

Proyecto de Grado - Taller V

***METODOLOGIAS Y LINEAMIENTOS  
PARA EL DISEÑO DE UN DW DESDE  
UNA METADATA GENERICA***

Virginia Ros - Paola Di Yorio - Gustavo González

Diciembre 2002

Tutor: Dr. Raúl Ruggia

Instituto de Computación – Facultad de Ingeniería

Universidad de la República

## Resumen

Un Data Warehouse (DW) es una base de datos que almacena información para la toma de decisiones. Las características de los mismos hacen que los modelos de datos y las estrategias de diseño usadas para bases de datos operacionales no sean aplicables. Lo que motiva el desarrollo de nuevas técnicas y estrategias de diseño.

Surge en el proceso de obtención de un DW la necesidad de automatizar la forma de llegar al diseño lógico del mismo. Para esto se parte de una Metadata Genérica y se siguen determinados pasos hasta llegar al diseño lógico.

Se entiende que una Metadata es una colección de datos sobre los datos, los que describen el contenido, calidad, condición y otras características de los datos. En el proceso se destacan dos tipos de Metadata una genérica y una particular, la genérica trata de obtener toda la información necesaria para describir una realidad, por otro lado la particular lo que hace es obtener de allí la información requerida para modelar los datos de una porción de esa realidad teniendo como agregados la información particular recogida por quienes vayan a hacer uso de la misma. Como ejemplo en este documento la Metadata General (MG) hace referencia a los Recursos del Mar, mientras que la Metadata Particular (MP) sólo toma lo relacionado con la Pesca para la realidad del I.N.A.P.E.

En este documento se presenta una herramienta para describir Metadatas, Bases de Datos Relacionales y requerimientos de forma homogénea y a partir de allí se desarrolla una Metodología que procesa una MG y obtiene una MP.

Propone lineamientos que faciliten la obtención de un Modelo Conceptual Multidimensional (CMDM) a partir de una MP.

Finalmente se mostrará el proceso descrito mediante un caso de estudio.

# Indice

1.	INTRODUCCIÓN .....	5
	Motivación.....	5
	Aportes y Limitaciones.....	5
	Trabajos Existentes.....	5
	Descripción del Proyecto.....	6
	Objetivos.....	7
	Desarrollo.....	7
	Organización del documento .....	8
2.	DEFINICIÓN DEL MARCO DE TRABAJO.....	9
3.	MÉTODOS PARA OBTENER UNA METADATA PARTICULAR.....	11
	Definiciones básicas.....	12
	Definición de Metadata.....	12
	Definición de Requerimientos .....	15
	Definición de Algoritmos .....	17
	Consideraciones generales .....	21
4.	MÉTODOS PARA EL PASAJE A CMDM.....	23
	Identificación de Niveles.....	23
	Identificación de atributos de interés.....	24
	Identificar Dimensiones.....	25
	Criterios a tomar por el Diseñador .....	26
	Definición de Primitivas .....	27
5.	CASO DE ESTUDIO .....	29
	Obtención de la Metadata Particular (MP).....	29
	Obtención de Modelo Conceptual (CMDM).....	33
	Identificación de Niveles.....	33
	Identificación de atributos de interés.....	34
	Identificar Dimensiones.....	34
	Criterios a tomar por el Diseñador .....	35
	Obtención del Modelo Lógico .....	37
	Lineamientos de Diseño .....	37
	Materialización de Relaciones Dimensionales.....	37
	Fragmentación de Dimensiones.....	38
	Fragmentación de Cubos.....	38
	Mapeo del CMDM con las bases fuentes.....	38
	Aplicación de Reglas de Transformación .....	41
	Conclusiones.....	44
6.	CONCLUSIONES Y TRABAJOS FUTUROS .....	45
	Bibliografía y Lecturas Relacionadas .....	47
7.	ANEXOS.....	48

## Algoritmos y Definiciones

Definición 3.1 – Metadata.....	12
Definición 3.2 – Atributo y Conjunto de Atributos.....	13
Definición 3.3 – Relación y Conjunto de Relaciones.....	13
Definición 3.4 – Entidad y Conjunto de Entidades .....	14
Definición 3.5 – Requerimiento.....	15
Definición 3.6 – CA, conj. de atrib. seleccionados en el requerimiento .....	15
Definición 3.7 – CAF, conj. de atrib. seleccionados en las fórmulas y fórmula .....	16
Definición 3.8– CAC, conj. de atrib. seleccionados en las condiciones y condición.....	16
Algoritmo 3.1 – $MG \cup BD$ .....	18
Algoritmo 3.2 – $SMG \cap Req$ .....	19
Algoritmo 3.3 – MG y Req. sin OLTP .....	21
Algoritmo 4.1 – Crear niveles para CMDM .....	23
Algoritmo 4.2 – Crear Dimensiones del tipo fecha .....	24
Algoritmo 4.3 – Identificar medidas.....	24
Algoritmo 4.4 – Dimensiones a partir de agregaciones .....	25
Algoritmo 4.5 – Dimensiones a partir de niveles relacionados.....	25
Algoritmo 4.6 – Dimensiones en base a los requerimientos .....	26

## Tabla de Ilustraciones

Fig. 1.1 – Obtención del Modelo Lógico .....	6
Fig. 1.2 – Obtención del CMDM.....	6
Fig. 2.1 – Obtención del Modelo Lógico .....	10
Fig. 2.2 – Obtención del CMDM.....	10
Fig. 3.1 – Obtención de SMG .....	17
Fig. 3.2 – Obtención de MP .....	18
Fig. 3.3 - SMG .....	20
Fig. 3.4 – No existe la OLTP.....	20
Fig. 5.1 - MG .....	30
Fig. 5.2 – MG con relaciones.....	31
Fig. 5.3 - OLTP .....	31
Fig. 5.4 – OLTP con relaciones .....	32
Fig. 5.5 - MP .....	32
Fig. 5.6 – MP con relaciones .....	33
Fig. 5.7 – Identificación de Niveles.....	33
Fig. 5.8 – Identificación de atributos de interés.....	34
Fig. 5.9 – Identificación de dimensiones.....	34
Fig. 5.10 – Eliminación de redundancias y establecimiento de jerarquías.....	35
Fig. 5.11 – Relación Dimensional Pesca.....	36
Fig. 5.12 – Cubo Pesca.....	37
Fig. 5.13 – Cubo Pesca.....	38
Fig. 5.14 – Mapeo de la Dimensión Pesca .....	39
Fig. 5.15 – Mapeo de la Dimensión Fecha Llegada .....	39
Fig. 5.16 – Mapeo de las Dimensiones Puertos, Medios de Transporte y Especies Animales .....	40
Fig. 5.17 – Mapeo del Cubo Pesca .....	40
Fig. 6.1 – Resumen considerando OLTP definida .....	45
Fig. 6.2 - Resumen sin considerar OLTP .....	46

## 4.1. Introducción

### Motivación

Dado el uso cada vez más frecuente de las herramientas de DW surge el interés de obtener procesos más automáticos para llegar al diseño lógico de los mismos.

A la vez ha crecido el uso de Metadatos, dada la facilidad que brindan al usuario para ubicar y entender las principales características de los datos, sin necesidad de transferirlos. Algunos de los principales usos son: organizar y mantener las inversiones realizadas en datos de una organización, proveer información a catálogos de datos y proveer información para facilitar la transferencia de los mismos.

La idea inicial era obtener a partir de una Metadata Genérica una Metadata Particular, luego con ella y el análisis de requerimientos llegar al diseño conceptual multidimensional para finalmente a partir de éste poder formular el diseño lógico del DW.

Del análisis de las distintas fases resulta que no se cuenta con una metodología para llegar al Modelo Conceptual a partir de una Metadata y las bases fuentes, que describen la realidad del usuario, es así que surge la idea de obtener una metodología que complete el ciclo de desarrollo.

### Aportes y Limitaciones

Los aportes fundamentales son la metodología que permite llegar desde el planteo de una MG de una realidad determinada a una MP combinando para eso el análisis de requerimientos y las bases fuentes existentes; una vez concluida esta etapa se brindan lineamientos que permiten llegar al Modelo Conceptual Multidimensional de forma automatizada.

Esta metodología surge al formalizar los pasos que se plantean como necesarios para realizar el objetivo inicial del proyecto. Es así que la falta de existencia de una metodología previa marcó la necesidad de plantear una de manera formal, para ser aplicada en otras realidades.

El desarrollo del Método surge de una aplicación concreta, ya que no fue desarrollado en un ambiente puramente teórico, sino con bases prácticas, sorteando las dificultades que se fueron dando, por lo que éste desarrollo del proyecto es en sí mismo una descripción del análisis de la necesidad de una metodología.-

Surge también de este trabajo una forma de representar Metadatos y bases OLTP de forma que manejen una misma estructura que permita la interacción entre ellas.

Otro punto interesante es que se aplicaron a una realidad determinada el CMDM, permitiendo definir lineamientos generales para futuras aplicaciones y el Diseño lógico de un DW a partir del CMDM

### Trabajos Existentes

Como se mencionó anteriormente la idea es que a partir de una realidad, y una metadata general de la misma llegar al diseño de un DW, para ello se utilizaron los trabajos realizados para definir un modelo conceptual dimensional y un modelo lógico.

En referencia con las definiciones de Metadata se estudiaron las definiciones brindadas por el FGDC (Comité Federal de Datos Geográficos) de Estados Unidos, que vienen siendo divulgados como un estándar de metadatos geoespaciales [CH00].

El modelo conceptual multidimensional se basa en el definido en la tesis de maestría de Fernando Carpani [FC00], "CMDM: Un Modelo Conceptual para la Especificación de Bases Multidimensionales". El cual permite una especificación detallada de una base de datos multidimensional, que se construye mediante un lenguaje gráfico que permite describir las estructuras de datos y algunas restricciones de integridad.

Para la obtención del Modelo Lógico del DW se utilizó la metodología brindada por Verónica Peralta[VP01] en la tesis llamada "Diseño Lógico de un DW a partir de un Esquema Conceptual Multidimensional". En el mencionado trabajo se tratan los problemas de diseño lógico de un DW

y se propone una metodología para traducir el CMDM a un esquema lógico relacionado. La traducción se lleva a cabo aplicando transformaciones de alto nivel al esquema relacional de la base operacional, las cuales abstraen y generalizan diferentes técnicas de diseño de DWs. El esquema lógico es una especificación más detallada que el esquema conceptual donde se incorporan nociones de almacenamiento, performance y estructuración de los datos.

## Descripción del Proyecto

La metodología descrita en este documento no surge como el objetivo inicial del proyecto, sino que se llega a ella como una consecuencia en el desarrollo del mismo.

Para dar una visión general de este desarrollo, se presenta a continuación a modo de resumen gráfico los pasos seguidos por la metodología (Fig. 1.1, 1.2)

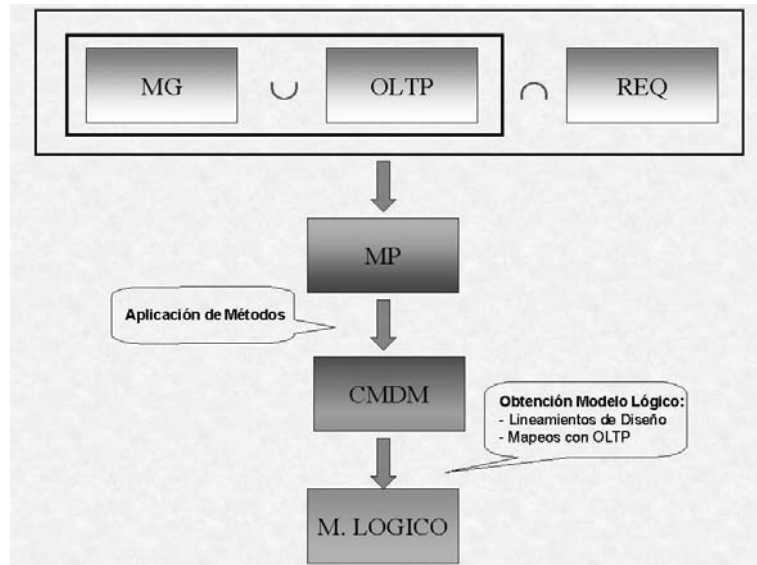


Fig. 1.1 – Obtención del Modelo Lógico

En este primer caso se resume el proceso para obtener el Modelo Lógico de un DW, a partir de una MG, la OLTP y los requerimientos, al aplicar los métodos descritos en los capítulos siguientes.

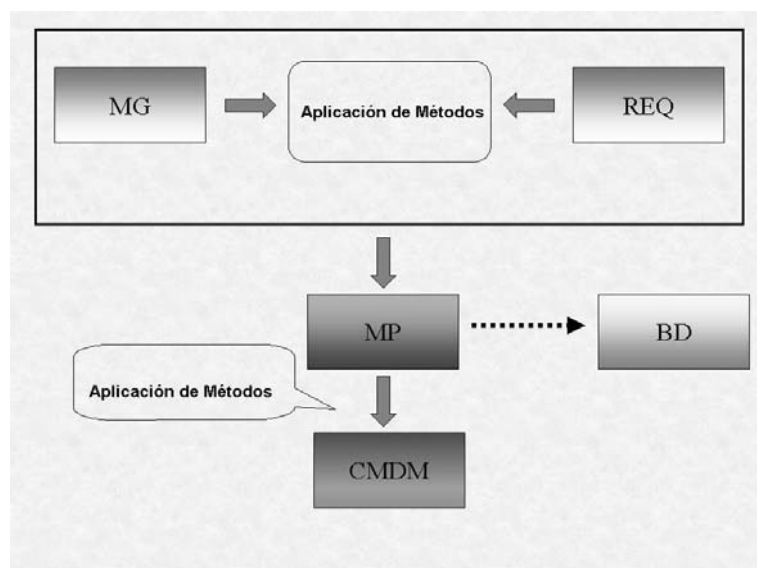


Fig. 1.2 – Obtención del CMDM

En este caso se muestra el proceso para obtener el Modelo Conceptual y la definición de una posible OLTP, a partir de una MG y los requerimientos, al aplicar los métodos descritos en los capítulos siguientes.

## Objetivos

Inicialmente el objetivo principal era desarrollar una herramienta informática que permitiera realizar operaciones de análisis de información, especializado en Recursos del Mar. Esta herramienta debería permitir el análisis de datos numéricos y geo-referenciados, combinando funcionalidades tipo OLAP con consultas relacionales y cartográficas. La información en cuanto a la aplicación de la herramienta, así como la interpretación de los datos, se obtendría a partir del INAPE (Instituto Nacional de Pesca).

Durante el desarrollo del proyecto surge la necesidad de profundizar en las metodologías aplicadas hasta el momento en forma intuitiva, de manera de brindar una formalización de las mismas, simplificando así la tarea de futuras aplicaciones, redefiniendo así los objetivos iniciales.

Es así que se plantea como objetivo que dada una Metadata Genérica, un Análisis de Requerimientos y la posible existencia de una OLTP, se obtenga una Metadata Particular, a partir de la cual obtener un modelo conceptual multidimensional.

Para lograr este objetivo, dada la necesidad de interoperar entre distintos ambientes de trabajo, es que se brinda una definición de cómo presentar una Metadata, una OLTP y los requerimientos, de forma de obtener documentos homogeneizados, para así simplificar la tarea de cruzarlos y obtener un documento único como la MP.

Una vez obtenida la MP, y volviendo a tener en cuenta el análisis de requerimientos se obtendrá el CMDM, contando con lineamientos que brinden facilidades para la definición del mismo.

Para finalizar, a partir del CMDM, se aplica el caso de estudio para así obtener el modelo lógico.

## Desarrollo

Este ítem intenta mostrar la carencia existente en el proceso de pasar de documentos en los que se representan Metadatos, al modelo conceptual para el diseño de un DW, ya que si bien el mismo se puede realizar de forma intuitiva, no se cuenta con ninguna formalización al respecto, de forma que el pasaje pueda hacerse en forma guiada y ordenada.

Se comenzó a trabajar con el fin de cumplir los objetivos iniciales, por lo tanto, se trabajó en la búsqueda de una Metadata Genérica de Recursos del Mar, utilizando para esto como fuente de datos principal, la información obtenida en Internet.

Al comienzo del proyecto, se pretendía obtener los siguientes productos de forma incremental:

- ◆ Análisis de Requerimientos de la realidad del INAPE, orientado hacia el diseño de un DW
- ◆ Metadata Particular del usuario INAPE, como una particularización de una Metadata Genérica de Recursos del Mar
- ◆ Modelo Conceptual Multidimensional (CMDM), aplicado a la Metadata Genérica, y aplicado a la Metadata Particular
- ◆ Modelo Lógico de la realidad del usuario a partir del CMDM, la OLTP y el análisis de requerimientos
- ◆ Aplicación de consultas sobre la realidad modelada integrando herramientas OLAP y GIS

Al no existir una Metadata como se buscaba, se resolvió con el apoyo del INAPE, desarrollar la misma, para ello en principio se utilizó el estándar definido por la FGDC (Federal Geographic Data Committee), que si bien está especializado en datos geográficos brindaba un marco para definir los datos de interés. La creación de esta Metadata, en una primera instancia se realizó basada en estas definiciones. Su contenido se nutrió de la realidad del INAPE y una generalización intuitiva de los mismos.

