

# Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería Escuela de Ingeniería Civil en Informática

## DESARROLLO DE UN SISTEMA DE *BUSINESS INTELLIGENCE* PARA EL AREA DE COMPRAS DE LA EMPRESA SALMOFOOD S.A.

Tesis para optar al título de Ingeniero Civil en Informática

PATROCINANTE: MARTIN SOLAR MONSALVES

MAURICIO ANDRES BARRIENTOS BARRIENTOS

VALDIVIA - CHILE AÑO 2005

- De : Martín Gonzalo Solar Monsalves
- A : Directora Escuela Ingeniería Civil en Informática

Ref. : Informe Calificación Trabajo de Titulación

## Nombre Trabajo de Titulación:

"DESARROLLO DE UN SISTEMA DE BUSINESS INTELLIGENCE PARA EL ÁREA DE COMPRAS DE LA EMPRESA SALMOFOOD S. A."

## Nombre Alumno:

Mauricio Andrés Barrientos Barrientos.

### Evaluación:

Cumplimiento del objetivo propuesto	7.0
Satisfacción de alguna necesidad	7.0
Aplicación del método científico	6.0
Interpretación de los datos y obtención de conclusiones	6.5
Originalidad	7.0
Aplicación de criterios de análisis y diseño	6.5
Perspectivas del trabajo	7.0
Coherencia y rigurosidad lógica	6.5
Precisión del lenguaje técnico en la exposición, composición, redacción e ilustración	6.5

Nota Final

6.7

Sin otro particular, atte.:

Martín Solar Monsalves

- DE : Miguelina Vega Rosales Profesor Instituto Informática
- A: Dirección Escuela Ingeniería Civil en Informática

Informo a usted que el Proyecto de Título " Desarrollo de un Sistema de Business Inteligence para el área de Compras de la Empresa Salmofood S.A", presentado por el señor Mauricio Andrés Barrientes Barrientes, cumple con el objetivo general propuesto, que es implementar un sistema BI para la empresa Salmofood S.A en el área de compras.

La metodología de trabajo y el lenguaje utilizado es el adecuado, sin embargo no queda claramente explicitada como fue realizada la implementación. Por lo anteriormente expuesto, califico este proyecto de título con nota 6,5 (seis, cinco).

Atentamente

Miguelina Vegá R.



## Universidad Austral de Chile Instituto de Informática

Valdivia, 11 de julio del 2005.

A: Prof. Miguelina Vega. Directora Escuela de Ing. Civil en Informática. De: Luis A. Álvarez G.

Motivo.

Informar calificación del siguiente Trabajo de Titulación:

### Título:

DESARROLLO DE UN SISTEMA DE BUSINESS INTELIGENCE PARA EL ÁREA DE COMPRAS DE LA EMPRESA SALMOFOOD S.A.

Alumno:

SR. MAURICIO ANDRÉS BARRIENTOS BARRIENTOS.

Nota: SEIS COMA CERO (6,0)

### JUSTIFICACIÓN DE LA NOTA:

Tiene el mérito de explorar un área nueva de desarrollo y de ser un proyecto aplicado y escalable en la empresa donde fue desarrollado.

Sin embargo, existen problemas de redacción, los principales son:

- No hace referencia a las figuras en el texto, no las explica y por lo tanto aparecen descontextualizadas.
- En el Capítulo 3, muchos párrafos no corresponden a redacción propia y no cita la fuente.
- En el desarrollo del Capítulo 4, no existe claridad suficiente en los pasos que se siguen para la implementación.

Sin otro particular, se despide atentamente

Luis Alberto Álvarez González

#### AGRADECIMIENTOS

A mi familia, que siempre me brindaron su apoyo y confiaron en que iba a ser capaz de llegar a esta especial instancia de mi vida.

A mis amigos y compañeros de universidad que hicieron que mi paso por la casa de estudios nunca podré olvidarlo.

A la Familia Loyola Castillo, quienes me permitieron vivir con ellos durante mi paso por la ciudad de Valdivia, adoptándome como un miembro más de su familia y siempre apoyándome con cada problema que enfrenté.

A la empresa Salmofood S.A. que permitió que comience mi desarrollo laboral con ellos y en especial a mis compañeros de trabajo en el departamento de Informática y Tecnología, quienes me apoyaron en todo momento con la realización de la tesis y me orientaron sobre cualquier otra consulta que pudiera tener.

A mi profesor patrocinante, por asesorarme en la realización de esta tesis. También por darse el tiempo y la paciencia para revisar cada documento que le entregué y los consejos que me dio.

También quiero agradecer a las familias Cifuentes Stegmaier y Pabst Riquelme, quienes fueron muy importantes en los últimos años de mi época estudiantil, brindándome su amistad y permitiendo que sea uno más sus respectivas familias. Gracias por todos los momentos que compartieron conmigo, cada palabra de aliento y consejo que me dieron.

## ÍNDICE

RESUMEN		1
<u>SUMMARY</u>		3
<u>CAPÍTULO 1</u>	1: INTRODUCCION	4
1.1	Objetivos Generales	5
1.2	Objetivos Específicos	6
1.3	Organización de la tesis	6
<u>CAPÍTULO 2</u>	2: ESTADO ACTUAL	8
2.1	Objetivos del capítulo	8
2.2	Descripción de la empresa y su situación actual	8
2.3	Solución propuesta	10
<u>CAPÍTULO 3</u>	3: DESCRIPCIÓN DE TECNOLOGÍAS	12
3.1	Objetivos del capítulo	12
3.2	Inteligencia de Negocios	12
3.3	Data Warehouse	15
	3.3.1 Arquitecturas de DW	16
	3.3.2 Esquemas de DW	18
3.4	Data Mart	22
3.5	Proceso ETL	24
	3.5.1 <u>Criterios en la selección de un producto ETL</u>	26
	3.5.2 <u>Herramientas de desarrollo de procesos ETL</u>	28
	3.5.2.1 Microsoft DTS	29

		3.5.2.2 Informatica Powercenter	31
		3.5.2.3 Ascential DataStage	32
		3.5.2.4 Cognos DecisionStream	33
		3.5.2.5 SAS ETL Studio	34
		3.5.2.6 Sagent Data Flow Server	35
<u>CAP</u>	<u>ÍTULO</u>	4: IMPLEMENTACIÓN DE LA SOLUCIÓN	39
4.1	<u>Objeti</u>	vos del capítulo	39
4.2	Implei	nentación de sistema de BI	39
	4.2.1	Ciclo de vida del proyecto	40
	4.2.2	Análisis de Riesgo	40
	4.2.3	Gestión de Configuración	45
	4.2.4	Gestión de la Calidad, Control y Garantía	45
		4.2.4.1 Definición de modelo de Calidad	46
		4.2.4.2 Control de la Calidad	50
		4.2.4.3 Garantía de Calidad	52
		4.2.4.4 Sistema de Gestión de la Calidad	52
	4.2.5	Especificación de requisitos	53
	4.2.6	Estimación del proyecto	54
	4.2.7	Diseño del sistema	63
	4.2.8	Instalación y Configuración de Herramientas	66
	4.2.9	Implementación de procesos ETL	69
		4.2.9.1 Crear nuevo proyecto	71
		4.2.9.2 Definir módulos Origen y Destino	71

	4.2.9.3 Definir el movimiento y transformación de datos	76
	4.2.9.4 Creación y programación de jobs	87
4.2.10	) Creación de informes de gestión	<u>90</u>
4.2.1	Configurando el acceso a los informes	99
<u>CAPÍTULO</u>	5: CONCLUSIONES	104
<u>GLOSARIO</u>	DE TÉRMINOS	106
<b>BIBLIOGR</b>	AFÍA	114
Anexo A:	Especificación de requerimientos mínimos y hardware	
	usado	123
Anexo B:	Acerca de la base de datos	127
Anexo C:	Acerca de Oracle Warehouse Builder	143
Anexo D:	Acerca de Oracle Discoverer	164
Anexo E:	Abreviaturas usadas	170
Anexo F:	Especificación de Requisitos de Software	173

v

## ÍNDICE DE FIGURAS

## Página

FIGURA 3.1	Diagrama de una solución de BI	13
FIGURA 3.2	Arquitectura básica de un DW	16
FIGURA 3.3	Arquitectura de DW con Staging Area	17
FIGURA 3.4	Arquitectura de DW con Staging Area y DM	17
FIGURA 3.5	Ejemplo de un esquema estrella	19
FIGURA 3.6	Ejemplo de un esquema snowflake	21
FIGURA 3.7	Ejemplo de DM dependiente	23
FIGURA 3.8	Ejemplo de DM independiente	23
FIGURA 3.9	Diagrama de un proceso ETL	25
FIGURA 3.10	Arquitectura de metadato de la OMG	37
FIGURA 4.1	Triángulo de McCall	46
FIGURA 4.2	Tipos de actividades de control	50
FIGURA 4.3	Factores de Escala	57
FIGURA 4.4	Parámetro de Programación	57
FIGURA 4.5	Cálculo del tamaño del módulo	58
FIGURA 4.6	Parámetros del cálculo del EAF	60
FIGURA 4.7	Resultados de estimación obtenidos para el módulo	61
FIGURA 4.8	Resultados de estimación obtenidos para el proyecto	61
FIGURA 4.9	Tabla de hechos de OC	64
FIGURA 4.10	Tabla de hechos de envíos vencidos	65
FIGURA 4.11	Tabla de hechos de solicitudes	65

FIGURA 4.12	Configuración de credenciales	68
FIGURA 4.13	Información de conexión del módulo destino	72
FIGURA 4.14	Configuración del módulo destino	76
FIGURA 4.15	Tablas del módulo destino	<u>79</u>
FIGURA 4.16	Mapping para carga de proveedores	81
FIGURA 4.17	Detalles de conexión al repositorio de Runtime	82
FIGURA 4.18	Resultados de la ejecución del mapping	
	MAP_PO_VENDORS	85
FIGURA 4.19	Conexión al repositorio de Runtime	86
FIGURA 4.20	Parte del informe de ejecución	87
FIGURA 4.21	Parámetros de ejecución del mapping	
	MAP_PO_VENDORS	89
FIGURA 4.22	Creación de la carpeta de Centros de Costo	93
FIGURA 4.23	Creación de join	95
FIGURA 4.24	Otorgando acceso a un área de negocios	100
FIGURA 4.25	Privilegios del usuario LANTIHUAL	101
FIGURA 4.26	Parte del informe OC x CC al mes	102
FIGURA A.1	Esquema de red de Salmofood S.A.	127
FIGURA B.1	Arquitectura de 3 capas de la base de datos Oracle	129
FIGURA C.1	Arquitectura de OWB	147
FIGURA D.1	Componentes de Oracle Discoverer	165

#### RESUMEN

Desde hace algunos años que las empresas del mundo entero han tratado de usar de mejor forma toda la información que posee en sus distintos sistemas. En este ámbito es que ha nacido la llamada Inteligencia de Negocios, que se encarga de transformar los datos en información útil y fácil de entender para las personas encargadas de la toma de decisiones.

Este proceso de Inteligencia de Negocios se hace en varias etapas: realiza un proceso de extracción de datos, se cargan en un repositorio central, se realizan consultas y se presenta a los usuarios finales para que puedan sacarle el máximo provecho posible.

Durante los siguientes capítulos se explicarán los conceptos básicos que se necesitan para implementar una solución de este tipo, centrándose en el caso particular del área de Compras de la empresa Salmofood S.A. En general, los pasos a seguir para lograr una implementación de un sistema de este tipo son los siguientes:

- Se necesita extraer la información de distintas fuentes y cargarla en un repositorio centralizado.
- Este repositorio de información puede ser por ejemplo: un *Data Warehouse* o un *Data Mart*.
- Se realizarán consultas sobre el repositorio.
- Los resultados de las consultas efectuados se mostrarán a los usuarios finales para que puedan usar esta información y trabajarla para poder tomar sus decisiones.

Es este esquema el que se utilizará para implementar la solución de Inteligencia de Negocios para la empresa Salmofood S.A.

#### SUMMARY

In the last years, companies all over the world have try to use all the data they have in their different systems in the best way. In this area, it has been created the socalled Business Intelligence, which is in charge of transforming data into useful and understandable information for people who are responsible for taking decisions.

This Business Intelligence process is developed in several stages: it makes a process in which data is taken out, after that this data is loaded into a central repository, then it's made some queries and finally it's presented to final users, so they can optimize the use of it as much as possible.

During the next chapters it will be explained basic concepts needed to carry out this kind of solution, focused in the specific case of the Purchasing department of Salmofood S.A. company. In general, to carry out a system like this, these are the steps to follow:

- It's needed to take out data from different sources and load it in a centralized repository.
- This data repository may be, for example: a Data Warehouse or a Data Mart.
- It will be made queries about the repository.
- Final users will be shown the results obtained from the queries, in order to use this data and work it, so they can make decisions.

This is the scheme that will be used to develop the Business Intelligence solution in Salmofood S.A. company.

#### **CAPÍTULO 1: INTRODUCCIÓN**

Un tema emergente durante los últimos años es la Inteligencia de Negocios (en adelante BI por su sigla en inglés *Business Intelligence*), el que ha surgido como respuesta a la necesidad de las empresas que han buscado una forma de tratar la gran cantidad de información que poseen en sus distintos sistemas operacionales. La idea es poder analizar todos los datos históricos que se poseen y de ser capaces de predecir tendencias futuras u obtener resultados que sirvan de base para orientar a las personas que se encuentran a cargo del proceso de toma de decisiones de una compañía.

La empresa Salmofood S.A. mantiene sus distintos sistemas operacionales desde el año 2001, por lo que ya posee una gran cantidad de información que puede utilizar para implementar un sistema de BI. Es en este punto cuando nace la oportunidad de desarrollar una solución de este tipo.

Comenzar el desarrollo de un sistema de estas características que se enfoque en toda la empresa resulta ser un proceso demasiado largo, por lo que se recomienda realizarlo por etapas o abordar a la compañía por departamentos, áreas o secciones. Es por esto, que para realizar una primera implementación de un sistema de BI, se trabajará solamente con un área específica.

Como se trabajará con un área específica, es que se implementará un *Data Mart* (DM) para usarlo de repositorio de toda la información importante para las personas que toman las decisiones en esta área. En etapas posteriores y que no forman parte de este proyecto, se implementarán otros DM que incorporen las áreas restantes de la compañía, como son: Despacho, Ventas, Producción, Contabilidad, Finanzas, Asistencia Técnica y Recursos Humanos. Finalmente se podrán integrar todos los DM dentro de un *Data Warehouse* (DW) que abarque a todas las áreas existentes.

No toda la información existente respecto a Compras es analizada o es importante para el personal a cargo de la toma de decisiones. Muchas veces, las tablas existentes en los distintos sistemas de información poseen varios campos que no necesitan ser considerados para tomar una decisión, por lo que no tendría sentido copiar toda la información de un lado para otro. Es por este motivo que se usarán procesos ETL (*Extract Transform Load data*) que llevarán solamente los datos necesarios al DM de Compras. Usando los procesos ETL, se almacenará la información en un repositorio central, que usará como esquema de almacenamiento el modelo multidimensional de tipo estrella.

Sobre el DM de Compras se realizarán todas las consultas necesarias para extraer la información que se presentará a las personas que están encargadas del proceso de toma de decisiones relacionadas con el área de compras. Estas personas serán capaces de trabajar con está información utilizando herramientas de análisis especialmente diseñadas para este fin y podrán llevar está información a otras aplicaciones para que le puedan sacar el mayor provecho posible.

#### 1.1 OBJETIVOS GENERALES

Se tienen dos objetivos generales para esta tesis:

• Lograr la implementación de un sistema de BI para el área de compras de la empresa Salmofood S.A.

• Estudiar las distintas tecnologías que ayudan en el proceso de implementación de una solución de BI.

#### **1.2 OBJETIVOS ESPECÍFICOS**

Los objetivos específicos que se resolverán en esta tesis son los siguientes:

- Lograr un entendimiento de las distintas tecnologías que ayudan en el proceso de implementación de un proceso de BI.
- Seguir el proceso de Ingeniería de Software para garantizar que lo que se desarrollará será realmente lo que la empresa y sus usuarios requieren.
- Desarrollar un sistema de BI que satisfaga fielmente los requerimientos de los usuarios, de modo que una vez finalizado, éste sea utilizado en las actividades diarias de los empleados.

#### **1.3 ORGANIZACIÓN DE LA TESIS**

La tesis se organiza de la siguiente forma:

- En el capítulo 2 se describirá la situación actual de la empresa, por qué se necesita un sistema de estas características y la solución que se ha de implementar para ayudarlos.
- En el capítulo 3 se explicarán las distintas tecnologías y herramientas que se usan para construir un sistema de BI, como lo son: el modelamiento multidimensional, DWs, DMs y herramientas ETL.
- En el capítulo 4 se mostrará el proceso de implementación completo del sistema de BI realizado para la empresa Salmofood S.A., respaldado teóricamente por técnicas de Ingeniería de Software.

- En el capítulo 5 se presentarán las conclusiones e impresiones obtenidas tras la realización del sistema de BI.
- Adicionalmente se presentan anexos explicando principalmente la forma en que se instalaron los distintos productos de software usados para desarrollar el sistema de BI y cómo fueron configurados.

#### **CAPÍTULO 2: ESTADO ACTUAL**

#### 2.1 OBJETIVOS DEL CAPÍTULO

En este capítulo se describirá a la empresa y la política de calidad que tiene definida para operar. En este punto se establecerá la importancia de contar con un sistema de BI para mejorar su sistema de gestión. Al final del capítulo de propondrá una solución adecuada para ayudar a la empresa en la toma de decisiones y que de esta forma pueda cumplir con la política de calidad.

## 2.2 DESCRIPCIÓN DE LA EMPRESA Y SU SITUACIÓN ACTUAL

Ubicada en el centro de la Isla Grande de Chiloé, Salmofood S.A. posee una de las plantas productoras de alimentos para peces más modernas del mundo, abarcando una superficie construida de más de 14.000 metros cuadrados y tiene una capacidad para generar más de 140.000 toneladas de alimentos por año.

La misión de Salmofood S.A. es apoyar la competitividad de sus clientes con los mejores factores de conversión, productos específicos que respondan a cada mercado y cultura de consumo que enfrenten, conseguir junto a ellos la satisfacción, salud y seguridad del consumidor final de los cultivos marinos.

Algunos de los valores que la empresa asegura tener a sus clientes son los siguientes:

- **Profesionalismo**: Alcanzar altos niveles de competencia y habilidades.
- Velocidad: Tomar decisiones rápidas y actuar oportunamente.
- **Compromiso**: Actuar con responsabilidad e iniciativa para alcanzar los objetivos de la organización.

8

- Visión: Crear una visión de futuro claro y comunicarse para compartirla dentro de la organización.
- Desarrollo humano: Apoyar integralmente al desarrollo potencial de otros.

Además, la empresa trabaja con un sistema de gestión que permite asegurar que se identificarán y satisfarán las necesidades de los clientes, siendo eficaces y eficientes en la planificación, el mantenimiento y la mejora del desempeño de los procesos internos con el objetivo de lograr competitividad.

La empresa Salmofood cuenta con muchos proveedores para satisfacer sus distintas necesidades, con los cuales se establecen compromisos que se deben cumplir sin faltas, de modo que se asegure el correcto abastecimiento de la organización para poder responder de la misma forma a sus clientes. Los proveedores de la empresa van desde los transportistas, responsables de la entrega del producto terminado en el lugar acordado con el cliente hasta las empresas que proveen de materias primas para la elaboración de los distintos productos que se comercializan.

Salmofood S.A., según su sistema de gestión, se ha comprometido a evaluar el desempeño de sus proveedores para asegurar la correcta atención de los requerimientos de los clientes, por lo que siempre se busca la forma de tener un mejor control sobre éste y otros aspectos dentro de la compañía.

Actualmente, el proceso de compra consiste de las siguientes etapas:

- Especificación de los requisitos de compra.
- Información al proveedor.
- Cotización y negociación.
- Compra.

- Recepción.
- Inspección, pesaje y control de calidad.
- Evaluación del proveedor.
- Se informa al proveedor de los resultados.

Los proveedores están informados de los requisitos que deben cumplir para mantener una relación comercial fructífera con Salmofood S.A. Para la empresa, le es primordial que los proveedores sean capaces de cumplir con los requisitos establecidos, fundamentalmente en cuanto a la calidad del producto, seriedad de la entrega y precio. Es importante que exista un mecanismo de control sobre el desempeño de cada uno de los proveedores, de modo que las personas que se relacionan directamente con ellos, puedan extraer información sobre el desempeño de cada uno y tomar decisiones acerca de si seguir o no con determinado proveedor, aumentar o disminuir la cantidad de productos o servicios por proveedor, etc. También existe gente que desea conocer el desempeño de los compradores, conociendo aspectos como su actividad de compra, el ahorro por comprador u órdenes de compra abiertas o canceladas por cada uno de ellos.

Es a raíz de lo anterior, que se presenta la necesidad de contar con un mecanismo que sirva para controlar todos estos aspectos y ayude a las personas encargadas de la toma de decisiones en su tarea.

#### 2.3 SOLUCIÓN PROPUESTA

Para satisfacer las necesidades de la empresa Salmofood S.A. en su gestión sobre el proceso de compras, proveedores y compradores, se implementará un DM del módulo de compras de la compañía y se ayudará con éste al proceso de la toma de decisiones.

Para realizar esto, es necesario tomar la información de las bases de datos que tiene la organización y extraer todo lo que sea importante para la toma de decisiones. Estos datos serán llevados a una nueva base de datos, especialmente construida para recibir las consultas sobre el proceso de compras.

El acceso a esta base de datos será permitido solamente a las personas que están encargadas de la toma de decisiones.

#### **CAPÍTULO 3: DESCRIPCIÓN DE TECNOLOGÍAS**

#### 3.1 OBJETIVOS DEL CAPÍTULO

En este capítulo se describirá lo que es la BI. Además se explicará con más detalle los elementos que forman parte de este proceso, como lo son DW, DM y los procesos ETL. En relación a estos procesos, se especificará una guía acerca de cómo escoger la herramienta de software a utilizar. Además se nombrarán algunos de los productos más usados para el desarrollo de este tipo de procesos y sus características.

#### **3.2 INTELIGENCIA DE NEGOCIOS**

BI es el proceso mediante el cual se pretende incrementar la ventaja competitiva de una organización por medio del uso inteligente de los datos disponibles para la toma de decisiones.

Un proceso de BI consta de 5 etapas claves:

- *Data Sourcing*. Trata de extraer la información de las distintas fuentes.
- Análisis de Datos. Trata de sintetizar el conocimiento útil de los datos recopilados o extraídos, usando *data mining* o técnicas de análisis.
- Conocimiento de la Situación. En esta etapa se reconocen los hechos útiles y se filtra la información relevante para la toma de decisiones.
- Análisis de Riesgo. Se identifican las decisiones razonables o los cursos de acción basados en la espera de un riesgo específico.
- **Toma de Decisiones**. Se emplea software semi-interactivo para identificar las buenas decisiones y estrategias.

Las soluciones de BI incluyen software para realizar los procesos de extracción, transformación y carga de datos, *data warehousing*, consultas sobre las

bases de datos, informes, análisis OLAP, *data mining* y visualización de información. El diagrama de la figura 3.1 muestra las alternativas que se tienen a la hora de implementar una solución de BI:



#### Figura 3.1: Diagrama de una solución de BI.

El proceso que se seguirá para implementar la solución de BI en la empresa

Salmofood S.A. es el siguiente:

- Se tendrán los orígenes de datos para extraer la información importante.
- Se realizará el proceso de Extracción / Transformación / Carga de datos.
- Se cargarán los datos en un DM.
- Se usará una herramienta de consulta para trabajar la información del DM.

 Se usarán computadores que tendrán instalado el software de consulta en cada uno de ellos, lugar donde cada usuario trabajará con la información que necesita.

Algunos de los beneficios que trae el hecho de incorporar una solución de BI son los siguientes:

- La capacidad de producir informes y efectuar análisis que de otro modo resultaría difícil o costoso desarrollar y mantener usando las herramientas y los sistemas operacionales.
- Los datos son consistentes y confiables en todos los niveles de la organización. Esto minimiza los problemas de integridad de datos.
- Con un acceso más fácil a la información, los usuarios son capaces de producir sus propios informes, crear múltiples vistas, ejecutar análisis de "what if" y ver tendencias de la empresa.
- Las medidas de seguridad aseguran que la información sensible se protege y ayuda a mantener un control sobre a quién se le da acceso y sobre qué.
- Es mucho más fácil compartir información con las personas que se necesiten, incluyendo si es necesario, a clientes y proveedores.
- El reporte, análisis y visualización de los datos es más flexible e intuitivo, mejorando el entendimiento de las métricas de desempeño del negocio, lo que lleva a un mejor proceso de toma de decisiones.

A continuación se explicarán algunos conceptos que son importantes de entender en un esquema de solución de BI. Se explicarán en detalle solamente aquellas que son primordiales para implementar BI dentro de la empresa Salmofood S.A.

#### 3.3 DATA WAREHOUSE

Un DW es una base de datos relacional que se diseña para consultas y análisis en vez de para procesar transacciones. Usualmente contiene datos históricos derivados de las transacciones de datos, pero puede incluir datos desde otras fuentes. Además de una base de datos relacional, un ambiente de DW incluye un proceso ETL, herramientas de análisis para el usuario final y otras aplicaciones que controlan procesos encargados de reunir los datos y entregárselos a los usuarios finales.

Un DW debe tener las siguientes características:

- Está diseñado para ayudar en el análisis de datos.
- Puede integrar datos de distintos orígenes de forma consistente, sin conflictos de nombres o inconsistencias en cosas como las unidades de medida.
- Los datos deben ser no volátiles, es decir, que cuando los datos ingresan al DW, éstos no deberían cambiar.

Existen varias diferencias entre un DW y un sistema OLTP (*On-Line Transaction Processing*), entre las que se encuentran:

- Un DW generalmente no está definido en 3FN (Tercera Forma normal), muy común en los sistemas OLTP.
- Los DW usan a menudo esquemas desnormalizados o parcialmente desnormalizados para optimizar las consultas. Los sistemas OLTP usan frecuentemente esquemas completamente normalizados para mejorar el

desempeño de operaciones de inserción/actualización/eliminación de registros y garantizar la consistencia de los datos.

- Un DW se actualiza regularmente mediante procesos ETL. Los usuarios finales no actualizan directamente el DW. En los sistemas OLTP la base de datos siempre está actualizada y refleja el estado actual de cada transacción del negocio.
- Un DW almacena muchos meses o años de datos. Los sistemas OLTP almacenan algunas semanas o meses de datos.

#### 3.3.1 ARQUITECTURAS DE DW

Los DW y sus arquitecturas varían dependiendo de la situación específica de la organización en la que se implemente. Existen tres arquitecturas que son comunes:

• Arquitectura básica de DW.



Figura 3.2: Arquitectura básica de un DW.

• Arquitectura de DW con área de "estacionamiento" (*staging area*).



Figura 3.3: Arquitectura de DW con Staging Area.

• Arquitectura de DW con "staging area" y DM.



Figura 3.4: Arquitectura de DW con Staging Area y DM.

#### 3.3.2 ESQUEMAS DE DW

Un esquema es una colección de objetos de base de datos, incluyendo tablas, vistas, índices y sinónimos. Los esquemas más usados para modelar DW son: *esquema star (estrella)* y *esquema snowflake (copo de nieve)*.

El esquema estrella es quizás el más simple. Se denomina estrella porque el diagrama entidad-relación (E-R) de este esquema se parece a una estrella, con puntos que provienen de una tabla central. El centro de la estrella consiste de una gran tabla de hechos (*fact table*) y los puntos de la estrella son las tablas dimensión (*dimension tables*).

Un esquema estrella se caracteriza por una o más tablas de hechos que contienen la información principal en el DW y un número de tablas dimensión, cada una de las cuales contiene información acerca de entradas para un atributo particular en la tabla de hechos.

Una tabla de hechos es la tabla principal en un modelo dimensional, donde se almacenan las medidas del negocio. Ejemplo de estas medidas pueden ser: número de unidades compradas o recepcionadas, ventas promedio en dólares, etc.

Una tabla dimensión contiene el detalle de los valores que se encuentran asociados a la tabla de hechos.

En algunas ocasiones, cuando se diseña una base de datos, no está claro si el campo de datos numérico que se extrae de una fuente de datos de producción es un atributo de una dimensión o de un hecho. Generalmente se puede tomar esa decisión preguntándose si el campo es una medida que puede tomar muchos valores y participa en cálculos (en este caso debería ser un hecho) o si es una descripción valorada discretamente que es más o menos constante y participa en restricciones (debería ser una dimensión).

Cada tabla dimensión se une a la tabla de hechos usando un *primary key* o un *foreign key*, pero las tablas dimensión no se unen entre si.

Una típica tabla de hechos contiene claves y medidas. Por ejemplo, la tabla de hechos *ventas*, contiene medidas como *cantidad\_vendida* o *precio* y contiene claves como *cliente\_id, fecha, producto\_id o vendedor\_id*. Pueden existir tablas dimensión de *vendedores, clientes, fecha y productos*. Un esquema estrella que muestra esta situación se puede apreciar en la figura 3.5.



Figura 3.5: Ejemplo de un esquema estrella.

Una unión de la estrella es un enlace desde un *primary key* de la tabla dimensión a un *foreign key* de la tabla de hechos.

Las principales ventajas del esquema estrella son:

- Da un *mapping* directo e intuitivo entre las entidades de negocio que están siendo analizadas por los usuarios finales y el diseño del esquema.
- Da un desempeño altamente optimizado para las típicas consultas de estrella.
- Son ampliamente soportadas por un gran número de herramientas de BI, los cuales pueden anticipar o incluso requerir que el esquema de DW contenga tablas de dimensión.

Los esquemas estrella son usados también en los DM.

El esquema *snowflake* es más complejo que el esquema estrella y es un tipo de modelo estrella. Se llama esquema *snowflake* porque el diagrama del esquema se parece a un copo de nieve. El esquema *snowflake* normaliza las dimensiones para eliminar la redundancia, esto es, que los datos de las dimensiones se agrupan en varias tablas en vez de una sola gran tabla. El resultado de este esquema son consultas más complejas y que reducen el desempeño de éstas. El esquema *snowflake* aumenta el número de tablas en el modelo que se quiera implementar. Un ejemplo de esquema *snowflake* se puede apreciar en la figura 3.6.



#### Figura 3.6: Ejemplo de un esquema snowflake.

El problema ahora es cuál esquema escoger para implementar el DW o el DM. En muchas oportunidades se recomienda usar el esquema *snowflake*, ya que como tradición se ha tenido la idea de normalizar los modelos de datos para remover toda la información redundante. Sin embargo, el separar la información de una dimensión a través de varias tablas, provoca que se aumente la cantidad de uniones necesarias para ejecutar una consulta, complicando el modelo de datos a los ojos del usuario y a menudo disminuyendo el desempeño de la consulta.

Dada la diferencia en tamaño entre la tabla de hechos y las de dimensión que la rodean, las tablas de hechos son donde se concentra todo el esfuerzo para mejorar el desempeño y reducir el espacio de almacenamiento, dado que estas tablas ocupan entre el 80% y el 90% del espacio total que ocupa el DW.

Es cierto que el esquema estrella presenta mayor redundancia de datos, pero eso no es un impedimento a la hora de tener el medio de almacenamiento que soporte grandes cantidades de información (tratándose de un sistema grande), ya que en la actualidad se dispone de discos muy grandes y que trabajan a grandes velocidades, por lo que ya dejó de ser una desventaja para este esquema. Esto, unido a que este esquema generalmente mejora el desempeño de las consultas ha sido gravitante a la hora de escoger al esquema estrella como el que será utilizado para el caso de estudio particular.

