

**ESCUELA POLITÉCNICA DEL EJÉRCITO
EXTENSIÓN LATACUNGA**



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN DE TIEMPOS EN PROYECTOS DE DESARROLLO DE SOFTWARE Y CONTROL DE DESEMPEÑO MEDIANTE CUBOS DE INFORMACIÓN PARA TOMA DE DECISIONES GERENCIALES. CASO PRÁCTICO: CUBOS DE INFORMACIÓN PARA EL CONTROL DEL DESEMPEÑO EN LA EMPRESA BABEL SOFTWARE”

SR. DIEGO EDUARDO GALLARDO CORRALES

**TESIS PRESENTADA COMO REQUISITO PREVIO A LA OBTENCIÓN
DEL GRADO DE**

INGENIERO EN SISTEMAS E INFORMÁTICA

Año 2012

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICADO

ING. RAÚL CAJAS (DIRECTOR)

ING. FABIÁN MONTALUISA (CODIRECTOR)

CERTIFICAN:

Que el trabajo titulado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN DE TIEMPOS EN PROYECTOS DE DESARROLLO DE SOFTWARE Y CONTROL DE DESEMPEÑO MEDIANTE CUBOS DE INFORMACIÓN PARA TOMA DE DECISIONES GERENCIALES. CASO PRÁCTICO: CUBOS DE INFORMACIÓN PARA EL CONTROL DEL DESEMPEÑO EN LA EMPRESA BABEL SOFTWARE” realizado por el señor: GALLARDO CORRALES DIEGO EDUARDO ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, SI recomiendan su publicación.

El mencionado trabajo consta de UN empastado y UN disco compacto el cual contiene los archivos en formato portátil de Acrobat. Autorizan al señor: GALLARDO CORRALES DIEGO EDUARDO que lo entregue al ING. SANTIAGO JÁCOME, en su calidad de Director de Carrera.

Latacunga, 27 de Febrero de 2012.

Ing. Raúl Cajas
DIRECTOR

Ing. Fabián Montaluisa
CODIRECTOR

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por el Sr. Diego Eduardo Gallardo Corrales, bajo nuestra supervisión.

Ing. Raúl Cajas

DIRECTOR DE PROYECTO

Ing. Fabián Montaluisa

CODIRECTOR DE PROYECTO

Ing. Santiago Jácome

DIRECTOR DE CARRERA

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO

ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

DECLARACIÓN DE RESPONSABILIDAD

Yo, GALLARDO CORRALES DIEGO EDUARDO,

DECLARO QUE:

El proyecto de grado denominado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN DE TIEMPOS EN PROYECTOS DE DESARROLLO DE SOFTWARE Y CONTROL DE DESEMPEÑO MEDIANTE CUBOS DE INFORMACIÓN PARA TOMA DE DECISIONES GERENCIALES. CASO PRÁCTICO: CUBOS DE INFORMACIÓN PARA EL CONTROL DEL DESEMPEÑO EN LA EMPRESA BABEL SOFTWARE” ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, 27 de Febrero del 2012.

GALLARDO CORRALES DIEGO EDUARDO

C. C. No.- 0503201717

ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, GALLARDO CORRALES DIEGO EDUARDO,

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución del trabajo “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN DE TIEMPOS EN PROYECTOS DE DESARROLLO DE SOFTWARE Y CONTROL DE DESEMPEÑO MEDIANTE CUBOS DE INFORMACIÓN PARA TOMA DE DECISIONES GERENCIALES. CASO PRÁCTICO: CUBOS DE INFORMACIÓN PARA EL CONTROL DEL DESEMPEÑO EN LA EMPRESA BABEL SOFTWARE”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Latacunga, 27 de Febrero del 2012.

GALLARDO CORRALES DIEGO EDUARDO
C. C. No.- 0503201717

DEDICATORIA

A mis padres, Eduardo y Fanny, por haberme hecho la persona que soy y por su amor incondicional durante toda mi vida, por sus desvelos y caricias, por darme el empuje que necesitaba, por ser ese pilar y ese ejemplo a seguir, dedicado para ustedes con mucho cariño.

A mi hermana María José que siempre fue, es y será un complemento para mí.

Diego

AGRADECIMIENTO

Mi más grande agradecimiento a mis padres, ya que al fin entiendo el sacrificio que hicieron por darme lo mejor, por todo lo que me han dado, esta es mi manera de decirles gracias papá y mamá.

A mi familia que desde mi niñez me han apoyado a culminar con éxito cada una de mis metas. A Lady, a quien tengo mucho cariño y que es parte esencial de mi vida, por su apoyo e incluso desvelos mientras realizaba este proyecto. A mis grandes amigos del barrio y del grupo catequético recordando que somos uno en Cristo. A mis profesores que más que maestros se convirtieron en amigos. A mis compañeros de labores en Babel Software, por enseñarme valores más allá del deber. A Dios mismo, por darme sus bendiciones cada día.

Gracias a ustedes, por brindarme su mano cuando la necesité, y por ayudarme a culminar estos escritos, sepan además que este logro también es suyo.

Diego

ÍNDICE

ÍNDICE.....	vii
TABLAS.....	ix
FIGURAS.....	xi
ANEXOS.....	xiii
RESUMEN.....	xiv
SUMMARY.....	xv
CAPÍTULO 1.....	1
1.1. INTRODUCCIÓN.....	1
1.2. TECNOLOGÍA OLAP.....	2
1.3. TECNOLOGÍA OLAP VS. TECNOLOGÍA OLTP.....	4
1.3.1. OLTP (On-line Transaction Processing).....	4
1.3.2. OLAP (On-Line Analytical Processing).....	5
1.4. TIPOS DE SISTEMAS OLAP.....	7
1.5. DIMENSIONES.....	9
1.5.1. Dimensiones Regulares.....	11
1.5.2. Dimensión De Medida.....	11
1.6. DATA WAREHOUSE.....	12
1.6.1. Definición.....	12
1.6.2. Características de un Data Warehouse.....	14
1.6.3. Arquitectura de un Data Warehouse.....	20
1.7. DATA MART.....	26
1.8. ETL.....	27
1.8.1. Extraer.....	27
1.8.2. Transformar.....	28
1.8.3. Cargar.....	29
CAPÍTULO 2.....	31
2.1. DEFINICIÓN DE UN MODELO DIMENSIONAL.....	31
2.2. MODELOS DE DATOS.....	33
2.2.1. Modelo Entidad Relación.....	34
2.2.2. Modelo Multidimensional.....	36
2.2.3. Modelo Multidimensional vs. Modelo Entidad Relación.....	38
2.3. TIPOS DE ESQUEMAS.....	39
2.3.1. Esquema en estrella.....	39
2.3.2. Esquema en copo de nieve.....	41
2.4. TIPOS DE TABLAS.....	42
2.4.1. Tabla Fact o de hechos.....	42
2.4.2. Tabla Lock-Up o de dimensiones y granularidad.....	43
2.5. PROFUNDIZAR EN EL DISEÑO.....	43
2.5.1. La dimensión Tiempo.....	43

2.5.2.	<i>Dimensiones que varían lentamente en el tiempo</i>	44
2.5.3.	<i>Niveles o Granularidad</i>	45
2.5.4.	<i>Jerarquías</i>	45
2.6.	CUBO OLAP	46
2.6.1.	<i>Definición</i>	46
2.6.2.	<i>Operaciones básicas</i>	47
2.7.	BENEFICIOS DE USAR MODELOS DIMENSIONALES	47
2.8.	PASOS PARA EL MODELADO MULTIDIMENSIONAL	48
2.8.1.	<i>Metodología HEFESTO</i>	49
CAPÍTULO 3.		66
3.1.	MODELO MULTIDIMENSIONAL	66
3.1.1.	<i>Análisis de Requerimientos</i>	66
3.1.2.	<i>Análisis de los OLTP</i>	69
3.1.3.	<i>Modelo lógico del Data Warehouse</i>	79
3.1.4.	<i>Integración de Datos</i>	92
3.2.	CUBO OLAP	194
3.2.1.	<i>Descripción de Dimensiones</i>	195
3.2.2.	<i>Descripción de Hechos</i>	199
3.2.3.	<i>Procedimiento para generar servicios OLAP con Analysis Services 2008 R2</i>	200
3.3.	CREACIÓN DEL REPORTE UTILIZANDO MS EXCEL 2007	207
3.3.1.	<i>Crear Reporte de tabla dinámica de Excel</i>	207
3.3.2.	<i>Explorar cubo "Cubo Asignación de Tiempos" usando el Reporte de Tabla Dinámicos</i>	210
3.3.3.	<i>Reporte Gráfico Dinámico</i>	212
3.4.	IMPLEMENTACIÓN E INTEGRACIÓN DEL CUBO DE INFORMACIÓN Y EL REPORTE AL SISTEMA DE PLANEACIÓN DE TAREAS DEL PERSONAL TÉCNICO	213
3.4.1.	<i>Implementar los paquetes de SQL Server Integration Services con los ETLs para la migración de datos</i>	213
3.4.2.	<i>Implementar el Cubo de Información</i>	216
3.4.3.	<i>Automatizar ejecución de paquetes de Integration Services y procesamiento de la base de datos de Analysis Services</i>	218
3.4.4.	<i>Implementar el reporte creado en MS Excel 2007</i>	224
3.5.	CONSIDERACIONES FINALES	225
CAPÍTULO 4.		227
4.1.	CONCLUSIONES	227
4.2.	RECOMENDACIONES	228
	ANEXOS	229
	BIBLIOGRAFÍA	243

TABLAS

Tabla No. 1.1: Tabla de Características y Diferencias entre OLAP y OLTP	6
Tabla No. 3.1: Descripción Tabla TD_Empleado	79
Tabla No. 3.2: Descripción Tabla TD_Empresa	81
Tabla No. 3.3: Descripción Tabla TD_Nacionalidad.....	82
Tabla No. 3.4: Descripción Tabla TD_Oficina	83
Tabla No. 3.5: Descripción Tabla TD_Proyecto	84
Tabla No. 3.6: Descripción Tabla TD_Rol.....	85
Tabla No. 3.7: Descripción Tabla TD_Tiempo	86
Tabla No. 3.8: Descripción Tabla TH_Asignacion	89
Tabla No. 3.9: pkgSGTipoRol - Data Sources.....	99
Tabla No. 3.10: pkgSGTipoRol – Variables.....	99
Tabla No. 3.11: pkgSGTipoRol – Mapeado	101
Tabla No. 3.12: pkgSGRol – Data Sources.....	102
Tabla No. 3.13: pkgSGRol – Variables	103
Tabla No. 3.14: pkgSGRol – Mapeado	104
Tabla No. 3.15: pkgSGProyectoEmpresa – Data Sources	106
Tabla No. 3.16: pkgSGProyectoEmpresa – Variables.....	107
Tabla No. 3.17: pkgSGProyectoEmpresa – Mapeado.....	108
Tabla No. 3.18: pkgSGProyectoEmpleado – Data Sources	110
Tabla No. 3.19: pkgSGProyectoEmpleado – Variables.....	110
Tabla No. 3.20: pkgSGProyectoEmpleado – Mapeado.....	112
Tabla No. 3.21: pkgSGProyecto – Data Sources	114
Tabla No. 3.22: pkgSGProyecto – Variables.....	115
Tabla No. 3.23: pkgSGProyecto – Mapeado.....	116
Tabla No. 3.24: pkgSGOficina – Data Sources	119
Tabla No. 3.25: pkgSGOficina – Variables.....	119
Tabla No. 3.26: pkgSGOficina – Mapeado.....	121
Tabla No. 3.27: pkgSGNacionalidad – Data Sources.....	122
Tabla No. 3.28: pkgSGNacionalidad – Variables	123
Tabla No. 3.29: pkgSGNacionalidad – Mapeado	125
Tabla No. 3.30: pkgEstado – Data Sources	126
Tabla No. 3.31: pkgEstado – Variables.....	127
Tabla No. 3.32: pkgEstado – Mapeado.....	128
Tabla No. 3.33: pkgSGEmpresa – Data Sources.....	130
Tabla No. 3.34: pkgSGEmpresa – Variables	131
Tabla No. 3.35: pkgSGEmpresa – Mapeado.....	132
Tabla No. 3.36: pkgSGEmpleado – Data Sources	134

Tabla No. 3.37: pkgSGEmpleado – Variables.....	135
Tabla No. 3.38: pkgSGEmpleado – Mapeado.....	137
Tabla No. 3.39: pkgDwTD_Rol – Data Sources.....	142
Tabla No. 3.40: pkgDwTD_Rol – Variables.....	142
Tabla No. 3.41: pkgDwTD_Rol - Mapeado (TD_Rol)	145
Tabla No. 3.42: pkgDwTD_Rol – Mapeado (Failed Rows)	148
Tabla No. 3.43: pkgDwTD_Proyecto – Data Sources	150
Tabla No. 3.44: pkgDwTD_Proyecto - Variables.....	151
Tabla No. 3.45: pkgDwTD_Proyecto – Mapeado	152
Tabla No. 3.46: pkgDwTD_Oficina – Data Sources	154
Tabla No. 3.47: pkgDwTD_Oficina – Variables	155
Tabla No. 3.48: pkgDwTD_Oficina – Mapeado (TD_Oficina)	158
Tabla No. 3.49: pkgDwTD_Oficina – Mapeado (Failed Rows)	160
Tabla No. 3.50: pkgDwTD_Nacionalidad – Data Sources.....	162
Tabla No. 3.51: pkgDwTD_Nacionalidad – Variables.....	162
Tabla No. 3.52: pkgDwTD_Nacionalidad – Mapeado (TD_Nacionalidad)	165
Tabla No. 3.53: pkgDwTD_Nacionalidad – Mapeado (Failed Rows)	168
Tabla No. 3.54: pkgDwTD_Empresa – Data Sources	169
Tabla No. 3.55: pkgDwTD_Empresa – Variables.....	170
Tabla No. 3.56: pkgDwTD_Empresa – Mapeado (TD_Empresa).....	173
Tabla No. 3.57: pkgDwTD_Empresa – Mapeado (TD_Empresa).....	176
Tabla No. 3.58: pkgDwTD_Empleado – Data Sources	178
Tabla No. 3.59: pkgDwTD_Empleado – Variables	178
Tabla No. 3.60: pkgDwTD_Empleado – Mapeado (TD_Empleado)	182
Tabla No. 3.61: pkgDwTD_Empleado – Mapeado (Failed Rows)	189
Tabla No. 3.62: pkgDwTH_Asignacion – Data Sources	192
Tabla No. 3.63: pkgDwTH_Asignacion – Variables.....	192
Tabla No. 3.64: Dimensión Empleado.....	195
Tabla No. 3.65: Dimensión Empresa	195
Tabla No. 3.66: Dimensión Nacionalidad	196
Tabla No. 3.67: Dimensión Oficina	197
Tabla No. 3.68: Dimensión Proyecto	197
Tabla No. 3.69: Dimensión Rol	198
Tabla No. 3.70: Dimensión Tiempo.....	198
Tabla No. 3.71: Hechos Asignación.....	199

FIGURAS

Figura No. 1.1: Representación Gráfica Cubo OLAP	8
Figura No. 1.2: Operaciones base de datos operacionales y Data Warehouse.	19
Figura No. 1.3: Arquitectura de un Data Warehouse.....	20
Figura No. 2.1: Ejemplo de un modelo estrella	32
Figura No. 2.2: Ejemplo Modelo Dimensional (Modelo en Estrella).....	36
Figura No. 2.3: Ejemplo Esquema en Estrella.....	40
Figura No. 2.4: Ejemplo Esquema Copo de Nieve	41
Figura No. 2.5: Granularidad de la dimensión Zona geográfica, con una jerarquía de cinco niveles	45
Figura No. 2.6: Pasos metodología HEFESTO	49
Figura No. 2.7: Modelo Conceptual	53
Figura No. 2.8: Modelo Conceptual Ampliado	55
Figura No. 2.9: Diseño de Tablas de Dimensiones	56
Figura No. 2.10: Jerarquía de “Geografía”	57
Figura No. 2.11: Normalización de “Geografía”	57
Figura No. 2.12: Tabla de hechos	58
Figura No. 2.13: Caso 1: Diseño de tablas de hechos	59
Figura No. 2.14: Caso 2: Diseño de tablas de hechos	60
Figura No. 3.1: Modelo conceptual de porcentaje de asignación de tiempos	69
Figura No. 3.2: Diagrama de tablas BabelSys	72
Figura No. 3.3: Modelo Conceptual ampliado	78
Figura No. 3.4: Relaciones Lógicas	92
Figura No. 3.5: Arquitectura de flujo de datos	93
Figura No. 3.6: pkgSGTipoRol – Control Flow	100
Figura No. 3.7: pkgSGTipoRol – Data Flow (Data Flow Task)	100
Figura No. 3.8: pkgSGRol – Control Flow	103
Figura No. 3.9: pkgSGRol – Data Flow (Data Flow Task)	104
Figura No. 3.10: pkgSGProyectoEmpresa – Control Flow	108
Figura No. 3.11: pkgSGProyectoEmpresa – Data Flow (Data Flow Task)	108
Figura No. 3.12: pkgSGProyectoEmpleado – Control Flow.....	111
Figura No. 3.13: pkgSGProyectoEmpleado – Data Flow (Data Flow Task)	111
Figura No. 3.14: pkgSGProyecto – Control Flow	115
Figura No. 3.15: pkgSGProyecto – Data Flow (Data Flow Task)	116
Figura No. 3.16: pkgSGOficina – Control Flow	120

Figura No. 3.17: pkgSGOficina – Data Flow (Data Flow Task)	120
Figura No. 3.18: pkgSGNacionalidad – Control Flow	124
Figura No. 3.19: pkgSGNacionalidad – Data Flow (Data Flow Task)	124
Figura No. 3.20: pkgEstado – Control Flow	128
Figura No. 3.21: pkgEstado – Data Flow (Data Flow Task).....	128
Figura No. 3.22: pkgSGEmpresa – Control Flow	131
Figura No. 3.23: pkgSGEmpresa – Data Flow (Data Flow Task)	132
Figura No. 3.24: pkgSGEmpleado – Control Flow	136
Figura No. 3.25: pkgSGEmpleado – Data Flow (Data Flow Task).....	136
Figura No. 3.26: pkgDwTD_Rol – Control Flow.....	143
Figura No. 3.27: pkgDwTD_Rol – Data Flow (TD_Rol)	144
Figura No. 3.28: pkgDwTD_Rol – Data Flow (Failed Rows).....	145
Figura No. 3.29: pkgDwTD_Proyecto – Control Flow.....	151
Figura No. 3.30: pkgDwTD_Proyecto – Data Flow (TD_Proyecto).....	152
Figura No. 3.31: pkgDwTD_Oficina – Control Flow.....	156
Figura No. 3.32: pkgDwTD_Oficina – Data Flow (TD_Oficina).....	157
Figura No. 3.33: pkgDwTD_Oficina – Data Flow (Failed Rows)	158
Figura No. 3.34: pkgDwTD_Nacionalidad – Control Flow	163
Figura No. 3.35: pkgDwTD_Nacionalidad – Data Flow (TD_Nacionalidad)	164
Figura No. 3.36: pkgDwTD_Nacionalidad – Data Flow (Failed Rows) ...	165
Figura No. 3.37: pkgDwTD_Empresa – Control Flow.....	171
Figura No. 3.38: pkgDwTD_Empresa – Data Flow (TD_Empresa)	172
Figura No. 3.39: pkgDwTD_Empresa – Data Flow (Failed Rows).....	173
Figura No. 3.40: pkgDwTD_Empleado – Control Flow.....	179
Figura No. 3.41: pkgDwTD_Empleado – Data Flow (TD_Empleado).....	181
Figura No. 3.42: pkgDwTD_Empleado – Data Flow (Failed Rows)	182
Figura No. 3.43 pkgDwTH_Asignacion – Control Flow	193
Figura No. 3.44: Vista de orígenes de datos DW Babel Sys	203
Figura No. 3.45: Esqueleto de Reporte de Tabla Dinámica	210
Figura No. 3.46: Reporte de Tabla Dinámica aplicando filtros en dos dimensiones	211
Figura No. 3.47: Gráfico de Reporte de Tabla Dinámica.....	212

ANEXOS

ANEXO A: Script para crear el Store Procedure: “paLlenarAsignacion”	230
ANEXO B: Script para crear el Store Procedure: “paLlenarTiempo”	235
ANEXO C: Script de procesamiento del cubo de información en comandos XMLA	239
ANEXO D: Carta de conformidad de la empresa auspiciante Babel Software S.A.....	241

RESUMEN

Tome la decisión de trabajar en Babel Software este proyecto, en vista de que es una empresa respetable y con amplios conocimientos en Business Intelligence, los cuales ampliarían los conocimientos adquiridos durante mi vida universitaria.

El desarrollo del tema tratado en este proyecto se lo detalla a continuación:

En el primer capítulo se conocerá gran parte del marco teórico sobre las diferencias y ventajas existentes entre la tecnología OLAP y la tecnología OLTP, revisaremos además las características de un Data Warehouse y que elementos constituyen su arquitectura.

En el segundo capítulo analizaremos la estructura de un modelo dimensional, identificaremos sus tablas y analizaremos sus esquemas, además examinaremos las operaciones básicas de un Cubo OLAP, pero esencialmente trataremos la metodología HEFESTO, la misma que nos ayudará con el desarrollo e implementación del Data Warehouse.

En el tercer capítulo documentaremos el desarrollo del Cubo OLAP y del reporte mismo, utilizando la metodología HEFESTO apoyada en las herramientas de Business Intelligence tales como Analysis Services e Integration Services, propiciadas por SQL Server 2008 R2.

Y por último en el cuarto capítulo se describen las conclusiones obtenidas durante el desarrollo del proyecto, se recomienda eventos y acciones que mejorarían el desempeño de la aplicación a futuro.

SUMMARY

This project was the inclination to work in Babel Software in view of that it is a respectable company with extensive knowledge in Business Intelligence's, which would extend the knowledge acquired during my college life.

The development of the subject matter in this project as follows: In the first chapter will be known much of the theoretical framework about the differences and advantages between the OLAP and OLTP technology, also review the characteristics of a data warehouse and elements that constitute the architecture.

In the second chapter we will analyze the structure of a dimensional model, we will identify its tables and analyze their schemes also examine the basic operations of an OLAP cube, but essentially will treat the methodology HEFESTO, the same that will help with the development and implementation of Data Warehouse.

The third chapter will document the development of OLAP Cube and the report itself, using the methodology HEFESTO supported by Business Intelligence's tools such as Analysis Services and Integration Services, fueled by SQL Server 2008 R2.

And finally in the fourth chapter describes the conclusions reached during the development of the project and recommended events and actions that would improve application performance in the future.

CAPÍTULO 1

TECNOLOGÍA OLAP

1.1. Introducción

En el marco de la inteligencia de negocios (BI, Business Intelligence), es necesario soportar el procesamiento para transformar datos en información “accionable”. Es decir, aprovechar todos los datos que se tienen almacenados y que son producto de la labor diaria, para convertirlos en información como base de las decisiones y acciones a ejecutar.

Los sistemas OLTP proporcionan los medios para la ejecución altamente eficiente de un gran número de pequeñas transacciones y el acceso confiable a los datos almacenados. El volumen de los datos almacenados y procesados por un sistema OLTP podría ser de varios gigabytes por día, después de un período de tiempo, el volumen total de datos puede llegar a las decenas y hasta los cientos de terabytes. Este gran volumen de datos puede ser difícil de almacenar; pero es una valiosa fuente de información para la comprensión del modo en que funciona la empresa.

Esta información puede resultar muy útil para hacer proyecciones y/o mediciones que conducen al éxito de las decisiones estratégicas y para mejorar la toma de decisiones cotidianas. Por lo que es fácil ver

por qué el análisis de datos se ha vuelto tan importante a la gestión de las empresas modernas. Sin embargo, los sistemas OLTP no son muy adecuados para el análisis de datos.

En las últimas décadas, un nuevo mercado entero se ha convertido para los sistemas que pueden proporcionar un acceso fiable y rápido para el análisis de grandes cantidades de datos: procesamiento analítico en línea (OLAP).¹

En definitiva, una herramienta OLAP nos ayuda a cambiar y a enriquecer nuestro punto de vista y nuestra experiencia, nos permite ser creativos, innovadores y resolutivos en nuestra toma de decisiones obteniendo importantes ventajas a la hora de aumentar el beneficio de la empresa, bien por la optimización de procesos, bien por el ahorro de costos, bien por la identificación de las líneas de negocio más rentables o bien porque somos capaces de mejorar nuestros ratios de ventas.

1.2. Tecnología OLAP

OLAP es el acrónimo para Online Analytical Processing. Se trata de un método para dar respuestas rápidas a las peticiones de bases de datos complejas. Se está usando para informes de negocios, marketing, informes de gestión y en áreas de similares y organizacionales. Está diseñada para ayudar a la toma de decisiones y una mejor comprensión de la información.

¹ CAMERON, Scott. Microsoft SQL Server 2008 Analysis Services Step by Step, Microsoft Press, Washington – Estados Unidos, 2009, p. 7.

La tecnología OLAP es una forma específica para representar datos financieros, operacionales, comerciales y estadísticos orientados a los ejecutivos, especialistas y analistas.

En general, el servidor OLAP es un componente independiente que contiene algoritmos especializados y herramientas de indexación que permita procesar los datos de tareas de minería con un impacto mínimo en el rendimiento de base de datos.

El procesamiento analítico en línea es una parte integral de las empresas. Le ayuda en el análisis y la toma de decisiones de una organización. Por ejemplo, las organizaciones de TI a menudo se enfrentan al reto de la entrega de sistemas que permiten a los trabajadores del conocimiento para tomar decisiones estratégicas y tácticas sobre la base de la información corporativa. Estos sistemas de soporte de decisiones son los sistemas OLAP que permiten a los profesionales de manera intuitiva, rápida y flexible manipular cuestiones operacionales para proporcionar una visión analítica.² Por lo general, los sistemas OLAP están diseñadas para:

- ✓ Apoyar las necesidades de análisis complejos de toma de decisiones.
- ✓ Analizar los datos de un número de diferentes perspectivas (dimensiones de negocios).
- ✓ Soporte de análisis complejos contra la entrada de grandes (a nivel atómico) conjuntos de datos.

Las motivaciones que han dado origen al OLAP y al Data Warehouse (DW) en modo alguno pueden ser consideradas

² Online Analytical Processing (OLAP). <http://www.computers.com/es/consejos-3155893.htm>. [Ref. 20 noviembre 2010]

recientes. Sus antecedentes históricos pueden encontrarse en las llamadas bases de datos estadísticas (BDE), que aún hoy siguen siendo utilizadas y que poseen muchos componentes comunes a sus pares de OLAP y DW.³

Una estructura OLAP creada de los datos operacionales se llama cubo OLAP. Dicho cubo se crea de unas tablas con un esquema en estrella. En el centro está la tabla de hechos (fact table), en donde se enlistan los hechos centrales que construyen la petición. Numerosas tablas de dimensión se enlazan con la tabla de hechos. Estas tablas indican cómo se debe analizar las agregaciones de datos relacionales. El número de posibles agregados se determina por todas las posibles maneras en la que la información original se puede enlazar en una jerarquía.⁴

1.3. Tecnología OLAP vs. Tecnología OLTP

1.3.1. OLTP (On-line Transaction Processing)

Se caracteriza por un gran número de cortas transacciones on-line (INSERT, UPDATE, DELETE). El énfasis principal de los sistemas OLTP se pone en el procesamiento muy rápido de consultas, manteniendo la integridad de datos en entornos multi-acceso y una efectividad medida por el número de transacciones por segundo. En la base de datos OLTP hay datos detallados y actuales, y el

³ Dr. MARCH, Alan. Dr. Leguiza, Jorge. Ing. LAURÍA, Etiel. Tecnologías OLAP en la gestión de servicios de salud: su aplicación en el estudio del gasto en medicamentos. [PDF].

http://www.hospitalitaliano.org.ar/archivos/servicios_attachs/1115.pdf

⁴ Ibíd.

esquema utilizado para almacenar bases de datos transaccionales es el modelo de entidad.⁵

El acceso a los datos está optimizado para tareas frecuentes de lectura y escritura. (Por ejemplo, la enorme cantidad de transacciones que tienen que soportar las BD de bancos o hipermercados diariamente). Los datos se estructuran según el nivel aplicación (programa de gestión a medida, ERP o CRM implantado, sistema de información departamental...).

1.3.2. OLAP (On-Line Analytical Processing)

Se caracteriza por el volumen relativamente bajo de transacciones. Las consultas son a menudo muy complejas e implican agregaciones. Para los sistemas OLAP un tiempo de respuesta es una medida de eficacia. Aplicaciones OLAP son ampliamente utilizados por técnicas de Data Mining. En la base de datos OLAP no hay datos globales, históricos, almacenados en los esquemas multidimensionales (esquema general, estrellas).⁶

El historial de datos es a largo plazo, normalmente de dos a cinco años.

Las bases de datos OLAP se suelen alimentar de información procedente de los sistemas operacionales existentes, mediante un proceso de extracción, transformación y carga (ETL).

⁵ Datawarehouse4u.Info. OLTP vs. OLAP [En línea].
<http://datawarehouse4u.info/OLTP-vs-OLAP.html>. [Ref. 20 noviembre 2010]

⁶ Ibíd.

A continuación presento en la Tabla No. 1 el resumen de las diferencias entre los sistemas OLTP y OLAP:

Tabla No. 1.1: Tabla de Características y Diferencias entre OLAP y OLTP⁷

	<i>OLTP Sistema Procesamiento de transacciones en línea (Sistema Operativo)</i>	<i>OLAP Sistema Procesamiento analítico en línea (Datos Galería)</i>
Fuente de los datos	Los datos operativos; OLTPs son la fuente original de los datos.	La consolidación de datos, los datos OLAP proviene de los diferentes bases de datos OLTP
Finalidad de los datos	Para controlar y ejecutar las tareas fundamentales del negocio	Para ayudar con la planificación, resolución de problemas, y apoyo a las decisiones
Qué de los datos	Revela una instantánea del negocio de los procesos	puntos de vista multi-dimensional de los diversos tipos de actividades comerciales
Inserciones y actualizaciones	Corto y rápido inserciones y actualizaciones iniciado por fin los usuarios	Periódico trabajos por lotes de larga duración a cargar los datos
Queries	Consultas relativamente estandarizadas y simples Devolviendo relativamente pocos registros	A menudo consultas complejas que implica agregaciones
Velocidad de Procesamiento	Normalmente muy rápido	Depende de la cantidad de datos implicados, actualiza los datos del lote y consultas complejas puede tomar muchas horas, velocidad de

⁷Ibíd.

		las consultas se pueden mejorar mediante la creación de índices
Requisitos de espacio	Pueden ser relativamente pequeños si los datos históricos se archiva	debido a la existencia de estructuras de agregación de datos y la historia más grande, requiere más índices de OLTP
Base de datos de diseño	Altamente normalizados con muchas tablas	Normalmente desnormalizada con menos tablas, el uso de estrellas y / o esquemas de copo de nieve
Copia de seguridad y recuperación	Copia de seguridad, los datos operativos es fundamental para manejar el negocio, la pérdida de los datos puede provocar una pérdida monetaria importante y la responsabilidad legal	En lugar de copias de seguridad regulares, algunos entornos pueden considerar simplemente volver a cargar los datos OLTP como un método de recuperación

1.4. Tipos de Sistemas OLAP

Los cubos, las dimensiones y las jerarquías son la esencia de la navegación multidimensional del OLAP. Al describir y representar la información en esta forma, los usuarios pueden navegar intuitivamente en un conjunto complejo de datos tal como se presenta en la Figura No. 1.1. Sin embargo, el solo describir el modelo de datos en una forma más intuitiva, hace muy poco para ayudar a entregar la información al usuario más rápidamente.

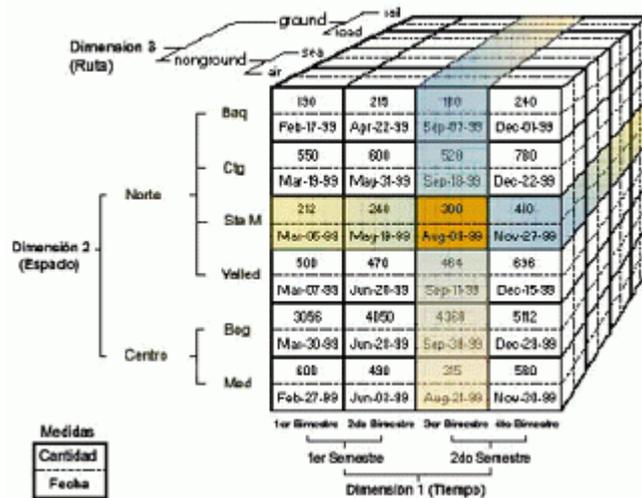


Figura No. 1.1: Representación Gráfica Cubo OLAP⁸

Un principio clave del OLAP es que los usuarios deberían obtener tiempos de respuesta consistentes para cada vista de datos que requieran. Dado que la información se recolecta en el nivel de detalle solamente, el resumen de la información es usualmente calculado por adelantado. Estos valores pre calculados son la base de las ganancias de desempeño del OLAP.

En los primeros días de la tecnología OLAP, la mayoría de las compañías asumía que la única solución para una aplicación OLAP era un modelo de almacenamiento no relacional. Después, otras compañías descubrieron que a través del uso de estructuras de base de datos (esquemas de estrella y de copo de nieve), índices y el almacenamiento de agregados, se podrían utilizar sistemas de

⁸ Sinnexus. Business Intelligence [En línea]. http://www.sinnexus.com/business_intelligence/olap_avanzado.aspx. [Ref. 20 noviembre 2010]

administración de bases de datos relacionales (RDBMS) para el OLAP.⁹

En un Data Warehouse, los datos OLTP normalmente se transforman y se almacena en una base de datos relacional y entonces se carga en una base de datos multidimensional para su análisis. Para conectar las dos bases de datos, se puede elegir entre tres métodos, cada uno con un tipo diferente de interacción:

- Relational OLAP (ROLAP), en el cual ningún dato es almacenado directamente en la base de datos multidimensional. Se carga desde la base de datos relacional cuando sea necesario.
- Multidimensional OLAP (MOLAP), en el cual los datos son cargados en la base de datos multidimensional y se almacenan en caché. Las consultas futuras se ejecutan en los datos almacenados en caché.
- Hybrid OLAP (HOLAP), en los que se almacena en caché los datos agregados en la base de datos multidimensional. Cuando surge la necesidad de obtener información más detallada, que los datos son cargados desde la base de datos relacional.

1.5. Dimensiones

Con OLAP se puede ver un conjunto de datos de su negocio de muchas y diversas formas sin mucho esfuerzo. Los archivos OLAP o cubos modelan los datos en dimensiones. Una dimensión es una clasificación de alguna actividad en una organización por la cual se

⁹ Ibíd.

puede medir su éxito. Por ejemplo, puede monitorear sus ventas contra los productos o clientes en un periodo de tiempo.

Las dimensiones constituyen un componente fundamental de los cubos. Las dimensiones organizan los datos en función de un área de interés para los usuarios, por ejemplo clientes, almacenes o empleados. Las dimensiones en Analysis Services contienen atributos que corresponden a columnas de tablas de dimensiones. Estos atributos aparecen como jerarquías de atributo y se pueden organizar en jerarquías definidas por el usuario, o bien se pueden definir como jerarquías de elementos primarios y secundarios basadas en columnas en la tabla de dimensiones subyacente.¹⁰

Las dimensiones están dispuestas en niveles jerárquicos, con una posición única dentro de cada nivel. Por ejemplo, una dimensión de tiempo puede tener cuatro niveles, como el Año, Trimestre, Mes y Día. O la dimensión podría tener sólo tres niveles, por ejemplo, Año, Semana y Día. Los valores dentro de los niveles se denominan miembros. Por ejemplo, los años 2002 y 2003 son miembros del Año nivel en la dimensión de tiempo.

Hay dos clases de dimensiones que se pueden utilizar, dimensiones regulares y dimensión de medida.

¹⁰ MSDN Library. Dimensiones (Analysis Services) [En línea]. [http://msdn.microsoft.com/es-es/library/ms174527\(v=SQL.90\).aspx](http://msdn.microsoft.com/es-es/library/ms174527(v=SQL.90).aspx). [Ref. 04 diciembre 2010].

1.5.1. Dimensiones Regulares

Dimensiones regulares son aquellos datos que se quieren medir¹¹, por ejemplo, si desea seguir el control de sus ventas, puede utilizar:

- *Clientes*: Quienes son los mejores, donde se encuentran, ¿Qué es lo que compran?
- *Productos*: Con respecto a los clientes, ¿Quién los compra? ¿Qué productos se están vendiendo?
- *Tiempo*: ¿Cómo voy ahora con respecto al último año o último mes?

Una dimensión regular es una dimensión cuyos miembros son los registros actuales de las tablas de dimensiones o los valores actuales de la tabla de hechos (*fact table*). Una dimensión regular podría ser plana o jerárquica. Las jerarquías de algunas dimensiones son definidas por una relación *maestro detalle*, y puede ser balanceada, no balanceada o irregular. Una dimensión regular contiene un número de atributos iguales al número de columnas seleccionadas en su definición.¹²

1.5.2. Dimensión De Medida

Dimensión de medida son los números que aparecen en el análisis dependiendo de los elementos seleccionados en las dimensiones regulares. Por ejemplo, en un cubo de ventas, podríamos escoger

¹¹ OlapX. Que es OLAP [En línea]. <http://www.olapxsoftware.com/es/WhatIsOlap.asp>. [Ref. 04 diciembre 2010].

¹² RadarSoft. Types of dimensions [En línea]. <http://www.radar-soft.com/articles/20070403112348.aspx>. [Ref. 04 diciembre 2010].

ver las ventas, el número de artículos vendidos, ganancia, costo, etc.¹³

Las medidas son los indicadores clave de rendimiento que se desea evaluar. Para determinar cuál de los números en los datos pueden ser medidas, una regla de oro es: Si una serie tiene sentido cuando se agrega, a continuación, se trata de una medida. Por ejemplo, tiene sentido que el volumen diario total a mes, trimestre y año. Por otra parte, la agregación de códigos postales o números de teléfono no tendrían sentido, por lo tanto, los códigos postales y números de teléfono no son medidas. Las medidas típicas incluyen el volumen, las ventas y el costo.

La diferencia entre los tipos está en la manera de generar miembros dimensionales. Mientras que los miembros de dimensiones regulares son los registros desde las tablas de dimensiones o valores desde la tabla de hechos (*fact table*), los miembros de otros tipos de dimensiones son generados de otra manera.

1.6. Data Warehouse

1.6.1. Definición

El concepto de Data Warehouse (DW) surge como solución a las necesidades de información global de la empresa, que los sistemas operacionales no pueden satisfacer. Este término se traduce literalmente como Almacén de Datos, y sus aplicaciones se centran principalmente en dar soporte al proceso de toma de decisiones

¹³ OlapX. Que es OLAP [En línea]. Op. cit.

gerenciales, ya que los datos que se almacenan han pasado por un proceso de calidad que asegura su consistencia.

Un Data Warehouse es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza. Se trata, sobre todo, de un expediente completo de una organización, más allá de la información transaccional y operacional, almacenada en una base de datos diseñada para favorecer el análisis y la divulgación eficiente de datos (OLAP). El almacenamiento de los datos no debe usarse con datos de uso actual. Los almacenes de datos contienen a menudo grandes cantidades de información que se subdividen a veces en unidades lógicas más pequeñas dependiendo del subsistema de la entidad del que procedan o para el que sea necesario.¹⁴

Su construcción se va haciendo por etapas que normalmente corresponden a las principales áreas operativas de la empresa, por ejemplo: Área de Ventas, Área Financiero Contable, Área de Recursos Humanos, etc. Estas áreas reciben el nombre de Data Marts.

Muchos se harán la pregunta de por qué es necesario tener un Data Warehouse si ya tengo toda la información en la base de datos, para qué necesito un repositorio adicional. Pues, la respuesta es sencilla, las bases de datos operacionales o cualquier otro sistema de archivos que está orientado a los sistemas de producción están diseñadas para tener una respuesta rápida a transacciones

¹⁴ Wikipedia. Almacén de datos [En línea].
http://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos. [Ref. 04 diciembre 2010].

relativamente pequeñas: Emitir una factura, registrar un cheque, etc. No están orientadas hacia el proceso masivo de información para reportes. En cambio el Data Warehouse se construye pensando en que tenga una gran capacidad para obtener reportes que involucren el procesamiento de enormes cantidades de información y en el menor tiempo posible.

1.6.2. Características de un Data Warehouse

El Data Warehouse se caracteriza por ser:

a. Integrado

Los datos almacenados en el Data Warehouse deben integrarse en una estructura consistente, por lo que las inconsistencias existentes entre los diversos sistemas operacionales deben ser eliminadas. La información suele estructurarse también en distintos niveles de detalle para adecuarse a las distintas necesidades de los usuarios.

La integración de datos se muestra de muchas maneras: en convenciones de nombres consistentes, en la medida uniforme de variables, en la codificación de estructuras consistentes, en atributos físicos de los datos consistentes, fuentes múltiples y otros.¹⁵

¹⁵ MILÁN, José. LÓPEZ, Juan. FERNÁNDEZ, Lourdes. Data Warehouse [DOC]. <http://serdis.dis.ulpgc.es/~a013775/assignaturas/ii-abd/ABD/Trabajos%20Te%A2ricos/Tema%207.2.-%20Datawarehouse-Datamining/Trabajo%20-%20Data%20Warehouse.doc>

b. Temático

Sólo los datos necesarios para el proceso de generación del conocimiento del negocio se integran desde el entorno operacional. Los datos se organizan por temas para facilitar su acceso y entendimiento por parte de los usuarios finales.

El ambiente operacional se diseña alrededor de las aplicaciones y funciones por ejemplo: préstamos, ahorros, tarjeta bancaria y depósitos para una institución financiera. En el caso una aplicación de ingreso de órdenes puede acceder a los datos sobre clientes, productos y cuentas. La base de datos combina estos elementos en una estructura que acomoda las necesidades de la aplicación.¹⁶

En el ambiente Data Warehousing se organiza alrededor de sujetos tales como cliente, vendedor, producto y actividad. Por ejemplo, para un fabricante, éstos pueden ser clientes, productos, proveedores y vendedores. Para una universidad pueden ser estudiantes, clases y profesores. Para un hospital pueden ser pacientes, personal médico, medicamentos, etc.

