
UPDATE stg_archivos_ventas SET ID_SERVICIO_KTL=ID_SERVICIO;
ALTER TABLE stg_archivos_ventas DROP COLUMN ID_SERVICIO;
ALTER TABLE stg_archivos_ventas RENAME ID_SERVICIO_KTL TO
ID_SERVICIO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN ASESOR_KTL TEXT;
UPDATE stg_archivos_ventas SET ASESOR_KTL=ASESOR;
ALTER TABLE stg_archivos_ventas DROP COLUMN ASESOR;
ALTER TABLE stg_archivos_ventas RENAME ASESOR_KTL TO ASESOR;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN NOMBRE_ASESOR_KTL
TEXT;
UPDATE          stg_archivos_ventas          SET
NOMBRE_ASESOR_KTL=NOMBRE_ASESOR;
```

```
ALTER TABLE stg_archivos_ventas DROP COLUMN NOMBRE_ASESOR;
ALTER TABLE stg_archivos_ventas RENAME NOMBRE_ASESOR_KTL TO
NOMBRE_ASESOR;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN SUBPRODUCTO_KTL
TEXT;
UPDATE stg_archivos_ventas SET SUBPRODUCTO_KTL=SUBPRODUCTO;
ALTER TABLE stg_archivos_ventas DROP COLUMN SUBPRODUCTO;
ALTER TABLE stg_archivos_ventas RENAME SUBPRODUCTO_KTL TO
SUBPRODUCTO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN OFICINA_KTL TEXT;
UPDATE stg_archivos_ventas SET OFICINA_KTL=OFICINA;
ALTER TABLE stg_archivos_ventas DROP COLUMN OFICINA;
ALTER TABLE stg_archivos_ventas RENAME OFICINA_KTL TO OFICINA;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN
DESCRIPCION_OFICINA_KTL TEXT;
UPDATE stg_archivos_ventas SET
DESCRIPCION_OFICINA_KTL=DESCRIPCION_OFICINA;
ALTER TABLE stg_archivos_ventas DROP COLUMN
DESCRIPCION_OFICINA;
ALTER TABLE stg_archivos_ventas RENAME DESCRIPCION_OFICINA_KTL
TO DESCRIPCION_OFICINA;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN PLANES_KTL TEXT;
UPDATE stg_archivos_ventas SET PLANES_KTL=PLANES;
ALTER TABLE stg_archivos_ventas DROP COLUMN PLANES;
ALTER TABLE stg_archivos_ventas RENAME PLANES_KTL TO PLANES;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN
DESCRIPCION_PLAN_KTL TEXT;
```



```
UPDATE          stg_archivos_ventas          SET
DESCRIPCION_PLAN_KTL=DESCRIPCION_PLAN;
ALTER TABLE stg_archivos_ventas DROP COLUMN DESCRIPCION_PLAN;
ALTER TABLE stg_archivos_ventas RENAME DESCRIPCION_PLAN_KTL TO
DESCRIPCION_PLAN;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN MODELO_EQUIPO_KTL
TEXT;
UPDATE          stg_archivos_ventas          SET
MODELO_EQUIPO_KTL=MODELO_EQUIPO;
ALTER TABLE stg_archivos_ventas DROP COLUMN MODELO_EQUIPO;
ALTER TABLE stg_archivos_ventas RENAME MODELO_EQUIPO_KTL TO
MODELO_EQUIPO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN EQUIPO_PROPIO_KTL
TEXT;
UPDATE          stg_archivos_ventas          SET
EQUIPO_PROPIO_KTL=EQUIPO_PROPIO;
ALTER TABLE stg_archivos_ventas DROP COLUMN EQUIPO_PROPIO;
ALTER TABLE stg_archivos_ventas RENAME EQUIPO_PROPIO_KTL TO
EQUIPO_PROPIO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN CEDULA_KTL TEXT;
UPDATE stg_archivos_ventas SET CEDULA_KTL=CEDULA;
ALTER TABLE stg_archivos_ventas DROP COLUMN CEDULA;
ALTER TABLE stg_archivos_ventas RENAME CEDULA_KTL TO CEDULA;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN CLIENTE_KTL TEXT;
UPDATE stg_archivos_ventas SET CLIENTE_KTL=CLIENTE;
ALTER TABLE stg_archivos_ventas DROP COLUMN CLIENTE;
ALTER TABLE stg_archivos_ventas RENAME CLIENTE_KTL TO CLIENTE;
;
```

```
ALTER TABLE stg_archivos_ventas ADD COLUMN USUARIO_AXIS_KTL
TEXT;
UPDATE stg_archivos_ventas SET USUARIO_AXIS_KTL=USUARIO_AXIS;
ALTER TABLE stg_archivos_ventas DROP COLUMN USUARIO_AXIS;
ALTER TABLE stg_archivos_ventas RENAME USUARIO_AXIS_KTL TO
USUARIO_AXIS;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN PORTABILIDAD_KTL
TEXT;
UPDATE stg_archivos_ventas SET PORTABILIDAD_KTL=PORTABILIDAD;
ALTER TABLE stg_archivos_ventas DROP COLUMN PORTABILIDAD;
ALTER TABLE stg_archivos_ventas RENAME PORTABILIDAD_KTL TO
PORTABILIDAD;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN TELEFONO1_KTL TEXT;
UPDATE stg_archivos_ventas SET TELEFONO1_KTL=TELEFONO1;
ALTER TABLE stg_archivos_ventas DROP COLUMN TELEFONO1;
ALTER TABLE stg_archivos_ventas RENAME TELEFONO1_KTL TO
TELEFONO1;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN TELEFONO2_KTL TEXT;
UPDATE stg_archivos_ventas SET TELEFONO2_KTL=TELEFONO2;
ALTER TABLE stg_archivos_ventas DROP COLUMN TELEFONO2;
ALTER TABLE stg_archivos_ventas RENAME TELEFONO2_KTL TO
TELEFONO2;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN LOCALIDAD_KTL TEXT;
UPDATE stg_archivos_ventas SET LOCALIDAD_KTL=LOCALIDAD;
ALTER TABLE stg_archivos_ventas DROP COLUMN LOCALIDAD;
ALTER TABLE stg_archivos_ventas RENAME LOCALIDAD_KTL TO
LOCALIDAD;
;
```