A partir de este documento y de la realidad del INAPE - modelada en su OLTP -, se obtuvo la Metadata Particular. A partir de ella y junto con el análisis de requerimientos, determinó el armado del Modelo Conceptual Multidimensional, para así obtener el DW deseado. Es aquí que surgen de forma práctica e intuitiva algunos lineamientos no muy acabados que fueron la base de ese desarrollo.

Una vez obtenido el CMDM, se realizó el mapeo de las estructuras representadas en éste, a las bases de datos fuentes, utilizando para ello las definiciones brindadas en la tesis de maestría de Verónica Peralta [VP01] para la obtención de un modelo lógico. Al llegar a este punto se debe rever algunas estructuras planteadas en el CMDM.

Una vez llegada a esta instancia, y evaluando el trabajo realizado hasta el momento, es que se decide hacer una formalización de la metodología aplicada hasta ahora de forma intuitiva. Permitiendo que futuras aplicaciones cuenten con la existencia de una metodología automatizada que facilite y ordene el desarrollo de un proceso similar.

## Organización del documento

El documento estará dividido en ocho capítulos, estructurados de la siguiente forma:

1. **Introducción**
2. **Definición del marco de trabajo**, aquí se plantean las distintas realidades a las que se puede enfrentar el usuario.
3. **Metodología para la obtención de una Metadata Particular**, se presenta un estándar para definir una metadata y la formalización de cómo a partir de ella, un conjunto de requerimientos y una posible OLTP llegar a una MP.
4. **Lineamientos para el pasaje al CMDM**, se indican los lineamientos que facilitan el pasaje al CMDM y permiten una evolución más ágil hacia el modelo final.
5. **Caso de estudio**, breve presentación de un caso de estudio aplicando lo visto en los capítulos anteriores.
6. **Conclusiones y trabajos futuros**
7. **Anexos**



## 4.2. Definición del Marco de Trabajo

En este capítulo, se va a dar un marco de referencia, para situar la participación de una Metadata Particular con relación al diseño e implementación de un DataWarehouse.

Se entiende por Metadata a una colección de datos que describen otros datos. En el proceso es de interés destacar dos tipos de Metadata una genérica y una particular, la genérica describe los datos implicados en una rama de un negocio, por otro lado la particular lo que hace es obtener de allí la información requerida para modelar los datos de una porción de ese negocio teniendo como agregados la información particular recogida por quienes vayan a hacer uso de la misma. Como ejemplo en este documento la Metadata General (MG) hace referencia a los Recursos del Mar, mientras que la Metadata Particular (MP) sólo toma lo relacionado con la Pesca para la realidad del I.N.A.P.E.

Se van a distinguir dos ambientes de trabajo posibles, el primero que considera la existencia de una Base de Datos transaccional a partir de la cuál se va a realizar la carga del DW, y el otro es un ambiente en el que, al momento de diseñar el mismo, no exista la mencionada base de datos.

Es claro que cuando se quiera implementar el DW, el mismo va a cargarse con los datos existentes en la base de producción u otros medios en que se almacenen datos de interés, pero podría querer diseñarse sin considerar la existencia de la misma, o realizar un diseño conjunto, en el que la propia base de producción se diseñe para poder satisfacer los requerimientos del DW.

Para el primer caso (Fig. 2.1), los elementos de interés que participan son los documentos referentes a:

- ◆ la Metadata Genérica (MG)
- ◆ los Requerimientos (REQ), que atienden las necesidades de la implementación del DW.
- ◆ el esquema de la base de datos de producción (OLTP).

A raíz de estos elementos se procede al cruce de los mismos, de acuerdo a la metodología que se define en el Capítulo 3. De esta forma se obtiene una Metadata Particular (MP) que recoge los datos que son de interés para el diseño del DW, de acuerdo a la especificación de los requerimientos, más un conjunto de Entidades y Relaciones que el usuario considera de su interés para el desarrollo del DW.

Una vez obtenida la MP, junto con los requerimientos se procede a la definición del Modelo Conceptual (CMDM), que se detallará en el Capítulo 4.

Luego de definido el diseño conceptual del DW, se procede a obtener el Modelo Lógico. Para ello es necesario tomar decisiones de diseño, como por ejemplo, que esquema de DW se utilizará (Estrella, Copo de Nieve, etc.), y los tipos de fragmentación a utilizar (horizontal, vertical). Una vez obtenidos estos lineamientos se procede a realizar el mapeo entre los distintos objetos del CMDM y la OLTP, lo que se detallará en el Capítulo 5.

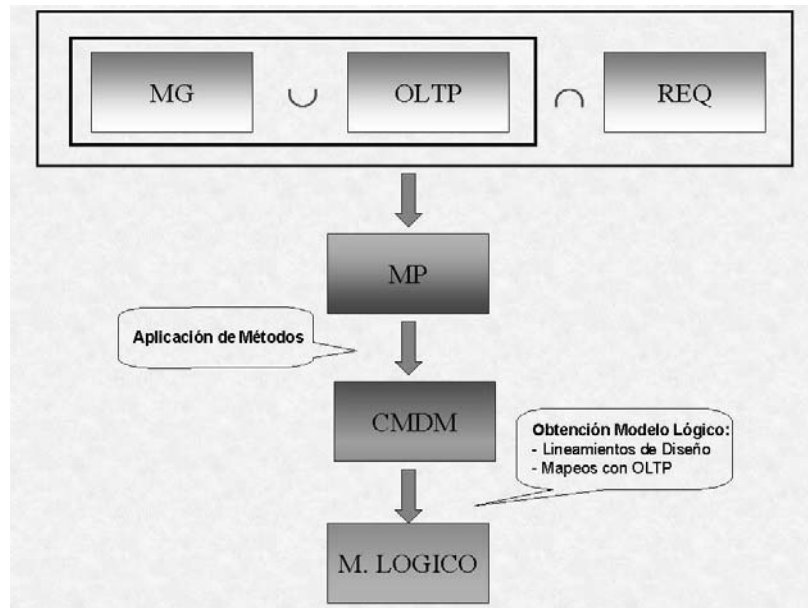


Fig. 2.1 – Obtención del Modelo Lógico

A través del cruce entre la MG, la OLTP y los requerimientos, de acuerdo a la metodología a definir en el Capítulo 3, se obtiene la MP que recoge los datos que son de interés para el diseño del DW, de acuerdo a la especificación de los requerimientos. Aplicando los métodos a definir en el Capítulo 4, se obtiene el diseño CMDM, finalmente se aplican los metodología definida para la obtención del modelo lógico.

Para el segundo caso (Fig. 2.2), sólo se cuenta con los documentos referentes a:

- ◆ la Metadata Genérica (MG)
- ◆ los Requerimientos (REQ).

Se procede de la misma forma que en el caso anterior, sin tener en cuenta la OLTP, ya que aún no está definida. En este caso, sólo se llega a obtener el CMDM, ya que para obtener el Modelo Lógico, es necesario la OLTP, como se verá en los capítulos sucesivos.

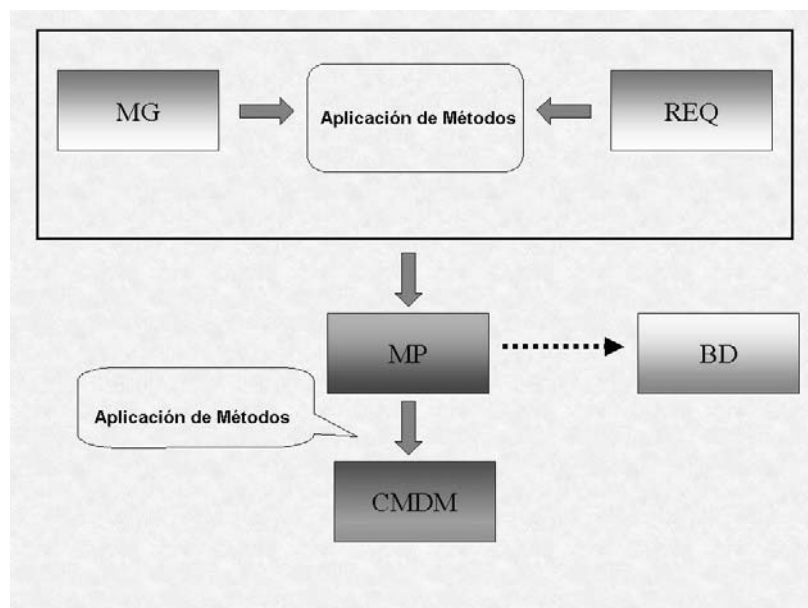


Fig. 2.2 – Obtención del CMDM

Se obtiene el modelo conceptual a partir de los requerimientos y la metadata genérica, a partir de este modelo conceptual se puede definir la base transaccional para un sistema enfocado a consultas multidimensionales.

### 4.3. Métodos para obtener una Metadata Particular

Una descripción a grandes rasgos de los pasos que se siguen para implementar un DW, podría incluir:

- ◆ Análisis de Requerimientos
- ◆ Diseño Conceptual
- ◆ Diseño Lógico
- ◆ Implementación

En éste capítulo se va a definir una metodología que permita hacer más fácil y ordenado el pasaje de la abstracción de los datos de la realidad, junto con el Análisis de Requerimientos al Diseño Conceptual del DW.

Para eso, se supone la existencia de una Metadata Genérica que describe una realidad mayor que la que interesa representar en el DW.

A modo de ejemplo, y dentro del caso de estudio que se tratará en el presente documento, se toma como punto de partida una Metadata Genérica que describe ampliamente los distintos elementos que se pueden encontrar asociados al mar. Se hará referencia a esta Metadata, como Metadata de Recursos del Mar.

Surge entonces la idea de, que a partir de una Metadata Genérica y los requerimientos sobre una realidad concreta, se pueda definir una Metadata Particular que satisfaga los requerimientos del usuario y a partir de la MP definir el Diseño Conceptual del DW.

El DW toma sus datos de la base transaccional o de producción, al momento de estar implementado, pero puede suceder que al diseñarse el mismo, la base de producción todavía no esté creada, o incluso de existir, querer mantenerse bastante alejado de la misma a la hora del diseño, de forma que el diseño conceptual no se vea restringido por la implementación de la OLTP. Es por eso que se van a considerar dos casos en la metodología que se definirá, el primero que considera la existencia de una OLTP, y el segundo en el que la misma no existe.

A los efectos de dar una metodología que permita definir una Metadata Particular se darán un conjunto de definiciones, con las cuales poder representar una **METADATA**, como un conjunto formado por **Entidades, Atributos y Relaciones**.

Este conjunto de definiciones va a ser útil en el momento de representar el esquema de la Base de Datos. El contar con una forma uniforme de representación, permite interoperar entre los dos esquemas de forma bastante natural.

En resumen la metodología para el pasaje de una MG a una MP consta de dos pasos fundamentales:

- ◆ A partir de las definiciones que se brindan a continuación, se obtendrán los documentos de la Metadata general, de los requerimientos y de la OLTP, en caso de existir. Estos documentos serán la entrada para el segundo paso.
- ◆ El segundo paso consiste en aplicar los algoritmos definidos sobre dichos documentos.

## Definiciones básicas

Se utilizará como base para la definición de una metadata las definiciones realizadas en el libro de referencia "Applying Reuse in Conceptual Modelling: a toolkit approach" [RR96] que permiten representar un Modelo Entidad Relación Extendido (EER), el mismo está basado en el Modelo Entidad-Relación [Chen76] extendido con los constructores de generalización y agregación, y con atributos compuestos y multivaluados. [Bati92] [Bouz94].

Con la intención de obtener representaciones compatibles, es que se define la metadata y los requerimientos en base a las estructuras determinadas por el modelo conceptual EER.

### Consideraciones:

- ◆ Los campos que aparecen subrayados en las definiciones son obligatorios.
- ◆ Los nombres de entidades y relaciones son únicos para todo el esquema de metadata
- ◆ Los nombres de los atributos son únicos dentro de cada entidad o relación en la cual participan
- ◆ El campo origen, definido tanto en las entidades como en los atributos es necesario para el momento de correr los algoritmos de apareo para obtener una metadata particular, a partir de ciertos requerimientos. Estos requerimientos pueden ser realizados si sus componentes forman parte de la base de datos, en caso contrario ésta debe redefinirse o modificar el requerimiento.
- ◆  $\langle e_1, e_2, \dots, e_n \rangle$  Representa una n-upla .
- ◆ Todo aquel elemento definido entre llaves  $\{ \}$  define un conjunto de ese elemento.

## Definición de Metadata

La abstracción que se realiza a continuación, permite conceptualizar tanto el Esquema de la Base de Datos, como la representación de la Metadata Genérica en términos de un conjunto de Entidades, cada una de ellas integrada por un conjunto de Atributos y un conjunto de Relaciones sobre las Entidades.

El esquema de Metadata definido de esta manera deberá ser instanciado por el Esquema de Base de Datos (si corresponde) y la Metadata Genérica para poder luego aplicar los algoritmos que se definen más adelante, obteniendo una nueva instancia de estas definiciones: la Metadata Particular.

**METADATA**  $\equiv$   $\langle$  nombre, CE, CR, descripción  $\rangle$

**Donde:**

**nombre:** String, es el nombre del esquema de Metadata

**CE :** CONJ\_ENTIDADES

Debe cumplirse que:

$CE \leftrightarrow \emptyset$

$\{ \forall e_i, e_j \in CE, i \neq j, e_i.\text{nombre} \neq e_j.\text{nombre} \}$

**CR :** CONJ\_RELACIONES, es el conjunto de relaciones que intervienen en el esquema

**descripción:** String, breve descripción del contenido y significado de la metadata

Definición 3.1 – Metadata

**ATRIBUTO**  $\equiv$  <nombre, tipo\_dato, card\_min, card\_max, largo, descripción, unidad\_medida, dominio, origen>

**Donde:**

**nombre:** String, es el nombre del atributo

**tipo\_dato:** tipo\_at, es el tipo de datos del atributo

**tipo\_At ::=** Real | Integer | String | Date | char | Set(tipo\_at) | tupla(list\_at)

**lista\_at ::=** nombre:tipo\_at

**card\_min:** N, donde es la cardinalidad mínima del atributo, es decir cuantas ocurrencias como mínimo deben haber para un valor del atributo

**card\_max:** N, donde es la cardinalidad máxima del atributo, es decir cuantas ocurrencias como máximo pueden haber para un valor del atributo

**largo:**  $Z^+$ , es el largo máximo que puede tener la instancia de un atributo

**descripción:** String, explica brevemente a que hace referencia el mismo

**unidad\_medida:** String, explica en caso de ser una cantidad si ésta está tomada en una medida en particular

**dominio:**  $\{e_i / \text{tipo}(e_i) = e_i.\text{tipo\_dato} \forall i\}$ , rango de valores permitidos para las ocurrencias del atributo

**origen:** {'MG', 'BD'}, hace referencia al origen del mismo, por defecto tiene el valor MG (metadata), pero al ejecutar los algoritmos de apareo con la base de datos si existe allí se reemplaza con el valor BD

**CONJ\_ATRIBUTOS**  $\equiv$   $\{a_i / a_i \text{ es un ATRIBUTO } \forall i\}$

Definición 3.2 – Atributo y Conjunto de Atributos.

**RELACION**  $\equiv$  < nombre, CA, ent\_participantes>

**Donde:**

**nombre:** String, es el nombre de la relación

**CA:** CONJ\_ATRIBUTOS, es el conjunto de atributos que integra la relación

**ent\_participantes:**  $\{E_p_i / E_p_i \text{ es una ent\_participante } \forall i\}$

**ent\_participante:** <e, card\_min, card\_max, rol>

**Donde:**

**e:** String, y es el nombre de la entidad que participa en la relación

**card\_min:**  $Z^+$ , es el mínimo número de instancias de la relación asociada a cada instancia de la entidad participante

**card\_max:**  $Z^+$ , es el máximo número de instancias de la relación asociada a cada instancia de la entidad participante

**rol:** String, es el rol que juega la entidad en la relación

**CONJ\_RELACIONES**  $\equiv$   $\{r_i / r_i \text{ es una RELACION } \forall i\}$

Definición 3.3 – Relación y Conjunto de Relaciones.

**ENTIDAD**  $\equiv$   $\langle$  nombre, CA, clave, descripción, tipo\_ent, origen  $\rangle$

**Donde:**

**nombre:** String, hace referencia al nombre de la entidad y debe ser único en el esquema

**CA:** CONJ\_ATRIBUTOS, es el conjunto de atributos que integran la entidad. Debe cumplir las siguientes restricciones:

$$CA \leftrightarrow \emptyset$$

$$\{\forall a_i, a_j \in CA, i < j, a_i.\text{nombre} \neq a_j.\text{nombre}\}$$

**clave:**  $\{n_i / n_i \text{ es un CA.nombre } \forall i\}$ , es el conjunto de atributos que conforman un identificador para la entidad.

**descripción:** String, explica brevemente a que hace referencia el mismo

**tipo\_ent:**  $\{\langle E.\text{nombre}, \text{min\_comp}, \text{max\_comp}, \text{min\_agr}, \text{max\_agr}, \text{rol} \rangle / E \in ENTIDAD\}$  Este campo, solo existe en la definición de una entidad si ésta es del tipo agregación, y describe cuales son los nombres de las entidades que la componen, y cual es el rol que juegan.

**min\_comp:** Es el mínimo número de instancias de la entidad componente asociado a cada instancia de la entidad que conforma la agregación.