Las aplicaciones están relacionadas con el diseño de la base de datos y del proceso. En Data Warehousing se enfoca el modelamiento de datos y el diseño de la base de datos. El diseño del proceso (en su forma clásica) no es separado de este ambiente.

Las diferencias entre la orientación de procesos y funciones de las aplicaciones y la orientación a temas, radican en el contenido de los datos a escala detallada. En el Data Warehouse se excluye la información que no será usada por el proceso de sistemas de

¹⁶ Ibid.

soporte de decisiones, mientras que la información de las orientadas a las aplicaciones, contiene datos para satisfacer de inmediato los requerimientos funcionales y de proceso, que pueden ser usados o no por el analista de soporte de decisiones.¹⁷

c. Histórico

El tiempo es parte implícita de la información contenida en un Data Warehouse. En los sistemas operacionales, los datos siempre reflejan el estado de la actividad del negocio en el momento actual.¹⁸ Por el contrario, la información almacenada en el Data Warehouse sirve, entre otras cosas, para realizar análisis de tendencias. Por lo tanto, el Data Warehouse se carga con los distintos valores que toma una variable en el tiempo para permitir comparaciones.

Toda la información del Data Warehouse es requerida en algún momento. Esta característica básica de los datos en un depósito, es muy diferente de la información encontrada en el ambiente operacional. En éstos, la información se requiere al momento de acceder. En otras palabras, en el ambiente operacional, cuando se accede a una unidad de información, se espera que los valores requeridos se obtengan a partir del momento de acceso.

Los datos históricos son de poco uso en el procesamiento operacional. La información del depósito por el contraste, debe incluir los datos históricos para usarse en la identificación y evaluación de tendencias.

¹⁷ Ibid.

¹⁸ Ibid.

Un tiempo variante se muestra de varias maneras¹⁹:

1. La más simple es que la información representa los datos sobre un horizonte largo de tiempo - desde cinco a diez años. El horizonte de tiempo representado para el ambiente operacional es mucho más corto - desde valores actuales hasta sesenta a noventa días.

Las aplicaciones que tienen un buen rendimiento y están disponibles para el procesamiento de transacciones, deben llevar una cantidad mínima de datos si tienen cualquier grado de flexibilidad. Por ello, las aplicaciones operacionales tienen un corto horizonte de tiempo, debido al diseño de aplicaciones rígidas.

2. La segunda manera en la que se muestra el tiempo variante en el Data Warehouse está en la estructura clave. Cada estructura clave en el Data Warehouse contiene, implícita o explícitamente, un elemento de tiempo como día, semana, mes, etc.

El elemento de tiempo está casi siempre al pie de la clave concatenada, encontrada en el Data Warehouse. En ocasiones, el elemento de tiempo existirá implícitamente, como el caso en que un archivo completo se duplica al final del mes, o al cuarto.

3. La tercera manera en que aparece el tiempo variante es cuando la información del Data Warehouse, una vez registrada correctamente, no puede ser actualizada. La información del Data Warehouse es, para todos los propósitos

¹⁹ SQL Max Connections. Data Warehousing [En línea].
<http://www.sqlmax.com/dataw1.asp>. [Ref. 04 diciembre 2010]

prácticos, una serie larga de "snapshots" (vistas instantáneas).

d. No Volátil

El almacén de información de un Data Warehouse existe para ser leído, y no modificado. La información es por tanto permanente, significando la actualización del Data Warehouse la incorporación de los últimos valores que tomaron las distintas variables contenidas en él sin ningún tipo de acción sobre la información que ya existía. En el Data Warehouse hay dos únicos tipos de operaciones: la carga inicial de datos y el acceso a los mismos.²⁰

En la Figura No. 1.2 se muestra que la actualización (insertar, borrar y modificar), se hace regularmente en el ambiente operacional sobre una base de registro por registro. Pero la manipulación básica de los datos que ocurre en el Data Warehouse es mucho más simple. Hay dos únicos tipos de operaciones: la carga inicial de datos y el acceso a los mismos. No hay actualización de datos (en el sentido general de actualización) en el depósito, como una parte normal de procesamiento.

Hay algunas consecuencias muy importantes de esta diferencia básica, entre el procesamiento operacional y del Data Warehouse. En el nivel de diseño, la necesidad de ser precavido para actualizar las anomalías no es un factor en el Data Warehouse, ya que no se hace la actualización de datos. Esto significa que en el nivel físico de diseño, se pueden tomar libertades para optimizar el acceso a los

²⁰ MILÁN, José. LÓPEZ, Juan. FERNÁNDEZ, Lourdes. Data Warehouse [DOC]. Op. cit.

datos, particularmente al usar la normalización y desnormalización física.

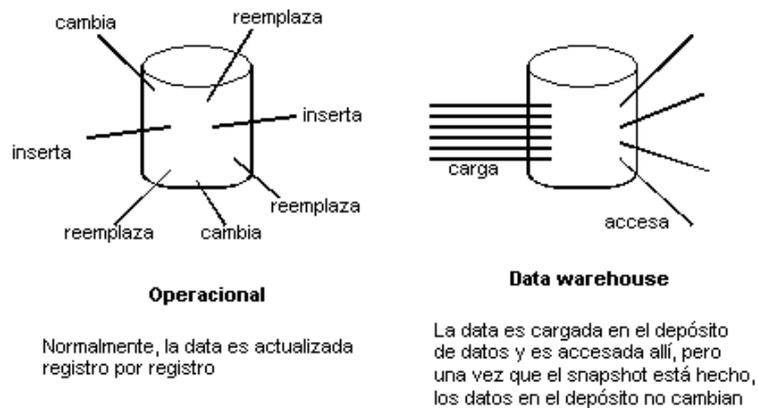


Figura No. 1.2: Operaciones base de datos operacionales y Data Warehouse.²¹

La fuente de casi toda la información del Data Warehouse es el ambiente operacional. A simple vista, se puede pensar que hay redundancia masiva de datos entre los dos ambientes. Desde luego, la primera impresión de muchas personas se centra en la gran redundancia de datos, entre el ambiente operacional y el ambiente de Data Warehouse. Dicho razonamiento es superficial y demuestra una carencia de entendimiento con respecto a qué ocurre en el Data Warehouse. De hecho, hay una mínima redundancia de datos entre ambos ambientes.

²¹ SQL Max Connections. Data Warehousing [En línea]. <http://www.sqlmax.com/dataw1.asp>. [Ref. 04 diciembre 2010]

1.6.3. Arquitectura de un Data Warehouse

Una de las razones por las que el desarrollo de un Data Warehouse crece rápidamente, es que realmente es una tecnología muy entendible. De hecho, Data Warehousing puede representar mejor la estructura amplia de una empresa para administrar los datos informacionales dentro de la organización. A fin de comprender cómo se relacionan todos los componentes involucrados en una estrategia Data Warehousing, es esencial tener una Arquitectura Data Warehouse.

A continuación en la Figura No. 1.3, se presenta un esquema de la arquitectura de un Data Warehouse.

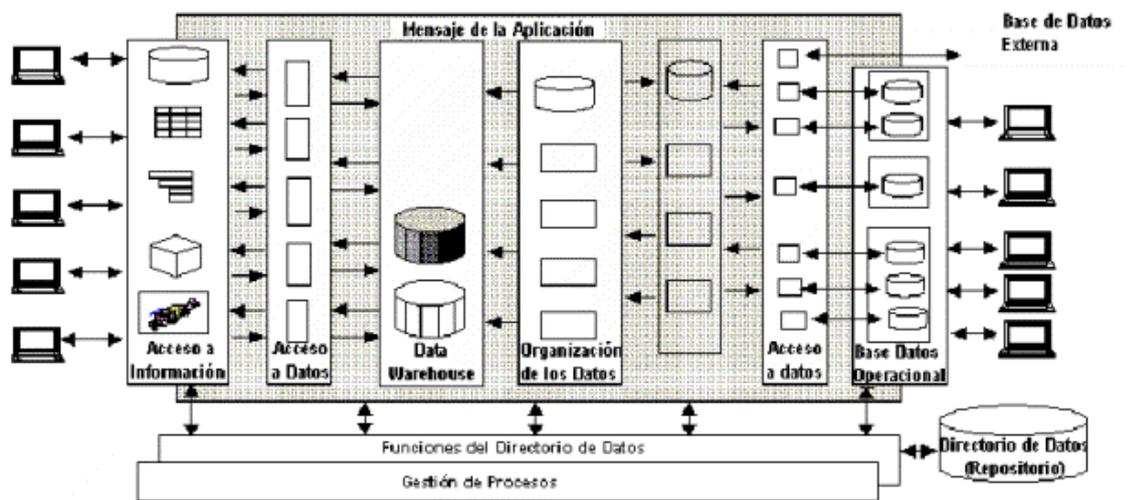


Figura No. 1.3: Arquitectura de un Data Warehouse.²²

²² Ibid.

a. Elementos constituyentes de una Arquitectura Data Warehouse

Una Arquitectura Data Warehouse (Data Warehouse Architecture - DWA) es una forma de representar la estructura total de datos, comunicación, procesamiento y presentación, que existe para los usuarios finales que disponen de una computadora dentro de la empresa.

La arquitectura se constituye de un número de partes interconectadas:

- Base de datos operacional / Nivel de base de datos externo
- Nivel de acceso a la información
- Nivel de acceso a los datos
- Nivel de directorio de datos (Metadata)
- Nivel de gestión de proceso
- Nivel de mensaje de la aplicación
- Nivel de Data Warehouse
- Nivel de organización de datos
- Base de datos operacional / Nivel de base de datos externo

Los sistemas operacionales procesan datos para apoyar las necesidades operacionales críticas. Para hacer eso, se han creado las bases de datos operacionales históricas que proveen una estructura de procesamiento eficiente, para un número relativamente pequeño de transacciones comerciales bien definidas.

Sin embargo, a causa del enfoque limitado de los sistemas operacionales, las bases de datos diseñadas para soportar estos sistemas, tienen dificultad al acceder a los datos para otra gestión o propósitos informáticos.

Esta dificultad en acceder a los datos operacionales es amplificada por el hecho que muchos de estos sistemas tienen de 10 a 15 años de antigüedad. El tiempo de algunos de estos sistemas significa que la tecnología de acceso a los datos disponible para obtener los datos operacionales, es así mismo antigua.

Ciertamente, la meta del Data Warehousing es liberar la información que es almacenada en bases de datos operacionales y combinarla con la información desde otra fuente de datos, generalmente externa.

a.1. Nivel de acceso a la información

El nivel de acceso a la información de la arquitectura Data Warehouse, es el nivel del que el usuario final se encarga directamente. En particular, representa las herramientas que el usuario final normalmente usa día a día. Por ejemplo: Excel, Lotus 1-2-3, Focus, Access, SAS, etc.

Actualmente, existen herramientas más y más sofisticadas para manipular, analizar y presentar los datos, sin embargo, hay problemas significativos al tratar de convertir los datos tal como han sido recolectados y que se encuentran contenidos en los sistemas operacionales en información fácil y transparente para las herramientas de los usuarios finales. Una de las claves para esto es encontrar un lenguaje de datos común que puede usarse a través de toda la empresa.

a.2. Nivel de acceso a los datos

El nivel de acceso a los datos de la arquitectura Data Warehouse está involucrado con el nivel de acceso a la información para conversar en el nivel operacional. En la red mundial de hoy, el lenguaje de datos común que ha surgido es SQL. Originalmente, SQL fue desarrollado por IBM como un lenguaje de consulta, pero en los últimos veinte años ha llegado a ser el estándar para el intercambio de datos.²³

El nivel de acceso a los datos no solamente conecta DBMS diferentes y sistemas de archivos sobre el mismo hardware, sino también a los fabricantes y protocolos de red. Una de las claves de una estrategia Data Warehousing es proveer a los usuarios finales con "acceso a datos universales".

El acceso a los datos universales significa que, teóricamente por lo menos, los usuarios finales sin tener en cuenta la herramienta de acceso a la información o ubicación, deberían ser capaces de acceder a cualquier o todos los datos en la empresa que es necesaria para ellos, para hacer su trabajo.

El nivel de acceso a los datos entonces es responsable de la interfaces entre las herramientas de acceso a la información y las bases de datos operacionales. En algunos casos, esto es todo lo que un usuario final necesita. Sin embargo, en general, las organizaciones desarrollan un plan mucho más sofisticado para el soporte del Data Warehouse.

²³ *Ibíd.*

a.3. Nivel de Directorio de Datos (Metadata)

A fin de proveer el acceso a los datos universales, es absolutamente necesario mantener alguna forma de directorio de datos o repositorio de la información Metadata. La Metadata es la información alrededor de los datos dentro de la empresa.

Las descripciones de registro en un programa COBOL son Metadata. También lo son las sentencias DIMENSIÓN en un programa FORTRAN o las sentencias a crear en SQL.

a.4. Nivel de Gestión de Procesos

El nivel de gestión de procesos tiene que ver con la programación de diversas tareas que deben realizarse para construir y mantener el Data Warehouse y la información del directorio de datos. Este nivel puede depender del alto nivel de control de trabajo para muchos procesos (procedimientos) que deben ocurrir para mantener el Data Warehouse actualizado.

a.5. Nivel de Mensaje de la Aplicación

El nivel de mensaje de la aplicación tiene que ver con el transporte de información alrededor de la red de la empresa. El mensaje de aplicación se refiere también como "subproducto", pero puede involucrar sólo protocolos de red. Puede usarse por ejemplo, para aislar aplicaciones operacionales o estratégicas a partir del formato de datos exactos, recolectar transacciones o los mensajes y entregarlos a una ubicación segura en un tiempo seguro.

a.6. Nivel Data Warehouse (Físico)

En la Data Warehouse (núcleo) es donde ocurre la data actual, usada principalmente para usos estratégicos. En algunos casos, uno puede pensar del Data Warehouse simplemente como una vista lógica o virtual de datos. En muchos ejemplos, el Data Warehouse puede no involucrar almacenamiento de datos.

En un Data Warehouse físico, copias, en algunos casos, muchas copias de datos operacionales y/o externos, son almacenadas realmente en una forma que es fácil de acceder y es altamente flexible. Cada vez más, los Data Warehouse son almacenados sobre plataformas cliente/servidor, pero por lo general se almacenan sobre mainframes.

a.7. Nivel de Organización de Datos

El componente final de la arquitectura Data Warehouse es la organización de los datos. Se llama también gestión de copia o réplica, pero de hecho, incluye todos los procesos necesarios como seleccionar, editar, resumir, combinar y cargar datos en el depósito y acceder a la información desde bases de datos operacionales y/o externas.

La organización de datos involucra con frecuencia una programación compleja, pero cada vez más, están creándose las herramientas Data Warehousing para ayudar en este proceso. Involucra también programas de análisis de calidad de datos y filtros que identifican modelos y estructura de datos dentro de la data operacional existente.

1.7. Data Mart

Un Data Mart es una versión especial de almacén de datos (Data Warehouse). Son subconjuntos de datos con el propósito de ayudar a que un área específica dentro del negocio pueda tomar mejores decisiones. Los datos existentes en este contexto pueden ser agrupados, explorados y propagados de múltiples formas para que diversos grupos de usuarios realicen la explotación de los mismos de la forma más conveniente según sus necesidades.²⁴

El Data Mart es un sistema orientado a la consulta, en el que se producen procesos batch de carga de datos (altas) con una frecuencia baja y conocida. Es consultado mediante herramientas OLAP que ofrecen una visión multidimensional de la información. Sobre estas bases de datos se pueden construir EIS (Executive Information Systems, Sistemas de Información para Directivos) y DSS (Decision Support Systems, Sistemas de Ayuda a la toma de Decisiones).²⁵ Por otra parte, se conoce como *Data Mining* al proceso no trivial de análisis de grandes cantidades de datos con el objetivo de extraer información útil, por ejemplo para realizar clasificaciones o predicciones.

La estrategia de usar Data Marts es particularmente apropiada cuando el Data Warehouse central crece muy rápidamente y los distintos departamentos requieren sólo una pequeña porción de los datos contenidos en él. La creación de estos Data Marts requiere algo más que una simple réplica de los datos (se necesitan tanto la segmentación como algunos métodos adicionales de consolidación).

²⁴ Wikipedia. Data Mart [En línea]. http://es.wikipedia.org/wiki/Data_mart. [Ref. 11 diciembre 2010].

²⁵ *Ibíd.*

1.8. ETL

Extract, Transform and Load (Extraer, transformar y cargar en inglés, frecuentemente abreviado a ETL) es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, Data Mart, o Data Warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

Los procesos ETL también se pueden utilizar para la integración con sistemas heredados.

1.8.1. Extraer

La primera parte del proceso ETL consiste en extraer los datos desde los sistemas de origen. La mayoría de los proyectos de almacenamiento de datos fusionan datos provenientes de diferentes sistemas de origen. Cada sistema separado puede usar una organización diferente de los datos o formatos distintos. Los formatos de las fuentes normalmente se encuentran en bases de datos relacionales o ficheros planos, pero pueden incluir bases de datos no relacionales u otras estructuras diferentes. La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.²⁶

Una parte intrínseca del proceso de extracción es la de analizar los datos extraídos, de lo que resulta un chequeo que verifica si los

²⁶ Wikipedia. Extract, Transform and load [En línea]. http://es.wikipedia.org/wiki/Extract,_transform_and_load. [Ref. 11 diciembre 2010]

datos cumplen la pauta o estructura que se esperaba. De no ser así los datos son rechazados.

Un requerimiento importante que se debe exigir a la tarea de extracción es que ésta cause un impacto mínimo en el sistema origen. Si los datos a extraer son muchos, el sistema de origen se podría ralentizar e incluso colapsar, provocando que éste no pueda utilizarse con normalidad para su uso cotidiano. Por esta razón, en sistemas grandes las operaciones de extracción suelen programarse en horarios o días donde este impacto sea nulo o mínimo.

1.8.2. Transformar

La fase de transformación aplica una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. Algunas fuentes de datos requerirán alguna pequeña manipulación de los datos. No obstante en otros casos pueden ser necesarias aplicar algunas de las siguientes transformaciones:²⁷

- Seleccionar sólo ciertas columnas para su carga (por ejemplo, que las columnas con valores nulos no se carguen).
- Traducir códigos (por ejemplo, si la fuente almacena una "H" para Hombre y "M" para Mujer pero el destino tiene que guardar "1" para Hombre y "2" para Mujer).
- Codificar valores libres (por ejemplo, convertir "Hombre" en "H" o "Sr" en "1").
- Obtener nuevos valores calculados (por ejemplo, total_venta = cantidad * precio).

²⁷ Ibíd.

- Unir datos de múltiples fuentes (por ejemplo, búsquedas, combinaciones, etc.).
- Calcular totales de múltiples filas de datos (por ejemplo, ventas totales de cada región).
- Generación de campos clave en el destino.
- Transponer o pivotar (girando múltiples columnas en filas o viceversa).
- Dividir una columna en varias (por ejemplo, columna "Nombre: García, Miguel"; pasar a dos columnas "Nombre: Miguel" y "Apellido: García").
- La aplicación de cualquier forma, simple o compleja, de validación de datos, y la consiguiente aplicación de la acción que en cada caso se requiera:
 - Datos OK: Entregar datos a la siguiente etapa (Carga).
 - Datos erróneos: Ejecutar políticas de tratamiento de excepciones (por ejemplo, rechazar el registro completo, dar al campo erróneo un valor nulo o un valor centinela).

1.8.3. Cargar

La fase de carga es el momento en el cual los datos de la fase anterior (transformación) son cargados en el sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes. En algunas bases de datos se sobrescribe la información antigua con nuevos datos. Los Data Warehouse mantienen un historial de los registros de manera que se pueda hacer una auditoría de los mismos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo.

Existen dos formas básicas de desarrollar el proceso de carga:²⁸

- **Acumulación simple:** La acumulación simple es la más sencilla y común, y consiste en realizar un resumen de todas las transacciones comprendidas en el período de tiempo seleccionado y transportar el resultado como una única transacción hacia el Data Warehouse, almacenando un valor calculado que consistirá típicamente en un sumatorio o un promedio de la magnitud considerada.
- **Rolling:** El proceso de Rolling por su parte, se aplica en los casos en que se opta por mantener varios niveles de granularidad. Para ello se almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones de la unidad de tiempo o diferentes niveles jerárquicos en alguna o varias de las dimensiones de la magnitud almacenada (por ejemplo, totales diarios, totales semanales, totales mensuales, etc.).

La fase de carga interactúa directamente con la base de datos de destino. Al realizar esta operación se aplicarán todas las restricciones y triggers (disparadores) que se hayan definido en ésta (por ejemplo, valores únicos, integridad referencial, campos obligatorios, rangos de valores). Estas restricciones y triggers (si están bien definidos) contribuyen a que se garantice la calidad de los datos en el proceso ETL, y deben ser tenidos en cuenta.

²⁸ *Ibíd.*

CAPÍTULO 2

MODELOS DIMENSIONALES

2.1. Definición de un modelo dimensional

El modelo dimensional es una adaptación especializada del modelo relacional, solía representar datos en depósitos de datos, en un camino que los datos fácilmente pueden ser resumidos usando consultas OLAP. En el modelo dimensional, una base de datos consiste en una mesa sola grande de los hechos que son descritos usando dimensiones y medidas. Una dimensión proporciona el contexto de un hecho (como quien participó, cuando y donde pasó, y su tipo) y es usado en preguntas al grupo hechos relacionados juntos. Las dimensiones tienden a ser discretas y son a menudo jerárquicas; por ejemplo, la posición (ubicación) podría incluir el edificio, el estado, y el país.

Se habla de modelos dimensionales al diseñar Data Marts, bodegas de datos y sistemas de inteligencia de negocios sobre tecnología de Data Warehousing. Teniendo un fin analítico todos estos sistemas, es necesario utilizar otras técnicas de modelado que permitan registrar las relaciones existentes entre los datos para ofrecer una fuente de consulta que facilite el análisis y toma de decisiones.

Un indicador es una cantidad que describe el hecho, como el ingreso. Es importante que los indicadores significativamente

puedan ser agregados, por ejemplo, el ingreso de ubicaciones diferentes pueden ser añadidas juntas.

En una consulta OLAP, las dimensiones son escogidas y los hechos son agrupados y añadidos juntos para crear un reporte.

De esta manera, es común empleado términos tales como dimensión, nivel, jerarquía, *fact table*, granularidad, entre otros. Todos estos objetos siguen un patrón llamado el **modelo estrella** (ver Figura No. 2.1), en el que una tabla de hechos o medidas (*fact table*) es el centro de todo el modelo y está unida a múltiples dimensiones que contextualizan las medidas registradas.

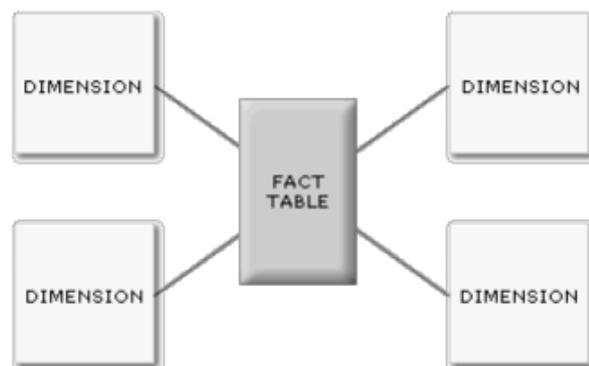


Figura No. 2.1: Ejemplo de un modelo estrella²⁹

Sus elementos principales son:

- Hechos,
- Dimensiones

²⁹ VERÁSTEGUI, Hazbleydi. Modelo Dimensional de datos [PDF]. http://www.db-system.com/pls/portal/docs/PAGE/SITIOWWWDB/ARTICULOS/MODELADO%20DIMENSIONAL%20DE%20DATOS_V2.PDF. [Ref. 03 enero 2011].

Los Hechos serían aquéllos datos que nos proporcionan una información cuantitativa sobre las características del Negocio que queremos analizar. Su finalidad es proporcionar información necesaria para la gestión, facilitando el conocimiento del Negocio o Proceso a modelar, y fundamentar, entre otras, la toma de decisiones, etc.

Las Dimensiones pueden estar jerarquizadas o no, y se pueden representar en un esquema en árbol.

La relación entre los Hechos y las Dimensiones tiene en cuenta la “Granularidad”. Definimos la “Granularidad” como el menor grado de detalle de nuestro análisis. Otra forma de definirlo es cómo el menor nivel al que existe relación entre los Dimensiones y el conjunto de Hechos.

La tecnología Data Warehouse debido a su orientación analítica, impone un procesamiento y pensamiento distinto, la cual se sustenta por un modelamiento de Bases de Datos propio, conocido como Modelamiento Multidimensional, el cual busca ofrecer al usuario su visión respecto de la operación del negocio.³⁰

En el modelo multidimensional cada eje corresponde a una dimensión particular.

2.2. Modelos de datos

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base

³⁰ WOLFF, Carmen G. Modelamiento multidimensional [PDF]. <http://www.inf.udec.cl/~revista/ediciones/edicion4/modmulti.PDF>. [Ref. 03 enero 2011].

formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información. Con respecto al diseño de bases de datos, el modelado de datos puede ser descrito así (Brodie 1984:20): "dados los requerimientos de información y proceso de una aplicación de uso intensivo de datos (por ejemplo, un sistema de información), construir una representación de la aplicación que capture las propiedades estáticas y dinámicas requeridas para dar soporte a los procesos deseados (por ejemplo, transacciones y consultas). Además de capturar las necesidades dadas en el momento de la etapa de diseño, la representación debe ser capaz de dar cabida a eventuales futuros requerimientos".³¹

2.2.1. Modelo Entidad Relación

ER es una técnica de diseño lógico que busca eliminar la redundancia de datos.

El modelo ER en su forma más simple implica identificar los asuntos de importancia dentro de una organización (**entidades**), las propiedades de esos asuntos (**atributos**) y cómo se relacionan entre sí (**relación**). Pero esto tiene valor solamente dentro del contexto de lo que se realiza en la empresa y en la forma de actuar de estas funciones de gestión sobre el modelo de información.

a. Entidad

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por

³¹ MORENO, Antonio. Bases de Datos: Modelos de datos [En línea]. <http://elies.rediris.es/elies9/4-2.htm>. [Ref. 07 enero 2011].

ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

b. Relación (interrelación)

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función.

La cardinalidad con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es obligatoria (total) si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es opcional (parcial). Las reglas que definen la cardinalidad de las relaciones son las reglas de negocio.

c. Atributo

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un atributo monovalente es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un atributo polivalente es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece.

2.2.2. Modelo Multidimensional

Las bases de datos multidimensionales se utilizan principalmente para crear aplicaciones OLAP y pueden verse como bases de datos de una sola tabla, su peculiaridad es que por cada dimensión tienen un campo (o columna), y otro campo por cada métrica o hecho, es decir estas tablas almacenan registros cuyos campos son de la forma:

(d1,d2,d3,...,f1,f2,f3,...)

A continuación en la Figura No. 2.2, se presenta un ejemplo de un modelo dimensional.

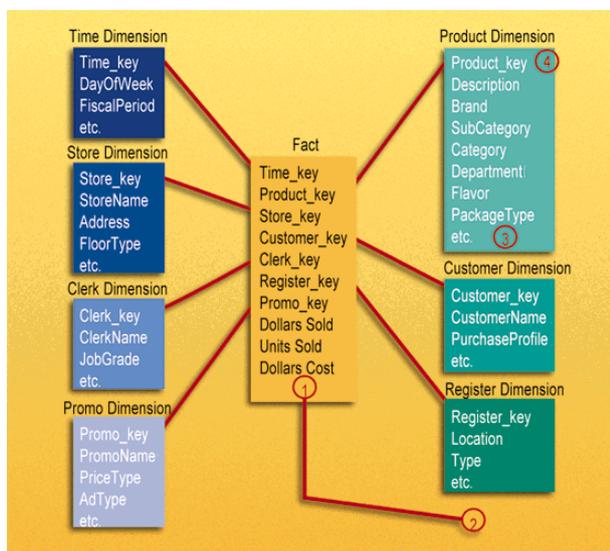


Figura No. 2.2: Ejemplo Modelo Dimensional (Modelo en Estrella)

Donde los campos 'di' hacen referencia a las dimensiones de la tabla, y los campos 'fi' a las métricas o hechos que se quiere almacenar, estudiar o analizar.

El modelo multidimensional dentro del entorno de las bases de datos, es una disciplina de diseño que se sustenta en el modelo entidad-relación y en las realidades de la ingeniería de texto y datos numéricos.

Las medidas, se registran en las tablas de hechos, siendo la llave de esta tabla, la combinación de las múltiples llaves foráneas que hacen referencia a las dimensiones que describen la ocurrencia de este hecho, en otras palabras, cada una de las llaves extranjeras en las tablas de hecho se corresponden con la llave primaria de una dimensión.³²

Las dimensiones proporcionarán por lo tanto el contexto. En un espacio multidimensional para designar un punto son necesarias todas sus coordenadas, dependiendo del número de dimensiones que tenga el espacio. Pues esto es lo que ocurre en este modelo, ya que al indicar un valor para cada dimensión se situará un hecho concreto dentro del espacio formado.

Sobre las dimensiones se pueden definir jerarquías. Estas lo que van a permitir es acceder a los datos a diferentes niveles de detalle, es decir, ver los datos a diferentes granularidades.³³

³² CEDEÑO, Alexis. Modelo Multidimensional [PDF]. www.revistaii.cujae.edu.cu/index.php/revistaind/article/download/103/82. [Ref. 14 enero 2011].

³³ MOLINA, Carlos. (2005). Imprecisión e Incertidumbre en el Modelo Multidimensional: Aplicación a la Minería de Datos. (Tesis de Doctorado –

2.2.3. Modelo Multidimensional vs. Modelo Entidad Relación

Para construir un DW se debe primero tener claro que existe una diferencia entre la estructura de la información y la semántica de la información, y que esta última es mucho más difícil de abarcar y que también es precisamente con ella con la que se trabaja en la construcción de un DW.

Aquí se encuentra la principal diferencia entre los sistemas operacionales y el DW: Cada uno de ellos es sostenido por un modelo de datos diferente. Los sistemas operacionales se sustentan en el Modelo Entidad Relación (MER) y DW trabaja con el Modelo Multidimensional.

Un modelo de datos multidimensional contiene un conjunto de medidas numéricas objeto de análisis. Dichas medidas dependen de una serie de dimensiones. Cada medida particular es un punto en un espacio multidimensional, en el que los valores de cada dimensión se suelen jerarquizar.

La forma más elevada del modelado entidad relación es eliminar toda la redundancia en los datos. Es seguro decir que el éxito de procesamiento de transacciones en bases de datos relacionales se debe principalmente a la disciplina de los modelos entidad relación.

Sin embargo, en nuestro afán de hacer el procesamiento de transacciones eficientes, hemos perdido de vista nuestro, la más

importante meta original. Hemos creado bases de datos que no se puede consultar, incluso nuestro simple para la toma de ejemplo, se crea una base de datos de docenas de tablas que están vinculadas entre sí por una tela de araña desconcertante de combinaciones. El modelo entidad relación de la empresa cuenta con cientos de entidades lógicas.

La diferencia entre el modelo entidad relación y entre el modelo dimensional es abismal. Muchos diseñadores reaccionan a esto diciendo: "Debe haber menos información en la estrella" o "La combinación en estrella es utilizado para los resúmenes de alto nivel." Ambas afirmaciones son falsas.

2.3. Tipos de Esquemas

2.3.1. Esquema en estrella

En general, el modelo multidimensional también se conoce con el nombre de esquema estrella, pues su estructura base es similar: una tabla central y un conjunto de tablas que la atienden radialmente.

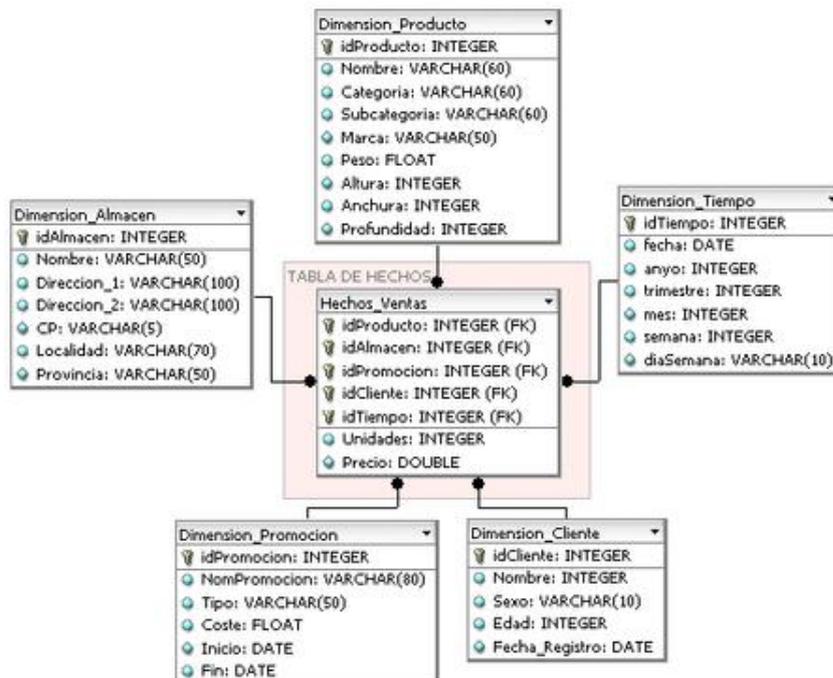


Figura No. 2.3: Ejemplo Esquema en Estrella³⁴

El esquema estrella presentado en la Figura No. 2.3, deriva su nombre del hecho que su diagrama forma una estrella, con puntos radiales desde el centro. El centro de la estrella consiste de una o más tablas fact (Hechos), y las puntas de la estrella son las tablas lock-up (Dimensiones).

Este modelo entonces, resulta ser asimétrico, pues hay una tabla dominante en el centro con varias conexiones a las otras tablas. Las tablas lock-up tienen sólo la conexión a la tabla fact y ninguna más

Las tablas de dimensiones tendrán siempre una clave primaria simple, mientras que en la tabla de hechos, la clave principal estará compuesta por las claves principales de las tablas dimensionales.

³⁴ Wikipedia. Esquema en Estrella [En línea]. http://es.wikipedia.org/wiki/Esquema_en_estrella. [Ref. 15 enero 2011]

2.3.2. Esquema en copo de nieve

El esquema en copo de nieve (snowflake schema) es un esquema de representación derivado del esquema en estrella, en el que las tablas de dimensión se normalizan en múltiples tablas.

Seguidamente se presenta un esquema copo de nieve, donde se presenta un esquema normalizado en la Figura No. 2.4:

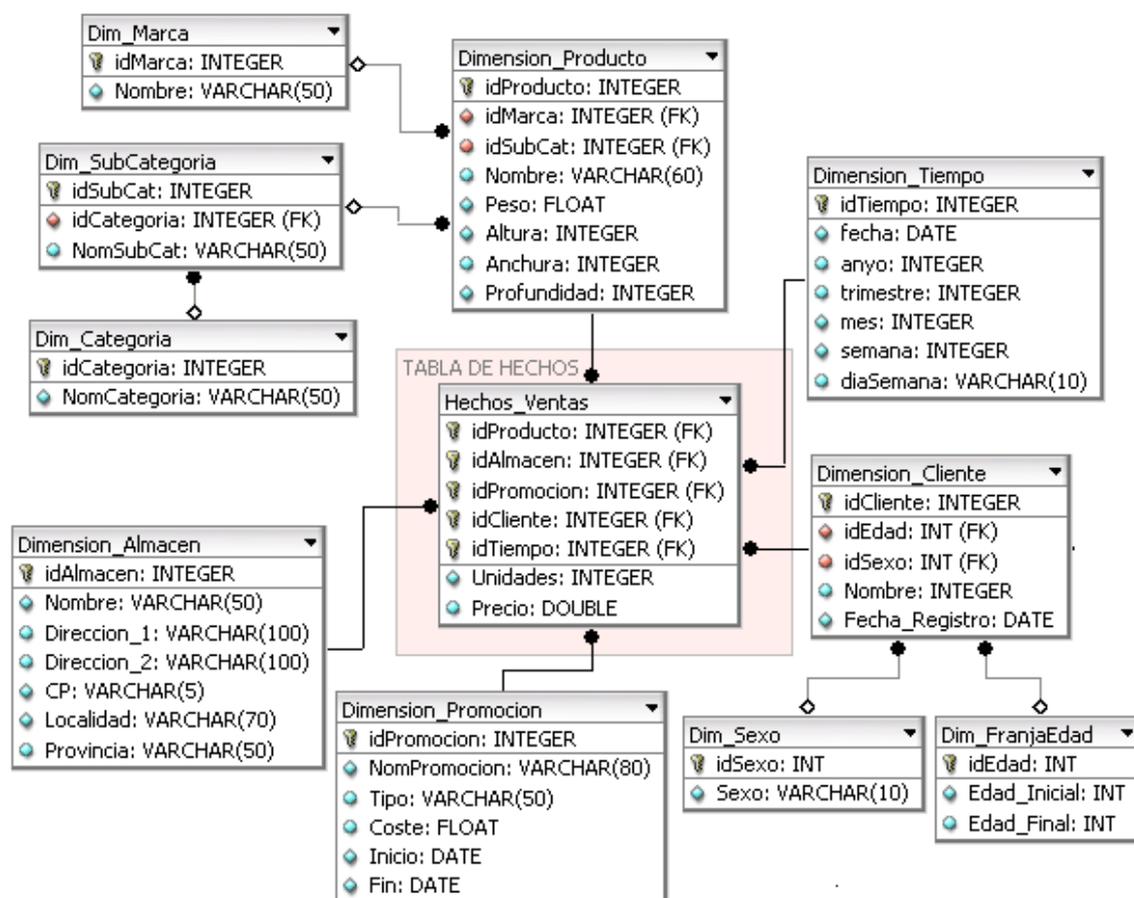


Figura No. 2.4: Ejemplo Esquema Copo de Nieve³⁵

³⁵ Wikipedia. Archivo: Esquema en copo de nieve [PNG]. http://es.wikipedia.org/wiki/Archivo:Esquema_en_copo_de_nieve.png. [Ref. 20 enero 2011]

Por esta razón, la tabla de hechos deja de ser la única tabla del esquema que se relaciona con otras tablas, y aparecen nuevos joins gracias a que las dimensiones de análisis se representan ahora en tablas de dimensión normalizadas. En la estructura dimensional normalizada, la tabla que representa el nivel base de la dimensión es la que hace join directamente con la tabla de hechos.³⁶

2.4. Tipos de Tablas

2.4.1. Tabla Fact o de hechos

Cada almacén de datos incluye una o varias tablas de hechos. Una tabla de hechos (o tabla fact) es la tabla central de un esquema dimensional (en estrella o en copo de nieve) y contiene los valores de las medidas de negocio. Cada medida se toma mediante la intersección de las dimensiones que la definen, dichas dimensiones estarán reflejadas en sus correspondientes tablas de dimensiones que rodearán la tabla de hechos y estarán relacionadas con ella.

La tabla de hechos contiene las claves subrogadas de aquellas dimensiones que definen su nivel de detalle, y los indicadores. Nada más. Por lo tanto, antes de crear la tabla de hechos debe entenderse perfectamente la información que se guardará.

³⁶ Information Management. Diseño de un Data Warehouse: estrella y copo de nieve [En línea].
<http://informationmanagement.wordpress.com/2007/11/19/disenodeun-data-warehouse-estrella-y-copo-de-nieve/>. [Ref. 20 enero 2011].

2.4.2. Tabla Lock-Up o de dimensiones y granularidad

En un almacén de datos o un sistema OLAP, la construcción de Cubos OLAP requiere de una tabla de hechos y varias tablas de dimensiones, éstas acompañan a la tabla de hechos y determinan los parámetros (dimensiones) de los que dependen los hechos registrados en la tabla de hechos. Las tablas de dimensiones son elementos que contienen atributos (o campos) que se utilizan para restringir y agrupar los datos almacenados en una tabla de hechos.

Estos datos sobre dimensiones son parámetros de los que dependen otros datos que serán objeto de estudio y análisis y que están contenidos en la tabla de hechos. Las tablas de dimensiones ayudan a realizar ese estudio/análisis aportando información sobre los datos de la tabla de hechos, por lo que puede decirse que en un cubo OLAP, la tabla de hechos contiene los datos de interés y las tablas de dimensiones contienen metadatos sobre dichos hechos.

2.5. Profundizar en el diseño

2.5.1. La dimensión Tiempo

En cualquier Data Warehouse se pueden encontrar varios cubos con sus tablas de hechos repletas de registros sobre alguna variable de interés para el negocio que debe ser estudiada. Como ya se ha comentado, cada tabla de hechos estará rodeada de varias tablas de dimensiones, según que parámetros sirvan mejor para realizar el análisis de los hechos que se quieren estudiar. Un parámetro que casi con toda probabilidad será común a todos los cubos es el

tiempo, ya que lo habitual es almacenar los hechos conforme van ocurriendo a lo largo del tiempo, obteniéndose así una serie temporal de la variable a estudiar.

2.5.2. Dimensiones que varían lentamente en el tiempo

Son aquellas dimensiones que se mantienen “casi” constantes en el tiempo y que pueden preservar la estructura dimensional independiente del tiempo, con sólo agregados menores relativos para capturar la naturaleza cambiante del tiempo.

Cuando se encuentra una de estas dimensiones se está haciendo una de las siguientes fundamentales tres elecciones. Cada elección resulta en un diferente grado de seguimiento sobre el tiempo:

Tipo 1: Sobrescribir el viejo valor en el registro dimensional y por lo tanto perder la capacidad de seguir la vieja historia.

Tipo 2: Crear un registro dimensional adicional (con una nueva llave) que permita registrar el cambio presentado por el valor del atributo. De esta forma permanecerían en la base tanto el antiguo como el nuevo valor del registro con lo cual es posible segmentar la historia de la ocurrencia.

Tipo 3: Crear un campo “actual” nuevo en el registro dimensional original el cual almacene el valor del nuevo atributo, manteniendo el atributo original también. Cada vez que haya un nuevo cambio en el atributo, se modifica el campo “actual” solamente. No se mantiene un registro histórico de los cambios intermedios.

jerarquía puede pero no exige contener todos los valores existentes en la dimensión.

Se debe evitar caer en la tentación de convertir en tablas dimensionales separadas cada una de las relaciones muchos-a-uno presentes en las jerarquías. Esta descomposición es irrelevante en el planeamiento del espacio ocupado en disco y sólo dificulta el entendimiento de la estructura para el usuario final, además de destruir el desempeño del browsing.

