



---

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

# FACULTAD DE INGENIERÍA

ANÁLISIS DE INFORMACIÓN Y TOMA DE  
DECISIONES PARA ADMINISTRACIÓN DE  
NEGOCIOS

# T E S I S

PARA OBTENER EL TÍTULO DE  
**INGENIERO EN COMPUTACIÓN**

P R E S E N T A

LEONEL SÁNCHEZ LARA

DIRECTOR: ING. DANIEL MARTÍNEZ GUTIÉRREZ



MÉXICO, D.F.

MARZO, 2014

### Agradecimientos

*A Dios*

Gracias por darme sabiduría, fuerza, salud, motivación y fe para culminar esta etapa de mi vida. Te amo por sobre todas las cosas.

*A mis padres.*

Por su apoyo, amor y paciencia incondicional, pero sobretodo por ser el gran motor en todos años de estudio y hacer de mi una persona de bien. Este logro es para ustedes. Nunca terminare de agradecerles, los amo profundamente.

*A mi esposa.*

Por compartir esta gran etapa de mi vida, por sus consejos y total comprensión muestra de su gran amor hacia mí. Gracias te amo.

*A mis hermanas.*

Por brindarme siempre todas esas palabras de amor, fe y esperanza hacia mis metas. Ellos son mi gran impulso y modelo a seguir.

*A mi director de tesis Daniel.*

Por su compromiso, entrega y total guía para obtener este logro, siempre le estaré agradecido por todo el tiempo, las atenciones y su amistad.

*Amigos y compañeros.*

Por acompañarme en este proceso y escuchar de su boca siempre palabras que me llenan de confianza e impulso para culminar esta etapa.

*“Un poco de ciencia aleja de Dios, pero mucha ciencia devuelve a Él”*

**Louis Pasteur**

# ANÁLISIS DE INFORMACIÓN Y TOMA DE DECISIONES PARA ADMINISTRACIÓN DE NEGOCIOS

## ÍNDICE DE CONTENIDO

<b>ABREVIATURAS .....</b>	<b>i</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>1. TEORÍA DEL BUSINESS INTELLIGENCE Y DATA WAREHOUSE.....</b>	<b>4</b>
1.1    Sistemas Data Warehouse/OLAP y OLTP .....	4
1.2    La alineación de los datos .....	7
1.3    Integración de datos .....	7
1.4    Información histórica.....	9
1.5    Acceso y manipulación de datos.....	10
1.6    Patrones de uso .....	11
1.7    Granularidad de los datos.....	11
1.8    Perfil de usuario.....	11
1.9    Ciclo de vida .....	12
1.10   Evolución de arquitecturas.....	13
1.10.1 MOLAP .....	13
1.10.2 ROLAP .....	14
<b>2. EL CICLO DE VIDA DIMENSIONAL DEL NEGOCIO (BDL).....</b>	<b>17</b>
2.1    Planificación de proyectos .....	17
2.2    Definición de los requerimientos del negocio .....	19
2.3    Modelado Dimensional de Negocio (BDM) .....	20
2.3.1    Componentes básicos .....	20
2.3.2    Construcción básica del esquema físico.....	24
2.3.3    Mapeo desde modelo lógico hacia esquema físico .....	25
2.3.4    Construcción de tablas Lookup .....	26
2.3.5    Relaciones de atributos dentro de una dimensión .....	27
2.3.6    Definición de tablas de hechos .....	27
2.3.7    Esquema estrella normalizado I .....	28
2.3.8    Esquema estrella normalizado II.....	29

2.3.9	Esquema estrella normalizado III .....	30
2.4	Diseño físico .....	31
2.5	Diseño y desarrollo del Data Staging .....	33
2.6	Diseño de la arquitectura técnica .....	34
2.7	Selección de productos e instalación .....	34
2.8	Diseño de la aplicación de Business Intelligence .....	35
2.9	Desarrollo de aplicación de Business Intelligence .....	37
2.10	Implementación .....	39
2.11	Mantenimiento y expansión .....	41
2.12	Administración del proyecto .....	42
2.13	Puntos problemáticos de un Data Warehouse .....	42
<b>3.</b>	<b>IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE.....</b>	<b>45</b>
3.1	Planificación del Proyecto .....	45
3.2	Definición de Requerimientos del Negocio .....	45
3.2.1	Requerimiento Funcional .....	45
3.2.2	Requerimiento Técnico .....	45
3.3	Modelado Dimensional de Negocio (BDM) .....	46
3.3.1	Matriz Dimensional .....	46
3.4	Diseño Físico .....	48
3.5	Diseño y Desarrollo de Presentación de Datos .....	55
3.6	Diseño de la Arquitectura Técnica .....	56
3.7	Selección de Productos .....	56
3.8	Especificación de Aplicaciones para Usuarios Finales .....	57
3.9	Desarrollo de Aplicaciones para Usuarios Finales .....	58
<b>4.</b>	<b>PLAN DE PRUEBAS Y RESULTADOS .....</b>	<b>82</b>
4.1	Usabilidad .....	82
	<b>CONCLUSIÓN.....</b>	<b>86</b>
	<b>GLOSARIO DE TÉRMINOS .....</b>	<b>88</b>
	<b>BIBLIOGRAFÍA.....</b>	<b>90</b>

### ABREVIATURAS

Abreviatura	Significado
BI	<i>Business Intelligence</i>
TI	Tecnologías de la Información
OLAP	<i>On-Line Analytical Processing</i>
OLTP	<i>On-Line Transaction Processing</i>
DSS	<i>Decision Support Systems</i>
SDLC	<i>Systems Development Life Cycle</i>
DFM	<i>Dimensional Fact Model</i>
BDL	<i>Business Dimensional Lifecycle</i>
BDM	<i>Business Dimensional Model</i>
RDBMS	<i>Relational Database Management System</i>
FT	<i>Fact Table</i>
GA	<i>General Availability</i>
ETL	<i>Extract, Transform and Load</i>
EIS	<i>Executive Information System</i>
SH	<i>Sales History</i>
DIM	Dimensión
ROLAP	<i>Relational On-Line Analytical Processing</i>
MOLAP	<i>Multidimensional On-Line Analytical Processing</i>
OBIEE	<i>Oracle Business Intelligence Enterprise Edition</i>
KPI	<i>Key Performance Indicators</i>

### INTRODUCCIÓN

La presente tesis tiene como objetivo mejorar la visión del negocio, en específico del área de Ventas, por medio del análisis de información, además de detectar por medio de indicadores la eficiencia del área. Esto es importante para la toma de decisiones en las empresas, ya que hoy día, pueden presentar pérdidas considerables por no interpretar de forma acertada las enormes cantidades de datos que generan, los cuales deberían de ser transformados en información relevante que ayude a mejorar la efectividad de la institución.

Las Tecnologías de Información (TI) forman un papel importante en la vida cotidiana. Se han convertido en parte importante de nuestra vida y por tanto impactan con fuerza a las industrias y grandes empresas; las cuales tienen que recurrir a nuevos métodos tecnológicos para poder satisfacer las necesidades de negocio.

Los sistemas de: Planificación de Recursos Empresariales, los de Administración de Relación con los Clientes y los de Recursos Humanos; son aplicaciones implementadas en las organizaciones que la mayoría de las ocasiones se encuentran desarrolladas en plataformas diferentes. A toda esta problemática, se le suman la producción de documentos impresos, archivos de diversas herramientas ofimáticas, entre otros, convirtiendo a las organizaciones en un mar de información, dificultando encontrar la información relevante para tomar decisiones en el proceso de negocio.

La Inteligencia de Negocios o *Business Intelligence* analiza la información de la organización a fin de generar escenarios, pronósticos y reportes que son suministrados a los ejecutivos y analistas para la toma de decisiones.

Entre las áreas en que el uso de las soluciones de inteligencia de negocios aporta beneficios encontramos: Ventas (análisis de ventas, detección de clientes importantes, análisis de productos, líneas, mercados, pronósticos y proyecciones), *Marketing* (segmentación y análisis de clientes, seguimiento a nuevos productos), Finanzas (análisis de gastos, rotación de cartera, razones financieras) y Manufactura (productividad en líneas, análisis de desperdicios, análisis de calidad, rotación de inventarios y partes críticas).

Los departamentos que se han enlistado son los más comunes, pero esto no delimita que sean los únicos, ya que en donde exista información, se puede implementar una solución de inteligencia de negocios.

Este trabajo está conformado por cinco capítulos. En el primero de ellos se presenta el marco teórico de la inteligencia de negocios. En el segundo capítulo se ilustran las diferentes etapas por las que debe pasar todo proceso de *Data Warehouse*. En el tercer capítulo se muestra un ejemplo aplicando la metodología descrita para la construcción de un sistema de inteligencia de negocios. En el cuarto capítulo se realiza las pruebas de usabilidad y los resultados del sistema de inteligencia de negocio. Finalmente las conclusiones sobre la implementación des sistema.



# **CAPÍTULO 1. TEORÍA DEL BUSINESS INTELLIGENCE Y DATA WAREHOUSE**



### 1. TEORÍA DEL BUSINESS INTELLIGENCE Y DATA WAREHOUSE

#### 1.1 Sistemas Data Warehouse/OLAP y OLTP

Desde su aparición, a mediados de la década de los setentas, las bases de datos (y la teoría sobre bases de datos) no se han detenido. Las primeras versiones de las bases de datos se centraron alrededor de un único repositorio sirviendo a todos los propósitos orientados al procesamiento de la información (desde el transaccional, pasando por el procesamiento *batch*, hasta lo analítico). En la mayoría de los casos, el principal foco de las primeras bases de datos fueron los sistemas operacionales o transaccionales. En las últimas décadas, ha surgido una noción más sofisticada de las bases de datos. Por un lado, el objetivo de servir a las necesidades operacionales, y por otro, cubrir las necesidades analíticas de la información.

El término *Data Warehouse* fue acuñado por Bill Inmon a principios de la década de los noventas y lo definió de la siguiente manera: “Un *Data Warehouse* es una colección de datos *subject-oriented*, *integrated*, *time-variant* y *non-volatile* para ayudar al proceso de toma de decisiones gerenciales”.

- **Subject-Oriented:** datos que brindan información sobre un “sujeto” del negocio en particular, en lugar de concentrarse en la dinámica de las transacciones de la organización.
- **Integrated:** los datos con los que se construye el *Data Warehouse* vienen de diferentes fuentes y son integrados para dar una visión de un “todo” coherente.
- **Time-Variant:** todos los datos en el *Data Warehouse* son asociados con un período de tiempo específico.
- **Non-Volatile:** los datos son estables en el *Data Warehouse*. Más datos son agregados, pero los datos existentes no son removidos.

Esta definición debe tomarse como la definición “pura” sobre *Data Warehouse*. Después de diez años, sin embargo, algunos términos han sido manejados según las necesidades y capacidades del negocio, dando origen a conceptos como el de *Data Mart* (para referirse a *Data Warehouse* sobre áreas específicas en lugar del *Data Warehouse* corporativo) o *Data Warehouse* volátiles, que ante la imposibilidad de almacenar toda la información histórica, almacenan una foto sobre determinado períodos, entre otros.

Ralph Kimball define *Data Warehouse* de una forma más sencilla y práctica pero igual de importante, un *Data Warehouse* es “una copia de los datos transaccionales específicamente estructurados para consultas y análisis”.

En síntesis se puede decir que un *Data Warehouse* es una base de datos orientada al análisis de la información histórica contenida en ella. Dependiendo las necesidades de análisis de la organización puede almacenarse desde unos meses hasta varios años de información. El modelo que soporta la información que contiene se encuentra diseñado, estructurado e implementado con la finalidad y propósito del análisis y navegación de los datos. Se entiende por navegación o *drilling* de los datos, la posibilidad de ver información correspondiente a diferentes contextos o entornos, por ejemplo, analizar las ventas anuales y poder “abrirlos” por sucursal, después analizar en más en detalle una sucursal para ver cómo se discriminan las ventas por cada producto, entre otras características.

Según la implementación seleccionada, los datos son almacenados en forma relacional (RDBMS *Relational Database Management System*, respetando ciertos estándares a nivel de definición y consulta de datos) o en formato multidimensional (las bases de datos multidimensionales son arquitecturas propietarias definidas por cada proveedor que son frecuentemente actualizadas desde base de datos relacionales).

Típicamente los usuarios tienen sólo permisos de lectura sobre el *Data Warehouse*. Comúnmente se dice que los *Data Warehouse* son fuentes secundarias de información pues no generan información por sí mismos, sino que son actualizados desde sistemas fuentes existentes internamente en la organización (sistema de ventas, sistema presupuestario, entre otros) o sistemas externos de información (datos meteorológicos, información de la competencia, cotizaciones de la bolsa, etcétera).

El *Data Warehouse* es el núcleo de la arquitectura de un Sistema de Soporte de Decisiones (DSS, *Decision Support Systems*). Es la parte fundamental del funcionamiento de un DSS, dado que es una única fuente integrada de datos y los mismos, dentro del *Data Warehouse*, son realmente accesibles permitiendo al analista del negocio trabajar en un ambiente más simple que el ambiente clásico transaccional.

El Sistema de Soporte de Decisiones provee los mecanismos de acceso a los datos y el análisis de los mismos. El DSS, el “cerebro” de la arquitectura, es el motor que traducirá los requerimientos de los usuarios en las correspondientes sentencias de consulta para el *Data Warehouse* y el que interpretará los resultados devueltos por el mismo para mostrarlos según lo solicitado por el usuario. Realmente es el componente que concentra la inteligencia del sistema. Diferentes DSS pueden cambiar notablemente la explotación y el comportamiento de un mismo *Data Warehouse*.

El DSS asiste a los tomadores de decisiones dando varios tipos de análisis como: reportes de tendencias, reportes comparación y análisis *ad-hoc*. El Sistema de Soporte de Decisiones basado en técnicas de *Data Warehouse* transforma los datos que tiene una organización en información, con el objetivo de tomar mejores decisiones de negocio basados en un análisis dimensional.

Los sistemas OLAP, *On-Line Analytical Processing*, son diseñados para cumplir con un conjunto de metas; las cuales son diferentes de los objetivos de un sistema transaccional (OLTP, *On-Line Transaction Processing*). Por ejemplo, una meta de los OLTP es maximizar la concurrencia mediante el uso de bloqueos. Dicho objetivo no es pertinente en el diseño de OLAP, donde las consultas son sólo del tipo selección (*select*). Además de las técnicas de diseño, un desarrollador de OLAP debe focalizarse en entregar un análisis multidimensional y capacidades de reportes *ad-hoc* (generación de reportes por parte del usuario experto basados en el conocimiento del negocio). Para realizar esto, el diseñador necesita conocer los requerimientos del negocio tan bien como las técnicas de diseño multidimensional. Los OLTP y los OLAP difieren ambos en diseño y funcionalidad. Entre los puntos de diferencias se encuentran: los objetivos principales de construcción, la orientación o alineación de los datos, la integración, los datos históricos, el acceso de datos y manipulación, los patrones de uso, la granularidad de los datos, el perfil de los usuarios y el ciclo de vida.

### 1.2 La alineación de los datos

Los OLTP están alineados por aplicación. En cambio, los sistemas OLAP están alineados por dimensión, además todos los tipos de datos integrados en un solo sistema. En los OLAP los datos son organizados definiendo dimensiones del negocio (áreas temáticas o sujetos) y se focaliza en el cumplimiento de requerimientos del análisis del negocio.

Los distintos mercados verticales tienen diferentes perspectivas desde los ambientes OLTP y desde los ambientes OLAP, por ejemplo, en el mercado bancario existen numerosas aplicaciones de cuentas y préstamos a nivel operacional mientras que en un OLAP la información estaría organizada por cliente, tipo de cuenta y tiempo. En los *retails* (empresas especializadas en la comercialización masiva de productos o servicios uniformes a grandes cantidades de clientes) hay aplicaciones sobre registro de ventas, manejo de inventario y presupuesto y en el ambiente de OLAP se hablaría de productos, sucursales, tiempo y las diferentes variables propias de negocio (unidades vendidas, monto neto vendido, monto bruto vendido, precio promedio, unidades presupuestadas, cantidad de piezas, entre otros).

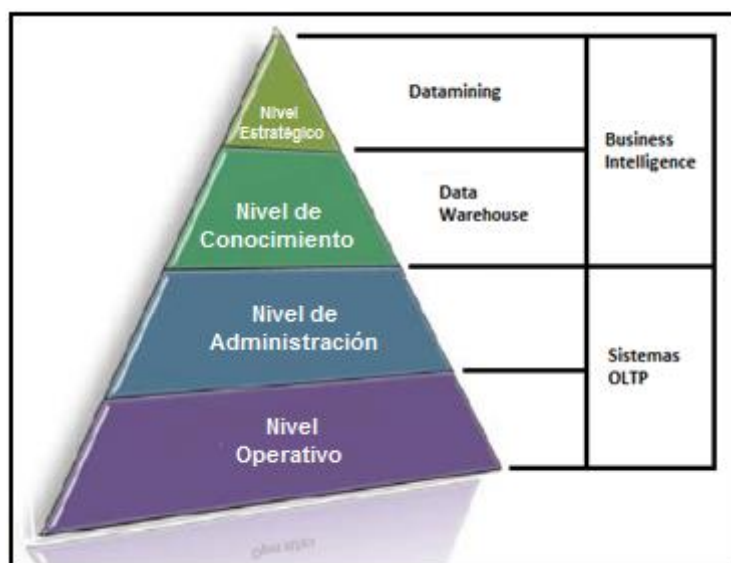
### 1.3 Integración de datos

En los OLTP, los datos se encuentran típicamente no integrados, son calificados como datos primitivos. Los mismos son estructurados independientemente uno de otros, pudiendo tener diferentes estructuras de claves y convenciones de nombres. Son usualmente almacenados en diferentes formatos de archivos, por ejemplo relacionales, archivos planos, entre otros. Incluso, si todos los datos están en formato relacional, los mismos pueden residir en diferentes plataformas de hardware y en distintas RDBMS.

Sin embargo, en los ambientes OLAP, los datos deben estar integrados. Son conocidos como datos derivados o datos DSS, *Decision Support Systems*, dado que provienen de sistemas transaccionales o sistemas de archivos maestros preexistentes en las mismas organizaciones o de sistemas externos de información. El *Data Warehouse*, con el objetivo de alinear los datos por áreas temáticas, debe integrar datos operacionales estandarizando estructuras y convenciones de nombres (concepto de diccionario de datos). Este concepto da paso a otro importante tema de interés en la actualidad, *Data*

*Cleansing*, que se refiere al reformato de datos para conformar los estándares del *Data Warehouse*. La limpieza debe ser realizada por los programas de extracción para asegurar que los datos estén en un formato consistente antes de ser cargados dentro del *Data Warehouse*.

Dentro de las organizaciones se pueden reconocer distintos niveles de uso de datos como lo muestra la Figura 1 y éstos se describen a continuación.



**Figura 1.** Niveles de uso de Datos

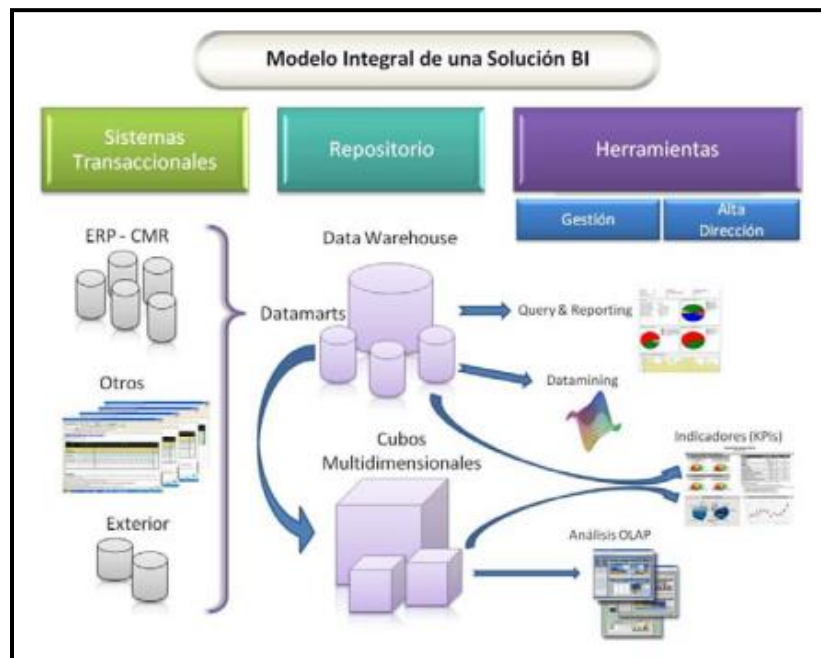
**Nivel operativo:** Se utilizan sistemas de información que monitorean las actividades y transacciones elementales de la organización.

**Nivel de administración:** Se realizan tareas de administradores de nivel intermedio apoyando las siguientes actividades: de análisis, de seguimiento, de control y toma de decisiones, realizar consultas sobre información almacenada en el sistema, realizar informes y facilitar la gestión de la información por parte de los niveles intermedios.

**Nivel de conocimientos:** En este nivel encontramos a los analistas de datos, que cubren el núcleo de operaciones tradicionales de captura masiva de datos y servicios básicos de tratamiento de datos, con tareas predefinidas.

**Nivel estratégico:** Tiene como objetivo realizar las actividades de planificación a largo plazo, tanto del nivel de administración como de los objetivos que la empresa posee.

Una herramienta de BI debe de ser capaz de reunir información dispersa en toda la empresa e incluso en diferentes fuentes para así proporcionar a los departamentos la accesibilidad, poder y flexibilidad que necesitan para analizar la información como lo muestra la Figura 2.



**Figura 2.** Modelo Integral de una Solución de BI

### 1.4 Información histórica

Los OLTP usualmente retienen datos en periodos de 60 a 90 días, después son resguardados por los administradores de base de datos en almacenamientos secundarios fuera de línea (por ejemplo cintas o en disco a nivel de *backup*). También es común que contengan sólo valores corrientes, por ejemplo, el actual balance de cuentas para clientes y no valores históricos. Puede no incluir el tiempo como un componente de la clave, sólo el balance corriente de cuentas es almacenado, por lo tanto, no tiene sentido guardar el tiempo como parte de la clave de los datos.

Los OLAP, a diferencia de los OLTP, almacenan tanta historia como sea necesario para el análisis del negocio, típicamente dos a cinco años de datos históricos. Retienen valores para cada período (el atributo más atómico de la dimensión tiempo) en la base de datos. Almacenan una serie de “fotos” instantáneas de los datos operacionales y la frecuencia con la cual se define el nivel de detalle se indica en la correspondiente hoja de la dimensión tiempo. Toda esta cantidad y tipo de historia ayuda a generar reportes de comparación de tendencias y períodos de tiempo.

Por otro lado, las bases de datos orientadas al análisis siempre contienen el tiempo como clave dado que una de las principales razones para la construcción del *Warehouse* es el almacenamiento de datos históricos y el análisis a lo largo del tiempo.

### 1.5 Acceso y manipulación de datos

Los OLTP realizan una manipulación de datos registro por registro con grandes cantidades de inserciones (*inserts*), actualizaciones (*updates*) y borrados (*deletes*). Además necesitan de rutinas de validación y transacciones a nivel registro. Generalmente poseen pequeñas cantidades de datos involucrados en un solo proceso o transacción y la puesta a punto de la base de datos para el procesamiento de transacciones, se focaliza en mecanismos de bloqueo de objetos y asignación de recursos (*tuning* específico).

En cambio, los OLAP tienen una carga y acceso masivo de datos, no se realizan *inserts*, *updates* o *deletes*. La validación de datos se realiza antes o después de la carga, (nunca a nivel registro o transacción). Principalmente se realizan sentencias de *select* sobre varios registros y tablas, teniendo grandes volúmenes de datos involucrados en un único proceso o análisis. Es por ello que generalmente no se respetan las formas normales tan necesarias en los sistemas operacionales clásicos. Las anomalías que tienden a subsanar estas reglas de normalización no se presentan en los sistemas OLAP donde la carga de la información está automatizada y puede permitirse el manejo de redundancia controlada como punto para la mejora de los tiempos de respuesta de las consultas a la base de datos.

### 1.6 Patrones de uso

Los OLTP normalmente mantienen un patrón de uso constante requiriendo grandes cantidades de recursos y consumiendo sólo el tiempo referido a la transacción. En contraposición, los OLAP tienen un patrón de uso liviano con picos de usos eventuales en el tiempo (afectados por la disponibilidad de los datos y el flujo de trabajo del negocio). Los picos de uso suceden el mismo día de cada semana y el mismo día de cada mes (cuando los datos están por primera vez disponibles o cuando el negocio necesita por primera vez un reporte).

### 1.7 Granularidad de los datos

En los OLTP se encuentran los datos a nivel detallado o nivel transaccional. Una transacción incluye a nivel atómico cada uno de los componentes de su estructura (fecha, hora, código de cliente, código de movimiento, importe, entre otros).

En un comienzo, los OLAP contenían información sumada hasta cierto nivel (el permitido por los volúmenes de consolidación de los cubos multidimensionales de la década de los ochenta). Con el avance en el desarrollo de las bases de datos durante la década de los noventa se pudo incorporar la posibilidad de llegar incluso a nivel de detalle (nivel de transacción).

La diferencia en la granularidad de los datos viene dada por el uso de los mismos. Si bien un *Data Warehouse* puede tener información a nivel transaccional, el objetivo de esta granularidad mínima está asociado con el deseo de realizar ciertos tipos de análisis que requieren que la información esté a ese nivel de detalle (análisis de *market basket*), pero no significa que se vea la información a nivel transaccional.

### 1.8 Perfil de usuario

Dado que los OLTP tienen como objetivo asistir a aplicaciones específicas y asegurar la integridad de los datos, el perfil de usuario que interactúa con dichos sistemas se



encuadra dentro de los empleados operacionales de una organización (comunidad operativa).

Por el contrario, dado el objetivo estratégico y el nivel de información que manejan los OLAP, el perfil de usuario sobre este tipo de sistemas corresponde a la comunidad gerencial, la cual está a cargo de la toma de decisiones.

### 1.9 Ciclo de vida

Los OLTP pueden ser desarrollados por el clásico SDLC, *Systems Development Life Cycle* (Ciclo de Vida del Desarrollo de Sistemas). Según señala Inmon, los OLTP operan bajo un ciclo de vida bastante diferente, a veces denominado CLDS (el inverso de SDLC). El clásico SDLC es guiado por los requerimientos. En una etapa posterior se comienza con el diseño y luego con el desarrollo. El CLDS es casi la inversa, comienza con los datos. Una vez que se identifica a los datos, los mismos son integrados y luego probados.

Más tarde son desarrolladas las aplicaciones de explotación y finalmente son atendidos los requerimientos de consulta de los usuarios. Dado el particular flujo del ciclo de vida, se le suele llamar *data-driven* (guiado por los datos) en contraposición al tradicional guiado por los requerimientos *requirement-driven* (guiados por los requerimientos) del SDLC. Ejemplos de esta metodología de trabajo puede también encontrarse en donde se propone un enfoque metodológico al estilo de Inmon denominado *Dimensional Fact Model* (DFM), que comienza con el análisis de los sistemas de información y luego se cubre los requerimientos de los usuarios en base a la información disponible en los sistemas fuentes.

Es de importancia a lo que se conoce como *query profile* (perfil de consulta) de los usuarios, es decir, el análisis que hará y cuáles son los requerimientos para hacer un análisis exitoso que ayude a cumplir con los objetivos de negocio marcados por las organizaciones. Si bien los datos disponibles tienen un valor fundamental en el modelado y diseño del *Data Warehouse*, en esta nueva visión se utiliza más para contrastar con los requerimientos especificados con los usuarios más que como fuente única de modelado. Los requerimientos del negocio son el “centro” del *Data Warehouse*.

### 1.10 Evolución de arquitecturas

#### 1.10.1 MOLAP

Las arquitecturas de los *Data Warehouse* han evolucionado con base en el crecimiento, funcionalidad y rendimiento de las bases de datos. Es por ello, que la arquitectura predominante de los ochentas fue el modelo MOLAP (*Multidimensional OLAP*). Estos *Data Warehouse* funcionaban con bases de datos propietarias (comúnmente denominados “cubos multidimensionales”) dado que muy pocas bases de datos soportaban el tipo de trabajo al que eran sometidas en ambientes de *Data Warehouse*.

##### a) Capacidad de análisis

- Ofrece vistas de objetos multidimensionales
- Tiempo de respuesta cero, pues tiene todo pre-calculado
- Si no se pre-calcula todo (en general todo el pre-cálculo tiene volúmenes inaceptables) la capacidad de análisis se limita a aquellas porciones del cubo que fueron pre-calculadas.

Existen variantes de los MOLAP donde en caso de no poder responder a una consulta, se pre-calcula el cubo que responde a esa pregunta y a la periferia, pudiendo llevar esta generación varias horas de construcción.

##### b) Sistema de diseño propietario

- Generalmente el cubo se trata de una “caja negra” de datos encriptados que pueden residir de forma local o en un servidor MOLAP
- El cliente interactúa contra la MDDb vía un lenguaje de acceso propietario.
- MDDb se alimenta desde una RDBMS
- Flexibilidad y escalabilidad limitados
- Cambios en el modelo dimensional del negocio implican la generación de todos los cubos nuevamente
- Nuevos datos disponibles implican una ventana *batch* de carga bastante extensa para la generación de todos los cubos multidimensionales

### c) Unidad de almacenamiento

- Multidimensional data cube, unidad de almacenamiento contra la cual se disparan las consultas
- Pre-agregación de todas las posibles intersecciones entre las dimensiones
- Rápida respuesta de consultas obviando la agregación en tiempo de ejecución
- Ineficiencia a medida que crecen volúmenes de datos y/o dimensiones
- Crecimiento exponencial ante cualquier adición al cubo multidimensional
- Reducido potencial de almacenamiento de datos
- Reducido número potencial de dimensiones

### d) Ambientes adecuados

- Modelos dimensionales pequeños y estáticos
- Instalaciones dónde el tiempo de respuesta sea crítico
- Pocos volúmenes de datos
- Análisis de información a nivel agregado

## 1.10.2 ROLAP

A comienzos de los noventa, comienza a hacerse posible la utilización, en los ambientes de *Data Warehouse*, de las bases de datos relacionales en forma directa y no como fuente secundaria para el armado de cubos multidimensionales. Es así como surge una nueva arquitectura de *Data Warehouse* denominada ROLAP (*Relational OLAP*).