**max\_comp:** Es el máximo número de instancias de la entidad componente asociado a cada instancia de la entidad que conforma la agregación.

**min\_agr:** Es el mínimo número de instancias de la entidad que conforma la Agregación, asociado a cada instancia de la entidad componente.

**max\_agr:** Es el máximo número de instancias de la entidad que conforma la Agregación, asociado a cada instancia de la entidad componente.

**rol:** String, es el rol que juega la entidad componente en la agregación

**origen:** {'MG', 'BD'}, hace referencia al origen del mismo, por defecto tiene el valor MG (metadata), pero al ejecutar los algoritmos de apareo con la base de datos si existe allí se reemplaza con el valor BD

**CONJ\_ENTIDADES**  $\equiv$   $\{e_i / e_i \text{ es una ENTIDAD } \forall i\}$

Definición 3.4 – Entidad y Conjunto de Entidades

## Definición de Requerimientos

La obtención de la Metadata Particular, se realizará cruzando la Metadata Genérica y la Base de Datos - si existiera -, junto con los requerimientos que se quieren realizar sobre el DW.

Las definiciones que siguen permitirán expresar los requerimientos de forma tal que sea fácil realizar el cruzamiento entre estas estructuras.

Para eso se definirá un requerimiento como:

- ◆ un conjunto de entidades (CE), a las cuales pertenecen los atributos seleccionados y otras de interés para expresar la condición del requerimiento
- ◆ un conjunto de atributos que se quieren seleccionar, sobre las cuales pueden existir cálculos y/o funciones de agregación (CAS). Este conjunto es la unión entre el conjunto de nombres de atributos que son de interés al realizar la consulta (CA) y el conjunto de nombres de aquellos atributos sobre los cuales se aplican fórmulas y o cálculos (CAF)
- ◆ un conjunto de nombres de atributos que intervienen en el requerimiento, ya sea en la selección, en las funciones de agregación o en las condiciones (CAT). Este conjunto está compuesto por la unión del CAS, y el conjunto de nombres de atributos que participan en las condiciones (CAC)
- ◆ un conjunto de condiciones (CC), que pueden tanto relacionar instancias de distintas entidades, así como filtrar instancias
- ◆ un conjunto de fórmulas que participan en el requerimiento (CF).

Se asume que el atributo esta prefijado con el nombre de la entidad a la que pertenece.

**REQUERIMIENTO**  $\equiv \langle \underline{CE}, \underline{CAS}, \underline{CAT}, CC, CF \rangle$

**CE:** {e.nombre / e es una ENTIDAD}, es el conjunto nombres de entidades participantes en el requerimiento

**CAS** = CA  $\cup$  CAF

**CAT** = CAS  $\cup$  CAC, es el conjunto de nombres de atributos que intervienen en el requerimiento, ya sea en la selección, en las funciones de agregación o en las condiciones.

**CC** : Conj\_Condicion

**CF** : Conj\_Formula

### Definición 3.5 – Requerimiento

**CA:** {nombre / nombre es un Nomb\_At}, es el conjunto de nombres de atributos seleccionados en el requerimiento, tienen como condición que éstos deben ser de las entidades participantes.

**nomb\_At:** {nombre / nombre  $\equiv$  a.nombre, a es un ATRIBUTO  $\wedge$  ( $\forall$  a,  $\exists$  e  $\in$  CE / a  $\in$  e.CA)}, es el nombre de algún atributo de las entidades participantes en el requerimiento

### Definición 3.6 – CA, conj. de atrib. seleccionados en el requerimiento

**CAF:**  $\{n_i \in \text{Conj\_Formula} / n_i \text{ es un Nomb\_At}\}$ , son los nombres de atributos que participan de las fórmulas.

**Conj\_Formula:**  $\{f_i / f_i \text{ es una FORMULA } \forall i\}$ , son las fórmulas que participan del requerimiento.

**FORMULA :**  $\langle \text{OPF (nombre), nombre es un Nomb\_At} \rangle \mid$   
 $\langle \text{OPF (calc), calc es un CALCULO} \rangle \mid$   
 $\langle \text{CALCULO}(n_{i_1}, \dots, n_{i_n}), \forall i / i \geq 1, n_{i_j} \text{ es un Nomb\_At} \rangle$

**OPF:**  $\{\text{COUNT} \mid \text{SUM} \mid \text{MIN} \mid \text{MAX} \mid \text{AVG}\}$

**CALCULO:** Operaciones que se pueden realizar entre dos o más atributos, o atributos y constantes como por ejemplo  $a_1 + a_2$  ó  $a_1 * 10$ , los atributos que participan deben pertenecer a las entidades participantes del requerimiento.

Definición 3.7 – CAF, conj. de atrib. seleccionados en las fórmulas y fórmula

**CAC:**  $\{n_i \in \text{Conj\_Condicion} / n_i \text{ es un Nomb\_At}\}$ , es el conjunto de nombres de atributos, que intervienen en las condiciones.

**Conj\_Condicion:**  $\{c / c \in \text{CONDICION} \wedge (\forall \text{exp}_i \in c, \text{ si } \text{exp}_i \equiv \text{ATRIBUTO.nombre} \Rightarrow \text{exp}_i \in \text{e.CA} / \text{e} \in \text{CE})\}$ , es el conjunto de condiciones que participa en el requerimiento, pueden haber o no. Si en estas condiciones participan nombres de atributos, éstos deben formar parte de las entidades participantes

**CONDICION**  $\equiv \langle \text{exp}_1 \text{ OP } \text{exp}_2 \rangle \mid \langle \text{OP } \text{exp}_1, \text{ OP} \in \{ \forall, \exists \} \rangle$

**EXPRESIÓN**  $\equiv \{ \text{exp} \mid \neg \text{exp} \}$

**Exp**  $\equiv \{ \text{ATRIBUTO.nombre} \mid \text{ConjValores} \mid \text{CONDICION} \}$

**OP**  $\equiv \{ < \mid > \mid = \mid <> \mid \in \mid \geq \mid \notin \mid \leq \mid \vee \mid \wedge \mid \forall \mid \exists \}$

Definición 3.8– CAC, conj. de atrib. seleccionados en las condiciones y condición



## Definición de Algoritmos

Una vez relevados y homogeneizados los datos de la Metadata Genérica, la Base de Datos y los Requerimientos, se está en condición de interoperar entre estos, para obtener la definición de la Metadata Particular.

Como se mencionó anteriormente, existen dos posibles ambientes en los cuáles aplicar la metodología, los que se diferencian por la existencia, ó no, de la Base de Datos.

A continuación se detallan los algoritmos a aplicar según el ambiente en que se trabaje.

### 3.1 Existe un esquema de Base de Datos para la realidad particular

En este caso el procedimiento a seguir es el siguiente:

- ◆ Obtener una “Super Metadata” (SMG) al realizar la unión entre la MG y la BD
- ◆ A la SMG se la intersecta con el conjunto de requerimientos obteniendo así la MP
- ◆ En el caso de tener requerimientos que no se puedan satisfacer en la BD el algoritmo dará un error de forma que se modifique la BD o alguno de los requerimientos.
- ◆ Se realiza una recorrida entre las entidades, que han sido incorporadas a la MP y se agregan las entidades asociadas a estas, consultando para esto las relaciones, según distintos criterios de alcance.

#### 3.1.1 $SMG = MG \cup BD$

Se realiza la unión de los elementos de la metadata genérica con los elementos de la base de datos. (Fig. 3.1)

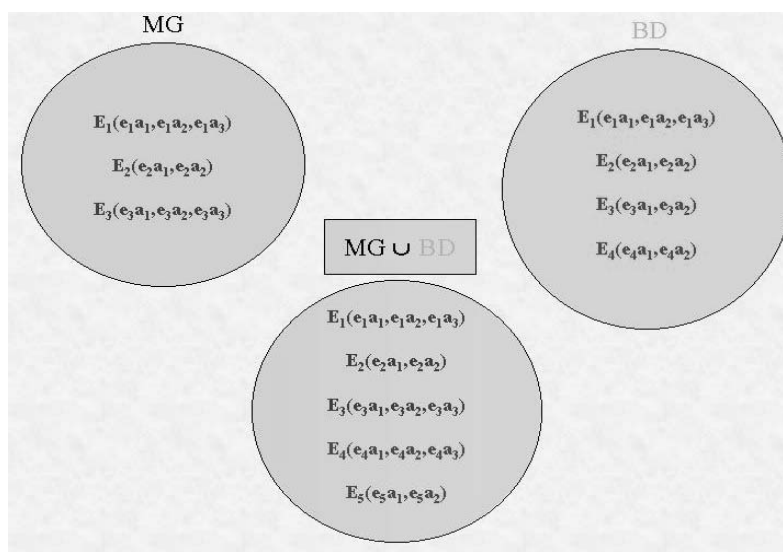


Fig. 3.1 – Obtención de SMG

En la figura se puede observar, el resultado del algoritmo al ejecutarse la unión de los elementos de MG y BD. En esta instancia, es el usuario que a través de la definición de correspondencias semánticas, entre los elementos de MG y BD, permite identificar aquellos elementos comunes y no comunes a ambas. Es así que va realizando la recorrida entre las entidades de la Base y verificando la existencia de las mismas en la MG. De esta forma al encontrar la entidad  $E_1$ , no encuentra ninguna que corresponda semánticamente con ella en MG, y la incorpora a SMG, como  $E_4$ . Luego identifica la correspondencia entre  $E_2$  de BD y  $E_2$  de MG, y entre sus atributos, por lo que lo único que hace el algoritmo es marcar  $E_2$  con origen en BD. Al proceder con  $E_3$ , reconoce la correspondencia semántica con  $E_3$ , pero al tener menos atributos que la  $E_3$  de MG, el atributo  $e_3a_3$  queda marcado con origen en MG, mientras que los otros con origen en BD. Finalmente la entidad  $E_4$ , que no tiene correspondencia en MG, es agregada tal cuál está en BD. De igual forma se procede recorriendo las relaciones de BD, agregándolas a MG, junto con sus atributos, en caso de que corresponda.

Este algoritmo (3.1), procede de la siguiente forma, parte de MG, y va recorriendo el conjunto de entidades pertenecientes a BD, si la entidad existe en MG, marca a la que está en MG, como perteneciente a la BD (origen = BD). Para cada entidad revisa si los atributos que están en la base, están en MG, sino están los agrega a MG. En caso de que la entidad no exista en MG, la agrega completamente.

```

Para cada e ENTIDAD / e ∈ BD hacer:
  Si existe_semántica(e, MG.CE) =>
    e.origen = BD
    Para cada ai ∈ e.CA hacer:
      Si existe_semántica(ai, eMG.CA) =>
        ai.origen = BD
      Si No =>
        Agregar (ai, eMG.CA)
      Fin si
    Fin Para
  Si no =>
    Agregar (e, MG.CE)
  Fin si
Fin Para
Para cada r RELACION / r ∈ BD hacer:
  Si not(existe_semántica(r, MG.CR) =>
    Agregar (r, MG.CR)
  Fin Para
  
```

Algoritmo 3.1 –  $MG \cup BD$

**3.1.2 MP =  $SMG \cap REQ$**

En la fig. 3.2 se muestra la intersección de la SMG obtenida con el algoritmo anterior y los requerimientos.

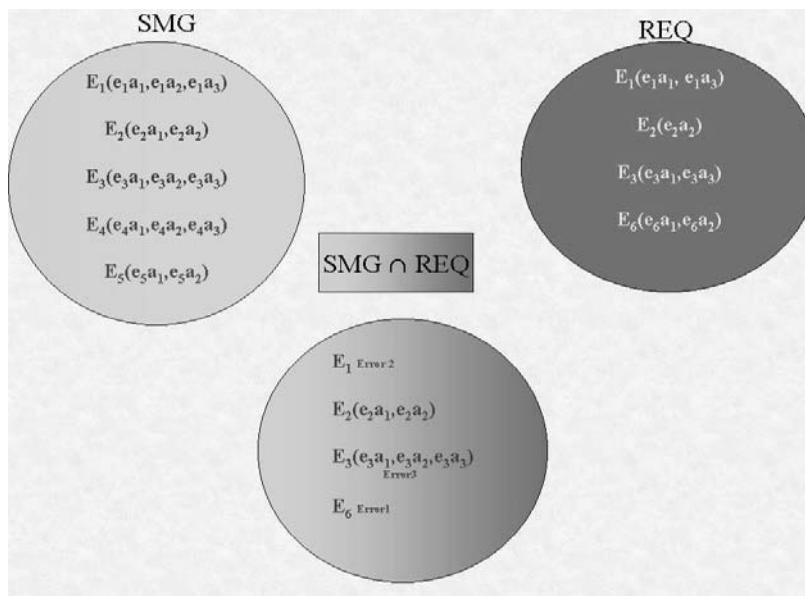


Fig. 3.2 – Obtención de MP

En la figura se abrevió la notación de los requerimientos, considerándolos solo como un grupo de entidades y atributos, sin distinguir que rol cumplen los atributos dentro del requerimiento, es decir si pertenecen al conjunto de selección, al de fórmulas, o al de condiciones.

Se parte de los requerimientos y se busca la pertenencia de las entidades y atributos en SMG.

Si la entidad a la que hace referencia el requerimiento, está en SMG, pero su origen es MG (es decir, el algoritmo anterior, no la marcó como origen en BD), el algoritmo devuelve un error (Error 2), que debe ser interpretado, como que el requerimiento no puede ser satisfecho, contra la OLTP actual. En caso de existir, la entidad y los atributos, se agrega la entidad a MP, tal cual está en SMG. El otro caso es cuando existe la entidad, pero hay algún atributo que no es de BD, en esa situación se produce otro error (Error 3), ya que el requerimiento tampoco puede ser satisfecho.

Finalmente se encuentra en los requerimientos la Entidad  $E_6$ , que no pertenece a la SMG, produciéndose un error (Error 1).

En éste algoritmo (3.2), se recorren todos los requerimientos existentes, de forma de seleccionar de la SMG aquellos elementos participantes. No sólo se seleccionan los elementos participantes en los requerimientos, sino que además se eligen aquellos elementos de la SMG que están relacionados con los ya seleccionados, como por ejemplo, todos los atributos de una entidad, todas las entidades participantes en alguna relación con la entidad seleccionada, así como las relaciones. También se logra advertir tempranamente cuáles requerimientos podrían quedar insatisfechos, si no se altera la BD.

```

Para cada r requerimiento hacer:
  Para cada e ∈ r.CE hacer:
    e' = buscar_semántica(e, SMG.CE)
    Si e' = null =>
      Error 1
    Si no
      Si e'.origen <> 'BD' =>
        Error 2
      Si no
        Para cada ai ∈ r.CAT / ai ∈ e hacer:
          Si ai.origen <> 'BD' =>
            Error 3
          Fin Para
        Fin si
      Si e'.tipo_ent <> ∅ =>
        Para cada ei ∈ e'.tipo_ent hacer:
          Agregar(ei, MP.CE);
        Fin para
      Fin si
    Agregar(e', MP.CE)
  Fin si
Fin Para
Para cada e entidad / e ∈ MP hacer:
  DFS(e, SMG)
Fin Para

```

Algoritmo 3.2 –  $SMG \cap Req$

**DFS** - La idea, tras la recorrida del DFS, es poder obtener, una MP, más completa, que permita satisfacer requerimientos posteriores, que no surgieron en la etapa de análisis inicial. Por ese motivo, se recorre un esquema en el cuál se representan las entidades y las relaciones existentes entre ellas, según lo relevado en SMG (Fig.3.3)

Se parte de las entidades de MP, y se va recorriendo el esquema.

El DFS, puede ser parametrizado de forma de recabar tantos elementos como el usuario defina de su interés. Así podría definir un conjunto mínimo, en el que sólo recabe aquellas relaciones, de las cuales participan entidades que ya están en MP. De igual forma podría avanzar relevando todas las entidades que están en relación con alguna entidad de MP, pero que no pertenecen a ella, y así sucesivamente, se podría incrementar el nivel de recorrida deseado, hasta llegar a un máximo en el que se recorriera todo el esquema.