```
-----  
-- Step          : Salida_stg_archivos_ventas_datos_limpiados  
-- Database Connection : stage  
-- SQL           : ALTER TABLE stg_archivos_ventas_datos_limpiados ADD  
COLUMN cedula_KTL TEXT;  
UPDATE stg_archivos_ventas_datos_limpiados SET cedula_KTL=cedula;  
ALTER TABLE stg_archivos_ventas_datos_limpiados DROP COLUMN cedula;  
ALTER TABLE stg_archivos_ventas_datos_limpiados RENAME cedula_KTL  
TO cedula;  
;  
-----  
-- Step          : Salida_archivo_xls  
-- Database Connection : stage  
-- SQL           : ALTER TABLE stg_planes ADD COLUMN ID_PLAN_KTL  
TEXT;  
UPDATE stg_planes SET ID_PLAN_KTL=ID_PLAN;  
ALTER TABLE stg_planes DROP COLUMN ID_PLAN;  
ALTER TABLE stg_planes RENAME ID_PLAN_KTL TO ID_PLAN;  
;  
  
ALTER TABLE stg_planes ADD COLUMN DESCRIPCION_KTL TEXT;  
UPDATE stg_planes SET DESCRIPCION_KTL=DESCRIPCION;  
ALTER TABLE stg_planes DROP COLUMN DESCRIPCION;  
ALTER TABLE stg_planes RENAME DESCRIPCION_KTL TO DESCRIPCION;  
;  
ALTER TABLE stg_planes ADD COLUMN CUOTA_MENSUAL_PLAN_KTL  
DOUBLE PRECISION;  
UPDATE          stg_planes          SET  
CUOTA_MENSUAL_PLAN_KTL=CUOTA_MENSUAL_PLAN;  
ALTER TABLE stg_planes DROP COLUMN CUOTA_MENSUAL_PLAN;  
ALTER TABLE stg_planes RENAME CUOTA_MENSUAL_PLAN_KTL TO  
CUOTA_MENSUAL_PLAN;
```

;

ETLs (stage al data mart), carga de tablas de dimensiones y de hechos

Transformación TRS_D_FECHA

-- Step : Salida_fecha_carga_dm_ventas

-- Database Connection : dm_ventas

-- SQL :

ALTER TABLE d_fecha ADD COLUMN fecha_inicial TIMESTAMP

;

ALTER TABLE d_fecha ADD COLUMN numero_dias DOUBLE PRECISION

;

ALTER TABLE d_fecha ADD COLUMN fecha_aux TIMESTAMP

;

ALTER TABLE d_fecha ADD COLUMN fecha_aux_string TEXT

;

ALTER TABLE d_fecha ADD COLUMN semestre_numero VARCHAR(1)

;

ALTER TABLE d_fecha ADD COLUMN dia_semana_1 VARCHAR(9)

;

ALTER TABLE d_fecha ADD COLUMN cs_fecha_KTL DOUBLE PRECISION;

UPDATE d_fecha SET cs_fecha_KTL=cs_fecha;

ALTER TABLE d_fecha DROP COLUMN cs_fecha;

ALTER TABLE d_fecha RENAME cs_fecha_KTL TO cs_fecha;

;

ALTER TABLE d_fecha ADD COLUMN anio_KTL DOUBLE PRECISION;

UPDATE d_fecha SET anio_KTL=anio;

ALTER TABLE d_fecha DROP COLUMN anio;

ALTER TABLE d_fecha RENAME anio_KTL TO anio;

;

ALTER TABLE d_fecha ADD COLUMN semestre_KTL DOUBLE PRECISION;

UPDATE d_fecha SET semestre_KTL=semestre;

ALTER TABLE d_fecha DROP COLUMN semestre;

```
ALTER TABLE d_fecha RENAME semestre_KTL TO semestre;
;
ALTER TABLE d_fecha ADD COLUMN trimestre_KTL DOUBLE PRECISION;
UPDATE d_fecha SET trimestre_KTL=trimestre;
ALTER TABLE d_fecha DROP COLUMN trimestre;
ALTER TABLE d_fecha RENAME trimestre_KTL TO trimestre;
;
ALTER TABLE d_fecha ADD COLUMN mes_KTL DOUBLE PRECISION;
UPDATE d_fecha SET mes_KTL=mes;
ALTER TABLE d_fecha DROP COLUMN mes;
ALTER TABLE d_fecha RENAME mes_KTL TO mes;
;
ALTER TABLE d_fecha ADD COLUMN dia_KTL DOUBLE PRECISION;
UPDATE d_fecha SET dia_KTL=dia;
ALTER TABLE d_fecha DROP COLUMN dia;
ALTER TABLE d_fecha RENAME dia_KTL TO dia;
;
ALTER TABLE d_fecha ADD COLUMN dia_semana_KTL DOUBLE
PRECISION;
UPDATE d_fecha SET dia_semana_KTL=dia_semana;
ALTER TABLE d_fecha DROP COLUMN dia_semana;
ALTER TABLE d_fecha RENAME dia_semana_KTL TO dia_semana;
;
ALTER TABLE d_fecha ADD COLUMN semestre_nombre_KTL
VARCHAR(10);
UPDATE d_fecha SET semestre_nombre_KTL=semestre_nombre;
ALTER TABLE d_fecha DROP COLUMN semestre_nombre;
ALTER TABLE d_fecha RENAME semestre_nombre_KTL TO
semestre_nombre;
;
ALTER TABLE d_fecha ADD COLUMN trimestre_nombre_KTL VARCHAR(11);
UPDATE d_fecha SET trimestre_nombre_KTL=trimestre_nombre;
```