2.6. Cubo OLAP

2.6.1. Definición

Los cubos de datos (Cubos OLAP, Cubos de información, Hypercubos) proporcionan la última pieza del rompecabezas. Un cubo agrega los hechos en cada nivel de cada dimensión en un determinado esquema de OLAP. La industria de inteligencia de negocios usa la palabra "cubo", ya que describe mejor los datos resultantes.

Debido a que el cubo contiene todos los datos de forma agregada, parece conocer las respuestas de antemano. Por ejemplo, si un usuario pide que las ventas totales por año y la ciudad, los números ya están disponibles. Si el usuario pide total de ventas por trimestre, categoría, el cliente, los números y nombres ya están disponibles. Se puede pensar en los cubos como pequeñas bases de datos especializadas que conocen las respuestas, incluso antes de hacer las preguntas.

2.6.2. Operaciones básicas

La funcionalidad de los sistemas OLAP se caracteriza por ser un análisis multidimensional de datos corporativos, que soportan los análisis del usuario y unas posibilidades de navegación, seleccionando la información a obtener. Normalmente este tipo de selecciones se ve reflejada en la visualización de la estructura multidimensional, en unos campos de selección que nos permitan elegir el nivel de agregación (jerarquía) de la dimensión, y/o la elección de un dato en concreto, la visualización de los atributos del sujeto, frente a una(s) dimensiones en modo tabla, pudiendo con ello realizar, entre otras las siguientes acciones:

- ✓ **Rotar (Swap)**
Alterar las filas por columnas (permutar dos dimensiones de análisis)
- ✓ **Bajar (Down)**
Bajar el nivel de visualización en las filas a una jerarquía inferior.
- ✓ **Detallar (Drilldown)**
Informar para una fila en concreto, de datos a un nivel inferior.
- ✓ **Expandir (Expand)**
Id. anterior sin perder la información a nivel superior para éste y el resto de los valores.
- ✓ **Colapsar (Collapse)**
Operación inversa a Expandir.

2.7. Beneficios de usar modelos dimensionales

Algunos de los beneficios son mencionados a continuación:

- Es extensible para acomodar nuevos elementos de datos inesperados y nuevas decisiones de diseño. Esto se lleva a cabo añadiendo nuevos hechos de forma inesperada, nuevas dimensiones, nuevos atributos dimensionales y cambiando datos de una cierta granularidad para pasarlos a otra.
- Múltiples fuentes. Es posible acceder a información de diversas bases de datos teniendo un DW, ya que permite consolidar los datos y poder tener acceso a la información de todas las bases en una sola.
- Información histórica. Una de las posibilidades que nos ofrece el DW es el manejo de la información histórica, mediante el uso de versiones, las cuales se van almacenando cada vez que se ejecute una actualización a los datos.
- Resumen de la información. Mediante la agregación se reduce significativamente el tamaño de los archivos; esto permite el manejo de mayores volúmenes de información en archivos más pequeños.
- Menor tiempo de respuesta. Dado que la información está ya agregada, el tiempo de respuesta se reduce considerablemente en lo referente a la generación de reportes y consultas.
- Capacidad de análisis. Es posible mostrar la información mediante gráficas o reportes globalizados, dado que los datos se encuentran ya resumidos. Esto genera que se evalúe las situaciones por las que la empresa está atravesando.

2.8. Pasos para el modelado multidimensional

Para la realización del DW se utilizará la metodología HEFESTO V2.0 la cual se describe a continuación:

2.8.1. Metodología HEFESTO

a. Introducción

HEFESTO es una metodología cuya propuesta está fundamentada en una muy amplia investigación, comparación de metodologías existentes, experiencias propias en procesos de confección de almacenes de datos. Cabe destacar que HEFESTO está en continua evolución, y se han tenido en cuenta, como gran valor agregado, todos los feedbacks que han aportado quienes han utilizado esta metodología en diversos países y con diversos fines.

b. Descripción

La metodología HEFESTO puede resumirse en la Figura No. 2.6:



Figura No. 2.6: Pasos metodología HEFESTO

Como se puede apreciar, se comienza recolectando las necesidades de información de los usuarios y se obtienen las preguntas claves del negocio. Luego, se deben identificar los indicadores resultantes de los interrogativos y sus respectivas perspectivas de análisis, mediante las cuales se construirá el modelo conceptual de datos del DW.

Después, se analizarán los OLTP para determinar cómo se construirán los indicadores, señalar las correspondencias con los datos fuentes y para seleccionar los campos de estudio de cada perspectiva. Una vez hecho esto, se pasará a la construcción del modelo lógico del depósito, en donde se definirá cuál será el tipo de esquema que se implementará. Seguidamente, se confeccionarán las tablas de dimensiones y las tablas de hechos, para luego efectuar sus respectivas uniones.

c. Características

Esta metodología cuenta con las siguientes características:

- ✓ Los objetivos y resultados esperados en cada fase se distinguen fácilmente y son sencillos de comprender.
- ✓ Se basa en los requerimientos de los usuarios, por lo cual su estructura es capaz de adaptarse con facilidad y rapidez ante los cambios en el negocio.
- ✓ Reduce la resistencia al cambio, ya que involucra a los usuarios finales en cada etapa para que tome decisiones respecto al comportamiento y funciones del DW.
- ✓ Utiliza modelos conceptuales y lógicos, los cuales son sencillos de interpretar y analizar.

- ✓ Es independiente del tipo de ciclo de vida que se emplee para contener la metodología.
- ✓ Es independiente de las herramientas que se utilicen para su implementación.
- ✓ Es independiente de las estructuras físicas que contengan el DW y de su respectiva distribución.
- ✓ Cuando se culmina con una fase, los resultados obtenidos se convierten en el punto de partida para llevar a cabo el paso siguiente.
- ✓ Se aplica tanto para Data Warehouse como para Data Mart.

d. Pasos y aplicación metodológica

d.1. Análisis de requerimientos

Lo primero que se hará será identificar los requerimientos de los usuarios a través de preguntas que expliciten los objetivos de su organización. Luego, se analizarán estas preguntas a fin de identificar cuáles serán los indicadores y perspectivas que serán tomadas en cuenta para la construcción del DW. Finalmente se confeccionará un modelo conceptual en donde se podrá visualizar el resultado obtenido en este primer paso.

d.1.1. Identificar preguntas

El primer paso comienza con el acopio de las necesidades de información, el cual puede llevarse a cabo a través de muy variadas y diferentes técnicas, cada una de las cuales poseen características inherentes y específicas, como por ejemplo entrevistas, cuestionarios, observaciones, etc.

El objetivo principal de esta fase, es la de obtener e identificar las necesidades de información clave de alto nivel, que es esencial para llevar a cabo las metas y estrategias de la empresa, y que facilitará una eficaz y eficiente toma de decisiones.

d.1.2. Identificar indicadores y perspectivas

Una vez que se han establecido las preguntas de negocio, se debe proceder a su descomposición para descubrir los indicadores que se utilizarán y las perspectivas de análisis que intervendrán.

Para ello, se debe tener en cuenta que los indicadores, para que sean realmente efectivos son, en general, valores numéricos y representan lo que se desea analizar concretamente, por ejemplo: saldos, promedios, cantidades, sumatorias, fórmulas, etc.

En cambio, las perspectivas se refieren a los objetos mediante los cuales se quiere examinar los indicadores, con el fin de responder a las preguntas planteadas, por ejemplo: clientes, proveedores, sucursales, países, productos, rubros, etc. Cabe destacar, que el Tiempo es muy comúnmente una perspectiva.

d.1.3. Modelo Conceptual

En esta etapa, se construirá un modelo conceptual a partir de los indicadores y perspectivas obtenidas en el paso anterior. A través de este modelo, se podrá observar con claridad cuáles son los alcances del proyecto, para luego poder trabajar sobre ellos, además al poseer un alto nivel de definición de los datos, permite que pueda

ser presentado ante los usuarios y explicado con facilidad. La representación gráfica del modelo conceptual se presenta en la Figura No. 2.7:

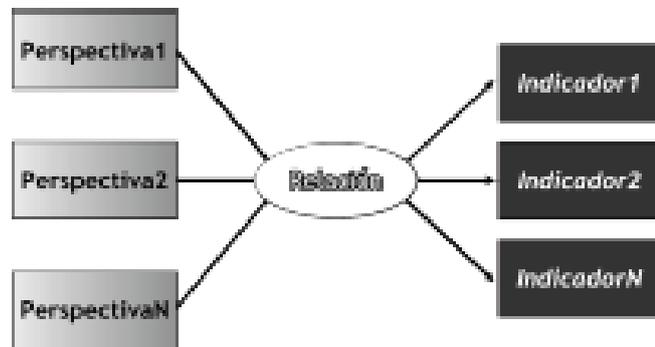


Figura No. 2.7: Modelo Conceptual

A la izquierda se colocan las perspectivas seleccionadas, que serán unidas a un óvalo central que representa y lleva el nombre de la relación que existe entre ellas. La relación, constituye el proceso o área de estudio elegida. De dicha relación y entrelazadas con flechas, se desprenden los indicadores, estos se ubican a la derecha del esquema.

d.2. Análisis de los OLTP

Seguidamente, se analizarán las fuentes OLTP para determinar cómo serán calculados los indicadores y para establecer las respectivas correspondencias entre el modelo conceptual creado en el paso anterior y las fuentes de datos. Luego, se definirán qué campos se incluirán en cada perspectiva. Finalmente, se ampliará el modelo conceptual con la información obtenida en este paso.

d.2.1. Conformar indicadores

En este paso se deberán explicitar cómo se calcularán los indicadores, definiendo los siguientes conceptos para cada uno de ellos:

- ✓ Hecho/s que lo componen, con su respectiva fórmula de cálculo. Por ejemplo: Hecho1 + Hecho2.
- ✓ Función de sumarización que se utilizará para su agregación. Por ejemplo: SUM, AVG, COUNT, etc.

d.2.2. Establecer correspondencias

El objetivo de este paso, es el de examinar los OLTP disponibles que contengan la información requerida, como así también sus características, para poder identificar las correspondencias entre el modelo conceptual y las fuentes de datos. La idea es, que todos los elementos del modelo conceptual estén correspondidos en los OLTP.

d.2.3. Nivel de granularidad

Una vez que se han establecido las relaciones con los OLTP, se deben seleccionar los campos que contendrá cada perspectiva, ya que será a través de estos por los que se examinarán y filtrarán los indicadores.

Para ello, basándose en las correspondencias establecidas en el paso anterior, se debe presentar a los usuarios los datos de análisis disponibles para cada perspectiva.

d.2.4. Modelo Conceptual ampliado

En este paso, y con el fin de graficar los resultados obtenidos en los pasos anteriores, se ampliará el modelo conceptual, colocando bajo cada perspectiva los campos seleccionados y bajo cada indicador su respectiva fórmula de cálculo como se muestra en la Figura No. 2.8:

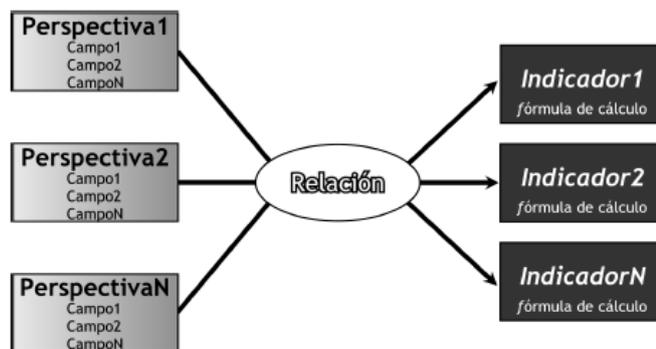


Figura No. 2.8: Modelo Conceptual Ampliado

d.3. Modelo Lógico del Data Warehouse

A continuación, se confeccionará el modelo lógico de la estructura del DW, teniendo como base el modelo conceptual que ya ha sido creado. Para ello, primero se definirá el tipo de modelo que se utilizará y luego se llevarán a cabo las acciones propias al caso, para diseñar las tablas de dimensiones y de hechos. Finalmente, se realizarán las uniones pertinentes entre estas tablas.

d.3.1. Tipo de Modelo Lógico del DW

Se debe seleccionar cuál será el tipo de esquema que se utilizará para contener la estructura del depósito de datos, que se adapte mejor a los requerimientos y necesidades de los usuarios. Es muy importante definir objetivamente si se empleará un esquema en

estrella, constelación o copo de nieve, ya que esta decisión afectará considerablemente la elaboración del modelo lógico.

d.3.2. Tablas de dimensiones

En este paso se deben diseñar las tablas de dimensiones que formaran parte del DW.

Cada perspectiva definida en el modelo conceptual constituirá una tabla de dimensión. Para ello deberá tomarse cada perspectiva con sus campos relacionados y realizarse el siguiente proceso:

- ✓ Se elegirá un nombre que identifique la tabla de dimensión.
- ✓ Se añadirá un campo que represente su clave principal.
- ✓ Se redefinirán los nombres de los campos si es que no son lo suficientemente intuitivos.

Esto se lo puede visualizar en la Figura No. 2.9:

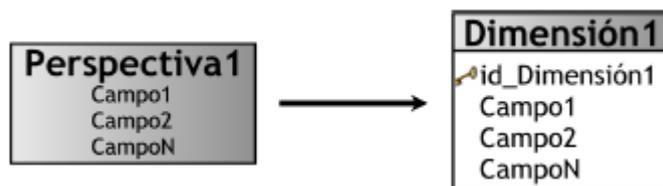


Figura No. 2.9: Diseño de Tablas de Dimensiones

Para los esquemas copo de nieve, cuando existan jerarquías dentro de una tabla de dimensión, esta tabla deberá ser normalizada. Por ejemplo, se tomará como referencia la siguiente tabla de dimensión y sus respectivas relaciones padre-hijo entre sus campos, como se muestra en la Figura No. 2.10:

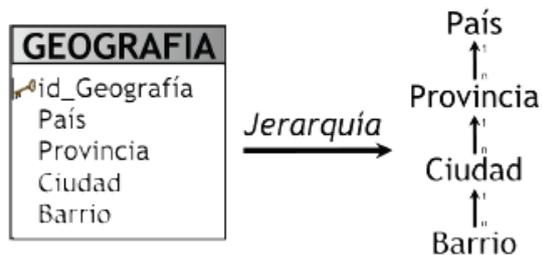


Figura No. 2.10: Jerarquía de “Geografía”

Entonces, en la Figura No. 2.11 se puede observar la tabla normalizada:

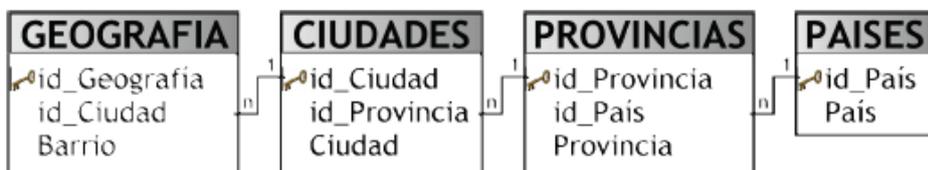


Figura No. 2.11: Normalización de “Geografía”

d.3.3. Tablas de hechos

En este paso, se definirán las tablas de hechos, que son las que contendrán los hechos a través de los cuales se construirán los indicadores de estudio.

Para los esquemas en estrella y copo de nieve, se realizará lo siguiente:

- ✓ Se le deberá asignar un nombre a la tabla de hechos que represente la información analizada, área de investigación, negocio enfocado, etc.

- ✓ Se definirá su clave primaria, que se compone de la combinación de las claves primarias de cada tabla de dimensión relacionada.
- ✓ Se crearán tantos campos de hechos como indicadores se hayan definido en el modelo conceptual y se les asignará los mismos nombres que estos. En caso que se prefiera, podrán ser nombrados de cualquier otro modo.

Como se muestra en la Figura No. 2.12:

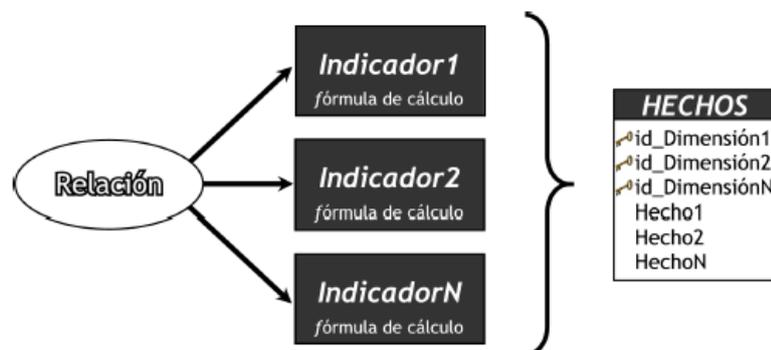


Figura No. 2.12: Tabla de hechos

Para los esquemas constelación se realizará lo siguiente:

- ✓ Las tablas de hechos se deben confeccionar teniendo en cuenta el análisis de las preguntas realizadas por los usuarios en pasos anteriores y sus respectivos indicadores y perspectivas.
- ✓ Cada tabla de hechos debe poseer un nombre que la identifique, contener sus hechos correspondientes y su clave debe estar formada por la combinación de las claves de las tablas de dimensiones relacionadas.

Al diseñar las tablas de hechos, se deberá tener en cuenta:

Caso 1: Si en dos o más preguntas de negocio figuran los mismos indicadores pero con diferentes perspectivas de análisis, existirán tantas tablas de hechos como preguntas cumplan esta condición. Por ejemplo:

“Analizar el **Indicador1** por **Perspectiva1** y por **Perspectiva2**”

“Analizar el **Indicador1** por **Perspectiva2** y por **Perspectiva3**”

Entonces se obtendrá:

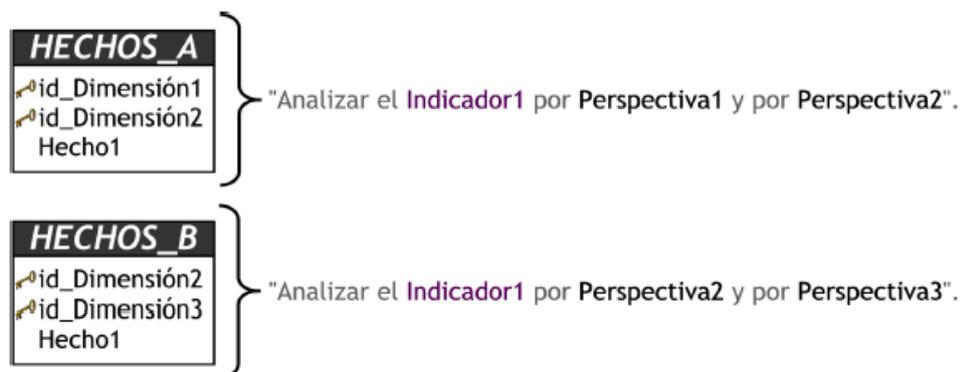


Figura No. 2.13: Caso 1: Diseño de tablas de hechos

Caso 2: Si en dos o más preguntas de negocio figuran diferentes indicadores con diferentes perspectivas de análisis, existirán tantas tablas de hechos como preguntas cumplan esta condición. Por ejemplo:

“Analizar el **Indicador1** por **Perspectiva1** y por **Perspectiva2**”

“Analizar el **Indicador2** por **Perspectiva2** y por **Perspectiva3**”

Entonces se obtendrá:

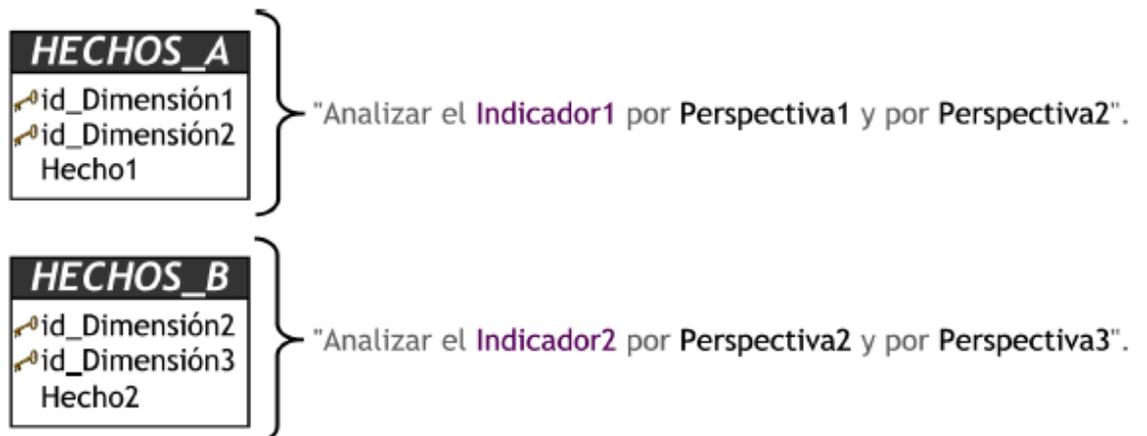


Figura No. 2.14: Caso 2: Diseño de tablas de hechos

Caso 3: Si el conjunto de preguntas de negocio cumplen con las condiciones de los dos puntos anteriores se deberán unificar aquellos interrogantes que posean diferentes indicadores pero iguales perspectivas de análisis, para luego reanudar el estudio de las preguntas. Por ejemplo:

"Analizar el **Indicador1** por **Perspectiva1** y por **Perspectiva2**"

"Analizar el **Indicador2** por **Perspectiva1** y por **Perspectiva2**"

Se unificarán en:

"Analizar el **Indicador1** y el **Indicador2** por **Perspectiva1** y por **Perspectiva2**"

d.3.4. Uniones

Para los dos tipos de esquemas (estrella, copo de nieve), se realizarán las uniones correspondientes entre sus tablas de dimensiones y sus tablas de hechos.

d.4. Integración de Datos

Una vez construido el modelo lógico, se deberá proceder a poblarlo con datos, utilizando técnicas de limpieza y calidad de datos, procesos ETL, etc.; luego se definirán las reglas y políticas para su respectiva actualización, así como también los procesos que la llevarán a cabo.

d.4.1. Carga Inicial

Debemos en este paso realizar la Carga Inicial al DW, poblando el modelo de datos que hemos construido anteriormente. Para lo cual debemos llevar adelante una serie de tareas básicas, tales como limpieza de datos, calidad de datos, procesos ETL, etc.

Se debe evitar que el DW sea cargado con valores faltantes o anómalos, así como también se deben establecer condiciones y restricciones para asegurar que solo se utilicen los datos de interés.

Es importante tener presente, que al cargar los datos en las tablas de hechos pueden utilizarse pre agregaciones, ya sea al nivel de granularidad de la misma o a otros niveles diferentes.

2.8.1.4.4.1.1. Arquitectura de flujo de datos

El sistema Data Warehouse tiene dos arquitecturas principales: la arquitectura de flujo de datos y la arquitectura del sistema. La arquitectura de flujo de datos se refiere a cómo los almacenes de datos (Data Stores) se ubican dentro del Data Warehouse y cómo los datos fluyen de los sistemas de fuente a los usuarios a

través de estos almacenes de datos. La arquitectura del sistema se refiere a la configuración física de los servidores, redes, software, almacenamiento, y los clientes. En este capítulo, me referiré a la arquitectura de flujo de datos.

En Data Warehouse, la arquitectura de flujo de datos es la configuración de los almacenes de datos dentro de un sistema Data Warehouse, junto con la disposición de cómo fluyen los datos de los sistemas fuente a través de estos almacenes de datos a las aplicaciones utilizadas por los usuarios finales. Esto incluye cómo los flujos de datos son controlados, registrados y supervisados, así como el mecanismo para garantizar la calidad de los datos en los almacenes de datos.

Los almacenes de datos son componentes importantes de la arquitectura de flujo de datos. Un almacén de datos es una o más bases de datos o ficheros que contengan datos de Data Warehouse, organizados en un formato determinado y participan en los procesos de almacenamiento de datos.

Basado en el formato de datos, se puede clasificar mediante como se almacenan los datos en cuatro tipos:

- ✓ Un Stage es un almacén de datos internos utilizados para la transformación y preparación de los datos obtenidos de los sistemas fuente, antes de cargar los datos a otros almacenes de datos en un almacén de datos.
- ✓ Un almacén de datos normalizado (NDS) es un maestro interno que almacena datos en la forma de una o más bases de datos relacionales normalizados con el fin de integrar datos de varias fuentes de los sistemas de captura en una etapa,

antes de cargar los datos a un usuario frente a almacenar datos .

- ✓ Un almacén de datos operacionales (ODS) es un almacén de datos híbrido, en forma de una o más bases de datos relacional normalizada, que contiene los datos de la transacción y la versión más reciente de los datos maestros, con el propósito de apoyar las aplicaciones operativas.
- ✓ Un almacén de datos dimensional (DDS) es un repositorio de cara al usuario de datos, en el formato de una o más bases de datos relacionales, donde se dispone de los datos en formato tridimensional con el fin de apoyar las consultas analíticas.

La arquitectura de flujo de datos está diseñada en base a las necesidades de datos de las aplicaciones, incluyendo los requisitos de calidad de datos. Las aplicaciones Data Warehouse requieren datos en diferentes formatos. Estos formatos determinan los almacenes de datos que es necesario tener. Si las aplicaciones requieren el formato tridimensional, entonces usted necesita construir un DDS. Si las aplicaciones requieren de un formato normalizado para fines operativos, entonces usted necesita construir un ODS. Si la aplicación requiere el formato multidimensional, entonces usted necesita construir una MDB. Una vez que determine los almacenes de datos que usted necesita construir, se pueden diseñar los ETL para poblar los almacenes de datos. A continuación, crear un mecanismo de calidad de los datos para asegurarse de que los datos del almacén de datos son correctos y completos.

En una sola arquitectura DDS, tiene un solo almacén de datos dimensional. El DDS se compone de uno o varios Data Marts dimensionales. Un Data Mart dimensional es un grupo

de tablas de hechos relacionados y sus correspondientes tablas de dimensiones que contienen las mediciones de eventos de negocios, clasificados por sus dimensiones. Un paquete ETL extrae los datos de los sistemas de fuente diferente y los pone en el Stage.

Un Stage es un lugar donde almacenar los datos que extrae del sistema de almacenamiento temporal, antes de su procesamiento posterior. Un Stage es necesario cuando la transformación es compleja (en otras palabras, no se puede hacer sobre la marcha en un solo paso en la memoria), cuando el volumen de datos es grande (en otras palabras, no es suficiente para poner en la memoria), o cuando datos de varios sistemas de origen llegan en momentos diferentes (en otras palabras, no extraída por una sola ETL). Un Stage también es necesario si usted necesita reducir al mínimo el tiempo necesario para extraer los datos del sistema de origen. En otras palabras, los procesos ETL vuelcan los datos extraídos al disco y se desconectan del sistema de origen tan pronto como sea posible, y luego en su propio tiempo pueden procesar los datos.

La forma física de un Stage puede ser una base de datos o archivos. El ETL que extrae datos de la fuente del sistema inserta los datos en una base de datos o la escribe como archivos. Un segundo paquete ETL recoge los datos de la primera etapa, integra los datos del sistema de origen diferentes, aplica algunas normas de calidad de datos, y pone los datos consolidados en el DDS.

d.4.2. Actualización

Cuando se haya cargado en su totalidad el DW, se deben establecer sus políticas y estrategias de actualización o refresco de datos.

Una vez realizado esto, se tendrán que llevar a cabo las siguientes acciones:

- ✓ Especificar las tareas de limpieza de datos, calidad de datos, procesos ETL, etc., que deberán realizarse para actualizar los datos del DW.
- ✓ Especificar de forma general y detallada las acciones que deberá realizar cada software.

CAPÍTULO 3

DISEÑO DEL CUBO DE INFORMACIÓN EN SQL SERVER 2008

3.1. Modelo Multidimensional

Después de haber definido la información que va a ser útil para el desarrollo del cubo de información para el control del desempeño de la empresa Babel Software, es primordial empezar con el análisis de los requerimientos del cubo, para determinar sus dimensiones, granularidad, jerarquías, hechos, etc. que deben estar presentes en el modelo dimensional.

3.1.1. Análisis de Requerimientos

a. Identificar Preguntas

Se indago al usuario que utilizará directamente el reporte en busca de sus necesidades de información en base a lo cual se delimitó el aplicativo a un solo proceso que es el de asignación de tiempos por proyecto a los empleados, con la finalidad de obtener cuadros estadísticos e históricos que permitan la medición y comparación del porcentaje laboral de los empleados de la empresa a través de ciertas perspectivas que permitan la toma de decisiones basadas en la información obtenida.

A continuación, se procedió a identificar que era lo que le interesaba conocer acerca de este proceso; cuales son los indicadores que requieren y con qué perspectivas o variables; en las indagaciones se dio como resultado de que el indicador a usarse es el porcentaje de asignación mensual de labores de los empleados en proyectos de la empresa.

Luego se le preguntó cuáles serían las variables o perspectivas desde las cuales se consultará dicho indicador. Gracias al amplio conocimiento que el usuario posee sobre el desarrollo de cubos OLAP, supo explicar detalladamente que parte del negocio le gustaría ver inmersa en el desarrollo del mismo:

- Se desea conocer cuál es el porcentaje de asignación de labores de un empleado dentro de un proyecto en un período determinado. Añadiendo datos como la nacionalidad del proyecto, la empresa cliente que contrato el proyecto, el rol del empleado desempeñado en el proyecto y la empresa (sucursal) donde se desarrolló el proyecto. En otras palabras: “Asignación laboral de cada empleado por proyecto, nacionalidad, rol, empresa, oficina en un tiempo determinado”

Debido a que la dimensión Tiempo es un elemento fundamental en el DW, se hizo hincapié en él. Determinando de que la asignación del empleado debe ser estandarizada en días hábiles de labores. Además, se puso mucho énfasis en dejar en claro al usuario, a través de ejemplos prácticos, que es este componente el que permitirá tener varias versiones de los datos a fin de realizar un correcto análisis posterior.

Como se puede apreciar, las necesidades de información expuestas están acorde a los objetivos que desempeña el usuario para quien se elaborará el proyecto, ya que es precisamente esta información requerida la que proveerá un ámbito para la toma de decisiones, que en este caso permitirá analizar la asignación de talento humano (empleados) a los distintos proyectos de la empresa y su desempeño en el mismo.

b. Identificar indicadores y perspectivas

En síntesis, los indicadores son:

- Porcentaje Asignación diario

Y las perspectivas de análisis son:

- Empleado
- Empresa
- Tiempo
- Nacionalidad
- Oficina
- Proyecto
- Rol

c. Modelo conceptual

El modelo conceptual resultante de los datos que se han recolectado, es el siguiente:

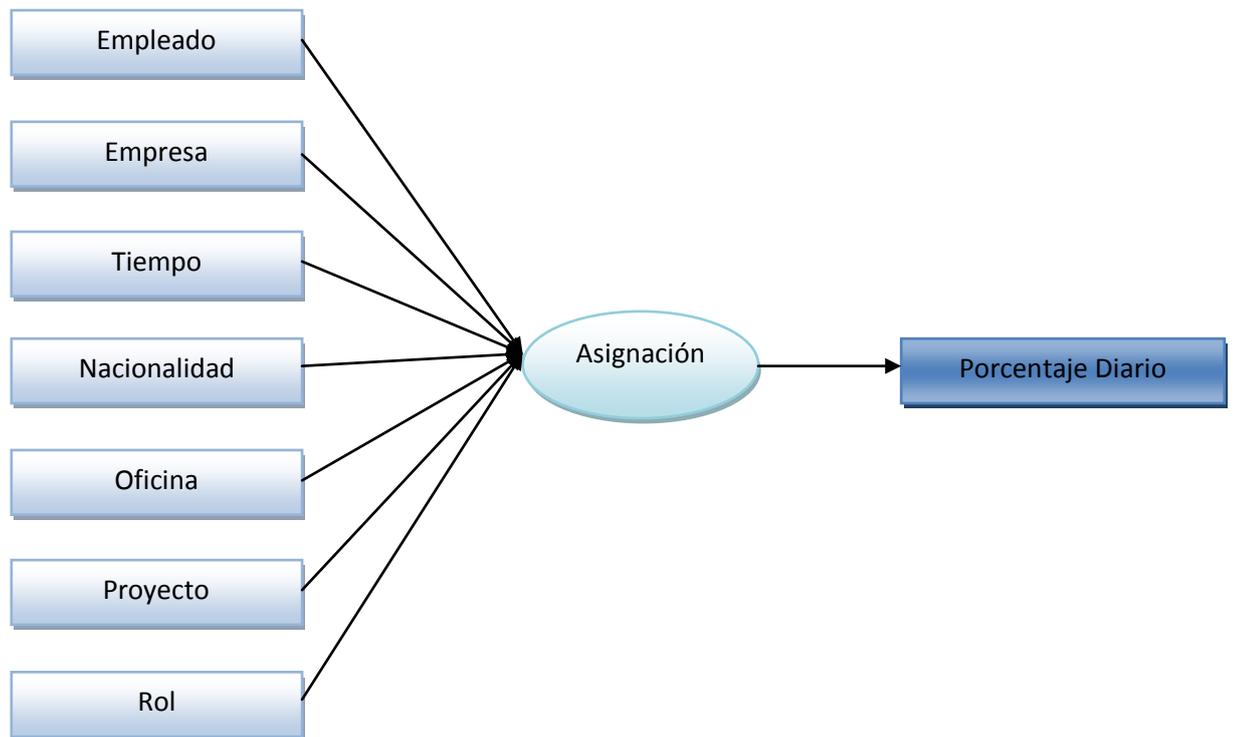


Figura No. 3.1: Modelo conceptual de porcentaje de asignación de tiempos

Como puede observarse, la relación mediante la cual se unen las diferentes perspectivas, para obtener como resultado el indicador requerido por el usuario, es precisamente “Asignación”.

3.1.2. Análisis de los OLTP

a. Conformar indicadores

El indicador se calculará de la siguiente manera:

- “Porcentaje asignación diario”

✓ Hechos:

$$\frac{\left(\frac{\text{cnt num días hab per mes} \times \text{asig emp pry}}{100} \right)}{\text{cnt num días hab mes}}$$

Ecuación No. 1: Cálculo de porcentaje de asignación diario

✓ Función de sumarización: SUM.

- Aclaración: el indicador “Porcentaje asignación diario” representa la sumatoria del porcentaje de asignación laboral que se ha establecido entre un empleado y un proyecto en un determinado tiempo, y se obtiene con el siguiente proceso:

1. Resultado de la multiplicación entre el conteo de número de días hábiles del período de labores en el mes correspondiente (cnt num días hab per mes) por el porcentaje de asignación del empleado en el proyecto (asig emp pry).
2. Resultado de la división entre la multiplicación anterior para el valor cien.
3. El resultado anterior se lo divide para el conteo de número de días hábiles del mes (cnt num días hab mes).
4. El resultado anterior se lo divide para el conteo de número de días hábiles del período de labores en el mes

correspondiente (cnt num dias hab per mes).

- El conteo de días hábiles cuenta los días laborados en un rango de fechas o del mes, sin contar fines de semana.

b. Establecer correspondencias

La base de datos transaccional que se utiliza actualmente en la empresa se denomina BabelSys, en la cual se encuentran plasmados varios de los procesos que se llevan a cabo en las labores de la misma. En este OLTP obtenemos el siguiente diagrama de entidad relación que utilizaremos en el desarrollo del DW que se indica en la Figura No. 3.2:

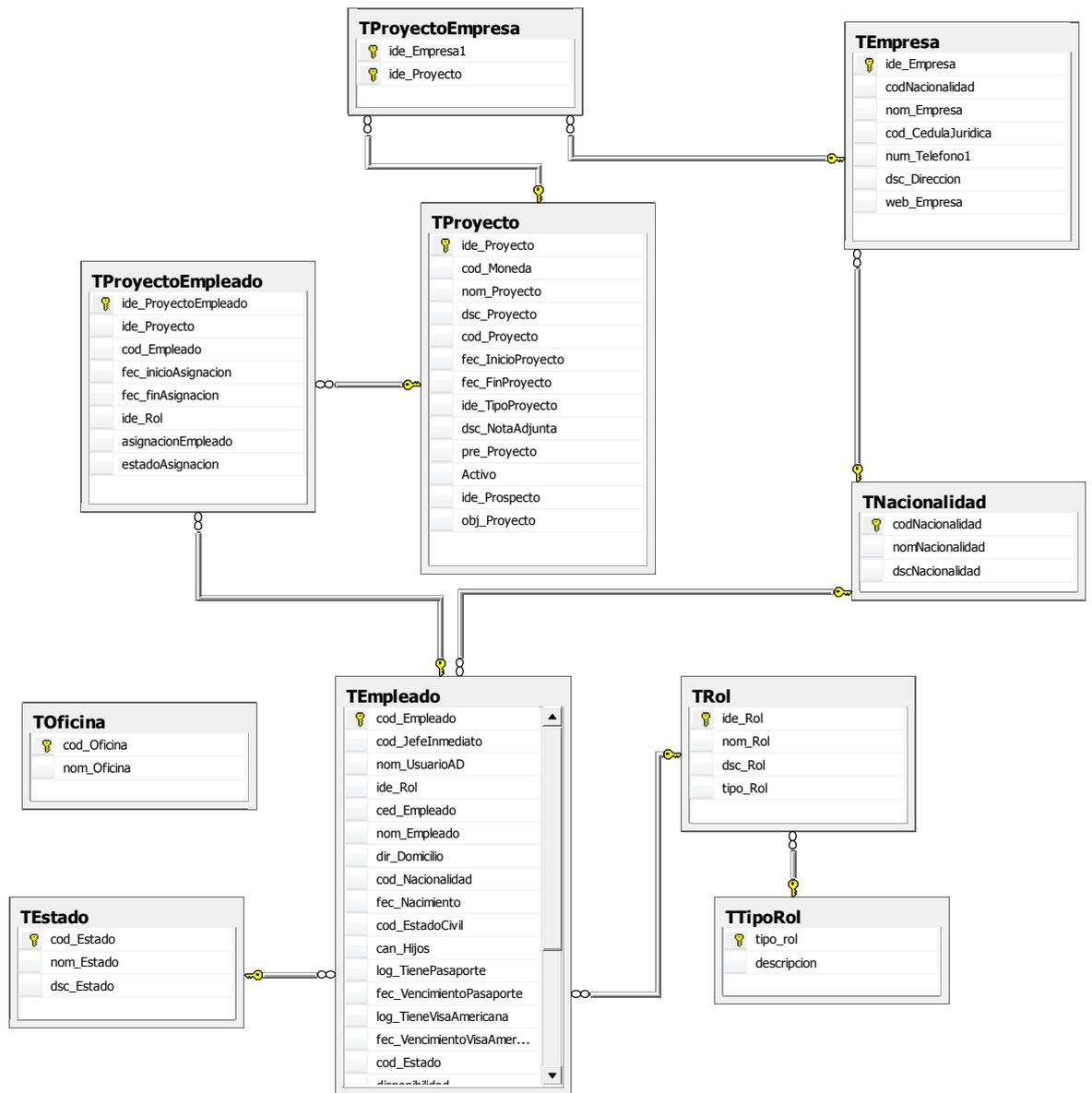


Figura No. 3.2: Diagrama de tablas BabelSys

Las relaciones identificadas fueron las siguientes:

- Las tablas TEmpleado, TRol, TOficina, TNacionalidad y TEstado se relacionan con la perspectiva Empleado
- La tabla TEmpresa se relaciona con la perspectiva Empresa
- Los campos fec_inicioasignacion y fec_fin asignacion de la tabla TProyectoEmpleado se relacionan con la perspectiva Tiempo y nos ayudarán en el conteo de número de días

hábiles del período de labores en el mes correspondiente (cnt num dias hab per mes) y para obtener el conteo de número de días hábiles del mes (cnt num dias hab mes).

- La tabla TNacionalidad se relaciona con la perspectiva Nacionalidad
- La tabla TOficina se relaciona con la perspectiva Oficina
- La tabla TProyecto se relaciona con la perspectiva Proyecto
- Las tablas TTipoRol y TRol se relacionan con la perspectiva Rol
- El campo asignacionempleado en la tabla TProyectoEmpleado se utilizará en el cálculo de porcentaje diario, esta asignación determinada por el usuario en el sistema se ingresa de manera ocupacional en el rango de fechas fec_inicioasignacion y fec_finassignacion de la misma tabla, lo que nos describiría que el empleado E1 en el proyecto 10 se encuentra asignado en un 75% (por ejemplo) en las fechas de 1/8/2011 hasta el 1/9/2011.

c. Nivel de granularidad

De acuerdo a las correspondencias establecidas, se analizaron los campos residentes en cada tabla a la que se hacía referencia, a través de dos métodos diferentes. Primero se examinó la base de datos para intuir los significados de cada campo, y luego se consultó con el encargado del sistema sobre algunos aspectos de los cuales no se comprendía su sentido.

De todas formas, y como puede apreciarse en el diagrama de entidad relación antes expuesto, los nombres de los campos son

bastante explícitos y se deducen con facilidad, pero aún así fue necesario investigarlos para evitar cualquier tipo de inconvenientes.

- Con respecto a la perspectiva Empleado, los datos disponibles son los siguientes:
 - ✓ Id_empleado: Llave Autogenerada, y representa unívocamente a un empleado en particular
 - ✓Codigo: Identificador del empleado.
 - ✓Cedula: Cédula de Identidad del empleado.
 - ✓Nombre: Nombre del empleado.
 - ✓Cod_jefeinmediato: Indicador del jefe inmediato.
 - ✓Cedula_jefe: Cédula del jefe inmediato.
 - ✓Nombre_jefe: Nombre del jefe inmediato.
 - ✓Codigo_rol: Código del rol
 - ✓Nombre_rol: Nombre del rol
 - ✓Codigo_nacionalidad: Código de la nacionalidad (País) del empleado.
 - ✓Nombre_nacionalidad: Nombre de la nacionalidad (País) del empleado.
 - ✓Codigo_oficina: Código de la oficina de labores del empleado.
 - ✓Nombre_oficina: Nombre de la oficina de labores del empleado.
 - ✓Cod_estado: Código del estado del empleado.
 - ✓Nom_estado: Nombre del estado del empleado.
 - ✓Dsc_estado: Descripción del estado del empleado.

- Con respecto a la perspectiva Empresa, los datos disponibles son los siguientes:
 - ✓ id_empresa: Llave Autogenerada.
 - ✓ Código: Identificador de la empresa.