### a) Capacidad de análisis

- Ofrece vistas de objetos multidimensionales
- Tiempos de respuestas aproximadamente en segundos y minutos. Existen técnicas de *tuning*, *cache*, materialización de vistas, indexación y esquema de diseño que mejoran la performance de respuesta de los ROLAP
- La capacidad de análisis abarca todo lo que se encuentra en el *Data Warehouse* o *Data Mart*

### **b) Sistema de diseño abierto**

- Permite estrategias con distintos motores de base de datos
- El cliente interactúa directamente contra el RDBMS vía SQL
- Provee flexibilidad y escalabilidad
- Los cambios en el modelo dimensional del negocio son trasladados al *Data Warehouse* e inmediatamente se encuentra disponible para las consultas pertinentes
- La ventana de carga del *Data Warehouse* es menor pues no existe el tiempo de generación de los multi-cubos

### **c) Unidad de almacenamiento**

- *Data Warehouse*, unidad de almacenamiento contra la cual son procesadas las consultas
- Permite almacenar la información en formato relacional
- Iguala la performance de los cubos MOLAP con un apropiado esquema de diseño
- Explota al máximo las capacidades de paralelismo de los RDBMS
- No decae en eficiencia a medida que crecen los volúmenes de datos o números de dimensiones
- No tiene límite inherente en cuanto a capacidad de almacenamiento. El límite está dado por el RDBMS. Hay motores que están más preparados que otros para actuar en ambientes de *Data Warehouse*, aunque en la actualidad casi todos los motores ya incorporaron funcionalidad asociada a los *Data Warehouse*
- No tiene límite inherente en cuanto a cantidad potencial de dimensiones

### **d) Ambientes adecuados**

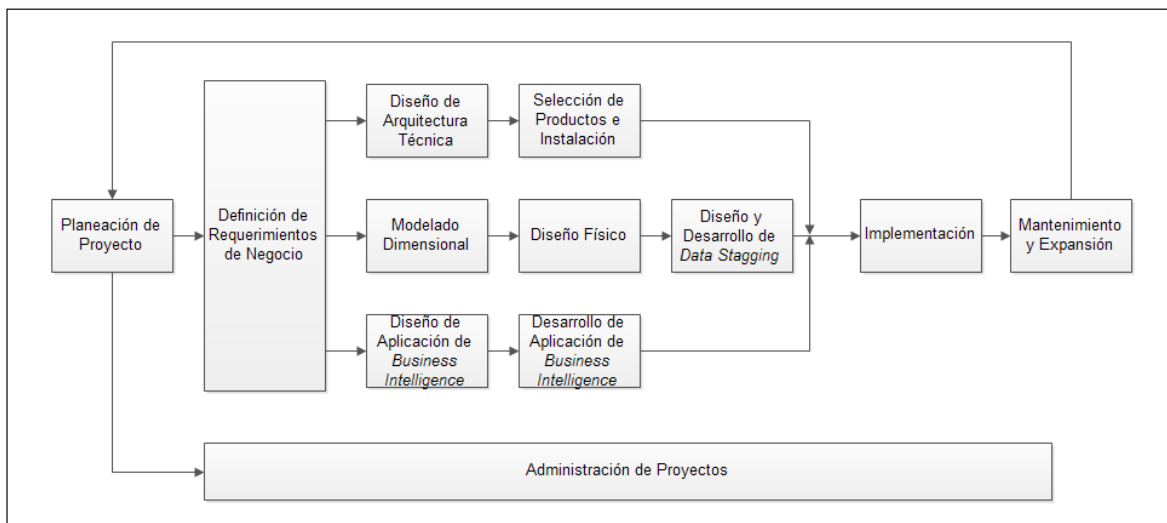
- Modelos dimensionales grandes y dinámicos
- Grandes volúmenes de datos
- Necesidad de análisis a nivel transaccional



# **CAPÍTULO 2. EL CICLO DE VIDA DIMENSIONAL DEL NEGOCIO (BDL)**

## 2. EL CICLO DE VIDA DIMENSIONAL DEL NEGOCIO (BDL)

Ralph Kimball presenta con el nombre de *Business Dimensional Lifecycle* (BDL) las diferentes etapas por las que debe pasar todo proceso de *Data Warehouse*. Este enfoque de implementación de *Data Warehouse* es ilustrado en la Figura 3. El diagrama muestra la secuencia de tareas de alto nivel requeridas para el efectivo diseño, desarrollo e implementación de *Data Warehouse*. Es una vista general del mapa de ruta de un proyecto, en el cual cada bloque indica en dónde se está ubicado, por dónde se pasará y hacia dónde dirigirse.



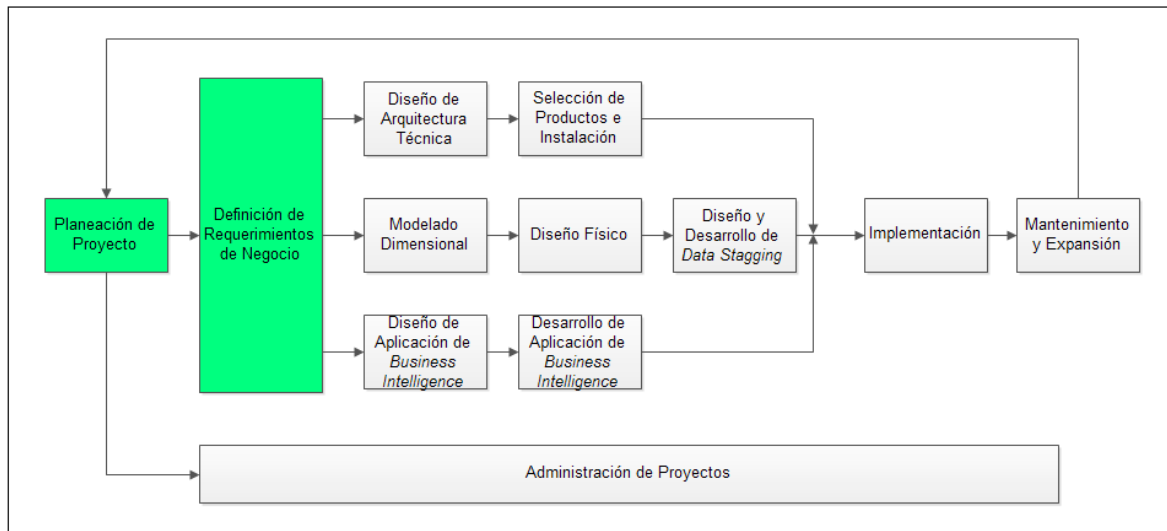
**Figura 3.** *Business Dimensional Lifecycle* propuesto por Ralph Kimball

Es importante aclarar, que el BDL no intenta reflejar un proyecto en término de tiempos y plazos. Como se puede notar cada rectángulo del diagrama tiene el mismo ancho, con la excepción del la Administración del Proyecto. La magnitud de recursos y el tiempo requerido para cada etapa del ciclo de vida no es igual. El BDL se focaliza en secuencia y concurrencia no en tiempos y plazos.

### 2.1 Planificación de proyectos

La planificación busca identificar la definición y el alcance del proyecto de *Data Warehouse*, incluyendo justificaciones del negocio y evaluaciones de factibilidad. Se enfoca sobre los recursos, perfiles, tareas, duraciones y secuencia.

El plan de proyecto resultante identifica todas las tareas asociadas con el BDL e identifica las partes involucradas.



**Figura 4.** Interacción entre Planeación y Requerimientos

La planeación del proyecto es dependiente de los requerimientos del negocio, como es denotado en el diagrama del BDL en la Figura 4, ya que los requerimientos del negocio determinan el alcance del proyecto, definen los recursos necesarios, entre otros y la planificación acotará los requerimientos ya sea por cuestiones de recursos y/o tiempo.

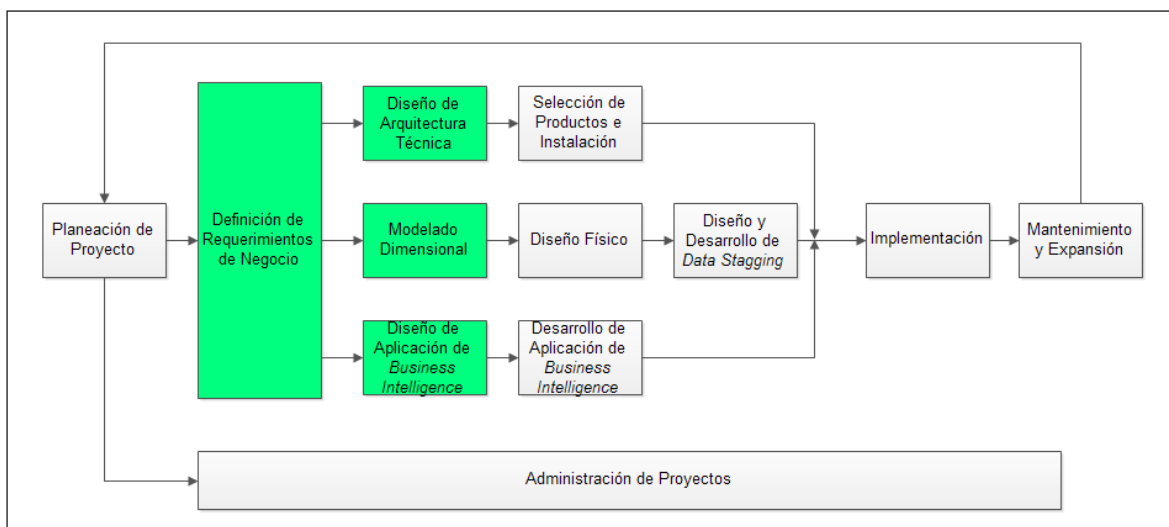
Esta etapa se concentra sobre la definición del proyecto (identificación del escenario del proyecto para saber de dónde surge la necesidad del *Data Warehouse*). Factores asociados con estas etapas incluyen: identificación de los usuarios finales, motivaciones del negocio, cooperación entre áreas de sistemas y negocios, cultura analítica de la organización y análisis de factibilidad (tanto tecnológica como de disponibilidad de datos).

Para medir estos factores se propone una prueba de buena disposición del proyecto dónde describe diferentes escenarios posibles. Adicionalmente propone técnicas (especificaciones de alto nivel, priorización de requerimientos y pruebas de concepto) para mitigar las deficiencias que el proyecto pudiera tener en algunos de los factores mencionados con anterioridad. A nivel de planificación del proyecto, establece la identidad del mismo, el personal (*staff*): los usuarios finales, líderes, gerentes del proyecto (tanto de sistemas como del sector usuarios), equipo del proyecto (analistas, arquitectos, DBA,

diseñadores, responsables de extracción, desarrolladores, instructores, entre otros), el desarrollo del plan del proyecto, el seguimiento y monitoreo.

## 2.2 Definición de los requerimientos del negocio

Un factor determinante en el éxito de un proceso de *Data Warehouse* es la interpretación correcta de los diferentes niveles de requerimientos expresados por los diferentes niveles de usuarios. La técnica utilizada para especificar los requerimientos de los analistas del negocio difiere de los enfoques tradicionales guiados por los datos. Los diseñadores de los *Data Warehouse* deben entender los factores claves que guían al negocio para determinar efectivamente los requerimientos y traducirlos en consideraciones de diseño apropiadas.



**Figura 5.** Iteración entre Requerimientos del Negocio y etapas subsiguientes

La definición de los requerimientos del negocio establece la base para las tres etapas paralelas subsiguientes mostradas en la Figura 5, enfocadas en la tecnología, los datos y las aplicaciones por lo cual es altamente crítico y es el centro de atención del BDL.

Los usuarios finales y sus requerimientos impactan siempre en las implementaciones realizadas de un *Data Warehouse*. Los requerimientos del negocio deben determinar el alcance del *Data Warehouse* (qué datos debe contener, cómo debe estar organizado, cada cuánto debe actualizarse, quiénes y desde dónde accederán, entre otros). Estas



tácticas y estrategias se focalizan sobre las entrevistas (diferentes tipos, preparación de la entrevista, roles a cubrir, búsqueda de información pre-entrevista, selección de entrevistados, desarrollo de los cuestionarios, planificación, preparación de los entrevistados, conducción de la entrevista, contenido, cierre, revisión de resultados).

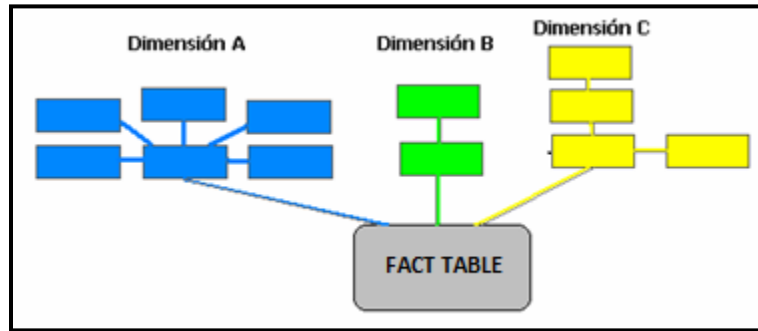
### 2.3 Modelado Dimensional de Negocio (BDM)

El modelo dimensional utilizado en el *Data Warehouse*, organiza y presenta los datos definiendo dimensiones (líneas o áreas temáticas del negocio). Por ejemplo, en los modelos de empresas es muy común encontrar las mediciones estructuradas por las dimensiones sucursal, producto y tiempo. De esta forma, permiten analizar la información a distintos niveles de agregación dentro de las diferentes dimensiones. Dentro de cada dimensión se puede definir los niveles de agregación o sumas para cada análisis, a estos niveles de granularidad se les conoce con el nombre de atributos.

#### 2.3.1 Componentes básicos

Al momento de realizar un modelo dimensional del negocio, BDM, es necesario contemplar los siguientes componentes ilustrados en la Figura 6:

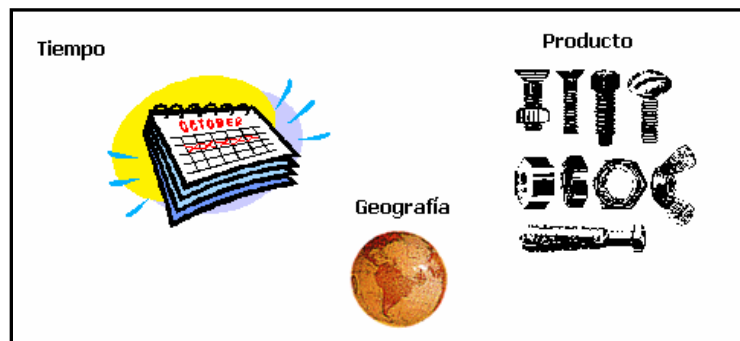
- Dimensiones (*Dimensions*)
- Atributos (*Attributes*)
- Elementos (*Attributes Elements*)
- Relaciones (*Attribute Relationships*)
- Jerarquías (*Hierarchies*)
- Métricas (*Facts o Metrics*)



**Figura 6.** Esquema de un modelo dimensional

### 2.3.1.1 Dimensiones

Las dimensiones son el sujeto del negocio, las mismas proveen un método general para organizar la información corporativa como los ejemplos mostrados en la Figura 7. Definidas como un grupo de uno o más atributos, separados y distintos uno de otros (es decir, que no se comparten atributos).

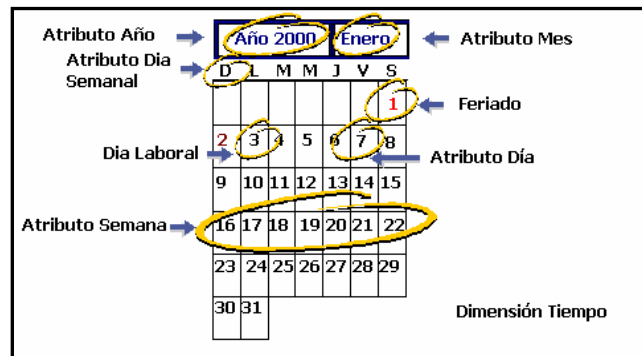


**Figura 7.** Dimensiones

### 2.3.1.2 Atributos

Los atributos son una agrupación de elementos dentro de una dimensión. Representan categorías o clases de elementos que tienen el mismo nivel lógico dentro de una dimensión donde todos los elementos de un atributo se relacionan con otros atributos de la dimensión de la misma forma. La finalidad de los atributos es ver la información de cada dimensión a diferentes niveles de detalle y agrupar los datos para ser analizados. Permiten definir niveles de agregación y presentar datos agrupados por uno o más

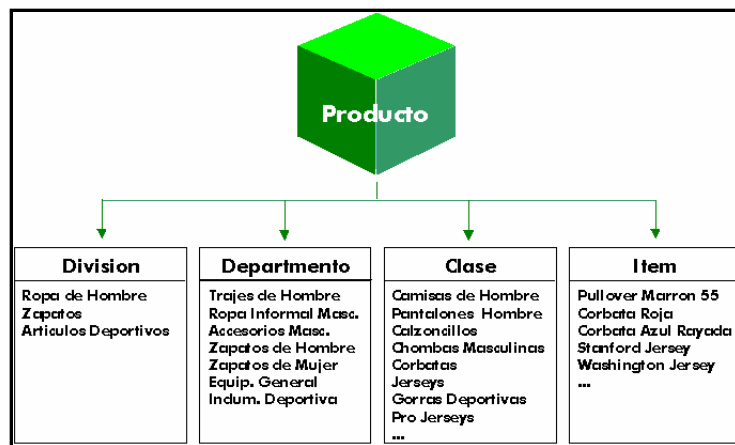
atributos como se muestra en la Figura 8. Los atributos son niveles dentro de una dimensión o calificadores de una dimensión.



**Figura 8.** Atributos

### 2.3.1.3 Elementos

Son las instancias o valores de los atributos que, como componentes “atómicos” del modelo, permiten clasificar el rendimiento del negocio. Un ejemplo se muestra en la Figura 9. Es importante aclarar que si bien no forman parte del BDM es aconsejable su incorporación para un mayor entendimiento del modelo en etapas tempranas del ciclo de vida.



**Figura 9.** Elementos o instancias de atributos

### 2.3.1.4 Relaciones

Son asociaciones lógicas de atributos dentro de una jerarquía definida por las instancias de los atributos y transitivas dentro de una jerarquía. Existen diferentes tipos de relaciones clasificadas por su “cardinalidad”.

- Uno-a-uno (1:1)
- Uno-a-muchos (1:M)
- Muchos-a-uno (M:1)
- Muchos-a-muchos (M:N)

Es importante aclarar que los atributos dentro de una dimensión están directamente relacionados uno con otros a través de los diferentes tipos de relaciones antes definidos. En cambio, los atributos en diferentes dimensiones están relacionados uno con otros a través de las métricas del negocio definidas como intersección de las dimensiones.

### 2.3.1.5 Jerarquías

Representadas por un ordenamiento lógico dentro de la dimensión, se encuentran formadas por los diferentes tipos de relaciones entre los atributos de una misma dimensión. Pueden existir múltiples jerarquías dentro de una dimensión pero siempre es posible identificar una jerarquía principal o columna vertebral de la dimensión y jerarquías secundarias o descriptivas compuestas por atributos característicos definidos desde la jerarquía principal.

Dentro del contexto de navegación del modelo dimensional, se puede decir que las diferentes jerarquías definen el mapa de caminos para el “*drilling*” o la “navegación” de los datos. Se hace *drill-up* o *roll-up* cuando se mueva hacia un atributo superior dentro de la jerarquía (navegación ascendente), *drill-down* o *roll-down* cuando se analice información a mayor nivel de detalle (navegación descendente), *drill-within* cuando se mueva entre la jerarquía principal y la característica desde un atributo hacia cualquier otro que no sea ni descendiente ni ancestro dentro de la misma dimensión (navegación intra-dimensional) y *drill-across* para analizar información sobre diferentes dimensiones (navegación inter-dimensional).

Como convenciones del modelado, la jerarquía principal se dibuja verticalmente desde el atributo más agregado (arriba) hasta el más atómico (abajo) y las jerarquías características se adicionan por los costados como se muestra en la Figura 10.

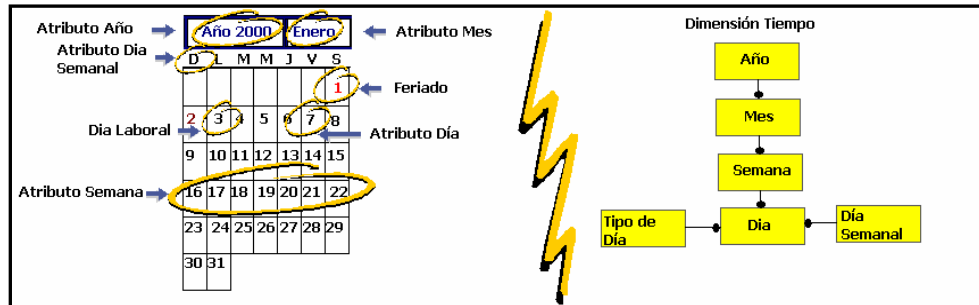


Figura 10. Relaciones y Jerarquías

### 2.3.1.6 Indicadores

Son las métricas que ayudarán a medir la eficiencia del negocio. Existen dos tipos de indicadores: básicos y derivados. Los indicadores básicos, primitivos o crudos existen físicamente en el *Data Warehouse* junto a los atributos que los caracterizan, pueden venir de diferentes sistemas fuentes y tener distintos niveles de granularidad o agregación. Por ejemplo, la métrica Importe de Venta es llevada diariamente mientras que el indicador Unidades en Stock es seguido semanalmente.

Por otro lado, los indicadores derivados o métricas calculadas se construyen a partir de los indicadores base y pueden o no estar almacenados físicamente en el *Data Warehouse* (es típicamente una decisión de *tuning*). Un ejemplo clásico de métrica derivada es margen de ganancia, la cual se define como la resta entre Precio y Costo (ambos indicadores básicos).

### 2.3.2 Construcción básica del esquema físico

Como se sabe, un esquema físico relacional está organizado, básicamente, con los siguientes componentes:

- Tablas
- Nombre de tabla
- Nombres de columnas
- Tipos de datos de columnas
- Definición de clave primaria
- Referencias de claves foráneas

Se debe de traducir el modelo dimensional del negocio al “lenguaje” que entienden las bases de datos. Este mapeo puede realizarse siguiendo unos pasos básicos que finalizan en un esquema normalizado. Luego de obtener el esquema pueden realizarse diversas técnicas de des-normalización que apunten a mejorar los tiempos de respuestas de las consultas realizadas sobre el esquema definido. También puede identificarse cuán dinámicas o estáticas son las relaciones entre los atributos con el objetivo de planificar el seguimiento de la historia entre estas relaciones.

### 2.3.3 Mapeo desde modelo lógico hacia esquema físico

A la hora de definir el esquema físico del modelo dimensional se identifican tres tipos de tablas.

- **Tablas *Lookup*:** Contiene una columna identificadora (*id*) y, si existe, una columna de descripción del atributo que representa
- **Tablas relacionales:** Contiene el *id* de dos o más atributos definiendo, de esta forma, la asociación entre ellos
- **Tablas de hechos:** Contiene columnas que representan los indicadores o variables del negocio e *id* de los atributos que indican el nivel de granularidad de las métricas. Se ve entonces que surgen tres tipos de columnas o campos:

#### Identificación

- Contiene códigos de identificación
- Requerido para todos los atributos
- Preferiblemente numéricos (procesamiento más rápido)

## Descripción

- Contiene descripciones, generalmente, en formato texto.
- Opcional para todos los atributos

## Variables (Fact Columns)

- Numéricas

### 2.3.4 Construcción de tablas Lookup

Se comienza asociando una *lookup* a cada atributo definido en el BDM de tal forma de ir construyendo un esquema normalizado. Se identifica la clave primaria y la descripción de cada atributo para representarla como los correspondientes campos de la tabla *lookup* como se muestra en la Figura 11. Dado que el campo descripción es opcional puede dejarse sólo el identificador y utilizarlo con ambos propósitos.

Existen algunos atributos característicos, como por ejemplo el domicilio de un cliente, que no tienen su propia *lookup* sino que residen en la *lookup* de algún otro atributo (por ejemplo, en este caso estaría incluido en la *lookup* cliente).

Si las tablas *lookups* representan una dimensión completa y no un atributo en particular, entonces se les conoce también como tablas dimensionales y son utilizadas ampliamente en el esquema estrella (*Star Schema*).

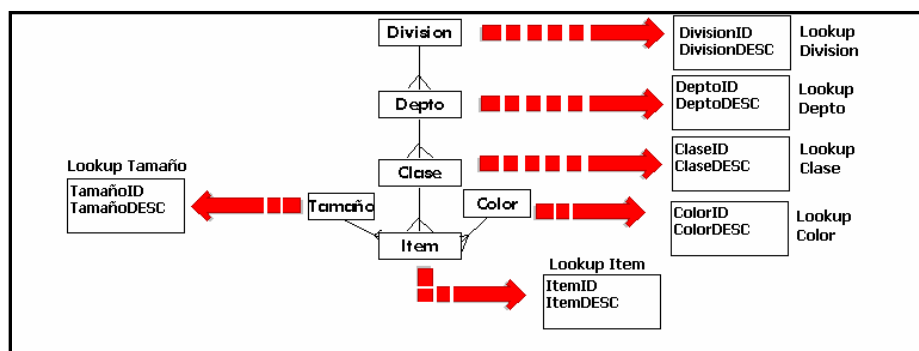


Figura 11. Tablas *Lookup*

### 2.3.5 Relaciones de atributos dentro de una dimensión

Se identifica las relaciones directas entre los distintos atributos (relaciones Padre-Hijo), después se define una tabla para cada relación que contiene los campos, *ids*, de cada *lookup* que representa los atributos asociados. Como se realiza comúnmente, la definición de la clave primaria estará asociada a la “cardinalidad” de la relación relevada en el BDM. En el caso de las relaciones que no sean de muchos-a-muchos se puede evitar la creación de una tabla relacional, incluyendo el *id* del padre en la *lookup* del hijo de la relación y definiéndola como clave foránea referenciándola a la *lookup* padre. De esta forma también se pueden incluir los *ids* de todos sus ancestros y mejorar los tiempos de respuesta al reducir la cantidad de *joins* en las consultas, incrementando, por otro lado el mantenimiento de las tablas un ejemplo se ilustra en la Figura 12.

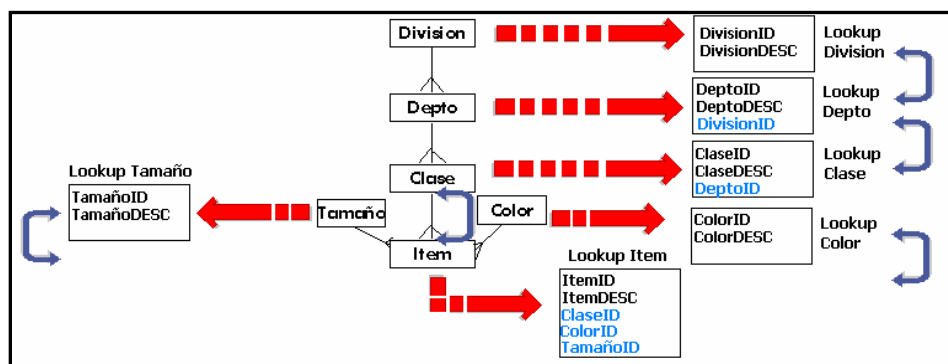


Figura 12. Implementación de relaciones

### 2.3.6 Definición de tablas de hechos

Se debe definir una tabla para cada conjunto de indicadores, agrupados según su granularidad. Como se explicó anteriormente, cada tabla de hechos (*fact table* o *base table*) tiene campos numéricos asociados a los indicadores y campos *ids* para cada atributo que interviene en la “dimensionalidad” y “granularidad” de la variable a medir. La “dimensionalidad” de una variable está asociada a los conceptos generales (dimensiones) que participan en la definición de un indicador y la granularidad al nivel de detalle o agregación de la misma y define los atributos de cada dimensión que caracterizan el valor de la variable. Por ejemplo, si las ventas se cargan diariamente para cada sucursal y cada familia de productos, la dimensión del indicador Unidades Vendidas será Tiempo-



Geografía-Productos y la granularidad será Dia-Sucursal-Familia, en cambio, si el *stock* de mercadería se sigue semanalmente, la “dimensionalidad” de Unidades en *Stock* será la misma pero la granularidad será Semana-Sucursal-Familia.

La clave primaria de la tabla base estará compuesta por todos los *ids* que definan la “dimensionalidad” de los indicadores y cada *id* individualmente referenciará, como clave foránea, a la tabla *lookup* que represente el atributo asociado como muestra la Figura 19.

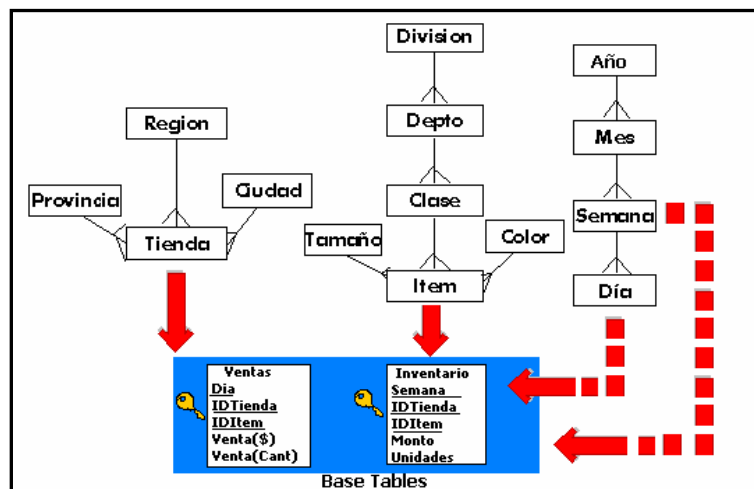


Figura 13 Tabla de Hechos

### 2.3.7 Esquema estrella normalizado I

La tabla base tiene un único nivel de datos (no tiene agregaciones incluidas). La clave primaria está formada por los *id* de los atributos hojas de las dimensiones. Este esquema es comúnmente conocido con el nombre de esquema copo de nieve (*SnowFlaked Schema*). Como puede verse, este esquema se encuentra en tercera forma normal en la Figura 14.