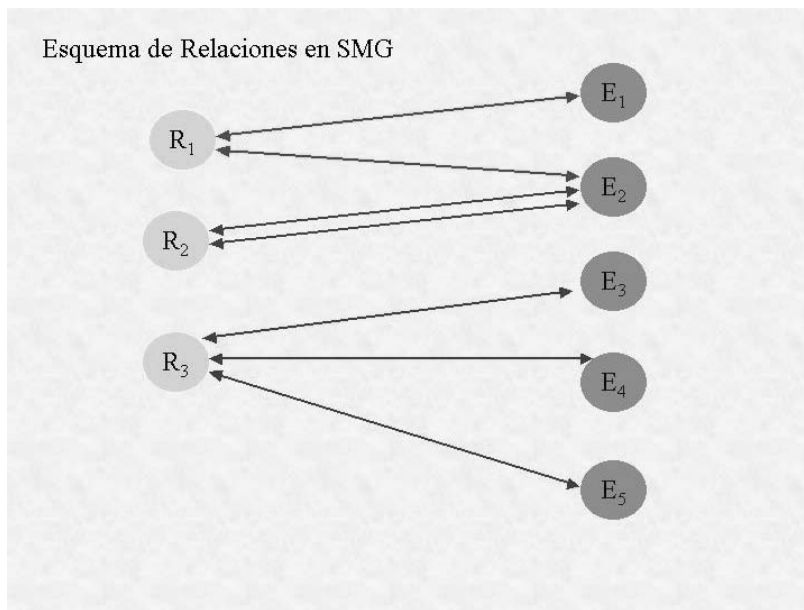


Fig. 3.3 - SMG

**3.2 No existe un esquema de Base de Datos para la realidad particular.**

En este caso sólo se cruzarán la MG con los requerimientos agregando así a la MG aquellas entidades y atributos que sean necesarios. Lo que hace este algoritmo es, quedarse con los elementos de la MG existentes que participan en los requerimientos y agregar aquellos elementos que la MG original no tiene pero son necesarios. (Fig. 3.4)

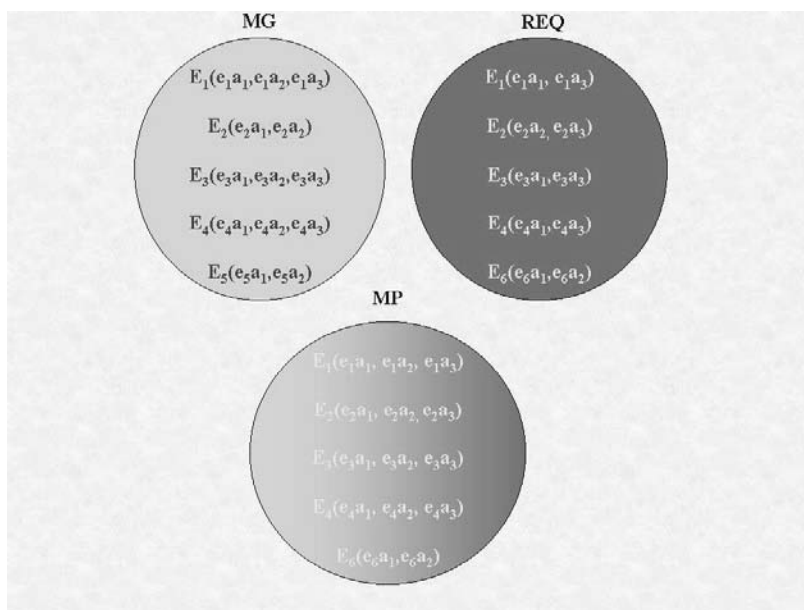


Fig. 3.4 – No existe la OLTP

En la figura se observan los conjuntos formados por la Metadata Genérica y los Requerimientos, y la aplicación del algoritmo para determinar MP. La idea es que en MP, queden todas las entidades con sus atributos tal cual están en los requerimientos. A estas se le agregan los atributos que están en MG, para esas entidades. También se agregarán si existieran, las entidades y relaciones que contemplen los intereses y criterios definidos por el usuario, al aplicar el algoritmo DFS, de la misma forma que en el caso anterior.

La idea de este algoritmo (3.3) es obtener una Metadata particular a partir de los requerimientos y de la Metadata general. La misma se formará de la siguiente manera:

- ◆ Dados los nombres de entidades que figuran en el requerimiento, para cada uno de ellos se realiza una búsqueda semántica en MG, si existe una entidad similar se agregará a MP la entidad encontrada. En caso de no existir se creará una entidad con los atributos que figuren en el requerimiento para la misma. Para el caso en que se encuentre la entidad, a ésta se le agregarán los atributos que figuren en el requerimiento y no existan en la misma.
- ◆ Finalmente, se hará una recorrida (DFS) por las relaciones existentes en la MG, partiendo de las entidades definidas en MP y se agregarán las entidades faltantes en la MP que contemplen los criterios definidos por el usuario para la búsqueda, así como las relaciones.

```

Para cada r requerimiento hacer:
  Para cada e ∈ r.CE hacer:
    Crear_entidad(e')
    e' = Buscar_Semántica(e, MG.CE)
    Para cada a ∈ r.CAT / a ∈ e hacer:
      a' = Buscar_semántica(a, e')
      Si a' = null =>
        Agregar(a, e')
      Fin si
    Fin para
  Agregar(e', MP.CE)
Fin para
Fin Para
Para cada e entidad / e ∈ MP hacer:
  DFS(e, MG)
FinPara
  
```

Algoritmo 3.3 – MG y Req. sin OLTP

## Consideraciones generales

Hay que notar que estos algoritmos son sugerencias para el usuario, por lo que él define si cada atributo que quedó en la MP es de su interés o no, pudiendo agregar o quitar los elementos necesarios.

- El algoritmo *DFS* utilizado, busca en la estructura original e inserta en la estructura MP
- La MP antes del DFS tiene sólo las entidades (con todos sus atributos) de los requerimientos.
- Cuando se hacen referencia a los nombres de atributos en los requerimientos, estos vienen prefijados con el nombre de la entidad a la cual pertenecen
- **agregar**, procedimiento que agrega el elemento al conjunto correspondiente, no inserta elementos repetidos
  - agregar* : Entidad x Conj\_Entidades -> Conj\_Entidades
  - agregar* : Atributo x Conj\_Atributos -> Conj\_Atributos
  - agregar* : Relacion x Conj\_Relaciones-> Conj\_Relaciones
- **existe\_semántica** retorna true o false, dependiendo si se encontró un elemento equivalente al recibido como parámetro en la estructura destino
  - existe\_semántica* : Entidad x Conj\_Entidades -> Boolean
  - existe\_semántica* : Atributo x Conj\_Atributos -> Boolean
  - existe\_semántica* : Relacion x Conj\_Relaciones-> Boolean

- **buscar\_semántica** retorna un elemento del tipo buscado en caso de encontrarlo en la estructura, en caso contrario devuelve un elemento vacío
  - buscar\_semantica* : Entidad x Conj\_Entidades -> Entidad
  - buscar\_semantica* : Atributo x Conj\_Atributos -> Atributo
  - buscar\_semantica* : Relacion x Conj\_Relaciones-> Relacion

## 4. Métodos para el pasaje a CMDM

Como se había mencionado en capítulos anteriores, la idea es a partir de una Metadata Genérica y un análisis de requerimientos de una realidad particular llegar a definir un DW, para ello puede existir de antemano una OLTP o no. En el capítulo anterior se mostró la manera de como obtener una Metadata Particular a partir de dichos elementos. Con la MP obtenida junto al Análisis de requerimientos se debe poder obtener un diseño conceptual multidimensional (CMDM).

Los principales elementos que componen el CMDM son:

- ◆ Niveles con sus atributos
- ◆ Dimensiones en las cuales pueden participar uno o varios niveles jerarquizados entre sí.
- ◆ Relaciones Dimensionales
- ◆ Cubos
- ◆ Restricciones de Integridad que pueden ser Intra-nivel, Intra-dimensión, Intra-Relación Dimensional y Del esquema.

Dada la experiencia desarrollada al aplicar el pasaje a CMDM en el caso de estudio (Realidad de Pesca del INAPE), se llega a formular los lineamientos que automaticen el pasaje mencionado como un primer acercamiento para la obtención de una metodología.

Intuitivamente, lo que se realiza es ir identificando elementos de la Metadata, con los correspondientes en el CMDM, es así que primero surgen los Niveles, luego las Dimensiones que estos conforman, para finalmente definir las Relaciones Dimensionales que darán lugar a los posibles cubos.

Para obtener estos elementos haciendo uso de la MP y de los requerimientos se seguirán los siguientes pasos, luego de la explicación de los mismos, se dará un pseudo-código que facilite el llevar a cabo los mismos. También se utilizarán primitivas, que permitan interactuar entre la Metadata y el CMDM.

Es importante para comprender estos pasos estar familiarizado con la notación y definiciones brindadas en el capítulo anterior.

### Identificación de Niveles

- Se procede a crear los Niveles como imagen de las entidades definidas en la MP (entidades completas, con sus atributos), para ello se utilizará la primitiva *Crear\_Nivel*, que tiene como entrada una entidad y como salida un nivel del CMDM.

```

Para cada  $e \in MP.CE$  hacer
     $N := Crear\_Nivel(e)$ 
Fin Para
  
```

Algoritmo 4.1 – Crear niveles para CMDM

**Identificación de atributos de interés**

- Recorriendo los atributos seleccionados en los requerimientos, seleccionar aquellos que representen una fecha, a partir de esta selección se dará la opción de crear dimensiones con estos atributos. En caso de que el usuario aceptara, se utilizará la primitiva *Crear\_Dimensión*, la cuál creará una dimensión con un único nivel en el que conste como atributo, el seleccionado. A continuación se procede a eliminar el atributo de su nivel original, para ello se utilizará la primitiva *Borrar\_Atributo*, que recibe como entrada un Nivel y un Atributo, y devuelve el Nivel modificado.

```

Para cada r requerimiento hacer
  Para cada nomb_at ∈ r.CAS hacer
    Si Tipo_Dato(nomb_at,MP)=Date =>
      Si el usuario acepta =>
        N := Buscar_Nivel(nomb_at)
        N' := Crear_Nivel(nomb_at)
        D:=Crear_Dimensión()
        Agregar_Nivel (D, N')
        Borrar_Atributo(N,nomb_at)
      Fin Si
    Fin Si
  Fin Para
Fin Para

```

Algoritmo 4.2 – Crear Dimensiones del tipo fecha

- Para cada atributo participante de Fórmulas y/o funciones de agregación definidos en los requerimientos, se sugerirá la creación de medidas a integrar el CMDM. Igualmente que en el paso anterior y bajo la aprobación del usuario, se procederá a crear la dimensión correspondiente, así como borrar el atributo de su origen.

```

Para cada r requerimiento hacer
  Para cada nomb_at ∈ r.CAS hacer
    Si Pertenece_Formula(nomb_at,r) =>
      Si el usuario acepta =>
        N := Buscar_Nivel(nomb_at)
        N' := Crear_Nivel(nomb_at)
        D:=Crear_Dimensión()
        Agregar_Nivel (D, N')
        Borrar_Atributo(N,nomb_at)
      Fin Si
    Fin Si
  Fin Para
Fin Para

```

Algoritmo 4.3 – Identificar medidas



**Identificar Dimensiones**

- Para cada entidad de MP, en la cual el atributo tipo\_ent sea no nulo, crear una Dimensión con este Nivel y los Niveles asociados a las entidades de la agregación.

```

Para cada e ∈ MP.CE hacer
  Si e.tipo_ent <> ∅ =>
    D:= Crear_Dimension()
    N:= Buscar_Nivel(e)
    Agregar_Nivel(D,N)
    Para cada ei ∈ e.tipo_ent hacer:
      N:= Buscar_Nivel(e)
      Agregar_Nivel(D,N)
    Fin para
  Fin Si
Fin Para

```

**Algoritmo 4.4 – Dimensiones a partir de agregaciones**

- Recorriendo las relaciones definidas en la MP, se sugiere crear una Dimensión agrupando los Niveles correspondientes a las entidades participantes de cada relación, de forma de respetar la cardinalidad (1 a N, entre niveles, como se define en el CMDM). La identificación de dimensiones se realiza bajo una fuerte interacción del usuario, de modo de no crear una única dimensión que contenga todas las entidades relacionadas. Se trata de identificar cuales niveles realmente integran una dimensión y que niveles se asocian con simples codigueras que podrían participar de varias relaciones dimensionales formando así una dimensión propia.

```

Para cada r ∈ MP.CR hacer
  Si Relacion_Valida( r ) =>
    D:= Crear_Dimension()
    Para cada e ∈ r.ent_participantes hacer
      N:= Buscar_Nivel(e)
      Agregar_Nivel(D,N)
    Fin para
  Fin Si
Fin Para

```

**Algoritmo 4.5 – Dimensiones a partir de niveles relacionados**

- Volviendo a los requerimientos, se determinan agrupaciones de interés, que llevan a crear nuevos niveles dentro de una dimensión. Para ello, se analizan los atributos del conjunto CAS, en aquellos requerimientos en los que existan funciones de agregación. Para cada uno de estos atributos se sugiere la posibilidad de crear el nivel correspondiente en la dimensión a la cuál pertenece. Lo que se va a hacer es crear un nuevo Nivel a partir de ese atributo y asociarlo a la Dimensión.

```

Para cada r requerimiento hacer
  Si Existe_Formula(r) =>
    Para cada nomb_at ∈ r.CAS hacer
      Si not (Pertenece_Formula(nomb_at,r)) =>
        Si el usuario acepta =>
          N := Buscar_Nivel(nomb_at)
          Borrar_Atributo(N,nomb_at)
          D:= Buscar_Dimension(N)
          N:=Crear_Nivel(nomb_at)
          Agregar_Nivel(D,N)
        Fin Si
      Fin Si
    Fin Para
  Fin Si
Fin Para

```

Algoritmo 4.6 – Dimensiones en base a los requerimientos

### Crterios a tomar por el Diseñador

Llegado este punto es necesario un refinamiento que dependerá de la realidad particular y de los criterios que adopte el diseñador manteniendo su estilo propio. Estas decisiones son por lo tanto subjetivas y variables, por lo que se sugieren pasos que encausen este refinamiento.

#### Decisiones del Diseñador:

- ◆ Eliminar Redundancias
- ◆ Establecer Jerarquías
- ◆ Eliminar Atributos que no son de interés
- ◆ Identificar Relaciones Dimensionales

### Eliminar redundancias

- Una vez definidas las dimensiones, el usuario se encuentra con atributos en los niveles que en realidad identifican otro nivel que a su vez participa en otra dimensión, es bueno que se eliminen del nivel dicho atributo, para así no tener un modelo redundante, donde se repitan atributos. Se verá que este vínculo que en una BD es una relación se reflejará a través de las relaciones dimensionales a definir.

### Establecer jerarquías

- Se han identificado dimensiones como por ejemplo la dimensión fechas, en la cual hasta el momento no se han identificado niveles dentro de ella. Es en este momento en que el usuario define como quiere representar esta dimensión y que apertura quiere darle.
- Una instancia que estaría faltando luego de utilizadas la MP y el análisis de requerimientos es utilizar un poco de sentido común. Es decir, analizar las dimensiones y los niveles participantes en ellas y ver que atributos pueden llegar a ser de interés como agrupadores para nuevos requerimientos que aún no estén definidos. Esto se debe a que como es un modelo conceptual hay que tratar que el mismo sea lo más general posible permitiendo así que la incorporación de nuevas consultas no impacte de forma que haya que redefinir prácticamente todo el modelo. Es así que el usuario identificará dentro de los niveles los posibles agrupadores y con ellos creará nuevos niveles en la dimensión.

### Eliminar atributos que no son de interés

- Hasta el momento se han aplicado dentro de lo posible criterios lógicos de obtención de los datos. Pero como ya se mencionó anteriormente hay muchas cosas definidas por el usuario. Es entonces necesario permitirle eliminar aquellos atributos que no son de relevancia para la realidad modelada, y que si existen por ejemplo en las bases fuentes o en la Metadata Genérica.

### Identificar Relaciones Dimensionales

- Para sugerir posibles relaciones dimensionales, se analiza cada dimensión, buscando para cada nivel participante en la dimensión su entidad asociada en la MP. Una vez encontrada, se recorren las relaciones de esta entidad, y se observa en esa dimensión si las mismas están incluidas o no. En caso de encontrar alguna entidad que no esté incluida en la dimensión, buscar la dimensión a la cual corresponda. Estas dos dimensiones son candidatas a integrar la misma relación dimensional.
- Otra forma de integrar dimensiones en la relación dimensional es teniendo en cuenta las dimensiones que se crearon a partir de los atributos que formaban parte del nivel integrante de la dimensión de interés, por ejemplo dimensiones asociadas con las fechas o medidas.
- Se recorren las entidades participantes en los requerimientos, y se identifican las distintas dimensiones a las cuales pertenecen las mismas. Estas dimensiones se sugieren como candidatas para integrar una relación dimensional.

### Definición de Primitivas

A continuación se describe el significado y el alcance de las primitivas utilizadas anteriormente

- **Crear\_Nivel:** Crea un nivel, a partir de una entidad o un atributo. En el caso de recibir una entidad lo crea con todos los atributos de la misma.  
*Crear\_nivel* : Entidad -> Nivel  
*Crear\_Nivel* : Atributo -> Nivel
- **Tipo\_Dato:** Dado un nombre de atributo y una Metadata retorna el tipo del mismo, el atributo se asume prefijado con el nombre de la entidad.  
*Tipo\_Dato* : Nombre\_atributo -> Tipo\_Dato
- **Crear\_Dimension:** Crea una dimensión vacía  
*Crear\_Dimension* : -> Dimension
- **Buscar\_Nivel:** Dada una entidad o un nombre de atributo (prefijado con el nombre de la entidad a la cual pertenece) retorna el primer nivel encontrado que lo incluye.  
*Buscar\_Nivel* : Entidad -> Nivel  
*Buscar\_Nivel* : Nombre\_atributo -> Nivel
- **Borrar\_Atributo:** Dados un nivel y un atributo, borra el atributo de ese nivel  
*Borrar\_Atributo* : Nivel X Nombre\_atributo -> Nivel
- **Pertenece\_Formula:** Dados un nombre de atributo y un requerimiento, busca en el conjunto de fórmulas del requerimiento, retornando True si éste pertenece a alguna fórmula y False en caso contrario  
*Pertenece\_Formula* : Nombre\_Atributo X Requerimiento -> Boolean
- **Agregar\_Nivel:** Dadas una dimensión y un nivel, agrega éste a la dimensión mencionada manteniendo el orden jerárquico con los niveles existentes en ella.  
*Agregar\_Nivel*: Dimension X Nivel -> Dimension
- **Relacion\_Valida:** Dada una relación valida, es decir, que ésta cumpla los requisitos necesarios exigidos por el CMDM para jerarquizar niveles, lo cual implica que la relación sólo tenga dos entidades participantes y que para la primera entidad la cardinalidad mínima sea uno y la cardinalidad máxima sea también uno y para la segunda entidad la cardinalidad mínima sea uno y la cardinalidad máxima sea N (relación 1 a N). Si se cumplen las condiciones anteriores retorna True y en caso contrario False.  
*Relacion\_Valida* : Relacion -> Boolean

- **Existe\_Formula:** Dado un requerimiento retorna True si éste tiene en su definición un conjunto de fórmulas asociado, en caso contrario retorna False.  
*Existe\_Formula* : Requerimiento -> Boolean
- **Buscar\_Dimension:** Dado un nivel, busca en todas las dimensiones del CMDM y retorna la primera encontrada que lo contenga.  
*Buscar\_Dimension* : Nivel -> Dimension

## 4.5. Caso de estudio

En éste capítulo se relatará la realidad en la que se aplicaron los distintos pasos y se ejemplificarán los mismos al trabajar sobre una porción de esta realidad.