#### 3.4 DATA MART

Ralph Kimball dijo en diciembre de 1997 que "El DW no es más que la unión de todos los DM" [23]. Kimball es de las personas más reconocidas en lo que a DW y modelos multidimensionales se refiere.

Un DM es una base de datos diseñada para ayudar a los administradores a tomar decisiones estratégicas sobre el negocio. Si el DW toma bases de datos que son para satisfacer las necesidades de información de toda la empresa, el DM se enfoca en un departamento o en un área específica dentro de la compañía. Se podría decir que son subconjuntos de un DW.

Existen dos clases de DM:

Dependientes, en los cuales el origen de la información proviene de un DW.
Todos los DM dependientes se alimentan por la misma fuente (un DW). En la figura 3.7 el DW está compuesto de distintas fuentes (varios DM) de colores usados para formar un solo DM.



Figura 3.7: Ejemplo de DM dependiente.

• Independientes, en los cuales el origen de la información proviene del ambiente de aplicaciones de la compañía. Cada DM independiente es alimentado única y separadamente por un ambiente de aplicaciones. En la figura 3.8 se ve un ejemplo ilustrando cómo se carga un DM a partir de varias bases de datos.



Figura 3.8: Ejemplo de DM independiente.

Un DM se diseña para satisfacer las necesidades de un departamento específico. En el actual caso de estudio, se trabajará con el departamento de Compras.

A diferencia de un DW, el cual puede tomar mucho tiempo para construirse, un DM puede construirse de forma muy rápida, en semanas o meses, a un costo considerablemente menor que el que lleva construir un DW.

Escoger la construcción de un DM en vez de un DW también reduce la necesidad de coordinar tantas actividades y mantener la cooperación a través de las unidades funcionales y operacionales por largos periodos de tiempo.

#### 3.5 PROCESO ETL

ETL es un proceso mediante el cual se extraen datos desde varias bases de datos, aplicaciones y sistemas, se transforman estos datos según sea necesario y se cargan en los sistemas destinos, incluyendo, pero no limitado a DW, DM, aplicaciones analíticas, etc.

El proceso ETL puede dividirse en tres claras etapas:

- Extracción. En esta etapa se identifican los datos que se necesitan y se leen desde sus respectivas ubicaciones (bases de datos, CRM (*Customer Relationship Management*), ERP (*Enterprise Resource Planning*), archivos planos, etc.)
- Transformación. En esta etapa se toman los datos extraídos y se transforman desde su estado o formato original al formato que se necesite. A menudo, esta etapa contempla un paso de depuración de los datos que se han transformado, de modo de eliminar cualquier inconsistencia de información.

• **Carga**. Durante esta etapa se toman los datos ya transformados y depurados para que sean cargados en la base de datos destino.

Todo esto puede verse apreciarse en la figura 3.9.



Figura 3.9: Diagrama de proceso ETL.

Más de la mitad del tiempo ocupado en proyectos de *data warehousing* se dedica típicamente al diseño e implementación de los procesos ETL. Los procesos ETL diseñados de forma mediocre son altamente costosos para mantener, cambiar o actualizar, por lo que es crítico realizar las decisiones correctas en términos de escoger las tecnologías adecuadas y las herramientas que se usarán para desarrollar y mantener los procesos ETL.

La funcionalidad de un producto ETL involucra cerca del 50% del criterio durante una evaluación normal. El objetivo de evaluar estos criterios es descubrir si el producto es capaz de extraer los tipos de datos que se tienen, hacer la mayoría de las transformaciones y cargar los destinos.

#### 3.5.1 CRITERIOS EN LA SELECCIÓN DE UN PRODUCTO ETL

A la hora de escoger cuál será la herramienta que se usará para desarrollar el proceso ETL es necesario evaluar varios factores que serán la clave de una buena decisión, como son por ejemplo, la administración de estos procesos y el ambiente de desarrollo.

Después que el proceso ETL está completo, una fácil administración es una de las preocupaciones más grandes. El mejor ambiente de desarrollo de ETLs es aquel en que la administración se hace fácil. Respecto al poder de administración que debería ofrecer una buena herramienta ETL y que se tendrían que poder controlar están las siguientes:

- Administración centralizada. Es importante que exista un solo lugar para ver el ambiente ETL completo, aún si los trabajos se ejecutan tanto en las plataformas de origen como en las de destino o bajo distintos sistemas operativos. Actualizar los trabajos programados o administrarlos debería poder hacerse desde una ubicación independiente.
- Informes ETL. Cada producto debería dar informes sobre las sesiones ETL y los trabajos, de forma que se pueda ver que cantidad de datos fueron procesados, de dónde provienen los datos, hacia dónde fueron enviados y cuánto tiempo tomó el proceso. La herramienta debería grabar las estadísticas para poder ver cómo se ha comportado el sistema a través del tiempo.
- Programación. La programación de tareas dentro de una sesión ETL no es una tarea fácil. Pueden existir muchas dependencias entre los trabajos, haciendo difícil determinar qué tareas pueden continuar ejecutándose si falló
algún trabajo anterior. En la mayoría de los casos, la plataforma operativa tendrá un soporte mínimo para programar tareas, particularmente lidiando con problemas de dependencias y ejecución condicional de trabajos siguientes.

• **Despliegue**. Una de las dificultades de las tareas de administración es manejar el despliegue de todos los trabajos ETL. Un producto debería tener la habilidad de agrupar objetos fácilmente, desplegarlos a un ambiente de *test* o de producción y poder realizar un *roll-back* si es que existe algún problema con el despliegue.

El otro aspecto importante a tomar en consideración a la hora de escoger una herramienta ETL es el ambiente de desarrollo. El ambiente de desarrollo es el lugar donde los programadores de procesos ETL ocupan todo su tiempo. Los criterios más importantes son algo subjetivos, como por ejemplo: ¿es el producto fácil de usar?, ¿es fácil de aprender?, ¿es rápido para desarrollar un trabajo? Sin embargo, existen otros criterios que se deben tomar en cuenta a la hora de elegir una herramienta de desarrollo de procesos ETL, como son:

- Interfaces visuales. El hecho de que exista la posibilidad de que el programador de procesos ETL pueda generar los mapeos (*mappings*) en forma visual, hace que sea mucho más fácil la programación de estos procesos.
- Soporte para depuración. Para una persona que desarrolla procesos ETL, puede ser útil ver los datos tanto antes como después de las transformaciones,

para evaluar si sus procesos resultan como lo desea o son necesarias algunas modificaciones.

- Soporte a línea de comandos. Hay casos donde es necesario poder ejecutar trabajos desde otro ambiente, para verificar el estado de los trabajos en ejecución, detener o reiniciar sesiones y también para usar o administrar el ambiente ETL.
- Soporte para mantenimiento. Existen características que ayudan en la tarea de mantención más de lo que ayudan en el proceso de desarrollo. Por esto, a menudo estas características son pasadas por alto a la hora de evaluar una herramienta que permita desarrollar procesos ETL. Un ejemplo claro es la capacidad de hacer cambios masivos, como en el caso de que si un sistema origen cambia y que esto signifique que varios procesos ETL deban cambiar, no se tenga que buscar y cambiar cada objeto. Un cambio masivo evita errores y ahorra tiempo. Otro ejemplo es la posibilidad de encapsular un *mapping* de datos y reutilizarlo en otros lugares.

#### 3.5.2 HERRAMIENTAS DE DESARROLLO DE PROCESOS ETL

A nivel mundial existen muchas herramientas que permiten el desarrollo de procesos ETL que están siendo ampliamente utilizadas. Por lo general, todas estas herramientas tienen características similares y varían solamente en detalles sobre la cantidad de tipos de orígenes y destinos distintos que trabajan cada una de ellas. No se efectuará un análisis muy profundo de estas aplicaciones, ya que se nombran solamente para efectos de conocer las alternativas que se tienen a la hora de escoger la herramienta adecuada para el desarrollo de este tipo de procesos. La herramienta que se utilizará es *Oracle Warehouse Builder*. Las razones por las que se usará este software y las características que posee se explicarán más adelante. A continuación, se describen brevemente algunas de las alternativas para desarrollar procesos ETL.

#### **3.5.2.1 MICROSOFT DTS**

*Microsoft SQL Server Data Transformation Services* (DTS) es un conjunto de herramientas gráficas y de objetos programables que permiten extraer, transformar y consolidar datos desde diferentes orígenes dentro de uno o varios destinos.

La estructura principal dentro de esta herramienta son los paquetes DTS. Un paquete DTS se organiza como una colección de conexiones, tareas DTS, transformaciones DTS y flujos de trabajos. Cada paquete contiene uno o más pasos que se ejecutan secuencialmente o en paralelo. Cuando se ejecuta, el paquete se conecta al origen correcto de los datos, copia datos y objetos de base de datos, transforma los datos y notifica a otros usuarios o procesos. Los paquetes pueden ser editados, protegidos con contraseñas y programados para ejecución.

Las *tareas DTS* son un conjunto discreto de funcionalidad, ejecutada como un solo paso en el paquete. Cada tarea define un ítem de trabajo a ser ejecutado como parte de un movimiento de datos y un proceso de transformación de datos o como un trabajo a ser ejecutado. Con las tareas se pueden realizar operaciones como: importar y exportar datos, transformar datos, copiar objetos de base de datos, enviar y recibir mensajes a y desde otros usuarios y paquetes, ejecutar instrucciones en Transact-SQL o *scripts* ActiveX sobre los datos de origen. También existe la posibilidad de crear tareas en lenguajes de programación que soporten la creación de objetos de tipo

COM (como Visual Basic, tanto el antiguo como el .NET) y acceder a ellas a través del DTS *Designer*.

Las *transformaciones DTS* consisten de una o más funciones u operaciones ejecutadas sobre los datos antes de que éstos lleguen a su destino. Entre las transformaciones DTS que se pueden hacer prácticamente están las mismas que pueden realizar la gran mayoría de las herramientas, pero además posee la capacidad de utilizar funciones escritas en *scripts* ActiveX o usar transformaciones sacadas de algún objeto de tipo COM.

Microsoft DTS incluye varias herramientas para ayudar al usuario en la creación de paquetes, su ejecución y la administración de éstos. Entre estas herramientas están:

- DTS *Import/Export Wizard*, usado para importar y exportar paquetes, además de copiar objetos de base de datos.
- DTS *Designer*, el cual es una aplicación gráfica que permite paquetes que contienen flujos de trabajo complejos o cuando se manejan varias conexiones a orígenes de datos heterogéneos.
- DTS *Query Designer*, que es una herramienta visual de base de datos que hace mucho más simple la elaboración de consultas en el DTS *Designer*.
- DTS *Run utility* (dtsrunui), el que permite ejecutar paquetes usando cuadros de diálogo.
- La utilidad DTSRUN (distinta a la anterior), la que permite ejecutar paquetes desde la línea de comandos.

• El nodo DTS en la consola de SQL *Enterprise Manager*, el cual se usa para crear, ver, cargar y ejecutar paquetes DTS, además de controlar las configuraciones del DTS *Designer* y administrar los logs de ejecución.

Esta herramienta también tiene algunas características importantes sobre la conectividad a las bases de datos. Debido a que se basa en la arquitectura OLE DB, le permite copiar datos desde una variedad de orígenes, por ejemplo:

- SQL Server, Oracle y DB2.
- Orígenes ODBC, usando Microsoft OLE DB *Provider* para ODBC.
- Access, Excel, Visual FoxPro, dBase, Paradox, HTML y archivos de texto.
- Microsoft Exchange Server, Microsoft *Active Directory* u otras fuentes de datos no relacionales.
- Informix es soportado solamente para efectos de extracción de datos y no para exportar datos a una base de datos de este tipo.

# **3.5.2.2 INFORMATICA POWERCENTER**

Sin duda alguna este es de uno de los productos de software más robustos en lo que a herramientas para construir y administrar procesos ETL se refiere. No por nada fue galardonado por segundo año consecutivo como el mejor software en la categoría ETL por los más de 80.000 suscriptores de la revista *Intelligent Enterprise Magazine*, publicación dedicada a aplicaciones de negocios estratégicas que convierten la información en inteligencia.

Informatica PowerCenter puede extraer datos de todo tipo de orígenes, como por ejemplo: archivos planos, bases de datos Oracle, sistemas de Oracle Financial, SAP, IBM DB2, PeopleSoft, archivos XML y muchos otros sistemas o bases de datos.

Ya se ha mencionado que la posibilidad de tener un ambiente visual para crear los mappings es muy útil y por supuesto, Informatica PowerCenter trae incorporada esta característica en su herramienta *Designer*, la cual es de muy fácil uso e intuitiva al usuario.

Este software también tiene la posibilidad de administrar los mappings generados, controlar el repositorio de datos, ofrece la posibilidad de ejecutar los trabajos por medio de la línea de comandos y permite programar la ejecución de los mappings.

Este software tiene un valor de US\$200.000. A pesar de esto, es uno de los productos más usados en el mundo entero para trabajar con los procesos ETL. Este costo elevado, representó de forma inmediata una traba para seleccionar este producto y desarrollar los procesos ETL que se necesitan para dar solución al caso de estudio actual. No se ahondará más en las características que posee este producto, ni tampoco en los que le siguen, ya que prácticamente tienen las mismas funciones.

## **3.5.2.3 ASCENTIAL DATASTAGE**

Este software es una poderosa herramienta de desarrollo de procesos ETL, que soporta la recopilación, integración y transformación de grandes volúmenes de datos, cuyas estructuras pueden ser simples o complejas. Para ejecutar los procesos ETL puede hacerlo en tiempo real o programar las ejecuciones.

Al igual que los otros productos que se usan para implementar estos procesos, Ascential DataStage soporta muchos tipos de orígenes y destino de datos, como son: archivos de texto, estructuras de datos complejas en XML, SAP, PeopleSoft, SAS, Microsoft SQL Server, Oracle y DB2 entre otras muchas bases de datos.

Actualmente, DataStage está siendo cada vez más utilizado, ya que al igual que Informatica PowerCenter, tiene herramientas visuales que hacen mucho más fácil la generación de procesos ETL, ofrece la posibilidad de realizar auditorías para ver el estado de la ejecución de los procesos, monitorear los orígenes y destinos de los datos, permite ejecutar procesos desde la línea de comandos. Con este software, también se pueden encapsular los *mappings* para su reutilización.

#### **3.5.2.4 COGNOS DECISIONSTREAM**

Al igual que las otras herramientas ya descritas, Cognos DecisionStream permite extraer datos de distintas fuentes y llevarlos a uno o más destinos. Entre algunas de sus características se tienen:

- DecisionStream posee un ambiente de diseño gráfico basado en Windows y uno multi-plataforma, lo que permite definir procesos de forma intuitiva y rápida.
- Es una herramienta que soporta el diseño de un modelo multidimensional.
- Da la posibilidad de ejecutar los procesos desde la línea de comandos.
- Permite la reutilización de los objetos con los que se construyen los procesos.
- Permite la actualización de los datos incremental, que es reconocida por la industria de *data warehousing* como la mejor opción para poblar los datos de un DW o un DM. La función de actualización incremental se divide en dos etapas para mejorar su desempeño. El primer paso consiste en insertar los nuevos datos en una carga masiva (*bulk-load*), lo que reduce la demanda de

recursos de procesamiento. El próximo paso es actualizar los cambios a los datos existentes, un proceso que involucra ir dentro de la base de datos, encontrar la fila a modificar, actualizarla y en este momento grabar los cambios. También se puede hacer que se refresque toda la información del DW o DM sin mayor problema.

- Permite programar la ejecución de los procesos ETL en plataformas UNIX y Windows.
- Los tipos de orígenes y destinos que soporta Cognos DecisionStream son los siguientes: Oracle, Informix, Sybase, MS SQL Server, DB2, ODBC, archivos planos, Teradata y SAP R/3.
- Las plataformas en las cuales se puede ejecutar este producto son: Windows, IBM AIX, Sun SPARC Solaris y HP-UX.

Sin duda alguna esta es otra poderosa herramienta disponible para realizar los procesos ETL, los requerimientos básicos para poder usarla son mínimos (128 MB de RAM y 55 MB de disco duro).

# 3.5.2.5 SAS ETL STUDIO

Este producto es una herramienta de diseño visual que ayuda a implementar y administrar los procesos ETL. Al igual que las otras herramientas visuales que realizan la misma función, ésta permite a los usuarios trabajar con los procesos ETL de forma sencilla. Entre las características que posee este producto se encuentran las siguientes:

• Permite la reutilización de metadato utilizado en los procesos ETL.

- Permite programar la ejecución de los trabajos para que los procesos ETL se desplieguen en el momento que el usuario así lo requiera.
- Tiene la opción de poder depurar los procesos, dando la posibilidad de realizar un *roll-back* si fuera necesario.
- Las orígenes de información pueden ser los siguientes: MS SQL Server, MS Access, MS Excel, Oracle, Informix, Sybase, Teradata, IBM DB2, archivos te texto, archivos XML, SAS y SAS/Share.
- También se puede conectar a sistemas ERP como por ejemplo: SAP BW,
  SAP R/3, Oracle Applications, Siebel y PeopleSoft.
- Posee un repositorio que es utilizado para realizar auditorías, en las que se puede ver quién escribió o ejecutó un trabajo, cuánto se demoró la ejecución, que cantidad de datos se cargaron, resultados obtenidos, orígenes y destinos de los datos.

Una desventaja que posee este producto es que no soporta la ejecución mediante línea de comandos como lo hacen los demás productos usados para el desarrollo de los procesos ETL, ya que se ha enfocado en trabajar mediante un ambiente gráfico de modo de facilitar la tarea a los desarrolladores.

#### **3.5.2.6 SAGENT DATA FLOW SERVER**

Esta herramienta es la que trae incorporada a *Sagent Data Load Server*, como un componente, que también es una utilidad de desarrollo visual como todos los productos ya mencionados. Entre sus características se encuentran las siguientes:

• Permite a los desarrolladores ver los resultados de sus procesos ETL sin necesidad de tener que cargar los datos en la base de datos destino.

- Permite la extracción de datos en tiempo real o también da la posibilidad de programar trabajos para que los procesos se ejecuten cuando se desee.
- Puede extraer información de diversos orígenes, como por ejemplo: Oracle, IBM DB2, MS SQL Server, Informix, Sybase, archivos de texto y archivos XML.
- También es capaz que conectarse a diversos ERP, como: PeopleSoft, Siebel, SAP.
- Posee un repositorio de metadato que al igual que los otros productos ya mencionados, tiene el propósito de servir para administrar los procesos.

Existen muchas otras herramientas que permiten implementar y administrar procesos ETL, pero la intención era simplemente nombrar algunas de ellas y sus características, para resaltar la idea de que ante el momento de elegir se tienen muchas opciones, la mayoría de similares características, por lo que se podría escoger cualquiera de ellas. En este caso se escogió *Oracle Warehouse Builder* básicamente porque ofrece todas las características que poseen las herramientas ya descritas, pero principalmente debido a que la compañía trabaja con los productos Oracle en todas sus líneas, lo que hace que no existan problemas para trabajar con los datos en ningún momento.

Algunas otras herramientas que existen en el mercado y de las cuales no se explicarán más detalles por los motivos ya mencionados son las siguientes:

- IBM DB2 Warehouse Manager.
- Pervasive Data Integrator.
- BusinessObjects Data Integrator.

- HummingBird Genio.
- Embarcadero DT/Studio.

Todas estas herramientas que se utilizan para el desarrollo de procesos ETL y que trabajan con la generación de metadatos, usan un estándar de la industria para ello. Se trata de *Common Warehouse Meta-model* (CWM), arquitectura creada por OMG (*Object Management Group*).

El objetivo principal de CWM es permitir el fácil intercambio metadato de warehouse y BI entre herramientas, plataformas y repositorios de metadato de warehouse en distintos ambientes. CWM se basa en tres estándares de la industria:

- UML (Unified Modeling Language)
- MOF (*Meta Object Facility*)
- XMI XML Metadata Interchange

Estos tres estándares forman el núcleo de la arquitectura del repositorio de metadato de OMG como se muestra en la figura 3.10:



Figura 3.10: Arquitectura de metadato de la OMG.

El estándar UML define un lenguaje de modelamiento orientado a objetos que es soportada por una gran cantidad de herramientas de diseño gráficos. El estándar MOF define una estructura para la definición de modelos de metadatos y provee de las herramientas con interfaces programables para almacenar y acceder al metadato en un repositorio. El estándar XMI permite a los metadatos intercambiarse como flujos de información o archivos con formato estándar basados en XML.

No se encuentra en el alcance de esta tesis definir esta arquitectura de trabajo, ya que se desviaría del objetivo principal que se quiere conseguir, pero si se desea obtener más información, se encuentra toda disponible en la página web del OMG.

# CAPÍTULO 4: IMPLEMENTACIÓN DE LA SOLUCIÓN

# 4.1 OBJETIVOS DEL CAPÍTULO

Durante este capítulo se mostrará el desarrollo completo de la solución de BI implementada para la empresa Salmofood S.A. En la primera parte se detallará lo que significa el desarrollo de este proyecto usando conceptos de Ingeniería de Software. Posteriormente, se dará énfasis a las etapas de diseño, codificación y prueba del sistema, vale decir, lo que para este proyecto son el desarrollo de procesos ETL para la carga del DM de compras. También se mostrará la implementación de uno de los informes en su totalidad, ya que mostrar más informes sería prácticamente lo mismo para todos.

#### 4.2 IMPLEMENTACIÓN DE SISTEMA DE BI

Para implementar un sistema de BI que tenga éxito a la hora de ser usado como herramienta de apoyo a la toma de decisiones es necesario seguir al menos los siguientes pasos:

- Investigar los requerimientos de los usuarios.
- Estudiar los sistemas de información desde los cuales extraer la información requerida.
- Selección de tablas a utilizar.
- Creación de un nuevo modelo de datos utilizando algún esquema de DW.
- Diseñar e implementar los procesos ETL que poblarán la estructura de datos.
- Utilizar herramientas de consulta para explotar los datos extraídos.

Es éste el esquema de trabajo que se siguió para desarrollar el sistema de BI para el área de compras. Sin embargo, todo desarrollo de un proyecto de software, debe ir respaldado por un proceso que conduzca a la entrega de un producto de calidad, lo cual se logra utilizando las técnicas que posee la ingeniería de software. Es necesario tener claro lo que se quiere desarrollar y el resultado que se espera obtener una vez que se termine el producto de software.

## 4.2.1 CICLO DE VIDA DEL PROYECTO

El ciclo de vida de un proyecto de software son los estados por los que pasa el producto desde su inicio hasta que se completa el software. En general, las etapas por las que pasa el desarrollo de un producto, son las siguientes: necesidad, especificación de requisitos, diseño, código y sistema de software. Existen varios modelos de ciclo de vida que son usados para desarrollar un proyecto, como por ejemplo: cascada, prototipo, espiral o el modelo en V, cada unos con sus ventajas y desventajas. Para el caso que se está desarrollando, se utilizará el modelo de ciclo de vida en cascada, debido a las características que posee este tipo de modelo.

En la implementación del DM para el área de compras, los requisitos pueden ser conocidos en la primera etapa del proyecto sin problemas, ya que no son demasiados los informes que se generan en esta área. Al conocer perfectamente los requisitos, se puede ir avanzando en el ciclo de vida y verificar en cada etapa que el proyecto va efectivamente bien encaminado a lograr una correcta implementación. También está claro que no se puede mostrar nada al usuario hasta que se ha llegado a la etapa de los informes, lo cual es el final del desarrollo del DM.

## 4.2.2 ANÁLISIS DE RIESGO

Una etapa importante en el desarrollo de cualquier proyecto de software es la estimación y el control del riesgo que puede afectar al producto, de forma de poder

controlar su impacto en el caso de que se concreten, ya sea durante el desarrollo o en las etapas posteriores.

Uno de los escritores famosos en lo que al análisis de riesgo se refiere es el Dr. Barry Boehm. En su artículo "*Software Risk Management: Principles and Practices*" [4] afirma que el manejo del riesgo se comprende la dos actividades principales:

- Estimación del riesgo, que consiste en identificar cuáles son los riesgos y en qué enfocarnos. Dentro de este punto se recomienda hacer una lista de todos los peligros potenciales que puedan afectar al proyecto, estimar la probabilidad de ocurrencia para cada riesgo y finalmente realizar un ranking de toda la lista desde al más peligroso al menor de ellos.
- Control de riesgo, que consiste en especificar qué medidas tomar en caso de que el riesgo efectivamente se presente. Dentro de este punto se recomienda evaluar técnicas o estrategias para mitigar los riesgos evaluados como más peligrosos y monitorear la efectividad de las estrategias y los cambios en los niveles del riesgo a través del proyecto.

A continuación se muestra una lista ya priorizada con los riesgos que se han detectado para el desarrollo del proyecto de software que se está tratando. Las probabilidades de ocurrencia para cada riesgo identificado se fija de acuerdo a la tabla 2 del artículo del Dr. Boehm, en la cual fija una cuantificación de la probabilidad y el impacto en costo para la falla.

• No se tienen los recursos de hardware para implementar el proyecto. La probabilidad de que ocurra este evento es de 0,5, ya que existen algunas

restricciones de hardware para que el proyecto pueda ser implementado. Este riesgo se podría presentar en el inicio del proyecto, ya que una vez que se tiene todo el equipo para implementarlo se puede comenzar a trabajar. Una vez que se ha iniciado el proyecto, quiere decir que ya se tienen los productos de desarrollo instalados y ya se han superado los requerimientos mínimos de instalación para todos ellos. Es por esto que no existe control para este riesgo, de presentarse significaría que no se puede implementar una solución BI, por lo que el impacto de este riesgo es catastrófico para el proyecto.

• Las personas entrevistadas para obtener los requisitos no eran las más idóneas. La probabilidad de que ocurra esto se ha fijado en 0,2, puesto que el personal que trabaja en la empresa cuenta con bastante experiencia usando su ERP durante 4 años y conoce muy bien los informes que le son útiles o los nuevos que necesitan de forma mucho más rápida que lo que tienen actualmente. Para evitar la presencia de este riesgo será necesario realizar entrevistas desde el gerente del área hacia abajo, evaluando factores como la cantidad de aplicaciones que utiliza cada usuario, a qué módulos del ERP tiene acceso, las funciones que desempeña y tiempo que lleva desempeñando sus funciones, de forma de tener una idea sobre el grado de familiaridad que tiene con el sistema que usa a diario y de esta forma poder recoger los requerimientos de las personas que están más familiarizadas con el ERP y los datos que son necesarios explotar. En caso de presentarse este riesgo, será necesario entrevistar a otras personas del área para evaluar si los datos proporcionados por sus compañeros son válidos o se necesita generar los

informes que se están solicitando. Es importante evaluar a las personas que serán las que ayudarán en el desarrollo del proyecto cuando éste está comenzando, puesto que las pruebas serán realizadas una vez que ya todo esté implementado y si no se han entrevistado a las personas adecuadas, los resultados finales no serán los esperados por el personal de la empresa y se tendría que rehacer el proyecto desde la etapa de diseño. La realización de nuevas entrevistas podría retrasar el proyecto desde 1 a 2 semanas, dependiendo de la disponibilidad de tiempo de las personas a las cuales se desea entrevistar.

Incompatibilidad de versiones en los productos usados. La probabilidad de que ocurra este evento se ha fijado en 0,2, ya que antes de iniciar el proyecto se debe evaluar qué productos se utilizarán y ver la compatibilidad de las versiones, para lo que existe la matriz de compatibilidad entregada por Oracle. En caso de existir algún problema se debe evaluar la posibilidad de aplicar algún parche para actualizar el producto que esté presentando problemas o bien de bajar la versión de otro producto si fuese necesario. El impacto que tendría un cambio como este en el proyecto que se está implementando no es demasiado alto, puesto que la aplicación de estos parches cuando se trata de aplicaciones sin datos o casi vacías no requiere de tantas acciones de ejecución post-instalaciones, como sería regenerar todos los procesos ETL existentes por dar un ejemplo. La presencia de este riesgo podría retrasar el proyecto alrededor de 2 semanas.

- *No existe compromiso por parte de la empresa para realizar las pruebas del producto.* Para este evento, la probabilidad de ocurrencia se ha fijado en 0,1, ya que es muy poco probable que esto ocurra, debido a que se ha manifestado que el proyecto tiene que funcionar, avalado por la nueva gerencia y el directorio de la compañía. En caso de que se presente alguna circunstancia en la que el apoyo de la empresa a la hora de realizar las pruebas del sistema, será necesario evaluar los resultados con algunas de las personas que estuvieron involucradas en la especificación de requisitos dependiendo de la disponibilidad de cada una de ellas. El impacto que se presenta ante este evento es que se retrasaría el proyecto entre 2 a 3 semanas más de lo esperado.
- Los usuarios no quieren usar el producto. La probabilidad para que este evento se manifieste es de 0,3, ya que la empresa se encuentra con todas las intenciones de que se use el sistema. Si el motivo por el cual los usuarios no quieren usar el sistema es que el sistema es poco amigable, se deben realizar las modificaciones necesarias para que el acceso al producto final sea rápido y simple, de forma que no resulte tedioso para el personal utilizar la nueva aplicación que se les presenta. El impacto que implica este evento, en caso de que le motivo por el cual se presenta sea que el sistema es poco amigable sería de 4 semanas, de forma de mejorar el formato de los informes para facilitar su entendimiento.

# 4.2.3 GESTIÓN DE CONFIGURACIÓN

La gestión de configuración se puede definir como identificar, organizar y controlar las modificaciones en un producto de software que está siendo construido o mantenido por un equipo de personas de modo de minimizar la confusión.

La gestión de configuración es muy útil cuando se encuentra un equipo de gente trabajando sobre un proyecto de desarrollo de software, en el cual existe información compartida con más de una persona accediendo y modificando la información, los archivos están siendo modificados por más de un programador, con lo que pueden aparecer errores que ya han sido corregidos o perder algunos cambios realizados. Entre sus funciones se encuentra identificar cada objeto producido en el proceso (programa, documento, dato) y la versión de éste. También controla el acceso a un objeto, de forma de poder controlar los cambios simultáneos cuando existe más de una persona tratando de realizar cambios sobre un objeto. Para el actual caso de estudio, no tiene mucho sentido utilizar las técnicas que se aplican a la gestión de la configuración, ya que solamente existirá una persona que se encuentra con el control total sobre el proyecto, por lo cual no se producirían los problemas que esta disciplina cubre. Es por esto que no se usará la gestión de configuración en el desarrollo del DM de compras. Sin embargo, es importante mencionar que existen estándares que tratan este tema con profundidad, como por ejemplo: IEEE 1042-1987 o la norma ISO 10007:2003.

# 4.2.4 GESTIÓN DE LA CALIDAD, CONTROL Y GARANTÍA

En el desarrollo de cualquier producto de software, el aseguramiento de la calidad de éste es una tarea importante a realizar. Es importante tener clara tareas

como son las de definir: un modelo de calidad a utilizar, actividades de control y de garantía de la calidad, todo esto unido dentro de un sistema de gestión de la calidad.