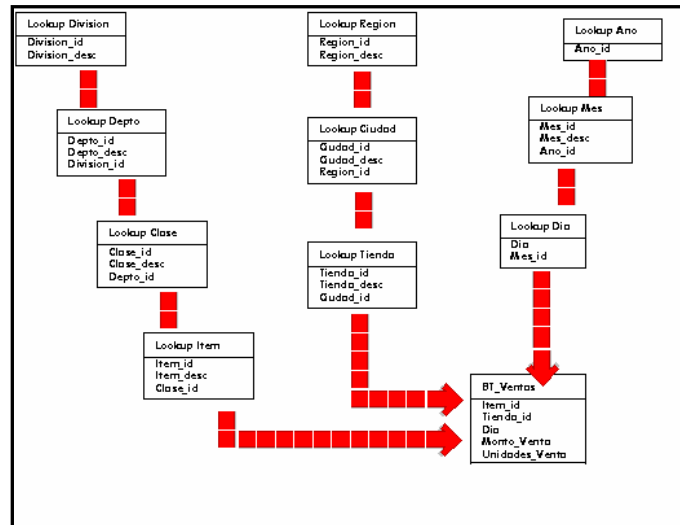


Figura 14. Esquema Estrella Normalizado I

### 2.3.8 Esquema estrella normalizado II

A las tablas *lookups* se le incorpora los *ids* de todos sus ancestros (todas los *ids* de las tablas que están por encima de ella en la jerarquía de la dimensión). La estructura de la tabla base se mantiene inalterable con respecto al esquema anterior véase en la Figura 15).

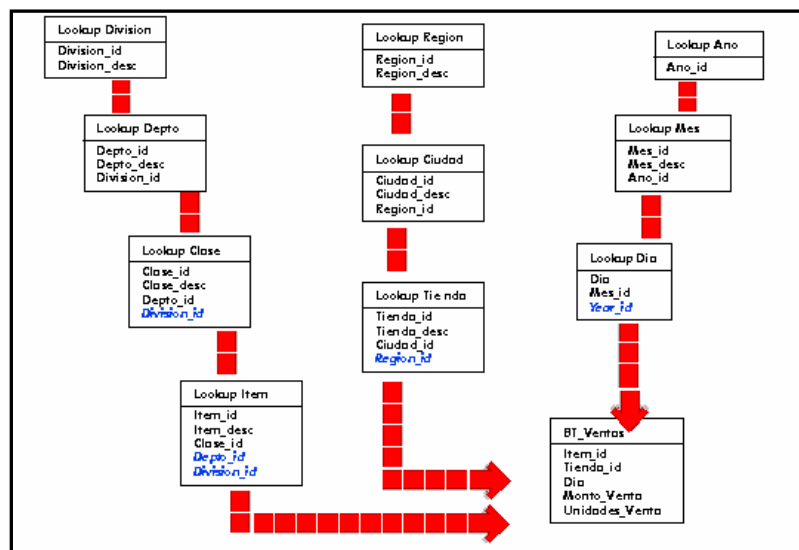


Figura 15. Esquema Estrella Normalizado II

### 2.3.9 Esquema estrella normalizado III

Incluye en las tablas *lookups* no sólo los *ids* de sus ancestros sino también las descripciones de los mismos. La estructura de la tabla base no sufre modificaciones véase en la Figura 16.

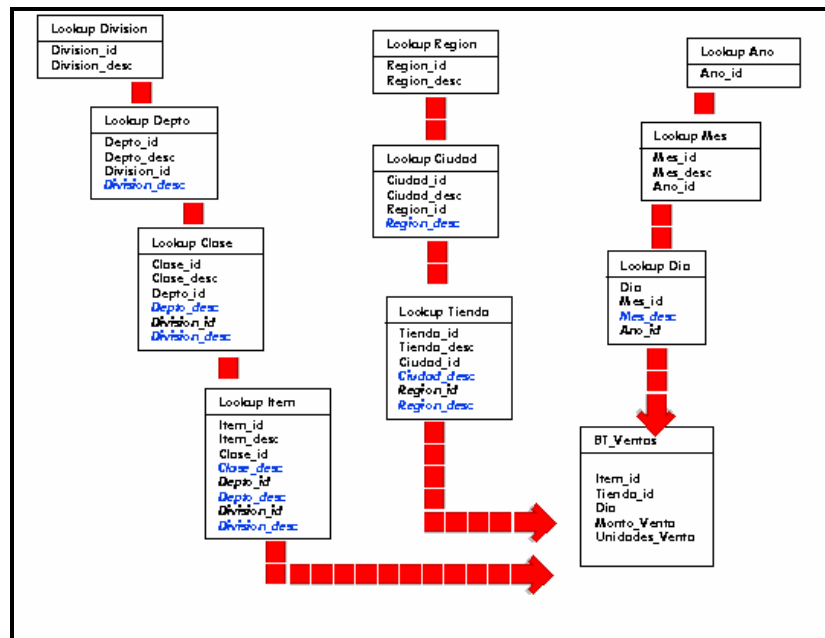
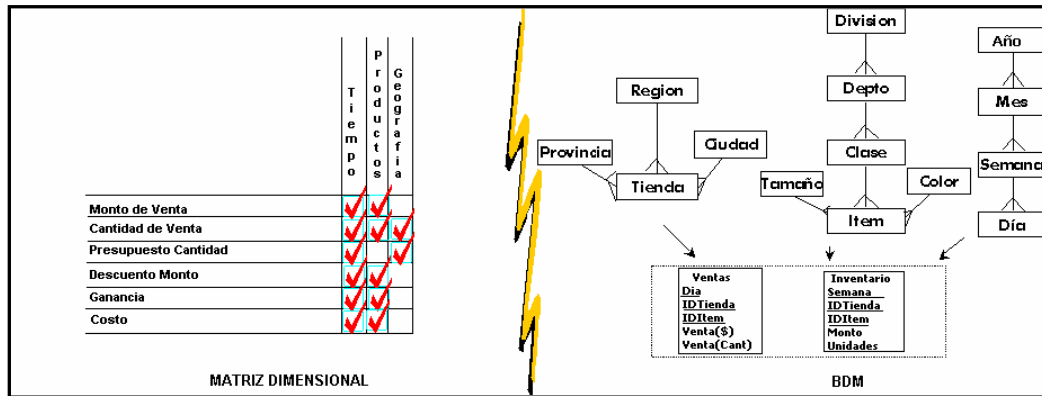


Figura 16. Esquema Estrella Normalizado III

La definición de los requerimientos del negocio determina los datos necesarios para cumplir los requerimientos analíticos de los usuarios. Básicamente se comienza con la matriz mostrada en la Figura 17, donde se determina cada indicador y después se especifican los diferentes grados de detalle (atributos) dentro de cada concepto del negocio (dimensión), como así también la granularidad de cada indicador (variable o métrica) y las diferentes jerarquías que dan forma al Modelo Dimensional del Negocio BDM.



**Figura 17.** Diagramas para Modelado Dimensional

Para construir modelos dimensionales en ambientes reales se comienza fijando los prerequisites para realizar el BDM, después identifica los correspondientes *Data Marts* (como conjunto de indicadores), las correspondientes dimensiones, luego sigue con la construcción de una matriz dimensional de alto nivel (*Data Warehouse Bus Architecture Matrix*), donde las filas son los *Data Marts* y las columnas son todas las dimensiones marcando la intersección de aquellas dimensiones que están en juego en determinados *Data Marts*.

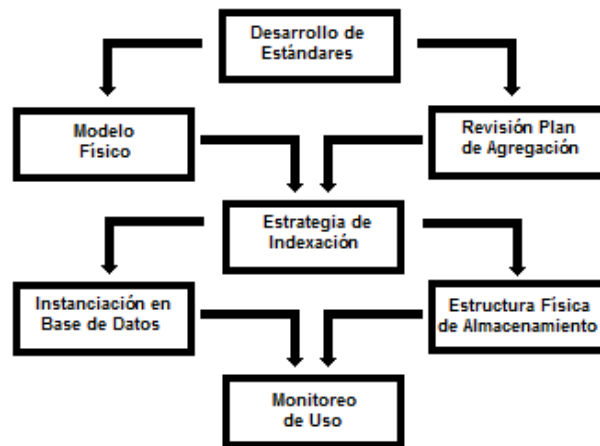
La metodología continúa con el diseño de cada *Data Mart* con los siguientes pasos:

- Elección del *Data Mart*,
- Declaración de la granularidad
- Elección de las dimensiones
- Elección de las *facts* (el centro de un modelo estrella) o indicadores
- Se recomienda construir un conjunto de gráficos y diagramas que apoyen a la implementación de esta etapa

## 2.4 Diseño físico

El diseño físico de las base de datos se centra sobre la selección de las estructuras necesarias para soportar el modelado dimensional. Algunos de los elementos principales de este proceso son la definición de convenciones estándares de nombres y definiciones específicas del ambiente de la base de datos. La indexación y las estrategias de partición

son también determinadas en esta etapa. La secuencia de etapas se observa en la Figura 18.



**Figura 18.** Proceso de diseño físico a alto nivel

Uno de los problemas inherentes al modelado dimensional y a su implementación relacional es el llamado *n-way join*, que se refiere al hecho de tener que hacer *join* de cada lookup contra la FT, (*Fact Table*), de una por vez. En algunos RDBMS esto se resuelve mediante la definición de *Star Join* en el motor de base de datos, mejorando el rendimiento en unas 60 veces por sobre la utilización del *join* secuencial en un equipo de iguales características.

El *Star Join* realiza todas las combinaciones posibles entre el criterio de selección fijado y con este conjunto realiza el *join* con la FT correspondiente. Dado que esta técnica demanda la realización de un producto cartesiano, algunos motores no la proveen pues en teoría el producto cartesiano no es recomendable, pero en el caso de un modelo dimensional es mucho más eficiente que hacer el *join* entre *lookups* y FT una a una. No sólo para el acceso a los indicadores o métricas se necesita la optimización del *join*, la consulta de todo modelo dimensional involucra la necesidad de la utilización de esta sentencia altamente costosa en los RDBMS, pues constantemente se están consultando los datos para tener diferentes vistas de la evolución y diferentes entornos de análisis de los indicadores del negocio.

### 2.5 Diseño y desarrollo del Data Staging

Las principales sub-etapas de esta zona del ciclo de vida son: la extracción, la transformación y la carga (*ETL process*).

Se definen como procesos de extracción a aquellos requeridos para obtener los datos que permitirán efectuar la carga del Modelo Físico acordado. Así mismo, se definen como procesos de transformación los procesos para convertir o recodificar los datos fuente a fin poder efectuar la carga efectiva del Modelo Físico. Por otra parte, los procesos de carga de datos son los procesos requeridos para poblar el *Data Warehouse*.

La calidad de los datos es un factor determinante en el éxito de un proyecto de *Data Warehouse*. Es en esta etapa donde deben depurarse todos los inconvenientes relacionados con la calidad de los datos fuente. Se proponen 10 puntos que ayudarán a guiar esta etapa del BDL:

#### Plan

1. Crear un diagrama esquemático de flujo fuente-destino, de una página y de muy alto nivel.
2. Probar, elegir e implementar una herramienta de *data staging*.
3. Profundizar en detalle por tabla destino. Gráficamente describir las reestructuraciones o transformaciones complejas e ilustrar la generación de las claves subrogadas (claves definidas por los desarrolladores). Desarrollo preliminar de la secuencia de los trabajos.

#### Carga de dimensiones:

4. Construir y probar la carga de una tabla dimensional estática. La principal meta de este paso es resolver los problemas de infraestructura que pudieran surgir (conectividad, transferencia, seguridad, entre otros).
5. Construir y probar los procesos de actualización de una dimensión.
6. Construir y probar las cargas de las restantes dimensiones.

### ***Fact Tables* y automatización:**

7. Construir y probar la carga histórica de las *fact tables* (carga masiva de datos). Debe incluirse la búsqueda y sustitución de claves.
8. Construir y probar los procesos de cargas incrementales.
9. Construir y probar la generación de agregaciones.
10. Diseñar, construir y probar la automatización de los procesos.

## **2.6 Diseño de la arquitectura técnica**

Los ambientes de *Data Warehouse* requieren la integración de numerosas tecnologías. Se debe tener en cuenta tres factores: los requerimientos del negocio, los actuales ambientes técnicos y las directrices técnicas estratégicas futuras; para poder establecer el diseño de la arquitectura técnica del ambiente de *Data Warehouse*. Una de las ventajas que trae la utilización de una arquitectura BUS o canal, es la fácil incorporación de nuevas instancias a la arquitectura (*plug and play*).

El *Data Warehouse Architecture Framework* incluye tres áreas: el área de arquitectura de datos (el qué), el área de arquitectura técnica (el cómo) y el área de infraestructura (el dónde). A su vez, cada una de estas áreas tienen diferentes niveles de detalles: Nivel de Requerimientos del Negocio, Nivel de Modelo Arquitectónico, Nivel de Modelo Detallado y Nivel de Implementación.

La arquitectura técnica se divide en dos partes, el *back room* (la parte interna del *Data Warehouse*) y el *front room* (la cara pública del *Data Warehouse*), ambas interactuando constantemente. Mientras los requerimientos del negocio dicen qué se necesita hacer, la arquitectura técnica responde la interrogante de cómo se hace.

## **2.7 Selección de productos e instalación**

Utilizando el diseño de arquitectura técnica como marco, es necesario evaluar y seleccionar componentes específicos de la arquitectura como la plataforma de hardware, el motor de base de datos, la herramienta de ETL o el desarrollo pertinente, herramientas de acceso, entre otros, para ello se puede utilizar una matriz de evaluación de productos

como se muestra en la Figura 19. Una vez evaluados y seleccionados los componentes determinados se procede con la instalación y prueba de los mismos en un ambiente integrado de *Data Warehouse*.

Característica	Peso	Producto A	Producto B	Producto C	...
<b>Capacidades Basicas de ETL</b>					
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	85				
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	70				
<b>Job Control &amp; Scheduling</b>					
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	90				
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	85				
<b>Metada &amp; Estandares</b>					
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx					
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx					
<b>Info de Vendedor</b>	70				
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	85				
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx					
<b>Puntaje Total</b>					
<b>Ranking</b>					

Figura 19. Ejemplo de matriz de evaluación de productos

## 2.8 Diseño de la aplicación de Business Intelligence

No todos los usuarios del *Data Warehouse* necesitan el mismo nivel de análisis. Es por ello que en esta etapa se identifican los diferentes roles o perfiles de usuarios para determinar los diferentes tipos de aplicaciones necesarias con base al alcance de los diferentes perfiles (gerencial, analista del negocio, vendedor, entre otros) como se muestra en la Figura 20.

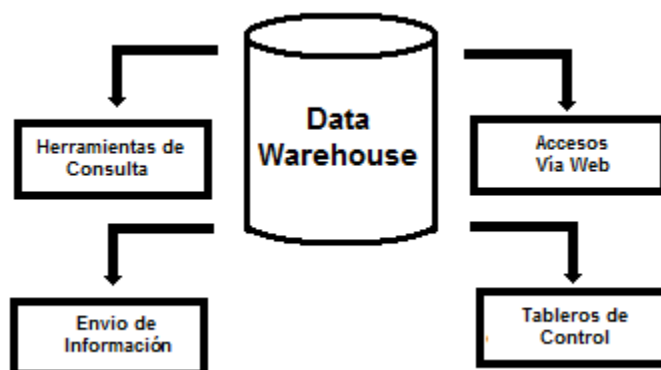


Figura 20. Variedad de interfaces por perfil



Los diferentes roles o perfiles de usuarios determinan la interfaz o ventana al *Data Warehouse*, por ejemplo: herramientas de diseño de reportes y consultas avanzadas para analistas, tableros de control para gerentes, acceso mediante *Internet* para usuarios internos/externos y envío de información por dispositivos no estándares para usuarios internos/externos.

Se debe poner especial atención sobre el proceso de creación de aplicaciones “*templates*”. El diseño de la plantilla empieza al definir el concepto de la aplicación dirigido hacia el usuario final.

Se divide el proceso de creación de las aplicaciones para usuarios finales en dos grandes fases: especificación y desarrollo. Se clasifican a los usuarios según su perfil de consulta, desde usuarios con un perfil más estratégico y menos predecibles (*power users*) hasta usuarios netamente operacionales que consumen una serie de reportes estándares (*final users*) pasando por los usuarios gerenciales o EIS (*Executive Information System*) con uso de interfaces del tipo interactivo.

Otro tipo de aplicaciones para usuarios finales son las de análisis *ad-hoc* en la cual se pueden cambiar los parámetros sobre un reporte para crear sus propias versiones personalizadas del mismo. De esta forma se dará respuesta a necesidades como: “*Quiero ver este reporte, por mes en lugar de trimestre*”, “*¿Puedo ver este reporte a nivel provincia en lugar de región?*”, “*¿Puedo ver esta evolución de ventas mensual pero sólo de mi equipo de ventas?*”. Las respuestas a estas preguntas las resuelven los propios usuarios sin la necesidad de la intervención del departamento sistemas y maximizando el tiempo de análisis sobre el tiempo de construcción e integración de la información.

Los *templates* para usuarios finales proveen el marco (*layout*) y la estructura de un reporte para ser especificado por un conjunto de parámetros. El usuario selecciona los parámetros de una lista desplegable (*pick list*) o aceptando los valores por defecto cuando ejecuta el *template*. Este enfoque orientado a parámetros permite a los usuarios generar varios reportes de estructura similar desde un mismo *template*. Todo esto de forma simple y con interfaces gráficas que hacen uso de todo el trabajo realizado en la construcción del *Data Warehouse* en las etapas previas del BDL.

Se destacan cuatro pasos principales para el desarrollo de los *templates* (siempre enfatizando el hecho de involucrar a los usuarios en cada uno de estos pasos):

- Determinación del conjunto de *templates* iniciales (identificar reportes candidatos, clasificarlos y priorizarlos)
- Diseño de la estrategia de navegación dentro de la aplicación (esquema de pantallas, esquema de carpetas –directorios-, criterios de agrupamiento -por datos, por dueño, por regla del negocio, entre otros-)
- Determinación de estándares (nombre de objetos, ubicación de objetos, formato de las salidas)
- Detalle de las especificaciones (definición: nombre, descripción o propósito, frecuencia, parámetros, restricciones, *layout*, entre otros)

### 2.9 Desarrollo de aplicación de Business Intelligence

Siguiendo a la especificación de las aplicaciones para los usuarios finales, el desarrollo de las aplicaciones para los usuarios finales involucra configuraciones del *metadata* y construcción de reportes específicos.

Ya que se han completado los pasos de la especificación y se tiene la posibilidad de trabajar con algunos datos de prueba se comienza el desarrollo de la aplicación. Para ello, deben considerarse los siguientes aspectos:

#### a) Selección de un enfoque de implementación

- Basado en *Web*
- *Internet/Intranet*
- Usuarios altamente distribuidos
- Manejo centralizado de nuevas versiones
- Herramienta propietaria
- Mayor complejidad de uso
- Para usuarios más capacitados
- Instalación local

- EIS (*Executive Information System*)
- Acceso estructurado
- Secuencia de pantallas
- Vista Interactiva
- Interfaz personalizada
- API (*Application Programming Interface*)

### **b) Desarrollo de la aplicación**

- Definición de herramienta de acceso al *metadata*
- Desarrollo de *templates* y esquema de navegación de la aplicación
- Selección de reportes para ejecución previa

### **c) Prueba y verificación de datos**

- Descripciones
- Información duplicada
- Relaciones entre atributos
- Consistencia e integridad de datos con sistemas fuentes

### **d) Documentación y *Roll Out***

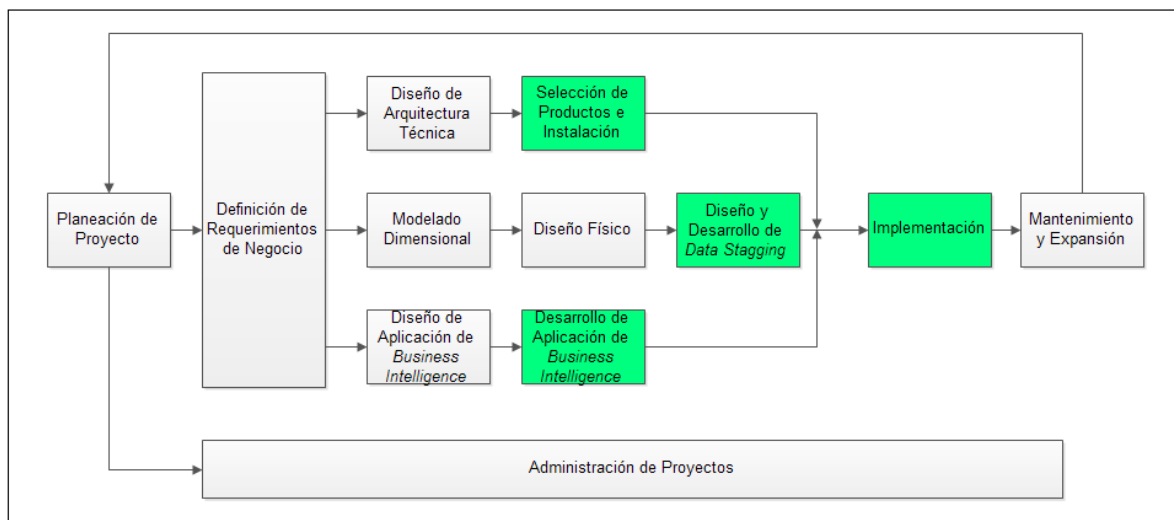
- Retroalimentación con los resultados de la puesta en producción

### **e) Mantenimiento**

- Nuevos *templates*
- Incorporación de nuevos sistemas fuentes
- Monitoreo de desempeño
- Eliminación de *templates* en desuso

## 2.10 Implementación

La implementación representa la convergencia de la tecnología, los datos y las aplicaciones de usuarios finales accesible desde el escritorio del usuario del negocio como se muestra en la Figura 21. Hay varios factores extras que aseguran el correcto funcionamiento de todos estos elementos, entre ellos se encuentran: la capacitación, el soporte técnico, la comunicación y las estrategias de retroalimentación. Todas estas tareas deben tenerse en cuenta antes de que cualquier usuario pueda ingresar al *Data Warehouse*.



**Figura 21.** Implementación como convergencia de la tecnología, los datos y las aplicaciones

Se tiene que realizar un *checklist* de tareas que incluyen: configuración de hardware, conexión a las bases, acceso a *intranet* o *internet*, direcciones LAN *Local Area Network* (si no son dinámicamente asignadas), auditorías de tecnología sobre las configuraciones en las que se encontraban las PC *Personal Computer*, prever actualizaciones de *hardware* y *software* (determinando responsables, proyecto o área de usuario), verificaciones de seguridad (*logon* de red y base de datos), prueba de procedimientos de instalación en una variedad de máquinas, planificación de instalación con la correspondiente capacitación a los usuarios, entre otros.

Se tienen dos aspectos a los cuales se les debe poner énfasis, por un lado el alcance de la capacitación, y por otro, la metodología de presentación del sistema. Para lo primero,

debe instruirse al usuario en tres aspectos claves: contenido del *Data Warehouse*, aplicación y herramientas de acceso. En cuanto a la metodología, el usuario final debe entender los límites entre las aplicaciones, los contenidos y las herramientas del *Data Warehouse*.

El *Data Warehouse* es un todo, no la suma de componentes discretos. Por lo tanto, la capacitación también debe reflejar la misma perspectiva. No debe interpretarse la capacitación de los usuarios finales como sólo el adiestramiento sobre las herramientas de acceso pues esto resultaría inútil al menos que se complemente con la descripción del contenido en el *Data Warehouse* (cuáles datos están disponibles, qué significan, cómo se usan y para qué usarlos). Otro factor clave es la correcta entrega de niveles de capacitación según el perfil del usuario (usuario final, *power user*, entre otros).

El ensamble de un ambiente de capacitación (como subconjunto de datos de producción) es siempre recomendable, dado que los tiempos de respuesta en producción, si bien pueden ser apropiados para el análisis de la información, tal vez no sean los mejores para un curso de un día de duración. Se debe capacitar al usuario final sólo si el *Data Warehouse* está terminado (en tiempo y forma) y establecer la política de “*Sin capacitación, no hay acceso*”.

Se deberán definir claramente los niveles de soporte (a nivel aplicación, modelo, calidad de datos) de forma transparente para el usuario. La documentación tiene un papel importante en la ayuda a usuarios finales, dado que el *Data Warehouse* evolucionará rápidamente y la documentación escrita puede quedar obsoleta prematuramente.

Para la metodología de implementación se propone un esquema de versiones (*release framework*). Primero se pasa por la versión *Alpha*, primera oportunidad para el grupo de trabajo de conducir una prueba del sistema de principio a fin. Todos los componentes del sistema deben ser probados (infraestructura técnica, extracción, transformación, carga, procedimientos de calidad, *performance*, *templates*, entre otros).

Durante la etapa *Alpha*, a un conjunto limitado de usuarios finales se les permite el acceso al *Data Warehouse* (sólo aquellos que pertenecen al grupo de trabajo).

Posteriormente se realiza la versión *Beta*. El objetivo de esta versión es conducir una prueba a nivel usuario de principio a fin. El grupo *Beta* está formado por los *power users*. La determinación de la cantidad de usuarios en esta versión es importante, pues si son pocos, la aplicación no se probará adecuadamente; y si son muchos, se estará omitiendo la etapa de pruebas, lo cual implica directamente en una salida a producción.

Una vez superadas estas dos versiones, al igual que en un proceso de diseño e implementación de software, se llega a un estado GA (*General Availability*). Se sugiere que todo cambio y/o modificación que se realice posteriormente al *Data Warehouse*, pase internamente por un estado *Alpha* y *Beta* aunque externamente sea una nueva versión, desde el último GA.

### 2.11 Mantenimiento y expansión

Una vez que se ha construido e implantado el *Data Warehouse* se debe estar preparado para administrar el mantenimiento y la expansión del mismo. Si bien las tareas pueden llegar a parecer similares a las tratadas en otras etapas del BDL, existe una diferencia clave: los usuarios están ahora accediendo al *Data Warehouse*. Existen puntos a tomar en cuenta para mantener exitosamente el *Data Warehouse*. Entre ellos se destacan: el continuo soporte y la constante capacitación a usuarios de negocios, el manejo de la infraestructura (monitoreo de base de datos, tráfico, entre otros), *tuning* de rendimiento sobre las consultas, mantenimiento del *metadata* y procesos ETL.

Se recomienda la creación de un comité, conformado por analistas del negocio y analistas de sistemas, que establezca prioridades y procedimientos. El manejo de prioridades lo discrimina según sean mejoras menores o mayores (clasificadas según el impacto en el *Data Warehouse* y excluyendo los errores). Como mejoras menores incluye incorporación de datos sencillos, nuevas agregaciones, cambios en niveles superiores del modelo (atributos de alto nivel, lejanos a las *bases tables*).

El BDL, se debe volver a iterar sobre él mismo pasando nuevamente por cada una de las etapas para de esta forma administrar correctamente los cambios al *Data Warehouse*.

## 2.12 Administración del proyecto

La administración del proyecto asegura que las actividades del BDL se lleven en forma y sincronización. Como lo indica el diagrama de la Figura 22, la administración acompaña todo el ciclo de vida. Entre sus actividades principales se encuentran: el monitoreo del estado del proyecto y la comunicación entre los requerimientos del negocio, así como las restricciones de información para poder manejar correctamente las expectativas en ambos sentidos.

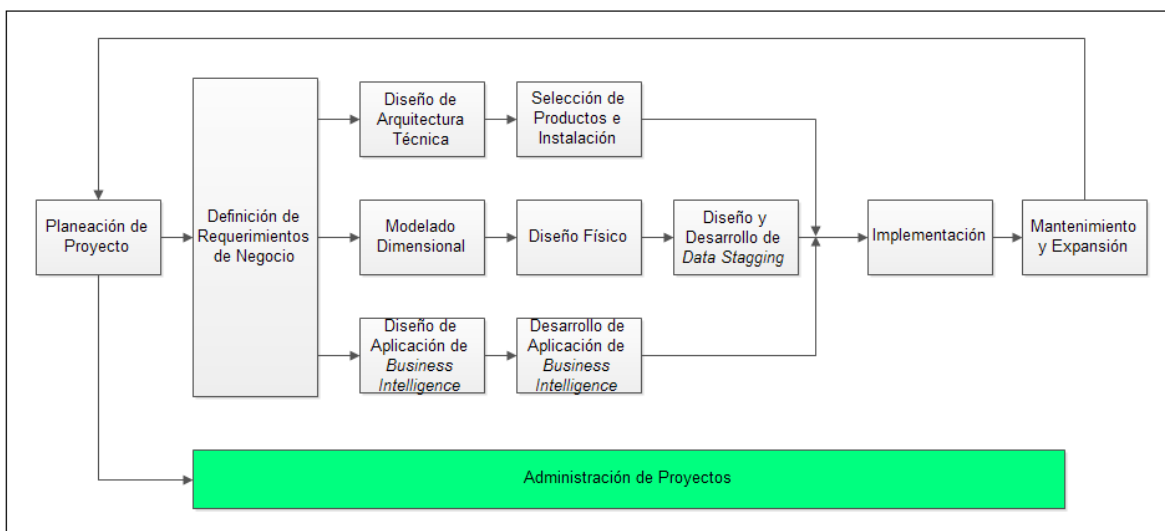


Figura 22. Administración del proyecto

## 2.13 Puntos problemáticos de un Data Warehouse

Los principales puntos que pueden llegar a complicar un proyecto de *Data Warehouse* se localizan en tres áreas:

- **Rutinas de Carga.** Incluye programas de extracción y limpieza de datos. Surgen problemas en este punto dada la falta de integración y estructura consistente (alineada) entre los sistemas fuentes.
- **Mantenimiento.** Dados los diferentes períodos de almacenamiento para OLTP y OLAP y el hecho de que los *Data Warehouse* son sistemas secundarios de información, el

problema surge para sincronizar los datos entre los sistemas operacionales fuentes y los *Data Warehouse*.