```
ALTER TABLE d_fecha DROP COLUMN trimestre_nombre;
ALTER TABLE d_fecha RENAME trimestre_nombre_KTL TO
trimestre_nombre;
```

```
;
```

```
ALTER TABLE d_fecha ADD COLUMN mes_nombre_KTL VARCHAR(10);
```

```
UPDATE d_fecha SET mes_nombre_KTL=mes_nombre;
```

```
ALTER TABLE d_fecha DROP COLUMN mes_nombre;
```

```
ALTER TABLE d_fecha RENAME mes_nombre_KTL TO mes_nombre;
```

```
;
```

Transformación TRS_D_CAC

```
-- Step : Carga_d_cac
```

```
-- Database Connection : dm_ventas
```

```
-- SQL :
```

```
ALTER TABLE d_cac ADD COLUMN cp_cac_KTL VARCHAR(3);
```

```
UPDATE d_cac SET cp_cac_KTL=cp_cac;
```

```
ALTER TABLE d_cac DROP COLUMN cp_cac;
```

```
ALTER TABLE d_cac RENAME cp_cac_KTL TO cp_cac;
```

```
;
```

```
ALTER TABLE d_cac ADD COLUMN descripcion_KTL VARCHAR(32);
```

```
UPDATE d_cac SET descripcion_KTL=descripcion;
```

```
ALTER TABLE d_cac DROP COLUMN descripcion;
```

```
ALTER TABLE d_cac RENAME descripcion_KTL TO descripcion;
```

```
;
```

Transformación TRS_D_PLAN

```
-- Step : Carga_d_plan
```

```
-- Database Connection : dm_ventas
```

```
-- SQL :
```

```
ALTER TABLE d_plan ADD COLUMN cp_plan_KTL VARCHAR(20);
```

```
UPDATE d_plan SET cp_plan_KTL=cp_plan;
```

```
ALTER TABLE d_plan DROP COLUMN cp_plan;
```

```
ALTER TABLE d_plan RENAME cp_plan_KTL TO cp_plan;
```

```
;  
ALTER TABLE d_plan ADD COLUMN descripcion_KTL VARCHAR(120);  
UPDATE d_plan SET descripcion_KTL=descripcion;  
ALTER TABLE d_plan DROP COLUMN descripcion;  
ALTER TABLE d_plan RENAME descripcion_KTL TO descripcion;  
;  
ALTER TABLE d_plan ADD COLUMN subproducto_KTL VARCHAR(3);  
UPDATE d_plan SET subproducto_KTL=subproducto;  
ALTER TABLE d_plan DROP COLUMN subproducto;  
ALTER TABLE d_plan RENAME subproducto_KTL TO subproducto;  
;
```

Transformación TRS_D_CLIENTE

```
-- Step          : Carga_d_cliente  
-- Database Connection : dm_ventas  
-- SQL           :  
ALTER TABLE d_cliente ADD COLUMN nombres_KTL VARCHAR(64);  
UPDATE d_cliente SET nombres_KTL=nombres;  
ALTER TABLE d_cliente DROP COLUMN nombres;  
ALTER TABLE d_cliente RENAME nombres_KTL TO nombres;  
;  
ALTER TABLE d_cliente ADD COLUMN ciudad_KTL VARCHAR(30);  
UPDATE d_cliente SET ciudad_KTL=ciudad;  
ALTER TABLE d_cliente DROP COLUMN ciudad;  
ALTER TABLE d_cliente RENAME ciudad_KTL TO ciudad;  
;
```

Transformación TRS_D_ASESOR

```
-- Step          : Carga_d_asesor  
-- Database Connection : dm_ventas  
-- SQL           :
```

```
ALTER TABLE d_asesor ADD COLUMN nombres_KTL VARCHAR(64);
UPDATE d_asesor SET nombres_KTL=nombres;
ALTER TABLE d_asesor DROP COLUMN nombres;
ALTER TABLE d_asesor RENAME nombres_KTL TO nombres;
;
ALTER TABLE d_asesor ADD COLUMN usuario_axis_KTL VARCHAR(8);
UPDATE d_asesor SET usuario_axis_KTL=usuario_axis;
ALTER TABLE d_asesor DROP COLUMN usuario_axis;
ALTER TABLE d_asesor RENAME usuario_axis_KTL TO usuario_axis;
;
CREATE INDEX idx_d_asesor_lookup
ON d_asesor
(
  cp_asesor
)
;
```

JOB_CARGA_DIMENSIONES

```
-- Step          : Carga_d_cac
-- Database Connection : dm_ventas
-- SQL           :
ALTER TABLE d_cac ADD COLUMN cp_cac_KTL VARCHAR(3);
UPDATE d_cac SET cp_cac_KTL=cp_cac;
ALTER TABLE d_cac DROP COLUMN cp_cac;
ALTER TABLE d_cac RENAME cp_cac_KTL TO cp_cac;
;
ALTER TABLE d_cac ADD COLUMN descripcion_KTL VARCHAR(32);
UPDATE d_cac SET descripcion_KTL=descripcion;
ALTER TABLE d_cac DROP COLUMN descripcion;
```