# 4.2.4.1 DEFINICIÓN MODELO DE CALIDAD

Existen varios modelos de calidad a utilizar para el desarrollo de un producto de software, pero sin duda alguna, el más conocido de ellos es el de McCall. El modelo que desarrolló McCall contempla la existencia de 3 ejes o puntos de vista del software, con los cuales se forma el triángulo de McCall. Para evaluar estos 3 ejes, aparecen 11 factores de calidad. Todo esto puede verse apreciado en la figura 4.1



#### Figura 4.1: Triángulo de McCall.

Los 11 factores que aparecen en la figura 4.1 se descomponen a su vez en 23 criterios. Para utilizar el modelo de McCall se debe tener en claro cuáles serán los requisitos de calidad del producto. Debido a las características del producto que se desarrollará, se pueden priorizar los 11 factores de la siguiente forma:

- Corrección. Este factor se califica como muy alto, ya que se espera que los requerimientos especificados por los usuarios se cumplan en su totalidad, puesto que es muy importante que se entregue lo que el usuario necesita, con los cálculos que requiere y no con valores incorrectos.
- Fiabilidad. Este factor se califica como muy alto, puesto que los datos entregados deben ser exactos, ya que las decisiones que se tomarán a partir de los informes generados pueden afectar fuertemente a la empresa.
- Eficiencia. Este factor se califica como bajo, puesto que la mayoría del desarrollo se realiza a través de asistentes, los cuales generan solos el código necesario para realizar todas las funciones que se requieran.
- Integridad. La integridad se califica como de nivel medio, puesto que si bien no puede permitirse que personas que no tienen nada que ver con la toma de decisiones tengan acceso a la información que se entregará, para este caso particular, no se trabaja con datos que sean tan sensibles para la gerencia de la empresa, como pueden ser por ejemplo, los estados financieros de la compañía.
- Facilidad de uso. Este factor se ha calificado como bajo, puesto que los usuarios ya tienen experiencia usando los productos de Oracle y están familiarizados con su entorno, por lo que no les será muy difícil de aprender a utilizar el nuevo sistema que se quiere implantar. Por este motivo, se estima que no serán necesarios cursos de capacitación externos.
- Facilidad de mantenimiento. Se ha calificado a este factor como de nivel medio, ya que se necesita que se puedan realizar las modificaciones a los

procesos ETL o los informes que se generen, todo esto en el menor tiempo posible, de forma que los costos de tener el sistema sin operar sea lo más bajo posible.

- Facilidad de prueba. Se ha dado el nivel de medio a este factor, ya que se necesita que se pueda probar el producto y que el hecho de estar realizando pruebas no sea tan costoso para la compañía.
- Flexibilidad. Este factor se ha calificado como bajo, ya que los productos que se ocupan para el desarrollo del DM de compras ofrecen la posibilidad de realizar las modificaciones que sean necesarias en el menor tiempo posible, generando informes sobre el impacto de los cambios que se están realizando.
- Portabilidad. Se ha calificado como de nivel bajo, sobre la base de que los productos de software Oracle, en las versiones que se trabajan, ofrecen la posibilidad de generar *scripts* que son multiplataformas, con lo que se podrían llevar a la aplicación a otro ambiente sin mayores problemas.
- Facilidad de reutilización. Se ha fijado este factor como de nivel bajo, basados en la misma justificación que el punto anterior.
- Interoperatibilidad. El nivel para este factor se ha dejado como bajo, ya que los costos para hacer que el nuevo sistema funcione bien con los ya existentes es muy bajo, debido a la total compatibilidad que existe entre las aplicaciones de la empresa, al ser todas pertenecientes a Oracle.

Tras clasificar los factores, se puede notar que los que tienen una mayor importancia para el desarrollo del proyecto, por las características que posee, son las siguientes: corrección, fiabilidad, integridad, facilidad de prueba y facilidad de mantención.

Siguiendo la metodología que se usa para utilizar el modelo de McCall, una vez que se han identificado los factores de calidad que serán claves para el desarrollo del producto de software y se han ordenado de acuerdo a su importancia, es necesario evaluar dichos factores según los criterios que tienen asociados.

- Corrección. El factor de corrección tiene asociado tres criterios para poder evaluarlo, los cuales son: completitud, consistencia y trazabilidad. Se suman los valores para los tres criterios mencionados y se divide el resultado por 3. Este cálculo entrega un valor entre 0 y 1. Para la corrección se obtiene finalmente un valor de 0,86.
- Fiabilidad. Este factor tiene asociado cinco criterios para poder realizar una evaluación sobre él, los cuales son: precisión, consistencia, tolerancia a fallos, modularidad y simplicidad. Para la fiabilidad se ha obtenido un valor de de 0,99.
- Integridad. Los criterios asociados para poder realizar una evaluación sobre este factor son los siguientes: control de accesos y facilidad de auditoría. Para la integridad se ha obtenido un valor de 0,81.
- Facilidad de prueba. Los criterios que influyen en la evaluación de este factor son los siguientes: modularidad, simplicidad, auto descripción e instrumentación. Para este factor se ha obtenido un valor de 0,77.

• Facilidad de mantenimiento. Sobre este factor influyen los siguientes criterios: modularidad, simplicidad, consistencia, concisión y auto descripción. Para este factor se ha obtenido un valor de 0,90.

## **4.2.4.2 CONTROL DE LA CALIDAD**

El objetivo principal que persigue el control de la calidad del software es comprobar si el producto posee o no determinadas características de calidad en el grado requerido. Si no la posee quiere decir que existe un defecto en el producto, para lo cual se trata de corregir esto. Las actividades de control de la calidad se pueden apreciar en la figura 4.2.



Figura 4.2: Tipos de actividades de control.

Para tener un control sobre la calidad del producto que se está desarrollando se usará lo que son los controles estáticos, del tipo manuales y disciplinados, específicamente, las auditorías. Mediante las auditorías se pretende certificar la conformidad de los usuarios con el producto que se desarrolla. Existen diversos tipos de auditorías a realizar, como son:

- Auditoría del producto, cuyo objetivo es cuantificar el grado de conformidad del producto con las características requeridas.
- Auditoría del proceso, cuyo objetivo principal es el de evaluar el proceso de desarrollo o de gestión, y evaluar completitud y efectividad, determinando dónde se puede mejorar.
- Auditoría del sistema de calidad, en el que el objetivo es evaluar la completitud y efectividad del propio sistema de calidad establecido.

Para el caso particular que se está desarrollando se realizarán auditorías del producto. Esta auditoría será llevada a cabo después de la etapa de diseño del sistema y finalmente después de las pruebas funcionales del mismo. La forma en que se realizarán estas auditorías es mediante una reunión con las personas directamente implicadas en el proyecto que se está desarrollando, es decir, las siguientes: personal del área de compras, jefe del departamento de Informática y quien está desarrollando el proyecto. En estas reuniones se tendrá preparado un cuestionario para que las personas involucradas reflejen su nivel de satisfacción con el producto que se les está mostrando. Como una característica de las auditorías, los participantes pueden dar sus opiniones abiertamente para sugerir soluciones a los problemas que hayan encontrado. Luego de finalizadas y de analizar los resultados obtenidos de las encuestas, se elaborarán los informes correspondientes y se entregarán a sus participantes.

# 4.2.4.3 GARANTÍA DE LA CALIDAD

Se puede definir la garantía de calidad como todas las actividades que se realizan con el objetivo de asegurar un nivel de calidad sobre el producto desarrollado. Cuando los proyectos de software son grandes, existen equipos que se encargan de la garantía de calidad, pero en casos como el que se está desarrollando no será necesario, quedando esta tarea como otra actividad de quien está desarrollando.

Las principales tareas de un grupo de garantía de la calidad son:

- Planificación de la calidad.
- Supervisión de la calidad.
- Construcción de la calidad.

La garantía de calidad supervisa el cumplimiento de la política de calidad adoptada. Como en el desarrollo de este proyecto no interviene más de una persona, suena ilógico realizar más controles sobre uno mismo que los que ya se han previsto al fijar la política de calidad. Se hará uso solamente de las auditorías y se seguirán las etapas del ciclo de vida del producto como ya ha sido definido.

## 4.2.4.4. SISTEMA DE GESTIÓN DE LA CALIDAD

El sistema de gestión de la calidad define de qué forma se reparten las tareas para hacer efectiva la garantía de calidad. Es por esto que no se tomará en cuenta este sistema de gestión para el caso particular que se está desarrollando, basado en lo mismo que se explicó para el caso de la garantía de calidad. El sistema de gestión de la calidad abarca todo lo que a la calidad del software se refiere y la forma en que se alcanzará. Es por esto que incluye la definición de la política de la calidad y el control sobre ésta.

# 4.2.5 ESPECIFICACIÓN DE REQUISITOS

Esta es sin duda alguna una de las etapas más importantes en lo que al desarrollo de software se refiere es realizar una correcta especificación de los requisitos, de forma que las etapas posteriores se basen en lo que el cliente realmente desea. Realizar cambios a las especificaciones en etapas avanzadas al proyecto puede ser muy costoso para el proyecto, ya que puede llegar a significar, en caso extremo, que se debe comenzar todo nuevamente. Es por todo esto que lo que a requisitos se refiere, que ha captado la preocupación de muchas personas e incluso han llegado a lo que es la Ingeniería de Software.

En la especificación de requisitos se identifica a los interesados en el producto y se establecen las relaciones con el equipo de desarrollo. Para el caso actual, los interesados son: las personas que trabajan en el área de compras de la empresa y el jefe del área de operaciones. También existen otras personas que pueden beneficiarse del uso de los informes que se generarán, pero de los cuales no será necesario obtener sus requerimientos, ya que el personal de compras es el que maneja toda la información que se trabajará.

Existen varias técnicas para obtener una especificación de requisitos, entre las que están: tormenta de ideas, entrevistas con los usuarios y la observación y análisis de ideas, por nombrar algunas. En el caso que se está desarrollando se utilizará el método más tradicional, es decir, las entrevistas. También se utilizará lo que es la observación del sistema y el análisis de tareas, de forma de poder complementar las entrevistas.

En una primera etapa, la observación del sistema será muy importante, de forma de familiarizarse con el sistema que maneja la empresa, los informes que utilizan y los que se pueden necesitar. Una vez que ya se tiene un conocimiento y manejo sobre el sistema, se pueden preparar entrevistas para conversar con las personas que están directamente involucradas en el tema que se desarrolla, de forma de capturar todos los requerimientos que debe satisfacer el producto. Se deben analizar cuidadosamente todos los requisitos obtenidos, de manera que se puedan detectar posibles conflictos entre requisitos.

Con los requisitos obtenidos y correctamente analizados se genera un documento que contiene todo lo pactado entre las partes acerca de lo que deberá satisfacer el producto. Cómo redactar este documento o qué es lo que tiene que incluir se puede encontrar en el estándar IEEE 830-1998, el cual es el formato de Especificación de Requisitos de Software.

En el anexo F se puede encontrar el documento de especificación de requisitos claramente detallado según el estándar ya mencionado, el cual fue generado luego de haber observado el sistema y concluidas las entrevistas con las personas que están directamente relacionadas con el desarrollo del proyecto.

#### 4.2.6 ESTIMACIÓN DEL PROYECTO

La estimación es una de las tareas críticas en la gestión de un proyecto. La estimación debe responder a interrogantes como ¿cuánto costará? o ¿cuánto tiempo llevará hacerlo? Para realizar una estimación sobre un proyecto de software, se

recomienda que la persona que la realice deba ser un profesional que no tenga ningún interés en los resultados del proceso de estimación.

La estimación es un proceso que debe realizarse en varias ocasiones durante el ciclo de vida de un proyecto, de forma de que la exactitud del valor estimado obtenido sea cada vez más cercano al valor real. Aunque se recomienda que la estimación sea un proceso continuo y sea realizada entre cada etapa del ciclo de vida del proyecto, para el caso particular que se está desarrollando se hará solamente una estimación, luego de la especificación completa de los requisitos. En las etapas posteriores del ciclo de vida, como son: diseño, código, prueba, implantación y uso del sistema no se realizarán más estimaciones. Para este caso particular solamente se efectuará una estimación de modo de ilustrar su uso y relacionarlo con el proyecto, ya que como existe solamente una persona involucrada en el proyecto, no se cumple la recomendación hecha sobre que un estimador no debe tener intereses en los resultados del proyecto.

La estimación se realiza en base a métricas, las cuales son usadas para estimar los costos y tamaños del producto a desarrollar. En el mercado existen varios paquetes diseñados para ayudar en la estimación de proyectos de software, como son por ejemplo: COCOMO, ESTIMACS y SOFTCOST. Sin duda que el más conocido y utilizado de todos es el modelo de COCOMO, desarrollado en 1981. En la actualidad ya se está utilizando el modelo COCOMO II-1999.0. Este será el modelo que se utilizará para realizar la estimación en el actual proyecto en desarrollo. Para realizar la estimación se utilizará el software disponible en Internet, COCOMO II 1999.0, distribuido por la Universidad de Carolina del Sur. Lo que se estimará con este software será lo siguiente: tamaño del proyecto, riesgo asociado, esfuerzo requerido y el costo. Se obtendrán tres tipos de estimaciones: pesimista, optimista y realista.

Antes de realizar la estimación del proyecto usando el software, es necesario configurar los siguientes parámetros que se utilizarán para realizar la estimación: factor de escala, programación y modelo de desarrollo. Cabe mencionar que no se explicará la forma de ponderar estos valores, sino que se mostrarán los valores obtenidos. Cómo ponderar cada valor que se utilizará para efectuar las estimaciones no será ilustrado en esta tesis y solamente se mostrarán los valores que fueron asignados a cada parámetro.

Existen dos modelos de desarrollo que se pueden utilizar, que son el diseño anticipado y el de post-arquitectura. La elección de un modelo u otro afecta a los valores obtenidos para las estimaciones pesimista y optimista. En este caso se utilizará el modelo post-arquitectura, ya que se conoce bastante estimación acerca del proyecto a desarrollar.

El factor de escala se calcula en base a 5 parámetros más, los cuales son los siguientes: precedencia, flexibilidad de desarrollo, arquitectura/riesgo, cohesión del equipo de desarrollo y madurez del proceso. Los valores que se asignan a cada uno de estos parámetros van desde muy bajo a extra alto, y el software entrega valores ya calculados de acuerdo a lo que se escoja. Para el cálculo del factor de escala se ha obtenido lo siguiente:

- **Precedencia**: Valor alto.
- Flexibilidad de desarrollo: Valor nominal.

- Arquitectura / Riesgo: Valor nominal.
- Cohesión del equipo: Valor alto.
- Madurez de proceso: Valor de nominal.

En la figura 4.3 se muestra cómo se configuraron los parámetros para el cálculo del factor de escala.

Scale Factors		×
Precedentedness	HI	2.48
Development Flexibility	NOM	3.04
Architecture / risk resolution	NOM	4.24
Team cohesion	HI	2.19
Process maturity	NOM	4.68
OK Cancel	Help	

Figura 4.3: Factores de escala.

En el caso de la programación se ha escogido un valor de alto. Esto puede

apreciarse en la figura 4.4.

Schedule	
Schedule	HI 0% 1.00
OK Cancel	Help

Figura 4.4: Parámetro de programación.

Para este proyecto se creará un módulo, el cual es el DM de compras, dentro del cual se configurarán otros parámetros que ayudarán a efectuar las estimaciones del proyecto en cuanto al tamaño, costo, personal y duración. Luego de crear el módulo y asignarle un nombre para efectos de informes del software de estimación, se realiza el primer cálculo, que corresponde al tamaño del módulo (que para este caso particular resulta ser el tamaño del proyecto).

Como ya se tiene bastante información acerca de lo que se elaborará, se puede calcular con una amplia certeza esta parte del proyecto, al conocer valores aproximados sobre parámetros como: número de entradas, salidas, interfaces, archivos y consultas. En la figura 4.5 se muestran los resultados obtenidos para el tamaño del módulo, basado en el método de los puntos de función y suponiendo que el lenguaje de programación es un Lenguaje de Consultas.

SLOC Input Dialog - Data Mart Compras										
Sizing Method Breakage										
C SLOC		% of code throw			vn away due					
• Function Performance	oints	to requirements			volitility					
C Adaptation					BRAK 0.00					
Module Size in Function Points										
	Langua	ge 💌	Query	7 language	2					
E.		#ofF	unction	Points						
Fur	iction 1 ype	e Low Average High			SubTotal					
			-	-	243					
	npuis	15	18	21	240					
C	Jutputs	170	180	200	2980					
F	ïles	10	12	15	415					
I	nterfaces	10	12	15	284					
	Jueries	12	15	20	216					
To	otal Unadjuste	4138								
Ea	Equivalent Total in SLOC									
Equivalent Total In SLOC 55/94										
······										
	OK	Cance	el		Help					

Figura 4.5: Cálculo de tamaño del módulo.

El siguiente valor que es necesario configurar en el software es la columna de tarifa de trabajo, en la cual se especifica la cantidad de dinero que se le podría pagar a un desarrollador por mes para trabajar sobre un módulo. En este caso particular el valor ha sido de \$30.000 al mes.

A continuación se debe modificar el valor para la columna de EAF (*Effort Adjusment Factor* o Factor de Ajuste de Esfuerzo). Para el cálculo del EAF se evalúa el sistema desde cuatro perspectivas: producto, plataforma, personal y proyecto. Cada una de estas perspectivas tiene sus parámetros de forma de obtener un valor ponderado para el EAF.

Para el caso del producto se tienen los siguientes parámetros: confiabilidad requerida por el software, tamaño de la base de datos, documentación, complejidad del producto, reusabilidad requerida.

Para el caso de la plataforma los parámetros son los siguientes: restricción de tiempo de ejecución, restricción de almacenamiento principal y volatibilidad de la plataforma.

Para el caso del personal se evalúa lo siguiente: capacidades del personal, experiencia en aplicaciones, capacidades del programador, experiencia en la plataforma, experiencia en el lenguaje de programación y continuidad del personal.

Para los atributos del proyecto se evalúa el uso de las herramientas de software y desarrollo en varios lugares.

Todos estos factores se incluyen en el cálculo para la estimación de la programación y el esfuerzo realizada por el software de COCOMO. En la figura 4.6

se pueden ver los valores obtenidos para cada uno de los parámetros que influyen en el cálculo del EAF.

EAF - Data Mart Compras								
	base	+ Incr	: * = r	ating				
Product:	RELY	DATA	DOCU	CPLX	RUSE			
base	LO	NOM	VLO	VLO	NOM			
Incr*	0%	0%	0%	0%	25%			
Platform:	TIME	STOR	PVOL					
base	HI	NOM	NOM					
Incr*	0%	0%	0%					
Personnel:	ACAP	AEXP	PCAP	PEXP	LTEX	PCON		
base	LO	NOM	LO	NOM	HI	VHI		
Incr*	0%	0%	25%	0%	25%	0%		
Project:	TOOL	SITE						
base	LO	NOM						
Incr*	25%	0%						
User:	USR1	USR2						
base	NOM	NOM						
Incr*	0%	0%						
EAF is also affected by Schedule								
EAF: 0.63								
OK Cancel Help								

Figura 4.6: Parámetros para cálculo del EAF.

Ya con todos estos parámetros se puede obtener una estimación del proyecto de software. Los resultados obtenidos son los siguientes:

- Tamaño del módulo: 53794 líneas de código.
- Esfuerzo sin incluir el EAF en su cálculo es de 214.4 personas-mes.
- Esfuerzo en personas-mes incluyendo el EAF para el cálculo: 134.6 personasmes.
- Productividad: el número de líneas de código fuente que debiera escribir cada persona al mes es de 399.8.

- El costo obtenido para el desarrollo completo del módulo es de \$4.036.627,06.
- El costo por instrucción es de \$75.
- El número de personas para desarrollar el módulo es de 6,1.

Todo lo anterior puede verse en la figura 4.7.

	LABOR		NOM	EST				
Module	Rate		Effort	Effort			INST	
Size	(\$/month)	EAF	DEA	DEV	PROD	COST	COST	Staff
7. 50504	00000 00	0.00	014.4	104 0	000.0	400/200 02	25.0	<i>C</i> 1

Figura 4.7: Resultados de estimación obtenidos para el módulo.

Para el proyecto, que como ya se mencionó, consiste solamente de un módulo, se obtienen obviamente los mismos resultados que se han mostrado, pero además se agregan valores estimados para una situación optimista y pesimista. En la figura 4.7 se pueden apreciar los resultados obtenidos.

En la figura 4.8 se pueden apreciar tres filas con resultados, que representan las estimaciones optimistas, realistas y pesimistas. En la primera columna, se obtiene el esfuerzo, que representa la cantidad de personas-mes que tomaría realizar el proyecto. Se tiene que los valores obtenidos son 107.6, 134.6 y 168.2 para las estimaciones optimistas, realistas y pesimistas respectivamente.

Estimated	Effort	Sched	PROD	COST	INST	Staff
Optimistic	107.6	20.7	499.7	3229301.65	60.0	5.2
Most Likely	134.6	22.2	399.8	4036627.06	75.0	6.1
Pessimistic	168.2	23.8	319.8	5045783.82	93.8	7.1

Figura 4.8: Resultados de estimación obtenidos para el proyecto.

En la segunda columna de la figura 4.8 se tiene el tiempo en semanas que se demoraría el proyecto en ser realizado. Los valores obtenidos fueron 20.7, 22.2 y 23.8 para las estimaciones optimistas, realistas y pesimistas respectivamente.

En la tercera columna de la figura 4.8 se obtiene la productividad en líneas de código que debiera tener cada persona al mes para realizar el proyecto. Los valores obtenidos son de 499.7, 399.8 y 318.8 para las estimaciones optimistas, realistas y pesimistas respectivamente.

En la cuarta columna de la figura 4.8 se obtiene el costo mensual para poder solventar el proyecto y pagarle a las personas que trabajan en él. Los valores obtenidos fueron 3229301.65, 4036627.06 y 5045783.82 para las estimaciones optimistas, realistas y pesimistas respectivamente.

En la quinta columna de la figura 4.8 se obtiene el costo por cada instrucción escrita. Los valores obtenidos fueron de 60, 75 y 93.8 para las estimaciones optimistas, realistas y pesimistas respectivamente.

En la sexta columna de la figura 4.8 se obtiene el número de personas que debiera estar integrada al proceso de desarrollo del producto de forma de realizar el proyecto en el tiempo estimado. Los valores obtenidos fueron de 5.2, 6.1 y 7.1 para las estimaciones optimistas, realistas y pesimistas respectivamente.

Todos estos valores obtenidos fue de una primera estimación hecha por el proyecto, la cual se puede ir acercando cada vez más a la realidad a medida que se avance en el ciclo de vida del proyecto, puesto que se conocerán más detalles sobre él. Ya se ha explicado que no se realizarán más estimaciones sobre el proyecto en
desarrollo y que esta estimación tenía solamente la finalidad de servir para ilustrar el modo de realizar una estimación usando el modelo COCOMO II 1999.

## 4.2.7 DISEÑO DEL SISTEMA

Luego de que los requisitos ya han sido recopilados y se ha hecho el respectivo análisis para detectar posibles conflictos entre ellos se pasa a la siguiente etapa en el ciclo de vida del proyecto, la cual es el diseño del sistema. En esta etapa se diseñará la estructura de datos que soportará toda la información que se necesite para poder satisfacer todos los requerimientos de los usuarios, para que a partir de ahí se pueda comenzar a implementar el sistema que se desea.

Tras analizar cuidadosamente los requisitos se ha podido obtener el origen de todos los datos necesarios para generar los informes que los usuarios han solicitado. Se ha identificado cada tabla, columna y esquema desde donde se requiere información, por lo solamente resta formar las tablas de hechos y las tablas de dimensión necesarias para formar un modelo adecuado (según la topología estrella) que soporte todos los datos que se cargarán a la base de datos.

La tabla de hechos de OC que se ha diseñado se puede apreciar en la figura 4.9.



Figura 4.9: Tabla de hechos de OC.

Con la tabla de hechos que se ve en la figura 4.9 servirá para dar respuesta a los siguientes informes:

- OC por proveedor al mes.
- OC por Centro de Costo al mes.
- OC abiertas por proveedor.
- OC canceladas por comprador.
- OC canceladas por proveedor.
- Registro de actividad de compras.
- Análisis de ahorro por categoría.
- Registro de transacciones de recepción.

La tabla de hechos de envíos vencidos que se ha diseñado se puede apreciar en la figura 4.10.



Tabla 4.10: Tabla de hechos de envíos vencidos.

Con la tabla de hechos de envíos vencidos se dará respuesta al informe de envíos vencidos por proveedor.

La tabla de hechos de solicitudes que se ha diseñado se puede ver en la figura 4.11.



Figura 4.11: Tabla de hechos de solicitudes.

Con la tabla de hechos de solicitudes se dará respuesta a los siguientes informes:

- Registro de actividad de solicitud.
- Informe de solicitudes canceladas.

Usando los datos que serán cargados para responder los informes mencionados se generará el informe de solicitudes versus OC filtrado por solicitante.

## 4.2.8 INSTALACIÓN Y CONFIGURACIÓN DE HERRAMIENTAS

Luego de conocer de dónde extraer cada dato para generar todos los informes y ya definido el modelo de datos en el que se almacenará la información se deben instalar y configurar todas las herramientas que se utilizarán para el desarrollo del DM de compras.

La primera herramienta a configurar es el servidor de base de datos, el cual es Oracle Database 9.0.1.4.0. Los pasos a seguir para instalar y configurar la base de datos, de forma que sirva para almacenar el DM se explica con detalle en el anexo B del presente documento.

Resumiendo algunos de los detalles acerca de la creación de la base de datos a utilizar se puede extraer lo siguiente:

- El nombre de la instancia creada es BDWH, a la cual se accede a través del protocolo TCP/IP a la máquina Basilisco.
- El usuario con el privilegio de ser SYSDBA que se ha creado para configurar lo que sea necesario más adelante y que requiera de permisos especiales es "mauricio".
- Se ha dejado configurado el OMS de forma que se puedan programar y ejecutar los trabajos que realizarán la carga de datos. El sistema operativo ha sido configurado para que se puedan ejecutar estos trabajos programados.

Cuando la base de datos ya ha sido correctamente instalada y configurada (siguiendo las especificaciones detalladas en el Anexo B) se debe continuar con la instalación de la herramienta que será utilizada para la creación de los procesos ETL, es decir, el OWB.

La instalación y configuración del OWB se encuentra detallada en el Anexo C del presente documento.

Para resumir la instalación que fue hecha (siguiendo la pauta entregada en el Anexo C) se puede destacar lo siguiente:

- Se han creado 4 usuarios de base de datos:
  - adminwh: Usuario administrador del Design Repository.
  - runtimewh: Usuario dueño del esquema del repositorio de *Runtime*.
  - **runtimeac**: Usuario de acceso al esquema del repositorio de *Runtime*.
  - targetwh: Usuario del esquema en que se guardan los datos finales.
- La instalación en el equipo Cliente ha sido correctamente realizada y siguiendo las advertencias a las cuales se hace referencia en el Anexo C.
- La instalación en el equipo servidor se ha llevado a cabo correctamente.
- Se ha actualizado el producto de software de la forma en que se ha detallado en el Anexo C.

En este punto es que se hace necesario realizar una configuración más en el OMS ya configurado, de forma de otorgar los permisos sobre el esquema *targetwh* al usuario que será el que ejecutará los trabajos de carga de datos. Esto se debe a que si no se hace, entonces no será capaz de actualizar el esquema destino de datos, con lo que los mappings no se ejecutarán correctamente. Para hacer esto es que se hace necesario iniciar el OMS y dirigirse a configurar *Preferred Credentials*. En este

punto se debe configurar lo referente a la base de datos, donde se debe ingresar al usuario *targetwh* y su contraseña. Esto se puede apreciar en la figura 4.12.

👯 Edit Administrator p	orefere	ences : MAU	RICIO					X
General Notific	ation	Schedule	A <u>c</u> cess	Preferred	Creder	ntials		
Target Name			Target Type		Crede	entials		
<default></default>			Database					
bdwh			Database			× .		
<default></default>			HTTP Server					
basilisco:80			HTTP Server					
<default></default>			Listener					
listener_basilisco			Listener			×		
<default></default>			Node					
basilisco			Node			×		
— Database bdwh Cr	edenti	als: ———						
Username:	TAR	∋ETWH						
Password:	*****	***						
Confirm Password:	*****	***						
Roļe:	NOR	MAL					-	
¢					ОK	Cancel	Help	

Figura 4.12: Configuración de las credenciales.

Los datos de configuración del *Listener* y el Nodo ya han sido ingresados si se ha seguido en detalle la configuración entregada en el Anexo B, pero para resumir, la credencial que hace referencia al *listener* debe ir configurada con la contraseña del usuario *"mauricio"* y el nodo debe ir configurado con los datos del usuario perteneciente al sistema operativo del servidor y su respectiva contraseña.

Es en este punto que ya se tienen las herramientas de diseño completamente instaladas, configuradas y listas para ser usadas en la creación del sistema de BI.

## 4.2.9 IMPLEMENTACIÓN DE PROCESOS ETL

Ahora es el momento de crear los procesos ETL que serán los encargados de poblar el DM de compras. Para realizar esto se debe iniciar el programa OWB *Client*, que se encuentra en el computador Cliente (descrito en el Anexo A). El OWB *Client* es el componente A de la figura C.1.

Cuando se inicia el programa aparece la ventana de conexión, en la cual se solicita un nombre de usuario, contraseña y la información de conexión. El usuario que debe usarse para realizar la conexión es el dueño del *Design Repository*, es decir, adminwh. También es necesario dar algo de información sobre la conexión, que consiste en el nombre del servidor donde se encuentra la base de datos, el puerto de comunicación por defecto y el nombre del servicio que se le asignó a la base de datos. Una vez que ya se han ingresado todos estos datos, se puede ingresar sin problemas al cliente para comenzar a crear los procesos ETL. Una vez ya dentro del programa se puede ver el árbol de exploración que usa este programa para trabajar y el cual tiene opciones como: bases de datos, archivos, aplicaciones y conexiones de Runtime. Sin embargo, antes de poder comenzar a utilizar sin problemas este programa se debe realizar una configuración más al OWB, que es cambiar el idioma por defecto a inglés. Esto es debido a un BUG del software que impide la ejecución de los mappings creados cuando el idioma utilizado por el cliente es diferente del inglés y que a la hora de realizar la creación de los procesos ETL estaba sin solución. Esta modificación se debe hacer en el menú de Proyecto, bajo la opción Preferencias. En la ficha General que aparece se debe especificar que la configuración local y el MLS (Multi Language Support) sea el inglés. Después de realizada esta modificación

se debe ingresar nuevamente al programa para que la configuración surta efecto. Es en este momento que se puede comenzar a utilizar la herramienta para crear y ejecutar los procesos ETL.

Las etapas en las que se puede dividir el proceso de creación de un DM usando el OWB son las siguientes:

- Crear un nuevo proyecto. Es necesario crear una nueva etapa de desarrollo para crear todos los procesos ETL que se necesiten. Un proyecto almacena y organiza todas las definiciones de metadato para el sistema destino. Estas definiciones incluyen orígenes de información, mappings, operaciones de transformación y parámetros de configuración.
- Definir módulos de origen y destino. Los módulos son objetos de almacenamiento que ayudan a organizar los objetos de origen y destino de información. Los módulos de origen contienen el metadato sobre los sistemas de origen existentes desde los cuales se extraerán datos. Los módulos destino contienen el metadato sobre lo que se está diseñando. El orden de creación de los módulos no es importante.
- Definir el movimiento y transformación de datos. Esto es lo que corresponde a la definición de los procesos ETL. La mayor parte del trabajo realizado en OWB corresponde a la definición de estos procesos.
- Validación y Generación. Esta es la etapa en la cual se validan los procesos
   ETL que han sido creados y en caso de estar bien se generan los *scripts* para su ejecución. En el proceso de validación se asegura que no existen

definiciones incompletas o inválidas. En la generación se crean los *scripts* para su posterior ejecución.

• Despliegue y ejecución. En este paso se registran todos los objetos creados y se pueden ejecutar los procesos correctamente definidos y validados.