- ***Tuning***. Dado los patrones de uso y los métodos de acceso de los sistemas OLAP, diseñadores y administradores deben realizar cambios significativos a los implementados en el *tuning* de sistemas OLTP.





# **CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE**

### 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Para probar que la metodología BDL expuesta en este trabajo es válida, se construye un sistema ejemplo de inteligencia de negocios.

#### 3.1 Planificación del Proyecto

**Delimitación de Alcance del Proyecto.**

**Área:** Ventas

**Objetivo:** Determinar con respecto al tiempo las ventas de la organización por producto, cliente, país del cliente, canal de venta y promociones.

#### 3.2 Definición de Requerimientos del Negocio

##### 3.2.1 Requerimiento Funcional

*“Se requiere conocer las ventas que está realizando la institución desde varios enfoques, para lo cual se necesita un tablero de control que detecte lo rentable que está siendo el negocio y así compararlo contra las ventas de años anteriores.”*

##### 3.2.2 Requerimiento Técnico

*“Se necesita adquirir un sistema de Business Intelligence que pueda ser útil para presentar una gran diversidad de Informes a la Dirección General.”* Como ejemplo, se muestra el *layout* de la Figura 23.

Año	Cliente	País del Cliente	Canal de Venta	Producto	Promoción	Importe de Venta
-----	---------	------------------	----------------	----------	-----------	------------------

**Figura 23.** *Layout* Reporte de Ventas

### 3.3 Modelado Dimensional de Negocio (BDM)

El Modelo Dimensional de Negocio debe de ser analizado con detenimiento, es importante conocer el modelo de la fuente de datos, porque a partir de él se formulan las interrogantes y se dan respuestas a ellas.

#### 3.3.1 Matriz Dimensional

Es importante realizar una matriz dimensional ya que con esta se puede observar la relación que existe entre los Hechos y las Dimensiones. En la Figura 24 se observa que los hechos de Importe de Ventas e Importe de Costos de Producto pueden relacionarse mediante las dimensiones de tiempo y producto.

Hechos	Dimensiones					
	Cliente	País del Cliente	Tiempo	Producto	Canal de Venta	Promoción
Importe de Ventas	x		x	x	x	x
Importe de Costos de Producto			x	x		

**Figura 24.** Matriz Dimensional Ventas

El BDM que se propone en este trabajo está enfocado en el área ventas, los hechos, cantidad de venta e importe de venta están definidos en la dimensión ventas, la cual es el centro del modelo estrella, por lo que se puede conocer la cantidad e importe de venta por producto, canal de venta, cliente, país del cliente y con respecto al tiempo. El modelo se muestra en la Figura 25.

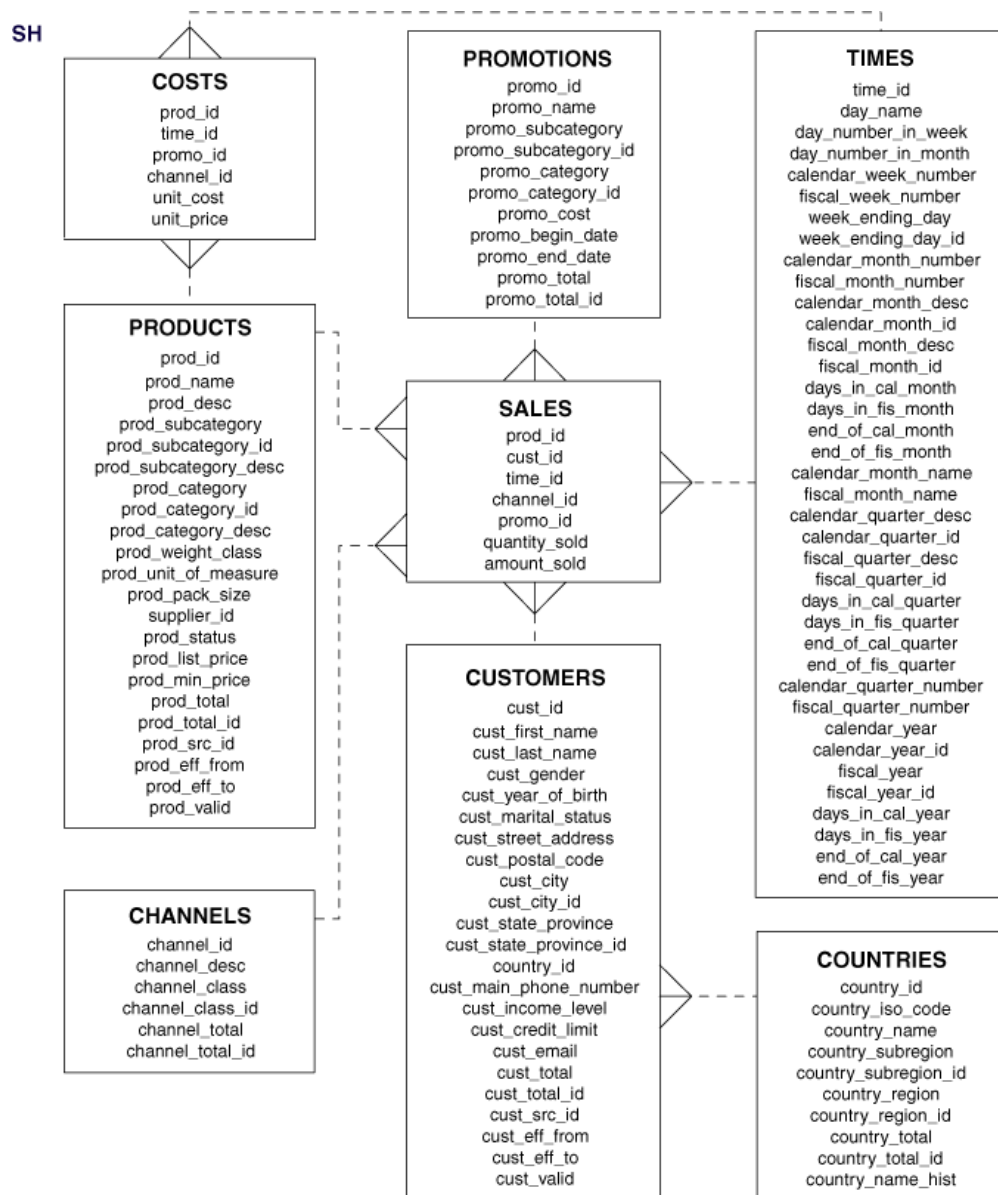


Figura 25. BDM Ventas

### 3.4 Diseño Físico

Después de conocer los requerimientos y el modelo de negocio se prosigue con el diseño físico del mismo. Por parte del requerimiento se cuenta con una Base de Datos *Oracle* con las siguientes características:

#### ***Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 – 64 bit Production***

- *PL/SQL Release 11.2.0.1.0 - Production*
- *"CORE 11.2.0.1.0 Production"*
- *TNS for 64-bit Windows: Version 11.2.0.1.0 - Production*
- *NLSRTL Versión 11.2.0.1.0 - Production*

Además de un Cliente de Base de Datos:

#### ***Oracle SQL Developer 3.1.06.82***

- *Plataforma Java(TM) 1.6.0\_34*

La base de datos incluye un esquema de ejemplo que ya está “poblado” con datos, con el nombre de SH (*Sales History*), que se toma de ejemplo en este trabajo para su análisis. A continuación se describen a detalle las tablas, atributos, tipo de datos, índice, entre otros.

En donde el prefijo *FACT* indica que la tabla es de hechos y *DIM* dimensión.

**DIM\_CHANELS.** La dimensión Canal de Venta contiene información de los diferentes medios por los cuales se ha realizado una venta por ejemplo (véase la Tabla 1): ventas directas, ventas por distribuidor, ventas por internet, entre otras.

COLUMN_NAME	DATA_TYPE	NULLABLE	COMMENTS
CHANNEL_ID	NUMBER	No	<i>primary key column</i>
CHANNEL_DESC	VARCHAR2(20 BYTE)	No	<i>e.g. telesales, internet, catalog</i>
CHANNEL_CLASS	VARCHAR2(20 BYTE)	No	<i>e.g. direct, indirect</i>
CHANNEL_CLASS_ID	NUMBER	No	
CHANNEL_TOTAL	VARCHAR2(13 BYTE)	No	
CHANNEL_TOTAL_ID	NUMBER	No	

INDEX_OWNER	INDEX_NAME	UNIQUENESS	INDEX_TYPE	JOIN_INDEX	COLUMNS
SH	CHANNELS_PK	UNIQUE	NORMAL	NO	CHANNEL_ID

Tabla 1. Dimensión Canal de Venta

**DIM\_COUNTRIES.** La dimensión países contiene datos acerca del país del cliente, esta información es de importancia ya que así se puede conocer los países o regiones en los que determinado producto se vende con facilidad (véase la Tabla 2).

COLUMN_NAME	DATA_TYPE	NULLABLE	COMMENTS
COUNTRY_ID	NUMBER	No	<i>primary key</i>
COUNTRY_ISO_CODE	CHAR(2 BYTE)	No	
COUNTRY_NAME	VARCHAR2(40 BYTE)	No	<i>country name</i>
COUNTRY_SUBREGION	VARCHAR2(30 BYTE)	No	<i>e.g. Western Europe, to allow hierarchies</i>
COUNTRY_SUBREGION_ID	NUMBER	No	
COUNTRY_REGION	VARCHAR2(20 BYTE)	No	<i>e.g. Europe, Asia</i>
COUNTRY_REGION_ID	NUMBER	No	
COUNTRY_TOTAL	VARCHAR2(11 BYTE)	No	
COUNTRY_TOTAL_ID	NUMBER	No	
COUNTRY_NAME_HIST	VARCHAR2(40 BYTE)	Yes	

INDEX_OWNER	INDEX_NAME	UNIQUENESS	INDEX_TYPE	PARTITIONED	JOIN_INDEX	COLUMNS
SH	COUNTRIES_PK	UNIQUE	NORMAL	NO	NO	COUNTRY_ID

Tabla 2. Dimensión País del Cliente

**DIM\_CUSTOMERS.** La dimensión Clientes (véase la Tabla 3) contiene la información relacionada con el cliente como pueden ser datos personales y el límite de crédito que se le otorga.

COLUMN_NAME	DATA_TYPE	NULLABLE	COMMENTS
CUST_ID	NUMBER	No	<i>primary key</i>
CUST_FIRST_NAME	VARCHAR2(20 BYTE)	No	<i>first name of the customer</i>
CUST_LAST_NAME	VARCHAR2(40 BYTE)	No	<i>last name of the customer</i>
CUST_GENDER	CHAR(1 BYTE)	No	<i>gender; low cardinality attribute</i>
CUST_YEAR_OF_BIRTH	NUMBER(4,0)	No	<i>customer year of birth</i>
CUST_MARITAL_STATUS	VARCHAR2(20 BYTE)	Yes	<i>customer marital status; low cardinality attribute</i>
CUST_STREET_ADDRESS	VARCHAR2(40 BYTE)	No	<i>customer street address</i>
CUST_POSTAL_CODE	VARCHAR2(10 BYTE)	No	<i>postal code of the customer</i>
CUST_CITY	VARCHAR2(30 BYTE)	No	<i>city where the customer lives</i>
CUST_CITY_ID	NUMBER	No	
CUST_STATE_PROVINCE	VARCHAR2(40 BYTE)	No	<i>customer geography: state or province</i>
CUST_STATE_PROVINCE_ID	NUMBER	No	
COUNTRY_ID	NUMBER	No	<i>foreign key to the countries table (snowflake)</i>
CUST_MAIN_PHONE_NUMBER	VARCHAR2(25 BYTE)	No	<i>customer main phone number</i>
CUST_INCOME_LEVEL	VARCHAR2(30 BYTE)	Yes	<i>customer income level</i>
CUST_CREDIT_LIMIT	NUMBER	Yes	<i>customer credit limit</i>
CUST_EMAIL	VARCHAR2(30 BYTE)	Yes	<i>customer email id</i>
CUST_TOTAL	VARCHAR2(14 BYTE)	No	
CUST_TOTAL_ID	NUMBER	No	
CUST_SRC_ID	NUMBER	Yes	
CUST_EFF_FROM	DATE	Yes	
CUST_EFF_TO	DATE	Yes	
CUST_VALID	VARCHAR2(1 BYTE)	Yes	

INDEX_OWNER	INDEX_NAME	UNIQUENESS	INDEX_TYPE	JOIN_INDEX	COLUMNS
SH	CUSTOMERS_PK	UNIQUE	NORMAL	NO	CUST_ID
SH	CUSTOMERS_YOB_BIX	NONUNIQUE	BITMAP	NO	CUST_YEAR_OF_BIRTH
SH	CUSTOMERS_GENDER_BIX	NONUNIQUE	BITMAP	NO	CUST_GENDER
SH	CUSTOMERS_MARITAL_BIX	NONUNIQUE	BITMAP	NO	CUST_MARITAL_STATUS

**Tabla 3. Dimensión Cliente**

**DIM\_PRODUCTS.** La dimensión producto informa las características principales de producto como son la categoría, unidad de medida, precio de lista entre otros como se muestra en la Tabla 4.

COLUMN_NAME	DATA_TYPE	NULLABLE	COMMENTS
PROD_ID	NUMBER(6,0)	No	<i>primary key</i>
PROD_NAME	VARCHAR2(50 BYTE)	No	<i>product name</i>
PROD_DESC	VARCHAR2(4000 BYTE)	No	<i>product description</i>
PROD_SUBCATEGORY	VARCHAR2(50 BYTE)	No	<i>product subcategory</i>
PROD_SUBCATEGORY_ID	NUMBER	No	
PROD_SUBCATEGORY_DESC	VARCHAR2(2000 BYTE)	No	<i>product subcategory description</i>
PROD_CATEGORY	VARCHAR2(50 BYTE)	No	<i>product category</i>
PROD_CATEGORY_ID	NUMBER	No	
PROD_CATEGORY_DESC	VARCHAR2(2000 BYTE)	No	<i>product category description</i>
PROD_WEIGHT_CLASS	NUMBER(3,0)	No	<i>product weight class</i>
PROD_UNIT_OF_MEASURE	VARCHAR2(20 BYTE)	Yes	<i>product unit of measure</i>
PROD_PACK_SIZE	VARCHAR2(30 BYTE)	No	<i>product package size</i>
SUPPLIER_ID	NUMBER(6,0)	No	<i>this column</i>
PROD_STATUS	VARCHAR2(20 BYTE)	No	<i>product status</i>
PROD_LIST_PRICE	NUMBER(8,2)	No	<i>product list price</i>
PROD_MIN_PRICE	NUMBER(8,2)	No	<i>product minimum price</i>
PROD_TOTAL	VARCHAR2(13 BYTE)	No	
PROD_TOTAL_ID	NUMBER	No	
PROD_SRC_ID	NUMBER	Yes	
PROD_EFF_FROM	DATE	Yes	
PROD_EFF_TO	DATE	Yes	
PROD_VALID	VARCHAR2(1 BYTE)	Yes	

INDEX_OWNER	INDEX_NAME	INDEX_TYPE	PARTITIONED	JOIN_INDEX	COLUMNS
SH	PRODUCTS_PK	NORMAL	NO	NO	PROD_ID
SH	PRODUCTS_PROD_CAT_IX	NORMAL	NO	NO	PROD_CATEGORY
SH	PRODUCTS_PROD_SUBCAT_IX	NORMAL	NO	NO	PROD_SUBCATEGORY
SH	PRODUCTS_PROD_STATUS_BIX	BITMAP	NO	NO	PROD_STATUS

**Tabla 4. Dimensión Producto**



**DIM\_PROMOTIONS.** La dimensión Promociones se enfoca en la mercadotecnia que se le haya dado a la venta y contiene datos como la fecha de inicio y fin de la promoción según se muestra en la Tabla 5.

COLUMN_NAME	DATA_TYPE	NULLABLE	COMMENTS
PROMO_ID	NUMBER(6,0)	No	<i>primary key column</i>
PROMO_NAME	VARCHAR2(30 BYTE)	No	<i>promotion description</i>
PROMO_SUBCATEGORY	VARCHAR2(30 BYTE)	No	<i>enables to investigate promotion hierarchies</i>
PROMO_SUBCATEGORY_ID	NUMBER	No	
PROMO_CATEGORY	VARCHAR2(30 BYTE)	No	<i>promotion category</i>
PROMO_CATEGORY_ID	NUMBER	No	
PROMO_COST	NUMBER(10,2)	No	<i>promotion cost, to do promotion effect calculations</i>
PROMO_BEGIN_DATE	DATE	No	<i>promotion begin day</i>
PROMO_END_DATE	DATE	No	<i>promotion end day</i>
PROMO_TOTAL	VARCHAR2(15 BYTE)	No	
PROMO_TOTAL_ID	NUMBER	No	

INDEX_OWNER	INDEX_NAME	UNIQUENESS	INDEX_TYPE	TEMPORARY	JOIN_INDEX	COLUMNS
SH	PROMO_PK	UNIQUE	NORMAL	N	NO	PROMO_ID

**Tabla 5. Dimensión Promoción**

**DIM\_TIMES.** Se puede decir que la dimensión tiempo es la más importante ya que contiene la línea del tiempo dependiendo la granularidad que se necesite (véase la Tabla 6): año, semestre, trimestre, mes hasta llegar al día.

COLUMN_NAME	DATA_TYPE	NULLABLE	COMMENTS
TIME_ID	DATE	No	<i>primary key; day date, finest granularity, CORRECT ORDER</i>
DAY_NAME	VARCHAR2(9 BYTE)	No	<i>Monday to Sunday, repeating</i>
DAY_NUMBER_IN_WEEK	NUMBER(1,0)	No	<i>1 to 7, repeating</i>
DAY_NUMBER_IN_MONTH	NUMBER(2,0)	No	<i>1 to 31, repeating</i>
CALENDAR_WEEK_NUMBER	NUMBER(2,0)	No	<i>1 to 53, repeating</i>
FISCAL_WEEK_NUMBER	NUMBER(2,0)	No	<i>1 to 53, repeating</i>

WEEK_ENDING_DAY	DATE	No	<i>date of last day in week, CORRECT ORDER</i>
WEEK_ENDING_DAY_ID	NUMBER	No	
CALENDAR_MONTH_NUMBER	NUMBER(2,0)	No	<i>1 to 12, repeating</i>
FISCAL_MONTH_NUMBER	NUMBER(2,0)	No	<i>1 to 12, repeating</i>
CALENDAR_MONTH_DESC	VARCHAR2(8 BYTE)	No	<i>e.g. 1998-01, CORRECT ORDER</i>
CALENDAR_MONTH_ID	NUMBER	No	
FISCAL_MONTH_DESC	VARCHAR2(8 BYTE)	No	<i>e.g. 1998-01, CORRECT ORDER</i>
FISCAL_MONTH_ID	NUMBER	No	
DAYS_IN_CAL_MONTH	NUMBER	No	<i>e.g. 28,31, repeating</i>
DAYS_IN_FIS_MONTH	NUMBER	No	<i>e.g. 25,32, repeating</i>
END_OF_CAL_MONTH	DATE	No	<i>last day of calendar month</i>
END_OF_FIS_MONTH	DATE	No	<i>last day of fiscal month</i>
CALENDAR_MONTH_NAME	VARCHAR2(9 BYTE)	No	<i>January to December, repeating</i>
FISCAL_MONTH_NAME	VARCHAR2(9 BYTE)	No	<i>January to December, repeating</i>
CALENDAR_QUARTER_DESC	CHAR(7 BYTE)	No	<i>e.g. 1998-Q1, CORRECT ORDER</i>
CALENDAR_QUARTER_ID	NUMBER	No	
FISCAL_QUARTER_DESC	CHAR(7 BYTE)	No	<i>e.g. 1999-Q3, CORRECT ORDER</i>
FISCAL_QUARTER_ID	NUMBER	No	
DAYS_IN_CAL_QUARTER	NUMBER	No	<i>e.g. 88,90, repeating</i>
DAYS_IN_FIS_QUARTER	NUMBER	No	<i>e.g. 88,90, repeating</i>
END_OF_CAL_QUARTER	DATE	No	<i>last day of calendar quarter</i>
END_OF_FIS_QUARTER	DATE	No	<i>last day of fiscal quarter</i>
CALENDAR_QUARTER_NUMBER	NUMBER(1,0)	No	<i>1 to 4, repeating</i>
FISCAL_QUARTER_NUMBER	NUMBER(1,0)	No	<i>1 to 4, repeating</i>
CALENDAR_YEAR	NUMBER(4,0)	No	<i>e.g. 1999, CORRECT ORDER</i>
CALENDAR_YEAR_ID	NUMBER	No	
FISCAL_YEAR	NUMBER(4,0)	No	<i>e.g. 1999, CORRECT ORDER</i>
FISCAL_YEAR_ID	NUMBER	No	
DAYS_IN_CAL_YEAR	NUMBER	No	<i>365,366 repeating</i>
DAYS_IN_FIS_YEAR	NUMBER	No	<i>e.g. 355,364, repeating</i>
END_OF_CAL_YEAR	DATE	No	<i>last day of cal year</i>
END_OF_FIS_YEAR	DATE	No	<i>last day of fiscal year</i>

INDEX_OWNER	INDEX_NAME	UNIQUENESS	INDEX_TYPE	TEMPORARY	JOIN_INDEX	COLUMNS
SH	TIMES_PK	UNIQUE	NORMAL	N	NO	TIME_ID

Tabla 6. Dimensión Tiempo

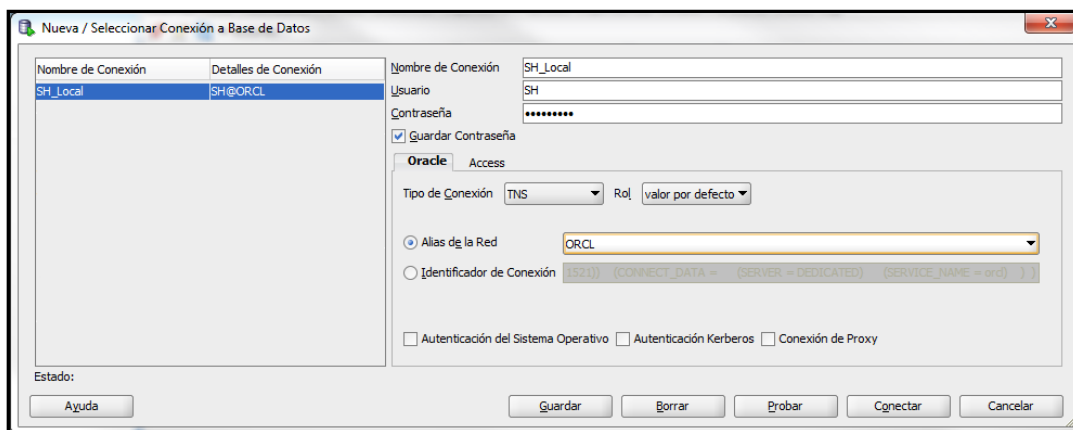
**FACT\_SALES.** La tabla de hechos Ventas (tabla 7) contiene la información más importante del modelo ya que esta proporciona las métricas que se relacionaran con todas las dimensiones, por ejemplo, la cantidad y el importe.

COLUMN_NAME	DATA_TYPE	NULLABLE	COMMENTS
PROD_ID	NUMBER	No	<i>FK to the products dimension table</i>
CUST_ID	NUMBER	No	<i>FK to the customers dimension table</i>
TIME_ID	DATE	No	<i>FK to the times dimension table</i>
CHANNEL_ID	NUMBER	No	<i>FK to the channels dimension table</i>
PROMO_ID	NUMBER	No	<i>promotion identifier, without FK constraint (intentionally) to show outer join optimization</i>
QUANTITY_SOLD	NUMBER(10,2)	No	<i>product quantity sold with the transaction</i>
AMOUNT_SOLD	NUMBER(10,2)	No	<i>invoiced amount to the customer</i>

INDEX_OWNER	INDEX_NAME	INDEX_TYPE	TEMPORARY	PARTITIONED	JOIN_INDEX	COLUMNS
SH	SALES_CUST_BIX	BITMAP	N	YES	NO	CUST_ID
SH	SALES_PROD_BIX	BITMAP	N	YES	NO	PROD_ID
SH	SALES_TIME_BIX	BITMAP	N	YES	NO	TIME_ID
SH	SALES_PROMO_BIX	BITMAP	N	YES	NO	PROMO_ID
SH	SALES_CHANNEL_BIX	BITMAP	N	YES	NO	CHANNEL_ID

**Tabla 7. Tabla de Hechos Ventas**

La conexión hacia el esquema SH se realiza por medio de *tnsname* (conexión nativa de Oracle) del cliente SQL *Developer* y requiere un nombre de usuario y su contraseña como se muestra en la Figura 26.



**Figura 26. Conexión a Base de Datos**

El archivo *tnsname.ora* contiene la cadena de conexión a la Base de Datos como se muestra en la Figura 27.

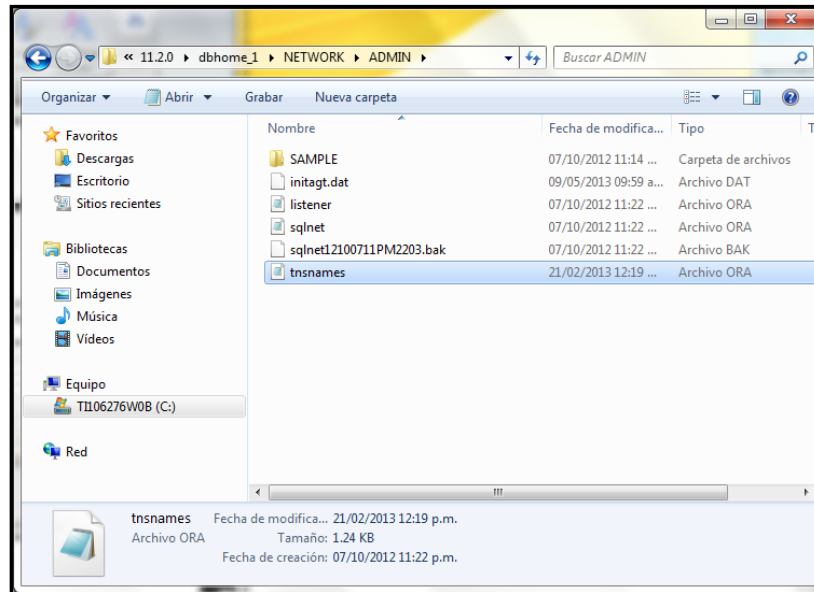


Figura 27. Archivo *Tnsnames*

Y su contenido es el siguiente:

```
ORCL =  
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = orcl)  
  )  
)
```

### 3.5 Diseño y Desarrollo de Presentación de Datos

En este caso no será necesario realizar procesos de ETL (*Extract, Transform, Load*) ya que el esquema SH presenta la información tal cual se necesita para poder ser explotada. Por lo tanto, el esquema SH se convierte en el *Data Warehouse*.

### 3.6 Diseño de la Arquitectura Técnica

En este caso se cuenta con un solo servidor local (*localhost*), ya que solo se cuenta con un solo equipo, en el cual se instalan la Base de Datos y el aplicativo OBIEE (*Oracle Business Intelligence*). La aplicación de usuario final puede ejecutarse desde cualquier otro equipo o *smartphone* dentro de la misma red como se muestra en la siguiente Figura 28.

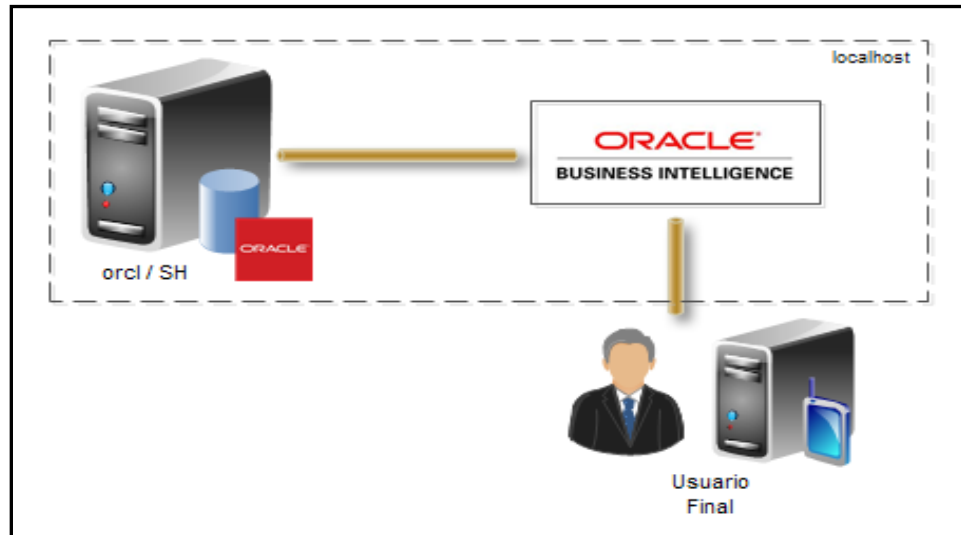


Figura 28. Arquitectura Técnica

### 3.7 Selección de Productos

Se utilizaran los siguientes productos:

- **Sistema Operativo:** *Windows 7 Professional x64*
- **Base de Datos:** *Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production*
- **Cliente de Base de Datos:** *Oracle SQL Developer 3.1.06.82*
- **Aplicación de BI:** *Oracle Business Intelligence 11.1.1.6.0*
- **Browser:** *Mozilla Firefox 16.0.2*

Los cuales cuentan con los siguientes términos de uso:

### **El uso del Software.**

*“La utilización del Software está sujeto a los acuerdos como un acuerdo de licencia o acuerdo de usuario que acompaña o se incluye con el software, los documentos pedidos, exposiciones, y otros términos y condiciones que se aplican ("Términos de Licencia"). En el caso de que el Software es proporcionado a través del Sitio y no tiene licencia para su uso en Términos de Licencia específicos para el Software, podrá utilizar el Software sujeto a lo siguiente: (a) el Software se puede utilizar exclusivamente para su uso personal, con fines informativos y no comerciales, (b) el Software no podrá ser modificado o alterado de ninguna manera, y (c) el Software no se puede redistribuir.”<sup>1</sup>*

### **3.8 Especificación de Aplicaciones para Usuarios Finales**

Como éste trabajo es con fines académicos y según los Términos de Uso de *Oracle*, solo existirá un tipo de usuario que se le denomina Administrador, el cual representa todos los roles desde usuario, desarrollo y usuario final.