- ✓ Nombre: Nombre de la empresa.
 - ✓ Dirección: Dirección de la empresa.
 - ✓ cod_nacionalidad: Identificar de la nacionalidad (País) de la empresa.
 - ✓ nombre_nacionalidad: Nombre de la nacionalidad (País) de la empresa.
- Con respecto a la perspectiva Nacionalidad, los datos disponibles son los siguientes:
 - ✓ id_nacionalidad : Llave Autogenerada.
 - ✓ Código: Identificador de la nacionalidad.
 - ✓ Nombre: Nombre de la nacionalidad.
- Con respecto a la perspectiva Oficina, los datos disponibles son los siguientes:
 - ✓ id_oficina: Llave Autogenerada.
 - ✓ Código: Identificador de la oficina.
 - ✓ Nombre: Nombre de la oficina.
- Con respecto a la perspectiva Proyecto, los datos disponibles son los siguientes:
 - ✓ id_proyecto: Llave Autogenerada.
 - ✓ Código: Identificador del proyecto.
 - ✓ Nombre: Nombre del proyecto.
 - ✓ Descripción: Descripción del proyecto.
 - ✓ fec_inicio: Fecha en la que inicia el proyecto.
 - ✓ fec_fin: Fecha en la que finaliza el proyecto.
 - ✓ Estado: Estado en el que se encuentra el proyecto (1 = Activo, 0 = Inactivo).

- Con respecto a la perspectiva Rol, los datos disponibles son los siguientes:
 - ✓ id_rol: Llave Autogenerada.
 - ✓ Codigo: Identificador del rol.
 - ✓ Nombre: Nombre del rol.
 - ✓ Descripción: Descripción del rol.
 - ✓ id_tiporol: Identificador del tipo de rol.
 - ✓ descripcion_tiporol: Descripción del tipo de rol.

- Con respecto a la perspectiva Tiempo, los datos disponibles son los siguientes:
 - ✓ id_tiempo: Llave Autogenerada.
 - ✓ Fecha: Identificador único de la fecha.
 - ✓ Anio: Año de la fecha
 - ✓ Semestre: Semestre de la fecha
 - ✓ Trimestre: Trimestre de la fecha
 - ✓ Mes: Mes de la fecha
 - ✓ Semana: Número de semana de la fecha
 - ✓ Dia: Número de día del mes de la fecha
 - ✓ DiaSemana: Número de día de la semana de la fecha
 - ✓ NSemestre: Nombre del semestre de la fecha
 - ✓ NTrimestre: Nombre del trimestre de la fecha
 - ✓ ENMes: Nombre del mes en inglés de la fecha
 - ✓ ESMes: Nombre del mes en español de la fecha
 - ✓ ENMes3L: Nombre del mes en inglés en 3 letras de la fecha
 - ✓ ESMes3L: Nombre del mes en español en 3 letras de la fecha
 - ✓ NSemana: Nombre de la semana de la fecha
 - ✓ ENDia: Nombre del día en inglés del mes de la fecha
 - ✓ ESDia: Nombre del día en español del mes de la fecha

- ✓ ENDiaSemana: Nombre del día en inglés, de la semana de la fecha
- ✓ ESDiaSemana: Nombre del día en español de la semana de la fecha

d. Modelo Conceptual ampliado

Teniendo esto en cuenta, se completará el diseño del diagrama conceptual mostrado en la Figura No. 3.3:

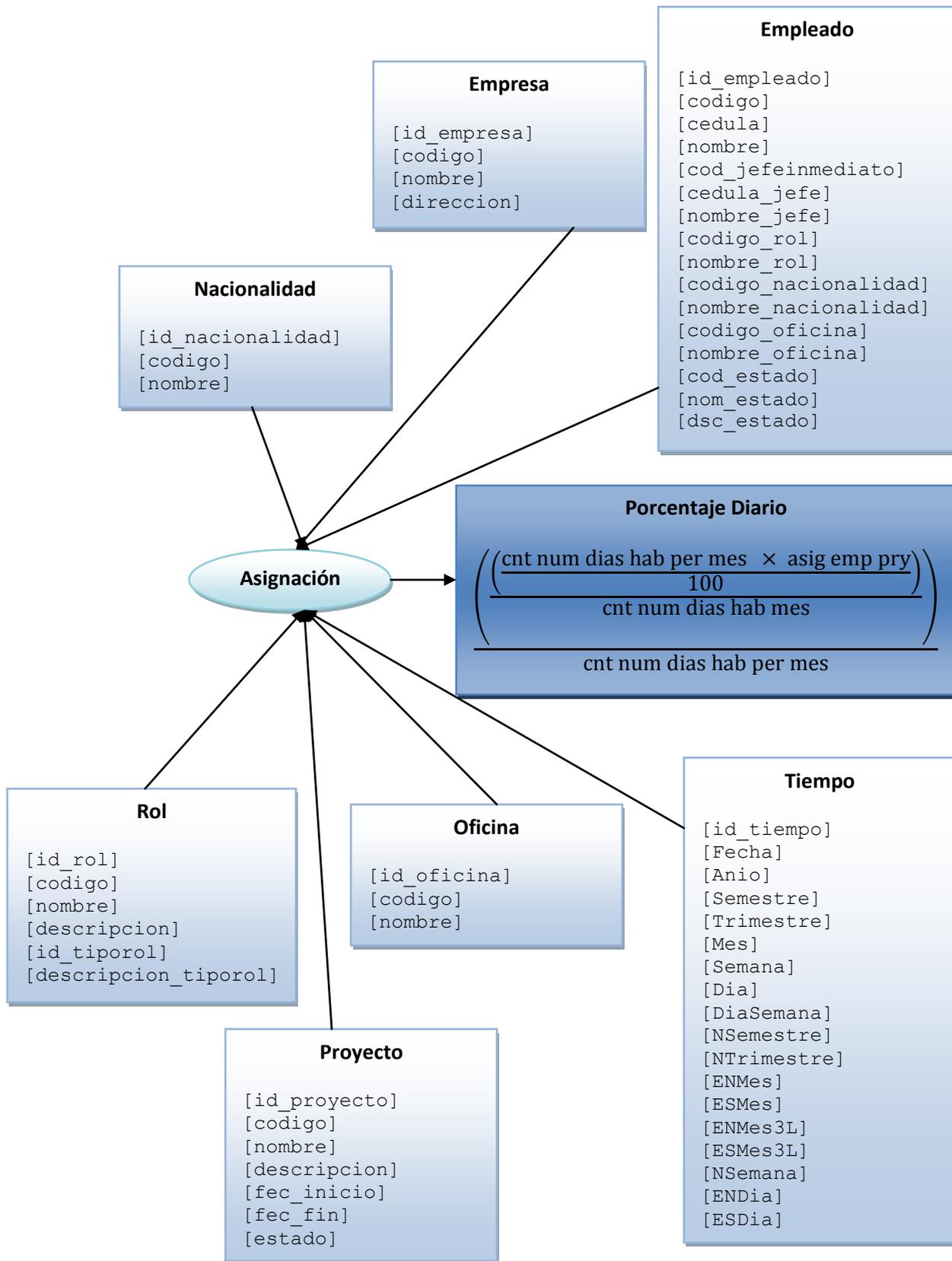


Figura No. 3.3: Modelo Conceptual ampliado

3.1.3. Modelo lógico del Data Warehouse

La estructura de estas tablas no dispondrá de llaves primarias y secundarias en vista que toda relación se realizará de forma lógica utilizando condiciones tales como “JOIN”.

a. Tipo de Modelo Lógico del DW

El esquema que se utilizará será en estrella, debido a sus características, ventajas y diferencias con los otros esquemas.

b. Tablas de dimensiones

A continuación se identifican las tablas que cumplen con alguna funcionalidad específica como dimensión para el comportamiento del conjunto de las soluciones.

Tabla No. 3.1: Descripción Tabla TD_Empleado

Tabla	TD_Empleado
Descripción	<p>La funcionalidad de esta tabla es almacenar la información cualitativa de los empleados.</p> <p>La siguiente dimensión pretende contener toda característica de índole informativa que permita identificar algún empleado en particular, así como también servir de enlace con la tabla de hechos.</p>

La información correspondiente al empleado está compuesta por un conjunto de atributos solicitados por el usuario.

Atributo	Tipo	Descripción
id_empleado	INT	Llave Autogenerada.
codigo	VARCHAR(7)	Identificador Único del empleado.
cedula	VARCHAR(20)	Cédula de Identidad del empleado.
nombre	VARCHAR(200)	Nombre del empleado.
cod_jefeinmediato	VARCHAR(7)	Indicador del jefe inmediato.
cedula_jefe	VARCHAR(20)	Cédula del jefe inmediato.
nombre_jefe	VARCHAR(200)	Nombre del jefe inmediato.
codigo_rol	INT	Código del rol
nombre_rol	VARCHAR(50)	Nombre del rol
codigo_nacionalidad	INT	Código de la nacionalidad (País) del empleado.
nombre_nacionalidad	VARCHAR(50)	Nombre de la nacionalidad (País) del empleado.
codigo_oficina	INT	Código de la oficina de labores

		del empleado.
nombre_oficina	VARCHAR(50)	Nombre de la oficina de labores del empleado.
cod_estado	TINYINT	Código del estado del empleado.
nom_estado	VARCHAR(50)	Nombre del estado del empleado.
dsc_estado	VARCHAR(100)	Descripción del estado del empleado.

Tabla No. 3.2: Descripción Tabla TD_Empresa

Tabla	TD_Empresa
Descripción	<p>La funcionalidad de esta tabla es almacenar la información cualitativa de las empresas clientes.</p> <p>La siguiente dimensión pretende contener toda característica de índole informativa que permita identificar alguna empresa en particular, así como también servir de enlace con la tabla de hechos.</p> <p>La información correspondiente a la empresa está compuesta por un conjunto de atributos solicitados por el usuario.</p>

Atributo	Tipo	Descripción
id_empresa	INT	Llave Autogenerada.
codigo	INT	Identificador Único de la empresa.
nombre	VARCHAR(50)	Nombre de la empresa.
direccion	VARCHAR(250)	Dirección de la empresa.
cod_nacionalidad	INT	Identificar de la nacionalidad (País) de la empresa.
nombre_nacionalidad	VARCHAR(50)	Nombre de la nacionalidad (País) de la empresa.

Tabla No. 3.3: Descripción Tabla TD_Nacionalidad

Tabla	TD_Nacionalidad
Descripción	<p>La funcionalidad de esta tabla es almacenar la información cualitativa de los países o nacionalidades en general.</p> <p>La siguiente dimensión pretende contener toda característica de índole informativa que permita identificar alguna nacionalidad en particular, así como también servir de enlace con la tabla de hechos.</p>

<p>La información correspondiente a la nacionalidad está compuesta por un conjunto de atributos solicitados por el usuario.</p>		
Atributo	Tipo	Descripción
id_nacionalidad	INT	Llave Autogenerada.
Codigo	INT	Identificador Único de la nacionalidad.
Nombre	VARCHAR(50)	Nombre de la nacionalidad.

Tabla No. 3.4: Descripción Tabla TD_Oficina

Tabla	TD_Oficina
Descripción	<p>La funcionalidad de esta tabla es almacenar la información cualitativa de las oficinas pertenecientes a Babel Software.</p> <p>La siguiente dimensión pretende contener toda característica de índole informativa que permita identificar alguna oficina en particular, así como también servir de enlace con la tabla de hechos.</p> <p>La información correspondiente a la oficina está compuesta</p>

	por un conjunto de atributos solicitados por el usuario.	
Atributo	Tipo	Descripción
id_oficina	INT	Llave Autogenerada.
Codigo	INT	Identificador único de la oficina.
Nombre	VARCHAR(50)	Nombre de la oficina.

Tabla No. 3.5: Descripción Tabla TD_Proyecto

Tabla	TD_Proyecto
Descripción	<p>La funcionalidad de esta tabla es almacenar la información cualitativa de los proyectos que tienen cada una de las oficinas.</p> <p>La siguiente dimensión pretende contener toda característica de índole informativa que permita identificar algún proyecto en particular, así como también servir de enlace con la tabla de hechos.</p> <p>La información correspondiente al proyecto está compuesta por un conjunto de atributos solicitados por el usuario.</p>

Atributo	Tipo	Descripción
id_proyecto	INT	Llave Autogenerada.
Codigo	VARCHAR(15)	Identificador único del proyecto.
Nombre	VARCHAR(50)	Nombre del proyecto.
descripcion	VARCHAR(4000)	Descripción del proyecto.
fec_inicio	SMALLDATETIME	Fecha en la que inicia el proyecto.
fec_fin	SMALLDATETIME	Fecha en la que finaliza el proyecto.
estado	BIT	Estado en el que se encuentra el proyecto (1 = Activo, 0 = Inactivo).

Tabla No. 3.6: Descripción Tabla TD_Rol

Tabla	TD_Rol
Descripción	<p>La funcionalidad de esta tabla es almacenar la información cualitativa de los roles que cada empleado tuvo en un proyecto dado, en vista de que aunque ya tiene un rol asignado por defecto, este puede variar dependiendo de la necesidad del proyecto.</p> <p>La siguiente dimensión pretende contener toda característica de índole informativa que permita identificar</p>

	<p>algún rol en particular, así como también servir de enlace con la tabla de hechos.</p> <p>La información correspondiente al rol está compuesta por un conjunto de atributos solicitados por el usuario.</p>	
Atributo	Tipo	Descripción
id_rol	INT	Llave Autogenerada.
Codigo	INT	Identificador único del rol.
Nombre	VARCHAR(50)	Nombre del rol.
descripcion	VARCHAR(250)	Descripción del rol.
id_tiporol	INT	Identificador del tipo de rol.
descripcion_tiporol	VARCHAR(100)	Descripción del tipo de rol.

Tabla No. 3.7: Descripción Tabla TD_Tiempo

Tabla	TD_Tiempo
Descripción	<p>La funcionalidad de esta tabla es almacenar la información de las fechas en las que se realizan los proyectos y que se utilizarán para la asignación diaria.</p> <p>La siguiente dimensión pretende contener toda</p>

característica de índole informativa que permita identificar alguna fecha de realización de proyectos en particular, así como también servir de enlace con la tabla de hechos.

La información correspondiente al tiempo está compuesta por un conjunto de atributos solicitados por el usuario.

Atributo	Tipo	Descripción
id_tiempo	INT	Llave Autogenerada.
Fecha	DATE	Identificador único de la fecha.
Anio	SMALLINT	Año de la fecha
Semestre	SMALLINT	Semestre de la fecha
Trimestre	SMALLINT	Trimestre de la fecha
Mes	SMALLINT	Mes de la fecha
Semana	SMALLINT	Número de semana de la fecha
Dia	SMALLINT	Número de día del mes de la fecha
DiaSemana	SMALLINT	Número de día de la semana de la fecha
NSemestre	CHAR(7)	Nombre del semestre de la fecha
NTrimestre	CHAR(7)	Nombre del trimestre de la

		fecha
ENMes	CHAR(15)	Nombre del mes en inglés de la fecha
ESMes	CHAR(15)	Nombre del mes en español de la fecha
ENMes3L	CHAR(6)	Nombre del mes en inglés en 3 letras de la fecha
ESMes3L	CHAR(6)	Nombre del mes en español en 3 letras de la fecha
NSemana	CHAR(10)	Nombre de la semana de la fecha
ENDia	CHAR(10)	Nombre del día en inglés del mes de la fecha
ESDia	CHAR(10)	Nombre del día en español del mes de la fecha
ENDiaSemana	CHAR(10)	Nombre del día en inglés de la semana de la fecha
ESDiaSemana	CHAR(10)	Nombre del día en español de la semana de la fecha

c. Tablas de hechos

A continuación se identifican las tablas que cumplen con alguna funcionalidad específica como hechos para el comportamiento del conjunto de las soluciones.

Tabla No. 3.8: Descripción Tabla TH_Asignacion

Tabla	TH_Asignacion	
Descripción	<p>La funcionalidad de esta tabla es almacenar la información de las asignaciones diarias de los empleados en los proyectos en los que han trabajado o están trabajando.</p> <p>La creación de esta tabla tiene como objetivo el servir de insumo para la construcción de la solución multidimensional conteniendo atributos que hacen referencia a las diferentes tablas de dimensiones utilizadas en el modelo y las métricas que muestran información cuantitativa acerca de las asignaciones laborales de los empleados.</p>	
Atributo	Tipo	Descripción
id_proyectoempleado	INT	Llave que referencia al proyecto por el empleado.
id_empleado	INT	Llave autogenerada que referencia a la información del empleado.
id_empresa	INT	Llave autogenerada que referencia a la información de la empresa.
id_nacionalidad	INT	Llave autogenerada que referencia a la información de

		la nacionalidad del proyecto.
id_oficina	INT	Llave autogenerada que referencia a la información de la oficina a la que pertenece el empleado.
id_proyecto	INT	Llave autogenerada que referencia a la información del proyecto.
id_rol	INT	Llave autogenerada que referencia a la información del rol que fue asignado al empleado para el proyecto.
id_fec_ini_proyecto	INT	Llave autogenerada que referencia a la información de la fecha de inicio del proyecto.
id_fec_fin_proyecto	INT	Llave autogenerada que referencia a la información de la fecha de finalización del proyecto.
id_tiempo	INT	Llave autogenerada que referencia a al día dentro de las fechas de realización del proyecto sin contar los fines de semana.
AsignacionEmpleado	DECIMAL(18,0)	Porcentaje de asignación designada al empleado en el tiempo que se realice el

		proyecto.
NumDiasLaborablesMes	INT	Número de días laborables en el mes dependiendo de la fecha del registro y del período de desarrollo del proyecto.
EquivalenteDiasLaborables Mes	DECIMAL(18,0)	Número de días laborados totalmente en el mes con el porcentaje de asignación dado.
PorcentajeMensual	DECIMAL(18,10)	Porcentaje de asignación laboral mensual de un empleado en el proyecto.
PorcentajeDiario	DECIMAL(18,10)	Porcentaje de asignación laboral diaria de un empleado en el proyecto.
Fec_insert	DateTime	Fecha y hora de creación del registro
Fec_update	DateTime	Fecha y hora de la actualización del registro

d. Uniones

Se realizarán las uniones o relaciones lógicas pertinentes, de acuerdo a lo que corresponda, las cuales se muestran en la Figura No. 3.4:

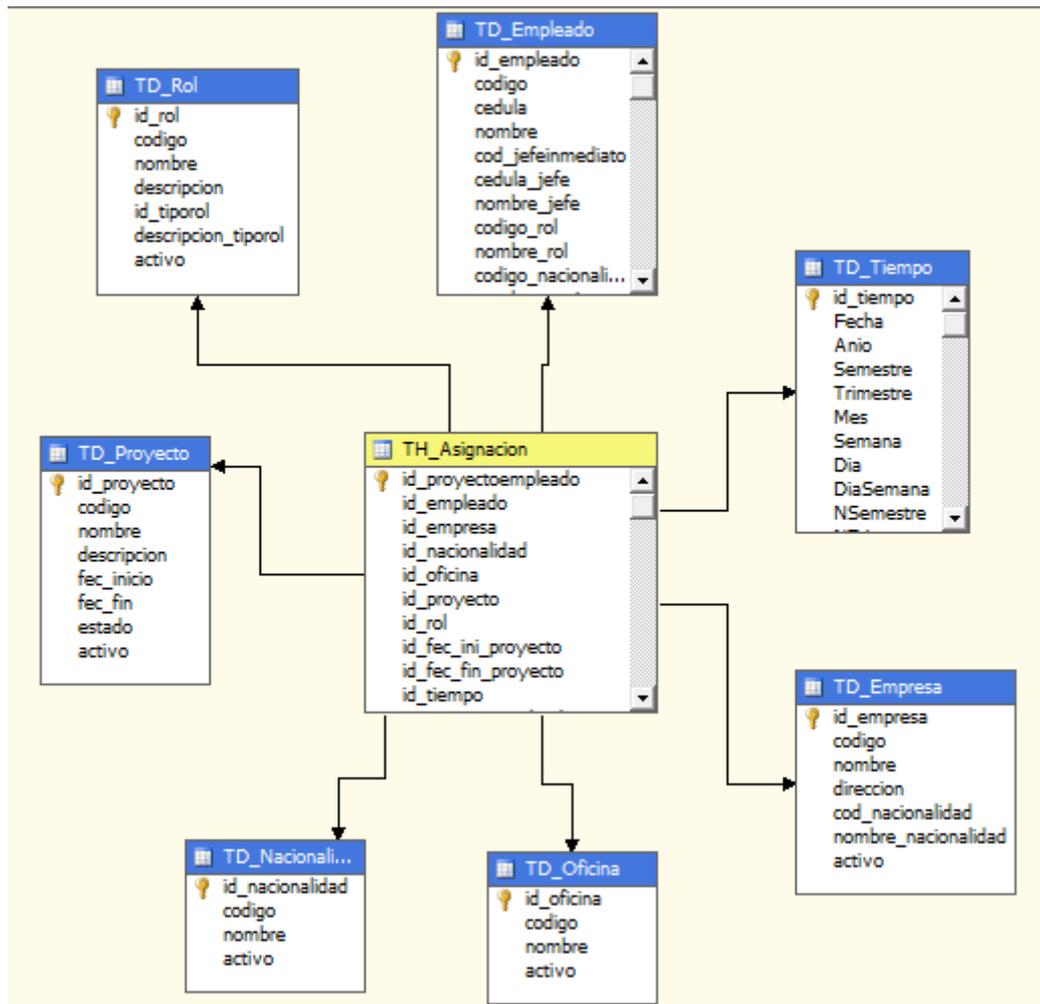


Figura No. 3.4: Relaciones Lógicas

3.1.4. Integración de Datos

a. Carga Inicial

a.1. Arquitectura de flujo de datos

a.1.1. Arquitectura

La arquitectura que se define para el proyecto, unifica diferentes herramientas y las integra, dando como resultado un producto de inteligencia de negocios robusto, que engloba un Data Warehouse complementado con características de Single DDS y reportaría operativa que será integrada por medio de servicios Excel.

A continuación, se muestra la Figura No. 3.5 que unifica la arquitectura del proyecto, y posterior a ella se encuentra la explicación de cada uno de sus componentes.

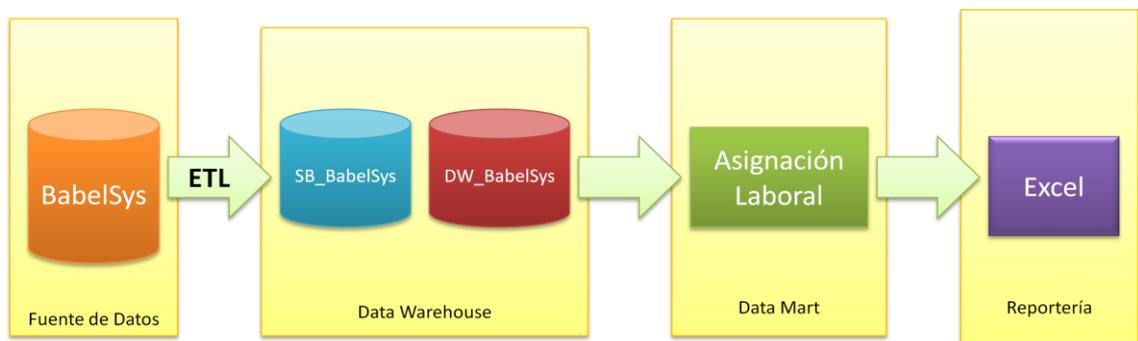


Figura No. 3.5: Arquitectura de flujo de datos

- ✓ **Fuentes de Datos:** Las fuentes de Datos no son otra cosa que las bases de datos de los sistemas transaccionales que utiliza Grupo Babel y que están dentro del alcance del proyecto de control de desempeños, en este caso, la fuente de datos para el proyecto es la base transaccional BabelSys establecida en SQL Server 2008 en el servidor Irazú en las instalaciones de Babel Software en Heredia – Costa Rica
- ✓ **ETL:** Acrónimo de Extraction, Transformation and Load, es decir, procesos de extracción, transformación y carga de datos. Suele ser el elemento fundamental para la carga de datos en el Data

Warehouse. Para el proyecto de control de desempeños, este componente se puede observar dividido en dos:

- *Extracción:* En el caso particular del proyecto de control de desempeños, las extracciones se conocen como Interfaces. Consiste en tomar la información de cada una de las tablas desde el sistema transaccional y moverlas a un área de Staging, que es una base de datos transitoria, entre la fuente de datos y el Data Warehouse.
 - *Transformación y Carga:* El proceso de transformación y carga consiste en la aplicación de todas las reglas de negocio necesarias para la depuración y consolidación de información que se extrajo del sistema transaccional, y que se encuentra en el área de Staging, hacia la base de datos del Data Warehouse.
- ✓ **Área de Staging:** Es un área temporal donde se recogen los datos requeridos del sistema origen. Se extraen los datos estrictamente necesarios para las cargas, y se aplica el mínimo de transformaciones a los mismos. No se aplican restricciones de integridad ni se utilizan claves, los datos se tratan como si las tablas fueran ficheros planos. De esta manera, se minimiza la afectación a los sistemas origen, la carga es lo más rápida posible para minimizar la ventana horaria necesaria, y se reduce también al mínimo la posibilidad de error. Una vez que los datos han sido traspasados, el Data Warehouse se independiza de los sistemas origen hasta la siguiente carga.

Obviamente estos datos no van a dar servicio a ninguna aplicación de reporting; son datos temporales que servirán únicamente de paso al Data Warehouse, de hecho en el esquema lógico de la arquitectura generalmente no aparecen, ya que su función es meramente operativa.

- ✓ **Data Warehouse:** El Data Warehouse contiene datos históricos, y está orientado a la explotación analítica de la información que recoge. En él se almacenan datos que provienen del área de Staging.

El esquema se parece al de un modelo relacional normalizado, pero en él se aplican técnicas de desnormalización. No debe contener un número excesivo de tablas ni de relaciones ya que, por ejemplo, muchas relaciones jerárquicas que en un modelo normalizado se implementarían con tablas separadas aquí deben crearse en una misma tabla, que después representará una dimensión.

Adicionalmente, el Data Warehouse se complementa con estructuras de soporte que forman una especie de DDS, las cuales tienen un corte más operacional, y podrán ser consumidas con una herramienta de reporte.

- ✓ **Data Mart:** Se obtiene a partir de la información recopilada en el área del Data Warehouse. Cada Data Mart es un subconjunto de este almacén, pero orientado a un tema de análisis, para nuestro caso el Data Mart correspondiente es el proceso de asignación de carga laboral con el cual cumpliremos el control de desempeño de los empleados que es objeto de este proyecto.

Los Data Marts se diseñan con estructura multidimensional, cada objeto de análisis consiste en una o varias tablas de hechos enlazadas con diversas tablas de dimensiones. En el caso del proyecto de control de desempeño se diseña siguiendo el Modelo en Estrella, por lo que habrá prácticamente una tabla para cada dimensión.

Un Data Mart está conformado por uno o más modelos multidimensionales, los cuales son estructuras de fácil uso para el usuario y que tienen como idea fundamental que el usuario visualice fácilmente la relación que existe entre los distintos componentes del modelo.

En general, la estructura básica de un Data Warehouse para el Modelo Multidimensional está definida por dos elementos: esquemas y tablas.

- ✓ **Tablas Data Warehouse:** como cualquier base de datos relacional, un Data Warehouse se compone de tablas. Hay dos tipos básicos de tablas en el Modelo Multidimensional:
 - **Tablas de Hechos:** contienen los valores de las medidas de negocios, por ejemplo: porcentaje de asignación diaria, etc.
 - **Tablas de Dimensión:** contienen el detalle de los valores que se encuentran asociados a la tabla de hechos.

- ✓ **Esquemas Data Warehouse:** La colección de tablas en el Data Warehouse se conoce como Esquema. Los esquemas caen

dentro de dos categorías básicas: esquemas estrellas y esquemas snowflake. Aunque para efectos del proyecto solamente se utilizarán esquemas de estrella.

- ✓ **Reportería Analítica:** Una vez que se tiene un almacén de datos, se usan herramientas de reporte y graficación, para construir reportes, gráficas e integrarlas en una sola pantalla de consulta para el usuario. Para la construcción de estos reportes se utilizará Excel Services.

a.1.2. Cargar Stage

El Stage es una base de datos, que es común, es mejor no poner todos los índices o las restricciones (por ejemplo, no es nulo, las claves principales o restricciones de comprobación) en la base de datos de la etapa. La razón principal de esto no se debe a la ejecución, sino porque queremos capturar y reportar los "malos datos" en el proceso de calidad de los datos. Queremos permitir que los datos malos como nulo y duplicados de las llaves primarias en la tabla de la etapa para que podamos informar de ello. Si restringimos nuestra mesa de escenario para rechazar nulos y duplicados de las llaves principales, el proceso de calidad de los datos que se encuentra entre el escenario y la NDS no sería capaz de captar estos temas y reportarlos para su corrección en el sistema de origen. Por ejemplo, si se restringe la tabla de la etapa con una única restricción en la columna de clave principal, cuando tenemos dos filas duplicadas, la segunda fila no será insertada en el escenario. Es mejor que les permita en el escenario. de esta manera, cuando nos pueblan la NDS, podemos poner en práctica un conjunto de reglas completo y riguroso que filtrar los datos

erróneos en el área de calidad de los datos de cuarentena para la presentación de informes y el procesamiento, tales como limpieza, fijación, reimportación o archivado.

Sin embargo no usaremos un NDS, en vista de que la fuente de dato es una sola base de datos transaccional OLTP, entonces no concurremos con problemas de datos repetidos, por lo que pasaremos del Stage a la DDS directamente. Usaremos el Stage como una base de datos temporal para liberar de procesos a la fuente de datos (BabelSys).

Usando SQL Server Integration Services (SSIS) crearemos los ETLs necesarios de esta primera etapa para los flujos de datos, los cuales alimentarán al Stage (SB_BabelSys) desde la fuente de datos.

Dentro de la solución ISP_BabelSys se encuentra el proyecto *ISP_SB_BabelSys* el cuál contiene los paquetes necesarios para realizar la migración de la fuente de datos a la base Stage,

✓ **Usando SSIS para poblar el Stage *SB_BabelSys***

I. Paquete “*pkgSGTipoRol*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TTipoRol] desde la fuente de datos [BabelSys] hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal para desde ahí realizar las transformaciones de datos, necesarias y

fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.9: pkgSGTipoRol - Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgSGTipoRol.txt	C:\log\pkgSGTipoRol.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.10: pkgSGTipoRol – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por una única tarea de flujo de datos denominada “*Data Flow Task*” como se muestra en la Figura No. 3.6

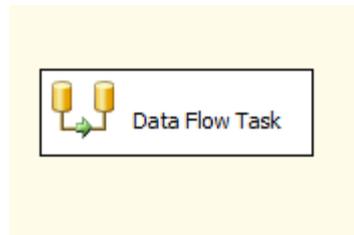


Figura No. 3.6: pkgSGTipoRol – Control Flow

e. DATA FLOW

El control Data Flow Task tiene el siguiente flujo de datos mostrado en la Figura No. 3.7:

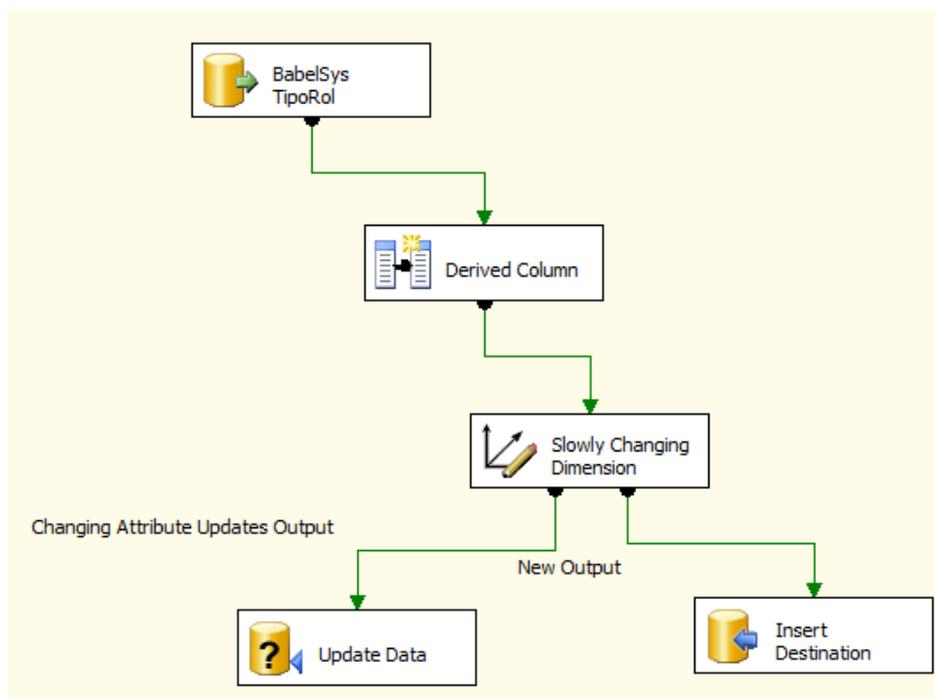


Figura No. 3.7: pkgSGTipoRol – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.11: pkgSGTipoRol – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
BabelSys TipoRol		
OLE DB Source		
	tipo_rol	tipo_rol
	descripcion	descripcion
Derived Column		
Derived Column		
	activo	"1"
	id_tiporol	tipo_rol
Insert Destination		
OLE DB Destination		
	id_tiporol	id_tiporol
	descripcion	descripcion
	activo	activo
Slowly Changing Dimension		
Slowly Changing Dimension		
	id_tiporol	id_tiporol
	descripcion	descripcion
	activo	activo
Update Data		

OLE DB Command		
	id_tiporol	id_tiporol
	descripcion	descripcion
	activo	"1"

II. Paquete “*pkgSGRol*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TRol] desde la fuente de datos [BabelSys] hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal para desde ahí realizar las transformaciones de datos, necesarias y fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.12: pkgSGRol – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.BabelSys.sa	Server Oficinas	Se conecta a la

	Babel Software Latacunga	fFuente de datos.
pkgSGRol.txt	C:\log\pkgSGRol.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.13: pkgSGRol – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por una única tarea de flujo de datos denominada “*Data Flow Task*” como se muestra en la Figura No. 3.8:

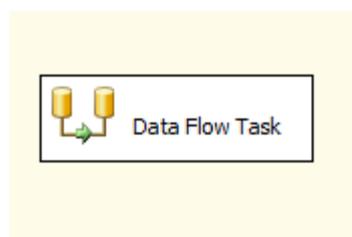


Figura No. 3.8: pkgSGRol – Control Flow

e. DATA FLOW

El control Data Flow Task tiene el siguiente flujo de datos, como se presenta en la Figura No. 3.9:

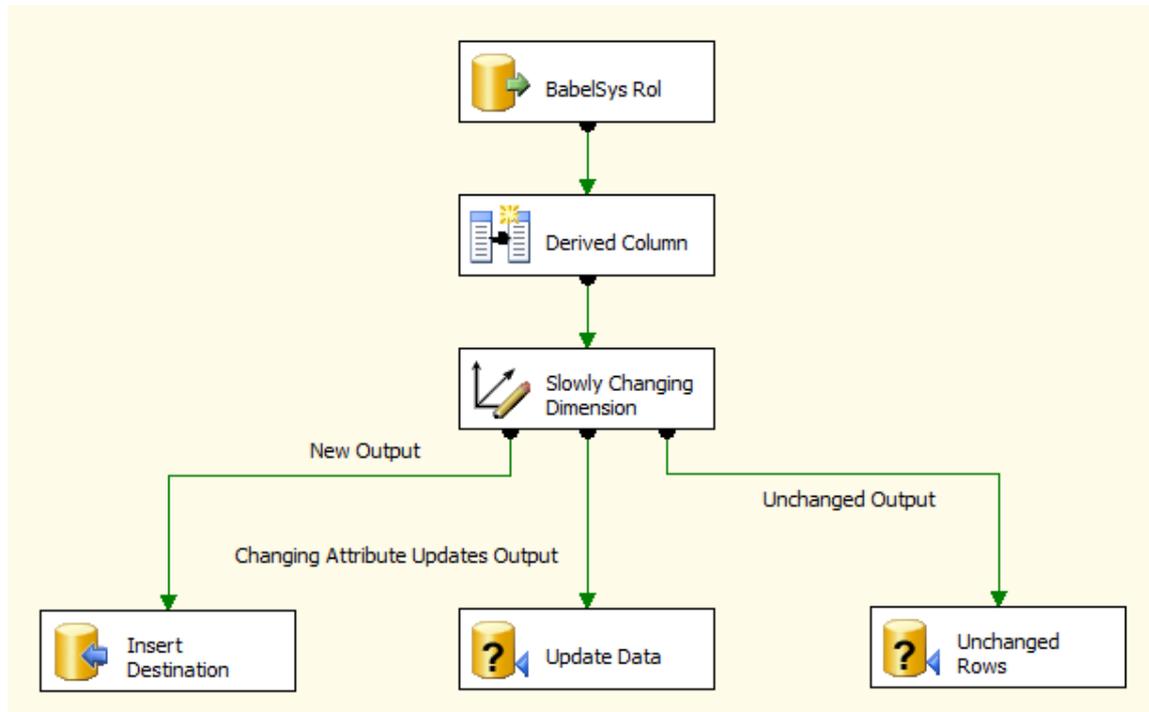


Figura No. 3.9: pkgSGRol – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.14: pkgSGRol – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
BabelSys Rol		

OLE DB Source		
	ide_rol	ide_rol
	nom_rol	nom_rol
	dsc_rol	dsc_rol
	tipo_rol	tipo_rol
Derived Column Derived Column		
	activo	"1"
	id_tiporol	tipo_rol
Insert Destination OLE DB Destination		
	ide_rol	ide_rol
	id_tiporol	id_tiporol
	nom_rol	nom_rol
	dsc_rol	dsc_rol
	activo	Active
Slowly Changing Dimension Slowly Changing Dimension		
	activo	activo
	dsc_rol	dsc_rol
	id_tiporol	id_tiporol
	ide_rol	ide_rol
	nom_rol	nom_rol
Unchanged Rows OLE DB Command		
	ide_rol	ide_rol
	activo	0
Update Data		

OLE DB Command	
dsc_rol	dsc_rol
nom_rol	nom_rol
id_tiporol	id_tiporol
ide_rol	ide_rol
activo	"1"

III. Paquete “*pkgSGProyectoEmpresa*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TProyectoEmpresa] desde la fuente de datos BabelSys hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal para desde ahí realizar las transformaciones de datos, necesarias y fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.15: pkgSGProyectoEmpresa – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
--------	-----------	---------------

192.168.30.252.SB_Babel Sys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.BabelSys. sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgSGProyectoEmpresa. txt	C:\log\pkgSGProyectoEmp resa.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.16: pkgSGProyectoEmpresa – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por:

- Execute SQL Task
- Data Flow Task

Como se muestra en la Figura No. 3.10:

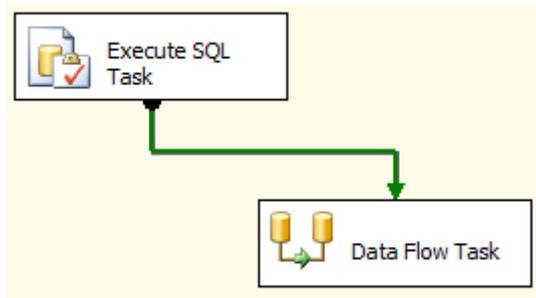


Figura No. 3.10: pkgSGProyectoEmpresa – Control Flow

e. DATA FLOW

El control Data Flow Task tiene en la Figura No. 3.11:

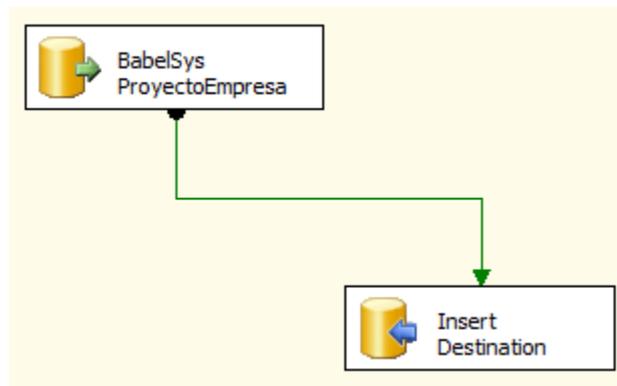


Figura No. 3.11: pkgSGProyectoEmpresa – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.17: pkgSGProyectoEmpresa – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
BabelSys		
ProyectoEmpresa		
OLE DB Source		
	ide_Empresa1	ide_Empresa1
	ide_proyecto	ide_proyecto
Insert Destination		
OLE DB Destination		
	ide_Empresa	ide_Empresa1
	ide_proyecto	ide_proyecto

IV. Paquete “*pkgSGProyectoEmpleado*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TProyectoEmpleado] desde la fuente de datos BabelSys hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal para desde ahí realizar las transformaciones de datos, necesarias y fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.18: pkgSGProyectoEmpleado – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_Babel Sys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.BabelSys .sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgSGProyectoEmplead o.txt	C:\log\pkgSGProyectoEmpl eado.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.19: pkgSGProyectoEmpleado – Variables

NOMBRE	ALCANCE	TIPO
--------	---------	------

d. CONTROL FLOW

El control de flujo se encuentra constituido por una única tarea de flujo de datos denominada “*Data Flow Task*” como se muestra en la Figura No. 3.12:

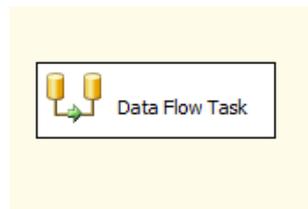


Figura No. 3.12: pkgSGProyectoEmpleado – Control Flow

e. DATA FLOW

El control Data Flow Task tiene el siguiente flujo de datos:

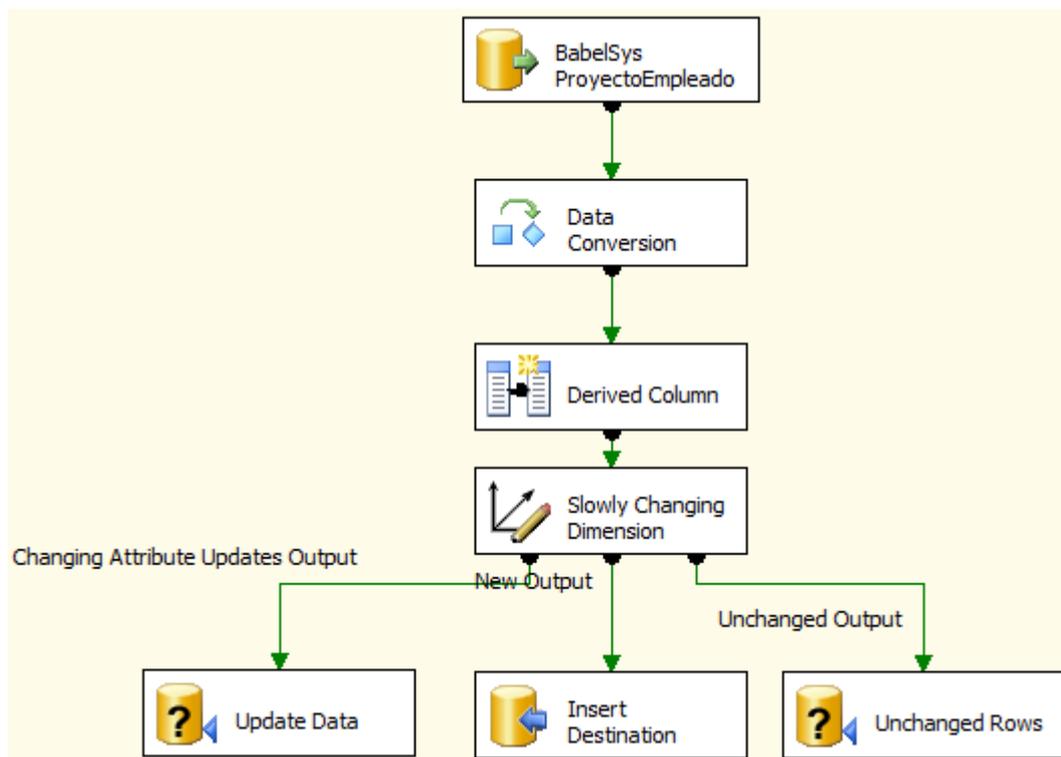


Figura No. 3.13: pkgSGProyectoEmpleado – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.20: pkgSGProyectoEmpleado – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
BabelSys		
ProyectoEmpleado		
OLE DB Source		
	ide_ProyectoEmpleado	ide_ProyectoEmpleado
	ide_proyecto	ide_proyecto
	cod_empleado	cod_empleado
	fec_inicioAsignacion	fec_inicioAsignacion
	fec_finAsignacion	fec_finAsignacion
	ide_rol	ide_rol
	asignacionEmpleado	asignacionEmpleado
	estadoAsignacion	estadoAsignacion
Data Conversion		
Data Conversion		
	fec_inicioAsignacion	cv_fec_inicioAsignacion
	fec_finAsignacion	cv_fec_finAsignacion
	asignacionEmpleado	cv_asignacionEmpleado
Derived Column		
Derived Column		
	activo	"1"
Insert Destination		
OLE DB Destination		

	activo	activo
	asignacionEmpleado	cv_asignacionEmpleado
	cod_empleado	cod_empleado
	estadoAsignacion	estadoAsignacion
	fec_ini_asignacion	cv_fec_inicioAsignacion
	fec_fin_asignacion	cv_fec_finAsignacion
	ide_proyecto	ide_proyecto
	ide_proyectoEmpleado	ide_proyectoEmpleado
	ide_rol	ide_rol
Slowly Changing Dimension		
Slowly Changing Dimension		
	activo	activo
	cv_asignacionEmpleado	cv_asignacionEmpleado
	cod_empleado	cod_empleado
	estadoAsignacion	estadoAsignacion
	cv_fec_inicioAsignacion	cv_fec_inicioAsignacion
	cv_fec_finAsignacion	cv_fec_finAsignacion
	ide_proyecto	ide_proyecto
	ide_proyectoEmpleado	ide_proyectoEmpleado
	ide_rol	ide_rol
Unchanged Rows		
OLE DB Command		
	ide_proyectoempleado	ide_proyectoempleado
	activo	0
Update Data		
OLE DB Command		
	asignacionEmpleado	cv_asignacionEmpleado
	cod_empleado	cod_empleado
	estadoAsignacion	estadoAsignacion

fec_ini_asignacion	cv_fec_inicioAsignacion
fec_fin_asignacion	cv_fec_finAsignacion
ide_proyecto	ide_proyecto
ide_proyectoEmpleado	ide_proyectoEmpleado
ide_rol	ide_rol
activo	"1"

V. Paquete “*pkgSGProyecto*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TProyecto] desde la fuente de datos BabelSys hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal para desde ahí realizar las transformaciones de datos, necesarias y fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.21: pkgSGProyecto – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys.s.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.