#### **4.2.9.1 CREAR UN NUEVO PROYECTO**

Cuando se ingresa a OWB *Client* por primera vez, se crea por defecto un proyecto, con el nombre de *My Project*. Si se desea se puede utilizar el proyecto creado por defecto o escoger la opción *Create Project*, ubicada en el menú *Project*. Si se escoge crear un nuevo proyecto, lo único que ocurrirá es que en el árbol de navegación existirán dos proyectos: MY\_PROJECT y el nuevo, ambos con las mismas opciones. Es por esto que se usará el que aparece creado por defecto para definir los procesos ETL que se utilizarán para poblar el DM de Compras, ya que es posible cambiar el nombre del proyecto creado por defecto.

## 4.2.9.2 DEFINIR MÓDULOS ORIGEN Y DESTINO

Ya se ha mencionado que el orden de creación de los módulos de origen o de destino no es importante. Para este caso particular se creará y configurará primero el módulo destino. Para crear el módulo destino se deben seguir los siguientes pasos:

**Paso 1**: Expandir el árbol de navegación en Databases. En este punto se pueden crear módulos para bases de datos Oracle o para otras bases de datos.

**Paso 2**: En la ventana de identificación del módulo se debe ingresar un nombre. También se debe seleccionar el estado del módulo (desarrollo, aseguramiento de calidad o producción). Este estado del módulo no tiene ningún efecto sobre el sistema final y es solamente para tener una referencia. Finalmente se debe definir el tipo de módulo que se está creando, vale decir, corresponde a un origen o un destino de datos. En este caso se seleccionará la opción *Warehouse Target*.

**Paso 3**: A continuación aparece la ventana de información de conexión. La información ingresada será usada para que el usuario *adminwh* se pueda conectar al esquema de base de datos que será el destino, es decir, el esquema *targetwh*. Lo primero que hay que hacer es crear un DB Link al esquema destino de los datos y para esto se debe hacer click en el botón *New DB Link*, donde se deberá ingresar la información de la cadena de conexión a la base de datos, el usuario y la contraseña. Luego de crear el *DB Link* la ventana de información de conexión queda como se ve en la figura 4.13.

New Module Wizard: Co	onnection Infor	mation				X
	Select or create physically exist: metadata retrie Source for Meta © Oracle Dat © Oracle Des Please provide Database link:	a database link t s in the Warehous val. data Import: a Dictionary igner <u>R</u> epository the connection in DEST.US.ORAC Owner:	o import metadata into the e Builder repository sche formation by selecting a d CLE.COM	e module. 1 ma and is d atabase lin	The database link only used for k and a schema. New DB Link	
Cancel He	Schema:	User Name: Connect String: TARGETWH	TARGETWH basilisco:1521:bdwh	< Next	Change Schema	]

Figura 4.13: Información de conexión del módulo Destino.

**Paso 4**: En la siguiente ventana del asistente se crea una localización que será relacionada al módulo que se está creando. Los datos de la ubicación son simplemente un nombre, una descripción, y la base de datos en la que se encuentra el esquema destino de los datos.

**Paso 5**: Finalmente, aparecerá la última ventana del asistente, en la que se encuentra un resumen de todos los parámetros que se han configurado para crear el módulo destino.

El siguiente paso es el de crear y configurar el módulo de origen de los datos. Para realizar esto se deben seguir los siguientes pasos:

**Paso 1**: Expandir el árbol de navegación en Databases. En este punto se pueden crear módulos para bases de datos Oracle o para otras bases de datos.

**Paso 2**: Los datos que aparecen son los mismos que para el paso 2 explicado en la creación del módulo destino, salvo que el tipo del módulo que se debe seleccionar es *Data Source* (origen de datos).

**Paso 3**: Seleccionar información del origen de los datos. Esta información consiste en indicar de qué base de datos se extraerá la información.

**Paso 4**: En la siguiente ventana se debe configurar un DB Link para conectarse a la base de datos de origen y poder extraer toda la información que se necesite. En este lugar se tendría que ingresar la base de datos Oracle Database 7.3, pero esto no será así. El motivo de esta elección es que se traerán los datos en bruto de la base de datos 7.3.4.5.0 a la base de datos 9i que se ha instalado en un *tablespace* temporal y luego se usarán estos datos y se trabajará con ellos para crear los procesos ETL y ejecutarlos. Luego de su ejecución se eliminarán los datos en bruto de la base de

datos. Esto fue hecho de esta forma debido a que por problemas con los productos de Oracle, el OWB 9.2 no funciona bien con la base de datos Oracle 7.3.4.5.0.

Si bien es cierto que se conocía este problema antes de comenzar y se podía haber utilizado una versión menor, tras conversar con el personal de informática y ante la pronta migración de la base de datos 7.3.4.5.0 a una versión 9i, se decidió trabajar con las últimas versiones de los productos, de la misma forma que el resto del personal ya había comenzado a hacerlo para el desarrollo de aplicaciones. De esta manera, el hecho de que se carguen los datos de esta forma, tampoco entorpece el buen desempeño de la base de datos de producción, ya que se extraen los datos durante la noche cuando la carga de trabajo es mínima. Por esto, se escogerá como versión de base de datos origen *Oracle Database 8/9i*.

Para la creación del DB Link, se hará de la misma forma que para el módulo destino. Este *DB Link* debe contener la información del usuario dueño del esquema temporal.

**Paso 5**: En la siguiente ventana se debe introducir información sobre la localización del módulo de origen. Eso se hará de la misma forma que para el módulo de destino, con la única diferencia que cambia solamente el nombre de la localización.

**Paso 6**: Finalmente, se llega a la ventana final del asistente, en la que se encuentra un resumen con todos los parámetros configurados para la creación del módulo de origen.

Luego de definir los módulos solamente falta crear conectores entre ellos, de forma que puedan comunicarse y compartir información sin problemas usando el cliente de OWB. Para definir los conectores se deben seguir los siguientes pasos: **Paso 1**: Expandir el árbol de navegación desplegando todos los módulos Oracle y sus localizaciones.

**Paso 2**: Hacer click sobre la localización destino (de nombre UBIDEST) y escoger la opción *Create Connector*.

**Paso 3**: En la siguiente ventana se debe ingresar un nombre para el conector y una breve descripción de él.

**Paso 4**: A continuación se debe indicar la localización a la que hace referencia el conector. Para este caso se debe seleccionar la opción Database e indicar el nombre de la ubicación origen (para este caso es UBIORI). Para finalizar se presiona simplemente el botón *Finish*.

**Paso 5**: Hacer click sobre la localización de origen (de nombre UBIORI) y escoger la opción *Create Connector*.

**Paso 6**: En la siguiente ventana se debe ingresar un nombre para el conector y una breve descripción de él.

**Paso 7**: A continuación se debe indicar la localización a la que hace referencia el conector. Para este caso se debe seleccionar la opción Database e indicar el nombre de la ubicación origen (para este caso es UBIDEST). Para finalizar se presiona simplemente el botón *Finish*.

El último paso antes de comenzar la creación de los procesos ETL es configurar el módulo destino para que utilice los tablespaces creados para los datos y los índices. Para hacer esto se debe hacer click con el botón derecho del mouse sobre el nombre del módulo destino y escoger la opción *Configure*. La figura 4.14 muestra

cómo debería quedar configurado el módulo destino de acuerdo a los parámetros configurados para el servidor de base de datos y los tablespaces que se crearon.

🗱 Configuration Properties DESTINO "R 🔀				
۰.				
🛛 🖯 Tablespace Defa				
Default Index Tab	TB_INDEX			
Default Object Ta	TB_DATOS			
⊕ Generation Prefe				
⊕ Run Time Directo				
⊕ Generation Targe				
Main Application Sh	ora			
Application Short N	wb			
Schema Owner	owb			
Connect String				
Remote Host Name	basilisco			
Port	1521			
Service Name	bdwh			
Top Directory	\\codegen\			
Deployable	true			

Figura 4.14: Configuración módulo destino.

En este momento ya se tienen completamente configurados los módulos a utilizar para la creación de los procesos ETL con el software cliente de OWB.

# 4.2.9.3 DEFINIR EL MOVIMIENTO Y TRANSFORMACIÓN DE DATOS

Antes de crear los procesos ETL se explicará el proceso que será usado para poder realizar la carga de datos, ya que como se mencionó en el punto anterior existe un problema de compatibilidad entre versiones que impide extraer directamente los datos desde la base de datos 7.3.4.5.0. Para extraer los datos en bruto se usará un *job* que creará tablas en el esquema temporal de datos. Luego, se usarán los procesos ETL para cargar los datos del módulo origen al destino ya definidos en OWB. La ejecución de los procesos se realizará mediante otro *job*. Tras la finalización de la carga de datos se usará un *job* que eliminará todas las tablas creadas en el esquema temporal. Todos estos *jobs* se ejecutarán con el OMS que ya se encuentra configurado según lo explicado con detalle en el anexo B y lo especificado en el punto 4.2.8 del presente documento.

El *job* que cargará los datos al esquema de datos temporal consiste simplemente de instrucciones SQL estándar que crearán tablas como las originales de la siguiente forma:

## create table nombre\_tabla as (select \* from tabla\_original);

El *job* que eliminará las tablas del esquema temporal contiene instrucciones del siguiente tipo:

## drop table nombre\_tabla;

Con la forma de trabajar ya clara se procederá a la creación de los procesos ETL para realizar la carga de datos del DM de compras.

La forma de crear los procesos ETL es similar para todos, por lo que no se hace necesario explicar con detalle cada uno de ellos.

Cada proceso ETL que se cree se usará para cargar los datos de las tablas de origen a tablas destino en el esquema del usuario *targetwh*. Es por eso que se necesita saber qué datos serán los que se guardarán en las tablas, ya que éstas deben ser creadas antes de ejecutar los procesos ETL. Como se mencionó anteriormente, en la etapa del diseño del sistema, se identificaron todos los orígenes de información claramente y con el conocimiento de estos datos se sabe con certeza cuáles serán los datos y de qué tipo es cada uno de ellos, por lo que es posible crear de antemano las tablas que serán el destino de toda la información y con la cual se generarán los informes que han sido requeridos por los usuarios.

Las tablas destino son creadas usando un asistente del OWB. A modo de ejemplo se mostrarán los pasos a seguir para crear la tabla que guardará la información de los proveedores, ya que el resto de las tablas se crearán de la misma forma. Más adelante se mostrará cómo se creo el proceso ETL que realiza la carga de esta tabla.

Los pasos a seguir para crear la tabla de proveedores son los siguientes:

**Paso 1**: Expandir el árbol de navegación en el módulo Destino y hacer click con el botón derecho sobre la opción *Tables*. En este lugar se debe seleccionar la opción *Create Table*.

**Paso 2**: En la ventana inicial, se debe especificar un nombre para la tabla que se creará. También se puede ingresar una descripción sobre la tabla, la cual es opcional. El nombre que se ha escogido para la tabla de proveedores es PO\_VENDORS.

**Paso 3**: En la siguiente ventana se deben agregar las columnas que tendrá la tabla que se está creando. También se define el tipo de dato que se almacenará en cada columna y su largo.

**Paso 4**: Se pasa a la ventana final del asistente, dónde aparecerá un resumen de lo que se está creando. En este punto y si todo se encuentra bien definido, se debe finalizar el asistente para que la tabla sea creada.

Este procedimiento se debe seguir para la creación de las demás tablas que se necesitarán a la hora de crear los procesos ETL. Luego de creadas las tablas se tienen todas las tablas que se ven en la figura 4.15.



Figura 4.15: Tablas del módulo destino.

Ahora que ya se tienen creadas las tablas que contendrán todos los datos necesarios para generar los informes de gestión, se deben crear los procesos para cargar todos los datos. Como ya se mencionó anteriormente, se explicará la forma de creación de los procesos ETL solamente para el caso de los proveedores, ya que para los demás se realiza de la misma forma.

Para crear el proceso ETL para la carga de la tabla de proveedores se deben seguir los siguientes pasos:

**Paso 1**: Expandir el módulo destino y hacer click con el botón derecho del mouse sobre la opción Mappings. Seleccionar *Create Mapping*.

**Paso 2**: En la ventana que aparece se debe ingresar un nombre para el *mapping*. También es posible agregar una descripción, la cual es opcional. El nombre ingresado para el *mapping* fue MAP\_PO\_VENDORS. Luego de ingresados los datos solicitados se abre el editor de mappings.

En el editor de mappings deben cargarse todos los objetos que se usarán para cargar los datos y/o efectuar transformaciones sobre ellos. Para el caso de los proveedores se cargaran tres tablas, dos que serán origen de información y una que será el destino de los datos. Además se utilizará una herramienta para relacionar los datos de las tablas de origen formando así una sola estructura que será el resultado final para el caso de los proveedores.

La herramienta para unir las tablas tiene 3 partes. La primera es para indicar qué campos de la tabla de origen número uno serán extraídos para realizar la operación de unión. La segunda parte es para indicar qué campos de la segunda tabla de origen número dos serán usados para la operación de unión. La tercera parte indica los datos que resultan de la unión de las dos tablas de acuerdo a una condición que se especificará. Desde la tercera parte es que se deben sacar los datos para que sean insertados en la tabla destino. En la figura 4.16 pueden verse claramente las tablas de origen que utilizando la herramienta *join* une datos de las dos tablas para cargarlas en la tabla destino PO\_VENDORS.



Figura 4.16: Mapping para carga de proveedores.

**Paso 3**: Luego de definir el *mapping* es necesario validarlo, de forma de evitar que existan problemas de incongruencia de tipos de datos entre los orígenes y el destino de la información. De esta forma se asegura que el *mapping* se ejecutará más tarde sin problemas. Para realizar todo esto, se debe ir al menú del editor de *mappings* y escoger primero la opción *Validate*. Si todo resulta bien y no hay incoherencias se debe ir nuevamente al menú del editor de mappings y escoger la opción *Generate\Mapping* para ver si se puede generar el código que define el *mapping* recién creado. El código que resulta son instrucciones SQL que podrían servir para generar el *mapping* de forma manual. Si todo resulta sin errores ni advertencias se puede cerrar el editor de *mappings*.

De la forma descrita se deben crear todos los procesos ETL necesarios para que se carguen los datos en las tablas del módulo Destino.

Luego de que ya se tienen definidas todas las transformaciones necesarias para cargar el módulo destino, se tienen que validar y desplegar los mappings, de forma que puedan ser ejecutados más tarde las veces que se requiera. Para hacer esto se debe utilizar la herramienta *Deployment Manager*. Para ingresar al *Deployment Manager*, se debe crear antes que nada una conexión al repositorio de *Runtime*, en el cual se deben ingresar los datos del servidor de base de datos, el usuario dueño del repositorio de *Runtime* y el usuario de conexión al repositorio. Para esto se debe hacer click con el botón derecho del mouse sobre la opción *Runtime Repository Connections* y escoger la opción de crear una nueva conexión. Lo primero a ingresar es un nombre para la conexión que se está creando, para luego ingresar los datos del servidor y los usuarios de base de datos. Los datos ingresados pueden verse más claramente en la figura 4.17.

New Runtime Repository	Connection Wizard: Details	;	X
	Type in the connection details	8: 	
	Host Name:	basilisco	
21-71	Service Name:	bdwh	
	– Connect As <u>U</u> ser:	runtimeac	
1	Runtime Repository Owner:	runtimewh	
Cancel <u>H</u> elp		🛛 🔇 🗄 🖉 🖉 🖉 🖉 🖉 🖉 🖉 🖉	h ]

Figura 4.17: Detalles de conexión al repositorio de Runtime.

Luego de ingresar los datos que aparecen en la figura 4.17 se finaliza el asistente para crear la conexión al repositorio de *Runtime*. Ahora, para que finalmente se pueda ejecutar el *Deployment Manager*, se debe iniciar el OWB OC4J (OWB *Oracle Containers for Java*) y luego se debe hacer click sobre la conexión al repositorio de Runtime recién creada y escoger la opción Editor. En este momento aparece un cuadro de dialogo en el cual se debe ingresar la contraseña del usuario con el que se fijó el acceso al repositorio de Runtime, es decir, *runtimeac*. Después de ingresada la contraseña correcta aparece el OWB *Deployment Manager*.

Lo que se hace con el *Deployment Manager* es desplegar todos los objetos que se han creado dentro del OWB. El despliegue de los objetos consiste de generar los *scripts* SQL que servirán para cargar los datos desde los orígenes al módulo destino. Con estos *scripts* se podrá más tarde programar *jobs* que ejecuten los mappings creados cuando se desee.

Antes que los objetos puedan ser desplegados, se deben registrar las ubicaciones que se han definido. Para este caso particular son dos: UBIDEST y UBIORI. Para registrarlas se debe hacer simplemente un click con el botón derecho del mouse sobre el nombre de la ubicación y se escoge la opción *Register*. Al hacer esto, aparece una ventana en la cual se pide el ingreso de los datos sobre el esquema en el que se encuentran los datos del módulo que se está registrando. Los datos solicitados son los siguientes: nombre de esquema, contraseña, nombre del servicio, nombre del servidor de base de datos y el puerto de comunicación con la base de datos. Luego de ingresados todos estos datos para las dos ubicaciones se puede comenzar a desplegar los objetos de los dos módulos.

Primero se desplegarán los objetos del módulo de origen. Para esto se debe seleccionar la ubicación del módulo de origen (UBIORI) y automáticamente aparecerán los objetos que se encuentran en este módulo al lado derecho. La acción por defecto viene ya definida, que es la de la cual es la de crear los objetos. Como esta es la opción que sirve se debe comenzar a desplegar el módulo. Para hacerlo se debe ir al menú de archivo del OWB *Deployment Manager* y escoger la opción *Generate/Deploy*. Al realizar todo lo anterior, lo primero que hace el programa es generar el objeto y si todo sale bien pide que se despliegue.

Después de que se han generado y desplegado todos los objetos del módulo de origen sin problemas se procede a trabajar con los objetos del módulo destino. El proceso es el mismo, pero se recomienda generar y desplegar todos los objetos que no son *mappings* antes que estos. Luego de que ya todo ha resultado bien, se pueden generar y desplegar los *mappings*.

Cuando ya todos los objetos han sido creados y desplegados, se pueden ejecutar los *mappings* para cargar los datos. La ejecución puede hacerse desde el mismo OWB *Deployment Manager*. Para realizar esto se debe seleccionar uno a uno cada *mapping* y dirigirse al menú de archivo, donde se debe seleccionar la opción *Execute*. Luego de ejecutar los mappings se debe obtener un resultado como el que se muestra en la figura 4.18.

Execution Results for MAP_PO_V	ENDORS			
Summan				
ounnary				
Run Name MAP_PO_VENDC	RS			
Completion Status Completed succe	essfully.			
Dou totinit				
Row Activity				
MAR RO VENDORORO VENDOR	Inserted	Updated	Deleted	Merged
Output Parameters				
Output Parameters Message Log				
Output Parameters Message Log Starting Execution MAP_PO_VENDOF				
Output Parameters Message Log Starting Execution MAP_PO_VENDOR Starting Task MAP_PO_VENDORS				
Output Parameters Message Log Starting Execution MAP_PO_VENDOR Starting Task MAP_PO_VENDORS Completing Task MAP_PO_VENDORS				
Output Parameters Message Log Starting Execution MAP_PO_VENDOF Starting Task MAP_PO_VENDORS Completing Task MAP_PO_VENO Completing Execution MAP_PO_VEN	RS S DORS			
Output Parameters Message Log Starting Execution MAP_PO_VENDOF Starting Task MAP_PO_VENDORS Completing Task MAP_PO_VENDOR Completing Execution MAP_PO_VEN	RS S DORS			
Output Parameters Message Log Starting Execution MAP_P0_VENDOR Starting Task MAP_P0_VENDORS Completing Task MAP_P0_VENDOR Completing Execution MAP_P0_VENI	RS S DORS			
Output Parameters Message Log Starting Execution MAP_PO_VENDOR Starting Task MAP_PO_VENDORS Completing Task MAP_PO_VENDOR Completing Execution MAP_PO_VEN	RS S DORS			
Output Parameters Message Log Starting Execution MAP_PO_VENDOF Starting Task MAP_PO_VENDORS Completing Task MAP_PO_VENDOR Completing Execution MAP_PO_VENI	RS S DORS			
Output Parameters Message Log Starting Execution MAP_PO_VENDOR Starting Task MAP_PO_VENDORS Completing Task MAP_PO_VENDOR Completing Execution MAP_PO_VENI (4	RS S DORS			
Output Parameters Message Log Starting Execution MAP_PO_VENDOR Starting Task MAP_PO_VENDORS Completing Task MAP_PO_VENDOR Completing Execution MAP_PO_VENI [4]	RS S DORS			
Output Parameters  Message Log  Starting Execution MAP_PO_VENDOR Starting Task MAP_PO_VENDORS Completing Task MAP_PO_VENDOR Completing Execution MAP_PO_VENI	RS S DORS			

Figura 4.18: Resultados de la ejecución del mapping MAP\_PO\_VENDORS.

Resultados como el que se ve en la figura 4.18 se deben obtener al ejecutar todos los *mappings* para corroborar su correcta ejecución y no se tengan problemas cuando se ejecuten por medio de los *jobs* y el OMS.

Existe una utilidad que tiene el OWB para auditar el funcionamiento y el estado de los mappings y los objetos de los módulos, la cual se llama OWB *Runtime Audit Browser*. Para poder ejecutar esta utilidad se tiene que cargar en memoria el OC4J.

Al iniciar el OWB *Runtime Audit Browser* se abre una ventana del browser de Internet por defecto del equipo en la cual se debe ingresar la información de conexión al repositorio de Runtime. El usuario que debe utilizarse debe ser el dueño del repositorio de *Runtime*, es decir, *runtimewh*. Esto se puede ver en la figura 4.19.

## ORACLE Warehouse Builder Browser

## **Runtime Repository Report**

(i) Information	
Please supply connection details	
Select Role	Warehouse Administrator 💌
Runtime Repository	runtimewh
Password	•••••
Specify the host address, port number, and service name Host Address	hasilissa
Host Port Number	1521
Host Service Name	bdwh
◯ Specify the net service name	
Net Service Name	
View Deployment Report View Executio	n Report) View Management Report)

Copyright @ 1996, 2002, Oracle Corporation. All rights reserved.

Runtime Repository | Help

#### Figura 4.19: Conexión al Runtime Repository.

Como se aprecia claramente en la figura 4.19, existen tres informes para visualizar: informe de despliegue, de ejecución y el de administración. En el informe de despliegue se puede ver el estado de despliegue separado por localizaciones, objetos y por fecha de despliegue. En el informe de ejecución se puede ver el estado de la ejecución de todos los *mappings*. En el informe de administración se pueden ver datos sobre el repositorio, como son: versión, servidor de base de datos, puerto de comunicación y el nombre de servicio. A modo de ejemplo se mostrará en la figura 4.20 el resumen de ejecución de los *mappings*.

Repository <b>runtimewh</b> Description		Type OWB RUNTIME	
Filter Options			
Filter on Type	Filter on Ex	ecution Status 🔽 🔽	
Execution Runs			
Name	Туре 🛆	Latest Execution	Execution Status
MAP_FND_FLEX_VALUES	PL/SQL Map	23 Mar 2005 11:26:36	Complete
MAP_GL_CODE_COMBINATIONS	PL/SQL Map	23 Mar 2005 11:26:48	Complete
MAP_GL_PERIODS	PL/SQL Map	23 Mar 2005 11:27:13	Complete
MAP_HR_LOCATIONS	PL/SQL Map	23 Mar 2005 11:27:30	Complete
MAP_MTL_CATEGORIES	PL/SQL Map	23 Mar 2005 11:27:45	Complete
MAP_MTL_PARAMETERS_ALL_V	PL/SQL Map	23 Mar 2005 11:28:01	Complete
MAP_MTL_SYSTEM_ITEMS	PL/SQL Map	23 Mar 2005 11:28:20	Complete
MAP_PER_PEOPLE_F	PL/SQL Map	23 Mar 2005 11:28:38	Complete
MAP_PO_ACTION_HISTORY	PL/SQL Map	23 Mar 2005 11:28:56	Complete
MAP_PO_DISTRIBUTIONS_ALL	PL/SQL Map	23 Mar 2005 11:29:12	Complete

## runtimewh - Execution Summary Report

Figura 4.20: Parte del informe de ejecución.

## 4.2.9.4 CREACIÓN Y PROGRAMACIÓN DE LOS JOBS

Con la ayuda de los *jobs* es que se ejecutarán los mappings de forma automática para realizar la carga de los datos durante las horas de menor carga de los sistemas de información de la empresa. Ya se ha mencionado que esto se hará utilizando el OMS, el cual a esta altura ya se encuentra configurado para ejecutar los *jobs*. La creación de los *jobs* debe realizarse directamente en el servidor de base de datos donde se encuentra instalado el OMS.

Ya se mencionó que se iba a crear un *job* para cargar los datos en bruto a un esquema temporal y otro *job* que los eliminará después de que se ejecuten correctamente los procesos ETL. La creación de estos *jobs* ya fueron explicados en

los primeros párrafos del punto 4.2.9.3. Se explicará con detalle cómo se crea el *job* que ejecutará los *mappings* para cargar los datos al módulo destino.

Al ingresar al OMS aparece un árbol de exploración al lado izquierdo de la pantalla, en la cual existe la opción de *Jobs*. Para crear un nuevo *job* se debe hacer click con el botón derecho del mouse y escoger la opción de *Create job*.

En la ficha General se debe ingresar un nombre para el job que se está creando, además de definir sobre qué base de datos se creará el job. También se puede agregar una descripción del job de forma opcional.

En la ficha *Tasks* se ingresa qué tipo de trabajo se realizará. Para que se carguen los datos al esquema temporal y para eliminarlos, el tipo de tarea que se ejecutará será *Run DBA Script*. En tanto que para ejecutar los procesos ETL se tiene que escoger la opción *Run SQL\*Plus Script*.

Para ejecutar los procesos ETL se debe usar un *script* especial que viene incluido con el OWB. Este *script* se llama *oem\_exec\_template*. En la figura 4.21 se pueden ver los parámetros con los que se ejecutó el *mapping* para cargar los datos de los proveedores.

🍀 Create Job
General Tasks Parameters Schedule Access
Selected Tasks:       Run SQL*Plus Script (8)         Run SQL*Plus Script (9)       Parameters:         Run SQL*Plus Script (10)       Parameters:         Run SQL*Plus Script (10)       Parameters:         Run SQL*Plus Script (10)       Parameters:         Run SQL*Plus Script (11)       Parameters:         Run SQL*Plus Script (12)       Parameters:         Run SQL*Plus Script (13)       Parameters:         Run SQL*Plus Script (14)       Parameters:         Run SQL*Plus Script (15)       Parameters:         Run SQL*Plus Script (16)       Parameters:         Run SQL*Plus Script (17)       Parameters:         Run SQL*Plus Script (18)       Parameters:         Run SQL*Plus Script (19)       Parameters:         Run SQL*Plus Script (20)       Parameters:         Run SQL*Plus Script (21)       Parameters:         Run SQL*Plus Script (21)
© Sybmit © Add to Library © Submit and Add to Library Submit Cancel Help

## Figura 4.21: Parámetros para ejecución del mappings MAP PO VENDORS.

Se ve en la figura 4.21 los parámetros usados para ejecutar el *mapping* que cargará los datos de los proveedores. El primer parámetro corresponde al nombre del dueño del repositorio de *Runtime*. El segundo parámetro corresponde al nombre de la ubicación del módulo destino. El tercer parámetro indica si el tipo de trabajo que se realizará. Las opciones que se manejan son de 3 tipos: PLSQL, SQL\*Loader o un proceso. El cuarto parámetro que se ingresa corresponde al nombre de la tarea a ejecutar. En este caso, el cuarto parámetro corresponde al nombre de la mapping. Los últimos dos parámetros son utilizados para ingresar parámetros de sistema y parámetros personalizados, en caso de que el *script* haya sido modificado para realizar alguna acción especial. En este caso particular no se utilizan los últimos dos parámetros, ya que se ejecutará el *script* solamente con los datos necesarios para ejecutar los mappings.

De la misma forma en que se hizo para la carga de los datos de los proveedores (figura 4.21) se realiza la ejecución del resto de los mappings.

En la figura 4.21 se puede ver la ficha *Schedule*, desde la cual se programará la ejecución de los *jobs*. La programación del *job* será para que se ejecute todos los días a las 12:00 AM. Luego de que ya se ha definido el *job*, se presiona el botón *Submit* para que se ejecute de acuerdo a la programación especificada.

## 4.2.10 CREACIÓN DE INFORMES DE GESTIÓN

Luego de tener el trabajo que se encargará de la ejecución de los procesos ETL y que mantendrá actualizadas las tablas del módulo destino, se tienen que crear los informes de gestión que han solicitado los usuarios. Los informes se crearán usando el software Oracle Discoverer, el cual ya se ha instalado siguiendo los pasos explicados en el Anexo D.

El primer paso para crear los informes de gestión es crear un EUL para poder interactuar con la base de datos. Para crear un EUL se necesita tener un usuario de la base de datos con los siguientes privilegios: *create session, create table, create view, create sequence* y *create procedure*. También se recomienda tener un tablespace nuevo para generar el EUL con un mínimo de 3 MB. El usuario que se ha creado previamente se llama *admineul*.

Al iniciar el programa Oracle Discoverer Administrator se abre el EUL Manager, con el cual se debe crear un EUL para poder trabajar en la generación de los informes. Para crear el EUL se escoge la opción *Create an EUL* para pasar a la ventana en la que se especifican los datos del usuario dueño del EUL. También se debe escoger la opción de dar acceso al usuario PUBLIC. Antes de crear el área de negocios se debe dar acceso a las tablas y vistas del esquema destino de los datos para que puedan ser manipulados por el dueño del EUL. Se necesita solamente acceso para poder realizar consultas (instrucción *Select*).

Luego de que ya se dieron los accesos para que se puedan realizar consultas sobre las tablas del esquema destino, se debe ingresar nuevamente al programa de administración de Discoverer. Cuando se ingresa nuevamente aparece un asistente en el que se puede crear la nueva área de negocios, escogiendo la opción *Create a new Business area*. Los pasos para crear el área de negocios son los siguientes:

**Paso 1**: En la primera ventana se escoge el esquema de base de datos que contiene la información para generar los informes. En este caso corresponde a *targetwh*.

**Paso 2**: Seleccionar los objetos del esquema que se utilizarán para generar los informes. En este punto se cargarán las tablas que fueron creadas con el OWB y se poblaron con los procesos ETL.

**Paso 3**: Pasar a la última ventana del asistente, en la que se debe ingresar un nombre para el área de negocios y una descripción de la misma. El resto de los valores se deja por defecto.

**Paso 4**: Tras crear el área de negocios, se recomienda dejar todas las carpetas que se han agregado como ocultas al usuario. Para hacer esto se deben seleccionar todas las tablas y escoger la opción de propiedades (utilizando el botón derecho del mouse), donde se encuentra la opción *Visible to user* y escoger la opción *No*.

En este momento ya se tiene creada el área de negocios dentro de la cual se crearán los informes solicitados por los usuarios.

Luego de tener creada el área de negocios, se recomienda definir la jerarquía de fecha que usarán todos los campos de este tipo para la generación de los informes. Discoverer, genera una jerarquía de fechas por defecto cuando se crea el área de negocios, por lo que se puede editar esta jerarquía y modificarla para que quede como se desee. Para modificar la jerarquía se debe seleccionar la ficha *Hierarchies* y expandir el árbol de navegación *Date hierarchy templates*, lugar donde se puede editar la jerarquía de fechas por defecto. En este lugar se ha seleccionado que la jerarquía será del tipo *YYYY - Mon - DD* (año - mes - día) y será utilizada por todos los campos de tipo fecha que se usen.

Para ilustrar el uso del Discoverer en la generación de informes de gestión, se mostrará cómo se creó el informe de OC por centro de costo durante un periodo contable, el cual se encuentra entre los requerimientos de los usuarios.