```
ALTER TABLE d_cac RENAME descripcion_KTL TO descripcion;
```

```
;
```

```
-----  
-- Step          : Carga_d_plan
```

```
-- Database Connection : dm_ventas
```

```
-- SQL           :
```

```
ALTER TABLE d_plan ADD COLUMN cp_plan_KTL VARCHAR(20);
```

```
UPDATE d_plan SET cp_plan_KTL=cp_plan;
```

```
ALTER TABLE d_plan DROP COLUMN cp_plan;
```

```
ALTER TABLE d_plan RENAME cp_plan_KTL TO cp_plan;
```

```
;
```

```
ALTER TABLE d_plan ADD COLUMN descripcion_KTL VARCHAR(120);
```

```
UPDATE d_plan SET descripcion_KTL=descripcion;
```

```
ALTER TABLE d_plan DROP COLUMN descripcion;
```

```
ALTER TABLE d_plan RENAME descripcion_KTL TO descripcion;
```

```
;
```

```
ALTER TABLE d_plan ADD COLUMN subproducto_KTL VARCHAR(3);
```

```
UPDATE d_plan SET subproducto_KTL=subproducto;
```

```
ALTER TABLE d_plan DROP COLUMN subproducto;
```

```
ALTER TABLE d_plan RENAME subproducto_KTL TO subproducto;
```

```
;
```

```
-----  
-- Step          : Carga_d_cliente
```

```
-- Database Connection : dm_ventas
```

```
-- SQL           :
```

```
ALTER TABLE d_cliente ADD COLUMN nombres_KTL VARCHAR(64);
```

```
UPDATE d_cliente SET nombres_KTL=nombres;
```

```
ALTER TABLE d_cliente DROP COLUMN nombres;
```

```
ALTER TABLE d_cliente RENAME nombres_KTL TO nombres;
```

```
;
```

```
ALTER TABLE d_cliente ADD COLUMN ciudad_KTL VARCHAR(30);
```

```
UPDATE d_cliente SET ciudad_KTL=ciudad;
```

```
ALTER TABLE d_cliente DROP COLUMN ciudad;
ALTER TABLE d_cliente RENAME ciudad_KTL TO ciudad;
;
-----
-- Step          : Carga_d_asesor
-- Database Connection : dm_ventas
-- SQL           :
ALTER TABLE d_asesor ADD COLUMN nombres_KTL VARCHAR(64);
UPDATE d_asesor SET nombres_KTL=nombres;
ALTER TABLE d_asesor DROP COLUMN nombres;
ALTER TABLE d_asesor RENAME nombres_KTL TO nombres;
;
ALTER TABLE d_asesor ADD COLUMN usuario_axis_KTL VARCHAR(8);
UPDATE d_asesor SET usuario_axis_KTL=usuario_axis;
ALTER TABLE d_asesor DROP COLUMN usuario_axis;
ALTER TABLE d_asesor RENAME usuario_axis_KTL TO usuario_axis;
;
CREATE INDEX idx_d_asesor_lookup
ON d_asesor
(
cp_asesor
)
;
-----
-- Step          : Salida_fecha_carga_dm_ventas
-- Database Connection : dm_ventas
-- SQL           :
ALTER TABLE d_fecha ADD COLUMN fecha_inicial TIMESTAMP
;
ALTER TABLE d_fecha ADD COLUMN numero_dias DOUBLE PRECISION
;
ALTER TABLE d_fecha ADD COLUMN fecha_aux TIMESTAMP
```

```
;  
ALTER TABLE d_fecha ADD COLUMN fecha_aux_string TEXT  
;  
ALTER TABLE d_fecha ADD COLUMN semestre_numero VARCHAR(1)  
;  
ALTER TABLE d_fecha ADD COLUMN dia_semana_1 VARCHAR(9)  
;  
ALTER TABLE d_fecha ADD COLUMN cs_fecha_KTL DOUBLE PRECISION;  
UPDATE d_fecha SET cs_fecha_KTL=cs_fecha;  
ALTER TABLE d_fecha DROP COLUMN cs_fecha;  
ALTER TABLE d_fecha RENAME cs_fecha_KTL TO cs_fecha;  
;  
ALTER TABLE d_fecha ADD COLUMN anio_KTL DOUBLE PRECISION;  
UPDATE d_fecha SET anio_KTL=anio;  
ALTER TABLE d_fecha DROP COLUMN anio;  
ALTER TABLE d_fecha RENAME anio_KTL TO anio;  
;  
ALTER TABLE d_fecha ADD COLUMN semestre_KTL DOUBLE PRECISION;  
UPDATE d_fecha SET semestre_KTL=semestre;  
  
ALTER TABLE d_fecha DROP COLUMN semestre;  
ALTER TABLE d_fecha RENAME semestre_KTL TO semestre;  
;  
ALTER TABLE d_fecha ADD COLUMN trimestre_KTL DOUBLE PRECISION;  
UPDATE d_fecha SET trimestre_KTL=trimestre;  
ALTER TABLE d_fecha DROP COLUMN trimestre;  
ALTER TABLE d_fecha RENAME trimestre_KTL TO trimestre;  
;  
ALTER TABLE d_fecha ADD COLUMN mes_KTL DOUBLE PRECISION;  
UPDATE d_fecha SET mes_KTL=mes;  
ALTER TABLE d_fecha DROP COLUMN mes;  
ALTER TABLE d_fecha RENAME mes_KTL TO mes;
```