192.168.30.252.BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgSGProyecto.txt	C:\log\pkgSGProyecto.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.22: pkgSGProyecto – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por una única tarea de flujo de datos denominada “*Data Flow Task*” como se presenta en la Figura No. 3.14:

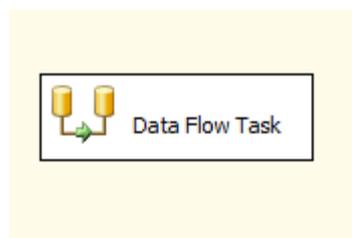


Figura No. 3.14: pkgSGProyecto – Control Flow

e. DATA FLOW

El control Data Flow Task tiene el siguiente flujo de datos mostrado en la Figura No. 3.15:

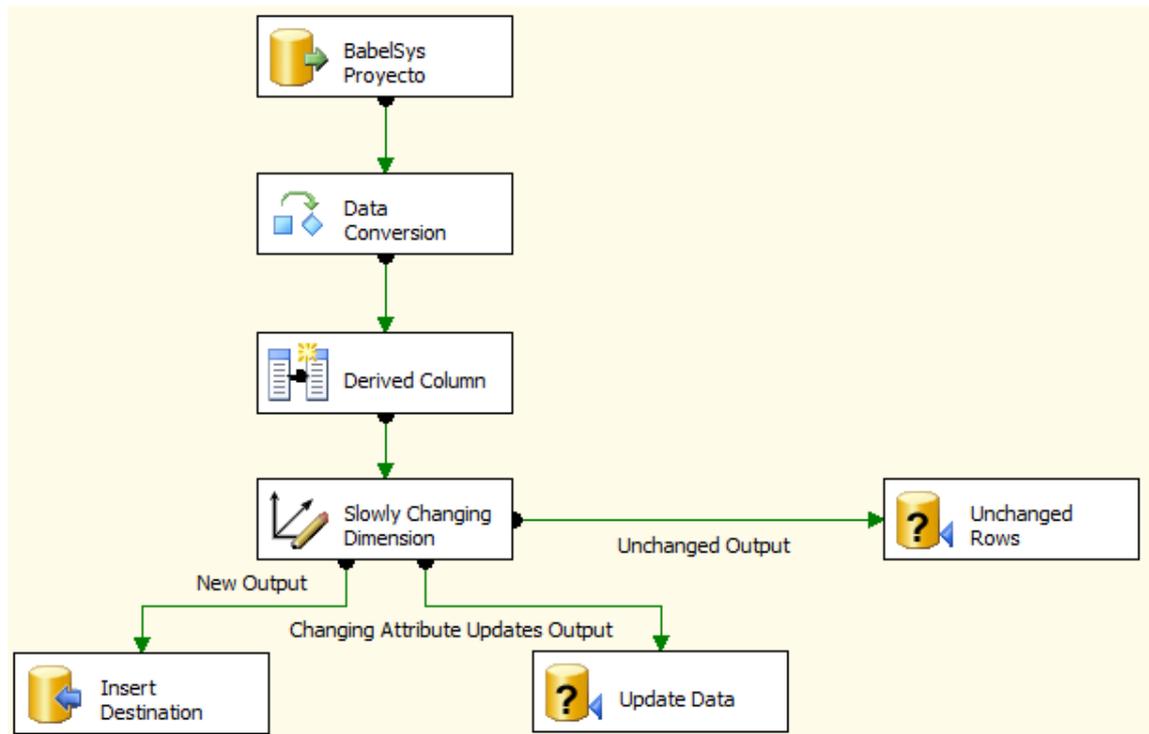


Figura No. 3.15: pkgSGProyecto – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.23: pkgSGProyecto – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
BabelSys Proyecto		

OLE DB Source		
	ide_Proyecto	ide_Proyecto
	nom_proyecto	nom_proyecto
	dsc_proyecto	dsc_proyecto
	fec_InicioProyecto	fec_InicioProyecto
	fec_FinProyecto	fec_FinProyecto
	cod_proyecto	cod_proyecto
	activo	activo
Data Conversion		
Data Conversion		
	cv_ide_proyecto	ide_proyecto
	cv_dsc_proyecto	dsc_proyecto
Derived Column		
Derived Column		
	activoDW	"TRUE"
Insert Destination		
OLE DB Destination		
	ide_proyecto	cv_ide_proyecto
	cod_proyecto	cod_proyecto
	nom_proyecto	nom_proyecto
	dsc_proyecto	cv_dsc_proyecto
	fec_ini_pro	fec_InicioProyecto
	fec_fin_pro	fec_FinProyecto
	estado	activo
	activo	activoDW
Slowly Changing Dimension		
Slowly Changing Dimension		
	activoDW	activoDW

	cod_proyecto	cod_proyecto
	cv_dsc_proyecto	cv_dsc_proyecto
	activo	Active
	fec_InicioProyecto	fec_InicioProyecto
	fec_FinProyecto	fec_FinProyecto
	cv_ide_proyecto	cv_ide_proyecto
	nom_proyecto	nom_proyecto
Unchanged Rows		
OLE DB Command		
	ide_proyecto	cv_ide_proyecto
	activo	0
Update Data		
OLE DB Command		
	cod_proyecto	cod_proyecto
	dsc_proyecto	dsc_proyecto
	fec_fin_pro	fec_FinProyecto
	fec_InicioProyecto	fec_InicioProyecto
	nom_proyecto	nom_proyecto
	ide_proyecto	cv_ide_proyecto
	activo	1

VI. Paquete “*pkgSGOficina*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TOficina] desde la fuente de datos [BabelSys] hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal

para desde ahí realizar las transformaciones de datos, necesarias y fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.24: pkgSGOficina – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgSGOficina.txt	C:\log\pkgSGOficina.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.25: pkgSGOficina – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por una única tarea de flujo de datos denominada “*Data Flow Task*” como se muestra en la Figura No. 3.16:

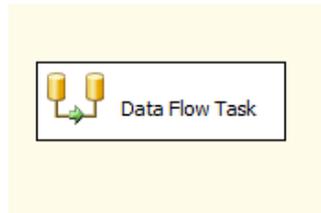


Figura No. 3.16: pkgSGOficina – Control Flow

e. DATA FLOW

El control Data Flow Task tiene el siguiente flujo de datos:

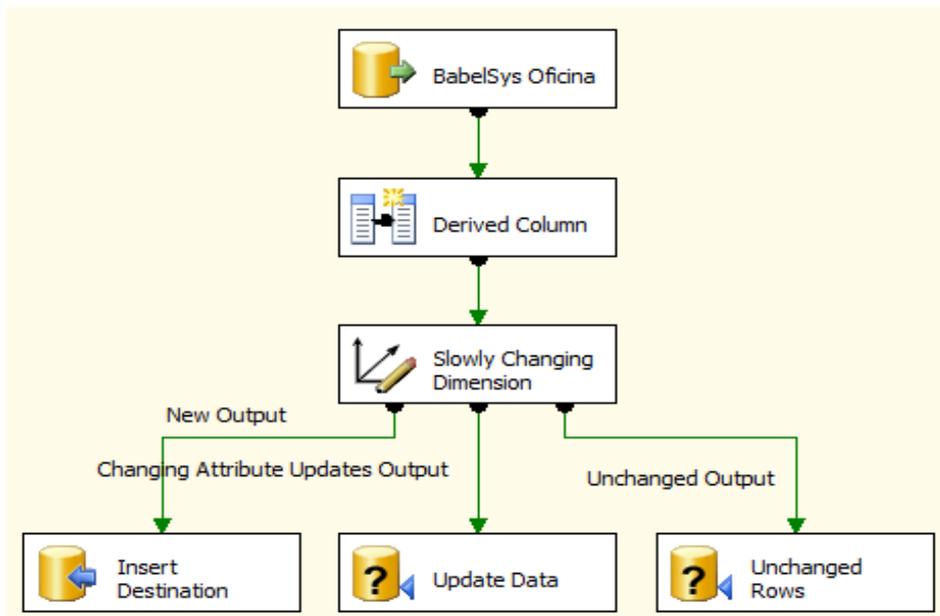


Figura No. 3.17: pkgSGOficina – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.26: pkgSGOficina – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
BabelSys Oficina OLE DB Source	cod_oficina nom_oficina	cod_oficina nom_oficina
Derived Column Derived Column	activo codOficina	"1" cod_oficina
Insert Destination OLE DB Destination	cod_oficina nom_oficina activo	codOficina nom_oficina activo
Slowly Changing Dimension Slowly Changing Dimension	activo codOficina nom_oficina	activo codOficina nom_oficina
Unchanged Rows		

OLE DB Command		
	cod_oficina	cod_oficina
	activo	0
Update Data		
OLE DB Command		
	nom_oficina	nom_oficina
	cod_oficina	cod_oficina
	Activo	1

VII. Paquete “*pkgSGNacionalidad*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TNacionalidad] desde la fuente de datos [BabelSys] hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal para desde ahí realizar las transformaciones de datos, necesarias y fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.27: pkgSGNacionalidad – Data Sources

NOMBRE	UBICACIÓN	OBSERVACION
--------	-----------	-------------

		ES
192.168.30.252.SB_BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgSGNacionalidad.txt	C:\log\pkgSGOficina.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.28: pkgSGNacionalidad – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por una única tarea de flujo de datos denominada “*Data Flow Task*” como se muestra en la Figura No. 3.18:

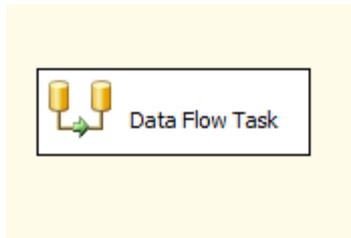


Figura No. 3.18: pkgSGNacionalidad – Control Flow

e. DATA FLOW

El control Data Flow *Task* tiene el siguiente flujo de datos que se muestra en la Figura No. 3.19:

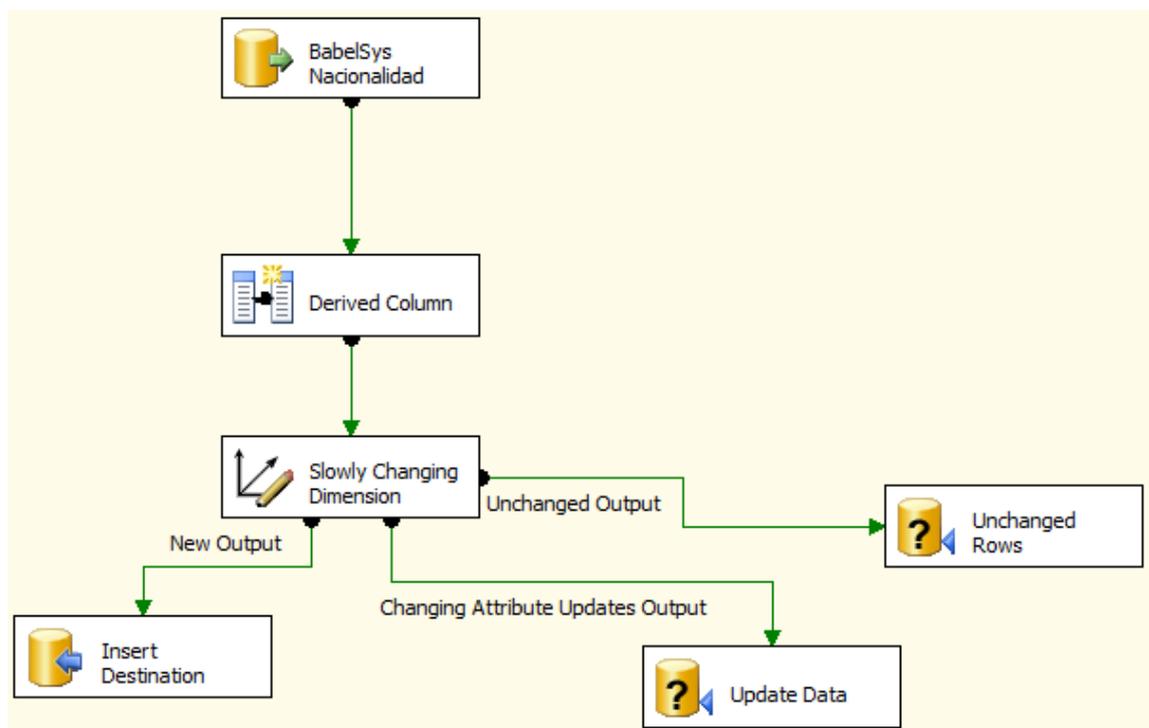


Figura No. 3.19: pkgSGNacionalidad – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.29: pkgSGNacionalidad – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
BabelSys		
Nacionalidad		
OLE DB Source		
	codNacionalidad	codNacionalidad
	nomNacionalidad	nomNacionalidad
Derived Column		
Derived Column		
	activo	"1"
	cod_Nacionalidad	codNacionalidad
Insert Destination		
OLE DB Destination		
	codNacionalidad	cod_nacionalidad
	nomNacionalidad	nomNacionalidad
	activo	activo
Slowly Changing Dimension		
Slowly Changing Dimension		
	activo	activo
	cod_nacionalidad	cod_nacionalidad
	nomNacionalidad	nomNacionalidad
Unchanged Rows		

OLE DB Command		
	codNacionalidad	cod_nacionalidad
	activo	0
Update Data		
OLE DB Command		
	nomNacionalidad	nomNacionalidad
	codNacionalidad	cod_nacionalidad
	activo	1

VIII. Paquete “*pkgSGEstado*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TEstado] desde la fuente de datos BabelSys hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal para desde ahí realizar las transformaciones de datos, necesarias y fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.30: pkgEstado – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgSGEstado.txt	C:\log\pkgSGEstado.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.31: pkgEstado – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por una única tarea de flujo de datos denominada “*Data Flow Task*” como se muestra en la Figura No. 3.20:

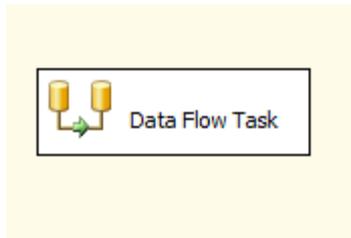


Figura No. 3.20: pkgEstado – Control Flow

e. DATA FLOW

El control Data Flow Task tiene el siguiente flujo de datos que se presenta en la Figura No. 3.20:

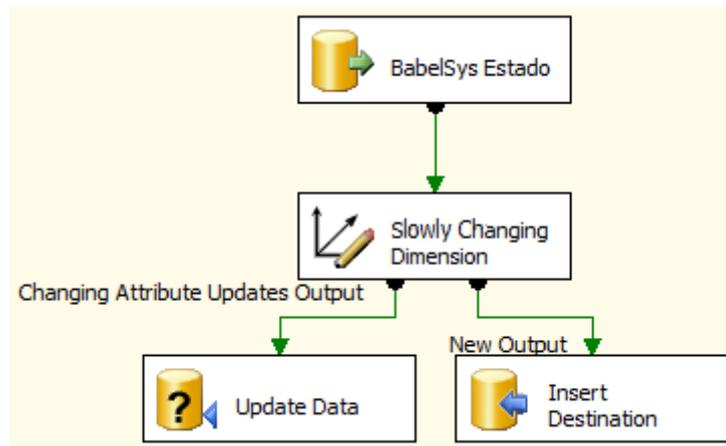


Figura No. 3.21: pkgEstado – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.32: pkgEstado – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
BabelSys Estado OLE DB Source		
	cod_estado	cod_estado
	nom_estado	nom_estado
	dsc_estado	dsc_estado
Insert Destination OLE DB Destination		
	activo	activo
	cod_empleado	cod_empleado
	cod_jefeinmediato	cod_jefeinmediato
	ide_rol	ide_rol
	ced_empleado	ced_empleado
	nom_empleado	nom_empleado
	codNacionalidad	codNacionalidad
	cod_oficina	codOficina
	cod_estado	cod_estado
Slowly Changing Dimension Slowly Changing Dimension		
	cod_estado	cod_estado
	nom_estado	nom_estado
	dsc_estado	dsc_estado
Update Data OLE DB Command		
	cod_estado	cod_estado
	nom_estado	nom_estado

dsc_estado	dsc_estado
activo	1

IX. Paquete “*pkgSGEmpresa*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TEmpresa] desde la fuente de datos [BabelSys] hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal para desde ahí realizar las transformaciones de datos, necesarias y fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.33: pkgSGEmpresa – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys.s.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgSGEmpresa.txt	C:\log\pkgSGEmpresa .txt	Genera un archivo plano

	con datos de errores si es que se produjeran.
--	---

c. VARIABLES

Tabla No. 3.34: pkgSGEmpresa – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por una única tarea de flujo de datos denominada “*Data Flow Task*” como se muestra en el Figura No. 3.22:

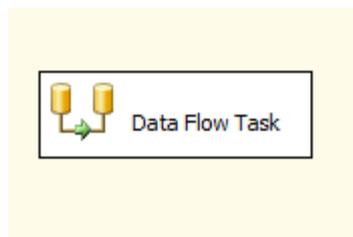


Figura No. 3.22: pkgSGEmpresa – Control Flow

e. DATA FLOW

El control Data Flow Task tiene el siguiente flujo de datos presentado en la Figura No. 3.23:

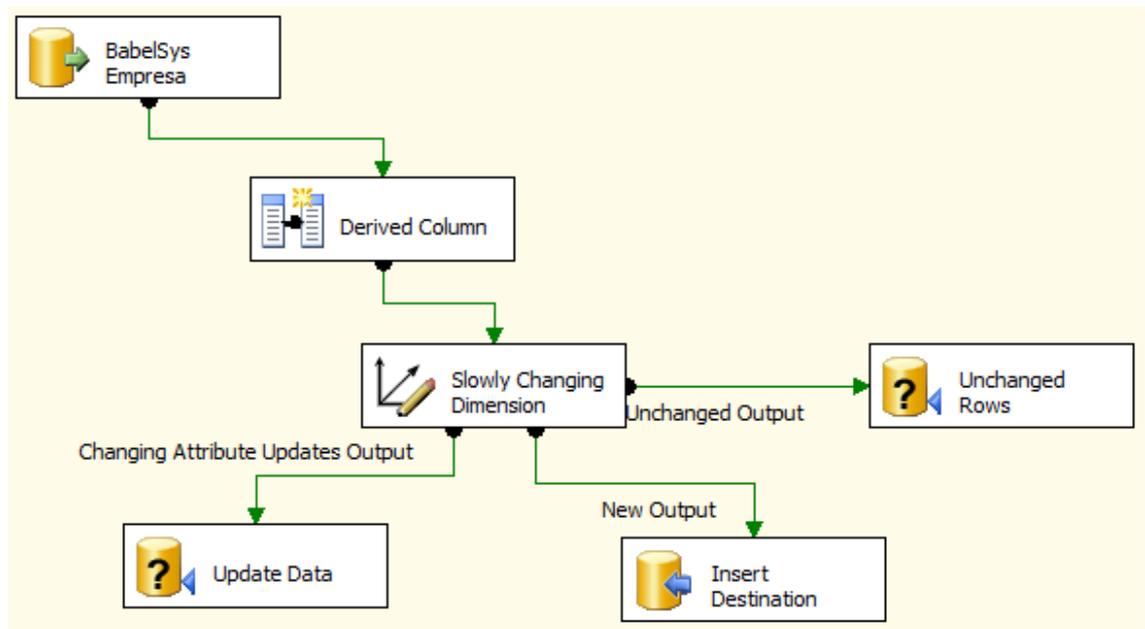


Figura No. 3.23: pkgSGEmpresa – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.35: pkgSGEmpresa – Mapeado

COMPONENTE	COLUMNA SALIENTE	COLUMNA ENTRANTE
-------------------	-------------------------	-------------------------

	(Source)	(Objetivo)
BabelSys Empresa		
OLE DB Source		
	ide_Empresa	ide_Empresa
	codNacionalidad	codNacionalidad
	nom_empresa	nom_empresa
	dsc_direccion	dsc_direccion
Derived Column		
Derived Column		
	activo	"1"
	cod_nacionalidad	codNacionalidad
Insert Destination		
OLE DB Destination		
	activo	activo
	codNacionalidad	cod_nacionalidad
	nom_empresa	nom_empresa
	dsc_direccion	dsc_direccion
	ide_Empresa	ide_Empresa
Slowly Changing Dimension		
Slowly Changing Dimension		
	activo	activo
	cod_nacionalidad	cod_nacionalidad
	nom_empresa	nom_empresa
	dsc_direccion	dsc_direccion
	ide_Empresa	ide_Empresa
Unchanged Rows		
OLE DB Command		
	ide_empresa	ide_empresa

	activo	0
Update Data		
OLE DB Command		
	codNacionalidad	cod_nacionalidad
	dsc_direccion	dsc_direccion
	nom_empresa	nom_empresa
	ide_empresa	ide_empresa
	activo	1

X. Paquete “*pkgSGEmpleado*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de la tabla [TEmpleado] desde la fuente de datos BabelSys hasta el Stage a la tabla con el mismo nombre. Los datos migrados serán íntegros tal cual se encuentran en la base de BabelSys, esto lo realizamos con el propósito de tener los datos migrados en nuestra base temporal para desde ahí realizar las transformaciones de datos, necesarias y fuertes. Puede darse el caso de que se necesiten transformaciones de menor escala mientras se realiza el proceso de migración.

b. DATA SOURCES

Tabla No. 3.36: pkgSGEmpleado – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
--------	-----------	---------------

192.168.30.252.SB_BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgSGEmpleado.txt	C:\log\pkgSGEmpleado.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.37: pkgSGEmpleado – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por una única tarea de flujo de datos denominada “*Data Flow Task*” presentado en la Figura No. 3.24:

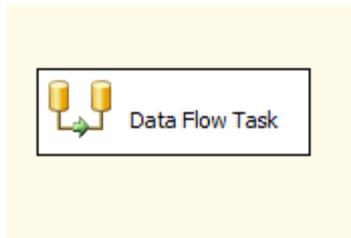


Figura No. 3.24: pkgSGEmpleado – Control Flow

e. DATA FLOW

El control Data Flow Task tiene el siguiente flujo de datos, que se muestra en la Figura No. 3.25:

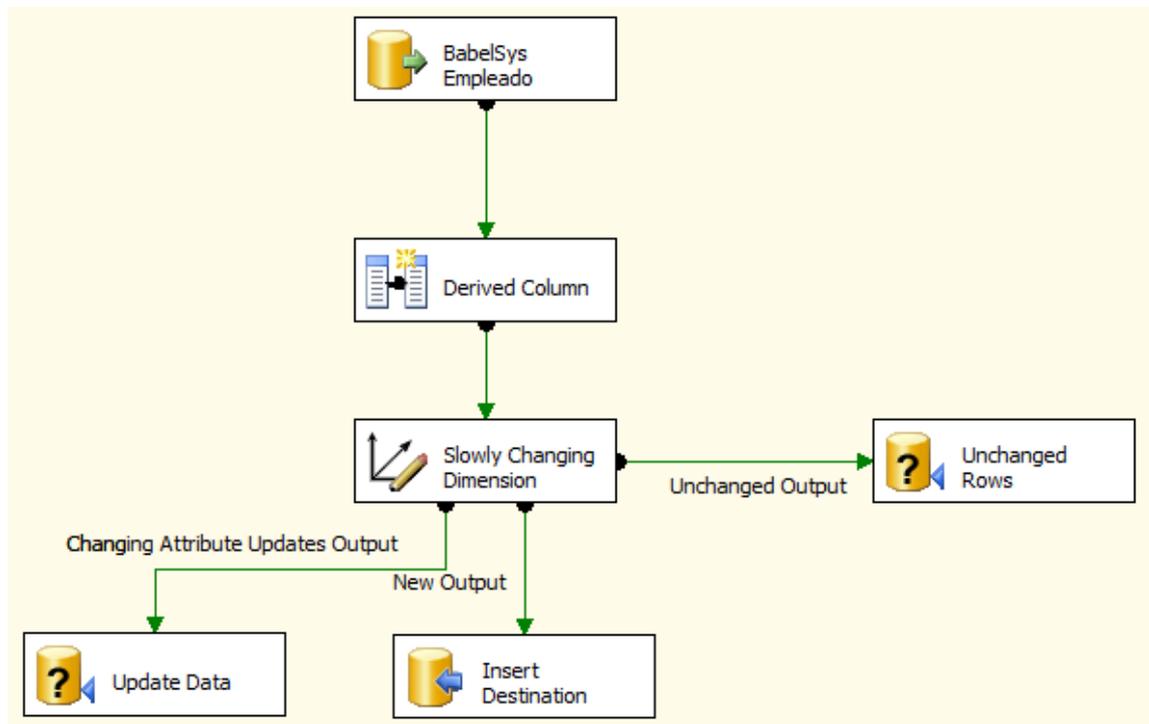


Figura No. 3.25: pkgSGEmpleado – Data Flow (Data Flow Task)

f. MAPEADO

Tabla No. 3.38: pkgSGEmpleado – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
BabelSys Empresa OLE DB Source		
	cod_empleado	cod_empleado
	cod_jefeInmediato	cod_jefeInmediato
	ide_rol	ide_rol
	ced_empleado	ced_empleado
	nom_empleado	nom_empleado
	cod_nacionalidad	cod_nacionalidad
	cod_oficina	cod_oficina
	cod_estado	cod_estado
Derived Column Derived Column		
	activo	"1"
	codOficina	cod_oficina
	codNacionalidad	cod_nacionalidad
Insert Destination OLE DB Destination		
	activo	activo
	cod_empleado	cod_empleado
	cod_jefeInmediato	cod_jefeInmediato
	ide_rol	ide_rol
	ced_empleado	ced_empleado
	nom_empleado	nom_empleado
	codNacionalidad	codNacionalidad

	cod_oficina	codOficina
	cod_estado	cod_estado
Slowly Changing Dimension		
Slowly Changing Dimension		
	activo	activo
	ced_empleado	ced_empleado
	cod_empleado	cod_empleado
	cod_estado	cod_estado
	cod_jefeinmediato	cod_jefeinmediato
	codOficina	codOficina
	codNacionalidad	codNacionalidad
	ide_rol	ide_rol
	nom_empleado	nom_empleado
Unchanged Rows		
OLE DB Command		
	cod_empleado	cod_empleado
	activo	0
Update Data		
OLE DB Command		
	ced_empleado	ced_empleado
	cod_estado	cod_estado
	cod_jefeinmediato	cod_jefeinmediato
	cod_oficina	codOficina
	codNacionalidad	codNacionalidad
	ide_rol	ide_rol
	nom_empleado	nom_empleado
	cod_empleado	cod_empleado
	activo	1

a.1.3. Cargar Data Warehouse

Una vez que la base Stage se haya llenado con los datos de la fuente de datos BabelSys, podremos pasar a realizar la carga correspondiente del Stage al Data Warehouse (DW_BabelSys) también conocido como el DDS.

Debemos recordar el DDS es una base desnormalizada y que tenemos dos tipos de tablas: tablas de hechos (fact tables) y tablas de dimensiones (dimension tables); de las cuales primero serán llenadas las tablas de dimensiones y por último las tablas de hechos.

Usando SQL Server Integration Services (SSIS) crearemos los ETLs necesarios de esta segunda etapa para los flujos de datos, los cuales alimentarán al DDS.

Dentro de la solución *ISP_BabelSys* se encuentra el proyecto *ISP_DW_BabelSys* el cuál contiene los paquetes necesarios para realizar la migración del Stage al DDS.

Sin embargo antes de crear los paquetes debemos tomar en cuenta:

- 1. Carga incremental:** Al rellenar las tablas de dimensión en el DDS, tenemos el lujo de la carga incremental. No tenemos que preocuparnos de cómo extraer los datos de forma incremental a partir del Stage, porque en las tablas del Stage se añadió una columna extra denominada “activo”. Al cargar los datos en la etapa de Stage, siempre ponemos el valor de “1” en la columna que marca una actualización. Por lo tanto, sabemos que cada registro con marca en “1” en la columna “activo” debe realizar algún tratamiento. Esto nos permite

cargar los datos del Stage en el DDS de forma incremental. En otras palabras, carga sólo las filas del Stage que han cambiado desde la última ejecución ETL.

- 2. *Clave de gestión:*** En el DDS, no tenemos para mantener las claves suplentes. Las claves suplentes son gestionados en el NDS. Al hacer el paso inicial del Stage al DDS podemos usar la misma clave suplente, sin embargo en caso de que dicha clave se encuentre ocupada ya en el DDS para otro registro, se pasará el registro fallido a una tabla temporal similar a la tabla dimensional, para darle una nueva clave generada en un proceso ulterior.

- 3. *Desnormalización:*** Las tablas de dimensiones en el DDS está sin normalizar, mientras que en el Stage se encuentran normalizadas aunque sus relaciones sean lógicas. Por lo tanto, para cargar una tabla única de dimensión en el DDS, es necesario obtener datos de varias tablas en el Stage para unirse a ellas. Por ejemplo, para cargar la dimensión de rol en el DDS, en el Stage es necesario unirse al rol (TRol) el tipo de rol (TtipoRol). Esto se debe a que el rol solo tiene la clave del tipo de rol. Por favor refiérase en la Figura No. 3.2.

- 4. *Dimensión de variación lenta (“Slowly Changing Dimension” SCD):*** El SCD es una técnica utilizada para almacenar el valor histórico de los atributos de dimensión. Los valores de los atributos de dimensiones cambian a medida que pasa el tiempo. Cuando estos atributos cambian, puede sobrescribir los valores antiguos con los nuevos, o se puede conservar el valor anterior.

Las tablas de dimensiones en el DDS tiene uno y otro tipos de SCD; SCD tipo 1 (sobrescribir), SCD tipo 2 (filas), o el tipo SCD 3 (columna). Si la dimensión es SCD tipo 1, sólo sobrescribir los antiguos valores de los atributos con los nuevos valores. Si la dimensión es SCD tipo 2, tenemos que crear una nueva fila que contiene los valores de atributos y las vencerá el antiguo registro. Si la dimensión es SCD Tipo 3, tenemos que poner los viejos valores en otra columna y poner los nuevos valores en el atributo de la columna principal.

✓ **Usando SSIS para poblar el Data Warehouse
*DW_BabelSys***

I. Paquete “*pkgDwTD_Rol*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de las tablas TRol y TTipoRol desde la fuente de datos SB_BabelSys a la tabla de dimensión TD_Rol en la base DW_BabelSys. Los datos migrados serán transformados en caso de necesitarlo, debe tenerse en cuenta que se deben generar claves primarias para los registros ingresados, y al finalizar el proceso se debe cambiar el campo activo en la base de Stage a los registros procesados. Además debe tenerse en cuenta los cambios históricos de los que se contienen en el Stage y los que tiene las dimensiones.

b. DATA SOURCES

Tabla No. 3.39: pkgDwTD_Rol – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys.s a	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.DW_BabelSys.s a	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgDwTD_Rol.txt	C:\log\ pkgDwTD_Rol.tx t	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.40: pkgDwTD_Rol – Variables

NOMBRE	ALCANCE	TIPO
vgn_totalFilas	pkgDwTD_Rol	Int32

d. CONTROL FLOW

El control de flujo se encuentra constituido por:

- Data Flow Task: TD_Rol
- Execute SQL Task : Count TD_Rol
- Data Flow Task: Failed Rows
- Execute SQL Task : Delete Failed Rows
- Execute SQL Task : Cambiar bandera de Rol

Los cuales se muestran en la Figura No. 3.26:

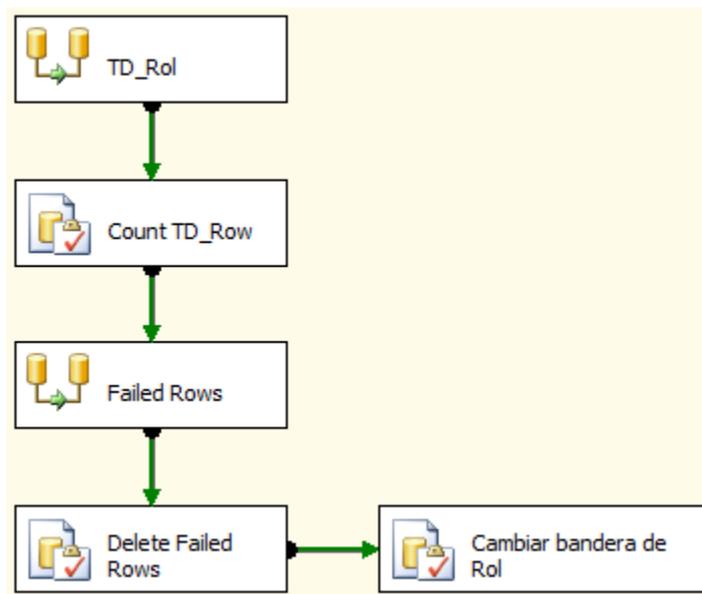


Figura No. 3.26: pkgDWTD_Rol – Control Flow

e. DATA FLOW

1. TD_Rol

El control TD_Rol tiene el siguiente flujo de datos que se presenta en la Figura No. 3.27:

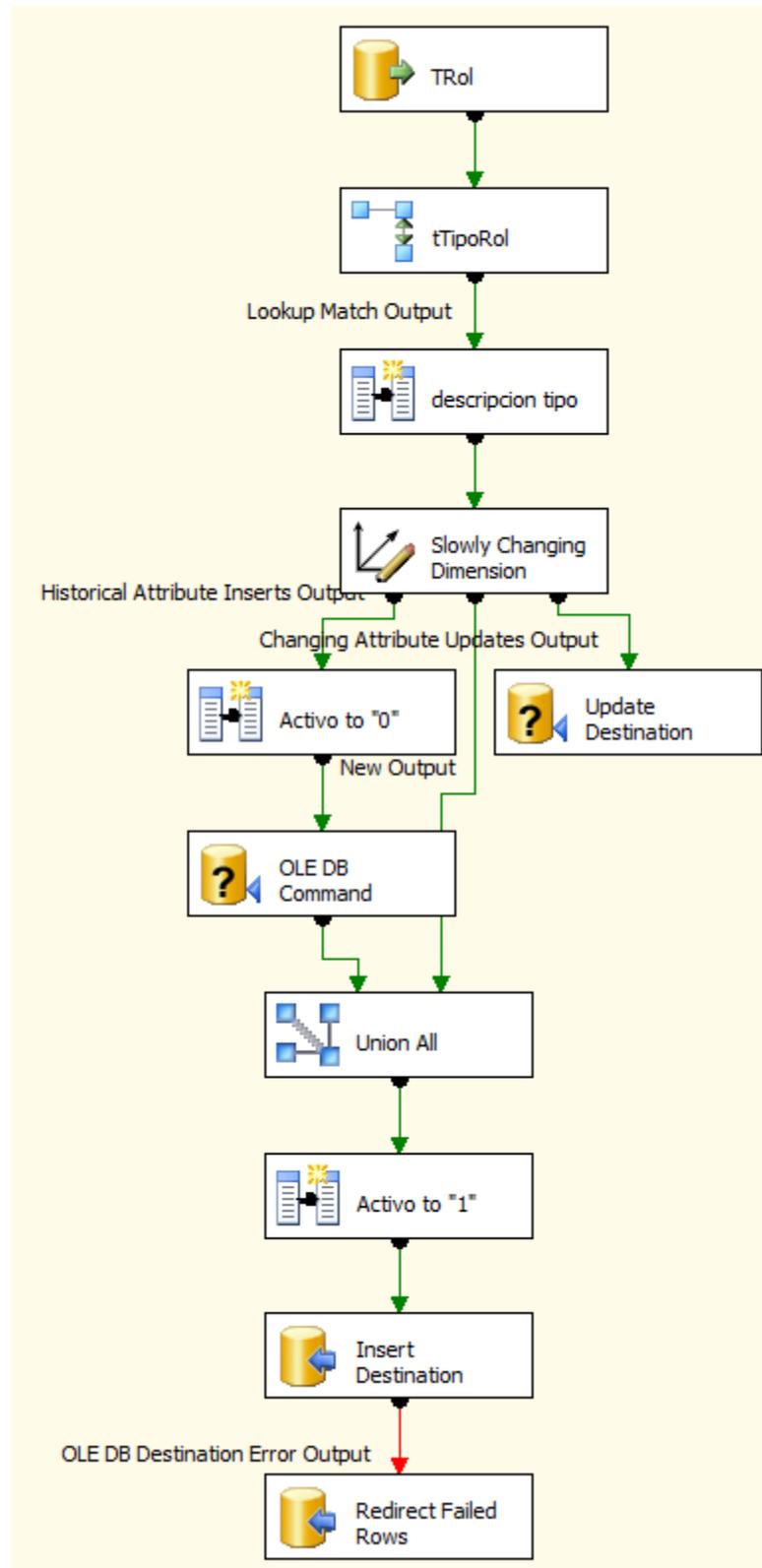


Figura No. 3.27: pkgDwTD_Rol – Data Flow (TD_Rol)

2. Failed Rows

El control Failed Rows tiene el siguiente flujo de datos que se muestra en la Figura No. 3.28:

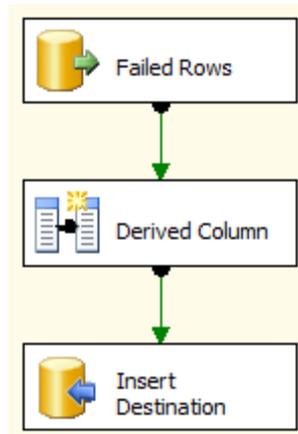


Figura No. 3.28: pkgDwTD_Rol – Data Flow (Failed Rows)

f. MAPEADO

1. TD_Rol

Tabla No. 3.41: pkgDwTD_Rol - Mapeado (TD_Rol)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Activo to "0" Derived Column	activo	0

Activo to "1"		
Derived Column		
	activo	1
Descripcion Tipo		
Derived Column		
	codigo	ide_rol
	des_tiporol	descripcion
Insert Destination		
OLE DB Destination		
	id_rol	ide_rol
	Codigo	Codigo
	nombre	nom_rol
	descripcion	dsc_rol
	id_tiporol	id_tiporol
	descripcion_tiporol	des_tiporol
	activo	activo
OLE DB Command		
OLE DB Command		
	activo	activo
	id_rol	ide_rol
Redirect Failed Rows		
OLE DB Destination		
	id_rol	ide_rol
	codigo	codigo
	nombre	nom_rol
	descripcion	dsc_rol
	id_tiporol	id_tiporol
	descripcion_tiporol	des_tiporol
	activo	activo
Slowly Changing		