Como se mencionó anteriormente el usuario ligado a esta aplicación era el INAPE, sobre su realidad es que se trabajó. Ellos principalmente trabajan en dos áreas relacionadas con el mar, la Pesca y la Investigación. Para este análisis se hizo hincapié en los temas relacionados con la Pesca, dado que es la parte más importante para ellos de analizar, y tienen la necesidad de implantar un mayor control sobre la información recibida por parte de las empresas de pesca.

Para este análisis se recabaron datos de la base de datos que tienen implementada y de algunos de los procedimientos que se realizan. La realidad encontrada se puede resumir de la siguiente forma:

- ◆ Partes de pesca. Se obtiene uno por buque, detallando la información de la captura.
- ◆ Parte mensual. Cada empresa brinda información desglosada por cada uno de los viajes que realizó.
- ◆ Inspecciones. Se realizan esporádicamente y se obtienen los mismos datos que en el Parte de pesca.

Del área de Pesca les interesa confrontar las distintas informaciones recibidas para detectar inconsistencias entre ellas. La idea es obtener resúmenes en períodos móviles, para poder analizar la evolución de la captura.

Además interesan obtener datos estadísticos como por ejemplo:

- ◆ Cantidad capturada por especie y zona en un período de tiempo
- ◆ Cantidad capturada por empresa y barcos en un período de tiempo
- ◆ Análisis de cantidad de pesca permitida contra cantidad capturada en un período de tiempo para una especie dada
- ◆ Cantidad capturada por especie y arte de pesca

Para que este documento no se extienda demasiado, sólo se presentará una parte de todo el análisis realizado, así como la aplicación de los métodos explicados en los capítulos anteriores. Es así que el ejemplo presentado se concentrará en los informes de parte de pesca, dejando de lado el resto de la realidad. El estudio completo se adjunta en los anexos.

Los ejemplos se mostrarán de forma gráfica para que sea más sencilla la comprensión de los mismos.

## Obtención de la Metadata Particular (MP)

Para obtener la misma, se cuenta con la Metadata Genérica, la OLTP y los requerimientos a detallar.

A modo de ejemplo de representación de cómo expresar los requerimientos se muestran dos de ellos a continuación.

- Req. 1 – Cantidad capturada por especie en un período de tiempo

CE = {Especie, Captura, Partes\_Pesca}

**CAT = CAS  $\cup$  CAC**

CAT = {Especie.Cod\_esp\_animal, Especie.Nombre\_comun, Captura.Cant\_capturada, Partes\_Pesca.Fecha\_Llegada, Captura.Nro\_Parte}

CAS = {Especie.Cod\_esp\_animal, Especie.Nombre\_comun, Captura.Cant\_capturada, Captura.Nro\_Parte}

**CAS = CA ∪ CAF**

CA = {Especie.Cod\_esp\_animal, Especie.Nombre\_comun, Captura.Nro\_Parte}

CAF = {Captura.Cant\_capturada}

CAC = {Partes\_Pesca.Fecha\_Llegada}

CF = {SUM(Captura.Cant\_capturada)}

CC= {Partes\_Pesca.Fecha\_Llegada >= param1∧

Partes\_Pesca.Fecha\_Llegada <= param2}

- Req. 2 – Cantidad capturada por empresa y barcos

CE = {Captura, Buques, Empresa\_Armadora, Partes\_Pesca}

**CAT = CAS ∪ CAC**

CAT = {Empresa\_Armadora.Cod\_empr\_arm, Empresa\_Armadora.Nomb\_empr\_arm, Partes\_Pesca.Cod\_Buque, Buques.Nom\_Buque, Captura.Cant\_capturada, Captura.Nro\_Parte}

CAS = {Empresa\_Armadora.Cod\_empr\_arm, Empresa\_Armadora.Nomb\_empr\_arm, Partes\_Pesca.Cod\_Buque, Buques.Nom\_Buque, Captura.Cant\_capturada, Captura.Nro\_Parte}

**CAS = CA ∪ CAF**

CA = {Empresa\_Armadora.Cod\_empr\_arm, Empresa\_Armadora.Nomb\_empr\_arm, Partes\_Pesca.Cod\_Buque, Buques.Nom\_Buque, Captura.Nro\_Parte}

CAF = {Captura.Cant\_capturada}

CF = {SUM(Captura.Cant\_capturada)}

**Nota:** No se explicitan las condiciones de join, para facilitar la lectura, en el conjunto CC, deberían incluirse las mismas.

Continuando con el ejemplo se muestra a continuación la representación gráfica de la MG, referida a la porción de caso de estudio presentado en este capítulo. Esto se muestra en dos figuras, la primera (5.1) muestra todo el contenido de las entidades y la segunda (5.2) como se relacionan entre sí.

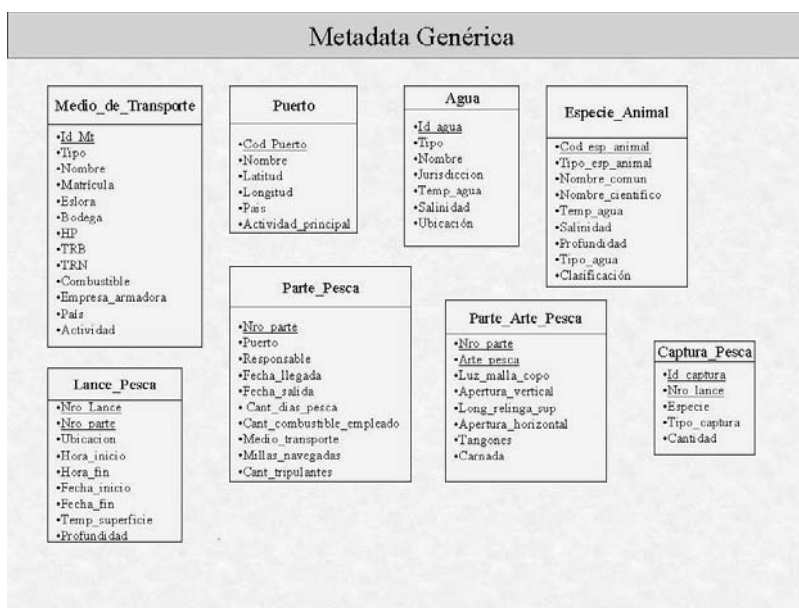


Fig. 5.1 - MG

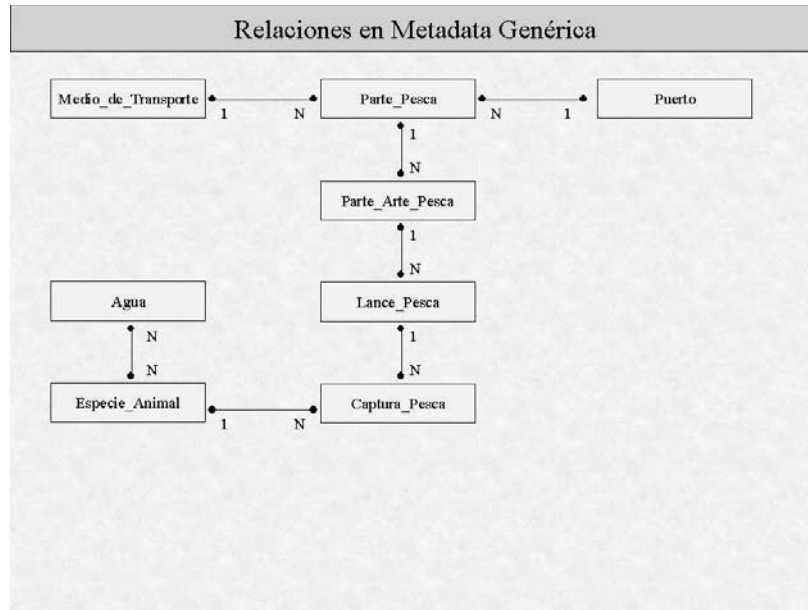


Fig. 5.2 – MG con relaciones

En las próximas dos figuras se muestra la OLTP relevada, de la misma forma que en el caso anterior (Fig. 5.3 y 5.4)

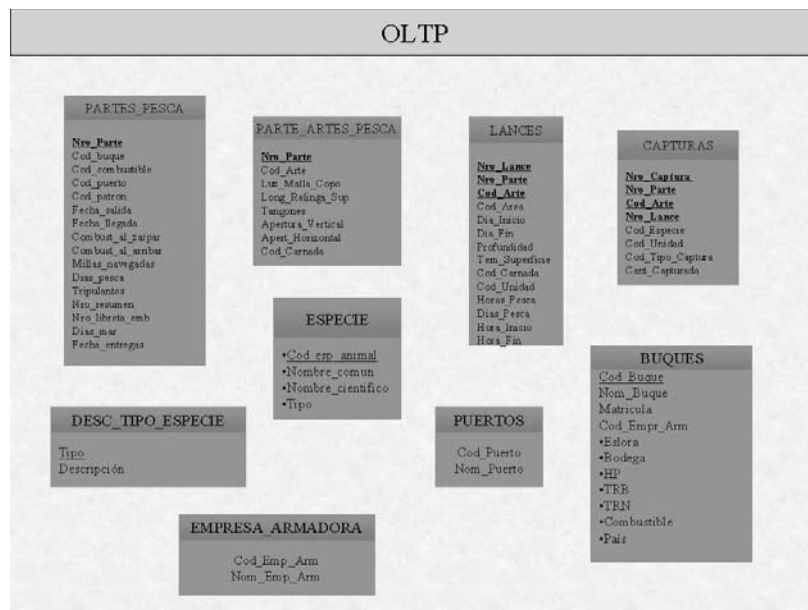


Fig. 5.3 - OLTP

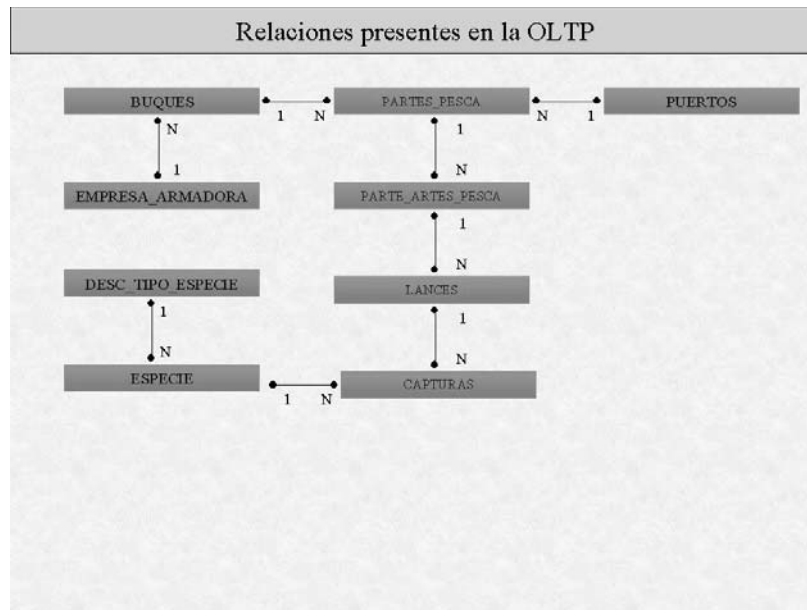


Fig. 5.4 – OLTP con relaciones

Estos esquemas que se han mostrado gráficamente cumplen con las definiciones dadas para representar tanto las metadatos como las OLTP presentados en el capítulo 3. Como para este caso, existe un esquema de BD se van a ejecutar los algoritmos 3.1 y 3.2 para la obtención de la MP. En las figuras que siguen a continuación se muestra el resultado final, es decir primero se realizó la unión entre la MG y la OLTP, luego analizando los requerimientos se seleccionan los elementos de interés para los mismos, generando así la MP. En la figura 5.5 se muestran las entidades completas de la MP y al igual que en los casos anteriores en la figura 5.6 se muestra como se relacionan entre sí. Como se indicaba en los algoritmos, había que distinguir que elementos pertenecen a la OLTP y cuales no, por lo que en la figura los atributos en color negro son los pertenecientes a la OLTP y los en color rojo son los que surgen de la MG.

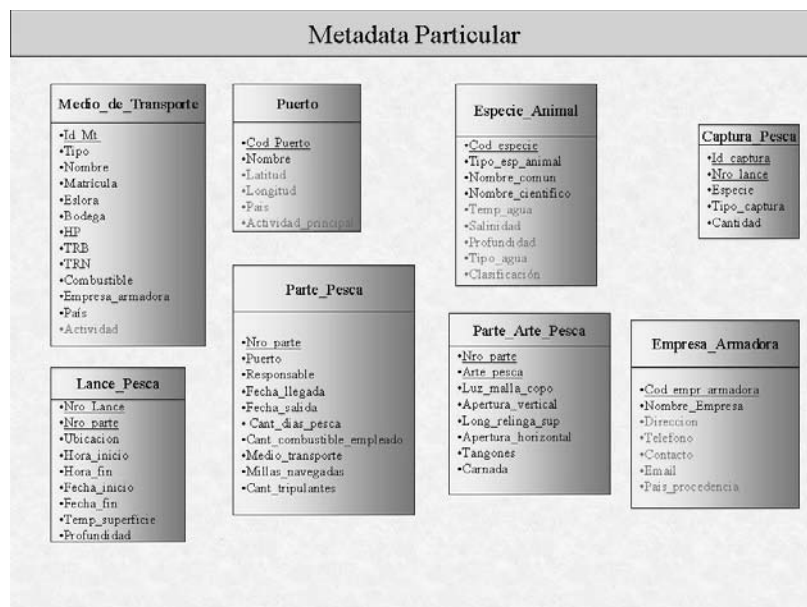


Fig. 5.5 - MP



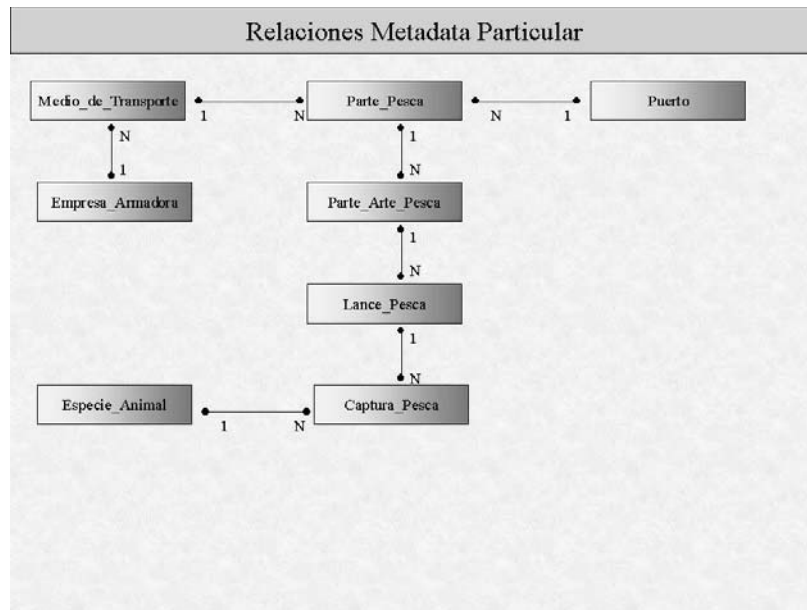


Fig. 5.6 – MP con relaciones

## Obtención de Modelo Conceptual (CMDM)

A continuación se pasará a mostrar de forma gráfica como se obtiene el CMDM aplicando los algoritmos y lineamientos definidos en el capítulo 4.

### Identificación de Niveles

Como primer paso se aplica el algoritmo 4.1 el cual crea para cada entidad perteneciente a la MP un nivel. Sólo se mostrará los niveles que hacen referencia a pesca como se mencionó anteriormente.