El enfoque del análisis de información por parte del usuario va dirigido a visualizar las Ventas de la empresa desde las siguientes perspectivas:

- Cliente
- Canal de Venta
- Producto
- Promociones
- País del Cliente
- Tiempo

Se requiere presentar la información mediante un *Dashboard* interactivo con el cual se cumple el requerimiento funcional. El *Dashboard* debe de tener la propiedad de ser consultado por demanda.

---

<sup>1</sup> Términos de Uso Oracle.

### 3.9 Desarrollo de Aplicaciones para Usuarios Finales

Antes de comenzar el desarrollo en OBIEE se verifica que los servicios de la base de datos se encuentren iniciados para poder iniciar los servicios de OBIEE como se muestra en la Figura 29.

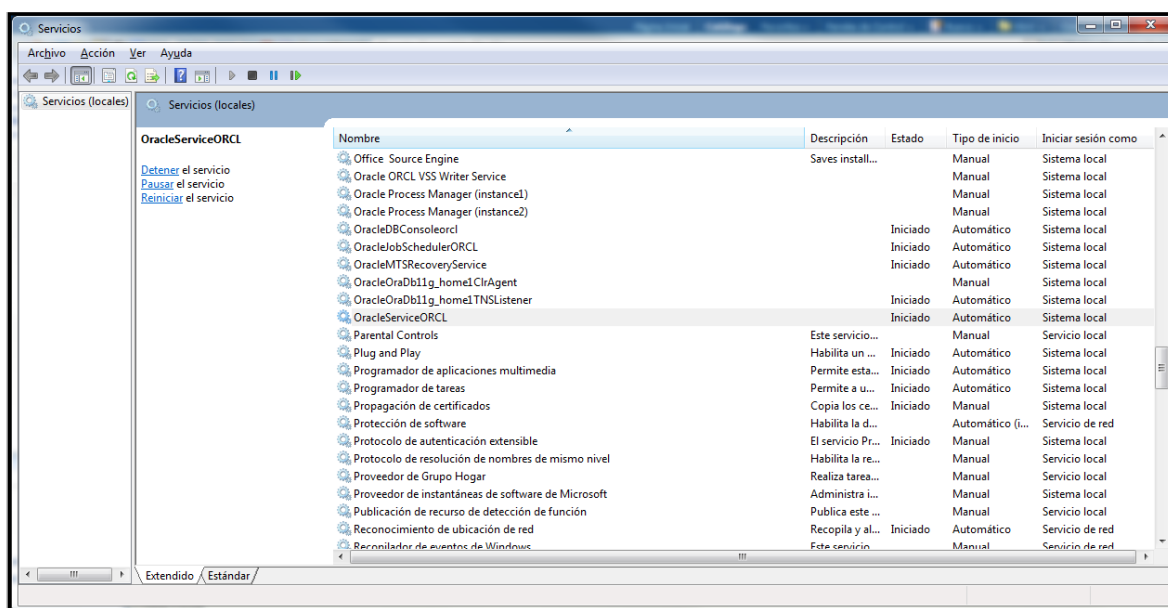


Figura 29. Estado de servicios de Base de Datos

Ahora se inician los servicios de OBIEE como se muestra en la Figura 30.

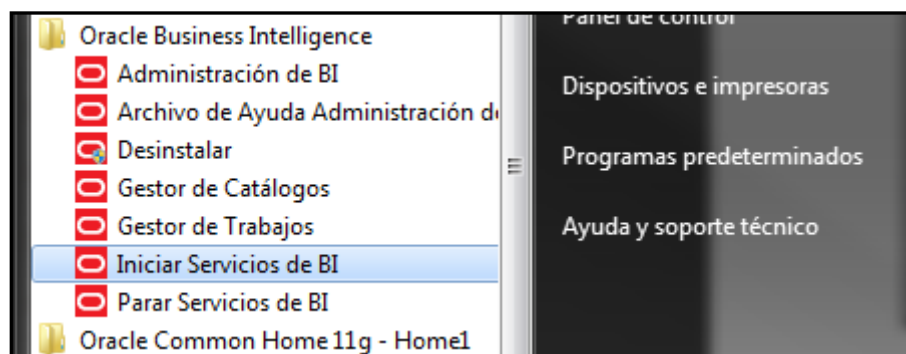
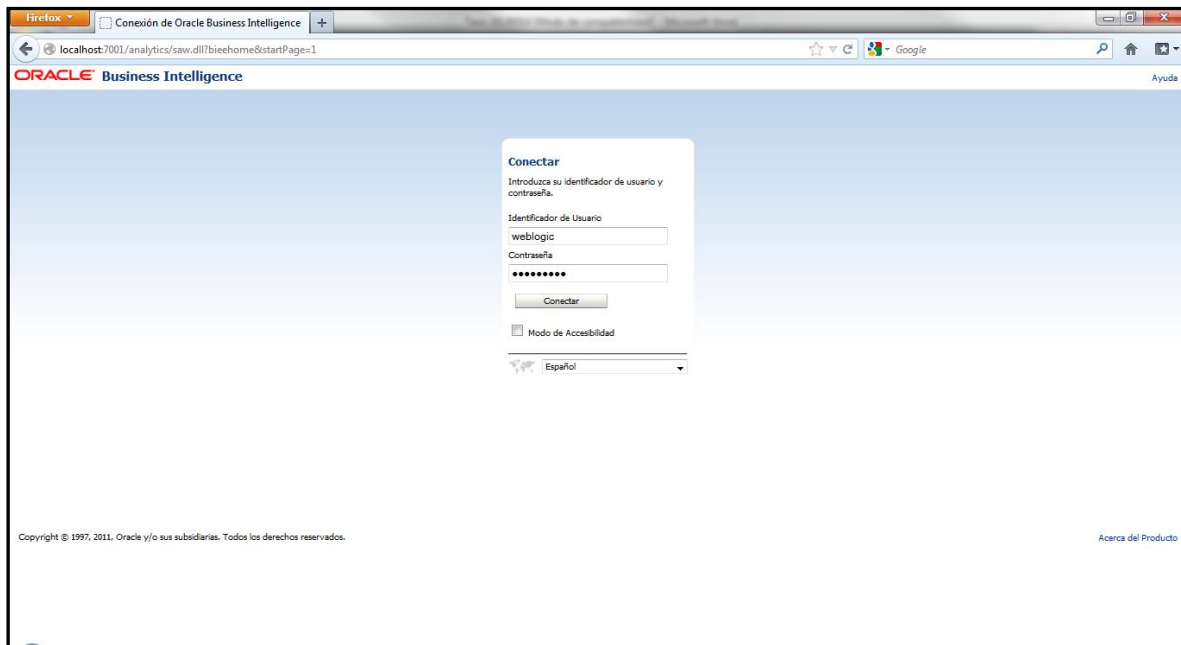


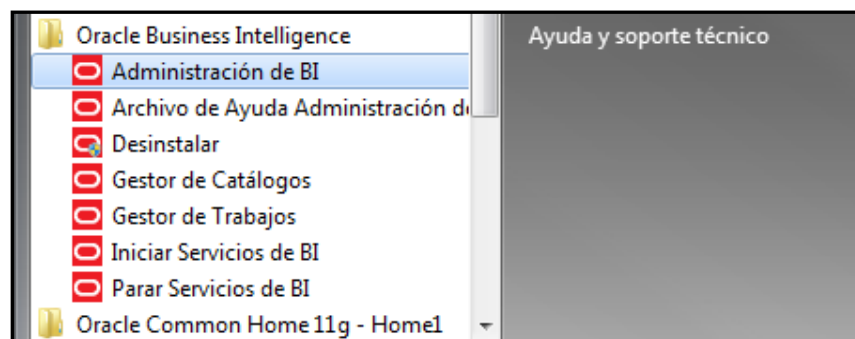
Figura 30. Inicio Servicios BI

Se ingresa a: <http://localhost:7001/analytics>, y se proporciona usuario y contraseña como se muestra en la Figura 31.



**Figura 31.** Pantalla de Ingreso de OBIEE

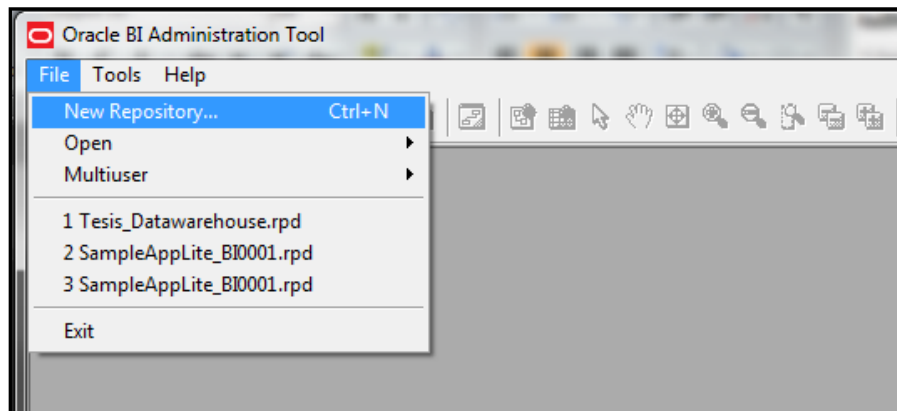
Se prosigue con la apertura de la consola de Administración de BI para la construcción del repositorio de OBIEE como se muestra en la Figura 32.



**Figura 32.** Administración de BI



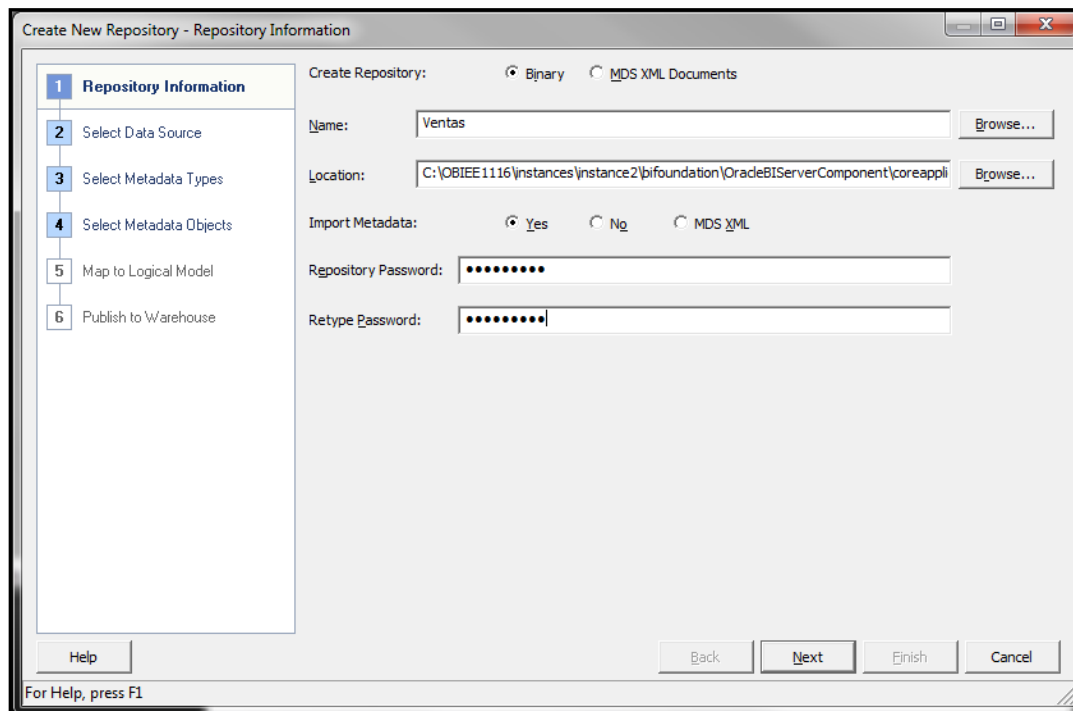
Al ingresar se da clic en *File>New Repository* como se muestra en la Figura 33.



**Figura 33.** Nuevo Repositorio

A continuación se crea el repositorio llamado Ventas de tipo binario ubicado en la ruta por omisión como se muestra en la Figura 34.

*C:\OBIEE1116\instances\instance2\bifoundation\OracleBIServerComponent\coreapplication\_obis1\repository\*



**Figura 34.** Creación de Repositorios

La fuente de información es el esquema SH (*Sales History*), se ingresan los datos de conexión como se muestra en la Figura 35.

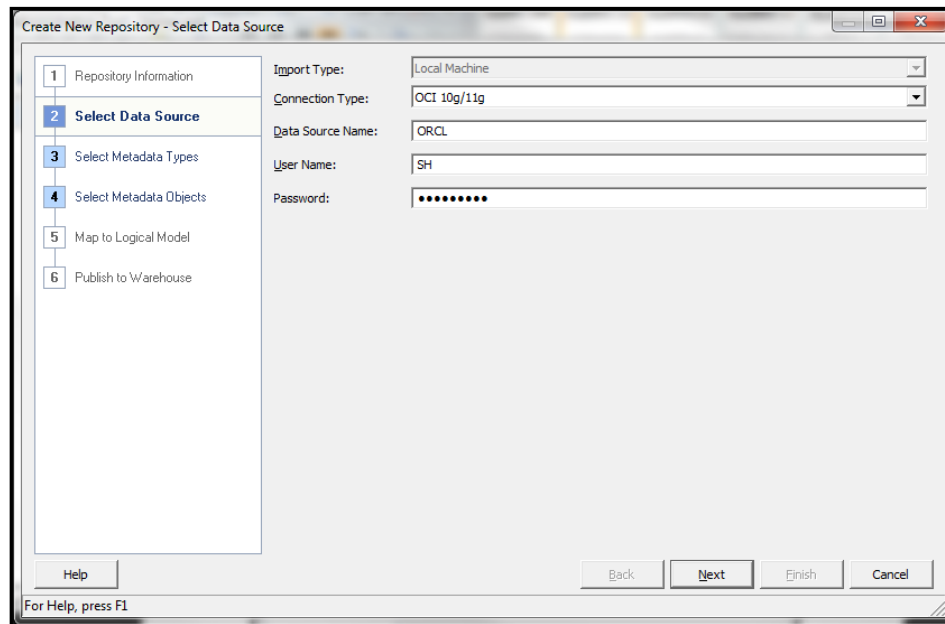


Figura 35. Fuente de Datos

Se seleccionan las dimensiones y hechos a importar como se muestra en la Figura 36.

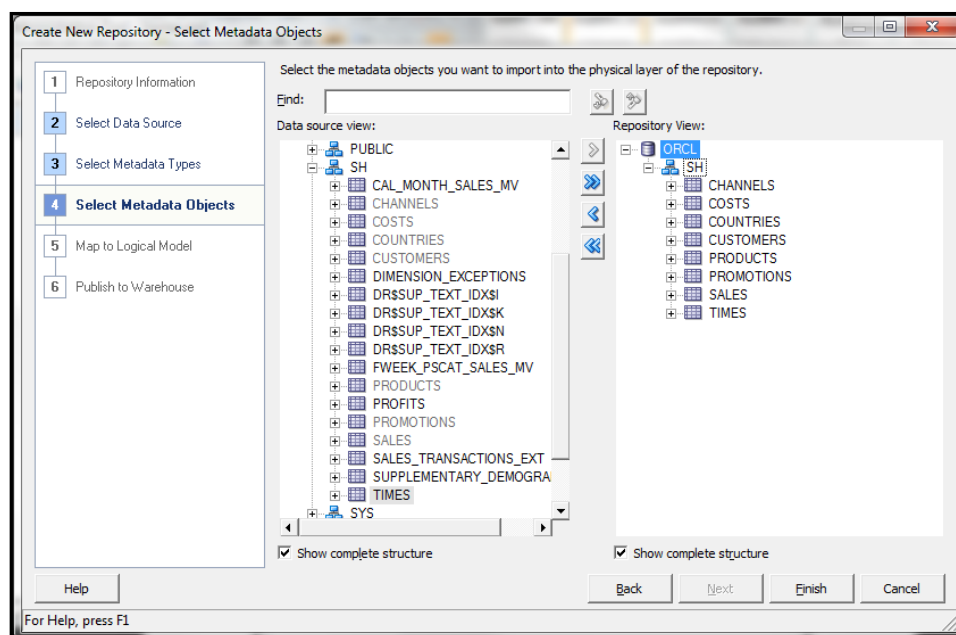


Figura 36. Selección de Objetos

Se verifica que cada una de las dimensiones y hechos importados se localicen en la capa física como se muestra en la Figura 37.

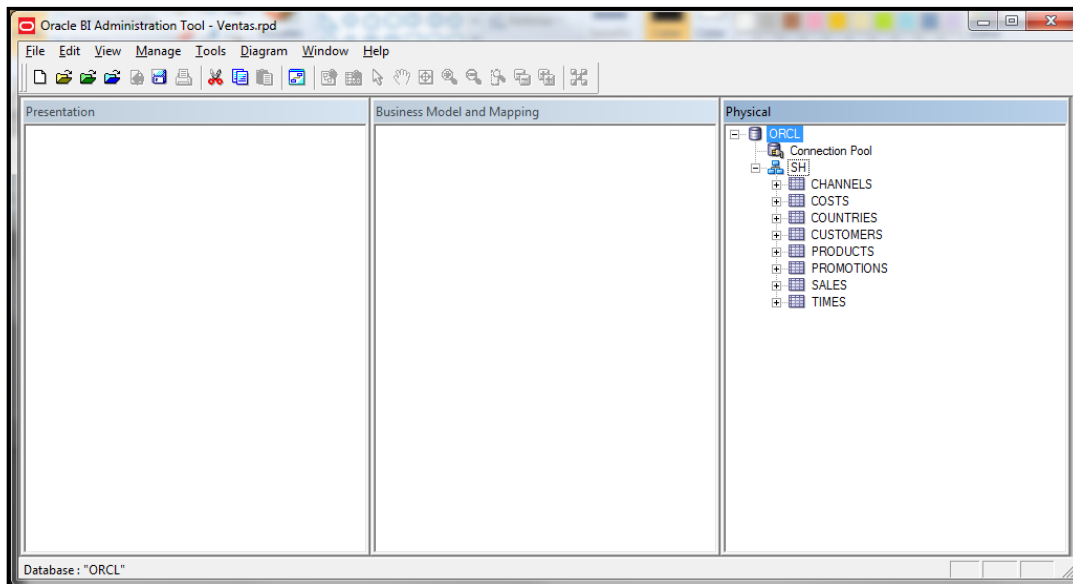


Figura 37. Importación a Capa Física

Se continua creando las relaciones entre tablas, se da clic derecho en *SH>Physical Diagram>Objects(s) an All Joins* como se muestra en la Figura 38.

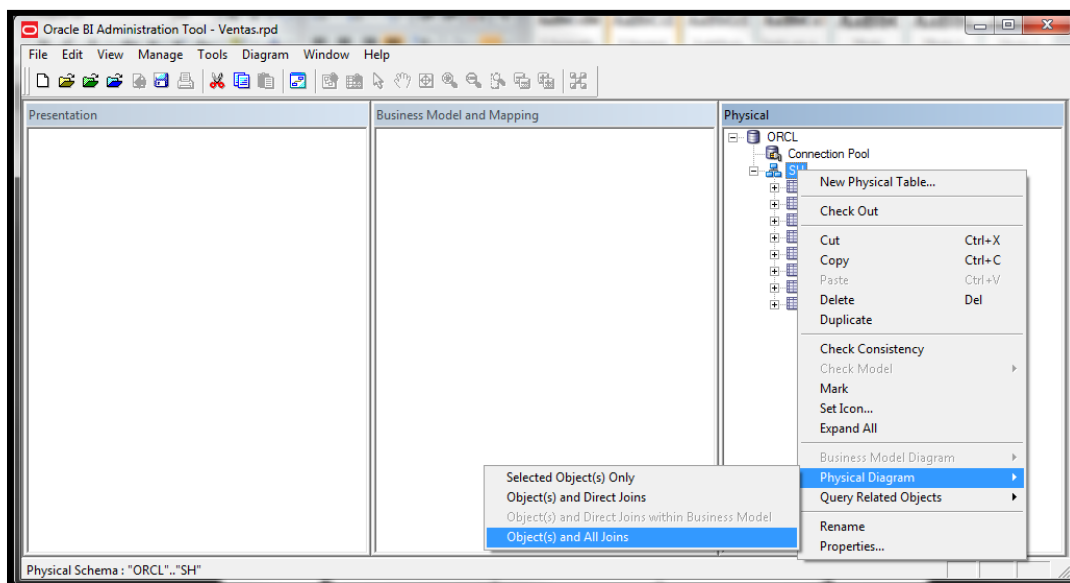
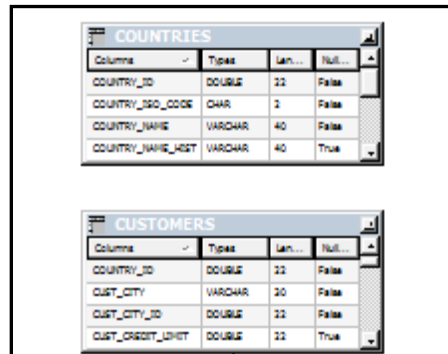


Figura 38. Esquema Diagrama Físico

Se realiza el *join* entre clientes (*CUSTOMERS*) y países del cliente (*COUNTRIES*) como se muestra en la Figura 39.



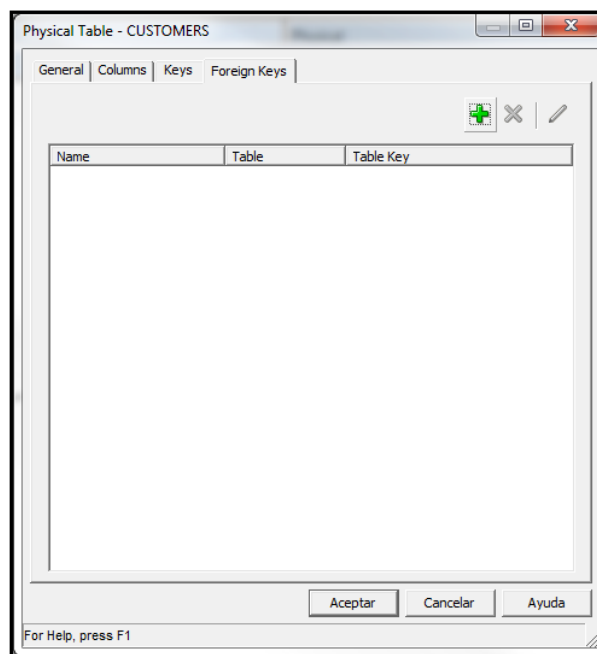
Columna	Tipos	Len.	Null.
COUNTRY_ID	DOUBLE	22	False
COUNTRY_ISO_CODE	CHAR	2	False
COUNTRY_NAME	VARCHAR	40	False
COUNTRY_NAME_HIST	VARCHAR	40	True


Columna	Tipos	Len.	Null.
COUNTRY_ID	DOUBLE	22	False
CUST_CITY	VARCHAR	20	False
CUST_CITY_ID	DOUBLE	22	False
CUST_CREDIT_LIMIT	DOUBLE	22	True

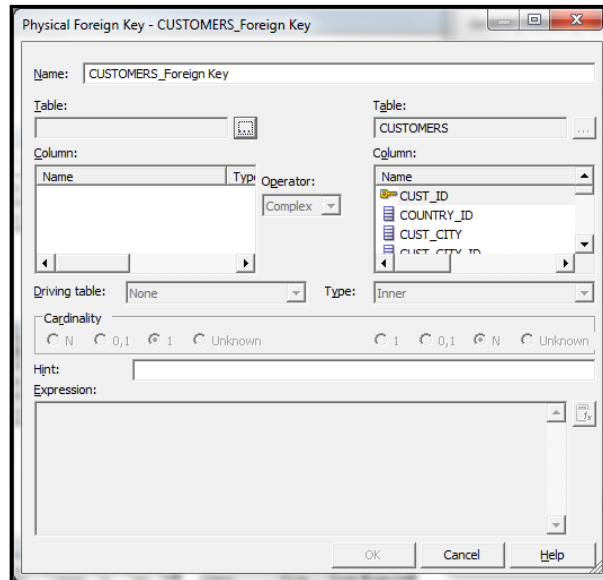
**Figura 39.** Relación entre Cliente y País de Cliente

Se da doble clic en *CUSTOMERS* y se agrega su llave foránea como se muestra en la Figura 40.



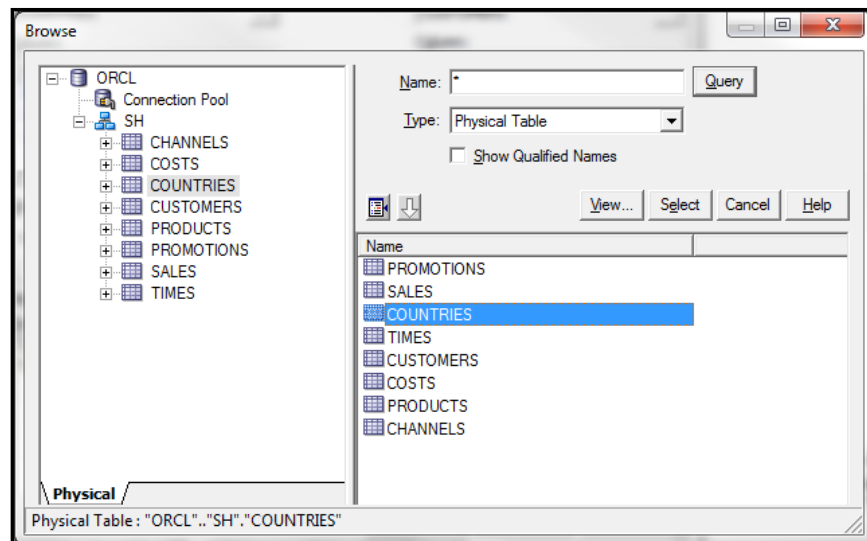
**Figura 40.** Llave Foránea

Se da clic en el botón  para seleccionar la dimensión *COUNTRIES* que contendrá la llave primaria como se muestra en la Figura 41.



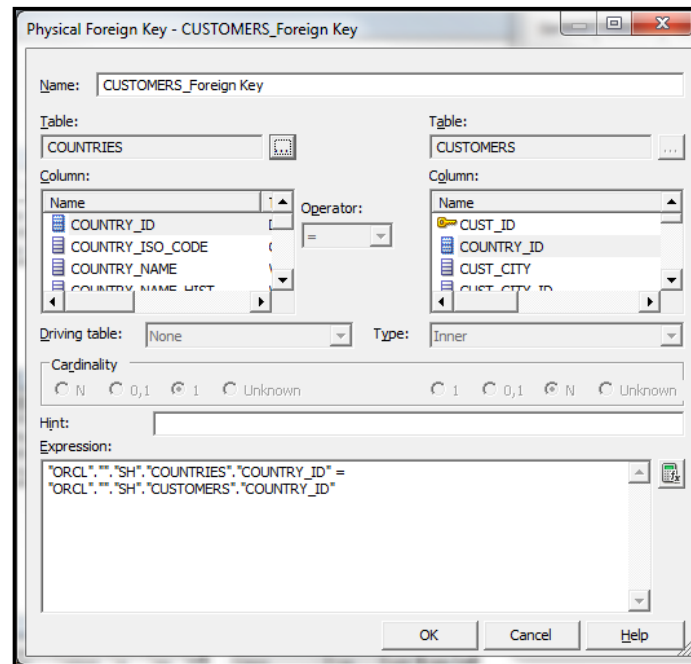
**Figura 41.** Join entre Dimensiones

Se selecciona *COUNTRIES* como se muestra en la Figura 42.