Antes de poder generar este informe, es necesario tener un listado con los centros de costo que están cargados en el ERP de la compañía. Para hacer esto, se debe seleccionar la ficha *Data*, lugar donde se encuentran todas las carpetas que inicialmente se cargaron cuando se creó el área de negocios. Al hacer un click con el botón derecho del mouse aparece la opción de crear una carpeta personalizada (*New Custom Folder*). Al seleccionar esta opción aparece una ventana en la cual se debe introducir una instrucción SQL que servirá para crear la nueva carpeta con el resultado de la consulta introducida.

En la figura 4.22 se puede ver lo introducido en la ventana para la creación del listado de centros de costo. Inmediatamente después de que se crea, se debe dejar como oculta al usuario de la misma forma en que ocultaron las carpetas cargadas inicialmente.

📌 Custom Folder	
Enter the SQL to define this custom folder:	
select distinct flex_value ; (flex_value)    ' '    description Centro_Costo from targetwh.fnd_flex_values where flex_value in ('000','110','210','310','320','330','410','420','510','610','705 '765','805','810','815','820','825','830','835')	5','710','71
	>
Name: Centros de Costo	
Validate SQL OK Cancel	Help

Figura 4.22: Creación de carpeta de Centros de Costo.

El último paso antes de crear el informe es definir listas de valores que servirán de parámetros para de entrada cuando éstos se ejecuten. Para el caso del informe de OC por centro de costo al mes, se necesita seleccionar un mes, un año y un centro de costo, por lo que será necesario crear tres listas de valores. Más adelante, estas mismas listas de valores podrían servir para que otros informes las utilicen. Para crear la lista de valores de los centros de costo se deben seguir los siguientes pasos:

**Paso 1**: Seleccionar la ficha *Item classes*, para luego hacer click con el botón derecho del mouse sobre el nombre del área de negocios y escoger la opción *New item class*.

**Paso 2**: En la primera ventana que aparece marcar las opciones *List of values* y también *Drill to detail*. Luego continuar con el siguiente paso del asistente.

Paso 3: Seleccionar el campo que generará la lista de valores. En este caso se debe seleccionar el campo Centro Costo de la carpeta que se creó de forma personalizada.
Paso 4: En la siguiente ventana, se pide ingresar los campos que utilizarán esta lista de valores. En este caso se debe pasar a la ventana siguiente y presionar botón para

finalizar el asistente, ya que no se requieren más modificaciones.

De la misma forma en que se creó esta lista de valores, se deben crear las listas para seleccionar los años y los meses. En este momento se puede generar el informe. Para hacer esto se debe seleccionar la ficha *Data* y tras hacer click con el botón derecho del mouse sobre el nombre del área de negocios, escoger la opción *New folder*. Al aparecer la nueva carpeta, ingresarle un nombre, que en este caso fue "Informe OC x CC al mes".

La forma de confeccionar el informe es arrastrando con el mouse los campos que se necesiten desde las carpetas que se cargaron de forma inicial hacia la carpeta que contendrá el informe. Para hacer esto se debe tener claramente identificado de qué tablas se extraerán los datos. Para ir agregando campos de diferentes tablas se deben definir relaciones entre las distintas tablas desde las cuales se extraerán los datos. Estas relaciones son de la misma forma que se ejecutan usando SQL. Por ejemplo, si quiero extraer datos de las tablas po\_headers\_all y de po\_lines\_all, se relacionan usando el campo po\_header\_id. En SQL podría existir una consulta de la siguiente manera:

select ph.segment1, pl.line\_num, pl.item\_description

from po\_headers\_all ph, po\_lines\_all pl
where ph.po\_header\_id = pl.po\_header\_id;

Para hacer lo mismo que la consulta SQL usando Discoverer, se debe definir la relación entre las carpetas por el campo po\_header\_id. Estas relaciones en Discoverer se llaman *Join*. Para crear un *join*, se debe ir al menú de *Insert* y escoger la opción *Join*. A modo de ejemplo se creará un *join* entre las dos tablas que se han mencionado, ya que de la misma forma se definirán los otros *joins* para relacionar más tablas y así, poder utilizar columnas de varias tablas en la generación de un informe.

Al insertar un nuevo *join* aparece una ventana en la cual se deben especificar los campos que servirán para relacionar dos tablas. En la parte de la carpeta maestra se seleccionará el campo po\_header\_id de la tabla po\_headers\_all y en la carpeta detalle se seleccionará el mismo campo de la tabla po\_lines\_all. En la figura 4.23 se pueden ver las opciones ingresadas para la creación del *join* del que se ha hablado.

New Join	×
Mast Po Headers A	er Folder Operator Detail Folder
N <u>a</u> me:	Join Options 🛛 🔀
D <u>e</u> scription: <u>M</u> ulti-Item >>	Join Options Outer join on detail Outer join on master Detail foreign key can have null values. (summary tables will not be used with this join) One to one join relationship between master and detail
	OK Cancel Help

Figura 4.23: Creación de join.

En este caso particular no se necesita especificar ningún tipo de *outer join* ni en la tabla maestra ni tampoco en la de detalle, ya que siempre existe el campo po\_header\_id en las dos tablas, por lo que se han dejado las opciones sin marcar.

Para crear el resto de los *joins* hay que tener cuidado en que si existirá siempre o no algún campo entre las tablas que se relacionen.

Luego de que se creen todas las relaciones entre las tablas se puede comenzar a arrastrar los campos necesarios para que se genere el informe. Los campos también pueden ser renombrados para que los nombres sean claros para el usuario final de Discoverer.

Luego de que ya se han cargado todos los campos que necesitará el informe, se debe utilizar el software Oracle Discoverer Desktop, con el cual se configurará de la forma en que realmente lo verá el usuario final de Discoverer. Tras iniciar el programa aparece una ventana de conexión, donde se debe ingresar el nombre de usuario, contraseña y la base de datos con la cual se conectará el software. En esta ventana se deben ingresar los datos del usuario que se usó para crear el EUL anteriormente, es decir, *admineul*. Luego de ingresar estos datos se inicia automáticamente un asistente que permite escoger entre la creación de un nuevo informe o la posibilidad de abrir uno ya existente. En este caso se escogerá la opción de crear uno nuevo (*Create a new workbook*). Los pasos a seguir para crear el informe son los siguientes:

**Paso 1**: Al escoger la opción de crear un nuevo libro se puede escoger entre cuatro opciones:

- Estilo de tabla: Es una tabla simple con datos distribuidos en filas y columnas.
- Estilo tabla con detalles: Es una tabla pero con varias páginas de datos.
- Estilo de tabulación cruzada: Relaciona dos conjuntos de datos y los relaciona resumiendo en un tercer conjunto de datos.
- Estilo tabulación cruzada con detalles: A diferencia de la anterior contiene múltiples páginas de datos.

Para implementar el informe de OC por centro de costo al mes basta solamente con escoger el estilo tabla, ya que se obtendrán datos de un centro de costo por mes. Por ejemplo, si se hubiera querido realizar el informe para varios centros de costo, se podría haber escogido la opción de estilo tabla con detalles.

**Paso 2**: Seleccionar el área de negocios en la que se creará el informe y de dónde se extraerán los datos para él.

**Paso 3**: En la ventana siguiente se puede cambiar la forma en que aparecerán las columnas y filas del informe. Para este caso particular se dejarán las columnas tal como están, puesto que ya fueron agregadas en el orden correcto cuando fueron creadas con el Oracle Administrator.

**Paso 4**: Agregar las condiciones que existieran para que se ejecute el informe. En este caso particular no se tiene ninguna condición, por lo que se pasa a la siguiente ventana del asistente.

**Paso 5**: Escoger la(s) columna(s) que servirá para realizar un ordenamiento de los datos que contiene el informe al momento de ser mostrado al usuario. En este caso

particular se deben ordenar los datos por número de OC y luego por el número de la línea de OC.

**Paso 6**: Definir los cálculos que sean necesarios. Para este informe se necesita realizar el cálculo del total de cada línea de OC, teniendo la cantidad y el precio unitario.

Paso 7: Finalizar el asistente para generar el informe.

Tomará algunos segundos antes de que muestren los primeros resultados, pero luego de que aparezcan se debe terminar su confección. Lo que falta por especificar es la utilización de los parámetros para el informe, vale decir, que sea ejecutado para un centro de costo en particular y para un periodo contable determinado.

Para definir los parámetros que utilizará el informe se debe ir al menú de herramientas y seleccionar la opción *Parameters* y luego escoger la opción *New*. En la ventana que aparece, se debe seleccionar el ítem que se tomará como parámetro y también un mensaje para que el usuario seleccione un valor para dicho parámetro. También aparece la opción opción *Let user enter multiple values*, la cual, de estar marcada, permitirá seleccionar varios valores para un parámetro. Por ejemplo, ejecutar el informe para varios meses.

Tras definir los parámetros de año, mes y centro de costo, se vuelve a ejecutar el informe, seleccionando los parámetros que se deseen. Finalmente se muestran los resultados de acuerdo a todo lo configurado. Una vez que ya se muestran los datos, se debe grabar el informe en la base de datos para que más adelante pueda ser usado por más usuarios. Para hacer esto se debe seleccionar la opción *Save as* del menú de
archivo. Tras asignarle un nombre al informe y presionar el botón *Save* se grabará en el *tablespace* para datos del dueño del EUL.

Ahora que ya se tiene el informe bien configurado y grabado en la base de datos se tiene que dar permisos a los usuarios para que puedan tener acceso a él cuando lo requieran.

#### 4.2.11 CONFIGURANDO EL ACCESO A LOS INFORMES

Para darle permiso a un usuario en particular para que vea un informe que se encuentra dentro de un área de negocios en particular, antes tiene que existir el usuario en la base de datos. Para crear al usuario se utilizará el OEM de la misma forma en que se creó al usuario con privilegio de SYSDBA según lo descrito en el anexo B del presente documento. El usuario que se creará tendrá las siguientes características:

- Acceso protegido con contraseña.
- El único Rol que tiene es el de CONNECT.
- Los privilegios de sistema que poseerá son: *alter session, create cluster, create database link, create sequence, create session, create synonym* y *create view.*
- Los tablespaces por defecto que utilizará serán USERS y TEMP.

Con estas características, el usuario podrá conectarse utilizando el programa Oracle Discoverer Desktop.

Una vez que se tiene al usuario creado, se debe utilizar el programa administrador de Discoverer para permitirle el ingreso al área de negocios. Para hacer esto, se debe ingresar al Oracle Discoverer Administrator con la cuenta del usuario dueño del EUL y abrir el área de negocios a la que se quiere otorgar acceso al usuario recién creado. Al ingresar al programa se debe ir al menú de herramientas y seleccionar la opción *Security*. Luego se debe agregar al usuario para que tenga acceso a un área de negocios en especial. En la figura 4.24 se puede ver como queda la ventana de permisos cuando se le da acceso a otro usuario para ingresar al área de negocios.



Figura 4.24: Otorgando acceso a un área de negocios.

Como se ve en la figura 4.24, los usuarios que tendrán acceso al área de negocios de Operaciones son: ADMINEUL (dueño del EUL en donde se encuentra definida el área de negocios) y LANTIHUAL. Este último es un usuario que tendrá acceso al informe de Discoverer para poder trabajar con él.

Luego de que se ha permitido el ingreso del usuario al área de negocios de Operaciones, se debe especificar que privilegios tendrá cada usuario sobre el área. Para el caso del usuario LANTIHUAL, se puede apreciar en la figura 4.25 los privilegios que se le otorgaron. Se ve que no podrá grabar el informe a la base de datos para evitar que realice cambios qua afecten a los demás usuarios. Tampoco podrá otorgar acceso al informe a otros usuarios ni programar la ejecución automática del informe. En el resto de las fichas que aparecen en la figura 4.25 no es necesario realizar ningún tipo de cambios.

Privileges				X
Privileges	User/Role Que	ry Governor	Scheduled W	Vorkbooks
Show privileg	es for: 🔽 User 🔽 Role	LANTIHUA	L	•
Privilege		Description		
<ul> <li>✓ User Edition</li> <li>✓ Create</li> <li>✓ Collect</li> <li>✓ Item Di</li> <li>✓ Drill Out</li> <li>Grant N</li> <li>Schedu</li> <li>Save W</li> <li>✓ Change</li> </ul>	on /Edit Query Query Statistics rill t Workbook Jorkbooks /orkbooks to Database e Password			
Select a syste	em profile:			<b>_</b>
	ОК	Cancel	Apply	Help

Figura 4.25: Privilegios del usuario LANTIHUAL.

Ahora solamente falta dar acceso al informe que se ha estado configurando. Para hacer esto, se debe iniciar el programa Oracle Discoverer Desktop e ingresar con el usuario dueño del EUL (con el que se creó el informe). No es necesario que el informe se ejecute para que se pueda dar el acceso a un usuario particular, por lo que no se hace necesario el ingreso de los parámetros. Se debe ir al menú de archivo y escoger la opción *Manage workbooks*, dentro de la cual hay que seleccionar la opción *Sharing*. Luego de que se han seguido estos pasos aparece una ventana en la que se debe seleccionar el informe al cual se quiere dar acceso y también se deben escoger a los usuarios que podrán trabajar con este informe.

En este punto, el usuario LANTIHUAL ya tiene acceso al informe que se ha mencionado, por lo que podrá ingresar usando el programa Oracle Discoverer Desktop para poder trabajar sin problemas. Tras ingresar al programa utilizando al usuario LANTIHUAL y ejecutar el informe se extraen los resultados deseados. El usuario puede exportar sus datos a Excel o grabarlos con otro formato que le sea de su agrado y realizar las modificaciones que estime conveniente.

👂 Oracle	9i Discover	er Desk	top - [ADMIN	IEUL.AO-	NFORME OC	C X CC A	L MES]				
🔊 File Ec	dit View Sh	ieet For	mat Tools Gr	raph Wind	low Help						
🔇 🍃		<b>a</b>		Z 🗩	3	🔁 ź	↓ Ž↓ 🖷	X			
Tahoma	3		• 8	• B	t <u>U</u>			ŝ	<b>6</b> , (	00 00	F   M
								INEO			<b>DO COETO</b> .
								1110	RIME DE UL P	OK LENI	KU LUSTU /
								140	KIME DE UL P		KULUSIUI
	Creacion sol	NOC	Creacion OC	Pro	veedor	Empresa	Cuenta	CCosto	Producto	Item	Estado OC
1	Creacion sol	N OC 24825	Creacion OC 03-FEB-2005	Pro ENTEL CHI	veedor LE S.A.	Empresa 01	Cuenta 410201009	CCosto 720	Producto	Item 38033	Estado OC APPROVED
1 2	Creacion sol	N OC 24825 24874	Creacion OC 03-FEB-2005 07-FEB-2005	Pro ENTEL CHI ENTEL PCS	veedor LE S.A. 5 TELECOMUNI	Empresa 01 01	Cuenta 410201009 410201009	CCosto 720 720	Producto 000000000 000000000	Item 38033 38027	Estado OC APPROVED APPROVED
1 2 3	Creacion sol	N OC 24825 24874 25051	Creacion OC 03-FEB-2005 07-FEB-2005 17-FEB-2005	Pro ENTEL CHI ENTEL PCS ENTEL CHI	veedor LE S.A. 5 TELECOMUNI LE S.A.	Empresa 01 01 01	Cuenta 410201009 410201009 410201009	CCosto 720 720 720	Producto 000000000 000000000 000000000	Item 38033 38027 38033	Estado OC APPROVED APPROVED APPROVED
1 2 3 4	Creacion sol	N OC 24825 24874 25051 25051	Creacion OC 03-FEB-2005 07-FEB-2005 17-FEB-2005 17-FEB-2005	Pro ENTEL CHI ENTEL PCS ENTEL CHI ENTEL CHI	veedor LE S.A. 5 TELECOMUNI LE S.A. LE S.A.	Empresa 01 01 01 01 01	Cuenta 410201009 410201009 410201009 410201009	CCosto 720 720 720 720 720	Producto 000000000 000000000 000000000 00000000	Item 38033 38027 38033 38034	Estado OC APPROVED APPROVED APPROVED APPROVED
1 2 3 4 5	Creacion sol	N OC 24825 24874 25051 25051 25094	Creacion OC 03-FEB-2005 07-FEB-2005 17-FEB-2005 17-FEB-2005 17-FEB-2005	Pro ENTEL CHJ ENTEL PCS ENTEL CHJ ENTEL CHJ SOC.DEPA	veedor LE S.A. ; TELECOMUNI IE S.A. LE S.A. RTAMENTOS E	Empresa 01 01 01 01 01 01	Cuenta 410201009 410201009 410201009 410201009 410201009	CCosto 720 720 720 720 720 720	Producto 000000000 000000000 000000000 00000000	Item 38033 38027 38033 38034 38034	Estado OC APPROVED APPROVED APPROVED APPROVED APPROVED

Figura 4.26: Parte del informe OC X CC AL MES.

Tras ejecutar el informe se obtiene lo que en parte se muestra en la figura

4.26. En esta figura no se muestran todos los datos, ya que corresponde a información que es propiedad de la empresa y no se tiene la autorización para hacer de uso público datos que le pertenezcan.

De la misma forma en que se generó este informe se generará el resto, siguiendo la especificación de requisitos hecha anteriormente. Todos los informes se crearán dentro del área de negocios de Operaciones, por lo que si se desea que el usuario LANTIHUAL tenga acceso a otro informe, se deberá dar permiso al informe en especial uno a uno.

Para que más usuarios puedan acceder a los informes dentro del área de negocios se deben seguir los mismos pasos que los explicados para el caso de LANTIHUAL.

Cuando se genere un nuevo informe utilizando el Oracle Discoverer Administrator se debe tener claro de dónde extraer los datos para cada uno de los informes que se confeccione. Es una buena práctica realizar la consulta SQL y tenerla clara antes de confeccionar el informe a través de Discoverer. Hay que ver si los *joins* ya están definidos, puesto que como se trata de las mismas tablas para extraer distintos informes, puede ser que ya estén relacionadas.

## **CAPÍTULO 5: CONCLUSIONES**

En Chile, hoy existe gran cantidad de empresas que se encuentran implementando sistemas de BI, en diversas áreas. En este caso se ha llevado este tipo de tecnología a la industria de alimento para Salmones, la cual recién está comenzando la actualización de sus sistemas y ha notado la necesidad de utilizar de buena forma la información que ya posee, de forma que pueda tener más herramientas que la ayuden en la toma de decisiones.

Se ha hecho un estudio de las distintas tecnologías para desarrollar sistemas de BI, donde se ha visto que ha visto que parte importante en la implementación de estos sistemas es el proceso de extracción y manipulación de los datos. Los procesos ETL son ampliamente usados para extraer la información de las distintas fuentes de datos. Se ha confeccionado una lista con todo lo que es necesario evaluar a la hora de escoger una herramienta adecuada para el desarrollo de procesos ETL. También han sido revisadas las características principales de distintos productos existentes en la actualidad, con los que se pueden implementar este tipo de procesos, mencionando ventajas y desventajas de cada uno de ellos.

Luego de implementar el sistema de BI para el área de compras de la empresa se ha conseguido que los usuarios utilicen lo que se les ha desarrollado, lo cual indica un alto grado de conformidad respecto a lo que han solicitado. Se ha logrado que el tiempo que antes ocupaban para poder desarrollar algunas de sus tareas sea muchísimo menor, con lo que pueden ocupar el tiempo que les queda libre en desarrollar nuevos requerimientos para su área y luego poder implementarla, de forma que la gestión de la empresa iría mejorando cada vez. La importancia de que este grupo de usuarios utilice el sistema en el desarrollo de sus tareas diarias es que ayudará a que las demás áreas se interesen en ayudar al desarrollo de un sistema de BI para ellos, ya que sólo va en beneficio de ellos mismos.

El hecho de que se haya logrado que se satisfagan los requerimientos de los usuarios va de la mano con el control del proyecto de desarrollo que fue llevado a cabo usando la Ingeniería de Software, ya que ayudó a tener una correcta especificación de requisitos para que luego éstos no sean modificados en etapas avanzadas del proyecto. También esto fue reforzado por la primera auditoría hecha al sistema, luego de la etapa de diseño del sistema, en la que se vio si se había capturado lo que los usuarios finalmente deseaban o no. Tras la confirmación de esto, se pudo seguir adelante con el desarrollo del resto del proyecto, confiando en lograr la satisfacción de los requerimientos de los usuarios.

En estos momentos, el sistema se encuentra en un estado estable, ejecutando los procesos ETL sin problemas, con lo que los datos son refrescados a la tasa deseada (1 vez al día) y también se pueden ejecutar las veces que se desee si es que fuera necesario. Con esto, los informes siempre estarán actualizados con la última información que se ha ingresado a la base de datos del ERP de la compañía. Día a día el personal que trabaja en el área de compras puede acceder a los informes y obtenerlos en muy poco tiempo.

Luego de tener el sistema de BI implementado para el área de compras de la empresa, se espera que en el corto plazo se vayan integrando otras áreas a este proyecto, ya que sin duda alguna será beneficioso para toda la compañía.

# **GLOSARIO DE TÉRMINOS**

- **3FN** 3ra Forma Normal, del estándar para normalizar bases de datos de Boyce-Codd. En esta etapa de la normalización se eliminan las dependencias funcionales sobre los campos no claves, colocándolos en tablas separadas. Al finalizar esta etapa, todos los campos no claves son dependientes del campo clave.
- Character Set
  Es un esquema de codificación en el cual cada caracter se representa por un valor binario diferente. Por ejemplo, ISO8859-1 es un *character set* Latín extendido, que soporta más de 40 lenguajes del oeste de Europa.
- COM Component Object Model. Es un modelo de programación orientado a objetos que define cómo interactúan los objetos dentro de una aplicación o entre aplicaciones. Tanto OLE como ActiveX están basados en COM.
- CPU *Central Process Unit.* Es el cerebro de un computador y el lugar donde se realizan la mayoría de los cálculos. Es el elemento más importante de un computador.
- **CWM** *Common Warehouse Meta-model.* Estándar utilizado por las herramientas que generan metadatos para facilitar el intercambio de información entre ellas.
- DADDatabase Access Descriptor. El descriptor de acceso a la basede datos contiene la información que necesitan los productos

Oracle Portal, WebDB u Oracle Application Server para conectarse a una base de datos Oracle. La información que se incluye es: nombre de usuario, contraseña y el *string* de conexión Net8 (servidor:puerto:SID).

- DBA DataBase Administrator. Usuario de la base de datos que es capaz de administrar a los demás usuarios y objetos que existan.
- DDLData Definition Language. Es un lenguaje usado por elsistema de administración de la base de datos el cual permite alos usuarios definir bases de datos, especificar tipos de datos,estructuras y restricciones sobre los datos.
- **DDR**Double Data Rate. Tipo de memoria que permite realizar dosoperaciones durante un solo ciclo de reloj.
- Data Mart
   Es un subconjunto de información relevante a un grupo de usuarios que se traslada a un servidor departamental. La base de datos puede ser relacional, aunque un servidor OLAP multidimensional es a menudo más apropiado.
- Data MiningEs el proceso de detectar patrones ocultos, tendencias y<br/>asociaciones dentro de los datos almacenados en grandes bases<br/>de datos, tal como un DW. Los métodos típicamente usados<br/>son: redes neuronales, reconocimiento de patrones y análisis<br/>estadístico.

- **Dimension Table** Es una tabla, típicamente en un DW, que contiene más información acerca de un atributo en un tabla de hechos.
- **DSS** *Decision Support System*, es un sistema interactivo basado en computadores que tiene la intención de ayudar a los que toman decisiones a utilizar los datos y modelos para identificar y solucionar distintos problemas.
- **DOS**Disk Operating System. Sistema Operativo de Disco. Términoque puede usarse para referirse a muchos sistemas operativos.
- DTSData Transformation Services.Utilidad de Microsoft SQLServer que permite realizar un proceso ETL.
- *Data Warehouse* Una base de datos separada dedicada al soporte para la toma de decisiones. Los datos se transfieren de sistemas de procesamiento de transacciones y son integrados. Es una arquitectura de software.
- **ERP** *Enterprise Resource Planning*. Es un sistema de información que integra todo el proceso de manufactura y las aplicaciones relacionadas para una empresa.
- ETL Extract / Transform / Load data. Es el proceso de adquisición de datos para un DW o un DM. En este proceso se extrae, transforma y se cargan los datos desde distintas fuentes a un destino común.

- EULEnd User Layer. Es la capa que interactúa sobre la base de<br/>datos. A través del EUL los distintos usuarios de Discoverer<br/>son capaces de extraer información para realizar sus análisis.
- Fact TableTabla de hechos. Es una tabla, típicamente en un DW o DM,que contiene las medidas y los hechos (es el dato principal).

**GB** 1 GB (Gigabyte). Son 1024 MB.

- HTML *HyperText Markup Language*. Es un estándar para la publicación electrónica. Es el estándar utilizado en la *World Wide Web* (WWW).
- **HTTP** *HyperText Transfer Protocol.* Es el protocolo de comunicaciones actual que permite la visualización en los distintos browsers Web.
- Hypertext
   Es un dato textual que se enlaza a través de varios documentos o ubicaciones.
- Informix Es un sistema de administración de base de datos relacional recientemente comprado por IBM. Se espera que IBM incorpore Informix dentro de DB2.

Init.oraArchivo de inicialización de parámetros de Oracle (similar al<br/>DSNZPARM de DB2).

Instancia Una instancia Oracle es una base de datos Oracle ejecutándose, compuesta de estructuras de memoria y procesos de fondo. Una instancia solamente existe cuando la base de datos está arriba y ejecutándose. Una base de datos reside en disco, mientras que una instancia reside en memoria.

**KB** 1 KB (Kilobyte) que equivale a 1024 bytes.

- LAN Local Area Network. Permite la transmisión de datos entre dispositivos conectados en una sola instalación. Estos dispositivos pueden ser por ejemplo: computadores, terminales, impresoras, unidades de almacenamiento, etc.
- Metadata Metadato. Es dato que se usa para describir otro dato.

MB 1 Megabyte, que equivale a 1024 KB.

- MDLMetadata Definition Language. Es un lenguaje usado para<br/>definir metadatos con el OWB OMB (Oracle Metadata<br/>Builder), herramienta que trae OWB para crear y manipular<br/>objetos de metadato.
- MLSMulti Language Support. Permite a una aplicación ejecutarseen más de un lenguaje o en el mismo de la instancia Oracle.
- MOFMeta Object Facility. Es un estándar adoptado por la OMGpara representar y administrar metadatos.

MS Microsoft. Conocida empresa proveedora de productos de software.

NLS National Language Support. Se usa para definir la configuración de la fecha nacional, número, moneda y lenguaje.

- OC4J Oracle Containers for Java/J2EE. Permite un ambiente de ejecución para aplicaciones J2EE y viene con productos Oracle como iAS o Jdeveloper. El OWB Runtime Audit Browser también usa OC4J.
- ODBC Open DataBase Conectivity. Estándar desarrollado por Microsoft que ofrece conectividad en un amplio rango de bases de datos.
- **OEM** *Oracle Enterprise Manager.* Es un producto usado para controlar, monitorear y administrar bases de datos y otros servidores en un ambiente Oracle.
- OLAP
   OnLine Analytical Processing. OLAP usa una vista multidimensional de agregación de datos, de forma de dar un rápido acceso a información estratégica para futuros análisis.
   OLAP y los DW son complementarios. Un DW almacena y administra datos. Un OLAP transforma estos datos en información estratégica.
- OLE DB Object Linking and Embedding DataBase. Es una forma de acceso a datos para comunicarse tanto con bases de datos relacionales como las no relacionales.
- **OMG** *Object Management Group.* Es un consorcio sin fines de lucro que produce y mantiene especificaciones de la industria computacional para la interoperabilidad de aplicaciones empresariales.

- OMSOracle Management Server. Es parte de la arquitectura OEM.El OMS es la capa intermedia entre el OEM y la base de datos.
- Oracle\_Home Es una variable de ambiente que apunta al directorio bajo el cual se instalan los productos Oracle.
- OWBOracle Warehouse Builder. Producto de Oracle que sirve para<br/>desarrollar ambientes de DW o DM.
- RAM Random Access Memory. Memoria de acceso directo, que como lo indica su nombre, puede accederse a la ubicación exacta desde donde se encuentran los datos. Esta memoria es volátil, es decir, que si se corta el suministro de energía, se borra completamente.
- SchemaUn esquema es un conjunto de objetos (tablas, vista, índices,<br/>etc.) perteneciente a una cuenta. A menudo se usa como una<br/>forma de referirse a una cuenta Oracle.
- SIDSystem ID. Identificador de sistema con el que se puedenombrar una instancia y servirá para hacer referencia a ella.

**Snapshot** Es una copia de una tabla en un sistema remoto.

SQL Structured Query Language. Es un lenguaje que provee una interfaz a sistema de bases de datos relacionales. Fue desarrollado por IBM en los años 70 para usarlo en System R.

SYSDBASYStem DataBase Administrator. Es el primer usuario de la<br/>base de datos. Este usuario se genera al momento de creación

de la instancia y posee todos los permisos y privilegios para hacer lo que sea con la base de datos.

- TablaEs una colección de datos que son organizados, definidos y<br/>almacenados en filas y columnas. En sistemas no relacionales,<br/>una tabla se llama archivo.
- **TCP/IP**Transmission Control Protocol / Internet Protocol. Es un<br/>protocolo para la transmisión de datos a través de Internet.
- **TNSNames.ora** Es una archivo de texto ASCII que contiene información sobre la localización del servidor y las cadenas de conexión necesarias para conectarse a bases de datos Oracle.
- UML Unified Modeling Language. Es un estándar del OMG usado para el modelamiento de software. Usando UML, los desarrolladores y arquitectos pueden hacer un modelo de un proyecto.
- XMI XML-Based Metadata Interchange. Es un estándar adoptado por la OMG para un formato de intercambio de metadato que se basa en la especificación XML de la W3C.
- XML eXtensive Markup Language. Es una iniciativa de W3C que permite que la información y los servicios puedan ser codificados con una estructura y semántica que puedan ser entendidos por humanos y computadores.

## BIBLIOGRAFÍA

[1] ANTONIO, Angélica de. Gestión, control y garantía de la calidad del software.
 [en línea] <a href="http://www.inf.uach.cl/rvega/asignaturas/info265/">http://www.inf.uach.cl/rvega/asignaturas/info265/</a>
 G\_Calidad.pdf> [consulta: 20 de noviembre de 2004]

[2] ASCENTIAL SOFTWARE. Ascential Datastage Features and Benefits. [en
 línea] http://www.ascential.com/products/ds\_features.html [consulta: 22 de
 Noviembre de 2004]

[3] BARRY & ASSOCIATES. Common Warehouse Meta-Model (CWM). [en línea] <a href="http://www.service-architecture.com/web-services/articles/">http://www.service-architecture.com/web-services/articles/</a> common\_warehouse\_metamodel\_cwm.html> [consulta: 22 de Noviembre de 2004]

[4] BOHEM, Barry W. Software Risk Management: Principles and Practices.IEEE Software. Páginas:32-41. Enero 2001.