```
;  
ALTER TABLE d_fecha ADD COLUMN dia_KTL DOUBLE PRECISION;  
UPDATE d_fecha SET dia_KTL=dia;  
ALTER TABLE d_fecha DROP COLUMN dia;  
ALTER TABLE d_fecha RENAME dia_KTL TO dia;  
;  
ALTER TABLE d_fecha ADD COLUMN dia_semana_KTL DOUBLE  
PRECISION;  
UPDATE d_fecha SET dia_semana_KTL=dia_semana;  
ALTER TABLE d_fecha DROP COLUMN dia_semana;  
ALTER TABLE d_fecha RENAME dia_semana_KTL TO dia_semana;  
;  
ALTER TABLE d_fecha ADD COLUMN semestre_nombre_KTL  
VARCHAR(10);  
UPDATE d_fecha SET semestre_nombre_KTL=semestre_nombre;  
ALTER TABLE d_fecha DROP COLUMN semestre_nombre;  
ALTER TABLE d_fecha RENAME semestre_nombre_KTL TO  
semestre_nombre;  
;  
ALTER TABLE d_fecha ADD COLUMN trimestre_nombre_KTL VARCHAR(11);  
UPDATE d_fecha SET trimestre_nombre_KTL=trimestre_nombre;  
ALTER TABLE d_fecha DROP COLUMN trimestre_nombre;  
ALTER TABLE d_fecha RENAME trimestre_nombre_KTL TO  
trimestre_nombre;  
;  
ALTER TABLE d_fecha ADD COLUMN mes_nombre_KTL VARCHAR(10);  
UPDATE d_fecha SET mes_nombre_KTL=mes_nombre;  
ALTER TABLE d_fecha DROP COLUMN mes_nombre;  
ALTER TABLE d_fecha RENAME mes_nombre_KTL TO mes_nombre;  
;
```

Transformación TRS_F_VENTAS

```
-- Step          : Carga_dm_ventas
-- Database Connection : dm_ventas
-- SQL           :
ALTER TABLE f_ventas ADD COLUMN cs_fecha_KTL DOUBLE PRECISION;
UPDATE f_ventas SET cs_fecha_KTL=cs_fecha;
ALTER TABLE f_ventas DROP COLUMN cs_fecha;
ALTER TABLE f_ventas RENAME cs_fecha_KTL TO cs_fecha;
;
ALTER TABLE f_ventas ADD COLUMN cs_ventas_KTL DOUBLE PRECISION;
UPDATE f_ventas SET cs_ventas_KTL=cs_ventas;
ALTER TABLE f_ventas DROP COLUMN cs_ventas;
ALTER TABLE f_ventas RENAME cs_ventas_KTL TO cs_ventas;
;
ALTER TABLE f_ventas ADD COLUMN ventas_KTL DOUBLE PRECISION;
UPDATE f_ventas SET ventas_KTL=ventas;
ALTER TABLE f_ventas DROP COLUMN ventas;
ALTER TABLE f_ventas RENAME ventas_KTL TO ventas;
;
```

JOB_CARGA_HECHOS

```
-- Step          : Carga_dm_ventas
-- Database Connection : dm_ventas
-- SQL           :
ALTER TABLE f_ventas ADD COLUMN cs_fecha_KTL DOUBLE PRECISION;
UPDATE f_ventas SET cs_fecha_KTL=cs_fecha;
ALTER TABLE f_ventas DROP COLUMN cs_fecha;
ALTER TABLE f_ventas RENAME cs_fecha_KTL TO cs_fecha;
;
ALTER TABLE f_ventas ADD COLUMN cs_ventas_KTL DOUBLE PRECISION;
UPDATE f_ventas SET cs_ventas_KTL=cs_ventas;
ALTER TABLE f_ventas DROP COLUMN cs_ventas;
```

```
ALTER TABLE f_ventas RENAME cs_ventas_KTL TO cs_ventas;
;
ALTER TABLE f_ventas ADD COLUMN ventas_KTL DOUBLE PRECISION;
UPDATE f_ventas SET ventas_KTL=ventas;
ALTER TABLE f_ventas DROP COLUMN ventas;
ALTER TABLE f_ventas RENAME ventas_KTL TO ventas;
;

JOB_CARGA_DM_VENTAS
-- Step          : Salida_archivos_csv
-- Database Connection : stage
-- SQL           :
ALTER TABLE stg_archivos_ventas ADD COLUMN FECHA_INGRESO_KTL
TEXT;
UPDATE          stg_archivos_ventas          SET
FECHA_INGRESO_KTL=FECHA_INGRESO;
ALTER TABLE stg_archivos_ventas DROP COLUMN FECHA_INGRESO;
ALTER TABLE stg_archivos_ventas RENAME FECHA_INGRESO_KTL TO
FECHA_INGRESO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN ID_SERVICIO_KTL TEXT;
UPDATE stg_archivos_ventas SET ID_SERVICIO_KTL=ID_SERVICIO;
ALTER TABLE stg_archivos_ventas DROP COLUMN ID_SERVICIO;
ALTER TABLE stg_archivos_ventas RENAME ID_SERVICIO_KTL TO
ID_SERVICIO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN ASESOR_KTL TEXT;
UPDATE stg_archivos_ventas SET ASESOR_KTL=ASESOR;
ALTER TABLE stg_archivos_ventas DROP COLUMN ASESOR;
ALTER TABLE stg_archivos_ventas RENAME ASESOR_KTL TO ASESOR;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN NOMBRE_ASESOR_KTL
```