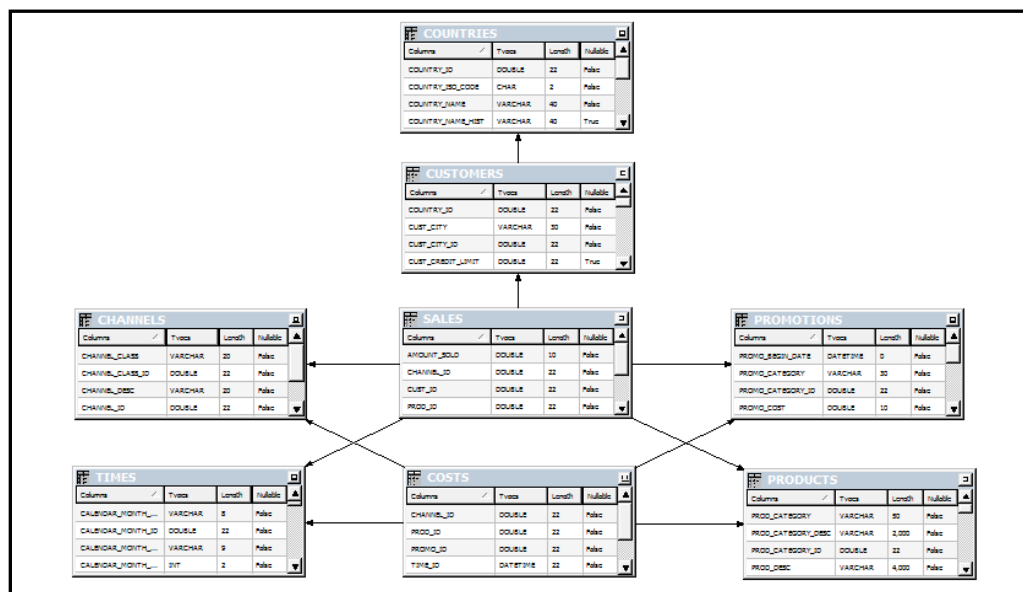
**Figura 42.** Selección de Dimensión

Se selecciona en ambas dimensiones el atributo con el cual se relacionan *COUNTRY\_ID*, y la expresión se forma en la parte inferior como se muestra en la Figura 43.



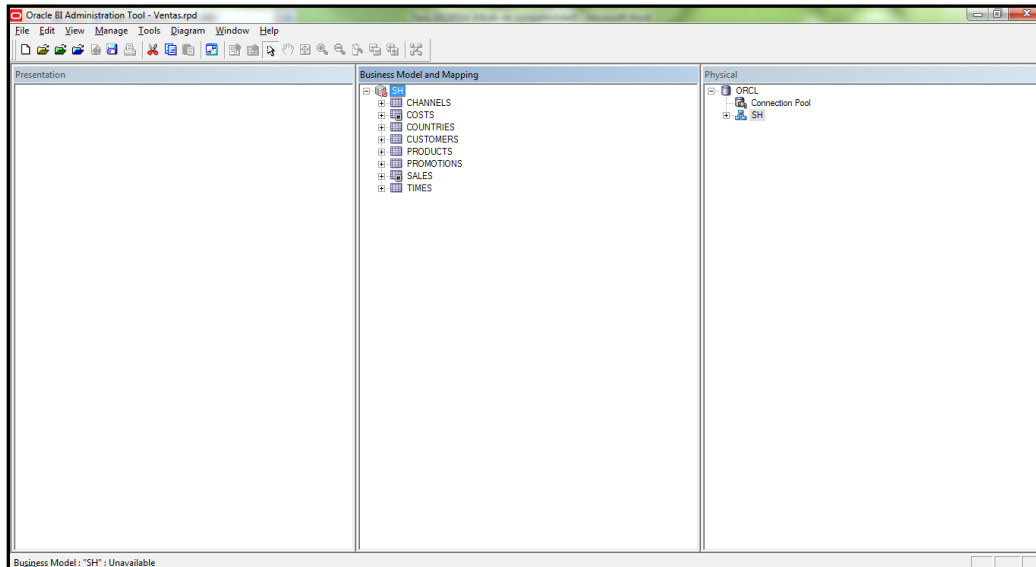
**Figura 43.** Relación entre Dimensiones

Se hace lo mismo para cada una de las dimensiones y hechos como se muestra en la Figura 44.



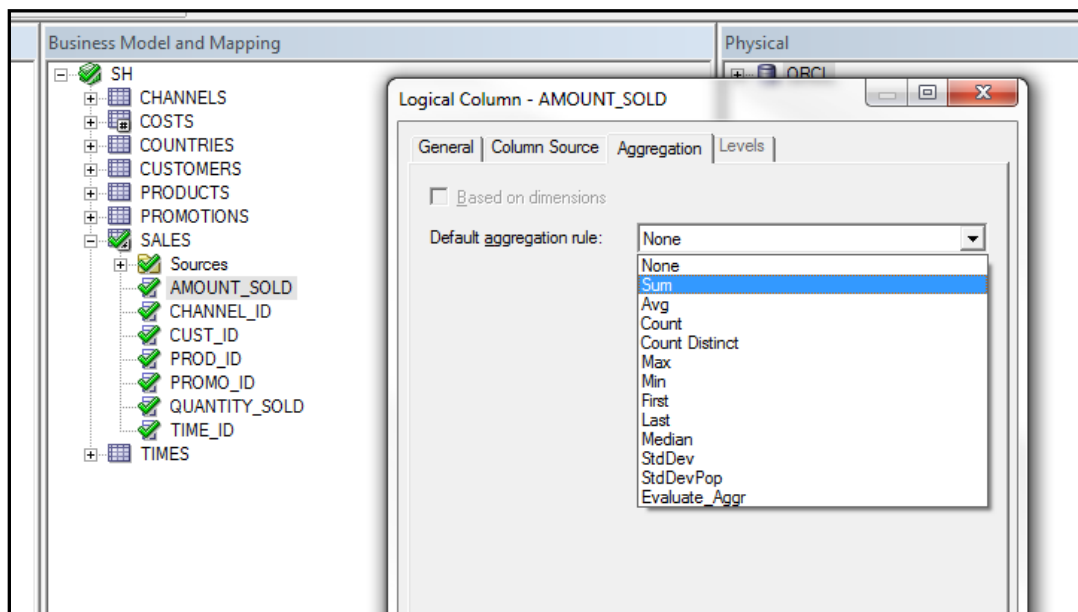
**Figura 44.** Esquema *Data Mart* de Ventas

Se continua arrastrando el esquema SH hacia la capa de negocio, se puede dar cuenta que de inmediato reconoció los hechos mediante el símbolo “#” como se muestra en la Figura 45, esto es porque las tablas de hechos solo contienen llaves foráneas.



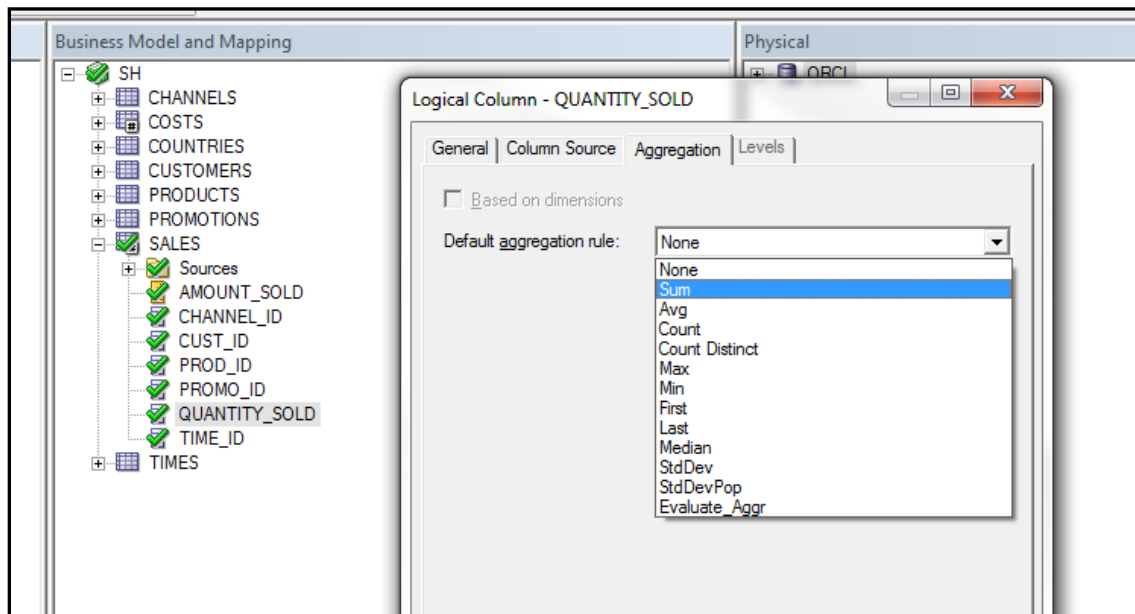
**Figura 45.** Construcción Capa de Negocio

Se selecciona la función de agregación en la cual aplicaran los hechos, en este caso para el importe vendido se selecciona la suma, como se muestra en la Figura 46.



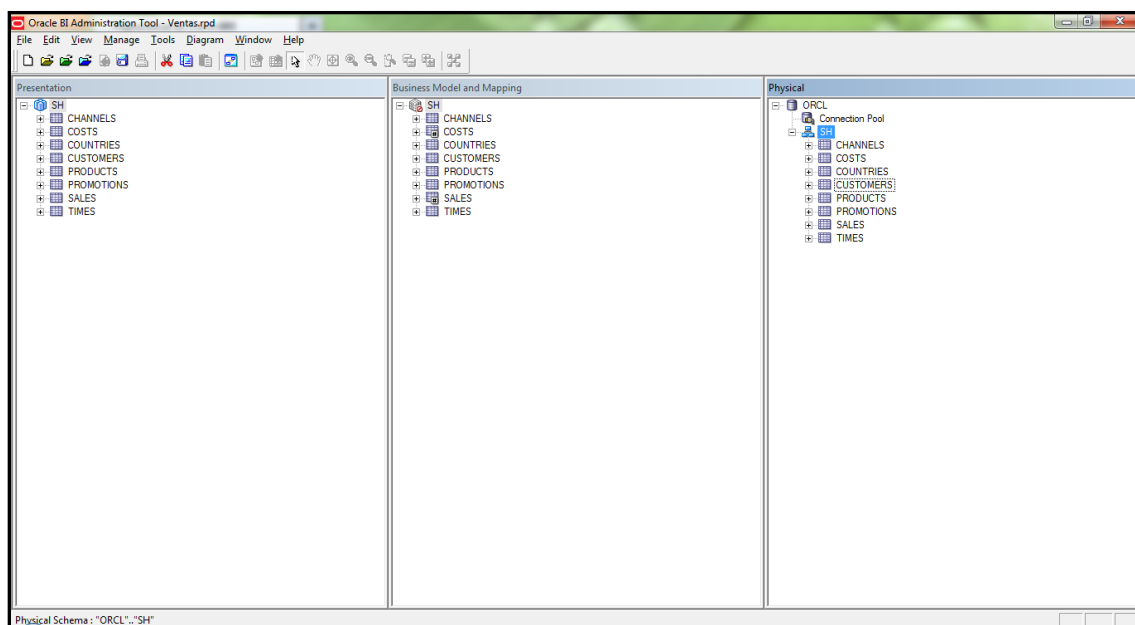
**Figura 46.** Definición de Métrica Importe Vendido

De igual forma para la cantidad vendida la suma como se muestra en la Figura 47.



**Figura 47.** Definición de Métrica Cantidad Vendida

Se prosigue a arrastrar el esquema SH a la capa de presentación como se muestra en la Figura 49.



**Figura 48.** Capa de Presentación



Se ingresa *Enterprise Manager* <http://localhost:7001/em>, para montar el repositorio creado, como se muestra en la Figura 49.

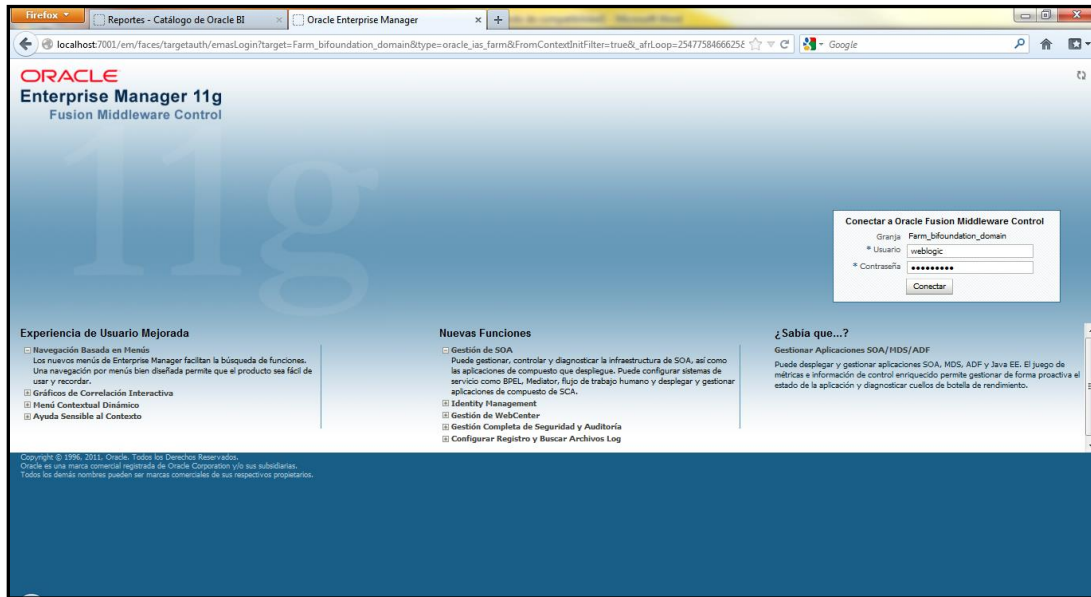


Figura 49. Enterprise Manager

Se presiona el botón (Actualizar...) y se selecciona el repositorio creado *Ventas.rpd* con su respectiva contraseña, se presiona aplicar y activar cambios, como se muestra en la Figura 50.

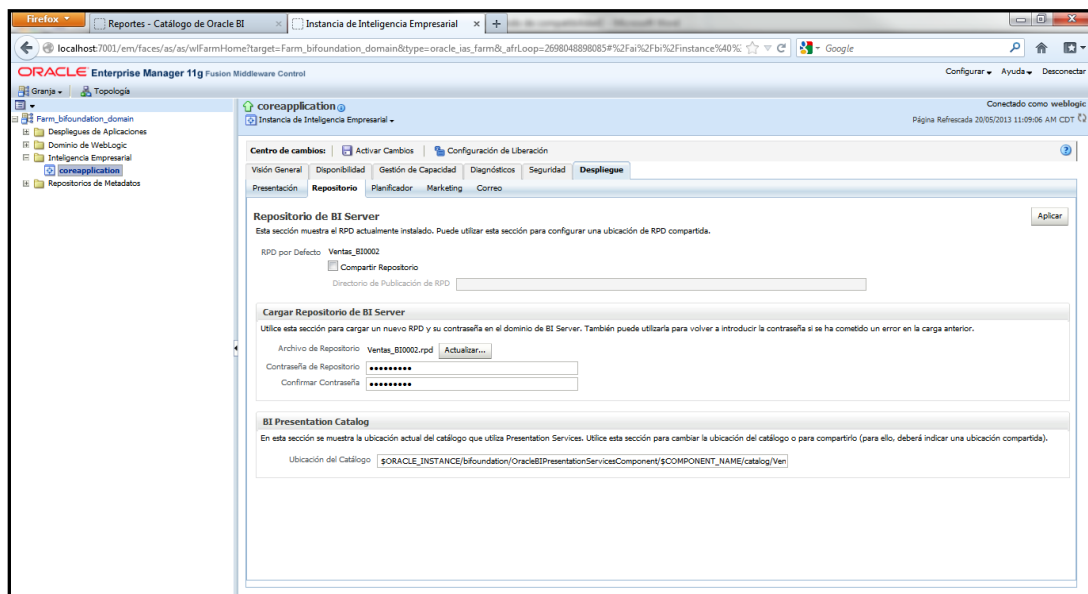


Figura 50. Actualización de Repositorio

## CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Se verifica que los servicios se encuentren disponibles, como se muestra en la Figura 51.

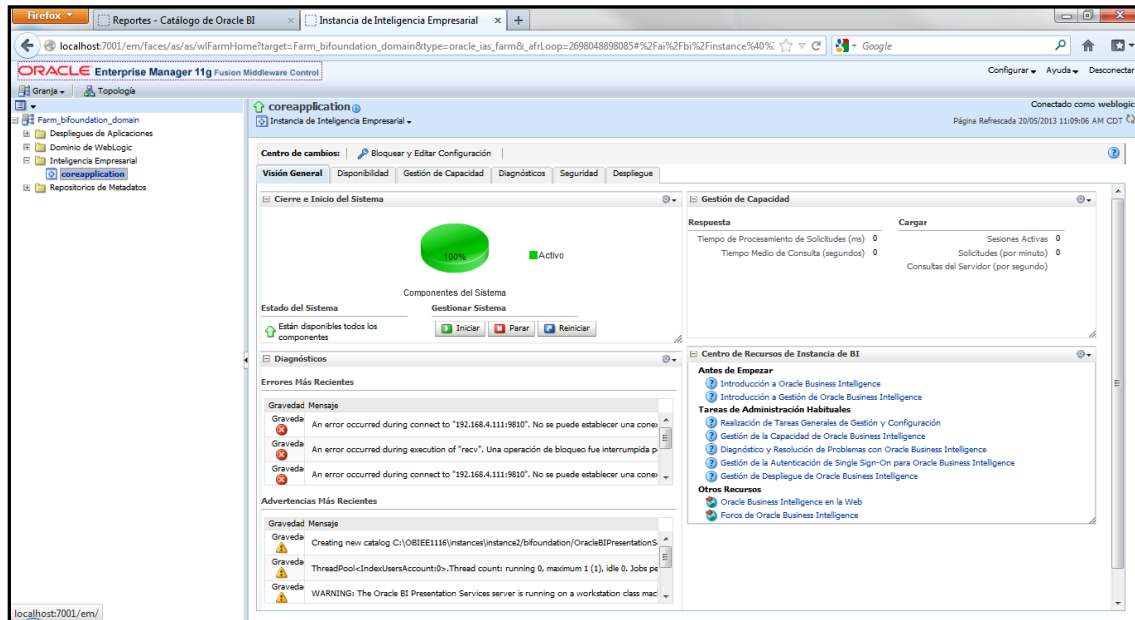


Figura 51. Servicios Iniciados

Se ingresa a la consola de *Analytics*, como se muestra en la Figura 52.

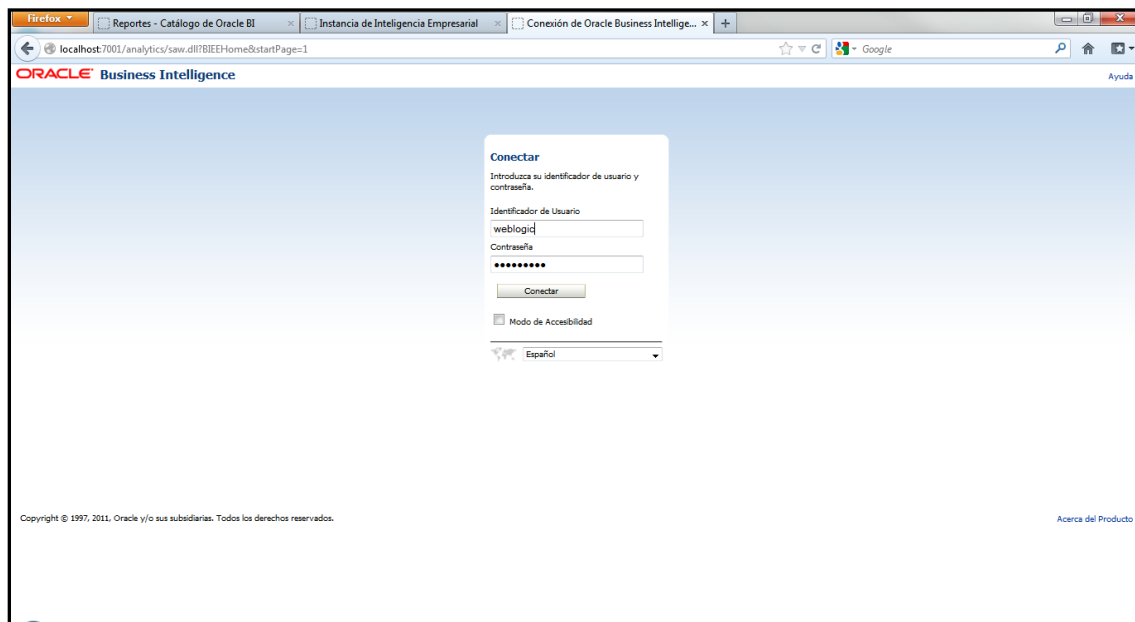


Figura 52. Pantalla de Inicio *Analytics*

## CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Se comprueba que el repositorio haya sido montado de manera exitosa, se selecciona **Nuevo>Análisis>SH**, como se muestra en la Figura 53.

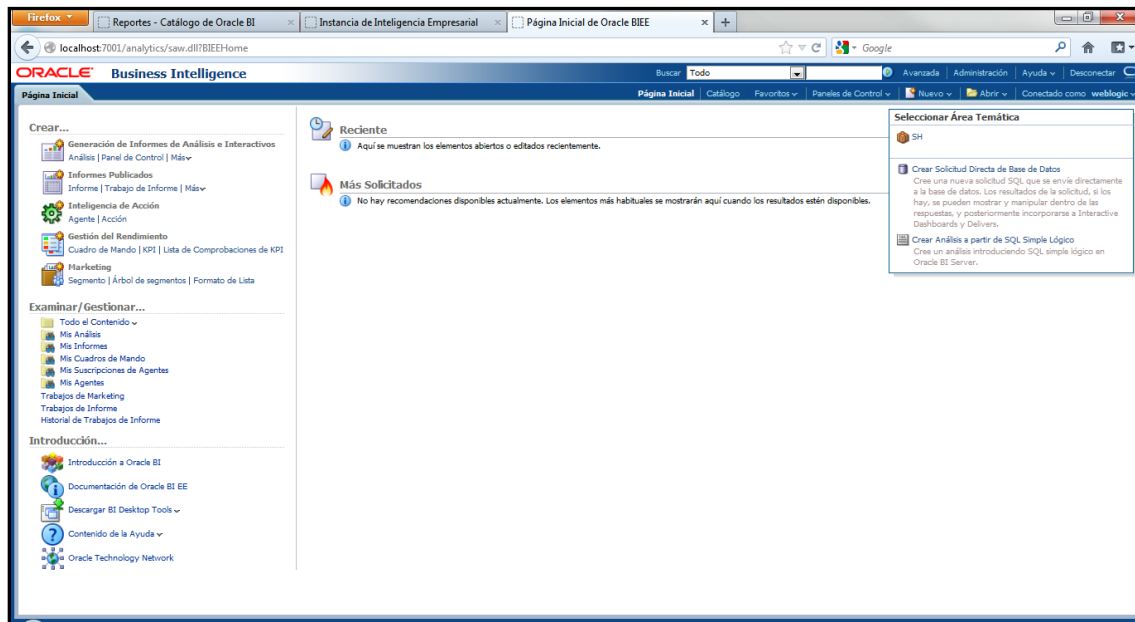


Figura 53. Selección de Área Temática

Se comprueba que la información esté disponible, como se muestra en la Figura 54.

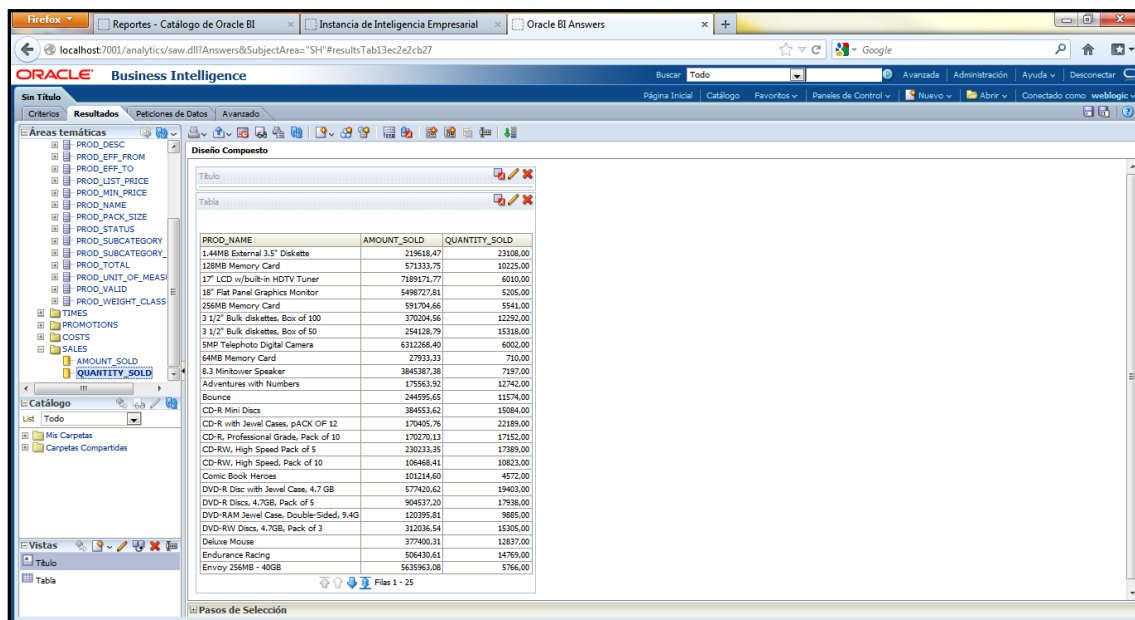


Figura 54. Comprobación de Datos

## CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Se realiza un nuevo análisis sobre el área temática SH, el cual permitirá conocer el detalle de las ventas seleccionando los atributos necesarios para el análisis, como se muestra en la Figura 55.

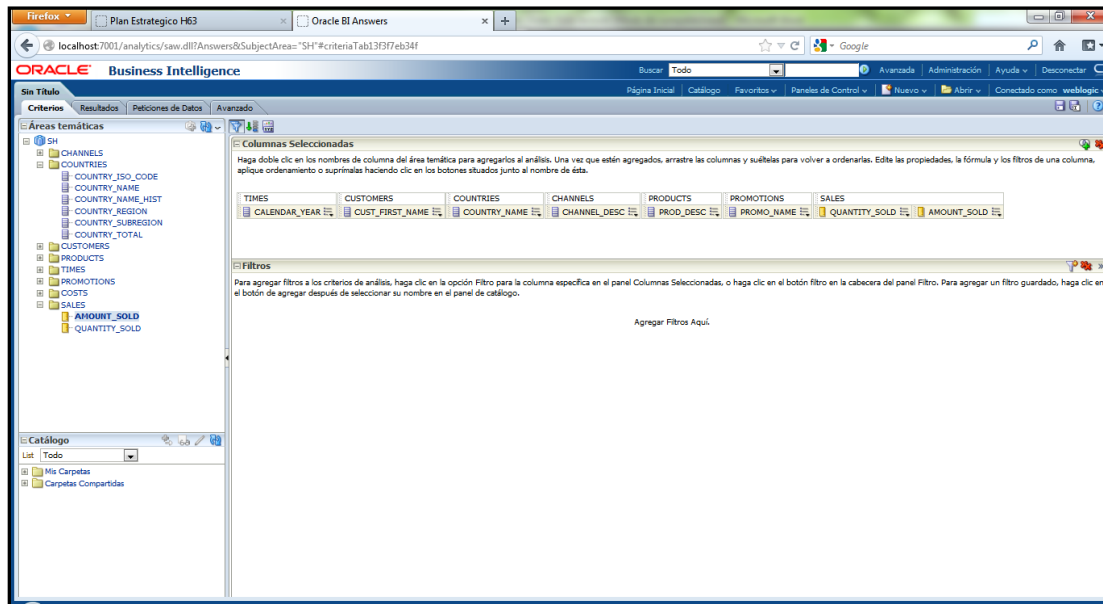


Figura 55. Selección de Atributos

Dando formato a los nombres el resultado se muestra en la Figura 56.

The screenshot shows the Oracle Business Intelligence web interface with the 'Diseño Compuesto' (Composite Design) view. The table displays sales data for the year 1998, categorized by client (Abel, Partners, Italy) and sales channel (Direct Sales, Partners, Direct Sales). The table is formatted with bold headers and includes a footer indicating 'Filas 1 - 25' (Rows 1 - 25).

AÑO	CLIENTE	PAIS DEL CLIENTE	CANAL DE VENTA	PRODUCTO	PROMOCION	CANTIDAD VENDIDA	IMPORTE DE VENTA
1998	Abel	Australia	Direct Sales	1.44MB External 3.5" Diskette	NO PROMOTION #	2,00	\$19,56
				Envoy External 6X CD-ROM	NO PROMOTION #	2,00	\$103,48
				Envoy External 6X CD-ROM	NO PROMOTION #	2,00	\$125,60
				External 6X CD-ROM	NO PROMOTION #	2,00	\$91,98
				External 6X CD-ROM	NO PROMOTION #	2,00	\$114,38
				Internet promotion #29-350	Internet promotion #29-350	1,00	\$57,79
				Internal 6X CD-ROM	NO PROMOTION #	1,00	\$34,19
				Internal 6X CD-ROM	NO PROMOTION #	2,00	\$80,48
				PCMCIA modem/fax 28800 baud	NO PROMOTION #	5,00	\$244,75
				Standard Mouse	Internet promotion #29-350	1,00	\$46,83
				Standard Mouse	NO PROMOTION #	1,00	\$25,18
			Partners	1.44MB External 3.5" Diskette	NO PROMOTION #	2,00	\$19,56
				Internet promotion #29-350	Internet promotion #29-350	1,00	\$9,71
				Envoy External 6X CD-ROM	NO PROMOTION #	2,00	\$126,48
				External 6X CD-ROM	NO PROMOTION #	1,00	\$45,59
				External 6X CD-ROM	NO PROMOTION #	1,00	\$45,59
				External 6X CD-ROM	NO PROMOTION #	1,00	\$56,99
				Internet promotion #29-350	Internet promotion #29-350	1,00	\$57,79
				Internal 6X CD-ROM	NO PROMOTION #	2,00	\$80,34
				PCMCIA modem/fax 28800 baud	NO PROMOTION #	2,00	\$97,32
				Standard Mouse	Internet promotion #29-350	1,00	\$46,66
				Standard Mouse	NO PROMOTION #	1,00	\$24,92
			Italy	1.44MB External 3.5" Diskette	Internet promotion #29-350	1,00	\$9,71
				3 1/2" Bulk diskettes, Box of 100	NO PROMOTION #	2,00	\$60,72
				Internet promotion #29-350	Internet promotion #29-350	1,00	\$30,12

Figura 56. Resultado de Análisis

## CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Se continúa con el siguiente análisis el cual mostrara lo vendido por año, como se muestra en la Figura 57.