[5] BUSINESS OBJECT SA. BussinessObjects Data Integrator. [en línea] <http://www.businessobjects.com/products/dataintegration/dataintegrator/default.asp > [consulta: 22 de Noviembre de 2004]

[6] CHANDRAS, Radjan. (Re)Enter the Leader?. [en línea] Intelligent Enterprise. 10 de Julio de 2004. <a href="http://www.intelligententerprise.com/">http://www.intelligententerprise.com/</a> showArticle.jhtml?articleID=22102279> [consulta: 18 de Noviembre de 2004]

[7] CHANDRAS, Radjan. Review: Informatica PowerCenter 7.0 [en línea] bizintelligencepipeline. 21 de Julio de 2004. <http://www.bizintelligencepipeline.com/GLOBAL/btg/pipeline/shared/article/show Article.jhtml?articleId=23903761&pgno=2> [consulta: 18 de Noviembre de 2004] [8] COGNOS INCORPORATED. Cognos DecisionStream. [en línea] Burlington, MA, Estados Unidos. <a href="http://datawarehouse.ittoolbox.com/documents/">http://datawarehouse.ittoolbox.com/documents/</a> document.asp?i=1285> [consulta: 22 de Noviembre de 2004]

[9] COGNOS INCORPORATED. Cognos DecisionStream System Requirements. [en línea] <http://www.cognos.com/products/business\_intelligence/ data\_preparation/systemrequirements.html> [consulta: 22 de Noviembre de 2004]

[10] COMMON WAREHOUSE METAMODEL. The Data Warehousing, CWM
 And MOF Resource Page. [en línea] <a href="http://www.omg.org/cwm/">http://www.omg.org/cwm/</a> [consulta: 22 de
 Noviembre de 2004]

[11] COMMON WAREHOUSE METAMODEL. Common Warehouse Metamodel (CWM) Specification. [en línea] Cambridge, MA, Estados Unidos. <a href="http://www.omg.org/cgi-bin/pps/doc?ad/01-02-01.pdf">http://www.omg.org/cgi-bin/pps/doc?ad/01-02-01.pdf</a>> [consulta: 22 de Noviembre de 2004]

[12] CRYSTAL DECISIONS, INC. Advantages and Disadvantages of Native, ODBC and OLE DB Connections. <en línea> Estados Unidos <http://support.businessobjects.com/communityCS/TechnicalPapers/cr\_connection\_a dvantages.pdf> [consulta: 18 de Noviembre de 2004]

[13] EMBARCADERO TECHNOLOGIES INC. DT/Studio [en línea] <http://www.embarcadero.com/products/dtstudio/> [consulta: 22 de Noviembre de 2004]

[14] EVALTECH RESEARCH GROUP. Choose to Build. [en línea] <http://evaltech.com/tooleval.htm> [consulta: 22 de Noviembre de 2004] [15] GEORGE Mason University. Introduction to software risk and risk management. [en línea] <a href="http://www.baz.com/kjordan/swse625/intro.html">http://www.baz.com/kjordan/swse625/intro.html</a> [consulta: 22 de noviembre de 2004]

[16] GROUP 1 SOFTWARE INC. ETL solutions. [en línea] <http:// www.sagent.com/products/ETL.asp> [consulta: 22 de Noviembre de 2004]

[17] HUMMINGBIRD.DataIntegration.[enlínea]<http://www.hummingbird.com/</td>products/etl/index.html?cks=y>[consulta: 22 deNoviembre de 2004]

[18] IBM CORPORATION. DB2 Cube Views: Guía y Consulta. [en línea] Estados Unidos <a href="http://publib.boulder.ibm.com/infocenter/db2help/index.jsp?">http://publib.boulder.ibm.com/infocenter/db2help/index.jsp?</a> topic=/com.ibm.db2.udb.db2\_olap.doc/db2\_olap/cmdrelationalex.htm> [consulta: 22 de Noviembre de 2004]

[19] IBM CORPORATION. DB2 Warehouse Manager: Overview. [en línea] <http://www-306.ibm.com/software/data/db2/datawarehouse/> [consulta: 22 de Noviembre de 2004]

[20] IBM CORPORATION. DB2 Warehouse Manager: Features and Benefits [en
 línea] <a href="http://www-306.ibm.com/">http://www-306.ibm.com/</a> software/data/db2/datawarehouse/>
 [consulta: 22
 de Noviembre de 2004]

[21] INFORMATICA CORPORATION. Informatica PowerCenter. [en línea] <http://www.informatica.com/products/powercenter/default.htm> [consulta: 18 de Noviembre de 2004]

[22] INFORMATICA CORPORATION. Informatica voted number one for data integration by readers of Intelligent Enterprise. [en línea]

<http://www.informatica.com/ news /press\_releases/2004/08252004a\_intel\_ent.htm> [consulta: 18 de Noviembre de 2004]

[23] INMON, Bill. Data Mart does not equal to Data Warehouse. [en línea]
DMReview. 20 de Noviembre de 1999. <a href="http://www.dmreview.com/">http://www.dmreview.com/</a>
article\_sub.cfm?articleId=1675> [consulta: 18 de Noviembre de 2004]

[24] JARRAD, Geoff. Business Intelligence - What is it? [en línea] Mathematical and Information Sciences. 28 de Mayor de 2003. <a href="http://www.cmis.csiro.au/bi/whatis-BI.htm">http://www.cmis.csiro.au/bi/whatis-BI.htm</a>> [consulta: 22 de Noviembre de 2004]

[25] KAYRONLTDA.BusinessIntelligence.[enlínea]<http://www.kairon.com/refbi.htm>[consulta: 22 de Noviembre de 2004]

[26] KIMBALL, Ralph y ROSS, Margy. The Datawarehouse Toolkit. Segunda Edición. New York, Estados Unidos. Wiley Computer Publishing. 2002. 447p.

[27] MADSEN, Mark. Criteria for ETL Product Selection. [en línea] DMReview.
1 de Octubre de 2004. <a href="http://www.dmreview.com/portals/portalarticle.cfm?articleId">http://www.dmreview.com/portals/portalarticle.cfm?articleId</a>
=1011217&topicId=230206> [consulta: 18 de Noviembre de 2004]

[28] Microsoft MSDN Library. Data Transformation Services. [en línea] Estados Unidos. <http://msdn.microsoft.com/library/default.asp?url=/library/enus/dtssql/dts\_tools\_des\_07xh. asp> [consulta: 18 de Noviembre de 2004] NAUDÉ, [29] Frank. Glossary of Terms: Α [en línea] <a href="http://www.orafaq.net/glossary/faqglosa.htm">http://www.orafaq.net/glossary/faqglosa.htm</a>> [consulta: 22 de Noviembre de 2004] ORACLE CORPORATION. Oracle9i Database New Features - Release 2 [30] (9.2). [en línea] <http://www.utexas.edu/its/unix/reference/oracledocs/v92/ B10501 01/server.920 /a96531/ch1 over.htm> [consulta: 22 de Noviembre de 2004]

[31] ORACLE CORPORATION. Oracle9i Data Warehousing Guide Release 2
(9.2). [en línea] <<u>http://download-west.oracle.com/docs/cd/B10501\_01/server.920/</u>
a96520/toc.htm> [consulta: 18 de Noviembre de 2004]

[32] ORACLE CORPORATION. Applying Patch Sets with Phisical Standby Database in Place. [en línea] Redwood Shores, CA, Estados Unidos. Nota: 187242.1 <http://metalink.oracle.com/metalink/plsql/ml2\_documents.showDocument?p\_datab ase\_id=NOT&p\_id=187242.1> [consulta: 22 de Noviembre de 2004]

[33] ORACLE CORPORATION, Listener Status Fails with NL-00857 from Enterprise Manager. [en línea] Redwood Shores, CA, Estados Unidos. Nota: 201252.1 <a href="http://metalink.oracle.com/metalink/plsql/ml2\_documents.show">http://metalink.oracle.com/metalink/plsql/ml2\_documents.show</a> Document?p\_database\_id=NOT&p\_id=201252.1> [consulta: 22 de Noviembre de 2004]

[34] ORACLE CORPORATION. Setting up OEM 2.2 to work with OWB 3x. [en línea] Redwood Shores, CA, Estados Unidos. Nota: 169922.1 <http://metalink.oracle.com/metalink/plsql/ml2\_documents.showDocument?p\_datab ase\_id=NOT&p\_id=169922.1> [consulta: 22 de Noviembre de 2004]

[35] ORACLE CORPORATION. Which Oracle Client versions will connect to and work against which version of the Oracle Database? [en línea] Redwood Shores, CA, Estados Unidos. Nota: 172179.1 <http://metalink.oracle.com/metalink/plsql/ml2\_documents.showDocument?p\_datab ase\_id=NOT&p\_id=172179.1> [consulta: 22 de Noviembre de 2004]

[36] ORACLE CORPORATION. How to Start the EM 2 Management Server at Bootup. [en línea] Redwood Shores, CA, Estados Unidos. Nota: 107276.1 <http://metalink.oracle.com/metalink/plsql/ml2\_documents.showDocument?p\_datab ase\_id=NOT&p\_id=107276.1> [consulta: 22 de Noviembre de 2004]

[37] ORACLE CORPORATION. The correct NLS\_LANG in a Windows Environment. [en línea] Redwood Shores, CA, Estados Unidos. Nota: 179133.1 <http://metalink.oracle.com/metalink/plsql/ml2\_documents.showDocument?p\_datab ase\_id=NOT&p\_id=179133.1> [consulta: 22 de Noviembre de 2004]

[38] ORACLE CORPORTAION. An Overview of the development phases of OWB. [en línea] Redwood Shores, CA, Estados Unidos. Nota: 106456.1 <http://metalink.oracle.com/metalink/plsql/ml2\_documents.showDocument?p\_datab ase\_id=NOT&p\_id=106456.1> [consulta:22 de Noviembre de 2004]

[39] ORACLE CORPORATION. OWB 9.2 Runtime Repository Installation Fails with ORA-06502. [en línea] Redwood Shores, CA, Estados Unidos. Nota: 243318.1 <http://metalink.oracle.com/metalink/plsql/ml2\_documents.showDocument?p\_datab ase\_id=NOT&p\_id=243318.1> [consulta: 22 de Noviembre de 2004]

[40] ORACLE CORPORATION. Oracle9i Warehouse Builder - User's Guide -Release 9.2. [en línea] Redwood Shores, CA, Estados Unidos. <a href="http://download.oracle.com/docs/pdf/B10996\_01.pdf">http://download.oracle.com/docs/pdf/B10996\_01.pdf</a>> [consulta: 22 de Noviembre de 2004]

[41] ORACLE CORPORATION. Oracle9i Warehouse Builder Installation and Configuration Guide - Release 9.2.0.3.0 [en línea] Redwood Shores, CA, Estados Unidos. <a href="http://download.oracle.com/docs/pdf/B11000\_02.pdf">http://download.oracle.com/docs/pdf/B11000\_02.pdf</a>> [consulta: 22 de Noviembre de 2004] [42] ORACLE CORPORATION. Oracle9i Warehouse Builder Release Notes -Release 9.2 [en línea] Redwood Shores, CA, Estados Unidos <a href="http://download.oracle.com/docs/pdf/B10999\_01.pdf">http://download.oracle.com/docs/pdf/B10999\_01.pdf</a>> [consulta: 22 de Noviembre de 2004]

[43] ORACLE CORPORATION. Oracle9i Data Server 9.0.1 Patch Set 4 for Windows NT. [en línea] Redwood Shores, CA, Estados Unidos. <http://updates.oracle.com/ARULink/PatchDetails/process\_form?patch\_num=25173 00&release=8090140&plat\_lang=912P&email=mbarrientos%40salmofood.cl&useri d=mbarrientos&> [consulta: 22 de Noviembre de 2004]

[44] ORACLE CORPORATION. Oracle9i Database Release Notes Release 1
(9.0.1.1.1) for Windows XP. [en línea] Redwood Shores, CA, Estados Unidos. Nota:
183076.1. <a href="http://metalink.oracle.com/metalink/plsql/showdoc?db=NOT&id=18">http://metalink.oracle.com/metalink/plsql/showdoc?db=NOT&id=18</a>
3076.1> [consulta: 22 de Noviembre de 2004]

[45] ORACLE CORPORATION. Oracle 9i Discoverer Administrator Administration Guide [en línea] Redwood Shores, CA, Estados Unidos. <a href="http://oraclelon1.oracle.com/docs/cd/A91773\_01/ids902dl/bi.902/a9088102/a90881">http://oraclelon1.oracle.com/docs/cd/A91773\_01/ids902dl/bi.902/a9088102/a90881</a> \_02.pdf> [consulta: 22 de Noviembre de 2004]

[46] ORACLE CORPORATION. Oracle 9i Discoverer Desktop User's Guide [en línea] Redwood Shores, CA, Estados Unidos. <a href="http://oraclelon1.oracle.com/docs/cd/A91773\_01/ids902dl/bi.902/a9088101/a90886\_01.pdf">http://oraclelon1.oracle.com/docs/cd/A91773\_01/ids902dl/bi.902/a9088101/a90886\_01.pdf</a>
[consulta: 22 de Noviembre de 2004]

[47] ORACLE CORPORATION. Oracle9i Warehouse Builder Patch 9.2.0.3. [en línea] Redwood Shores, CA, Estados Unidos. <a href="http://updates.oracle.com/ARULink/">http://updates.oracle.com/ARULink/</a> PatchDetails/process\_form?patch\_num=3152515&release=8092030&plat\_lang=200 0P&email=mbarrientos%40salmofood.cl&userid=mbarrientos&> [consulta: 22 de Noviembre de 2004]

[48] ORACLE CORPORATION. RTA Fails with German Locale on the Desktop.
[en línea] Redwood Shores, CA, Estados Unidos. Parche: 3043523
<a href="http://www.metalink.oracle.com/metalink/plsql/ml2\_gui.startup">http://www.metalink.oracle.com/metalink/plsql/ml2\_gui.startup</a>> [consulta: 22 de
Noviembre de 2004]

[49] PERVASIVE SOFTWARE INC. Pervasive Data Integrator. [en línea]
<a href="http://www.pervasive.com/dataintegrator/>[consulta: 22 de Noviembre de 2004]">http://www.pervasive.com/dataintegrator/>[consulta: 22 de Noviembre de 2004]</a>
[50] RENGARAJAN, Jayaraj. [en línea] Differences between UTF8 vs
AL16UTF16 vs AL32UTF8. Oracle Corporation. Oracle Enterprise Manager and
Management Packs Technical Forum. <a href="http://metalink.oracle.com/metalink/plsql/ml2\_documents.showDocument?p\_database\_id=FOR&p\_id=438466.995>">http://metalink.oracle.com/metalink/plsql/ml2\_documents.showDocument?p\_database\_id=FOR&p\_id=438466.995></a> [consulta: 22 de Noviembre de 2004]

[51] RITTMAN, Mark. What are the advantages of the star schemas over other design patterns? From a performance point of view, why are the star schemas is better than snowflakes?. [en línea] <a href="http://www.rittman.net/archives/000512.html">http://www.rittman.net/archives/000512.html</a> [consulta: 18 de Noviembre de 2004]

[52] SALAZAR, Juan Pablo. Gestión de configuración. [Diapositivas] Valdivia[2004]

[53] SAS INSTITUTE INC. SAS ETL Studio. [en línea] <http://www.sas.com/technologies/dw/etl /etlstudio/> [consulta: 22 de Noviembre de 2004] [54] SELECTOR, Lev. Selectorweb: Informatica. [en línea] <http://www.selectorweb.com /informatica.html> [consulta: 22 de Noviembre de 2004]

[55] SOLANO, Alex. Eliminar el rastreador de sucesos de apagado en Windows
Server 2003. [en línea] <a href="http://windowsxp.ethek.com/WindowsXP/contenido.asp">http://windowsxp.ethek.com/WindowsXP/contenido.asp</a>
?IDContenido=711> [consulta: 18 de Noviembre de 2004]

[56] SONGINI, Marc. ETL. [en línea] <a href="http://www.computerworld.com/databasetopics/businessintelligence/datawarehouse/story/0,10801,89534,00.html">http://www.computerworld.com/databasetopics/businessintelligence/datawarehouse/story/0,10801,89534,00.html</a>
[consulta: 22 de Noviembre de 2004]

[57] UNIVERSITY SOUTH CAROLINA. USC Cocomo II Reference Manual.
[en línea] <a href="http://www.inf.uach.cl/rvega/asignaturas/info265/Keun\_lee/userman.pdf">http://www.inf.uach.cl/rvega/asignaturas/info265/Keun\_lee/userman.pdf</a>
[consulta: 22 de Noviembre de 2004]

[58] VEGA, Raimundo. Ciclo de vida y proceso de software. [Diapositivas]Valdivia. [2001]

[59] VEGA, Raimundo. Control de Riesgo. [Diapositivas] Valdivia [2001]

[60] VEGA, Raimundo. Introducción a la Ingeniería de Requisitos. [Diapositivas]Valdivia [2002]

[61] WOLFF, Carmen Gloria. Modelamiento Multidimensional. [en línea] Revista Ingeniería Informática. 28 de Agosto de 2002. <a href="http://www.inf.udec.cl/">http://www.inf.udec.cl/</a> revista/ediciones/edicion4/modmulti.PDF> [consulta: 22 de Noviembre de 2004]

# ANEXO A: ESPECIFICACIÓN DE REQUERIMIENTOS MÍNIMOS Y

## HARDWARE USADO

# A.1 REQUERIMIENTOS MÍNIMOS DE EQUIPOS

Para poder utilizar el Oracle Database 9.0.1.4.0 se deben cumplir los siguientes requisitos:

- Procesador Pentium de 266 Mhz o superior.
- Memoria RAM superior o igual a 256 MB.
- Sistema operativo Windows NT 5 Service Pack 5 o superior, Windows 2000 Service Pack 1, Windows XP Professional, Windows Server 2003 Enterprise.

Para poder utilizar el Oracle Warehouse Builder 9.2.0.3.0 se deben cumplir

los siguientes requisitos:

- Procesador Pentium de 500 Mhz o superior.
- 800 MB de espacio libre en disco duro.
- 384 MB de memoria RAM disponible.
- Windows NT 4 SP6 o superior, Windows 2000, Windows XP Professional Edition.
- Base de datos en un servidor, con versión 8.1.7 o superior.

Para poder utilizar el Oracle9i Discoverer Administrator 9.0.2.0.0 se deben

cumplir los siguientes requisitos:

- Procesador Pentium de 500 Mhz o superior.
- 128 MB de memoria RAM disponible.
- Sistema operativo NT/2000/XP.

• 1.7 GB de espacio libre en disco duro (esto es para instalar la suite completa de Oracle9i Developer Suite).

Para poder utilizar el Oracle9i Discoverer Desktop 9.0.2.0.0 se deben cumplir los mismos requisitos que para el Administrator. Esto no es impedimento para los equipos que usan los usuarios de Salmofood S.A., ya que estos son de características similares a la del equipo Cliente que se describe más abajo, en el punto A.2.

#### A.2 DESCRIPCIÓN DE EQUIPOS USADOS

Las características básicas de los equipos y que son las que importan para poder instalar y ejecutar los programas necesarios para desarrollar y trabajar con el DM se detallarán a continuación, pero se obviarán descripciones que no son importantes, como por ejemplo: tarjeta de video, teclado, mouse, monitor u otros software instalados en el equipo y que no tienen relación con el tema tratado.

Para el servidor se ha usado un equipo con las siguientes características:

- Microsoft Windows Server 2003 Enterprise.
- Procesador Intel Pentium 4 2.66 Ghz.
- Tarjeta de Red BCM5703 Gigabit.
- 1.25 GB de memoria RAM DDR2700.
- Disco duro de 80 GB 7200 rpm.

Para respaldar la información del servidor se usan los respaldos automáticos en disco durante esta primera etapa, ya que la empresa tiene previsto un servidor especial dedicado a mantener un DW, el cual tendrá incorporado una unidad de Tape Backup, al igual que los demás servidores que usa en la actualidad.

Para el cliente se ha usado un equipo con las siguientes características:

- Microsoft Windows XP Professional.
- Procesador Intel Pentium 4 2.4 Ghz.
- 512 MB de memoria RAM DDR3200.
- Tarjeta de Red Realtek RTL8139 (100 mbps).
- Disco duro de 60 GB 7200 rpm.

Se puede apreciar que el equipo Servidor posee las características necesarias para poder manejar sin problemas la base de datos y el OWB. En tanto que el equipo Cliente, tiene el potencial para poder trabajar sin problemas con el Cliente de Oracle Warehouse Builder y el Oracle9i Discoverer Administrator.

#### A.3 ORIGEN DE LOS DATOS

El servidor en el que se encuentra la base de datos, llamado internamente Dataserver, que tiene la información de origen para el actual caso de estudio es un Dell PowerEdge 6400, con 4 GB de memoria RAM, sistema operativo Windows NT4 SP6 y se encuentra en una red de datos que transmite a 1 Gbps. En este servidor se encuentra instalada la base de datos Oracle 7.3.4.5.0, la que soporta tres instancias de Oracle Financials 10.7.NCA. Las tres instancias son: Producción (PDFO), Prototipo (PDPE) y Test (TEST), donde los nombres encerrados en paréntesis son los SID de cada instancia. La instancia desde la cual se extraerán los datos necesarios para llenar el DM que se está desarrollando es Producción (PDFO). El usuario, con el cual se tienen los permisos para poder extraer datos de todos los esquemas de usuarios es APPS. En la figura A.1 se puede apreciar un esquema de la red de datos que actualmente posee Salmofood S.A. Se puede identificar claramente la red de 1 Giga en la cual están conectados los servidores de la compañía.



Figura A.1: Esquema de red de Salmofood S.A.

#### ANEXO B: ACERCA DE LA BASE DE DATOS

# B.1 ORACLE DATABASE 9.0.1.4.0

Inicialmente la base de datos elegida fue Oracle Database 9.0.1.1.1. Lo que influyó en que se usara una base de datos 9.0.1.1.1 y no una versión 9.2 fue que el resto de las aplicaciones de la empresa Salmofood S.A. se estaban construyendo sobre una base de datos de esta versión y se quiso seguir con esta línea de trabajo. Esta consideración se tomó con las personas que trabajan en el Departamento de Informática y Tecnología afectando solamente a las bases de datos, ya que el resto de las aplicaciones que se usan en la empresa, como Oracle Application Server y Oracle Developer Suite se usan en las últimas versiones disponibles en el mercado para el caso de desarrollo de nuevas tecnologías dentro de la compañía. Más tarde, al momento anterior a la instalación del software OWB, se estuvo en la obligación de migrar la base de datos a una versión un poco superior, la 9.0.1.4.0. El motivo de esta migración fue que al usar el OWB 9.2.0.3.0 se necesitaba una versión de base de datos 9.0.1.3.0 como mínimo.

Lo que significa cada número cuando se habla de una versión sobre un producto Oracle se detalla a continuación:

- El primer número representa el número de la versión del producto.
- El segundo número representa nuevas características sobre el número de la versión del producto.
- El tercer número representa mantenimiento sobre el número de la versión.
- El cuarto número representa un parche genérico sobre el producto.

• El quinto número representa un parche específico de la plataforma.

La arquitectura de base de datos Oracle9i es de tres capas. Un esquema que puede mostrar como trabajan las aplicaciones Oracle a grandes rasgos se ve en la figura B.1.



Figura B.1: Arquitectura de 3 capas de la base de datos Oracle

En esta figura se puede apreciar cómo las máquinas Clientes (capa 1) se conectan a través del protocolo HTTP con Oracle Application Server (capa 2). A su vez, el Oracle Application Server se comunica directamente con la base de datos Oracle (capa 3). En tanto, el Oracle Developer Suite se comunica con las tres capas directamente.

En el esquema que se implementará para desarrollar el DM no se instalará el Application Server, por lo que el Cliente usará las herramientas de Developer Suite para comunicarse con la base de datos y obtener la información que necesita.

# B.1.1 CONSIDERACIONES ANTES DE INSTALAR LA BASE DE DATOS

La base de datos se instalará en el servidor Basilisco que ya se ha descrito en apartados anteriores. Antes de instalar la base de datos es necesario tener algunas consideraciones para que no existan problemas de compatibilidad con las que ya existen y desde las cuales se extraerán los datos:

- Configuración de los parámetros de NLS del servidor.
- Parámetros de configuración que irán en el archivo *init.ora* de la base de datos.
- Configuración regional y de idioma del servidor.

Los dos primeros puntos son bastante importantes, ya que pueden presentar muchísimos problemas de comunicación con otras bases de datos u aplicaciones que se encuentren configuradas de una forma distinta. Incluso existen parámetros que no se pueden cambiar una vez que se crea la base de datos, como por ejemplo el *db\_block\_size*. También existen otros como el *nls\_lang* que pueden ser manipulados en el registro de Windows.

Para poder configurar correctamente los parámetros NLS del servidor, se ha visto la configuración que tiene actualmente la base de datos que contiene el origen de la información. De esta forma, se tratará de que la nueva base de datos, tenga los mismos parámetros de configuración que la base de datos de origen.

De la misma forma, se tratará de mantener la configuración regional y de idiomas del equipo de forma similar a la que posee el servidor de base de datos que contiene la información de origen.

## **B.2** INSTALACIÓN DE LA BASE DE DATOS

Una vez que se han tomado las precauciones necesarias para poder hacer una instalación de la base de datos que sirva para mantener el DM se puede comenzar a instalarla sin problemas.

Los pasos a seguir para crear la base de datos son los siguientes:

**Paso 1**: Ir a Configuración Regional y de Idioma, donde se deben dejar todos los parámetros configurados de igual forma que el servidor y después reiniciar el equipo para asegurar de que los cambios han surtido efecto.

Paso 2: Iniciar el *Oracle Universal Installer* para iniciar la instalación de la base de datos.

**Paso 3**: Seleccionar que tipo de base de datos se instalará. En este caso particular será una base de datos para BI, la cual viene con parámetros configurados por defecto para soportar esta tecnología.

**Paso 4**: Introducir el nombre global de la base de datos y un SID (Identificador del sistema Oracle) para ella. En este caso se ha escogido el nombre BDWH para los dos parámetros.

**Paso 5**: En la siguiente ventana seleccionar el modo del servidor. En este punto se puede elegir entre los modos *Dedicado* y *Compartido*. En el modo de servidor dedicado, la base de datos asignará un recurso dedicado exclusivo para cada cliente. Se recomienda utilizar este modo cuando el número de clientes que se conecte a la base de datos sea pequeño o cuando los clientes que se conecten lo hagan por largos periodos de tiempo. En el modo de servidor compartido, varios clientes comparten un conjunto de recursos asignados a la base de datos. Se recomienda utilizar este

modo cuando el número de usuarios que se conecte a la base de datos sea muy grande. Para el caso particular que se está estudiando se utilizará un modo de servidor dedicado, ya que los usuarios no se conectarán a cada instante a la base de datos y solamente lo ocuparán el personal de compras y la Gerencia de Administración y Finanzas.

**Paso 6**: Este paso es sin duda alguna de las más importantes en lo que se refiere a la configuración inicial para la creación de la base de datos. Ver Anexo C de este documento para configuraciones de parámetros adicionales. En este punto se configura lo siguiente:

- 1. Memoria
  - La configuración de la memoria que ocupará Oracle se deja por defecto.
  - Se deben modificar los parámetros de inicialización que una vez creada la base de datos no se pueden cambiar, como *DB\_CACHE\_SIZE*. Más adelante se explicará el porque es necesario cambiar este parámetro.
  - Los demás parámetros de inicialización se pueden dejar por defecto, ya que más tarde y en caso de ser necesario, se pueden modificar. Para efectos de la instalación se cambiarán cuando sea necesario. En apartados siguientes se detallarán los parámetros que requieren modificaciones y la forma de hacerlas.
- 2. Archivado

• El Archivado, *Archive Log*, es esencial para bases de datos de producción, donde la pérdida de una transacción podría ser fatal. Para este caso es considerada innecesaria, ya que los datos se recrearan en intervalos de tiempo programadas y en ocasiones que los usuarios así lo requieran. Si existe alguna falla en la carga de los datos, existirá la forma de saber que esto ocurrió y se podrán cargar los datos en cualquier momento y casi sin interferencia con la carga de transacciones del ambiente de producción de la compañía.

#### 3. Tamaño de la Base de datos

- En esta ventana se selecciona en primera etapa el tamaño del área de ordenación. Por defecto viene en 1 MB, que es el valor que se deja para la situación actualmente en estudio, ya que no se realizarán ordenaciones frecuentes ni mucho menos grandes.
- Otro punto que se configura en esta etapa es la selección del characterset. Para seleccionar el characterset se eligen:

• **Juego de caracteres de la base de datos**. El juego de caracteres que viene por defecto para la localización que tiene nuestro país y para el sistema operativo que se utiliza es WE8MSWIN1252. Sin embargo, este characterset no sirve para lo que se va a utilizar la base de datos, la cual se tiene que comunicar con otra de diferente characterset, WE8ISO8859P9. Por este motivo, es que este es el momento adecuado para seleccionar el characterset indicado.

 Juego de caracteres nacional. El juego de caracteres nacional puede ser UTF8 ó AL16UTF16. Para el caso en estudio se escogió AL16UTF16.

#### 4. Ubicación de los archivos

 En este punto se configuran las ubicaciones de los archivos de inicialización de la base de datos. Para el caso particular en estudio se dejaron las ubicaciones por defecto.

**Paso 7**: Finalizar la instalación. En este punto se muestra un resumen de todos los datos introducidos y que se usarán para la creación de la base de datos. En este punto se pueden revisar estos datos y cambiarlos si es que se desea o simplemente Seleccionar *Finish* si es que se encuentra que todo está bien para dar comienzo a la instalación.

En este momento, solo resta esperar que se complete la instalación y velar porque las tareas de finalización se realicen correctamente. Es en este momento que se puede cambiar la contraseña de los usuarios: SYS y SYSTEM, además de que se pueden habilitar los usuarios que la base de datos instala por defecto para mostrar los ejemplos.

Completadas todas las actividades que se han descrito ya se puede trabajar con la base de datos. Es ahora cuando se verifica que los parámetros de NLS sean los que se esperaban para poder trabajar sin problemas con la base de datos de origen de información y el servidor remoto que la alberga. Usando el TOAD se ha obtenido que los parámetros están configurados como se muestra a continuación:

• NLS\_LANGUAGE = AMERICAN
- NLS\_TERRITORY = AMERICA
- NLS\_CURRENCY = \$
- NLS\_ISO\_CURRENCY = AMERICA
- NLS\_NUMERIC\_CHARACTERS = .,
- NLS\_CHARACTERSET = WE8ISO8859P9
- NLS\_CALENDAR = GREGORIAN
- NLS\_DATE\_FORMAT = DD-MON-YY
- NLS\_DATE\_LANGUAGE = AMERICAN
- NLS\_SORT = BINARY
- NLS\_DUAL\_CURRENCY = \$
- NLS\_COMP = BINARY
- NLS\_LENGTH\_SEMANTICS = BYTE
- NLS\_NCHAR\_CONV\_EXCP = FALSE
- NLS\_NCHAR\_CHARACTERSET = AL16UTF16

Estos datos coinciden con los que inicialmente tenía la base de datos con el origen de la información.

# B.3 MIGRACIÓN DE LA BASE DE DATOS A LA VERSIÓN9.0.1.4.0

Como se mencionó anteriormente, para poder trabajar con OWB 9.2.0.3.0 se necesitaba una base de datos 9.0.1.3.0 o superior. Al momento de buscar una actualización disponible en el Metalink, se ha encontrado que el último patchset era el 9.0.1.4.0 y a raíz de esto fue que se instaló esta versión. El número del parche es el 2517300. Al bajar el archivo correspondiente al parche y descomprimirlo, se puede apreciar la existencia de un archivo en formato .DOC que contiene la explicación detallada de cómo instalar el parche y que mejoras trae al producto.

Para aplicar el patchset se deben seguir las instrucciones paso a paso que se detallan en el archivo adjunto descargado del Metalink. A modo de referencia, esta información se encuentra como documento electrónico adjunto del presente documento.

Oracle posee una utilidad que se instala junto con la base de datos, llamada Oracle Enterprise Manager (OEM), con la cual se pueden administrar todas las bases de datos Oracle que se encuentren instaladas. Para ejecutar esta utilidad se debe iniciar el Oracle Manager Console y una vez aquí, se puede escoger entre dos modalidades de inicio:

- Launch Standalone.
- Login to the Oracle Management Server.