```
TEXT;
UPDATE          stg_archivos_ventas          SET
NOMBRE_ASESOR_KTL=NOMBRE_ASESOR;
ALTER TABLE stg_archivos_ventas DROP COLUMN NOMBRE_ASESOR;
ALTER TABLE stg_archivos_ventas RENAME NOMBRE_ASESOR_KTL TO
NOMBRE_ASESOR;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN SUBPRODUCTO_KTL
TEXT;
UPDATE stg_archivos_ventas SET SUBPRODUCTO_KTL=SUBPRODUCTO;
ALTER TABLE stg_archivos_ventas DROP COLUMN SUBPRODUCTO;
ALTER TABLE stg_archivos_ventas RENAME SUBPRODUCTO_KTL TO
SUBPRODUCTO;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN OFICINA_KTL TEXT;
UPDATE stg_archivos_ventas SET OFICINA_KTL=OFICINA;
ALTER TABLE stg_archivos_ventas DROP COLUMN OFICINA;
ALTER TABLE stg_archivos_ventas RENAME OFICINA_KTL TO OFICINA;
;
ALTER      TABLE      stg_archivos_ventas      ADD      COLUMN
DESCRIPCION_OFICINA_KTL TEXT;
UPDATE          stg_archivos_ventas          SET
DESCRIPCION_OFICINA_KTL=DESCRIPCION_OFICINA;
ALTER      TABLE      stg_archivos_ventas      DROP      COLUMN
DESCRIPCION_OFICINA;
ALTER TABLE stg_archivos_ventas RENAME DESCRIPCION_OFICINA_KTL
TO DESCRIPCION_OFICINA;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN PLANES_KTL TEXT;
UPDATE stg_archivos_ventas SET PLANES_KTL=PLANES;
ALTER TABLE stg_archivos_ventas DROP COLUMN PLANES;
ALTER TABLE stg_archivos_ventas RENAME PLANES_KTL TO PLANES;
```

```
;  
ALTER TABLE stg_archivos_ventas ADD COLUMN  
DESCRIPCION_PLAN_KTL TEXT;  
UPDATE stg_archivos_ventas SET  
DESCRIPCION_PLAN_KTL=DESCRIPCION_PLAN;  
ALTER TABLE stg_archivos_ventas DROP COLUMN DESCRIPCION_PLAN;  
ALTER TABLE stg_archivos_ventas RENAME DESCRIPCION_PLAN_KTL TO  
DESCRIPCION_PLAN;  
;  
ALTER TABLE stg_archivos_ventas ADD COLUMN MODELO_EQUIPO_KTL  
TEXT;  
UPDATE stg_archivos_ventas SET  
MODELO_EQUIPO_KTL=MODELO_EQUIPO;  
ALTER TABLE stg_archivos_ventas DROP COLUMN MODELO_EQUIPO;  
ALTER TABLE stg_archivos_ventas RENAME MODELO_EQUIPO_KTL TO  
MODELO_EQUIPO;  
;  
ALTER TABLE stg_archivos_ventas ADD COLUMN EQUIPO_PROPIO_KTL  
TEXT;  
UPDATE stg_archivos_ventas SET  
EQUIPO_PROPIO_KTL=EQUIPO_PROPIO;  
ALTER TABLE stg_archivos_ventas DROP COLUMN EQUIPO_PROPIO;  
ALTER TABLE stg_archivos_ventas RENAME EQUIPO_PROPIO_KTL TO  
EQUIPO_PROPIO;  
;  
ALTER TABLE stg_archivos_ventas ADD COLUMN CEDULA_KTL TEXT;  
UPDATE stg_archivos_ventas SET CEDULA_KTL=CEDULA;  
ALTER TABLE stg_archivos_ventas DROP COLUMN CEDULA;  
ALTER TABLE stg_archivos_ventas RENAME CEDULA_KTL TO CEDULA;  
;  
ALTER TABLE stg_archivos_ventas ADD COLUMN CLIENTE_KTL TEXT;  
UPDATE stg_archivos_ventas SET CLIENTE_KTL=CLIENTE;
```



```
ALTER TABLE stg_archivos_ventas DROP COLUMN CLIENTE;
ALTER TABLE stg_archivos_ventas RENAME CLIENTE_KTL TO CLIENTE;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN USUARIO_AXIS_KTL
TEXT;
UPDATE stg_archivos_ventas SET USUARIO_AXIS_KTL=USUARIO_AXIS;
ALTER TABLE stg_archivos_ventas DROP COLUMN USUARIO_AXIS;
ALTER TABLE stg_archivos_ventas RENAME USUARIO_AXIS_KTL TO
USUARIO_AXIS;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN PORTABILIDAD_KTL
TEXT;
UPDATE stg_archivos_ventas SET PORTABILIDAD_KTL=PORTABILIDAD;
ALTER TABLE stg_archivos_ventas DROP COLUMN PORTABILIDAD;
ALTER TABLE stg_archivos_ventas RENAME PORTABILIDAD_KTL TO
PORTABILIDAD;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN TELEFONO1_KTL TEXT;
UPDATE stg_archivos_ventas SET TELEFONO1_KTL=TELEFONO1;
ALTER TABLE stg_archivos_ventas DROP COLUMN TELEFONO1;

ALTER TABLE stg_archivos_ventas RENAME TELEFONO1_KTL TO
TELEFONO1;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN TELEFONO2_KTL TEXT;
UPDATE stg_archivos_ventas SET TELEFONO2_KTL=TELEFONO2;
ALTER TABLE stg_archivos_ventas DROP COLUMN TELEFONO2;
ALTER TABLE stg_archivos_ventas RENAME TELEFONO2_KTL TO
TELEFONO2;
;
ALTER TABLE stg_archivos_ventas ADD COLUMN LOCALIDAD_KTL TEXT;
UPDATE stg_archivos_ventas SET LOCALIDAD_KTL=LOCALIDAD;
```

```
ALTER TABLE stg_archivos_ventas DROP COLUMN LOCALIDAD;
ALTER TABLE stg_archivos_ventas RENAME LOCALIDAD_KTL TO
LOCALIDAD;
```

```
;
```

```
-----
-- Step          : Salida_stg_archivos_ventas_datos_limpiados
-- Database Connection : stage
-- SQL           :
ALTER TABLE stg_archivos_ventas_datos_limpiados ADD COLUMN
cedula_KTL TEXT;
UPDATE stg_archivos_ventas_datos_limpiados SET cedula_KTL=cedula;
ALTER TABLE stg_archivos_ventas_datos_limpiados DROP COLUMN cedula;
ALTER TABLE stg_archivos_ventas_datos_limpiados RENAME cedula_KTL
TO cedula;
```