Figura 57. Análisis Importe Vendido Anualmente

Se muestra el resultado en la Figura 58.

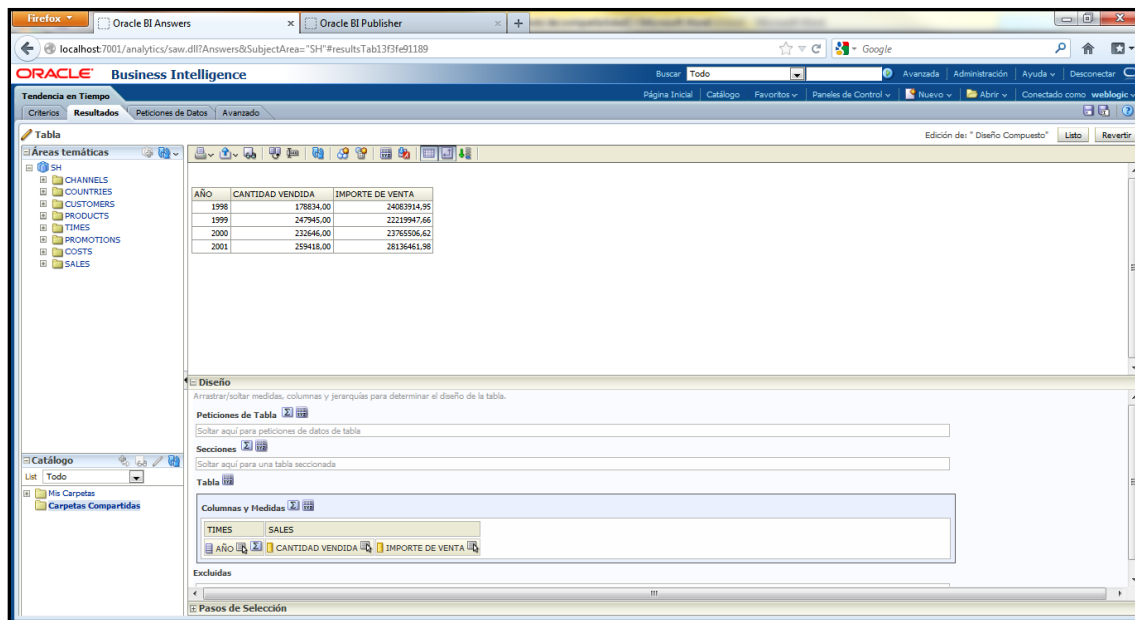


Figura 58. Resultado de Análisis Importe Vendido Anualmente



## CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Se selecciona el Atributo a usar como *prompt*, como se muestra en la Figura 61.

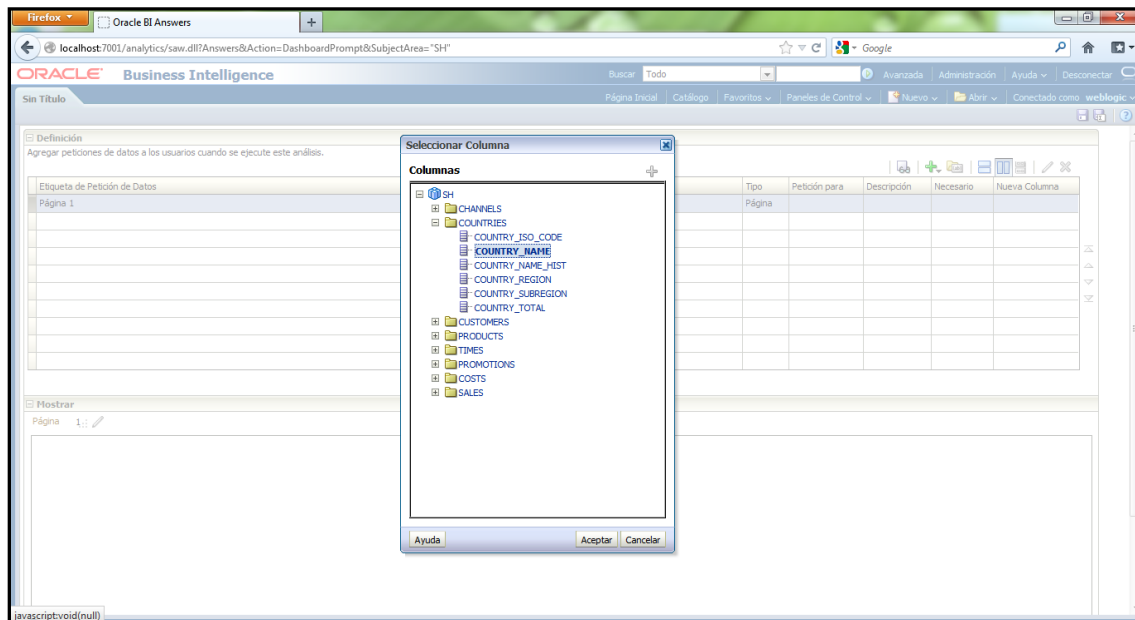


Figura 61. Selección de Atributo para *Prompt*

Se prueba el parámetro de entrada como se muestra en la Figura 62.

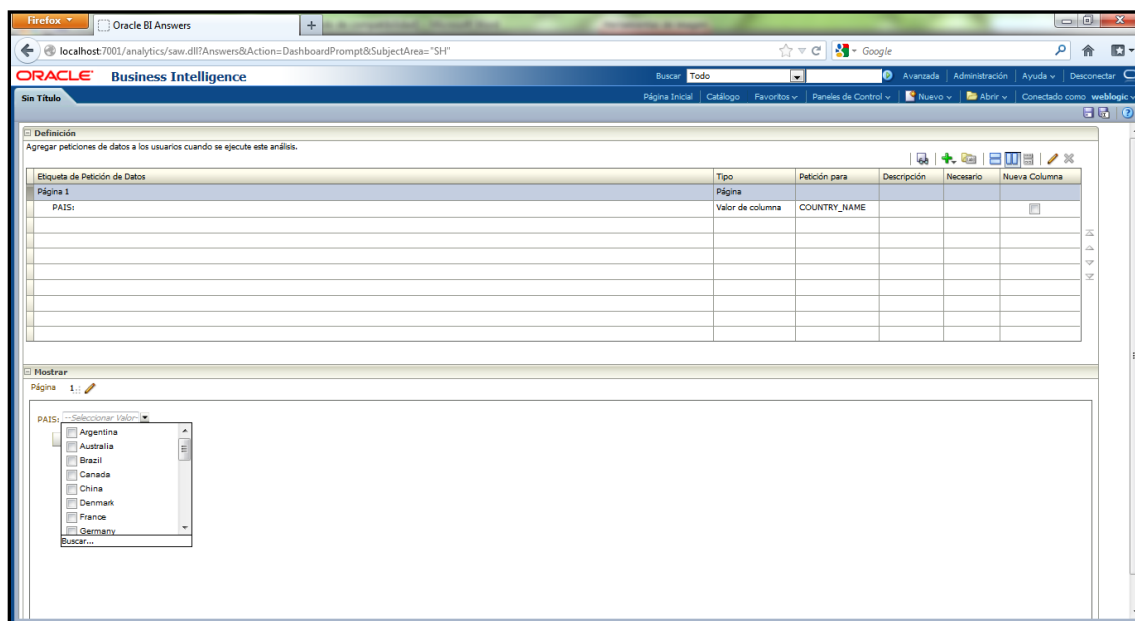


Figura 62. Prueba de *Prompt*

## CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Para cada uno de los *prompts* se añade un filtro de solicitud de datos, como se muestra en la Figura 63.

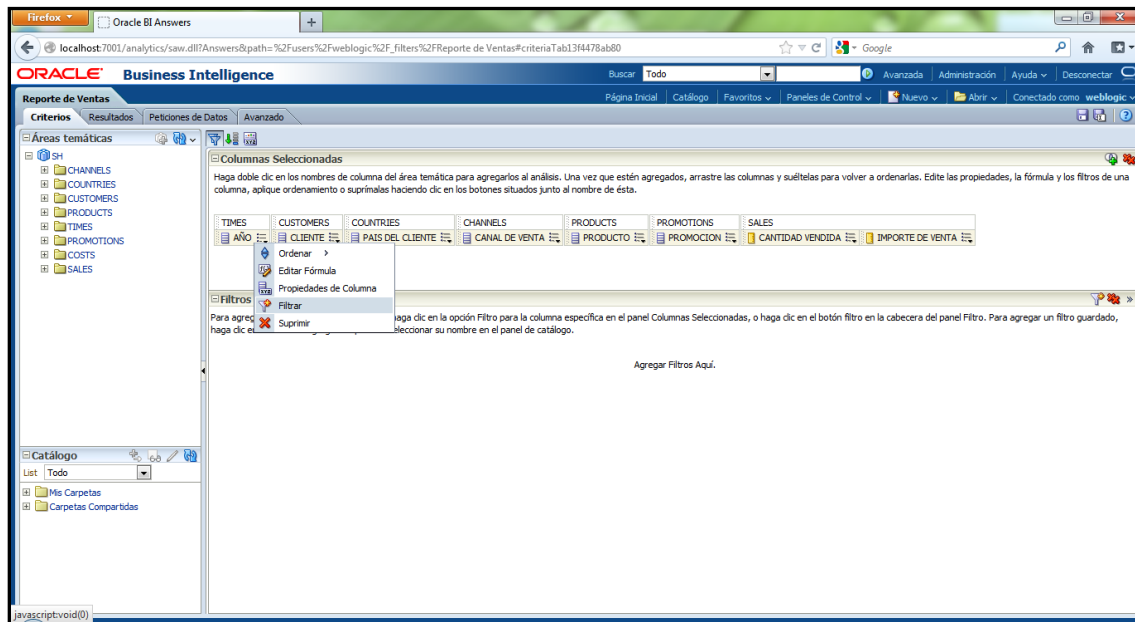


Figura 63. Agregación de Filtro

Se elige la opción *se solicita* como se muestra en la Figura 61.

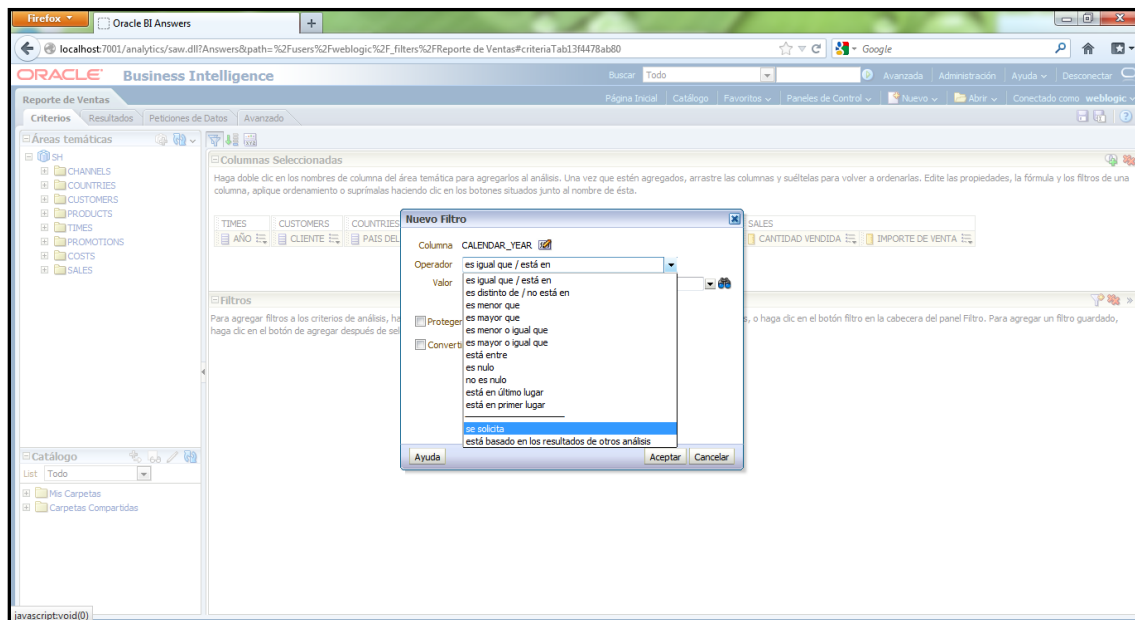


Figura 64. Filtro de Tipo “se solicita”



Por último se crea el análisis que proporcionara el *top* 10 de clientes como se muestra en la Figura 65.

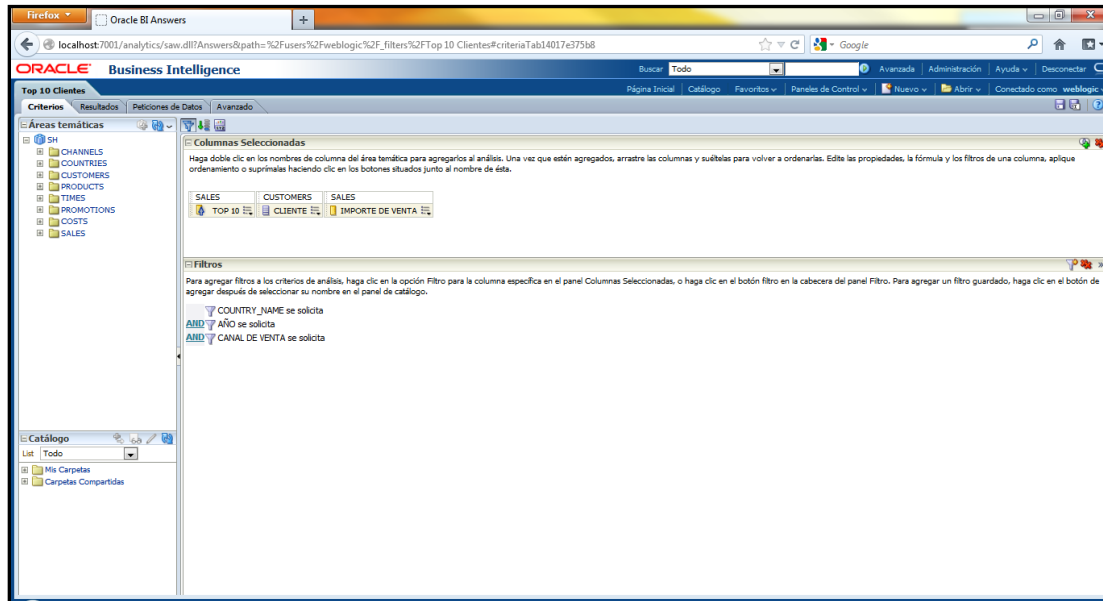


Figura 65. Análisis Top 10 Clientes

Se agrega una columna calculada a la cual se aplicara la formula *top*, la cual identifica los N valores más altos de un atributo numérico, *TOPN (expr, integer)* como se muestra en la Figura 66.

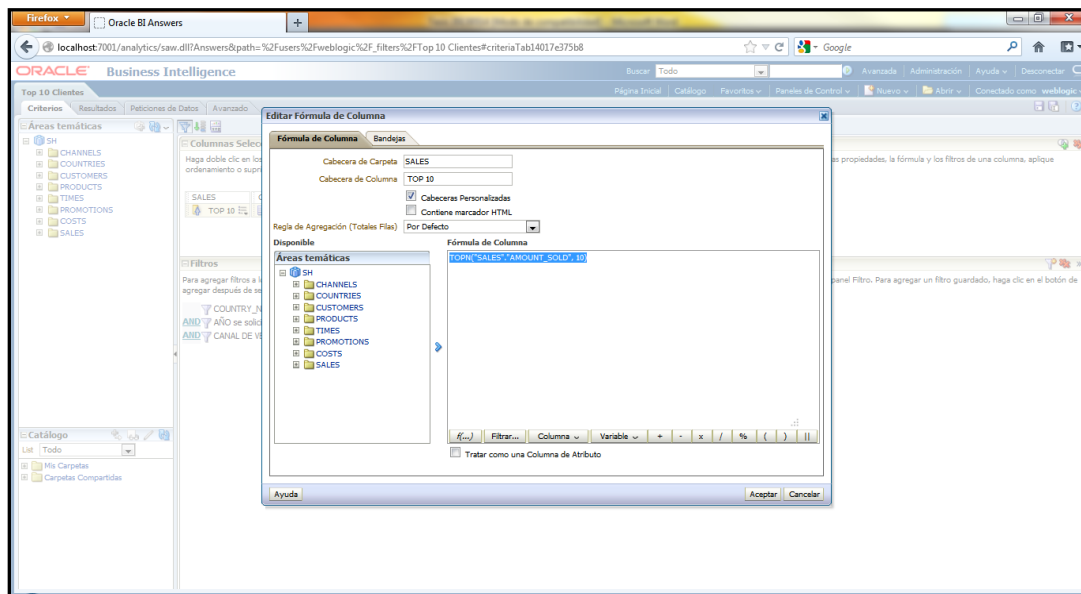
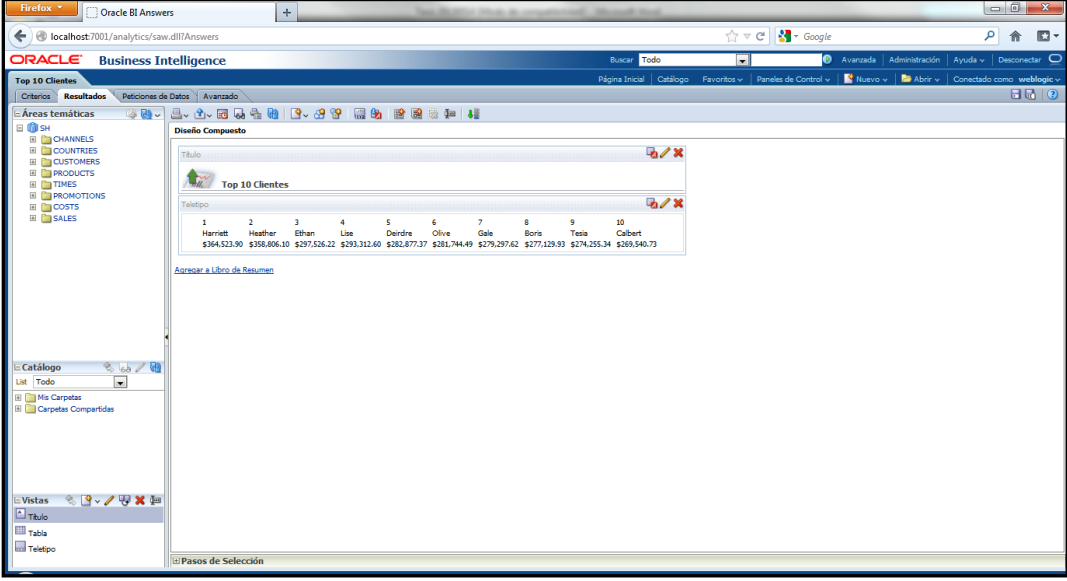


Figura 66. Agregación de Imagen

## CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Se muestra la salida, no como una tabla, sino como un teletipo como se muestra en la Figura 67.

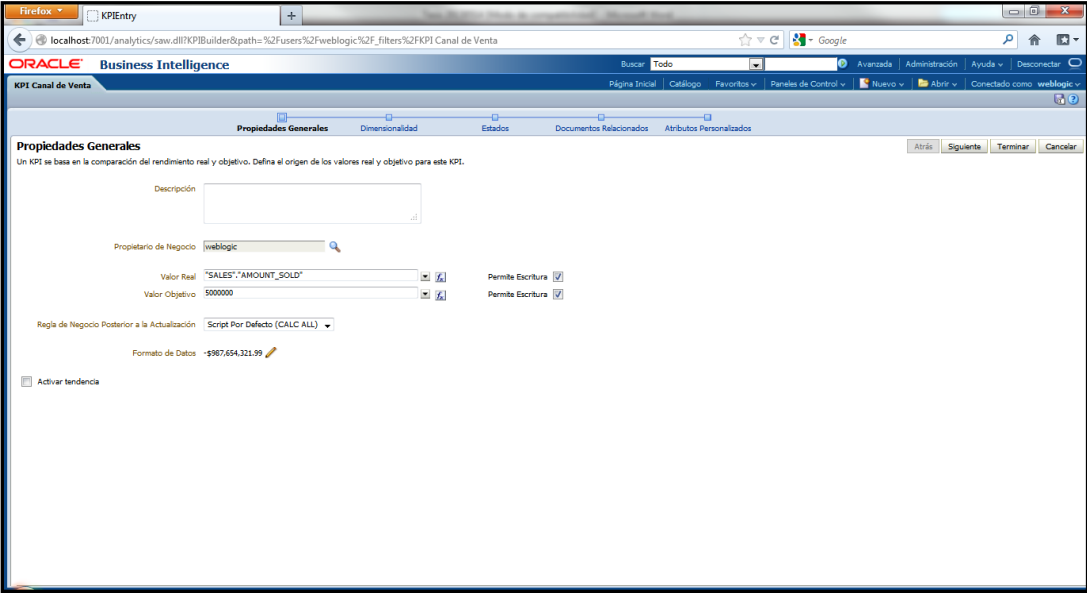


The screenshot shows the Oracle BI Answers interface. On the left, there's a navigation pane with 'Áreas temáticas' (Channels, Countries, Customers, Products, Times, Promotions, Costs, Sales) and 'Catálogo' (List, Mis Carpetas, Carpetas Compartidas). The main area displays a 'Diseño Composto' (Composite Design) with a 'Teletipo' (Teletype) report titled 'Top 10 Clientes'. The report shows a list of 10 clients with their names and sales amounts.

1	2	3	4	5	6	7	8	9	10
Harriet	Heather	Ethan	Lisa	Deirda	Olive	Gale	Boris	Tessa	Calbert
\$364,523.90	\$358,856.10	\$297,526.22	\$293,312.60	\$282,877.37	\$281,744.49	\$279,297.62	\$277,129.93	\$274,255.34	\$268,540.73

Figura 67. Resultado Top 10 Clientes Teletipo

Se crea un KPI (Key Performance Indicators), para las ventas con respecto al tiempo y al canal de venta. Tomando como valor objetivo la cantidad de 5,000,000.00 como se muestra en la Figura 68.



The screenshot shows the 'KPI Canal de Venta' configuration page in Oracle BI Answers. The 'Propiedades Generales' (General Properties) tab is active. The page includes fields for 'Descripción' (Description), 'Propietario de Negocio' (Business Owner), 'Valor Real' (Actual Value), 'Valor Objetivo' (Target Value), 'Regla de Negocio Posterior a la Actualización' (Business Rule After Update), and 'Formato de Datos' (Data Format). The 'Valor Objetivo' is set to 5000000.00. The 'Regla de Negocio' is set to 'Script Por Defecto (CALC ALL)'. The 'Formato de Datos' is set to '-\$987,654,321.99'.

Propiedades Generales

Un KPI se basa en la comparación del rendimiento real y objetivo. Defina el origen de los valores real y objetivo para este KPI.

Descripción

Propietario de Negocio: weblogic

Valor Real: "SALES"."AMOUNT\_SOLD"

Valor Objetivo: 5000000

Regla de Negocio Posterior a la Actualización: Script Por Defecto (CALC ALL)

Formato de Datos: -\$987,654,321.99

Activar tendencia

Figura 68. KPI Canal de Venta

## CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Se seleccionan las dimensiones a utilizar como se muestra en la Figura 69.

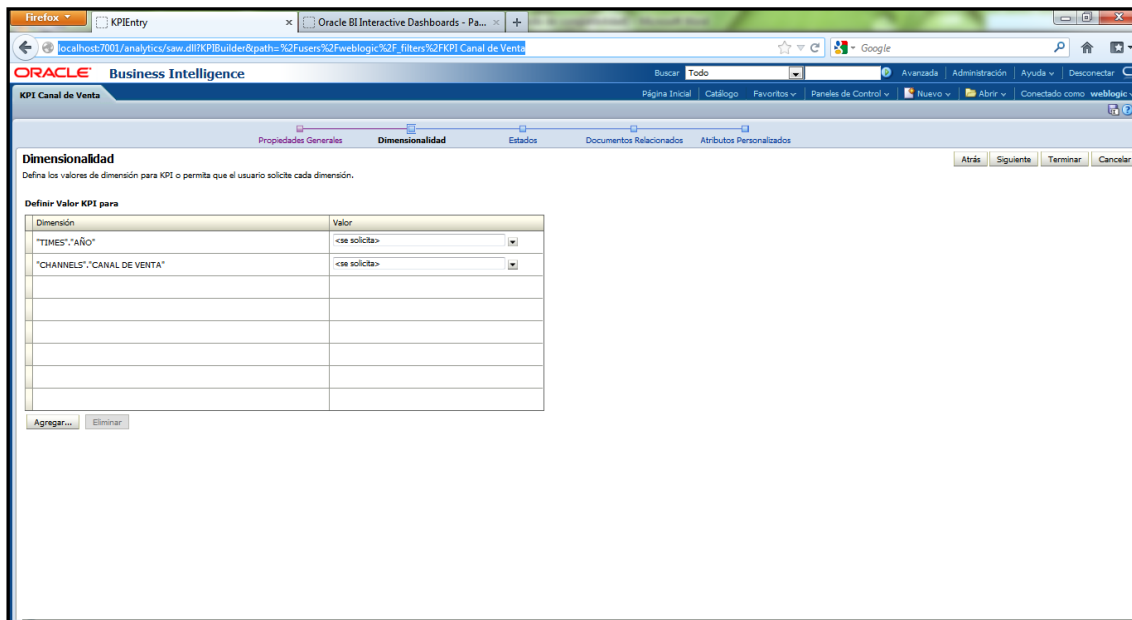


Figura 69. "Dimensionalidad" KPI

Se determinan los rangos para cada uno de los estados como se muestra en la Figura 70.

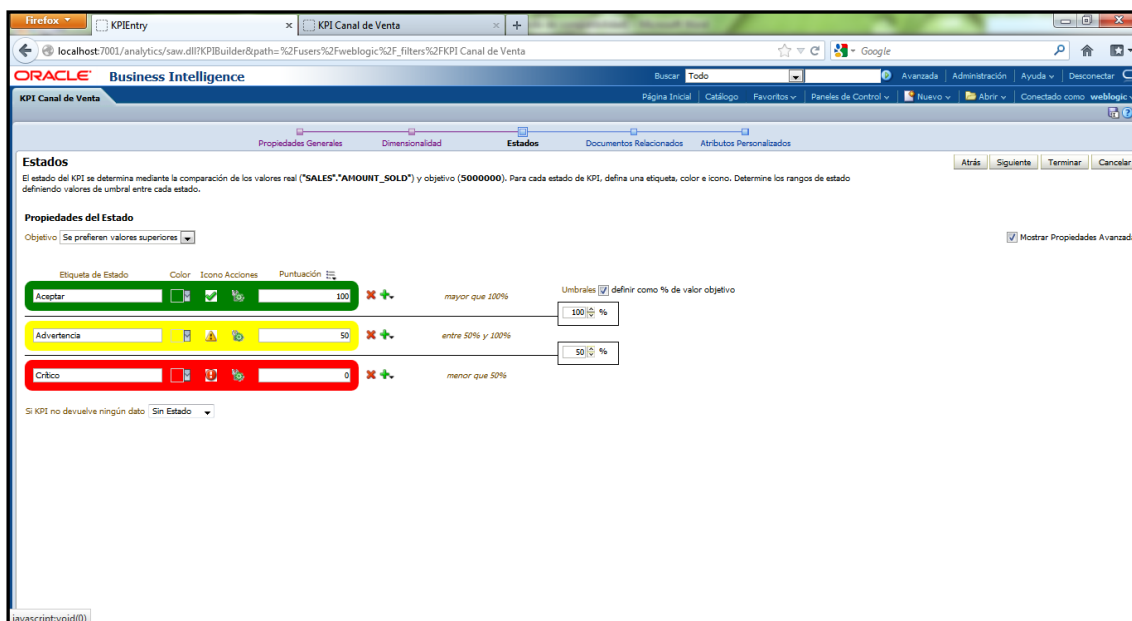
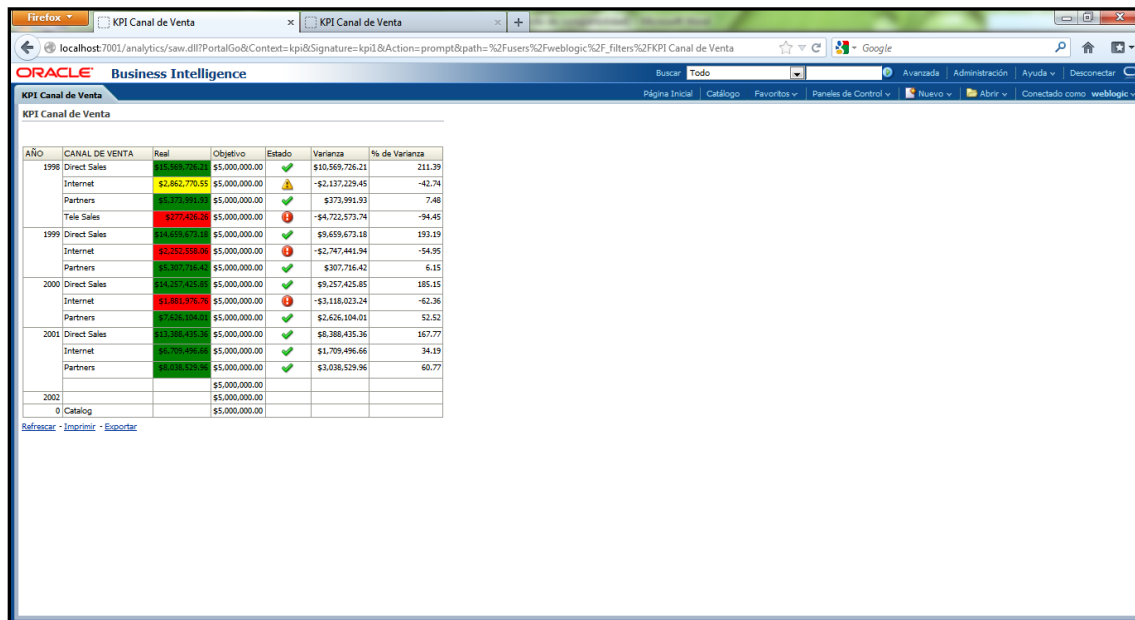


Figura 70. Estados de KPI

## CAPÍTULO 3. IMPLEMENTACIÓN DEL SISTEMA DE BUSINESS INTELLIGENCE

Se tiene como resultado del KPI lo mostrado en la Figura 71.