Dimension Slowly Changing Dimension		
	activo	activo
	codigo	codigo
	dsc_rol	dsc_rol
	des_tiporol	des_tiporol
	ide_rol	ide_rol
	id_tiporol	id_tiporol
	nom_rol	nom_rol
TRol OLE DB Source		
	id_tiporol	id_tiporol
	ide_rol	ide_rol
	nom_rol	nom_rol
	dsc_rol	dsc_rol
activo	activo	
TTipoRol LookUp Input		
	id_tiporol	id_tiporol
Union All (Input 1) Union All		
	id_tiporol	id_tiporol
	ide_rol	ide_rol
	nom_rol	nom_rol
	dsc_rol	dsc_rol
	activo	activo
	descripcion	descripcion
codigo	codigo	

	des_tiporol	des_tiporol
Union All (Input 2)		
Union All		
	id_tiporol	id_tiporol
	ide_rol	ide_rol
	nom_rol	nom_rol
	dsc_rol	dsc_rol
	activo	activo
	descripcion	descripcion
	codigo	codigo
	des_tiporol	des_tiporol
Update Destination		
OLE DB Command		
	descripcion	dsc_rol
	descripcion_tiporol	des_tiporol
	id_rol	ide_rol

2. Failed Rows

Tabla No. 3.42: pkgDwTD_Rol – Mapeado (Failed Rows)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Derived Column		

Derived Column		
	newId	id_rol + @[User::vgn_totalFilas]
Failed Rows		
OLE DB Source		
	id_rol	id_rol
	Codigo	Codigo
	nombre	nombre
	descripcion	descripcion
	id_tiporol	id_tiporol
	descripcion_tiporol	descripcion_tiporol
	activo	activo
Insert Destination		
OLE DB Destination		
	id_rol	newId
	Codigo	Codigo
	nombre	nombre
	descripcion	descripcion
	id_tiporol	id_tiporol
	descripcion_tiporol	descripcion_tiporol
	activo	activo

II. Paquete “*pkgDwTD_Proyecto*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de las tablas TProyecto desde la fuente de datos SB_BabelSys a la tabla de dimensión TD_Proyecto en la base DW_BabelSys. Los datos

migrados serán transformados en caso de necesitarlo, debe tenerse en cuenta que se deben generar claves primarias para los registros ingresados, y al finalizar el proceso se debe cambiar el campo activo en la base de Stage a los registros procesados. Además debe tenerse en cuenta los cambios históricos de los que se contienen en el Stage y los que tiene las dimensiones.

b. DATA SOURCES

Tabla No. 3.43: pkgDwTD_Proyecto – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSy s.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.DW_BabelSy s.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgDwTD_Proyecto.txt	C:\log\ pkgDwTD_Projectova .txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.44: pkgDwTD_Proyecto - Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido por:

- Data Flow Task: TD_Proyecto
- Execute SQL Task : Cambiar bandera nac

Los cuales se muestran en la Figura No. 3.29:

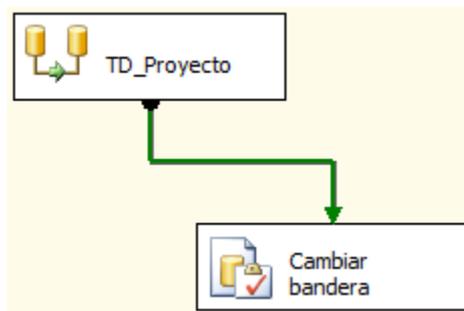


Figura No. 3.29: pkgDwTD_Proyecto – Control Flow

e. DATA FLOW

TD_Proyecto

El control TD_Proyecto tiene el siguiente flujo de datos que se indica en la Figura No. 3.30:

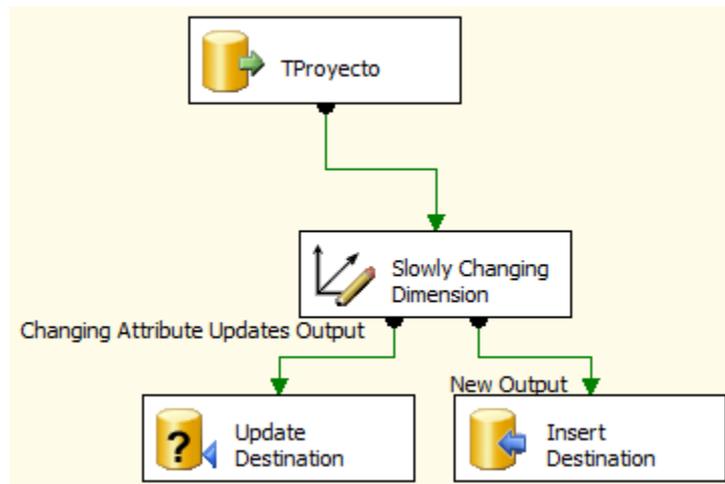


Figura No. 3.30: pkgDwTD_Proyecto – Data Flow (TD_Proyecto)

f. MAPEADO

Tabla No. 3.45: pkgDwTD_Proyecto – Mapeado

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Insert Destination		
OLE DB Destination		
	id_proyecto	ide_proyecto
	codigo	cod_proyecto
	nombre	nom_proyecto
	descripcion	dsc_proyecto
	fec_inicio	fec_ini_pro
	fec_fin	fec_fin_pro
	estado	Estado
	activo	activo

Slowly Changing Dimension Slowly Changing Dimension		
	activo	activo
	cod_proyecto	cod_proyecto
	dsc_proyecto	dsc_proyecto
	estado	estado
	fec_ini_pro	fec_ini_pro
	fec_fin_pro	fec_fin_pro
	ide_proyecto	ide_proyecto
	nom_proyecto	nom_proyecto
TProyecto OLE DB Source		
	ide_proyecto	ide_proyecto
	cod_proyecto	cod_proyecto
	nom_proyecto	nom_proyecto
	dsc_proyecto	dsc_proyecto
	fec_ini_pro	fec_ini_pro
	fec_fin_pro	fec_fin_pro
	activo	activo
	estado	estado
Update Destination OLE DB Command		
	codigo	cod_proyecto
	descripcion	dsc_proyecto
	estado	Estado
	fec_inicio	fec_ini_pro
	fec_fin	fec_fin_pro
	nombre	nom_proyecto
	id_proyecto	ide_proyecto

III. Paquete “*pkgDwTD_Oficina*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de las tablas TOficina desde la fuente de datos SB_BabelSys a la tabla de dimensión TD_Oficina en la base DW_BabelSys. Los datos migrados serán transformados en caso de necesitarlo, debe tenerse en cuenta que se deben generar claves primarias para los registros ingresados, y al finalizar el proceso se debe cambiar el campo activo en la base de Stage a los registros procesados. Además debe tenerse en cuenta los cambios históricos de los que se contienen en el Stage y los que tiene las dimensiones.

b. DATA SOURCES

Tabla No. 3.46: *pkgDwTD_Oficina* – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys. sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.DW_BabelSys. sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
<i>pkgDwTD_Oficina.txt</i>	C:\log\ <i>pkgDwTD_Oficina.t</i>	Genera un archivo plano con

xt	datos de errores si es que se produjeran.
----	---

c. VARIABLES

Tabla No. 3.47: pkgDwTD_Oficina – Variables

NOMBRE	ALCANCE	TIPO
vgn_totalOficinas	pkgDwTD_Oficina	Int32

d. CONTROL FLOW

El control de flujo se encuentra constituido por:

- Data Flow Task: TD_Oficina
- Execute SQL Task : Count TD_Oficina
- Data Flow Task: Failed Rows
- Execute SQL Task : Delete Failed Rows
- Execute SQL Task : Cambiar bandera

Como se muestra en la Figura No. 3.31:

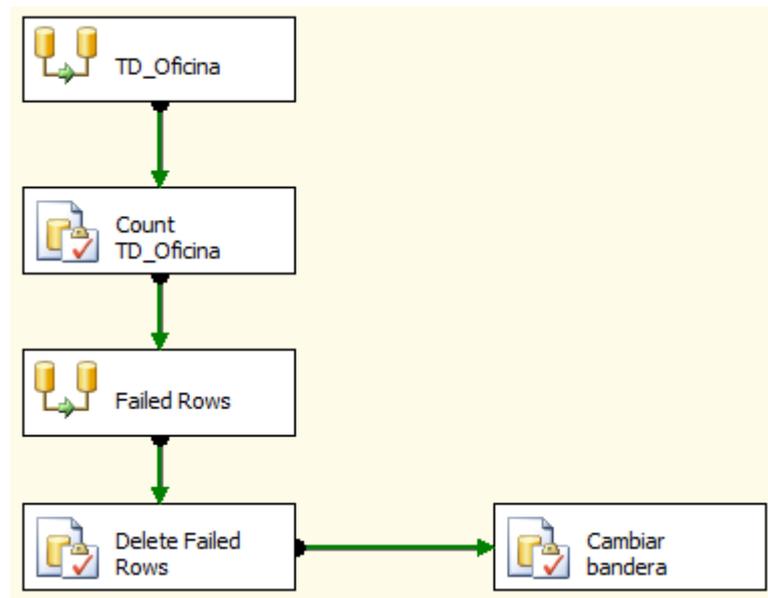


Figura No. 3.31: pkgDwTD_Oficina – Control Flow

e. DATA FLOW

1. TD_Oficina

El control TD_Oficina tiene el siguiente flujo de datos que se indica en la Figura No. 3.32:

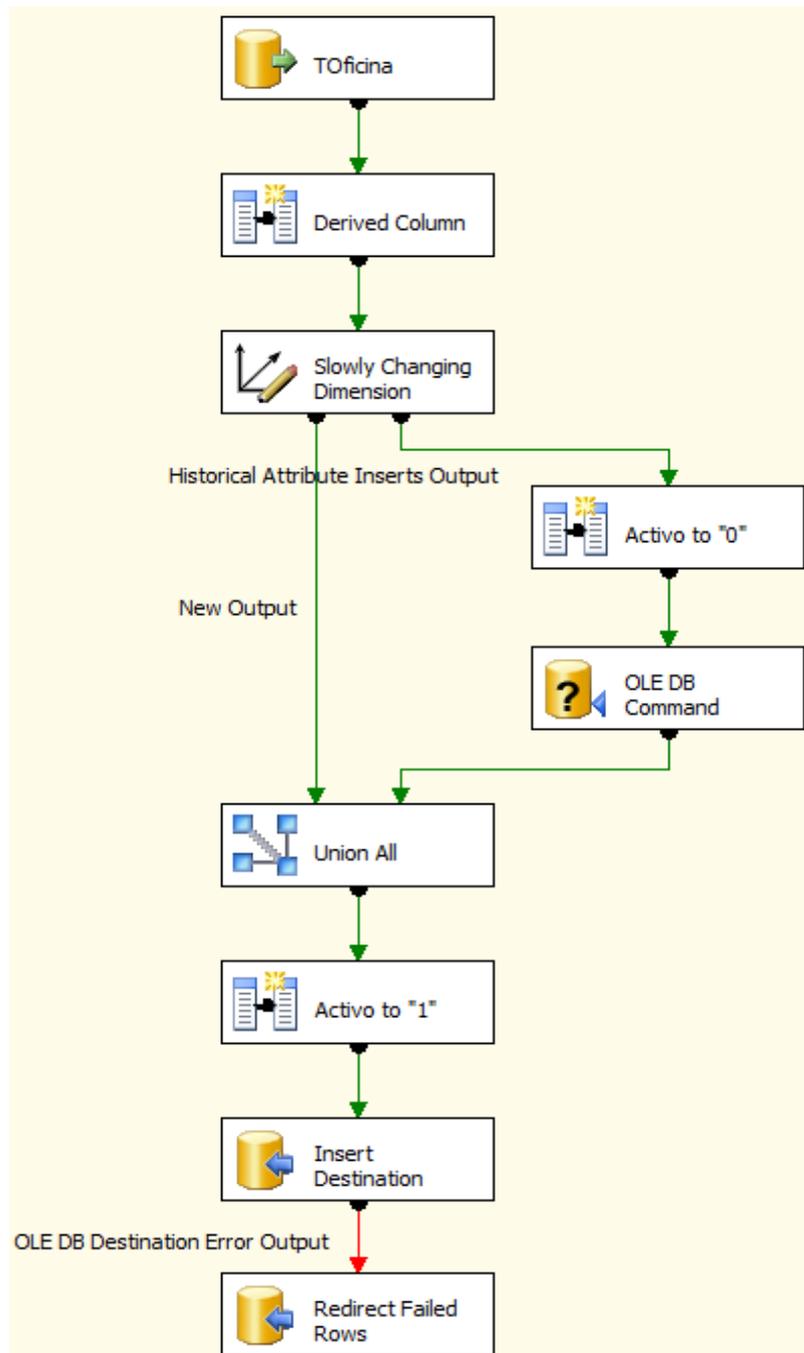


Figura No. 3.32: pkgDwTD_Oficina – Data Flow (TD_Oficina)

2. Failed Rows

El control Failed Rows tiene el siguiente flujo de datos, que se muestra en la Figura No. 3.33:

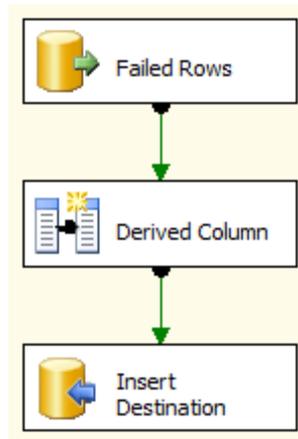


Figura No. 3.33: pkgDwTD_Oficina – Data Flow (Failed Rows)

f. MAPEADO

1. TD_Oficina

Tabla No. 3.48: pkgDwTD_Oficina – Mapeado (TD_Oficina)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Activo to "0"		
Derived Column	activo	0
Activo to "1"		
Derived Column	activo	1
Derived Column		

Derived Column		
	id_oficina	cod_oficina
	nombre	nom_oficina
Insert Destination		
OLE DB Destination		
	id_oficina	id_oficina
	cod_oficina	codigo
	nombre	nombre
	activo	activo
OLE DB Command		
OLE DB Command		
	activo	activo
	id_oficina	id_oficina
Redirect Failed Rows		
OLE DB Destination		
	id_oficina	id_oficina
	cod_oficina	codigo
	nombre	nombre
	activo	activo
Slowly Changing Dimension		
Slowly Changing Dimension		
	activo	activo
	cod_oficina	cod_oficina
	id_oficina	id_oficina
	nombre	nombre
TOficina		
OLE DB Source		
	cod_oficina	cod_oficina

Union All (Input 1) Union All	nom_oficina	nom_oficina
	activo	activo
	cod_oficina	cod_oficina
	nom_oficina	nom_oficina
	activo	activo
	id_oficina	id_oficina
Union All (Input 2) Union All	nombre	nombre
	cod_oficina	cod_oficina
	nom_oficina	nom_oficina
	activo	activo
	id_oficina	id_oficina
	nombre	nombre

2. Failed Rows

Tabla No. 3.49: pkgDwTD_Oficina – Mapeado (Failed Rows)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Derived Column		
Derived Column	idOficina	id_oficina +

		@[User::vgn_totalOficinas]	
Failed Rows OLE DB Source			
	id_oficina	id_oficina	
	codigo	codigo	
	nombre	nombre	
	activo	activo	
Insert Destination OLE DB Destination			
	id_oficina	idOficina	
	codigo	codigo	
	nombre	nombre	
	activo	activo	

IV. Paquete “*pkgDwTD_Nacionalidad*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de las tablas TNacionalidad desde la fuente de datos SB_BabelSys a la tabla de dimensión TD_Nacionalidad en la base DW_BabelSys. Los datos migrados serán transformados en caso de necesitarlo, debe tenerse en cuenta que se deben generar claves primarias para los registros ingresados, y al finalizar el proceso se debe cambiar el campo activo en la base de Stage a los registros procesados. Además debe tenerse en cuenta los cambios históricos de los que se contienen en el Stage y los que tiene las dimensiones.

b. DATA SOURCES

Tabla No. 3.50: pkgDwTD_Nacionalidad – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSy s.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.DW_BabelS ys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgDwTD_Nacionalidad.txt	C:\log\ pkgDwTD_Nacionalida d.txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.51: pkgDwTD_Nacionalidad – Variables

NOMBRE	ALCANCE	TIPO
vgn_totalNacionalidad	pkgDwTD_Nacionalidad	Int32

d. CONTROL FLOW

El control de flujo se encuentra constituido por:

- Data Flow Task: TD_Nacionalidad

- Execute SQL Task : Count TD_Nacionalidad
- Data Flow Task: Failed Rows
- Execute SQL Task : Delete Failed Rows
- Execute SQL Task : Cambiar bandera nac

Como se muestra en la Figura No. 3.34:

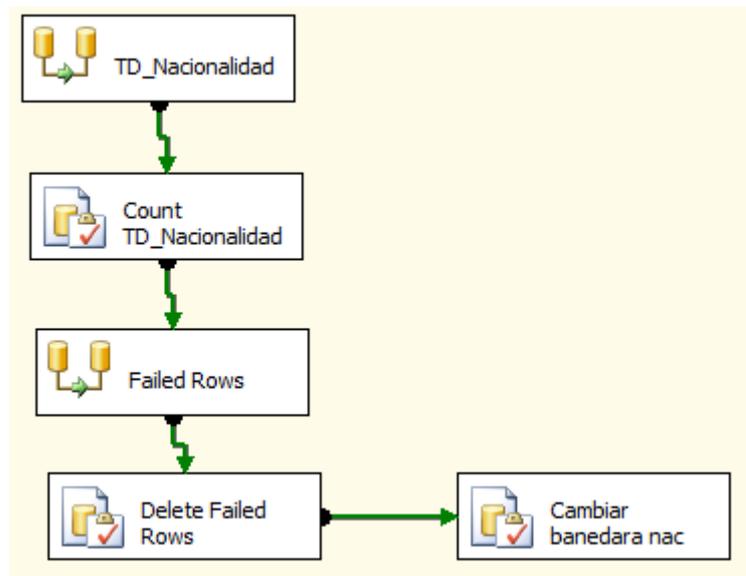


Figura No. 3.34: pkgDwTD_Nacionalidad – Control Flow

e. DATA FLOW

1. TD_Nacionalidad

El control TD_Nacionalidad tiene el siguiente flujo de datos, como se muestra en la Figura No. 3.35:

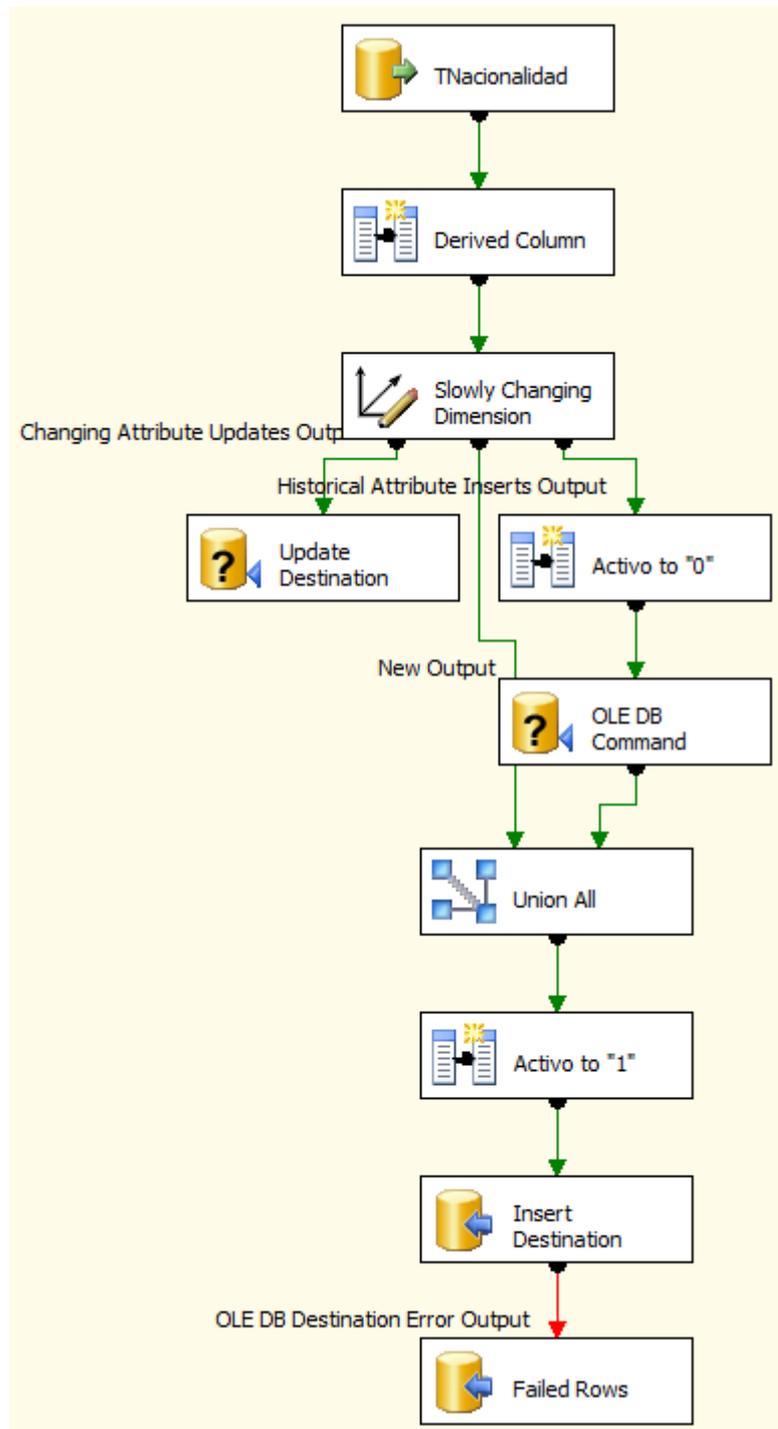


Figura No. 3.35: pkgDwTD_Nacionalidad – Data Flow (TD_Nacionalidad)

2. Failed Rows

El control Failed Rows tiene el siguiente flujo de datos, como se indica en la Figura No. 3.36:

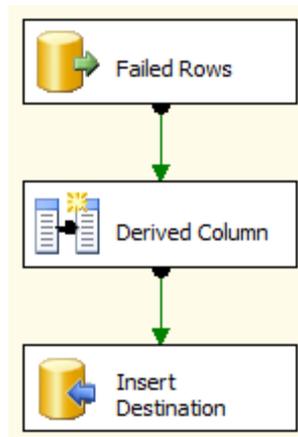


Figura No. 3.36: pkgDwTD_Nacionalidad – Data Flow (Failed Rows)

f. MAPEADO

1. TD_Nacionalidad

Tabla No. 3.52: pkgDwTD_Nacionalidad – Mapeado (TD_Nacionalidad)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Activo to "0"		
Derived Column		

	activo	0
Activo to "1"		
Derived Column		
	activo	1
Derived Column		
Derived Column		
	codigo	codnacionalidad
	id_nacionalidad	codnacionalidad
Insert Destination		
OLE DB Destination		
	id_nacionalidad	id_nacionalidad
	codigo	codigo
	nombre	nomnacionalidad
	activo	activo
OLE DB Command		
OLE DB Command		
	activo	activo
	id_nacionalidad	id_nacionalidad
Failed Rows		
OLE DB Destination		
	id_nacionalidad	id_nacionalidad
	codigo	codigo
	nombre	nomnacionalidad
	activo	activo
Slowly Changing Dimension		
Slowly Changing Dimension		
	activo	activo
	codigo	codigo

	id_nacionalidad	id_nacionalidad
	nomnacionalidad	nomnacionalidad
TNacionalidad		
OLE DB Source		
	codnacionalidad	codnacionalidad
	nomnacionalidad	nomnacionalidad
	activo	activo
Union All (Input 1)		
Union All		
	id_nacionalidad	id_nacionalidad
	codigo	codigo
	nomnacionalidad	nomnacionalidad
	activo	activo
	codnacionalidad	codnacionalidad
Union All (Input 2)		
Union All		
	id_nacionalidad	id_nacionalidad
	codigo	codigo
	nomnacionalidad	nomnacionalidad
	activo	activo
	codnacionalidad	codnacionalidad
Update Destination		
OLE DB Command		
	nombre	nomnacionalidad
	id_nacionalidad	id_nacionalidad

2. Failed Rows

Tabla No. 3.53: pkgDwTD_Nacionalidad – Mapeado (Failed Rows)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Derived Column Derived Column	idNacionalidad	id_nacionalidad + @[User::vgn_totalNacionalidad]
Failed Rows OLE DB Source	id_nacionalidad	id_nacionalidad
	codigo	codigo
	nombre	nombre
	activo	activo
Insert Destination OLE DB Destination	id_nacionalidad	idNacionalidad
	codigo	codigo
	nombre	nombre
	activo	activo

V. Paquete “*pkgDwTD_Empresa*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de las tablas TEmpresa y TNacionalidad desde la fuente de datos SB_BabelSys a la tabla de dimensión TD_Empresa en la base DW_BabelSys. Los datos migrados serán transformados en caso de necesitarlo, debe tenerse en cuenta que se deben generar claves primarias para los registros ingresados, y al finalizar el proceso se debe cambiar el campo activo en la base de Stage a los registros procesados. Además debe tenerse en cuenta los cambios históricos de los que se contienen en el Stage y los que tiene las dimensiones.

b. DATA SOURCES

Tabla No. 3.54: pkgDwTD_Empresa – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys .sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.DW_BabelSy s.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgDwTD_Empresa.txt	C:\log\ pkgDwTD_Empresa. txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.55: pkgDwTD_Empresa – Variables

NOMBRE	ALCANCE	TIPO
vgn_totalEmpresa	pkgDwTD_Empresa	Int32

d. CONTROL FLOW

El control de flujo se encuentra constituido por:

- Data Flow Task: TD_Empresa
- Execute SQL Task : Count TD_Empresa
- Data Flow Task: Failed Rows
- Execute SQL Task : Delete Failed Rows
- Execute SQL Task : Cambiar bandera

Como se muestra en la Figura No. 3.37:

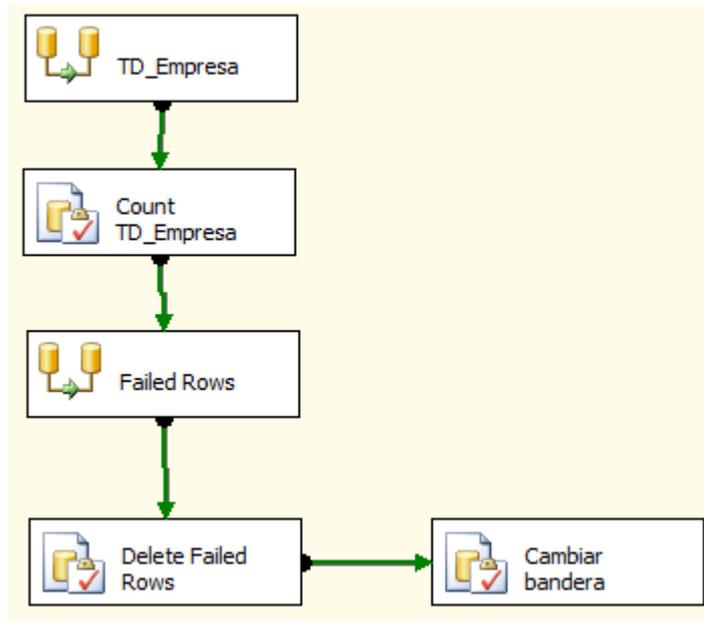


Figura No. 3.37: pkgDwTD_Empresa – Control Flow

e. DATA FLOW

1. TD_Empresa

El control TD_Empresa tiene el siguiente flujo de datos, como se indica en la Figura No. 3.38:

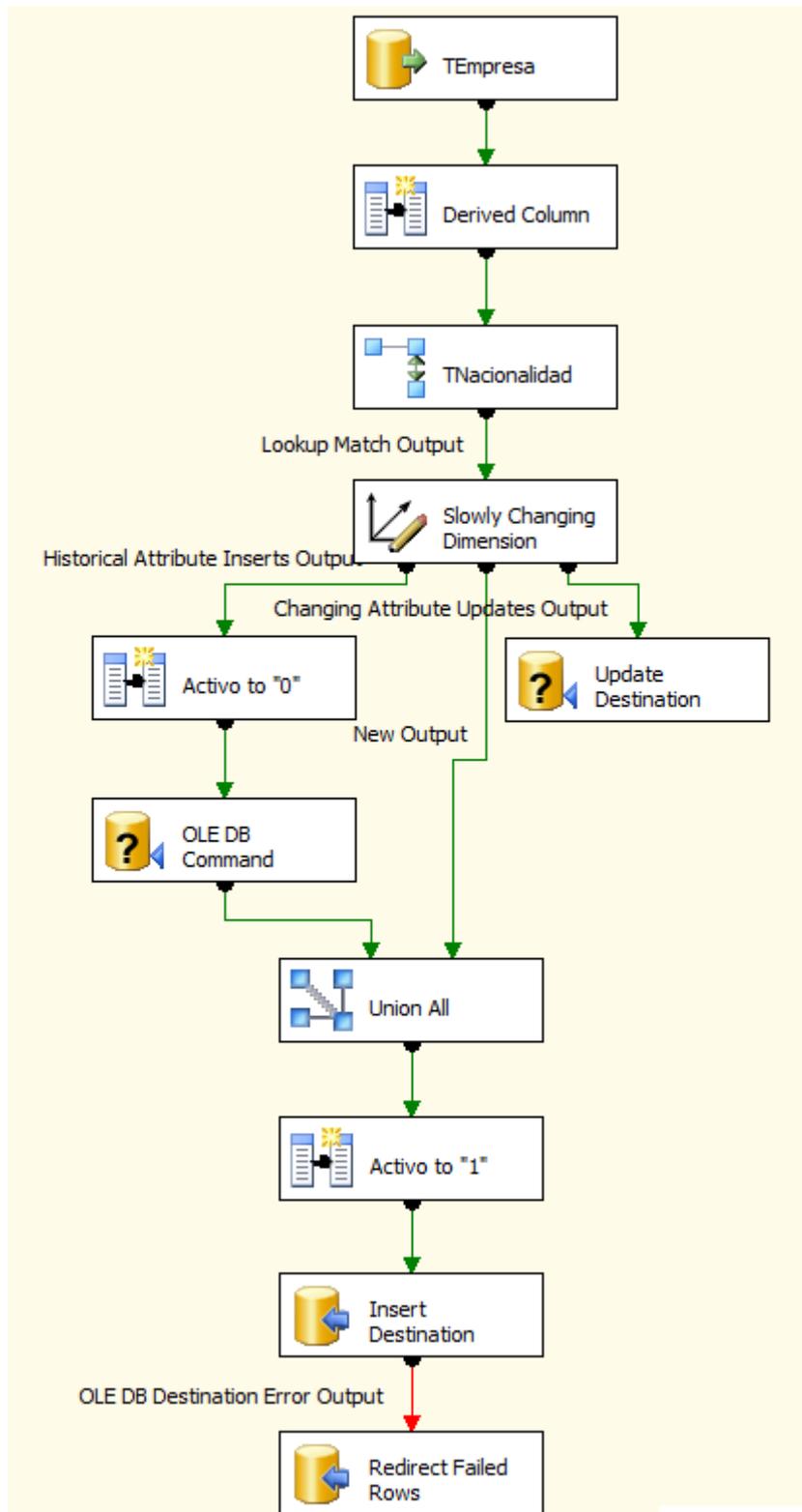


Figura No. 3.38: pkgDwTD_Empresa – Data Flow (TD_Empresa)

2. Failed Rows

El control Failed Rows tiene el siguiente flujo de datos:

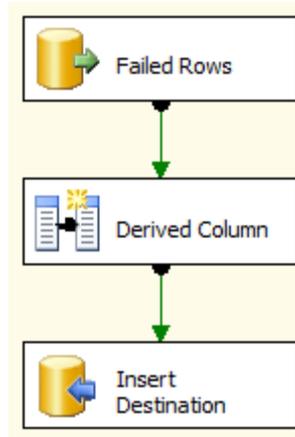


Figura No. 3.39: pkgDwTD_Empresa – Data Flow (Failed Rows)

f. MAPEADO

1. TD_Nacionalidad

Tabla No. 3.56: pkgDwTD_Empresa – Mapeado (TD_Empresa)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Activo to "0"		
Derived Column		
	activo	0
Activo to "1"		

Derived Column		
	activo	1
Derived Column		
Derived Column		
	cod_empresa	id_empresa
Insert Destination		
OLE DB Destination		
	id_empresa	ide_empresa
	codigo	cod_empresa
	nombre	nom_empresa
	direccion	dsc_direccion
	cod_nacionalidad	codnacionalidad
	nombre_nacionalidad	nomnacionalidad
	activo	activo
OLE DB Command		
OLE DB Command		
	activo	activo
	id_empresa	ide_empresa
Redirect Failed Rows		
OLE DB Destination		
	id_empresa	ide_empresa
	codigo	cod_empresa
	nombre	nom_empresa
	direccion	dsc_direccion
	cod_nacionalidad	codnacionalidad
	nombre_nacionalidad	nomnacionalidad
	activo	activo
Slowly Changing Dimension		
Slowly Changing		

Dimension		
	activo	activo
	codnacionalidad	codnacionalidad
	cod_empresa	cod_empresa
	dsc_direccion	dsc_direccion
	ide_empresa	ide_empresa
	nom_empresa	nom_empresa
	nomnacionalidad	nomnacionalidad
TEmpresa		
OLE DB Source		
	ide_empresa	ide_empresa
	nom_empresa	nom_empresa
	dsc_empresa	dsc_empresa
	codnacionalidad	codnacionalidad
	activo	activo
TNacionalidad		
LookUp input		
	codnacionalidad	codnacionalidad
Union All (Input 1)		
Union All		
	id_empresa	id_empresa
	nom_empresa	nom_empresa
	dsc_direccion	dsc_direccion
	codnacionalidad	codnacionalidad
	activo	activo
	cod_empresa	cod_empresa
	nomnacionalidad	nomnacionalidad
Union All (Input 2)		
Union All		

	id_empresa	id_empresa
	nom_empresa	nom_empresa
	dsc_direccion	dsc_direccion
	codnacionalidad	codnacionalidad
	activo	activo
	cod_empresa	cod_empresa
	nomnacionalidad	nomnacionalidad
Update Destination		
OLE DB Command		
	cod_nacionalidad	codnacionalidad
	direccion	dsc_direccion
	nombre_nacionalidad	nomnacionalidad
	id_empresa	ide_empresa

2. Failed Rows

Tabla No. 3.57: pkgDwTD_Empresa – Mapeado (TD_Empresa)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Derived Column		
Derived Column	idEmpresa	id_empresa + @[User::vgn_totalEmpresa]
Failed Rows		

OLE DB Source		
	id_empresa	id_empresa
	codigo	codigo
	nombre	nombre
	direccion	direccion
	cod_nacionalidad	cod_nacionalidad
	nombre_nacionalidad	nombre_nacionalidad
	activo	activo
Insert Destination		
OLE DB Destination		
	id_empresa	id_empresa
	codigo	codigo
	nombre	nombre
	direccion	direccion
	cod_nacionalidad	cod_nacionalidad
	nombre_nacionalidad	nombre_nacionalidad
	activo	activo

VI. Paquete “*pkgDwTD_Empleado*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad migrar los datos de las tablas TEmpleado, TOficina, Testado, Trol, TNacionalidad y TEmpleado para obtener información del Jefe Inmediato respectivamente, desde la fuente de datos SB_BabelSys a la tabla de dimensión TD_Empleado en la base DW_BabelSys. Los datos migrados serán transformados en caso de necesitarlo, debe tenerse en cuenta que se deben generar claves primarias para los registros

ingresados, y al finalizar el proceso se debe cambiar el campo activo en la base de Stage a los registros procesados. Además debe tenerse en cuenta los cambios históricos de los que se contienen en el Stage y los que tiene las dimensiones.

b. DATA SOURCES

Tabla No. 3.58: pkgDwTD_Empleado – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.SB_BabelSys.sa	Server Oficinas Babel Software Latacunga	Se conecta a la base Stage.
192.168.30.252.DW_BabelSy s.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgDwTD_Empleado.txt	C:\log\ pkgDwTD_Empleado .txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.59: pkgDwTD_Empleado – Variables

NOMBRE	ALCANCE	TIPO
--------	---------	------

d. CONTROL FLOW

El control de flujo se encuentra constituido por:

- Data Flow Task: TD_Empleado
- Execute SQL Task : Count TD_Empleado
- Data Flow Task: Failed Rows
- Execute SQL Task : Delete Failed Rows
- Execute SQL Task : Cambiar bandera

Como se muestra en la Figura No. 3.40:

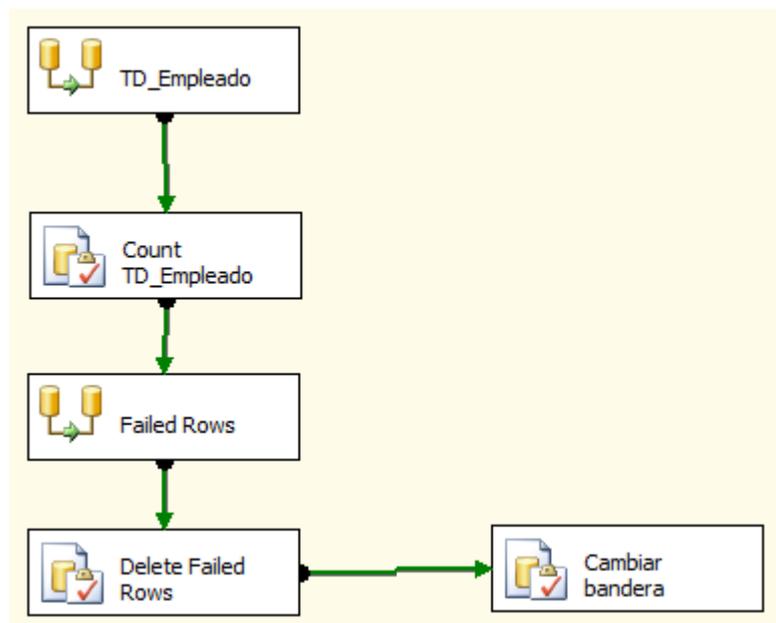


Figura No. 3.40: pkgDwTD_Empleado – Control Flow

e. DATA FLOW

1. TD_Empleado

El control TD_Empleado tiene el siguiente flujo de datos, tal como se muestra en la Figura No. 3.41:

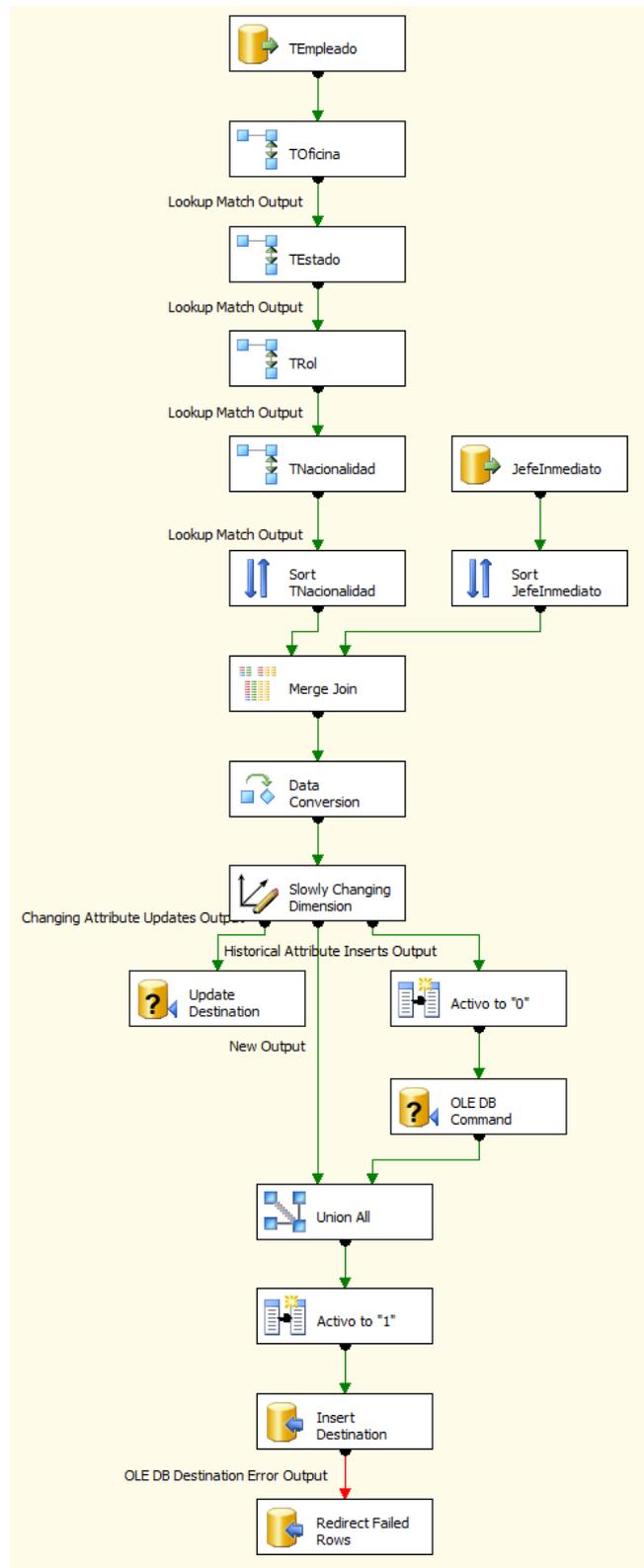


Figura No. 3.41: pkgDwTD_Empleado – Data Flow (TD_Empleado)

2. Failed Rows

El control Failed Rows tiene el siguiente flujo de datos:

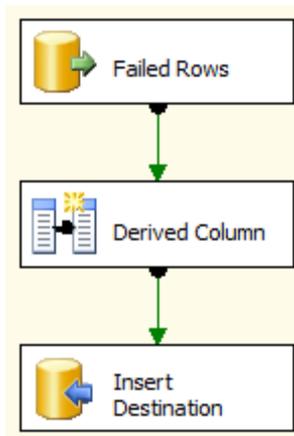


Figura No. 3.42: pkgDwTD_Empleado – Data Flow (Failed Rows)

f. MAPEADO

1. TD_Nacionalidad

Tabla No. 3.60: pkgDwTD_Empleado – Mapeado (TD_Empleado)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Activo to "0"		
Derived Column	activo	0