```
;
```

```
-----
-- Step          : Salida_archivo_xls
-- Database Connection : stage
-- SQL           :
ALTER TABLE stg_planes ADD COLUMN ID_PLAN_KTL TEXT;
UPDATE stg_planes SET ID_PLAN_KTL=ID_PLAN;
ALTER TABLE stg_planes DROP COLUMN ID_PLAN;
ALTER TABLE stg_planes RENAME ID_PLAN_KTL TO ID_PLAN;
;
ALTER TABLE stg_planes ADD COLUMN DESCRIPCION_KTL TEXT;
UPDATE stg_planes SET DESCRIPCION_KTL=DESCRIPCION;
ALTER TABLE stg_planes DROP COLUMN DESCRIPCION;
ALTER TABLE stg_planes RENAME DESCRIPCION_KTL TO DESCRIPCION;
;
ALTER TABLE stg_planes ADD COLUMN CUOTA_MENSUAL_PLAN_KTL
DOUBLE PRECISION;
UPDATE stg_planes SET
```

```
CUOTA_MENSUAL_PLAN_KTL=CUOTA_MENSUAL_PLAN;
ALTER TABLE stg_planes DROP COLUMN CUOTA_MENSUAL_PLAN;
ALTER TABLE stg_planes RENAME CUOTA_MENSUAL_PLAN_KTL TO
CUOTA_MENSUAL_PLAN;
```

```
;
```

```
-----
```

```
-- Step          : Carga_d_cac
```

```
-- Database Connection : dm_ventas
```

```
-- SQL           :
```

```
ALTER TABLE d_cac ADD COLUMN cp_cac_KTL VARCHAR(3);
```

```
UPDATE d_cac SET cp_cac_KTL=cp_cac;
```

```
ALTER TABLE d_cac DROP COLUMN cp_cac;
```

```
ALTER TABLE d_cac RENAME cp_cac_KTL TO cp_cac;
```

```
;
```

```
ALTER TABLE d_cac ADD COLUMN descripcion_KTL VARCHAR(32);
```

```
UPDATE d_cac SET descripcion_KTL=descripcion;
```

```
ALTER TABLE d_cac DROP COLUMN descripcion;
```

```
ALTER TABLE d_cac RENAME descripcion_KTL TO descripcion;
```

```
;
```

```
-----
```

```
-- Step          : Carga_d_plan
```

```
-- Database Connection : dm_ventas
```

```
-- SQL           :
```

```
ALTER TABLE d_plan ADD COLUMN cp_plan_KTL VARCHAR(20);
```

```
UPDATE d_plan SET cp_plan_KTL=cp_plan;
```

```
ALTER TABLE d_plan DROP COLUMN cp_plan;
```

```
ALTER TABLE d_plan RENAME cp_plan_KTL TO cp_plan;
```

```
;
```

```
ALTER TABLE d_plan ADD COLUMN descripcion_KTL VARCHAR(120);
```

```
UPDATE d_plan SET descripcion_KTL=descripcion;
```

```
ALTER TABLE d_plan DROP COLUMN descripcion;
```

```
ALTER TABLE d_plan RENAME descripcion_KTL TO descripcion;
```

```
;  
ALTER TABLE d_plan ADD COLUMN subproducto_KTL VARCHAR(3);  
UPDATE d_plan SET subproducto_KTL=subproducto;  
ALTER TABLE d_plan DROP COLUMN subproducto;  
ALTER TABLE d_plan RENAME subproducto_KTL TO subproducto;  
;  
-----  
-- Step          : Carga_d_cliente  
-- Database Connection : dm_ventas  
-- SQL           :  
ALTER TABLE d_cliente ADD COLUMN nombres_KTL VARCHAR(64);  
UPDATE d_cliente SET nombres_KTL=nombres;  
ALTER TABLE d_cliente DROP COLUMN nombres;  
ALTER TABLE d_cliente RENAME nombres_KTL TO nombres;  
;  
ALTER TABLE d_cliente ADD COLUMN ciudad_KTL VARCHAR(30);  
UPDATE d_cliente SET ciudad_KTL=ciudad;  
ALTER TABLE d_cliente DROP COLUMN ciudad;  
ALTER TABLE d_cliente RENAME ciudad_KTL TO ciudad;  
;  
-----  
-- Step          : Carga_d_asesor  
-- Database Connection : dm_ventas  
-- SQL           :  
ALTER TABLE d_asesor ADD COLUMN nombres_KTL VARCHAR(64);  
UPDATE d_asesor SET nombres_KTL=nombres;  
ALTER TABLE d_asesor DROP COLUMN nombres;  
ALTER TABLE d_asesor RENAME nombres_KTL TO nombres;  
;  
ALTER TABLE d_asesor ADD COLUMN usuario_axis_KTL VARCHAR(8);  
UPDATE d_asesor SET usuario_axis_KTL=usuario_axis;  
ALTER TABLE d_asesor DROP COLUMN usuario_axis;
```