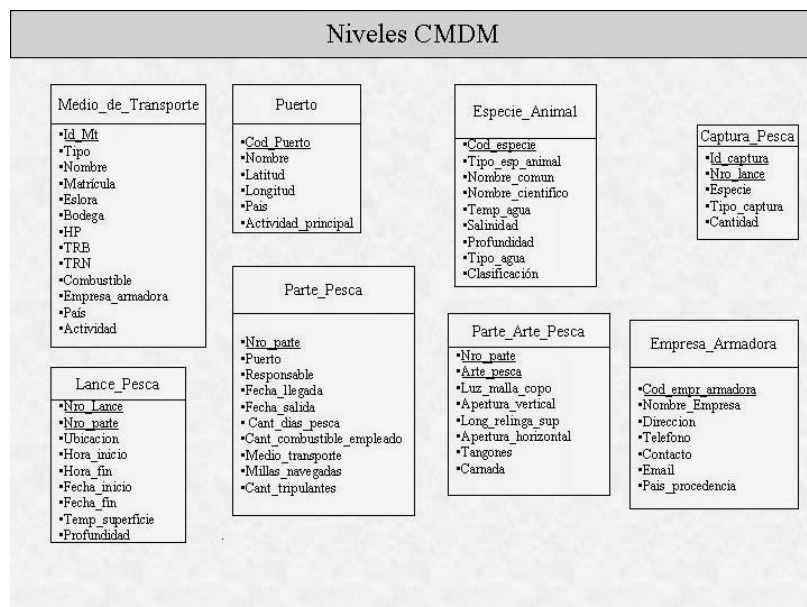


Fig. 5.7 – Identificación de Niveles

### Identificación de atributos de interés

Se procede a utilizar los algoritmos 4.2 y 4.3, los que ofrecen crear dimensiones para aquellos atributos utilizados en los requerimientos de tipo fecha o candidatos a medida.

Del análisis de los requerimientos aplicando los algoritmos 4.2 y 4.3 se identifican como atributos de interés candidatos a dimensiones Parte\_Pesca.Fecha\_llegada y Captura.Cant\_Capturada. Con la aprobación del usuario se pasan a crear las dimensiones respectivas. Con el primer atributo se crea la dimensión Fecha Llegada y con el segundo Cantidad Pesca.

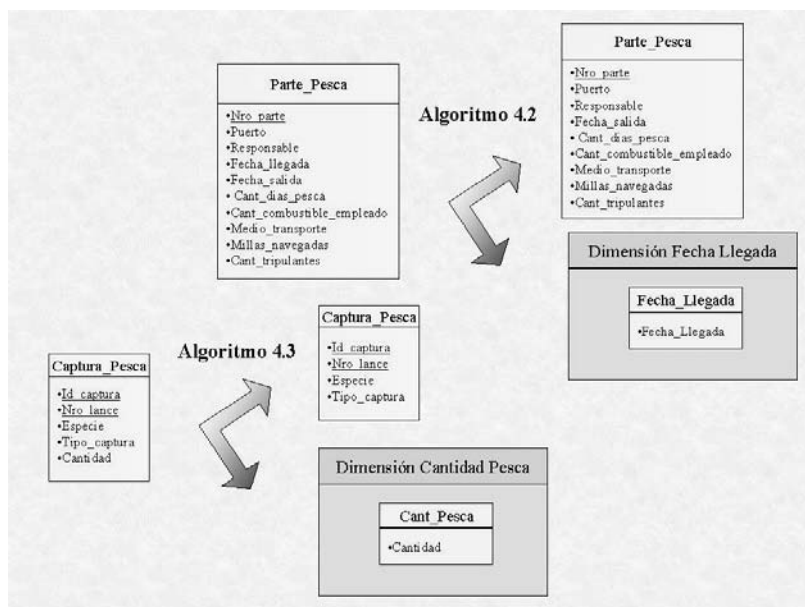


Fig. 5.8 – Identificación de atributos de interés

### Identificar Dimensiones

A continuación se pasará a detallar como se crean las dimensiones en el CMDM, utilizando los algoritmos definidos para ello en el capítulo anterior.

Utilizando el algoritmo 4.5, se recorren las relaciones existentes en la MP y se construyen las dimensiones correspondientes. De la misma forma usando el algoritmo 4.6, se identifica el atributo tipo\_esp\_animal como agrupador de interés dando paso a un nuevo nivel. A continuación se presenta la figura en la que se muestran las dimensiones obtenidas.

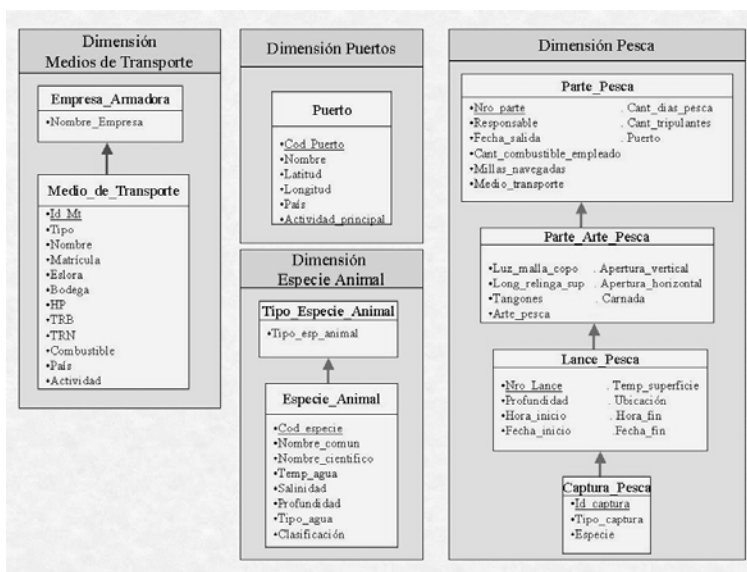


Fig. 5.9 – Identificación de dimensiones

**Criterios a tomar por el Diseñador**

**Eliminar redundancias y establecer jerarquías**

Las redundancias se eliminan observando atributos en los niveles que hacen referencia a niveles en otras dimensiones, como por ejemplo:

- ◆ Partes\_pesca.puerto se elimina ya que hace referencia a Puerto.cod\_puerto de la dimensión puertos.
- ◆ De la misma forma Partes\_pesca.medio\_transporte refiere a Medio\_de\_transporte.id\_mt de la dimensión Medios de transporte
- ◆ De Captura\_pesca.Especie hace referencia a Especie\_animal.cod\_especie de la dimensión Especie Animal

Establecer jerarquías implica una interacción con el usuario que mediante el análisis de las dimensiones existentes resuelve que atributos pueden ser agrupadores de interés para próximos requerimientos creando así nuevos niveles en las dimensiones existentes, como por ejemplo:

- ◆ En la dimensión Medios de Transporte se crean los niveles Actividad y Tipo\_MT que surgen de los atributos del nivel medio\_de\_transporte actividad y tipo.
- ◆ En la dimensión Puertos se crea el nivel Actividad, que surge del atributo actividad\_principal del nivel Puerto.
- ◆ En la dimensión Pesca surge el nivel Arte\_pesca a partir del atributo arte\_pesca del nivel Parte\_Arte\_Pesca.
- ◆ Es en este momento que a la dimensión fecha se la jerarquiza de acuerdo a los intereses del usuario.

**Eliminar atributos que no son de interés**

Como se mencionó en el capítulo anterior, de la metodología aplicada surgen atributos que no son de interés real para la implementación del Data Warehouse, por lo que en este caso de estudio se eliminan los siguientes atributos.

- ◆ Del nivel Medio\_de\_Transporte se eliminan: eslora, bodega, HP, TRB, TRN, combustible y país.
- ◆ Del nivel Puerto se elimina el atributo país.
- ◆ Del nivel Especie\_Animal se eliminan: Temp\_agua, salinidad, profundidad, tipo\_agua y clasificación.
- ◆ Del nivel Lance\_Pesca se eliminan: hora\_inicio, fecha\_inicio, hora\_fin y fecha\_fin.

A continuación se presentan la figura que muestra como quedarían los niveles luego de aplicar estos criterios.

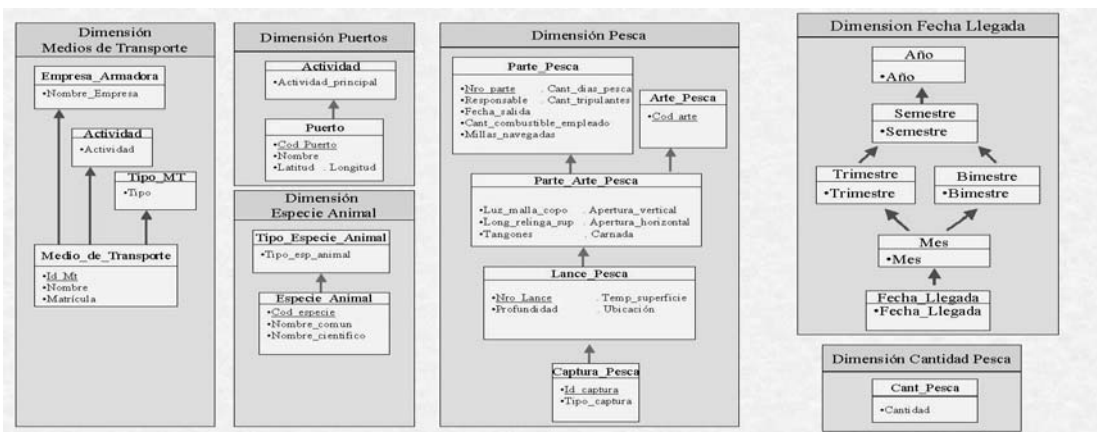


Fig. 5.10 – Eliminación de redundancias y establecimiento de jerarquías

### Identificar Relaciones Dimensionales

De acuerdo a lo expresado en el capítulo anterior surge el siguiente esquema de Relación Dimensional.

Estudiando las relaciones que existen entre las entidades de la MP se seleccionan las siguientes dimensiones como candidatas a integrar una Relación Dimensional:

- ◆ Desde la dimensión Pesca y se selecciona el nivel Parte\_Pesca, este se identifica con la entidad Parte\_Pesca de la MP, se analizan las relaciones de la misma allí se distinguen las entidades relacionadas Puerto, Medio\_de\_Transporte y Parte\_Arte\_Pesca. Al ver los niveles asociados a estas entidades en el CMDM, se observa que las dos primeras pertenecen a otras dimensiones, mientras que la última se encuentra en la misma dimensión. Es así que se proponen como candidatas a integrar una Relación Dimensional, junto a la dimensión Pesca, las dimensiones Medios de Transporte y Puertos
- ◆ Siguiendo con el mismo criterio, se identifica la dimensión Especie Animal como otra candidata, al observar la relación existente entre las entidades Captura\_Pesca y Especie\_Animal de la MP.

Analizando las dimensiones creadas a partir de los atributos que formaban parte de los niveles incluidos en las dimensiones tratadas, es que surgen las siguientes candidatas:

- ◆ Dimensiones Fecha Llegada y Cantidad Pesca

La siguiente figura muestra la integración de estas dimensiones en la Relación Dimensional Pesca, partiendo del supuesto que el usuario aceptara las candidatas propuestas.

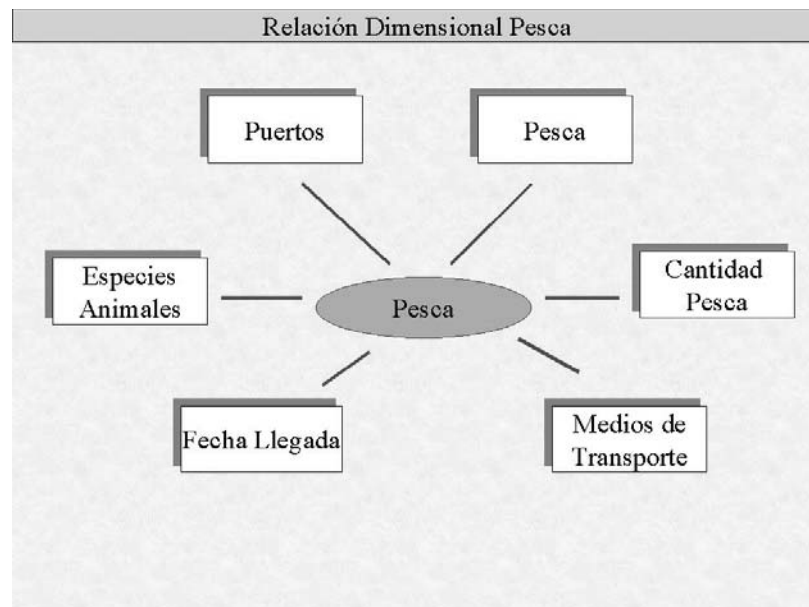


Fig. 5.11 – Relación Dimensional Pesca

Para finalizar la obtención del modelo conceptual, estaría faltando la definición de los cubos a implementar. Para ello se utiliza la metodología descrita en la definición del CMDM. Es así que el cubo se integra con los gránulos de cada dimensión participante de la Relación Dimensional y se identifica que dimensión o dimensiones son las que aportan las medidas.

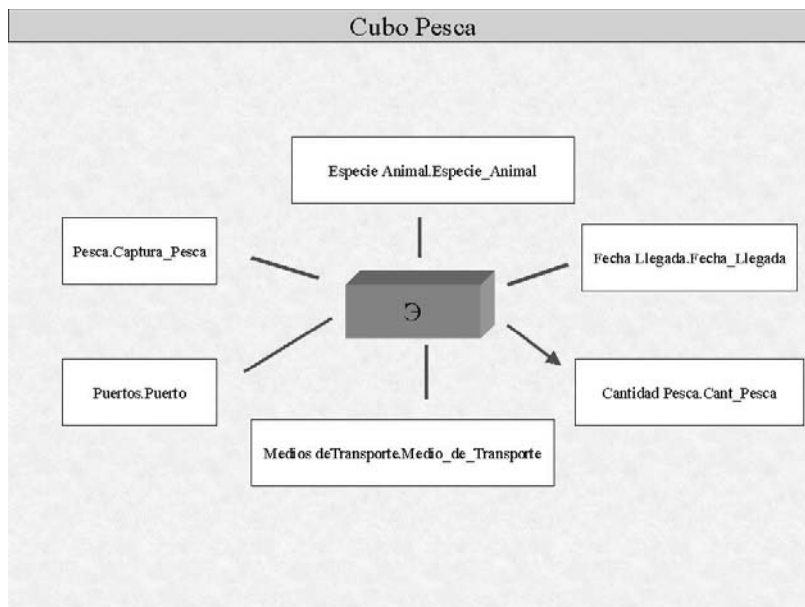


Fig. 5.12 – Cubo Pesca.

## Obtención del Modelo Lógico

Dentro de la evolución del proyecto se realizó la obtención del Modelo Lógico de la realidad planteada a partir del CMDM obtenido.

A continuación se presenta la porción correspondiente a la aplicación de la metodología definida en la Tesis "Diseño Lógico de un DW a partir de un Esquema Conceptual Multidimensional", asociada al caso de estudio que se viene tratando.

### Lineamientos de Diseño

Los lineamientos son información de diseño lógico que complementan al esquema conceptual y permiten al diseñador dar pautas sobre el esquema lógico deseado.

Permiten elegir el estilo de diseño para el DW (snowflakes, estrella, mixto), indicando requerimientos de performance y almacenamiento.

En todos los casos son información adicional ingresada por el diseñador en forma explícita.

En el Anexo III se encuentra el caso de estudio completo con las figuras correspondientes.

### Materialización de Relaciones Dimensionales

En el esquema conceptual una relación dimensional representa un espacio de cubos resultado de cruzar algunos niveles de dimensiones.

La decisión de que cubos se materializan no se toma en el momento de obtener el diseño conceptual. Se entiende por materializar al hecho de precalcular los valores para los cruzamientos de ese cubo y almacenarlos en una relación.

Se decide que cubos materializar atendiendo a los requerimientos de performance y almacenamiento.

En éste caso de estudio, se resuelve materializar un único cubo para la relación dimensional Pesca.

A continuación se muestra la figura del cubo Pesca que surge de la relación dimensional Pesca.

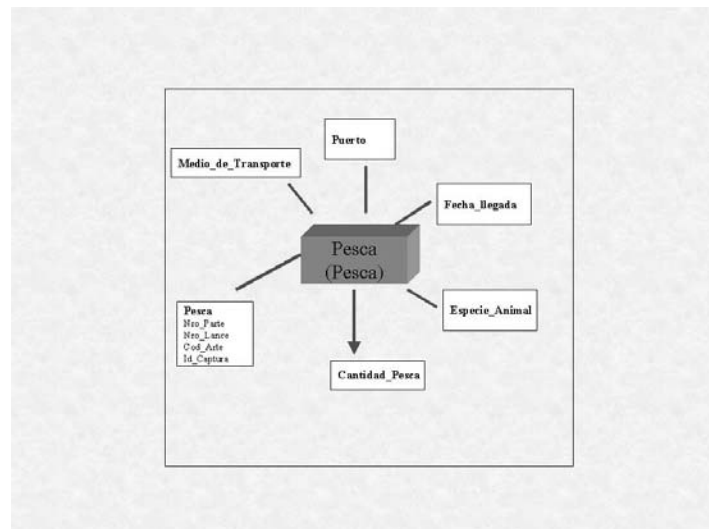


Fig. 5.13 – Cubo Pesca

### Fragmentación de Dimensiones

El diseñador debe indicar el grado de normalización que quiere lograr al traducir cada dimensión al modelo relacional. Este diseño depende de la realidad, se pueden representar algunas dimensiones con un esquema y otras con otro.