Activo to "1" Derived Column		
	activo	1
Data Conversion Data Conversion		
	Copy of nom_empleado	nom_empleado
	Copy of ord	ord
	Copy of ced_empleado	ced_empleado
	Copy of ced_jefe	ced_jefe
	Copy of nom_jefe	nom_jefe
JefeInmediato OLE DB Destination		
	cod_empleado	cod_empleado
	ced_empleado	ced_empleado
	nom_empleado	nom_empleado
Merge Join Merge Join		
	cod_empleado	cod_empleado
	cod_jefeinmediato	cod_jefeinmediato
	ide_rol	ide_rol
	nom_empleado	nom_empleado
	codnacionalidad	codnacionalidad
	cod_oficina	cod_oficina
	activo	activo
	ord	ord
	ced_empleado	ced_empleado
	cod_estado	cod_estado
	nom_oficina	nom_oficina
	nom_estado	nom_estado
	dsc_estado	dsc_estado

	nom_rol	nom_rol
	nomnacionalidad	nomnacionalidad
	cod_jefe	cod_jefe
	nom_jefe	nom_jefe
Insert Destination		
OLE DB Destination		
	id_employado	Copy of ord
	codigo	cod_employado
	cedula	Copy of ced_employado
	nombre	Copy of nom_employado
	cod_jefeinmediato	cod_jefeinmediato
	cedula_jefe	Copy of ced_jefe
	nombre_jefe	Copy of nom_jefe
	codigo_rol	ide_rol
	nombre_rol	nom_rol
	codigo_nacionalidad	codnacionalidad
	nombre_nacionalidad	nomnacionalidad
	codigo_oficina	cod_oficina
	nombre_oficina	nom_oficina
	cod_estado	cod_estado
	nom_estado	nom_estado
	dsc_estado	dsc_estado
	activo	activo
OLE DB Command		
OLE DB Command		
	activo	activo
	id_employado	Copy of ord
Redirect Failed Rows		
OLE DB Destination		

	id_empleado	Copy of ord
	codigo	cod_empleado
	cedula	Copy of ced_empleado
	nombre	Copy of nom_empleado
	cod_jefeinmediato	cod_jefeinmediato
	cedula_jefe	Copy of ced_jefe
	nombre_jefe	Copy of nom_jefe
	codigo_rol	ide_rol
	nombre_rol	nom_rol
	codigo_nacionalidad	codnacionalidad
	nombre_nacionalidad	nomnacionalidad
	codigo_oficina	cod_oficina
	nombre_oficina	nom_oficina
	cod_estado	cod_estado
	nom_estado	nom_estado
	dsc_estado	dsc_estado
	activo	activo
Slowly Changing Dimension		
Slowly Changing Dimension		
	activo	activo
	Copy of ced_empleado	Copy of ced_empleado
	Copy of ced_jefe	Copy of ced_jefe
	cod_estado	cod_estado
	cod_jefeinmediato	cod_jefeinmediato
	cod_empleado	cod_empleado
	codnacionalidad	codnacionalidad

	cod_oficina	cod_oficina
	ide_rol	ide_rol
	dsc_estado	dsc_estado
	Copy of nom_empleado	Copy of nom_empleado
	Copy of nom_jefe	Copy of nom_jefe
	nomnacionalidad	nomnacionalidad
	nom_oficina	nom_oficina
	nom_rol	nom_rol
Sort JefeInmediato		
Sort		
	cod_empleado	cod_empleado
	ced_empleado	ced_empleado
	nom_empleado	nom_empleado
Sort TNacionalidad		
Sort		
	cod_empleado	cod_empleado
TEmpleado		
OLE DB Source		
	cod_empleado	cod_empleado
	cod_jefeinmediato	cod_jefeinmediato
	ide_rol	ide_rol
	nom_empleado	nom_empleado
	codnacionalidad	codnacionalidad
	cod_oficina	cod_oficina
	activo	activo
	ord	ord
	ced_empleado	ced_empleado
	cod_estado	cod_estado
TEstado		

LookUp Input		
	cod_estado	cod_estado
TNacionalidad		
LookUp Input		
	nomnacionalidad	nomnacionalidad
TOficina		
LookUp Input		
	nom_oficina	nom_oficina
TRol		
LookUp Input		
	nom_rol	nom_rol
Union All (Input 1)		
Union All		
	cod_empleado	cod_empleado
	cod_jefeinmediato	cod_jefeinmediato
	ide_rol	ide_rol
	nom_empleado	nom_empleado
	codnacionalidad	codnacionalidad
	cod_oficina	cod_oficina
	activo	activo
	ord	ord
	ced_empleado	ced_empleado
	cod_estado	cod_estado
	nom_oficina	nom_oficina
	nom_estado	nom_estado
	dsc_estado	dsc_estado
	nom_rol	nom_rol
	nomnacionalidad	nomnacionalidad
	ced_jefe	ced_jefe

	nom_jefe	nom_jefe
	Copy of nom_empleado	Copy of nom_empleado
	Copy of ord	Copy of ord
	Copy of ced_empleado	Copy of ced_empleado
	Copy of ced_jefe	Copy of ced_jefe
	Copy of nom_jefe	Copy of nom_jefe
Union All (Input 2)		
Union All		
	cod_empleado	cod_empleado
	cod_jefeinmediato	cod_jefeinmediato
	ide_rol	ide_rol
	nom_empleado	nom_empleado
	codnacionalidad	codnacionalidad
	cod_oficina	cod_oficina
	activo	activo
	ord	ord
	ced_empleado	ced_empleado
	cod_estado	cod_estado
	nom_oficina	nom_oficina
	nom_estado	nom_estado
	dsc_estado	dsc_estado
	nom_rol	nom_rol
	nomnacionalidad	nomnacionalidad
	ced_jefe	ced_jefe
	nom_jefe	nom_jefe
	Copy of nom_empleado	Copy of nom_empleado
	Copy of ord	Copy of ord
	Copy of ced_empleado	Copy of

		ced_empleado
	Copy of ced_jefe	Copy of ced_jefe
	Copy of nom_jefe	Copy of nom_jefe
Update Destination		
OLE DB Command		
	cedula_jefe	Copy of ced_jefe
	cod_jefeinmediato	cod_jefeinmediato
	codigo_nacionalidad	Codnacionalidad
	codigo_oficina	cod_oficina
	codigo_rol	ide_rol
	dsc_estado	dsc_estado
	nom_estado	nom_estado
	nombre_jefe	Copy of nom_jefe
	nombre_nacionalidad	nomnacionalidad
	nombre_oficina	nom_oficina
	nombre_rol	nom_rol
	id_empleado	Copy of ord

2. Failed Rows

Tabla No. 3.61: pkgDwTD_Empleado – Mapeado (Failed Rows)

COMPONENTE	COLUMNA SALIENTE (Source)	COLUMNA ENTRANTE (Objetivo)
Derived Column		

Derived Column		
	idEmpleado	id_empleado + @[User::vgn_totalEmpleados]
Failed Rows		
OLE DB Source		
	codigo	codigo
	cedula	cedula
	nombre	nombre
	cod_jefeinmediato	cod_jefeinmediato
	cedula_jefe	cedula_jefe
	nombre_jefe	nombre_jefe
	codigo_rol	codigo_rol
	nombre_rol	nombre_rol
	codigo_nacionalidad	codigo_nacionalidad
	nombre_nacionalidad	nombre_nacionalidad
	codigo_oficina	codigo_oficina
	nombre_oficina	nombre_oficina
	cod_estado	cod_estado
	nom_estado	nom_estado
	dsc_estado	dsc_estado
	id_empleado	id_empleado
	activo	activo
Insert Destination		
OLE DB Destination		
	codigo	codigo
	cedula	cedula
	nombre	nombre
	cod_jefeinmediato	cod_jefeinmediato

cedula_jefe	cedula_jefe
nombre_jefe	nombre_jefe
codigo_rol	codigo_rol
nombre_rol	nombre_rol
codigo_nacionalidad	codigo_nacionalidad
nombre_nacionalidad	nombre_nacionalidad
codigo_oficina	codigo_oficina
nombre_oficina	nombre_oficina
cod_estado	cod_estado
nom_estado	nom_estado
dsc_estado	dsc_estado
id_empleado	id_empleado
activo	activo

VII. Paquete “*pkgDwTH_Asignacion*”

a. FUNCIONALIDAD

El presente paquete tiene como finalidad llenar la tabla de hechos TH_Asignacion en la cual se unirán las claves de las dimensiones correspondientes y para ello utilizaremos como tablas principales las tablas del Stage TProyectoEmpleado en la cual se encuentra la asignación del empleado a un proyecto y la tabla TProyecto Empresa la cual nos dirá que proyectos pertenecen a que empresas clientes. Los datos migrados serán insertados en la tabla de hechos de manera diaria y se obtendrá su porcentaje de asignación como ya se explicó anteriormente (Ver Ecuación 1), omitiendo los fines de semana, y estarán delimitados por las fechas de inicio y fin de asignación de la tabla TProyectoEmpleado; al finalizar el proceso se

debe cambiar el campo activo en la base de Stage a los registros procesados.

b. DATA SOURCES

Tabla No. 3.62: pkgDwTH_Asignacion – Data Sources

NOMBRE	UBICACIÓN	OBSERVACIONES
192.168.30.252.DW_BabelSy s.sa	Server Oficinas Babel Software Latacunga	Se conecta a la fuente de datos.
pkgDwTH_Asignacion.txt	C:\log\ pkgDwTH_Asignacion .txt	Genera un archivo plano con datos de errores si es que se produjeran.

c. VARIABLES

Tabla No. 3.63: pkgDwTH_Asignacion – Variables

NOMBRE	ALCANCE	TIPO

d. CONTROL FLOW

El control de flujo se encuentra constituido únicamente por un Execute SQL Task, como se presenta en la Figura No. 3.43, el

mismo que llama a un procedimiento almacenado dentro del Data Warehouse llamado paLlenarAsignacion el cual se encarga de llenar la tabla TH_Asignacion tomando las claves de las tablas dimensionales.

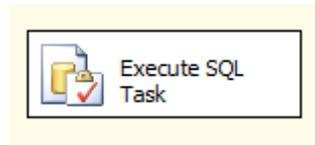


Figura No. 3.43 pkgDwTH_Asignacion – Control Flow

b. Actualización

Las políticas de Actualización que se han convenido con el usuario son las siguientes:

- La información se refrescará cada 15 días, los días domingos a las cero horas.
- Los datos de las tablas de dimensiones serán cargados únicamente aquellos registros que han sufrido actualizaciones en sus datos o que hayan sido insertados.
- Los datos de la tabla de dimensión “TD_FECHA” se cargarán de manera incremental teniendo en cuenta la fecha de su último registro. Para ello se puede utilizar el procedimiento almacenado: “paLlenarTiempo” que se encuentra en el DW y solicita dos parámetros de entrada que son: “@FecDesde” que indica la fecha desde donde va a iniciar la creación de las fechas y el parámetro “@Anios” que especifica cuantos años recorrerá el procedimiento desde la fecha. Al momento esta dimensión posee 12419 fechas disponibles desde 01/01/1999 hasta 31/12/2032.

Estas acciones se realizarán durante un periodo de prueba, para analizar cuál es el período de actualización más eficiente, basadas en el estudio de los cambios que se producen en los OLTP y que afectan al contenido del DW.

Los procesos ETL para la actualización del DW son los mismos que se utiliza en la carga inicial, debido a que están pensados y considerados con la posibilidad de obtener datos históricos y no necesitan ningún tratamiento especial para su actualización en las tablas dimensionales.

Sin embargo se debe tener en cuenta que un cambio provocado en las fechas de inicio y final de asignación del empleado de la tabla TProyectoEmpleado en el Stage, provocará que los registros en la tabla de hechos sean inconsistentes ya que en esta tabla se está almacenando un registro diario por y entre el rango de fechas especificado; de esta manera ya que no es aconsejable eliminar registros de las tablas de hechos, por la forma en que se presenta este requerimiento insertaremos registros con valores cuantificados negativos, deberán tenerse en cuenta las fechas de inserción de los registros antes de realizar dichas inserciones.

3.2. Cubo OLAP

El objetivo de la siguiente sección es describir la estructura del cubo de información que se brindará, como parte del alcance del presente proyecto, a Grupo Babel. Este cubo funcionará como herramienta

que brinde soporte a la toma de decisiones por parte del personal de la dirección anteriormente mencionada.

Para el cubo de información de Asignación de Tiempos se incluyen todas aquellas dimensiones y métricas que están creadas en el Data Warehouse.

3.2.1. Descripción de Dimensiones

Tabla No. 3.64: Dimensión Empleado

Dimensión	Empleado	
Definición	La funcionalidad de esta dimensión es mostrar la información cualitativa de los empleados.	
Estructura	Atributo	Descripción
	Id Empleado	Id del empleado
	Empleado	Nombre del empleado
Jerarquías	Nombre	Atributos

Tabla No. 3.65: Dimensión Empresa

Dimensión	Empresa	
Definición	La funcionalidad de esta dimensión es mostrar la información cualitativa de las empresas.	
Estructura	Atributo	Descripción
	Id Empresa	Id de la empresa
	Empresa	Nombre de la empresa
Jerarquías	Nombre	Atributos

Tabla No. 3.66: Dimensión Nacionalidad

Dimensión	Nacionalidad	
Definición	La funcionalidad de esta dimensión es mostrar la información cualitativa de las nacionalidades.	
Estructura	Atributo	Descripción
	Id Nacionalidad	Id de la nacionalidad
	Nacionalidad	Nombre de la nacionalidad
Jerarquías	Nombre	Atributos

Tabla No. 3.67: Dimensión Oficina

Dimensión	Oficina	
Definición	La funcionalidad de esta dimensión es mostrar la información cualitativa de las oficinas.	
Estructura	Atributo	Descripción
	Id Oficina	Id de la oficina
	Oficina	Nombre de la oficina
Jerarquías	Nombre	Atributos

Tabla No. 3.68: Dimensión Proyecto

Dimensión	Proyecto	
Definición	La funcionalidad de esta dimensión es mostrar la información cualitativa de los proyectos.	
Estructura	Atributo	Descripción
	Id Proyecto	Id del proyecto
	Proyecto	Nombre del proyecto

Jerarquías	Nombre	Atributos

Tabla No. 3.69: Dimensión Rol

Dimensión	Rol	
Definición	La funcionalidad de esta dimensión es mostrar la información cualitativa de los roles.	
Estructura	Atributo	Descripción
	Id Rol	Id del rol
	Rol	Nombre del rol
	Tipo de Rol	Descripción del tipo de rol
Jerarquías	Nombre	Atributos
	Roles	Tipo de Rol
		Rol

Tabla No. 3.70: Dimensión Tiempo

Dimensión	Tiempo
-----------	--------

Definición	La funcionalidad de esta dimensión es mostrar la información cualitativa de las fechas diarias de asignación.	
Estructura	Atributo	Descripción
	Id Tiempo	Id del tiempo
	Anio	Año
	Mes	Año - Mes
	Dia	Año – Mes – Día
Jerarquías	Nombre	Atributos
	Período	Anio
		Mes
		Dia

3.2.2. Descripción de Hechos

Tabla No. 3.71: Hechos Asignación

Hecho	Asignacion
Definición	La funcionalidad de este hecho es mostrar la información cuantitativa resultado de los procesos de asignación de proyectos al empleado.

	<p>Para aspectos de las funciones de agrupamiento se utilizaran las siguientes:</p> <p>Count = Para medidas que corresponden a cantidades.</p> <p>Sum= Para medidas que corresponden a montos.</p> <p>Distinct Count = Para las cantidades que correspondan a totales en la sección de Medidas Calculadas.</p>	
Medidas	Métrica	Descripción
	Porcentaje Diario	Porcentaje diario de asignación del empleado en un proyecto
Medidas Calculadas	Total de Porcentaje Asignación	Suma del porcentaje diario por mes.

3.2.3. Procedimiento para generar servicios OLAP con Analysis Services 2008 R2

Para desarrollar e implementar proyectos de Analysis Services vamos a utilizar “*Business Intelligence Development Studio*” que es un entorno de desarrollo basado en “*Microsoft Visual Studio 2008*” que se emplea para crear y modificar soluciones de Business Intelligence.

a. Definir una vista de origen de datos

Para diseñar la aplicación en primer lugar debemos crear un proyecto de Analysis Services en BI Development Studio. En este proyecto, definiremos todos los elementos de la solución empezando por una vista de origen de datos. El Proyecto será añadido a la solución que creamos anteriormente para los procesos de *"Integration Services"* llamada: *"ISP_BabelSys"*; este proyecto se llamará: *"ASP_CB_Asignacion"*.

Tras crear el proyecto empezamos definiendo un origen de datos para que utilice el proyecto. Al definir el origen de datos, se define la información de cadena de conexión que se utilizará para establecer la conexión con el origen de datos en nuestro caso con la base *"DW_BabelSys"* ubicada en el servidor de las oficinas de la empresa en Latacunga.

Una vez definido el origen de datos, definiremos una vista del origen de datos para el proyecto. Una vista de origen de datos es una sola vista unificada de metadatos de las tablas y vistas especificadas que el origen de datos define en el proyecto. Durante el desarrollo trabajaremos con metadatos, sin ninguna conexión abierta con ningún origen de datos subyacente.

Entonces seguiremos los siguientes pasos para crear una nueva vista de origen de datos:

1. En el Explorador de soluciones, haga clic con el botón secundario en Vistas de origen de datos, a continuación, haga clic en Nueva vista de origen de datos.
2. En la página Asistente para vistas de origen de datos, haga clic en Siguiente.

3. Aparece la página Seleccionar un origen de datos. En Orígenes de datos relacionales, el origen de datos “*DW Babel Sys*” aparece seleccionado. Haga clic en Siguiente.
4. Recordemos que la base DW_BabelSys no posee relaciones físicas, por lo que en esta ventana crearemos relaciones lógicas, así que utilizaremos los valores por defecto de la pantalla mediante los cuales buscará la igualdad en el nombre de los campos y los unificará como claves foráneas automáticamente, por lo que daremos clic en siguiente.
5. En la página Seleccionar tablas y vistas seleccionamos las tablas de la lista de objetos disponibles del origen de datos seleccionado. Puede filtrar esta lista para facilitar la selección. En la lista Objetos disponibles, seleccionamos los siguientes objetos:
 - TD_Empleado (dbo)
 - TD_Empresa (dbo)
 - TD_Nacionalidad (dbo)
 - TD_Oficina (dbo)
 - TD_Proyecto (dbo)
 - TD_Rol (dbo)
 - TD_Tiempo (dbo)
 - TH_Asignacion (dbo)

Hacemos clic en ➤ para agregar las tablas seleccionadas a la lista Objetos incluidos y hacemos clic en Siguiente.

6. En el campo Nombre, escribimos “DW Babel Sys”, a continuación, hacemos clic en Finalizar para definir la vista del origen de datos, podemos observarlo en la Figura No. 3.44:

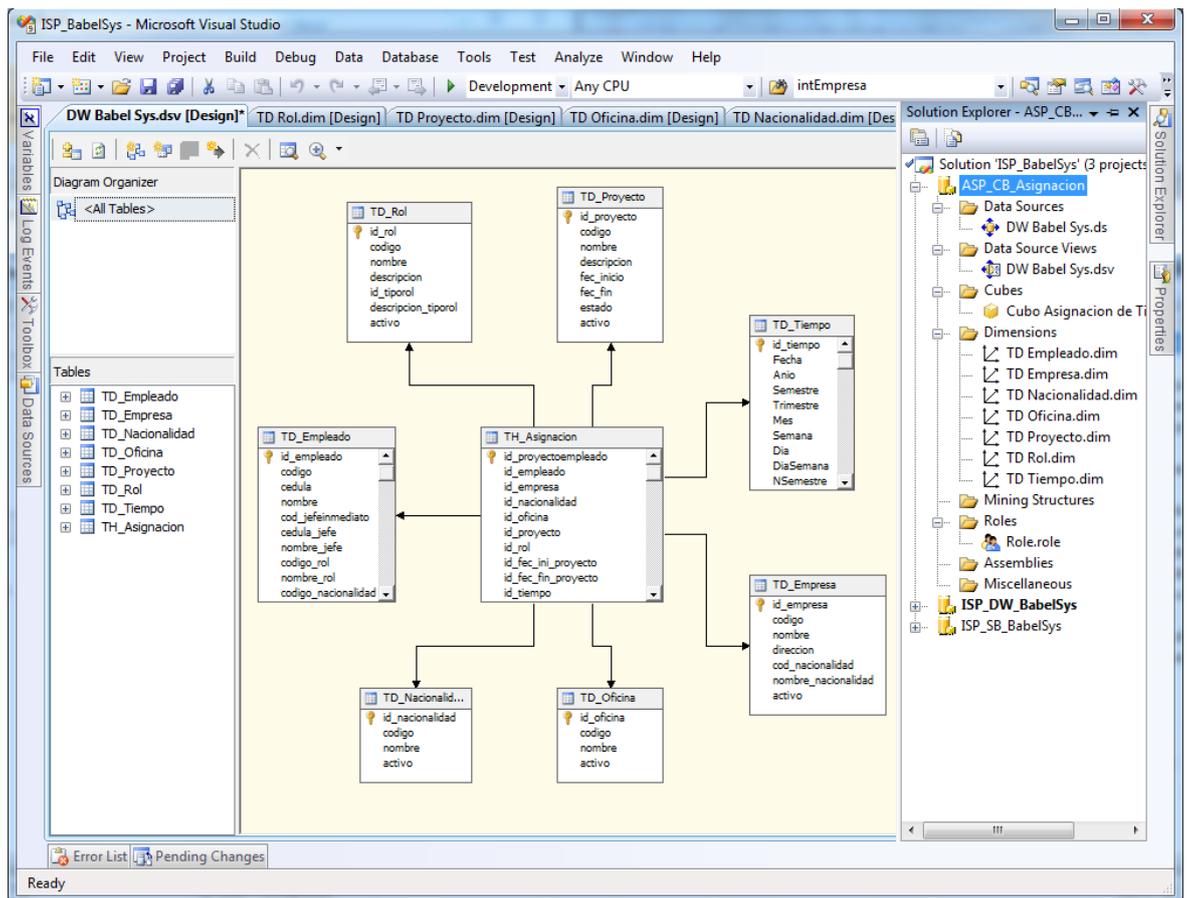


Figura No. 3.44: Vista de orígenes de datos DW Babel Sys

b. Crear una dimensión de una tabla existente

Se puede utilizar el Asistente de dimensiones para crear una dimensión a partir de una tabla existente. Para hacerlo, debe seleccionar la opción Usar una tabla existente en la página Seleccionar método de creación del asistente. Con esta opción, el asistente basa la estructura de la dimensión en las tablas de dimensiones, sus columnas y las posibles relaciones entre esas columnas en una vista de origen de datos existente, para nuestro caso será la vista de datos “*DW Babel Sys.dsv*”. El asistente prueba los datos en la tabla de origen y las tablas relacionadas. Usa estos

datos para definir columnas de atributos que se basan en las columnas de las tablas de dimensiones, así como para definir jerarquías de atributos (denominadas jerarquías definidas por el usuario). Tras utilizar el Asistente para dimensiones para crear su dimensión, puede usar el Diseñador de dimensiones para agregar, quitar o configurar atributos y jerarquías de la dimensión.

Si está utilizando una tabla existente para crear una dimensión, el Asistente para dimensiones lo guía a través de los siguientes pasos:

- *Especificar la información de origen:* Aquí seleccionamos nuestra vista de origen de datos que contiene la tabla en la que deseamos basar la dimensión. Y a continuación definimos las tablas de dimensión principal que está vinculada directamente a la tabla de hechos, en nuestro caso basaremos la creación de cada dimensión de acuerdo a la descripción previa de las dimensiones en la sección de Descripción de Dimensiones. El asistente selecciona automáticamente una columna de clave, aunque nosotros podemos agregar más columnas de la tabla en caso de necesitarlas. También podemos incluir de manera predeterminada una columna de nombre de miembro para presentar a los usuarios de manera predeterminada en la misma pantalla. Terminado esto damos clic en siguiente.
- *Seleccionar tablas relacionadas:* Si está generando una dimensión de copo de nieve, debe especificar las tablas relacionadas a partir de las cuales se definirán atributos adicionales en la página Seleccionar tablas relacionadas.
- *Seleccionar atributos de dimensión:* Para seleccionar los atributos que quiera incluir en la dimensión desde esas tablas. Todas las columnas subyacentes de todas estas tablas están

disponibles como posibles atributos de dimensión. El atributo de clave de dimensión debe estar seleccionado y activado para ser examinado. De forma predeterminada, el asistente establece el tipo de un atributo en Regular. Sin embargo, es posible que usted quiera asignar atributos concretos a un tipo de atributo diferente que represente mejor los datos. Al terminar de clic en siguiente

- *Completar el asistente:* Por ultimo de un nombre a la dimensión que desea crear y automáticamente pasara al diseñador en donde podrá crear las jerarquías necesarias para su dimensión si es que la requiere y/o brindar la edición correspondiente a cada dimensión.

Como paso adicional ocultaremos las claves de cada dimensión, seleccionando el atributo clave y en la pestaña de propiedades cambiamos el valor del atributo “AttributeHierarchyVisible” a “False” para que no sean visibles las aplicaciones clientes.

c. Definir el cubo “Cubo Asignacion de Tiempos”

Para la creación de nuestro cubo seguiremos los siguientes pasos del Asistente para cubos:

1. En el Explorador de soluciones, haga clic con el botón secundario en Cubos, a continuación, haga clic en Nuevo cubo.
2. En la página Asistente para cubos, haga clic en Siguiente.
3. En la página Seleccionar método de creación, compruebe que la opción Usar tablas existentes está seleccionada, a continuación, haga clic en Siguiente.

4. En la página Seleccionar tablas de grupo de medida, compruebe que la vista del origen de datos *DW Babel Sys* está seleccionada.
5. Vamos a escoger la tabla TH_Asignacion de la cual vamos a obtener las medidas o hechos basándonos en la descripción que tenemos en la sección de Descripción de Hechos previamente descrita, y hacemos clic en Siguiente
6. En la página Seleccionar medidas, vamos a seleccionar las medidas que pertenecerán a nuestro cubo de información, para nuestro caso únicamente necesitamos una medida o hecho que es: “Porcentaje Diario” el resto de medidas no las seleccionaremos. De forma predeterminada, el asistente selecciona como medidas todas las columnas numéricas de la tabla de hechos que no están vinculadas a dimensiones. No obstante, no todas las columnas son miembros reales. Existen algunos valores clave que vinculan la tabla de hechos con tablas de dimensiones que no se utilizan en la versión inicial de este cubo. Terminada esta parte damos clic en Siguiente
7. En la página Seleccionar dimensiones existentes, seleccione la dimensión todas las dimensiones que creamos anteriormente y hacemos clic en Siguiente.
8. En la página Finalización del asistente, cambiamos el nombre del cubo por “*Cubo Asignacion de Tiempos*”. En el panel de vista previa, puede ver el grupo de medida TH Asignacion y sus medidas. También puede ver todas las dimensiones.
9. Haga clic en Finalizar para completar el asistente.

En el Explorador de soluciones, en el proyecto *ASP_CB_Asignacion*, el cubo “*Cubo Asignación de Tiempos*” aparece en la carpeta *Cubos*, y las dimensiones de base de

datos aparecen en la carpeta *Dimensiones*. Adicionalmente, en el centro del entorno de desarrollo, el Diseñador de cubos muestra el cubo de información recién creado.

10. Guardamos todo.

3.3. Creación del reporte utilizando MS Excel 2007

Inmediatamente se va a crear nuevos reportes utilizando los componentes de Office, conectándose a la misma fuente de origen de datos (DW_BabelSys), la cual contiene las jerarquías, dimensiones y cubos creados en Analysis Services. Se utilizará un componente de office: Excel Pívor Table Reporte.

3.3.1. Crear Reporte de tabla dinámica de Excel

El método que utiliza Excel en Office 2007, es proveer los valores desde un cubo OLAP. Usando un cubo OLAP, un reporte de tabla dinámica puede comunicarse con el servicio de PivotTable para traer los valores desde el Analysis Server. Para crear un reporte de tabla dinámica basada en un cubo OLAP, se usa el “*Asistente de Conexión de Datos (Data Connection Wizard)*”; el cual crea dinámicamente la conexión al cubo OLAP mediante archivos de conexión de base de datos de Office (.odc); además usa la aplicación “*Microsoft Query*” para definir y crear el archivo de consulta OLAP y por último despliega la tabla dinámica en una hoja de Excel.

El procedimiento para generar las tablas dinámicas y poder conectar desde Excel un cubo OLAP de Analysis Server es el siguiente:

1. En la ficha **Datos**, en el grupo **Obtener datos externos**, se hace clic en **De Otras Fuentes** y luego en **Desde Analysis Services**.

Se mostrará el Asistente para la conexión de datos. Este asistente tiene tres pantallas:

Pantalla 1: Conectar con el servidor de la base de datos

2. En el paso 1, escriba el nombre del servidor OLAP en el cuadro de texto **Nombre del servidor**. El nombre del servidor para nuestro caso será la IP externa que tiene contratada la empresa: 186.42.253.98
3. En el paso 2, en **Credenciales de conexión**, vamos a hacer clic en **Utilizar el nombre de usuario y la contraseña siguientes** y a continuación, escribiremos el nombre del usuario y la contraseña en los cuadros de texto **Nombre de usuario** y **Contraseña** correspondientes. El usuario deberá ser el nombre del servidor \ el nombre del usuario así: "server252\usrolap".

Pantalla 2: Seleccionar base de datos y tabla

4. En **Seleccione la base de datos que contiene la información que desea**, seleccione una base de datos. La base deseada se llama ASP_CB_Asignacion.
5. Para conectar con un cubo específico de la base de datos, asegúrese de que la opción **Conectar con una tabla o a un cubo específico** está activada y seleccione un cubo de la lista. Para nuestro caso debe existir un único cubo llamado "*Cubo Asignación de Tiempos*".

Pantalla 3: Guardar archivo de conexión de datos y finalizar

6. De manera opcional, en el cuadro de texto **Nombre de archivo**, revise el nombre de archivo predeterminado. Haga clic en **Examinar** para cambiar la ubicación de archivo predeterminada

de **Mis archivos de origen de datos** o compruebe los nombres de archivo existentes.

7. De manera opcional, escriba una descripción del archivo, un nombre descriptivo y palabras de búsqueda comunes en los cuadros de texto **Descripción, Nombre descriptivo y Palabras clave de búsqueda**.
8. Para asegurarse de que siempre se utilice el archivo de conexión cuando se actualice la tabla dinámica, haga clic en **Intentar utilizar siempre este archivo para actualizar los datos**. De esta forma, se asegurará de que todos los libros que utilicen ese archivo de conexión utilicen siempre las actualizaciones realizadas en dicho archivo.
9. Haga clic en **Finalizar** para cerrar el Asistente para la conexión de datos y se mostrará el cuadro de diálogo **Importar Datos**.
10. En **Seleccione cómo desea ver estos datos en el libro**, se puede escoger una de las siguientes opciones:
 - Para crear solamente un informe de tabla dinámica, haga clic en **Informe de tabla dinámica**.
 - Para crear un informe de tabla dinámica y un informe de gráfico dinámico, haga clic en **Informe de gráfico y tabla dinámicos**.
 - Haga clic en **Crear sólo conexión** para almacenar la conexión seleccionada en el libro para su uso posterior. De esta forma, se asegurará de que las fórmulas que contienen funciones de cubo que ha creado utilicen la conexión y de que no se cree un informe de tabla dinámica.
11. En **¿Dónde desea situar los datos?**, escoja una celda desde donde desea que se ubique la tabla dinámica, o en su defecto puede crear una nueva hoja. Por defecto lo dejaremos en la celda A1 de la Hoja 1.

Ahora se ha creado un esqueleto para el Reporte de Tabla Dinámica o PivotTable llamada PivotTable1 y una Lista de Campos de Tabla Dinámica que contiene todas las dimensiones y jerarquías que pertenecen al cubo “Cubo Asignación de Tiempos”. A continuación lo podemos observar en la Figura No.3.45:

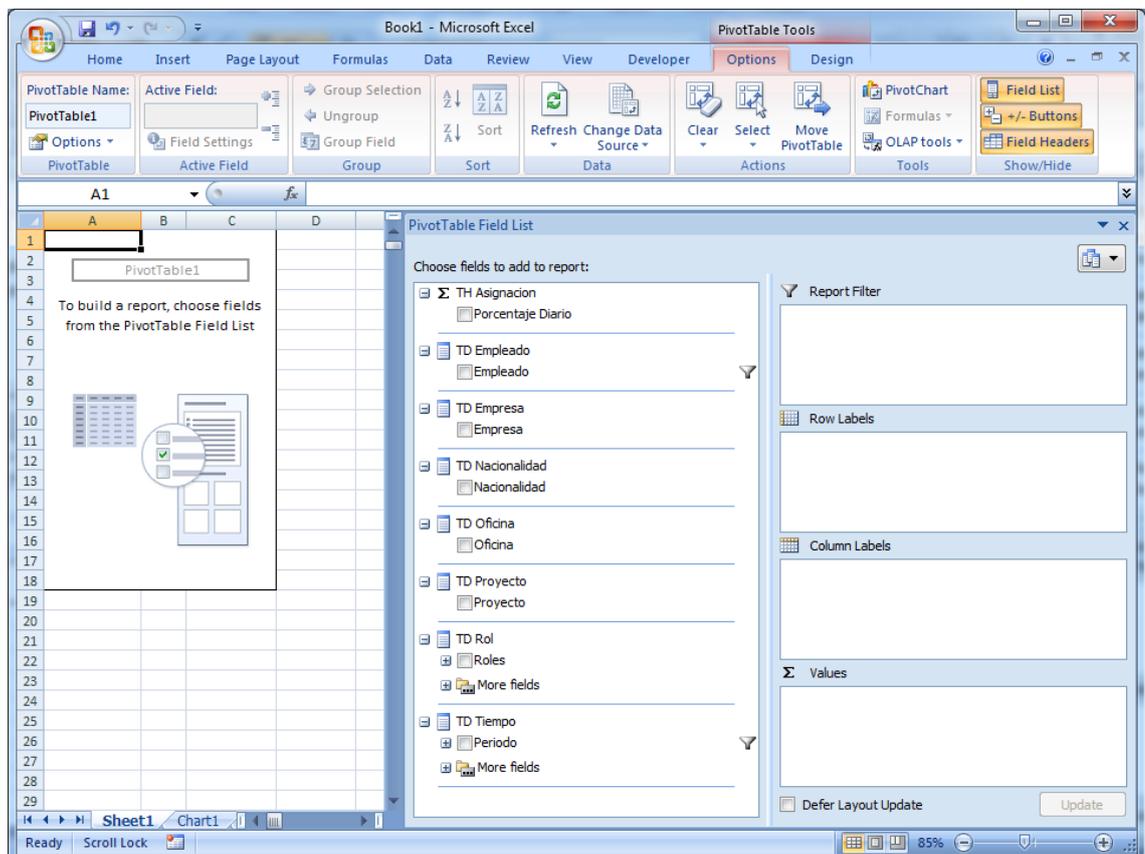


Figura No. 3.45: Esqueleto de Reporte de Tabla Dinámica

3.3.2. Explorar cubo “Cubo Asignación de Tiempos” usando el Reporte de Tabla Dinámicos

Un reporte de tabla Dinámica consiste de cuatro áreas: las primeras tres áreas son, área de página, área de columna y área de fila, que

contiene nombres de miembros y son llamados ejes. La cuarta área, el área de datos o valores, contiene los valores y medidas.

Se puede formar diferentes reportes de acuerdo a las necesidades. En nuestro caso se tiene en el área de fila las dimensiones: “Empleado” y “Proyecto”. En el área de Columnas tenemos la jerarquía “Periodo”. En el área de datos se tiene como medida o hecho “Porcentaje Diario”. Y por último en el área de página se tiene las dimensiones que servirán como filtros: “Empresa”, “Nacionalidad”, “Oficina” y la jerarquía “Roles”, como se presenta en la Figura No. 3.46:

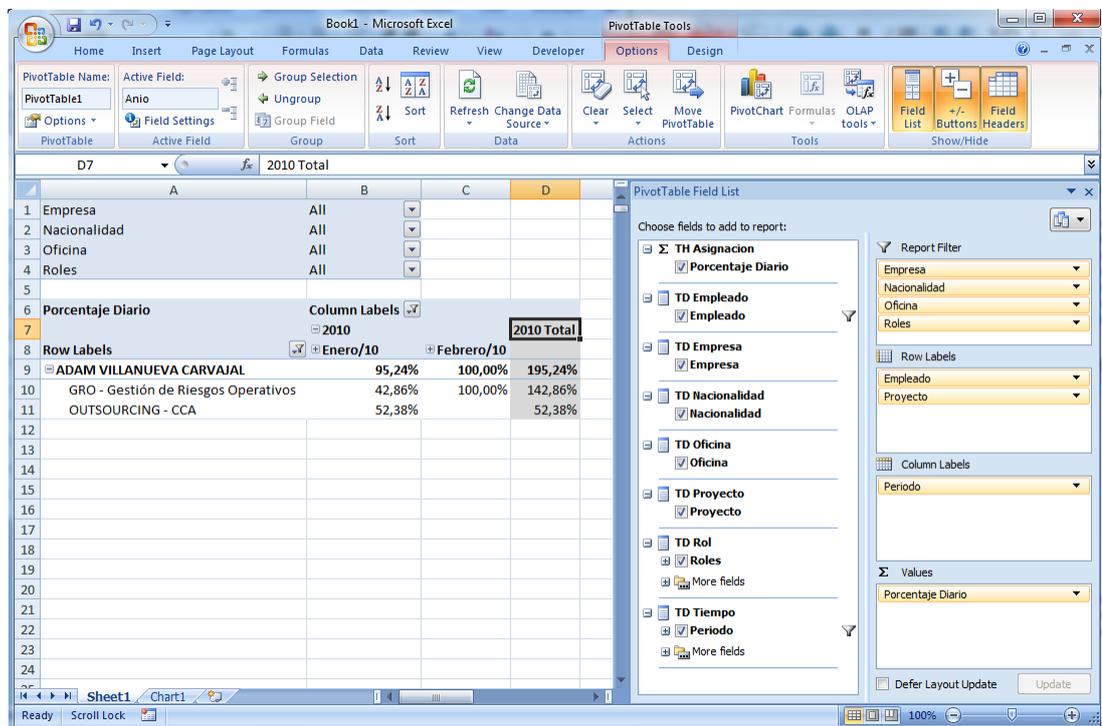


Figura No. 3.46: Reporte de Tabla Dinámica aplicando filtros en dos dimensiones

3.3.3. Reporte Gráfico Dinámico

Además del reporte tabular presentado anteriormente, Excel nos permite una presentación de datos gráfica. Una Gráfica Dinámica es ligada directamente al Reporte de Tabla Dinámica como se muestra en la Figura No. 3.47:

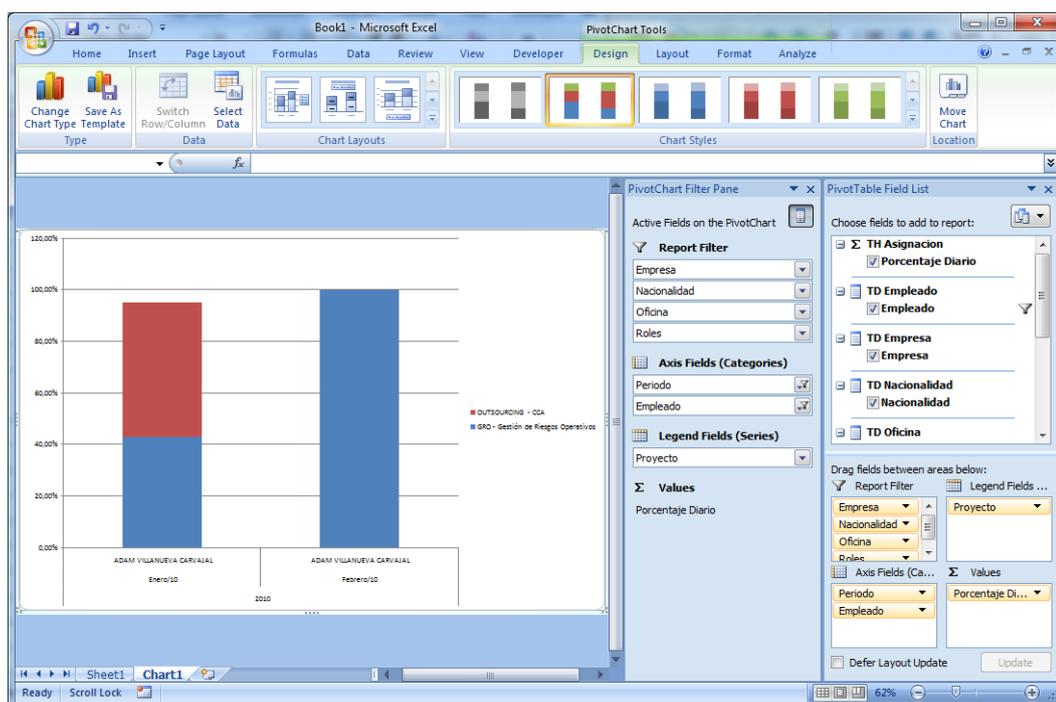


Figura No. 3.47: Gráfico de Reporte de Tabla Dinámica

Se puede visualizar en el gráfico los datos como el “Período” y el “Proyecto”; y como valores nuevamente tenemos al “Porcentaje Diario”, para tener una mejor visualización de los datos se puede apreciar que hemos movido la ubicación de los períodos y el reporte ha cambiado, esto sin tener que escribir ni una sola línea de SQL.

3.4. Implementación e integración del cubo de información y el reporte al sistema de planeación de tareas del personal técnico.

3.4.1. Implementar los paquetes de SQL Server Integration Services con los ETLs para la migración de datos

Para implementar paquetes de SQL Server Integration Services, puede utilizar el Asistente para la instalación de paquetes. Utilizando este asistente, puede implementar paquetes en una de las dos ubicaciones siguientes:

- En una instancia de SQL Server.
- En el sistema de archivos.

Sin embargo el primer paso para implementar paquetes es crear una utilidad de implementación para un proyecto de Integration Services. La utilidad de implementación es una carpeta que contiene los archivos necesarios para implementar los paquetes de un proyecto de Integration Services en un servidor distinto. La utilidad de implementación se crea en el equipo en el que se almacena el proyecto de Integration Services.

Puede crear una utilidad de implementación de paquetes para un proyecto de Integration Services configurando primero el proceso de generación de una utilidad de implementación y después, generando el proyecto. Al generar el proyecto, todos los paquetes y

configuraciones de paquetes del proyecto se incluyen automáticamente.