```

ALTER TABLE d_asesor RENAME usuario_axis_KTL TO usuario_axis;
;
CREATE INDEX idx_d_asesor_lookup
ON d_asesor
(
cp_asesor
)
;
-----
-- Step          : Salida_fecha_carga_dm_ventas
-- Database Connection : dm_ventas
-- SQL           :
ALTER TABLE d_fecha ADD COLUMN fecha_inicial TIMESTAMP
;
ALTER TABLE d_fecha ADD COLUMN numero_dias DOUBLE PRECISION
;
ALTER TABLE d_fecha ADD COLUMN fecha_aux TIMESTAMP
;
ALTER TABLE d_fecha ADD COLUMN fecha_aux_string TEXT
;
ALTER TABLE d_fecha ADD COLUMN semestre_numero VARCHAR(1)
;
ALTER TABLE d_fecha ADD COLUMN dia_semana_1 VARCHAR(9)
;
ALTER TABLE d_fecha ADD COLUMN cs_fecha_KTL DOUBLE PRECISION;
UPDATE d_fecha SET cs_fecha_KTL=cs_fecha;
ALTER TABLE d_fecha DROP COLUMN cs_fecha;
ALTER TABLE d_fecha RENAME cs_fecha_KTL TO cs_fecha;
;
ALTER TABLE d_fecha ADD COLUMN anio_KTL DOUBLE PRECISION;
UPDATE d_fecha SET anio_KTL=anio;

```

```
ALTER TABLE d_fecha DROP COLUMN anio;
ALTER TABLE d_fecha RENAME anio_KTL TO anio;
;
ALTER TABLE d_fecha ADD COLUMN semestre_KTL DOUBLE PRECISION;
UPDATE d_fecha SET semestre_KTL=semestre;
ALTER TABLE d_fecha DROP COLUMN semestre;
ALTER TABLE d_fecha RENAME semestre_KTL TO semestre;
;
ALTER TABLE d_fecha ADD COLUMN trimestre_KTL DOUBLE PRECISION;
UPDATE d_fecha SET trimestre_KTL=trimestre;
ALTER TABLE d_fecha DROP COLUMN trimestre;
ALTER TABLE d_fecha RENAME trimestre_KTL TO trimestre;
;
ALTER TABLE d_fecha ADD COLUMN mes_KTL DOUBLE PRECISION;
UPDATE d_fecha SET mes_KTL=mes;
ALTER TABLE d_fecha DROP COLUMN mes;
ALTER TABLE d_fecha RENAME mes_KTL TO mes;
;
ALTER TABLE d_fecha ADD COLUMN dia_KTL DOUBLE PRECISION;
UPDATE d_fecha SET dia_KTL=dia;
ALTER TABLE d_fecha DROP COLUMN dia;
ALTER TABLE d_fecha RENAME dia_KTL TO dia;
;
ALTER TABLE d_fecha ADD COLUMN dia_semana_KTL DOUBLE
PRECISION;
UPDATE d_fecha SET dia_semana_KTL=dia_semana;
ALTER TABLE d_fecha DROP COLUMN dia_semana;
ALTER TABLE d_fecha RENAME dia_semana_KTL TO dia_semana;
;
ALTER TABLE d_fecha ADD COLUMN semestre_nombre_KTL
VARCHAR(10);
UPDATE d_fecha SET semestre_nombre_KTL=semestre_nombre;
```

```

ALTER TABLE d_fecha DROP COLUMN semestre_nombre;
ALTER TABLE d_fecha RENAME semestre_nombre_KTL TO
semestre_nombre;
;
ALTER TABLE d_fecha ADD COLUMN trimestre_nombre_KTL VARCHAR(11);
UPDATE d_fecha SET trimestre_nombre_KTL=trimestre_nombre;
ALTER TABLE d_fecha DROP COLUMN trimestre_nombre;
ALTER TABLE d_fecha RENAME trimestre_nombre_KTL TO
trimestre_nombre;
;
ALTER TABLE d_fecha ADD COLUMN mes_nombre_KTL VARCHAR(10);
UPDATE d_fecha SET mes_nombre_KTL=mes_nombre;
ALTER TABLE d_fecha DROP COLUMN mes_nombre;
ALTER TABLE d_fecha RENAME mes_nombre_KTL TO mes_nombre;
;

```

```

-----
-- Step          : Carga_dm_ventas
-- Database Connection : dm_ventas
-- SQL           :
ALTER TABLE f_ventas ADD COLUMN cs_fecha_KTL DOUBLE PRECISION;
UPDATE f_ventas SET cs_fecha_KTL=cs_fecha;
ALTER TABLE f_ventas DROP COLUMN cs_fecha;
ALTER TABLE f_ventas RENAME cs_fecha_KTL TO cs_fecha;
;
ALTER TABLE f_ventas ADD COLUMN cs_ventas_KTL DOUBLE PRECISION;
UPDATE f_ventas SET cs_ventas_KTL=cs_ventas;
ALTER TABLE f_ventas DROP COLUMN cs_ventas;
ALTER TABLE f_ventas RENAME cs_ventas_KTL TO cs_ventas;
;
ALTER TABLE f_ventas ADD COLUMN ventas_KTL DOUBLE PRECISION;

```

```
UPDATE f_ventas SET ventas_KTL=ventas;  
ALTER TABLE f_ventas DROP COLUMN ventas;  
ALTER TABLE f_ventas RENAME ventas_KTL TO ventas;
```

Anexo 9

GLOSARIO

Base de datos: Conjunto de datos almacenados de manera sistemática.

Business Intelligence: Conjunto de herramientas que extraen datos de distintas fuentes, para su posterior análisis y toma de decisiones.

Data Mart: Es un subconjunto del *data warehouse*, sirve para realizar el análisis estratégico.

ETL: Significa Extracción, transformación y carga de datos desde una base de datos a otra.

Metadatos: Son datos que describen otros datos.

Clave subrogada: Identificador único que se asigna a cada registro de una tabla de tipo de dato entero y se genera automáticamente.

Query: Consulta a base de datos por medio de código SQL.

Transformaciones: Conjunto de pasos que se realizan de manera secuencial en *Pentaho Data Integration*, para la ejecución de los procesos *ETL*.

JOB: Conjunto de transformaciones que se ejecutan de manera sistemática en *Pentaho Data Integration*, para la ejecución de los procesos *ETL*.