The screenshot displays the Oracle Business Intelligence interface. The main content area shows a table titled 'KPI Canal de Venta' with columns: AÑO, CANAL DE VENTA, Real, Objetivo, Estado, Varianza, and % de Varianza. The table lists data for the years 1998 through 2002, categorized by sales channels: Direct Sales, Internet, Partners, and Tele Sales. Each row includes numerical values for 'Real' and 'Objetivo', a status icon (green checkmark, yellow triangle, or red X), and calculated variance values. The interface includes a top navigation bar with 'Página Inicial', 'Catálogo', 'Favoritos', and 'Paneles de Control'. A search bar and user information are also visible.

AÑO	CANAL DE VENTA	Real	Objetivo	Estado	Varianza	% de Varianza
1998	Direct Sales	\$1,059,726.21	\$5,000,000.00	✓	\$10,569,726.21	211.39
	Internet	\$2,862,776.55	\$5,000,000.00	⚠	-\$2,137,229.45	-42.74
	Partners	\$5,773,991.93	\$5,000,000.00	✓	\$373,991.93	7.48
	Tele Sales	\$2,774,426.29	\$5,000,000.00	✗	-\$4,722,573.74	-94.45
1999	Direct Sales	\$9,659,673.18	\$5,000,000.00	✓	\$9,659,673.18	193.19
	Internet	\$2,252,338.06	\$5,000,000.00	✗	-\$2,747,441.94	-54.95
	Partners	\$5,377,712.40	\$5,000,000.00	✓	\$307,716.42	6.15
	Tele Sales	\$4,217,421.85	\$5,000,000.00	✓	\$9,257,425.85	185.15
2000	Direct Sales	\$1,811,976.79	\$5,000,000.00	✗	-\$3,118,023.24	-62.36
	Internet	\$5,116,104.01	\$5,000,000.00	✓	\$2,626,104.01	52.52
	Partners	\$5,388,435.36	\$5,000,000.00	✓	\$8,388,435.36	167.77
	Tele Sales	\$6,709,496.66	\$5,000,000.00	✓	\$1,709,496.66	34.19
2001	Direct Sales	\$8,036,529.96	\$5,000,000.00	✓	\$3,036,529.96	60.77
	Internet					
	Partners					
	Tele Sales					
2002	Direct Sales					
	Internet					
	Partners					
	Tele Sales					

Figura 71. Resultado de KPI

Se continúa con la creación del *dashboard*, en cada una de las secciones se agregan los análisis y el *prompt*, como se muestra en la Figura 72.

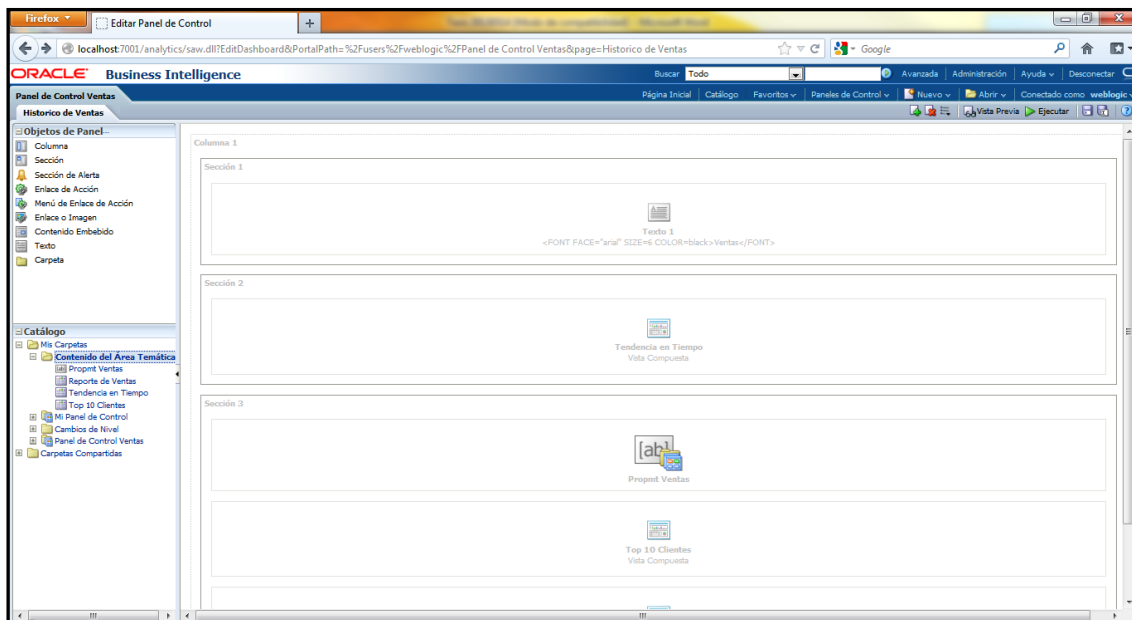


Figura 72. Adición de Objetos al *Dashboard*

Se tiene como resultado final el *dashboard* de ventas, como se muestra en la Figura 73.

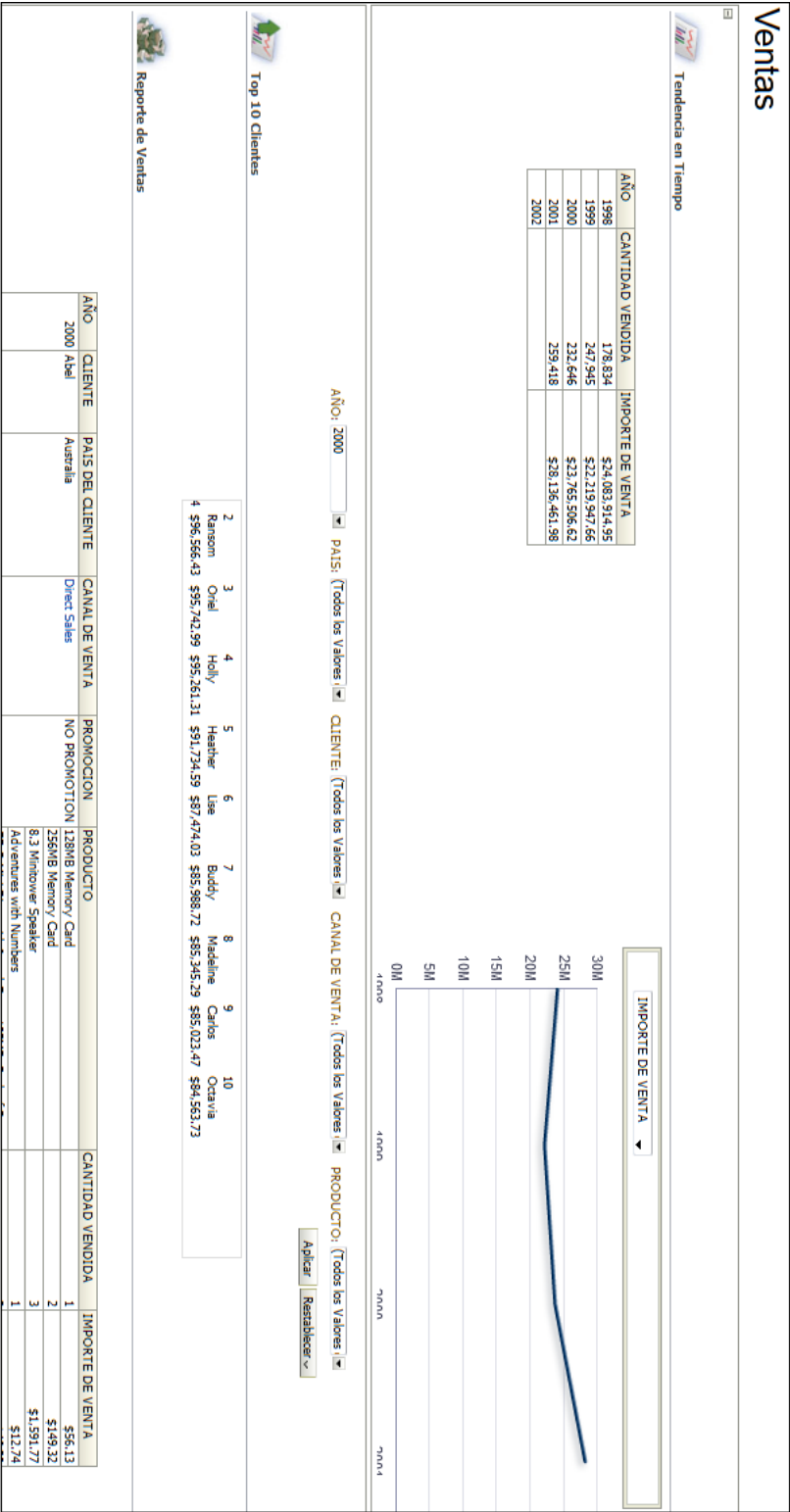


Figura 73. Dashboard de Ventas



# **CAPÍTULO 4. PLAN DE PRUEBAS Y RESULTADOS**

## 4. PLAN DE PRUEBAS Y RESULTADOS

Se proba el buen funcionamiento del sistema de *Business Intelligence* mediante la **usabilidad** utilizando preguntas de negocio a las cuales se darán respuesta.

### 4.1 Usabilidad

Para las pruebas se formulan tres preguntas de negocio relacionadas con las ventas.

**Pregunta 1.** ¿Cuál es el cliente con más ventas en el año 2001 y que cantidad e importe vendió?

Para responder la pregunta se necesita dar clic en el *prompt* de Año y seleccionar 2001 y se da clic en el botón *Apply* como se muestra en la Figura 74.

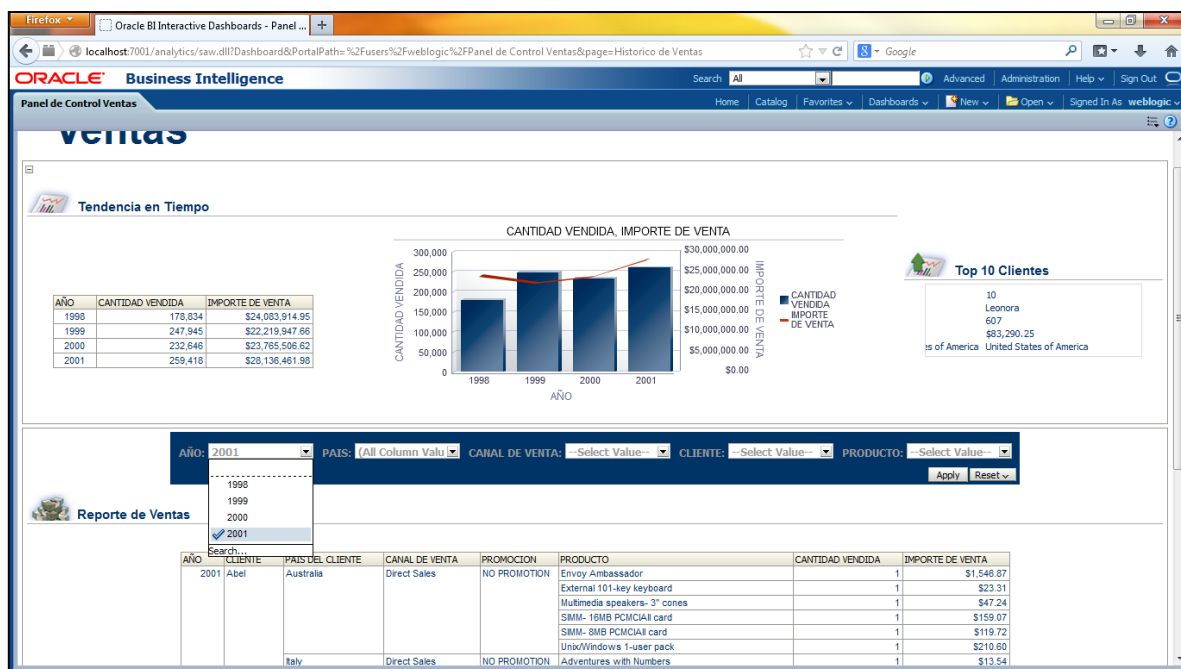


Figura 74. Selección de Año

En la sección **Top 10 de Clientes** se observa que el cliente con la posición número 1 es **Reginald** con una cantidad de **502** productos comprados y un importe de compra **\$99,413.84** en el año **2001** como se muestra en la Figura 75.

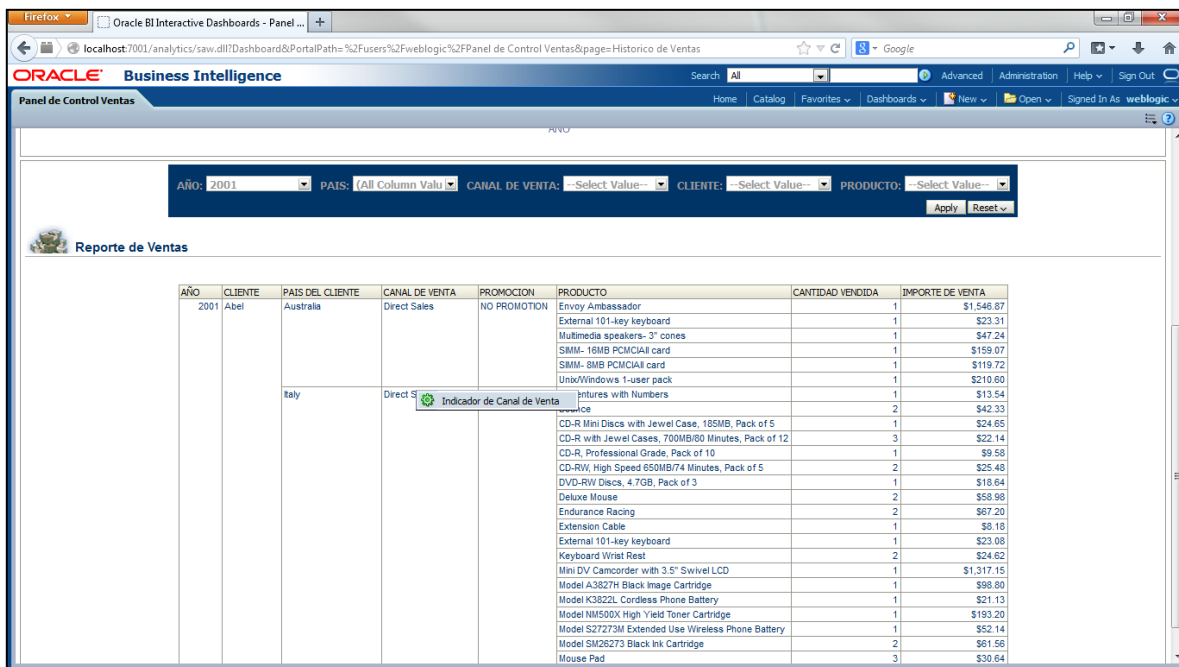


Posición	Nombre del Cliente	Cantidad de Productos	Importe de Compra	País
1	Reginald	502	\$99,413.84	United States of America
2	Gale	813	\$96,259.68	United States of America

**Figura 75.** Top 10 de Clientes

**Pregunta 2.** ¿Qué canal de venta ha sido el que mejor y peor resultados ha tenido a lo largo del tiempo?

La respuesta la se puede obtener dando clic sobre cualquier canal de venta en la sección reporte de ventas y se selecciona **Indicador de Canal de Venta** como se muestra en la Figura 76.



AÑO	CLIENTE	PAIS DEL CLIENTE	CANAL DE VENTA	PROMOCION	PRODUCTO	CANTIDAD VENDIDA	IMPORTE DE VENTA
2001	Abel	Australia	Direct Sales	NO PROMOTION	Envoy Ambassador	1	\$1,546.87
					External 101-key keyboard	1	\$23.31
					Multimedia speakers- 3" cones	1	\$47.24
					SMML- 16MB PCMCIAII card	1	\$159.07
					SMML- 8MB PCMCIAII card	1	\$119.72
					Unico/Windows 1-user pack	1	\$210.60
					Intunes with Numbers	1	\$13.54
					Service	2	\$42.33
					CD-R Mini Discs with Jewel Case, 105MB, Pack of 5	1	\$24.65
					CD-R with Jewel Cases, 700MB/80 Minutes, Pack of 12	3	\$22.14
					CD-R, Professional Grade, Pack of 10	1	\$9.58
					CD-RW, High Speed 650MB/74 Minutes, Pack of 5	2	\$25.48
					DVD-RW Discs, 4.7GB, Pack of 3	1	\$18.64
					Deluxe Mouse	2	\$58.98
					Endurance Racing	2	\$67.20
					Extension Cable	1	\$8.18
					External 101-key keyboard	1	\$23.08
					Keyboard Wrist Rest	2	\$24.62
					Mini DV Camcorder with 3.5" Swivel LCD	1	\$1,317.15
					Model A3827H Black Image Cartridge	1	\$98.80
					Model K3822L Cordless Phone Battery	1	\$21.13
					Model NM500X High Yield Toner Cartridge	1	\$193.20
					Model S27273M Extended Use Wireless Phone Battery	1	\$52.14
					Model SM26273 Black Ink Cartridge	2	\$61.56
					Mouse Pad	3	\$30.64

**Figura 76.** Selección de Indicador de Ventas



## CAPÍTULO 4. PLAN DE PRUEBAS Y RESULTADOS

Se puede observar en la tabla que el objetivo a cumplir es de **\$5,000,000.00** para todos los canales de venta con una varianza de advertencia del 50% por debajo del objetivo, ver Figura 77. Las ventas a través de Internet han sido las que peores resultados han tenido, se puede observar que en el año 1998 presenta una advertencia con una varianza del -42.74%, mientras que en los años de 1999 y 2000 quedaron muy por debajo del objetivo con -54.95% y -62.36% de varianza respectivamente como se muestra en la Figura 78.

Etiqueta de Estado	Color	Icono	Acciones	Puntuación		Umbral
Aceptar	Verde	✓	+	100	mayor que 100%	100 %
Advertencia	Amarillo	⚠	+	50	entre 50% y 100%	50 %
Crítico	Rojo	✗	+	0	menor que 50%	

Umbral ☒ definir como % de valor objetivo

Figura 77. Umbrales de Varianza

AÑO	CANAL DE VENTA	Actual	Target	Status	Variance	% Variance
1998	Direct Sales	\$15,569,726.21	\$5,000,000.00	✓	\$10,569,726.21	211.39
	Internet	\$2,862,770.55	\$5,000,000.00	⚠	-\$2,137,229.45	-42.74
	Partners	\$5,373,991.93	\$5,000,000.00	✓	\$373,991.93	7.48
	Tele Sales	\$277,426.26	\$5,000,000.00	✗	-\$4,722,573.74	-94.45
1999	Direct Sales	\$14,659,673.18	\$5,000,000.00	✓	\$9,659,673.18	193.19
	Internet	\$2,252,558.06	\$5,000,000.00	✗	-\$2,747,441.94	-54.95
	Partners	\$5,307,716.42	\$5,000,000.00	✓	\$307,716.42	6.15
	Tele Sales	\$277,426.26	\$5,000,000.00	✗	-\$4,722,573.74	-94.45
2000	Direct Sales	\$14,257,425.85	\$5,000,000.00	✓	\$9,257,425.85	185.15
	Internet	\$1,881,976.76	\$5,000,000.00	✗	-\$3,118,023.24	-62.36
	Partners	\$7,626,104.01	\$5,000,000.00	✓	\$2,626,104.01	52.52
	Tele Sales	\$277,426.26	\$5,000,000.00	✗	-\$4,722,573.74	-94.45
2001	Direct Sales	\$13,388,435.36	\$5,000,000.00	✓	\$8,388,435.36	167.77
	Internet	\$6,709,496.66	\$5,000,000.00	✓	\$1,709,496.66	34.19
	Partners	\$8,038,529.96	\$5,000,000.00	✓	\$3,038,529.96	60.77
	Tele Sales	\$277,426.26	\$5,000,000.00	✗	-\$4,722,573.74	-94.45

[Return](#) - [Create Bookmark Link](#)

Figura 78. KPI de Ventas I

Por otra parte el que mejores resultados ha tenido son las ventas directas, estando desde el año 1998 hasta el año 2001 con ventas muy superiores al objetivo véase en la Figura 79.

KPI Canal de Venta						
AÑO	CANAL DE VENTA	Actual	Target	Status	Variance	% Variance
1998	Direct Sales	\$15,569,726.21	\$5,000,000.00	✓	\$10,569,726.21	211.39
	Internet	\$2,862,770.55	\$5,000,000.00	⚠	-\$2,137,229.45	-42.74
	Partners	\$5,373,991.93	\$5,000,000.00	✓	\$373,991.93	7.48
	Tele Sales	\$277,426.26	\$5,000,000.00	❗	-\$4,722,573.74	-94.45
1999	Direct Sales	\$14,659,673.18	\$5,000,000.00	✓	\$9,659,673.18	193.19
	Internet	\$2,252,558.06	\$5,000,000.00	❗	-\$2,747,441.94	-54.95
	Partners	\$5,307,716.42	\$5,000,000.00	✓	\$307,716.42	6.15
2000	Direct Sales	\$14,257,425.85	\$5,000,000.00	✓	\$9,257,425.85	185.15
	Internet	\$1,881,976.76	\$5,000,000.00	❗	-\$3,118,023.24	-62.36
	Partners	\$7,626,104.01	\$5,000,000.00	✓	\$2,626,104.01	52.52
2001	Direct Sales	\$13,388,435.36	\$5,000,000.00	✓	\$8,388,435.36	167.77
	Internet	\$6,709,496.66	\$5,000,000.00	✓	\$1,709,496.66	34.19
	Partners	\$8,038,529.96	\$5,000,000.00	✓	\$3,038,529.96	60.77

[Return](#) - [Create Bookmark Link](#)

Figura 79. KPI de Ventas II

**Pregunta 3.** ¿Cuál es la tendencia de venta y que se puede pronosticar para años posteriores al 2001?

La respuesta se encuentra en la sección **Tendencia en Tiempo** en la cual se puede observar que las ventas a partir del año 1998 con **\$24,083,914.95** exceptuando el año 1999 con **\$22,219,947.66**, en donde se presenta una bajo con respecto al año anterior, han ido a la alza con **\$23,755,508.62** en el año 2000 y **\$28,136,461.98** en el año 2001.

Se puede pronosticar que la empresa se encuentra con una tendencia al alta para los próximos años como se muestra en la Figura 80.

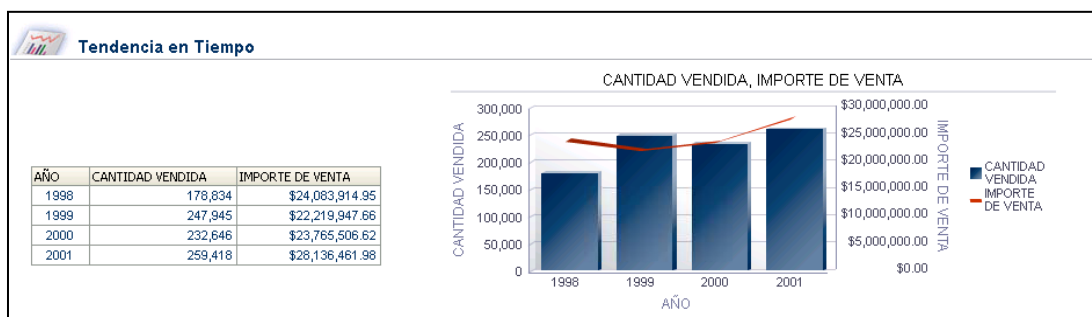


Figura 80. KPI de Ventas II

### CONCLUSIÓN

La presente tesis expuso la implementación de un sistema de *Business Intelligence*, la cual puede ser aplicada a cualquier tipo de organización, tomando como sistema fuente un *Data Warehouse* montado en una base de Datos *Oracle 11.2.0.1.0* y una herramienta de análisis *Oracle Business Intelligence 11.1.1.6.0*, el cual está basado en un *Data Mart* de Ventas. El sistema demostró que, mediante un *Dashboard*, puede determinar la tendencia de la empresa con respecto al tiempo, y si esta va en aumentado en los últimos años.

Se conoce, además, quienes son los diez clientes más importantes en la empresa por año, país y canal de venta, además de saber a detalle las ventas por año, cliente, país, canal de venta, promoción y producto. Por otra parte por medio de un KPI se puede conocer qué canales de venta están cumpliendo con el objetivo de ingresos anualmente.

Se logro tener un sistema confiable que brinda la información necesaria para la toma de decisiones. El enfoque dado esta dirigido al área de ventas pero no obstante se puede implementar la misma para cualquier área en la que se requiera hacer análisis.

El aprendizaje obtenido consta de poder ver los datos de otra manera, transformarlos en información que sea útil y proporcionarla en el momento que sea necesaria, quiénes toman decisiones en las organizaciones hoy en día han de tener, si quieren obtener ventajas competitivas, un acceso rápido y fácil a la información estratégica del negocio. La información estratégica confiable puede medirse y monitorearse con facilidad además de compartirse en toda la organización.

Una desventaja que se percibió en la implementación del sistema de *Business Intelligence* es el tiempo de respuesta de las consultas que genera el modelo de datos, mientras los sistemas transaccionales crecen en volumen de información las consultas tardan más tiempo en resolverse.

El futuro del *Business Intelligence* es prometedor ya que entre las tendencias destacan el *Real Time BI* ( Inteligencia de Negocios en Tiempo Real) que se refiere a la capacidad de proporcionar información sobre las operaciones de un negocio tan pronto como éstas ocurran, *Big Data* que conceptualiza el movimiento de grandes cantidades de información, *Mobile BI* que proporciona la toma de decisiones desde cualquier lugar por medio de las tecnologías móviles y *Social Media* que ayudara a realizar análisis de las redes sociales.

Tecnológicamente hablando el área de *Business Intelligence* representa un reto y una oportunidad para futuros ingenieros que deseen encontrar en los datos de las organizaciones información que cambie el futuro de las mismas.

### GLOSARIO DE TÉRMINOS

**Ad-hoc.-** Se refiere a personalizar una consulta en tiempo real, en vez de estar atado a las consultas prediseñadas para informes.

**Backup.-** Copia de los datos originales que se realiza con el fin de disponer de un medio de recuperarlos en caso de su pérdida.

**Batch.-** Se conoce como sistema por lotes a la ejecución de un programa sin el control, la interacción y la supervisión directa del usuario.

**Cache.-** Es un canal especial de memoria que poseen los ordenadores. Funciona de una manera similar a como lo hace la memoria principal (RAM *Random Access Memory*), pero es de menor tamaño y de acceso más rápido.

**Cardinalidad.-** Se define como el grado de participación de las entidades en una relación.

**Checklist.-** Una lista alfabética o sistemática de acciones que se usa para referencia, control, comparación, verificación o identificación.

**Data Mart.-** Es una versión especial de almacén de datos (*Data Warehouse*). Son subconjuntos de datos con el propósito de ayudar a que un área específica dentro del negocio pueda tomar mejores decisiones.

**Data Staging.-** Repositorio intermedio usado para carga temporal de datos en un proceso ETL.

**Drilling.-** Se define como profundizar o navegar en los datos.

**Id.-** Identificador único del registro

**Join.-** La sentencia join en SQL permite combinar registros de dos o más tablas en una base de datos relacional.

**Layout.-** Cuadrícula que divide en espacios o campos la página que se diseña para facilitar la distribución de elementos como textos ó gráficos en la misma.

**Lookup.-** Es una estructura de datos, normalmente un arreglo o un arreglo asociativo, que se usa para substituir una rutina de computación con una simple indexación de los arreglos.

**Market Basket.-** Un subconjunto de los productos y activos que está diseñado para imitar el funcionamiento de un mercado global.

**Metadata.-** Se refiere a la descripción de un conjunto de datos hechos entidades de información.

**Plug and Play.-** Es la tecnología que permite a un dispositivo informático ser conectado a una computadora sin tener que configurar ni proporcionar parámetros a sus controladores.

**Tuning.-** También conocido como afinación de bases de datos describe un grupo de actividades utilizadas para optimizar y homogenizar el desempeño de éstas.

## BIBLIOGRAFÍA

Bedell Jeffrey. (1997). *"Data Warehousing, Data Modeling and Design"*, DSSTraining, MicroStrategy, Inc.

Hanlon Denelle. (2013). *Best of Business Intelligence Volumen 9*. Recuperado de <http://tdwi.org>.

Kimball Ralph y Ross Margy. (2002). *The Data Warehouse Toolkit (2 ed.)*. USA: Wiley Computer Publishing.

Kimball Ralph y Ross Margy. (2008). *The Data Warehouse Lifecycle Toolkit (2 ed.)*. Indianapolis: Wiley Computer Publishing.

Moss T. Larissa y Atre Shaku. (2008). *Business Intelligence Roadmap. The complete Project Lifecycle for Decision-Support Applications*. Indiana: Addison-Wesley Information Technology Series.

Pressman S. Roger. (2012). *Ingeniería de software. Un enfoque práctico (5 ed.)*. Madrid: McGraw-Hill.

Reed Micheal. (2013). *A definition of Data Warehouse*. Recuperado de [www.technologyevaluation.com](http://www.technologyevaluation.com)

Oracle. (2009). *Legal Notices* Recuperado de <http://www.oracle.com/us/legal/terms/index.html>

Reed Micheal. (2013). *The necessity of Data Warehouse*. Recuperado de [www.technologyevaluation.com](http://www.technologyevaluation.com)

William H. Inmon. (2005). *Building the Data Warehouse*. Indianapolis: Wiley Computer Publishing.