En nuestra solución ISP_BabelSys en cada uno de los proyectos ISP_DW_BabelSys e ISP_SB_BabelSys realizaremos lo siguiente:

1. En el explorador de soluciones, hacemos clic con el botón secundario en los proyectos mencionados en el párrafo anterior y damos clic en Propiedades para que aparezca el cuadro de diálogo “Páginas de propiedades de <nombre del proyecto>”.
2. En el nodo Propiedades de configuración del panel de la izquierda, hacemos clic en “*Deployment Utility (Utilitarios de Implementación)*”. Y colocamos en valor “*True*” a la propiedad *CreateDeploymentUtility*. La propiedad *DeploymentOutputPath* nos dará la ubicación, respecto al proyecto de Integration Services, de la utilidad de implementación y la propiedad *AllowConfigurationChange* nos especificará si se actualizarán las configuraciones durante la implementación. Dejar a estas dos últimas con sus valores por defecto.
3. Hacer clic en Aceptar.
4. En el explorador de soluciones, hacemos clic con el botón secundario en los proyectos mencionados en el párrafo anterior y damos clic “*Build (Generar)*”.

Al generar un proyecto de Integration Services, se crea un de manifiesto <nombre del proyecto>.SSISDeploymentManifest.xml, junto con las copias de los paquetes del proyecto y las dependencias del paquete, y se agrega a la carpeta bin\Deployment del proyecto o a la ubicación especificada en la propiedad DeploymentOutputPath. El archivo de manifiesto muestra los paquetes, las configuraciones de paquetes y todos los demás archivos del proyecto.

Para instalar los paquetes, puede utilizar el Asistente para la instalación de paquetes. Para iniciar el asistente, haga doble clic en el archivo de la utilidad de implementación una vez copiada la carpeta de implementación en el servidor en las oficinas de Latacunga. El archivo recibe el nombre *<nombre del proyecto>.SSISDeploymentManifest*, y se puede buscar en la carpeta de implementación del equipo de destino.

El Asistente para la instalación de paquetes le guía paso a paso durante la instalación de paquetes en el sistema de archivos o en SQL Server, sin embargo para nuestro caso instalaremos nuestros paquetes dentro de una instancia de SQL Server. Puede configurar la instalación de las maneras siguientes:

5. Eligiendo el tipo de ubicación: Escogeremos la opción de SQL Server Deployment. Opcional podemos marcar la casilla de validación paquetes después de la instalación. Y damos clic en Siguiente.
6. La ubicación para instalar los paquetes: Donde escogemos el servidor local, escogemos la forma de autenticación que de preferencia sea el usuario "sa" y además le damos un Path en donde se implementarán los paquetes. Damos clic en Siguiente.
7. En la carpeta de instalación dejamos por defecto la ruta que el asistente nos brinda y damos clic en Siguiente.
8. Confirmamos la instalación dando clic en Siguiente y el asistente validará los paquetes una vez instalados en el servidor de destino si fuera el caso.
9. Dar clic en Finalizar.

3.4.2. Implementar el Cubo de Información

Para ver los datos de dimensión y de cubo de los objetos del cubo “Cubo Asignacion de Tiempos” de nuestro proyecto, debemos implementar el proyecto en una instancia determinada de Analysis Services y luego procesar el cubo y sus dimensiones. Al implementar un proyecto de Analysis Services se crean y definen objetos en una instancia de Analysis Services. Cuando se procesan los objetos en una instancia de Analysis Services, se copian los datos de los orígenes de datos subyacentes en los objetos del cubo.

En este punto del proceso de implementación, generalmente se implementa el cubo en una instancia de Analysis Services en un servidor de implementación que para nuestro caso será el mismo servidor donde se encuentran las bases de datos SG_BabelSys y DW_BabelSys en las oficinas de la empresa en Latacunga.

Para implementar el proyecto de Analysis Services

1. En el explorador de soluciones, hacemos clic con el botón secundario en el proyecto “ASP_CB_Asignacion” y damos clic en Propiedades para que aparezca el cuadro de diálogo “Páginas de propiedades de ASP_CB_Asignacion”.
2. En el nodo Propiedades de configuración del panel de la izquierda, hacemos clic en “Deployment (Implementación)”. En “*Server*” introducimos la dirección del servidor de en las oficinas de Babel SW de Latacunga y en “*Database*” colocaremos: “ASP_CB_Asignacion” que será el nombre de la base de datos en donde se hará el deploy del proyecto.

3. Damos clic en Aceptar.
4. En el explorador de soluciones, hacemos clic con el botón secundario en el proyecto “*ASP_CB_Asignacion*” y damos clic en “Deploy (Implementar)”.

Business Intelligence Development Studio genera e implementa el proyecto en la instancia especificada de Analysis Services mediante una secuencia de comandos de implementación. El progreso de la implementación se muestra en dos ventanas: la ventana Resultados y la ventana Progreso de la implementación – *ASP_CB_Asignacion*. Para abrir la ventana Resultados, si es necesario, haga clic en Resultados en el menú Ver. La ventana Resultados muestra el progreso global de la implementación. La ventana Progreso de la implementación – *ASP_CB_Asignacion* muestra los detalles de cada paso realizado durante la implementación.

5. En el explorador de soluciones, hacemos clic con el botón secundario en el proyecto “*ASP_CB_Asignacion*” y damos clic en “Process (Procesar)”.
6. Aparecerá la ventana Procesar Base de Datos – *ASP_CB_Asignacion* en donde tendremos un lista objetos que se procesarán en donde se puede observar que como objeto se encuentra la base de datos para ser procesada completamente y un Sumario de Opciones del proceso Batch.

El procesamiento es el paso, o una serie de pasos, que pueblan objetos de Analysis Services con datos de fuentes de datos relacionales. El procesamiento varía dependiendo del tipo de objeto y la selección de opciones de procesamiento.

Cuando se procesa un cubo, las agregaciones diseñadas para el cubo son calculadas y el cubo se carga con las agregaciones calculadas y los datos. El procesamiento de un cubo implica la lectura de las tablas de dimensiones para poblar los niveles con

los miembros de los datos reales, la lectura de la tabla de hechos, calculando las agregaciones especificadas, y almacenar los resultados en el cubo. Después que el cubo se procesa, los usuarios pueden consultar.

7. Dar clic en “*Run (Correr o Procesar)*” y aparecerá la pantalla de Progreso de Proceso la misma que indicará el avance del proceso de los objetos del cubo. Al finalizar el proceso podemos solicitar un Reproceso, ver los detalles o realizar una copia de los resultados. Por último damos clic en Cerrar en las sub pantallas abiertas.

3.4.3. Automatizar ejecución de paquetes de Integration Services y procesamiento de la base de datos de Analysis Services

El servicio SQL Server Agent tiene como funcionalidad principal ejecutar tareas programadas, en base a criterios de administración remota de un servidor, donde los administradores aprovechan para llevar a cabo tareas en el momento en que la Base de Datos se encuentra en un estado de poco uso, entre estas tareas están los Trabajos (Jobs).

Para crear un trabajo:

1. En el Explorador de Objetos, conectarse a una instancia de SQL Server Database Engine y luego expandir la instancia.
2. Expandir SQL Server Agent.
3. Hacer clic derecho en la carpeta “*Jobs*”, y luego clic en “*New Job*”.

4. En la página General, en la caja de nombre, tipiar un nombre para el trabajo, en nuestro caso usaremos: *“JobAsignacionTiempos”*.
5. Opcional puede dejar marcado la casilla *“Enabled”* si se desea que el trabajo corra inmediatamente después de su creación.
6. En la caja Descripción, ingresar una descripción de que los que hace el trabajo. En un máximo de 512 caracteres.

A continuación vamos a definir los pasos para crear una tarea para ejecutar un paquete de Integration Services el cuál replicaremos para todos los paquetes que tengamos. Y también para realizar el proceso automático de la base de datos en Analysis Services.

Cuando creamos un trabajo de SQL Server Agent para correr paquetes, debemos crear un “step” separado por cada paquete que queramos correr. El trabajo puede ser asociado con una o más ejecuciones programadas o pueden ser corridas de forma manual.

Para correr un paquete desde un trabajo debemos crear una cuenta proxy que tiene las credenciales que el trabajo usará para ejecutar el paquete. Una cuenta proxy define un contexto de seguridad en el cual un paso de trabajo puede correr:

1. En el explorador de objetos, expandir Seguridad, hacer clic derecho sobre Credenciales, y luego clic en Nueva Credencial.
2. En la ventana nueva credencial, dar un nombre a la credencial para nuestro caso será *“SSIS_Credencial”*.

3. En la caja de identidad, dar el nombre de la cuenta usada para conexiones de salida, preferentemente será el usuario de Windows.
4. En contraseña y confirmar contraseña, escriba la contraseña de la cuenta especificada en la caja de identidad. Al ser el usuario de Windows, debemos escribir la contraseña de este usuario para el ingreso del sistema.
5. Clic Ok.
6. En el explorador de objetos, expandir la instancia.
7. Expandir SQL Server Agent.
8. Hacer clic derecho en “*Proxies*” y luego clic en nuevo proxy.
9. En la página General de la ventana nueva cuenta proxy, especificar el nombre del proxy para nuestro caso será “*SSIS_Proxy*”, el nombre de la credencial que es “*SSIS_Credential*”, y la descripción del nuevo proxy.
10. Marcar los apropiados sub sistemas para este proxy, en nuestro caso solo necesitamos marcar “*SQL Server Integration Services Package*”.
11. Clic Ok.

Este proxy funcionará para crear todos los “*Job Steps*” que ejecuten un paquete de *Integration Services*.

7. Dar clic en “*Steps*” en el panel izquierdo. Aquí se podrán agregar “*Job Steps*” o las tareas que cumplirá el trabajo.
8. Dar clic en nuevo.

9. Dar un nombre, en la lista de tipo, seleccionar un tipo de paso de trabajo. En nuestro caso será “*SQL Server Integration Services Package*” para ejecutar paquetes.
10. En la lista Correr como, vamos a seleccionar la cuenta proxy que creamos previamente “*SSIS_Proxy*”.
11. En la pestaña General, seleccionar la fuente del paquete: “*SSIS Package Store*”, seleccionar el servidor local, en el modo de logeo usaremos la autenticación de Windows, y por último buscaremos el paquete que implementamos anteriormente.
12. Clic en Ok.

Repetiremos simultáneamente desde el paso número 8 para crear las tareas para ejecutar todos los paquetes de Integration Services creados en este orden:

1. pkgSGTipoRol
2. pkgSGRol
3. pkgSGNacionalidad
4. pkgSGOficina
5. pkgSGEmpresa
6. pkgSGEstado
7. pkgSGEmpleado
8. pkgSGProyecto
9. pkgSGProyectoEmpleado
10. pkgSGProyectoEmpresa
11. pkgDwTD_Rol
12. pkgDwTD_Nacionalidad
13. pkgDwTD_Proyecto
14. pkgDwTD_Oficina
15. pkgDwTD_Empresa

- 16.pkgDwTD_Empleado
- 17.pkgDwTH_Asignacion

Una vez terminado de implementar las tareas para cada paquete, seguimos con la automatización del proceso de la base de datos de Analysis Services.

- 13.Repetimos los pasos 7 y 8
- 14.Dar un nombre, en la lista de tipo, seleccionar un tipo de paso de trabajo. En nuestro caso será “*SQL Server Analysis Services Command*” para ejecutar una sentencia.
La sentencia debe ser un XML para Analysis Services Execute method (XMLA) que es un estándar que soporta accesos a datos a fuentes de datos que residen en la Web. Este lenguaje nos permite procesar objetos de Analysis Services tal cual si fuera un script.
- 15.En la lista Correr como, vamos a seleccionar un proxy que ha estado definido para usar el sub sistema “*Analysis Services Command*”. Por lo que escogeremos el proxy que viene por defecto ya que nos servirá correctamente para nuestro propósito: “*SQL Agent Service Account*”.
- 16.Escoger el servidor de Analysis Services donde correrá el proceso o escribir su nombre, para nuestro caso debemos buscar el servidor local.
- 17.En el cuadro de comando, vamos a insertar nuestro código XMLA que permitirá realizar el procesamiento de los objetos del Analysis Services.
Este código XMLA puede obtenerse así:
 - a. Abrir una instancia de SQL Server Analysis Services
 - b. Expandir la instancia

- c. Expandir el nodo base de datos
 - d. Hacemos clic derecho sobre la base “ASP_CB_Asignacion” y hacemos clic en procesar.
 - e. En el cuadro de diálogo “Procesar base de datos – ASP_CB_Asignacion” verificamos que el tipo de procesamiento de la base de datos sea completo y damos clic en el botón “Script” y nos dará un script de código XMLA para ejecutar un proceso completo en todos los objetos que contenga la base de datos ASP_CB_Asignacion.
 - f. Clic en Cancelar.
18. Al ser el último proceso que realizaremos debemos cambiar la acción que realizará el trabajo al terminar con éxito este paso. Entonces damos clic en “Advanced” y en la lista “En acción exitosa” escogemos la acción “Terminar el trabajo reportando éxito (Quit the job reporting success)”.
19. Clic en Ok.
20. En la opción del panel izquierdo “Programado (Schedules)” damos clic en nuevo, y le damos un nombre al programa, escogemos la frecuencia en la que se va a ejecutar ese trabajo, tomando en cuenta que para iniciar el período de prueba lo programaremos para que se repita cada 15 días los días domingo a las cero horas. Y lo vamos a dejar con fecha de iniciación que por defecto es la fecha de creación del trabajo y sin fecha de finalización. Y damos clic en Ok.
21. Para crear la tarea damos clic en Ok.

3.4.4. Implementar el reporte creado en MS Excel 2007

Si bien en una sección anterior explicamos la creación del reporte debemos tener en cuenta que la conexión creada en primer lugar no almacena la contraseña de conexión salvo que lo solicitemos y en cuyo caso quedaría expuesto a que otro usuario pueda visualizarlo, además de ello debe poseer la conexión física, el “.odc”, para poder conectarse al cubo OLAP ya que sin ello únicamente se convertiría en una tabla estática sin conexión.

Tomando en cuenta estos puntos la solución que se acordó con el usuario final sería de implementar una clave de lectura al archivo Excel el cual permitirá que guardemos el password del usuario de conexión.

Para evitar la utilización de la conexión física, provocaremos que la conexión sea almacenada de manera interna en el libro de Excel. Utilizando la opción conexiones del grupo conexión en la ficha datos, podemos añadir conexiones al *workbook* y de esta manera evitaríamos la utilización de la conexión física, claro está que primero debemos crear la conexión física y después añadirla a la lista de conexiones del *workbook*.

Una vez almacenada dicha conexión, en vez de crear una conexión usaremos la opción conexión existente en el grupo Obtener datos externos en la ficha datos. Y escogeremos la conexión que se encuentre en el grupo de “*Conexiones en este Libro (Connections in this Workbook)*”. Y continuaremos con los pasos de elaboración del

reporte. Al finalizar guardaremos todo y estableceremos la clave de lectura en el documento mediante el cuadro de diálogo de Guardar como, en la opción de Herramientas en Opciones Generales.

De esta manera lo relevante del reporte es que no se encuentra ligado a una instalación y puede ser llevado en un dispositivo de almacenamiento y ser ocupado en cualquier computadora con acceso a internet. Incluso puede ser compartido siempre y cuando se comparta la contraseña de lectura en cuyo caso la utilización de terceros de este reporte queda en total responsabilidad del usuario.

3.5. Consideraciones finales

1. El período de prueba será de un mes, en ese período se establecerá si el tiempo de refrescado de los datos es conveniente de acuerdo a las necesidades del usuario, en caso de ser cambiado existirá una persona encargada (IT Guy) de coordinar el cambio deseado por el usuario, si el período es demasiado corto o demasiado separado.
2. Se entregarán aparte del reporte con clave de seguridad, el archivo de conexión “.odc” y un reporte sin la clave de seguridad mediante los cuales el usuario puede realizar nuevos reportes si así lo deseara y los mismos servirán como respaldos de la conexión.
3. Una parte de la configuración obliga que el puerto de conexión del servicio de Analysis Services se cambie de aleatorio “0” al “2383” por cuestiones de seguridad de la empresa.

4. Es muy importante que los ETLs sean lo más eficientes posibles, ya que una vez que se tenga un gran volumen de datos, el espacio en los discos se volverá fundamental y los tiempos incurridos en el procesamiento y acceso a la información serán esenciales, y más aún si el DW es considerado o tomado como un sistema de misión crítica.
5. Una de las maneras más eficientes para mejorar el performance del DW es prestando especial atención a los tipos de datos utilizados, por ejemplo, para valores enteros pequeños conviene utilizar *tinyint* o *smallint*, con el fin de no asignar tamaños de datos mayores a los necesarios. Esto toma vital importancia cuando se aplica en las claves primarias, debido a que formarán parte de la tabla de hechos que es la que contiene el volumen del almacén de datos.

CAPÍTULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

1. Al implementar un Cubo de Información es fundamental que los usuarios del mismo participen activamente durante todo su desarrollo, debido a que son ellos los que conocen en profundidad su negocio y saben cuáles son los resultados que se desean obtener. Además, es precisamente en base a la utilización que se le dé, que el almacén de datos madurará y se adaptará a las situaciones cambiantes por las que atraviese la empresa. Además un cubo de información introduce nuevos conceptos tecnológicos y de inteligencia de negocios, lo cual requiere que se aprenda nuevas técnicas, herramientas, métodos, destrezas, formas de trabajar, etc. Entre ellas al utilizar SQL Server tenemos Integration Services, Analysis Services, SQL Services Agent, etc.
2. Un sistema de reportería integrado a un cubo OLAP sin duda es mucho más versátil y dinámico que una reportería integrada a un sistema OLTP en vista de que con un cubo de información los tiempos de respuesta son muchos más rápidos y en volúmenes grandes, además de mencionar que se pueden realizar cambios en los ejes de dichos reportes, hecho que no puede ser realizado en una reportería estática; y la mayor ventaja entre ellos es que un solo cubo de información puede proporcionarnos

tantos reportes requiramos sin la necesidad de utilizar sentencias SQL.

3. La integración de un cubo de información y reportería, facilita el funcionamiento de las aplicaciones de los sistemas de apoyo a la decisión tales como informes de tendencia o informes de excepción que muestran los resultados reales frente a los objetivos planteados a priori.

4.2. Recomendaciones

1. Es recomendable que el 70% del tiempo total dedicado al proyecto se invierta en definir el problema y en preparar la tabla de datos, por lo que consecuentemente la mayor parte del trabajo consista en la definición y análisis del mismo. Además de ello es recomendable mejorar el funcionamiento del cubo incrementando KPI's.
2. El sistema de reportería puede ser mejorado utilizando una lista de tabla dinámica de Office (Office PivotTable List) en vista de que esta herramienta fue creada con los cubos OLAP en mente, haciendo de ella una herramienta más flexible para trabajar con cubos OLAP. O a su vez podemos generar Reportes en Web usando el control "CellSetGrid" que es un control del lado del servidor bajo ASP.NET.
3. Para tener una integración mucho más a la par entre el sistema transaccional y el sistema OLAP se recomienda que las aplicaciones finales en ambas instancias sean realizadas en ambiente web. O a su vez integrar el reporte de Excel al sitio SharePoint recientemente implementado en la empresa.

ANEXOS

ANEXO A: Script para crear el Store Procedure: “paLlenarAsignacion”

```

USE [DW_BabelSys]
GO
/***** Object:  StoredProcedure [dbo].[paLlenarAsignacion]
Script Date: 02/23/2012 12:15:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Dgallardo
-- Create date:    05/09/2011
-- Description:    Llena la tabla TH_Asignacion
-- =====
ALTER PROCEDURE [dbo].[paLlenarAsignacion]

AS
BEGIN

declare @fec_insert datetime = getdate();
declare @fec_update datetime = getdate();

select * into #tmp_asignacion from TH_Asignacion where 1 = 0;

insert into #tmp_asignacion
    select tpe.id_proyectoempleado, em.id_empleado,
           tmr.id_empresa,
           tnc.id_nacionalidad,
           tof.id_oficina,
           tpr.id_proyecto,
           trl.id_rol,
           tp1.id_tiempo id_fec_ini_proyecto,
           tp2.id_tiempo id_fec_fin_proyecto,
           tp3.id_tiempo id_tiempo,
           tpe.asignacionempleado,

dw_babelsys.dbo.DiasRangoFechas (tp1.Fecha, tp2.Fecha, tp3.Fecha)
NumDiasLaborablesMes,

cast((dw_babelsys.dbo.DiasRangoFechas (tpe.fec_ini_asignacion, tpe.fec_fin_asignacion, tp3.fecha)*tpe.asignacionempleado) as
decimal)/100 EquivalenteDiasLaborablesMes,
           -- Los porcentajes son representados por numeros
inferiores a 1

((cast((dw_babelsys.dbo.DiasRangoFechas (tpe.fec_ini_asignacion, tpe.fec_fin_asignacion, tp3.fecha)*tpe.asignacionempleado) as
decimal)/100)/dw_babelsys.dbo.DiasEnMes (tp3.fecha))
           PorcentajeMensual,

(((cast((dw_babelsys.dbo.DiasRangoFechas (tpe.fec_ini_asignacion, tpe.fec_fin_asignacion, tp3.fecha)*tpe.asignacionempleado) as
decimal)/100)/dw_babelsys.dbo.DiasEnMes (tp3.fecha)))/cast(dw_babelsys.dbo.DiasRangoFechas (tpe.fec_ini_asignacion, tpe.fec_fin_asignacion, tp3.fecha) as decimal)
           PorcentajeDiario,
           @fec_insert,

```

```

        @fec_update
    from  sb_babelsys.dbo.TEmpleado tem
    join  dw_babelsys.dbo.td_empleado em on tem.cod_empleado =
em.codigo
        and cast(tem.nom_empleado as varchar(200)) = em.nombre
        and tem.cod_estado = em.cod_estado
    join  sb_babelsys.dbo.TOficina ofc on ofc.cod_oficina =
em.codigo_oficina
    join  dw_babelsys.dbo.td_Oficina tof on tof.codigo =
ofc.cod_oficina
        and tof.nombre = ofc.nom_oficina
    join  sb_babelsys.dbo.TProyectoEmpleado tpe on tpe.cod_empleado
= tem.cod_empleado

        and tpe.activo = 1
    join  sb_babelsys.dbo.TRol rol on rol.ide_rol = tpe.ide_rol
    join  dw_babelsys.dbo.td_Rol trl on trl.codigo = rol.ide_rol
        and trl.nombre = rol.nom_rol
        and trl.id_tiporol = rol.id_tiporol
    join  sb_babelsys.dbo.TProyecto pro on pro.ide_proyecto =
tpe.id_proyecto
    join  dw_babelsys.dbo.td_Proyecto tpr on pro.ide_proyecto =
tpr.id_proyecto
    join  dw_babelsys.dbo.TD_Tiempo tp1 on tp1.Fecha =
tpe.fec_ini_asignacion
    join  dw_babelsys.dbo.TD_Tiempo tp2 on tp2.Fecha =
tpe.fec_fin_asignacion
    join  sb_babelsys.dbo.TProyectoEmpresa tpm on tpm.ide_proyecto =
pro.ide_proyecto
    join  sb_babelsys.dbo.TEmpresa emp on emp.ide_empresa =
tpm.ide_empresa
    join  dw_babelsys.dbo.td_Empresa tmr on tmr.codigo =
emp.ide_empresa
        and tmr.nombre = emp.nom_empresa
    join  sb_babelsys.dbo.TNacionalidad nac on
nac.codnacionalidad = emp.codnacionalidad
    join  dw_babelsys.dbo.td_Nacionalidad tnc on tnc.codigo =
emp.codnacionalidad
    join  dw_babelsys.dbo.TD_Tiempo tp3 on tp3.Fecha >=
tpe.fec_ini_asignacion
        and tp3.Fecha <= tpe.fec_fin_asignacion
        and tp3.ENDiaSemana != 'Saturday' and
tp3.ENDiaSemana != 'Sunday'

```

```
-- Actualiza los ultimos registros que no sean negativos
```

```

update ta set
ta.id_empleado = tm.id_empleado
,ta.id_empresa = tm.id_empresa
,ta.id_nacionalidad = tm.id_nacionalidad
,ta.id_proyecto = tm.id_proyecto
,ta.id_rol = tm.id_rol
,ta.AsignacionEmpleado = tm.AsignacionEmpleado
,ta.NumDiasLaborablesMes = tm.NumDiasLaborablesMes
,ta.EquivalenteDiasLaborablesMes = tm.EquivalenteDiasLaborablesMes
,ta.PorcentajeMensual = tm.PorcentajeMensual
,ta.PorcentajeDiario = tm.PorcentajeDiario

```

```

,ta.fec_update = tm.fec_update
from TH_Asignacion ta
inner join #tmp_asignacion tm on tm.id_proyectoempleado =
ta.id_proyectoempleado
and ta.id_tiempo = tm.id_tiempo
and (
ta.id_empleado != tm.id_empleado
or ta.id_empresa != tm.id_empresa
or ta.id_nacionalidad != tm.id_nacionalidad
or ta.id_proyecto != tm.id_proyecto
or ta.id_rol != tm.id_rol
or ta.AsignacionEmpleado != tm.AsignacionEmpleado
)
and ta.fec_insert =
(select MAX(fec_insert)
 from TH_Asignacion
 where id_proyectoempleado = ta.id_proyectoempleado
 and id_tiempo = ta.id_tiempo)
and ta.PorcentajeDiario > 0

-- Actualiza las fechas de todos los registros ultimos con valor
positivo
update ta set
ta.id_fec_fin_proyecto = tm.id_fec_fin_proyecto
,ta.id_fec_ini_proyecto = tm.id_fec_ini_proyecto
,ta.fec_update = tm.fec_update
from TH_Asignacion ta
inner join #tmp_asignacion tm on tm.id_proyectoempleado =
ta.id_proyectoempleado
--and ta.id_tiempo = tm.id_tiempo
and (
ta.id_fec_ini_proyecto != tm.id_fec_ini_proyecto
or ta.id_fec_fin_proyecto != tm.id_fec_fin_proyecto
)
and ta.fec_insert =
(select MAX(fec_insert)
 from TH_Asignacion
 where id_proyectoempleado = ta.id_proyectoempleado
 and id_tiempo = ta.id_tiempo)
and ta.PorcentajeDiario > 0

-- Inserta en valores negativos aquellos registros que esten fuera
del rango de fechas
insert into TH_Asignacion
select ta.id_proyectoempleado
,ta.id_empleado
,ta.id_empresa
,ta.id_nacionalidad
,ta.id_oficina
,ta.id_proyecto
,ta.id_rol
,ta.id_fec_ini_proyecto
,ta.id_fec_fin_proyecto
,ta.id_tiempo
,ta.AsignacionEmpleado * -1
,ta.NumDiasLaborablesMes

```

```

        ,ta.EquivalenteDiasLaborablesMes
        ,ta.PorcentajeMensual * -1
        ,ta.PorcentajeDiario * -1
        ,@fec_insert
        ,@fec_update
    from TH_Asignacion ta
    where ta.fec_update = @fec_update
        and (ta.id_tiempo > ta.id_fec_fin_proyecto
            or ta.id_tiempo < ta.id_fec_ini_proyecto);

-- Inserta los registros nuevos
insert into TH_Asignacion
select ta.*
from #tmp_asignacion ta
left join TH_Asignacion th on th.id_proyectoempleado =
ta.id_proyectoempleado
                                and th.id_tiempo =
ta.id_tiempo
where isnull(th.id_proyectoempleado,0) = 0;

-- Inserta los registros que se encuentran dentro del rango de
fechas y
-- no fueron insertados porque ya existian con registros negativos
insert into TH_Asignacion
select ta.*
from #tmp_asignacion ta
inner join TH_Asignacion th on th.id_proyectoempleado =
ta.id_proyectoempleado
                                and th.id_tiempo =
ta.id_tiempo
                                and th.fec_insert =
(select MAX(fec_insert)
 from TH_Asignacion
 where id_proyectoempleado =
th.id_proyectoempleado
                                and id_tiempo = th.id_tiempo)
                                and th.PorcentajeDiario < 0
;

drop table #tmp_asignacion;

update TD_Empleado set activo = 0;
update TD_Empresa set activo = 0;
update TD_Nacionalidad set activo = 0;
update TD_Oficina set activo = 0;
update TD_Proyecto set activo = 0;
update TD_Rol set activo = 0;
update SB_BabelSys.dbo.TProyectoEmpleado set activo = 0;

END

```

ANEXO B: Script para crear el Store Procedure: "paLlenarTiempo"

```

USE [DW_BabelSys]
GO
/***** Object: StoredProcedure [dbo].[paLlenarTiempo]      Script
Date: 02/23/2012 12:15:38 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:          Diego Gallardo
-- Create date:    06/09/2011
-- Description:    Llena la tabla de tiempo
-- =====
ALTER PROCEDURE [dbo].[paLlenarTiempo]
    -- Add the parameters for the stored procedure here
    @FecDesde varchar(8),
    @Anios int
AS
BEGIN
/*Script de carga*/

DECLARE @FechaDesde as smalldatetime, @FechaHasta as smalldatetime
DECLARE @FechaAAAAMMDD int
DECLARE @Año as smallint, @Semestre char(2), @Trimestre char(2),
@Mes smallint
DECLARE @Semana smallint, @Dia smallint, @DiaSemana smallint
DECLARE @NSemestre char(7), @NTrimestre char(7), @ENMes char(18)
DECLARE @ENMes31 char(6)
DECLARE @NSemana char(10), @ENDia char(10), @ENDiaSemana char(10)
DECLARE @ESMes char(18), @ESMes31 char(6)
DECLARE @ESDia char(10), @ESDiaSemana char(10)
--Set inicial por si no coincide con los del servidor
SET DATEFORMAT dmy
SET DATEFIRST 1

BEGIN TRANSACTION
    --Borrar datos actuales, si fuese necesario
    --TRUNCATE TABLE FROM DI_TIEMPO
    SET LANGUAGE us_english
    --Rango de fechas a generar: del 01/01/2006 al 31/12/Año
actual+2
    SELECT @FechaDesde = CAST(@FecDesde AS smalldatetime)
    SELECT @FechaHasta = CAST(CAST(YEAR(GETDATE()))+@Anios AS
CHAR(4)) + '1231' AS smalldatetime)

    WHILE (@FechaDesde <= @FechaHasta) BEGIN
        SELECT @FechaAAAAMMDD = YEAR(@FechaDesde)*10000+
            MONTH(@FechaDesde)*100+
            DATEPART(dd, @FechaDesde)
        SELECT @Año = DATEPART(yy, @FechaDesde)
        SELECT @Trimestre = DATEPART(qq, @FechaDesde)
        SELECT @Semestre = CASE
            WHEN @Trimestre <= 2 THEN 1
            ELSE 2
        END
    END

```

```

SELECT @Mes = DATEPART(m, @FechaDesde)
SELECT @Semana = DATEPART(wk, @FechaDesde)
SELECT @Dia = RIGHT('0' + DATEPART(dd, @FechaDesde), 2)
SELECT @DiaSemana = DATEPART(DW, @FechaDesde)
SELECT @ENMes = DATENAME(mm, @FechaDesde) + '/' + RIGHT(@Año,
2)
SELECT @ENMes31 = LEFT(@ENMes, 3) + '/' + RIGHT(@Año, 2)
SELECT @NSemestre = 'S' + CAST(@Semestre as CHAR(1)) + '/' +
RIGHT(@Año, 2)
SELECT @NTrimestre = 'T' + CAST(@Trimestre as CHAR(1)) + '/' +
RIGHT(@Año, 2)
SELECT @NSemana = 'Sem ' + CAST(@Semana AS CHAR(2)) + '/' +
RIGHT(RTRIM(CAST(@Año as CHAR(4))), 2)
SELECT @ENDia = case when len(CAST(@Dia as CHAR(2))) = 1 then
'0'+CAST(@Dia as CHAR(2)) else CAST(@Dia as CHAR(2)) end + ' ' +
@ENMes31
SELECT @ENDiaSemana = DATENAME(dw, @FechaDesde)

SET LANGUAGE spanish
SELECT @ESMes = DATENAME(mm, @FechaDesde) + '/' + RIGHT(@Año,
2)
SELECT @ESMes31 = LEFT(@ESMes, 3) + '/' + RIGHT(@Año, 2)
SELECT @ESDia = case when len(CAST(@Dia as CHAR(2))) = 1 then
'0'+CAST(@Dia as CHAR(2)) else CAST(@Dia as CHAR(2)) end + ' ' +
@ESMes31
SELECT @ESDiaSemana = DATENAME(dw, @FechaDesde)

SET LANGUAGE us_english
INSERT INTO dbo.TD_Tiempo
(
    id_tiempo,
    Fecha,
    Anio,
    Semestre,
    Trimestre,
    Mes,
    Semana,
    Dia,
    DiaSemana,
    NSemestre,
    NTrimestre,
    ENMes,
    ENMes3L,
    NSemana,
    ENDia,
    ENDiaSemana,
    ESMes,
    ESMes3L,
    ESDia,
    ESDiaSemana
) VALUES
(
    @FechaAAAAMDD,
    @FechaDesde,
    @Año,

```

```

        @Semestre,
        @Trimestre,
        @Mes,
        @Semana,
        @Dia,
        @DiaSemana,
        @NSemestre,
        @NTrimestre,
        @ENMes,
        @ENMes31,
        @NSemana,
        @ENDia,
        @ENDiaSemana,
        @ESMes,
        @ESMes31,
        @ESDia,
        @ESDiaSemana
    )

    --Incremento del bucle
    SELECT @FechaDesde = DATEADD(DAY, 1, @FechaDesde)
    END
    COMMIT TRANSACTION
END

```

ANEXO C: Script de procesamiento del cubo de información en
comandos XMLA

```

<Batch
xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Parallel>
    <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ddl2="http://schemas.microsoft.com/analysiservices/2003/engine/2"
xmlns:ddl2_2="http://schemas.microsoft.com/analysiservices/2003/engine/2/2"
xmlns:ddl100_100="http://schemas.microsoft.com/analysiservices/2008/engine/100/100"
xmlns:ddl200="http://schemas.microsoft.com/analysiservices/2010/engine/200"
xmlns:ddl200_200="http://schemas.microsoft.com/analysiservices/2010/engine/200/200">
      <Object>
        <DatabaseID>ASP_CB_Asignacion</DatabaseID>
      </Object>
      <Type>ProcessFull</Type>
      <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
    </Process>
  </Parallel>
</Batch>

```

ANEXO D: Carta de conformidad de la empresa auspiciante Babel
Software S.A.

CARTA DE CONFORMIDAD – RECEPCIÓN DEL DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN DE TIEMPOS EN PROYECTOS DE DESARROLLO DE SOFTWARE Y CONTROL DE DESEMPEÑO MEDIANTE CUBOS DE INFORMACIÓN PARA TOMA DE DECISIONES GERENCIALES. CASO PRÁCTICO: CUBOS DE INFORMACIÓN PARA EL CONTROL DEL DESEMPEÑO EN LA EMPRESA BABEL SOFTWARE S.A.

En la ciudad de Latacunga a los tres días del mes de Enero del presente año, se reúnen los señores Ing. Juan Peñaherrera, Lcda. Maribel Silva por Babel Software S.A. y por otra parte el Sr. Diego Gallardo con la finalidad de proceder con la entrega del proyecto: CUBOS DE INFORMACIÓN PARA EL CONTROL DEL DESEMPEÑO EN LA EMPRESA BABEL SOFTWARE S.A.

1.- ANTECEDENTES.-

El 7 de Octubre del 2010 el consejo académico por medio del Ing. José Luis Carrillo aprobó el tema de tesis del Sr. Diego Gallardo, el cual a su vez es auspiciado por la empresa Babel Software S.A.

Una vez otorgada la aprobación, el Sr. Diego Gallardo desarrolló su tesis en las instalaciones de la Empresa contando con la revisión de funcionarios del área de Business Intelligence de Babel Software.

2.- RECEPCIÓN.-

El 3 de Enero del 2012 se reúnen las personas arriba mencionadas y proceden a la verificación del proyecto de tesis. La misma que contiene:

- El cubo de información sobre proyectos.
- Los ETLs correspondiente al proceso de Data Warehouse.
- Los JOBS para actualizar las bases de datos y el cubo.
- Y por último el reporte en Excel.

3.- CONCLUSIÓN.-

Los funcionarios de la empresa Babel Software verificaron toda la información, documentación y el proyecto en el que consta que se ha cumplido con lo establecido en el plan de tesis del Sr. Diego Gallardo, por lo que resuelven dar por recibido y expresar formalmente su conformidad.



Gerente Administrativa – Babel Software
Lcda. Maribel Silva



Gerente Regional – Babel Software
Ing. Juan Peñaherrera

BIBLIOGRAFÍA

-  CAMERON, Scott. Microsoft SQL Server 2008 Analysis Services Step by Step, Microsoft Press, Washington – Estados Unidos, 2009, p. 401.
-  JACOBSON, Reed. Microsoft SQL Server 2005 Analysis Services Step by Step, Microsoft Press, Washington – Estados Unidos, 2006, p. 384.
-  MISTRY, Ross. Introducing Microsoft SQL Server 2008 R2, Microsoft Press, Washington – Estados Unidos, 2010, p. 215
-  ZVENGER, Patricia A. Incorporación de Soluciones OLAP en entornos empresariales, Director: DR. Manuel Fidel, Universidad Nacional del Sur de Argentina, Diciembre 2005
-  VERÁSTEGUI, Hazbleydi. Modelo Dimensional de datos [PDF]. http://www.db-system.com/pls/portal/docs/PAGE/SITIOWWWDB/ARTICULOS/MODELADO%20DIMENSIONAL%20DE%20DATOS_V2.PDF. [Ref. 19 agosto 2010].
-  WOLFF, Carmen G. Modelamiento multidimensional [PDF]. <http://www.inf.udec.cl/~revista/ediciones/edicion4/modmulti.PDF>. [Ref. 19 agosto 2010].
-  BROBELO. Manual Analysis Services [DOC]. http://www.slideshare.net/brobelo/manual-analysis-services?src=related_normal&rel=574868. Necesita subscripción a la página para descargar el documento. [Ref. 19 agosto 2010].
-  KIMBALL, Ralph. Drawing the line between dimensional modeling and ER modeling techniques [En línea]. http://www.ralphkimball.com/html/articles_search/articles1997/9708d15.html. [Ref. 18 agosto 2010].
-  Wikipedia. OLAP [En línea]. <http://es.wikipedia.org/wiki/OLAP>. [Ref. 23 agosto 2010].

- 🌐 OLAP. OLAP [En línea].
http://olap.com/w/index.php/OLAP_Education_Wiki. [Ref. 20 agosto 2010].
- 🌐 KRISHNASWAMY, Jayaram. Creating an Analysis Services Cube with Visual Studio 2008 – Part 1 [En línea], Julio 2008.
<http://www.packtpub.com/article/creating-analysis-services-cube-visual-studio-2008-part1>. [Ref. 19 agosto 2010].
- 🌐 Accelebrate, Inc. SSAS Tutorial: SQL Server 2008 Analysis Services Tutorial [PDF].
http://www.accelebrate.com/sql_training/PDF/SSAS_2008_tutorial.pdf. [Ref. 24 agosto 2010].
- 🌐 BANDARUPALLI, Kalyan. Creating an Analysis Services Project in Visual Studio 2008 [En línea], 10 Marzo 2010.
<http://www.techbubbles.com/sql-server/creating-an-analysis-services-project-in-visual-studio-2008/>. [Ref. 19 agosto 2010].
- 🌐 PEARSON, William. Introduction to SQL Server 2000 Analysis Services: Creating our first cube [En línea], 3 septiembre 2002.
http://www.databasejournal.com/features/mssql/article.php/10894_1429671_1/Introduction-to-SQL-Server-2000-Analysis-Services-Creating-Our-First-Cube.htm. [Ref. 18 agosto 2010].
- 🌐 Aprende Dynamics AX. Tecnología OLAP [En línea].
<http://www.aprendedynamics.com/olap1.html>. [Ref. 23 agosto 2010].
- 🌐 Wikipedia. Microsoft Visual Studio [En línea].
http://es.wikipedia.org/wiki/Microsoft_Visual_Studio. [Ref. 24 agosto 2010].
- 🌐 Wikipedia. Microsoft Excel [En línea].
http://es.wikipedia.org/wiki/Microsoft_Excel. [Ref. 24 agosto 2010].
- 🌐 Sinnexus. Datamart [En línea].
http://www.sinnexus.com/business_intelligence/datamart.aspx. [Ref. 19 agosto 2010].

- 🌐 Wikipedia. Extract, transform and load [En línea].
http://es.wikipedia.org/wiki/Extract,_transform_and_load. [Ref. 20 agosto 2010].
- 🌐 Talend. ETL para Analytics [En línea].
<http://es.talend.com/solutions-data-integration/etl-for-analytics.php>. [Ref. 20 agosto 2010].
- 🌐 Software GREENHOUSE. Data Mart [En línea].
<http://www.swgreenhouse.com/Productos/Hi-Spins/DataMart.html>. [Ref. 19 agosto 2010].
- 🌐 TopBits.com. Cubo OLAP [En línea]. <http://www.tech-faq.com/es/cubo-olap.html>. [Ref. 24 agosto 2010].
- 🌐 ESPINOSA, R. Cubos OLAP (On – Line Analytic Processing) [En línea]. <http://www.dataprix.com/blogs/respinosamilla/cubos-olap-line-analytic-processing>. [Ref. 19 agosto 2010].
- 🌐 BISBÉ, Ana. Dimensiones, Atributos y Jerarquías en cubos OLAP y Excel 2007 (I) [En línea]. <http://amby.net/2010/01/22/dimensiones-atributos-y-jerarquias-en-cubos-olap-y-excel-2007-i/>. [Ref. 23 agosto 2010].
- 🌐 MARTÍNEZ, Alfredo. OLAP y el diseño de cubos [En línea].
<http://www.gestiopolis.com/canales8/ger/olap-online-analytic-processing.htm>. [Ref. 19 agosto 2010].
- 🌐 Wikipedia. MultiDimensional eXpressions [En línea].
http://en.wikipedia.org/wiki/MultiDimensional_eXpressions. [Ref. 23 agosto 2010].
- 🌐 MSDN Microsoft. Consulta de MDX básica (MDX) [En línea].
<http://msdn.microsoft.com/es-es/library/ms144785.aspx>. [Ref. 23 agosto 2010].