Con *Launch Standalone*, se realiza una conexión a la consola desde donde se ven todas las bases de datos que se encuentran instaladas y se pueden realizar conexiones para administrar cada una de ellas. Entre las operaciones que se pueden realizar están las siguientes:

- Cerrar e iniciar la base de datos.
- Cambiar los parámetros de inicialización.
- Monitorear las conexiones que existen a la base de datos.
- Ver las transacciones pendientes.
- Administrar los esquemas de los usuarios de la base de datos.
- Administrar a usuarios, roles y perfiles.

- Crear, eliminar o cambiar los tablespaces.
- Crear, eliminar o cambiar los datafiles.

Con el Oracle Management Server (OMS) se pueden realizar todas las operaciones que con Launch Standalone, pero tiene otras utilidades muy prácticas, como son:

- Crear, eliminar y ejecutar trabajos (*Jobs*) en la base de datos.
- Ver el estado del Agente de la base de datos y reiniciarlo si se desea.
- Ver el estado del *Listener* de la base de datos y reiniciarlo si se desea.
- Ver el estado del HTTP Server y reiniciarlo si se desea.

Es la posibilidad de crear y ejecutar *Jobs* lo que es atractivo para el problema que se está solucionando, ya que con esta utilidad se ejecutarán los procesos ETL que se crearán con el OWB más adelante.

Para poder crear y ejecutar *jobs* debe existir un usuario que tenga privilegio de SYSDBA en la base de datos y que además exista como usuario en el grupo de administradores del equipo local. Este usuario debe tener el mismo nombre y contraseña tanto en el sistema operativo como en la base de datos. Además, debe tener un derecho especial y para poder otorgárselo se deben seguir los siguientes pasos:

Paso 1: Ir a Herramientas Administrativas.

Paso 2: Seleccionar Directivas de Seguridad local.

Paso 3: Elegir Directivas Locales y después Asignación de Derechos de Usuario.

**Paso 4**: Se debe editar la opción Inicio de Sesión como Proceso por Lotes y agregar al usuario a este grupo.

Para crear al usuario de la base de datos se puede abrir el Oracle Manager Console y seleccionar la opción *Launch Standalone*. Al moverse por el árbol de navegación que aparece, llega el momento en que el software pide el ingreso de un usuario, contraseña y el modo de conexión del usuario. Para esto es que se usará al usuario SYS, cuya contraseña es MANAGER y el modo de conexión será SYSDBA, de forma de poder crear otro usuario y se le pueda asignar el privilegio de SYSDBA que necesita.

Para crear al usuario con privilegios de SYSDBA se deben seguir los siguientes pasos:

Paso 1: Ir hasta la carpeta Security y expandirla.

**Paso 2**: Seleccionar la carpeta *Users* y hacer click con el botón secundario del mouse y seleccionar *Create*.

Paso 3: Escribir nombre de usuario y contraseña.

Paso 4: Asignarle un *tablespace* para datos y otro temporal.

Paso 5: Asignarle los roles de DBA y CONNECT.

Paso 6: Asignarle el privilegio de SYSDBA.

Paso 7: Dejar el resto de las ventanas por defecto y crear al usuario.

De esta forma ya se puede proceder a configurar el OMS como se explicará a continuación.

Para poder usar el OMS, primero debe usar el *Oracle Enterprise Manager Configuration Assistant* y configurar un OMS local (todo esto en el servidor). Los pasos a seguir para configurar este servidor son los siguientes:

Paso 1: Iniciar el Oracle Enterprise Manager Configuration Assistant.

**Paso 2**: Escoger la opción de *Create a New Repository* y posteriormente *Custom*.

**Paso 3**: Seleccionar una base de datos existente. En este momento se selecciona la base de datos BDWH que es la única instalada.

**Paso 4**: En la siguiente ventana pide ingresar el nombre del usuario dueño del repositorio y la contraseña. Es en este momento cuando se escribe el nombre del usuario con el privilegio de SYSDBA que se acaba de crear.

**Paso 5**: Se pide en la siguiente ventana el nombre de un *tablespace* para el repositorio, el cual se recomienda que sea en uno nuevo y especialmente para el uso del servidor de base de datos.

**Paso 6**: Finalizar la instalación. En este momento, aparece un mensaje del software de instalación en el que se indica que la primera conexión al OMS debe realizarse usando el usuario SYSMAN que se creará solo por el programa y cuya contraseña inicial, OEM\_TEMP, deberá ser cambiada al momento del inicio de sesión.

Ahora que ya se configuró el OMS, éste puede ser iniciado y la contraseña de SYSMAN podrá ser modificada. Para hacer esto, iniciar el *Oracle Manager Console* y seleccionar la opción *Login to the OMS*. A continuación se deben seguir los siguientes pasos:

Paso 1: Escribir como nombre de usuario y contraseña de SYSMAN. Además se debe seleccionar el Management Server, que en este caso es la máquina local, BASILISCO.

**Paso 2**: Lo primero que se debe hacer una vez que se ingresa es agregar al usuario SYSDBA como administrador del servidor. Para hacer esto se debe ir a *Configuration* y escoger la opción *Manage Administrators*. Una vez hecho esto,

escoger la opción *ADD* e ingresar nombre de usuario y la contraseña del usuario. Además se debe seleccionar el *checkbox* que da el permiso de súper usuario.

**Paso 3**: Para finalizar la configuración del usuario se debe seleccionar al usuario recién creado y hacer click en el botón *Edit* para darle permisos especiales. Esto se hace en la ficha *Preferred Credentials*, en la que se debe dar permisos sobre los siguientes elementos:

- *Listener*. Para que se tenga acceso se debe utilizar el nombre de usuario y contraseña de este usuario, que ya tiene privilegio de SYSDBA en la base de datos.
- Nodo. Para que pueda manejar el sistema operativo local, se debe ingresar el nombre de usuario y contraseña de este mismo usuario, al que se agregó al grupo de usuarios que pueden iniciar sesión como proceso por lotes.
- *HTTP Server*. Para esto no se necesitan permisos adicionales.
- Base de datos. Se le puede dar acceso sobre un esquema particular, para lo cual se debe ingresar el nombre de usuario dueño del esquema y su contraseña. Esto será útil más adelante, ya que se le dará el acceso al dueño del esquema que contendrá los datos finales del DM. Como este usuario no está creado todavía se ingresará temporalmente el mismo nombre de usuario y contraseña que se han ingresado hasta el momento y el modo de conexión será SYSDBA.

Ahora que ya se tiene configurado al usuario administrador del servidor, es cuando se prueba de que todo ha funcionado. Para esto se cierra el OMS y se ingresa nuevamente, usando para conectarse a él, al usuario recién configurado. Es acá que se usa el árbol de navegación para ver las bases de datos, *listeners*, servidores HTTP, *Jobs* y Eventos. Desde este árbol de navegación se puede hacer un ping al Agente para ver si se encuentra activo y funcionando. Lo mismo se puede hacer con el HTTP Server y el *Listener*. Es en este momento que se encuentra un error haciendo ping al *listener*. El error que se encuentra es el *NL-00857*, el cual aparece en el Metalink como la nota 201251.1, en la que se explica que este error ocurre por un Bug de la base de datos. El número del Bug es el 1728795, para lo cual la solución es aplicar un pequeño parche, el número 1815492.

Para poder aplicar el parche al OMS se descarga desde el Metalink. Al descomprimir el archivo descargado, se ve que consta de dos carpetas: *Classes y Sysman*. Lo único que hay que hacer es copiar estas dos carpetas dentro del *Oracle Home* que tiene a la base de datos y sobrescribir todos los archivos.

Ahora, se abre nuevamente el OMS y se intenta usar el ping al *listener* obteniendo los resultados deseados.

Finalmente, para probar que todo funcione bien, se recomienda reiniciar el equipo y ver que todos los servicios se inicien de forma normal. Al hacer esto se puede probar que la base de datos se inicia sola sin problemas y que la conexión a través del Oracle Manager Console y la opción *Launch Standalone* funciona normalmente, pero si se trata de iniciar sesión conectándose al OMS, ocurre un error. Esto se debe a que el servicio de OMS no se ha iniciado. Este servicio no se puede iniciar automáticamente como todos los demás, por lo que es necesario iniciarlo manualmente. Para que esto no ocurra cada vez que se reinicia el servidor, se ha

creado un archivo por lotes, el cual se colocará en la parte de inicio del sistema operativo. Este archivo ejecuta la siguiente instrucción:

## oemctl start oms

Con esta instrucción se inicia el servicio cada vez que el computador se reinicie.

Ahora solamente resta un problema, ya que el servidor, al ser Windows 2003 Server, tiene una a utilidad llamada Rastreador de Procesos de Apagado, la que viene configurada por defecto a la hora de instalar el sistema operativo y a la cual hay que ingresar algún caracter cada vez que se quiere Apagar o Reiniciar el equipo. En el caso de que el equipo falle y se apague solo, cuando se inicie nuevamente aparecerá el rastreado preguntando cuál fue la razón de por qué se apagó la última vez, lo que impedirá que el servidor de inicie automáticamente y pueda iniciar todos los servicios, por lo que no se iniciará la base de datos. Para solucionar esto es necesario seguir los siguientes pasos:

Paso 1: Desde ejecutar el siguiente comando: *gpedit.msc*, lo que abrirá el Editor deObjetos de Directivas de Grupo.

Paso 2: Se deben seleccionar las siguientes opciones:

Configuración de Equipos > Plantillas Administrativas > Sistema **Paso 3**: Se debe buscar la opción Rastreador de Sucesos de Apagado, para editarlo y seleccionar la opción *deshabilitada*.

Es en este momento que ya se tiene la base el servidor con la base de datos lista para poder ser utilizada.

#### **ANEXO C: ACERCA DE ORACLE WAREHOUSE BUILDER**

#### C.1 ORACLE WAREHOUSE BUILDER 9.2.0.3

Oracle Warehouse Builder (OWB) es una herramienta de Inteligencia de Negocios que da una solución completa para el diseño y desarrollo de DW empresariales, DM y aplicaciones de BI. Soluciona el complejo problema de la integración de datos dispersos en varios orígenes y destinos de datos. Este software combina los componentes claves tanto del proceso ETL como de la herramienta de diseño en un solo producto. Además OWB está completamente integrado con la tecnología de base de datos Oracle (que como se mencionó anteriormente, es la tecnología usada en la empresa Salmofood S.A.). Este motivo fue determinante a la hora de escoger cuál sería la forma de realizar los procesos ETL. La versión escogida para trabajar fue OWB 9.2.0.3.0, ya que era la última versión disponible a la hora de comenzar a trabajar en la presente tesis y los manuales disponibles en la página de Oracle se encuentran escritos basados en esta versión. Además, la compañía se encuentra en estado de migración de sus aplicaciones a las últimas versiones en cada producto, entonces es importante para la Gerencia y el Jefe del Departamento de Informática y Tecnología que esto se vaya realizando, no obstante sea posible la comunicación entre distintas versiones.

La arquitectura de este producto se compone de dos componentes: el ambiente de diseño y el ambiente de ejecución. Cada uno de estos componentes maneja un aspecto diferente del sistema. El ambiente de diseño maneja el metadato, mientras que el ambiente de ejecución maneja los datos físicos.

#### C.1.1 EL COMPONENTE DE DISEÑO

El componente de diseño de OWB consiste de un repositorio de metadato que se almacena en una base de datos Oracle y un conjunto de herramientas de diseño y reportes (escritos en JAVA y HTML). Usando estas herramientas, los metadatos pueden ser vistos y manipulados.

La creación de metadato es una actividad de diseño que usa editores en la herramienta Cliente para diseñar objetos, procesos y trabajos. A través del Cliente, OWB soporta el diseño de esquemas de base de datos relacional, multidimensional, procesos ETL y ambientes de desarrollo para el usuario final.

Los sistemas de origen juegan un papel muy importante dentro de cualquier solución ETL. En vez de crear metadato manualmente, OWB dispone de componentes integrados que importan la información importante dentro del repositorio.

Para asegurar la calidad y completitud del metadato en el repositorio, OWB valida toda la información que es introducida en el repositorio. Como ayuda adicional en el desarrollo y la validación del metadato, se encuentra disponible un ambiente de informes basado en Web. Este ambiente de informes permite a los desarrolladores y a los usuarios de la empresa investigar los elementos del sistema que se está construyendo sin usar herramientas de diseño. Una característica muy importante que tiene este ambiente de informes es el Análisis de Impacto, el cual permite analizar el efecto que causaría un cambio en el sistema sin necesidad de realizar la modificación en la realidad.

#### C.1.2 EL COMPONENTE DE EJECUCIÓN

Una vez que se ha diseñado un ambiente ETL a un nivel lógico, se necesitará moverse a un ambiente de base de datos físico. Antes que esto pueda hacerse, se agrega información al diseño lógico acerca del ambiente físico de base de datos cuando el destino se configura para el Despliegue. Después que la configuración está completa, puede ser generado el código.

OWB genera lenguajes específicos de extracción para los procesos ETL y las instrucciones SQL DDL (*Data Definition Language*) para los objetos de la base de datos. El código generado se despliega a un archivo de sistema o dentro de la base de datos.

Ejecutar las funciones ETL significa ejecutar el código desplegado dentro de la base de datos. Esto puede hacerse usando OWB *Deployment Manager* o desde una herramienta externa como *Oracle Enterprise Manager* (OEM). Es entonces cuando el proceso ETL saca los datos desde el origen y los inserta en la base de datos destino. Para permitir informes sobre la carga de los datos, el código generado por OWB contiene rutinas de auditoría. Estas rutinas escriben información acerca del curso de la carga dentro de las tablas de *runtime*. La información capturada mientras se ejecuta el código puede incluir el número de filas seleccionadas, actualizadas e insertadas. Si ocurre algún error mientras se cargan los datos, las rutinas de auditoría

Resumiendo, las funciones que realiza OWB son:

- Importar las definiciones de los datos fuentes.
- Diseñar y crear el esquema de base de datos destino.

- Definir el movimiento de datos y las transformaciones entre los orígenes y los destinos.
- Asignar dependencias entre los procesos ETL.
- Manejar y actualizar las definiciones de las fuentes de datos.
- Desplegar, actualizar y administrar los esquemas destino de datos.
- Diseñar y crear un ambiente de consultas ad-hoc.
- Diseñar y crear un ambiente OLAP

## C.2 ARQUITECTURA Y COMPONENTES DE OWB

La arquitectura de OWB es la que se muestra en la figura C.1.



Figura C.1: Arquitectura de OWB.

- El rectángulo etiquetado con la letra A, representa al *Design Client*, el cual contiene el *Deployment Manager*. El código generado por el *Deployment Manager* va al *Runtime Access User* (etiquetado con el número 1). Para usar el *Design Client* es necesario usar un usuario creado como administrador del repositorio a la hora de instalar los componentes de OWB, específicamente con la herramienta *OWB Repository Assistant*.
- La base de datos (etiquetado con la letra B) contiene el esquema de diseño y está comunicado de dos formas con el *Design Client*. El esquema de diseño

contiene al Repositorio de Diseño, el cual recibe la comunicación del *Design Browser* (etiquetado como C1).

- Existe otra base de datos que contiene 3 usuarios: Runtime Access User (etiquetado con el número 1), Runtime Repository Schema (etiquetado con el número 2) y el Target Schema (etiquetado con el número 3).
- El Runtime Repository Schema requiere casi todos los privilegios de DBA.
  Para proteger a este usuario de abusos, es que se crea el usuario Runtime Access, el cual no tiene privilegios, pero permite a los usuarios usar el Runtime Repository (Repositorio de Runtime que contiene las tablas que sirven para realizar las auditorías). En el contexto de OWB, el Runtime Access User puede desplegar objetos, ejecutarlos y puede hacer lo mismo con los procesos ETL, pero no desde SQL\*Plus, éste usuario no puede ver objetos que no le pertenecen. El Runtime Access User es por lo tanto, una medida de seguridad. Recibe el código generado desde el Deployment Manager y accede a los servicios del Runtime Repository Schema.
- El Runtime Repository Schema es dueño de las tablas de auditoría y los paquetes de auditorías/servicios que son usados por el Target Schema. El Runtime Repository Schema es el usuario central dentro del sistema que maneja las conexiones a varios destinos en el Target Schema.
- El *Runtime Audit Browser Client* (etiquetado como D1) también se comunica con el *Runtime Repository Schema* para permitir los informes de *Runtime*.
- El *Target Schema* contiene los paquetes para ejecutar los procesos ETL que acceden a los paquetes de auditoría/servicios en el *Runtime Repository*

Schema. El Target Schema es el destino actual y contiene los datos del destino y objetos de datos, tales como: cubos, dimensiones, vistas y mappings. Es posible tener múltiples destinos asociados a un *Runtime Repository*. Toda la información pertinenente a este destino, tales como información de ejecución y de despliegue, se almacena en el *Runtime Repository*.

El Runtime Platform Service (etiquetado con el número 4) se ejecuta sobre la base de datos soportando a los 3 usuarios. Finalmente, la configuración del browser (opcional), en la cual el Design Browser y el Runtime Audit Browser se integran con Oracle9iAS se representa independientemente de otros componentes. El Design Browser (etiquetado como C2) y el Runtime Audit Browser (etiquetado como D2) son instalados en Oracle9iAS junto al Middle Tier en la base de datos de la Infraestructura.

## C.3 INSTALACIÓN DE OWB 9.2.0.2.8

Se había mencionado que se iba a usar la versión 9.2.0.3.0 y esto es efectivamente así, pero los archivos de instalación dejan en el equipo la versión 9.2.0.2.8, por lo que es necesario aplicar un *patchset* para dejar el producto en la versión 9.2.0.3.0.

Antes de instalar el OWB es necesario que exista una base de datos (local o remota) para que dentro de ésta se creen todos los esquemas, usuarios y objetos necesarios para que funcione correctamente. La base de datos para el actual caso de estudio está instalada en el servidor remoto dispuesto para esta función. También es necesario tener acceso como SYSDBA a esta base de datos para poder crear todo lo

necesario para realizar una instalación correcta del producto. Como se mencionó anteriormente y se aprecia claramente en la figura C.1, los componentes de OWB pueden ser instalados en dos bases de datos distintas: una para el *Design Repository* y la otra para el *Runtime Repository*. A continuación se detallarán los parámetros que debe tener cada base de datos, en caso de hacer uso de dos de ellas.

Para que la base de datos pueda funcionar con el *Design Repository* y el *Runtime Repository*, la base de datos debe tener ciertos parámetros con algunos valores mínimos. Estos valores aparecen detallados en el manual de instalación de OWB, el cual se encuentra como documento electrónico anexo. Resumiendo lo que aparece en el manual de instalación se deben tener los siguientes parámetros configurados de la siguiente forma:

- GLOBAL\_NAMES = False. Esto es necesario para usar el OWB Runtime Audit Browser.
- REMOTE\_LOGIN\_PASSWORDFILE= *Exclusive*. Esta variable se configura para poder habilitar el modo de conexión de SQL\*Plus:

Connect sys/sys\_password@TNSName as sysdba;

Este parámetro permite asegurar que el usuario SYS tiene el privilegio SYSDBA.

- 07\_DICTIONARY\_ACCESIBILITY = *True*. En caso de que la opción anterior esté configurada como *None* en vez de *Exclusive*.
- OPEN\_CURSORS = 500. Se puede especificar un valor más alto si se desea.
- DB\_BLOCK\_SIZE = 16384. Para el *Design Repository* no se recomienda un valor mayor. Este parámetro es uno de los pocos que no pueden ser

cambiados una vez que ya se ha creado la base de datos, por lo que es muy importante informarse sobre cuál será el uso que tendrá la base de datos al momento de crearla y ver qué parámetros necesita cumplir. Si el servidor no soporta un valor tan alto, se recomienda usar al máximo permitido. Si se tiene menos de 512 MB de RAM se recomienda un valor de 9600. Como ya se mencionó, es importante escoger el valor adecuado para este parámetro, ya que no se puede cambiar después de crear la base de datos. Además, es necesario decir que el valor que se asigna, tiene relación con el tamaño del BLOCK\_SIZE del sistema operativo con que se trabaja. Para este caso particular, y basado en recomendaciones de gente de Oracle se ha dejado en el tamaño por defecto de 8192.

- DB\_CACHE\_SIZE = 314632800 (300 MB). Oracle recomienda dejar este valor tan alto como el sistema lo permita.
- COMPATIBLE = 9.0
- JAVA\_POOL\_SIZE = 20 MB. Este es el mínimo recomendado. Al momento de crear la base de datos, este parámetro viene con un valor predeterminado de 33 MB.
- OPTIMIZER\_MODE = *Choose*.
- QUERY\_REWRITE\_ENABLED = *True*. Esta opción sirve para generar vistas materializadas.
- DB\_FILE\_MULTIPLE\_BLOCK\_READ\_COUNT = 16.

- DBWR\_IO\_SLAVES = n. Este valor es el número de CPU que se encuentra en el servidor. Se recomienda 2\*n por la gente de Oracle y como mínimo el valor de 1.
- LOG\_BUFFER = n. Este valor es de 20\*db\_block\_size como mínimo. La recomendación es de 40 \*db\_block\_size.
- PGA\_AGGREGATE\_TARGET = 314572800. Si se efectúan ordenaciones y agregaciones de forma frecuente se recomienda un valor superior. Sin embargo, DB\_CACHE\_SIZE y PGA\_AGGREGATE\_TARGET deben ser menores que la memoria física total disponible en el equipo.
- ENQUEUE\_RESOURCES = 3000. Un valor superior si es que se importarán archivos MDL (*Metadata Definition Language*) muy grandes. El valor mínimo es de 1 para poder realizar una instalación que no arroje errores.
- PARALLEL\_AUTOMATIC\_TUNING = *True*.
- LARGE\_POOL\_SIZE = 0
- OPEN\_LINK=100
- JOB\_QUEUE\_PROCESSES > 0 (Se recomienda mínimo de 5).

Antes de comenzar la instalación, se debe verificar que las configuraciones regionales y de idioma en el equipo Cliente sean las mismas que se configuraron anteriormente para el Servidor (Basilisco).

Al iniciar el programa de instalación se tienen tres opciones de instalación: *OWB Design Time Install, OWB Server Side Install* y *Custom Install.* Típicamente, las dos primeras opciones son instaladas en máquinas distintas. Si se quiere instalar estas dos opciones en una sola máquina solamente es necesario que tengan un ORACLE\_HOME distinto.

Siguiendo las recomendaciones de Oracle para instalar esta aplicación, el orden para instalar los productos es el siguiente:

- Se instalará OWB Design Time Install.
- Se instalará el Design Repository usando el Repository Assistant.
- Se activará el *Design Browser*.
- Se instalará el *OWB Server Side*.
- Se instalarán los objetos de *runtime* usando el *Runtime Assistant*.
- Se configurará el navegador Web para poder usar el *Design Browser* y el *Runtime Audit Browser*.

## C.3.1 INSTALACIÓN DE OWB DESIGN TIME INSTALL

En este paso se instala el *Design Client* (etiquetado por la letra A en la figura C.1). Esta instalación debe hacer se desde la máquina cliente. Para realizar esto simplemente es necesario iniciar la aplicación *Oracle Universal Installer* que viene incorporado con el software de instalación de OWB. Una vez que se ha iniciado esta aplicación se siguen los siguientes pasos de instalación:

- Indicar la ubicación del origen de los archivos, el directorio destino desde donde se instalará el OWB y el nombre del Oracle\_Home para el mismo.
- En el tipo de instalación, seleccionar OWB Design Time Install.
- Instalar de forma normal y esperar a que finalice la instalación. Si es que el programa no encuentra los requerimientos mínimos de instalación entregará un mensaje de error indicando esta situación.

## C.3.2 INSTALACIÓN DE DESIGN REPOSITORY

En esta etapa se instalará el *Design Repository* (etiquetado con la letra B en la figura C.1). Para conectarse con el *Design Client*, es necesario tener un repositorio de OWB dentro de un esquema de la base de datos. El repositorio de Warehouse Builder almacena definiciones de metadato para todos los objetos creados para llenar el data warehouse. Para crear el *Design Repository* se hace uso de *OWB Repository Assistant*. Esta instalación da el privilegio de *Create Public Database Link* de forma automática al dueño del repositorio. Los pasos para crear el repositorio son los siguientes:

Paso 1: Iniciar OWB Repository Assistant.

**Paso 2**: Introducir el nombre de un usuario con el privilegio de SYSDBA en la base de datos donde se guardará el repositorio. Esto es necesario debido a que el usuario requiere acceder a algunas tablas v\_\$ y esos GRANT los otorga un usuario que tiene el privilegio de SYSDBA cuando se crea el repositorio. Ingresar la contraseña del usuario.

Introducir el resto de los datos para conexión, como son: el hostname (nombre del servidor en que se encuentra la base de datos BDWH), puerto de comunicación por defecto y el nombre del servicio de la base de datos (BDWH).

**Paso 4**: En la siguiente ventana seleccionar la opción *Create a New Warehouse Builder Repository*.

**Paso 5**: Escoger si se desea crear un nuevo usuario para que sea dueño del repositorio o se desea usar uno existente para este fin. Si se selecciona uno existente hay que tomar en consideración que un usuario sólo puede ser dueño de un

repositorio. En este caso particular se escogido la opción de instalar en un nuevo esquema de usuario.

**Paso 6**: Ingresar el nombre del usuario (*adminwh*) y la contraseña. En este paso es que se está creando el esquema para el dueño del *Design Repository*.

Luego de este paso debiera aparecer algún mensaje de advertencia si la base de datos contiene parámetros configurados de tal forma que no cumple los requerimientos mínimos para soportar al *Design Repository*. Si no aparece nada se puede continuar con el siguiente paso.

**Paso 7**: Escoger los *tablespaces* por defecto a utilizar. En este punto se necesita escoger 4 *tablespace*:

- Tablespace de datos. Se recomienda un mínimo de 50 MB
- *Tablespace* de índices: Se recomienda un mínimo de 50 MB
- *Tablespace* temporal: Se puede usar simplemente el tablespace temporal por defecto (TEMP).
- *Tablespace* para *snapshots*: Se puede usar el *tablespace* por defecto (USERS). Este debe ser mayor a 5 MB si no se planea usar *snapshots* o mayor a 60 MB en caso contrario.

**Paso 8**: En la siguiente ventana se debe escoger el lenguaje base MLS. Por limitaciones de la versión de OWB que se trabaja es necesario seleccionar el American English y agregar el English como MLS secundario. Más adelante se verá que el lenguaje English es necesario para que se pueda ejecutar correctamente el OWB y el *Deployment Manager*.

Paso 9: El último paso es finalizar el asistente para que comience la instalación.

#### C.3.3 ACTIVAR EL DESIGN BROWSER

Para que funcionen los Browsers es necesario ejecutar *Start HTTP Server*, el cual está en el servidor que tiene la base de datos. A continuación seguir los siguientes pasos:

**Paso 1**: Abrir el Explorador Web e ir a *http://hostname* si es que se ejecuta este paso desde el Cliente o *http://localhost* si es que se ejecuta desde el mismo servidor.

Paso 2: Seleccionar mod\_plsql.

Paso 3: Seleccionar Gateway Database Access Descriptor Settings.

Paso 4: Elegir la opción Add Default (Blank Configuration).

**Paso 5**: Introducir el *Database Access Descriptor Name (DAD Name)*. En este lugar debe ir el nombre del dueño del *Design Repository* (adminwh).

**Paso 6**: En la sección de *Database Connectivity Information* dejar el nombre de usuario y la contraseña en blanco. Introducir el *String* de conexión de la base de datos, de la forma *hostname:puerto:sid*. Para este caso particular el *string* de conexión es basilisco:1521:bdwh.

Paso 7: Dejar los demás campos en blanco.

**Paso 8**: Hacer click en OK para grabar las opciones y después simplemente hacer click en *Close*.

## C.3.4 INSTALACIÓN DE OWB SERVER SIDE

Esta instalación debe realizarse directamente en el equipo donde se encuentra instalada la base de datos. Cuando se ejecutó *OWB Design Client* se instalaron los componentes de *runtime*, así que si se instala *OWB Server Side* en la misma máquina, se estarían instalando estos componentes dos veces. Si todavía se quiere

instalar en la misma máquina se debe especificar un Oracle\_Home distinto. El *OWB Server Side* agrega otras funciones que no ejecuta el *OWB Design Install*, por lo que es necesaria su instalación. El *OWB Server Side* instala lo siguiente:

- *Runtime Platform Service*, etiquetado como 4 en la figura C.1.
- *Runtime Assistant*, el cual sirve para instalar los usuarios: *Runtime Access User*, *Runtime Repository Schema* y el *Target Schema*.
- SQL *scripts*, para poder realizar ciertas funciones especiales con el *Runtime Platform Service*.

Para realizar la instalación, se debe iniciar el Oracle Universal Installer que se ha usado el punto C.3.1 (en el servidor) y seguir los siguientes pasos:

**Paso 1**: Confirmar la ubicación donde se instalarán los archivos. En este punto se deben ingresar: el origen de los datos, el directorio de instalación y nombre del nuevo *Oracle Home*.

**Paso 2**: Seleccionar como tipo de instalación *OWB Server Side* para instalar los componentes ya mencionados.

**Paso 3**: Las ubicaciones en donde se instalará cada componente se dejará por defecto y simplemente se finalizará el asistente para comenzar la instalación.

## C.3.5 INSTALACIÓN DE LOS OBJETOS DE RUNTIME

Antes de poder desplegar los objetos del warehouse y ejecutar trabajos en el destino de datos es necesario configurar los objetos de *Runtime*. Para esto, se ejecuta el *Runtime Assistant* en el servidor, pero antes se debe tomar una última precaución.

Por un defecto en el producto de OWB, es necesario que la configuración regional y de idioma al momento de ejecutar el *Runtime Assistant* sea Inglés(EEUU),

en caso contrario se obtendrá un error de instalación cuando vaya en 59%. Esta información puede ser corroborada leyendo la nota de Metalink número 243318.1. Si todavía existiera algún error de instalación es posible instalar un parche que se diseño para solucionar este problema. El número del parche es el 3043523 y se puede descargar desde la página de Metalink. Este parche consiste de un solo archivo que debe ser copiado en la ubicación que se especifica en el archivo *Readme* del parche.

En este momento se puede iniciar el *OWB Runtime Assistant* en el servidor y completar la instalación del resto de los objetos siguiendo los siguientes pasos:

**Paso 1**: Introducir la información de la base de datos en la cual se instalarán los objetos y para esto hay que contar con un usuario que tenga el privilegio de SYSDBA. Los datos necesarios son los de un usuario con privilegio de SYSDBA, contraseña del usuario, hostname del servidor, puerto de comunicación por defecto y el nombre del servicio de base de datos.

Paso 2: Seleccionar Runtime Repository, denotado como número 2 en la figura C.1.

**Paso 3**: En la ventana siguiente, seleccionar si se desea crear un nuevo repositorio o eliminar un repositorio existente. Para este caso particular es necesario crear un nuevo repositorio de runtime.

**Paso 4**: Especificar si se desea utilizar un nuevo esquema de usuario o instalar en un esquema de usuario existente. Se recomienda escoger un nuevo esquema de usuario al ser esta la primera instalación que se hace. Para tener en consideración, si de desea instalar en un nuevo esquema de usuario, éste no debe ser dueño de ningún repositorio.

**Paso 5**: Ingresar el nombre de usuario y la contraseña. Esta información se usa para tener acceso al *Runtime Repository*. Basado en el nombre de usuario que se ha ingresado, para este caso fue *runtimewh*, el *Runtime Assistant* creará cuatro roles para controlar el acceso al *Runtime Repository*, los cuales son lo siguientes:

- WB\_D\_\* que tiene privilegios de Desarrollador.
- WB\_R\_\* que tiene privilegios de *Runtime*.
- WB\_A\_\* que tiene privilegios de Administrador.
- WB\_U\_\* que tiene privilegios de Usuario.