En el caso del I.N.A.P.E debido a la poca complejidad de las dimensiones y a la cantidad de datos que se almacenan en el cubo se resuelve utilizar un esquema estrella, es decir desnormalizar todas las dimensiones y mantener fact tables. Además para tomar esta decisión se toma en cuenta la velocidad con la que cambian los datos, en este caso los datos son bastante estáticos, por eso la generación de redundancia no afecta la carga del mismo, en caso que los datos fueran más volátiles se recomendaría tener un esquema normalizado (snowflake).

Según esta metodología la fragmentación se puede representar de forma gráfica como una coloración de niveles. Los niveles de un mismo fragmento se recuadran con el mismo color. Un nivel tiene varios colores si está en más de un fragmento.

En las figuras donde se muestran los mapeos con las Bases Fuentes se observa que cada dimensión tiene un único fragmento, como se mencionó anteriormente.

### Fragmentación de Cubos

El traducir el cubo al modelo relacional puede dar como resultado una o varias tablas (fact tables) dependiendo del grado de fragmentación que se quiera lograr.

Fragmentar horizontalmente una tabla relacional corresponde a construir varias tablas con la misma estructura y dividir las instancias entre ellas. Con esto se logra almacenar juntas las tuplas que se consultan a la vez y tener tablas más pequeñas, lo que hace que las consultas sean más performantes.

En el caso de estudio, dado que el volumen de información almacenada no es muy grande, no se verá afectada la performance de las consultas al tener una única tabla, por lo cual no es necesaria la fragmentación horizontal del cubo.

### Mapeo del CMDM con las bases fuentes

Un mapeo es una función que muestra como se corresponden los objetos del modelo conceptual con la base fuente.

Los mapeos son funciones que asocian a cada item de un objeto del esquema concpetual una expresión de mapeo, construida en base a las tablas y atributos de la fuente.

Se mostrará de forma gráfica como se llega al mapeo del CMDM y las bases fuentes.

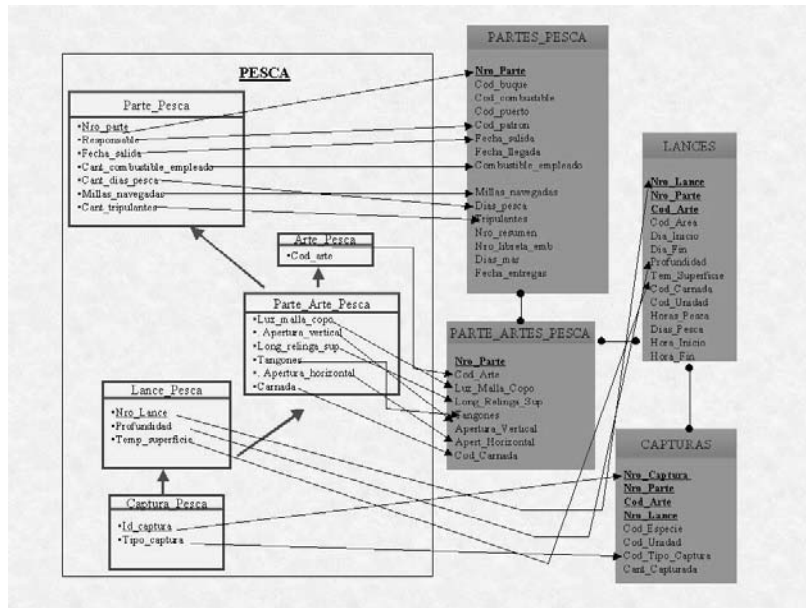


Fig. 5.14 – Mapeo de la Dimensión Pesca

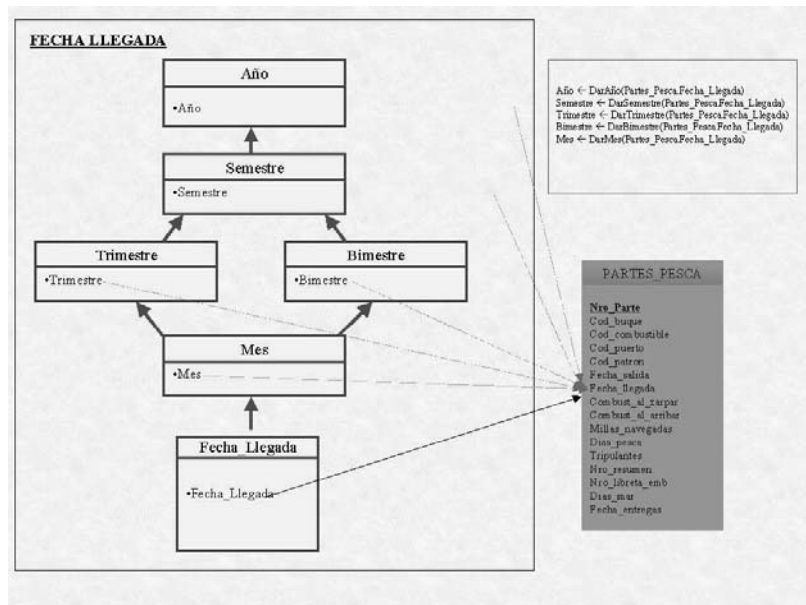


Fig. 5.15 – Mapeo de la Dimensión Fecha Llegada

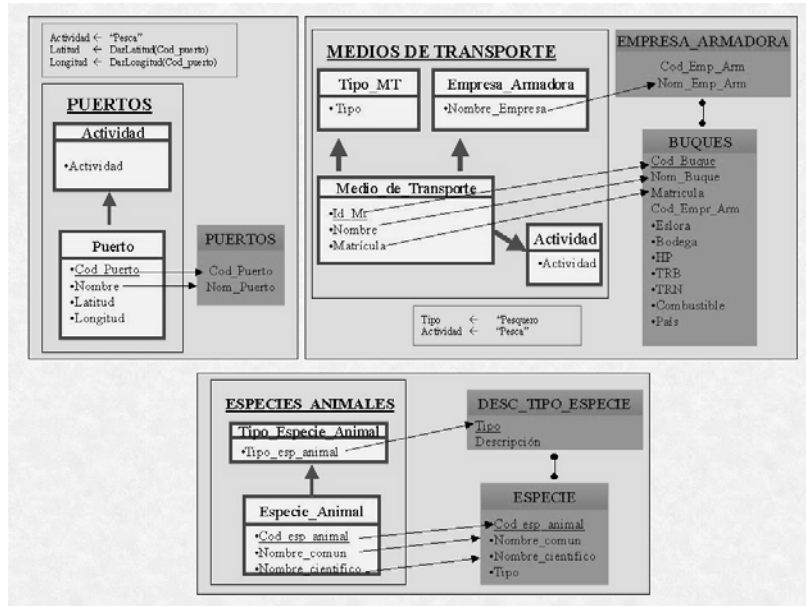


Fig. 5.16 – Mapeo de las Dimensiones Puertos, Medios de Transporte y Especies Animales

A continuación se presenta la forma de obtener el cubo de Pesca a través de las relaciones con las bases fuentes.

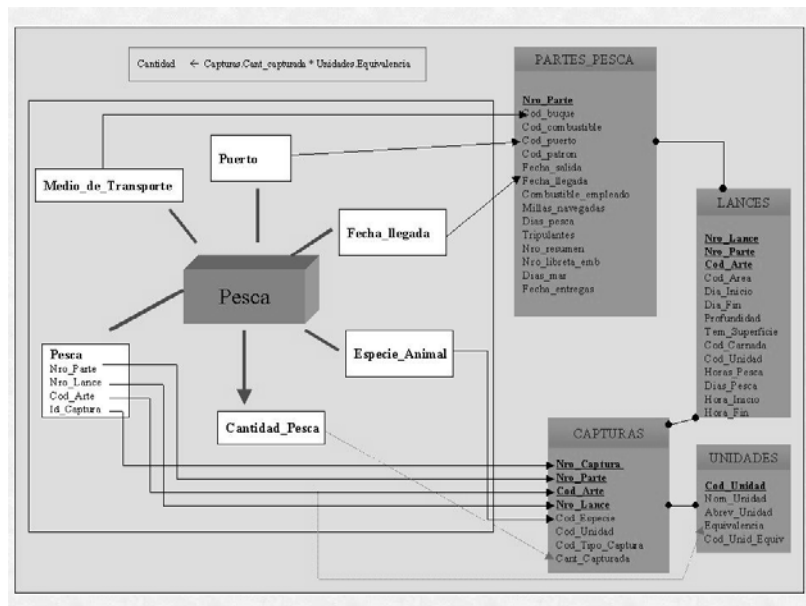


Fig. 5.17 – Mapeo del Cubo Pesca



## Aplicación de Reglas de Transformación

A continuación se presenta la aplicación de las reglas de transformación a las dimensiones y cubos mostrados anteriormente, de forma de obtener la estructura de las tablas en las que se almacenarán los datos extraídos de las bases fuentes.

### Dimensión Puertos

Se quiere desnormalizar la dimensión puertos, la cuál mapea a una única tabla Puertos, a esta dimensión se le agregan los atributos Latitud y Longitud que son calculados y el atributo Actividad, que es un atributo ExternME, que es una constante.

- ♦ Aplicando primero la *regla 3-Calculate* se obtiene:

DWPuertos01(**Cod Puerto**, Nom\_Puerto, Latitud)

- ♦ Aplicando nuevamente la *regla 3-Calculate*, se obtiene:

DWPuertos02(**Cod Puerto**, Nom\_Puerto, Latitud, Longitud)

- ♦ Con la regla 4-New Attribute, se obtiene:

DWPuertos03(**Cod Puerto**, Nom\_Puerto, Latitud, Longitud, Actividad)

- ♦ Aplicando la *regla 2-Rename*, queda:

DWPuertos04(**Cod Puerto**, Nombre, Latitud, Longitud, Actividad)

### Dimensión Medios de Transporte

Se quiere desnormalizar la dimensión puertos, la cuál mapea a dos tablas.

- ♦ Con la *regla 1-Join* entre Buques y Empresa\_Armadora, se obtiene:

DWMedios\_de\_Transporte01(**Cod Emp Arm**, Nom\_Emp\_Arm, **Cod Buque**, Nom\_Buque, Matricula, Cod\_Emp\_Arm2, Eslora, Bodega, HP, TRN, TRB, Combustible, Pais)

- ♦ Con la *regla 2-Rename*, se llega:

DWMedios\_de\_Transporte02(**Cod Emp Arm**, Nombre\_Empresa, **Id Mt**, Nombre, Matricula, Cod\_Emp\_Arm2, Eslora, Bodega, HP, TRN, TRB, Combustible, Pais)

- ♦ Con la *regla 4-New Attribute*, se agregan los atributos Tipo y Actividad, que para esta realidad se presentan como constantes y se obtiene:

DWMedios\_de\_Transporte03(**Cod Emp Arm**, Nombre\_Empresa, **Id Mt**, Nombre, Matricula, Cod\_Emp\_Arm2, Eslora, Bodega, HP, TRN, TRB, Combustible, Pais, Tipo)

DWMedios\_de\_Transporte04(**Cod Emp Arm**, Nombre\_Empresa, **Id Mt**, Nombre, Matricula, Cod\_Emp\_Arm2, Eslora, Bodega, HP, TRN, TRB, Combustible, Pais, Tipo, Actividad)

- ♦ Aplicando la *regla 5-Group*, queda:

DWMedios\_de\_Transporte05(Nombre\_Empresa, **Id Mt**, Nombre, Matricula, Tipo, Actividad)

### Dimensión Especies Animales

Se quiere desnormalizar la dimensión Especies Animales, la cuál mapea a dos tablas, entonces con la *regla 1-Join* entre Desc\_Tipo\_Especie y Especie, se obtiene:

DWEspecies\_Animales01(**Tipo**, Descripcion, **Cod esp animal**, Nombre\_comun, Nombre\_cientifico, Tipo2)

- ♦ Con la *regla 2-Rename*, se obtiene:

DWEspecies\_Animales02(**Tipo esp animal**, Descripcion, **Cod esp animal**, Nombre\_comun, Nombre\_cientifico, Tipo2)

- ♦ Con la *regla 5-Group*, queda:

DWEspecies\_Animales03(**Tipo esp animal**, **Cod esp animal**, Nombre\_comun, Nombre\_cientifico)

- ◆ Se aplica la *regla 7-PrimaryKey*, para que la clave coincida con la de la dimensión:

DWEspecies\_Animales04(Tipo\_esp\_animal, **Cod esp animal**, Nombre\_comun, Nombre\_cientifico)

#### Dimensión Pesca

- ◆ Con la *regla 1-Join*, entre las tablas Partes\_Pesca y Parte\_Artes\_Pesca, se obtiene:

DWPesca01 (**Nro Parte**, Cod\_buque, Cod\_combustible, Cod\_puerto, Cod\_patron, Fecha\_salida, Fecha\_llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, **Nro Parte2**, **Cod Arte**, Luz\_Malla\_Copo, Long\_Relinga\_Sup, Tangones, Apertura\_Vertical, Apert\_Horizontal, Cod\_Carnada)

- ◆ Nuevamente con la *regla 1-Join*, entre DWPesca01 y la tabla Lances, se llega a:

DWPesca02 (**Nro Parte**, Cod\_buque, Cod\_combustible, Cod\_puerto, Cod\_patron, Fecha\_salida, Fecha\_llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, **Nro Parte2**, **Cod Arte**, Luz\_Malla\_Copo, Long\_Relinga\_Sup, Tangones, Apertura\_Vertical, Apert\_Horizontal, Cod\_Carnada, **Nro Lance**, **Nro Parte3**, **Cod Arte2**, Cod\_Area, Dia\_Inicio, Dia\_Fin, Profundidad, Temp\_Superficie, Cod\_Carnada, Cod\_Unidad, Horas\_Pesca, Dias\_Pesca, Hora\_Inicio, Hora\_Fin)

- ◆ Ahora con la *regla 1-Join*, entre DWPesca02 y la tabla Capturas, se obtiene:

DWPesca03 (**Nro Parte**, Cod\_buque, Cod\_combustible, Cod\_puerto, Cod\_patron, Fecha\_salida, Fecha\_llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, **Nro Parte2**, **Cod Arte**, Luz\_Malla\_Copo, Long\_Relinga\_Sup, Tangones, Apertura\_Vertical, Apert\_Horizontal, Cod\_Carnada, **Nro Lance**, **Nro Parte3**, **Cod Arte2**, Cod\_Area, Dia\_Inicio, Dia\_Fin, Profundidad, Temp\_Superficie, Cod\_Carnada, Cod\_Unidad, Horas\_Pesca, Dias\_Pesca, Hora\_Inicio, Hora\_Fin, **Nro Captura**, **Nro Parte4**, **Cod Arte3**, **Nro Lance2**, Cod\_Especie, Cod\_Unidad, Cod\_Tipo\_Captura, Cant\_Capturada)

- ◆ Ahora se renombran los atributos, de acuerdo a la definición utilizada en el CMDM, utilizando la *regla 2-Rename*

DWPesca04 (**Nro Parte**, Cod\_buque, Cod\_combustible, Cod\_puerto, responsable, fecha\_salida, Fecha\_llegada, Cant\_combustible\_employado, Millas\_navegadas, Cant\_Dias\_pesca, Cant\_Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, **Nro Parte2**, **Cod Arte**, Luz\_Malla\_Copo, Long\_Relinga\_Sup, Tangones, Apertura\_Vertical, Apert\_Horizontal, Carnada, **Nro Lance**, **Nro Parte3**, **Cod Arte2**, Cod\_Area, Dia\_Inicio, Dia\_Fin, Profundidad, Temp\_Superficie, Cod\_Carnada, Cod\_Unidad, Horas\_Pesca, Dias\_Pesca, Hora\_Inicio, Hora\_Fin, **Id Captura**, **Nro Parte4**, **Cod Arte3**, **Nro Lance2**, Cod\_Especie, Cod\_Unidad, Tipo\_Captura, Cant\_Capturada)

- ◆ Ahora para eliminar los atributos sobrantes se aplica la *regla 5-Group*

DWPesca05 (**Nro Parte**, Responsable, Fecha\_salida, Millas\_navegadas, Cant\_Dias\_pesca, Cant\_Tripulantes, **Cod Arte**, Luz\_Malla\_Copo, Long\_Relinga\_Sup, Tangones, Apertura\_Vertical, Apert\_Horizontal, Carnada, , **Nro Lance**, Profundidad, Temp\_Superficie, **Id Captura**, Tipo\_Captura, Cant\_Combustible\_employado)

#### Dimensión Fecha Llegada

Esta dimensión se mapea directamente a la tabla Partes\_Pesca a través de varios cálculos.