En este punto, podría aparecer un mensaje de advertencia si es que la base de datos en la que se quiere instalar el *Runtime Repository* contiene parámetros que no cumplen con las especificaciones mínimas. Si no es así se puede continuar con el siguiente paso.

**Paso 6**: Seleccionar los tablespaces que se usarán para los datos, los índices y el temporal. Se han seleccionado los mismos que se escogieron en la instalación del *OWB Design Client*.

**Paso 7**: Especificar el nombre y contraseña del *Runtime Access User*, denotado por el número 1 en la figura C.1. Como esta es la primera instalación es necesario crear un nuevo usuario. Este usuario es el usado para conectarse al *Runtime Repository Schema*. En este caso se ha utilizado el nombre de usuario de *runtimeac*.

**Paso 8**: En la ventana siguiente se pide ingresar el puerto de comunicación del servidor de Nombres y Direcciones. El valor por defecto es el 4040. Este paso no tiene relevancia en nuestra instalación, ya que no se usará el servidor de nombres que

usa *OWB Name and Address*, porque es otro producto opcional, pero sin embargo, se exige ingresar un número de puerto.

Paso 9: Finalizar la instalación de forma normal.

Cuando la instalación se completa, *OWB Runtime Assistant* da la posibilidad de instalar inmediatamente el *Target Schema*, etiquetado como 3 en la figura C.1. Si se desea se puede realizar la instalación más tarde sin problemas, pero en este caso se realizará de inmediato.

Cuando se instala un *Target Schema*, el *Runtime Assistant* crea de forma automática los privilegios de *Create any directory* y el de *Drop any directory*. Esto permite crear o eliminar conectores, los cuales se usan como directorios dentro de OWB y permiten la conexión con otra base de datos.

Los pasos para instalar el *Target Schema* son los siguientes:

**Paso 1**: Seleccionar si se usará un esquema de usuario existente o uno nuevo. Como esta es la primera instalación se seleccionará instalar en un nuevo esquema de usuario. Debido a que estamos continuando con la primera instalación de los objetos de un nuevo *Repositorio de Runtime*, se toma por defecto los datos ya ingresados respecto a lo siguiente: usuario con privilegio de SYSDBA, nombre del Host, puerto de comunicación con la base de datos (1521), nombre de servicio de Oracle y dueño del repositorio de *Runtime*(*runtimewh*).

**Paso 2**: Ingresar el nombre de usuario (*targetwh*) y la contraseña para el nuevo *Target Schema*.

**Paso 3**: Ingresar los *tablespaces* de datos, índices y temporal que se utilizarán por el *Target Schema*. Los *tablespaces* que se utilizarán serán los mismos que se han elegido en la etapa anterior.

Paso 4: Finalizar la instalación de forma normal.

## C.3.6 CONFIGURACIÓN DEL NAVEGADOR WEB

Para poder utilizar el *Design Browser* y el *Runtime Audit Browser* es necesario configurar el navegador Web. El navegador que se utiliza en este caso particular es el Internet Explorer 6. Los pasos a seguir para configurar el navegador son los siguientes:

**Paso 1**: Agregar *localhost* a la lista de *bypass* del servidor de Proxy. Esta configuración debe hacerse del lado del servidor. Como en la empresa no se está utilizando un proxy, el navegador no usa proxy y no se hace necesario marcar al ticket para *Bypass proxy server for local address*, opción que se encuentra en Opciones de Internet/Conexiones/Configuración de LAN.

**Paso 2**: Configurar el administrador de caché del navegador Web verifique en cada visita por nuevas versiones de las páginas almacenadas. Esta configuración debe hacerse en cada equipo cliente que se usará con el *Design Browser* o con el *Runtime Audit Browser*. Para cambiar esta opción es necesario ir a Herramientas/Opciones de Internet/General/Archivos Temporales de Internet/Configuración/comprobar si hay nuevas versiones de las páginas guardadas y seleccionar *Cada vez que se visita la página*.

También es necesario configurar el archivo *TNSNames.ORA* que se encuentra tanto en el cliente. Este archivo lleva los parámetros que indican dónde y cómo

conectarse con la base de datos. Para configurar esto se usa el *Net Manager* desde donde se pueden ver los nombres de servicios que están configurados bajo la opción *Nomenclatura de Servicios*. Si no se encuentra el nombre de servicio de la base de datos que contiene al repositorio, es necesario agregarla. Para agregar un nuevo nombre de servicio se deben realizar los siguientes pasos:

**Paso 1**: Ir al menú de Edición y Crear un nuevo nombre de servicio de red. Este nombre corresponde al alias de la base de datos que queremos agregar.

**Paso 2**: Ingresar el protocolo de comunicación con la base de datos, que para este caso es TCP/IP.

**Paso 3**: Especificar el nombre del host y el número del puerto que se utilizará para realizar la comunicación TCP/IP.

**Paso 4**: Ingresar un SID (o nombre corto) de la base de datos.

**Paso 5**: Probar la conexión y si es satisfactoria presionar el botón Terminar. En caso contrario, volver atrás y revisar los pasos uno a uno.

Para este caso particular es necesario agregar el alias de la base de datos que contiene la información fuente, para que de esta forma podamos usar el OWB y sacar lo que se necesite de la base de datos original. El TNSNames debería contener además los datos para conectarse a la base de datos donde se dejarán los datos.

## C.4 MIGRACIÓN DE OWB A LA VERSIÓN 9.2.0.3.0

Como se explicó anteriormente, la versión que se usará para generar y administrar los procesos ETL es la 9.2.0.3.0, por que es necesario instalar un *patchset* para poder actualizar el producto. El *patchset* que es necesario descargar de la página de Oracle es el número 3152515.

La guía para poder instalar correctamente este parche se encuentra como documento electrónico adjunto, donde se detalla paso a paso las tareas a ejecutar para poder cambiar la versión del OWB, tanto en el lado del cliente como del servidor.

#### **ANEXO D: ACERCA DE ORACLE DISCOVERER**

#### D.1 ORACLE DISCOVERER 9.0.2.0.0

Oracle Discoverer es la herramienta de Inteligencia de Negocios que será utilizada para que los usuarios puedan realizar sus consultas sobre la información que les interesa. Esta herramienta es muy útil, puesto que los usuarios pueden escoger qué datos analizar y pueden continuar manipulando sus resultados hasta que tengan la información necesaria para tomar sus decisiones de negocios.

Algunas de las características de Discoverer que ayudarán a los usuarios de Salmofood S.A. a realizar análisis para que los ayuden en la toma de decisiones son las siguientes:

- Muestra los datos en un formato de hoja de trabajo similar al que usan los usuarios día a día, por lo que la información es fácil de leer e interpretar
- Puede preparar informes para mostrar el resultado de los análisis
- Tiene la posibilidad de compartir datos con otros usuarios.
- Puede exportar datos directamente a Excel, que es usado por los usuarios de la compañía como hoja de cálculo principal para efectuar sus trabajos. También puede exportar a HTML y otros formatos conocidos.
- Tiene una alta integración con el resto de los productos Oracle que se usan en la empresa
- Ofrece la posibilidad de realizar ordenaciones, filtrar datos según condiciones, se pueden realizar las operaciones de *drill-up* y *drill-down* sobre los datos que se analizan.

Oracle Discoverer está compuesto de cuatro componentes: *Administrator*, *Desktop, Viewer* y *Plus*. La forma en que se relacionan estos cuatro componentes se muestra en la figura D.1.



Figura D.1: Componentes de Oracle Discoverer

En la figura D.1. se aprecia que ningún componente de Discoverer se relaciona directamente a la base de datos. La forma en que tienen acceso a los datos es a través de un EUL (*End User Layer*).

El EUL es la capa de metadato (información acerca de los datos actuales en la base de datos) que resguarda a los usuarios finales de la complejidad y de la estructura física de la base de datos presentando una vista amigable de la información. El EUL permite a los usuarios finales enfocarse en los problemas del negocio en vez de preocuparse en problemas de acceso a la información. La estructura del metadato del EUL protege la integridad de los datos de la base de datos. Lo que el Administrador o cualquier usuario hagan sobre el metadato, afecta solamente a éste y no a la base de datos.

El EUL es una colección de cerca de 50 tablas en la base de datos. Estas son las únicas tablas que pueden ser instaladas por el administrador de Discoverer. El administrador define las áreas de negocios dentro de un EUL.

Un área de negocio es un grupo conceptual de vistas y/o tablas diseñadas para satisfacer los requerimientos específicos de datos de los usuarios finales. Para poder construir un área de negocios es necesario realizar las siguientes tareas:

- Entrevistar a los usuarios para entender sus requerimientos.
- Identificar el origen de los datos y tener un entendimiento claro de su diseño.
- Identificar cuales tablas, vistas o columnas se requieren. Identificar aquellas que podrían estar incluidas en otras áreas de negocios. Por ejemplo, datos de los empleados podrían estar incluidos en un área de Recursos Humanos o en un área de Ventas.
- Identificar los problemas de seguridad y los privilegios de accesos.

Las tablas o vistas que se cargan a un área de negocio se presentan a los usuarios de Discoverer como carpetas. Las columnas dentro de una tabla o vista se presentan como ítems.

A menudo, las columnas y las tablas de una base de datos tienen nombres que para los usuarios finales no tienen significado. Usando el Discoverer *Administrator* se puede hacer que los nombres de las carpetas y los ítems sean más útiles que los nombres de las tablas y columnas de la base de datos en los cuales se basan. Las carpetas en un área de negocios no tienen que basarse directamente en tablas o vistas de la base de datos. También se pueden crear carpetas complejas que contienen ítems basados en columnas de varias tablas o vistas. También puedes crear carpetas personalizadas basadas en instrucciones SQL que se hayan escrito.

De la misma manera, los ítems en un área de negocios no tienen que basarse directamente en columnas. Se pueden crear ítems calculados que ejecutan cálculos sobre varias columnas o que hacen uso de funciones analíticas disponible dentro de la base de datos Oracle.

## D.2 INSTALACIÓN DE ORACLE DISCOVERER 9.0.2.0.0

Antes de instalar el Oracle Discoverer se debe dejar clara la forma en que se utilizará este software. Se utilizará el Discoverer *Administrator* para crear y administrar las áreas de negocios. Para que los usuarios trabajen la información que necesitan y efectúen sus análisis se usará el Discoverer *Desktop*. El Discoverer *Viewer* o el *Plus* no serán utilizados en esta etapa de la implementación de la solución al caso de estudio particular que se está trabajando, ya que para esto es necesario el uso de otra aplicación de Oracle, el Application Server, el cual necesita un servidor más poderoso para que funcione correctamente, pero que está instalación que se explicará a continuación, deberá ser hecha en cada equipo cliente que se necesite.

Para poder instalar el Oracle Discoverer se debe instalar un paquete completo de instalación del Oracle9i Developer Suite. Esta instalación se realizará en el equipo Cliente, ya que desde éste se ejecuta el Discoverer Administrator y se configuran las áreas de negocios. Desde el mismo equipo Cliente se ejecuta el Discoverer *Desktop*.

Como se puede ver, es necesario instalar una suite completa en cada equipo, lo que se puede prestar para pensar que el usuario final podrá ser capaz de acceder directamente a la base de datos. Esto no es cierto, ya que la única base de datos a la que se conectarán los usuarios será a la que se ha llenado usando el OWB y no a la base de datos transaccional de la compañía, ya que sus equipos no tienen el *TNSNames.ora* configurado para que se puedan conectar a otra base de datos. Tampoco podrán crear nuevas áreas de negocios usando el Oracle Discoverer, ya que sus cuentas de usuarios no tendrán los privilegios que se necesita para esto, por lo que solamente podrán hacer un uso del Oracle Discoverer *Desktop* y acceder a las áreas de negocios en las que tienen los permisos suficientes como para trabajar en ellas. Respecto a la utilización de OWB que se instala con esta instalación del Developer Suite, no podrán realizar la creación de nuevos repositorios, ya que para ello tendrían que tener el privilegio de SYSDBA, el cual no tienen.

Antes de dar inicio a la instalación se deben verificar que no existen otros procesos de Oracle ejecutándose y tampoco que se encuentren otras aplicaciones abiertas.

Para comenzar la instalación se debe iniciar el *Oracle Universal Installer* y seguir los siguientes pasos:

Paso 1: Iniciar el Oracle Universal Installer del Oracle Developer Suite.

**Paso 2**: En la localización de archivos dejar el origen con el valor por defecto. En el caso del destino, se debe especificar el nombre de un nuevo *Oracle\_Home* para la

instalación en curso. De acuerdo al nombre del *Oracle\_Home* es que se crea el nombre del directorio donde se instalarán los archivos.

**Paso 3**: En la siguiente ventana se tiene que escoger algún tipo de instalación. Para esta ocasión, existe una opción para instalar la suite de BI, dentro de la cual se encuentra el Oracle Discoverer *Administrator* y el Oracle Discoverer *Desktop*, que son los programas que se utilizarán para dar solución al caso de estudio.

**Paso 4**: En la siguiente ventana se pide ingresar el nombre de un servidor de correo saliente, pero en este caso particular no se utilizará ninguno, por lo que se puede dejar en blanco este campo y continuar con la ejecución del asistente.

Paso 5: Finalizar el asistente y posteriormente salir del Oracle Universal Installer.

Una vez que ha finalizado la instalación de Oracle Discoverer, se recomienda revisar el parámetro NLS\_LANG que ha sido creado en el registro de Windows. Este parámetro debería ser igual al que se encuentra en el servidor. Además es necesario que los archivos *TNSNames.ORA* que están tanto en el *Oracle\_Home* del OWB 9.2.0.3.0 como en el *Oracle\_Home* del Oracle Discoverer 9.0.2.0.0 sean iguales (de esta forma se evitará que ante un cambio del *Oracle\_Home* por defecto no se pueda tener acceso a alguna base de datos).

Ahora que ya todo lo anterior ha sido concretado se puede comenzar a utilizar el Oracle Discoverer *Administrator* para crear las áreas de negocios o el Oracle Discoverer Desktop para llevar a cabo los análisis sobre la información.

## ANEXO E: ABREVIATURAS USADAS

3FN	3ra. Forma Normal.
AP	Accounts Payables.
BI	Business Intelligence. Aplicado también a Inteligencia de
	Negocios.
СОМ	Component Object Model.
CPU	Central Processing Unit.
CRM	Customer Relationship Planning.
CWM	Common Warehouse Data-model.
DAD	Database Access Descriptor.
DBA	Database Administrator.
DDL	Data Definition Language.
DDR	Double Data Rate.
DM	Data Mart.
DOS	Disk Operating System.
DTS	Data Transformation Services.
DW	Data Warehouse.
EAF	Effort Adjustment Factor.
ERP	Enterprise Resource Planning.
ETL	Extract / Transform / Load
EUL	End User Layer.
GB	Gigabyte.
GL	General Ledger.
GHZ	Gigahertz.
--------	--
HR	Human Resources.
HTML	Hypertext Markup Language.
НТТР	Hypertext Transer Protocol.
INV	Inventory.
LAN	Local Area Network.
MB	Megabyte.
MBPS	Megabits por segundo.
MDL	Metadata Definition Language.
MHZ	Megahertz.
MLS	Multi Language Support.
MOF	Meta Object Facility.
MS	Microsoft.
NLS	National Language Support.
OC	Orden de Compra.
OC4J	Oracle Containers for Java.
ODBC	Open Database Connectivity.
OEM	Oracle Enterprise Manager.
OLAP	Online Analitycal Processing.
OLE DB	Object Linking and Embedding Database.
OMG	Object Management Group.
OMS	Oracle Management Server.
OWB	Oracle Warehouse Builder.

PO	Purchasing Order.
RAM	Read Only Memory.
RPM	Revoluciones por minuto.
SID	System ID.
SQL	Structured Query Language.
SYSDBA	System Database Administrator.
TCP/IP	Transfer Control Protocol / Internet Protocol.
UML	Unified Modeling Language.
XMI	XML-based Metadata Interchange.
XML	eXtensive Markup Language.

# ANEXO F: ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

# F.1 INTRODUCCIÓN

# F.1.1 PROPÓSITO

La finalidad del presente documento es especificar claramente lo que se quiere del sistema por parte de los usuarios y lo que finalmente se ha pactado entre la empresa y quien desarrolla, de forma de evitar problemas futuros con posibles cambios de requisitos. Este documento está dirigido al personal del área de compras de la empresa Salmofood S.A. y a quien desarrolla el proyecto.

# F.1.2 ÁMBITO DEL SISTEMA

Con el sistema que se desarrollará se obtendrán los beneficios de tener informes de forma mucho más rápida de lo que actualmente lo hace Oracle Financials. Además se generarán nuevos informes que el ERP de la empresa no tiene prediseñados. Lo que se espera del sistema en el futuro, es que éste sea usado por el personal del área de compras de forma que puedan tener un mejor tiempo de respuesta en sus labores, al tener resultados de forma mucho más rápida.

## F.1.3 VISIÓN GENERAL DEL DOCUMENTO

En la segunda parte de este documento se detallarán las funciones del producto, características de los usuarios y las restricciones sobre el producto si las hubiese.

En la tercera parte de este documento se detallarán de forma específica los requisitos de los usuarios.

# F.2 DESCRIPCIÓN GENERAL

## F.2.1 PERSPECTIVA DEL PRODUCTO

El sistema que se desarrollará tiene directa relación con el ERP que la empresa utiliza actualmente, ya que obtiene todos los datos de este sistema. La forma en que se obtienen los datos no necesitan de ninguna interface, ya que se realiza mediante los procesos ETL que se generarán de forma que los datos obtenidos sean los que se necesitan para generar todos los informes que se requieran por parte de los usuarios.

#### F.2.2 FUNCIONES DEL PRODUCTO

El producto final será de gran utilidad para medir el desempeño de los proveedores, la actividad de compra de cada uno de sus compradores por periodos de tiempo, se podrá realizar un rastreo de las solicitudes de compra y las OC asociadas a cada una de ellas, se verá la actividad de compra de cada centro de costo de la empresa en determinados periodos contables, se tendrá control sobre las recepciones y el estado en que se encuentran.

## F.2.3 CARACTERÍSTICAS DE LOS USUARIOS

Los usuarios que tendrán acceso al sistema, serán en una primera etapa el personal del área de compras, el jefe del área de operaciones y quien desarrolla. Existirá una cuenta de súper usuario que podrá ser utilizada por quien desarrolla y el personal del departamento de Informática y Tecnología. Si bien este sistema obtiene datos del ERP de la compañía, no quiere decir que los usuarios que están creados en este ERP tendrán acceso al nuevo sistema, ya que es totalmente independiente en lo que a accesos se refiere, por lo que tendrán que ser creados nuevos usuarios y otorgarse los permisos y privilegios necesarios para que puedan trabajar libremente con los informes que se generen.

#### **F.2.4 RESTRICCIONES**

La única limitación impuesta sobre quien desarrolla el producto es que no afecte al desempeño del ERP mientras se encuentra trabajando en la creación del nuevo sistema, por lo que todo lo que a carga de datos se refiere deberán ser programadas para horas de la madrugada en las que nadie se encuentra utilizando Oracle Financials.

## F.2.5 SUPOSICIONES Y DEPENDENCIAS

Todo el sistema está dirigido a soportar los requerimientos de Salmofood S.A. Existe la posibilidad de que en el futuro, Cetecsal S.A. - antes un departamento de Salmofood y ahora una empresa diferente - requiera los mismos informes de gestión, para lo cual se deberá crear un área de negocios distinta que haga la diferencia entre las dos organizaciones. Se utilizará el mismo servidor y la misma base de datos. Se deberá tener cuidado en los permisos sobre las dos áreas de negocios, ya que no todos los usuarios pueden ingresar y ver datos de las dos organizaciones.

## F.2.6 REQUISITOS FUTUROS

Una vez que ya todo se encuentre implementado y los informes funcionen sin problemas, será posible agregar nuevo parámetros de entrada a los informes, con el fin de hacerlos más específicos o bien de entregar más detalle. Estos parámetros serán: seleccionar más de un proveedor, seleccionar más de un periodo contable, seleccionar más de un comprador, seleccionar a más de un centro de costo, restringir por periodos contables y restringir por compradores.

#### F.3 **REQUISITOS ESPECÍFICOS**

## F.3.1 INTERFACES EXTERNAS

Los usuarios tendrán acceso al sistema a través del programa Oracle Discoverer, con el cual deben seleccionar el informe que desean generar y a continuación aparece la interfaz con los parámetros a ingresar si es que los hubiera. Ya algunos usuarios conocen el programa que van a utilizar, puesto que ya se han generado algunos informes para otras áreas que utilizan este programa, por lo que no se considerará la interfaz en la que se selecciona al informe como importante para el desarrollo del sistema y sí serán consideradas todas aquellas en las que es necesaria la intervención del usuario para que se genere de forma correcta cada informe.

En una primera etapa, se generarán 15 informes, pudiendo agregarse más de ellos pero solo una vez que los informes que se generen se encuentren funcionando completamente, de forma de no retrasar el desarrollo del proyecto. Todos estos informes presentan una ventana similar en la que se introducen los parámetros de entrada necesario para su generación.

## **F.3.2 FUNCIONES**

Se especificarán las funciones que ofrecerá el sistema ordenada por cada funcionalidad que se desea, es decir, por cada informe que los usuarios desean obtener. Para cada informe se especificará en detalle que es lo que entrega, sus entradas y la salida de cada uno de ellos.

 OC por proveedor en el mes. Este informe entregará un detalle de todas las OC que se han hecho para un determinado proveedor en un periodo contable. Se tienen como entradas para este informe dos parámetros: periodo y proveedor, pudiendo seleccionarse más de un proveedor. Este informe está restringido a solamente un periodo contable. La salida de este informe entrega los siguiente datos: nombre de proveedor, RUT del proveedor, número de OC, tipo de OC, fecha de creación de la OC, divisa, descripción general de la OC, importe total de la OC, importe funcional de la OC y características del proveedor (propiedad femenina, negocio peque o y propiedad minoritaria).

- OC por Centro de Costo al mes. Este informe entregará el detalle de todas las OC asociadas a un Centro de Costo durante un periodo contable. Los parámetros de entrada serán el periodo contable y el centro de costo. Ambos parámetros deben ser únicos. La salida del informe entregará la siguiente información: Solicitud asociada (en caso de existir), fecha de creación de solicitud (en caso de existir la solicitud), número de OC, fecha de creación de la OC, estado de la OC, proveedor, número de la línea de la OC, descripción de la línea de OC, cantidad pedida, precio unitario, unidad de medida, descripción de la cuenta de gastos (mostrar los 5 segmentos por separado: empresa, cuenta, centro de costo, producto e ítem) y finalmente la fecha de recepción (en caso de que ya se haya recepcionado lo solicitado).
- OC abiertas por comprador. Este informe entrega todas las OC que se encuentran en un estado abierto. No tiene restricciones de periodos. Como único parámetro de entrada está el comprador y se puede seleccionar solamente uno de ellos. La salida contiene los siguientes datos: proveedor, número de OC, número de línea de OC, divisa, categoría, artículo, número de revisión de la línea de OC, descripción de la línea de OC, número de envío, fecha prometida (en caso de no existir entregar la fecha de necesidad), precio

unitario del artículo, unidad de medida, cantidad pedida, cantidad recibida, cantidad facturada y estado de la línea de OC.

- OC canceladas por comprador. Este informe entrega el detalle de todas las OC que han sido canceladas por un comprador. No tiene restricciones de periodos, ya que entrega la totalidad de las OC canceladas por el comprador que se está buscando. El único parámetro de entrada que existe es el nombre del comprador, para el cual no se puede seleccionar a más de uno de ellos. La salida del informe incluye lo siguiente: número de OC, nombre del proveedor, sucursal del proveedor, fecha de creación de la OC, importe asociado a la OC, divisa, empleado que la generó, fecha de cancelación de la OC, nombre de la persona que la canceló y el motivo por el cual fue cancelada la OC.
- OC canceladas por proveedor. Este informe entrega el detalle de todas las OC que han sido canceladas, separadas por proveedor. No tiene restricciones de periodos, ya que entrega la totalidad de las OC canceladas y que pertenecen a determinado proveedor. El único parámetro de entrada para el informe es el nombre del proveedor y para el cual no puede seleccionarse más de uno de ellos. La salida del informe incluye los siguientes datos: número de OC, nombre del proveedor, sucursal del proveedor, fecha de creación de la OC, importe asociado a la OC, divisa, empleado que la generó, fecha de cancelación de la OC, nombre de la persona que la canceló y el motivo por el cual fue cancelada la OC.

- Solicitudes versus OC. Este informe entrega el detalle de las solicitudes realizadas y las OC que están asociadas a cada una de ellas. No tiene restricciones de tiempo, ya que entrega el total de las solicitudes hechas por un empleado. El único parámetro de entrada es el nombre del solicitante, para el cual se puede seleccionar a más de uno. La salida del informe incluye los siguientes datos: nombre del empleado que generó la solicitud, número de solicitud, fecha de creación de la solicitud, estado de la solicitud, fecha de cancelación de la línea de la solicitud, fecha de realización de última acción sobre la solicitud, empleado que realizó la última acción sobre la solicitud, número de la línea de la solicitud, descripción de la línea de la solicitud, cantidad solicitada, número de la OC asociada a la línea de la solicitud, fecha de creación de la OC, estado de la OC, fecha de cancelación de la línea de OC, empleado que realizó la última acción sobre la OC, categoría a la que pertenece el artículo solicitado, código del artículo que se está solicitando, número de la línea de OC, descripción de la línea de OC, cantidad pedida en la OC, cantidad recibida, nombre del proveedor, número de la línea de recepción y la fecha en que se realizó la recepción.
- Registro de actividad de compras. Este informe entrega un resumen con todas
  las OC realizadas por un comprador durante un periodo contable. Se tienen
  dos parámetros de entrada, que son el comprador y el periodo contable. De
  ambos parámetros se puede escoger solamente un valor. La salida presenta
  los siguientes datos: fecha de creación de la OC, número de la OC, nombre
  del comprador, tipo de OC, proveedor, divisa e importe de la OC.

- Registro de actividad de solicitud. Este informe entrega un resumen con todas las solicitudes realizadas por un empleado durante un periodo contable. Se tienen dos parámetros de entrada, solicitante y el periodo contable. De ambos parámetros se puede escoger solamente un valor para cada uno de ellos. La salida del informe contiene los siguientes datos: fecha de creación de la solicitud, número de la solicitud, solicitante, tipo de solicitud, descripción de la solicitud y el importe asociado a cada una de las solicitudes.
- Informe de solicitudes canceladas. Este informe entrega todas las solicitudes que han sido canceladas por diversos motivos. No tiene parámetros de entrada y entrega el informe ordenado por número de solicitud. La salida del informe contiene los siguientes datos: número de la solicitud, tipo de la solicitud, persona que la preparó, fecha de creación de la solicitud, persona que la canceló, fecha de cancelación de la solicitud, motivo de cancelación, número de línea de la solicitud, persona solicitante, artículo solicitado, descripción de la línea de la solicitud, fecha de cancelación de la línea, cantidad solicitada, unidad de medida y precio unitario del artículo.
- Análisis de ahorro por comprador. Este informe entrega el detalle de las OC realizadas por un comprador y calcula el ahorro hecho en cada OC. Este cálculo se efectúa en cada una de las líneas que posee la OC, comparando el precio existente en la lista de precios con el precio con que realmente se generó la OC. Se tienen dos parámetros para este informe, el nombre del comprador y el periodo contable. Ambos parámetros deben ser únicos. La salida de este informe contiene los siguientes datos: nombre del comprador,

número de la OC, fecha de creación de la OC, nombre del proveedor, tipo de OC, se encuentra o no en retención, número de la línea de OC, número de envío, tipo de línea de la OC, descripción de la línea de OC, importe negociado, importe real, importe ahorrado, unidad de medida del artículo, precio negociado, precio real y cantidad pedida.

- Análisis de ahorro por categoría. Este informe entrega un listado de las OC que contiene artículos de ciertas categorías y calcula el ahorro en cada línea de la OC. El cálculo se realiza comparando el precio existente en la lista de precios y el que realmente fue ingresado a la OC. Se tiene dos parámetros de entra, periodo contable y la categoría. De estos dos parámetros solamente se puede seleccionar un valor. Para el caso de la categoría, se selecciona solamente la categoría principal y no los sub-grupos que existen dentro de cada una de ellas, pero solamente para efectos de ingreso del parámetro, ya que el informe realiza la distinción entre las categorías y los sub-grupos existentes. La salida de este informe incluye los siguientes datos: categoría, número de OC, fecha de creación de OC, comprador, tipo de la OC, número de línea de OC, número de envío, tipo de línea de OC, descripción del artículo, importe negociado, importe real, importe ahorrado, unidad de medida del artículo, precio negociado, precio real y la cantidad solicitada.
- Envíos vencidos por proveedor. Este informe entrega un listado de todas las OC que tienen líneas que no han sido recibidas, ya sea total o de forma parcial. No tiene restricciones de tiempo, ya que entrega el total de OC por cada proveedor. El único parámetro que tiene es el nombre del proveedor, el

cual debe ser único. La salida del informe contiene los siguientes datos: fecha de vencimiento, nombre del proveedor, comprador, número de la OC, fecha de creación de la OC, estado de la OC, línea de la OC, divisa, unidad de medida, dirección de envío del artículo, organización a la que debe ser enviado el artículo, cantidad recibida, cantidad vencida, nombre del artículo y la descripción de la línea de OC.

Registro de transacciones de recepción. Este informe entrega el registro de transacciones de recepción de las OC durante un periodo contable. El único parámetro para este informe es el periodo contable y para el cual se puede seleccionar solamente un valor. La salida del informe contiene los siguientes datos: tipo de origen, nombre del proveedor, nombre del artículo, categoría, descripción de la línea de recepción, número de recibo de recepción, número de la OC, fecha de la recepción de la línea, tipo de movimiento de recepción (recibir, entregar), unidad de medida del artículo, cantidad del movimiento, precio unitario, tipo de destino, tipo de documento, persona a quien se entregará el artículo, nombre del comprador, persona que realiza la recepción y fecha de la recepción en el sistema.

## F.3.3 REQUISITOS DE RENDIMIENTO

El sistema debe cumplir fielmente los siguientes requisitos de rendimiento:

- Deberá operar sin problemas aún cuando la cantidad de usuarios conectados y realizando consultas sobre él sea de 10 personas.
- El tiempo máximo de espera para una consulta no deberá exceder los 3 minutos.

 Los datos deberán ser actualizados día a día. Esta actualización se realizará durante la noche, antes de que se haga el respaldo diario de la base de datos de la empresa. También debe quedar abierta la posibilidad de efectuar una actualización de los datos fuera de la programación, para ejecutarse en caso extraordinario.

## F.3.4 REQUISITOS DE DISEÑO

No existen restricciones de hardware ni de otros sistemas que afecten el diseño del nuevo sistema.

## F.3.5 ATRIBUTOS DEL SISTEMA

El sistema deberá encontrarse siempre operando, esperando consultas en cualquier momento. Los únicos momentos en que el sistema estará sin funcionar será cuando se realicen los respaldos de la base de datos que contiene la información.

Los usuarios serán creados en la nueva base de datos con sus privilegios y permisos necesarios como para que puedan ver los informes y trabajar con ellos. El acceso a cada informe será restringido a través del Oracle Discoverer, software que ofrece esta característica.

Cada cuenta de usuario se creará usando la primera letra del nombre y a continuación el apellido. Si ya existe una cuenta con estas características, se agregará un número al final del apellido. La contraseña inicial será la misma del nombre de usuario, la cual deberá ser cambiada al momento de iniciar sesión por primera vez. Cada 90 días caducará la contraseña, obligando al usuario a cambiarla.