- ◆ Aplicando la *regla 3-Calculate* sucesivamente se agregan los atributos Mes, Bimestre, Trimestre, Semestre, Año

DWFecha\_Llegada01(**Nro Parte**, Cod\_Buque, Cod\_Combustible, Cod\_Puerto, Cod\_Patron, Fecha\_Salida, Fecha\_Llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, Mes)

DWFecha\_Llegada02(**Nro Parte**, Cod\_Buque, Cod\_Combustible, Cod\_Puerto, Cod\_Patron, Fecha\_Salida, Fecha\_Llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, Mes, Bimestre)

DWFecha\_Llegada03(**Nro Parte**, Cod\_Buque, Cod\_Combustible, Cod\_Puerto, Cod\_Patron, Fecha\_Salida, Fecha\_Llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, Mes, Bimestre, Trimestre)

DWFecha\_Llegada04(**Nro Parte**, Cod\_Buque, Cod\_Combustible, Cod\_Puerto, Cod\_Patron, Fecha\_Salida, Fecha\_Llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, Mes, Bimestre, Trimestre, Semestre)

DWFecha\_Llegada05(**Nro Parte**, Cod\_Buque, Cod\_Combustible, Cod\_Puerto, Cod\_Patron, Fecha\_Salida, Fecha\_Llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, Mes, Bimestre, Trimestre, Semestre, Año)

- ◆ Con la *regla 5-Group* se obtiene:

DWFecha\_Llegada06(Fecha\_Llegada, Mes, Bimestre, Trimestre, Semestre, Año)

- ◆ Con la *regla 7-PrimaryKey* se obtiene:

DWFecha\_Llegada07(**Fecha Llegada**, Mes, Bimestre, Trimestre, Semestre, Año)

### Cubo Pesca

El cubo pesca es una única tabla, por lo que es necesario joinear las tablas Partes\_Pesca, Lances, Capturas y Unidades.

- ◆ Aplicando la *regla 1-Join* entre Partes\_Pesca y Lances se obtiene:

DWCuboPesca01(**Nro Parte**, Cod\_buque, Cod\_combustible, Cod\_puerto, Cod\_patron, Fecha\_salida, Fecha\_llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, **Nro Lance**, **Nro Parte2**, **Cod Arte**, Cod\_Area, Dia\_Inicio, Dia\_Fin, Profundidad, Temp\_Superficie, Cod\_Carnada, Cod\_Unidad, Horas\_Pesca, Dias\_Pesca, Hora\_Inicio, Hora\_Fin)

- ◆ Ahora con la *regla 1-Join*, entre DWCuboPesca01 y la tabla Capturas, se obtiene:

DWCuboPesca02(**Nro Parte**, Cod\_buque, Cod\_combustible, Cod\_puerto, Cod\_patron, Fecha\_salida, Fecha\_llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, **Nro Lance**, **Nro Parte2**, **Cod Arte**, Cod\_Area, Dia\_Inicio, Dia\_Fin, Profundidad, Temp\_Superficie, Cod\_Carnada, Cod\_Unidad, Horas\_Pesca, Dias\_Pesca, Hora\_Inicio, Hora\_Fin, **Nro Captura**, **Nro Parte3**, **Cod Arte2**, **Nro Lance2**, Cod\_Especie, Cod\_Unidad, Cod\_Tipo\_Captura, Cant\_Capturada)

- ◆ Ahora con la *regla 1-Join*, entre DWCuboPesca02 y la tabla Unidades, se obtiene:

DWCuboPesca03(**Nro Parte**, Cod\_buque, Cod\_combustible, Cod\_puerto, Cod\_patron, Fecha\_salida, Fecha\_llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, **Nro Lance**, **Nro Parte2**, **Cod Arte**, Cod\_Area, Dia\_Inicio, Dia\_Fin, Profundidad, Temp\_Superficie, Cod\_Carnada, Cod\_Unidad, Horas\_Pesca, Dias\_Pesca, Hora\_Inicio, Hora\_Fin, **Nro Captura**, **Nro Parte3**, **Cod Arte2**, **Nro Lance2**, Cod\_Especie, Cod\_Unidad, Cod\_Tipo\_Captura, Cant\_Capturada, **Cod Unidad2**, Nom\_unidad, Abrev\_unidad, Equivalencia, Cod\_unid\_equiv)

- ◆ Se aplica la *regla 2-Rename* para que coincidan los nombres con las dimensiones:

DWCuboPesca04(**Nro Parte**, Id\_Mt, Cod\_combustible, Cod\_puerto, Cod\_patron, Fecha\_salida, Fecha\_llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, **Nro Lance**, **Nro Parte2**, **Cod Arte**, Cod\_Area, Dia\_Inicio, Dia\_Fin, Profundidad, Temp\_Superficie, Cod\_Carnada, Cod\_Unidad, Horas\_Pesca, Dias\_Pesca, Hora\_Inicio, Hora\_Fin, **Id Captura**, **Nro Parte3**, **Cod Arte2**, **Nro Lance2**, Cod\_Especie\_Animal, Cod\_Unidad, Cod\_Tipo\_Captura, Cant\_Capturada, **Cod Unidad2**, Nom\_unidad, Abrev\_unidad, Equivalencia, Cod\_unid\_equiv)

- ◆ Se agregan los campos calculados con la *regla 3-Calculate*:

DWCuboPesca05(**Nro Parte**, Id\_Mt, Cod\_combustible, Cod\_puerto, Cod\_patron, Fecha\_salida, Fecha\_llegada, Combustible\_employado, Millas\_navegadas, Dias\_pesca, Tripulantes, Nro\_resumen, Nro\_libreta\_emb, Dias\_mar, Fecha\_entregas, **Nro Lance**, **Nro Parte2**, **Cod Arte**, Cod\_Area, Dia\_Inicio, Dia\_Fin, Profundidad, Temp\_Superficie, Cod\_Carnada, Cod\_Unidad, Horas\_Pesca, Dias\_Pesca, Hora\_Inicio, Hora\_Fin, **Id Captura**, **Nro Parte3**, **Cod Arte2**, **Nro Lance2**, Cod\_Especie\_Animal, Cod\_Unidad, Cod\_Tipo\_Captura, Cant\_Capturada, **Cod Unidad2**, Nom\_unidad, Abrev\_unidad, Equivalencia, Cod\_unid\_equiv, Cantidad\_pesca)

- ♦ Con la *regla 5-Group* se eliminan los atributos que no se usan:

DWCuboPesca06(**Nro Parte**, Id\_Mt, Cod\_puerto, Fecha\_llegada, **Nro Lance**, **Cod Arte**, **Id Captura**, Cod\_Especie\_Animal, Cantidad\_pesca)

- ♦ Se ajusta la clave aplicando la *regla 7-PrimaryKey*.

DWCuboPesca07(**Nro Parte**, **Id Mt**, **Cod puerto**, **Fecha Llegada**, **Nro Lance**, **Cod Arte**, **Id Captura**, **Cod Especie Animal**, Cantidad\_pesca)

## Conclusiones

En este capítulo se muestra como se trabajó desde el comienzo del análisis hasta la obtención del diseño lógico del DW. Se muestra como al partir de una Metadata Genérica, una realidad particular con sus Bases de Datos definidas y un análisis de requerimientos se puede llegar al diseño lógico de un DW pronto para ser implementado.

Si bien el ejemplo presentado es sólo una porción de la realidad modelada durante el desarrollo del proyecto, se quiso abreviar el mismo, tratando de mostrar paso a paso que fue lo que se realizó sin extenderse demasiado en detalles. El caso de estudio total es mucho más extenso, y se adjunta en los anexos. Brindando en este capítulo un breve ejemplo de todos los pasos aplicados.

Se puede observar que esta metodología brinda facilidades para obtener un modelo conceptual completo para cualquier realidad, cabe destacar que ésta no se puede automatizar totalmente ya que se necesita una fuerte interacción del usuario para dar semántica a los objetos que pretende diseñar.

## 4.6. Conclusiones y trabajos futuros

En el capítulo 1 se puede apreciar la evolución de un proyecto que tenía planteado llegar desde una MG al diseño lógico de un DW y su posterior aplicación para instrumentar una herramienta que permitiera consultas multidimensionales.

Esta evolución marcó que la falta de una metodología adecuada y un proceso de desarrollo puramente intuitivo determinaron que hubieran avances y retrocesos al diseñar el DW.

Para resolver estas dificultades se define un Método que permite, en forma completa, a partir de una MG, un análisis de requerimientos y una posible OLTP, llegar al Modelo Lógico de un DW, pasando por una MP y un Modelo Conceptual.

En forma gráfica, a continuación se presenta un resumen de la metodología aplicada, así como su resultado.

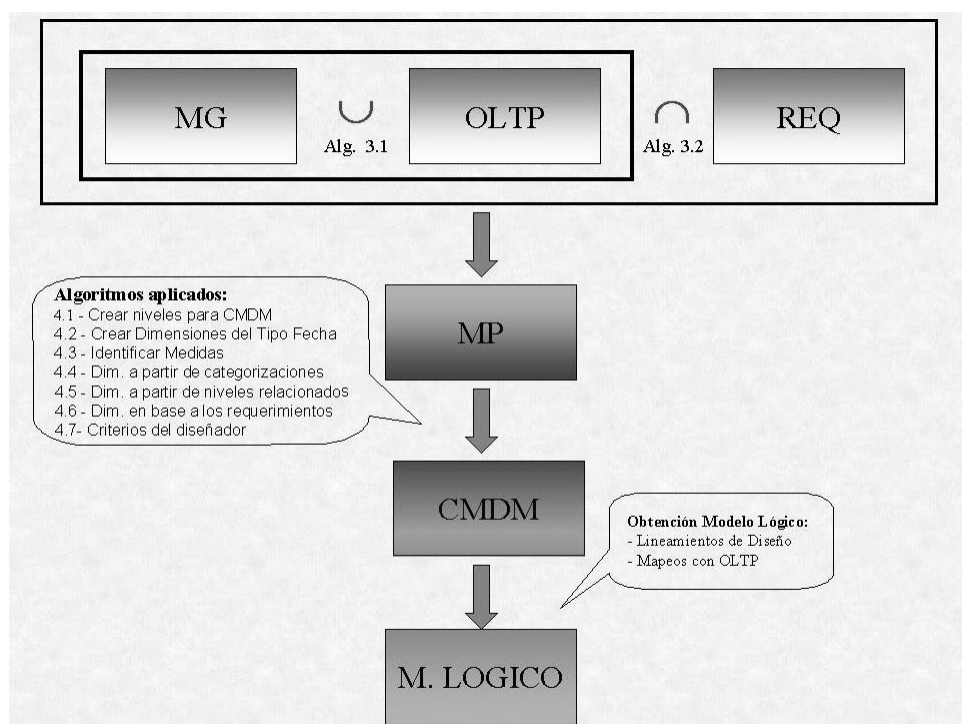


Fig. 6.1 – Resumen considerando OLTP definida

Lo implementado en este proyecto, son los algoritmos de intersección y de unión, de forma de obtener la MP. También se implementaron los algoritmos que permiten pasar de la MP al CMDM. Se aplicó la metodología existente para el pasaje del CMDM al Modelo Lógico. Para representar la MG, la OLTP, los Requerimientos y la MP se realizaron un conjunto de definiciones de forma de expresarlas de manera homogénea.

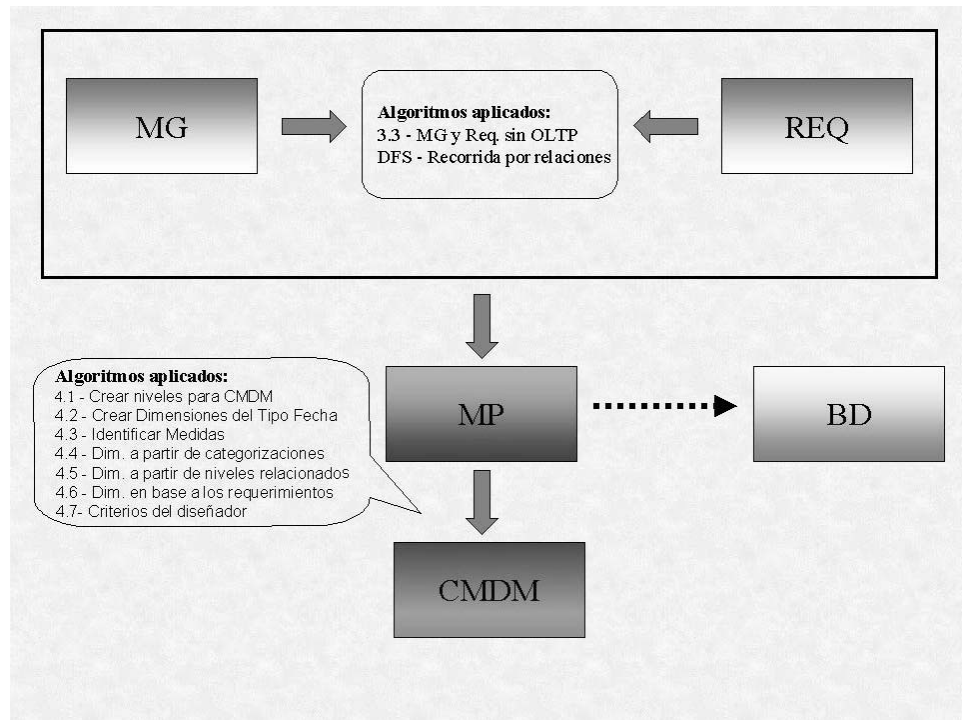


Fig. 6.2 - Resumen sin considerar OLTP

En este caso se implementa el algoritmo 3.3 que no consideran la OLTP, y permite obtener una MP. Los algoritmos para el pasaje a CMDM, son los ya explicados anteriormente. En este caso no se llega al Modelo Lógico ya que para eso es necesario tener la OLTP definida, pero se puede inferir desde la MP, la Base de Datos (BD) a crear de forma de satisfacer los requerimientos del DW que se plantearon.

Un punto a destacar es que el desarrollo y formalización de una MP brinda al diseñador un conocimiento más acabado de la realidad y los requerimientos.

La aplicación del Método definido, en el Capítulo 3, integra datos y requerimientos expresados en términos que les son comunes. Conjuntamente con la metodología allí vista permite, de una manera natural, que datos y requerimientos, restrinjan o amplíen una MP y que acerquen las estructuras de datos iniciales al Modelo Lógico final. Aunque el caso de estudio al que se aplicó esta metodología, estuvo fuertemente ligado a la estructura de la OLTP existente, este Método permite dar origen al diseño de bases de datos transaccionales a partir de la MP obtenida.

En el Capítulo 4 se obtuvo un conjunto de métodos para orientar la tarea del diseñador al momento de definir el Modelo Conceptual. Los métodos definidos no son del todo rigurosos de forma de poder contemplar la diversidad de escenarios posibles. Un ejemplo del porque de esto se planteó en el transcurso del proyecto al tratar de modelar las zonas geográficas (Anexo II).

El conjunto de métodos obtenidos se podrá enriquecer tanto por aportes de la aplicación de los mismos a otros proyectos, como por su especialización en diferentes ramas de “negocios” en los que se apliquen.

Para completar el trabajo presentado en este documento sería necesario prototiparlo y hacer de él un elemento más formal. Como por ejemplo el desarrollo de una herramienta CASE que partiendo de esta metodología simplifique el diseño de un DW desde sus inicios. A esta herramienta CASE se podrían incorporar el diseño conceptual y lógico del DW que son los utilizados en este documento, o interconectarse con las herramientas propuestas en los trabajos relacionados [FC00] [VP01] .

## Bibliografía y Lecturas Relacionadas

- [Bati92] C. Batini, S. Ceri and S. Navathe: "Conceptual Database Design: An ER Approach" The Benjamin-Cummings Publishing Company, 1992.
- [Bouz94] M. Bouzeghoub, G Gardarin and P. Valduriez: "Du C++ à Merise Objet – OBJETS", Ed. Eyrolles, 1994.
- [Chen76] Chen: "The Entity-Relationship Model : Towards a Unified View of Data". ACM-TODS, Marzo 1976.
- [CH00] ClearingHouse Nacional de Datos Geográficos Uruguay: "Estándar de Contenido de Metadatos Geoespaciales", Mayo 2000  
<http://www.clearinghouse.gub.uy/ppoint/metaprim/metaprim.html>
- [CR01] F.Carpani, R. Ruggia: "An Integrity Constraints Language for a Conceptual Multidimensional Data Model", 2001  
<http://www.fing.edu.uy/inco/grupos/csi/esp/Publicaciones/2001/seke2001-fc.pdf>
- [FC00] Fernando Carpani: "CMDM: Un Modelo Conceptual para la Especificación de Bases Multidimensionales". Tesis de maestría Agosto 2000  
<http://www.fing.edu.uy/inco/grupos/csi/esp/Publicaciones/2000/tesis-fc.pdf>
- [RR96] Raúl Ruggia: "Applying Reuse in Conceputal Modelling: a toolkit approach". Tesis de Doctorado, Universidad Paris VI - Francia. Abril 1996.
- [VP01] Verónica Peralta: "Diseño Lógico de un Data Warehouse a partir de un Esquema Conceptual Multidimensional". Tesis de maestría Noviembre 2001  
<http://www.fing.edu.uy/inco/grupos/csi/esp/Publicaciones/2001/tesis-vp.pdf>
- [VP02] Verónica Peralta: "Un Escenario para Diseño Lógico de Data Warehouses", 2002  
<http://www.fing.edu.uy/inco/grupos/csi/esp/Publicaciones/2002/jcc-wbd2002-vp.pdf>

## **4.7. Anexos**

### **Anexo I**

Definición de XML-Schema para las metadatos y requerimientos

### **Anexo II**

Aplicación de la Metodología para obtención del CMDM al caso de estudio completo

### **Anexo III**

Diseño Lógico de un Data Warehouse aplicado al caso de estudio

### **Anexo IV**

Reportes Técnicos y Trabajos Relacionados